Novel Methods for Analyzing and Visualizing Phylogenetic Placements

zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

von der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte
Dissertation

von

Lucas Czech
aus Neuss

Tag der mündlichen Prüfung:	15.01.2020
Erster Gutachter:	Prof. Dr. Alexandros Stamatakis
Zweiter Gutachter:	Prof. Dr. Emmanuel Müller

Hiermit erkläre ich, dass ich diese Arbeit selbständig angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht habe. Ich habe die Satzung des Karlsruher Institutes für Technologie (KIT) zur Sicherung guter wissenschaftlicher Praxis beachtet.

Heidelberg, 15.01.2020

. . . . . .

(**Lucas Czech**)

Zusammenfassung

Die DNS (englisch: DNA) bildet die vererbbare Grundlage allen bekannten Lebens auf dem Planeten. Entsprechend wichtig ist ihre "Entschlüsselung" für die Biologie im Allgemeinen, und für die Erforschung der evolutionären Zusammenhänge verschiedener biologischer Artern im Besonderen. In den letzten Jahrzehnten hat eine rasante technologische Entwicklung im Bereich der DNS-Sequenzierung stattgefunden, die auch auf absehbare Zeit noch nicht zum Stillstand kommen wird. Die biologische Forschung hat daher den Bedarf an computer-gestützten Methoden erkannt, sowohl in Bezug auf die Speicherung und Verarbeitung der immensen Datenmengen, die bei der Sequenzierung anfallen, als auch in Bezug auf deren Analyse und Visualisierung.

Eine grundlegene Fragestellung ist dabei die nach dem Stammbaum des Lebens, der die evolutionäre Verwandtschaft der Arten beschreibt. Diese Wissenschaft wird Phylogenetik, und die resultierenden Strukturen phylogenetische Bäume genannt. Häufig basieren diese Bäume auf dem Vergleich von DNS-Sequenzen der Arten, mit der Idee, dass Arten mit ähnlicher DNS auch im Baum nah beieinander liegen. Die Berechnung eines solchen Baumes aus DNS-Daten kann als Optimierungsproblem formuliert werden, das durch die stetig wachsende Menge an Daten für die Informatik eine Herausforderung darstellt. Aktuell beschäftigt sich die Mikrobiologie zum Beispiel mit der Erkundung und Erforschung von Proben (Samples), die aus Meereswasser, dem Erdreich, dem menschlichen Körper, und ähnlichen Umgebungen gewonnen wurden: Welche mikrobischen Arten, Bakterien und andere Einzeller, bewohnen diese Umgebungen und Proben? Das zugehörige Forschungsfeld ist die Meta-Genetik. Einen verlässlichen Stammbaum für die aber-millionen an Sequenzen aus solchen Proben zu errechnen ist praktisch unmöglich. Eine Alternative bietet die phylogenetische Platzierung der Sequenzen auf einem gegebenen Referenz-Baum von bekannten Arten (so genanntes phylogenetisches Placement): Hierbei wird ein Stammbaum aus Referenz-Sequenzen bekannter Arten gewählt, der möglichst viel der in den Proben zu erwartenden Artenvielfalt abdeckt, und dann für jede Sequenz aus den Proben die nächste Verwandtschaft innerhalb des Baumes bestimmt. Dies resultiert in einer Zuordnung von Sequenzen auf die Positionen verwandter Arten im Referenz-Baum. Diese Zuordnung kann auch als Verteilung der Sequenzen auf dem Baum verstanden werden: In dieser Interpretation kann man beispielsweise erkennen, welche Arten (und deren Verwandtschaft) besonders häufig in den Proben vertreten sind.

Diese Arbeit beschäftigt sich mit neuen Methoden zur Vor- und Nachbereitung, Analyse, und Visualisierung rund um den Kernbereich des phylogenetischen Placements von DNS-Sequenzen. Zunächst stellen wir eine Methode vor, die einen geeigneten Referenz-Baum für die Platzierung liefern kann. Die Methode heißt *PhAT* (Phylogenetic Automatic (Reference) Trees), und nutzt Datenbanken bekannter DNS-Sequenzen, um geeigenete Referenz-Sequenzen für den Baum zu bestimmen. Die durch PhAT produzierten Bäume sind beispielsweise dann interessant, wenn die in den Proben zu erwartende Artenvielfalt noch nicht bekannt ist: In diesem Fall kann ein breiter Baum, der viele der bekannten Arten abdeckt, helfen, neue, unbekannte Arten zu entdecken. Im gleichen Kapitel stellen wir außerdem zwei Behilfs-Methoden vor, um den Prozess und die Berechnungen der Placements von großen Datensätzen zu beschleunigen und zu ermöglichen. Zum einen stellen wir Multilevel-Placement vor, mit dem besonders große Referenz-Bäume in kleinere, geschachtelte Bäume aufgeteilt werden können, um so schnellere und detalliertere Platzierungen vornehmen können, als auf einem einzelnen großen Baum möglich wären. Zum anderen beschreiben wir eine Pipeline, die durch geschickte Lastverteilung und Vermeidung von Duplikaten den Prozess weiter beschleunigen kann. Dies eignet sich insbesondere für große Datensätze von zu platzierenden Sequenzen, und hat die Berechnungen erst ermöglicht, die wir zum testen der im weiteren vorgestellten Methoden benötigt haben.

Im Anschluss stellen wir zwei Methoden vor, um die Placement-Ergebnisse verschiedener Proben miteinander zu vergleichen. Die Methoden, *Edge Dispersion* und *Edge Correlation*, visualisieren den Referenz-Baum derart, dass die in Bezug auf die Proben interessanten und relevanten Regionen des Baumes sichtbar werden. Edge Dispersion zeigt dabei Regionen, in denen sich die Häufigkeit der in den Proben vorhandenen mikrobischen Arten besonders stark zwischen den einzelnen Proben unterscheided. Dies kann als erste Erkundung von neuen Datensätzen dienen, und gibt Aufschluss über die Varianz der Häufigkeit bestimmter Arten. Edge Correlation hingegen bezieht zusätzlich Meta-Daten mit ein, die zu den Proben gesammelt wurden. Dadurch können beispielsweise Abhängigkeiten zwischen Häufigkeiten von Arten und Faktoren wie dem pH-Wert des Bodens oder dem Nitrat-Gehalt des Wassers, aus dem die Proben stammen, aufgezeigt werden. Es hat damit Ähnlichkeiten zu einer bestehenden Methode names Edge PCA, die ebenfalls relevante Regionen des Baumen identifizieren kann, allerdings die vorhandenen Meta-Daten nur indirekt einbeziehen kann.

Eine weitere Fragestellung ist die Gruppierung (Clustering) von Proben anhand von Gemeinsamkeiten, wie beispielweise einer ähnlichen Verteilungen der Sequenzen auf dem Referenz-Baum. Anhand geeigneter Distanz-Maße wie der Kantorovich-Rubinstein-Distanz (KR-Distanz) können Ähnlichkeiten zwischen Proben quantifiziert werden, und somit ein Clustering erstellt werden. Für große Datensätze mit hunderten und tausenden von einzlnen Proben stoßen bestehende Methoden für diesen Einsatzzweck, wie zum Beispiel das so genannte Squash Clustering, an ihre Grenzen. Wir haben daher die k-means-Methode derart erweitert, dass sie für Placement-Daten genutzt werden kann. Dazu präsentieren wir zwei Methoden, *Phy-*

logenetic k-means und *Imbalance k-means*, die verschiedene Distanzmaße zwischen Proben (KR-Distanz, und ein weiteres geeignetes Maß) nutzen, um Bäume mit ähnlichen Verteilungen von platzierten Sequenzen zu gruppieren. Sie betrachten jede Probe als einen Datenpunkt, und nutzen die zugrunde liegende Struktur des Referenz-Baumes für die Berechnungen. Mit diesen Methoden können auch Datensätze mit zehntausenden Proben verarbeitet werden, und Clusterings und Ähnlichkeiten von Proben erkannt und visualisiert werden.

Wir haben außerdem ein Konzept namens *Balances* für Placement-Daten adaptiert, welches ursprünglich für so genannte OTU-Sequenzen (Operational Taxonomic Units) entwickelt wurde. Balances erlauben eine Beschreibung des Referenz-Baumes und der darauf platzierten Sequenzen, die ganze Gruppen von Referenz-Arten zusammenfasst, statt jede Art einzeln in die Berechnungen einfließen zu lassen. Diese Beschreibung der Daten bietet verschiedene Vorteile für die darauf basierenden Analysen, wie zum Beispiel eine Robustheit gegenüber der exakten Wahl der Referenz-Sequenzen, und einer anschaulichen Beschreibung und Visualisierung der Ergebnisse. Insbesondere aus mathematischer Sicht sind Balances für die Analyse interessant, da sie problematische Artefakte aufgrund der kompositionellen Natur meta-genetischer Daten beheben. Im Zuge dieser Arbeit dienen Balances hauptsächlich als Zwischenschritt zur Daten-Repräsentation.

Eine Anwendung von Balances ist die so genannte *Phylofactorization*. Diese recht neue Methode teilt einen gegebenen Baum derart in Sub-Bäume ein, dass jeder Sub-Baum eine Gruppe von Arten darstellt, die in Bezug auf gegebene Meta-Daten pro Probe relevant sind. Dadurch können beispielsweise Gruppen identifiziert werden, deren evolutionäre Merkmale sich in Abhängigkeit von Meta-Daten wie pH-Wert angepasst haben im Vergleich zu anderen Gruppen. Dies ist ähnlich zur oben genannten Edge Correlation, aber kann zum einen durch geschickte mathematische Ansätze (insbesondere der Nutzung von Generalized Linear Models) mehrere Meta-Daten gleichzeitig einbeziehen, und zum anderen auch verschachtelte Gruppen finden. Die zugrunde liegenden Ideen dieser Methoden bieten einen großen Spielraum sowohl für Analysen von Daten, als auch für Weiterentwicklungen und Ergänzungen für verwandte Fragestellungen. Wir haben diese Methode für Placement-Daten adaptiert und erweitert, und stellen diese Variante, genannt *Placement-Factorization*, vor. Im Zuge dieser Adaption haben wir außerdem verschiedene ergänzende Berechnungen und Visalisierungen entwickelt, die auch für die ursprüngliche Phylofactorization nützlich sind.

Alle genannten neuen Methoden wurden ausführlich getestet in Bezug auf ihre Eignung zur Erforschung von mikrobiologischen Zusammenhängen. Wir haben dazu verschiedene bekannte Datzensätze von DNS-Sequenzen aus Wasser- und Bodenproben, sowie Proben des menschlichen Mikrobioms, verwendet und diese auf geeigneten Referenz-Bäumen platziert. Anhand dieser Daten haben wir zum einen die Plausibilität der durch unsere Analysen erzielten Ergebnisse geprüft, als auch Vergleiche der Ergebnisse mit ähnlichen, etablierten Methoden vorgenommen. Sämtliche Analysen, Visualisierungen, und Vergleiche werden in den jeweils entsprechenden Kapiteln vorgestellt, und die Ergebnisse dargestellt. Alle Tests zeigen, dass unsere Methoden

auf den getesteten Datensätzen zu Resultaten führen, die konsistent mit anderen Analysen sind, und geeignet sind, um neue biologische Erkenntnisse zu gewinnen.

Sämtliche hier vorgestellten Methoden sind in unserer Software-Bibliothek GENESIS implementiert, die wir im Zuge dieser Arbeit entwickelt haben. Die Bibliothek ist in modernem `C++11` geschrieben, hat einen modularen und funktions-orientierten Aufbau, ist auf Speichernutzung und Rechengeschwindigkeit optimiert, und nutzt vorhandene Multi-Prozessor-Umgebungen. Sie eignet sich daher sowohl für schnelle Tests von Prototypen, als auch zur Entwicklung von Analyse-Software für Endanwender. Wir haben GENESIS bereits erfolgreich in vielen unserer Projekte eingesetzt. Insbesondere bieten wir sämtliche hier präsentierten Methoden über unser Software-Tool GAPPA an, das intern auf GENESIS basiert. Das Tool stellt einen einfachen Kommandozeilen-Zugriff auf die vorhandenen Analysemethoden bereit, und bietet ausreichend Optionen für die Analysen der meisten End-Anwender.

Im abschließenden Kapitel wagen wir einen Ausblick in weitere Forschungsmöglichkeiten im Bereich der Methoden-Entwicklung für meta-genetische Fragestellungen im Allgemeinen, und der placement-basierten Methoden im Speziellen. Wir benennen verschiedene Herausforderungen in Bezug auf die Nutzbarkeit solcher Methoden für Anwender und ihrer Skalierbarkeit für immer größer werdende Datensätze. Außerdem schlagen wir verschiedene weitergehende Ansätze vor, die zum Beispiel auf neuronalen Netzwerken und Deep Learning basieren könnten. Mit aktuellen Datensätzen wären solche Methoden nicht robust trainierbar; durch das in Zukuft zu erwartenden Wachstum an Daten kann dies allerdings bald in den Bereich des Möglichen kommen. Schließlich identifizierenden wir einige tiefer gehende Forschungsfragen aus der Biologie und Medizin, bei deren Beantwortung unsere Methoden in Zukunft helfen können.

Abstract

The DNA is the hereditary basis of all known life on the planet. Deciphering this "code of life" is hence of key importance for biology in general, and for unravelling the evolutionary relationships between biological species in particular. The last few decades have seen rapid technological advances in DNA sequencing, with no slowdown of this trend being in sight. Research in biology hence has high demand for computational methods, both with respect to storage and processing of these huge datasets, and with respect to analysis and visualization thereof.

A basic concept in biology is that of the tree of life, which describes the evolutionary relationship between species. The respective field of science is called phylogenetics, and the resulting structures are called phylogenetic trees. Often, these trees are based on the comparison of DNA sequences of the species, and are build on the idea that species with similar sequences are located on nearby branches of the tree. The inference of such a tree based on DNA data can be formulated as an optimization problem, that poses a challenge for computer science due to the ever increasing amount of available data. For example, a current directive in micro-biology is to investigate the composition of samples taken from environments such as ocean water, soil, or the human body: Which microbial species, bacteria and other single cellular organisms, are present in these environments and samples? This field of research is called meta-genomics. It is infeasible to compute a robust phylogenetic tree for the millions of sequences obtained from such samples. An alternative approach is the so called phylogenetic placement of the sequences on a reference tree: Given a tree of reference sequences of known species that covers the expected diversity in the samples as much as possible, the evolutionary relationship of the sequences in the samples to the reference tree is determined. This yields a mapping from sequences to positions of related species in the reference tree. This mapping can also be understood as a distribution of sequences on the tree: This interpretation allows for example to visualize which species (and their next of kin) are frequently present in the samples.

In this work, we developed novel methods for pre- and post-processing, analysis, and visualization of phylogenetic placement of sequences. Firstly, we present a method to automatically obtain a suitable reference tree to be used for placement. The method is called *PhAT* (Phylogenetic Automatic (Reference) Trees), and uses databases of known DNA sequences in order to determine suitable reference sequences. The trees produced by PhAT are for example useful when the expected species diversity in the

samples is not yet known: In this case, a broad tree that covers many known species can help to discover novel, unknown species. In the same chapter, we also present two auxiliary methods that accelerate and enable the process and the computations needed for the placement of very large datasets. On the one hand, we present Multilevel-Placement, that uses a divide-and-conquer approach to split large reference trees into small, nested trees. It thereby improves speed and accuracy of the placement process compared to using one large reference tree. On the other hand, we describe a pipeline that maximizes load distribution and further accelerates the placement process by avoiding duplicate computations. This is particularly suited for large datasets, and was a necessary improvement to enable the computations needed for the tests of the further methods presented in this work.

Subsequently, we present two methods to compare the placement results of distinct samples with each other. The methods, *Edge Dispersion* and *Edge Correlation*, visualize the reference tree so that the interesting and relevant regions of the tree (with respect to the samples) become apparent. Edge Dispersion shows regions where the frequency of microbial species in the samples differs most in between samples. This can serve as a first exploration of a dataset, and indicates the variance of the occurrences of species. Edge Correlation on the other hand additionally takes meta-data into account that was collected per sample. It hence can for example show the dependencies between occurrences of species and environmental factors such as the pH-value of the soil, or the nitrate content of the water where each sample was taken from. This bears some similarity to an existing method called Edge PCA, which also highlights relevant regions of the reference tree, but can only indirectly incorporate meta-data features.

Another research question is that of grouping or clustering of samples based on similarities, for example a similar distribution of sequences on the reference tree. By using suitable distance measures such as the Kantorovich-Rubinstein distance (KR distance), similarities between samples can be quantized, and leveraged to cluster them. For large datasets with hundreds to thousands of distinct samples, existing methods for this purpose, such as the so called Squash Clustering, reach their scalability limits. We thus extended the k-means method to be applicable to placement data. To this end, we present two methods, *Phylogenetic k-means* and *Imbalance k-means*, that use two different distance measures between samples (KR distance, and another suitable measure) to cluster trees with similar distributions of placed sequences. These methods regard each sample as a distinct data item, and use the underlying structure of the reference tree for the computations. These methods can be applied to datasets with tens of thousands of samples, in order to find clusters and similarities between samples, and visualize these.

Furthermore, we adapted a concept called *Balances* to placement data, which was originally intended for so called OTU sequences (Operational Taxonomic Units). Balances allow for a description of the reference tree and the sequences placed on it in a way that summarizes groups of referece species, instead of taking each species into account individually. This description of the data offers several advantages for subsequent analysis steps; for example, it is robust in terms of the exact choice of

reference sequences, and offers an intuitive way of visualizing results obtained from these analyses. Balances are in particular helpful from a mathematical standpoint, as they circumvent problematic artifacts due to the compositional nature of metagenomic data. In this work, balances are mainly used as an intermediate step for data representation purposes.

One application of balances is the so called *Phylofactorization*. This relatively recent method splits a given tree into a set of sub-trees so that each sub-tree represents a group of species that are relevant with respect to the meta-data features of the given samples. This allows for example to identify groups whose evolutionary traits changed depending on meta-data such as pH-value in comparison to other groups. This is similar to the Edge Correlation method mentioned above, but further allows to incorporate several meta-data features at once and can find nested groups of species, by leveraging mathematical approaches such as Generalized Linear Models. The underlying concepts of Phylofactorization are versatile both for data analysis as well as for extension and adaptation to releated research questions. We have adapted the method to placement data, and present this variant, which we call *Placement-Factorization*. Additionally, we developed several auxiliary computations and visualizations of the results that are also useful for the original Phylofactorization.

All mentioned novel methods were extensively tested with respect to their suitability for discovering biological knowledge. To this end, we used several DNA sequence datasets from water and soil, as well as from the human body, and phylogenetically placed them on suitable reference trees. Based on this, we tested the plausibility of the results obtained from our analyses, and compared them to the results of simiar, established methods. All analyses, visualizations, and comparisons are described in detail in the respective chapters, along with the results and their interpretations. All tests show that our methods yield results on the datasets that are consistent with other types of analyses, and are suitable for discovering novel biological knowledge.

The methods presented here are implemented in our software library GENESIS, which we devloped alongside this work. The library is written in modern `C++11`, has a modular and function-oriented design, is optimized for memory consumption and computing speed, and leverages multi-core environments. It is hence suitable for rapid testing of prototype software, as well as for developing analysis software for end users. We already have successfully deployed GENESIS in several of our projects. In particular, all presented methods are incorporated into our command line tool GAPPA, which is internally based on GENESIS. The tool has a simple command line interface to our analysis methods that offers sufficient options for most end users.

In the final chapter, we dare an outlook into possible research directions for method development in meta-genetics in general, and placement-based methods in particular. We identify several challenges with respect to the usability of such methods for researchers, and their scalability to ever larger datasets. Furthermore, we suggest several further approaches, for instance based on neural networks and deep learning. With current datasets, such methods cannot robustly be trained; due to the expected growth of data in the near future however, such approaches are likely

to become feasible. Finally, we identify some in-depth research questions from the fields of biology and medicine for which our methods might be useful in the future.

Acknowledgments

"Evolution forged the entirety of [...] life on this planet using only one tool – the mistake."

— Dr. Robert Ford (Anthony Hopkins),
Westworld, Season 1: The Original

Trying new things, making mistakes, and thus "to err forward" are important parts of the scientific method. In the course of this work, I tried and learned quite a few new things; I am deeply grateful that I had the opportunity to make all the mistakes that come along with this. This would not have been possible without the support of a whole lot of marvelous people, to whom I wish to express my gratitude here.

First and foremost, I want to thank Prof. Alexandros Stamatakis for his excellent scientific supervision, for sharing his expertise, and for providing me with the freedom to grow and learn. It is rare to find an advisor who is so dedicated to guiding and supporting his students, while also being approachable on a human level.

Second, I am grateful to Prof. Emmanuel Müller, who agreed to be my second advisor and reviewer of this thesis. His support and interest in my work, as well as his contributions of ideas, helped to shape this work, and were the basis for many of the results presented here.

Furthermore, I wish to thank my colleagues at the Exelixis Lab for their support, for many hours of valuable discussions, be it at the white board or in private, and for making my time in the lab as enjoyable as it was: Jiajie Zhang, Tomáš Flouri, Paschalia Kapli, Andre Aberer, Kassian Kobert, Diego Darriba, Alexey Kozlov, Sarah Lutteropp, Benoit Morel, Rudolf Biczok, Dora Serdari, and Ben Bettisworth. In particular, I want to thank Pierre Barbera, with whom I shared a great many coffee breaks and other occasions to discuss software design questions, method development strategies, and also personal matters; without him, this work and my software would be of far inferior quality.

Similarly, I appreciate the scientific and non-scientific exchange I had with the people at the institute, namely, Kira Feldmann, Benjamin Heinzerling, Johannes Wagner, and Johannes Resin, as well as Christian Goll, Bernd Doser, Thomas Rasem, and Frauke Bley. Some of the most fruitful discussions were with Nikos Gianniotis and Kashif Sadiq; I really hope our collaborative ideas work out one day. Moreover, I am

happy to have shared my workplace and to exchange ideas with the visitors in our lab, Mark Holder, Emily McTavish, Rebecca Harris, Nikos Psonis, Mourad Elloumi, Khouloud Madhbouh, and Laura Rubinat-Ripoll.

I am also thankful to my collaborators and scientific colleagues, from whom I learned a lot and which helped conducting and shaping my research, in particular, Micah Dunthorn, Frédéric Mahé, David Bass, Cédric Berney, Colomban de Vargas, and Jaime Huerta-Cepas, but also Javier del Campo, Pelin Yilmaz, Christian Quast, Guillaume Lentendu, Torbjørn Rognes, Mahwash Jamy, Antonis Rokas, and Xiaofan Zhou. There were also several other researchers who kickstarted many ideas of this work and offered their help and advice when needed, particularly, Alex Washburne, Justin Silverman, Michael Robeson, Sujatha Srinivasan, Frederick Matsen, Gavin Douglas, and Lionel Guidi; thank you for your inspiration and initiative. I am also happy that my paths crossed with Nick Goldman, Adam Leaché, Brian Moore, Ben Redelings, Asif Tamuri, James Pease, Ziheng Yang, Emmanouela Karameta, David Matten, and Sandra Alvarez Carretero, and all the other instructors and participants at the Computational Molecular Evolution courses.

On a more personal note, I want to thank my family, my parents Peter and Maria and my sister Judith, who not only constantly supported me during this thesis, but through all of my years of study in Karlsruhe, Heidelberg and around the world. Also, I want to thank Malina Graf and Julia Klawitter, as well as my friends in Heidelberg and all other places, for their support and understanding. More thanks go to David Dao, who pointed out the opportunity to work at the institute to me, as well as Andreas Veit, who is a scientific inspiration to me. I also want to thank the people of the Heidelberg Unseminars in Bioinformatics for the opportunities they gave me.

Furthermore, I wish to thank Richard Dawkins, whose books inspired me at the exactly right time in my life, as well as Phil Collins and Peter Gabriel, who provided the background music for this thesis. A special thanks to `xkcd` and Randall Munroe for enlightening my journey through academia. I feel I must also thank the open source software community, as well as the people and journals supporting open access publication; you help to put knowledge and science where it belongs: in the public hand. We can only stand on the shoulders of giants if those are not protected by pay-walls.

Finally, I would like to to express my gratitude towards Klaus Tschira, the Klaus Tschira Stiftung, and in particular, the Heidelberg Institute for Theoretical Studies, both for funding my position as well as providing an excellent, rewarding, and fun work environment. It was a pleasure to work with and alongside so many lovely people.

Contents

List of Figures

List of Tables

List of Acronyms

API	Application Programming Interface
bp	Base Pair
BT	Backbone Tree
BV	Bacterial Vaginosis
CLV	Conditional Likelihood Vector
CT	Clade Tree
DNA	Deoxyribonucleic Acid
EDPL	Expected Distance between Placement Locations
FPA	Felsenstein Pruning Algorithm
GB	Giga-Byte
GLM	Generalized Linear Model
GTR	Generalized Time-Reversible (Model)
HMP	Human Microbiome Project
KR	Kantorovich-Rubinstein (Distance)
LWR	Likelihood Weight Ratio
MB	Mega-Byte
MC	Markov Chain
MDS	Multidimensional Scaling
ML	Maximum Likelihood
MPI	Message Passing Interface
MSA	Multiple Sequence Alignment
NGS	Next Generation Sequencing
OTU	Operational Taxonomic Unit
PCA	Principal Component Analysis

PCR	Polymerase Chain Reaction
PhAT	Phylogenetic Automatic (Reference) Tree
QS	Query Sequence
RA	Reference Alignment
RF	Robinson-Foulds (Distance)
RNA	Ribonucleic Acid
rRNA	Ribosomal Ribonucleic Acid
RT	Reference Tree
SSU	Short Subunit
TO	Tara Oceans
UPGMA	Unweighted Pair Group Method with Arithmetic Mean

1. Introduction

1.1 Background and Motivation

The concept of evolution is one of the cornerstones of modern biology [84]. All life on earth descends from a common ancestor and continuously evolves and adapts over generations, which leads to a diversification of biological species. The resulting branching pattern of the evolutionary relationships between species is the key for unraveling many biological questions, ranging from paleontology [317] to medicine [145]. These evolutionary relationships are described by *phylogenetic trees* (Section 2.3), which are important in both fundamental [167, 261, 398] and applied research [118, 150, 321].

Characteristics and traits of biological species are inherited via their DNA (Section 2.1). DNA data is hence often used for inferring a phylogenetic tree for a set of species (Section 2.4). In order to conduct such analyses, the DNA has to be sequenced, that is, it has to be "read" into some human-accessible format, typically in form of a sequence of characters (Section 2.2). In recent decades, the throughput of sequencing technologies has increased substantially, while at the same time, the cost has decreased faster than Moore's law, as shown in Figure 1.1. Currently, the sequencing capacity doubles roughly every seven months [345]. This lead to a "tsunami" of sequence data, which constitutes a major challenge for conducting computational analyses of these data.

In particular, these high-throughput technologies allow to directly sequence DNA contained in samples that have been extracted from environments such as water, soil, or the human gut. This results in so-called *metagenomic* sequences, that is, anonymous DNA sequences from the (microbial) organisms that were present in the environmental sample. A key question in the analysis of such data is to determine the evolutionary relationships of the sequences. While these DNA sequences can hypothetically be used to infer phylogenetic trees from scratch, this approach is limited by several theoretical and practical difficulties: For instance, typical metagenomic

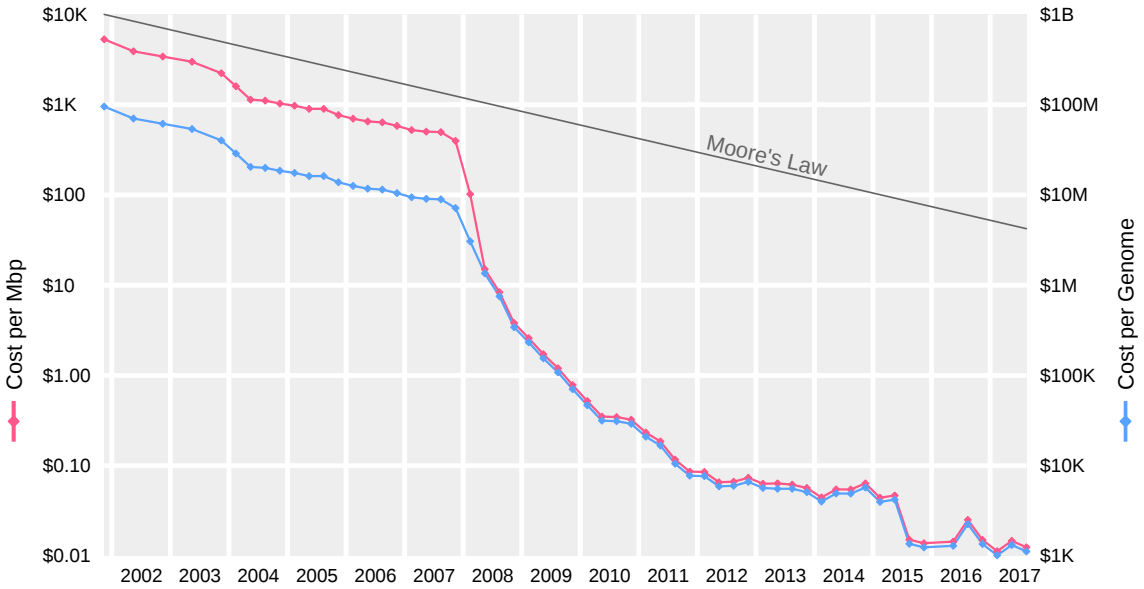

Figure 1.1: Sequencing costs per Mbp and per genome. The cost for DNA sequencing have decreased substantially in the past 15 years. The figure shows the cost per mega-basepair (Mbp) of DNA (red line, left y-axis) as well as the cost per human-sized genome of ≈ 3 Gbp (blue line, right y-axis). A basepair represents one character in the DNA. Note the logarithmic scaling of the y-axes. For comparison, Moore's law [263] is shown. The particularly steep decrease in the beginning of 2008 is caused by the adoption of novel (high-throughput) sequencing technologies in sequencing centers, see Section 2.2.1. Source: Image based on data from [383].

samples contain too many, and too short, sequences for a feasible and reliable tree inference [166, 241].

One approach to tackle this issue is to deploy so-called *phylogenetic placement* [25, 241] of metagenomic sequences on a given phylogenetic tree (Section 2.5). Phylogenetic placement methods classify a set of *query sequences* into the context of known evolutionary relationships, provided in form of a *phylogenetic reference tree*. While this information already represents biological knowledge per se, it can also be used for further downstream analyses [239]. Although such phylogeny-aware methods offer large potential for sequence data analysis and interpretation, the research in this field is relatively recent and only a few such analysis methods have been developed so far.

An important task prior to conducting phylogenetic placement of metagenomic sequences is to obtain a suitable phylogenetic reference tree that captures the biological diversity of the species to be placed. The assembly of a set of reference sequences from biological databases that can be used to infer such a tree is typically a manual process, and hence both labor-intensive and potentially error-prone. This might detain researchers from employing phylogenetic placement in the first place, and make them resort to simpler methods based on sequence similarity/identity instead.

Furthermore, while the existing downstream analysis methods for phylogenetic placement (introduced in Section 2.5.5) allow for in-depth interpretation and visualization of the data, they were not developed with a particular focus on large-scale studies comprising thousands of environmental samples and billions of sequences. For large datasets, these methods might provide too much detail, making it hard to interpret results, to detect patterns or clusters in the data, and to discover correlations with per-sample meta-data.

Lastly, the problem of scalability on large datasets does not only affect the methods themselves. Because of the ever growing amount of sequence data, scalability is becoming an issue for the software pipelines as well. State-of-the-art phylogenetic placement implementations can process billions of sequences within a few hours [18]. Methods for handling and analyzing the data, in particular phylogenetic placement data, hence require efficient and scalable software implementations.

1.2 Scientific Contribution

This thesis makes several contributions to the field of computational phylogenetics, specifically regarding phylogenetic placements as well as analyzing and visualizing the resulting data. In particular, we introduce multiple novel methods to overcome the issues and limitations explained above. We described the methods in two already peer-reviewed open-access publications [67, 69], and made their data and scripts available at `http://github.com/lczech/placement-methods-paper`. We provide an overview of these methods in Section 1.3; more details on the contribution of each method to the research community are provided in the respective chapters.

Apart from the two publications on which this thesis is based, we published a review on several software tools for visualizing phylogenetic trees [68]. In this publication, we investigated whether certain types of per-branch and per-node meta-data of phylogenetic trees were correctly displayed in 20 distinct widely used tree visualization tools. We found that most of the tested tools exhibited problems or undocumented behavior and did not properly support the `Newick` file format for phylogenetic trees. We also showed that this has already affected trees published in peer-reviewed journals. At the time of its publication, our review had already led to improvements in eight of the twenty tested tools.

In addition to these theoretical projects, we also contributed to several empirical data analysis studies by conducting established analyses as well as testing prototypes of our novel methods presented here. In particular, we ran the phylogenetic analyses for a study of 154 locations in neotropical rainforest soils, which was published in *Nature Ecology & Evolution* [230]. The study found that the microbial diversity in these soils is dominated by hyper-diverse protistic parasites. For this project, we developed prototypes of our multilevel placement approach (Section 3.2.2) as well as of some visualization techniques (Section 4.1).

Further data analysis studies that we contributed to include the 1KITE project [261] (`http://1kite.org`), for which we conducted phylogenetic tree reconstructions for

a diverse set of 1500 insects, as well as a study of *Microsporidia* and *Cryptomycota* [20], to which we contributed phylogenetic placement analyses in order to resolve some branches in the phylogenetic tree of these groups of microbial *Eukaryota*. An ongoing large-scale endeavor is the UniEuk project [27], which aims to offer a unified reference database for the *Eukaryota*. For UniEuk, we provided consultancy and helped in planning workflows and pipelines. Furthermore, for its sub-project EukBank, we conducted preliminary phylogenetic analyses of their entire metagenomic database as a showcase for other researchers. In a current empirical data analysis project, we develop tools for analyzing a dataset of microbial *Eukaryota* [164]. The goal is to show that novel sequencing technologies that yield longer sequences per run can improve taxonomic and phylogenetic analyses compared to other sequencing technologies. Lastly, we are currently working on an empirical dataset of *Dinoflagellates*, for which we also conducted phylogenetic placement analyses.

We implemented all of our novel methods in `C++11`, and provide the resulting code in our open-source library GENESIS (`https://github.com/lczech/genesis`). Apart from the novel methods, the library provides efficient re-implementations of existing methods, for example the Edge PCA and Squash Clustering methods of GUPPY [241], which we introduce in Section 2.5.5. Our implementations are faster than the original one by orders of magnitude [69]. Furthermore, the library also offers a multitude of data structures and functions for working with phylogenetic placements, genetic sequences, phylogenetic trees, taxonomies, and many other data types. It also provides a plethora of auxiliary functions for tasks such as visualization, statistical evaluation, and data storage. Moreover, in order to also offer a user-friendly command line interface for the most important novel and established methods, we developed the open-source tool GAPPA (`https://github.com/lczech/gappa`), which stands for "Genesis Applications for Phylogenetic Placement Analysis". It is intended for researchers who desire to conduct analyses using our methods. It internally uses the GENESIS library for its computations. For details on the software implementations, see Appendix C. We also published an application note [70], which describes GENESIS and GAPPA in detail, and evaluates their runtime and memory requirements in comparison to analoguous software.

Moreover, we contributed to several other open-source bioinformatics pipelines and software projects. We helped to develop an efficient approach for merging so-called *paired-end reads* and provided SIMD (single instruction, multiple data) implementations using `SSE` and `AVX` instructions to accelerate the merging algorithm [116]. Again for acceleration, we implemented a custom `MPI` wrapper for PaPaRa 2.0 [23, 24], which we made available at `https://github.com/lczech/papara_nt`. PaPaRa is a tool for aligning query sequences to a given reference alignment and phylogenetic tree, which is a necessary pre-processing step for phylogenetic placement. During the development of the recent high-performance re-implementation of the phylogenetic placement algorithm in EPA-NG [18], we provided support for software design decisions and implementation details. Also, our GENESIS library was used and extended for this project. We moreover contributed to the implementation and acceleration of a novel quartet-based method to accurately and robustly mea-

sure incongruence between phylogenetic trees [403]; the GENESIS library was used for this project as well. Furthermore, we contributed code to the sequence clustering tool SWARM [228, 229], which we briefly introduce in Section 2.2.5. Our code was the basis for accelerating the runtime of the tool by factors of up to 20-fold, while at the same time significantly reducing its memory requirements. These improvements will be part of the upcoming version 3 of SWARM. Lastly, we are currently working on SCRAPP, a pipeline that combines several tools developed in our lab, such as EPA-NG [18], PARGENES [265], and MPTP [173], in order to estimate the species diversity of metagenomic sequence samples based on phylogenetic placement data; see also Chapter 8.

1.3 Structure and Overview

The remainder of this thesis is structured as follows. Initially, we introduce the general concepts and existing methods of computational phylogenetics and phylogenetic placement (Chapter 2). In subsequent chapters, we describe our novel methods for conducting and analyzing phylogenetic placements: Firstly, we describe an automated approach for obtaining suitable reference trees for phylogenetic placement, as well as pre-processing pipelines to accelerate and enable phylogenetic placement of large metagenomic datasets (Chapter 3). Secondly, we introduce methods to visualize such large datasets in order to detect patterns within the data and correlations with per-sample meta-data (Chapter 4). Then, we present an approach for clustering metagenomic samples by measuring similarity between samples in terms of the species diversity they contain (Chapter 5). In between, we describe an adaptation of a novel type of data representation to phylogenetic placement data (Chapter 6). Lastly, we present an adaptation of a recent method called Phylofactorization to phylogenetic placement data (Chapter 7), which allows to find parts of a given reference tree that are meaningful with respect to meta-data features collected per environmental sample. Finally, we conclude and discuss potential directions of future work (Chapter 8). We furthermore provide additional supporting information (Appendix A), an overview of the empirical datasets used for our evaluations and their pre-processing (Appendix B), as well as some information on our software implementation (Appendix C).

2. Foundations

This chapter mostly contains original contributions by Lucas Czech that were created for this thesis. Some of the text and figures introducing phylogenetic placements are however derived from the introductory sections of the following peer-reviewed open-access publications:

> **Lucas Czech**, Pierre Barbera, and Alexandros Stamatakis. "Methods for Automatic Reference Trees and Multilevel Phylogenetic Placement." *Bioinformatics*, 2018, Volume 35, Issue 7, Pages 1151–1158.
>
> **Lucas Czech** and Alexandros Stamatakis. "Scalable Methods for Analyzing and Visualizing Phylogenetic Placement of Metagenomic Samples." *PLOS ONE*, 2019, Volume 14, Issue 5, Page e0217050.

The respective texts and figures were originally created by Lucas Czech, and modified here to fit into the context of this chapter. Furthermore, some of the introductory figures in this chapter are derived from permissively licensed work of others, as stated below the respective figures.

2.1 Evolution and Genetics

Life on Earth is at least 3.77 billion years old [85], and is continuously evolving due to *natural selection* [73]. Driven by *variation*, biological populations diversify through successive generations, leading to the emergence of new species. This continuous process is called *evolution* [142]. Heritable characteristics are passed down from parent to offspring, with occasional random mutations leading to variation. Thus, some organisms are better adapted to their environment than others, and have greater reproductive success. There is hence a natural selection for advantageous mutations, which can subsequently spread through generations.

The characteristics and traits of an organism are carried by, and inherited via, the *deoxyribonucleic acid* (DNA). The DNA is the molecule that encodes the genetic information needed for the functioning of all living organisms. It is structured in form of a double helix [379], and built from two strands of molecules called *nucleotides*. The nucleotides form the backbone of the double helix, and connect the two strands via opposing pairs of *nucleobases*, see Figure 2.1(a). The redundant structure of pairs of nucleobases stabilizes the DNA molecule, and also serves as a mechanism of error correction when reading the genetic information during cellular processes. In the DNA, there are four distinct nucleobases: adenine (`A`), cytosine (`C`), guanine (`G`), and thymine (`T`), where `A` pairs with `T`, and `C` pairs with `G`, respectively, as shown in Figure 2.1(b).

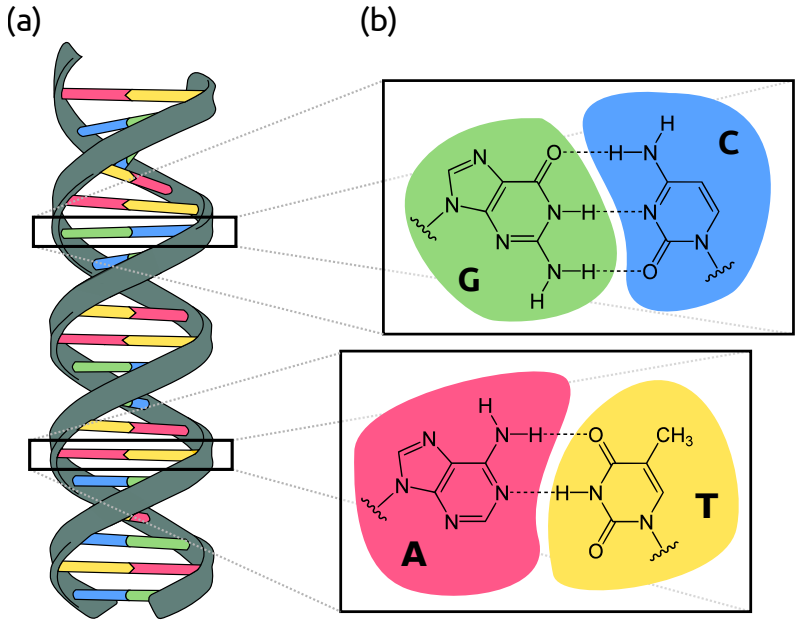

Figure 2.1: DNA double helix and nucleobases. (a) The double helix structure of the DNA, with the backbone in gray, connected by pairs of nucleobases. (b) The atomic structure of the four nucleobases, and their connection to each other. Source and license: see [66], image derived from [183, 338, 393, 394].

The sequence of nucleobases along the strands of DNA is what encodes the genetic information used by all known living organisms. Parts of the DNA encode for proteins, which perform a plethora of different and important functions within organisms. Proteins consist of long chains of amino acid residues, and are assembled in a process called protein (bio)synthesis. This is described by the central dogma of molecular biology [62, 63]: First, DNA is *transcribed* into the intermediate ribonucleic acid (RNA), which is then *translated* into the actual protein.

In each step of this process, the molecular alphabet used to encode information is different. While DNA uses the four nucleobases as described above, in RNA, the nucleobase uracil (`U`) is used instead of thymine (`T`). Proteins on the other hand are mostly built from a set of 20 *standard* amino acids, with the exception of two

non-standard amino acids that are not directly encoded in the DNA. The set of rules used by the molecular machinery for translating nucleobases into amino acids is called the *genetic code*: In a DNA sequence, three consecutive nucleobases are needed to encode one amino acid [328].

The entirety of the genetic material of an organism, that is, its complete DNA sequence, is called its *genome*. A *gene* is a sequence which codes for a molecule that has a particular function, such as a protein [121]. The DNA and the genes are the basic units of heredity. They vary across generations, and are under selection in the process of evolution [74]. The study of genes, their variation and heredity is called *genetics* [135].

2.2 Sequence Analysis

All life on this planet is related to each other and descends from a common ancestor. Still, it is remarkable that the basic molecular principles and mechanisms of life—DNA, amino acids, and the genetic code—are virtually identical for all living organisms. This implies that by understanding and comparing the genetic information encoded in the genetic sequences of different organisms, one can understand the diversification patterns of evolution [405].

2.2.1 Genome Sequencing

Prior to analyzing the DNA of an organism, the physical order of nucleobases in the DNA molecule has to be determined. That is, the DNA has to be "read" and stored in a human-accessible text format, typically a computer file. This technical process is called DNA *sequencing*.

For many decades, the main technique for this purpose was Sanger sequencing [311, 312]. It is labor- and time-intensive, but through optimization and automation, costs were constantly reduced. Eventually, this allowed for large-scale sequencing projects, such as the Human Genome Project [362], which sequenced the whole human genome comprising more than three billion nucleobases. Sanger sequencing allows to determine long parts of the sequence at once (> 500 nucleobases), which then have to be *assembled* to build the final genome sequence.

In the last decades, a variety of novel *high-throughput sequencing* (HTS) technologies have been developed [131, 289, 303]. In particular, *Next Generation Sequencing* (NGS) technologies [218, 234] have revolutionized biology by transforming it into a data-driven and compute-intense discipline [103]. In recent years, *Third* and *Fourth Generation* technologies have lead to even more substantial improvements [147, 256, 273, 280]. The costs of these technologies are decreasing faster than Moore's law [263, 383], as shown in Figure 1.1, while at the same time, their throughput (in terms of sequences being produced) is increasing. This leads to an ever growing amount of sequence data, which poses a challenge for computational methods analyzing these data. Compared to Sanger sequencing, NGS technologies are generally cheaper and faster [251, 371], but come at the price of introducing more errors in the sequence

output, or only being able to determine shorter parts of the sequence at once. Both limitations constitute a challenge for the subsequent assembly of the final sequence.

The result of DNA sequencing is a textual representation of the order of nucleobases. Although this representation ignores the physical and chemical properties of the respective molecules, it is helpful in many applications, and allows to leverage existing algorithms. Each contiguous sequence coming from the sequencing machine is called a *read*. Reads are typically stored in the `fastq` file format [57]. Because of the pairing of nucleobases, both DNA strands can be sequenced. These so-called *paired-end* reads provide a means of error correction and can be merged to form a final sequence representation [401]. The sequence representation of a read consists of the characters `A`, `C`, `G`, and `T`. The resulting set of sequences is stored in formats such as the `fasta` file format [287]. Due to the pairing of nucleobases, the length of a DNA sequence is measured in *base pairs* (abbreviated bp): 1 bp represents one character in the file. The `fastq` or `fasta` files are then used as input for computational methods for analyzing DNA sequences.

2.2.2 Metagenomics

Sanger sequencing requires careful preparation of the genetic material, and is thus best suited for sequencing single organisms. There are however many (microbial) organisms that cannot be cultured in the laboratory, and are hence hard to sequence with this technique. Apart from being cheaper, high-throughput sequencing machines also "digest" all genetic material given to them. They thus allow for directly studying microbial samples that have been extracted from their environment [97, 266, 349]. This enables to study environments such as water [123, 134, 174], soil [92, 230], the human body [160, 238, 250, 375], and many others. Each sample from such an environment represents a geographical location, a body site, a point in time, etc. The DNA of all organisms present in a sample is sequenced, resulting in a large number of reads per sample. These reads are anonymous, as it is unclear to which organism they belonged to. The study of these data, that is, the genetic material from environmental samples, is called *metagenomics* [277].

A first step in metagenomic studies often consists in characterizing the reads obtained from an environment with respect to *reference sequences* of known species [155]. Reads that are similar to (parts of) these reference sequences can be assigned to them, while reads with low similarity to known sequences might originate from novel, undescribed species [285, 354]. Key tasks in metagenomic studies are the identification and classification of the anonymous reads ("Who is there?"), and their functional annotation ("What are they doing?") [81, 211, 285]. Typical pipelines for both tasks are summarized in Figure 2.2 and described in more detail in the following.

Functional annotation [343] is the prediction of the genetic functions of the sequences, and the inference of metabolic capacity of microbial communities [41]. As the proteins that are needed in the pathways of such functions can be encoded by genes across the genome, whole-genome sequencing is necessary to capture all genes

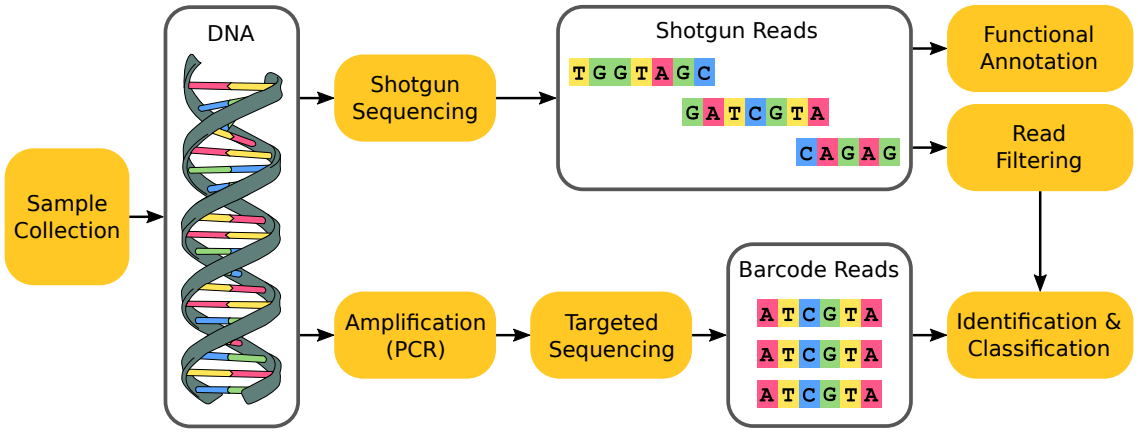

Figure 2.2: Typical metagenomic sequencing pipelines. After collecting a set of metagenomic samples from an environment such as soil, water, or the human body, there are two commonly used approaches for sequencing the DNA contained in the samples: (i) In shotgun sequencing, the DNA of the whole genome is fragmented into pieces, resulting in reads that can be used for the prediction and annotation of gene functions, or filtered for reads from a specific region. (ii) In targeted sequencing, the DNA of a specific barcode region is first amplified via the PCR process, yielding copies of the reads from the region which are then sequenced. Reads from certain regions can be used for identification and classification of the organisms that were present in the samples.

of interest. For example, in shotgun sequencing [10, 340], the DNA is fragmented into small pieces within the size range that the sequencing technology that is deployed can handle (typically, a few hundred bp). This allows to sequence all genetic material contained in a sample. Thus, the resulting reads originate from different parts of the genomes of their corresponding organisms, which can then be functionally annotated [125]. This however necessitates the use of whole-genome reference sequences in order to be able to assign reads to known species and functions. Typical databases of reference sequences however lack many of the protein sequences from the microbial species present in a sample, mostly because of the existence of organisms that cannot be cultured [41].

For the task of identification and classification of reads however, whole genome references are not needed. Instead, specific *marker genes* can be targeted, which are gene regions that are known to be particularly well-suited for differentiating between different species [302]. The use of marker genes to identify species is called *DNA (meta-)barcoding* [80, 148, 314]; the technological method is called *targeted sequencing*. The choice of genes to use as markers is important, and depends on the types of organisms to be studied. A marker gene should ideally be present in all organisms of interest, short enough to be sequenced with current technology, and exhibit sufficient between-species variation to distinguish them from each other, and finally show low within-species variation [194].

In many metagenomic studies of *Bacteria* and *Eukaryota*, the 16S [381] and 18S [252] rRNA regions are used as marker genes, respectively [385, 386]. These regions belong to the small subunit (SSU) of the ribosomal ribonucleic acid (rRNA), which is an essential component of the ribosome. The ribosome is a molecular machinery that is responsible for protein synthesis (translation) in all living organisms. Often, prior to sequencing, these regions are amplified by many orders of magnitude, using the polymerase chain reaction (PCR) to create a sufficient number of copies of these regions for sequencing [19]. The resulting reads are then de-replicated again, which results in sequences called *amplicons*. The PCR amplification process however is known to introduce biases [41, 219]. For example, the relative abundances of the final amplicons do not necessarily reflect the original ratio of the input gene regions [171, 208], which can be problematic in comparative studies (see also Section 2.5.3). Still, this inexpensive method is commonly used in practice, particularly for the 16S and 18S rDNA regions. We later discuss implications and solutions to normalizing these abundances in Section 2.5.3.

An alternative to using PCR for obtaining reads from barcode regions is as follows. First, shotgun sequencing is used to obtain reads from the entire genome of the organisms in the sample. These reads are then filtered to only contain reads from the target region, e. g., the 16S region for *Bacteria*, which capture the diversity of the sample without bias. Such an approach are for example the so-called $_{\text{mi}}$tags [219].

Because of the ubiquity of the 16S and 18S rDNA regions in organisms and, consequently, in sequencing studies, many databases provide reference sequences of these marker regions for known species. The reads or amplicons obtained from an environmental sample can then be employed to estimate the composition and diversity of the microbial community [157, 285], for example by comparison against the known species in databases.

2.2.3 Sequence Alignment

Organisms that evolved from a (not too distant) common ancestor share genetic information. Regions of their DNA that have a shared ancestry are called *homologous* regions [184]. This homology is typically inferred from sequence similarity. However, due to mutations, differences in the sequences can occur. There are three main types of sequence mutations that can occur in the course of evolution: a *substitution* is the exchange of a nucleobase for another; an *insertion* inserts one or more extra nucleotides into the sequence; a *deletion* removes one or more nucleotides from the sequence. The latter two types of mutations change the length of the sequence; a mutation that is either an insertion or a deletion is called an *indel*.

Because of indels, sequences have to be aligned to each other in order to establish and compare their homologous regions. That is, gap characters (-) have to be added to the sequences such that homologous characters in the sequence get aligned to each other. This results in an $n \times m$ matrix, where n is the number of sequences (rows), and m is the number of homologous characters (columns), called *sites*. This matrix

is called a *multiple sequence alignment* (MSA), or simply an *alignment*. Figure 2.3 shows an example of the alignment process and the resulting MSA.

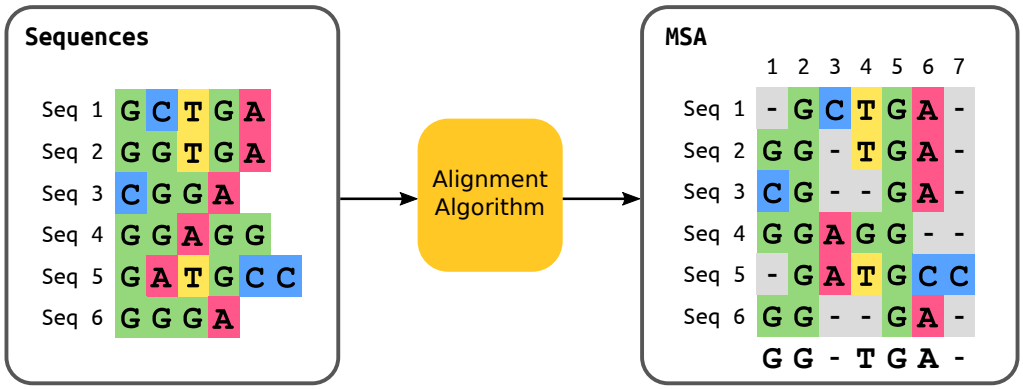

Figure 2.3: Multiple Sequence Alignment. The left hand side shows a set of six sequences. Using an alignment algorithm, gaps are inserted into these sequences at presumed indel positions. The right hand side shows the result of this process, where homologous characters at the sites of the multiple sequence alignment (MSA) are aligned to each other. Below the MSA, the majority rule consensus sequence is shown, see Section 2.2.4.

Sequence alignment can be understood as an optimization problem under a given optimality criterion. Identifying a suitable criterion is however often an empirical rather than a theoretical issue [47, 369]. On the one hand, *global alignments* attempt to align every character in every sequence, which is most useful for similar sequences of roughly equal size. For example, the Needleman-Wunsch algorithm [268] is a general global alignment technique based on dynamic programming, and appropriate for the pairwise alignment of two sequences. On the other hand, *local alignments* are better suited for dissimilar sequences which might contain similar regions within a larger sequence context. The Smith-Waterman algorithm [332] is a general local alignment technique using the same dynamic programming scheme, which additionally allows to start and end at any place in the sequence, and again is most appropriate for pairwise alignments. As both algorithms have their particular use cases [291], hybrid methods have also been developed [42]. Furthermore, for related tasks such as searching and clustering of similar sequences, heuristic approaches such as BLAST [8] and USEARCH [96] can calculate millions of near-optimal alignments in reasonable time.

These algorithms are efficient for the pairwise alignment of two sequences. Calculating an MSA however has been shown to be NP-hard [170, 374]. Thus, for most empirical datasets, other approaches and heuristics are needed [355]. Tools such as CLUSTAL [151], MUSCLE [95], and MAFFT [176] can calculate multiple sequence alignments for many thousands of sequences. Their output is typically stored in file formats such as `fasta` [287] or `phylip` [109].

A special use case for aligning sequences arises in metagenomic studies, where environmental sequences are often compared to a set of known reference sequences.

In these studies, one often first calculates (or is given) an MSA of the reference sequences, and then successively aligns the environmental sequences against this MSA. This is because calculating an MSA for millions or billions of sequences from scratch is too expensive even for modern tools. Hence, specialized algorithms for this use case have been developed, such as PaPaRa [23, 24] and hmmalign, which is a subprogram of the HMMER suite [93, 94].

2.2.4 Consensus Sequences

When working with a number of related but not identical sequences, it is often convenient to "summarize" homologous characters in form of a *consensus sequence*. Such a sequence is typically calculated based on the relative character frequencies per alignment site. It then represents typical features and motifs of the input sequence set.

The most straight-forward method is to construct *majority rule consensus* sequences [76, 245], where each site is represented by the most frequent character at that site. Figure 2.3 shows a corresponding example below the MSA on the right hand side. In order to also include information from the less frequent characters at a site in the consensus sequence, *ambiguity characters* can be used [161]. They allow to denote multiple alternative nucleobases as a single character. For example, if the nucleobases `A` and `G` are similarly frequent at a site, this site can be represented by the ambiguity character `R`. See Table A.1 for the full list of character representations.

Using ambiguity characters allows for more involved consensus methods. For example, *threshold consensus* sequences [75, 76] include the most frequent characters that are needed to achieve some given frequency threshold per site, and represent these characters by their ambiguity character. Furthermore, many methods based on fixed thresholds have been proposed, such as Cavener's method [50, 51]; see Day and McMorris (1992) [76] for a review and comparison.

It is also possible (yet not common) to directly use the relative per-site character frequencies in the mathematical frameworks of many downstream analysis methods. This allows to leverage all of the information contained in the input set of sequences. However, to our knowledge, there is no standard file format to store such information, and consequently, no way of forwarding this information to the respective tools.

2.2.5 Operational Taxonomic Units

The metagenomic sequences obtained from environments such as soil or water are usually anonymous (Section 2.2.2), as it is unclear to which organisms they originally belonged to [277]. Furthermore, for most microbial organisms, traditional systems of biological classification (Section 2.3.1) cannot be readily applied [31]. In many studies, a pragmatic solution is thus to group closely releated individuals into so-called *operational taxonomic units* (OTUs) [334] instead. Most commonly, sequences are grouped by similarity of a specific marker gene, such as 16S or 18S [31]. This allows to define a microbial "species" concept, where OTUs represent proxy units for assessing microbial (species) diversity through genetic (sequence) diversity. These

units (the OTUs) are then defined by the—mostly arbitrary—similarity threshold employed by the researcher; typically, 97% similarity is used.

There are different algorithms and methods for clustering sequences into OTUs, see Chen et al. (2013) [54] for a review. Common tools for OTU clustering are UCLUST [96], VSEARCH [306], and SWARM [228, 229]. SWARM works differently from other methods and produces more "natural" OTU clusters based on a graph representation of the sequences instead of simple similarity thresholds. This approach also helps to reduce the number of OTUs, which can be inflated in similarity-based methods due to sequencing errors [198, 229]. Although these tools employ different strategies and concepts to cluster sequences, the resulting OTUs are generally consistent across environments and yield comparable ecological results [318].

Once a sequence dataset has been clustered into OTUs, it can be summarized by the per-sample abundances of representative sequences for each OTU. These OTU representatives can be thought of as the cluster centroids resulting from the algorithm. The data are most commonly stored in a so-called OTU (contingency) table that lists the abundances (counts) of each OTU in each sample. Whether to use these abundances for downstream analyses, or instead to use the presence/absence of each OTU in a sample as an indicator of the sample diversity, depends on the research question at hand. Furthermore, whether to use OTUs at all instead of all sequences is a current debate [44, 126]. We further discuss these questions later in the context of this work in Section 2.5.3. For broad-scale ecological studies however, which is what we are mostly interested here, the obtained results are mostly consistent across these variants [126].

2.3 The Tree of Life

The common evolutionary history of life gives rise to a branching pattern, where new *lineages* split from a common ancestor. This branching pattern forms a tree-like structure, which classifies organisms in a hierarchy based on common descent.

While this *tree of life* is an expedient and, hence, pervasive model [155, 258], it ignores certain biological and evolutionary events. A strict hierarchy does not allow for reticulate events, such as hybridization [226], genetic recombination [149], or horizontal gene transfer [90, 275, 305]. Although approaches such as networks have been proposed to model these events [46, 159], the simplicity of a hierarchy or a tree structure still remains useful and is widely employed for classifying and naming organisms, as well as understanding their evolutionary relationships.

2.3.1 Taxonomy and Nomenclature

Early attempts to classify organisms date back to the Greek philosopher Aristotle, who used observable attributes to categorize living things into groups [204]. This approach as well as the efforts of later centuries were non-uniform and inconsistent. The basis for the modern system of classification was established by the Swedish botanist Carl Linnaeus in the mid-18th century [86]. He proposed a *nomenclature*,

that is, a naming system for organisms, as well as a *taxonomy*, that is, a rank-based classification of organisms [212, 213].

A taxonomic group of organisms is called a *taxon* (plural: *taxa*). Each taxon is associated with a *taxonomic rank*, which can subsume other ranks, thus forming a hierarchy of higher and lower ranks. A taxonomic rank represents the relative level of a group of organisms in the taxonomy. The principal contemporary ranks are *domain*, *kingdom*, *phylum*, *class*, *order*, *family*, *genus*, and *species*, see Figure 2.4. If needed, further ranks can be included in between (such as *sub-genus*), or more refined lower levels can be added (such as *strain*, which serves as a further distinction within a species).

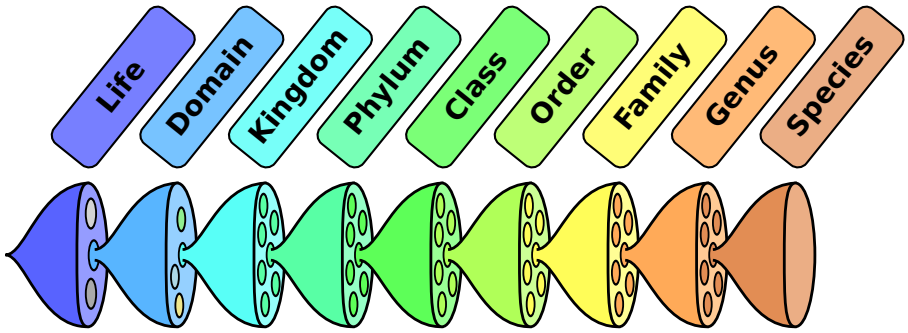

Figure 2.4: Biological classification into taxonomic ranks. The figure depicts a typical set of nested taxonomic ranks [386], which form a hierarchy with increasingly deeper levels towards the right. Source: Image derived from [141].

In order to scientifically name the groups of organisms (taxa) in a taxonomy, the *binomial nomenclature* as introduced by Linnaeus is still prevalent to date. It uses two terms, often of Latin origin, which respectively specify the taxonomic ranks *genus* and *species* that an organism belongs to, for example *Homo sapiens*.

While early classifications relied on *phenotypes*, that is, observable characteristics or traits of an organism, modern approaches to taxonomy take the genetic information into account [247]. For example, the three-domain system [385, 386] resolves the oldest evolutionary relationships, that is, the highest taxonomic levels, based on 16S rRNA data. Although this classification has been challenged [49, 139, 246], it is still widely used [155]. It divides cellular life forms into the three domains *Bacteria*, *Archaea*, and *Eukaryota*. The *Eukaryota* are further separated into kingdoms, which include the kingdoms *fungi*, *plants*, and *animals*.

2.3.2 Phylogenetic Trees

The classification of organisms into a taxonomy is based on (subjective) dissimilarity and thus arbitrary: The number of organisms that are grouped into a taxon at a given rank can vary, and the separation into discrete ranks does not reflect the gradual nature of evolution [124]. A more involved approach that can resolve these issues is *phylogenetics*, which is the study of the evolutionary history and relationships of biological entities (individuals, species, populations).

The evolutionary relationships of such entities are called their *phylogeny*. As the true phylogeny of a set of taxa is lost in time and thus unknowable [140], it has to be inferred from data that are available. A phylogenetic analysis uses inference methods that evaluate observed heritable traits in order to reconstruct the phylogeny under a given model of evolution for these traits. While phenotypes can be used, modern phylogenetic inference is mostly based on molecular data such as DNA. The result of a phylogenetic analysis is a *phylogenetic tree*, also called an *evolutionary tree*, or—synonymously—a *phylogeny*. Figure 2.5 shows two examples of phylogenetic trees.

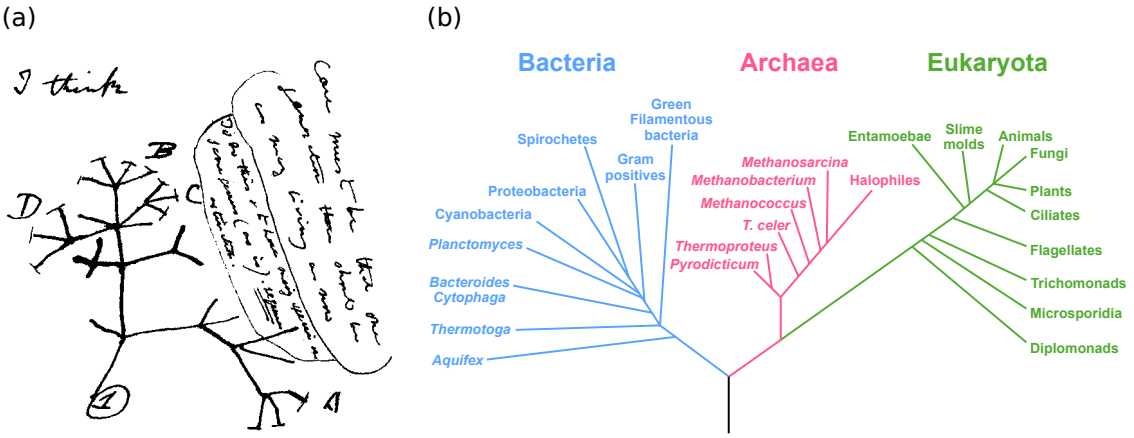

Figure 2.5: Exemplary phylogenetic trees. (a) In 1837, Charles Darwin sketched his first evolutionary tree below the words "I think" in his notebook on "Transmutation of Species". Source: Image derived from [72]. (b) A modern phylogenetic tree showing the three-domain system [385, 386], which emphasizes the separation of *Bacteria*, *Archaea*, and *Eukaryota* based on 16S rRNA genes. The black branch at the bottom represents the speculative last universal common ancestor of all living organisms. Source: Image derived from [119].

Properties of Phylogenetic Trees

The leaf nodes, or *tips*, of the tree represent living (*extant*) biological entities such as species, strains, individual organisms, or even single cells. Thus, the tips are often referred to as the *taxa* of the tree. This is a generic term that includes all of the above entities. The tips are often named according to the entities they represent (e. g., species names); such a tree is called a *labeled* tree. The inner nodes on the other hand are usually anonymous and represent hypothetical speciation events, where two novel lineages arose from a putative common ancestor. The branching pattern of a phylogenetic tree thus reveals the evolutionary history of its taxa.

The edges of the tree, also called *branches*, can have associated *branch lengths*, which represent the evolutionary time between the two adjacent nodes, for example, measured as the average change in nucleobases between their respective sequences.

The unique path between any two nodes can thus be interpreted as a measure of evolutionary relatedness of the taxa represented by the nodes.

A phylogenetic tree is *rooted* if it is a directed tree that has a unique *root node*, which corresponds to the putative common ancestor of the other nodes in the tree. See Figure 2.5(b) for an example. As evolution is a processes that occurs over time, from a biological point of view, every tree has a root. However, most models of DNA evolution are time-reversible, meaning that the direction of change in the sequences cannot be inferred from the data under such models, see Section 2.4.2 for details. Thus, tree inference methods commonly yield *unrooted* trees without direction and without a root node. In these methods, for computational reasons, often a *virtual root* is used, which is a hypothetical additional node placed on a branch of the tree. For tasks such as traversing a tree, but also in order to store a tree in a file, unrooted trees usually have a distinguished, but arbitrary, "starting" node called a *top-level trifurcation*. An unrooted tree can be rooted a posteriori, for example by using one or more *outgroup* sequences of taxa that are closely related to the group of taxa of interest (the *ingroup*), but not part of it. Then, a root node can be placed on the branch that separates the outgroup from the ingroup in the tree.

An inner node that has exactly three neighboring nodes is called a *bifurcation* or a *bifurcating* node. In rooted trees, these neighbors are the parent and the two children of the node, hence the name. An inner node with more neighbors is called a *multifurcation* or a *multifurcating* node. This naming also applies to the whole tree: A tree, where each inner node (with the exception of the root node in a rooted tree) is bifurcating, is also called a bifurcating tree. Otherwise (if there is at least one multifurcating node), it is a multifurcating tree. Note that in evolution, an actual multifurcation event is highly unlikely, as it corresponds to the simultaneous formation of more than two new lineages from a single ancestral lineage. Multifurcating trees are, for example, used when relationships cannot be properly resolved based on the available data, or to summarize a set of otherwise contradicting trees.

Each edge of the tree induces a *bipartition* or *split* of the taxa of the tree into two disjoined groups, one on each side of the edge. Splits of edges that are adjacent to tip nodes are *trivial*, as they appear in every possible tree topology for a given set of taxa. Therefore, only the *non-trivial* splits are generally of interest. The set of bipartitions induced by the edges of a tree uniquely defines the tree topology, and is hence often used in topology-related algorithms; for example, the tree in Figure 2.6(a) is described by the set of bipartitions $B = \{(\,\mathsf{ABC}|\mathsf{DE}\,), (\,\mathsf{AB}|\mathsf{CDE}\,)\}$.

For two trees T_1 and T_2 with the same set of taxa, but differing topologies, their bipartitions sets B_1 and B_2 can be used to define distance metrics between them. The *Robinson-Foulds* (RF) distance [304], or *symmetric difference* metric, for instance is defined as the number of bipartitions that are unique to either of the trees:

$$\begin{aligned} \mathrm{RF}(T_1, T_2) &= |\, B_1 \cup B_2 \,| - |\, B_1 \cap B_2 \,| \\ &= |\, (B_1 \setminus B_2) \cup (B_2 \setminus B_1) \,| \end{aligned} \tag{2.1}$$

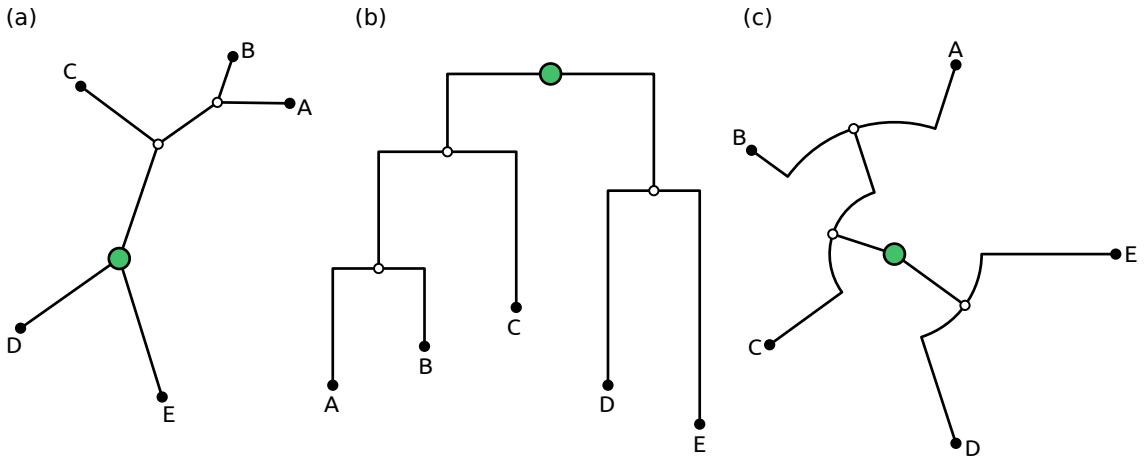

Figure 2.6: Types of phylogenetic trees. Here, we show three different types of labeled, bifurcating trees. Tip nodes are marked with black dots, inner nodes with white dots, and the top-level trifurcation or root node with a larger green dot. (a) An unrooted tree with five taxa. One node is arbitrarily selected as top-level trifurcation. (b) The same tree topology, but rooted on the inner branch that splits the taxa D and E from the other taxa. The tree is drawn in rectangular style, where vertical lines correspond to branch lengths. The horizontal lines are simply used to visualize the taxa, and have no biological interpretation. (c) The same tree again, but this time drawn in circular style. Here, radial lines correspond to branch lengths, while arcs only serve visualization purposes.

This unweighted, absolute distance is often used when comparing phylogenies. There also exists weighted and relative variants, as well as a variant called *branch score* [197], which also takes the branch lengths of the trees into account. There are also further measures to assess the incongruence between trees, such as as the *internode certainty* [310, 403].

A set of taxa is called *monophyletic*, if it includes the most recent common ancestor of the taxa in the set, and includes all descendants of that ancestor. In this work, we use an equivalent definition that also works for unrooted trees: A set of taxa is called *monophyletic*, if there is a bipartition (i. e., an edge) of the tree that separates these taxa from all other taxa of the tree. In a rooted tree, the node at the end of that edge is then the common ancestor of these taxa. A monophyletic set of taxa is also called a *clade* of the tree; in other words, a clade is a subtree that is separated from the rest of the tree by one edge. For example, the set of taxa containing A, B, and C, as well as the two small white inner nodes in Figure 2.6(a) is monophyletic—these taxa form a clade of the tree. Lastly, a non-monophyletic set of taxa is either called *paraphyletic* or *polyphyletic*, depending on whether the most recent common ancestor of the taxa is part of the set or not. For instance, the taxa A and C in Figure 2.6(a) are polyphyletic—they are not separated from the other taxa by a single edge, and do not contain their most recent common ancestor (which is the white node next to C).

Practical Aspects of Trees

While the topology of the tree describes evolutionary relationships, there are several ways of visualizing this information. Figure 2.6 shows some examples, which all depict the same topology (except for the rooting). The figure also summarizes some of the terms and concepts introduced above. The different drawing styles each have their advantages and disadvantages. For example, in a rectangular tree, as shown in Figure 2.6(b), branch lengths are easier to read and compare, while a circular tree, as shown in Figure 2.6(c), can fit more taxa in the same drawing area.

Taxonomy and phylogeny serve a related, but different purpose [153]: While the former is a system of classification, the latter describes the evolutionary history. However, there is a correspondence between a taxonomy and a rooted phylogeny: Inner nodes of the tree constitute older evolutionary relationships, which are represented by the higher ranks of the taxonomy. Figure 2.5(b) shows such a correspondence for the three domains of life. It is however possible that the taxa at one rank of the taxonomy are not monophyletic in the phylogenetic tree [153]. In this case, the two are *incongruent*.

The most common file format for storing phylogenetic trees is the `Newick` format [12]. It uses parentheses to specify the nesting structure of the tree, and also allows to store node labels and branch lengths. It however lacks proper support for additional branch- and node-related meta-data, which can lead to erroneous results [68]. The `NEXUS` format [225] is a container format for biological data, and internally also relies on the `Newick` format for storing trees. Furthermore, the `phyloXML` format [144] is an `XML` based format that allows to store arbitrary (meta-)data at the nodes and edges of the tree.

2.3.3 Tree Inference

A phylogeny can be inferred from data that has per-taxon traits which are homologous, that is, which have evolved from the same traits in the common ancestor and are thus comparable [111, 391]. While historically these traits were mostly phenotypes (bone shapes and sizes, metabolism, etc.), the focus has since shifted towards molecular data such as DNA and amino acid sequences [405], as their *phylogenetic signal* is generally more abundant [152]. Most often, a multiple sequence alignment is used, whose homologous sites represent the traits of the taxa. To determine the degree of relatedness between taxa, mathematical models of trait evolution are employed.

The general concept of tree inference is to then put closely related taxa close to each other in the phylogeny. Hence, a tree inference can be thought of as an optimization problem, which searches for the best tree given an optimality criterion. However, the space of all possible tree topologies is too large for an exhaustive brute-force search, even for small datasets with few taxa. For a given number of taxa n, the number of distinct tree topologies N is given as $N(n) = \prod_{i=3}^{n} (2i - 5)$, which grows over-exponentially fast [111]. There are thus different approaches and heuristics to conduct tree searches.

Distance based methods such as *Unweighted Pair Group Method with Arithmetic Mean* (UPGMA) [333] and *Neighbor Joining* [309] use a pairwise distance matrix between sequences, and thus do not necessarily need an alignment. They can be considered fast heuristics that yield a tree that is optimal under their respective criteria. On the other hand, there are conceptually different approaches that compute a *score* for a given tree, and search the tree space for the tree with the best score. *Maximum Parsimony* [313] uses a tree scoring scheme that is based on Occam's razor, that is, it tries to find the tree that explains the observed tip sequences (taxa) with the minimal number of substitutions (mutations). The *Maximum Likelihood* (ML) method [109] employs statistical techniques in order to evaluate the probability that a particular phylogenetic tree generated the given alignment, and successively searches the tree space for the most likely tree, see Section 2.4. Furthermore, *Bayesian Inference* also relies on the evaluation of tree probability [391], and uses Bayes' theorem to calculate the posterior distribution of the relevant evolutionary processes by integrating over the tree and the parameter space; it can thereby also incorporate prior empirical knowledge into the process.

Typical software tools for inferring ML trees include IQ-TREE [271], FastTree [293], GARLI [406], and RAxML [192, 342]; see Zhou et al. (2017) [404] for a critical review. Bayesian inference on the other hand can for example be conduced using software tools such as BEAST [348], MrBayes [307], and ExaBayes [3].

2.4 Maximum Likelihood Tree Inference

In the context of this work, we are mostly interested in Maximum Likelihood (ML) based tree inference. It uses a probabilistic framework in which the (phylogenetic) likelihood

$$\mathcal{L}(\ \text{MSA} \mid T, \bar{b}, M, \bar{\theta}\ ) \tag{2.2}$$

is evaluated that an observed MSA is the outcome of an evolutionary history described by a phylogenetic tree with topology T and branch lengths $\bar{b}$, under a given model of trait evolution M with parameters $\bar{\theta}$. For a fixed model M (see Section 2.4.2), the likelihood can be expressed as a function of T, $\bar{b}$ and $\bar{\theta}$, which is known as the *phylogenetic likelihood function* (PLF).

2.4.1 Tree Search

By maximizing the PLF using maximum likelihood estimation, the parameter values (including tree topology) are determined which best explain the observed data. This process hence uses the likelihood as a scoring function, and is called *tree search*. Typically, the parameter estimates are obtained in an iterative process, which alternates between two phases until a (potentially local) optimum is found:

1. Optimizing the tree topology T (as well as the few branch lengths that are most affected by the respective topological change), while keeping the other branch lengths $\bar{b}$ and the model parameters $\bar{\theta}$ fixed.

2. Optimizing all branch lengths $\bar{b}$ of the given tree topology, as well as the model parameters $\bar{\theta}$.

Finding the most likely tree topology is a discrete optimization problem, which has been shown to be NP-hard under the ML criterion [56]. Furthermore, the evaluation of the PLF is computationally expensive, as it involves many floating point operations, see Section 2.4.4. A general heuristic for the tree search that avoids an exhaustive evaluation of the tree space can, for instance, be designed as follows. First, a starting tree is generated, either randomly, or using methods such as Neighbor Joining or Maximum Parsimony. Then, the likelihood of the tree is successively improved by applying topological rearrangements (*moves*) to the tree. For instance, in *greedy hill-climbing* [342], only those moves are applied (*accepted*) that immediately improve the likelihood score.

For a fixed tree topology T, the maximum likelihood estimates of the branch lengths $\bar{b}$ and the model parameters $\bar{\theta}$ are usually obtained using general-purpose numerical optimization methods. Since the derivatives of the PLF can easily be computed, the Newton-Raphson method [396] is often used for optimizing the branch lengths, see Section 2.4.5. Other model parameters are commonly optimized using Brent's method [38] or the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [115].

2.4.2 Models of Molecular Sequence Evolution

So far, we assumed that a model M is given for describing the evolution of the traits that are used for inferring the tree. For sequence data, such a model yields an estimate of the evolutionary distance between the sequences of different taxa. Because the inference assumes homologous traits, the only mutations that are considered for aligned sequence data are substitutions.

Markov Chain Model

Most commonly, a continuous-time Markov chain (MC) is used to describe the evolution of a single site within a set of aligned sequences [120]. For DNA data, the MC has four states `A`, `C`, `G`, and `T`, which correspond to the nucleobases. Transitions between the states correspond to their substitutions, see Figure 2.7. While the MC model ignores aspects such as natural selection and the molecular mechanisms of evolution, it describes the relative rate of changes in a way that allows multiple substitutions to occur along the same branch (`T` $\rightarrow$ `A` $\rightarrow$ `G`).

The process of state transitions is defined by the substitution rate matrix

$$Q = \begin{pmatrix} -q_A & q_{AC} & q_{AG} & q_{AT} \\ q_{CA} & -q_C & q_{CG} & q_{CT} \\ q_{GA} & q_{GC} & -q_G & q_{GT} \\ q_{TA} & q_{TC} & q_{TG} & -q_T \end{pmatrix} \tag{2.3}$$

$$-q_i = -\sum_{j \neq i} q_{ij}, \; i, j \in \{ A, C, G, T \} \tag{2.4}$$

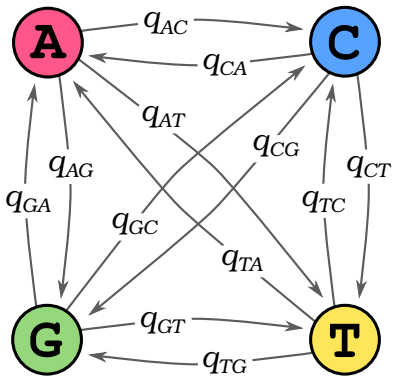

Figure 2.7: Markov chain model of nucleotide substitutions. The evolution of characters at a site in an alignment can be modeled as a Markov chain (MC). The states of the MC for DNA data are the four nucleobases `A`, `C`, `G`, and `T`. The model allows transitions with rates q_{ij} with $i, j \in \{ A, C, G, T \}, i \neq j$ between all states, which correspond to substitutions of the nucleobases.

where the elements q_{ij} are the *instantaneous transition rates* from state i to state j. The rows of the Q-matrix have the requirement to sum to 0, by which the diagonal elements q_i are defined.

The expected number of substitutions at an alignment site between two nodes of the tree is expressed as the branch length b between the nodes, and is a component for measuring the (real) evolutionary distance/time t between them. Under the MC model, evolutionary time and branch length are proportional to each other with the *evolutionary rate* r being their proportionality factor: $t = r \cdot b$. Then, for a given time t, the *transition probabilities* $p_{ij}(t)$ between states in a stationary process are obtained by exponentiating the Q-matrix [392]. These probabilities are specified by the matrix

$$P(t) = e^{Qt} \tag{2.5}$$

For positive transition rates $q_{ij} > 0, \forall i \neq j$, if the process runs long enough, the Markov chain eventually reaches the unique *stationary* distribution $\Pi = (\pi_A, \pi_C, \pi_G, \pi_T)$, with π_i being the proportion of time spent in state i. If the Markov process reached equilibrium after running long enough, it can be interpreted as having generated the sequences of the MSA. In that case, Π is the *equilibrium base composition* of the MSA, and π_i are the *equilibrium* or *stationary base frequencies* of the MSA.

Time-Reversible Models

As mentioned before, most models of DNA evolution assume a *time-reversible* Markov process, which means that $\pi_i q_{ij} = \pi_j q_{ji} \; \forall \, i \neq j$. This assumption is biologically not meaningful, as evolution is a process in time, and thus does have a direction. It however allows for simplified calculations: The Q-matrix of a time-reversible model

can be formulated as the product of a symmetric rate matrix $R = \{r_{i \leftrightarrow j}\}$ and a diagonal matrix of the stationary base frequencies:

$$Q = R \cdot \text{diag}(\pi_i) = \begin{pmatrix} -q_A & r_{A\leftrightarrow C} \cdot \pi_C & r_{A\leftrightarrow G} \cdot \pi_G & r_{A\leftrightarrow T} \cdot \pi_T \\ r_{A\leftrightarrow C} \cdot \pi_A & -q_C & r_{C\leftrightarrow G} \cdot \pi_G & r_{C\leftrightarrow T} \cdot \pi_T \\ r_{A\leftrightarrow G} \cdot \pi_A & r_{C\leftrightarrow G} \cdot \pi_C & -q_G & r_{G\leftrightarrow T} \cdot \pi_T \\ r_{A\leftrightarrow T} \cdot \pi_A & r_{C\leftrightarrow T} \cdot \pi_C & r_{G\leftrightarrow T} \cdot \pi_G & -q_T \end{pmatrix} \tag{2.6}$$

The matrix describes the most general model of DNA evolution, where all 6 substitution rates $r_{i \leftrightarrow j}, i \neq j$ and all 4 base frequencies π_i can be different. This model is called the Generalized Time-Reversible (GTR) model [352]. As the sum of the base frequencies must be 1, and as the substitution rates are usually normalized by requiring that $r_{G\leftrightarrow T} = 1.0$, the GTR model has a total of 8 free parameters (that is, 3 base frequencies, and 5 substitution rates). The base frequencies can also be estimated from the character frequencies in the given MSA, in which case they are called *empirical* base frequencies.

There are also simpler, more restrictive models, which have fewer free parameters, and are thus more robust if data for estimating them is sparse, at the expense of descriptiveness. The Jukes-Cantor model (JC69) [169] has no free parameters at all and assumes equal substitution rates $r_{i \leftrightarrow j} = 1, i \neq j$ as well as equal base frequencies $\pi_i = 1/4$. The K80 model [180] adds a free parameter κ, which describes the ratio between two types of substitutions that are not equally likely to occur at the same rate in evolution: $r_{A\leftrightarrow C} = r_{G\leftrightarrow T} = \kappa \cdot r_{A\leftrightarrow G} = \kappa \cdot r_{A\leftrightarrow T} = \kappa \cdot r_{C\leftrightarrow G} = \kappa \cdot r_{C\leftrightarrow T}$. The F81 model [109] instead extends the JC69 model by allowing different base frequencies: $\pi_A \neq \pi_C \neq \pi_G \neq \pi_T$. The HKY85 model [146] combines the K80 model and the F81 model, and hence has 4 free parameters. Further models have also been proposed, which offer compromises between the number of free parameters and the expressiveness of the model [392]. Including the named models above, there are a total of 203 possible time-reversible nucleotide models [154].

The state space of the Markov process becomes significantly larger for protein data, as it needs to comprise all 20 standard amino acids instead of just 4 nucleobases. Hence, the GTR model for protein data has $(400 - 20)/2 - 1 + 19 = 208$ free parameters. Typical amino acid alignment sizes do not contain enough data to reliably estimate these parameters, and thus are prone to over-fitting. Therefore, so-called *empirical* amino acid models are commonly used, which have substitution rates and equilibrium base frequencies that were pre-estimated on large collections of reference alignments. Among others, some popular models include the DAYHOFF [77], WAG [384], and LG [201] models of protein substitution.

2.4.3 Further Aspects of Tree Inference

Evolution is a complex process with intricate details. Many additional and more sophisticated models and methods have thus been proposed to improve tree inference, accuracy, and realism of the models [392]. In the following, we introduce the ones that are relevant in the context of this work.

Rate Heterogeneity

The models of sequence evolution described above make the simplifying assumption that the alignment sites evolve independently and are identically distributed. However, certain regions of DNA or amino acid sequences are under higher evolutionary pressure than others, for example if they describe important molecular functions that need to be conserved in their evolutionary history. It is thus expected that some alignment sites evolve faster than others. That is, the evolutionary rate r of sites is not constant across the alignment. In the context of phylogenetic inference, several models of *rate heterogeneity among sites* have been proposed to account for this, some of which are described in the following.

A simple model is the *proportion of invariable sites*, where the likelihood of an alignment site is influenced by a parameter $p \in [0, 1]$ that describes the proportion of sites that are assumed to be identical (*invariable*) across all taxa. The more elaborate Γ model [389] postulates a shape parameter $\alpha > 0$ which models the rate heterogeneity as a gamma distribution $\Gamma(\alpha)$. The distribution shape ranges from exponential-like ($\alpha < 1$, high rate heterogeneity) to normal-like ($\alpha > 10$, low rate heterogeneity). Thus, by optimizing the single free parameter α, different unimodal rate heterogeneity profiles can be approximated. The likelihood under this model is then computed by integrating over the Γ distribution, typically using a discretization into four intervals for computational reasons. The CAT or *per-site rates* model [341] is a compute- and memory-efficient approximation of the Γ model, which explicitly assigns one of K rate categories to each alignment site instead of using a distribution of rates. Lastly, the FreeRate model [390] allows for multimodal distributions by using K rate categories and respective weights, which can approximate any distribution at the cost of having to estimate these free parameters.

Alignment Partitioning

Apart from the evolutionary rate r, the substitution patterns among the sites of an MSA can also differ. In order to account for this, the MSA can be split into different *partitions*, where each such partition is assigned an individual model of evolution. For example, as three nucleobases code for one amino acid in regions that encode for proteins (see Section 2.1), three partitions can be used, each modeling the evolution of the first, second, and third nucleobase of each amino acid. Furthermore, when sequence data for multiple genes is available, large multi-gene MSAs can be constructed by horizontal concatenation of per-gene MSAs [359]. In these *supermatrix* approaches, the multi-gene MSA can use partitions corresponding to individual genes, which might be under different evolutionary pressures.

Constrained Trees

The tree search (see Section 2.4.1) can (theoretically) yield any topology from the vast space of possible trees. It might thus be helpful to conduct a *constrained* tree search, for example to include prior knowledge about the taxa, to maintain

congruence with a given taxonomy, or because some other constraints need to be complied with. Such a constraint can, for example, be specified by enforcing certain bipartitions to be retained in the tree, that is, disjoined splits of the taxa that must be separated from each other in the tree. As a bipartition is induced by a branch in the tree, this is equivalent to starting the tree search with a multifurcating tree, and then resolving these multifurcations without changing the other parts of the tree. A constrained search yields a *constrained tree*. Note however that constrained tree searches can induce substantial bias due to constraints that contradict the phylogenetic signal of the sequence data. They hence often yield worse likelihood scores than unconstrained searches; see for example Table 3.2.

2.4.4 Likelihood Computation

Here, we introduce the basic computational aspects of the Maximum Likelihood score calculation. For a more in-depth description, see Yang (2014) [392]. We assume a fixed tree topology T, fix branch lengths $\bar{b}$, as well as a given model of sequence evolution M with parameters $\bar{\theta}$. That is, we do not cover the tree search itself here, but describe only how to compute the likelihood $\mathcal{L}$ (Equation 2.2) for a given MSA under these conditions.

A central point of the ML method is to account for the unknown states at the inner nodes of the tree. That is, the total likelihood is obtained by summing over the probabilities of every possible state at the inner nodes, which can be efficiently computed via the *Felsenstein pruning algorithm* (FPA) [109]. It traverses the tree in post-order fashion, that is from the tips towards the (virtual) root, and recursively calculates a so-called *conditional likelihood vector* (CLV) at each inner node.

In a sense, a CLV summarizes the subtree below its corresponding node. For every alignment site and every state, it describes the *conditional likelihood* of the node to be in the given state at the given site, given the subtree topology and its branch lengths, for the respective subset of the alignment (tip sequences). We here assume a set N of states, that is, the sequences consist of characters $c \in N$, for example $N = \{ \texttt{A}, \texttt{C}, \texttt{G}, \texttt{T} \}$. Furthermore, for simplicity, we do not consider alignment partitioning or rate heterogeneity among sites here, and thus use a fixed evolutionary rate r (usually, $r := 1.0$). Then, a CLV contains $|N|$ elements per alignment site, each describing the conditional likelihood of being in the corresponding state. For DNA data, these are CL(A), CL(C), CL(G), and CL(T).

Felsenstein Pruning Algorithm

In order to start the recursion of the FPA, first, the CLVs at the tip nodes have to be initialized. Each tip CLV expresses the likelihoods of observing each of the characters $c \in N$. In principle, these can be the actual likelihoods of the characters at the corresponding alignment site of the input data. Hence, uncertainty in the sequencing process can be incorporated into the computations [193]. However, this uncertainty is often not available in empirical datasets. Thus, tip nodes are usually initialized with "pseudo-CLVs", where for instance a nucleobase A in the alignment yields CL(A) := 1, and CL(C) := CL(G) := CL(T) := 0 at a particular site.

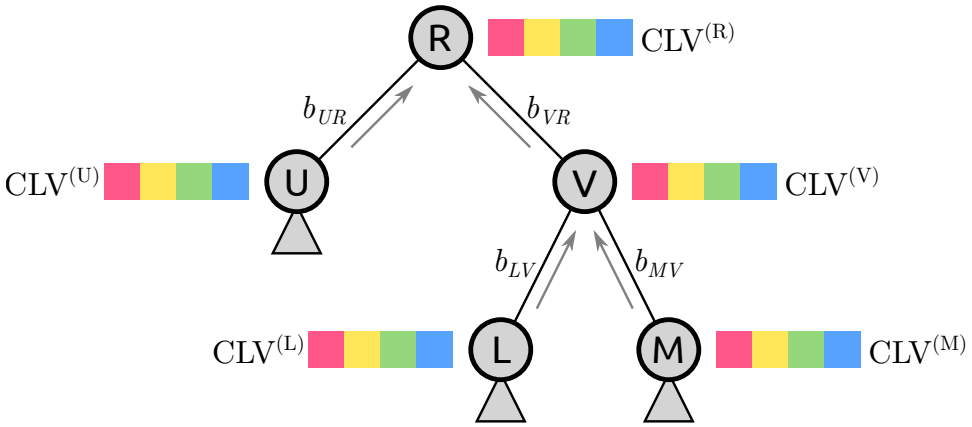

Figure 2.8: Felsenstein pruning algorithm. An exemplary tree topology with a (virtual) root R, an inner node V, and three other nodes U, L, and M, and branch lengths between nodes. Potential subtrees are marked by triangles below nodes. Each node has a CLV assigned to it, which "summarizes" the subtree below it. A CLV stores a conditional likelihood for every alignment site and for every state. Here, for simplicity, we show the CLVs for one site and for four states, which, for instance, represent the likelihood of that site to be in either state of the four nucleobases.

After the tip CLVs have been initialized, the algorithm moves up the tree, see Figure 2.8. This can be thought of as moving along the branches towards a parent node, which induces the possibility of state transitions. This step is thus where the model M is employed (see Section 2.4.2). As we assumed a fixed evolutionary rate r, we can infer the time t between two nodes from the branch length b between them: $t = r \cdot b$. Then, for a given branch, we can compute the probability $p_{ij}(t)$ of a transition from state i to state j after moving along the branch, see Equation 2.5. Note that the probability p_{ij} depends on the branch length, meaning that for every branch (and every update in its branch length during optimization, see Section 2.4.5), a separate P-matrix has to be computed from the Q-matrix of the model, as explained in Section 2.4.2.

At a parent node whose children have been computed, we can now apply the recursion step of the algorithm. For instance, in the topology shown in Figure 2.8, the CLV of an inner node V can be computed given its children L and M. For the computation, the CLVs of the two child nodes, as well as the transition probabilities p_{ij} for the branch lengths b_{LV} and b_{MV} of the branches towards the parent are needed. Then, a single entry of the CLV, that is, the conditional likelihood of node V to be in state c at site s, is

$$\mathrm{CLV}_{s,c}^{(V)} = \left(\sum_{j \in N} p_{cj}(r \cdot b_{LV}) \cdot \mathrm{CLV}_{s,j}^{(L)} \right) \left(\sum_{k \in N} p_{ck}(r \cdot b_{MV}) \cdot \mathrm{CLV}_{s,k}^{(M)} \right) \tag{2.7}$$

The equation can be interpreted as follows: The product of transition probability and conditional likelihood represents a change from state c to another state $j \in N$.

By summing this product for all states, all possible inner states in the child node are accounted for. Finally, the product of the two sums is the new conditional likelihood of the node being in state c at site s, given the evolutionary history of its two children and their subtrees.

By repeating the computation for every state $c \in N$ and every site s of the alignment, the complete CLV for node V is computed. The recursion is then applied to all nodes upwards the tree, until all CLVs have been computed.

Likelihood Evaluation at the Root

Once all CLVs are computed, the overall likelihood $\mathcal{L}$ can be computed using the CLV of the root node. Given the root node R as shown in Figure 2.8, the overall *per-site likelihood* $\mathcal{L}_s$ of an alignment site s is accumulated from the conditional likelihoods of all states, taking their respective base frequencies π_i into account:

$$\mathcal{L}_s = \sum_{i \in N} \pi_i \cdot \mathrm{CLV}_{s,i}^{(R)} \tag{2.8}$$

Due to the time reversibility of the model, for unrooted trees, a *virtual* root can be used, that is, an additional node that is placed on an arbitrary branch of the tree. If, for example, the node R in Figure 2.8 is the virtual root, the two branches between nodes U and V are actually one branch with branch length $b_{UV} = b_{UR} + b_{VR}$. Then, an alternative way of computing the per-site likelihood is what we call the (per-site) *edge likelihood*. Instead of using the CLV at the virtual root R, we can use the CLVs of U and V, and the corresponding branch length b_{UV} for the computation. In that case, state transitions along the branch have to be additionally accounted for in the computation. The per-site edge likelihood of an alignment site s can then be computed as

$$\mathcal{L}_s = \sum_{i \in N} \sum_{j \in N} \pi_i \cdot \mathrm{CLV}_{s,i}^{(U)} \cdot p_{ij}(r \cdot b_{UV}) \cdot \mathrm{CLV}_{s,j}^{(V)} \tag{2.9}$$

This way of calculating the edge likelihood $\mathcal{L}_s$ works for any two adjacent nodes, if their respective CLVs represent the two subtrees attached to the nodes at either end of the branch.

Finally, the overall likelihood $\mathcal{L}(\mathrm{MSA} \mid T, \bar{b}, M, \bar{\theta})$ for the entire MSA can be computed. For mathematical simplicity, the sites are generally assumed to evolve independently, although this is not expected to be the case from a biological perspective. Under this assumption, for an alignment with m sites, the overall likelihood is simply the product of the per-site likelihoods:

$$\mathcal{L} = \prod_{s=1}^{m} \mathcal{L}_s \tag{2.10}$$

For computational reasons, and to avoid numerical underflow, in practice, the logarithm of the likelihood (*log-likelihood*) is typically computed:

$$\mathcal{L}^* = \sum_{s=1}^{m} \log \mathcal{L}_s \tag{2.11}$$

As the tree search is an optimization towards the highest likelihood score, using the log-likelihood is equivalent to using the likelihood in Equation 2.10. Furthermore, as identical sites (that is, identical columns of the MSA) yield exactly the same per-site likelihood, such sites are often compressed. That is, the respective likelihood is only computed once, and accordingly weighted in the overall likelihood computation.

2.4.5 Branch Length Optimization

Another important aspect of the tree search is the optimization of the branch lengths of the tree. That is, for a given tree topology T, and a fixed model M with parameters $\bar{\theta}$, we want to compute the branch lengths $\bar{b}$ that maximize the likelihood $\mathcal{L}(\text{MSA} \mid T, \bar{b}, M, \bar{\theta})$. This procedure is called *branch length optimization* (BLO), and typically uses the Newton-Raphson method [396], as mentioned in Section 2.4.1.

We consider the optimization of a single branch length b. In order to maximize $\mathcal{L}$, we need to find the root of the first derivative $\mathcal{L}'$. The Newton-Raphson method takes an initial value for b and then iteratively approximates values that take it closer to the root:

$$b_{n+1} = b_n - \frac{\mathcal{L}'}{\mathcal{L}''} \tag{2.12}$$

Note that the derivatives $\mathcal{L}'$ and $\mathcal{L}''$ can be obtained analytically [392], and have to be re-computed in every iteration. The algorithm stops when the change in b between two iterations is below a given threshold, that is, when then optimization *converges*. This procedure is repeated for all branch lengths $\bar{b}$ in the tree.

2.5 Phylogenetic Placement

In studies of sequence data, one of the most common tasks is a phylogenetic analysis of the data, that is, to infer the evolutionary context of the sequences. However, since the amount of sequence data produced in typical metagenomic studies can be enormous, computational challenges and bottlenecks arise [320]. In particular, both calculating an MSA and inferring a phylogeny are NP-hard [56, 170], and thus impractical or infeasible for large datasets. Furthermore, metagenomic reads are often short, and hence lack phylogenetic signal to robustly infer a tree and to properly resolve their relationships [166, 241].

Thus, *phylogenetic placement* (also called *evolutionary placement*) has been developed for conducting phylogenetic analyses of such data [262, 372]. It is implemented

in tools such as PPLACER [241], RAxML-EPA [25], and EPA-NG [18]. Instead of resolving the phylogeny of a set of metagenomic sequences, phylogenetic placement treats each sequence, called a *Query Sequence* (QS), separately. It evaluates how these QSs relate to an existing *Reference Tree* (RT) based on known reference sequences. For each QS, it computes the probabilities of *placing* the sequence on all branches of the RT, thereby classifying them into a phylogenetic context of related sequences, without the need to resolve relationships between the QSs.

2.5.1 Pipeline and Computation

In the most common use case, the QSs are reads or amplicons from environmental samples. Most often barcoding regions or marker genes such as 16S or 18S are used (see Section 2.2.2), but there also exist studies that use different, or even a set of, marker genes [349]. Furthermore, other types of sequences such as $_{\text{mi}}$tags [219] can be used.

The RT and the reference sequences it represents are typically assembled by the user so that they capture the expected species diversity in the samples. To expedite this process, we proposed an automated approach for assembling suitable sets of reference sequences [69], which we describe in Chapter 3. Distinct samples from one study are typically placed on the same underlying RT in order to facilitate comparisons between the samples, see Section 2.5.5.

We here assume to be given a set of suitable reference sequences, their alignment, and an RT inferred from them. As phylogenetic placement uses the maximum likelihood criterion, the RT has to be strictly bifurcating. Prior to the placement, the QSs need to be aligned against the reference alignment of the RT by programs such as PaPaRa [23, 24] or HMMALIGN [93, 94], see also Section 2.2.3. The input to phylogenetic placement are (i) the Reference Tree (RT), (ii) its underlying alignment, and (iii) the aligned Query Sequences (QSs). The output are the probabilities of placing the QSs on the branches of the RT. The placement pipeline is shown in Figure 2.9.

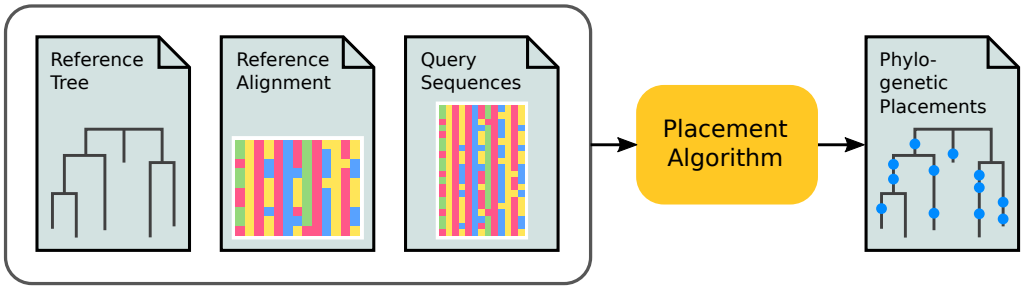

Figure 2.9: Phylogenetic placement pipeline. The input to phylogenetic placement are three files: the reference tree (RT), the corresponding reference alignment, and the aligned query sequences (QSs). The placement algorithm then computes the probabilities of placing the QSs on the branches of the RT, which are stored in an output file.

Computation for one Query Sequence

The placement is conducted for each QS separately, always using the same fixed RT as a starting point. Each branch of the tree is evaluated as a potential *placement location* of the QS, which indicates how likely the branch is to be the ancestor of the QS. In Figure 2.10, the procedure for one QS and one branch (between nodes D and P) is shown: The sequence is inserted as a new tip node Q into the branch, connected to it by a new *pendant* branch and a new node C. This splits the original branch into two parts, called the *distal* and the *proximal* branch, respectively, which are named according to the direction of the root of the tree. Note that the tree can also be unrooted, in which case the top-level trifurcation is typically used as root.

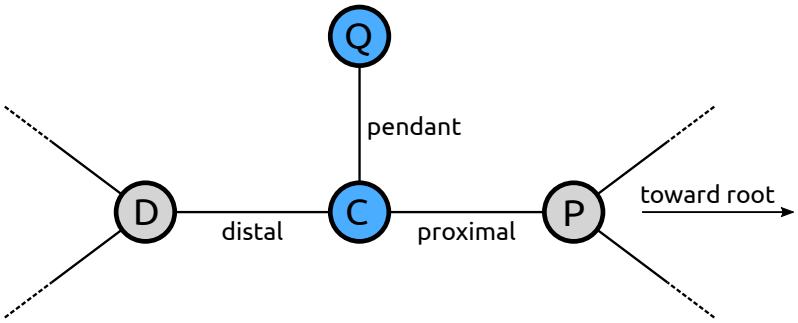

Figure 2.10: Terminology of a phylogenetic placement. The nodes D and P belong to the reference tree (RT). When placing a query sequence (QS), the branch between them is split into two parts by a new node C, which serves as the attachment point for another new node Q that represents the QS. The *pendant* branch leads to Q. The original branch is split into the *proximal* branch, which leads towards the root of the RT, and the *distal* branch, which leads away from the root.

In the next step, the branch lengths of the tree are optimized (or at least the most relevant ones, see below), using the method as explained in Section 2.4.5. After the optimization, the sum of the lengths of the distal and proximal branches is not necessarily equal to the original branch length between D and P. Thus, typically, the two lengths are proportionally rescaled to maintain this equality.

Lastly, the likelihood of the tree with the newly attached sequence is evaluated as explained in Section 2.4.4. Note that the likelihood computation uses the MSA (extended by the query sequence), the tree topology and branch lengths, as well as the model and its parameters as before. The model of nucleotide evolution should be the same that was used for inferring the tree, in order to keep the meaning of the branch lengths consistent, see Section 2.4.2. After this, the newly created nodes on the branch are removed again and the branch lengths are reset, thus restoring the original reference tree.

The above procedure is repeated for every branch i of the tree T, yielding a set of likelihood scores $\mathcal{L}(i)$ for each possible placement location of the QS. In other words, for each branch of the tree, the process yields a so-called *placement* of the QS, that is, an optimized position on the branch, along with a likelihood score for the whole

tree. The likelihood scores for a QS are then transformed into probabilities, which quantify the uncertainty of placing the sequence on the respective branch [347, 372]. The probability of placing a QS on a branch q is called the *Likelihood Weight Ratio* (LWR) and is calculated relative to the other branches i of the tree T as

$$\mathrm{LWR}(q) = \frac{\mathcal{L}(q)}{\sum_{i \in T} \mathcal{L}(i)} \tag{2.13}$$

By construction, the sum of the LWRs of all branches for a single QS is 1.0. It can thus be interpreted as a probability distribution over the branches of the tree, as shown in Figure 2.11. For most use cases, the LWRs and the respective placement on the branches are the most important values, while pendant lengths are rarely used in downstream analyses. However, long pendant lengths can indicate that the RT does not comprise reference sequences that are closely related to the QS.

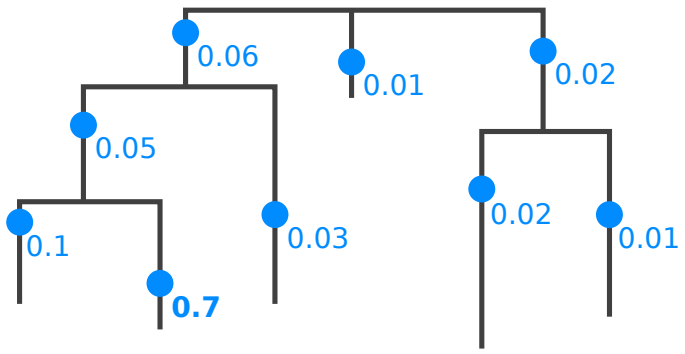

Figure 2.11: Phylogenetic placement of a query sequence. Each branch of the reference tree is tested as a potential insertion position, called a *placement* (blue dots; pendant lengths are ignored here). Note that placements have a specific position on their branch, due to the branch length optimization process. A probability of how likely it is that the sequence belongs to a specific branch is computed (numbers next to dots), which is called the *likelihood weight ratio* (LWR). The bold number (0.7) denotes the most probable placement of the sequence in this example.

Optimizations

The most expensive part of the placement computation are the branch lengths optimizations. Thus, several techniques have been developed to accelerate the placement process [18].

Firstly, above, all branches of the tree were optimized when evaluation a placement location, which gives the most accurate results. In practice however, it suffices to only optimize the three branches of the placement location without losing too much accuracy.

Secondly, because the reference tree is fixed (except for the temporarily created nodes during the computation), the CLVs of all possible subtrees can be precomputed,

which substantially accelerates the likelihood evaluation. Using Figure 2.10 as an example, with two CLVs of the nodes D and P, and one application of the Felsenstein Pruning Algorithm (see Section 2.4.4), the CLV of node C can be computed. Then, the final likelihood can be evaluated as the edge likelihood of the pendant edge, using this CLV as well as the pseudo-CLV of node Q.

Thirdly, further acceleration can be achieved with a so-called *pre-placement* heuristic: A first approximate evaluation of a placement location can be conducted without branch length optimization by using distal and proximal lengths that each are half of the respective original branch length of the tree, and a fixed default pendant length. Then, only the most likely fraction of locations is thoroughly evaluated, that is, including branch length optimizations. This way, millions or even billions of sequences can be placed within acceptable time [18]. This is however a heuristic that might lead to suboptimal results by ignoring placement locations whose LWR is significantly improved by the branch length optimization process. If the RT is however well suited for the QSs, this situation is generally not expected to occur, because the phylogenetic signal is strong enough to properly place the QSs even without optimized branch lengths.

Placement Result

The placement process is repeated independently for every QS. That is, for each QS, the algorithm starts calculating placements from scratch on the original RT. The result thus classifies each QS in the phylogenetic context of the RT, without resolving the evolutionary relationships between the QSs.

The output data is usually stored in so-called `jplace` files [243], which is a standard based on the `JSON` format [35, 65]. It stores the RT in `Newick` format, including tip names and branch lengths, and is extended by a post-order numbering of the edges to be referenced by the placements. The main part of the file is a list of lists: For each QS, its list of placements is stored. A placement is described by its edge number (referencing the RT), the LWR, the pendant length, and the distal length. The proximal length is usually omitted, as it can be inferred from the branch length of the tree. Furthermore, the format can summarize multiple identical QSs by allowing several names for each list of placements, where each name can also have a weight (called its *multiplicity*). Lastly, usually not all placement locations are stored in the file, as the ones with low LWRs are mostly irrelevant for post-analysis methods.

2.5.2 Use Cases and Applications

Phylogenetic placement is a flexible tool that yields useful biological information *per se*, but it can also be utilized for a variety of downstream analyses.

Comparison to Existing Methods

A typical task in metagenomic studies is to identify and classify the environmental sequences with respect to known reference sequences, either in a taxonomic or phylogenetic context, as mentioned in Section 2.2.2. Conventional methods for this task,

such as BLAST [8], are based on sequence similarity or identity. Such methods are fast, and directly assign query sequences to the most similar reference sequences, as shown in Figure 2.12(a). However, this implies that they only attain satisfying accuracy levels if the query sequences are sufficiently similar to the reference sequences. Furthermore, BLAST might yield suboptimal results [324], and the best BLAST hit does often *not* represent the most closely related species [185]. This is particularly true for environments where available reference databases do not exhibit sufficient taxon coverage [230]. Moreover, as insufficient taxon coverage cannot be detected by methods that are based on sequence similarity, they can potentially bias downstream analyses.

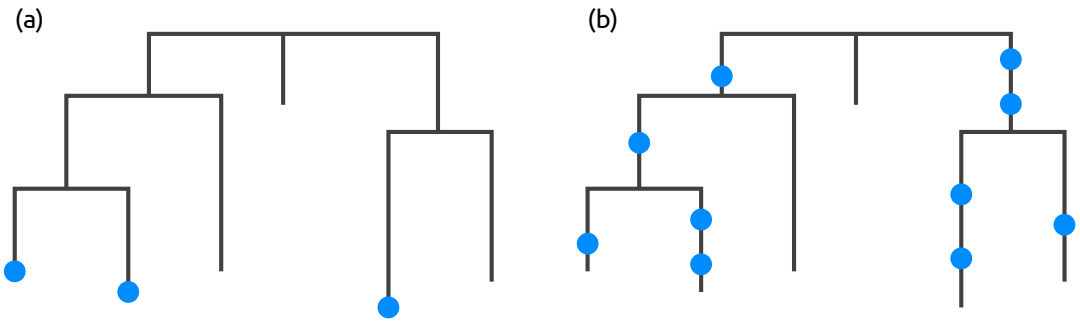

Figure 2.12: Comparison of similarity-based methods to phylogenetic placement. (a) Methods based on sequence similarity yield an assignment of query sequences to known reference sequences. This is equivalent to assigning the query sequences to only the tips of a reference tree (blue dots). (b) Phylogenetic placement provides more detail and assigns query sequences also to inner branches of the reference tree.

More recent methods can alleviate some of these issues, for instance by using machine learning techniques to obtain a taxonomic classification of metagenomic sequences [363], or by utilizing a phylogeny inferred from metagenomic sequence clusters to classify microbial communities [351]. However, the common shortcoming of these methods is that they lack a way of incorporating phylogenetic information of known sequences.

A phylogenetic placement analysis *does* incorporate known phylogenetic relationships, and hence provides a more accurate means for read identification and classification, as shown in Figure 2.12(b). For example, the classification of query sequences can be summarized by means of sequence abundances [156, 278], or to obtain taxonomic assignments [191, 272].

Furthermore, phylogenetic placement also allows for more elaborate downstream analyses. Firstly, the reference tree usually offers a higher resolution than simple per-taxon abundance counts, and the amount of mapped QSs per branch can be directly visualized on the RT [230], as shown in Section 4.1. Secondly, established methods such as Edge PCA and Squash Clustering [239], which we introduce in Section 2.5.5, allow for identifying subtle differences between distinct samples, thus enabling comparative studies directly based on phylogenetic placement. Lastly, we

proposed novel methods for visualizing and clustering phylogenetic placement data [67], which we describe in Chapter 4 and Chapter 5. Another typical task in metagenomic studies is to unravel diversity within the samples, which can be conducted using phylogenetic placement, as outlined in Chapter 8.

Variants and Derived Tools

The placement algorithm presented above relies on the standard ML framework for evaluating placement locations on the tree. There also exist variants of phylogenetic placement that use maximum parsimony [25] and minimum evolution [113] instead of maximum likelihood, and variants that calculate Bayesian posterior probabilities [241]. Moreover, the boosting method SEPP has been proposed to improve the accuracy of the placements [260]. The recently proposed tools RAPPAS [210] and APPLES [16] are alignment-free approaches that are based on comparing sequence k-mers (subsequences) instead of using an ML framework. These tools yield comparable results to standard ML-based phylogenetic placement implementations, but are faster and can handle larger reference trees.

Phylogenetic placement has further been used for a variety of applications and derived pipelines, such as species delimitation as in PTP [400] and mPTP [173], genome and metagenome analysis as in PhyloSift [71], taxonomic identification and phylogenetic profiling as in TIPP [272], and identification and correction of taxonomically mislabeled sequences as in Sativa [191].

2.5.3 Placement Processing

Phylogenetic placement is a useful tool to investigate the microbial composition and diversity of a set of distinct environmental samples. When placing multiple samples, for instance, from different geographical locations, or different human patients, typically, the same RT is used, in order to facilitate the comparison of the phylogenetic composition of these samples. There are furthermore several other considerations to take into account before running downstream analysis methods, some of which we discuss in the following.

Normalization

Firstly, it is important to consider how to properly normalize the samples. Normalization is required as the sample size (often also called library size), that is, the number of sequences per sample, can vary by several orders of magnitude within one study. This is due to technical aspects of the sequencing process, such as efficiency variations, or biases introduced by the amplification process, as explained in Section 2.2.2. As a consequence, metagenomic sequence data are inherently compositional [129, 208, 295], which can lead to spurious statistical analyses [6, 127, 163, 360]. This impedes statistical analyses of the data and hence needs to be considered in all analysis steps [208, 295, 330].

We here briefly outline common types of problems due to normalization that also affect phylogenetic placement data. Selecting an appropriate normalization strategy

constitutes a general problem in many metagenomic studies. The appropriateness depends on data characteristics [382], but also on the biological question asked. For example, estimating indices such as the species richness are often implemented via so-called *rarefaction* and rarefaction curves [132] by randomly re-drawing sequences from the set of sequences in a sample to obtain comparable sample sizes. This ignores however a potentially large amount of the available valid data [249]. Furthermore, the specific type of input sequence data has to be taken into account for normalization: Biases induced by the amplification process can potentially be avoided if, instead of amplicons, data based on shotgun sequencing are used, such as $_{\text{mi}}$tags [219]. Moreover, similar sequences can be clustered prior to phylogenetic placement analysis, for instance, by constructing Operational Taxonomic Units (OTUs), as introduced in Section 2.2.5. OTU clustering substantially reduces the number of sequences, and hence greatly decreases the computational cost of placement analyses. Lastly, one may completely disregard the abundances (which correspond to the *multiplicities* of placements) of the placed sequences, reads, or OTUs, and only be interested in their presence/absence when comparing samples.

All of the above analysis strategies are also applicable to phylogenetic placement, e. g., by placing OTUs instead of sequences. Which of these strategies is deployed, depends on the specific design of the study and the research question at hand. The common challenge is that the number of sequences per sample differs, which affects most post-analysis methods, and can lead to conflicting interpretations and irreproducible results [128, 360].

In the following, we therefore explain how the necessary sample size normalizations can be performed. We also introduce established terminology, and describe general techniques for interpreting and working with phylogenetic placement data. These are not methods on their own, but rather tools and building blocks that are necessary for the analysis methods explained and introduced later in this thesis.

Edge Masses

Methods that compare samples directly based on their sequences, such as the UniFrac distance [221, 223] (see Section 2.5.4), can benefit from rarefaction [382]. However, in the context of phylogenetic placement, rarefaction is not necessary. Thus, more valid data can be kept. For a thorough comparison of UniFrac to methods based on phylogenetic placement, see Matsen and Evans (2011) [239].

In order to compare the placements obtained from a set of samples, it is convenient to think of the reference tree as a graph. Then, the per-branch LWRs for a single QS can be interpreted as mass points distributed over the edges of the RT, including their respective placement positions on the branches, as shown in Figure 2.11. We call this the *mass interpretation* of the placed QSs, and henceforth use mass and LWR interchangeably. For simplicity, we ignore that typically not all placements are stored in `jplace` files, meaning that the mass per QS can also be < 1. Hence, each QS is assumed to have a total accumulated mass of 1.0 on the RT. The *mass of an edge* or *edge mass* refers to the sum of the LWRs on that edge for all QSs of

a sample, as shown in Figure 2.13(a). The edge masses can serve as a simplification of the data that summarizes a whole sample in a vector with m entries, for an RT with m edges. A larger example for the full RT is shown in Figure 2.14(a). The *total mass* of a sample is then the sum over all edge masses, which is identical to the number of QSs in the sample.

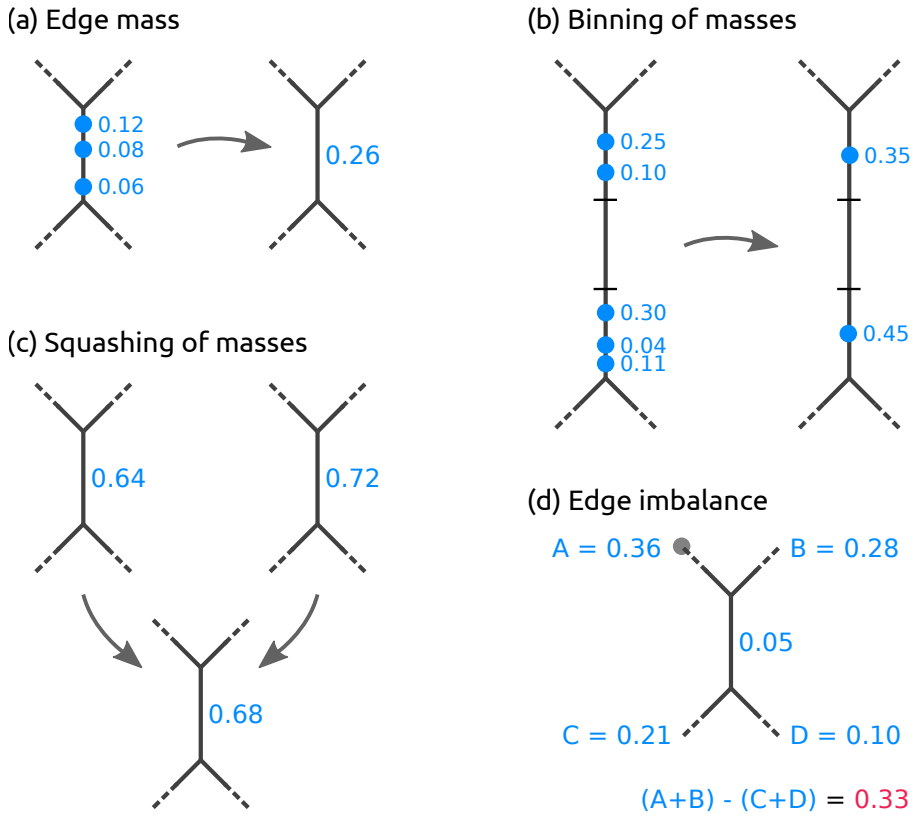

Figure 2.13: Operations on placement masses. (a) The *edge mass* or *mass of an edge* is the sum of LWRs on that edge for all QSs in a sample. (b) In order to reduce time and memory of the computations, masses can be *binned* by summarizing them across QSs in intervals along the edges. (c) The masses on corresponding edges of the RT of two or more samples can be *squashed* to represent the average mass distribution of the samples. For simplicity, we here use equal weights, and show edge masses instead of individual LWRs. (d) The *edge imbalance* of an edge is the sum of masses on all edges on the root side of the edge (A+B, with the root in subtree A denoted as a gray dot) minus the sum of the masses on the edges on the non-root side (C+D), while ignoring the mass on the edge itself.

The key idea is to use the distribution of placement mass points over the edges of the RT to characterize a sample. This allows for normalizing samples of different size by scaling the total sample mass to unit mass 1.0. This is done by dividing the mass of each placement location by the total sum of all masses in the sample. In other words, absolute abundances—which are not meaningful for analyses of metagenomic sequences due to the compositional nature of the data [129]—are converted into relative abundances. This way, rare species, which might have been removed by

rarefaction, can be kept, as they only contribute a negligible mass to the branches into which they have been placed. This approach is analogous to using proportional values for methods based on OTU abundance tables (see Section 2.2.5). For each OTU, such tables store a count of how often it appeared in each sample, which can be transformed into proportional values by scaling each sample/column of the table by its sum of OTU counts [382]. Most of the methods presented here use normalized samples, that is, they use relative abundances. As relative abundances are however compositional data, certain caveats occur [6, 128, 220], which we discuss where appropriate.

Binning and Squashing of Edge Masses

When working with large numbers of QSs, the mass interpretation allows to further simplify and reduce the amounts of data: The masses on each edge of the tree can be quantized into b discrete bins, as shown in Figure 2.13(b). That is, each edge is divided into b intervals (or bins) of the corresponding branch length, and all mass points on that edge are accumulated into their respective nearest bin. For example, by accumulating mass points at their nearest interval midpoint, or at the weighted average of all masses in each bin, masses are only minimally moved. The parameter b controls the resolution and accuracy of this approximation. In the extreme case of $b := 1$, all masses on an edge are grouped into one single bin (which is equivalent to only considering the edge mass instead of individual LWRs). This *branch binning* process drastically reduces the number of mass points that need to be stored and analyzed in several methods we present, while only inducing a negligible decrease in accuracy. As shown later in Table 5.1, branch binning can yield a speedup of up to 75% for post-analysis run-times without altering the results of the analysis.

The interpretation of placements as masses on the edges of the tree further allows to summarize a set of samples by annotating the RT with their (weighted) average per-edge mass distribution, as shown in Figure 2.13(c). This procedure is called *squashing* [239].

In order to squash two samples, they have to be placed on a fixed RT. Then, the mass distribution m_s of the squash tree is calculated as the weighted average of the two mass distributions m_l and m_r of the two input trees, using some weights w_l and w_r:

$$m_s \;=\; \frac{w_l}{w_l + w_r} \cdot m_l \;+\; \frac{w_r}{w_l + w_r} \cdot m_r \tag{2.14}$$

When using squashing to simply represent the average placement distribution of a set of samples, the weights can be set in accordance with the used normalization strategy (as explained above): When working with absolute abundances, the weights can be set to the total number of QSs per sample; for relative abundances, the weights are set to 1. The above calculation can also be trivially extended to more than two samples to get their average placement mass. Squashing is also a central technique of Squash Clustering [239], which we introduce below in Section 2.5.5, and which

also shows a larger example of squashing in Figure 2.16. There, the weights represent the influence of sample clusters during an agglomerative hierarchical clustering algorithm.

In practice, instead of only considering per-edge masses, the average mass distributions are calculated per placement location; that is, each LWR is taken into account individually. Because phylogenetic placements represent point masses on the edges of the tree, squashing can be seen as joining the (weighted) set of placements of the samples on corresponding edges of the RT. The resulting masses can then be normalized again to obtain unit mass for the resulting average tree.

Edge Imbalances

So far, we have only considered the per-edge masses. Often, however, it is also of interest to "summarize" the mass of an entire clade, that is, to consider per-clade masses. For example, sequences of the RT that represent species or strains might not provide sufficient phylogenetic signal for properly resolving the phylogenetic placement of short sequences [91]. In these cases, the placement mass of a sequence can be spread across different edges representing the same genus or species, thus blurring analyses based on per-edge masses.

Instead, a clade-based summary can yield clearer analysis results. It can be computed by using the tree structure to appropriately transform the edge masses. Each edge splits the tree into two parts (bipartitions, see Section 2.3.2), of which only one contains the root (or top-level trifurcation) of the tree. For a given edge, its mass difference is then calculated by summing all masses in the root part of the tree and subtracting all masses in the other part, while ignoring the mass of the edge itself [239], as shown in Figure 2.13(d). This difference is called the *imbalance* of the edge [239]. It is usually normalized to represent unit total mass, as the absolute (not normalized) imbalance otherwise propagates the effects of differing sample sizes all across the tree. It is irrelevant where the root of the tree is, as any re-rooting changes the sign of edge imbalance values consistently across different samples. A larger example for the full RT is shown in Figure 2.14(a).

The edge imbalance relates the masses on the two sides of an edge to each other. They directly compare the two sides of an edge and thus take all of the tree into account. This implicitly captures the RT topology and reveals information about its clades, which further implies that a change in the underlying mass distribution affects the imbalance of all edges in the tree.

The transformation into imbalances can reveal differences in the placement mass distribution of nearby branches of the tree. This is in contrast to the KR distance (see Section 2.5.4 below), which yields low values for masses that are close to each other on the tree. Note that the imbalance of a leaf edge is simply the total mass of the tree minus the mass of the edge. It thus contains mostly irrelevant information and can often be omitted.

The concept of edge imbalances has similarity with the Phylogenetic Isometric Log-Ratio (PhILR) transformation [330]. In this approach, a *balance* between the OTU

abundances in the two subtrees below a node of a rooted binary phylogenetic tree is calculated. This balance thus expresses which of the two subtrees has more OTUs in it. The transformation into balances yields orthogonal components, which are not compositional. Thereby, balances alleviate the normalization issues described above, and can hence be used with many standard analysis methods. Balances can also be computed for the two sides of a tree induced by an edge, suggesting the conceptual similarity to imbalances. We hence developed an adaptation of the concept to phylogenetic placement data, which we present in Chapter 6.

Placement Data Matrices

An example of the edge masses and edge imbalances for a sample is shown in Figure 2.14(a). These values can be summarized by two matrices, which we use for many downstream edge- and clade-related analyses, respectively. In these matrices, each row corresponds to a sample, and each column to an edge of the RT. These matrices have dimensions $m \times n$ for m edges of the RT and n samples. In other words, the edge masses matrix collects the edge mass vectors mentioned above, while the edge imbalance matrix transforms these masses as described before. Note that these matrices can either store absolute or relative abundances, depending on whether the placement mass was normalized or not.

Furthermore, many studies provide meta-data for their samples, for instance, the pH value or temperature of the samples' environment. Such meta-data features can also be summarized in a per-sample matrix, where each column corresponds to one feature. The three matrices are shown in Figure 2.14(b). Quantitative meta-data features are the most suitable for computational purposes, as they can be used for calculations such as detecting correlations with the placement mass distributions of samples, see for instance Section 4.2.2.

2.5.4 Distances between Samples

Given a set of metagenomic samples, one key question is how much they differ from each other. Pairwise distances are valuable for downstream tasks, for instance, clustering algorithms such as UPGMA [202, 254, 333] and ordination methods such as *principal component analysis* (PCA) [168, 286], or gradient discovery in microbial communities with respect to meta-data features.

General Metagenomic Distance Measures

In many comparative ecology studies, commonly used methods for assessing sample (dis-)similarity are based on sequence abundances, or OTU count tables (as introduced in Section 2.5.3). For example, the Jaccard index [162] and the related Sørensen-Dice coefficient [83, 336] use the presence/absence of species in two samples to measure their similarity and diversity. The Bray-Curtis dissimilarity [34, 202] furthermore uses abundances, that is, species counts, in order to quantify compositional dissimilarity between samples. Note that working with such similarity indices entails certain pitfalls, in particular when comparing them across studies [32].

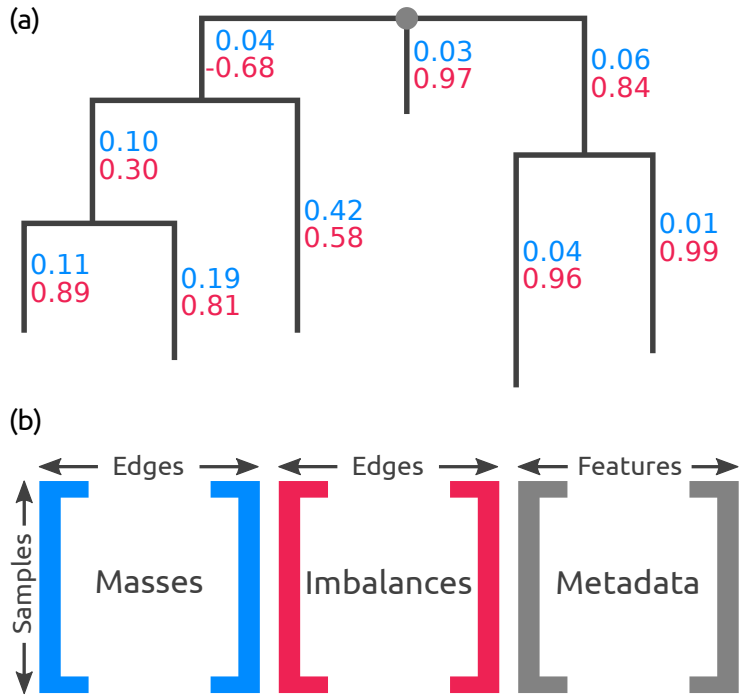

Figure 2.14: Edge Masses and Imbalances. (a) Reference tree where each edge is annotated with the normalized mass (first value, blue) and imbalance (second value, red) of the placements in a sample. The depicted tree is unrooted, hence, its top-level trifurcation (gray dot) is used as "root" node. (b) The masses and imbalances for the edges of a sample constitute the rows of the first two matrices. The third matrix contains the available meta-data features for each sample. These matrices are used to calculate, for instance, the Edge PCA (Section 2.5.5) or correlation coefficients.

These indices do however not take the evolutionary history of the sequences into account. One method that does use the relatedness of the species under study is the UniFrac distance [221, 223]. To compare two metagenomic samples, a phylogenetic tree is employed, either by inference from all sequences of the two samples, or by assigning the sequences to the tips of an existing tree. Then, the branches of the tree are marked as either shared or unique, depending on whether they lead to taxa that appear in both or only one of the samples. The distance is computed as the fraction of total unique branch lengths, which satisfies the requirements of a distance metric. The UniFrac distance can be calculated quantitatively (weighted UniFrac), or qualitatively (unweighted UniFrac), depending on whether sequence abundances are considered, or only their presence and absence is used.

The Phylogenetic Kantorovich-Rubinstein Distance

The idea of using phylogenetic distances on a tree to assess sample similarity has been extended and generalized to the context of phylogenetic placement in form of the *phylogenetic Kantorovich-Rubinstein* (KR) distance [104, 239]. In other contexts, the KR distance is also called Wasserstein distance, Mallows distance, or

Earth Mover's distance [206, 231, 296, 365]. The KR distance between two metagenomic samples is a metric that describes by at least how much the normalized mass distribution of one sample has to be moved across the RT to obtain the distribution of the other sample. In other words, it is the minimum work needed to solve the transportation problem between the two distributions. The distance is symmetrical, and increases the more mass needs to be moved, and the larger the respective displacement (moving distance) is.

The linear case of moving the mass of one distribution to transform it into another distribution is shown in Figure 2.15. This linear case corresponds to the path between two locations on a tree, and is thus a measure of evolutionary distance between these locations. It can be extended to a tree via post-order traversal, by starting at the tips and moving the mass differences towards the (arbitrary) root. In order to conduct the transformation, the two samples being compared need to have equal masses. Hence, the KR distance operates on normalized samples; that is, it compares relative abundances.

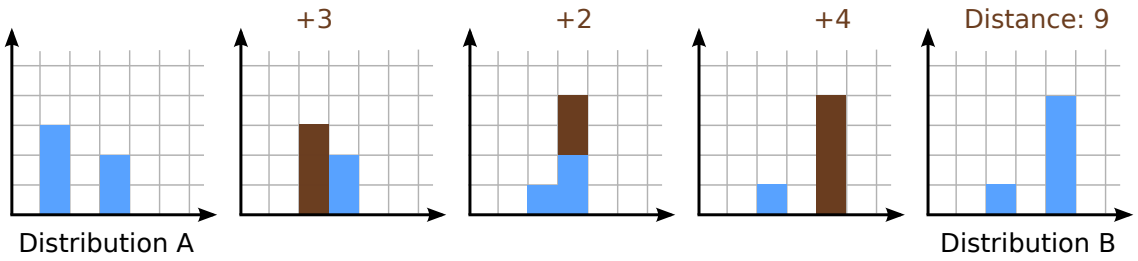

Figure 2.15: Linear KR distance. Distribution A is transformed into distribution B by moving mass along the axis, while keeping track of the moved distances. The dark blocks represent the masses being moved along the grid. For simplicity, masses and distances are discretized here; the continuous case works analogously.

As the tree needs to be traversed once per pairwise distance calculation, the computation of the phylogenetic KR distance is linear in the tree size, and in the number of placements. It is hence suitable for analyzing the large datasets of typical metagenomic studies. Note however that the computation of a pairwise distance matrix between the samples is quadratic in the number of samples. In the special case of assigning mass only to the tips of the tree (for example, by "placing" the QSs via similarity based methods such as BLAST, see Figure 2.12), the KR distance is equivalent to the weighted UniFrac distance [104].

The mathematical properties of the phylogenetic KR distance have been thoroughly examined by Evans and Matsen [104]. In summary, the KR distance can be formulated as an integral over distances λ along the branches of the tree T. Let $\tau(x)$ denote the subtree below point x on T for an arbitrary rooting, and let P and Q be the probability distributions of the two samples on the branches of T, given by the placement masses. Then, the KR distance can be expressed in closed form as

$$\mathrm{KR}(P, Q) = \int_T \Big| P(\tau(x)) - Q(\tau(x)) \Big| \; \lambda(dx) \tag{2.15}$$

This notation treats the placements as a continuous distribution over the branches of the tree instead of a collection of point masses, and hence describes a more general form of the KR distance. The distance can further be generalized by introducing an additional parameter p, with $0 < p < \infty$, which controls the impact of mass relative to transport [297, 298]:

$$\mathrm{KR}_p(P, Q) = \left[\int_T \left| P(\tau(x)) - Q(\tau(x)) \right|^p \lambda(dx) \right]^{\min(1/p, 1)} \tag{2.16}$$

Large $p > 1$ emphasize the impact of mass differences, while small $p < 1$ increase the influence of the distance traveled. In typical applications however, the default of $p = 1$ is used, which yields Equation 2.15 and corresponds to the physical interpretation of mass movements.

2.5.5 Existing Analysis Methods

The pairwise KR distance matrices between metagenomic samples as introduced in Section 2.5.4 above can be used in conjunction with general-purpose data analysis methods, such as PCA [168, 286] and UPGMA [202, 254, 333]. Although appropriate to apply, such methods do not make use of the fact that the distances were calculated on a phylogenetic tree. Taking this information into account however improves the capacity for visualization and interpretation of the results. To this end, the ordination method *Edge PCA*, as well as the clustering method *Squash Clustering* have been developed [239]. As we later compare our novel methods to these existing ones, we briefly introduce them here.

Edge PCA

The *edge principal components analysis* (Edge PCA) [239] is a method that utilizes the imbalance matrix (as explained in Section 2.5.3) to detect and visualize edges with a high heterogeneity of mass difference between samples. In particular, it computes the principal components of the imbalance matrix, using standard PCA. The result can be interpreted as a weighted sum of variables that maximizes variance between samples.

Similar to standard PCA on other types of input matrices, such as count tables or pairwise distances, the principal components can be visualized in form of a scatter plot of samples along the principal component axes. Samples are separated from each other depending on their placement mass distribution, where each principal axis in the plot explains some additional variance between the samples. These plots can further be annotated with meta-data features, for instance, by coloring, thus establishing a connection between differences in samples and differences in their meta-data [339]. Examples of this are shown later on in Figure 3.11 and Figure 5.3.

In contrast to standard PCA, using the imbalance matrix allows for further, enhanced, visualizations. As the columns of the imbalance matrix correspond to edges of the tree (see Section 2.5.3), the resulting eigenvectors (principal components) can

be projected back onto to tree. Hence, while the scatter plots show *how* samples are separated from each other, these visualizations on the tree explain *why* they are separated. Each principal component yields a tree visualization, where edges are highlighted that are responsible for the observed differences in the corresponding principal axis of the plot. An example of this is shown later in Figure 4.4.

Squash Clustering

A fundamental task for a set of metagenomic samples consists in finding clusters of samples that are similar to each other according to some distance measure, such as the KR distance. Standard linkage-based clustering methods like UPGMA are solely based on the distances between samples, that is, they calculate the distances between clusters as a function of pairwise sample distances. Other methods such as Multivariate Regression Trees [79] can take environmental per-sample meta-data into account, but do not make use of the phylogenetic information provided by, e. g., a reference tree.

In contrast, *Squash Clustering* [239] is a method that also takes the intrinsic structure of phylogenetic placement data into account. It uses the KR distance (see Section 2.5.4) to perform agglomerative hierarchical clustering of samples. Instead

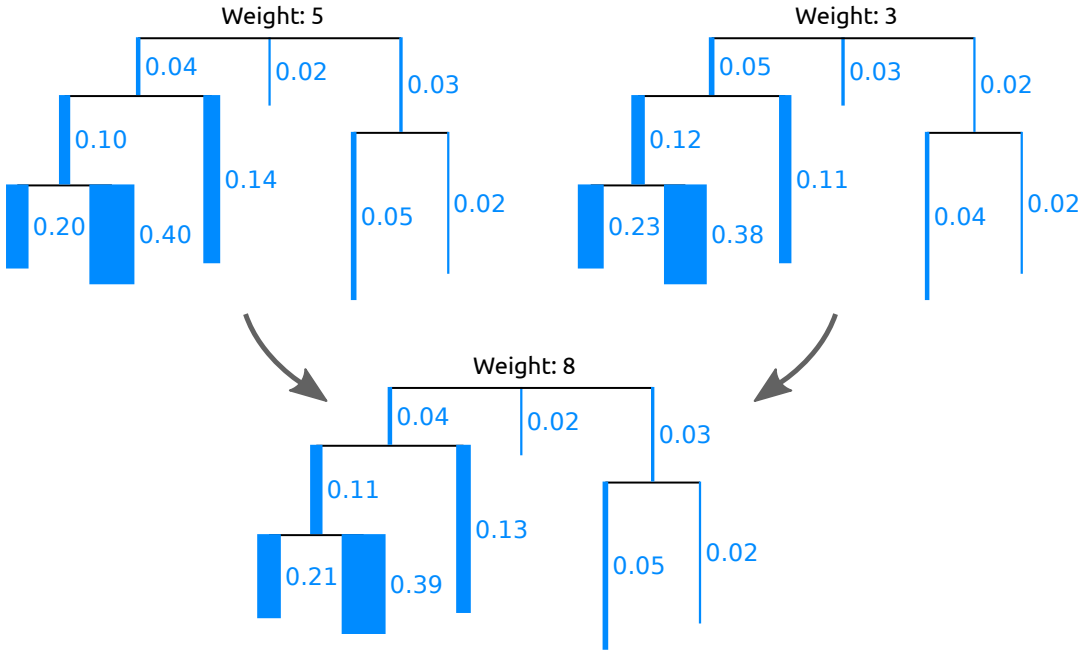

Figure 2.16: Squashing of edge masses. Two trees are merged (*squashed*) by calculating the weighted average of the respective mass distributions on their branches. By squashing, a cluster of (similar) samples can be summarized and visualized. For simplicity, we here show the masses per edge and visualize them as branch widths. In practice however, each placement location of each query sequence is considered individually. The figure is based on the similar Figure 3/2 of Matsen and Evans (2011) [239]; see there for more details on squashing.

of using pairwise sample distances, it merges (*squashes*, see Section 2.5.3) clusters of similar samples by calculating their weighted average per-edge placement mass, as shown in Figure 2.16.

Squash Clustering starts with each sample being its own cluster. Then, iteratively, it calculates the pairwise KR distance between all clusters of samples, and merges the pair of clusters that are closest to each other, until only one cluster containing all samples remains. The merging is conducted by squashing all samples in the respective clusters, using the total mass (number of samples) per cluster as weights.

This results in a hierarchical clustering tree, where tips correspond to samples, and branch lengths correspond to KR distances between clusters. Thus, in each step, Squash Clustering operates on the same type of data, namely, mass distributions on the fixed RT. In the beginning, each data item considered in the clustering is one sample, while later steps of the clustering operate on the merged (squashed) samples. Each inner node of the clustering tree represents a set of merged/squashed samples. Examples of such cluster trees are shown later on in Figure 3.9 and Figure 5.2. Furthermore, as the inner nodes of the cluster tree are again mass distributions, they can be visualized, and thus allow for interpreting the features of each set of merged samples; we show a similar visualization later in Figure 5.4.

3. Preprocessing

This chapter is derived from the peer-reviewed open-access publication:

> **Lucas Czech**, Pierre Barbera, and Alexandros Stamatakis. "Methods for Automatic Reference Trees and Multilevel Phylogenetic Placement." *Bioinformatics*, 2018, Volume 35, Issue 7, Pages 1151–1158.

The text, figures, and tables in this chapter were created by Lucas Czech, with the following exception: Pierre Barbera conducted the processing and analysis of the CAMI Challenge data as described in Section 3.3.4, and helped to write the part of the above publication on which that section is based.

3.1 Background and Motivation

Molecular sequencing costs are decreasing exponentially, leading to unprecedented amounts of genetic sequence data, as explained in Section 2.2.2. In most metagenomic studies, an initial analysis step consists in assessing the evolutionary provenance of the sequences. Phylogenetic placement, as introduced in Section 2.5, can be employed to determine the evolutionary position of sequences with respect to a given reference phylogeny. This is particularly helpful for studying new, unexplored environments, for which no closely related sequences exist in reference databases yet [230]. However, the selection process of suitable reference sequences for inferring a reference tree is typically conducted manually. This constitutes a major challenge and hindrance for studying such environments with placement methods.

This limitation also concerns the use of phylogenetic placement for taxonomic assignment. In studies that specifically look for certain kinds of organisms, e.g, protists [230], it usually suffices to use a taxonomy covering the organisms of interest, potentially including some outgroups from more distantly related species to also catch

outliers. As metagenomic analyses become cheaper, it is however to be expected that researchers want to target more than one group of organisms within one study. Particularly in cases where the environment contains a yet unknown diversity of organisms, this hence necessitates to use a broad reference that covers many taxonomic clades. At the same time however, the number of taxa in the reference phylogeny should be small enough to allow for visually inspecting and interpreting the results.

Lastly, phylogenetic placement methods have generally already reached their scalability limits: They require a higher computational effort with respect to the placement algorithms *per se*, but also the pre- and post-processing, than, for instance, similarity-based methods such as BLAST (see Section 2.5.2).

3.2 Methods and Implementation

In this chapter, we introduce methods to overcome the aforementioned limitations, that is, to (1) automatically obtain a high-quality reference tree for conducting phylogenetic placement, (2) split up the placement process into two steps using smaller phylogenies, and (3) accelerate the computation of placements via appropriate data pre-processing approaches. All methods are implemented as part of our GAPPA tool; see Appendix C for implementation details.

3.2.1 Phylogenetic Automatic (Reference) Trees

Motivation

Molecular environmental sequencing studies, particularly those that aim to conduct phylogenetic placement of Query Sequences (QSs), often rely on a set of manually selected and aligned reference sequences to infer a Reference Tree (RT) [78, 230, 353, 356]. Creating and maintaining databases of such reference sequences constitutes a labor-intensive and potentially error-prone process. Moreover, this approach is impractical for highly diverse samples that comprise sequences from many taxonomic clades, or samples obtained from unexplored environments, where it is yet unknown which reference sequences are necessary. Lastly, even if a large RT is available, the visualization of placements on such a large RT might be confusing and thus hard to interpret.

The RT used for phylogenetic placement should ideally (a) cover all major taxonomic groups that occur in the QSs, (b) use high-quality error-free reference sequences, and (c) not be too large to allow for unambiguous visualization and interpretation. These criteria can be met for small datasets by manually selecting curated sequences from databases, potentially informed by literature describing these sequences. In order to increase coverage, often additional sequences are selected based on their similarity to the already selected ones. For large and taxonomically diverse samples one key challenge is that sequence databases such as GREENGENES [82], UNITE [1], PR2 [138], EZTAXON [179], SILVA [294], and RDP [58] maintain reference collections of thousands to millions of taxonomically annotated sequences. Therefore, one needs to

appropriately sub-sample sequences such that the RT can be inferred in reasonable time *and* that it sufficiently covers the diversity of the sample.

Previous approaches mainly relied on phylogenetic diversity [106, 259, 279] and related methods [244]. The major drawback is that they require a comprehensive phylogeny as input. Inferring such large comprehensive phylogenies with hundreds of thousands of taxa, to subsequently reduce the taxon set again, is computationally inefficient and in certain cases infeasible.

To this end, we present a computationally efficient approach for obtaining sequences from large databases to infer an RT, which we call the *Phylogenetic Automatic (Reference) Tree* (PhAT) method. The workflow of the method is summarized in Figure 3.1. The input of our method is a database of aligned sequences of known species, including their taxonomic labels. Our approach then identifies sets of sequences that are similar to each other based on their entropy. It subsequently reduces the sequences in these sets to a predefined number of consensus sequences. This set of sequences is the output of our method. It represents the taxonomic clades and is then used to infer an RT for conducting phylogenetic placement analyses.

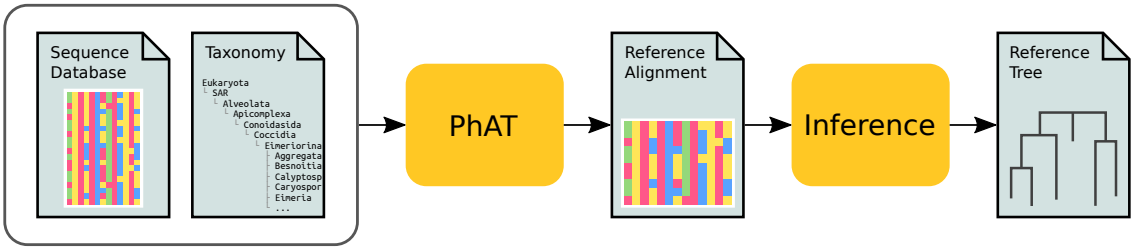

Figure 3.1: PhAT pipeline. The figure shows how our PhAT method is used to obtain a reference tree for phylogenetic placement using a database of reference sequences and their associated taxonomy as input.

Sequence Entropy

Firstly, the approach requires a measure to assess the similarity of a set of sequences to each other, that is, their ensemble similarity. Conventional methods for sequence similarity are often based on edit distance and other pairwise comparison methods [8, 268, 332]. This however necessitates to transform the pairwise distances to some form of ensemble measure that describes the similarity of all sequences to each other, for which there is no obvious approach [402]. There also exist methods that describe genetic variation and nucleotide diversity of sets of sequences [30, 269] which could be used for this purpose.

Instead, we here use entropy [325] to define a measure for quantifying the ensemble similarity of a set s of sequences. Variants of sequence entropy have been used before in numerous biological and phylogenetic contexts, for example, to asses the information content of sequences [59, 64, 207, 319, 366–368], or to measure substitution

saturation [387]. Here, we use entropy for alignment sites, that is, we define the entropy (uncertainty) H at alignment site i as

$$H_i \;=\; -\sum_c f_{c,i} \cdot \log f_{c,i} \tag{3.1}$$

where $c \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}, \mathtt{-}\}$ is the set of nucleotide states including gaps, and $f_{c,i}$ is the frequency of character c at site i of the alignment. Including gaps (-) in the summation reduces the contribution of sites that contain a large fraction of gaps. Their contribution is weighed down as all standard phylogenetic inference tools model gaps as undetermined states, that is, they do not contribute anything to the likelihood score. The entropy is 0 for sites that only contain a single character. It increases the more different characters an alignment site contains, *and* the more similar their frequencies are. Its maximum occurs if all characters appear with the same frequency (each of them 20%). Some examples for the entropy when using four nucleobases are shown in Figure A.1. Note that we also treat ambiguous characters as gaps (see Section 2.2.4). As only 0.008% of the non-gap characters in our test database (SILVA) are ambiguous, their influence is negligible. Ambiguous characters could however be incorporated by using fractional character counts.

Finally, the total entropy of a set s of aligned sequences is simply the sum over all per-site entropies: $H(s) = \sum_i H_i$. It is also possible to normalize this value by dividing it by the length of the alignment to obtain comparable values across alignments. Here, this however does not make a difference, as we are always comparing sequences with the same length. We use this entropy to quantify the ensemble similarity of a set of sequences. This can be regarded as an information content estimate of the sequences.

Sequence Grouping

The goal of this step is to group the sequences of a database into a given target number of groups/sets, such that the groups reflect the diversity of the sequences in the database. At the same time, the number of sequences needs to be small enough such that a maximum likelihood RT can be inferred in reasonable time.

A possible approach would be to use agglomerative clustering: In each step, the sequences that have the lowest entropy are clustered until the desired number of reference sequences is reached. A supposed advantage of this approach is that it does not rely on any taxonomic information (in contrast to our approach presented below). This procedure is however computationally expensive, having a complexity in $\mathcal{O}(n^2 \log n)$, and is thus not applicable to large databases with millions of sequences. Furthermore, the resulting sequence clusters can generally not be assigned unambiguous taxonomic labels, that is, they lack an interpretable naming scheme. This severely limits the types of useful post-analyses that can be executed; we did therefore not explore this approach.

Instead, we use the taxonomic information (see Section 2.3.1) of the sequence database to identify potential candidate groups of sequences that could be represented

by a consensus sequence (see Section 2.2.4). We interpret a taxonomy as a sequence labeling, where similar sequences have related labels. Thus, a taxonomy represents a pre-classification of similar sequences that can be exploited to group them. For instance, Figure 3.2 shows six sequences (S1–S6), which the underlying database has classified into the *Calyptosporidae* clade of the taxonomy, and which thus form a group of related sequences. Note that this assumes that the taxonomic classification is correct for most sequences [191]; we assess this problem later in Section 3.3.2. The goal is then to find clades/groups of sequences that represent the diversity of the database.

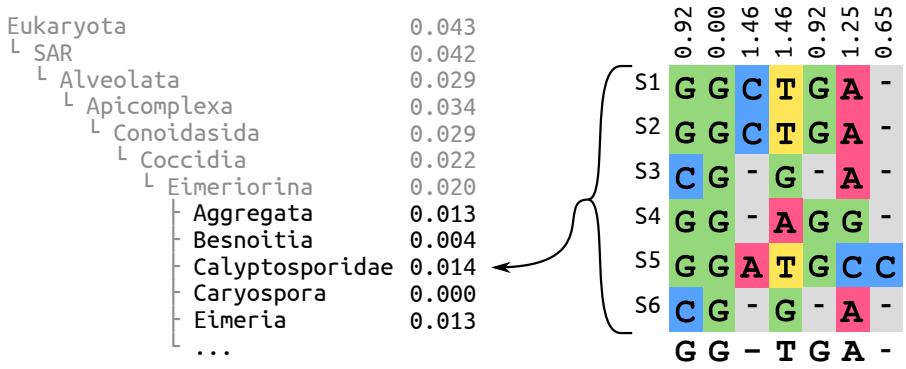

Figure 3.2: Entropy and consensus sequence of a taxonomic clade. The left hand side shows the exemplary clade *Eimeriorina* in its taxonomic context, listing its super- and sub-clades with the normalized entropy of their respective sequences. The right hand side is an excerpt from the alignment of six sequences that belong to the *Calyptosporidae* sub-clade. At the top of the alignment, the per-site entropies of the alignment sites are shown. At the bottom, the majority rule consensus sequence is shown, which is used to represent the sub-clade.

To this end, for a clade t of the taxonomic tree, we denote by $H(t)$ the entropy of all sequences that belong to that clade, including all sequences in its sub-clades, that is, its lower taxonomic ranks. Clades with low entropy imply that they contain highly similar sequences that can in turn be represented by a consensus sequence without sacrificing too much diversity. Inversely, clades with high entropy contain diverse sequences, implying that a consensus sequence is not likely to sufficiently capture the inherent sequence diversity. It is thus better to expand these clades and construct separate consensus sequences for their respective sub-clades. Examples of the per-site entropies are shown in Figure 3.2. As the clade structure of a taxonomy forms a tree, this criterion can then be applied recursively, as shown in Algorithm 3.1.

The algorithm works as follows: We initialize a list of candidate clades with the highest ranking clades that we want to consider. In the most general case, these can be “Archaea”, “Bacteria”, and “Eukaryota”. We then select the most diverse candidate clade, that is, the clade t whose sequences exhibit the highest entropy $H(t)$. This clade is then expanded, and we do not consider it as a potential candidate for building a consensus sequence. The high entropy clade is then removed from our list and its immediate sub-clades are added as new candidates to the list. Finally,

Algorithm 3.1 Taxonomy Expansion

1: *Candidates* ← list of highest ranking clades
2: *TaxaCount* ← size of *Candidates*
3: **while** *TaxaCount* < *TargetCount* **do**
4: *MostDiverse* ← $\arg\max_{t \in Candidates} H(t)$
5: remove *MostDiverse* from *Candidates*
6: add sub-clades of *MostDiverse* to *Candidates*
7: *TaxaCount* ← *TaxaCount* − 1 + size of *MostDiverse*
8: **return** *Candidates*

the current count of how many candidates we have already selected is updated accordingly. By expanding clades with high entropy, we descend into the lower ranks of the taxonomy. On average, this decreases the entropy, because low ranking clades generally tend to contain more similar sequences. This process is repeated until our list contains approximately as many candidate clades as the desired target count of reference sequences, which is provided as input. As the sizes of expanded clades can vary substantially, the target count cannot always be met exactly. In our tests, the average deviation was 0.2%, see Table 3.1.

Given this list of clades from different taxonomic ranks, we can now compute the consensus sequences. For each clade, all sequences in that clade and its sub-clades are used to construct a consensus sequence, which represents the clade diversity, and serves as the reference sequence for that clade. An exemplary majority rule consensus sequence is shown in Figure 3.2 below the six sequences S1–S6. This has several advantages: If only a few sequences diverge from the majority of that clade, the entropy might underestimate the molecular diversity of a clade. The consensus sequence for such a clade however compensates for this. Using consensus sequences furthermore alleviates the impact of spurious and erroneous sequences in the database. A simple per-site majority rule consensus [76, 245] works well, but we also assessed alternative methods; see Figure 3.7 and Figure 3.8 for details.

The algorithm can start at any rank of the taxonomy in order to only group sequences from specific clades. It is computationally cheap, while still yielding reasonable representative sequences for large taxonomic clades. Note that it would also be possible to directly use the relative character frequencies at each site to obtain more accurate representations. Maximum likelihood-based phylogenetic inference tools do, in principle, not require discrete input sequences, as explained in Section 2.4.4. The likelihood model allows to account for uncertainty in the input data [111], although this is generally not implemented in the mainstream software packages, see also [193]. The above process yields a set of consensus reference sequences which capture the diversity of distinct taxonomic clades.

Inferring a Reference Tree

Once we have identified the consensus sequences, which are already aligned to each other, we can use them to infer a maximum likelihood tree, which is the resulting

Phylogenetic Automatic (Reference) Tree (PhAT). As each consensus sequence is associated with a taxonomic clade, the corresponding taxonomic path can be used to label the tips of the tree. Note that since clades with low entropy might not be expanded, the tip labels do not necessarily correspond to the species or genus level. Also, the PhAT will not necessarily be congruent to the taxonomy, unless the tree search is explicitly constrained by the taxonomy (see Section 2.4.3).

A PhAT satisfies all criteria we listed above: (a) all taxonomic groups occurring in the QSs can be covered by using a suitable taxonomy as input, (b) by using consensus sequences, potential sequencing errors can be alleviated, and (c) the size of the tree can be specified by the user. Nonetheless, the resolution of the trees is limited by the underlying taxonomy, see Section 3.3.3 and Section 3.3.4 for details. Thus, one needs to verify that the resulting tree is appropriate for the data to be placed on it. This however also holds for manually selected reference sequences, and is hence not a specific disadvantage of our method. Furthermore, using consensus sequences may obscure the degree of sequence diversity in sub-clades, which in turn can affect the accuracy of subsequent phylogenetic placements on that tree. Our algorithm can not fully compensate for this. We present a method to address both issues (insufficient tree resolution and obscured diversity) in the next Section.

3.2.2 Multilevel Placement

When conducting phylogenetic placement, the computationally limiting factors are (i) the number of QSs to be placed (addressed in the following Section 3.2.3) and (ii) the size of the RT (number of taxa) and corresponding alignment length (addressed here). Using RTs with more taxa increases the phylogenetic resolution of the placements, at the cost of increased computational effort for inferring the RT, aligning the QSs (if the RT is used in this step), and placing the QSs. Furthermore, longer reference alignments (if appropriate data is available in the first place) are required to accurately infer large trees under the maximum likelihood criterion [388], thus further increasing the computational costs. Lastly, placement on large trees that comprise reference sequences with high evolutionary distances can reduce placement accuracy [260]. Thus, using a large number of reference sequences is not always desirable in practice.

One solution is to divide the tree and its alignment into more conquerable subsets, for example as implemented in SATÉ [215, 216]. This approach has also been extended to phylogenetic placement in SEPP [260] and TIPP [272], which divide the tree into disjoint subsets of taxa and conduct placement on each of them separately. While yielding more accurate placements and taxonomic classifications in less computing time, these methods might still result in large reference trees, which are hard to inspect and visualize.

To address this issue, we present an approach called *Multilevel* or *Russian Doll* Placement, which is summarized in Figure 3.3. Instead of working with one large RT comprising *all* taxa of interest, we use a smaller, but taxonomically broader Backbone Tree (BT) for pre-classifying the QSs (first level), and a set of refined

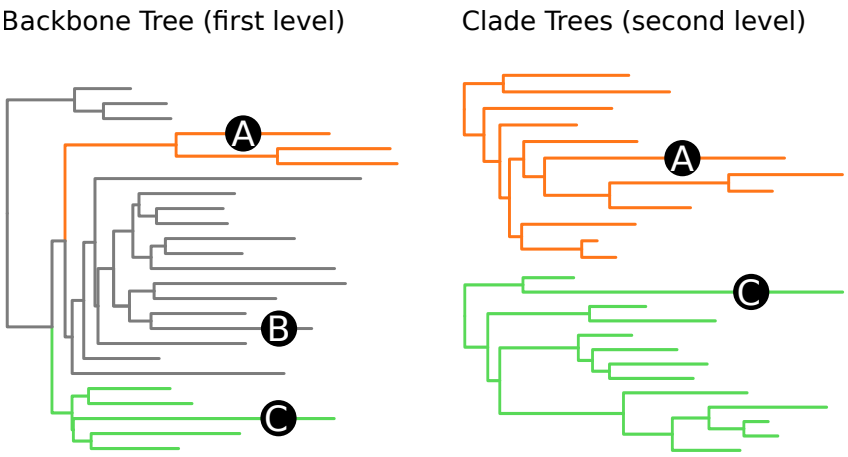

Figure 3.3: Multilevel placement. The left shows a backbone tree (BT); the right shows two clade trees (CTs) in orange and green. Branches in the BT that are associated with a CT are marked in its color. The trees "overlap" each other, meaning that each CT is represented by multiple branches in the BT. Three sequences A, B, and C are placed on the BT, which is the first level. A and C are placed on branches associated with a CT. Hence, their second level placement is conducted on the respective CT. B is placed on a branch that is not associated with any CT, and thus not used at the second level.

Clade Trees (CTs) for the final, more accurate placements (second level). These CTs comprise the reference sequences that are of interest for a particular study. For example, if a study is concerned with *Apicomplexa* and *Cercozoa*, a broad *Eukaryota* BT can be used for the first level, and two respective CTs for the second level, in analogy to [230]. Each CT is associated with the set of branches of a specific BT clade.

The method then works in three steps:

1. Align and place the QSs using the BT (first level).
2. For each CT, collect the QSs that are placed on the BT branches that are associated with the CT.
3. Align and place these QSs again, using their specific CTs (second level).

The workflow is also shown in Figure 3.4. Steps 1 and 3 follow the typical pipeline for phylogenetic placement as explained in Section 2.5.1. In step 2, there are two alternative ways to determine to which clade a QS belongs: (i) The most probable placement location is used to determine the branch a QS is placed on, or (ii) the LWRs of different locations/branches are accumulated and the QS is only considered to be part of a clade if more than a given threshold value of placement mass is placed within the clade. QSs with less placement mass within one clade are assigned to a special collection of "uncertain" query sequences.

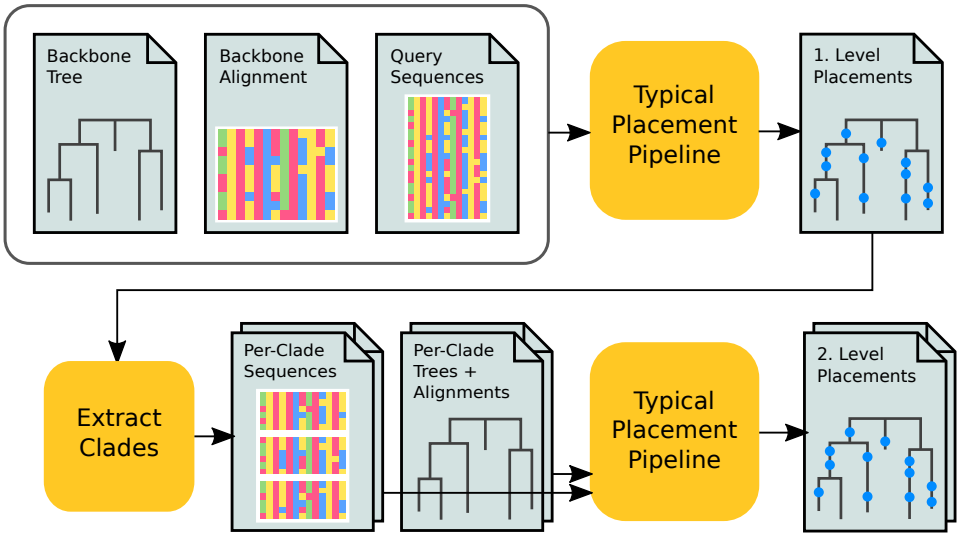

Figure 3.4: Multilevel pipeline for phylogenetic placement. The figure exemplifies the input and output of the multilevel placement approach as explained in the text. The yellow boxes correspond to the three steps of the workflow. See Figure 2.9 for a description of the typical placement pipeline.

While this approach requires some additional bookkeeping, the total computational cost is reduced, because the QSs do not have to be placed on all branches of all CTs. The speed gain depends on the sizes of the BT *and* the CTs with respect to the size of the substantially larger (often one order of magnitude or more) comprehensive tree. For example, by splitting a tree with 10 000 taxa into a BT and 10 CTs with 1000 taxa each, the computational cost decreases by a factor of 5 (two placement levels with 10% of the cost each). Furthermore, at each level, the amount of required main memory is reduced by a factor of 10 compared to the large tree. Lastly, this method allows for fine-grained control over the clades of interest at both placement levels:

Firstly, the BT provides a means for phylogenetically informed sequence filtering—that is, to identify and remove "spurious" QSs. Sequences with low similarity to known references are often removed in environmental sequencing studies [346]. However, using sequence similarity as a filter criterion can remove too many QSs, particularly when studying new, unexplored environments [230]. By using phylogenetic placement as a filter instead, substantially more sequences can be retained for downstream analyses. Only the QSs that are placed onto the inner branches of the BT, that is, branches with no associated CT, are omitted at the second placement level. Such placements may indicate that suitable reference sequences are missing from the RT, or that the respective QSs represent novel species. Either way, as these QSs are not well represented by the RT, they are not informative for most downstream analyses and can thus be removed. This therefore represents a phylogenetically informed sequence filtering method as an alternative to plain sequence similarity.

Secondly, using specific clade trees for lower level taxonomic clades offers the phylogenetic resolution that is necessary for downstream analyses and for biological reasoning. It is, for example, possible to use manually curated "expert" trees for each clade of interest at the second level.

In this setup, the BT is only used for pre-classification, and can, for example, use our PhAT method as presented in Section 3.2.1. The aforementioned issue of obscured diversity in sub-clades can be circumvented by "overlapping" the CTs with the BT. That is, a CT can be associated with several branches of the BT, so that placements on each of these BT branches are collected and placed onto the same CT. See Figure 3.3 and Figure 3.13 for examples of such an overlap. We recommend users to ensure that the branches of the BT that are associated with one CT are monophyletic, meaning that there is one split that separates these branches from the rest of the BT. This can be achieved by inferring the BT with a high-level constraint that maintains the monophyly of the CTs. It ensures phylogenetic consistency between the BT and the CTs, and improves the accuracy of the first placement level, as shown in Section 3.3.5. Lastly, it is also possible to use more than two levels, which might become necessary when working with RTs and datasets that are even larger than what is currently available.

3.2.3 Data Preprocessing for Phylogenetic Placement

Apart from the RT size, handling the sheer number of QSs also induces computational limitations for conducting phylogenetic placements. Most metagenomic studies publish their data in unprocessed formats, which are sometimes filtered to contain only reads from certain barcoding or marker regions. For instance, they store the raw sequencing output in `fasta` [287] or `fastq` [57] format (see Section 2.2.1). Those data often contain duplicates of exactly identical sequences, both *within* and *across* samples. Identical sequences are however treated the same in phylogenetic placement algorithms and therefore induce unnecessary computational overhead. Furthermore, sample sizes, that is, the number of sequences per sample, can vary by several orders of magnitude. For example, the "HM16STR" dataset of the Human Microbiome Project (HMP) [160, 250] contains an average of 12 911 sequences per sample, but also an outlier sample with 0 sequences and one with 403 211 sequences. If the placement algorithm is parallelized over samples, this leads to an uneven load balance across compute nodes. A potential solution is to initially cluster the sequences into OTUs (see Section 2.5.3), which however annihilates the accuracy benefit of using individually placed sequences.

In order to solve these issues, that is, reduce the computational cost and achieve good load balance, one can pre-process the sequences as summarized in Figure 3.5 and explained in the following (see Appendix C for implementation details). First, the query sequences are de-duplicated across all samples and fused into chunks of equal size, in a step which we call *chunkify*. The chunk size should be chosen such as to allow for aligning and placing a chunk within reasonable time on the intended hardware; for computer clusters for example, the wall time (that is, the total allowed

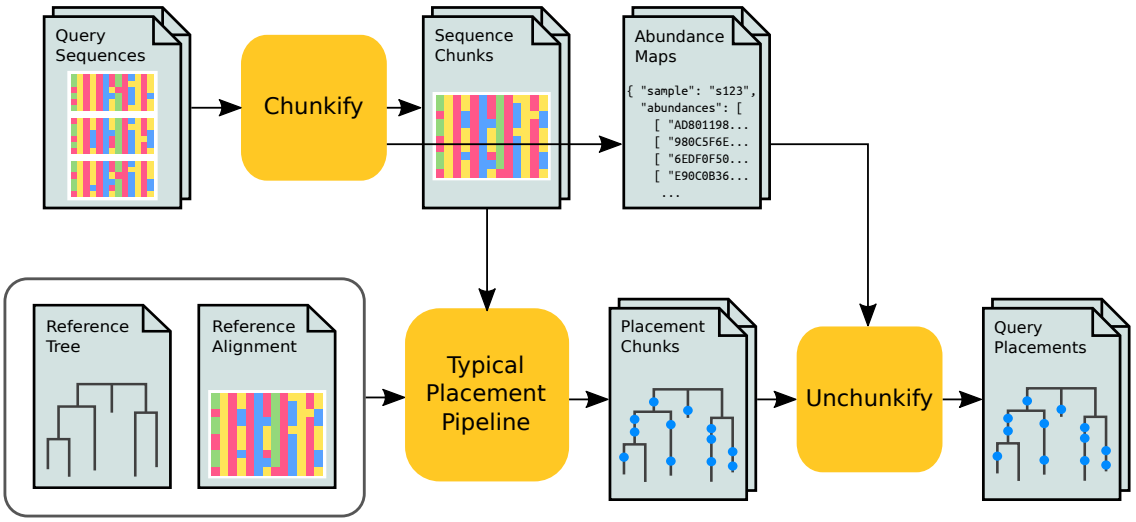

Figure 3.5: Pre-processing pipeline for phylogenetic placement. The figure shows our pre-processing pipeline that improves both speed and parallel efficiency of the phylogenetic placement computation. The gains are achieved by re-distributing and de-duplicating the query sequences of all samples into chunks of equal size prior to the placement computation. Final per-sample placement files are generated using mapping files that count of often each unique sequence appeared in each sample. See Figure 2.9 for the typical placement pipeline.

execution time of each job) should be considered. For modern hardware, we recommend chunk sizes of 50 000 or larger. Each unique sequence is assigned an identifier, and a list of abundance counts for each sequence in a sample is computed. This way, each strictly identical sequence is only processed once in the downstream steps.

Given an RT and its underlying alignment, the QS chunks are then aligned to the reference multiple sequence alignment, and subsequently placed on the RT, using a typical placement pipeline as explained in Section 2.5.1. In the last step, which we call *unchunkify*, the resulting per-chunk placement result files in combination with the per-sample abundance counts are used to generate final per-sample placement files, containing a placement for each sequence in the original sample.

The speedup induced by this preprocessing is proportional to the ratio of total versus unique sequences; the gain in parallel efficiency depends on the ratio of smallest to largest sample (in terms of number of sequences). This approach allows to analyze datasets that are orders of magnitude larger than in previous published studies. For example, in 2012, an analysis of Bacterial Vaginosis (BV) data placed a total of 426 612 sequences, thereof 15 060 unique, on an RT with 796 tips [339]. Using a prototype version of our implementation, we were able to analyze a neotropical soils dataset with 50 118 536 total sequences, thereof 10 567 804 unique, with an RT comprising 512 taxa [230]. Furthermore, to demonstrate the scalability of our method, and for the evaluation of the methods presented in later chapters, we also analyzed datasets with up to 116 520 289 total sequences, thereof 63 221 538 unique, from the HMP [160, 250], using RTs with up to 2059 tips. This corresponds to

a computational effort that is four orders of magnitude greater than for the BV study. For an overview and further details on the datasets used in our evaluation, see Appendix B. Without the improvements presented here, these datasets could not have been processed and placed in manageable time. As we use the placement results of these datasets on the PhATs in the evaluations of later chapters, these improvements represent necessary preprocessing steps for this work, and for large analyses using phylogenetic placement in general.

3.3 Evaluation and Results

3.3.1 Reference Tree Setup

To test the Phylogenetic Automatic (Reference) Tree (PhAT) method, we used the "SSU Ref NR 99" sequences of the SILVA database [294] version 123.1 and the corresponding taxonomic framework [395]. The database contains 598 470 aligned sequences from all three domains of life, classified into 11 860 distinct taxonomic labels, and mainly contains bacterial sequences. In detail, there are

- 22 913 sequences with 347 taxonomic labels for the *Archaea*,
- 62 436 sequences with 7441 taxonomic labels for the *Eukaryota*, and
- 513 121 sequences with 4072 taxonomic labels for the *Bacteria*.

The overall number of taxonomic labels is counted here, that is, it includes higher level labels. We use the SILVA alignment as-is, and thereby implicitly assume that it is of sufficient quality for our purposes; see Section 3.3.2 for an evaluation of this assumption.

Sequence Selection

We constructed four sets of consensus sequences from the SILVA database: a *General* set ("all of life"), as well as separate sets for the domains *Archaea*, *Bacteria*, and *Eukaryota*. The target sizes for the recursive expansion of taxonomic clades (see Section 3.2.1) were chosen to be large enough to cover the diversity well, while still being computationally feasible and visually interpretable for the subsequent steps. The target size for the *General* tree was set to 2000 taxa, while the *Bacteria* and *Eukaryota* tree were targeting 1800 domain-specific taxa, which is approximately reached, but not exactly (see Table 3.1). This is because the sizes of sub-clades in the taxonomy vary. Because each tip of the tree is a consensus sequence that represents the respective lowest taxonomic level, the number of available taxa is smaller than the total number of taxonomic labels in the SILVA database. For example, the *Archaea* have a total of 347 taxonomic labels across all ranks, but only 248 labels at *Genus* level. Thus, the *Archaea* tree used here comprises 248 taxa, which represents the fully resolved *Archaea* taxonomy at the *Genus* level. In the three domain-specific trees, we furthermore included consensus sequences at the

Phylum level of the respective two remaining domains, in order to make sure that the evaluation also works if such "outgroups" are included. The assembly of these four datasets required a total of about 30 min of runtime and 10 GB of memory on a standard laptop computer. This includes counting alignment characters, calculating entropies and constructing consensus sequences. The resulting dataset and tree sizes, as well as the fraction of sequences from each domain that the PhATs contain are shown in Table 3.1.

Table 3.1: Taxonomic composition of the four PhATs. The table lists the four trees used in our evaluation and their sizes (in number of sequences/tips), as well as how many of these tips originate from each of the three domains of life. The underlined values represent the resulting tree sizes, which slightly deviate from the intended target sizes (2000 and 1800 taxa, respectively). The *Archaea*, being a small enough clade, did not have a target size, but were fully resolved.

		Thereof number of		
Tree	Size	*Archaea*	*Bacteria*	*Eukaryota*
General	<u>1998</u>	210	508	1280
Archaea	511	248	205	58
Bacteria	1914	59	<u>1797</u>	58
Eukaryota	2059	59	205	<u>1795</u>

Our implementation of the method contains some further details that are worth mentioning for reproducibility: It is possible to constrain the maximal size of clades in order to not build a consensus sequence for an overly large clade, which might not be a good representative of that clade. For the same reason, it is possible to first expand the highest ranks of the taxonomy into separate candidates. We used conservative values for these two constraints (a maximal clade size of 2000 and an expansion of only the first two taxonomic ranks), in order to give more impact to the sequence entropy. Lastly, some clades contain only one sub-clade. Those are immediately expanded, as they do not change the length of the candidate list during the algorithm.

Tree Inference

Givens the four sets of consensus sequences (the *General* sequences, as well as the three domain-specific sequences), we then inferred unconstrained and constrained maximum likelihood trees, running 50 independent tree searches for each tree and subsequently selecting the best-scoring tree. Unconstrained trees were inferred using RAxML 8.2.8 [342]. Constrained trees were inferred with Sativa 0.9-55 [191], which internally again relies on RAxML, and offers a convenient way to transform a taxonomy into a constraint tree. The unconstrained trees best adhere to the phylogenetic signal of the sequences and thus typically work better for conducting phylogenetic placement. The constrained trees comply with the Silva taxonomy,

as this compliance might be necessary in comparative studies. They are used here to assess how taxonomic constraints affect the phylogenetic placement and the subsequent analyses. In total, our setup yields eight distinct RTs for evaluation: the *General* tree, the three domain trees, and the respective taxonomically constrained variants. Figure 3.13 shows the unconstrained *Bacteria* tree as an example.

The relative Robinson-Foulds distances [304] (see Section 2.3.2) between the four pairs of trees (unconstrained versus constrained) range between 45.8% and 49.7%. These differences probably occur because our trees span diverse clades, whose ancient branches are hard to resolve. Also, single gene data might not be sufficient to resolve these clades [199]. The differences between the trees however mostly concern inner branches. When conducting phylogenetic placement, QSs generally tend to be placed more towards the terminal branches of the tree. As these branches are more stable across our trees, the differences in the inner branches are thus acceptable for our evaluation purposes. Furthermore, we performed significance tests comparing the unconstrained trees to the constrained ones, as shown in Table 3.2. The tests show that in all cases, the unconstrained trees fit the sequence data significantly better, and should hence be preferred in cases where congruence with the taxonomy is not needed.

3.3.2 Accuracy

Measurement Method

Using the eight trees described above, we assess how using our PhAT affects phylogenetic placement accuracy. Each terminal branch of our RTs represents a consensus sequence, which is computed from corresponding species-level sequences in SILVA that share the same taxonomic label. We evaluate an RT by placing these species sequences onto the RT: Each species sequence is expected to be placed onto the branch leading to the consensus sequence representing this particular species sequence. As the consensus sequences are derived from the taxonomy, all terminal branches of the tree have taxonomic labels. These labels thus identify the expected placement position for each species sequence. For example, sequences `S1-6` in Figure 3.2 are represented by the consensus sequence for the *Calyptosporidae* clade, which is shown below the 6 sequences in the figure. They are thus expected to be placed onto the *Calyptosporidae* branch in the RT.

In order to conduct this accuracy evaluation, we placed the respective subset of the SILVA database species sequences onto each of the eight RTs. As the sequences in SILVA are already aligned to each other, no alignment step was necessary for this. We further removed sites consisting entirely of gaps from the alignment because they contain no phylogenetic signal. This was done to reduce the memory footprint of downstream analysis steps. Phylogenetic placement was conducted using EPA-NG [18].

We quantify the placement accuracy for a sequence by the distance to its expected placement branch. More precisely, we measured (a) the (discrete) number of branches between the actual placement and the expected branch, and (b) the

Table 3.2: Tree topology significance tests. Here, we report typical significance tests comparing the four pairs of unconstrained (U) and constrained (C) trees used in our evaluation. The tests were performed with IQ-TREE v1.5.6 [271] under the "GTR+G" model (see Section 2.4.2 and Section 2.4.3; the "+G" stands here for the Γ model of rate heterogeneity), using 10 000 resamplings for the RELL method [182]. The table shows that the unconstrained trees fit the sequence data significantly better in all four cases and for all test statistics.
Columns are as follows. logL and deltaL: log likelihood score and difference in log likelihood scores between constrained and unconstrained tree. bp: bootstrap proportion using RELL method [182]. p-(W)KH: p-value of the one sided and the weighted Kishino-Hasegawa test [181]. p-(W)SH: p-value of the (weighted) Shimodaira-Hasegawa test [327]. c-ELW: Expected Likelihood Weight [347]. p-AU: p-value of approximately unbiased (AU) test [326].

Tree	logL	deltaL	bp	p-KH	p-WKH	p-SH	p-WSH	c-ELW	p-AU
General (U)	-725199.040		1.0	1.0	1.0	1.0	1.0	1.0	0.9987
General (C)	-731949.568	6750.528	0.0	0.0	0.0	0.0	0.0	0.0	0.0012
Archaea (U)	-131862.815		1.0	1.0	1.0	1.0	1.0	1.0	1.0000
Archaea (C)	-133110.463	1247.648	0.0	0.0	0.0	0.0	0.0	0.0	0.0000
Bacteria (U)	-405028.378		1.0	1.0	1.0	1.0	1.0	1.0	1.0000
Bacteria (C)	-412464.820	7436.442	0.0	0.0	0.0	0.0	0.0	0.0	0.0000
Eukaryota (U)	-745442.969		1.0	1.0	1.0	1.0	1.0	1.0	1.0000
Eukaryota (C)	-753944.998	8502.030	0.0	0.0	0.0	0.0	0.0	0.0	0.0000

(continuous) distance in branch lengths units. The former is more important in the context of phylogenetic placement, as the placement branch of a sequence is more significant in most analysis methods than its exact location on that branch. As a sequence can have multiple placement locations, the distances are in fact weighted averages that incorporate the placement probabilities (LWRs, see Section 2.5.1). For sequences with a clear phylogenetic signal, that is, one placement with a high LWR, the averaging procedure only slightly affects the measurements. However, sequences with less clear placement locations are accounted for more precisely this way.

Results for the Unconstrained and Constrained Trees

The placement accuracy results for the four unconstrained and the four constrained trees are shown in Figure 3.6. Further details of the trees and the respectively achieved placement accuracies are provided in Table 3.3.

Considering the size of the trees, most sequences are placed in close vicinity to their expected branches. This is corroborated by the short average distances reported in Table 3.3. Furthermore, the average expected distance between placement locations (EDPL) [241] is low, indicating that the placements of a specific sequence mostly cluster in a small neighborhood of the tree. We observed that errors occur mostly in parts of the tree with short branches. This might be explained by the inability of 16S SSU sequences to properly resolve certain clades [165]. Also, the placement likelihood differences are small between neighboring, short branches, such that the placement signal is fuzzy.

With 77% of the sequences placed exactly on their expected branch, the accuracy is generally lowest for the *Bacteria* tree. This might be because the *Bacteria* have the most sequences in SILVA, that exhibit a high diversity. In the other three trees, more than 90% of the sequences are placed at most one branch away from their respective expected branch. The constrained trees exhibit similar placement accuracy, indicating that the topological differences in the inner branches of the constrained versus unconstrained trees indeed do not substantially affect the placement accuracy. Finally, we note that the results are reported without any manual corrections, and use overly broad RTs. Thus, in real world studies, where trees are often more specific for a clade of interest, better results are to be expected. Particularly when using Multilevel Placement with overlapping RTs, placement differences of a few branches at the first level tree are acceptable, as they do not change the second level tree on which the sequence is placed; see Section 3.3.5 for details.

Alternative Consensus Methods

As outlined in the method description (see Section 3.2.1), we represent clade diversity via majority rule consensus sequences. To assess the impact of the consensus method on the placement accuracy, we repeated the above evaluation using alternative consensus methods. In particular, we used Cavener's method [50, 51], and threshold consensus sequences [75, 76]. As shown in Figure 3.7, we found little difference between the methods.

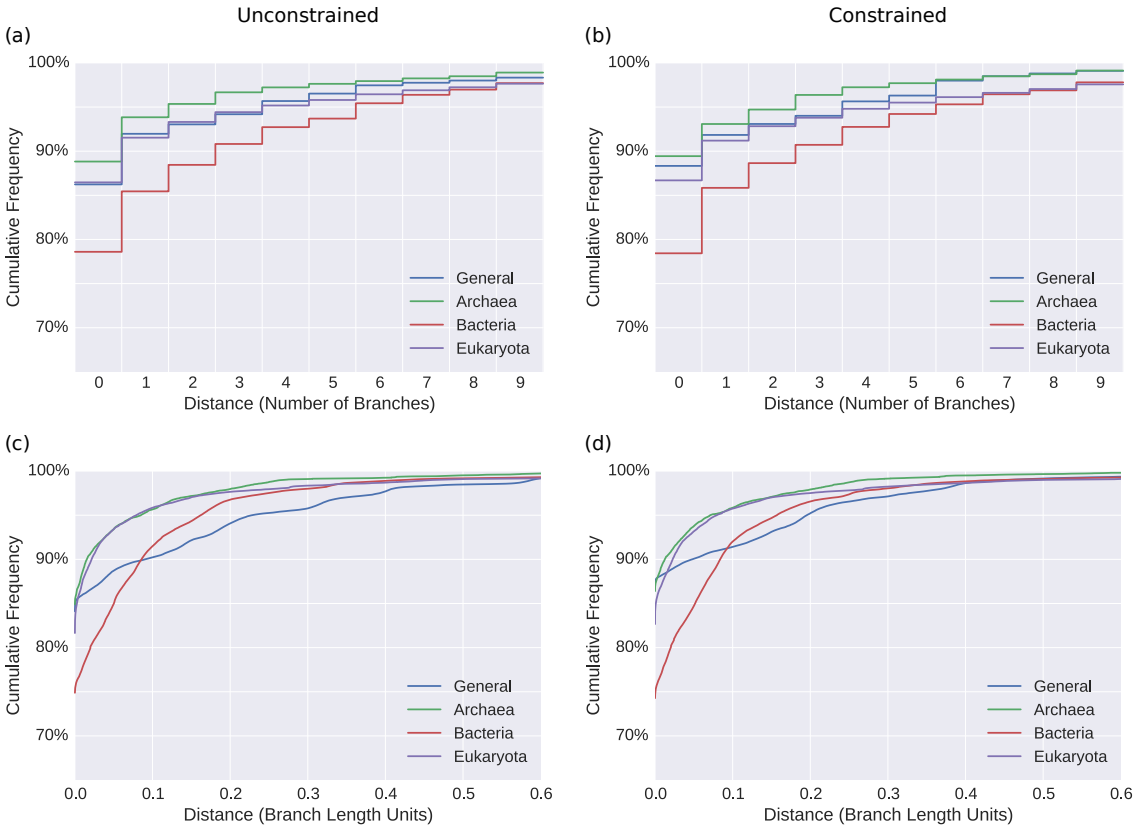

Figure 3.6: Induced placement accuracy on the unconstrained and constrained Phylogenetic Automatic (Reference) Trees (PhATs). We evaluated the accuracy of our PhATs by placing sequences and measuring the weighted distances to their respective expected placement branches. The figure shows the cumulative frequencies of number of sequences versus distances, measured in number of branches (top row, Subfigures (a) and (b)) and in branch length units (bottom row, Subfigures (c) and (d)). In other words, it shows how many sequences are placed within a certain radius from their expected branches. For example, in (a), more than 85% of the sequences of the *Bacteria* (red) are placed within a radius of at most one branch from their expected branch, and in (c), more than 95% of the *Eukaryota* (purple) are within a radius of 0.1 branch length units from their expected branches.
The figure compares the accuracy of using the unconstrained trees (left, Subfigures (a) and (c)) to using the SILVA taxonomy as constraint for the tree inference (right, Subfigures (b) and (d)). As explained in Section 3.3.1, the differences between the unconstrained and constrained trees mostly concern their inner branches, and are thus not expected to affect the accuracy to a large extent. This is confirmed by the fact that, overall, the results are similar between the constrained/unconstrained tree pairs. A slight improvement can be observed for the constrained General tree (blue), which performs better according to both distance measures. In most other cases, no significant differences can be observed.

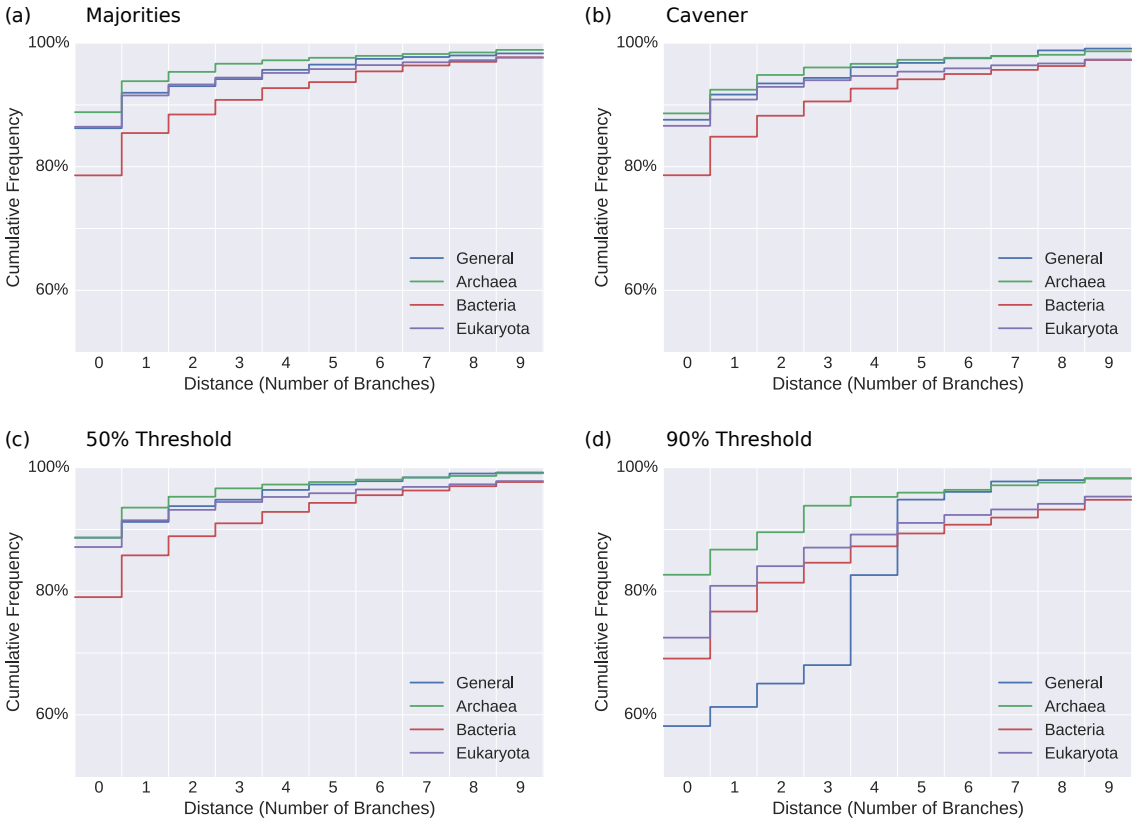

Figure 3.7: Effect of alternative consensus sequence methods on accuracy. In the main evaluation of our PhAT method, we use reference trees and alignments based on majority rule consensus sequences [76, 245] of the SILVA database sequences. Here, we evaluate the effect of using alternative consensus sequence methods on phylogenetic placement accuracy. In addition to (a) majority rule consensus, we tested (b) Cavener's method [50, 51], as well as threshold consensus sequences [75, 76] using thresholds of 50%, 60%, 70%, 80%, and 90%, of which two are shown in (c) and (d). The three remaining threshold methods exhibit accuracies that lie almost exactly in between the shown plots, that is, accuracy decreases with increasing thresholds. For comparison, we also included Figure 3.6(a) again, here as Subfigure (a), using the same y-axis scaling as for the other plots. All trees used in this part of the evaluation are unconstrained. We only show distances measured in number of branches here, because this is more relevant in the context of our methods.

Table 3.3: Overview of the Phylogenetic Automatic (Reference) Trees (PhATs) and their evaluation statistics. Details of the four unconstrained (U) and the four constrained (C) trees are shown. "Size" is the number of leaves in the tree, that is, the number of consensus sequences that the tree was inferred from, see Table 3.2. "% Seqs." is the percentage of sequences from SILVA placed on it. The *General* tree does not cover all sequences, because there are some sequence labels in the database that could not be mapped to the taxonomy. "∅ Br. Len." is the average branch length in the tree. The evaluation results are reported in the remaining columns: Average distances of the placed sequences to their respective expected branch are listed in numbers of branches (Discrete) and in branch length units (Continuous), as explained in the text. Furthermore, "Exp. Br. Hits" shows how often the most probable placement was located exactly on the expected branch. Lastly, the average expected distance between placement locations (EDPL) is shown. The EDPL is the sum of the pairwise distances between the placements of a sequence weighted by their probability [241].

				Average Distance			
Reference Tree	Size	% Seqs.	∅ Br. Len.	Discrete	Continuous	Exp. Br. Hits	∅ EDPL
General (U)	1998	98.7%	0.084	0.63	0.034	85.9%	0.00058
General (C)	1998	98.7%	0.086	0.57	0.027	88.2%	0.00046
Archaea (U)	511	3.4%	0.070	0.46	0.013	86.4%	0.00038
Archaea (C)	511	3.4%	0.071	0.45	0.013	88.2%	0.00041
Bacteria (U)	1914	84.6%	0.067	1.13	0.031	77.0%	0.00095
Bacteria (C)	1914	84.6%	0.071	1.11	0.031	76.6%	0.00091
Eukaryota (U)	2059	10.0%	0.080	0.79	0.022	84.9%	0.00032
Eukaryota (C)	2059	10.0%	0.083	0.81	0.024	85.7%	0.00031

By using alternative consensus methods, the consensus sequences and thus the alignment sites change. Furthermore, we used random initial starting trees for the maximum likelihood tree inference. Hence, the obtained reference trees (not shown) differ substantially from each other. Across the corresponding trees of the tested consensus methods, that is, when comparing, e. g., the *Bacteria* trees obtained with different consensus methods to each other, we observed an average relative Robinson-Foulds (RF) distance [304] of 49.5%; see Section 2.3.2 for details on the RF distance. This is similar to our findings depicted above, e. g., in Figure 3.6. For the different consensus methods, again, the accuracy for the respective constrained variants of the trees (data not shown) does not change substantially compared to the accuracy obtained for the unconstrained trees shown in Figure 3.7. Thus, the accuracy differences shown in the figure are most likely due to the interplay of alignment and placed sequences (which is what we are interested in), and not due to differences in the trees (which are not of interest here).

The first three plots in Figure 3.7(a)–(c) exhibit similar accuracies. On average, majority rule, Cavener's, and low threshold ($\leq 70\%$) consensus methods place 82–83% of the sequences on the expected branch. As a general trend, the *Archaea*, being the smallest tree, tend to have the highest accuracy. On the other hand, the *Bacteria*, for which there is by far the largest amount of sequences in SILVA, perform worst. This changes for high consensus thresholds. At high thresholds, many sites contain ambiguity characters that blur the phylogenetic signal. The *General* tree, representing the highest diversity, is most affected by this, as can be seen in the last two plots Figure 3.7(c)–(d).

Non-Consensus Sequences

For the above evaluations of the PhAT method, we used some form of consensus sequence representation for the clades of the taxonomy, see e. g., Figure 3.6 and Figure 3.7. However, we also tested how the method behaves when using actual representative sequences from the database instead to represent the taxonomic clades. This aims to avoid unnecessary blur of the phylogenetic signal, and other potential drawbacks of consensus sequences.

As manually selecting representative sequences from the database was not practical, we used the following automated approach. First, we took the 90% threshold consensus sequences of the PhAT method that were already evaluated in Figure 3.7(d). By using a high threshold, a large fraction of the diversity of each clade is included. Then, for each such consensus sequence, we calculated a score for all sequences from the database that were used to construct this consensus sequence, by measuring the number of differing nucleotides between the consensus sequence and the database sequence. The sequence with the lowest score (that is, with most matching nucleotides) was then used to represent the clade. Thereby, the taxonomic clades are represented by actual sequences from the database. However, as these sequences are close to the respective consensus sequence, they are still "good" representatives of the diversity of the clade. This resulted in a set of sequences of the same size (both in number of sequences, and in number of alignment sites) as the original set

of consensus sequences. Using these sequences, we then again inferred a tree and conducted the evaluation procedure by placing all sequences of the database on that tree, as described before. The results for the four unconstrained trees are shown in Figure 3.8.

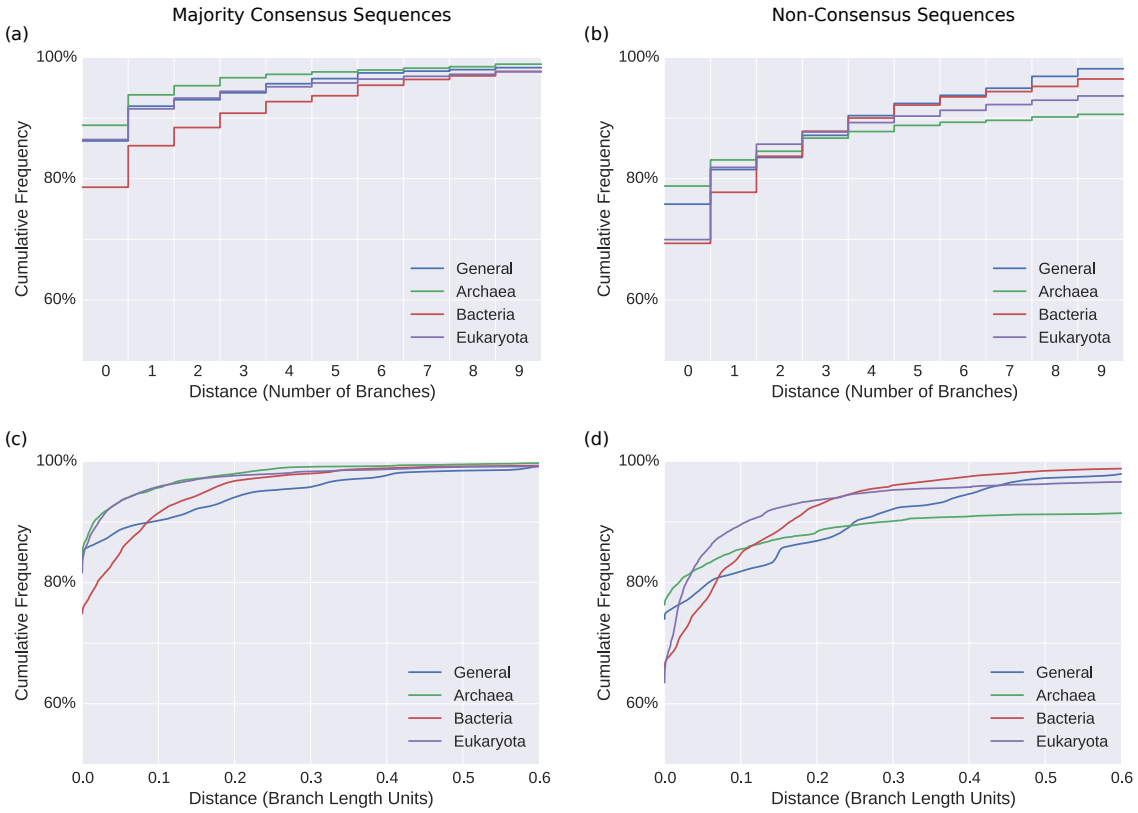

Figure 3.8: Effect of using actual sequences (instead of consensus sequences) on placement accuracy. Subfigures (a) and (c) show the evaluation of the majority rule consensus sequences, and are identical to Figure 3.6(a) and Figure 3.6(c), respectively. They are included here for ease of comparison. However, the y-axis is scaled to fit the remaining subfigures. Subfigures (b) and (d) show the evaluation of the approach of using actual (non-consensus) sequences from the database (instead of consensus sequences), as explained in the text. The top row (Subfigures (a) and (b)) shows distances in number of branches away from the expected placement branch; the bottom row (Subfigures (c) and (d)) shows distances in branch length units. All trees used for this evaluation are unconstrained.

We found that this approach yields trees that are less accurate for phylogenetic placement. The resulting accuracy is worse in all cases. That is, on average, the sequences were placed further away from their respective expected branch. We suspect that this is (i) because single sequences do not capture the diversity of their clade as well as consensus sequences, and (ii) because they do not incorporate as much biological information (e. g., in form of ambiguity characters). We conclude

that using consensus sequences to represent clades in our PhAT approach is more accurate and hence the preferrable approach.

Further Aspects and Observations

In the above evaluations, we generally found that using a constraint when inferring the tree only slightly affects the placement accuracy. However, when considering only the distance of the most likely placement (highest LWR) to its correct edge instead of using average distances weighted by the LWR per QS, the constrained trees consistently yield better results (data not shown). In other words, the most likely placement is more often on the correct branch of the constrained trees. For example, the most pronounced change is observed for the *Eukaryota* tree, with 84% correct placements for the unconstrained tree, and 89% for the constrained one. We suspect that this is an artifact of our evaluation process, as we consider a sequence to be correctly placed if the placement branch belongs to the consensus sequence to which the sequence contributed. As the selection of sequences for each consensus sequence is guided by the taxonomy, using the same taxonomy as constraint for the tree thus might also improve the placement accuracy.

As a final remark, we implicitly assumed the taxonomic label of each sequence to be correct. That is, in the evaluations, we measured the accuracy of the placements using the taxonomic labels of the sequences in Silva as an indicator of the expected branch of each sequence. However, errors are expected due to incongruity between the taxonomy and the phylogeny [264], as well as due to taxonomically mislabeled sequences [191]. For example, Sativa [191], found 9934 mislabeled sequences in the Silva database. Furthermore, 17 452 sequences contain one of "incertae", "unclassified" or "unknown" in their taxonomic name, indicating that those sequences might not be reliable. In total, there are 25 910 (or 4.3%) such dubious sequences in version 123.1 of the Silva database. Not all sequences should hence be expected to be placed on their expected branches. We therefore evaluated how these dubious sequences affect the accuracy of the trees. To this end, we used the same four trees as used in the main part of the evaluation (i. e., they were constructed with all sequences, including the dubious sequences), but for the evaluation step itself excluded the dubious sequences. That is, those dubious sequences were not placed on the trees, and their distance to the expected branch was not used for the evaluation. In most cases, this improved the results slightly, but not by much (data not shown). This shows that our trees are robust to such potentially erroneous sequences. Therefore, we decided to only report the unfiltered results in the above evaluations.

3.3.3 Empirical Datasets

PhATs are intended for conducting phylogenetic placement of environmental sequences. As such sequences are anonymous and their true evolutionary history thus unknown, we can not repeat the previous accuracy tests on empirical environmental datasets. Instead, we assess if the PhATs yield meaningful quantitative results for typical post-analysis methods. To this end, we placed two already well-studied empirical metagenomic amplicon barcoding datasets (see Appendix B for their details)

on our unconstrained *Bacteria* tree. To assess the placement results obtained from the PhAT, we performed Squash Clustering and Edge PCA [239] post-analyses (see Section 2.5.5) on the placement results.

Bacterial Vaginosis Dataset

We used an empirical sequence dataset of the vaginal microbiome of 220 women with a total of 426 612 sequences [339] for this evaluation. For details on the dataset and its processing, see Appendix B.1. The original study showed associations between the presence of certain bacterial species and the diagnosis of Bacterial Vaginosis (BV), a condition caused by changes in the vaginal microbiome. In the study, the Nugent score [274] was used as a clinical diagnostic criterion for BV. This score ranges from 0 (healthy) to 10 (severe illness). We placed the sequences of the dataset on their original tree and on our unconstrained *Bacteria* tree, and reproduce some of the results from the original study to assess differences induced by using distinct references trees. The results reveal that the PhAT reproduces certain aspects of the original study based on custom RTs with manually selected reference sequences, at least to the extent that is expected from its phylogenetic resolution.

First, we conducted Squash Clustering [239] (see Section 2.5.5) of the samples placed on the two trees. The resulting hierarchical cluster trees of the samples are shown in Figure 3.9.

The general features of the two cluster trees are comparable, indicating that our tree is able to distinguish between healthy and sick patients. However, there is a major difference in the lower half of the trees: While Figure 3.9(b) shows some small branch lengths and even a separated sub-clade of samples with low Nugent score, these branches have a length of virtually zero in Figure 3.9(a). As shown in Srinivasan et al. (2012) [339], the healthy patients are divided into two classes, based on the presence of two species of *Lactobacillus*. The original reference tree contains sequences of those species, and can thus distinguish between them. Our broad *Bacteria* tree however does not have this degree of species-level resolution and thus treats them the same, yielding a negligible KR distance (Section 2.5.4) between the samples. Although this finding is expected, it serves as an example for the limits of our method.

Second, we conducted an Edge PCA [239] (see Section 2.5.5) of the samples. The resulting scatter plots are shown in Figure 3.10.

The scatter plots show that the PhAT is able to separate samples by their Nugent score, that is, to classify them into healthy (left, blue items) and sick patients (right, red items). However, as with Squash Clustering, samples that only differ in placements at the species level are not separated from each other in the Edge PCA plot of Figure 3.10(a). Hence, the two classes of healthy patients do again not exhibit the separation based on the two *Lactobacillus* species that is apparent when using the original reference tree as in Figure 3.10(b). Thus, the samples with low Nugent score form one blob in Figure 3.10(a).

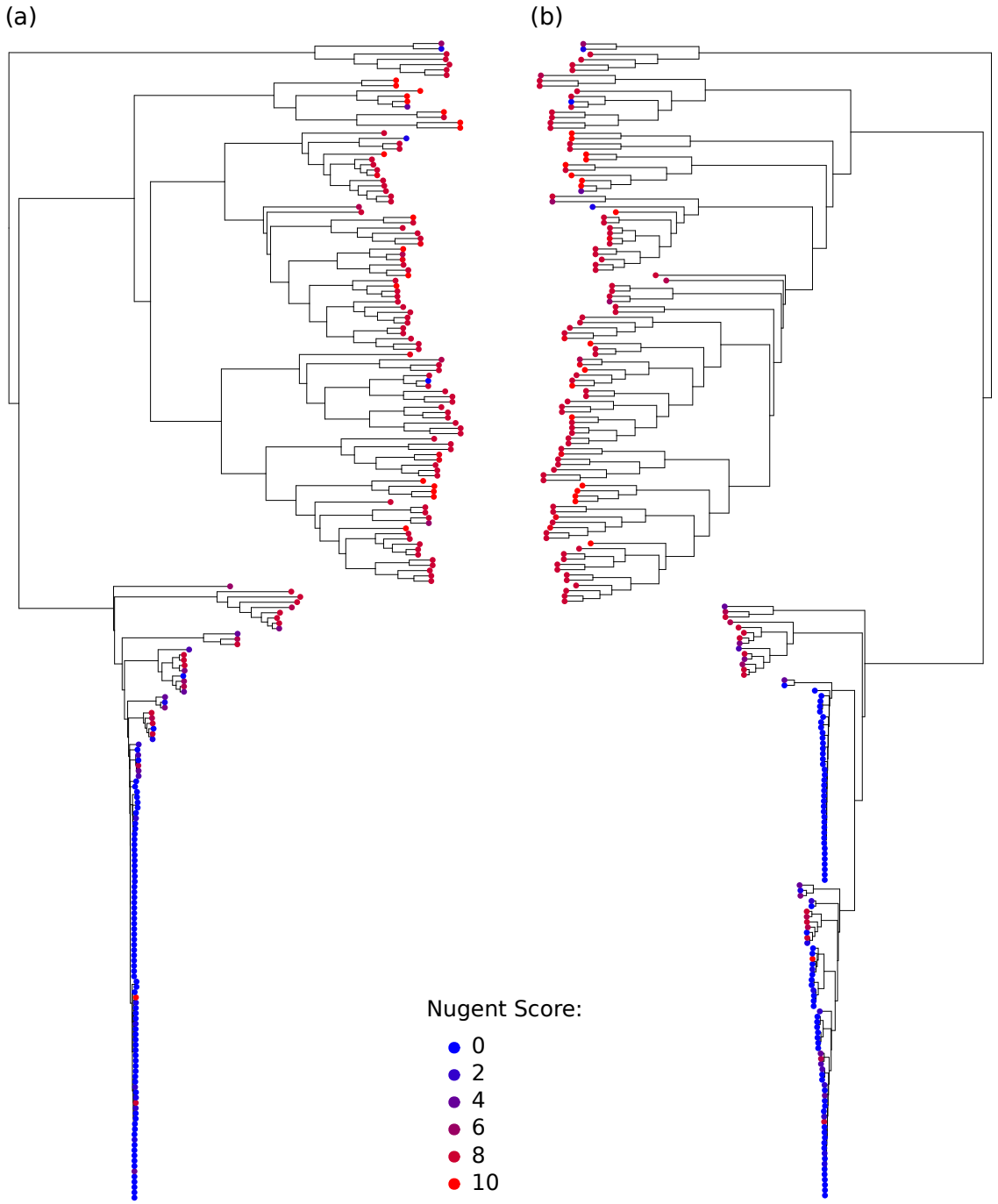

Figure 3.9: Assessment of a PhAT for conducting Squash Clustering. The figure compares the hierarchical clustering trees resulting from a Squash Clustering analysis (see Section 2.5.5) using (a) our unconstrained *Bacteria* PhAT and (b) the original reference tree of Srinivasan et al. (2012) [339]. Subfigure (b) is a recalculation of Figure 1(A) of Srinivasan et al. (2012) [339], and has been horizontally flipped for ease of comparison. The tips of both clustering trees correspond to samples, which are colored by the respective per-sample Nugent score, where high values indicate women with severe Bacterial Vaginosis.

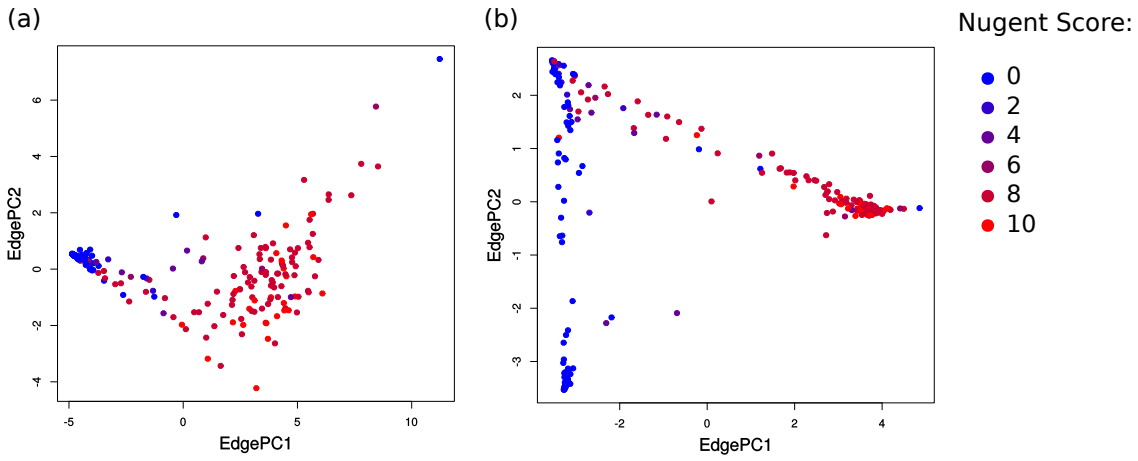

Figure 3.10: Assessment of a PhAT for conducting Edge PCA. The figure compares the scatter plots resulting from an Edge PCA (see Section 2.5.5) using (a) our unconstrained *Bacteria* PhAT and (b) the original reference tree of Srinivasan et al. (2012) [339]. Subfigure (b) is a recalculation of Figure 3(A) of Srinivasan et al. (2012) [339]. The items represent samples, which are colored by the respective Nugent score of each sample, where higher values indicate women with severe Bacterial Vaginosis.

This limitation can be overcome in two ways: On the one hand, one can use a PhAT with finer taxonomic resolution, that is, with more taxa that resolve down to species level. On the other hand, our multilevel placement approach (see Section 3.3.5) can be used with a refined second level tree that, for example, contains species sequences of the relevant *Lactobacillus* clades.

We however generally note that similar issues of deficient resolution or missing species can potentially also arise when hand-selecting reference sequences, and are thus not an inherent disadvantage of our method. In the end, it is the responsibility of the researcher to ensure that the selected reference sequences are suitable for the dataset to be placed.

Human Microbiome Project Dataset

Next, we tested the unconstrained *Bacteria* tree generated by our PhAT method for placing and analyzing a large sequence dataset. For this, we used the Human Microbiome Project (HMP) [160, 250] data, and selected 9192 samples from different body sites with a total of 117 million sequences. For details on the dataset and its processing, see Appendix B.3. The sequences were placed on the tree, and subsequently analyzed with two different methods, as shown in Figure 3.11.

First, we computed the pairwise KR distance matrix (see Section 2.5.4) between all samples. This high-dimensional matrix was then embedded into the plot by performing Multidimensional Scaling (MDS). MDS [105, 196, 233] is a dimensionality reduction technique that finds an embedding of a distance matrix into lower

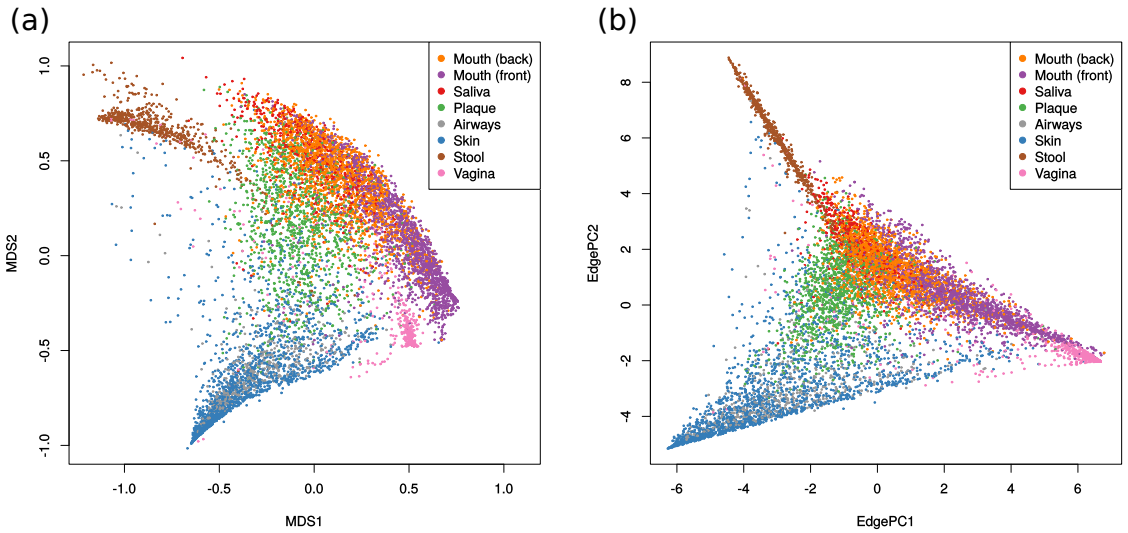

Figure 3.11: Assessment of a PhAT for large dataset analyses. Sequences from the HMP dataset [160, 250] were placed on our unconstrained *Bacteria* tree, and analyzed with two analysis methods. In subfigure (a), we visualized the pairwise KR distances between all samples, using a two-dimensional Multidimensional Scaling (MDS). In subfigure (b), we performed Edge PCA (see Section 2.5.5) on the samples. For both plots, we categorized the 18 original body site labels into 8 groups, in order to yield the plot more readable. See Table B.2 for the mapping between the original labels and the groups used here.

dimensions (in this case, 2 dimensions), while, at the same time, preserving higher dimensional distances as well as possible. Second, we again performed Edge PCA [239] (see Section 2.5.5) on the samples.

Both subfigures show that the tree, despite only representing higher taxonomic levels, suffices to separate different body site regions from each other. Even different oral regions are mostly separated, although there is quite some overlap. We hence conclude that our PhAT is capable of analyzing such datasets and that it yields useful results.

3.3.4 Taxonomic Assignment and Profiling

Here, we assess how PhATs perform when used for obtaining a taxonomic profile of a set of samples in conjunction with phylogenetic placement. Taxonomic profiling is the prediction of the taxonomic identities and their relative abundances of sequences from samples obtained via shotgun sequencing (Section 2.2.2); it does not result in taxonomic assignments for individual sequences (although this can be an intermediate step), but yields a summary of the abundances of different taxa in the samples [253]. We emphasize though that taxonomic assignment and profiling are neither the focus of PhATs, nor the intended standard use case of phylogenetic placement.

To perform the evaluation, we conducted parts of the CAMI Challenge [322], which is a community-driven effort to assess taxonomic profiling methods using a common set of benchmark datasets. To assess the feasibility of using trees generated with PhAT to obtain taxonomic profiles of microbiome data, we utilized the *mouse gut* dataset of the 2nd CAMI Challenge [36]. See Appendix B.4 for details on our processing of this dataset. In short, we phylogenetically placed the reads of the 16S region of the dataset on our unconstrained and constrained *Bacteria* trees. We then used this placement data to taxonomically label the reads based on the underlying Silva taxonomy of the trees, in analogy to the method used by Sativa [191].

Unfortunately, the CAMI Challenge requires taxonomic assignments that conform with the NCBI taxonomy [21, 315]. As our reference tree is however based on the Silva taxonomy [395], we thus had to compute a mapping between the two taxonomies. To this end, we developed a dedicated mapping procedure to, in a best effort approach, map our results to NCBI taxonomic names and IDs. The mapping is based on the *loose mapping* procedure of Balvočiūtė and Huson (2017) [17]. More specifically, we tried to map taxonomic paths to their name, rank, and ID in the NCBI taxonomy, if we find a name-based match between the two. When this fails, the phylogenetic placement mass assigned to a taxonomic path by our approach is instead added to the last successful mapping further up in the taxonomic hierarchy. By initiating this procedure for each taxonomic path from its root downwards, we ensure that all placement masses are taken into account.

This mapping is a major disadvantage of our approach when using a Silva-based reference, as the Silva and NCBI taxonomies are far from congruent [17]. Also note that our reference tree is limited to the 16S rDNA region. This substantially reduces the volume of data we can evaluate; in this particular test, only ≈0.08% of the total mouse gut data was identified as belonging to the 16S region (see also Appendix B.4). This means that our taxonomic profiling only uses a small fraction of the available data.

The resulting per-read assignment was then used to generate a taxonomic profile of the data. We used the CAMI evaluation tool for taxonomic profilers OPAL [253, 322] to compare our approach to competing software on the "gold standard" result for the dataset. OPAL assesses the performance of a set of tools relative to each other. We compared our approach to the profilers that were tested in the original publications [253, 322], which are: CommonKmers (corresponding to MetaPalette 1.0.0) [187, 188], CAMIARKQuikr 1.0.0 [186] (which is a combination of Quikr [189], ARK [190], and SEK [53], and abbreviated here as Quikr), TIPP 2.0.0 [272], MetaPhlAn 2.2.0 [323], MetaPhyler 1.25 [214], mOTU 1.1 [349], and FOCUS 0.31 adapted for CAMI [329].

We here show the most important OPAL results: Figure 3.12 compares the tools based on different error metrics; Table 3.4 shows the scores and ranks of our approach compared to the other CAMI participants. Details on the error metrics and their interpretation are explained in the OPAL publication [253].

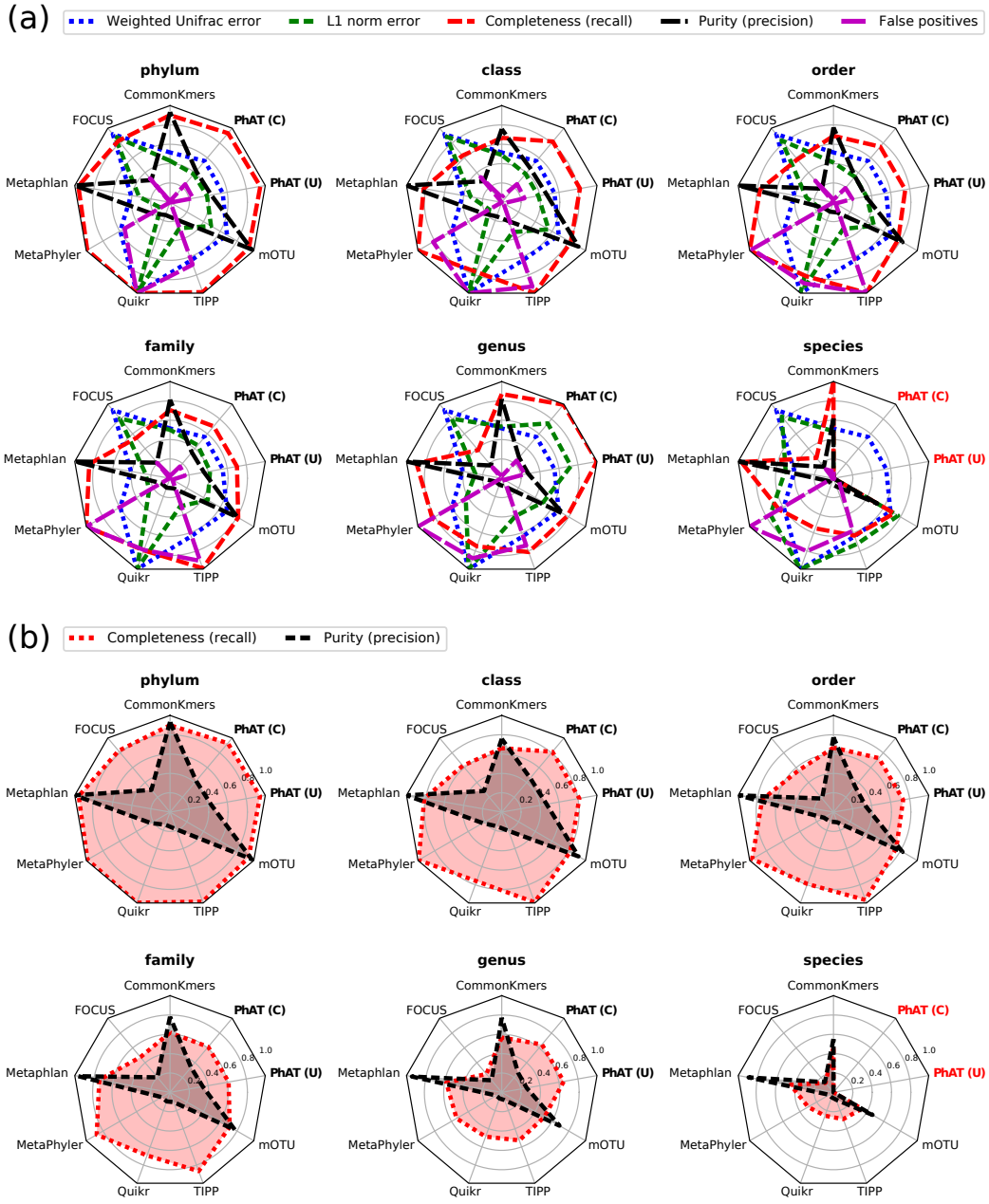

Figure 3.12: CAMI profiling results. The figure compares the taxonomic profiling conducted with our *Bacteria* trees to the tools of the 2nd CAMI challenge [36, 322], and was created with the OPAL tool [253]. The unconstrained and constrained tree are abbreviated here as "PhAT (U)" and "PhAT (C)", respectively. Subfigure (a) shows the *relative* performance of the tools across taxonomic ranks using the CAMI error metrics: Weighted Unifrac error, L1 norm error, recall (completeness), precision (purity) and false positives. Subfigure (b) shows the *absolute* recall (completeness) and precision (purity) for each tool across the taxonomic ranks. In both subfigures, the red text for our PhAT evaluations indicates that no predictions at the corresponding taxonomic rank were returned. This is because our SILVA-based tree does not have *species* resolution and does hence not allow for taxonomic profiling at this level.

Table 3.4: CAMI scores and ranks. The table shows the scores and ranks of different tools evaluated with data from, and following the protocol of, the 2nd CAMI challenge [36, 322]. Here, we compare the taxonomic assignment and profiling based on our PhATs to the tools that took part in the 2nd CAMI challenge. For this, we used the unconstrained and constrained *Bacteria* tree, which are abbreviated in the table as "PhAT (U)" and "PhAT (C)", respectively.
Four metrics are used in CAMI for evaluating tools: Recall (completeness), precision (purity), L1 norm error (abbreviated here as L1 NE), and Weighted Unifrac Error (abbreviated here as WUE). For each metric, the comparative scores of the tools are shown, as well as their rankings, relative to each other. Also, the total sum of scores and the total rank are shown, which add up the values of the four metrics. The procedure of the scoring and ranking is explained in detail in the Online Supplement of Sczyrba et al. (2017) [322].
Despite the caveats and limitations that are explained in the text, using our PhATs trees to obtain a taxonomic profile yields rankings in the middle of the field for all metrics, even when comparing them to tools dedicated to the purpose of taxonomic profiling.

	Total		Recall		Precision		L1 NE		WUE	
Tool	Sum	Rank	Score	Rank	Score	Rank	Score	Rank	Score	Rank
MetaPhlAn	1814	1	958	3	75	1	731	2	50	1
MetaPhyler	2995	2	314	1	2119	7	322	1	240	5
CommonKmers	3333	3	1448	6	409	2	1318	7	158	3
mOTU	3751	4	1703	8	488	3	1260	5	300	6
PhAT (C)	3784	5	1202	4	1116	4	1280	6	186	4
PhAT (U)	3933	6	1436	5	1153	5	1208	4	136	2
TIPP	4263	7	892	2	2126	8	930	3	315	7
FOCUS	6153	8	2079	9	1636	6	2018	8	420	8
Quikr	6838	9	1488	7	2398	9	2453	9	499	9

The resolution of the assignment is limited by the taxonomy used when running the PhAT method, that is, we could not assign reads at *Species* level. Furthermore, as mentioned, we were only able to use a small fraction of the reads (16S) and had to use incongruent taxonomies. Despite all these additional layers of complexity and potential error sources, we find that the performance of our approach is in the mid-range of the tools evaluated by CAMI. Note that this is a comparison to dedicated tools for taxonomic profiling, which also typically can assign more of the available reads. Therefore, our method yields reasonable accuracy for taxonomic assignment and profiling.

3.3.5 Subclades and Multilevel Placement

We selected five bacterial clades to evaluate PhAT accuracy on smaller clades, as well as to assess some properties of the Multilevel Placement approach. The same clades were already scrutinized in SATIVA [191]. Figure 3.13 shows the unconstrained *Bacteria* tree from the previous evaluations, with the branches of these five test clades highlighted.

Subclade Accuracy

First, using the SILVA sequences and sub-taxonomies of these five clades as input, we built unconstrained and constrained PhATs. We then conducted the same accuracy analysis as explained before on the resulting ten trees. That is, we placed the SILVA sequences of the five clades onto their respective PhATs and evaluated the distances to expected "true" branches. Thereby, we evaluated the accuracy of these PhATs when used as second level Clade Trees. The results are shown in Figure 3.14.

The placement accuracy is slightly worse for the sub-clade trees than for the eight comprehensive PhATs evaluated before (see Section 3.3.2), which can be seen by comparison to Figure 3.6. On average, 73.4% of the sequences were placed exactly on their expected branch, dominated by *Proteobacteria* and *Firmicutes*, which combined make up 75% of the sequences in the five clades, and have an accuracy of 71%. The *Actinobacteria* have the highest accuracy, with 82% of their sequences placed on the expected branch.

There are however also differences between the clades. The two smallest clades, *Actinobacteria* and *Cyanobacteria*, exhibit the shortest distances in branch length units. In fact, the longest distance of any sequence from its expected branch in the *Cyanobacteria* clade is around 0.4, which is indicated by the end of the red line in the lower two plots of Figure 3.14. On the other hand, the *Firmicutes* generally have the lowest accuracy. In Figure 3.13, which shows the unconstrained *Bacteria* tree, the *Firmicutes* clade exhibits many paraphyletic clades, which is a known issue [281]. This indicates that there is a high incongruence between the *Firmicutes* taxonomy and phylogeny in SILVA, which might explain why the *Firmicutes* score worst in Figure 3.14.

These results are likely due to the inability of 16S SSU sequences to properly resolve lower taxonomic levels [165, 255, 290]. For example, Table 2 of Janda and Abbott

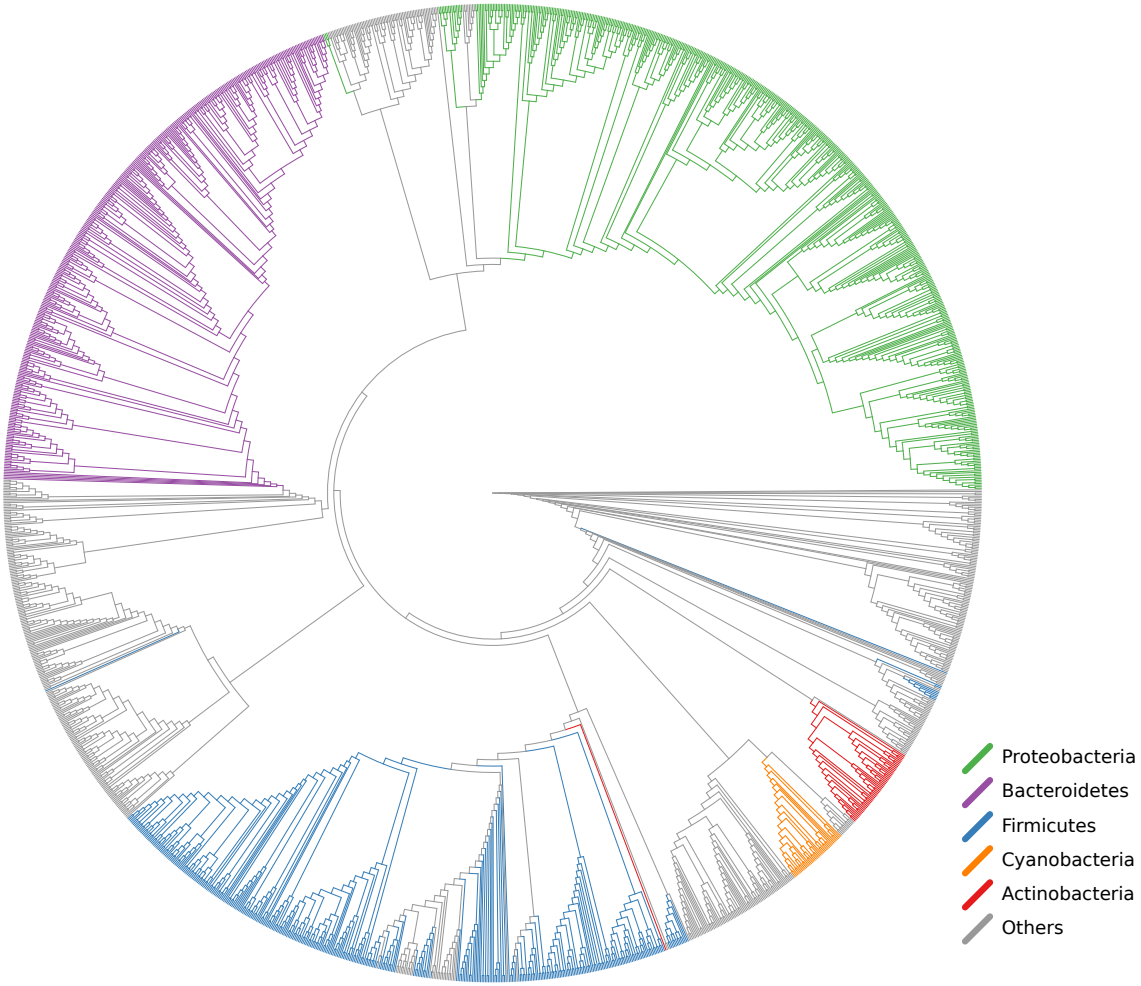

Figure 3.13: Unconstrained *Bacteria* tree with five bacterial sub-clades. This tree is the result of our PhAT method applied to the *Bacteria* sequences in Silva. The tree contains a total of 1914 taxa. Colorized are the five *Phylum* level sub-clades that we used for testing multilevel placement: *Proteobacteria* (505 taxa), *Bacteroidetes* (362 taxa), *Firmicutes* (360 taxa), *Cyanobacteria* (39 taxa), and *Actinobacteria* (53 taxa). The incongruence between taxonomy and phylogeny is visualized here as non-monophyletic colored branches. We define a clade to consist of all branches that are part of a monophyletic split of the tree with respect to the taxa in the clade. In other words, all branches on one side of a split are considered to belong to a clade, if that side of the split only contains taxa belonging to that clade. These branches then receive the same color here. Then, for multilevel placement, a sequence is considered to be part of a clade if its most probable placement falls into that clade. For example, a sequence that is placed onto one of the orange branches on this tree is subsequently placed in the *Cyanobacteria* tree for the second level placement. Each of the five sub-clades is represented by multiple branches here. We call this the "overlap" with the *Bacteria* tree.

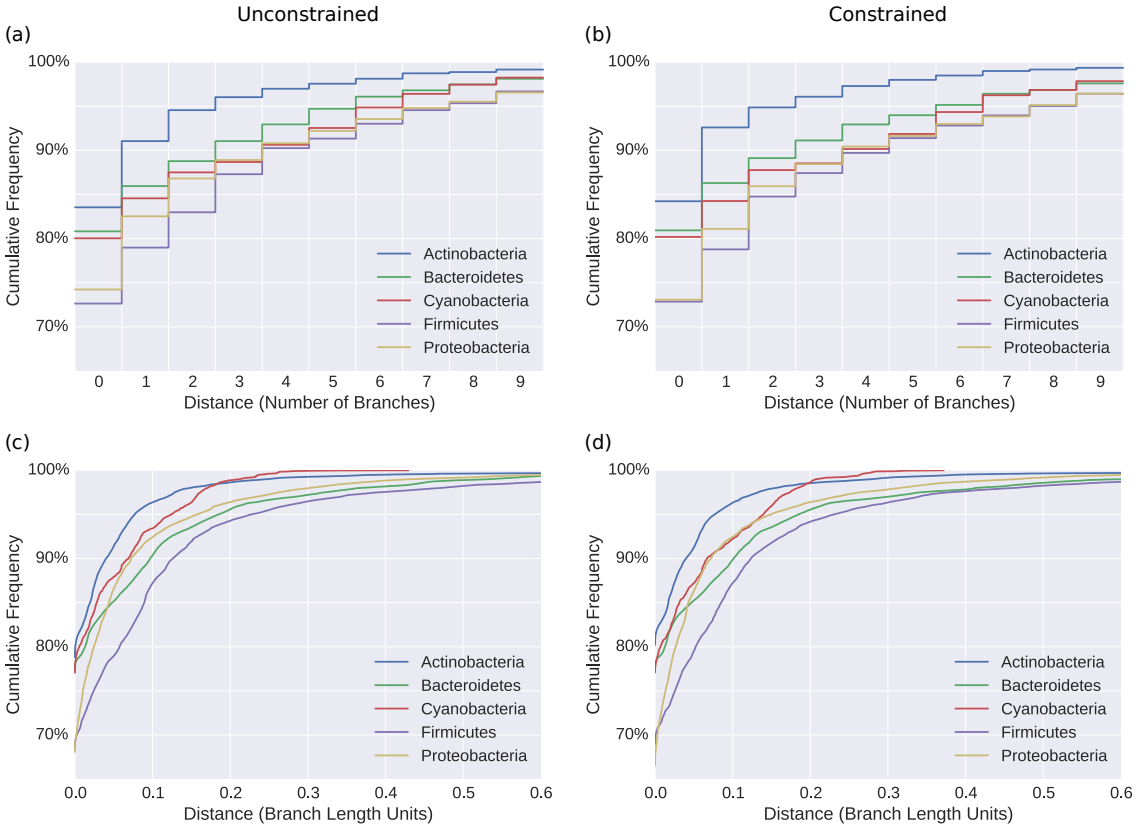

Figure 3.14: Accuracy of the PhATs of five bacterial sub-clades. We used five sub-clades of the *Bacteria* in SILVA, which were already scrutinized in Kozlov et al. (2016) [191], to asses the placement accuracy of our PhATs for less diverse sets of sequences. These five clades are also highlighted in Figure 3.13; see there for a description of the clades. The evaluation was conducted as explained in the text, using the same accuracy measurement as before (see Section 3.3.2). In short, we placed the SILVA sequences of the clades on their respective tree, and measured how far each of them is away from the branch of the consensus sequence it is represented by.
The top row (Subfigures (a) and (b)) shows discrete distances in number of branches; the bottom row (Subfigures (c) and (d)) shows continuous distances in branch length units. The left side shows the accuracy of the unconstrained trees, the right side shows the accuracy for trees constrained by the SILVA taxonomy.

(2007) [165] lists 10 bacterial genera that are known to be hard to identify using 16S sequences. These genera account for 7.9% of the 2846 taxa that are represented by the five bacterial trees tested here. Furthermore, 95 553 of the 450 313 sequences that were placed on those trees (21.2%) belong to one of these genera. This might explain the worse scores for these clade trees. Lastly, the consensus sequences at the tips of the trees represent the *Genus* level. This induces short branches, which increases the probability of misplacements.

First Level Accuracy

Next, using the five clades, we evaluated the accuracy of the first placement level when conducting Multilevel Placement (as introduced in Section 3.2.2). So far, our evaluation focused on the distance from a sequence placement to its expected placement branch. For the first placement level on a Backbone Tree (BT), it is however more important that a sequence is placed into the correct clade, while the exact placement branch is mostly irrelevant. A sequence that is placed in the correct clade of the first level tree can subsequently be placed on the correct second level Clade Tree (CT). Thus, we used the unconstrained *Bacteria* BT again, and assessed how many sequences were placed in the clades shown in Figure 3.13. Of the 450 313 sequences in SILVA in these clades, 98.0% were placed (most likely placement) into a branch of their corresponding clade. Thus, for multilevel placement, they will be assigned to the correct second level CT. More specifically, the *Firmicutes* perform worst, as only 94.7% of the *Firmicute* sequences are placed into the corresponding correct clade. This can be explained by the high amount of paraphyly of this clade, as mentioned above, which is a known issue [281]. The sequences of the other four clades we tested achieve a clade identification accuracy exceeding 99%. This shows that a high overlap of the clades with with the BT yields high accuracy. In other words, second level clade trees should be represented by multiple branches in the backbone tree.

As already mentioned in Section 3.2.2, a high-level taxonomic constraint can improve the accuracy of placing a sequence into the correct BT clade. To show this, we inferred the *Bacteria* RT again, but used a *Phylum* level constraint that separates the five clades from each other and from the rest of the tree. All branches within the clades were resolved using maximum likelihood. The tree (not shown) is similar to the tree in Figure 3.13, but all five clades are now monophyletic. Using this tree, 99.3% of the sequences were placed into the correct clade. Particularly the accuracy for *Firmicutes* improved, yielding an accuracy of 99.5%.

Overall, our experiments show that the first level placement is highly accurate, even if an extremely diverse "all bacteria" Backbone Tree is used. The accuracy on the second level is slightly worse when using PhATs as CTs.

3.4 Summary and Outlook

We presented methods and algorithms to facilitate and accelerate phylogenetic placement of large environmental sequencing studies. The Phylogenetic Automatic (Ref-

erence) Tree (PhAT) method (Section 3.2.1) provides a means for automatically obtaining suitable reference trees by using the taxonomy of large sequence databases. Using the SILVA database as a test case, we showed that it can be applied for accurately (pre-)placing environmental sequences into taxonomic clades. In combination with our multilevel placement approach (Section 3.3.5), even very broad PhATs achieve high accuracy, particularly when using high-level clade constraints. The method can also be used for rapid data exploration in environmental sequencing studies: A PhAT might be useful to obtain an overview of the taxa that are necessary to capture the diversity of a sequence dataset, without the substantial human effort and potential bias of manually selecting reference sequences. As we showed, PhATs can also be used to obtain taxonomic assignments and profiles for a set of samples, in conjunction with phylogenetic placement (Section 3.3.4). To capture clade diversity with finer resolution, for example for a second placement level, clade-specific PhATs can be inferred. If species-level resolution is required, we recommend that the sequences are inspected by an expert, in order to confirm that the tree is appropriate for the data to be placed on it. Furthermore, as our automated approach inevitably suffers from errors in the database it is based on, we recommend using SATIVA [191] to identify potentially mislabeled sequences in the respective database. One should also keep in mind that phylogenetic placement does not necessarily provide resolution at the *species* level [91].

As we show, our multilevel placement method (Section 3.2.2) as well as the preprocessing pipeline (Section 3.2.3) accelerate the placement process without sacrificing accuracy. By first placing the query sequences on a broad Backbone Tree (BT), as described in the method, novel environments with sequences of unknown evolutionary origin can be classified without having to process a large tree comprising all taxa of interest. The method hence offers the benefits of high resolution reference trees, without the aforementioned limitations induced by large alignments. A second placement level on a set of Clade Trees (CTs) provides sufficient taxonomic resolution for biological interpretation. Placement accuracy can be further improved by inferring the BT with a high-level constraint that separates the clades of the CTs from each other and thus ensures monophyly of these clades. Furthermore, for the practical applicability and relevance of this approach, we refer to our results presented in Mahé et al. (2017) [230].

Apart from exploring sequence data from unknown environments, we see online services as a potential additional application of our methods. A web service that offers phylogenetic placement of user-submitted sequences is confronted with two issues: Firstly, the potentially large number of query sequences, and secondly, their unknown provenance. Both can be solved by using a broad "all-of-life" backbone tree for pre-classification, and subsequently distributing the second-level placement to different compute nodes.

4. Visualization

This chapter is derived from parts of the peer-reviewed open-access publication:

> **Lucas Czech** and Alexandros Stamatakis. "Scalable Methods for Analyzing and Visualizing Phylogenetic Placement of Metagenomic Samples." *PLOS ONE*, 2019, Volume 14, Issue 5, Page e0217050.

All text and figures in this chapter were created by Lucas Czech.

4.1 Background and Motivation

When analyzing a set of metagenomic sequence samples (Section 2.2.2) using phylogenetic placement (Section 2.5), a first step is often to visualize the data. For small samples, it is possible to mark individual placement locations on the Reference Tree (RT), as offered for example by iTOL [205], or even to create a tree where the most probable placement per Query Sequence (QS) is attached as a new branch, as implemented in the guppy tool from the pplacer suite [241], RAxML-EPA [25, 342], and our tool gappa [70]. For larger samples, one can alternatively display the per-edge placement mass, either by adjusting the line widths of the edges according to their mass, or by using a color scale, as offered in ggtree [397], guppy, and gappa. Using per-edge colors corresponds to binning all placements of an edge into one bin (see Section 2.5.3). For large datasets, the per-edge masses can vary by several orders of magnitude. In such cases, it is often preferable to use a logarithmic scaling, as shown in Mahé et al. (2017) [230]. In addition to visualizing each sample separately, the average mass distribution (after *squashing* the masses, see Section 2.5.3) visually summarizes a set of samples.

The visualizations provide an overview of the species abundances over the tree. They can be regarded as a more detailed version of abundance charts [160], which

are typically shown in the form of pie or bar plots [108, 203, 230]. For instance, Figure 4.1 shows a comparison of a simple abundances pie chart compared to a visualization of per-branch abundances on a reference tree for the same dataset. Although there exist more advanced variants such as hierarchical pie charts, for instance as offered by the KRONA tool [276], it should be apparent that the tree visualization provides more detailed information. In Figure 4.1(b), we combined the information of the pie chart with the tree visualization by displaying the abundances next to each clade.

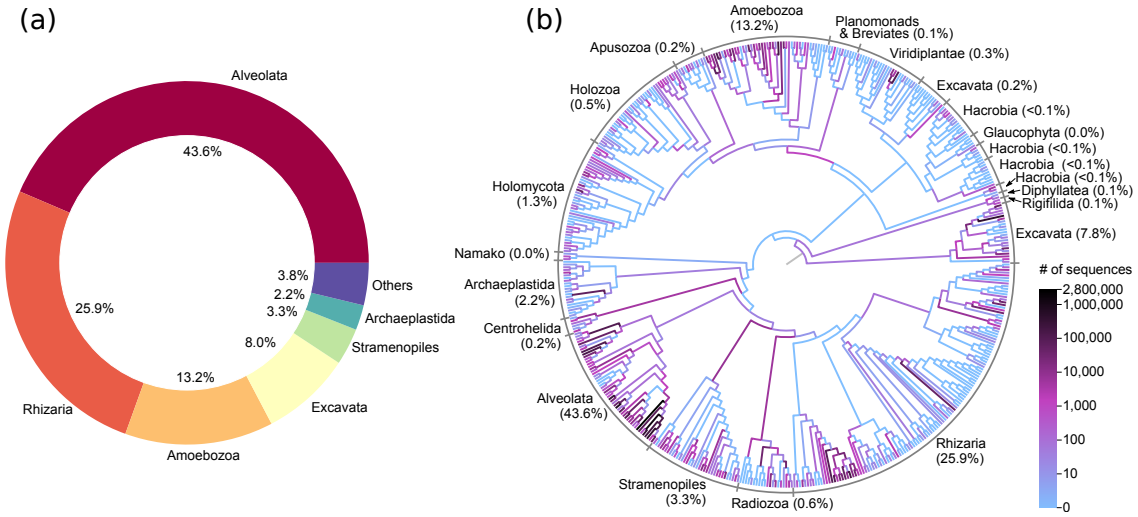

Figure 4.1: Visualizations of sequence abundances. The figure shows an example of (a) a typical pie chart of taxonomic abundances and (b) the substantially more informative per-branch mass visualization using phylogenetic placement on a reference tree. The data is from Mahé et al. (2017) [230]. The branches of the tree are colored by abundances on a logarithmic scale, and clades of the tree are annotated with the per-clade abundances, effectively combining the information of the pie chart with the tree visualization.

A more detailed visualization of the abundances per taxon and per sample can be obtained via a heat map that shows the per-sample abundances at the tips of the tree. This is often used for OTU trees, where tips correspond to the OTUs that the tree was inferred from. This can be extended to placements by also showing abundances/masses at the inner edges, as shown in Figure 4.2. The figure shows the placements per edge and per sample for the BV dataset; see Section B.1 for details on this dataset.

On the left hand side of Figure 4.2, note the two particularly dark branches, *Lactobacillus iners* and *Lactobacillus crispatus*, which are the major species associated with a healthy vaginal microbiome [339]. This can also be seen on the right hand side of the figure, where the high abundance of *Lactobacillus* in healthy patients is visible as red stripes in the lower part of the matrix, while the diseased patients exhibit high placement masses at several other taxa [339].

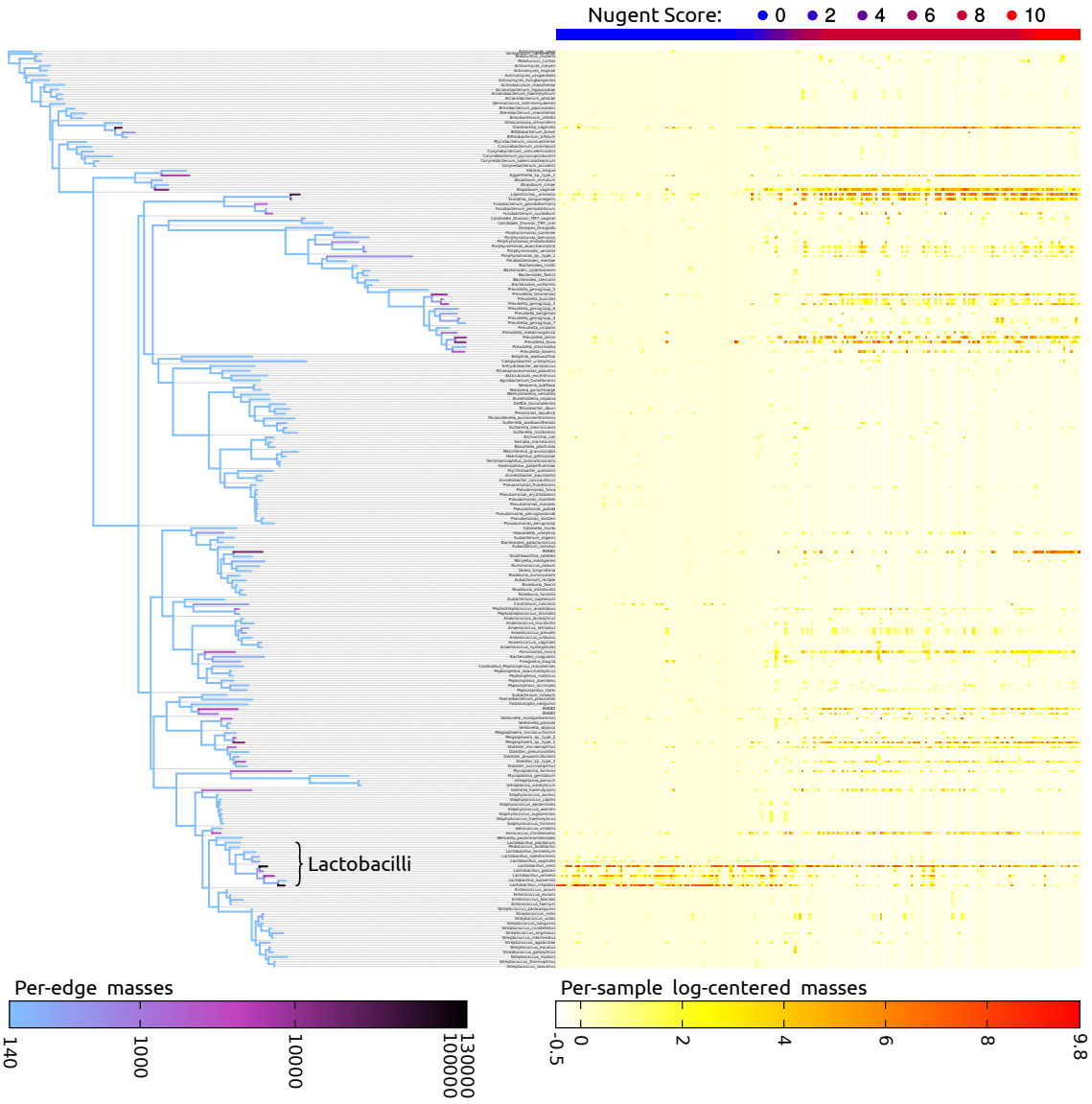

Figure 4.2: Visualization of per-edge and per-sample masses of the BV dataset. The figure provides an overview of the placement of the BV dataset: The left hand side shows a condensed version of the original reference tree of Srinivasan et al. (2012) [339], colored by log-scaled placement mass of all samples accumulated. For clarity and simplicity, in this figure we used a reference tree built from the consensus sequences of each original reference taxon, so that each species is represented by exactly one tip here. The *Lactobacillus* clade is highlighted in the tree, which is an important clade for this dataset.
The right hand side shows a heat map that further resolves the placement masses per sample: Each row corresponds to a branch of the tree on the left (note that dashed lines also start from inner branches), and each column represents one sample. The values are log-centered in order to be consistent with typical OTU abundance heat map representations [376]. The samples/columns are sorted by their Nugent score, from 0 at the left for the healthy patients to 10 at the right for the sick ones. The Nugent score of each sample is also shown at the top as a blue to red bar.

Such visualizations directly depict the placement masses on the tree. When visualizing the accumulated masses of multiple samples at once, it is important to chose an appropriate normalization strategy for the task at hand, as explained in Section 2.5.3. For example, if samples represent different locations, one might prefer to use normalized masses, as comparing relative abundances is common for this type of data. On the other hand, if samples from the same location are combined (e. g., from different points in time, or different size fractions), it might be preferable to use absolute abundances instead, so that the total number of sequences per sample can be visualized.

When placing OTUs (see Section 2.5.3), or ignoring sequence abundances, the resulting visualizations can be interpreted as a depiction of species diversity. Moreover, these visualizations can be used to assess the quality of the RT. For example, placements into inner branches of the RT may indicate that appropriate reference sequences (i) have either not been included or (ii) are simply not available yet. This complements the sequence filtering that relies on so-called backbone trees as previously described in Section 3.2.2.

These visualizations are useful tools for initial dataset and feature exploration in terms of species abundances. However, when working with multiple samples, they do not immediately reveal relative differences between samples that might hint at underlying biological or ecological properties of the samples or their environment.

4.2 Methods and Implementation

Here, we introduce visualization methods for phylogenetic placement of a set of metagenomic sequence samples that highlight (1) regions of the tree with a high variance in their placement distribution (called *Edge Dispersion*), and (2) regions with a high correlation to meta-data features (called *Edge Correlation*).

Both methods take as input a set of samples, each consisting of a set of Query Sequence (QS) placed on a fixed Reference Tree (RT). They then use the edge masses matrix and the edge imbalances matrix as introduced in Section 2.5.3 to calculate per-branch quantities, which are subsequently visualized on the RT.

4.2.1 Edge Dispersion

The Edge Dispersion is derived from the edge masses or edge imbalances matrix (Section 2.5.3) by calculating a measure of dispersion for each matrix column, for example, the standard deviation σ. Because each column corresponds to an edge, this information can be mapped back to the tree, and visualized, for instance, via color coding. This allows to examine which edges exhibit a high heterogeneity of placement masses across samples, and hence indicates which edges discriminate samples.

As the abundances of different species, and hence also the edge mass values, can span many orders of magnitude, it might be necessary to scale the variance logarithmically. Often, one is more interested in the branches with high placement mass,

as they comprise the most abundant species in the samples. In these cases, using the standard deviation or variance is appropriate, as they also indicate the mean per-edge mass. On the other hand, by calculating the per-edge Index of Dispersion [105], that is, the variance-mean-ratio σ^2/μ, differences on edges with little mass also become visible. Note that this is a valid operation, as edge masses are non-negative count variables. The Index of Dispersion is useful to explore heterogeneity on edges with low species abundances.

As Edge Dispersion relates placement masses from different samples to each other, the choice of the normalization strategy *is* important (see also Section 2.5.3). When using normalized masses, the magnitude of dispersion values needs to be cautiously interpreted [220]. The Edge Dispersion can also be calculated for edge imbalances in form of the standard deviation. As edge imbalances are usually normalized to $[-1.0, 1.0]$, their dispersion can be visualized directly without any further normalization steps. However, because imbalances can be negative, the Index of Dispersion is not applicable to them. An example for an Edge Dispersion visualization is shown in Figure 4.3(a), and discussed in Section 4.3.

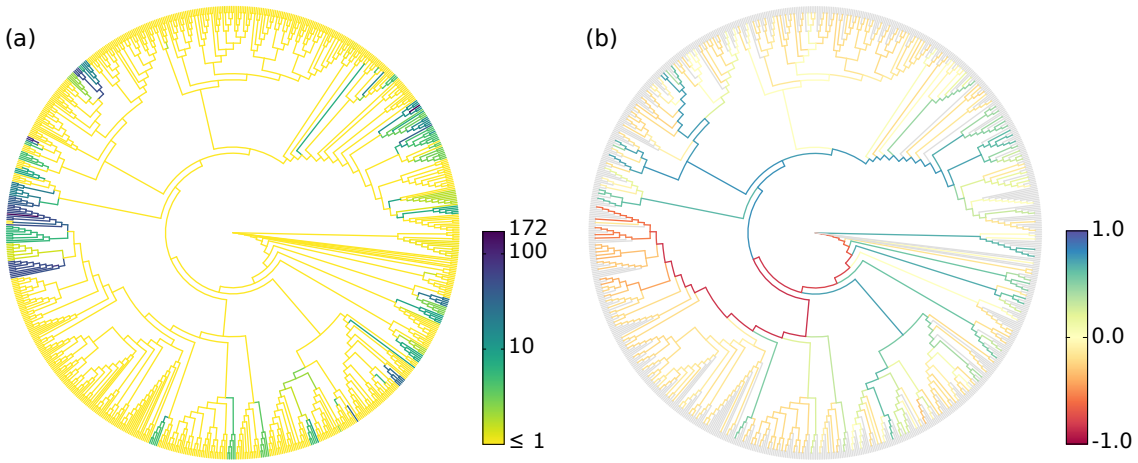

Figure 4.3: Examples of Edge Dispersion and Edge Correlation. We applied our novel visualization methods to the Bacterial Vaginosis (BV) dataset (see Appendix B.1 for details) to compare them to the existing data analysis methods. (a) Edge Dispersion, measured as the standard deviation of the edge masses across samples, logarithmically scaled. (b) Edge Correlation, in form of Spearman's Rank Correlation Coefficient between the edge imbalances and the Nugent score. Tip edges are gray, because they do not have a meaningful imbalance. This example also shows the characteristics of edge masses and edge imbalances: The former highlights individual edges, the latter highlights paths to clades.

4.2.2 Edge Correlation

In addition to the per-edge masses, the Edge Correlation can further take a specific meta-data feature into account, that is, a column of the meta-data matrix. The

Edge Correlation is calculated as the correlation between each edge column and the feature column, for example by using the Pearson Correlation Coefficient or Spearman's Rank Correlation Coefficient [105]. This yields a per-edge correlation of the placement masses or imbalances with the specific meta-data feature, and can again be visualized by color coding the edges.

The Pearson Correlation Coefficient r between the per-edge mass or imbalance column c (with average $\bar{c}$) and the meta-data column m (with average $\overline{m}$) for a set of s samples is calculated as

$$r = \frac{\sum_{i=1}^{s}(c_i - \bar{c})(m_i - \overline{m})}{\sqrt{\sum_{i=1}^{s}(c_i - \bar{c})^2}\,\sqrt{\sum_{i=1}^{s}(m_i - \overline{m})^2}} \tag{4.1}$$

Spearman's Rank Correlation Coefficient is calculated in the same way, but instead of using the actual values of the matrix columns, their ranking numbers are used. That is, the values are reduced to ordinal numbers, where the lowest value is transformed into 1, the second lowest value into 2, etc. As a consequence, instead of a linear correlation, this coefficient calculates the strength of monotonic correlations.

Both variants are inexpensive to calculate and hence scale well to large datasets. As typical correlation coefficients are within $[-1.0, 1.0]$, there is again no need for further normalization. This yields a tree where edges or clades with either a high linear or high monotonic correlation with the selected meta-data feature are highlighted. Figure 4.3(b) shows an exemplary visualization of this method.

In contrast to Edge PCA [239] that can use meta-data features to annotate samples in its scatter plots (see Section 2.5.5), our Edge Correlation method directly displays the influence of a feature on the branches or clades of the tree. It can thus, for example, help to identify and visualize dependencies between species abundances and environmental factors such as temperature or nutrient levels. Again, the choice of the normalization strategy is important to draw meaningful conclusions. However, the correlation is *not* calculated between samples or sequence abundances. Hence, even when using normalized samples, the pitfalls induced by correlations of compositional data [220] do not apply here.

The method further bears some conceptual similarity to Phylofactorization [376], for which we present adaptation to phylogenetic placements in Chapter 7. Phylofactorization also takes meta-data features into account and can hence identify relationships between changes in environmental variables and changes in abundances in clades of the tree. It typically uses linear regression in form of a Generalized Linear Model (GLM) to assess these relationships. Note that the correlation coefficient used in our Edge Correlation can be interpreted as the slope of the regression line of the standardized values, which establishes a connection between Edge Correlation and Phylofactorization. The advantage of using correlations here instead of a GLM lies in its simplicity for the interpretation of results: We are here interested in whether changes in a meta-data feature lead to an increase or decrease of abundances on branches or in clades of the tree. Using a GLM for this purpose would not yield any advantage.

4.3 Evaluation and Results

4.3.1 BV Dataset

We re-analyzed the Bacterial Vaginosis (BV) dataset (see Appendix B.1 for details) by inferring a tree from the original reference sequence set and conducting phylogenetic placement of the 220 samples. The characteristics of this dataset were already explored by Srinivasan et al. (2012) [339] and Matsen and Evans (2011) [239]. We use it here to give exemplary interpretations of our Edge Dispersion and Edge Correlation methods, and to evaluate them in comparison to existing methods.

Figure 4.3 shows our novel visualizations of the BV dataset. Edge Dispersion is shown in Figure 4.3(a), while Figure 4.3(b) shows the Edge Correlation with the Nugent score. The Nugent score [274] is a clinical standard for the diagnosis of Bacterial Vaginosis, ranging from 0 (healthy) to 10 (severe illness). Bacterial Vaginosis (BV) is a disease of the vagina that manifests itself in form of an abnormal vaginal microbiome [339].

The connection between the Nugent score and the abundance of placements on particular edges was already explored by Matsen and Evans (2011) [239], but only visualized indirectly (i. e., not on the RT itself). For example, Figure 6 of the original study [239] plots the first two Edge PCA components colorized by the Nugent score. We recalculated this figure for comparison, and show it later in Figure 5.3(i).

In contrast to this, our Edge Correlation measure directly reveals the connection between the Nugent score and placements on the reference tree: The clade on the left hand side of the tree, to which the red and orange branches lead in Figure 4.3(b), are *Lactobacillus iners* and *Lactobacillus crispatus*, respectively, which were identified in Srinivasan et al. (2012) [339] to be associated with a healthy vaginal microbiome. Thus, their presence in a sample is anti-correlated with the Nugent score, which is lower for healthy individuals. The branches leading to this clade are therefore colored in red. On the other hand, there are several other clades that exhibit a positive correlation with the Nugent score, that is, were green and blue paths lead to in the figure, again a finding already reported in Srinivasan et al. (2012) [339].

Both trees in Figure 4.3 highlight the same parts of the tree: The dark branches with high deviation in Figure 4.3(a) represent clades attached to either highly correlated (blue) or anti-correlated (red) paths Figure 4.3(b). This indicates that edges that have a high dispersion also exhibit variation in placement mass between samples of different Nugent score. Both methods hence reveal the clades that are relevant for discriminating among the samples of this dataset.

We further compared our methods to the visualization of Edge PCA components on the reference tree. To this end, we recalculated Figures 4 and 5 of Matsen and Evans (2011) [239], and visualized them with our color scheme in Figure 4.4 for ease of comparison. The figures show the first two components of Edge PCA, mapped back to the branches of the RT. The first component, Figure 4.4(a), reveals that the *Lactobacillus* clade represents the axis with the highest heterogeneity across samples,

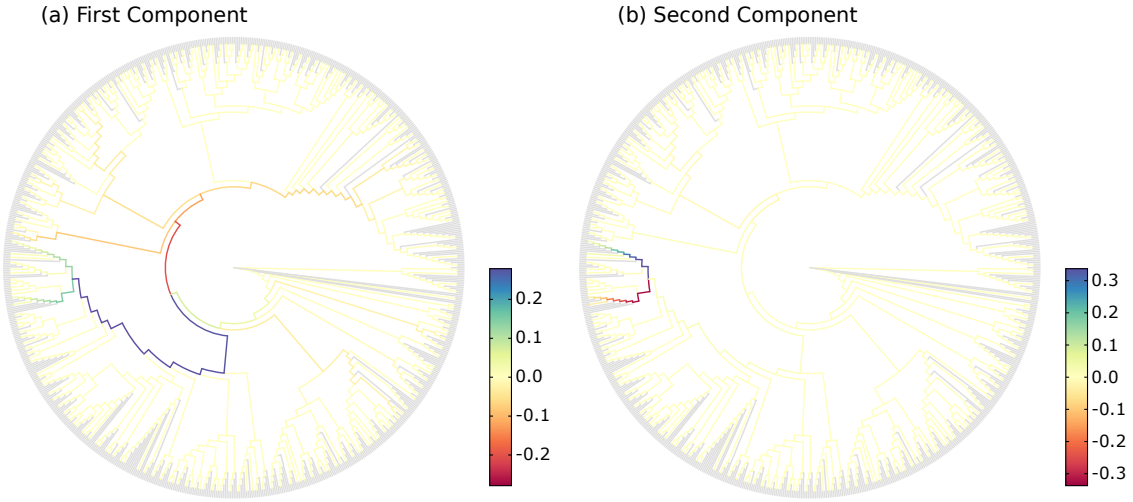

Figure 4.4: Recalculation of the Edge PCA tree visualization. Subfigures (a) and (b) are recalculations of Figures 4 and 5 of Matsen and Evans (2011) [239], respectively. However, we show them here in our coloring scheme in order to facilitate comparison with other figures. The original publication instead uses two colors for a positive and a negative sign of the principal components, and branch widths to show their magnitude. Note that the actual sign is arbitrary, as it is derived from principal components.
The figure shows the first two Edge PCA components, visualized on the reference tree. This form of visualization is useful to interpret results such as the Edge PCA projection plot as shown later in Figure 5.3(i). It reveals which edges are mainly responsible for separating the samples into the PCA dimensions. Here, the first principal component in (a) indicates that the main PCA axis separates samples based on the presence of placements in the *Lactobacillus* clade, which is what the blue and green path leads to. The second component in (b) then further distinguishes between two species in this clade, namely *Lactobacillus iners* and *Lactobacillus crispatus*.

while the second component, Figure 4.4(b), further distinguishes between the two aforementioned clades within the *Lactobacilli*. As shown in Figure 4.3(b), Edge Correlation also highlights the *Lactobacillus* clade, but does not distinguish further between its sub-clades. This is because a high Nugent score is associated with a high abundance of placements in either of the two relevant *Lactobacillus* clades.

Further examples of variants of Edge Dispersion on the BV dataset are shown in Figure 4.5. In Figure 4.5(a), which is linearily scaled, it is striking that one outlier edge, marked with an arrow, is dominating the values, and thereby hiding the values on less variable edges. This outlier occurs for the species *Prevotella bivia* in one of the 220 samples, where 2781 out of 2782 sequences in the sample have some placement mass on that branch. Upon close examination, this outlier can also be seen in Figure 1D of Srinivasan et al. (2012) [339], but is less apparent there. Thus, our novel visualization can help to detect such outlier samples. In Figure 4.5(b) and (c), we used logarithmic scaling instead, in order to reveal more details on the edges with

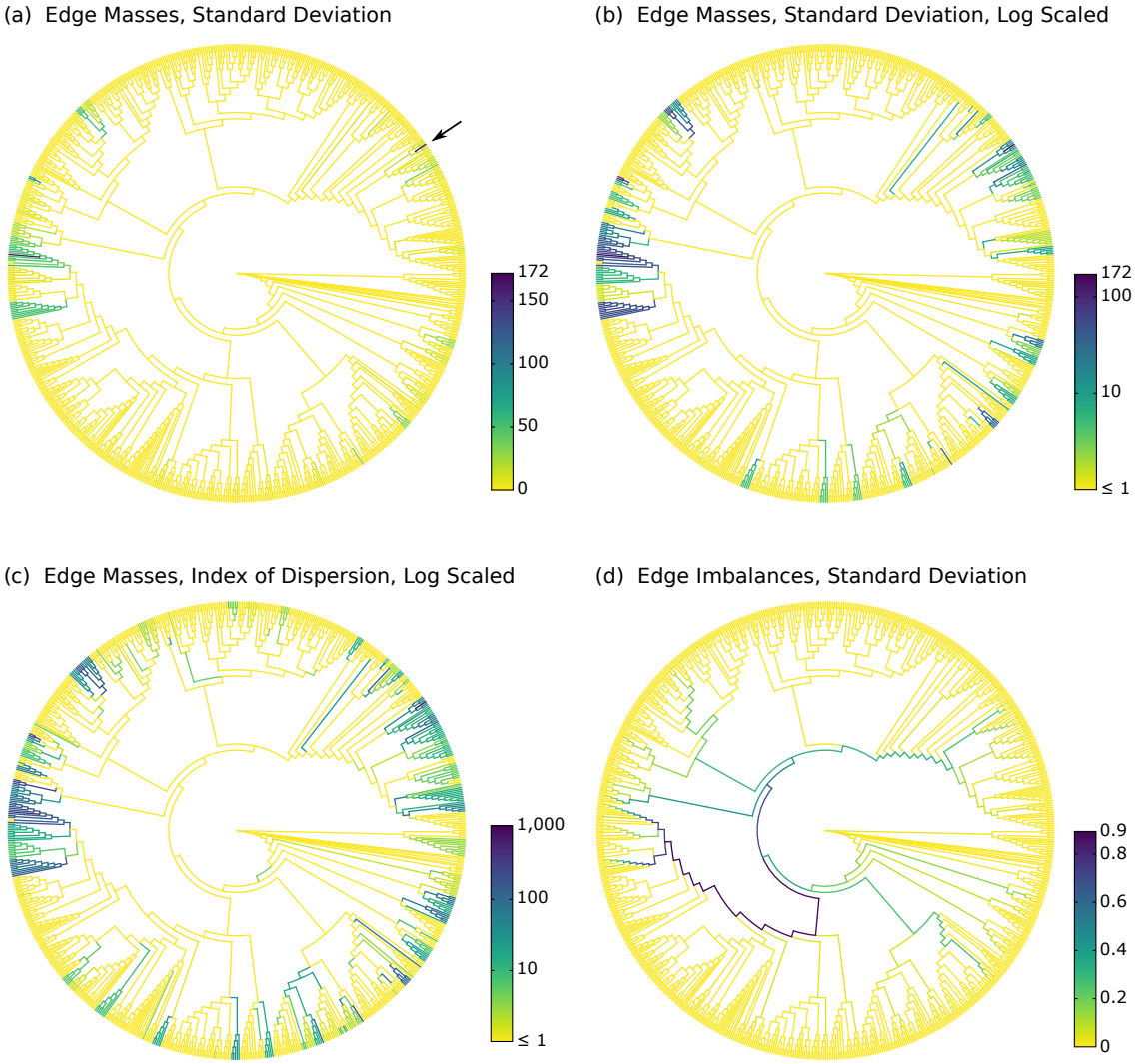

Figure 4.5: Examples of variants of Edge Dispersion. The figure shows further visualizations of Edge Dispersion on the BV dataset. All subfigures highlight the same branches and clades as found by other methods such as Edge PCA. Subfigure (a) shows the standard deviation of the absolute edge masses, without any further processing. Subfigure (b) is identical to Figure 4.3(a), for comparison, and shows the standard deviation again, but this time using logarithmic scaling, thereby revealing more details on the edges with lower placement mass variance. Subfigure (c) shows the Index of Dispersion of the edge masses, that is, the variance normalized by the mean. Hence, edges with a higher number of placements are also allowed to have a higher variance. The figure reveals more details on the edges with lower variance, highlighted in medium green colors. Subfigure (d) shows the standard deviation of edge imbalances. Because we used imbalances of unit mass samples, the values are already normalized. Note that imbalances can be negative; thus, the Index of Dispersion is not applicable to them.

lower placement mass variance. When comparing these two Figures to Figure 4.6, we see that the same clades that exhibit a high correlation or anti-correlation with meta-data there are also highlighted here. There are only few medium values, which indicates that there are two classes of edges: Those which have a high placement mass heterogeneity (and thus can help to distinguish patients), and those who have almost no placements at all. Lastly, Figure 4.5(d) shows the Edge Dispersion of the edge imbalances. The path to the *Lactobacillus* clade is again clearly visible, indicating that the placement mass in this clade has a high variance across samples.

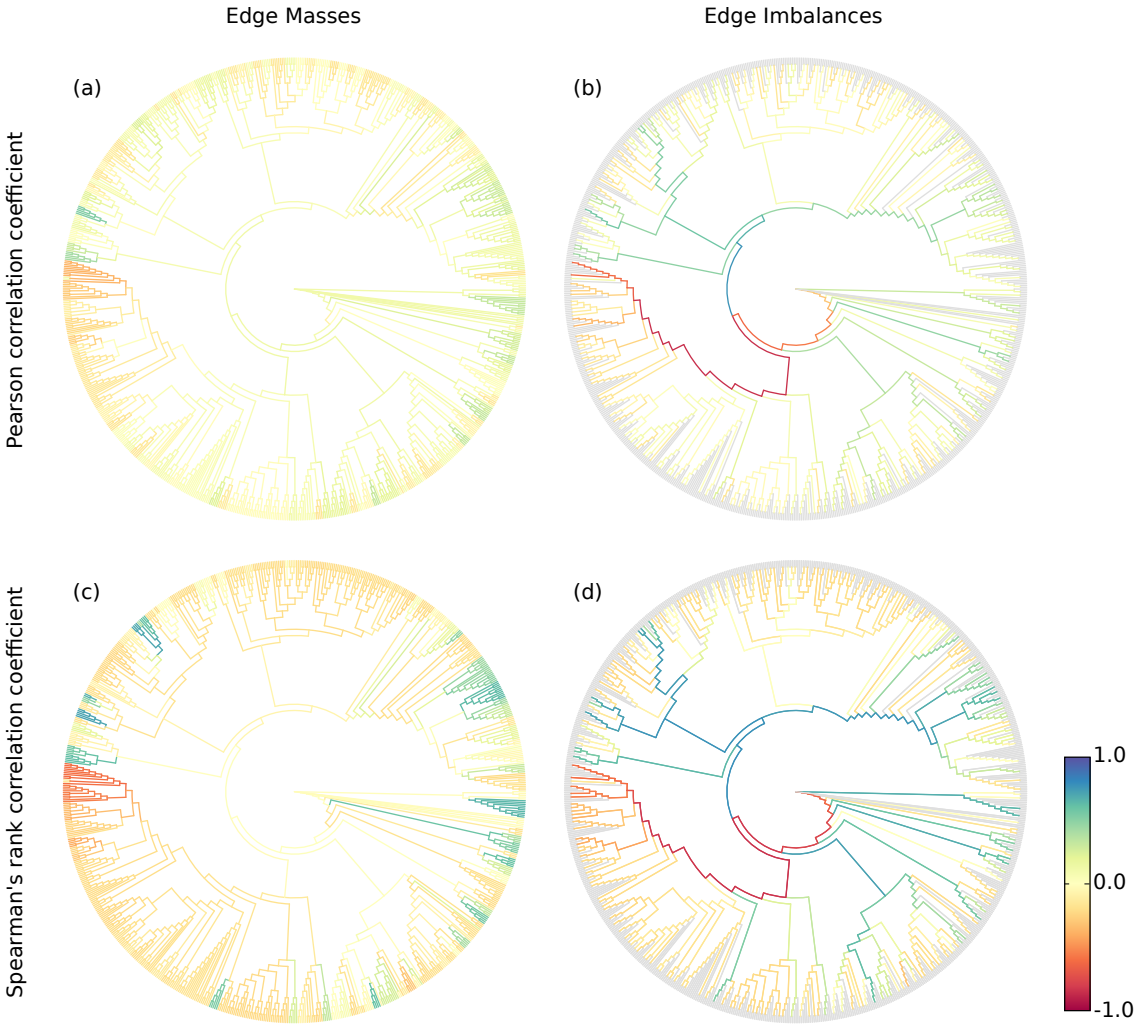

Figure 4.6: Examples of variants of Edge Correlation. The Figure shows the correlation of edge masses and imbalances with the Nugent score on the BV dataset. The Nugent score measures the severeness of Bacterial Vaginosis, and ranges from 0 for healthy subjects to 10 for heavily affected patients. Subfigures (a) and (b) use the Pearson Correlation Coefficient, that is, they show the linear correlation with the meta-data feature, while subfigures (c) and (d) use Spearman's Rank Correlation Coefficient, and thus show monotonic correlations. Subfigure (d) is identical to Figure 4.3(b), for comparison.

In Figure 4.6, we show further examples of variants of the Edge Correlation on the BV dataset. All subfigures show red edges or red paths at the *Lactobacillus* clade. This indicates that the presence of placements in this clade is anti-correlated with the Nugent score, which is consistent with the findings of Srinivasan et al. (2012) [339] and Matsen and Evans (2011) [239]. In other words, the presence of *Lactobacillus* correlates with a healthy vaginal microbiome. On the other hand, blue and green edges, which represent positive correlations, are indicative of edges that correlate to Bacterial Vaginosis. The extent of correlation is larger for Spearman's Coefficient, indicating that the correlation is monotonic, but not strictly linear.

Lastly, we conducted Edge Correlation visualizations using additional meta-data features that are available for the BV dataset, in order to further confirm the consistency of our methods with existing results. In particular, we visualize the correlation with Amsel's criteria [9] and the vaginal pH value in Figure 4.7, both of which were already used in Srinivasan et al. (2012) [339] as additional indicators of Bacterial Vaginosis. We again found similar correlations as for the Nugent score.

4.3.2 Tara Oceans Dataset

We analyzed the Tara Oceans (TO) dataset [137, 174, 350] to provide further exemplary use cases for our visualization methods; see Appendix B.2 for details on this dataset. To this end, we used the unconstrained *Eukaryota* RT with 2059 taxa as described in Section 3.3.1. The meta-data features of the TO dataset that best fit for our methods are the sensor values for chlorophyll, nitrate, and oxygen concentration, as well as the salinity and temperature of the water samples. Other available meta-data features such as longitude and latitude where each sample was taken are available for the dataset; in particular, latitude can be used as an indicator for species diversity [350]. As species diversity is however a concept that is distinct from species abundances (here represented as the placement masses per branch), we do not use the geographical coordinates of the samples here. The Edge Correlation of the 370 samples with the nitrate concentration, the salinity, the chlorophyll concentration, and the water temperature are shown in Figure 4.8.

We selected the *Diatoms* and the *Animals* as two exemplary clades for closer examination of the results. Diatoms are mainly photosynthetic, and thus depend on nitrates as key nutrients [222, 292]. This is clearly visible by the high correlation of the clade with the nitrate concentration in Figure 4.8(a). Furthermore, the diatoms exhibit a positive correlation with the chlorophyll concentration in Figure 4.8(c), which again is indicative of their photosynthetic behavior. On the other hand, they prefer environments with low salt concentrations, and thus show a high anti-correlation with the salt content in Figure 4.8(b). Salinity is a strong environmental factor which heavily affects community structures and species abundances [222], particularly diatoms [292].

The correlations in the animal clade are less pronounced. They exhibit a negative correlation with nitrate in Figure 4.8(a), as well as an increase in absolute abundance with higher temperatures in Figure 4.8(d). While these findings are not surprising, they show that the method is able to find meaningful relationships in the data.

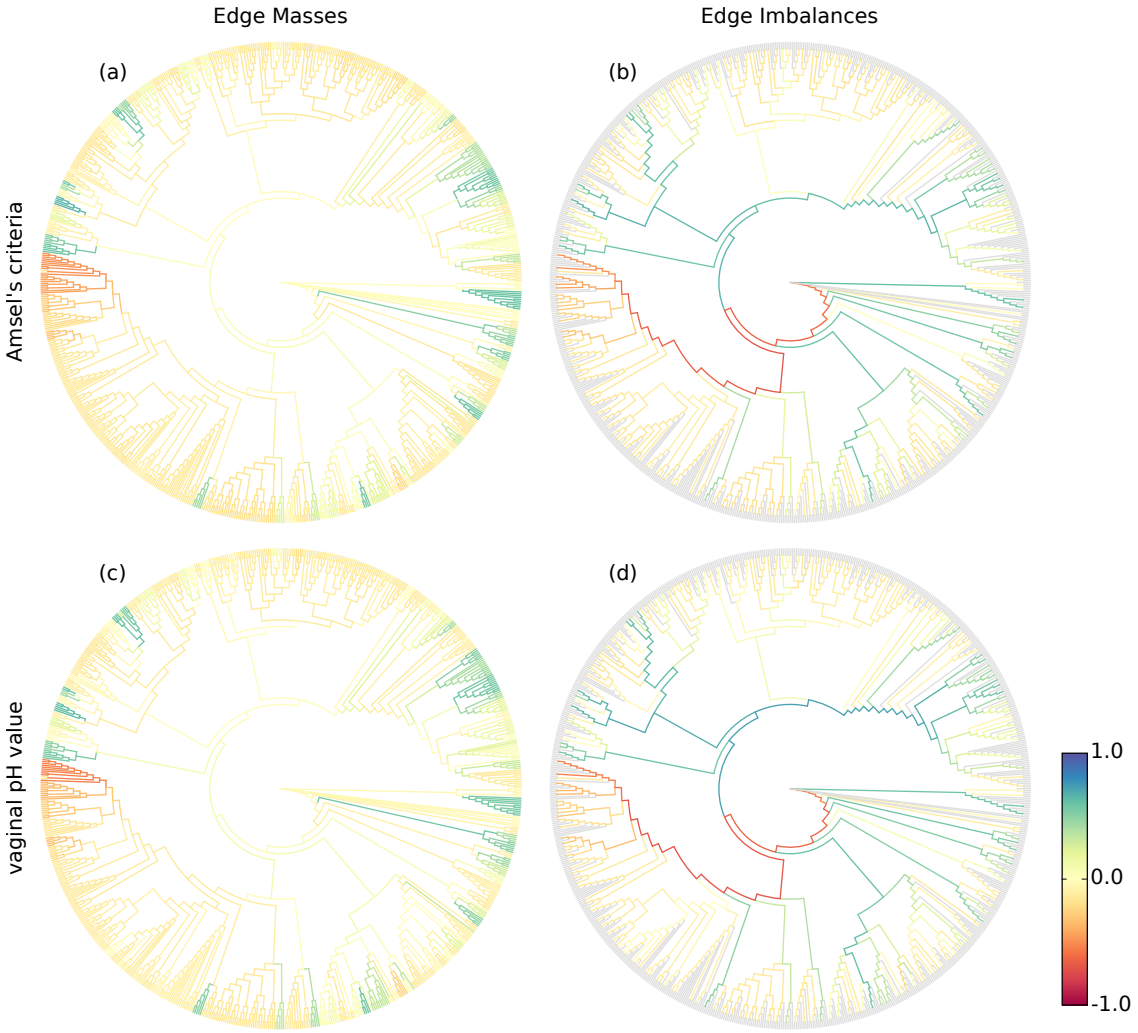

Figure 4.7: Edge Correlation with more meta-data features. Here, we use additional meta-data features of the BV dataset to show that Edge Correlation yields consistent results with existing methods. In particular, we calculated Spearman's Coefficient with Amsel's criteria [9] in Subfigures (a) and (b), as well as with the vaginal pH value in Subfigures (c) and (d). Both features were also used in Srinivasan et al. (2012) [339] as additional indicators of Bacterial Vaginosis. The figures are almost identical to the ones shown in Figure 4.6; that is, they yield results that are consistent with the previously used Nugent score, and that are also consistent with existing methods.

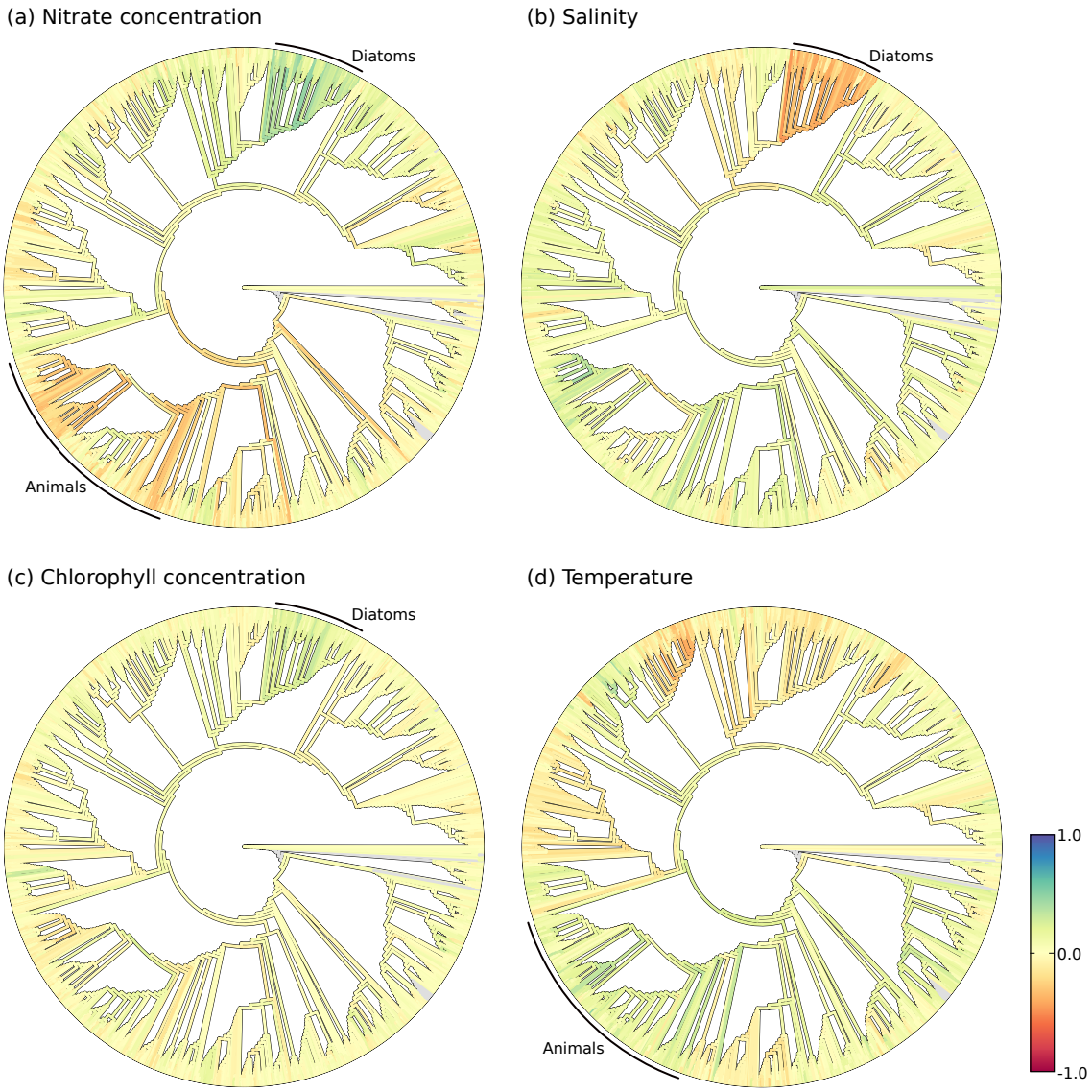

Figure 4.8: Examples of Edge Correlation using Tara Oceans samples. The figure shows the correlation of Tara Oceans sequence placements with (a) the nitrate, (b) the salinity, (c) the chlorophyll, and (d) the temperature sensor data of each sample. The sensor values range from -2.2 to $33.1\,\mu\text{mol/l}$ (nitrate), from 33.2 to 40.2 psu (salt), from -0.02 to $1.55\,\text{mg/m}^3$ (chlorophyll), and from -0.8 to $30.5\,^\circ\text{C}$ (temperature), respectively. The negative nitrate and chlorophyll concentrations are values below the detection limit of the measurement method (pers. comm. with L. Guidi on 2018-04-25), and hence simply denote low concentrations. We used Spearman's Rank Correlation Coefficient in all subfigures, and examine two exemplary clades, namely the *Animals* and the *Diatoms*, which are marked by arcs around the tree here.

These findings indicate that the Edge Correlation method is able to identify known relationships. It will therefore also be useful to investigate and discover insights of novel relationships between sequence abundances and environmental parameters.

4.3.3 Performance

Both methods (Edge Dispersion and Edge Correlation) are computationally inexpensive, as they only require a few operations per input matrix entry. They are thus applicable to large datasets. The calculation of the above visualizations took about 30 s each, which were mainly required for reading in the placement data. The required main memory for these computations is also relatively low, and mostly determined by the size of the input matrices, which contain $s \cdot b$ floating point numbers for a dataset of s samples placed on a tree with b branches.

Furthermore, in order to scale to large datasets, we reimplemented Edge PCA (Section 2.5.5), which was originally implemented as a command in the GUPPY program [241]. For the BV dataset with 220 samples (Appendix B.1), GUPPY required 9 min and used 2.2 GB of memory, while our implementation only required 33 s on a single core, using less than 600 MB of main memory. Furthermore, we tested our reimplementation of Edge PCA on the large Human Microbiome Project (HMP) dataset (see Appendix B.3 for details). For this dataset, GUPPY took 11 days and 75.1 GB memory, as it is only single-threaded and seems to use an inefficient parser for the `jplace` input format (Section 2.5.1), while our implementation needed 7.5 min on 16 cores and used 43.5 GB of memory.

4.4 Summary and Outlook

The chapter presented two novel methods to visually explore phylogenetic placement data in order to derive biological and ecological knowledge and unravel new patterns in the data. The methods complement existing analysis tools such as Edge PCA, and yield consistent results on known datasets.

Edge Dispersion is an exploratory tool that highlights branches of the phylogenetic tree which exhibit variations in the number of placements across samples. It thus allows to identify "interesting" regions of the tree with a high placement heterogeneity. In contrast to Edge Correlation, it can however not explain the reasons for the observed heterogeneity.

Edge Correlation additionally takes meta-data features into account, and identifies branches of the tree that correlate with quantitative features, such as the temperature or the pH value of the environmental samples. It thus can indicate those parts of the reference tree where changes in environmental variables drive changes in the abundances of species.

The methods are currently limited to correlations with singular continuous value meta-data features. In their current form, they hence do not allow for more challenging analyses, such as finding patterns and correlations that depend on multiple

features at once, or taking more complex data into account, such as the geographical distribution of samples.

In Chapter 6, we later present adaptations of recent tree-based concepts such as the *Phylogenetic ILR Transformation* and *Balances* [330] to phylogenetic placement data. These concepts can also be used in combination with Edge Correlation, which we briefly explore in Section 6.3.2. We further introduce a technique for weighting taxa/edges based on their "importance" for a dataset. This could also be adapted and used for Edge Dispersion and Edge Correlation, for instance as an alternative to "weighting" edges by the mean placement mass (as employed by the Index of Dispersion).

Furthermore, in Chapter 7, we describe how to use Generalized Linear Models (GLMs) in order to quantify relationships between meta-data variables and per-edge values such as placement masses. In particular, GLMs allow to take different types of meta-data variables (such as binary or categorical variables) into account, and also allow to consider multiple variables simultaneously. Quantifications of how well the model fits the data can be visualized per edge of the tree, as for instance shown later in Section 7.3.1, which can reveal relationships between multiple variables of different types with per-edge masses or imbalances. This poses an interesting and promising extension to the methods presented in this chapter. When using GLMs however, the direction of the relationship is not immediately visible (for instance, whether a variable is positively or negatively correlated with edge masses), and it is more challenging to assess which meta-data variables are more influential than others when using them simultaneously (for instance, assessing the strength of the correlation). While these issues can certainly be solved [122], the results might not be easily visualizable on a single tree. We hence leave a full exploration of these ideas as an extension to Edge Correlation as future work.

Lastly, in biogeographic and ecological studies, one might be interested in questions such as (i) how the regional diversity per area in a rain forest depends on distances between these regions [203], or (ii) how oceanic currents influence species diversity and distribution in the global oceans [350]. The integration of features such as geographical coordinates into our correlation analysis is however challenging. As mentioned above, a starting point could be to employ latitude as an indicator for species diversity [350]. Furthermore, geographical coordinates yield pairwise distances between samples, which could be used in more involved methods than the ones presented here to discover complex patterns in global ecological data. While it is unlikely that such questions can be answered via a single visualization, it might still be interesting and helpful to explore methods that utilize phylogenetic placement data to help answering them.

5. Clustering

This chapter is derived from parts of the peer-reviewed open-access publication:

> **Lucas Czech** and Alexandros Stamatakis. "Scalable Methods for Analyzing and Visualizing Phylogenetic Placement of Metagenomic Samples." *PLOS ONE*, 2019, Volume 14, Issue 5, Page e0217050.

All text, tables, and figures in this chapter were created by Lucas Czech.

5.1 Background and Motivation

Given a set of metagenomic sequence samples (Section 2.2.2), and a distance measure between them (Section 2.5.4), a fundamental task consists in clustering samples that are similar to each other. For example, Squash Clustering performs agglomerative hierarchical clustering of samples (Section 2.5.5). It is based on the phylogenetic placement of the Query Sequences (QSs) of the samples on a Reference Tree (RT), and employs the *Kantorovich-Rubinstein* (KR) distance [104, 239] to assess sample similarity. An example of the resulting clustering tree is shown in Figure 5.1(a).

For large datasets, producing a clustering tree can however considered to be a downside of Squash Clustering, as the number of tips in this tree is equal to the number n of samples that are being clustered. Thus, for datasets with more than a few hundred samples, the clustering result becomes hard to inspect and visually interpret.

Furthermore, depending on the data and research question at hand, the KR distance is not always the best measure of sample similarity. As explained in Section 2.5.3, edge imbalances can often reveal more subtle differences between samples than edge masses. Thus, for some datasets, it might make sense to use a distance that is based on edge imbalances instead.

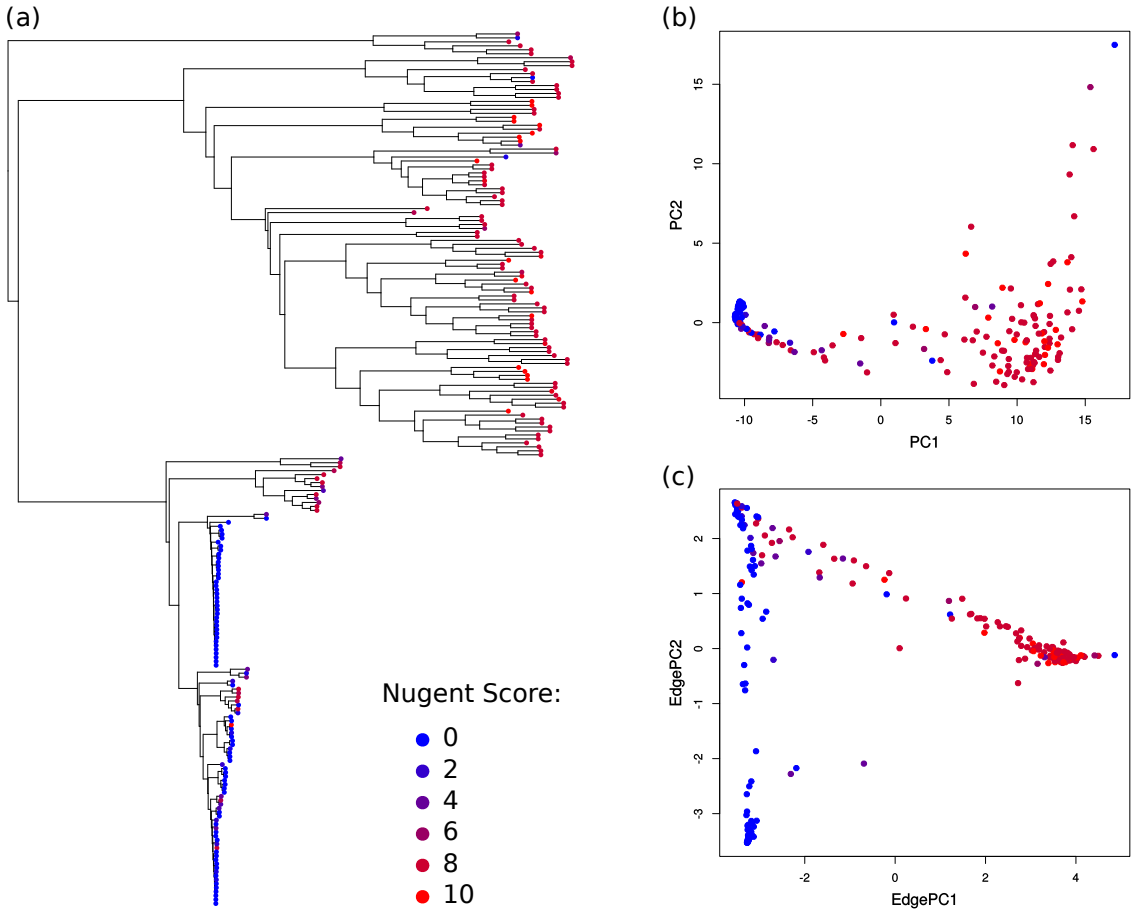

Figure 5.1: Existing analysis methods on the BV dataset. We applied (a) Squash Clustering, (b) PCA on the pairwise KR distance matrix between samples, and (c) Edge PCA, using the Bacterial Vaginosis (BV) dataset [339]. The subfigures are recalculations of Figure 1(A) of Srinivasan et al. (2012) [339], and Figures 4 and 3 of Matsen and Evans (2011) [240], respectively. All subfigures represent samples as colored dots according to their respective Nugent score, which indicates the severeness of the BV infection of the samples/patients. See Section 2.5.5 for a description of the methods, and Appendix B.1 for details on the dataset.

To further illustrate this, Figure 5.1(b) shows the result of a standard principal component analysis (PCA) [168, 286] on the pairwise KR distance matrix of the Bacterial Vaginosis (BV) dataset [339] that we used before. On the left hand side of the figure, the blue samples, representing healthy women with a low Nugent score, form a dense cluster. Towards the right hand side however, the red samples, which belong to sick patients, are spread over the rest of the graph. As Squash Clustering also uses the KR distance, the same pattern can be observed in its resulting clustering tree, as shown in Figure 5.1(a): The bottom half of the clustering tree, containing mainly healthy (blue) samples, has short branches, which correspond to the dense blue region in Figure 5.1(b). At the same time, the top half, mostly containing

samples from sick (red) patients, exhibits many long branches, corresponding to the scattered red region in Figure 5.1(b). Thus, Squash Clustering represents equivalent information to a standard PCA on this dataset. It thus "suffers" from the same shortcomings that Edge PCA is solving by using mass imbalance instead (see Section 2.5.5). This can be seen in Figure 5.1(c), which shows the result of Edge PCA on the dataset. There, the healthy (blue) samples clearly separate into two groups for the two dominant *Lactobacillus* clades in healthy patients, which is due to the edge imbalances that resolve smaller differences between the placements on nearby clades in this dataset. Hence, for this dataset, instead of using a distance that is based on edge masses such as the KR distance, edge imbalances should be used to measure distances between samples.

5.2 Methods and Implementation

We here propose two variants of k-means clustering [224, 344], which we call *Phylogenetic k-means*, and *Imbalance k-means*, respectively. They are clustering methods for phylogenetic placement of a set of metagenomic sequence samples, and address the issues described above. Note that these methods are clustering samples, and not single sequences [178]. Both methods produce a predefined number of clusters, and hence are able to work with arbitrarily large datasets. Phylogenetic k-means uses the KR distance to assess sample similarity, that is, it uses edge masses, while Imbalance k-means uses edge imbalances instead.

The methods take as input a set of n samples, each consisting of their Query Sequences (QSs) placed on a fixed Reference Tree (RT). They then assign the samples to k clusters, each represented by a cluster *centroid* that describes the average placement mass distribution of the samples assigned to it. We later also discuss how to chose a reasonable value for k.

5.2.1 Phylogenetic k-means

Phylogenetic k-means employs the KR distance (see Section 2.5.4) and hence yields results that are consistent with the clustering tree of Squash Clustering.

Algorithm

The input samples and the cluster centroids are of the same data type, namely, they are mass distributions on a fixed RT. It is thus possible to calculate the KR distances between samples and centroids, and to calculate their average mass distributions by *squashing*, as described in Section 2.5.3.

The objective is to then find an assignment of the n samples into k clusters that minimizes the total distance between each sample and the cluster centroid it is assigned to, measured as the KR distance between them. That is, for k clusters, each represented by a set A_k of samples assigned to it, and its centroid C_k, the objective is to find

$$\hat{A} \;=\; \arg\min_{A} \sum_{i=1}^{k} \sum_{s \in A_i} \mathrm{KR}(s, C_i) \tag{5.1}$$

The optimal solution $\hat{A}$ for the assignment of samples to clusters can be found via a brute-force search. However, the number of possible assignments of n samples to k clusters is given by the Stirling partition number $S(n, k)$ [133], which is too large for any dataset of realistic size.

Hence, our implementation follows the Lloyd-Forgy algorithm [117, 217], which is the standard heuristical method to solve the k-means problem. It iteratively improves the assignments A and the centroids C in two alternating steps, as shown in Algorithm 5.1.

Algorithm 5.1 Phylogenetic k-means

1: initialize k *Centroids*
2: **while** not converged **do**
3: assign each *Sample* to nearest *Centroid* (A)
4: update *Centroids* as mass averages of their *Samples* (C)
5: **return** *Assignments* A and *Centroids* C

By default, we use the k-means++ initialization algorithm [14] to obtain an initial set of k centroids. It works by subsequent random selection of samples as initial centroids, until k centroids have been selected. In each step, the probability of selecting a sample is proportional to its squared distance to the nearest already selected sample. Hence, centroids are preferably selected that are far away from each other. An alternative initialization is to select samples as initial clusters entirely at random. This is however more likely to yield sub-optimal clusterings [172].

Then, each sample is assigned to its nearest centroid, using the KR distance. Lastly, the centroids are updated to represent the average mass distribution of all samples that are currently assigned to them. This iterative process alternates between improving the assignments and improving the centroids. Thus, the main difference to normal k-means in the $\mathbb{R}^d$ vector space is the use of phylogenetic information: Instead of Euclidean distances on vectors, we use the KR distance, and instead of averaging vectors to obtain centroids, we use the average mass distribution on the tree.

The process is repeated until it converges, that is, the cluster assignments do not change any more between subsequent iterations, or until a maximum number of iterations have been executed. This second stopping criterion is added to avoid the super-polynomial worst case running time of k-means, which however almost never occurs in practice [13, 33].

The result of the algorithm is an assignment of each sample to one of the k clusters. As the algorithm relies on the KR distance, it clusters samples with similar relative abundances. The cluster centroids can be visualized as trees with a mass distribution, analogous to how Squash Clustering visualizes inner nodes of the clustering tree. That is, each centroid can be represented as the average mass distribution of the samples that were assigned to it. This allows for inspecting the centroids and thus interpreting how the samples were clustered. Examples of this are shown in Figure 5.4.

Algorithmic Improvements

In each assignment step of the algorithm, distances from all n samples to all k centroids are computed. This has a time complexity of $\mathcal{O}(n \cdot k)$. In order to accelerate this step, we can apply branch binning as introduced in Section 2.5.3. For the BV dataset, we found that even using just 2 bins per edge does not alter the cluster assignments. Branch binning reduces the number of mass points that have to be accessed in memory during KR distance calculations; however, the costs for tree traversals remain. Thus, we observed a maximal speedup of 75% when using one bin per branch, see Table 5.1 for details. Intermediate binning strategies are also possible: instead of binning all masses of the input samples, one can just bin the centroid masses.

Furthermore, during the execution of the algorithm, empty clusters can occur, for example, if k is greater than the number of natural clusters in the data. Although this problem did not occur in our tests, we implemented the following solution: First, find the cluster with the highest variance. Then, choose the sample of that cluster that is furthest from its centroid, and assign it to the empty cluster instead. This process is repeated if multiple empty clusters occur at once in an iteration.

5.2.2 Imbalance k-means

We further propose *Imbalance k-means*, which is a variant of k-means that makes use of the edge imbalance transformation (see Section 2.5.3), and thus takes the clades of the reference tree into account. In order to quantify the difference in imbalances between two samples, we use the Euclidean distance between their imbalance vectors (that is, rows of the imbalance matrix). This is a suitable distance measure, as the imbalances implicitly capture the tree topology as well as the placement mass distributions. As a consequence, the expensive tree traversals required for Phylogenetic k-means are not necessary for these calculations. The algorithm takes the edge imbalance matrix of normalized samples as input, and performs a standard Euclidean k-means clustering following the Lloyd-Forgy algorithm [117, 217].

This variant of k-means tends to find clusters that are consistent with the results of Edge PCA, as both use the same input data (imbalances) and both operate in Euclidean space. Furthermore, as the method does not need to calculate KR distances, and thus neither involves tree traversals nor needs to consider each placement location (or bins of those) separately, it is several orders of magnitude faster than Phylogenetic k-means. For example, on the HMP dataset (see Appendix B.3), it runs in a few seconds, instead of several hours needed for Phylogenetic k-means; see Section Section 5.3.4 for details.

5.2.3 Finding Appropriate Values for k

A commonly criticized general downside of k-means clustering is that the number of clusters k is an input parameter of the algorithm. Hence, a key question is how to select an appropriate k that reflects the number of "natural" clusters in the data. There exist various suggestions in the literature [29, 143, 288, 308, 357, 358].

We evaluated the Elbow method [357], which is a straight forward method that yielded reasonable results for our test datasets. It works by plotting the cluster variance, that is, the average squared distance of the samples to their assigned cluster centroids, for different values of k. On the one hand, for low values of k, many samples exhibit a large distance to their assigned centroid, inducing a high variance. On the other hand, higher values of k further split the clusters and hence reduce the variance. Thus, at a given point, increasing k only yields a marginal change in variance. If the data has a natural number of clusters, the corresponding k at this point produces an angle in the plot, called the *elbow*. Thus, the presence of an elbow in the variance plot indicates reasonable values for k. The Elbow method, as well as other methods for finding a reasonable number of clusters [308], induce additional computational cost by having to run the algorithm repeatedly with a range of values for k.

Moreover, for a quantitative evaluation of the clusterings, we used the k that arose from the number of distinct categories or labels based on the available meta-data for the data. For example, the samples of the HMP dataset are labeled with 18 distinct body sites, describing where each sample was taken from, c. f. Figure 5.5.

5.3 Evaluation and Results

We now evaluate the two k-means variants in terms of their clustering accuracy and performance. We used the Bacterial Vaginosis (BV) dataset (see Appendix B.1 for details) as an example of a small dataset to which methods such as Squash Clustering [239] are still applicable for comparison, and the Human Microbiome Project (HMP) dataset (see Appendix B.3) to showcase that our methods scale to datasets that are too large for existing methods.

5.3.1 BV Dataset

We placed the samples of the BV dataset [339] on the re-inferred reference tree of their original reference sequences to test whether our methods work as expected. To this end, we ran both Phylogenetic k-means and Imbalance k-means on the BV dataset, and compare the results to the existing analyses of the data [239, 339]. We chose $k := 3$, inspired by the findings of Srinivasan et al. (2012) [339]. There, they distinguish between subjects affected by Bacterial Vaginosis and healthy subjects, and further separate the healthy ones into two categories depending on the dominating clade in the vaginal microbiome, which is either *Lactobacillus iners* or *Lactobacillus crispatus*. Any choice of $k > 3$ would simply result in smaller, more fine-grained clusters, but would not change the general findings of these experiments. The number of clusters is also evaluated using the Elbow method later in Section 5.3.3.

For each of the 220 samples of the dataset, we hence obtained two cluster assignments: First, by using Phylogenetic k-means, we obtained the cluster assignment *PKM*. Second, by using Imbalance k-means, we obtained assignment *IKM*. In the following figures, the samples are represented by colored dots: red, green, and blue

show the cluster assignments *PKM*, while purple, orange, and gray show the cluster assignments *IKM*. We use two different color sets for the two methods, in order to make them distinguishable at first glance. Note that the mapping of colors to clusters is arbitrary and depends on the random initialization of the algorithm.

We then conducted Squash Clustering and Edge PCA (Section 2.5.5) on the dataset, thereby reproducing results of previous studies, as well as two alternative dimensionality reduction methods. This allows for a direct comparison between our novel and the existing methods.

Comparison to Squash Clustering

The comparison of our k-means clustering assignments to Squash Clustering is shown in Figure 5.2. As can be seen in Figure 5.2(a), Squash Clustering as well as Phylogenetic k-means can distinguish healthy subjects from those affected by Bacterial Vaginosis. Healthy subjects constitute the lower part of the cluster tree. They have shorter branches between each other, indicating the smaller KR distance between them, which is a result of the dominance of *Lactobacillus* in healthy subjects. The same clusters are found by Phylogenetic k-means: As it uses the KR distance, it assigns all healthy subjects to one cluster (shown in red), which is consistent with the short cluster tree branches in Figure 5.2(a). The green and blue clusters contain the individuals that are most affected by the disease.

In Figure 5.2(b), we compare Squash Clustering to Imbalance k-means. Here, the distinction between the two *Lactobacillus* clades can be seen by the purple and orange cluster assignments. The cluster tree also separates those clusters into two clades. The separate small group of orange samples above the purple clade is an artifact of the tree visualization (ladderization), and actually is close to the other orange samples below. The diseased subjects are all assigned to the gray cluster, represented by the upper half of the cluster tree. It is apparent that both methods separate the same samples from each other.

Comparison to Edge PCA

In Figure 5.3, we compare the assignments obtained from our k-means variants to several dimensionality reduction methods, such as Edge PCA. The figure reveals additional details about how the k-means method works, that is, which samples are assigned to the same cluster.

The first row of Figure 5.3 shows the result of Multidimensional Scaling (MDS) of the pairwise KR distance matrix between the samples. MDS [105, 196, 233] is a dimensionality reduction method that can be used for visualizing levels of similarity between data points. Given a pairwise distance matrix, it finds an embedding into lower dimensions (in this case, 2 dimensions) that preserves higher dimensional distances as well as possible.

The distinguishing features between the green and the blue cluster are not apparent in the Squash cluster tree in Figure 5.2(a). This can however be seen in Figure 5.3(a),

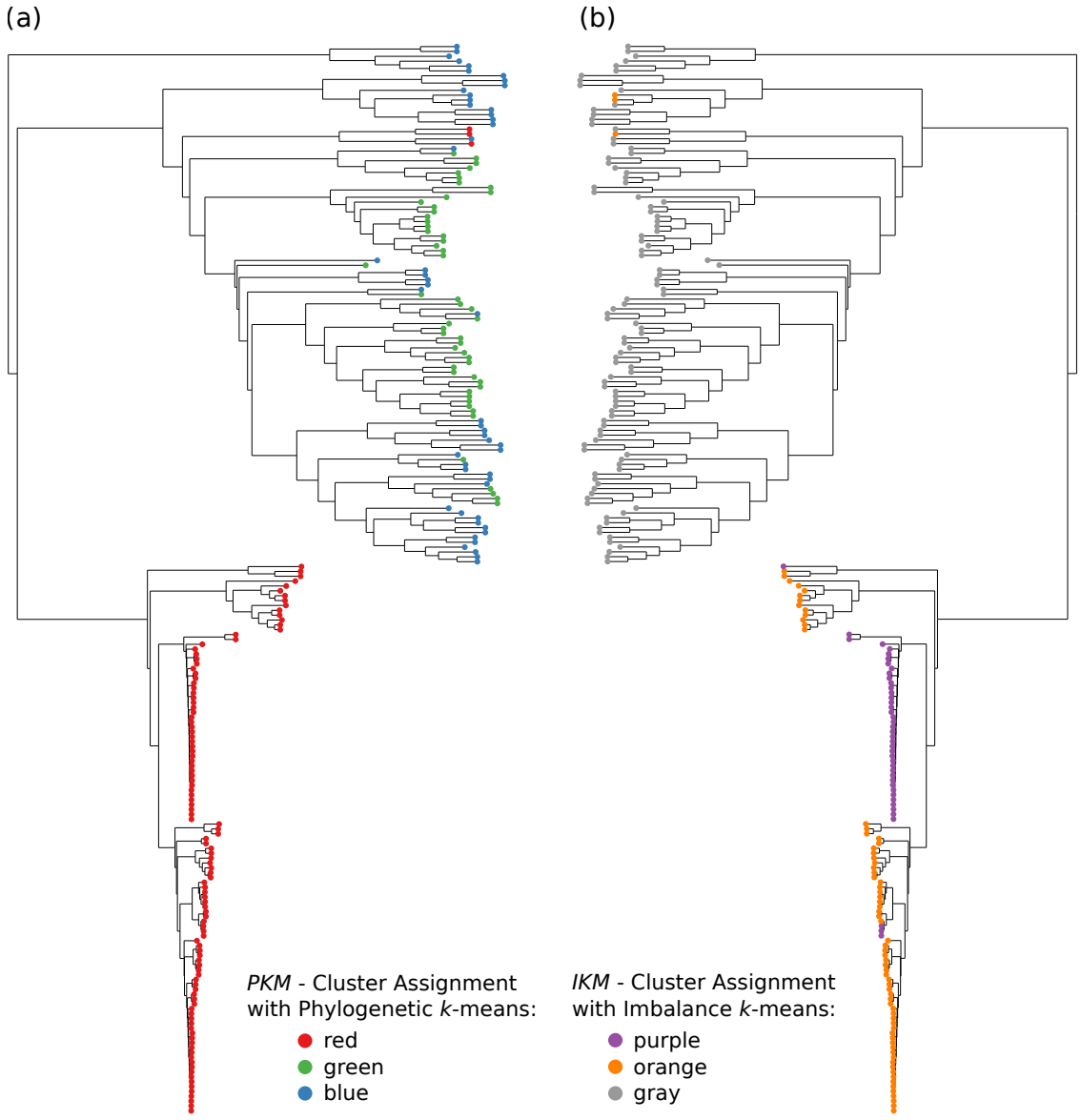

Figure 5.2: Comparison of k-means clustering to Squash Clustering. We applied Squash Clustering to the BV dataset [339], to compare it to the assignments obtained from our k-means variants. (a) Hierarchical cluster tree of the samples, using Squash Clustering. The tree is a recalculation of Figure 1(A) of Srinivasan et al. (2012) [339]. Each leaf represents a sample; branch lengths are KR distances. We added color coding for the samples, using *PKM*. The lower half of the red samples are mostly healthy individuals, while the green and blue upper half are patients affected by Bacterial Vaginosis. (b) The same tree, but annotated by *IKM*. The tree is flipped horizontally for ease of comparison. The healthy subjects are split into two sub-classes, discriminated by the dominating species in their vaginal microbiome: orange and purple represent samples were *Lactobacillus iners* and *Lactobacillus crispatus* dominate the microbiome, respectively. The patients that are most affected by BV are clustered in gray.

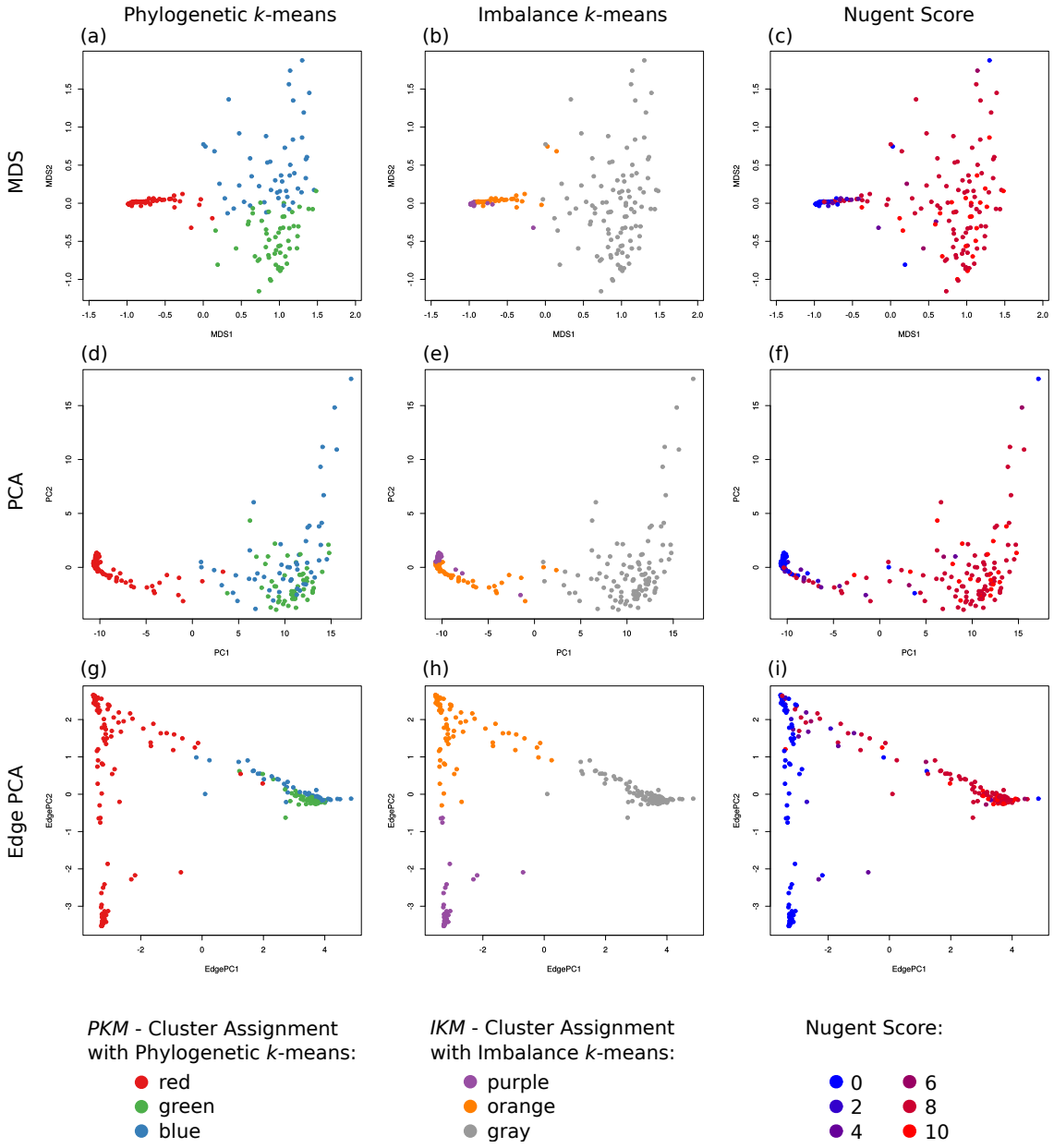

Figure 5.3: Comparison of k-means clustering to MDS, PCA, and Edge PCA. Here, we show the dimensionality reduction methods MDS, PCA, and Edge PCA (one per row) on the BV dataset. MDS and PCA were calculated on the pairwise KR distance matrix of the samples, Edge PCA was calculated using the placements on the re-inferred RT of the original publication [339]. The plots are colored by the cluster assignments *PKM* and *IKM* as found by our k-means variants (first two columns), and by the Nugent score of the samples (third column). The Nugent score is included to allow comparison of the health status of patients with the clustering results. Figures (f) and (i) are recalculations of Figures 4 and 3 of Matsen and Evans (2011) [240], respectively.

which shows the MDS plot colored by *PKM*. Here, the red cluster forms a dense region, which is in agreement with the short branch lengths in the cluster tree of Figure 5.2(a). At the same time, the green and blue cluster are separated in the MDS plot, but form a coherent region of low density. This indicates that $k := 3$ might be too large when applying Phylogenetic k-means to this dataset. That is, the actual clustering just distinguishes healthy from sick patients (c. f. Figure 5.7), meaning that 2 dimensions might also suffice here. Although the separation between green and blue samples is smooth, it shows that Phylogenetic k-means finds clusters that are based on the KR distance between samples, and thus yields results that are consistent with Squash Clustering and MDS.

A similar visualization of the pairwise KR distances is shown in the second row of Figure 5.3, where we applied standard Principal Component Analysis (PCA) [105, 196] to the pairwise KR distance matrix by interpreting it as a data matrix. Although it is mathematically sound, the direct application of PCA to a distance matrix lacks a simple interpretation, which was previously used to motivate Edge PCA (c. f. Section 2.5.5). Still, this can be seen as a visualization of the distances that helps understanding our methods.

For example, in Figure 5.3(d), which shows the PCA plot colored by *PKM*, the red cluster again is clearly separated from the rest. This time however, the distinction between the green and the blue cluster is not as apparent as in Figure 5.3(a).

Furthermore, Figures 5.3(b) and 5.3(e) show the MDS and the PCA plot, respectively, this time colored by *IKM*. Here, the purple cluster found by Imbalance k-means forms a dense cluster of close-by samples on the left of the plots, which is in accordance with the short branch lengths of this cluster as shown in the clustering tree in Figure 5.2(b). The orange cluster is slightly more spread out in the plots, which again can be seen by the longer branch lengths in Figure 5.2(b).

Finally, we applied Edge PCA to the samples, as shown in the last row of Figure 5.3. In particular, Figure 5.3(h) compares Imbalance k-means to Edge PCA by coloring the plot using *IKM*. Because both methods work on edge imbalances, they group the data in the same way, and are thus consistent with each other. That is, they clearly separate the two healthy groups and the diseased one from each other. Edge PCA forms a plot with three corners, which are colored by the three Imbalance k-means cluster assignments.

Cluster Centroids

As mentioned before, an advantage of using phylogenetic placements as input to our k-means clustering methods is the ability to visualize cluster centroids by showing the average mass distribution of the samples assigned to them on the underlying Reference Tree (RT). In Phylogenetic k-means, the mass distributions of the centroids are already part of the algorithm, as they are needed for calculating the KR distances between samples and centroids. In Imbalance k-means however, the placement data is used in form of edge imbalances instead of masses. Still, after convergence of the

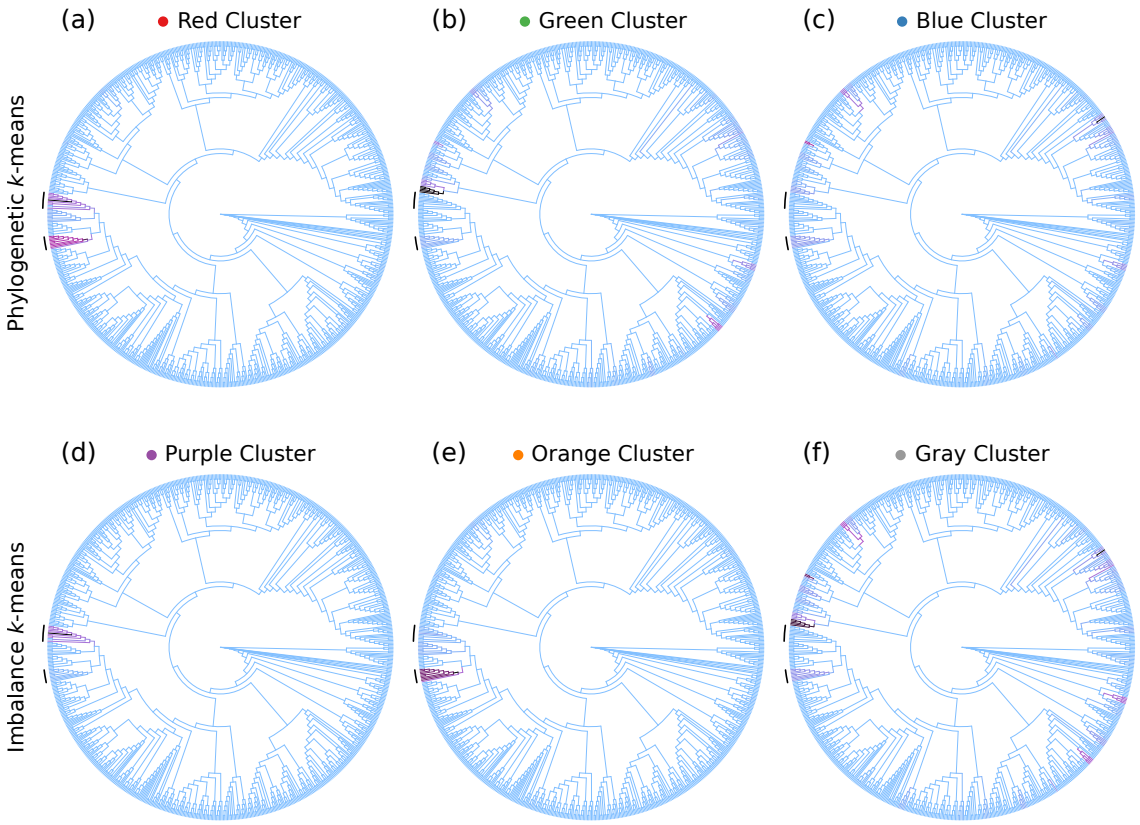

Figure 5.4: Example of k-means cluster centroid visualization. Here we show the cluster centroids as found by our k-means variants using the BV dataset, visualized on the reference tree via color coding. The cluster assignments are the same as in Figures 5.2 and 5.3; the first row show the three clusters found by Phylogenetic k-means, the second row the clusters found by Imbalance k-means. Each tree represents one centroid around which the samples were clustered, that is, it shows the combined masses of the samples that were assigned to that cluster. The edges are colored relative to each other, using a linear scaling of light blue (no mass), purple (half of the maximal mass), and black (maximal mass). The two *Lactobacillus* clades are marked with black arcs on the left of the trees.

algorithm, the per-centroid average mass can be calculated from the input samples and again visualized on the RT.

Examples of this are shown in Figure 5.4, for both sample assignments *PKM* (first row) and *IKM* (second row) of the BV dataset. As explained above, the samples can be split into three groups: The diseased individuals, which have placement mass in various parts of the tree, as well as two groups of healthy individuals, with placement mass in one of two *Lactobacillus* clades, marked with black arcs in Figure 5.4. This grouping is also clearly visible in the trees. The red cluster in 5.4(a) for example represents all healthy subjects; thus, most of its mass is located in the two *Lactobacillus* clades. The purple and orange clusters in 5.4(d) and (e) on the other hand show a difference in placement mass between those clades. Furthermore,

the placement mass of the gray cluster in 5.4(f) is mostly a combination of the masses of the green and blue cluster in 5.4(b) and (c), all of which represent diseased subjects. These observations are in accordance with the previous findings above, and further support that our methods yield results that are in agreement with existing methods.

5.3.2 HMP Dataset

The Human Microbiome Project (HMP) dataset [160, 250] (see Appendix B.3 for details) is used here as an example to show that our method scales to large datasets. To this end, we used the unconstrained *Bacteria* RT with 1914 taxa obtained from our PhAT method, see Section 3.3.1 for details. The tree represents a broad taxonomic range of *Bacteria*, that is, the sequences were *not* explicitly selected for the HMP dataset, in order to test the robustness of our clustering methods. We then placed the 9192 samples of the HMP dataset with a total of 118 701 818 sequences on that tree, and calculated Phylogenetic and Imbalance k-means on the samples.

The freely available meta-data for the HMP dataset labels each sample by the body site where it was taken from. As there are 18 different body site labels, we used $k := 18$. The resulting clustering assignments are shown in Figure 5.5. Furthermore, in Figure 5.6, we show a clustering of this dataset into $k := 8$ broader body site groups to exemplify the effect of using different values of k. See Table B.2 for the grouping of the original body site labels. The effect of k on the clustering results is further explored by using the Elbow method, as described later in Section 5.3.3.

Ideally, all samples from one body site would be assigned to the same cluster, hence forming a diagonal in the plots of Figure 5.5 and Figure 5.6. However, as there are several nearby body sites, which share a large fraction of their microbiome [160], we do not expect a perfect clustering. Furthermore, we used a broad reference tree that might not be able to resolve details in some clades. Nonetheless, the clustering is reasonable, which indicates a robustness against the reference taxa choice.

The plots of the two k-means variants generally exhibit similar characteristics in Figure 5.5. Most prominently, the stool and vaginal samples are clearly clustered into coherent blocks in both variants. There are however also some differences. For example, the samples from the body surface (arm, nose, and ear) form two relatively dense clusters (columns) in Figure 5.5(a), whereas those sites are spread across four or five clusters in Figure 5.5(b). On the other hand, the mouth samples are more densely clustered in Figure 5.5(b).

It is remarkable that the oral samples are mostly split into two blocks, corresponding to the front of the mouth and its back, in both variants of k-means in Figure 5.5. This indicates that the clustering is sensitive to such subtle differences. However, within these blocks, there is some fuzziness in the clustering. This might be caused by our broad reference tree, and could potentially be resolved by using a tree that is more specialized for the data/region. It could however also simply be an artifact of the homogeneity of samples taken from the close-by oral regions of the human body.

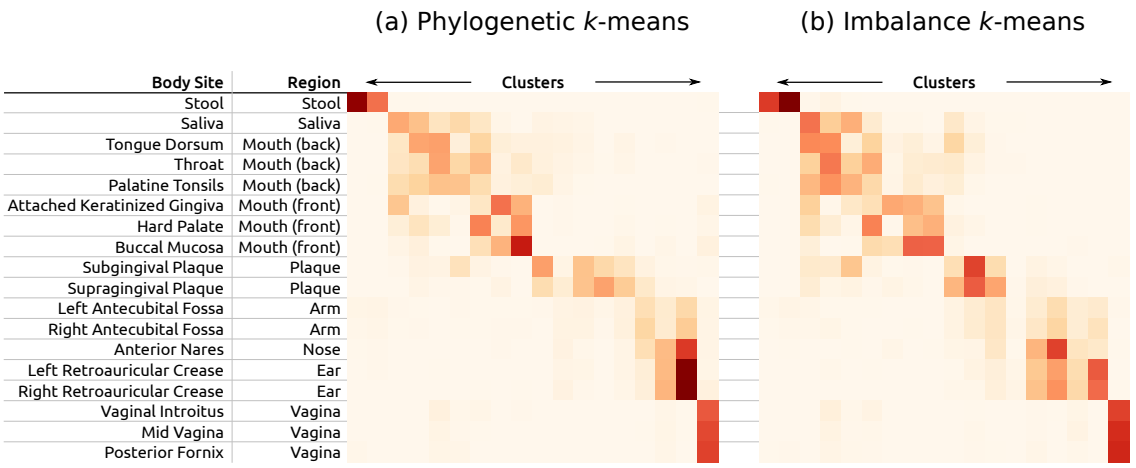

Figure 5.5: k-means cluster assignments of the HMP dataset with $k := 18$. Here, we show the cluster assignments as computed via (a) Phylogenetic k-means and (b) Imbalance k-means on the HMP dataset. We used $k := 18$, which is the number of body site labels in the dataset, in order to compare the clusterings to this "ground truth". Each row represents a body site; each column one of the 18 clusters found by the algorithm. We also show a translation of the body site labels into the corresponding body regions. The color values indicate how many samples of a body site are assigned to each cluster. Similar body sites are clearly grouped together in coherent blocks, indicated by darker colors. For example, the stool samples are split into two clusters (topmost row), while the three vaginal sites are all put into one cluster (rightmost column). However, the algorithm cannot always distinguish between nearby body sites, as can be seen from the fuzziness of the oral sample clusters.

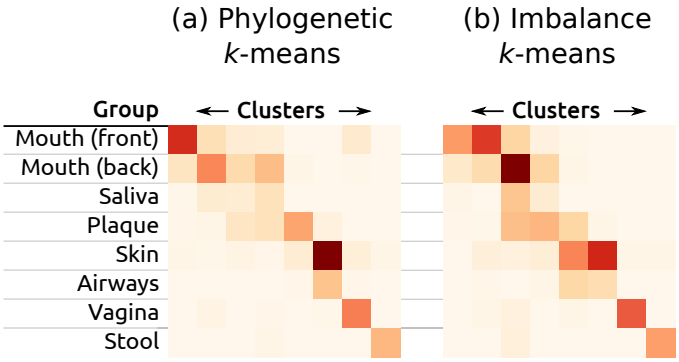

Figure 5.6: k-means cluster assignments of the HMP dataset with $k := 8$. Here, we again show the cluster assignments as yielded by (a) Phylogenetic k-means and (b) Imbalance k-means on the HMP dataset, but with k being set to 8, instead of $k := 18$ as in Figure 5.5. These 8 clusters are based on an aggregation of the original body site labels into groups, as shown in Table B.2. Each row represents a body site group; each column one of the 8 clusters found by the algorithm. Some of the body sites are again clearly separated, while particularly the samples from the oral region are distributed over different clusters.

Lastly, using the two *General* trees of our PhAT method as described in Section 3.3.1, we again evaluated the cluster assignments on the HMP datasets (data not shown). Using this even less specific set of reference taxa yielded cluster assignments which are almost identical to the ones shown above, except for a slightly increased fuzziness. We thus expect that the clustering improves when using an RT containing taxa specifically chosen for the type of sequences in the dataset.

5.3.3 Elbow Method

As explained in Section 5.2.3, plots of the cluster variance can be used for the Elbow method in order to find an appropriate number of clusters in a dataset [357]. Figure 5.7 shows these plots for our two test datasets.

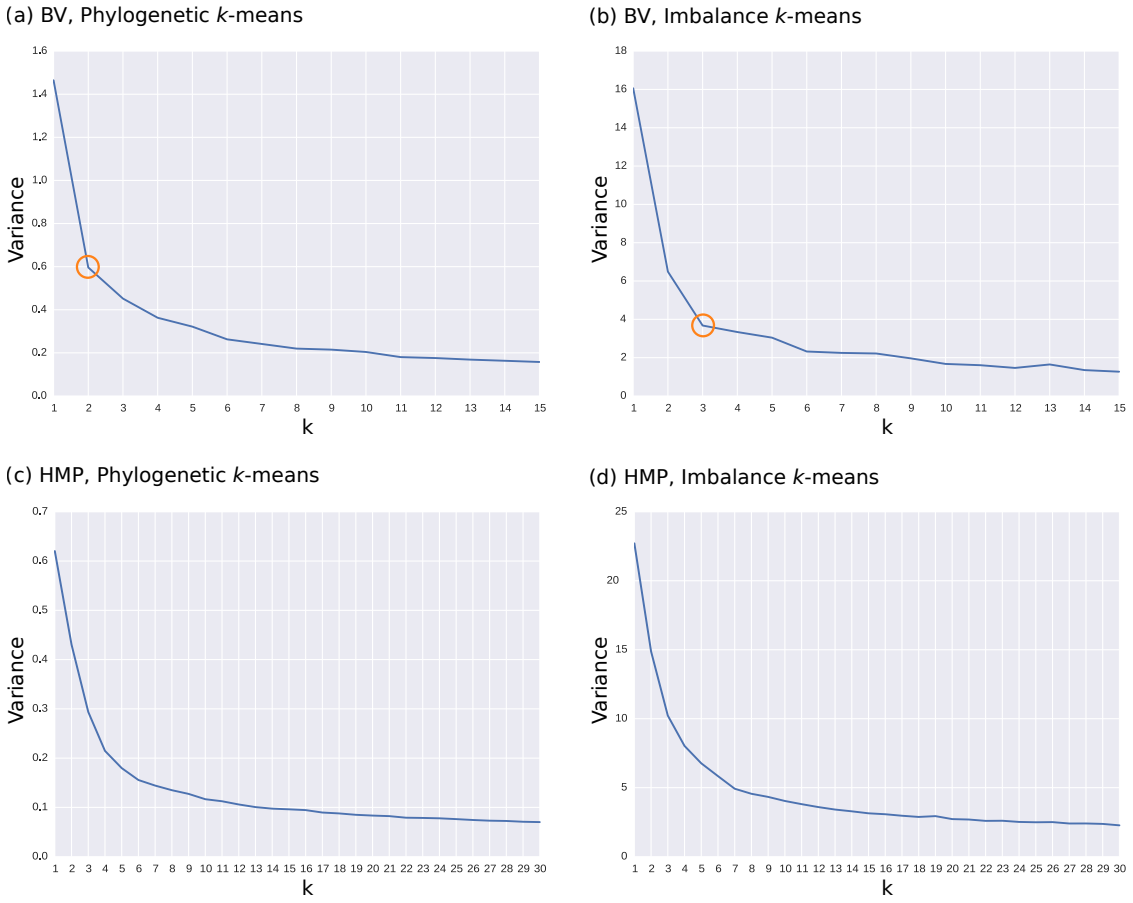

Figure 5.7: Variances of k-means clusters in our test datasets. The figures show the cluster variance for different values of k. The first row are clusterings of the BV dataset, the second row of the HMP dataset. They were clustered using Phylogenetic k-means (first column), and Imbalance k-means (second column), respectively. Accordingly, (a) and (c) use the KR distance, while (b) and (d) use the Euclidean distance to measure the variance. Orange circles mark potential elbow points.

The plots for the BV dataset in Figures 5.7(a) and 5.7(b) show elbows at $k := 2$ and 3, respectively, which are marked with orange circles. These values are consistent with previous findings, c. f. Figures 5.2 and 5.3. On the one hand, Phylogenetic k-means splits the samples into a distinct red cluster, separated from the green and blue clusters, effectively forming two clusters, which represent the health status of the patients. On the other hand, Imbalance k-means yields three separate clusters in purple, orange, and gray, which correspond to two clusters for the dominant *Lactobacillus* clades, as well as a cluster for the patients affected by BV.

In the plots for the HMP dataset, the elbow is less pronounced. We suspect that this is due to two reasons, as explained in Section 5.3.2: (i) the broad reference tree might not being able to adequately resolve fine-grained differences between samples, and (ii) nearby body sites might simply be too homogeneous in their metagenomic composition for a clear separation into clusters. Likely candidates for k are 4–6 for Phylogenetic k-means in Figure 5.7(c) and around 7 for Imbalance k-means in Figure 5.7(d). These values are consistent with the number of coherent "blocks" of clusters, as shown in Figure 5.5. Clearer results for this dataset might be obtained with other methods for finding "good" values for k, although we did not test them here.

5.3.4 Performance

The complexity of Phylogenetic k-means is in $\mathcal{O}(k \cdot i \cdot n \cdot e)$, with k clusters, i iterations, and n samples, and e being the number of tree edges, which corresponds to the number of dimensions in standard Euclidean k-means. As the centroids are randomly initialized, the number of iterations can vary; in our tests, it was mostly below 100. For the BV dataset with 220 samples and a reference tree with 1590 edges, using $k := 3$, our implementation executed 9 iterations, requiring 35 s and 730 MB of main memory on a single core. For the HMP dataset with 9192 samples containing 119 million sequences, and a reference tree with 3824 edges, we used $k := 18$, which took 46 iterations and ran in 2.7 h on 16 cores, using 48 GB of memory.

In contrast to this, Imbalance k-means neither needs to conduct any expensive tree traversals, nor take single placement locations into account, but instead operates on compact vectors with one entry per edge, using Euclidean distances. It is hence several orders of magnitude faster than Phylogenetic k-means, and only needs a fraction of the memory. For example, using again $k := 18$ for the HMP dataset, the algorithm executed 75 iterations in 2 s. It is thus also applicable to extremely large datasets.

Furthermore, our implementation of the KR distance calculation is highly optimized and outperforms the existing implementation in GUPPY [241] by orders of magnitude. The KR distance between two samples has a linear computational complexity in both the number of QSs and the tree size. As a test case, we computed the pairwise distance matrix between sets of samples. Calculating this matrix is quadratic in the number of samples, and is thus expensive for large datasets. For example, in order

to calculate the matrix for the BV dataset with 220 samples, GUPPY can only use a single core and required 86 min. Our KR distance implementation is faster and also supports multiple cores. It only needed 90 s on a single core; almost half of this time is used for reading input files. When using 32 cores, the matrix calculation itself only took 8 s. This allows to process larger datasets: The distance matrix of the HMP dataset with 9192 samples placed on a tree with 3824 branches for instance took less than 10 h to calculate using 16 cores. In contrast, GUPPY needed 43 days for this dataset. As the KR distance is used in Squash Clustering, our re-implementation of this method is also orders of magnitude faster than the original GUPPY implementation.

Lastly, in order to achieve additional speedup for even larger datasets, the mass binning method can be used, as explained in Section 2.5.3. The performance and the effects of binning on the distance values are shown in Table 5.1.

Table 5.1: Effect of branch binning on the KR distance of the HMP dataset. Here we show the effect of per-branch placement binning on the run-time and on the resulting relative error when calculating the pairwise KR distance matrix between samples, by example of the Human Microbiome Project (HMP) [160, 250] dataset (see Appendix B.3 for details). Because of the size of the dataset (9192 samples) and reference tree (1914 taxa), we executed this evaluation in parallel on 16 cores. The first row shows the baseline performance, that is, without binning.

Bins	Time (h:mm)	Speedup	Relative Δ
-	9:46	1.00	0.000000
256	6:58	1.40	0.000008
128	6:39	1.47	0.000015
64	6:30	1.50	0.000035
32	6:25	1.52	0.000124
16	6:13	1.57	0.000272
8	6:08	1.59	0.000669
4	6:07	1.60	0.002747
2	6:04	1.61	0.004284
1	5:35	1.75	0.011585

The first row represents the baseline case of using no binning, where each placement location of the 118 million sequences is taken into account in the computation of the KR distance. Hence, even binning the masses on each of the 3824 branches of the tree into 256 intervals already yields a substantial speedup. When using fewer bins per branch, the run-time further decreases, at the cost of slightly increasing the average relative error. Still, even when compressing the placement masses into only one bin per branch (that is, just using per-branch masses), the average relative error of the KR distances is around 1%, which is acceptable for most applications. However, considering that the run-time savings are not substantially better for a low

number of bins, we recommend using a relatively large number of bins, e. g., 32 or more. This is because run-times of KR distance calculations also depend on other effects such as the necessary repeated full tree traversals. We also conducted these tests on the BV dataset, were the relative error is even smaller. However, because of the comparatively small size of this dataset, the run-times are too short for accurate measurements, and thus not shown here.

5.4 Summary and Outlook

In this chapter, we presented two adapted variants of the k-means method, which exploit the structure of phylogenetic placement data to identify clusters of environmental samples. The methods build upon algorithms such as Squash Clustering and can be applied to substantially larger datasets, as they construct a pre-defined number of clusters. They are thus useful to identify similarities between large sets of metagenomic samples.

Phylogenetic k-means uses the KR distance to assess sample similarity, and hence yields cluster assignments that are consistent with Squash Clustering. Imbalance k-means on the other hand is based on edge imbalances, and thus yields assignments that are consistent with Edge PCA, which also uses edge imbalances. Furthermore, Imbalance k-means operates in the Euclidean space instead of using mass distributions on trees. It is therefore several orders of magnitude faster than Phylogenetic k-means and can hence be applied to extremely large datasets.

The choice of a reasonable value for k is a general issue in k-means clustering. It might hence be worth to experiment with more sensitive methods for estimating k than the Elbow method evaluated here. For future exploration however, other forms of cluster analyses might offer more potential. For example, methods such as soft k-means clustering [28, 89] or density-based methods [195] could be explored for clustering metagenomic sequence samples.

The main challenge when adopting such methods to phylogenetic placement data consists in making them phylogeny-aware. That is, they have to be extended to this type of data by using mass distributions on trees instead of operating on $\mathbb{R}^d$ vectors in the Euclidean space, and using appropriate distance measures such as the KR distance to assess sample similarity. In case of using edge imbalances (instead of edge masses), the adaptation of existing clustering methods is more straight forward and might work by plugging in the edge imbalance matrix into existing implementations.

6. Balances

This chapter is derived from parts of the peer-reviewed open-access publication:

> **Lucas Czech** and Alexandros Stamatakis. "Scalable Methods for Analyzing and Visualizing Phylogenetic Placement of Metagenomic Samples." *PLOS ONE*, 2019, Volume 14, Issue 5, Page e0217050.

All text and figures in this chapter were created by Lucas Czech. Mathematical decisions of the balances and taxon weighting scheme emerged from discussion with Justin Silverman.

6.1 Background and Motivation

The concepts and methods presented in the previous Chapters 4 and 5 resemble two recent approaches for analyzing phylogenetic data: the Phylogenetic Isometric Log-Ratio (*PhILR*) transformation and balances [330], as well as Phylogenetic Factorization (*Phylofactorization*) [376]. These methods use a tree inferred from the OTU sequences of the samples (instead of a fixed reference tree), and annotate the abundances of OTUs per sample on the tips of this tree (instead of placement masses on the branches). The methods use these data to draw conclusions about compositional changes of OTU abundances in clades of the tree in different samples, as well as relationships of per-clade OTU abundances with environmental meta-data variables. See Section 2.2.5 for details on OTU clustering of metagenomic samples.

In both of these approaches, a *balance* between OTU abundances in two subtrees of the underlying tree is computed. This is a measure of contrast that expresses which of the two subtrees comprises more OTUs in a sample. In the PhILR transform [330], these balances are computed for the two subtrees below each inner node of a rooted binary tree, while ignoring abundances in the respective remainder of the tree. In

Phylofactorization however, these balances are computed for the two subtrees that are induced by the splits/edges of the tree [376]. This is highly similar to the concept of edge imbalances that we introduced in Section 2.5.3. Note that despite sharing a similar name and exhibiting conceptual similarities, balances and edge imbalances are distinct approaches that should not be confused. We later discuss respective similarities and differences in more detail.

Furthermore, we remark that the *Balance Trees* method [267] employs analogous concepts by calculating the balance of nodes using the isometric log-ratio of OTU abundances. However, instead of using a phylogenetic tree, it assumes any binary partitioning of the OTUs, e. g., obtained from a UPGMA clustering [202] of the OTUs based on a meta-data feature. These nodes thus correspond to specific meta-data values, again allowing for statements about the changes in OTU abundances that occur with changing environmental variables. As we already have a binary partitioning in form of the reference tree, we do not further consider the Balance Trees approach here.

In this and the following chapter, we present adaptations of the PhILR transform (balances) and of Phylofactorization to phylogenetic placement data. The main adaptation step consists in placing masses on the branches of our (fixed) reference tree, instead of only considering masses (abundances) at the tips of the OTU tree. Here, we focus on balances that contrast the subtrees induced by edges of the tree, as used by Phylofactorization [376], because this is more natural in the context of phylogenetic placement data. The same concepts could however also be employed for subtrees below nodes, as used by the PhILR transform [330].

6.2 Methods and Implementation

In Section 2.5.3, we briefly outlined the inherently compositional nature of metagenomic sequence data [129, 295]. For a thorough discussion of the implications of this, see Silverman et al. (2017) [330]. One solution is to transform the data into an unconstrained space that is not compositional. This can, for example, be achieved via the Isometric Log-Ratio (ILR) transform [102, 295], which, given a compositional space, creates a new coordinate system with an orthonormal basis [100]. The ILR transform requires a sequential binary partitioning of the underlying original space [284]. As suggested by Silverman et al. (2017) [330], a bifurcating phylogenetic tree (e. g., our RT) represents such a partitioning, which also provides a meaningful way of interpreting the resulting coordinates. This so-called *Phylogenetic ILR* (PhILR) transform yields an ILR coordinate system that captures the evolutionary relationships of the phylogeny [330]. The resulting coordinates are called *balances* [100, 102]. The balances obtained from an ILR transform represent the log-ratio of the geometric means of the data in the two subtrees. Hence, they can be interpreted as a contrast (log-ratio) between two aggregates (geometric means) of data. Furthermore, due to the orthogonality of the ILR basis vectors, the balances can be used by conventional statistical tools without suffering from compositional artifacts.

6.2.1 Phylogenetic ILR Transform for Placements

In the following, we present an adaptation of the PhILR transform and balances to placement data, based on the work of Silverman et al. (2017) [330]. See there for more details on the method and the underlying mathematical concepts, such as the connection between the ILR transform and the centered log-ratio (CLR) transform. We describe the computation of the Phylogenetic ILR transform, along with the changes needed for phylogenetic placement data. We focus on the computation for a single sample; for multiple samples, the process is simply repeated. We assume that a fixed Reference Tree (RT) (a sequential binary partitioning) along with the per-branch placements of the sequences in the sample are given.

The per-edge placements of the sample are represented by a vector $\boldsymbol{c}$ of size m, containing the absolute (not normalized) edge masses, where m is the number of edges in the tree. In other words, our input is a single row (one sample) of the edge masses matrix, as shown in Figure 6.1(a). The absolute masses are transformed into relative abundances as described in Section 2.5.3: Each element of $\boldsymbol{c}$ is divided by the sum of all elements, yielding the relative masses vector $\boldsymbol{x}$ for the given sample. In compositional data analysis, this operation is known as the *closure* of the data [6], and computed as:

$$\boldsymbol{x} = \left[\frac{c_1}{\sum_m c_m}, \ldots, \frac{c_m}{\sum_m c_m} \right] \tag{6.1}$$

The original PhILR furthermore allows to use per-taxon weights $\boldsymbol{p}$ in order to down-weigh the impact of low abundant taxa/OTUs [101, 330]. In our adaptation, this weighting scheme is accordingly changed to *per edge* weights $\boldsymbol{p}$ of the RT. Unfortunately, the nomenclature of existing publications (namely, Matsen and Evans (2011) [239] and Silverman et al. (2017) [330]) creates a conflict here: These weights are not to be confused with our terminology of likelihood weights and edge masses. Hence, in order to avoid confusion, and in line with the original terminology, we also call them *taxon weights* here, although they here refer to edges instead of taxa.

The default case of taxon weights $\boldsymbol{p} = (1, \ldots, 1)$ represents no weighting, where each edge equally contributes to the balance, while any $\boldsymbol{p} \neq (1, \ldots, 1)$ is a generalized form of the ILR transform [330]. We later describe an appropriate choice of weights in Section 6.2.2. These weights are applied to the relative masses $\boldsymbol{x}$ to obtain the shifted composition $\boldsymbol{y} = \boldsymbol{x}/\boldsymbol{p}$, using element-wise division [101].

In the original PhILR, balances are calculated for the two subtrees below a given node of the tree [330]. In the context of Phylofactorization, this has been generalized to balances between any two disjoint sets R and S of taxa (tips of the tree) [376]. We here build on the latter, but again change R and S to refer to disjoint sets of edges of our reference tree. We use the notation $\boldsymbol{y}_R$ and $\boldsymbol{p}_R$ to refer to the subsets

of masses and weights of the given sample at the edges in R. Then, the balance y^* between the sets R and S is computed as:

$$y^*(R,S) \;=\; \sqrt{\frac{\nu_R \cdot \nu_S}{\nu_R + \nu_S}} \;\cdot \log \frac{\mathrm{gm}(\boldsymbol{y}_R, \boldsymbol{p}_R)}{\mathrm{gm}(\boldsymbol{y}_S, \boldsymbol{p}_S)} \tag{6.2}$$

The first term is a scaling term that, for a given edge, is constant across all samples. It ensures unit length of the ILR basis elements, and uses the sums of weights in $\boldsymbol{p}$:

$$\nu_R = \sum_{r \in R} p_r \quad \text{and} \quad \nu_S = \sum_{s \in S} p_s \tag{6.3}$$

The second term is the log-ratio of geometric means, where $\mathrm{gm}(\boldsymbol{y}_R, \boldsymbol{p}_R)$ is the weighted geometric mean of the values in $\boldsymbol{y}_R$ with weights $\boldsymbol{p}_R$:

$$\mathrm{gm}(\boldsymbol{y}_R, \boldsymbol{p}_R) = \left(\prod_{r \in R} {y_r}^{p_r} \right)^{\frac{1}{\nu_R}} = \exp\left(\frac{\sum_{r \in R} p_r \cdot \log y_r}{\sum_{r \in R} p_r} \right) \tag{6.4}$$

Note that if $\boldsymbol{p} = (1, \ldots, 1)$, Equation 6.2 represents the original ILR transform without a weighting scheme [102], Equation 6.3 equals the number of edges in R and S, respectively, and Equation 6.4 is the standard (unweighted) geometric mean. We show an example of the balance computation in Figure 6.1, which results in what we call the *balance matrix*.

Balances as defined here can be computed as a measure of *contrast* between any disjoint sets R and S of edges. Interchanging S and R flips the sign of the balance; this is irrelevant for the subsequent steps presented here, as long as the interchange is applied consistently. When computing the balance between the edges in the two subtrees induced by some given edge e, the conceptual similarity with the previously described edge imbalances (Section 2.5.3) becomes apparent: Imbalances use the difference of sums for contrasting and aggregating, while balances use the ratio of means for the same purpose. Hence, balances represent a similar transformation of the placement data, that can also be used to conduct analyses, such as the Phylofactorization, as presented in Chapter 7.

We however remark that using (unweighted) balances in our previously presented methods, such as Edge Correlation (Section 4.2.2) and k-means clustering (Section 5.2.1), might lead to spurious results, due to the insensitivity of the geometric mean to singular large values. That is, individual branches that accumulate a large fraction of the placement mass (sequence abundances) might only insignificantly change the geometric mean of their clade. However, such branches are typically the interesting ones, and hence should exert more influence on the transformation, which is exactly the purpose of the taxon weighting scheme. This further implies that balances are not indifferent to splittings of reference taxa into multiple representatives (pers. comm. with A. Washburne on 2018-11-23). We discuss the implications of this in more detail in the evaluation of the method (Section 6.3).

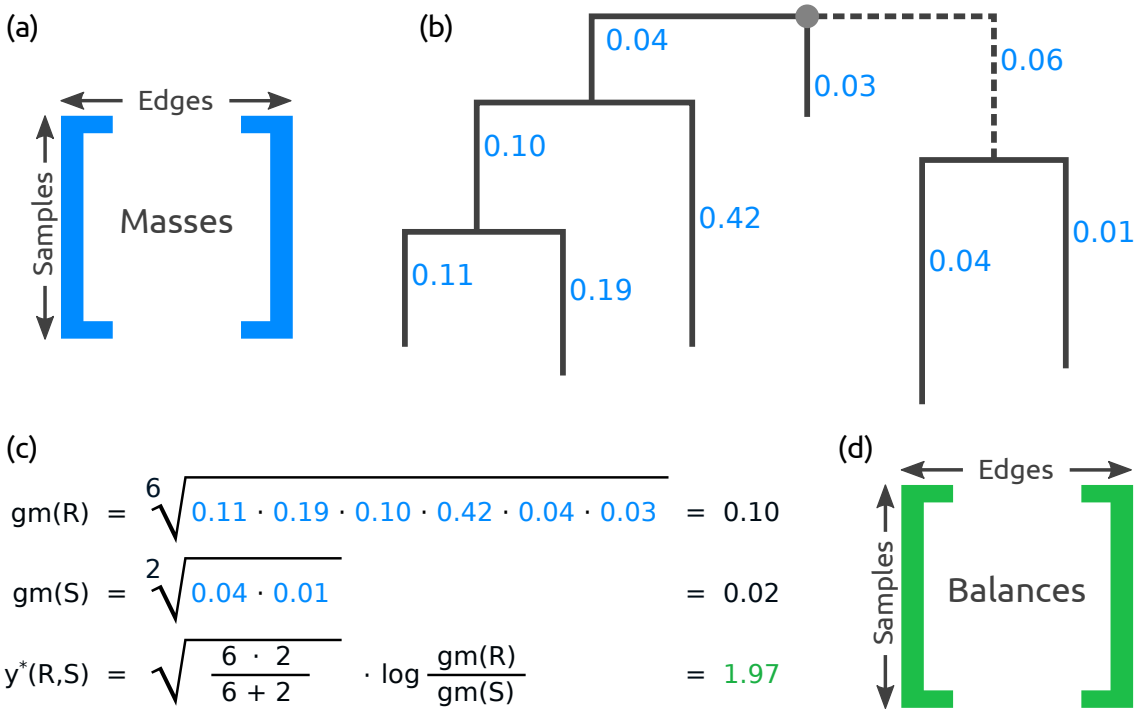

Figure 6.1: Example computation of the balances between two subtrees. (a) The basis of the computation are the per-branch masses of the samples, which are here summarized in the mass matrix, c. f. Figure 2.14(b). (b) We here show the computation of the balance for the two subtrees induced by the dashed edge of the tree, for one sample. Numbers next to edges are the accumulated per-edge placement masses of the sequences in the sample, that is, one row of the matrix. We call the left hand side of the tree R, and the right hand side S, as seen from the dashed edge. For simplicity, we do not use weighting here; that is, we assume $\boldsymbol{p} = (1, \ldots, 1)$. (c) First, the geometric means for both subtrees are calculated, then, their balance. The balance is positive, indicating that subtree R contains more placement mass on (geometric) average. (d) The compuation is repeated for all edges and for all samples, yielding the *balance matrix* shown here.

6.2.2 Taxon Weighting Scheme

The PhILR also allows for incorporating two distinct weighting schemes for the balances, one based on taxon abundances, and one based on the branch lengths of the underlying phylogeny [330]. As mentioned above, we implemented the former, while leaving the latter as future work.

We here describe how to adapt the taxon weights of Silverman et al. (2017) [330] to our placement-based approach, that is, how an appropriate vector $\boldsymbol{p}$ of taxon weights for the edges can be constructed. Originally, this taxon weighting scheme down-weighs the influence of low abundant taxa [330], which are known to be less reliable and more variable [130]. Here, we accordingly down weigh edges with low placement mass, for the same reasons. We follow the approach of Silverman et al.

(2017) [330], and construct the taxon weights by multiplicatively combining two terms:

1. A measure for the central tendency of the absolute edge masses, for example, their mean across all samples. This is the main component of the weight that yields low values for edges with low mass and vice versa.
2. A vector norm of the relative edge masses across the samples. This term additionally weighs edges by their specificity.

Our implementation allows to use the median, the arithmetic mean, and the geometric mean, as well as different ℓ_p-norms (such as the Manhattan, Euclidean, and maximum norm), and the Aitchison norm [284]. We follow the advice of [330], and by default use the geometric mean (with pseudo-counts added to the masses to avoid skew from edges without any placement mass) and the Euclidean norm. In that case, the weights for edge j are computed as follows:

$$p_j \;=\; \sqrt[n]{\prod_{i=1}^{n}(\tilde{c}_{ji}+1)} \;\cdot\; \sqrt{\sum_{i=1}^{n}\tilde{x}_{ji}^2} \tag{6.5}$$

Here, n is the number of samples, $\tilde{c}_j$ is the vector of absolute masses at edge j across all samples, and $\tilde{x}_j$ the vector of relative masses at edge j across all samples, both of length n. That is, these measures use the masses of all n samples; consequently, we here use columns instead of rows of the edge masses matrix of Figure 6.1(a) (which is identical to Figure 2.14(b)), where each column is used for the weights of the corresponding edge. The resulting taxon weights $\boldsymbol{p}$ are then fixed and used across the balance computation of all samples.

6.3 Evaluation and Results

As a first test of our adaptation of balances to placement data, we apply it to the BV dataset [339]. Further assessment of balances for placement data, also with the HMP dataset [160, 250], is implicitly conducted by the evaluation of Placement-Factorization in Section 7.3, which uses balances for aggregation and contrasting of subtrees. See Appendix B.1 and Appendix B.3 for descriptions of the datasets.

6.3.1 Principal Components

As balances are conceptually similar to edge imbalances, we perform analogous evaluations. To this end, we computed the per-edge balance for all edges of the BV reference tree, across all 220 samples. That is, for each edge, we computed the balance between the two subtrees induced by the edge. This yields the *balance matrix*, as shown in Figure 6.1(d), which corresponds to the imbalance matrix used for Edge PCA, c. f. Figure 2.14(b). Hence, a natural first visualization of the balances is to analyze their principal components, that is, to compute the PCA of the balance

matrix. The first two components of the BV balances are shown in Figure 6.2, for both variants of the balance computation (with and without taxon weighting). Furthermore, we can again employ the visualization of PCA eigenvectors on the reference tree as used in Edge PCA [239]. We show the results for the BV dataset in Figure 6.3, where we visualize the first two principal components of the balances with and without taxon weighting on the tree.

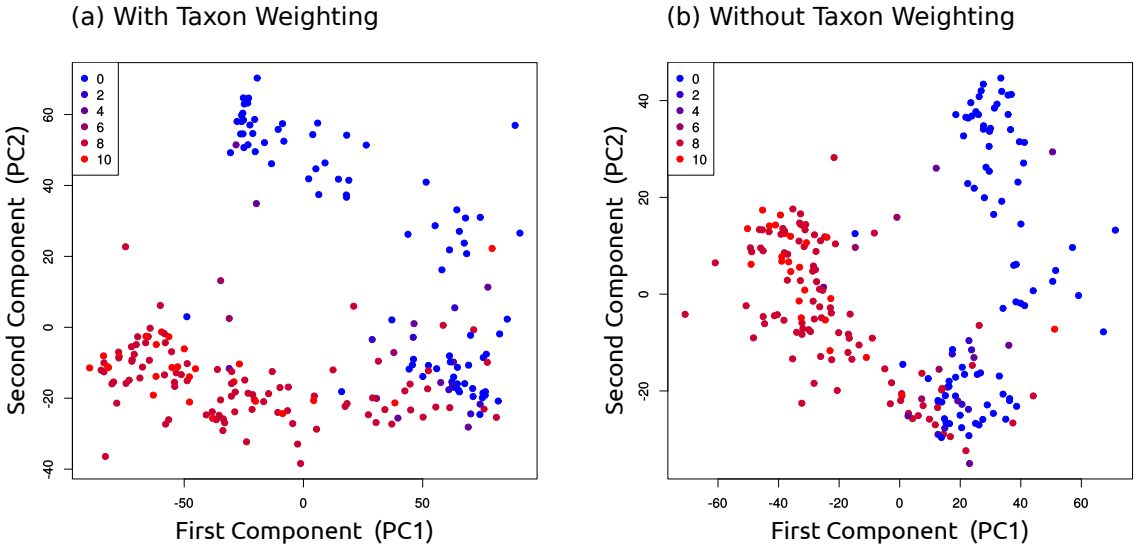

Figure 6.2: Projection of edge balance PCA components of the BV dataset. The plots show the first two principal components of a PCA on the per-edge balances, calculated on placements of the data on reference tree of the BV dataset. That is, for each edge of the tree, we calculated the balance (log-ratio of geometric means) of the placement masses of the BV samples between the two sides of the tree induced by the edge. Then, we computed a PCA on the resulting balance matrix. (a) shows the result when using taxon weighting [330] in the balances calculation, while (b) shows the result without taxon weighting. Each item represents a sample, colored by its Nugent score (0 means healthy, 10 means severe illness); the Nugent score had no influence on the PCA calculations.

Figure 6.3 hence indicates how the axes of the principal components in the PCA scatter plots of Figure 6.2 can be interpreted: The first component leads to the *Lactobacillus* clade, while the second one splits this clade into *Lactobacillus iners* and *Lactobacillus crispatus*. Both plots of Figure 6.2 separate the healthy from the sick patients. However, in contrast to Edge PCA, the first component of Figure 6.2(a) does not fully distinguish between the healthy (blue) and diseased (red) samples. For yet to explore reasons, the component only takes *Lactobacillus iners* into account, while mostly ignoring *Lactobacillus crispatus*. This can be seen in Figure 6.3(a), which shows the eigenvectors of this component visualized on the reference tree. There, the path leading to the *Lactobacillus* clade does not include the branches of *Lactobacillus crispatus*, which is marked with a black arc. Including the second component however, which distinguishes between the two types of *Lactobacillus*, as

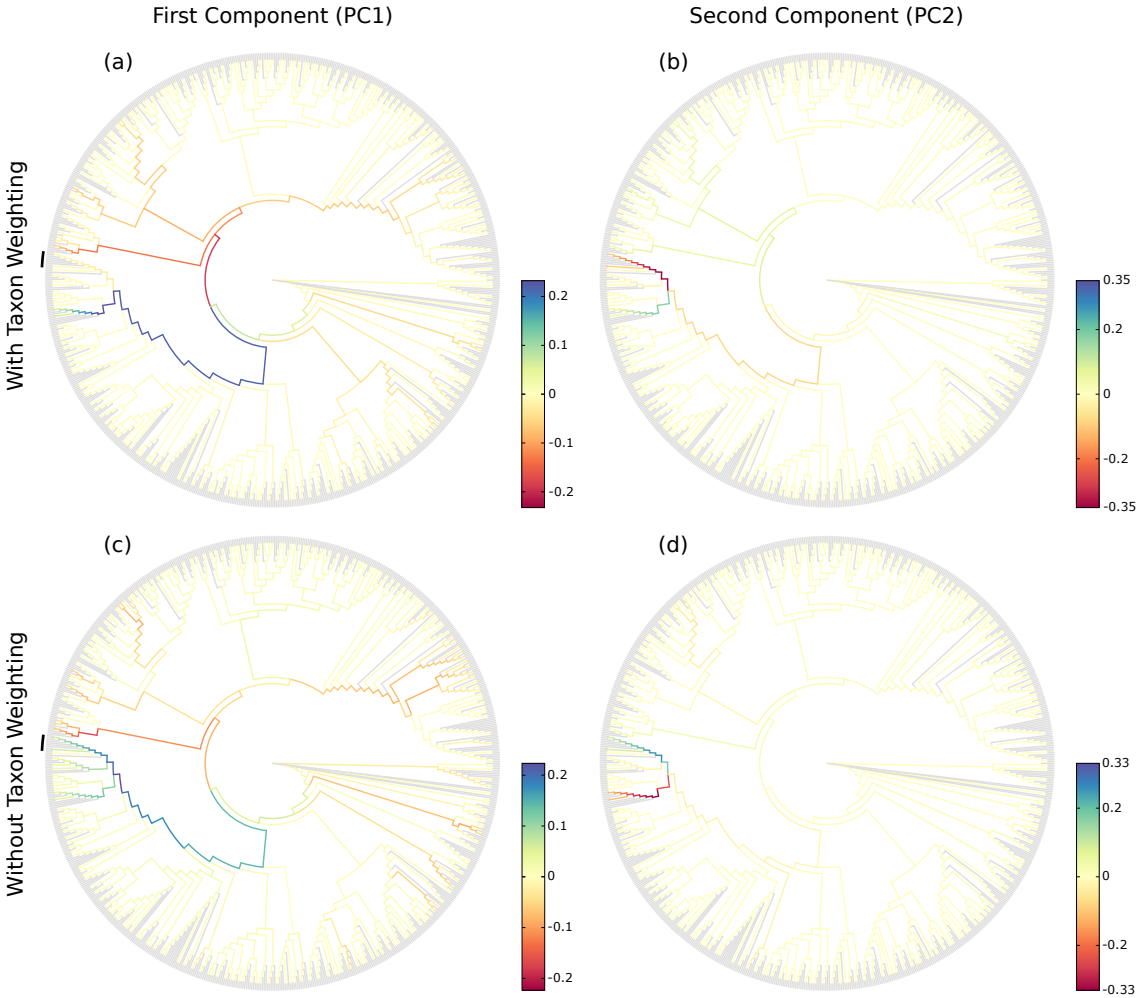

Figure 6.3: Eigenvectors of edge balance PCA of the BV dataset. The figure shows the eigenvectors of the first two principal components of PCA on the per-edge balances with and without taxon weighting, visualized on the reference tree of the BV dataset. The visualization of the components on the reference tree is analogous to the Edge PCA tree visualization as for example shown in Figure 4.4. As the data that is considered in the PCA corresponds to the edges of the tree, the resulting eigenvectors can be mapped back onto the tree, which is shown here. Each edge is colored according to the corresponding value of the first principal component in (a) and (c), and the second principal component in (b) and (d), respectively. In (a) and (c), we marked the *Lactobacillus crispatus* clade with a black arc at the left of the tree.

shown in Figure 6.3(b), yields a clear separation of the samples. On the other hand, Figure 6.2(b) exhibits closer similarities to the Edge PCA plot shown in Figure 5.3(i), in that the first component separates healthy from sick, and the second component further splits the healthy individuals apart.

Hence, the results obtained from this analysis are consistent with our previous findings, in particular with Edge PCA. The principal components separate the samples by Nugent score, with the first component mostly separating *Lactobacillus* from the rest of the tree, and the second component further distinguishing between *Lactobacillus crispatus* and *Lactobacillus iners*.

6.3.2 Edge Correlation

As mentioned in the method description (Section 6.2.1), balances could in principle be used as input to our previously presented methods, such as Edge Correlation and k-means clustering (which adequately might be called Balance k-means), in the same manner that we used imbalances in these methods. To provide an example of using balances with these methods, we show the correlation of the Nugent score with balances in Figure 6.4.

The result of this balance-based Edge Correlation with taxon weighting, shown in Figure 6.4(a), is similar to the correlation with imbalances in Figure 4.6(d): An anti-correlation with the *Lactobacillus* clade is again visible (less placement mass in this clade means higher Nugent score, that is, indicates a more severe illness), while several other clades exhibit a positive correlation with Nugent score. Hence, balance-based Edge Correlation *with* taxon weighting is consistent with our previous findings, and with the imbalance-based variant of this method.

However, artifacts might arise from the underlying mathematical framework of balances, in particular the usage of the geometric mean *without* taxon weighting: The geometric mean is *not* sensitive to singular large values, such as the high amount of placement mass on one of the *Lactobacillus* branches. It only significantly increases if multiple high values are present, such as the multitude of bacterial taxa with high abundance in diseased patients of the BV dataset [339]. This can lead to spurious results, as shown in Figure 6.4(b), where the correlation of the unweighted balances with the Nugent score yields unrealistically high negative correlations for almost all branches that have little placement mass on them (red branches).

The reason that the insensitivity of the geometric mean to the presence of singular large values leads to the majority of spuriously anti-correlated edges is as follows: If most values are small, so will be their geometric mean, even if a few very large values are also present. As can be seen in Figure 4.2 and Figure 4.5, the clades that exhibit a high anti-correlation (red) in Figure 6.4(b) have low placement mass with a low variance. Hence, the geometric mean of the masses in these clades is consistently low across samples, which means that the numerator of the log-ratio in the balances computation has little effect on the correlation. This implies that the denominator, which represents the rest of the tree, drives the anti-correlations seen in Figure 6.4(b). Women affected by BV show a presence of several different

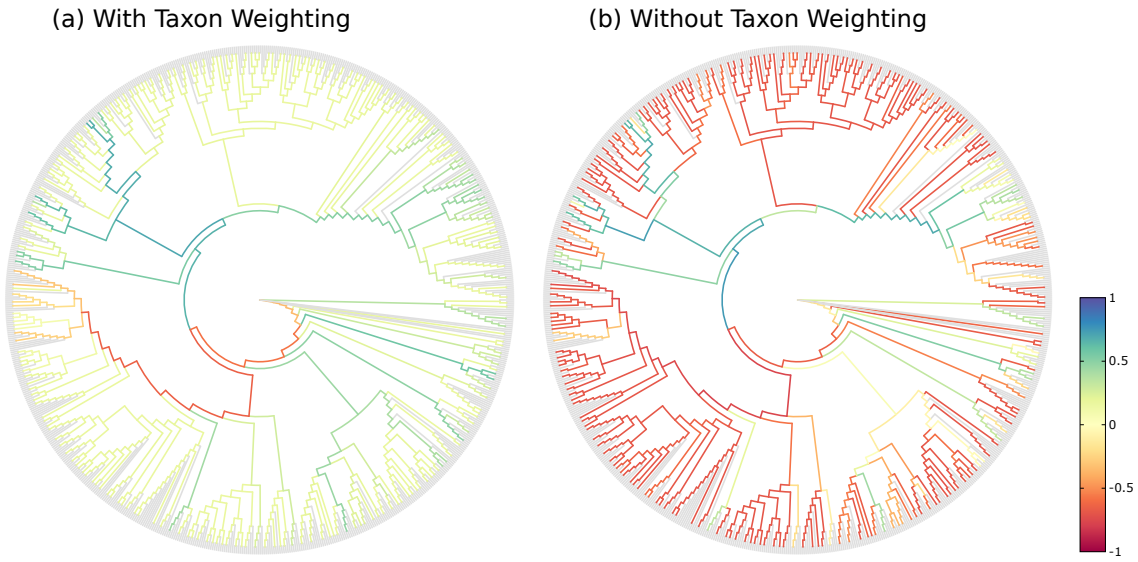

Figure 6.4: Correlation of the edge balances of the BV dataset with Nugent score. The Edge Correlation method as presented in Section 4.2.2, and for example shown in Figure 4.6, can also be conducted using balances (instead of masses or imbalances). We here show Edge Correlation using Spearman's Rank Correlation Coefficient, calculated on the per-edge balances and the Nugent score, based on the placement of the BV dataset. That is, for each edge of the tree, we calculated the balance (log-ratio of geometric means) of the placement masses of the BV samples between the two sides of the tree induced by the edge. Then, we calculated the correlation with the Nugent score of each sample, and visualized it on the tree. (a) shows the result when using taxon weighting [330] in the balances calculation, while (b) shows the result without taxon weighting.

bacterial clades, while healthy women without BV almost exclusively have high presence of one of two types of *Lactobacillus* [339]. Hence, in samples with BV (high Nugent score), there are several distinct edges that have an elevated mass, which is enough to change the geometric mean, while in samples without BV (low Nugent score), most of the mass is concentrated on a single edge of the *Lactobacillus* clade, which is not enough to significantly change the geometric mean. In consequence, the denominator of the balance at the spurious edges is consistently larger for samples with BV compared to those without BV. Thus, the balance is smaller for samples with a high Nugent score, which finally explains the observed anti-correlations. Note that despite this, there are still edges that exhibit positive correlation (blue and green), which is where the actual patterns in the data outweigh the insensitivity of the geometric mean.

Lastly, this property of insensitivity of the geometric mean implies that it *is* sensitive to taxa splitting, that is, to the number of reference sequences that a taxon or species is represented by in the reference tree (pers. comm. with A. Washburne on 2018-11-23): For example, in the context of balances of phylogenetic placements, it *does*

make a difference whether masses are focused on a single branch, or distributed across several representatives of the same species. Hence, in summary, we do not recommend to use (unweighted) balances for computations such as correlations or k-means clustering.

6.4 Summary and Outlook

In this chapter, we introduced an adaptation of the Phylogenetic ILR transform and balances [330] to phylogenetic placements. Balances are conceptually similar to edge imbalances, exhibit similar properties, and can be used for similar types of analyses. Hence, with this adaptation, we helped to bridge the methodological gap between OTU-based and placement-based approaches to analysing metagenomic data. This might lead to future development of novel methods and adaptations, with the intention of obtaining a better, more complete picture of metagenomic data, by combining the strengths of both the OTU-based and the placement-based approaches.

As balances are a transformation that yields orthogonal components (one for each node or branch of the tree), issues pertaining to the normalization of compositional data do not arise. With samples being represented as a vector of balances, numerous standard tools for data visualization, ordination, and clustering in the Euclidean space can be readily applied to phylogenetic placement data. Applying these methods to placements instead of OTUs allows for more detailed analyses, as the entire original sequence data can be used. Furthermore, using a fixed reference tree instead of one inferred from the OTUs present in a set of samples enables comparative studies across datasets.

A drawback of both balances and imbalances in the context of phylogenetic placements is that the edges leading to tips of the reference tree do not have meaningful values. This is because they measure differences between masses on the two sides of the edge (of which tip edges only have one), while ignoring the mass on the respective edge itself. For most forms of analysis, this does not pose a concerning issue, as the trees are usually large enough to have enough edges that can be used in the downstream steps. Still, we see potential for future improvement of the concepts of balances and imbalances by expanding their definition to also yield meaningful values for tip edges. A simple approach for example would be to declare all placement mass on an edge to be on the distal (away from the root) side of the edge. We however did not assess the implications of this approach for the mathematical consistency of the methods yet.

In the context of this work, balances are used as an intermediary tool in order to describe a set of samples in the context of a reference tree. In this chapter, we evaluated some basic analysis and visualization techniques in order to show that balances yield consistent results on empirical datasets compared to concepts such as edge imbalances. In the following chapter, we introduce Placement-Factorization, which uses balances as a description of clades of the reference tree.

7. Placement-Factorization

This chapter is derived from parts of the peer-reviewed open-access publication:

> **Lucas Czech** and Alexandros Stamatakis. "Scalable Methods for Analyzing and Visualizing Phylogenetic Placement of Metagenomic Samples." *PLOS ONE*, 2019, Volume 14, Issue 5, Page e0217050.

All text, tables, and figures in this chapter were created by Lucas Czech. During the revision of the above publication, reviewer Michael Robeson suggested to compare our methods of Chapters 4–5 to Phylofactorization, and thereby inspired this and the previous chapter (Chapters 6–7). Furthermore, a lot of mathematical understanding and many design choices emerged from discussion with Alex Washburne.

7.1 Background and Motivation

Phylofactorization is a method to identify edges in a phylogenetic tree that drive patterns in the composition of microbial communities [376]. An edge constitutes a separation or split of groups of taxa into the two subtrees induced by the edge. In an evolutionary context, an edge denotes a difference in (putative) traits that may have arisen along the edge. That is, an edge might describe characteristics and traits that are only present in the set of taxa on one side of the edge, but not in the set on the other side. The goal of Phylofactorization is to identify edges that are related to differences in per-sample meta-data features. To this end, it aggregates and contrasts the abundances in the subtrees (groups of taxa) induced by an edge, and evaluates how changes in environmental variables across samples are reflected in abundance changes.

The original method [376] uses a tree inferred from the OTUs that are present in the set of samples, and iteratively identifies edges that split the tree into nested subtrees which exhibit the largest predictable differences between the taxa in these subtrees. Each such edge can be interpreted as a *phylogenetic factor* (or short, *phylofactor*) for splitting the tree: Once an edge has been selected in one iteration, its induced subtrees are then considered separately in subsequent iterations. The resulting factors are hence independent of each other, which ensures orthogonality of the factors. In other words, each factor describes a different dimension in which samples differ. Furthermore, by iteratively considering subtrees of decreasing size, nested factors can be found, which correspond to relationships within a subtree that only affect the taxa in the subtree itself. The algorithm stops after a predefined number of iterations/factors, or until a stopping criterion is met.

In a typical use case, each environmental sample is represented by its OTU abundances at the tips of the tree, that is, by counts of how often each OTU is present in the sample; see Section 2.2.5 for details on OTUs of metagenomic samples. Given a per-sample meta-data feature such as the pH-value, Phylofactorization can then be employed to find edges where a change of the pH-value between samples predicts a change in OTU abundances in the two subtrees induced by the edge. For example, an increasing ph-value might indicate a relative increase in the OTU abundances in one subtree compared to another subtree. The resulting factorization can serve as a dimensionality-reduction mechanism, as an ordination and visualization tool, and as an inferential tool that can identify edges corresponding to changes in functional ecological traits [376]. We later showcase some of these applications in Section 7.3.

7.2 Methods and Implementation

We here present an adaptation of Phylofactorization to phylogenetic placements, which we call *Placement-Factorization*. We explain our adaptation following the description of the Generalized Phylogenetic Factorization (GPF) [377, 378]. The GPF is a recent generalization of Phylofactorization that also allows for types of input data other than relative OTU abundances, for example, presence/absence data. It is hence suited for a wider range of community ecological data [378]. Conceptually and algorithmically, Phylofactorization, GPF, and our Placement-Factorization, work the same; we here use the mathematical notation of GPF as a scaffold to explain our adaptation. In the following, we briefly introduce the original method, outline the necessary adaptations, and explain how to use the balances obtained from (our adaptation of) the PhILR transform (as explained in Chapter 6) in the context of Placement-Factorization.

7.2.1 Placement-Factorization

Phylofactorization can be understood as an iterative greedy graph-partitioning algorithm for a tree T [377]. In each iteration, a *winning edge* e^* is identified that splits edges of the tree into two disjoint groups R and S. To determine the winning edge, an *objective function* is maximized that expresses the intensity of the relationship between abundances and meta-data variables. We later discuss this objective function in more detail in Section 7.2.2.

Input

The input to the original Phylofactorization is an $n \times m$ data matrix X, containing $j = 1, \ldots, n$ samples, and $i = 1, \ldots, m$ species (corresponding to the OTUs at the tips of the tree). The values of the matrix can represent abundances, presence/absence data, or other data related to the species in the tree [378].

In our adaptation, we use the per-edge masses from the phylogenetic placement of the samples. That is, instead of m species representing the tips of the tree, we use an $n \times m$ data matrix X where the m columns correspond to the edges of our reference tree (for consistency of notation, we re-use and re-purpose the index m here, and transpose X compared to the original notation). This is again the mass matrix that we used in the previous methods (Chapter 4 and Chapter 5), and which we showed before in Figure 2.14(b) and Figure 6.1(a).

Lastly, Phylofactorization uses an $n \times p$ meta-data matrix Z for the n samples and p per-sample meta-data variables. This is the same meta-data matrix that we used for example in Edge Correlation (Section 4.2.2) and showed in Figure 2.14(b).

Algorithm

In analogy to the Generalized Phylofactorization [377, 378], our adapted algorithm requires three functions:

1. An *aggregation function* $A_R = A(X_{j,R},\ T)$, which aggregates (summarizes) a subset R of edges for a sample j.
2. A *contrast function* $C_{R,S} = C(A_R,\ A_S,\ T,\ e)$, which contrasts (compares) the aggregates of two disjoint subsets R and S of edges on the two sides induced by an edge e.
3. An *objective function* $\omega(C,\ Z)$ that evaluates a contrast for all samples in the context of the per-sample meta-data, in order to determine the winning edge.

We later discuss appropriate choices for these functions. For now, we assume that we are given functions that allow identifying edges whose induced subtrees exhibit predictable differences in the edge masses X driven by changes in the meta-data Z of different samples. The algorithm starts by considering the entire tree T as one large "subtree". Then, in each iteration, Phylofactorization and Placement-Factorization work as follows:

1. For each edge e that separates disjoint groups R_e and S_e of edges within the subtree that contains e:
   (a) Compute the aggregates $A_{R_e} = A(X_{j,R_e},\ T)$ and $A_{S_e} = A(X_{j,S_e},\ T)$.
   (b) Compute their contrast $C_e = C(A_{R_e},\ A_{S_e},\ T,\ e)$.

(c) Compute the objective value $\omega_e = \omega(C_e,\ Z)$.

The aggregates A_{R_e} and A_{S_e}, as well as the contrast C_e are computed separately for every sample. The value ω_e of the objective function then expresses the relationship of the contrasts of all samples with their respective meta-data values in Z.

2. Select the winning edge $e^* = \arg\max_e(\omega_e)$ that maximizes the value of the objective function.

3. Partition the subtree that contains the winning edge e^* into two disjoint subtrees, separated by e^*.

4. Repeat until a stopping criterion is met.

This closely follows the description of the algorithm in [377, 378], see there for details. The difference between the algorithms is that the groups R and S in our case consist of reference tree edges, instead of species at the tips of the OTU tree. Because of this, the aggregates of edges that lead to tip nodes are empty, meaning that we do not consider those edges as candidates in the algorithm. This is analogous to "tip edges" not having a meaningful edge imbalance, as described in Section 2.5.3. An example of the first two iterations of the algorithm is shown in Figure 7.1.

Each iteration further splits a subtree at the respective winning edge, so that after i iterations, $i + 1$ subtrees are produced. It is important to note that the winning edges of previous iterations split the tree into *disjoint* subtrees, and that in later iterations, the aggregates and contrasts induced by an edge are only computed *within* their respective subtrees. This ensures the previously mentioned orthogonality of the phylogenetic factors (winning edges), meaning that systematic dependencies between the contrasts of any two factors are eliminated, and that instead, nested relationships can be identified.

The original publication proposes a stopping criterion using a Kolmogorov-Smirnov (KS) test [237] based on test statistics of the identified phylofactors [376, 378]. Although these could be implemented for Placement-Factorization, we leave this as future work; our implementation currently runs for a given number i of iterations, and hence computes i phylofactors.

So far, we have assumed to be given the three functions required for Phylofactorization. The choice of these functions depends on the data X, the data Z, and the research question at hand. In order to be consistent and comparable with the original implementation [376], in our evaluation we used the same set of functions, namely the balances of the ILR transformation as explained in Chapter 6 for aggregating and contrasting subtrees, and an objective function based on Generalized Linear Models (GLMs), which we explain in the following, see Section 7.2.2 and Section 7.2.3.

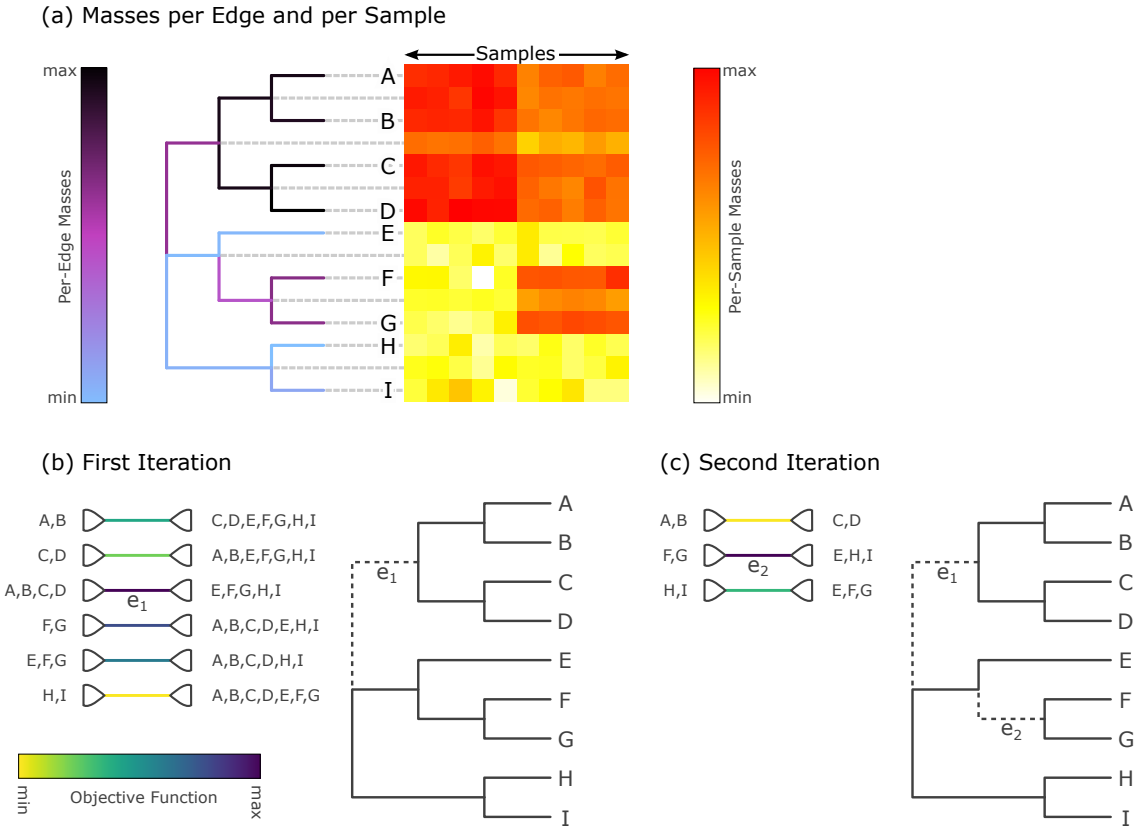

Figure 7.1: Input data and first two iterations of Placement-Factorization. The figure resembles Figure 2 of Washburne et al. (2017) [376]. It shows the adaptation of concepts from Phylofactorization to phylogenetic placement data.
(a) The input data is a set of samples with placement masses on each edge of the tree. The tree is colorized by the total mass across all samples, that is, by the row sums of the heat map. The heat map then shows the detailed mass per edge (rows) and per sample (columns), and hence is an example of the mass matrix of Figure 2.14(b). Note that the heat map also contains rows for each inner edge of the tree, as phylogenetic placement also considers these edges. This is different from OTU abundance heat maps that only have entries for the tips of the tree. We show a further example of this visualization for empirical data in Figure 4.2.
(b) In the first iteration, the objective function for all inner edges is evaluated. Here, edge e_1 is the winning edge that maximizes the objective function, which separates the clade (A, B, C, D) from the rest of the tree.
(c) In the second iteration, only the contrasts within the two subtrees are calculated, but not across the winning edges of previous iterations (here, e_1). That is, the winning edge e_2 maximizes the objective function that contrasts clade (F, G) with clade (E, H, I), but does not consider the edges in the subtree (A, B, C, D). Note that in our adaption, edges that lead to a tree tip are not considered as potential factors.

Output

The main output of the algorithm is the list of winning edges, that is, of the phylogenetic factors that have been identified. Furthermore, one can store detailled tables with the balances per sample for all factors, the values of the objective function at each edge for all factors, as well as much more intermediary data of the algorithm. We later show examples of how these outputs can be used for analysis and visualization purposes in Section 7.3.

7.2.2 Objective Function

Phylofactorization requires an objective function $\omega(C_e,\ Z)$ that quantifies the relationship between C_e and Z for a given edge e, where C_e are the contrasts between the two subtrees induced by e for all samples (for example, the balances), and Z are the per-sample meta-data variables. That is, both C_e and Z have size n, the number of samples, with Z potentially containing multiple columns (one for each meta-data feature). In order to identify the winning edge e^* of an iteration (the *phylofactor*), the function is evaluated for all edges, and the edge maximizing ω is selected. The choice of the objective function depends on the research question at hand; see Washburne et al. (2018) [377] for a thorough discussion.

Our implementation is as general as the original Phylofactorization [376], in that it allows for an arbitrary objective function. For simplicity, and in line with the original publication, we here focus on functions that treat the meta-data variables Z as independent variables and the contrasts C_e as dependent variables whose relationship with Z is assessed, for instance, via a predictive model. Then, the selected phylogenetic factors correspond to edges where a change in Z most strongly predicts a change in C_e across the samples, that is, where the effect of the (independent) meta-data variables on the (dependent) underlying data (e. g., per-clade abundances) is most pronounced. We show an example in Figure 7.2.

Generalized Linear Models

A powerful approach is to model the relationship between C_e and Z via linear regression, that is, we assess how well Z can predict C_e. In the simple one-dimensional case, this can be thought of as fitting a line through a scatter plot of the meta-data feature on the x-axis and the contrasts on the y-axis, where each point represents one sample, c. f. Figure 7.2. This concept is generalized via the Generalized Linear Model (GLM) [5, 248, 270].

We introduce GLMs in more detail in Section 7.2.3. In short, GLMs allow to predict a single (response) variable using multiple input (explanatory) variables. Typically, the response variable is assumed to follow any distribution from the exponential family (normal, exponential, Poisson, Binomial, etc.), which is given for balances as used here. In contrast to this, the explanatory variables (the meta-data features) are assumed to have a linear relationship with the response. Note that this mathematical restriction of the model does not mean that only meta-data features can be used that behave linearly; transformations and interactions of the features basically allow

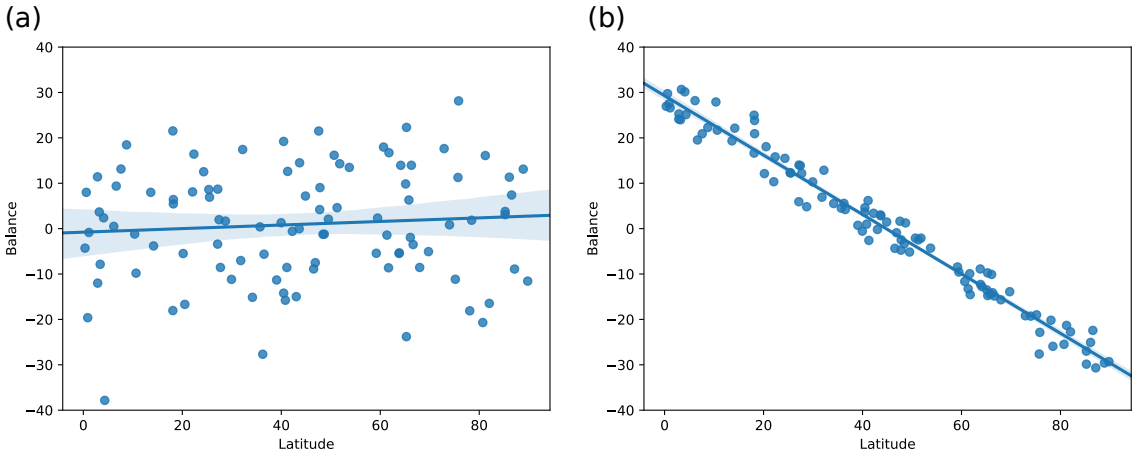

Figure 7.2: Exemplary relationships between independent and dependent variables. The figure shows simulated examples of the relationship between the latitude where some (hypothetical) oceanic samples were taken from and the balance of two edges of a reference tree for these samples. In (a), there is no apparent relationship between the balances at this edge and the latitudes. This means that the clades of the tree that are split by the edge do not separate samples by latitude, and hence contain species whose abundances are not affected by latidute. In (b) on the other hand, the balance of each sample exhibits a strong linear relationship with its latidute. This indicates that the corresponding edge across which the balance was calculated is a good candidate for a factor.

for arbitrary types of data. For example, categorical variables such as the body site where a sample was taken from can be transformed into so-called dummy variables that fulfill the requirements.

Once the model parameters of the GLM have been estimated, that is, once it has been fit to the data via some optimization algorithm, we need to evaluate the GLM for the purposes of Phylofactorization. We are interested in a value for ω that expresses how well the meta-data variables explain the balances. To this end, Phylofactorization and our adaptation thereof use the difference between the null deviance of the balances and the deviance obtained from the GLM. This difference expresses how much better the model explains the balances than just predicting them from their mean. For details on the usage of GLMs for Phylofactorization, see Washburne et al. (2017) [376].

Usage in Phylofactorization

Predictive models such as the GLM expect the response variable (that is, the predicted values; here, the contrasts) to have certain statistical properties. In particular, linear models assume the deviation of response from the predicted value to be normally distributed. The ILR transform for compositional data has been proven to behave asymptotically normal [102, 283], which allows their application within standard multivariate methods, and within GLMs as presented here.

Lastly, we note that depending on the research question, other objective functions can be used, see Washburne et al. (2017) [376] for some examples. For instance, simple test statistics such as the variation in C_e explained by regression on Z can be used. Furthermore, instead of predicting contrasts from meta-data, one could be interested in the opposite, that is, predicting a meta-data variable given the per-sample contrasts. In this case, the maximization of the objective function yields edges that best predict a certain feature of the data; this is suitable for identifying clades that can serve as a bio-indicator, i. e., in order to predict diseases. Using GLMs for this allows to model any type of meta-data variable; for example, the binary information encoded in presence/absence data can be predicted using logistic regression. While our implementation supports all those use cases, they have been explored and discussed before [376, 378]. For the sake of simplicity, we hence focus on linear (gaussian) modeling of $C_e \sim Z$, that is, predicting balances from meta-data.

7.2.3 Generalized Linear Models

We now take a short digression to introduce the Generalized Linear Model (GLM) [248, 270]; for a more detailled explanation see McCullagh and Nelder (1989) [248] and Agresti (2018) [5]. Note that the abbriviation GLM is sometimes also used for the *general linear model*, as opposed to the *generalized* linear model that we discuss here, which are distinct concepts. Hence, sometimes, the latter is abbriviated as GLIM instead [248]. The GLM is a generalization of linear regression that allows (a) for the response variable to have an arbitrary distribution (instead of a normal distribution, which is implicitly assumed in linear regression), and (b) for an arbitrary function of the response variable—called the *link function*—to vary linearly with the predicted values (instead of the response variable itself varying linearly). In the context of the GLM, the independent variables (e. g., the meta-data Z) are called the *predictor variables* or *explanatory variables*, while the dependent variable (e. g., the contrasts C_e) is called the *response variable*. The goal is then to best predict the (single) response variable from the (multiple) explanatory variables.

Model Components

We here use the standard notation of X being the predictor of size $n \times p$ and Y being the response of size n, with n the number of data points (e. g., samples) and p the number of individual predictor variables (e. g., meta-data features). Then, a GLM is described by three components [248]: a *random component*, a *systematic component*, and a *link component*.

The random component specifies the probability distribution of Y (conditioned on X); it is also called the *error* or *noise model*. For example, in linear regression, we assume Y to be generated from a normal distribution, while in logistic regression, the distribution is assumed to be binomial.

The systematic component specifies the linear combination η of the predictors X using the (unknown) parameters β:

$$\eta \;:=\; \beta^{\intercal} X \;=\; \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \tag{7.1}$$

This is the "linear" part of the model, which is analogous to normal linear regression. Note that, for conciseness, in the matrix notation $\beta^\intercal X$ we assue X to have an implicit column of 1's to accommodate the intercept parameter β_0.

The link component connects the random and the systematic component via a smooth and invertible *link function* g. In particular, the link function provides a connection between the expected value $\mathbb{E}$ of Y and the linear combination η of X:

$$g(\mathbb{E}(Y|X)) \;=\; \eta \tag{7.2}$$

That is, instead of directly predicting (the expected values of) Y from X, the link function transforms the expectation of the response variable to the linear predictor. A GLM can hence be thought of as a non-linear regression model for the response variable. If the distribution of Y (the random component) is assumed to be a member of the exponential family of distributions (normal, exponential, Poisson, Binomial, etc.), there is a *canonical* link function for each member of the family. For example, for the normal distribution, g is the identity function, while for a binomial distribution, g is often set to the logit (log-odds) function.

The expected value of Y is usually assumed to be the mean value μ of the distribution of Y, meaning that $\mathbb{E}(Y|X) = \mu$. The full model is then given by:

$$\mathbb{E}(Y|X) \;=\; \mu \;=\; g^{-1}(\eta) \;=\; g^{-1}(\beta^\intercal X) \tag{7.3}$$

The choice of distribution and link is typically informed by the type of data Y being modelled: For example, if Y are continuous data (that are assumed to respond linearly to changes in η), the normal distribution and the identity link are well suited; if Y are categorical data or counts ("yes"/"no" choices), a binomial distribution and a logit link can be used; and if Y are counts of occurrences in a fixed amount of time, the Poisson distribution and a log link are typical choices.

In short, a GLM uses a transformed (via the link function) linear combination of the explanatory variables X (the systematic component) to predict the response variable Y, assuming some model of error (the random component).

Different Types of Data

As described above, the GLM uses the link function to allow for non-linear behaviour of the response variable Y. However, as evident from Equation 7.1, the predictor variables X are linearily combined to form η. Note that this does not necessarily impose a linear relationship of X with Y: The predictor variables can be arbitrarily transformed prior to their usage in the GLM, for example, by taking their logarithm, or using their reciprocal values. By (multiplicatively) combining variables, it is also possible to encode *interactions* of variables. In fact, it is possible (and common) to use multiple transformations and interactions of the same underlying predictor variables simulationsly in a GLM.

Furthermore, the predictor variables do not have to be numerical. For instance, binary variables (e. g., the presence/absence of species) or categorical variables (e. g., the body site that a sample was taken from) can be transformed into dummy variables that encode different outcomes or categories as a vector of 0's and 1's. Such variables are also often called *factor* variables.

This flexibility allows to use GLMs to model many types of relationships between a given response variable and multiple explanatory variables.

Fitting the Model

For given data X and Y, and a link function g, the model is fit to the data by estimating the parameters β. This optimization is typically conducted via a maximum likelihood estimation, using for example the Iterative Reweighted Least Squares (IRLS) method [43]. The IRLS method is an instance of Newton's method [396] and minimizes the squared differences δ^2 between the values of the response variable Y and the *predicted values* $\hat{Y}$:

$$\hat{Y} = g^{-1}(\beta^\intercal X) \tag{7.4}$$

The differences between the values of the response variable Y and the predicted values $\hat{Y}$ are called the *residuals* δ of the regression for each data point i:

$$\delta_i = Y_i - \hat{Y}_i \tag{7.5}$$

As the GLM is optimized to minimize the (squared) difference of these values, both the sum and the mean of the residuals are equal to zero. This procedure is analogous to minimzing the sum of squares in standard linear regression models.

Assessing the Fitness of the Model

In the context of Phylofactorization and Placement-Factorization, the GLM is mostly used as a intermediary tool: We are not so much interested in using it to actually predict response values from a given set of predictor values, but instead want to know how predictable the data is in general. In other words, we want to assess *how well* the model is able fit the data.

The fitness of a model m can be assessed by comparing it to the *saturated model* and the *null model*. On the one hand, the saturated model is an abstract model that has as many estimated parameters as data points n, and hence (by definition) obtains a perfect fit of the data. On the other hand, the null model only has one free parameter β_0, the *intercept*, which is optimized by simply predicting the mean $\beta_0 = \overline{Y}$ for all values. We exemplify and summarize these concepts in Figure 7.3. Note that a comparison of these models is valid, as they are nested.

We can then calcuate the *(residual) deviance* D_m of the model m using the log-likelihood $\mathcal{L}_s^*$ of the saturated model and the log-likelihood $\mathcal{L}_m^*$ of the fitted model

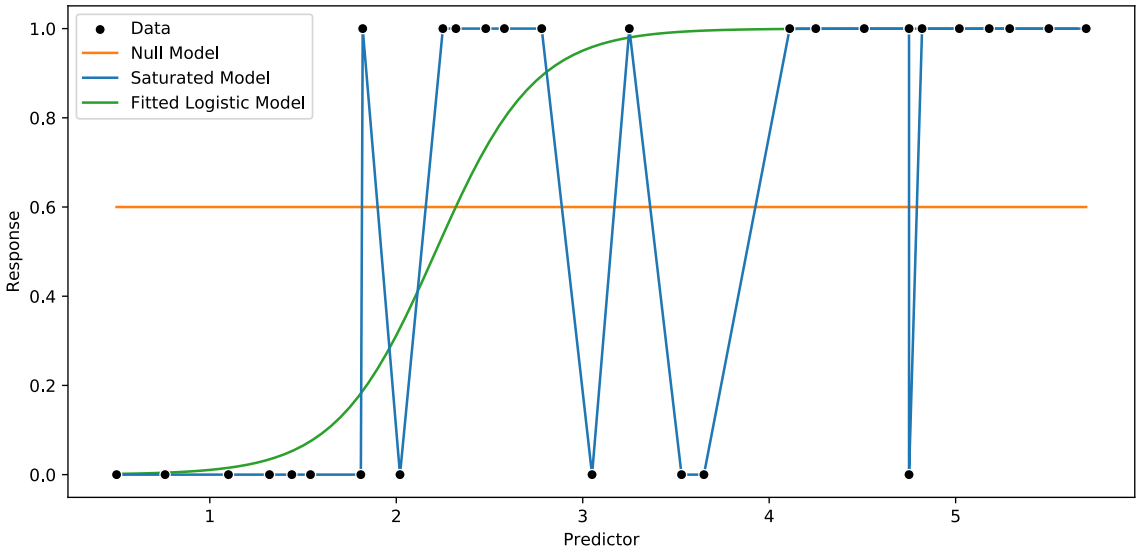

Figure 7.3: Example of logistic regression. The data represents a continous predictor (explanatory) variable and a binary response variable; for example, the number of days a student spent learning for an exam, and whether the student passed (1) or failed (0) the exam. The null model only has a single parameter β_0 to predict the outcome; the optimum is hence to always predict the mean of the data. The saturated model on the other hand is able to predict each datum correctly, at the expense of model simplicity. Because of the binary response variable, a binomial logistic regression is a good model for the data: it attempts to fit the data using two parameters β_0 and β_1.

m. By definition, the likelihood of the saturated model is exactly 1 (and hence, its log-likelihood is 0). Thus, the deviance of a model is simply an expression of its likelihood:

$$D_m \;:=\; 2 \cdot (\mathcal{L}_s^* - \mathcal{L}_m^*) \;=\; -2 \cdot \mathcal{L}_m^* \tag{7.6}$$

Consequently, the residual deviance is always larger than or equal to zero, being zero only for a perfect fit. The deviance of a model is an analogous generalization of the residual sum of squares for a linear model. That is, in the case of linear regression, which assumes normally distributed errors with constant variance, the residual deviance is calculated as:

$$D_m \;=\; \sum_{i=1}^{n} \delta_i^2 \;=\; \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2 \tag{7.7}$$

In order to evaluate the magnitude of the deviance, it can be compared to the *null deviance* D_n, that is, the residual deviance of the null model. The null model can be understood as the worst model, being fitted without any predictors. The null deviance hence serves as a benchmark of how much the model m in question improves the fitness by taking the predictors X into account. The null model only uses the

intercept β_0 to predict the values Y. Hence, in linear regression, the residuals of the null model are the differences from the mean $\overline{Y}$, and its deviance is equivalent to the total sum of squares of the data Y:

$$D_n \;=\; \sum_{i=1}^{n} \left(Y_i - \overline{Y}\right)^2 \tag{7.8}$$

Often, the model comparison is done via the R^2 statistic, which is a generalization of the determination coefficient in linear regression, and calculated as $R^2 \;:=\; 1 - \frac{D_m}{D_n}$. However, we here follow Washburne et al. (2017) [376] and use the differences of deviances for the comparison instead. That is, for our purposes, we define the fitness ω of a model m as:

$$\omega \;:=\; D_n - D_m \tag{7.9}$$

For the GLM, the difference in deviances is equivalent to the difference in log-likelihood between the null model and the fitted model. As the null model is nested in the model m, its likelihood is less or equal to the likelihood of m, and hence its deviance greater or equal to the deviance of m. Consequently, $\omega \geq 0$, with $\omega = 0$ only if the model m is equivalent to the null model. Intuitively, ω measures how much better the model m is for predicting the response variable Y using the explanatory variables X, than just predicting the mean value $\overline{Y}$. This value hence serves as our measure of "how well" the model is able to predict the data, and is the objective function that is used in Phylofactorization and in Placement-Factorization.

For example, again, for linear regression with normally distributed errors and constant variance, this can be computed by summing squared differences:

$$\omega \;=\; \sum_{i=1}^{n} \left(Y_i - \overline{Y}\right)^2 \;-\; \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2 \tag{7.10}$$

with $\overline{Y}$ being the mean, and $\hat{Y}_i$ being the predicted values. In practice, these terms can be divided by the number of samples n without altering their suitability as an objective function to be maximized in Placement-Factorization. In that case, the objective function becomes equal to the difference of the variance of the data and the mean squared error of the prediction. This suggests an alternative interpretation of ω as being a measure that expresses how much of the variance in the data can be explained by the predictor variables.

Usage of GLMs with Balances and in Placement-Factorization

As stated before, the ILR transform for compositional data, and hence balances as defined in Section 6.2.1, behave asymptotically normal [102, 283]. This allows to use them as the response variable in a GLM, using a normal (Gaussian) error distribution with an identiy link function. In Placement-Factorization, we predict

balances using the per-sample meta-data as explanatory variables. That is, a GLM is estimated separately for each edge of the tree, predicting the respective balances of all samples at that edge. By maximizing the objective function ω across all edges, we hence are looking for the edge where the balances are most predictable from their meta-data. As we assume a normal distribution for the balances, Equation 7.10 describes the compuation that we use to evaluate the objective function.

In Section 6.3.2, we described several pitfalls of balances that are due to the mathematical properties of geometric means, such as its insensitivity to singular large values. This prohibits for example to use balances *without* taxon weighting as input for our Edge Correlation (Section 4.2.2). However, when used in GLMs as described here, these issues do not arise: The winning edge is chosen to maximize the difference between the null deviance and the deviance of the model. That difference is small for clades with almost no mass (such as the ones affected by the geometric mean issues), so that the value of the objective function for such edges is lower than for edges with more mass. Hence, the factorization does not incorrectly identify these low-abundance clades as potential factors. This can also be seen in our visualization of the values of the objective function, as for example shown later in Figure 7.5.

7.2.4 Method Comparison

In summary, Phylofactorization and our adaptation Placement-Factorization identify edges of the phylogeny that exhibit a predictable relationship between changes in meta-data variables and abundance changes in the subtrees induced by these edges. Our adaptation can be understood as a generalization of the original method [376, 378], where masses/counts can be placed along the edges of the tree, instead of just at its tips.

While the original method uses abundances of taxa/OTUs per sample on a tree inferred from the OTU sequences, we use the placement masses on a fixed Reference Tree (RT). For many use cases, this has several advantages: The RT can be inferred from reference sequences that are longer than typical metagenomic reads used for OTU-based analysis, such as the whole 16S or 18S regions of the genome; hence, phylogenetic inference will be more reliable. Furthermore, the size of the RT can be chosen as needed, for example via our Phylogenetic Automatic (Reference) Tree (PhAT) method as presented in Chapter 3, instead of having to use the number of OTUs that result from the clustering and preprocessing steps. This also eliminates the need for the (mostly arbitrary) OTU cutoff step that is common to many metagenomic analyses, where OTUs with low abundance or low spread across samples are filtered out in order to keep the number of OTUs manageable. That is, with our approach, all sequences in a dataset can be placed and analyzed.

Another advantage of a fixed RT is the availability of taxonomic annotation for the reference sequences. Often, in metabarcoding studies, the environmental sequences are anonymous and might not be closely related to any known species [174, 230, 350], which can hinder common taxonomic assignment methods [185]. Placing the sequences onto an RT with known taxonomic labels allows to easily interpret results

within a given taxonomic framework. Using a taxonomically constrained RT can further improve interpretability. Lastly, using a fixed reference tree better allows to conduct cross- or meta-studies that compare samples from different sources, or to easily run analyses for samples that were added to the dataset later on. Using a fixed tree means that the context of interpretation remains unaltered. This is not easily possible with trees inferred from OTUs, as those change depending on the input sequences.

For further details on Phylofactorization, in particular the mathematical properties of the method, we refer to Washburne et al. (2019) [378], which also covers different objective functions, elaborates on stopping criteria, and compares the method to other phylogenetic methods for analyzing ecological data. Compared to other tools and methods that use the phylogeny as a guide or scaffold for analyzing microbial data, both, the original Phylofactorization as well as our adaptation allow for a direct interpretation of the results in terms of the edges of the tree, while avoiding nested dependencies between overlapping subtrees and circumventing issues associated with the compositional nature of the data.

7.3 Evaluation and Results

We here present results from *Placement-Factorization*. We compare these to the results from our other methods (as described in previous chapters), as well as to the original Phylofactorization. For comparability with the original method, we solely use balances for aggregating and contrasting, and an objective function that maximizes the difference between the null deviance and the deviance obtained by a Generalized Linear Model (GLM). Other choices of functions for Phylofactorization have been explored in Washburne et al. (2017) [376] and Washburne et al. (2019) [378], see there for details. The exploration of their effect on Placement-Factorization is left as future work, although based on the consistency of our results with the original method, we conjecture that they behave according to the findings of the original publications.

Furthermore, we note that our implementation supports taxon weighting as proposed in the Phylogenetic ILR transform [330] and described in Section 6.2.2. Taxon weighting is however not (yet) supported by the original Phylofactorization [376]. We found this weighting scheme to be a natural and valuable addition in the balances computation that yields results closer to those obtained with edge imbalances. We suspect that this is because the weighting scheme can alleviate the issues of the geometric mean that we observed in Section 6.3.2.

7.3.1 BV dataset

We analyzed the Bacterial Vaginosis (BV) dataset [339] with our Placement-Factorization with and without taxon weighting, using balances for aggregating and contrasting, and GLMs for the objective function. See Appendix B.1 for details on the dataset. As GLMs support multiple predictors at the same time, we used all three available meta-data features of the dataset simultaneously for the regression, that

is, Nugent score [274], Amsel's criteria [9], and the pH-value of the samples. We also tested with only the Nugent score to be consistent with our previous analyses, and to asses the robustness of the method with respect to the specific choice of meta-data features. We observe only minor differences in the ordering of the identified factors, that is, which clades were "winning" in which iteration. Hence, we here focus on the results obtained with all three meta-data features taken into account.

Comparison to Phylofactorization

For comparison with the original method, we clustered the dataset into OTUs using two different OTU clustering methods, VSEARCH [306] and SWARM [228, 229], and inferred two trees from these OTU clusterings. We used two distinct OTU clustering methods to asses how they affect factorization; see Appendix B.1 for details on these preprocessing steps. We then conducted an analysis of both trees with the original Phylofactorization, again using balances and GLMs. We compare the results of Placement-Factorization to our previous analyses of the data as well as to the original Phylofactorization on the two alternative OTU trees.

In Table 7.1, we compare the clades found by the two Phylofactorization variants (with VSEARCH and with SWARM) to the clades found by Placement-Factorization *without* taxon weighting. As shown in Figure 4.2 and in the original study of the dataset [339], there are multiple different taxa that are associated with Bacterial Vaginosis. That is, there are several clades or branches of the reference tree where the placement mass differs between healthy and sick patients. It is thus expected that a Phylofactorization of these data exhibits some variation in the clades being found, depending on the preprocessing and exact settings being used. Still, Table 7.1 shows that—apart from ordering—the factored clades are mostly consistent across variants, and consistent with previous findings. All of the taxa found by the SWARM-based Phylofactorization and by our Placement-Factorization, as well as all taxa except some of the *Streptococcus* found as part of the first factor of the VSEARCH-based Phylofactorization, were already shown to play important roles for this dataset [339]. The inclusion of *Streptococcus* in the VSEARCH variant is due to an inner edge that has a slightly higher value of the objective function than the actually more relevant edges leading to the *Lactobacillus* clade. We observed a similar behavior of large clades being split with our implementation when using taxon weights, as shown later in Figure 7.5. Lastly, the normalized mutual information [370] between the three variants ranges between 71% and 81%, further showing that they mostly find the same clades.

Moreover, we visualize the clades found by Placement-Factorization in Figure 7.4, which correspond to the clades listed in Table 7.1. The clades are consistent with the findings of the original study of the dataset [339]: Healthy women without BV exhibit high abundances of *Lactobacillus*, while women affected by BV have a more diverse vaginal microbiome, containing multiple different bacterial taxa. Hence, the first factor represents the most prominent split of the data into healthy vs. diseased, based on the presence of *Lactobacillus*. Differences within the healthy samples are then further distinguished in factors five and ten, which further split parts of the

Table 7.1: First ten factors of the BV dataset found by Phylofactorization. In this table, we compare our Placement-Factorization of the BV dataset to the results from the original Phylofactorization. The table lists the taxa in the clades that were split by the first 10 factors in each variant, using VSEARCH and SWARM for the OTU clustering of the data. As the original implementation does not support taxon weighting, we also do not use it here. See also Figure 7.4 for a visualization of the clades found by our adaptation (right-most column).

	Original (VSEARCH)	Original (SWARM)	Placement-Factorization
1	Lactobacillus crispatus, Lactobacillus jensenii, Lactobacillus iners, Lactobacillus coleohominis, Lactobacillus gasseri, Lactobacillus vaginalis, Streptococcus agalactiae, Streptococcus anginosus, Streptococcus gallolyticus, Streptococcus oralis, Aerococcus christensenii	Sneathia sanguinegens, Leptotrichia amnionii	Lactobacillus crispatus, Lactobacillus jensenii, Lactobacillus kalixensis
2	Lactobacillus crispatus	Lactobacillus crispatus	Sneathia sanguinegens, Leptotrichia amnionii
3	Gardnerella vaginalis	Gardnerella vaginalis	Gardnerella vaginalis
4	Leptotrichia amnionii	Atopobium vaginae	Megasphaera
5	Megasphaera	Megasphaera	Lactobacillus crispatus
6	Atopobium vaginae	Eggerthella	Eggerthella
7	Eggerthella	Prevotella bivia, Prevotella amnii	Prevotella timonensis, Prevotella buccalis
8	Sneathia sanguinegens	Prevotella timonensis	Prevotella bivia, Prevotella amnii
9	Prevotella timonensis	BVAB2	Atopobium vaginae
10	Lactobacillus jensenii	Lactobacillus jensenii	Lactobacillus iners

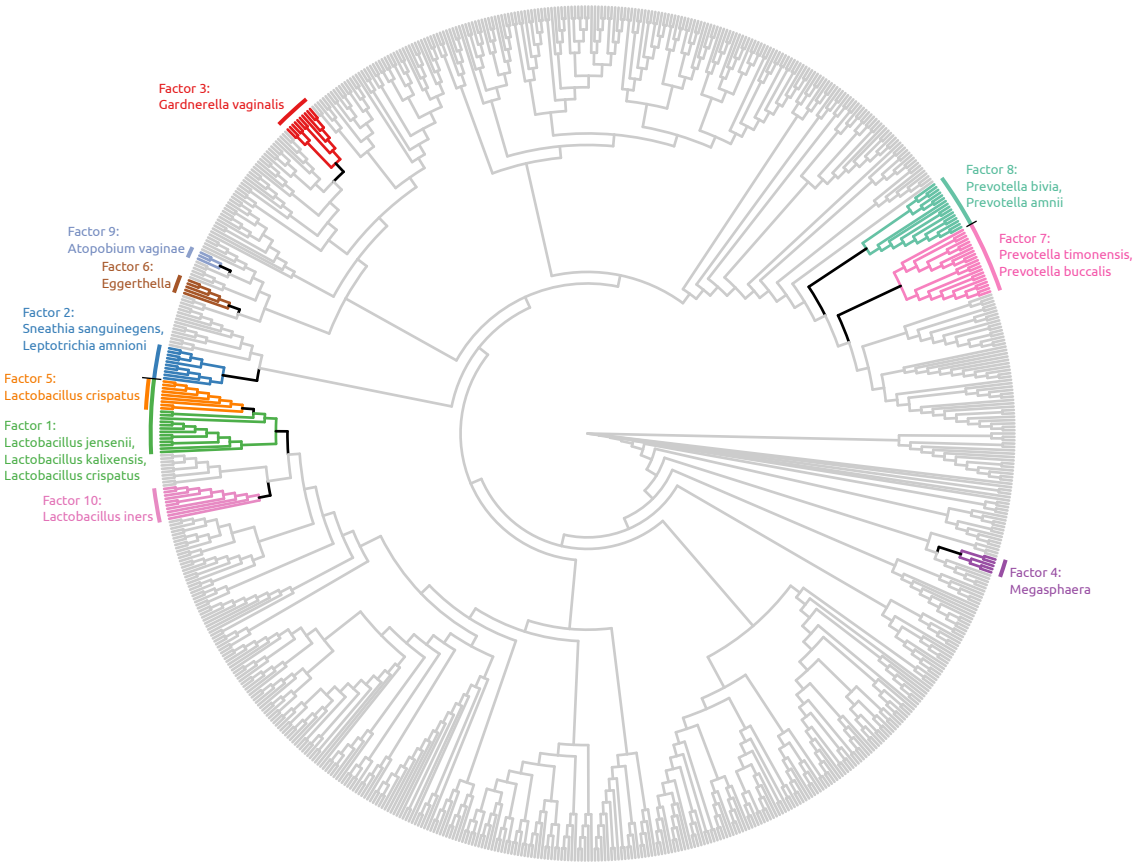

Figure 7.4: Visualization of the first ten factors of the BV dataset. Here, we show the first ten factors found by Placement-Factorization without taxon weighting on the BV dataset. The black edges are the winning edges of each iteration, which split the tree into several subtrees. For simplicity, we only colored the clades leading away from the (arbitrarily placed) root, while leaving the paraphyletic "remainder" clade in gray. Note that factor 5 is nested in factor 1, that is, it further splits the branches within the first factor, thus separating *Lactobacillus crispatus* from *Lactobacillus jensenii* and *Lactobacillus kalixensis*. See Table 7.1 for a comparison of the clades separated by each factor to the factors found by the original Phylofactorization. Furthermore, see Figure 7.7(b) for an ordination of the first two factors, showing how these factors separate the samples in the dataset.

Lactobacillus clade. The remaining factors split away clades that further separate the diseased samples from each other, based on several distinct bacterial taxa. All clades that are found by these factors where shown in the original study to be associated with BV [339], meaning that Placement-Factorization on this dataset yields results that are consistent with previous analyses.

These outcomes show that our results are consistent with the existing Phylofactorization, in that similar clades are split from the tree, albeit with some variation in the order by which clades are selected. The clades being split are also consistent with previous analyses of the dataset [339], as all taxa found by the first ten

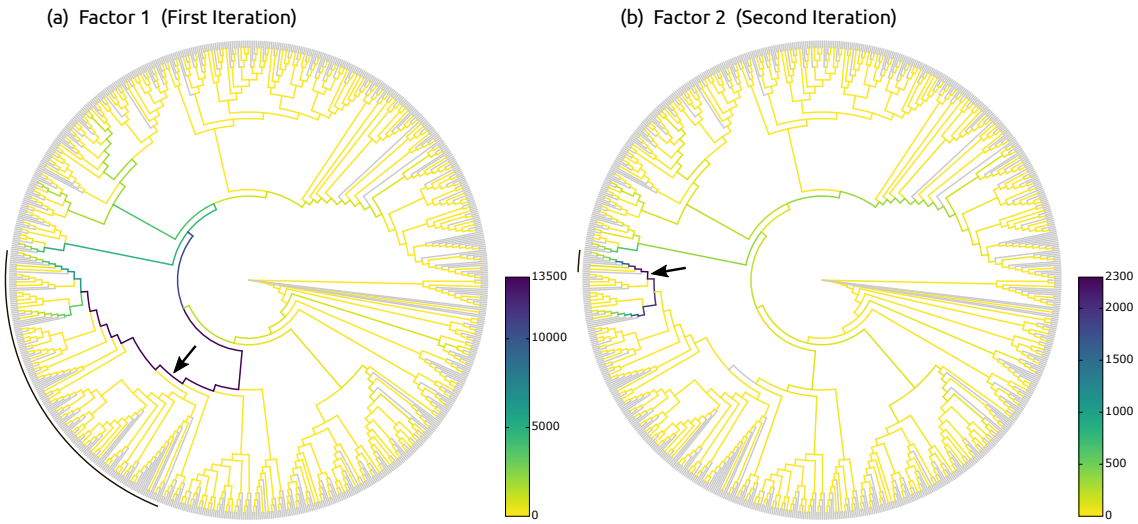

Figure 7.5: Objective function values of Placement-Factorization with taxon weighting of the BV dataset. Here, we show the values of the objective function for each inner edge, for the first two factors found by Placement-Factorization *with* taxon weights of the BV dataset. The winning edge of each iteration is marked by a black arrow, and the resulting clade is marked with a black arc. See also Figure 7.6 for the version of this visualization *without* taxon weights.

factors of Placement-Factorization were also found to be relevant in the context of Bacterial Vaginosis in [339]. However, the VSEARCH-based Phylofactorization is the only evaluated variant that split the *Lactobacillus* clade in the first factor and further *Lactobacillus crispatus* from *Lactobacillus iners* in the second factor. The SWARM-based variant and our Placement-Factorization without taxon weighting also identified these clades, but not in the first two iterations.

Visualization of the Objective Function

Above, we have compared Placement-Factorizaiton *without* taxon weighting to the original Phylofactorization, which currently does not support any taxon weighting schemes. When using taxon weighting on the other hand, Placement-Factorization also finds the two *Lactobacillus* clades in the first two factors, and is hence more consistent with existing analyses of the dataset. However, due to small differences in the value of the objective function, the winning edge of the first iteration is chosen to be relatively basal in the tree, meaning that a large clade is factored out. We observed a similar behavior with the VSEARCH-based Phylofactorization, as can be seen by the long list of taxa of the first factor in Table 7.1. In order to identify the provenance of this effect and to correctly interpret the factors, we developed a novel visualization of the results: In Figure 7.5, we show the reference tree, where each edge is colored by the value of the objective function at that edge.

This novel type of visualization helps to assess the uncertainty involved in identifying the winning edge of a specific iteration: The objective function of the first

iteration in Figure 7.5(a) for instance yields high values for the path towards the *Lactobacillus* clade, consistent with previous findings. Random variability however leads to the first iteration splitting a larger clade than expected when using taxon weighting; the winning edge is marked with an arrow. This obfuscates the fact that this factor is mostly concerning the *Lactobacillus* clade, and not so much the remaining taxa in that clade. The clade includes many branches and taxa with a low value of the objective function (yellow branches), which are branches with low placement mass that do not contribute much to this factor. There is however a path of comparably high values of the objective function (dark branches) that leads down to the *Lactobacillus* clade. This indicates that there are several 'good' candidate edges for distinguishing patients by their health status, and that the smaller *Lactobacillus* clade is the actual clade of interest in this factor. The winning edge just happened to have a slightly higher value than other edges on this path. The visualization thus aids interpretation of the factors, helps to understand why a particular edge was chosen in an iteration, and allows to identify the parts of a factored clade that are most relevant to the factor.

To address this issue of random variability, a proper statistical test of the significance of each winning edge compared to the other edges evaluated in the iteration could be employed. This is connected to the idea of *confidence regions* of each factor on the tree, as presented in Washburne et al. (2019) [378], which we discuss later. Note that in the second iteration in Figure 7.5(b), the tree clearly shows the distinction between the two relevant clades of *Lactobacillus* again, consistent with previous findings. Hence, even with the first factor splitting a relatively large clade, the second factor correctly identified the relevant edges within this large clade.

Lastly, we show the same type of visualization for Placement-Factorization *without* taxon weighting in Figure 7.6. The figure hence again corresponds to the analyses presented above in Table 7.1 and Figure 7.4.

Apart from the guidance for interpreting the factors, the figure also reveals another aspect of Phylofactorization: By comparing the values of the objective function on the tree between subsequent iterations, one can observe the effect of "factoring out" an edge. Due to the nature of comparing the two sides induced by an edge, high values of the objective function usually propagate across several connected edges, e. g., the region of dark branches around the marked edge in Figure 7.6(a). Once a factor has been split from the tree, the values for the whole path drop, which can be seen by comparing Figure 7.6(a) and (b), where the region around the gray arrow has much lower values of the objective function. This behavior can consistently be observed in the other subfigures as well. The figure hence shows that factoring out an edge actually removes nested depenendices between factors, as expected.

Further Assessment

To further assess how the samples are split by individual factors, we used the balances at each iteration/factor as an ordination of the data, as suggested in Washburne et al. (2017) [376], which we show in Figure 7.7. These plots reveal that the splitting into

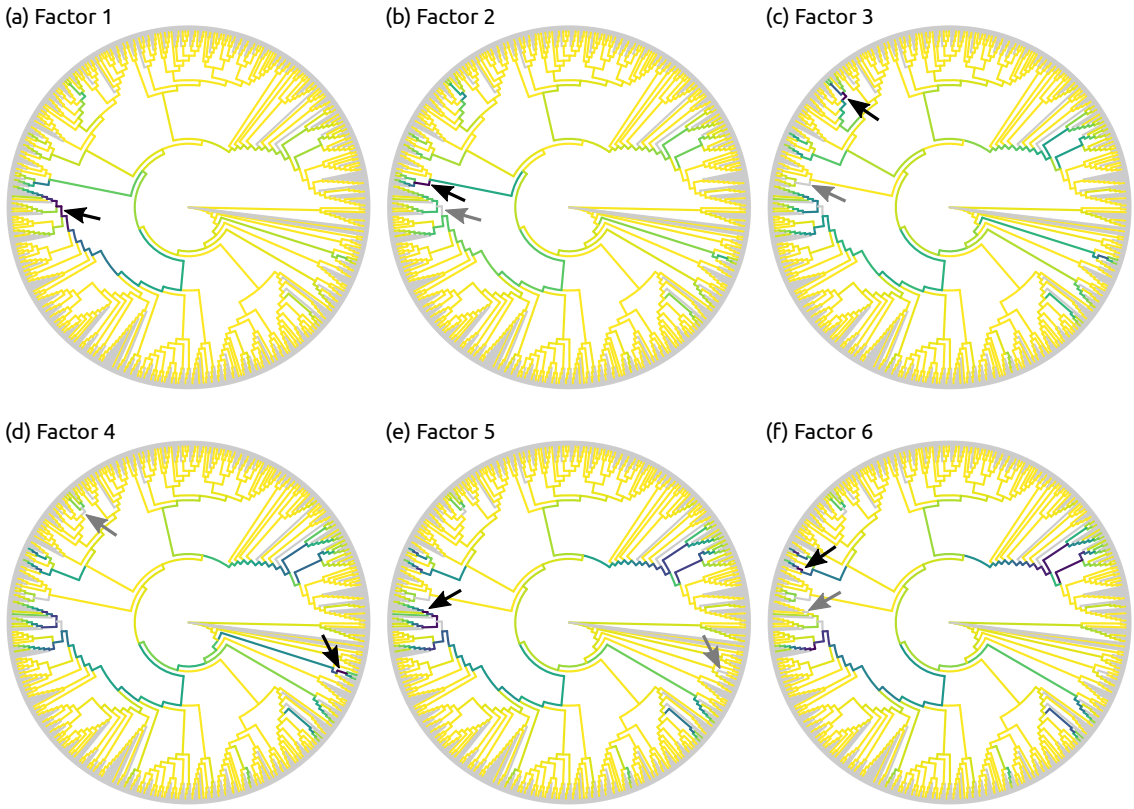

Figure 7.6: Objective function values for the first six factors of the BV dataset without taxon weighting. The figure visualizes the value of the objective function for the first six iterations of Placement-Factorization of the BV dataset, *without* taxon weighting. Darker edges represent higher values; the highest value of each iteration (the winning edge) is marked with a black arrow. Gray arrows further mark the winning edge of the respective previous iteration, which allows to examine the effect of "factoring out" an edge. See also Figure 7.5 for the according visualization *with* taxon weights.

healthy vs. diseased patients works both with and without taxon weighting, albeit the differences in the respective plot shapes are pronounced.

On the one hand, Placement-Factorization *with* taxon weighting in Figure 7.7(a) is highly similar to PCA on the balances as shown in Figure 6.2(a), despite the fact that PCA does not take the meta-data into account. We suspect that is is due to the nature of the dataset, were the abundances in the *Lactobacillus* clade almost solely dictate the health status of each individual, and roughly half the samples belong to either the healthy or the sick group of patients. Hence, the *Lactobacillus* clade naturally is a major driver of differences between samples, and is thus identified by PCA as the most important component/axis.

On the other hand, Placement-Factorization *without* taxon weighting in Figure 7.7(b) yields an ordination that separates healthy from sick patients in the first factor, and

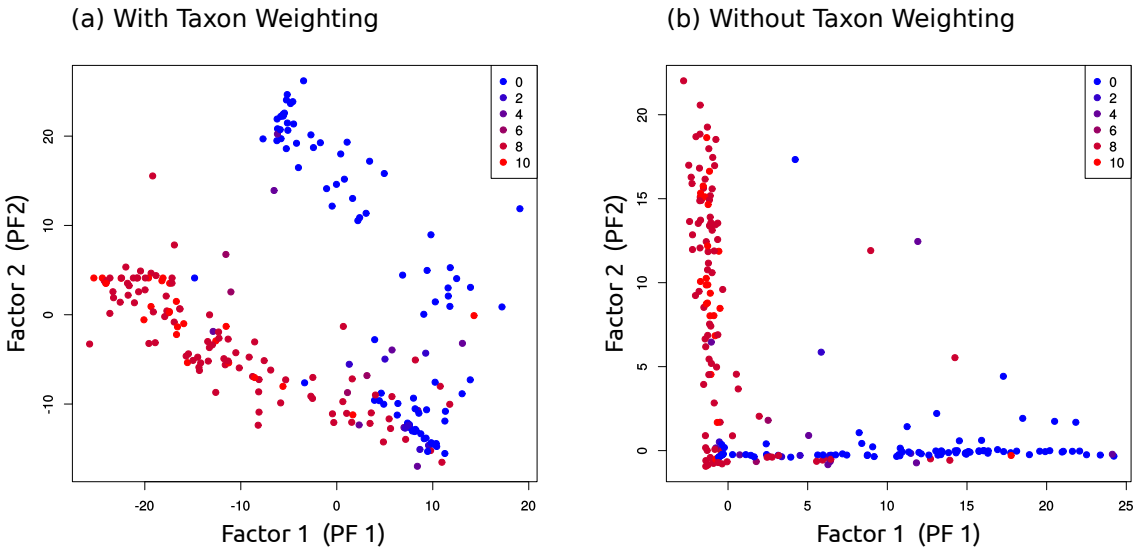

Figure 7.7: Ordination of the first two factors of the BV dataset. The figure shows ordination-visualization plots of the ILR coordinates (balances) of the first two factors found by Placement-Factorization of the BV dataset, (a) with and (b) without taxon weighting. That is, the axes correspond to the splits induced by the first two factors, while values along the axes are the balances of each sample calculated on the sets of edges of each split. Samples are again colored by their Nugent score, with 0 being the healthy patients, and 10 being the patients with severe BV. See Figure 7.5 for the (winning) edges that correspond to the axes in Subfigure (a) (with taxon weighting), and see Figure 7.4 and Table 7.1 for the edges corresponding to the axes in Subfigure (b) (without taxon weighting).

further splits the sick patients in the second factor. The reason for this can be seen in the clades that each factor splits away from the tree, as shown in Figure 7.4: The first factor separates part of the *Lactobacillus* clades, which explains why it distinguishes samples based on heath status. The second factor separates a clade containing *Sneathia sanguinegens* and *Leptotrichia amnioni*, which is an important clade among several clades that are associated with BV [339]. This can also be seen in Figure 4.2, where the healthy patients with low Nugent score almost exclusively exhibit high abundances of *Lactobacillus*, while the diseased patients with high Nugent score show abundances in several clades all over the tree.

Figure 7.7 hence serves as a caveat for Phylofactorization and Placement-Factorization, and as an example of the limitations of this type of plot. These plots were suggested in Washburne et al. (2017) [376] as an additional way of depicting how the factors separate samples according to meta-data features, see their Figure 5(a). Note that such scatter plots can only reasonably visualize the first two or three factors, which is why they are now discontinued in the original Phylofactorization (pers. comm. with A. Washburne on 2019-01-16).

In the case of the BV dataset, two axes/factors are sufficient to separate the samples by Nugent score. That is, the BV dataset does indeed have two important features concerning the *healthy* patients, namely the *Lactobacillus* clade, and the further distinction into *Lactobacillus crispatus* and *Lactobacillus iners*. However, as discussed above and visible in Figure 4.2, the *diseased* patients exhibit high abundances in a multitude of other clades, which cannot be expressed by just two or three factors.

It is hence crucial to compute all significant factors—otherwise, important aspects of the data get lost and results are incomplete. We also developed a novel way of visualizing the balances of further factors, as for example shown later for the HMP dataset in Figure 7.9, which alleviates this issue.

7.3.2 Oral/Fecal Subset of the HMP dataset

We here show an analysis of a subset of the Human Microbiome Project (HMP) dataset [160, 250], in order to compare Placement-Factorization to findings of the original Phylofactorization on a similar dataset [376]. See Appendix B.3 for details on the dataset and its preprocessing.

The original publication of Phylofactorization used a dataset from Caporaso et al. (2011) [45] as one of their case studies, which comprises oral and fecal samples from the human microbiome. See Figure 4 and Supplementary Figures S3–S8 of Washburne et al. (2017) [376] for the original analysis with Phylofactorization. For our comparison, we selected a suitable subset of the HMP dataset: In particular, we selected all 600 stool samples of the dataset, as well as 600 randomly chosen samples from the mouth region, that is, from the samples labeled "Mouth (back)" and "Mouth (front)" in Table B.2. We again used the placement of these samples on the unconstrained *Bacteria* tree of our Phylogenetic Automatic (Reference) Tree (PhAT) method to conduct Placement-Factorization. The tree contains 1914 taxa, as explained in Section 3.3. We henceforth assume that the oral/fecal dataset of Caporaso et al. (2011) [45] and our oral/fecal subset of the HMP dataset exhibit comparable sequence compositions. Furthermore, as the tree used for Phylofactorization in Washburne et al. (2017) [376] is based on the OTUs of the sequences, it only contains taxa that are sufficiently abundant in the input. It thus differs from the more general *Bacteria* reference tree used for our evaluation here. Therefore, we had to map the taxa found by Phylofactorization to the underlying Silva taxonomy [294, 395] that was used for constructing our reference tree.

Comparison to Phylofactorization

Despite these differences, Placement-Factorization yielded factors that are similar to the ones found by Phylofactorization. We again used Placement-Factorization *with* and *without* taxon weighting, and compare the taxa identified by the first 10 factors to the the first 10 factors found in the original oral/fecal dataset with Phylofactorization [376]. We visualized the clades found by all three variants in Figure 7.8.

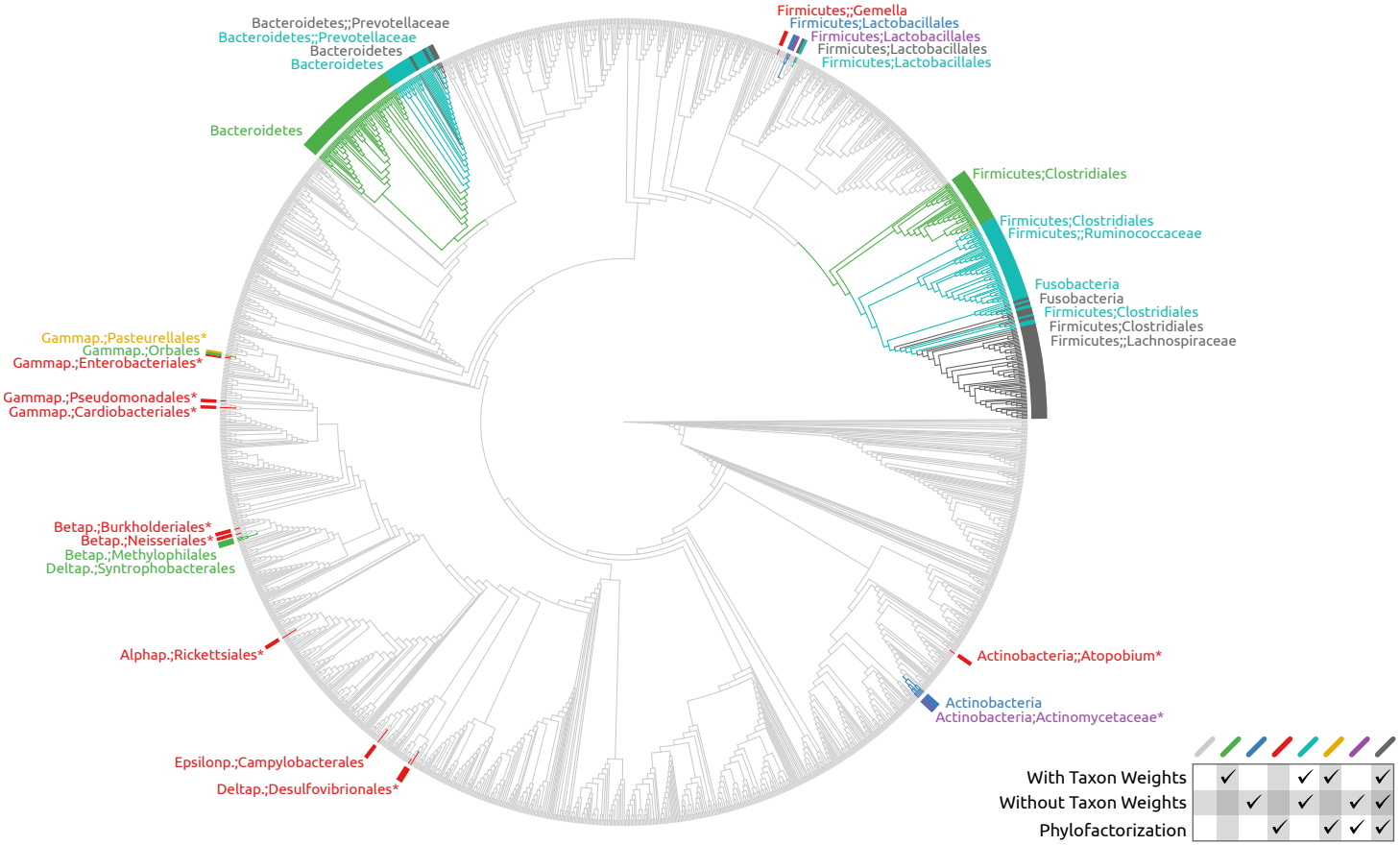

Figure 7.8: Comparison of factors found in the oral/fecal subset of the HMP dataset. Here, we compare the first 10 factors found by Placement-Factorization with and without taxon weighting on an oral/fecal subset of the HMP dataset to the first 10 factors found by Phylofactorization on their oral/fecal test dataset [45, 376]. The clades of the tree are colored so that green, blue, and red mark branches that only appear in one of the variants, cyan, yellow, and purple for branches that occurred in two variants, and dark gray for branches that were found by all three variants. For simplicity, we here neglect the order and nesting of factors. That is, if a branch is part of the non-root side of any one of the first ten factors, it is colorized here.

For simplicity, we only compare the clades on the non-root side of the (arbitrarily rooted) reference tree; the paraphyletic "remainder" clade is not taken into account. Furthermore, we do not consider the order of the factors here. Similar to the findings of the BV dataset above, Placement-Factorization *with* taxon weighting yielded larger clades than *without* taxon weighting, which again yielded larger clades than the OTU-based Phylofactorization. The latter is a consequence of the OTU tree containing fewer taxa than our broad *Bacteria* tree. We found that 84% of the taxa identified by Phylofactorization were also part of the factors of our variants, with the major difference being a set of *Proteobacteria* that were part of the split in the first factor of Phylofactorization [376], but not by our variants. This is most likely an artifact of the differing trees being used in the factorization. Furthermore, 95% of the taxa found by Placement-Factorization *without* taxon weighting were also part of the clades *with* taxon weighting.

In particular, the larger clades found by the variant with taxon weights are shown as green edges in Figure 7.8. The values of the objective function however indicate that the focus of the factor is in fact much smaller and more in agreement with the other two variants compared here. Again, this behavior is similar to the BV data with taxon weighting; see Figure 7.5 for details. Furthermore, Phylofactorization found several small clades and single branches, which are shown as red edges in Figure 7.8. These branches are part of the *Actinobacteria* as well as the *Alpha-*, *Beta-*, *Gamma-*, and *Deltaproteobacteria*, and are actually all part of the first factor. They are marked with asterisks (*) in Figure 7.8. Due to their OTU tree only having few *Proteobacteria*, these were monophyletic in their tree [376]. They are polyphyletic here, as our tree has more reference taxa from that group.

Most of the remaining factors found by Phylofactorization are part of the gray branches of the two large clades in the upper half of Figure 7.8, which are the clades that were found by all three variants. Similarly, our variants found many nested factors (factors that further split a factor of a previous iteration): The two large clades are in fact split into seven nested clades by both variants, with the remaining three factors spread across the rest of the tree (e. g., the green and blue branches of Figure 7.8). In particular, the *Prevotellaceae* and parts of the *Firmicutes* were described in Washburne et al. (2017) [376] as important clades for the distinction between oral and fecal samples, all of which were found by all three variants here.

In total, despite the mismatching trees, most of the found clades agree in all three variants, with their disagreement mostly concerning the clade sizes. More importantly, despite their differences, all variants produce factors that are well suited for separating oral from fecal samples, as further shown in the next section below.

The actual differences in taxa (such as the *Proteobacteria* not being found by our variants) serve as a caveat for the importance of the underlying reference tree: Differences in topology will inevitably be reflected in different factors, which might in turn suggest a different interpretation of results. In an ideal world with a known phylogeny of all of life, alternative OTU clusterings and alternative trees would simply collapse nodes at different depths (pers. comm. with A. Washburne on 2019-03-01). Unfortunately, real world data, and particularly different OTU clustering methods

and tree inference methods, will yield discordant trees. The influence of uncertainty in the phylogeny is further discussed in Washburne et al. (2019) [378].

Factor Ordination

Next, we investigated how well the factors found by Placement-Factorization separate oral from fecal samples. To this end, we again employed the balances of the winning edge of each factor for an ordination visualization [376], which we show in Figure 7.9(a) and (b). The ordination clearly separates the samples, both with and without taxon weighting. Again, ordination scatter plots can only reveal up to three dimensions/factors. In order to evaluate the separation of samples at later factors, we use a visualization of the factor balances, which we call *balance swarm plots*, and which are similar to the per-factor ordination plots used in Washburne et al. (2019) [378]. These plots can show the ordination of arbitrarily many factors at the same time, as shown in Figure 7.9(c) and (d).

Figure 7.9 indicates that most factors found by our Placement-Factorization are indeed capable of separating body sites from each other. In particular, Figure 7.9(a) exhibits a clear separation of the two body sites, similar to Figure S3 of Washburne et al. (2017) [376]. Furthermore, Figure 7.9(c) shows that almost all factors individually suffice to separate the data by body site: eight out of the first ten factors found by Placement-Factorization *with* taxon weighting clearly separate the oral from the fecal samples. The remaining two factors (PF7 and PF9) separate most of the samples, but also have an interval of balances that contains samples from both body sites. Placement-Factorization *without* taxon weighting also separates samples based on their body site, as shown in Figure 7.9(d), but with a less clear distinction. This is also obvious from the ordination scatter plots shown in Figure 7.9(b).

7.3.3 Full HMP dataset

Finally, we conducted Placement-Factorization on the whole HMP dataset with all 9192 samples, instead of just the oral/fecal subset, in order to evaluate how the method performs on large datasets with more than two categories (body sites) to distinguish. See Table B.2 for an overview of the samples, as well as a list of the eight body site labels that we used for classifying the samples. We here do not discuss the taxa that were split by each factor, as such an in-depth biological discussion is beyond the scope of this work. Instead, we evaluate how well different body sites were separated by the factors. To this end, we show ordination plots of the first two and three factors in Figure 7.10.

These plots already reveal that Placement-Factorization indeed separates samples from each other based on their body site. However, given the eight body site labels that we used, these plots are overloaded and hard to read. Hence, we extended on the idea of balance swarm plots (as introduced above for the oral/fecal subset) by separating them into individual plots per factor, each showing the balance distribution of groups of samples based on their respective body site. This allows to clearly see the distribution of balances at the factor for all groups of samples. An example for the first four factors is shown in Figure 7.11.

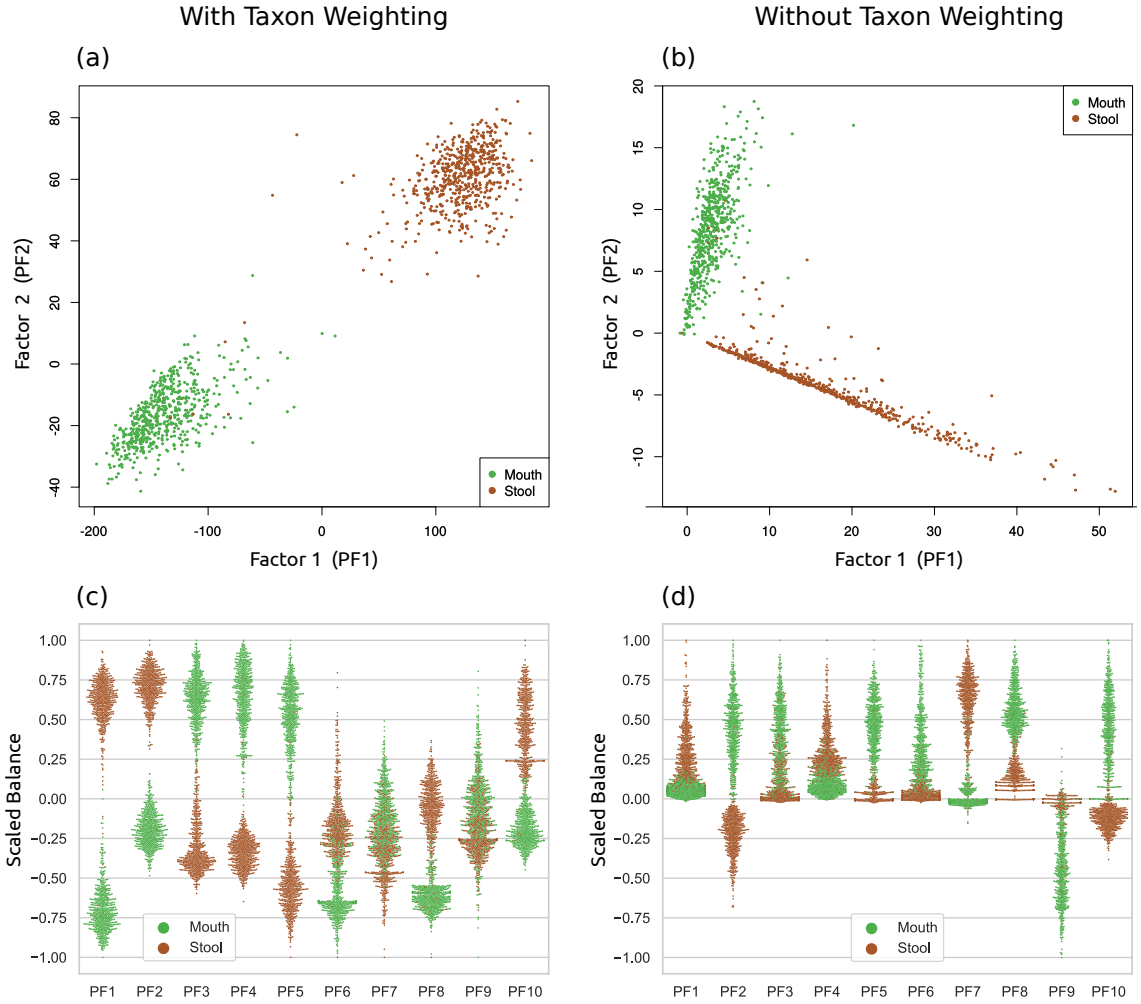

Figure 7.9: Ordination of an oral/fecal subset of the HMP dataset. The figure shows the ordination-visualization of factors found by Placement-Factorization on our oral/fecal subset of the HMP dataset. In (a) and (b), we show the balances at the winning edges of the first two factors, colorized by the body site of each sample, with and without taxon weighting.
Moreover, in order to examine how well further factors of later iterations split the data, we here employ a visualization of phylofactors, which we call *balance swarm plots*, by plotting the balances of each factor individually. This type of per-factor visualization is similar to, e. g., Figure 4 of Washburne et al. (2019) [378]. Subfigures (c) and (d) show the first ten factors (PF1–PF10), again with and without taxon weighting, respectively. These can be understood as multi-dimensional scatter plots, where each dimension is shown separately: Each column corresponds to a factor, with the vertical axis being the balances, and horizontal space within each column used to spread samples at nearby positions, revealing their distribution density. That is, the first two columns of (c) and (d) correspond to the scatter plots of (a) and (b), respectively. The balances were scaled to bring them into the $[-1.0, 1.0]$ interval for better comparability across factors, while keeping the centering at 0.

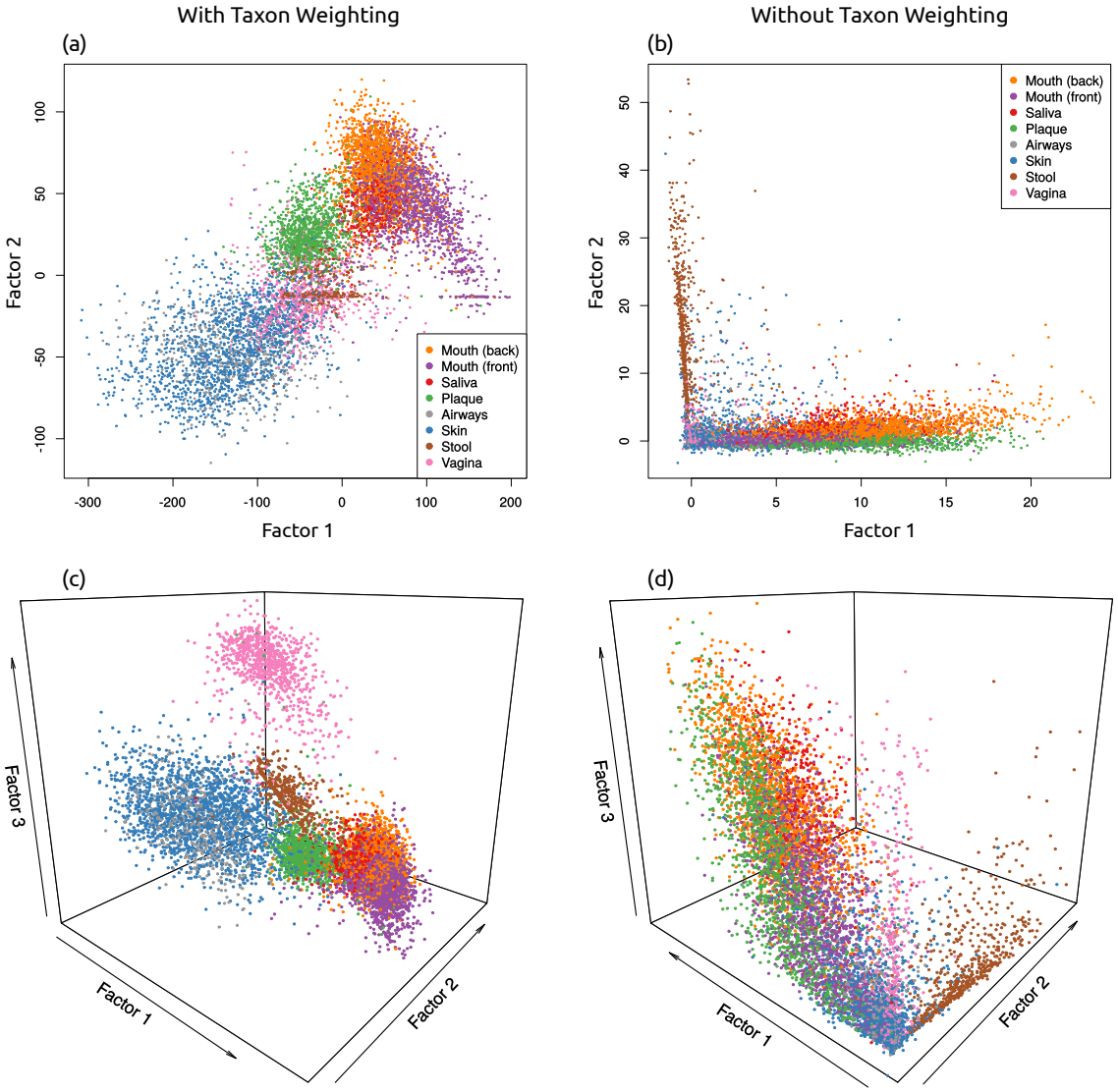

Figure 7.10: Ordination of Placement-Factorization of the full HMP dataset. Here, we use the whole HMP dataset, labeled by 8 body site regions as listed in Table B.2, to asses how well Placement-Factorization with a GLM objective function can separate samples based on their body site label. The figure again shows the balances of the winning edges of the first two and three factors, respectively, with and without taxon weighting.

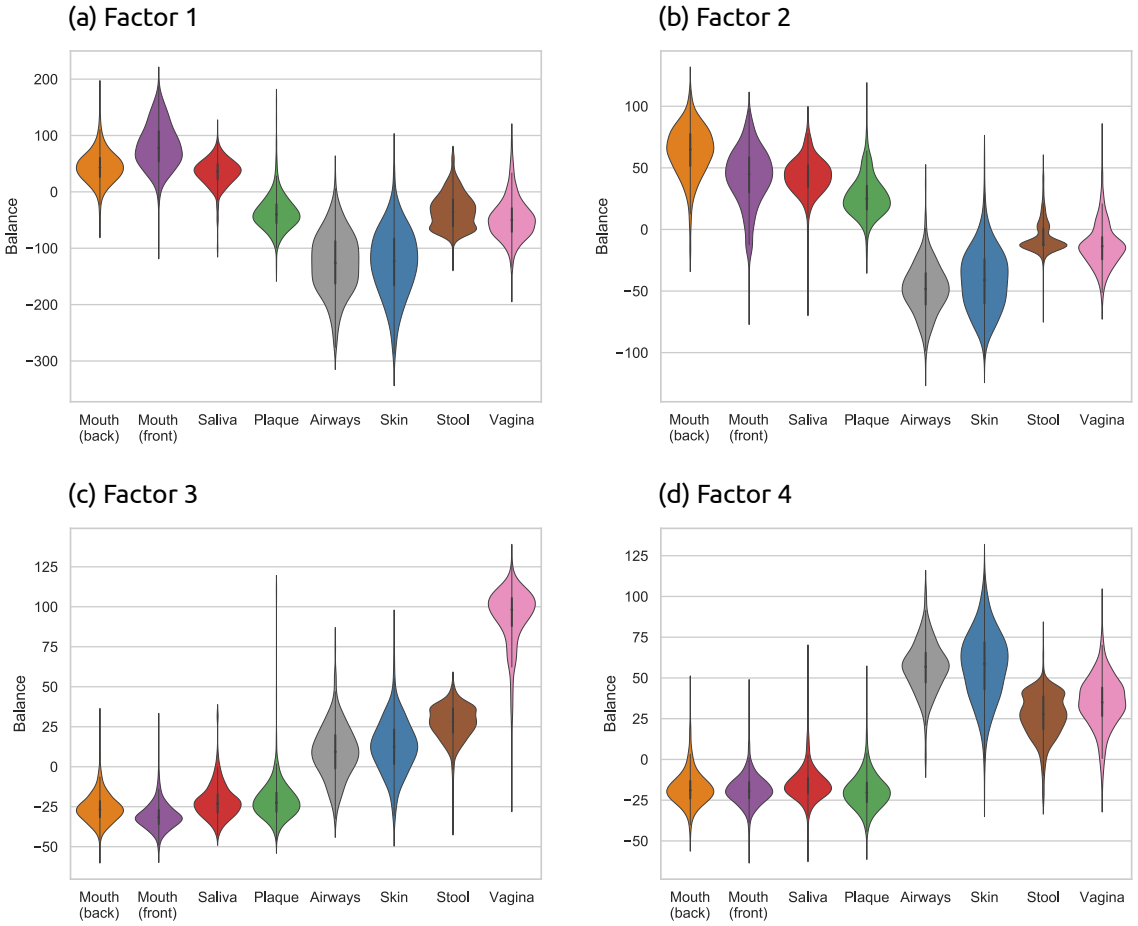

Figure 7.11: Ordination of the first four factors of the HMP dataset. The balance swarm plots as shown in Figure 7.9(c) and Figure 7.9(d) allow for a more detailed understanding of how each factor separates the samples. They can be colored by either continuous meta-data variables, similar to Figure 5(a) of Washburne et al. (2017) [376], or a categorical variable with a limited number of categories, as shown in Figure 7.9. However, for the eight body regions that we use for the HMP data, this type of visualization becomes hard to inspect visually. Hence, we here extend on the idea of balance swarm plots, and show the distribution of balances for each factor and for each body sites separately.
The data shown here is the result of Placement-Factorization with taxon weighting on the full HMP dataset. Each subfigure here shows the balances of the winning edge of a factor, grouped by the categorical meta-data variable body site. That is, the subfigures correspond to the scatter plots of the first two and three factors shown in Figure 7.10. In other words, each subfigure here represents a disentangled column of a balance swarm plot, where each body site is displayed separately by its own violin.

The visualizations shown in Figure 7.10 and Figure 7.11 indicate that Placement-Factorization separates samples mainly based on the distinction oral vs. remaining body sites, with a further separation of plaque samples in the oral region. This can, for example, be seen in Figure 7.11(a), where the first three groups "Mouth (back)", "Mouth (front)", and "Saliva" exhibit balances above 0, while all other groups have balances below 0. Further factors then separate vaginal samples and skin and airways samples from the rest of the samples, as shown in Figure 7.11(b)–(d).

As shown in Figure 7.7 and Figure 7.9 before, the plots *with* taxon weighting form "clouds", whereas the plots *without* taxon weighting form an "L"-shape. In all cases, a separation of the oral samples from the other regions is clearly visible. Noticeably, in Figure 7.10(a), a part of the stool and mouth samples form a horizontal line, which indicates that the second factor does not distinguish between samples from those regions. It is striking that the samples from the vaginal region in Figure 7.10(a) and (c) are not separated from the other samples until the third factor, which is visible as a pink cloud above the rest of the samples in Figure 7.10(c). This again serves as a caveat that one needs to consider enough factors in order to get a complete understanding of the results.

This can be seen in more detail in Figure 7.11. In all subfigures, the oral regions are separated from the other regions: For example, in Figure 7.11(a), mouth and saliva samples exhibit balances above 0, in contrast to all other samples. In Figure 7.11(b)–(d), the plaque samples join the other oral samples in terms of their balance values. In Figure 7.11(b), the stool samples have a distinct bulge near 0, which corresponds to the horizontal line (factor 2) in Figure 7.10(a). Furthermore, the third factor in Figure 7.11(c) again clearly separates the vaginal samples from the rest, corresponding to the pink cloud in Figure 7.10(c).

Overall, Placement-Factorization can distinguish the HMP samples by body site, at least to the extent that can expected from abundance differences in the samples. For example, it would be unrealistic to expect the algorithm to perfectly separate samples from the back and front of the mouth from each other, as their microbial compositions are expected to be highly similar.

7.3.4 Performance

The run time of Placement-Factorization depends on (a) the number of input samples, (b) the number of branches of the reference tree, and (c) the number of iterations to run. As the computations are conducted on the mass matrix instead of single placements, the performance and memory requirements of Placement-Factorization are independent from the total number of sequences/placements in the dataset. In each iteration, and for each edge of the tree (except the ones that won previous factors), the balances of all samples are computed, and the objective function is evaluated. In case of using a GLM to express the relationship of balances with meta-data, this involves fitting a model across all samples. In our implementation, all these computations are parallelized.

Our relatively small BV test dataset ran on a standard laptop with 4 cores, taking 30 s per iteration. The full HMP dataset with 9192 samples and our reference tree

with 3825 branches required 13.0 GB of memory in our non-optimized prototype implementation, and took less than 90 s per iteration using 20 cores. Also, note that each iteration splits away a clade of the tree; later iterations thus become faster, as the sizes of the subtrees within which the balances need to be computed get smaller each time. Hence, we conclude that Placement-Factorization is well suited even for very large datasets.

7.4 Summary and Outlook

In this chapter, we presented an adaption of Phylofactorization [376, 378] to phylogenetic placement data, which we call *Placement-Factorization*. Placement-Factorization identifies branches of the reference tree, called *phylogenetic factors*, that exhibit a relationship with environmental meta-data features, that is, branches along which putative functional traits might have arisen in conjunction with changes in environmental variables. This factorization of the tree can be used as an ordination tool to visualize how samples are separated by changes along the factors, and as a dimensionality-reduction tool [376]. It thus complements Edge Correlation (see Section 4.2.2), in that it further allows to identify nested dependencies within subclades of the reference tree, and that it can take multiple meta-data features into account at once, however without the immediate visualization of the direction of the correlation/dependency with these features.

We contributed novel ideas to Phylofactorization by (a) adapting the concept to phylogenetic placement, which can be thought of as placing abundances along branches of the tree instead of just at its tips; and (b) suggesting a novel visualization for the objective function value at each edge of the reference tree, which helps in the interpretation of the factors being split in each iteration. Furthermore, we explored several advantages of using a fixed reference tree instead of a tree inferred from OTUs in Section 7.2.4. We leave the adaptation of some of the original concepts of Phylofactorization to phylogenetic placements as future work, such as binned phylogenetic units (BPUs), stopping criteria for the iterations, as well as further experimentation with different objective functions and aggregation and contrast functions [376, 377]. Based on our findings and experiments, we conjecture that these concepts should be readily applicable to our Placement-Factorization.

In contrast to the original Phylofactorization [376], our implementation also supports the taxon weighting scheme used in the balances computation, as explained in Section 6.2.2. We find that Placement-Factorization *without* taxon weighting behaves similar to the original Phylofactorization (which also does not employ a taxon weighting scheme), while Placement-Factorization *with* taxon weighting yields results that are more in line with our previous results based on edge imbalances (Section 2.5.3). The latter is likely because taxon weighting has a similar effect of reducing the influence of low mass branches (low abundance taxa) as the summation-based aggregation step of imbalances.

As discussed in Section 7.3, we found that both, the original Phylofactorization, as well as our Placement-Factorization, can split clades that are larger than one would

expect from other types of analyses of the data. Considering the distribution of objective function values, as for instance shown in Figure 7.5, it is likely that such large clades are the result of random variability along a path of branches that are equally relevant for the factor. Further research is needed to confirm this.

These findings suggest that it might be beneficial to introduce a significance value for each factor, which assesses how relevant the particular winning edge is compared to other edges that yielded a high objective value in an iteration. This idea is intrinsically connected (pers. comm. with A. Washburne on 2019-03-01) to the stopping function of the original Phylofactorization [376], which uses a Kolmogorov-Smirnov (KS) test [237] to conservatively estimate when a sufficient number of factors have been identified. Another strongly connected idea is that of confidence regions of the phylogeny, defined by regions of the tree in which the "true" winning edge falls with a certain confidence [378]. Such a significance value for the winning edge might also enable a form of *soft* factorization, that does not greedily pick one winning edge per iteration.

Furthermore, the paths of high objective values as for example seen in Figure 7.5 indicate that there is a gradient of the objective function along the edges of the tree. This could be exploited in a gradient-ascending graph-walking algorithm to identify the phylogenetic factors of extremely large datasets without having to exhaustively evaluate the objective function at every edge (pers. comm. with A. Washburne on 2019-03-01). For example, one could start at one or more random edges on the tree, and ascend along the edges in the direction of the gradient until a local maximum of the objective function is found.

Phylofactorization is a very recent method whose full potential has just begun being explored [378]. Given the ideas for future research that we outlined here, we hence conclude that more work is needed in order to fully reach that potential.

8. Conclusion and Future Directions

This chapter mostly contains original contributions by Lucas Czech written for this thesis. Some of the text is however derived from the concluding sections of the peer-reviewed open-access publication:

> **Lucas Czech** and Alexandros Stamatakis. "Scalable Methods for Analyzing and Visualizing Phylogenetic Placement of Metagenomic Samples." *PLOS ONE*, 2019, Volume 14, Issue 5, Page e0217050.

The respective text was originally written by Lucas Czech, and modified here to fit into the context of this chapter.

In this thesis, we made several contributions to the field of computational phylogenetics, and phylogenetic placement in particular. We have already described the methods in two peer-reviewed publications [67, 69], and made the respective data and scripts available at `http://github.com/lczech/placement-methods-paper`. The implementation is available in our GENESIS library (`https://github.com/lczech/genesis`); we furthermore offer a ready-to-use command line interface for the presented methods via our GAPPA tool (`https://github.com/lczech/gappa`). Both these tools are described in Appendix C as well as our application note [70]. This chapter briefly summarizes our contributions and discusses potential future developments and open research questions in the field of phylogenetic placement.

Having been introduced between 2009 and 2010 [26, 242], phylogenetic placement and its downstream analysis methods are still relatively recent tools, but are becoming more and more popular in the research community. Their advantages compared to other metagenomic analysis methods include improved interpretability and visualizability of results, at the cost of increased methodological and computational

complexity. There are hence still obstacles to overcome before phylogeny-aware methods (such as phylogenetic placement) can reach maturity and widespread use in the research community:

- **Usability:** Current software pipelines for phylogenetic placement are mostly build from separate programs and tools that are not well integrated with each other, and only interact via intermediate files and scripts. This induces labor costs for first-time users of these tools, for instance, because of the manual compilation and setup of the software, as well as the time required for ad-hoc scripting solutions to connect different steps of the pipeline.
- **Scalability:** The amount of available sequence data is growing exponentially (see Section 1.1), currently doubling roughly every seven months [345]. Phylogeny-aware analysis methods for sequence data are generally more computationally intense than, for instance, methods that are based on sequence similarity. The speed and scalability of these tools thus need to stay on a par with the amount of sequencing data being produced.
- **Analysis Methods and Research Questions:** Downstream analysis methods that take phylogenetic information into account are not yet as versatile and mature as classical data analysis methods. Hence, researchers might refrain from using phylogenetic placement and resort to more established methods. There are however numerous research questions that might benefit from analyses using phylogeny-aware methods, for example, in disciplines such as bio-geography or medicine.

In the following, we discuss our contributions to the field with respect to these obstacles, and describe potential future research directions.

8.1 Usability and Scalability

In this section, we discuss our contribution to software development for analyzing phylogenetic data, as well future directions for such software in general, and our software in particular.

Contribution

In Chapter 3, we presented an approach to automatically obtain reference trees for phylogenetic placement called *Phylogenetic Automatic (Reference) Trees (PhATs)* using large reference sequence databases. We showed that PhATs are valuable and sufficiently accurate trees for conducting phylogenetic placement and taxonomic assignment of metagenomic sequences. They thus help with the labor-intense and potentially error-prone process of collecting suitable reference sequences, and might even replace such “manual” trees in some applications. In the same chapter, we presented a multi-level placement approach as well as our pre-processing pipeline to enable and accelerate phylogenetic placement of large, diverse datasets with hundreds to thousands of environmental sequence samples.

General Future Development

In order to further facilitate the use of phylogenetic placement methods, without the need for complex setups and scripts, online services for quickly testing standard methods might be worth to offer. For example, uploaded query sequences could be placed on our four PhATs used in the evaluation (Section 3.3.1), or on custom reference trees. These services could further offer standard analysis methods such as Edge PCA and Squash Clustering (Section 2.5.5), or the methods that we presented in this thesis (Chapters 4–7).

For more advanced or large-scale studies, a custom setup is mostly inevitable, because users need to scale up the processing using computer clusters as well as customize the workflow. Such studies might benefit from offering well-defined integrated pipelines for the basic steps of phylogenetic placement, with a simple setup for all common functionality, for instance by using platforms such as Galaxy [4] or Conda/Bioconda [136]. Tools and pipelines that integrate different steps of the analysis furthermore allow for more efficient implementations, as they can make better use of computing infrastructure, e. g., via parallel programming interfaces such as `MPI`, and distribute computations with less overhead for intermediate files and bookkeeping between pipeline steps.

Moreover, to achieve a tighter integration of the tools in a pipeline, and to incentivize experimentation with existing methods and development of novel methods, file formats need to be flexible and extensible. For example, the `jplace` file standard for phylogenetic placement [243] is based on the `JSON` format [35, 65], and hence easily extensible. The standard however currently (as of `jplace` version 3) lacks support for multiple samples per file, and additional per-sample or even per-query annotations and other meta-data. A similar issue applies to file formats for phylogenetic trees, as we showed in Czech et al. (2017) [68]. In particular, the `Newick` format is often extended ad-hoc for specific needs (this is even done within the `jplace` format itself), with no standardized support from downstream tools or tree viewers. This leads to ambiguities, misinterpretations, and errors in study results that could be avoided by creating proper standards for these data types [68].

Future Development of our Software

With respect to the methods and tools presented in this thesis, we plan to extend our gappa tool [70] to include additional existing analysis methods, such as placement-based diversity measures [241] and UniFrac calculations [221], and additional pre- and post-processing functions, such as filtering, merging and manipulating placement files. In the long term, we also intend to re-implement all functionality offered by the guppy tool from the pplacer suite [241], to turn gappa into a more efficient and scalable replacement for guppy. Moreover, there is a Conda/Bioconda [136] recipe for installing gappa [87]. The recipe was created as part of the PICRUSt2 pipeline [88], which internally relies on gappa for some of its steps. The recipe hence is not maintained by us; we however plan to monitor it in order to ensure its usability.

Furthermore, we are planning to extend the GENESIS library [70], which is written in `C++11`, to also offer API bindings for `Python`. Hence, GENESIS could combine the advantages of both programming languages: For standard functions such as file processing and most internal computations, it can use our highly efficient `C++` implementations, which relies on multi-threading via `OpenMP` (Open Multi-Processing) where appropriate to allow usage of the available compute cores. The modular and (hopefully) clear API of GENESIS further allows it to be used in production code, as evident by its use in several of our publications [18, 67, 69, 70, 230, 403]. At the same time, ad-hoc solutions for pipeline tasks such as file conversions or extraction of certain data could be implemented as `Python` scripts. This also facilitates to use GENESIS for experimentation and rapid prototyping of novel ideas and methods.

8.2 Analysis Methods

Downstream methods for analyzing metagenomic data in a phylogenetic context, and in paricular, methods based on phylogenetic placements, are not yes as plentiful as standard sequence-based methods. We contributed several novel ideas and adaptations of existing concepts, which we summarize in the following. Then, we present some ideas for adapting existing machine learning techniques to phylogenetic data.

Contribution

In Chapter 4, we described methods for visualizing the phylogenetic placement of large metagenomic datasets. The methods allow to detect differences between samples (*Edge Dispersion*), as well as correlations with per-sample meta-data (*Edge Correlation*), and are thus intended for similar use cases as the established Edge PCA [239]. However, our novel methods directly visualize important features of the samples (and their meta-data) on the underlying reference tree. This allows for interpreting them in a phylogenetic context. Furthermore, in Chapter 5, we introduced clustering methods for metagenomic samples (*Phylogenetic k-means* and *Imbalance k-means*), which serve a similar purpose as Squash Clustering [239], but are better suited for larger datasets due to their fixed output size.

In Chapter 6, we introduced an adaptation of the Phylogenetic ILR transformation and balances [330] to phylogenetic placements. Our adaptation uses a fixed reference tree (instead of a tree inferred from the OTUs present in the data), and allows to place sequences along the branches of the tree (instead of only at its tips). As balances are a transformation that yields orthogonal components, issues like the normalization of compositional data (Section 2.5.3) do not arise. As we showed, with samples being represented as a vector of balances, many standard tools for visualization, ordination, and clustering of data in the Euclidean space can be readily applied to phylogenetic placement data.

In Chapter 7, we presented an adaptation of Phylofactorization [376] to phylogenetic placements, which we call *Placement-Factorization*. These methods are able to identify edges of the tree that exhibit predictable changes in species abundances based on per-sample meta-data features. This allows to understand evolutionary

and ecological patterns (e. g., abundances in certain clades of the tree) that are driven by environmental factors (e. g., pH value of the soil). The output and figures produced by these methods allow for a detailled interpretation of the results, and allow to simultaneously understand many different aspects of the data. As outlined in the chapter, the capabilities of these methods have just started being explored. We hence see large potential for futher developing Phylofactorization and Placement-Factorization.

Classical Machine Learning Approaches

The novel methods that we introduced in Chapters 4–7 extend the portfolio of available analysis tools for phylogenetic placement. Additionally, there exist several approaches from the fields of machine learning and data mining that could be useful to adapt to phylogenetic placement data. It might also be worth to develop integrative methods that can incorporate heterogeneous features such as phylogenetic data and different types of meta-data in a combined analysis [235]. While Placement-Factorization already is a first step in this direction, there is more potential for fully integrated data analysis methods. The challenge for adapting existing methods usually consists in making the methods phylogeny-aware, for example by having them operate on mass distributions on trees (including respective distance measures such as the KR distance) rather than on $\mathbb{R}^d$ vectors that are typically used in machine learning algorithms.

As we showed, *unsupervised* methods (such as k-means clustering) can be extended to phylogenetic placement data, and are thus valuable tools for metagenomic data exploration. It might therefore be interesting to also adapt other types of unsupervised machine learning methods to such data, for instance, different clustering [195] or dimensionality reduction [361] techniques, or visualization [200] and anomaly detection approaches.

The adaptation of *supervised* machine learning approaches to phylogenetic data might however be more complicated. In recent years, several approaches have been proposed for metagenomic sequence data, using large-scale machine learning [337, 363] and deep learning [11, 112] techniques. These approaches consider each sequence to be a data point, meaning that enough training data are available. They are applied for tasks such as OTU clustering, sequence classification, taxonomic assignment, and gene prediction. In this thesis however, we are interested in entire metagenomic samples (instead of single sequences) as the data points to be considered. Current datasets are often not large enough (in number of distinct samples) to allow for robust learning at this level of granularity without over-fitting the training data [11]. There are however some recent approaches in this direction, which focus on comparative studies [337]. They are mainly based on features such as abundances and presence/absence patters of sequences (or OTUs) within the samples [282], or use so-called k-mers (subword occurrences within sequences) as features [15]. These features are simple enough to allow for training with current dataset sizes. However, given the growth rate of metagenomic sequence data, sufficient training data for

more detailed analyses are likely to become available in the future. Hence, incorporating phylogenetic information in supervised machine learning methods might yield improved accuracy as well as better interpretability. This could be used for tasks such as classification of samples, and prediction and regression with respect to per-sample meta-data features. Placement-based balances and phylogenetic factorization are potential ways to approach this idea [330, 376], as indicated by the results of this thesis.

Neural Networks and Deep Learning Approaches

Notable supervised approaches that are interesting and promising for problems in the life sciences are *neural networks*, and *deep learning* in particular [331]. Such methods are, in principle, well suited for the complex and high-dimensional data produced in these domains. Initial approaches have recently been described for a variety of use cases in biology and medicine [55, 232, 257]. There are however some general issues that need to be solved when employing deep learning. Firstly, similar to other supervised methods, the amount of available (labeled) data in current biological studies is often too limited for training deep neural networks [55, 257]. Secondly, deep learning is often considered as a "black box" that lacks interpretability and testability; there are however recent approaches to alleviate this [257, 301, 373]. Future data collection and method development efforts will likely be able to overcome these obstacles [373]. Then, deep learning approaches that take phylogenetic information into account could represent viable tools for analyzing and understanding metagenomic data.

There have also been first attempts that directly use tree structures and phylogenetic trees in neural networks. For example, the architecture of a neural network can be modeled according to a given graph structure [39, 316]. In such models, data on the nodes of the graph is propagated across the graph during the training of the network. This allows to exploit the additional information encoded in the graph. This could be used for phylogenetic trees as well, using the tree as a fixed graph, and the distribution of phylogenetic placements as the data that the system is trained on. A very recent and promising idea is to embed the evolutionary information of phylogenetic trees into the Euclidean space, and then use convolutional neural networks (CNNs) on the resulting matrices to analyze the underlying data [114, 300, 301]. These methods are able to accurately predict meta-data features of a set of metagenomic samples, given the per-sample taxonomic profile. In detail, these methods infer a tree from the OTUs present in the dataset, and use the per-sample OTU abundances to train the network. In consequence, the underlying tree differs in each study, which hinders the use of pre-trained networks, for example in medical applications, where data of new patients needs to be classified. Instead, one could easily embed a fixed reference tree in a similar manner, and use phylogenetic placement distributions on the tree instead of abundances for the training.

Incorporating Uncertainty

Phylogenetic placements already incorporate a measure of uncertainty for each query sequence in form of the likelihood weight ratio (LWR) of each placement position

(Section 2.5.1). They however assume that the given fixed reference tree on which the placement is conducted is correct. As discussed in Section 2.3.2, the "true" tree of life is unknowable [140]. In an ideal world, alternative OTU clusterings and alternative trees would simply collapse nodes at different depths (pers. comm. with A. Washburne on 2019-03-01, c. f. Section 7.3.2). Due to uncertainty in both the OTU clustering and the tree inference however, errors are introduced.

Measures such as *bootstrap support values* [99, 110, 335] and methods such as *Bayesian inference* (Section 2.3.3) can help to assess the uncertainty introduced by these errors. The possibility to incorporate the uncertainty of the tree inference has already been discussed in the context of Phylofactorization [378]: the more certain the scaffold provided by the phylogeny, the more certain the inferences about a clade obtain from the factorization. For future development, it might be informative to incorporate uncertainty in the analyses methods presented here as well.

8.3 Research Questions

The general concepts and techniques described above can be employed for developing more specialized methods to help answering novel research questions. In the following, we present some ideas for future applications and use cases in metagenomics, to which phylogeny-aware methods, and phylogenetic placement in particular, could be adapted by using these techniques.

Contribution

As mentioned in Section 1.2, we are currently working on SCRAPP, which stands for "Species Counting on Reference trees viA Phylogenetic Placements". It is a tool for estimating the per-branch species diversity in metagenomic samples on a given reference tree, and is hence useful for discovering and describing novel species in microbial data. Thus, SCRAPP adapts a common question in biology (species counting and diversity estimation), and brings it into a phylogenetic context, where results can be visualized and interpreted using the additional information of the reference tree. It is a pipeline that combines several tools developed in our lab, such as EPA-NG [18], PARGENES [265], and MPTP [173], and uses GENESIS [70] for file conversions and other intermediate tasks between pipeline steps. It is hence a first step in the direction of integrated pipelines as explained above.

A related question is that of estimating the microbial community composition in a metagenomic sample, for example in the form of per-taxon relative abundances and taxonomic assignments [211]. We explained the compositional nature of phylogenetic placements in Section 2.5.3, and described our ad-hoc implementation for taxonomic assignment based on phylogenetic placement in Section 3.3.4. In order to complement methods specifically developed for taxonomic assignment [7, 211, 322], our implementation could be extended and refined, and hence become a tool that uses phylogenetic context to answer questions of community composition.

Further Research Directions

Further fields in which we see an opportunity for developing phylogeny-aware methods are biogeography and ecology, as they ask questions concerning the distribution of species across geographic space and through time, as well as regarding their interactions with each other.

Research has been conducted on unravelling the bio-geographical patterns of microbial communities [158], and there also exist evolutionary approaches [61]. This could, for instance, be extended to phylogenetic placements by combining them with geographical data: The phylogenetic Kantorovich-Rubinstein (KR) distance between two metagenomic samples is a pairwise measure (Section 2.5.4); similarly, the geographic locations where the samples were collected yield pairwise distances between samples. These measures could be used to infer microbial patterns that change with geographical distance. This is similar to correlating the placement distribution with per-sample meta-data as described in Section 4.2.2, but requires some other form of correlation measure. We conducted initial tests in this direction, but unfortunately, the available datasets were shown to not exhibit such biogeographical patterns [203].

Instead of location, time is also used for comparing metagenomic samples. In such studies, the same location is sampled at different points in time, thus allowing to investigate the dynamics, variations, and (potentially) seasonal periodic patterns experienced in the microbial communities across time [60, 108]. This is typically assessed using the taxonomic composition of the samples. By extending such methods to phylogenetic placements, the time-dependent patterns could be displayed on a phylogenetic tree (or a series of trees), which again might provide additional information for interpreting and visualizing them. One could for instance construct a time series (video) of a tree showing how abundance patterns similar to Figure 4.1(b) vary over time.

Furthermore, in the field of ecology, the co-occurrence patterns between species have been studied for numerous habitats and ecosystems [209, 364, 399]. These interactions are often visualized in so-called co-occurrence networks [52, 107], which aggregate abundances at a high taxonomic level, and show links that indicate co-presence and exclusion between the depicted taxa. Unfortunately, even at high taxonomic levels, these networks are often extremely complex and hard to interpret, while potentially omitting valuable interaction details at lower taxonomic levels. Thus, it might be interesting to visualize such co-occurrences with respect to phylogenetic information. This entails the challenge to simplify the visualization and only show relevant co-occurrences, for example by highlighting branches of the reference tree that exhibit a high correlation with other branches.

Throughout this thesis, we used the Bacterial Vaginosis (BV) [339] and the Human Microbiome Project (HMP) [160, 250] datasets as examples for health-related applications of our methods. Many of the above research questions are also interesting from a medical point of view, and can be readily applied to other human datasets for assessing health status. In recent years, there have been studies that conduct phylogenetic analyses of widely researched diseases such as HIV/AIDS [37, 48, 299] and

different types of cancer [2, 40]. Thus, we see a large potential in phylogeny-aware methods for diagnostics, and human health in general.

In conclusion, the development of novel methods that use phylogenetic information is highly promising for many fields of research in biology, medicine, and the life sciences. In particular, methods based on phylogenetic placement allow to process large metagenomic datasets, and to visualize them in different ways using the underlying reference tree as an additional source of information that aids interpretation of the results.

A. Supporting Information

Table A.1: IUPAC notation of nucleobases and ambiguity characters. The table lists the character representations of nucleobases and their combinations (which are used to denote ambiguity) as suggested by the IUPAC (International Union of Pure and Applied Chemistry) Commission [161]. The names and symbols for ambiguity characters are chosen based on bio-chemical properties of the nucleobases. See Section 2.2.4 for more information.

Symbol	Description	Represented Bases					Complement
A	**A**denine	A				1	T
C	**C**ytosine		C			1	G
G	**G**uanine			G		1	C
T	**T**hymine				T	1	A
U	**U**racil				U	1	A
W	**W**eak	A			T	2	W
S	**S**trong		C	G		2	S
M	a**M**ino	A	C			2	K
K	**K**eto			G	T	2	M
R	pu**R**ine	A		G		2	Y
Y	p**Y**rimidine		C		T	2	R
B	not A (**B** comes after A)		C	G	T	3	V
D	not C (**D** comes after C)	A		G	T	3	H
H	not G (**H** comes after G)	A	C		T	3	D
V	not T (**V** comes after T and U)	A	C	G		3	B
N	any **N**ucleotide (not a gap)	A	C	G	T	4	N
Z	**Z**ero					0	

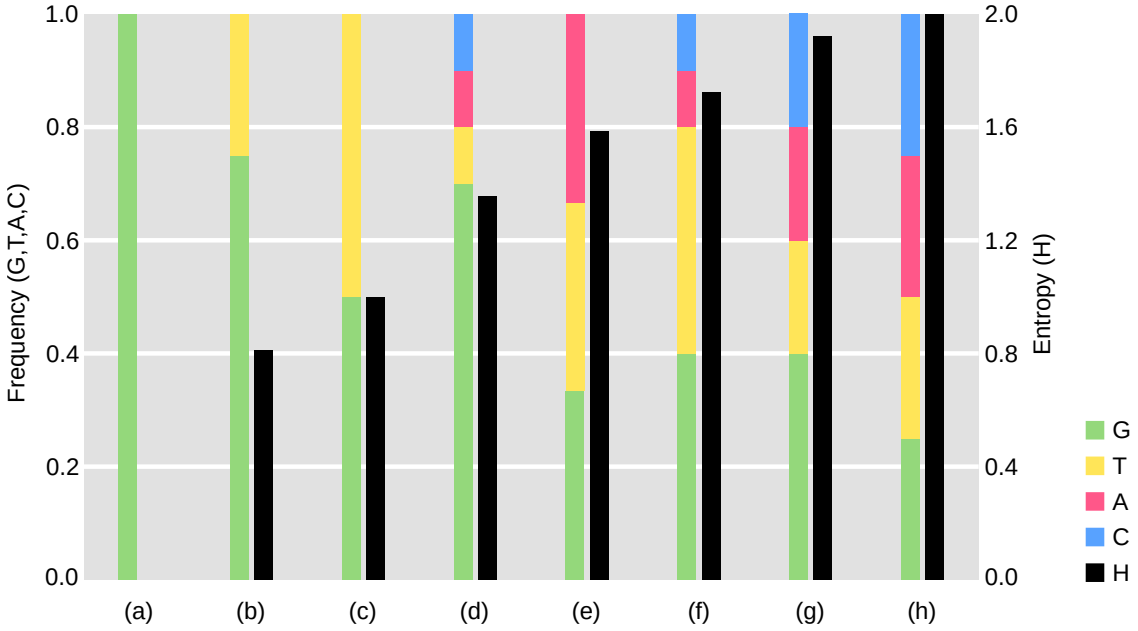

Figure A.1: Examples of the per-site entropy for different character frequencies. The figure shows the entropy H that results from an alignment site with some exemplary nucleotide frequencies. See Section 3.2.1 for details of the calculation of the per-site entropy, and for its application in the context of multiple sequence alignments. The entropy is symmetric with respect to permutations of the nucleobases; we here show examples using the nucleobase `G` as the most frequent character at the site. For simplicity, we here do not include the gap character.
The subfigures are ordered by increasing entropy, which ranges from 0 for a site with only a single character, as in (a), to 2 for a site with all four nucleobases equally frequent, as in (h). The maximum possible entropy is given by the base of the logarithm. The choice of base is irrelevant when comparing entropies with each other, as it simply introduces a constant factor.

B. Empirical Datasets

This chapter is derived from parts of the following peer-reviewed open-access publications:

> **Lucas Czech**, Pierre Barbera, and Alexandros Stamatakis. "Methods for Automatic Reference Trees and Multilevel Phylogenetic Placement." *Bioinformatics*, 2018, Volume 35, Issue 7, Pages 1151–1158.
>
> **Lucas Czech** and Alexandros Stamatakis. "Scalable Methods for Analyzing and Visualizing Phylogenetic Placement of Metagenomic Samples." *PLOS ONE*, 2019, Volume 14, Issue 5, Page e0217050.

Text and tables in this chapter were created by Lucas Czech, with the following exception: Pierre Barbera conducted the processing and analysis of the CAMI Challenge data as described in Appendix B.4, and helped to write the text of that section.

Throughout this thesis, we used several real-world empirical datasets to evaluate our methods. We here present details on these datasets and their respective preprocessing. The datasets are as follows:

1. Bacterial Vaginosis (BV) [339]. This small dataset was already analyzed with phylogenetic placement in the original publication. We used it as an example of an established study to compare our results to. It has 220 samples with a total of 15 060 unique sequences.

2. Tara Oceans (TO) [137, 174, 350]. This world-wide sequencing effort of the open oceans provides a rich set of meta-data, such as geographic location,

temperature, and salinity. Unfortunately, the sample analysis for creating the official data repository is still ongoing. We thus were only able to use 370 samples with 27 697 007 unique sequences.

3. Human Microbiome Project (HMP) [160, 250]. This large data repository intends to characterize the human microbiota. It contains 9192 samples from different body sites with a total of 63 221 538 unique sequences. There is additional meta-data such as age and medical history, which is available upon special request. We only used the publicly available meta-data.

4. Mouse Gut [36, 322]. This small dataset is part of the 2nd CAMI Challenge [36], which is a community-driven effort to assess taxonomic profiling methods using a common set of benchmark datasets. We solely used this dataset for the taxonommic evaluation of our PhAT method in Section 3.3.4.

These datasets represent a wide range of environments, number of samples, and sequence lengths; see Table B.1 for details. At the time of writing, about two years after we initially downloaded the data, the TO repository has grown to 1170 samples, while the HMP even published a second phase and now comprises 23 666 samples of the 16S region. This further emphasizes the need for scalable methods to analyze such data (such as the ones presented in this work).

Table B.1: Overview of the dataset dimensions. The "Samples" column shows how many metagenomic samples were used in our evaluations. This might differ from the number of available samples due to filtering as explaind in the text. The subsequent two columns show the number of total sequences over these samples, and the number of unique sequences therein. Note that for the Mouse Gut dataset, the number of total sequences is also reported after the filtering as explained in Section B.4. The last column shows the average length of all sequences in the dataset.

Dataset	Samples	Total Seqs.	Unique Seqs.	Avg. Length
Bacterial Vaginosis	220	426,612	15,060	226
Tara Oceans	370	49,023,231	27,697,007	128
Human Microbiome	9,194	118,701,818	63,221,538	413
Mouse Gut	64	620,882	616,405	240

We used these datasets to evaluate our methods and to exemplify which method is applicable to what kind of data. To this end, the sequences of the datasets were placed on appropriate reference trees, in order to obtain phylogenetic placements that our methods can be applied to. Firstly, for the BV dataset, we used the original set of reference sequences, and re-inferred a tree on them. Secondly, for the TO, HMP, and mouse gut datasets, we used our Phylogenetic Automatic (Reference)

Tree PhAT method (Chapter 3) to construct sets of suitable reference sequences from the SILVA database [294, 395].

For all analyses, we used the following software setup: Unconstrained maximum likelihood trees were inferred using RAXML v8.2.8 [342]. For aligning reads against reference alignments and reference trees, we used a custom `MPI` wrapper for PAPARA 2.0 [23, 24], which we also made freely available [22]. We then applied the `chunkify` procedure (Section 3.2.3) to split the sequences into chunks of unique sequences prior to conducting the phylogenetic placement, in order to minimize processing time. Phylogenetic placement was conducted using EPA-NG [18], which is a faster and more scalable phylogenetic placement implementation than RAXML-EPA [25] and PPLACER [241]. Lastly, given the per-chunk placement files produced by EPA-NG, we executed the `unchunkify` procedure (Section 3.2.3) to obtain per-sample placement files. These subsequently served as the input data for the methods presented in this work.

B.1 Bacterial Vaginosis

We used the Bacterial Vaginosis (BV) dataset [339] as an example of a well-designed study in order to compare our novel methods to existing ones such as Edge PCA and Squash Clustering [104, 239] (Section 2.5.5). The dataset contains metabarcoding sequences of the vaginal microbiome of 220 women, and was kindly provided by Sujatha Srinivasan. This small dataset with a total of 426 612 query sequences, thereof 15 060 unique, was already analyzed with phylogenetic placement methods in the original publication. We re-inferred the reference tree of the original publication using the original alignment, which contains 797 reference sequences specifically selected to represent the vaginal microbiome. As the query sequences were already prepared, no further preprocessing was applied prior to alignment and phylogenetic placement. The query sequences of the dataset were then aligned to our re-inferred reference tree and alignment, and subsequently placed on the tree. The available per-sample quantitative meta-data for this dataset comprise the Nugent score [274], the value of Amsel's criteria [9], and the vaginal pH value. We used all three meta-data types in our analyses.

We first used the BV dataset for testing the accuracy of the unconstrained *Bacteria* tree obtained from the PhAT method; see Section 3.3.3 for details, and see Figure 3.9 and Figure 3.10 for the respective results. For this evaluation, we also placed the BV dataset on the *Bacteria* tree, and compared the results obtained from analyses such as Edge PCA and Squash Clustering to the results obtained on the original reference tree. Next, we used the BV dataset throughout Chapters 4 to 6 for evaluating our methods.

Finally, for our comparison of Placement-Factorization to the original Phylofactorization [376] in Section 7.3.1, we conducted OTU clustering of the BV sequences, using two different methods: We used VSEARCH v2.9.1 [306] as well as SWARM v2.2.2 [228, 229] to obtain two sets of OTU clusters. We filtered the OTU table to remove low abundance OTUs, by only keeping those that appear in more than

10% of the samples. In order to assign each OTU to a fitting taxonomic path, we used the ASSIGN command of our toll GAPPA [70], see Appendix C.2. To this end, we placed the OTUs on the BV reference tree mentioned above, in order to obtain taxonomic assignments for the OTUs that are in line with the taxonomic labels used in our other analyses of the dataset. Each set of OTU sequences was subsequently aligned with MAFFT v7.310 [175, 176], using the L-INS-I strategy [177]. Finally, we inferred an OTU tree for each set, using the recent RAxML-NG v0.7.0 [192]. These two OTU trees were then used with the meta-data for conducting an analysis with PHYLOFACTOR [376], based on the excellent tutorials at `https://github.com/reptalex/phylofactor`.

B.2 Tara Oceans

The Tara Oceans (TO) dataset [137, 174, 350] that we used in this work contains amplicon sequences of protists, and is available at `https://www.ebi.ac.uk/ena/data/view/PRJEB6610`. At the time of download (2016-09-20), there were 370 samples available with a total of 49 023 231 sequences. As the available data are raw `fastq` files, we followed Mahé (2016) [227] to generate cleaned per-sample `fasta` files. For this, we used the tool PEAR [401] to merge the paired-end reads; CUTADAPT [236] for trimming tags as well as forward and reverse primers; and VSEARCH [306] for filtering erroneous sequences and generating per-sample `fasta` files. We filtered out sequences below 95 bps and above 150 bps, to remove potentially erroneous sequences. No further preprocessing (such as chimera detection) was applied.

This resulted in a total of 48 036 019 sequences, thereof 27 697 007 unique. The sequences were then used for phylogenetic placement as explained above. The TO dataset has a rich variety of per-sample meta-data features; in the context of this work, we mainly focus on quantitative features such as temperature, salinity, as well as oxygen, nitrate and chlorophyll content of the water. Furthermore, each sample has meta-data features indicating the date, longitude and latitude, depth, etc. of the sampling location. This data might be interesting for further correlation analyses based on geographical information, as mentioned in Chapter 8.

B.3 Human Microbiome Project

We used the Human Microbiome Project (HMP) dataset [160, 250] as an example of a large-scale dataset, and hence also for testing the scalability of our methods. In particular, we used the "HM16STR" data of the initial phase "HMP1", which are available from `http://www.hmpdacc.org/hmp/`. Each sample is labeled with one of 18 human body site locations where it was sampled. This is the only publicly available meta-data feature. See Table B.2 for an overview of those labels.

The dataset consists of trimmed 16S rRNA sequences of the `V1V3`, `V3V5`, and `V6V9` regions. The data are further divided into a "by_sample" set and a "healthy" set, which we merged in order to obtain one large dataset, with a total of 9811 samples. We then removed 10 samples that were larger than 70 MB as well as 605 samples

that had fewer than 1500 sequences, because we considered them as defective or unreliable outliers. Finally, we also removed 2 samples that did not have a valid body site label assigned to them. This resulted in a set of 9192 samples containing a total of 118 702 967 sequences with an average length of 413 bps. From these samples, sequences with a length of less than 150 bps as well as sequences longer than 540 bps were removed, as we considered them potentially erroneous. No further preprocessing (such as chimera detection) was applied.

This resulted in a total of 116 520 289 sequences, of which 63 221 538 were unique. For most of the evaluation, we then used the unconstrained *Bacteria* tree of our PhAT method [69] for phylogenetic placement; see Section 3.3.1 for details. The tree comprises 1914 taxa, thereof 1797 bacterial sequences. The remaining 117 taxa are *Archaea* and *Eukaryota*, and were included as a broad outgroup.

Table B.2: HMP dataset overview. The table lists the 18 body site labels used by the Human Microbiome Project (HMP) [160, 250], and a "translation" into the corresponding body region. For simplicity, in the evaluations in Section 3.3.3 and Section 5.3.2, we summarized some of the labels into eight location groups, as shown in the third column. The last column lists how many samples from each body site were used in our evaluation.

Body Site	Region	Group	Samples
Stool	Stool	Stool	600
Saliva	Saliva	Saliva	529
Tongue Dorsum	Mouth (back)	Mouth (back)	610
Throat	Mouth (back)	Mouth (back)	638
Palatine Tonsils	Mouth (back)	Mouth (back)	599
Attached Keratinized Gingiva	Mouth (front)	Mouth (front)	600
Hard Palate	Mouth (front)	Mouth (front)	566
Buccal Mucosa	Mouth (front)	Mouth (front)	597
Subgingival Plaque	Plaque	Plaque	595
Supragingival Plaque	Plaque	Plaque	608
Anterior Nares	Nose	Airways	541
Left Antecubital Fossa	Arm	Skin	290
Right Antecubital Fossa	Arm	Skin	328
Left Retroauricular Crease	Ear	Skin	596
Right Retroauricular Crease	Ear	Skin	604
Vaginal Introitus	Vagina	Vagina	292
Mid Vagina	Vagina	Vagina	298
Posterior Fornix	Vagina	Vagina	301
Sum			9192

B.4 Mouse Gut

We utilized the *Mouse Gut* dataset of the 2nd CAMI Challenge [36, 322] to evaluate the accuracy of our PhAT trees for taxonomic assigment in Section 3.3.4. More specifically, we used the unpaired HiSeq reads of the dataset, which comprises 64 samples of simulated reads. The preprocessing involved read de-interleaving following Watson-Haigh (2012) [380], paired-end read merging using PEAR [401], as well as quality filtering and conversion to `fasta` using VSEARCH v2 [306]. This yielded a total of 800 341 409 reads. As our trees are based on small ribosomal subunit sequences, we also performed read filtering to obtain reads from the 16S rDNA region (see Section 2.2.2). This filtering was performed using the protocol of Logares et al. (2014) [219], which relies on HMMER [93, 94], and respective profiles for the 16S rDNA locus. We performed a global identity based de-replication step on the resulting reads that yielded 616 405 unique query sequences. We aligned these query sequences to our *Bacteria* reference alignment using PaPaRa 2.0 [23, 24]. We then performed phylogenetic placement of the aligned query sequences onto the unconstrained and constrained reference trees, respectively, using EPA-ng [18]. We performed de-de-replication to obtain per-sample data again, resulting in 64 `jplace` files (one per original sample) with placements of the 16S rDNA sequences, for each of the two trees (constrained and unconstrained). Finally, we performed taxonomic assignment and taxonomic profiling of the per-sample results using the `assign` command implemented in gappa [70], which works analogously to the method used in Sativa [191]. Its basic steps are described in Appendix C.

C. Software Implementation

This chapter is derived from parts of the following open-access publications:

> **Lucas Czech**, Pierre Barbera, and Alexandros Stamatakis. "Methods for Automatic Reference Trees and Multilevel Phylogenetic Placement." *Bioinformatics*, 2018, Volume 35, Issue 7, Pages 1151–1158.
>
> **Lucas Czech** and Alexandros Stamatakis. "Scalable Methods for Analyzing and Visualizing Phylogenetic Placement of Metagenomic Samples." *PLOS ONE*, 2019, Volume 14, Issue 5, Page e0217050.
>
> **Lucas Czech**, Pierre Barbera, and Alexandros Stamatakis. "Genesis and Gappa: Processing, Analyzing and Visualizing Phylogenetic (Placement) Data." *bioRxiv*, 2019, Page 647958.

All text and figures in this chapter were created by Lucas Czech. The software described here, GENESIS and GAPPA, was developed and written by Lucas Czech, with the exception of the GAPPA command `assign`, which was mainly developed and written by Pierre Barbera.

In this work, we presented several novel methods for analyzing and visualizing phylogenetic placement data. In order to test these methods, evaluate their results, and produce the respective figures shown here, we implemented all of these methods as part of our software library GENESIS. Furthermore, for end users who want to apply our methods to their own data, we provide a tool called GAPPA, which offers a command line interface with sufficient options for most needs. In this chapter, we briefly introduce GENESIS and the relevant commands of GAPPA. For more details, see our application note that describes both tools [70].

GENESIS and GAPPA are freely available under GPLv3 at `http://github.com/lczech/genesis` and `http://github.com/lczech/gappa`. They are already used as an integral part in several of our previous publications and programs [18, 67, 69, 230, 403]. Furthermore, GAPPA is used as part of the PICRUST2 pipeline [88], which however is not developed or maintained by us. These use cases hence prove the flexibility and usefulness of our tools for research and development purposes.

C.1 Overview of Genesis

When developing scientific software, there are several important, yet often times competing design objectives: (a) Most users require a fast and simple application for analysing their data, (b) some power users desire customization, e. g., via scripting etc., and (c) developers require a flexible toolkit for rapid prototyping. At the same time, with the on-going data growth, the implementation needs to be scalable and efficient with respect to both, memory, and execution times. We aimed to meet all of the above objectives. To this end, GENESIS is written in `C++11`, using a modern, modular, and function-centric software design.

GENESIS is a highly flexible library for reading, manipulating, and evaluating phylogenetic data, and in particular phylogenetic placement data. It has a simple and straight forward API, but is also computationally highly efficient. Typical tasks such as parsing and writing files, iterating over the elements of a data structure, and other frequently used functions are mostly one-liners that integrate well with modern `C++` and its standard library `STL`. Where possible, the library is multi-threaded, allowing for fully leveraging the computational power of multi-core systems. Hence, GENESIS allows for scalable parsing and processing of huge datasets. The functionality is divided into loosely coupled modules, which are described in more detail in the application note [70] and the online documentation.

We evaluate the code quality, the runtime behavior, and the memory requirements for conducting typical tasks such as file parsing and data processing in the application note [70]. We find that GENESIS has the overall best code quality score compared to other scientific codes written in C or C++. It is also consistently faster than all evaluated Python and R libraries in our tests, and in most of the tests, more memory efficient as well.

C.2 Commands of Gappa

The flexibility of a library such as GENESIS is primarily useful for method developers. For most users, it is however more convenient to have a simple interface for typical,

frequent tasks. To this end, we have developed the command line program GAPPA, which we present in the following.

GAPPA is short for "Genesis Applications for Phylogenetic Placement Analyses". Its original purpose was to implement and make available the methods we presented in this work, see also Czech et al. (2018) [69] and Czech and Stamatakis (2019) [67]. GAPPA has since been substantially extended and now also contains a multitude of other auxiliary commands, as well as re-implementations of some prominent methods of GUPPY [241]. These re-implementations are far more computationally efficient than the original, which was necessary in order to be able to apply them for the large datasets that we used in this work.

The methods that we described in this work are implemented via the following sub-commands of GAPPA:

- `phat`: Phylogenetic Automatic (Reference) Tree method, see Section 3.2.1. The command expects a taxonomy file and an alignment file from a sequence database, e. g., SILVA [294, 395], as well as the target number of consensus sequences to be generated for the intended phylogeny. The result is a `fasta` file with consensus sequences representing taxonomic clades. The command can be further customized, e. g., by changing the consensus sequence method, using only a specified subclade of the taxonomy for running the algorithm, as well as several detail settings for the method. It can also output additional info files that allow to inspect details of the calculations, like the number of sequences and their entropy per clade.

- `extract`: Extract/collect placements in specific sub-clades of the tree. The command performs the main step of the multilevel placement approach, see Section 3.2.2. Its input is a set of `jplace` files containing placements on the backbone tree, as well as a file listing the clade name that each taxon of the backbone tree belongs to. For each clade, it then writes a new `jplace` file, containing all queries that were placed in that clade with more than a customizable threshold of their placement mass.
  Furthermore, if provided with the sequence files that were used to make the input `jplace` files, the corresponding sequence of each query are also written to `fasta` files per clade. Thus, a per-clade collection of sequences is created, where each result file contains the sequences that were placed in this clade of the backbone tree. These can then be used for the second level placement on separate clade-specific trees.

- `chunkify`: Split a set of `fasta` files into chunks of equal size, and write abundance maps. This is the first step of the preprocessing pipeline as described in Section 3.2.3. The command re-names the sequences using a configurable hash function (`MD5`, `SHA1` or `SHA256`), and de-duplicates across all input sequences. Its output are chunk files of sequences, as well as an abundance map file for each input sequences file. The sequence chunk files can then be used to perform phylogenetic placement to obtain per-chunk `jplace` files.

- `unchunkify`: Take the per-chunk `jplace` files as well as the abundance map files, and generate a `jplace` for each original sequence file, including the correct abundances. This command is the second step of the `chunkify` command, and reverts its effect, so that the resulting `jplace` files are as if they were created using the original sequence files.

- `assign`: Perform taxonomic assignment using phylogenetic placements, which was used for the evaluation in Section 3.3.4. The command uses a taxonomic labeling of the tips of the reference tree to annotate all inner branches with the longest common taxonomic label for the induced subtree of the inner branch, in analogy to SATIVA [191]. Then, each query sequence in the provided `jplace` files is taxonomically assigned according to the labels of the branches where it does have placement mass. This can subsequently either be used for taxonomic assignment of the query sequences themselves, or to obtain a taxonomic profile of one or more samples.

- `dispersion`: The command implements Edge Dispersion as described in Section 4.2.1. It takes a set of `jplace` files (the samples) as input, and calculates and visualizes the Edge Dispersion per edge of the reference tree. For this, it offers different modes to measure the dispersion, such as the standard deviation or the coefficient of variation, and log-scaled variants thereof, as explained in the main text.

- `correlation`: This command implements Edge Correlation, see Section 4.2.2. The command takes a set of `jplace` samples, as well as a table containing meta-data features for each sample. It then calculates and visualizes the Edge Correlation with the metadata features per edge of the reference tree. The command offers options to chose the type of values to use (masses or imbalances), as well as the correlation measure (Pearson Correlation Coefficient or Spearman's Rank Correlation Coefficient).

- `phylogenetic-kmeans` and `imbalance-kmeans`: Performs k-means clustering of a set of `jplace` files according to the methods described in Section 5.2.1 and Section 5.2.2. The commands output information such as the cluster assignments and visualizations of the centroid trees, but also allow for clustering with different values for k in order to create, e. g., the data for Elbow plots.

- `placement-factorization`: Performs our adaptation of Phylofactorization [376] to phylogenetic placement data as explained in Section 7.2.1, and outputs all relevant analysis results. The command takes a set of `jplace` samples as well as a per-sample meta-data table, and computes a given number of phylogenetic factors. The meta-data can be of many different types, such as numerical or boolean values, or categorical data. It outputs all relevant data and visualizations as shown in this work, for example, the edges that are factored out in each step, as well as the visualization of the values of the objective function on the reference tree.

- `squash` and `edgepca`: Re-implementations of the two existing methods [104, 239] as introduced in Section 2.5.5. As mentioend above, the original implementation in GUPPY [241] was not efficient enough for conducting the large scale analyses that we needed for this work. We hence offer these commands in GAPPA, which scale to larger datasets.

These are the GAPPA commands that are relevant in the context of this work. For more details and a full list of the available commands, see `http://github.com/lczech/gappa`. Furthermore, we provide prototype implementations, scripts, data, and other tools used for the tests and figures in this work at `http://github.com/lczech/placement-methods-paper`.

Bibliography

[1] K. Abarenkov, R. Henrik Nilsson, K.-H. Larsson, I. J. Alexander, U. Eberhardt, S. Erland, K. Høiland, R. Kjøller, E. Larsson, T. Pennanen, R. Sen, A. F. S. Taylor, L. Tedersoo, B. M. Ursing, T. Vrålstad, K. Liimatainen, U. Peintner, and U. Kõljalg. The UNITE database for molecular identification of fungi–recent updates and future perspectives. *New Phytologist*, 186(2): 281–285, 2010.

[2] C. Abbosh, N. J. Birkbak, G. A. Wilson, M. Jamal-Hanjani, T. Constantin, R. Salari, J. L. Quesne, D. A. Moore, S. Veeriah, R. Rosenthal, T. Marafioti, E. Kirkizlar, T. B. K. Watkins, N. McGranahan, S. Ward, L. Martinson, J. Riley, F. Fraioli, M. A. Bakir, E. Grönroos, F. Zambrana, R. Endozo, W. L. Bi, F. M. Fennessy, N. Sponer, D. Johnson, J. Laycock, S. Shafi, J. Czyzewska-Khan, A. Rowan, T. Chambers, N. Matthews, S. Turajlic, C. Hiley, S. M. Lee, M. D. Forster, T. Ahmad, M. Falzon, E. Borg, D. Lawrence, M. Hayward, S. Kolvekar, N. Panagiotopoulos, S. M. Janes, R. Thakrar, A. Ahmed, F. Blackhall, Y. Summers, D. Hafez, A. Naik, A. Ganguly, S. Kareht, R. Shah, L. Joseph, A. M. Quinn, P. A. Crosbie, B. Naidu, G. Middleton, G. Langman, S. Trotter, M. Nicolson, H. Remmen, K. Kerr, M. Chetty, L. Gomersall, D. A. Fennell, A. Nakas, S. Rathinam, G. Anand, S. Khan, P. Russell, V. Ezhil, B. Ismail, M. Irvin-Sellers, V. Prakash, J. F. Lester, M. Kornaszewska, R. Attanoos, H. Adams, H. Davies, D. Oukrif, A. U. Akarca, J. A. Hartley, H. L. Lowe, S. Lock, N. Iles, H. Bell, Y. Ngai, G. Elgar, Z. Szallasi, R. F. Schwarz, J. Herrero, A. Stewart, S. A. Quezada, K. S. Peggs, P. V. Loo, C. Dive, C. J. Lin, M. Rabinowitz, H. J. W. L. Aerts, A. Hackshaw, J. A. Shaw, B. G. Zimmermann, The TRACERx Consortium, The PEACE Consortium, and C. Swanton. Phylogenetic ctDNA analysis depicts early-stage lung cancer evolution. *Nature*, 545(7655):446–451, 2017.

[3] A. J. Aberer, K. Kobert, and A. Stamatakis. ExaBayes: Massively Parallel Bayesian Tree Inference for the Whole-Genome Era. *Molecular Biology and Evolution*, 31(10):2553–2556, 2014.

[4] E. Afgan, D. Baker, B. Batut, M. van den Beek, D. Bouvier, M. Čech, J. Chilton, D. Clements, N. Coraor, B. A. Grüning, A. Guerler, J. Hillman-Jackson, S. Hiltemann, V. Jalili, H. Rasche, N. Soranzo, J. Goecks, J. Taylor, A. Nekrutenko, and D. Blankenberg. The Galaxy platform for accessible, re-

producible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Research*, 46(W1):W537–W544, 2018.

[5] A. Agresti. *An Introduction to Categorical Data Analysis*. Wiley-Interscience, 3rd edition, 2018.

[6] J. Aitchison. *The statistical analysis of compositional data*. Chapman and Hall London, 1986.

[7] A. Almeida, A. L. Mitchell, A. Tarkowska, and R. D. Finn. Benchmarking taxonomic assignments based on 16S rRNA gene profiling of the microbiota from commonly sampled environments. *GigaScience*, 7(5), 2018.

[8] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[9] R. Amsel, P. A. Totten, C. A. Spiegel, K. C. S. Chen, D. Eschenbach, and K. K. Holmes. Nonspecific vaginitis: Diagnostic Criteria and Microbial and Epidemiologic Associations. *The American Journal of Medicine*, 74(1):14–22, 1983.

[10] S. Anderson. Shotgun DNA sequencing using cloned DNase I-generated fragments. *Nucleic Acids Research*, 9(13):3015–3027, 1981.

[11] C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle. Deep learning for computational biology. *Molecular Systems Biology*, 12(7):878, 2016.

[12] J. Archie, W. H. Day, W. Maddison, C. Meacham, F. J. Rohlf, D. Swofford, and J. Felsenstein. The Newick tree format, 1986. Online: `http://evolution.genetics.washington.edu/phylip/newicktree.html`. Accessed: 2015-07-26.

[13] D. Arthur and S. Vassilvitskii. How Slow is the K-means Method? In *Proceedings of the Twenty-second Annual Symposium on Computational Geometry*, SCG '06, pages 144–153, New York, NY, USA, 2006. ACM.

[14] D. Arthur and S. Vassilvitskii. k-means++ : The Advantages of Careful Seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.*, pages 1027–1035. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2007.

[15] E. Asgari, K. Garakani, and M. R. K. Mofrad. A New Approach for Scalable Analysis of Microbial Communities. 2015.

[16] M. Balaban, S. Sarmashghi, and S. Mirarab. APPLES: Fast Distance-Based Phylogenetic Placement. *bioRxiv,* page 475566, 2018, and *Systematic Biology,* in press, 2019.

[17] M. Balvočiūtė and D. H. Huson. SILVA, RDP, Greengenes, NCBI and OTT — how do these taxonomies compare? *BMC Genomics*, 18(2):114, 2017.

[18] P. Barbera, A. M. Kozlov, L. Czech, B. Morel, D. Darriba, T. Flouri, and A. Stamatakis. EPA-ng: Massively Parallel Evolutionary Placement of Genetic Sequences. *Systematic Biology*, 68(2):365–369, 2018.

[19] J. M. S. Bartlett and D. Stirling. *A Short History of the Polymerase Chain Reaction*. PCR Protocols, Methods in Molecular Biology, volume 226, 2003.

[20] D. Bass, L. Czech, B. A. P. Williams, C. Berney, M. Dunthorn, F. Mahe, G. Torruella, G. D. Stentiford, and T. A. Williams. Clarifying the Relationships between Microsporidia and Cryptomycota. *Journal of Eukaryotic Microbiology*, 65(6):773–782, 2018.

[21] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers. GenBank. *Nucleic Acids Research*, 37(Database):D26–D31, 2009.

[22] S. Berger and L. Czech. PaPaRa 2.0 with MPI, 2016. Online: `https://github.com/lczech/papara_nt`. Accessed: 2017-11-04.

[23] S. Berger and A. Stamatakis. Aligning short reads to reference alignments and trees. *Bioinformatics*, 27(15):2068–2075, 2011.

[24] S. Berger and A. Stamatakis. PaPaRa 2.0: A Vectorized Algorithm for Probabilistic Phylogeny-Aware Alignment Extension. Technical report, Heidelberg Institute for Theoretical Studies, Heidelberg, 2012.

[25] 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.

[26] S. A. Berger and A. Stamatakis. Evolutionary Placement of Short Sequence Reads. *arXiv*, page 0911.2852, 2009.

[27] C. Berney, A. Ciuprina, S. Bender, J. Brodie, V. Edgcomb, E. Kim, J. Rajan, L. W. Parfrey, S. Adl, S. Audic, D. Bass, D. A. Caron, G. Cochrane, L. Czech, M. Dunthorn, S. Geisen, F. O. Glöckner, F. Mahé, C. Quast, J. Z. Kaye, A. G. B. Simpson, A. Stamatakis, J. del Campo, P. Yilmaz, and C. de Vargas. UniEuk: Time to Speak a Common Language in Protistology! *Journal of Eukaryotic Microbiology*, 38(1):42–49, 2017.

[28] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Advanced applications in pattern recognition. Plenum Press, 1981.

[29] H. Bischof, A. Leonardis, and A. Selb. MDL Principle for Robust Vector Quantisation. *Pattern Analysis & Applications*, 2(1):59–72, 1999.

[30] B. E. Blaisdell. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proceedings of the National Academy of Sciences*, 83(14): 5155–5159, 1986.

[31] M. Blaxter, J. Mann, T. Chapman, F. Thomas, C. Whitton, R. Floyd, and E. Abebe. Defining operational taxonomic units using DNA barcode data. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 360(1462):1935–43, 2005.

[32] S. A. Bloom. Similarity Indices in Community Studies: Potential Pitfalls. *Marine Ecology Progress Series*, 5(2):125–128, 1981.

[33] L. Bottou and Y. Bengio. Convergence properties of the k-means algorithms. In *Advances in neural information processing systems*, pages 585–592, 1995.

[34] J. R. Bray and J. T. Curtis. An ordination of the upland forest communities of southern Wisconsin. *Ecological Monographs*, 27(4):325–349, 1957.

[35] T. Bray. The JavaScript Object Notation (JSON) Data Interchange Format, RFC, 2014. Online: `https://tools.ietf.org/html/rfc7159`. Accessed: 2018-08-14.

[36] A. Bremges and A. C. McHardy. Critical Assessment of Metagenome Interpretation Enters the Second Round. *mSystems*, 3(4), 2018.

[37] B. Brenner, M. A. Wainberg, and M. Roger. Phylogenetic inferences on HIV-1 transmission: implications for the design of prevention and treatment interventions. *AIDS*, 27(7):1045–57, 2013.

[38] R. P. Brent. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4):422–425, 1971.

[39] M. Bronstein, X. Bresson, Y. Lecun, A. Szlam, and J. Bruna. Geometric Deep Learning. *IEEE Signal Processing Magazine*, pages 18–42, 2017.

[40] D. Brown, D. Smeets, B. Székely, D. Larsimont, A. M. Szász, P.-Y. Adnet, F. Rothé, G. Rouas, Z. I. Nagy, Z. Faragó, A.-M. Tőkés, M. Dank, G. Szentmártoni, N. Udvarhelyi, G. Zoppoli, L. Pusztai, M. Piccart, J. Kulka, D. Lambrechts, C. Sotiriou, and C. Desmedt. Phylogenetic analysis of metastatic progression in breast cancer using somatic mutations and copy number aberrations. *Nature Communications*, 8:14944, 2017.

[41] S. M. Brown, Y. Hao, H. Chen, B. P. Laungani, T. A. Ali, C. Dong, C. Lijeron, B. Kim, K. Krampis, and Z. Pei. Fast functional annotation of metagenomic shotgun data by DNA alignment to a microbial gene catalog. *bioRxiv*, page 120402, 2017.

[42] M. Brudno, S. Malde, A. Poliakov, C. B. Do, O. Couronne, I. Dubchak, and S. Batzoglou. Glocal alignment: finding rearrangements during alignment. *Bioinformatics*, 19(Suppl 1):i54–i62, 2003.

[43] C. S. Burrus. Iterative Reweighted Least Squares. *OpenStax-CNX*, 2012.

[44] B. J. Callahan, P. J. McMurdie, and S. P. Holmes. Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. *The ISME Journal*, 11(12):2639–2643, 2017.

[45] J. G. Caporaso, C. L. Lauber, E. K. Costello, D. Berg-Lyons, A. Gonzalez, J. Stombaugh, D. Knights, P. Gajer, J. Ravel, N. Fierer, J. I. Gordon, and R. Knight. Moving pictures of the human microbiome. *Genome Biology*, 12 (5):R50, 2011.

[46] G. Cardona, F. Rosselló, and G. Valiente. Extended Newick: it is time for a standard representation of phylogenetic networks. *BMC Bioinformatics*, 9: 532, 2008.

[47] H. Carroll, P. Ridge, M. Clement, and Q. Snell. Effects of Gap Open and Gap Extension Penalties. *International Journal of Bioinformatics Research And Applications*, 2007.

[48] E. Castro-Nallar, M. Pérez-Losada, G. F. Burton, and K. A. Crandall. The evolution of HIV: Inferences using phylogenetics. *Molecular Phylogenetics and Evolution*, 62(2):777–92, 2012.

[49] T. Cavalier-Smith. The neomuran origin of archaebacteria, the negibacterial root of the universal tree and bacterial megaclassification. *International Journal of Systematic and Evolutionary Microbiology*, 52(1):7–76, 2002.

[50] D. R. Cavener. Comparison of the consensus sequence flanking translational start sites in Drosophila and vertebrates. *Nucleic Acids Research*, 15(4), 1987.

[51] D. R. Cavener and S. C. Ray. Eukaryotic start and stop translation sites. *Nucleic Acids Research*, 19(12):3185–3192, 1991.

[52] S. Chaffron, H. Rehrauer, J. Pernthaler, and C. von Mering. A global network of coexisting microbes from environmental and whole-genome sequence data. *Genome Research*, 20(7):947–59, 2010.

[53] S. Chatterjee, D. Koslicki, S. Dong, N. Innocenti, L. Cheng, Y. Lan, M. Vehkapera, M. Skoglund, L. K. Rasmussen, E. Aurell, and J. Corander. SEK: sparsity exploiting k-mer-based estimation of bacterial community composition. *Bioinformatics*, 30(17):2423–2431, 2014.

[54] W. Chen, C. K. Zhang, Y. Cheng, S. Zhang, and H. Zhao. A comparison of methods for clustering 16S rRNA sequences into OTUs. *PloS ONE*, 8(8): e70837, 2013.

[55] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman, W. Xie, G. L. Rosen, B. J. Lengerich, J. Israeli, J. Lanchantin, S. Woloszynek, A. E.

Carpenter, A. Shrikumar, J. Xu, E. M. Cofer, C. A. Lavender, S. C. Turaga, A. M. Alexandari, Z. Lu, D. J. Harris, D. DeCaprio, Y. Qi, A. Kundaje, Y. Peng, L. K. Wiley, M. H. S. Segler, S. M. Boca, S. J. Swamidass, A. Huang, A. Gitter, and C. S. Greene. Opportunities and obstacles for deep learning in biology and medicine. *Journal of the Royal Society, Interface*, 15(141): 20170387, 2018.

[56] B. Chor and T. Tuller. Maximum Likelihood of Evolutionary Trees Is Hard. In S. Miyano, J. Mesirov, S. Kasif, S. Istrail, P. A. Pevzner, and M. Waterman, editors, *Research in Computational Molecular Biology*, pages 296–310, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[57] P. J. A. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research*, 38(6):1767–1771, 2009.

[58] J. R. Cole, Q. Wang, J. A. Fish, B. Chai, D. M. McGarrell, Y. Sun, C. T. Brown, A. Porras-Alfaro, C. R. Kuske, and J. M. Tiedje. Ribosomal database project: data and tools for high throughput rRNA analysis. *Nucleic Acids Research*, 42, 2014.

[59] M. Comin and D. Verzotto. Alignment-free phylogeny of whole genomes using underlying subwords. *Algorithms for Molecular Biology: AMB*, 7(1):34, 2012.

[60] E. K. Costello, C. L. Lauber, M. Hamady, N. Fierer, J. I. Gordon, and R. Knight. Bacterial Community Variation in Human Body Habitats Across Space and Time. *Science*, 326(5960):1694–1697, 2009.

[61] C. B. Cox, I. N. Healey, and P. D. Moore. *Biogeography: An Ecological and Evolutionary Approach*. Wiley-Blackwell, 9th edition, 2016.

[62] F. Crick. Central Dogma of Molecular Biology. *Nature*, 227(5258):561–563, 1970.

[63] F. H. C. Crick. On Protein Synthesis. In *Symposia of the Society for Experimental Biology*, volume 12, page 8, 1958.

[64] A. Criscuolo and S. Gribaldo. BMGE (Block Mapping and Gathering with Entropy): a new software for selection of phylogenetic informative regions from multiple sequence alignments. *BMC Evolutionary Biology*, 10(1):210, 2010.

[65] D. Crockford. The application/json Media Type for JavaScript Object Notation (JSON), RFC, 2006. Online: `https://tools.ietf.org/html/rfc4627`. Accessed: 2018-08-14.

[66] L. Czech. DNA double helix and nucleobases, 2018. The image is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license, see `https://creativecommons.org/licenses/by-sa/3.0/deed.en`. It is based on [393] and [394], and parts of [338], which itself is based on [183]. We changed

some colors, added background colors, and connected the original images with boxes and lines.

[67] L. Czech and A. Stamatakis. Scalable methods for analyzing and visualizing phylogenetic placement of metagenomic samples. *PLOS ONE*, 14(5):e0217050, 2019.

[68] L. Czech, J. Huerta-Cepas, and A. Stamatakis. A Critical Review on the Use of Support Values in Tree Viewers and Bioinformatics Toolkits. *Molecular Biology and Evolution*, 17(4):383–384, 2017.

[69] L. Czech, P. Barbera, and A. Stamatakis. Methods for Automatic Reference Trees and Multilevel Phylogenetic Placement. *Bioinformatics*, 35(7):1151–1158, 2018.

[70] L. Czech, P. Barbera, and A. Stamatakis. Genesis and Gappa: Processing, Analyzing and Visualizing Phylogenetic (Placement) Data. *bioRxiv*, page 647958, 2019.

[71] A. E. Darling, G. Jospin, E. Lowe, F. A. Matsen, H. M. Bik, and J. a. Eisen. PhyloSift: phylogenetic analysis of genomes and metagenomes. *PeerJ*, 2:e243, 2014.

[72] C. Darwin. First diagram of an evolutionary tree, from the first notebook on Transmutation of Species, 1837. Online: `https://en.wikipedia.org/wiki/File:Darwin_tree.png`. Accessed: 2018-08-01. The image is public domain due to its age.

[73] C. Darwin. *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. John Murray, 1859.

[74] R. Dawkins. *The Selfish Gene*. Oxford University Press, Oxford, 1989.

[75] W. H. E. Day and F. R. McMorris. Threshold consensus methods for molecular sequences. *Journal of Theoretical Biology*, 159(4):481–489, 1992.

[76] W. H. E. Day and F. R. McMorris. Critical comparison of consensus methods for molecular sequences. *Nucleic Acids Research*, 20(5):1093–1099, 1992.

[77] M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 5:345–352, 1978.

[78] C. de Vargas, S. Audic, N. Henry, J. Decelle, F. Mahe, R. Logares, E. Lara, C. Berney, N. Le Bescot, I. Probert, M. Carmichael, J. Poulain, S. Romac, S. Colin, J.-M. Aury, L. Bittner, S. Chaffron, M. Dunthorn, S. Engelen, O. Flegontova, L. Guidi, A. Horak, O. Jaillon, G. Lima-Mendez, J. Luke, S. Malviya, R. Morard, M. Mulot, E. Scalco, R. Siano, F. Vincent, A. Zingone, C. Dimier, M. Picheral, S. Searson, S. Kandels-Lewis, S. G. Acinas, P. Bork, C. Bowler, G. Gorsky, N. Grimsley, P. Hingamp, D. Iudicone, F. Not, H. Ogata, S. Pesant,

J. Raes, M. E. Sieracki, S. Speich, L. Stemmann, S. Sunagawa, J. Weissenbach, P. Wincker, E. Karsenti, E. Boss, M. Follows, L. Karp-Boss, U. Krzic, E. G. Reynaud, C. Sardet, M. B. Sullivan, and D. Velayoudon. Eukaryotic plankton diversity in the sunlit ocean. *Science*, 348(6237):1261605–1261605, 2015.

[79] G. De'ath. Multivariate Regression Trees: A new technique for modeling species–environment relationships. *Ecology*, 83(4):1105–1117, 2002.

[80] K. Deiner, H. M. Bik, E. Mächler, M. Seymour, A. Lacoursière-Roussel, F. Altermatt, S. Creer, I. Bista, D. M. Lodge, N. de Vere, M. E. Pfrender, and L. Bernatchez. Environmental DNA metabarcoding: Transforming how we survey animal and plant communities. *Molecular Ecology*, 26:5872–5895, 2017.

[81] N. Desai, D. Antonopoulos, J. A. Gilbert, E. M. Glass, and F. Meyer. From genomics to metagenomics. *Current Opinion in Biotechnology*, 23(1):72–76, 2012.

[82] T. Z. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E. L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G. L. Andersen. Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Applied and Environmental Microbiology*, 72(7):5069–5072, 2006.

[83] L. R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, 1945.

[84] T. Dobzhansky. Nothing in Biology Makes Sense Except in the Light of Evolution. *The American Biology Teacher*, 35(3):125–129, 1973.

[85] M. S. Dodd, D. Papineau, T. Grenne, J. F. Slack, M. Rittner, F. Pirajno, J. O'Neil, and C. T. S. Little. Evidence for Early Life in Earth's Oldest Hydrothermal Vent Precipitates. *Nature*, 543(7643):60, 2017.

[86] M. A. Donk. Typification and Later Starting-Points. *Taxon*, 6(9):245, 1957.

[87] G. M. Douglas. Bioconda recipe for gappa, 2018. Online: `https://bioconda.github.io/recipes/gappa/README.html` and `https://anaconda.org/bioconda/gappa`. Accessed: 2019-06-20.

[88] G. M. Douglas, V. J. Maffei, J. Zaneveld, S. N. Yurgel, J. R. Brown, C. M. Taylor, C. Huttenhower, and M. G. I. Langille. PICRUSt2: An improved and extensible approach for metagenome inference. *bioRxiv*, page 672295, 2019.

[89] J. C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

[90] J. C. Dunning Hotopp. Horizontal gene transfer between bacteria and animals. *Trends in Genetics: TIG*, 27(4):157–63, 2011.

[91] M. Dunthorn, J. Otto, S. A. Berger, A. Stamatakis, F. Mahé, S. Romac, C. De Vargas, S. Audic, A. Stock, F. Kauff, T. Stoeck, B. Edvardsen, R. Massana, F. Not, N. Simon, and A. Zingone. Placing environmental next-generation sequencing amplicons from microbial eukaryotes into a phylogenetic context. *Molecular Biology and Evolution*, 31(4):993–1009, 2014.

[92] A. Ö. C. Dupont, R. I. Griffiths, T. Bell, and D. Bass. Differences in soil micro-eukaryotic communities over soil pH gradients are strongly driven by parasites and saprotrophs. *Environmental Microbiology*, 18(6):2010–2024, 2016.

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

[94] S. R. Eddy. A new generation of homology search tools based on probabilistic inference. In *Genome Informatics*, volume 23, pages 205–211. World Scientific, 2009.

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

[96] R. C. Edgar. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, 26(19):2460–2461, 2010.

[97] D. J. Edwards and K. E. Holt. Beginner's guide to comparative bacterial genome analysis using next-generation sequence data. *Microbial Informatics and Experimentation*, 3(1):2, 2013.

[98] B. Eer and W. Ine. If you read this, write me an email and next time we meet, I'll buy you a drink. *Lucas Czech*, 2019.

[99] B. Efron. Bootstrap methods: another look at the jacknife. *The Annals of Statistics*, 7(1):1–26, 1979.

[100] J. J. Egozcue and V. Pawlowsky-Glahn. Groups of Parts and Their Balances in Compositional Data Analysis. *Mathematical Geology*, 37(7):795–828, 2005.

[101] J. J. Egozcue and V. Pawlowsky-Glahn. Changing the Reference Measure in the Simplex and its Weighting Effects. *Austrian Journal of Statistics*, 45(4): 25, 2016.

[102] J. J. Egozcue, V. Pawlowsky-Glahn, G. Mateu-Figueras, and C. Barceló-Vidal. Isometric Logratio Transformations for Compositional Data Analysis. *Mathematical Geology*, 35(3):279–300, 2003.

[103] A. Escobar-Zepeda, A. Vera-Ponce De León, and A. Sanchez-Flores. The road to metagenomics: From microbiology to DNA sequencing technologies and bioinformatics. *Frontiers in Genetics*, 6(348):1–15, 2015.

[104] S. N. Evans and F. A. Matsen. The phylogenetic Kantorovich-Rubinstein metric for environmental sequence samples. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 74:569–592, 2012.

[105] B. S. Everitt and A. Skrondal. *The Cambridge Dictionary of Statistics*. Cambridge University Press, 4th edition, 2010.

[106] D. P. Faith. Conservation evaluation and phylogenetic diversity. *Biological Conservation*, 61:1–10, 1992.

[107] K. Faust and J. Raes. Microbial interactions: from networks to models. *Nature Reviews Microbiology*, 10(8):538–550, 2012.

[108] K. Faust, L. Lahti, D. Gonze, W. M. de Vos, and J. Raes. Metagenomics meets time series analysis: unraveling microbial community dynamics. *Current Opinion in Microbiology*, 25:56–66, 2015.

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

[110] J. Felsenstein. Confidence Limits on Phylogenies: An Approach Using the Bootstrap. *Evolution*, 39(4):783–791, 1985.

[111] J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates Sunderland, MA, 2nd edition, 2004.

[112] A. Fiannaca, L. La Paglia, M. La Rosa, G. Lo Bosco, G. Renda, R. Rizzo, S. Gaglio, and A. Urso. Deep learning models for bacteria taxonomic classification of metagenomic data. *BMC Bioinformatics*, 19(S7):198, 2018.

[113] A. Filipski, K. Tamura, P. Billing-Ross, O. Murillo, and S. Kumar. Phylogenetic placement of metagenomic reads using the minimum evolution principle. *BMC Genomics*, 16(1):6947, 2015.

[114] D. Fioravanti, Y. Giarratano, V. Maggio, C. Agostinelli, M. Chierici, G. Jurman, and C. Furlanello. Phylogenetic convolutional neural networks in metagenomics. *BMC Bioinformatics*, 19(2):49, 2018.

[115] R. Fletcher. *Practical Methods of Optimization*. Wiley, 1987.

[116] T. Flouri, J. Zhang, L. Czech, K. Kobert, and A. Stamatakis. An Efficient Approach to Merging Paired-End Reads and Incorporation of Uncertainties. In M. Elloumi, editor, *Algorithms for Next-Generation Sequencing Data*, chapter 13, pages 299–326. Springer International Publishing AG, 1st edition, 2017.

[117] E. W. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21(3):768–769, 1965.

[118] D. J. Futuyma. The Uses of Evolutionary Biology. *Science*, 267(5194):41–2, 1995.

[119] E. Gaba. A phylogenetic tree of living things, based on RNA data and proposed by Carl Woese, showing the separation of bacteria, archaea, and eukaryotes, 2006. Online: `https://en.wikipedia.org/wiki/File:Phylogenetic_tree.svg`. Accessed: 2018-08-01. The image was released into the public domain.

[120] P. A. Gagniuc. *Markov Chains: From Theory to Implementation and Experimentation*. Wiley, 1st edition, 2017.

[121] N. M. Gericke and M. Hagberg. Definition of historical models of gene function and their relation to students' understanding of genetics. *Science & Education*, 16(7-8):849–881, 2007.

[122] M. Gevrey, I. Dimopoulos, and S. Lek. Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological Modelling*, 160(3):249–264, 2003.

[123] C. R. Giner, I. Forn, S. Romac, R. Logares, and C. D. Vargas. Environmental Sequencing Provides Reasonable Estimates of the Relative Abundance of Specific Picoeukaryotes. *Applied and Environmental Microbiology*, 82(15): 4757–4766, 2016.

[124] P. D. Gingerich. Evolution and the fossil record: patterns, rates, and processes. *Canadian Journal of Zoology*, 65(5):1053–1060, 1987.

[125] E. M. Glass, J. Wilkening, A. Wilke, D. Antonopoulos, and F. Meyer. Using the Metagenomics RAST Server (MG-RAST) for Analyzing Shotgun Metagenomes. *Cold Spring Harbor Protocols*, 2010(1):pdb.prot5368, 2010.

[126] S. I. Glassman and J. B. H. Martiny. Broadscale Ecological Patterns Are Robust to Use of Exact Sequence Variants versus Operational Taxonomic Units. *mSphere*, 3(4), 2018.

[127] G. B. Gloor, J. M. Macklaim, M. Vu, and A. D. Fernandes. Compositional uncertainty should not be ignored in high-throughput sequencing data analysis. *Austrian Journal of Statistics*, 45(4):73, 2016.

[128] G. B. Gloor, J. R. Wu, V. Pawlowsky-Glahn, and J. J. Egozcue. It's all relative: analyzing microbiome data as compositions. *Annals of Epidemiology*, 26(5): 322–9, 2016.

[129] G. B. Gloor, J. M. Macklaim, V. Pawlowsky-Glahn, and J. J. Egozcue. Microbiome Datasets Are Compositional: And This Is Not Optional. *Frontiers in Microbiology*, 8:2224, 2017.

[130] I. J. Good. On the Estimation of Small Frequencies in Contingency Tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, 18(1):113–124, 1956.

[131] S. Goodwin, J. D. McPherson, and W. R. McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333–351, 2016.

[132] N. J. Gotelli and R. K. Colwell. Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. *Ecology Letters*, 4(4):379–391, 2001.

[133] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, Reading, 1989.

[134] S. Gran-Stadniczeñko, L. Šupraha, E. D. Egge, and B. Edvardsen. Haptophyte Diversity and Vertical Distribution Explored by 18S and 28S Ribosomal RNA Gene Metabarcoding and Scanning Electron Microscopy. *Journal of Eukaryotic Microbiology*, pages 1–19, 2017.

[135] A. J. F. Griffiths, J. H. Miller, D. T. Suzuki, and R. C. Lewontin. *An Introduction to Genetic Analysis*. W.H. Freeman, 2000.

[136] B. Grüning, R. Dale, A. Sjödin, B. A. Chapman, J. Rowe, C. H. Tomkins-Tinch, R. Valieris, J. Köster, and The Bioconda Team. Bioconda: A sustainable and comprehensive software distribution for the life sciences. *Nature Methods*, 15(7):475–476, 2018.

[137] L. Guidi, S. Chaffron, L. Bittner, D. Eveillard, A. Larhlimi, S. Roux, Y. Darzi, S. Audic, L. Berline, J. Brum, L. P. Coelho, J. C. I. Espinoza, S. Malviya, S. Sunagawa, C. Dimier, S. Kandels-Lewis, M. Picheral, J. Poulain, S. Searson, Tara Oceans coordinators, L. Stemmann, F. Not, P. Hingamp, S. Speich, M. Follows, L. Karp-Boss, E. Boss, H. Ogata, S. Pesant, J. Weissenbach, P. Wincker, S. G. Acinas, P. Bork, C. de Vargas, D. Iudicone, M. B. Sullivan, J. Raes, E. Karsenti, C. Bowler, and G. Gorsky. Plankton networks driving carbon export in the oligotrophic ocean. *Nature*, 532(7600):465–470, 2016.

[138] L. Guillou, D. Bachar, S. Audic, D. Bass, C. Berney, L. Bittner, C. Boutte, G. Burgaud, C. de Vargas, J. Decelle, J. del Campo, J. R. Dolan, M. Dunthorn, B. Edvardsen, M. Holzmann, W. H. C. F. Kooistra, E. Lara, N. Le Bescot, R. Logares, F. Mahé, R. Massana, M. Montresor, R. Morard, F. Not, J. Pawlowski, I. Probert, A.-L. Sauvadet, R. Siano, T. Stoeck, D. Vaulot, P. Zimmermann, and R. Christen. The Protist Ribosomal Reference database (PR2): a catalog of unicellular eukaryote small sub-unit rRNA sequences with curated taxonomy. *Nucleic Acids Research*, 41(D1):D597–D604, 2012.

[139] R. S. Gupta. Life's Third Domain (Archaea): An Established Fact or an Endangered Paradigm?: A New Proposal for Classification of Organisms Based on Protein Sequences and Cell Structure. *Theoretical Population Biology*, 54 (2):91–104, 1998.

[140] M. H. Haber. On Probability and Systematics: Possibility, Probability, and Phylogenetic Inference. *Systematic Biology*, 54(5):831–841, 2005.

[141] P. Halasz. Biological Classification, 2007. Online: `https://en.wikipedia.org/wiki/File:Biological_classification_L_Pengo_vflip.svg`. Accessed: 2018-08-03. The image was released into the public domain.

[142] B. Hall, B. Hallgrímsson, and M. W. Strickberger. *Strickberger's Evolution*. Jones & Bartlett Learning, 2008.

[143] G. Hamerly and C. Elkan. Learning the k in k-means. In S. Thrun, L. K. Saul, and P. B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 281–288. MIT Press, 2004.

[144] M. V. Han and C. M. Zmasek. phyloXML: XML for evolutionary biology and comparative genomics. *BMC Bioinformatics*, 10:356, 2009.

[145] M. Hartfield, C. L. Murall, and S. Alizon. Clinical applications of pathogen phylogenies. *Trends in Molecular Medicine*, 20(7):394–404, 2014.

[146] M. Hasegawa, H. Kishino, and T.-a. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution*, 22(2):160–174, 1985.

[147] J. M. Heather and B. Chain. The sequence of sequencers: The history of sequencing DNA. *Genomics*, 107(1):1–8, 2016.

[148] P. D. N. Hebert, A. Cywinska, S. L. Ball, and J. R. DeWaard. Biological Identifications Through DNA Barcodes. *Proceedings in Biological Sciences*, 270(1512):313–21, 2003.

[149] J. Hein. A heuristic method to reconstruct the history of sequences subject to recombination. *Journal of Molecular Evolution*, 36(4):396–405, 1993.

[150] A. P. Hendry, M. T. Kinnison, M. Heino, T. Day, T. B. Smith, G. Fitt, C. T. Bergstrom, J. Oakeshott, P. S. Jørgensen, M. P. Zalucki, G. Gilchrist, S. Southerton, A. Sih, S. Strauss, R. F. Denison, and S. P. Carroll. Evolutionary Principles and their Practical Application. *Evolutionary Applications*, 4 (2):159–83, 2011.

[151] D. G. Higgins and P. M. Sharp. CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, 73(1):237–44, 1988.

[152] D. Hillis and J. Wiens. Molecules versus morphology in systematics: conflicts, artifacts, and misconceptions. *Phylogenetic Analysis of Morphological Data*, pages 1–19, 2000.

[153] C. E. Hinchliff, S. A. Smith, J. F. Allman, J. G. Burleigh, R. Chaudhary, L. M. Coghill, K. A. Crandall, J. Deng, B. T. Drew, R. Gazis, K. Gude, D. S. Hibbett, L. A. Katz, H. D. Laughinghouse, E. J. McTavish, P. E. Midford, C. L. Owen, R. H. Ree, J. A. Rees, D. E. Soltis, T. Williams, and K. A. Cranston. Synthesis of phylogeny and taxonomy into a comprehensive tree of

life. *Proceedings of the National Academy of Sciences of the United States of America*, 112(41):12764–9, 2015.

[154] M. Hoff, S. Orf, B. Riehm, D. Darriba, and A. Stamatakis. Does the choice of nucleotide substitution models matter topologically? *BMC Bioinformatics*, 17:143, 2016.

[155] L. A. Hug, B. J. Baker, K. Anantharaman, C. T. Brown, A. J. Probst, C. J. Castelle, C. N. Butterfield, A. W. Hernsdorf, Y. Amano, K. Ise, Y. Suzuki, N. Dudek, D. A. Relman, K. M. Finstad, R. Amundson, B. C. Thomas, and J. F. Banfield. A new view of the tree of life. *Nature Microbiology*, 1(5):16048, 2016.

[156] P. Hugenholtz, B. M. Goebel, and N. R. Pace. Impact of Culture-Independent Studies on the Emerging Phylogenetic View of Bacterial Diversity. *Journal of Bacteriology*, 180(18):4765–4774, 1998.

[157] L. W. Hugerth and A. F. Andersson. Analysing Microbial Community Composition through Amplicon Sequencing: From Sampling to Hypothesis Testing. *Frontiers in Microbiology*, 8:1561, 2017.

[158] J. B. Hughes Martiny, B. J. Bohannan, J. H. Brown, R. K. Colwell, J. A. Fuhrman, J. L. Green, M. C. Horner-Devine, M. Kane, J. A. Krumins, C. R. Kuske, P. J. Morin, S. Naeem, L. Øvreås, A.-L. Reysenbach, V. H. Smith, and J. T. Staley. Microbial biogeography: putting microorganisms on the map. *Nature Reviews Microbiology*, 4(2):102–112, 2006.

[159] D. H. Huson and C. Scornavacca. A Survey of Combinatorial Methods for Phylogenetic Networks. *Genome Biology and Evolution*, 3:23–35, 2011.

[160] C. Huttenhower, D. Gevers, R. Knight, S. Abubucker, J. H. Badger, A. T. Chinwalla, H. H. Creasy, A. M. Earl, M. G. FitzGerald, R. S. Fulton, M. G. Giglio, K. Hallsworth-Pepin, E. A. Lobos, R. Madupu, V. Magrini, J. C. Martin, M. Mitreva, D. M. Muzny, E. J. Sodergren, J. Versalovic, A. M. Wollam, K. C. Worley, J. R. Wortman, S. K. Young, Q. Zeng, K. M. Aagaard, O. O. Abolude, E. Allen-Vercoe, E. J. Alm, L. Alvarado, G. L. Andersen, S. Anderson, E. Appelbaum, H. M. Arachchi, G. Armitage, C. A. Arze, T. Ayvaz, C. C. Baker, L. Begg, T. Belachew, V. Bhonagiri, M. Bihan, M. J. Blaser, T. Bloom, V. Bonazzi, J. Paul Brooks, G. A. Buck, C. J. Buhay, D. A. Busam, J. L. Campbell, S. R. Canon, B. L. Cantarel, P. S. G. Chain, I.-M. A. Chen, L. Chen, S. Chhibba, K. Chu, D. M. Ciulla, J. C. Clemente, S. W. Clifton, S. Conlan, J. Crabtree, M. A. Cutting, N. J. Davidovics, C. C. Davis, T. Z. DeSantis, C. Deal, K. D. Delehaunty, F. E. Dewhirst, E. Deych, Y. Ding, D. J. Dooling, S. P. Dugan, W. Michael Dunne, A. Scott Durkin, R. C. Edgar, R. L. Erlich, C. N. Farmer, R. M. Farrell, K. Faust, M. Feldgarden, V. M. Felix, S. Fisher, A. A. Fodor, L. J. Forney, L. Foster, V. Di Francesco, J. Friedman, D. C. Friedrich, C. C. Fronick, L. L. Fulton, H. Gao, N. Garcia, G. Giannoukos, C. Giblin, M. Y. Giovanni, J. M. Goldberg, J. Goll, A. Gonzalez, A. Griggs,

S. Gujja, S. Kinder Haake, B. J. Haas, H. A. Hamilton, E. L. Harris, T. A. Hepburn, B. Herter, D. E. Hoffmann, M. E. Holder, C. Howarth, K. H. Huang, S. M. Huse, J. Izard, J. K. Jansson, H. Jiang, C. Jordan, V. Joshi, J. A. Katancik, W. A. Keitel, S. T. Kelley, C. Kells, N. B. King, D. Knights, H. H. Kong, O. Koren, S. Koren, K. C. Kota, C. L. Kovar, N. C. Kyrpides, P. S. La Rosa, S. L. Lee, K. P. Lemon, N. Lennon, C. M. Lewis, L. Lewis, R. E. Ley, K. Li, K. Liolios, B. Liu, Y. Liu, C.-C. Lo, C. A. Lozupone, R. Dwayne Lunsford, T. Madden, A. A. Mahurkar, P. J. Mannon, E. R. Mardis, V. M. Markowitz, K. Mavromatis, J. M. McCorrison, D. McDonald, J. McEwen, A. L. McGuire, P. McInnes, T. Mehta, K. A. Mihindukulasuriya, J. R. Miller, P. J. Minx, I. Newsham, C. Nusbaum, M. O'Laughlin, J. Orvis, I. Pagani, K. Palaniappan, S. M. Patel, M. Pearson, J. Peterson, M. Podar, C. Pohl, K. S. Pollard, M. Pop, M. E. Priest, L. M. Proctor, X. Qin, J. Raes, J. Ravel, J. G. Reid, M. Rho, R. Rhodes, K. P. Riehle, M. C. Rivera, B. Rodriguez-Mueller, Y.-H. Rogers, M. C. Ross, C. Russ, R. K. Sanka, P. Sankar, J. Fah Sathirapongsasuti, J. A. Schloss, P. D. Schloss, T. M. Schmidt, M. Scholz, L. Schriml, A. M. Schubert, N. Segata, J. A. Segre, W. D. Shannon, R. R. Sharp, T. J. Sharpton, N. Shenoy, N. U. Sheth, G. A. Simone, I. Singh, C. S. Smillie, J. D. Sobel, D. D. Sommer, P. Spicer, G. G. Sutton, S. M. Sykes, D. G. Tabbaa, M. Thiagarajan, C. M. Tomlinson, M. Torralba, T. J. Treangen, R. M. Truty, T. A. Vishnivetskaya, J. Walker, L. Wang, Z. Wang, D. V. Ward, W. Warren, M. A. Watson, C. Wellington, K. A. Wetterstrand, J. R. White, K. Wilczek-Boney, Y. Wu, K. M. Wylie, T. Wylie, C. Yandava, L. Ye, Y. Ye, S. Yooseph, B. P. Youmans, L. Zhang, Y. Zhou, Y. Zhu, L. Zoloth, J. D. Zucker, B. W. Birren, R. A. Gibbs, S. K. Highlander, B. A. Methé, K. E. Nelson, J. F. Petrosino, G. M. Weinstock, R. K. Wilson, and O. White. Structure, function and diversity of the healthy human microbiome. *Nature*, 486(7402):207–214, 2012.

[161] IUPAC-IUB Commission on Biochemical Nomenclature (CBN). Abbreviations and symbols for nucleic acids, polynucleotides, and their constituents. *Biochemistry*, 9(20):4022–4027, 1970.

[162] P. Jaccard. Etude de la distribution florale dans une portion des Alpes et du Jura. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[163] D. A. Jackson. Compositional data in community ecology: The paradigm or peril of proportions? *Ecology*, 78(3):929–940, 1997.

[164] M. Jamy, R. Foster, P. Barbera, L. Czech, A. M. Kozlov, A. M. Stamatakis, D. Bass, and F. Burki. Long metabarcoding of the eukaryotic rDNA operon to phylogenetically and taxonomically resolve environmental diversity. *bioRxiv*, 2019.

[165] J. M. Janda and S. L. Abbott. 16S rRNA Gene Sequencing for Bacterial Identification in the Diagnostic Laboratory: Pluses, Perils, and Pitfalls. *Journal of Clinical Microbiology*, 45(9):2761–2764, 2007.

[166] S. Janssen, D. McDonald, A. Gonzalez, J. A. Navas-Molina, L. Jiang, Z. Z. Xu, K. Winker, D. M. 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, 2018.

[167] E. D. Jarvis, S. Mirarab, A. J. Aberer, B. Li, P. Houde, C. Li, S. Y. W. 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-Núñez, 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. Jønsson, W. Johnson, K.-P. Koepfli, S. O\textquoterightBrien, D. Haussler, O. A. Ryder, C. Rahbek, E. Willerslev, G. R. Graves, T. C. Glenn, J. McCormack, D. Burt, H. Ellegren, P. Alström, 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.

[168] I. Jolliffe. *Principal Component Analysis*. Springer, New York, 2nd edition, 2002.

[169] T. H. Jukes and C. R. Cantor. Evolution of Protein Molecules. *Mammalian Protein Metabolism*, 3(21):132, 1969.

[170] W. Just. Computational Complexity of Multiple Sequence Alignment with SP-Score. *Journal of Computational Biology*, 8(6):615–623, 2001.

[171] T. Kanagawa. Bias and Artifacts in Multitemplate Polymerase Chain Reactions (PCR). *Journal of Bioscience and Bioengineering*, 96(4):317–323, 2003.

[172] T. Kanungo, D. M. Mount, N. S. Netanyahu, A. Y. Wu, C. D. Piatko, R. Silverman, and A. Y. Wu. A Local Search Approximation Algorithm for k-Means Clustering. *Computational Geometry*, 28(2-3):89–112, 2003.

[173] P. Kapli, S. Lutteropp, J. Zhang, K. Kobert, P. Pavlidis, A. Stamatakis, and T. Flouri. Multi-rate Poisson tree processes for single-locus species delimitation

under maximum likelihood and Markov chain Monte Carlo. *Bioinformatics*, 33(11):1630–1638, 2017.

[174] E. Karsenti, S. G. Acinas, P. Bork, C. Bowler, C. de Vargas, J. Raes, M. Sullivan, D. Arendt, F. Benzoni, J. M. Claverie, M. Follows, G. Gorsky, P. Hingamp, D. Iudicone, O. Jaillon, S. Kandels-Lewis, U. Krzic, F. Not, H. Ogata, S. Pesant, E. G. Reynaud, C. Sardet, M. E. Sieracki, S. Speich, D. Velayoudon, J. Weissenbach, and P. Wincker. A holistic approach to marine Eco-systems biology. *PLoS Biology*, 9(10):7–11, 2011.

[175] K. Katoh and D. M. Standley. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. *Molecular Biology and Evolution*, 30(4):772–780, 2013.

[176] K. Katoh, K. Misawa, K. Kuma, and T. Miyata. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research*, 30(14):3059–3066, 2002.

[177] K. Katoh, K.-i. Kuma, H. Toh, and T. Miyata. MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic Acids Research*, 33 (2):511–518, 2005.

[178] D. R. Kelley and S. L. Salzberg. Clustering metagenomic sequences with interpolated Markov models. *BMC Bioinformatics*, 11(1):544, 2010.

[179] O.-S. Kim, Y.-J. Cho, K. Lee, S.-H. Yoon, M. Kim, H. Na, S.-C. Park, Y. S. Jeon, J.-H. Lee, H. Yi, S. Won, and J. Chun. Introducing EzTaxon-e: a prokaryotic 16S rRNA gene sequence database with phylotypes that represent uncultured species. *International Journal of Systematic and Evolutionary Microbiology*, 62(3):716–721, 2012.

[180] M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16(2):111–120, 1980.

[181] H. Kishino and M. Hasegawa. Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in hominoidea. *Journal of Molecular Evolution*, 29(2):170–179, 1989.

[182] H. Kishino, T. Miyata, and M. Hasegawa. Maximum likelihood inference of protein phylogeny and the origin of chloroplasts. *Journal of Molecular Evolution*, 31(2):151–160, 1990.

[183] D. Kocyła. DNA structure and bases, 2006. Online: `https://commons.wikimedia.org/wiki/File:DNA_structure_and_bases.svg`. Accessed: 2018-08-04. The image is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license, see `https://creativecommons.org/licenses/by-sa/3.0/deed.en`.

[184] E. V. Koonin. Orthologs, Paralogs, and Evolutionary Genomics. *Annual Review of Genetics*, 39(1):309–338, 2005.

[185] L. B. Koski and G. B. Golding. The Closest BLAST Hit is Often not the Nearest Neighbor. *Journal of Molecular Evolution*, 52(6):540–2, 2001.

[186] D. Koslicki. CAMIARKQuikr: v1.0.0, 2018. Online: `http://doi.org/10.5281/zenodo.1730572`. Accessed: 2018-11-30.

[187] D. Koslicki and F. Boulund. MetaPalette v1.0.0, 2018. Online: `http://doi.org/10.5281/zenodo.1730624`. Accessed: 2018-11-30.

[188] D. Koslicki and D. Falush. MetaPalette: a k-mer Painting Approach for Metagenomic Taxonomic Profiling and Quantification of Novel Strain Variation. *mSystems*, 1(3):e00020–16, 2016.

[189] D. Koslicki, S. Foucart, and G. Rosen. Quikr: a method for rapid reconstruction of bacterial communities via compressive sensing. *Bioinformatics*, 29(17): 2096–2102, 2013.

[190] D. Koslicki, S. Chatterjee, D. Shahrivar, A. W. Walker, S. C. Francis, L. J. Fraser, M. Vehkaperä, Y. Lan, and J. Corander. ARK: Aggregation of Reads by K-Means for Estimation of Bacterial Community Composition. *PLOS ONE*, 10(10):e0140644, 2015.

[191] A. M. Kozlov, J. Zhang, P. Yilmaz, F. O. Glöckner, and A. Stamatakis. Phylogeny-aware identification and correction of taxonomically mislabeled sequences. *Nucleic Acids Research*, 44(11):5022–5033, 2016.

[192] A. M. Kozlov, D. Darriba, T. Flouri, B. Morel, and A. Stamatakis. RAxML-NG: A fast, scalable, and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics*, 2019.

[193] O. Kozlov. *Models, Optimizations, and Tools for Large-Scale Phylogenetic Inference, Handling Sequence Uncertainty, and Taxonomic Validation.* PhD thesis, Karlsruher Institute für Technologie (KIT), 2018.

[194] W. J. Kress and D. L. Erickson. DNA Barcodes: Genes, Genomics, and Bioinformatics. *Proceedings of the National Academy of Sciences of the United States of America*, 105(8):2761–2, 2008.

[195] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3): 231–240, 2011.

[196] W. J. Krzanowski and F. Marriott. *Multivariate Analysis.* Wiley, 1994.

[197] M. K. Kuhner and J. Felsenstein. A Simulation Comparison of Phylogeny Algorithms under Equal and Unequal Evolutionary Rates. *Molecular Biology and Evolution*, 11(3):459–468, 1994.

[198] V. Kunin, A. Engelbrektson, H. Ochman, and P. Hugenholtz. Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates. *Environmental Microbiology*, 12(1):118–123, 2010.

[199] A. Kupczok, H. A. Schmidt, and A. von Haeseler. Accuracy of phylogeny reconstruction methods combining overlapping gene data sets. *Algorithms for Molecular Biology*, 5:37, 2010.

[200] C. C. Laczny, N. Pinel, N. Vlassis, and P. Wilmes. Alignment-free visualization of metagenomic data by nonlinear dimension reduction. *Scientific Reports*, 4: 4516, 2014.

[201] S. Q. Le and O. Gascuel. An Improved General Amino Acid Replacement Matrix. *Molecular Biology and Evolution*, 25(7):1307–1320, 2008.

[202] P. Legendre and L. F. J. Legendre. *Numerical Ecology*. Developments in Environmental Modelling. Elsevier Science, 1998.

[203] G. Lentendu, F. Mahé, D. Bass, S. Rueckert, T. Stoeck, and M. Dunthorn. Consistent patterns of high alpha and low beta diversity in tropical parasitic and free-living protists. *Molecular Ecology*, 27(13):2846–2857, 2018.

[204] A. M. Leroi. *The Lagoon: How Aristotle Invented Science*. Bloomsbury Circus, 1st edition, 2014.

[205] 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.

[206] E. Levina and P. Bickel. The earth mover's distance is the Mallows distance: some insights from statistics. *Eighth IEEE International Conference on Computer Vision*, pages 251–256, 2001.

[207] C. Li and J. Wang. Relative entropy of DNA and its application. *Physica A: Statistical Mechanics and its Applications*, 347:465–471, 2005.

[208] H. Li. Microbiome, Metagenomics, and High-Dimensional Compositional Data Analysis. *Annual Review of Statistics and Its Application*, 2(1):73–94, 2015.

[209] G. Lima-Mendez, K. Faust, N. Henry, J. Decelle, S. Colin, F. Carcillo, S. Chaffron, J. C. Ignacio-Espinosa, S. Roux, F. Vincent, L. Bittner, Y. Darzi, J. Wang, S. Audic, L. Berline, G. Bontempi, A. M. Cabello, L. Coppola, F. M. Cornejo-Castillo, F. D'Ovidio, L. De Meester, I. Ferrera, M.-J. Garet-Delmas, L. Guidi, E. Lara, S. Pesant, M. Royo-Llonch, G. Salazar, P. Sánchez, M. Sebastian, C. Souffreau, C. Dimier, M. Picheral, S. Searson, S. Kandels-Lewis, T. O. Tara Oceans coordinators, G. Gorsky, F. Not, H. Ogata, S. Speich, L. Stemmann, J. Weissenbach, P. Wincker, S. G. Acinas, S. Sunagawa, P. Bork, M. B. Sullivan, E. Karsenti, C. Bowler, C. de Vargas, and J. Raes. Determinants of community structure in the global plankton interactome. *Science*, 348(6237):1262073, 2015.

[210] B. Linard, K. Swenson, and F. Pardi. Rapid alignment-free phylogenetic identification of metagenomic sequences. *Bioinformatics*, 2019.

[211] S. Lindgreen, K. L. Adair, and P. P. Gardner. An evaluation of the accuracy and speed of metagenome analysis tools. *Scientific Reports*, 6(1):19233, 2016.

[212] C. Linnaeus. *Systema Naturae*. Haak, Leiden, 1735.

[213] C. Linnaeus. *Species Plantarum*. Laurentius Salvius, Stockholm, 1753.

[214] 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):S4, 2011.

[215] 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.

[216] 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, 2012.

[217] S. P. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[218] R. Logares, T. H. Haverkamp, S. Kumar, A. Lanzén, A. J. Nederbragt, C. Quince, and H. Kauserud. Environmental microbiology through the lens of high-throughput DNA sequencing: Synopsis of current platforms and bioinformatics approaches. *Journal of Microbiological Methods*, 91(1):106–113, 2012.

[219] R. Logares, S. Sunagawa, G. Salazar, F. M. Cornejo-Castillo, I. Ferrera, H. Sarmento, P. Hingamp, H. Ogata, C. de Vargas, G. Lima-Mendez, J. Raes, J. Poulain, O. Jaillon, P. Wincker, S. Kandels-Lewis, E. Karsenti, P. Bork, and S. G. Acinas. Metagenomic 16S rDNA Illumina tags are a powerful alternative to amplicon sequencing to explore diversity and structure of microbial communities. *Environmental Microbiology*, 16(9):2659–2671, 2014.

[220] D. Lovell, V. Pawlowsky-Glahn, J. J. Egozcue, S. Marguerat, and J. Bähler. Proportionality: A Valid Alternative to Correlation for Relative Data. *PLOS Computational Biology*, 11(3):e1004075, 2015.

[221] C. Lozupone and R. Knight. UniFrac: a New Phylogenetic Method for Comparing Microbial Communities. *Applied and Environmental Microbiology*, 71 (12):8228–8235, 2005.

[222] C. A. Lozupone and R. Knight. Global patterns in bacterial diversity. *Proceedings of the National Academy of Sciences*, 104(27):11436–11440, 2007.

[223] C. A. Lozupone, M. Hamady, S. T. Kelley, and R. Knight. Quantitative and Qualitative β Diversity Measures Lead to Different Insights into Factors That Structure Microbial Communities. *Applied and Environmental Microbiology*, 73(5):1576–1585, 2007.

[224] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1(233):281–297, 1967.

[225] D. R. Maddison, D. L. Swofford, and W. P. Maddison. NEXUS: an extensible file format for systematic information. *Systematic biology*, 46(4):590–621, 1997.

[226] W. P. Maddison and J. J. Wiens. Gene Trees in Species Trees. *Systematic Biology*, 46(3):523–536, 1997.

[227] F. Mahé. Fred's metabarcoding pipeline, 2016. Online: `https://github.com/frederic-mahe/swarm/wiki/Fred's-metabarcoding-pipeline`. Accessed: 2018-01-15.

[228] F. Mahé, T. Rognes, C. Quince, C. de Vargas, and M. Dunthorn. Swarm: Robust and fast clustering method for amplicon-based studies. *PeerJ*, 2:1–12, 2014.

[229] F. Mahé, T. Rognes, C. Quince, C. De Vargas, and M. Dunthorn. Swarm v2: Highly-scalable and high-resolution amplicon clustering. *PeerJ*, 2015.

[230] F. Mahé, C. de Vargas, D. Bass, L. Czech, A. Stamatakis, E. Lara, D. Singer, J. Mayor, J. Bunge, S. Sernaker, T. Siemensmeyer, I. Trautmann, S. Romac, C. Berney, A. Kozlov, E. A. D. Mitchell, C. V. W. Seppey, E. Egge, G. Lentendu, R. Wirth, G. Trueba, and M. Dunthorn. Parasites dominate hyperdiverse soil protist communities in Neotropical rainforests. *Nature Ecology & Evolution*, 1(4):0091, 2017.

[231] C. L. Mallows. A Note on Asymptotic Joint Normality. *Annals of Mathematical Statistics*, 43(2):508–515, 1972.

[232] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov. Applications of Deep Learning in Biomedicine. *Molecular Pharmaceutics*, 13(5):1445–1454, 2016.

[233] K. V. Mardia. Some Properties of Classical Multi-Dimesional Scaling. *Communications in Statistics-Theory and Methods*, 7(13):1233–1241, 1978.

[234] E. R. Mardis. Next-Generation Sequencing Platforms. *Annual Review of Analytical Chemistry*, 6(1):287–303, 2013.

[235] J. Mariette and N. Villa-Vialaneix. Unsupervised multiple kernel learning for heterogeneous data integration. *Bioinformatics*, 34(6):1009–1015, 2018.

[236] M. Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, 17(1):10, 2011.

[237] F. J. Massey Jr. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*, 46(253):68–78, 1951.

[238] F. A. Matsen. Phylogenetics and the Human Microbiome. *Systematic Biology*, 64(1):e26–e41, 2015.

[239] F. A. Matsen and S. N. Evans. Edge Principal Components and Squash Clustering: Using the Special Structure of Phylogenetic Placement Data for Sample Comparison. *PLOS ONE*, 8(3):1–17, 2011.

[240] F. A. Matsen and S. N. Evans. Edge Principal Components and Squash Clustering: Using the Special Structure of Phylogenetic Placement Data for Sample Comparison. *arXiv*, 2011.

[241] F. A. Matsen, R. B. 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.

[242] F. A. Matsen, R. B. Kodner, and E. V. Armbrust. pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *arXiv*, 2010.

[243] F. A. Matsen, N. G. Hoffman, A. Gallagher, and A. Stamatakis. A format for phylogenetic placements. *PLoS ONE*, 7(2):1–4, 2012.

[244] F. A. Matsen, A. Gallagher, and C. O. McCoy. Minimizing the average distance to a closest leaf in a phylogenetic tree. *Systematic Biology*, 62(6):824–836, 2013.

[245] K. O. May. A set of independent necessary and sufficient conditions for simple majority decision. *Econometrica: Journal of the Econometric Society*, pages 680–684, 1952.

[246] E. Mayr. Two empires or three? *Proceedings of the National Academy of Sciences of the United States of America*, 95(17):9720–3, 1998.

[247] E. Mayr and W. J. Bock. Classifications and other ordering systems. *Journal of Zoological Systematics and Evolutionary Research*, 40(4):169–194, 2002.

[248] P. McCullagh and J. A. Nelder. *Generalized Linear Models*, volume 37. CRC press, 1989.

[249] P. J. McMurdie and S. Holmes. Waste Not, Want Not: Why Rarefying Microbiome Data Is Inadmissible. *PLoS Computational Biology*, 10(4):e1003531, 2014.

[250] B. A. Methé, K. E. Nelson, M. Pop, H. H. Creasy, M. G. Giglio, C. Huttenhower, D. Gevers, J. F. Petrosino, S. Abubucker, H. Jonathan, A. T. Chinwalla, A. M. Earl, M. G. Fitzgerald, R. S. Fulton, K. Hallsworth-Pepin, E. A. Lobos, R. Madupu, V. Magrini, J. C. Martin, M. Mitreva, D. M. Muzny, E. J. Sodergren, A. M. Wollam, K. C. Worley, J. R. Wortman, S. K. Young, Q. Zeng,

K. M. Aagaard, O. O. Abolude, E. Allen-vercoe, J. Eric, L. Alvarado, G. L. Andersen, S. Anderson, E. Appelbaum, H. M. Arachchi, G. Armitage, C. A. Arze, T. Ayvaz, C. C. Baker, L. Begg, T. Belachew, V. Bhonagiri, M. Bihan, M. J. Blaser, T. Bloom, J. V. Bonazzi, P. Brooks, G. A. Buck, J. Christian, D. A. Busam, J. L. Campbell, S. R. Canon, B. L. Cantarel, P. S. Chain, I.-M. A. Chen, L. Chen, S. Chhibba, K. Chu, M. Dawn, J. C. Clemente, S. W. Clifton, S. Conlan, J. Crabtree, A. Cutting, N. J. Davidovics, C. C. Davis, T. Z. Desantis, K. D. Delehaunty, F. E. Dewhirst, E. Deych, Y. Ding, J. H. Badger, A. T. Chinwalla, A. M. Earl, M. G. Fitzgerald, R. S. Fulton, K. Hallsworth-Pepin, E. A. Lobos, R. Madupu, V. Magrini, J. C. Martin, M. Mitreva, D. M. Muzny, E. J. Sodergren, J. Versalovic, A. M. Wollam, K. C. Worley, J. R. Wortman, S. K. Young, Q. Zeng, K. M. Aagaard, O. O. Abolude, E. Allen-vercoe, E. J. Alm, L. Alvarado, G. L. Andersen, S. Anderson, E. Appelbaum, H. M. Arachchi, G. Armitage, C. A. Arze, T. Ayvaz, C. C. Baker, L. Begg, T. Belachew, V. Bhonagiri, M. Bihan, M. J. Blaser, T. Bloom, V. R. Bonazzi, P. Brooks, G. A. Buck, C. J. Buhay, D. A. Busam, J. L. Campbell, S. R. Canon, B. L. Cantarel, P. S. Chain, I.-M. A. Chen, L. Chen, S. Chhibba, K. Chu, D. M. Ciulla, J. C. Clemente, S. W. Clifton, S. Conlan, J. Crabtree, M. A. Cutting, N. J. Davidovics, C. C. Davis, T. Z. Desantis, C. Deal, K. D. Delehaunty, F. E. Dewhirst, E. Deych, Y. Ding, D. J. Dooling, S. P. Dugan, W. Michael Dunne, A. Scott Durkin, R. C. Edgar, R. L. Erlich, C. N. Farmer, R. M. Farrell, K. Faust, M. Feldgarden, V. M. Felix, S. Fisher, A. A. Fodor, L. Forney, L. Foster, V. Di Francesco, J. Friedman, D. C. Friedrich, C. C. Fronick, L. L. Fulton, H. Gao, N. Garcia, G. Giannoukos, C. Giblin, M. Y. Giovanni, J. M. Goldberg, J. Goll, A. Gonzalez, A. Griggs, S. Gujja, B. J. Haas, H. A. Hamilton, E. L. Harris, T. A. Hepburn, B. Herter, D. E. Hoffmann, M. E. Holder, C. Howarth, K. H. Huang, S. M. Huse, J. Izard, J. K. Jansson, H. Jiang, C. Jordan, V. Joshi, J. A. Katancik, W. A. Keitel, S. T. Kelley, C. Kells, S. Kinder-Haake, N. B. King, R. Knight, D. Knights, H. H. Kong, O. Koren, S. Koren, K. C. Kota, C. L. Kovar, N. C. Kyrpides, P. S. La Rosa, S. L. Lee, K. P. Lemon, N. Lennon, C. M. Lewis, L. Lewis, R. E. Ley, K. Li, K. Liolios, B. Liu, Y. Liu, C.-C. Lo, C. A. Lozupone, R. Dwayne Lunsford, T. Madden, A. A. Mahurkar, P. J. Mannon, E. R. Mardis, V. M. Markowitz, K. Mavrommatis, J. M. McCorrison, D. McDonald, J. McEwen, A. L. McGuire, P. McInnes, T. Mehta, K. A. Mihindukulasuriya, J. R. Miller, P. J. Minx, I. Newsham, C. Nusbaum, M. O'Laughlin, J. Orvis, I. Pagani, K. Palaniappan, S. M. Patel, M. Pearson, J. Peterson, M. Podar, C. Pohl, K. S. Pollard, M. E. Priest, L. M. Proctor, X. Qin, J. Raes, J. Ravel, J. G. Reid, M. Rho, R. Rhodes, K. P. Riehle, M. C. Rivera, B. Rodriguez-Mueller, Y.-H. Rogers, M. C. Ross, C. Russ, R. K. Sanka, P. Sankar, J. Fah Sathirapongsasuti, J. A. Schloss, P. D. Schloss, T. M. Schmidt, M. Scholz, L. Schriml, A. M. Schubert, N. Segata, J. A. Segre, W. D. Shannon, R. R. Sharp, T. J. Sharpton, N. Shenoy, N. U. Sheth, G. A. Simone, I. Singh, C. S. Smillie, J. D. Sobel, D. D. Sommer, P. Spicer, G. G. Sutton, S. M. Sykes, D. G. Tabbaa, M. Thiagarajan, C. M. Tomlinson, M. Torralba, T. J. Treangen, R. M. Truty, T. A.

Vishnivetskaya, J. Walker, L. Wang, Z. Wang, D. V. Ward, W. Warren, M. A. Watson, C. Wellington, K. A. Wetterstrand, J. R. White, K. Wilczek-Boney, Y. Qing Wu, K. M. Wylie, T. Wylie, C. Yandava, L. Ye, Y. Ye, S. Yooseph, B. P. Youmans, L. Zhang, Y. Zhou, Y. Zhu, L. Zoloth, J. D. Zucker, B. W. Birren, R. A. Gibbs, S. K. Highlander, G. M. Weinstock, R. K. Wilson, and O. White. A framework for human microbiome research. *Nature*, 486(7402): 215–221, 2012.

[251] M. L. Metzker. Sequencing Technologies—The Next Generation. *Nature Reviews Genetics*, 11(1):31–46, 2010.

[252] A. Meyer, C. Todt, N. T. Mikkelsen, and B. Lieb. Fast evolving 18S rRNA sequences from Solenogastres (Mollusca) resist standard PCR amplification and give new insights into mollusk substitution rate heterogeneity. *BMC Evolutionary Biology 2010 10:1*, 10(1):70, 2010.

[253] F. Meyer, A. Bremges, P. Belmann, S. Janssen, A. C. McHardy, and D. Koslicki. Assessing taxonomic metagenome profilers with OPAL. *Genome Biology*, 20(1):51, 2019.

[254] C. D. Michener and R. R. Sokal. A quantitative approach to a problem in classification. *Evolution*, 11(2):130–162, 1957.

[255] S. Mignard and J. P. Flandrois. 16S rRNA sequencing in routine bacterial identification: a 30-month experiment. *Journal of Microbiological Methods*, 67 (3):574–581, 2006.

[256] M. Mignardi and M. Nilsson. Fourth-generation sequencing in the cell and the clinic. *Genome Medicine*, 6(4):31, 2014.

[257] S. Min, B. Lee, and S. Yoon. Deep learning in bioinformatics. *Briefings in bioinformatics*, 18(5):851–869, 2017.

[258] D. P. Mindell. The Tree of Life: Metaphor, Model, and Heuristic Device. *Systematic Biology*, 62(3):479–489, 2013.

[259] B. Minh, S. Klaere, and A. Haeseler. Phylogenetic Diversity within Seconds. *Systematic Biology*, 55(5):769–773, 2006.

[260] S. Mirarab, N. Nguyen, and T. Warnow. SEPP: SATé-Enabled Phylogenetic Placement. In *Pacific Symposium on Biocomputing*, pages 247–258. World Scientific, 2012.

[261] 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): 763–7, 2014.

[262] A. Monier, J.-M. Claverie, and H. Ogata. Taxonomic distribution of large DNA viruses in the sea. *Genome Biology*, 9(7):R106, 2008.

[263] G. E. Moore. Cramming More Components onto Integrated Circuits. *Electronics*, pages 114–117, 1965.

[264] D. Moreira and H. Philippe. Molecular phylogeny: pitfalls and progress. *International Microbiology*, 3(1):9–16, 2000.

[265] B. Morel, A. M. Kozlov, and A. Stamatakis. ParGenes: a tool for massively parallel model selection and phylogenetic tree inference on thousands of genes. *Bioinformatics*, 35(10):1771–1773, 2019.

[266] J. L. Morgan, A. E. Darling, and J. A. Eisen. Metagenomic sequencing of an in vitro-simulated microbial community. *PLoS ONE*, 5(4):1–10, 2010.

[267] J. T. Morton, J. Sanders, R. A. Quinn, D. McDonald, A. Gonzalez, Y. Vázquez-Baeza, J. A. Navas-Molina, S. J. Song, J. L. Metcalf, E. R. Hyde, M. Lladser, P. C. Dorrestein, and R. Knight. Balance Trees Reveal Microbial Niche Differentiation. *mSystems*, 2(1), 2017.

[268] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[269] M. Nei and W. H. Li. Mathematical model for studying genetic variation in terms of restriction endonucleases. *Proceedings of the National Academy of Sciences of the United States of America*, 76(10):5269–5273, 1979.

[270] J. A. Nelder and R. W. M. Wedderburn. Generalized Linear Models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):370–384, 1972.

[271] L.-T. T. Nguyen, H. A. Schmidt, A. Von Haeseler, and B. Q. Minh. IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, 32(1):268–74, 2015.

[272] N. P. Nguyen, S. Mirarab, B. Liu, M. Pop, and T. Warnow. TIPP: Taxonomic identification and phylogenetic profiling. *Bioinformatics*, 30(24):3548–3555, 2014.

[273] T. P. Niedringhaus, D. Milanova, M. B. Kerby, M. P. Snyder, and A. E. Barron. Landscape of Next-Generation Sequencing Technologies. *Analytical Chemistry*, 83(12):4327–4341, 2011.

[274] R. P. Nugent, M. A. Krohn, and S. L. Hillier. Reliability of diagnosing bacterial vaginosis is improved by a standardized method of gram stain interpretation. *Journal of Clinical Microbiology*, 29(2):297–301, 1991.

[275] H. Ochman, J. G. Lawrence, and E. A. Groisman. Lateral gene transfer and the nature of bacterial innovation. *Nature*, 405(6784):299–304, 2000.

[276] B. D. Ondov, N. H. Bergman, and A. M. Phillippy. Interactive metagenomic visualization in a Web browser. *BMC Bioinformatics*, 12(1):385, 2011.

[277] A. Oulas, C. Pavloudi, P. Polymenakou, G. A. Pavlopoulos, N. Papanikolaou, G. Kotoulas, C. Arvanitidis, and I. Iliopoulos. Metagenomics: Tools and Insights for Analyzing Next-Generation Sequencing Data Derived from Biodiversity Studies. *Bioinformatics and Biology Insights*, 9:75–88, 2015.

[278] N. R. Pace. A molecular view of microbial diversity and the biosphere. *Science*, 276(5313):734–740, 1997.

[279] F. Pardi and N. Goldman. Species Choice for Comparative Genomics: Being Greedy Works. *PLOS Genetics*, 1(6):1, 2005.

[280] C. S. Pareek, R. Smoczynski, and A. Tretyn. Sequencing technologies and genome sequencing. *Journal of Applied Genetics*, 52(4):413–435, 2011.

[281] D. H. Parks, M. Chuvochina, D. W. Waite, C. Rinke, A. Skarshewski, P.-A. Chaumeil, and P. Hugenholtz. A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. *Nature Biotechnology*, 36(10):996, 2018.

[282] E. Pasolli, D. T. Truong, F. Malik, L. Waldron, and N. Segata. Machine Learning Meta-analysis of Large Metagenomic Datasets: Tools and Biological Insights. *PLOS Computational Biology*, 12(7):e1004977, 2016.

[283] V. Pawlowsky-Glahn and A. Buccianti. *Compositional Data Analysis: Theory and Applications*. John Wiley & Sons, 2011.

[284] V. Pawlowsky-Glahn, J. J. Egozcue, and R. Tolosana-Delgado. *Modeling and Analysis of Compositional Data*. John Wiley & Sons, Chichester, UK, 2015.

[285] M. A. Peabody, T. Van Rossum, R. Lo, and F. S. L. Brinkman. Evaluation of shotgun metagenomics sequence classification methods using in silico and in vitro simulated communities. *BMC Bioinformatics*, 16:363, 2015.

[286] K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[287] W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448, 1988.

[288] D. Pelleg and A. W. Moore. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *ICML*, volume 1, pages 727–734, 2000.

[289] E. Pettersson, J. Lundeberg, and A. Ahmadian. Generations of Sequencing Technologies. *Genomics*, 93(2):105–111, 2009.

[290] C. A. Petti. Detection and identification of microorganisms by gene amplification and sequencing. *Clinical Infectious Diseases*, 44(8):1108–1114, 2007.

[291] V. O. Polyanovsky, M. A. Roytberg, and V. G. Tumanyan. Comparative analysis of the quality of a global algorithm and a local algorithm for alignment of two sequences. *Algorithms for Molecular Biology*, 6(1):25, 2011.

[292] M. Potapova. Patterns of Diatom Distribution In Relation to Salinity. In J. Kociolek and J. Seckbach, editors, *The Diatom World*, pages 313–332. Springer, 2011.

[293] M. N. Price, P. S. Dehal, and A. P. Arkin. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLoS ONE*, 5(3):e9490, 2010.

[294] C. Quast, E. Pruesse, P. Yilmaz, J. Gerken, T. Schweer, P. Yarza, J. Peplies, and F. O. Glockner. The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Research*, 41(D1): D590–D596, 2013.

[295] T. P. Quinn, I. Erb, M. F. Richardson, and T. M. Crowley. Understanding sequencing data as compositions: an outlook and review. *Bioinformatics*, 34 (16):2870–2878, 2018.

[296] S. T. Rachev. The Monge-Kantorovich Mass Transference Problem and its Stochastic Applications. *Theory of Probability and its Applications*, 29(4): 647–676, 1985.

[297] S. T. Rachev. *Probability Metrics and the Stability of Stochastic Models.* Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. John Wiley & Sons Ltd, Chichester, 1991.

[298] S. T. Rachev and L. Rüschendorf. *Mass Transportation Problems.* Volume 1: Theory. Springer-Verlag, New York, 1 edition, 1998.

[299] O. Ratmann, E. B. Hodcroft, M. Pickles, A. Cori, M. Hall, S. Lycett, C. Colijn, B. Dearlove, X. Didelot, S. Frost, A. M. M. Hossain, J. B. Joy, M. Kendall, D. Kühnert, G. E. Leventhal, R. Liang, G. Plazzotta, A. F. Poon, D. A. Rasmussen, T. Stadler, E. Volz, C. Weis, A. J. Leigh Brown, C. Fraser, and PANGEA-HIV Consortium. Phylogenetic Tools for Generalized HIV-1 Epidemics: Findings from the PANGEA-HIV Methods Comparison. *Molecular Biology and Evolution*, 34(1):185–203, 2017.

[300] D. Reiman, A. Metwally, and Y. Dai. Using convolutional neural networks to explore the microbiome. In *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4269–4272. IEEE, 2017.

[301] D. Reiman, A. A. Metwally, and Y. Dai. PopPhy-CNN: A Phylogenetic Tree Embedded Architecture for Convolution Neural Networks for Metagenomic Data. *bioRxiv*, page 257931, 2018.

[302] R. Ren, Y. Sun, Y. Zhao, D. Geiser, H. Ma, and X. Zhou. Phylogenetic Resolution of Deep Eukaryotic and Fungal Relationships Using Highly Conserved Low-Copy Nuclear Genes. *Genome Biology and Evolution*, 8(9):2683–701, 2016.

[303] J. A. Reuter, D. V. Spacek, and M. P. Snyder. High-Throughput Sequencing Technologies. *Molecular Cell*, 58(4):586–97, 2015.

[304] D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1):131–147, 1981.

[305] K. M. Robinson, K. B. Sieber, and J. C. Dunning Hotopp. A Review of Bacteria-Animal Lateral Gene Transfer May Inform Our Understanding of Diseases like Cancer. *PLoS Genetics*, 9(10):e1003877, 2013.

[306] T. Rognes, T. Flouri, B. Nichols, C. Quince, and F. Mahé. VSEARCH: a versatile open source tool for metagenomics. *PeerJ*, 4:e2584, 2016.

[307] F. Ronquist and J. P. Huelsenbeck. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19(12):1572–1574, 2003.

[308] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[309] N. Saitou and M. Nei. The Neighbor-Joining Method: A new Method for Reconstructing Phylogenetic Trees. *Molecular Biology and Evolution*, 4(4): 406–425, 1987.

[310] L. Salichos and A. Rokas. Inferring ancient divergences requires genes with strong phylogenetic signals. *Nature*, 497(7449):327–331, 2013.

[311] F. Sanger and A. Coulson. A Rapid Method for Determining Sequences in DNA by Primed Synthesis with DNA Polymerase. *Journal of Molecular Biology*, 94(3):441–448, 1975.

[312] F. Sanger, S. Nicklen, and A. R. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, 74(12):5463–7, 1977.

[313] D. Sankoff. Minimal Mutation Trees of Sequences. *SIAM Journal on Applied Mathematics*, 28(1):35–42, 1975.

[314] V. Savolainen, R. S. Cowan, A. P. Vogler, G. K. Roderick, and R. Lane. Towards Writing the Encyclopedia of Life: An Introduction to DNA Barcoding. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 360(1462):1805–11, 2005.

[315] E. W. Sayers, T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, R. Edgar, S. Federhen, M. Feolo, L. Y. Geer, W. Helmberg, Y. Kapustin, D. Landsman, D. J. Lipman, T. L. Madden, D. R. Maglott, V. Miller, I. Mizrachi, J. Ostell, K. D. Pruitt, G. D. Schuler, E. Sequeira, S. T. Sherry, M. Shumway, K. Sirotkin, A. Souvorov, G. Starchenko, T. A. Tatusova, L. Wagner, E. Yaschenko, and J. Ye. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, 37(Database):D5–D15, 2009.

[316] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1): 61–80, 2009.

[317] B. Schaeffer, M. K. Hecht, and N. Eldredge. Phylogeny and Paleontology. In T. Dobzhansky, editor, *Evolutionary Biology*, pages 31–46. Springer US, New York, NY, 1972.

[318] T. S. B. Schmidt, J. F. Matias Rodrigues, and C. von Mering. Ecological Consistency of SSU rRNA-Based Operational Taxonomic Units at a Global Scale. *PLoS Computational Biology*, 10(4):e1003594, 2014.

[319] A. O. Schmitt and H. Herzel. Estimating the Entropy of DNA Sequences. *Journal of Theoretical Biology*, 188(3):369–377, 1997.

[320] M. B. Scholz, C. C. Lo, and P. S. G. Chain. Next generation sequencing and bioinformatic bottlenecks: The current state of metagenomic data analysis. *Current Opinion in Biotechnology*, 23(1):9–15, 2012.

[321] R. Schwartz and A. A. Schäffer. The Evolution of Tumour Phylogenetics: Principles and Practice. *Nature Reviews Genetics*, 18(4):213–229, 2017.

[322] A. Sczyrba, P. Hofmann, P. Belmann, D. Koslicki, S. Janssen, J. Dröge, I. Gregor, S. Majda, J. Fiedler, E. Dahms, A. Bremges, A. Fritz, R. Garrido-Oter,

T. S. Jørgensen, N. Shapiro, P. D. Blood, A. Gurevich, Y. Bai, D. Turaev, M. Z. DeMaere, R. Chikhi, N. Nagarajan, C. Quince, F. Meyer, M. Balvočiūtė, L. H. Hansen, S. J. Sørensen, B. K. H. Chia, B. Denis, J. L. Froula, Z. Wang, R. Egan, D. Don Kang, J. J. Cook, C. Deltel, M. Beckstette, C. Lemaitre, P. Peterlongo, G. Rizk, D. Lavenier, Y.-W. Wu, S. W. Singer, C. Jain, M. Strous, H. Klingenberg, P. Meinicke, M. D. Barton, T. Lingner, H.-H. Lin, Y.-C. Liao, G. G. Z. Silva, D. A. Cuevas, R. A. Edwards, S. Saha, V. C. Piro, B. Y. Renard, M. Pop, H.-P. Klenk, M. Göker, N. C. Kyrpides, T. Woyke, J. A. Vorholt, P. Schulze-Lefert, E. M. Rubin, A. E. Darling, T. Rattei, and A. C. McHardy. Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software. *Nature Methods*, 14(11):1063–1071, 2017.

[323] 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):811–814, 2012.

[324] N. Shah, M. G. Nute, T. Warnow, and M. Pop. Misunderstood parameter of NCBI BLAST impacts the correctness of bioinformatics workflows. *Bioinformatics*, 2018.

[325] C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, 1951.

[326] H. Shimodaira. An Approximately Unbiased Test of Phylogenetic Tree Selection. *Systematic Biology*, 51(3):492–508, 2002.

[327] H. Shimodaira and M. Hasegawa. Multiple Comparisons of Log-Likelihoods with Applications to Phylogenetic Inference. *Molecular Biology and Evolution*, 16(8):1114, 1999.

[328] J.-J. Shu. A new integrated symmetrical table for genetic codes. *Biosystems*, 151:21–26, 2017.

[329] G. G. Z. Silva, D. A. Cuevas, B. E. Dutilh, and R. A. Edwards. FOCUS: an alignment-free model to identify organisms in metagenomes using non-negative least squares. *PeerJ*, 2:e425, 2014.

[330] J. D. Silverman, A. D. Washburne, S. Mukherjee, and L. A. David. A phylogenetic transform enhances analysis of compositional microbiota data. *eLife*, 6:e21887, 2017.

[331] S. Skansi. *Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence.* Springer, 2018.

[332] T. F. Smith and M. S. Waterman. Identification of Common Molecular Subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[333] R. R. Sokal and C. Michener. A Statistical Method for Evaluating Systematic Relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958.

[334] R. R. Sokal and P. H. A. Sneath. *Principles of Numerical Taxonomy*. W.H. Freeman, San Francisco, 1963.

[335] P. S. Soltis and D. E. Soltis. Applying the Bootstrap in Phylogeny Reconstruction Applying the Bootstrap in Phylogeny Reconstruction. *Statistical Science*, 18(2):256–267, 2003.

[336] T. Sørensen. A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons. *Biologiske Skrifter*, 5(4):1–34, 1948.

[337] H. Soueidan and M. Nikolski. Machine learning for metagenomics: methods and tools. *Metagenomics*, 1, 2015.

[338] Sponk. Comparison of a single-stranded RNA and a double-stranded DNA with their corresponding nucleobases, 2010. Online: `https://commons.wikimedia.org/wiki/File:Difference_DNA_RNA-DE.svg`. Accessed: 2018-08-04. The image is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license, see `https://creativecommons.org/licenses/by-sa/3.0/deed.en`. It is based on [183], which is published under the same license.

[339] S. Srinivasan, N. G. Hoffman, M. T. Morgan, F. A. Matsen, T. L. Fiedler, R. W. Hall, F. J. Ross, C. O. McCoy, R. Bumgarner, J. M. Marrazzo, and D. N. Fredricks. Bacterial communities in women with bacterial vaginosis: High resolution phylogenetic analyses reveal relationships of microbiota to clinical criteria. *PLOS ONE*, 7(6):e37818, 2012.

[340] R. Staden. A strategy of DNA sequencing employing computer programs. *Nucleic Acids Research*, 6(7):2601–2610, 1979.

[341] A. Stamatakis. Phylogenetic models of rate heterogeneity: A high performance computing perspective. In P. Spirakis and H. J. Siegel, editors, *20th International Parallel and Distributed Processing Symposium, (IPDPS) 2006*, page 278, Rhodes Island, Greece, 2006. IEEE.

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

[343] L. Stein. Genome Annotation: From Sequence to Biology. *Nature Reviews Genetics*, 2(7):493–503, 2001.

[344] H. Steinhaus. Sur la division des corp materiels en parties. *Bulletin L'Académie Polonaise des Science*, 1(804):801, 1956.

[345] Z. D. Stephens, S. Y. Lee, F. Faghri, R. H. Campbell, C. Zhai, M. J. Efron, R. Iyer, M. C. Schatz, S. Sinha, and G. E. Robinson. Big Data: Astronomical or Genomical? *PLoS Biology*, 13(7):e1002195, 2015.

[346] T. Stoeck, D. Bass, M. Nebel, R. Christen, M. D. M. Jones, H.-W. Breiner, and T. A. Richards. Multiple marker parallel tag environmental DNA sequencing reveals a highly complex eukaryotic community in marine anoxic water. *Molecular Ecology*, 19(s1):21–31, 2010.

[347] K. Strimmer and A. Rambaut. Inferring confidence sets of possibly misspecified gene trees. *Proceedings of the Royal Society of London B: Biological Sciences*, 269(1487):137–142, 2002.

[348] M. A. Suchard, P. Lemey, G. Baele, D. L. Ayres, A. J. Drummond, and A. Rambaut. Bayesian phylogenetic and phylodynamic data integration using BEAST 1.10. *Virus Evolution*, 4(1), 2018.

[349] 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, P. Bork, J. Dore, S. D. Ehrlich, A. Stamatakis, and P. Bork. Metagenomic species profiling using universal phylogenetic marker genes. *Nature Methods*, 10(12):1196, 2013.

[350] S. Sunagawa, L. P. Coelho, S. Chaffron, J. R. Kultima, K. Labadie, G. Salazar, B. Djahanschiri, G. Zeller, D. R. Mende, A. Alberti, F. M. Cornejo-castillo, P. I. Costea, C. Cruaud, F. Ovidio, S. Engelen, I. Ferrera, J. M. Gasol, L. Guidi, F. Hildebrand, F. Kokoszka, C. Lepoivre, F. D\textquoterightOvidio, S. Engelen, I. Ferrera, J. M. Gasol, L. Guidi, F. Hildebrand, F. Kokoszka, C. Lepoivre, G. Lima-Mendez, J. Poulain, B. T. Poulos, M. Royo-Llonch, H. Sarmento, S. Vieira-Silva, C. Dimier, M. Picheral, S. Searson, S. Kandels-Lewis, C. Bowler, C. de Vargas, G. Gorsky, N. Grimsley, P. Hingamp, D. Iudicone, O. Jaillon, F. Not, H. Ogata, S. Pesant, S. Speich, L. Stemmann, M. B. Sullivan, J. Weissenbach, P. Wincker, E. Karsenti, J. Raes, S. G. Acinas, and P. Bork. Structure and function of the global ocean microbiome. *Science*, 348 (6237):1–10, 2015.

[351] O. Tanaseichuk, J. Borneman, and T. Jiang. Phylogeny-based classification of microbial communities. *Bioinformatics*, 30(4):449–456, 2014.

[352] S. Tavaré. Some probabilistic and statistical problems in the analysis of DNA sequences. *American Mathematical Society: Lectures on Mathematics in the Life Sciences*, 17:57–86, 1986.

[353] L. Tedersoo, M. Bahram, S. Põlme, U. Kõljalg, N. S. Yorou, R. Wijesundera, L. V. Ruiz, A. M. Vasco-Palacios, P. Q. Thu, A. Suija, M. E. Smith, C. Sharp, E. Saluveer, A. Saitta, M. Rosas, T. Riit, D. Ratkowsky, K. Pritsch, K. Põldmaa, M. Piepenbring, C. Phosri, M. Peterson, K. Parts, K. Pärtel, E. Otsing, E. Nouhra, A. L. Njouonkou, R. H. Nilsson, L. N. Morgado, J. Mayor, T. W. May, L. Majuakim, D. J. Lodge, S. S. Lee, K.-H. Larsson, P. Kohout, K. Hosaka, I. Hiiesalu, T. W. Henkel, H. Harend, L.-d. Guo, A. Greslebin,

G. Grelet, J. Geml, G. Gates, W. Dunstan, C. Dunk, R. Drenkhan, J. Dearnaley, A. De Kesel, T. Dang, X. Chen, F. Buegger, F. Q. Brearley, G. Bonito, S. Anslan, S. Abell, and K. Abarenkov. Global diversity and geography of soil fungi. *Science*, 346(6213):1256688, 2014.

[354] B. Temperton and S. J. Giovannoni. Metagenomics: Microbial diversity through a scratched lens. *Current Opinion in Microbiology*, 15(5):605–612, 2012.

[355] 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):e18093, 2011.

[356] L. R. Thompson, J. G. Sanders, D. McDonald, A. Amir, J. Ladau, K. J. Locey, R. J. Prill, A. Tripathi, S. M. Gibbons, G. Ackermann, J. A. Navas-Molina, S. Janssen, E. Kopylova, Y. Vázquez-Baeza, A. González, J. T. Morton, S. Mirarab, Z. Zech Xu, L. Jiang, M. F. Haroon, J. Kanbar, Q. Zhu, S. Jin Song, T. Kosciolek, N. A. Bokulich, J. Lefler, C. J. Brislawn, G. Humphrey, S. M. Owens, J. Hampton-Marcell, D. Berg-Lyons, V. McKenzie, N. Fierer, J. A. Fuhrman, A. Clauset, R. L. Stevens, A. Shade, K. S. Pollard, K. D. Goodwin, J. K. Jansson, J. A. Gilbert, R. Knight, and T. E. M. P. Consortium. A communal catalogue reveals Earth's multiscale microbial diversity. *Nature*, 2017.

[357] R. L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.

[358] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[359] J. Tonini, A. Moore, D. Stern, M. Shcheglovitova, and G. Ortí. Concatenation and Species Tree Methods Exhibit Statistically Indistinguishable Accuracy under a Range of Simulated Conditions. *PLoS Currents*, 7, 2015.

[360] M. C. B. Tsilimigras and A. A. Fodor. Compositional data analysis of the microbiome: fundamentals, tools, and challenges. *Annals of Epidemiology*, 26 (5):330–335, 2016.

[361] L. J. P. Van Der Maaten and G. E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[362] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, J. D. Gocayne, P. Amanatides, R. M. Ballew, D. H. Huson, J. R. Wortman, Q. Zhang, C. D. Kodira, X. H. Zheng, L. Chen, M. Skupski, G. Subramanian, P. D. Thomas, J. Zhang, G. L. Gabor Miklos, C. Nelson, S. Broder, A. G. Clark, J. Nadeau, V. A. McKusick, N. Zinder, A. J. Levine, R. J. Roberts, M. Simon, C. Slayman,

M. Hunkapiller, R. Bolanos, A. Delcher, I. Dew, D. Fasulo, M. Flanigan, L. Florea, A. Halpern, S. Hannenhalli, S. Kravitz, S. Levy, C. Mobarry, K. Reinert, K. Remington, J. Abu-Threideh, E. Beasley, K. Biddick, V. Bonazzi, R. Brandon, M. Cargill, I. Chandramouliswaran, R. Charlab, K. Chaturvedi, Z. Deng, V. Di Francesco, P. Dunn, K. Eilbeck, C. Evangelista, A. E. Gabrielian, W. Gan, W. Ge, F. Gong, Z. Gu, P. Guan, T. J. Heiman, M. E. Higgins, R. R. Ji, Z. Ke, K. A. Ketchum, Z. Lai, Y. Lei, Z. Li, J. Li, Y. Liang, X. Lin, F. Lu, G. V. Merkulov, N. Milshina, H. M. Moore, A. K. Naik, V. A. Narayan, B. Neelam, D. Nusskern, D. B. Rusch, S. Salzberg, W. Shao, B. Shue, J. Sun, Z. Wang, A. Wang, X. Wang, J. Wang, M. Wei, R. Wides, C. Xiao, C. Yan, A. Yao, J. Ye, M. Zhan, W. Zhang, H. Zhang, Q. Zhao, L. Zheng, F. Zhong, W. Zhong, S. Zhu, S. Zhao, D. Gilbert, S. Baumhueter, G. Spier, C. Carter, A. Cravchik, T. Woodage, F. Ali, H. An, A. Awe, D. Baldwin, H. Baden, M. Barnstead, I. Barrow, K. Beeson, D. Busam, A. Carver, A. Center, M. L. Cheng, L. Curry, S. Danaher, L. Davenport, R. Desilets, S. Dietz, K. Dodson, L. Doup, S. Ferriera, N. Garg, A. Gluecksmann, B. Hart, J. Haynes, C. Haynes, C. Heiner, S. Hladun, D. Hostin, J. Houck, T. Howland, C. Ibegwam, J. Johnson, F. Kalush, L. Kline, S. Koduru, A. Love, F. Mann, D. May, S. McCawley, T. McIntosh, I. McMullen, M. Moy, L. Moy, B. Murphy, K. Nelson, C. Pfannkoch, E. Pratts, V. Puri, H. Qureshi, M. Reardon, R. Rodriguez, Y. H. Rogers, D. Romblad, B. Ruhfel, R. Scott, C. Sitter, M. Smallwood, E. Stewart, R. Strong, E. Suh, R. Thomas, N. N. Tint, S. Tse, C. Vech, G. Wang, J. Wetter, S. Williams, M. Williams, S. Windsor, E. Winn-Deen, K. Wolfe, J. Zaveri, K. Zaveri, J. F. Abril, R. Guigó, M. J. Campbell, K. V. Sjolander, B. Karlak, A. Kejariwal, H. Mi, B. Lazareva, T. Hatton, A. Narechania, K. Diemer, A. Muruganujan, N. Guo, S. Sato, V. Bafna, S. Istrail, R. Lippert, R. Schwartz, B. Walenz, S. Yooseph, D. Allen, A. Basu, J. Baxendale, L. Blick, M. Caminha, J. Carnes-Stine, P. Caulk, Y. H. Chiang, M. Coyne, C. Dahlke, A. Mays, M. Dombroski, M. Donnelly, D. Ely, S. Esparham, C. Fosler, H. Gire, S. Glanowski, K. Glasser, A. Glodek, M. Gorokhov, K. Graham, B. Gropman, M. Harris, J. Heil, S. Henderson, J. Hoover, D. Jennings, C. Jordan, J. Jordan, J. Kasha, L. Kagan, C. Kraft, A. Levitsky, M. Lewis, X. Liu, J. Lopez, D. Ma, W. Majoros, J. McDaniel, S. Murphy, M. Newman, T. Nguyen, N. Nguyen, M. Nodell, S. Pan, J. Peck, M. Peterson, W. Rowe, R. Sanders, J. Scott, M. Simpson, T. Smith, A. Sprague, T. Stockwell, R. Turner, E. Venter, M. Wang, M. Wen, D. Wu, M. Wu, A. Xia, A. Zandieh, and X. Zhu. The Sequence of the Human Genome. *Science*, 291(5507):1304–51, 2001.

[363] K. Vervier, P. Mahé, M. Tournoud, J.-B. Veyrieras, and J.-P. Vert. Large-scale machine learning for metagenomics sequence classification. *Bioinformatics*, 32 (7):1023–1032, 2015.

[364] F. Villalobos, M. Á. Olalla-Tárraga, M. V. Cianciaruso, T. F. Rangel, and J. A. F. Diniz-Filho. Global patterns of mammalian co-occurrence: phylogenetic and body size structure within species ranges. *Journal of Biogeography*, 44(1):136–146, 2017.

[365] C. Villani. *Optimal transport: old and new*. Springer Science & Business Media, 2008.

[366] S. Vinga. Information theory applications for biological sequence analysis. *Briefings in Bioinformatics*, 15(3):376–389, 2014.

[367] S. Vinga and J. Almeida. Alignment-free sequence comparison - A review. *Bioinformatics*, 19(4):513–523, 2003.

[368] S. Vinga and J. S. Almeida. Rényi continuous entropy of DNA sequences. *Journal of Theoretical Biology*, 231(3):377–388, 2004.

[369] M. Vingron and M. S. Waterman. Sequence Alignment and Penalty Choice. Review of Concepts, Case Studies and Implications. *Journal of Molecular Biology*, 235(1):1–12, 1994.

[370] N. X. Vinh, J. Epps, and J. Bailey. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010.

[371] K. V. Voelkerding, S. A. Dames, and J. D. Durtschi. Next-Generation Sequencing: From Basic Research to Diagnostics. *Clinical Chemistry*, 55(4): 641–58, 2009.

[372] C. von Mering, P. Hugenholtz, J. Raes, S. G. Tringe, T. Doerks, L. J. Jensen, N. Ward, and P. Bork. Quantitative Phylogenetic Assessment of Microbial Communities in Diverse Environments. *Science*, 315(5815):1126–1130, 2007.

[373] M. Wainberg, D. Merico, A. Delong, and B. J. Frey. Deep learning in biomedicine. *Nature Biotechnology*, 36(9):829–838, 2018.

[374] L. Wang and T. Jiang. On the Complexity of Multiple Sequence Alignment. *Journal of Computational Biology*, 1(4):337–348, 1994.

[375] W.-L. Wang, S.-Y. Xu, Z.-G. Ren, L. Tao, J.-W. Jiang, and S.-S. Zheng. Application of metagenomics in the human gut microbiome. *World Journal of Gastroenterology*, 21(3):803–814, 2015.

[376] A. D. Washburne, J. D. Silverman, J. W. Leff, D. J. Bennett, J. L. Darcy, S. Mukherjee, N. Fierer, and L. A. David. Phylogenetic factorization of compositional data yields lineage-level associations in microbiome datasets. *PeerJ*, 5:e2969, 2017.

[377] A. D. Washburne, J. D. Silverman, J. T. Morton, D. Becker, D. Crowley, S. Mukherjee, L. A. David, and R. K. Plowright. Phylofactorization - a graph partitioning algorithm to identify phylogenetic scales of ecological data. *bioRxiv*, page 235341, 2018.

[378] A. D. Washburne, J. D. Silverman, J. T. Morton, D. J. Becker, D. Crowley, S. Mukherjee, L. A. David, and R. K. Plowright. Phylofactorization: a graph partitioning algorithm to identify phylogenetic scales of ecological data. *Ecological Monographs*, (e01353), 2019.

[379] J. D. Watson and F. H. C. Crick. Molecular Structure of Nucleic Acids. *Nature*, 171:737–738, 1953.

[380] N. S. Watson-Haigh. Deinterleave FASTQ files, 2012. Online: `https://gist.github.com/nathanhaigh/3521724`. Accessed: 2018-07-04.

[381] W. G. Weisburg, S. M. Barns, D. A. Pelletier, and D. J. Lane. 16S Ribosomal DNA Amplification for Phylogenetic Study. *Journal of Bacteriology*, 173(2): 697–703, 1991.

[382] S. Weiss, Z. Z. Xu, S. Peddada, A. Amir, K. Bittinger, A. Gonzalez, C. Lozupone, J. R. Zaneveld, Y. Vázquez-Baeza, A. Birmingham, E. R. Hyde, and R. Knight. Normalization and microbial differential abundance strategies depend upon data characteristics. *Microbiome*, 5(1):27, 2017.

[383] K. A. Wetterstrand. DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP), 2018. Online: `https://www.genome.gov/sequencingcostsdata`. Accessed: 2018-07-24.

[384] S. Whelan and N. Goldman. A General Empirical Model of Protein Evolution Derived from Multiple Protein Families Using a Maximum-Likelihood Approach. *Molecular Biology and Evolution*, 18(5):691–699, 2001.

[385] C. R. Woese and G. E. Fox. Phylogenetic structure of the prokaryotic domain: the primary kingdoms. *Proceedings of the National Academy of Sciences of the United States of America*, 74(11):5088–90, 1977.

[386] C. R. Woese, O. Kandler, and M. L. Wheelis. Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya. *Proceedings of the National Academy of Sciences of the United States of America*, 87(12):4576–9, 1990.

[387] X. Xia, Z. Xie, M. Salemi, L. Chen, and Y. Wang. An index of substitution saturation and its application. *Molecular Phylogenetics and Evolution*, 26(1): 1–7, 2003.

[388] Z. Yang. Statistical Properties of the Maximum Likelihood Method of Phylogenetic Estimation and Comparison with Distance Matrix Methods. *Systematic Biology*, 43(3):329–342, 1994.

[389] Z. Yang. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *Journal of Molecular Evolution*, 39(3):306–314, 1994.

[390] Z. Yang. A Space-Time Process Model for the Evolution of DNA Sequences. *Genetics*, 139(2), 1995.

[391] Z. Yang. *Computational Molecular Evolution*. Oxford University Press, 2006.

[392] Z. Yang. *Molecular Evolution: A Statistical Approach*. Oxford University Press, 2014.

[393] Yikrazuul. Base pair Adenine Tyhmine (AT), 2008. Online: `https://en.wikipedia.org/wiki/File:Base_pair_AT.svg`. Accessed: 2018-08-04. The image was released into the public domain.

[394] Yikrazuul. Base pair Guanine Cytosine (GT), 2008. Online: `https://en.wikipedia.org/wiki/File:Base_pair_GC.svg`. Accessed: 2018-08-04. The image was released into the public domain.

[395] P. Yilmaz, L. W. Parfrey, P. Yarza, J. Gerken, E. Pruesse, C. Quast, T. Schweer, J. Peplies, W. Ludwig, and F. O. Gl?ckner. The SILVA and "All-species Living Tree Project (LTP)" taxonomic frameworks. *Nucleic Acids Research*, 42(D1):D643–D648, 2014.

[396] T. J. Ypma. Historical Development of the Newton-Raphson Method. *SIAM Review*, 37(4):531–551, 1995.

[397] G. Yu, D. K. Smith, H. Zhu, Y. Guan, and T. T.-Y. Lam. ggtree: an r package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods in Ecology and Evolution*, 8(1):28–36, 2017.

[398] A. E. Zanne, D. C. Tank, W. K. Cornwell, J. M. Eastman, S. A. Smith, R. G. FitzJohn, D. J. McGlinn, B. C. O'Meara, A. T. Moles, P. B. Reich, D. L. Royer, D. E. Soltis, P. F. Stevens, M. Westoby, I. J. Wright, L. Aarssen, R. I. Bertin, A. Calaminus, R. Govaerts, F. Hemmings, M. R. Leishman, J. Oleksyn, P. S. Soltis, N. G. Swenson, L. Warman, and J. M. Beaulieu. Three keys to the radiation of angiosperms into freezing environments. *Nature*, 506(7486): 89–92, 2014.

[399] A. Zelezniak, S. Andrejev, O. Ponomarova, D. R. Mende, P. Bork, and K. R. Patil. Metabolic dependencies drive species co-occurrence in diverse microbial communities. *Proceedings of the National Academy of Sciences of the United States of America*, 112(20):6449–54, 2015.

[400] J. Zhang, P. Kapli, P. Pavlidis, and A. Stamatakis. A general species delimitation method with applications to phylogenetic placements. *Bioinformatics*, 29(22):2869–2876, 2013.

[401] J. Zhang, K. Kobert, T. Flouri, and A. Stamatakis. PEAR: A fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics*, 30(5):614–620, 2014.

[402] S. K. Zhou and R. Chellappa. From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel hilbert space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):917–929, 2006.

[403] X. Zhou, S. Lutteropp, L. Czech, A. Stamatakis, M. von Looz, and A. Rokas. Quartet-based computations of internode certainty provide accurate and robust measures of phylogenetic incongruence. *bioRxiv*, 2017.

[404] X. Zhou, X.-X. Shen, C. T. Hittinger, and A. Rokas. Evaluating Fast Maximum Likelihood-Based Phylogenetic Programs Using Empirical Phylogenomic Data Sets. *Molecular Biology and Evolution*, 35(2):486–503, 2017.

[405] E. Zuckerkandl and L. Pauling. Molecules as Documents of Evolutionary History. *Journal of Theoretical Biology*, 8(2):357–366, 1965.

[406] D. J. Zwickl. *Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion*. Dissertation, The University of Texas at Austin, 2006.