UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

SUPERVISED MACHINE LEARNING APPROACHES FOR PHYLOGENETIC ANALYSES

A DISSERTATION
SUBMITTED TO THE GRADUATE FACULTY
in partial fulfillment of the requirements for the
Degree of
DOCTOR OF PHILOSOPHY

by ULISES ROSAS PUCHURI

Norman, Oklahoma

2026

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/342633

SUPERVISED MACHINE LEARNING APPROACHES FOR PHYLOGENETIC ANALYSES

A DISSERTATION APPROVED FOR THE
SCHOOL OF BIOLOGICAL SCIENCES

BY THE COMMITTEE CONSISTING OF

Dr. Ingo Schlupp (Chair)

Dr. Ricardo Betancur (Co-Chair)

Dr. Abigail Moore

Dr. Daniel Becker

Dr. Sina Khanmohammadi

©Copyright by ULISES ROSAS PUCHURI 2026

All Rights Reserved.

Abstract

The following dissertation adds a new level of realism to multiple phylogenetic analyses by using supervised machine learning algorithms, which focus on prediction rather than data pattern discovery. Supervised machine learning has recently gained popularity for its application in large language models. However, applying these ideas to improve performance in phylogenetics remains underexplored. Through three chapters, the present dissertation showcases improvements across different phylogenetic analyses using supervised machine learning algorithms.

Chapter 2 addresses the scalability problem in likelihood-based phylogenetic network inference. As genomic datasets continue to grow, phylogenetic network inference faces serious scalability challenges, since the input size increases quartically with respect to the number of species. This chapter proposes a new subsampling algorithm tailored to large datasets, in which subsampling is guided by a sparse machine learning model. To the best of our knowledge, this is the first time a sparse non-parametric model has been used for data subsampling in phylogenetics. Previous approaches for phylogenetic trees relied on parametric models and required the subsample size to be fixed in advance. Chapter 2 shows that both assumptions can be relaxed, allowing the data to determine the optimal subsample size as a function of model complexity—restricted to at most four levels—without compromising estimation accuracy.

Chapter 3 tackles the parametric assumptions commonly made in phylogenetic regression. These assumptions break down, for example, when there are complex interactions among species traits or when the response function is more complex than a linear relationship. This chapter reformulates standard phylogenetic regression as kernel ridge regression, a non-parametric machine learning method, such that accommodates both correlation among observations and nonlinear effects. This approach is particularly useful in the subfield of

phylogenetic comparative methods, where the correlation structure—phylogenetic relatedness among species derived from a phylogenetic tree—is known. Through simulations and analyses of empirical datasets, this model outperforms standard phylogenetic regression. It also provides an initial framework for non-parametric regression with multispecies data while explicitly accounting for phylogenetic structure.

Chapter 4 explores factors influencing error in standard phylogenetic tree inference using deep learning. A deep learning model is trained to predict support for species trees based on a comprehensive set of features typically associated with methodological error. Deep learning outperforms simple linear models and enables model interrogation through feature importance analyses, revealing the dominant factors affecting inference in a given dataset. These insights allow for more informed choices regarding evolutionary models and data preprocessing strategies.

The intersection of phylogenetics and machine learning is an active area of research, and this dissertation contributes to this growing field. Throughout the dissertation, we demonstrate that machine learning methods can both improve phylogenetic analyses and provide insight into their behavior. Conversely, incorporating evolutionary information can enhance the performance of machine learning models when analyzing data from multiple species.

Acknowledgments

> The Secret of Life is always to set yourself attainable goals. They may be easy or difficult, depending on the circumstances and your character and abilities, but they should always be at-tai-na-ble!
>
> *Uncle Petros and Goldbach's Conjecture*

I would like to acknowledge my undergraduate advisor *i*) Ximena Vélez-Zuazo, for introducing me to the field of bioinformatics and encouraging me to pursue graduate studies; *ii*) Guillermo Ortí, for giving me my first opportunity to work in evolutionary biology at GWU; and *iii*) Ricardo Betancur-R., for welcoming me into his lab at OU and allowing me to work on what interests me most. I also thank my lab mates—Emanuell, Carmen, Aintzane, Melissa, and Fernando—who, through co-authored papers, conferences, and discussions, helped shape my view in academia.

This dissertation is dedicated to my mother, Doris, who always encouraged me to pursue higher education; to my aunt, Yrma, who gave me a second home and supported me throughout my undergraduate studies; and to my father, Ulises, who may still be confused about what I study but remains supportive and always open to dialogue. Finally, to Mehrnaz, for her boundless encouragement and love.

Table of Contents

List of Figures ix

List of Tables xi

1 Introduction 1
- 1.1 Machine learning 1
- 1.2 Phylogenetic inference 3
- 1.3 Phylogenetic comparative methods 6
- 1.4 Dissertation overview 8

2 Sparse learning for scalable phylogenetic network inference 12
- 2.1 Abstract 12
- 2.2 Introduction 13
- 2.3 Materials and Methods 17
  - 2.3.1 Regression Model Setup 17
  - 2.3.2 Random Phylogenetic Networks 18
  - 2.3.3 Elastic Net 19
  - 2.3.4 Importance Sampled Learning Ensemble (ISLE) 20
  - 2.3.5 Phylogenetic Network Inference 22
  - 2.3.6 Model Evaluation 25
- 2.4 Results 26
- 2.5 Discussion 36

3 Non-linear phylogenetic regression using regularized kernels 45
- 3.1 Abstract 45
- 3.2 Introduction 46
- 3.3 Material and Methods 49
  - 3.3.1 Implementation of phyloKRR 49
  - 3.3.2 Simulated datasets 53
  - 3.3.3 Empirical datasets 54
  - 3.3.4 Evaluation 55
- 3.4 Results 59
- 3.5 Discussion 66

4 Dissecting factors underlying phylogenetic uncertainty using machine learning models 77

4.1 Abstract . . . 77
4.2 Introduction . . . 78
4.3 Materials and methods . . . 83
4.3.1 Empirical Datasets . . . 83
4.3.2 Exon Capture, Assembly, and Quality Control . . . 84
4.3.3 Phylogenomic Inference . . . 85
4.3.4 Gene-Genealogy Interrogation (GGI) . . . 86
4.3.5 Machine Learning Models . . . 87
4.3.6 Feature Importance Analysis . . . 94
4.3.7 Simulated datasets . . . 95
4.3.8 Partial Dependence Plot (PDP) . . . 96
4.4 Results . . . 96
4.4.1 Dataset properties, Phylogenomic analyses, and GGI . . . 96
4.4.2 Machine Learning and Feature Importance . . . 98
4.5 Discussions . . . 105
4.5.1 Phylogenomic analyses . . . 105
4.5.2 Machine Learning and Feature Importance . . . 106
4.5.3 Conclusions . . . 110

A Supplementary material: Sparse learning for scalable phylogenetic network inference . . . 123

B Supplementary Material: Non-linear phylogenetic regression using regularized kernels . . . 153

C Supplementary Material: Dissecting factors underlying phylogenetic uncertainty using machine learning models . . . 172

List of Figures

1.1 Phylogenetic networks are a generalization of phylogenetic trees 4

1.2 Phylogenetic regression example 7

2.1 The Elastic Net path was directly applied to a CFs table with 15 rows, involving 6 species 27

2.2 Accuracy of reconstruction using subsampling guided by Elastic Net, i.e, Qsin algorithm without ensemble creation, on a CFs table with 15 rows 31

2.3 Hyperparameter space of ISLE ensemble with different level of non-linearity as measured by the number of leaf nodes in the regression trees 32

2.4 The Elastic Net path applied to the ISLE ensemble on a CFs table with 1,365 rows, involving 15 species 33

2.5 Accuracy of reconstruction using subsampling guided by ISLE, i.e., Qsin algorithm, on a CFs table with 1365 rows 34

2.6 Phylogenetic network reconstruction for the *Xiphophorus* dataset using a subsample of 763 rows (7.81% of the CFs table) and the two best log-pseudolikelihood scores 35

3.1 Schematic representation illustrating two possible scenarios that can occur when weighting data points with the modeled covariance matrix to account for phylogenetic non-independence and heteroscedasticity in trait data 48

3.2 Empirical datasets before and after weighting data points using the modeled covariance matrix assuming Brownian motion 56

3.3 Performance of phyloKRR and PGLS under varying degrees of non-linearity in simulated datasets 61

3.4 Performance of phylokRR and PGLS when $b = 0$ and $b = 1$ according to Eq. 3.5 62

3.5 Error difference between phyloKRR (using RBF kernel and linear kernel) and PGLS under varying degrees of non-linearity in simulated datasets 62

3.6 Model inspection analyses using phyloKRR with the RBF kernel on simulated data generated from Eq. 3.6 63

3.7 Hyperparameter space of phyloKRR with RBF and linear kernel in both tested empirical datasets 64

3.8 Error distribution after 20 rounds of cross-validations for all models in both empirical datasets 65

4.1 Phylogenetic trees obtained for Protacantopterygii (**A**) and Carangaria (**B**) . 86

4.2 The pipeline for finding the best fit model and its feature importance 94

4.3 Two major phylogenetic trees estimated with the Protacanthopterygii dataset 100

4.4 Testing MSE distribution from 20 rounds of CV for each model, and pairwise comparison of model testing MSE based on Wilcox Rank Sum Test 101

4.5 Feature importance profiles for DNN models measured by the average increase in testing MSE for the empirical datasets 102

4.6 Feature importance profiles for DNN models measured by the average increase in testing MSE for the simulated datasets 103

4.7 2D Partial dependence plots for both datasets using the DNN model 104

List of Tables

4.1 List of features used based on alignment and tree information for training machine learning models 91

Chapter 1

Introduction

1.1 Machine learning

Machine learning is generally referred to as a suite of statistical models focused on prediction and data pattern discovery, given an available dataset (Hastie et al., 2009). It is commonly divided into two broad classes: supervised learning (i.e., a response variable exists) and unsupervised learning (i.e., a response variable does not exist). This dissertation focuses on the supervised case, which is concerned with prediction and has gained significant popularity in recent years due to its success at large scales (e.g., large language models and genomic language models) (Vaswani et al., 2017; Ji et al., 2021).

While many architectures and modeling techniques exist, in the case of a continuous response variable, the simplest formulation for predicting n responses are obtained by minimizing the mean squared error (Hastie et al., 2009):

$$\underset{\mu}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n}\left(\mu(\mathbf{x}_i) - y_i\right)^2 \tag{1.1}$$

Where $\mathbf{x}_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$ denote the predictor vector and response variable of the i^{th} observation, and the function μ is the best average estimator of y given $\mathbf{x}$. The form of the μ depend on the type of machine learning model considered. If μ is specified as a weighted

sum of the predictors, with $\beta \in \mathbb{R}^{p+1}$ denoting the vector of weights:

$$\mu(\mathbf{x}_i) = \beta_0 + \sum_{i=1}^{p} x_{i,j}\beta_j, \tag{1.2}$$

Then we call μ linear estimator, and the resulting problem is known as linear regression. However, the assumption that the response variable is a linear function (i.e., a weighted sum) of the predictors is often violated in practice. Modern machine learning techniques challenge this restriction in order to capture more realistic data-generating mechanisms.

For instance, consider a response variable whose optimal estimator μ takes the following form (Friedman and Popescu, 2008, Eq. 27) in 35 dimensions:

$$\mu(\mathbf{x}_i) = 10 \prod_{j=1}^{5} e^{-2x_{i,j}^2} + \sum_{j=6}^{35} x_{i,j}.$$

In this example, the estimator is not merely a weighted sum of predictors (as in the second term, where the weights are equal to one), but also includes nonlinear interactions among predictors through the product term in the first component. Approximating such a response using a purely linear model would lead to an underfitting scenario. In contrast, modern machine learning methods adopt a data-driven approach, allowing the data to determine the most appropriate form of μ, including linear solutions as a special case. This leads to a nonparametric formulation for μ, and thus a broader generalization. This perspective is adopted throughout all chapters of this dissertation.

Another generalization of the weighted-sum framework is achieved by introducing a penalization term into the optimization problem. For simplicity, consider the case of linear regression:

$$\underset{\beta}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n} \left(\mu(\mathbf{x}_i;\beta) - y_i\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{1.3}$$

Here, λ denotes the penalization strength. This penalty constrains the behavior of the

coefficients β_j and has the notable property of shrinking some of them exactly to zero (Hastie et al., 2015). Chapter 2 leverages this property to enable more efficient network phylogenetic inference. A nonparametric form of μ is also considered in that context.

1.2 Phylogenetic inference

Phylogenetic trees and networks are graph-based representations of the evolutionary history of species over millions of years (Felsenstein, 2004; Kong et al., 2025). Advances in data collection, particularly the availability of large-scale genomic data, have revealed that species evolution often involves increasingly complex evolutionary histories. For instance, in contrast to phylogenetic trees, phylogenetic networks allow nodes representing ancestral species to have more than one parent, thereby modeling signals of hybridization, introgression, and other reticulation events (Fig. 1.1) (Solís-Lemus and Ané, 2016).

However, the rapid increase in data volume has posed significant challenges for standard inference methods, particularly with respect to scalability. While tree-based inference can be scaled to approximately 10,000 species using methods such as ASTRAL or RAxML (Zhang et al., 2018; Stamatakis, 2014), and even to hundreds of thousands of species through divide-and-conquer strategies (Balaban et al., 2024), inference for phylogenetic networks is, in practical terms, limited to at most about 50 species. This dissertation addresses this gap in inference methodology for phylogenetic networks, with a particular focus on modifying likelihood-based estimation procedures to enable more effective searches for optimal network topologies.

Searching for an optimal tree—and, by extension, an optimal network—for a given set of species is a notoriously hard problem (Roch, 2006). For example, for 50 species, the total number of possible trees is comparable to the estimated number of atoms in the universe (Felsenstein, 2004). As a result, inference methods must rely on heuristics, most commonly in local search-based algorithms.

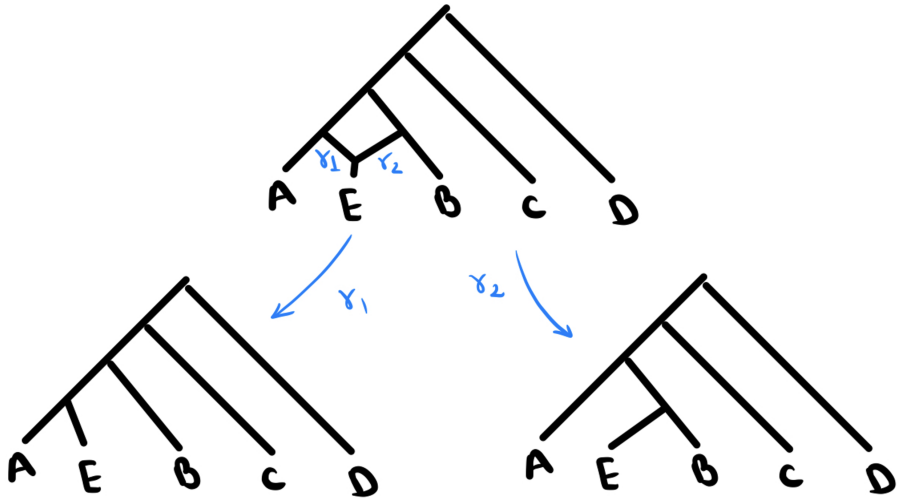


Figure 1.1: Phylogenetic networks are a generalization of phylogenetic trees. For instance, from the network at the top, we can see that it can be decomposed into two trees, depending on which parent of the reticulation node is chosen, each with inheritance probabilities γ_1 and γ_2, and represented in blue. Although the two trees shown below are admissible, they only provide a partial view of species evolution.

Many heuristic strategies have been proposed, but the most successful ones optimize a well-defined optimality criterion, such as maximizing the likelihood or the posterior probability. Among the methods that balance accuracy and scalability are likelihood-based approaches. These methods can also serve as inputs for other likelihood-based procedures (e.g., maximum pseudolikelihood inference for phylogenetic networks) or even for alternative optimality criteria, such as the maximum quartet score used by ASTRAL.

Although likelihood-based inference for trees and networks differs in implementation, both approaches share a common heuristic framework (Algorithm 1). In both cases, the objective is to maximize the log-likelihood of the data given a proposed topology $\mathcal{T}$ across m observations:

$$\underset{\mathcal{T},\theta}{\text{maximize}} \sum_{i=1}^{m} \log f(\mathbf{x}_i|\theta) \ , \tag{1.4}$$

Where f denotes the likelihood function for the ith observation $\mathbf{x}_i$ given a set of parameters θ (e.g., branch lengths, substitution rates, and stationary probabilities).

In the case of phylogenetic trees, $\mathbf{x}_i$ typically corresponds to the ith column of a multiple sequence alignment (MSA) with m columns, and the likelihood function f is computed using Felsenstein's pruning algorithm, which recursively evaluate conditional probabilities from leaf nodes to the root, or artificial root, of the tree. For phylogenetic networks, $\mathbf{x}_i$ can represent ith row in the quartet concordance factor table. This table has three columns, representing the observed proportions of all possible unrooted trees for a given set of four species. In this setting, f is obtained by first extracting all possible quartets of four species from the network and then computing the expected concordance factors under the network multispecies coalescent model (Solís-Lemus and Ané, 2016, see their Supplementary Information).

While in phylogenetic trees the number of observations m is typically independent of the number of species, in phylogenetic networks there is an explicit dependence on the number of species. For n species, the total number of rows in the concordance factors is:

$$m = \binom{n}{4}.$$

This causes the computational burden to grow on the order of the fourth power of n. Chapter 1 proposes a strategy for subsetting this collection of quartets without compromising the maximization of Eq. 1.4. This subsetting procedure is guided by a machine learning model based on Eq. 1.3 that nonparametrically approximates the likelihood objective.

In Line 6 of Algorithm 1 for phylogenetic trees, the parameters θ consist of substitution rates and branch lengths, and the data typically comprise alignment sites. Chapter 4 explores the methodological factors that affect this step and the optimization problem in Eq. 1.4, as well as the factors that influence the quality of inferred topologies by correlating these factors with the level of support for a given topology. This correlation is assessed using a deep learning–based machine learning model.

Algorithm 1 Local search for phylogenetic inference

Input: A set of m observations $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ (e.g., m columns in a MSA).
Output: A phylogenetic topology, including parameters θ, that maximizes the likelihood.
Initialize $\mathcal{T}^{curr}$, θ^{curr}.
repeat
    Let $\mathcal{T}^{new}$ be a local move (e.g., nearest neighboor interchange) from $\mathcal{T}^{curr}$.
    Numerically optimize parameters θ^{new} given the evolutionary model.
    Calculate new likelihood using Eq. 1.4.
    if Likelihood increased **then**

$$\mathcal{T}^{curr} \leftarrow \mathcal{T}^{new}$$
$$\theta^{curr} \leftarrow \theta^{new}$$

    end if
until Stopping criterion is met

1.3 Phylogenetic comparative methods

Once we have a phylogenetic tree or network, we obtain a hypothesis of species evolution. This hypothesis can be used as a prior for multiple downstream analyses, including regression analyses, reconstruction of ancestral traits from present-day observations, and diversification analyses, among others. This subfield is known as phylogenetic comparative methods and is a particularly fruitful area of research in evolutionary biology (Harmon, 2019). Chapter 3 focuses on the case of phylogenetic regression (Felsenstein, 1985).

In phylogenetic regression, as in any other regression framework, we aim to minimize an error criterion, as shown in Eq. 1.1. However, Eq. 1.1—and, more generally, many machine learning models—assumes that observations across species are independent. For example, in image classification, models capture complex relationships among pixels within an image but typically assume independence across images. In comparative biology, however, such independence assumptions are inappropriate. If images encode limb morphology in vertebrates, they are not independent: there is a phylogenetic effect. The limbs of whales are more similar to those of other marine mammals than to those of rodents. Consequently,

using a weighted sum as in Eq. 1.2, phylogenetic regression weights each error term according to the phylogenetic relationships among species. This leads to a model known as Phylogenetic Generalized Least Squares (PGLS; Grafen, 1989) (Fig. 1.2).

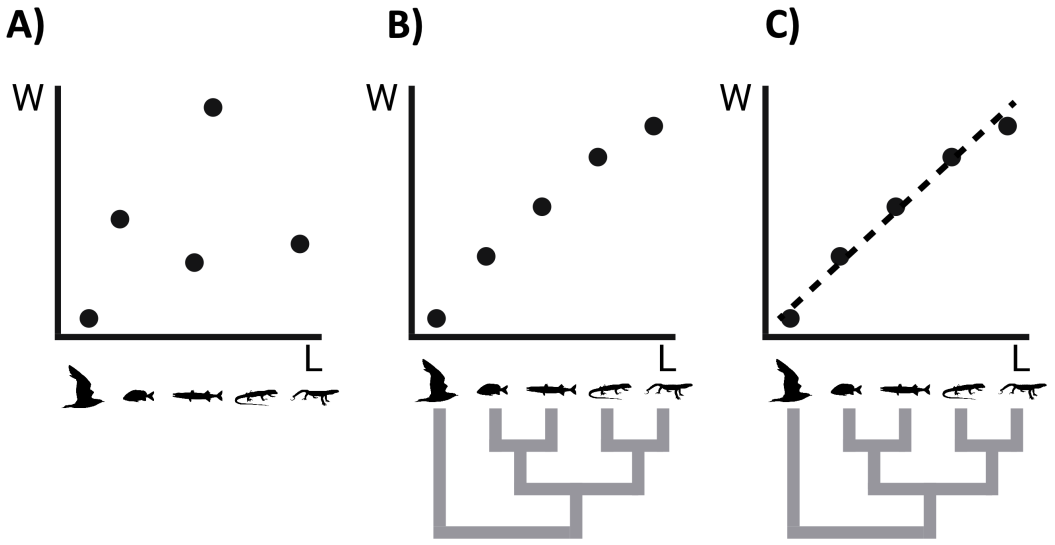


Figure 1.2: Phylogenetic regression example. **A)** We illustrate a phylogenetic regression problem involving multiple species, including lizards and fishes, where the predictor is body weight, and the response variable is body length. **B)** Because growth patterns are more strongly correlated within lizards and within fishes than between these two groups, we incorporate a phylogenetic effect, represented by the phylogeny along the x-axis. **C)** Accounting for phylogenetic effects reduces within-group variance and it is generally assumed that allows a linear fit. This dissertation challenges that assumption.

However, more complex relationships beyond a weighted-sum estimator for μ do not exist in this framework. This limitation is precisely what Chapter 3 explores through the use of kernels (Bishop and Nasrabadi, 2006, pp. 291). Specifically, via a mapping function ϕ, we project each predictor vector $\mathbf{x}_i$ into a high (possibly infinite) dimensional feature space, which includes the original predictors as well as all possible interactions among them:

$$\phi(\mathbf{x}_i) = \begin{bmatrix} 1 & x_{i,1} & x_{i,2} & \dots & x_{i,1}^2 & x_{i,2}^2 & \dots & x_{i,1}x_{i,2} & \dots \end{bmatrix} .$$

Under this representation, the estimator μ takes the form:

$$\mu(\mathbf{x}_i) = \lim_{p \to \infty} \sum_{j=1}^{p} \phi(\mathbf{x}_i)_j \, \beta_j.$$

The feature vector $\phi(\mathbf{x}_i)$ does not live in standard Euclidean space, but rather in a Reproducing Kernel Hilbert Space (RKHS, Wahba and Wang, 2019). Although this representation is infinite-dimensional, the problem can be solved efficiently using kernel methods. When the resulting solution is mapped back into the original Euclidean space, it yields a non-parametric estimator.

Chapter 3 presents the first attempt to incorporate a phylogenetic prior into kernel-based regression, thereby providing the first nonparametric formulation of phylogenetic regression. Many improvements are possible within this framework, including the treatment of non-continuous response variables (e.g., binary response variables) through the kernelization of other standard comparative phylogenetic methods, such as Generalized Estimating Equations (GEE, Paradis and Claude, 2002).

1.4 Dissertation overview

This dissertation introduces a new level of realism to phylogenetic analyses by leveraging supervised machine learning algorithms that emphasize prediction rather than pattern discovery. Chapter 2 proposes an algorithm to improve the efficiency of phylogenetic network inference by reducing the size of large-scale datasets via sparse ensemble learning. Chapter 3 introduces a kernel-based regression framework that accounts for phylogenetic structure, allowing a continuous response variable to exhibit arbitrarily complex relationships with predictor variables of taxa. Chapter 4 presents a general deep learning framework for nonlinearly exploring factors that influence phylogenetic uncertainty. The intersection of phylogenetics and machine learning is an active area of research, and this dissertation contributes to this growing field. Throughout, we demonstrate that machine learning methods

can both enhance phylogenetic analyses and provide insight into their behavior.

Reference List

Balaban, M., Y. Jiang, Q. Zhu, D. McDonald, R. Knight, and S. Mirarab, 2024: Generation of accurate, expandable phylogenomic trees with udance. *Nature biotechnology*, **42 (5)**, 768–777.

Bishop, C. M., and N. M. Nasrabadi, 2006: *Pattern recognition and machine learning*, Vol. 4. Springer.

Felsenstein, J., 1985: Phylogenies and the comparative method. *The American Naturalist*, **125 (1)**, 1–15.

Felsenstein, J., 2004: *Inferring Phylogenies*. Sinauer Associates, Sunderland, MA.

Friedman, J. H., and B. E. Popescu, 2008: Predictive learning via rule ensembles.

Grafen, A., 1989: The phylogenetic regression. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, **326 (1233)**, 119–157.

Harmon, L., 2019: *Phylogenetic comparative methods: learning from trees*. EcoEvoRxiv.

Hastie, T., R. Tibshirani, and J. Friedman, 2009: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed., Springer Series in Statistics, Springer, New York.

Hastie, T., R. Tibshirani, and M. Wainwright, 2015: *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, Boca Raton, FL.

Ji, Y., Z. Zhou, H. Liu, and R. V. Davuluri, 2021: Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, **37 (15)**, 2112–2120.

Kong, S., C. Solís-Lemus, and G. P. Tiley, 2025: Phylogenetic networks empower biodiversity research. *Proceedings of the National Academy of Sciences*, **122 (31)**, e2410934 122.

Roch, S., 2006: A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. *IEEE/ACM transactions on computational biology and bioinformatics*, **3 (1)**, 92–94.

Solís-Lemus, C., and C. Ané, 2016: Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. *PLoS genetics*, **12 (3)**, e1005 896.

Stamatakis, A., 2014: Raxml version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, **30 (9)**, 1312–1313.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, 2017: Attention is all you need. *Advances in neural information processing systems*, **30**.

Zhang, C., M. Rabiee, E. Sayyari, and S. Mirarab, 2018: Astral-iii: polynomial time species tree reconstruction from partially resolved gene trees. *BMC bioinformatics*, **19 (6)**, 15–30.

Chapter 2

Sparse learning for scalable phylogenetic network inference

Under revision in *Systematic Biology*.

Ulises Rosas-Puchuri, Nathan Kolbow, Claudia Solís-Lemus, Sina Khanmohammadi, Ricardo Betancur-R.

2.1 Abstract

Phylogenetic networks account for signals of hybridization, reticulation, and gene flow, and provide an opportunity to analyse species evolution from a more complex perspective than bifurcating phylogenetic trees. However, even the fastest algorithms for inferring these networks face scalability challenges as the number of species increases. This limitation arises because these methods use as input a large concordance factors (CFs) table, which summarizes the observed CFs of all possible four-species combinations in each row. The size of this table scales with the fourth power of the number of species, creating computational bottlenecks and highlighting the need for more efficient solutions. Sparse learning has been shown to reduce the dimensionality of large-scale datasets while producing results of comparable quality to those obtained using the full dataset. In this study, we adapted two sparse machine learning models—Elastic Net and Ensemble Learning + Elastic Net—to guide the subsample of an optimal number of rows from the CFs table required to accurately predict

the overall phylogenetic network pseudolikelihood. Both methods account for the inherent correlation among rows, which arises because rows overlap in species information. We call this method *Qsin*. In two simulated datasets, Qsin reduced the dataset by approximately half without compromising accuracy. For the *Xiphophorus* fishes dataset, which contains 10,626 rows in the CFs table, we recovered the same topology as with the full CFs table but using only 763 rows. Using these subsamples also reduced running times by up to 60% without compromising accuracy. These gains are expected to persist as species numbers increase. Qsin contributes to ongoing efforts to make phylogenetic network inference more efficient and opens the door to analyses of more complex evolutionary histories. The source code for Qsin is freely available at: `https://github.com/ulises-rosas/qsin`.

2.2 Introduction

Phylogenetic networks are a generalization of phylogenetic trees (Kong et al., 2025). They model species evolution by accounting for signals from hybridization, gene flow, and reticulation (Solís-Lemus and Ané, 2016; Blischak et al., 2023). For instance, consider a phylogenetic network with a single hybridization event, resulting in two hybrid edges converging at a node. This scenario can produce two distinct underlying tree histories of species evolution, depending on which hybrid edge is followed. While both trees represent valid evolutionary histories, each may offer only a partial view of the species evolution. Phylogenetic networks, therefore, provide a more comprehensive representation of complex evolutionary histories.

One of the fastest algorithms for inferring phylogenetic networks is called SNaQ (Solís-Lemus and Ané, 2016). Currently, SNaQ models networks in which the sets of edges from hybridization cycles are all disjoint—known as level-1 networks. Among the reasons this method gained popularity includes that it i) accounts for incomplete lineage sorting (ILS), ii) models hybridization in small to medium-sized sets of taxa, and iii) does not rely on branch length information from gene trees, which can be sensitive to estimation error (Frankel and

Ané, 2023). To reconstruct a phylogenetic network using SNaQ we typically start from the following table of concordance factors (CFs hereafter):

4-taxon set	CF_1	CF_2	CF_3
r_1	$\mathrm{f}_{1,1}$	$\mathrm{f}_{1,2}$	$\mathrm{f}_{1,3}$
r_2	$\mathrm{f}_{2,1}$	$\mathrm{f}_{2,2}$	$\mathrm{f}_{3,3}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_p	$\mathrm{f}_{p,1}$	$\mathrm{f}_{p,2}$	$\mathrm{f}_{p,3}$

In the table above, let r_j denote a subset of four species in row j of the CFs table, obtained from the full set of species. Let CF_k represent topology k for this set of four species, also called a quartet topology. For example, for species A, B, C, and D, the three possible quartets are $AB|CD$, $AC|BD$, and $AD|BC$, where | indicates the edge separating sister species. Finally, $\mathrm{f}_{j,k}$ is the proportion of gene trees displaying quartet topology k for taxa in r_j, hence the name concordance factors.

Given a proposed phylogenetic network topology N_i, we calculate its log-pseudolikelihood score $\tilde{\ell}(N_i)$ as:

$$\tilde{\ell}(N_i) = \sum_{j=1}^{p} n_{j,1} \ln \hat{\mathrm{f}}_{j,1} + n_{j,2} \ln \hat{\mathrm{f}}_{j,2} + n_{j,3} \ln \hat{\mathrm{f}}_{j,3} \ , \tag{2.1}$$

Where $\hat{\mathrm{f}}_{j,k}$ is the expected value of $\mathrm{f}_{j,k}$ obtained after branch length optimization, and $n_{j,k}$ is the number of times quartet topology k for taxa in r_j appears among the observed gene trees. This score assumes that $\mathrm{f}_{j,k}$ follows a multinomial distribution and is used to search the network space: if the score increases for a new network topology, the topology is accepted; otherwise, the algorithm continues searching (Solís-Lemus and Ané, 2016). However, as the number of species increases, the SNaQ algorithm faces scalability challenges. This limitation stems from the quartic growth of the number of CFs rows with respect to the number of species. For example, with 50 species, SNaQ must evaluate the log-pseudolikelihood of all four-taxon sets, requiring the multiple assessments of 230,300 CFs rows per topology pro-

posal. Moreover, as the number of species increases, the topology space expands rapidly—at least as fast as the tree space—further compounding the computational burden.

The challenge of working with large datasets is not exclusive to phylogenetics; it arises in many other fields as well (Yao and Wang, 2021). Typically, two scenarios are encountered: there are far more observations than variables (denoted as $n \gg p$), or there are far more variables than observations (denoted as $p \gg n$). One common approach to address such problems is through regression methods that impose sparsity constraints on the model weights—that is, forcing some weights to be exactly zero (Guyon and Elisseeff, 2003). For instance, to reconstruct a phylogenetic tree from a large DNA sequence alignment, Ecker et al. (2022) employed Lasso regression (Tibshirani, 1996). In their study, the variables were the site loglikelihoods calculated from random trees and the observations corresponded to the number of random trees used. Using Lasso regression, many variable weights were shrunk to exactly zero, and the authors used the non-zero weights, representing relevant sites to the overall loglikelihood, to reconstruct the tree. When the runtime of Lasso regression is lower than that of tree reconstruction, this approach is convenient because site loglikelihoods evaluations from random trees are inexpensive to compute: there is no need to optimize branch lengths, and one does not need to wait for the evaluation of a tree to finish before proposing the next one.

The above example shows how sparse regressions can effectively reframe the dimensionality reduction challenge posed by large datasets as a feature-selection problem (Lemhadri et al., 2021; Singh and Stufken, 2023). Intuitively, we could adapt a similar approach to improve the scalability of SNaQ. Rather than using alignment site log-likelihoods under random trees, we would treat the individual loglikelihoods of CF table rows, computed under random phylogenetic networks, as predictor variables. Correspondingly, we would use random network log-pseudolikelihood scores as the response variable. However, there are two key considerations we need to address: i) unlike sites in a sequence alignment, the rows in the CF table are not independent, and ii) in a high-dimensional settings (i.e., $p \gg n$), a unique

Lasso solution selects at most n features, where n is the number of observations (Rosset and Zhu, 2007; Hastie et al., 2015; Tibshirani and Wasserman, 2017), such as the chosen number of simulations in our case.

To address these issues, we propose the use of Elastic Net (Zou and Hastie, 2005), a sparse learning method that produces sparse solutions—also referred to as sparse learning. Elastic Net balances sparsity with the inclusion of correlated variables, enabling the selection of more than n non-zero coefficients when $p \gg n$, by leveraging a grouping effect among correlated variables. Furthermore, we account for complex interactions among variables by modeling the log-pseudolikelihood score with a sparse ensemble of regression trees, also known as Importance Sampled Learning Ensemble (ISLE) (Friedman et al., 2003). Depending on the ensemble complexity, ISLE also has the added advantage of rapid testing of error convergence toward a specific sparse solution, allowing data-driven determination of the optimal percentage of row subsampling.

Here, we (1) propose an algorithm that reduces the number of rows in the CF table, guided by Elastic Net-based machine learning; (2) apply cross-validation (Hastie et al., 2009) to estimate all major hyperparameters of the machine learning models, including ISLE ensemble parameters; (3) analyze the time complexity of the algorithm as a function of the number of taxa; and (4) evaluate the algorithm on simulated data and on the best-characterized empirical network dataset for *Xiphophorus* fishes (Cui et al., 2013; Solís-Lemus and Ané, 2016). Our primary goal is to improve the runtime of phylogenetic network inference without compromising accuracy.

2.3 Materials and Methods

2.3.1 Regression Model Setup

Given a phylogenetic network N_i, we define the quartet log-likelihood for row r_j in the CFs table as

$$\ell(r_j \mid N_i) := n_{j,1} \ln \hat{\mathrm{f}}_{j,1} + n_{j,2} \ln \hat{\mathrm{f}}_{j,2} + n_{j,3} \ln \hat{\mathrm{f}}_{j,3} \ ,$$

such that Eq. 2.1, which denotes the log-pseudolikelihood score, can be rewritten as

$$\tilde{\ell}(N_i) = \sum_{j=1}^{p} \ell(r_j \mid N_i).$$

Similarly to Ecker et al. (2022), we then assume the existence of a scalar $\beta_0 \in \mathbb{R}$ and a vector $\beta \in \mathbb{R}^p$ such that the calculation of $\tilde{\ell}(N_i)$ can be approximated by:

$$\begin{aligned} \tilde{\ell}(N_i) &\approx \sum_{j=1}^{p} \ell(r_j \mid N_i)\beta_j + \beta_0 + \varepsilon_i \\ &= \begin{bmatrix} \ell(r_1 \mid N_i) & \cdots & \ell(r_p \mid N_i) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \beta_0 + \varepsilon_i, \end{aligned}$$

Where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ is the error of the approximation with mean 0 and a constant variance σ^2. For n phylogenetic networks, we obtain the following regression formulation:

$$\underbrace{\begin{bmatrix} \tilde{\ell}(N_1) \\ \tilde{\ell}(N_2) \\ \vdots \\ \tilde{\ell}(N_n) \end{bmatrix}}_{\mathbf{y}} \approx \underbrace{\begin{bmatrix} \ell(r_1 \mid N_1) & \cdots & \ell(r_p \mid N_1) \\ \ell(r_1 \mid N_2) & \cdots & \ell(r_p \mid N_2) \\ \vdots & \ddots & \vdots \\ \ell(r_1 \mid N_n) & \cdots & \ell(r_p \mid N_n) \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}}_{\beta} + \underbrace{\begin{bmatrix} \beta_0 \\ \beta_0 \\ \vdots \\ \beta_0 \end{bmatrix}}_{\mathbf{1}\beta_0} + \varepsilon. \tag{2.2}$$

We define the left-hand-side response vector as $\mathbf{y} \in \mathbb{R}^{n\times 1}$, the predictor matrix as $\mathbf{X} \in \mathbb{R}^{n\times p}$, and the right-hand-side vector as the intercept vector $\mathbf{1}\beta_0 \in \mathbb{R}^{n\times 1}$. A total of n phylogenetic networks can be obtained from simulations (see Section 2.3.2). Notice that there is a one-to-one correspondence between the columns of $\mathbf{X}$ and the rows of the CFs table. Specifically, column j of $\mathbf{X}$ represents the quartet log-likelihood of row j in the CFs table, given a set of n random phylogenetic networks. The coefficient vector β assigns weights to these columns.

If the weight for the column j is set to zero (i.e., a sparse solution), then it makes no contribution to the overall prediction. Because of the direct correspondence between column j in $\mathbf{X}$ and row j in the CFs table, this implies that row j in the CFs table is also negligible and can be removed. Thus, the input size for SNaQ can be reduced in proportion to the sparsity of β. Using sparse learning techniques, as described in Sections 2.3.3 and 2.3.4, we obtain sparse versions of β.

2.3.2 Random Phylogenetic Networks

We conducted random phylogenetic network simulations using SiPhyNetworks (Justison et al., 2023). Following Kolbow et al. (2025), we used a speciation rate of 1, an extinction rate of 0.2, and drew inheritance probabilities from a Beta(10, 10) distribution. We also used a hybridization rate of $2/\binom{T}{2}$, where T is the number of taxa. This hybridization rate causes hybridization events to occur more frequently toward the end of the process, when more lineages are available to choose from. As a result, more level-1 networks are generated, since the likelihood of hybridization events forming independent hybridization cycles increases. To accelerate lineage generation, we enabled only the lineage-generative mode. To ensure at least n level-1 networks, we simulated up to $4n$ networks and retained only those that satisfied the level-1 constraint. After filtering, we randomly permuted the tip labels and rescaled the branch lengths so that the average branch length was 1.25, approximately reflecting a moderate level of ILS (Kolbow et al., 2025).

Section 2.3.6 describes the simulated and empirical CFs tables used here. Following Eq. 2.1, we computed the quartet log-likelihood for each row in the CFs table across all n random phylogenetic networks generated using `topologyQPseudolik` in PhyloNetworks v0.16.4 (Solís-Lemus et al., 2017). In this study, we set $n = 1000$. Notably, this step is highly parallelizable, as each simulation—including the log-pseudolikelihood calculation—is independent.

2.3.3 Elastic Net

From the above data at Eq. 2.2 definition and after data centering, which allow us to omit the intercept term (Wang, 2022), we can define the following Elastic Net objective function:

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \frac{1}{n}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \left\{ \frac{1}{2}(1-\alpha)\|\beta\|_2^2 + \alpha\|\beta\|_1 \right\},$$

where λ is the sparsity parameter, and α is the weight between the ℓ_1-norm and ℓ_2-norm penalizations. The Elastic Net model is a generalization of the lasso model, and this becomes more apparent if we set $\alpha = 1$. The addition of the l_2-norm serves two purposes: it provides a grouping effect for correlated predictors, and it allows the selection of more than n solutions, a limitation of the lasso model (Rosset and Zhu, 2007). Hyperparameters α and λ are priors of the Elastic Net model and are usually obtained via cross-validation (CV) (Hastie et al., 2009, pp. 241).

Let $\Lambda \in \mathbb{R}^k$ denote an ordered set of k values of λ arranged in decreasing order, and define the Elastic Net path as the set of k solutions corresponding to each $\lambda \in \Lambda$. Computing the full Elastic Net path is more computationally efficient than testing randomly selected λ values from the middle of the Λ range. This efficiency arises from the warm-start strategy: the solution obtained for a previous value, say λ_{k-1}, serves as a good initial estimate for the model with the next value, λ_k.

While the value of α is often pre-chosen on qualitative grounds (Hastie et al., 2009;

Friedman et al., 2010), in this study we followed the 5-fold CV strategy proposed by Zou and Hastie (2005): from a relatively small set of candidate α values, each is used to compute the full Elastic Net path across all CV folds. The α that yields the lowest average CV error along its path is then selected.

The set we chose for α is $\{0.5, 0.95, 0.99\}$, and for the values of Λ, we let $k = 100$, where all the values are equally spaced on a logarithmic scale. The maximum value from this set, $\lambda_{\max} \in \Lambda$, is the one that produces a fully sparse solution, and it can be theoretically obtained from data (Hastie et al., 2015, pp. 114). The minimum value from this set can be obtained as $\lambda_{\min} = \lambda_{\max} \cdot \epsilon$, where ϵ is a small constant. In this study, we set $\epsilon = 0.0001$.

2.3.4 Importance Sampled Learning Ensemble (ISLE)

Applying Elastic Net directly to $\mathbf{X}$ constrains the prediction of $\mathbf{y}$ to a linear function of the predictor variables, which can lead to underfitting. To address this limitation, we relax the linearity assumption, allowing arbitrarily nonlinear functions of the predictor variables in predicting $\mathbf{y}$. This, in turn, allows non-parametric interactions among subsets of rows from the CFs table to predict the log-pseudolikelihood score of a phylogenetic network, while still retaining the sparsity benefits of Elastic Net. We achieve this using the ISLE algorithm (Friedman et al., 2003; Seni and Elder, 2010; Zeng, 2022), which consists of two main steps: (i) generating a strong ensemble of nonlinear basis functions, and (ii) post-processing of these functions with a sparse regression method such as Elastic Net to select a subset with optimal predictive performance. In this study, we use regression trees as the nonlinear basis functions.

Let the prediction of n log-pseudolikelihood scores from the matrix $\mathbf{X}$, as defined in Eq. 2.2, using a regression tree be denoted as $\mathcal{T}(\mathbf{X}; \gamma_j) \in \mathbb{R}^{n \times 1}$, where γ_j is the optimal set of parameters in the regression tree (e.g., splitting variables at internal nodes, tree depth). Then, we can form the following matrix $\mathbf{T} \in \mathbb{R}^{n \times M}$ containing the predictions of n log-

pseudolikelihood scores from M regression trees:

$$\mathbf{T} = \begin{bmatrix} | & & | \\ \mathcal{T}(\mathbf{X}; \gamma_1) & \cdots & \mathcal{T}(\mathbf{X}; \gamma_M) \\ | & & | \end{bmatrix} .$$

Let $\mathbf{w} \in \mathbb{R}^M$ be a vector containing a weight for the prediction performance of each regression tree; then, by making this vector $\mathbf{w}$ sparse using Elastic Net on the centered data, we can create a subsample of these regression trees:

$$\underset{\mathbf{w} \in \mathbb{R}^M}{\text{minimize}} \frac{1}{n} \|\mathbf{y} - \mathbf{T}\mathbf{w}\|_2^2 + \lambda \left\{ \frac{1}{2}(1-\alpha)\|\mathbf{w}\|_2^2 + \alpha \|\mathbf{w}\|_1 \right\} . \tag{2.3}$$

However, if all regression trees are generated randomly, we may undersample important variables—in our case, informative rows of the CFs. Then, the ISLE algorithm also proposes leveraging an importance sampling approach to obtain a stronger ensemble (Seni and Elder, 2010, pp. 43). The details of the algorithm in the specific case of regression trees can be found in Algorithm 2. $S_m(\eta)$ represents the m subsample of $n \cdot \eta$ observations, where $\eta \in (0, 1]$ is a constant. $\nu \in [0, 1]$ is another constant that represent the memory rate. Intuitively, if we decrease η, then we increase the randomness of the regression trees; and if we increase ν, then the regression trees avoid similarity between them. Friedman et al. (2003) recommended using $\nu = 0.1$ and $\eta \leq 0.5$ for very shallow regression trees.

Via 5-fold CV, we tested the hyperparameter values near recommended defaults: five equally spaced η values from 0.0316 ($\approx 1/\sqrt{n}$) to 0.5, five equally spaced ν values from 0.01 to 0.1, and maximum leaf nodes of 2, 3, 4, and 5. More generally, as the number of possible combinations of η, ν, and leaf node counts increases, we randomly selected up to 100 hyperparameter combinations, following the strategy of Bergstra and Bengio (2012).

Each of these 100 ISLE ensemble hyperparameters was paired with the three values of α introduced in Section 2.3.3 for the Elastic Net path, yielding a total of 300 tested

hyperparameters (100×3). All M regression trees were constructed using scikit-learn v1.5.1 (Pedregosa et al., 2011). In this study, we set $M = 4000$.

Algorithm 2 Ensemble generation for the ISLE algorithm with regression trees as base learners (Friedman et al., 2003; Seni and Elder, 2010)

1: Initialize the memory function as the mean of the response:

$$F_0(\mathbf{X}) = \bar{y}$$

2: **for** $m = 1$ to M **do**

3: Draw a random subsample $S_m(\eta)$ (without replacement).

4: Fit a regression tree by minimizing the squared error with current memory F_{m-1}:

$$\gamma_m = \arg\min_{\gamma} \sum_{i \in S_m(\eta)} \left(F_{m-1}(\mathbf{x}_i) + \mathcal{T}(\mathbf{x}_i; \gamma) - y_i\right)^2$$

5: Update the memory function:

$$F_m(\mathbf{X}) = F_{m-1}(\mathbf{X}) + \nu \cdot \mathcal{T}(\mathbf{X}; \gamma_m)$$

6: **end for**

7: Define the ensemble as the collection of all fitted regression trees:

$$\text{ensemble} = \{\mathcal{T}(\mathbf{X}; \gamma_1), \ldots, \mathcal{T}(\mathbf{X}; \gamma_M)\}$$

2.3.5 Phylogenetic Network Inference

Once we obtain a sparse model—either from Elastic Net or ISLE, as described in the previous sections—we aim to identify the variables in $\mathbf{X}$ that have been selected. These selected variables correspond to specific rows in the CFs table, enabling the inference of a phylogenetic network from this reduced table.

In the case of Elastic Net, the selected variables are those associated with non-zero coefficients in the solution vector β, given a particular pair (λ, α) from the Elastic Net path. Specifically, let $\lambda_k \in \Lambda$ denote the k^{th} value of λ tested, and let β^{λ_k} be the corresponding

solution. The index set I^{λ_k}, representing the selected variables, is defined as:

$$\mathrm{I}^{\lambda_k} = \{j : \beta_j^{\lambda_k} \neq 0,\ j = 1, \ldots, p\}. \tag{2.4}$$

A similar process applies with ISLE (see Algorithm 3). Here, we examine the splitting variables used by the regression trees selected in the ISLE ensemble instead of non-zero coefficients in β. Selection is again determined via the Elastic Net path. Let $\mathbf{w}^{\lambda_k}$ denote the ISLE solution for a given $\lambda_k \in \Lambda$ and α, where $w_m^{\lambda_k} \neq 0$ indicates an active regression tree m in the ensemble. Let $X_j \in \gamma_m$ denote a splitting variable j used in the set of optimal parameters γ_m that define regression tree m. The index set I^{λ_k} is then constructed as described in Algorithm 3. Importantly, each taxon must appear in at least one selected row of the subsampled CF table induced by I^{λ_k}.

Algorithm 3 Compute the index set of CFs rows from a subsample of regression trees.

1: Choose a sparsity level λ_k (e.g., the one minimizing test error)
2: Initialize an empty set: $I^{\lambda_k} \leftarrow \emptyset$
3: **for** $m = 1$ to M **do**
4: **if** $w_m^{\lambda_k} = 0$ **then**
5: **continue** ▷ Skip inactive models
6: **end if**
7: Extract split indices from regression tree m:

$$I_m^{\lambda_k} = \bigcup_{X_j \in \gamma_m} \{j\}$$

8: Update the global index set:

$$I^{\lambda_k} \leftarrow I^{\lambda_k} \cup I_m^{\lambda_k}$$

9: **end for**
10: **return** I^{λ_k}

We now summarize all previously described steps in Algorithm 4, which we refer to as *Qsin* (Quartet Subsampling for Inference of Networks). Notably, subsampling guided by Elastic Net instead of ISLE can also be viewed as part of Qsin by skipping step 2, setting $\mathbf{T} = \mathbf{X}$ in step 3, and using the row selection defined in Eq. 2.4 instead of Algorithm 3 in

step 4. We refer to this variant as Qsin without ensemble creation. For phylogenetic network inference using SNaQ and subsamples (step 5 in Qsin), we used the default parameters and performed 15 independent runs, unless otherwise specified.

Detailed analyses of the time complexity for each step in Algorithm 4 are provided in Supplementary Material A.2. Building upon these individual results, we also derive the overall computational complexity of the complete algorithm, which we formalize in the following Theorem 1:

Theorem 1 (Proof available in Supplementary Material A.2)**.** *Let T be the number of taxa, n be the number of simulations, M the number of used regression trees, and $\rho \in (0, 1]$ the fraction of selected CFs rows used in reconstruction. Assume the number of SNaQ iterations is at most on the order of T. Then, the total time complexity of Algorithm 4 is $O\left(nT^5 + MT^4 n \log n + \rho T^7\right)$.*

We can also consider two extreme cases for ρ: one where all rows are selected and another where only enough rows are chosen to cover all species. By instantiating ρ under these two extreme scenarios, we can define the boundaries of the speed-up, within which the observed speed-up can lie:

Corollary 1 (Proof available in Supplementary Material A.3)**.** *Let T be the number of taxa. Under the same assumptions outlined in Theorem 1, and as $T \to \infty$, the worst-case scenario yields a speed-up factor of $O(1)$ (i.e., no speed-up) and the best-case scenario yields a speed-up factor of $O(T^3)$ relative to SNaQ.*

Therefore, as the number of species increases and ρ falls below 1, a speed-up in runtime can generally be expected.

For each row subsample, we repeated network inference at least 30 times, with the specified number of runs in each case. For each repetition, we min–max standardized the log-pseudolikelihood deviance, which is the reported log-pseudolikelihood score from SNaQ. If we let D be this log-pseudolikelihood deviance, then the relative deviance $D_{\text{relative deviance}}$ is

as follows:

$$D_{\text{relative deviance}} = \frac{D - D_{\min}}{D_{\max} - D_{\min}}$$

where $D_{\max}$ and $D_{\min}$ represent the maximum and minimum deviance, respectively, across all phylogenetic network inference repetitions using a given row subsample.

Algorithm 4 Qsin, a pipeline to reconstruct phylogenetic networks.

1. **Simulate Data:** Generate n random level-1 phylogenetic networks (see Section 2.3.2) and compute quartet log-loglikelihoods for each of the p rows in the CFs table to obtain the predictor matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ and the response vector $\mathbf{y} \in \mathbb{R}^{n \times 1}$ (Eq. 2.2).
2. **Fit Ensemble:** Train an ensemble of M regression trees on $(\mathbf{X}, \mathbf{y})$ (ISLE, Algorithm 2) and generate its prediction matrix $\mathbf{T} \in \mathbb{R}^{n \times M}$.
3. **Post-process:** Apply Elastic Net on $(\mathbf{T}, \mathbf{y})$ for sparse model selection (Eq. 2.3).
4. **Row selection:** Identify the set of informative variables given a sparsity parameter λ_k (Algorithm 3).
5. **Reconstruct Network:** Use the selected subset of CFs rows to reconstruct the phylogenetic network.

2.3.6 Model Evaluation

We used datasets presented in Solís-Lemus and Ané (2016) to evaluate the performance of the proposed pipeline. Two simulated datasets were included: the first contains 6 species, resulting in 15 rows in the CFs table based on 100 gene trees, and includes one hybridization event. The second simulated dataset contains 15 species, resulting in 1,365 rows in the CFs table based on 300 gene trees, and includes three hybridization events. A third dataset consists of empirical data from *Xiphophorus*, comprising 24 species and 10,626 rows in the CFs table based on 1,183 gene trees. Although the true number of hybridization events in *Xiphophorus* is unknown, we set h_{max}=2 in SNaQ, as this value was found to be optimal using the slope heuristic in Solís-Lemus and Ané (2016).

The quality of log-pseudolikelihood predictions was assessed using the root mean squared

error (RMSE), defined as $\sqrt{\frac{1}{n}\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2}$, where $\hat{\mathbf{y}}$ denotes the predicted values of $\mathbf{y}$ (the observed log-pseudolikelihood score; see step 1 in Algorithm 4). For simulated datasets, where the true network is known, we evaluated topological accuracy using the hardwired distance between the inferred and true networks. This metric, implemented as `hardwiredclusterdistance` in PhyloNetworks v0.16.4 (Solís-Lemus et al., 2017), captures the topological dissimilarity between two networks. Additionally, we computed the distance to the true network while ignoring the direction of the hybrid edge, referred to as the unrooted distance. To do this, we shifted the position of the hybrid node within the hybridization cycle of the inferred network and recalculated the distance. The new position of the hybrid node was chosen so that its descendant taxa matched those of the hybrid node in the true network. A custom script, named `ChangeDirection.py`, for performing this network manipulation is available in the accompanying GitHub repository: `https://github.com/ulises-rosas/qsin`.

We applied this unrooted distance to the second simulated dataset, which included 15 species, because it contains a hybridization cycle with four nodes and few taxa attached to the corresponding subnetworks, making the inference particularly challenging (see Fig. 9 in Solís-Lemus and Ané (2016)). As noted in that study, the unrooted distance provides a more appropriate measure of performance for this dataset. Accordingly, all comparisons for this dataset were made using the unrooted distance.

2.4 Results

When applying Elastic Net directly to the first simulated dataset (i.e., Qsin algorithm without ensemble creation), which included 15 rows in the CFs table, we found that the Elastic Net path selected more rows as the sparsity parameter decreased (Fig. 2.1). In the Elastic Net path (Fig. 2.1A), we subsampled across the region between the point at which no rows are selected (i.e., the maximum sparsity parameter is used) and the point at which all rows are selected. We found that a subsample consisting of approximately half of the

dataset achieved an accuracy of 93% (Fig. 2.2A), as measured by the proportion of times the inferred network had zero distance from the true network. This accuracy was the same as when using the complete dataset (93%, Fig. A.2A). Additionally, reducing the number of rows also decreased runtime for phylogenetic network inference (Fig. 2.2B): While processing the full dataset took a median of 47.19 seconds, the subsampled version completed in a median of 39.77 seconds. Obtaining only the subsample, which includes steps 1, 3, 4 of the Qsin algorithm and hyperparameter tuning, took 28.98 seconds. There was approximately a 15% decrease in network inference running time on this small dataset.

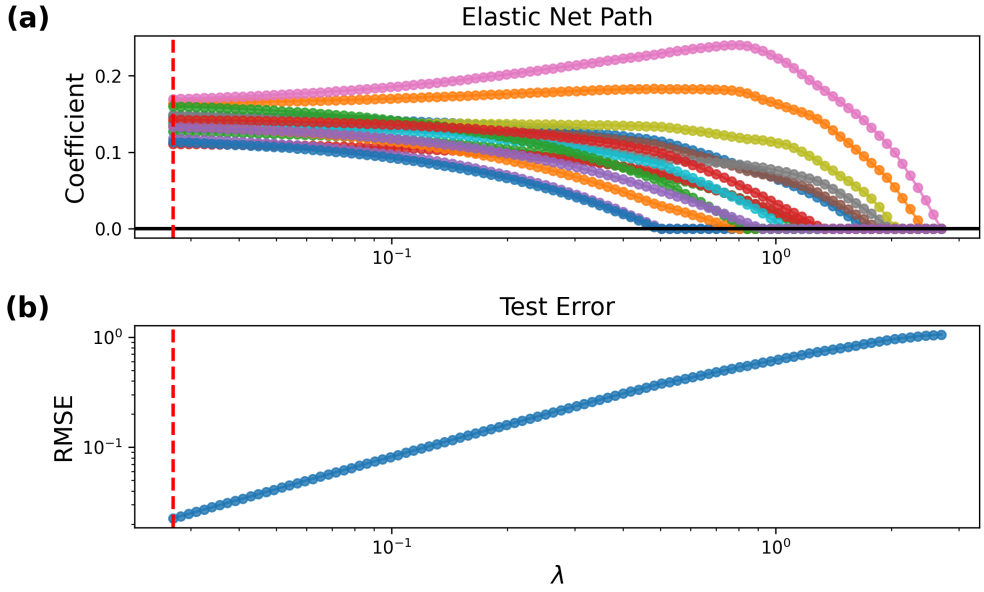


Figure 2.1: The Elastic Net path was directly applied to a CFs table with 15 rows, involving 6 species. Elastic Net alone tended to keep adding rows without a clear point of convergence, as test error steadily decreased until nearly all rows were included. In **(a)**, each colored line represents a row in the CFs table. From right to left, we observe that when λ is at its largest value, no rows are selected—as indicated by all colored lines being at zero. Dashed red vertical line indicates the point in the path with the lowest error, which is when all the rows are active in the prediction. **(b)** shows the testing RMSE along the path, which decreases almost monotonically as more rows are selected. Before reaching the lowest point, the testing error does not show sign of converging. The best α obtained via 5-fold CV was 0.5.

We observed similar performance when we modified step 1 of Qsin algorithm (see Algorithm 4) by drawing the branch lengths of the random networks from an exponential distri-

bution (mean = 1.25; Fig. A.3) or left unmodified (Fig. A.4). However, when the branch lengths were optimized to obtain log-pseudolikelihood scores using `topologyMaxQPseudolik` instead of `topologyQPseudolik` (see Section 2.3.2) in PhyloNetworks v0.16.4, a subsample of half the dataset resulted in 0% accuracy (Fig. A.5). Uniform random selection of rows of the same size as in Fig. 2.2 also yielded 0% accuracy (Fig. A.6A).

While Elastic Net can successfully generate optimal subsamples on the first simulated dataset, there was no clear point along the path at which to stop subsampling, as the testing error did not converge to a distinct minimum within the region between no row selection and full row selection (Fig. 2.1B). Therefore, we proceeded with the ensemble learning-based approach of Qsin (see Algorithm 4).

We applied the ensemble learning-based approach of Qsin to the second simulated dataset which included 1365 rows in the CFs table. The ensemble of trees used by Qsin in step 2 introduces additional hyperparameters. To visualize this hyperparameter space, we used the second simulated dataset, holding out a random fold of the data and fixing $\alpha = 0.5$ (Fig. 2.3). We observed that varying the level of nonlinearity in the sparse ensemble—measured by the maximum allowed number of leaves in each regression tree—resulted in different prediction errors for the log-pseudolikelihood score. Across all tested values for the number of leaves, prediction error increased as η approached 0.5, a region corresponding to fewer random ensembles. The optimal value of ν, the hyperparameter controlling ensemble diversity, varied depending on the number of leaves. We also observed that increasing the number of leaves generally reduced the prediction error. For ensembles with a maximum of 2 and 5 leaves, the lowest errors were 0.175 and 0.155, respectively, in this held-out fold. However, increasing the number of leaves also led to less sparse solutions. For example, the best solution with 5 leaves included 1,258 out of 1,365 rows in the CFs table (92.2%).

Since our goal is to obtain highly sparse solutions, we fixed the maximum number of leaves to 2 and performed a 5-fold CV to estimate the remaining hyperparameters: α, η, and ν. Across the five folds, the hyperparameters that minimized the average prediction error were

$\alpha = 0.5$, $\eta = 0.0316$, and $\nu = 0.1$. Applying the Elastic Net path to the ensemble generated with these hyperparameters, we observed that the testing error reached its minimum between the extremes of no row selection and full row selection (Fig. 2.4). At the point of minimal error (0.178), as measure by RMSE along the path, 750 regression trees were selected, which, via Algorithm 3, spanned 578 rows of the CFs table (42.3%).

As in Fig. 2.1, we created equally spaced subsamples along the Elastic Net path, from close to the beginning up to the point of minimal error (i.e., optimal subsample), and reconstructed phylogenetic networks using only these row subsamples (Fig. 2.5). We observed a smooth increase in accuracy as the subsample size approached the optimal subsample, which achieved an accuracy of 70%. This accuracy was essentially the same as that obtained using the complete dataset (69%; Fig. A.2B). In contrast, uniform random selection of the same number of rows resulted in a lower accuracy of 58% (Fig. A.6B). Although both subsampling strategies had a median unrooted distance to the true network of 0, the interquartile range of these distances was 2 for the optimal subsample and 10.5 for uniform random selection. Thus, the optimal subsample not only improved accuracy but also produced networks with less variability. Additionally, reducing the number of rows also decreased runtime for phylogenetic network inference: While processing the full dataset required a median of 10.76 hours, the subsampled version completed in a median of 4.25 hours. Obtaining only the subsample, which includes steps 1-4 of the Qsin algorithm and hyperparameter tuning, required only 0.71 hours. There was approximately a 60% decrease in network inference running time.

The speed-up behavior shown in Fig. 2.2 and Fig. 2.5 is consistent with the theoretical result stated in Corollary 1. As the number of species increases and the fraction of selected rows in the CFs table, denoted by ρ, is less than 1, we can generally expect a speed-up in runtime.

We next applied Qsin, as described above, to the *Xiphophorus* dataset which included 10,626 rows in the CFs table. Fixing the maximum number of leaves to 2, the optimal set of hyperparameters obtained via 5-fold CV was $\alpha = 0.5$, $\eta = 0.1487$, and $\nu = 0.0775$.

The Elastic Net path generated from this configuration resembled that observed in the simulated dataset: the prediction error converged before reaching its minimum, at which point 763 out of 10,626 rows from the CFs table (7.81% of all rows) were selected (Fig. A.7). Since there was variability in inference accuracy for the previous datasets, we repeated the inference six times using different random seeds and selected the two networks with the highest log-pseudolikelihood scores, using the 7.81% subsample and $h_{\max} = 2$. The network with the highest score of -709.19 matched exactly the most supported network topology described in Solís-Lemus and Ané (2016), with $h_{\max} = 2$ (Fig. 2.6). Similarly, the second-best network (-742.63; Fig. 2.6B) matched the second most supported topology in that study (Fig. 2.6C). All remaining networks with lower scores captured different features of the top-scoring networks. For example, the first hybridization event originating from the branch of *X. xiphidium* to the branch of the most recent common ancestor of *X. cortezi* and *X. continens* (Fig. A.8A, B, C); others matched the second most supported network shown in Fig. 2.6B (Fig. A.8C), or matched the major tree—defined as the tree obtained by removing all hybrid edges with inheritance probabilities below 0.5—which corresponds to the major tree reported in Solís-Lemus and Ané (2016) for both $h_{\max} = 0$ and $h_{\max} = 1$ (Fig. A.8A, D). Finally, we increased the number of independent runs from 15 to 25 and then to 30. In these extended runs, a single repetition recovered the top-scoring network (Fig. A.9). In contrast, uniform random selection of rows of the same size did not recover the top-scoring network in any of these extended runs (Fig. A.10).

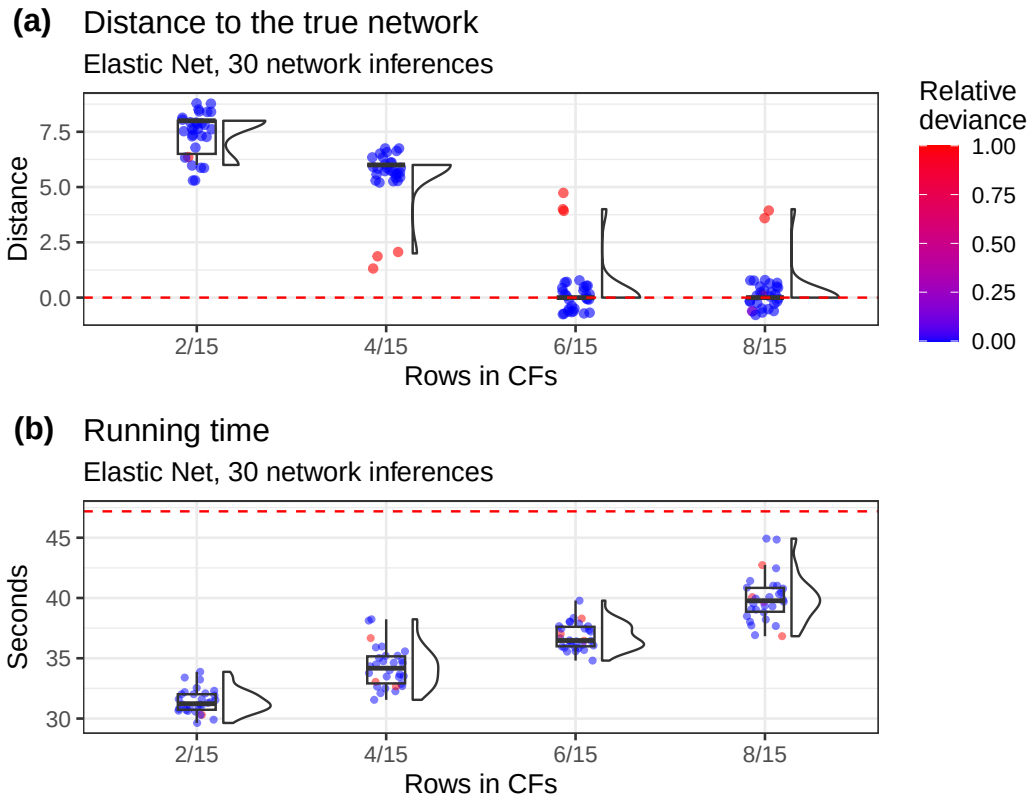


Figure 2.2: Accuracy of reconstruction using subsampling guided by Elastic Net, i.e, Qsin algorithm without ensemble creation, on a CFs table with 15 rows. **(a)** shows the distance to the true network. With 8 rows, the true network is recovered 93% of the 30 repetitions. The red dashed line indicates zero distance. **(b)** shows inference time for different row selection sizes. The red dashed line indicates the median time required to reconstruct the network using the full dataset, which is approximately 47.2 seconds. Each repetition used 10 independent runs. Uniform random selection of 8 rows had lower accuracy (Fig. A.6A). Relative deviance denotes the scaled log-pseudolikelihood score. In summary, results show that using Elastic Net to select about half of the CF rows recovered the true network in most runs while reducing inference runtime by approximately 15%, whereas random subsampling of the same size failed.

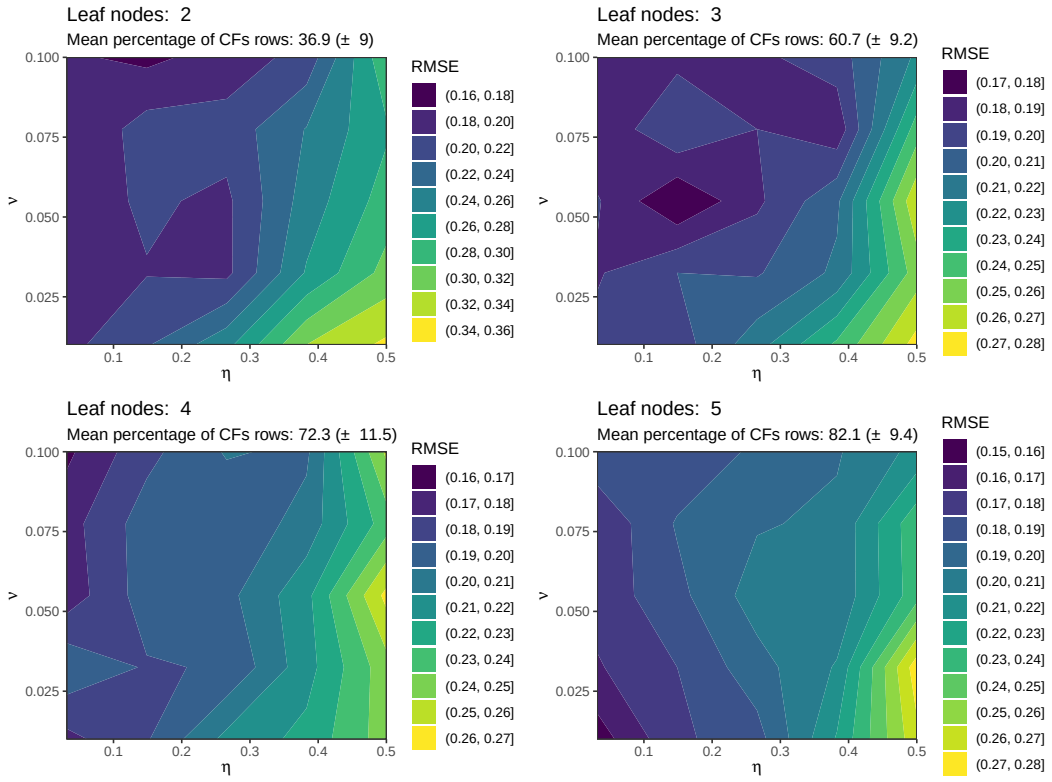


Figure 2.3: Hyperparameter space of ISLE ensemble with different level of non-linearity as measured by the number of leaf nodes in the regression trees. The darkest area in each panel represents the space with the lowest testing error, RMSE. The grid exploration is based on one realization of the training and testing set with random assignment. All the RMSEs represent the lowest error in the Elastic Net path. In summary, ensembles with 2 leaves achieved the best balance of accuracy and sparsity, while 5-leaf trees selected most of the rows without a clear gain in accuracy.

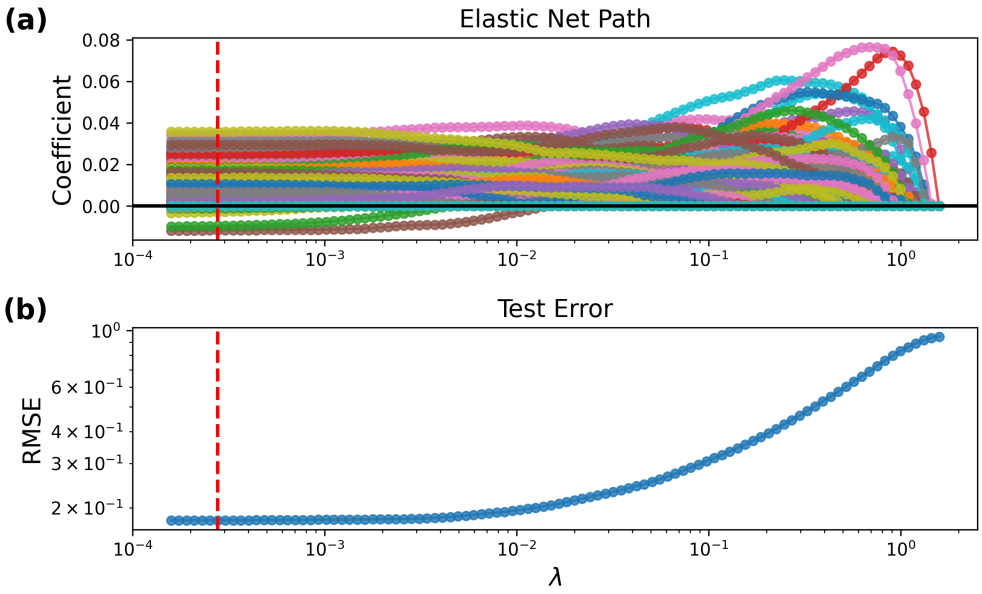


Figure 2.4: The Elastic Net path applied to the ISLE ensemble on a CFs table with 1,365 rows, involving 15 species. Hyperparameters were set to 2 leaf nodes, $\eta = 0.0316$, and $\nu = 0.1$. In **(a)**, each colored line represents a regression tree. From right to left, we observe that when λ is at its largest value, no regression trees are selected — as indicated by all colored lines remaining at zero. The red dashed vertical line marks the point along the path with testing error convergence. **(b)** shows the testing RMSE along the elastic net path, reaching its testing error convergence when 750 out of 4,000 regression trees are active—corresponding to 578 rows in the CFs table. In summary, using the ISLE ensemble, prediction error converged well before all trees were active, so that only a fraction of the rows (42.3%) was enough to reach stable accuracy.

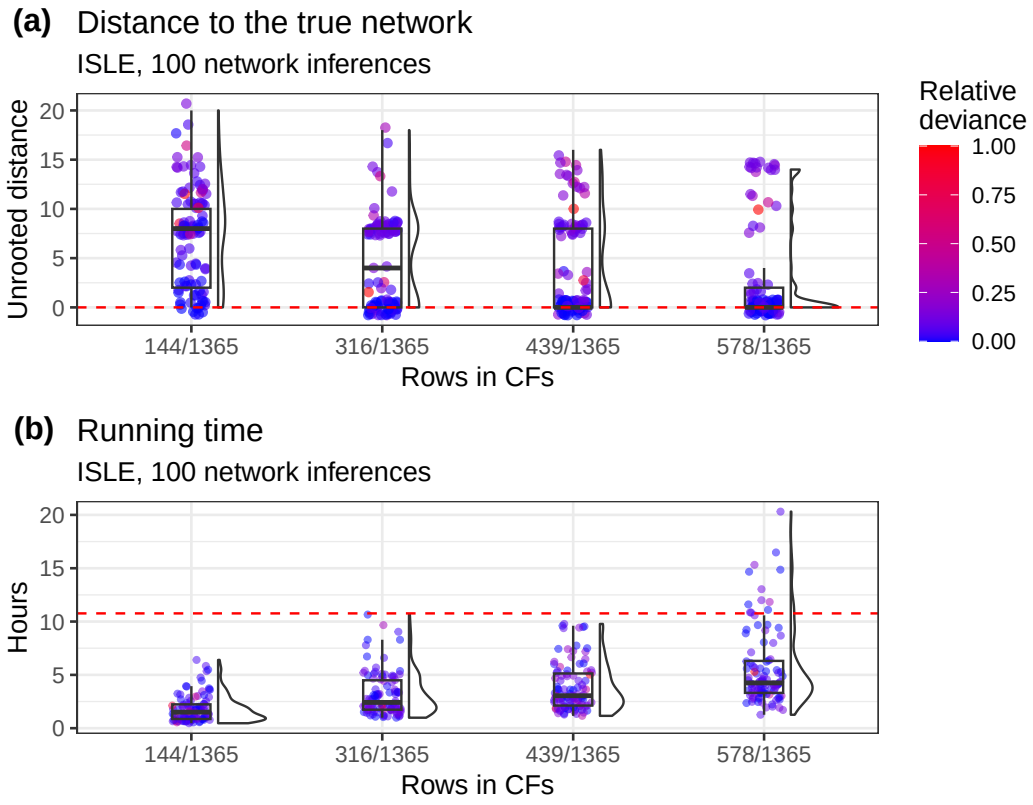


Figure 2.5: Accuracy of reconstruction using subsampling guided by ISLE, i.e., Qsin algorithm, on a CFs table with 1365 rows. **(a)** shows the unrooted distance to the true network. With 578 rows, the unrooted true network is recovered 70% of the 100 repetitions. The red dashed line indicates zero distance. **(b)** shows inference time for different row selection sizes. The red dashed line indicates the median time required to reconstruct the network using the full dataset, which is approximately 10.76 hours. Uniform random selection of 578 rows had lower accuracy (Fig. A.6B). Relative deviance denotes the scaled log-pseudolikelihood score. In summary, ISLE-guided subsampling reliably recovered the true network in most runs while reducing runtime by $\approx 60\%$, whereas random subsampling of the same size did not achieve similar accuracy.

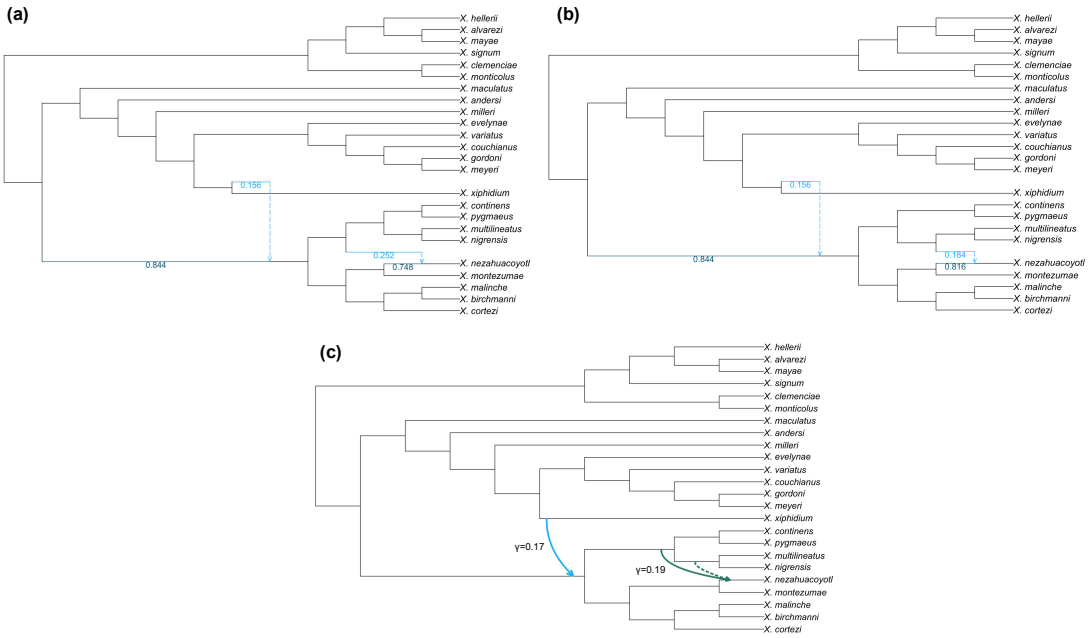


Figure 2.6: Phylogenetic network reconstruction for the *Xiphophorus* dataset using a subsample of 763 rows (7.81% of the CFs table) and the two best log-pseudolikelihood scores. **a)** shows the network with a log-pseudolikelihood score of -709.218, and **b)** shows the one with -742.635. The subsample was obtained using the ISLE algorithm with hyperparameters set to $\alpha = 0.5$, two leaf nodes, $\eta = 0.1487$, and $\nu = 0.0775$. The hyperparameters α, η, and ν were selected via 5-fold CV. For reference, **c)** shows the best networks obtained in Solís-Lemus and Ané (2016). The solid lines indicate the most supported network, while the skyblue solid line and green dashed line represent the second most supported one.

2.5 Discussion

We demonstrate that redundancy in the CFs table can be effectively exploited through machine learning–guided subsampling (Qsin, Algorithm 4). This subsampling approach accounts for the non-independence among CFs rows and can greatly reduce running times. In two simulated datasets, we were able to recover the true networks with high accuracy using only about half of the data. Such reductions yielded results comparable to those obtained with the complete dataset. In the empirical dataset, we found that the same phylogenetic network could be reconstructed as when using the entire dataset. The reduction in running time achieved by decreasing the number of rows is consistent with the theoretical result stated in Theorem 1 (Corollary 1).

Sparse learning has shown great promise for reducing model complexity across domains, including neural networks (Srivastava et al., 2014; Lemhadri et al., 2021). The core principle is that, even if the true data-generating process is unknown, it can often be represented in a lower-dimensional space, allowing variance and noise to be reduced (Hastie et al., 2009, pp. 610). Previous approaches using sparse learning for data size reduction first fixed an expected subsample size and then selected a sparse model that best matched that number of variables (Ecker et al., 2022). In contrast, our approach fixed model complexity, as measured by the number of leaf nodes, and allowed the data to determine the optimal subsample size. To our knowledge, this represents the first use of a non-parametric sparse model such as ISLE for data reduction in phylogenetics. This reduction is possible because the ISLE algorithm typically converges quickly to a solution along the Elastic Net path (Hastie et al., 2009, pp. 620), owing to its importance sampling strategy (Seni and Elder, 2010). This strategy favors regression trees that resample variables most relevant for improving prediction (i.e., building a strong ensemble) (Liu et al., 2022). Adding variables outside this set of important predictors tends only to introduce noise into the ensemble, or at best, provide no improvement—a behavior that is consistent with our ISLE solutions (Fig. 2.4,

Fig. A.7).

A non-linear model such as ISLE appears necessary because relationships among quartet log-likelihoods of rows in the CFs table can be far more complex than linear, particularly when predicting the overall log-pseudolikelihood score in the presence of beyond-level-1 phylogenetic networks, an area of active research (Allman et al., 2025a). Since our goal was to achieve substantial data reduction, we restricted the ensemble to trees with only two leaf nodes, also known as stump trees (Oliver and Hand, 1994). Although these regression trees are very simple, they can still yield complex relationships when used in an ensemble (Seni and Elder, 2010). For instance, they were the first trees used in AdaBoost algorithms (Freund et al., 1996) and have more recently been applied to improve out-of-distribution predictions in deep learning (Andriushchenko and Hein, 2019).

The Qsin algorithm makes use of Elastic Net to guide the subselection which is more suitable in high dimensional problems ($p \gg n$) than Lasso. Unlike Lasso, the number of variables selected by Elastic Net is not upper-bounded by the number of observations (n) (Friedman et al., 2010; Gaines et al., 2018), and it also accommodates correlated variables by leveraging a grouping effect. When variables are correlated, Lasso tends to pick one and ignore the rest, leading to have erratic behavior in the path (Friedman et al., 2010). Accounting for variable correlation is important as rows in the CFs table are inherently not independent. In the context of phylogenetics, this is not the first study to recognize the importance of correlation when implementing sparse learning methods. For instance, in the $\ell 1ou$ method for selecting the optimal number of shifts in trait optima, Khabbazian et al. (2016) accounted for the correlation of the coefficients in their sparse learning model using Lasso with a backward selection step that discards variables not improving the information criterion. More recently, the ESL method for associating trait presence or absence with a hot-encoded DNA alignment, Kumar and Sharma (2021) accounted for the cluster in groups of different genes by using grouped Lasso. Thus, when strong correlations among variables are expected, sparse learning models that explicitly address these correlations are preferable.

Uniform random selection had lower performance across all tested datasets compared to Qsin selection. It also showed greater variability in its unrooted distance to the true network in the second simulated dataset with 15 species. However, in this same dataset, uniform random selection of rows achieved an accuracy of about 58%, outperforming naïve random inference of the phylogenetic network. The utility of random subsampling in large datasets has also been observed in phylogenetic tree inference from quartets and is widely employed in methods such as SVDquartets (Chifman and Kubatko, 2014). This effectiveness further reinforces the idea that the dimensionality required to evaluate the fit of a phylogenetic hypothesis (e.g., number of sites in the alignment, number of gene trees, number of rows in the CFs) lies in a much lower-dimensional space. Consequently, we can exploit this redundancy using statistically principled algorithms (Vachaspati and Warnow, 2015; Warnow, 2018; Gupta et al., 2025), as demonstrated here.

Previous work has demonstrated the advantages of sparse learning for improving runtime (Khabbazian et al., 2016; Ecker et al., 2022), and our study confirms these benefits. Beyond empirical evidence, we also theoretically show that reductions in running time necessarily occur whenever a subsample of the dataset is taken (Corollary 1) and as the number of taxa T approaches infinity. These derivations align with intuition—the less data, the fewer computations—but our analysis further provides a precise characterization of how these savings scale with T, n, M, and ρ, thereby clarifying the conditions under which subsampling yields the greatest gains. As the number of species increases, the computational benefit described in Eq. 1 is primarily influenced by the fraction of rows selected, ρ. While we can set $\rho = 1/T$ to consistently achieve an order-of-magnitude speed-up compared with conventional inference, ρ is ultimately data-driven, and forcing it otherwise may not yield optimal solutions. Nonetheless, the Qsin algorithm effectively addresses the data size problem, which represents the first step in inference, and can also serve as a preprocessing stage for recent algorithmic advances such as divide-and-conquer approaches (Kolbow et al., 2025; Allman et al., 2025b) in phylogenetic network inference.

One limitation of the Qsin algorithm is the influence of parameters such as n and M on the overall time complexity (see Theorem 1). These parameters affect the runtime of steps such as log-pseudolikelihood evaluation, $O(nT^5)$, and the regression step, $O(MT^4 n \log n)$ (see Supplementary Material A.2)—both of which remain less costly than network inference when T is sufficiently large. However, these parameters can be reduced, forcing the sparse learning model to search for the best solution within a more constrained space, thereby yielding overall runtime improvements even for datasets with relatively few taxa. Another limitation is that the sparse learning model presented here does not incorporate network topology information. Instead, we rely on the rationale that an accurate prediction of the log-pseudolikelihood score implies an accurate prediction of the network topology. A more direct approach should consider both simultaneously. For example, Voznica et al. (2022) and Xie et al. (2025) successfully encoded phylogenetic trees in their machine learning models. However, the most effective data representation of phylogenetic network topology for machine learning models—particularly sparse models—remains an open question. Future improvements should also consider subsampling not only the full regression trees, as we did here, but potentially their components (e.g., internal nodes), an approach recently shown to improve ensemble accuracy (Liu et al., 2025).

On the other hand, if we want greater control over the frequency of taxa selected—in addition to the number of rows—we can use the constrained Lasso (Gaines et al., 2018), a direction that remains underexplored in phylogenetics. For example, let $\mathbf{z}_t \in \mathbb{R}^{M\times 1}$ be the column vector encoding the presence or absence of taxon t in an ensemble of M regression trees, such that each element in the vector is

$$z_{t,m} = \begin{cases} 1, & \text{if taxon } t \text{ is present in regression tree } m, \\ 0, & \text{otherwise.} \end{cases}$$

If we constrain $\mathbf{w}$ in Eq. 2.3 to take values from 0 to 1, then $\mathbf{w}$ can be interpreted as the ensemble posterior probabilities (Hastie et al., 2009, pp. 290). The weighted sum $\mathbf{z}_t^\top \mathbf{w}$

approximates the weighted average frequency of taxon t across the ensemble subselection. We can further require this weighted sum to be at least some constant, thereby enforcing a minimum frequency for each taxon (see Supplementary Material A.5 for the incorporation of these constraints into Eq. 2.3). This approach offers a promising direction when there is a need not only to obtain a subsample of the CFs table but also to ensure a minimum frequency of taxa in the selection. Such constraints can improve identifiability issues that arise from too few taxa in the hybridization cycle (Solís-Lemus and Ané, 2016; Tiley et al., 2023).

In this study, we showed that it is possible to reduce the dataset without compromising the accuracy of phylogenetic network inference. Reduced datasets allow us to include more taxa in current phylogenetic network analyses and potentially unlock more complex hypotheses on evolutionary histories, including speciation patterns (Mallet, 2007; Fontaine et al., 2015), comparative methods (Bastide et al., 2018), and biogeography (Burbrink and Gehara, 2018). This reduction in data size helps avoid computationally intensive runs, aligning with the principles of green computing (Kumar, 2022). Many avenues remain for applying sparse learning in phylogenetics (e.g., selection of topology-proposal moves), and our study contributes to these efforts. We provide the Qsin algorithm introduced here as a freely available set of scripts with a user-friendly tutorial on GitHub at `https://github.com/ulises-rosas/qsin`.

Reference List

Allman, E. S., C. Ane, H. Banos, and J. A. Rhodes, 2025a: Beyond level-1: Identifiability of a class of galled tree-child networks. *arXiv preprint arXiv:2504.21116*.

Allman, E. S., H. Baños, J. A. Rhodes, and K. Wicke, 2025b: Nanuq+: a divide-and-conquer approach to network estimation. *Algorithms for Molecular Biology*, **20 (1)**, 14.

Andriushchenko, M., and M. Hein, 2019: Provably robust boosted decision stumps and trees against adversarial attacks. *Advances in neural information processing systems*, **32**.

Bastide, P., C. Solís-Lemus, R. Kriebel, K. William Sparks, and C. Ané, 2018: Phylogenetic comparative methods on phylogenetic networks with reticulations. *Systematic biology*, **67 (5)**, 800–820.

Bergstra, J., and Y. Bengio, 2012: Random search for hyper-parameter optimization. *The journal of machine learning research*, **13 (1)**, 281–305.

Blischak, P. D., and Coauthors, 2023: *Species tree inference: a guide to methods and applications*. Princeton University Press.

Burbrink, F. T., and M. Gehara, 2018: The biogeography of deep time phylogenetic reticulation. *Systematic biology*, **67 (5)**, 743–755.

Chifman, J., and L. Kubatko, 2014: Quartet inference from snp data under the coalescent model. *Bioinformatics*, **30 (23)**, 3317–3324.

Cui, R., M. Schumer, K. Kruesi, R. Walter, P. Andolfatto, and G. G. Rosenthal, 2013: Phylogenomics reveals extensive reticulate evolution in xiphophorus fishes. *Evolution*, **67 (8)**, 2166–2179.

Ecker, N., D. Azouri, B. Bettisworth, A. Stamatakis, Y. Mansour, I. Mayrose, and T. Pupko, 2022: A lasso-based approach to sample sites for phylogenetic tree search. *Bioinformatics*, **38 (Supplement_1)**, i118–i124.

Fontaine, M. C., and Coauthors, 2015: Extensive introgression in a malaria vector species complex revealed by phylogenomics. *Science*, **347 (6217)**, 1258 524.

Frankel, L. E., and C. Ané, 2023: Summary tests of introgression are highly sensitive to rate variation across lineages. *Systematic Biology*, **72 (6)**, 1357–1369.

Freund, Y., R. E. Schapire, and Coauthors, 1996: Experiments with a new boosting algorithm. *icml*, Bari, Italy, Vol. 96, 148–156.

Friedman, J. H., T. Hastie, and R. Tibshirani, 2010: Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, **33**, 1–22.

Friedman, J. H., B. E. Popescu, and Coauthors, 2003: Importance sampled learning ensembles. *Journal of Machine Learning Research*, **94305**, 1–32.

Gaines, B. R., J. Kim, and H. Zhou, 2018: Algorithms for fitting the constrained lasso. *Journal of Computational and Graphical Statistics*, **27 (4)**, 861–871.

Grafen, A., 1989: The phylogenetic regression. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, **326 (1233)**, 119–157.

Gupta, A., S. Mirarab, and Y. Turakhia, 2025: Accurate, scalable, and fully automated inference of species trees from raw genome assemblies using roadies. *Proceedings of the National Academy of Sciences*, **122 (19)**, e2500553 122.

Guyon, I., and A. Elisseeff, 2003: An introduction to variable and feature selection. *Journal of machine learning research*, **3 (Mar)**, 1157–1182.

Hastie, T., R. Tibshirani, and J. Friedman, 2009: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed., Springer Series in Statistics, Springer, New York.

Hastie, T., R. Tibshirani, and M. Wainwright, 2015: *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, Boca Raton, FL.

Justison, J. A., C. Solis-Lemus, and T. A. Heath, 2023: Siphynetwork: An r package for simulating phylogenetic networks. *Methods in Ecology and Evolution*, **14 (7)**, 1687–1698.

Khabbazian, M., R. Kriebel, K. Rohe, and C. Ané, 2016: Fast and accurate detection of evolutionary shifts in ornstein–uhlenbeck models. *Methods in Ecology and Evolution*, **7 (7)**, 811–824.

Kolbow, N., S. Kong, and C. Solis-Lemus, 2025: Massively scalable inference of level-1 phylogenetic networks. *bioRxiv*, 2025–05.

Kong, S., C. Solís-Lemus, and G. P. Tiley, 2025: Phylogenetic networks empower biodiversity research. *Proceedings of the National Academy of Sciences*, **122 (31)**, e2410934 122.

Kumar, S., 2022: Embracing green computing in molecular phylogenetics. Oxford University Press, msac043 pp.

Kumar, S., and S. Sharma, 2021: Evolutionary sparse learning for phylogenomics. *Molecular Biology and Evolution*, **38 (11)**, 4674–4682.

Lefebvre, M., 2007: *Applied stochastic processes*. Springer Science & Business Media.

Lemhadri, I., F. Ruan, L. Abraham, and R. Tibshirani, 2021: Lassonet: A neural network with feature sparsity. *Journal of Machine Learning Research*, **22 (127)**, 1–29.

Liu, B., R. Mazumder, and P. Radchenko, 2025: Extracting interpretable models from tree ensembles: Computational and statistical perspectives. *arXiv preprint arXiv:2506.20114*.

Liu, B., M. Xie, H. Yang, and M. Udell, 2022: Controlburn: Nonlinear feature selection with sparse tree ensembles. *arXiv preprint arXiv:2207.03935*.

Mallet, J., 2007: Hybrid speciation. *Nature*, **446 (7133)**, 279–283.

Oliver, J. J., and D. Hand, 1994: Averaging over decision stumps. *European conference on machine learning*, Springer, 231–241.

Paradis, E., and J. Claude, 2002: Analysis of comparative data using generalized estimating equations. *Journal of theoretical biology*, **218 (2)**, 175–185.

Pedregosa, F., and Coauthors, 2011: Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, **12**, 2825–2830.

Rosset, S., and J. Zhu, 2007: Piecewise linear regularized solution paths. *The Annals of Statistics*, 1012–1030.

Seni, G., and J. Elder, 2010: *Ensemble methods in data mining: improving accuracy through combining predictions*. Morgan & Claypool Publishers.

Singh, R., and J. Stufken, 2023: Subdata selection with a large number of variables. *The New England Journal of Statistics in Data Science*, **1 (3)**.

Solís-Lemus, C., and C. Ané, 2016: Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. *PLoS genetics*, **12 (3)**, e1005 896.

Solís-Lemus, C., P. Bastide, and C. Ané, 2017: Phylonetworks: a package for phylogenetic networks. *Molecular biology and evolution*, **34 (12)**, 3292–3298.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014: Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, **15 (1)**, 1929–1958.

Tibshirani, R., 1996: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **58 (1)**, 267–288.

Tibshirani, R., and L. Wasserman, 2017: Sparsity, the lasso, and friends. *Lecture notes from*

"Statistical Machine Learning," Carnegie Mellon University, Spring.

Tiley, G., N. Liu, and C. Solís-Lemus, 2023: Extracting diamonds: identifiability of 4-node cycles in level-1 phylogenetic networks under a pseudolikelihood coalescent model. *bioRxiv*, 2023–10.

Vachaspati, P., and T. Warnow, 2015: Astrid: accurate species trees from internode distances. *BMC genomics*, **16 (Suppl 10)**, S3.

Voznica, J., A. Zhukova, V. Boskova, E. Saulnier, F. Lemoine, M. Moslonka-Lefebvre, and O. Gascuel, 2022: Deep learning from phylogenies to uncover the epidemiological dynamics of outbreaks. *Nature Communications*, **13 (1)**, 3896.

Wahba, G., and Y. Wang, 2019: Representer theorem. *Wiley StatsRef: Statistics Reference Online*, 1–11.

Wang, H., 2022: A note on centering in subsample selection for linear regression. *Stat*, **11 (1)**, e525.

Warnow, T., 2018: Supertree construction: opportunities and challenges. *arXiv preprint arXiv:1805.03530*.

Xie, T., H. Richman, J. Gao, I. F. A. Matsen, and C. Zhang, 2025: Phylovae: Unsupervised learning of phylogenetic trees via variational autoencoders. *ArXiv*, arXiv–2502.

Yao, Y., and H. Wang, 2021: A selective review on statistical techniques for big data. *Modern Statistical Methods for Health Research*, 223–245.

Zeng, Y., 2022: A study of tree-based methods and their combination. *arXiv preprint arXiv:2204.13916*.

Zou, H., and T. Hastie, 2005: Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **67 (2)**, 301–320.

Chapter 3

Non-linear phylogenetic regression using regularized kernels

Published in *Methods in Ecology and Evolution* (https://doi.org/10.1111/2041-210X.14385).

Ulises Rosas-Puchuri, Aintzane Santaquiteria, Sina Khanmohammadi, Claudia Solís-Lemus, Ricardo Betancur-R.

3.1 Abstract

Phylogenetic regression is a type of Generalized Least Squares (GLS) method that incorporates a modeled covariance matrix based on the evolutionary relationships between species (i.e., phylogenetic relationships). While this method has found widespread use in hypothesis testing via phylogenetic comparative methods, such as phylogenetic ANOVA, its ability to account for non-linear relationships has received little attention. To address this, here we implement a phylogenetic Kernel Ridge Regression (phyloKRR) method that utilizes GLS in a high-dimensional feature space, employing linear combinations of phylogenetically weighted data to account for non-linearity. We analyzed two biological datasets using the Radial Basis Function and linear kernel function. The first dataset contained morphometric data, while the second dataset comprised discrete trait data and diversification rates as response variable. Hyperparameter tuning of the model was achieved through cross-validation rounds

in the training set. In the tested biological datasets, phyloKRR reduced the error rate (as measured by RMSE) by around 20% compared to linear-based regression when data did not exhibit linear relationships. In simulated datasets, the error rate decreased almost exponentially with the level of non-linearity. These results show that introducing kernels into phylogenetic regression analysis presents a novel and promising tool for complementing phylogenetic comparative methods. We have integrated this method into Python package named phyloKRR, which is freely available at: `https://github.com/ulises-rosas/phylokrr`.

3.2 Introduction

Phylogenetic regression is mathematically equivalent to the Generalized Least Squares (GLS), which takes into account the non-independence of data points by incorporating information from a phylogenetic tree. This is necessary because species traits do not evolve independently, as they share a common evolutionary history (Felsenstein, 1985; Harvey et al., 1991). For instance, the length-weight relationships can vary across different clades due to specific growth rates associated with each clade (Harvey et al., 1991). To incorporate phylogenetic information into the GLS, we typically infer a covariance matrix derived from species traits (Harmon, 2019). The inference of the covariance matrix assumes that the traits evolve according to a specific stochastic process, such as Brownian motion (Lefebvre, 2007), with the timing determined by the branch lengths of the phylogenetic tree. Once the covariance matrix is modeled, data is weighted by the inverse of this matrix (see Eq. 3.1), and then an standard least-squares algorithm can draw a linear fit over the transformed data by finding an optimal coefficient vector β. This approach was initially introduced by Grafen 1989, also refered to as Phylogenetic Generalized Least-Squares (PGLS hereafter). The error function of PGLS can be expressed as:

$$\arg\min_{\beta} \frac{1}{n}(\mathbf{y} - \mathbf{X}\beta)^{\top}\mathbf{\Omega}^{-1}(\mathbf{y} - \mathbf{X}\beta) = J(\beta) \ , \tag{3.1}$$

where $\mathbf{y} \in \mathbb{R}^{n\times 1}$ is the response variable, $\mathbf{X} \in \mathbb{R}^{n\times p}$ is the matrix representing n taxa observations, each with p traits, and $\beta \in \mathbb{R}^{p\times 1}$ is the vector of coefficients. Additionally, $\mathbf{\Omega} \in \mathbb{R}^{n\times n}$ represents the modeled covariance matrix. It is important to notice that we are ignoring the intercept term in Eq. 3.1 as we are assuming data is already centered using the $\mathbf{\Omega}$ matrix (for more details, see the Materials and Methods section). Further justification and background for the above equation are available in the Supporting Information S.I.

The above approach assumes that a linear fit provides the most optimal relationship between the traits and response variable. However, little attention has been given to the ability of phylogenetic regression to account for the non-linear interaction of variables in this relationship (see Fig. 3.1) (Quader et al., 2004). For instance, the length-weight relationship, which is supported by multiple lines of evidence, is known to follow a cubic association. This relationship can be interpreted as the length variable interacting with itself in a non-linear manner. While it is possible to apply suitable transformations, like log-transformations in the context of length-weight relationships, to fit a linear function effectively, we often do not have prior knowledge about the variable interactions that drive the relationship (Harvey et al., 1991; Quader et al., 2004). Despite this limitation, the linear model remains the most widely used method in comparative biology and finds extensive application in hypothesis testing, such as phylogenetic ANOVA (Adams and Collyer, 2018b).

While non-linear models for phylogenetic regression that use convex optimization (i.e., to minimize the error, this type of optimization assumes that every local minimum is the global minimum) have already been proposed through the use of generalized estimating equations (GEE; Martins and Hansen, 1997; Paradis and Claude, 2002; Ives and Garland Jr, 2010), they have a finite set of basis functions that are parameterized from the available

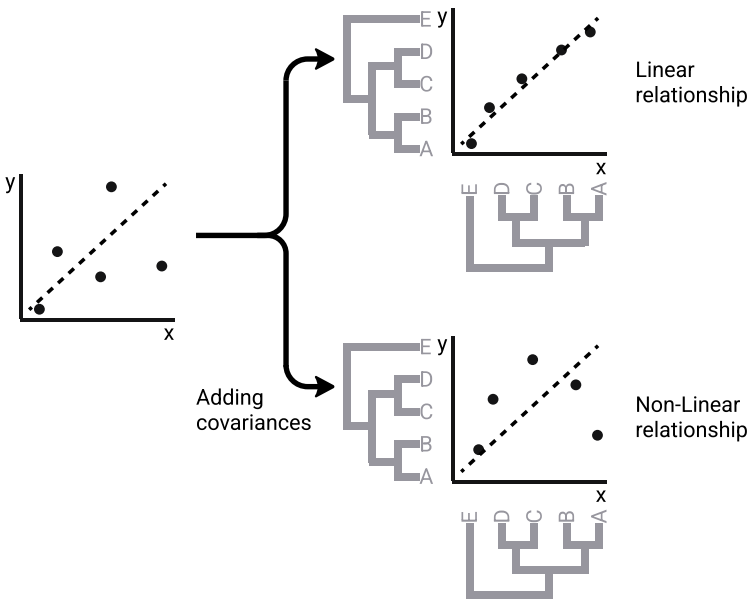


Figure 3.1: Schematic representation illustrating two possible scenarios that can occur when weighting data points with the modeled covariance matrix to account for phylogenetic non-independence and heteroscedasticity in trait data. In the first scenario, the incorporation of covariances can effectively linearize the relationships among data points. As a result, PGLS performs well in fitting data, capturing the linear relationships present. Conversely, in the second scenario, even after incorporating the covariances, data points do not linearize due to non-linear interactions between variables. In such cases, the limitations of PGLS become apparent as it may struggle to accurately capture the non-linear relationships present in data.

exponential distribution family, which constrains the hypothesis space. On the other hand, methods like Neural Network-based models (LeCun et al., 2015) or decision tree-based models (Breiman, 2001a; Chen et al., 2015) offer adaptive non-linear basis functions. However, as they are non-convex, they require substantial amounts of data to build robust models that can handle multiple local minima. An intermediate approach that combines convexity and non-linearity, and has the potential to account for an infinite number of basis functions, is kernel regression.

Kernel regression expands the number of dimensions based on linear combinations (inner products) of the original data. This expansion makes it possible to fit a linear model in cases where it was previously unattainable. Upon projecting the resulting linear model back from this higher dimensional space to the original space, we obtain a non-linear model

(Cortes and Vapnik, 1995). This technique, known as the "Kernel trick," can be efficiently integrated into standard regression models, even when the number of additional dimensions approaches infinity (Bishop and Nasrabadi, 2006). To address the risk of overfitting, a regularization term can be added to the regression model that already employs the Kernel trick. When this regularization term is in the form of the ℓ_2-norm (see Materials and Methods), the regularized kernel is referred to as Kernel Ridge Regression (KRR) (Cristianini and Shawe-Taylor, 2000; Cortes and Vapnik, 1995). Although Regularized Kernels have found widespread use in supervised machine learning for classification models, particularly through Support Vector Machines (Cortes and Vapnik, 1995; Bishop and Nasrabadi, 2006), their potential application in phylogenetic regression remains largely unexplored. The utilization of Regularized Kernel regression is well-suited for comparative analyses due to its ability to handle a relatively small number of data features for training the model. This is particularly beneficial for biological datasets, where obtaining a large number of traits for numerous species can be costly. Additionally, Regularized Kernel regression requires fine-tuning only a few parameters, usually just one or two.

In this study we (1) implemented a Regularized Kernel Regression model, more specifically a KRR model, that accounts for phylogenetic non-independence (phyloKRR hereafter); (2) tested two kernel functions that allowed expanding the number of dimensions; (3) validated the obtained models using simulations; and (4) applied the model to two biological datasets. Our main aim is to address non-linearity in phylogenetic regressions.

3.3 Material and Methods

3.3.1 Implementation of phyloKRR

In this section, our first step is to apply weights to the original data using the modeled covariance matrix. By doing so, we ensure that data incorporate the phylogenetic information. Next, we present the mathematical implementation of regularized kernel regression on this

weighted data. This approach allows us to leverage the benefits of kernel regression while accounting for the phylogenetic structure present in the data.

Let $\mathbf{Q}$ and $\mathbf{\Lambda}$ be the eigenvectors and eigenvalues matrices of the modeled covariance matrix $\mathbf{\Omega}$, respectively. Since $\mathbf{\Omega}$ is a positive semidefinite matrix, $\mathbf{Q}$ is an orthogonal matrix. Let $\mathbf{P} = \mathbf{Q}\mathbf{\Lambda}^{-1/2}\mathbf{Q}^\top$ such that $\mathbf{P}^\top\mathbf{P} = \mathbf{\Omega}^{-1}$ (Tung Ho and Ané, 2014). Then, we can re-write Eq. 3.1 as:

$$\begin{aligned} J(\beta) &= \frac{1}{n}(\mathbf{y} - \mathbf{X}\beta)^\top\mathbf{P}^\top\mathbf{P}(\mathbf{y} - \mathbf{X}\beta) \\ &= \frac{1}{n}(\mathbf{P}\mathbf{y} - \mathbf{P}\mathbf{X}\beta)^\top(\mathbf{P}\mathbf{y} - \mathbf{P}\mathbf{X}\beta) \\ &= \frac{1}{n}(\mathbf{y}^* - \mathbf{X}^*\beta)^\top(\mathbf{y}^* - \mathbf{X}^*\beta) \ . \end{aligned} \tag{3.2}$$

The transformed data, denoted as $\mathbf{y}^*$ and $\mathbf{X}^*$, represents the original data weighted with the phylogenetic information derived from the $\mathbf{P}$ matrix. The transformed data, denoted as $\mathbf{y}^*$ and $\mathbf{X}^*$, represents the original data weighted with the phylogenetic information derived from the $\mathbf{P}$ matrix. Notice that we are not using the intercept term in Eq 3.2 as we assume the data are already centered. This assumption is convenient as it allows us to ignore the intercept, β_0 (Hastie et al., 2009; Wang, 2022). If we have $\mathbf{y}_{w,uc}$ and $\mathbf{X}_{w,uc}$ as the weighted and uncentered response and taxa observation values, respectively, then we can define the weighted and centered response and observation values as $\mathbf{y}^* = \mathbf{y}_{w,uc} - \mathbf{1}\bar{y}_w$ and $\mathbf{X}^* = \mathbf{X}_{w,uc} - \bar{\mathbf{X}}_w$, respectively, where $\bar{y}_w \in \mathbb{R}^{1\times 1}$ is the average of $\mathbf{y}_{w,uc}$, and $\bar{\mathbf{X}}_w \in \mathbb{R}^{n\times p}$ is the matrix where each column has the average of each column of $\mathbf{X}_{w,uc}$ repeated n times. However, if necessary, we can recover the value of β_0 after finding the optimal β from the centered data, using $\beta_0 = \bar{y}_w - \bar{\mathbf{x}}_w^\top\beta$ (see Supporting Information S.I.3 for more details).

Hereafter, we follow the standard derivation for regularized kernel regression, as described in Bishop and Nasrabadi (2006) and Hastie et al. (2009). From Eq. 3.2, we can add a ℓ_2-norm regularization term for β to control over-fitting, and increase the number of dimensions for

each taxa observation for $\mathbf{X}^*$ matrix as follows:

$$
\begin{aligned}
J(\beta) &= \frac{1}{n}(\mathbf{y}^* - \mathbf{X}^*\beta)^\top(\mathbf{y}^* - \mathbf{X}^*\beta) + \lambda\beta^\top\beta \\
&= \frac{1}{n}\sum_{i=1}^{n}(\phi(a_i)^\top\beta - y_i^*)^2 + \lambda\beta^\top\beta \ , \qquad (3.3)
\end{aligned}
$$

where λ is the regularization coefficient that controls the relative importance of the regularization term, the a_i is the i^{th} taxa observation of $\mathbf{X}^*$, the function $\phi : \mathbb{R}^p \rightarrow \mathcal{H}$ maps the taxa observation a_i into the high dimensional feature space $\mathcal{H}$, also known as the Reproducing Kernel Hilbert Space, and y_i^* is the i^{th} element of the $\mathbf{y}^*$ vector. Eq. 3.3 is our new error function and we can minimize it by setting set the gradient $J(\beta)$ with respect to β equal to zero, which produces the following dual form:

$$
G(\alpha) = \frac{1}{n}(\mathbf{K}\alpha - \mathbf{y}^*)^\top(\mathbf{K}\alpha - \mathbf{y}^*) + \lambda\alpha^\top\mathbf{K}\alpha \ , \qquad (3.4)
$$

where $\mathbf{K} \in \mathbb{R}^{n\times n}$ is the Gram matrix (Bishop and Nasrabadi, 2006) whose elements are inner product of taxa observations mapped in the high dimensional feature space $\mathcal{H}$, and $\alpha \in \mathbb{R}^{n\times 1}$ is the column vector containing elements of the form $-(\phi(a_i)^\top\beta - y_i^*)/(n\lambda)$. We can define each element of matrix $\mathbf{K}$ as $k(i,j) = \phi(a_i)^\top\phi(a_j)$, where $k(i,j)$ is the kernel function that evaluate the above dot product without explicitly mapping the taxa observations into the high dimensional feature space $\mathcal{H}$. Eq. 3.4 is now operating in high-dimensional space and we can utilize least-squares to search for an optimal solution for α. This optimal solution will ultimately yield a non-linear solution to the original problem (Eq. 3.3), also known as the primal form. If we set the gradient of $G(\alpha)$ in equation 3.4 with respect to α equal to zero, we obtain:

$$\alpha = (\mathbf{K}^\top \mathbf{K} + \lambda n \mathbf{K})^{-1} \mathbf{K}^\top \mathbf{y}^*.$$

More detailed mathematical justification of the above derivations can be seen at the Supporting Information S.II. It can be shown that the time complexity of the above model is $O(n^3 + n^2 p)$ (Murphy, 2012), where n is the number of taxa observations and p the number of traits. Then, even if we increase the number of dimensions possibly to infinity (see below) the time complexity is mostly influenced by the number of taxa observations and the original number of dimensions (i.e., traits).

We tested two kernel functions, with the first one being the Radial Basis Function (RBF kernel hereafter) defined as $k(i, j) = \exp(-\gamma \|a_i - a_j\|_2^2)$. By definition, the exponential function can be expressed as a power series, and it can be demonstrated that this infinite summation results in a power series involving the inner product $a_i^\top a_j$ in the RBF. This property allows us to evaluate the inner product in an infinite-dimensional space. The value of γ controls the relative importance of the terms in the power series. For example, if γ is set to zero, the kernel evaluation yields a value of 1 for every inner product. When γ lies between 0 and 1, it is expected that the evaluation will gradually diminish in higher-order terms. However, depending on the data, even when the evaluation vanishes, it can still capture a high-dimensional space. Conversely, if the value of γ is too high, more weight is assigned to the higher-order terms in the power series, resulting in a possible data overfitting scenario.

The second kernel function is the linear kernel, defined as $k(i, j) = a_i^\top a_j + c$. Unlike the first kernel function, this kernel directly computes the inner product without previously mapping taxa observations into a higher-dimensional space. Although it is a simple kernel function, it proves useful in cases where the dataset already exists in a high-dimensional space, such as in the case of text classification datasets (Murphy, 2012; Géron, 2019). To

introduce flexibility, we have incorporated an additional parameter c, which allows for adjusting the shift of the inner product.

3.3.2 Simulated datasets

For a given tree i with 500 species, we simulated three traits under Brownian motion (Lefebvre, 2007). The time of this diffusion process is determined by the branch lengths of the tree i. Each species trait is considered a realization of this process (Felsenstein, 1985): its variance is proportional to the total length of tree i, and the covariance between two species is proportional to the branch lengths they share in tree i. Let $(\mathbf{X}_i)_{:j}$ be the column with trait j of the matrix containing all three traits $\mathbf{X}_i \in \mathbb{R}^{500\times 3}$. Then, $(\mathbf{X}_i)_{:j}$ can have the following multivariable normal distribution:

$$(\mathbf{X}_i)_{:j} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Omega}_i) \ ,$$

where $\mathbf{0} \in \mathbb{R}^{500\times 1}$ is the mean vector containing zeros, and $\mathbf{\Omega}_i \in \mathbb{R}^{500\times 500}$ is the matrix that stores the sum of shared of branch lengths among the species in tree i, being proportional to the covariance matrix assuming Brownian motion. We generated 100 random coalescent trees (i.e., $i \in \{1, \ldots, 100\}$), each consisting of 500 tips, and extracted their $\mathbf{\Omega}$ matrices by using the R package Ape v.5.7 (Paradis et al., 2004). Traits drawn from the multivariable normal distribution were obtained by using the Python package Numpy v.1.21.6 (Harris et al., 2020).

We weighted the trait matrix with the phylogeny information as follows: $\mathbf{X}_i^* = \mathbf{P}_i\mathbf{X}_i$, where $\mathbf{P}_i$ is derived from $\mathbf{\Omega}_i$ as explained in the previous section. Since $\mathbf{X}_i$ already has zero mean, $\mathbf{X}_i^*$ is centered as well. To introduce non-linearity in the simulated response variable for tree i, we employed the following relationship:

$$\mathbf{y}_i^* = \{(\mathbf{X}_i^*)_{:1} + (\mathbf{X}_i^*)_{:2} + (\mathbf{X}_i^*)_{:3}\}^b + \varepsilon \ , \tag{3.5}$$

Where $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ adds random noise to the response variable, and b determines the order of non-linearity. For example, when $b = 0$, $\mathbf{y}_i$ takes a vector of just ones. On the other hand, when $b = 1$, $\mathbf{y}_i$ can be fully explained by a linear model. For other values of b, $\mathbf{y}_i$ takes non-linear relationships, including interactions between columns of $\mathbf{X}_i$ due to the expansion of the polynomial. We tested the performance of the models (i.e., Kernel models and PGLS) over nine values of b ranging from 0 to 8.

3.3.3 Empirical datasets

We applied the phyloKRR model to two empirical datasets, each accompanied by its respective time-calibrated tree. The first dataset, referred to as the "fish dataset", was sourced from Santaquiteria et al. (under revision). It comprises both discrete (biogeographic areas) and continuous (tip diversification rates) data, serving as observation and response variables, respectively. This dataset includes a clade of tropical and temperate marine fishes, Syngnatharia (seahorses, dragonets, goatfishes, and relatives), consisting of 323 species. The aim was to test for a correlation between the geographic distributions of syngnatharian lineages and their tip diversification rates. We processed the categorical data from the biogeographic areas – encompassing the Indo-Pacific, eastern Pacific and Atlantic oceans – into continuous ones by using the one-hot encoding technique (Géron, 2019). This approach creates new binary columns, the count of which equals the number of categories. Each column takes a value of 1 if the species possesses the attribute related to that category, or 0 otherwise. As these binary columns represent categorical data, data were not standardized. Though one-hot encoding is widely used in machine learning applications (Géron, 2019), this technique has also been applied in the context of phylogenetic regressions (Harvey et al., 1991).

The second dataset, referred to as the "primate dataset", consists of continuous data from 90 primate species, as documented by Kirk and Kay (2004). The phylogenetic tree was obtained from the 10kTrees project (https://10ktrees.nunn- lab.org, last accessed on July 19 2023), which was reconstructed using Bayesian inference. In this case, our goal was to examine the correlation of morphological traits, specifically skull length and optic foramen area, with the orbit area. Since each column in this dataset has different units (i.e., area vs. length), we standardized each column by its standard deviation before inputting it into the model.

Assuming a Brownian motion stochastic process for all characters in both datasets, we obtained the modeled phylogenetic variance-covariance matrices using Ape v.5.7 (Paradis et al., 2004) assuming Brownian motion. The points were transformed by the modeled covariance matrix in the response variable and trait matrix. For visualization purposes only, we reduced these to one dimension using their first principal component (Fig. 3.2), as both trait matrices are multivariate. Notably, in the fish dataset, the linear relationship between diversification rates and the PC1 of geographical distribution is difficult to discern. Conversely, in the primates dataset, a linear relationship between orbit area and the PC1 of morphological characters, while not perfect, is more straightforward. While these datasets are not as large as the simulated datasets (notably, the primate dataset is rather small), where the kernel models can have better performance, the main purpose of adding these datasets is to complement the performance seen in simulated datasets with real case scenarios.

3.3.4 Evaluation

3.3.4.1 Error estimation

To assess the performance of the phyloKRR model, we can divide the complete dataset into two subsets: the training set and the testing set. The training set is used to obtain the optimal α and corresponding hyperparameters (see below), while the testing set contains

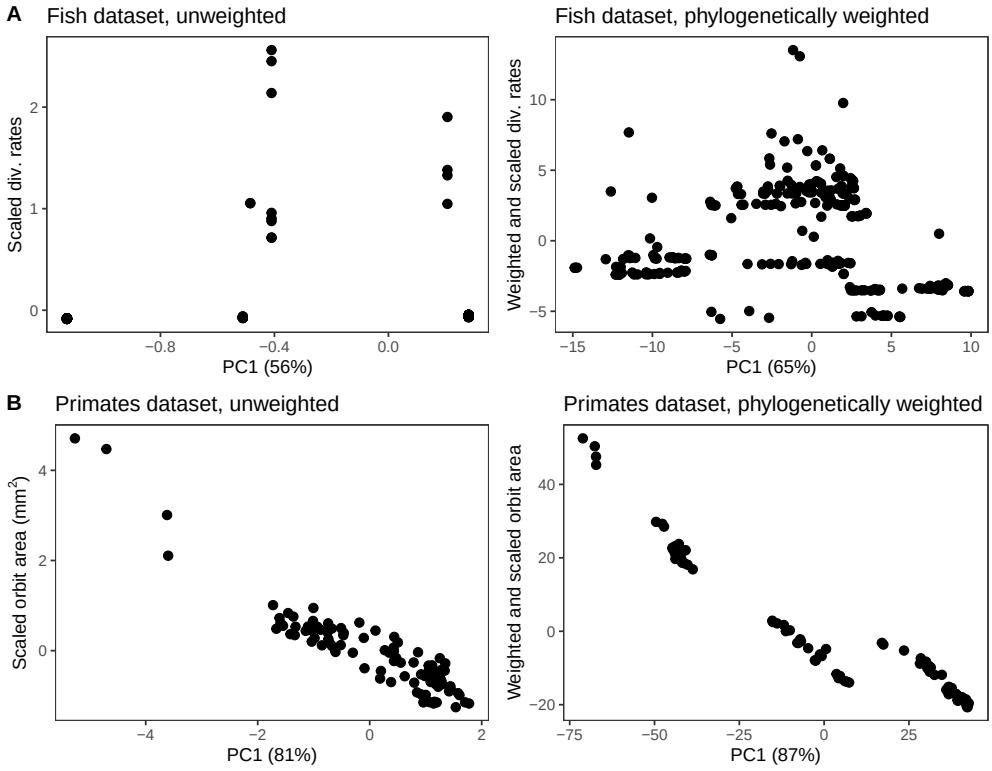


Figure 3.2: Empirical datasets before and after weighting data points using the modeled covariance matrix assuming Brownian motion. The weighting of data points (i.e., the $\mathbf{X}$ matrix with taxa observations and the response variable $\mathbf{y}$) was based on $\mathbf{\Omega}$ (see Section 2.1). **A**) shows the fish dataset, and **B**) shows the primate dataset. The y-axis in the fish dataset represents tip diversification rates, which is the number of speciation events minus the number of extinction events in a given time interval. In the left panels for **A**) and **B**), the x-axis corresponds to the PC1 projection of taxa observations, and the y-axis corresponds to the response variable. In the right panels for **A**) and **B**), the x-axis corresponds to the PC1 projection of the phylogenetically transformed taxa observations. The percentage of variance captured by PC1 is presented in parentheses in all panels.

unseen data that allows for a final evaluation of the model. Here, the training set comprised 60% of the data, while the testing set contained the remaining 40%.

Let $\alpha \in \mathbb{R}^{n\times 1}$ represent the optimal solution derived from the training set, and let $\mathbf{Z} \in \mathbb{R}^{t\times p}$ and $\mathbf{y}_z \in \mathbb{R}^{t\times 1}$ denote the matrix of taxa observations and the vector containing the response variable, respectively, in the testing set. Then, we can assess the model error of regularized kernel regression using the root mean squared error, RMSE_{kernel}, as follows:

$$\text{RMSE}_{kernel} = \sqrt{\frac{1}{t}\sum_{i=1}^{t}(\mathbf{K}_z\alpha - \mathbf{y}_z)_i^2} \ ,$$

where $\mathbf{K}_z \in \mathbb{R}^{t\times n}$ is the matrix where each row corresponds to the evaluated kernel function of each taxa observation in the testing set with respect to all training taxa observations.

3.3.4.2 Hyperparameter tuning

We also need to determine three hyperparameters: λ, which adjusts the regularization process, and kernel-specific function parameters such as γ for the RBF kernel and c for the linear kernel. Under a probabilistic approach, these hyperparameters act as priors for the model (Murphy, 2012, pp. 232). The optimal solution for the model is obtained by maximizing the posterior probability given the priors and data (see Supplementary Information S.II.1). Therefore, these hyperparameters cannot be directly learned from data. However, instead of testing all possible hyperparameter priors, which is computationally unfeasible, we can take random samples from a given hyperparameter distribution and evaluate their generalizability to unseen data. One approach for this is cross-validation (Hastie et al., 2009, pp. 241). This process involves randomly splitting the training set into k-folds. Each fold serves as a test set for a particular combination of hyperparameters, while the remaining k-1 folds are used for training the model. At the end of k iterations, we obtain a set of k errors, and we report the average. We repeated the process for a set of uniformly distributed hyperparameters and defined the optimal hyperparameter combination as the one with the lowest average error. For our analyses, we used 2-fold cross-validation for the primate dataset and 3-fold cross-validation for the rest of the datasets, including the simulations.

While selecting random points, it has been shown that if the range of hyperparameters is broad enough, near-optimal solutions can be obtained without the need to fit models for all possible random combinations of hyperparameters (Bergstra and Bengio, 2012). Then,

using cross-validation, we randomly chose 100 hyperparameter combinations from a broad range of values on a logarithmic scale. Specifically, for the empirical datasets, we set the range of λ from 2^{-5} to 2^{15}, the range of γ from 2^{-15} to 2^{3}, and the range of c from 2^{-5} to 2^{15}. For the simulated datasets, we broadened the range from 2^{-15} to 2^{15} for all hyperparameters to account for more possible scenarios of non-linearity.

3.3.4.3 Model comparison

The aforementioned approach provides us with a single optimal combination of hyperparameters. Subsequently, we can evaluate the model's performance using the obtained α and hyperparameters on the testing set. To ensure the robustness of phyloKRR in the empirical datasets, we ran 20 rounds of cross-validation with random assignments of training and testing sets. After completing the loop, we obtained 20 sets of errors for phyloKRR, either using the RBF kernel or linear kernel functions, and PGLS. For each iteration where phyloKRR was tested, either in the empirical or simulated dataset, we also performed PGLS on the testing set of the same partition and assessed its error. We later compared the models with the Wilcoxon rank-sum test, where p-values were adjusted for multiple comparisons.

3.3.4.4 Model inspection

While the kernel approach used here can account for very complex non-linear relationships among taxa traits, it can be seen as a black box model (Breiman, 2001b). To improve model explainability, we employed two techniques widely used in the machine learning community: Feature Importance (FI hereafter) analysis (Breiman, 2001a; Fisher et al., 2019) and Partial Dependence Plots (DPD hereafter) (Elith et al., 2008; Hastie et al., 2009, pp. 369), which are essentially heuristic solutions. The idea of FI analysis is to see how much the model error increases when the values of a given trait are permuted. If the MSE increases notably compared with the rest of the traits, it means that the trait has high importance in the model.

Once we know the top traits, we can characterize how they behave in the model by measuring the marginal effect of trait values, which is the main goal PDP. This is done by obtaining a number of quantiles of the trait of interest and, for each value, calculating the model's outcome by repeating the value across the entire trait column instead of using the full set of values. We repeated this process for all quantile values of the trait. The average prediction from this perturbation across the trait column is the partial dependence. For two interacting traits, instead of using a single set of quantiles, we obtained a grid of values from the two traits of interest and proceeded as explained above, yielding a 2D PDP.

To assess the performance of these two approaches, we simulated the following relationship:

$$\mathbf{y}^* = \{\mathbf{X}^*_{:1} + 2\mathbf{X}^*_{:2} + 3\mathbf{X}^*_{:3}\}^2 + \varepsilon \ , \tag{3.6}$$

Which has the shape of a bowl in four dimensions. Given the coefficients above, we expect the order of trait importance to be: third, second, and then first trait. Furthermore, expanding the polynomial, we expect to obtain first order interactions among the traits. We fit the simulated dataset generated by the Eq. 3.6 model using the same range of hyperparameters described in Section 3.3.4.2. For FI, the number of permutations applied to each feature column was 100. For PDP with one trait, the sampling was 100 quantile values from the trait distribution, and for 2D PDP, it was 50 quantile values from a pair of traits, forming a grid of 2500 (50 x 50) trait combinations.

3.4 Results

When testing with simulated datasets, phyloKRR with the RBF kernel consistently captured the non-linearity of data across varying degrees of non-linearity (Fig. 3.3). When the data were linear ($b = 1$), phyloKRR with the RBF kernel showed no significant difference ($p >$

0.05, Fig. 3.4) in performance compared to PGLS. For taxa observations where b differed from 1, the performance improvement of phyloKRR with the RBF kernel was statistically significant (from $p = 0.0036$ for $b = 7$ to $p < 10^{-33}$ for $b = 2$). Regarding phyloKRR with the linear kernel, no significant differences were detected between it and PGLS for all tested b values ($p > 0.05$), except when $b = 0$ ($p < 10^{-8}$, Fig. 3.4).

To assess the magnitude of the difference in the non-linear simulated datasets, we examined the difference between PGLS and phyloKRR errors within each simulated dataset (Fig. 3.5). In the case of phyloKRR with the RBF kernel, the minimum average difference was 0.00253 for $b = 0$, and the maximum average difference was $10^{4.06}$ for $b = 8$. We can also observe that the difference in error between PGLS and phyloKRR with the RBF kernel exponentially increases as the degree of non-linearity increases. For phyloKRR with the linear kernel, the minimum average difference was 0.00465 for $b = 0$, and the maximum average difference was $10^{3.49}$ for $b = 8$. However, we did not observe the error between PGLS and phyloKRR with the linear kernel to exponentially increase as the degree of non-linearity increases, as it highly overlaps with the zero line.

Once we determined that phyloKRR with the RBF kernel appropriately captures the non-linear datasets, we performed model inspection analyses (refer to Section 3.3.4.4). For the simulated data generated from Eq. 3.6, which describes a quadratic relationship, the FI analysis was able to identify the expected most important traits (Fig. 3.6A), and the PDP successfully captured the quadratic response variable (Figs. 3.6B, 3.6C, 3.6D), something that was not achieved with phyloKRR with the linear kernel (Fig. S1). Furthermore, the 2D PDP also captured different trait interaction sections of this quadratic relationship (Fig. S2).

For the empirical datasets, we conducted a grid search covering all combinations of hyperparameters for a single assignment of the training and testing sets (refer to Section 3.3.4.3). We found that the hyperparameter space differed across the examined empirical datasets, as illustrated in Fig. 3.7. With phyloKRR with the linear kernel, it became evident that the

c hyperparameter had a minimal impact on the model performance in both datasets; the error rate remained relatively consistent beyond specific ranges of λ values. Based on these hyperparameter spaces, we then randomly chose the optimal hyperparameter combination for subsequent assignments of the training and testing sets.

The distribution of errors is summarized in Fig. 3.8 for both empirical datasets. In the fish dataset, phyloKRR with the RBF kernel exhibits a significantly superior performance compared to phyloKRR with the linear kernel and PGLS ($p < 10^{-8}$ using the Wilcoxon rank-sum test). When comparing the average root mean squared error (RMSE), phyloKRR with the RBF kernel achieves a reduction in the error of up to 20% compared to PGLS. Conversely, for the primates dataset, no statistically significant differences were observed among the tested models.

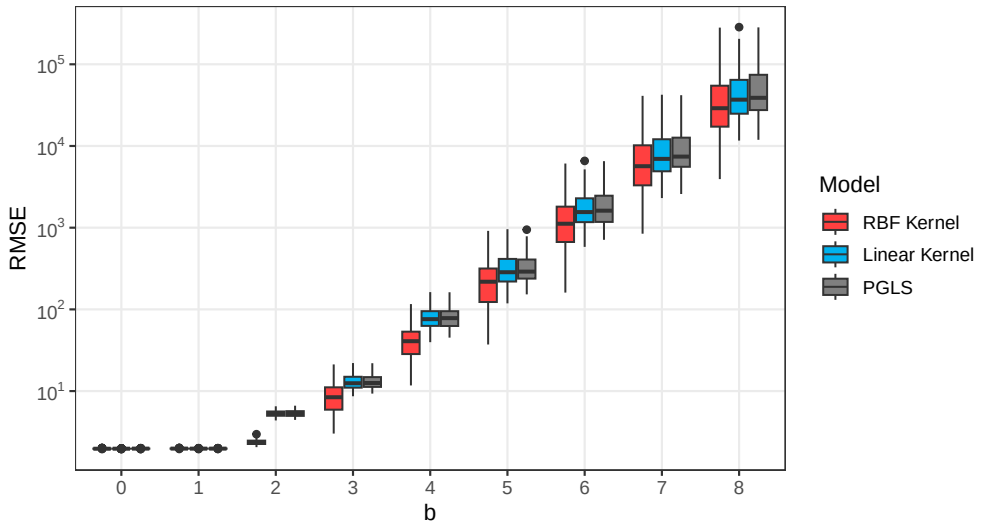


Figure 3.3: Performance of phyloKRR and PGLS under varying degrees of non-linearity in simulated datasets. Performance for phyloKRR is shown using the RBF kernel and the linear kernel. The parameter b from Eq. 3.5 controls the non-linearity. Each model's performance at a given value of b was evaluated using 100 simulations. We scaled the y-axis for improved visualization of the trend. A close-up of the point distribution when $b = 0$ and $b = 1$ can be seen in Fig. 3.4.

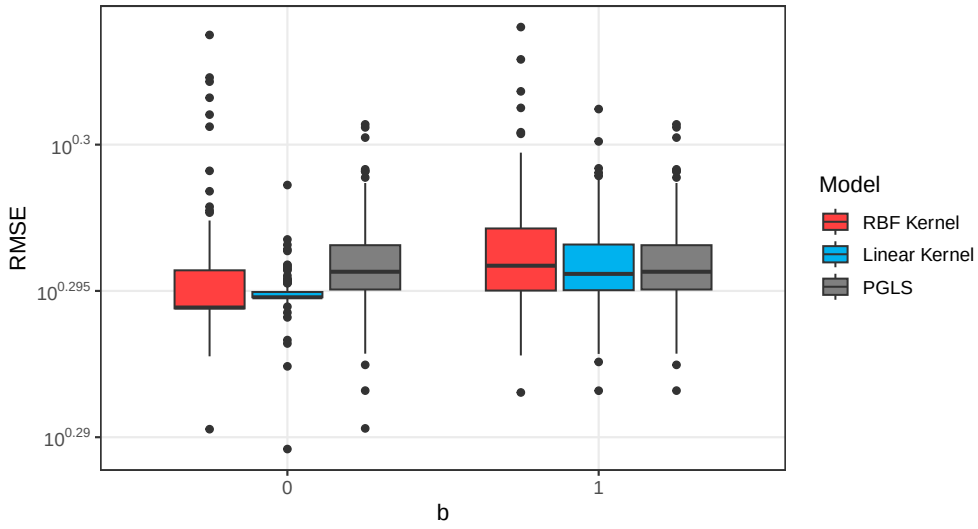


Figure 3.4: Performance of phylokRR and PGLS when $b = 0$ and $b = 1$ according to Eq. 3.5. Performance for phyloKRR is shown using the RBF kernel and the linear kernel. When $b = 0$, Eq. 3.5 simulates datasets with $\mathbf{y}^* = 1$ and noise variance of 1. When $b = 1$, it simulates datasets with linear relationships. Each model's performance at a given value of b was evaluated using 100 simulations. We scaled the y-axis for improved visualization of the trend. When $b = 1$, both kernel models were able to find a linear model.

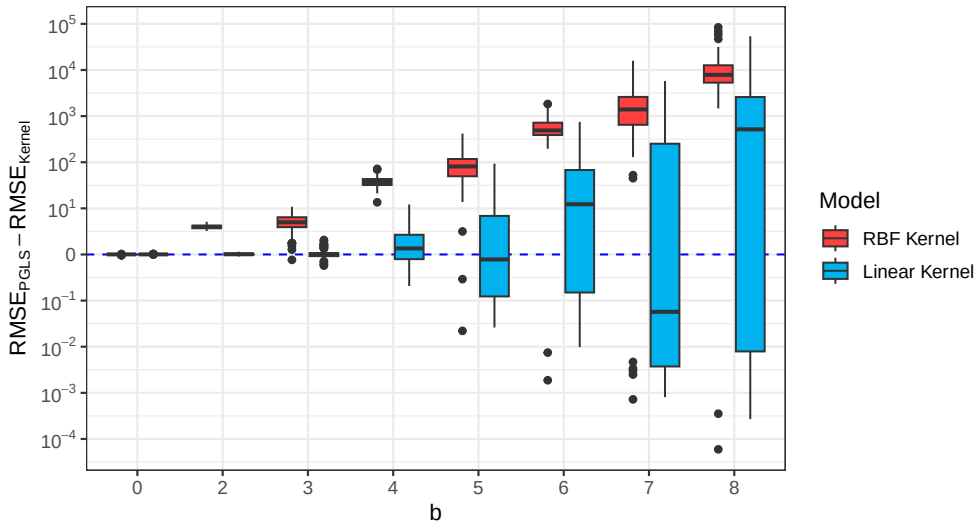


Figure 3.5: Error difference between phyloKRR (using RBF kernel and linear kernel) and PGLS under varying degrees of non-linearity in simulated datasets. The parameter b from Eq. 3.5 controls the non-linearity. Each point in the box plot represents the difference between the RMSE of PGLS and phyloKRR in a given simulated dataset. The blue dashed line represents no difference between the PGLS and phyloKRR. We scaled the y-axis for better visualization of the trend.

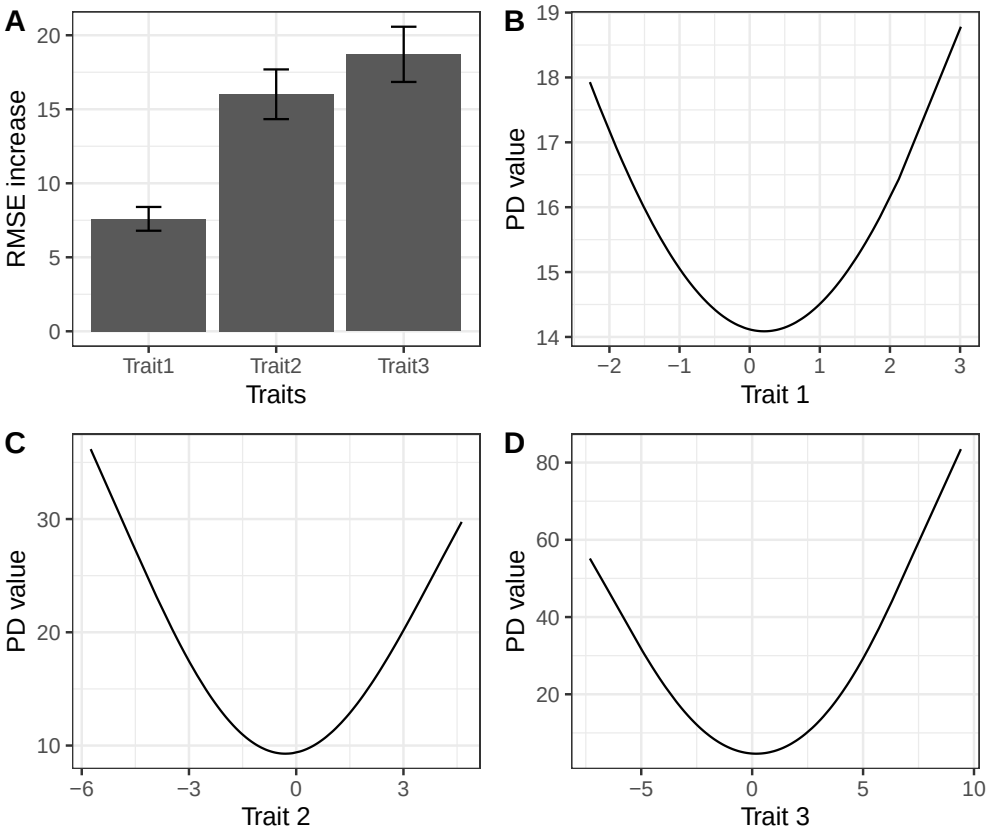


Figure 3.6: Model inspection analyses using phyloKRR with the RBF kernel on simulated data generated from Eq. 3.6. In **A**), we present permutation FI analysis. Notice that the rank for FI follows the weights given in Eq. 3.6. In **B**), we show the marginal contribution of the first trait in the overall model. In **C**) and **D**), we show the same for traits 2 and 3, respectively. Notice that the curves in **B**), **C**), and **D**) follow the quadratic response as we expect from Eq. 3.6. PD value stands for Partial Dependence value.

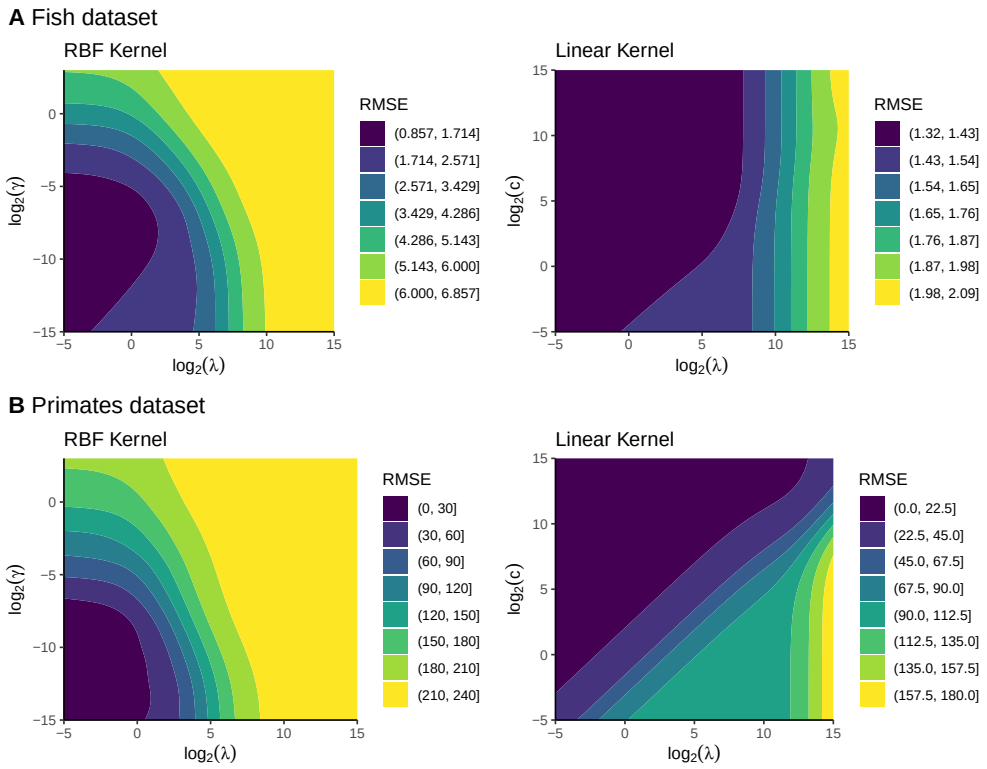


Figure 3.7: Hyperparameter space of phyloKRR with RBF and linear kernel in both tested empirical datasets. **A**) shows the hyperparameter space for the fish dataset, whereas **B**) shows the hyperparameter space for the primates dataset. The darkest area in each panel represents the space with the lowest error, as measured by RMSE. The grid search is based on one realization of the training and testing set with random assignment.

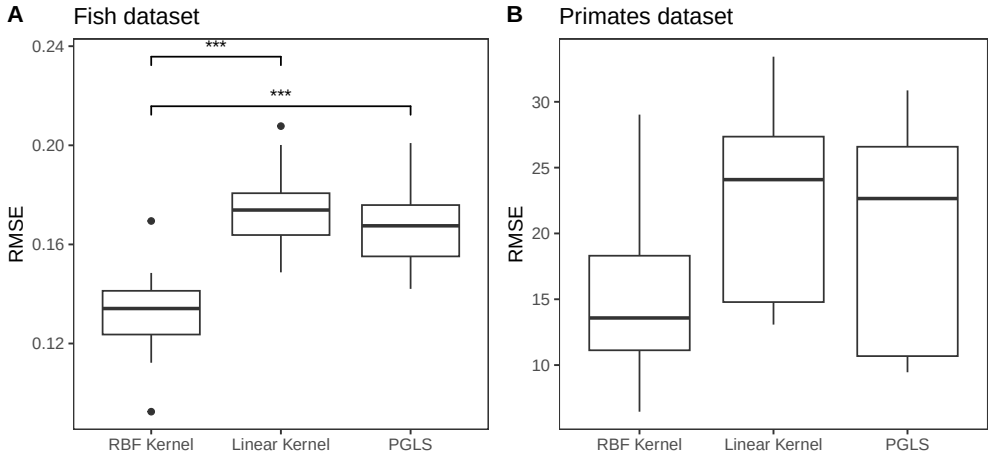


Figure 3.8: Error distribution after 20 rounds of cross-validations for all models in both empirical datasets. **A**) shows the errors of the tested models for the fish dataset. A statistically significant difference was observed between phyloKRR with the RBF kernel and both phyloKRR with the linear kernel and PGLS ($p < 10^{-8}$ according to the Wilcoxon rank-sum test). **B**) shows the errors of the tested models for the primate dataset. In this case, no statistically significant differences were found among the models.

3.5 Discussion

To account for non-linearity in phylogenetic regression, we implemented regularized kernel regression that accounts for phylogenetic non-independence (phyloKRR). Our results show that phyloKRR with the RBF kernel function outperform PGLS when linearity cannot be assumed. This performance advantage was apparent in both simulated and empirical datasets. When simulated datasets displayed linear behavior, phyloKRR with both tested kernel functions were also able to recover a linear model (Fig. 3.4). Using model inspection analyses, we were able to characterize the underlying non-linear behavior of a simulated dataset with a response variable following a quadratic function (Fig. 3.6). These results show that introducing kernels into phylogenetic regression analysis presents a novel and promising tool for complementing phylogenetic comparative methods.

Machine learning models can capture the non-linearity of data efficiently, especially in the context of large datasets. They have found use in various phylogenetic analyses (e.g., Voznica et al., 2022; Azouri et al., 2021; Jiang et al., 2022). However, these methods rely on the assumption that observations are independent and identically distributed (e.g. Stochastic Gradient Descent methods) (LeCun et al., 2002; Vovk, 2013; Ruder, 2016). This assumption may not hold true if the taxa observations show high covariance. As a result, it is essential to use phylogenetic information to weight data, as demonstrated in Eq. 3.2, before feeding it into machine learning models. This step ensures the proper implementation of non-linear models that handle multi-species information data. Additionally, a similar approach as outlined in the methods section (Eq. 3.3), can be employed using generalized estimated equations (Eq. 3 from Paradis and Claude, 2002) to address non-continuous response variables such as binary or count response variables.

The use of a regularization term in the error function enhances the model's generalizability and robustness to outliers (Tibshirani, 1996; Zhang et al., 2020; Jiang et al., 2022), provided it is properly tuned (Srivastava et al., 2014; Zhang et al., 2021). Regularization

techniques applied to prediction errors have successfully been adapted for PGLS (Adams et al., 2022), making it more robust to outliers. In our simulations, we also found a similar effect from regularization when the data had many outliers ($b = 0$ in Eq. 3.5, Fig. 3.4). Although we used a random search to adjust the regularization term, there are theoretical guarantees that this approach does not diverge significantly from a complete hyperparameter grid search (Bergstra and Bengio, 2012), suggesting that a fully regularized model could still be achieved. The variability in the hyperparameter space across datasets underscores the importance of hyperparameter tuning when using kernel-based models, as is also the case with other machine learning models that require such tuning (Krizhevsky et al., 2012; Géron, 2019).

Unregularized kernels that account for phylogenetic information have already been implemented in the context of microbiome analyses (Zhao et al., 2015; Xiao and Chen, 2017). However, these studies primarily focused on improving or accounting for non-linearity in the UniFrac distance (Lozupone and Knight, 2005), which is a useful metric for conducting association studies by incorporating patient traits to make predictions about diseases. Nevertheless, the use of kernels was limited to microbial abundance traits or presence/absence data, while the rest of the model remained linear. Therefore, since the observations were primarily represent the same species (i.e., humans), it can be considered that it is not a true phylogenetic regression, as a phylogeny is not involved in all variables in the model.

In the case of the simulated datasets, we noticed exponentially increasing performance primarily due to the RBF kernel function operating in an infinite-dimensional space. This is why we also saw high performance in the fish dataset. However, we did not observe such performance with the linear kernel function. This might be because we simulated few traits, and the linear kernel function is most effective with a large number of traits, where the relationship already exists in high dimensionality (Hsu et al., 2003; Murphy, 2012). Although the RBF kernel function converges to the linear kernel function as γ approaches 0 (Keerthi and Lin, 2003), directly employing the linear kernel function — when there is an assumption

that data are inherently high-dimensional — can be more computationally efficient. This is because there is no need for the exponentiation step to derive the complete Gram matrix.

One analytical implementation for kernels is the use of subkernels to obtain a complete kernel (Bishop and Nasrabadi, 2006). This strategy becomes particularly valuable when traits fall under different categories. For instance, in mammals, the evolution of the head bone structure may diverge from that of the rest of the body (Goswami et al., 2022), given the unique functions the head performs. However, determining the specific weights in a standard phylogenetic regression model for both head traits and rest-of-body traits presents challenges, especially when these traits interact in a non-linear manner. In such cases, the implementation of subkernels can be beneficial. For instance, we can define the following kernel form:

$$k(i,j) = k(a_{i,h}, a_{j,h}) + k(a_{i,r}, a_{j,r}) \ .$$

Here, $a_{i,h}$ represents the subset of traits from taxa observation a_i that belongs to the category h (e.g., head structures), and $a_{i,r}$ represents the subset of traits from taxa observation a_i that belongs to the category r (e.g., the rest of the bone structure). It is worth noting that the trait compositions for each subkernel do not need to be disjoint (Bishop and Nasrabadi, 2006). Moreover, there are several alternative methods to combine kernels, facilitating an analytical approach to understanding the structure of data (Bishop and Nasrabadi, 2006; Duvenaud, 2014).

The incorporation of domain knowledge into the analytical model, as outlined above, provides kernel regression with an advantage over other machine learning models, such as random forests or neural networks, which also account for non-linearities. However, when handling very large datasets, the process of inverting the Gram matrix in kernel regression (see Section 3.3.1) can become computationally demanding. In such cases, one might consider

faster implementations designed for larger problems, such as the aforementioned machine learning models. Alternatively, one could select the first m taxa observations to compute the Gram matrix. It has been observed that this method does not significantly compromise the accuracy of the solution in Kernel-based models (Williams and Seeger, 2000).

One limitation of the presented model is that it operates under the assumption that the inferred tree topology and its corresponding modeled covariance matrix are correct. However, this may not always be true if phylogenetic estimations are affected by biological processes or methodological factors (Kumar et al., 2012; Solís-Lemus and Ané, 2016; Kapli et al., 2020). Moreover, the stochastic process underlying trait evolution could potentially be misspecified (Harmon et al., 2010; Martin and Richards, 2019). In linear models, we use the phylogenetic heritability index, which estimates the degree to which related taxa share similar trait values (Lynch, 1991). This allows for adjustments to the off-diagonal values of the modeled covariance matrix (Revell, 2010). For instance, off-diagonal values of the modeled covariance matrix can be set to zero if the phylogenetic signal is close to zero. This provides a potential method to relax the rigidity of the modeled covariance matrix. An alternative approach could involve the use of a set of trees (Garamszegi, 2014; Rincon-Sandoval et al., 2020; Santaquiteria et al., 2021), allowing for the testing of a range of modeled covariance matrices and combining the model predictions using methods seen in ensemble learning approaches (Hastie et al., 2009, pp. 290). For instance, Stacked Generalization of multiple phyloKRR models where each model is fitted with different modeled covariance matrices. However, the challenge of incorporating covariance information into machine learning models and allowing it to evolve during the training process remains an open problem (but see McElreath, 2018, Chapter 14).

Another possible limitation of kernel-based models compared to PGLS is the absence of p-values for testing the significance of traits in the model. While we resorted to heuristic approaches using FI analysis and PDP (Elith et al., 2008; Hastie et al., 2009), the interpretability of complex machine learning models remains an open problem (Lundberg and

Lee, 2017; Sapoval et al., 2022).

Given that kernel regression proved effective when data were not linear, this study can serve as a baseline for future implementations of non-linear phylogenetic regressions using machine learning and complement current tools for comparative analyses. While this study only worked with a single response variable, the presented model can naturally support multi-output regression (see Supporting Information S.I.3 for more details). However, this approach predicts each response variable independently (Borchani et al., 2015). Future implementations will aim to introduce multitask learning (Alvarez et al., 2012), where multiple response variables can inform each other within a single model to make predictions.

Since machine learning models such as kernel regressions are very sensitive to the range of hyperparameters (Géron, 2019), another future extension for the current model will aim to implement techniques such as Bayesian optimization (Bergstra et al., 2013), which does not require specifying a range of hyperparameters but rather a prior distribution of hyperparameters. While out of the scope of this study, kernelized regressions can also be generalized into kernelized correlations (e.g., Soltan Ghoraie et al., 2015), where the observation matrix and the response variable matrix are used interchangeably. This can serve as a foundation for a non-linear version of canonical correlation analysis and partial least squares, which are currently used for comparative analyses (Rohlf and Corti, 2000; Adams and Collyer, 2018a).

The present study explored the use of kernels in non-linear relationships of comparative data and opens up opportunities for further improvements. However, greater efforts are required to consider data non-linearity in biological analyses using machine learning models (Moreta et al., 2021; Sapoval et al., 2022; Elhamod et al., 2023). We have encapsulated the model presented in this study in a Python package called phyloKRR, which is freely accessible on GitHub at: https://github.com/ulises-rosas/phylokrr. We have also provided a user-friendly tutorial for the package and added functions to improve the transparency of the model, including feature importance analysis (Fisher et al., 2019) and partial dependence plots (Elith et al., 2008).

Reference List

Adams, D. C., and M. L. Collyer, 2018a: Multivariate phylogenetic comparative methods: evaluations, comparisons, and recommendations. *Systematic biology*, **67 (1)**, 14–31.

Adams, D. C., and M. L. Collyer, 2018b: Phylogenetic anova: Group-clade aggregation, biological challenges, and a refined permutation procedure. *Evolution*, **72 (6)**, 1204–1215.

Adams, R., Z. Cain, R. Assis, and M. DeGiorgio, 2022: Robust phylogenetic regression. *bioRxiv*, 2022–08.

Alvarez, M. A., L. Rosasco, N. D. Lawrence, and Coauthors, 2012: Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, **4 (3)**, 195–266.

Azouri, D., S. Abadi, Y. Mansour, I. Mayrose, and T. Pupko, 2021: Harnessing machine learning to guide phylogenetic-tree search algorithms. *Nature Communications*, **12 (1)**, 1–9, https://doi.org/10.1038/s41467-021-22073-8, URL http://dx.doi.org/10.1038/s41467-021-22073-8.

Bergstra, J., and Y. Bengio, 2012: Random search for hyper-parameter optimization. *Journal of machine learning research*, **13 (2)**.

Bergstra, J., D. Yamins, and D. Cox, 2013: Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *International conference on machine learning*, PMLR, 115–123.

Bishop, C. M., and N. M. Nasrabadi, 2006: *Pattern recognition and machine learning*, Vol. 4. Springer.

Borchani, H., G. Varando, C. Bielza, and P. Larranaga, 2015: A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **5 (5)**, 216–233.

Breiman, L., 2001a: Random forests. *Machine learning*, **45**, 5–32.

Breiman, L., 2001b: Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, **16 (3)**, 199–231.

Chen, T., T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, and Coauthors, 2015: Xgboost: extreme gradient boosting. *R package version 0.4-2*, **1 (4)**, 1–4.

Cortes, C., and V. Vapnik, 1995: Support-vector networks. *Machine learning*, **20**, 273–297.

Cristianini, N., and J. Shawe-Taylor, 2000: *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Duvenaud, D., 2014: Automatic model construction with gaussian processes. Ph.D. thesis.

Elhamod, M., and Coauthors, 2023: Discovering novel biological traits from images using phylogeny-guided neural networks. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery (ACM), 3966–3978.

Elith, J., J. R. Leathwick, and T. Hastie, 2008: A working guide to boosted regression trees. *Journal of animal ecology*, **77 (4)**, 802–813.

Felsenstein, J., 1985: Phylogenies and the comparative method. *The American Naturalist*, **125 (1)**, 1–15.

Fisher, A., C. Rudin, and F. Dominici, 2019: All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.*, **20 (177)**, 1–81.

Garamszegi, L. Z., 2014: *Modern phylogenetic comparative methods and their application in evolutionary biology: concepts and practice*. Springer.

Géron, A., 2019: *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Goswami, A., and Coauthors, 2022: Attenuated evolution of mammals through the cenozoic. *Science*, **378 (6618)**, 377–383.

Harmon, L., 2019: *Phylogenetic comparative methods: learning from trees*. EcoEvoRxiv.

Harmon, L. J., and Coauthors, 2010: Early bursts of body size and shape evolution are rare in comparative data. *Evolution*, **64 (8)**, 2385–2396, https://doi.org/10.1111/j.1558-5646.2010.01025.x.

Harris, C. R., and Coauthors, 2020: Array programming with numpy. *Nature*, **585 (7825)**, 357–362.

Harvey, P. H., M. D. Pagel, and Coauthors, 1991: *The comparative method in evolutionary biology*, Vol. 239. Oxford university press Oxford.

Hastie, T., R. Tibshirani, J. H. Friedman, and J. H. Friedman, 2009: *The elements of statistical learning: data mining, inference, and prediction*, Vol. 2. Springer.

Hsu, C.-W., C.-C. Chang, C.-J. Lin, and Coauthors, 2003: A practical guide to support

vector classification. Taipei, Taiwan.

Ives, A. R., and T. Garland Jr, 2010: Phylogenetic logistic regression for binary dependent variables. *Systematic biology*, **59 (1)**, 9–26.

Jiang, Y., P. Tabaghi, and S. Mirarab, 2022: Learning Hyperbolic Embedding for Phylogenetic Tree Placement and Updates. *Biology*, **11 (9)**, 1–24, https://doi.org/10.3390/biology11091256.

Kapli, P., Z. Yang, and M. J. Telford, 2020: Phylogenetic tree building in the genomic age. *Nature Reviews Genetics*, **21 (7)**, 428–444, https://doi.org/10.1038/s41576-020-0233-0, URL http://dx.doi.org/10.1038/s41576-020-0233-0.

Keerthi, S. S., and C.-J. Lin, 2003: Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, **15 (7)**, 1667–1689.

Kirk, E. C., and R. F. Kay, 2004: The evolution of high visual acuity in the anthropoidea. *Anthropoid origins: new visions*, Springer, 539–602.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012: Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, **25**.

Kumar, S., A. J. Filipski, F. U. Battistuzzi, S. L. Kosakovsky Pond, and K. Tamura, 2012: Statistics and truth in phylogenomics. *Molecular Biology and Evolution*, **29 (2)**, 457–472, https://doi.org/10.1093/molbev/msr202.

LeCun, Y., Y. Bengio, and G. Hinton, 2015: Deep learning. *nature*, **521 (7553)**, 436–444.

LeCun, Y., L. Bottou, G. B. Orr, and K.-R. Müller, 2002: Efficient backprop. *Neural networks: Tricks of the trade*, Springer, 9–50.

Lefebvre, M., 2007: *Applied stochastic processes*. Springer Science & Business Media.

Lozupone, C., and R. Knight, 2005: Unifrac: a new phylogenetic method for comparing microbial communities. *Applied and environmental microbiology*, **71 (12)**, 8228–8235.

Lundberg, S. M., and S.-I. Lee, 2017: A unified approach to interpreting model predictions. *Proceedings of the 31st international conference on neural information processing systems*, 4768–4777.

Lynch, M., 1991: Methods for the analysis of comparative data in evolutionary biology. *Evolution*, **45 (5)**, 1065–1080.

Martin, C. H., and E. J. Richards, 2019: The Paradox behind the Pattern of Rapid Adaptive Radiation: How Can the Speciation Process Sustain Itself Through an Early Burst?

Annual Review of Ecology, Evolution, and Systematics, **50**, 569–593, https://doi.org/10.1146/annurev-ecolsys-110617-062443.

Martins, E. P., and T. F. Hansen, 1997: Phylogenies and the comparative method: a general approach to incorporating phylogenetic information into the analysis of interspecific data. *The American Naturalist*, **149 (4)**, 646–667.

McElreath, R., 2018: *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC.

Moreta, L. S., O. Rønning, A. S. Al-Sibahi, J. Hein, D. Theobald, and T. Hamelryck, 2021: Ancestral protein sequence reconstruction using a tree-structured ornstein-uhlenbeck variational autoencoder. *International Conference on Learning Representations*, International Conference on Learning Representations (ICLR).

Murphy, K. P., 2012: *Machine learning: a probabilistic perspective*. MIT press.

Paradis, E., and J. Claude, 2002: Analysis of comparative data using generalized estimating equations. *Journal of theoretical biology*, **218 (2)**, 175–185.

Paradis, E., J. Claude, and K. Strimmer, 2004: Ape: analyses of phylogenetics and evolution in r language. *Bioinformatics*, **20 (2)**, 289–290.

Quader, S., K. Isvaran, R. Hale, B. Miner, and N. Seavy, 2004: Nonlinear relationships and phylogenetically independent contrasts. *Journal of Evolutionary Biology*, **17 (3)**, 709–715.

Revell, L. J., 2010: Phylogenetic signal and linear regression on species data. *Methods in Ecology and Evolution*, **1 (4)**, 319–329, https://doi.org/10.1111/j.2041-210x.2010.00044.x.

Rincon-Sandoval, M., and Coauthors, 2020: Evolutionary determinism and convergence associated with water-column transitions in marine fishes. *Proceedings of the National Academy of Sciences*, **117 (52)**, 33 396–33 403.

Rohlf, F. J., and M. Corti, 2000: Use of two-block partial least-squares to study covariation in shape. *Systematic biology*, **49 (4)**, 740–753.

Ruder, S., 2016: An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Santaquiteria, A., and Coauthors, 2021: Phylogenomics and historical biogeography of seahorses, dragonets, goatfishes, and allies (teleostei: Syngnatharia): assessing factors driving uncertainty in biogeographic inferences. *Systematic Biology*, **70 (6)**, 1145–1162.

Sapoval, N., and Coauthors, 2022: Current progress and open challenges for applying deep learning across the biosciences. *Nature Communications*, **13 (1)**, 1–12, https://doi.org/

10.1038/s41467-022-29268-7.

Solís-Lemus, C., and C. Ané, 2016: Inferring Phylogenetic Networks with Maximum Pseudolikelihood under Incomplete Lineage Sorting. *PLoS Genetics*, **12 (3)**, 1–21, https://doi.org/10.1371/journal.pgen.1005896, 1509.06075.

Soltan Ghoraie, L., F. Burkowski, and M. Zhu, 2015: Using kernelized partial canonical correlation analysis to study directly coupled side chains and allostery in small g proteins. *Bioinformatics*, **31 (12)**, i124–i132.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014: Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, **15**, 1929–1958.

Tibshirani, R., 1996: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **58 (1)**, 267–288.

Tung Ho, L. s., and C. Ané, 2014: A linear-time algorithm for gaussian and non-gaussian trait evolution models. *Systematic biology*, **63 (3)**, 397–408.

Vovk, V., 2013: Kernel ridge regression. *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*, Springer, 105–116.

Voznica, J., A. Zhukova, V. Boskova, E. Saulnier, F. Lemoine, M. Moslonka-Lefebvre, and O. Gascuel, 2022: Deep learning from phylogenies to uncover the epidemiological dynamics of outbreaks. *Nature Communications*, **13 (1)**, 3896.

Wang, H., 2022: A note on centering in subsample selection for linear regression. *Stat*, **11 (1)**, e525.

Williams, C., and M. Seeger, 2000: Using the nyström method to speed up kernel machines. *Advances in neural information processing systems*, **13**.

Xiao, J., and J. Chen, 2017: Phylogeny-based kernels with application to microbiome association studies. *New Advances in Statistics and Data Science*, 217–237.

Zhang, C., S. Bengio, M. Hardt, B. Recht, and O. Vinyals, 2021: Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, **64 (3)**, 107–115.

Zhang, Z., S. Cheng, and C. Solís-Lemus, 2020: Towards a robust out-of-the-box neural network model for genomic data. *arXiv*, 1–27, 2012.05995.

Zhao, N., and Coauthors, 2015: Testing in microbiome-profiling studies with mirkat, the microbiome regression-based kernel association test. *The American Journal of Human*

Genetics, **96 (5)**, 797–807.

Chapter 4

Dissecting factors underlying phylogenetic uncertainty using machine learning models

To be submitted.

Ulises Rosas-Puchuri, Emanuell Duarte-Ribeiro, Sina Khanmohammadi, Dahiana Arcila, Guillermo Ortí, Ricardo Betancur-R.

4.1 Abstract

Phylogenetic inference is shaped by both biological processes and methodological conditions. While biological processes can be explicitly modeled, phylogenetic models often implicitly assume that methodological factors exert little influence on inference. When substantial, however, these factors can introduce inconsistency and uncertainty into tree reconstruction. Previous attempts to identify their origins have relied on correlations between data features and support for conflicting tree topologies, but such approaches typically analyze predictors in isolation or assume strictly linear relationships among them. Here we present a machine-learning framework that leverages the increasing scale of phylogenomic datasets to model complex, non-linear relationships between gene features and support for competing phylogenetic hypotheses. We analyzed two phylogenomic datasets of teleost fishes: a

newly generated dataset focusing on protacanthopterygians (including salmonids, galaxiids, marine smelts, and relatives) and a reanalysis of an existing dataset of carangarians (flatfishes and allied lineages). Five supervised machine-learning algorithms were evaluated and all performed significantly better than a linear model (p ¡ 0.05), with a deep neural network (DNN) providing the best predictive performance across both empirical datasets. Feature-importance analyses revealed that the most influential predictors differed between datasets, highlighting the context-dependent nature of methodological effects in phylogenomic inference. Using simulated datasets in which the source of topological discordance was controlled, DNN-based feature-importance analyses successfully identified the manipulated factor responsible for variation in topology support. The insights generated by this framework can improve downstream phylogenetic analyses, for example by informing the choice of substitution models and data-processing strategies. Together, our results establish a general approach for identifying non-linear interactions among gene features that influence phylogenetic support. This framework complements existing phylogenetic tools and provides a new avenue for investigating the methodological drivers of discordance in large-scale phylogenomic datasets.

4.2 Introduction

The advent of genomic datasets has further improved our capacity to consistently elucidate sections of the tree of life previously unresolved. However, due to underlying biological processes and methodological or analytical factors, phylogenetic inference can yield incongruent or inconsistent tree topologies even using genomic-scale datasets (Steenwyk et al., 2023). Biological processes that produce misleading phylogenetic results —if not accounted for— include incomplete lineage sorting, introgression, paralogy and natural selection (Maddison, 1997; De Queiroz, 2007; Kapli et al., 2020). While current methodologies allow for the exploration of certain biological processes, there remain limitations in their comprehen-

sive application. For instance, incomplete lineage sorting (ILS) can be approached through statistical consistency with the multispecies coalescent (MSC) model (Zhang et al., 2018). Additionally, reticulations in phylogenetic trees, potentially indicative of introgression, gene flow, or hybridization, can be explored using a pseudo-likelihood score (Solís-Lemus and Ané, 2016). However, in practice, jointly modeling all these processes (e.g., ILS, introgression, paralogy) in phylogenomic datasets is currently unfeasible. While methods exist to address these individual processes (Simão et al., 2015; Zhang et al., 2018), they typically do not account for the potential interplay of analytical factors that could contribute to estimation errors. In reality, the outcomes of phylogenomic analyses often arise from a combination of discordances originating both from biological processes and analytical factors (Cai et al., 2020).

Analytical factors including properties or features of the data vary and impact phylogenetic reconstructions in distinct ways (Kumar et al., 2012; Betancur et al., 2013; Susko and Roger, 2021). Missing taxa provides a relevant example, as studies have reported that increasing taxon sampling can enhance inference stability (Heath et al., 2008; Betancur-R et al., 2019; Singhal et al., 2021). But there are also cases where specific taxa consistently destabilize the inference (Aberer et al., 2013). Additionally, datasets with an insufficient number of sites can produce low-quality gene trees, which can introduce subsequent errors when inferring the species tree from them (Springer and Gatesy, 2016; Molloy and Warnow, 2018). Misspecification of the evolutionary model is another factor that can introduce bias. This problem is particularly pronounced in the presence of a large number of sites, a typical feature of current genomic datasets (Kumar et al., 2012; Doyle et al., 2015). At such large scales, the variance is reduced, which can misleadingly increase confidence in an incorrect phylogenetic inference. While the above-mentioned sources of incongruence can be addressed individually, proposed corrections often require computationally demanding approaches or more sampling efforts (Xi et al., 2016; Kapli et al., 2020). For example, it might be necessary to increase the sequencing budget to obtain more markers, increase the

sampling effort to collect missing species, or employ more thorough runs for model-selection software. Additionally, analytical factors may covary or interact with one another (Shen et al., 2016; Mongiardino Koch, 2021), which adds complexity to the decision-making process about which factors to account for. Therefore, understanding these factors beforehand we can make more targeted decisions about the necessary corrections, rather than attempting a large and computationally intensive array of them, an approach that also aligns with green computing principles (Kumar, 2022).

To understand how these factors affect phylogenetic inference within a dataset, we typically examine associations between data features and support metrics (e.g., likelihood scores, bootstrap support, topology test p-values) for specific phylogenetic hypotheses (Shen et al., 2021; Steenwyk et al., 2023). These associations help researchers assess the relative importance of different factors in generating phylogenetic incongruence and guide the design of future corrections to reduce bias and error. However, studies using these approaches often assume that features are independent (Gosselin et al., 2021; Cunha et al., 2022; Smith et al., 2023) or linearly related (Vankan et al., 2022). Although linearity can be a useful starting point, it is often biologically unrealistic (Strogatz, 2015), as many biological features show strong non-linear interactions. Thus, for phylogenomic datasets, a better strategy is to use methods that both relax linearity assumptions and explicitly model feature interactions (Azouri et al., 2021; Nguyen et al., 2024). At present, such methods are lacking because it is difficult to construct and parameterize them from first principles without detailed knowledge of the theoretical distributions of these features and their interactions (Susko and Roger, 2021). There is therefore a need for non-linear models that do not depend on prior assumptions about feature distributions and interactions.

With the current emergence of large phylogenomic datasets containing thousands of markers (Harvey et al., 2020; Álvarez-Carretero et al., 2022), we now have the opportunity to extract a substantial number of feature instances from each marker, including alignment information (e.g., GC-content, number of taxa) and tree information (e.g., average boot-

strap support, sum of branch lengths). This presents an unprecedented opportunity to train machine learning models that can potentially learn complex non-linear relationships within the phylogenomic dataset without imposing a fixed set of probability distribution families to construct the model (i.e., non-parametric models) (Hastie et al., 2009; Murphy, 2012; Bokulich et al., 2018). Machine learning models, capable of identifying complex non-linear relationships, have found applications in various phylogenetic analyses, including phylogenetic tree search (Azouri et al., 2021, 2024), phylogenetic placement (Jiang et al., 2022; Rachtman et al., 2025), learning phylodynamic parameters (Voznica et al., 2022), among others. However, whereas most existing work emphasizes complex predictive tasks, systematic analyses of relevant features for predicting competing phylogenetic hypotheses remains largely unexplored.

We propose using machine learning models to relate features—factors known to introduce bias and error in phylogenomic inference—to the phylogenetic signal within a dataset. This approach allows features to interact non-linearly, potentially revealing insights from the trained models. Because modern phylogenomic datasets often contain hundreds or thousands of gene observations, they provide many feature instances for effective model training. Several methods can estimate the relative support each gene provides for alternative phylogenetic hypotheses, including likelihood scores (Shen et al., 2017), concordance factors (Minh et al., 2020), and quartet scores (Zhang et al., 2018; Shen et al., 2021). Here, we use Gene-Genealogy Interrogation (GGI) (Arcila et al., 2017) to quantify the relative support for competing hypotheses using p-values derived from site log-likelihood distributions (Shimodaira, 2002). These values are used as labels for gene features. Although machine learning models can be complex and often viewed as "black boxes," their interpretability can be improved through feature importance analyses (Breiman, 2001; Lundberg and Lee, 2017), which evaluate how permuting each feature affects overall model performance.

Supervised learning models, although non-parametric, can learn phylogenetic signal from a dataset (Nguyen et al., 2024; Albors et al., 2025) and differ in their error-minimization

strategies. For example, Support Vector Machines (Cortes and Vapnik, 1995) fit a linear function after transforming the feature space into a potentially infinite-dimensional one. Random Forest and Gradient Boosting (Breiman, 2001; Chen and Guestrin, 2016) combine multiple decision trees for prediction. Deep Neural Networks (LeCun et al., 2015) and related architectures, such as autoencoders, use layered networks with weighted connections, enabling increasingly abstract data representations as the number of hidden layers increases (Goodfellow et al., 2016). Their flexibility allows tuning multiple hyperparameters to create architectures tailored to a specific dataset.

We evaluated multiple supervised machine learning models and identified the most influential features affecting support predictions for alternative phylogenetic hypotheses using both empirical and simulated datasets. The empirical dataset includes newly generated data from "protacanthopterygian" fishes. Protacanthopterygii is a deep-branching euteleost lineage proposed 57 years ago (Greenwood et al., 1966) whose circumscription has changed at least 13 times (Ishiguro et al., 2003; Betancur et al., 2017), yet previous studies have lacked comprehensive species coverage or genome-wide sampling. The second dataset comprises fishes in the series Carangaria (i.e., flatfishes, billfishes, remoras, and allies; Duarte-Ribeiro et al., 2024), which we reanalyzed here. The most contentious relationship in Carangaria concerns the monophyly of flatfishes and whether their asymmetric body plan evolved once or twice (Betancur et al., 2013; Campbell et al., 2013; Lü et al., 2021). For simulated datasets, we used AliSim (Ly-Trong et al., 2022) to generate sequence data under controlled conditions. Our primary goal is to use machine learning to explore dataset-specific features contributing to phylogenetic uncertainty, accounting for the non-linearity inherent in phylogenomic data rather than selecting the best-supported hypothesis.

4.3 Materials and methods

4.3.1 Empirical Datasets

We generated a new dataset for Protacanthopterygii (77 species and 1023 exons) and reanalyzed a recently generated dataset for Carangaria (395 species and 991 exons; Duarte-Ribeiro et al., 2024), both of which are part of a large initiative—the FishLife project—aimed at estimating phylogenomic trees for over six thousand fish species based on exon capture approaches. Ever since the pre-cladistic definition of Protacanthopterygii by (Greenwood et al., 1966), this clade was conceived as encompassing various taxa deemed as "precursors" of acanthopterygians. Although this concept is no longer supported, numerous alterations to its membership and interrelationships have been suggested over the past 57 years, totaling at least 13 different proposed changes. A major problem with previous studies is that they have not adequately dealt with both comprehensive species coverage and genome-wide genetic sampling. We chose the Protacanthopterygii dataset with the goal of analyzing phylogenomic evidence to assess support for alternative hypotheses regarding the relationships within this group and among its close allies. Our objective is also to pinpoint variables that could provide explanations for, or enhance the resolution of, the persistently debated deep-branching euteleost lineages (e.g., Hughes et al., 2018). We also obtained publicly available sequences to expand the taxonomic coverage for the Protacanthopterygii dataset to 123 species (see Table C1). We chose the Carangaria dataset for reanalysis here given that previous studies have indicated that base-composition heterogeneity is a major factor affecting exonic datasets generated for this clade (Betancur et al., 2013; Ribeiro et al., 2018), with resulting phylogenetic analyses often (though not always) failing to resolve the monophyly of flatfishes, one of the largest clades in Carangaria. Such conflicting results have important evolutionary implications as they may either imply that the flatfish asymmetric body plan has a single (see Betancur et al., 2013; Harrington et al., 2016) or dual (see Campbell et al.,

2013; Lü et al., 2021) origin.

4.3.2 Exon Capture, Assembly, and Quality Control

We extracted high-yield DNA for 62 Protacanthopterygii species (orders Salmoniformes, Esociformes, Argentiniformes, Osmeriformes, Stomiatiformes, Galaxiiformes) and 15 outgroup species (orders Lepidogalaxiiformes, Alepocephaliformes, Clupeiformes, Neoteleostei). We generated enriched genomic libraries with the "backbone 1" and "backbone 2" probe sets of Hughes et al. (2021). Arbor Biosciences (Ann Arbor, MI) prepared libraries using a dual-round capture protocol (Li et al., 2013), and the University of Chicago Genomics Facility sequenced them on a HiSeq4000 to obtain paired-end 150 bp reads, multiplexing 190 samples per lane.

We filtered out reads with adaptor contamination (> 15 bp matching adaptor), > 10% low-quality bases (score < 10), or > 2% unknown bases, then added publicly available sequences. We assembled all filtered reads with an optimized FishLifeExonCapture pipeline (`https://github.com/ulises-rosas/FishLifeExonCapture`; Hughes et al., 2021). This pipeline uses Trimmomatic v0.36 (Bolger et al., 2014) to trim FASTQ files, BWA-MEM (Li and Durbin, 2009) to map reads to references, SAMtools (Li and Durbin, 2009) to remove PCR duplicates, and aTRAM (Allen et al., 2017) to extend mapped reads after initial assembly with Velvet (Zerbino and Birney, 2008) and final assembly with Trinity (Haas et al., 2013). CD-HIT (Fu et al., 2012) identifies identical sequences, and exonerate (Slater and Birney, 2005) defines open reading frames for all target exons. MACSE (Ranwez et al., 2011) then aligns exon sequences while accounting for codon structure. The final Protacanthopterygii dataset contains 1133 exon alignments for 123 species (see Results).

The Carangaria dataset was compiled in Duarte-Ribeiro et al. (2024) following the same protocol. It consists of 991 exons from 395 species, of which 12 are non-carangarian outgroups. Following assembly, we applied a quality-control pipeline to both datasets to identify contamination, mislabeling, sequencing errors, and other irregularities, adopting the frame-

work of Arcila et al. (2021) for assessing missing data, taxon mislabeling, and taxon misplacement. These procedures were implemented in the Python command-line tool "`fishlifeqc`" (`https://github.com/ulises-rosas/fishlifeqc`); see Supplemental Information C.1 for further details.

4.3.3 Phylogenomic Inference

All downstream analyses applied to both datasets are based on trees estimated using both concatenation and multispecies coalescent approaches. However, for this study we only inferred phylogenetic trees for the Protacanthopterygii dataset, and instead reused trees for the Carangaria dataset from Duarte-Ribeiro et al. (2024) study, which were estimated following similar procedures described below.

We used IQ-TREE v2.0.6 (Nguyen et al., 2015) to estimate gene trees and concatenation-based phylogenies under the maximum likelihood (concatenation-based ML hereafter) approach. We first generated concatenated gene alignments using `concatenate.py` (a utility of "fishlifeqc") for Protacanthopterygii, and then used ModelPartition (Kalyaanamoorthy et al., 2017) to select the best-fit partitioning scheme (using the `-p` option) based on by-gene and by-codon a priori partitions. This step generated 258 partitions, including best-fit evolutionary models for each partition. Concatenation-based analyses used the following IQ-TREE options: `iqtree2 -s [alignment file] -p [partition file] -T [Number of cores] --ufboot 1000`. We then used Ultrafast Bootstrap (Hoang et al., 2018) to assess edge support (set to 1000 using the `--ufboot 1000` option).

Using the same options in IQ-TREE without the `-p` argument, we generated gene trees as input for multispecies coalescent (MSC hereafter) analyses as implemented in ASTRAL-III (Zhang et al., 2018). To minimize artifacts caused by gene tree estimation error, we collapsed edges with low support (aLRT = 0, Anisimova and Gascuel, 2006), following recommendations by Simmons and Gatesy (2021). As a measure of support, we let ASTRAL-III to use the default setting, that is local posterior, to assess the edge support. Each dataset pro-

duced two major alternative hypotheses, one under the ML approach and the other under the multispecies coalescent approach. These competing hypotheses differed in key aspects of the evolutionary history of each clade (see details under Results, Fig. 4.1, and Fig. C1). For instance, the Protacanthopterygii trees differed in the relative placement of the order Argentiniformes, and the Carangaria dataset on the monophyletic status of flatfishes (comprised by suborders Psettodoidei and Pleuronectoidei; Fig. 4.1). Subsequent analyses assess the relative support that each competing hypotheses receive from individual exon alignments.

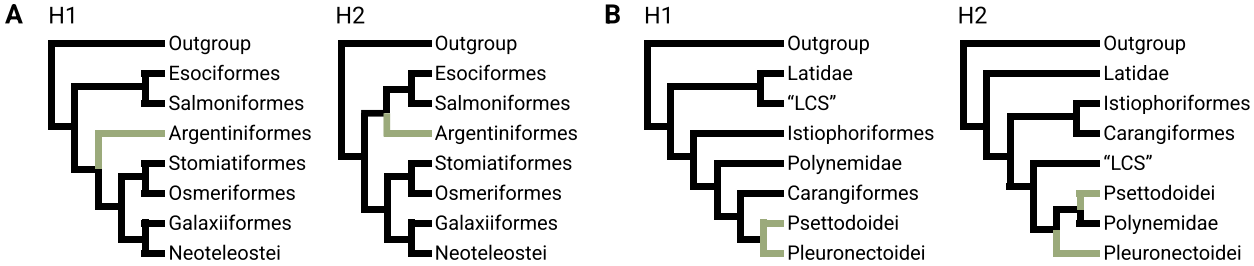


Figure 4.1: Phylogenetic trees obtained for Protacantopterygii (**A**) and Carangaria (**B**). In **A**), H1 is the concatenation-based ML tree and H2 the MSC tree (also see Results and Fig. 4.3). These trees differ in the placement of Argentiniformes (highlighted with green). In **B**), H1 is the MSC tree and H2 the concatenation-based ML tree (see Fig. C1). While multiple differences exist between these two alternative trees, a major discrepancy comprises the monophyletic status of flatfishes which includes the suborders Psettodoidei and Pleuronectoidei. In both hypotheses, these suborders are highlighted in green. The acronym "LCS" designates the clade consisting of Lactariidae, Centropomidae, and Sphyraenidae.

4.3.4 Gene-Genealogy Interrogation (GGI)

We applied GGI (Arcila et al., 2017) to quantify the amount of phylogenetic signal in the loci supporting each competing hypothesis. Briefly, the GGI procedure includes defining the competing hypotheses; building constraint trees by pruning the taxon set to match those in the individual alignments; estimating constrained ML gene trees under the GTR+GAMMA model in RAxML v8.2.12 (Stamatakis, 2014); estimating site log-likelihoods (SLL) for each constrained gene tree; and conducting AU tests using the SLL values as input to assess the relative support that each hypothesis receives from individual genes. We ran each constrained tree separately to ensure independent tree-parameter optimization—and therefore

independent SLL estimates—within each RAxML run. These steps were integrated into a Python-based command line interface called "ggpy" (available from `https://github.com/ulises-rosas/ggpy`). The output of each AU test from above provides a rank of preferences for the alternative hypotheses for each exon, based on their likelihood score.

For the Protacanthopterygii dataset, we evaluated two hypotheses from the previous step (concatenation-based ML and MSC trees) plus 103 additional rooted topologies in which Lepidogalaxiiformes, Alepocephaliformes, and Clupeiformes formed the outgroup, totaling 105 trees representing all possible relationships among five well-supported subclades (Table C2): (1) Esociformes + Salmoniformes ("Eso-Salmo"), (2) Osmeriformes + Stomiatiformes ("Osme-Stomia"), (3) Neoteleostei, (4) Galaxiiformes, and (5) Argentiniformes. GGI analyses (Fig. C5) identified the same two top trees favored by concatenation-based ML and MSC inference, which we treated as the main hypotheses. However, only 114 and 55 loci achieved rank 1 (highest AU p-value) for these trees. To increase the statistical power of AU tests for these two hypotheses, we then ran GGI using only these topologies, following Shen et al. (2017). The Carangaria dataset includes all 30 families in this radiation (Ribeiro et al., 2018; Duarte-Ribeiro et al., 2024), implying $\approx 4.95 \times 10^{38}$ possible interfamilial topologies; because GGI over all such trees is infeasible, we restricted analyses to the two best-supported phylogenetic hypotheses. For both datasets, GGI was thus performed on each gene alignment under two competing hypotheses—the concatenation-based ML and MSC species trees—yielding two AU p-values per exon, which were then used as labels for the machine learning models.

4.3.5 Machine Learning Models

To examine the relationship between phylogenetic signal for a specific species tree and exon alignment characteristics, we used five supervised machine learning models for multi-output regression analyses. For each exon alignment, the predictors included 40 features relevant to phylogenetic inference (Table 4.1), and the response consisted of two labels corresponding

to the AU p-values.

More generally, if an exon alignment i has two AU p-values, $p_{i,1}$ and $p_{i,2}$, for the first and second hypotheses, respectively, then the label for this exon alignment i is the row vector $\begin{bmatrix} p_{i,1} & p_{i,2} \end{bmatrix}$. The predictor matrix was derived from gene alignments and gene trees and incorporated 40 previously proposed properties considered relevant for phylogenetic inference (Shen et al., 2016; Mongiardino Koch, 2021; Haag et al., 2022). Figures S3 and S4 show the distribution of these features across both datasets. Several studies have identified these features as being associated with factors that introduce uncertainty into phylogenetic inference. For instance, high variance in GC content may indicate violations of model assumptions (Kolaczkowski and Thomton, 2004; Kapli et al., 2020), whereas the number of variable sites is related to the noise-to-signal ratio (Zhang et al., 2021; Di Franco et al., 2022). Although no consensus exists on the optimal feature set for machine learning in phylogenetics, supervised learning algorithms can adjust feature importance through regularization, provided the hyperparameters controlling regularization strength are known. Below, we describe how these hyperparameters were determined. A comprehensive feature set is therefore advantageous, as the supervised models should be able to identify the most informative patterns, which can subsequently be examined through feature-importance analyses.

Each dataset consisted of one matrix of predictors and one of labels. We split each pair of matrices into training and testing sets comprising 66% and 33% of the rows, respectively (Fig. 4.2A). The training sets were used for hyperparameter tuning via cross-validation (CV), whereas the testing sets were used solely for model evaluation. Because each exon has two labels, all models were implemented as multi-output regressions, with error measured using the Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{2n} \sum_{j=1}^{2} \sum_{i=1}^{n} (\hat{y}_{i,j} - y_{i,j})^2,$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{n \times 2}$ and $\mathbf{y} \in \mathbb{R}^{n \times 2}$ are the predicted and observed values, respectively, and

n is the number of observations, in our case, the number of exon alignments.

Although all models are non-parametric, they differ in their optimization strategies. Support Vector Machines (SVM; Cortes and Vapnik, 1995), for example, assume a single optimal solution under convex optimization when the feature space is mapped to a high-dimensional space. In contrast, regression-tree and neural-network methods can accommodate multiple local optima (i.e., non-convex optimization). Examples of regression-tree models include Random Forest (RF; Breiman, 2001) and Gradient Boosting (GB; Chen and Guestrin, 2016). RF uses bootstrapped samples to build an ensemble of regression trees (Breiman et al., 1984) and averages their predictions, whereas GB sequentially fits weak learners (i.e., shallow regression trees) while accounting for residuals from previous learners Hastie et al. (2009).

Neural-network-based models comprise many architectures, including those used for text and image classification, most relying on Deep Neural Networks (DNN; LeCun et al., 2015; Goodfellow et al., 2016). The simplest DNN architecture is a fully connected input hidden-output structure. The edges are weighted and randomly initialized, each input node represents an exon feature, and nodes in subsequent layers apply a nonlinear activation function to a weighted sum plus bias (Géron, 2022). The stochastic gradient descent algorithm (LeCun et al., 1989) updates weights by minimizing an error function between the output layer values and observed labels.

A variation of this architecture is the Variational Autoencoder (VAE; Kingma and Welling, 2013), which learns low-dimensional embeddings of noisy data. Its symmetric, hourglass-shaped structure consists of an encoder, which maps inputs to a latent representation at the central layer, and a decoder, which reconstructs the inputs while incorporating Gaussian noise. Because the encoder learns low-level structure, its layers can be transferred to a new DNN for supervised learning—a process known as transfer learning (Géron, 2022).

We used the Scikit-Learn v1.5.1 (Pedregosa et al., 2011) to implement the SVM, RF, and GB models (Fig. 4.2B) and used the `RandomizedSearchCV` function for hyperparameter tuning. Two rounds of cross-validation were performed for each hyperparameter combina-

tion. The Linear Model (LM) did not require tuning because it has a closed-form solution via least squares.

Most tuned hyperparameters directly affected overfitting. For SVMs, we tuned the C parameter ($1 - 30$; reciprocal distribution), which controls the width of the margin, and the gamma parameter from the RBF kernel (scale = 1; exponential distribution), which controls the importance of higher-order terms. For RF, we tuned the maximum tree depth ($5 - 100$), the minimum sample size for splitting nodes ($2 - 50$), and the maximum number of leaf nodes ($100 - 250$). For GB, implemented via XGBoost (Chen and Guestrin, 2016), we tuned maximum tree depth ($3 - 15$), subsampled feature proportion ($0.1 - 1$), gamma values ($1 - 10$), minimum leaf weight ($0.1 - 10$), and learning rate ($10^{-7} - 0.9$).

We implemented the DNN and the VAE-based transfer learning architecture (DNN + Encoder) in TensorFlow v2.18.0 using the Keras API (Abadi et al., 2016). Hyperparameters were tuned using the `BayesianOptimization` function from the KerasTuner v1.4.7 (O'Malley et al., 2019). Tuned hyperparameters included the number of hidden layers ($4-10$), number of neurons per layer ($5-100$), dropout rate per layer ($10^{-4}-0.9$), learning rate ($10^{-4}-0.1$), and learning-rate decay ($10^{-7} - 0.9$). Fixed hyperparameters included the SELU non-linear activation function (Klambauer et al., 2017), LeCun weight initialization (Glorot and Bengio, 2010), and the Nesterov Accelerated Gradient optimizer (Nesterov, 1983). A complete list of selected hyperparameters for each dataset is provided in the Supplemental Material.

For DNN + Encoder, we used a transfer-learning strategy. Specifically, we trained a VAE for 100,000 iterations with an architecture descending from 40 units (input layer) to 20 units (central layer) and symmetrically ascending back to 40 units, with steps of 5 units between adjacent hidden layers. The central layer used a Gaussian sampling size of 10, with mean and variance updated each iteration, and optimization used RMSProp (Hinton et al., 2012). The VAE achieved an MSE below 0.16. We then transferred the encoder layers into a new DNN architecture (Fig. 4.2B), froze their weights, and tuned only the additional hidden layers as described above.

We selected the best model for each dataset by evaluating the average MSE across 20 rounds of cross-validation on different subsets of the data (Fig. 4.2C). For each subset, the training–testing split was 66% and 33%, respectively.

Table 4.1: List of features based on alignment and tree information used to train machine learning models.

Feature	**Description**
nheaders	Number of taxa in the alignment. Used in Azouri et al. (2024).
pis	Percentage of parsimony-informative sites. Used in Mongiardino Koch (2021).
vars	Percentage of variable sites. Used in Mongiardino Koch (2021).
seq_len	Number of sites in the alignment. Used in Haag et al. (2022).
seq_len_nogap	Percentage of sites without gaps. Used in Shen et al. (2016).
gap_prop	Proportion of gap characters in the alignment matrix. Used in Molloy and Warnow (2018).
nogap_prop	Proportion of non-gap characters in the alignment matrix. Used in Cunha et al. (2022).
gc_mean	Mean GC-content per sequence. Used in Betancur et al. (2013).
gc_var	Variance of GC-content per sequence. Used in Shen et al. (2016).
gap_mean	Mean gap percentage per sequence. Used in Shen et al. (2016).
gap_var	Variance of gap percentage per sequence. Used in Shen et al. (2016).
pi_mean	Mean pairwise identity. Used in Gosselin et al. (2021).
pi_std	Standard deviation of pairwise identity. Used in Shen et al. (2016).
entropy	Average Shannon entropy across all alignment columns. Used in Haag et al. (2022).

Continued on next page

Table 4.1 – *Continued from previous page*

Feature	**Description**
singletons	Number of sites that have a nucleotide that occurs multiple times while the rest only once. Used in Shen et al. (2016).
invariants	Percentage of sites containing the same nucleotide in all sequences. Used in Haag et al. (2022).
pattern	Number of unique site patterns across all columns. Used in Haag et al. (2022).
total_tree_len	Sum of all branch lengths in the tree. Used in Vankan et al. (2021).
treeness	Sum of internal branch lengths divided by `total_tree_len` (as described by Phillips and Penny (2003)). Used in Vankan et al. (2021).
inter_len_mean	Mean internal branch length in the tree. Used in Shen et al. (2016).
inter_len_var	Variance of internal branch lengths in the tree. Used in Shen et al. (2016).
ter_len_mean	Mean terminal branch length in the tree. Used in Shen et al. (2016).
ter_len_var	Variance of terminal branch lengths in the tree. Used in Shen et al. (2016).
supp_mean	Mean node support values in the tree. Used in Azouri et al. (2024).
rcv	Relative composition variability (as described by Phillips and Penny (2003)). Used in Shen et al. (2016).
treeness_o_rcv	`treeness` divided by `rcv` (as described by Phillips and Penny (2003)). Used in Shen et al. (2016).
saturation	Slope of the correlation between substitution distance and sequence pair distance (as described by Philippe et al. (2011)). Used in Duchêne et al. (2021).
LB_std	Standard deviation of the long-branch score per sequence (as described by Struck (2014)).

Continued on next page

Table 4.1 – *Continued from previous page*

Feature	**Description**
gc_mean_pos1	Mean GC-content per sequence at codon position 1. Used in Shen et al. (2016).
gc_var_pos1	Variance of GC-content per sequence at codon position 1. Used in Shen et al. (2016).
gap_mean_pos1	Mean gap percentage per sequence at codon position 1. Used in Shen et al. (2016).
gap_var_pos1	Variance of gap percentage per sequence at codon position 1. Used in Shen et al. (2016).
gc_mean_pos2	Mean GC-content per sequence at codon position 2. Used in Shen et al. (2016).
gc_var_pos2	Variance of GC-content per sequence at codon position 2. Used in Shen et al. (2016).
gap_mean_pos2	Mean gap percentage per sequence at codon position 2. Used in Shen et al. (2016).
gap_var_pos2	Variance of gap percentage per sequence at codon position 2. Used in Shen et al. (2016).
gc_mean_pos3	Mean GC-content per sequence at codon position 3. Used in Shen et al. (2016).
gc_var_pos3	Variance of GC-content per sequence at codon position 3. Used in Shen et al. (2016).
gap_mean_pos3	Mean gap percentage per sequence at codon position 3. Used in Shen et al. (2016).
gap_var_pos3	Variance of gap percentage per sequence at codon position 3. Used in Shen et al. (2016).

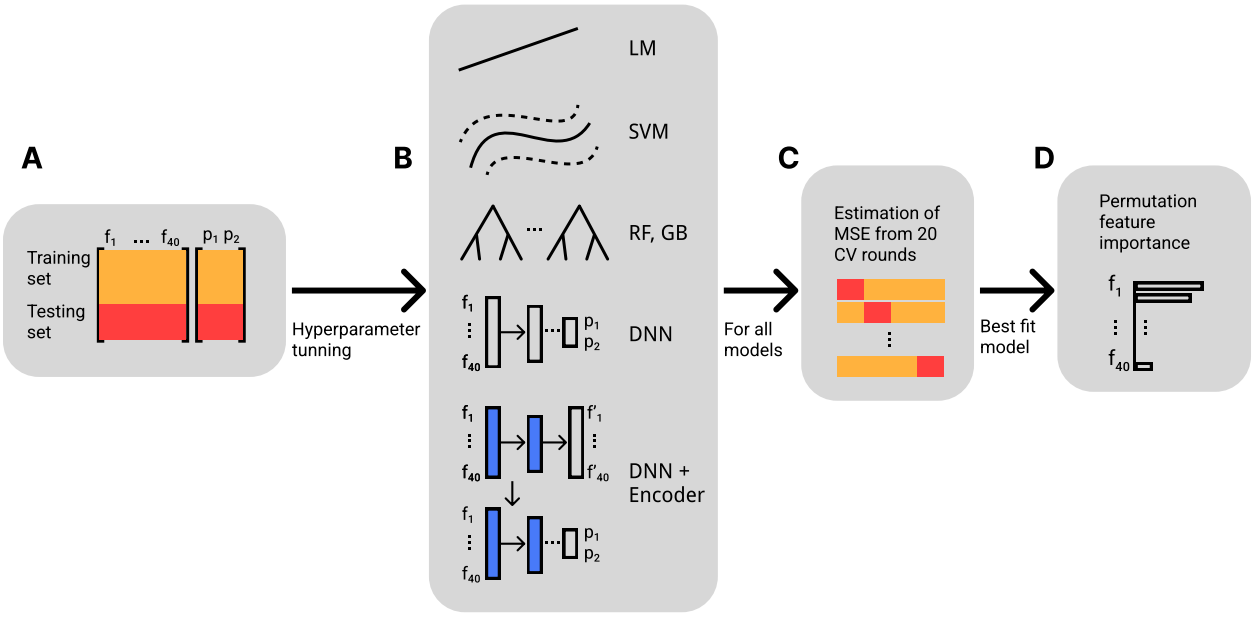


Figure 4.2: The pipeline starts with a case study dataset (e.g., Protacanthopterygii, Carangaria). **A)** Each dataset is first split into training (66% of rows) and testing (33% of rows) sets. **B)** Then all considered models go through a hyperparameter tuning process. **C)** We define the best-fit model for a given dataset as the model with the lowest average MSE of 20 cross-validations (CV) rounds. **D)** With the best-fit model, we obtain the average feature importance (see Section 4.3.6). LM: Linear Model; SVM: Support Vector Machine; RF: Random Forest; GB: Gradient Boosting; DNN: Deep Neural Network; and DNN + Encoder: Deep Neural Network with Encoder layers.

4.3.6 Feature Importance Analysis

After obtaining the best-fitting machine learning model for each dataset, we quantified feature importance using random permutations of the input features (Molnar, 2020). This permutation-based method was originally introduced for Random Forests by Breiman (2001) and subsequently generalized to a model-agnostic framework by Fisher et al. (2019). Let MSE denote the test error of the selected model (e.g., a DNN), and let PMSE_j represent the error on the same test set when feature j is randomly permuted. The importance of feature

j across m permutations is then defined as:

$$\mathrm{FI}_j = \frac{1}{m}\sum_{i=1}^{m}(\mathrm{PMSE}_j - \mathrm{MSE}) \ , \qquad j \in \{1, \ldots, 40\}.$$

We used $m = 100$ permutations to estimate each PMSE_j. The feature-importance profile for each dataset is obtained by ranking features in descending order of FI_j.

4.3.7 Simulated datasets

In the case of deep neural networks (DNNs), we evaluated their ability to identify the most important factor when predicting support. This evaluation used simulated datasets in which the factor of interest could be controlled. Simulations were performed with AliSim (Ly-Trong et al., 2022), a tool available in IQ-TREE. The trees used for simulations were the same as those obtained in Section 4.3.3. Using IQ-TREE, we also optimized the branch lengths of the ASTRAL trees.

The general simulation strategy consisted of generating half of the sequence alignments from one species tree topology (e.g., H1) under a given value of a factor, and the other half from an alternative topology (e.g., H2) under a higher value of the same factor. We then followed the pipeline described in Sections 4.3.4 and 4.3.5 to infer gene trees and extract features and support values from the simulated sequences. In total, we generated 700 sequence alignments for each simulation scheme using the following IQ-TREE options: `iqtree2 --alisim [output name] -t [input tree] -m MG{2.0} --length [sites in output]`.

Using the previously generated Protacanthopterygii species trees, we simulated data under two factors: insertion/deletion rates and alignment length. For the insertion/deletion simulations, half of the sequences (from topology H1) were generated with a deletion rate of 0.02, whereas the other half (from topology H2) were simulated with higher insertion and deletion rates of 0.1 using the `--indel` option. Both sets had an average alignment length of 461 sites with a variance of 10 sites. In the second simulation scheme, sequences from

H1 had an average length of 461 sites, whereas those from H2 had an average length of 153 sites, both with a variance of 10 sites.

4.3.8 Partial Dependence Plot (PDP)

Once we identify the most influential features, we examine how they interact in predicting support for a hypothesis. We isolate the effect of a chosen set of features (e.g., top features from a feature importance analysis) while averaging over all others, yielding the partial dependence between these features and the model. This is implemented via partial dependence plots (PDP; Elith et al., 2008; Hastie et al., 2009), which can be visualized in up to three dimensions.

Specifically, we selected five quantiles of the feature of interest and, for each quantile, replaced the entire feature column with that value, then computed the model's prediction. The average prediction from this perturbation across the feature column is the partial dependence. For two features, we constructed a grid of their values and applied the same procedure to obtain a 2D partial dependence surface. To incorporate uncertainty, we repeated this procedure across 20 cross-validation folds (as in hyperparameter tuning), averaged the resulting 2D partial dependence values, and plotted them.

4.4 Results

4.4.1 Dataset properties, Phylogenomic analyses, and GGI

We assembled 1133 exon alignments for the Protacanthopterygii dataset for 123 species (99 ingroup and 24 outgroup species). Raw reads generated in this study have been deposited in GenBank, under the following Bioproject Accession ID: PRJNA1013549. After applying a number of quality control steps, we retained 1023 (319,386 aligned sites) and 991 (457,167 aligned sites) exon alignments for the Protacanthopterygii and Carangaria (see de-

tails in Duarte-Ribeiro et al., 2024) datasets, respectively. The length of exon alignments obtained is as follows: Protacanthopterygii dataset, 93–2349 sites (mean = 367.4, std = 276.1); Carangaria dataset, 117–5049 sites (mean = 461.3, std = 387.5).

Fig. 4.3 and Fig. C1 show both concatenation-based ML and MSC trees for the Protacanthopterygii and Carangaria datasets, respectively. For Protacanthopterygii, Argentiniformes was the sole major clade with a variable position among the top two competing trees (H1 and H2). This lineage grouped with Esociformes + Salmoniformes in the MSC tree (H2), but was resolved closer to Osmeriformes, Stomitatiformes, Galaxiiformes, and Neoteleostei in the concatenation-based ML tree (H1). For Carangaria, the ML and MSC trees include many more incongruent clades, but a remarkable difference relates to the monophyly of Pleuronectiformes. This clade is resolved as monophyletic in the MSC tree (H1) but is deemed paraphyletic in the concatenation-based ML tree (H2), with Psettodoidei being closer to the (non-flatfish) Polynemidae than to pleuronectoid flatfishes (see Fig. C1; Duarte-Ribeiro et al., 2024).

Neither dataset shows a noticeable separation between H1 and H2 in terms of AU p-values (see Fig. S5), although H1 is the most frequently favored tree topology for Protacanthopterygii (554 exons vs. 463 exons for H2; we discarded six exons due to long-running times of GGI) while the same is true for H2 in Carangaria (582 exons vs. 409 exons for H1). Average AU p-values for H1 and H2 were 0.48 and 0.46, respectively, for the Protacanthopterygii dataset, and 0.44 and 0.55, respectively, for the Carangaria dataset (see Fig. S6). For the Protacanthopterygii dataset, the average of the rank 1 AU p-value (i.e., highest AU p-value) among exon alignments was 0.27, of which only 98 had an AU p-value < 0.05. Likewise, in the Carangaria dataset, the average of the rank 1 AU p-value among exons was 0.22, of which 172 had an AU p-value < 0.05. For both datasets, no evidence of noticeable separation of H1 and H2 in terms of AU p-values was found between exons with a large number of sites (see Fig. S5). Furthermore, site concordance factors in the Protacanthopterygii dataset do not show a significant difference in support for each hypothesis: 30.8 for H1 and 38.2 for H2

in the nodes of interest associated with discordance (Fig. S7). Similarly, in the Carangaria dataset, the values are 35 for H1 and 35.2 for H2 (Fig. S8).

4.4.2 Machine Learning and Feature Importance

The best-fit model for both datasets was the DNN, which achieved the lowest average MSE across all CV rounds (see Fig. 4.4). In contrast, the LM model had the highest average MSE for both datasets. The difference in CV MSEs between the DNN and LM models was statistically significant for both datasets according to the Wilcoxon Rank Sum Test (Fig. 4.4). Moreover, the LM model was the only one that differed significantly from the other models, with a p-value of at least 0.0304 for Protacanthopterygii and at least 0.0203 for Carangaria. All non-LM models share the ability to capture non-linear relationships among tested features. Although the DNN showed lower bias than LM for both datasets, its MSE variance for Protacanthopterygii (3.07×10^{-5}) exceeded that for Carangaria (2.26×10^{-5}).

Tested features (see table 4.1) received different weights across datasets when the DNN predicted AU p-values. Ordering features by the average MSE increase from the DNN model across 40 CV rounds (feature importance analysis; see Materials and Methods) revealed distinct rankings between datasets (Fig. 4.5). For example, the Protacanthopterygii profile identifies the variance of gap percentage per sequence at codon position 3 ("`gap_var_pos3`") as the top feature, whereas this feature ranked 32nd in the Carangaria dataset. Conversely, the top feature in Carangaria was the variance of GC-content per sequence at codon position 3 ("`gc_mean_pos3`"), which ranked 7th in Protacanthopterygii. The top-ranked features in both datasets were sequence-based. Unlike Carangaria, Protacanthopterygii lacked a clearly dominant feature: the fold difference between "`gap_var_pos3`" and the second-ranked feature "`LB_std`" (standard deviation of the long branch score per sequence as described by Struck (2014)) was 1.0009. In contrast, for Carangaria the fold difference between "`gc_mean_pos3`" and the second-ranked feature "`inter_len_mean`" (mean internal branch length in the tree) was 2.8132. Feature rankings from the DNN also differed markedly from those produced by

the LM (Fig. C9).

In the simulation settings, where the correct hypothesis is known, we expect models in the first simulation to identify a feature related to the number of gaps as predictive of support. The DNN model (MSE = 0.000735) matches this expectation more closely than the LM model (Fig. 4.6A), ranking "`seq_len_nogap`," the percentage of sites without gaps, as the top feature. In contrast, the LM model (MSE = 0.004247) ranked "`gc_mean`," the mean GC-content per sequence, as the most important feature. For the DNN model, the fold difference between the first and second features was 3.23. In the second simulation, we expect models to identify a feature related to the number of sites as predictive of support. The DNN model (MSE = 0.000329) again aligns better with this expectation than the LM model (Fig. 4.6B), ranking "`pis`," the number of parsimony-informative sites, as the top feature. In contrast, the LM model (MSE = 0.00532) again ranked "`gc_mean`" as the most important feature. For the DNN model, the fold difference between the first and second features was 4.193847.

To further characterize feature interactions in the empirical datasets, we generated 2D partial dependence plots (Fig. 4.7). In the Protacanthopterygii dataset, lower values of "`gap_var_pos3`" and "`LB_std`" favor hypothesis H1, whereas higher values support H2. A more horizontal gradient for "`LB_std`" under H2 suggests a stronger influence of this feature on that hypothesis. Similarly, in the Carangaria dataset, lower values of "`gc_mean_pos3`" and "`inter_len_mean`" favor H1, whereas higher values support H2.

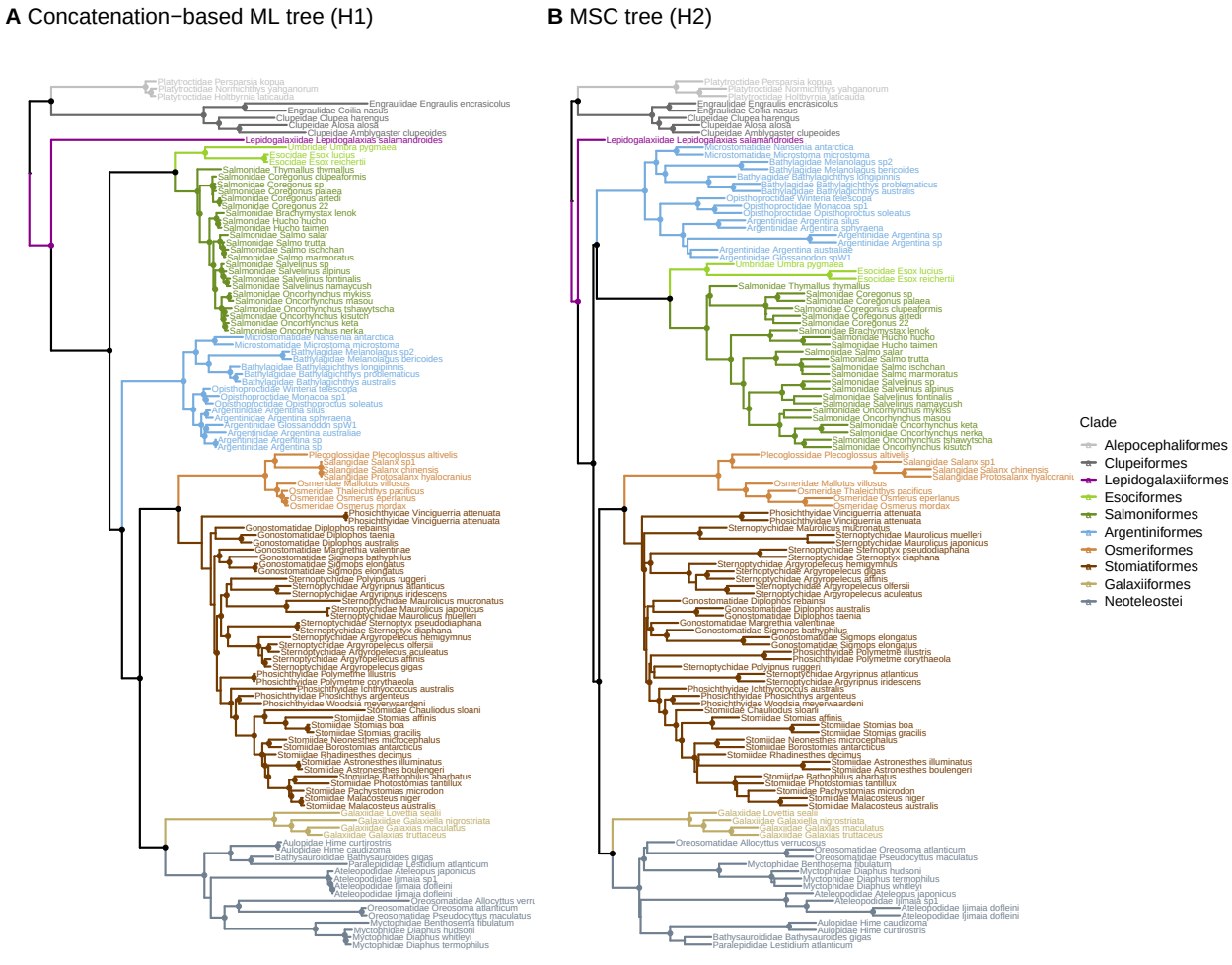


Figure 4.3: Two major phylogenetic trees estimated with the Protacanthopterygii dataset. **A)** concatenation-based ML tree obtained with IQ-TREE (H1); **B)** MSC tree obtained with ASTRAL (H2). Note the variable position of Argentiniformes (in sky-blue). Nodes with filled circles indicate ultrafast bootstrap value $\geq 99\%$ (H1) or quartet score of 1 (H2).

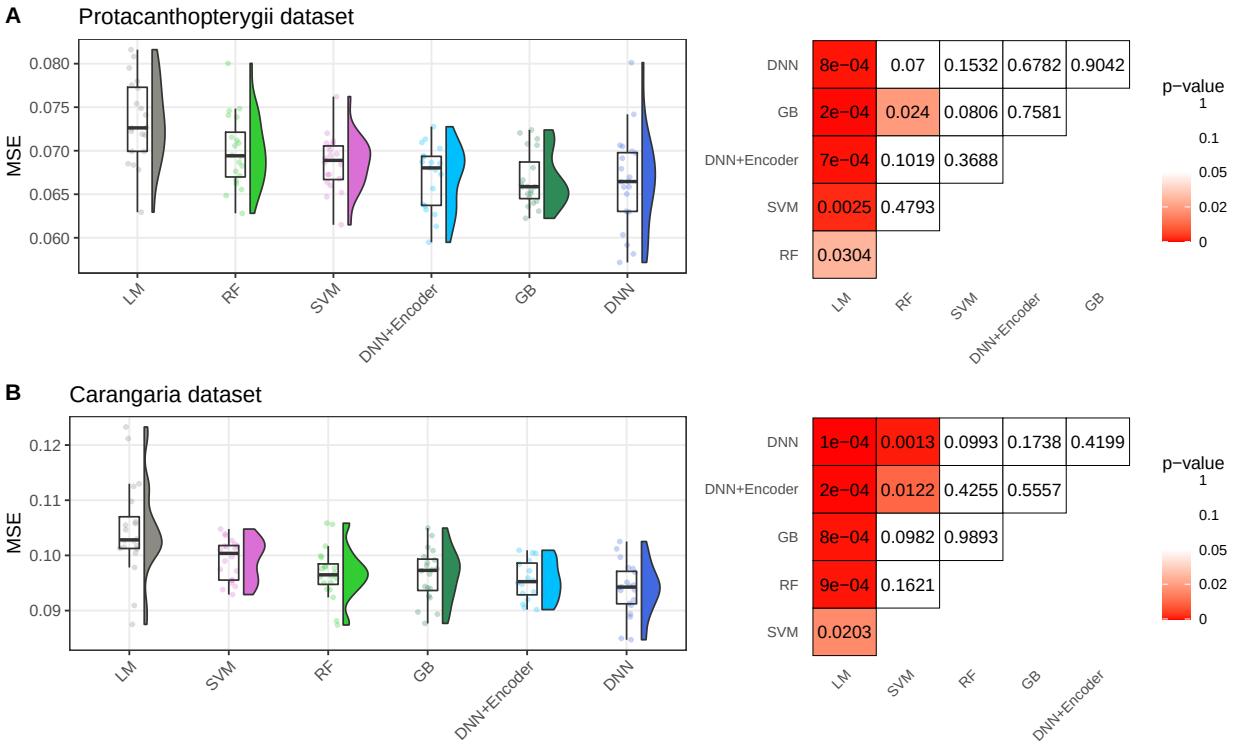


Figure 4.4: Testing MSE distribution from 20 rounds of CV for each model, and pairwise comparison of model testing MSE based on Wilcox Rank Sum Test. Panels **A)** and **B)** show the distribution of MSE and p-values for the Protacantopterygii and Carangaria datasets, respectively. Box plots are arranged in descending order of the average MSE for each model. LM: Linear Model; SVM: Support Vector Machine; RF: Random Forest; GB: Gradient Boosting; DNN: Deep Neural Network; and DNN + Encoder: Deep Neural Network with Encoder layers.

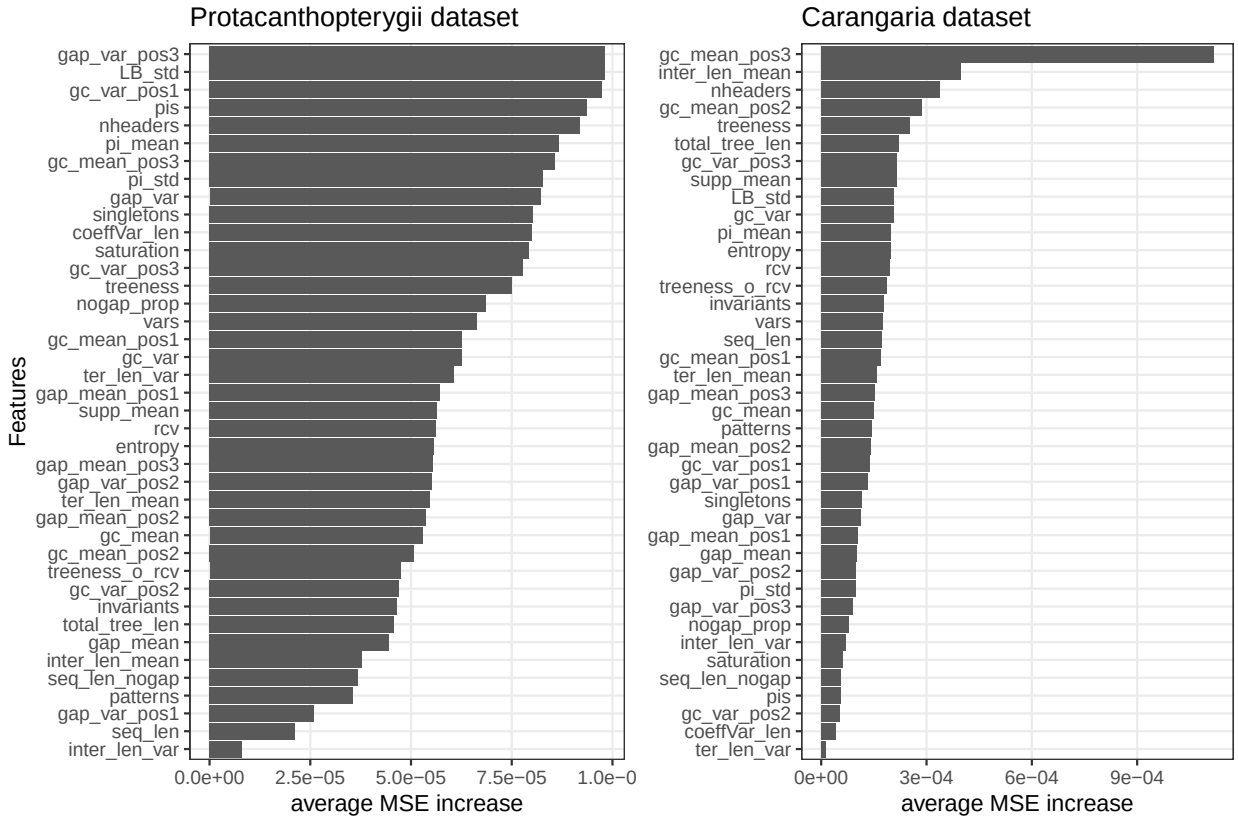


Figure 4.5: Feature importance profiles for DNN models measured by the average increase in testing MSE for the empirical datasets. **A)** Protacanthopterygii dataset; **B)** Carangaria dataset. Bars are ordered in descending importance. The top two features for the Protacanthopterygii dataset were the variance of gap percentage per sequence at codon position 3 ("`gap_var_pos3`") and the standard deviation of the long branch score per sequence ("`LB_std`", as described by Struck, 2014). The top two features for the Carangaria dataset were the mean GC content per sequence at codon position 3 ("`gc_mean_pos3`") and the mean internal branch length in the tree ("`inter_len_mean`"). Remaining features have relatively lower influence on the DNN model. Table 4.1 provides the full names and descriptions of all tested features.

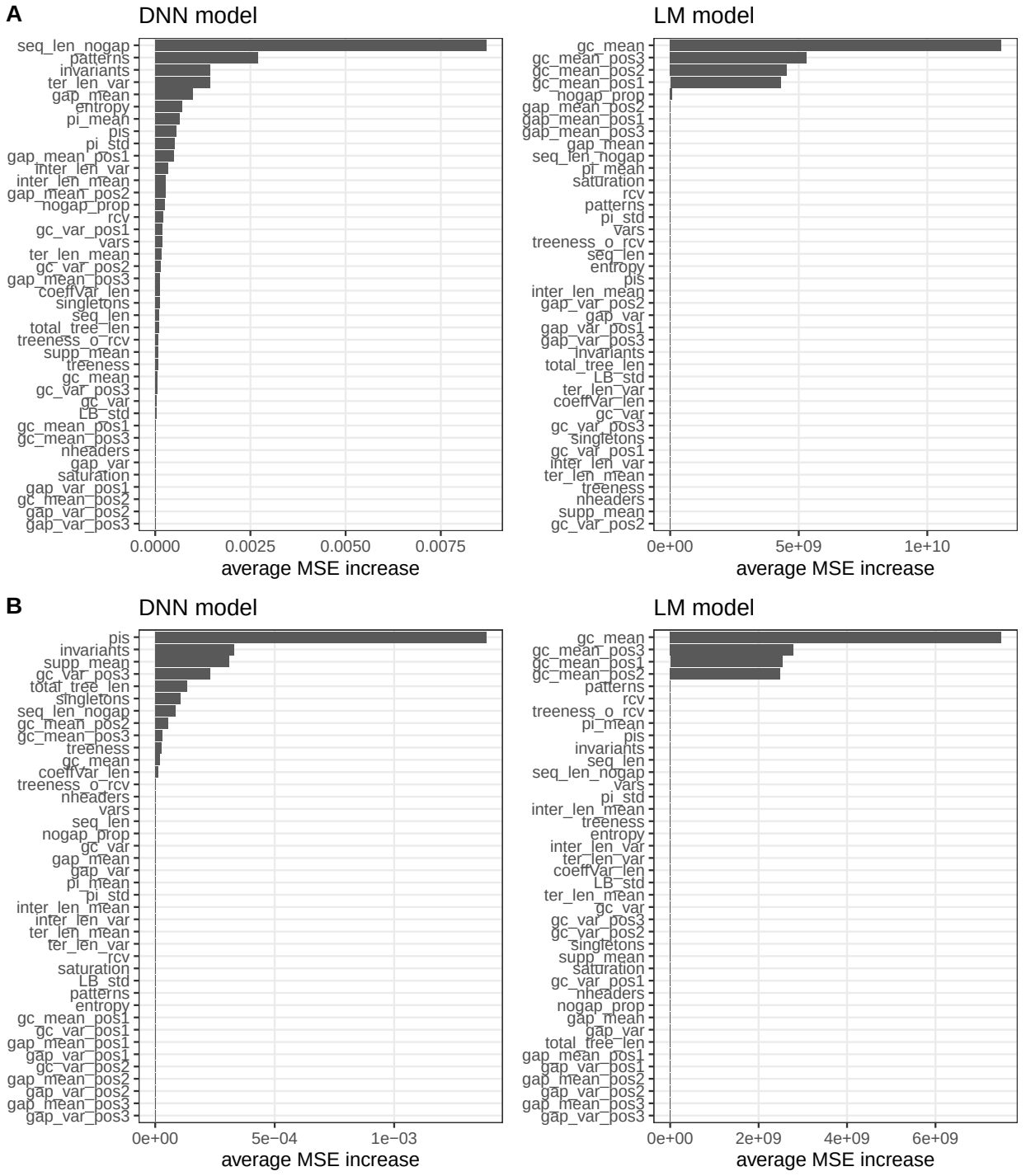


Figure 4.6: Feature importance profiles for DNN models measured by the average increase in testing MSE for the simulated datasets. Bars are ordered in descending importance. **A)** Dataset in which the known factor is the indel rate. The top feature identified by the DNN (first panel) was the percentage of sites without gaps, whereas the LM (second panel) identified GC content as the top feature. **B)** Dataset in which the known factor is the number of sites. The DNN (first panel) identified the number of parsimony-informative sites as the top feature, whereas the LM (second panel) identified GC content as the top feature. Table 4.1 provides the full names and descriptions of all tested features.

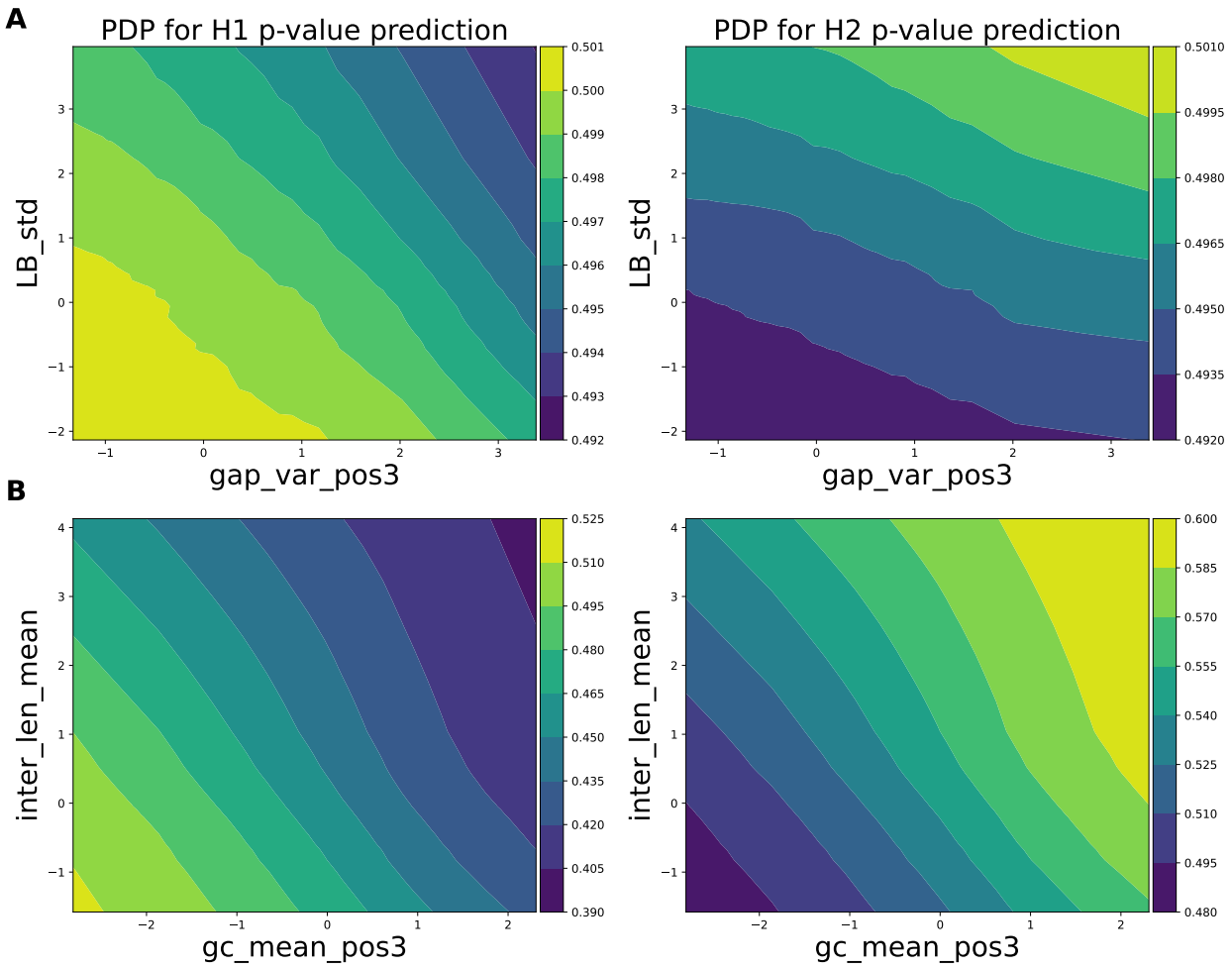


Figure 4.7: 2D Partial dependence plots for both datasets using the DNN model. Panel **A)** shows the results for the Protacanthopterygii dataset, and Panel **B)** shows the results for the Carangaria dataset. Each panel measures the marginal contribution of two features to the model. None of the panels display strictly horizontal or vertical behavior, indicating an interaction between the features in the DNN model. The lighter the color, the greater the marginal contribution to the given p-value and the more support for a given hypothesis. Notice that for each dataset, the panel colors are complementary, indicating that the DNN model learns a set of values that allows separation for each hypothesis.

4.5 Discussions

4.5.1 Phylogenomic analyses

We generated a new dataset for Protacanthopterygii and utilized an existing dataset for Carangaria to estimate phylogenetic trees through concatenation-based maximum likelihood (ML) and multispecies coalescent (MSC) methods. As a result, two major hypotheses emerged for each dataset. In this section, we will focus the discussion on phylogenetic relationships within Protacanthopterygii. For a detailed discussion on the phylogeny of Carangaria and the monophyly of flatfishes, please refer to (Duarte-Ribeiro et al., 2024).

The phylogeny of Protacanthopterygii, long viewed as the sister group of Neoteleostei and an early diverging euteleost lineage (Greenwood et al., 1966), has been addressed by many morphology- and molecule-based studies, including ours (Fig. 4.1). Most agree on two stable supra-ordinal clades (Esociformes + Salmoniformes and Stomiatiformes + Osmeriformes) (e.g., Li et al., 2010; Campbell et al., 2013; Betancur et al., 2017), but other relationships remain contentious. In particular, the composition and monophyly of Protacanthopterygii are unresolved (Campbell et al., 2013; Miya and Nishida, 2015; Mirande, 2017; Betancur et al., 2017). The latest "Fishes of the World" (Nelson et al., 2016) restricts Protacanthopterygii to Salmoniformes and Esociformes, whereas other molecular classifications (Betancur et al., 2013, 2017) include Argentiniformes, Galaxiiformes, Salmoniformes, Esociformes, Osmeriformes, and Stomiatiformes (Stomiati), with Osmeriformes + Stomiatiformes as sister to Neoteleostei. Our results likewise do not recover Protacanthopterygii (sensu Greenwood et al., 1966) as a monophyletic clade or as sister to Neoteleostei. Given this persistent lack of agreement on its limits and monophyly, and the fact that Protacanthopterygii was defined before modern cladistic methods, its recognition as a natural group sensu (Greenwood et al., 1966) appears mainly historical. Rather than focusing on which taxa belong to Protacanthopterygii, it may be more informative to investigate the processes

shaping branching patterns among these lineages along the backbone of the Fish Tree of Life (Fig. 4.3; Betancur et al., 2017).

4.5.2 Machine Learning and Feature Importance

The poor performance of the linear regression model relative to all non-linear models, and the lack of clear differences among the non-linear models (Fig. 4.4), indicate a highly non-parametric feature space, a pattern increasingly recognized in phylogenetic datasets (Zou et al., 2020; Nguyen et al., 2024; Benegas et al., 2025). Models that include non-linear factor interactions thus offer an alternative to previous work, which identified factors underlying phylogenetic uncertainty but did not assess their non-linear interactions (Shen et al., 2021; Vankan et al., 2021; Duchêne et al., 2021). Given our medium-sized datasets (≈ 1000 instances each), non-convex algorithms such as decision tree-based (RF, GB) and neural network-based methods (DNN, DNN+Encoder) can explore the feature space more thoroughly. With the growing availability of chromosome-level genomes (Simakov et al., 2022; Schultz et al., 2023), DNNs are expected to capture even more complex non-linear relationships in whole-genome phylogenetics (Brixi et al., 2025). Regularization techniques—such as limiting tree depth in GB and RF or applying dropout in DNNs—also help manage correlated features and improve generalization. Although linear models like Lasso or Ridge regression incorporate regularization (Hastie et al., 2015), linear approaches commonly used to relate these features to phylogenetic outcomes typically lack such techniques.

The DNN with the lowest testing MSE used early stopping and dropout, which improved its generalizability. Early stopping halts training when testing MSE begins to rise, while dropout randomly deactivates neurons during training, encouraging the network to rely on diverse neurons (Srivastava et al., 2014). Optimal dropout rates differed among datasets and were selected via hyperparameter tuning (see Supplementary Data), underscoring the importance of tuning in machine learning models for phylogenetic studies (Lajaaiti et al., 2023; Ly-Trong et al., 2024). Despite the DNN's generalizability, its learned feature interactions

remain opaque, motivating our subsequent feature importance analyses.

Using DNN-based feature importance analysis, we found that the model effectively identified factors driving uncertainty in support prediction for simulated datasets (Fig. 4.6), whereas the influential features in empirical datasets were dataset-specific (Fig. 4.5). Previous studies have examined relationships between gene properties and support for alternative phylogenetic hypotheses using metrics such as gene likelihood differences or tree distances (Smith et al., 2023). However, these metrics are indirect proxies of phylogenetic support. Here we extend this approach by using machine learning models to predict AU test p-values, providing a direct statistical measure of support for competing hypotheses as it accounts for the distribution of site log-likelihoods within a hypothesis-testing framework (Efron et al., 1996; Shimodaira, 2002). This approach is particularly suitable for binary classification settings. However, selecting between hypotheses based solely on AU p-values and fixed thresholds can be difficult, as some trees may receive comparable support from the data (Simion et al., 2020). We encountered such cases in our analyses and therefore used a multi-output regression model to predict both AU p-values simultaneously, rather than relying on a single p-value in a binary classification framework. This approach is more appropriate when datasets provide evidence supporting multiple hypotheses rather than clearly favoring one.

In the Protacanthopterygii dataset, the variance of gap percentage per sequence at codon position 3 ("`gap_var_pos3`" in Fig. 4.5A) was the top feature. This likely reflects that the sampled lineages diverged at least 100 million years ago (Betancur et al., 2017; Hughes et al., 2018), leading to many indels across taxonomic scales, a pattern intensified by the diversified sampling used to build this dataset. Nodal support for order-level clades was strong under both hypotheses (Fig. 4.3), but high indel variance within gene alignments can still bias gene tree estimation and affect the statistical consistency of ASTRAL-based inference (H2 of the Protacanthopterygii dataset) (Molloy and Warnow, 2018). Likewise, variation in root-to-tip distances ("`LB_std`" in Fig. 4.5A) has been linked to inconsistency

in multispecies coalescent inference (Vankan et al., 2022). Higher values of these features, as indicated by their marginal effects in the PDP (Fig. 4.7A), predict support for H2. Two strategies can reduce these biases. First, adding more species can yield more accurate gene trees via denser alignments (Xi et al., 2016). Second, adding more sites, for example by sampling sliding windows from whole-genome alignments, allows focus on genomic regions directly affecting branches of interest (e.g., the rogue placement of Argentiniformes). In large datasets, this approach can improve the signal-to-noise ratio (Zhang et al., 2021; Di Franco et al., 2022), provided there is no DNA model misspecification (Kumar et al., 2012) or introgression (Solís-Lemus and Ané, 2016).

In the Carangaria dataset, feature importance analyses identified GC content at third codon positions ("`gc_mean_pos3`") as the most influential feature, consistent with previous studies identifying GC content as a biasing factor in this group (Betancur et al., 2013). The interaction between GC content at the third codon position and the branch length, which is represented by "`inter_len_mean`", also aligns with findings by Jarvis et al. (2014), who used linear models, particularly for exons with high substitution rates. In clades that experienced rapid early radiations, phylogenetic inference may require non-homogeneous substitution models that account for rate shifts across sites through time (Huelsenbeck, 2002; Jermiin et al., 2017, 2020). Such shifts arise because the probability of state changes within rapidly radiating lineages differs from that in the rest of the tree, a phenomenon known as heterotachy (Lopez et al., 2002; Kolaczkowski and Thomton, 2004; Kapli et al., 2020). When GC bias is explicitly modeled—for example, using the non-homogeneous GHOST model (Crotty et al., 2020)—analyses of exon markers largely support flatfish monophyly (Duarte-Ribeiro et al., 2024). However, this conclusion is based on a reduced taxonomic dataset due to the computational demands of the GHOST model. Similar support for flatfish monophyly has also been obtained from datasets using UCE markers (which are less affected by GC bias; Jarvis et al., 2014), analyzed under both homogeneous (Harrington et al., 2016) and non-homogeneous models (Duarte-Ribeiro et al., 2024).

By gaining an understanding of the underlying factors that influence a dataset, systematists can make more informed decisions about the appropriate type of phylogenetic analysis or data pre-processing methods to employ before inputting the data into existing software for phylogenetic inference. However, if we let the machine learning models such as Convolutional Neural Networks (Fonseca et al., 2021; Yang et al., 2022) or Graph Neural Networks (Voznica et al., 2022; Lajaaiti et al., 2023) learn optimized features directly from the alignment or tree during the training process, we may sacrifice model explainability, which is a current challenge in applying machine learning models to biological datasets (Sapoval et al., 2022). This is because the features learned from the model may not correlate with existing literature that addresses factors influencing phylogenetic uncertainty. In contrast, our study's Feature Importance analyses demonstrate connections with previous studies, reinforcing the notion that well-defined features can inform decision-making in heuristic search moves within the tree space (Azouri et al., 2021, 2024) and predict the difficulty of a dataset for phylogenetic reconstruction (Haag et al., 2022). However, our study, especially following the analysis of the Carangaria dataset, suggests that current machine learning models used in empirical phylogenetic studies should be trained on datasets encompassing a wide range of processes, including homogeneity, stationarity, and reversibility (Pupko and Mayrose, 2020). Training on diverse datasets is crucial to ensure the machine learning model's robustness against these varied processes.

Non-parametric bootstrap p-values, which form the statistical basis for labeling our machine learning models, do not converge to a single point value as the number of sites approaches infinity (Susko, 2009; Huang et al., 2021). Moreover, depending on the curvature of tree space, selection bias can arise in p-value calculations (Efron et al., 1996; Susko, 2009). Although previous studies have addressed this issue (Shimodaira, 2002; Anisimova and Gascuel, 2006), including approaches that avoid p-values altogether (Minh et al., 2020), further work is needed to evaluate these limitations in the context of machine learning labels. For example, more conservative hypothesis-testing methods such as the KH test (Kishino and

Hasegawa, 1989) could be explored for labeling models when analyzing longer sequences. While the random permutation approach effectively assessed factor interactions, more theoretically grounded methods, such as SHAP values (Lundberg et al., 2020), could provide a more comprehensive measure of marginal feature contributions. Although AliSim enables simulations to validate our approach using known factors, it is not currently feasible to simulate all 40 features from first principles, highlighting the need for further methodological development. Finally, because the performance difference between DNN and DNN+Encoder was not significant (Fig. 4.4), retraining the DNN from scratch for each new dataset may be unnecessary. Instead, the encoder layers from the DNN+Encoder architecture could be reused. As datasets continue to grow, employing pre-trained neural networks may help capture higher-level abstractions from summary statistics and reduce the number of training iterations required for new datasets.

4.5.3 Conclusions

This study introduces a novel non-linear approach to systematically explore factors that explain differences among competing phylogenetic hypotheses by using machine learning models to capture non-linear interactions among these factors. Feature importance analysis with the DNN model correctly identified the expected factors contributing to uncertainty in simulated datasets (Fig. 4.6), and in the empirical datasets the identified factors were consistent with findings from the literature. These exploratory insights can inform decision-making in downstream phylogenetic analyses.

Machine learning models, and particularly DNN models, have already proven useful in several areas of phylogenetic research, including biogeographic modeling (Smith et al., 2017; Fonseca et al., 2021), phylodynamic parameter estimation (Voznica et al., 2022), ancestral state reconstruction (Moreta et al., 2021), and improving heuristics for phylogenetic inference (Azouri et al., 2024). However, the performance of such models ultimately depends on the quality of the training data, which may itself may contain uncertainty. For example, Dotan

et al. (2023) developed a machine learning model to predict alignments from unaligned sequences and identified the assumption of homogeneous substitution rates as a limitation of their approach. Identifying factors that drive uncertainty in phylogenomic datasets—as we do here—and developing models that capture non-linear relationships among features are therefore essential for improving machine learning applications in phylogenetic inference and advancing our understanding of the tree of life.

Our analyses of phylogenomic datasets show that dataset-specific factors contribute unequally to phylogenetic uncertainty. In the Protacanthopterygii dataset, the most influential features are associated with errors in gene tree estimation, suggesting that increasing both species sampling and sequence length could improve inference. Nevertheless, these findings contribute to resolving the debated early evolutionary history of teleost fishes (Betancur et al., 2017; Hughes et al., 2018). Although molecular evidence supports flatfish monophyly (i.e., the H1 hypothesis in the Carangaria dataset; Fig. C1), additional evidence from morphology and developmental pathways (Duarte-Ribeiro et al., 2024) may further clarify the evolutionary history of this rapidly radiating group.

Reference List

Abadi, M., and Coauthors, 2016: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. URL http://arxiv.org/abs/1603.04467, 1603.04467.

Aberer, A. J., D. Krompass, and A. Stamatakis, 2013: Pruning rogue taxa improves phylogenetic accuracy: An efficient algorithm and webservice. *Systematic Biology*, **62 (1)**, 162–166, https://doi.org/10.1093/sysbio/sys078.

Albors, C., J. C. Li, G. Benegas, C. Ye, and Y. S. Song, 2025: A phylogenetic approach to genomic language modeling. *International Conference on Research in Computational Molecular Biology*, Springer, 99–117.

Allen, J. M., and Coauthors, 2017: Phylogenomics from whole genome sequences using aTRAM. *Systematic Biology*, **66 (5)**, 786–798.

Álvarez-Carretero, S., and Coauthors, 2022: A species-level timeline of mammal evolution integrating phylogenomic data. *Nature*, **602 (7896)**, 263–267.

Anisimova, M., and O. Gascuel, 2006: Approximate likelihood-ratio test for branches: a fast, accurate, and powerful alternative. *Systematic biology*, **55 (4)**, 539–552.

Arcila, D., and Coauthors, 2017: Genome-wide interrogation advances resolution of recalcitrant groups in the tree of life. *Nature Ecology and Evolution*, **1 (2)**, 20.

Arcila, D., and Coauthors, 2021: Testing the utility of alternative metrics of branch support to address the ancient evolutionary radiation of tunas, stromateoids, and allies (teleostei: Pelagiaria). *Systematic Biology*, **70 (6)**, 1123–1144.

Azouri, D., S. Abadi, Y. Mansour, I. Mayrose, and T. Pupko, 2021: Harnessing machine learning to guide phylogenetic-tree search algorithms. *Nature communications*, **12 (1)**, 1983.

Azouri, D., O. Granit, M. Alburquerque, Y. Mansour, T. Pupko, and I. Mayrose, 2024: The tree reconstruction game: phylogenetic reconstruction using reinforcement learning. *Molecular Biology and Evolution*, **41 (6)**, msae105.

Benegas, G., C. Albors, A. J. Aw, C. Ye, and Y. S. Song, 2025: A dna language model based on multispecies alignment predicts the effects of genome-wide variants. *Nature Biotechnology*, 1–6.

Betancur, R., C. Li, T. A. Munroe, J. A. Ballesteros, and G. Ortí, 2013: Addressing gene tree discordance and non-stationarity to resolve a multi-locus phylogeny of the flatfishes

(Teleostei: Pleuronectiformes). *Systematic Biology*, **62 (5)**, 763–785, https://doi.org/10.1093/sysbio/syt039.

Betancur, R. R., E. O. Wiley, G. Arratia, A. Acero, N. Bailly, M. Miya, G. Lecointre, and G. Ortí, 2017: Phylogenetic classification of bony fishes. *BMC Evolutionary Biology*, **17 (1)**, 1–40, https://doi.org/10.1186/s12862-017-0958-3.

Betancur-R, R., D. Arcila, R. P. Vari, L. C. Hughes, C. Oliveira, M. H. Sabaj, and G. Ortí, 2019: Phylogenomic incongruence, hypothesis testing, and taxonomic sampling: The monophyly of characiform fishes. *Evolution*, **73 (2)**, 329–345.

Bokulich, N. A., M. R. Dillon, E. Bolyen, B. D. Kaehler, G. A. Huttley, and J. G. Caporaso, 2018: q2-sample-classifier: machine-learning tools for microbiome classification and regression. *Journal of open research software*, **3 (30)**.

Bolger, A. M., M. Lohse, and B. Usadel, 2014: Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, **30 (15)**, 2114–2120.

Breiman, L., 2001: Statistical modeling: The two cultures. *Statistical Science*, **16 (3)**, 199–215, https://doi.org/10.1214/ss/1009213726.

Breiman, L., J. Friedman, R. Olshen, and C. Stone, 1984: Classification and Regression T rees (Monterey, California: Wadsworth). Inc.

Brixi, G., and Coauthors, 2025: Genome modeling and design across all domains of life with evo 2. *BioRxiv*, 2025–02.

Cai, L., Z. Xi, E. M. Lemmon, A. R. Lemmon, A. Mast, C. E. Buddenhagen, L. Liu, and C. C. Davis, 2020: The perfect storm: Gene tree estimation error, incomplete lineage sorting, and ancient gene flow explain the most recalcitrant ancient angiosperm clade, Malpighiales. *bioRxiv*, https://doi.org/10.1101/2020.05.26.112318.

Campbell, M. A., J. A. López, T. Sado, and M. Miya, 2013: Pike and salmon as sister taxa: Detailed intraclade resolution and divergence time estimation of Esociformes+Salmoniformes based on whole mitochondrial genome sequences. *Gene*, **530 (1)**, 57–65, https://doi.org/10.1016/j.gene.2013.07.068, URL http://dx.doi.org/10.1016/j.gene.2013.07.068.

Chen, T., and C. Guestrin, 2016: Xgboost: a scalable tree boosting system Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016: 785-794. *ACM, New York, NY.*

Cortes, C., and V. Vapnik, 1995: Support-vector networks. *Machine learning*, **20 (3)**, 273–297.

Crotty, S. M., B. Q. Minh, N. G. Bean, B. R. Holland, J. Tuke, L. S. Jermiin, and A. V. Haeseler, 2020: GHOST: Recovering Historical Signal from Heterotachously Evolved Sequence Alignments. *Systematic Biology*, **69 (2)**, 249–264, https://doi.org/10.1093/sysbio/syz051.

Cunha, T. J., J. D. Reimer, and G. Giribet, 2022: Investigating sources of conflict in deep phylogenomics of vetigastropod snails. *Systematic Biology*, **71 (4)**, 1009–1022.

De Queiroz, K., 2007: Species concepts and species delimitation. *Systematic Biology*, **56 (6)**, 879–886, https://doi.org/10.1080/10635150701701083.

Di Franco, A., D. Baurain, G. Glöckner, M. Melkonian, and H. Philippe, 2022: Lower statistical support with larger data sets: insights from the Ochrophyta radiation. *Molecular Biology and Evolution*, **39 (1)**, msab300.

Dotan, E., G. Jaschek, T. Pupko, Y. Belinkov, T. Henry, and M. Taub, 2023: Effect of Tokenization on Transformers for Biological Sequences. *bioRxiv*, 2023.08.15.553415, URL https://www.biorxiv.org/content/10.1101/2023.08.15.553415v1%0Ahttps://www.biorxiv.org/content/10.1101/2023.08.15.553415v1.abstract.

Doyle, V. P., R. E. Young, G. J. Naylor, and J. M. Brown, 2015: Can we identify genes with increased phylogenetic reliability? *Systematic Biology*, **64 (5)**, 824–837, https://doi.org/10.1093/sysbio/syv041.

Duarte-Ribeiro, E., and Coauthors, 2024: Phylogenomic and comparative genomic analyses support a single evolutionary origin of flatfish asymmetry. *Nature Genetics*, **56 (6)**, 1069–1072.

Duchêne, D. A., N. Mather, C. Van Der Wal, and S. Y. W. Ho, 2021: Excluding Loci With Substitution Saturation Improves Inferences From Phylogenomic Data. *Systematic Biology*, 1–41, https://doi.org/10.1093/sysbio/syab075.

Efron, B., E. Halloran, and S. Holmes, 1996: Bootstrap confidence levels for phylogenetic trees. *Proceedings of the National Academy of Sciences of the United States of America*, **93 (23)**, 13 429–13 434, https://doi.org/10.1073/pnas.93.23.13429.

Elith, J., J. R. Leathwick, and T. Hastie, 2008: A working guide to boosted regression trees. *Journal of Animal Ecology*, **77 (4)**, 802–813, https://doi.org/10.1111/j.1365-2656.2008.01390.x.

Fonseca, E. M., G. R. Colli, F. P. Werneck, and B. C. Carstens, 2021: Phylogeographic model selection using convolutional neural networks. *Molecular Ecology Resources*, https://doi.org/10.1111/1755-0998.13427.

Fu, L., B. Niu, Z. Zhu, S. Wu, and W. Li, 2012: CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28 (23)**, 3150–3152.

Géron, A., 2022: *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow.* " O'Reilly Media, Inc.".

Glorot, X., and Y. Bengio, 2010: Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256.

Goodfellow, I., Y. Bengio, and A. Courville, 2016: *Deep learning*. MIT press.

Gosselin, S., M. S. Fullmer, Y. Feng, and J. P. Gogarten, 2021: Improving Phylogenies Based on Average Nucleotide Identity, Incorporating Saturation Correction and Non-Parametric Bootstrap Support. *Systematic Biology*, 1–49.

Greenwood, P. H., D. E. Rosen, S. H. Weitzman, G. S. Myers, and Others, 1966: Phyletic studies of teleostean fishes, with a provisional classification of living forms. Bulletin of the AMNH; v. 131, article 4.

Haag, J., D. Höhler, B. Bettisworth, and A. Stamatakis, 2022: From Easy to Hopeless - Predicting the Difficulty of Phylogenetic Analyses. *bioRxiv*, 2022.06.20.496790, URL https://www.biorxiv.org/content/10.1101/2022.06.20.496790v1%0Ahttps://www.biorxiv.org/content/10.1101/2022.06.20.496790v1.abstract.

Haas, B. J., and Coauthors, 2013: De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. *Nature protocols*, **8 (8)**, 1494–1512.

Harrington, R. C., B. C. Faircloth, R. I. Eytan, W. L. Smith, T. J. Near, M. E. Alfaro, and M. Friedman, 2016: Phylogenomic analysis of carangimorph fishes reveals flatfish asymmetry arose in a blink of the evolutionary eye. *BMC evolutionary biology*, **16**, 1–14.

Harvey, M. G., and Coauthors, 2020: The evolution of a tropical biodiversity hotspot. *Science*, **370 (6522)**, 1343–1348, https://doi.org/10.1126/science.aaz6970.

Hastie, T., R. Tibshirani, and J. Friedman, 2009: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed., Springer Series in Statistics, Springer, New York.

Hastie, T., R. Tibshirani, and M. Wainwright, 2015: *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, Boca Raton, FL.

Heath, T. A., S. M. Hedtke, and D. M. Hillis, 2008: Taxon sampling and the accuracy of phylogenetic analyses. *Journal of Systematics and Evolution*, **46 (3)**, 239–257, https://doi.org/10.3724/SP.J.1002.2008.08016, URL http://www.plantsystematics.com.

Hinton, G., N. Srivastava, and K. Swersky, 2012: Neural networks for machine learning

lecture 6a overview of mini-batch gradient descent. *Cited on*, **14 (8)**, 2.

Hoang, D. T., O. Chernomor, A. Von Haeseler, B. Q. Minh, and L. S. Vinh, 2018: UFBoot2: Improving the ultrafast bootstrap approximation. *Molecular Biology and Evolution*, **35 (2)**, 518–522, https://doi.org/10.1093/molbev/msx281.

Huang, J., Y. Liu, T. Zhu, and Z. Yang, 2021: The Asymptotic Behavior of Bootstrap Support Values in Molecular Phylogenetics. *Systematic biology*, **70 (4)**, 774–785, https://doi.org/10.1093/sysbio/syaa100.

Huelsenbeck, J. P., 2002: Testing a covariotide model of DNA substitution. *Molecular Biology and Evolution*, **19 (5)**, 698–707, https://doi.org/10.1093/oxfordjournals.molbev.a004128.

Hughes, L. C., and Coauthors, 2018: Comprehensive phylogeny of ray-finned fishes (Actinopterygii) based on transcriptomic and genomic data. *Proceedings of the National Academy of Sciences of the United States of America*, **115 (24)**, 6249–6254, https://doi.org/10.1073/pnas.1719358115.

Hughes, L. C., and Coauthors, 2021: Exon probe sets and bioinformatics pipelines for all levels of fish phylogenomics. *Molecular Ecology Resources*, **21 (3)**, 816–833.

Ishiguro, N. B., M. Miya, and M. Nishida, 2003: Basal euteleostean relationships: A mitogenomic perspective on the phylogenetic reality of the "Protacanthopterygii". *Molecular Phylogenetics and Evolution*, **27 (3)**, 476–488, https://doi.org/10.1016/S1055-7903(02)00418-9.

Jarvis, E. D., and Coauthors, 2014: Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science*, **346 (6215)**, 1320–1331.

Jermiin, L. S., R. A. Catullo, and B. R. Holland, 2020: A new phylogenetic protocol: Dealing with model misspecification and confirmation bias in molecular phylogenetics. *NAR Genomics and Bioinformatics*, **2 (2)**, 1–14, https://doi.org/10.1093/nargab/lqaa041.

Jermiin, L. S., V. Jayaswal, F. M. Ababneh, and J. Robinson, 2017: *Identifying optimal models of evolution*, Vol. 1525. 379–420 pp., https://doi.org/10.1007/978-1-4939-6622-6_15.

Jiang, Y., P. Tabaghi, and S. Mirarab, 2022: Learning Hyperbolic Embedding for Phylogenetic Tree Placement and Updates. *Biology*, **11 (9)**, 1–24, https://doi.org/10.3390/biology11091256.

Kalyaanamoorthy, S., B. Q. Minh, T. K. F. Wong, A. Von Haeseler, and L. S. Jermiin, 2017: ModelFinder: fast model selection for accurate phylogenetic estimates. *Nature methods*, **14 (6)**, 587–589.

Kapli, P., Z. Yang, and M. J. Telford, 2020: Phylogenetic tree building in the genomic age. *Nature Reviews Genetics*, **21 (7)**, 428–444, https://doi.org/10.1038/s41576-020-0233-0, URL http://dx.doi.org/10.1038/s41576-020-0233-0.

Kingma, D. P., and M. Welling, 2013: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kishino, H., and M. Hasegawa, 1989: Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in Hominoidea. *Journal of molecular evolution*, **29**, 170–179.

Klambauer, G., T. Unterthiner, A. Mayr, and S. Hochreiter, 2017: Self-normalizing neural networks. *Advances in neural information processing systems*, **30**.

Kolaczkowski, B., and J. W. Thomton, 2004: Performance of maximum parsimony and likelihood phylogenetics when evolution is heterogenous. *Nature*, **431 (7011)**, 980–984, https://doi.org/10.1038/nature02917.

Kumar, S., 2022: Embracing green computing in molecular phylogenetics. Oxford University Press, msac043 pp.

Kumar, S., A. J. Filipski, F. U. Battistuzzi, S. L. Kosakovsky Pond, and K. Tamura, 2012: Statistics and truth in phylogenomics. *Molecular Biology and Evolution*, **29 (2)**, 457–472, https://doi.org/10.1093/molbev/msr202.

Lajaaiti, I., S. Lambert, J. Voznica, H. Morlon, and F. Hartig, 2023: A comparison of deep learning architectures for inferring parameters of diversification models from extant phylogenies. *bioRxiv*, 2023.

LeCun, Y., Y. Bengio, and G. Hinton, 2015: Deep learning. *nature*, **521 (7553)**, 436–444.

LeCun, Y., B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, 1989: Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, **2**.

Li, C., M. Hofreiter, N. Straube, S. Corrigan, and G. J. P. Naylor, 2013: Capturing protein-coding genes across highly divergent species. *Biotechniques*, **54 (6)**, 321–326.

Li, H., and R. Durbin, 2009: Fast and accurate short read alignment with Burrows–Wheeler transform. *bioinformatics*, **25 (14)**, 1754–1760.

Li, J., R. Xia, R. McDowall, J. A. López, G. Lei, and C. Fu, 2010: Phylogenetic position of the enigmatic lepidogalaxias salamandroides with comment on the orders of lower euteleostean fishes. *Molecular Phylogenetics and Evolution*, **57 (2)**, 932–936.

Lopez, P., D. Casane, and H. Philippe, 2002: Heterotachy, an important process of protein evolution. *Molecular biology and evolution*, **19 (1)**, 1–7.

Lü, Z., and Coauthors, 2021: Large-scale sequencing of flatfish genomes provides insights into the polyphyletic origin of their specialized body plan. *Nature genetics*, **53 (5)**, 742–751.

Lundberg, S. M., and S. I. Lee, 2017: A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, **2017-Decem (Section 2)**, 4766–4775, 1705.07874.

Lundberg, S. M., and Coauthors, 2020: From local explanations to global understanding with explainable AI for trees. *Nature machine intelligence*, **2 (1)**, 56–67.

Ly-Trong, N., F. A. Matsen IV, and B. Q. Minh, 2024: Treeformer: A transformer-based tree rearrangement operation for phylogenetic reconstruction. *bioRxiv*, 2024–10.

Ly-Trong, N., S. Naser-Khdour, R. Lanfear, and B. Q. Minh, 2022: Alisim: a fast and versatile phylogenetic sequence simulator for the genomic era. *Molecular biology and evolution*, **39 (5)**, msac092.

Maddison, W. P., 1997: Gene trees in species trees. *Systematic Biology*, **46 (3)**, 523–536, https://doi.org/10.1093/sysbio/46.3.523.

Minh, B. Q., M. W. Hahn, and R. Lanfear, 2020: New methods to calculate concordance factors for phylogenomic datasets. *Molecular biology and evolution*, **37 (9)**, 2727–2733.

Mirande, J. M., 2017: Combined phylogeny of ray-finned fishes (actinopterygii) and the use of morphological characters in large-scale analyses. *Cladistics*, **33 (4)**, 333–350.

Miya, M., and M. Nishida, 2015: The mitogenomic contributions to molecular phylogenetics and evolution of fishes: a 15-year retrospect. *Ichthyological Research*, **62 (1)**, 29–71, https://doi.org/10.1007/s10228-014-0440-9.

Molloy, E. K., and T. Warnow, 2018: To Include or Not to Include: The Impact of Gene Filtering on Species Tree Estimation Methods. *Systematic Biology*, **67 (2)**, 285–303, https://doi.org/10.1093/sysbio/syx077.

Molnar, C., 2020: *Interpretable machine learning*. Lulu. com.

Mongiardino Koch, N., 2021: Phylogenomic subsampling and the search for phylogenetically reliable loci. *Molecular biology and evolution*, **38 (9)**, 4025–4038.

Mongiardino Koch, N., 2021: Phylogenomic subsampling and the search for phylogenetically reliable loci. *Molecular biology and evolution*, **38 (9)**, 4025–4038.

Moreta, L. S., O. Rønning, A. S. Al-Sibahi, J. Hein, D. Theobald, and T. Hamelryck, 2021: Ancestral protein sequence reconstruction using a tree-structured Ornstein-Uhlenbeck variational autoencoder. *International Conference on Learning Representations*.

Murphy, K. P., 2012: *Machine learning: a probabilistic perspective*. MIT press.

Nelson, J. S., T. C. Grande, and M. V. Wilson, 2016: *Fishes of the World*. John Wiley & Sons.

Nesterov, Y., 1983: A method of solving a convex programming problem with convergence rate O (1/k** 2). *Doklady Akademii Nauk SSSR*, **269 (3)**, 543.

Nguyen, E., and Coauthors, 2024: Sequence modeling and design from molecular to genome scale with evo. *Science*, **386 (6723)**, eado9336.

Nguyen, L. T., H. A. Schmidt, A. Von Haeseler, and B. Q. Minh, 2015: IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, **32 (1)**, 268–274, https://doi.org/10.1093/molbev/msu300.

O'Malley, M. A., M. M. Leger, J. G. Wideman, and I. Ruiz-Trillo, 2019: Concepts of the last eukaryotic common ancestor. *Nature Ecology and Evolution*, **3 (3)**, 338–344, https://doi.org/10.1038/s41559-019-0796-3.

Pedregosa, F., and Coauthors, 2011: Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, **12**, 2825–2830.

Philippe, H., H. Brinkmann, D. V. Lavrov, D. T. J. Littlewood, M. Manuel, G. Wörheide, and D. Baurain, 2011: Resolving difficult phylogenetic questions: why more sequences are not enough. *PLoS biology*, **9 (3)**, e1000 602.

Phillips, M. J., and D. Penny, 2003: The root of the mammalian tree inferred from whole mitochondrial genomes. *Molecular Phylogenetics and Evolution*, **28 (2)**, 171–185, https://doi.org/10.1016/S1055-7903(03)00057-5.

Pupko, T., and I. Mayrose, 2020: A gentle Introduction to Probabilistic Evolutionary Models. *Phylogenetics in the Genomic Era*, No commercial publisher, 0–21.

Rachtman, E., Y. Jiang, and S. Mirarab, 2025: Machine learning enables alignment-free distance calculation and phylogenetic placement using k-mer frequencies. *Molecular Ecology Resources*, e70055.

Ranwez, V., S. Harispe, F. Delsuc, and E. J. Douzery, 2011: MACSE: Multiple alignment of coding SEquences accounting for frameshifts and stop codons. *PLoS ONE*, **6 (9)**, https://doi.org/10.1371/journal.pone.0022594.

Ribeiro, E., A. M. Davis, R. A. Rivero-Vega, G. Ortí, and R. Betancur-R, 2018: Post-Cretaceous bursts of evolution along the benthic-pelagic axis in marine fishes. *Proceedings of the Royal Society B*, **285 (1893)**, 20182 010.

Sapoval, N., and Coauthors, 2022: Current progress and open challenges for applying deep learning across the biosciences. *Nature Communications*, **13 (1)**, 1728.

Schultz, D. T., S. H. Haddock, J. V. Bredeson, R. E. Green, O. Simakov, and D. S. Rokhsar, 2023: Ancient gene linkages support ctenophores as sister to other animals. *Nature*, **618 (7963)**, 110–117, https://doi.org/10.1038/s41586-023-05936-6.

Shen, X. X., C. T. Hittinger, and A. Rokas, 2017: Contentious relationships in phylogenomic studies can be driven by a handful of genes. *Nature Ecology and Evolution*, **1 (5)**, 1–10, https://doi.org/10.1038/s41559-017-0126, URL http://dx.doi.org/10.1038/s41559-017-0126.

Shen, X. X., L. Salichos, and A. Rokas, 2016: A genome-scale investigation of how sequence, function, and tree-based gene properties influence phylogenetic inference. *Genome Biology and Evolution*, **8 (8)**, 2565–2580, https://doi.org/10.1093/gbe/evw179.

Shen, X.-X., J. L. Steenwyk, and A. Rokas, 2021: Dissecting Incongruence between Concatenation- and Quartet-Based Approaches in Phylogenomic Data. *Systematic Biology*, **70 (5)**, 997–1014, https://doi.org/10.1093/sysbio/syab011.

Shimodaira, H., 2002: An approximately unbiased test of phylogenetic tree selection. *Systematic Biology*, **51 (3)**, 492–508, https://doi.org/10.1080/10635150290069913.

Simakov, O., and Coauthors, 2022: Deeply conserved synteny and the evolution of metazoan chromosomes. *Science Advances*, **8 (5)**, https://doi.org/10.1126/sciadv.abi5884.

Simão, F. A., R. M. Waterhouse, P. Ioannidis, E. V. Kriventseva, and E. M. Zdobnov, 2015: BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics*, **31 (19)**, 3210–3212.

Simion, P., F. Delsuc, and H. Philippe, 2020: To What Extent Current Limits of Phylogenomics Can Be Overcome? 0–34.

Simmons, M. P., and J. Gatesy, 2021: Collapsing dubiously resolved gene-tree branches in phylogenomic coalescent analyses. *Molecular Phylogenetics and Evolution*, **158**, 107 092.

Singhal, S., and Coauthors, 2021: Congruence and Conflict in the Higher-Level Phylogenetics of Squamate Reptiles: An Expanded Phylogenomic Perspective. *Systematic biology*, **70 (3)**, 542–557, https://doi.org/10.1093/sysbio/syaa054.

Slater, G. S. C., and E. Birney, 2005: Automated generation of heuristics for biological

sequence comparison. *BMC bioinformatics*, **6**, 1–11.

Smith, B. T., and Coauthors, 2023: Phylogenomic analysis of the parrots of the world distinguishes artifactual from biological sources of gene tree discordance. *Systematic Biology*, **72 (1)**, 228–241.

Smith, M. L., M. Ruffley, A. Espíndola, D. C. Tank, J. Sullivan, and B. C. Carstens, 2017: Demographic model selection using random forests and the site frequency spectrum. *Molecular Ecology*, **26 (17)**, 4562–4573, https://doi.org/10.1111/mec.14223.

Solís-Lemus, C., and C. Ané, 2016: Inferring Phylogenetic Networks with Maximum Pseudolikelihood under Incomplete Lineage Sorting. *PLoS Genetics*, **12 (3)**, 1–21, https://doi.org/10.1371/journal.pgen.1005896, 1509.06075.

Springer, M. S., and J. Gatesy, 2016: The gene tree delusion. *Molecular phylogenetics and evolution*, **94**, 1–33.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014: Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, **15**, 1929–1958.

Stamatakis, A., 2014: RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, **30 (9)**, 1312–1313, https://doi.org/10.1093/bioinformatics/btu033.

Steenwyk, J. L., Y. Li, X. Zhou, X. X. Shen, and A. Rokas, 2023: Incongruence in the phylogenomics era. *Nature Reviews Genetics*, **24 (12)**, 834–850, https://doi.org/10.1038/s41576-023-00620-x.

Strogatz, S. H., 2015: *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. 2nd ed., Westview Press, Boulder, CO.

Struck, T. H., 2014: Trespex—detection of misleading signal in phylogenetic reconstructions based on tree information. *Evolutionary Bioinformatics*, **10**, EBO–S14 239.

Susko, E., 2009: Bootstrap support is not first-order correct. *Systematic Biology*, **58 (2)**, 211–223, https://doi.org/10.1093/sysbio/syp016.

Susko, E., and A. J. Roger, 2021: Long Branch Attraction Biases in Phylogenetics. *Systematic Biology*, **00 (0)**, 1–6, https://doi.org/10.1093/sysbio/syab001.

Vankan, M., S. Y. Ho, and D. A. Duchêne, 2021: Evolutionary Rate Variation Among Lineages in Gene Trees has a Negative Impact on Species-Tree Inference. *Systematic Biology*, 2–32.

Vankan, M., S. Y. W. Ho, and D. A. Duchêne, 2022: Evolutionary rate variation among lineages in gene trees has a negative impact on species-tree inference. *Systematic Biology*, **71 (2)**, 490–500.

Voznica, J., A. Zhukova, V. Boskova, E. Saulnier, F. Lemoine, M. Moslonka-Lefebvre, and O. Gascuel, 2022: Deep learning from phylogenies to uncover the epidemiological dynamics of outbreaks. *Nature Communications*, **13 (1)**, 3896.

Xi, Z., L. Liu, and C. C. Davis, 2016: The impact of missing data on species tree estimation. *Molecular Biology and Evolution*, **33 (3)**, 838–860, https://doi.org/10.1093/molbev/msv266.

Yang, B., Z. Zhang, C.-Q. Yang, Y. Wang, M. C. Orr, H. Wang, and A.-B. Zhang, 2022: Identification of species by combining molecular and morphological data using convolutional neural networks. *Systematic Biology*, **71 (3)**, 690–705.

Zerbino, D. R., and E. Birney, 2008: Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome research*, **18 (5)**, 821–829.

Zhang, C., M. Rabiee, E. Sayyari, and S. Mirarab, 2018: ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. *BMC bioinformatics*, **19**, 15–30.

Zhang, D., and Coauthors, 2021: Most genomic loci misrepresent the phylogeny of an avian radiation because of ancient gene flow. *Systematic Biology*, **14**, 1–12.

Zou, Z., H. Zhang, Y. Guan, J. Zhang, and L. Liu, 2020: Deep Residual Neural Networks Resolve Quartet Molecular Phylogenies. *Molecular Biology and Evolution*, **37 (5)**, 1495–1507, https://doi.org/10.1093/molbev/msz307.

Appendix A

Supplementary material: Sparse learning for scalable phylogenetic network inference

Table A.1: Summary of notation used throughout the proofs

Symbol	Description
T	Number of taxa (leaf nodes) in the network
n	Number of random phylogenetic networks
M	Number of regression trees in the ISLE ensemble
ρ	fraction of selected CFs rows used in phylogenetic network reconstruction
H	Number of hybridization events
h_i	i^{th} hybridization event
$\mathbf{X}$	Predictor matrix used in regression
$\mathbf{y}$	Response vector used in regression
Λ	Set containing k sparsity parameter values
λ_k	k^{th} sparsity parameter value
$\tilde{\lambda}$	birth rate in the birth-death-hybridization process
τ	Hitting time of a particular state
N_t	State of a discrete-time process at time t
r_i	Net rate of change at state i

A.1 SNaQ time complexity

Before analyzing the time complexity of SNaQ, we can state two useful claims that provide a firmer foundation for the analysis:

Claim 1. *If a network is level-1, then at most one hybrid edge can be added to any given tree edge of a displayed tree from the network.*

Proof. Assume, for contradiction, that a level-1 network allows at least two hybrid edges to be added to the same tree edge of a displayed tree. By minimal counterexample, suppose exactly two such hybrid edges exist. This necessarily leads to a contradiction with the definition of a level-1 network.

Let the tree edge be e_t, corresponding to the node pair $(t-1, t)$. Let h_i denote the i^{th} hybrid node, and let p_{h_i} denote the one parent of h_i. We consider the four possible scenarios in which two hybrid edges could lie on the same tree edge:

Case 1 ($\underline{h_1 \text{ and } h_2 \text{ added}}$)**.** *Without loss of generality, assume that the hybrid nodes h_1 and h_2 are inserted along e_t from top to bottom, subdividing e_t into three segments:*

$$e_{h_1} = (t-1, h_1),$$
$$e_{h_2} = (h_1, h_2),$$
$$e_{t_u} = (h_2, t).$$

Here, e_{h_1} appears in both the h_1-based and h_2-based hybridization cycles. This contradicts the level-1 assumption, which states that each edge can be part of at most one cycle. We follow the similar rationale for the rest of the cases.

Case 2 ($\underline{p_{h_1} \text{ and } p_{h_2} \text{ added}}$)**.** *Without loss of generality, assume that p_{h_1} and p_{h_2} are inserted along e_t from top to bottom, subdividing it into:*

$$e_{p_{h_1}} = (t-1, p_{h_1}), \quad e_{p_{h_2}} = (p_{h_1}, p_{h_2}), \quad e_{t_u} = (p_{h_2}, t).$$

Again, $e_{p_{h_1}}$ appears in both the h_1-based and h_2-based hybridization cycles, violating the level-1 assumption.

Case 3 ($\underline{h_1 \text{ and } p_{h_2} \text{ added}}$)**.** *Without loss of generality, assume that h_1 and p_{h_2} are inserted along e_t from top to bottom, subdividing it into:*

$$e_{h_1} = (t-1, h_1), \quad e_{p_{h_2}} = (h_1, p_{h_2}), \quad e_{t_u} = (p_{h_2}, t).$$

Here, e_{h_1} is shared between the cycles associated with h_1 and h_2, again contradicting the level-1 assumption.

Case 4 ($\underline{p_{h_1} \text{ and } h_2 \text{ added}}$)**.** *Without loss of generality, assume that p_{h_1} and h_2 are inserted along e_t from top to bottom, resulting in:*

$$e_{p_{h_1}} = (t-1, p_{h_1}), \quad e_{h_2} = (p_{h_1}, h_2), \quad e_{t_u} = (h_2, t).$$

In this case, $e_{p_{h_1}}$ appears in both the h_1- and h_2-based cycles, which again violates the level-1 constraint.

In all four cases, we arrive at a contradiction of the level-1 network assumption. Therefore, no more than one hybrid edge can be added to any given tree edge in a level-1 network. □

Claim 2. *Let T be the number of taxa and H the number of hybridization events. If a network is level-1 in SNaQ, then $T > H$ in any displayed tree from the network.*

Proof. Assume, for contradiction, that the network is level-1 in SNaQ and that $T \leq H$. Then we can derive the following inequalities:

$$H \geq T > T - \frac{3}{2} \geq \left\lfloor \frac{2T-3}{2} \right\rfloor.$$

Thus, we must have

$$H > \left\lfloor \frac{2T-3}{2} \right\rfloor,$$

which implies

$$H = \left\lfloor \frac{2T-3}{2} \right\rfloor + m$$

for some integer $m \geq 1$.

In each hybridization event, a new hybrid node is added, which must have two distinct parents inserted into two different edges. If both parents were added to the same edge, then the entire hybridization cycle would lie within a single tree edge. However, such cases are ignored by SNaQ as they are unidentifiable (Solís-Lemus and Ané, 2016). Therefore, to support H hybridization events, at least $2H$ distinct edges are needed.

But the displayed tree contains only $2T-3$ edges. So to accommodate $H = \lfloor \frac{2T-3}{2} \rfloor + m$ hybridization events, we would need

$$2H = (2T-3) + 2m - 1$$

edges, which exceeds the number available in the tree. By the pigeonhole principle, this implies that at least one edge must contain two or more hybridization events.

To illustrate this, for example, let $m = 1$, and consider the following setup, where the bottom row lists all the edges in the displayed tree, and the rows above indicate the hybrid events they support:

$h_{\lfloor \frac{2T-3}{2} \rfloor + 1}$								
h_1	h_2	$\ldots$	$h_{\lfloor \frac{2T-3}{2} \rfloor}$	h_1	h_2	$\ldots$	$h_{\lfloor \frac{2T-3}{2} \rfloor}$	$h_{\lfloor \frac{2T-3}{2} \rfloor + 1}$
e_1	e_2	$\ldots$	$e_{\lfloor \frac{2T-3}{2} \rfloor}$	$e_{\lfloor \frac{2T-3}{2} \rfloor + 1}$	$e_{\lfloor \frac{2T-3}{2} \rfloor + 2}$	$\ldots$	$e_{(2T-3)-1}$	$e_{(2T-3)}$

Here, edge e_1 supports both h_1 and $h_{\lfloor \frac{2T-3}{2} \rfloor + 1}$ hybridization events, which means it contains two hybrid edges—implying two overlapping hybridization cycles.

By the contrapositive of Claim 1: if at least two hybrid edges lie on the same tree edge in a displayed tree, then the network is not level-1. Therefore, this configuration contradicts our original assumption that the network is level-1. □

A.1.1 Objective function in SNaQ

Algorithm A.1 describes the objective function evaluation, which corresponds to the log-pseudolikelihood score defined in Eq. 2.1 in the main text. Thus, each evaluation of the objective function requires $O(T^5)$ operations in total, which comes from repeating $O(T^4)$ times step 2a in Algorithm A.1.

Algorithm A.1 SNaQ objective function evaluation

1. **Input:** a fixed network topology and a vector of parameters $\mathbf{x}$, which includes internal branch lengths and inheritance probabilities.
2. For $j = 1$ to $\binom{T}{4}$:
   (a) Extract the subnetwork induced by the quartet j: $O(T)$ operations
   (b) Using $\mathbf{x}$, compute the quartet j log-likelihood score: $O(1)$ operations
3. Aggregate all quartet log-likelihoods into an overall log-pseudolikelihood score: $O(T^4)$ operations

A.1.2 Parameter optimization in SNaQ

Now, to obtain the log-pseudolikelihood score for a given network topology, it is necessary to optimize both the internal branch lengths and the inheritance probabilities. SNaQ accomplishes this using a derivative-free optimization algorithm called BOBYQA (see Algorithm A.2) (Powell et al., 2009).

Let m denote the total number of parameters to be optimized—including both branch lengths and inheritance probabilities—and let d denote the number of iterations of the algorithm. A quadratic approximation of the objective function is constructed by evaluating the objective function (see Section A.1.1) approximately m times to numerically approximate its gradient.

The most computationally expensive components of parameter optimization are outlined

Algorithm A.2 SNaQ parameter optimization using BOBYQA

1. Initialize the parameter vector $\mathbf{x} \in \mathbb{R}^m$; let $F(\mathbf{x})$ be the SNaQ objective function (see Algorithm A.1); let $\mathbf{e}_j \in \mathbb{R}^m$ denote a standard basis vector (i.e., a vector of zeros with a 1 in the j-th component); and let h be a constant defined by the BOBYQA algorithm.

2. For $i = 1$ to m:

   (a) Define perturbations $\mathbf{x} + h\mathbf{e}_i$ and $\mathbf{x} - h\mathbf{e}_i$: $O(m)$ operations

   (b) Evaluate $F(\mathbf{x} + h\mathbf{e}_i)$ and $F(\mathbf{x} - h\mathbf{e}_i)$: $O(2T^5)$ operations

   (c) Approximate the gradient component:

   $$g_i = \frac{F(\mathbf{x} + h\mathbf{e}_i) - F(\mathbf{x} - h\mathbf{e}_i)}{2h}$$

   $O(1)$ operations

3. Construct the initial quadratic model Q_1 using the approximated gradients: $O(m^2)$ operations

4. For $j = 1$ to d:

   (a) Solve the quadratic problem Q_j to update $\mathbf{x}$: $O(m^2)$ operations

   (b) Evaluate the objective function at the new interpolation point: $O(T^5)$ operations

   (c) Update the quadratic model Q_j: $O(m^2)$ operations

---

in Algorithm A.2 and considering the most dominant components for each step in it, we find:

$$\underbrace{O(m)}_{\text{Step 1}} + \underbrace{O\left(mT^5\right)}_{\text{Step 2}} + \underbrace{O\left(m^2\right)}_{\text{Step 3}} + \underbrace{O(d) \times \left[O\left(2m^2\right) + O\left(T^5\right)\right]}_{\text{Step 4}} \leq O\left(2mT^5\right) + O\left(3dm^2\right) + O\left(dT^5\right)$$

$$= O\left(mT^5\right) + O\left(m^2\right) + O\left(T^5\right)$$

If we are creating networks with T taxa, then the number of internal branches in the unrooted displayed tree is $T-3$. Let H denote the number of hybridization events. Since each hybridization event involves approximately three edges, then we can assume $m \approx T + 3H$.

By Claim 2 which states that $T > H$, the above simplifies as:

$$\begin{aligned} O\left(mT^5\right) + O\left(m^2\right) + O\left(T^5\right) &= O\left(\{T+3H\}T^5\right) + O\left(\{T+3H\}^2\right) + O\left(T^5\right) \\ &\leq O\left(4T \times T^5\right) + O\left(16T^2\right) + O\left(T^5\right) \\ &= O\left(4T^6\right) + O\left(16T^2\right) + O\left(T^5\right) \\ &\leq O\left(21T^6\right) \\ &= O\left(T^6\right) \end{aligned} \tag{A.1}$$

Therefore, the overall time complexity of optimizing the branch lengths and inheritance probabilities with SNaQ using BOBYQA is $O(T^6)$.

A.1.3 Heuristic search

Finally, the most computationally dominant steps of the SNaQ heuristics are found in Algorithm A.3. Let D be the total number of iterations; then the total time complexity of SNaQ is $O(DT^6)$. If the algorithm visits at least all the edges as part of the hill-climbing search, then $D \geq T$. Assuming $D = T$, the total time complexity of SNaQ becomes $O(T^7)$.

Algorithm A.3 SNaQ heuristics

Repeat until the stopping criterion is met:

1. Copy the current network: $O(T)$ operations.
2. Perform a move and update attributes: $O(1)$ operations.
3. Optimize network parameters (e.g., branch lengths; see Algorithm A.2): $O(T^6)$ operations.

We arbitrarily assumed that the number of moves per run is in the order of T so that we can upper bound SNaQ's time complexity. This is similar to how Neighbor Joining gradually constructs the phylogenetic tree edge by edge (Yang, 2014). However, we also conducted simulations to evaluate how well this assumption holds in practice (Fig. A.1), and we found

that it seems reasonable. Although it is a theoretical assumption, empirically, it is not too far off.

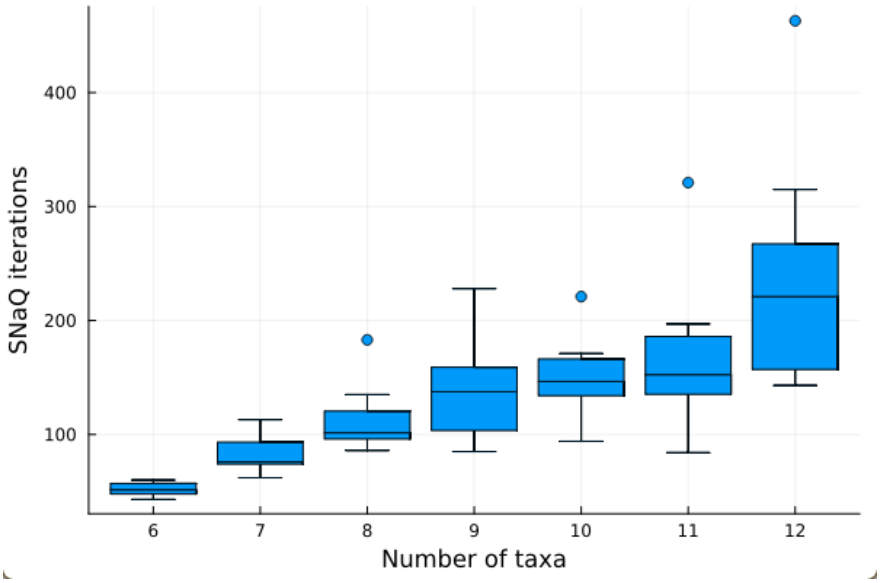


Figure A.1: The number of iterations per run approximately increases linearly as the number of species increases in simulated datasets. Simulations conducted using version 1.0 of the SNaQ Julia package with default arguments. True networks were generated at random by (1) simulating a random tree with the given number of taxa from a star tree using the PhyloCoalSimulations Julia package (Fogg et al., 2023), before (2) randomly selecting origin and destination edges from which a single reticulation is added into the network, and then finally (3) scaling branch lengths such that the average branch length of the network is 1.0. The concordance factor data used as input to SNaQ was then computed from 250 gene trees that were simulated from each network using PhyloCoalSimulations.

A.2 Qsin time complexity

Before analyzing the time complexity of the Qsin pipeline presented in Algorithm 4—which is developed through six propositions and one main theorem—we first state a useful claim that provides a foundation for the analysis:

Claim 3. *Let $\tilde{\lambda}$, μ, ν_+, ν_- be the birth and death rates, and the generative and degenerative hybridization rates, respectively. If $\tilde{\lambda} > \mu$ and $\nu_+ > \nu_-$, then, on average, the SiPhyNetwork algorithm, under the birth–death–hybridization process, requires T iterations to generate a*

network with T taxa.

Proof. Although the time complexity of the algorithm for generating a single network using SiPhyNetworks (Justison et al., 2023) was not explicitly described when the method was proposed, we can estimate it based on established analyses of birth-death process simulation (Huelsenbeck, 2018).

The birth–death–hybridization process is an irreducible Markov process (i.e., all states communicate) whose expected change in the number of lineages per unit time is positive and grows approximately quadratically with the state index i. More precisely, the net rate of increase at state i is (Justison and Heath, 2023):

$$r_i = \tilde{\lambda}_i - \mu_i = i(\tilde{\lambda} - \mu) + \binom{i}{2}(\nu_+ - \nu_-).$$

Since by assumption $\tilde{\lambda} > \mu$ and $\nu_+ > \nu_-$, then $r_i > 0$ for all i. When $r_i > 0$ for all i, the process is transient (Lefebvre, 2007, pp. 141) and exhibits a systematic drift toward larger states. Let $\nu_{i,i}$ denote the total jump rate out of state i, defined as:

$$\nu_{i,i} = \tilde{\lambda}_i + \mu_i = i(\tilde{\lambda} + \mu) + \binom{i}{2}(\nu_+ + \nu_-),$$

Let N_t be the random variable describing the state of the algorithm at time step $t \in \{0, 1, 2, \dots\}$. Then, the one-step mean drift of the discrete-time chain is given by:

$$\mathbb{E}[N_{t+1} - N_t \mid N_t = i] = (-1)p_{i,i-1} + (0)p_{i,i} + (1)p_{i,i+1} = \frac{r_i}{\nu_{i,i}} = \frac{2(\tilde{\lambda} - \mu) + (i-1)(\nu_+ - \nu_-)}{2(\tilde{\lambda} + \mu) + (i-1)(\nu_+ + \nu_-)}.$$

This expression converges to $\frac{\nu_+ - \nu_-}{\nu_+ + \nu_-}$ as $i \to \infty$. Moreover, by induction (Abbott, 2015, pp. 56), the sequence defined by this expression over all i is monotone. When $i = 1$ (i.e., at $t = 0$), the drift simplifies to $\frac{\tilde{\lambda} - \mu}{\tilde{\lambda} + \mu}$, which is strictly positive. Thus, we have two bounds but do not know which is the smaller. Therefore, for all $i \geq 1$, we can set the lower bound

of the mean drift by the minimum of the two:

$$\mathbb{E}[N_{t+1} - N_t \mid N_t = i] \geq \varepsilon := \min\left\{\frac{\tilde{\lambda} - \mu}{\tilde{\lambda} + \mu}, \frac{\nu_+ - \nu_-}{\nu_+ + \nu_-}\right\} > 0.$$

Now, define another random variable $X_t := T - N_t$, where T is the target state at which the algorithm stops. Then the drift inequality becomes:

$$\mathbb{E}[X_t - X_{t+1} \mid X_t = T - i] \geq \varepsilon.$$

Let τ denote the first hitting time of state $X_t = 0$, which is equivalent to the first time the process reaches state T. Then, by the Additive Drift Theorem (He and Yao, 2004; Kötzing, 2014; Lengler, 2019), the expected value of τ satisfies:

$$\mathbb{E}[\tau] \leq \frac{\mathbb{E}[X_0]}{\varepsilon} = \frac{T - 1}{\varepsilon} = O(T).$$

□

Proposition 1. *In Algorithm 4, simulating data takes $O(nT^5)$ operations.*

Proof. By Claim 3, we know that simulating a network under the birth–death–hybridization process, requires T iterations to generate a network with T taxa.

In each iteration, hybridization, birth, and death events occur with different frequencies. However, all events take constant time since they involve, at most, the creation of three new nodes (in the case of a hybridization event) or simple rearrangements—each of which requires constant time. Therefore, the time complexity per network is $O(1) \times O(T) = O(T)$.

We generate up to $4n$ networks to ensure that at least n of them are level-1 networks, resulting in an overall time complexity of $O(nT)$ for network generation.

For each network, we traverse all the branch lengths to rescale them to a specified average value. A level-1 network contains three branch lengths per hybridization event, giving $3H$, and approximately $2T - 3$ branch lengths from the underlying displayed tree, yielding a total

of $O(3H + 2T)$ passes per network. By Claim 2, we know that $T > H$, which implies:

$$O(3H + 2T) \leq O(5T) = O(T)$$

Thus, over n networks, the total complexity for branch length scaling is $O(nT)$.

Finally, evaluating the objective function takes $O(T^5)$ time per network (see Section A.1.1), leading to $O(nT^5)$ time for all networks.

Putting it all together—network generation, branch length transformation, and objective function evaluation—the total time complexity is:

$$O(nT) + O(nT) + O(nT^5) \leq O(3nT^5) = O(nT^5)$$

This dominates the cost of writing and reading the generated matrix $(\mathbf{X}, \mathbf{y})$ (see Eq. 2.2) to and from memory, which takes $O(nT^4)$ for n observations and T^4 columns. □

Proposition 2. *In Algorithm 4, fitting the ensemble takes $O(MT^4\, n \log n)$ operations.*

Proof. The time complexity of constructing a regression tree primarily involves two components (Hastie et al., 2009, pp. 334):

i) Sorting the p feature columns, which takes $O(p\, n \log n)$ time.

ii) Performing the split computations, which takes $O(dnp)$ time for a tree of depth d.

In our study, we use shallow trees (small d), so the split computation simplifies to $O(np)$. Thus, the dominant cost of constructing the tree becomes the sorting step:

$$O(p\, n \log n) + O(np) \leq O(2p\, n \log n) = O(p\, n \log n)$$

For predictions, evaluating a single observation requires $O(d)$ operations—one per level of the tree. With n observations and shallow trees, the prediction complexity becomes $O(nd) = O(n)$.

These two dominant operations are repeated M times in the ISLE algorithm, so the total time complexity is:

$$O(M) \times [O(p\, n \log n) + O(n)] = O(Mp\, n \log n)$$

Finally, we assume that $p = \frac{1}{3}T^4$, since p corresponds to the number of rows in the concordance factor table, which is on the order of T^4. Substituting this into the expression yields the overall complexity:

$$O(M\frac{1}{3}T^4\, n \log n) \leq O(MT^4\, n \log n)$$

□

Proposition 3. *In Algorithm 4, post-processing takes $O(nM)$ operations.*

Proof. We implemented Elastic Net using the naive coordinate descent method (Hastie et al., 2015) and it has a time complexity of $O(np)$, where n is the number of samples and p is the number of predictors. In our setting, the predictors correspond to the M models in the ensemble, i.e., $p = M$. Therefore, the total time complexity for this step is $O(nM)$. □

Proposition 4. *In Algorithm 4, row selection takes $O(nM + \rho T^4)$ operations.*

Proof. We consider two scenarios: one where λ_k is pre-specified, and another where it is selected by minimizing test error.

Case 1 (Selection with a pre-specified λ_k)**.** *Algorithm 3 contains three main steps inside its loop:*

i) Checking if a regression tree is inactive, which takes constant time $O(1)$.

ii) Extracting the index set from split variables used in the regression tree, which takes $O(d)$ time, where d is the (fixed) depth of the tree.

iii) Merging these indices into a global index set. Using a hash table, this takes $O(d)$ in expectation.

Each iteration thus takes $O(d)$ time, and the loop runs for M models. Since d is constant, the total time complexity is:

$$O(M) \times [O(1) + O(d) + O(d)] \leq O(3dM) = O(M)$$

Case 2 (Selection of λ_k via test error minimization)**.** *In Algorithm A.4, we first generate the matrix $\mathbf{T}_{test} \in \mathbb{R}^{n \times M}$ from unseen test data. As discussed previously, predicting outcomes from M models over n samples requires $O(nM)$ operations.*

Within the loop over sparsity levels $\lambda_i \in \Lambda$, we compute the weighted prediction $\mathbf{T}_{test}\mathbf{w}^{\lambda_i}$, which also takes $O(nM)$ time. Since the number of sparsity values $|\Lambda| = k$ is constant, the total complexity of this loop is $O(knM) = O(nM)$. Other operations (e.g., error comparison and assignment) are negligible in comparison.

According to Case 1, generating the index set I^{λ_k} takes $O(M)$. Therefore, the overall complexity for this case is:

$$O(nM) + O(M) \leq O(2nM) = O(nM)$$

The worst-case scenario in terms of computational cost occurs when λ_k is selected via error minimization. Thus, we take $O(nM)$ as the complexity of this part.

Finally, writing the selected indices I^{λ_k} to a file involves looping through all selected features. Since these correspond to a fraction $\rho \in (0, 1]$ of the T^4 total rows in the concordance factor table, this step adds $O(\rho T^4)$ operations. Hence, the overall complexity for row selection is: $O(nM + \rho T^4)$ □

Proposition 5. *Cross-validation for the ISLE algorithm takes $O(MT^4\, n \log n)$ operations.*

Proof. To perform cross-validation, it is necessary to first streamline the fitting process—which

Algorithm A.4 Select λ_k by minimizing test error

Generate matrix $\mathbf{T}_{\text{test}}$ from unseen (test) data as described in Section 2.3.4.
Initialize: $\lambda_k \leftarrow \emptyset$, $\varepsilon_k \leftarrow \infty$
for $\lambda_i \in \Lambda$ **do**
    Compute prediction error on test data:

$$\varepsilon_i = \sqrt{\frac{1}{n} \left\| \mathbf{y}_{\text{test}} - \mathbf{T}_{\text{test}} \mathbf{w}^{\lambda_i} \right\|_2^2}$$

    if $\varepsilon_i < \varepsilon_k$ **then**
        $\lambda_k \leftarrow \lambda_i$
        $\varepsilon_k \leftarrow \varepsilon_i$
    end if
end for
return λ_k

includes both ensemble generation and post-processing via Elastic Net—in order to evaluate a given hyperparameter set on l-folds of the data. Let n_l denote the approximate number of observations in each fold. Then, by Propositions 2 and 3, the total number of operations required to fit a hyperparameter set on n_l observations is

$$O(MT^4\, n_l \log n_l + Mn_l).$$

By Proposition 4, Case 2, adding the cost of error evaluation, this becomes

$$O(MT^4\, n_l \log n_l + 2Mn_l).$$

Let c be the total number of hyperparameter sets to evaluate on each of the l-folds. This results in a total of cl evaluations. Note that cl is a positive constant since $l = 5$ and c is capped at 100 due to the use of a random sampling strategy for parameter combinations (Bergstra and Bengio, 2012). Therefore, the total computational complexity for fitting and

evaluation error during cross-validation is:

$$\begin{aligned} O\left(cl \cdot \{MT^4\, n_l \log n_l + 2Mn_l\}\right) &\leq O\left(cl \cdot \{MT^4\, n \log n + 2Mn\}\right) \\ &\leq O(cl \cdot 3MT^4\, n \log n) \\ &= O(MT^4\, n \log n). \end{aligned}$$

We store all error vectors—where each item corresponds to the error for a given sparsity parameter λ_k—in a linked list, so the total cost of appending them into the list is $O(cl)$.

Next, we use a hash table with keys corresponding to hyperparameter sets to compute the average error across all folds. For each key, this requires sum vectors of length k (the number of λ_k values in the Elastic Net path), and this operation is repeated cl times. Thus, the total complexity for computing the fold-average errors is: $O(clk)$.

We then make one pass over the hash table to identify the key (i.e., the hyperparameter set) with the lowest average error. This is similar to what is done in Algorithm A.4. For each iteration, finding the minimum in the error vector takes $O(k)$, and since there are l folds, the total complexity of this process is $O(lk)$.

Putting everything together—including fitting and error evaluation—the total complexity of cross-validation for selecting the best hyperparameter set is:

$$\begin{aligned} O(MT^4\, n \log n) + O(cl) + O(clk) + O(lk) &\leq O(MT^4\, n \log n) + O(3clk) \\ &= O(MT^4\, n \log n) + O(1) \\ &\leq O(2MT^4\, n \log n) \\ &= O(MT^4\, n \log n). \end{aligned}$$

□

Proposition 6. *If we assume the number of SNaQ iterations is at most on the order of T,*

then, in Algorithm 4, reconstructing the network takes $O(\rho T^7)$ operations.

Proof. We essentially adopt the same assumptions outlined in Section A.1.2. Specifically, the parameter optimization from Eq. A.1 can be rewritten as:

$$\begin{aligned} O\left(m\rho T^5\right) + O\left(m^2\right) + O\left(\rho T^5\right) &= O\left(\{T+3H\}\rho T^5\right) + O\left(\{T+3H\}^2\right) + O\left(\rho T^5\right) \\ &\leq O\left(4T \times \rho T^5\right) + O\left(16T^2\right) + O\left(\rho T^5\right) \\ &= O\left(4\rho T^6\right) + O\left(16T^2\right) + O\left(\rho T^5\right) \\ &\leq O\left(21\rho T^6\right) \\ &= O\left(\rho T^6\right) \end{aligned} \tag{A.2}$$

Assuming the number of optimization iterations is at most T, the total number of operations for this step is:

$$\underbrace{O(\rho T^6)}_{\substack{\text{Parameter} \\ \text{optimization}}} \times \underbrace{O(T)}_{\substack{\text{Number of} \\ \text{iterations}}} = O(\rho T^7)$$

□

Theorem 1 (Overall Time Complexity of Algorithm 4)**.** *Let T be the number of taxa, n be the number of simulations, M the number of used regression trees, and $\rho \in (0, 1]$ the fraction of selected CFs rows used in reconstruction. Assume the number of SNaQ iterations is at most on the order of T. Then, the total time complexity of Algorithm 4 is $O\left(nT^5 + MT^4 n \log n + \rho T^7\right)$.*

Proof. We count the operations contributed by each step:

$$
\begin{aligned}
O\left(nT^5\right) &\quad \text{(Simulate data, Proposition 1)} \\
+ O\left(MT^4\, n \log n\right) &\quad \text{(Cross-validation, Proposition 5)} \\
+ O\left(MT^4\, n \log n\right) &\quad \text{(Fit ensemble, Proposition 2)} \\
+ O(nM) &\quad \text{(Post-process, Proposition 3)} \\
+ O(nM + \rho T^4) &\quad \text{(Row selection, Proposition 4)} \\
+ O(\rho T^7) &\quad \text{(Reconstruct network, Proposition 6)}
\end{aligned}
$$

Combining and simplifying these terms:

$$
\begin{aligned}
O(nT^5) + O(2MT^4 n \log n + 2nM) + O(\rho T^4 + \rho T^7) &\leq O(nT^5) + O(4MT^4 n \log n) + O(2\rho T^7) \\
&= O(nT^5) + O(MT^4 n \log n) + O(\rho T^7) \\
&= O\left(nT^5 + MT^4 n \log n + \rho T^7\right).
\end{aligned}
$$

□

A.3 Qsin speed-up analysis

Corollary 1. *Let T be the number of taxa. Under the same assumptions outlined in Theorem 1, and as $T \to \infty$, the worst-case scenario yields a speed-up factor of $O(1)$ (i.e., no speed-up) and the best-case scenario yields a speed-up factor of $O(T^3)$ relative to SNaQ.*

Proof. Under the same assumptions outlined in Theorem 1, the asymptotic complexity of the SNaQ algorithm is $O(T^7)$ (see Supplementary Material A.1 for details). We now consider two extreme cases.

Case 1 (Worst-case scenario)**.** *Suppose the entire dataset is selected, which is on the order*

of T^4. Then,

$$\rho = O\left(\frac{T^4}{T^4}\right) = O(1).$$

In this case, the runtime complexity from Theorem 1 becomes

$$O(nT^5 + T^4Mn\log n + \rho T^7) = O(nT^5 + T^4Mn\log n + T^7).$$

The corresponding speed-up factor as a function of T is

$$\lim_{T\to\infty} \frac{T^7}{nT^5 + T^4Mn\log n + T^7} \leq \lim_{T\to\infty} \frac{nT^5 + T^4Mn\log n + T^7}{nT^5 + T^4Mn\log n + T^7} = O(1).$$

Hence, there is essentially no speed-up.

Case 2 (Best-case scenario)**.** *Suppose the algorithm selects only the minimum number of rows required to cover all species. Since each row contains four species, this number is approximately $\lceil T/4 \rceil$. In this case,*

$$\rho = O\left(\frac{\lceil T/4 \rceil}{T^4}\right) = O\left(\frac{1}{T^3}\right).$$

The runtime complexity then becomes

$$O(nT^5 + T^4Mn\log n + \rho T^7) = O(nT^5 + T^4Mn\log n + T^4).$$

The resulting speed-up factor is

$$\lim_{T\to\infty} \frac{T^7}{nT^5 + T^4Mn\log n + T^4} = \lim_{T\to\infty} \frac{1}{\frac{nT}{T^3} + \frac{Mn\log n}{T^3} + \frac{1}{T^3}} \leq \lim_{T\to\infty} \frac{1}{3/T^3} = O(T^3).$$

Therefore, the overall speed-up lies between $O(1)$ (Case 1) and $O(T^3)$ (Case 2). □

A.4 Supplementary Figures

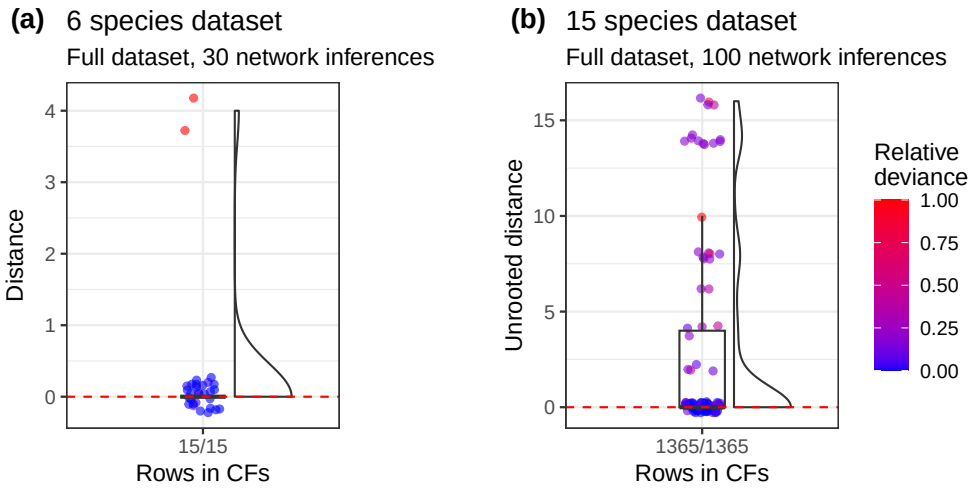


Figure A.2: Accuracy using all rows of the CFs table for both simulated datasets. **(a)** shows 93.3% accuracy for the first simulated dataset, where each repetition used 10 independent runs. **(b)** shows 69% accuracy for the second simulated dataset. Relative deviance denotes the normalized log-pseudolikelihood score.

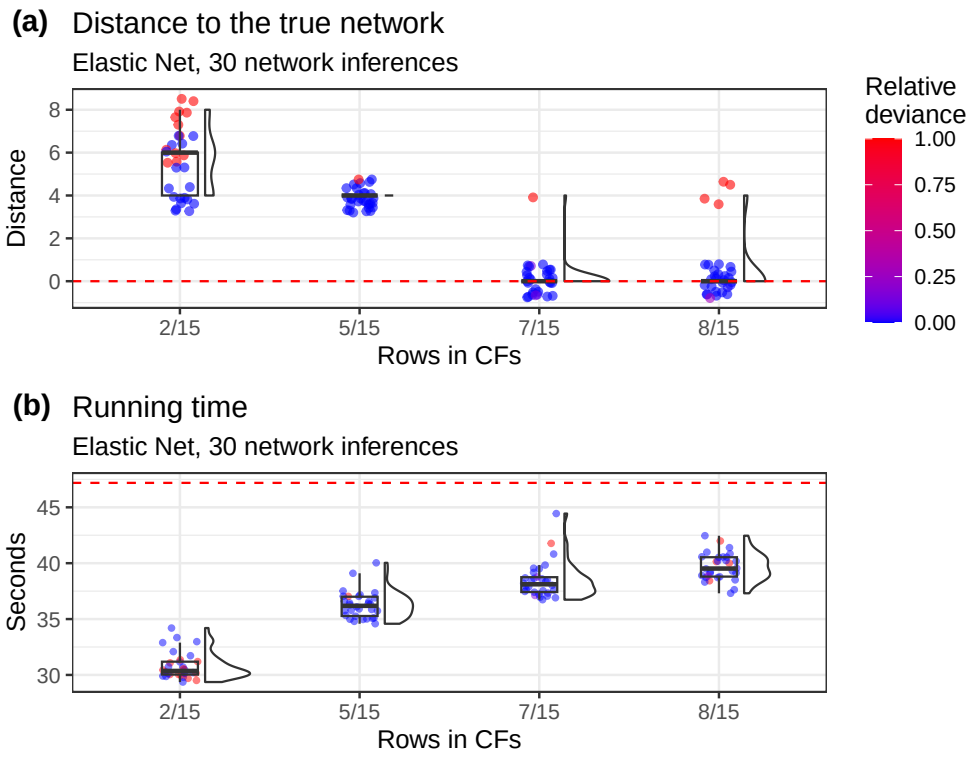


Figure A.3: Accuracy of reconstruction using subsampling guided by Elastic Net when branch lengths come from an exponential distribution draw with an average of 1.25 on a CFs table with 15 rows. **(a)** shows the distance to the true network. With 8 rows, the true network is recovered 86% of the 30 repetitions. The red dashed line indicates zero distance. **(b)** shows inference time for different row selection sizes. The red dashed line indicates the median time required to reconstruct the network using the full dataset, which is approximately 47.2 seconds. Each repetition used 10 independent runs. Relative deviance denotes the normalized log-pseudolikelihood score.

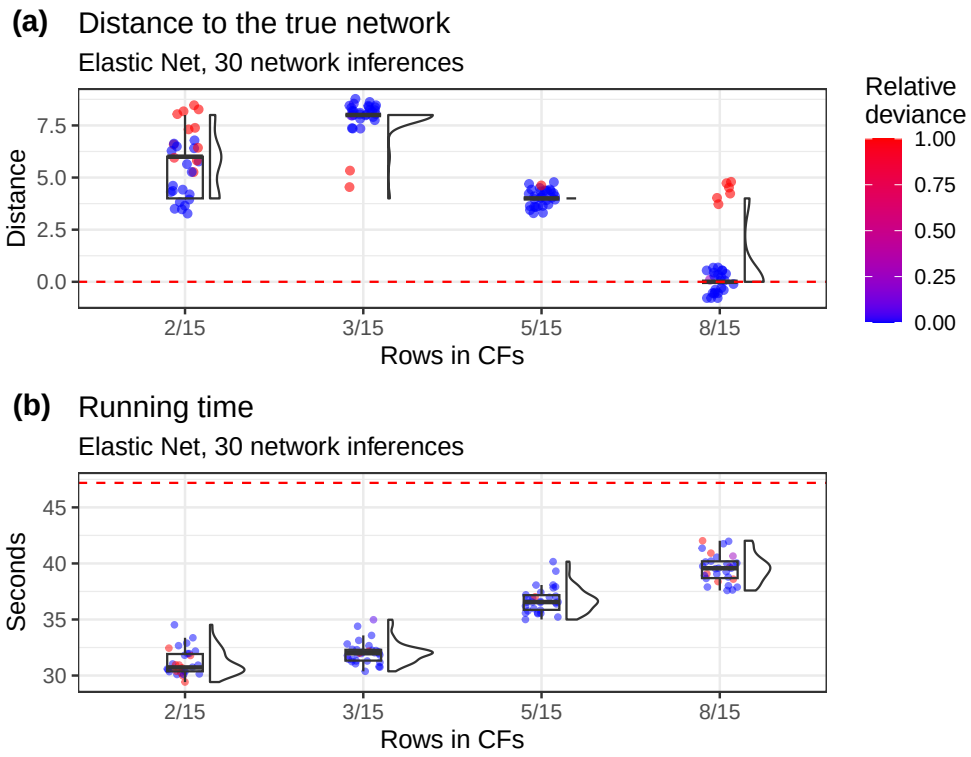


Figure A.4: Accuracy of reconstruction using subsampling guided by Elastic Net when branch lengths are not modified and come directly from the SiPhyNetworks on a CFs table with 15 rows. **(a)** shows the distance to the true network. With 8 rows, the true network is recovered 80% of the 30 repetitions. The red dashed line indicates zero distance. **(b)** shows inference time for different row selection sizes. The red dashed line indicates the median time required to reconstruct the network using the full dataset, which is approximately 47.2 seconds. Each repetition used 10 independent runs. Relative deviance denotes the normalized log-pseudolikelihood score.

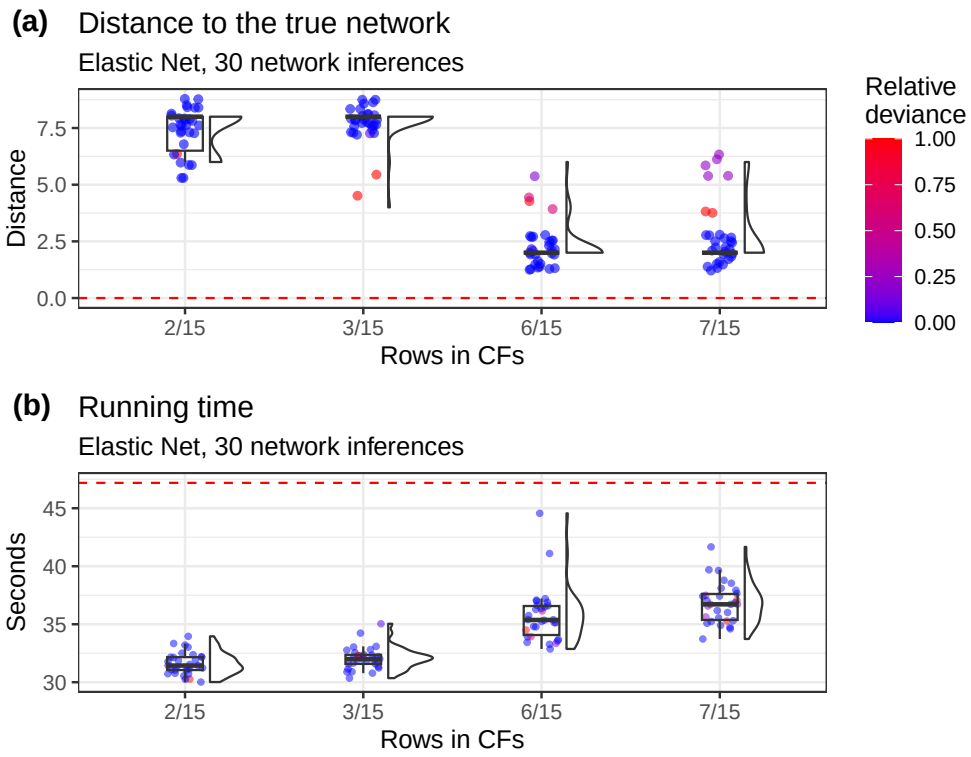


Figure A.5: Accuracy of reconstruction using subsampling guided by Elastic Net when branch lengths of the random network topologies are optimized on a CFs table with 15 rows. **(a)** shows the distance to the true network. With 7 rows, the true network is recovered 0% of the 30 repetitions. The red dashed line indicates zero distance. **(b)** shows inference time for different row selection sizes. The red dashed line indicates the median time required to reconstruct the network using the full dataset, which is approximately 47.2 seconds. Each repetition used 10 independent runs. Relative deviance denotes the normalized log-pseudolikelihood score.

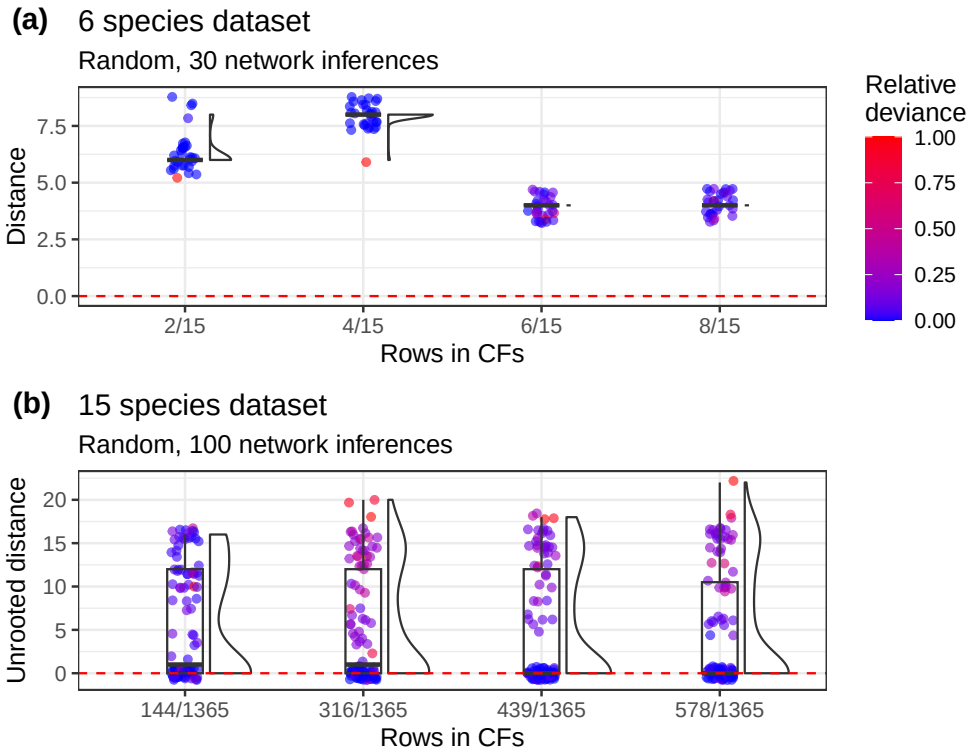


Figure A.6: Accuracy for uniform random selection of rows. **a)** Accuracy of randomly selected rows with the same sizes as in Fig. 2.2, where each repetition used 10 independent runs. **b)** Accuracy of randomly selected rows with the same size as in Fig. 2.5. Relative deviance denotes the normalized log-pseudolikelihood score.

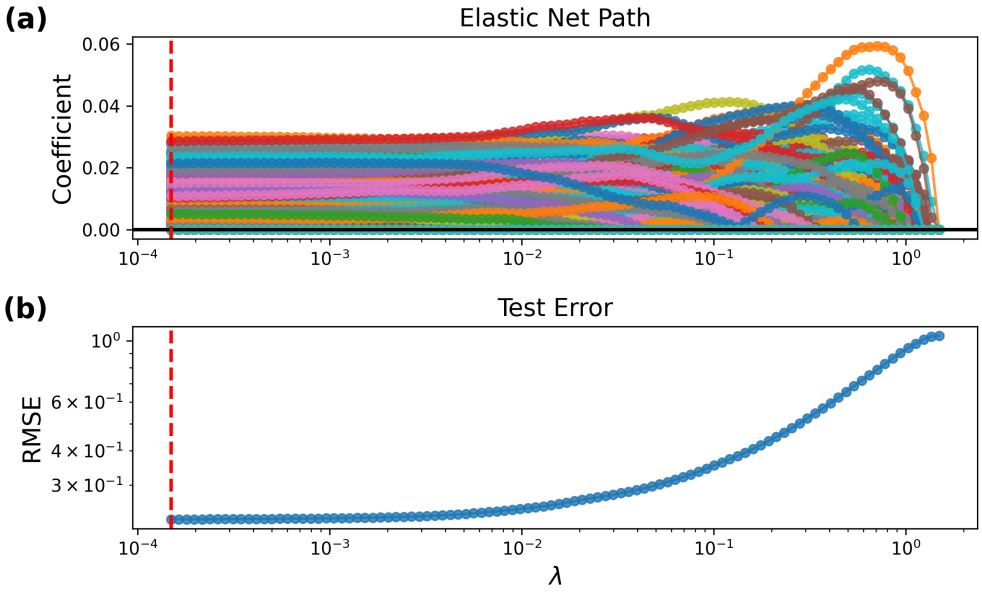


Figure A.7: The Elastic Net path applied to the ISLE ensemble on the *Xiphophorus* dataset, which involves a CFs table with 10,626 rows and 24 species. Hyperparameters were set to $\alpha = 0.5$, 2 leaf nodes, $\eta = 0.1487$, and $\nu = 0.0775$. In **(a)**, each colored line represents a regression tree. From right to left, we observe that when λ is at its largest value, no regression trees are selected — as indicated by all colored lines remaining at zero. The red dashed vertical line marks the point along the path with minimun testing error. **(b)** shows the testing RMSE along the elastic net path, reaching its minimun testing error when 788 out of 4,000 regression trees are active—corresponding to 763 rows in the CFs table.

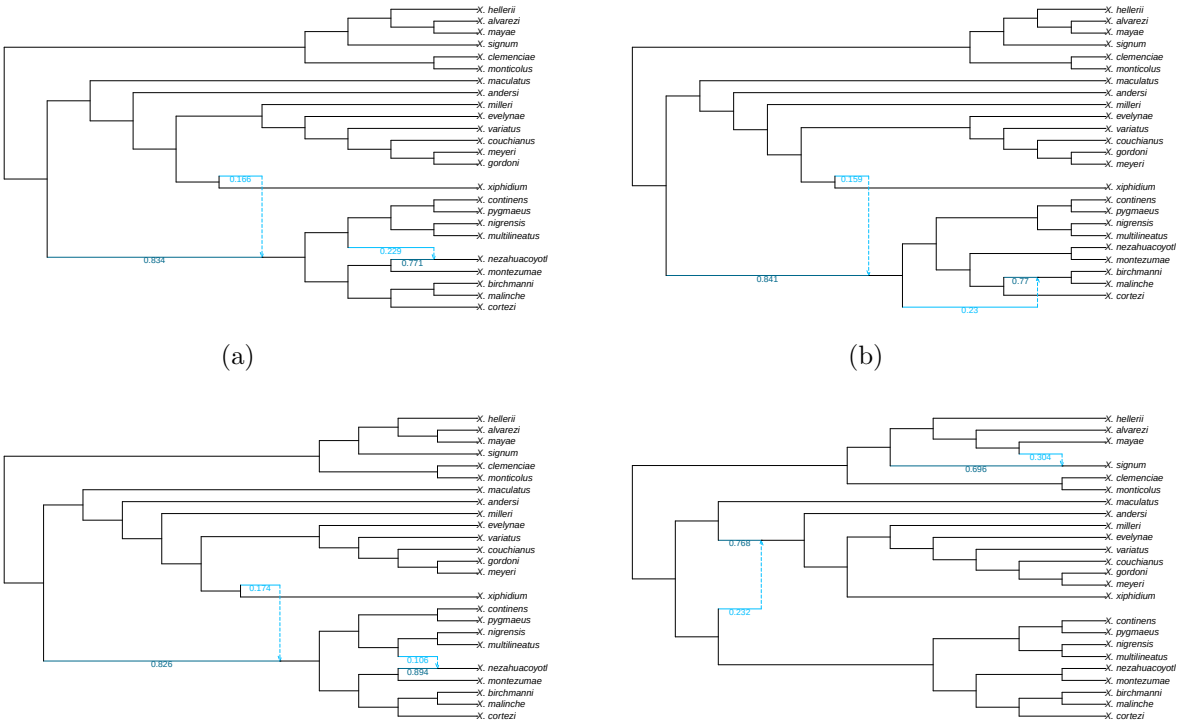


Figure A.8: Phylogenetic network reconstruction for the *Xiphophorus* dataset using the subsample of 763 rows (7.81% of the CFs table), with log-pseudolikelihood scores of **(a)** −758.770, **(b)** −775.370, **(c)** −798.085, and **(d)** −996.226. The subsample is identical to that used in Fig. 2.6.

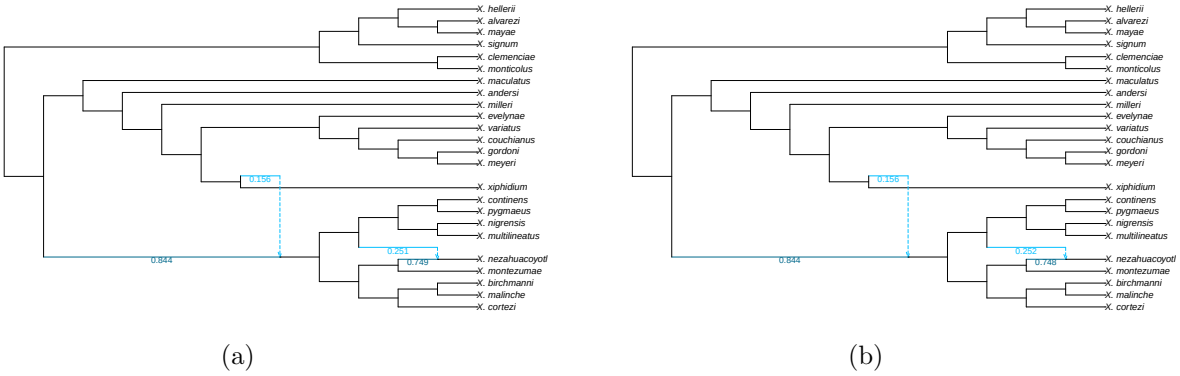
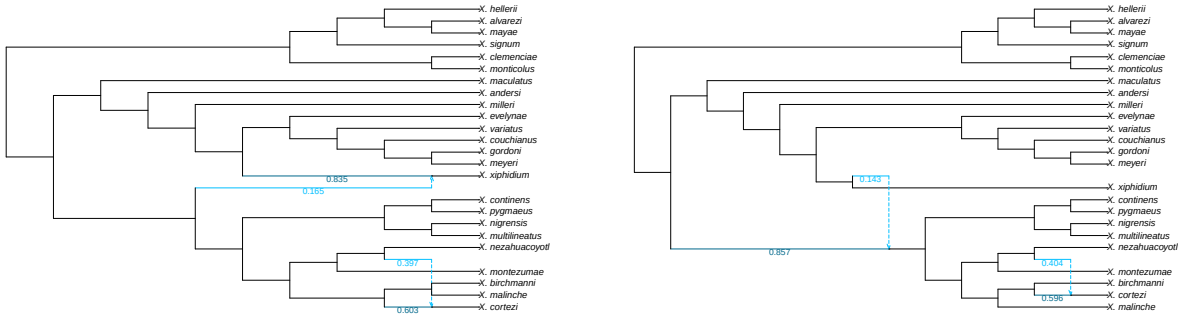


Figure A.9: Phylogenetic network reconstruction for the *Xiphophorus* dataset using the subsample of 763 rows (7.81% of the CFs table), with **(a)** 25 and **(b)** 30 independent runs. The corresponding log-pseudolikelihood scores were −709.162 and −709.158, respectively. The subsample is identical to that used in Fig. 2.6.



(a) (b)

Figure A.10: Phylogenetic network reconstruction for the *Xiphophorus* dataset using a random subsample of 763 rows (7.81% of the CFs table), with **(a)** 25 and **(b)** 30 independent runs. The corresponding log-pseudolikelihood scores were −908.548 and −795.571, respectively.

A.5 Constrained Elastic Net for accounting taxa frequencies

In Eq. 2.3, we can constrain $\mathbf{w}$ to take values between 0 and 1 with the following condition:

$$\mathbf{1}^\top \mathbf{w} = 1 \quad \text{and} \quad \mathbf{w} \geq \mathbf{0},$$

where $\mathbf{0} \in \mathbb{R}^M$ and $\mathbf{1} \in \mathbb{R}^M$ are column vectors of zeros and ones, respectively. This is a common technique in stacked generalization, where $\mathbf{w}$ can be interpreted as the posterior model probabilities for the ensemble (Hastie et al., 2009, pp. 290). When $\mathbf{w}$ takes values in $[0, 1]$, the weighted sum $\mathbf{z}_t^\top \mathbf{w}$ can be interpreted as the weighted average frequency of taxon t, with the weights given by $\mathbf{w}$.

To enforce a minimum frequency, we add the linear constraint:

$$\mathbf{z}_t^\top \mathbf{w} \geq c \quad \text{for all} \quad t \in \{1, \ldots, T\},$$

which ensures that the average frequency of each taxon among the T taxa is at least c. For convenience, let $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_T]^\top \in \mathbb{R}^{T \times M}$ denote the matrix whose rows are the vectors $\mathbf{z}_t^\top$ for $t \in \{1, \ldots, T\}$. With this notation, we can then incorporate the above constraints into Eq. 2.3 using the following equality and inequality constraints:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{n}\|\mathbf{y} - \mathbf{T}\mathbf{w}\|_2^2 + \lambda \left\{ \frac{1}{2}(1-\alpha)\|\mathbf{w}\|_2^2 + \alpha\|\mathbf{w}\|_1 \right\} \\ &\text{subject to} \quad \mathbf{1}^\top \mathbf{w} = 1 \quad \text{and} \quad \begin{bmatrix} \mathbf{I} \\ \mathbf{Z} \end{bmatrix} \mathbf{w} \geq \begin{bmatrix} \mathbf{0} \\ \mathbf{c} \end{bmatrix}, \end{aligned}$$

where $\mathbf{c} \in \mathbb{R}^T$ is a column vector containing the desired minimum weighted averages for each taxon, and $\mathbf{I} \in \mathbb{R}^{M \times M}$ is the identity matrix. Multiplying both sides of the inequality constraint by -1 yields the standard form of the constrained Lasso, for which the entire

solution path can be efficiently computed, including the Elastic Net case (see Eq. 6 in Gaines et al., 2018).

Supplemental references

Abbott, S., 2015: *Understanding analysis*. Springer.

Bergstra, J., and Y. Bengio, 2012: Random search for hyper-parameter optimization. *The journal of machine learning research*, **13 (1)**, 281–305.

Fogg, J., E. S. Allman, and C. Ané, 2023: PhyloCoalSimulations: A simulator for network multispecies coalescent models, including a new extension for the inheritance of gene flow. *Systematic Biology*, **72 (5)**, 1171–1179, https://doi.org/10.1093/sysbio/syad030.

Gaines, B. R., J. Kim, and H. Zhou, 2018: Algorithms for fitting the constrained lasso. *Journal of Computational and Graphical Statistics*, **27 (4)**, 861–871.

Hastie, T., R. Tibshirani, and J. Friedman, 2009: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed., Springer Series in Statistics, Springer, New York.

Hastie, T., R. Tibshirani, and M. Wainwright, 2015: *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, Boca Raton, FL.

He, J., and X. Yao, 2004: A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, **3**, 21–35.

Huelsenbeck, J. P., 2018: Programming for evolutionary biologists. *University of California, Berkeley*.

Justison, J. A., and T. A. Heath, 2023: Exploring the distribution of phylogenetic networks generated under a birth-death-hybridization process. *Bulletin of the Society of Systematic Biologists*, **2 (3)**, 1–22.

Justison, J. A., C. Solis-Lemus, and T. A. Heath, 2023: Siphynetwork: An r package for simulating phylogenetic networks. *Methods in Ecology and Evolution*, **14 (7)**, 1687–1698.

Kötzing, T., 2014: Concentration of first hitting times under additive drift. *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 1391–1398.

Lengler, J., 2019: Drift analysis. *Theory of evolutionary computation: Recent developments in discrete optimization*, Springer, 89–131.

Powell, M. J., and Coauthors, 2009: The bobyqa algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, **26**, 26–46.

Yang, Z., 2014: *Molecular evolution: a statistical approach*. Oxford University Press.

Appendix B

Supplementary Material: Non-linear phylogenetic regression using regularized kernels

B.1 Phylogenetic Generalized Least Squares

In this section we review results obtained from Grafen (1989); Garland and Ives (2000) for deriving the phylogenetic linear model, and give the mathematical justification for the usage of the **P** matrix for data transformation. Let the prediction of a linear model be affected by a random noise $\varepsilon \in \mathbb{R}^{n\times 1}$ that is not normally distributed:

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon \ , \tag{B1}$$

where $\mathbf{y} \in \mathbb{R}^{n\times 1}$ is the response variable, $\mathbf{X} \in \mathbb{R}^{n\times p}$ is the matrix with instances, and $\beta \in \mathbb{R}^{p\times 1}$ is the coefficients vector.

Assume there exist an invertible matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ such that its multiplication into both sides of equation B1 generates a new random noise $\mathbf{E} \in \mathbb{R}^{n\times 1}$ that follows a multivariable normal distribution with mean vector $\mathbf{0} \in \mathbb{R}^{n\times 1}$ and covariance matrix $\sigma^2\mathbf{I} \in \mathbb{R}^{n\times n}$:

$$\mathbf{E} = \mathbf{P}\varepsilon = \mathbf{P}(\mathbf{y} - \mathbf{X}\beta) \ , \tag{B2}$$
$$\mathbf{E} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) \ .$$

We can rewrite equation B2 as $\mathbf{y} = \mathbf{X}\beta + \mathbf{P}^{-1}\mathbf{E}$.

B.1.1 Finding distribution parameters of y

We can estimate the new covariance matrix and mean vector of $\mathbf{X}\beta + \mathbf{P}^{-1}\mathbf{E}$ from the distribution of $\mathbf{E}$. For the $\mathbf{P}^{-1}\mathbf{E}$ term, the mean vector and covariance matrix would be:

$$\begin{aligned}
\mathbb{E}[\mathbf{P}^{-1}\mathbf{E}] &= \mathbf{P}^{-1}\,\mathbb{E}[\mathbf{E}] = \mathbf{0} \ , \\
\mathrm{Cov}[\mathbf{P}^{-1}\mathbf{E}] &= \mathbb{E}[(\mathbf{P}^{-1}\mathbf{E} - \mathbb{E}[\mathbf{P}^{-1}\mathbf{E}])(\mathbf{P}^{-1}\mathbf{E} - \mathbb{E}[\mathbf{P}^{-1}\mathbf{E}])^\top] \\
&= \mathbb{E}[(\mathbf{P}^{-1})(\mathbf{E} - \mathbb{E}[\mathbf{E}])(\mathbf{E} - \mathbb{E}[\mathbf{E}])^\top(\mathbf{P}^{-1})^\top] \\
&= \mathbf{P}^{-1}\mathrm{Cov}[\mathbf{E}](\mathbf{P}^{-1})^\top \\
&= \mathbf{P}^{-1}\sigma^2\mathbf{I}(\mathbf{P}^{-1})^\top \\
&= \sigma^2\mathbf{P}^{-1}(\mathbf{P}^{-1})^\top \ ,
\end{aligned}$$

without loss of generality, notice that we can define $\mathrm{Cov}[\varepsilon]$ as:

$$\begin{aligned}
\mathrm{Cov}[\varepsilon] &= \mathrm{Cov}[\mathbf{P}^{-1}\mathbf{P}\varepsilon] \\
&= \mathrm{Cov}[\mathbf{P}^{-1}\mathbf{E}] \\
&= \sigma^2\mathbf{P}^{-1}(\mathbf{P}^{-1})^\top \ .
\end{aligned} \tag{B3}$$

likewise, we can define $\mathbb{E}[\varepsilon]$ as:

$$\begin{aligned}\mathbb{E}[\varepsilon] &= \mathbb{E}[\mathbf{P}^{-1}\mathbf{P}\varepsilon] \\ &= \mathbb{E}[\mathbf{P}^{-1}\mathbf{E}] \\ &= \mathbf{P}^{-1}\,\mathbb{E}[\mathbf{E}] = \mathbf{0}\ .\end{aligned} \tag{B4}$$

For the complete equation $\mathbf{y} = \mathbf{X}\beta + \mathbf{P}^{-1}\mathbf{E}$, the mean vector and covariance matrix would be:

$$\begin{aligned}\mathbb{E}[\mathbf{X}\beta + \mathbf{P}^{-1}\mathbf{E}] &= \mathbf{X}\beta + \mathbb{E}[\mathbf{P}^{-1}\mathbf{E}] \\ &= \mathbf{X}\beta\ , \\ \mathrm{Cov}[\mathbf{X}\beta + \mathbf{P}^{-1}\mathbf{E}] &= \mathrm{Cov}[\mathbf{P}^{-1}\mathbf{E}] \\ &= \sigma^2\mathbf{P}^{-1}(\mathbf{P}^{-1})^\top\end{aligned}$$

and its distribution is

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{P}^{-1}\mathbf{E} \sim \mathcal{N}(\mathbf{X}\beta, \sigma^2\mathbf{P}^{-1}(\mathbf{P}^{-1})^\top)\ .$$

Let the covariance matrix of $\mathbf{X}$ be $\mathbf{\Omega} \in \mathbb{R}^{n\times n}$, where $\mathbf{\Omega}$ is symmetric matrix and, thus, positive semi-definite matrix. Let $\mathbf{P} = \mathbf{Q}\mathbf{\Lambda}^{-1/2}\mathbf{Q}^\top$, where $\mathbf{Q}$ and $\mathbf{\Lambda}$ are the eigenvectors and eigenvalues matrices of $\mathbf{\Omega}$, respectively, such that $\mathbf{P}^{-1}(\mathbf{P}^{-1})^\top = \mathbf{\Omega}$ (Tung Ho and Ané, 2014). Then, the covariance of $\mathbf{y}$ can be expressed in function of $\mathbf{\Omega}$:

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{P}^{-1}\mathbf{E} \sim \mathcal{N}(\mathbf{X}\beta, \sigma^2\mathbf{\Omega})\ . \tag{B5}$$

B.1.2 Finding optimal β

Assume $\mathbf{y}$ is independent and identically distributed (i.e., "i.i.d." assumption). Then, its density of equation B5 given by:

$$
\begin{aligned}
p(\mathbf{y} \mid \mathbf{X}, \beta) &= \frac{1}{(2\pi)^{n/2}} \frac{1}{|\sigma^2 \mathbf{\Omega}|^{1/2}} \exp\left\{ -\frac{1}{2} (\mathbf{y} - \mathbf{X}\beta)^\top (\sigma^2 \mathbf{\Omega})^{-1} (\mathbf{y} - \mathbf{X}\beta) \right\} \\
\ln p(\mathbf{y} \mid \mathbf{X}, \beta) &= \ln \left\{ \frac{1}{(2\pi)^{n/2}} \right\} + \ln \left\{ \frac{1}{|\sigma^2 \mathbf{\Omega}|^{1/2}} \right\} - \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\beta)^\top \mathbf{\Omega}^{-1} (\mathbf{y} - \mathbf{X}\beta) \\
\ln p(\mathbf{y} \mid \mathbf{X}, \beta) &= c_1 + c_2 - c_3 (\mathbf{y} - \mathbf{X}\beta)^\top \mathbf{\Omega}^{-1} (\mathbf{y} - \mathbf{X}\beta)
\end{aligned}
$$

where c_1, c_2, and c_3 are constants of the model that do not depend on β. From above equation we can clearly see that the maximization of $\ln p(\mathbf{y} \mid \mathbf{X}, \beta)$ depends on the minimization of right most term:

$$
\arg\max_{\beta} \ln p(\mathbf{y} \mid \mathbf{X}, \beta) = \arg\min_{\beta} (\mathbf{y} - \mathbf{X}\beta)^\top \mathbf{\Omega}^{-1} (\mathbf{y} - \mathbf{X}\beta) \ .
$$

From the above relationship, we can define our objective function as:

$$
J(\beta) = \frac{1}{n} (\mathbf{y} - \mathbf{X}\beta)^\top \mathbf{\Omega}^{-1} (\mathbf{y} - \mathbf{X}\beta) \tag{B6}
$$

Note that we have included the term 1/n, bringing the definition of the objective function closer to that of mean squared error. This term should not impact the optimal value, as multiplying the objective function by a positive constant does not alter its optimal solution (Boyd and Vandenberghe, 2004). Expanding above equation, differentiating it with respect to β, and setting it to zero, we can obtain our optimal β:

$$
\begin{aligned}
J(\beta) &= \frac{1}{n}(\mathbf{y}^\top \boldsymbol{\Omega}^{-1}\mathbf{y} - 2\beta^\top \mathbf{X}^\top \boldsymbol{\Omega}^{-1}\mathbf{y} + \beta^\top \mathbf{X}^\top \boldsymbol{\Omega}^{-1}\mathbf{X}\beta) \\
\frac{\partial J(\beta)}{\partial \beta} &= -\frac{2}{n}\mathbf{X}^\top \boldsymbol{\Omega}^{-1}\mathbf{y} + \frac{1}{n}(\mathbf{X}^\top \boldsymbol{\Omega}^{-1}\mathbf{X} + (\mathbf{X}^\top \boldsymbol{\Omega}^{-1}\mathbf{X})^\top)\beta = 0 \\
&\Rightarrow \beta = (\mathbf{X}^\top \boldsymbol{\Omega}^{-1}\mathbf{X})^{-1}(\mathbf{X}^\top \boldsymbol{\Omega}^{-1}\mathbf{y})
\end{aligned}
$$

B.1.3 Finding optimal β from transformed data

We can obtain the same optimal β by transforming data with the $\mathbf{P}$ matrix. Since we can rewrite $\boldsymbol{\Omega}^{-1}$ as $\mathbf{P}^\top\mathbf{P} = \boldsymbol{\Omega}^{-1}$, then the objective function (i.e., equation B6) can also can take this form:

$$
\begin{aligned}
J(\beta) &= \frac{1}{n}(\mathbf{y} - \mathbf{X}\beta)^\top \mathbf{P}^\top \mathbf{P}(\mathbf{y} - \mathbf{X}\beta) \\
&= \frac{1}{n}(\mathbf{P}\mathbf{y} - \mathbf{P}\mathbf{X}\beta)^\top (\mathbf{P}\mathbf{y} - \mathbf{P}\mathbf{X}\beta) \ .
\end{aligned}
$$

Let $\mathbf{y}^* = \mathbf{P}\mathbf{y}$ and $\mathbf{X}^* = \mathbf{P}\mathbf{X}$, such that we can rewrite above equation as:

$$
J(\beta) = \frac{1}{n}(\mathbf{y}^* - \mathbf{X}^*\beta)^\top (\mathbf{y}^* - \mathbf{X}^*\beta) \ , \tag{B7}
$$

whose solution, by least squares, generates the same optimal β as in the previous section. Furthermore, from equation B1 we can see the error can also get transformed by $\mathbf{P}$:

$$
\mathbf{E}^* = \mathbf{y}^* - \mathbf{X}^*\beta = \mathbf{P}\varepsilon \ , \tag{B8}
$$

and, from equation B3, the covariance of $\mathbf{E}^*$ is the identity matrix times a constant:

$$
\begin{aligned}
\text{Cov}[\mathbf{E}^*] &= \text{Cov}[\mathbf{P}\varepsilon] \\
&= \mathbf{P}\text{Cov}[\varepsilon]\mathbf{P}^\top \\
&= \mathbf{P}\sigma^2\mathbf{P}^{-1}(\mathbf{P}^{-1})^\top\mathbf{P}^\top \\
&= \sigma^2\mathbf{I} \ .
\end{aligned}
$$

Likewise, from equation B4, its mean vector is $\mathbf{0}$:

$$
\mathbb{E}[\mathbf{E}^*] = \mathbb{E}[\mathbf{P}\varepsilon] = \mathbf{P}\,\mathbb{E}[\varepsilon] = \mathbf{0} \ .
$$

B.1.3.1 Finding the intercept of uncentered data

Up to this point, we have assumed that data in Eq. B7 and Eq. B6 are centered. This assumption is convenient as it allows us to ignore the intercept, β_0 (Hastie et al., 2009). However, if necessary, we can recover this value after finding the optimal β from the centered data, using the average of the transformed data (Wang, 2022). In this section, we present a brief argument supporting this approach. The argument stems from the recognition that centering data do not change the slope values of β; it merely shifts the location of data to the origin.

Let $\mathbf{y}_{w,uc}$ and $\mathbf{X}_{w,uc}$ be the uncentered response and taxa observations values repectively such that Eq. B7 turns into:

$$
J(\beta) = \frac{1}{n}(\underbrace{\mathbf{y}_{w,uc} - \mathbf{1}\beta_0 - \mathbf{X}_{w,uc}\beta}_{\mathbf{e}})^\top(\mathbf{y}_{w,uc} - \mathbf{1}\beta_0 - \mathbf{X}_{w,uc}\beta) \ . \tag{B9}
$$

Let $\bar{y}_w \in \mathbb{R}^{1\times 1}$ be the average of $\mathbf{y}_{w,uc}$, let $\bar{\mathbf{X}}_w \in \mathbb{R}^{n\times p}$ be the matrix where each column

has the average of each column of $\mathbf{X}_{w,uc}$ repeated n times, and let $\mathbf{e} = \mathbf{y}_{w,uc} - \mathbf{1}\beta_0 - \mathbf{X}_{w,uc}\beta$ such that:

$$\begin{aligned}
\mathbf{e} &= \mathbf{y}_{w,uc} - \mathbf{1}\beta_0 - \mathbf{X}_{w,uc}\beta + \mathbf{1}\bar{y}_w - \mathbf{1}\bar{y}_w + \bar{\mathbf{X}}_w\beta - \bar{\mathbf{X}}_w\beta \\
&= \underbrace{\mathbf{y}_{w,uc} - \mathbf{1}\bar{y}_w}_{\mathbf{y}^*} + \underbrace{\mathbf{1}\bar{y}_w - \mathbf{1}\beta_0 - \bar{\mathbf{X}}_w\beta}_{-\mathbf{1}\beta_0^c} - (\underbrace{\mathbf{X}_{w,uc} - \bar{\mathbf{X}}_w}_{\mathbf{X}^*})\beta \\
&= \mathbf{y}^* - \mathbf{1}\beta_0^c - \mathbf{X}^*\beta \\
&= \mathbf{r} - \mathbf{1}\beta_0^c
\end{aligned}$$

where $\mathbf{1}\beta_0^c = -\mathbf{1}\bar{y}_w + \mathbf{1}\beta_0 + \bar{\mathbf{X}}_w\beta$ is the intercept from the centered data and $\mathbf{r} = \mathbf{y} - \mathbf{X}\beta$. Plugging this result back into Eq. B9, we have:

$$J(\beta) = \frac{1}{n}(\mathbf{r} - \mathbf{1}\beta_0^c)^\top(\mathbf{r} - \mathbf{1}\beta_0^c)$$

By least squares, we find that β_0^c is:

$$\begin{aligned}
\beta_0^c &= (\mathbf{1}^\top\mathbf{1})^{-1}\mathbf{1}^\top\mathbf{r} \\
&= \frac{1}{n}\mathbf{1}^\top\mathbf{y}^* - \frac{1}{n}\mathbf{1}^\top\mathbf{X}^*\beta \\
&= \frac{1}{n}\underbrace{\mathbf{1}^\top(\mathbf{y}_{w,uc} - \mathbf{1}\bar{y}_w)}_{0} - \frac{1}{n}\underbrace{\mathbf{1}^\top(\mathbf{X}_{w,uc} - \bar{\mathbf{X}}_w)}_{0}\beta \\
&= 0
\end{aligned}$$

Plugging back this result in the definition of β_0^c, we obtain:

$$
\begin{aligned}
\mathbf{0} &= -\mathbf{1}\bar{y}_w + \mathbf{1}\beta_0 + \bar{\mathbf{X}}_w\beta \\
\mathbf{1}\beta_0 &= \mathbf{1}\bar{y}_w - \mathbf{1}\bar{\mathbf{x}}_w^\top\beta \\
\implies \beta_0 &= \bar{y}_w - \bar{\mathbf{x}}_w^\top\beta
\end{aligned} \tag{B10}
$$

B.2 Kernel Ridge Regression

B.2.1 Regularized least squares

Assume β follows a multivariable normal distribution with mean vector $\mathbf{0} \in \mathbb{R}^{p\times 1}$ and covariance matrix $t^2\mathbf{I} \in \mathbb{R}^{p\times p}$ such that its density is given by:

$$
p(\beta) = \frac{1}{(2\pi)^{p/2}} \frac{1}{|t^2\mathbf{I}|^{1/2}} \exp\left\{ -\frac{1}{2t^2}\beta^\top\beta \right\} ,
$$

Since the density of equation B8 is given by:

$$
p(\mathbf{y}^* \mid \mathbf{X}^*, \beta) = \frac{1}{(2\pi)^{n/2}} \frac{1}{|\sigma^2\mathbf{I}|^{1/2}} \exp\left\{ -\frac{1}{2\sigma^2}(\mathbf{y}^* - \mathbf{X}^*\beta)^\top(\mathbf{y}^* - \mathbf{X}^*\beta) \right\} ,
$$

then we can approach the posterior estimation of β given $\mathbf{X}^*$ and $\mathbf{y}^*$ as:

$$
\begin{aligned}
p(\beta \mid \mathbf{X}^*, \mathbf{y}^*) &\propto p(\mathbf{y}^* \mid \beta, \mathbf{X}^*) p(\beta) \\
\ln p(\beta \mid \mathbf{X}^*, \mathbf{y}^*) &\propto \ln p(\mathbf{y}^* \mid \beta, \mathbf{X}^*) p(\beta) \\
&= \ln \left(\frac{1}{(2\pi)^{(n+p)/2}} \frac{1}{t^p \sigma^n} \exp \left\{ -\frac{1}{2\sigma^2} (\mathbf{y}^* - \mathbf{X}^* \beta)^\top (\mathbf{y}^* - \mathbf{X}^* \beta) - \frac{1}{2t^2} \beta^\top \beta \right\} \right) \\
&= \ln c_1 + \ln c_2 - \frac{1}{2\sigma^2} (\mathbf{y}^* - \mathbf{X}^* \beta)^\top (\mathbf{y}^* - \mathbf{X}^* \beta) - \frac{1}{2t^2} \beta^\top \beta
\end{aligned}
$$

where c_1 and c_2 represent constants of the model. Thus, the Maximum A Posterior (MAP) estimation of β is equivalent to:

$$
\arg\max_{\beta} \ln p(\beta \mid \mathbf{X}^*, \mathbf{y}^*) = \arg\min_{\beta} \frac{1}{n} (\mathbf{y}^* - \mathbf{X}^* \beta)^\top (\mathbf{y}^* - \mathbf{X}^* \beta) + \lambda \beta^\top \beta = J(\beta) \ , \tag{B11}
$$

where λ is $\sigma^2/(nt^2)$.

B.2.2 Addition of kernels into regularized least squares

We can rewrite the objective function $J(\beta)$ from equation B11 as follows:

$$
\begin{aligned}
J(\beta) &= \frac{1}{n} (\mathbf{y}^* - \mathbf{X}^* \beta)^\top (\mathbf{y}^* - \mathbf{X}^* \beta) + \lambda \beta^\top \beta \\
&= \frac{1}{n} \sum_{i=1}^{n} (\phi(a_i)^\top \beta - y_i^*)^2 + \lambda \beta^\top \beta \ ,
\end{aligned} \tag{B12}
$$

where $\phi(a_i)$ is the transformation i^{th} instance of the $\mathbf{X}^*$ matrix into a high dimensional space, and y_i^* is the i^{th} element of the $\mathbf{y}^*$ vector. Differentiating above equation, and setting it to zero:

$$\frac{\partial J(\beta)}{\partial \beta} = \frac{1}{n}\sum_{i=1}^{n} 2(\phi(a_i)^\top \beta - y_i^*)\phi(a_i) + 2\lambda\beta = 0$$
$$\Rightarrow \beta = \sum_{i=1}^{n} -\frac{1}{n\lambda}(\phi(a_i)^\top \beta - y_i^*)\phi(a_i) = \sum_{i=1}^{n} \alpha_i \phi(a_i) \ , \tag{B13}$$

where the multiplier α_i represents $-(\phi(a_i)^\top \beta - y_i^*)/(n\lambda)$. If we plug back equation B13 into equation B12, we can get the dual form of equation B12:

$$\begin{aligned} G(\alpha) &= \frac{1}{n}\sum_{i=1}^{n}\left(\phi(a_i)^\top \sum_{j=1}^{n}\alpha_j\phi(a_j) - y_i^*\right)^2 + \lambda\left(\sum_{i=1}^{n}\alpha_i\phi(a_i)\right)^\top\left(\sum_{j=1}^{n}\alpha_j\phi(a_j)\right) \\ &= \frac{1}{n}\sum_{i=1}^{n}\left\{\sum_{j=1}^{n}\alpha_j\phi(a_i)^\top\phi(a_j) - y_i^*\right\}^2 + \lambda\sum_{i=1}^{n}\left\{\sum_{j=1}^{n}\alpha_i\alpha_j\phi(a_i)^\top\phi(a_j)\right\} \ . \end{aligned} \tag{B14}$$

Let $\phi(a_i)^\top\phi(a_j) = k(i,j)$, where $k(i,j)$ is the kernel function over a_i and a_j. Let $i = 1$, such that

$$\begin{aligned} \sum_{j=1}^{n}\alpha_j\phi(a_1)^\top\phi(a_j) &= \alpha_1\phi(a_1)^\top\phi(a_1) + \cdots + \alpha_n\phi(a_1)^\top\phi(a_n) \\ &= \alpha_1 k(1,1) + \cdots + \alpha_n k(1,n) = \mathbf{k}_{1:}\alpha \ , \end{aligned}$$

where $\mathbf{k}_{i:}$ is the i^{th} row of the Gram matrix $\mathbf{K} \in \mathbb{R}^{n\times n}$, and $\alpha \in \mathbb{R}^{n\times 1}$ is the column vector containing all alphas. Then, we can rewrite the equation B14 as:

$$
\begin{aligned}
G(\alpha) &= \frac{1}{n}\sum_{i=1}^{n}\{\mathbf{k}_{i:}\alpha - y_i^*\}^2 + \lambda\sum_{i=1}^{n}\{\alpha_i\mathbf{k}_{i:}\alpha\} \\
&= \frac{1}{n}(\mathbf{K}\alpha - \mathbf{y}^*)^\top(\mathbf{K}\alpha - \mathbf{y}^*) + \lambda\alpha^\top\mathbf{K}\alpha \\
&= \frac{1}{n}(\alpha^\top\mathbf{K}^\top\mathbf{K}\alpha - 2\alpha^\top\mathbf{K}^\top\mathbf{y}^* + \mathbf{y}^{*\top}\mathbf{y}) + \lambda\alpha^\top\mathbf{K}\alpha \ .
\end{aligned}
$$

Differentiating above equation with respect to α, and set it to zero we have:

$$
\begin{aligned}
\frac{\partial G(\alpha)}{\partial\alpha} &= \frac{2}{n}\mathbf{K}^\top\mathbf{K}\alpha - \frac{2}{n}\mathbf{K}^\top\mathbf{y}^* + 2\lambda\mathbf{K}\alpha = 0 \\
&\Rightarrow \alpha = (\mathbf{K}^\top\mathbf{K} + \lambda n\mathbf{K})^{-1}\mathbf{K}^\top\mathbf{y}^* \ .
\end{aligned}
$$

If $\mathbf{K}$ is a positive semidefinite matrix, we can also rewrite α as follows:

$$
\alpha = (\mathbf{K} + \lambda n\mathbf{I})^{-1}\mathbf{y}^* \ .
$$

While throughout this section we assumed that the response variable is a column vector, it can also be shown that the kernel applies when we have multiple response variables (i.e., a matrix of response variables). We will demonstrate how in the next section.

B.2.3 Multi-output regression

In this section, we show that the results similar to those in the equations above also naturally hold for the multi-output regression setting, where each output is independently estimated (Borchani et al., 2015). We first need to demonstrate that the dual variable can be obtained and then construct the dual problem as we did above. However, in this case, the dual variables α will be a matrix instead of a column vector. If we are able to create the dual

problem, then we will have a linear problem in high-dimensional space. Since the linear function has a multi-output solution, the dual and therefore the original (primal) problem will also have a solution.

Consider the following primal objective function, which minimizes the mean squared error for multiple outputs:

$$J(\beta) = \frac{1}{nk} \sum_{c=1}^{n} \sum_{j=1}^{k} \{a_c^\top \beta_{:j} - y_{c,j}^*\}^2 + \lambda \sum_{r=1}^{p} \sum_{j=1}^{k} \beta_{r,j}^2 \ ,$$

Where k is the number of outputs, $y_{j,c}^*$ is the observed output j for the taxa observation c, and it is an element of the matrix $\mathbf{Y}^* \in \mathbb{R}^{n \times k}$ containing multiple output columns. $\beta_{:j}$ is the column j of the matrix $\beta \in \mathbb{R}^{p \times k}$, and $a_c^\top$ is a row vector from the taxa observation matrix $\mathbf{X}^* \in \mathbb{R}^{n \times p}$.

As we did in Eq. B12, we can increase the dimensionality as follows by including the function ϕ. Furthermore, we can re-write the above objective function as the following Frobenius norm:

$$\begin{aligned}
J(\beta) &= \frac{1}{nk} \sum_{c=1}^{n} \sum_{j=1}^{k} \{\phi(a_c)^\top \beta_{:j} - y_{c,j}^*\}^2 + \lambda \sum_{r=1}^{p} \sum_{j=1}^{k} \beta_{r,j}^2 \\
&= \frac{1}{nk} \left\| \begin{bmatrix} \phi(a_1)^\top \beta_{:1} - y_{1,1} & \cdots & \phi(a_1)^\top \beta_{:k} - y_{1,k} \\ \vdots & \ddots & \vdots \\ \phi(a_n)^\top \beta_{:1} - y_{n,1} & \cdots & \phi(a_n)^\top \beta_{:k} - y_{n,k} \end{bmatrix} \right\|_F^2 + \lambda \left\| \begin{bmatrix} \beta_{1,1} & \cdots & \beta_{1,k} \\ \vdots & \ddots & \vdots \\ \beta_{p,1} & \cdots & \beta_{p,k} \end{bmatrix} \right\|_F^2 \\
&= \frac{1}{nk} \left\| \phi(\mathbf{X}^*)\beta - \mathbf{Y}^* \right\|_F^2 + \lambda \left\| \beta \right\|_F^2 \ . \qquad \text{(B15)}
\end{aligned}$$

Differentiating Eq. B15 with respect to β we have:

$$\frac{\partial J(\beta)}{\partial \beta} = \frac{1}{nk}\frac{\partial}{\partial \beta}\left\|\phi(\mathbf{X}^*)\beta - \mathbf{Y}^*\right\|_F^2 + \lambda\frac{\partial}{\partial \beta}\left\|\beta\right\|_F^2 \ . \tag{B16}$$

For the left-hand side differentiation of Eq. B16 we have:

$$\begin{aligned}\frac{\partial}{\partial \beta}\left\|\phi(\mathbf{X}^*)\beta - \mathbf{Y}^*\right\|_F^2 &= \frac{\partial}{\partial \beta}\mathrm{Tr}\left\{(\phi(\mathbf{X}^*)\beta - \mathbf{Y}^*)^\top(\phi(\mathbf{X}^*)\beta - \mathbf{Y}^*)\right\} \\ &= \frac{\partial}{\partial \beta}\sum_{j=1}^{k}\underbrace{(\phi(\mathbf{X}^*)\beta - \mathbf{Y}^*)_{:j}^\top(\phi(\mathbf{X}^*)\beta - \mathbf{Y}^*)_{:j}}_{e_j^2} \ . \end{aligned} \tag{B17}$$

Where Tr denotes the trace of the matrix. Let e_j^2 be the error for the output prediction j. For that specific output, the differentiation is:

$$\begin{aligned}\frac{\partial}{\partial \beta}e_j^2 &= \begin{bmatrix} | & & | & & | \\ \frac{\partial}{\partial \beta_{:1}}e_j^2 & \cdots & \frac{\partial}{\partial \beta_{:j}}e_j^2 & \cdots & \frac{\partial}{\partial \beta_{:k}}e_j^2 \\ | & & | & & | \end{bmatrix} \\ &= \begin{bmatrix} | & & | & & | \\ \mathbf{0} & \cdots & \frac{\partial}{\partial \beta_{:j}}e_j^2 & \cdots & \mathbf{0} \\ | & & | & & | \end{bmatrix} \ . \end{aligned}$$

Isolating the j column from the above matrix, we have:

$$
\begin{aligned}
\frac{\partial}{\partial \beta_{:j}} e_j^2 &= \frac{\partial}{\partial \beta_{:j}} (\phi(\mathbf{X}^*)\beta - \mathbf{Y}^*)_{:j}^\top (\phi(\mathbf{X}^*)\beta - \mathbf{Y}^*)_{:j} \\
&= \frac{\partial}{\partial \beta_{:j}} (\phi(\mathbf{X}^*)\beta_{:j} - \mathbf{Y}_{:j}^*)^\top (\phi(\mathbf{X}^*)\beta_{:j} - \mathbf{Y}_{:j}^*) \\
&= \frac{\partial}{\partial \beta_{:j}} (\beta_{:j}^\top \phi(\mathbf{X}^*)^\top \phi(\mathbf{X}^*)\beta_{:j} - 2\beta_{:j}^\top \phi(\mathbf{X}^*)^\top \mathbf{Y}_{:j}^* + \mathbf{Y}_{:j}^{*\top} \mathbf{Y}_{:j}^*) \\
&= 2\phi(\mathbf{X}^*)^\top \phi(\mathbf{X}^*)\beta_{:j} - 2\phi(\mathbf{X}^*)^\top \mathbf{Y}_{:j}^* \ .
\end{aligned}
$$

Using the above result for all output errors in the summation of Eq. B17, we will have the following:

$$
\begin{aligned}
\frac{\partial}{\partial \beta} \left\| \phi(\mathbf{X}^*)\beta - \mathbf{Y}^* \right\|_F^2 &= \begin{bmatrix} | & & | \\ 2\phi(\mathbf{X}^*)^\top \phi(\mathbf{X}^*)\beta_{:1} - 2\phi(\mathbf{X}^*)^\top \mathbf{Y}_{:1}^* & \cdots & 2\phi(\mathbf{X}^*)^\top \phi(\mathbf{X}^*)\beta_{:k} - 2\phi(\mathbf{X}^*)^\top \mathbf{Y}_{:k}^* \\ | & & | \end{bmatrix} \\
&= 2\phi(\mathbf{X}^*)^\top \phi(\mathbf{X}^*) \begin{bmatrix} | & & | \\ \beta_{:1} & \cdots & \beta_{:k} \\ | & & | \end{bmatrix} - 2\phi(\mathbf{X}^*)^\top \begin{bmatrix} | & & | \\ \mathbf{Y}_{:1}^* & \cdots & \mathbf{Y}_{:k}^* \\ | & & | \end{bmatrix} \\
&= 2\phi(\mathbf{X}^*)^\top \phi(\mathbf{X}^*)\beta - 2\phi(\mathbf{X}^*)^\top \mathbf{Y}^* \ . \qquad \text{(B18)}
\end{aligned}
$$

Using the property of trace differentiation (Petersen et al., 2008, pp. 13), for the right-hand side of the differentiation in Eq. B16 we have:

$$
\begin{aligned}
\frac{\partial}{\partial \beta} \left\| \beta \right\|_F^2 &= \frac{\partial}{\partial \beta} \mathrm{Tr} \left\{ \beta^\top \beta \right\} \\
&= 2\beta \ . \qquad \text{(B19)}
\end{aligned}
$$

Plugging Eq. B18 and Eq. B19 into Eq. B16 and setting it to zero, we have:

$$\begin{aligned}
&\frac{\partial J(\beta)}{\partial \beta} = \frac{1}{nk}\left(2\phi(\mathbf{X}^*)^\top \phi(\mathbf{X}^*)\beta - 2\phi(\mathbf{X}^*)^\top \mathbf{Y}^*\right) + 2\lambda\beta = 0 \\
&\implies \beta = \phi(\mathbf{X}^*)^\top \underbrace{\frac{1}{\lambda nk}(\mathbf{Y}^* - \phi(\mathbf{X}^*)\beta)}_{\alpha \in \mathbb{R}^{n\times k}} \\
&\qquad = \phi(\mathbf{X}^*)^\top \alpha \ .
\end{aligned}$$

Where $\alpha \in \mathbb{R}^{n\times k}$ is the matrix containing dual variables. With above definition of β we can create the following dual problem:

$$\begin{aligned}
G(\alpha) &= \frac{1}{nk}\left\|\phi(\mathbf{X}^*)\phi(\mathbf{X}^*)^\top \alpha - \mathbf{Y}^*\right\|_F^2 + \lambda \left\|\phi(\mathbf{X}^*)^\top \alpha\right\|_F^2 \\
&= \frac{1}{nk}\left\|\mathbf{K}\alpha - \mathbf{Y}^*\right\|_F^2 + \lambda \left\|\phi(\mathbf{X}^*)^\top \alpha\right\|_F^2 \ .
\end{aligned}$$

To obtain the optimal solution of this dual form, we need to differentiate with respect to α and set it to zero. Differentiating the above equation with respect to α we have:

$$\frac{\partial G(\alpha)}{\partial \alpha} = \frac{1}{nk}\frac{\partial}{\partial \alpha}\left\|\mathbf{K}\alpha - \mathbf{Y}^*\right\|_F^2 + \lambda \frac{\partial}{\partial \alpha}\left\|\phi(\mathbf{X}^*)^\top \alpha\right\|_F^2 \ . \tag{B20}$$

The differentiation of the left-hand side part of Eq. B20 should follow similar steps as we did for Eq. B18, but instead of using $\phi(\mathbf{X}^*)$ and β, we replace them with $\mathbf{K}$ and α, correspondingly:

$$\frac{\partial}{\partial \alpha}\left\|\mathbf{K}\alpha - \mathbf{Y}^*\right\|_F^2 = 2\mathbf{K}^\top \mathbf{K}\alpha - 2\mathbf{K}^\top \mathbf{Y}^* \ . \tag{B21}$$

For the right-hand side part of Eq. B20 we have:

$$\begin{aligned}\frac{\partial}{\partial\alpha}\left\|\phi(\mathbf{X}^*)^\top\alpha\right\|_F^2 &= \frac{\partial}{\partial\alpha}\mathrm{Tr}\left\{(\phi(\mathbf{X}^*)^\top\alpha)^\top(\phi(\mathbf{X}^*)^\top\alpha)\right\}\\ &= \frac{\partial}{\partial\alpha}\mathrm{Tr}\left\{\alpha^\top\phi(\mathbf{X}^*)\phi(\mathbf{X}^*)^\top\alpha\right\}\\ &= \frac{\partial}{\partial\alpha}\mathrm{Tr}\left\{\alpha^\top\mathbf{K}\alpha\right\}\\ &= 2\mathbf{K}\alpha\ . \end{aligned} \tag{B22}$$

Plugging Eq. B21 and Eq. B22 into Eq. B20 and setting it to zero we have:

$$\begin{aligned}&\frac{\partial G(\alpha)}{\partial\alpha} = \frac{1}{nk}\left(2\mathbf{K}^\top\mathbf{K}\alpha - 2\mathbf{K}^\top\mathbf{Y}^*\right) + 2\lambda\mathbf{K}\alpha = 0\\ &\implies \alpha = (\mathbf{K}^\top\mathbf{K} + \lambda nk\mathbf{K})^{-1}\mathbf{K}^\top\mathbf{Y}^*\ .\end{aligned}$$

If $\mathbf{K}$ is a positive semidefinite matrix, then we can also rewrite α as follows:

$$\alpha = (\mathbf{K} + \lambda nk\mathbf{I})^{-1}\mathbf{Y}^*\ .$$

B.3 Supplementary Figures

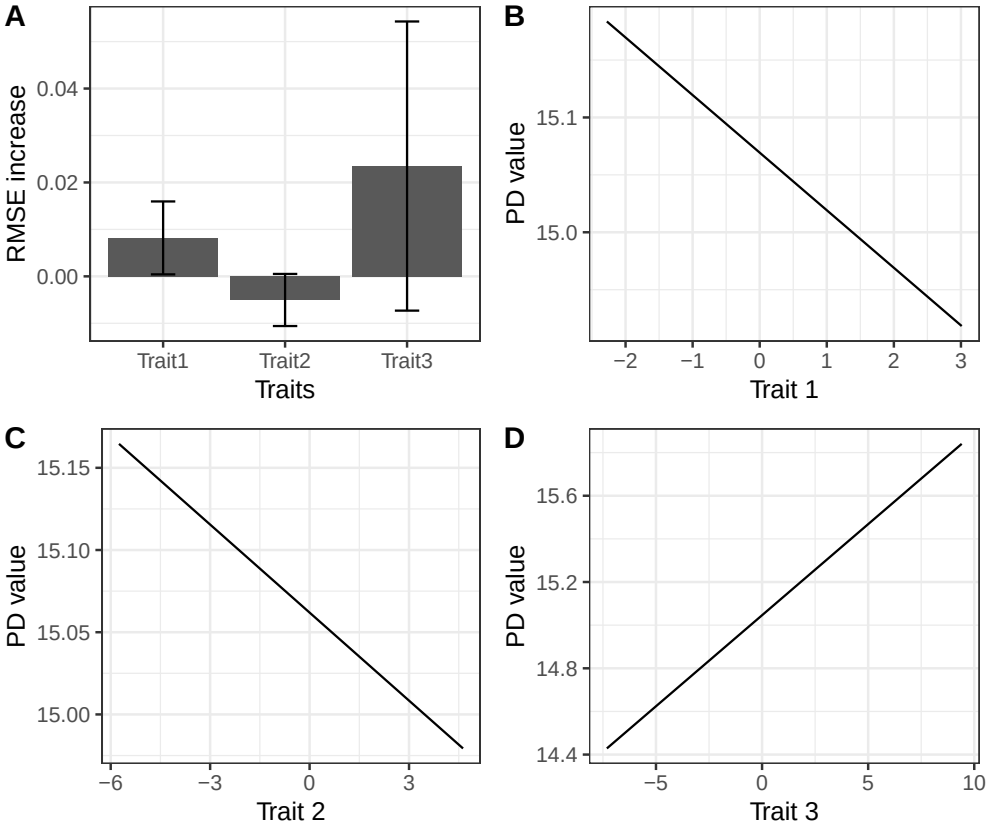


Figure B1: Model inspection analyses using of phyloKRR with the linear kernel model fitting simulated data generated from Eq. 6 in the main text. In **A**), we present permutation FI analysis. Notice that the rank for FI does not follow the weights given in Eq. 6 in the main text. In **B**), we show the marginal contribution of the first trait in the overall model. In **C**) and **D**), we show the same for traits 2 and 3, respectively. Notice that the curves in **B**), **C**), and **D**) follow a linear response instead of a quadratic response expected from Eq. 6 in the main text. PD value stands for Partial Dependence value.

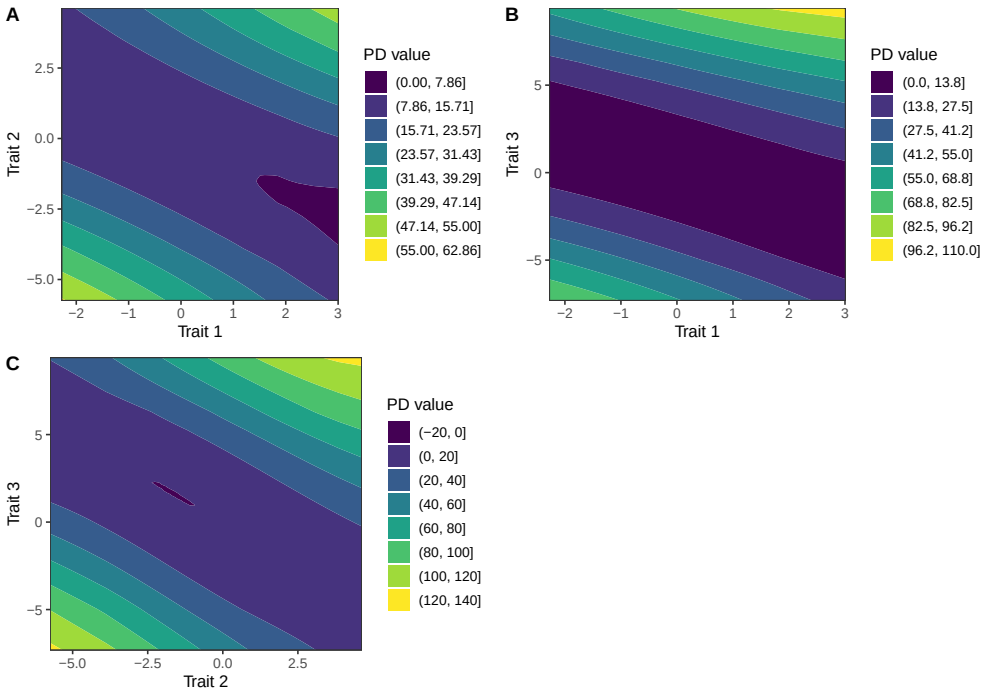


Figure B2: Marginal contribution of the interaction of traits in phyloKRR with the RBF kernel model as measured by 2D PDP. In **A**), we present the interaction between traits 1 and 2. In **B**), we present the interaction between traits 1 and 3. In **C**), we present the interaction between traits 2 and 3. Notice that each panel approaches the shape of a bowl, with the lowest values in the center and the highest values outwardly. PD value stands for Partial Dependence value.

Supplemental references

Borchani, H., G. Varando, C. Bielza, and P. Larranaga, 2015: A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **5 (5)**, 216–233.

Boyd, S. P., and L. Vandenberghe, 2004: *Convex optimization*. Cambridge university press.

Garland, T., Jr, and A. R. Ives, 2000: Using the past to predict the present: confidence intervals for regression equations in phylogenetic comparative methods. *The American Naturalist*, **155 (3)**, 346–364.

Grafen, A., 1989: The phylogenetic regression. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, **326 (1233)**, 119–157.

Hastie, T., R. Tibshirani, and J. Friedman, 2009: Springer Series in Statistics. *The Elements of Statistical Learning*, **27 (2)**, 83–85, https://doi.org/10.1007/b94608, URL http://www.springerlink.com/index/D7X7KX6772HQ2135.pdf, arXiv:1011.1669v3.

Petersen, K. B., M. S. Pedersen, and Coauthors, 2008: The matrix cookbook. *Technical University of Denmark*, **7 (15)**, 510.

Tung Ho, L. s., and C. Ané, 2014: A linear-time algorithm for gaussian and non-gaussian trait evolution models. *Systematic biology*, **63 (3)**, 397–408.

Wang, H., 2022: A note on centering in subsample selection for linear regression. *Stat*, **11 (1)**, e525.

Appendix C

Supplementary Material: Dissecting factors underlying phylogenetic uncertainty using machine learning models

C.1 Quality Control

We used a quality control (QC) pipeline to address potential sources of contamination, mislabeling, sequencing errors, or other confounding features of the data. All steps involved in this pipeline were integrated into a Python-based command-line interface called `"fishlifeqc"` (available at: github.com/ulises-rosas/fishlifeqc). We based this pipeline on the conceptual framework proposed by Simion et al. (2017); Arcila et al. (2021) and it currently includes three main functions:

1. Missing data: Missing data has well known impact on the phylogeny estimation (Xi et al., 2016; Smirnov and Warnow, 2021; Portik and Wiens, 2021), and we included a step for controlling the amount of missing data in this pipeline. Since adoption of a

universal threshold for eliminating poorly aligned regions can be problematic (Molloy and Warnow, 2018), our implementation eliminates complete exon alignments with less than 75% taxa. After this step, we excluded 79 and 114 exon alignments for the Protacanthopterygii and Carangaria datasets, respectively. A complete list of options for this step can be obtained with the command `"fishlifeqc mdata -h"`.

2. Mislabeling: Errors involving mislabeling of samples (sequences) can induce phylogenetic errors or misleading conclusions on, for instance, the timing of coalescent events (Springer and Gatesy, 2016). This step is designed to verify correct species identification in a dataset by using mitochondrial COI barcode data by taking leveraging the BOLD ID Engine webservice. After this step, we update 2 and 15 species names for the Protacanthopterygii and Carangaria datasets, respectively. In the case of the Carangaria dataset, we eliminated 20 samples as they belong to non-Carangarian species. A complete list of options for this step can be obtained with the command `"fishlifeqc bold -h"`.

3. Taxon misplacement due to orthology error or contamination: This step compares the terminal branch lengths between a reference tree and the gene tree for each exon alignment, and it was first proposed by Simion et al. (2017) as their "branch length correlation" or BLC test. We removed sequences for which the branch-length ratio > 5 and we used the Pearson correlation coefficient (R^2) between branch lengths in the same two trees to identify (and eliminate) gene alignments that are outliers. After this step, we excluded 30 exon alignments for the Protacanthopterygii dataset. A complete list of options for this step can be obtained with the command `"fishlifeqc bl -h"`.

Table C1: List of species examined of the Protacanthopterygii dataset

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
Alepocephaliformes	Platytroctidae	Holtbyrnia	*Holtbyrnia laticauda*	This study (EPLATE_28_H06)	Genome
		Normichthys	*Normichthys yahganorum*	This study (EPLATE_28_H07)	Genome
		Persparsia	*Persparsia kopua*	This study (EPLATE_46_D03)	Genome
Argentiniformes	Argentinidae	Argentina	*Argentina australiae*	This study (EPLATE_46_F03)	Genome

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Argentina silus	SRR11679483	Genome
			Argentina sp	SRR5997680	Transcriptome
			Argentina sp	SRX3153196	Transcriptome
			Argentina sphyraena	SRR11537143	Genome
		Glossanodon			
			Glossanodon spW1	This study (EPLATE_46_E03)	Genome
	Bathylagidae				
		Bathylagichthys			
			Bathylagichthys australis	This study (EPLATE_46_H03)	Genome
			Bathylagichthys longipinnis	This study (EPLATE_46_G03)	Genome
			Bathylagichthys problematicus	This study (EPLATE_46_A04)	Genome
		Melanolagus			

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Melanolagus bericoides	This study (EPLATE_46_C04)	Genome
			Melanolagus sp2	This study (EPLATE_46_B04)	Genome
	Microstomatidae	Microstoma	*Microstoma microstoma*	This study (EPLATE_46_D04)	Genome
		Nansenia	*Nansenia antarctica*	This study (EPLATE_46_E04)	Genome
	Opisthoproctidae	Monacoa	Monacoa sp1	This study (EPLATE_46_G04)	Genome
		Opisthoproctus			

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Opisthoproctus soleatus	ERR3332507	Genome
		Winteria			
			Winteria telescopa	This study (EPLATE_46_H04)	Genome
Ateleopodiformes	Ateleopodidae	Ateleopus	*Ateleopus japonicus*	This study (EPLATE_26_E02)	Genome
		Ijimaia	*Ijimaia dofleini*	This study (EPLATE_26_E03)	Genome
			Ijimaia dofleini	This study (EPLATE_46_A05)	Genome

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Ijimaia sp1	This study (EPLATE_46_C05)	Genome
Aulopiformes	Aulopidae	Hime	*Hime caudizoma*	This study (EPLATE_60_B05)	Genome
			Hime curtirostris	This study (EPLATE_61_B03)	Genome
	Bathysauroididae	Bathysauroides	*Bathysauroides gigas*	This study (EPLATE_46_E05)	Genome
	Paralepididae	Lestidium			

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Lestidium atlanticum	This study (EPLATE_61_C05)	Genome
Clupeiformes	Clupeidae	Alosa	*Alosa alosa*	PRJNA256955	Transcriptome
		Amblygaster	*Amblygaster clupeoides*	SRX3153191	Transcriptome
		Clupea	*Clupea harengus*	GCA_000966335.1	Genome
	Engraulidae	Coilia	*Coilia nasus*	SRX3153210	Transcriptome
		Engraulis			

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Engraulis encrasicolus	PRJNA261165	Transcriptome
Esociformes	Esocidae	Esox	*Esox lucius*	GCA_000721915.2	Genome
			Esox reichertii	This study (EPLATE_44_C07)	Genome
	Umbridae	Umbra	*Umbra pygmaea*	PRJNA257000	Transcriptome
Galaxiiformes	Galaxiidae	Galaxias	*Galaxias maculatus*	Hughes et al. (2018)	Genome

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Galaxias truttaceus	This study (EPLATE_25_B09)	Genome
		Galaxiella	*Galaxiella nigrostriata*	SRX3153920	Transcriptome
		Lovettia	*Lovettia sealii*	This study (EPLATE_25_B10)	Genome
Lepidogalaxiiformes	Lepidogalaxiidae	Lepidogalaxias	*Lepidogalaxias salamandroides*	SRX3153919	Transcriptome
Myctophiformes	Myctophidae	Benthosema			

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Benthosema fibulatum	This study (EPLATE_49_A06)	Genome
		Diaphus			
			Diaphus hudsoni	This study (EPLATE_49_G02)	Genome
			Diaphus termophilus	This study (EPLATE_49_E01)	Genome
			Diaphus whitleyi	This study (EPLATE_49_G09)	Genome
Osmeriformes	Osmeridae				
		Mallotus			
			Mallotus villosus	GCA_903064625.1	Genome
		Osmerus			
			Osmerus eperlanus	GCA_900302275.1	Genome

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Osmerus mordax	This study (EPLATE_44_E11)	Genome
		Spirinchus	*Spirinchus thaleichthys*	SRR2758850	Transcriptome
		Thaleichthys	*Thaleichthys pacificus*	SRR1786619	Genome
	Plecoglossidae	Plecoglossus	*Plecoglossus altivelis*	This study (EPLATE_44_G03)	Genome
	Salangidae	Protosalanx	*Protosalanx hyalocranius*	PRJNA328051	Genome

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
		Salanx	*Salanx chinensis*	GCA_010882115.1	Genome
			Salanx sp1	This study (EPLATE_64_D11)	Genome
Salmoniformes	Salmonidae	Brachymystax	*Brachymystax lenok*	SRR8873459	Transcriptome
		Coregonus	Coregonus 22		
			Coregonus artedi	SRR9718093	Genome
			Coregonus clupeaformis	PRJNA256999	Transcriptome
			Coregonus lavaretus	SRR9951409	Genome
			Coregonus palaea	SRR2175595	Transcriptome
			Coregonus sp	GCA_902810595.1	Genome

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
		Hucho			
			Hucho hucho	GCA_003317085.1	Genome
			Hucho taimen	ERR1857179	Transcriptome
		Oncorhynchus			
			Oncorhynchus keta	GCF_012931545.1	Genome
			Oncorhynchus kisutch	GCF_002021735.2	Genome
			Oncorhynchus masou	DRR239371	Transcriptome
			Oncorhynchus mykiss	PRJEB4421	Genome
			Oncorhynchus nerka	GCF_006149115.1	Genome
			Oncorhynchus tshawytscha	This study (EPLATE_62_C09)	Genome
		Salmo			
			Salmo ischchan	SRR6854514	Genome
			Salmo marmoratus	SRR7687456	Transcriptome

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Salmo salar	GCA_000233375.4	Genome
			Salmo trutta	GCF_901001165.1	Genome
		Salvelinus			
			Salvelinus alpinus	SRR8477003	Genome
			Salvelinus fontinalis	PRJNA256998	Transcriptome
			Salvelinus namaycush	SRR11144341	Genome
			Salvelinus sp	GCF_002910315.2	Genome
		Thymallus			
			Thymallus thymallus	PRJNA256970	Transcriptome
Stomiatiformes	Gonostomatidae	Diplophos	*Diplophos australis*	This study (EPLATE_53_G09)	Genome

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Diplophos rebainsi	This study (EPLATE_53_F09)	Genome
			Diplophos taenia	This study (EPLATE_18_F09)	Genome
		Margrethia	*Margrethia valentinae*	This study (EPLATE_53_D10)	Genome
		Sigmops	*Sigmops bathyphilus*	This study (EPLATE_53_C10)	Genome
			Sigmops elongatus	This study (EPLATE_53_H09)	Genome
			Sigmops elongatus	This study (EPLATE_53_A10)	Genome
	Phosichthyidae	Ichthyococcus			

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Ichthyococcus australis	This study (EPLATE_53_E10)	Genome
		Phosichthys	*Phosichthys argenteus*	This study (EPLATE_53_A11)	Genome
		Polymetme	*Polymetme corythaeola*	This study (EPLATE_53_G10)	Genome
			Polymetme illustris	This study (EPLATE_53_H10)	Genome
		Vinciguerria	*Vinciguerria attenuata*	This study (EPLATE_53_B11)	Genome
			Vinciguerria attenuata	This study (EPLATE_53_D11)	Genome

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Vinciguerria nimbaria	This study (EPLATE_53_C11)	Genome
		Woodsia			
			Woodsia meyerwaardeni	This study (EPLATE_53_E11)	Genome
	Sternoptychidae				
		Argyripnus			
			Argyripnus atlanticus	This study (EPLATE_18_H06)	Genome
			Argyripnus iridescens	This study (EPLATE_53_F08)	Genome
		Argyropelecus			
			Argyropelecus aculeatus	This study (EPLATE_33_F02)	Genome
			Argyropelecus affinis	This study (EPLATE_18_F08)	Genome

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Argyropelecus gigas	This study (EPLATE_53_B08)	Genome
			Argyropelecus hemigymnus	This study (EPLATE_53_D08)	Genome
			Argyropelecus olfersii	This study (EPLATE_18_H09)	Genome
			Argyropelecus sladeni	This study (EPLATE_53_E08)	Genome
		Maurolicus			
			Maurolicus japonicus	This study (EPLATE_56_F05)	Genome
			Maurolicus mucronatus	SRR6008916	Transcriptome
			Maurolicus muelleri	This study (EPLATE_24_C05)	Genome
		Polyipnus			

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Polyipnus ruggeri	This study (EPLATE_53_B09)	Genome
		Sternoptyx			
			Sternoptyx diaphana	This study (EPLATE_18_E07)	Genome
			Sternoptyx pseudodiaphana	This study (EPLATE_18_G07)	Genome
	Stomiidae				
		Astronesthes			
			Astronesthes boulengeri	This study (EPLATE_53_F11)	Genome
			Astronesthes illuminatus	This study (EPLATE_53_G11)	Genome
		Bathophilus			
			Bathophilus abarbatus	This study (EPLATE_53_G12)	Genome

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
		Borostomias	*Borostomias antarcticus*	GCA_900323325.1	Genome
		Chauliodus	*Chauliodus sloani*	This study (EPLATE_18_A09)	Genome
		Malacosteus	*Malacosteus australis*	This study (EPLATE_53_C12)	Genome
			Malacosteus niger	This study (EPLATE_18_A10)	Genome
		Neonesthes	*Neonesthes microcephalus*	This study (EPLATE_53_B12)	Genome
		Pachystomias			

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
			Pachystomias microdon	SRR5069729	Transcriptome
		Photostomias	*Photostomias tantillux*	This study (EPLATE_18_D09)	Genome
		Rhadinesthes	*Rhadinesthes decimus*	This study (EPLATE_53_A12)	Genome
		Stomias	*Stomias affinis*	This study (EPLATE_18_G08)	Genome
			Stomias boa	This study (EPLATE_24_B12)	Genome
			Stomias gracilis	This study (EPLATE_53_F12)	Genome
Zeiformes					

Continued on next page

Table C1 – *Continued from previous page*

Order	**Family**	**Genera**	**Species**	**Accession**	**Data type**
	Oreosomatidae	Allocyttus	*Allocyttus verrucosus*	This study (EPLATE_33_E07)	Genome
		Oreosoma	*Oreosoma atlanticum*	This study (EPLATE_53_A08)	Genome
		Pseudocyttus	*Pseudocyttus maculatus*	This study (EPLATE_53_G07)	Genome

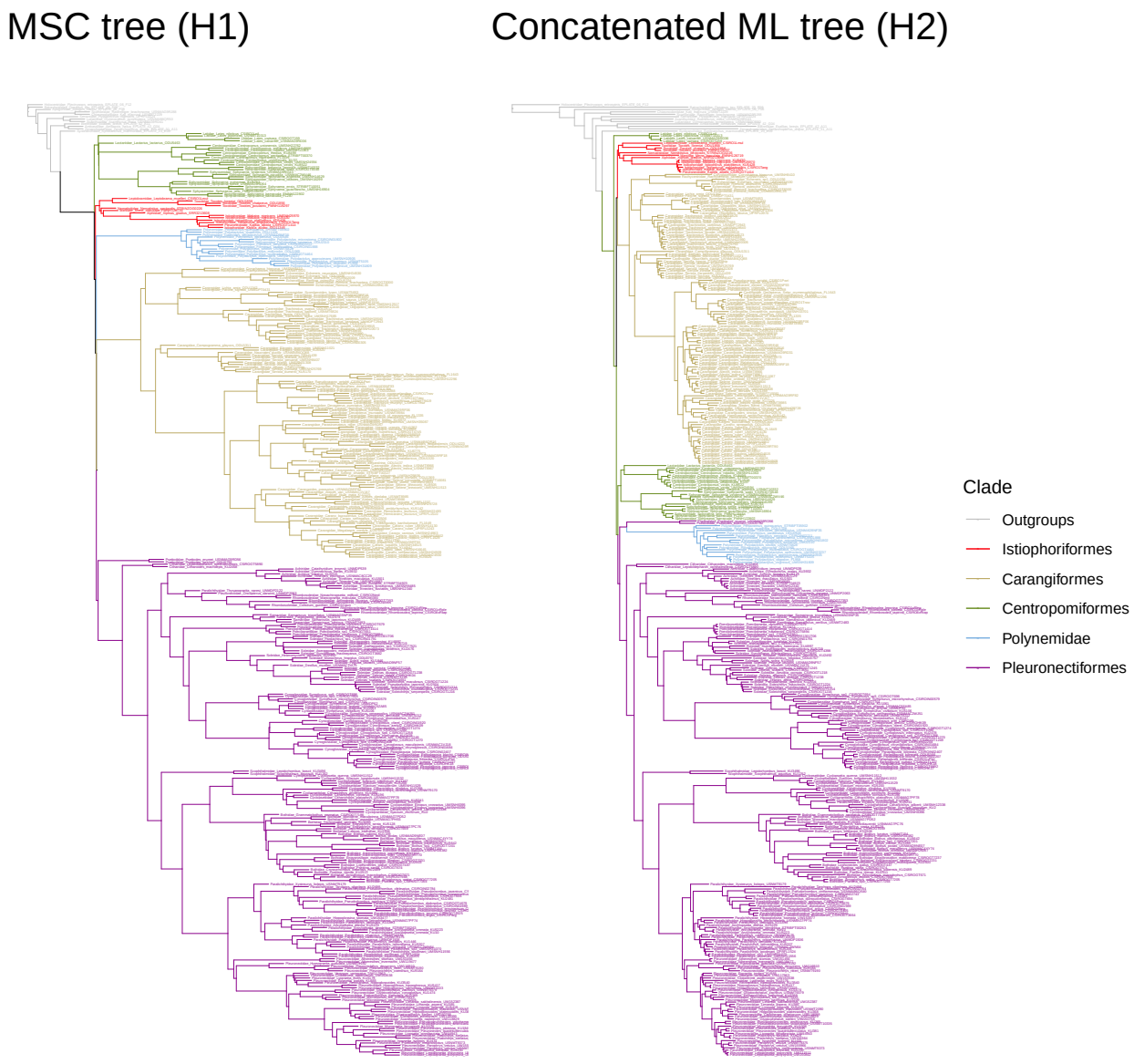


Figure C1: Complete phylogenetic hypotheses for the Carangaria dataset. The left panel shows the MSC tree obtained with ASTRAL (H1) and the right panel shows the concatenated ML tree obtained with IQ-TREE (H2)

Table C2: List of hypotheses tested with GGI for the Protacanthopterygii dataset. Outgroup is composed by the orders Lepidogalaxiiformes, Alepocephaliformes, and Clupeiformes

Label	**Hypothesis**	**Authors**
H1	(Outgroup,(Eso_salmo,(Argentiniformes,(Osme_Stomia,(Neoteleostei,Galaxiiformes)))));	Miya and Nishida (2015)
H2	(Outgroup,((Eso_salmo,Argentiniformes),(Osme_Stomia,(Neoteleostei,Galaxiiformes))));	Near et al. (2012); Ybazeta (2012); Hughes et al. (2018)
H3	(Outgroup,(Argentiniformes,(Eso_salmo,(Osme_Stomia,(Neoteleostei,Galaxiiformes)))));	
H4	(Outgroup,((Neoteleostei,Galaxiiformes),(Osme_Stomia,(Eso_salmo,Argentiniformes))));	
H5	(Outgroup,(Galaxiiformes,(Neoteleostei,(Osme_Stomia,(Eso_salmo,Argentiniformes)))));	
H6	(Outgroup,((Eso_salmo,Argentiniformes),(Neoteleostei,(Osme_Stomia,Galaxiiformes))));	López et al. (2004)
H7	(Outgroup,(Osme_Stomia,((Eso_salmo,Argentiniformes),(Neoteleostei,Galaxiiformes))));	
H8	(Outgroup,(Eso_salmo,(Osme_Stomia,(Argentiniformes,(Neoteleostei,Galaxiiformes)))));	
H9	(Outgroup,(Eso_salmo,((Neoteleostei,Galaxiiformes),(Osme_Stomia,Argentiniformes))));	
H10	(Outgroup,(Eso_salmo,(Argentiniformes,(Neoteleostei,(Osme_Stomia,Galaxiiformes)))));	
H11	(Outgroup,(Argentiniformes,(Osme_Stomia,(Eso_salmo,(Neoteleostei,Galaxiiformes)))));	
H12	(Outgroup,((Neoteleostei,Galaxiiformes),(Argentiniformes,(Osme_Stomia,Eso_salmo))));	

Continued on next page

Table C2 – *Continued from previous page*

Label	**Hypothesis**	**Authors**
H13	(Outgroup,(Galaxiiformes,(Osme_Stomia,((Eso_salmo,Argentiniformes),Neoteleostei))));	
H14	(Outgroup,(Neoteleostei,(Galaxiiformes,(Osme_Stomia,(Eso_salmo,Argentiniformes)))));	Li et al. (2010)
H15	(Outgroup,(Osme_Stomia,((Eso_salmo,(Neoteleostei,Galaxiiformes)),Argentiniformes)));	
H16	(Outgroup,(Eso_salmo,((Argentiniformes,Neoteleostei),(Osme_Stomia,Galaxiiformes))));	
H17	(Outgroup,(Eso_salmo,(Argentiniformes,(Galaxiiformes,(Osme_Stomia,Neoteleostei)))));	
H18	(Outgroup,(Argentiniformes,((Neoteleostei,Galaxiiformes),(Osme_Stomia,Eso_salmo))));	
H19	(Outgroup,(Galaxiiformes,(Neoteleostei,(Argentiniformes,(Osme_Stomia,Eso_salmo)))));	
H20	(Outgroup,(Osme_Stomia,(Eso_salmo,(Argentiniformes,(Neoteleostei,Galaxiiformes)))));	
H21	(Outgroup,(Eso_salmo,(Osme_Stomia,((Argentiniformes,Galaxiiformes),Neoteleostei))));	
H22	(Outgroup,(((Eso_salmo,Argentiniformes),Neoteleostei),(Osme_Stomia,Galaxiiformes)));	
H23	(Outgroup,((Neoteleostei,Galaxiiformes),(Eso_salmo,(Osme_Stomia,Argentiniformes))));	
H24	(Outgroup,(Galaxiiformes,((Eso_salmo,Argentiniformes),(Osme_Stomia,Neoteleostei))));	Mirande (2017)
H25	(Outgroup,(Neoteleostei,(Osme_Stomia,((Eso_salmo,Galaxiiformes),Argentiniformes))));	
H26	(Outgroup,(Neoteleostei,(Osme_Stomia,((Eso_salmo,Argentiniformes),Galaxiiformes))));	
H27	(Outgroup,(Osme_Stomia,(((Eso_salmo,Argentiniformes),Galaxiiformes),Neoteleostei)));	
H28	(Outgroup,(Argentiniformes,((Eso_salmo,Neoteleostei),(Osme_Stomia,Galaxiiformes))));	

Continued on next page

Table C2 – *Continued from previous page*

Label	**Hypothesis**	**Authors**
H29	(Outgroup,(Eso_salmo,(Neoteleostei,(Argentiniformes,(Osme_Stomia,Galaxiiformes)))));	Nelson et al. (2016); Hughes et al. (2018)
H30	(Outgroup,(Eso_salmo,(Osme_Stomia,((Argentiniformes,Neoteleostei),Galaxiiformes))));	
H31	(Outgroup,(Neoteleostei,((Eso_salmo,Argentiniformes),(Osme_Stomia,Galaxiiformes))));	Campbell et al. (2013)
H32	(Outgroup,(Neoteleostei,(Eso_salmo,(Osme_Stomia,(Argentiniformes,Galaxiiformes)))));	
H33	(Outgroup,((Argentiniformes,(Neoteleostei,Galaxiiformes)),(Osme_Stomia,Eso_salmo)));	
H34	(Outgroup,(Argentiniformes,((Eso_salmo,Galaxiiformes),(Osme_Stomia,Neoteleostei))));	
H35	(Outgroup,(Eso_salmo,(Neoteleostei,(Osme_Stomia,(Argentiniformes,Galaxiiformes)))));	
H36	(Outgroup,(Neoteleostei,(Eso_salmo,(Argentiniformes,(Osme_Stomia,Galaxiiformes)))));	
H37	(Outgroup,(((Eso_salmo,Argentiniformes),Galaxiiformes),(Osme_Stomia,Neoteleostei)));	
H38	(Outgroup,(((Eso_salmo,Galaxiiformes),Neoteleostei),(Osme_Stomia,Argentiniformes)));	
H39	(Outgroup,((Eso_salmo,(Argentiniformes,Galaxiiformes)),(Osme_Stomia,Neoteleostei)));	
H40	(Outgroup,((Eso_salmo,Argentiniformes),(Galaxiiformes,(Osme_Stomia,Neoteleostei))));	
H41	(Outgroup,(Argentiniformes,(Eso_salmo,(Neoteleostei,(Osme_Stomia,Galaxiiformes)))));	
H42	(Outgroup,(Eso_salmo,(Neoteleostei,(Galaxiiformes,(Osme_Stomia,Argentiniformes)))));	
H43	(Outgroup,(Galaxiiformes,(Eso_salmo,(Argentiniformes,(Osme_Stomia,Neoteleostei)))));	

Continued on next page

Table C2 – *Continued from previous page*

Label	**Hypothesis**	**Authors**
H44	(Outgroup,(Galaxiiformes,(Neoteleostei,(Eso_salmo,(Osme_Stomia,Argentiniformes)))));	
H45	(Outgroup,(Galaxiiformes,(Osme_Stomia,(Eso_salmo,(Argentiniformes,Neoteleostei)))));	
H46	(Outgroup,(Neoteleostei,((Argentiniformes,Galaxiiformes),(Osme_Stomia,Eso_salmo))));	
H47	(Outgroup,(Neoteleostei,((Eso_salmo,Galaxiiformes),(Osme_Stomia,Argentiniformes))));	
H48	(Outgroup,((Argentiniformes,Neoteleostei),(Eso_salmo,(Osme_Stomia,Galaxiiformes))));	
H49	(Outgroup,((Eso_salmo,Galaxiiformes),(Argentiniformes,(Osme_Stomia,Neoteleostei))));	
H50	(Outgroup,(Argentiniformes,(Osme_Stomia,((Eso_salmo,Neoteleostei),Galaxiiformes))));	
H51	(Outgroup,(Galaxiiformes,(Argentiniformes,(Osme_Stomia,(Eso_salmo,Neoteleostei)))));	
H52	(Outgroup,(Neoteleostei,(Galaxiiformes,(Argentiniformes,(Osme_Stomia,Eso_salmo)))));	
H53	(Outgroup,(Osme_Stomia,(((Eso_salmo,Galaxiiformes),Argentiniformes),Neoteleostei)));	
H54	(Outgroup,(Osme_Stomia,(((Eso_salmo,Neoteleostei),Galaxiiformes),Argentiniformes)));	
H55	(Outgroup,(((Eso_salmo,Galaxiiformes),Argentiniformes),(Osme_Stomia,Neoteleostei)));	Betancur et al. (2017)
H56	(Outgroup,(((Eso_salmo,Neoteleostei),Argentiniformes),(Osme_Stomia,Galaxiiformes)));	
H57	(Outgroup,((Argentiniformes,Galaxiiformes),(Osme_Stomia,(Eso_salmo,Neoteleostei))));	
H58	(Outgroup,((Eso_salmo,Galaxiiformes),(Neoteleostei,(Osme_Stomia,Argentiniformes))));	
H59	(Outgroup,(Argentiniformes,(Neoteleostei,(Eso_salmo,(Osme_Stomia,Galaxiiformes)))));	

Continued on next page

Table C2 – *Continued from previous page*

Label	**Hypothesis**	**Authors**
H60	(Outgroup,(Argentiniformes,(Neoteleostei,(Osme_Stomia,(Eso_salmo,Galaxiiformes)))));	
H61	(Outgroup,(Eso_salmo,((Argentiniformes,Galaxiiformes),(Osme_Stomia,Neoteleostei))));	
H62	(Outgroup,(Eso_salmo,(Galaxiiformes,(Osme_Stomia,(Argentiniformes,Neoteleostei)))));	
H63	(Outgroup,(Galaxiiformes,(Argentiniformes,(Eso_salmo,(Osme_Stomia,Neoteleostei)))));	
H64	(Outgroup,(Galaxiiformes,(Eso_salmo,(Osme_Stomia,(Argentiniformes,Neoteleostei)))));	
H65	(Outgroup,(Osme_Stomia,(((Eso_salmo,Argentiniformes),Neoteleostei),Galaxiiformes)));	
H66	(Outgroup,(Osme_Stomia,(((Eso_salmo,Neoteleostei),Argentiniformes),Galaxiiformes)));	
H67	(Outgroup,(((Argentiniformes,Galaxiiformes),Neoteleostei),(Osme_Stomia,Eso_salmo)));	
H68	(Outgroup,((Argentiniformes,Galaxiiformes),(Eso_salmo,(Osme_Stomia,Neoteleostei))));	
H69	(Outgroup,((Argentiniformes,Neoteleostei),(Osme_Stomia,(Eso_salmo,Galaxiiformes))));	
H70	(Outgroup,((Eso_salmo,(Argentiniformes,Neoteleostei)),(Osme_Stomia,Galaxiiformes)));	
H71	(Outgroup,((Eso_salmo,Neoteleostei),(Argentiniformes,(Osme_Stomia,Galaxiiformes))));	
H72	(Outgroup,(Argentiniformes,(Eso_salmo,(Galaxiiformes,(Osme_Stomia,Neoteleostei)))));	
H73	(Outgroup,(Argentiniformes,(Neoteleostei,(Galaxiiformes,(Osme_Stomia,Eso_salmo)))));	
H74	(Outgroup,(Argentiniformes,(Osme_Stomia,((Eso_salmo,Galaxiiformes),Neoteleostei))));	
H75	(Outgroup,(Eso_salmo,(Galaxiiformes,(Argentiniformes,(Osme_Stomia,Neoteleostei)))));	

Continued on next page

Table C2 – *Continued from previous page*

Label	**Hypothesis**	**Authors**
H76	(Outgroup,(Eso_salmo,(Galaxiiformes,(Neoteleostei,(Osme_Stomia,Argentiniformes)))));	
H77	(Outgroup,(Galaxiiformes,((Argentiniformes,Neoteleostei),(Osme_Stomia,Eso_salmo))));	
H78	(Outgroup,(Galaxiiformes,((Eso_salmo,Neoteleostei),(Osme_Stomia,Argentiniformes))));	
H79	(Outgroup,(Galaxiiformes,(Argentiniformes,(Neoteleostei,(Osme_Stomia,Eso_salmo)))));	
H80	(Outgroup,(Neoteleostei,(Argentiniformes,(Galaxiiformes,(Osme_Stomia,Eso_salmo)))));	
H81	(Outgroup,(Neoteleostei,(Argentiniformes,(Osme_Stomia,(Eso_salmo,Galaxiiformes)))));	
H82	(Outgroup,((Eso_salmo,(Neoteleostei,Galaxiiformes)),(Osme_Stomia,Argentiniformes)));	
H83	(Outgroup,((Eso_salmo,Neoteleostei),(Osme_Stomia,(Argentiniformes,Galaxiiformes))));	
H84	(Outgroup,(Galaxiiformes,(Osme_Stomia,((Eso_salmo,Neoteleostei),Argentiniformes))));	
H85	(Outgroup,(Neoteleostei,(Eso_salmo,(Galaxiiformes,(Osme_Stomia,Argentiniformes)))));	
H86	(Outgroup,(Neoteleostei,(Osme_Stomia,(Eso_salmo,(Argentiniformes,Galaxiiformes)))));	
H87	(Outgroup,(Osme_Stomia,(((Eso_salmo,Galaxiiformes),Neoteleostei),Argentiniformes)));	
H88	(Outgroup,(Osme_Stomia,((Eso_salmo,(Argentiniformes,Galaxiiformes)),Neoteleostei)));	
H89	(Outgroup,(Osme_Stomia,((Eso_salmo,Galaxiiformes),(Argentiniformes,Neoteleostei))));	
H90	(Outgroup,(Osme_Stomia,((Eso_salmo,Neoteleostei),(Argentiniformes,Galaxiiformes))));	
H91	(Outgroup,(((Argentiniformes,Neoteleostei),Galaxiiformes),(Osme_Stomia,Eso_salmo)));	

Continued on next page

Table C2 – *Continued from previous page*

Label	**Hypothesis**	**Authors**
H92	(Outgroup,((Argentiniformes,Neoteleostei),(Galaxiiformes,(Osme_Stomia,Eso_salmo))));	
H93	(Outgroup,((Eso_salmo,Neoteleostei),(Galaxiiformes,(Osme_Stomia,Argentiniformes))));	
H94	(Outgroup,(Argentiniformes,(Galaxiiformes,(Neoteleostei,(Osme_Stomia,Eso_salmo)))));	
H95	(Outgroup,(Neoteleostei,(Galaxiiformes,(Eso_salmo,(Osme_Stomia,Argentiniformes)))));	
H96	(Outgroup,(Osme_Stomia,((Eso_salmo,(Argentiniformes,Neoteleostei)),Galaxiiformes)));	
H97	(Outgroup,(Osme_Stomia,(Eso_salmo,((Argentiniformes,Galaxiiformes),Neoteleostei))));	
H98	(Outgroup,(((Eso_salmo,Neoteleostei),Galaxiiformes),(Osme_Stomia,Argentiniformes)));	
H99	(Outgroup,((Argentiniformes,Galaxiiformes),(Neoteleostei,(Osme_Stomia,Eso_salmo))));	
H100	(Outgroup,((Eso_salmo,Galaxiiformes),(Osme_Stomia,(Argentiniformes,Neoteleostei))));	
H101	(Outgroup,(Argentiniformes,(Galaxiiformes,(Eso_salmo,(Osme_Stomia,Neoteleostei)))));	
H102	(Outgroup,(Argentiniformes,(Galaxiiformes,(Osme_Stomia,(Eso_salmo,Neoteleostei)))));	
H103	(Outgroup,(Galaxiiformes,(Eso_salmo,(Neoteleostei,(Osme_Stomia,Argentiniformes)))));	
H104	(Outgroup,(Neoteleostei,(Argentiniformes,(Eso_salmo,(Osme_Stomia,Galaxiiformes)))));	
H105	(Outgroup,(Osme_Stomia,(Eso_salmo,((Argentiniformes,Neoteleostei),Galaxiiformes))));	

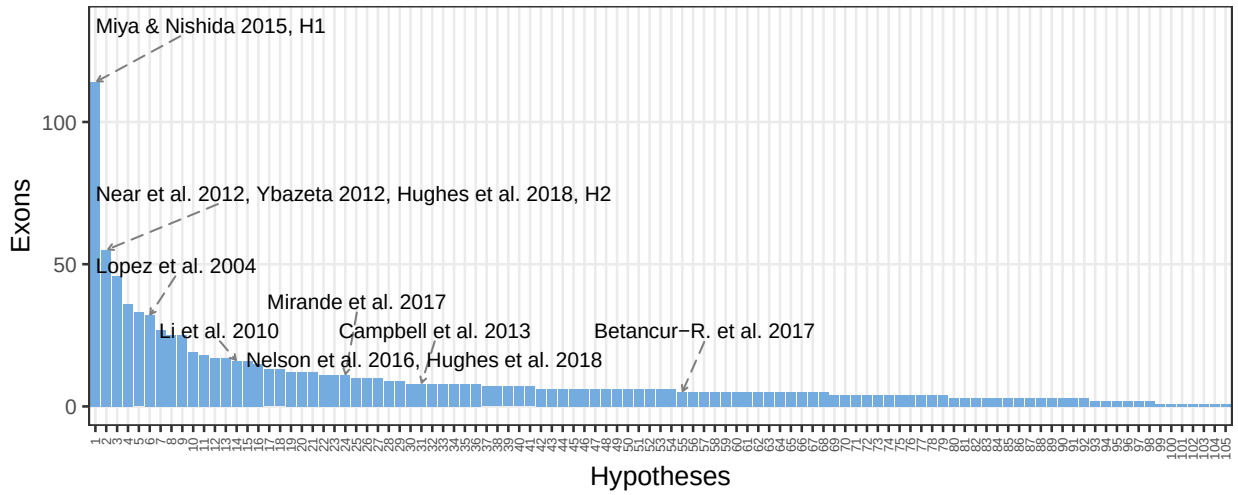


Figure C2: The frequency distribution of 'rank 1' exon loci for each GGI hypothesis in the Protacanthopterygii dataset. Bars represent the frequency of the preferred hypothesis (i.e., the one with the highest AU p-value) for exon loci based on the AU p-value. Arrows above the bars indicate previous studies that support a given GGI hypothesis.

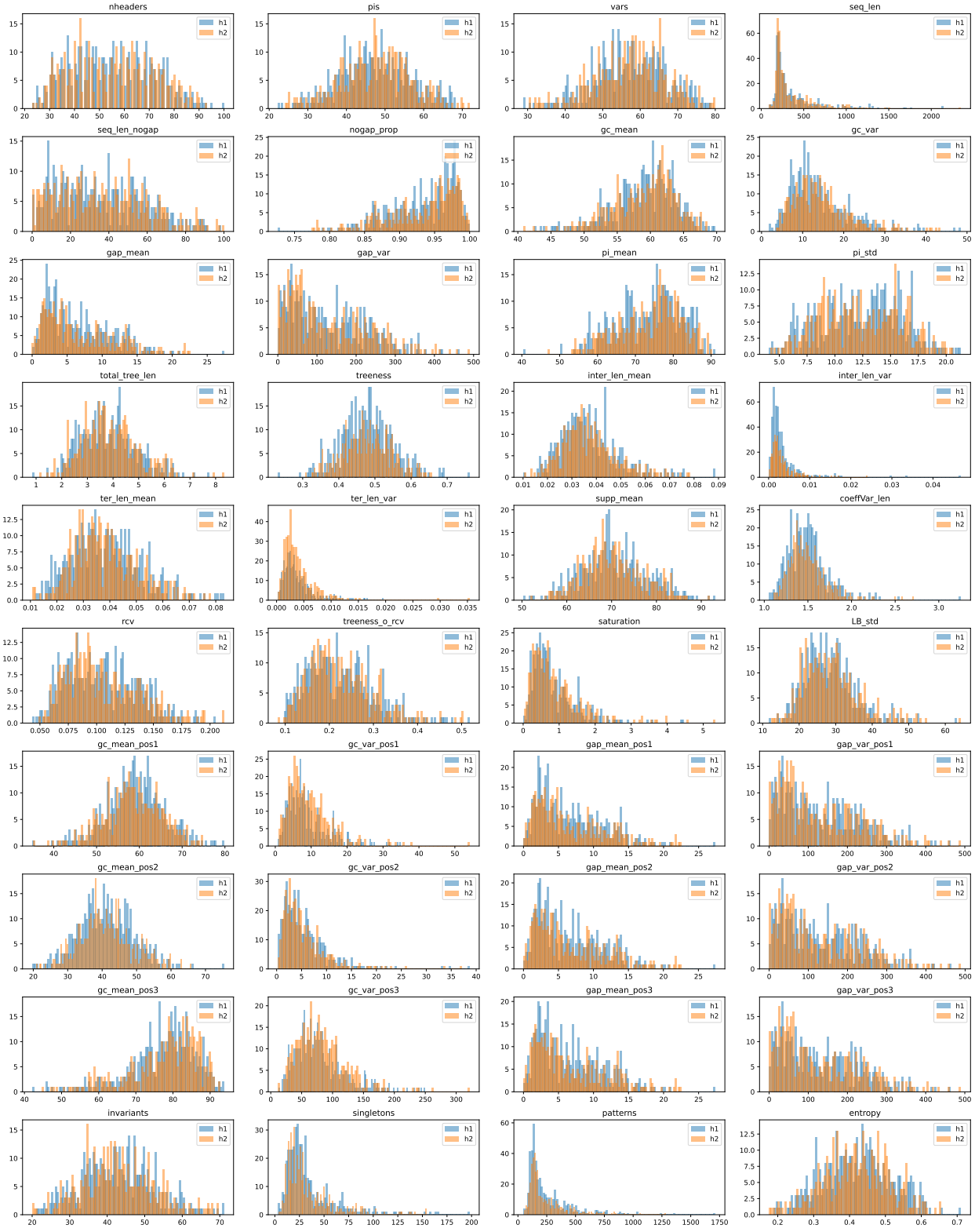


Figure C3: The distribution of the values of the 40 features in the Protacanthopterygii dataset is depicted. The details of the features are provided in Table 4.1. Histograms are colored based on the exons supporting a given hypothesis with the highest probability (i.e., rank 1 exons).

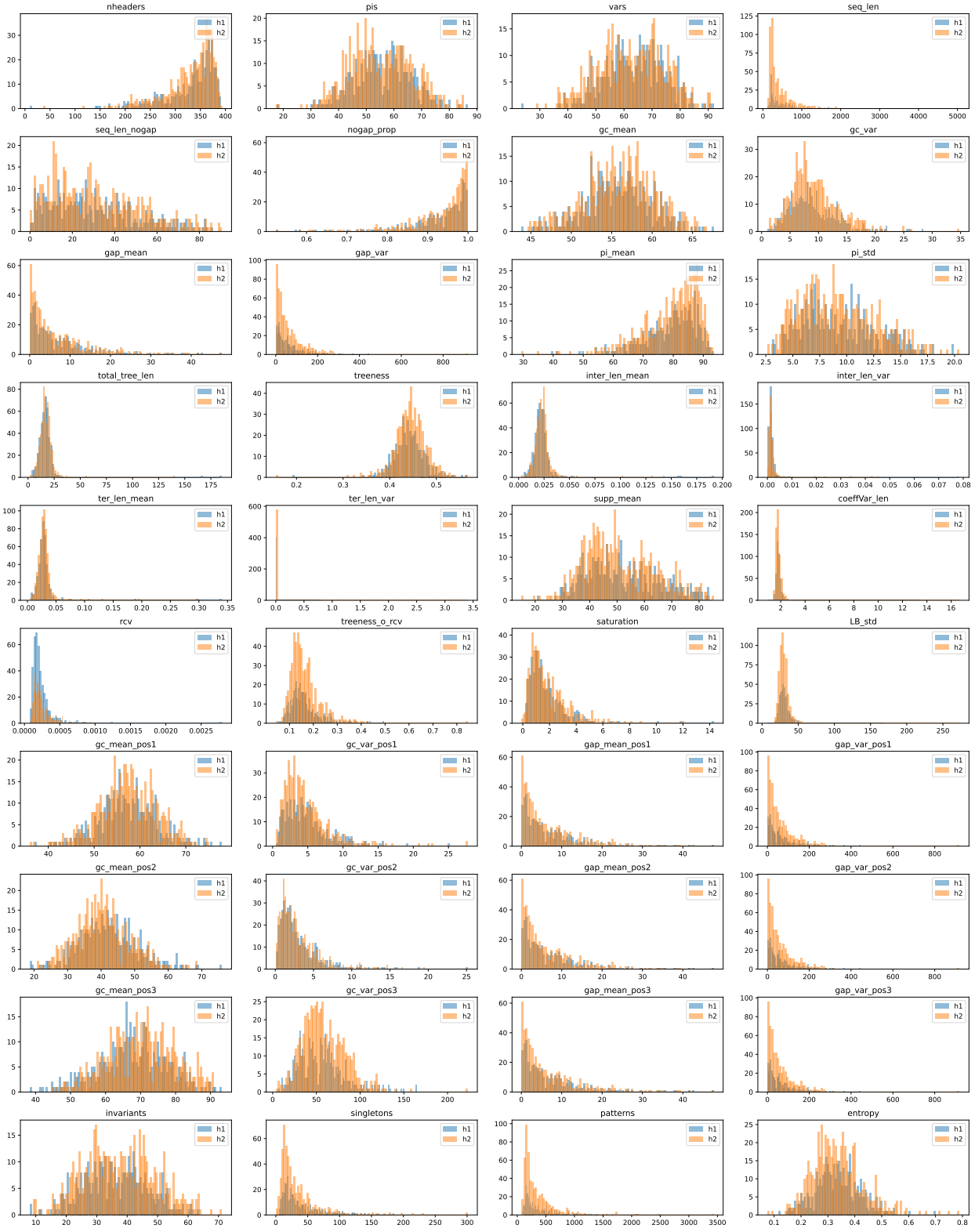


Figure C4: The distribution of the values of the 40 features in the Carangaria dataset is depicted. The details of the features are provided in Table 4.1. Histograms are colored based on the exons supporting a given hypothesis with the highest probability (i.e., rank 1 exons).

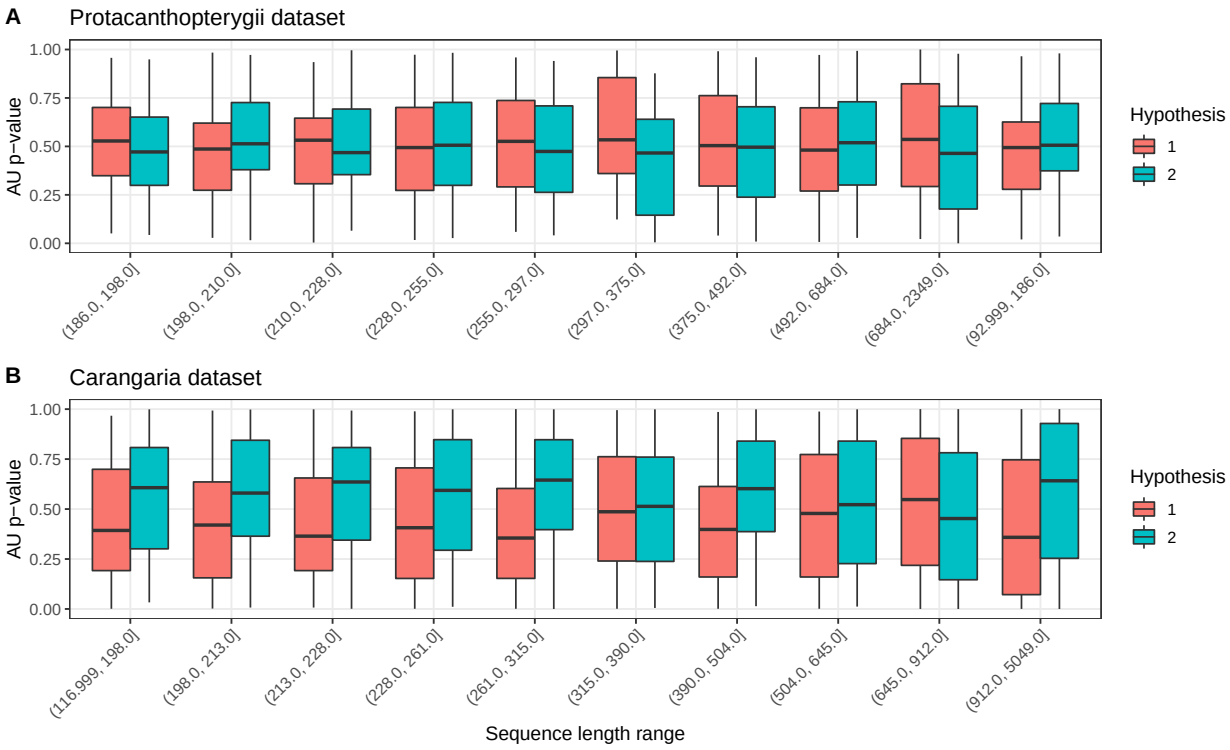


Figure C5: Distribution of both H1 and H2 AU p-values along different range of sequence lengths in both datasets. **A** shows the Protacanthopterygii dataset and **B** shows the Carangaria dataset. For the Protacanthopterygii dataset, the number of loci per bin ranged from 154 to 190. Likewise, in the Carangaria dataset, the count ranged from 164 to 226.

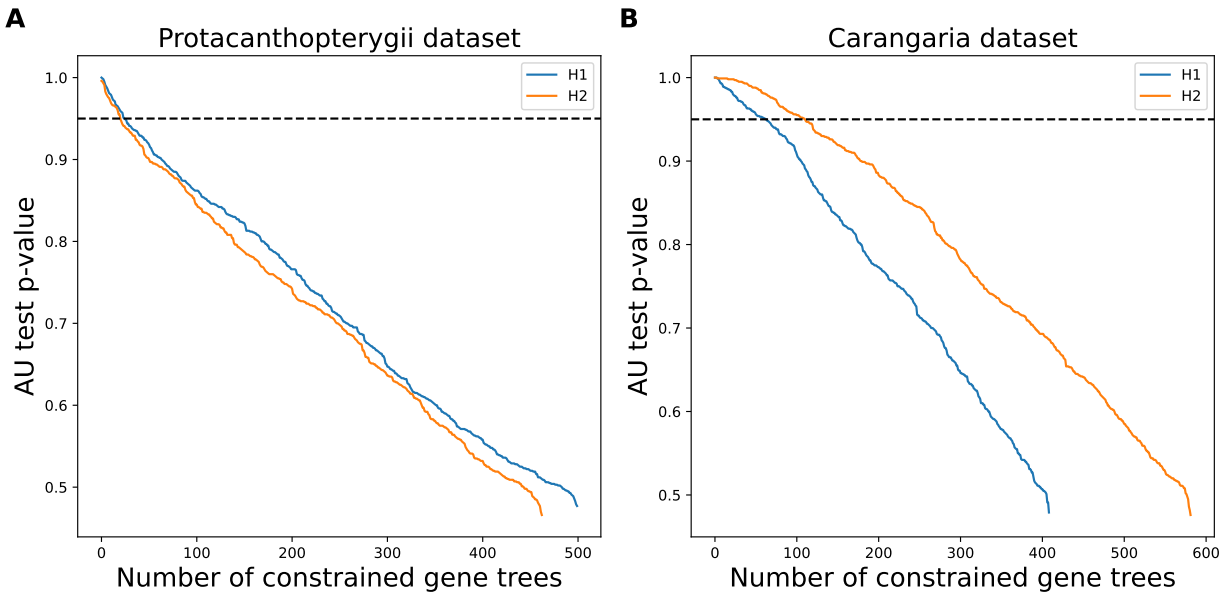


Figure C6: Distribution of p-values for both H1 and H2 in the two datasets. **A** represents the Protacanthopterygii dataset, while **B** represents the Carangaria dataset. Lines in the panel indicate rank 1 hypotheses (the hypothesis with the highest probability). The dashed lines indicate where the hypothesis is significantly better than the alternative one (i.e., the alternative has a p-value < 0.05). It is noticeable that there is no clear rejection of either hypothesis in either dataset for many genes, as the majority of their p-values are between 0.95 and 0.05.

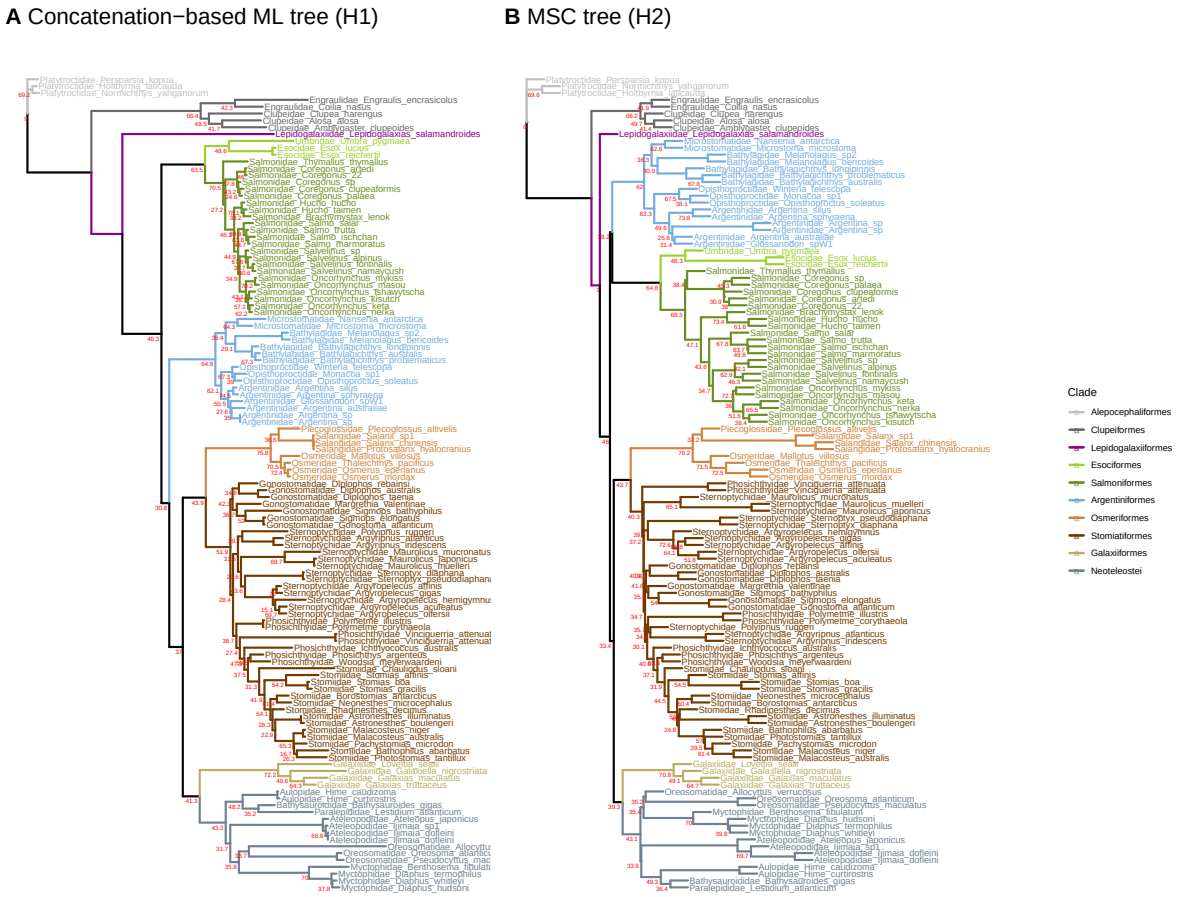


Figure C7: Site concordance factors for the Protacanthopterygii dataset. Only nodes with support lower than 80 are displayed for better visualization. These nodes coincide with clades that form discordance. In **A**, the concatenated-based ML tree is shown with a support of 30.8 for Argentiniformes as a sister clade of ((Stomiatiformes, Osmeriformes), (Galaxiiformes, Neoteleostei)). In **B**, the MSC tree is shown with a support of 38.2 for Argentiniformes as a sister clade of (Esociformes, Salmoniformes)

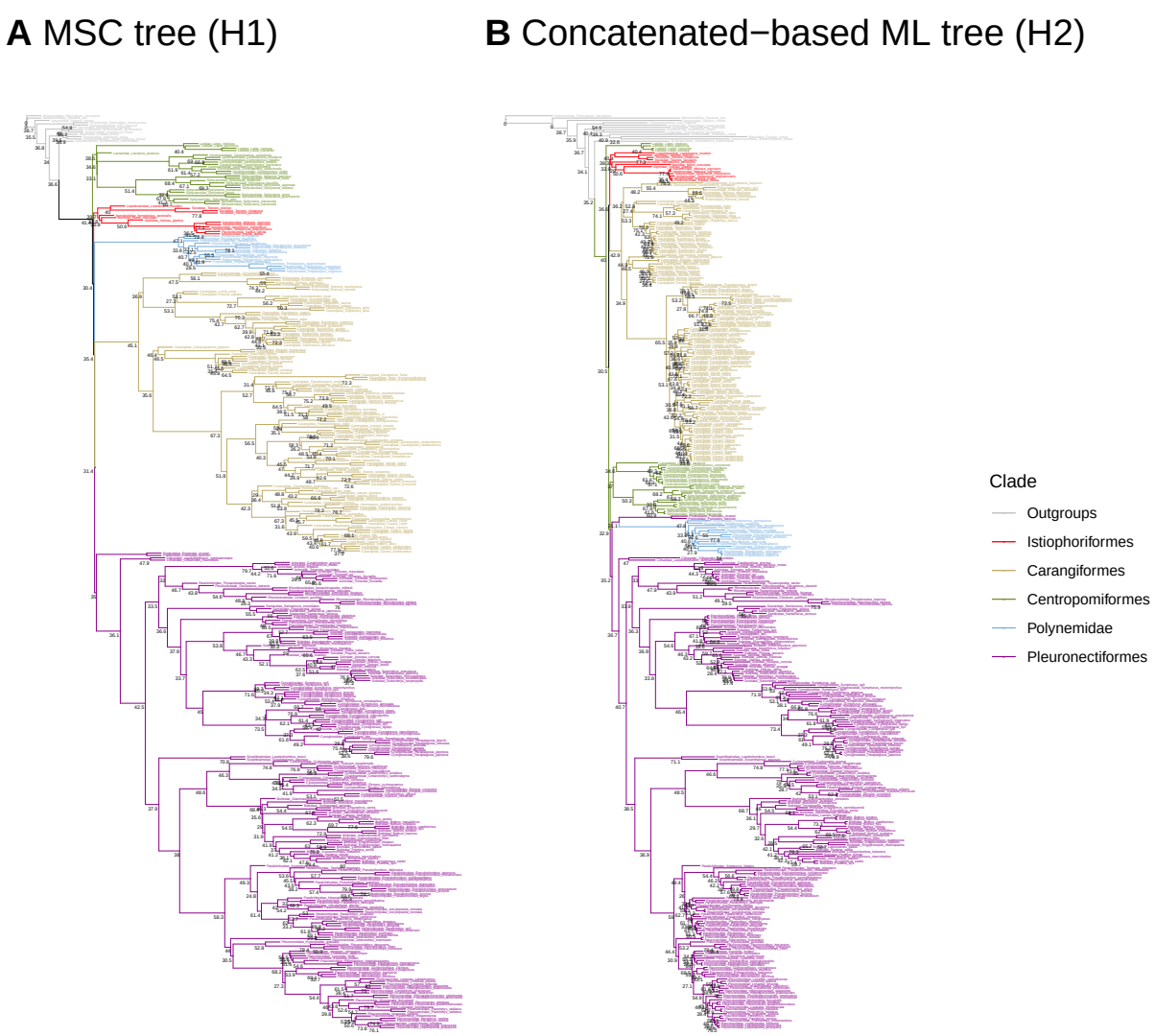


Figure C8: Site concordance factors for the Carangaria dataset. Only nodes with support lower than 80 are displayed for better visualization. These nodes coincide with clades that form discordance. In **A**, the MSC tree is shown with a support of 35 for Pleuronectiformes monophyletic. In **B**, the concatenation-based ML tree is shown with a support of 35.2 for Pleuronectiformes paraphyletic.

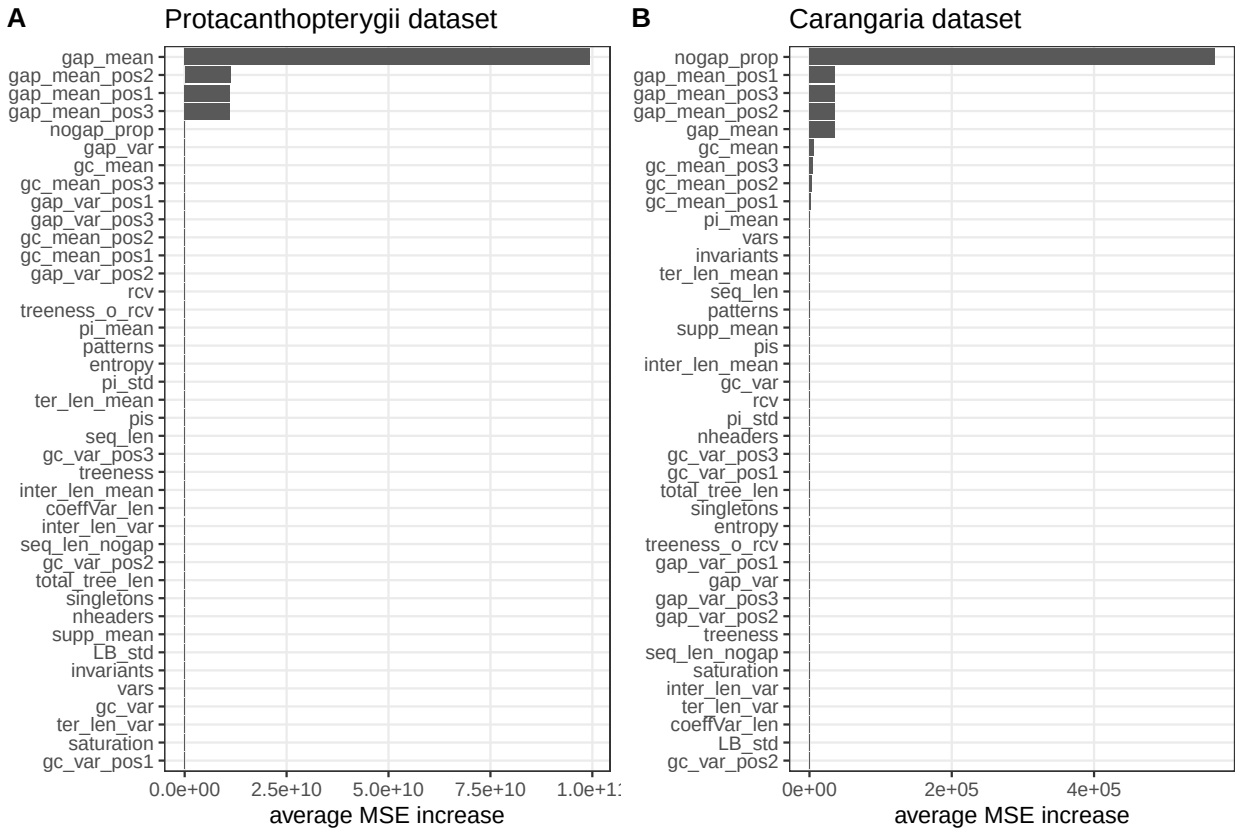


Figure C9: Feature importance profiles for LM models measured by the average increase in testing MSE for the empirical datasets. **A)** Protacanthopterygii dataset, where the LM model identified the average gap percentage ("`gap_mean`") as the top feature. **B)** Carangaria dataset, where the LM model identified the proportion of non-gap characters ("`nogap_prop`") as the top feature. Table 4.1 provides the full names and descriptions of all tested features.

Supplemental references

Arcila, D., and Coauthors, 2021: Testing the utility of alternative metrics of branch support to address the ancient evolutionary radiation of tunas, stromateoids, and allies (teleostei: Pelagiaria). *Systematic Biology*.

Betancur, R. R., E. O. Wiley, G. Arratia, A. Acero, N. Bailly, M. Miya, G. Lecointre, and G. Ortí, 2017: Phylogenetic classification of bony fishes. *BMC Evolutionary Biology*, **17 (1)**, 1–40, https://doi.org/10.1186/s12862-017-0958-3.

Campbell, M. A., J. A. López, T. Sado, and M. Miya, 2013: Pike and salmon as sister taxa: Detailed intraclade resolution and divergence time estimation of Esociformes+Salmoniformes based on whole mitochondrial genome sequences. *Gene*, **530 (1)**, 57–65, https://doi.org/10.1016/j.gene.2013.07.068, URL http://dx.doi.org/10.1016/j.gene.2013.07.068.

Hughes, L. C., and Coauthors, 2018: Comprehensive phylogeny of ray-finned fishes (Actinopterygii) based on transcriptomic and genomic data. *Proceedings of the National Academy of Sciences of the United States of America*, **115 (24)**, 6249–6254, https://doi.org/10.1073/pnas.1719358115.

Li, J., R. Xia, R. McDowall, J. A. López, G. Lei, and C. Fu, 2010: Phylogenetic position of the enigmatic lepidogalaxias salamandroides with comment on the orders of lower euteleostean fishes. *Molecular Phylogenetics and Evolution*, **57 (2)**, 932–936.

López, J. A., W.-J. Chen, and G. Ortí, 2004: Esociform phylogeny. *Copeia*, **2004 (3)**, 449–464.

Mirande, J. M., 2017: Combined phylogeny of ray-finned fishes (actinopterygii) and the use of morphological characters in large-scale analyses. *Cladistics*, **33 (4)**, 333–350.

Miya, M., and M. Nishida, 2015: The mitogenomic contributions to molecular phylogenetics and evolution of fishes: a 15-year retrospect. *Ichthyological Research*, **62 (1)**, 29–71, https://doi.org/10.1007/s10228-014-0440-9.

Molloy, E. K., and T. Warnow, 2018: To Include or Not to Include: The Impact of Gene Filtering on Species Tree Estimation Methods. *Systematic Biology*, **67 (2)**, 285–303, https://doi.org/10.1093/sysbio/syx077.

Near, T. J., and Coauthors, 2012: Resolution of ray-finned fish phylogeny and timing of diversification. *Proceedings of the National Academy of Sciences*, **109 (34)**, 13 698–13 703.

Nelson, J. S., T. C. Grande, and M. V. Wilson, 2016: *Fishes of the World*. John Wiley &

Sons.

Portik, D. M., and J. J. Wiens, 2021: Do Alignment and Trimming Methods Matter for Phylogenomic (UCE) Analyses? *Systematic Biology*, **70 (3)**, 440–462, https://doi.org/10.1093/sysbio/syaa064.

Simion, P., and Coauthors, 2017: A Large and Consistent Phylogenomic Dataset Supports Sponges as the Sister Group to All Other Animals. *Current Biology*, **27 (7)**, 958–967, https://doi.org/10.1016/j.cub.2017.02.031.

Smirnov, V., and T. Warnow, 2021: Phylogeny Estimation Given Sequence Length Heterogeneity. *Systematic Biology*, **70 (2)**, 268–282, https://doi.org/10.1093/sysbio/syaa058.

Xi, Z., L. Liu, and C. C. Davis, 2016: The impact of missing data on species tree estimation. *Molecular Biology and Evolution*, **33 (3)**, 838–860, https://doi.org/10.1093/molbev/msv266.

Ybazeta, G., 2012: MOLECULAR SYSTEMATICS AND BIOGEOGRAPHY OF THE GALAXIIDAE. Ph.D. thesis, 1689–1699 pp.