

# **Big Data Phylogenomics: Methods and Applications**

---

A Dissertation  
Submitted to  
the Temple University Graduate Board

---

In Partial Fulfillment  
of the Requirements for the Degree  
DOCTOR OF PHILOSOPHY

---

by  
Sudip Sharma  
December 2023

Examining Committee Members:

Dr. Sudhir Kumar, Advisory Chair, Department of Biology  
Dr. S. Blair Hedges, Examining Committee Chair, Department of Biology  
Dr. Sergei Pond, Examining Committee Member, Department of Biology  
Dr. Xinghua Shi, External Member, Department of Computer and Information Science,  
Temple University

## ABSTRACT

Phylogenomics, the study of genome-scale data containing many genes and species, has advanced our understanding of patterns of evolutionary relationships and processes throughout the Tree of Life. Recent research studies frequently use such large-scale datasets with the expectation of recovering historical species relationships with high statistical confidence. At the same time, the computational complexity and resource requirements for analyzing such large-scale data increase with the number of genomic loci and sites. Therefore, different crucial steps of phylogenomic studies, like model selection and estimating bootstrap confidence limits on inferred phylogenetic trees, are often not feasible on regular desktop computers and generally time-consuming on high-performance computing systems. Moreover, increasing the number of genes in the data increases the chance of including genomic loci that may cause biased and cause fragile species relationships that spuriously receive high statistical support. Such data errors in phylogenomic datasets are major impediments to building a robust tree of life. Contemporary approaches to detect such data error require alternative tree hypotheses for the fragile clades, which may be unavailable a priori or too numerous to evaluate. In addition, finding causal genomic loci under these contemporary statistical frameworks is also computationally expensive and increases with the number of alternatives to be compared. In my Ph.D. dissertation, I have pursued three major research projects: (1)

Introduction and advancement of the bag of little bootstraps approach for placing the confidence limits on species relationships from genome-scale phylogenetic trees. (2) Development of a novel site-subsampling approach to select the best-fit substitution model for genome-scale phylogenomic datasets. Both of these approaches analyze data subsamples containing a small fraction of sites from the full phylogenomic alignment. Before analysis, sites in a subsample are repeatedly chosen randomly to build a new alignment that contains as many sites as the original dataset, which is shown to retain the statistical properties of the full dataset. Analyses of simulated and empirical datasets exhibited that these approaches are fast and require a minuscule amount of computer memory while retaining similar accuracy as that achieved by full dataset analysis. (3) Development of a supervised machine learning approach based on the Evolutionary Sparse Learning framework for detecting fragile clades and associated gene-species combinations. This approach first builds a genetic model for a monophyletic clade of interest, clade probability for the clade, and gene-species concordance scores. The clade model and these novel matrices expose fragile clades and highly influential as well as disruptive gene-species candidates underlying the fragile clades. The efficiency and usefulness of this approach are demonstrated by analyzing a set of simulated and empirical datasets and comparing their performance with the state-of-the-art approaches. Furthermore, I have actively contributed to research projects exploring applications of these newly developed approaches to a variety of research projects.

*To my parents, Santosh Sharma and Sankori Sharma*

*To my uncle, Sankor Sharma*

*To my wife, Mousumi Datta*

*I am immensely grateful for all your unwavering support  
and encouragement throughout my academic journey  
and the love and tolerance you show at every step of my life.*

## **ACKNOWLEDGMENTS**

I want to begin by paying tribute to my father, whom I lost in April 2021 during my graduate study. His support, guidance, and moral values profoundly influenced my life and academic journey. His vision for life, uncompromising honesty, and remarkable personality influenced me to become the person I am today. He would have been the happiest person to see me complete my Ph.D. I cherish my memories with him on this significant milestone in my academic career.

I am incredibly grateful for the opportunity of pursuing my doctoral studies and research under the mentorship of my thesis advisor, Dr. Sudhir Kumar. His expertise in molecular evolution and computational phylogenetics and his enthusiasm for scientific research have shaped my academic research and philosophy. The transition from an Applied Statistics graduate to a Ph.D. student in Statistical Molecular Evolution would not have been possible without his mentorship and continuous guidance. I also want to thank Dr. Sabysachi Das, who introduced me to Dr. Kumar when I began applying to graduate schools in the US. I am also grateful for all the facilities and intellectually stimulating environment at the Institute for Genomics and Evolutionary Medicine (iGEM) at Temple University, where I have interacted with brilliant minds and gained invaluable experiences throughout my journey.

I also want to express my gratitude and thank my thesis committee for their support and valuable suggestions: Dr. Blair Hedges, Dr. Sergei Pond, and Dr. Mindy Shi. In addition, I would like to express my appreciation to Dr. Richard Waring and Sandhya Verma for their continued assistance, which has been instrumental in ensuring a smooth progression in my graduate studies.

I thank my fellow current and past labmates in Kumar lab at iGEM: Dr. Sayaka Miura, Dr. Marcos Caraballo-Ortiz, Dr. Qiqing Tao, Dr. Ravi Patel, Dr. Jose Barba, Dr. Caryn Babaian, Dr. Alessandra P. Lamarca, Lisa Schmelkin, Jhon Allard, Julia Davis, and Sara Vahdatshoar. I also want to thank Glen Strecher and Maxwell Sanderford for much technical support that helped me in my dissertation research. I also thank the Professors and staff at the Department of Biology and Kumar lab for making such a friendly and pleasant working environment.

Finally, I would like to extend my deepest gratitude to my wife, whose support and sacrifices have been a constant source of strength throughout my doctoral journey. She selflessly put her well-established career on hold to be by my side, providing support and encouragement. I am genuinely grateful for her love, understanding, and belief in my aspirations. I would also like to thank my parents-in-law for letting their daughter accompany me on this journey.

# TABLE OF CONTENTS

<b>ABSTRACT</b> .....	<b>ii</b>
<b>ACKNOWLEDGMENTS</b> .....	<b>v</b>
<b>LIST OF TABLES</b> .....	<b>x</b>
<b>LIST OF FIGURES</b> .....	<b>xi</b>
<b>CHAPTER</b>	
<b>1. INTRODUCTION</b> .....	<b>1</b>
<b>2. FAST AND ACCURATE BOOTSTRAP CONFIDENCE LIMITS ON GENOME-SCALE PHYLOGENIES USING LITTLE BOOTSTRAPS</b> .....	<b>10</b>
2.1 Introduction .....	10
2.2 The little bootstraps approach .....	12
2.3 Performance of little BS for computer-simulated datasets .....	15
2.3.1 Simulation protocol .....	15
2.3.2 Standard and Little Bootstraps analyses.....	16
2.3.3 Computational resource savings.....	17
2.3.4 Little BS with median-bagging .....	18
2.3.5 Phylogenomic subsampling approaches without upsampling.....	20
2.3.6 Automatic parameter tuning and the Precision of BCL estimates .....	22
2.3.7 The little bootstraps analysis pipeline. ....	23
2.3.8 Analysis of empirical datasets.....	24
2.3.9 Combining Little BS with other optimizations .....	28
2.4 Conclusion .....	28

**3. TAMING THE SELECTION OF OPTIMAL SUBSTITUTION MODELS IN PHYLOGENOMICS BY SITE SUBSAMPLING AND UPSAMPLING ..... 29**

3.1 Introduction ..... 29

3.2 Methods..... 31

    3.2.1 The approach of upsampling sites from subsamples ..... 31

    3.2.2 Minimum subsample size for efficient model selection..... 33

3.3 Adaptive tool for model selection ..... 37

3.4 Evaluation of ModelTmaer Performance..... 38

    3.4.1 ModelTamer analysis. .... 38

    3.4.2 ModelTamer performance for empirical datasets ..... 39

    3.4.3 ModelTamer performance for simulated datasets ..... 40

    3.4.4 The efficiency of ModelTamer for partitioned data ..... 45

    3.4.5 Estimating model parameters using SU datasets..... 46

3.5 Conclusions ..... 46

**4. MACHINE LEARNING DETECTS FRAGILE CLADES AND CAUSAL SEQUENCES IN PHYLOGENOMIC DATASETS ..... 48**

4.1 Introduction ..... 48

4.2 Estimating Gene-species Concordances and Clade Probabilities ..... 52

    4.2.1 Gene-Species Concordance (GSC) metric ..... 53

    4.2.2 Clade Probability (CP) ..... 54

4.3 DrPhylo Implementation..... 55

4.4 Analysis of empirical datasets..... 57

    4.4.1 Analysis of empirical fungi dataset ..... 57

    4.4.2 The model grid (M-grid) ..... 60

    4.4.3 An analysis of an expanded fungi dataset ..... 62

    4.4.4 ESL analysis of a “control” fungi clade. .... 64

4.5 Analysis of additional empirical datasets..... 66

    4.5.1 Analysis of plants dataset ..... 66

    4.5.2 Analysis of an animal phylogeny ..... 68

4.6 Conclusion ..... 69

<b>5. EXPLORING NEW FRONTIERS: FUTURE DIRECTIONS FOR NEW METHODS .....</b>	<b>72</b>
5.1 Introduction .....	72
5.2 Non-parametric one-sample test using Little Bootstrap .....	73
5.3 Application of Little Bootstraps for Divergence Time Estimation .....	77
5.4 Detecting the genomic signature of convergent evolution.....	80
5.5 Conclusion .....	83
<b>BIBLIOGRAPHY .....</b>	<b>84</b>
<b>APPENDICES .....</b>	<b>99</b>
<b>A. R CODES FOR AUTOMATIC LITTLE BOOTSTRAPS ANALYSIS USING IQTREE .....</b>	<b>99</b>
<b>B. R CODES FOR AUTOMATIC MODEL TAMER ANALYSIS USING IQTREE .....</b>	<b>108</b>

## LIST OF TABLES

2.1	Summary of Little Bootstraps analysis.....	27
3.1	Model selection using full and Subsampled-upsampled (SU) datasets. ....	36
3.2	ModelTamer analysis for empirical and simulated datasets.....	41

## LIST OF FIGURES

2.1.	Computational Complexity of ML analysis.....	11
2.2.	Overview of the Little Bootstraps Approach.....	14
2.3.	Advantages of Median Bagging.....	17
2.4.	Accuracy of Little Bootstraps for small datasets.....	19
2.5.	Limitations of the subsampling approach.....	21
2.6.	Subsample size and Precision.....	23
3.1.	Computational resource requirements for model Selection.....	30
3.2.	Determination of subsample size.....	32
3.3.	Advantages of upsampling.....	33
3.4.	Computational resource savings.....	35
3.5.	Performance of ModeTamer.....	44
4.1.	A schematic outlining of DrPhylo analysis pipeline.....	56
4.2.	Clade model for fungi phylogeny.....	59
4.3.	The Model-grid (M-grid) for clade A+B.....	61
4.4.	Clade model extended fungi dataset.....	63
4.5.	Simulated data in the control branch.....	65
4.6.	Analysis of plants and animal datasets.....	67
5.1.	Accuracy of little bootstraps with mean and median bagging.....	76
5.2.	Time requirements for joint analysis in Bayesian framework.....	78

# CHAPTER 1

## INTRODUCTION

Inferring historical relationships of species evolution is the key to evolutionary analyses in many fields of biological research like molecular evolution<sup>1</sup>, ecology<sup>2</sup>, microbiology<sup>3</sup>, conservation<sup>4,5</sup>, biochemistry<sup>6</sup>, biotechnology<sup>7</sup>, and bioinformatics<sup>8</sup>. The inferred evolutionary relationships of present-day species help us to understand the mode and tempo of evolution<sup>9</sup>, classification of newly found species<sup>10</sup>, track the origin and spread of pathogens<sup>11</sup>, estimate the divergence time of species evolution<sup>12,13</sup>, the emergence of new pathogens<sup>14</sup>, identify endangered species for conservation<sup>15</sup>, track the evolution of tumor cells in cancer<sup>8,16</sup>, annotate biological function for genes<sup>17</sup>, and detecting genomic signature convergence of trait in independently evolved species<sup>18</sup>.

The history of inferring phylogenetic trees of species dates back to the mid-19th century when the Tree of Life (ToL) was proposed independently by Charles Darwin and Ernst Haeckel<sup>19,20</sup>. In the early days, the study of species evolution was initially dominated by building phylogenetic trees of living organisms using their morphological characters or fossil records<sup>7,21,22</sup>. Even though morphological data provide the initial view of how species are related to each other by their shared morphological characteristics, the convergence of traits that evolve independently in many distantly related species, incomplete fossil records, the ambiguity of morphological characters, sexual dimorphisms, and subjective selection

of morphological characters limits its ability to build reliable species tree reconstruction<sup>23</sup>. A crucial limitation of morphological characters is the lack of universal comparability across various species in the Tree of Life.

Therefore, the unambiguous nature of molecular sequences (protein or nucleotide) among all organisms was foreseen as the basis for inferring phylogenetic relationships of species<sup>24,25</sup>. Fitch and Margolish showed that estimating a phylogenetic tree using molecular sequence data is similar to the parameter estimation problem in the statistics and presented a statistical framework for estimating a phylogenetic tree by analyzing molecular sequence data<sup>26</sup>. Due to the limited data in the early days, the inference of phylogenetic trees relied on a single gene or locus<sup>27,28</sup> or smaller genome segments<sup>28</sup>. Phylogenetic trees inferred from smaller datasets provided exciting insights into species relationships<sup>29,30</sup>. For instance, by comparing the amino acid sequence of immunoglobulin proteins, scientists discovered that humans were closely related to both chimpanzees and gorillas (trichotomy) but distantly related to orangutans<sup>28,30</sup>.

In phylogenetics, inferring a phylogenetic tree from molecular sequence data estimates the true evolutionary relationships of species<sup>31,32</sup>. This is analogous to parameter estimation in statistical inference<sup>31</sup>. A parameter estimate obtained by analyzing a small dataset has a higher variance. As a result, the variation in inferred phylogenetic trees across studies was frequently observed as those were estimated from smaller genomic datasets. Consequently, the species clades in the inferred phylogenetic trees received low statistical support (e.g., low bootstrap supports). While molecular sequence data helped to re-establish the evolutionary relationships of numerous species, we observed less reconciliation of estimated trees of the same set of species among studies while analyzing small molecular

sequence data. Furthermore, phylogenetic signals from small datasets can be obscured by data noise in smaller datasets which ameliorates the phylogenetic tree inference. Therefore, the sampling variance and stochastic error were the major impediments to establishing a well-established tree of life in the small data era of molecular phylogenetics.

One potential solution to this problem was to analyze large-scale genomic data (genome-scale dataset) containing many gene sequences from the studied organisms. Statistically, the sampling variance in the phylogeny inference decreases with the increasing data size. Therefore, analyzing many genomic loci is expected to reduce the standard error of the estimated phylogenetic tree. Consequently, the inferred species relationships receive high statistical support (e.g., high bootstrap support), leading to a well-resolved species phylogeny<sup>7,33-35</sup>. One of the earliest instances of using big genomic datasets includes resolving the trichotomy of human, chimpanzee, and gorilla relationships by analyzing 10kbp-long nucleotide sequence data<sup>26</sup>. Many recent studies also demonstrated the potential of analyzing large-scale genomic loci to establish species relationships throughout the Tree of Life<sup>36-38</sup>.

With the advancement of sequence technology, analyzing multigene and multi-species genomic data is becoming commonplace in evolutionary biology<sup>27-29</sup>. The comparative study of genome-scale sequences to recover historical relationships of present-day species as well as other evolutionary parameters is usually referred to as phylogenomics. Although the term initially originated as a study of protein function using a phylogenetic tree<sup>30,31</sup>. Therefore, phylogenomics can be defined as an interplay between inferring species relationships by analyzing genome-scale datasets and investigating more biological and functional insights of different loci (e.g., genes) using the inferred species relationships<sup>25</sup>.

While inferring species relationships with high statistical confidence remains the main focus, phylogenomics also demonstrated the potential to determine functions and functional interactions among many proteins<sup>15</sup>. Phylogenomics also helps to recover and establish many well-accepted species relationships that have transformed our knowledge of evolutionary patterns and processes throughout the Tree of Life (ToL). Analyzing multigene and multi-species sequence data also helps us to reduce stochastic and systematic errors often encountered when analyzing small datasets<sup>39,40</sup>. Overall, phylogenomics revolutionized our understanding of species evolution and its application in many fields of biological research. However, researchers still encounter many analytical challenges when analyzing such large-scale datasets.

One major challenge for analyzing big phylogenomic datasets is the increasing requirements of computational resources. The process of inferring species trees involves multiple steps, including selecting a best-fit substitution model, inference of a phylogenetic tree, and estimating statistical support using non-parametric bootstrap<sup>7,32</sup>. Statistical framework, especially widely used maximum likelihood (ML) based approaches, for performing these crucial steps become extensively slow and require prohibitive computer memory. The computational resource requirements exhibit an exponential increase with the number of sequences in the multiple sequence alignments (MSA) and linear increment for the number of sites<sup>33,34</sup>. For example, the model selection of a concatenated sequence alignments 4,682 proteins from 58 vertebrate species required 138 GB (gigabyte) of computational memory and 4,604 CPU hours (~192 CPU days)<sup>34</sup>. Estimating the statistical support using the non-parametric bootstrap approach for the inferred species clades using the best-fit model also required prohibitive memory and computational time<sup>41</sup>. Thus,

performing these crucial steps for such phylogenomic datasets with a moderate number of species and a large number of genomic loci becomes prohibitive even on a cluster computing system<sup>33,34,41</sup>.

Over the years, different heuristics have been developed to accelerate the ML tree search, leading to fast model selection and bootstrap analysis for large-scale datasets with a large number of taxa in the MSA. This class of heuristics includes Nearest Neighbor Interchanges (NNI)<sup>42</sup>, Subtree pruning and regrafting (SPR)<sup>43</sup>, Lazy subtree rearrangements<sup>44</sup>, and Tree bisection and reconnection (TBR)<sup>43</sup>. These heuristics are commonly used to reduce the tree search space and are efficiently implemented in tree inference software like MEGA<sup>45</sup>, IQTREE<sup>46</sup>, or RAxML<sup>47</sup> to accelerate the ML tree estimation and eventually bootstrapping.

Another class of approaches includes methods that analyze subsets of the full MSA. These approaches can be defined as the Divide-and-Conquer (DaC) approach. In DaC, datasets are divided into small subsets of sequences or sites/genes. Phylogenetic analyses are performed on these small subsets and ensembled results from each subset for estimating the parameter of interest. It is hypothesized that a DaC approach can recover the parameter estimate similar to the full MSA. The DaC approaches are performed by dividing the number of taxa into small subsets or the number of loci/sites into small subsets.

The first type of DaC approach is employed to reduce the computational burden of inferring species trees from a large number of sequences. In these approaches, an MSA is divided into multiple small subsets of sequences subject to tree inferences. Such sequence subsampling methods are usually performed to accelerate species tree inferences<sup>48-51</sup>. Another type of DaC approach includes subsampling of genomic loci or sites that primarily

focus on assess the robustness of species groupings in the inferred phylogeny<sup>34,35</sup>. Therefore, phylogenetic subsampling was defined as the analysis of smaller subsets (site or loci) of full MSA to assess the robustness of the inferred species relationships<sup>36-39</sup>. Subsampling of sites or loci is usually performed in three ways: i) subsampling of loci, ii) subsampling of sites, and iii) subsampling of both loci and sites from the full MSA<sup>36,40</sup>.

Subsampling of smaller subsets of sites from the MSA is commonly used to investigate the impact of loci's gradual addition or removal on clade stability<sup>38,52</sup>. These approaches are analogous to the estimation of clade support in an inferred phylogeny. Supertree approaches can also be classified as a type of subsampling approach as it estimates gene trees from each genomic loci and ensembles them to infer the final species tree<sup>41-44</sup>. However, none of these approaches were developed solely for reducing the computational burden imposed by a large number of genomic loci in phylogenomic datasets. Moreover, the lack of adaptive choice for the number and size of subsamples to be analyzed make these approaches computationally scalable for such big datasets.

Another challenge researchers encountered while analyzing phylogenomic datasets is the presence of data errors which may be attributed to one or few genetic loci with an extreme phylogenetic resolution for a particular species clade in the inferred phylogeny. The outlier influence of such a few loci masks the signal from most of the loci in the dataset and infers biased species relationships<sup>41,53,54</sup>. These species relationships may receive spuriously high statistical support but remain fragile. The presence of such bias-causing loci in the dataset is one of the major impediments to reconciliation among studies. Detecting such data errors is essential for establishing a well-resolved tree of life.

In my dissertation research, I have worked on developing statistical and machine-learning approaches that are computationally efficient and provide potential solutions to these problems. We introduced novel subsampling-upsampling-based approaches as a new paradigm of phylogenomic analyses for selecting best-fit substitution models and performing non-parametric bootstrapping for estimating statistical support for inferred species relationships. I have also worked on developing a machine-learning-based framework for building genetic models for monophyletic species clades which were used for detecting highly influential genes underlying fragile species relationships. In addition, I was also involved in research projects focused on exploring diverse applications of these newly developed approaches.

In Chapter 2, I present the little bootstrap approach to place confidence limits on ML phylogenies inferred using long multiple sequence alignment. Little Bootstrap approach performs bootstrapping on small subsamples of sites, and each bootstrap replicates from a subsample are generated by random sampling with replacement, and the replicate has the same length as the full MSA. By analyzing many simulated and empirical datasets, it was shown that the summary of subsample confidence limits which were estimated by bootstrapping on subsampled MSA produces accurate bootstrap confidence for phylogenetic relationships in a small fraction of computational time and memory. An automatic procedure has also been developed for choosing the subsample size, the number of subsamples, and the number of replicates per subsample adaptively. Little Bootstraps enhances rigor, efficiency, and parallelization in big data phylogenomics, even on personal computers.

Chapter 3 discusses an ingenious phylogenomic subsample method in which site subsampling is coupled with an upsampling approach that proved to be a powerful, accurate, and computationally-efficient approach for model selection, overcoming the known deficiencies of phylogenomic subsampling. We show that only a small representative fraction of all unique site patterns, which can be determined automatically from a long phylogenomic alignment, contain the necessary phylogenetic information to accurately select the optimal substitution model and estimate its rate parameters. This advance is implemented in a software tool, ModelTamer, which automatically decides subsamples to infer optimal substitution models hundreds to thousands of times faster while needing only megabytes rather than gigabytes of computer memory. Consequently, researchers with even commodity computers will be able to conduct big data analysis on their desktops, and those utilizing high-performance computing infrastructure will benefit by achieving greater calculation parallelization because of the very small memory footprint of ModelTamer. These computational advances will promote higher scientific rigor, broader participation, and environment-friendly computing in molecular evolutionary research.

Chapter 4 presents the utility of supervised machine learning in phylogenomics. Phylogenomic analysis of hundreds to thousands of genes or genomic segments from multiple species may fail to reconstruct organismal relationships with high statistical support. The disproportionate influence of a few genes within certain species can significantly skew the inferred phylogeny, overshadowing useful evolutionary signals from numerous other genes. Evolutionary sparse learning (ESL), a supervised machine learning approach, builds a clade-specific genetic model with highly influential genes in the

alignments and detects genes causal for fragile clades without requiring any alternative phylogenetic hypothesis – a common requirement for Maximum Likelihood (ML) or Bayesian approach. Different novel metrics, a gene-species concordance, and a clade probability metric have been developed. Upon analyzing three empirical datasets (fungi, plants, and animals), we discovered that these metrics could efficiently and effectively pinpoint fragile clades and associated gene-species combinations, both known and novel.

Chapter 5 explores the broader applicability of the newly developed approaches proposed in this thesis. I discuss the general applicability of the little bootstrap approach, showcasing its utility in scenarios such as the one-sample t-test in statistical hypothesis testing and estimating species divergence time in the presence of phylogenetic uncertainty. In addition, I have discussed the potential of evolutionary sparse learning (ESL) models in identifying the genomic signatures of molecular convergence of trait evolution.

## CHAPTER 2

# FAST AND ACCURATE BOOTSTRAP CONFIDENCE LIMITS ON GENOME-SCALE PHYLOGENIES USING LITTLE BOOTSTRAPS

### 2.1 Introduction

The standard bootstrap approach<sup>55</sup>, introduced more than 35 years ago by Joseph Felsenstein<sup>56</sup> in phylogenetics, has been the standard method to assess the robustness of inferred molecular phylogenies<sup>57</sup>. This approach has been applied in over forty thousand research articles to place confidence limits on species clades in an inferred phylogeny<sup>57</sup>. In standard bootstrap, pseudo multiple sequence alignments (MSA) are generated by random sampling of sites with replacement, and each MSA is considered as a bootstrap replicate dataset. A phylogenetic tree is inferred using commonly used statistical approaches<sup>56,58</sup> (e.g., Maximum Likelihood) from each of the bootstrap replicates (Figure 1a). If a group

of sequences is recovered in a large proportion of bootstrap phylogenies (bootstrap confidence limit, BCL), their evolutionary relationship is considered well-supported<sup>56,58</sup>.

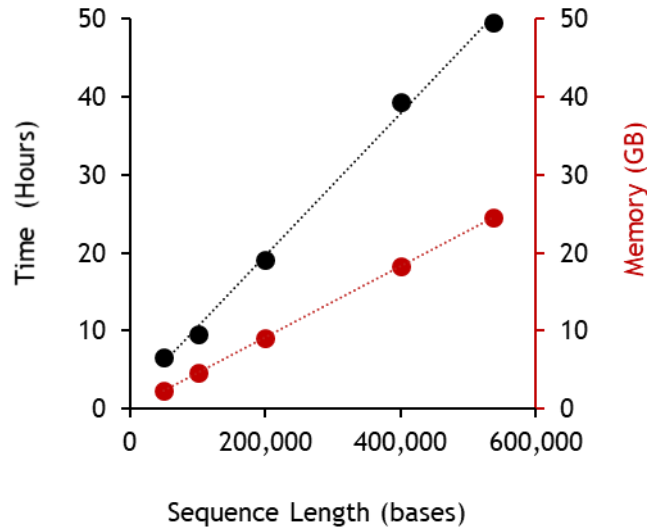


Figure 2.1. Computational Complexity of ML analysis. The computational time and memory requirements for ML tree inference of a bootstrap replicate increases with the number of bases in the full MSA. The X-axis represents the number of bases in the MSA, time (Black) and memory (Red) requirements for the ML analysis are represented in Y-axis.

Due to the widespread accessibility of genome sequence databases and the assembly of multispecies and multigene alignments containing hundreds of thousands of bases is commonly used in phylogenetic analysis (e.g.,<sup>38,59,60</sup>), the standard bootstrap (BS) approach is being applied to increasingly larger datasets. These large datasets have the power to reconstruct hard-to-resolve evolutionary relationships with high confidence ( $BCL \geq 95\%$ )<sup>7,38,59,61,62</sup>. However, the increasing number of sites or loci in the phylogenomic dataset imposes increasingly onerous computational demands because the computational complexity of phylogenomic analyses using the maximum likelihood (ML) method increases exponentially with the number of sequences and linearly with sequence length<sup>63</sup>. Consequently, the standard BS resampling procedure can take days to complete for big

datasets<sup>59,63</sup> as it requires estimating an ML phylogenetic tree from each of the bootstrap replicates (Figure 2.1).

Many heuristics have been proposed to moderate the escalation due to the increasing number of sequences (e.g., ref.<sup>63,64</sup>). Even though different types of subsampling procedures are used for multi-locus datasets for assessing the robustness of species clades in the inferred phylogeny<sup>38,52</sup>, the requirement for the locus-partition and the ad-hoc specification of the size and number of subsamples to be analyzed make these approaches discovery limiting for genome-scale data. Therefore, no effective approaches are available to deal with the onerous computational burden imposed by an increase in sequence length due to the widespread adoption of next-generation sequencing methods. Thus, the standard BS approach's computational burden has become a new bottleneck in ensuring robust and reproducible phylogenomic analyses<sup>65,66</sup>.

## **2.2 The little bootstraps approach**

Kleiner et al. 2012 proposed a bag of little bootstraps<sup>67</sup> approach to overcome statistical limitations and computational burden of divide-and-conquer approaches<sup>52,67,68</sup>, most commonly the subsampled bootstrapping in statistics<sup>69,70</sup>. In the bag of little bootstrap, the dataset is divided into small subsamples containing a very small fraction of observation, and the bootstrapping is performed on these small subsamples. Finally, parameter estimates from subsamples are aggregated (e.g., mean bagging) to calculate the final estimate of the parameter<sup>67</sup>.

Here, we introduce the little bootstraps (little BS) approach to place confidence limits on molecular phylogenies inferred using sequence alignments. In little BS for phylogenetics,

bootstrapping is performed independently on  $s$  little datasets, each containing  $l$  sites sampled randomly without replacement from the full dataset with  $L$  sites ( $l \ll L$ ). A bootstrap confidence limit ( $bcl_i$ ) is estimated for each little dataset  $i$  by generating  $r$  phylogenies from bootstrap resampled datasets (Figure 2.2a).

In little BS, the bootstrap resampling of little subsample alignments is different from that of the standard BS, as  $L$  sites are sampled with replacement from  $l$  sites of the little subsample to build replicate datasets. Because  $l \ll L$ , the same site is selected many times (up-sampling) to build the bootstrap replicate dataset (Figure 2.2b). A replicate phylogeny is estimated for each little BS replicate dataset. Then, the bootstrap confidence limit ( $\widehat{BCL}$ ) for a given group of species is derived from  $s$  subsample-wise  $bcl$  values<sup>67</sup>, a procedure referred to as bagging. The average of  $s$  subsample  $bcl$  values, called mean-bagging ( $\widehat{BCL} = \frac{1}{s} \sum_{i=1}^s bcl_i$ ), was found to work well in general statistical analyses, including computer-simulated datasets<sup>67</sup>.

In the little BS approach, every site of the little subsample is included, on average,  $L/l$  times in the bootstrap replicate dataset. As these replicate datasets have the same number of sites as the full dataset, it obviates *ad hoc* corrections needed in other divide-and-conquer approaches and has other desirable asymptotic theoretical properties<sup>52,67,68</sup>. The computational burden of ML phylogeny estimation is proportional to the number of distinct site configurations, so each little BS replicate's time and memory requirements are of order  $O(L/l)$  needed for a standard BS replicate. Kleiner et al.<sup>67</sup> have suggested that little subsamples of size  $l = L^g$  ( $0.5 < g < 1.0$ ) can reduce time and memory by orders of

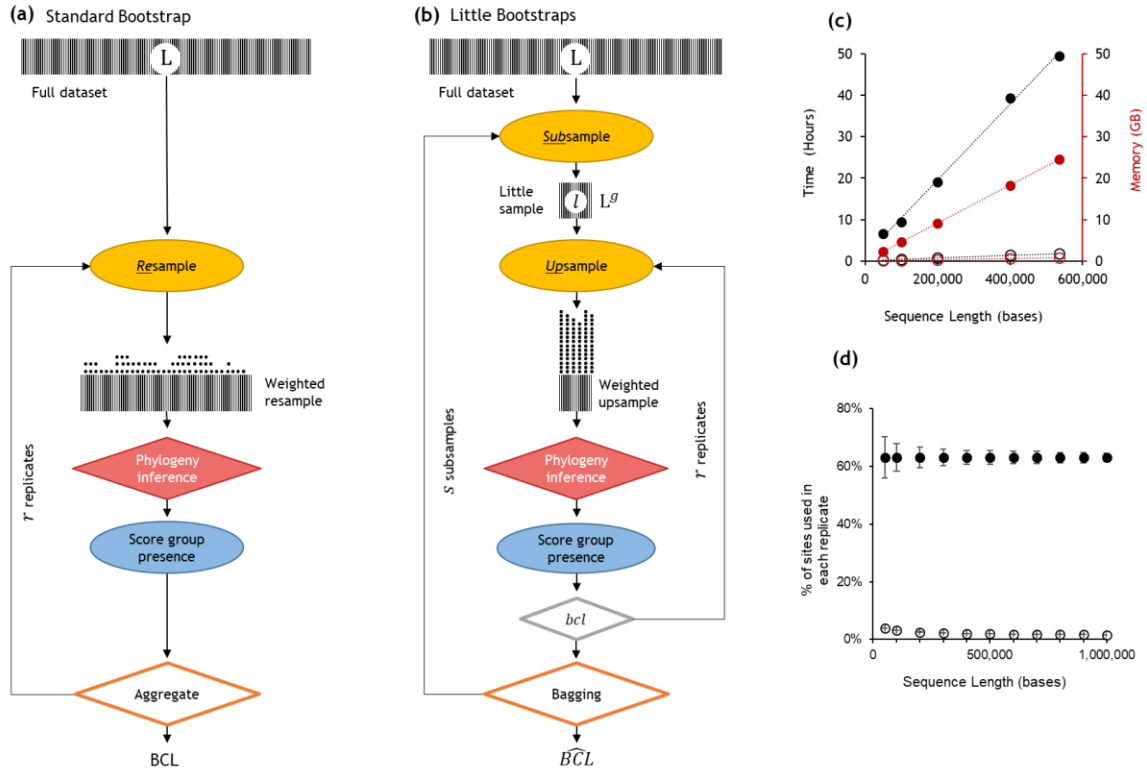


Figure 2.2. Overview of the Little Bootstraps approach. Steps of (a) the standard phylogeny bootstrap, and (b) the bag of little bootstraps (BS) approach. Shaded boxes represent sequence alignments in which denser hatching corresponding to a larger number of site configurations. The width of the box represents the sequence length. The generation of bootstrap replicate datasets differs between standard and little BS. In standard BS,  $L$  sites are randomly sampled with replacement from the original dataset containing  $L$  sites. In this resampling process,  $\sim 63.2\%$  of the data points are expected to be represented in a bootstrap replicate dataset. Each replicate dataset is compressed into weighted resamples that contain only distinct site configurations and a vector of their counts (represented by stacks of dots). In little BS,  $L$  sites are randomly sampled with replacement from the little dataset consisting of only  $l$  sites to build each replicate dataset. Because  $l \ll L$ , each site will be represented many times in the little bootstrap replicate dataset, which we refer to as upsampling. Stacks of dots are much higher for little BS due to upsampling than for standard BS, which involves only resampling. The number of distinct site configurations in the upsampled dataset is smaller than that in the standard bootstrap replicate dataset. Therefore, ML phylogeny for little BS replicates is expected to require less time and memory, as long as  $l$  is less than  $0.632L$  on average. (c) Time and memory savings per replicate of little bootstrap (open circles) compared to the standard bootstrap (closed circles) for large datasets. Simulated dataset contained 446 taxa and sequence length ranges from 50,000 to 536,534. (d) The proportion of sites included in the bootstrap replicates for little datasets with  $l = L^{0.7}$  (open circles) and standard bootstrap (closed circles) The choice of  $l = L^{0.7}$  offers increasingly greater computational savings for longer sequences because of a decreasing proportion of sites included in the little samples. For example, the little dataset size is  $\sim 3.1\%$  of the original alignment for  $L = 100,000$  bases, but it decreases to  $\sim 1.6\%$  when  $L$  increases 10-fold (1,000,000 bases). Overall, memory and time savings greater than  $\sim 95\%$  can be achieved for phylogenomic data with long sequences.

magnitude. In phylogenomics, these savings can be substantial (Figure 2.2c) and grows as the length of the sequence alignment increases from thousands to millions of sites for a given value of  $g$  (Figure 2.2d).

## **2.3 Performance of little BS for computer-simulated datasets**

### ***2.3.1 Simulation protocol***

Simulations are frequently used to test the accuracy of computational phylogenetic methods because the true evolutionary relationships are known<sup>71,72</sup>. So, we first present the results of ML phylogenetic analysis of a computer-simulated alignment containing 446 species and 134,131 sites. Multigene sequence alignments were assembled from a collection of simulated datasets analyzed in the previous studies<sup>72-75</sup>. These datasets were simulated using an evolutionary tree of 446 species<sup>72,76</sup>. A wide range of biologically realistic parameter values derived from empirical data<sup>72</sup> was used in simulating hundreds of gene alignments, including sequence length (445 – 4,439 bases), G+C content (39 – 82%), transition/transversion rate ratio (1.9 – 6.0), and gene-wise evolutionary rates (1.35 –  $2.60 \times 10^{-6}$  per site per billion years)<sup>72,73</sup>. Evolutionary rates were also heterogeneous across lineages, simulated for each gene independently under autocorrelated and uncorrelated rate models<sup>72,73</sup>. Simulated alignments of 100 genes that evolved with the autocorrelated rate model were concatenated to form the 446×134,131 (species x bases) dataset. The 446×536,524 sequence alignment was generated by concatenating sequence alignments generated by concatenating 100 randomly selected gene alignments from each of the four different lineage rate variation models simulated in ref.<sup>72</sup>. Three smaller datasets

were analyzed, corresponding to individual simulated genes: 446×4,070, 446×7,002, and 446×9,359 bases.

### 2.3.2 *Standard and Little Bootstraps analyses*

We performed the standard bootstrap analyses for these simulated datasets using IQTREE software<sup>46</sup> with a general time-reversible nucleotide substitution model with gamma-distributed rate variation (GTR+ $\Gamma$ )<sup>77,78</sup> and default ML search parameters. We conducted 100 standard BS replicates, an *ad hoc* convention adopted in many studies to make calculations feasible (e.g., ref.<sup>65</sup>). The confidence limits obtained using the standard bootstrap analyses were the ground truth in our analyses, as the bag of little bootstraps is being investigated as a computationally efficient alternative. The true tree used in computer simulations was the reference in the analysis of simulated datasets. For three single-gene datasets, 1,000 bootstrap replicates were used to generate stable BCLs. The confidence limits obtained using the standard bootstrap analyses were the ground truth in our analyses, as the bag of little bootstraps is being investigated as a computationally efficient alternative.

For the 446×134,131 dataset, we generated 100 little BS replicate datasets ( $s = 10$ ,  $r = 10$ ) with a subsample size of  $l = L^{0.7}$  (3,884 sites). For each of the little BS replicate, an ML tree was inferred using IQTREE software<sup>46</sup> using a general time-reversible nucleotide substitution model with gamma-distributed rate variation (GTR+ $\Gamma$ )<sup>77,78</sup>, and all other ML tree search parameters were used as default. These inferred ML trees from each replicate were used for mapping the Little BS support on the given phylogenetic tree.

### 2.3.3 Computational resource savings

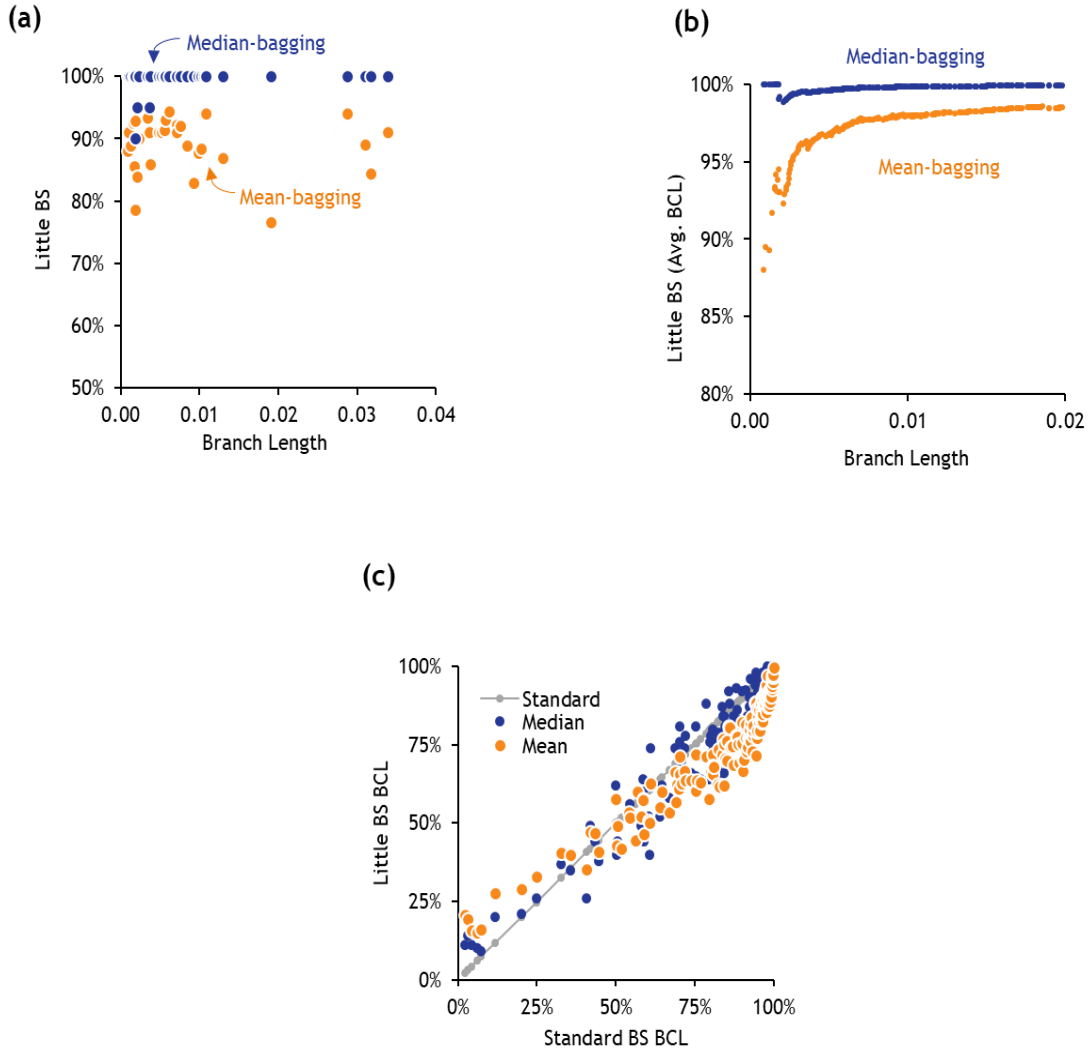


Figure 2.3. Advantages of Median Bagging. (a) The relationship of branch lengths and  $\widehat{BCL}$  produced by little BS with mean-bagging (golden) and median-bagging (purple) for  $l = L^{0.7}$ . The x-axis is restricted up to the branch length of 0.04 because  $\widehat{BCL} = 100\%$  for mean and median bagging for longer branches. (b) The distribution of  $bcl_i$ s for 49 species groups that received  $\widehat{BCL} < 100\%$  in little BS with mean-bagging analysis of large datasets. (c) The average  $\widehat{BCL}$  for all the species groups connected to the phylogeny with a given cutoff branch length (x-axis). The x-axis is restricted to 0.02 because mean- and median-bagging performance does not change any further for longer branches.

For the 446×134,131 dataset, it required 6.1 GB of memory and 13.1 CPU hours to estimate the ML phylogenetic tree using each BS replicate (54 CPU days of total computation). On the other hand, ML phylogeny inference of these little BS replicate datasets required, on average, only 0.3 GB RAM and 0.6 hours of CPU time, offering a 95% reduction in memory and in time. With these computational efficiency improvements, many small BS replicates could be run concurrently on a multicore desktop with 8 GB of RAM, unlike the standard bootstrap analyses that took up almost all the memory (6.1 GB) for estimating the ML phylogeny for one replicate dataset.

#### ***2.3.4 Little BS with median-bagging***

The standard bootstrap analyses established the true evolutionary relationships among sequences with very high confidence, i.e.,  $BCL \geq 95\%$  for all 443 correct species groupings from the 446×134,131 dataset. We first assessed the mean-bagging in little BS and found that little BS with mean-bagging did not produce  $\widehat{BCL} \geq 95\%$  for 32 species groups, which are false negatives (7.2%) because the standard BS supported all correct species groups at this  $BCL$  cutoff. These 32 species groups were connected with relatively short branches ( $< 0.04$  substitutions per site; Figure. 2.3a). Their confidence limits were underestimated by as much as 24% (Figure. 2.3b). We also observed the similar pattern little BS supports for smaller datasets. The confidence limits for species groupings using little BS with mean-bagging were underestimated for  $BCL > 50\%$ , and overestimated when the standard BS supports were less than 50% (Figure. 2.3c, orange dots). These results suggested that the little BS with mean-bagging fails to accurately estimate the BCL for phylogenies estimated from large or smaller dataset.

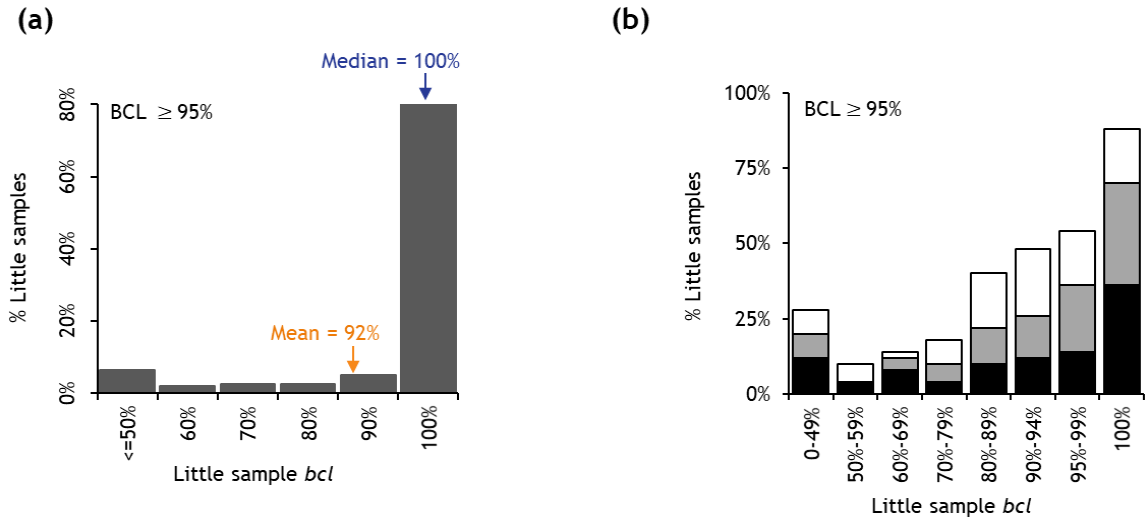


Figure 2.4. Accuracy of Little Bootstraps for small datasets. (a) The relationship of standard BS ( $BCL$ ) and little BS ( $\widehat{BCL}$ ) with mean-bagging (golden circles) and median-bagging (purple circles) for datasets smaller than 10,000 sites ( $l = L^{0.9}$ ). The gray line shows the 1:1 relationship with the standard BS. The little BS offered time savings up to 37% and memory savings up to 42% in these small data analyses. The linear regression slope is 0.97 ( $R^2 = 0.93$ ) for median-bagging and 0.89 ( $R^2 = 0.89$ ) for the mean-bagging. However, a second-order polynomial fits the mean-bagging results better ( $R^2 = 0.93$ ). (b) The distribution of little sample  $bcls$  for species groups in smaller datasets for which standard BS  $BCL \geq 95\%$  (black bars = 9,359 sites, gray bars = 7,002 sites, and white bars = 4,070 sites).

In little BS analyses, we estimate the bootstrap confidence limits ( $\widehat{BCL}$ ) by ensembling the subsample  $bcl$  values. Therefore, an investigation into the cause of this underestimation or overestimation revealed that the distribution of subsample  $bcls$  for these species groups was skewed and that the mean was not the accurate measure of central tendency (Figure. 2.4a). This prompted us to consider median-bagging because the median is more resilient to outliers, and the little BS with median is expected to have the same statistical properties as those established for mean-bagging<sup>67,79</sup>. However, median bagging has not been previously applied with the bag of little BS in any application. This unique application of

median-bagging with little BS analyses made this approach suitable for such a scenario when ensembling skewed  $bcl$  distribution.

The use of median-bagging eliminated 31 of the false negatives (Figure. 2.3a), with the remaining species group receiving  $\widehat{BCL} = 90\%$  (Figure. 2.3a). The average  $\widehat{BCL}$  at every branch length cutoff value was greater than 95% for median-bagging, but not for mean-bagging (Figure. 2.3b). We confirmed the improvement offered by median-bagging for a greater range of  $BCL$  values by analyzing three gene-specific sequence alignments ( $4,000 < L < 10,000$ ; 446 species). Median-bagging performed much better than mean-bagging for these short alignments (Figure. 2.3c) because the distribution of  $bcls$  was skewed and contained many outliers for each dataset (Figure. 2.4b).

### ***2.3.5 Phylogenomic subsampling approaches without upsampling.***

To assess the statistical advantages of upsampling, we compared the performance of little BS with and without upsampling. The subsampling was performed to generate ( $\widehat{BCL}$ ) values by a little BS procedure in which upsampling was replaced by the standard BS resampling such that the replicate datasets contained only  $l$  sites rather than  $L$  sites. We refer to this as the Phylogenomic Subsampling with Resampling (PSR) approach, in which one may use either mean- or median-bagging. We also generated ( $\widehat{BCL}$ s) without any resampling or upsampling (i.e.,  $r = 0$ ) such that the ML phylogenies were inferred from  $s$  subsample datasets containing  $l$  sites each. We call it the Phylogenomic Subsampling (PS) approach. We compared the true positive rates ( $\widehat{BCL} \geq 95\%$ ) of little BS, PSR, and PS

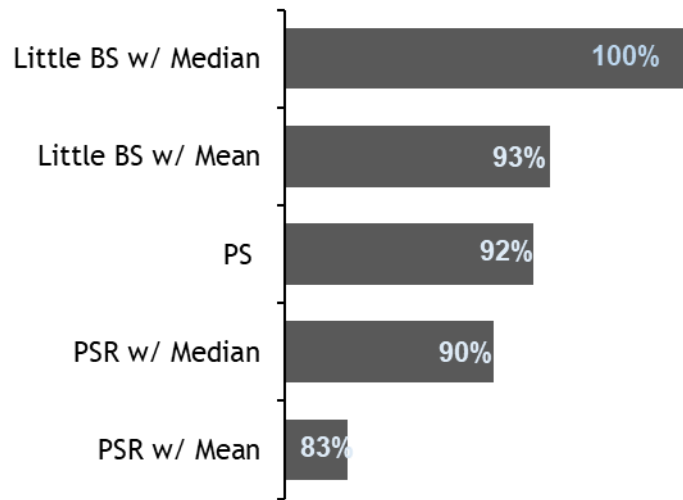


Figure 2.5. Limitations of subsampling approaches. The true positive rates (TPR) for little BS with mean- and median-bagging compared to other phylogenomic subsampling approaches (PS and PSR with Mean and with Median) in which upsampling was not applied

approaches for the computer-simulated 446x134,131 dataset ( $g = 0.7$ ) For all analyses, 100 replicate phylogenies were generated by using  $s = 10$  and  $r = 10$  for little BS and PSR, and (92%) compared to both little BS with a mean (93%) and median bagging (100%). The performance of PS could not be improved with resampling. We found 90% TPR for PSR with median bagging while the PSR with mean could recover only 83% of true species  $s = 100$  for the PS approach. The TPR for the phylogenetic subsampling is very low for groupings with high statistical support (Figure 2.5). The performance for PS was expected because subsamples without any upsampling contained reduced evolutionary information (e.g., number of substitutions), and statistical properties for smaller subsamples were not the same compared to the full MSA.

### 2.3.6 Automatic parameter tuning and the Precision of BCL estimates

We have developed a simple, automated protocol to determine the three key parameters, the size of the subsample ( $g$ ), the number of subsamples ( $s$ ), and the number of replicates ( $r$ ) for little BS analysis ( $g$ ,  $s$ , and  $r$ ). It starts with user-provided (or default) initial values and increases  $r$  and  $s$  iteratively to generate a stable average  $\widehat{BCL}$  for the whole phylogeny. This step is followed by increasing the size of the little subsamples ( $g$ ) and re-optimizing  $r$  and  $s$ . This procedure is continued until the average  $\widehat{BCL}$  over the whole phylogeny and the number of species groups receiving  $\widehat{BCL} \geq 95\%$  are maximized.

The procedure starts with  $g = 0.7$  if the sequence alignment contains  $\geq 100,000$  unique site configurations (such that  $l < 50,000$ ), otherwise, we set  $g = 0.8$ . Investigators may set any starting or fixed value of  $g$ . In step 1, we conduct little BS with  $s = 3$  and  $r = 3$  to generate initial  $\widehat{BCL}$  for all the nodes in the given phylogeny (if provided) or from a majority rule bootstrap consensus tree. From these estimates, we estimate average  $\widehat{BCL}$  ( $A_v$ ) and the fraction of inferred tree partitions with  $\widehat{BCL} \geq 95\%$  ( $N_v$ ). Through an iterative process, we stabilize and maximize both  $A_v$  and  $N_v$ , as follows. In step 2, we add one little BS replicates to each subsample (i.e.,  $r$  increases by 1) and compute  $A_v$ . Until the difference in successive  $A_v$  values is less than 0.1% (or a user-specified threshold,  $\delta_r$ ), we repeat steps 2 and 3 by increasing  $r$ . In step 4, we increase  $s$  by one and generate  $r$  additional replicate datasets and phylogenies for the new little subsample. If the difference between  $A_v$  for the current ( $s$ ) and previous ( $s-1$ ) set of subsamples is greater than 1 (or user-specified  $\delta_s$ ), then we repeat step 4. In step 5, we check and see if  $N_v$  is less than 100% or the standard error ( $SE$ ) of estimated  $\widehat{BCL} \geq 95\%$  is too high ( $>5\%$ ). If so, we increase the little subsample size by  $l$  and restart the analysis from step 2. In step 6, we go to step 4

if the user-specified precision (SE) has not been achieved. Its application to 446×134,131 dataset suggested using  $g = 0.8$ ,  $s = 4$ , and  $r = 6$ , which confirmed all correct species groups with  $\widehat{BCL} \geq 95\%$  (Average  $\widehat{BCL} = 100\%$ ). We used this automated system to analyze empirical sequence alignments and discussed its usefulness in the next section (Table 2.1).

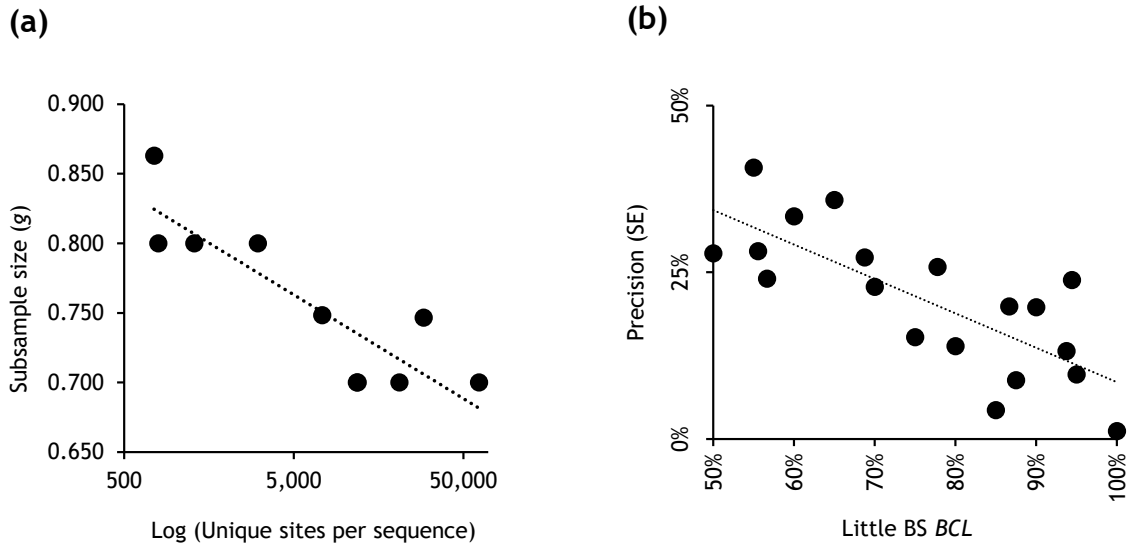


Figure 2.6. Subsample size and Precision. (a) The relationship of subsample size ( $g$ ) and the number of unique site configurations per sequence ( $C$ ) in empirical datasets. The log-linear regression slope is 0.032 ( $R^2 = 0.79$ ). (b) The relationship between the little BS  $\widehat{BCL}$ s and their precision (standard errors, SEs) for the selected little BS parameters (Table 2.1). The linear regression slope is -0.52 ( $R^2 = 0.59$ ).

### 2.3.7 The little bootstraps analysis pipeline.

We developed an R<sup>80</sup> pipeline to conduct little bootstraps analysis by using IQTREE. In this case, we used the Biostrings<sup>81</sup> package to generate little datasets of the specified lengths ( $l$ ) and then bootstrap replicate datasets in which  $L$  sites were resampled with replacement from  $l$  sites. The resulting datasets were used to obtain ML phylogenies that were summarized by using the function *plotBS* from the phangorn<sup>82</sup> library that produced the *bcl* for each of the phylogenetic groups in the standard bootstrap phylogeny. Mean and

median-bagging estimates were obtained from sample-wise *bcls* from *s* little samples using a customized function in *R*. We also developed a customized *R* function for estimating SEs of  $\widehat{BCLs}$ . We applied the automated protocol using a customized *R* function for performing little BS analyses which takes the full MSA and the phylogenetic tree on which the BCL will be mapped. This function output a phylogenetic tree with little BS supports and their corresponding precision (SE) values. The function for automated little BS analyses is provided in the Appendix A, which will also be found and downloaded from GitHub (<https://github.com/ssharma2712/Little-Bootstraps>). We have also developed a CodeOcean capsule (API) which any user can use without installing any software<sup>33</sup>. This capsule will be found in the CodeOcean repository (DOI: 10.24433/CO.6432188.v1).

### ***2.3.8 Analysis of empirical datasets***

To assess the performance and general applicability of little BS, we analyzed ten empirical datasets consisting of DNA sequence alignments. These datasets consisted of sequences from eutherian mammals<sup>38</sup>, butterflies<sup>83</sup>, plants<sup>84,85</sup>, insects (A<sup>86</sup>, B<sup>87</sup> and C<sup>88</sup>), spider<sup>89,90</sup>, and birds<sup>91</sup>. The number of taxa ranged from 16 to 193, and the number of sites ranged from 61,794 to 5,267,461 (Table 2.1). We used the phylogenetic trees (ML trees) presented in the original studies as the reference trees for empirical datasets. The ground truth for little BS confidence limits was the standard BS confidence limits reported in those published articles. The average BS for these datasets varied from 87% to 100%. To infer the ML tree for each of the little BS replicate from these empirical datasets, we utilized substitution models similar to those employed for performing BS analyses in the published articles.

In little BS analyses, the subsample size ( $g$ ) ranged from 0.70 to 0.86, and the number of sites in subsamples varied from 67,401 to 9,128. The total number of little BS replicates in these analyses ranged from 20 to 135, which were selected using the automated protocol described in the previous section 2.3.6. The accuracy of little BS with median bagging was excellent for all empirical datasets analyzed. The true positive rate (TPR) at  $\widehat{BCL} \geq 95\%$  was greater than 95% for eight datasets and 90% for the other two (Table 2.1). TPR at  $\widehat{BCL} \geq 70\%$  was greater than 95% for all datasets. Phylogeny-wide average  $\widehat{BCL}$  was close to that from standard BS  $BCL$ , as the average difference was only 0.4% (Table 2.1).

The high TPR and  $\widehat{BCL}$  accuracies for larger datasets were achieved by analyzing little subsamples containing only a fraction of all sites (Table 2.1). Consequently, the memory required to analyze each little BS replicate was tens to hundreds of MBs rather than multiple GBs. The computation time was in minutes or a few hours, depending on the number of sequences (Table 2.1). For example, the little BS analysis of the mammalian dataset (Table 2.1) required 0.1 GB per replicate, on average, rather than 3.1 GB RAM (~29-fold memory savings) and 0.32 CPU hours rather than 9.8 CPU hours per bootstrap replicate (31-fold time efficiency). This translated into greater than 95% savings in both memory and time, which enabled us to run many little BS replicates on a standard multicore personal desktop equipped with modest memory (8 GB).

The little BS analysis needed smaller subsamples (smaller  $g$ ) for empirical datasets with larger numbers of unique site configurations per sequence (C/S; Table 2.1, and Figure 2.6a). Analysis of datasets with fewer than 100,000 unique site configurations required subsamples containing a much larger fraction of site configurations (13%-27%) than those with millions of site configurations (1.3%-3.3%). This means that the little subsamples

already contained sufficient information for robust phylogenetic inference, as their  $C/S$  ratio was very large (197 – 2,726) even though an order of magnitude smaller than the full dataset (1,584 – 265,403; Table 2.1) in little BS replicate datasets. Interestingly, however, almost all the unique site configurations for smaller datasets were included in at least one little subsample, but more than 74% of the unique site configurations were never included in any little subsamples for datasets with greater than 100,000 unique sites in our empirical data analysis (Table 2.1).

During the automatic determination of little BS parameters for these empirical datasets, we also estimated the standard error (SE) of  $\widehat{BCL}$  estimates by a procedure in which subsamples and replicate phylogenies are resampled. Notably, high precision for  $\widehat{BCL}$  was achieved even when using small  $s$  and  $r$  because  $\widehat{BCL}$  values were very high for most of the species groupings in large datasets. Given  $r$  bootstrap replicate-phylogenies for  $s$  samples, we employ a bootstrap procedure to generate SE of  $\widehat{BCL}$ . We use already computed phylogenies of  $r \times s$  little BS replicates and derive  $\widehat{BCL}$  for all the nodes from collections of phylogenies by resampling  $s$  samples with replacement and  $r$  replicates with replacement every time a subsample is selected. It is carried out 100 times, and the standard deviation of each tree partition's  $\widehat{BCL}$  is generated to estimate  $SE$ . This process is extremely fast because precomputed phylogenies are used. The estimated  $SE(\widehat{BCL})$  was inversely proportional to  $\widehat{BCL}$  (Figure 2.6b). These results indicate that the precision for  $\widehat{BCL}$  is very high for large  $BCL$  values, while the precision decrease for smaller bootstrap confidence limits.

Table 2.1. Summary of Little Bootstraps analysis

Species	Full Dataset			Power factor (g)	s × r	Little BS samples			
	No. of Sites (L)	No. of Seqs (S)	Unique Sites/Seq. (C/S)			Sites (l)	Uniq. Sites (c)	Uniq. Sites/Seq. (c/S)	Total Uniq. Sites (U)
Butterflies	5,267,461	61	61,684	0.700	4×10	50,714	1.3%	793	5%
Plants A	4,246,454	16	11,897	0.700	4×5	43,614	3.3%	389	9%
Insects A	3,011,544	174	11,758	0.700	6×8	34,289	1.6%	188	8%
Insects B	2,938,039	48	29,092	0.747	10×12	67,401	3.8%	1,103	25%
Insects C	1,719,036	193	7,346	0.748	5×7	46,331	3.2%	236	15%
Mammals	1,391,742	37	20,962	0.700	7×9	19,976	2.3%	485	13%
Spiders A	137,170	27	3,071	0.800	4×8	12,877	13.4%	411	39%
Plants B	135,243	30	795	0.800	7×9	12,732	14.2%	113	56%
Spiders B	89,212	34	1,296	0.800	9×15	9,128	14.4%	186	66%
Birds	61,794	39	750	0.863	6×10	13,633	26.7%	201	80%

Table 2.1. (continued)

Species	Full Dataset	Little BS Results			Little BS Resources			
		Avg. $\widehat{BCL}$	$\Delta BCL$	TPR ( $\geq 70\%$ )	TPR ( $\geq 95\%$ )	Time (Hours)	Memory (GB)	Total Time (Hours)
Butterflies		100%	0.0%	100%	100%	1.37	0.38	54.8
Plants A		100%	0.0%	100%	100%	0.08	0.01	1.6
Insects A		97%	-1.8%	98%	98%	3.80	0.74	182.0
Insects B		91%	4.5%	100%	94%	3.80	0.33	546.0
Insects C		97%	1.1%	96%	99%	5.80	1.14	548.0
Mammals		98%	0.0%	97%	100%	0.32	0.11	18.9
Spiders A		94%	0.0%	96%	90%	0.08	0.04	2.5
Plants B		99%	-1.0%	100%	96%	0.03	0.01	1.9
Spiders B		97%	0.0%	93%	90%	0.06	0.03	14.9
Birds		90%	-3.3%	97%	93%	0.11	0.04	17.4

Note (Table 2.1).  $s$  and  $r$  are the numbers of little samples and bootstrap replicates, respectively, which were selected along with the number of sites in little samples ( $l = L^g$ ) by the automatic pipeline.  $U$  is the number of unique site configurations that were used in all the little samples. Avg. ( $\widehat{BCL}$ ) is the average of ( $\widehat{BCL}$ ) produced by little BS for all species groupings in a phylogeny.  $\Delta BCL$  is the difference between average bootstrap supports produced by the standard and little BS approaches. The true positive rate (TPR) is the percentage of species groups statistically supported by standard BS ( $BCL$ ) at the given cutoff value, which were also supported in the by little BS analysis ( $\widehat{BCL}$ ) at that cutoff value. The time and memory estimates are for one little BS dataset. The total time is for the completion of all little bootstrap replicates in a single computing thread.

### ***2.3.9 Combining Little BS with other optimizations***

We also evaluated the performance of little BS when combined with the Ultrafast bootstrap<sup>64</sup> (UFB) that makes standard bootstrapping faster for a large number of sequences. Little BS + UFB required only 50 minutes (0.2 GB RAM) on a computer with 5 cores when using ten little samples ( $r = 1,000$ , default in IQTREE<sup>64</sup>). This was much faster and leaner than using only one of the optimizations: UFB itself required 7.1 GB of RAM and 4.5 hours, whereas little BS alone required 19.8 hours and 0.1 GB of RAM. Therefore, plugging-in the UFB optimization for generating sample-wise *bcl*'s further increases memory and time savings. In the future, we expect little BS to be used along with other efficient heuristics developed to speed up bootstrap calculations<sup>63,64</sup>, and one may use Transfer Bootstrap<sup>57</sup> when estimating confidence limits.

## **2.4 Conclusion**

With the rise in large genomic datasets assembled from burgeoning sequence databases, the computational demands of Felsenstein's traditional bootstrap approach have become a major bottleneck limiting robust and reproducible phylogenetic research. The little bootstraps approach helps remove this bottleneck and enables parallelization even with modest computational resources. Ultimately, computationally efficient approaches will promote greater scientific rigor for all involved in building the tree of life, which requires assessing the robustness of inferences to selecting biologically distinct subsets of data, choice of substitution models and strategies, and application of a myriad of ways for combining multigene datasets.

## CHAPTER 3

# TAMING THE SELECTION OF OPTIMAL SUBSTITUTION MODELS IN PHYLOGENOMICS BY SITE SUBSAMPLING AND UPSAMPLING

### 3.1 Introduction

Mathematical substitution models of evolutionary rates between molecular bases and among sites in a multiple sequence alignment (MSA) are among the most fundamental descriptions of molecular evolution<sup>92-96</sup>. These models have become invaluable in phylogenetic analyses to track pathogen origins<sup>97</sup> and spread<sup>98</sup>, reconstruct the evolutionary history of genes and species<sup>99</sup>, and determine the tempo and mode of evolution<sup>36</sup>. Thousands of research articles report selecting the optimal substitution model<sup>45,94,100</sup> using Bayesian and other information criteria<sup>94,101,102</sup> to compare the Maximum Likelihood (ML) fit of several nested and non-nested substitution models.

The computational needs of model selection analyses are growing exponentially with the acquisition and assembly of increasingly longer sequence alignments<sup>7,33</sup>. For example,

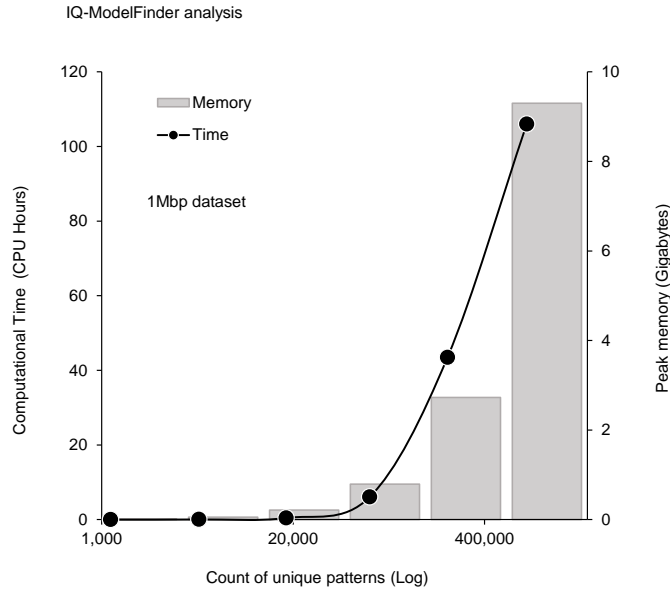


Figure 3.1. Computational resource requirements for model selection. The computational time (dots), and memory (bars) increases with the number of unique site patterns in the full MSA. We have analyzed six different subsets of different number of sites from the 1Mbp dataset and the model selection was performed using ModelFinder in IQTREE.

IQ-TREE's ModelFinder (IQ-MF) needed 9.3 GB of computer memory (RAM) and more than four days of computing (CPU time) to evaluate 286 models needed to select the optimal substitution model for concatenated DNA sequence alignment from 37 mammals ( $L = 1,391,742$  sites; hereafter 1Mbp dataset)<sup>38</sup>. This is because the computational costs are a function of the total count of unique site patterns ( $U$ ) in the whole alignment (Figure 3.1)<sup>33</sup>. Partitioning of 1 Mbp dataset by codon positions also produced very long alignments (each  $>460,000$  sites) that required more than 3.6 GB of RAM and 55 CPU hours of computing. In our survey of recently published articles using phylogenomics, we found that scientists routinely compare results from the analysis of both concatenated and partitioned datasets<sup>103–106</sup>. In these analyses, model selection for concatenated sequences

and long partitions requires many hours of computing and up to gigabytes of computer memory (Table 3.1).

## 3.2 Methods

### *3.2.1 The approach of upsampling sites from subsamples*

In the 1Mbp dataset, the number of distinct site patterns ( $U = 775,579$ ) is orders of magnitude larger than the number of free parameters in the most complex substitution model evaluated by IQ-MF. From this observation, we hypothesized that a fraction of site patterns ( $g$ ) is likely sufficient to infer the optimal model reliably, i.e.,  $g < 100\%$ . If true, this property will enable computational efficiency of the order  $1/g$  in both time and memory for ML analyses. To test this hypothesis, we empirically determined the smallest  $g$  that consistently produced the optimal substitution model identical to that selected using the full MSA by using IQ-MF. We constructed 100 phylogenomic subsamples of the 1Mbp dataset, each containing 1% of the unique site patterns ( $g = 1\%$ ) selected randomly without replacement from the 1Mbp alignment until the subsample contained  $g \times U$  different site patterns. Before applying IQ-MF to the phylogenomic subsample, we expanded the subsample by randomly upsampling its sites until the new alignment contained as many sites as the original MSA. Specifically, sites were selected randomly with replacement from the subsample until the total number of sites became the same as the full MSA<sup>33,67</sup>. Therefore, the subsample-upsample (SU) dataset contained 1,391,742 sites, equal to the number of sites in the 1Mbp dataset. We surmised that an SU dataset would have statistical power similar to the full MSA's in selecting the optimal model for large enough  $g$ .

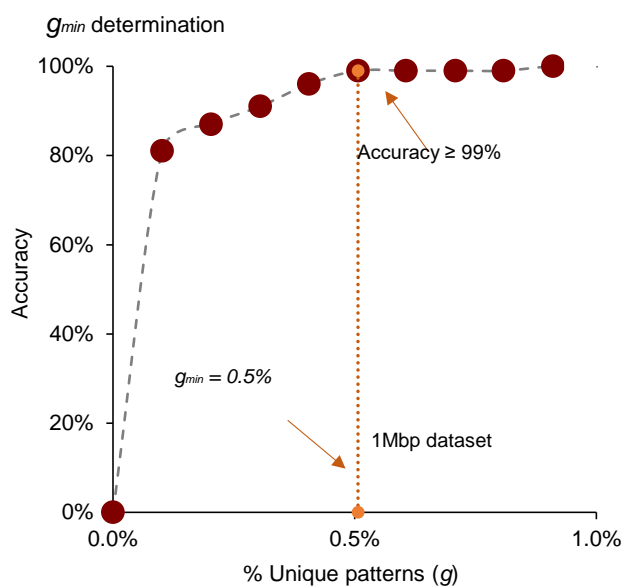


Figure 3.2. Determination of subsample size. The accuracy of model selection for subsampled–upsampled (SU) datasets for different fractions of unique site patterns ( $g$ ) sampled from the 1-Mbp dataset. The accuracy is the percentage of SU datasets for which the model selected was the same as that for the full MSA. The dotted line marks the point ( $g_{min}=0.5\%$ ) at which the accuracy becomes 99%.

We used ModelFinder in IQ-TREE (IQ-MF) with default options to select the optimal model in all analyses, skipping the advanced search option (*-mtree*) due to excessive computational time requirement as this option uses a separate initial tree for each of the models tested. We chose IQ-MF because it is now widely used in empirical data analysis. Other approaches, such as jModelTest<sup>100,107,108</sup>, were also tested and produced similar relative computational savings. Unfortunately, our attempts to use machine learning methods<sup>93</sup> for large datasets were not always fruitful because of the absence of machine learning methods for amino acid sequence alignments and the failure of all available online/offline tools to produce optimal models for large nucleotide sequence alignments. The proportion of SU datasets that produced the same optimal model as the full MSA is the accuracy of the SU approach for the given  $g$ . This accuracy was 100% for SU datasets

with  $g = 1\%$  when using IQ-MF for both SU and full MSA analyses. The SU dataset contains a small fraction of unique site patterns but has the same number of total sites as the full MSA. This means that every site pattern occurs many times in the SU dataset. Because the time and memory needs of the ML analysis are a function of the number of unique site patterns rather than the total sequence length, the analysis of the 1% SU dataset was 100 times faster and required proportionately less memory. SU datasets utilized only 94 megabytes of peak RAM and 1.4 CPU hours, on average.

### 3.2.2 Minimum subsample size for efficient model selection

Experimenting with phylogenomic subsamples of the 1Mbp dataset, we found that a high model selection accuracy could be achieved for even smaller subsamples (Figure 3.2). One

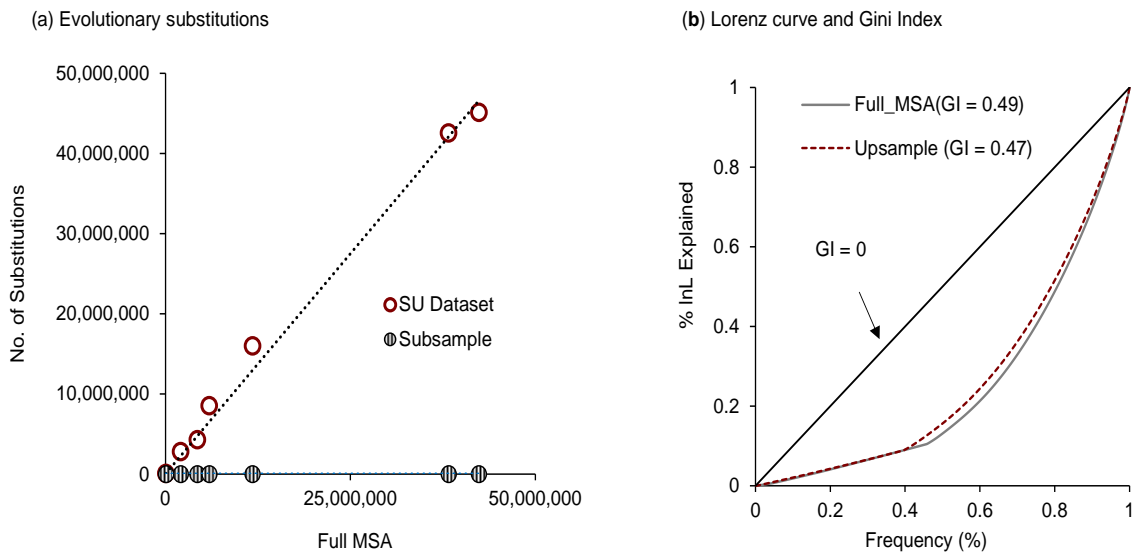


Figure 3.3. Advantages of upsampling. (a) Relationships between the total number of substitutions in the full MSAs and their SU datasets (dotted line; slope = 1.1) and subsample-only datasets (dots on the x-axis; slope =  $3 \times 10^{-6}$ ). (b) The Lorenz Curve for the relationship between the frequencies of site patterns and the proportion of the overall log-likelihood (lnL) contributed by those site patterns for 1 Mbp full MSA (lower curve) and the SU dataset with  $g = 0.5\%$  (higher curve). The Gini Index (GI) shown measures the inequality of information content distributed among site patterns.

hundred subsample-upsampled (SU) datasets were generated for each  $g$  (0.1% to 0.9%). The accuracy is the proportion of times SU datasets selected the same optimal model as the full MSA using IQ-MF. The  $g_{min}$  is the minimum  $g$  needed to achieve accuracy  $\geq 99\%$ . Accuracy was also calculated for subsamples in which no upsampling was performed. Accuracies  $\geq 99\%$  were observed for  $g \geq 0.5\%$ , i.e., the minimum fraction of unique site patterns ( $g_{min}$ ) needed to select an optimal model reliably for the 1Mbp dataset was 0.5%. This analysis required only 42 megabytes of peak RAM and was 130 times faster (0.81 versus 106 CPU hours). In contrast, the analysis of subsamples *without* upsampling had a low accuracy (12%) for  $g = 0.5\%$ . The performance of phylogenomic subsampling *without* upsampling could not be improved through any *post hoc* linear transformations of the information criteria (e.g., BIC) to account for the underrepresentation of the number of substitutions in the subsample because such linear adjustments may not change the relative rank of the tested models. Therefore, the upsampling procedure can overcome the analytical limitations of phylogenomic subsamples by achieving higher accuracy without increasing the computational burden. This is because the numbers of unique site patterns are almost the same in datasets with and without upsampling, but the total number of evolutionary substitutions in the SU datasets was similar to that in the full MSAs for the 1Mbp dataset (Figure 3.3a; ratio = 0.99). Also, the Lorenz curve and the Gini index for the SU dataset were similar to the full MSA (Figure 3.3b), showing that SU datasets recapture the pattern of information contents among the site patterns in the full MSA. This result suggests that the upsampling procedure ensures the inclusion of sufficient counts of different types of base substitutions to select the optimal model reliably. This was not the case for site subsamples without upsampling (ratio = 0.000003; Figure 3.3a), which results

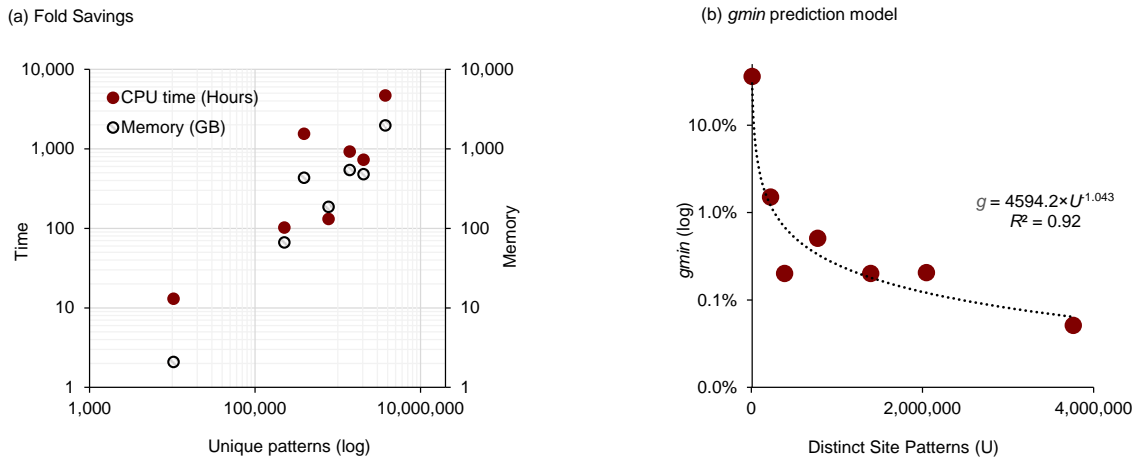


Figure 3.4. Computational resource savings. (a) Fold savings in computational time and memory were achieved in SU analysis of many large datasets for subsamples of size  $g_{min}$  at which the accuracy was at least 99% (Table 3.1). (b) The power relationship between the number of total unique patterns ( $U$ ) and the fraction of site patterns needed ( $g_{min}$ ) for  $\geq 99\%$  accuracy in model selection.

in a lower accuracy than SU datasets (12% versus 95%). To establish a general pattern between the  $g_{min}$  and the number of unique site patterns in the full MSA, we estimated  $g_{min}$  for many large empirical datasets. These datasets are gathered from diverse species (yeasts<sup>109</sup>, insects A<sup>86</sup>, and B<sup>87</sup>, butterflies<sup>83</sup>, birds<sup>91</sup>, mammals<sup>38</sup>) and the estimated  $g_{min}$  assess and measure the generality of the pattern observed for the 1Mbp dataset (Table 3.1). Among these empirical datasets, the yeast data only contained amino acid (AA) sequences from 23 yeast species. These MSAs contained 23 – 200 sequences and as many as 3.7 million distinct site patterns. We also estimated the  $g_{min}$  for a simulated DNA dataset containing 52 sequences and 10,348 unique site patterns. The optimal models for the DNA sequence alignments were found to be GTR+F+R, and TIM2 + F + R, while we found LG + F + R as the best fit model of substitution for the AA dataset. For large datasets, accuracy

Table 3.2 Model selection using full and Subsampled-upsampled (SU) datasets.

Data Summary					Full MSA		
Data	Type	Sequences	All Sites	Unique Patterns	Memory (GB)	Time (Hours)	Optimal Model
Butterflies	DNA	61	5,267,461	3,762,723	75.0	3140.7	GTR+F+R
Insects A	DNA	174	3,011,544	2,045,783	115.4	5000.3	GTR+F+R
Insects B	DNA	48	2,938,039	1,396,402	21.7	656.0	GTR+F+R
Mammals	DNA	39	1,391,742	775,579	9.3	106.0	GTR+F+R
Yeasts	AA	23	634,530	390,960	13.0	973.5	LG+F+R
Birds	DNA	200	394,684	226,490	14.6	258.0	GTR+F+R
Simulated	DNA	52	12,300	10,348	0.1	0.1	TIM2+F+R

Table 3.1. (continued)

Data Summary	Subsample-upsampled (SU) dataset				
Data	Patterns Used	$g_{min}$	Accuracy	Memory (GB)	Time (Hours)
Butterflies	1,918	0.05%	100%	0.04	0.67
Insects A	4,183	0.20%	99%	0.24	6.85
Insects B	2,796	0.20%	99%	0.04	0.71
Mammals	3,930	0.51%	99%	0.05	0.81
Yeasts	783	0.20%	100%	0.03	0.63
Birds	3,389	1.50%	100%	0.22	2.53
Simulated	3,709	36.0%	99%	0.06	0.01

Note (Table 3.1):  $g_{min}$  is the percentage of the site patterns sampled in each subamples. Accuracy is percentage of SU datasets that selected the same model as the full MSA

$\geq 99\%$  was achieved with  $g_{\min} < 1\%$ , saving  $>98\%$  of computational time and memory (Table 3.1). For many of these datasets,  $>1,000\times$  computational efficiency was achieved (Figure 3.4a). However, an accuracy  $\geq 99\%$  was achieved with  $g_{\min} = 36\%$  for the smaller dataset, which also offered modest time and memory savings for model selection. Generally,  $g_{\min}$  was smaller for longer sequence alignments (Table 3.1). Based on this trend, we built a non-linear power function between the number of unique site patterns ( $U$ ) in the MSA and  $g_{\min}$ . Mathematically, the function was  $g = 4594.2 \times U^{-1.043}$  (Figure 3.4b;  $R^2 = 0.92$ ), and used to predict the number of the percentage of minimum site patterns required for reliable model selection as the full MSA.

### 3.3 Adaptive tool for model selection

We implemented the subsample-upsample (SU) method into an adaptive tool (ModelTamer) that automatically determines the minimum  $g$  and selects the optimal model for use in empirical data analysis (Figure 2c). ModelTamer can be used with any method for selecting the optimal model, e.g., IQ-MF, jModelTest<sup>100</sup>, ModelTest-NG<sup>108</sup>, and MEGA-CC<sup>110</sup>. ModelTamer first calculates the initial fraction of site patterns ( $g_0$ ) to subsample, which is predicted using the relationship between  $g_{\min}$  and  $U$  shown in figure 3.4b. Then, it subsamples  $U \times g_0$  unique site patterns from the full MSA and generates a SU dataset by upsampling. In the next step, the SU dataset is analyzed using the chosen model selection method (IQ-MF here). The optimal model for the SU dataset can be based on BIC, AIC, AICc, likelihood ratio test, or other statistical criteria. In the next step, the number of patterns subsampled is increased to  $2 \times g_0$ . Optimal models produced by the analysis of  $g_0$ -SU and  $2 \times g_0$ -SU are then compared. If they do not match, then subsample

size is expanded ( $k \times g_0$ ,  $k = 3, 4, \dots$ ) and model selection applied. ModelTamer stops when two consecutive analyses produce the same substitution model (Figure 2c). We implemented ModelTamer coupled with IQ-MF in an R program, which also gives users the flexibility to further validate the selected model by increasing the number of site patterns in the SU dataset.

### 3.4 Evaluation of ModelTmaer Performance

#### 3.4.1 *ModelTamer* analysis.

We used the ModelTamer protocol (Figure 2c) implemented in R<sup>80</sup>. This package has a customized function, "SU\_MSA," to generate SU datasets using the "Biostrings" package<sup>81</sup>. IQ-MF<sup>94</sup> was applied to each SU dataset; one can couple other tools for model selection with ModelTamer. We expect the relative resource-saving to be similar when using other tools because the cost of ML analysis is a function of the unique site patterns used in all the software packages. The "aggregator\_model" function processes all the outputs and provides the optimal model and its parameters. It also outputs peak memory usage and the CPU time required by *ModelTamer*. We have also developed an automated function (*ModelTamer.R*) in R described in the main text, which takes the sequence alignment as input for model selection and produces the optimal substitution model and its parameters. R functions and the automated pipeline for model selection are available from <https://github.com/ssharma2712/ModelTamer>. A description file with an example dataset for implementing each function of *ModelTamer* is provided in the repository. The R codes for ModelTmaer is provided in the Appendix B.

### 3.4.2 *ModelTamer* performance for empirical datasets

We analyzed a total of eleven (11) empirical DNA and amino acid (AA) sequence alignments yeasts<sup>109</sup>, plants<sup>84</sup>, insects (A<sup>86</sup>, and B<sup>87</sup>), butterflies<sup>83</sup>, birds<sup>91</sup>, mammals (A<sup>38</sup>, and B<sup>111</sup>), Lassa viruses<sup>112</sup>, green plants<sup>113</sup>, and jawed vertebrates<sup>114</sup> (Tables 1-2). Six empirical datasets (butterflies, birds, Insects A and B, mammal A, and yeast) were analyzed to generate the  $g_{min}$  prediction model. The number of species ranged from 16 to 360, and the number of sites ranged from 3,186 to 5,267,461. The ground truth for these datasets were the best-fit substitution models selected by analyzing the full MSA using IQ-MF.

We applied *ModelTamer* with IQ-MF to many large and small empirical datasets and found it to produce the same model as the IQ-MF analysis of the full MSAs (Table 3.2). *ModelTamer* realized  $\geq 95\%$  savings in computational memory and time for large empirical datasets, as the estimated  $\hat{g}_{min}$  from 0.1% to 2.4% (Table 3.2, Figure 2d). These savings are expected to be smaller for datasets that contain a small number of unique site patterns because *ModelTamer* will need to use a larger fraction of site patterns in each subsample to include a few thousand unique site patterns necessary for a reliable substitution model selection (Table 3.2, Figure 2d). In all of these analyses, *ModelTamer* did not select the same model as the full MSA for one small empirical DNA dataset (Lassa Virus; Table 3.2). For this dataset, *ModelTamer* selected a model that was the second best in the IQ-MF analysis of the full MSA. Interestingly, the difference in BIC between the top two models was less than 10, which means that these two models will be considered statistically indistinguishable<sup>94</sup>. This suggests that for smaller datasets, *ModelTamer* may sometimes produce a model that is statistically equivalent to that produced by the analysis of the full MSA.

### 3.4.3 ModelTamer performance for simulated datasets

The optimal models selected by IQ-MF for large empirical datasets were usually the most complex models tested. Therefore, ModelTamer (IQ-MF) also selected complex models as the full MSA. We examined ModelTamer's performance when the actual underlying substitution process was simple, but the sequences were long. For this purpose, we performed new simulations to generate datasets with specific properties. DNA sequence alignments were simulated using simple models: Jukes-Cantor (1969) model (JC), Kimura (1981) 2-parameter model (K2P)<sup>115</sup>, and Hasegawa-Kishino-Yano (1985) model (HKY)<sup>116</sup>. The transition vs. transversion rate ratio for both K2P and HKY models was set to 2.00, and the base frequencies for the HKY model were set to be (A = 31%, C = 27%, G = 20%, and T = 22%) referring to HKY+F model in IQTREE<sup>46</sup>. Each simulated DNA sequence alignment contained 50 sequences with a sequence length of 100,000 (Table 3.2). Similarly, a set of AA datasets were simulated under an equal substitution probability (Poisson model) and more complex models: JTT<sup>117</sup> and WAG<sup>118</sup>. The AA sequence alignments simulated were 50,000 long and contained 20 sequences. For simulating each sequence alignment, a random tree was generated using an R function (`-rtree`) from the *ape* package where the branch length varied uniformly within the range from 0 to 0.2. The multiple sequence alignments were simulated using IQTREE (`--alisim` option). The ground truth for these simulated datasets was the substitution modes determined by analyzing full sequence alignment using IQ-MF<sup>46,94</sup>.

Table 3.3. ModelTamer analysis for empirical and simulated datasets

Data	Bases	Sequences	All Site Patterns	Used Patterns	$\hat{g}_{min}$
<b><u>Empirical Datasets</u></b>					
<b><u>Big Datasets</u></b>					
Butterflies	DNA	61	3,762,723	3,810	0.1%
Insects A	DNA	174	2,045,783	4,190	0.2%
Vertebrates	AA	58	1,547,914	1,806	0.1%
Insects B	DNA	48	1,396,402	4,217	0.3%
Mammals A	DNA	39	775,579	4,702	0.6%
Yeasts	AA	23	390,960	783	0.2%
Birds	DNA	200	226,490	4,504	2.0%
Plants	DNA	16	190,352	4,615	2.4%
<b><u>Small datasets</u></b>					
Green plants	AA	360	17,789	883	5.0%
Mammals B	DNA	274	4,303	2,710	63.0%
Lassa Virus	DNA	179	1,475	931	63.0%
<b><u>Simulated Datasets</u></b>					
<b><u>Big Datasets</u></b>					
This article #1	DNA	50	95,852	26,895	28.1%
This article # 2	DNA	50	95,820	31,508	32.9%
This article #3	DNA	50	92,600	21,916	23.7%
This article #4	AA	20	43,864	5,794	12.6%
This article #5	AA	20	44,895	5,840	13.3%
This article #6	AA	20	46,327	8,624	18.5%
<b><u>Small Datasets</u></b>					
Abadi et al. 1	DNA	44	13,110	2,612	19.9%
Abadi et al. 2	DNA	51	10,348	4,336	41.9%
Kalyaanamoorthy et al. 1	AA	100	9,806	994	10.0%
Kalyaanamoorthy et al. 2	AA	100	9,781	995	10.0%
Kalyaanamoorthy et. al. 3	AA	100	9,775	993	10.0%
Abadi et al. 3	DNA	52	7,442	4,524	60.8%

Table 3.2. (continued)

Data	Optimal Model (MT)	Memory (GB)	Time (Hours)	Memory Saving	Time Saving
<b><u>Empirical Datasets</u></b>					
<b><u>Big Datasets</u></b>					
Butterflies	GTR+F+R	0.08	1.00	99.9%	99.97%
Insects A	GTR+F+R	0.24	7.20	99.8%	99.9%
Vertebrates	JTT+F+R	0.16	3.50	99.9%	99.9%
Insects B	GTR+F+R	0.07	1.95	99.7%	99.7%
Mammals A	GTR+F+R	0.06	0.63	99.4%	99.4%
Yeasts	LG+F+R	0.03	1.00	99.8%	99.9%
Birds	GTR+F+R	0.29	3.10	98.0%	98.8%
Plants	GTR+F+R	0.02	0.13	97.7%	99.3%
<b><u>Small datasets</u></b>					
Green plants	JTT+F+R	0.51	10.10	94.9%	94.9%
Mammals B	GTR+F+R	0.24	1.95	36.9%	5.3%
Lassa Virus	GTR+F+R**	0.05	0.03	36.5%	91.2%
<b><u>Simulated Datasets</u></b>					
<b><u>Big Datasets</u></b>					
This article #1	JC	0.44	0.41	71.9%	47.2%
This article # 2	K2P	0.51	0.75	67.2%	-.04%
This article #3	HKY	0.36	1.15	76.3%	33.2%
This article #4	Poisson	0.18	0.38	86.8%	92.2%
This article #5	WAG	0.18	0.39	87.0%	86.5%
This article #6	JTT	0.26	1.22	81.6%	78.1%
<b><u>Small Datasets</u></b>					
Abadi et al. 1	TPM2u+F+R	0.08	0.02	59.4%	86.4%
Abadi et al. 2	TIM2+F+R	0.05	0.06	70.8%	27.3%
Kalyaanamoorthy et al. 1	LG+R	0.16	1.96	86.5%	93.6%
Kalyaanamoorthy et al. 2	LG+R	0.16	1.98	89.7%	96.0%
Kalyaanamoorthy et. al. 3	LG+R	0.16	2.00	89.7%	93.4%
Abadi et al. 3	HKY+F+R	0.08	0.02	39.2%	88.0%

Both IQ-MF on the full MSA and ModelTamer(IQ-MF) could select the one parameter JC model when the dataset was simulated under the simplest JC model (Table 3.2). ModelTamer analysis of sequence alignments produced by computer simulations under models with additional parameters, such as Kimura's 2-parameter (K2P)<sup>115</sup> and Hasegawa-Kishino-Yano (HKY)<sup>116</sup> models, also produced correct models (Table 3.2). In these analyses, ModelTamer frequently offered memory and time savings (Table 3.2).

ModelTamer performance was also evaluated for an AA sequence alignment when the instantaneous substitution rates between amino acid residues were the same (Poisson model). ModelFinder produced the correct model (Table 3.2). We also tested ModelTamer's ability to distinguish among equally complex amino acid substitution models by analyzing sequence alignments simulated using the JTT<sup>117</sup>, WAG<sup>118</sup>, and LG<sup>119</sup>. Both IQ-MF with full MSA and ModelTamer worked well (Table 3.2). In these analyses, ModelTamer saved more than 75% of computational time and memory. Based on these analyses, we expect the accuracy of ModelTamer in selecting the correct optimal model to be the same as that of the tool used in *ModelTamer* for evaluating the fit of different models (e.g., IQ-MF) because the ModelTamer system is intended to reduce the time and memory needs of model selection faithfully through site subsampling and upsampling. ModelTamer can be coupled with any method, including methods that consider data errors introduced during molecular sequencing and sequence alignments<sup>120</sup>.

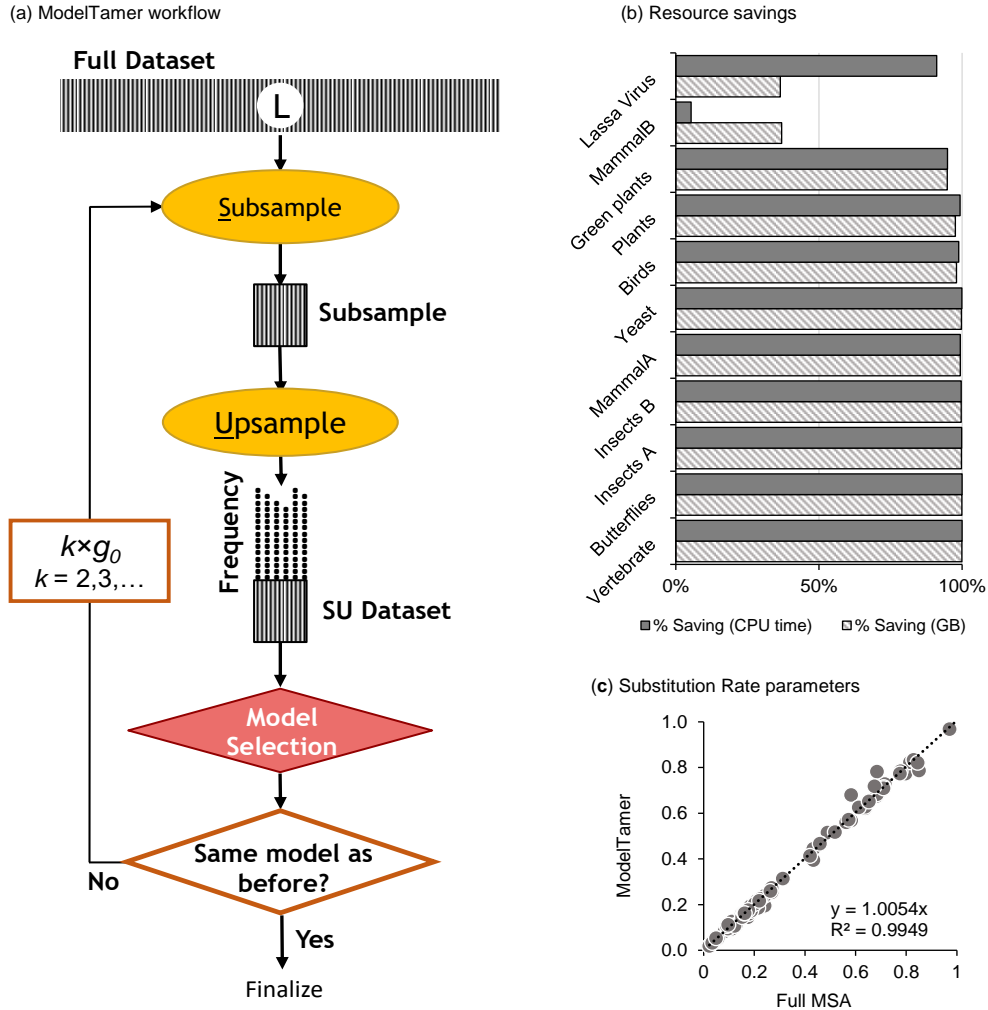


Figure 3.5. Performance of *ModelTamer*. (a) Flowchart of *ModelTamer* analysis. The shaded box represents the original sequence alignment containing sequences of length  $L$ , which has  $U$  unique site patterns. A subsample (small, shaded box) contains a specified fraction ( $g$ ) of unique site patterns from the full MSA. The initial value of  $g$  is predicted using the trend in panel 1D. A random sample of sites is drawn with replacement (multinomial sampling) from the subsample, which is then upsampled. The upsampled dataset has the same number of sites ( $L$ ) as the full MSA, but the number of unique site patterns remains the same as the subsample. Each site pattern is represented many times in the SU dataset, represented by many black dots above each position in the shaded box (SU dataset). The model selection is performed on this SU dataset. The optimal model found in this analysis can be validated by building SU datasets containing an increasing number of unique site patterns ( $k \times g$ ,  $k = 2, 3, \dots$ ) until two consecutive runs produce the same optimal model. (b) Computational savings in memory (GB) and time (CPU hours) achieved by *ModelTamer* for large and small empirical datasets (see Table 3.2). (c) Scatter plot showing the relationship of the estimated instantaneous substitution rate between bases from full MSA and SU analysis (slope  $\sim 1.0$ ) for empirical DNA datasets in Table 3.2.

In addition, a set of simulated DNA and AA sequence alignments were gathered from published articles which are <sup>93,94</sup> (Table 3.2). The number of sequences in these simulated datasets ranged from 44 to 100, and the number of unique site patterns varied between 7,442 and 13,110 (Table 3.2). The best-fit model for AA acid datasets was found to be LG+R when analyzing the full MSA, and HKY+F+R, TPM2u+F+R, and HKY+F+R were selected as the best-fit models for simulated small DNA datasets (Table 3.2). ModelTamer selected the same model as the IQ-MF for these small simulated datasets while offering  $\geq$  60% saving in computer memory and 27%-96% savings in computational time (Table 3.2).

#### ***3.4.4 The efficiency of ModelTamer for partitioned data***

In the above, we have presented the efficiency of ModelTamer for sequence alignments in which all the genes and genomic segments were concatenated for phylogenomic analysis. In addition to concatenated MSA, most systematic studies also designate collections of sites (partitions) based on biological, functional, and/or genomic considerations. ModelTamer can be applied to each partition separately to select the best-fit model efficiently. The adaptive nature of ModelTamer will automatically use all the site patterns for shorter partitions, taking no more time and memory than the standard tool used for model selection. For shorter sequence lengths, *ModelTamer* offered 5%-95% for computing time and 36%-95% of peak RAM (Table 3.2, Figure 3.5b). ModelTamer will offer high memory and time savings for longer partitions, such as those based on genome source (e.g., mitochondrion, chloroplast, and nucleus) <sup>121</sup>, specific codon positions <sup>122</sup>, functional annotations (e.g., coding and noncoding) <sup>121,123</sup>, and prior biological and evolutionary features <sup>103-105</sup>. Of course, one may eliminate the expense of model selection by simply using the most complex substitution model, but this approach has been debated

in the literature<sup>93,124,125</sup>. For amino acid sequence analysis, however, many models are equally complex and would require model selection for which *ModelTamer* is efficient and accurate. Furthermore, as shown below, *ModelTamer* greatly reduces the time and memory needed for estimating substitution rate matrix parameters for any model for long sequences.

### 3.4.5 Estimating model parameters using SU datasets

For a long amino acid sequence alignment from 58 vertebrate species (1,806,035 sites), IQ-MF required 4,604 CPU hours (6.4 CPU months) to finish the optimal model selection on a high-performance computer with 139 GB of RAM. For this dataset, *ModelTamer* analysis required less than 3 hours and less than 1 GB of memory ( $g_{min} = 0.1\%$ ), making optimal selection feasible. The time required was orders of magnitude less than that demanded by IQ-MF for even fitting a given substitution model and a fixed phylogeny to this dataset, as IQ-MF needed 69 GB of RAM and 130 CPU hours. Interestingly, estimates of the substitution rates and other model parameters (e.g., mean relative rate) produced by *ModelTamer* were very similar to those from the analysis of full MSA. *ModelTamer* estimates of the substitution rate matrix parameters showed a 1:1 relationship with those produced from the analysis of empirical DNA datasets (Figure 3.5c, slope  $> 0.99$ ;  $R^2 > 0.99$ ). *ModelTamer*'s estimates of site-wise substitution rates were also close to those from full MSA analysis (slope = 0.96-1.00;  $R^2 \geq 0.99$ ).

## 3.5 Conclusions

The power of upsampling of site subsamples and its desirable theoretical properties are already known for estimating confidence intervals<sup>33,67</sup>. Here, we have demonstrated that only a small representative fraction of unique site patterns contains sufficient information

to select the optimal substitution model and estimate its rate parameters effectively. We have also shown that the fraction of site patterns necessary can be determined automatically by a simple protocol (ModelTamer). These findings are likely to have implications for the general application of the SU approach. Ultimately, we expect *ModelTamer* to reduce the enormous computational demands of model selection that precede big data phylogeny inference for which many efficient tools exist<sup>33,46,47</sup>. Consequently, researchers with even commodity computers will be able to conduct big data analysis on their desktops, and those utilizing high-performance computing infrastructure will benefit by achieving greater calculation parallelization because of the small memory footprint of individual calculations in *ModelTamer*. These computational efficiencies will promote higher scientific rigor, broader participation, and environment-friendly computing in molecular evolutionary research<sup>126</sup>.

## CHAPTER 4

# MACHINE LEARNING DETECTS FRAGILE CLADES AND CAUSAL SEQUENCES IN PHYLOGENOMIC DATASETS

### 4.1 Introduction

The practice of assembling long sequence alignments, which include numerous genes and genomic segments, is now routine in molecular systematics<sup>7,31,33,127–129</sup>. Theoretically, leveraging hundreds to thousands of genomic segments --such as genes and ultraconserved elements-- should provide a robust basis for constructing reproducible and strongly supported organismal relationships<sup>7,126,128,130</sup>. However, molecular phylogenetic analyses of such extensive datasets have not consistently yielded high confidence or reproduced the relationships<sup>41,53,54,131–133</sup>. Notably, a single gene was responsible for the unstable placement of a fungi family (Ascoideacea in the CUG-Ser1 clade) within a phylogeny inferred from a dataset containing 1,233 genes<sup>41</sup>. Similarly, one exon caused phylogenetic instability in phylogenomic-scale datasets<sup>129</sup>. Such findings challenge the intuition that the

cumulative phylogenetic signals from numerous genes will neutralize the effects of a few disruptive gene-species combinations and produce correct and well-supported relationships.

Ideally, researchers would like to identify specific gene-species combinations in large phylogenomic data metrics--containing thousands of gene-species combinations--that may harbor potentially disruptive phylogenetic signals. This task is especially challenging for datasets meticulously curated to exclude non-orthologous sequences and other forms of contaminants, e.g., ref<sup>134</sup>. Indeed, searching for hidden gene-species combinations that impact the robustness of specific clades is akin to looking for a needle in the haystack, a challenge that we sought to address.

The existing approaches often require the evaluation of alternative phylogenies to find genes that may have undue influence on specific clades of interest<sup>41,53,54</sup>. But, these methods do not pinpoint individual gene-species combinations and need time-consuming reanalyses of the data. For instance, researchers estimate the difference in gene-wise support for alternative phylogenetic hypotheses identifying genes with outsized influence, followed by repeated phylogenomic analyses excluding these to establish their importance<sup>41,53,54</sup>. This process necessitates a priori selection of clades to investigate and knowledge of plausible alternative phylogenetic hypotheses to test. However, only a limited set of clades or hypotheses may be testable in this type of analysis due to a lack of prior knowledge or an excess of plausible combinations. Commonly utilized Maximum Likelihood (ML)<sup>41,54</sup> and Bayes Factor (BF)<sup>53</sup> analyses also impose constraints through substantial computational burden.

Researchers also use different subsets of genes and species to identify fragile clades in the phylogeny inferred from the entire dataset, requiring repetitive analyses of data subsets. For instance, subsamples containing varying numbers of genes are sometimes analyzed to assess the stability of the placement of certain species in the inferred phylogeny<sup>38</sup>. However, choosing the optimal subsample size and determining the number of subsamples to analyze can prove challenging<sup>52</sup>. Again, these types of analyses may not reveal individual gene-species combinations that may render clade inferences fragile. This limitation also applies to many other methods designed to identify problematic genes and species prior to phylogenetic analyses (e.g., ref<sup>41,53,54,135</sup>).

In this chapter, we present our findings from an exploration of a novel supervised machine-learning approach to address current limitations in detecting fragile clades in an inferred phylogeny and gene-species combinations contributing to this fragility. In this analysis, we used the Evolutionary Sparse Learning (ESL) framework, which builds clade-specific genetic models using supervised machine learning<sup>35</sup>. Using ESL, we developed clade models directly from the inferred phylogeny and phylogenomic alignment without the need for alternative phylogenies or pre-training. In this model building, ESL automatically compares many genetic models, each representing combinations of included genes and sites, to identify the model that best predicts the species composition of the clade of interest. The genes and sites in the clade-specific ESL model are those with the strongest association with the inferred clade. For these genes, we expect that individual species within the clade will exhibit phylogenetic signals concordant with the monophyly of the clades of interest. (One may analyze a given clade or all clades one by one.)

We introduce a novel gene-species concordance (GSC) metric, which is positive when the sequence of the specific gene in a species supports its inclusion in the inferred clade. Gene-species combinations with large negative GSCs strongly contradict the inclusion of the species within the inferred clade of interest. These could be due to data errors or biological reasons yet to be revealed. We introduce another metric, clade probability (CP), based on the GSC values for all the gene-species combinations in the ESL model. CP is the minimum of the classification probabilities of all the member species in the clade, with the member probability of a species calculated using the sum of GSC for that species over all the genes. CP would be low for clades for which some gene-species combinations have negative GSC scores, i.e., they harbor signals discordant with the inferred clade.

In the following, we first outline the model building for a monophyletic clade using the ESL approach and then introduce the rationale and formulae to calculate GSC and CP. The entire approach has been implemented in a software tool named DrPhylo for practical application. Using *DrPhylo*, we analyzed three phylogenomic datasets: fungi (comprising 86 species and 1,233 proteins)<sup>41,136</sup>, plants (102 species and 620 nuclear genes)<sup>37,41</sup>, and animals (37 species and 1,245 genes)<sup>132,137</sup>. These datasets were selected because of the presence of highly-supported fragile clades and the implications of influential genes identified through the analysis of alternative phylogenies in previous studies<sup>37,41,132,137</sup>. The use of these datasets enabled direct comparisons that act as a test of the new metrics' effectiveness and efficiency in detecting fragile clades and the disruptive gene-species combinations for the clades investigated. Because the calculation of GSCs and CPs does not require alternative hypotheses and repeated phylogenetic analysis, one can apply DrPhylo to all or any clade in an inferred phylogeny.

## 4.2 Estimating Gene-species Concordances and Clade Probabilities

We derive gene-species concordance (GSC) and clade probability (CP) from the ESL model reconstructed through supervised machine learning. Briefly, an ESL model is expressed as  $f(Y) = X\beta$ , where  $f(Y)$  is a logit link function of the category assigned to each species: +1 for species members of the clade of interest and -1 for all others in the inferred phylogeny<sup>35</sup>. Because class balance is important in supervised machine learning, we devised a phylogenetic-aware sampling approach to select balanced datasets containing equal numbers of species in +1 and -1 classes (see section ). In the ESL model,  $X$  is a matrix of one-hot encoded sequence alignment produced as previously described (see Figure 1 in ref.<sup>35</sup>).  $\beta$  is the vector of coefficients, which is estimated by the machine learning procedure, in which individual elements quantify the strength of association between the pattern of sequence evolution at individual sites.

Individual sites and genes (group of sites) are the model parameters in the ESL analysis. In fact, groups of sites can be collections of contiguous sites (e.g., genes, proteins, exons, and introns), non-contiguous sites (e.g., individual codon positions), and/or sites with functional annotation (e.g., coding and non-coding genomic segments)<sup>35</sup>. For the purpose of this article, we will henceforth consider genes as groups, as the empirical datasets analyzed here are partitioned by genes or proteins. It is worth noting that ESL can be used to analyze clades in a phylogeny inferred using any approach, e.g., concatenated super-matrix or partitioned alignment by Maximum Likelihood or Multi-species Coalescence (MSC) approach. One may also use it for any investigator-specified grouping of species. Also, while we consider species phylogenies throughout this article, tips of the tree may be any operational taxonomic unit.

The ESL model optimizes  $\beta$ 's by minimizing the logistic loss that penalizes the inclusion of individual sites and groups of sites to avoid overfitting<sup>35,138,139</sup>. A multitude of alternative quantitative genetic models, each featuring different combinations and quantities of genes and sites, are explored. The model that most accurately predicts the numerical labels (+1 and -1) for the species in the dataset. Two penalty parameters are used in ESL: one for penalizing the inclusion of sites into the genetic model ( $\lambda_1$ ), and the other for incorporating genes ( $\lambda_2$ ). We used a fixed value for these two sparsity parameters ( $\lambda_1=0.1$ ,  $\lambda_2=0.2$ ). In the final optimal model, the majority of genes and sites receive a  $\beta$  value of 0, leading to a sparse solution for inferring the genetic model. Hence, this type of learning is referred to as sparse learning<sup>35,139</sup>. This sparsity aligns with biological expectations, as only a select few sites and genes might carry the phylogenetic signals necessary to unify species within the analyzed clade. We employ these  $\beta$  values to compute GSC and CP, as described below.

#### ***4.2.1 Gene-Species Concordance (GSC) metric***

GSC measures the relative phylogenetic signal contributed by the sequence of a gene ( $g$ ) from species ( $s$ ) within the given clade ( $c$ ). It is the sum of the product of one-hot encoded bases of constituent sites of a gene  $g$  in species  $s$  with the regression coefficients in the ESL model for clade  $c$ . Mathematically,

$$GSC = \sum_{k=1}^K y_s \times \beta_k \times x_k,$$

where  $K$  is the number of bit columns representing sites belonging to gene  $g$ , and  $x_k$  and  $\beta_k$  are the  $k$ th bit-column and its estimated regression coefficient in a gene  $g$ .  $y_s$  is the numeric value (+1/-1) of the label for the species  $s$ .

A positive value of GSC indicates that the gene  $g$  supports the placement of species  $s$  in clade  $c$ . The magnitude of GSC quantifies the relative strength of support. A negative value of GSC means that the sequence of a gene  $g$  is discordant with that species' inclusion in the clade, i.e., it is missing the shared-derived base found in other members of the species. Gene sequences with  $GSC \approx 0$  are ambivalent about the inclusion of the species  $s$  in clade  $c$ . The GSC metric is analogous to the SHAP value<sup>140</sup>, commonly used to quantify a feature's contribution to the predictive ability of the machine learning model. However, GSC does not require re-estimation of the regression coefficients by excluding/including parameters. Notably, GSC is distinct from group sparsity score (GSS), which is always positive, not species-specific, and measures the global importance of a gene in the clade model<sup>35</sup>. GSS is estimated as follows:

$$GSS = \sum_{k=1}^K |\beta_k|. \quad [3]$$

Here  $K$  is the number of bit columns in the one-hot encoded matrix for gene  $g$ .

#### 4.2.2 Clade Probability (CP)

The probability of occurrence of a clade  $c$  consisting of  $m$  species is defined as the minimum of normalized classification probabilities ( $MP_i$ ) of its member species:

$$CP = \min \{MP_i\}_{i=1}^m.$$

Here  $MP_i$  is the normalized classification probability of species  $i$  of the clade  $c$ . It is estimated using a sigmoid function of the sequence prediction score<sup>35</sup>, the sum of GSCs of all the genes for species  $i$ .

$$MP_i = 1/\{1 + \exp(-\sum_{g=1}^G GSC_g^i)\} = 1/\{1 + \exp(-SPS_i)\}.$$

Here,  $GSC_g^i$  is the gene-species concordance score from the  $i^{th}$  species and  $g^{th}$  gene, and  $SPS_i$  is the sequence prediction score from the  $i^{th}$  species. Next,  $MP_i$  is scaled to a normalized range of 0 to 1. The normalization was performed to make the clade probability (CP) for the monophyletic clade analogous to the statistical confidence limits (0 to 1) typically estimated in regular phylogenetic analysis. Henceforth, the term  $MP_i$  will be referred to as the normalized inclusion probability of a species inside the monophyletic clade.

### **4.3 DrPhylo Implementation**

We have programmed all the above calculations in an analysis pipeline using the myESL software, which is packaged in a distribution called DrPhylo. DrPhylo can accept a phylogeny in newick format and a collection of FastA files that contain sequence alignments for individual genes. DrPhylo processes the input data and hypotheses, constructs ESL model(s) for specific clades, and computes all the pertinent metrics, including GSC and CP. The software outputs comma-separated files and user-friendly

graphical displays, simplifying the process of identifying fragile clades and their associated gene-species combinations (see Figure 4.1).

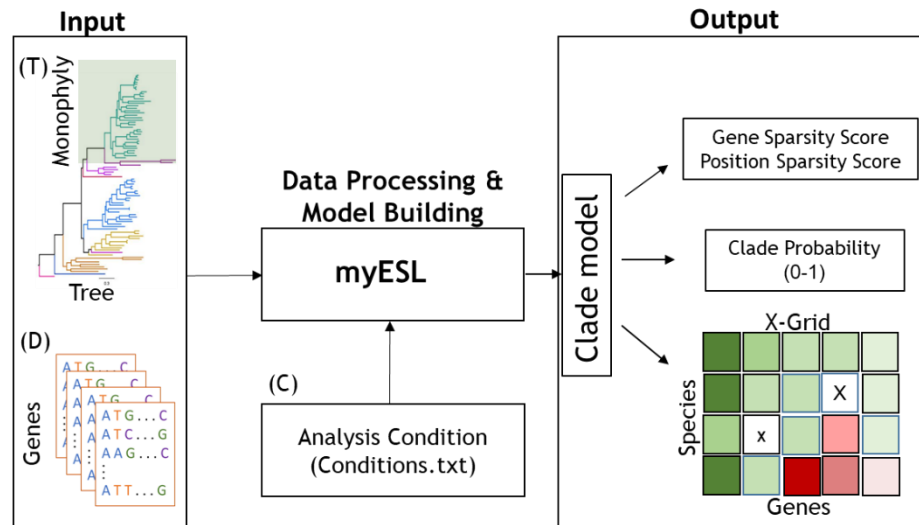


Figure 4.1. A schematic outlining of DrPhylo analysis pipeline. DrPhylo takes a rooted phylogeny in newick format (T) and a collection of gene-wise sequence alignments in the FastA format (D). These input data are processed by DrPhylo using data analysis options specified in the conditions.txt file (C) in the myESL software (Sanderford et al. 2023) that has been enhanced to implement phylogeny-aware data balancing and GSC and CP calculations presented in this article. Users can select the sparsity parameters (or calculate them automatically) and provide the response (hypothesis) using a text file (e.g., response.txt). DrPhylo produces multiple output files. It outputs the clade model, which presents GSS for genes and PSS for sites, as well as MP for species and CP for the clade of interest. It also outputs the X-grid representing the gene-species concordance for all gene-species combinations in the clade model. Cells with a positive GSC value (Green) indicate concordance, while a negative value (Red) represents the discordance of a gene for placing the sequence inside the clade. It also indicates the missing gene for a specific taxon using a cross mark inside the white grid.

## 4.4 Analysis of empirical datasets

### 4.4.1 Analysis of empirical fungi dataset

We employed *DrPhylo* to analyze the phylogeny of 86 fungi species derived from an alignment of 1,233 nuclear proteins (609,899 amino acid positions). This dataset was used to evaluate the robustness of the inference of the clade marked as A+B in the phylogeny depicted in figure 4.2a<sup>41</sup>. *DrPhylo* required less than a minute and 2.2 GB of peak memory on a standard desktop computer to infer the genetic model for clade A+B. Interestingly, Clade A+B received a low CP (0.18), which did not agree with the 100% bootstrap support for this clade in the ML analysis of the concatenated supermatrix<sup>41,136</sup>. However, the removal of a single gene (BUSCOFEOG7W9S51; 7W9S51 hereafter), out of 1,233, broke the monophyly of clade A+B, which is more in line with the low CP produced by the ESL analysis. Multispecies coalescence (MSC) analysis<sup>68,141</sup> also produced moderately low statistical support for the monophyly of A+B<sup>41</sup>. Therefore, the clade model inferred by ESL analysis produced the same result as others but without needing to compare any alternative phylogenies<sup>41</sup> or individual gene phylogenies.

Next, we examined the distribution of GSC scores for all gene-species combinations for clade A+B (Figure 2b). The distribution is centered around 0, with most values falling between -0.1 to 0.1. Notably, we found a set of GSC values from two genes that were outliers for the GSC distribution (red and green insets, Figure 4.2b). The majority of positive outliers (green inset) were from the 7W9S51 in this distribution. This suggests that the gene 7W9S51 has the outlier influence for clade A+B. Interestingly, this gene was also found in the ML analysis as the most influential gene by comparing two alternative

hypotheses<sup>41</sup>. Therefore, our approach successfully identified the most influential gene supporting the monophyly of A+B without comparing alternative hypotheses. In addition, the GSC distribution also revealed another gene, BUSCOfEOG7TN012 (hereafter 7TN012), associated with outlier GSCs. The gene-species combinations from this gene support the placement of all species inside the clade A+B, except *A. rubescenc*. The extremely high negative GSC (red inset) for *A. rubescenc*-7TN012 strongly contradicted *A. rubescenc* placement inside the monophyletic clade. This observation now explains why the exclusion of 7W9S51 from the data matrix resulted in an alternative phylogeny in which *A. rubescence* moved out of A+B (Shen et al. 2017). Interestingly, *A. rubescence* received the lowest MP in the clade model. In summary, our clade model and new metrics GSC and CP could successfully identify influential gene-species combinations (7W9S51 in *A. rubescence* ), rogue species (*A. rubescence*), and fragile clade (A+B).

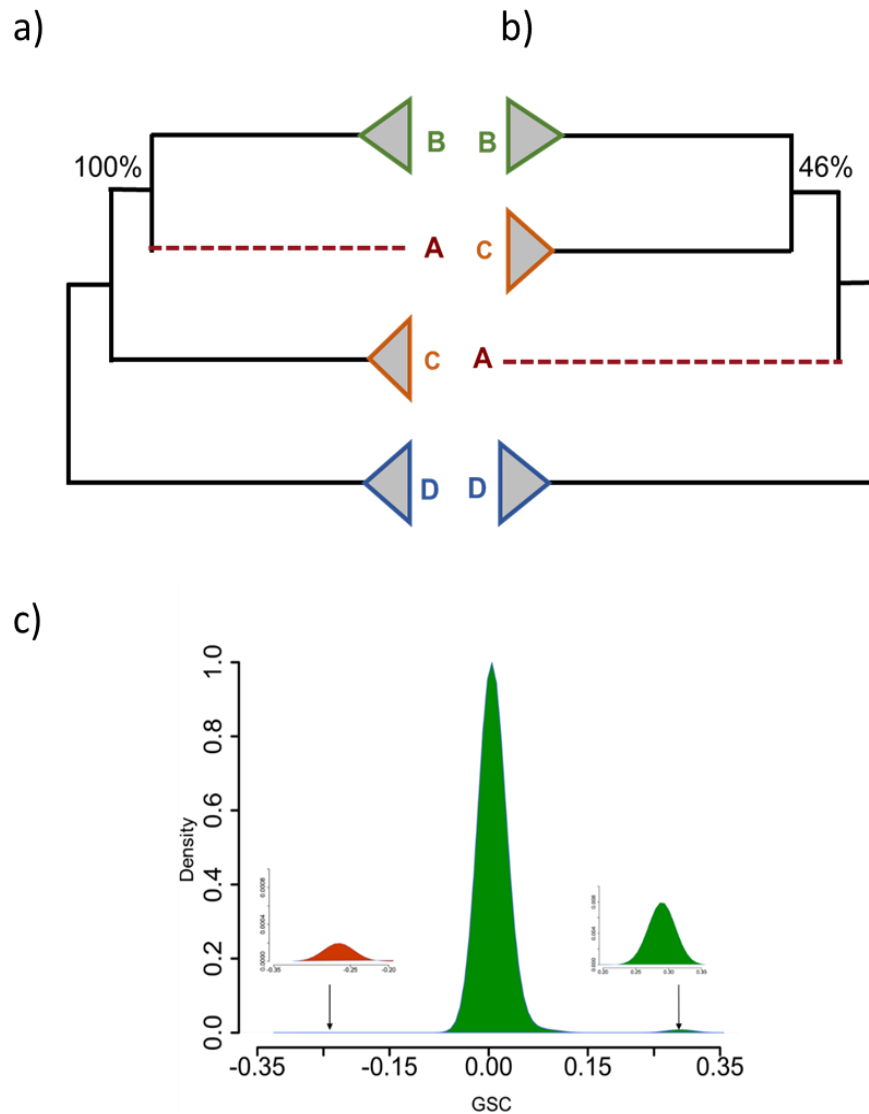


Figure 4.2. Clade model for fungi phylogeny. a) The Maximum Likelihood (ML) phylogeny of Fungi species inferred from a concatenated supermatrix of 1233 nuclear proteins (609,899 AA sites). The species **A** represents *Ascoidea rubescence* from the family of *Ascoideaceae* (CUG-Ser1 clade, 1), and is sister to a bigger clade **B** that includes yeast species from the families *Phaffomycetaceae* (4), *Saccharomycodaceae* (3), and *Saccharomycetaceae* (36). The clade **C** included yeast species from *Pichiaceae* (11), CUG-Ser2 clade (22), and **D** contained species from other fungi clades. b) The ML tree from the concatenated supermatrix after the removal of one influential gene. It inferred that **A** is the sister to a bigger clade that included both **B** and **C**. For ESL model building, we assigned +1 for all 44 taxa from clade **A+B**, and the rest of the taxa were assigned -1. The clade model was built in myESL software by analyzing multiple sequence alignments of 1233 nuclear genes from 86 Fungi species. We used the option "--method logistic" to perform sparse logistic group lasso regression by employing sparsity constraints on both loci and sites. c) The distribution of gene-species concordance (GSC). The green and red humps in the plot represent outlier concordant or discordant gene-species combinations for the clade **A+B**.

#### ***4.4.2 The model grid (M-grid)***

To facilitate a global view of the clade model, clade support, and gene-species combinations, DrPhylo outputs a graphical representation of the ESL model shown in Figure 3b. The model is presented in a grid format, where species are in the rows and genes are in the columns. We refer to this as the model grid, where cells are colored based on the GSCs for gene-species combinations. The green color is used for concordance, and the red color for discordance, with the color richness depicting the intensity of concordance or discordance. We sort rows to expose species receiving the lowest MP by placing them on the top and columns such that the genes with the highest sum of the positive GSC across species are placed first on the left. Cells with a cross-mark indicate genes with missing data. This M-grid showed that only a few GSCs out of 10,6083 gene-species combinations ( $= 1,233 \times 86$ ) are responsible for the fragility of the A+B clade. The grid also immediately reveals that gene 7W9S51 provides the strongest phylogenetic signal for A+B, while 7TN012 carries the strongest conflicting signal in *A. rubescens* against A+B. A few other genes harbor faint discordance signals, which are likely offset by those with concordance with A+B (Figure 4.3).

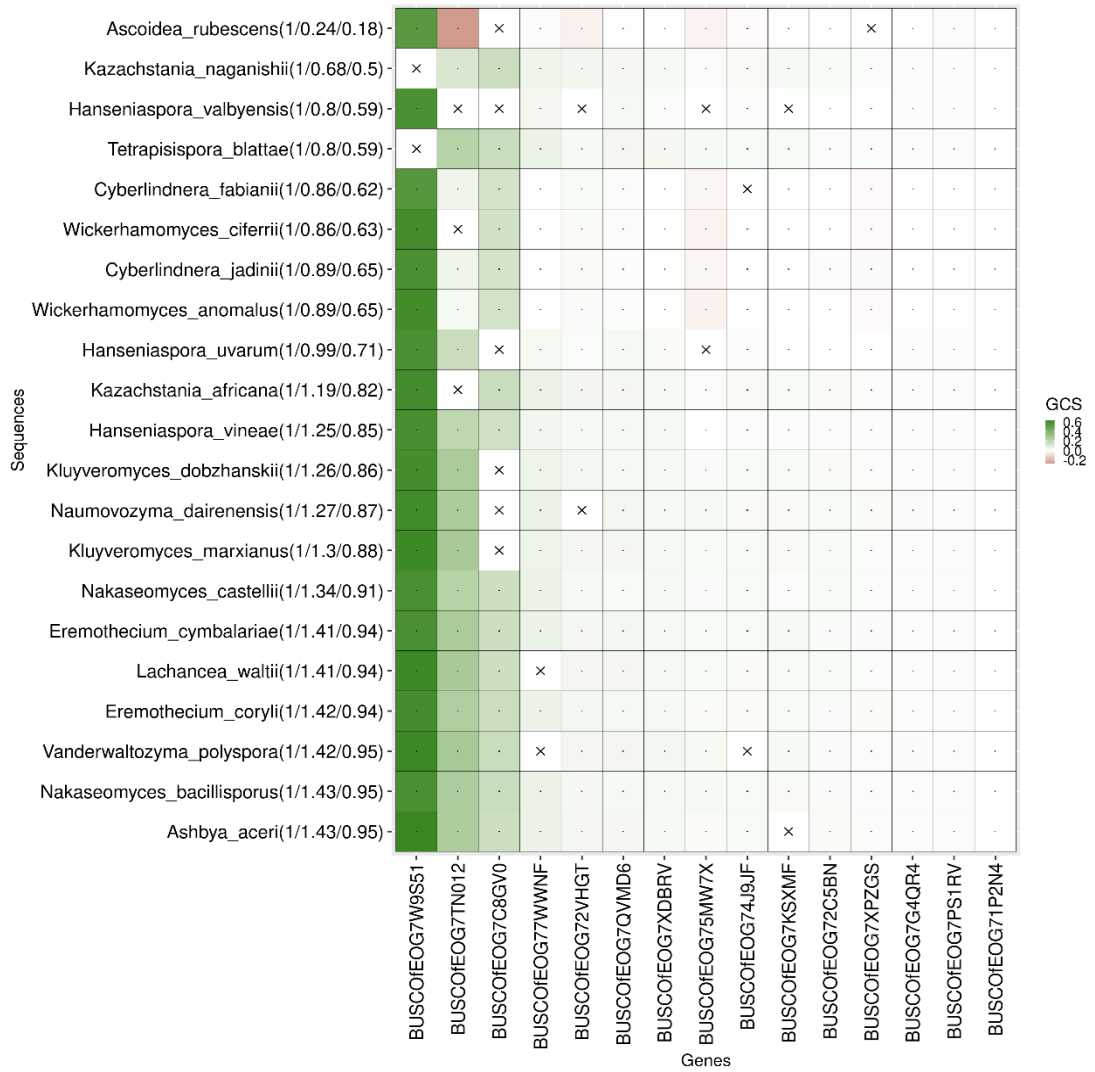


Figure 4.3. The Model-grid (M-grid) for clade A+B. The columns in the M-grid for clade A+B represents top genes selected in the clade model, and rows are for twenty fungi species from the clade with MP values. The top row represent the species with lowest MP, attributed by *A. rubescens* and the left most gene 7W9S51 showed the highest average concordance with the clade of interest.

#### ***4.4.3 An analysis of an expanded fungi dataset***

Shen et al. 2018 collected data from three additional species for taxon A, one more from the family Ascoideacea and two from the genus *Sacchromycopsi*, increasing the number of species from 1 to 4<sup>36</sup>. The number of species in clade B (42 to 150) as well as other clades. In addition, the number of genes was also increased to 1,289. The M-model produced by DrPhylo for the A+B clade is shown in Figure 4.4a. In the selected model, CP decreased from 0.18 to 0.0. This is because only 40% of the gene-specie combinations A+B clade, i.e., their GSC was greater than 0, which is similar to 39% quartets supporting clade A+B in the MSC analysis<sup>36</sup>. Interestingly, MSC gives a posterior probability of 100% to A+B, despite such low quartet support.

The reason for the fragility of clade A+B in the expanded dataset is that gene EOG09343FGH in two *Sacchromycopsis* species harbors high contradictory phylogenetic signal (Figure 4a). This result is confirmed by inspecting the gene tree for EOG09343FGH, which contains a very long internal branch (6.2 substitutions per site). Interestingly, two members of clade A (newly added *Sacchromycopsis* species) appear on the opposite ends of this branch, which is inconsistent with the monophyly of clade A itself. We found that as many as 70% of the amino residues are different between species in clade A, which may be due to hidden paralogy or biological factors, e.g., horizontal gene transfer, a frequently observed phenomenon in many clades of fungal species<sup>36</sup>. By the way, both EOG09343FGH and 7W9S51 are the orthologs of DMP1 gene in *Saccharomyces cerevisiae*<sup>36,41</sup>. In summary, ESL successfully pinpointed conflicting gene-species

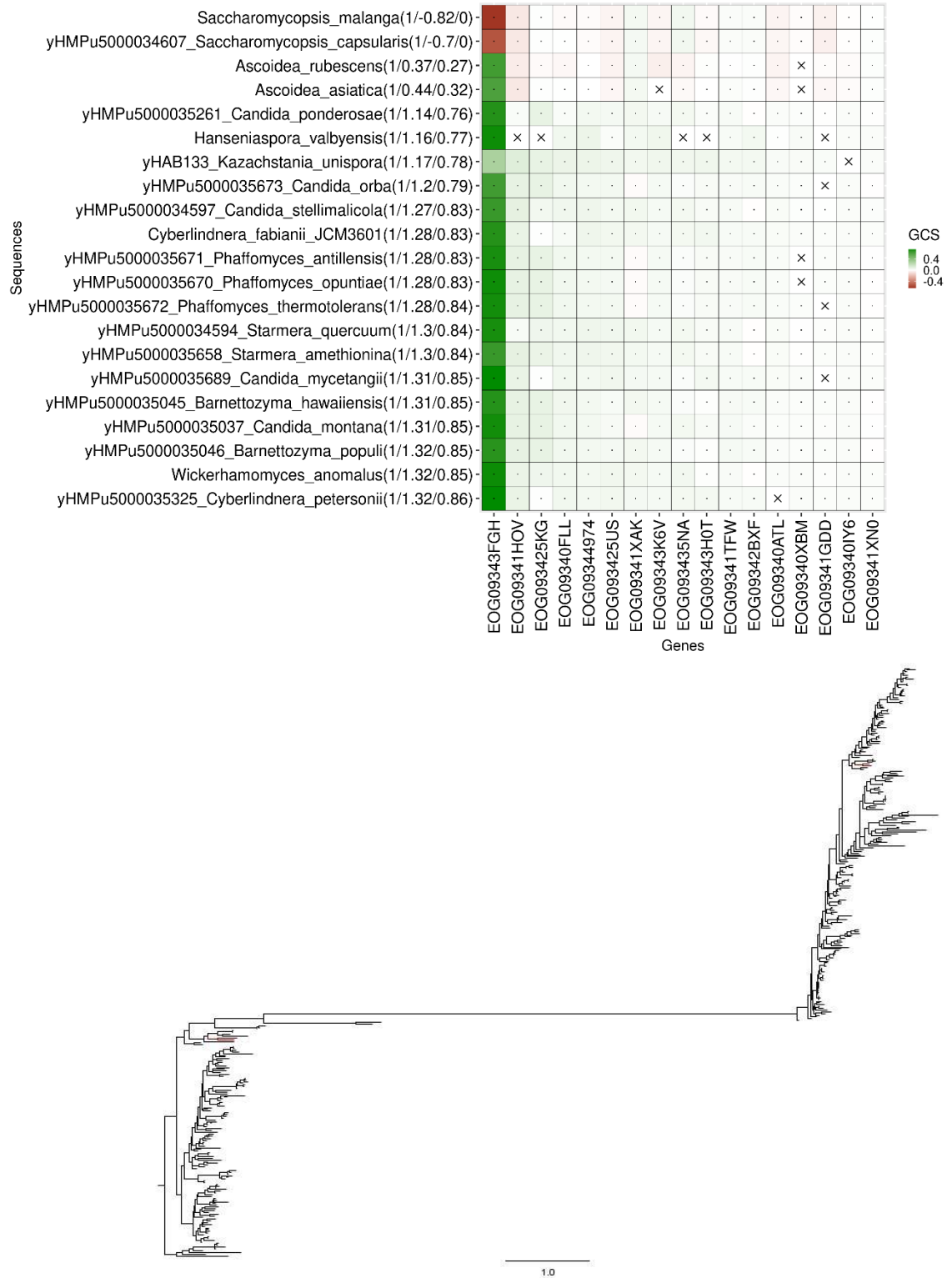


Figure 4.4. Clade model extended fungi dataset. a) The M-grid for extended fungi data. b) Gene

combinations in *Sacchromycopsis* species and EOG09343FGH without needing gene phylogenies or alternative species relationships for clade A+B.

#### ***4.4.4 ESL analysis of a “control” fungi clade.***

In addition to the analysis of a known fragile, we test DrPhylo on a clade that is known to be robust. We selected the monophyletic Saccharomycetaceae clade that was used as a control<sup>41</sup>. DrPhylo analysis of this 36-species clade produced a model with no contradictory signal and very high CP values (0.80) (Figure 4.5). We then used this clade to investigate the ability of DrPhylo to detect contaminant gene-species combinations by deliberately introducing errors in the empirical data matrix. We hypothesized that data errors in the form of gene introgression across species would be exposed by DrPhylo.

First, we simulated introgressions of the most important gene and swapped its sequences between species: one from within the Saccharomycetaceae clade and the other from outside the clade. DrPhylo was run on 100 datasets, each with one such introgression. A total of 98 such introgressions were detected, as GSC became negative for the introgressed gene-species combination. In two cases, the new GSC for the affected combination was close to zero. Another 100 datasets in which a randomly selected Saccharomycetaceae species received a gene replacement from a non-Saccharomycetaceae species, i.e., the horizontal gene transfer was not reciprocal. Again, 98 of these introgressions were detected by DrPhylo. Again, the GSC was only slightly positive (GSC ~0) for two failed cases (Figure 4.5b-c). Therefore, the new metric, GSC, calculated from the clade model could successfully identify the simulated errors because the swapped genes remained no longer concordant with the clade of interest.

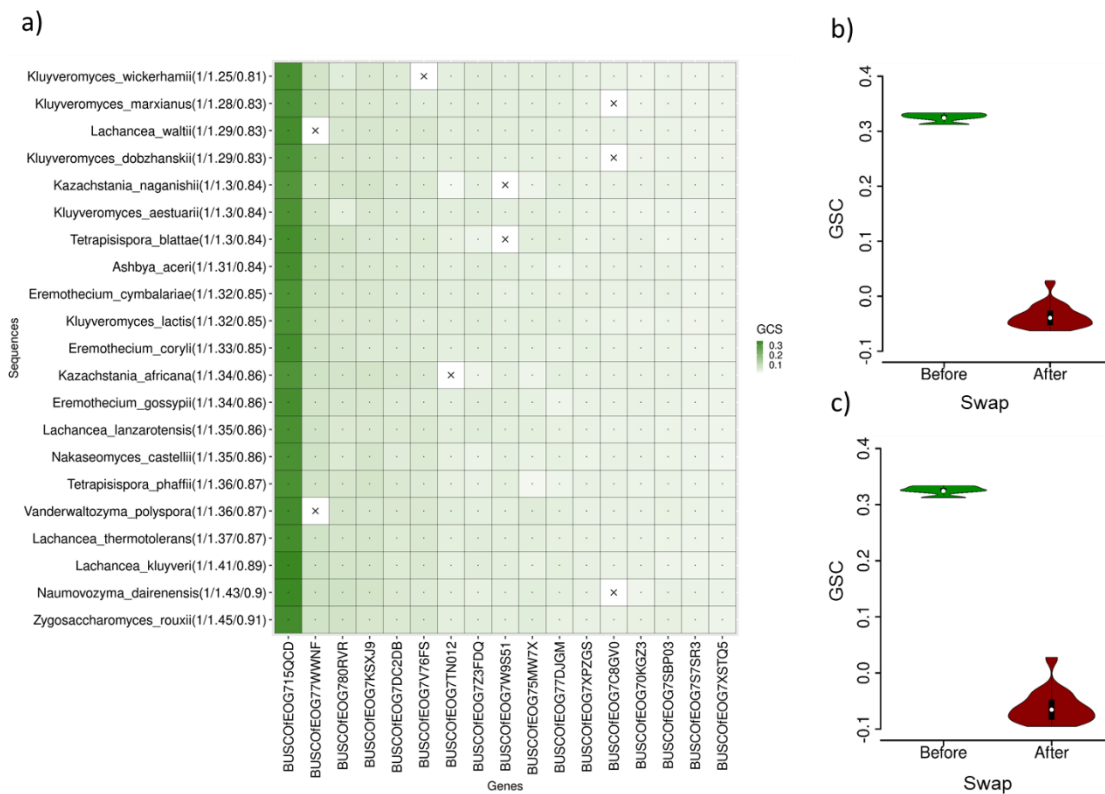


Figure 4.5. Simulated data in control branch. a) The M-grid for the control branch in the fungi phylogeny. Violin plots shows the change in the GSC scores after (b) Reciprocal swaps, and (c) Non-reciprocal Swap. Before swapping, the GSC scores were positive (green violin). The GSC scores became negative (red violin) after swaps.

## 4.5 Analysis of additional empirical datasets

It is evident from the ESL analysis of the fungi dataset that DrPhylo can identify the fragile clade and likely causal genes in species that. To further assess the generality of these results, we analyzed two other empirical datasets from other eukaryotic kingdoms.

### 4.5.1 Analysis of plants dataset

We applied DrPhylo to the phylogeny inferred in an analysis of 620 nuclear gene sequences from 103 plant species, with a focus on identifying the closest relatives of Chloranthales. The concatenated supermatrix approach united Chloranthales and Eudicots (C+E) with a bootstrap support of 100%<sup>37,41</sup>. DrPhylo model found this clade to be fragile, as the CP was low (0.54). The M-grid for this clade in Figure 4.6a revealed many gene-species combinations with a negative GSC for Chloranthales (represented by a single species *Saracandra glabra*). This result is consistent with MSC analysis that also assigned a low posterior probability (< 0.3) for the grouping of Chloranthales with Eudicots. Instead, MSC united Chloranthales with Magnolids, with Eudicots appearing as their sister clade (PP = 71%)<sup>41</sup>.

The M-grid for C+E clade shows that the gene (6040\_C12, 6040 hereafter) is the most influential (Figure 4.6), consistent with a previous study that used two alternative phylogenetic hypotheses about the placement of Chloranthales in ML analyses<sup>41</sup>. However, the clade model provided a greater resolution by revealing that the gene 6040 sequences in three other species of the C+E clade harbor phylogenetic signals that oppose the C+E clade (red cells, Figure. 4.6a). An inspection of the phylogeny of gene 6040 confirmed this

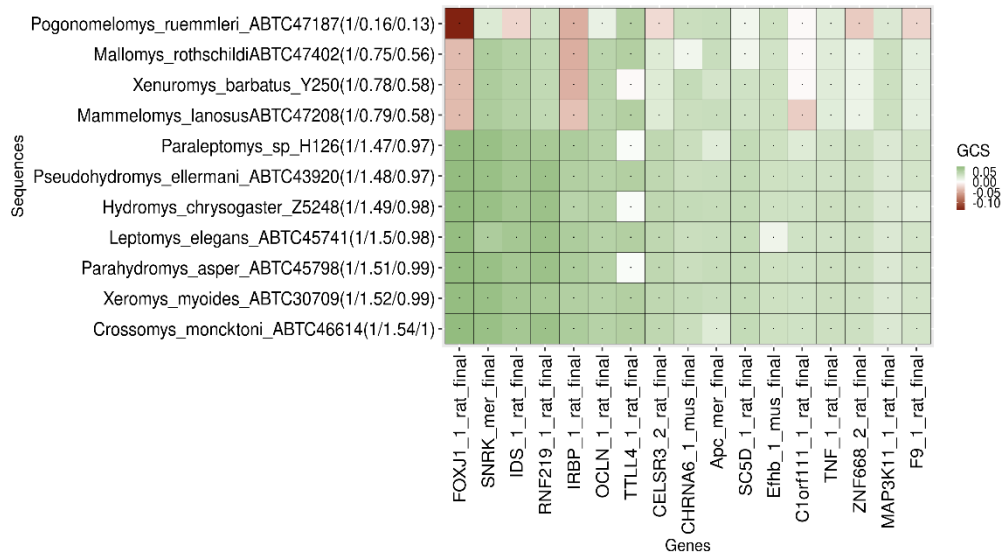
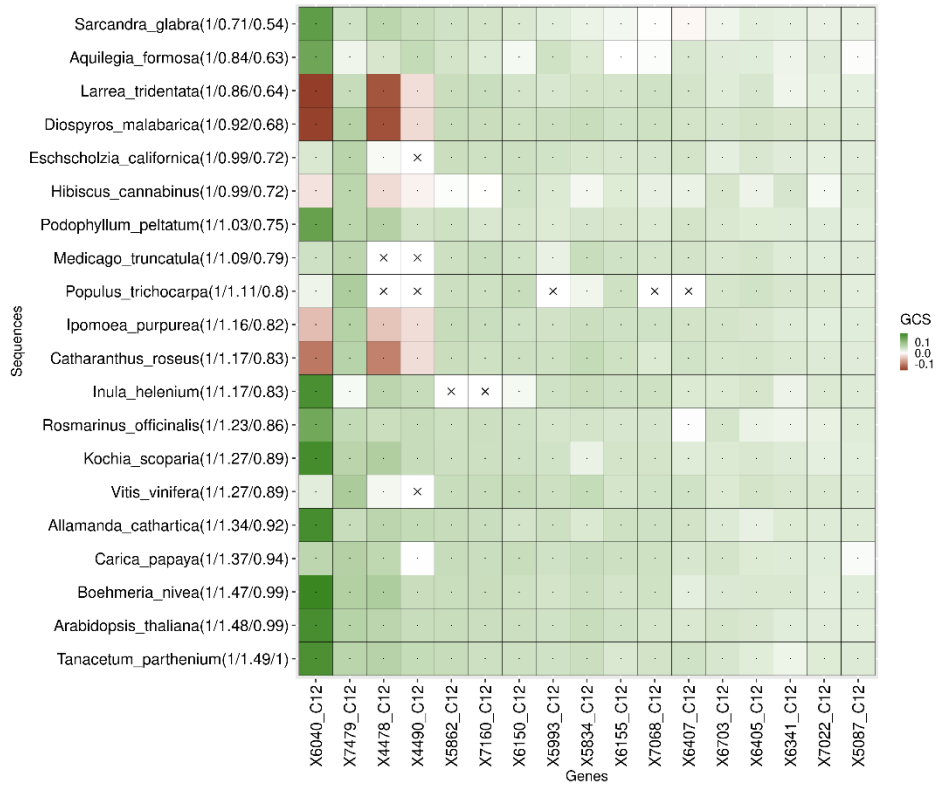


Figure 4.6. Analysis of plants and animal datasets. a) The M-grid for clade C+E from the plants dataset. b) The M-grid for SHL+C. rummelri clade form animal dataset.

diagnosis as these three species were not clustered with other member species of C+E clade and were separated by an unexpectedly long branch (~0.8 substitutions/site). Such a long branch length suggests difficulty with sequence alignment, hidden paralogy, or other types of data errors. Therefore, the new approach for this plant dataset was performed, as well as the fungi datasets presented above.

#### ***4.5.2 Analysis of an animal phylogeny***

We also applied DrPhylo to a dataset of 37 rodents in which the phylogenies inferred by the concatenated supermatrix and MSC approaches differed in their inclusion of *Coccymys ruemmleri* in the SHL clade<sup>132,137</sup>. The MSC phylogenies place *C. ruemmleri* within SHL, giving it a high local PP (95%). But DrPhylo analysis of SHL+*C. ruemmleri* clade produced a very low CP (0.13). This was because of low MP for *C. ruemmleri*, which is caused by large negative GSC values for many genes, e.g., *FOXJ1*, *IDS\_1*, *IRBP\_1*, and *ZNF68\_2* (Figure 4.6b). By the way, the support for placing *C. ruemmleri* inside the SHL clade is driven by the gene *TTL4\_1*, which was also found to be among the top genes in the ML analysis in a previous study<sup>132</sup>.

In contrast to MSC, the concatenation supermatrix approach places *C. ruemmleri* as a sister taxon to SHL with high bootstrap support (rapid bootstrap = 98%). The application of DrPhylo to the SHL clade, excluding *C. ruemmleri*, still produced a low CP (0.12) because a group of three species (*Mallomys rothschildi*, *Mammelomys lanosus*, and *Xenuromys barbatus*) received very low MPs. Interestingly, all of these three species appeared as a sister to *C. ruemmleri* in the MSC phylogeny. This shows that the monophyly of the SHL clade as well as the position of *C. ruemmleri* remains tenuous at present, likely because of incomplete lineage sorting suggested by Shen et al. 2021<sup>132</sup>.

## 4.6 Conclusion

Our approach seeks to identify data partitions (e.g., genes and sites) whose patterns of sequence variation across taxa (such as species) correlate most significantly with the presence of species in an inferred or hypothesized clade. We have shown that evolutionary sparse learning with bilevel sparsity constraints provides a natural approach to employ biological information that categorizes sites into genes or other types of partitions, including exons, introns, codons, contiguous or non-contiguous sites, or other biological partitions. We found that this approach effectively generates genetic models, containing sites and genes as parameters, for any clade in an inferred phylogeny or a collection of taxa. Our success in building these clade models suggests that the "curse of dimensionality," caused by vastly more variable sites than sequences in phylogenomic datasets, is also greatly ameliorated by the use of bilevel sparsity.

Our analysis of empirical and simulated datasets suggests that clade-specific evolutionary sparse learning can assess clade fragility and likely genes-species combinations involved through the use of novel metrics introduced here: namely GSC and CP. Their implementation in DrPhylo as well as the Model-Grid display, present clear visualizations to spot influential as well as problematic sequences. The use of the clade model and different metrics on three diverse datasets produced several novel findings that are also more taxon, gene, and clade-specific than the traditional contrasting of alternative phylogenetic hypotheses by Maximum Likelihood analyses. The ability to find problematic individual gene sequences directly in particular species obviates the need to know alternative phylogenetic hypotheses, which may be too numerous or unknown beforehand. This attribute of our approach also makes the diagnostic analysis of phylogenomic trees

more general, offering opportunities for more effective downstream analysis of the robustness of the inferred phylogeny to idiosyncrasies and errors in the data. Outlier gene-species combinations may themselves be interesting, as they may potentially be created by biological processes such as gene losses and gains and horizontal gene transfers, e.g., ref<sup>39,53,131,142</sup>.

We anticipate ESL to be especially beneficial when only a small fraction of gene-species combinations carry signals conflicting with the taxa membership of the clade. This is because the ESL process of building clade models is unlikely to incorporate genes whose sequences harbor conflicting signals with many member species of the clade. Therefore, if a gene with extensive phylogenetic information to unite species in a clade has a limited number of disruptive gene-species combinations, that gene will be included in the ESL model. Such sequences will receive negative GCS values and be recognizable by red-coloring in the M-matrix. By the way, ESL will work best for clades containing many species (e.g., >5) and a small number of contaminants in the important genes. Both of these properties will allow influential genes to be included in the genetic model. We suggest applying the new approach to well-curated phylogenomic datasets like those analyzed here. Instead, it will be most useful to diagnose fragile clades and associated gene-species combinations after a thoroughly curated phylogenomic dataset has been assembled and phylogeny inferred.

Moreover, it is important to recognize that one run of the new approach may not identify all the data errors and idiosyncrasies in a phylogenomic dataset. One may apply DrPhylo iteratively by excluding the most influential genes one-by-one, which would not be computationally demanding as the current process runs quickly. DrPhylo could also be

applied to all large clades in phylogeny to quickly learn if they may have received spuriously high statistical support. Still, no one method can find all the problems and patterns in a dataset, so we expect DrPhylo to complement many existing methods in the field (reviewed in ref <sup>39</sup>).

## **Chapter 5**

### **EXPLORING NEW FRONTIERS: FUTURE DIRECTIONS FOR NEW METHODS**

#### **5.1 Introduction**

Efficient and accurate phylogenomic inferences analyzing genome-scale sequence data are keys for advancing our understanding of organismal evolutionary relationships. However, the computational resource requirements and the presence of outlier gene-species combinations in genome-scale datasets pose significant challenges in robust and efficient phylogenomic inferences. In previous chapters, innovative phylogenomic subsampling coupled with upsampling was demonstrated as a new paradigm for efficient phylogenomic inferences. The analysis of both empirical and simulated datasets demonstrated that smaller subsamples of sites from the full MSA could effectively estimate bootstrap confidence limits and accurately select the optimal substitution model. Notably, these approaches offer significant resource savings in terms of computational time and require only a minuscule amount of computer memory compared to the analysis of the full MSA. Another significant bottleneck in achieving robust and efficient inference of species relationships is the presence of gene sequences from certain species that exert outlier influence, thereby

contributing to the fragility of certain clades. DrPhylo based on the ESL framework, can efficiently identify such outlier gene sequences and fragile clades in a species phylogeny inferred from any contemporary approaches.

Although the newly proposed approaches have successfully addressed the computational bottleneck in phylogenomic analyses, their potential applications extend beyond this scope. This chapter explores the broader applicability of these approaches in various contexts. Firstly, we examine the general applicability of the little bootstraps approach with median bagging in non-parametric t-tests, showcasing its accuracy in estimating p-values similar to parametric methods. Additionally, we delve into the utility of phylogenomic subsampling for estimating the divergence time of species evolution, particularly addressing the impact of uncertainty in inferred species clades on divergence time estimation and confidence intervals. Finally, we demonstrate how the ESL framework can uncover shared molecular and genomic signatures of convergent evolution by analyzing large-scale sequence datasets. A description on how the ESL model surpasses the limitations of current approaches in detecting the shared genetic basis for convergence of traits among distantly related species in the Tree of Life is provided.

## **5.2 Non-parametric one-sample test using Little Bootstrap**

The little bootstraps (BS)<sup>33</sup>, a subsampling-based approach, estimate the bootstrap confidence limits (BCL) for species clades within an inferred phylogenetic tree. Inspired by the Bag of Little Bootstraps (BLB) method<sup>143</sup>, designed initially to evaluate the quality of parameter estimates (Confidence interval width; CI, Standard error; SE), the little BS approach offers a fast and accurate alternative for assessing the statistical robustness of

species relationships in the genome-scale phylogeny<sup>33</sup>. The novelty of the little BS approach is the use of median-bagging to ensemble subsample-wise confidence limits. To assess the general applicability of the median bagging approach, we evaluated its performance for a non-parametric hypothesis test and compared its results with a parametric test.

Estimating the bootstrap confidence limits on inferred species relationships in a phylogenetic tree is the same as estimating the p-value for a non-parametric test using an indicator variable. For instance, the bootstrap support for a clade,  $C$  is the proportion of inferred trees from bootstrap resamples containing the specific clade. Therefore, the clade support is the sum of an indicator variable, which receive 1 if the clade  $C$  is present in the resampled tree and 0 otherwise. The p-value in a non-parametric test is also estimated in a similar way using an indicator variable. Therefore, we investigated the performance of the little bootstrap approach for estimating p-value in a non-parametric one-sample t-test.

We explored the general applicability of the little bootstrap approach with median bagging using a simulated dataset. First, a set of observations (population,  $N = 100000$ ) was drawn from a normal distribution with a mean 5 and a standard deviation 1, and  $X \sim N(\mu = 5, \sigma = 1)$ . We have drawn 100 random samples of size,  $L = 25000$ , from the simulated population. We set the null hypothesis for this test whether the population mean is greater than or equal to -5.01, and  $H_0: \mu \geq -5.01$ . Therefore, the alternative hypothesis was  $H_1: \mu < -5.01$ . The mean value in the hypothesis was arbitrarily chosen to get a wide range of p-values. The ground truth for estimated p-values from little bootstraps are p-values calculated from the parametric test. Therefore, we also computed each random sample using a parametric one-sample t-test. To perform the parametric t-test, we compute

the test statistic ( $t_0$ ) which is the function of the sample mean ( $\bar{X}$ ) and the standard error (SE) for the sample mean. We compared the test statistic ( $t_0$ ) with the theoretical density from the student-t distribution ( $t_{df}$ ).

$$t_0 = \frac{\bar{X} - \mu}{SE(\bar{X})}$$

$$= \frac{\bar{X} - (-5.01)}{SE(\bar{X})}$$

The p-value is the probability of rejecting the null hypothesis when it is true. We defined the p-value for the parametric test on this hypothesis as

$$P(t_0 \leq t_{df} | H_0) = P_{\text{value}}$$

Hereafter, we consider  $P_{\text{value}}$  as  $P$ , which is the ground truth for our experiment. Next, we estimated p-value using the little bootstraps approach with median bagging. For estimating p-values, we created 100 ( $s = 100$ ) subsamples of size  $l = L^{0.7}$  by random sampling without replacement. Each subsample contained 1200 observations. For each subsample, we generated 100 bootstrap replicates ( $r = 100$ ) by sampling observations with replacement, and each replicate dataset retained the same number of observations as the original sample ( $N = 25,000$ ). We calculated the sample mean ( $\bar{x}$ ) and compared with the null hypothesis. The proportion of time the sample mean greater or equal to the -5.01 is the estimated p-value from the subsample. The estimated p-value from the subsample is defined as

$$\hat{p} = \frac{\sum_{r=1}^{100} I(\bar{x} \geq -5.01)}{r}, \text{ where } I(\bar{x}) = \begin{cases} 1, & \text{if } \bar{x} < -5.01 \\ 0, & \text{if } \bar{x} \geq -5.01 \end{cases}$$

We computed  $\hat{p}$  from each of the subsamples, and the median of 100 is the estimated P-value ( $\hat{P}$ ) by using the little bootstraps approach for the test. For comparing results with BLB, we also estimated p-values using mean bagging, which was the mean of  $\hat{p}$  values from subsamples. We calculated  $\hat{P}$  from each of the 100 samples and for varying subsample sizes ( $g = 0.8$  and  $0.9$ ).

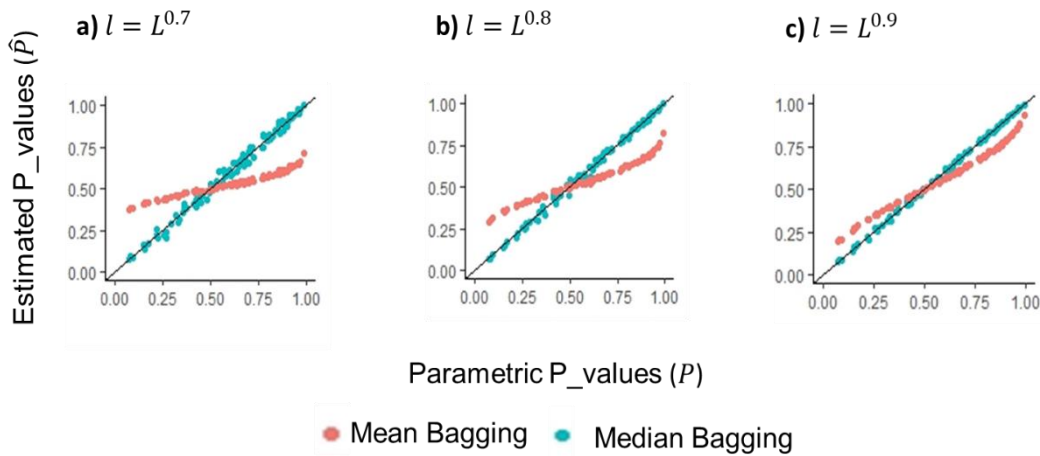


Figure 5.1. Accuracy of little bootstraps with mean and median bagging. These scatter plots show the accuracy of the median bagging compared to the mean bagging approach. The X-axis represents the ground truth of p-values ( $P$ ), and Y-axis represents the little bootstraps estimates ( $\hat{P}$ ) using mean (red), and median (green) bagging. (a) - (c) Accuracy of little bootstraps with mean and median bagging for estimating p-values using different subsample size ( $l$ )

The utilization of the little bootstraps approach with median bagging enables accurate estimation of p-values in the non-parametric setting, with improved performance observed as the subsample size increases (Fig 5.1a-c). However, it should be noted that the mean bagging approach tends to overestimate p-values when the ground truth p-values are below 0.5, while underestimating them for the p-values greater than 0.5. The accuracy of the estimated p-values further improves for both mean and median bagging with larger subsample sizes (Fig 5.1a-c). Similar observations were made in the context of

phylogenomic analysis, as depicted in Figure 2.3a. The over and under-estimations of p-values using mean bagging occurred due to the heavy-tailed distribution of subsample-wise p-values ( $\hat{p}$ ). Therefore, little bootstraps approach with median bagging performs better in estimating p-values using indicator variables.

In the field of phylogenomics, the estimation of bootstrap confidence limits for species clades is performed using an indicator variable that is scored from each replicate dataset. The usage of such an indicator variable is a common practice in resampling-based non-parametric statistical tests, which we have adopted in our experiment. It is noteworthy that our experiment of a one-sample t-test using little bootstraps approach with median bagging exhibits its general applicability, as it can be employed in various statistical tests that utilize an indicator variable. Additionally, this approach holds potential for application in divide-and-conquer methodologies within the field of machine learning.

### **5.3 Application of Little Bootstraps for Divergence Time Estimation**

Estimating the divergence time of organismal evolution helps us add the temporal dimension of the ToL and provide another means of the geological or spatial timescale of the earth's history<sup>72,144</sup>. It is a common practice in molecular systematics that researchers first infer the phylogenetic tree of organisms and subsequently estimate divergence times using a relaxed clock method and calibration points<sup>145,146</sup>. In this process, the inferred tree nodes are used as the true specie relationships without considering the uncertainty of their inferences in the phylogenetic tree. While there is a general belief that phylogenetic uncertainty may impact time estimates and credibility intervals<sup>147–151</sup>, these approaches

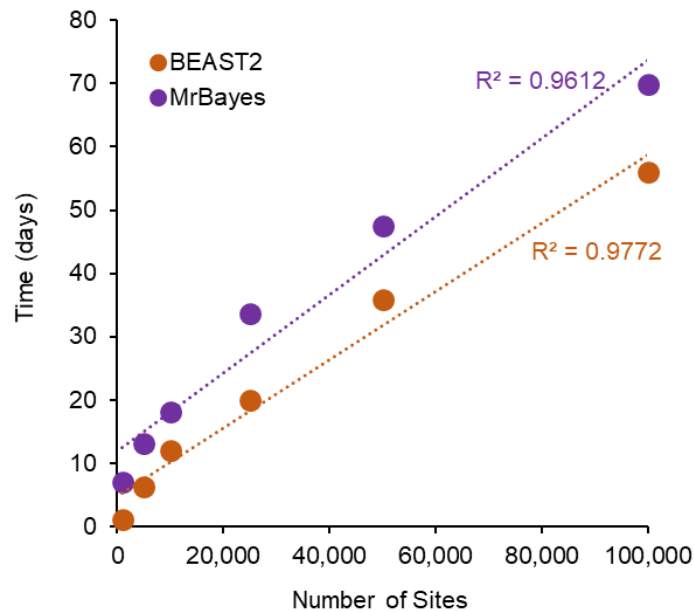


Figure 5.2. Time requirements for divergence time estimation using Bayesian framework. The scatter plot demonstrates a positive correlation between computational time and sequence lengths during the estimation of divergence times in joint analysis within the Bayesian Framework using Beast2 (orange dots) and MrBayes (purple).

overlook the impact of the phylogenetic tree uncertainty in the accuracy of divergence time and their confidence interval estimation.

An alternative strategy is to simultaneously (joint analysis) infer the phylogeny and divergence times, which can incorporate the standard error of the inferred species relationship in both the estimation of divergence times and their CIs. The Bayesian-based framework commonly employs the joint analysis for inferring phylogeny and times of species divergence. However, the computational complexity of Bayesian approaches for joint analysis increases with the number of genes or sites in the datasets (Figure 5.2). Therefore, the computational time as well as the computer memory requirements increases with the sequence length in the dataset. For example, the computational time for a concatenated dataset containing 72 mammalian species and a sequence length of ~33

million sites<sup>152</sup> was projected to require 49 years of CPU time if we perform the Joint analysis in the Bayesian framework. This is because the computational complexity of the underlying maximum likelihood analysis increases with the number of sites in the dataset (Figure 2.1). The computationally efficient relative rate framework does not incorporate phylogenetic uncertainty for estimating divergence times<sup>72,153</sup>.

For this reason, many investigators resort to using data subsamples or data subsets and combining the estimated dates<sup>122,152,154–157</sup>. These divide-and-conquer procedures effectively reduce computational times but often require sequential analysis to estimate times. Therefore, there are no efficient and fast approaches that can estimate the divergence time accurately by incorporating phylogenetic uncertainty.

In my tenure as a Ph.D. student, I have contributed to a research project for developing a computationally efficient approach for molecular dating that can incorporate phylogenetic uncertainty. We developed a bootstraps phylogeny approach for jointly inferring phylogeny and times for large datasets of tens of thousands to millions of sites. Our method uses the Bag of little bootstraps framework (LBS<sup>158</sup>) that analyzes tiny subsamples of site patterns to reduce computational time and memory needs by orders of magnitude. Our method uses the little bootstraps resampling method to generate alternative phylogenies, each subjected to relaxed clock dating using the relative rate framework<sup>72</sup>. The resulting little bootstrap replicate time trees are then used to produce a consensus phylogeny, divergence times, and confidence intervals that automatically incorporate phylogenetic uncertainty. We use the maximum likelihood (ML) approach to infer phylogenies and estimate branch lengths. We also explored using the standard bootstrap

resampling method (BS<sup>58</sup>; Felsenstein 1985) for small datasets that may contain hundreds to thousands of sites.

We present our new approach and compare its performance with Bayesian methods using empirical and computer-simulated datasets. Specifically, we focus on inferred phylogenies containing many clades with low statistical support, addressing gaps in our knowledge about the usefulness of joint inference and the need for computationally efficient methods for bigger datasets. We focused our investigation on dating analyses in which no time constraints on internal nodes were applied, except for a single ingroup root calibration. This choice allowed us to directly examine the power of both methods in dealing with phylogenetic uncertainty without the aid of internal calibrations that are expected to make results from joint analysis and sequential analysis more similar. The root was specified in all the analyses because Bayesian methods may produce biased times when the root is required to be inferred, and the specification of an ingroup clade is a requirement in RelTime.

#### **5.4 Detecting the genomic signature of convergent evolution**

Convergent evolution is a phenomenon where organisms independently develop similar genetic and molecular traits or adaptations in response to their natural environment<sup>159</sup>. This process can be observed across different branches of the Tree of Life<sup>160,161</sup>. For instance, two evolutionarily distant relatives in ToL, bats and toothed whales, have evolved the ability to echolocate, and the convergence of the echolocation trait arises as a consequence of significant transitions to new environments. The identification of the common functional pathway, genes, and/or base substitutions involved in such adaptations has been a

longstanding focus of research in the field of evolutionary biology. However, "the extent to which convergent traits evolve by similar genetic and molecular pathways is unclear"<sup>159</sup>.

Molecular evolutionary investigations conducted to identify the similar genetic basis of convergent evolution found marginally significant enrichment of functional genes and molecular pathways underlying such convergent adaptation. For example, previous studies found enrichment of sound perception genes (false discovery rate, FDR = 0.049) that converged independently and exhibited parallel amino acid substitutions to acquire the echolocation ability<sup>162–164</sup>. These findings suggest the potential presence of a common genetic basis for echolocation in independent clades. However, some studies could not replicate these findings, casting doubt on the robustness and general applicability of the methodology or the existence of a shared genetic basis altogether<sup>163</sup>.

The lack of robustness and marginally statistically significant results may be due to insufficient commonality in the genetic bases of these traits, i.e., different genes and different sites may perform similar functions in independent clades. The inability to fully exclude non-adaptive convergence reduces the statistical power of existing approaches for detecting genes and sites associated with the evolution of convergent traits<sup>165–167</sup>. Furthermore, current state-of-the-art approaches primarily reveal retrospective patterns. Still, they do not explicitly model quantitative genetic changes in convergent trait evolution to make statistical predictions of the presence or absence of the convergent trait.

These challenges can be overcome by employing evolutionary sparse learning (ESL)<sup>35</sup> to construct predictive genetic models for the evolution of convergent traits. In the ESL framework, the genetic model can be designed to identify genes that can predict the presence or absence of a specific trait represented by a binary vector. Notably, species

selection in ESL model building is independent of their phylogenetic relationships. This allows us to include closely related species pairs, with and without the trait, to identify the genetic basis of the convergent trait. The use of paired species may help to effectively mask any neutral (background) sequence convergence, which can yield misleading conclusions and diminish the ability to detect the genetic basis of convergence<sup>165,166,168</sup>. Importantly, the ESL approach considers all genetic loci and their respective substitutions simultaneously during computational analysis, eliminating biases introduced by arbitrary thresholds for evolutionary conservation and convergent substitution cut-offs that are necessary for some other approaches<sup>163,167,169–171</sup>.

In previous years, I have collaborated on a research project for developing a machine-learning approach for detecting genetic signatures of convergent evolution. This approach uses evolutionary sparse learning with novel paired species contrast (ESL-PSC) to produce a quantitative genetic model to predict the presence/absence of a convergent trait in any species based on its genome sequence. Lists of loci selected in the genetic model are subjected to additional analysis to test if there is an enrichment of functional categories relevant to the trait analyzed<sup>172,173</sup>. We applied ESL-PSC to build genetic models of convergent evolution of C4 photosynthesis in grasses and of echolocation in mammals because they have been extensively investigated previously<sup>174–178</sup>. ESL-PSC approach found RuBisCO and other chloroplast proteins that contributed to the convergent evolution of C4 photosynthesis in grasses. For the convergent evolution of echolocation in mammals, proteins selected by the genetic models were highly enriched for the “sensory perception of sound” genes with an adjusted  $P$ -value  $< 10^{-4}$ . This is an improvement in the statistical

significance of more than two orders of magnitude compared to the best previous findings of this term (adjusted  $P = 0.049$ ) in FDR-corrected analyses<sup>164,179</sup>.

## **5.5 Conclusion**

In conclusion, the field of phylogenomics has significantly contributed to our understanding of species evolution. However, the analysis of large-scale genome datasets has posed challenges in terms of computational limitations. Throughout my journey as a graduate student, I have successfully completed three major research projects such as Little Bootstraps<sup>158</sup>, ModelTamer<sup>34</sup>, and DrPhylo. These projects have not only addressed the computational bottleneck in phylogenomic analysis but have also provided valuable tools for researchers in the field of molecular evolution. The broad applicability of these methods and their ability to be applied in various directions underscore their usefulness and potential impact on future research endeavors.

## BIBLIOGRAPHY

1. Aidlin Harari, O. *et al.* Molecular Evolution of the Glutathione S-Transferase Family in the Bemisia tabaci Species Complex. *Genome Biol. Evol.* **12**, 3857–3872 (2020).
2. Pellens, R. & Grandcolas, P. Phylogenetics and Conservation Biology: Drawing a Path into the Diversity of Life. in *Topics in Biodiversity and Conservation* vol. 14 1–15 (Springer, Cham, 2016).
3. Kumar, S. *et al.* An Evolutionary Portrait of the Progenitor SARS-CoV-2 and Its Dominant Offshoots in COVID-19 Pandemic. *Mol. Biol. Evol.* **38**, 3046–3059 (2021).
4. Vázquez, D. P. & Gittleman, J. L. Biodiversity conservation: Does phylogeny matter? *Curr. Biol.* **8**, R379–R381 (1998).
5. Webb, C. O., Ackerly, D. D., McPeck, M. A. & Donoghue, M. J. Phylogenies and Community Ecology. <https://doi.org/10.1146/annurev.ecolsys.33.010802.150448> **33**, 475–505 (2003).
6. Brochier-Armanet, C. & Madern, D. Phylogenetics and biochemistry elucidate the evolutionary link between L-malate and L-lactate dehydrogenases and disclose an intermediate group of sequences with mix functional properties. *Biochimie* **191**, 140–153 (2021).
7. Kapli, P., Yang, Z. & Telford, M. J. Phylogenetic tree building in the genomic age. *Nat. Rev. Genet.* **21**, 428–444 (2020).
8. Miura, S. *et al.* Computational enhancement of single-cell sequences for inferring tumor evolution. *Bioinformatics* **34**, i917–i926 (2018).
9. Shen, X. X. *et al.* Tempo and Mode of Genome Evolution in the Budding Yeast Subphylum. *Cell* **175**, 1533--1545.e20 (2018).
10. Gu, J. *et al.* Phylogeny and species delimitation of the genus Longgenacris and Fruhstorferiola viridifemorata species group (Orthoptera: Acrididae: Melanoplineae) based on molecular evidence. *PLoS One* **15**, e0237882 (2020).
11. Lemieux, J. E. *et al.* Phylogenetic analysis of SARS-CoV-2 in Boston highlights the

- impact of superspreading events. *Science* (80-. ). **371**, (2021).
12. Tamura, K. *et al.* Estimating divergence times in large molecular phylogenies. *Proc. Natl. Acad. Sci. U. S. A.* **109**, 19333–19338 (2012).
  13. Tamura, K., Tao, Q. & Kumar, S. Theoretical Foundation of the RelTime Method for Estimating Divergence Times from Variable Evolutionary Rates. *Mol. Biol. Evol.* **35**, 1770–1782 (2018).
  14. Kenah, E., Britton, T., Halloran, M. E. & Longini, I. M. Molecular Infectious Disease Epidemiology: Survival Analysis and Algorithms Linking Phylogenies to Transmission Trees. *PLOS Comput. Biol.* **12**, e1004869 (2016).
  15. Robuchon, M. *et al.* Revisiting species and areas of interest for conserving global mammalian phylogenetic diversity. *Nat. Commun. 2021 121* **12**, 1–11 (2021).
  16. Kumar, S. *et al.* PathFinder: Bayesian inference of clone migration histories in cancer. *Bioinformatics* **36**, i675–i683 (2020).
  17. Stupp, D. *et al.* Co-evolution based machine-learning for predicting functional interactions between human genes. *Nat. Commun. 2021 121* **12**, 1–14 (2021).
  18. Fukushima, K. & Pollock, D. D. Detecting macroevolutionary genotype–phenotype associations using error-corrected rates of protein convergence. *Nat. Ecol. Evol. 2023 71* **7**, 155–170 (2023).
  19. Hossfeld, U. & Levit, G. S. ‘Tree of life’ took root 150 years ago. *Nat. 2016 5407631* **540**, 38–38 (2016).
  20. Gray, A. I. THE ORIGIN OF SPECIES BY MEANS OF NATURAL SELECTION. in *Darwiniana* 7–50 (Murray, 2014). doi:10.4159/harvard.9780674368552.c2.
  21. Bininda-Emonds, O. R. P., Gittleman, J. L. & Purvis, A. Building large trees by combining phylogenetic information: a complete phylogeny of the extant Carnivora (Mammalia). *Biol. Rev.* **74**, 143–175 (1999).
  22. Turbeville, J. M. C., Schulz, J. R. & Raff, R. A. Deuterostome phylogeny and the sister group of the chordates: evidence from molecules and morphology. *Mol. Biol. Evol.* **11**, 648–655 (1994).
  23. Scotland, R. W., Olmstead, R. G. & Bennett, J. R. Phylogeny Reconstruction : The Role of Morphology Phylogeny Reconstruction : The Role of Morphology. *Society* **52**, 539–548 (2008).

24. Zuckerkandl, E. & Pauling, L. Molecules as documents of evolutionary history. *J. Theor. Biol.* **8**, 357–366 (1965).
25. Brown, H., Sanger, F. & Kitai, R. The structure of pig and sheep insulins. *Biochem. J.* **60**, 556–565 (1955).
26. Fitch, W. M. & Margoliash, E. Construction of phylogenetic trees. *Science (80-. )*. **155**, 279–284 (1967).
27. Farris, J. S. Estimating Phylogenetic Trees from Distance Matrices. *Am. Nat.* **106**, 645–668 (1972).
28. Ferris, S. D., Wilson, A. C. & Brown, W. M. Evolutionary tree for apes and humans based on cleavage maps of mitochondrial DNA. *Proc. Natl. Acad. Sci.* **78**, 2432–2436 (1981).
29. Olsen, G. J. & Woese, C. R. Ribosomal RNA: a key to phylogeny. *FASEB J.* **7**, 113–123 (1993).
30. Sarich, V. M. & Wilson, A. C. Immunological Time Scale for Hominid Evolution. *Science (80-. )*. **158**, 1200–1203 (1967).
31. Kumar, S., Filipski, A. J., Battistuzzi, F. U., Kosakovsky Pond, S. L. & Tamura, K. Statistics and truth in phylogenomics. *Mol. Biol. Evol.* **29**, 457–472 (2012).
32. Nei, M. & Kumar, S. *Molecular evolution and phylogenetics*. (Oxford university press, NY., 2000).
33. Sharma, S. & Kumar, S. Fast and accurate bootstrap confidence limits on genome-scale phylogenies using little bootstraps (CodeOcean, 2021). *Nat. Comput. Sci.* **1**, 573–577 (2021).
34. Sharma, S. & Kumar, S. Taming the Selection of Optimal Substitution Models in Phylogenomics by Site Subsampling and Upsampling. *Mol. Biol. Evol.* **39**, (2022).
35. Kumar, S. & Sharma, S. Evolutionary Sparse Learning for Phylogenomics. *Mol. Biol. Evol.* **38**, 4674–4682 (2021).
36. Shen, X. X. *et al.* Tempo and Mode of Genome Evolution in the Budding Yeast Subphylum. *Cell* **175**, 1533-1545.e20 (2018).
37. Wickett, N. J. *et al.* Phylotranscriptomic analysis of the origin and early diversification of land plants. *Proc. Natl. Acad. Sci. U. S. A.* **111**, E4859–E4868 (2014).

38. Song, S., Liu, L., Edwards, S. V. & Wu, S. Resolving conflict in eutherian mammal phylogeny using phylogenomics and the multispecies coalescent model. *Proc. Natl. Acad. Sci. U. S. A.* **109**, 14942–14947 (2012).
39. Steenwyk, J. L., Li, Y., Zhou, X., Shen, X.-X. & Rokas, A. Incongruence in the phylogenomics era. *Nat. Rev. Genet.* **2023** 1–17 (2023) doi:10.1038/s41576-023-00620-x.
40. Anderson, F. E. & Swofford, D. L. Should we be worried about long-branch attraction in real data sets? Investigations using metazoan 18S rDNA. *Mol. Phylogenet. Evol.* **33**, 440–451 (2004).
41. Shen, X. X., Hittinger, C. T. & Rokas, A. Contentious relationships in phylogenomic studies can be driven by a handful of genes. *Nat. Ecol. Evol.* **2017** *15* **1**, 1–10 (2017).
42. Robinson, D. F. Comparison of labeled trees with valency three. *J. Comb. Theory, Ser. B* **11**, 105–119 (1971).
43. Allen, B. L. & Steel, M. Subtree Transfer Operations and Their Induced Metrics on Evolutionary Trees. *Ann. Comb.* **5**, 1–15 (2001).
44. Hordijk, W. & Gascuel, O. Improving the efficiency of SPR moves in phylogenetic tree search methods based on maximum likelihood. *Bioinformatics* **21**, 4338–4347 (2005).
45. Tamura, K., Stecher, G. & Kumar, S. MEGA11: Molecular Evolutionary Genetics Analysis Version 11. *Mol. Biol. Evol.* **38**, 3022–3027 (2021).
46. Nguyen, L. T., Schmidt, H. A., Von Haeseler, A. & Minh, B. Q. IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Mol. Biol. Evol.* **32**, 268–274 (2015).
47. Stamatakis, A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* **30**, 1312–1313 (2014).
48. Bayzid, M. S., Hunt, T. & Warnow, T. Disk covering methods improve phylogenomic analyses. *BMC Genom.* **15**, S7 (2014).
49. Molloy, E. K. & Warnow, T. NJMerge: A Generic technique for scaling phylogeny estimation methods and its application to species trees. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **11183 LNBI**, 260–276 (2018).

50. Molloy, E. K. & Warnow, T. TreeMerge: a new method for improving the scalability of species tree estimation methods. *Bioinformatics* **35**, i417–i426 (2019).
51. Nelesen, S., Liu, K., Wang, L. S., Randal Linder, C. & Warnow, T. DACTAL: divide-and-conquer trees (almost) without alignments. *Bioinformatics* **28**, i274–i282 (2012).
52. Edwards, S. V. Phylogenomic subsampling: a brief review. *Zoologica Scripta* vol. 45 63–74 (2016).
53. Brown, J. M. & Thomson, R. C. Bayes factors unmask highly variable information content, bias, and extreme influence in phylogenomic analyses. *Syst. Biol.* **66**, 517–530 (2016).
54. Walker, J. F., Brown, J. W. & Smith, S. A. Analyzing contentious relationships and outlier genes in phylogenomics. *Syst. Biol.* **67**, 916–924 (2018).
55. Efron, B. Bootstrap Methods: Another Look at the Jackknife. *Ann. Stat.* **7**, 1–26 (1979).
56. Felsenstein, J. Confidence Limits on Phylogenies: An approach Using the Bootstrap. *Evolution.* **39**, 783–791 (1985).
57. Lemoine, F. *et al.* Renewing Felsenstein’s phylogenetic bootstrap in the era of big data. *Nature* **556**, 452–456 (2018).
58. Efron, B., Halloran, E. & Holmes, S. Bootstrap confidence levels for phylogenetic trees. *Proc. Natl. Acad. Sci. U. S. A.* **93**, 13429–13434 (1996).
59. Delsuc, F., Brinkmann, H. & Philippe, H. Phylogenomics and the reconstruction of the tree of life. *Nat. Rev. Genet.* **6**, 361–375 (2005).
60. Stephens, Z. D. *et al.* Big data: Astronomical or genetical? *PLoS Biol.* **13**, e1002195 (2015).
61. Philippe, H. *et al.* Resolving difficult phylogenetic questions: Why more sequences are not enough. *PLoS Biol.* **9**, (2011).
62. Kumar, S., Filipski, A. J., Battistuzzi, F. U., Kosakovsky Pond, S. L. & Tamura, K. Statistics and truth in phylogenomics. *Mol. Biol. Evol.* **29**, 457–472 (2012).
63. Stamatakis, A., Hoover, P. & Rougemont, J. A rapid bootstrap algorithm for the RAxML web servers. *Syst. Biol.* **57**, 758–771 (2008).
64. Minh, B. Q., Nguyen, M. A. T. & Von Haeseler, A. Ultrafast approximation for

- phylogenetic bootstrap. *Mol. Biol. Evol.* **30**, 1188–1195 (2013).
65. Pattengale, N. D., Alipour, M., Bininda-Emonds, O. R. P., Moret, B. M. E. & Stamatakis, A. How many bootstrap replicates are necessary? *J. Comput. Biol.* **17**, 337–354 (2010).
  66. Hoang, D. T. *et al.* MPBoot: Fast phylogenetic maximum parsimony tree inference and bootstrap approximation. *BMC Evol. Biol.* **18**, 11 (2018).
  67. Kleiner, A., Talwalkar, A., Sarkar, P. & Jordan, M. I. A scalable bootstrap for massive data. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **76**, 795–816 (2014).
  68. Seo, T.-K. Calculating Bootstrap Probabilities of Phylogeny Using Multilocus Sequence Data. *Mol. Biol. Evol.* **25**, 960–971 (2008).
  69. Bickel, P. J., Götze, F. & Van Zwet, W. R. Resampling fewer than n observations: Gains, losses, and remedies for losses. *Stat. Sin.* **7**, 1–31 (1997).
  70. Bickel, P. J. & Freedman, D. A. Some Asymptotic Theory for the Bootstrap. *Ann. Stat.* **9**, 1196–1217 (1981).
  71. Paradis, E. Simulation of phylogenetic data. in *Modern Phylogenetic Comparative Methods and their Application in Evolutionary Biology* 335–350 (Springer Berlin Heidelberg, 2014).
  72. Tamura, K. *et al.* Estimating divergence times in large molecular phylogenies. *Proc. Natl. Acad. Sci. U. S. A.* **109**, 19333–19338 (2012).
  73. Rosenberg, M. S. & Kumar, S. Heterogeneity of nucleotide frequencies among evolutionary lineages and phylogenetic inference. *Mol. Biol. Evol.* **20**, 610–621 (2003).
  74. Battistuzzi, F. U., Filipowski, A., Hedges, S. B. & Kumar, S. Performance of relaxed-clock methods in estimating evolutionary divergence times and their credibility intervals. *Mol. Biol. Evol.* **27**, 1289–1300 (2010).
  75. Tao, Q., Tamura, K., Battistuzzi, F. U. & Kumar, S. A machine learning method for detecting autocorrelation of evolutionary rates in large phylogenies. *Mol. Biol. Evol.* **36**, 811–824 (2019).
  76. Hedges, S. B. & Kumar, S. Discovering the Timetree of Life. in *The Timetree of Life* 3–18 (Oxford Univ Press, New York, 2009).
  77. Abadi, S., Azouri, D., Pupko, T. & Mayrose, I. Model selection may not be a

- mandatory step for phylogeny reconstruction. *Nat. Commun.* **10**, 1–11 (2019).
78. Tavaré, S. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lect. Math. life Sci.* **17**, 57–86 (1986).
  79. Leys, C., Ley, C., Klein, O., Bernard, P. & Licata, L. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *J. Exp. Soc. Psychol.* **49**, 764–766 (2013).
  80. R Core Team. R: A language and environment for statistical computing. *R Foundation for Statistical Computing, Vienna, Austria*. <https://www.R-project.org/> (2020).
  81. Pagès, H., Aboyoun, P., Gentleman, R. & DebRoy, S. Biostrings: Efficient manipulation of biological strings. *R package version 2.46.0* (2017).
  82. Schliep, K. P. phangorn: Phylogenetic analysis in R. *Bioinformatics* **27**, 592–593 (2011).
  83. Allio, R. *et al.* Whole genome shotgun phylogenomics resolves the pattern and timing of swallowtail butterfly evolution. *Syst. Biol.* **69**, 38–60 (2020).
  84. Ran, J. H., Shen, T. T., Wu, H., Gong, X. & Wang, X. Q. Phylogeny and evolutionary history of Pinaceae updated by transcriptomic analysis. *Mol. Phylogenet. Evol.* **129**, 106–116 (2018).
  85. Pessoa-Filho, M., Martins, A. M. & Ferreira, M. E. Molecular dating of phylogenetic divergence between *Urochloa* species based on complete chloroplast genomes. *BMC Genomics* **18**, 1–14 (2017).
  86. Peters, R. S. *et al.* Evolutionary History of the Hymenoptera. *Curr. Biol.* **27**, 1013–1018 (2017).
  87. Peters, R. S. *et al.* Transcriptome sequence-based phylogeny of chalcidoid wasps (Hymenoptera: Chalcidoidea) reveals a history of rapid radiations, convergence, and evolutionary success. *Mol. Phylogenet. Evol.* **120**, 286–296 (2018).
  88. Johnson, D. J., Tress, T., Burkel, N., Taylor, C. & Cesario, J. Officer characteristics and racial disparities in fatal officer-involved shootings. *Proceedings of the National Academy of Sciences of the United States of America* vol. 116 15877–15882 (2019).
  89. Kuntner, M. *et al.* Golden Orbweavers Ignore Biological Rules: Phylogenomic and Comparative Analyses Unravel a Complex Evolution of Sexual Size Dimorphism.

- Syst. Biol.* **68**, 555–572 (2019).
90. Hedin, M., Derkarabetian, S., Alfaro, A., Ramírez, M. J. & Bond, J. E. Phylogenomic analysis and revised classification of atypoid mygalomorph spiders (Araneae, Mygalomorphae), with notes on arachnid ultraconserved element loci. *PeerJ* **7**, e6864 (2019).
  91. Yonezawa, T. *et al.* Phylogenomics and Morphology of Extinct Paleognaths Reveal the Origin and Evolution of the Ratites. *Curr. Biol.* **27**, 68–77 (2017).
  92. Johnson, J. B. & Omland, K. S. Model selection in ecology and evolution. *Trends Ecol. Evol.* **19**, 101–108 (2004).
  93. Abadi, S., Avram, O., Rosset, S., Pupko, T. & Mayrose, I. Modelteller: Model selection for optimal phylogenetic reconstruction using machine learning. *Mol. Biol. Evol.* **37**, 3338–3352 (2020).
  94. Kalyaanamoorthy, S., Minh, B. Q., Wong, T. K. F., von Haeseler, A. & Jermini, L. S. ModelFinder: fast model selection for accurate phylogenetic estimates. *Nat. Methods* **14**, 587–589 (2017).
  95. Buckley, T. R. & Cunningham, C. W. The effects of nucleotide substitution model assumptions on estimates of nonparametric bootstrap support. *Mol. Biol. Evol.* **19**, 394–405 (2002).
  96. Lemmon, A. R. & Moriarty, E. C. The Importance of Proper Model Assumption in Bayesian Phylogenetics. *Syst. Biol.* **53**, 278–298 (2004).
  97. Boni, M. F. *et al.* Evolutionary origins of the SARS-CoV-2 sarbecovirus lineage responsible for the COVID-19 pandemic. *Nat. Microbiol.* **5**, 1408–1417 (2020).
  98. Li, J., Lai, S., Gao, G. F. & Shi, W. The emergence, genomic diversity and global spread of SARS-CoV-2. *Nature* **600**, 408–418 (2021).
  99. Kim, J. *et al.* Reconstruction and evolutionary history of eutherian chromosomes. *Proc. Natl. Acad. Sci.* **114**, E5379–E5388 (2017).
  100. Darriba, D., Taboada, G. L., Doallo, R. & Posada, D. jModelTest 2: more models, new heuristics and parallel computing. *Nat. Methods* **9**, 772–772 (2012).
  101. Posada, D. & Crandall, K. A. MODELTEST: Testing the model of DNA substitution. *Bioinformatics* **14**, 817–818 (1998).
  102. Hurvich, C. M. & Tsai, C. L. Regression and time series model selection in small

- samples. *Biometrika* **76**, 297–307 (1989).
103. Vasilikopoulos, A. *et al.* An integrative phylogenomic approach to elucidate the evolutionary history and divergence times of Neuropterida (Insecta: Holometabola). *BMC Evol. Biol.* **20**, 1–24 (2020).
  104. Haelewaters, D., Park, D. & Johnston, P. R. Multilocus phylogenetic analysis reveals that Cyttariales is a synonym of Helotiales. *Mycol. Prog.* **2021** *2010* **20**, 1323–1330 (2021).
  105. Prasanna, A. N. *et al.* Model Choice, Missing Data, and Taxon Sampling Impact Phylogenomic Inference of Deep Basidiomycota Relationships. *Syst. Biol.* **69**, 17–37 (2020).
  106. Li, Y. *et al.* A genome-scale phylogeny of the kingdom Fungi. *Curr. Biol.* **31**, 1653–1665.e5 (2021).
  107. Posada, D. jModelTest: Phylogenetic model averaging. *Mol. Biol. Evol.* **25**, 1253–1256 (2008).
  108. Darriba, Di. *et al.* ModelTest-NG: A New and Scalable Tool for the Selection of DNA and Protein Evolutionary Models. *Mol. Biol. Evol.* **37**, 291–294 (2020).
  109. Salichos, L. & Rokas, A. Inferring ancient divergences requires genes with strong phylogenetic signals. *Nature* **497**, 327–331 (2013).
  110. Kumar, S., Stecher, G., Peterson, D. & Tamura, K. MEGA-CC: computing core of molecular evolutionary genetics analysis program for automated and iterative data analysis. *Bioinformatics* **28**, 2685–2686 (2012).
  111. dos Reis, M. *et al.* Phylogenomic datasets provide both precision and accuracy in estimating the timescale of placental mammal phylogeny. *Proc. R. Soc. B Biol. Sci.* **279**, 3491–3500 (2012).
  112. Andersen, K. G. *et al.* Clinical Sequencing Uncovers Origins and Evolution of Lassa Virus. *Cell* **162**, 738–750 (2015).
  113. Ruhfel, B. R., Gitzendanner, M. A., Soltis, P. S., Soltis, D. E. & Burleigh, J. G. From algae to angiosperms-inferring the phylogeny of green plants (Viridiplantae) from 360 plastid genomes. *BMC Evol. Biol.* **14**, 1–27 (2014).
  114. Chen, M.-Y., Liang, D., Zhang, P. & Chen, Y. Selecting Question-Specific Genes to Reduce Incongruence in Phylogenomics: A Case Study of Jawed Vertebrate

- Backbone Phylogeny. *Syst. Biol* **64**, 1104–1120 (2015).
115. Kimura, M. Estimation of evolutionary distances between homologous nucleotide sequences. *Proc. Natl. Acad. Sci.* **78**, 454–458 (1981).
  116. Hasegawa, M., Kishino, H. & Yano, T. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.* **22**, 160–174 (1985).
  117. Jones, D. T., Taylor, W. R. & Thornton, J. M. The rapid generation of mutation data matrices from protein sequences. *Bioinformatics* **8**, 275–282 (1992).
  118. Whelan, S. & Goldman, N. A General Empirical Model of Protein Evolution Derived from Multiple Protein Families Using a Maximum-Likelihood Approach. *Mol. Biol. Evol.* **18**, 691–699 (2001).
  119. Le, S. Q. & Gascuel, O. An Improved General Amino Acid Replacement Matrix. *Mol. Biol. Evol.* **25**, 1307–1320 (2008).
  120. Spielman, S. J. & Miraglia, M. L. Relative model selection of evolutionary substitution models can be sensitive to multiple sequence alignment uncertainty. *BMC Ecol. Evol.* **21**, 1–11 (2021).
  121. Kimball, R. T., Hosner, P. A. & Braun, E. L. A phylogenomic supermatrix of Galliformes (Landfowl) reveals biased branch lengths. *Mol. Phylogenet. Evol.* **158**, 107091 (2021).
  122. Dos Reis, M. *et al.* Using Phylogenomic Data to Explore the Effects of Relaxed Clocks and Calibration Strategies on Divergence Time Estimation: Primates as a Test Case. *Syst. Biol.* **67**, 594–615 (2018).
  123. Thode, V. A., Lohmann, L. G. & Sanmartín, I. Evaluating character partitioning and molecular models in plastid phylogenomics at low taxonomic levels: A case study using *Amphilophium* (Bignoniaceae, Bignoniaceae). *J. Syst. Evol.* **58**, 1071–1089 (2020).
  124. Hoff, M., Orf, S., Riehm, B., Darriba, D. & Stamatakis, A. Does the choice of nucleotide substitution models matter topologically? *BMC Bioinformatics* **17**, 1–13 (2016).
  125. Keane, T. M., Creevey, C. J., Pentony, M. M., Naughton, T. J. & McInerney, J. O. Assessment of methods for amino acid matrix selection and their use on empirical data shows that ad hoc assumptions for choice of matrix are not justified. *BMC Evol.*

- Biol.* **6**, 1–17 (2006).
126. Kumar, S. Embracing Green Computing in Molecular Phylogenetics. *Mol. Biol. Evol.* **39**, 43 (2022).
  127. Sharma, Sudip; Kumar, S., Sharma, S. & Kumar, S. Taming the selection of optimal substitution models in Phylogenomics. *CodeOcean* <https://doi.org/10.6084/m9.figshare.19439966.v1> (2022)  
doi:<https://doi.org/10.24433/CO.6690079.v1>.
  128. Young, A. D. & Gillung, J. P. Phylogenomics — principles, opportunities and pitfalls of big-data phylogenetics. *Syst. Entomol.* **45**, 225–247 (2020).
  129. Smith, S. A., Walker-Hale, N. & Walker, J. F. Intra-genetic conflict in phylogenomic data sets. *Mol. Biol. Evol.* **37**, 3380–3388 (2020).
  130. Gadagkar, S. R., Rosenberg, M. S. & Kumar, S. Inferring species phylogenies from multiple genes: Concatenated sequence tree versus consensus gene tree. *J. Exp. Zool. Part B Mol. Dev. Evol.* **304B**, 64–74 (2005).
  131. Chiari, Y., Cahais, V., Galtier, N. & Delsuc, F. Phylogenomic analyses support the position of turtles as the sister group of birds and crocodiles (Archosauria). *Bmc Biol.* **10**, 65 (2012).
  132. Shen, X. X., Steenwyk, J. L. & Rokas, A. Dissecting Incongruence between Concatenation- and Quartet-Based Approaches in Phylogenomic Data. *Syst. Biol.* **70**, 997–1014 (2021).
  133. Salichos, L. Quantifying Phylogenetic Incongruence and Identifying Contributing Factors in a Yeast Model Clade. (Vanderbilt University, 2014).
  134. Philippe, H. *et al.* Pitfalls in supermatrix phylogenomics. *Eur. J. Taxon.* **2017**, 1–25 (2017).
  135. Mongiardino Koch, N. Phylogenomic Subsampling and the Search for Phylogenetically Reliable Loci. *Mol. Biol. Evol.* **38**, 4025–4038 (2021).
  136. Shen, X. X. *et al.* Reconstructing the backbone of the saccharomycotina yeast phylogeny using genome-scale data. *G3 Genes, Genomes, Genet.* **6**, 3927–3939 (2016).
  137. Roycroft, E. J., Moussalli, A. & Rowe, K. C. Phylogenomics Uncovers Confidence and Conflict in the Rapid Radiation of Australo-Papuan Rodents. *Syst. Biol.* **69**,

- 431–444 (2020).
138. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B* **58**, 267–288 (1996).
  139. Hastie, T., Tibshirani, R. & Wainwright, M. *Statistical learning with sparsity: The lasso and generalizations*. (CRC Press: Boca Raton, FL., 2015).
  140. Lundberg, S. M. *et al.* From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2020 21 2**, 56–67 (2020).
  141. Mirarab, S. *et al.* {ASTRAL}: genome-scale coalescent-based species tree estimation. *Bioinformatics* **30**, i541–i548 (2014).
  142. Nakhleh, L. Computational approaches to species phylogeny inference and gene tree reconciliation. *Trends in Ecology and Evolution* vol. 28 719–728 (2013).
  143. Kleiner, A., Jordan, M. I., Talwalkar, A., Sarkar, P. & Jordan, M. I. The big data bootstrap. in *Proceedings of the 29th International Conference on Machine Learning, ICML 2012* vol. 2 1759–1766 (2012).
  144. Kumar, S. & Blair Hedges, S. Advances in Time Estimation Methods for Molecular Data. *Mol. Biol. Evol.* **33**, 863–869 (2016).
  145. dos Reis, M., Donoghue, P. C. J. & Yang, Z. Bayesian molecular clock dating of species divergences in the genomics era. *Nat. Rev. Genet.* **17**, 71–80 (2016).
  146. Tao, Q., Kumar, S. & Tamura, K. Efficient Methods for Dating Evolutionary Divergences. in *The Molecular Evolutionary Clock* (ed. Ho, S. Y. W.) 197–220 (Springer US, 2020).
  147. Drummond, A. J., Ho, S. Y. W., Phillips, M. J. & Rambaut, A. Relaxed phylogenetics and dating with confidence. *PLoS Biol.* **4**, 699–710 (2006).
  148. Ho, S. Y. W. & Phillips, M. J. Accounting for calibration uncertainty in phylogenetic estimation of evolutionary divergence times. *Syst. Biol.* **58**, 367–380 (2009).
  149. Ronquist, F. *et al.* MrBayes 3.2: Efficient Bayesian Phylogenetic Inference and Model Choice Across a Large Model Space. *Syst. Biol.* **61**, 539–542 (2012).
  150. Ho, S. Y. W. & Duchêne, S. Molecular-clock methods for estimating evolutionary rates and timescales. *Mol. Ecol.* **23**, 5947–5965 (2014).
  151. Bromham, L. *et al.* Bayesian molecular dating: opening up the black box. *Biol. Rev.* **93**, 1165–1191 (2018).

152. Álvarez-Carretero, S. *et al.* A species-level timeline of mammal evolution integrating phylogenomic data. *Nature* **602**, 263–267 (2022).
153. Tao, Q., Tamura, K. & Kumar, S. Efficient Methods for Dating Evolutionary Divergences. *Mol. Evol. Clock Theory Pract.* 197–219 (2020) doi:10.1007/978-3-030-60181-2\_12.
154. Jetz, W., Thomas, G. H., Joy, J. B., Hartmann, K. & Mooers, A. O. The global diversity of birds in space and time. *Nature* **491**, 444–448 (2012).
155. Tonini, J. F. R., Beard, K. H., Ferreira, R. B., Jetz, W. & Pyron, R. A. Fully-sampled phylogenies of squamates reveal evolutionary patterns in threat status. *Biol. Conserv.* **204**, 23–31 (2016).
156. Jetz, W. & Pyron, R. A. The interplay of past diversification and evolutionary isolation with present imperilment across the amphibian tree of life. *Nat. Ecol. Evol.* **2**, 850–858 (2018).
157. Upham, N. S., Esselstyn, J. A. & Jetz, W. Inferring the mammal tree: Species-level sets of phylogenies for questions in ecology, evolution, and conservation. *PLoS Biol.* **17**, 1–44 (2019).
158. Sharma, S. & Kumar, S. Fast and accurate bootstrap confidence limits on genome-scale phylogenies using little bootstraps. *Nat. Comput. Sci.* **1**, 573–577 (2021).
159. Sackton, T. B. & Clark, N. Convergent evolution in the genomics era: new insights and directions. *Philos. Trans. R. Soc. B* **374**, (2019).
160. Farhat, M. R. *et al.* Genomic analysis identifies targets of convergent positive selection in drug-resistant *Mycobacterium tuberculosis*. *Nat. Genet.* **45**, 1183–1189 (2013).
161. Chen, L., Devries, A. L. & Cheng, C. H. C. Convergent evolution of antifreeze glycoproteins in Antarctic notothenioid fish and Arctic cod. *Proc. Natl. Acad. Sci. U. S. A.* **94**, 3817–3822 (1997).
162. Marcovitz, A. *et al.* A functional enrichment test for molecular convergent evolution finds a clear protein-coding signal in echolocating bats and whales. *Proc. Natl. Acad. Sci.* **116**, 21094–21103 (2019).
163. Lee, J. H. *et al.* Molecular parallelism in fast-twitch muscle proteins in echolocating mammals. *Sci. Adv.* **4**, 9660–9686 (2018).

164. Liu, Z., Qi, F. Y., Xu, D. M., Zhou, X. & Shi, P. Genomic and functional evidence reveals molecular insights into the origin of echolocation in whales. *Sci. Adv.* **4**, (2018).
165. Zou, Z. & Zhang, J. No Genome-Wide Protein Sequence Convergence for Echolocation. *Mol. Biol. Evol.* **32**, 1237–1241 (2015).
166. Thomas, G. W. C. & Hahn, M. W. Determining the Null Model for Detecting Adaptive Convergence from Genomic Data: A Case Study using Echolocating Mammals. *Mol. Biol. Evol.* **32**, 1232–1236 (2015).
167. Xu, S. *et al.* Genome-Wide Convergence during Evolution of Mangroves from Woody Plants. *Mol. Biol. Evol.* **34**, 1008–1015 (2017).
168. Parker, J. *et al.* Genome-wide signatures of convergent evolution in echolocating mammals. *Nat. 2013 5027470* **502**, 228–231 (2013).
169. Marcovitz, A. *et al.* A functional enrichment test for molecular convergent evolution finds a clear protein-coding signal in echolocating bats and whales. *Proc. Natl. Acad. Sci. U. S. A.* **116**, 21094–21103 (2019).
170. Yuan, Y. *et al.* Comparative genomics provides insights into the aquatic adaptations of mammals. *Proc. Natl. Acad. Sci. U. S. A.* **118**, e2106080118 (2021).
171. He, Z. *et al.* Convergent adaptation of the genomes of woody plants at the land–sea interface. *Natl. Sci. Rev.* **7**, 978–993 (2020).
172. Subramanian, A. *et al.* Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U. S. A.* **102**, 15545–15550 (2005).
173. Huang, D. W., Sherman, B. T. & Lempicki, R. A. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res.* **37**, 1–13 (2009).
174. Liu, Y. *et al.* Convergent sequence evolution between echolocating bats and dolphins. *Curr. Biol.* **20**, R53–R54 (2010).
175. Li, Y., Liu, Z., Shi, P. & Zhang, J. The hearing gene Prestin unites echolocating bats and whales. *Curr. Biol.* **20**, R55–R56 (2010).
176. Christin, P. A. *et al.* Evolutionary Switch and Genetic Convergence on *rbcl* following the Evolution of C4 Photosynthesis. *Mol. Biol. Evol.* **25**, 2361–2368

(2008).

177. Parto, S. & Lartillot, N. Molecular adaptation in Rubisco: Discriminating between convergent evolution and positive selection using mechanistic and classical codon models. *PLoS One* **13**, e0192697 (2018).
178. Casola, C. & Li, J. Beyond RuBisCO: convergent molecular evolution of multiple chloroplast genes in C4 plants. *PeerJ* **10**, e12791 (2022).
179. Chabrol, O., Royer-Carenzi, M., Pontarotti, P. & Didier, G. Detecting the molecular basis of phenotypic convergence. *Methods Ecol. Evol.* **9**, 2170–2180 (2018).

## APPENDICES

### A. R CODES FOR AUTOMATIC LITTLE BOOTSTRAPS ANALYSIS USING IQTREE

```
lb_automatic <- function(data_path, candidate_tree, evo_model = NULL,
output_tree = NULL, del = 0.001, precision = FALSE){

  # data_path      : Directory path where the fasta file located
  # candidate_tree: The candidate tree that will be assessed
  # evo_model      : Substitution model
  # output_tree    : Output tree file name
  # del            : Treshold value for choosing s and r
  # precision      : Precision (SE) for little bootstrap BCL's
  #               : If true there will be an output tree with precision

  ##### Package required #####

  if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

  if (!requireNamespace("Biostrings", quietly = TRUE))
    BiocManager::install("Biostrings")

  if (!requireNamespace("stringr", quietly = TRUE))
    install.packages("stringr")

  if (!requireNamespace("ape", quietly = TRUE))
    install.packages("ape")

  if (!requireNamespace("phyclust", quietly = TRUE))
    install.packages("phyclust")

  if (!requireNamespace("phangorn", quietly = TRUE))
    install.packages("phangorn")

  ##### Library Required #####

  if (!library('Biostrings',logical.return = TRUE)){
    stop("'Biostrings' package not found, please install it to run
```

```

        lb_automatic")
    }

    if (!library('stringr',logical.return = TRUE)){
        stop("'stringr' package not found, please install it to run
            lb_automatic")
    }

    if (!library('phyclust',logical.return = TRUE)){
        stop("'phyclust' package not found, please install it to run
            lb_automatic")
    }

#####Helper Function #####

aggregator <- function(path, tree_format, candidate_tree, s = NULL, r = NULL,
output_tree = NULL){

    # path          : Directory path where tree file for each replicate
                    : datasets located
    # tree_format   : Tree file format (.nwk, .treefile)
    # candidate_tree: The candidate tree that will be assessed
    # s             : Number of subsamples
    # r             : Number of replicates
    # output_tree   : Output tree file name

##### Required Package #####

    if (!requireNamespace("ape", quietly = TRUE))
        install.packages("ape")

    if (!requireNamespace("phangorn", quietly = TRUE))
        install.packages("phangorn")

##### Library required #####

    if (!library('ape',logical.return = TRUE)){
        stop("'ape' package not found, please install it to run aggregator")
    }

    if (!library('phangorn',logical.return = TRUE)){
        stop("'phangorn' package not found, please install it to run
            aggregator")
    }

#####

    sub          <- NULL
    sub_dir      <- dir(path, pattern = "Subsample", full.names = T)
    candidate_tree <- ape::read.tree(candidate_tree)

    if(is.null(s) == T) {
        s <- length(sub_dir)
    }

    for(k in 1:s){
        di <- sub_dir[k]
        lf <- list.files(di, pattern = tree_format, full.names = TRUE)

        if(is.null(r) == T) {
            r <- length(lf)
        }
    }

```

```

x<- ape::rmtree(r, candidate_tree$Nnode)

for(l in 1:r){ # number of replicates
  x[l] <- list(ape::read.tree(lf[l]))
  print(c("Subsample=", k, "Replicate=", l))
}
ff <- tempfile()
png(filename = ff)
b <- phangorn::plotBS(candidate_tree, x, p =10, type = "unrooted")
dev.off()
unlink(ff)
sub <- cbind(sub, as.numeric(b$node.label)/100)

}
med_sup <- apply(sub, 1, function(x){median(x, na.rm = T)})
b$node.label <- med_sup
if(is.null(output_tree) == T) {
  output_tree <- 'output_tree_lb.nwk'
}else{
  output_tree <- paste(output_tree, '.nwk', sep = "")
}

ape::write.tree(b, file = output_tree)
}
lb_avg_bcl <- function(path, tree_format, candidate_tree, s = NULL, r =
  NULL){

# path          : Directory path where tree file for each replicate
                  datasets located
# tree_format   : Tree file format (.nwk, .treefile)
# candidate_tree: The candidate tree that will be assessed
# s             : Number of subsamples
# r             : Number of replicates
# output_tree   : Output tree file name

##### Required Package #####

if (!requireNamespace("ape", quietly = TRUE))
  install.packages("ape")

if (!requireNamespace("phangorn", quietly = TRUE))
  install.packages("phangorn")

##### Library required #####

if (!library('ape',logical.return = TRUE)){
  stop("'ape' package not found, please install it to lb_avg_bcl")
}

if (!library('phangorn',logical.return = TRUE)){
  stop("'phangorn' package not found, please install it to lb_avg_bcl")
}

#####

sub          <- NULL
sub_dir     <- dir(path, pattern = "Subsample", full.names = T)
candidate_tree <- ape::read.tree(candidate_tree)

if(is.null(s) == T) {
  s <- length(sub_dir)
}

```

```

for(k in 1:s){
  di <- sub_dir[k]
  lf <- list.files(di, pattern = tree_format, full.names = TRUE)

  if(is.null(r) == T) {
    r <- length(lf)
  }

  x<- ape::rmtree(r, candidate_tree$Nnode)

  for(l in 1:r){ # number of replicates
    x[l] <- list(ape::read.tree(lf[l]))
    print(c("Subsample=", k, "Replicate=", l))
  }
  ff <- tempfile()
  png(filename = ff)
  b <- phangorn::plotBS(candidate_tree, x, p =10, type = "unrooted")
  dev.off()
  unlink(ff)
  sub <- cbind(sub, as.numeric(b$node.label)/100)
}
med_sup <- apply(sub, 1, function(x){median(x, na.rm = T)})
return(mean(med_sup, na.rm = T))
}

lb_precision <- function(path, tree_format, candidate_tree, s = NULL, r =
  NULL, rep = 100, output_tree = NULL){

# path      : Directory path where tree file for each replicate
              datasets located
# tree_format : Tree file format (.nwk, .treefile)
# candidate_tree: The candidate tree that will be assessed
# s          : Number of subsamples
# r          : Number of replicates
# rep       : Number of replicate for computing precision of lb
              BCL's
# output_tree : Output tree file name

##### Required Package #####

if (!requireNamespace("ape", quietly = TRUE))
  install.packages("ape")

if (!requireNamespace("phangorn", quietly = TRUE))
  install.packages("phangorn")

##### Library required #####

if (!library('ape',logical.return = TRUE)){
  stop("'ape' package not found, please install it to run aggregator")
}

if (!library('phangorn',logical.return = TRUE)){
  stop("'phangorn' package not found, please install it to run
  aggregator")
}

#####

sub          <- NULL
sub_dir     <- dir(path, pattern = "Subsample", full.names = T)

```

```

candidate_tree <- ape::read.tree(candidate_tree)

if(is.null(s) == T) {
  s <- length(sub_dir)
}

for(k in 1:s){

  di <- sub_dir[k]
  lf <- list.files(di, pattern = tree_format, full.names = TRUE)

  if(is.null(r) == T) {
    r <- length(lf)
  }

  x<- ape::rmtree(r, candidate_tree$Nnode)
  for(l in 1:r){ # number of replicates
    x[l] <- list(ape::read.tree(lf[l]))
    print(c("Subsample=", k, "Replicate=", l))
  }
  ff <- tempfile()
  png(filename = ff)
  b <- phangorn::plotBS(candidate_tree, x, p =10, type = "unrooted")
  dev.off()
  unlink(ff)
  sub <- cbind(sub, as.numeric(b$node.label)/100)

}
med_sup <- apply(sub, 1, function(x){median(x, na.rm = T)})
b$node.label <- med_sup
b_tree <- b
rm(b)

#####Precision calculation#####

sub_all <- NULL
for(m in 1:rep){
  sub <- NULL
  sub_dir <- dir(path, pattern = "Subsample", full.names = T)

  if(is.null(s) == T) {
    s <- length(sub_dir)
  }

  ran_sub <- sample(1:s, s, replace = T)
  for(k in 1:s){
    di <- sub_dir[ran_sub[k]]
    lf <- list.files(di, pattern = tree_format, full.names = TRUE)

    if(is.null(r) == T) {
      r <- length(lf)
    }

    x<- ape::rmtree(r, candidate_tree$Nnode)
    ran_rep <- sample(1:r, r, replace = T)
    for(l in 1:r){ # number of replicates
      x[l] <- list(ape::read.tree(lf[ran_rep[l]]))
    }
    ff <- tempfile()
    png(filename = ff)
    b <- phangorn::plotBS(candidate_tree, x, p =10, type = "unrooted")
    dev.off()
  }
}

```

```

        unlink(ff)
        sub <- cbind(sub, as.numeric(b$node.label)/100)
    }
    med_sup <- apply(sub, 1, function(x){median(x, na.rm = T)})
    sub_all <- cbind(sub_all, med_sup)
    print(c("Replicate = ", m))
}
pre <- apply(sub_all, 1, function(x){sd(x, na.rm = T)})
b$node.label <- pre

if(is.null(output_tree) == T) {
    output_tree <- 'output_tree_lb.nwk'
    output_tree_p <- 'output_tree_lb_precision.nwk'
}else{
    output_tree <- paste(output_tree, '.nwk', sep = "")
    output_tree_p <- paste(output_tree, '_precision.nwk', sep = "")
}

ape::write.tree(b_tree, file = output_tree)
ape::write.tree(b, file = output_tree_p)

}

#####
## Main Function
#####

f_name <- data_path
sub_name <- str_replace(basename(data_path), ".fasta", "")
motherfile <- Biostrings::readAAStringSet(f_name, format = "fasta")
sln <- as.numeric(fasta.seqlengths(f_name)[1])
directory <- getwd()
a <- phyclust::read.fasta(data_path)
a <- a$org.code
a <- unique(a, MARGIN = 2)
if(dim(a)[2] > 100000){
    g <- 0.7
}else{
    g <- 0.8
}
rm(a)
s1 <- list()
for(i in 1:4){
    setwd(directory)
    sub_dir <- paste("Subsample", i , sep = "")
    dir.create(sub_dir)
    # Create Subsample folder where Replicates will be kept
    setwd(sub_dir)
    s1 <- append(s1, list(sample(1:sln, ceiling(sln^g), replace = F)))
# Setting subsample length and sites. The power value changes [0.5, 1]
#subsample[[]] <- endoapply(motherfile, function(x) x[s])
# Making subsample

    for(j in 1:4){
        s2 <- sample(s1[[i]], sln, replace = T)
        upsample <- endoapply(motherfile, function(x) x[s2])
        file_name <- paste(sub_name, '_sub', i, "rep", j, ".fasta", sep = "")
        Biostrings::writeXStringSet(upsample, file_name) #
Saving Replicates
        #print(c('Subsample=',i, 'Replicates=', j))
    }
}

```

```

}
setwd(directory)
for(i in 1:4){
  for(j in 1:4){
    dname <- paste(sub_name, '_sub', i, "rep", j, ".fasta", sep = "")
    if(is.null(evo_model) == F){
      shell(paste("iqtree -s", paste("Subsample", i, "/", dname, sep =
        ""), "-m", evo_model, sep = " "))
    }else{
      shell(paste("iqtree -s", paste("Subsample", i, "/", dname, sep =
        ""), sep = " "))
    }
  }
}

#####
sub_avg <- c(0,0)
rep_avg <- c(0,0)

for(j in 3:4){
  a <- lb_avg_bcl(dirname(data_path), ".treefile", candidate_tree, s = 3,
    r = j)
  rep_avg <- c(rep_avg, a)
}

while(abs(rep_avg[j]-rep_avg[j-1])>del){
  j = j +1
  for(m in 1:i){
    for(n in j:j){
      s2 <- sample(s1[[m]], sln, replace = T)
      upsample <- endoapply(motherfile, function(x) x[s2])
      file_name <- paste(sub_name, '_sub', m, "rep", n, ".fasta", sep= "")
      setwd(paste("Subsample", m, sep = ""))
      Biostrings::writeXStringSet(upsample, file_name)
      setwd(directory)
      if(is.null(evo_model) == F){
        shell(paste("iqtree -s", paste("Subsample", m, "/", file_name,
          sep = ""), "-m", evo_model, sep = " "))
      }else{
        shell(paste("iqtree -s", paste("Subsample", m, "/", file_name,
          sep = ""), sep = " "))
      }
      #print(c('Subsample=',m, 'Replicates=', n))
    }
  }
  a <- lb_avg_bcl(dirname(data_path), ".treefile", candidate_tree, s = i,
    r = j)
  rep_avg <- c(rep_avg, a)
}

for(i in 3:4){
  a <- lb_avg_bcl(dirname(data_path), ".treefile", candidate_tree, s = i,
    r = j)
  sub_avg <- c(sub_avg, a)
}
while(abs(sub_avg[i]-sub_avg[i-1])>del){
  i = i +1
  print(i)
  for(m in i:i){
    setwd(directory)
    sub_dir <- paste("Subsample", i , sep = "")
    dir.create(sub_dir)
    setwd(sub_dir)
  }
}

```

```

s1 <- append(s1, list(sample(1:sln, ceiling(sln^g), replace = F)))
subsample[[]] <- endoapply(motherfile, function(x) x[s
for(n in 1:j){
  s2 <- sample(s1[[m]], sln, replace = T)
  upsample <- endoapply(motherfile, function(x) x[s2])
  file_name <- paste(sub_name, '_sub', i, "rep", j, ".fasta", sep =
    "")
  Biostrings::writeXStringSet(upsample, file_name)
  setwd(directory)
  if(is.null(evo_model) == F){
    shell(paste("iqtree -s", paste("Subsample", m, "/", file_name,
      sep = ""), "-m", evo_model, sep = " "))
  }else{
    shell(paste("iqtree -s", paste("Subsample", m, "/", file_name,
      sep = ""), sep = " "))
  }
  setwd(sub_dir)
  #print(c('Subsample=', i, 'Replicates=', j))
}
}
a <- lb_avg_bcl(dirname(data_path), ".treefile", candidate_tree, s = i,
  r = j)
sub_avg <- c(sub_avg, a)
}
setwd(directory)

j = j + 1
for(m in 1:i){
  for(n in j:j){
    s2 <- sample(s1[[m]], sln, replace = T)
    upsample <- endoapply(motherfile, function(x) x[s2])
    file_name <- paste(sub_name, '_sub', m, "rep", n, ".fasta", sep = "")
    setwd(paste("Subsample", m, sep = ""))
    Biostrings::writeXStringSet(upsample, file_name)
    setwd(directory)
    if(is.null(evo_model) == F){
      shell(paste("iqtree -s", paste("Subsample", m, "/", file_name, sep
        = ""), "-m", evo_model, sep = " "))
    }else{
      shell(paste("iqtree -s", paste("Subsample", m, "/", file_name, sep
        = ""), sep = " "))
    }
    #print(c('Subsample=', m, 'Replicates=', n))
  }
}
a <- lb_avg_bcl(dirname(data_path), ".treefile", candidate_tree, s = i, r
  = j)
rep_avg <- c(rep_avg, a)

while(abs(rep_avg[j]-rep_avg[j-1])>del){
  j = j + 1
  for(m in 1:i){
    for(n in j:j){
      s2 <- sample(s1[[m]], sln, replace = T)
      upsample <- endoapply(motherfile, function(x) x[s2])
      file_name <- paste(sub_name, '_sub', m, "rep", n, ".fasta", sep="")
      setwd(paste("Subsample", m, sep = ""))
      Biostrings::writeXStringSet(upsample, file_name)
      setwd(directory)
      if(is.null(evo_model) == F){
        shell(paste("iqtree -s", paste("Subsample", m, "/", file_name,
          sep = ""), "-m", evo_model, sep = " "))
      }else{

```

```

        shell(paste("iqtree -s", paste("Subsample", m, "/", file_name,
        sep = ""), sep = " "))
    }
    #print(c('Subsample=',m, 'Replicates=', n))
}
}
a <- lb_avg_bcl(dirname(data_path), ".treefile", candidate_tree, s = 3,
               r = i)
rep_avg <- c(rep_avg, a)
}
print(c("Subsample = ", i, "Replicate = ", j))
setwd(directory)

if(precision == FALSE){
    aggregator(dirname(data_path), ".treefile", candidate_tree, s = i, r =
               j, output_tree = output_tree)
}else{
    lb_precision(dirname(data_path), ".treefile", candidate_tree, s = i, r
                 = j, output_tree = output_tree)
}
}
}

```

## B. R CODES FOR AUTOMATIC MODEL TAMER ANALYSIS USING IQTREE

```
MT_automatic <- function(data_path, data_type = c("DNA", "AA"), Redo = FALSE,
max.iter = 2){

  # data_path    : path for the sequence alignment in fasta format
  # data_type    : DNA or Amino Acid (AA)

  ##### Package required #####

  if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

  if (!requireNamespace("Biostrings", quietly = TRUE))
    BiocManager::install("Biostrings")

  if (!requireNamespace("stringr", quietly = TRUE))
    install.packages("stringr")

  if (!requireNamespace("lubridate", quietly = TRUE))
    install.packages("lubridate")

  ##### Library required #####

  if (!library('Biostrings', logical.return = TRUE)){
    stop("'Biostrings' package not found, please install it to run
    MT_automatic")
  }

  if (!library('stringr', logical.return = TRUE)){
    stop("'stringr' package not found, please install it to run MT_automatic")
  }

  if (!library('lubridate', logical.return = TRUE)){
    stop("'lubridate' package not found, please install it to run
    MT_automatic")
  }

  ##### ModelTamer Sampling #####

  MT_sampler_auto <- function(MSA, num_of_dsp, distinct_positions, g,
                              SU_name, sln = sln){

    # data_path: path for the sequence alignment in fasta format
    # g         : % of distinct site patterns required in a subsample
    # s         : Number of subsamples
    # r         : Number of upsamples
```

```

directory <- getwd()
SU_name <- SU_name          # subsample-upsample dataset name
setwd(directory)

if(g < 100){
  expected_dsp <- ceiling(num_of_dsp*(g/100))
  initial_sample <- sample(distinct_positions, expected_dsp, replace = F,
                           prob = NULL)
  expected_site <- ceiling((expected_dsp/length(unique(initial_sample)))
                          * expected_dsp)

  sln1 <- sample(1:sln, expected_site, replace = F, prob = NULL)
  sln2 <- sample(sln1, sln, replace = T)

}else{
  sln2 <- sample(1:sln, sln, replace = T)
}

#print(c("% Distinct site pattern sampled = ",
paste(signif((length(unique(sln1))/num_of_dsp)*100, 4),"%", sep ="")) )
SU_dataset <- endoapply(motherfile, function(x) x[sln2])

file_name <- paste(SU_name, sep = "")
Biostrings::writeXStringSet(SU_dataset, file_name)
}

#####
##### ModelTamer Aggregation #####
#####

MT_aggregator <- function(log_file_path, iqtrees_file_path, data_t =
                          c("DNA", "AA"), num_of_dsp, MT_time = NULL,
                          output_file = NULL){

  # log_file_path          : path for the log file
  # iqtrees_file_path      : % of distinct site patterns required in a
                           subsample
  # data_t                 : DNA or Amino Acid (AA)
  # num_of_dsp             : Number of distinct site patterns

#####log to Substitution matrix extraction #####
log2submat <- function(iqtrees_file){
  cline1 <- which (iqtrees_file == grep(pattern = "Rate matrix Q:",
                                       iqtrees_file, value = TRUE))
  iqtrees_file <- iqtrees_file[(cline1+2):(cline1+5)]
  iqtrees_file <- sapply(iqtrees_file, function(x){unlist(strsplit(x,
                                                                    "\\s+"))})}

  iqtrees_file <- data.frame(matrix(unlist(iqtrees_file), nrow=4,
                                    byrow=TRUE), stringsAsFactors=FALSE)
  rownames(iqtrees_file) <- iqtrees_file[,2]
  iqtrees_file <- iqtrees_file[,-c(1,2)]
  colnames(iqtrees_file) <- rownames(iqtrees_file)
  return(iqtrees_file)
}

#####

output <- list()

a_iqtrees <- readLines(iqtrees_file_path)

```

```

a_log    <- readLines(log_file_path)

cline1 <- which (a_igtree == grep(pattern = "Input data:", a_igtree, value
                                = TRUE))
cline2 <- which (a_log == grep(pattern = "Corrected Akaike Information
                                Criterion:", a_log, value = TRUE))

output[[1]] <- "\n"
output[[2]] <- "Best-fit model by Information Criteria (ModelTamer)"
output[[3]] <- "\n\n"
output[[4]] <- a_log[cline2+1]
output[[5]] <- "\n"
output[[6]] <- a_log[cline2-1]
output[[7]] <- "\n"
output[[8]] <- a_log[cline2]

output[[9]] <- "\n\n"
output[[10]] <- a_igtree[cline1]
output[[11]] <- "\n"
output[[12]] <- paste(a_igtree[cline1+3], "(reported by IQTREE)", sep=
                    " ")

output[[11]] <- "\n"
output[[12]] <- paste("Total distinct site patterns: ",
                    format(num_of_dsp, big.mark="," ,scientific=FALSE),
                    sep = "")

output[[13]] <- "\n"
output[[14]] <- paste("Distinct patterns used by ModelTamer:
                    ", sapply(a_igtree[cline1+4],
                    function(x){unlist(strsplit(x, "\\s+"))})[6],
                    sep = "")

output[[15]] <- "\n\n"

cline2 <- which (a_log == grep(pattern = "NOTE: ModelFinder requires",
                                a_log, value = TRUE))
a <- sapply(a_log[cline2], function(x){unlist(strsplit(x,
                    "\\s+"))})[4:6]
output[[16]] <- paste("Peak memory used by ModelTamer: ", a[1], a[2], a[3],
                    sep = " ")
output[[17]] <- "\n"

if(is.null(MT_time) == TRUE){
  cline2 <- which(a_log == grep(pattern = "CPU time for ModelFinder:",
                                a_log, value = TRUE))
  a <- sapply(a_log[cline2], function(x){unlist(strsplit(x,
                    "\\s+"))})[5:7]
  output[[18]] <- paste("CPU time for ModelTamer: ", a[1], a[2], a[3], sep
                    = " ")

  output[[19]] <- "\n"
  a <- sapply(a_log[cline2+1], function(x){unlist(strsplit(x,
                    "\\s+"))})[5:7]
  output[[20]] <- paste("Wall-clock time for ModelTamer: ", a[1], a[2],
                    a[3], sep = " ")
}else{
  formatted_time <- function(x){
    tolower(str_replace(str_replace(seconds_to_period(x), " ", ":"), " ",
    ":"))}
  output[[18]] <- paste("CPU time for ModelTamer: ", MT_time[1],
                    "seconds", "(", formatted_time(MT_time[1]), ")", sep = " ")
  output[[19]] <- "\n"
  output[[20]] <- paste("Wall-clock time for ModelTamer: ", MT_time[2],
                    "seconds", "(", formatted_time(MT_time[2]), ")", sep = " ")
}

```

```

if(is.null(output_file) == T){
  sink("MT_output.txt")
  cat(unlist(output))
  sink()
}else{
  final_output_name <- paste("MT_output_", output_file, ".txt", sep = "")
  sink(final_output_name)
  cat(unlist(output))
  sink()
}
}

#####
##### Main Function #####
#####

f_name <- data_path
motherfile <- Biostrings::readAAStringSet(f_name, format = "fasta")
sln <- as.numeric(fasta.seqlengths(f_name)[1])
a <- as.data.frame(t(as.matrix(motherfile)))

a2 <- unique(a)
a2$ID <- as.vector(as.numeric(rownames(unique(a))))
a2 <- merge(a, a2)

distinct_positions <- a2$ID
distinct_prob <- prop.table(distinct_positions)
num_of_dsp <- length(unique(distinct_positions))

rm(a)
rm(a2)

SU_name_base <- str_replace(basename(data_path), ".fasta", "")

if(data_type == "DNA"){
  g_est <- round((4594.2*num_of_dsp^(-1.043))*100, 1)
}else{
  g_est <- round(((4/20)*4594.2*num_of_dsp^(-1.043))*100, 1)
}

if(g_est >= 63){

  print("ModelTamer will analyze 63% of distinct site patterns")
  print("ModelTamer suggests to analyze full MSA")

  g_est <- 100
}else{
  g_est <- g_est
}

if(Redo == "FALSE"){

  SU_name1 <- paste("Example/",SU_name_base, "_g_", g_est, ".fasta", sep =
  "")
  MT_sampler_auto(motherfile, num_of_dsp = num_of_dsp, distinct_positions
  = distinct_positions,
  g = g_est, SU_name = SU_name1, sln = sln)
}

```

```

if(as.character(Sys.info()[1]) == "Windows"){
  ex_cmd <- paste("iqtree2 -s", SU_name1, "-m MF -nt 1 -quiet", sep = ""
    )
  system(ex_cmd)
}else{
  ex_cmd <- paste("./iqtree2 -s", SU_name1, "-m MF -nt 1 -quiet", sep = ""
    )
  system(ex_cmd)
}

MT_aggregator(paste(SU_name1, ".log", sep = ""), paste(SU_name1,
  ".iqtree", sep = ""), data_t = data_type, num_of_dsp =
  num_of_dsp, output_file = basename(data_path))

}else{
  MT_time <- c(0,0)
  g1 <- (1:max.iter)*g_est
  g1 <- c(g1[g1<63], 63, 100)
  if(length(g1)>max.iter){
    g1 <- g1[1:max.iter]
  }

  main_model <- NULL
  gamma_or_R <- NULL

  for(i in 1:length(g1)){
    SU_name1 <- paste( SU_name_base, "_g_", g1[i], ".fasta", sep = "")
    MT_sampler_auto(motherfile, num_of_dsp = num_of_dsp,
      distinct_positions = distinct_positions, g = g1[i],
      SU_name = SU_name1, sln = sln)

    if(as.character(Sys.info()[1]) == "Windows"){
      ex_cmd <- paste("iqtree -s",SU_name1, "-m MF -nt 1 -quiet",sep = ""
        )
      system(ex_cmd)
    }else{
      ex_cmd <-paste("./iqtree -s", SU_name1, "-m MF -nt 1 -quiet",sep = ""
        )
      system(ex_cmd)
    } # else
    temp <- readLines(paste(SU_name1, ".log", sep = ""))
    cline <- which (temp == grep(pattern = "CPU time for ModelFinder:",
      temp, value = TRUE))
    ct <- as.numeric(sapply(temp[cline], function(x){unlist(strsplit(x,
      "\\s+"))})[5])
    wt <- as.numeric(sapply(temp[cline+1], function(x){unlist(strsplit(x,
      "\\s+"))})[5])
    MT_time <- MT_time + c(ct, wt)

    count <- i
    cline <- which (temp == grep(pattern = "Bayesian Information
      Criterion:", temp, value = TRUE))
    a_log_model <- sapply(temp[cline], function(x){unlist(strsplit(x,
      "\\s+"))})[4]
    a_log_model <- unlist(Biostrings::strsplit(a_log_model, "+", fixed =
      T))
    print(a_log_model)
    a_log_model <- a_log_model[! a_log_model %in% c('ASC')]
    a_log_model_full <- a_log_model
  }
}

```

```

main_model[count] <- a_log_model[1]
a_log_model      <- a_log_model[-1]

g <- grep("G4", a_log_model)
R <- grep("R", a_log_model)
gr <- length(g)+length(R)
if(gr > 0){
  if(length(g) == 0){
    gamma_or_R[count] <- "R"
  }else{
    gamma_or_R[count] <- "G4"
  }

}

}else{
  gamma_or_R[count] <- "NA"
}

print(c(main_model[count], gamma_or_R[count]))   ### Remove later

convergence_score <- 0

if(count > 1){
  if(main_model[count-1] == main_model[count]){
    if(gamma_or_R[count -1] == gamma_or_R[count]){
      convergence_score <- 2
    }else{
      convergence_score <- 1
    }
  }else{
    convergence_score <- 0
  }
} #end count if

if(convergence_score > 1){
  break
}

} # for loop
MT_aggregator(paste(SU_name1, ".log", sep = ""), paste(SU_name1,
  ".iqtree", sep = ""), data_t = data_type, num_of_dsp =
  num_of_dsp, MT_time = MT_time , output_file =
  basename(data_path))

} # else

}

```