STATISTICAL ESTIMATION PROBLEMS IN PHYLOGENOMICS AND APPLICATIONS IN MICROBIAL ECOLOGY

BY

MICHAEL NUTE

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Statistics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Doctoral Committee:

Professor Tandy Warnow, Chair, Director of Research
Professor Dave Zhao
Professor William Gropp
Professor Yuguo Chen
Professor Mihai Pop
Professor Rebecca Stumpf

Abstract

With the growing awareness of the potential for microbial communities to play a role in human health, environmental remediation and other important processes, the challenge of understanding such a complex population through the lens of high-throughput sequencing output has risen to the fore. For a de novo sequenced community, the first step to understanding the population involves comparing the sequences to a reference database in some form. In this dissertation, we consider some challenges and benefits of organizing the reference data according to evolution, with orthologous genes grouped together and stored as a multiple sequence alignment and phylogenetic tree.

First we consider the related problem of estimating the population-level phylogeny of a group of species based on the alignments and phylogenies of several individual genes. Under one common model, species tree estimation is provably statistically consistent by several different methods, but those proofs rely on two separate and potentially shaky assumptions: that every species appears in the data for every gene (i.e., there is no missing data), and that since gene tree estimation is itself consistent, the gene trees used to compute the population-level tree are correct. Second, we explore some novel ways to use a Bayesian MCMC algorithm for jointly estimating alignment and phylogeny. The result is increased accuracy for large alignments, where the MCMC method alone would not be tractable. In the process, we identify a peculiar property of this Bayesian algorithm: it performs much differently on simulated sequences than on sequences from biological alignment benchmarks. No other alignment method tested showed the same divergence.

Finally, we present two different practical applications a reference database containing an alignment and tree for a group of gene families in the context of microbial ecology. The first is an algorithm that uses the tree and alignment to construct an ensemble of profile hidden Markov models that improves remote homology detection. The second is a data visualization technique that generates an image of the community with a high density of data, but one that makes it naturally easy to compare many different samples at a time, potentially uncovering otherwise elusive patterns in the data.

Dedicated to Jen, without whose support in so many forms this whole journey would have ended before it began. It was difficult at the time to think of quitting a good job at age 30 to get a Ph.D. as anything but a quixotic fantasy, and if she hadn't taken the idea seriously that's all it would have been. In the time since she has been the breadwinner, been a single parent on many nights, and has dealt with everything else that being the spouse of a graduate student entails without ever once complaining. The research contained herein has been made possible by this heroic support.

Also dedicated to my parents, who have repeated the mantra "work hard" at every opportunity, often to mixed effect. It's taken a long time to fully appreciate that advice and to heed it consistently over a long period, but for everything that went into this manuscript I am confident that, finally, I have done as they suggested.

Acknowledgments

I would like to acknowledge the help and support of a number of collaborators, friends and mentors who have contributed in a variety of ways over the past four years to bring the work in this manuscript to life. I thank Nam Nguyen and Siavash Mirarab for their guidance and collaboration, especially my first year in the lab; I have been working on paying that forward but it's going to take a while. I thank Pranjal Vachaspathi and Erin Molloy for making the journey with me; this whole thing would have been way less fun and interesting without you and it is an honor to have been considered, on paper anyway, your peer. Thank you to all the other members of the lab that I've had the pleasure of working with and getting to know: Kodi Collins, Nidhi Shah, Sarah Christensen, Jed Chou, Ehsan Saleh, Abby Assangba, Negin Valizadegan, Emma Hammel, Thien Le, Srilakshmi Pattabiraman, Karthik Yarlagadda, Nicole Murray, Taylor Crooks, David Wood, Summer Sanford and Ruth Davidson. Thank you to the support staff who have made my life easier at every turn: Elaine Wilson, Melissa Banks, Aaron Thompson, and Samantha Smith.

I also emphatically thank the many senior and junior faculty who have invested their time teaching and collaborating with me: Sebastien Roch, Katie Amato, Dave Zhao, Amy Willis, Bill Gropp, Jian Peng, Yuguo Chen, Luay Nakhleh, Matt Sullivan, Doug Simpson, Alex Stepanov, and Todd Treangen. I thank John Rhodes for his close examination of the proofs in the original paper behind Chapter 3 [164], which led to several important corrections that are reflected in this version. I am especially grateful for the ongoing support and mentorship from Profs. Mihai Pop and Rebecca Stumpf. I am every bit as excited about the microbiome today as when we started working together, and I am grateful that you have let me join you in studying it.

Finally, it is challenging to adequately capture my gratitude for my advisor, Tandy Warnow, but I will try. Thank you for taking a chance on a guy from the stats department with a kid, a four-hour commute and no relevant experience to speak of when you invited me to join your lab. Thank you for sending me to all those conferences; every single time it would fill me with excitement about the possibilities for this work we do. Thank you being so continually supportive despite the burden of my unique geographic situation, my ever-expanding domestic duties and my mercurial disposition. Thank you for letting me postpone graduating for a year to figure things out, *twice*. Thank you for the sheer volume of time you have spent working with me and making me better at this. Thank you for putting so many interesting problems in front of me over the years and for allowing me to follow my curiosity wherever it happened to go. Thank you for many, many lunches. Thank you for insisting on so many practice talks. Thank you for knowing when to refocus my priority list to include getting 30 minutes of sun a day, and knowing when that needed to be the only thing on

it. Thank you for all the long conversations about research, especially the ones that turned into arguments; those were always productive. Lastly, thank you for, in so many varied ways, making all the work in this dissertation possible. I will never forget it.

This work was funded by NSF grant ABI-1458652 (PI: Tandy Warnow), NSF grant III:AF:1513629 (PI: Tandy Warnow), and NSF grant DBI-1062335/1461364 (PI: Tandy Warnow). It was additionally funded by a graduate fellowship from the CompGen Institute at the University of Illinois at Urbana-Champaign for the 2016-2017 school year.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

BV	Bacterial Vaginosis
ESS	Effective Sample Size
GTR	Generalized Time Reversible
HGT	Horizontal Gene Transfer
HMMER	HMMER Profile HMM software [55]
ILS	Incomplete Lineage Sorting
MCMC	Markov-Chain Monte Carlo
ML	Maximum-Likelihood
MSA	Multiple Sequence Alignment
PASTA	Practical Alignments with SATé and Transitivity [139]
PD	Posterior Decoding
PICAN-PI	Population-level Identity, Composition and Novelty via Phylogenetic Insertion
SPFN	Sum-of-Pairs False Negative
SPFP	Sum-of-Pairs False Positive
RAxML	Randomized Accelerated Maximum-Likelihood [206]
RF	Robinson-Foulds error
TC	Total Column Score
TIPP	Taxon Identification via Phylogenetic Placement [157]
SEPP	SATé-Enabled Phylogenetic Placement [140]
UPP	Ultra-large Alignments using Phylogeny-aware Profiles [156]

Part I

Introduction

Chapter 1

Bioinformatic Pipelines for Metagenomics

The last decade has seen an explosion in scientific understanding of the role that a community of microbes can have shaping its environment. The microbes that inhabit the human digestive tract have received the most attention because they appear to be important biomarkers for a variety of disease states and other metrics of host health, but equally fascinating relationships between microbial communities and host environments have occurred in many other contexts. The microbes in soil, for example, affect plant health, and microbes deep within hydraulic fracturing wells could have unforeseen effects on the structural integrity of barriers protecting ground water. But understanding these relationships requires somehow quantifying the contents of a population of diverse organisms, a step that is invariably fraught with trade-offs in choosing what metrics are most important and one that has only become feasible with advent of high throughput DNA sequencing. As discussed in the following section, high-throughput sequencing yields an unordered set of short nucleotide sequences that can number anywhere from several thousand to over a billion for a single environmental sample. In this way, microbial ecology research is reliant on a pipeline of statistical estimation problems to convert such a rich set of data into a useful picture of the community.

1.1 Data: High Throughput DNA Sequencing

The process of capturing data on a particular microbial community works essentially as follows: first a sample of the community is taken, for example by a physical swab of the environment, then it is run through a battery of physical and chemical processes to extract all DNA from the sample from the larger biomass, and the result is a large number of DNA fragments that can be inserted into a sequencing machine where a subset of them are read one-by-one. Since the initial sample contained every organism present in the environment (by contrast with a culture sample, in which one or more specific organisms are selected for), the resulting fragments are assumed to be drawn from organisms representative of the community as a whole.

The data on which this pipeline operates are thus strings, with letters taken from the alphabet $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$. For the standard DNA extraction protocol described above, nothing further is known about each string, including where in the microbial genome it originated, in which case the data is called **metagenomic** sequencing. In some cases though, the DNA extraction may include a step that selects for only sequences that begin with a particular string, which is often done using a string shared by all microbes on one particular gene (called 16S). In that case, called **16S amplicon** sequencing, the location of origin for each sequence within the microbial genome is known, making comparing one sequence to another much more meaningful right

away, but makes 99.9% of the microbial genome unavailable for more in depth analysis. In either case, the strings that are returned form the data that are the basis for statistical estimation.

A first observation about this data is that even in the simpler amplicon scenario, with fewer reads drawn from a single region, the data space is astronomically large. Accordingly, the main challenge with this data is finding a transformation that simplifies it enough to be analytically useful but without obscuring meaningful variations. Techniques for that are myriad but can be broadly classified into those that do not rely on a reference database and those that do; the content of this dissertation deals with the latter. The common thread of parts II, III and IV is the use of molecular phylogenetics to prepare and organize the reference database.

1.2 Organization of This Work

The following chapter covers relevant background such as problem descriptions and statistical models referred to throughout. Following that, the thesis is divided into three parts, most of which is published.

Part II

The two chapters in Part II deal with theoretical models relevant to the problem of estimating a population-level "species" phylogenetic tree. In particular, this can differ in topology from the phylogeny of any individual gene for a variety of reasons, and that makes it a separate and unique estimation problem. The species phylogeny is of interest to microbial ecology because it is the basis for taxonomic organization and classification, and taxonomic identity is the lingua franca of microbiology.

Estimating a species tree relies on the aggregation of multiple genes, and statistical consistency is established as the gene count grows large. In practice, a common constraint is that many genes that may be useful may only exist, or only be available, in a subset of the species. The use of "incomplete" genes like this was not considered in previous proofs of statistical consistency, and in the first chapter we explore models for this kind of data and establish sufficient conditions for statistical consistency for several classes of algorithm.

For the phylogeny of a single gene, on the other hand, the measure of sample size as it relates to statistical consistency is the number of sites, i.e. the length of the gene. But species trees, in addition to requiring a large gene count, require that the gene trees themselves have been estimated consistently. But in practice those assumptions, that both the number of genes and the length of each approach infinity, are at odds with one another, so an interesting theoretical question is whether consistency for the species tree holds when the length of each gene is finite. The second chapter of this part is based on a paper that examined this condition and found that overwhelmingly, under the most common models, the answer is no.

Part III

Part III deals with multiple sequence alignment, an integral step in molecular phylogenetics. Specifically, all three chapters include a quantitative analysis of one particular method: BAli-Phy [184]. This method was first released in 2005 and has been reasonably well cited in that time. As a Bayesian Markov-Chain Monte

Carlo simulation method, the required computational resources have made it difficult to evaluate as part of a large quantitative multiple sequence alignment study. In the few times that it has, however, it has done quite well, albeit often crashing due to memory constraints [116].

The primary question with BAli-Phy is therefore whether it can be made to run reliably and in a reasonable amount of time and still show the same accuracy. And if so, could it be used as part of a divide-and-conquer algorithm to extend that accuracy to alignments of several thousand sequences. The first chapter in Part III is a short description of the first exploratory study of BAli-Phy as a DNA aligner, which were encouraging but inconsistent, showing a distinct difference in accuracy on biological versus simulated data.

The second chapter examines the scalability of BAli-Phy within PASTA and UPP, leveraging the observation of increased accuracy on simulated data.

The third chapter discusses a follow up to the first study, in which a curious dichotomy slowly emerged between the alignments BAli-Phy would produce on simulated data versus on manually curated biological data. In the follow up, we examine protein benchmarks but confirm the observations from the first chapter. BAli-Phy is the only method to show this degree of distinction in performance on biological and simulated data.

Part IV

The final two chapters of this thesis cover two methodological developments with practical applications for microbiome studies. Both show, in different ways, how phylogenetics can be used as the basis for comparing in a sample to known sequences for context.

The first is a technique for screening reads from shotgun sequencing output, specifically when the reference data is organized into gene families. By subdividing the families based on their phylogeny and comparing the reads to each subgroup, an increase in both sensitivity and specificity is possible compared to BLAST and a single HMM. The gain in sensitivity was particularly large and bodes well for the ability to recognize the presence of novel organisms in a sample.

The final chapter gives the design of a novel schema for graphically representing the content of a microbial sample. Specifically, it presents the data output from phylogenetic placement, including specifically a largely ignored metric that quantifies the degree to which a read is different from any sequence in the reference. This results is a high level view of the breadth and extent of the biodiversity in a sample that is, at a minimum, unique.

Chapter 2

Background

2.1 Multiple Sequence Alignment and Phylogenetic Tree Estimation

Molecular phylogenetics operates by gathering a set of orthologous sequences from organisms that share an evolutionary ancestor and using patterns of mutations, insertions and deletions to infer the evolutionary history of the group. Mechanically, this tends to operate in two separate steps: first the sequences are combined into a single alignment that matches homologous sites, and then the tree is estimated using the alignment as input. This is not always exactly the procedure since knowledge of the tree can likewise affect inference about the alignment, but it is illustrative of the two separate but inter-related problems. Once estimated, a phylogenetic tree and its companion alignment can be the basis for additional analysis, such as the addition of sequences from new organisms, or the estimation of a population-level phylogeny whose constraints and evolutionary history may not be the same as individual genes.

Estimation algorithms exist for each of the problems noted above, but two statistical techniques are especially common and deserve their own introduction.

Profile Hidden Markov Models

Hidden Markov Models (HMMs) are a common tool for sequence analysis of all kinds [79]. Generally, there is assumed to be a sequence of hidden states each of which generates (or "emit") an observed variable. The parameters of the model are defined by initial probabilities on the hidden states, transition probabilities between them, and emission probabilities conditional on the hidden state. Variations of all kinds are possible. For nucleotide sequences, we can construct an HMM where the hidden state is either *Match*, *Insertion*, or *Deletion*, and the emission is a nucleotide, or a gap (represented as a "-") if the hidden state is a deletion (see [52] for details, and Figure 2.1 for an illustration of the model structure).

This is the basis for a probabilistic model of sequence generation. Given a set of n aligned sequences the parameters of such a model can be trained using open source software [56]. Once such a model is trained, denoted M, it will have k columns, an initial state distribution $\pi = (\pi_M, \pi_D, \pi_I)$, a transition probability matrix T and for each column i, and a column-wise vector of emission probabilities $\mathbf{p}_i = (p_A, p_C, p_T, p_G)$ (in the case of nucleotide sequences). Thus a path through the hidden states of the model corresponds to an alignment between some new sequence and the sequences the model was trained on. Given a new sequence q we can use the Viterbi algorithm to align q to the training sequences by finding the maximum-likelihood path through the model and the forward algorithm to find the probability that q was generated by this HMM,

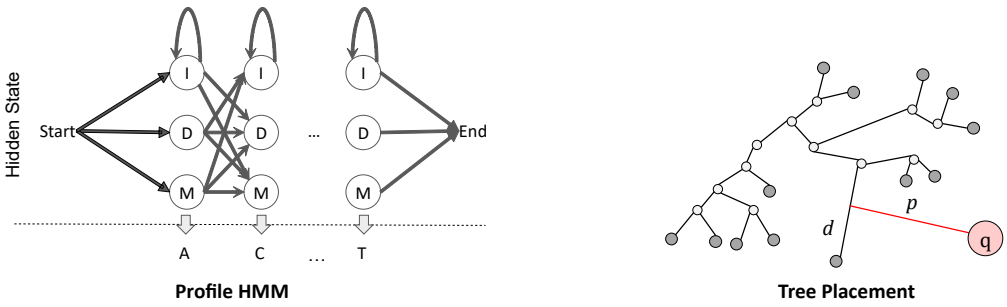


Figure 2.1: Profile HMMs and Phylogenetic Placement Diagram. (Left) A diagram of the graphical model representing a profile HMM for nucleotide sequence generation. The emission probabilities are typically specific to each column in the case of a match state and generic in the case of an insertion state. (Right) A diagram of the output of a tree placement operation on query sequence q. The red branch is added to the tree, and is defined by a) the identify of the branch to which it attaches, b) the *distal* length d between the attachment point and the node in the direction away from the root, and c) the length of the new branch p, also known as the *pendant* length.

$\mathbf{P}_M \left[q|\pi, T, \{\mathbf{p}_i\}_{i=1}^k\right]$.

An HMM of this sort is not a multiple sequence alignment method *per se*, but rather a model representing a group of orthologous sequences. Profile HMMs in general have the nice property that for a given set of orthologous sequences, each column of the alignment is fitted individually and thus proper consideration is given to the variety in levels of conservation across sites. The drawback is that no two sequences are more or less alike than any other in the model, although in a biological context the sequences have a meaningful evolutionary relationship, which is lost in this construction. Also, an HMM is not identifiable from the resulting sequence distribution [173].

Generalized Time Reversible Model

The Generalized Time Reversible (GTR) model is another probabilistic model of sequence generation [216] which, when run, outputs a multiple sequence alignment. This model takes the opposite approach: sequence generation occurs by an evolutionary process over a model tree, although each site (column) of the alignment is necessarily assumed to be *i.i.d.* Formally, let $\mathbf{X} = \{x_i\}_{i=1}^n$ be a set of aligned sequences with k columns in the alignment. The GTR model will generate one column at a time from $\mathbf{X}$: let $\mathcal{T} = (r, T, E, L, \Lambda)$ be a rooted, binary tree with root r, topology T over edge set E, and with leaf nodes $L = \{l_i\}_{i=1}^n$ and branch lengths λ_e, $\forall e \in E$. The root of the tree is defined formally but is a nuisance parameter and not identifiable due to time-reversibility. Let $p_0 = (p_A, p_C, p_T, p_G)$ be a probability vector over the initial character states at the root. Let Q be a 4×4 transition rate matrix for a continuous-time Markov process over the four character

states:

$$Q = \begin{bmatrix} 0 & q_{AC} & q_{AG} & q_{AT} \\ & 0 & q_{CG} & q_{CT} \\ \vdots & \ddots & 0 & q_{GT} \\ & \cdots & & 0 \end{bmatrix}$$

Under this model a single character is simulated from $\mathbf{p}$ at the root. From this initial character state a new character is sampled across each edge leading away from the root. For an edge e with length λ_e, the new character state is sampled from the transition probability matrix given by $e^{-Q\lambda_e}$. (With no constraints on branch lengths or the off-diagonal of Q_{ij} the model has one extra parameter, so conventionally $q_{GT} \equiv 1$ to avoid this.) This process is iterated until there is a newly sequenced character at each leaf node l_i. If k characters are simulated and positioned sequentially, we can simulate aligned sequences such as $\mathbf{X}$.

Most often the GTR model is used to estimate the phylogeny; $\mathcal{T}$, Q and $\mathbf{p}$ are treated as unknown model parameters, with the sequences $\mathbf{X}$ as observed data. With that model, a given tree and model parameters can be given a likelihood score $\mathcal{L}(\mathcal{T}, Q, \mathbf{p})$, which can be used as the basis for a maximum-likelihood (ML) tree search. Unfortunately this optimization problem is known to be NP-Hard [40, 190], but software is nonetheless available to get reasonable estimates, even for large k and large n. Often when the alignment includes insertions and deletions (leading to blank characters), the character values are treated as missing data. Additional refinements on this model are possible, but this is the bedrock optimization problem for phylogenetic gene tree estimation, and it is known to be statistically consistent in the length k [63].

Phylogenetic Placement An extension of maximum-likelihood tree estimation is what's known as phylogenetic placement. Here, the input is an already-estimated multiple sequence alignment X and ML tree $(\mathcal{T}, Q, \mathbf{p})$, along with a single "query" sequence q that is orthologous to those in the MSA and we can assume has been aligned to it. The output is a new tree $\mathcal{T}' = \mathcal{T} + e_q$ which is equal to $\mathcal{T}$ but one additional branch e_q having the q at its leaf node. With $\mathcal{T}$ fixed, e_q is defined by $(e_0, t_{e_0}, \lambda_{e_q})$, where $e_0 \in E$ is the edge in $\mathcal{T}$ to which the new branch attaches, $t \in [0, 1]$ is the attachment location as a proportion of λ_{e_0} (typically where $t = 0$ is the distal end of e_0), and λ_{e_q} is the length of the new branch. See Figure 2.1 for an illustration of the placement process and its outputs. This can be done using maximum likelihood as well:

$$\hat{e}_q = \arg\max_{e_q} \mathcal{L}(\mathcal{T} \cup e_q, Q, \mathbf{p})$$

Phylogenetic placement can be solved in polynomial time, but finding an exact optimum for a moderately sized reference tree can quickly become memory constrained, so in practice some heuristics are often used to accelerate the search.

Discussion

In addition to ML phylogeny estimation, multiple sequence alignment is also NP-complete in its simplest form. Specifically, maximizing the pairwise site matches within an alignment (a.k.a. the sum-of-pairs score)

remains NP-complete even when the length of the sequences is strictly bounded from above [99]. As a result, the combined problem of alignment and tree estimation is uniquely challenging in an era where in practice the number of sequences needing analysis is growing rapidly, and solutions must rely on clever heuristics to find suitable approximations. Much of the work contained herein, particularly in Parts III and IV, is the intellectual descendant of one such heuristic method developed in 2009 called SATé [116] that greatly increased the number of sequences that could be aligned with any expectation of accuracy. The details of SATé and what came shortly after require a separate background.

2.2 Large-Scale Multiple Sequence Alignment

The original SATé publication was followed three years later by SATé-II [117] and again in 2015 by PASTA [139], all of which follow broadly the same divide-and-conquer approach to large-scale multiple sequence alignment. The particular nuances that distinguish them from one another are largely beyond the scope of this introduction so this discussion applies to all three incarnations but will default to PASTA as it is the most recent.

Algorithm

The algorithm employs a divide-and-conquer strategy to iteratively generate an alignment, followed by a tree until a stopping condition is met. At each step, the most recent tree guides the divide-and-conquer operations, which produces comparatively easy-to-align subsets that are then merged to create an improved alignment on the whole set. The improved alignment is then used to estimate an improved tree, et cetera. Figure 2.2 illustrates the steps of one iteration. At each step, PASTA relies on other tools to do certain individual tasks. Subset alignments are performed (by default) using MAFFT [97], pairwise merging of subset alignments is done using OPAL [234], and tree estimation is done by FastTree-2 [179], although in each case other options are available.

Differences between the three incarnations of this algorithm are driven by, for example, the algorithm used to choose breaking edges in the tree decomposition in Step 2, the specifics of how pairwise subset alignments were merged into a single MSA, and the stopping criteria for the iterations. Much of the work in Part III began shortly after PASTA was published. Since PASTA and SATé-II each represented significant and consistent improvements over the previous version developed largely by modifying and tuning the previous version, that strategy prevailed as PASTA was developed further, with particular focus on finding a way to use BAli-Phy [184] within the subset alignment step to leverage higher accuracy that had sometimes been observed.

Other Methods Based on SATé

As SATé and PASTA were being developed, other research looked at the usefulness of pairing the tree decomposition strategy from SATé with profile HMMs to create a hybrid structure that enjoys the benefits of both: the site-specific detail of the HMM with the evolutionary consistency of a tree model. The approach

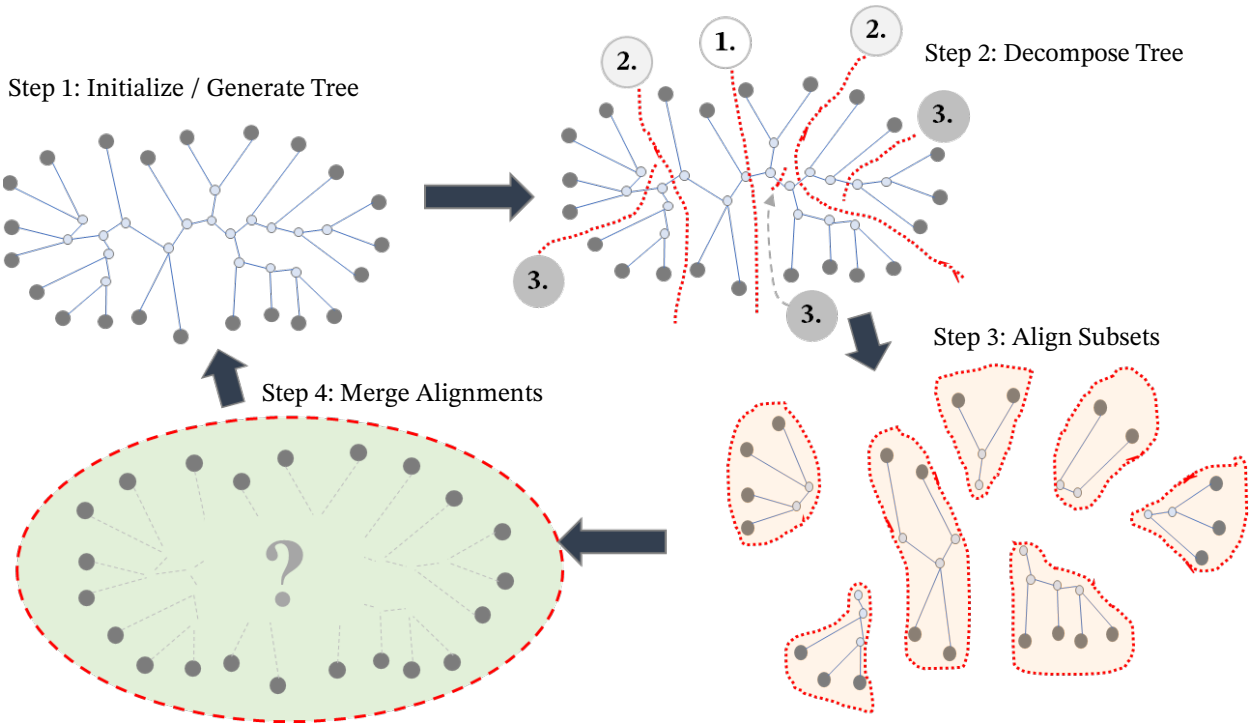


Figure 2.2: **One iteration of PASTA.** (Top Left) A fast tree estimation algorithm is used to initialize a tree from a different, fast alignment. (Top Right) The tree is recursively bisected as long as it has a minimum number of leaves: the numbers accompanying each bisection (red dotted-line) denotes the order in which it occurs. (Bottom Right) The subsets of sequences grouped by the decomposition are each, individually aligned using a separate algorithm. (Bottom Left) The subset alignments are merged pairwise to generate a single MSA. Returning to step 1, a tree is estimated from this MSA and iterations continue until a stopping condition is met.

was simple: take a well-constructed MSA-tree pair, apply the SATé decomposition to the tree, and generate a profile HMM for every subset. The resulting ensemble of HMMs has in practice been shown to improve both sensitivity and precision for analyses that depend on detection of homology and sequence alignment.

This basic approach was used in SEPP [140] to obtain more accurate alignments of the query sequences to the reference, which in turn improves placement accuracy. Additionally, this step effectively localizes the query sequence to a region of the tree and allows the option of placing into a sufficiently large and properly chosen subtree with a minimum cost of accuracy, a convenient boosting option when the full tree is too large. In TIPP, the SEPP algorithm was used in the context of sequences from a microbial community, but placed into a refined taxonomy so that the placement would correspond to a taxon identification. In UPP [156] it was used to conduct alignment on a large input set when the input set contained an unknown but potentially large number of fragmentary sequences. It was additionally used in HIPPI [158], which is described in Chapter 8.

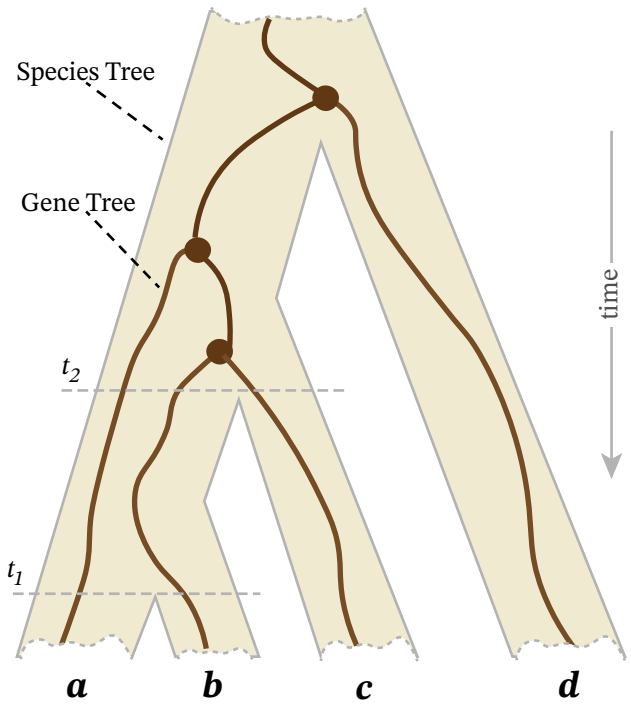


Figure 2.3: **ILS Example.** Above, a population level species tree on four taxa is represented by the thick yellow tree in the background. Within it, the lineages of an individual gene tree coalesce at random after they have entered a common population. This can result, as it does here, in a gene having a different evolutionary history from its host. Here the gene tree displays $(a, (b, c), d)$ whereas the species tree displays $(((a, b), c), d)$. The MSC models this as a compound Poisson process occurring in reverse time. Once two lineages enter the same population, their time until coalescence follows an exponential distribution.

2.3 Species Tree Estimation and ILS

The approach to estimating a phylogenetic tree described in Section 2.1 above refers specifically to the evolutionary history of a *single gene*. A well-known phenomenon in evolutionary biology is that two genes shared among the same set of organisms can demonstrate different evolutionary histories, but an important example is Incomplete Lineage Sorting (ILS) [130]. ILS is a population-level phenomenon, where the population of a particular organism has a tree that represents its overall speciation events, but within the population tree are the lineages of the individual genes, each with their own branching events at their own times (see Figure 2.3).

The multi-species coalescent (MSC) [102] is a stochastic model of this phenomenon that takes the species tree $\mathcal{T}$ as an input parameter and describes the probability of the gene trees it may generate. Here the edges of $\mathcal{T}$ have a branch length denominated in "coalescent-units", not the same as time although analogous with the exception that $\mathcal{T}$ need not be ultrametric in coalescent units. Gene trees are simulated according to the following process. One lineage begins at each leaf of the species tree and advances in reverse time. When two lineages enter a common population, they continue advancing but are subject to coalescent events, which happen according to a (compound) homogenous Poisson process with rate θ. If three or more enter

the same population, all pairs are subject to the same rate of coalescence. In Figure 2.3, a and b enter the same population at the point t_1, but they do not coalesce until above the point t_2, where c entered the same population.

Methods and Challenges

Let $\mathcal{T}$ be the (unknown) model species tree that has generated a set $\mathcal{G} = \{g_i, t_i\}_{i=1}^{m}$ of m genes on n species. Each gene consists of a set of (aligned) sequences $X_m = \{x_{mj}\}_{j=1}^{n}$ which are generated by its gene tree t_i, which is also unknown. For a biologist looking for an accurate representation of the population-level evolutionary history of a set of species in the presence of ILS, reconstructing it from genetic data is non-trivial; in some cases, the most like gene tree topology may actually be different from $\mathcal{T}$ [2].

One often used approach historically has been to concatenate the sequences from multiple individual genes and get a maximum likelihood tree. That approach often works well in practice when ILS is low but is not statistically consistent as $m \to \infty$. That is, for some model trees the error will not approach zero in probability as more genes are observed.

SVDquartets [39] uses site pattern frequencies as an indicator of quartet likelihood for the true species tree.

A different category known as "summary" methods involve first estimating the gene trees for each gene individually, then combining the resulting trees. The advantage is that given all correct gene trees, these can be statistically consistent for $m \to \infty$. ASTRAL [143] does this by tabulating quartet frequencies over $\mathcal{G}$ and seeks the tree whose quartet topologies agree with the most possible quartets from among the gene trees. NJst [119] and ASTRID [227] both construct a distance-matrix D from the average internode (topological) distance over the gene trees and constructs $\hat{\mathcal{T}}$ from D.

There are numerous other methods as well, although each one has to balance a) finite gene length creating gene tree error, b) optimization problems to find $\hat{\mathcal{T}}$ that are often NP-hard, c) model misspecification and d) questions about statistical consistency.

Part II

Some Limit Theorems in Phylogenomics[1]

[1] With apologies to R.R. Bahadur.

Chapter 3

Statistical Consistency of Coalescent-Based Species Tree Estimation under a Model of Deleted Taxa

3.1 Introduction [1]

The estimation of a species phylogeny from multiple loci is confounded by biological processes such as horizontal gene transfer and incomplete lineage sorting that cause individual gene tree topologies to differ from that of the overall species tree [130]. Incomplete lineage sorting (ILS), which is modeled by the well-studied multi-species coalescent (MSC) model, is considered to be perhaps the major cause for this discordance [59]. Many methods have been developed to estimate the species tree in the presence of ILS in a statistically consistent manner, which means that as the amount of data increases, the species tree topology estimated by the method converges in probability to the true species tree topology. Examples of methods for species tree estimation that are statistically consistent under the MSC include ASTRAL [141, 143], ASTRID [227], *BEAST [82], BEST [118], the population tree in BUCKy [106], GLASS [149], MP-EST [120], METAL [47], NJst [119], SMRT [48], SNAPP [31], STEAC [116], STAR [116], STEM [104], and SVDquartets [38]. Some of these methods (e.g., ASTRAL, ASTRID, BUCKy-pop, and NJst) estimate just the species tree topology but not the branch lengths in coalescent units, while others (e.g., BEST, *BEAST, and MP-EST) also estimate the branch lengths. In this chapter, we will refer to all methods that have been proven to be statistically consistent under the MSC as "coalescent-based species tree estimation methods".

One of the key assumptions in the proofs of statistical consistency for standard methods is that every gene is present in every species. This assumption is unrealistic for many empirical datasets (e.g., the plant transcriptome dataset studied in [236]), which can have substantial missing data. The impact of missing data on species tree estimation has mostly been investigated from an empirical rather than theoretical standpoint. Early studies focused on the impact of missing data on the estimation of individual gene trees [75, 177, 253], while later studies examined the impact on multi-locus species tree estimation but without any gene tree discord [111, 238, 239, 240]. Four recent studies have examined the impact of missing data on species tree estimation using multiple loci, when gene trees can differ from the species tree due to ILS [87, 227, 245, 209]. These studies have largely focused on whether it is better to include taxa and/or genes that have substantial amounts of missing data (e.g., taxa that are absent for 50% or more of the genes), and the relative performance of different coalescent-based species tree estimation methods in the presence of missing data. In general these studies have shown that although deleting whole genes from the overall data reduces accuracy

[1] This chapter contains material previously published in [163]. This chapter contains a correction to the proof of Theorem 12 as it appears in that publication. The proofs were developed in part with the help of Jed Chou and the writing was revised by Tandy Warnow. The copyright owner has provided permission to reprint.

compared to having no missing data, including gene data (even if they are highly incomplete) may be on the whole beneficial to species tree estimation efforts, at least for many coalescent-based species tree estimation methods.

Yet, the question of whether coalescent-based methods are statistically consistent under the MSC in the presence of missing data has not been addressed. This chapter examines whether standard coalescent-based species tree estimation methods remain statistically consistent in the presence of missing data. We explore this question under a simple *i.i.d.* model of missing data (where every species is missing from every gene with the same probability $p > 0$, and denoted M_{iid}), and also under a more general model of missing data where, for some constant k, each subset of k species has non-zero probability of being present in a randomly selected gene. We refer to this as the "full subset coverage" model (denoted M_{fsc}). The M_{fsc} model includes the simpler *i.i.d.* model as a special case, but also includes the models of taxon deletion considered in [87] and [245].

In this study, we address the question of whether coalescent-based species tree estimation methods are statistically consistent under the M_{iid} or M_{fsc} models of taxon deletion. We focus on coalescent-based species tree methods that operate by computing summary statistics for subsets of the taxon set and using those summary statistics to estimate the species tree. We show that whenever these calculated summary statistics are not impacted by deleting species outside the subset of interest, then the coalescent-based species tree method will be statistically consistent under the M_{fsc} model of taxon deletion. We also discuss taxon-deletion models under which species tree estimation methods cannot be statistically consistent, and we finish by discussing the impact of missing data on species tree estimation in practice.

3.2 Background

3.2.1 Problem statement and notation

The multi-species coalescent is a population genetics model that describes the evolution of individual genes within a population-level species phylogeny [102]. Specifically, a species phylogeny $\mathcal{T} = (T, \Theta)$ with topology T and branch lengths Θ is given (but unknown) on a set of n-taxa, $\mathcal{X} = \{x_i\}_{i=1}^{n}$, where the branch lengths are denominated in "coalescent units." This species tree then parameterizes a probability density function for a random variable $G(\mathcal{T})$ defined over all possible phylogenies of $\mathcal{X}$. For a gene tree $g \sim G(\mathcal{T})$, an additional assumption can be made regarding a sequence evolution model which may generate a set of sequences $s_g = (s_{g1}, \ldots, s_{gn})$ for each taxon in $\mathcal{X}$. Let the leaf set of gene tree g be denoted as $\mathcal{L}(g)$. Given a collection of genes $g_1, \ldots, g_m \sim G(\mathcal{T})$, the coalescent-based species tree estimation problem is the challenge of estimating the topology T from the input data I_m, which may include the gene trees, the accompanying sequences or both.

Thus coalescent-based species tree estimation methods can work with a variety of different types of inputs. Usually such methods assume that the estimation of gene trees given sequence data can be done with statistical consistency, which is true in the case of the most common models [199]. In this chapter, we will consider the input data I to include, broadly, the gene trees themselves (one per gene), with or without

branch lengths, or the multiple sequence alignments (one per gene), or both depending on the method. In either case, it is natural to consider the input data I as being potentially restricted to a subset $\mathcal{X}'$ of the taxa by considering, respectively, the subtrees of each gene tree restricted to the leaves with taxa in $\mathcal{X}'$, or the multiple sequence alignment of only the sequences corresponding to taxa in $\mathcal{X}'$. We will refer occasionally to this restricted data as $I|_{\mathcal{X}'}$. In contexts where the number of genes may vary and is indexed by m, the input data I on m genes is correspondingly indexed as I_m.

Tuple-based methods We will establish properties about statistical consistency in the presence of missing data for a class of coalescent-based species tree estimation methods that we collectively refer to as "tuple-based methods". As we will show, nearly all coalescent-based species tree estimation methods that have been proven to be statistically consistent under the MSC are tuple-based, so this restriction covers most of the methods in use.

A coalescent-based method is a "tuple-based" method if there is some $\ell \in \mathbb{Z}_{\geq 2}$ such that the method operates by computing a set of summary statistics from the input I for every subset of ℓ species, and then uses these summary statistics to compute the species tree. Furthermore a tuple-based method is called an ℓ-tuple-based method (or more simply an ℓ-tuple method) to reflect the specific value of ℓ on which it bases its summary statistics. We write each tuple-based method as a pair (F, α), with F the function that computes the set of summary statistics from I, and α the function that computes a species tree given $F(I)$. Also, the set of summary statistics computed by an ℓ-tuple method includes one statistic for every tree topology (possibly rooted) on every subset of ℓ species.

Since a "tree" on two species is just a path, the 2-tuple methods compute pairwise distances for every pair of species. Examples of 2-tuple methods include NJst and ASTRID, which operate by computing the "average internode distance" between every pair of species. Other 2-tuple based methods include GLASS [149] and its variants (e.g., [94]), METAL [47], STAR [116], and STEAC [116], which also compute a pairwise distance between every pair of species, but use a different technique to do the calculation. 2-tuple methods then compute a tree on the matrix of pairwise distances, using methods such as Neighbor Joining (NJ) [196] or FastME [110]; thus, NJ and FastME serve as the function α in the 2-tuple method.

MP-EST and SMRT are 3-tuple methods. MP-EST requires rooted gene trees (and so depends on the strict molecular clock), and uses the frequency of each rooted 3-leaf tree t induced in the input set of gene trees as the summary statistic for t. It then seeks the model species tree (topology and branch lengths) that is most likely to produce the observed distribution of rooted 3-leaf gene tree frequencies. SMRT is a site-based method that estimates rooted three-leaf subtrees from the concatenated gene sequence alignments, and so depends on the strict molecular clock. SMRT then combines the rooted three-leaf subtrees into a tree on the full set of taxa using the Modified Min Cut algorithm [169].

In contrast to 3-tuple methods (e.g., MP-EST and SMRT), 4-tuple methods operate on unrooted gene trees, and so do not depend on the strict molecular clock. 4-tuple methods begin by computing either the most likely tree on every four leaves, or by computing some real-valued statistic for each unrooted tree on every four leaves. An example of a 4-tuple method is ASTRAL [141, 143], which uses the frequency of quartet tree t induced in the input gene trees as the real-valued support for t. Other 4-tuple methods include

the population tree in BUCKy [106] (called BUCKy-pop in [249]) and the implementation of SVDquartets [38] within PAUP* [213]; in these two cases, the real-valued support for a quartet tree is either 1 or 0 (i.e., the best quartet tree on every four leaves is determined). All these 4-tuple methods then compute a species tree by applying some quartet amalgamation method to the set of quartet trees, weighted by their support values. For these 4-tuple methods, α is the quartet amalgamation technique used to construct the tree T from the set of weighted quartet trees.

The number of summary statistics that each type of method computes depends on the value for ℓ and the number n of species: 2-tuple methods compute $\binom{n}{2}$ summary statistics (one for each pair of species), 3-tuple methods compute $3\binom{n}{3}$ summary statistics (one for each rooted three-leaf tree), and 4-tuple methods compute $3\binom{n}{4}$ summary statistics (one for each unrooted four-leaf tree).

The proofs of statistical consistency for ℓ-tuple-based methods have the following basic steps: first, they show that as the number m of genes increases, the vector of summary statistics computed by F on input data I_m converges in probability to a constant vector (which we will refer to as F_0). Second, they show that $\alpha(F_0) = T$, where T is the topology of the true species tree. Third, they show that there is some $\delta > 0$ so that whenever $L_\infty(F_1, F_0) < \delta$ then $\alpha(F_1) = \alpha(F_0) = T$ (here L_∞ is the infinity-norm, i.e. the maximum absolute difference of individual vector components). It follows that the algorithm $A = (F, \alpha)$ is statistically consistent under the MSC. Therefore, when we refer to a statistically consistent ℓ-tuple method, we will assume that these properties hold for the method, and then study the impact of missing data on the method.

Proofs of statistical consistency for many coalescent-based methods typically require several extra conditions. For example, the proofs of statistical consistency of SVDquartets, MP-EST, STEM, STAR, and SMRT require that sequences evolve under the strict molecular clock. Similarly, the proofs of statistical consistency for nearly all methods that operate by combining gene trees require completely correct gene trees (for an exception to this rule see [193]), and it is unknown whether any standard coalescent-based methods that estimate species trees by combining gene trees are statistically consistent in the presence of gene tree estimation error. Another complication in the proofs of statistical consistency is the typical requirement that α provide an exact solution to an optimization problem (e.g., finding the species tree that maximizes some optimization criterion with respect to the input gene data). This is generally not an issue for 2-tuple methods, which use methods like neighbor joining [196] to compute trees from distance matrices, but can be a problem for 3-tuple and 4-tuple methods. For example, 4-tuple methods tend to have two steps, where the first step computes a set of quartet trees (using F) and the second step computes a tree from the set of quartet trees using α. Since quartet tree compatibility is NP-hard [207], quartet amalgamation methods are typically heuristics that have no guarantees (the dynamic programming algorithm in ASTRAL is one of the few exceptions to this), and may not even be guaranteed to return a tree T when given its set of quartet trees. Thus, statistical consistency of coalescent-based methods is complicated, even when there are no missing data.

Extension of tuple-based methods to missing data These tuple-based methods are defined and described on the assumption of gene trees or sequence alignments without missing data, and the statistics or the algorithm may not be fully defined if not all taxa are present. For example, for a given gene tree, the topology for

quartet $ijkl$ does not exist if one or more of the species is missing from the gene. Intuitively, if the method would have called for the calculation of a statistic on a particular set of taxa for a particular gene, it is not possible to calculate this if any taxon in the set is not present, so that gene should be excluded for purposes of that statistic. Thus, the natural extension of a tuple-based method (F, α) to inputs with missing data (species missing from genes) is as follows:

Definition 1. *Let $A = (F, \alpha)$ be an ℓ-tuple species tree estimation method. The **natural extension** of A computes the summary statistics for a given set B of ℓ species based only on those genes that contain all the species in B.*

Type 1 and Type 2 ℓ-tuple methods Since the set of summary statistics includes a real number for every tree t on ℓ species, we will let $F_t(I)$ denote the summary statistic computed by the function F for tree t given input I. For a set B of ℓ species drawn from the full set $\mathcal{X}$ of species, let $I|_B$ denote the input set I restricted to B; thus, all species in $\mathcal{X} \setminus B$ are deleted entirely from the input. Then tuple-based methods can be characterized further depending on how they behave on such inputs. Specifically, we will partition ℓ-tuple methods (F, α) into two categories:

- Type 1: For all inputs I, all sets B of ℓ species from $\mathcal{X}$, and all trees t on B, $F_t(I) = F_t(I|_B)$.
- Type 2: There is at least one input I, one set B of ℓ species, and one tree t on B such that $F_t(I) \neq F_t(I|_B)$.

Thus, a Type 1 ℓ-tuple method has the property that deleting taxa from outside a set B does not impact the summary statistics it computes for any tree on B. Note that taxon deletion impacts both Type 1 and Type 2 methods, in that if enough taxa are deleted from enough genes then accuracy must decrease. As we will see, Type 1 methods are easier to analyze than Type 2 methods, and in particular it is easy to prove that a Type 1 method remains statistically consistent in the presence of missing data for some models of random taxon deletion. Most statistically consistent coalescent-based methods are Type 1 tuple-based methods; for example, ASTRAL, GLASS, METAL, MP-EST, STEAC, and SVDquartets are all Type 1 tuple-based methods. ASTRID, NJst, and STAR are Type 2 methods.

3.2.2 Taxon Deletion Models

Let $\mathcal{T}$ be a species tree on a set $\mathcal{X}$ of n species, with $\mathcal{X} = \{x_i\}_{i=1}^n$, and let m gene trees evolve within $\mathcal{T}$ under the multi-species coalescent model. We denote the set of gene trees by $\mathbf{T} = \{T_i\}_{i=1}^m$, and the set of genes by $\mathcal{G} = \{g_i\}_{i=1}^m$. To model taxon deletion, we let g_i denote an arbitrary gene, and $Y_i = [Y_{i1}, \ldots, Y_{in}]^T$ be a random n-dimensional vector where

$$Y_{ij} = \mathbb{I}_{\{x_j \text{ is present in } g_i\}} \tag{3.2.1}$$

Here each individual Y_{ij} is a binary random variable that represents whether a species x_j is present for a random gene g_i.

Exchangeability For the following lemma, we will assume that if T_i is generated before Y_i, then the post-deletion tree $T_i^* = T_i|Y_i = y_i$ is obtained by taking the subtree of T_i restricted to the set of leaves $\{x_j|y_{ij} = 1\}$, that is, the set of leaves whose taxa have not been deleted. If Y_i is generated before T_i, then T_i^* is obtained by taking the same subtree of the species tree $\mathcal{T}$, $\mathcal{T}|Y_i$ and simulating a gene tree within this species subtree under the multi-species coalescent.

Lemma 2. *If T_i and Y_i are independent, the two variables are exchangeable and the distribution of T_i^* does not depend on the sequence.*

Proof. If Y_i is generated first, then the conditional distribution of T_i^* is equal to the distribution of gene trees under the multi-species coalescent on $\mathcal{T}|Y_i$, by definition.

If T_i is generated first, then the pruning operations described above mean that T_i^* will lie entirely within the subtree $\mathcal{T}|Y_i$. It remains to show that the probability of any given pattern of coalescence on the remaining branches is identical to the MSC under $\mathcal{T}|Y_i$. This follows from the memoryless property of coalescence under the MSC: the probability of any two lineages originating within $\mathcal{T}|Y_i$ coalescing at any given point is not dependent on either lineage's coalescent history. □

It should be noted by this model description, taxa are absent or present independently of the generation of the gene data, including tree topology and sequence evolution, and the two processes are exchangeable. Also, as is the case with the general multi-species coalescent model, gene trees evolve under a process that is *i.i.d.* with respect to one another.

We will now define the two models for taxon deletion described briefly earlier.

The *i.i.d.* Model (M_{iid}) M_{iid} is a family of models parameterized by p, with $0 < p < 1$, where p is the probability that a random gene is present in a random species. For the M_{iid} model for parameter p, we assume that $Y_{ij} \sim Bernoulli(p)$ for all genes i and all taxa j, and that Y_{ij} and Y_{kj} are independent for $k \neq i$. (By extension of the statement earlier that genes evolve independently of one another, this also implies that Y_{ij} and Y_{ik} are independent for genes j, k where $j \neq k$.)

The Full Subset Coverage Model (M_{fsc}) M_{fsc} is a family of models parameterized by $k \geq 2$. We assume that the taxon deletion process is *i.i.d.* across the genes but we do not assume that it is *i.i.d.* across species. An M_{fsc} model for parameter k satisfies the property that for any subset B of at most k species there is a strictly positive probability p_B (that can depend on B) so that given a random gene, every member of B is present in the data for that gene with probability p_B. Since the number of taxon sets of size at most k is finite, $p^* = \min\{p_B : B \subseteq \mathcal{X}, |B| \leq k\} > 0$; hence, every taxon subset of size at most k appears in a random gene with probability at least p^*. Note that every M_{iid} model satisfies the property of being an M_{fsc} model for every k.

Comparison to previous models Most prior studies of the impact of missing data on phylogenomic analysis have been performed under the M_{iid} model; this model is referred to as **R** in [245] and as the "random allocation" model in [87]. [245] also considered the **G** model, where missing data are concentrated in a

subset of randomly chosen genes, and then ingroup taxa are deleted under an *i.i.d.* process from these genes. [245] also studied the **S** model, where missing data are allowed only in a subset of randomly chosen ingroup species, and that the genes are deleted from the selected species under an *i.i.d.* process. Note that the **S** and **G** models studied in [245] are M_{fsc} models.

3.3 Results

3.3.1 Results under an adversary model of taxon deletion

Theorem 3. *Let taxon deletion be dependent on gene tree topology. There exists a dependency structure under which no method is statistically consistent.*

Proof. Let $\mathcal{T}$ and $\mathcal{T}'$ be possible species trees; note that $\mathcal{T}'$ has a topology that appears with strictly positive probability under the MSC for species tree $\mathcal{T}$. For each gene g_i (whose true gene tree topology we'll denote as t_i for clarity), consider the dependency structure where all taxa are present in the data for g_i with probability 1 if the topology of t_i is identical to $\mathcal{T}'$, and all taxa are absent with probability 1 otherwise. Effectively, gene g_i is observed if and only if it has topology identical to $\mathcal{T}'$. Then the distribution of observed gene data is not unique to the species tree and identifiability of the species tree is lost, so no method can be statistically consistent. □

The theorem above demonstrates that a dependence between gene tree topology and taxon presence can quickly unravel statistical consistency guarantees in the absence of additional assumptions. But such a dependence may exist for some realistic models of gene presence/absence, including a birth/death type model where a gene may be present only for a clade of the tree. Such models are interesting and unsolved, but are beyond the scope of this chapter.

3.3.2 Results for Type 1 methods under M_{fsc}

We now discuss the statistical consistency guarantees of Type 1 tuple-based methods. As we will see, most of the tuple-based methods remain statistically consistent even in the presence of missing data, as long as the process that generates the missing taxa is well-behaved (e.g., not generated by an adversary that biases the method towards the wrong tree).

Let $A = (F, \alpha)$ be a Type 1 ℓ-tuple method that satisfies the following properties:

- (i) For all model species trees $\mathcal{T} = (T, \Theta)$, as the number m of genes increases, $F(I_m) \xrightarrow{p} F_0$, where F_0 is a constant vector parameterized by $\mathcal{T}$.
- (ii) There exists $\delta > 0$ such that for all vectors of summary statistics F_1 satisfying $L_\infty(F_1, F_0) < \delta$, $\alpha(F_1) = \alpha(F_0) = T$.

Theorem 4. *Let $A = (F, \alpha)$ be a Type 1 ℓ-tuple species tree estimation method satisfying the two properties (i) and (ii) above, and assume that the number of species is at least ℓ. The natural extension of A is statistically consistent under M_{fsc} with parameter $k \geq \ell$, and thus also under M_{iid} for any parameter p.*

Proof. Let $\mathcal{T} = (T, \Theta)$ be the model species tree, I_m be the input dataset containing m genes, C be the number of summary statistics computed by algorithm $A = (F, \alpha)$ on input I_m. Since $A = (F, \alpha)$ satisfies condition (i) when there are no missing data, then as the number of genes m increases, $F(I_m) \xrightarrow{p} F_0$, where F_0 is a vector of constants. We will denote the i^{th} summary statistic computed on input I_m by $F_i(I_m)$ and the i-th component of F_0 as F_{0_i}. We write $F(I_m) = \left(F_1(I_m|_{\mathbf{x}_1}), \ldots, F_C(I_m|_{\mathbf{x}_C})\right)$ where $\mathbf{x}_i$ denotes a particular set of ℓ taxa. In other words, since A satisfies condition (i) when there are no missing data, for all $i = 1, \ldots, C$ there exist a constant F_{0_i} such that $F_i(I_m|_{\mathbf{x}_i}) \xrightarrow{p} F_{0_i}$ as $m \to \infty$. Since the data for each gene are independent of all others, to prove statistical consistency under the M_{fsc} model we merely require that $I_m|_{\mathbf{x}_i}$ include an infinite number of genes as $m \to \infty$. Under the M_{fsc} model, $Pr[\mathbf{x}_i \subseteq L(g)] > 0$ for every gene g (where $\mathcal{L}(g)$ denotes the set of species for gene g). Hence, by the Borel-Cantelli lemma, the number of genes that include all ℓ taxa in $\mathbf{x}_i$ will also approach infinity. Thus $I_m|_{\mathbf{x}_i}$ will include an infinite number of genes, and $F(I_m|_{\mathbf{x}_i}) \xrightarrow{p} F_{0_i}$. By the definition of the natural extension of A, α does not change under deleted taxa. Since A satisfies condition (ii), $\exists \delta > 0$ such that $\forall F_1$ with $L_\infty(F_1, F_0) < \delta$, $\alpha(F_1) = T$, and so the natural extension of A is statistically consistent under M_{fsc}. Since M_{iid} is a subset of M_{fsc}, it is also statistically consistent under M_{iid}. □

Corollary 5. *ASTRAL and METAL are statistically consistent under the MSC even when taxa are deleted under an M_{fsc} model, provided that each is run in exact mode and so finds globally optimal species trees. MP-EST and STEM are statistically consistent under the MSC even when taxa are deleted under an M_{fsc} model, if sequence evolution is under a strict molecular clock and they find globally optimal species trees. SVDquartets is statistically consistent under the MSC even when taxa are deleted under an M_{fsc} model, if sequence evolution is under a strict molecular clock and the quartet amalgamation heuristic used is modified to ensure that it returns a compatibility tree when the input set of quartets is compatible. SMRT is statistically consistent under the MSC even when taxa are deleted under an M_{fsc} model, if sequence evolution is under the symmetric two-state model with a strict molecular clock.*

3.3.3 Statistical consistency of versions of ASTRAL under M_{fsc}

ASTRAL-1 [141] (and its improved versions, ASTRAL-II [143] and ASTRAL-III [251]) are coalescent-based methods for estimating species trees that take unrooted gene trees as input, and return a tree that minimizes the quartet tree distance to the input gene trees. Each can be run in exact mode, which guarantees that the tree that is returned has the minimum distance to the input gene trees. However, the exact versions are computationally intensive (running in time that grows exponentially with the number of species), and so heuristic versions are also available. These heuristic versions operate by constraining the search space using the input set of gene trees, and then guarantee that an optimal tree is returned within the search space. The important difference between the two methods is how the search space is constrained, and ASTRAL-2 explicitly enlarges the space compared to ASTRAL-1 when the input gene trees can be incomplete (i.e., when some species are missing from some gene trees). Because the search space is constrained using the input gene trees, the two ASTRAL algorithms depends on the input in a way that makes the analysis of their statistical guarantees non-trivial.

This section shows that both ASTRAL-1 and ASTRAL-2 are statistically consistent under the M_{iid} model, but not under any M_{fsc} model. We then present a modification to ASTRAL-1, denoted by ASTRAL*, and which differs from ASTRAL-1 only in how the search space is constrained. We then show that ASTRAL* is statistically consistent under many M_{fsc} models.

ASTRAL-1

We begin with a formal description of the ASTRAL-1 algorithm.

Notation. We let $\mathcal{X}$ denote the full set of species, and $\mathcal{X}'$ denote an arbitrary subset of $\mathcal{X}$. Every tree t we consider is assumed to be a binary unrooted tree with leaves taken from a subset of $\mathcal{X}$, and as earlier we denote the leafset of t by $\mathcal{L}(t)$. Each edge of t defines a **bipartition** of the set $\mathcal{L}(t)$ (denoted by $B|B'$, for some set $B \subseteq \mathcal{X}$ and $B' = \mathcal{L}(t) \setminus B$) obtained by deleting the edge but not its endpoints from t. We will refer to the set of all these bipartitions as $Bip(t)$, and the set of halves of the bipartitions of t as the **clades** of t. (Note that the term "clades" is normally used only in the context of rooted trees, but we extend the term here to allow us to refer to halves of bipartitions using the same term.) We let $T_{\mathcal{X}}(X)$ denote the set of unrooted binary trees on leafset $\mathcal{X}$ that satisfy $Bip(t) \subseteq X$. If X is not provided, then we assume the set of unrooted binary trees is not constrained, and let $T_{\mathcal{X}}$ denote the set of all unrooted binary trees on leafset $\mathcal{X}$.

We let $Q(t)$ denote all 4-leaf homeomorphic subtrees of t induced by a set of four leaves in t, and we note that when t is binary (i.e., fully resolved), then $Q(t)$ contains only binary quartet trees. Let Q be the set of all $\binom{n}{4}$ 4-taxon subsets of the taxon set $\mathcal{X}$. Let $q \in Q$, let t be an arbitrary tree topology on $\mathcal{X}$ and let $Top(q,t)$ denote the induced quartet subtree topology for quartet q in t.

Definition 6. ***ASTRAL Optimization Problem***

Input*: Taxon set $\mathcal{X} = \{x_i\}_{i=1}^n$, gene trees $t_1, \ldots, t_m$, and set X of allowed bipartitions of $\mathcal{X}$.*

Output*: Binary tree T where*

$$T = \arg \max_{t \in T_{\mathcal{X}}(X)} \sum_{q \in Q} \sum_{i=1}^{m} \mathbb{I}_{\{Top(q,t)=Top(q,g_i)\}}$$

ASTRAL-1 will specifically return the tree that maximizes the optimization criteria in the innermost summation subject to the constraint that every bipartition in the output tree be included in the set $\mathcal{X}$. Therefore, to show consistency under a model of missing data it is necessary to show not only that the optimization criteria still works, but also that the true topology will be still be included in this constrained search space.

ASTRAL-1 and ASTRAL-2 differ in how they define the default set X of allowed bipartitions, and also use slightly different dynamic programming techniques to assemble the optimal tree from the bottom up. Note that to run ASTRAL-1 or ASTRAL-2 in exact mode, the set X is defined to be all bipartitions on $\mathcal{X}$. In the default version of ASTRAL-1 (referred to as the "heuristic version"), X is the set of all bipartitions that appear in any gene tree. Hence, when there are no missing data, then as the number of genes increases, the set X will include all possible bipartitions on the taxon set with probability converging to 1 (and hence in particular the bipartitions in the true species tree). However, when there are missing data, then proving

that the set X contains all the bipartitions in the species tree takes some care. In particular, if every gene tree is incomplete, then no bipartition in any gene tree is a bipartition of the full set of taxa, and so this default setting will not enable a statistically consistent estimation method.

ASTRAL*: We will modify ASTRAL-1 by changing how it defines the set X of allowed bipartitions, and refer to this modification as ASTRAL*. Specifically, we will add bipartitions to the default setting computed by ASTRAL-1. Hence, ASTRAL*'s extra bipartitions could also be added to ASTRAL-2.

Note that ASTRAL-1, ASTRAL-2, and hence also ASTRAL, when run in heuristic mode, are different from the species tree estimation methods described previously, in that α depends not only on the summary statistics $F(I)$ but also on the input data I. Therefore, we denote the output of the function by $\alpha(F, I)$.

For every clade $C \subset \mathcal{X}$ occurring in a gene tree, we require that ASTRAL* adds the bipartition $C|C'$ where $C' = \mathcal{X} \setminus C$, to its set X. (Note that since the trees in this problem are unrooted, a clade and one half of a bipartition are equivalent concepts.) This is a trivial extension of the algorithm for a model of incomplete genes and one that strengthens the conditions under which the method is consistent, as we will see below.

Theorem 7. *(1) ASTRAL*, as well as ASTRAL-1 and ASTRAL-2 run in default heuristic mode, are statistically consistent under the MSC for any M_{iid} model of taxon deletion. (2) ASTRAL-1 is not statistically consistent under an M_{fsc} model with parameter k if the number of species is greater than k and ASTRAL-1 is run in default heuristic mode. (3) ASTRAL* is statistically consistent under any M_{fsc} model of taxon deletion with parameter k if the number n of species is at most* $2k$.

Proof. (1) Let $\mathcal{T}$ be a model species tree, and consider taxon deletion under some M_{iid} model. We will show that there is non-zero probability that every bipartition in the species tree appears in the search space computed by ASTRAL-1 in its default setting. Since the search space computed by ASTRAL-1 is a subset of the search space computed by ASTRAL-2 and ASTRAL*, the result will follow. Recall that ASTRAL-1 includes all bipartitions $C|C'$ that appear in any input gene tree. Under the MSC model, every bipartition appears in some gene tree with probability increasing to 1 as the number of genes increases. Under any M_{iid} model, for every subset of taxa, the probability that none of the taxa in the subset are deleted is strictly greater than 0. Hence, under M_{iid}, the set X of bipartitions allowed in the ASTRAL-1 search space will converge to the set of all possible bipartitions. Therefore, ASTRAL-1 is statistically consistent under M_{iid}. Since the sets of bipartitions computed by ASTRAL* and ASTRAL-2 contain the set of bipartitions computed by ASTRAL-1, ASTRAL-2 and ASTRAL* are also statistically consistent under M_{iid}.

(2) Now consider the M_{fsc} model with parameter k: let $n > k$, and further let the taxon deletion process be such that every gene has exactly k taxa, (e.g. k taxa sampled uniformly, a valid model under M_{fsc}). Then if ASTRAL-1 is run in heuristic mode, it will compute a set X that contains no bipartitions on the set $\mathcal{X}$ of species, and so cannot be statistically consistent under all M_{fsc} models.

(3) We now show that ASTRAL* run in heuristic mode is statistically consistent under any M_{fsc} model with parameter k when $n \leq 2k$. Let $C|C'$ be an arbitrary bipartition on $\mathcal{X}$, and assume without loss of generality that $|C| \leq k$. Hence, under M_{iid}, the probability that C appears in a random gene tree is strictly positive. Under the MSC, any bipartition on $\mathcal{X}$ appears in a given true gene tree with strictly positive prob-

ability. Since this process is independent from the removal of taxa, and since there is non-zero probability that all members of a clade appear in the gene tree, the probability is non-zero that the set C appears as a clade in a random gene tree.

Hence, as the number m of gene trees increases, the probability approaches 1 that C appears as a clade in at least one gene tree. Thus the probability approaches 1 that the set X computed by ASTRAL* will contain $C|C'$, where $C' = \mathcal{X} \setminus C$. Therefore, ASTRAL*, run in heuristic mode, will be statistically consistent under the M_{fsc} model with parameter k, provided that the number of species $n \leq 2k$.

□

Theorem 8. *ASTRAL-1 and ASTRAL*, when run in heuristic mode, are not statistically consistent under the MSC_{fsc} class of models with parameter k, if the number of species $n > 2k$.*

Proof. Consider a model of taxon deletion where every gene tree has exactly k taxa, selected at random from the full set of taxa. This model satisfies the conditions of the M_{fsc} models with parameter k. Now assume $k < \lfloor n/2 \rfloor$.

Let $\mathcal{T}$ be a caterpillar tree on a set $\mathcal{X}$ of n taxa. Then $\mathcal{T}$ contains a clade B of size $\lfloor n/2 \rfloor$ whose complement is at least as large; hence both B and $\mathcal{X} \setminus B$ have more than k species. Hence, under this model of taxon deletion, neither B nor $\mathcal{X} \setminus B$ will be in any gene tree. Hence, the bipartition $B|B'$ (where $B' = \mathcal{X} \setminus B$) will not be in X (the constraint on the search space) as computed by ASTRAL and ASTRAL*. Hence, neither ASTRAL nor ASTRAL* can recover the true species tree under this random taxon deletion model. □

3.3.4 Statistical consistency of ASTRID and NJst under M_{iid}

As noted earlier, ASTRID, NJst, and STAR are Type 2 methods, and proofs we provided of statistical consistency for Type 1 tuple-based methods do not apply to these methods (or other Type 2 methods). In this section we will show ASTRID and NJst remain statistically consistent under the M_{iid} models of taxon deletion. However, the statistical consistency of these methods under the more general M_{fsc} models is unknown.

NJst is a distance-based method that uses the average topological "internode" distance between taxa in the gene trees. The internode distance between two taxa x_i and x_j in a tree is the count of individual nodes along the path from x_i to x_j, denoted $\rho(x_i, x_j)$. ASTRID is an extension of NJst with the averaging redefined to better accommodate missing taxa and is precisely the natural extension of NJst as defined earlier, where the statistic for each pair is calculated as usual but restricted to the genes in which both members of the pair appear. However, NJst and ASTRID are not Type 1 methods, under the definition provided above, because the internode distance for two taxa x_i and x_j can be affected by the presence or absence of a third taxon.

Distance methods are formally 2-tuple methods, and use an algorithm such as neighbor joining [196] to return a tree topology and branch lengths given $\binom{n}{2}$ pairwise distances (collectively, the distance "matrix"). We now state some well-known properties of such methods for reference in the proof below. For a tree with topology $T = (V, E)$ on n taxa and edge weights l_e, $e \in E$, if the distance for any two taxa j and k is equal to the sum of the edge weights over edges in the shortest path between leaves j and k, then neighbor-joining will return a tree with topology T, and the distance matrix is said to be **additive** on the topology T. An equivalent definition of an additive matrix is as follows:

Definition 9. ***The Four Point Condition***. *Let $T = (V, E)$ be a tree on n leaves labeled as $S = \{s_i\}_{i=1}^n$ with positive edge weights l_e, $e \in E$. Let $D = d_{ij}$ be the matrix of pairwise distances between all pairs of taxa (i, j), $i, j \in S$. The matrix D is additive on the topology T if and only if for all sets of four (not necessarily distinct) leaves $\{i, j, k, l\} \subset S$ with quartet-subtree topology $ij|kl$ in T, without loss of generality, the following holds:*

$$d_{ij} + d_{kl} < d_{ik} + d_{jl} = d_{il} + d_{jk}.$$

Furthermore, if instead of D as above we are given $\hat{D} = d_{ij} + \varepsilon_{ij}$, where ε_{ij} is an unknown noise term such that for all $i, j \in S$, $|\varepsilon_{ij}| < \frac{1}{2} \min_{a,b \in S} d_{ab}$, then neighbor-joining applied to the matrix $\hat{D}$ will also return the topology T with probability 1. Therefore, since NJst is a distance-method, to prove statistical consistency it suffices to show that the metric given by the average internode distance collectively converges to an additive matrix on the true topology.

Theorem 12. *Assume that taxa are absent from the data for each gene according to the $\mathbf{M}_{iid}$ model. Then NJst and ASTRID are statistically consistent under the MSC.*

First we give a helpful lemma in two parts. Note that since $\rho(x_i, x_j)$ is undefined when either of x_i or x_j are removed, the expectation of $\rho(x_i, x_j)$ is formally undefined as long as the probability of either is nonzero. We nonetheless use the notation $\mathbf{E}\left[\rho(x_i, x_j)\right]$ in the lemma and proofs below, which will refer implicitly to the conditional expectation on the event that neither x_i nor x_j are removed.

Lemma 13. *Under the MSC and $\mathbf{M}_{iid}$, let a, b, c and d be four taxa, and consider the event in the coalescent probability space in which the lineages of these taxa have entered a common population and no pair have coalesced with one another, denoted as $\mathcal{E}'$. Denote the points on each respective lineage in which they enter the common population as A, B, C, and D. Let Y be the random variable representing the taxon deletion process. Then for any two taxa $\{i, j\} \subset \{a, b, c, d\}$ and respectively $\{I, J\} \subset \{A, B, C, D\}$:*

$$\mathbf{E}\left[\rho(i, j)|\mathcal{E}'\right] = \mathbf{E}_Y\left[\rho(i, I)|\mathcal{E}'\right] + \mathbf{E}_Y\left[\rho(j, J)|\mathcal{E}'\right] + K$$

where K is a constant that does not depend on the identities of i and j.

Proof of Lemma 13. Consider any gene tree g meeting the condition of event $\mathcal{E}'$, and let G be the subtree of g sitting between the points A, B, C and D and the root. The topology of G in this model, given that $g \in \mathcal{E}$, is determined by a set of *i.i.d.* Exponentially distributed random variables corresponding to the pairwise times-to-coalescence of all remaining lineages concurrent with and including $\{A, B, C, D\}$. By De Finetti's theorem, the probability density function of G is unique up to a permutation of the indices of the random variables. Thus for any $\{I, J\} \subset \{A, B, C, D\}$, since $\rho(I, J)$ is dependent only on the topology of G, the probability density of $\rho(I, J)$ does not depend on the identity of I and J, and thus $\mathbf{E}\left[\rho(I, J)|\mathcal{E}'\right] = K$. □

Lemma 14. *Let the conditions of Lemma 13 in the manuscript hold. Let the species tree $\mathcal{T}$ display an unbalanced topology, without loss of generality labeled $(((a, b), c), d)$, and let $\mathcal{E}''$ denote the event in coalescent probability space that a, b, and c have entered a common population and no two have coalesced with one*

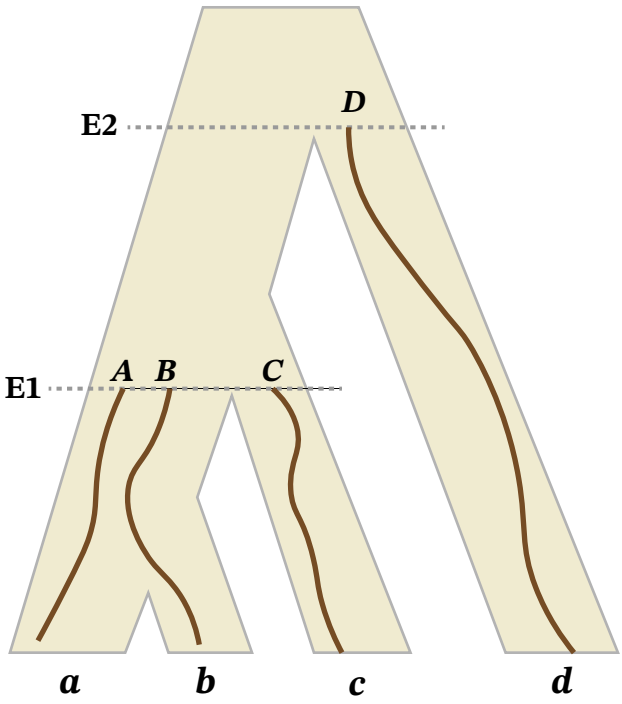


Figure 3.1: The species tree $\mathcal{T}$, having unbalanced topology and showing the conditions that constitute event $\mathcal{E}''$.

another. Let A, B and C denote the points where the lineages of a, b and c enter a common population, and let D denote the point where the lineage of d joins them. (See Figure 3.1.)

Then for any two distinct taxa $\{i, j\} \subset \{a, b, c, d\}$ *and respectively* $\{I, J\} \subset \{A, B, C, D\}$*:*

$$\mathbf{E}\left[\rho(i,j)|\mathcal{E}''\right] = \mathbf{E}_Y\left[\rho(i,I)|\mathcal{E}''\right] + \mathbf{E}_Y\left[\rho(j,J)|\mathcal{E}''\right] + H$$

where H *is a constant that does not depend on the identities of* i *and* j*.*

Additionally, the matrix $\mathbf{R}_{ij} = \left[\mathbf{E}\left[\rho(i,j)|\mathcal{E}''\right]\right]$ *is additive.*

Proof of Lemma 14. As with the proof of Lemma 13, since A, B and C are un-coalesced and in the same population, they are exchangeable. This gives us that:

$$\mathbf{E}\left[\rho(A,B)|\mathcal{E}''\right] = \mathbf{E}\left[\rho(A,C)|\mathcal{E}''\right] = \mathbf{E}\left[\rho(B,C)|\mathcal{E}''\right] = K$$

Additionally, it implies that:

$$\mathbf{E}\left[\rho(A,D)|\mathcal{E}''\right] = \mathbf{E}\left[\rho(B,D)|\mathcal{E}''\right] = \mathbf{E}\left[\rho(C,D)|\mathcal{E}''\right] = J$$

Together, those imply the first statement, where $H = K + J$.

Since the first statement applies only to distinct pairs of indices, showing that the Four Point condition

applies to the matrix $\mathbf{R}$ requires additionally verifying sets of non-distinct indices. In practice this only means checking sets with duplicates since triplicates and beyond are trivial. Exchangeability of A, B and C saves us time here as well. The first two cases are simple:
AB|CC:

$$\begin{aligned}\mathbf{E}\left[\rho(A,B)|\mathcal{E}''\right] + \mathbf{E}\left[\rho(C,C)|\mathcal{E}''\right] &= K \\ \mathbf{E}\left[\rho(A,C)|\mathcal{E}''\right] + \mathbf{E}\left[\rho(B,C)|\mathcal{E}''\right] &= 2K\end{aligned}$$

AD|CC:

$$\begin{aligned}\mathbf{E}\left[\rho(A,D)|\mathcal{E}''\right] + \mathbf{E}\left[\rho(C,C)|\mathcal{E}''\right] &= K \\ \mathbf{E}\left[\rho(A,C)|\mathcal{E}''\right] + \mathbf{E}\left[\rho(D,C)|\mathcal{E}''\right] &= K + J\end{aligned}$$

The final set will take a bit more work:
AB|DD:

$$\begin{aligned}\mathbf{E}\left[\rho(A,B)|\mathcal{E}''\right] + \mathbf{E}\left[\rho(D,D)|\mathcal{E}''\right] &= K \\ \mathbf{E}\left[\rho(A,D)|\mathcal{E}''\right] + \mathbf{E}\left[\rho(B,D)|\mathcal{E}''\right] &= 2J\end{aligned} \tag{3.3.1}$$

But it is not clear that $K < 2J$, so we must show that. We will in fact show that $K < J$, which is sufficient. To do this, we will partition event $\mathcal{E}''$ into a few cases and show that for each case i:

$$\mathbf{E}\left[\rho(A,B)|\mathcal{E}'', \text{Case } i\right] \leq \mathbf{E}\left[\rho(A,D)|\mathcal{E}'', \text{Case } i\right] \tag{3.3.2}$$

Figure 3.1 shows an illustration of the species tree $\mathcal{T}$ and the conditions that define the event $\mathcal{E}''$. The point where a, b and c join a common population is denoted as **E1**, and the point where those three join a common population with d as **E2**.

The cases that we will consider will be based on how many coalescent events (CEs) happen between the three lineages in the portion of the species tree between **E1** and **E2**. The possibilities are 0, 1 and 2. Crucially, the relative probability that the gene tree would take on Case 0/1/2 depends on the shape of the species tree and is difficult to compare arbitrarily. But by limiting the comparison to one case at a time, if (3.3.4) holds conditionally for every case then it also holds unconditionally, so relative probability is not an issue.

Case 1: 0 CEs This is the simplest case because it means all four lineages enter a common topology uncoalesced at the root **E2**. In this case the four lineages become exchangeable and (3.3.4) holds with strict equality.

Case 2: 2 CEs This is the next simplest. There are three possible ways for 2 CEs to occur in the E1-E2 region and each one fully determines the gene tree topology. The three possibilities are shown in Figure 3.2a along with the resultant topology. Once again by exchangeability we know that all three possibilities are equally likely, so we can compute the conditional expectation directly, shown in Table 3.1.

	$\rho(A, B)$	$\rho(A, D)$
(1)	2	4
(2)	3	4
(3)	3	3
avg.	**2.67**	**3.67**

Table 3.1: Internode distances from Figure 3.2a.

Thus (3.3.4) holds for Case 2.

Case 3: 1 CE This case is slightly more complicated just by the number of possible topologies, which are illustrated in Figure 3.2b. The three equally likely CE combinations are shown and labeled (4)-(6). In each case, there are three ways that the exiting lineage can coalesce with D to form the rooted gene tree (also equally likely). Those are labeled (a)-(c). Since that is the full range of possibilities for Case 3, we can compute the conditional means directly by computing the internode distances and the averages. Those values are given in Table 3.2.

		$\rho(A, B)$	$\rho(A, D)$
	(a)	2	4
(1)	(b)	2	4
	(c)	2	3
	(a)	3	4
(2)	(b)	4	4
	(c)	4	3
	(a)	3	3
(3)	(b)	4	2
	(c)	4	3
avg.		**3.11**	**3.33**

Table 3.2: Internode distances from Figure 3.2b.

Thus (3.3.4) holds for Case 3 as well, and in both of the last two cases the strict inequality has held. So we have that $K < J$, and the conditions in (3.3.4) check out.

Therefore, the four-point condition holds for the matrix $\mathbf{R}_{ij}$ and it is additive. □

Proof of Theorem 12. In order to show that the method is statistically consistent, we must show the following:

1. For a gene tree generated under the coalescent process in the MSC followed by the removal of taxa subject to M_{iid}, the expected value of the summary statistics $\rho(x_i, x_j)$ for each pair of taxa x_i, x_j form an additive matrix that defines the topology of the species tree.

2. The statistics themselves converge to their expectations as the number m of genes approaches infinity.

The second property follows from the *i.i.d.*-generation of gene trees and the weak law of large numbers; hence, we need only establish the first property.

Let G be a random gene tree on n taxa generated by the MSC on species tree $\mathcal{T}$, and let $Y_n \sim M_{iid}$ with probability of deletion p. We will show that for an arbitrary set of four taxa $\{x_1, x_2, x_a, x_b\}$, the expectations over G and Y_n of the internode distances are additive on the species tree, which we check by confirming that they obey the four point condition for the species tree $\mathcal{T}$. In what follows, we will refer to this by saying that the expectations "obey the four point condition", with the understanding that this is with reference to the species tree $\mathcal{T}$. Importantly, the event $\mathcal{E} = \mathcal{E}' \cup \mathcal{E}''$ defined in Lemmas 13 and 14 includes all cases in which the quartet-subtree topology in the gene tree does not match that of the species tree, and implies that the four point condition holds in these cases. As a result, it suffices to show that the four point condition holds when the quartet-subtree in G is identical to the topology in the species tree, which we assume without loss of generality has topology $x_1x_2|x_ax_b$. We will show this by induction on the number n of taxa, and begin with the smallest non-trivial case, $n = 4$.

Base Step: For $n = 4$ we can write the expected values of the internode distances in closed form and check the four point condition directly:

$$\begin{aligned}
\mathbf{E}\left[\rho(x_1, x_2)\right] &= 2 - p^2 \\
\mathbf{E}\left[\rho(x_a, x_b)\right] &= 2 - p^2 \\
\mathbf{E}\left[\rho(x_1, x_a)\right] &= 3(1-p)(1-p) + 2(2p(1-p)) + 1(p^2) \\
&= 3(1 - 2p + p^2) + 4(p - p^2) + p^2 \\
&= 3 - 6p + 3p^2 + 4p - 4p^2 + p^2 \\
&= 3 - 2p \\
\mathbf{E}\left[\rho(x_2, x_b)\right] &= 3 - 2p
\end{aligned}$$

Thus to test that the four point condition holds, we have:

$$\begin{aligned}
\mathbf{E}\left[\rho(x_1, x_2)\right] + \mathbf{E}\left[\rho(x_a, x_b)\right] &= 2 - p^2 + 2 - p^2 \\
&= 4 - 2p^2 \\
\mathbf{E}\left[\rho(x_1, x_a)\right] + \mathbf{E}\left[\rho(x_2, x_b)\right] &= 3 - 2p + 3 - 2p \\
&= 6 - 4p \\
\mathbf{E}\left[\rho(x_2, x_a)\right] + \mathbf{E}\left[\rho(x_1, x_b)\right] &= 6 - 4p
\end{aligned}$$

It remains to show that $6 - 4p > 4 - 2p^2$. Let:

$$\begin{aligned}
f(p) &= (6 - 4p) - (4 - 2p^2) \\
&= 2p^2 - 4p + 2 \\
&= 2(1-p)^2
\end{aligned}$$

Since $f(p)$ is the product of positive terms, $f(p) > 0$ for $0 \leq p \leq 1$. Thus $6 - 4p > 4 - 2p^2$ and the four point condition holds for $n = 4$.

Induction Step: Assume that for a set S_n of n taxa, the expected value of the matrix $D = [\rho(x_i, x_j)]$ for a random gene tree G_n and taxon-removal variable Y_n is additive on the true species tree. That is, for any set of n-taxa under the MSC and M_{iid} and any quartet (a, b, c, d), the four point condition holds for $\mathbf{E}_n[\rho(i, j)]$, $i, j \in \{a, b, c, d\}$, a fact that we will use below. Here $\mathbf{E}_n$ denotes the expectation operator under n taxa for notational clarity. We will now show that the same holds for a set of size $n + 1$.

Let x_{n+1} be a new taxon and let G be generated under the MSC on $S_n \cup \{x_{n+1}\}$ with taxon-removal variable $Y_{n+1} = (Y_n, y_{n+1})$ where $y_{n+1} \sim Bernoulli(p)$ in accordance with M_{iid}. Our approach will be to define three cases, each of which have non-zero probability, and show that regardless of the placement of x_{n+1} in the species tree, the four point condition holds on $\{x_1, x_2, x_a, x_b\}$.

Case 1: x_{n+1} is deleted from G (i.e. $y_{n+1} = 1$). In this case the conditional values of $\mathbf{E}_{n+1}[\rho(x, x')]$ for $x \neq x'$, $x, x' \in \{x_1, x_2, x_a, x_b\}$ are identical to the n-taxa case, and thus by our assumption they obey the four point condition.

Case 2: x_{n+1} is not deleted and it coalesces with G (*after* taxon-deletion has been applied) on a branch that is *not* on the induced quartet-subtree of G for the quartet $\{x_1, x_2, x_a, x_b\}$. In this case, the coalescence event for x_{n+1} does not add to the internode distances along any of the branches connecting any two members of the quartet. As a result, again the conditional values of $\mathbf{E}[\rho(x, x')]$ for $x \neq x'$, $x, x' \in \{x_i, x_j, x_k, x_l\}$ are identical to the n-taxa case, where again they obey the four point condition, by the induction assumption.

Case 3: x_{n+1} is not deleted and coalesces directly with a branch on the induced quartet subtree of G for the quartet $\{x_1, x_2, x_a, x_b\}$. This case is non-trivial and requires some analysis. For shorthand, we will refer to the quartet-subtree of G restricted to $\{x_1, x_2, x_a, x_b\}$ as simply q. For a pair of taxa $i, j \in S_n$ the value of the expected internode distance in the presence of x_{n+1} is:

$$\mathbf{E}_{n+1}[\rho(i, j)] = \mathbf{E}_n[\rho(i, j)] + P(i, j) \tag{3.3.3}$$

where $P(i, j)$ denotes the probability that x_{n+1} coalesces on a branch in the path from i to j, which would cause $\rho(i, j)$ to increase by 1. This quantity is non-trivial and depends jointly on both the topology of G *and* the value of Y_n. However, since the first term obeys the four point condition by the induction hypothesis, the proof depends on showing that $P(i, j)$ does as well.

To do that, we will first partition the joint probability space of the MSC and M_{iid}, denoted as $\mathcal{G}$ and assign each part of this partition to one of the five branches in the subtree q, then show that for $\{i, j\} \subset \{x_1, x_2, x_a, x_b\}$, $P(i, j)$ can be expressed as the sum of probabilities assigned to the branches between i and j. We will label the branches as b_1, b_2, b_a, and b_b for the four outer branches and b_m for the middle branch.

Figure 3.3 describes the partition based on the branch with which x_{n+1} coalesces (in each column) and the outcome of the taxon-removal process (in each row). For the illustrations in the header of each row, a branch represented with a dotted line represents the case that **all** lineages coalescing along that branch other than x_{n+1} are fully removed by the taxon-removal process. The positions of the relative taxa are given in the illustration in the first row header, and all taxon-removal outcomes that allow at least one $\rho(i, j)$,

$i, j \in \{x_1, x_2, x_a, x_b\}$ to be measured are represented in the rows. The assignment is not unique, as implied by the entries with offer two possibilities, but for the proof either will work so we consider the default to be the first branch listed.

For an event $(G, Y_{n+1}) \in \mathcal{G}$, denote the assignment of (G, Y_{n+1}) to branch b_e as $(G, Y_{n+1}) \to b_e$, and for that branch let

$$p_e = P\left(\{(G, Y_{n+1}) \in \mathcal{G} | (G, Y_{n+1}) \to b_e\}\right).$$

In this way, for any pair of taxa $i, j \in \{x_1, x_2, x_a, x_b\}$, the probability that $\rho(i, j)$ is incremented by 1 in the presence of x_{n+1} is given precisely by the sum of p_e for set of edges e between i and j in the quartet subtree q. This should be apparent by visual inspection of the table. Thus, $P(i, j)$ in (3.3.3) is the sum of positive edge weights on the topology $x_1x_2|x_ax_b$, which is also the species tree topology on these four taxa, and so is additive on the species tree. Hence, it meets the four point condition, completing the proof. □

This proof has been limited to ASTRID, and only provided under the M_{iid} model of taxon deletion because the independence of taxon deletion (between taxa as well as from gene tree generation) was used when we noted that

$$\mathbf{E}_{n+1}[\rho(x, x')] = \mathbf{E}_n[\rho(x, x')] + p_{xx'} \tag{3.3.4}$$

in case 3. Independence implies that the marginal probability distribution of $\rho(x, x')$ for n taxa is identical to the conditional distribution given the information that x_{n+1} has not been deleted.

While that is not strictly necessary for (3.3.4) to hold, counterexamples can be difficult to construct, and thus it is non-trivial to characterize conditions that may be weaker than pure independence and still imply that ASTRID is statistically consistent. It is an open question whether ASTRID is statistically consistent under any more general model (e.g., under the M_{fsc} model). It is also an open question whether other Type II methods (e.g., STAR) are statistically consistent under the M_{iid} model of taxon deletion.

Results shown here have focused on whether coalescent-based species tree estimation methods remain statistically consistent in the presence of missing data. However, inherent in the question is the assumption that the method being considered is statistically consistent when there are no missing data. Here we consider the conditions under which the different methods we discussed in this chapter are statistically consistent when there are no missing data.

3.4 Discussion

This study examined the theoretical properties of coalescent-based species tree estimation methods, and showed that the Type 1 methods are statistically consistent under a fairly general model of missing data (the "full subset coverage" model, M_{fsc}), while at least one Type 2 method is statistically consistent under an *i.i.d.* model of missing data that is a subclass of the M_{fsc} model. However, these statistical consistency results do not suggest that missing data do not have a negative impact on the accuracy of coalescent-based species tree estimation, nor that the impact may differ between methods. The practical implications of missing data

are best understood through experimental studies.

Four such studies [87, 227, 245, 209] examined the impact of missing data on coalescent-based species tree estimation methods. Hovmöller et al. [87] evaluated the impact of missing data on STEM and *BEAST, using two different models of taxon deletion and found that both STEM and *BEAST were highly robust to missing data.

Xi *et al.* [245] evaluated ASTRAL (v. 4.7.1), MP-EST, and STAR on simulated and biological datasets with missing data. They explored two taxon deletion models, **S** (missing data restricted to a subset of selected species) and **G** (missing data restricted to a subset of selected genes), although they never deleted any genes from the outgroup taxa. They found that the impact of missing data depended on the model and the amount of missing data, and that in general ASTRAL and MP-EST were highly robust to missing data, while STAR was much more negatively impacted.

Streicher et al. [209] examined the impact of missing data on the branch support of species trees estimated using NJst and ASTRAL (v. 4.7.6) on a biological dataset of Iguanian lizards. They explored different amounts of missing data, and concluded that in general it is best to include all the data, except (perhaps) when the amount of missing data in the gene or species exceeds 50%.

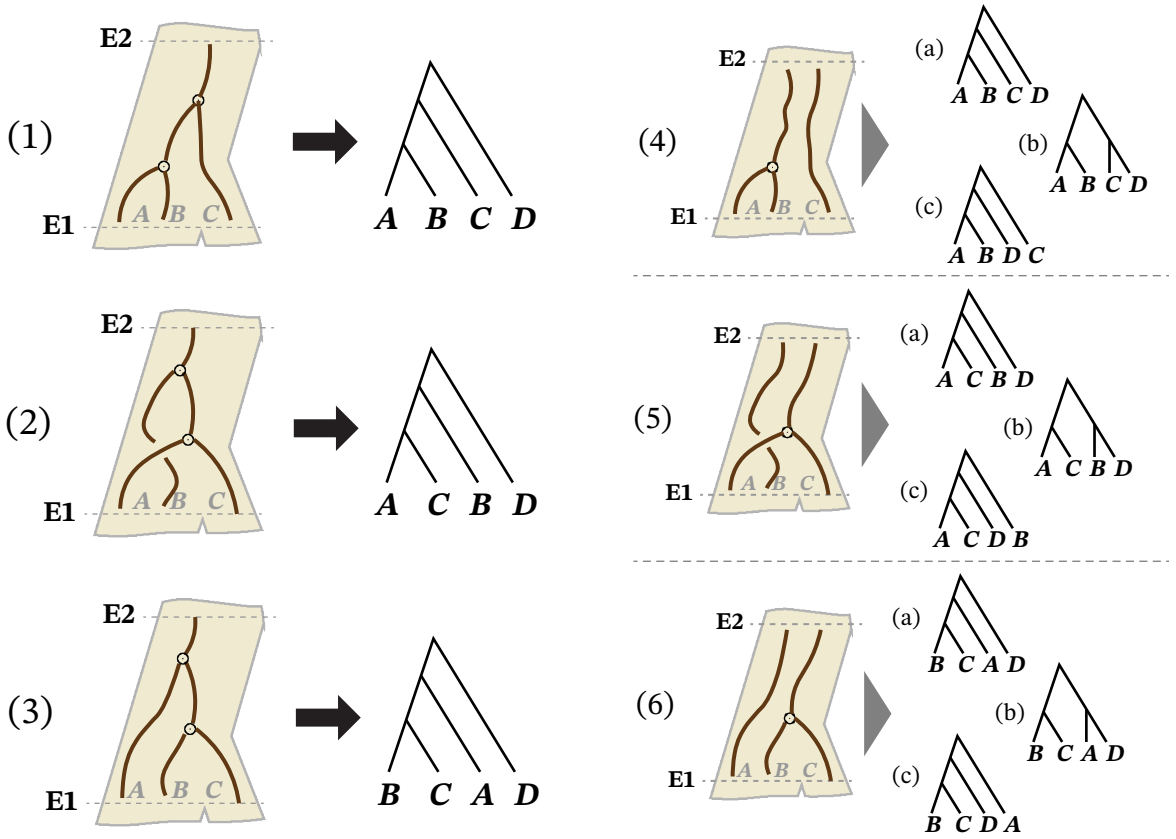
Finally, Vachaspati and Warnow [227] studied the impact of missing data on ASTRID and ASTRAL-2 (v. 4.7.8) on simulated datasets with 50 species under a model in which all genes had missing data, but the genes that were deleted from each species were selected under an *i.i.d.* model. They observed that both ASTRID and ASTRAL were substantially impacted by missing data, so that error increased with taxon deletion. However, even with very high missing data rates (i.e., all genes missing 80% of the species), both ASTRID and ASTRAL-2 achieved good accuracy with a large enough number of genes.

Overall these studies suggest that missing data is not especially detrimental to accuracy using MP-EST, ASTRID, ASTRAL, STEM, and *BEAST.

However, one particularly challenge of missing data is the possibility that outgroup taxa may be the missing taxa. When the phylogenomic estimation method requires rooted gene trees, this creates a challenge of using a different rooting method (e.g., midpoint rooting, or maximum likelihood under a strict molecular clock assumption). These rooting techniques are not statistically consistent when the data does not follow a strict clock, however. Only two of the studies above [87, 245] considered methods that require rooted gene trees, but both evolved sequences under a strict molecular clock. Xi, et al. [245] used outgroup rooting but did not simulate under conditions where data was missing for the outgroup. Thus our understanding of the empirical impact of missing data on methods that require rooted gene trees would benefit from additional study.

3.5 Summary

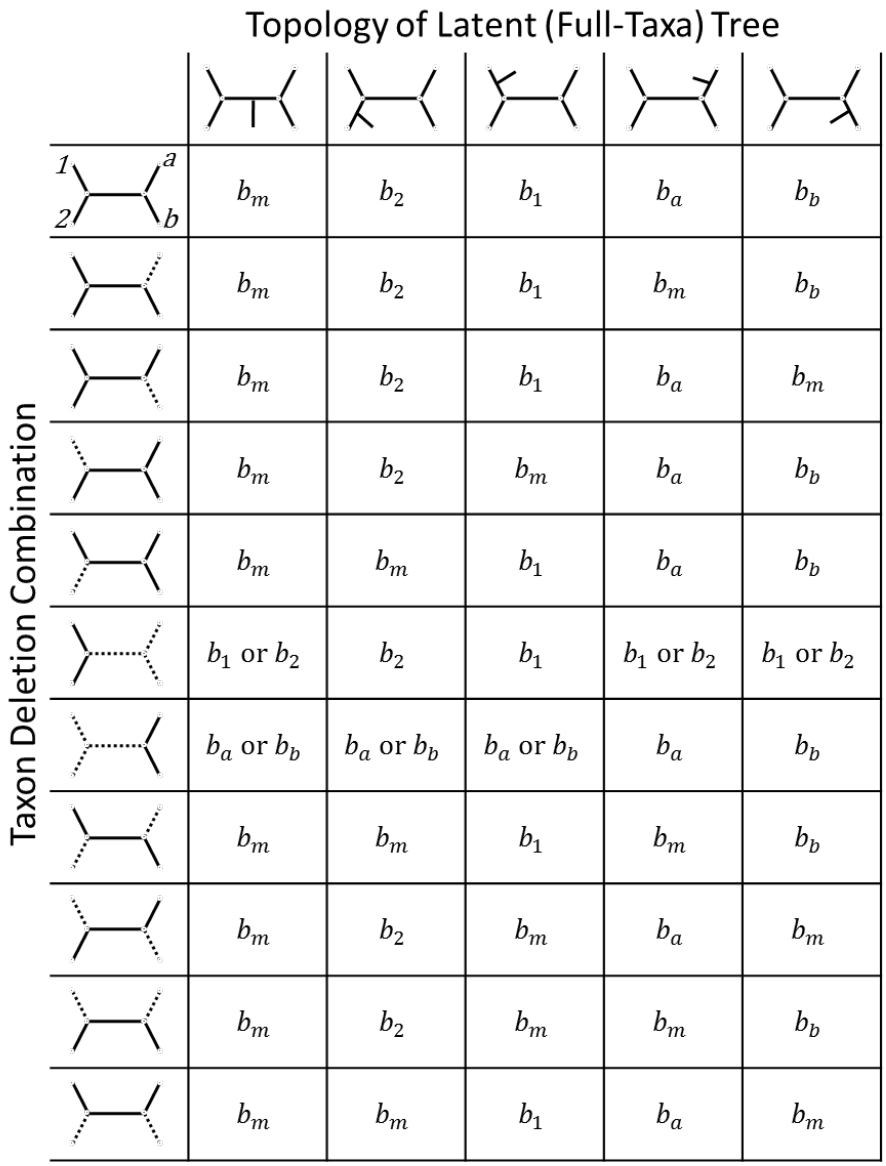
We have shown that if taxon absence/presence is independent of gene tree topology, then it is exchangeable with the MSC. We have further shown that full subset coverage is sufficient for a tuple-based method to be statistically consistent, and that *i.i.d.* taxon deletion is a necessary condition for NJst and ASTRID to be statistically consistent.



(a) Case 2: two coalescent events in the **E1**-**E2** region. Each combination uniquely determines the gene tree topology.

(b) Case 3: one coalescent event in the **E1**-**E2** region. Each combination entails three equally likely gene tree topologies.

Figure 3.2: Illustration of cases 2 and 3 as defined in the proof of Lemma 14.



Taxon Deletion Combination	Topology of Latent (Full-Taxa) Tree				
1, 2	b_m	b_2	b_1	b_a	b_b
	b_m	b_2	b_1	b_m	b_b
	b_m	b_2	b_1	b_a	b_m
	b_m	b_2	b_m	b_a	b_b
	b_m	b_m	b_1	b_a	b_b
	b_1 or b_2	b_2	b_1	b_1 or b_2	b_1 or b_2
	b_a or b_b	b_a or b_b	b_a or b_b	b_a	b_b
	b_m	b_m	b_1	b_m	b_b
	b_m	b_2	b_m	b_a	b_m
	b_m	b_2	b_m	b_m	b_b
	b_m	b_m	b_1	b_a	b_m

Figure 3.3: Table of attachment branches given the latent 5-taxon topology and set of deleted taxa from q, assuming no other taxa coalesce closer to the inner nodes. Branches (in rows) given by dotted lines are pruned as a result of taxon deletion.

Chapter 4

Inconsistency of Partitioned Likelihood and Topology-Based Summary Methods

4.1 Introduction[1]

Species trees are a key aspect of much biological research, including the detection of co-evolution, the inference of ancestral traits, and the dating of speciation events [178]. The availability of sequence data collected from diverse species representing a broad spectrum of life has led to the expectation that the construction of a robust Tree of Life should be possible using statistical estimation methods, such as maximum likelihood. These estimations are increasingly based on large numbers of loci (sometimes thousands) selected from across the genomes of different species [134, 92, 144, 236, 34, 129].

Many methods used for species tree estimation, however, have been designed for gene tree estimation, which is a simpler statistical estimation problem. For gene tree estimation, the assumption is that the input sequences have all evolved down a single model tree (called the "gene tree") under a sequence evolution model, such as Cavender-Farris-Neyman [36, 60, 154], Jukes-Cantor [95], or the Generalised Time Reversible (GTR) model [216]. The estimation of the gene tree under these models from the aligned sequence data is a well-studied problem, and many statistically consistent methods have been developed under these models [199]. Species tree estimation is much more complex, since gene trees can differ from the species tree due to multiple causes, including incomplete lineage sorting (ILS), as modeled by the multi-species coalescent (MSC) model [130]. Indeed, many recent phylogenetic analyses of genome-scale biological datasets for birds [92], land plants [236], worms [34], and other organisms, have revealed substantial heterogeneity across the genes that is consistent with ILS.

The construction of the species tree when there is gene tree heterogeneity due to ILS can be seen as a statistical estimation problem under a two-phase model of sequence evolution where gene trees evolve within a species tree under the MSC model, and then gene sequences evolve down each gene tree under a sequence evolution model. For example, under the MSC+JC model where true gene trees evolve within the species tree under the MSC model and gene sequences evolve down the gene trees under the Jukes-Cantor (JC) model, the estimation of species trees from gene sequence data needs to use the properties of the evolutionary models in order to be statistically consistent. One such approach for species tree estimation is to estimate gene trees for each locus, and then combine these gene trees into a species tree using a coalescent-based summary method (that takes gene tree incongruence due to ILS into account); such approaches can be

[1]This chapter contains material previously published in [191]. The proof of inconsistency for fully-partitioned Maximum Likelihood estimation was developed in collaboration with Sebastian Roch, and the majority of the writing was done by Sebastian Roch and Tandy Warnow. Figures were drawn by Sebastian Roch. The copyright owner has provided permission to reprint.

proven to converge in probability to the true species tree as the number of genes and number of sites per gene both increase. Thus, for example, statistically consistent species tree estimation under these conditions is possible under the MSC+JC model when gene trees are estimated using Jukes-Cantor maximum likelihood and then combined into a species tree using an appropriate coalescent-based summary method. Examples of these summary methods that enable statistically consistent species tree estimation include MP-EST [120], NJst [119], ASTRID [227], ASTRAL [141, 143], STEM [104], STEAC [121], STAR [121], and GLASS [149].

In contrast, many species trees are estimated using "unpartitioned maximum likelihood", where the gene sequence alignments are concatenated into a single supermatrix, and a tree is then estimated on that supermatrix under the assumption that all the sites evolve under the same model tree. As shown by [192], this approach is not statistically consistent and can even be positively misleading in the presence of gene tree heterogeneity due to ILS.

Although unpartitioned concatenated analysis with maximum likelihood (CA-ML) is known to be statistically inconsistent and coalescent-based species tree methods can be statistically consistent, performance in practice (and in particular on simulated datasets) has been mixed, with CA-ML sometimes more accurate than leading summary methods [109, 172, 138, 15, 41, 146]. One of the challenges to using summary methods is gene tree estimation error, resulting in part from limited sequence lengths per gene [16]. The "statistical binning" approach [138] was designed to improve the accuracy of species trees estimated using summary methods by binning sequences from different genes together using statistical techniques for detecting strongly supported incongruence (e.g., using bootstrap support on estimated gene trees) and then estimating new gene trees on the combined datasets. As shown in [15], weighted statistical binning (an improved version of the original statistical binning approach) followed by appropriate summary methods is statistically consistent under the MSC+JC model under the condition that the number of genes and number of sites per gene both increase.

Note however that the guarantees of statistical consistency provided so far have nearly always made the following assumptions: every locus is recombination-free, the number of sites per locus increases without bound, and the number of loci increases without bound. These assumptions are unrealistic, since as noted in [205] recombination-free loci are generally short. Therefore, of greater relevance to practice is the question of statistical consistency where the number of recombination-free loci increases, but the number of sites per locus is bounded by some $L \in \mathbf{Z}_+$ [231, 193]. We investigate this question for the following methods[2]:

- fully partitioned maximum likelihood,
- topology-based summary methods (i.e., methods that combine gene tree topologies), and
- weighted statistical binning pipelines followed by topology-based summary methods.

We address this question under the MSC+CFN model, where the CFN is the symmetric two-state sequence evolution model (i.e., the two-state version of the Jukes-Cantor model). Perhaps surprisingly, our

[2]Only the first of these three is covered in detail in this doctoral thesis, although the publication from which this chapter has been excerpted is broader in scope. The discussion may therefore touch on the last two items occasionally, but the proofs are beyond the scope of this thesis.

results are negative: for all L, none of the approaches is statistically consistent under the MSC+CFN model and can even be positively misleading. Furthermore, this problematic behavior occurs *even when* all the genes evolve down a single model CFN tree. Therefore, expectations of accurate species trees using any of these methods given large amounts of data may be unfounded.

The key challenge to species tree estimation is *long branch attraction*, a phenomenon that can confound maximum likelihood tree estimation when sequence lengths for each genomic region are finite. In fact, we show that many species tree estimation methods that are statistically consistent when the number of genomic regions and their lengths both increase become inconsistent when only the number of regions increases, and the sequence length for each genomic region is bounded (however arbitrarily). These results suggest that many of the common approaches to species tree estimation are far from being mathematically rigorous, even under highly simplified model conditions where there is no heterogeneity between the loci. This is a very substantial limitation for multilocus phylogeny estimation methods in general, and shows that new approaches for species tree estimation are needed.

4.2 Evolution under the MSC

Our analysis is based on the MSC+CFN model. A CFN model gene tree is an unrooted binary tree $(\mathcal{T}, \Lambda)$ with topology $\mathcal{T}$ and branch lengths Λ. Under the assumption that the tree has n leaves, each site (character) χ refers to the length-n vector of character states corresponding the same homologous site for each taxon. The possible character states are $\{0, 1\}$ and evolutionary changes are modeled by a continuous-time Markov process with instantaneous rate matrix $Q = \begin{pmatrix} -1/2 & 1/2 \\ 1/2 & -1/2 \end{pmatrix}$. In particular, the probability of a change along a branch of length λ is parametrized as $p = \frac{1}{2}\left(1 - e^{-2\lambda}\right)$. Under the MSC+CFN model, each locus j evolves independently on a random gene tree $(\mathcal{T}_j, \Lambda_j)$, which is derived from the multispecies coalescent on a species tree (S, Γ, θ), where the Γ_es are the branch lengths in units of $\theta_e = 2N_e\mu_e$ with N_e and μ_e the effective population size and mutation rate of branch e. That is, on each branch e of S, looking backwards in time, lineages entering the branch coalesce at rate $2/\theta_e$ according to the Kingman coalescent. The remaining lineages at the top of the branch enter the ancestral population, and so on. See Figure 4.1 for an illustration.

We assume that all m loci evolve on the same species tree and that each locus has a constant, finite sequence length L. Let χ_{ij} represent site i on locus j, where $1 \leq i \leq L$ and $1 \leq j \leq m$, and let $\chi_{\cdot j}$ represent the set of all characters for locus j. We refer to the $\chi_{\cdot j}$ as *j-th locus sequences*. Denote the entire set of characters on all loci as X.

Inconsistency of partitioned maximum likelihood

Let $\mathcal{L}(\mathcal{T}^0, \Lambda, \chi)$ denote the likelihood function for a single site χ under the CFN model on $(\mathcal{T}^0, \Lambda)$, and let $\ell = \log \mathcal{L}$ be the log-likelihood. Under fully partitioned maximum likelihood, we seek a single binary tree topology $\mathcal{T}^0$ but allow each locus to have its own branch length parameter Λ_j; hence, the general likelihood

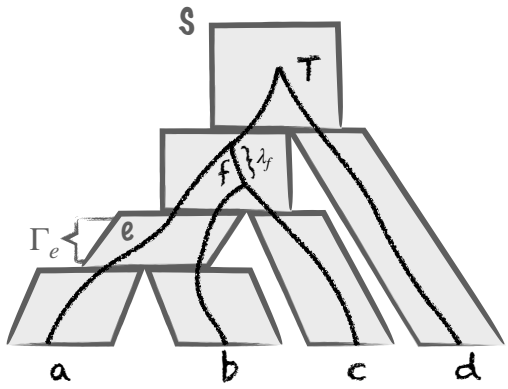


Figure 4.1: A species tree (S, Γ, θ), represented above by the "box tree," together with a gene tree $(\mathcal{T}, \Lambda)$ inside it. An incomplete lineage sorting event is depicted: in branch e of the species tree the lineages from a and b fail to coalesce, thereby producing an unrooted topology for $\mathcal{T}$ (i.e. $ad|bc$) that differs from the unrooted topology of S (i.e. $ab|cd$).

function over all sites and all loci is

$$\ell^*(\mathcal{T}^0, \Lambda_1, \ldots, \Lambda_m, X) \quad = \quad \sum_{j=1}^{m} \sum_{i=1}^{L} \ell(\mathcal{T}^0, \Lambda_j, \chi_{ij}),$$

and a maximum likelihood topology is any element of the set

$$\arg\max_{\mathcal{T}^0} \max_{\Lambda_1,\ldots,\Lambda_m} \ell^*(\mathcal{T}^0, \Lambda_1, \ldots, \Lambda_m, X). \tag{4.2.1}$$

Theorem 15 (Inconsistency of partitioned ML)**.** *Under the MSC+CFN model, fully partitioned maximum likelihood on loci with a bounded number of sites is not statistically consistent and is even positively misleading. That is, for any length $L \in \mathbb{N}$, there is a species tree with topology, branch lengths and mutation rates such that, given data generated under the MSC+CFN model, as the number of loci $m \to \infty$, the maximum likelihood topology is unique and is different from the true species tree topology with probability going to 1.*

The proof of this theorem is provided in Section 4.6.

4.3 Theoretical framework

Our analysis in fact establishes a stronger—perhaps more counter-intuitive—result. We show that partitioned maximum likelihood, topology-based summary methods, and weighted statistical binning pipelines are statistically inconsistent for multi-locus evolution *where there is no gene tree heterogeneity at all* and when all loci have only L sites for any arbitrarily selected L. By a continuity argument, we also establish that these negative results imply that these methods, which were designed to address heterogeneity across the genome resulting from ILS, are also statistically inconsistent under the MSC+CFN model.

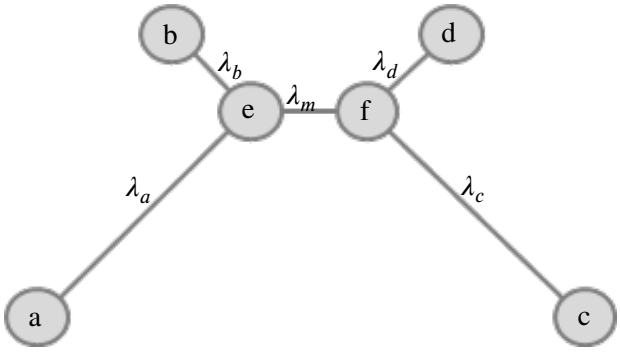


Figure 4.2: A four-taxon tree

Setting for analysis

Fix $\mathcal{T}^0$ to be the four-taxon topology $ab|cd$ on $\{a, b, c, d\}$ and let Λ^0 denote a vector of branch lengths on $\mathcal{T}^0$ under the CFN model. Specifically, denote the endpoint of the middle edge on the ab side as e, and on the cd side as f (see Figure 4.2). For this tree, denote the length of branch ae as λ_a^0, be as λ_b^0, cf as λ_c^0, df as λ_d^0 and ef as λ_m^0. For a branch length λ, we will also use the parametrization $\phi = -\frac{1}{2}\log\lambda$ in terms of which the probability of a change along this branch is

$$p = \frac{1}{2}\left(1 - e^{-2\lambda}\right) = \frac{1}{2}(1 - \phi), \tag{4.3.1}$$

and the probability of no change is $q = \frac{1}{2}(1 + \phi)$. See [199, Section 8.6] for more details on this standard parameterization. Denote the p-, q-, and ϕ-parameters as defined above for each branch using the same subscripts. We choose Λ^0 to construct a Felsenstein zone tree (i.e., a four-leaf model tree where some tree estimation methods are positively misleading, as shown in [62]) where, for a parameter $\rho > 0$, $p_a^0 = p_c^0 = \rho$ and $p_b^0 = p_d^0 = p_m^0 = \rho^3$. Note that for any $\rho > 0$, we can set $\lambda_a^0 = \lambda_c^0 = -\frac{1}{2}\log(1 - 2\rho)$ and $\lambda_b^0 = \lambda_d^0 = \lambda_m^0 = -\frac{1}{2}\log(1 - 2\rho^3)$ to satisfy this relationship. We assume that the characters $\chi_{\cdot j}$, $j = 1, 2, \ldots$, are generated under the CFN model on $(\mathcal{T}^0, \Lambda^0)$. We also denote the alternate topologies by $\mathcal{T}^* = ac|bd$ and $\mathcal{T}^1 = ad|bc$.

Basic claims

The key step in proving our main theorems is to establish the following three claims. In the first claim we show that, for any sequence length L, by taking the mutation rate small enough (through choosing ρ) in the four-taxon species tree $(\mathcal{T}^0, \Lambda^0)$ described above we can ensure that the wrong topology is chosen by partitioned ML in the limit of large gene numbers. The idea behind the proof is described in the Section "Analysis of partitioned ML" below. This claim implies Theorem 15 *in the absence of incomplete lineage sorting*. We then show in Section 4.6 that the effect of the MSC can be made negligible.

Claim 1 (Partitioned ML: Felsenstein zone)**.** *Assume that the length-L locus sequences $\chi_{\cdot j}$, $j = 1, 2, \ldots$, are generated under the CFN model on $(\mathcal{T}^0, \Lambda^0)$ and let $\hat{\mathcal{T}}_j$ be the fully partitioned maximum likelihood topology*

obtained from the sequences of the first j loci. For any length $L \geq 1$, there is $\rho > 0$ small enough such that, with probability one, $\hat{\mathcal{T}}_j \to \mathcal{T}^*$ as $j \to +\infty$.

Analysis of partitioned ML

We describe the main ideas used to prove Claim 1. First, we note that the case $L = 1$ of a single site per gene corresponds to the No Common Mechanism model of [225], under which it was shown that maximum likelihood is equivalent to maximum parsimony, establishing statistical inconsistency (together with [62]).

So we assume from now on that $L \geq 2$. Extending the results of [225] to this more general multilocus setting requires a delicate asymptotic argument. We proceed as follows:

(a) By choosing ρ small enough, we show that we can restrict the analysis to the five most common dataset types, which we refer to as *locus patterns*.

(b) We then show that, for these locus patterns, the likelihood on $\mathcal{T}^*$ dominates the likelihood on $\mathcal{T}^0, \mathcal{T}^1$, and that this domination is strict in one case.

Under our choice of branch lengths, as $\rho \to 0$, the five most common locus patterns, which we refer to as *dominant* (see Lemma 16 below for justification), are:

1. *All constant sites:* Every character has the same state on all four taxa, but that state can change from one character to another (e.g. $x^a = x^b = x^c = x^d = 0001010$). We let $\mathcal{X}_0$ be the set of such datasets and we let Q_0 be the probability of observing any $x \in \mathcal{X}_0$ under $(\mathcal{T}^0, \Lambda^0)$.

2. *One singleton site on a or c:* All sites are constant except for one, on which either a or c is different from all others (e.g. $x^a = 0111110$, $x^b = x^c = x^d = 1111110$). We let $\mathcal{X}_{11}$ be the set of such datasets and we let Q_{11} be the probability of observing any $x \in \mathcal{X}_{11}$ under $(\mathcal{T}^0, \Lambda^0)$.

3. *Two identical singleton sites on a or c:* All sites are constant except for two, each of which has the same taxon a or c different from the others (e.g. $x^a = 0011110$, $x^b = x^c = x^d = 1111110$). We let $\mathcal{X}_{2=}$ be the set of such datasets and we let $Q_{2=}$ be the probability of observing any $x \in \mathcal{X}_{2=}$ under $(\mathcal{T}^0, \Lambda^0)$.

4. *Two different singleton sites on a and c:* All sites are constant except for two, one of which has a different character state on a and the other a different character state on c (e.g. $x^a = 1001110$, $x^c = 0101110$, $x^b = x^d = 0001110$). We let $\mathcal{X}_{2\neq}$ be the set of such datasets and we let $Q_{2\neq}$ be the probability of observing any $x \in \mathcal{X}_{2\neq}$ under $(\mathcal{T}^0, \Lambda^0)$.

5. *One site with a $2/2$-split $ac|bd$:* $L - 1$ sites are constant with a single site having a and c different from b and d (e.g. $x^a = x^c = 1001110$, $x^b = x^d = 0001110$). We let $\mathcal{X}_{12}$ be the set of such datasets and we let Q_{12} be the probability of observing any $x \in \mathcal{X}_{12}$ under $(\mathcal{T}^0, \Lambda^0)$.

Note that above only the last pattern is informative and it supports the split in $\mathcal{T}^*$ rather than $\mathcal{T}^0$. Let $\widetilde{\mathcal{X}}$ be the set of all remaining locus patterns.

The next lemma, which encapsulates the key technical steps in the proof of Theorem 15, has two parts: in (a) the probability of observing the dominant locus patterns is bounded analytically; in (b) we show that the wrong topology has higher expected locus-wise likelihood under these dominant patterns. Claim 1 then follows by the law of large numbers, as detailed in Sections 4.6.

Lemma 16 (Dominant patterns and their likelihood contributions)**.** *Assume $L \geq 2$.*

(a) The probabilities of observing the dominant locus patterns are bounded as follows:

$$Q_0 = \left(\frac{1}{2}\right)^L - \mathcal{O}(\rho), \quad Q_{11} = \mathcal{O}(\rho), \quad Q_{2=} = \mathcal{O}(\rho^2),$$

$$Q_{2\neq} = \mathcal{O}(\rho^2) \text{ and } Q_{12} = \left(\frac{1}{2}\right)^L \rho^2 + \mathcal{O}(\rho^3).$$

Moreover, for all $x \in \widetilde{\mathcal{X}}$, the probability of observing x under the CFN model on $(\mathcal{T}^0, \Lambda^0)$ is $\mathcal{O}(\rho^3)$.

(b) For all $x \in \mathcal{X}_0 \cup \mathcal{X}_{11} \cup \mathcal{X}_{2=} \cup \mathcal{X}_{2\neq}$, it holds that

$$\sup_{\Lambda} \ell(\mathcal{T}^*, \Lambda, x) - \sup_{\Lambda} \ell(\mathcal{T}^0, \Lambda, x) \geq 0,$$

while, for all $x \in \mathcal{X}_{12}$,

$$\sup_{\Lambda} \ell(\mathcal{T}^*, \Lambda, x) - \sup_{\Lambda} \ell(\mathcal{T}^0, \Lambda, x) \geq K_{12} > 0,$$

for some positive constant K_{12} depending only on L. The same holds if one replaces $\mathcal{T}^0$ with $\mathcal{T}^1$ above.

Note that the big-O notation implicitly includes the contribution from L, which we treat as a constant. The detailed proofs of Lemma 16 and Claim 1 are provided in Section 4.6.

4.4 Discussion

Our results show that fully partitioned maximum likelihood is inconsistent (even positively misleading) even when there is no gene tree heterogeneity at all (i.e., when all loci evolve down a common CFN model tree), and hence by continuity under the multi-locus MSC+CFN model. The inconsistency result occurs because each locus has at most L sites (for an arbitrarily selected bound L), and the loci all evolve down gene trees that have long branch attraction (LBA). It is well known that maximum likelihood is statistically consistent even in the presence of LBA, but our results show that LBA is sufficient to bias fully partitioned ML towards the same wrong tree on each locus, and hence towards the same wrong tree for the partitioned concatenation analysis.

The same argument is used to establish that reasonable summary methods and weighted statistical binning pipelines that use these reasonable summary methods can be positively misleading when each locus has only L sites, even when there is no gene tree heterogeneity. Hence, summary methods and weighted

statistical binning pipelines do not solve this challenge, either. All the methods we addressed in this study can be seen as partitioned analyses – partitioned maximum likelihood estimates numeric parameters for each locus but keeps the tree topology the same across the loci, and summary methods estimate the gene trees independently across the loci. The fundamental challenge to multi-locus species tree estimation using these partitioned analyses (whether partitioned maximum likelihood or summary methods) is that maximum likelihood tree estimation is impacted by conditions such as LBA when the number of sites is not allowed to increase.

It is interesting to consider unpartitioned maximum likelihood under the same set of conditions. When all the loci evolve down the same CFN model tree, even though each locus has only L sites, as the number of loci increases, the unpartitioned maximum likelihood analysis will converge to the true tree; thus, unpartitioned maximum likelihood analysis is consistent under this setting. On the other hand, when there is gene tree heterogeneity resulting from ILS (as modelled by the MSC), then unpartitioned ML is inconsistent and can be positively misleading [192]. Hence, unpartitioned maximum likelihood can be statistically consistent under one setting and inconsistent (and even positively misleading) under another. In other words, unpartitioned maximum likelihood is not the solution to the challenge raised by this study.

Our analysis does not apply to multilocus methods that estimate the species tree directly from sequence data—without a gene tree reconstruction step. These include for instance METAL [47], SNAPP [31], SVDquartets [38, 39, 122], and *BEAST [82]. In particular, METAL has been shown to be consistent on finite-length genes under some assumptions on the multispecies coalescent [47]. It is also worthwhile pointing out that our results, while being based on the MSC, are likely to hold more generally for other sources of gene tree discordance, including horizontal gene transfer (HGT). Indeed, as long as rates of HGT are low enough, in the Felsenstein zone similar conclusions about inconsistency will follow for partitioned ML and summary-based methods.

4.5 Conclusion

Prior to this study, many coalescent-based species tree estimation methods were assumed to be statistically consistent under this regime, but no proofs had been provided. This study now establishes that many of the standard methods used in phylogenomic species tree estimation are statistically inconsistent.

Moreover, only a small number of methods have been proven to be statistically consistent for bounded L. Some of the summary methods described in [193] for instance are statistically consistent for $L = 1$, but the proofs depend on the strict molecular clock.

When the strict molecular clock assumption does not hold, few methods are statistically consistent for bounded L. METAL [47] is one of the few coalescent-based methods that does not require a molecular clock, and that has been proven to be statistically consistent under the MSC+CFN model. It should be noted however that the model of evolution in [47] allows mutation rates to vary across branches of the species tree, but those rates must be the same across loci, a major constraint. More recently, a log-det distance based extension of METAL has been shown consistent under more general models of substitution and population size variation, as well as certain clock-constrained models allowing variation of rates across

loci [3]. SVDquartets [38], a quartet-based method which has also been formally shown to be statistically consistent [233], builds on identifiability results [39, 122] that allow mutation rate variation across sites on each gene, not necessarily under a molecular clock.

Much remains to be understood about the important theoretical question of fixed locus length consistency of multilocus methods in general.

4.6 Proofs of the main results

We provide detailed proofs of the main claims.

Key lemmas

Proof of Lemma 16. Recall that we assume $L \geq 2$.

(a) Under our choice of branch lengths, as $\mu \to 0$, the five most common locus site patterns are:

1. *All constant sites:* Every character has the same value on all four taxa (e.g. $x^a = x^b = x^c = x^d =$ 000101). For any such $x \in \mathcal{X}_0$, x occurs with probability

$$\begin{aligned} Q_0 &= \left[\frac{1}{2}\left(1-\rho^3\right)^3(1-\rho)^2 + \mathcal{O}(\rho)\right]^L \\ &= \left(\frac{1}{2}\right)^L - \mathcal{O}(\rho), \end{aligned}$$

   where the first term in the brackets corresponds to the case of no substitution, while the second term accounts for all possibilities with at least one substitution. For convenience we denote the expression in brackets—the probability of a single site being identical on all four taxa—as q_0.

2. *One singleton site on a or c:* All sites are constant except for one, on which either a or c is different from all others (e.g. $x^a = 01\ldots$, $x^b = x^c = x^d = 11\ldots$). Any dataset with this locus site pattern occurs with probability

$$Q_{11} = q_0^{L-1}\left[\frac{1}{2}(1-\rho^3)^3(1-\rho)\rho + \mathcal{O}(\rho^2)\right] = \mathcal{O}(\rho),$$

   where the first term in the brackets corresponds to the case of a single substitution along the edge leading to the differing taxon, while the second term accounts for all possibilities involving at least two substitutions.

3. *Two identical singleton sites on a or c:* All sites are constant except for two, each of which has the same taxon a or c different from the others (e.g. $x^a = 001\ldots$, $x^b = x^c = x^d = 111\ldots$). Any dataset with this locus site pattern occurs with probability

$$Q_{2=} = q_0^{L-2}\left[\frac{1}{2}(1-\rho^3)^3(1-\rho)\rho + \mathcal{O}(\rho^2)\right]^2$$

$$= \mathcal{O}(\rho^2),$$

which follows from the same computation as in the one singleton case.

4. *Two different singleton sites on a and c:* All sites are constant except for two, one of which has a different character on a and the other a different character on c (e.g. $x^a = 100\ldots$, $x^c = 010\ldots$, $x^b = x^d = 000\ldots$). Any dataset with this locus site pattern occurs with probability

$$\begin{aligned} Q_{2\neq} &= q_0^{L-2}\left[\frac{1}{2}(1-\rho^3)^3(1-\rho)\rho + \mathcal{O}(\rho^2)\right]^2 \\ &= \mathcal{O}(\rho^2), \end{aligned}$$

which follows from the same computation as in the one singleton case.

5. *One site with a 2/2-split ac|bd:* $L-1$ sites are constant with a single site having a and c different from b and d (e.g. $x^a = x^c = 100\ldots$, $x^b = x^d = 000\ldots$). Any dataset with this locus site pattern occurs with probability

$$\begin{aligned} Q_{12} &= q_0^{L-1}\left[\frac{1}{2}(1-\rho^3)^3\rho^2 + \mathcal{O}(\rho^3)\right] \\ &= \left(\frac{1}{2}\right)^L \rho^2 + \mathcal{O}(\rho^3), \end{aligned} \tag{4.6.1}$$

where the first term in the brackets corresponds to the case of substitutions along the edges leading to the differing taxa, while the second term accounts for all possibilities with at least one substitution along the other edges.

Any remaining locus site pattern must include either a change along one of the short branches, which involves multiplication by ρ^3, or three changes along one of the long branches, which also means multiplication by ρ^3. Thus all x in $\widetilde{\mathcal{X}}$ have probability $\mathcal{O}(\rho^3)$. That concludes the proof of the claim in (a).

(b) It remains to prove (b). For each locus site pattern we will put an upper bound on the maximum of the likelihood function for topology $\mathcal{T}^0 = ab|cd$, and show that in every case the alternate topology $\mathcal{T}^* = ac|bd$ has maximum likelihood greater than or equal to this upper bound, and in at least one case is strictly greater.

Some remarks about notation first. Note that the labels we have used for the branch lengths of $\mathcal{T}^0$ can be used similarly regardless of the topology of the tree: λ_m represents the middle branch in any topology, and the others represent the branch leading to their respective taxon. Also we use Λ and Φ interchangeably, where Φ is the corresponding collection of ϕ-parameters as defined in (4.3.1). Finally we will use the following property of the ϕ-parametrization [199]: the ϕ's multiply along paths; indeed, we have for instance,

$$\begin{aligned} &\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi^0)}[x_1^a \neq x_1^b] \\ &= (1-p_a^0)p_b^0 + p_a^0(1-p_b^0) \\ &= \frac{1}{2}(1+\phi_a^0)\frac{1}{2}(1-\phi_b^0) + \frac{1}{2}(1-\phi_a^0)\frac{1}{2}(1+\phi_b^0) \end{aligned}$$

$$= \frac{1}{2}(1 - \phi_a^0 \phi_b^0). \tag{4.6.2}$$

Finally, because by inclusion the probability of observing $\chi_{\cdot 1}$ is at most the probability of observing χ_{a1}, which is simply $\left(\frac{1}{2}\right)^L$ by independence of the sites, we have

$$\sup_{\Lambda} \ell(\mathcal{T}, \Lambda, \chi_{\cdot 1}) \leq \log\left(\frac{1}{2}\right)^L = -L \log 2. \tag{4.6.3}$$

We divide up the proof of by locus site pattern.

1. *All constant sites:* Recall from (4.6.3) that, for any $\mathcal{T}$ (and, in particular, for $\mathcal{T}^0$),

$$\sup_{\Lambda} \ell(\mathcal{T}, \Lambda, x) \leq -L \log 2.$$

   For $x \in \mathcal{X}_0$, that can always (in particular, for $\mathcal{T}^*$) be achieved by setting all branch lengths to 0.

2. *One singleton site on a or c:* Without loss of generality, assume the non-constant site is site 1 and that it has $(x_1^a, x_1^b, x_1^c, x_1^d) = (1, 0, 0, 0)$. Assume also that $(x_i^a, x_i^b, x_i^c, x_i^d) = (0, 0, 0, 0)$ for all $i = 2, \ldots, L$. We can put the following upper bound on the likelihood function for $\mathcal{T}^0$. Letting $\phi_{ab} = \phi_a \phi_b$ and using (4.6.2), we have

$$\begin{aligned}
&\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_1^a{=}1, x_1^b{=}x_1^c{=}x_1^d{=}0\right) \\
&\qquad \times \mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_i^a{=}x_i^b{=}x_i^c{=}x_i^d{=}0\right)^{L-1} \\
&\leq \mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_1^a{=}1{\neq}x_1^b\right) \\
&\qquad \times \mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_i^a{=}0{=}x_i^b\right)^{L-1} \\
&\leq \frac{1}{2}\left(\frac{1-\phi_{ab}}{2}\right)\left[\frac{1}{2}\left(\frac{1+\phi_{ab}}{2}\right)\right]^{L-1},
\end{aligned} \tag{4.6.4}$$

   where the first inequality follows by inclusion. To derive our upper bound, we maximize the expression on the last line as a function of ϕ_{ab}. Taking the log, differentiating and equating to 0, we get

$$\frac{-1}{1-\phi_{ab}} + (L-1)\frac{1}{1+\phi_{ab}} = 0$$

   that is, $\phi_{ab} = \frac{L-2}{L}$. Plugging this back above, we get the upper bound

$$\begin{aligned}
&\sup_{\Phi} \ell(\mathcal{T}^0, \Phi, x) \\
&\leq L \log\left(\frac{1}{2}\right) + \log\left(\frac{1}{L}\right) + (L-1)\log\left(1 - \frac{1}{L}\right).
\end{aligned}$$

   On the other hand, for $\mathcal{T}^*$ (or, in fact, any topology), setting $\lambda_b = \lambda_c = \lambda_d = \lambda_m = 0$ and λ_a so that

$p_a = \frac{1}{L}$, we get the matching bound

$$\begin{aligned}
&\mathbf{P}_{x\sim(\mathcal{T}^*,\Phi)}\left(x_1^a{=}1, x_1^b{=}x_1^c{=}x_1^d{=}0\right)\\
&\quad\times\mathbf{P}_{x\sim(\mathcal{T}^*,\Phi)}\left(x_i^a{=}x_i^b{=}x_i^c{=}x_i^d{=}0\right)^{L-1}\\
&=\frac{1}{2}\left(\frac{1}{L}\right)\left[\frac{1}{2}\left(1-\frac{1}{L}\right)\right]^{L-1},
\end{aligned}$$

which establishes the required lower bound on $\sup_\Phi \ell(\mathcal{T}^*, \Phi, x)$.

3. *Two identical singleton sites on a or c:* For this locus site pattern, the argument is identical to the previous locus site pattern when $L \geq 4$, with the difference that the exponents in (4.6.4) are 2 and $L-2$, and accordingly throughout, giving an optimal ϕ_{ab} of $\frac{L-4}{L}$ and the upper bound $L\log(1/2) + 2\log\left(\frac{2}{L}\right) + (L-2)\log\left(1-\frac{2}{L}\right)$. This can likewise be achieved with topology $\mathcal{T}^*$ (or, in fact, any topology) if $\lambda_b = \lambda_c = \lambda_d = \lambda_m = 0$ and λ_a is set so that $p_a = \frac{2}{L}$. When $L = 2$ or $L = 3$, the optimal ϕ_{ab} is 0 and the upper bound is $2L\log(1/2)$. This can likewise be achieved with topology $\mathcal{T}^*$ (or, in fact, any topology) if $\lambda_b = \lambda_c = \lambda_d = \lambda_m = 0$ and λ_a is set so that $p_a = \frac{1}{2}$.

4. *Two different singleton sites on a and c:* Assume that $(x_1^a, x_1^b, x_1^c, x_1^d) = (1,0,0,0)$, $(x_1^a, x_1^b, x_1^c, x_1^d) = (0,0,1,0)$ and $(x_i^a, x_i^b, x_i^c, x_i^d) = (0,0,0,0)$ for all $i = 3,\ldots,L$, without loss of generality. (Recall that the case of two different singletons not involving a and c has negligible probability of being observed by part (a) and is therefore not considered here.) We will use the following property of the CFN model: on $\mathcal{T}^0$, because the path joining a, b and the path joining c, d are disjoint, the event $\{x_1^c = x_1^d\}$ is independent of the states x_1^a and x_1^b. This is immediate by the symmetry of the CFN model and the Markov property [199]. (Indeed, conditioning on the state at f has no effect on the agreement between c and d.) Using this fact as well as inclusion and (4.6.2), we get

$$\begin{aligned}
&\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_1^a{=}1, x_1^b{=}x_1^c{=}x_1^d{=}0\right)\\
&\quad\times\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_1^c{=}1, x_1^a{=}x_1^b{=}x_1^d{=}0\right)\\
&\quad\times\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_i^a{=}x_i^b{=}x_i^c{=}x_i^d{=}0\right)^{L-2}\\
&\leq \mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_1^a{=}1{\neq}x_1^b, x_1^c{=}x_1^d\right)\\
&\quad\times\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_1^a{=}0{=}x_1^b, x_1^c{\neq}x_1^d\right)\\
&\quad\times\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_i^a{=}0{=}x_i^b, x_i^c{=}x_i^d\right)^{L-2}\\
&=[\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}(x_1^a{=}1{\neq}x_1^b)\\
&\qquad\times P_{x\sim(\mathcal{T}^0,\Phi)}(x_1^c{=}x_1^d)]\\
&\quad\times[\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}(x_1^a{=}0{=}x_1^b)\\
&\qquad\times\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}(x_1^c{\neq}x_1^d)]\\
&\quad\times[\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}(x_i^a{=}0{=}x_i^b)\\
&\qquad\times\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}(x_i^c{=}x_i^d)]^{L-2}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\left(\frac{1-\phi_{ab}}{2}\right)\left(\frac{1+\phi_{cd}}{2}\right) \\
&\qquad \times \frac{1}{2}\left(\frac{1+\phi_{ab}}{2}\right)\left(\frac{1-\phi_{cd}}{2}\right) \\
&\qquad \times \left[\frac{1}{2}\left(\frac{1+\phi_{ab}}{2}\right)\left(\frac{1+\phi_{cd}}{2}\right)\right]^{L-2} \\
&= \left(\frac{1}{2}\right)^{L}\left(\frac{1-\phi_{ab}}{2}\right)\left(\frac{1+\phi_{ab}}{2}\right)^{L-1} \\
&\qquad \times \left(\frac{1-\phi_{cd}}{2}\right)\left(\frac{1+\phi_{cd}}{2}\right)^{L-1},
\end{aligned}
$$

where $\phi_{ab} = \phi_a\phi_b$ and $\phi_{cd} = \phi_c\phi_d$. Maximizing this last expression over ϕ_{ab} and ϕ_{cd} proceeds as in (4.6.4). We then get the upper bound

$$
\begin{aligned}
&\sup_{\Phi} \ell(\mathcal{T}^0, \Phi, x) \\
&\leq L\log\left(\frac{1}{2}\right) + 2\log\left(\frac{1}{L}\right) + 2(L-1)\log\left(1-\frac{1}{L}\right).
\end{aligned}
$$

On the other hand, for $\mathcal{T}^*$ (or, in fact, any topology), setting $\lambda_b = \lambda_d = \lambda_m = 0$ and $\lambda_a = \lambda_c$ so that $p_a = p_c = \frac{1}{L}$, we get

$$
\begin{aligned}
&\mathbf{P}_{x\sim(\mathcal{T}^*,\Phi)}\left(x_1^a{=}1, x_1^b{=}x_1^c{=}x_1^d{=}0\right) \\
&\qquad \times \mathbf{P}_{x\sim(\mathcal{T}^*,\Phi)}\left(x_1^c{=}1, x_1^a{=}x_1^b{=}x_1^d{=}0\right) \\
&\qquad \times \mathbf{P}_{x\sim(\mathcal{T}^*,\Phi)}\left(x_i^a{=}x_i^b{=}x_i^c{=}x_i^d{=}0\right)^{L-2} \\
&= \frac{1}{2}\left(\frac{1}{L}\right)\left(1-\frac{1}{L}\right) \times \frac{1}{2}\left(\frac{1}{L}\right)\left(1-\frac{1}{L}\right) \\
&\qquad \times \left[\frac{1}{2}\left(1-\frac{1}{L}\right)^2\right]^{L-2},
\end{aligned}
$$

which establishes the required lower bound on $\sup_\Phi \ell(\mathcal{T}^*, \Phi, x)$.

5. *One site with a 2/2-split $ac|bd$:* Without loss of generality, we assume that $(x_1^a, x_1^b, x_1^c, x_1^d) = (1,0,1,0)$ and $(x_i^a, x_i^b, x_i^c, x_i^d) = (0,0,0,0)$ for all $i = 2, \ldots, L$. Arguing as in the previous case,

$$
\begin{aligned}
&\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_1^a{=}x_1^c{=}1, x_1^b{=}x_1^d{=}0\right) \\
&\qquad \times \mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_i^a{=}x_i^b{=}x_i^c{=}x_i^d{=}0\right)^{L-1} \\
&\leq \mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_1^a{=}1{\neq}x_1^b, x_1^c{\neq}x_1^d\right) \\
&\qquad \times \mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}\left(x_i^a{=}0{=}x_i^b, x_i^c{=}x_i^d\right)^{L-1} \\
&= \left[\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}(x_1^a{=}1{\neq}x_1^b)\,\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}(x_1^c{\neq}x_1^d)\right] \\
&\qquad \times \left[\mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}(x_i^a{=}0{=}x_i^b)\right. \\
&\qquad\qquad \left.\times \mathbf{P}_{x\sim(\mathcal{T}^0,\Phi)}(x_i^c{=}x_i^d)\right]^{L-1}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{2}\left(\frac{1-\phi_{ab}}{2}\right)\left(\frac{1-\phi_{cd}}{2}\right) \\
&\qquad \times \left[\frac{1}{2}\left(\frac{1+\phi_{ab}}{2}\right)\left(\frac{1+\phi_{cd}}{2}\right)\right]^{L-1} \\
&= \left(\frac{1}{2}\right)^{L}\left(\frac{1-\phi_{ab}}{2}\right)\left(\frac{1+\phi_{ab}}{2}\right)^{L-1} \\
&\qquad \left(\frac{1-\phi_{cd}}{2}\right)\left(\frac{1+\phi_{cd}}{2}\right)^{L-1},
\end{aligned}
$$

where, again, $\phi_{ab} = \phi_a\phi_b$ and $\phi_{cd} = \phi_c\phi_d$. This bound matches the bound we obtained in the previous case. Hence, we once again get the upper bound

$$
\begin{aligned}
&\sup_{\Phi} \ell(\mathcal{T}^0, \Phi, x) \\
&\leq L\log\left(\frac{1}{2}\right) + 2\log\left(\frac{1}{L}\right) + 2(L-1)\log\left(1-\frac{1}{L}\right).
\end{aligned}
$$

However, in this case, we claim that the maximum likelihood under $\mathcal{T}^*$ is strictly greater. Indeed, letting $\lambda_a = \lambda_b = \lambda_c = \lambda_d = 0$ and setting λ_m such that $p_m = \frac{1}{L}$, we get

$$
\begin{aligned}
&\mathbf{P}_{x\sim(\mathcal{T}^*,\Phi)}\left(x_1^a{=}x_1^c{=}1, x_1^b{=}x_1^d{=}0\right) \\
&\qquad \times \mathbf{P}_{x\sim(\mathcal{T}^*,\Phi)}\left(x_i^a{=}x_i^b{=}x_i^c{=}x_i^d{=}0\right)^{L-1} \\
&= \frac{1}{2}\left(\frac{1}{L}\right) \times \left[\frac{1}{2}\left(1-\frac{1}{L}\right)\right]^{L-1},
\end{aligned}
$$

so

$$
\begin{aligned}
&\sup_{\Phi} \ell(\mathcal{T}^*, \Phi, x) \\
&\geq L\log\left(\frac{1}{2}\right) + \log\left(\frac{1}{L}\right) + (L-1)\log\left(1-\frac{1}{L}\right).
\end{aligned}
$$

Therefore

$$
\begin{aligned}
&\sup_{\Phi} \ell(\mathcal{T}^*, \Phi, x) - \sup_{\Phi} \ell(\mathcal{T}^0, \Phi, x) \\
&\geq -\log\left(\frac{1}{L}\right) - (L-1)\log\left(1-\frac{1}{L}\right) \\
&=: K_{12} > 0,
\end{aligned}
$$

where the last equality is a definition. Positivity of K_{12} can be seen, for instance, by noticing that dividing it by L gives the entropy of a 2-state random variable.

In all the above cases, a similar argument still applies if one replaces $\mathcal{T}^0$ with $\mathcal{T}^1$ (by exchanging the roles of b and d throughout). That concludes the proof of the claim in (b). □

Partitioned ML on CFN model

Proof of Claim 1. Using Lemma 16, we are now ready to prove Claim 1.

We first show that, for a fixed topology, as the number of loci grows to infinity the maximum likelihood value converges almost surely to the expected value of the maximum likelihood value on a single locus.

Lemma 17 (Convergence of the partitioned log-likelihood)**.** *Let $\mathcal{T}'$ be a fixed topology on the four taxa with branch lengths Λ'. Let also $\mathcal{T}''$ be a fixed topology on the four taxa (possibly, but not necessarily, equal to $\mathcal{T}'$). If the length-L locus sequence datasets $\chi_{\cdot j}$, $j = 1, 2, \ldots$, are generated under the CFN model on $(\mathcal{T}', \Lambda')$, then it holds that*

$$\begin{aligned} &\frac{1}{m}\sum_{j=1}^{m} \sup_{\Lambda_j} \ell(\mathcal{T}'', \Lambda_j, \chi_{\cdot j}) \\ &\to \mathbf{E}_{\chi_{\cdot 1}\sim(\mathcal{T}',\Lambda')}\left[\sup_{\Lambda} \ell(\mathcal{T}'', \Lambda, \chi_{\cdot 1})\right] \in [-4L\log 2, -L\log 2], \end{aligned} \tag{4.6.5}$$

almost surely as $m \to +\infty$. Above, the subscript $\chi_{\cdot 1} \sim (\mathcal{T}', \Lambda')$ indicates that the expectation is taken over a single locus under the CFN model on $(\mathcal{T}', \Lambda')$.

Proof. For a given topology and data set there is a unique maximum likelihood value, though the branch lengths at which it is attained may not themselves be unique. For any given locus j, there are a finite number of four-sequence data sets χ_j of length L that can occur under the CFN model. As the number of loci approaches infinity, the frequency of each data set approaches its expected value by the Strong Law of Large Numbers (SLLN) (see, e.g., [53]). To check that the conditions of the SLLN are satisfied, note that the log-likelihood is non-positive. In fact, by taking branch lengths to $+\infty$ under the CFN model, we have for any topology $\mathcal{T}$ on $\{a, b, c, d\}$ and any locus data set $\chi_{\cdot 1}$

$$\sup_{\Lambda} \ell(\mathcal{T}, \Lambda, \chi_{\cdot 1}) \geq \log\left(\frac{1}{2}\right)^{4L} = -4L\log 2. \tag{4.6.6}$$

On the other hand, because by inclusion the probability of observing $\chi_{\cdot 1}$ is at most the probability of observing χ_{a1}, which is simply $\left(\frac{1}{2}\right)^{L}$ by independence of the sites, we also have

$$\sup_{\Lambda} \ell(\mathcal{T}, \Lambda, \chi_{\cdot 1}) \leq \log\left(\frac{1}{2}\right)^{L} = -L\log 2. \tag{4.6.7}$$

So the expectation on the RHS of (4.6.5) lies in the interval $[-4L\log 2, -L\log 2]$. □

Hence, in view of Lemma 17, our goal is to show that there is $\rho > 0$ small enough such that the expected log-likelihood under $(\mathcal{T}^0, \Lambda^0)$ is higher for $\mathcal{T}^*$ than it is for $\mathcal{T}^0$ or $\mathcal{T}^1$. That is, it suffices to establish the following claim.

Lemma 18 (Expected locus-wise maximum likelihood on a fixed topology: key inequality)**.** *There exists* $\rho > 0$ *such that*

$$\begin{aligned}&\mathbf{E}_{\chi_{\cdot 1}\sim(\mathcal{T}^0,\Lambda^0)}\left[\sup_{\Lambda}\ell(\mathcal{T}^0,\Lambda,\chi_{\cdot 1})\right]\\&<\mathbf{E}_{\chi_{\cdot 1}\sim(\mathcal{T}^0,\Lambda^0)}\left[\sup_{\Lambda}\ell(\mathcal{T}^*,\Lambda,\chi_{\cdot 1})\right],\end{aligned}\tag{4.6.8}$$

and

$$\begin{aligned}&\mathbf{E}_{\chi_{\cdot 1}\sim(\mathcal{T}^0,\Lambda^0)}\left[\sup_{\Lambda}\ell(\mathcal{T}^1,\Lambda,\chi_{\cdot 1})\right]\\&<\mathbf{E}_{\chi_{\cdot 1}\sim(\mathcal{T}^0,\Lambda^0)}\left[\sup_{\Lambda}\ell(\mathcal{T}^*,\Lambda,\chi_{\cdot 1})\right].\end{aligned}\tag{4.6.9}$$

Proof. Let $\mathcal{X}$ be the set of all possible single-locus datasets. To prove Lemma 18, we expand the expectations in (4.6.8) over $\mathcal{X}$. In other words, we seek to show that

$$\begin{aligned}&\sum_{x\in\mathcal{X}}\mathbf{P}_{\chi_{\cdot 1}\sim(\mathcal{T}^0,\Lambda^0)}[\chi_{\cdot 1}=x]\\&\qquad\times\left\{\sup_{\Lambda}\ell(\mathcal{T}^*,\Lambda,x)-\sup_{\Lambda}\ell(\mathcal{T}^0,\Lambda,x)\right\}>0.\end{aligned}\tag{4.6.10}$$

We then use Lemma 16 as follows. By (a),

$$\begin{aligned}&\sum_{x\in\widetilde{\mathcal{X}}}\mathbf{P}_{\chi_{\cdot 1}\sim(\mathcal{T}^0,\Lambda^0)}[\chi_{\cdot 1}=x]\\&\qquad\times\left|\sup_{\Lambda}\ell(\mathcal{T}^*,\Lambda,x)-\sup_{\Lambda}\ell(\mathcal{T}^0,\Lambda,x)\right|=\mathcal{O}(\rho^3).\end{aligned}\tag{4.6.11}$$

Indeed, any locus site pattern in $\widetilde{\mathcal{X}}$ has probability $\mathcal{O}(\rho^3)$. Moreover, recall from (4.6.6) and (4.6.7) that the expression in absolute value is bounded by $3L\log 2$. In addition, by (a) and (b), we then arrive at

$$\begin{aligned}&\sum_{x\in\mathcal{X}}\mathbf{P}_{\chi_{\cdot 1}\sim(\mathcal{T}^0,\Lambda^0)}[\chi_{\cdot 1}=x]\\&\qquad\times\left\{\sup_{\Lambda}\ell(\mathcal{T}^*,\Lambda,x)-\sup_{\Lambda}\ell(\mathcal{T}^0,\Lambda,x)\right\}\\&\geq K_{12}\left\{\left(\frac{1}{2}\right)^L\rho^2+\mathcal{O}(\rho^3)\right\}+\mathcal{O}(\rho^3)>0,\end{aligned}$$

for $\rho > 0$ small enough.

The same argument applies for (4.6.9). □

Combining Lemmas 17 and 18 gives Claim 1. □

Part III

Statistical Multiple Sequence Alignment

General Background

BAli-Phy [210, 185, 182] is an uncommon software among those that address either of the related problems of multiple sequence alignment and phylogeny estimation. The algorithm underlying it uses a Bayesian model in which the joint alignment-tree space is endowed with a prior distribution and Markov-Chain Monte-Carlo (MCMC) simulation is used to sample from the posterior, conditional on the observed raw sequences. The model is statistically rigorous and the algorithm uses a Metropolis-Hastings step within Gibbs Sampling and clever transition kernels, one for the alignment and one for the tree, to ensure that the sampling reflects the posterior. This approach differs from the more common "two-phase" paradigm in which the alignment and tree are estimated separately or iteratively conditional on one another.

This model has several implications for how it is used. First, it effectively assumes a strictly tree-like process of sequence evolution. Although it is generally accepted that tree-like evolution is a realistic model, phenomena like recombination and HGT that violate this assumption are known to occur. Second, it offers the possibility that by jointly exploring the space of alignments and trees, a combination could be found that fits the data better than when each is estimated separately, leading to better understanding of evolution. Third, MCMC algorithms in general rely on sampling density to be effective, thus when the sample space scales exponentially in the input size (as it does here) they are naturally restricted to small inputs and require as many computational resources as are available for best results. Finally, accurately modeling the posterior density of the joint sample space is a much richer task than simple point estimation of the sequence alignment or phylogeny. Thus some additional steps are needed given the output of this method if a point estimate is desired, although it is not always clear what the best approach is or how it may affect sampling requirements.

Primarily as a result of the computational demands, BAli-Phy is often left out of quantitative accuracy studies. In the studies that have included it, which are discssed below, its results have shown greater accuracy but also the frustration of high computational demands, often failing to run on larger data. The chapters that follow aim to enhance our quantitative understanding of BAli-Phy's performance and its responsiveness to different kinds of input data, particularly on the task of multiple sequence alignment.

The first describes a small exploratory study designed to explore BAli-Phy's computational requirements and accuracy under some basic conditions. The eventual goal of that study was to find a way to run BAli-Phy reliably and accurately on alignments that were large enough that it could be used as the subset aligner within PASTA. The hope was that the more accurate alignments produced would translate to more accurate large-scale alignments by the divide-and-conquer approach. The second chapter describes the study in which that goal was realized, in part. Notably, the exploratory analysis showed a curious difference in BAli-Phy's accuracy when applied to simulated as opposed to biological data. The effort to scale BAli-Phy through PASTA was limited to simulated data as a result. The third chapter examines this difference in performance using a rigorous study and shows that this dichotomy persists for an impressive array of simulated and biological data.

Chapter 5

Preliminary Analyses

5.1 Introduction[1]

The main study evaluating alignment accuracy that included BAli-Phy was published in 2009 [116, Supplemental Information] and noted that in several cases, on data that was not unreasonably large, BAli-Phy had failed for reasons that were not clear. Since both software and hardware had evolved in that time, it was prudent to begin with an exploratory study that would show whether the higher accuracy could be reproduced on modern hardware. At the same time, that study would provide answers to several important questions related to the strategy of leveraging BAli-Phy within PASTA.

First, since BAli-Phy outputs a large set of alignment-tree pairs, a heuristic is needed to cull a single point estimate alignment from that. The previous study had used the *posterior decoding* alignment, which is the alignment that maximizes the column-wise posterior probability, but others were possible too so testing was needed to show that the heuristic was reliable. Second, it would be ideal to understand the effect of parameters like number of sequences, sequence length, running time and others on the ESS value (effective sample size, a measure of sampling adequacy in an MCMC simulation) at the point of termination. While that can be observed directly, a more subjective but arguably more important question is about the relationship between ESS and alignment accuracy. And third, the major difference between the computing environment and that of the 2009 study is the greatly increased amount of RAM available, so a reasonable question is whether BAli-Phy would encounter the same problems as before with more RAM available.

Although this was not a large study it produced some interesting data and laid the groundwork for the follow up studies contained in the following two chapters. First, it provided a clear indication over multiple replicates that there seemed to be a difference in how BAli-Phy aligned biological versus simulated data. Second, for the purposes of a point estimate of the alignment, there was no need to achieve an especially high ESS value in order to get consistently high-quality alignments.

5.2 Materials & Methods

5.2.1 Algorithms

PASTA PASTA v1.6.3 was run in its default settings, which involved three iterations, decomposes to 50% of the initial data set size and applies MAFFT on each subset. To construct the tree, it used FastTree-2 [179].

[1] The results presented in this chapter are unpublished.

MAFFT MAFFT v7.273 was run using the L-Ins-I option, which is designed for maximum accuracy at the expense of running time.

BAli-Phy BAli-Phy v2.3.6 was run on the raw unaligned sequences using no constraints and all default settings (unless otherwise specified). In every case, it was run on 32 cores in parallel, simultaneously for 24 hours, and the results were pooled into a single sample. Wherever the BAli-Phy alignment is referred to, the Posterior Decoding alignment was used.

5.2.2 Simulated Data

The simulated data included 10 replicates each of the following two data sets. Empirical statistics for all data sets are given in Table 5.1 later in this section.

- **Indelible**: this data was generated using the Indelible simulator [68] and is a subset of the simulations used in the original publication for PASTA [139].
- **RNAsim**: this data was generated using RNAsim [76] and is also a subset of the 10,000-taxon data used in [139].

In both cases, the data used in this chapter are randomly chosen 200- and 100-taxon subsets for 10 of the original 20 replicates. In particular, to ensure that additional variability was not introduced, the 100-taxon subset was chosen to be contained in the 200-taxon subset.

5.2.3 Biological Data

The biological data includes 100- and 200-taxon subsets randomly chosen from two separate, curated 16S alignments pulled from the Comparative RNA Website (CRW) [35]. As with the previous data, the subsets were pulled so that each 200-taxon subset properly contains exactly one 100-taxon subset. To allow for the possibility that 100 taxa may be too large for BAli-Phy, subsets of size 25 were also run. This also provided the opportunity to observe whether the smaller size would materially increase the ESS value.

The two CRW alignments are 16S.B.all (briefly, 16S.B) and 16S.M. Relevant descriptive statistics about these two alignments are given in Table 5.1, alongside the values for the simulated data. These two were chosen in particular from among the various CRW alignments available precisely because they represented different ends of the spectrum in terms of the number of sequences and their average similarity. The 16S.B alignment contains a large number of sequences that are highly conserved, averaging nearly 80% pairwise identity, whereas the 16S.M are closer to 60% on average, and come from a much smaller overall alignment.

Table 5.1 makes clear that these data sets represent four somewhat dramatically variable sets of conditions. The biological alignments have very high and very low gap length, but slow and mild rates of evolution, respectively, whereas the two simulated data sets have relatively higher rates of evolution, but also differing gap properties. Additionally, of course, two are simulated and two are manually curated RNA sequences.

Table 5.1: Average summary statistics for the alignments used in this section, which includes two simulated alignments (Indelible and RNAsim) and two biological alignment benchmarks (16S.B and 16S.M) from CRW. The statistics shown for the two CRW alignments reflect the properties of the single alignment prior to subsampling for individual replicates.

Statistic	**16S.B**	**16S.M**	**Indelible**	**RNAsim**
# Sequences	27,643	901	200	200
# Columns in Ref. Alignment	6,857	4,722	5,423	20,615
Avg # Gaps per Sequence	1,111	214	595	1,531
Avg. Gap Length	4.9	17.2	7.4	12.4
Avg. % seq. identity	79%	64%	33%	59%

5.2.4 Evaluation

For all data and all methods, we will examine pairwise false-negative and false-positive homology rates (SPFN and SPFP, respectively), as well as the total column score, which is the percentage of reference alignment columns that are fully reconstructed in the estimated alignment (briefly, TC).

Also, as noted earlier, one metric of interest with respect to BAli-Phy is the ESS value, so we present those and include a brief discussion thereof in the results section.

5.3 Results

5.3.1 Simulated Data

Figure 5.1 shows the results of the three methods on both sets of simulated data, at both 100 and 200 taxa. The interpretation of the data in this table is unambiguous: BAli-Phy, given 24 hours and 32 cores of running time and using the posterior decoding alignment, is a superior alignment methodology on these data. On the Indelible data, where the pairwise sequence similarity is 33%, BAli-Phy's error rate is an order of magnitude lower than MAFFT or PASTA. On the RNAsim data, which has a more complicated simulation model and indel parameters, the only exception is on the 200-taxon data, where MAFFT has a slight edge in pairwise alignment error. Despite this, BAli-Phy still generates a better total column score.

It is important to note that 200 taxa appears to be nearly an upper limit for BAli-Phy, with efforts to run any more taxa than this frequently simply failing to run and generating what appears to be a debug message in one of the output files. In fact, the ESS testing, which is discussed in further detail later, shows that for some particular parameter settings, even 200 is effectively too large of a sample to run on BAli-Phy. No experiments were done to tailor the computing environment to the data, by perhaps adding additional RAM or making other strategic changes.

One additional observation from this data is insightful. On the Indelible data, MAFFT performs quite poorly relative to PASTA, but on the RNAsim data the opposite is true. This illustrates simultaneously a significant disadvantage of MAFFT and the mechanism by which PASTA and its predecessors SATé-I [116] and SATé-II [117] were able to achieve such improvement by using MAFFT on smaller alignments. When the true alignment has such low pairwise similarity, MAFFT does not do well because it looks for spikes in

the cross-correlation functions at varying lags, but on such data the cross correlation is low to begin with so the potential to catch this signal is limited. The SATé decomposition, however, specifically divides the sequences into subsets chosen to (approximately) maximize the pairwise similarity within each group.

5.3.2 Biological Data

Figure 5.1 also contains the results of the same three methods on the two sets of replicates from the 16S reference alignments. Unfortunately, these results are more nebulous than the simulated data, but nonetheless some observations stand out.

First, as noted previously, this data is closer in average sequence similarity than either the RNAsim or Indelible, so it is perhaps not surprising to see MAFFT outperform PASTA on this data. The difference is narrow though, which suggests that the biological data have some properties that distinguish them from the simulations. Second, the high total column score compared to either simulated data set demonstrates that the 16S data contains a very high degree of variation in evolutionary rate across sites, which is of course a well-known property of the 16S data, and the conservation along the structural backbone is why such a high quality reference alignment is even possible.

But the most jarring result compared to the simulated data is the performance of BAli-Phy. BAli-Phy here shows a clear tendency to under-align and bias itself toward low false positive error but high false negative. To allow for the possibility that model misspecification could be an issue, BAli-Phy was additionally run with the substitution model specified as GTR (vs. the Tamura-Nei model [214], by default). It was also considered possible that BAli-Phy was being hampered by a long burn-in period on these sequences, so it was run additionally where it was given a sensible starting tree (in this case, the output tree from PASTA). Table 5.1 shows that neither of those modifications had a substantial effect on accuracy.

There are several possible reasons why BAli-Phy's performance could suffer in this specific way. One possibility is simply that the effective penalty for opening up a gap is too small, so when a column of the alignment has a high rate of evolution and is thus noisy, BAli-Phy prefers to insert gaps. Alternatively, it could be that BAli-Phy's assumed rates-across-sites variation is simply insufficient compared to the variation in the 16S sequences, so high-variation columns are seen as unlikely to occur very near low-variation columns, and thus indels are preferable.

Finally, note in Figure 5.1 that the performance did not improve significantly when we dropped the size of the data to 25 taxa. Here, we exclude PASTA because on such a small set, it is de facto equivalent to running MAFFT. The methods on 16S.M improved slightly, mainly because there was more room for improvement, but BAli-Phy still showed the same proclivity to under-align. Unfortunately, biological data sets with high quality reference alignments are rare, but nonetheless, these data emphasize the need to remain focused on this kind of alignment because of the unique part of the parameter space it appears to occupy.

5.3.3 ESS Values

We conclude this section with some brief observations from Table 5.2, which contains ESS values calculated from the BAli-Phy per-iteration summary statistics. By default, the minimum ESS value of all the statistics

is reported.

As discussed above, BAli-Phy was run under some additional conditions for the biological data, namely small sample size (25 taxa), using the GTR model and using a credible starting tree. The first observation is that none of these measures managed to consistently boost the ESS value. In the cases where there is a rise, it is still below 200 and thus not to a level conventially considered to have "converged". Also, on the 200-taxon data, several of the replicates under the GTR and starting-tree conditions were not able to finish running. The nature of this failure is unclear as the program does not stop running; it merely produces no output and leaves what appears to be a debugging message for one particular subroutine in the standard output file. This appears to be related to memory, or more broadly to the size of the input data. The same failure was observed more often at size 250, and universally at size 500.

Finally, the last two rows of Table 5.2 contain ESS values simulated data as a comparison. For the simulated data, where Bali-Phy consistently gave material gains in alignment accuracy, the ESS is still less than 70 for the 100-taxon data and less than 100 for the 200-taxon data. This strongly suggests that a large ESS value is not necessary for a high quality alignment. Moreover, by contrapositive, the results for BAli-Phy with the starting tree on 16S.B indicate that higher ESS values do not necessarily lead to a better alignment.

Table 5.2: ESS values for BAli-Phy runs on the various test datasets. Values shown are averages over 10 replicates in each case.

		Size		
Data	**Method**	**25**	**100**	**200**
16S.B	GTR		99	60
	1st Tree		150	69
	Default	56	66	119
16S.M	GTR		63	38
	1st Tree		64	41
	Default	196	65	84
Indelible	Default		68	39
RNAsim	Default		66	95

5.4 Conclusions

The experimental results in this chapter represent a useful comparison of a statistical co-estimation method (BAli-Phy) against an iterative method (PASTA) and a non-model-based method (MAFFT). The results show unambiguously that for two dissimilar simulation methodologies, at 100 and 200 taxa, BAli-Phy is able to improve the alignment accuracy according to essentially any measure over either of the other two methods. While the Indelible simulator uses a model quite close to BAli-Phy's, RNAsim does not, and in fact it is designed to resemble the conditions of structural RNA, e.g. 16S. However, results on the biological data show quite different phenomena when comparing BAli-Phy with the others, suggesting that there is some major difference between actual 16S RNA data and the simulated RNA sequences.

In the process, we have also shown that for estimation a single, accurate alignment, the posterior decoding

method within BAli-Phy appears to perform well without regard to ESS, or alternatively that ESS values above 60 are often sufficient. Unfortunately, this is the only clear recommendation we can offer from this data: BAli-Phy, on 32 cores with 64 GB RAM and 48 hours appears to be safe to use for alignments with an expectation of reasonable accuracy without testing for "convergence" in a traditional sense.

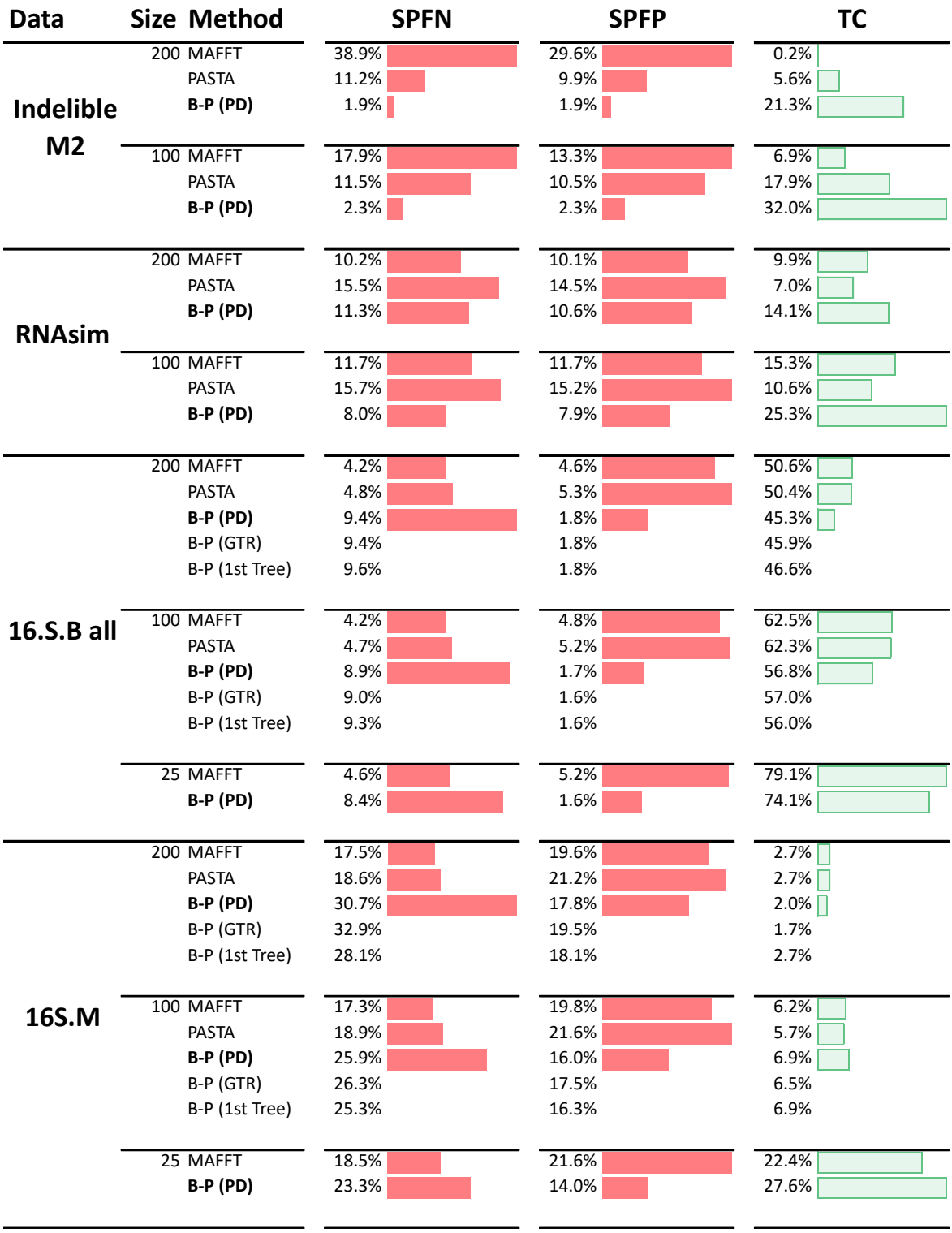


Figure 5.1: Alignment Results on all conditions tested. Here B-P (PD) refers to BAli-Phy under default settings, B-P (GTR) is BAli-Phy with the GTR substitution model specified, and B-P (1st Tree) is BAli-Phy with the PASTA tree given as a starting tree. (In all three versions of BAli-Phy, the posterior decoding alignment was used.) Bar charts are only shown for default BAli-Phy, MAFFT and PASTA for clarity.

Chapter 6

Scaling Statistical Multiple Sequence Alignment to Large Datasets

6.1 Background[1]

Multiple sequence alignment (MSA) of individual loci (where a locus is a recombination-free region within a genome) is the first step in many bioinformatics pipelines, including phylogeny estimation, protein classification, the detection of selection and co-evolution, and metagenomics. Several application areas could benefit directly from improved alignments and phylogenies of large-datasets. For example, metagenomic methods that rely on marker genes (e.g. [198], [114], [157]) invariably use genes that are present for well over 1,000 bacterial sequences and rely directly on the phylogeny to characterize the content of a shotgun sequencing sample. Improved alignments mean higher quality signal and thus more precise description of a given microbial community. Furthermore, it is well established that dense taxonomic sampling generally improves the estimation of phylogenies and multiple sequence alignments. Thus, multiple sequence alignment of datasets containing hundreds to many thousands of sequences is of increasing importance.

Numerous MSA methods have been developed, but only a few of these can analyze large datasets, and even fewer have been demonstrated to have good accuracy beyond a few hundred sequences [156]. The impact of multiple sequence alignment on downstream analyses is known to be substantial, with errors in multiple sequence alignment producing increased error rates in phylogeny estimation, false detection of positive selection, difficulties in detecting active sites in proteins, etc. [83]. Thus, highly accurate multiple sequence alignment, especially of large datasets spanning large evolutionary distances, is one of the major outstanding bioinformatics problems [151].

One of the most accurate approaches to multiple sequence alignment is statistical estimation under stochastic models of sequence evolution where sequences evolve down trees with indels (insertions and deletions) and substitutions. Yet statistical estimation of alignments or trees under these models is rarely performed, largely because the current methods for this type of analysis are too computationally intensive to use on more than about 100 sequences. While many methods have this approach [25, 161, 128, 188], BAli-Phy is the best-known, and the main such method that is used to estimate an alignment and phylogeny from unaligned sequences; [184] is the initial paper on this method, but subsequent publications extended and improved the statistical models on which the method is based.

Liu et al. [116] showed that BAli-Phy dominated SATé and other alignment and tree estimation methods on datasets with 100 sequences with respect to alignment and tree accuracy, but the analysis took several

[1]This chapter contains material previously published in [166]. The experiments were designed and supervised in collaboration with Tandy Warnow, who additionally contributed to the writing. The copyright owner has provided permission to reprint.

weeks for each dataset. Even smaller datasets can be computationally intensive (for example, a BAli-Phy analysis of a dataset with 68 sequences took about 21 CPU days [71]), and the largest dataset that BAli-Phy has analyzed may be the 117-sequence dataset studied in [133]. However, BAli-Phy may not be able to run on substantially larger datasets than this; indeed, our initial testing found that with 500 sequences, BAli-Phy failed at an early step on every run (and practical constraints tend to limit its use below even that). Indeed, although BAli-Phy has been cited often in the literature, very few benchmarks of performance are included; most simply note that BAli-Phy has a strong statistical model but is slow and computationally demanding [10, 215]. Thus, improving the scalability of BAli-Phy to larger datasets is of great interest and potentially substantial impact.

Our group has developed several techniques [156, 116, 117, 139] to improve the scalability of multiple sequence alignment methods to large datasets, of which PASTA [139] and UPP [156] provide the largest improvements. PASTA is an iterative divide-and-conquer method for co-estimating trees and alignments, in which each iteration begins with a maximum likelihood tree computed in the previous iteration, and then uses the tree to partition the sequences into small subsets that are local within the tree. Then, a selected MSA method is applied to each subset and the subset alignments on adjacent subsets (defined by the topology in the tree) are aligned together using profile-profile alignment methods. Finally, an alignment on the entire dataset is obtained by transitivity. As shown in [139], using PASTA with MAFFT [97] on the subsets made it possible to align ultra-large datasets, including one with 1,000,000 sequences, and to do so with high accuracy. UPP uses a different approach: it selects a random subset of the sequences, computes an alignment and tree (called the backbone alignment and backbone tree) on the subset using PASTA, and then represents this PASTA alignment using an ensemble of Hidden Markov Models (HMMs), each computed on a small subset of the sequences (see [156] and Section 6.2 for a description of how this ensemble is built). Each remaining sequence is then aligned to the backbone alignment using the best-scoring HMM in the ensemble. Finally, the entire set of sequences is aligned through transitivity. Like PASTA, UPP also produces highly accurate alignments of datasets with 1,000,000 sequences [156], and is more accurate than PASTA when the sequence dataset has fragmentary sequences [156].

In this study, we explore the use of both PASTA and UPP to boost BAli-Phy. PASTA is a method that has algorithmic parameters with default settings, and we use PASTA with its default settings as a starting tree. We then run one iteration of PASTA using BAli-Phy instead of MAFFT as the subset aligner, and we refer to this extension of PASTA by "PASTA+BAli-Phy". As we will show, PASTA+BAli-Phy can align 1000-sequence datasets with higher accuracy than default PASTA. We also use PASTA+BAli-Phy instead of default PASTA within UPP to compute backbone alignments with 1000 sequences, and we show that this approach produces more accurate alignments on 10,000-sequence datasets than default UPP. The improvements obtained over default PASTA and default UPP are significant, since these two methods are the current most accurate methods for large-scale and ultra-large scale multiple sequence alignment [156], especially (but not only) when alignments are used for phylogenetic estimation purposes.

The rest of this chapter is organized as follows. In Section 6.2, we describe BAli-Phy, PASTA, and UPP, and the performance study we used to evaluate the impact of integrating BAli-Phy into PASTA and UPP. In Section 6.3, we report the results of the performance study. In Section 6.3.2, we discuss the implications of

the study and future research. Additional results and discussion are provided in Section 6.5.

6.2 Methods

We ran two experiments in this study. The first experiment evaluated PASTA+BAli-Phy on 1000-sequence datasets in comparison to other alignment methods, and the second experiment evaluated the impact of using UPP to boost BAli-Phy. This is done by having PASTA+BAli-Phy compute the backbone alignment and tree, before adding the remaining sequences using UPP. We explore this on 10,000-sequence datasets. All datasets are available from prior publications.

6.2.1 Algorithms

BAli-Phy

BAli-Phy is a method that uses Gibbs sampling to alternately sample a new alignment, followed by a new phylogeny, each proportional to their likelihood under its sequence evolution model. Unlike standard phylogenetic models, such as the Generalized Time Reversible (GTR) model [216] in which only substitutions occur, the stochastic models in BAli-Phy, RS05 and subsequently RS07, also have insertions and deletions (indels). The resulting set of simulated phylogeny-alignment pairs constitutes an estimate of the joint posterior distribution.

BAli-Phy does not have a well-defined stopping rule, and will run indefinitely until it is terminated. Hence, to run BAli-Phy so as to compute a single MSA, it is necessary to define a stopping rule and a method for extracting the final alignment. In the study presented here, BAli-Phy was stopped after 24 hours of running independently on all 32 cores of a single node on the Blue Waters computing facility at UIUC [23]. Once completed, the posterior decoding (PD) alignment was computed using the `alignment-max` command within BAli-Phy and designated as the output alignment; this alignment obtained by scoring each column in the sample alignments according to how often it appears, and choosing the set of columns that a) constitutes a valid MSA on the data and b) has the largest cumulative score possible. We chose the PD alignment because prior studies have shown that the PD alignment was more accurate than the MAP (maximum a-posteriori) alignment [116, 127].

For all experiments described in this chapter, we use "BAli-Phy" to specifically refer to the protocol described above for computing a multiple sequence alignment from a given input, using BAli-Phy v2.3.6. No restrictions or starting data were provided to the software; commands for its execution, as well as for computation of the PD alignment, are provided in Section 6.5.

MAFFT

MAFFT is a well known method for multiple sequence alignment that has been consistently one of the top performing methods in terms of alignment accuracy on both nucleotide and amino acid benchmarks [57, 116]. MAFFT has many ways of being run, but its most accurate settings, such as using the local pairs (MAFFT L-INS-i) command, are computationally very intensive on large datsets. MAFFT run in default mode will

select the variant to run based on the dataset size, but will not typically have the same high accuracy as when run using the local pairs command.

PASTA

is an algorithm for large-scale multiple sequence alignment that has several algorithmic parameters that can be set by the user, but also has default settings, which we now describe. PASTA operates by initializing an alignment, then iteratively estimating a maximum likelihood (ML) tree using FastTree-2 [179] on the alignment, estimating an alignment with the help of this tree, and repeating. The calculation of the new alignment given the current tree is obtained using a specific divide-and-conquer strategy, wherein the tree is broken into subtrees through repeatedly deleting centroid edges until each subtree has a small enough number of sequences (the default maximum size is 200). Then, the preferred multiple sequence alignment method (default is MAFFT L-INS-i) is used to align each subset, yielding a set of subset MSAs. Then, every pair of subset alignments that are adjacent to each other in the tree are merged into a larger alignment using a profile-profile alignment technique (default is OPAL [234]). This produces a set of larger subset alignments that overlap and agree pairwise in all homologies for those sequences that they share; hence, an alignment on the entire set is then computed using transitivity. The number of times this process iterates can be set by the user, but the default is three. As shown in [139], PASTA improves on both SATé [116] and SATé-II [117] in terms of accuracy and scalability to large datasets.

PASTA Variants

PASTA has default settings as described above that were selected for use with MAFFT L-INS-i as the subset aligner. However, PASTA can be used with any preferred MSA method as the subset aligner. In this chapter, we examine the effect of using BAli-Phy instead of MAFFT L-INS-i within PASTA. In order to implement this, some additions to the infrastructure within PASTA were necessary. See Section 6.5 for details.

Because BAli-Phy requires 24 hours and a 32-core server to run whereas MAFFT L-INS-i runs on 200 sequences in a matter of minutes, replacing MAFFT L-INS-i for the initial iterations when the subsets are effectively (more) random would have been a poor use of expensive computing resources. We therefore chose to implement it by running PASTA in default mode (which involves three iterations), and then performing the fourth iteration using BAli-Phy as the subset aligner. Because BAli-Phy is able to run on datasets with 100 sequences, we set the decomposition size to 100 (instead of 200, which is the default setting). All other parameters were run in default mode. The two natural lines of inquiry with the tests were therefore (a) does the fourth iteration using BAli-Phy improve the alignment compared with the result after the first three iterations (i.e., PASTA in its default settings), and if so, (b) can we be sure it is due to BAli-Phy and not simply that we used an extra iteration? To explore these questions, the three variants of PASTA we tested are:

1. **PASTA(default):** PASTA with fully default settings, which means three iterations, maximum subset size 200, with MAFFT L-INS-i as the subset-aligner, and OPAL to align pairs of subset alignments. We denote this by **P(default)**.

	Sites	*p-distance* Avg	Max	Gaps / Seq	Gap Length	% Blank
RNAsim	4806	41%	61%	1036	3.1	68%
Indel. M2	2179	67%	74%	210	5.6	54%
Rose L1	3777	70%	77%	209	13.2	73%
Rose M1	3934	70%	77%	294	9.9	74%
Rose S1	2106	69%	77%	285	3.9	52%

Table 6.1: Summary statistics for true alignments on 1,000-sequence data. The p-distance is the normalized pairwise Hamming distance. Numbers shown are averages over 10 replicates.

2. **PASTA+BAli-Phy:** PASTA with three iterations under default settings, followed by one iteration with maximum subset size 100 and BAli-Phy as the subset aligner. (Equivalently, the final iteration was simply run with the phylogeny estimated in (1) specified as an input.) We denote this by **P+BAli-Phy**.

3. **PASTA+MAFFT-L:** PASTA with three iterations under default settings, followed by one iteration with maximum subset size 100 and MAFFT L-INS-i as the subset aligner. (Also equivalently specified as a single-iteration.) We denote this by **P+MAFFT-L**.

UPP

UPP is a fast multiple sequence alignment method that can be extended to 1,000,000 sequences easily, and is especially robust to fragmentary sequences compared to PASTA [156]. UPP works by choosing a random subset of (at most) 1000 sequences in the dataset to be the "backbone" and aligns those sequences with PASTA. It then constructs a collection of HMMs (called an "ensemble of HMMs") on the backbone alignment. For each of the remaining sequences, it finds the HMM from the ensemble that has the best bitscore, and uses that HMM to add the sequence to the backbone alignment. These additions are done independently, because the backbone alignment does not change during the process. UPP runs in time that is linear in the number of sequences in the input, and is also highly parallelizable. We present results using UPP with the three variants of PASTA described above to compute the backbone alignment and tree on 1000-sequence subsets of different 10,000-sequence datasets.

ML Trees

Maximum likelihood trees were estimated on each 1000-sequence alignment as well as the ground-truth alignment using RAxML [206] and FastTree-2 [179], two of the most accurate methods for large-scale maximum likelihood [115]. For the 10,000-sequence datasets, we only used FastTree-2, since RAxML is too slow on such datasets. We ran RAxML and FastTree-2 in their default modes under the GTR model with gamma-distributed rates across sites.

6.2.2 Data

In order to test the algorithms described above, a collection of simulated datasets used in [139] was downloaded from the authors' website. This collection included data generated by three separate sequence evolution simulators, Indelible [68], RNASim [76], and Rose [208]. Each of these simulators has distinct properties, and hence represents a unique set of simulation conditions. Two of the three (Indelible and RNAsim) included 10,000 sequences in each replicate, while the third (Rose) included only 1,000. For the former, ten replicates from each simulator were used and a single set of 1,000 sequences was randomly chosen from the original.

Table 6.1 contains some descriptive statistics for the reference alignments of each of the 1,000-sequence simulated data. The RNAsim data are considerably different from the other two, with longer sequences and shorter evolutionary diameter, as well as many more indels of shorter length. The Rose and Indelible data, on the other hand, are similar to each other, with the primary difference being the overall rate of evolution. Finally the individual Rose model conditions vary primarily on the length of the indels. In all, each of the three simulators provides insight into a unique part of the data space. Detailed descriptions of the simulators and the data used are provided below.

Rose

Rose is a subset of a larger collection of DNA sequences simulated using the ROSE simulator [208] that was used in [116] to evaluate SATé in comparison to other MSA methods. The ROSE simulator is a straightforward implementation of the HKY stochastic model, which is itself a close precursor to the standard Generalized Time Reversible (GTR) model [216] in use today. The simulator adds an additional model that allows the user to simulate insertions (and similarly deletions) by simulating, in order, the number of insertion events that occur, the position of each insertion followed by its length. We used 10 replicates of the 1,000-sequence datasets from the model conditions labeled 1000M1, 1000S1 and 1000L1 from [116], where the M/S/L moniker refers to the average gap length (i.e. Medium, Short or Long, respectively) of each indel event. The specific model conditions we selected have high rates of evolution, and were selected to provide a substantial challenge to the MSA methods.

Indelible

Indelible is similar to ROSE, but includes some additions that accommodate additional model complexity, such as gamma-distributed rates across sites and a codon-model. The Indelible data used for these experiments are the same data used in [139], and includes only the model condition labeled M2 in that paper, which is the highest rate of evolution of the three that were used.

RNAsim

RNAsim simulates RNA sequence evolution down a tree, specifically taking RNA structure into account, and hence represents a significant departure from the previous two. It uses a population genetics model with

selection to simulate sequence mutations, with selection favoring mutations with a relatively low free energy in its folded state. This is designed to emulate actual conditions that might plausibly be acting on mutations to RNA sequences, particularly those in a folded state such as ribosomal RNA. As a result, it has several major differences from the other simulators. First, there is no uniform substitution matrix used in the simulation. Second, site mutation probabilities are not independent of one another. Importantly, by contrast with the other two simulators, these differences are a departure from the likelihood model (GTRGAMMA) used in the maximum likelihood phylogeny estimation step of PASTA, and also a departure from the substitution model used by BAli-Phy. Therefore, results on the RNAsim data provide a test of the MSA method's robustness to model misspecification, and indirectly also test the ability of GTRGAMMA maximum likelihood phylogeny estimation to be robust to substantial model misspecification.

Evaluation criteria

We explore alignment accuracy using three standard criteria: modeller score (i.e., precision), SP score (i.e., recall), and total column (TC) score, as computed by FastSP [142]. The modeller score is equivalent to 1-SPFP, where SPFP is the "sum-of-pairs false positive rate"; similarly, the SP score is equivalent to 1-SPFN, where SPFN is the "sum-of-pairs false negative rate". These SPFP and SPFN error rates are based on homologies between nucleotides that appear in the true and estimated alignments [142]. The TC score is the fraction of the number of columns in the true alignment that are recovered in the estimated alignment. All accuracy criteria are given as a percentage, with 100% indicating perfect accuracy.

We explore phylogenetic accuracy of maximum likelihood (ML) trees computed on these alignments using the Robinson-Foulds (RF) error rate, where the RF error is the percentage of the non-trivial bipartitions in the true tree that are missing from the estimated tree. We report accuracy using "Delta-RF", which is the change in the RF error rate between the ML tree computed using the estimated alignment and the ML tree computed on the true alignment. The RF error rates were calculated using DendroPy [211].

6.3 Results and discussion

Results for experiment 1: We compare P+BAli-Phy, P+MAFFT-L, P(default), MAFFT L-INS-i, and MAFFT run in default mode; see Table 6.2. P+BAli-Phy has the top TC scores of all methods, with very substantial improvements over the second best method, which is typically P+MAFFT-L. P+BAli-Phy is also the best performing method in terms of alignment precision and recall on four of the five model conditions, and in second place on the fifth (Rose S1), where P(default) is best. However, P+BAli-Phy is within 1% of P(default) on the Rose S1 datasets in terms of precision and recall. MAFFT L-INS-i produces less accurate alignments than the PASTA variants we study, but is *much* more accurate than MAFFT run in default mode. The fact that default MAFFT has poor accuracy on these datasets shows that these are not datasets that are aligned with high accuracy by all methods; only the better methods provide good accuracy on these datasets.

Results in terms of tree error are somewhat more mixed: P+BAli-Phy is best on three of the five model conditions, in second place (behind MAFFT L-INS-i) on one condition (Rose 1000M1, or M1 for short), and

Data	Method	Prec.	Rec.	TC	Delta-*RF* RAxML	Delta-*RF* FT-2
Indelible M2	P(Default)	95.1%	94.6%	4.5%	1.86%	0.68%
	P+BAli-Phy	**98.7%**	**98.7%**	**14.6%**	**0.29%**	**-0.72%**
	P+MAFFT-L	97.2%	97.0%	6.8%	0.75%	-0.20%
	MAFFT-L	80.2%	75.0%	1.4%	15.73%	8.74%
	MAFFT-def	1.0%	0.4%	0.0%	(not run)	(not run)
RNAsim	P(default)	90.3%	90.4%	3.5%	0.56%	**0.33%**
	P+BAli-Phy	**92.1%**	**92.1%**	**8.5%**	0.70%	0.42%
	P+MAFFT-L	88.8%	89.0%	3.9%	**0.34%**	0.45%
	MAFFT-L	91.8%	91.5%	2.9%	0.73%	6.47%
	MAFFT-def	83.7%	71.5%	1.4%	(not run)	(not run)
Rose L1	P(default)	90.9%	90.6%	15.9%	2.07%	2.24%
	P+BAli-Phy	**91.8%**	**91.7%**	**33.2%**	**1.47%**	**1.51%**
	P+MAFFT-L	90.0%	89.8%	21.8%	1.98%	2.00%
	MAFFT-L	84.1%	76.6%	6.4%	3.45%	3.15%
	MAFFT-def	1.1%	0.4%	0.0%	(not run)	(not run)
Rose M1	P(default)	79.7%	79.0%	9.0%	5.35%	6.26%
	P+BAli-Phy	**79.8%**	**79.6%**	**24.4%**	4.70%	5.45%
	P+MAFFT-L	78.6%	78.2%	12.9%	5.96%	5.89%
	MAFFT-L	74.9%	63.3%	3.0%	**3.64%**	**3.90%**
	MAFFT-def	1.2%	0.5%	0.0%	(not run)	(not run)
Rose S1	P(default)	**85.3%**	**85.1%**	2.8%	3.94%	4.29%
	P+BAli-Phy	84.3%	84.3%	**10.3%**	**2.26%**	**3.59%**
	P+MAFFT-L	83.5%	83.3%	4.8%	3.55%	4.38%
	MAFFT-L	76.2%	68.2%	0.5%	3.80%	3.79%
	MAFFT-def	1.2%	0.5%	0.0%	(not run)	(not run)

Table 6.2: Alignment and tree accuracy metrics for all methods on 1,000 sequences. Note that precision, recall and TC are accuracy metrics (so larger is better) but Delta-RF is an error metric (so smaller is better). Metrics are averages over 10 replicates. Method names have been shortened slightly for space: P(default) refers to PASTA(default), P+(...) is shorthand for PASTA+(...), MAFFT-def refers to default MAFFT, and MAFFT-L refers to MAFFT L-INS-i. Bold numbers indicate best performing method.

in second or third place (depending on which ML software is used) on the remaining condition (RNASim). However, on those conditions where P+BAli-Phy does not have the highest tree accuracy, it is close to the best performing method (within 0.36% in terms of Delta-RF on the RNASim data, and within 1.6% on the Rose M1 data).

Overall, default MAFFT has the worst accuracy of all methods on these data with respect to all criteria. MAFFT L-INS-i is clearly more accurate than default MAFFT, but not as accurate as the PASTA variants in terms of alignment criteria. Hosever, MAFFT L-INS-i has the best tree accuracy on the Rose M1 datasets, and second best tree accuracy on the Rose S1 datasets.

Figure 6.1 shows results for each replicate comparing P+BAli-Phy to P(default), with respect to three metrics: TC score, Delta-RF, and SP-score. Results for Modeler score are nearly identical to SP-score,

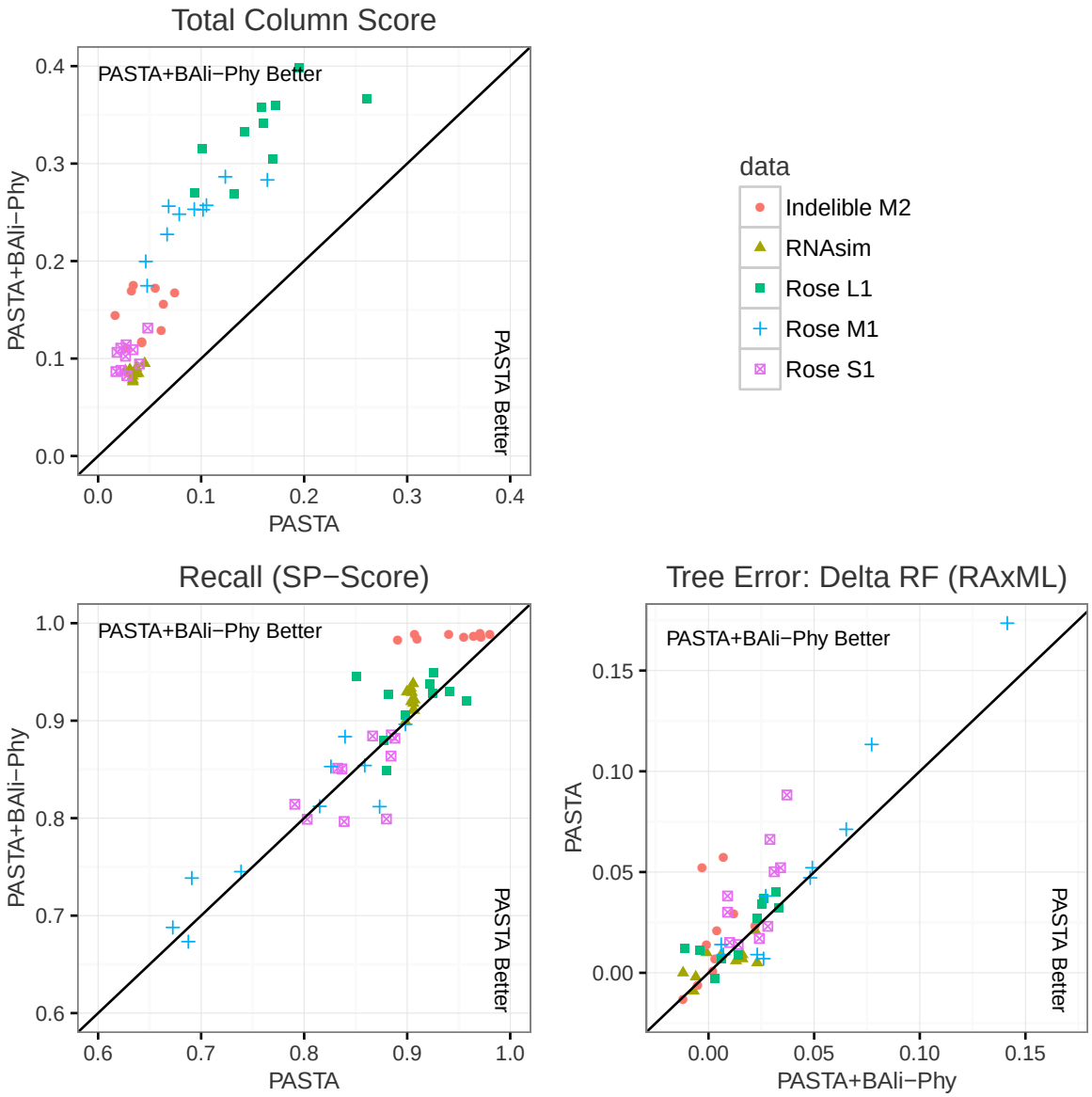


Figure 6.1: Results on 1,000 sequences, comparing default PASTA and PASTA+BAliPhy. Each point represents one replicate. PASTA denotes the alignment from PASTA under default settings (referred to as "PASTA(default)" in the text), and PASTA+BAli-Phy denotes the alignment after an additional iteration using BAli-Phy. Delta-RF refers to the difference between the RF error rates of ML trees computed on the estimated and true alignments. In each subfigure, a position above the 45-degree line indicates that PASTA+BAli-Phy is preferable; the axes for the subfigure for Delta-RF have been flipped to maintain this interpretation, since Delta-RF is an error metric rather than an accuracy metric.

and are shown in Section 6.5. Results for FastTree-2 and RAxML as the ML tree estimation method are similar; here we show results for RAxML; see Section 6.5 for FastTree-2. Points above the $x = y$ diagonal correspond to datasets in which P+BAli-Phy is more accurate than P(default) for the specified criterion, and conversely points below the diagonal correspond to datasets in which P+BAli-Phy is less accurate. Note

Data	Backbone	Prec.	Rec.	TC	Δ-RF
Indelible	P(default)	96.2%	93.6%	2.6%	0.77%
	P+BAli-Phy	**97.8%**	**95.6%**	**4.3%**	**0.54%**
	P+MAFFT-L	97.3%	95.0%	3.2%	0.62%
RNAsim	P(default)	90.8%	90.5%	0.5%	0.77%
	P+BAli-Phy	**91.4%**	**91.0%**	**0.6%**	**0.67%**
	P+MAFFT-L	89.4%	89.1%	0.5%	**0.67%**

Table 6.3: Alignment and tree accuracy metrics for UPP alignments on 10,000 sequences. Each method shown under Backbone is the method used to align the backbone of 1,000 sequences. Due to the running time required for RAxML on data of this size, Δ-RF shown is for FastTree-2 only. Bold numbers indicate best performing method.

that P+BAli-Phy has a higher TC score on *every* replicate than P(default) (all points are above the $x = y$ diagonal), and the improvement in TC score is particularly substantial (the distance to the $x = y$ diagonal is large). P+BAli-Phy also produces more accurate trees on nearly all replicates of all model conditions (note the particularly large improvements on several of the Rose S1 replicates). With the exception of the Rose S1 model condition, P+BAli-Phy is as good or better than default PASTA in terms of SP-score (more replicates above the $x = y$ diagonal than below). Furthermore, although default PASTA has slightly better SP-scores than P+BAli-Phy on several of the Rose S1 replicates, P+BAli-Phy is nearly always better with respect to tree accuracy on these replicates. The same figure comparing PASTA with BAli-Phy to PASTA with MAFFT shows virtually identical patterns and is contained in the Section 6.5.

Results for experiment 2: In Experiment 2, we compared three variants of UPP that differ only in how the backbone alignment and tree are computed; see Table 6.3. Clearly using P+BAli-Phy to compute the backbone alignment and tree has the highest alignment accuracy for all three criteria, and the gains in accuracy are largest in terms of the TC score; the second most accurate method uses P+MAFFT-L to compute the backbone alignment and tree. UPP only computes an alignment, so we computed ML trees on these three alignments using FastTree-2 (RAxML is too slow to run on 10,000 sequences). Note that the trees computed using UPP with P+BAli-Phy are within 0.67% in RF error of the tree computed using ML on the true alignment, showing that the alignment error is low enough to not impact the tree estimation by much in comparison to the tree computed on the true alignment.

6.3.1 Statistical significance

Table 6.4 shows p-values for each metric and each model condition for the hypothesis that P+BAli-Phy outperforms P(default) on measures of alignment accuracy, correcting for multiple tests using the Benjamini-Hochberg procedure, and Table 6.5 shows the same for measures of tree accuracy. P+BAli-Phy has statistically significant improvements over P(default) with respect to the TC score on all the model conditions. P+BAli-Phy also has statistically significant improvements over P(default) with respect to precision and recall (alignment modeller and SP-score) on the Indelible and RNASim datasets, but not on the Rose datasets.

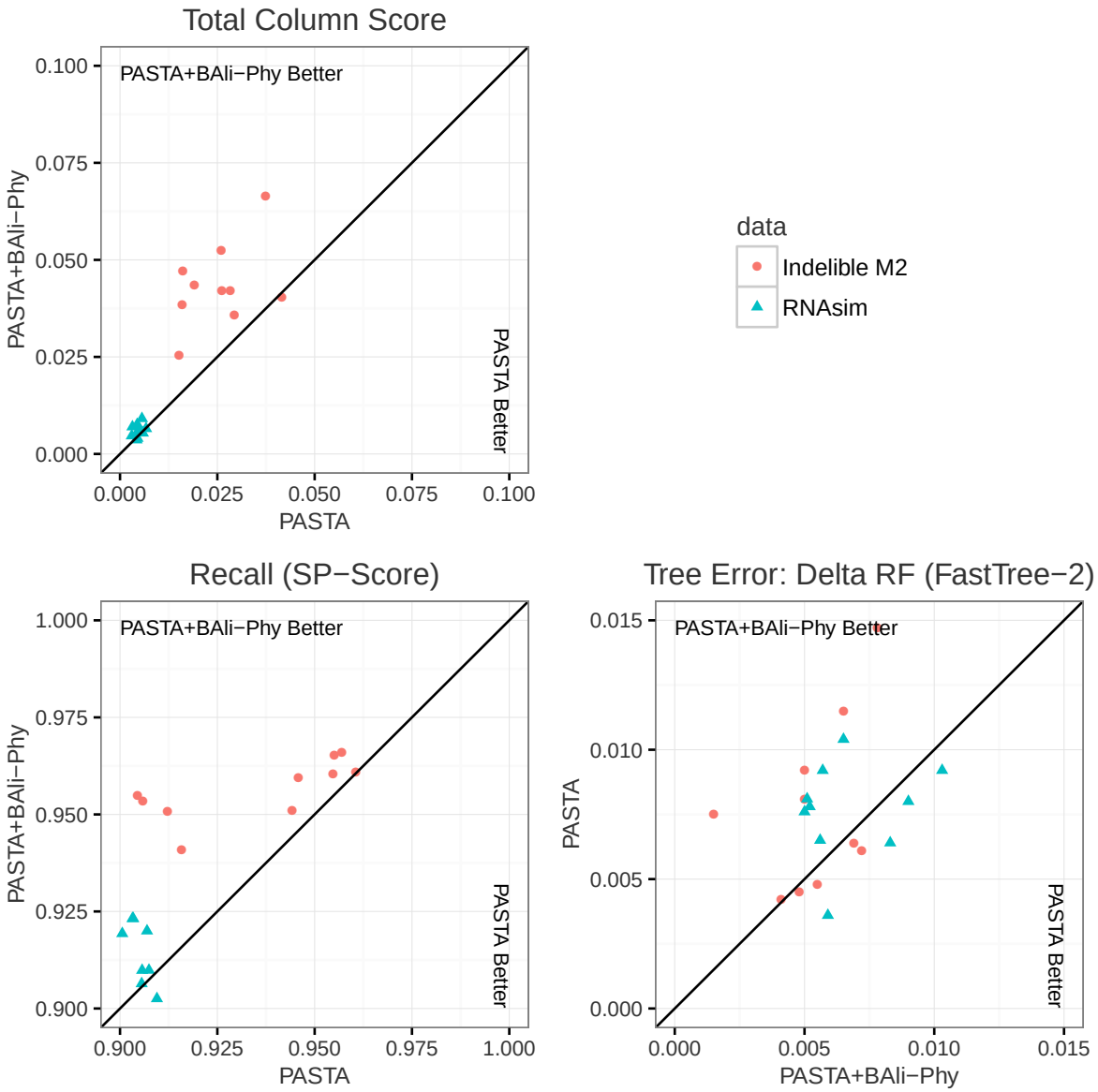


Figure 6.2: Results on 10,000 sequences Using UPP on two different backbones: one computed using default PASTA and the other computed using PASTA+BAliPhy (i.e., one iteration of PASTA using BAli-Phy as the subset aligner after default PASTA completes). Each point represents one replicate. Delta-RF refers to the difference between the RF error rates of ML trees computed on the estimated and true alignments. In each subfigure, a position above the 45-degree line indicates that PASTA+BAli-Phy is preferable; the axes for the subfigure for Delta-RF have been flipped to maintain this interpretation, since Delta-RF is an error metric rather than an accuracy metric.

ML trees computed on P+BAli-Phy alignments are also statistically significantly more accurate than ML trees computed on P(default) alignments for 6 of the 10 combinations of model condition and ML software.

Data	Precision	Recall	TC
Indelible M2	**0.001**	**0.001**	**<0.001**
RNAsim	**<0.001**	**0.001**	**<0.001**
Rose L1	0.211	0.188	**<0.001**
Rose M1	0.473	0.298	**<0.001**
Rose S1	0.820	0.770	**<0.001**

Table 6.4: *P*-values for each model condition and metric for the hypothesis test that P+BAli-Phy outperforms P(default) with respect to *alignment* accuracy. Values are based on one-sided Student's T-test for differences between the two methods on each replicate. Bolded values indicate significant differences using a Benjamini-Hochberg procedure to control the false discovery rate at 5% [17].

Data	Delta-*RF* **RAxML**	Delta-*RF* **FastTree-2**
Indelible M2	**0.021**	**0.014**
RNAsim	0.677	0.660
Rose L1	0.036	0.054
Rose M1	0.136	**0.030**
Rose S1	**0.010**	**0.007**

Table 6.5: *P*-values for each model condition and metric for the hypothesis test that P+BAli-Phy outperforms P(default) with respect to *tree* accuracy. Values are based on one-sided Student's T-test for differences between the two methods on each replicate. Bolded values indicate significant differences using a Benjamini-Hochberg procedure to control the false discovery rate at 5% [17].

6.3.2 General Observations

As this study showed, incorporating BAli-Phy into PASTA produced alignments that were generally more accurate than default PASTA, which is based on MAFFT; similarly, incorporating PASTA+BAli-Phy into UPP produced alignments that were more accurate than default UPP, which is based on default PASTA. The improvement in alignment accuracy was most noticeable for the Total Column (TC) score, where PASTA+BAli-Phy had much higher TC scores than the next best method, which was PASTA+MAFFT-L. For example, on the 1,000-sequence datasets we studied, PASTA+BAli-Phy had much higher TC scores than PASTA+MAFFT-L and default PASTA, by factors that ranged from 1.5 to 2.2 (for PASTA+MAFFT-L) and from 2.1 to 3.7 (for default PASTA). PASTA+BAli-Phy nearly always produced alignments that have higher modeller-score (precision) and SP-score (recall), with the single exception being the Rose S1 dataset with 1,000 sequences, where it was 1% lower than the best-performing (default PASTA), but both had good accuracy (precision and recall greater than 84%). The integration of PASTA+BAli-Phy into UPP produces alignments that strictly dominate the second best performing method, which is UPP run in default mode, using default PASTA to compute its backbone tree. Thus, integrating BAli-Phy into PASTA and UPP improves alignment accuracy with respect to all three criteria, with particularly large improvements for TC scores.

Perhaps the most important trend with respect to tree accuracy is that for all 10,000-sequence model conditions and nearly all 1,000-sequence model conditions, ML trees computed on the PASTA+BAli-Phy

alignments are within 1% (in terms of tree error) of the ML tree computed on the true alignment. Thus, in general, alignment error in PASTA+BAli-Phy does not increase tree error in a noticeable way over what could be computed given the true alignment. The only exceptions to this are the Rose datasets, where the increase in tree error obtained on the PASTA+BAli-Phy alignment compared to trees computed on the true alignment ranges from 1.47% (Rose L1) to 4.7% (Rose M1). However, ML trees on other alignments on those datasets also have somewhat higher Delta-RF error on these Rose datasets. Indeed, PASTA+BAli-Phy has the lowest Delta-RF error on four of the six combinations of ML method and model condition, and comes in second place on the remaining two conditions. Furthermore, when ML trees computed on PASTA+BAli-Phy alignments are not the most accurate, they are very close in accuracy to the the most accurate trees, with differences that range from 0.36% to 1.6%.

The gap length distribution affects alignment difficulty, with short gap datasets harder to align correctly than datasets with long gaps. The comparison between results on the 1,000-sequence Rose M1 (medium gap length) datasets and the 1000-sequence Rose S1 datasets is interesting, though. If alignment precision and recall are considered, then the Rose M1 datasets are more difficult, as they result in reduced precision and recall values for all methods; however, if TC scores are considered, then the Rose S1 datasets are more difficult. Clearly, model conditions impact performance with respect to the different alignment criteria differently, but generally short gaps combined with high rates of substitution create the hardest conditions.

6.4 Conclusions

This study was limited to simulated datasets where sequences evolve down model trees under processes that include insertions, deletions, and substitutions. Of the three simulators used to produce these datasets, RNASim is the most complex, and in particular includes sites that co-evolve based on the secondary structure for the RNA molecule used to design the simulation. On these datasets, we explored the use of BAli-Phy within PASTA (and then within UPP) as a point estimator of the true sequence alignment. Our study shows that incorporating BAli-Phy into PASTA and UPP enables BAli-Phy to be extended to large and ultra-large datasets, and to produce more accurate alignments than the default settings for PASTA and UPP, which are the current best alignment methods for large-scale and ultra-large-scale multiple sequence alignment. Indeed, what this study shows is that integrating BAli-Phy into PASTA means that a dataset with 1000 sequences can be aligned in the same time as 10 independent BAli-Phy analyses of 100 sequences each. Furthermore, once a dataset of this size is computed, larger datasets can be aligned very quickly by using the PASTA+BAli-Phy alignment as the backbone alignment and tree in UPP. Thus, even though this approach does not address how to speed up BAli-Phy for 100-taxon datasets, it does show that BAli-Phy can be scaled to much larger datasets in an essentially linear fashion.

Yet there are several limitations to this study. First, although we explored this technique with BAli-Phy, we did not explore it with other statistical methods. However, since the parameters of the divide-and-conquer strategy (especially the maximum subset size) can be adjusted to suit the given base MSA method, this extension can be easily done. Thus, methods such as StatAlign [161], which may be limited to even smaller datasets, could also be tested in this framework. Similarly, methods such as PAGAN [126] are impacted

by dataset size and the challenge in estimating good guide trees, and PASTA's phylogenetically-informed divide-and-conquer strategy might be useful techniques to improve their scalability to large datasets that have evolved quickly. Thus, future work should evaluate the impact of this type of strategy on StatAlign, PAGAN, and other statistical methods.

Our study also only examined minor adjustments to the algorithmic parameters for PASTA and UPP; additional research to optimize the parameters involved in this implementation could lead to substantial improvements, as essentially no parameter tuning was done.

This study was limited to simulated datasets, and so the potential for this type of approach to provide improvements on biological datasets is unknown. One of the challenges is that most biological alignment benchmarks are amino acid datasets; while BAli-Phy can analyze amino acid sequences, it is even more computationally intensive on amino acid datasets than on nucleotide datasets, and it is not known whether the statistical approach in BAli-Phy will provide advantages for structural alignment estimation.

Finally, one of the appealing aspects of the Bayesian approach in BAli-Phy is that it returns a sample from the distribution on multiple sequence alignments and trees. This study only explored BAli-Phy as a point estimator of the alignment, and so in a sense does not truly scale BAli-Phy to large datasets. Scaling Bayesian methods such as BAli-Phy so that they achieve their full potential on large datasets is clearly of great interest, and future work should attempt to do this.

6.5 Additional Information

6.5.1 Technical Modifications to PASTA

The default version of PASTA v1.6.3 is designed to run on a single node using a shared-memory implementation with multiple processors, and to run consecutively from start to finish using a single command. However, running PASTA in this manner is not practical using BAli-Phy as the subset aligner; with 1,000 sequences and a maximum subset size of 100, PASTA requires 10-16 subset alignments that require 32 CPU-days apiece. While this is theoretically possible given the right computing resource, shared resources such as university clusters frequently have constraints on job time, to name one, that prohibit a continuous implementation of PASTA. Instead, we needed an implementation of PASTA that allows subsets to be aligned separately, on different servers or at different times if necessary.

The solution was to add an optional checkpoint just prior to the subset alignment step where it will save its state, terminate, and when restarted, resume from the previous point assuming that the subsets had been aligned and copied to the appropriate path. When PASTA is run with this option, upon reaching the checkpoint it outputs a text file with paths to the subset sequence files and target paths for the aligned version. This addition is therefore useful not just for using BAli-Phy as the subset aligner, but any method that is not currently implemented, or even to achieve additional parallelism on data where the number of subsets makes a single node impractical. The commands and links to source code for this branch of the PASTA code are provided in the following section.

6.5.2 Software Commands Used

In the sections below, <datatype> refers to whether the sequences are DNA, RNA or amino acid sequences. This was DNA for all of our data except for RNAsim, which was RNA.

PASTA

The following commands assume that the work for PASTA(default), PASTA+MAFFT-L and PASTA+BAli-Phy sit in folders called respectively <pasta_def_folder>, <pasta_mafft_folder> and <pasta_bp_folder>. For PASTA(default) and PASTA+MAFFT-L, the commands are:

Default:

```
python run_pasta.py -d <datatype> -i <raw_seqs> -j pasta_default -o
<pasta_def_folder>/<model-replicate folder> --temporaries=<temps_folder> -k
```

MAFFT:

```
python run_pasta.py -i <raw_seqs>
-t <pasta_def_folder>/<model replicate folder>/pasta_default.tre
-o <pasta_mafft_folder>/<model/replicate folder> -j pj_mafft -d <datatype>
--temporaries=<temps_folder> --max-subproblem-size=100 -k --keepalignmenttemps
--iter-limit=1
```

For PASTA+BAli-Phy, there are two commands. The first starts PASTA and does the decomposition, and then exits after outputting a list of files that need to be aligned and saving its state. The second resumes from the previously saved state under the assumption that the alignments have been completed and are in their target locations on disk and proceeds with the remainder of the algorithm. The commands are:

Start:

```
python run_pasta.py -i <raw_seqs>
-t <pasta_def_folder>/<model replicate folder>/pasta_default.tre -o
<pasta_bp_folder>/<model-replicate folder> -j pastabp -d <datatype>
--interruptible --temporaries=<temps_folder>
--max-subproblem-size=100 -k --keepalignmenttemps
--iter-limit=1
```

Finish:

```
python run_pasta.py -i <raw_seqs> -o <pasta_bp_folder>/<model replicate folder>
-j pastabp -d <datatype>
-t <pasta_def_folder>/<model replicate folder>_default.tre
```

```
--interruptible --temporaries=<temps_folder> --max-subproblem-size=100
--resume-state-path=<pasta_bp_folder>/<model replicate folder>/
pastabp_temp_iteration_0_picklefile -k --keepalignmenttemps --iter-limit=1
```

The option `--interruptible` is necessary in both and indicates to PASTA that it should stop at the checkpoint. In the second command, the option `--resume-state-path` is the path to the file containing the state at checkpointing and the sample argument there is consistent with the naming that PASTA will use for that file.

BAli-Phy

BAli-Phy was run on Blue Waters by setting up a job with a wall-clock time limit of 24 hours and submitting it with the following shell script, which starts a background BAli-Phy process once for every processor found by the `nproc` Linux command (on Blue Waters, it is 32). In the following, the variable `${id}` is assumed to be a name that identifies the particular alignment being run.

```
for iteration in $(seq 1 $(nproc))
do
bali-phy <subset-fasta> --name <job_id_work_folder>/${id}_out/${id}-${iteration}
 > <job_id_work_folder>/${id}_out/log_$iteration.txt 2>&1 &
done
wait
```

After the job has completed the following is run, which removes the first 10 samples from the output for each processor, then concatenates the rest into a single file. Then the posterior-decoding alignment is computed in the final line with the `alignment-max` command.

```
for iteration in $(seq 1 32)
do
cut-range --skip=10 <
<job_id_work_folder>/${id}_out/${id}-${iteration}-1/C1.P1.fastas
>> <job_id_work_folder>/${id}_out/combined.fasta
done
cat <job_id_work_folder>/${id}_out/combined.fasta | alignment-max >
<job_id_work_folder>/${id}_out/${id}_posterior_decoding.fasta
```

MAFFT

We ran MAFFT in two ways: its default version, and its MAFFT L-INS-i version. The command for the L-INS-i algorithm is:

```
mafft-linsi <raw_seqs> > <output_file>
```

UPP

Other that specifying the backbone alignment and tree, UPP was run with default settings. The command is:

```
python run_upp.py -m <datatype> -s <raw_query_seqs> -d <work_folder> -t <tree>
-a <backbone_alignment> -p <temps_folder>
```

In this command, <raw_query_seqs> is a fasta file with the unaligned sequences, *excluding* the sequences included in the backbone alignment. <tree> refers to a specified phylogenetic tree on the backbone alignment and <backbone_alignment> is the corresponding alignment itself. Both were the default output by PASTA for that particular backbone. By default, PASTA uses FastTree-2 to find its final tree estimate using the parameters -nt -gtr -gamma -fastest, and this was the tree used in the command above.

RAxML & FastTree-2

Finally, for maximum-likelihood trees estimated directly (i.e. not from within PASTA), both methods were run using the GTR-Γ model. RAxML was run with 8 threads specifically, and FastTree was run without specifying the number of threads. The commands are:

RAxML

```
raxmlHPC-PTHREADS-AVX -s <alignment_fasta> -w <work_folder> -n <tree_name>.tre
-m GTRGAMMA -p 100 -T 8
```

FastTree-2

```
FastTreeMP -quiet -gtr -gamma -nt <alignment_fasta> > <tree_name>.tre
```

6.5.3 Additional Scatter Plots for 1000-sequence Data

In this section, we present all three pairwise comparisons of all metrics on the 1000-sequence data. The three methods considered are, briefly:

a. **PASTA(Default)** PASTA under default settings: 3 iterations, all with MAFFT (L-Ins-I) as the subset aligner and decomposition to a maximum size of 200 sequences.

   For each of the next two methods, we run PASTA for 1 iteration with the initial tree taken from the output of this method, which is equivalent to running PASTA for 4 iterations with slightly different settings on the final cycle.

b. **PASTA+BAli-Phy** PASTA for 1 iteration, with the BAli-Phy Posterior Decoding alignment on subsets and decomposition to a maximum size of 100 sequences. Using the PASTA(default) output phylogeny as an initial tree.

c. **PASTA+MAFFT-L** PASTA for 1 iteration, with MAFFT L-Ins-I as the subset aligner and decomposition to a maximum size of 100 sequences. Using the PASTA(default) output phylogeny as an initial tree.

Figure 6.3 shows full results (for all five criteria) corresponding to Figure 6.1, which is partially duplicated here. In all subfigures in this section, each point represents one replicate and a position above or below the 45-degree line is interpreted as favoring one method or the other consistently across the page. This requires inverting the axes for the bottom panels compared to the upper three, but maintains a consistent interpretation.

Note that the figures for SP-score and Modeller score are nearly identical. Delta-FN results using FastTree-2 and RAxML have some differences in terms of magnitude, but not in terms of relative performance. The remaining figures are the equivalents of Figure 6.1, with the difference that each compares a different pair of methods. The second compares PASTA+MAFFT-L to default PASTA. The fourth iteration with MAFFT L-INS-i improves the TC score but does not improve either precision or recall. The comparison with respect to precision and recall shows that PASTA+MAFFT-L is better than default PASTA on the Indelible datasets, but about the same on the Rose data, and less accurate than default PASTA on the RNAsim data.

The final figure compares PASTA+BAli-Phy to PASTA+MAFFT-L directly. Consistent with the previous figure, PASTA+BAli-Phy seems to be much better than PASTA+MAFFT-L in much the way that it was better on PASTA run in default setting.

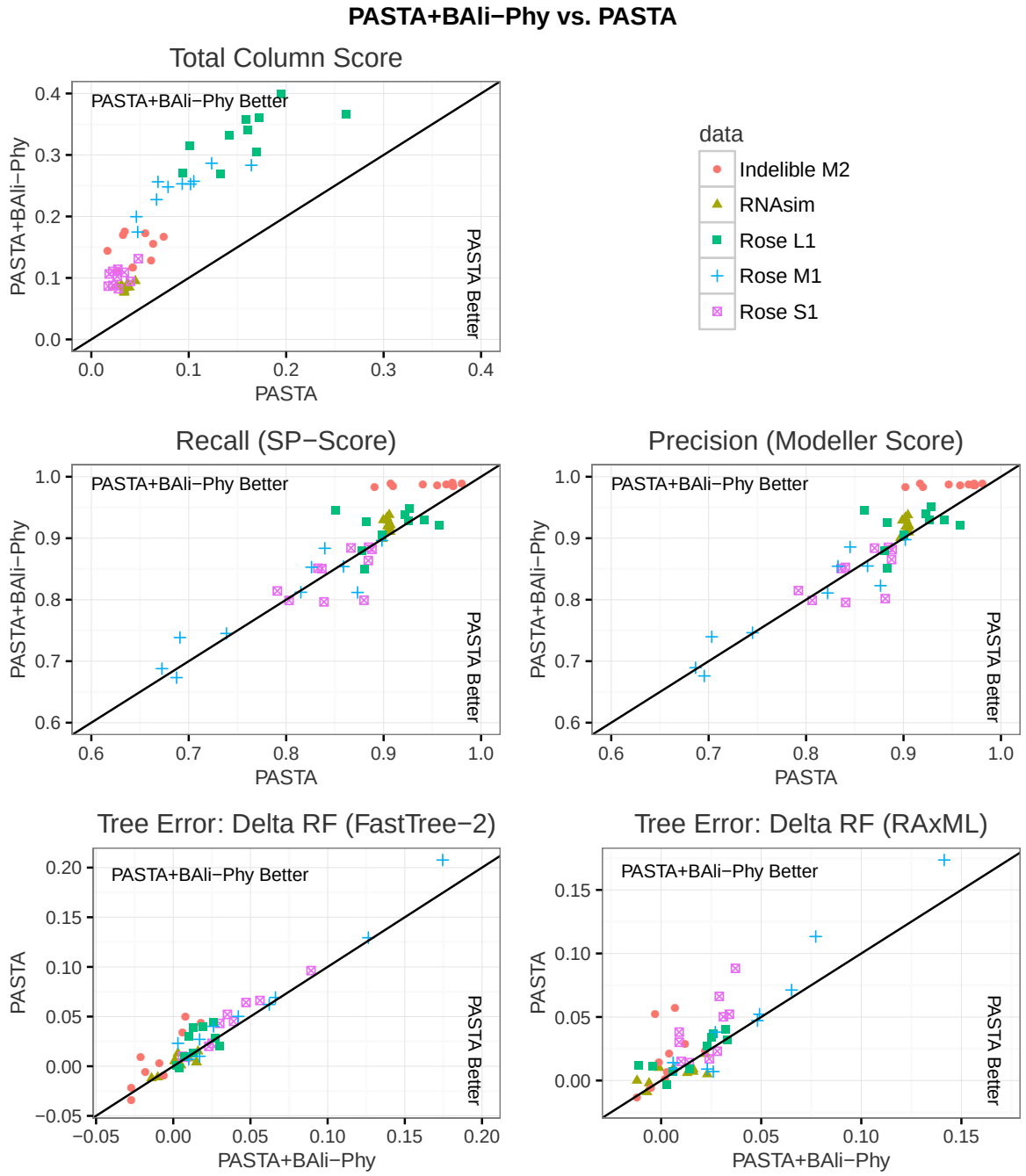


Figure 6.3: Pointwise comparison of five criteria for PASTA before and after a final iteration using BAli-Phy as the subset aligner. PASTA denotes the alignment from PASTA under default settings (referred to as "PASTA(default)" in the text). Delta-RF refers to the difference between the RF error rates of ML trees computed on the estimated and true alignments.

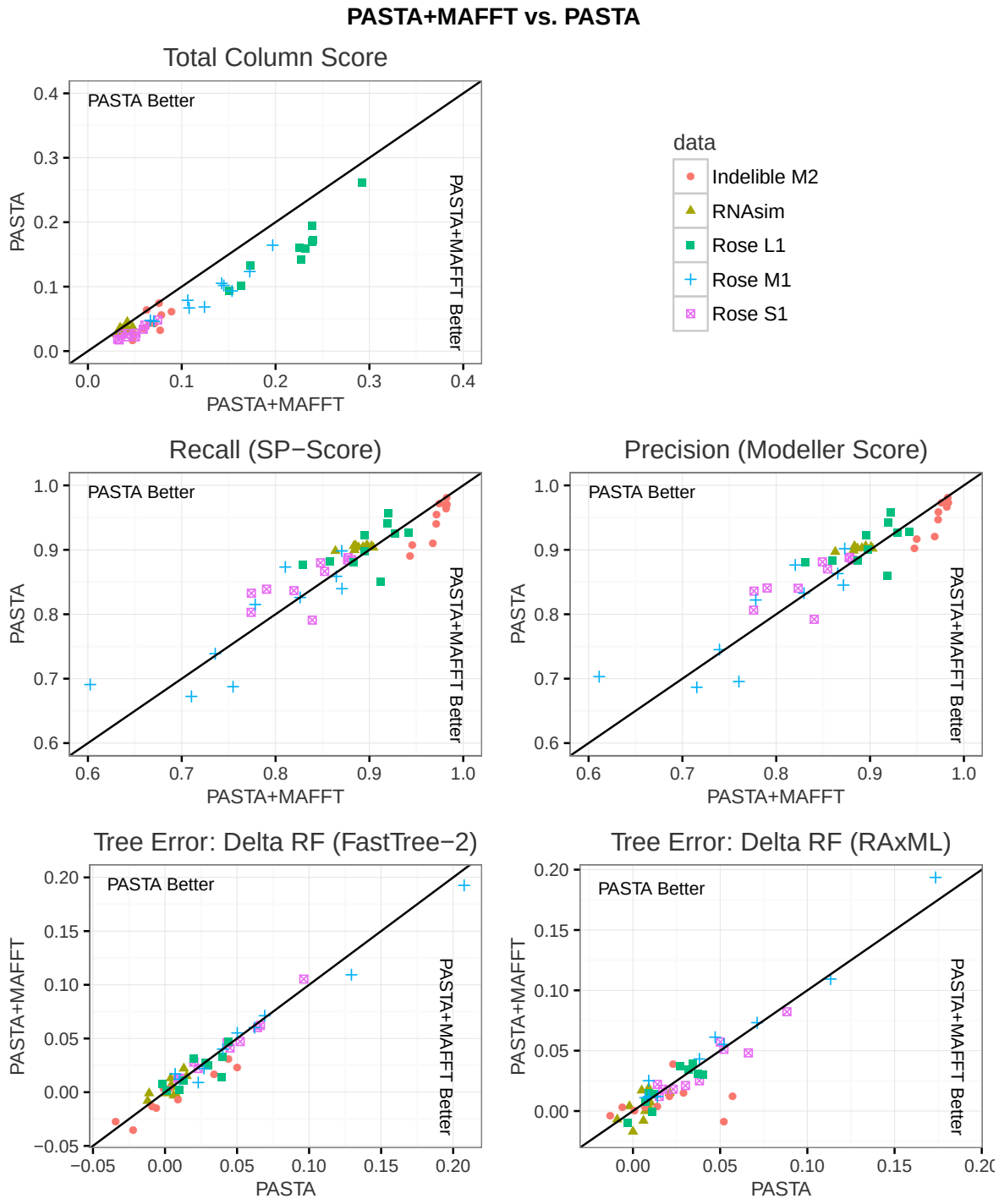


Figure 6.4: Pointwise comparisons between PASTA run in default setting (referred to as PASTA here) and PASTA+MAFFT-L, analogous to Figure 6.3. It appears that the fourth iteration with MAFFT improves the TC score but does not improve either precision or recall. PASTA+MAFFT-L is better for precision and recall on Indelible, but worse on RNAsim and roughly even on Rose.

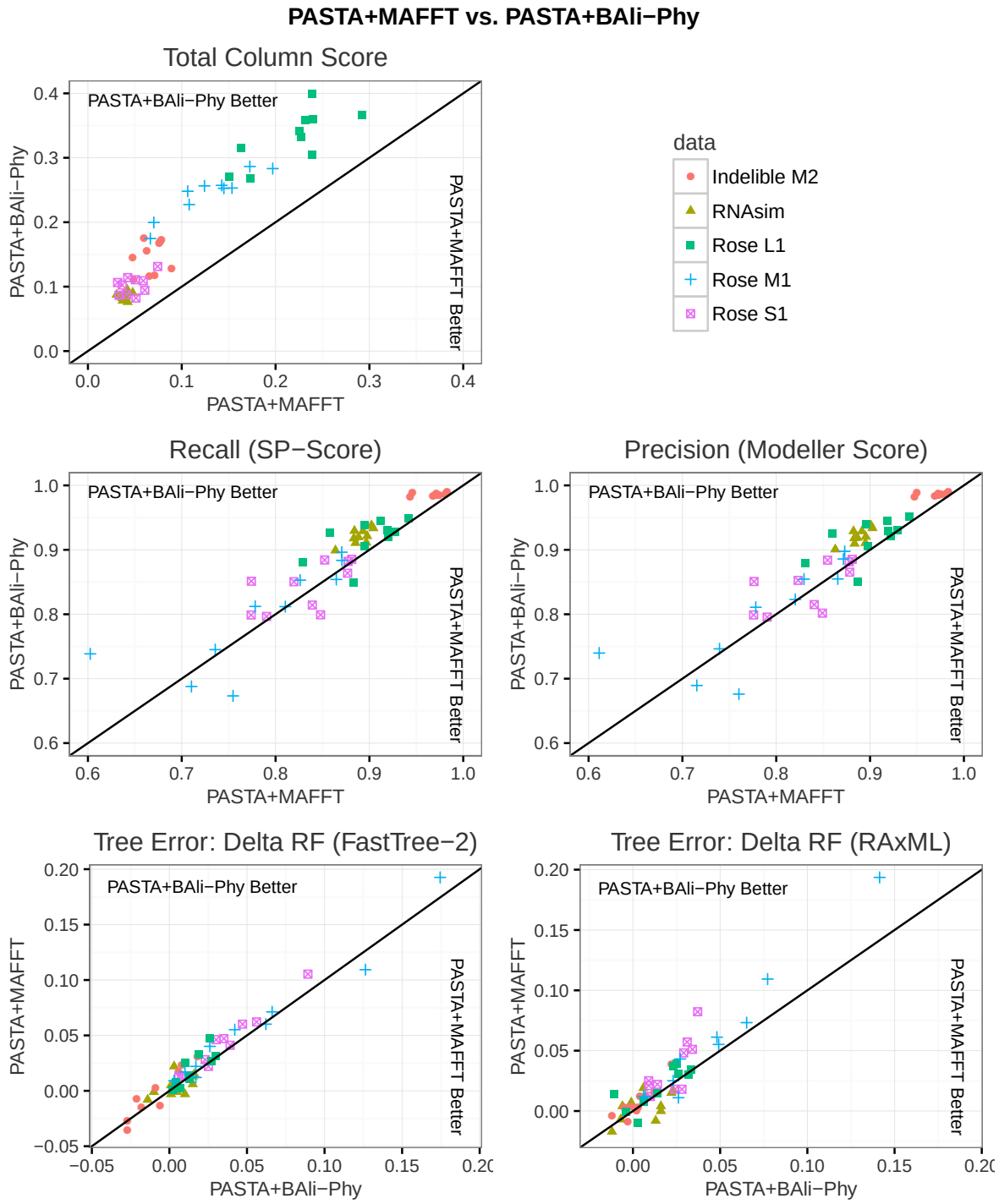


Figure 6.5: Same as Figure 6.4, but comparing PASTA+MAFFT-L to PASTA+BAli-Phy. Recall that PASTA+BAli-Phy refers to a single iteration of PASTA run with BAli-Phy as the subset aligner on subset-size 100, using the tree from PASTA as the starting tree. PASTA+MAFFT-L is analogous with MAFFT in place of BAli-Phy.

6.6 Comparison of RAxML vs. FastTree-2

Table 6.6 gives a detailed comparison of the actual error rates for each of the two maximum-likelihood tree estimation techniques.

		RF Error		
Data	**Alignment**	**RAxML**	**FT-2**	**Diff.**
Indelible M2	P(default)	24.1%	32.2%	-8.1%
	P(BAli-Phy)	22.6%	30.8%	-8.2%
	P(MAFFT)	23.0%	31.3%	-8.3%
	Reference	22.3%	31.5%	-9.2%
RNAsim	P(default)	16.1%	16.4%	-0.3%
	P(BAli-Phy)	16.3%	16.5%	-0.3%
	P(MAFFT)	15.9%	16.5%	-0.6%
	Reference	15.6%	16.1%	-0.5%
Rose L1	P(default)	14.0%	13.4%	0.6%
	P(BAli-Phy)	13.4%	12.7%	0.7%
	P(MAFFT)	13.9%	13.2%	0.7%
	Reference	11.9%	11.2%	0.7%
Rose M1	P(default)	16.8%	17.0%	-0.1%
	P(BAli-Phy)	16.2%	16.1%	0.1%
	P(MAFFT)	17.5%	16.6%	0.9%
	Reference	11.5%	10.7%	0.8%
Rose S1	P(default)	14.7%	14.0%	0.7%
	P(BAli-Phy)	13.0%	13.3%	-0.2%
	P(MAFFT)	14.3%	14.1%	0.3%
	Reference	10.8%	9.7%	1.1%

Table 6.6: RAxML and FastTree-2 RF error details for each dataset and each method. On Indelible, FastTree-2 generates a tree that is noticeably less accurate than RAxML for all alignments, including the reference, and it is not clear what causes that. On all other data the two perform about equally.

Chapter 7

Benchmarking Statistical Multiple Sequence Alignment on Amino Acid Sequences

7.1 Background[1]

Multiple sequence alignment is a basic step in many bioinformatics pipelines, including phylogenetic estimation, but also for analyses specifically aimed at understanding proteins. For example, protein alignment is used in protein structure and function prediction [44], protein family and domain identification [150, 72], functional site identification [4, 197], domain identification [19], inference of ancestral proteins [86], detection of positive selection [69], and protein-protein interactions [247]. However, multiple sequence alignment is often difficult to perform with high accuracy, and errors in alignments can have a substantial impact on the downstream analyses [105, 148, 167, 69, 49, 201, 230, 96, 176]. For this reason, the evaluation of multiple sequence alignment methods (and the development of new methods with improved accuracy), especially for protein sequences, has been a topic of substantial interest in the bioinformatics research community (e.g., [230, 219, 89, 170, 108]).

Protein alignment methods have mainly been evaluated using databases, such as BAliBase [13], Homstrad [145], SABmark [228], Sisyphus [8], and Mattbench [46], that provide reference alignments for different protein families and superfamilies based on structural features of the protein sequences (see discussions about these benchmarks in [9, 89]). Performance studies evaluating protein alignment methods using these benchmarks (e.g., [57, 22, 200, 219, 100, 139, 156]) have revealed conditions under which alignment methods degrade in accuracy (e.g., large data sets, or highly heterogeneous data sets with low average pairwise sequence identity) and have also revealed differences between alignment methods in terms of accuracy, computational efficiency, and scalability to large data sets. In turn, the databases have been used to provide training data for machine learning techniques to infer alignments on novel data sets (e.g., [50, 194]). Method development for protein alignment is thus strongly influenced by these databases, and has produced several protein alignment methods that are considered highly accurate and robust to many different challenging conditions.

An alternative approach to multiple sequence alignment has been developed within the statistical phylogenetics community in which an alignment is co-estimated with a phylogenetic tree by considering stochastic models of evolution in which sequences evolve down a model tree under a process that includes substitutions,

[1]This chapter contains material previously published in [165], which was a joint work with Ehsan Saleh and Tandy Warnow. It has been edited slightly for brevity. Author contributions on that publication were as follows: TW conceived of the project. TW & MN designed the experiments. ES & MN performed the experiments. TW, MN and ES analyzed the data. ES and MN created figures, and TW and MN wrote the paper. Permission to reprint has been provided by the copyright holders.

insertions, and deletions (jointly referred to as "indels"). Likelihood-based estimation of alignments and/or trees under these models provide a mathematically rigorous and therefore appealing approach, and was initially proposed in [20]. Subsequent extensions of this basic approach were made in a sequence of papers [221, 222, 223, 85, 135, 136, 137, 81, 128, 67, 127, 210, 185, 161, 28, 182]. BAli-Phy [210, 185, 182], a Bayesian method that uses MCMC sampling to jointly estimate the multiple sequence alignment and phylogenetic tree under a stochastic sequence evolution model that allows for indels and substitutions, is the most well-known of these methods.

A related approach is PRANK [125], which closely adheres to a phylogenetic model of sequence evolution but does not rely on a detailed stochastic model to the same degree. Because of its similarity in design objectives, [21] refer to PRANK as a "heuristic to full statistical alignment". Figure 1 in [21] examined alignments computed using BAli-Phy and PRANK in comparison to other methods on biological protein data sets, and showed that alignments produced by BAli-Phy and PRANK were outliers in the set, while the remaining methods largely grouped together.

Only a few studies have evaluated BAli-Phy for accuracy on either biological or simulated datas. Three studies [116, 182, 166] evaluated BAli-Phy on simulated nucleotide data sets and found it to have superior accuracy compared to the other alignment methods they examined; this question was examined directly in [116, 182] and indirectly in [166] through the substitution of MAFFT [97] by BAli-Phy within PASTA [139], a divide-and-conquer meta-method that is designed to scale MSA methods to larger data sets.

Additionally, [98] evaluated BAli-Phy on protein biological benchmarks as well as on simulated protein data sets (to the best of our knowledge, this is the only study that has evaluated BAli-Phy in terms of accuracy on biological data). In their study, BAli-Phy was much less accurate than some other MSA methods (PRANK, Muscle [58], and variants of MAFFT) on the biological data, but was very good (and for some criteria it was the best) on the simulated data. This study is intriguing but its evaluation of BAli-Phy is limited; the data analyzed were large for BAli-Phy (the simulated data sets had 100 sequences, and the biological data sets ranged up to 100 sequences), and each run was limited to 1000 MCMC iterations. As discussed by the authors (and in [183]), 1000 MCMC iterations may not have been sufficient to allow BAli-Phy to reach convergence on data sets of this size, and it is known that BAli-Phy can have reduced accuracy if stopped prematurely [183]. The contrast in performance on biological and simulated data is notable but a more careful evaluation of BAli-Phy is necessary to understand its performance on biological benchmark data sets.

Present Study

Here we report on an extensive performance study in which we compared BAli-Phy version 2.3.8 to a collection of multiple sequence alignment methods on a diverse range of input data. First we compare it to the leading protein sequence alignment methods. We used 1192 data sets from four established benchmark databases of protein multiple sequence alignments (BAliBASE v3.0, Sisyphus v1.2, Mattbench, and Homstrad, all downloaded in March 2017) as well as 120 simulated data sets in order to characterize the relative and absolute accuracy of the methods we explore. We limited our study to biological sequence data sets with at most 25 sequences and to simulated data sets (under 6 model conditions) with 27 sequences, so that we

were able to run BAli-Phy long enough to get adequate sampling density. In particular, we ran BAli-Phy on each data set using 32 independent runs, each for 48 hours (i.e., BAli-Phy was run somewhat longer than 2 months on each data set). In this way BAli-Phy generated many hundreds of thousands (and in several cases more than 1,000,000) MCMC samples for each data set that it analyzed and produced ESS values indicative of adequate sampling. This study used more than 230 CPU years for the BAli-Phy analyses alone in order to ensure that its performance would be measured fairly, and it provides a careful evaluation of how BAli-Phy performs on biological and simulated data sets. A separate study was done on a smaller number of simulated and biological *nucleotide* alignments to see whether the trends on protein data carried over. It also includes some more varied simulation conditions in the results. That study is covered in more detail in the following chapter, but its conclusions are relevant to our interpretation and are dicussed breifly here.

Our results show a peculiar pattern for multiple sequence alignments produced by BAli-Phy. On studies of *simulated* data, BAli-Phy's performance was superior to other methods virtually across the board, measured by both false-positive and false-negative rates, and produced alignments that were consistently close to the correct length. This held for both protein and nucleotide alignments, and it held regardless of the specific simulation model used. However, on *biological* data BAli-Phy appears to be consistently under-aligning the sequences, leading to estimated alignments that were longer than the curated reference alignments and causing accuracy metrics to show a bias toward a high pairwise false-negative rate (called SP-Score) but a low false-positive rate (called Modeler score). Again, this held for both the nucleotide and the protein alignment benchmarks. There, BAli-Phy generally had Modeler scores better than most other methods, but SP-scores that were lower than most; furthermore, BAli-Phy produced alignments that were generally longer than the reference alignment and also longer than all the other alignments. In other words, BAli-Phy tended to under-align on biological data but not on the simulated data and was visibly an outlier on the biological data in terms of alignment length. In light of this puzzling divergence it is worth noting that, upon closer examination, the limited evaluation of BAli-Phy in previous studies also suggest that biological data are troublesome. In these studies we provide a more detailed examination of the differences, but at present it is not clear why this occurs. There are several possible explanations dicussed below, and we close with some visual inspection of BAli-Phy alignments looking for additional clues.

7.2 Materials & Methods

7.2.1 Alignment Methods

We explored the following multiple sequence alignment methods: BAli-Phy v. 2.3.6, Clustal-Omega v. 1.2.4 [200], CONTRAlign v. 1.04 [50], DiAlign v. 2.2.2 [147, 74], Kalign v. 2.04 [107], MAFFT v. 7.305b [97], Muscle v. 3.8.31 [58], PRANK v. 140603 [124, 125], PRIME v. 1.1 [248], Probalign v. 1.4 [194], ProbCons v. 1.12 [51], PROMALS3D (retrieved Jan. 31, 2018, installed and run with Python 2.7.8 and GCC v. 4.7.1) [174], and T-Coffee v. 11.00.8cbe486 [160, 159, 168]. We explore two ways of running MAFFT: MAFFT-G-INS-i and MAFFT-Homologs (using the SwissProt Database [14]).

All methods other than BAli-Phy and Promals3D were performed in default mode. Promals-3D enables

structural alignment features, but we turned these off using the following sample command:

python promals < InputSequencesFile > -dali 0 -tmalign 0 -fast 0

BAli-Phy requires specific parameters (including the substitution model and the number of MCMC iterations) to be set by the user. To select a protein sequence evolution model for use in BAli-Phy, we used RAxML [206] version 8.2.9 to select the protein sequence evolution model based on likelihood scores computed on the alignment computed using MAFFT L-ins-i (see paper for details [165]). We ran 32 independent runs of BAli-Phy, each for 48 hours, discarding the first 25% of the alignments that were generated during the MCMC run, and then retaining every tenth alignment in the remaining sample. The point estimates of the alignments were computed using the posterior decoding (PD). According to the output from BAli-Phy, the vast majority of the BAli-Phy runs we performed had good ESS values, which suggests that BAli-Phy may have converged on those data; see paper for details.

7.2.2 Computational Resources

BAli-Phy and T-Coffee are the most computationally intensive methods we explored, and so these were run on the Blue Waters supercomputer at the National Center for Supercomputing Applications (NCSA); all other methods were run on the Campus Cluster at the University of Illinois at Urbana-Champaign.

7.2.3 Data Sets

Protein biological data sets. We took all the alignments from the four databases we selected (BAliBASE, Mattbench, Homstrad, and Sisyphus) that had between 4 and 25 sequences. Each alignment with more than 25 sequences was then sub-sampled to produce a data set with between 5 and 25 sequences; see paper for the protocol used for sub-sampling.

T-Coffee failed to align a number of data sets, returning empty folders; this was particularly pronounced on the BAliBase data, where 82 out of 742 alignments were not completed, although it also failed to align 2 data sets each from the other three benchmarks (see paper for discussion). BAli-Phy was able to analyze all the data sets, but on two data sets the posterior decoding algorithm failed due to the high computational complexity of having a small number of very long sequences. After eliminating the data sets where T-Coffee and the Bali-Phy posterior decoding failed to complete, we still had a large number (1192) of reference alignments from the four benchmarks. Table 7.1 presents empirical properties for the reference alignments for these 1192 data sets, including average pairwise sequence identity (PID), average sequence length, average number of sequences, average percentage gapped, and mean gap length.

Simulated data sets. We generated 120 simulated data sets (20 data sets from each of 6 different model conditions) to evaluate the alignment methods for this study. To obtain the basic model tree topology and branch lengths, we selected the 27-sequence serine protease data set from the Homstrad benchmark collection, computed a MAFFT L-ins-i alignment on the data set, and then used RAxML v8.2.9 to construct a phylogenetic tree with branch lengths (see paper for exact command). RAxML selected the WAG model

Table 7.1: Empirical properties of the 1192 reference alignments from the four biological benchmark collections. We report the average pairwise sequence identity (PID), average number of sequences, average alignment length, average fraction of the reference alignment occupied by gaps, and median gap length.

Database	PID	# seqs.	alignment length	% gapped	gap length
BAliBase	0.30	12.4	772.0	37.7	8.1
Homstrad	0.37	6.9	257.3	16.6	2.7
Mattbench	0.20	7.3	416.4	44.6	2.8
Sisyphus	0.26	9.4	172.3	21.0	4.9

[235] for this data set. We set the indel rate and the gap length distribution (a negative binomial) to match the empirical distribution for the serine protease data set. We then modified this basic model tree in two ways – by rescaling the branch lengths (by a factor of three) and reducing the indel rate – to produce six different model conditions (Table 7.2) that ranged in terms of the average percent gapped (from 18.3% to 46.4%) and average pairwise sequence identity (from 0.11 to 0.24). Hence, this process produced six different model conditions with a range of average PID and percentage gapped that cover the characteristics of the biological benchmark data sets we explored. The sequence length at the root is 200, and then sequences evolve down the model tree with indels and substitutions using the Indelible [68] simulator. We used WAG for the substitution model, and indels were generated with lengths drawn from a negative binomial distribution.

Table 7.2: Empirical properties of the true alignments for the simulated data sets, each with 27 sequences. Each submatrix represents one of the six model conditions, and the top row within each submatrix represents the mean pairwise sequence identity (PID) and the bottom row represents the percentage gapped.

			Low subst. rate	**High subst. rate**
Indel Rate	**High**	PID	0.24	0.11
		% gapped	*46.4*	*42.6*
	Medium	PID	0.24	0.11
		% gapped	*29.8*	*31.5*
	Low	PID	0.23	0.12
		% gapped	*18.3*	*19.2*

7.2.4 Evaluation Criteria

The accuracy of the estimated alignment was assessed in comparison to the reference alignment for the biological data sets, and to the true alignment for the simulated data sets. Each alignment on the same set of sequences can be represented by its set of "homology pairs", where a homology pair is a pair of residues, one from each of two different sequences, that are placed in the same column in the alignment (see [142, 232]). Two different alignments can then be compared to each other by examining the shared or unique homology pairs. Furthermore, when one alignment is treated as a reference or true alignment, then the error in an estimated alignment can be evaluated by calculating the number of homology pairs in the true alignment that are missing from the estimated alignment (i.e., the number of false negatives) as well as the number of homology pairs in the estimated alignment that do not appear in the true alignment (i.e., the number of false

positives). These error metrics are normalized to produce values between 0 and 1, which are then called the error rates. The first of these error rates is referred to as the SPFN (sum-of-pairs false negatives score) and the second is referred to as the SPFP (sum-of-pairs false positives score). Finally, the error rates can also be expressed as accuracy measures in the obvious way: 1-SPFN is a measure of recall, and is referred to as the Modeler Score, and 1-SPFP is a measure of precision, and is referred to as the SP-Score.

We also report the expansion ratio, which is the ratio of the number of sites in the estimated alignment to the number of sites in the reference or true alignment; values below 1.0 represent over-alignment (i.e., shorter alignments than the reference or true alignment) and values greater than 1.0 represent under-alignment. We used FastSP v. 1.6.0 [142] to calculate these values. Note that the classical tradeoff between the false positive rate (FPR) and false negative rate (FNR) has an analog in multiple sequence alignment as well: just as a classifier can achieve zero FPR by classifying everything as negative, an MSA can have zero SPFP if the sequences are completely unaligned (i.e., all sites in the alignment have at most one non-gap character). That is the extreme case, of course, but serves to indicate that although expansion ratios greater than 1.0 reflect under-alignment, a pattern of low SPFP and high SPFN (equivalently high Modeler score and low SP-score) is also indicative of under-alignment.

Finally, we examined the impact of alignment error on tree error. We computed maximum likelihood trees using RAxML v8.2.9 on estimated and true alignments for the simulated datasets, and then recorded the Robinson-Foulds [189] error rate (i.e., the fraction of the number of branches in the true tree that are missing from the estimated tree), computed using Dendropy [211].

The impact of pairwise sequence identity (PID) on multiple sequence alignment accuracy is well established (e.g., [220, 22, 116, 200, 230]), with alignment accuracy generally decreasing as PID decreases, and expected to be very low when PID is below 0.20 [9]. Therefore, we evaluated the impact of PID on alignment accuracy in our experiments. We grouped the sequence data sets into four bins according to the PID within each data set, with the highest PID bin (where PID is at least 0.5) expected to contain the easiest data sets to align and the lowest PID bin (where PID is at most 0.15) expected to contain the most difficult data sets to align.

7.3 Results

7.3.1 Biological Data

We began by exploring the overall accuracy of the different methods we examined with respect to Modeler score and SP-score (Fig. 7.1). The results shown are restricted to the 1192 data sets where all methods ran successfully.

There was a large range in scores on these data, with the average Modeler score varying from 0.66 to 0.80 and average SP-score varying from 0.63 to 0.77. PROMALS, T-Coffee, CONTRAlign, and MAFFT-homologs each came in the top four places for both criteria, and Kalign, DiAlign, and PRANK had the lowest overall SP-scores and Modeler scores of all the methods we tested. BAli-Phy had the top average Modeler score (0.80) but among the lowest average SP-scores of all the methods (0.67, ranking 11 out of 14); thus,

BAli-Phy's average Modeler score was substantially larger than its SP-score, a pattern that indicates under-alignment. All other methods had close average SP and Modeler scores (i.e., differences that were at most 0.04, and usually at most 0.01).

Results on the individual benchmarks for Modeler and SP-scores show similar trends (Fig. 7.2). For example, T-Coffee had the highest SP-scores on the Homstrad (0.89), Mattbench (0.78), and Sisyphus (0.80) benchmarks, and PROMALS had the highest SP-score (0.74) on the BAliBASE data (where T-Coffee had 0.71). Thus, T-Coffee was either best or close to best in terms of SP-score on these data sets. Similarly, although PROMALS was only top on the BAliBASE data sets, it came in second on the other benchmarks, where its average SP-scores were fairly close to the best score: 0.01 lower than best on the Homstrad data sets, 0.03 lower than best on the Sisyphus data sets, and 0.06 lower on the Mattbench data sets. MAFFT-Homologs had the second or third highest SP-score on all but the Sisyphus benchmark, and the third or fourth highest Modeler score on three of the benchmarks. Finally, BAli-Phy consistently ranked among the first three methods for Modeler Score (it was top on BAliBASE, in second place on Sisyphus and Mattbench, and in third place on Homstrad) and between eighth and twelfth for SP-score (Fig. 7.2). Thus, overall as well as on the individual benchmarks, BAli-Phy produced alignments with high Modeler scores and low SP-scores, a pattern that indicates under-alignment.

To better understand this trend, we examined the expansion ratios of the different alignment methods. A few methods (notably Clustal, Probcons, and Promals) had excellent expansion ratios (in the range 0.95 to 1.05) across all PID values. However, all others either under-aligned (i.e., expansion ratios greater than 1.05) or over-aligned (i.e., expansion ratios less than 0.95) for some condition (Fig. 7.3). As expected, the lowest PID condition (PID $\geq$ 0.5) was the most challenging for the remaining methods. For this condition, BAli-Phy, DiAlign, and Prank under-aligned the most, with expansion ratios 1.87, 1.31, and 1.17 respectively. The methods that over-aligned the most were Muscle (expansion ratio 0.81), Prime (expansion ratio 0.84), Kalign (expansion ratio 0.86), MAFFT-G-INS-i, MAFFT-Homologs, and CONTRAlign (all with expansion ratio 0.89). Most significantly, BAli-Phy's expansion ratio was the largest of all the methods for each bin, indicating that it under-aligned the most of all the methods and produced substantially longer alignments than all other methods. Hence, BAli-Phy was an outlier among these methods in terms of alignment length.

The remaining experiments were restricted to the top-performing alignment methods. Therefore, we exclude Kalign, DiAlign, and PRANK, each of which had among the lowest overall accuracy in terms of SP-score and Modeler score on the biological data sets.

PID also impacted the SP-score and Modeler score of the top methods, as shown in Figures 7.4-7.5. As expected, all methods had their best SP-scores and Modeler scores under the highest PID bin (i.e., when PID $\geq$ 0.5) and their scores dropped as PID decreased. The range in SP-scores and Modeler scores was narrowest for the highest PID bin (at most 0.05 difference between the largest and smallest scores) and increased as PID dropped. For example, on the highest PID bin (Fig. 7.4), the Modeler scores ranged from 0.91 (attained by T-Coffee) to 0.95 (attained by BAli-Phy), while on the lowest PID bin the Modeler scores ranged from 0.26 (Clustal) to 0.53 (BAli-Phy). Furthermore, although the Modeler scores dropped for all methods as PID dropped, this effect was smaller for BAli-Phy, Promals, and T-Coffee than for the other methods (i.e., the change in average Modeler score between the top and bottom PID bins for these three methods was at most

0.47, and all other methods had changes of between 0.59 and 0.68).

Similar trends hold for SP-score (Fig. 7.5), with the following main difference: under the low PID bin, BAli-Phy's SP-score tied for the lowest of all methods, indicating it is more impacted by changes in PID than we saw for its Modeler score (in particular, the change in BAli-Phy's SP-scores between the high and low PID bins was 0.64, which is approximately the same change as for the remaining methods other than Promals and T-Coffee). Finally, for the two bins where the differences between methods were large (i.e., the bottom two bins), T-Coffee and PROMALS had the top SP-scores.

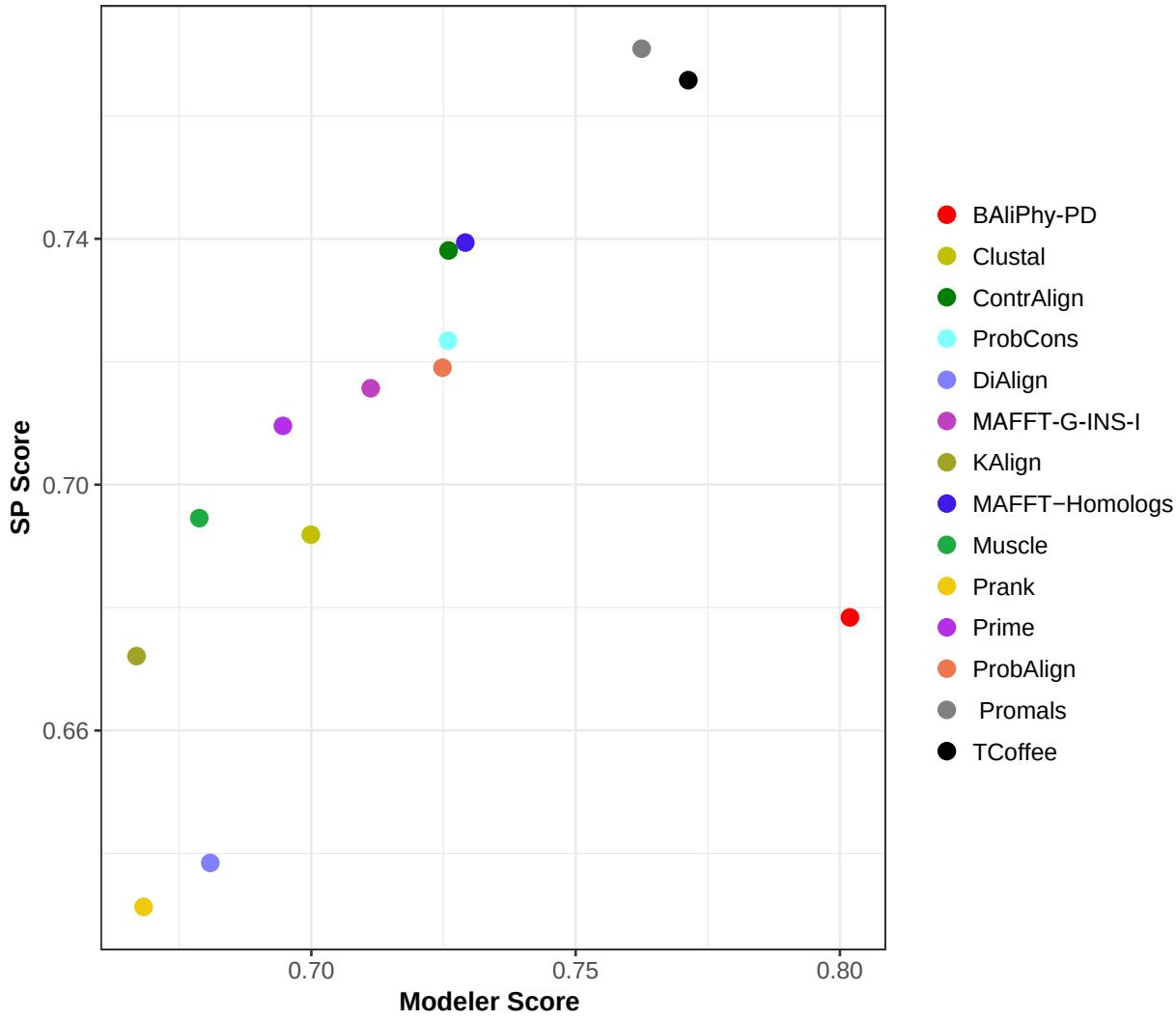


Figure 7.1: Average Modeler Score (i.e., precision) vs. average SP-score (i.e., recall) of the full set of multiple sequence alignment methods on the biological benchmark data sets, each with at least 4 and at most 25 sequences; each data point represents analyses of 1192 data sets from the four benchmark collections (658 from BAliBase, 231 from Homstrad, 202 from Mattbench, and 101 from Sisyphus). See Table 7.4 for actual numeric values.

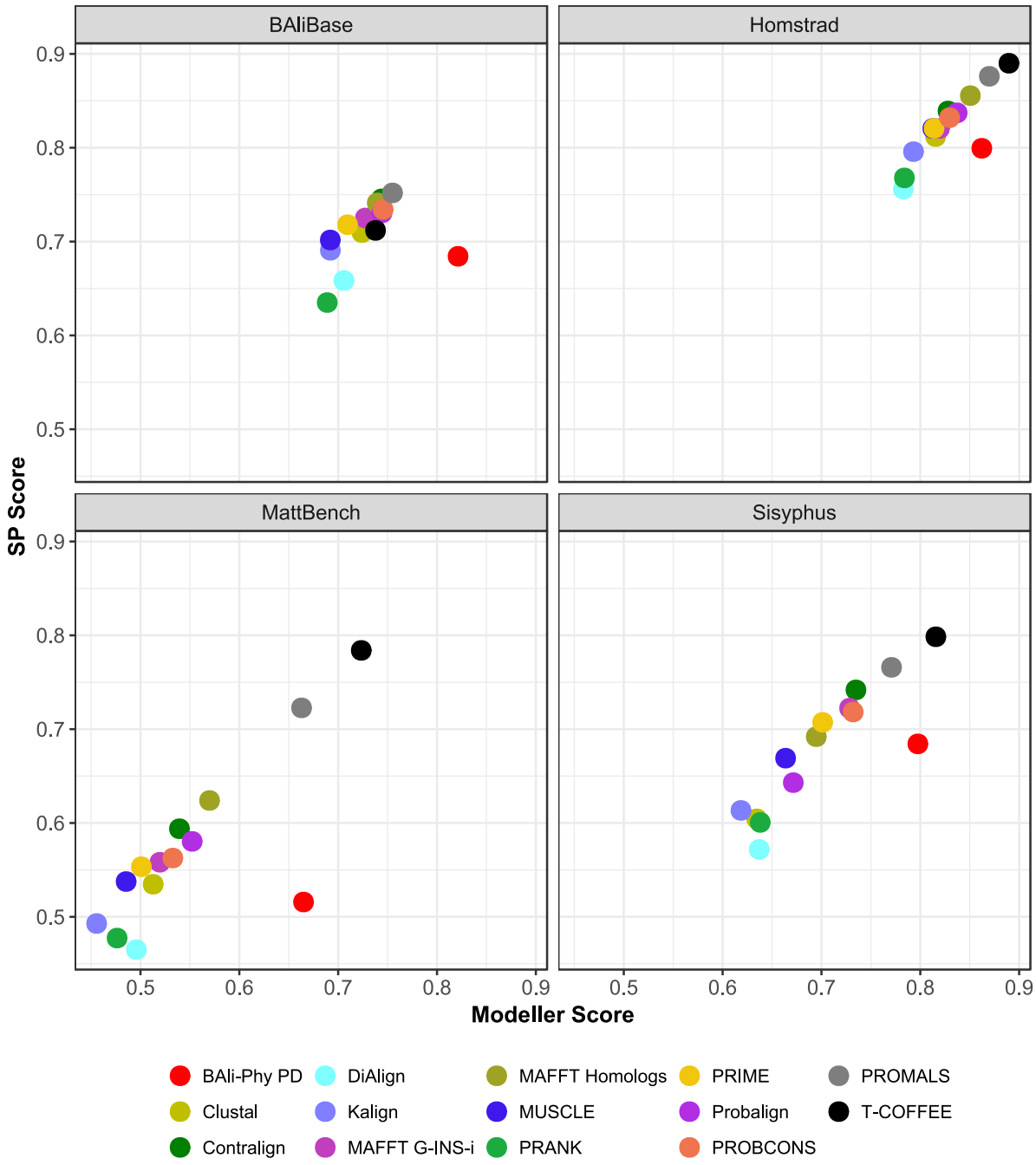


Figure 7.2: Average Modeler Score (i.e., precision) vs. SP-Score (i.e., recall) of all alignment methods on the individual biological benchmarks. Results shown are for 1192 data sets from the four benchmark collections (658 from BAliBase, 231 from Homstrad, 202 from Mattbench, and 101 from Sisyphus) See Table 7.5 for actual numeric values.

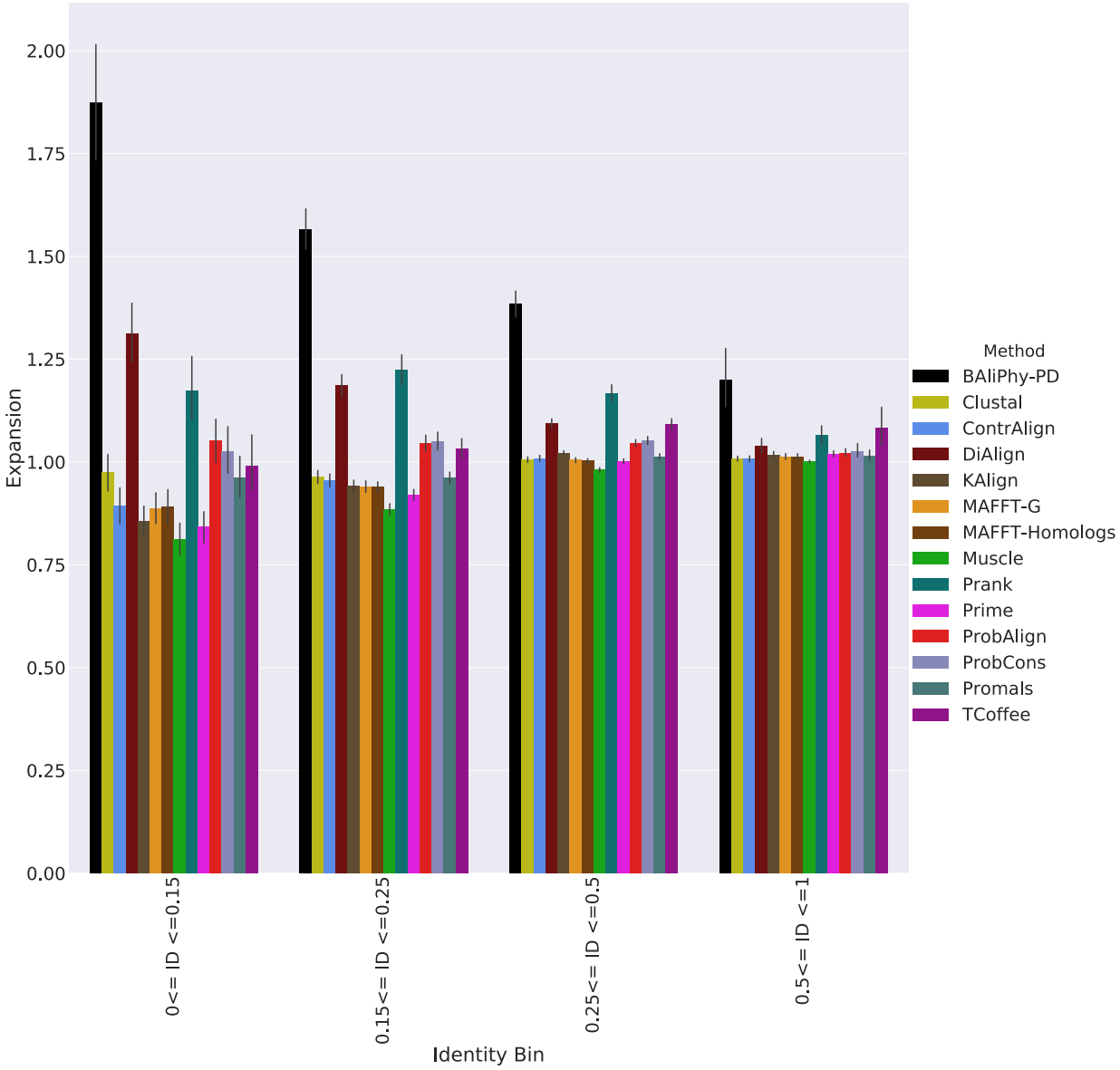


Figure 7.3: Average expansion ratios on the 1192 biological benchmark data sets, each with at most 25 sequences, by average percent ID (ID). Values more than 1.0 indicate under-alignment (i.e., longer alignments than the reference alignment), while values less than 1.0 indicate over-alignment (i.e., shorter alignments than the reference alignment). The four bins based on average sequence identity, ordered from smallest to largest, have 83, 417, 615, and 77 alignments, respectively. See Table 7.6 for actual numeric values.

7.3.2 Simulated Data

Methods that rely on gathering homologs from external databases (e.g., MAFFT-Homologs, T-Coffee and Promals) are expected to have poor accuracy on these simulated data, a prediction we confirmed (see paper for details). We therefore omit these three methods for the rest of this section, but we include PRANK since, like BAli-Phy, it is a phylogeny-aware method.

We explored the relative and absolute accuracy of the multiple sequence alignment methods on simulated

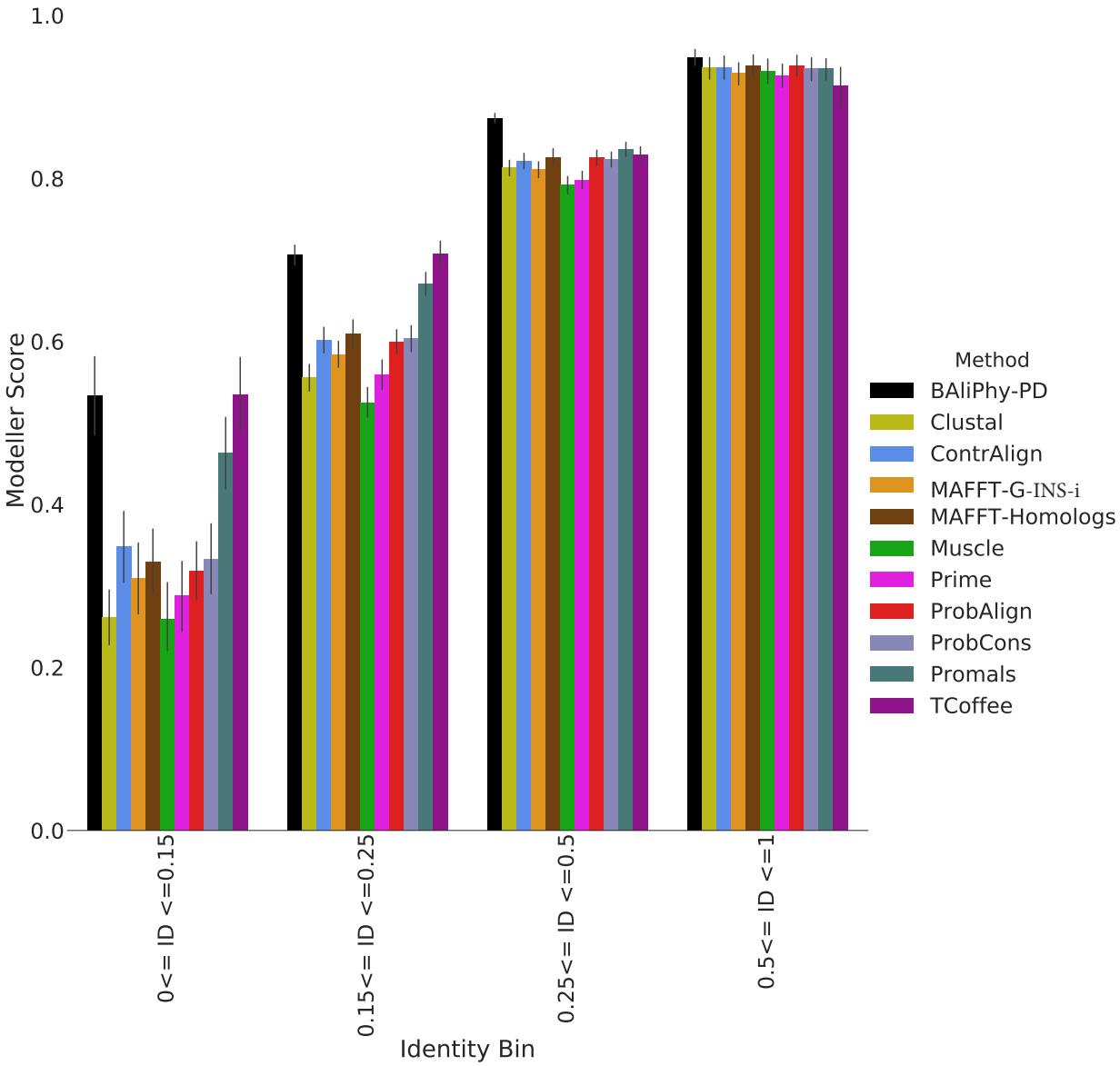


Figure 7.4: Average Modeler Scores (i.e., precision) for the top methods on the 1192 biological benchmark data sets, binned into different average pairwise sequence identity (ID) levels. The four bins based on average sequence identity, ordered from smallest to largest, have 83, 417, 615, and 77 alignments, respectively. See Table 7.7 for actual numeric values.

data sets with 27 sequences with 6 model conditions, each with 20 replicates. The accuracy of these methods varied across these six model conditions (Fig. 7.6). When both rates are low, all methods had excellent Modeler and SP-scores (i.e., at least 0.95) and the differences between them were small (e.g., the difference in score between any two methods under the easiest model condition was at most 0.02 for both criteria). However, with higher substitution rates or indel rates, the accuracy of all methods decreased and the range in scores increased.

The most striking observation on the simulated data sets is that BAli-Phy had the best accuracy of all

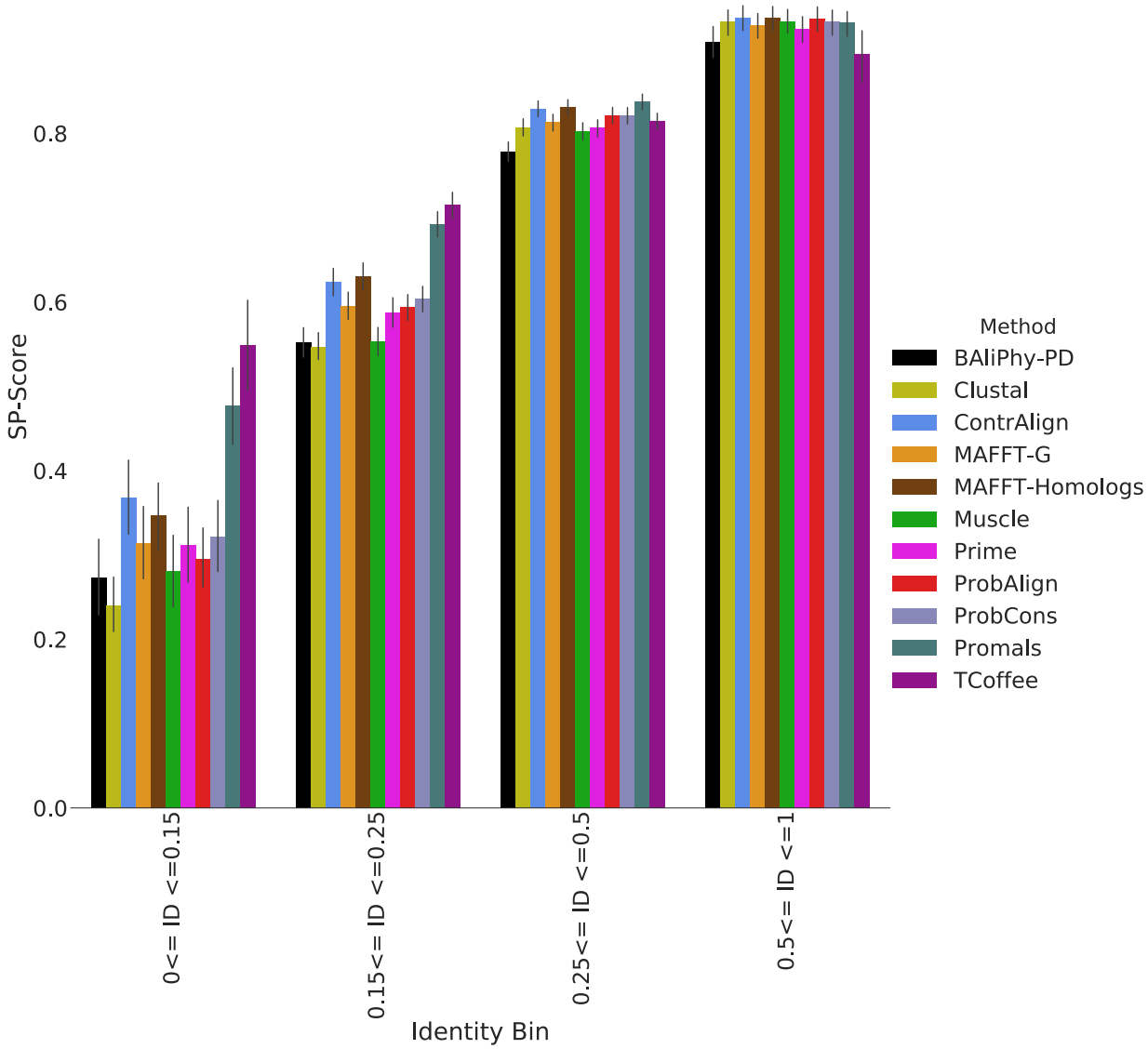


Figure 7.5: Average SP-scores (i.e., recall) for the top methods on the 1192 biological benchmark data sets, with data sets binned by average pairwise sequence identity (ID) levels. The four bins based on average PID, ordered from smallest to largest PID values, have 83, 417, 615, and 77 alignments, respectively. See Table 7.8 for actual numeric values.

methods with respect to both criteria. Furthermore, while the difference between BAli-Phy and the least accurate method was small (at most 0.02) for the easiest model condition, the difference in accuracy between BAli-Phy and the *second most accurate method* increased as the indel rate or the substitution rate increased. For example, under the most difficult model condition (where substitution and indel rates were the highest), BAli-Phy achieved average SP-score and Modeler score of 0.93; the second best SP-score was 0.87 (attained by Clustal) and the second best Modeler score was 0.84 (attained by MAFFT-G-ins-i). These are drops in accuracy in the 0.07 to 0.09 range (see Table 7.9).

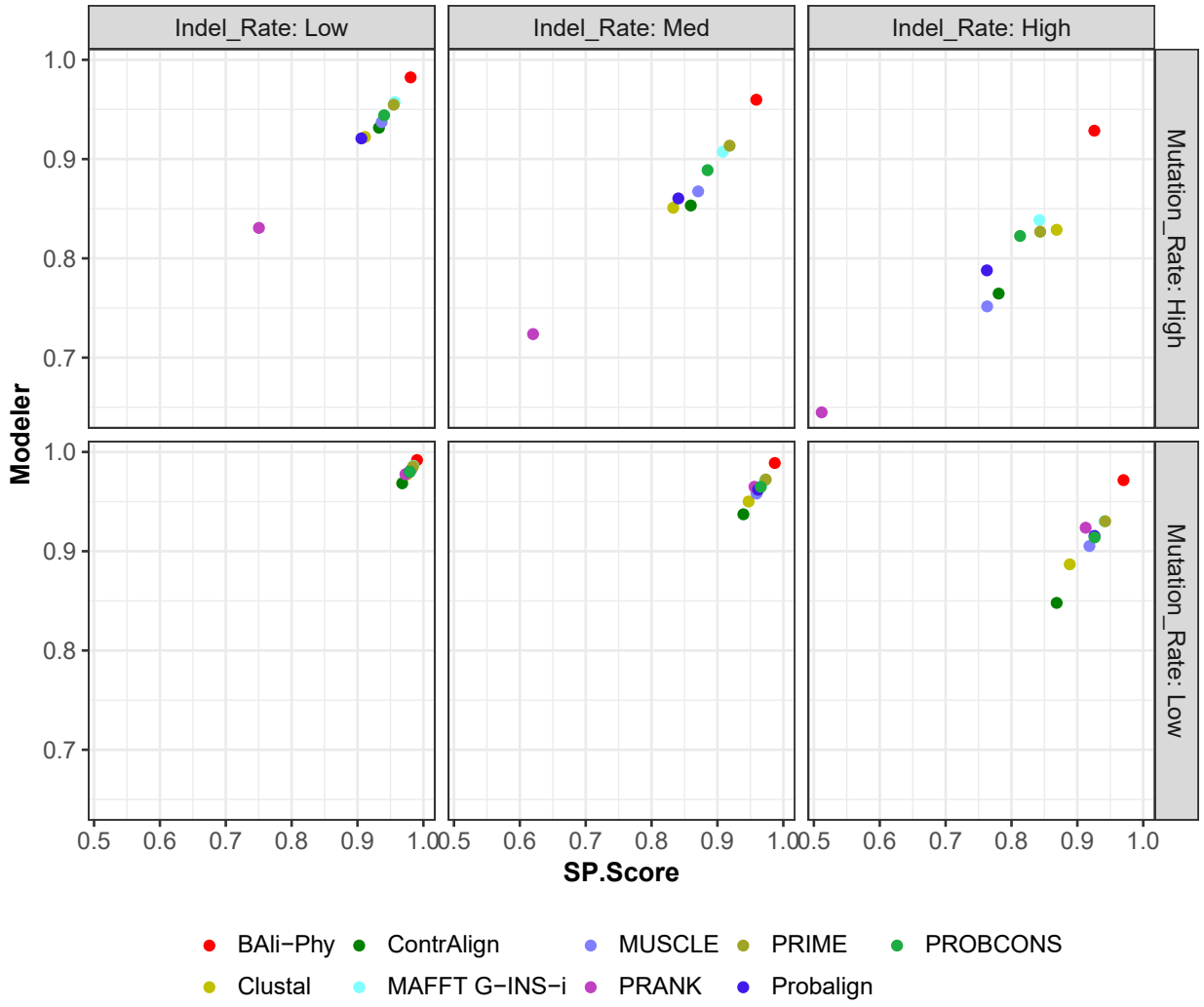


Figure 7.6: Modeler score (i.e., precision) vs. SP-Score (i.e., recall) for MSA methods on simulated amino acid data sets with 27 sequences for 6 different model conditions that vary by the substitution rate and indel rate; averages over 20 replicates are shown.

As shown in Figure 7.7, similar trends were seen with respect to expansion ratios. Results under the easiest model condition (with low mutation rates and indel rates that were at most moderate), all methods produced expansion ratios in the range 0.97-1.01 (i.e., nearly perfect). However, under the more difficult model conditions, the methods could be distinguished and we observed the following overall trends. BAli-Phy produced alignments with expansion ratios of 1.0 under all six model conditions (except when given the wrong substitution model, in which case it produced alignments with average expansion ratio 0.99). Most other methods over-aligned under difficult conditions (e.g., Muscle, MAFFT G-INS-i, CONTRAlign, Clustal, and PRIME over-aligned when mutation rates and indel rates were high, with expansion ratios less than 0.90). The three remaining methods (PRANK, Probalign, and ProbCons) showed somewhat different responses. PRANK tended to under-align (with an expansion ratio of 1.5 for the most difficult condition

where both indel and mutation rates were high) and Probalign under-aligned whenever substitution rates were high (expansion ratio of 1.06-1.08), over-aligned for the condition with high indel rates and low substitution rates (expansion ratio of 0.92), and had nearly perfect expansion ratios (in the 0.98-0.99 range) for the remaining two conditions. ProbCons had excellent expansion ratios (in the range 0.97-1.0) under five of the six conditions, but over-aligned (expansion ratio 0.91) when indel rates were high and substitution rates were low. Thus, of these methods, only BAli-Phy had consistently excellent expansion ratios under all six model conditions.

The performance of PRANK on the simulated data was generally not as strong as many of the better methods. Under the most difficult model condition, PRANK had the lowest average Modeler and SP-scores (0.65 and 0.52 respectively), and produced the longest alignments (with expansion ratio 1.5) of all the tested methods. However, under the two easiest conditions (low substitution rates with low or moderate indel rates), PRANK produced alignments that were very close to the correct length (expansion ratios between 0.96 and 1.0) and had SP-scores and Modeler scores of at least 0.96; it even achieved average SP-score and Modeler score of 0.92 and 0.93 respectively for the simulated data under the low substitution rate with high indel rate. Thus, PRANK's accuracy was impacted by the substitution rate: it was competitive with the better methods under conditions with low substitution rates but not when substitution rates were high. Also, when PRANK was given the true (model) tree as a guide tree, these scores increased (in one case by 0.10, see Table 7.9), but not enough to change its ranking within the experiment (e.g., PRANK used with the true tree was still in the bottom position for both SP-score and Modeler score under the most difficult model condition).

7.3.3 Running Time

We also did a small evaluation of the running time of a sample of the alignment methods. As noted, we always ran BAli-Phy for 48 hours on 32 independent runs, in order to improve the chances of convergence. Hence, the total running time for BAli-Phy always exceeded 2 months on each data set. In other words, the way we ran BAli-Phy is by design computationally intensive.

We selected four data sets (one from each of the benchmark collections), each containing 17 sequences. This comparison is meant to be approximate, as we used different platforms for the methods and did not ensure that all methods were run using the same environments, and only examined four data sets; hence, the results are not necessarily indicative of running time on other data sets. T-Coffee and BAli-Phy were run on the National Center of Supercomputing Applications Blue Waters supercomputer and the rest of the methods were run on the Campus Cluster at the University of Illinois at Urbana-Champaign. Some of these methods were compiled from the source code, and we used the precompiled versions for other methods.

As shown in Table 7.3, BAli-Phy was the most computationally intensive of all the methods. T-Coffee was the next most computationally intensive, using from 7 to 59 minutes on these four data sets. PROMALS and PRANK were faster than T-Coffee, but each was slow on at least one data set: PROMALS used 24 minutes on one data set and PRANK used 4 minutes on another. All the others were much faster, never using even a full minute on any of the four data sets, and several of these (i.e., DiAlign, PRIME, Clustal, Muscle, and MAFFT-G-ins-i) never exceeded two seconds on any data set.

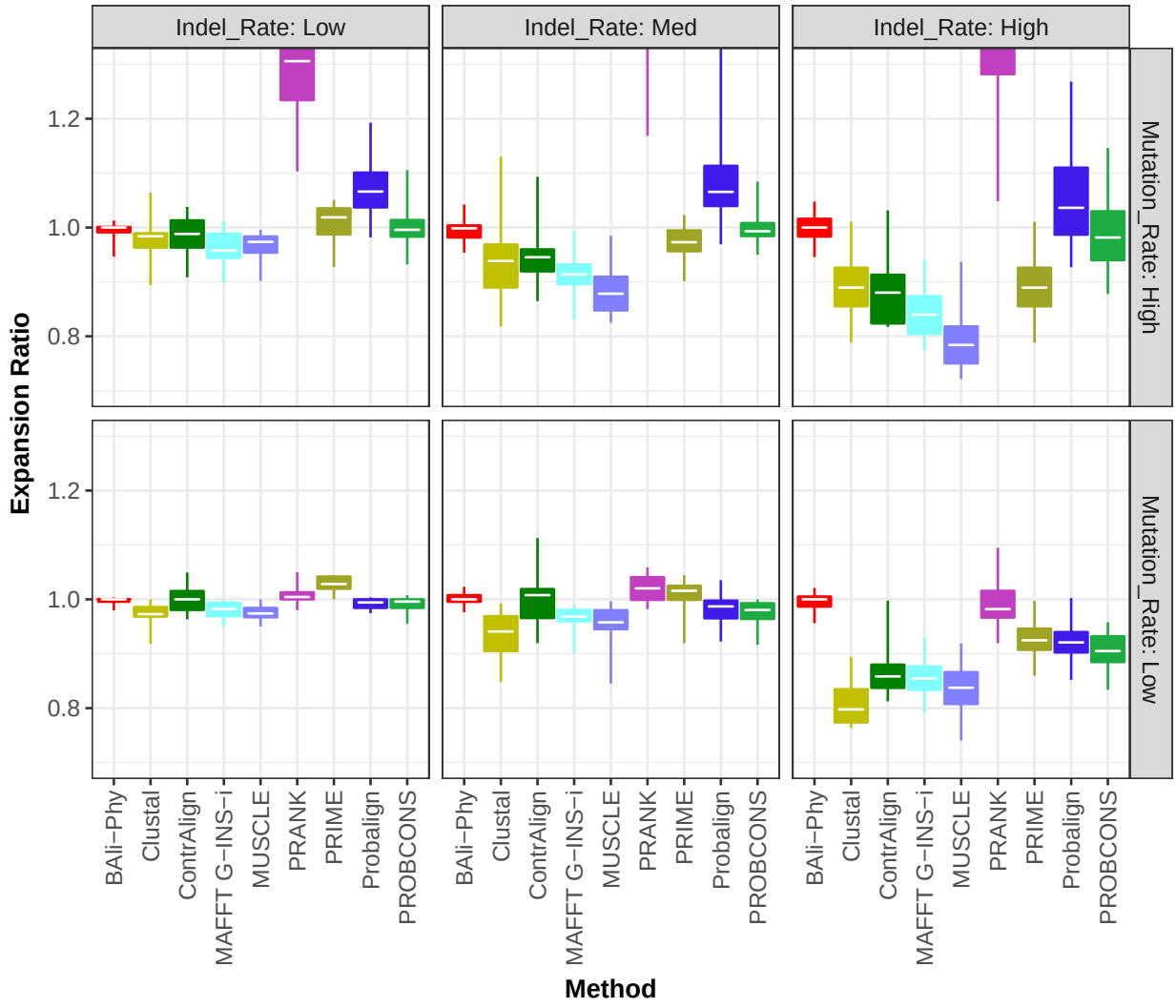


Figure 7.7: Box plot showing expansion ratios (1.0 is perfect, ratios below 1.0 indicate over-alignment, and ratios above 1.0 indicate under-alignment) for MSA methods on simulated amino acid data sets with 27 sequences for 6 different model conditions that vary by the substitution rate and indel rate; averages over 20 replicates are shown. Lines represent means and the lower and upper hinges of the box represents first and third quartiles; the upper whisker is the maximum value, and the lower whisker is the minimum value. See Table 7.4 for actual numeric values.

Although this was a limited study, the methods that were very fast on these data are likely to remain very fast for other data sets with similar characteristics (number of sequences and average sequence length), under other modern computational platforms. On the other hand, although we ran BAli-Phy 32 times, each for 48 hours, similar accuracy might have been obtained from a reduced number of hours or number of independent runs; also, the new version of BAli-Phy (v.3.1.5) may converge more quickly than the version we used (v.2.3.8). Thus, these running time values are not meant to be used to predict running time on other data sets or on other platforms, but mainly only to show that some of the better methods (e.g., MAFFT-G-ins-i) were very fast, and much faster than some of the other methods that also had very good accuracy.

Table 7.3: Running time (in seconds) information of a single 17-sequence data set in each of the biological benchmarks for a sample of the alignment methods, with methods roughly sorted by running time from fastest to slowest. The running times are rounded to the nearest hundredth of a second, and reflect wall clock time. The time reported for most methods is based on a single processor. However, BAli-Phy was run 32 independent times, each for 48 hours (in order to improve the chances of convergence) but the running time reported is for a single run; MAFFT uses 4 threads, and Clustal uses 12 threads.

Benchmark	**Mattbench**	**Homstrad**	**Sisyphus**	**BAliBASE**
Data set	*SF054*	*proteasome*	*AL00048098*	*BALBS213*
Max. Seq. Len.	*270*	*250*	*117*	*688*
DiAlign	0.0	0.0	0.0	0.0
PRIME	0.1	0.0	0.0	0.0
Clustal	0.4	0.3	0.1	1.5
Muscle	0.5	0.4	0.1	1.0
MAFFT-G-INS-i	0.7	0.7	0.3	2.0
Probalign	1.7	1.4	0.4	7.9
ProbCons	3.1	2.6	0.6	12.6
CONTRAlign	5.8	6.2	1.4	42.0
PRANK	48.5	1:16.1	9.4	4:14.7
PROMALS	14:11.5	12:22.1	5:06.2	24:03.2
T-Coffee	46:47.2	58:04.7	7:06.5	59:18.8
BAli-Phy	48:00:00.0	48:00:00.0	48:00:00.0	48:00:00.0

7.3.4 Impact on Tree Estimation

Alignment estimation is known to have an impact on tree estimation [49, 230, 139], and so we explored this issue as well. We evaluated the topological error of maximum likelihood trees computed using RAxML v8.2.9 on the true alignment and on estimated alignments. We did not explore the impact on tree error on the biological data sets because true trees are unknown, and the true species tree can differ from the true gene tree as the result of multiple biological processes, including incomplete lineage sorting [130].

We let RAxML select the protein substitution model for each data set , and report the normalized Robinson-Foulds (RF) error for the single best ML tree found by RAxML. We report the normalized RF error rates in Figure 7.8 and Delta-RF (the increase in error rate resulting from using an estimated alignment instead of the true alignment) in Table 7.10.

Under the model conditions with low mutation rates, all the methods had good accuracy, with Delta-RF error rates that were at most 1%. However, under the conditions with high substitution rates, the methods could be clearly distinguished Table 7.10. For example, under the hardest model condition (where the indel and substitution rates were both high), the Delta-RF rates were 28% for Clustal, 20% for PRANK, 9% for Probalign, 7% for ProbCons, and 4% for Muscle; in contrast, BAli-Phy and PRIME had 1% and MAFFT-G-ins-i had 0% Delta-RF rate. More generally, for all conditions, ML trees computed on the BAli-Phy, MAFFT-G-ins-i, and PRIME alignments had Delta-RF at most 1%, and so were very close in accuracy to ML trees computed on the true alignment. Thus, BAli-Phy came in among the top alignment methods with respect to topological accuracy of maximum likelihood trees computed on these alignments.

7.4 Discussion

Although our study was restricted to amino acid data sets with at most 27 sequences, the following trends were consistently observed. For both biological and simulated data and for all methods, the best Modeler and SP-scores were obtained for the high PID conditions and SP-scores and Modeler scores decreased as PID decreased. We also saw that the expansion ratios were very close to 1.0 for high PID conditions, but when PID was low the expansion ratios could be far from 1.0. Similarly, our simulation study showed that under the low substitution rate conditions (where PID was moderate at 0.24) then alignment error did not have a noteworthy impact on tree estimation (i.e., maximum likelihood trees estimated on estimated alignments were on average within 1% Robinson-Foulds error of the maximum likelihood trees estimated on the true alignment); however, under the high substitution rate conditions (where PID was low at 0.11) then maximum likelihood trees for some estimated alignments (e.g., Clustal and PRANK) were very far from the maximum likelihood tree on the true alignment. Thus, PID impacted the accuracy (measured in three different ways) of alignment methods and also had an impact on tree estimation computed on estimated alignments. This reduction in accuracy under low PID conditions explains why some biological benchmarks were more difficult than others. For example, all alignment methods had lower average Modeler and SP-scores on Mattbench than on the other benchmarks, and the average PID for the Mattbench data sets (0.20) is the lowest of the four biological collections we analyzed. Similarly, the Homstrad data sets have the highest average PID (0.37) of all these benchmarks, and the Modeler and SP-scores were highest on these data sets.

Another consistent trend throughout this study is that the differences between methods in terms of SP-score, Modeler score, and expansion ratio increased as PID decreased. Furthermore, under the high PID data sets, the differences between methods is very small, making distinctions between methods more difficult, but methods were easily distinguished on the low PID conditions. These trends suggest that the choice of alignment method may have little impact when PID is high but can be important when PID is low. The impact of PID on alignment accuracy and downstream analyses have been observed before (e.g., [22, 116, 200]), so these observations confirm prior studies.

The best performing methods on the biological data sets were typically T-Coffee and PROMALS (although the relative performance depended on the PID level and the criterion). For example, T-Coffee had the highest average SP-scores for the low PID data sets but not for the high PID data sets where PROMALS and many other methods had higher SP-scores. MAFFT-homologs and CONTRAlign also had good Modeler and SP-scores on the biological data sets, coming in the first four positions for all benchmarks. The good overall performance of MAFFT-homologs, PROMALS, and T-Coffee is noteworthy since these methods share a common strategy of recruiting homologs from an external database to use in the alignment task. Finally, BAli-Phy produced the best Modeler scores but came in at position 11 (out of 14) for its SP-score.

Results on the simulated data sets showed different trends: as they are inherently unsuited for simulated data, T-Coffee and PROMALS were not among the better methods for SP-score or Modeler score, and BAli-Phy had better scores than all the other methods for both criteria. Hence, the relative performance of methods seems to depend on PID, the criterion (i.e., Modeler score or SP-score), and – to some extent – whether the data were biological or simulated. In particular, our study shows that BAli-Phy, a leading statistical method

for co-estimating alignments and trees, had the best Modeler scores and SP-scores of all the methods we examined on simulated data sets but lower SP-scores than many methods on the biological data sets.

To understand this difference in performance, it is helpful to consider the tendency of methods to either under-align (i.e., produce alignments that are longer than the reference alignment) or over-align (i.e., produce alignments than are shorter than the reference alignment). Our study shows that that many methods tended to over-align (producing expansion ratios substantially less than 1.0) under challenging conditions; the major exceptions to this were BAli-Phy (which under-aligned the most of all methods), DiAlign, and PRANK (some other methods also under-aligned but to lesser degrees). Interestingly, in contrast to the other alignment methods, BAli-Phy never under-aligned on the simulated data, even under the most challenging conditions. Under-alignment is also demonstrated by higher Modeler scores than SP-scores, a trend consistently demonstrated by BAli-Phy on the biological data (where the overall gap was 0.13), but never on the simulated data (where BAli-Phy had average Modeler and SP-scores that were within 0.01 for every model condition). In other words, our data show that BAli-Phy under-aligned on the biological data with respect to the reference alignment, but did not under-align on the simulated data with respect to the true alignment. The fact that BAli-Phy under-aligned on biological data but not on simulated data explains the change in performance for BAli-Phy between biological and simulated data.

The performance of PRANK in our study is interesting to consider, since PRANK is designed to be "phylogeny-aware", and so has some similarities to BAli-Phy in terms of approach. On biological data Prank produced slightly larger Modeler scores than SP-scores (but on average within 0.04 of each other); on the simulated data Prank also produced larger Modeler scores, but the gap was larger (0.15), at least for the most difficult model condition. Prank under-aligned on both simulated and biological data, but the degree to which it under-aligned was larger on the simulated data. Thus, like BAli-Phy, PRANK tended to under-align on the biological data and responded differently to the biological and simulated data. However, PRANK was not competitive with the better methods in our study on either the biological or simulated data for any criterion, while BAli-Phy generally had the best (or close to the best) Modeler scores under all conditions, and only had reduced SP-scores on the biological data. As we have seen, PRANK had very good accuracy (even if not the best accuracy) under conditions with high PID, but relatively poor accuracy (compared to the better methods) under the low PID conditions, such as occur under high rates of evolution.PRANK's reduced accuracy on the simulated data sets under higher rates of evolution is perhaps surprising, given that PRANK had superior alignment accuracy in prior simulation studies [125]. However, a careful examination of [125] reveals that the simulation conditions in which PRANK provided outstanding accuracy had substitutions operating under the simplest model (Jukes-Cantor with a strict molecular clock), which may have favored PRANK in some way.

Conclusions

Statistical sequence alignment, and in particular statistical co-estimation of multiple sequence alignments and phylogenetic trees under phylogenetic models of sequence evolution, has been considered by many to be the most rigorous approach to alignment estimation. This study examined the accuracy of BAli-Phy, a

leading method for statistical co-estimation of alignments and trees, on both biological and simulated data under a range of model conditions, and explored the impact of alignment accuracy on tree accuracy.

Our study shows that BAli-Phy has the best (or close to best) accuracy of all methods for all criteria we examined (SP-score, Modeler score, expansion ratio, and accuracy of maximum likelihood trees estimated on the alignment) for the simulated data; however, BAli-Phy under-aligns on biological data to a sufficient extent that its overall SP-score drops to 11th place (out of 14) even though its Modeler score remains in top place. In other words, BAli-Phy has superior accuracy on simulated data but mixed accuracy on biological data caused by under-alignment. We do not know why BAli-Phy has this difference in performance between biological and simulated data.

Understanding this distinction in performance requires some care as there are multiple possible explanations, including the distinctions between evolutionary and structural alignments, the potential for model misspecification between the model assumed in BAli-Phy and how proteins evolve, and the possibility that reference alignments could have errors [9, 89]. Furthermore, each explanation is potentially valid, and each may contribute to a greater or lesser degree to this distinction in performance. The study in the following chapter provides some variety of model conditions, but it is ultimately a phylogenetic evolution at work so model misspecification is still limited.

The first potential explanation is that the reference alignments in the biological benchmarks are accurate as structural alignments but not as evolutionary alignments, which is feasible to the extent that structural similarity (sometimes referred to incorrectly as "structural homology") is different from "evolutionary homology" (see [186, 89, 37, 27] for discussion). If this is the major cause for this discordance, then BAli-Phy should be highly suitable for alignments used in phylogenetic inference, but perhaps not as suitable for alignments used to predict protein structures.

The second potential explanation is that the reference alignments are accurate evolutionary alignments, but the model assumed by BAli-Phy is a poor match to the true model under which the proteins evolve. There are many critiques of sequence evolution models used in phylogeny estimation [241, 113] and in simulation studies [89, 26], with two of the major concerns being the assumption that the sites evolve identically and independently (the *i.i.d.* assumption) and without any selection occurring. Although the model underlying BAli-Phy is more complex than the standard models discussed in these papers in that it addresses insertions and deletions (i.e., indels) rather than only substitutions, the BAli-Phy model nevertheless also has those two problematic features (*i.i.d.* site evolution and no selection operating) that are clearly violated by protein sequence evolution. If the degree of misspecification between the model in BAli-Phy and how proteins actually evolve is sufficient to explain much of the distinction in performance between BAli-Phy on biological and simulated data sets, then phylogeny estimation under standard models may also be impacted since many genomic regions (e.g., protein-coding sequences) are acted on by selection and evolve under processes that are not *i.i.d.*

Finally, it is possible that the reference alignments for the biological benchmarks are insufficiently accurate. This might occur if the reference alignments themselves have false positive homologies (i.e., are over-aligned) since then the true alignment would have a high Modeler score and a low SP-score with respect to the reference alignment (which is what we see with BAli-Phy on the biological data sets). Since

we observed the same general trends across all four benchmark collections, whatever the issues are they are likely to be impacting each collection rather than just one. If some of the reference alignments are incorrect, then more accurate structural alignments would need to be developed in order to provide reliable benchmarks for evaluating protein alignment methods.

Investigating these different possible explanations will require additional study. For example, the impact of model misspecification could be explored using simulations in which the various assumptions of the stochastic model assumed in BAli-Phy could be violated. Table 7.9 includes an initial evaluation of the impact of model misspecification of the substitution model (which shows that using JTT instead of WAG can reduce SP-score or Modeler score but not even by 0.01), but other types of model misspecification are likely to be more impactful. For example, selection is clearly relevant to protein sequence evolution, and so simulating under sequence evolution models with varying degrees and types of selection could potentially reveal the degree to which selection complicates alignment estimation. Similarly, heterotachy [123, 217, 252], where sites evolve independently not under identical models, is also expected to be present in many data sets and may complicate the inference of alignment using the stochastic sequence evolution model within BAli-Phy. Fortunately, some simulation tools have been developed that could be used for such studies, as described in [11, 73]. Determining whether the reference alignments in these biological benchmarks have errors will depend on experimental data that provide structural features of folded proteins as well as on alignments of multiple protein structures, and so may require new computational methods. Thus, determining the relative contribution of each of these possible explanations will require substantial effort, and should be the focus of future research.

Other directions for future work include examining these questions on larger data sets. Our study examined BAli-Phy version 2.3.8, but a new version has been developed (version 3.1.5) that is faster and uses reduced memory compared to the version we studied, and is designed to handle larger data sets; therefore, any subsequent evaluation of these issues on larger sequence data sets should use this new version. The questions we raise here are also relevant to RNA and DNA sequence evolution, and so future work should examine how statistical alignment methods perform compared to other methods on nucleotide data sets with structural alignments and also on simulated nucleotide data sets.

Abridged Supplementary Information

The full supplementary material is available along with the publication in which this chapter first appeared [165]. A selection of the contents of that document are provided below, as they are directly referred to in the text above.

Table 7.4: Results for Figure 7.1 above. We show overall average SP-scores and Modeler scores on the biological data sets. We also show the difference between the Modeler score and SP-score for each method ("Score-Diff"), noting that large positive differences indicate under-alignment. Finally, we show the size of the range of scores ("Max Diff").

Method	SP-score	Modeler score	Score-Diff
BAli-Phy	0.67	0.80	0.13
Clustal	0.68	0.69	0.01
CONTRAlign	0.73	0.72	0.01
DiAlign	0.63	0.67	0.04
MAFFT-GINSI	0.71	0.70	0.01
Kalign	0.66	0.66	0.00
MAFFT-Homologs	0.73	0.72	0.01
Muscle	0.69	0.67	0.02
Prank	0.62	0.66	0.04
Prime	0.70	0.69	0.01
Probalign	0.71	0.72	0.01
Probcons	0.72	0.72	0.00
PROMALS	0.77	0.76	0.01
T-Coffee	0.77	0.77	0.00
Max Diff	*0.15*	*0.14*	

Table 7.5: Results for Figure 7.2 above. We show average SP-scores and Modeler scores on the individual biological benchmarks.

	BAliBASE		Homstrad		Mattbench		Sisyphus	
	SP	Mod	SP	Mod	SP	Mod	SP	Mod
BAli-Phy	0.67	0.81	0.80	0.86	0.51	0.66	0.69	0.80
Clustal	0.70	0.71	0.81	0.82	0.53	0.51	0.61	0.64
CONTRAlign	0.73	0.73	0.84	0.83	0.59	0.54	0.74	0.74
DiAlign	0.65	0.70	0.76	0.78	0.46	0.49	0.58	0.64
MAFFT G-INS-i	0.71	0.72	0.82	0.82	0.56	0.52	0.72	0.73
Kalign	0.68	0.68	0.80	0.80	0.49	0.45	0.62	0.62
MAFFT Homologs	0.73	0.73	0.86	0.85	0.62	0.57	0.69	0.70
MUSCLE	0.69	0.68	0.82	0.82	0.54	0.48	0.67	0.67
PRANK	0.62	0.68	0.77	0.79	0.48	0.47	0.61	0.64
PRIME	0.71	0.70	0.82	0.82	0.55	0.50	0.71	0.70
Probalign	0.72	0.73	0.84	0.84	0.58	0.55	0.65	0.67
PROBCONS	0.72	0.74	0.84	0.83	0.56	0.53	0.72	0.73
PROMALS	0.74	0.75	0.88	0.87	0.72	0.66	0.77	0.77
T-Coffee	0.71	0.74	0.89	0.89	0.78	0.72	0.80	0.82

Table 7.6: Results for Figure 7.3 above. We show Expansion ratios on the biological data sets, partitioned by PID (pairwise sequence identity) into four bins. We also show the maximum difference between scores within each bin.

PID range:	0-0.15	0.15-0.25	0.25-0.5	0.5-1.0
BAli-Phy	1.87	1.56	1.38	1.20
Clustal	0.97	0.96	1.01	1.01
CONTRAlign	0.89	0.96	1.01	1.01
ProbCons	1.02	1.05	1.05	1.03
DiAlign	1.31	1.19	1.09	1.04
MAFFT-GINSI	0.89	0.94	1.00	1.01
KAlign	0.86	0.94	1.02	1.02
MAFFT-Homologs	0.89	0.94	1.00	1.01
Muscle	0.81	0.88	0.98	1.00
Prank	1.17	1.22	1.17	1.06
Prime	0.84	0.92	1.00	1.02
ProbAlign	1.05	1.05	1.05	1.02
Promals	0.96	0.96	1.01	1.01
T-Coffee	0.99	1.03	1.09	1.08
Max Diff	*1.03*	*0.64*	*0.38*	*0.20*

Table 7.7: Results for Figure 7.4 above (we also show DiAlign, Kalign, and PRANK). We show Modeler scores on the biological data sets, partitioned by PID (pairwise sequence identity) into four bins. We show Δ for each method, which is the difference between the average scores attained on the lowest and highest PID bins. We also show the maximum difference ("Max Diff") between scores within each bin.

PID range:	0-0.15	0.15-0.25	0.25-0.5	0.5-1.0	Δ
BAli-Phy	0.53	0.71	0.87	0.95	0.42
Clustal	0.26	0.56	0.81	0.94	0.68
CONTRAlign	0.35	0.60	0.82	0.94	0.59
ProbCons	0.33	0.60	0.82	0.93	0.60
DiAlign	0.28	0.54	0.79	0.92	0.64
MAFFT-GINSI	0.31	0.58	0.81	0.93	0.62
Kalign	0.23	0.51	0.79	0.93	0.70
MAFFT-Homologs	0.33	0.61	0.83	0.94	0.61
Muscle	0.26	0.53	0.79	0.93	0.67
Prank	0.25	0.51	0.78	0.92	0.67
Prime	0.29	0.56	0.80	0.93	0.64
ProbAlign	0.32	0.60	0.83	0.94	0.62
Promals	0.46	0.67	0.84	0.93	0.47
T-Coffee	0.53	0.71	0.83	0.91	0.38
Max Diff	*0.30*	*0.20*	*0.08*	*0.04*	

Table 7.8: Results for Figure 7.5 (we also show DiAlign, Kalign, and PRANK). We show SP-scores on the biological data sets, partitioned by PID (pairwise sequence identity) into four bins. We show Δ for each method, which is the difference between the average scores attained on the lowest and highest PID bins. We also show the maximum difference ("Max Diff") between scores within each bin.

PID range:	0-0.15	0.15-0.25	0.25-0.5	0.5-1.0	Δ
BAli-Phy	0.27	0.55	0.78	0.91	0.64
Clustal	0.24	0.55	0.81	0.93	0.69
CONTRAlign	0.37	0.62	0.83	0.94	0.57
ProbCons	0.32	0.60	0.82	0.93	0.61
DiAlign	0.20	0.48	0.76	0.91	0.71
MAFFT-GINSI	0.31	0.60	0.81	0.93	0.62
KAlign	0.24	0.52	0.79	0.92	0.68
MAFFT-Homologs	0.35	0.63	0.83	0.94	0.59
Muscle	0.28	0.55	0.80	0.93	0.65
Prank	0.22	0.47	0.75	0.91	0.69
Prime	0.31	0.59	0.81	0.92	0.61
ProbAlign	0.30	0.59	0.82	0.94	0.64
Promals	0.48	0.69	0.84	0.93	0.45
T-Coffee	0.55	0.71	0.81	0.89	0.34
Max Diff	*0.35*	*0.24*	*0.09*	*0.05*	

Table 7.9: Partial results from above, showing SP-score, Modeler score, and Expansion ratio on the simulated datasets with high substitution rates and indel rates (i.e., the most difficult model condition). The data were simulated under the WAG model; therefore, results under the JTT model represent a test of model misspecification (albeit one where the model misspecification is small). We also show results with PRANK run on the true (model) tree as the guide tree to evaluate the impact of providing PRANK with the true tree instead of the guide tree estimated by PRANK. As shown here, PRANK improves when given the true tree, but even so has the lowest SP-score and Modeler score, and still under-aligns more than any other method.

Method	SP-score	Modeler score	Expansion Ratio
BAli-Phy (JTT)	0.93	0.93	1.00
BAli-Phy (WAG)	0.93	0.93	1.00
Clustal	0.87	0.83	0.89
CONTRAlign	0.78	0.77	0.88
MAFFT G-INS-i	0.85	0.84	0.84
MUSCLE	0.77	0.75	0.79
PRANK	0.52	0.65	1.50
PRANK (on true tree)	0.61	0.70	1.28
PRIME	0.85	0.83	0.89
Probalign	0.77	0.79	1.06
PROBCONS	0.82	0.82	0.99

Table 7.10: The difference in Robinson-Foulds (RF) tree error rates (i.e., Delta-RF) on simulated datasets, for the three model conditions with high substitution rates. Each model condition (MC) is indicated by the indel rate (Low, Medium, or High). The Delta-RF error is the difference in mean RF error rate between the maximum likelihood tree on the given alignment and on the true alignment, as computed using RAxML.

	Low	Medium	High
BAli-Phy	0.00	0.01	0.01
Clustal	0.01	0.07	0.28
CONTRAlign	0.00	0.05	0.05
MAFFT-G-INS-i	0.00	0.01	0.00
Muscle	0.00	0.03	0.04
Prank	0.06	0.17	0.20
Prime	0.00	0.01	0.01
Probalign	0.02	0.05	0.09
Probcons	0.01	0.03	0.07

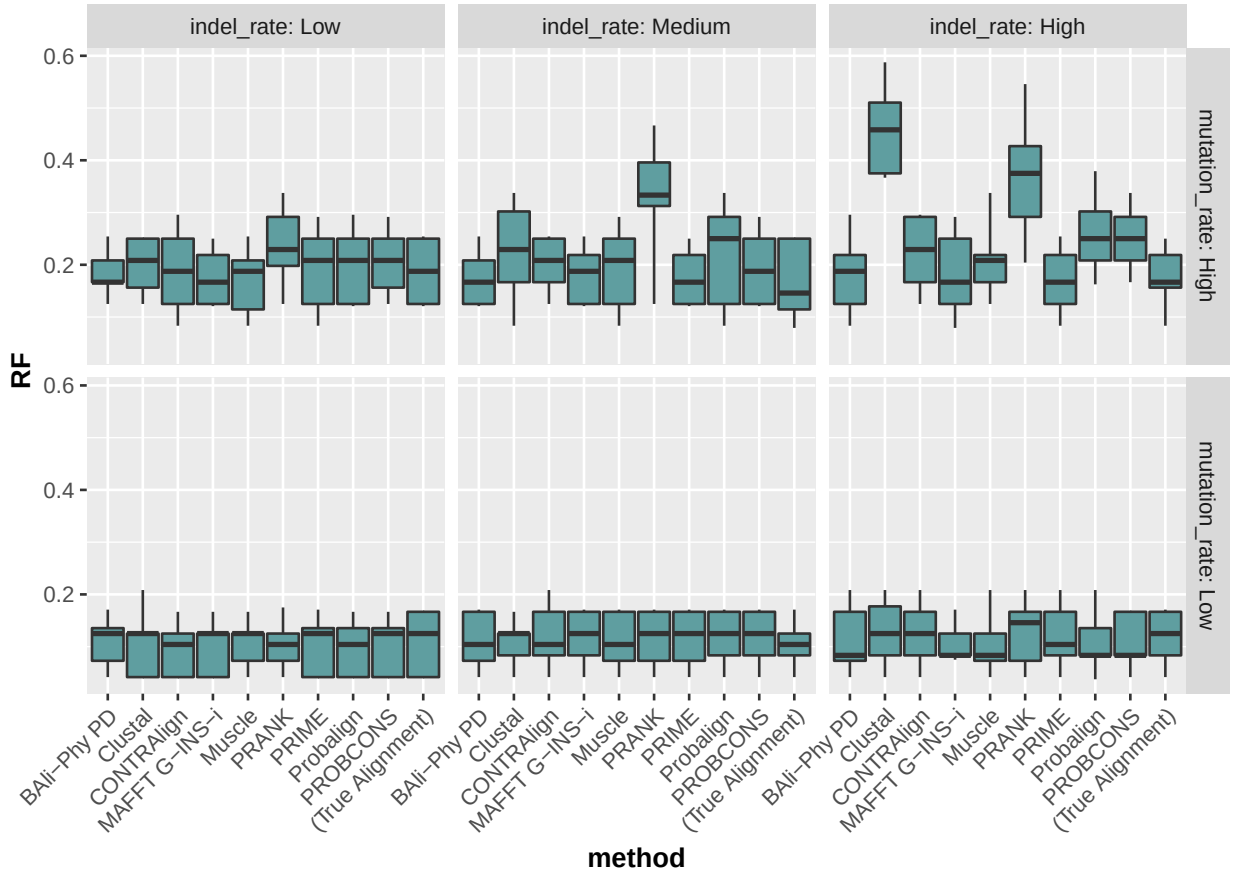


Figure 7.8: Box plot of the Robinson-Foulds tree error rates (maximum is 1.0) on the simulated datasets of maximum likelihood trees computed using RAxML on various alignments, including the true alignment. Lines represent medians and the lower and upper hinges of the box represents first and third quartiles; the upper whisker represents the 90th percentile and the lower whisker represents the 10th percentile.

Part IV

Metagenomic Analysis in the Presence of Biodiversity

Chapter 8

HIPPI

8.1 Introduction[1]

The assignment of newly obtained molecular sequences to gene families or protein families and superfamilies is a fundamental step in many bioinformatics analyses. For example, newly discovered protein sequences are assigned to protein families or superfamilies to enable functional annotation [77, 244]. Similarly, sequences obtained in shotgun sequencing analyses of environmental samples are often assigned to gene families in order to perform marker-based taxonomic identification [30, 114, 198, 157, 152]. This assignment is very difficult when the query sequence is very short (a typical problem for transcriptomic and metagenomic datasets), or when the query sequence shares little sequence similarity to any of the sequences in published databases [195]. Therefore, improving the precision and recall of methods that classify sequences into existing families is an area of active research.

Techniques for protein family classification and gene binning operate in two basic steps: first, the query sequence is compared to each family in a published database and the probability of membership in the family is assessed; then, the family with the highest probability is returned for that query sequence, provided the probability is above a required minimum threshold. The first step of this process thus uses tools for homology detection. One of the simplest methods for homology detection is BLAST [5], including variations designed specifically for proteins, such as blastp and PSI-BLAST [6]. While these sequence similarity-based approaches have good accuracy in many conditions, they can have poor accuracy when classifying query sequences that have low sequence similarity to all the sequences in the reference database [203].

A different approach is to represent each family as a single profile hidden Markov model (HMM) and assign the query sequence to the family whose profile HMM returns the highest bit score for the query sequence [54]. For example, this approach is used to assign query sequences to Pfam [66] protein families and superfamilies, where the HMMER [65, 55] software suite is used to build HMMs and assign query sequences to the best fitting Pfam family. Additionally, it has been used in several other applications, including the identification of viral families from metagenomic samples [202].

A related approach is the HHsearch pipeline [203], which is packaged with HHblits [187] in the publicly available HHpred server [204]. This pipeline takes the query sequence and finds a set of homologous

[1]This chapter contains material previously published in [158], which was a joint work with Nam Nguyen, Siavash Mirarab and Tandy Warnow. The work described in the chapter, including development of the algorithm, programming and execution of the experimental study, was conducted by the author jointly with Nam Nguyen. The work was directed and reviewed by Siavash Mirarab and Tandy Warnow, both of whom also contributed to the writing. Figures 8.1 and 8.2 were drawn by Nam Nguyen. Permission to reprint has been provided by the copyright holders.

sequences from a reference protein database. It then builds an HMM on a multiple sequence alignment of the query sequence and its homologous sequences, aligns the HMM against the individual HMMs built on each of the protein families, and finally assigns the query sequence to the family with the best score. The HHsearch pipeline has been shown to perform well on remote homology detection [175]. Some methods use a collection of profile HMMs (which we call ensembles of HMMs) to represent a protein family, including T-HMM [180] and SCI-PHY [30]. T-HMM takes as input an alignment and a tree constructed on the alignment. It then builds its ensemble of HMMs by building a profile HMM at every node in the tree. Similarly, SCI-PHY takes as input an alignment and builds a tree from the alignment. From the tree, SCI-PHY constructs a profile HMM on a subset of the clades in the tree. In both of these methods, the profile HMMs are based upon clades in their respective trees. These approaches have shown that using an ensemble of HMMs to represent a protein family has resulted in improved protein family identification for remote homologs, improved subfamily identification, and improved orthology detection [30, 180, 1, 103, 181].

We have developed similar methods for representing a multiple sequence alignment with an ensemble of HMMs. These methods include SEPP [140], TIPP [157], and UPP [156] (collectively referred to as the *PP methods). One of the key difference between the *PP methods and previous ensemble approaches (i.e., T-HMM and SCI-PHY) is that the profiles generated by the *PP methods are based upon a recursive subdivision of an input tree into approximately equally-sized subtrees (referred to as a "centroid decomposition"), resulting in fewer profile HMMs, without requiring the profile HMMs to be based upon clades. In the UPP study [156], the authors found that a clade-based decomposition created a more computationally intensive process without resulting in improved accuracy. The ensembles of HMMs generated by the *PP methods have been shown to improve the accuracy of phylogenetic placement, taxon identification of short metagenomic reads, and multiple sequence alignment estimation.

Here, we present a new method, HIPPI (**HI**erarchical **P**rofile HMMs for **P**rotein family **I**dentification; see Fig. 8.1 for an overview of the algorithm), that classifies query amino acid sequences into protein families and superfamilies. HIPPI modifies the previous *PP methods to address the problem of family selection. As in the *PP methods, HIPPI builds an ensemble of profile HMMs to represent each protein family, but it improves on earlier techniques for building ensembles of profile HMMs by changing the dataset decomposition strategy to take pairwise sequence identity into account. Given a query sequence q, HIPPI scores q against every profile HMM in every ensemble of profile HMMs built from the protein families. Finally, HIPPI assigns q to the protein family whose ensemble of HMMs has the profile HMM that reports the best bit score for the query sequence. This is a general approach that can be applied to any collection of protein families.

In our study, we present a comparison between our approach, HMMER, blastp, and the HHblits+HHsearch pipeline (referred to herein as "HHsearch") for the problem of protein family identification using the Pfam-A database of protein families [66]. As we will show, HIPPI provides greater precision and recall than the other methods. Furthermore, HIPPI has substantially better precision and recall than the other methods under the most challenging conditions where the protein family has low average sequence identity and the query sequence is fragmentary.

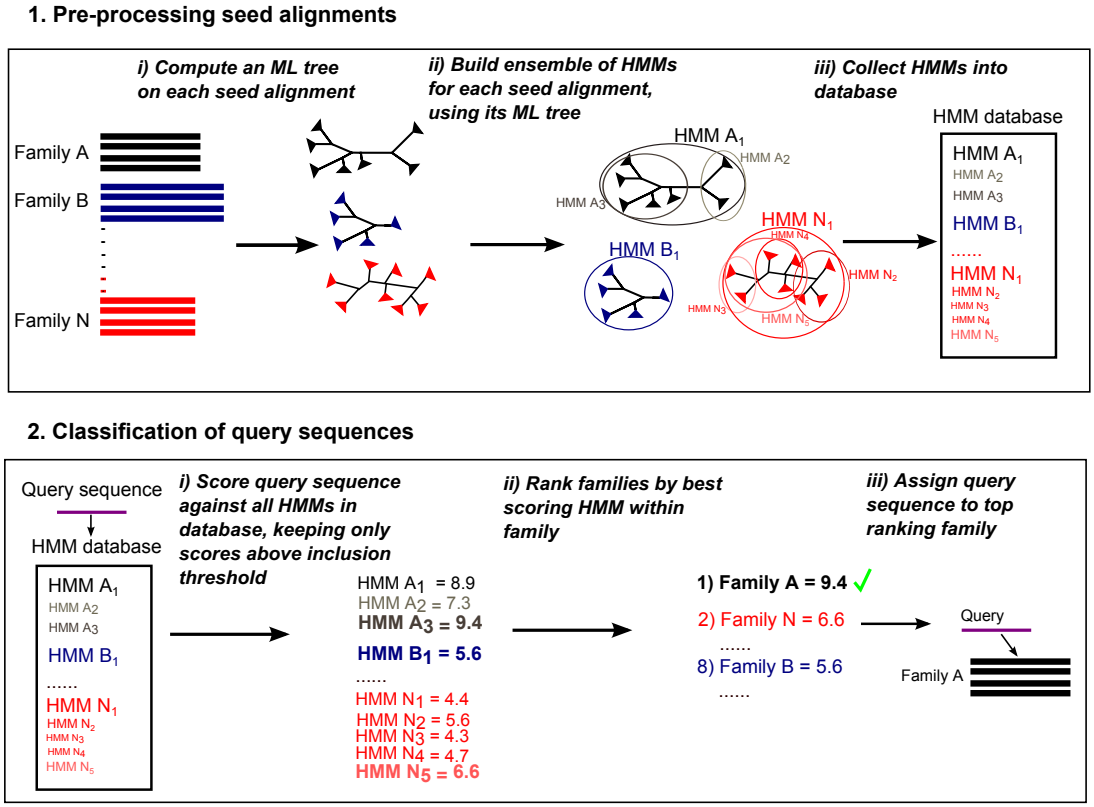


Figure 8.1: Overview of the HIPPI algorithm. The input is a collection of seed alignments. Box 1 shows the preprocessing phase of building the ensemble of HMMs database from the seed alignments. Box 2 shows the classification phase of using the ensemble of HMMs database to classify the query sequences. See Fig. 8.2 for details of how the ensemble of HMMs is constructed.

8.2 HIPPI

The input to the HIPPI algorithm is a database $\mathcal{D} = \{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_k\}$ of protein families and a query sequence q. Every family $\mathcal{F}_i$ in the database includes a known (curated) seed alignment. The key feature of HIPPI is in the preprocessing step described below, where the method builds an ensemble of HMMs for each family, based on an estimated maximum likelihood (ML) phylogeny for the family.

8.2.1 Preprocessing

In the preprocessing step (Box 1 in Fig. 8.1), we construct an ensemble H_i of HMMs for each protein family $\mathcal{F}_i$ in the database $\mathcal{D}$. Here we show how we compute H_i, given $\mathcal{F}_i$.

1. **Step 1:** We estimate a maximum likelihood tree T_i for the seed alignment for $\mathcal{F}_i$, using FastTree-2 [179].

2. **Step 2:** We build the collection of profile HMMs from the seed alignment and ML tree as follows (Fig. 8.2). Starting with the initial seed alignment, we build a profile HMM on the entire alignment and add the profile to our ensemble of HMMs. Next, if the seed alignment has more than ten sequences, the ML tree is partitioned into two subtrees by removing the centroid edge (i.e., an edge that splits the tree into two subtrees of approximately equal size). For each of the subtrees, we build a profile HMM on the alignment induced by the sequences in the leaf set of the subtree and we add the HMM to the ensemble of HMMs. This decomposition process is applied recursively on those subtrees that have at least ten sequences, until the stopping criterion based upon two parameters, X and Y, is met: the number of sequences in the subtree is at most $X\%$ of the initial seed alignment size (referred to as the "maximum decomposition size") and average pairwise sequence identity is at least $Y\%$ (referred to as the "minimum average identity threshold"). Therefore, if the initial seed alignment has ten or fewer sequences, the ensemble will only contain the HMM on the initial seed alignment.

   This decomposition process produces a collection of nested profile HMMs, with one profile HMM based upon the entire seed alignment, and (potentially) other profile HMMs based on induced alignments with fewer sequences.

This gives us a set of ensembles $\{H_i\}_{i=1}^{k}$, where each ensemble H_i is a set of p_i profile HMMs, $\{h_{ij}, j = 1, \ldots, p_i\}$.

8.2.2 Classification

In the homology detection step (Box 2 in Fig. 8.1), we are given a query sequences q, and we assign q to the family that contains the profile HMM with the highest bit score, provided the bit score is above the chosen threshold for that family. Therefore, if no family produces a bit score for q that is above the family's threshold, then $Family(q)$ is undefined, and q is not assigned to any family.

Formally, the assignment is performed as follows. First, we denote the bit score for sequence q given profile HMM h by *BS(q,h)*. We let *Family(q)* denote the family to which we assign q, and we set

$$Family(q) = \underset{i=1,\ldots,k}{\mathbf{argmax}} \left[\underset{j=1,\ldots,p_i}{\mathbf{max}} \left\{ BS(q, h_{ij}) > g_i \right\} \right]$$

where g_i is a chosen threshold for family $\mathcal{F}_i$ (see below for further discussion on thresholds). If the innermost set in this formula is empty, meaning no family has a bit score above its threshold, then no family is returned.

8.2.3 Comments

The inclusion of the minimum average identity threshold Y differentiates HIPPI from previous *PP methods, which only considered size of the subsets in building the ensemble of HMMs. For example, previous methods either decomposed until each subset was less than 10% of the full data [140] (roughly producing 20 HMMs in total); or used a fixed decomposition size [156], decomposing until the subsets have at most 10 sequences (typically producing hundreds of HMMs on an alignment with 1,000 sequences). HIPPI, on the other hand,

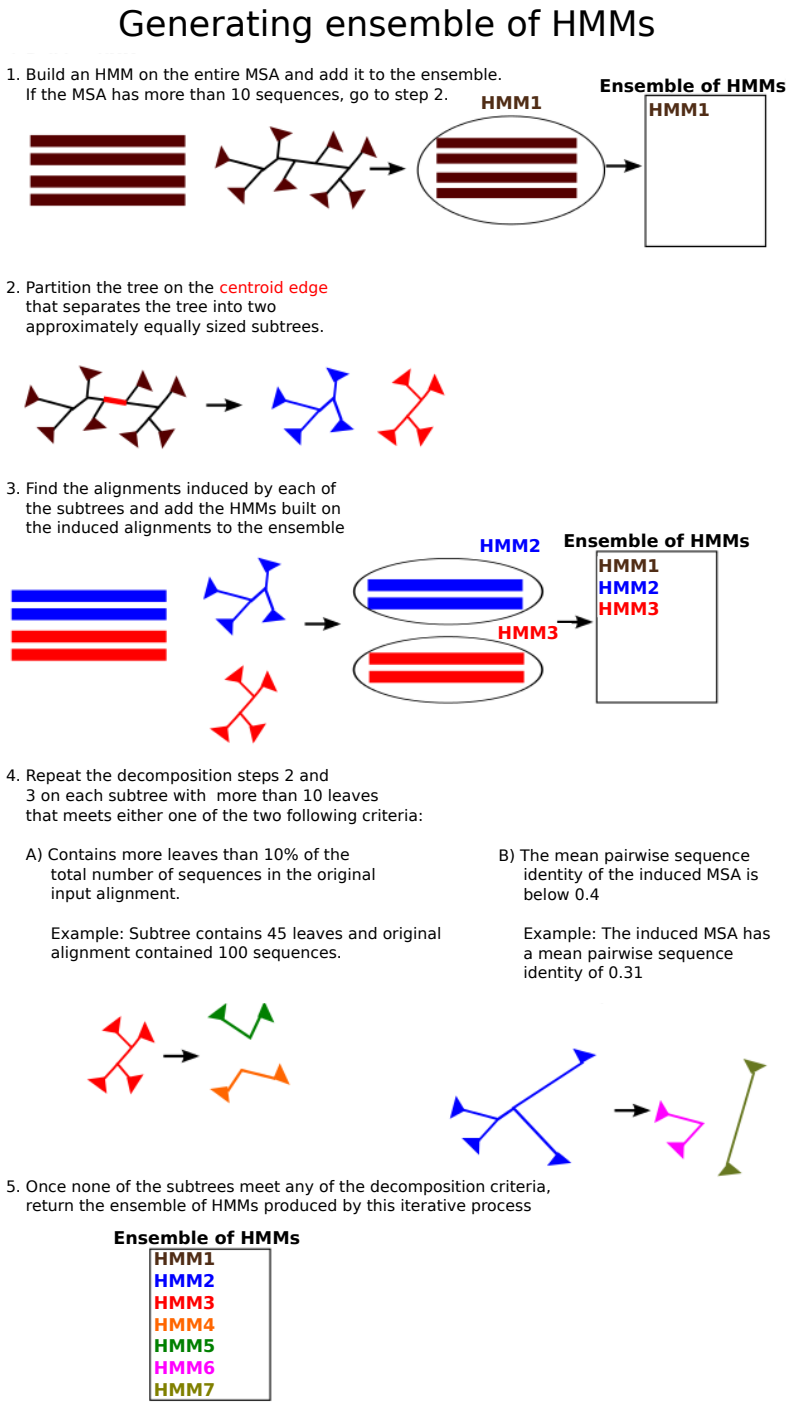


Figure 8.2: Algorithm for generating the ensemble of HMMs. The input is a seed alignment and a maximum likelihood (ML) tree that has been estimated for the seed alignment. The algorithm begins by adding the HMM built on the seed alignment to the ensemble. If the seed alignment has more than 10 sequences, the ML tree is decomposed into two subtrees by deleting the centroid edge (i.e., the edge that produces a maximally balanced split of the taxon set into two sets). The subtrees are used to generate induced alignments. HMMs are built for each induced alignment and added to the ensemble. The process iterates on those subtrees that meet the criterion for decomposition (subset size more than max(10, n/10), where n is the number of sequences in the seed alignment, and mean pairwise sequence identity less than 40%).

dynamically determines whether additional decompositions are needed based upon the heterogeneity of the subsets.

Finally, our current implementation of HIPPI, we use HMMER's *hmmbuild* to build the profile HMMs and HMMER's *hmmsearch* to score the query sequences against the ensemble of HMMs. However, HIPPI is a general approach for representing an MSA using profile HMMs, and thus can be used with any HMM-based software.

8.3 Performance study

8.3.1 Data

The Pfam-A database version 28.0 was retrieved and all families were restricted to their curated seed sequences and corresponding alignment. Since the sequences are manually curated with respect to their family assignment, we consider these assignments as de facto ground truth, which enables the cross-validation scheme described below. The set of over 16,000 families was further restricted to those with at least 10 sequences in the seed alignment. This gave us 11,156 families containing a total of 1,238,077 sequences and a diverse distribution of individual family sizes.

Additionally, for each sequence in the test set above, two fragmentary sequences were generated whose lengths were 1/4 and 1/2 of the full-length sequence. The fragments were generated by randomly selecting a contiguous subsequence (i.e., substring) of the desired length from somewhere in the full-length sequence. A link to the data used in this study is provided in HPPI's README at [155].

Cross-validation testing We conducted a four-fold cross validation test on the Pfam-A dataset by partitioning the seed sequences in each family into a test set and a training set. For each family, 75% of the seed sequences were retained and used to represent that family (i.e., the training set). The remaining 25% of the seed sequences were used as query sequences (i.e., the test set). The goal of this experimental design is to examine the accuracy of assigning the query sequences back to its original family using the reduced set of seed alignments to represent each Pfam family. We partitioned the Pfam-A dataset using this scheme four times (i.e., four pairs of test sets and training sets), with each pair of the test and training set corresponding to one cross-fold of the four-fold cross validation. The partitioning scheme used ensures that each seed sequence appears exactly once in each of the cross-folds, with three appearances in the training sets and one appearance in the test set. For very small families, the removal of even one or two sequences can substantially reduce the diversity of the sequences within the family, and thus a minimum family size of 10 was imposed to prevent spurious results on small families. This cutoff was chosen in advance and was not based upon the empirical properties of the Pfam-A dataset.

In each case, every sequence in the test set and each of its two corresponding fragmentary sequences were treated as *de novo* query sequences, and the remaining 75% was treated as the database of known families for which each query sequence could be determined to be homologous.

The computational requirements for HHsearch (see **Running Time** in **Results and Discussion**) meant

that we could only evaluate HHsearch on one of the four cross-fold subsets. However, we tested HIPPI, HMMER, and blastp on all four cross-fold subsets; the combined results of the all cross-folds are shown in the supplementary materials for the paper [155, Supplement]. The performance of these three methods across the different cross-folds is very close - no individual precision or recall for a fold was different from any other by more than 0.2%. Our main analysis, shown in Figures 8.3 and 8.4, contains the results of all methods tested on just one cross-fold.

8.3.2 Methods

The implementation of each method on this data is described below. Note that when a method assigns a sequence to a family, it is accompanied by a method-specific score; as a result, the assignment can be rejected if the score is below an inclusion threshold (see below for more details) for that combination of method and dataset. The commands used to replicate the results are given in the paper supplement.

HMMER The HMMER pipeline was implemented using HMMER v3.1b2 [65, 55]. An HMM was built on the curated Pfam alignment for the seed sequences of each family in the training data using the HMMER *hmmbuild* command. For a given query sequence, the sequence is aligned to the HMM for each family using the HMMER *hmmsearch* command and the family with the highest bit score that is above the inclusion threshold is returned.

HIPPI The preprocessing step for HIPPI was run on the training set to build ensembles of HMMs for each family (see Box 1 in Fig. 8.1), and the classification step was run on every query sequence in the test set (see Box 2 in Fig. 8.1). We report the family with the highest bit score that is above the inclusion threshold for each query sequence. HIPPI's performance is impacted by the choices of maximum decomposition size X (expressed as a percentage of the full set of sequences) and minimum average identity threshold Y (also expressed as a percentage) used in the stopping criterion. A preliminary analysis on a subset of the data (see Table 8.1 and discussion in **HIPPI parameter selection**) suggested a default setting of $X = 10$ and $Y = 40$ —decompose until each subset contains at most 10% of the initial seed sequences and has an average pairwise sequence identity that is at least 40%, or the subset contains 10 or fewer sequences. We call this default HIPPI, and results presented for HIPPI in comparison to HMMER, blastp, and HHsearch are under this parameter setting. Note that HIPPI run using a single HMM in the ensemble is equivalent to running HMMER, as described in the preceding paragraph.

HHsearch This pipeline was designed to emulate the HHpred server and uses the HH-suite software tools version 3.0.0. For each family in the training set, HHmake was used to generate an HMM on the seed alignment in the training dataset. HHblits was used with the pre-filtered version of the Uniprot20 [218] database, which is the database recommended by the software developers. For every query sequence, an HHblits search was conducted against the Uniprot20 data to generate a multiple sequence alignment. The multiple sequence alignment was scored against the HMMs built on the families in the training set using HHsearch.

The default setting for HHsearch is to use a local alignment and two iterations of HHblits to gather homologs. In order to ensure that the optimal parameters for this pipeline were used, we explored variants where we replaced local alignment by global alignment and used one iteration of HHblits instead of two. We tested these alternative settings on 30,000 sequences from the holdout data to understand their impact, particularly on fragmentary sequences. 10,000 of the sequences were randomly selected from the full-length set; we then included the half-length and quarter-length versions of the selected full-length sequences, for a total of 30,000 sequences. We label the default setting by "HHsearch: 2 Iterations". We began by comparing HHsearch: 2 Iterations using global alignment to the default setting; this comparison showed that for all sequence lengths studied, the local alignment strategy produced better results. We then compared the use of one iteration to two iterations of HHblits, followed by local alignment. The results of these initial tests are provided in the supplementary materials for the paper. No single way of running HHsearch provided the best accuracy across all the conditions explored, but the two best methods both use local alignment. Hence, we show the two top performing settings, HHsearch: 1 Iteration and HHsearch: 2 Iterations, on the holdout data; see **Results and Discussion**.

blastp For each of the four training sets, the data were first preprocessed by constructing a blastp database on all the sequences in the training set fold. For each query sequence in the test set, blastp performed a search against the database for the appropriate training set, and returns a list of the sequences that are most similar to the query sequence. We assign the query sequence to the family of the sequence with the best bit score.

Inclusion thresholds

We trace out multiple points on the precision-recall curve by varying the inclusion threshold of the classification. If the query sequence's best hit failed to meet the inclusion threshold for a given method, the query sequence was left as unclassified (i.e., it was not assigned to any family). Note that all methods have heuristics that might prevent a sequence from being scored against another family if the method can determine in advance that the match will likely be poor. Thus even if the inclusion threshold is not used, some sequences might be unclassified.

In the cases of HMMER and HIPPI, we explored the use of the Pfam curated sequence gathering cutoff thresholds for each family. However, since the Pfam threshold is chosen based on full-length sequences rather than fragmentary sequences, we also examined settings for the inclusion threshold by scaling the gathering cutoff threshold by 0% (i.e., no minimum threshold) to 100% (i.e., the default threshold). As the HHsearch pipeline returns the probability of membership, we examined thresholds for inclusion from 25% (i.e., accept the most probable family that has at least 25% support) to 99% (i.e., only accept the family if it has 99% support for membership). As blastp returns a BLAST bit score for the best hit (note that the BLAST bit score is not comparable to the HMMER bit score), we examine fixed BLAST bit scores thresholds from a bit score of 0 (i.e., accept the best hit regardless of the bit score) to a bit score of 45 (i.e., where the drop in recall was most noticeable).

8.3.3 Evaluation Criteria

For each method, we examine precision and recall for the best hit. Previous research has shown that using a single HMM on a large and evolutionarily divergent data set can lead to poor downstream analyses, and by using an ensemble of HMMs (as in the SEPP [140], TIPP [157], and UPP [156] studies) the accuracy of the analyses can be improved. These two dimensions (the number of sequences in the seed alignment and the average pairwise sequence identity of the seed alignment) are therefore of particular interest to examine the performance of our methods at the subgroup level.

8.3.4 HIPPI parameter selection

In order to determine the optimal parameter settings for the maximum decomposition size X and minimum average identity threshold Y, we tested different settings for these parameters on a very small subset of the Pfam database (called the "parameter selection set"), chosen as follows. Our goal was to find families that were very similar to each other such that the different HIPPI variants might have problems assigning the query sequences to the correct family.

We created a graph where each vertex represents a single Pfam family and the edges represented sufficient similarity between the families. Ten sequences from each family were chosen at random and scored against the single profile HMM for every other family. If the average bit score of the ten sequences scored against another family's HMM is greater than 25, an edge is drawn between the two Pfam families to represent that the families were sufficiently similar. All families with degree *at least five* (i.e., that are adjacent to at least five other families) are considered well-connected and are included in the parameter selection set; all the families they are adjacent to are also included. This produced a set of 142 Pfam families that were highly connected whose sequences would allow us to test precision at each parameter value since their sequences were mutually confusable (sequences from one family would score highly to HMMs built on the other connected families).

We consider stopping rules based on a maximum decomposition size X of 5 or 10 (expressed as a percentage of the full dataset) and on a minimum average identity threshold Y of 0 or 40 (expressed as a percentage). For each stopping rule, we consider two versions of HIPPI: a conservative version (indicated using "-c") that uses the gathering cutoff threshold g_i for each family $\mathcal{F}_i$, and an aggressive version (indicated using "-a") that drops this gathering cutoff threshold requirement. Methods are labeled as HIPPI-x (X%,Y%), where "-x" is either "-c" (conservative) or "-a" (aggressive). Note that a minimum required average sequence identity threshold of 0% would result in the same collection of profile HMMs as in UPP.

8.4 Results and Discussion

8.4.1 Impact of parameter selection

The results on the parameter selection dataset show that while methods had similar results under most conditions, the best overall results for both the aggressive and conservative versions of HIPPI used a maximum decomposition size of 10% and a minimum average identity threshold of 40% (i.e., HIPPI (10%,40%); see

Method	Fragment Size					
	Quarter-length		Half-length		Full-length	
	Prec.	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**
HIPPI-c (5%, 0%)	92.6%	33.8%	86.4%	47.7%	85.6%	79.2%
HIPPI-c (10%, 0%)	**93.1%**	**34.3%**	**86.6%**	45.1%	85.1%	78.0%
HIPPI-c (10%, 40%)	92.2%	33.4%	**86.6%**	**48.8%**	**85.8%**	**79.5%**
HIPPI-a (5%, 0%)	82.6%	39.9%	78.6%	67.4%	84.1%	83.3%
HIPPI-a (10%, 0%)	**82.8%**	37.8%	78.1%	65.2%	83.6%	82.7%
HIPPI-a (10%, 40%)	82.5%	**40.6%**	**79.3%**	**68.8%**	**84.6%**	**83.8%**

Table 8.1: Average precision / recall scores on the 143 Pfam families from the parameter selection dataset across all four-folds for the three different fragment lengths. The methods are labeled as HIPPI-x ($X\%$,$Y\%$), where "-x" is either "-c" (the conservative setting that uses the default gathering cutoff threshold for the inclusion threshold) or "-a" (the aggressive setting that uses no inclusion threshold), $X\%$ denotes the maximum decomposition size, expressed as a percentage of the number of sequences in the full dataset, and $Y\%$ denotes the minimum average pairwise sequence identity within each subset, expressed as a percentage. The best average precision and recall for the conservative and aggressive versions are bolded.

Table 8.1). Thus, we selected 10% as the default decomposition size and 40% as the minimum average identity threshold; we refer to this setting as default HIPPI. When we fix the decomposition size to be 10% and compare the impact of using a minimum average identity threshold versus not using a threshold (i.e., 40% identity versus 0% identity), the results show that using a minimum average identity threshold resulted in comparable or better precision (2 out of 6 cases with precision improvement of 1% or greater, remaining four cases with precision within 1% of each other) and better recall (5 out of 6 with recall improvement of 1% or greater; remaining one case with recall within 1% of each other). Prior uses of the *PP methods did not consider sequence identity as a determination for decomposition; these results show that taking the evolutionary diameter of the alignment into account results in improved accuracy.

8.4.2 Running time

All HHsearch variants took at least 10 seconds per query sequence on a node with 32 cores. Actual times were often longer, but I/O issues may have contributed in some cases so an accurate benchmark was not possible. Nonetheless, this meant that a single cross-fold set of test data required over 2,500 node hours, whereas all other methods ran in less than 24 node hours. On a node with 32 cores, blastp required 13 hours and 34 minutes to analyze a single cross-fold set, HMMER completed in 57 minutes, and HIPPI finished in 11 hours and 54 minutes. The running time difference between HMMER and HIPPI is not too surprising; HIPPI had 14 times more HMMs than HMMER (159,602 HMMs versus 11,156 HMMs). In terms of preprocessing time, blastp required less than a minute to build its BLAST database, HMMER required 2 minutes to build its profile HMMs, and HIPPI required 53 minutes to build its ensembles of profile HMMs, with more than 55% of HIPPI's time spent on building the ML trees used to generate the decompositions.

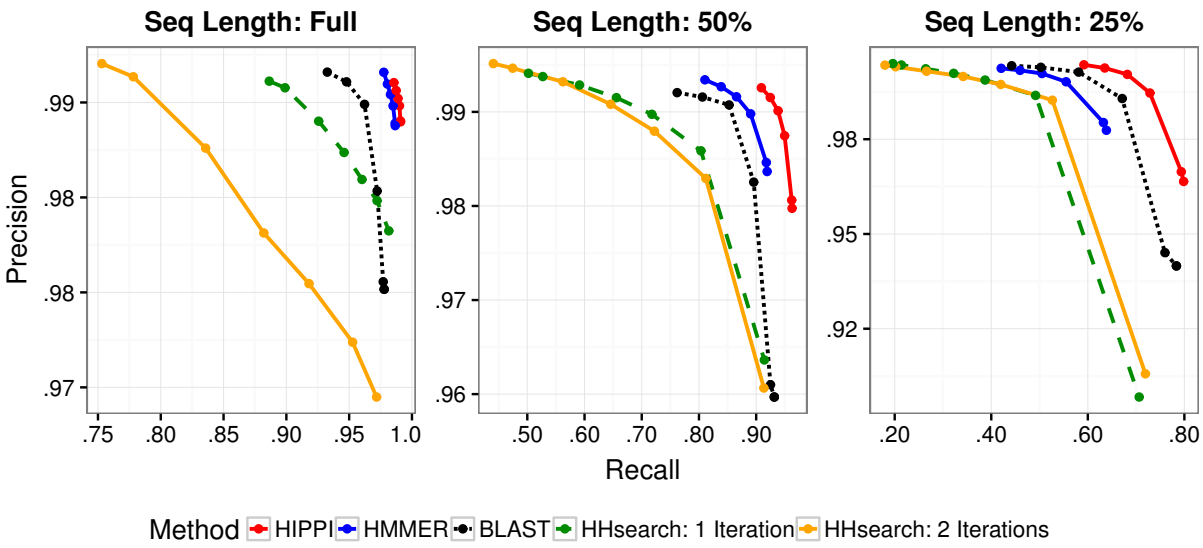


Figure 8.3: Precision-recall curves for the five methods, evaluated on one cross-fold subset of the data. We vary the length of the query sequence from unchanged (i.e., full) to half-length and quarter-length. The curves are estimated by varying an inclusion threshold parameter for the particular method and producing five to seven distinct points, with intermediate values interpolated linearly. Note that the scales for both axes vary between panels due to the significant impact of sequence fragmentation.

8.4.3 Precision and recall

We were only able to run HHsearch on a single cross-fold due to HHsearch's high computational costs. Thus, the data shown here for all methods are restricted to results on a single cross-fold set, which includes 312,841 query sequences of each fragment length (full-length, half-length, and quarter-length; 938,523 sequences total). For all other methods the precision and recall results were virtually identical across all four cross-fold sets (see paper supplement).

Figure 8.3 contains precision-recall curves for each of the five different methods under consideration. The curves are estimated by varying an inclusion threshold parameter for the particular method and producing five to seven distinct points, with intermediate values interpolated linearly. Figure 8.4 contains additional precision-recall curves that are similar to Figure 8.3, but where families are grouped according to the size of the seed alignment and its average pairwise sequence identity. The grouping by size has only two levels: families with 0 to 100 seed sequences and those with more than 100 sequences. The grouping by average pairwise sequence identity has three levels: 0-20%, 20-30%, and more than 30%, which represent about 5%, 28% and 67% of the 11,156 families, respectively.

Figure 8.3 shows that for all query sequence lengths, HIPPI dominates all other methods at every one of its computed points on the curve. Figure 8.4 shows that HIPPI's improvement over the other methods is not localized to one part of the data space; the HIPPI precision-recall curve is the most outward from the origin in nearly every case, though the degree of improvement that HIPPI has over the other methods depends on the query sequence length, average pairwise sequence identity within the seed alignment, and the number of

seed sequences.

For example, under the easiest conditions (full-length conditions with high sequence identity), all methods do very well, but HIPPI and HMMER are very close in performance, with the other methods having lower precision and recall. On the most fragmentary query sequences (quarter-length sequences), all methods degrade in recall (and, to a lesser extent, in precision), but HIPPI remains the most accurate with respect to both precision and recall and blastp becomes the next most accurate. Furthermore, on the most fragmentary sequences, the degree of improvement of HIPPI over the other methods is the largest.

Similarly, the query sequence length also impacts the choice of how to run the HHsearch pipeline (Fig. 8.3): 1 iteration of HHblits clearly dominates 2 iterations on the full-length sequences, has slightly better performance on the half-length sequences, and has slightly worse performance on the quarter-length sequences. For full-length sequences, we conjecture that the first iteration returns a sufficient number of homologs for building an HMM and that the second iteration therefore includes too many remote homologs, which in turn negatively impacts the resulting HMM. However, for shorter sequences the reverse is true: the first iteration returns fewer homologs than necessary, and a successive iteration improves results.

The differences between HIPPI and the other methods are significant. For example, the comparison between HIPPI and HMMER on the full-length sequences shows that HIPPI provides an increase in recall for the same precision of roughly 0.5%, which is statistically significant with $p < 0.001$ using a binomial test. The degree of improvement increases on the fragmentary sequences, where HIPPI provides a substantial increase in recall compared to HMMER, BLAST, and the HHsearch variants. For example, at 99.2% precision, on half-length sequences HIPPI's recall was 6% greater than HMMER, 16% greater than BLAST, and 26% greater than the HHsearch variants, and on quarter-length sequences it is 17% greater than HMMER, 10% greater than BLAST, and 34% greater than the HHsearch variants.

8.5 Conclusion and future work

HIPPI is a new method for assigning membership of query sequences to protein families and superfamilies. When used with the Pfam database, HIPPI provides higher precision and recall in comparison to the use of a single HMM, blastp, and HHsearch. Furthermore, the improvement in precision and recall is very substantial for those protein families that have seed sequence alignments with low average pairwise sequence identity, or when the query sequence is fragmentary. Thus, HIPPI enables highly accurate family identification of amino acid sequences, even for very challenging conditions.

The key advance in HIPPI over previous *PP methods is the method used to construct the ensemble of HMMs used to represent a seed alignment. While the previous *PP methods differed in various respects in how they built their ensemble of HMMs, their decompositions used stopping rules that only considered either the number of sequences in each subset or the total number of subsets created. The dataset decomposition technique that HIPPI uses, however, also considers the average pairwise sequence identity within the subsets to determine whether to continue the decomposition. This new technique for building an ensemble of HMMs enables HIPPI to select the "best" decomposition for protein families that can have very different properties in terms of the number of sequences and in the evolutionary diameter of the family (i.e.,

sequence heterogeneity), and leads to improved accuracy for protein family classification compared to the *PP decomposition strategies.

The simplicity of the HIPPI approach shows that dramatic improvements in accuracy - in terms of both precision and recall - are obtainable through divide-and-conquer strategies. Thus, although we tested HIPPI only in conjunction with profile HMMs computed using HMMER, comparable improvements might be achievable for other techniques (such as the profile HMMs used within the HHsearch pipeline, position-specific profiles, support vector machines, and Markov random fields) that are used to represent protein families, subfamilies, or superfamilies.

This study also showed that the accuracy of the HHsearch pipeline with respect to protein family identification on Pfam is substantially impacted by the choice of algorithmic parameter setting (local vs. global, and the number of HHblits iterations). However, every variant of this pipeline we studied was less accurate (for both precision and recall) than HIPPI, and typically less accurate than the use of a single profile HMM (results that are consistent with [246]). Yet, since the ensemble of HMM techniques used in HIPPI is designed to improve accuracy of bioinformatics tasks that use HMMs, we conjecture that integrating the ensemble of HMM technique in HIPPI into the HHsearch pipeline could lead to even better protein family classification than HIPPI currently achieves.

The current implementation of HIPPI uses the centroid edge in the tree to partition the subsets. However, another approach that might provide improved accuracy is partitioning the tree on the longest edge in order to increase the separation between the subsets, until all the subsets are sufficiently small and homogeneous. The study evaluating variants of the UPP multiple sequence alignment method [156] showed that dividing on the longest edge resulted in no improvements in alignment accuracy, but had an increased cost of running time due to more subsets being generated. It may be the case, however, that the longest edge decomposition has better performance for protein family identification; we are currently exploring this idea.

Currently, HIPPI assigns the query sequence to the protein family that resulted in the best bit score. However, the HIPPI algorithm also returns the bit score of the query sequence scored against all the profile HMMs. It is possible to convert the bit scores into confidence scores (i.e., the probability of a protein family for generating the query sequence) using the alignment support calculation equations provided in [157]. We plan to include confidence scores in future versions of HIPPI.

Given the improvement obtained using the new dataset decomposition strategy, we conjecture that modifications to the dataset decomposition strategies used in the previous *PP methods could lead to improvements for their respective tasks. For example, one of the key tasks in TIPP [157] is the assignment of metagenomic shotgun data to marker genes. HIPPI's superior accuracy on fragmentary sequences may lead to improved abundance profiles by enabling more accurate assignments of the fragmentary reads to protein families or gene families, and could result in improvements to other marker-based profiling methods such as mOTU [212], MetaPhyler [114], and MetaPhlAn [198].

However, other design strategies may provide even better advantages. Indeed, the overall observation in this study, as well as in SEPP [140], TIPP [157], UPP [156], T-HMM [180], and SCI-PHY [30], is that representations of multiple sequence alignments by ensembles of HMMs (however constructed) provide improved accuracy for many bioinformatics tasks compared to the use of a single HMM, and are often better

than the leading alternative methods. Thus, perhaps the design strategies used by T-HMM or SCI-PHY will provide even better accuracy. Further research is needed to find the best ways of constructing and using ensembles of HMMs, while also providing highly efficient and easy to use implementations.

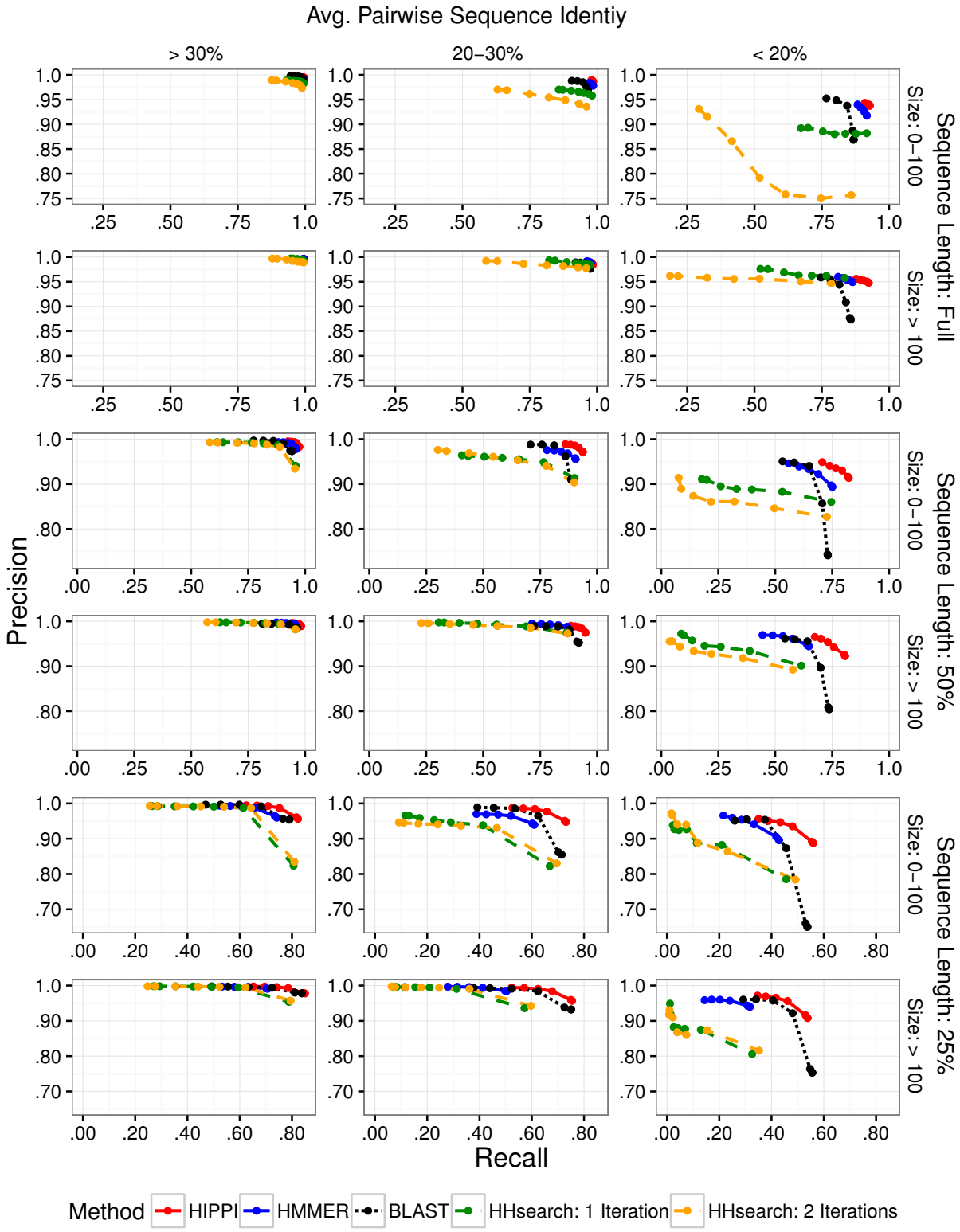


Figure 8.4: Precision-recall curves for the five methods across different model conditions, evaluated on one cross-fold subset of the data. The columns represent the mean sequence identity of the family, the rows are first grouped by sequence length, and are further subdivided by family size. Note that the scales for both axes vary between panels due to the significant impacts of segment fragmentation, family size, and sequence identity.

Chapter 9

Visualizing Microbial Biodiversity with Phylogenetic Placement

Note The following chapter describes a method of visualizing data representing a community of bacteria, the output of which is a graphic that is roughly on the order of one-page in size. As the chapter makes clear, a typical usage would involve generating (at least) one image per sample and comparing many images to analyze a study involving many samples. As a consequence, the graphics created for the two examples below are too large to be included in this dissertation and have thus been made available in Appendices A and B.

Finally, the content of this chapter is currently in preparation for submission to a journal. For that effort, all data has been made publicly available on the Illinois Data Bank (https://doi.org/10.13012/B2IDB-1678505_V1) and source code required to reproduce the graphics used is available on GitHub [162]. Additionally, the full sets of graphics and animations referred to in Section 9.4 have been included as electronic supplements with their contents described in the Appendix. This manuscript is a collaboration with Rebecca Stumpf and Karthik Yarlagadda, who have both helped to write the manuscript and interpret graphics in the two example studies.

9.1 Introduction

Microbial communities are observed principally by way of high-throughput sequencing applied to environmental samples, meaning the starting point for analysis is a tens of thousands up to billions of potentially unique sequences. Thus any quantitative method is necessarily a balancing act: it must simplify the data to the point of being useful for a researcher while still conveying as much important nuance as possible. Virtually all such methods contextualize this data first through a comparison to some set of reference sequences, which leads in turn to the challenge of quantifying the comparison. A read can match any number of those in the database, or none at all, and it can do so to a varying degree, and by way of mutation, insertion, or deletion. A common weakness of popular methods is that this varying degree and type of similarity to the reference data is not captured and mined for insight. Most often some kind of thresholding is applied below which the sequence is labeled as unidentified. It is clear that public databases of microbial biodiversity represent only a small fraction of what exists [7, 29], which means that the degree and nature of differences from the sphere of known microbes are important descriptors of a community, but these relative relationships are rarely leveraged in any substantial way. One method that does capture this, however, is phylogenetic placement.

Phylogenetic placement is a method for mapping query sequences onto a reference phylogenetic tree (generated from a reference multiple sequence alignment) according to the likely evolutionary relationship between the new sequence and the reference. Specifically, the placer adds a single new branch to the tree with the query sequence at its leaf, chosen according to maximum likelihood. The new branch is defined by two values: 1) the point of attachment on the original reference tree and 2) the length. The former represents the sequence(s) in the reference that most closely match the query. The latter represents the degree to which the query sequence differs from even its closest match; assuming the reference is comprehensive this could be considered a proxy for sequence novelty.

Algorithms for phylogenetic placement include EPA [18], pplacer [132] and SEPP [140]. In the context of microbial ecology, the relationship between phylogenetic placement and the commonly used UNIFRAC metric for microbial communities has been documented [131], and a recent study showed that using phylogenetic placement with exact amplicon sequences can improve accuracy for some common analyses [91]. Interestingly though, for all the cases where phylogenetic placement has been applied to a microbiome, to our knowledge none have used the branch lengths in the subsequent analysis in a meaningful way. [91] observed that placement is particularly useful in the presence of novel sequences, but stops there.

One major benefit of applying phylogenetic placement in microbial ecology is flexibility. Placing exact amplicon sequences is one specific use case involving PCR amplicon sequence output, but it can also be applied to representative sequences from a de-noising algorithm [153] for consistency with a separate analysis or for faster running time. It can be applied to amplicon data from any gene of particular interest as long as a suitable reference set is available, or it can be applied to shotgun sequencing data by recruiting reads to marker genes, as in MetaPhlan [224] or MetaPhyler [114]. Furthermore, the tree can even be constrained to agree with a specific taxonomy, so that the attachment point constitutes a form of taxonomic identification, as in [157]. In every case though, the degree of novelty for a sequence relative to a reference database is a useful statistic; systematic differences in sequence novelty for one or more microbial species across multiple samples could indicate structural differences that phylogenetic or taxonomic profiling may not. At a minimum, branch length data can potentially characterize the microbes in a sample in a unique way.

To that end, we describe a novel data graphic that visualizes phylogenetic placement output for a microbiome, referred to herein as a PICAN-PI chart or graph ("Population-level Identity, Composition and Novelty via Phylogenetic Insertion"). The PICAN-PI chart is a picture of the community that shows the relative phylogenetic abundance of its members *and* their degree of novelty relative to the reference data, for which it uses markers with a two-dimensional color scale (hue and saturation). The markers are positioned relative to a background image of the reference tree, suggesting phylogenetic relationships where possible. (Note that as discussed above, a tree derived from a taxonomy can be used in the placement step in lieu of an unrestricted phylogeny, so herein the term "phylogeny" is used with the understanding that a taxonomy could be substituted at the user's discretion.) This graphic is the first to quantitatively visualize population-level deviations from a reference database. The result is a graphic for a single microbial sample that has a high information density over a full screen or page. Accordingly, the design and usage presented here has been chosen with the goal of maximizing the speed and ease of interpretation, including the possibility of comparing many samples at a time.

The following sections first discuss the graphic and its interpretation, then evaluate it through a re-analysis of publicly available data from two previous studies. In both cases, phylogenetic placement was applied to exact 16S amplicon sequences and the output was plotted as a PICAN-PI image for each sample. As we show, the images not only reproduce the conclusions of the studies visually, they provide additional biologically relevant insights.

9.1.1 Related Work

Several methods are available for visualizing phylogenetic placements, including Genesis [45], guppy [132], EvolView [80] and iTOL [112], among others. Many of these have some way of visualizing the relative abundance of sequences attaching to a particular edge, and often that is accomplished by varying the edge thickness. Many of them also have several options for annotating the graphics and are designed for creative flexibility by the user.

Other tools allow the user to visualize relative abundance of a microbial sample over a tree-representation of a reference taxonomy or phylogeny, such as GraPhlan [12] and MetaCoder [70]. They are not purpose built for phylogenetic placement data, but could be adapted.

Importantly, the PICAN-PI graph is not a software but rather a schema for drawing a picture (although a reference implementation is provided). It is possible that one of the two software above could, by modifying settings carefully, produce a PICAN-PI graph, although the schema design differs in substantial ways from the defaults of each one. First, as noted above, the presentation of pendant branch lengths is a prominent feature and is unique. Second, the schema emphasizes that the layout of the reference tree and marker and branch *sizes* not be allowed to vary. These are crucial features that give the PICAN-PI graph its unique advantages and its caveats, both discussed in the following sections.

9.2 The PICAN-PI Graph

A PICAN-PI image takes the output of a phylogenetic placement algorithm as input. It begins with the reference phylogenetic tree used for placement drawn on a blank background with branch lengths accurately depicted, with the only condition that the layout be identical for every sample to be compared. Next, for each unique sequence in the placement step we have a multiplicity in the sample, an attachment location (branch id and distance from the distal end), and a subtending branch length. For each unique sequence, a marker is drawn on the reference tree with its center given by the attachment location and its color given, jointly, by the multiplicity and the subtending branch length. Specifically, using the hue-saturation-value color scale, the saturation (i.e. faint or vivid) is determined by the multiplicity and the hue is determined by the subtending branch length. (See Figure 9.1 for an example of such a color scale.)

The resulting graphic is analogous to a heat map. Each marker denotes a sequence in the sample, and its vividness denotes its relative abundance, up to a point. Because the reference phylogeny is drawn with a fixed layout, it serves as a de-facto "map" of the phylogenetic diversity in the reference sequences and allows the physical location of each marker to denote is identity within that group. Finally the hue denotes how

closely it groups with its nearest matches in the reference data.

This schema has several advantages. The layout means that physical proximity is a rough proxy for evolutionary similarity, although in some cases such as on the outer edges of large clades it can be misleading. But another advantage is that it can be "zoomed in", in a sense, by opting to display only a subset of the reference tree and optimizing the layout accordingly. Still another is that by making saturation proportional to abundance, visual significance corresponds to numeric significance and the user's attention is drawn naturally to where it is needed. Also, the coloring according to novelty allows the user to identify differences in sequences that might all otherwise map to the same taxonomic group. But arguably the most important is that the consistent interpretation of size and location means that the user can toggle between images of many samples, paying attention only to shifts in location and color, and detect trends and patterns of difference that might be very difficult to capture using purely quantitative methods.

9.2.1 Accessibility

A major disadvantage of this graphic at this point in development concerns accessibility for colorblind individuals, which is a direct consequence of using a two-dimensional color scale. Hue-coding, as described above for branch length, is a notoriously unfriendly practice in this regard [93]. Conversely, however, the information density is partly afforded by the ability to represent two dimensions of data in the same locus. One option is to replace the hue scale with an accessible gradient scale, with some colors having potentially ambiguous interpretation or possibly with a cutoff for relative abundance showing only minimally abundant sequences. Other alternatives are possible too; finding the right balance requires experimentation and can vary depending on the data at hand or user preferences. A purpose-built software could alleviate some of these issues, although unfortunately, while this is a priority for future research, it is beyond the scope of this manuscript.

To address this as well as possible, the figures contained in this chapter as well as the supplementary information have been reproduced with adjusted configurations to minimize the need for red-green discrimination, and are located in Appendices A and B. In each case, the adjustments are limited to changing the hue scale.

9.2.2 Simple Example

Figure 9.1 contains two PICAN-PI graphs to illustrate how the graphs are interpreted. Both are samples from the human vaginal microbiome, and are pulled from the primate study described in more detail below. This environment is a helpful minimal example because the healthy human vaginal microbiome is known to be colonized almost exclusively by the genera Lactobacillus and Pseudomonas, while in the case of bacterial vaginosis (BV) there is a notable presence of other microbes, among them *Gardnerella vaginalis* and *Mycoplasma hominus* [101, 250]. Figure 9.1 shows a healthy sample (right) and a sample with positive diagnosis of BV (left). The background tree is a phylogeny on all bacterial sequences in the reference data, and a full annotation is in Figure 9.2 (top left). It is omitted here to demonstrate that it is more effective to first identify the points of interest in the tree, then look more closely to see which part of the phylogeny is

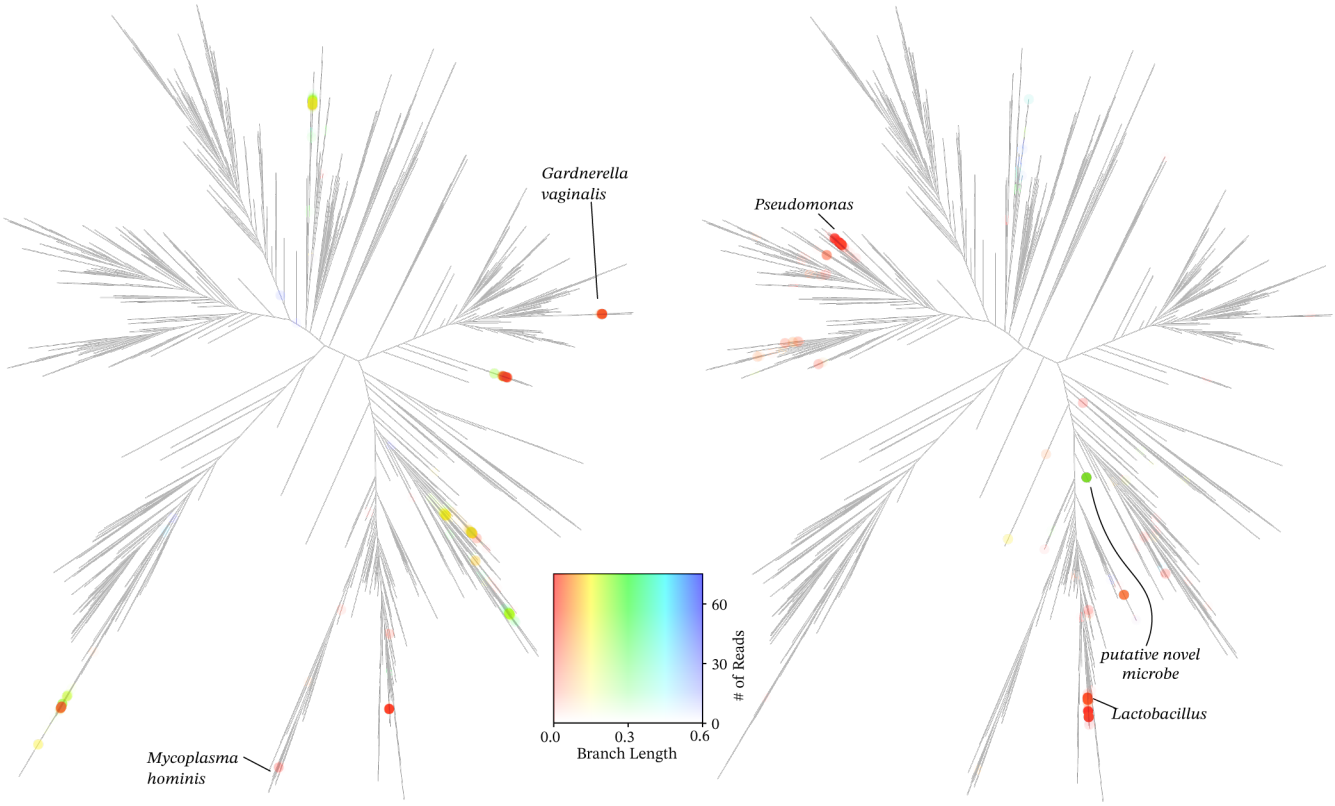


Figure 9.1: Examples of PICAN-PI graph outputs for two human vaginal microbiome samples [101]. Left: a patient diagnosed with bacterial vaginosis. Right: a negative control.

represented.

In the right graph, we see two small clusters of vivid red markers, with a small number of scattered faint markers elsewhere. The vivid clusters, on inspection, indeed correspond to Lactobacillus and Pseudomonas. In the left graph, we see a smaller cluster in the locus identified as Lactobacillus, nothing at Pseudomonas, and some additional vivid red markers. One of those can be identified as *G. vaginalis*, and another as *M. hominus*. In addition, there are several scattered groups of yellow markers, meaning slightly longer branch lengths, which indicate that the condition is characterized not just by an invasion from known pathogens, but by a general colonization of microbes that diverge from well-catalogued taxa.

As a final observation, the healthy graph contains an unmistakable green marker in the middle of the graph. On closer inspection, all nine healthy samples have a marker in the same hue and locus. The green indicates strong divergence from the nearest relative, which itself is situated on a sparsely populated clade relatively between the main Baciliales and Lactobaciliales clades. An interesting follow up study might examine whether this microbe is present in other healthy vaginal samples, and if so what it might be and what role it might play.

9.3 System & Methods

The details of the sequencing output for both studies as well as the reference phylogeny are described below. For each study, the raw sequences were de-duplicated and multiplicities recorded, and were given as input to the phylogenetic placement software SEPP [140] with the reference alignment and phylogeny specified and with parameters `-A 25 -P 200`.

The output from SEPP was then used as input by the reference implementation developed for this chapter. The complete code is available on GitHub [162] along with instructions for generating a graphic. The code relies on the DendroPy library [211] for operating on trees, and the cairo graphics library [243]. Figures for publication were assembled from the output using Adobe Illustrator, with no changes to the figures other than the removal of Archaea for display purposes. Animations were generated using the OpenCV library [90] using scripts included in the code provided on Github.

9.3.1 Data

Reference Phylogeny The reference phylogeny was constructed using 12,654 sequences from RDP release 11, update 5 [42], which were then aligned using PASTA [139] and used to construct a maximum likelihood phylogeny with RAxML [206]. These sequences were labeled using the NCBI taxonomy [61] and the tree was evaluated visually for general consistency with the taxonomy.

Primate Study For this study, we obtained sequencing data from 76 of the 77 samples directly from the authors of [250]. The details of data collection, sequencing and data availability are contained in the original study.

IBD Study Sequencing data from the original study were obtained from the NCBI sequencing read archive (SRA) using Accession PRJEB18471. Data from 683 samples were downloaded, along with metadata indicating disease subtype, patient ID, sample date and other relevant fields. Samples with fewer than 10,000 reads were excluded from our analysis, which left 611 samples.

Availability All data required to reproduce these graphics, along with the full set of graphics themselves, are publicly available in the Illinois Data Bank (https://doi.org/10.13012/B2IDB-1678505_V1). This includes the reference phylogeny (with alignment and taxonomic annotation), SEPP output, and the full set of graphics created in each of the two studies below, including versions tailored for accessibility. The required to generate these graphics is available on GitHub [162].

9.4 Results

In each of the following studies, data from a previous work is reanalyzed by creating PICAN-PI charts for each of the samples collected. Both studies used 16S PCR amplicon sequencing, so the phylogenetic placement

in both cases was done using the same reference phylogeny (i.e., Figure 9.2, top left). The examples below are chosen to show how this analysis can complement standard pipelines and provide additional insight.

The full set of images used in these studies is prohibitively large to include, so it has been included as supplementary material. The graphics created and used for the primate study in the following section is included and described in Appendix A. For the IBD study in Section 9.4.2, the graphics were generated and organized into animations grouped by disease subtype. In addition to the graphics on the entire reference tree, a separate set has been created showing only the class *Clostridia* at much higher resolution. Those animations are contained as AVI files in a separate supplementary file, described in Appendix B.

9.4.1 The Vaginal Microbiome of Human and Non-Human Primates

[250] conducted a broad survey of the vaginal microbiome in 77 samples across from 10 primate species (including human) which expanded on an earlier study of the human vaginal microbiome [101]. The initial work noted the overwhelming abundance of the genus *Lactobacillus* and to a lesser extent *Pseudomonas* in the healthy human samples, reflecting the microbes role in regulating ph. The motivating questions for the broader study were 1) whether the vaginal microbiome of other non-human primates has a similarly taxonomically narrow community, 2) whether the vaginal microbiome of non-human primates is host-specific, and 3) to what extent the microbial communities of these hosts reflects their own evolutionary relationships.

[250] found that the human vaginal microbiome is unique in being overwhelmingly restricted to a small number of genera. Every other host species has considerably larger diversity than the human samples by any metric. Additionally, communities from the same host species indeed clustered together indicating that each host had a characteristic community structure. Finally, the study did not find conclusive evidence of a structural relationship between the microbial communities and host evolution, although the abstract states directly: "The proportion of unclassified taxa observed in nonhuman primate samples increased with the phylogenetic distance from humans, indicative of the existence of previously unrecognized microbial taxa."

For our reanalysis, the interest is in exploring what more we can learn about the primate vaginal microbiome by visualizing the data, including what differentiates the communities of one host from that of another, what similarities they might have. Also, shedding some light on the curious observation about unclassified taxa would be useful.

Figure 9.2 contains composite PICAN-PI graphs for three of the host species in this study which suffice for discussion. It also contains an annotation of the background tree for reference. The composites are an average (using RGBA color values) of the PICAN-PI graphs across all samples for each host, in this case chimpanzee (14 samples), wild baboon (5 samples), and lemur (6 samples). The averaging has the effect that a marker is only vivid in the composite if it is shared by a relatively large share of the samples, so the composite graphs are a proxy for the "core" microbiome for each species. The graphics for each individual sample across all hosts are contained in the supplementary information.

First, note that by comparison to the PICAN-PI graphs of the human vaginal microbiome shown earlier (Figure 9.1), the observation that other nonhuman primates have a more diverse microbiome is quite clear; there are simply many more parts of the tree with vivid markers, even compared with the relatively diverse

community of the BV sample. Second, a limited narrative emerges through the graphics of what makes the communities host-specific and what they have in common. Four regions of the tree, labeled (a) through (d) in the lower left graphic, have a cluster of microbes in all three images, although all four of them change noticeably in hue across the three graphics, indicating at a minimum that the sequences are different. In other words, the vaginal microbiome for these three hosts each contain a group of microbes from (a) Bacteroidaceae, (b) Lactobacillaceae, (c) Clostridia, and (d) three groups in the phylum Actinobacteria, but in each case the microbes are, for the most part, different from one another. An interested researcher could use this as a starting point to see if other hosts have the same pattern, or to zoom in on each of these subtrees and characterize the overlap in additional detail.

Finally, the three hosts were chosen to examine the comment about unclassified taxa. The chimpanzees are the closest human relative, followed by the baboon and then the lemur. Recall that the hue scale represents branch length, or novelty relative to the reference database, of the sequences. Here, red indicates a perfect match to the database (zero-length), with gradations of orange and yellow indicating an imperfect or close match, followed by green indicating a more distant match and blue indicating a high degree of relative novelty. In the composite PICAN-PI graphs, the chimpanzee microbes are nearly all red, the baboons have some red, but more orange and yellow and even some green, while the lemurs are nearly all yellow to green. Other species, though not included here, show additional gradations of the same pattern, with none as uniformly green-yellow as lemurs.

That serves as an immediate explanation for why an increasing share of the sequences would be unrecognized, and gives a good idea of what kind of "previously unrecognized microbial taxa" are in there. Essentially, the microbes in these communities are similar to microbes that are well documented, but their similarity decreases as the host diverges from humans. That suggests, perhaps unsurprisingly, that the reference data may be slightly human-centric in its contents. But it also suggests that the hypothesis of a relationship between host evolution and microbiome should be examined more deeply.

9.4.2 The Microbiome of Inflammatory Bowel Disease

A recent study used longitudinal sampling to study the relationship between the gut microbiome and various subtypes of inflammatory bowel disease [78]. Subjects provided samples at roughly three month intervals. Subsequent analysis of 16S amplicon sequences relied on ordination plots and measures of alpha and beta diversity to describe the results and draw several relevant conclusions about these disease states. What those techniques have in common is that they each distill the complexity in the microbiome down to a small number of descriptive statistics that often quickly formulate a narrative about the communities. That naturally comes at the expense of detail, which is the strength of a PICAN-PI chart. This study was reanalyzed to illustrate how the PICAN-PI chart can complement such an analysis. This longitudinal data is particularly informative because it offers the chance to present many graphics together in an animation, revealing patterns in the temporal variability for the different disease states. The graphics for this reanalysis were converted into animations due to the large number, with one animation per condition. This includes animations of the PICAN-PI graphs on the full reference tree, as well as animations limited to the subtree of Clostridia only,

at better resolution.

ICD-r vs. Other Subtypes

The patients in the study reported in [78] were divided into six different disease subtypes, three types of colitis and two types of Crohn's disease, although patients with Ileal Crohn's were further divided based on whether they had had a resection surgery already (ICD-r, or ICD-nr if they had not), with a seventh group used as a healthy control. The first major conclusion of the paper is that the ICD-r group had low microbial richness and the highest volatility of any group and was most distant from the healthy group in ordination space. That the ileal resection surgery can have such an effect is noteworthy, and it raises many testable questions, e.g., Is the decrease in species richness accounted for by species in one or more particular taxonomic group? What does it mean in practice for a relatively homogenous community to be considered more volatile than a more diverse community?

The graphics make it immediately clear why the PCoA plots in the study were dominated by the difference between the ICD-r group and all others. First, in most samples in every other group, though especially so with the healthy controls, there is a consistent presence of blue and green throughout the images, that is, novel microbes relative to the database. But aside from a small number of notable samples the ICD-r graphs are almost exclusively red to orange, meaning they are close or exact matches with well known microbes. Moreover, microbes in the ICD-r samples appear to be restricted to some relatively specific locations in the tree, which show as thick red spots in the all-bacteria graphics. The Clostridia-only graphics show an example in detail. In all but the ICD-r samples, two clades are densely populated in almost every figure: *Ruminococcaceae* (at the 6-7 o'clock position) and *Lachnospiraceae* (at 9 o'clock). But in the ICD-r samples, again with a small number of notable exceptions, one of those has been almost totally cleared out (*Ruminococcaceae*) while the other is still well-represented. In the absence of the resected portion of the colon, it seems only certain microbes get to stay.

Interpreting Volatility

The other significant conclusion of this study is that all subtypes of IBD experience higher volatility in the makeup of the microbiome compared to healthy individuals, although this is more pronounced in Crohn's disease. Interestingly, another study published almost concurrently, with a similar methodology drew an essentially identical conclusion, leading to speculation that an effective, non-invasive diagnostic for Crohn's disease would have to involve repeated sampling to capture this [171, 229]. Interestingly, both papers measure volatility chiefly through unweighted Unifrac distance between subsequent samples from the same individual. That is a sensible metric but leaves open the question of what form it takes and whether it may be taxonomically localized, or whether there may be a core set whose abundance is not volatile, etc…Notably, both studies find, roughly, a mean of 0.4 for HC and ulcerative colitis (UC) patients versus 0.5 for the highly volatile Crohn's disease, but the sampling ranges for both groups are about ± 0.15 and overlap considerably. Thus, better understanding of the issue will still be valuable.

The first question then is simply whether a difference in volatility is visible when looking at the PICAN-

PI animation or individual graphs for the Crohn's patients (CCD) and the controls (HC), the set where the difference should be most extreme. It requires looking closely at both, side-by-side, but the difference suggests that for CCD samples, the set of microbes that are present at some minimum level seems to change from one sample to the next. That is, more microbes disappear from one sample to the next, and more new ones show up, and this happens roughly consistently throughout the phylogeny. In the HC samples, observed abundance may vary, but a microbe present in one sample is much more likely to be present in the next. Comparing the ulcerative colitis (UC) with the HC samples, the gap in volatility also seems to be due to this phenomenon on a much smaller level.

9.5 Discussion

Among the challenges of microbial ecology research, many of them stem from the simple fact that a microbiome is by its nature extremely complex and must be heavily simplified in order to be computationally tractable or usable to a researcher. That leads to an inevitable trade-off between the need reduce complexity to make the data manageable and the risk of oversimplification. The goal of the PICAN-PI graph is to occupy a niche at one particular end of that spectrum, favoring a complex, high-density representation of the data in the hope that it will reveal trends and meaning that may have been otherwise obscured. The approach to keeping it manageable is to design the graphic so that samples can be compared in rapid sequence, with minimal visual noise to distract the brain.

Another challenge is the general incompleteness of available databases relative to the universe of bacteria, which is addressed in small part by using the branch length from phylogenetic placement. As shown above, this is not always important, but in some cases it can draw some distinctions that might otherwise have been missed.

The benefits of using these graphics are varied and often surprising. In one case during development, one particular sample from a study caught our eye beacuse it was littered with blue markers denoting very long branches, including many attaching on the long segment of the tree separating bacteria and archaea (novel taxa indeed). This was a useful clue that something had likely gone wrong in the prep or sequencing for that sample and that it should be removed for analysis. Another example is the difference between inter- and intra-subject variability in the longitudinal IBD data above. Though it is well-documented that the microbiome varies more person-to-person that within a single person over time [43, 64], it is revealing to watch the animations, particularly of the healthy controls, and notice the break points in the stream of images where one patient ends and another begins. It is helpful to have an intuition for what constitutes a "normal" range of variation in a microbiome for a particular environment.

9.5.1 Future Work

As discussed in Section 9.2.1, the goal of a high data-density in the representation meant taking advantage of a two-dimensional color scale to present two dimensions of data, although unfortunately a two-dimensional color scale is not available all users. As a result, addressing accessibility is a priority for future develop-

ment and will require some experimentation and testing to find the best way to display both dimensions for colorblind individuals.

Another avenue for future research is addressing the question of what constitutes a significant difference between figures or groups of figures, analogous to identifying when a time series constitutes a trend or many other problems in classical statistics. Phylogenetic placement data can itself be subjected to statistical testing as to the difference in the community's distribution over the phylogeny [91, 131], but that relies on distributional assumptions that consider differences more or less significant based on distance in the phylogeny, although that may not correspond to actual priorities. Another approach would be to apply the principles of visual inference to these graphics [33, 237] to calibrate the reliability of human inference when looking at PICAN-PI graphs, although this would require developing a new way to simulate microbiome data variability. In the meantime, users are reminded that accuracy of visual inference is well correlated with sample size but *not* with user confidence [84], so caution is warranted.

9.6 Conclusion

Analysis of data is fundamentally about comparison, and properly designed data visualization makes that possible where it might otherwise have been difficult [226]. The PICAN-PI graph allows the user to compare the microbes in a sample to each other, to those of another sample, and to those of the reference database, and to do so at arbitrary resolution for any gene having a sufficiently large reference set. The examples above demonstrate how this can be used to enhance understanding of the data and find clues to causation and avenues for further research, specifically in ways that augment traditional lines of analysis.

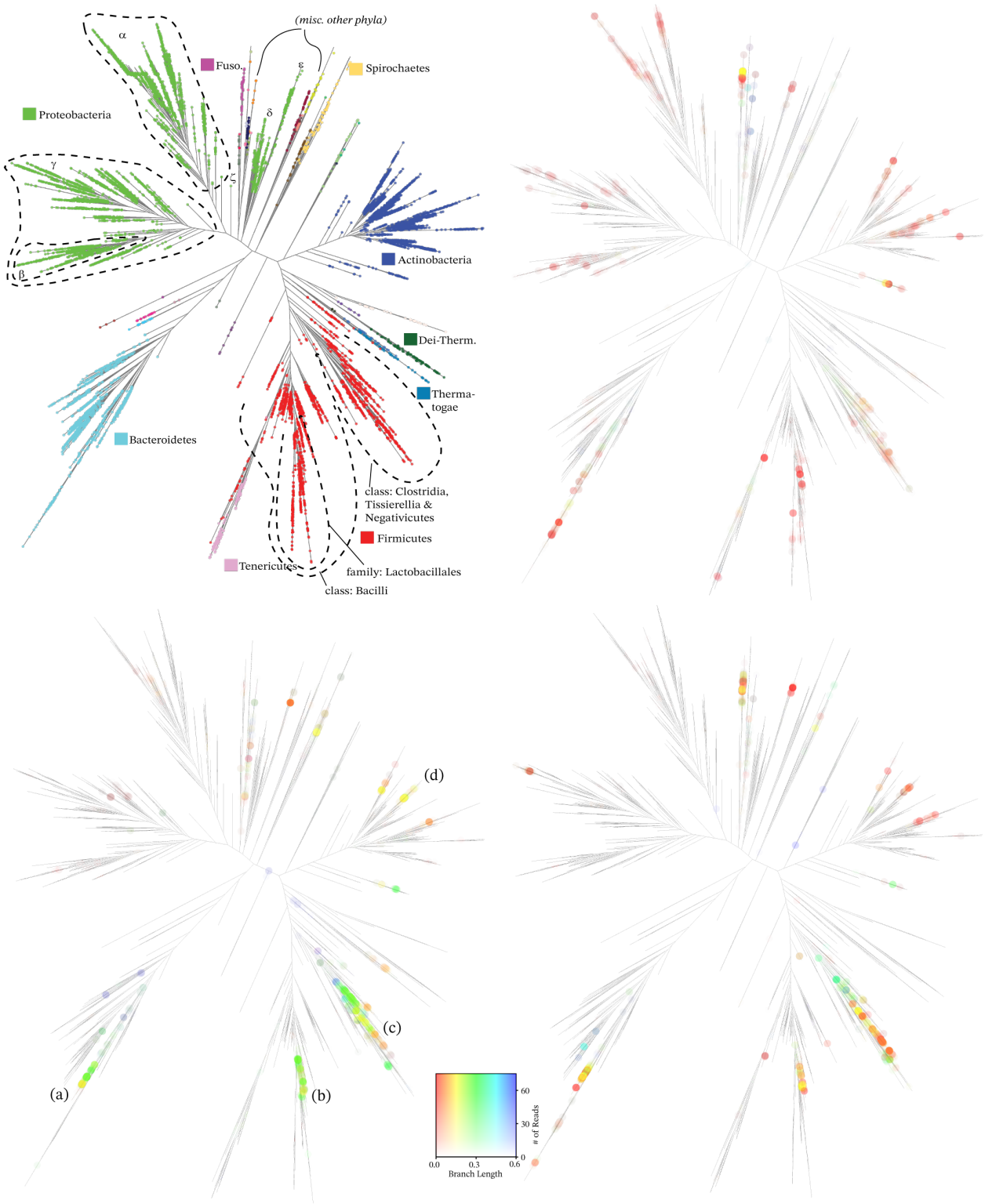


Figure 9.2: Composite PICAN-PI graphs for three host species with variable relatedness to humans: chimpanzee (top right), baboon (bottom right), and lemur (bottom left). All three graphics use the same scale. Four parts of the phylogeny with a presence in all three graphics are labeled (a)-(d) in the lower left image. An annotated background tree with major clades labeled is also shown for reference (top left).

Chapter 10

Conclusions and Future Work

10.1 Phylogenomics

10.1.1 Missing Data

The work described in Chapter 3 suggests that under some basic conditions that generate missing data, most phylogenomic species tree estimation methods maintain their statistical consistency, but nonetheless there is work to do to understand the practical implications of this.

For one, the theory developed in that section addressed statistical consistency in the proper sense of the limiting case, but a cursory look at the proofs or even basic statistical intuition suggests that the rate of missing data likely affects the convergence of these methods. The exact degree of the effect, or whether it is affected by the distribution of missing data, or whether it impacts some methods in a different way than others, has not been examined at all. It is an important matter, especially as this particular domain lacks descriptive statistics such as sample variance that might indicate confidence in the estimates to the user.

Another issue for future work is to explore additional models of missing data. Because the models described herein assume independence between the data-existence and tree topology, they represent in a sense the simplest possible probabilistic models for generating this missing data. That independence dodges the challenge of describing the correlation in detail, and as shown in Theorem 3 without independence an explicit correlation structure is necessary for consistency. But more to the point, as it pertains to researchers dealing with missing data, independence in this situation implies that the cause is strictly related to shortcomings in data gather. In other words, the reason the available sequences for a particular species may not contain a given gene has nothing to do with the species or gene in question. This may be true if the data is of uncertain provenance or is difficult to collect or if other human factors like funding constraints are at work. But as public databases are more and more densely populated and the cost of sequencing continues to decline rapidly, that is less and less the case. A much more likely cause for a researcher to be confronted with a missing gene for some species is that in those species the gene does not exist! Furthermore, the presence of a shared gene among a set of species is often a clear indicator of evolutionary closeness, which violates the idea of independence in the most direct possible way.

An expanded version of Chapter 3 included an experimental study that examined these models of missing data, but also a birth-death model where a gene may disappear or appear for the first time at a certain point one the species tree rendering it entirely present or entirely absent for all the species below [164]. The theory implications of this model were not explored, but the empirical performance comparison showed a distinct

advantage for ASTRID under that model. That makes some intuitive sense as this model would generate some extra trees with necessarily small internode distances for each clade, but the real reasons are not known with certainty. However, if the traditional models suggest that several methods are equally accurate while a more biologically realistic model favors one method in particular, then future work to explain this trend should be a priority.

The last major avenue for future research on this topic is to simply apply this in cases where missing data has thusfar precluded phylogenomic analysis outright. The most obvious example of this is in viruses, particularly double-stranded DNA viruses, where the number of available genes is large enough that a credible species-phylogeny may be within reach. Current viral taxonomies do not typically go above the Order level, and methodologies for viral taxon identification use shared gene content as the primary indicator of relatedness [24]. This suggests that ASTRID may be uniquely well suited to build a credible tree-of-life for viruses at a much higher level than previously possible.

10.1.2 Finite Site Length

The practical implication of the theory developed in Chapter 4 is that the estimation of gene trees on the way to a species tree is not a process that can be taken for granted. In practice this means that quantifying the effects of gene tree estimation error on species tree accuracy must become a *de rigueur* step in phylogenomics method development in the future. In fairness, that has already begun with recent papers including that analysis [146]. But much is still unexplored and this new dimension could change the consensus about which methods are optimal and when. Since in practice the property of finite site length applies universally, a method that shows robustness to gene tree estimation error should be preferred, all else equal.

10.2 BAli-Phy

Since they were developed exclusively on simulated data the encouraging results found in Chapter 6 are unfortunately neutralized, at least temporarily, by those of Chapter 7. Instead, future work should involve trying to more deeply understand why BAli-Phy, and *only* BAli-Phy among a large number of methods tested, shows this strange difference in how it reacts to simulated versus biological data.

The best strategic approach for this is to systematically examine to what extent the reference alignments on the biological data are consistent with BAli-Phy's core assumptions, chief among them the assumption of a tree-like evolution process. This is based on the observation that all alignment decisions in BAli-Phy are made conditional on the tree in use at the time, and if a particular column of the reference alignment displays reticulate evolution (particularly in the form of two disconnected clades that share a homologous site where the group separating them has a deletion), BAli-Phy will consider that option highly unlikely versus an alignment that put each clade in separate columns. Writ large, the effect of this would be to systematically under align, which is what has been observed on biological data relative to the curated reference alignments.

Although that explanation sounds promising, two observations from analysis not included here are relevant. First, in the alignments that I have explored in depth, BAli-Phy opts for this kind of split in the

overwhelming majority of un-gapped columns. Second, for a sequence of several un-gapped columns in a row, BAli-Phy usually performs the same split on the whole sequence, then moves to a different one in the next cluster. So that explanation would imply that if the reference alignments are correct, then reticulate evolution is not just occasional or even frequent, but more like universal, to the point of calling into question the meaning of a phylogenetic tree in the first place. So at a minimum this explanation needs to be verified with a strong study.

It's possible that the reference alignments might imply a different form of BAli-Phy model misspecification, for example the distribution of indel lengths, but it's not clear what else would lead to the particular outcomes that are observed.

A different approach that would be useful is to reach out to labs that have generated the reference data and understand what exactly the process is for that. One possible red flag would be if it was reported that the first step was to generate an alignment with MAFFT, ClustalW or MUSCLE (three of the most popular alignment software, particularly in years past) which was then refined manually based on the discretion of the curator. This may seem like a reasonable approach to generate such an extensive benchmark of curated alignments, but the simulated data implies that in those conditions, all three will tend to over align. That over alignment could easily survive the curation process, which would then yield over aligned references against which BAli-Phy, being more accurate, would appear to under align.

What seems clear based on the research presented here is that somehow BAli-Phy is inadvertently acting as a classifier that can detect the difference between simulated and biological data based on some latent property of the sequences. Regardless of what that turns out to be it should yield some insight, however small, in to the nature of biological sequences.

10.3 Applications to Microbial Ecology

10.3.1 HIPPI

HIPPI remains an interesting approach to a basic problem in microbial ecology, especially with the development of new technologies that return over a billion sequences from a single sample: what is the first thing to do to separate the known from the unknown reads? Nearly 30 years after its initial publication, BLAST [5] is still a leader for that problem. But with the number of reads per sample so large now and the size of public databases equally large, BLAST may be too computationally intensive to run [88]. As an alternative, some other methods using hashing methods to accelerate the search [32, 242]. This can speed up the search by a factor of 100 in some cases, but comes with a small tradeoff in sensitivity. HIPPI is a tradeoff in the opposite direction: additional sensitivity at the potential cost of more computing resources.

The benefit of HIPPI is therefore straightforward, and since the public repositories still only cover a small fraction of microbial biodiversity, the use case is too. The challenge is in getting it implemented consistently and with a fast enough implementation to deal with the larger samples and larger references. That should be the first goal of future research. The second should be in expanding the set of genes that it has been implemented to work with. Currently that set is the housekeeping genes used in TIPP, but there is no reason

why it has to be limited only to these.

10.3.2 Visualization

The work presented in Chapter 9 is still in its infancy. Future work on this will be devoted to getting it published, first, and then on finding opportunities to implement it for better understanding of a research question at hand. Unfortunately analysis that focuses strictly or almost strictly on alpha and beta diversity remains the dominant paradigm for microbial ecology research, but the usefulness of that approach beyond simple observations is dubious. The only way to break through that is to seek a deeper understanding of the data, and this visualization is designed to do that. Nonetheless, for a dataset as complex as sequencing output from a microbial sample, a deeper understanding requires wading through more of it and that is difficult and tedious. This visualization method is in exactly that spirit; it is abstract and requires practice to use effectively, which means it may not be adopted immediately. Future research, therefore, will have to work on ways to make this more clear and more usable by biological researchers.

Appendix A

PICAN-PI Graphics for the Primate Vaginal Microbiome Samples

The supplementary file PICAN-PI_supplement_Primate_Vaginal_all.zip contains the PICAN-PI graphics generated from the samples used in [250]. Specifically, there are two pdf files:

PICAN-PI_Full_Tree_Primate_Vaginal.pdf

contains the full set of PICAN-PI graphics generated for the re-analysis in Section 9.4.1. The first page contains an image of the reference phylogeny with every node color coded according to the phylum of that organism. The images in this file notably contain the Archaea clade, whereas the PICAN-PI graphs in Chapter 9 omitted that group to save space.

There are 76 graphics in this file other than the annotated reference on the first page. Each image has a label in the top left corner that identifies the subject, but denotes the host species first.

PICAN-PI_Full_Tree_Primate_Vaginal_COLORBLIND_ACCESSIBLE.pdf

This file contains the same set of images as in the previous file, with the exception that the hue scale has been narrowed to the blue-red range to be slightly more friendly to individuals that are colorblind.

Appendix B

PICAN-PI Animations for the Inflammatory Bowel Disease Samples

The supplementary file PICAN_PI_supplement_IBD_animations_all.zip contains four groups of animations that were generated because of the large number (over 600) of patients studied in [78] and again in Section 9.4.2. Each group has the same number of animations, along with a pdf of the annotated reference and a README file.

Before explaining the animations, the four groups are each in a separate subfolder within the ZIP file:

- **IBD_animations_full_tree**. Animations show graphics using the full reference phylogeny, including Archaea, using the full color hue-sale.
- **IBD_animations_clostridia_only**. Animations show graphics plotted only on the subtree that contains the class *Clostridia*. These also use the full color hue-scale.
- ***_accessible**. This group includes both folders with "accessible" in the name. These are identical to the images in the corresponding folders above, with the exception that the hue scale has been shift to only include the blue to red range. This is to accommodate individuals with red-green colorblindness.

In each subfolder there are seven AVI files, one for each IBD subtype studied (see the associated README also). In each one, the PICAN-PI images from all the patients and samples of that subtype are animated and presented in rapid sequence. Samples from any given patient are grouped to gether and ordered chronologically. Each image has a label that gives:

[Condition]_[PatientID]_[SampleTimePoint]

which will change along with the image. The README file in each subfolder describe the contents in slightly more detail.

References

[1] C. Afrasiabi, B. Samad, D. Dineen, C. Meacham, and K. Sjölander. The PhyloFacts FAT-CAT web server: ortholog identification and function prediction using fast approximate tree classification. *Nucleic Acids Res.*, 41(Web Server issue):1–7, 2013.

[2] E. Allman, J. Degnan, and J. Rhodes. Identifying the rooted species tree from the distribution of unrooted gene trees under the coalescent. *Journal of Mathematical Biology*, 62:833–862, 2011.

[3] E. S. Allman, C. Long, and J. A. Rhodes. Species tree inference from genomic sequences using the log-det distance. Arxiv publication arXiv:1806.04974, 2018.

[4] R. Alterovitz, A. Arvey, S. Sankararaman, C. Dallett, Y. Freund, and K. Sjölander. ResBoost: characterizing and predicting catalytic residues in enzymes. *BMC Bioinform.*, 10(1):197, Jun 2009.

[5] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–410, 1990.

[6] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25(17):3389–3402, 1997.

[7] K. Amato, J. Sanders, S. Song, M. Nute, J. Metcalf, L. Thompson, J. Morton, A. Amir, V. McKenzie, G. Humphrey, G. Gogul, J. Gaffney, A. Baden, G. Britton, F. Cuozzo, A. Di Fiore, N. Dominy, T. Goldberg, A. Gomez, M. Kowalewski, R. Lewis, A. Link, M. Sauther, S. Tecot, B. White, K. Nelson, R. Stumpf, R. Knight, and S. Leigh. Evolutionary trends in host physiology outweigh dietary niche in structuring primate gut microbiomes. *The ISME Journal*, page 1, jul 2018.

[8] A. Andreeva, A. Prlic, T. Hubbard, and A. Murzin. SISYPHUS–structural alignments for proteins with non-trivial relationships. *Nucleic Acids Res.*, 35(Database):D253–D259, jan 2007.

[9] M. R. Aniba, O. Poch, and J. D. Thompson. Issues in bioinformatics benchmarking: the case study of multiple sequence alignment. *Nucleic Acids Res.*, 38(21):7353–7363, 2010.

[10] M. Anisimova, G. Cannarozzi, and D. A. Liberles. Finding the balance between the mathematical and biological optima in multiple sequence alignment. *Trends Evol Biol*, 2(1):7, nov 2010.

[11] M. Arenas, H. Dos Santos, D. Posada, and U. Bastolla. Protein evolution along phylogenetic histories under structurally constrained substitution models. *Bioinformatics*, 29(23):3020–3028, 2013.

[12] F. Asnicar, G. Weingart, T. Tickle, C. Huttenhower, and N. Segata. Compact graphical representation of phylogenetic data and metadata with GraPhlAn. *PeerJ*, 3:e1029, jun 2015.

[13] A. Bahr, J. D. Thompson, J. C. Thierry, and O. Poch. BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations. *Nucleic Acids Res.*, 29(1):323–6, 2001.

[14] A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence database and its supplement trembl in 2000. *Nucleic Acids Res.*, 28(1):45–48, 2000.

[15] M. S. Bayzid, S. Mirarab, B. Boussau, and T. Warnow. Weighted statistical binning: enabling statistically consistent genome-scale phylogenetic analyses. *PLOS One*, 2015. DOI: 10.1371/journal.pone.0129183.

[16] M. S. Bayzid and T. Warnow. Naive binning improves phylogenomic analyses. *Bioinformatics*, 29(18):2277–84, 2013.

[17] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[18] S. Berger, D. Krompass, and A. Stamatakis. Performance, accuracy, and web server for evolutionary placement of short sequence reads under maximum likelihood. *Systematic Biology*, 60(3):291–302, 2011.

[19] J. Bernardes, G. Zaverucha, C. Vaquero, and A. Carbone. Improvement in protein domain identification is reached by breaking consensus, with the agreement of many profiles and domain co-occurrence. *PLOS Computational Biology*, 12(7):e1005038, 2016.

[20] M. J. Bishop and E. A. Thompson. Maximum likelihood alignment of DNA sequences. *J. Mol. Evol.*, 190(2):159–165, 1986.

[21] B. Blackburne and S. Whelan. Class of multiple sequence alignment algorithm affects genomic analysis. *Mol. Biol. Evol.*, 30(3):642–653, 2013.

[22] G. Blackshields, I. M. Wallace, M. Larkin, and D. G. Higgins. Analysis and comparison of benchmarks for multiple sequence alignment. *In Silico Biol.*, 6(4):321–339, 2006.

[23] B. Bode, M. Butler, T. Dunning, W. Gropp, T. Hoefler, W.-M. Hwu, and W. Kramer. The Blue Waters Super-System for Super-Science. In J. S. Vetter, editor, *Contemporary High Performance Computing: from Petascale toward Exascale*, volume 4, pages 339–366. Chapman and Hall/CRC 2013, London, UK, nov 2013.

[24] B. Bolduc, H. B. Jang, G. Doulcier, Z. You, S. Roux, and M. Sullivan. vConTACT: an iVirus tool to classify double-stranded DNA viruses that infect Archaea and Bacteria. *PeerJ*, 5:e3243, may 2017.

[25] A. Bouchard-Côté and M. I. Jordan. Evolutionary inference via the Poisson indel process. *Proceedings of the National Academy of Science*, 110(4):1601166, 2013.

[26] K. Boyce, F. Sievers, and D. Higgins. Simple chained guide trees give high-quality protein multiple sequence alignments. *Proceedings of the National Academy of Sciences*, 111(29):10556–10561, 2014. doi:10.1073/pnas.1405628111.

[27] K. Boyce, F. Sievers, and D. Higgins. Reply to Tan et al.: Differences between real and simulated proteins in multiple sequence alignments. *Proceedings of the National Academy of Science*, 112(2):E101–E101, 2015.

[28] R. Bradley, A. Roberts, M. Smoot, S. Juvekar, J. Do, C. Dewey, I. Holmes, and L. Pachter. Fast Statistical Alignment. *PLoS Comput. Biol.*, 5(5):e1000392, may 2009.

[29] C. T. Brown, L. A. Hug, B. C. Thomas, I. Sharon, C. J. Castelle, A. Singh, M. J. Wilkins, K. C. Wrighton, K. H. Williams, and J. F. Banfield. Unusual biology across a group comprising more than 15% of domain Bacteria. *Nature*, 523(7559):208–211, jun 2015.

[30] D. Brown, N. Krishnamurthy, and K. Sjölander. Automated protein subfamily identification and classification. *PLoS Comput. Biol.*, 3(8), 2007.

[31] D. Bryant, R. Bouckaert, J. Felsenstein, N. A. Rosenberg, and A. RoyChoudhury. Inferring species trees directly from biallelic genetic markers: bypassing gene trees in a full coalescent analysis. *Molecular Biology and Evolution*, 29(8):1917–1932, 2012.

[32] B. Buchfink, C. Xie, and D. H. Huson. Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, 12(1):59–60, jan 2015.

[33] A. Buja, D. Cook, H. Hofmann, M. Lawrence, E.-K. Lee, D. Swayne, and H. Wickham. Statistical inference for exploratory data analysis and model diagnostics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4361–4383, nov 2009.

[34] J. T. Cannon, B. C. Vellutini, J. Smith, F. Ronquist, U. Jondelius, and A. Hejnol. Xenacoelomorpha is the sister group to Nephrozoa. *Nature*, 530(7588):89–93, 2016.

[35] J. Cannone, S. Subramanian, M. Schnare, J. Collett, L. D'Souza, Y. Du, B. Feng, N. Lin, L. Madabusi, K. Müller, N. Pande, Z. Shang, N. Yu, and R. Gutell. The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC bioinformatics*, 3:2, jan 2002.

[36] J. A. Cavender. Taxonomy with confidence. *Math. Biosci.*, 40:271280, 1978.

[37] M. Chatzou, C. Magis, J.-M. Chang, C. Kemena, G. Bussotti, I. Erb, and C. Notredame. Multiple sequence alignment modeling: methods and applications. *Brief. Bioinform.*, 17(6):1009–1023, 2015.

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

[39] J. Chifman and L. Kubatko. Identifiability of the unrooted species tree topology under the coalescent model with time-reversible substitution processes, site-specific rate variation, and invariable sites. *Journal of theoretical biology*, 374:35–47, 2015.

[40] B. Chor and T. Tuller. Maximum likelihood of evolutionary trees: hardness and approximation. *Bioinformatics*, 21 Suppl 1:i97–106, 2005.

[41] J. Chou, A. Gupta, S. Yaduvanshi, R. Davidson, M. Nute, S. Mirarab, and T. Warnow. A comparative study of SVDquartets and other coalescent-based species tree estimation methods. *BMC Genomics*, 16(Suppl 10):S2, 2015.

[42] J. R. Cole, Q. Wang, E. Cardenas, J. Fish, B. Chai, R. J. Farris, A. S. Kulam-Syed-Mohideen, D. M. McGarrell, T. Marsh, G. M. Garrity, and J. M. Tiedje. The Ribosomal Database Project: improved alignments and new tools for rRNA analysis. *Nucleic acids research*, 37(Database issue):D141–5, jan 2009.

[43] E. Costello, C. Lauber, M. Hamady, N. Fierer, J. Gordon, and R. Knight. Bacterial community variation in human body habitats across space and time. *Science (New York, N.Y.)*, 326(5960):1694–7, dec 2009.

[44] J. A. Cuff and G. J. Barton. Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins: Structure, Function, and Genetics*, 40(3):502–511, aug 2000.

[45] L. Czech and A. Stamatakis. Scalable Methods for Post-Processing, Visualizing, and Analyzing Phylogenetic Placements. *bioRxiv*, page 346353, jun 2018.

[46] N. Daniels, A. Kumar, L. Cowen, and M. Menke. Touring Protein Space with Matt. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 9(1):286–293, jan 2012.

[47] G. Dasarathy, R. Nowak, and S. Roch. Data requirement for phylogenetic inference from multiple loci: A new distance method. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 12(2):422–432, Mar. 2015.

[48] M. DeGiorgio and J. H. Degnan. Fast and consistent estimation of species trees using supermatrix rooted triples. *Molecular Biology and Evolution*, 27(3):552–69, 2010.

[49] C. Dessimoz and M. Gil. Phylogenetic assessment of alignments reveals neglected tree signal in gaps. *Genome Biol.*, 11(4):R37, 2010.

[50] C. B. Do, S. Gross, and S. Batzoglou. Contralign: Discriminative training for protein sequence alignment. In *Proceedings of the 10th Annual International Conference on Research in Computational Molecular Biology*, RECOMB'06, pages 160–174, Berlin, Heidelberg, 2006. Springer-Verlag.

[51] C. B. Do, M. Mahabhashyam, M. Brudno, and S. Batzoglou. ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome research*, 15(2):330–340, feb 2005.

[52] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, New York, NY, USA, 1998.

[53] R. Durrett. *Probability: theory and examples*. Duxbury Press, Belmont, CA, second edition, 1996.

[54] S. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, 1998. PMID: 9918945.

[55] S. Eddy. A new generation of homology search tools based on probabilistic inference. *Genome Inform*, 23:205211, 2009.

[56] S. Eddy. HMMER: biosequence analysis using profile hidden Markov models. http://hmmer.org, 2016.

[57] R. Edgar and S. Batzoglou. Multiple sequence alignment. *Curr. Opin. Struct. Biol.*, 16(3):368–373, 2006.

[58] R. C. Edgar. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.*, 32(5):1792–1797, 2004.

[59] S. V. Edwards. Is a new and general theory of molecular systematics emerging? *Evolution*, 63:1–19, 2009.

[60] J. S. Farris. A probability model for inferring evolutionary trees. *Syst. Zool.*, 22:250 256, 1973.

[61] S. Federhen. The NCBI Taxonomy database. *Nucleic acids research*, 40(Database issue):D136–43, jan 2012.

[62] J. Felsenstein. Cases in which Parsimony or Compatibility Methods Will be Positively Misleading. *Systematic Zoology*, 27(4):401–410, Dec. 1978.

[63] J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, nov 1981.

[64] N. Fierer, C. Lauber, N. Zhou, D. McDonald, E. Costello, and R. Knight. Forensic identification using skin bacterial communities. *Proceedings of the National Academy of Sciences of the United States of America*, 107(14):6477–81, apr 2010.

[65] R. Finn, J. Clements, and S. Eddy. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res.*, 39:W29–W37, 2011.

[66] R. D. Finn, A. Bateman, J. Clements, P. Coggill, R. Y. Eberhardt, S. Eddy, A. Heger, K. Hetherington, L. Holm, J. Mistry, E. L. L. Sonnhammer, J. Tate, and M. Punta. Pfam: the protein families database. *Nucleic Acids Res.*, 42(D1):D222–D230, 2014.

[67] R. Fleissner, D. Metzler, A. von Haeseler, and P. Lewis. Simultaneous Statistical Multiple Alignment and Phylogeny Reconstruction. *Systematic Biology*, 54(4):548–561, aug 2005.

[68] W. Fletcher and Z. Yang. INDELible: a flexible simulator of biological sequence evolution. *Mol Biol Evol*, 26(8):1879–88, aug 2009.

[69] W. Fletcher and Z. Yang. The effect of insertions, deletions, and alignment errors on the branch-site test of positive selection. *Mol. Biol. Evol.*, 27(10):2257–2267, 2010.

[70] Z. Foster, T. Sharpton, N. Grünwald, M. Lefort, J. Malumbres-Olarte, and C. Vink. Metacoder: An R package for visualization and manipulation of community taxonomic diversity data. *PLOS Computational Biology*, 13(2):e1005404, feb 2017.

[71] E. Gaya, B. Redelings, P. Navarro-Rosinés, X. Llimona, M. D. Cáceres, and F. Lutzoni. Align or not to align? resolving species complexes within the Caloplaca saxicola group as a case study. *Mycologia*, 103(2):361378, 2011.

[72] R. A. George and J. Heringa. Protein domain identification and improved sequence similarity searching using PSI-BLAST. *Proteins: Structure, Function, and Genetics*, 48(4):672–681, sep 2002.

[73] R. A. Goldstein and D. D. Pollock. The tangled bank of amino acids. *Protein Science*, 25(7):1354–1362, 2016.

[74] T. Golubchik, M. J. Wise, S. Easteal, and L. S. Jermiin. Mind the Gaps: Evidence of Bias in Estimates of Multiple Sequence Alignments. *Mol. Biol. Evol.*, 24(11):2433–2442, aug 2007.

[75] A. Graybeal. Is it better to add taxa or characters to a difficult phylogenetic problem? *Systematic Biology*, 47(1):9–17, 1998.

[76] S. Guo, L.-S. Wang, and J. Kim. Large-scale simulation of RNA macroevolution by an energy-dependent fitness model. arXiv:0912.2326, 2009.

[77] D. H. Haft, B. J. Loftus, D. L. Richardson, F. Yang, J. A. Eisen, I. T. Paulsen, and O. White. TIGRFAMs: a protein family resource for the functional identification of proteins. *Nucleic Acids Res.*, 29(1):41–43, 2001.

[78] J. Halfvarson, C. Brislawn, R. Lamendella, Y. V'azquez-Baeza, W. Walters, L. Bramer, M. D'Amato, F. Bonfiglio, D. McDonald, A. Gonzalez, E. McClure, M. Dunklebarger, R. Knight, and J. Jansson. Dynamics of the human gut microbiome in inflammatory bowel disease. *Nature Microbiology*, 2:17004, feb 2017.

[79] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[80] Z. He, H. Zhang, S. Gao, M. Lercher, W. Chen, and S. Hu. Evolview v2: an online visualization and management tool for customized and annotated phylogenetic trees. *Nucleic acids research*, 44(W1):W236–41, 2016.

[81] J. Hein, J. L. Jensen, and C. Pedersen. Recursions for statistical multiple alignment. *Proceedings of the National Academy of Science*, 100:1496014965, 2003.

[82] J. Heled and A. J. Drummond. Bayesian inference of species trees from multilocus data. *Molecular Biology and Evolution*, 27(3):570–80, Mar. 2010.

[83] J. Herman, A. Novák, R. Lyngsø, A. Szabó, I. Miklós, and J. Hein. Efficient representation of uncertainty in multiple sequence alignments using directed acyclic graphs. *BMC Bioinformatics*, 16:108, 2015.

[84] H. Hofmann, L. Follett, M. Majumder, and D. Cook. Graphical Tests for Power Comparison of Competing Designs. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2441–2448, dec 2012.

[85] I. Holmes and W. J. Bruno. Evolutionary HMMs: a Bayesian approach to multiple alignment. *Bioinformatics*, 17:803–820, 2001.

[86] I. H. Holmes. Historian: accurate reconstruction of ancestral sequences and evolutionary rates. *Bioinformatics*, 33(8):1227–1229, 2017.

[87] R. Hovmoller, L. L. Knowles, and L. S. Kubatko. Effects of missing data on species tree estimation under the coalescent. *Molecular Phylogenetics and Evolution*, 69:1057–1062, 2013.

[88] D. Huson. Computational analysis of short and long microbiome sequencing reads, 2018. Invited lecture for the Institute for Genomic Biology at UIUC, Urbana, IL.

[89] S. Iantorno, K. Gori, N. Goldman, M. Gil, and C. Dessimoz. Who watches the watchmen? an appraisal of benchmarks for multiple sequence alignment. *Methods in molecular biology*, 1079:59–73, 2014.

[90] Itseez. Open source computer vision library, verson 3.30, 2017. https://github.com/itseez/opencv.

[91] S. Janssen, D. McDonald, A. Gonzalez, J. Navas-Molina, L. Jiang, Z. Z. Xu, K. Winker, D. Kado, E. Orwoll, M. Manary, S. Mirarab, and R. Knight. Phylogenetic Placement of Exact Amplicon Sequences Improves Associations with Clinical Information. *mSystems*, 3(3):e00021–18, apr 2018.

[92] E. Jarvis, S. Mirarab, A. J. Aberer, B. Li, P. Houde, C. Li, S. Ho, B. C. Faircloth, B. Nabholz, J. T. Howard, A. Suh, C. C. Weber, R. R. da Fonseca, J. Li, F. Zhang, H. Li, L. Zhou, N. Narula, L. Liu, G. Ganapathy, B. Boussau, M. S. Bayzid, V. Zavidovych, S. Subramanian, T. Gabaldón, S. Capella-Gutiérrez, J. Huerta-Cepas, B. Rekepalli, K. Munch, M. Schierup, B. Lindow, W. C. Warren, D. Ray, R. E. Green, M. W. Bruford, X. Zhan, A. Dixon, S. Li, N. Li, Y. Huang, E. P. Derryberry, M. F. Bertelsen, F. H. Sheldon, R. T. Brumfield, C. V. Mello, P. V. Lovell, M. Wirthlin, M. P. C. Schneider, F. Prosdocimi, J. A. Samaniego, A. M. V. Velazquez, A. Alfaro-Núnez, P. F. Campos, B. Petersen, T. Sicheritz-Ponten, A. Pas, T. Bailey, P. Scofield, M. Bunce, D. M. Lambert, Q. Zhou, P. Perelman, A. C. Driskell, B. Shapiro, Z. Xiong, Y. Zeng, S. Liu, Z. Li, B. Liu, K. Wu, J. Xiao, X. Yinqi, Q. Zheng, Y. Zhang, H. Yang, J. Wang, L. Smeds, F. E. Rheindt, M. Braun, J. Fjeldsa, L. Orlando, F. K. Barker, K. A. Jonsson, W. Johnson, K.-P. Koepfli, S. O'Brien, D. Haussler, O. A. Ryder, C. Rahbek, E. Willerslev, G. R. Graves, T. C. Glenn, J. McCormack, D. Burt, H. Ellegren, P. Alstrom, S. V. Edwards, A. Stamatakis, D. P. Mindell, J. Cracraft, E. L. Braun, T. Warnow, W. Jun, M. T. P. Gilbert, and G. Zhang. Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science*, 346(6215):1320–1331, 2014.

[93] B. Jenny and N. Kelso. Color design for the color vision impaired. *Cartographic Perspectives*, 0(58), 2007.

[94] E. Jewett and N. Rosenberg. iGLASS: an improvement to the GLASS method for estimating species trees from gene trees. *Journal of Computational Biology*, 19(3):293–315, 2012.

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

[96] E. L. Karin, E. Susko, and T. Pupko. Alignment errors strongly impact likelihood-based tests for comparing topologies. *Mol. Biol. Evol.*, 31(11):3057–3067, 2014.

[97] K. Katoh, K. Misawa, K.-i. Kuma, and T. Miyata. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.*, 30(14):3059–3066, 2002.

[98] K. Katoh and D. Standley. A simple method to control over-alignment in the MAFFT multiple sequence alignment program. *Bioinformatics*, 32(16):1933–1942, 2016.

[99] J. Kececioglu. The maximum weight trace problem in multiple sequence alignment. In *Combinatorial Pattern Matching*, pages 106–119. Springer-Verlag, Berlin/Heidelberg, 1993.

[100] C. Kemena, J.-F. Taly, J. Kleinjung, and C. Notredame. STRIKE: evaluation of protein MSAs using a single 3D structure. *Bioinformatics*, 27(24):3385–3391, 2011.

[101] T. Kim, S. Thomas, M. Ho, S. Sharma, C. Reich, J. Frank, K. Yeater, D. Biggs, N. Nakamura, R. Stumpf, S. Leigh, R. Tapping, S. Blanke, J. Slauch, H. Gaskins, J. Weisbaum, G. Olsen, L. Hoyer, and B. Wilson. Heterogeneity of vaginal microbial communities within individuals. *Journal of clinical microbiology*, 47(4):1181–9, apr 2009.

[102] J. F. C. Kingman. On the Genealogy of Large Populations. *Journal of Applied Probability*, 19:27, 1982.

[103] N. Krishnamurthy, D. Brown, and K. Sjölander. Flowerpower: clustering proteins into domain architecture classes for phylogenomic inference of protein function. *BMC Evol. Biol.*, 7(1):1–11, 2007.

[104] L. S. Kubatko, B. C. Carstens, and L. L. Knowles. Stem: species tree estimation using maximum likelihood for gene trees under coalescence. *Bioinformatics*, 25(7):971–973, 2009.

[105] J. A. Lake. The order of sequence alignment can bias the selection of tree topology. *Mol. Biol. Evol.*, 8(3):378–385, 1991.

[106] B. R. Larget, S. K. Kotha, C. N. Dewey, and C. Ané. BUCKy: gene tree/species tree reconciliation with Bayesian concordance analysis. *Bioinformatics*, 26(22):2910–2911, 2010.

[107] T. Lassmann and E. L. L. Sonnhammer. Automatic assessment of alignment quality. *Nucleic Acids Res.*, 33(22):7120–7128, 2005.

[108] Q. Le, F. Sievers, and D. G. Higgins. Protein multiple sequence alignment benchmarking through secondary structure prediction. *Bioinformatics*, 33(January):1331–1337, 2017.

[109] A. D. Leaché and B. Rannala. The accuracy of species tree estimation under simulation: a comparison of methods. *Systematic Biology*, 60(2):126–37, Mar. 2011.

[110] V. Lefort, R. Desper, and O. Gascuel. FastME 2.0: A Comprehensive, Accurate, and Fast Distance-Based Phylogeny Inference Program: Table 1. *Molecular Biology and Evolution*, 32(10):2798–2800, 2015.

[111] A. R. Lemmon, J. M. Brown, K. Stanger-Hall, and E. M. Lemmon. The effect of ambiguous data on phylogenetic estimates obtained by maximum likelihood and Bayesian inference. *Systematic Biology*, 58(1):130–45, 2009.

[112] I. Letunic and P. Bork. Interactive tree of life (iTOL) v3: an online tool for the display and annotation of phylogenetic and other trees. *Nucleic acids research*, 44(W1):W242–5, 2016.

[113] D. A. Liberles, S. A. Teichmann, I. Bahar, U. Bastolla, J. Bloom, E. Bornberg-Bauer, L. J. Colwell, A. P. J. de Koning, N. V. Dokholyan, J. Echave, A. Elofsson, D. L. Gerloff, R. A. Goldstein, J. A. Grahnen, M. T. Holder, C. Lakner, N. Lartillot, S. C. Lovell, G. Naylor, T. Perica, D. D. Pollock, T. Pupko, L. Regan, A. Roger, N. Rubinstein, E. Shakhnovich, K. Sjölander, S. Sunyaev, A. I. Teufel, J. L. Thorne, J. W. Thornton, D. M. Weinreich, and S. Whelan. The interface of protein structure, protein biophysics, and molecular evolution. *Protein Science*, 21(6):769–785, 2012.

[114] B. Liu, T. Gibbons, M. Ghodsi, T. Treangen, and M. Pop. Accurate and fast estimation of taxonomic profiles from metagenomic shotgun sequences. *BMC Genomics*, 12 Suppl 2(2):S4, jan 2011.

[115] K. Liu, C. Linder, and T. Warnow. RAxML and FastTree: comparing two methods for large-scale maximum likelihood phylogeny estimation. *PLoS One*, 6(11):e27731, 2011.

[116] K. Liu, S. Raghavan, S. Nelesen, C. R. Linder, and T. Warnow. Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science*, 324(5934):1561–1564, 2009.

[117] K. Liu, T. J. Warnow, M. T. Holder, S. M. Nelesen, J. Yu, A. P. Stamatakis, and C. R. Linder. SATé-II: very fast and accurate simultaneous estimation of multiple sequence alignments and phylogenetic trees. *Systematic Biology*, 61(1):90–106, 2012.

[118] L. Liu. BEST: Bayesian estimation of species trees under the coalescent model. *Bioinformatics*, 24(21):2542–2543, 2008.

[119] L. Liu and L. Yu. Estimating species trees from unrooted gene trees. *Systematic Biology*, 60(5):661–667, 2011.

[120] L. Liu, L. Yu, and S. V. Edwards. A maximum pseudo-likelihood approach for estimating species trees under the coalescent model. *BMC Evolutionary Biology*, 10(1):302, 2010.

[121] L. Liu, L. Yu, D. K. Pearl, and S. V. Edwards. Estimating species phylogenies using coalescence times among sequences. *Systematic Biology*, 58(5):468–77, Oct. 2009.

[122] C. Long and L. Kubatko. Identifiability and reconstructibility of species phylogenies under a modified coalescent. Arxiv publication arXiv:1701.06871, 2017.

[123] P. Lopez, D. Casane, and H. Philippe. Heterotachy, an important process of protein evolution. *Mol. Biol. Evol.*, 19:1–7, 2002.

[124] A. Löytynoja and N. Goldman. An algorithm for progressive multiple alignment of sequences with insertions. *Proceedings of the National Academy of Sciences*, 102(30):10557–10562, 2005.

[125] A. Löytynoja and N. Goldman. Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, 320(5883):1632–1635, 2008.

[126] A. Löytynoja, A. J. Vilella, and N. Goldman. Accurate extension of multiple sequence alignments using a phylogeny-aware graph algorithm. *Bioinformatics*, 28(13):1684–1691, 2012.

[127] G. Lunter, I. Miklós, A. Drummond, J. L. Jensen, and J. Hein. Bayesian coestimation of phylogeny and sequence alignment. *BMC Bioinf.*, 6:83, 2005.

[128] G. A. Lunter, I. Miklós, Y. S. Song, and J. Hein. An efficient algorithm for statistical multiple alignment on arbitrary phylogenetic trees. *J Comp Biol*, 10:869–889, 2003.

[129] D. Maddison. The rapidly changing landscape of insect phylogenetics. *Current Opinion in Insect Science*, 18:77–82, 2016.

[130] W. P. Maddison. Gene trees in species trees. *Systematic Biology*, 46(3):523–536, 1997.

[131] F. Matsen and S. Evans. The Phylogenetic Kantorovich-Rubinstein Metric for Environmental Sequence Samples. *Journal of the Royal Statistical Society B*, 74(Part 3):569–592, 2012.

[132] F. Matsen, R. Kodner, and E. V. Armbrust. pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC Bioinformatics*, 11(1):538, 2010.

[133] S. K. McKenzie, P. R. Oxley, and D. J. C. Kronauer. Comparative genomics and transcriptomics in ants provide new insights into the evolution and function of odorant binding and chemosensory proteins. *BMC Genomics*, 15(1):718, jan 2014.

[134] R. W. Meredith, J. E. Janečka, J. Gatesy, O. a. Ryder, C. a. Fisher, E. C. Teeling, A. Goodbla, E. Eizirik, T. L. L. Simão, T. Stadler, D. L. Rabosky, R. L. Honeycutt, J. J. Flynn, C. M. Ingram, C. Steiner, T. L. Williams, T. J. Robinson, A. Burk-Herrick, M. Westerman, N. a. Ayoub, M. S. Springer, and W. J. Murphy. Impacts of the Cretaceous Terrestrial Revolution and KPg extinction on mammal diversification. *Science (New York, N.Y.)*, 334(6055):521–4, 2011.

[135] I. Miklós. An improved algorithm for statistical alignment of sequences related by a star tree. *Bull. Math. Biol.*, 64:771–779, 2002.

[136] I. Miklós. Algorithm for statistical alignment of sequences derived from a Poisson sequence length distribution. *Disc. Appl. Math.*, 127(1):79–84, 2003.

[137] I. Miklós, G. A. Lunter, and I. Holmes. A "long indel model" for evolutionary sequence alignment. *Mol. Biol. Evol.*, 21(3):529–540, 2004.

[138] S. Mirarab, M. S. Bayzid, B. Boussau, and T. Warnow. Statistical binning improves phylogenomic analysis. *Science*, 346(6215):1250463, 2014.

[139] S. Mirarab, N. Nguyen, S. Guo, L.-S. Wang, J. Kim, and T. Warnow. PASTA: ultra-large multiple sequence alignment for nucleotide and amino-acid sequences. *J Comp Biol*, 22(5):377–386, 2015.

[140] S. Mirarab, N. Nguyen, and T. Warnow. SEPP: SATé-enabled phylogenetic placement. *Proceedings of the Pac. Symp. Biocomput.*, 17:247–58, Jan. 2012. PMID:22174280.

[141] S. Mirarab, R. Reaz, M. Bayzid, T. Zimmermann, M. Swenson, and T. Warnow. ASTRAL: genome-scale coalescent-based species tree estimation. *Bioinformatics*, 30(17):i541–i548, 2014.

[142] S. Mirarab and T. Warnow. FastSP: linear time calculation of alignment accuracy. *Bioinformatics*, 27(23):3250–8, dec 2011.

[143] S. Mirarab and T. Warnow. ASTRAL-II: coalescent-based species tree estimation with many hundreds of taxa and thousands of genes. *Bioinformatics*, 31(12):i44–i52, 2015.

[144] B. Misof, S. Liu, K. Meusemann, R. S. Peters, A. Donath, C. Mayer, P. B. Frandsen, J. Ware, T. Flouri, R. G. Beutel, O. Niehuis, M. Petersen, F. Izquierdo-Carrasco, T. Wappler, J. Rust, A. J. Aberer, U. Aspöck, H. Aspöck, D. Bartel, A. Blanke, S. Berger, A. Böhm, T. R. Buckley, B. Calcott, J. Chen, F. Friedrich, M. Fukui, M. Fujita, C. Greve, P. Grobe, S. Gu, Y. Huang, L. S. Jermiin, A. Y. Kawahara, L. Krogmann, M. Kubiak, R. Lanfear, H. Letsch, Y. Li, Z. Li, J. Li, H. Lu, R. Machida, Y. Mashimo, P. Kapli, D. D. McKenna, G. Meng, Y. Nakagaki, J. L. Navarrete-Heredia, M. Ott, Y. Ou, G. Pass, L. Podsiadlowski, H. Pohl, B. M. von Reumont, K. Schütte, K. Sekiya, S. Shimizu, A. Slipinski, A. Stamatakis, W. Song, X. Su, N. U. Szucsich, M. Tan, X. Tan, M. Tang, J. Tang, G. Timelthaler, S. Tomizuka, M. Trautwein, X. Tong, T. Uchifune, M. G. Walzl, B. M. Wiegmann, J. Wilbrandt, B. Wipfler, T. K. F. Wong, Q. Wu, G. Wu, Y. Xie, S. Yang, Q. Yang, D. K. Yeates, K. Yoshizawa, Q. Zhang, R. Zhang, W. Zhang, Y. Zhang, J. Zhao, C. Zhou, L. Zhou, T. Ziesmann, S. Zou, Y. Li, X. Xu, Y. Zhang, H. Yang, J. Wang, J. Wang, K. M. Kjer, and X. Zhou. Phylogenomics resolves the timing and pattern of insect evolution. *Science*, 346(6210):763–767, 2014.

[145] K. Mizuguchi, C. M. Deane, T. L. Blundell, and J. P. Overington. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci.*, 7(11):2469–71, nov 1998.

[146] E. K. Molloy and T. Warnow. To include or not to include: The impact of gene filtering on species tree estimation methods. *Systematic Biology*, page syx077, 2017.

[147] B. Morgenstern. DIALIGN 2: Improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, 15(3):211–218, 1999.

[148] D. A. Morrison and J. T. Ellis. Effects of nucleotide sequence alignment on phylogeny estimation: a case study of 18S rDNAs of apicomplexa. *Mol. Biol. Evol.*, 14(4):428–41, apr 1997.

[149] E. Mossel and S. Roch. Incomplete lineage sorting: consistent phylogeny estimation from multiple loci. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(1):166–71, 2010.

[150] N. J. Mulder and R. Apweiler. Tools and resources for identifying protein families, domains and motifs. *Genome Biol.*, 3(1), 2002.

[151] National Research Council. *Frontiers in Massive Data Analysis*. National Academies Press, Washington, DC, 2013. ISBN 978-0-309-28778-4.

[152] S. Nayfach, P. H. Bradley, S. K. Wyman, T. J. Laurent, A. Williams, J. A. Eisen, K. S. Pollard, and T. J. Sharpton. Automated and accurate estimation of gene family abundance from shotgun metagenomes. *PLoS Computational Biology*, 11(11):e1004573, nov 2015.

[153] J. T. Nearing, G. M. Douglas, A. M. Comeau, and M. G. Langille. Denoising the denoisers: an independent evaluation of microbiome sequence error-correction approaches. *PeerJ*, 6:e5364, Aug. 2018.

[154] J. Neyman. Molecular studies of evolution: a source of novel statistical problems. In S. Gupta and J. Yackel, editors, *Statistical decision theory and related topics*, page 127. Academic Press, New York and London, 1971.

[155] N. Nguyen. HIPPI README. https://github.com/smirarab/sepp/blob/master/README.HIPPI.md, 2016. [Online; accessed 26-July-2016].

[156] N. Nguyen, S. Mirarab, K. Kumar, and T. Warnow. Ultra-large alignments using phylogeny-aware profiles. *Genome Biol.*, 16(1):124, 2015.

[157] N. Nguyen, S. Mirarab, B. Liu, M. Pop, and T. Warnow. TIPP: taxonomic identification and phylogenetic profiling. *Bioinformatics*, 30(24):3548–3555, 2014.

[158] N. Nguyen, M. Nute, S. Mirarab, and T. Warnow. HIPPI: highly accurate protein family classification with ensembles of HMMs. *BMC Genomics*, 17(S10):89–100, nov 2016.

[159] C. Notredame. Recent evolutions of multiple sequence alignment algorithms. *PLoS Computational Biology*, 3(8):1405–1408, 2007.

[160] C. Notredame, D. Higgins, and J. Heringa. T-coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, 302(1):205–217, 2000.

[161] Á. Novák, I. Miklós, R. Lyngsø, and J. Hein. StatAlign: An extendable software package for joint Bayesian estimation of alignments and evolutionary trees. *Bioinformatics*, 24(20):2403–2404, 2008.

[162] M. Nute. Github site for PICAN-PI: a graphical schema to visualize microbial biodiversity, 2019. https://github.com/MGNute/pican_pi.

[163] M. Nute and J. Chou. Statistical Consistency of Coalescent-Based Species Tree Methods Under Models of Missing Data. In *Comparative Genomics, 15th International Workshop, RECOMB-CG 2017, Barcelona, Spain, October 4-6, 2017. Proceedings.*, pages 277–297. Springer, Cham, oct 2017.

[164] M. Nute, J. Chou, E. Molloy, and T. Warnow. The performance of coalescent-based species tree estimation methods under models of missing data. *BMC Genomics*, 19(S5):286, may 2018.

[165] M. Nute, E. Saleh, and T. Warnow. Evaluating statistical multiple sequence alignment in comparison to other alignment methods on protein data sets. *Systematic Biology*, page syy068, 2018.

[166] M. Nute and T. Warnow. Scaling statistical multiple sequence alignment to large datasets. *BMC Genomics*, 17(S10):135–144, 2016.

[167] T. H. Ogden and M. S. Rosenberg. Multiple sequence alignment accuracy and phylogenetic inference. *Systematic Biology*, 55(2):314–28, apr 2006.

[168] O. O'Sullivan, K. Suhre, C. Abergel, D. G. Higgins, and C. Notredame. 3DCoffee: Combining protein sequences and structures within multiple sequence alignments. *J. Mol. Biol.*, 340(2):385–395, 2004.

[169] R. Page. Modified mincut supertrees. In R. Guigó and D. Gusfield, editors, *Algorithms in Bioinformatics*, volume 2452 of *Lecture Notes in Computer Science*, pages 537–551. Springer, Berlin and Heidelberg, 2002.

[170] F. S.-M. Pais, P. d. C. Ruy, G. Oliveira, and R. S. Coimbra. Assessing the efficiency of multiple sequence alignment programs. *Algorithms Mol. Biol.*, 9(1):4, Mar 2014.

[171] V. Pascal, M. Pozuelo, N. Borruel, F. Casellas, D. Campos, A. Santiago, X. Martinez, E. Varela, G. Sarrabayrouse, K. Machiels, S. Vermeire, H. Sokol, F. Guarner, and C. Manichanh. A microbial signature for Crohn's disease. *Gut*, 66(5):813–822, feb 2017.

[172] S. Patel, R. T. Kimball, and E. L. Braun. Error in phylogenetic estimation for bushes in the tree of life. *Journal of Phylogenetics and Evolutionary Biology*, 1(2):110, 2013.

[173] S. Pattabiraman and T. Warnow. Are profile hidden Markov models identifiable? In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 448–456. ACM, 2018.

[174] J. Pei, B.-h. Kim, and N. V. Grishin. PROMALS3D : a tool for multiple protein sequence and structure alignments. *Nucleic Acids Res.*, 36(7):2295–2300, 2008.

[175] N. Perdigao, J. Heinrich, C. Stolte, K. S. Sabir, M. J. Buckley, B. Tabor, B. Signal, B. S. Gloss, C. J. Hammang, B. Rost, A. Schafferhans, and S. I. O'Donoghue. Unexpected features of the dark proteome. *Proceedings of the National Academy of Sciences*, 112(52):15898–15903, nov 2015.

[176] H. Philippe, D. Vienne, V. Ranwez, B. Roure, D. Baurain, and F. Delsuc. Pitfalls in supermatrix phylogenomics. *Eur. J. Taxon.*, 283, 2017.

[177] D. D. Pollock, D. J. Zwickl, J. A. McGuire, and D. M. Hillis. Increased taxon sampling is advantageous for phylogenetic inference. *Systematic Biology*, 51:664–671, 2002.

[178] D. Posada. Phylogenomics for systematic biology. *Systematic Biology*, 65(3):353–356, 2016.

[179] M. N. Price, P. S. Dehal, and A. P. Arkin. FastTree 2–approximately maximum-likelihood trees for large alignments. *PLOS One*, 5(3):e9490, jan 2010.

[180] B. Qian and R. A. Goldstein. Detecting distant homologs using phylogenetic tree-based HMMS. *Proteins: Structure, Function and Genetics*, 52(3):446–453, 2003.

[181] B. Qian and R. A. Goldstein. Performance of an iterated T-HMM for homology detection. *Bioinformatics*, 20(14):2175–2180, 2004.

[182] B. Redelings. Erasing errors due to alignment ambiguity when estimating positive selection. *Mol. Biol. Evol.*, 31(8):1979–1993, 2014.

[183] B. Redelings. BAli-Phy's User's Guide v3.0, 2018. Obtained from http://www.bali-phy.org/README.html#mixing_and_convergence; accessed 2018-02-27.

[184] B. Redelings and M. Suchard. Joint Bayesian estimation of alignment and phylogeny. *Systematic Biology*, 54(3):401–418, 2005.

[185] B. Redelings and M. Suchard. Incorporating indel information into phylogeny estimation for rapidly emerging pathogens. *BMC Evolutioanry Biology*, 7:40, 2007.

[186] G. Reeck, C. de Haen, D. Teller, R. Doolitte, W. Fitch, R. Dickerson, P. Chambon, A. McLachlan, E. Margoliash, T. Jukes, and E. Zuckerkandl. "Homology" in proteins and nucleic acids: a terminology muddle and a way out of it. *Cell*, 50:667, 1987.

[187] M. Remmert, A. Biegert, A. Hauser, and J. Söding. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature Methods*, 9(2):173–175, 2012.

[188] E. Rivas and S. Eddy. Parameterizing sequence alignment with an explicit evolutionary model. *BMC Bioinformatics*, 16:406, 2015.

[189] D. Robinson and L. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1-2):131–147, 1981.

[190] S. Roch. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 3(1):92–, Jan. 2006.

[191] S. Roch, M. Nute, and T. Warnow. Long-branch attraction in species tree estimation: Inconsistency of partitioned likelihood and topology-based summary methods. *Systematic Biology*, page syy061, 2018.

[192] S. Roch and M. Steel. Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent. *Theor. Popul. Biol.*, 100:56–62, 2015.

[193] S. Roch and T. Warnow. On the robustness to gene tree estimation error (or lack thereof) of coalescent-based species tree methods. *Systematic Biology*, 64(4):663–676, 2015.

[194] U. Roshan and D. R. Livesay. Probalign: multiple sequence alignment using partition function posterior probabilities. *Bioinform.*, 22(22):2715–2721, nov 2006.

[195] B. Rost. Twilight zone of protein sequence alignments. *Protein Eng.*, 12(2):85–94, 1999.

[196] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987.

[197] S. Sankararaman and K. Sjölander. INTREPID–INformation-theoretic TREe traversal for Protein functional site IDentification. *Bioinformatics*, 24(21):2445–2452, 2008.

[198] N. Segata, L. Waldron, A. Ballarini, V. Narasimhan, O. Jousson, and C. Huttenhower. Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature Methods*, 9(8):811–4, aug 2012.

[199] C. Semple and M. Steel. *Phylogenetics*. Oxford lecture series in mathematics and its applications. Oxford University Press, 2003.

[200] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Söding, J. D. Thompson, and D. G. Higgins. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular systems biology*, 7:539, jan 2011.

[201] M. P. Simmons, K. F. Müller, and A. P. Norton. Alignment of, and phylogenetic inference from, random sequences: The susceptibility of alternative alignment methods to creating artifactual resolution and support. *Mol. Phylogenet. Evol.*, 57(3):1004–1016, 2010.

[202] P. Skewes-Cox, T. Sharpton, K. Pollard, and J. DeRisi. Profile hidden Markov models for the detection of viruses within metagenomic sequence data. *PLOS ONE*, 9, 2014.

[203] J. Söding. Protein homology detection by HMM-HMM comparison. *Bioinformatics*, 21(7):951–960, 2005.

[204] J. Söding, A. Biegert, and A. N. Lupas. The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Res.*, 33(Web Server issue):W244–8, jul 2005.

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

[206] A. Stamatakis. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–2690, 2006.

[207] M. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, 9:91–116, 1992.

[208] J. Stoye, D. Evers, and F. Meyer. Rose: generating sequence families. *Bioinformatics*, 14(2):157–63, jan 1998.

[209] J. W. Streicher, J. A. Schulte, and J. J. Wiens. How should genes and taxa be sampled for phylogenomic analyses with missing data? an empirical study in iguanian lizards. *Systematic Biology*, 65(1):128–145, 2016.

[210] M. Suchard and B. Redelings. BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny. *Bioinformatics*, 22(16):2047–2048, aug 2006.

[211] J. Sukumaran and M. Holder. Dendropy: a Python library for phylogenetic computing. *Bioinformatics*, 26(12):1569–71, 2010.

[212] S. Sunagawa, D. R. Mende, G. Zeller, F. Izquierdo-Carrasco, S. A. Berger, J. R. Kultima, L. P. Coelho, M. Arumugam, J. Tap, H. B. Nielsen, S. Rasmussen, S. Brunak, O. Pedersen, F. Guarner, W. M. de Vos, J. Wang, J. Li, J. Doré, S. D. Ehrlich, A. Stamatakis, and P. Bork. Metagenomic species profiling using universal phylogenetic marker genes. *Nat. Methods*, 10:1196–1199, oct 2013.

[213] D. Swofford. *PAUP*: Phylogenetic analysis using parsimony (* and other methods) Ver. 4*. Sinauer Associates, Sunderland, Massachusetts, 2002.

[214] K. Tamura and M. Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Mol. Biol. Evol.*, 10(3):512–526, may 1993.

[215] G. Tan, M. Muffato, C. Ledergerber, J. Herrero, N. Goldman, M. Gil, and C. Dessimoz. Current methods for automated filtering of multiple sequence alignments frequently worsen single-gene phylogenetic inference. *Systematic Biology*, 64(5):778–91, sep 2015.

[216] S. Tavaré. Some probabilistic and statistical problems in the analysis of DNA sequences. In *Lectures on Mathematics in the Life Sciences*, volume 17, pages 57–86. American Mathematical Society, 1986.

[217] M. Taylor, C. Kai, J. Kawai, P. Carninci, Y. Hayashizaki, and C. Semple. Heterotachy in mammalian promoter evolution. *PLOS Genet.*, 2(4):e30, 2006. doi:10.1371/journal.pgen.0020030.

[218] The UniProt Consortium. Uniprot: a hub for protein information. *Nucleic Acids Res.*, 43(D1):D204–D212, 2015.

[219] J. D. Thompson, B. Linard, O. Lecompte, and O. Poch. A comprehensive benchmark study of multiple sequence alignment methods: Current challenges and future perspectives. *PLoS ONE*, 6(3), 2011.

[220] J. D. Thompson, F. Plewniak, and O. Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res.*, 27(13):2682–2690, 1999.

[221] J. L. Thorne, H. Kishino, , and J. Felsenstein. An evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.*, 33:114–124, 1991.

[222] J. L. Thorne, H. Kishino, and J. Felsenstein. Erratum – an evolutionary model for maximum likelihood alignment of DNA sequences. *J. Mol. Evol.*, 34:91–91, 1992.

[223] J. L. Thorne, H. Kishino, and J. Felsenstein. Inching toward reality: An improved likelihood model of sequence evolution. *J. Mol. Evol.*, 34(1):3–16, 1992.

[224] D. Truong, E. Franzosa, T. Tickle, M. Scholz, G. Weingart, E. Pasolli, A. Tett, C. Huttenhower, and N. Segata. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nature methods*, 12(10):902–3, oct 2015.

[225] C. Tuffley and M. Steel. Links between maximum likelihood and maximum parsimony under a simple model of site substitution. *Bulletin of Mathematical Biology*, 59(3):581–607, 1997.

[226] E. R. Tufte. *The visual display of quantitative information*. Graphics Press, 1983.

[227] P. Vachaspati and T. Warnow. ASTRID: Accurate species trees from internode distances. *BMC Genomics*, 16(Suppl 10):S3, 2015.

[228] I. Van Walle, I. Lasters, and L. Wyns. SABmark–a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics*, 21(7):1267–1268, apr 2005.

[229] Y. Vázquez-Baeza, A. Gonzalez, Z. Xu, A. Washburne, H. Herfarth, R. Sartor, and R. Knight. Guiding longitudinal sampling in IBD cohorts. *Gut*, 67(9):1743–1745, sep 2018.

[230] L.-S. Wang, J. Leebens-Mack, P. K. Wall, K. Beckmann, C. W. de Pamphilis, and T. Warnow. The impact of multiple protein sequence alignment on phylogenetic estimation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(4):1108–1119, 2011.

[231] T. Warnow. Concatenation analyses in the presence of incomplete lineage sorting. *PLOS Currents: Tree of Life*, 2015. doi: 10.1371/currents.tol.8d41ac0f13d1abedf4c4a59f5d17b1f7.

[232] T. Warnow. *Computational Phylogenetics: an introduction to designing methods for phylogeny estimation*. Cambridge University Press, 2017.

[233] M. Wascher and L. Kubatko. Consistency of SVDQuartets and Maximum Likelihood for Coalescent-based Species Tree Estimation. *bioRxiv*, 2019.

[234] T. Wheeler and J. Kececioglu. Multiple alignment by aligning alignments. *Bioinformatics*, 23(13):i559–68, 2007.

[235] S. Whelan and N. Goldman. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Mol. Biol. Evol.*, 18(5):691–699, 2001.

[236] N. J. Wickett, S. Mirarab, N. Nguyen, T. Warnow, E. Carpenter, N. Matasci, S. Ayyampalayam, M. S. Barker, J. G. Burleigh, M. A. Gitzendanner, B. R. Ruhfel, E. Wafulal, J. P. Derl, S. W. Graham, S. Mathews, M. Melkonian, D. E. Soltis, P. S. Soltis, N. W. Miles, C. J. Rothfels, L. Pokorny, A. J. Shaw, L. DeGironimo, D. W. Stevenson, B. Sureko, J. C. Villarreal, B. Roure, H. Philippe, C. W. dePamphilis, T. Chen, M. K. Deyholos, R. S. Baucom, T. M. Kutchan, M. M. Augustin, J. Wang, Y. Zhang, Z. Tian, Z. Yan, X. Wu, X. Sun, G. K.-S. Wong, and J. Leebens-Mack. Phylotranscriptomic analysis of the origin and diversification of land plants. *Proceedings of the National Academy of Sciences*, 111(45):E4859–E4868, 2014.

[237] H. Wickham, D. Cook, H. Hofmann, and A. Buja. Graphical inference for infovis. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):973–979, nov 2010.

[238] J. Wiens. Missing data, incomplete taxa, and phylogenetic accuracy. *Systematic Biology*, 52:528–538, 2003.

[239] J. Wiens. Missing data and the design of phylogenetic analyses. *Journal of Biomedical Informatics*, 39:34–42, 2006.

[240] J. J. Wiens and M. C. Morrill. Missing data in phylogenetic analysis: reconciling results from simulations and empirical data. *Systematic Biology*, 2011.

[241] C. Wilke. Bringing molecules back into molecular evolution. *PLoS Computational Biology*, 8(6), 2012. https://doi.org/10.1371/journal.pcbi.1002572.

[242] D. Wood and S. Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15(3):R46, 2014.

[243] C. Worth, B. Esfahbod, et al. The cairo graphics library, version 1.15, 2018. https://www.cairographics.org/.

[244] C. H. Wu, H. Huang, L. S. L. Yeh, and W. C. Barker. Protein family classification and functional annotation. *Comp. Biol. Chem.*, 27(1):37–47, 2003.

[245] Z. Xi, L. Liu, and C. C. Davis. The Impact of Missing Data on Species Tree Estimation. *Molecular Biology and Evolution*, 33(3):838–860, 2016.

[246] Q. Xu and R. L. Dunbrack. Assignment of protein sequences to existing domain and family classification systems: Pfam and the PDB. *Bioinformatics*, 28(21):2763–2772, 2012.

[247] L. C. Xue, D. Dobbs, A. M. Bonvin, and V. Honavar. Computational prediction of protein interfaces: A review of data driven methods. *FEBS Lett.*, 589(23):3516–3526, 2015.

[248] S. Yamada, O. Gotoh, and H. Yamana. Improvement in accuracy of multiple sequence alignment using novel group-to-group sequence alignment algorithm with piecewise linear gap cost. *BMC Bioinform.*, 7:524, 2006.

[249] J. Yang and T. Warnow. Fast and accurate methods for phylogenomic analyses. *BMC Bioinformatics*, 12(Suppl 9):S4, 2011.

[250] S. Yildirim, C. Yeoman, S. Janga, S. Thomas, M. Ho, S. Leigh, P. Consortium, B. White, B. Wilson, and R. Stumpf. Primate vaginal microbiomes exhibit species specificity without universal Lactobacillus dominance. *The ISME Journal*, 8(12):2431–2444, dec 2014.

[251] C. Zhang, M. Rabiee, E. Sayyari, and S. Mirarab. ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. *BMC Bioinformatics*, 19(S6):153, may 2018.

[252] Y. Zhou, N. Rodrigue, N. Lartillot, and H. Philippe. Evaluation of the models handling heterotachy in phylogenetic inference. *BMC Evolutionary Biology*, 7:206, 2007.

[253] D. J. Zwickl and D. M. Hillis. Increased taxon sampling greatly reduces phylogenetic error. *Systematic Biology*, 51:588–598, 2002.