From the University of Lübeck and Max
Planck Institute for Evolutionary Biology
Research Group Leader: Honorarprofessor Dr.
Bernhard Haubold

Models and Algorithms for Phylogenetic Marker Analysis

Dissertation
for Fulfillment of
Requirements
for the Doctoral Degree
of the University of Lübeck

Submitted by

Jiajie Zhang
from Zhengzhou, Henan, China

Lübeck 15.10.2014

First referee: Prof. Dr. Bernhard Haubold

Second referee: Prof. Dr. Alexandros Stamatakis

Date of oral examination: 23.03.2015

Approved for printing. Lübeck, 27.03.2015

Abstract

Phylogenetic markers are widely used in DNA barcoding, DNA taxonomy, and amplicon based metagenomics. In this thesis, we address problems with phylogenetic marker analysis by developing novel models and algorithms. We present the Poisson Tree Processes (PTP) model for species delimitation using single-locus phylogenetic markers. We develop and test algorithms for maximum likelihood inference under the PTP model, and extend the PTP model using a Bayesian framework. Further, employing the species delimitation method, we develop a new algorithm - PhyloMap, for visualizing large phylogenetic marker data sets. We also describe and make available an accurate and robust paired-end reads merger for the Illumina Next-Generation Sequencing (NGS) platform. Finally, we integrate PTP with the Evolutionary Placement Algorithm (EPA) to delimit species in amplicon based metagenomic data.

Zusammenfassung

Phylogenetische Marker finden breite Anwendung in DNA barcoding, in der DNA Taxanomie und in der amplicon-basierten Metagenomik. In dieser Arbeit identifizieren und lösen wir aktuelle Probleme bei der Analyse phylogenetischer Marker, indem wir neue Algorithmen und Modelle hierfür entwickeln. Wir führen das Poisson Tree Process (PTP) Modell zur Eingrenzung von Spezies ein. PTP verwendet phylogenetische Marker, die aus einem einzigen Lokus bestehen. Wir entwickeln und evaluieren Algorithmen zur Maximum Likelihood Berechnung mit Hilfe des PTP Modells und erweitern das PTP Modell um bayesianische Statistik. Anhand der Ergebnisse der Spezieseingrenzung entwickeln wir einen neuen Algorithmus (PhyloMap) für die Visualisierung großer Datensätze, die aus phylogenetischen Markern bestehen. Des Weiteren beschreiben wir einen exakten und robusten paired-end read merger für die Next-Generation Sequenzierungsplatform von Illumina. Außerdem integrieren wir PTP in den evolutionären Platzierungsalgorithmus (evolutionary placement algorithm), um Spezies anhand von amplikon-basierten metagenomischen Daten voneinander abzugrenzen.

Acknowledgments

Studying in Germany has been an extraordinary experience for me. New scientific ideas mingled with eastern and western culture conflicts always excite me to explore the unknown further. First, I want to express my gratitude to Prof. Bernhard Haubold who kindly reviewed my thesis. I am particularly grateful to Prof. Alexandros Stamatakis and Prof. Thomas Martinetz who stood up and supported me during the most difficulty time of my PhD study, and without whom, I might never have had the chance to write down my thesis. I want to thank my mentors Prof. Amir Madany Mamlouk and Dr. Pavlos Pavlidis who were always there when I needed help. I also want to express my gratitude to Prof. Norbert Tautz who gave critical advice on my scientific career.

I am extremely grateful to all my colleagues in the Exelixis lab (Fernando Izquierdo, Andre J. Aberer, Alexey Kozlov, Paschalia Kapli, Solon Pissis, Kassian Kobert, and Dr. Tomas Flouris) and in the University of Lübeck (Dr. Lei chen, Chaoqun Jiang and Yijing Xie), who offered advice and support when it was needed. In particular to Dr. Tomas Flouris who helped me on many projects and who is also my close friend. I want to thank Prof. Rolf Hilgenfeld who supervised me at the beginning of my PhD study, and allowed me to leave his lab when I found my research interests to lie somewhere else.

Finally, I want to dedicate this thesis to my parents, my wife Zheng Hao and my son William Hanning Zhang.

I was first funded by a scholarship from the Graduate School for Computing in Medicine and Life Sciences, University of Lübeck, Germany's Excellence Initiative [DFG GSC 235/1], and later by a scholarship from Heidelberg Institute for Theoretical Studies.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Computational molecular phylogenetics is the study of evolutionary relationships among groups of organisms through mathematical models of molecular sequences [187]. The molecular sequences used in such studies usually come from phylogenetic markers. Phylogenetic markers are molecular sequences ubiquitous in all organisms under study and carry strong phylogenetic signal [17, 75, 89, 135, 141]. Several commonly used phylogenetic markers in phylogenetic studies include the *16S ribosomal RNA* (16S) from prokaryotes [183], the *cytochrome c oxidase I gene* (COI) and the *cytochrome b gene* (cyt-b) from animals [172], the *RuBisCO large subunit* (rbcL) from plants [147], and the *internal transcribed spacer* (ITS) from fungi [83]. Other phylogenetic markers, which are often house keeping genes, have also been identified and used in phylogenetic studies [4, 24, 76, 138, 176, 196].

Besides inferring phylogenies, phylogenetic markers have a plethora of additional applications, including DNA barcoding, DNA taxonomy, and metagenetics (amplicon based metagenomics).

DNA barcoding is a technique for species identification based on short DNA sequences [148]. Phylogenetic marker sequences from unknown individuals are compared to databases of voucher sequences with given taxonomic units or species classification. The main goal of DNA barcoding is not to discover or define new species, but to label the query sequence [168]. Thus, it requires a most comprehensive database. A large database, BOLD: The Barcode of Life Data System [131], has been constructed in an inter-

national effort to aid the acquisition, storage, analysis and publication of DNA barcode records. BOLD currently contains more than 3.2 million specimens (as of June 2014) with phylogenetic marker sequences for eukaryotes (`http://www.barcodinglife.org`).

DNA taxonomy, as its name suggests, uses phylogenetic marker sequences as taxonomic references [175]. DNA taxonomy differs from DNA barcoding because the central analytical task is to classify phylogenetic marker sequences into entities that correspond to species, rather than (re-)identifying known species [67, 167, 175]. Once species boundaries have been established, those phylogenetic markers can also be used to supplement and refine the existing DNA barcode databases [175].

With the advances in the Next-Generation Sequencing (NGS) technologies [118], phylogenetic markers are being used in metagenetic studies for profiling microbial communities [19]. NGS technologies combined with universal PCR primers [88], provide an efficient and cost-effective way for sequencing the hypervariable regions of phylogenetic markers (typically 16S rRNA), in a culture-independent way. Those marker sequences serve as a surrogate that allows us to investigate the composition and diversity of microbial communities. Using this technique, recent studies have linked the dysfunction of the human microbiota with diseases such as diabetes [129], obesity [93], vaginosis [132], and inflammatory bowl diseases (IBD) [59]. Similar metagenetic approaches have also been employed in studying microscopic eukaryotic biodiversity [12], and in DNA metabarcoding of plants and animals [26].

Metagenetic studies can yield a large amout of short reads (DNA sequences), a typical Illumina MiSeq run produces over 25 million reads (as of 2014, `http://www.illumina.com/systems/sequencing.ilmn`). Such large data sets pose new challenges for bioinformatics. Dedicated pipelines and tools for analyzing metagenetic data, such as QIIME [18], mothur [150] and UPARSE [45], are under active development. These approaches share three important steps. In the first step, the raw reads are preprocessed, for instance by merging paired-end reads, applying quality filters and removing adapters [12, 14]. In the second step, the reads are grouped into entities that are intended to correspond to species. However, this represents a significant challenge for metagenetic data analysis, due to the difficulties with the species concept in bacteria and the lack of robust methods. Thus, the reads are clustered into so-called Molecular Operational Taxonomic Units (MOTUs) based on a predefined, mostly arbitrary, sequence similarity cutoff. Here, MOTUs are considered as a proxy for species [26, 88]. In the third step, MOTUs are classified taxonomically [18, 179] by comparison to references sequences with known taxonomic classifications [28, 105, 130].

In this thesis, we address several problems relating to the analysis of phy-

logenetic marker data sets. Firstly, we introduce new statistical models for species delimitation. As already mentioned, species delimitation is key to most studies involving phylogenetic markers. Secondly, we develop a novel algorithm for visualizing large samples of phylogenetic markers. It also provides a means for visual inspection of species delimitation results. Thirdly, we introduce a software for merging the paired-end reads, which may increase the reads length. Longer reads carry more information and enable us to use phylogeny-aware methods for analyzing metagenetic data. Finally, we develop a phylogeny-aware analysis pipeline to delimit species on metagenetic data.

1.2 Scientific Contribution

One central analytical task in phylogenetic marker analysis is to delimit species using molecular sequences. Despite its importance, currently the only widely used species delimitation method for single locus data that deploys a species concept, and does not require *a priori* definition of group memberships (such as BP&P [188]), is the Generalized Mixed Yule Coalescent model (GMYC) [61, 62, 92]. We have developed a new approach called Poisson Tree Processes (PTP) model for species delimitation using single locus phylogenetic marker data. We show that, our PTP model outperforms the GMYC in terms of delimitation accuracy, and it also greatly simplifies species delimitation process by only requiring phylogenetic input trees, instead of ultrametric trees. We have also extended the PTP model using a Bayesian framework. The Bayesian PTP model can use a single, fixed phylogenetic tree, as well as sets of phylogenetic trees derived from Bayesian phylogenetic tree inferences. PTP has already been used to delimit species for many organisms (e.g., [13, 108, 157, 169, 173]), and PTP has also been applied to study virus lineages of hantaviruses [21].

Data sets used in DNA taxonomy sometimes involve a large number of sequences. A phylogenetic tree with up to a few hundreds sequences can be displayed on a computer screen, or be printed on an A4 sheet of paper. It becomes increasingly difficult to visualize larger phylogenetic trees. Thus, alternative visualization methods are needed to display large phylogenetic marker sets, and to inspect species delimitation results. In this thesis, we present a novel method called PhyloMap, for visualizing large phylogenetic marker data sets. It combines phylogenetic tree inference, species delimitation and principal coordinates analysis to generate an easy-to-interpret visualization of a large sequence data sets.

In early metagenetic studies, the Roche 454 platform was often considered

superior to the Illumina platform [65], because the Roche 454 produces reads of 350 to 450 bp, while the Illumina reads only range from 75 to 100 bp [19]. Recently, the Illumina MiSeq platform acquired the ability to produce $2*300$ bp paired-end reads. With a careful experimental design [65], the reads can span over 550 bp of target DNA fragments. The Illumina MiSeq platform can generate over 25 million reads per run, compared with 1 million reads per run on a 454 plate [182]. In order to leverage these advantages of the Illumina platform, the paired-end reads first must be merged. In this thesis, we describe an Illumina paired-end read merging software - PEAR. We show that, PEAR outperforms all competing mergers in terms of accuracy, false-positive rate, and run times.

As explained above, the specie concept, although biologically more meaningful, has rarely been deployed in metagenetic studies. We introduce an open reference species delimitation approach by integrating PTP with the Evolutionary Placement Algorithm [10] (EPA-PTP). The EPA-PTP pipeline is the first integrated approach for analyzing metagenetic data that combines the phylogenetic placement approach with an explicit statistical criterion for species delimitation. EPA-PTP represents the first step towards a full phylogeny-aware analysis pipeline for metagenetic data.

The scientific work presented in this thesis has been published in 3 journal articles ([191, 192, 193]). Research on other topics related to bioinformatics not covered by this thesis was published in 2 journal articles and 1 peer-reviewed conference paper. These articles covered work on Influenza virus database design and sequence analysis [22], human DNA methylation and cancer data collection and database construction [72], and the multiprocessor scheduling problem in phylogenetics [194].

1.3 Structure of The Thesis

The rest of this thesis is divided in three parts. Part 1 includes chapters 2 and 3; it gives the mathematical background, and provides a brief introduction to evolutionary models and phylogenetic tree inference. Part 2 includes chapters 4 to 6. This part covers the PTP model for species delimitation, the Bayesian extension of PTP, and the PhyloMap visualization approach. Part 3 focuses on NGS data analysis and includes chapters 7 and 8. Finally, we conclude and discuss future work in chapter 9.

CHAPTER 2

An Introduction to Stochastic Processes

This chapter introduces the basic concepts of stochastic processes, the mathematical basis of this thesis.

The probabilistic models for phylogenetic tree inference described in chapter 3 are Markov processes, the PTP model (chapter 4) for species delimitation is closely related to Poisson processes, and the Bayesian extension of the PTP model in chapter 5 uses Markov Chain Monte Carlo techniques to sample posterior distributions. Stochastic processes are also fundamental to many other applications in biological sequence analysis such as hidden Markov models for sequence similarity search and alignment [80, 136], and coalescent theory [178].

In the following, I introduce the basic concepts and main theories used in this thesis without proofs. More details are provided in text books on stochastic processes [40, 91]. Part of the notation and terminology in this chapter follows [52].

2.1 Important Probability Distributions

In this Section, I review important probability distributions and their relationships, namely the Bernoulli, Binomial, Geometric, Poisson, Exponential, and Gamma distributions. Throughout the text, we use $Prob(a)$ to denote the probability of a.

First, we review a few terms for probability and statistics.

Definition 1 *A random variable is a function that assigns a real number to each possible outcome in the sample space.*

Definition 2 *The cumulative distribution function F for a random variable X is defined as:*

$$F(x) = Prob\{X \leq x\} \ . \tag{2.1}$$

Definition 3 *For a discrete random variable X, a function f is called its probability mass function if*

$$f(x) = Prob\{X = x\} \tag{2.2}$$

for all x in the range of X.

Definition 4 *For a continuous random variable X, a function g is called its probability density function if*

$$\int_a^b g(u)du = Prob\{a \leq X \leq b\} \tag{2.3}$$

for all a, b in the range of X.

Definition 5 *The mean, or the expected value of a random variable X is*

$$E(X) = \sum xf(x) \tag{2.4}$$

for a discrete random variable, or

$$E(X) = \int xg(x)dx \tag{2.5}$$

for a continuous random variable.

If $\varphi(X)$ is a function of X, then the expected value of $\varphi(X)$ is

$$E[\varphi(X)] = \sum \varphi(X)f(x) \tag{2.6}$$

for a discrete random variable, or

$$E[\varphi(X)] = \int \varphi(X)g(x)dx \tag{2.7}$$

for a continuous random variable.

Definition 6 *The variance of a random variable X is:*

$$V(X) = E[(X - E(X))^2] = E[X^2] - (E(X))^2 \ . \tag{2.8}$$

Next, I introduce four discrete probability distributions.

Definition 7 *The random variable X follows a Bernoulli distribution if there exists p, where $0 < p < 1$ and the probability mass function of X can be written as:*

$$f(x) = p^x(1-p)^{1-x} \ for \ x \in \{0, 1\} \ . \tag{2.9}$$

The number p is often thought of as the probability of a success.

$$E(X) = p \ and \ V(X) = p(1-p) \ . \tag{2.10}$$

Definition 8 *If a Binomial random variable X is the sum of n independent Bernoulli distributed random variables with the same probability p of success, the probability mass function can be written as*

$$f(x) = \frac{n!}{x!(n-x)!} p^x(1-p)^{n-x} \ for \ x = 0, 1, 2, ..., n \ . \tag{2.11}$$

Thus, a binomial distribution explains the number of successes in repeated experiments. Note that n is given and forms part of a particular instance of a Binomial distribution. Thus, n is not a parameter. The name "binomial" comes from the binomial theorem: let $Q = p$ and $P = 1 - p$

$$\begin{aligned} (Q+P)^n &= Q^n + nQ^{n-1}P + ... + nQP^{n-1} + P^n \\ &= f(0) + f(1) + ... + f(n-1) + f(n) \ . \end{aligned} \tag{2.12}$$

$$E(X) = np \ and \ V(X) = np(1-p) \ . \tag{2.13}$$

Definition 9 *A random variable X has a geometric distribution if there exists p, $0 < p < 1$, and its probability mass function can be written as:*

$$f(x) = (1-p)^x p \ . \tag{2.14}$$

The geometric distribution describes the probabilities of repeated experiments until first success, i.e., x failures followed by a success.

$$E(X) = \frac{1-p}{p}, V(X) = \frac{1-p}{p^2} \ . \tag{2.15}$$

Definition 10 *A random variable X has a Poisson distribution if there is a real number $\lambda > 0$ such that the probability mass function of X can be written as*

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!} \; for \; x = 0, 1, ... \tag{2.16}$$

Here λ is the rate of occurrence of a specific event, that is, the average number of its occurrence per unit time.

$$E(X) = \lambda, \; V(X) = \lambda \tag{2.17}$$

The Poisson distribution is the limiting form of the Binomial distribution when n is large and p is small. If we let $\lambda := np$ and substitute p in Equation 2.11 by λ/n:

$$\begin{aligned} f(x) &= \frac{n!}{x!(n-x)!} p^x (1-p)^{n-x} \\ &= \frac{n!}{x!(n-x)!} \frac{\lambda^x}{n^x} (1 - \frac{\lambda}{n})^{n-x} \\ &= \underbrace{[\frac{n}{n} \frac{(n-1)}{n} \frac{(n-2)}{n} ... \frac{(n-x+1)}{n}]}_{\to 1} \frac{\lambda^x}{x!} \underbrace{[(1 - \frac{\lambda}{n})^{n-x}]}_{\to e^{-\lambda}} \; . \end{aligned} \tag{2.18}$$

As $n \to \infty$, the expression in the first square bracket will tend to 1 and the expression in the last square bracket will tend to $e^{-\lambda}$, because $(1 - \frac{\lambda}{n})^n \to e^{-\lambda}$ and $(1 - \frac{\lambda}{n})^{-x} \to 1$.

In the following, I introduce two continuous probability distributions.

Definition 11 *A random variable X has an exponential distribution if there is a number $\lambda > 0$ such that the probability density function of X is*

$$f(x) = \lambda e^{-\lambda x} \; for \; x \geq 0 \; . \tag{2.19}$$

The cumulative probability function is:

$$F(x) = Prob(X \leq x) = \int_0^x \lambda e^{-\lambda x} dx = 1 - e^{-\lambda x} \; for \; x \geq 0 \; . \tag{2.20}$$

$$E(X) = \frac{1}{\lambda}, V(X) = \frac{1}{\lambda^2} \; . \tag{2.21}$$

The exponential distribution is the continuous analogue of the geometric distribution, it is often used to model waiting times between events. To see the connection between the Exponential distribution and the Poisson distribution, consider a certain event occurring with rate λ. Then, the average

number of events that occur in time t is λt. Hence, the number of events in time t follows a Poisson distribution with mean λt. The probability of no event taking place within t is $e^{-\lambda t}$, that is

$$\begin{aligned} Prob[X > t] &= e^{-\lambda t} \\ Prob[X \leq t] &= 1 - e^{-\lambda t} \ . \end{aligned} \tag{2.22}$$

The exponential distribution has the so-called "lack of memory" property:

$$Prob\{X > t + s | X > t\} = Prob\{X > s\} \ . \tag{2.23}$$

Given two sets A and B, we use the fact that if $B \subset A$, then $Prob(B|A) = Prob(B)/Prob(A)$:

$$\begin{aligned} Prob\{X > t + s | X > t\} = \frac{Prob\{X > t + s\}}{Prob\{X > t\}} &= \frac{e^{-\lambda(t+s)}}{e^{-\lambda t}} = e^{-\lambda s} \\ &= Prob\{X > s\} \ . \end{aligned} \tag{2.24}$$

Finally, we introduce the Gamma distribution:

Definition 12 *A random variable X has a Gamma distribution if its probability density function can be written as*

$$f(x) = \frac{x^{\alpha-1} e^{-x/\beta}}{\beta^{\alpha} \int_0^{\infty} y^{\alpha-1} e^{-y} dy} \text{ for } x, \alpha, \beta > 0 \ . \tag{2.25}$$

Then

$$E(X) = \beta\alpha, V(X) = \beta^2\alpha \ , \tag{2.26}$$

where α is called the shape parameter, and β is called the scale parameter. Varying values of the shape parameter α result in different shapes of the probability density function, while varying the scale parameter tends to "stretch" or "compress" the probability density function along the x-axis. If α is an integer, then the distribution represents an Erlang distribution, that is, the sum of α independent exponentially distributed random variables, each of which has a rate parameter of $1/\beta$.

2.2 Markov Chains

First we give some basic definitions of stochastic processes:

Definition 13 *A stochastic process $\{X_n; n \geq 0\}$ is a sequence of random variables indexed by time n.*

If n is a subset of the nonnegative integers $\{0, 1, 2, ...\}$, we call the process a discrete time process; if n is a subset of the nonnegative real numbers $[0, \infty\}$, we call the process a continuous time process.

Definition 14 *The states of a stochastic process are the possible values of X_n, the set of all states is called the state space.*

The state space can be discrete, for instance, the four nucleotides A, T, G and C; or continuous, for instance, all the real numbers.

2.2.1 Basic Concepts

In this section we discuss a special class of stochastic processes that satisfy the Markov property. The Markov property states that, to predict the future state, it suffices to consider only the current state and not the history of states.

Definition 15 *Consider a discrete-time stochastic process and let X_n be the states. We say that $\{X_n\}$ is a discrete time Markov chain with a transition probability matrix $P(i, j)$, if for any $j, i, i_{n-1}, ...i_0$, $Prob(X_{n+1} = j|X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0) = P(i, j)$.*

The transition probability matrix $P(i, j) = Prob(X_{n+1} = j|X_n = i)$ has the following properties:

1. $P(i, j) \geq 0$.
2. $\sum_j P(i, j) = 1$, that is, the rows of the matrix $P(i, j)$ sum to 1, because the transition probability from state i to other states (including state i) must sum to 1.

$P(i, j)$ gives the probability of going from state i to state j in one step, and let $P^{(m)}(i, j) = Prob(X_{n+m} = j|X_n = i)$ be the probability of going from state i to state j in m steps. Before we show how to calculate $P^{(m)}(i, j)$, we first need to introduce the Chapman-Kolmogorov equation:

$$P^{(n)}(i, j) = \sum_{k=1}^{\infty} P^{(s)}(i, k)P^{(n-s)}(k, j), 0 \leq s \leq n. \tag{2.27}$$

Using the Chapman-Kolmogorov equation:

$$P^{(m+1)}(i, j) = \sum_{k=1}^{\infty} P^{(m)}(i, k)P(k, j) \ , \tag{2.28}$$

thus, we have

Theorem 2.2.1 $P^{(m)}(i, j) = \left[P^{(1)}(i, j)\right]^m$, *i.e., the m steps transition probability matrix is the m_{th} power of the one step transition probability matrix.*

2.2.2 Classification of States

In order to explain the main theorem of Markov chains in subsection 2.2.3, we first need to introduce some notation.

Definition 16 *Consider a state j in the state space of a Markov chain. We define a random variable T^j as the time it takes to reach state j from any other states for the first time. This random variable is called first passage times, mathematically,*

$$T^j = min\{n \geq 1 : X_n = j\} \ . \tag{2.29}$$

The first passage time is used when we are interested in whether we can reach certain states.

Definition 17 *The probability of reaching a state j at least once, given that the initial state was i, is called the first passage probability, denoted by $F(i,j)$, where:*

$$F(i,j) = Prob(T^j < \infty | X_0 = i) \ . \tag{2.30}$$

Definition 18 *We define a random variable denoted by N^j, which is equal to the total number of visits to state j in n steps of the Markov chain, where $n \to \infty$.*

For N^j, we are interested in its expected value.

Definition 19 *The expected number of visits to state j given an initial state i is denoted by $R(i,j)$:*

$$R(i,j) = E[N^j | X_0 = i] \ . \tag{2.31}$$

Using the above notation, we can now define two types of states in Markov chains:

Definition 20 *A state j is called transient if $F(j,j) < 1$, or equivalently, $R(j,j) < \infty$. A state j is called recurrent if $F(j,j) = 1$, or equivalently, $R(j,j) = \infty$.*

A state j is transient if starting in state j, the Markov chain will eventually leave state j and never return; and a state j is recurrent if starting in state j, the Markov chain will continuously revisit state j.

Definition 21 *Let P be the transition probability matrix of a Markov chain, and C be a set of states contained in its state space. C is said to be closed if:*

$$\sum_{j \in C} P(i,j) = 1, \ for \ all \ i \in C \ . \tag{2.32}$$

Definition 22 *A closed set of states that does not contain a subset which is also closed is called an irreducible set. A single state forms an irreducible set by itself and is called an absorbing state. If a Markov chain reaches an absorbing state, the chain will never leave that state.*

Next we introduce two theorems that can help us to determine the class of the states and to identify irreducible sets.

Theorem 2.2.2 *All states within an irreducible set are of the same state type, that is, transient or recurrent.*

Definition 23 *State j can be reached from state i if there exists a positive integer n, such that $p^{(n)}(i,j) > 0$, this relationship is denoted as $i \to j$. If $i \to j$ and $j \to i$ both hold, we say i and j communicate, denoted as $i \leftrightarrow j$.*

Using the notion of communication, we have the following theory:

Theorem 2.2.3 *The closed set of states C is irreducible if and only if all states in C communicate with each other.*

Finally, we define the period of a state:

Definition 24 *The period of a state is the greatest common divisor of $\{n \geq 0 : P^{(n)}(x,x) > 0\}$, if state j has a period of 1, then state j is called aperiodic. If all states of the Markov chain are aperiodic, then the Markov chain is aperiodic.*

There is a simple way to check if a state is aperiodic:

Lemma 2.2.4 *If $P(x,x) > 0$, then state x has period 1.*

2.2.3 Steady-State Behavior

In this section we discuss the long-term behavior of Markov chains.

Theorem 2.2.5 *Let $\{X_n\}$ be a Markov chain with finite state space and a transition probability matrix $P(i,j)$. If the entire state space forms an irreducible, recurrent set, and all states are aperiodic, then*

$$\overrightarrow{\pi}(j) = \lim_{n\to\infty} Prob(X_n = j | X_0 = i), \ for\ any\ states\ i \tag{2.33}$$

$\overrightarrow{\pi}$ is the solution to:

$$\overrightarrow{\pi} P(i,j) = \overrightarrow{\pi}, and \sum_i \overrightarrow{\pi}(i) = 1\ . \tag{2.34}$$

Indeed, if we let a matrix

$$\Pi = \lim_{n\to\infty} P^{(n)}(i,j) \ . \tag{2.35}$$

The rows of Π are identical and each row equals $\overrightarrow{\pi}$. If we let $\overrightarrow{\gamma}$ be a vector whose elements sum to 1, then

$$\lim_{n\to\infty} \overrightarrow{\gamma} P^{(n)}(i,j) = \overrightarrow{\pi} \ , \tag{2.36}$$

$\overrightarrow{\pi}$ is called a stationary, equilibrium, or steady-state probability distribution of the Markov chain.

2.3 Markov Processes

2.3.1 Basic Definitions

Definition 25 *The continuous stochastic process $\{Y_t\}$ with a finite state space E is a Markov process if for all $j \in E$ and $t, s \leq 0$*

$$Prob(Y_{t+s} = j | Y_u; u \leq t) = Prob(Y_{t+s} | Y_t) \ . \tag{2.37}$$

The definition of Markov processes requires that the Markov property holds for all future times. If we think of time t as the present time, the left-hand side of Equation 2.37 predicts future time s from the present given all the past up to and including the current time t. The right-hand side of the equation tells us that the prediction only depends on the current time t.

There are two important elements in the Markov process: the times between events and the probabilities of switching to a certain state; so we have the following definitions:

Definition 26 *Let $\{Y_t\}$ be a Markov process with finite state space E and jump times denoted by $T_0, T_1, ...$ and the embedded Markov chain at jump time as $\{X_k\}$. The time between jumps, that is, $T_{n+1} - T_n$ is called the sojourn time, and it follows the exponential distribution for state $i \in E$ with rate $\lambda(i)$. The quantity $\lambda(i)$ is called the mean sojourn rate for state i. The embedded Markov chain has a transition probability matrix $P(i,j)$ with $P(i,i) = 0$ and satisfies:*

$$\begin{aligned} Prob(T_{n+1} - T_n \leq t | X_n = i) &= 1 - e^{-\lambda(i)t} \\ Prob(X_{n+1} = j | X_n = i) &= P(i,j) \ . \end{aligned} \tag{2.38}$$

Here, we set $P(i,i) = 0$ because for the embedded Markov chain, we assume the chain will always jump to a different state in the next step. The transition probability matrix of the embedded Markov chain and the mean sojourn rates can be combined into the so called generator matrix of the Markov process.

Definition 27 *The generator matrix Q for the Markov process is given by:*

$$Q(i,j) = \begin{cases} -\lambda(i) & \text{for } i = j \\ \lambda(i)P(i,j) & \text{for } i \neq j \end{cases} ,$$

where $Q(i,j)$ is the rate of the process going from state i to state j. Assume the transition probabilities $Prob^{(t)}(i,j)$ are continuous and differentiable for $t \geq 0$ and at $t = 0$

$$Prob^{(0)}(i,j) = 0 \text{ for } i \neq j;\ Prob^{(0)}(i,i) = 1 . \tag{2.39}$$

Mathematically, Q is defined as follows:

$$Q(i,j) = \lim_{\Delta t \to 0} \frac{Prob^{(\Delta t)}(i,j) - Prob^{(0)}(i,j)}{\Delta t} = \lim_{\Delta t \to 0} \frac{Prob^{(\Delta t)}(i,j)}{\Delta t} \text{ for } i \neq j \tag{2.40}$$

$$Q(i,i) = \lim_{\Delta t \to 0} \frac{Prob^{(\Delta t)}(i,i) - Prob^{(0)}(i,i)}{\Delta t} = \lim_{\Delta t \to 0} \frac{Prob^{(\Delta t)}(i,i) - 1}{\Delta t} . \tag{2.41}$$

Since the row sum of $Prob^{(t)}(i,j)$ must be 1 (Definition 15),

$$Prob^{(\Delta t)}(i,i) = 1 - \sum_{j=0, i \neq j}^{\infty} Prob^{(\Delta t)}(i,j) = 1 - \sum_{j=0, i \neq j}^{\infty} [Q(i,j)\Delta t + o(\Delta t)] , \tag{2.42}$$

thus

$$Q(i,i) = \lim_{\Delta t \to 0} \frac{-\sum_{j=0, i \neq j}^{\infty} [Q(i,j)\Delta t + o(\Delta t)]}{\Delta t} = -\sum_{j=0, i \neq j} Q(i,j) . \tag{2.43}$$

2.3.2 Steady-State Behavior

In discussing the steady-state behavior of the Markov chains(section 2.2), we had to classify states. Fortunately, the states of a Markov process can be easily classified using its embedded Markov chain. A state in a Markov

process is recurrent if it is recurrent in the embedded Markov chain; a state in a Markov process is transient if it is transient in the embedded Markov chain; a set of states is irreducible if it is irreducible for the embedded chain.

We use $\pi(j)$ to denote the steady-state probability of state j in the embedded Markov chain, then we have the steady state of probability for state j in a Markov process if all the states form an irreducible, recurrent set:

$$p(j) = \lim_{t\to\infty} Prob(Y_t = j|Y_0 = i) = \frac{\pi(j)/\lambda(j)}{\sum_{i\in E}\pi(i)/\lambda(i)} \ . \tag{2.44}$$

Using the generator matrix, we can derive the steady-state probabilities p directly, p is the solution to

$$\begin{aligned} pQ &= 0 \\ \sum_{j\in E} p(j) &= 1 \ . \end{aligned} \tag{2.45}$$

2.3.3 Time-dependent Transition Probabilities

In this section, we introduce how to calculate the transition probability of a Markov process over time t. We first review matrix exponentiation. Recall that for a real value a:

$$e^a = \sum_{n=0}^{\infty} \frac{a^n}{n!} \ , \tag{2.46}$$

analogously, we define the exponentiation of a matrix A as:

$$e^A = \sum_{n=0}^{\infty} \frac{A^n}{n!} \ . \tag{2.47}$$

Let $\{X_t\}$ be a Markov process with a generator matrix Q, then

$$\begin{aligned} &Prob(X_{t+\Delta t} = j|X_0 = i) \\ &= Prob(X_t = j|X_0 = i)Prob(X_{\Delta t} = j|X_t = j) \\ &+ \sum_{k\neq j} Prob(X_t = k|X_0 = i)Prob(X_{\Delta t} = j|X_t = k) \\ &= Prob(X_t = j|X_0 = i)[1 + Q(j,j)\Delta t + o(\Delta t)] \\ &+ \sum_{k\neq j} Prob(X_t = k|X_0 = i)[Q(k,j)\Delta t + o(\Delta t)] \ , \end{aligned} \tag{2.48}$$

subtracting $Prob(X_t = j|X_0 = i)$ from both sides, dividing by Δt and letting $\Delta t \to 0$:

$$Prob'(X_t = j|X_0 = i) = \sum_k Prob(X_t = k|X_0 = i)Q(k, j) \ . \tag{2.49}$$

Equation 2.49 is called the forward Kolmogorov differential equation. There is also another Equation called the backward Kolmogorov differential equation:

$$Prob'(X_t = j|X_0 = i) = \sum_k Q(i, k)Prob(X_t = j|X_0 = k) \ . \tag{2.50}$$

The solution to both the forward and the backward Kolmogorov differential equation with $Prob^{(0)}(i, i) = 1$ and $Prob^{(0)}(i, j) = 0$ for $i \neq j$ is

$$Prob^{(t)}(i, j) = e^{tQ}(i, j) \ . \tag{2.51}$$

An alternative way to understand Equation 2.51 is to consider $Prob^{(\Delta t)}(i, j) = I + Q\Delta t$, where I is the identity matrix. If we let $n = t/\Delta t$, then

$$Prob^{(t)}(i, j) = [Prob^{(\Delta t)}(i, j)]^n = (I + Qt/n)^n = e^{tQ}(i, j) \ . \tag{2.52}$$

For computational purposes, we try to write the Q matrix in the form $Q = ADA^{-1}$, where D is a diagonal matrix. The diagonal elements of D are the eigenvalues of Q, and the columns of A are the corresponding eigenvectors of Q. This is also known as the spectral decomposition of Q. Then computing Equation 2.51 can be simplified to:

$$Prob^{(t)}(i, j) = e^{tQ}(i, j) = Ae^{tD}A^{-1} \ . \tag{2.53}$$

We will end this section by briefly discussing the detailed balance condition and reversibility.

Definition 28 *A Markov process is said to satisfy the detailed balance condition if*

$$\pi(i)Q(i, j) = \pi(j)Q(j, i) \ , \tag{2.54}$$

where $\pi(i)$, $\pi(j)$ are the steady-state probabilities of states i and j.

A Markov process that satisfies this condition is also denoted as reversible. In other words, when we observe a reversible Markov process, we cannot tell if it is going forward or backward. Similar conditions hold for Markov chains, a Markov chain is said to satisfy the detailed balance condition if

$$\pi(i)P(i, j) = \pi(j)P(j, i) \ . \tag{2.55}$$

2.3.4 Birth-and-Death Processes

Birth-and-death processes (BDP) are a class of infinite space Markov processes. In the following, we present the basic BDP terms and a few essential conclusions that will be used in the remainder of this thesis.

Definition 29 *Let $\{X_n\}$ be a Markov process with state space $\{0, 1, 2, ...\}$ (all non-negative integers). We use $\lambda_n, n = 0, 1, 2, ...$ to denote the birth rates and $\mu_n, n = 0, 1, 2, ...$ to denote the death rates. $\{X_n\}$ is a birth-and-death process if its generator matrix Q has the following form:*

$$Q = \begin{bmatrix} -\lambda_0 & \lambda_0 & & & \\ \mu_1 & -(\mu_1 + \lambda_1) & \lambda_1 & & \\ & \mu_2 & -(\mu_2 + \lambda_2) & \lambda_2 & \\ & & \ddots & \ddots & \ddots \\ & & & & \end{bmatrix} \tag{2.56}$$

In BDP, the state changes will always be from n to $n + 1$, or n to $n - 1$.

Next, we consider the Yule process.

Definition 30 *A BDP with $\mu_n = 0$, and $\lambda_n = n\lambda$, where $\lambda > 0$, is called a Yule process.*

We let $Prob^{(t)}(n) = Prob(X_t = n)$, then, for a Yule process:

$$Prob^{(t)}(n) = e^{-\lambda t}[1 - e^{-\lambda t}]^{n-1}, n \geq 1. \tag{2.57}$$

$$E(X_t) = e^{\lambda t} \ . \tag{2.58}$$

If we use T_i to denote the time between state i and state $i+1$ in a Yule process, then T_i is exponentially distributed with parameter λi (Definition 11).

2.3.5 Poisson Processes

In this Section, we introduce another infinite space Markov process called the Poisson process.

Definition 31 *Let τ_i, $i = 1...n$ be independent exponential random variables with rate λ, $T_n = \sum_{i=1}^{n} \tau_i$ and $T_0 = 0$, we define the Poisson process as $\{X_t\} = max\{n : T_n \leq t\}$.*

The Poisson process is a birth-and-death process with $\mu_n = 0$, and $\lambda_n = \lambda$.

The state space of a Poisson process consists of nonnegative integers. Here τ_i can be considered as the arrival times between events, so T_n is the arrival time of the n_{th} event, and X_t is the number of arrivals by time t. To explain why $\{X_t\}$ is called a Poisson process rather than an exponential process, we note that X_t follows a Poisson distribution with rate λt. If we let $\chi = X_{s+u} - X_s$, then χ also follows a Poisson distribution with rate λu.

We can also show an alternative way of deriving the Poisson distribution using the Poisson process. If $X_{t+\Delta t} = i$, then $X_t = i$ *or* $i-1$ ($i \geq 1$ and the probability that X_t was in some other state is $o(\Delta t)$). Then $Prob(X_{t+\Delta t} = i|X_t = i) = 1 - \lambda\Delta t + o(\Delta t)$, $Prob(X_{t+\Delta t} = i|X_t = i-1) = \lambda\Delta t + o(\Delta t)$, so we have

$$\begin{aligned} Prob(X_{t+\Delta t} = i) = Prob(X_t = i)[1 - \lambda\Delta t + o(\Delta t)] \\ +Prob(X_t = i-1)[\lambda\Delta t + o(\Delta t)] + o(\Delta t) \ , \end{aligned} \tag{2.59}$$

subtracting $Prob(X_t = i)$ from both sides and dividing by Δt:

$$\begin{aligned} Prob'(X_t = i) &= \frac{Prob(X_{t+\Delta t} = i) - Prob(X_t = i)}{\Delta t} \\ &= -\lambda Prob(X_t = i) + \lambda Prob(X_t = i-1) + \frac{o(\Delta t)}{\Delta t} \ . \end{aligned} \tag{2.60}$$

Let $\Delta t \to 0$, then

$$Prob'(X_t = i) + \lambda Prob(X_t = i) = \lambda Prob(X_t = i-1), \ i \geq 1 \ . \tag{2.61}$$

For $i = 0$,

$$Prob(X_{t+\Delta t} = 0) = Prob(X_t = 0)Prob(X_{\Delta t} = 0) \ , \tag{2.62}$$

therefore

$$Prob(X_{t+\Delta t} = 0) = Prob(X_t = 0)[1 - \lambda\Delta t + o(\Delta t)] \ . \tag{2.63}$$

Similar to Equation 2.59 and Equation 2.60

$$Prob'(X_t = 0) = -\lambda Prob(X_t = 0) \ , \tag{2.64}$$

solving Equation 2.61 and Equation 2.64, we get

$$Prob(X_t = 0) = e^{-\lambda t} \tag{2.65}$$

$$Prob(X_t = i) = \frac{e^{-\lambda t}(\lambda t)^i}{i!} \ . \tag{2.66}$$

As a final remark for this section, we have been using t to represent time, but "time" should be considered as an abstract concept, it can also be, for instance a "kilometer" or a "mutation". Thereby we obtain the interpretation of the Poisson distribution, as events "per km" or "per mutation".

2.4 Markov Chain Monte Carlo Method

2.4.1 Monte Carlo Simulation

The idea of a Monte Carlo simulation is to repeatedly draw random samples from a given target probability distribution. These samples can be used to approximate the target density and obtain numerical results. For instance, suppose that we want to calculate the expectation of $h(\theta)$ over the probability density $p(\theta)$

$$E(h(\theta)) = \int h(\theta)p(\theta)d\theta \ . \tag{2.67}$$

We can draw N independent samples θ_i from $p(\theta)$, and then approximate $E(h(\theta))$ as:

$$E(h(\theta)) \approx \frac{1}{N}\sum_{i=1}^{N} h(\theta_i) \ . \tag{2.68}$$

For some simple probability distributions (e.g., the Bernoulli distribution Definition 7), we can use the so-called inverse mapping method for drawing samples. Let F be the cumulative distribution function of p, and let U be a random variable with a continuous uniform distribution between 0 and 1. If we denote the inverse of F by F^{-1}, then the random variable Θ defined by

$$\Theta = F^{-1}(U) \tag{2.69}$$

has a a cumulative distribution function given by F. However, it is not always possible to analytically calculate F^{-1}. There are other sampling techniques such as rejection sampling and importance sampling, but they usually scale badly with dimensionality [5]. In the next Section, we will introduce a sampling algorithm that can generate samples from $p(\theta)$ by exploring the state space of a Markov chain.

2.4.2 The Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm constructs a Markov chain that satisfies the detailed balance condition, such that the steady-state probability distribution $\pi(\theta)$, is the same as the probability density $p(\theta)$ from which we want to sample. The state space of the Markov chain consists of all possible values of θ, and the Metropolis-Hastings algorithm actually defines the transition probabilities between states.

First we need to define the so-called *acceptance ratio*, which is used to determine whether or not a proposal (see below) is accepted:

$$\alpha(i,j) = min\left(1, \frac{\pi(j)q(i|j)}{\pi(i)q(j|i)}\right) , \tag{2.70}$$

where $q(j|i)$ is the *proposal distribution* of the next sampling value j given the current value i; j is thus the proposal given i.

The Metropolis-Hastings algorithm works as follows:

Input: Maximal number of iteration m
Set $t = 0$;
Initialize θ_t with random values;
repeat
  Sample θ' from $q(\theta'|\theta_t)$;
  Draw a random number u between 0 and 1 ;
  if $u \leq \alpha(\theta, \theta')$ **then**
    Set $\theta_{t+1} = \theta'$;
  else
    Set $\theta_{t+1} = \theta_t$;
  end
  $t = t + 1$;
until $t > m$;

Algorithm 1: The Metropolis-Hastings Algorithm

If $q(i|j) = q(j|i)$, that is, the probability of proposal (next sampling value) from i to j is equal to the probability of proposal from j to i, then the above algorithm is called the Metropolis algorithm [184]. if $q(i|j) \neq q(j|i)$, then the so-called *Hastings ratio* $h = q(i|j)/q(j|i)$ is used to correct the acceptance ratio in Equation 2.70 (see [184] for a more detailed introduction).

In Bayesian statistics, we are interested in computing the posterior distribution

$$f(\theta|X) = \frac{f(\theta)f(X|\theta)}{\int f(\theta)f(X|\theta)d\theta} , \tag{2.71}$$

where $f(\theta)$ is the prior distribution of the parameter θ, $f(X|\theta)$ is the probability of obtaining the data X given parameter θ and $\int f(\theta)f(X|\theta)d\theta$ is the normalizing constant. The normalizing constant is often very hard to compute because it involves integrating over θ. However, using the Metropolis-Hastings algorithm, we can avoid the integration. Note that, if we plug Equation 2.71 into Equation 2.70 we obtain:

$$\alpha(\theta_i, \theta_j) = min\left(1, \frac{f(\theta_j)f(X|\theta_j)q(\theta_i|\theta_j)}{f(\theta_i)f(X|\theta_i)q(\theta_j|\theta_i)}\right) , \tag{2.72}$$

and the normalizing constant cancels out.

If we initialize θ_t at random, then the Markov chain will have to run for a number of iterations before it converges to the stationary distribution. Those initial iterations are called the *burn-in period* and should be discarded when analyzing the samples generated by the Markov chain. The Metropolis-Hastings algorithm generates dependent samples from the target distribution, so often we *thin* the chain by sub-sampling every n_{th} iteration to reduce autocorrelations.

2.5 Summary

This chapter introduced several important stochastic processes and their relationships. They form the basis for the novel models and algorithms presented later on. They are also the prerequisites for the mathematical models of molecular evolution we introduce in chapter 3. We focused on the derivation of the basic models and illustrated how they are connected. For some models, we also provided multiple interpretations, to better explain how these models can be applied to biological problems.

CHAPTER 3

Phylogenetic Tree Inference

This Chapter introduces statistical models of evolution and phylogenetic inference. We will see how Markov processes can be used to model DNA sequence changes and how phylogenetic trees can be reconstructed using probabilistic models. At the end of this chapter, we will describe an algorithm that uses a likelihood model to classify unannotated DNA sequences.

3.1 Introduction

DNA is a long polymer consisting of four distinct nucleotides or bases called Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). This long polymer can be represented abstractly as a string from the alphabet {A, T, G, C}. We call such a string a DNA sequence. Mathematically, we can consider {A, T, G, C} to be the sate space (see Definition 14) of a stochastic process. Throughout the thesis, we only discuss DNA sequences, but the models and methods introduced can be used analogously for RNA sequences with {A, U, G, C} as state space, and for protein sequences with 20 amino acids (AA) as state space.

Stochastic process models can be applied to model a single DNA sequence, for instance, using a hidden Markov model [80]. Here the nucleotide at the t_{th} position of the DNA sequence is the state of the Markov chain at time t. But in our application, we assume that each position in the DNA sequence evolves according to an independent stochastic process. That is, if the DNA sequence has a length of n nucleotides, then we model the sequence using n

independent stochastic processes. A single DNA sequence is a snapshot of n independent stochastic processes at time t.

To model n independent stochastic processes through time, we need more than one sequence. DNA sequences frequently undergo substitutions, insertions and deletions in the course of evolution. Therefore, homologous positions of distinct, but related sequences need to be aligned before we can model them with stochastic processes. This naturally gives rise to the sequence alignment problem. Given a scoring matrix, two sequences can be aligned globally with the Needleman–Wunsch algorithm [116], or locally with the Smith-Waterman algorithm [155]. The time and space complexity of these two algorithms is $\mathcal{O}(mn)$, where m and n are the lengths of the two sequences. The dynamic programming approaches of these two algorithms yield optimal solutions.

However, multiple sequence alignment is a more difficulty problem. The goal of multiple sequence alignment is to arrange m sequences into a m by n matrix, where each column of the matrix (or *site*) is derived from a position in an ancestral sequence [43]. Ideally, we would like to use a probabilistic model for the unaligned sequences, the sequence substitution models, the phylogeny and the alignment [133]. The alignment and phylogeny can be estimated simultaneously by searching for solutions that maximize a likelihood function [55, 133]. Such approaches are computationally intractable, except when using simulated annealing and heuristic search algorithms [55], or the Markov Chain Monte Carlo (MCMC) method to sample from the posterior of the model [133]. Due to the complexity of the model and the heavy computational demand, it can only be applied to small data sets [95].

In practice, we often treat multiple sequence alignment and phylogenetic inference as separate problems. For the multiple sequence alignment problem, the most common heuristic algorithm is to construct an alignment that yields an MSA with a "good" sum of pairwise sequence alignment scores (SP) [46]. Optimizing the SP score has been shown to be NP-hard [49]. Thus common heuristics in popular tools use guide trees and progressive alignment techniques [43, 120, 171]. They reduce the MSA problem to a sequence of pairwise alignments. Those approaches take only unaligned sequences as input and try to optimize the alignment with respect to a target function. They are generally referred as *de novo* multiple sequence alignment methods [11].

Although several heuristics are employed to compute the alignment, *de novo* multiple sequence alignment methods can still be computationally prohibitive when dealing with millions of reads obtained from next-generation sequencing (NGS) experiments. Alternatively, we can use information from existing alignments and align new sequences to such a reference alignment.

Several tools were developed for this purpose such as HMMALIGN [41], NAST [37], SINA [127], PaPaRa [11], and PAGAN [98]. These tools are computationally efficient and scale well with NGS data, because new sequences are only compared and aligned to the reference alignment and not aligned with each other.

In the following, we only consider the phylogenetic inference problem and always assume that the multiple sequence alignment is given.

3.2 Nucleotide Substitution Models

3.2.1 The GTR Model

Assume that we have a DNA sequence of length n, and that each position (or *site*) of the sequence evolves through time and is independent as well as identically distributed. We model each site with a Markov process. To define a Markov process, we need the state space, which is {A, C, G, T} for DNA sequences; and a generator matrix Q, which describes the transition rates among the four nucleotides. One of the most commonly used nucleotide substitution model is the GTR (General Time Reversible) model [185]. The Q matrix has the following form:

$$Q = \begin{pmatrix} \cdot & q_{A,C} & q_{A,G} & q_{A,T} \\ q_{C,A} & \cdot & q_{C,G} & q_{C,T} \\ q_{G,A} & q_{G,C} & \cdot & q_{G,T} \\ q_{T,A} & q_{T,C} & q_{T,G} & \cdot \end{pmatrix} = \begin{pmatrix} \cdot & a\pi_C & b\pi_G & c\pi_T \\ a\pi_A & \cdot & d\pi_G & e\pi_T \\ b\pi_A & d\pi_C & \cdot & f\pi_T \\ c\pi_A & e\pi_C & f\pi_G & \cdot \end{pmatrix}, \tag{3.1}$$

and the diagonal elements of Q are determined by each row of Q that needs to sums to 0. The GTR model, as its name suggests, satisfies the detailed balance condition (Definition 28). Let π_i be the steady-state probability of nucleotide i.

To see the symmetrical relationships of parameters a, b, c, d, e, f in Q, note that Equation 3.1 can be decomposed into the product of a symmetric matrix and a diagonal matrix:

$$Q = \begin{pmatrix} \cdot & a & b & c \\ a & \cdot & d & e \\ b & d & \cdot & f \\ c & e & f & \cdot \end{pmatrix} \begin{pmatrix} \pi_A & 0 & 0 & 0 \\ 0 & \pi_C & 0 & 0 \\ 0 & 0 & \pi_G & 0 \\ 0 & 0 & 0 & \pi_T \end{pmatrix}. \tag{3.2}$$

The three steady-state probabilities (because the four steady-state probabilities must sum to 1, thus only three of them are free parameters) and a, b, c, d, e, f are free parameters.

Usually the time unit is unknown, so we use Q to represent the relative transition rates. We can therefore let f always be 1, and multiply the whole matrix by $1/-\sum_i \pi_i Q(i,i)$, such that the average rate

$$\lambda = \sum_{i,j} \pi_i Q(i,j) = 1 \; for \; i \neq j. \tag{3.3}$$

The GTR model has therefore eight free parameters. Many other popular nucleotide substitution models are special cases of GTR with restrictions on the steady-state probabilities and/or a, b, c, d, e, f. These derived models are nested within GTR and have a lower number of free parameters. Table 3.1 provides a list of those models and how their parameters are restricted with respect to GTR.

Table 3.1: GTR family of nucleotide substitution models

Model	π_i	a, b, c, d, e, f	Free parameters
JC69 [82]	$\pi_A = \pi_C = \pi_G = \pi_T$	$a = b = c = d = e = f$	0
K80 [86]	$\pi_A = \pi_C = \pi_G = \pi_T$	$a = b = c = d \neq e = f$	1
F81 [53]	$\pi_A \neq \pi_C \neq \pi_G \neq \pi_T$	$a = b = c = d = e = f$	3
HKY85 [71]	$\pi_A \neq \pi_C \neq \pi_G \neq \pi_T$	$a = b = c = d \neq e = f$	4
GTR [185]	$\pi_A \neq \pi_C \neq \pi_G \neq \pi_T$	$a \neq b \neq c \neq d \neq e \neq f$	8

The most general none time-reversible model of nucleotide substitution is called UNREST [185], and does not impose any constrains on the Q matrix. It has 11 free parameters. However, this model is not frequently used in practice because of computational difficulties.

3.2.2 Sequence Distance and Likelihood Function

Since the Q matrix only models the relative rates of nucleotide substitution, we cannot estimate the divergence time between two sequences without further information. However, we can estimate the distance between two sequences:

Definition 32 *The distance between two sequences is the expected number of nucleotide substitutions per site. If we let λ be the mean substitution rate per site, and t be the divergence time between two sequences, then*

$$d = \lambda t \; . \tag{3.4}$$

In the GTR model, because we always set $\lambda := 1$ (Equation 3.3), d equals t. We can use the maximum likelihood method to estimate sequence distances.

Maximum likelihood methods find the parameter values that maximize the likelihood function.

Definition 33 *The likelihood of a model is the probability of the data D given the model with a parameter set θ. If we then define*

$$L(\theta) = Prob(D|\theta) \tag{3.5}$$

as a function of θ. L is the likelihood function.

The log likelihood of the GTR model for two sequences is

$$L(t, a, b, c, d, e, f, \pi_A, \pi_T, \pi_C, \pi_G) = \sum_{i,j} n_{i,j} log\{\pi_i Prob^{(t)}(i,j)\} \quad where\ i, j \in \{A, T, G, C\}\ , \tag{3.6}$$

where $n_{i,j}$ is the number of site patterns. The nucleotide in the first sequence is indexed by i and in the second sequence by j. For some simple models such as JC69 and K80, the maximal likelihood solution can be found analytically, but under more complex models, numerical optimization is usually needed.

3.2.3 Rate Heterogeneity

The GTR model discussed so far assumes that different sites in the sequence evolve under the same Markov process *and* at the same rate. This is generally unrealistic for real data, because functionally important sites usually change slowly while other sites might accumulate substitutions more rapidly.

A common approach is to assume that the rate follows a Γ distribution (Definition 12) [186]. We set $\alpha := 1/\beta$, such that the mean of the Γ distribution is 1. Then Equation 3.6 becomes:

$$L(t, a, b, c, d, e, \pi_A, \pi_T, \pi_C) = \sum_{i,j} n_{i,j} log\{\int \pi_i Prob^{(tr)}(i,j) g(r) dr\} \quad i, j \in \{A, T, G, C\}\ , \tag{3.7}$$

where $g(r)$ is the probability density function defined in Equation 2.25

3.3 Phylogenetic Trees

3.3.1 Basic Tree Concepts

In the previous Section we introduced nucleotide substitution models and how they can be used to estimate the distance between two sequences. If

we have more than two sequences, we can represent their relationships using a phylogenetic tree. In mathematical terms, a tree is an undirected graph where any two vertices are connected by exactly one edge. A phylogenetic tree is a model of the genealogical relationships among species or genes. In this thesis, we always consider a phylogenetic tree to be a gene tree. We use the term *nodes* for vertices and *branches* for edges. What we observe are the present-day genes and their DNA sequences, which are called *tips*, *external nodes*, or *taxa*. The *inner nodes* of the tree represent the ancestral genes whose DNA sequence is unknown. The branch lengths are defined as in Definition 32. The *degree* of a node is defined as the number of branches connected to it. If an inner node has a degree of more than three or the root has a degree of more than two, then the node is called a *polytomy* or *multifurcation*. A tree with no polytomies is called a *binary* or *bifurcating tree*.

The common ancestor of all taxa is the *root* of the tree. If a root is specified, then the substitution process is considered to start from the root and the tree is called a *rooted tree*. However, the GTR model assumes that time is reversible, that means the substitution processes are the same looking from any directions. So, when using a GTR model without further information, one can only infer an *unrooted tree*. We can, however, use *outgroup sequences* to root the tree. The outgroups are sequences known to be relatively distantly related to other sequences in the tree. In other words, we know that they share a common ancestor with the other neighboring taxa in the tree. Thus we can root the tree on the branch connecting the outgroups to the neighboring taxa (Figure 3.1).

Sometimes the *molecular clock* assumption can be made, meaning that the nucleotide substitution rate is constant over time. Then, the Q matrix represents the absolute rate of nucleotide substitution in unit time, and branch lengths represent how many units of time have passed. A tree constructed under the molecular clock assumption is ultrametric, which means that the distances from the root to any tip are identical. Note that the GTR model we discussed in subsection 3.2.1 does not assume constant nucleotide substitution rate over time, because we do not know the time unit. We will call the trees constructed under the molecular clock assumption *ultrametric trees*, and use *phylogenetic trees* to denote trees not assuming a molecular clock.

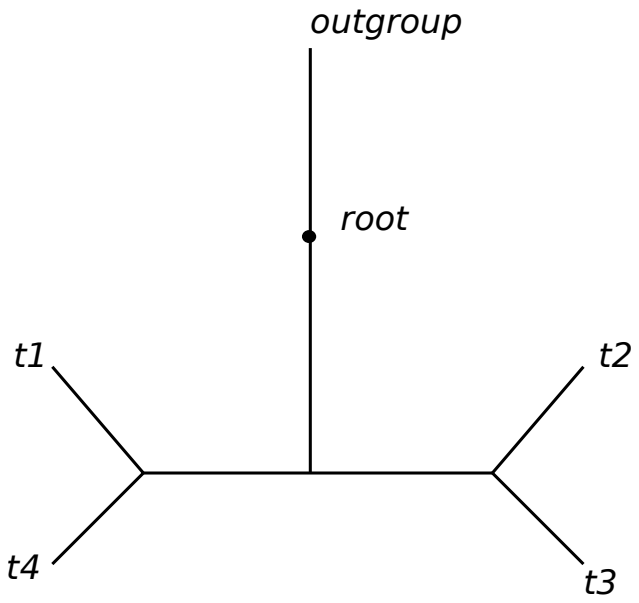

Figure 3.1: The root can be placed on the branch connecting the outgroups to the neighboring taxa.

3.3.2 Phylogenetic Tree Inference Methods and Problem Complexity

There are two main classes of phylogenetic tree inference methods, namely *distance-based* methods and *character-based* methods.

Distance methods comprise two steps: calculation of pair-wise distances between all sequences and reconstruction of a phylogenetic tree from this distance matrix. In the first step, we can use the substitution models described in section 3.2 to infer pair-wise distances between sequences. However, it is generally believed that distances estimated with only two sequences are too inaccurate for phylogenetic inference (see [187] Section 1.6.2). In the second step, a clustering algorithm is employed to reconstruct the tree. The clustering algorithm is nonparametric, hence it does not incorporate a substitution model. Two popular clustering algorithms are UPGMA [156] and Neighbor Joining [142].

The idea of character-based methods for phylogenetic tree inference is to fit the characters (nucleotides) to the tree for every alignment site, using an optimality criterion. Common approaches include the maximum parsimony method that uses the *parsimony score* to evaluate the fit of the tree to the

data, the maximum likelihood method that uses the *likelihood score*, and the Bayesian method which computes the *posterior probability* of the tree. In theory, we can evaluate every possible tree in the tree space and find the tree that maximizes the target function. However, the number of possible tree topologies is large, even for small numbers of taxa. We can compute the number of possible rooted and unrooted bifurcating tree topologies as follows:

$$N_{rooted} = \frac{(2n-5)!}{2^{n-3}(n-3)!} \quad n \geq 3 \tag{3.8}$$

$$N_{unrooted} = \frac{(2n-3)!}{2^{n-2}(n-2)!} \quad n \geq 2\ , \tag{3.9}$$

where n is the number of tips. Table 3.2 shows the number of rooted and unrooted trees with up to 100 tips:

Taxa	Rooted trees	Unrooted trees
3	3	1
4	15	3
5	105	15
6	945	105
7	10,395	945
8	135,135	10,395
9	2,027,025	135,135
10	34,459,425	2,027,025
15	$2.13 * 10^{14}$	$7.90 * 10^{12}$
20	$8.20 * 10^{21}$	$2.21 * 10^{20}$
25	$1.19 * 10^{30}$	$2.53 * 10^{28}$
50	$2.75 * 10^{76}$	$2.83 * 10^{74}$
100	$3.34 * 10^{184}$	$1.70 * 10^{182}$

Table 3.2: Number of possible rooted and unrooted trees with 3–100 taxa.

Thus, in practice, heuristic algorithms must be used for tree searches. These heuristic algorithms can be grouped into two categories. The first category is useful for generating a "good" starting tree. Methods in this category are usually greedy clustering algorithms that add one sequence at a time until all sequences are in the tree. Other methods resolve a star-like tree including all sequences step by step. The second category involves branch swapping, which conducts local or global topological rearrangements of the

comprehensive starting tree. These methods use hill climbing to explore the tree space. Commonly used topological alteration mechanisms include nearest-neighbor interchange (NNI), subtree pruning and regrafting (SPR) and tree bisection and reconnection (TBR). Z.Yang provides a good review of these algorithms in Section 3.2 of his book [187].

3.3.3 Computing the Likelihood of a Tree

In this Section, we look into the details of how Markov process models can be applied to calculate the likelihood of a tree. The likelihood score of the tree is used as an optimality criterion for maximum likelihood tree searches. First we will make some assumptions:

1. A multiple sequence alignment is given, the alignment has s sequences and n alignment sites.
2. A rooted bifurcating tree T relating the s sequences in the alignment, including all branch lengths is given.
3. Different sites evolve independently and follow a Markov process with a rate matrix Q.
4. The Q matrix follows the GTR model.

Under these assumptions, we can calculate the likelihood of the tree in Figure 3.2, for one alignment site l as:

$$\begin{aligned} LH(l) = \pi_{s1} Prob^{(b1)}(s1, s2) Prob^{(b2)}(s2, T) Prob^{(b3)}(s2, s3) Prob^{(b4)}(s3, C) \\ \times Prob^{(b5)}(s3, A) Prob^{(b6)}(s1, s4) Prob^{(b7)}(s4, C) Prob^{(b8)}(s4, G) \ . \end{aligned} \tag{3.10}$$

Here, π_{s1} is the steady-state probability of nucleotide $s1$ (see subsection 2.2.3), and $Prob^{(t)}(i, j)$ can be calculated using Equation 2.51. Equation 3.10 is a direct application of the Markov process model given that the ancestral states $s_i,\ i = 1...4$ are known. However, as mentioned before, $s_i,\ i = 1...4$ are hypothetical ancestral states that cannot be observed. To resolve this problem, we sum over all possible states at the inner nodes:

$$\begin{aligned} LH(l) = & \sum_{s1 \in \{A,T,G,C\}} \sum_{s2 \in \{A,T,G,C\}} \sum_{s3 \in \{A,T,G,C\}} \sum_{s4 \in \{A,T,G,C\}} \pi_{s1} Prob^{(b1)}(s1, s2) \\ & \times Prob^{(b2)}(s2, T) Prob^{(b3)}(s2, s3) Prob^{(b4)}(s3, C) Prob^{(b5)}(s3, A) \\ & \times Prob^{(b6)}(s1, s4) Prob^{(b7)}(s4, C) Prob^{(b8)}(s4, G) \ . \end{aligned} \tag{3.11}$$

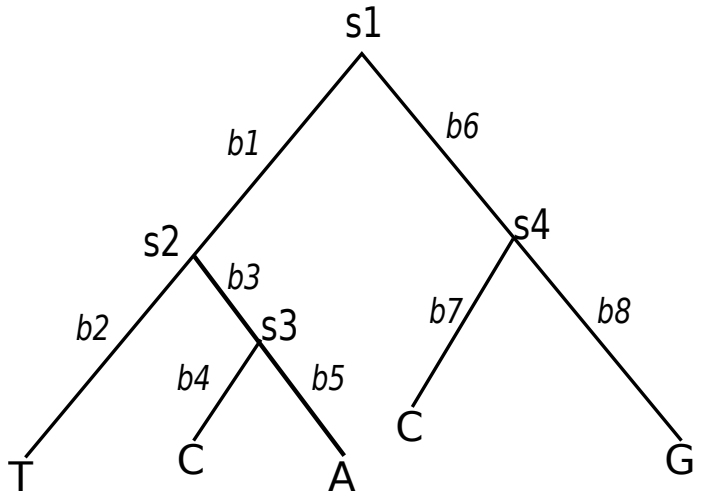

Figure 3.2: A five taxa tree to demonstrate the calculation of the liklihood function.

Calculating Equation 3.11 is very expensive, because there are 4^{s-1} possible combinations of ancestral states for $s-1$ inner nodes. Fortunately, a dynamic programming algorithm, also known as the *pruning algorithm* by J.Felsenstein [53] can be used to calculate the likelihood efficiently by a post-order traversal of the tree. Let $L_i(x_i)$ be the likelihood of the subtree below node i, given that the nucleotide at node i is x_i. Suppose further that the two descendant nodes of node i are node j and node k, then the likelihood function for inner node i can be rewritten as following:

$$L_i(x_i) = \left[\sum_{x_j} Prob^{(b_j)}(x_j, x_i) L_j(x_j)\right] \times \left[\sum_{x_k} Prob^{(b_k)}(x_k, x_i) L_k(x_k)\right] . \tag{3.12}$$

If node i is an external node, then $L_i(x_i) = 1$ if x_i is the observed nucleotide and 0 otherwise. Therefore, the likelihood of the tree for one site is given by the likelihood at the root node r

$$LH(l) = \sum_{x_r} \pi_{x_r} L_r(x_r) . \tag{3.13}$$

Note that, the likelihood calculation described above requires a rooted tree. With the GTR model, a root can be placed at an arbitrary location on any branch of an unrooted tree without changing the likelihood. This is

known as the *pulley principle* in [53] and is guaranteed by the reversibility of the GTR model.

Until now, we have not considered missing data or gaps in the alignment. Using the probabilistic models, we can treat them as unknown data by summing over all possible states. The sum can be simply conducted by letting $L_i(x_i) = 1$ for all x_i if the tip is a gap [54].

We can now also incorporate rate heterogeneity models into the likelihood calculation. In analogy to the two-sequence case described in subsection 3.2.3, we assume that the rate follows a Gamma distribution with probability density function $g(r)$:

$$LH(l) = \int g(r) \sum_{x_r} \pi_{x_r} L_r(x_r, r) dr \ . \tag{3.14}$$

In practice, the continuous Gamma distribution needs to be discretized to avoid the integration, and to be able to compute it numerically. Equation 3.14 then becomes

$$LH(l) = \sum_{j=1}^{k} p(r_k) \sum_{x_r} \pi_{x_r} L_r(x_r, r_k) \ , \tag{3.15}$$

where k is the number of rate classes, and r_k is calculated as a function of the Γ shape parameter α for k equal percentiles, and $p(r_k) := 1/k$.

Finally, because we assume that different sites evolve independently, the likelihood of the tree given the alignment is

$$LH = \prod_{l=1}^{n} LH(l) \ . \tag{3.16}$$

The likelihood LH is usually very small, and may cause numerical issues during computation. Thus, we use its logarithm instead:

$$log(LH) = \sum_{l=1}^{n} log(LH(l)) \ . \tag{3.17}$$

As the logarithm is a monotonically increasing function, the maximum value will be achieved at the same points for both LH and $log(LH)$.

3.4 Phylogenetic Placements

In this section, we introduce the Evolutionary Placement Algorithm (EPA) [10] that uses a likelihood approach to identify unknown sequences.

The EPA assumes that a reference alignment and a fully resolved reference phylogeny based on the reference alignment are given. Initially, all query sequences need to be aligned to the reference alignment. Then, for each query sequence q_i, we execute the placement algorithm described in Algorithm 2.

Input: A bifurcating reference tree T with n tips; full alignment including the reference sequences and the query sequence q_i.
Result: Likelihood scores of q_i being placed into the $2n-3$ branches of T.
foreach *Branch* B *of* T **do**
  Insert q_i into B ;
  Optimize the branch lengths of the new tree T' with $n+1$ tips ;
  Calculate the likelihood score of T';
  Remove q_i from T' ;
end

Algorithm 2: The Evolutionary Placement Algorithm

We can sort the likelihood scores of the $2n-3$ placement branches. Then, the best-scoring insertion branches for q_i on the reference tree can be used to annotate the query sequence q_i.

As a method for sequence identification, EPA has several advantages over sequence-similarity based methods such as BLAST [3]. First, the closest hit found by sequence similarity based methods is often not the closest relative phylogenetically [87]. EPA is phylogeny-aware and is based on the same probabilistic models used for phylogenetic tree inference. It has been shown to perform significantly better than sequence similarity based approaches [10]. Second, EPA provides a higher resolution for the unknown sequences, in the sense that, it can return $n-3$ additional labels, than sequence similarity based methods, which can assign queries to at most n labels.

3.5 Summary

This chapter introduced statistical models of molecular evolution and the basics of phylogenetic tree inference. We showed that, the Markov process models introduced in chapter 2, can directly be applied to calculate the likelihood of a tree. However, searching for the best tree is difficult because of the large tree space. Nonetheless, thanks to the continuous developments of phylogenetic tree inference programs such as RAxML [161] and MrBayes [139],

inferring phylogenetic trees has become a routine procedure in molecular sequence analysis. In the remainder of this thesis, we assume that the phylogenetic trees are given and develop models and algorithms that take phylogenetic trees as input (chapter 4, chapter 5, and chapter 6). Finally, the Evolutionary Placement Algorithm introduced in the last section motivated the development of the pipeline we introduce in chapter 8.

CHAPTER 4

The Poisson Tree Processes Model for Species Delimitation

The content of this Chapter has been partly derived from the following peer-reviewed publication:
J. Zhang, P. Kapli, P. Pavlidis, and A. Stamatakis. A general species delimitation method with applications to phylogenetic placements. *Bioinformatics (Oxford, England)*, 29(22):2869–76, Nov. 2013

Pavlos Pavlidis generated the simulated data sets described in section 4.4, and Paschalia Kapli collected the real data sets described in subsection 4.5.1

Delimiting species is of central importance to many areas of evolutionary biology [152] (see also in chapter 8). Phylogenetic markers can provide important information for species delimitation. This Chapter introduces a new model, called the Poisson Tree Processes (PTP) model, which is designed to propose putative species boundaries on single-locus phylogenetic marker data sets.

4.1 Introduction

To delimit species using molecular sequences, we initially need to define our species concept. However, this turns out to be difficult. Mayden [104] listed

24 different species concepts, and De Queiroz [35] further categorized them into 10 classes. To complicate matters, many species concepts are incompatible with each other and require different types of data (Interbreeding patterns, morphological characters, molecular sequences, etc). Thus, species delimitation should ideally be conducted using an integrative taxonomic framework [121, 146] (see also the discussion in section 4.6). However, for the purposes of this thesis, we will only consider molecular sequence data.

Mathematically, sequence-based species delimitation methods operate on a set of molecular sequences $X = \{x_1, x_2, ..., x_n\}$, which are derived from sequencing certain phylogenetic markers of n individuals. The output of a species delimitation method is a *partition* P of X, which groups X into k species. The partition can be represented as a set of k sets of sequences, such that $P = \{p_1, p_2, .., p_k\}$. P satisfies the following properties:

1. $\emptyset \notin P$;
2. $\bigcup_{p_i \in P} p_i = X$;
3. if $p_i, p_j \in P$ $i \neq j$, then $p_i \cap p_j = \emptyset$.

The above description is identical to sequence clustering. Sequence clustering algorithms, such as UCLUST [44] and CROP [70], rely on predefined sequence similarity thresholds. UCLUST is a fixed threshold clustering approach, while CROP is a soft threshold method that attempts to detect sequence clusters using a Gaussian mixture model. They can cluster sequences into so-called Molecular Operational Taxonomic Units (MOTUs) [57, 67, 175], and therefore they are often called OTU-picking methods. However, such approaches do not deploy any species concept, and it is currently unclear how MOTUs correspond to species [175].

Here, we adopt the Phylogenetic Species Concept (PSC). PSC was initially introduced by [48] and subsequently refined by [8, 32, 34, 119]. For a review of PSC, please refer to [9]. In general, phylogenetic species are the smallest units for which phylogenetic relationships can be reliably inferred. The PSC, in particular, from the genealogical point of view [9], states that species reside at the transition point between evolutionary relationships that are best represented phylogenetically and relationships that are best reflected by reticulating genealogical connections [66].

There already exist several PSC-based species delimitation approaches (e.g., see reviews in [62, 152, 153]). Most of them require multiple gene trees as input except for the General Mixed Yule Coalescent (GMYC) model [61, 124]. However, the GMYC model needs a time-calibrated ul-

trametric input tree, which may represent a major obstacle for applying the method (see section 4.2).

Inspired by the PSC, we introduce the PTP model that can delimit species using non-ultrametric phylogenies. Ultrametricity is not required because we model the speciation rate by directly using the number of substitutions. The PSC implies that phylogenetic reconstruction within a species is inappropriate. A hierarchical relationship can nonetheless be inferred for intra-species sequences using phylogenetic methods. However, we expect to observe significant (in the statistical sense) differences between the relationships reconstructed among and within species. These differences are reflected by branch lengths that represent the mean expected number of substitutions per site between two branching events. Thus, our fundamental assumption is that the number of substitutions between species is significantly higher than the number of substitutions within species. Because it does not require an ultrametric tree, PTP is easier to use than GMYC. As we will show in section 4.5, PTP also outperforms GMYC on simulated data and yields comparable results to GMYC on empirical data.

The remainder of this chapter is organized as follows: First, we review the GMYC model in section 4.2. Then, we describe the PTP model in section 4.3. Subsequently, in section 4.4 and section 4.5, we assess the performance of the GMYC and PTP approach using real and simulated data. We also compare PTP and GMYC to two representative OTU-picking methods UCLUST and CROP. Finally, we conclude in section 4.6.

4.2 The GMYC model

The General Mixed Yule Coalescent (GMYC) model for delimiting species on single-locus phylogenetic markers is frequently used in empirical studies [20, 58, 109, 125, 177]. Several implementations of GMYC are available, including the original R code by T.Fujisawa [61, 124] (available at `http://r-forge.r-project.org/projects/splits`), the Bayesian implementation in R by N.Reid [134] (available at `https://sites.google.com/site/noahmreid/home/software`), and a Python version implemented by myself (available at `https://github.com/zhangjiajie/pGMYC`).

The GMYC method models speciation (among-species branching events) via a pure birth process (subsection 2.3.4) and within-species branching events as neutral coalescent processes. GMYC identifies the transition points between inter- and intra-species branching rates on a time-calibrated ultrametric tree by maximizing the likelihood score of the model. Based on the ultrametric tree, GMYC assumes that all taxa are observed at the present

time and that branch lengths represent waiting times between branching events. The likelihood is computed on n waiting intervals $x_i,\ i = 1..n$ between successive branching events. Assume under the species delimitation assumption τ, there are k species, then the likelihood function for one waiting interval x_i is defined as:

$$L(x_i) = be^{-bx_i} \ , \tag{4.1}$$

where

$$b = \lambda_{spec} n_{i,spec} + \sum_{j=1}^{k} (\lambda_j n_{i,j}(n_{i,j} - 1)) \ . \tag{4.2}$$

The first term in Equation 4.2 comes from the pure birth process with a constant birth rate λ_{spec} and $n_{i,spec}$ lineages in time interval i belongs to the pure birth process. The second term comes from the neutral coalescent model, where

$$\lambda_j = \frac{1}{2N_j} \ , \tag{4.3}$$

where N_j is the effective population size of the coalescent process that belongs to species j, and $n_{i,j}$ is the number of lineages in waiting interval i belonging to coalescent process j. Usually, N_j is assumed to be constant over all species.

Equation 4.1 and Equation 4.2 make strict assumptions on constant branching rates. This can be relaxed by introducing two scaling parameters, such that b can be replaced by

$$b^* = \lambda_{spec} n_{i,spec}^{p_1} + \sum_{j=1}^{k} (\lambda_j (n_{i,j}(n_{i,j} - 1))^{p_2}) \ . \tag{4.4}$$

Finally, the likelihood of species delimitation assumption τ, given the ultrametric tree is given by

$$LH(\tau) = \prod_{i=1}^{n} L(x_i) \ . \tag{4.5}$$

As mentioned above, N_j is usually considered to be constant, therefore, the GMYC model has four free parameters, that is, $\lambda_{spec}, \lambda_j, p_1,\ and\ p_2$.

λ_{spec} can be estimated using its maximum likelihood estimator (the Moran estimator by Nee [115])

$$\widehat{\lambda_{spec}} = \frac{N}{S} \ , \tag{4.6}$$

where N is the number of speciation events (nodes) and S is the sum over all branch lengths belonging to the speciation process. To accommodate the

scaling parameter p_1, Equation 4.6 becomes:

$$\widehat{\lambda_{spec}} = \frac{N}{\sum_{i=1}^{l} n_{i,spec}^{p_1} x_i} \ , \tag{4.7}$$

where $x_i, i = 1..l$ are the waiting intervals for the speciation process. Similarly,

$$\widehat{\lambda_j} = \frac{N}{\sum_{j=1}^{k} \sum_{i=1}^{s} (n_{i,j}(n_{i,j} - 1))^{p_2} x_i} \ , \tag{4.8}$$

where $x_i, i = 1..s$ are the waiting intervals between coalescent events.

The remaining two parameters, p_1 and p_2, are not independent from each other. Thus, they need to be jointly estimated during model fitting. Fujisawa uses the Nelder–Mead method [117] to optimize the two parameters in his R code [61], while I used the L-BFGS-B algorithm [15, 110] from the `SciPy` package.

There are two general strategies to explore the species delimitation assumptions τ_i under the GMYC model. The first one is to infer a single cutoff time C where all nodes above C represent speciation events. The search algorithm therefore only needs to evaluate the likelihood of putting C to each of the internal nodes of the ultrametric tree. We call this the *single-threshold* GMYC model. The second GMYC model allows for *multiple-thresholds* C_i, but at the cost of a much larger search space [109]. The single-threshold GMYC is usually more accurate than the multi-threshold version (see [62] for details).

GMYC has been shown to work well for small population sizes and low birth rates [51]. One drawback of GMYC is that, it depends on the accuracy of the ultrametric input tree. Obtaining an ultrametric tree from a given phylogeny is a compute-intensive *and* potentially error-prone process. Most state-of-the-art likelihood-based tree calibration methods such as BEAST [39] or DPPDIV [73] rely on Bayesian sampling using MCMC (Markov Chain Monte Carlo) methods. The trees from MCMC samples usually contain multifurcations which violate the GMYC model assumptions. One idea is to randomly resolve the multifurcations using zero branch lengths, but the Moran estimator cannot be properly evaluated with zero branch lengths. Another idea is to use a small branch length when resolving multifurcations. However, short branches lead to numerical problems and long branches destroy ultrametricity.

Furthermore, when delimiting species in phylogenetic placements (see chapter 8), which would require calibrating (making ultrametric) thousands of trees, it becomes almost impossible to deploy GMYC in an automated pipeline. For instance, consider the problem to assess MCMC chain convergence in a relaxed molecular clock analysis.

4.3 The Poisson Tree Processes Model

Classic speciation models such as the birth-and-death process assume that new species will emerge and current species will become extinct at certain rates that are measured in unit time [6]. Usually, a time-calibrated tree is required as input. Thus, for molecular sequence data, a molecular clock model must be applied to calibrate the tree. Coalescent theory also relies on unit time to describe the relationships among ancestors and descendants in a population.

Instead, we may consider the number of substitutions between branching and/or speciation events, by modeling speciations using the number of substitutions instead of the time. The underlying assumption is that, each substitution has a small probability of generating a speciation. Note that, the substitutions are independent of each other. If we consider one substitution at a time in discrete steps, the probability of observing η speciations for κ substitutions is given by a binomial distribution (Definition 8). Because we assume that, each substitution has a very small probability ρ of generating a speciation, and the number of substitutions in a population of size η is large, the process follows a Poisson distribution (Definition 10) in continuous time with rate $\rho \times \eta$. Therefore, the number of substitutions until the next speciation event follows an exponential distribution (Definition 11).

Comparing this to the assumptions of a Birth-and-Death Processes (BDP) (subsection 2.3.4) we observe that, each generation (e.g., with a generation time of 20 years) on a time-calibrated ultrametric tree has a small probability of speciation. The BDP does not model substitutions, thus, substitutions are superimposed onto the BDP, whereas PTP explicitly models substitutions. Substitution information can easily be obtained by using the branch lengths of the phylogenetic input tree. Thus, in our model, the underlying assumptions for observing a branching event are consistent with the assumptions made for phylogenetic tree inference.

We can now consider two independent classes of Poisson processes (Definition 31). One process class describes speciation such that, the average number of substitutions until the next speciation event follows an exponential distribution. Given the species tree, we can estimate the rate of speciations per substitution in a straight-forward way. The second Poisson process class describes within-species branching events that are analogous to coalescent events. We assume that, the number of substitutions until the next within-species branching event also follows an exponential distribution. Thus, our model assumes that, the branch lengths of the input tree have been generated by two independent Poisson process classes.

In the following step, we assign/fit the Poisson processes to the tree. Let T be a rooted tree, and $PATH_i$ be a path from the root to leaf i, where $i = 1...l$ and l is the number of leaves. Let b_{ij}, $j = 1 \cdots z$ be the edge lengths of $PATH_i$, representing the number of substitutions. We further assume that b_{ij}, $j = 1 \cdots z$ are independent exponentially distributed random variables with parameter λ. Let $B_{ik} := b_{i1} + ... + b_{ik}$ be the sum over the edge lengths for $k \geq 1$. We further define $N_i(s) := max\{k : B_{ik} \leq s, s \geq 0\}$. B_{ik} is the number of substitutions of the kth branching event, and $N_i(s)$ is the number of branching events below B_{ik}. Note that, $\{N_i(s); s \geq 0\}$ constitutes a Poisson process. Thereby, T and $\{N_i(s); s \geq 0\}$, $i = 1 \cdots l$ together form a tree of Poisson processes which we denote as *Poisson Tree Processes (PTP)*. To a rooted phylogeny with m species, we apply/fit one among-species PTP and at most m within-species PTPs. An example is shown in Figure 4.1

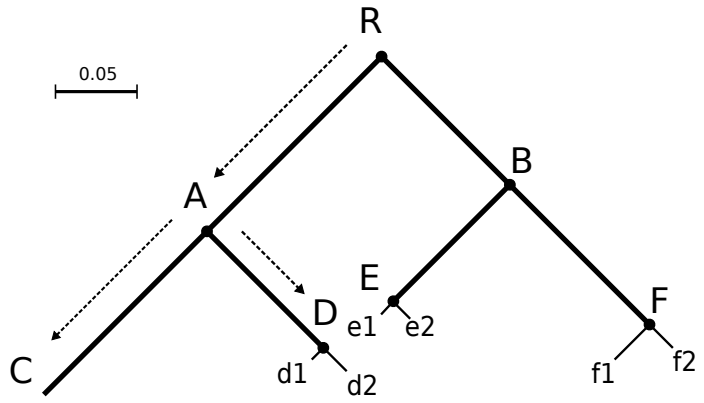

Figure 4.1: Illustration of the *Poisson Tree Processes*. The example tree contains 6 speciation events: *R, A, B, D, E, F*, and 4 species: *C, D, E, F*. Species *C* consists of one individual; species *D, E, F* have 2 individuals each. The thick lines represent among-species *PTP*, and the thin lines represent within-species *PTPs*. The Newick representation of this tree is *((C:0.14, (d1:0.01, d2:0.02)D:0.1)A:0.15, ((e1:0.015, e2:0.014)E:0.1, (f1:0.03, f2:0.02)F:0.12)B:0.11)R;*. The tree has a total of 16 different possible species delimitations. The maximum likelihood search returned the depicted species delimitation with a log-likelihood score of 24.77, and $\lambda_s = 8.33$, $\lambda_c = 55.05$.

In analogy to BP&P [188] and GMYC [124], we conduct a search for the transition points where the branching pattern changes from an among-species to a within-species branching pattern. The total number of possible delimita-

tions on a rooted binary tree with m tips ranges between m (caterpillar tree) and 1.502^m, depending on the actual tree shape [61]. Since the search space is generally too large for an exhaustive search, we need to devise heuristic search strategies. Given a fixed species delimitation, we fit two exponential distributions to the respective two branch length classes (among- and within-species branching events). We calculate the log-likelihood as follows:

$$L = \sum_{i=1}^{k} \log(\lambda_s e^{-\lambda_s x_i}) + \sum_{i=k+1}^{n} \log(\lambda_c e^{-\lambda_c x_i}) \ , \tag{4.9}$$

where x_1 to x_k are the branch lengths generated by among-species PTPs, x_{k+1} to x_n are the branch lengths of within-species PTPs, λ_s is the speciation rate per substitution, and λ_c is the rate of within-species branching events per substitution. The rates λ_s and λ_c can be obtained via the inverse of the average branch lengths that belong to the respective processes. Based on Equation 4.9, we search for the species delimitation that maximizes L. A standard likelihood-ratio test with one degree of freedom can be used to test if there are indeed two classes of branch lengths. Large p-values indicate that either all sequences belongs to the same species or that every sequence represents a single species.

We developed and assessed three heuristic search strategies for finding species delimitations with 'good' likelihood scores. For the experimental results presented here, we used the heuristic that performed best, based on our preliminary experiments.

Heuristic I: We order and store the branch lengths in descending order. We start with the longest branch and add one branch at a time to build consecutive sets that contain branches of among-species branching events. To each set, we add those missing branches that are required to obtain a valid species delimitation configuration, that is, span a tree starting at the root. We then evaluate the likelihood for each extended set. This approach requires $\mathcal{O}(n)$ time, where n is the number of branches in the tree. The rationale for this approach is that longer branches are more likely to form part of speciation events, rather than within-species branching events. An example is shown in Figure 4.2.

Heuristic II: We implement a greedy strategy that starts from the root and includes one child node at a time as speciation event via a breadth-first tree traversal. We then apply this procedure recursively by extending the child node that yielded the higher log likelihood score and re-considering the

other child node. This heuristic has time complexity $\mathcal{O}(n^2)$. The rationale for this approach is that it explicitly uses the tree data structure to explore a larger number of possible delimitations.

Heuristic III: This hybrid approach combines the ideas of the two previous heuristics. First we order the branches as in Heuristic I. Then, we determine the best bisection of this list into a within-species branch set C and among-species branch set S with respect to the likelihood score. This approach ignores the tree structure, but returns an upper bound for the likelihood score. Thereafter, we start with the longest branch again and add one branch at a time to the set S' of speciation event branches. In contrast to Heuristic I, the next branch we add to the set can be any branch in the original set S that is connected to a branch in S' via the tree. When no branch in S is connected to a branch of S' via the tree, we deploy the greedy strategy of Heuristic II to select the next branch we want to add. This approach combines the speed of Heuristic I with the more exhaustive search of Heuristic II.

PTP is implemented in Python and is freely available at `https://github.com/zhangjiajie/ptp`

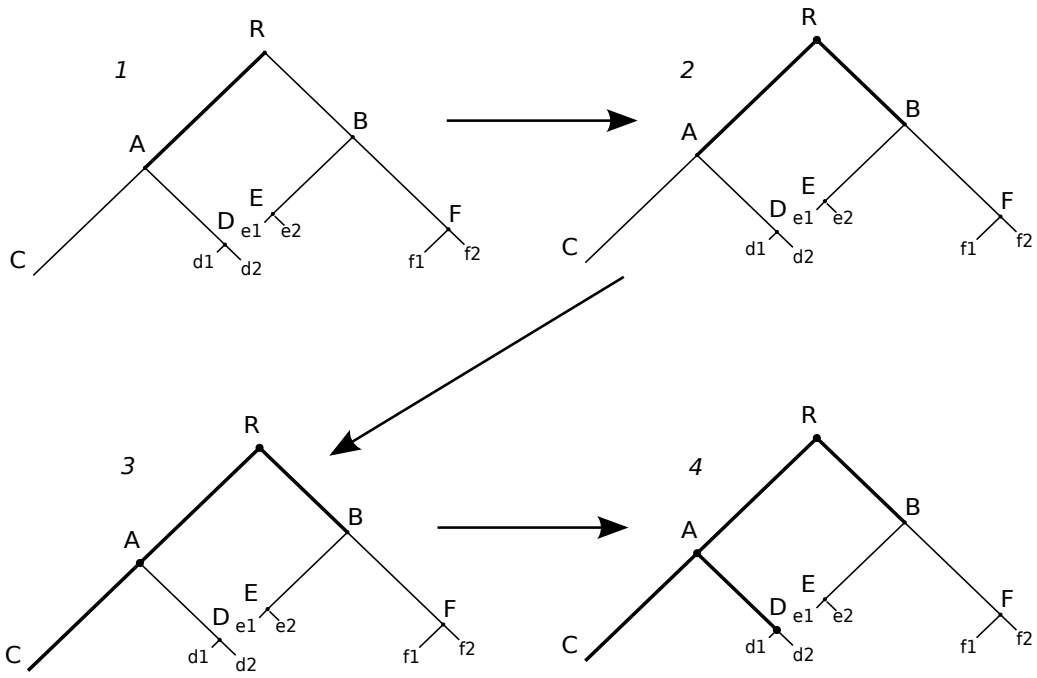

Figure 4.2: In step 1, branch RA (the longest branch) is added to the among-species branches set S; in step 2, RB is added to the same set to make it a valid species delimitation configuration; in step 3, the longest branch AC (from the remaining within-species branches) is added to the among-species branches set S; in step 4, AD is added to the same set to make it a valid species delimitation configuration. The Newick representation of the rooted tree is *((C:0.14, (d1:0.01, d2:0.02)D:0.1)A:0.15, ((e1:0.015, e2:0.014)E:0.1, (f1:0.03, f2:0.02)F:0.12)B:0.11)R;*. The thick lines represent among-species *PTP*, and the thin lines represent within-species *PTPs*.

4.4 Experimental Settings

We tested PTP and compared it to the single-threshold GMYC model on both simulated and real data sets. For simulated data, we used RAxML [160] to infer phylogenetic trees, and then used them as input to PTP. Subsequently, these phylogenies were made *ultrametric* by r8s [144] to test GMYC. For UCLUST and CROP, only molecular sequences are needed as input. In both programs we initially set the sequence dissimilarity threshold to 97%, a widely accepted threshold for bacterial sequences [158]. We also set the sequence dissimilarity threshold for UCLUST to 95% to analyze the effects of changing the dissimilarity threshold. For real data sets, we used the phylogenetic and ultrametric trees from the original publications whenever possible, otherwise we used the same procedures as described above.

4.4.1 Empirical Data Sets

4.4.1.1 Arthropod Datasets

The *Rivancidella* dataset comprises three phylogenetic markers (cyt b, COI, 16S) and was originally used in [124]. The total number of sequences is 472. They represent 24 morphological species and 4 outgroup taxa. The estimated number of putative species for the genus as inferred by GMYC was 48 (with confidence limits ranging between 46 and 52 species). Alternative methods (see 124 for details) used in this study yielded 46 and 47 putative species, respectively.

We also used COI marker datasets [122] of the genera *Dendarus*, *Pimellia*, and *Tentyria*. The datasets comprise 51, 56, and 59 sequences, respectively. The number of species that were attributed to each taxon using morphological criteria was seven, one, and one.

4.4.1.2 Gallotia Dataset

The lizard genus *Gallotia* comprises seven species (based on genetic *and* morphological markers) that are endemic to the Canary islands. The taxonomic species tree and the molecular phylogeny for this data set are fully congruent. The data [31] comprises four mitochondrial phylogenetic markers (cyt b, COI, 12S, 16S) and a total of 90 sequences (76 representing *Gallotia* and 14 outgroup sequences).

4.4.2 Simulations

We generated simulated datasets using a Yule-coalescent model. We used ms [78] and BioPerl [159] in combination with INDELible [56] to simulate sequences. Using a modified version of the BioPerl module `Bio::Phylo` that allows to vary the birth rate value in the simulations, we initially generated a set of random *species* trees $T = T_1, T_2, ..., T_{100}$. The leaves of each tree T_i ($1 <= i <= 100$) represent extant species. All 600 simulated datasets we generated contain 30 species. In the next step, we used ms to generate a structured coalescent gene tree. The node ages of the phylogenetic tree T_i are interpreted as divergence times between populations. In other words, we treat species as diverged populations that were completely isolated from each other after they diverged from their common ancestor. Thus, using ms we simulated a multi-species coalescent gene tree with 30 species and 100 individuals per species. For each species, we randomly selected 10 individuals to generate evenly sampled (in terms of the number of individuals per species) data sets. We also generated unevenly sampled data sets containing 2 species with 100 individuals, 4 with 50 individuals, 8 with 10 individuals, and 16 with 2 individuals. Finally, we employed INDELible to simulate DNA alignments of 250-bp, 500-bp, and 1000-bp on the above multi-species coalescent trees.

Note that, INDELible, ms, and BioPerl use different units for representing branch lengths. INDELible uses the expected number of substitutions (the standard unit in phylogenetics), whereas ms uses the coalescent time unit of $4N$ generations where N is the effective population size. BioPerl only uses the birth rate to generate trees (small birth rates generate longer trees, large birth rates generate shorter trees). We therefore converted all branch length units to the expected number of substitutions. In our simulations, we set $\mu := 10^{-7}$, where μ is the mutation rate per base pair, per individual, and per year. This value for μ is situated approximately in the middle of the empirical value range. For instance, human genomic DNA has a rate of 10^{-8} [113], human mitochondria have a rate of 10^{-5} [151], and viruses have a rate that ranges between 10^{-4} and 10^{-8} [38].

For the birth rate b, we used a value range around 0.5 speciation events per one million years. The value of 0.5 is realistic for several distinct types of species [106]. To convert b into units of speciations per substitution we apply $b' = \frac{b}{\mu \times 10^6}$, where b' is the scaled birth rate per substitution event. Thus, values of b' around 5 can be considered realistic.

With respect to coalescent units, let l be a branch length in coalescent units. For an effective population size of N and a mutation rate μ, the expected number of mutations on a branch is $\frac{l}{4N\mu}$. Thus, to convert the coalescent units into the expected number of substitutions, we need to divide

the branch length by $4N\mu$. Thereby, we implicitly assume that the expected number of mutations is approximately equal to the expected number of substitutions.

The key parameters for delimiting species are the birth rate and the effective population size. High birth rates decrease the evolutionary distance between species. High effective population sizes have a similar effect. This is because the coalescent rate is inversely proportional to the effective population size. When the population size is sufficiently large, coalescent events can occur prior to speciation and lead to incomplete lineage sorting. Thus, the effect of the birth rate on species delimitation accuracy also depends on the effective population size. Hence, the birth rate and the effective population size are not independent from each other. Therefore, we keep the effective population size constant at $N := 50,000$ and investigate the effect of varying the scaled birth rate ($b' := 5, 10, 20, 40, 80, 160$). Intuitively, small b' generate large evolutionary distances between species and vice versa, see Figure 4.3 and Figure 4.4 for two examples.

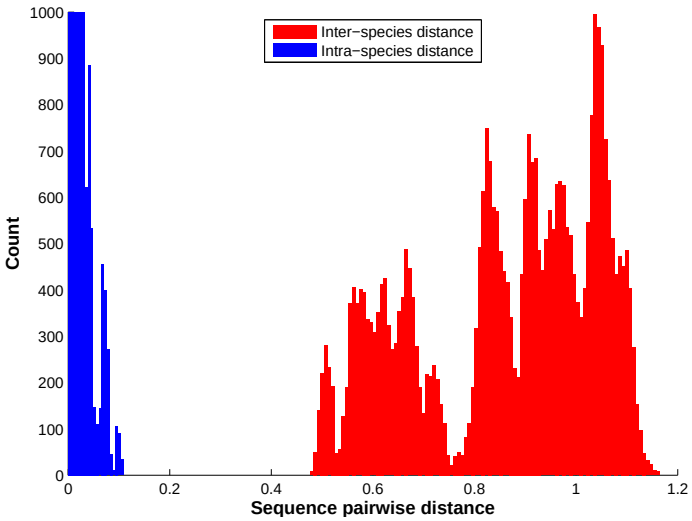

Figure 4.3: Histogram of pairwise sequence distances within and among species ($b' = 5$). A clear gap exists between two types of pairwise distances, sequence similarity based species delimitation approaches will work well for this case.

We used NMI, the normalized mutual information criterion [174] to asses to what degree the delimitation results of the different algorithms agree with

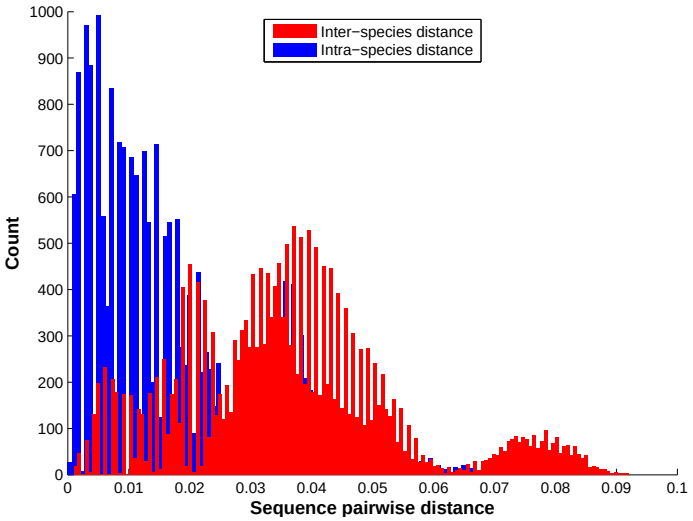

Figure 4.4: Histogram of pairwise sequence distances within and among species ($b' = 160$). The two types of pairwise distances overlap, sequence similarity based species delimitation approaches will not work for this case.

the ground truth. The NMI of two random variables U and V is defined as:

$$NMI(U,V) = \frac{2I(U,V)}{H(U)+H(V)} \ . \tag{4.10}$$

$H(U)$ and $H(V)$ are the entropies of U and V. The entropy is a measure of uncertainty in a random variable. Given a discrete random variable X with n possible values $\{x_1, ...x_n\}$, the entropy is defined as:

$$H(X) = -\sum_{x_i=1}^{n} Prob(x_i) log Prob(x_i) \ . \tag{4.11}$$

$I(U,V)$ is the mutual information (MI) of U and V:

$$I(U,V) = \sum_{u\in U}\sum_{v\in V} Prob(u,v) log\left(\frac{Prob(u,v)}{Prob(u)Prob(v)}\right) \ . \tag{4.12}$$

$I(U,V)$ measures the information shared by U and V. MI is nonnegative and symmetric (i.e. $I(U,V) = I(V,U)$). It is easy to see that if U and V are independent, then $Prob(u,v) = Prob(u)Prob(v)$, and therefore $I(U,V) = 0$.

NMI scales MI to a value between 0.0 and 1.0. In our case, the random variable is the partition p_i of taxa, and $Prob(p_i) = |p_i|/\sum_i p_i$. NMI = 1

means that the delimitation is identical to the ground truth, while NMI = 0 means that the delimited species are randomly partitioned compared to the ground truth.

4.5 Results

4.5.1 Results for Empirical Data Sets

The number of putative species delimited for *Dendarus*, *Pimelia*, *Tentyria*, and *Gallotia* are comparable for all four methods (Table 4.1). For the *Gallotia* data set, GMYC and PTP yield identical results. Three of the *Gallotia* species were split into two separate groups according to the geographical isolation of the corresponding populations on different islands.

On the *Rivacindela* dataset PTP yields a similar, but more conservative delimitation than GMYC. CROP and UCLUST yield dissimilar results, CROP only detects 6 clusters whereas UCLUST detects 82 clusters. It is worth noting that the PTP result presented here for the *Rivacindela* dataset is different from [191]. This is because I used the phylogenetic tree from [124] for the sake of a fair comparison. I also removed the outgroup taxa and upgraded the PTP code to ignore close to zero branch lengths (see the discussions in section 4.6).

4.5.2 Results for Simulated Data Sets

The results on evenly sampled simulated data are summarized in Table 4.2, Table 4.3 and Table 4.4. On average, PTP shows the best performance and outperforms GMYC in all test scenarios. OTU-picking methods work well on data sets with small b' values, that is when the evolutionary distances between species are large. For $b' \leq 20$, UCLUST generally outperforms PTP

Taxon	Morphological	GMYC	PTP	CROP	UCLUST
Rivacindela	24	48	43	6	82
Dendarus	7	10	9	7	11
Pimelia	1	10	9	7	10
Tentyria	1	2	2	1	3
Gallotia	7	10	10	9	15

Table 4.1: Number of species delimitated on real data.

and yields the best overall results. However, with increasing b' the accuracy of OTU-picking methods decreases steeply. As expected, for shorter sequence lengths (250-bp and 500-bp), accuracy deteriorates for all methods and in a more pronounced way for PTP and GMYC. However, even with sequence lengths of 250-bp, PTP still yields the best results on data sets with $b' > 20$.

On the unevenly sampled simulated data sets (Table 4.5, Table 4.6 and Table 4.7), the delimitation accuracy decreases for UCLUST and PTP. CROP and GMYC yield higher NMI scores than on evenly sampled datasets. On average, PTP yields the best results over all (evenly and unevenly sampled) simulated data-sets.

b'	5	10	20	40	80	160	Mean
UCLUST (0.03)	0.969	0.959	0.938	0.892	0.782	0.575	0.852
UCLUST (0.05)	0.971	0.947	0.904	0.798	0.576	0.249	0.741
CROP	0.964	0.930	0.848	0.646	0.232	0.038	0.609
GMYC	0.924	0.914	0.907	0.886	0.834	0.697	0.860
PTP	0.944	0.935	0.922	0.905	0.882	0.857	0.907

Table 4.2: Species delimitation accuracy (measured in NMI) on simulated evenly sampled data with a sequence length of 1000-bp

b'	5	10	20	40	80	160	Mean
UCLUST (0.03)	0.967	0.958	0.935	0.884	0.771	0.554	0.844
UCLUST (0.05)	0.969	0.945	0.897	0.787	0.555	0.269	0.737
CROP	0.964	0.927	0.836	0.613	0.187	0.027	0.592
GMYC	0.918	0.878	0.766	0.583	0.626	0.551	0.720
PTP	0.952	0.938	0.920	0.898	0.864	0.828	0.900

Table 4.3: Species delimitation accuracy (measured in NMI) on simulated evenly sampled data with a sequence length of 500-bp

We simulate the data in accordance with the GMYC model, that essentially adopts the PSC. To demonstrate the impact of the b' parameter on clustering-based delimitation accuracy, we plotted the pairwise sequence distances within species and between directly adjacent species in the simulated tree, for $b' := 5$ and $b' := 160$ in Figure 4.3 and Figure 4.4. Lower b'

b'	5	10	20	40	80	160	Mean
UCLUST (0.03)	0.967	0.954	0.930	0.871	0.735	0.522	0.829
UCLUST (0.05)	0.966	0.939	0.886	0.765	0.514	0.249	0.720
CROP	0.961	0.917	0.800	0.545	0.152	0.024	0.566
GMYC	0.892	0.620	0.484	0.464	0.550	0.503	0.585
PTP	0.946	0.927	0.907	0.881	0.833	0.780	0.879

Table 4.4: Species delimitation accuracy (measured in NMI) on simulated evenly sampled data with a sequence length of 250-bp

b'	5	10	20	40	80	160	Mean
UCLUST (0.03)	0.937	0.936	0.923	0.886	0.789	0.582	0.842
UCLUST (0.05)	0.968	0.957	0.922	0.829	0.607	0.264	0.758
CROP	0.971	0.946	0.892	0.723	0.303	0.047	0.647
GMYC	0.937	0.894	0.849	0.834	0.791	0.725	0.838
PTP	0.921	0.912	0.889	0.866	0.830	0.800	0.892

Table 4.5: Species delimitation accuracy (measured in NMI) on simulated unevenly sampled data with a sequence length of 1000-bp

values lead to larger evolutionary distances between species, that is, the so-called barcoding gap [128] is present. Increasing b' reduces the evolutionary distances between species and the barcoding gap disappears (see [128] for examples of this phenomenon on real data). Therefore, our simulations show that clustering algorithms work on data sets containing a barcoding gap, because phylogenetic species are mostly consistent with sequence clusters in this case. However, clustering methods are prone to fail when the barcoding gap is not present because sequences cannot be told apart any more via sequence similarity alone.

b'	5	10	20	40	80	160	Mean
UCLUST (0.03)	0.936	0.936	0.920	0.882	0.775	0.563	0.835
UCLUST (0.05)	0.966	0.956	0.914	0.824	0.586	0.266	0.752
CROP	0.971	0.945	0.875	0.682	0.232	0.031	0.622
GMYC	0.941	0.901	0.870	0.792	0.658	0.610	0.795
PTP	0.943	0.927	0.904	0.878	0.835	0.784	0.878

Table 4.6: Species delimitation accuracy (measured in NMI) on simulated unevenly sampled data with a sequence length of 500-bp

b'	5	10	20	40	80	160	Mean
UCLUST (0.03)	0.935	0.933	0.913	0.866	0.742	0.514	0.817
UCLUST (0.05)	0.962	0.948	0.902	0.791	0.545	0.269	0.736
CROP	0.970	0.937	0.852	0.616	0.192	0.021	0.598
GMYC	0.925	0.867	0.814	0.732	0.586	0.523	0.741
PTP	0.948	0.924	0.901	0.863	0.812	0.753	0.866

Table 4.7: Species delimitation accuracy (measured in NMI) on simulated unevenly sampled data with a sequence length of 250-bp

4.6 Summary

We introduced, implemented, and made available a new model (PTP) for species-delimitation that is mainly intended for delimiting species in single-locus molecular phylogenies. PTP can propose species boundaries that are consistent with the PSC. An important advantage of our method is that it models speciation in terms of the number of substitutions. Thereby, it circumvents the potentially error-prone and compute-intensive process of generating time-calibrated ultrametric trees which are required as input for GMYC.

Using real data sets, we show that delimitations inferred with PTP are comparable to delimitations obtained via GMYC. Simulations suggest PTP outperforms GMYC.

In addition, it is more straight-forward to use because it only requires a standard phylogenetic tree as input and because it is also substantially faster. On the 673-taxon meta-barcoding dataset (using a modern Intel desktop processor) for instance, r8s requires 3 days to complete while RAxML in

combination with PTP only requires a total of about 20 minutes to return a species delimitation.

Furthermore, the maximum likelihood estimators of λ_s and λ_c for PTP can easily accommodate multifurcations, while GMYC requires strictly bifurcating input trees.

We also compared GMYC and PTP to two clustering algorithms: CROP and UCLUST. From our point of view, the problem of species delimitation needs to incoporate data from various sources (e.g., sequences *and* trees) and also depends heavily on the species definition used. Thus, GMYC and PTP yield comparable results on real data because they are based on the phylogenetic species concept. In contrast, by their very definition, CROP and UCLUST simply identify sequence clusters. They are suited for data sets with the barcoding gap. The fact that there *is* a difference between sequence clusters and PSC-based species delimitation is underpinned by our simulations.

As we show, GMYC and PTP delimitation performance is more robust to the absence of the barcoding gap. Thus, when no prior information (barcoding gap presence) about the data set is available and the goal is to delimit phylogenetic species, GMYC and PTP should be preferred.

In the following, we discuss the current limitations of our approach.

Readers should keep in mind that, entities delimited by PTP are putative species only. The phylogenetic trees inferred on single-locus phylogenetic marker sequences are gene trees rather than species trees, albeit the hierarchical relationships above the species boundaries are expected to be mostly consistent with the species tree. However, the boundaries inferred by PTP are only approximate. Additional data needs to be integrated to further validate the delimitations, such as morphological characters and multi-locus sequence data [50] within an integrative taxonomy framework [121, 146]. The putative species delimited by PTP, can, for instance, be used as initial hypothesis that can be further scrutinized with multi-locus coalescent-based methods such as BP&P [188]. BP&P requires prior knowledge of species boundaries, and it represents a validation method, rather than a delimitation method. Due to its computational complexity, BP&P can currently only handle up to 20 species.

Since PTP initiates the search for the maximum likelihood delimitation at the root of the input phylogeny, the tree has to be correctly rooted to obtain accurate estimates. Also, PTP should be used with caution on datasets where the number of individuals sampled per species is unbalanced and where the over-sampled species exhibit small within-species variation.

In such cases, the inferred phylogeny will comprise both, subtrees (comprising one species and many individuals) with a large number of extremely

short branches, and subtrees (comprising one species but only few individuals) with short, but not extremely short branches. Such unbalanced samples may require the introduction of a third λ parameter class of branches to accommodate (i) over-sampled within-species branches, (ii) within-species branches, and (iii) among-species branches. Otherwise, the species that are not over-sampled can not be delimited properly, that is, each individual is likely to be identified as a separate species. Hence, we either need a criterion for removing over-sampled sequences, or a criterion to decide when and how many additional classes of *Poisson tree processes* (λ parameters) need to be introduced.

However, a major drawback of introducing additional *Poisson tree processes* classes is that, the delimitation search space becomes significantly larger. Hence, finding the maximum likelihood delimitation or a best-known delimitation represents a challenging task. Thus, before extending the number of classes, we feel that more work on the design and performance of heuristic search strategies for species delimitation is required to better characterize and understand the problem. This also applies to the heuristics used in multiple-threshold GMYC, given that the underlying optimization problems are very similar.

CHAPTER 5

A Bayesian Extension of the Poisson Tree Processes Model

UNPUBLISHED

This Chapter introduces the Bayesian extension of the PTP model and the PTP web server. We will first develop the Bayesian PTP model on a single, fixed phylogenetic tree. We illustrate the idea with a simple example that the posterior probabilities can be calculated explicitly. We also show that, the posterior probability of delimitations is strongly correlated with species delimitation accuracy using simulated data. We then extend the Bayesian PTP model to sets of phylogenetic trees as obtained from Bayesian phylogenetic analysis. Finally, we briefly introduce the PTP web server.

5.1 Introduction

In chapter 4, we introduced the PTP model for species delimitation and three heuristic algorithms to search for the best-known maximum likelihood solution. However, the maximum likelihood solution is a point estimate, and it is hard to derive a confidence interval due to the discrete nature of the model. Furthermore, the search is only conducted on a single, fixed phylogenetic tree, which might contain a high degree of uncertainty.

One solution is to use bootstrap trees, where we search for best-known maximum likelihood solutions of the PTP model on multiple phylogenetic trees obtained from the bootstrap replications [162]. These results can be combined or superimposed onto one phylogeny to derive confidence measures for species delimitations (see subsection 5.2.2).

Another solution, which is the main topic of this chapter, is to extend the PTP model using a Bayesian framework.

First, the uncertainty regarding the topologies can be accounted for, by using tree sets obtained via Bayesian phylogenetic inference. We can obtain the posterior distributions of model parameters, tree topology and branch lengths from Bayesian phylogenetic tree inference programs such as MrBayes [139] and ExaBayes [1]. Second, we apply the Bayesian framework to sample the marginal posterior distribution of species delimitations under the PTP model, independent of phylogenetic uncertainties (tree topologies and branch lengths). Finally, via Markov Chain Monte Carlo simulation (MCMC), we can potentially search a substantially larger species delimitation space than the three heuristic algorithms presented before. This may also yield delimitations with better likelihood scores than the heuristic search, if a maximum likelihood solution is desired.

I have implemented both the bootstrap and Bayesian extensions of PTP. They are freely available from `https://github.com/zhangjiajie/ptp`. I have also designed a web interface for the PTP software and the GMYC model. The web server is available at `http://species.h-its.org/`, and the code is available at `https://github.com/zhangjiajie/PTP-web-server`.

5.2 Bayesian Extension

5.2.1 Using a Single Phylogenetic Tree

We first consider a Bayesian PTP model for a single, fixed phylogenetic tree. Let T be a phylogenetic tree with m tips. Let θ_i be the species delimitation hypothesis, $i = 1, 2, ..., \xi$, where ξ is the total number of possible delimitations on T, which ranges between m and 1.502^m depending on the shape of T. Let λ_s and λ_c be the rate parameters for a between-species PTP and within-species PTP model respectively. Usually, we do not have prior information on how taxa should be clustered, so a flat prior $f(\theta)$ is assumed, that is, we consider all species delimitations to be equally likely. We further assume that λ_s and λ_c have a joint prior $f(\lambda_s, \lambda_c)$ and that it is independent of the delimitation prior $f(\theta)$. Then, the posterior probability of a species delimitation hypothesis θ_i is

$$Prob(\theta_i|T) = \frac{\iint f(\lambda_s, \lambda_c) f(\theta_i) f(T|\theta_i, \lambda_s, \lambda_c) d\lambda_s d\lambda_c}{\sum_{\theta_j} \iint f(\lambda_s, \lambda_c) f(\theta_j) f(T|\theta_j, \lambda_s, \lambda_c) d\lambda_s d\lambda_c} \ . \tag{5.1}$$

The likelihood of the model $f(T|\theta_i, \lambda_s, \lambda_c)$ is given by Equation 4.9. For simplicity, λ_s and λ_c are estimated by their maximum likelihood estimators, following [134]. The normalization constant in Equation 5.1 is generally not possible to compute, because it involves summing over all possible species delimitations. Therefore, we adopt the Metropolis-Hastings algorithm (subsection 2.4.2) to sample from the posterior distribution.

Note that, because λ_s and λ_c are estimated using their maximum likelihood estimators, the acceptance ratio simplifies to:

$$\alpha(\theta_i, \theta_j) = min\left(1, \frac{f(\theta_j) f(T|\theta_j, \lambda_s^j, \lambda_c^j) q(\theta_i|\theta_j)}{f(\theta_i) f(T|\theta_i, \lambda_s^i, \lambda_c^i) q(\theta_j|\theta_i)}\right) \ . \tag{5.2}$$

If we assume a flat prior for θ, then $f(\theta_j)/f(\theta_i) = 1$.

Following the approach used in the BP&P software [188], we design the proposal to either join or split a node based on the current delimitation in each step with equal probabilities. Once the join or split decision is made, each node that is eligible for a join or split will be chosen randomly with equal probabilities. A node is eligible for a split if it is the root of the tree, or its parent node has already been split and it is not a leaf node. After a split, the node being split represents a speciation event on the tree. A node is eligible for a join if both of its descendant nodes are either leaf nodes or joined nodes. If a node is joined, then all of its descendant (leaf) nodes belong to one species. An example for a join and split operation on a node is shown in Figure 5.1. We use x to denote the number of eligible nodes for a split and y' is the number of eligible nodes for a join after the split. Analogously, y is the number of eligible nodes for a join and x' denotes the number of eligible nodes for a split after the join operation. The Hastings ratio for a split is thus x/y', and for a join is y/x'.

We summarize the Metropolis-Hastings algorithm for sampling from the posterior distribution of the PTP model in algorithm 3.

Let b be the burn-in iterations, i be the sampling interval, and n be the number of MCMC iterations. The output of algorithm 3 is $N = (n - b)/i$ species delimitation results P_i, that is, N partitions of all taxa. We can compute the posterior probability S_j of a certain group of taxa forming one species, by simply dividing the number of occurrence of those taxa delimited as one species by N. We illustrate the idea with a simple example shown in Figure 5.2, where the the posterior probability can be calculated explicitly.

Data: A rooted phylogenetic tree T; burn-in iterations b; sampling interval i; number of MCMC iterations n
Result: A list of species delimitations sampled from the posterior distribution.
Set $t = 0$ and initialize delimitation configuration P with one of the three Heuristics or at random;
repeat
  Draw a random number u uniformly between 0 and 1;
  if $u \leq 0.5$ **then** `/* Decide to join */`
    `/* Compute PTP log likelihood with Equation 4.9 */`
    $L \leftarrow$ `ComputePTPLogLikelihood(`P`)`;
    $X \leftarrow$ `FindNodesCanJoin(`P`)`;
    Randomly choose one node η from X;
    Join node η and store the new delimitation configuration in P';
    $L' \leftarrow$ `ComputePTPLogLikelihood(`P'`)`;
    $Y \leftarrow$ `FindNodesCanSplit(`P'`)`;
    `/* Log-likelihood needs to be converted back */`
    `/* to likelihood */`
    $\alpha \leftarrow min\left(1, \frac{|X|}{|Y|} \times e^{L'-L}\right)$
  else `/* Decide to split */`
    $L \leftarrow$ `ComputePTPLogLikelihood(`P`)`;
    $X \leftarrow$ `FindNodesCanSplit(`P`)`;
    Randomly choose one node η from X;
    Split node η and store the new delimitation configuration to P';
    $L' \leftarrow$ `ComputePTPLogLikelihood(`P'`)`;
    $Y \leftarrow$ `FindNodesCanJoin(`P'`)`;
    $\alpha \leftarrow min\left(1, \frac{|X|}{|Y|} \times e^{L'-L}\right)$
  end
  Draw a uniform random number u between 0 and 1;
  if $u \leq \alpha$ **then**
    $P \leftarrow P'$
  end
  if $t > b$ *and* $t\%i == 0$ **then**
    Save P
  end
  $t = t + 1$;
until $t > n$;

Algorithm 3: The Metropolis-Hastings Algorithm for Bayesian PTP

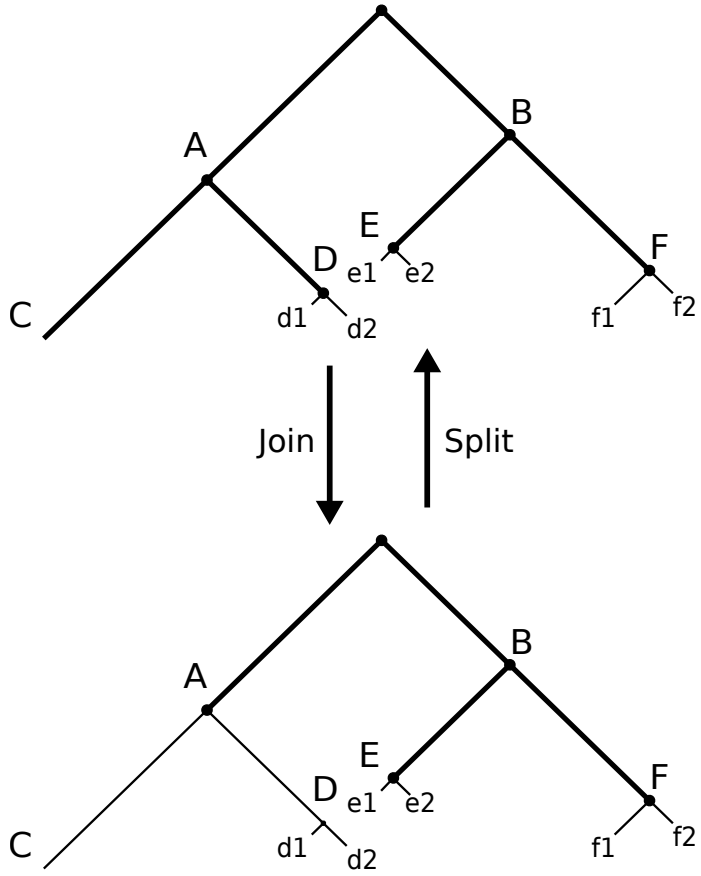

Figure 5.1: Illustration of join and split: join and split Node A. $y/x' = 2/3$, and $y/x' = 3/2$. The thick lines represent among-species *PTP*, and the thin lines represent within-species *PTPs*.

There are five possible species delimitations for this example: P_1, P_2, P_3, P_4 *and* P_5 (Figure 5.3). We use $LH(X)$ to denote the likelihood of delimitation X, then $LH(P_1) = 161850$, $LH(P_2) = 567913$, $LH(P_3) = 264116$, $LH(P_4) = 1346416$ and $LH(P_5) = 161850$. The normalization constant can be computed as

$$C = \sum_{\theta} f(\theta) f(T|\theta) = \sum_{i} LH(P_i) = 2502146 \ . \tag{5.3}$$

To compute the posterior probability of a set of taxa A form one species, we consider every case where all taxa in A are grouped together, for example, $A = \{e1, e2\}$ appears in P_2 and P_4, so the posterior probability of $e1$ and

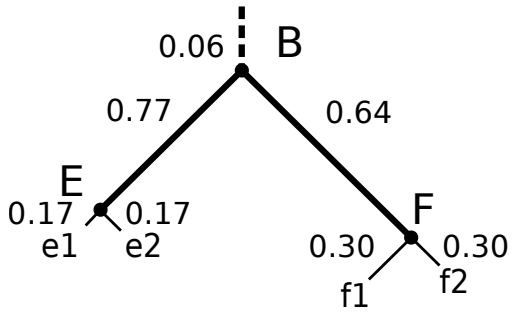

Figure 5.2: A simple example to illustrate the posterior probability calculation. The number on the branch is the posteriori probability that all its decedent taxa form one species. The posteriori probabilities are approximated by MCMC simulations, and they are superimposed onto the maximum likelihood delimitation. Number of MCMC iterations = 500,000, sampling interval = 100, burn-in = 50,000. The Newick representation of the example tree is *((e1:0.015, e2:0.014)E:0.1, (f1:0.03, f2:0.02)F:0.12)B;*.

$e2$ form one species is $\rho = LH(P_2) + LH(P_4)/C$. We assign ρ to node E, in order to denote the posterior probability of all descendant taxa of node E forming one species. We compare the posterior probabilities computed analytically and with the MCMC approximation in Table 5.1.

The posterior probabilities can be considered as support values to reflect our confidence on the species delimited. To determine the relationships between posterior probability and species delimitation accuracy, we compare the mean Bayesian support value (see below) and the NMI values (Equation 4.10) using the simulated data from subsection 4.4.2.

Similar to section 4.4, we infer the phylogenetic tree using RAxML [160], and search for the maximum likelihood delimitations. Then, the posterior probabilities are superimposed onto the maximum likelihood species delimitations. We define the mean Bayesian support value as:

$$\frac{\sum_{i=1}^{m} S_i}{m} \ , \tag{5.4}$$

where S_i is the posterior probability of the *i-th* species delimited in the maximum likelihood solution, and m is the number of species. The NMI values are calculated between the maximum likelihood solution and the ground truth. Finally, we plot the mean Bayesian support values and the NMI values in Figure 5.4. The mean Bayesian support values and the NMI values are highly correlated ($r = 0.91$), suggesting that species delimitations with

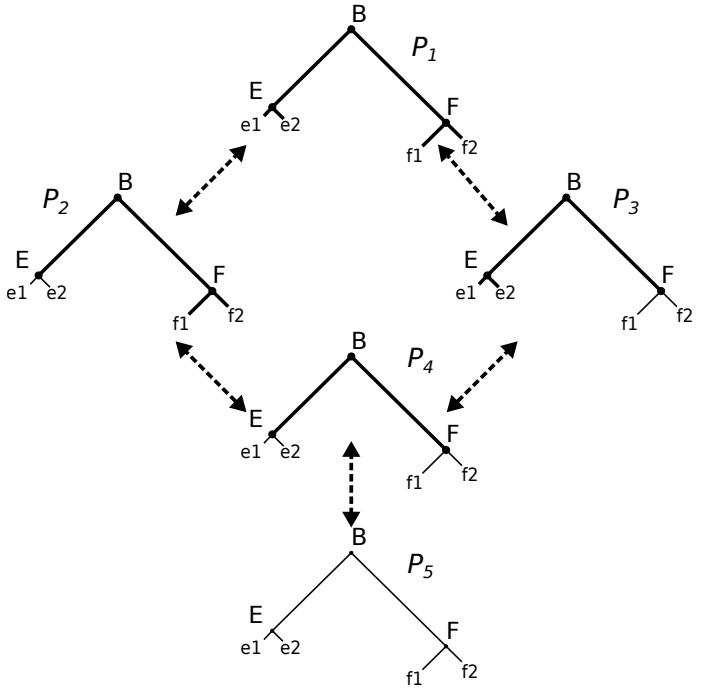

Figure 5.3: Five possible species delimitations of the tree in Figure 5.2, The thick lines represent among-species PTP, and the thin lines represent within-species PTPs.

higher posteriori probabilities are more likely to be correct.

5.2.2 Using Multiple Phylogenetic Trees

In this section, we extend the Bayesian PTP model for analyzing sets of phylogenetic trees as sampled from the posterior distribution of trees that are obtained by Bayesian phylogenetic inference methods.

Let D be the sequence alignment. Let θ_i be the species delimitation hypothesis, and λ_s, λ_c be the two rate parameters. Let t be phylogenetic trees with branch lengths. We can now define the posterior probability as:

$$\begin{aligned} f(\theta_i|D) &= \iiint f(\theta, \lambda_s, \lambda_c, t|D) d\lambda_s d\lambda_c dt \\ &= \frac{\iiint f(\theta_i) f(\lambda_s, \lambda_c) f(t) f(t|\theta_i, \lambda_s, \lambda_c) f(D|t) d\lambda_s d\lambda_c dt}{\sum_{\theta_j} \iiint f(\theta_j) f(\lambda_s, \lambda_c) f(t) f(t|\theta_j, \lambda_s, \lambda_c) f(D|t) d\lambda_s d\lambda_c dt} . \end{aligned} \tag{5.5}$$

Node	Species members	Posterior Probability	MCMC approximation (No. iterations)	
			50000	500000
B	e1, e2, f1, f2	$\frac{LH(P_1)}{C} = 0.064$	0.058	0.060
E	e1, e2	$\frac{LH(P_2)+LH(P_4)}{C} = 0.765$	0.758	0.771
F	f1, f2	$\frac{LH(P_3)+LH(P_4)}{C} = 0.644$	0.661	0.640
e1	e1	$\frac{LH(P_3)+LH(P_5)}{C} = 0.170$	0.184	0.169
e2	e2	$\frac{LH(P_3)+LH(P_5)}{C} = 0.170$	0.184	0.169
f1	f1	$\frac{LH(P_2)+LH(P_5)}{C} = 0.292$	0.282	0.300
f2	f2	$\frac{LH(P_2)+LH(P_5)}{C} = 0.292$	0.282	0.300

Table 5.1: Comparison of analytic solution and MCMC approximation. Number of MCMC iterations = 50000 and 500000, sampling interval = 100, burn-in = 10%.

Because

$$f(t|D) = \frac{f(t)f(D|t)}{\int f(t)f(D|t)} \ , \tag{5.6}$$

Equation 5.5 becomes

$$f(\theta_i|D) = \frac{\iiint f(\theta_i)f(\lambda_s,\lambda_c)f(t|\theta_i,\lambda_s,\lambda_c)f(t|D)d\lambda_s d\lambda_c dt}{\sum_{\theta_j}\iiint f(\theta_j)f(\lambda_s,\lambda_c)f(t|\theta_j,\lambda_s,\lambda_c)f(t|D)d\lambda_s d\lambda_c dt} \ . \tag{5.7}$$

$f(t|D)$ is the posterior distribution of the phylogenetic trees and can be approximated with samples from the MCMC simulations. Assume we have k trees sampled from the posterior distribution, then we can replace the integration over t in Equation 5.7 with summation over the k trees:

$$\begin{aligned} f(\theta_i|D) &= \frac{\sum_{i=1}^{k}\iint f(\theta_i)f(\lambda_s,\lambda_c)f(t_i|\theta_i,\lambda_s,\lambda_c)f(t_i|D)d\lambda_s d\lambda_c}{\sum_{\theta_j}\sum_{i=1}^{k}\iint f(\theta_j)f(\lambda_s,\lambda_c)f(t_i|\theta_j,\lambda_s,\lambda_c)f(t_i|D)d\lambda_s d\lambda_c} \\ &= \frac{\sum_{i=1}^{k}\iint f(\theta_i)f(\lambda_s,\lambda_c)f(t_i|\theta_i,\lambda_s,\lambda_c)d\lambda_s d\lambda_c}{\sum_{\theta_j}\sum_{i=1}^{k}\iint f(\theta_j)f(\lambda_s,\lambda_c)f(t_i|\theta_j,\lambda_s,\lambda_c)d\lambda_s d\lambda_c} \ . \end{aligned} \tag{5.8}$$

Equation 5.8 indicates that we can run k independent MCMC simulations on the k trees obtained from Bayesian phylogenetic analysis programs. If

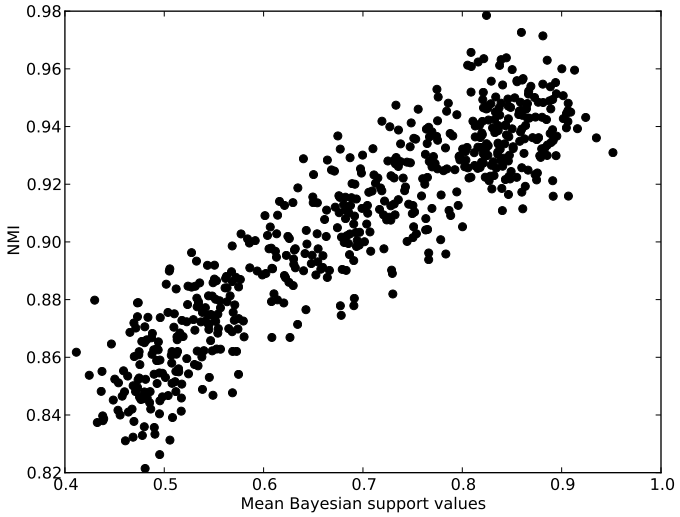

Figure 5.4: Correlation between mean Bayesian support values and delimitation accuracies. Pearson's correlation coefficient $r = 0.91$.

we keep N samples from each MCMC simulation, there will be kN species delimitations, or partitions P_i, where $i = 1...kN$ (see section 4.1) that are sampled from the posterior distribution.

The goal of species delimitation is to propose one single "best" partition. Thus, we need to summarize these kN partitions via a single partition. This is often called the clustering ensemble problem [163]. Suppose we are given r partitions of the same data set, denoted as $P_1, P_2, ..., P_r$, where $r := kN$. The cluster ensemble problem defines a consensus function

$$\Gamma : \{P_i, i = 1...r\} \rightarrow P \ . \tag{5.9}$$

Γ can be, for instance, a function to maximize the average NMI, then

$$P = \arg\max_{\hat{P}} \sum_{i=1}^{r} \mathrm{NMI}(\hat{P}, P_i) \ . \tag{5.10}$$

Solving the objective function in Equation 5.10 is a difficult combinatorial optimization problem. Because there are $\frac{1}{k}\sum_{l=1}^{k}\binom{k}{l}(-1)^{k-l}l^n$ possible partitions for n objects and k given partitions, conducting an exhaustive search for even on small data sets is not feasible.

A greedy heuristic algorithm was proposed by [163], however, evaluating the NMI function between two partitions requires $\mathcal{O}(mn)$ time, where m

and n are the number of clusters in the two partitions. Thus, computing the average NMI has a time complexity of $\mathcal{O}(mnr)$. Therefore, even the greedy heuristic algorithm is disappointingly slow. There exists a number of alternative consensus functions, a review is provided in [112].

Here, for practical reasons, we use a simple approach by choosing one of the kN partitions sampled with the objective function:

$$f(P_i) = \sum_{j=1}^{kN} p_{i,j},\ i = 1 \dots kN\ , \tag{5.11}$$

where $p_{i,j}$ is the posteriori probability of the j_{th} species in partition P_i. Although this approach does not really combine partitions, it only has a time complexity of $\mathcal{O}(kN)$ and has the added benefit that the species delimitation is monophyletic, at least with respect to one phylogeny. Therefore, the delimitation results can be easily plotted onto one phylogenetic tree.

5.3 PTP Web Server

A web server that provides a user-friendly graphical interface for both the maximum likelihood and Bayesian (single, fixed phylogenetic tree) versions of the PTP model is freely available at `http://species.h-its.org/`. The server has been developed with the standard Django Python web framework (`https://www.djangoproject.com/`).

The web server accepts a single phylogenetic tree in Newick or NEXUS format as input. The input tree can be rooted or unrooted. If it is unrooted, the tree is rooted with the outgroup taxa specified by the user. If the input tree is unrooted and no outgroup taxa has been specified, the server roots the tree on the longest branch. I use this rooting heuristic because if outgroup taxa exist in the tree, they are usually associated with long branches. The user can specify all MCMC parameters, and the server allows up to 500,000 MCMC iterations.

The results can be retrieved via the job id and user e-mail address. Users can download the output files in the results page including

- Samples of species delimitations from MCMC simulations.
- Likelihood trace for samples from MCMC simulations.
- Posterior probabilities of all delimited species.
- Maximum likelihood species delimitation.

- Species delimitation with highest Bayesian support (see Equation 5.11).

The maximum likelihood species delimitation can also be visualized via PhyloMap (chapter 6).

In addition to the PTP web server, I also created a web interface for the original R implementation of the GMYC model (see section 4.2), which was implemented by Tomochika Fujisawa and Tim Barraclough [61, 124]. The R code requires a strictly bifurcating ultrametric tree as input. Users can choose to use the single- or multi-threshold version of GMYC, and the output is summarized graphically. All of the above services are running on a dedicated machine with 48 AMD cores and 256GB of memory. The server has received over 5000 submissions at the time when this thesis was written.

5.4 Summary

This Chapter introduced the Bayesian extension of the PTP model, and the MCMC method to draw species delimitation samples from the posterior distribution. One important result from the Bayesian PTP analysis is the posterior probability of the species delimitation. We have shown that the posterior probability is highly correlated with species delimitation accuracy. The Bayesian PTP model can also account for phylogenetic uncertainty when applied to multiple trees sampled by Bayesian phylogenetic analysis. However, summarizing multiple species delimitations is a difficult task and requires further research. Finally, we presented a web server for both, the PTP and the GMYC model.

CHAPTER 6

Visualizing Large Sequence Data Sets

The content of this Chapter has been partly derived from the following peer-reviewed publication:
J. Zhang, A. M. Mamlouk, T. Martinetz, S. Chang, J. Wang, and R. Hilgenfeld. PhyloMap: an algorithm for visualizing relationships of large sequence data sets and its application to the influenza A virus genome. *BMC Bioinformatics*, 12:248, Jan. 2011

The content of this Chapter represents a substantial improvement to the original PhyloMap version, and other co-authors' contribution to the above publication is not included in this Chapter.

This Chapter introduces a new algorithm, PhyloMap, for visualizing large sequence data sets. PhyloMap combines ordination, species delimitation, and phylogenetic tree inference to generate a visual representation of a large sequence data sets.

6.1 Introduction

Phylogenetic trees are commonly used to visualize the evolutionary relationships among homologous sequences [126]. When the number of sequences is

small, the relationships can be easily extracted from the tree. However, when a large number of sequences are analyzed, it becomes increasingly difficult to study the trees and detect patterns [123]. A common approach is to build a reduced tree by sub-sampling a small amount of data rather than constructing a comprehensive tree using the entire dataset [23, 63, 96, 154]. However, the sub-sampling is usually performed according to the intuition of the researcher and is thus not objective nor reproducible. Hence, the conclusions drawn from such trees may be biased.

According to the phylogenetic species concept (see section 4.1), phylogenetic relationships can only be reliably inferred among species. Thus, the sub-sampling should ideally include one and only one representative sequence from each species in the data set. If species memberships are unknown, species delimitation methods, such as PTP and GMYC (see chapter 4) can be applied to identify the putative species boundaries, and thereby reduce the number of sequences. However, there exist uncertainties in the species delimitation processes (see discussion in section 4.6), and this sub-sampling method may lead to loss of information.

Higgins used Principal Coordinate Analysis (PCoA) [74] to visualize large sequence data sets, which are difficult to visualize using phylogenetic trees. He showed that, PCoA can be considered complementary to phylogenetic tree analysis as it does not assume an underlying hierarchical structure in the data. PCoA has also been widely used to aid delimiting species [64, 81, 85, 165, 166]. Ordination (i.e., displaying a set of data points in two or three dimensions so as to make the relationships among the points in higher dimensional space visible) is a powerful tool to visualize large datasets with high dimensionalities. Nevertheless, it only preserves the main trends in the data and detailed information gets lost. When the intrinsic dimensions of the data set are high, the results can sometimes be misleading.

Here, we present a new method - Phylogenetic Map (PhyloMap) - that combines PCoA, species delimitation, and phylogenetic tree inference to generate an easy-to-interpret visualization of a large sequence data sets using all the data. At the same time, it tries to capture detailed relationships. PhyloMap first applies PCoA to identify the main trends. Then, PhyloMap uses PTP to delimit putative species and extract a species-level phylogeny. Finally, PhyloMap maps the phylogenetic tree onto the PCoA result by preserving the tree topology and the branch lengths. As the two different data sources are superimposed, the resulting plot can help to reduce the risk of misinterpretation.

6.2 The PhyloMap Algorithm

The input to PhyloMap is a comprehensive phylogenetic tree T comprising all sequences S. First, a patristic distance matrix D of the distances between all pairs of sequence is computed from T. The distance d_{ij} between sequence s_i and s_j is the sum of all branch lengths along the path connecting s_i and s_j in T. D serves as the input to PCoA to compute the principal coordinates of each sequence. Then, we use PTP to delimit species on T. The species delimitation result is a partition of S, which clusters S into q disjoint, and mutually exclusive clusters (see section 4.1). Then, we choose one sequence r_i randomly from each delimited species as a representative. Subsequently, we prune T down to T' by only preserving r_i, $i = 1...q$. Finally, we adopt a multidimensional scaling technique similar to "Sammon's mapping" [143] to map T' onto the first two axes of the principal coordinates. The results can then be plotted for inspection. The steps of the PhyloMap algorithm are summarized in Figure 6.1.

PTP is implemented in Python, and an interactive GUI (Graphical User Interface) for visualization is implemented in Processing. The Processing programming language is based on JAVA, thus the PhyloMap Processing GUI can be used under most operating systems. It is freely available at `https://github.com/zhangjiajie/PhyloMap`. PhyloMap is also integrated with the PTP web server (section 5.3) to visualize species delimitation results.

The three most important steps in the PhyloMap algorithm are (i) species delimitation, (ii) Principal Coordinate Analysis, and (iii) mapping the species level phylogenetic tree onto PCoA results. Species delimitation has been described in chapter 4 and chapter 5. In the following, we will describe the Principal Coordinate Analysis, and the mapping algorithm.

6.2.1 Principal Coordinate Analysis

PCoA was first described by Gower [68]. It begins by converting the $n * n$ distance matrix D, with elements d_{ij}, to a similarity matrix E with elements:

$$e_{ij} = -\frac{1}{2} d_{ij}^2 \ . \tag{6.1}$$

E is then centralized to obtain a matrix F with elements:

$$f_{ij} = e_{ij} - \overline{e_i} - \overline{e_j} + \overline{e} \ , \tag{6.2}$$

where $\overline{e_i}$ is the mean of row i, $\overline{e_j}$ is the mean of column j, and $\overline{e}$ is the grand mean of matrix E.

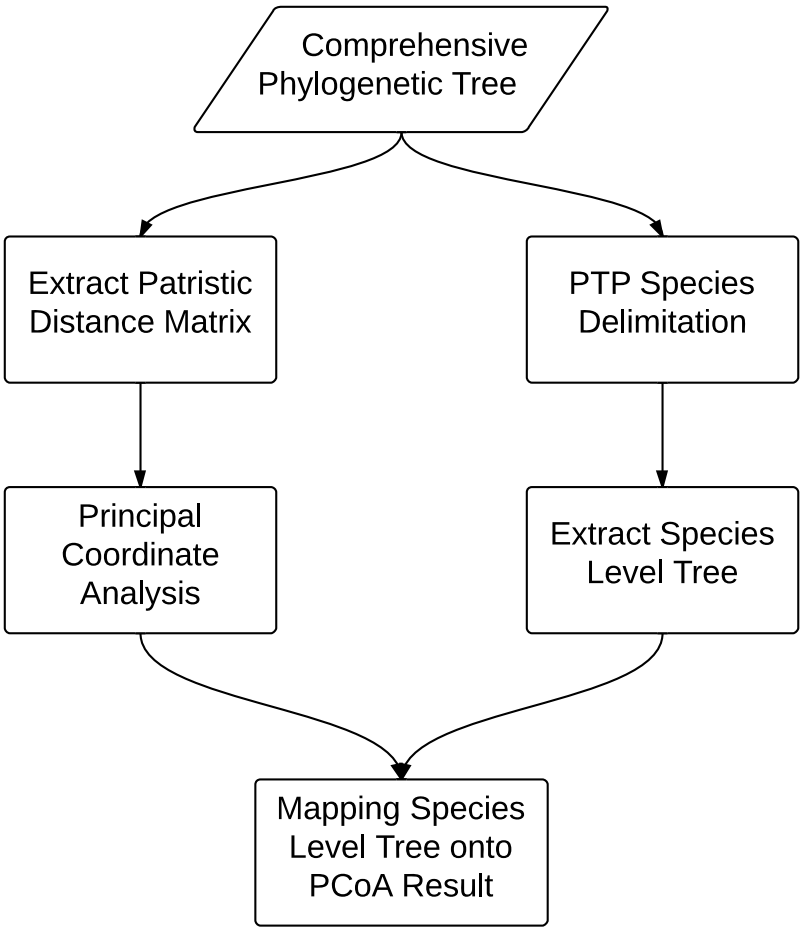

Figure 6.1: Flow chart of the PhyloMap algorithm.

Then, the eigenvectors and eigenvalues of matrix F are calculated. Each eigenvector is normalized so that its sum of squares is equal to the corresponding eigenvalue. Thereafter, the eigenvectors are ranked according to the eigenvalues in decreasing order. The first two eigenvectors are used as two-dimensional coordinates for each sequence. The information (variation) preserved by the first two eigenvectors is the ratio of the sum of the first two eigenvalues and the sum of all eigenvalues.

6.2.2 The Mapping Algorithm

The core algorithm of PhyloMap consists in mapping of the induced species level phylogenetic tree T' onto the two-dimensional coordinates calculated

by PCoA. We adopted a multidimensional scaling method (MDS) similar to "Sammon's mapping" [143], but a few changes have been made to adapt it to our specific problem.

A rooted phylogenetic tree has two types of nodes:

- Leaf nodes: nodes that do not have any children.
- Inner nodes: nodes that have child nodes and a parent node. The root node of the tree can be considered as a special inner node that does not have a parent node.

Each leaf node corresponds to one point in the two-dimensional PCoA result. The positions of these points are fixed, which means that, the coordinates of the leaf nodes are predefined and cannot be changed when drawing the tree. To preserve the branch lengths between nodes, we only need to move the inner nodes.

We first define an error function E:

$$E = \frac{1}{C} \sum_{i<j}^{N} \frac{(\delta^*_{ij} - \delta_{ij})^2}{\delta^*_{ij}} \ , \tag{6.3}$$

where C is the sum over all branch lengths of T', and δ^*_{ij} is the branch length of $branch_{ij}$ between two connecting nodes i and j in the tree, and δ_{ij} is the straight line distance between node i and node j in the 2D PCoA plot. If we denote $y_{ik,k=1,2}$ as the coordinates of node i, then

$$\delta_{ij} = \sqrt{\sum_{k=1}^{2} (y_{ik} - y_{jk})^2} \ . \tag{6.4}$$

The algorithm employs a gradient descent method on the inner node coordinates to minimize E. We denote $E^{(m)}$ and $y^{(m)}_{ik}$ as the mapping error and inner node i' coordinates after m_{th} iteration of the gradient descent procedure, respectively. The coordinate of inner node i in step $m+1$ is given by:

$$y^{(m+1)}_{ik} = y^{(m)}_{ik} - (MF) \frac{\partial E^{(m)}}{\partial y^{(m)}_{ik}} \Bigg/ \left| \frac{\partial^2 E^{(m)}}{\partial y^{(m)^2}_{ik}} \right| \ , \tag{6.5}$$

where MF is the "magic factor" which was determined empirically to be 0.3 [143]. The partial derivatives of coordinate k for inner node i are given by

$$\frac{\partial E}{\partial y_{ik}} = \frac{-2}{C} \sum_{p=1}^{L} \frac{\delta^*_{ip} - \delta_{ip}}{\delta_{ip}\delta^*_{ip}} (y_{ik} - y_{pk}) \tag{6.6}$$

and

$$\frac{\partial^2 E}{\partial y_{ik}^2} = \frac{-2}{C} \sum_{p=1}^{L} \frac{1}{\delta_{ip}\delta^*_{ip}} \left[(\delta^*_{ip} - \delta_{ip}) - \frac{(y_{ik} - y_{jk})^2}{\delta_{ip}} (1 + \frac{\delta^*_{ip} - \delta_{ip}}{\delta_{ip}}) \right] , \tag{6.7}$$

where $L := 3$ for an internal node and $L := 2$ for the root node. Note that, an inner node i is only constrained by three other nodes: one parent node and two child nodes. Thus, in Equation 6.6 and Equation 6.7, the partial derivatives only need to be computed for the three connecting nodes (two nodes for the root node).

Two types of errors can occur during the mapping:

1. $\delta^*_{ij} > \delta_{ij}$: the length of $branch_{ij}$ is longer in T' than in the 2D PCoA plot.

2. $\delta^*_{ij} < \delta_{ij}$: the length of $branch_{ij}$ is shorter in T' than in the 2D PCoA plot.

It is straightforward to display error type 1 in the 2D plot using a thicker stroke (line). We use the Gauss error function:

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{6.8}$$

to scale the width w_{ij} of $branch_{ij}$ between 1.0 to 5.0:

$$w_{ij} = \begin{cases} 1 & \text{if } \delta^*_{ij} \leq \delta_{ij} \\ 1 + 4erf(\frac{\delta^*_{ij}}{\delta_{ij}} - 1) & \text{if } \delta^*_{ij} > \delta_{ij} \end{cases} . \tag{6.9}$$

Thus, in the gradient descent procedure, we use a strategy which tries to minimize error type 2 by updating the branches which contain error type 2 more frequently than those contain error type 1.

The algorithm is summarized in Algorithm 4.

Input: The induced species level tree T' with q leaf nodes; Leaf-node coordinates x_{ik}, where $i = 1...q$ and $k = 1, 2$; Number of iterations n.
Result: Internal-node coordinates y_{ik}, where $i = 1...(q-1)$ and $k = 1, 2$; Branch width w_{ij} for each branch in T'.
Randomly initialize all y_{ik} ;
Compute δ^*_{ij} from T' ;
Compute δ_{ij} using x_{ik} and y_{ik} ;
repeat
  foreach *inner node* i *of* T' **do**
    if $m \ \% \ 5 == 0$ **then**
      update the y_{ik} with Equation 6.5 ;
    else
      update the y_{ik} with Equation 6.5 only if there exists at least one branch connected to this node with $\delta_{ij} > \delta^*_{ij}$;
    end
    update δ_{ij} using x_{ik} and y_{ik} ;
  end
  $m = m + 1$;
until $m > n$;
Compute w_{ij} for each branch in T' with Equation 6.9 ;

Algorithm 4: The PhyloMap Mapping Algorithm

6.3 Results and Discussion

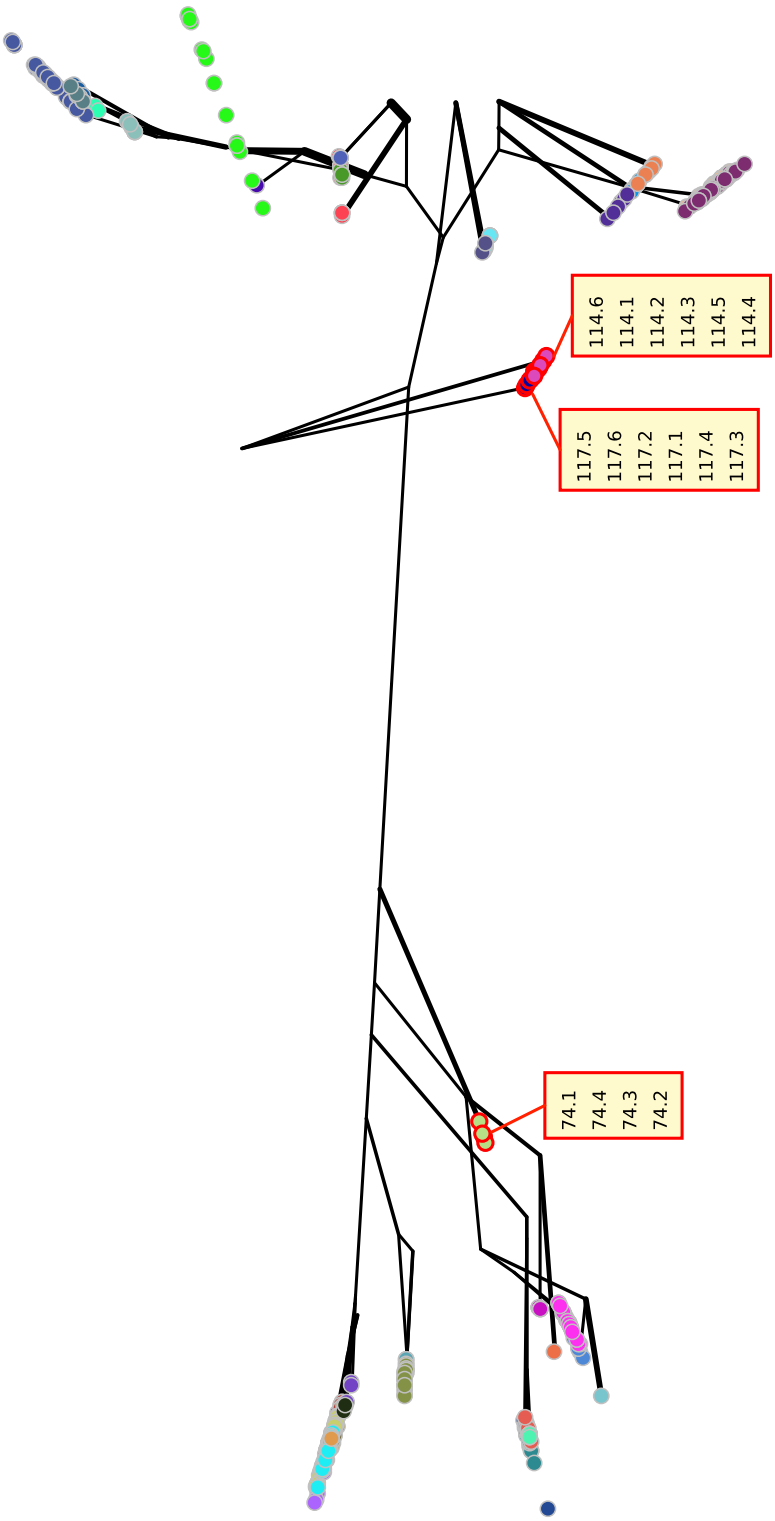

Figure 6.2: PhyloMap plot for the *Rivancidella* dataset (subsubsection 4.4.1.1). The plot is a screen shot of the Processing GUI showing three species with sequence names. The variance explained by the first axis (horizontal axis) is 65.05 %, and 7.53 % by the second axis (vertical axis). Each circle in the plot represents a sequence, and different species are colored differently.

We have computed a PhyloMap for the *Rivancidella* dataset (for details of this dataset, please see subsubsection 4.4.1.1), the result is displayed in Figure 6.3. Figure 6.3 is a screen shot of the PhyloMap GUI. The GUI allows users to view sequence names, all sequence names belonging to one species, change sequence name positions, and export publication quality figures. Furthermore, we make the GUI available via a web service using the Processingjs JavaScript translator (`http://processingjs.org/`). The PhyloMap web service is an integrate part of the PTP web server (section 5.3).

From the *Rivancidella* PhyloMap, we can clearly identify two main sister groups, which need to be shown as two separate subtrees on two printed pages in the original publication [124]. However, without the information from the mapping tree (T'), the plain PCoA result fails to portray the distances between some species. For instance, the species composed of sequences 117.1 - 117.6, and the species composed of sequences 114.1 - 114.6 are indistinguishable in the 2D PCoA plot. But if we take into account the mapped species-level tree, the distances are substantially larger. The real distance may require more dimensions in the PCoA to be properly displayed. Therefore, the tree can be considered as a means to add more dimensions to the 2D PCoA plot.

While phylogenetic tree inference methods have come of age, their interpretation still relies heavily on visual inspection [33, 111]. The difficulties of analyzing huge trees have mostly been addressed by developing sophisticated tree visualization software. Visual data exploration usually follows a three-step process [84]: overview, zoom and filter, and details-on-demand. Despite advances in visualization software [145, 190], it remains difficult to comprehend the entire tree in the overview stage. PhyloMap was developed specifically for improving the overview stage by summarizing the main phylogenetic information. Both PCoA and PTP can be considered data compression techniques that are suited to preserve the most important information in the data. Once the main trends in the data set have been identified, one can zoom into areas of interest, thus reducing the data set to a size that can be more easily visualized.

Other means of adding information to ordination such as superimposing a minimal spanning tree and a relative neighborhood graph have been proposed by Guiller [69]. However, these methods require using all data, and hence generate difficult-to-interpret results when the data set is large. Our method also serves as a generic way to add one more layer of information to a data ordination analysis that can alternatively be described via a tree.

The PCoA used here is a linear dimensionality-reduction technique [140, 170]. Despite the recent advances in nonlinear dimensionality reduction, we believe that PCoA is appropriate for PhyloMap. First, PCoA finds the

greatest variance in the data set. In other words, it preserves the global patterns and this is one of the main goals of PhyloMap. Other methods such as Isomap [170] that uses geodesic distances might not be well-suited for phylogenetic analyses. Methods such as LLE [140] are designed to preserve local properties which is obviously not the goal of PhyloMap. Second, PCoA is robust in the sense that it does not depend on the initialization conditions and is parameters-free.

It is worth noting that the method presented in this Chapter represents a substantial improvement with respect to the original PhyloMap version described in [193]. First, we replaced the "Neural-Gas" [101] sequence clustering method with the PTP species delimitation approach. Neural-Gas requires the number of clusters as input parameter, and lacks a clear biological interpretation of the clusters. PTP, however, can determine the number of sequence clusters automatically by searching for the maximum likelihood solution. The sequence clusters found by PTP can be considered as putative species according to the PSC. Second, we implemented an interactive GUI for displaying the PhyloMap plot. The GUI is essential for a visualization tool, but it was not available in the original PhyloMap.

6.4 Summary

PhyloMap is a robust algorithm for analyzing and displaying phylogenetic relationships in large sequence data sets. It uses the entire input data set (the comprehensive full tree) and avoids the bias introduced by empirical sub-sampling. PhyloMap introduces two data compression techniques (dimensionality reduction and species delimitation) to reduce the size of the data without losing important information. The visualization summarizes the main phylogenetic information and overcomes the shortcomings of standalone phylogenetic tree and ordination analyses.

CHAPTER 7

Paired-End Reads Merger

The content of this Chapter has been derived from the following peer-reviewed publication:
J. Zhang, K. Kobert, T. Flouri, and A. Stamatakis. PEAR: a fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics (Oxford, England)*, pages 1–7, Nov. 2013

Kassian Kobert and I designed the statistical test described in subsection 7.2.2 together, Tomas Flouris implemented the largest part of the C code and developed the memory manager described in subsection 7.2.4.

Illumina paired-end sequencing technology can generate reads from both ends of target DNA fragments, which can subsequently be merged to increase the overall read length (Figure 7.1). There already exist tools for merging these paired-end reads when the target fragments are equally long. However, when fragment lengths vary and, in particular, when either the fragment size is shorter than a single-end read, or longer than twice the size of a single-end read, most state-of-the-art mergers fail to generate reliable results.

We present the PEAR software for merging raw Illumina paired-end reads from target fragments of variable length. The program evaluates all possible paired-end read overlaps and does not require the target fragment size as input. It also implements a statistical test for minimizing the number of false-positive results. Tests on simulated and empirical data show that, PEAR consistently generates highly accurate merged paired-end reads. A

highly optimized implementation allows for merging millions of paired-end reads within a few minutes on a standard desktop computer. On multi-core architectures, the parallel version of PEAR shows linear speedups compared to the sequential version of PEAR.

PEAR is implemented in C and uses POSIX threads. It is freely available at `http://www.exelixis-lab.org/pear`

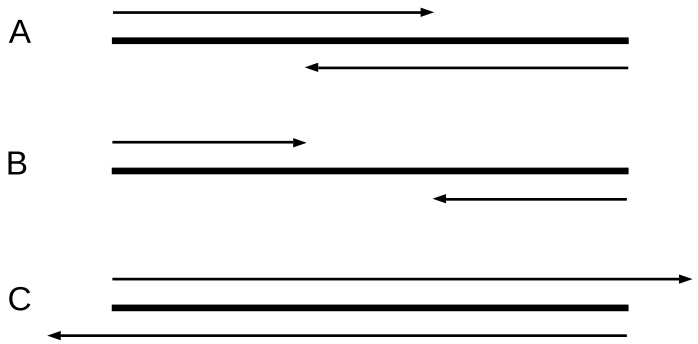

Figure 7.1: Three possible scenarios for paired-end read lengths and target DNA fragment lengths. A: short overlap between the paired-end reads; B: no overlap between the paired-end reads; C: single end read length is larger than the target DNA fragment length.

7.1 Introduction

The Illumina sequencing platform can produce millions of short reads in a single run. The deep sequencing capability and low cost of the sequencing-by-synthesis technology is useful for a plethora of applications ranging from whole-genome sequencing [94, 181] to profiling microbial communities by sequencing the hypervariable regions of the 16S rRNA gene [7, 19, 36, 137, 195]. However, single-end reads produced by the Illumina platform typically have a length that ranges from 75 to 300-bp. Furthermore, there is an exponential increase in error rates along the reads [30] (Figure 7.2).

The Illumina platform can also generate paired-end reads by sequencing the forward and reverse strands of each target DNA fragment. If the target DNA fragment size is smaller than twice the length of the single-end reads, that is, if there exists an overlap, the corresponding paired-end reads can be merged into a fragment. By merging paired-end reads, the overlapping region between them can also be deployed for correcting sequencing errors, and

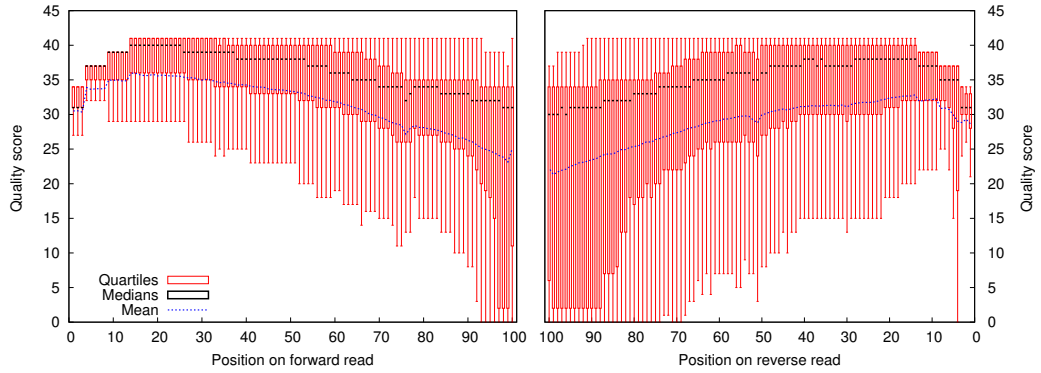

Figure 7.2: Quality score plot of simulated 100-bp paired-end reads before merging.

potentially yield sequences of higher quality (Figure 7.3). Merging paired-end reads is the first processing step in a plethora of sequence analysis pipelines. Hence, its accuracy is crucial for all downstream analyses.

There exist several proof-of-concept mergers such as iTag [36], BIPES [195], and Shera [137]. Some production-level mergers FLASH [100], PANDAseq [102], and COPE [94] have also been recently introduced.

Shera merges the reads by maximizing the number of matches between the paired-end reads. Both, Shera and FLASH (see below), ignore the quality scores of the base calls. Shera merges all reads and leaves it to the user to decide which merged reads are correct. Since it is a proof-of-concept implementation, it is up to 100 times slower than competing mergers.

FLASH constructs merged reads that maximize the overlap length-to-matches ratio. FLASH requires the mean DNA fragment size and standard deviation of the fragment size as input parameters. It can therefore only merge paired-end reads into fragments of "almost" identical size. Furthermore, our tests show that FLASH performs poorly when the overlaps between reads are short.

COPE deploys an analogous approach as FLASH for finding the best overlap, but also takes into account the quality scores of mismatches. COPE is designed to handle deep genome sequencing datasets. Thus, it considers that k-mers that occur infrequently are likely to be sequencing errors. COPE exhibits high memory requirements and also relatively long execution times.

PANDAseq merges fragments by maximizing the probability of true sequence matches, given the observed sequences. It combines quality scores with sequence matches and thereby improves merging quality. In contrast to

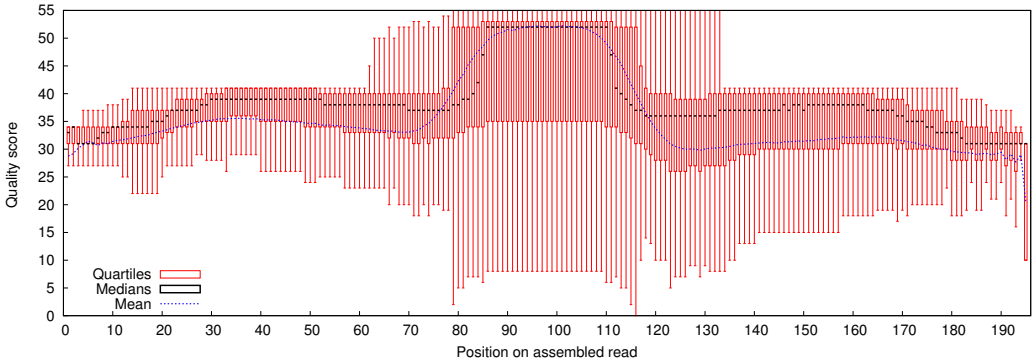

Figure 7.3: Quality score plot of simulated 100-bp paired-end reads after merging (mean overlap = 35-bp).

FLASH, PANDAseq works well with short overlap regions and does not require prior knowledge of the target DNA fragment size. However, it assumes that all paired-end reads can be merged. Thus, if the sample contains DNA fragments that are at least twice as long as the single-end reads, PANDAseq exhibits a high false-positive rate.

Finally, most current paired-end mergers assume that the DNA fragments are longer than the individual single-end reads. When this does not hold, for example when sequencing the V6 region of 16S rRNA genes of bacterial samples (fragment sizes range between 110 and 130-bp [65]) with read lengths of 150-bp (see case A in Figure 7.1), current mergers will generate erroneous results.

Here we present PEAR, a fast and accurate paired-end read merger. PEAR merges reads by maximizing the *assembly score* (AS) of the read overlap via a scoring matrix that penalizes mismatches with a negative value β and rewards matches with a positive value α. Our approach takes quality scores *and* sequence matches into account. It does not require pre-processing of the raw data or specifying the fragment size. Furthermore, PEAR neither requires prior information on read length nor target fragment size. It can reliably identify reads that can either be merged or need to be discarded. The program is accurate on datasets with (i) short overlaps and (ii) DNA target fragment sizes that are smaller than single-end read lengths.

To identify false-positive merged reads, we propose a statistical test that is based on the observed expected alignment scores. On simulated paired-end reads with a mean overlap of 20-bp, PEAR correctly merges 90.44% of the fragments with a false-positive rate of 2.78% when our statistical test is

disabled. It correctly merges 70.06% of the fragments with a false-positive rate of only 0.48% when the significance level of the test is set to 1%. The best competing merger (PANDAseq) correctly merges 83.51% of the fragments, but with a false-positive rate of 6.65%.

We implemented PEAR in C. It includes an optimized memory management scheme that allows the user to specify the amount of RAM available for executing the program. Therefore, it can be deployed on off-the-shelf desktop and laptop computers as well as on high-end multi-core servers. In subsection 7.2.4 we outline why PEAR becomes faster when using less memory. Finally, the run time of the parallel version of PEAR scales linearly with the number of cores.

7.2 The Merging Algorithm

7.2.1 Overlap Algorithm

In paired-end sequencing mode, the Illumina *Consensus Assessment of Sequence and Variation* (CASAVA) software generates two FASTQ files [25], one for each reading direction of the fragment. The files contain exactly the same number of reads. Corresponding paired-end reads can be identified by their coordinates in the flow cell. The Illumina flow cell is a planar optically transparent surface similar to a microscope slide. It contains a lawn of oligonucleotide anchors bound to its surface.

PEAR scores all possible overlaps for each pair of corresponding paired-end reads to determine the overlap with the highest AS (Assembly Score). Subsequently, PEAR conducts a statistical test to assess the statistical significance of the merged reads. If the merged reads do not pass this test or if the overlap length is smaller than a user-defined threshold (based on the expected approximate sequence length in the experiment) the pair of reads will not be merged. Otherwise, PEAR returns the merged fragment and will also correct errors using the Illumina quality scores.

For each base, CASAVA (v1.8) yields an ASCII-encoded quality score, that represents an integer value Q, which can be converted into the probability e of a sequencing error at the base via $e = 10^{\frac{33-Q}{10}}$ ($e = 10^{\frac{64-Q}{10}}$ in earlier CASAVA versions). The base frequency of a nucleotide is the number of occurrences of that nucleotide in the FASTQ files divided by the total number of bases. The probability q of a random base match is: $q = P_A^2 + P_T^2 + P_G^2 + P_C^2$. Given an overlapping region $C := (X, Y)$, where X and Y are the overlapping segments of the two reads, we denote the observed (resp. true) base at position i of the overlap by X_i, Y_i (resp. X'_i, Y'_i). We denote the length of the

overlap region by $|C|$. The probability that base X_i (resp. Y_i) is erroneous is e_{X_i} (resp. e_{Y_i}). Assuming that errors are independent events, we can calculate the probability of a true base match, given the observed base match as

$$\Pr[X'_i = Y'_i | X_i = Y_i] = (1 - e_{X_i})(1 - e_{Y_i}) + e_{X_i} e_{Y_i} \frac{\sum_{b \neq X_i}^{ACGT} P_b^2}{\left(\sum_{b \neq X_i}^{ACGT} P_b\right)^2} .$$

The probability of a true base match, given the observed base mismatch is

$$\begin{aligned} \Pr[X'_i = Y'_i | X_i \neq Y_i] = \quad & (1 - e_{Y_i}) e_{X_i} \frac{P_{Y_i}}{\sum_{b \neq X_i}^{ATGC} P_b} \\ & + (1 - e_{X_i}) e_{Y_i} \frac{P_{X_i}}{\sum_{b \neq Y_i}^{ATGC} P_b} \\ & + e_{X_i} e_{Y_i} \frac{\sum_{b \neq X_i, Y_i}^{ATGC} P_b^2}{(\sum_{b \neq X_i}^{ATGC} P_b)(\sum_{b \neq Y_i}^{ATGC} P_b)} , \end{aligned}$$

and the probability of a true base mismatch, given the observed base mismatch (or match) is

$$\begin{aligned} \Pr[X'_i \neq Y'_i \,|\, X_i \neq Y_i] &= 1 - \Pr[X'_i = Y'_i \,|\, X_i \neq Y_i] \\ \Pr[X'_i \neq Y'_i \,|\, X_i = Y_i] &= 1 - \Pr[X'_i = Y'_i \,|\, X_i = Y_i] . \end{aligned}$$

If any of the bases is undetermined (denoted by N),

$$\begin{aligned} \Pr[X'_i = Y'_i | X_i = N \text{ or } Y_i = N] &= q \\ \Pr[X'_i \neq Y'_i | X_i = N \text{ or } Y_i = N] &= 1 - q . \end{aligned}$$

PEAR calculates the AS for each possible overlap (assuming no gaps, since they are infrequent on Illumina platforms [114]) with a scoring matrix that rewards matches by a positive value α and penalizes mismatches with a negative value β. Scoring matrices for evaluating sequence alignments are routinely used, for instance, in BLAST [2] and Bowtie2 [90]. Elaborate tests using simulated data showed that setting $\alpha := 1.0$ and $\beta := -1.0$ yields the best results. Given the overlap $C := (X, Y)$, we define AS as

$$\sum_{i=1...|C|} (\Pr[X'_i = Y'_i | X_i = Y_i] \cdot \alpha)^{\delta_i} (\Pr[X'_i \neq Y'_i | X_i \neq Y_i] \cdot \beta)^{1-\delta_i} ,$$

where

$$\delta_i = \begin{cases} 1 & : \text{ A match is observed } (X_i = Y_i) \\ 0 & : \text{ A mismatch is observed } (X_i \neq Y_i) \end{cases} .$$

For the merged reads, PEAR computes the overlap that maximizes the AS. We denote the overlap that maximizes the AS by C^*.

7.2.2 Statistical Test to Control False Positive Rate

To test the significance of the merged reads and to identify reads that shall not be merged, we calculate a p-value for the null hypothesis that the two corresponding reads are independent from each other. By independent we mean that, any overlap between the two reads occurs purely by chance. For an overlap $C = (X, Y)$ between two reads x and y, we define $\mathrm{OES}(C)$ to be the *observed expected alignment score* (OES)

$$\mathrm{OES}(C) = \sum_{i=1\ldots|C|} \Pr[X'_i = Y'_i | X_i, Y_i] \cdot \alpha + \Pr[X'_i \neq Y'_i | X_i, Y_i] \cdot \beta$$

and

$$\widehat{\mathrm{OES}}(x, y, \omega) = \max_{\widehat{C} \in D(x,y,\omega)} \mathrm{OES}(\widehat{C}) \ ,$$

where $D(x, y, \omega)$ is the set of all possible overlaps between sequences x and y with a size of at least ω.

Let $\tilde{x}$ and $\tilde{y}$ be two independent random sequences and let us further assume that there are no sequencing errors. Then, the p-value, that is, the probability of a random sequence producing an OES that is at least as high as the OES obtained from the merged reads, is defined as the probability of $\widehat{\mathrm{OES}}(\tilde{x}, \tilde{y}, \omega)$ being greater or equal to the observed $\mathrm{OES}(C^*)$. We obtain

$$\begin{aligned}
\Pr(\widehat{\mathrm{OES}}(\tilde{x}, \tilde{y}, \omega) \geq \mathrm{OES}(C^*)) &= 1 - \Pr(\mathrm{OES}(C^*) > \widehat{\mathrm{OES}}(\tilde{x}, \tilde{y}, \omega)) \\
&\leq 1 - \prod_{C \in D(\tilde{x}, \tilde{y}, \omega)} \Pr\left(\mathrm{OES}(C^*) > \mathrm{OES}(C)\right) \\
&= 1 - \prod_{C \in D(\tilde{x}, \tilde{y}, \omega)} \sum_{k=0}^{\ell(|C|)} \binom{|C|}{k} \cdot q^k \cdot (1-q)^{|C|-k} \\
&\leq 1 - \Big(\prod_{i=\omega}^{\max(l_1, l_2)} \sum_{k=0}^{\ell(i)} \binom{i}{k} \cdot q^k \cdot (1-q)^{i-k} \Big)^2 \\
&\leq 1 - \Big(\prod_{i=\omega}^{\infty} \sum_{k=0}^{\ell(i)} \binom{i}{k} \cdot q^k \cdot (1-q)^{i-k} \Big)^2 \\
&=: p\text{-value} \ ,
\end{aligned}$$

where

$$\ell(c) = \lceil (\mathrm{OES}(C^*) - \beta \cdot c)/(\alpha - \beta) \rceil - 1 \ .$$

By default, PEAR uses an OES with a p-value < 0.01 as cutoff. If the OES of the best merged read is smaller than this value, the reads will not be merged.

Choosing a smaller p-value will reduce the false-positive rate of the merged sequences, but a lower number of reads will be merged.

If the underlying overlap size is unknown, ω can be set to 1.0. If, however, the overlap is known to be short ($\leq$ 35-bp in our simulations), our statistical test will reject up to 4% (based on our simulations, Table 7.2 - Table 7.4) of correctly merged sequences because of low quality scores. To recover more merged reads, we provide the possibility to set ω to the computed overlap size *after* the merging step, instead of using a predefined fixed value. However, when using this work-around, the p-value of the statistical test is not valid anymore, since ω depends on the output of our algorithm. This implies that, the random sequences are more restricted when choosing overlaps than the original input sequences. We will refer to the aforementioned, valid p-value as the *maximal accepted probability* (MAP). Our tests show that PEAR can produce 4% more merged sequences using MAP at the cost of a slight (approximately 0.1%) increase in false-positive rates.

7.2.3 Output

PEAR generates four FASTQ output files. One contains the successfully merged reads, two files contain the forward and reverse unmerged reads, and one the discarded reads. Discarded reads are reads that fail to pass one of the following quality filters, which are applied after the merging process. These filters require the user to set some program parameters, which are outlined below. By default, PEAR does not apply these quality filters.

Minimum quality score for trimming It is common to trim the reads and use their high quality part, due to the low quality of base calls toward the end of Illumina reads [19]. Consequently, PEAR includes the option to trim unmerged reads that contain at least two consecutive bases with quality scores lower than a user-specified *minimum quality score* value.

Minimum length of output sequences PEAR discards merged sequences or trimmed, unmerged reads that are shorter than this threshold.

Maximum length of the output sequences PEAR discards sequences that are longer than the specified maximum length.

Maximum proportion of uncalled bases This parameter allows for discarding reads that contain more than the specified proportion of uncalled bases `N`. When the value is set to 0, it will discard all reads containing one or more uncalled bases `N`.

Now, assume two reads x and y *can* be merged and have an overlapping region C. PEAR will correct errors in the overlapping region and compute updated quality scores for the overlap. For every pair of corresponding observed bases X and Y in C and their quality scores e_X and e_Y, respectively, we distinguish four cases — X and Y are identical, different, one of them is uncalled, or both of them are uncalled. When the two bases are identical, PEAR simply inserts this base into the corresponding position in the merged sequence and assigns the product of the quality scores: $e_X e_Y$ because errors are independent of each other (see section 7.2). However, many other programs assumes a maximum quality score of 40. Quality scores over 40 can easily crash many downstream analysis programs. Thus, PEAR by default caps any quality scores greater than 40 to 40.

When the base pairs are different, PEAR inserts the base with the highest quality score and the corresponding quality score. If (only) one of the two bases is uncalled (`N`), PEAR uses the called base and its quality score. Finally, if both bases are uncalled we arbitrarily use the lower of the two quality scores, since a quality score is required to generate a valid FASTQ output file.

7.2.4 Parallelization and Memory Management

PEAR runs on standard laptop and desktop computers. We implemented a memory allocator and manager that allows PEAR to only use a predefined amount of memory that the user can specify via a command-line switch. PEAR can use several gigabytes, but also just a few kilobytes of RAM.

Current off-the-shelf laptops and servers consist of multi-core processors with a minimum number of two cores per processor, thus increasing the total processing power of the system. However, RAM clock rates are still slower than CPU clock rates (also known as "memory-gap"). Thus, the time required for loading data from RAM into cache memories and registers can lead to performance deterioration. Currently, most tools process sets of paired-end reads iteratively. They load a set of reads and merge them until all reads in this set have been merged. Because disk accesses are serial, most tools suffer from waiting times induced by loading reads into RAM and the caches. To alleviate this performance bottleneck, PEAR uses a standard double-buffering technique.

The main idea of double buffering techniques is to split-up the available RAM (specified by the user) into two buffers of equal size, which we denote as *active* and *passive* buffer. At program start-up there is an initial latency until the active buffer has been filled with reads for the very first time. Then, a dedicated thread (which we denote as *reader* thread) loads a second set of

reads into the passive buffer while the remaining threads process the reads in the active buffer. If the reader thread has already loaded the next set while the remaining threads are still processing the reads in the active buffer, the reader thread will also start merging reads. Thereby, we can parallelize the process of reading from disk and merging reads. By using this technique, we hide the latency of disk accesses and can use the majority of threads/cores for merging short reads and thus reduce overall run time. The optimal RAM setting is machine- and data-set specific. Based on our observations on merging 150-bp paired-end reads data sets on a 48 core Magny-Cours system, we set the default memory buffering to 200MB.

7.3 Experimental Settings

To evaluate PEAR and compare it to the three state-of-the-art mergers (FLASH v1.2.6, PANDAseq v2.4, COPE v1.1.2), we used simulated data sets with varying overlap and DNA fragment sizes as well as the following two empirical data sets:

1. Deep sequencing data of the *Staphylococcus aureus* genome by [99],

2. Reads generated from paired-end sequencing of a known single sequence (template) used by [102] to test PANDAseq.

7.3.1 Simulated Data

To mimic the sequencing of multiple hypervariable regions of 16S rRNA, we extracted a reference sequence data set of 1000 full-length bacterial 16S rRNA gene sequences from the RDP classifier training data set [179]. We then used ART (v1.5.0) [77] to simulate 100-bp paired-end reads, with mean target DNA fragment sizes of 101, 150, 165, 180, 190, and 250-bp, and a standard deviation of 10-bp. We set the parameters of ART to generate target DNA fragments by randomly sampling the reference sequences until a ten-fold coverage of the reference data set was attained. To obtain a more realistic test dataset, we used two read quality profiles for simulating either end of the respective pairs. The target DNA fragments produced by ART provide the ground truth for the merged paired-end reads.

We also generated an additional set of 150-bp long reads with a mean fragment size of 101-bp, by extending all single reads in the above 101-bp fragment size set to a length of 150-bp. We extended the reads by complete random sequences with the lowest possible quality scores. This setup emu-

lates case C (see. Figure 7.1) where the DNA fragment size is smaller than the length of a single-end read.

We executed PEAR, COPE, FLASH, and PANDAseq on the above data sets and compared the lengths of the merged reads with the true fragment lengths. We only consider merged sequences whose length is equal to the true fragments size as correctly merged sequences. When the fragment size is at least twice as long as the single read length, we consider that a result returning unmerged reads is correct. We executed PEAR with three different settings: a) statistical test disabled, b) $p = 0.01$, and c) MAP $= 0.01$. In all tests the minimum overlap size is 1; for all other parameters we use the default values. We ran PANDAseq with default parameters as well as with a minimal overlap setting of 10-bp. FLASH requires the mean fragment length and a proper minimal overlap value in order to work correctly. Therefore, we ran it with the known/true mean fragment lengths. COPE includes four different modes of execution. Mode 0 is similar to the FLASH approach, but with more stringent alignment score parameters. Modes 1 and 2 further utilize k-mer frequencies, and full-mode runs all three modes sequentially and concatenates the results. COPE generated a segmentation fault on our simulated data under COPE modes 1 and 2. Therefore, we only report results obtained under COPE mode 0.

7.3.2 Staphylococcus Aureus Genome Data

This data set was initially generated by [99] (available for download at `http://gage.cbcb.umd.edu/data`) to assess short read-based genome assembly quality. We used the raw data set which contains 647 052 pairs of 101-bp long reads with a mean DNA fragment size of 180-bp and 45-fold coverage of the *Staphylococcus aureus* genome. To determine the true target DNA fragment sizes, we used Bowtie2 [90] to map the merged reads to the reference genome. We use the corresponding *end-to-end mode* in Bowtie2 and do not allow for opening gaps in either sequence (the reads *and* the reference genome). This guarantees that all merged reads that can be mapped to the reference genome are correctly merged. This is because there are two possible scenarios for incorrectly merged reads: (i) they can be longer than the correct one, in which case the sequences can be aligned by opening gaps in the reference sequence or (ii) they are shorter than the correct one, and the sequences can be aligned by opening gaps in the merged sequences. Therefore, we consider that a merged paired-end read is correct only if Bowtie2 finds a hit on the reference genome. Note that, the results are conservative because some of the correctly merged reads might be missed by Bowtie2 due to sequencing errors.

7.3.3 Single Known Sequence Data

We used a data set that was deployed by [102] to assess PANDAseq. The data set contains paired-end reads from a single template sequence. The template sequence is the V3 region of the *Methylococcus capsulatus* (ATCC 33009) 16S rRNA gene. It has a length of 198-bp, including the primers. The FASTQ files contain 673 845 pairs of 108-bp long paired-end reads. Each pair overlaps by exactly 18-bp. We calculate the "true" merged reads by computing a global pair-wise sequence alignment between the merged reads and the template sequence. Subsequently, we check if the overlapping region contains gap. We consider a merged read to be correct if there is no gap. We also calculate the error rate (ER) of the merged reads to evaluate error correction performance. The ER is the average number of errors per merged read (excluding gaps) with respect to the template sequence. We ran PEAR with default parameters. We executed PANDAseq with default parameters and with a minimum overlap setting of 10-bp. We applied FLASH with a template sequence length of 198-bp and a read length of 108-bp.

7.4 Resuls

7.4.1 Simulation

Table 7.1 - Table 7.6 shows experimental results. With the exception of the first test case (no overlaps), PEAR consistently generates a larger number of correctly merged sequences when the statistical test is disabled. PEAR merges fewer correct fragments when the statistical test is enabled. When setting the p-value or MAP to 0.01, PEAR shows lower *false-positive rates* (FPR) than all three competing mergers except for data sets with on overlaps. When we use MAP to evaluate the merged reads, PEAR produces more merged reads with a FPR that is analogous to the FPR generated by PEAR *with* the statistical test. PEAR is robust with respect to short overlaps because it can still merge approximately 40% of the reads when the mean overlap is only 10-bp. The FPR of 0.64% (MAP = 0.01) under this setting is ten times lower than for FLASH and PANDAseq. When reads do not overlap, PEAR classifies them as unmerged with a FPR of 0.03%. Here, the FPR is defined as the fraction of merged reads that should not have been merged. Overall, PEAR shows low FPRs across all test scenarios (overlap lengths). In addition, it does not require any prior knowledge regarding overlap lengths. Therefore, PEAR can be used for merging sequences with varying fragment sizes.

	Merged	Correct [-]	Correct [%]	FPR [%]
COPE (mode0)	31	23 065	99.87	0.13
FLASH	0	23 096	100	0
PANDAseq (default)	12 796	10 300	44.59	55.4
PANDAseq (-o = 10)	10 562	12 534	54.27	45.7
PEAR (test disabled)	8 184	14 912	64.57	35.4
PEAR (p-value=0.01)	8	23 088	99.96	0.03
PEAR (MAP=0.01)	33	23 063	99.86	0.14

Table 7.1: Simulated data set of 100-bp paired-end reads with no overlaps (23 096 pairs).

	Merged	Correct [-]	Correct [%]	FPR [%]
COPE(mode0)	5 755	5 709	22.86	0.80
FLASH	8 968	8 309	33.27	7.34
PANDAseq (default)	19 616	14 690	58.83	25.11
PANDAseq (-o = 10)	17 783	12 053	48.27	32.22
PEAR(test disabled)	19 691	17 112	68.53	13.10
PEAR(p-value=0.01)	9 365	9 315	37.31	0.53
PEAR(MAP=0.01)	10 080	10 015	40.11	0.64

Table 7.2: Simulated data set of 100-bp paired-end reads with 10-bp mean overlaps (24 969 pairs).

PANDAseq performs equally well as PEAR for the majority of cases where the overlaps exceed 20-bp. However, its FPR increases with decreasing overlap size, regardless of the minimal overlap size setting. Furthermore, PANDAseq incorrectly merges 55.4% of the reads that do not overlap and 25.11% of the reads when the mean overlap is set to 10-bp. We will discuss the reasons for this behavior in subsection 7.4.5.

FLASH failed to merge the majority of reads for small overlap sizes, but exhibits low FPRs for merged sequences. FLASH merges reads by maximizing the fraction f (number of matches to overlap size ratio). The default threshold of f in FLASH is 0.75 and the default minimal overlap size (ω) is 10. This setting can be shown to have a p-value of 0.00156 for merged reads by using the statistical test introduced in subsection 7.2.2 and replacing OES with f. However, overlaps that exclusively maximize f might not yield cor-

	Merged	Correct [-]	Correct [%]	FPR [%]
COPE(mode0)	9 819	9 750	37.71	0.70
FLASH	10 917	10 843	41.93	0.67
PANDAseq (default)	23 136	21 596	83.51	6.65
PANDAseq (-o = 10)	22 736	20 722	80.14	8.85
PEAR(test disabled)	24 153	23 386	90.44	3.16
PEAR(p-value=0.01)	18 202	18 115	70.06	0.48
PEAR(MAP=0.01)	19 265	19 165	74.12	0.52

Table 7.3: Simulated data set of 100-bp paired-end reads with 20-bp mean overlaps (25 858 pairs).

	Merged	Correct [-]	Correct [%]	FPR [%]
COPE(mode0)	11 771	11 693	43.27	0.66
FLASH	15 603	15 507	57.37	0.61
PANDAseq (default)	26 068	25 849	95.64	0.84
PANDAseq (-o = 10)	26 267	26 026	96.29	0.92
PEAR(test disabled)	26 866	26 712	98.84	0.57
PEAR(p-value=0.01)	25 939	25 833	95.59	0.41
PEAR(MAP=0.01)	26 380	26 273	97.21	0.41

Table 7.4: Simulated data set of 100-bp paired-end reads with 35-bp mean overlaps (27 026 pairs).

rectly merged sequences. Let us consider two possible overlap sizes ω_1 and ω_2 for paired-end reads x and y, where $\omega_1 < \omega_2$. As an example, we assume $\omega_1 := 10$ with 1 mismatch, $\omega_2 := 50$ with 6 mismatches, and a true overlap size of ω_2. Then $f_{\omega_1} = 0.9 > f_{\omega_2} = 0.88$ and FLASH will choose the overlap of size ω_1 as merged read. Because $OES_{\omega_1} = 9 < OES_{\omega_2} = 38$, PEAR will return the correct result. FLASH also requires the mean fragment length as input, which limits its applicability to datasets with uniform fragment length.

COPE, PANDAseq, and FLASH were unable to merge reads under application scenario C (see Figure 7.1) where the DNA fragment size is smaller than a single-end read Table 7.6). PANDAseq incorrectly merges over one third of the reads in this scenario.

	Merged	Correct [-]	Correct [%]	FPR [%]
COPE(mode0)	7 915	7 858	27.73	0.72
FLASH	20 025	19 940	70.36	0.42
PANDAseq (default)	27 939	27 834	98.21	0.37
PANDAseq (-o = 10)	28 049	27 944	98.61	0.37
PEAR(test disabled)	28 335	28 234	99.63	0.36
PEAR(p-value=0.01)	28 288	28 190	99.47	0.35
PEAR(MAP=0.01)	28 329	28 229	99.61	0.35

Table 7.5: Simulated data set of 100-bp paired-end reads with 50-bp mean overlaps (28 339 pairs).

	Merged	Correct [-]	Correct [%]	FPR [%]
COPE(mode0)	43	0	0	100
FLASH	44	0	0	100
PANDAseq (default)	11 417	0	0	100
PANDAseq (-o = 10)	14 146	0	0	100
PEAR(test disabled)	33 187	33 071	99.56	0.35
PEAR(p-value=0.01)	33 136	33 022	99.41	0.34
PEAR(MAP=0.01)	33 185	33 071	99.56	0.34

Table 7.6: Simulated data set of 150-bp paired-end reads with 100-bp mean overlaps (33 217 pairs).

7.4.2 Staphylococcus aureus genome data

We summarize the results in Table 7.7. All mergers work well in this setting. PANDAseq correctly merges the highest number of reads; PEAR about 2% fewer (stat. test disabled). Nonetheless, a quarter of the reads merged by PANDAseq were not mapped to the reference genome using Bowtie2. In contrast only 4.9% of the merged reads from PEAR could not be mapped. COPE merges fewer reads than PEAR and shows a lower FPR when the statistical test in PEAR is disabled. This is probably because COPE was specifically designed for such deep sequencing datasets.

	Merged	Correct [-]	Correct [%]	FPR [%]
COPE(full mode)	373 543	369 683	57.13	1.03
FLASH	369 276	361 663	55.89	2.06
PANDAseq(default)	534 839	418 747	64.72	21.71
PANDAseq(-o = 10)	533 618	407 477	62.97	23.64
PEAR(test disabled)	411 321	391 157	60.45	4.90
PEAR(p-value=0.01)	202 221	199 764	30.87	1.22
PEAR(MAP=0.01)	257 409	251 714	38.90	2.21

Table 7.7: 647 052 Paired-end reads with mean fragment size 180-bp and read length 101-bp (*Staphylococcus aureus* genome).

	Merged	Correct [-]	FPR [%]	ER
COPE(full mode)	0	0	-	-
FLASH	660984	660030	0.14	0.459
PANDAseq(default)	660593	657602	0.45	0.433
PANDAseq(-o = 10)	660522	657609	0.44	0.430
PEAR(test disabled)	663025	661717	0.20	0.475
PEAR(p-value=0.01)	576225	576035	0.03	0.147
PEAR(MAP=0.01)	578887	578679	0.04	0.149

Table 7.8: Single template 198-bp sequence data set of 673 845 108-bp paired-end reads.

7.4.3 Single known sequence data

For this data set, PEAR merges the highest number of reads when the statistical test is disabled (Table 7.8). When setting $p = 0.01$ and using the test, fewer reads are merged, but only 0.03% of the merged reads are false positives. Both, PANDAseq and FLASH, produce comparable results but with a slightly higher FPR. We executed COPE in full-mode (see subsection 7.3.1) on this data set. COPE did not merge any reads, however. The ER of the raw reads is 0.51. While the overlap size is only 18-bp, all mergers decrease the ER. Merged reads produced by FLASH and PANDAseq show ERs that are slightly lower than PEAR (statistical test disabled). However, PEAR yields 3 times lower ERs when the statistical test is enabled.

7.4.4 Run-time and Memory Requirement

To compare run times and speedups between PEAR and competing mergers we used the data set from subsection 7.3.3. We conducted experiments on an Intel Xeon X7560 4-processor machine with 8 cores each and a total of 32 cores. We excluded COPE from most experiments because it does not merge any reads for this dataset and because it has only been partially parallelized (only the k-mer computation is parallelized).

For the sake of fairness, we updated the three mergers to their latest versions at the time when this thesis was finalized. We used PEAR version 0.9.6 and disabled the empirical frequency option, PANDAseq version 2.8 with default parameters and FLASH version 1.2.11 with default parameters. We first tested the three mergers on the single template sequence data set. PEAR is much slower than PANDAseq and FLASH when a small number of cores were used (The sequential runtimes for the three mergers are: 147s, 14s, and 28s, respectively). However, while PEAR yields close to linear speedups, PANDAseq and FLASH perform poorly on speedups beyond 4 cores (see Figure 7.4). The runtime of PEAR is comparable to PANDAseq and FLASH when using 24 cores (13.5s, 10s, and 7s, respectively).

We then also tested PEAR, PANDAseq, and FLASH on a substantially larger data set of 36,504,800 101-bp long paired-end reads from the Human Chromosome 14 (data available at `http://gage.cbcb.umd.edu/data`). Using 24 cores, PEAR requires 165 seconds to finish, while PANDAseq and FLASH need 180 and 95 seconds, respectively.

7.4.5 Reasons for high false-positive rates in PANDASeq

PANDAseq merges reads by choosing the overlap C, such that $|C| \in [1, \min(|F|, |R|))$ that maximizes

$$\Pr[F, R \,|\, C] = \prod_{i=1\ldots|C|} \Pr[F'_{i+f} = R'_i] \cdot (1/4)^{f+r} \;, \tag{7.1}$$

where F is the forward read sequence and R is the reverse read sequence. When the DNA fragment size exceeds the sum of the lengths of the reads (see Figure 7.1, case B), a merger should not merge the reads. According to Equation 7.1, PANDAseq will merge reads with an overlap C, when:

$$\prod_{i=1\ldots|C|} \Pr[F'_{i+f} = R'_i] > (1/4)^{2|C|} \;. \tag{7.2}$$

Assuming that the merged sequences are generated randomly with all base frequencies being equally likely with probability 0.25 and that all bases have

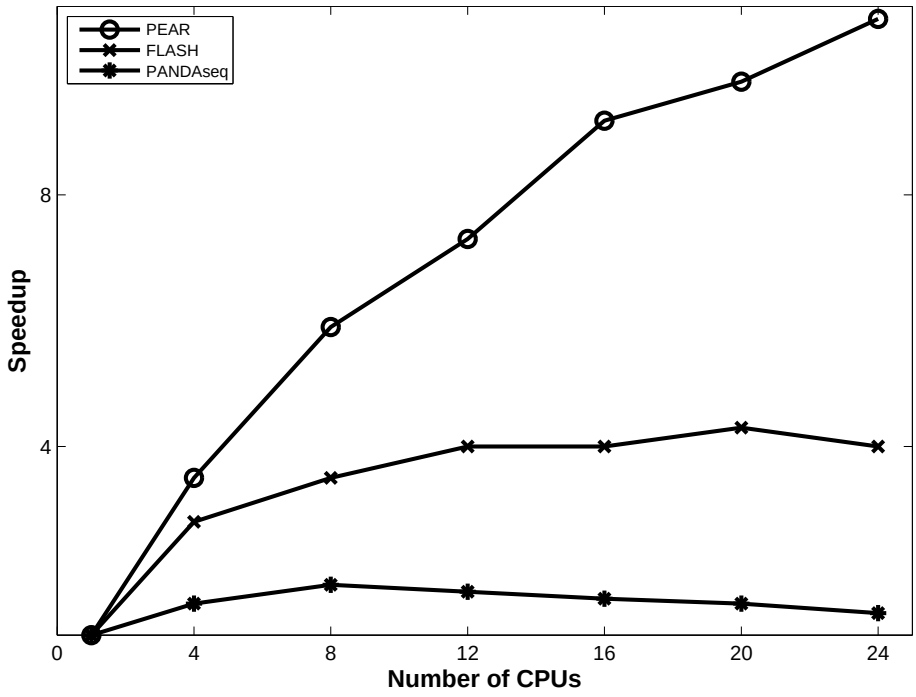

Figure 7.4: Parallel speedups of PEAR, FLASH and PANDAseq on the single template sequence data set.

an equal error probability e, we can simplify Equation 7.2 to

$$\begin{aligned}&\Pr[X'_i = Y'_i | X_i = Y_i]^{|C|/4} \cdot \Pr[X'_i = Y'_i | X_i \neq Y_i]^{3|C|/4} \\ =&((1-e)^2 + \frac{e^2}{3})^{|C|/4} \cdot (\frac{2}{3}(1-e)e + \frac{2}{9}e^2)^{3|C|/4} > (\frac{1}{4})^{2|C|} \ .\end{aligned}$$

Solving the above inequality we obtain $e > 0.039$. In other words, when the bases have an average error probability that is larger than 0.039, PANDAseq will favor merging randomly generated sequences. Since the quality of Illumina reads decreases toward the end of the reads (Figure 7.2), PANDAseq will therefore incorrectly merge reads that do not overlap.

7.5 Summary

We introduced PEAR, a new tool that produces highly accurate merged Illumina paired-end reads with low false-positive rates. It can merge paired-end read data sets under settings where most competing mergers fail. Furthermore, PEAR does neither require preprocessing nor quality control prior to merging. One main application is the merging of paired-end reads from data sets with varying DNA fragment sizes. We have also introduced a statistical test to evaluate the merged reads. Finally, PEAR scales well on most server and desktop architectures.

CHAPTER 8

Application of the PTP model to Phylogenetic Placements

The content of this Chapter has been partly derived from the following peer-reviewed publication:
J. Zhang, P. Kapli, P. Pavlidis, and A. Stamatakis. A general species delimitation method with applications to phylogenetic placements. *Bioinformatics (Oxford, England)*, 29(22):2869–76, Nov. 2013

Pavlos Pavlidis generated the simulated data sets used in section 8.3, and Paschalia Kapli' contribution to the above publication is not included in this Chapter.

This Chapter further extends the PTP model to delimit species on NGS data. We introduce an open reference species delimitation approach that combines PTP with the EPA (EPA-PTP). We show that EPA-PTP not only yields more accurate results than de novo species delimitation methods, but also scales on large datasets because it relies on the parallel implementations of the EPA and RAxML, thereby allowing to delimit species on next generation sequencing (NGS) data in reasonable times.

8.1 Motivation

DNA barcoding studies mostly rely on a single marker gene and are widely used for *DNA taxonomy* [67, 175]. More recently, amplicon based metagenomic (metagenetic) studies that use next generation sequencing (NGS) technologies to perform mass parallel sequencing of barcoding genes, have been deployed to disentangle the structure of microbial communities [19] and in *metabarcoding* biodiversity [26] studies. A central analytical task in such studies is to classify molecular sequences into entities that correspond to species; this is commonly denoted as OTU-picking in metagenomic studies [164]. The main goals of such methods are to identify known species and delimit new species [175].

Numerous approaches exist for associating anonymous reads/query sequences with known species, for instance, nearest-neighbor BLAST [97] or the Naïve Bayesian Classifier [180]. These methods use sequence similarity to associate reads with taxonomic ranks. Phylogeny-aware methods for identifying reads were introduced independently and simultaneously with the evolutionary placement algorithm (EPA [10]) and pplacer [103] (see section 3.4 for details). Instead of sequence similarity, they use the phylogenetic signal in the reference *and* query sequences to attain higher classification accuracy. Note that, obtaining a taxonomic classification from phylogenetic placements represents a difficult task, because phylogenies and taxonomies are frequently incongruent [27]. Placement methods are similar to *closed-reference* OTU-picking [12] or *taxonomy-dependent* methods [149]. Their ability to associate query sequences with species depends on the completeness of the taxon sampling in the reference data [107]. Closed-reference or taxonomy-dependent methods generally lack the ability to delimit new species, consequently they may underestimate the number of species and hence the diversity in the query sequences (see an example in Figure 8.1).

In order to identify new species, *taxonomy-independent* methods or *de novo* OTU-picking approaches are used to initially cluster sequences into so-called Molecular Operational Taxonomic Units (MOTUs) (see section 4.1). Then, one can use a representative sequence from each MOTU cluster and assign a taxonomic rank via taxonomy-dependent methods. While taxonomic assignments may still be inaccurate due to incomplete reference data, coarse-grain biodiversity estimates can be accurate when MOTUs are assigned to higher taxonomic ranks. De novo OTU-picking usually relies on unsupervised machine learning methods [16, 44, 60] that cluster sequences based on, mostly arbitrary, sequence similarity thresholds [128, 149]. As we have shown in section 4.5, MOTUs may correspond to species only when the so-

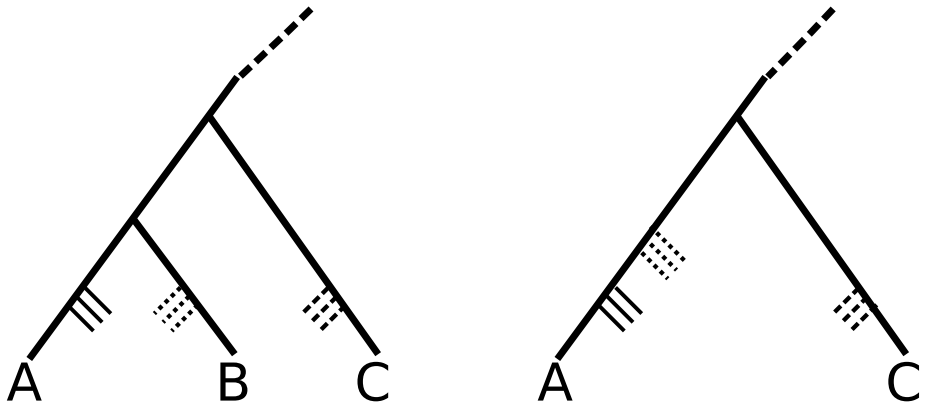

Figure 8.1: EPA may underestimate the number of species in an incomplete reference phylogeny. A, B, and C are closely-related species. If they are all present in the reference tree, the EPA will place the corresponding query sequences onto the three respective branches leading to A, B, and C. However, when B is missing, the EPA will place query sequences belonging to B into the branch leading to A. This will incorrectly classify the query sequences belonging to B and thus, underestimate the number of species. One can set a distance threshold for classifing the query sequences to the known species, however, such a distance threshold is hard to determine.

called barcoding gap is present and the sequence similarity thresholds were correctly set for the clustering algorithms.

The PTP model for species delimitation introduced in chapter 4 can delimit species that are consistent with the Phylogenetic Species Concept (PSC) [48]. However, high-throughput sequencing of barcoding genes can usually produce millions of sequences. Current phylogenetic tree inference software using a stand-alone server only scales to a few thousand sequences. In addition, it becomes increasing difficulty for PTP to find the maximum likelihood solution on large trees because the huge search space (see section 4.3). Thus directly apply PTP to NGS data is not feasible.

8.2 Species Delimitation using Phylogenetic Placements

We introduce an open reference species delimitation approach by integrating PTP with EPA (EPA-PTP). The EPA initially places a large number of query

sequences (short reads) into the branches of a given reference phylogeny. Thereafter, we execute PTP separately and independently for the query sequences assigned to each branch. This allows to annotate the branches of the reference tree by the number of species induced by the query sequences that were placed into each branch. The input of our pipeline is a reference alignment where each sequence represents one species and a reference phylogeny for that alignment.

The EPA-PTP pipeline is implemented in Python and relies on the ETE (python Environment for Tree Exploration) package [79] for tree manipulation and visualization. It is freely available at `https://github.com/zhangjiajie/EPA-classifier`.

Our pipeline executes the following steps:

1. Run UCHIME [47] against the reference alignment to remove chimeric query sequences.

2. Use EPA to place the query sequences onto the reference tree. Sequences that have a maximum placement likelihood weight of less than 0.5 (i.e., an uncertain placement, see 10 for details) are discarded.

3. For each branch in the reference tree, we extract the set of query sequences that have been placed into that branch and infer a tree on them using RAxML [160]. Because the PTP method requires a correctly rooted tree, we employ the following rooting strategy: If the branch leads to a tip, we extend the alignment of the query sequences to include the reference tree tip sequence and the reference sequence that is furthest away from the current tip. This most distant sequence is used as outgroup. Thereby, the tree will be rooted at the longest branch (see the discussion below). To analyze query sequence placements at internal branches we use the RAxML `-g` constraint tree option to obtain a rooted tree of the query sequences. The constraint tree consists of the bifurcating reference tree and a polytomy comprising the query sequences attached to the reference tree branch under consideration. The result of this constrained ML tree search is a resolved tree of query sequences that is attached to the reference tree branch. The attachment point is used as root.

4. Since we assume that the reference phylogeny is a species tree that reflects our knowledge about the speciation process and rate, we initially estimate λ_s only once on the reference phylogeny. Thereafter, we apply PTP to each query sequence (one for each branch of the reference

phylogeny) tree to delimit species. Note that, in this scenario we will only need to estimate λ_c since λ_s is fixed.

5. When PTP is applied to a placement of query sequences on a terminal branch, those queries that are delimited as one species with the reference sequence at the tip will be taxonomically assigned to the species represented by this reference sequence. Otherwise, they are identified as new species in the reference tree.

For the sake of comparison, we have also developed another pipeline that integrates a soft threshold clustering method CROP [70] with EPA (EPA-CROP). The method works analogously as EPA-PTP, with the only difference that CROP is used instead of PTP to calculate the number of MOTUs for each placement.

8.3 Experimental settings

8.3.1 Simulated Datasets

We used the same simulated data sets as described in subsection 4.4.2, in order to compare our open reference approaches to the *de novo* OTU-picking methods. In each simulated alignment, we randomly selected one individual sequence per species as reference sequence and treated the remaining sequences (of that species) as query sequences. To assess the impact of incomplete reference trees on species delimitations, we randomly removed up to 50% of the reference sequences. We deployed the same metrics (NMI) as in subsection 4.4.2 to quantify delimitation accuracy.

8.3.2 Arthropod Meta-barcoding Dataset

This data set contains 673 full-length COI arthropod sequences with a length of 658 bp. The sequences were obtained via PCR-amplification and Sanger-sequencing. Subsequently, these 673 sequences were re-sequenced with a 454-sequencer to generate a total of 133,057 short reads [189]. Using the Sanger data as reference, *Yu et al.* [189] developed meta-barcoding protocols that use the 454-reads to unravel the diversity in the reference data. The authors use a multi-step OTU-picking procedure with different similarity thresholds for clustering the 454 reads and the full-length reference sequences. The method clustered the 673 sequences into 547 MOTUs. The OTU-picking results for the 454 data are summarized in Table 8.13. Our PTP model finds 545 putative species in the 673 full-length sequences when directly applied

to the phylogenetic reference tree. To ensure comparability of results, we used the 547 MOTUs identified in the original study to build a reference tree and reference alignment for testing the EPA-PTP and EPA-CROP pipelines. Initially, we aligned 454 sequences with a length exceeding 100bp to these 547 reference sequences with HMMER [42]. *Yu et al.* [189] initially blasted the 454-MOTU (obtained via three alternative clustering methods) to the Sanger-MOTUs using a threshold of 10^{-10} and 97% minimum similarity. The Sanger-MOTUs that did not match any of the 454-MOTUs are called 'dropouts' by the authors. Inversely, 454-MOTUs that did not match Sanger-MOTUs are called 'no-matches'.

Analogously, in our pipelines, when the delimited species from 454 sequence placements contain one of the full-length reference sequences (see step 4 in section 8.2), we consider this as a 'match'. Further, we denote a full-length reference sequence that is not included in any short read placement delimitation as 'dropout'. Finally, we call a short read placement that is delimited as a new species (i.e., does not contain a reference sequence) as 'no-match'.

8.4 Results

8.4.1 Results for Simulated Datasets

By combining EPA with PTP (or CROP) and applying it to simulated data as described in subsection 4.4.2, we can substantially improve the delimitation accuracy on simulated data (Table 8.1 - Table 8.3, and Table 8.7 - Table 8.9).

When the reference phylogeny includes more than 70% of the reference data, EPA-PTP outperforms all competing approaches, including stand-alone PTP. EPA-PTP outperforms PTP even when the reference phylogeny contains only 50% of the simulated reference data for $b' \leq 20$ (b' is the scaled birth rate per substitution event, see subsection 4.4.2). With increasing b', the reference data needs to be more complete for EPA-PTP to outperform PTP. This is because with increasing b', internal branch lengths tend to get shorter and the EPA placement accuracy decreases. Hence, more data is needed to obtain accurate placements. Note that, under extremely high speciation rates, EPA-PTP performs worse than PTP. The estimation errors may be due to (i) discarding sequences with low likelihood weights, (ii) errors in phylogenetic inferences, or (iii) PTP heuristics failing to find the maximum likelihood species delimitation.

The results for the EPA-CROP pipeline are shown in Tables 8.4 - 8.6, and Tables 8.10 - 8.12. EPA-CROP outperforms the stand-alone version of

CROP, but the results are worse than for EPA-PTP.

b'	5	10	20	40	80	160	Mean
Full ref.	0.989	0.978	0.962	0.933	0.884	0.836	0.930
90% ref.	0.984	0.972	0.955	0.925	0.876	0.830	0.923
80% ref.	0.976	0.966	0.949	0.921	0.872	0.823	0.917
70% ref.	0.971	0.959	0.943	0.912	0.868	0.816	0.911
60% ref.	0.966	0.956	0.939	0.908	0.860	0.805	0.905
50% ref.	0.962	0.950	0.934	0.904	0.853	0.787	0.898

Table 8.1: Species delimitation accuracy (measured in NMI) on simulated, evenly sampled data using the EPA-PTP pipeline with a sequence length 1000-bp

b'	5	10	20	40	80	160	Mean
Full ref.	0.986	0.973	0.956	0.927	0.873	0.822	0.922
90% ref.	0.976	0.962	0.947	0.918	0.865	0.812	0.913
80% ref.	0.967	0.954	0.935	0.908	0.858	0.805	0.904
70% ref.	0.957	0.942	0.925	0.896	0.843	0.784	0.891
60% ref.	0.951	0.935	0.916	0.881	0.829	0.780	0.882
50% ref.	0.941	0.928	0.900	0.865	0.812	0.752	0.866

Table 8.2: Species delimitation accuracy (measured in NMI) on simulated, evenly sampled data using the EPA-PTP pipeline with a sequence length 500-bp

8.4.2 Results for Arthropod Meta-barcoding Dataset

On the *Arthropod* meta-barcoding data, the EPA-PTP pipeline yields substantially better results than the multi-step OTU-picking pipeline used in the original publication (Table 8.13). When the complete full-length reference sequence tree is used, the EPA-PTP pipeline shows substantially lower 'dropout' and 'no-match' rates. It recovers 12.5% more species with respect to the reference data which represents an improvement of over 50%. Here, we apply an analogous criterion as in the original study where at least 2 reads need to be contained in an OTU cluster for it to be considered. In our

b'	5	10	20	40	80	160	Mean
Full ref.	0.978	0.968	0.949	0.918	0.863	0.811	0.914
90% ref.	0.967	0.955	0.935	0.907	0.854	0.800	0.903
80% ref.	0.956	0.944	0.926	0.895	0.846	0.786	0.892
70% ref.	0.942	0.926	0.912	0.880	0.830	0.773	0.877
60% ref.	0.927	0.911	0.893	0.861	0.813	0.755	0.860
50% ref.	0.909	0.891	0.871	0.838	0.784	0.732	0.837

Table 8.3: Species delimitation accuracy (measured in NMI) on simulated, evenly sampled data using the EPA-PTP pipeline with a sequence length 250-bp

b'	5	10	20	40	80	160	Mean
Full ref.	0.986	0.971	0.950	0.907	0.839	0.759	0.902
90% ref.	0.974	0.959	0.940	0.896	0.831	0.750	0.891
80% ref.	0.963	0.949	0.929	0.890	0.825	0.735	0.881
70% ref.	0.951	0.938	0.916	0.870	0.811	0.728	0.869
60% ref.	0.947	0.929	0.904	0.859	0.791	0.712	0.857
50% ref.	0.941	0.917	0.887	0.839	0.770	0.694	0.841

Table 8.4: Species delimitation accuracy (measured in NMI) on simulated, evenly sampled data using the EPA-CROP pipeline with a sequence length 1000-bp

case >= 2 reads need to be contained in a species delimitation. If an OTU cluster or species delimitation only contains one read, it is highly likely that it represents a sequencing error. However, the availability of the complete reference data set is not granted for most meta-barcoding analyses. Thus, as for the simulated data, we randomly removed up to 50% of the reference sequences, and re-ran our pipelines. We then calculated the ratios between the number of species estimated on the reduced reference data relative to the number of species estimated on the complete reference data. The results are shown in Figure 8.2. When species are delimited with taxonomy-dependent approaches such as the EPA, the number of estimated species is expected to decrease with the number of species in the reference data. When combined with PTP (using >= 5 reads per delimitation as cutoff), EPA-PTP yields stable diversity estimates, irrespective of the completeness of the reference

b'	5	10	20	40	80	160	Mean
Full ref.	0.978	0.957	0.924	0.874	0.777	0.686	0.866
90% ref.	0.968	0.948	0.916	0.856	0.770	0.681	0.856
80% ref.	0.955	0.932	0.903	0.854	0.764	0.670	0.846
70% ref.	0.942	0.923	0.894	0.835	0.749	0.648	0.831
60% ref.	0.933	0.909	0.873	0.820	0.733	0.649	0.819
50% ref.	0.918	0.899	0.856	0.799	0.721	0.628	0.803

Table 8.5: Species delimitation accuracy (measured in NMI) on simulated, evenly sampled data using the EPA-CROP pipeline with a sequence length 500-bp

b'	5	10	20	40	80	160	Mean
Full ref.	0.957	0.934	0.877	0.798	0.683	0.564	0.802
90% ref.	0.945	0.923	0.872	0.788	0.674	0.565	0.794
80% ref.	0.934	0.904	0.859	0.784	0.660	0.554	0.782
70% ref.	0.921	0.901	0.839	0.768	0.653	0.563	0.774
60% ref.	0.907	0.876	0.834	0.758	0.647	0.543	0.760
50% ref.	0.886	0.869	0.812	0.735	0.643	0.549	0.749

Table 8.6: Species delimitation accuracy (measured in NMI) on simulated, evenly sampled data using the EPA-CROP pipeline with a sequence length 250-bp

phylogeny. EPA-CROP also yields better results than the multi-step OTU-picking pipeline and stand-alone CROP. The results are slightly worse than for EPA-PTP (Table 8.14).

b'	5	10	20	40	80	160	Mean
Full ref.	0.962	0.948	0.923	0.893	0.836	0.791	0.892
90% ref.	0.958	0.945	0.920	0.889	0.835	0.789	0.889
80% ref.	0.951	0.940	0.917	0.884	0.830	0.778	0.883
70% ref.	0.948	0.935	0.913	0.882	0.829	0.775	0.880
60% ref.	0.940	0.925	0.908	0.880	0.824	0.773	0.875
50% ref.	0.936	0.925	0.899	0.878	0.820	0.762	0.870

Table 8.7: Species delimitation accuracy (measured in NMI) on simulated, unevenly sampled data using the EPA-PTP pipeline with a sequence length 1000-bp

b'	5	10	20	40	80	160	Mean
Full ref.	0.969	0.956	0.931	0.899	0.832	0.776	0.893
90% ref.	0.966	0.953	0.925	0.894	0.829	0.768	0.889
80% ref.	0.957	0.943	0.920	0.891	0.822	0.762	0.882
70% ref.	0.951	0.938	0.918	0.883	0.814	0.750	0.875
60% ref.	0.940	0.930	0.950	0.868	0.815	0.741	0.874
50% ref.	0.934	0.920	0.897	0.856	0.801	0.724	0.855

Table 8.8: Species delimitation accuracy (measured in NMI) on simulated, unevenly sampled data using the EPA-PTP pipeline with a sequence length 500-bp

b'	5	10	20	40	80	160	Mean
Full ref.	0.968	0.954	0.924	0.890	0.819	0.758	0.885
90% ref.	0.960	0.946	0.917	0.881	0.813	0.750	0.877
80% ref.	0.950	0.935	0.911	0.867	0.805	0.739	0.867
70% ref.	0.942	0.925	0.902	0.861	0.796	0.724	0.858
60% ref.	0.927	0.917	0.888	0.843	0.785	0.706	0.844
50% ref.	0.922	0.890	0.873	0.833	0.765	0.685	0.828

Table 8.9: Species delimitation accuracy (measured in NMI) on simulated, unevenly sampled data using the EPA-PTP pipeline with a sequence length 250-bp

b'	5	10	20	40	80	160	Mean
Full ref.	0.967	0.953	0.923	0.876	0.796	0.716	0.871
90% ref.	0.966	0.950	0.923	0.874	0.792	0.710	0.869
80% ref.	0.959	0.942	0.915	0.868	0.783	0.705	0.862
70% ref.	0.951	0.937	0.910	0.861	0.779	0.693	0.855
60% ref.	0.948	0.934	0.902	0.850	0.774	0.690	0.849
50% ref.	0.949	0.922	0.891	0.826	0.767	0.679	0.839

Table 8.10: Species delimitation accuracy (measured in NMI) on simulated, unevenly sampled data using the EPA-CROP pipeline with a sequence length 1000-bp

b'	5	10	20	40	80	160	Mean
Full ref.	0.962	0.941	0.899	0.843	0.742	0.651	0.839
90% ref.	0.957	0.937	0.897	0.836	0.732	0.635	0.832
80% ref.	0.950	0.930	0.885	0.826	0.733	0.639	0.827
70% ref.	0.942	0.925	0.884	0.824	0.714	0.619	0.818
60% ref.	0.930	0.917	0.871	0.815	0.713	0.615	0.810
50% ref.	0.919	0.901	0.854	0.781	0.692	0.591	0.789

Table 8.11: Species delimitation accuracy (measured in NMI) on simulated, unevenly sampled data using the EPA-CROP pipeline with a sequence length 500-bp

b'	5	10	20	40	80	160	Mean
Full ref.	0.945	0.922	0.855	0.770	0.647	0.539	0.779
90% ref.	0.935	0.905	0.850	0.766	0.640	0.537	0.772
80% ref.	0.925	0.897	0.829	0.740	0.631	0.524	0.757
70% ref.	0.914	0.887	0.833	0.746	0.640	0.522	0.757
60% ref.	0.901	0.870	0.809	0.743	0.610	0.532	0.744
50% ref.	0.891	0.859	0.799	0.704	0.610	0.508	0.728

Table 8.12: Species delimitation accuracy (measured in NMI) on simulated, unevenly sampled data using the EPA-CROP pipeline with a sequence length 250-bp

	OTU-picking			EPA-PTP		
	No. cluster	drop-out	no-match	No. cluster	drop-out	no-match
>= 1 reads	973	19%	42.8%	587	7.3%	13.6%
>= 2 reads	602	24%	25.4%	516	11.5%	6.2%
>= 5 reads	-	36%	-	441	21.9%	3.2%

Table 8.13: *Arthropod* data set: Number of estimated MOTUs and species for the complete reference data and tree. Sanger data (the reference data set) has a total of 547 MOTUs. The '-' indicates that the number is not available in the original publication.

	CROP stand alone			EPA-CROP		
	No. cluster	drop-out	no-match	No. cluster	drop-out	no-match
>= 1 reads	671	33.6%	45.9%	652	7.5%	22.4%
>= 2 reads	465	37.7%	26.7%	538	11.9%	10.4%
>= 5 reads	349	44.6%	13.2%	442	22.5%	4.1%

Table 8.14: *Arthropod* data set: Number of estimated OTUs and species for the complete reference data and tree using CROP. Sanger data (the reference data set) has a total of 547 OTUs.

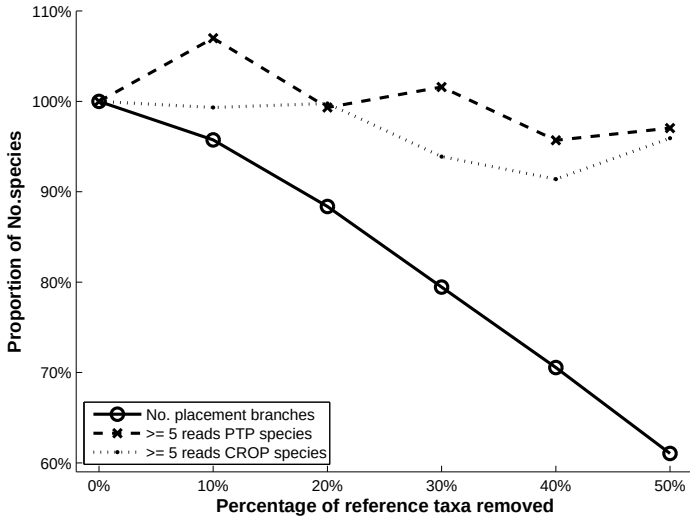

Figure 8.2: Number of estimated species on incomplete reference trees.

8.5 Summary

The EPA-PTP pipeline represents the first integrated approach for analyzing metagenetic data that combines the phylogenetic placement algorithm with an explicit statistical criterion for species delimitation. On a representative empirical dataset, our pipeline yields a substantially more accurate diversity estimate than traditional OTU-picking methods. Using simulated data, we show that, open-reference based approaches can improve delimitation accuracy compared to de novo approaches. More importantly, the EPA-PTP pipeline allows for applying a widely accepted species concept to metagenetic data, where millions of sequences need to be processed.

CHAPTER 9

Conclusion and Future Work

This final chapter provides the conclusion of the thesis and lists potential directions for future research.

9.1 Conclusion

This thesis introduced several novel models and algorithms for phylogenetic marker analysis. Our methods cover species delimitation (PTP, chapter 4 and chapter 5), data visualization (PhyloMap, chapter 6), NGS data processing (PEAR, chapter 7) and metagenetic data analysis (EPA-PTP, chapter 8).

The PTP model conducts species delimitations on single-locus data. Our simulations show that PTP generally outperforms other methods for species delimitation. It is also easier to use than a popular alternative method - the GMYC model. The Bayesian PTP extension further improves the likelihood search and provides reliable delimitation confidence measures. We also make the PTP available through a web server.

The PTP model models nucleotide substitution directly, in contrast to classical models such as Birth-Death Process (BDP), that model time. This gives PTP a few advantages over BDP. First, chronological data is usually difficulty to acquire while nucleotide substitutions are directly observable. Second, PTP greatly simplifies the models based on BDP, in the sense that, it does not depend on molecular clock assumptions. And finally, substitution models are well established and many related tools have been optimized, so there is a good foundation for applying PTP.

By combing ordination with PTP, we developed a new method, PhyloMap, for visualizing large phylogenetic marker data sets. The key contribution of PhyloMap is its mapping algorithm that overlaps two types of data representation.

PEAR is currently one of the most accurate paired-end read mergers, it utilizes all the information available from Illumina raw reads data and works well under all overlap scenarios. PEAR also implemented a statistical test that greatly reduces false-positive merges. We showed that, PEAR consistently produces higher quality and more reliable results than all other state-of-the-art mergers.

Finally, The EPA-PTP is the first phylogeny-aware pipeline for analyzing metagenetic data that offers an explicit statistical criterion for species delimitation. It allows PTP be applied to massive NGS data and improves species delimitation accuracy where "good" reference data sets are available.

9.2 Future Work

9.2.1 PTP

From a theoretical point of view, there are at least three directions that can be explored in the future. The first one is to study how PTP correspond to BDP and further extend the PTP model to better fit the data. Additional classes of *Poisson tree processes* (λ parameters) may be added to the model. However, this will be a challenge for the maximum likelihood search because of the huge search space. We can also allow the number of λ parameters to vary, thus the reversible-jump MCMC need to be introduced for the Bayesian version [188]. The second one is to integrate PTP with existing nucleotide substitution models to create a new family of models, because currently it is implicitly conditioned on the nucleotide substitution model for the maximum likelihood version, and marginalized over for the Bayesian version. The last one is to extend PTP to work on multi-locus data.

For the maximum likelihood version of PTP, the three heuristic algorithms should be more thoroughly tested with respect to their abilities to explore the likelihood landscape. One may consider to develop some more efficient heuristic algorithms based on the tree shape, because the search space is determined by the tree shape. Furthermore, the current PTP model is based on rooted trees, but it is possible to adapt it to unrooted trees if there exist more sophisticated search algorithms.

For the Bayesian PTP, it is important to evaluate other consensus functions [112] that can combine multiple partitions. Some consensus functions

can accommodate missing data, which will be important to extend PTP to multi-locus data. The Bayesian PTP is also lacking an automated convergence assessment procedure, which should ideally be developed by considering the tree shape as well.

It is unclear how species can be defined on viruses, however, we noticed that *Castel et al.* have applied GMYC and PTP to Hantaviruses [21]. Thus, it is tempting to apply species delimitation methods to other viruses, such as Influenza A viruses where massive amounts of data are available, and lineages are less well defined [193].

Finally, PTP is currently implemented in Python. In the future, it should be re-implemented in C or C++ to scale on larger trees and be able to compute more MCMC iterations.

9.2.2 PhyloMap

PhyloMap currently has a GUI that can display the results in two dimensions. It is straight-forward to extend PhyloMap to 3D, which could add more information to the plot. The GUI should also provide functions to zoom in and zoom out to certain regions of the plot and re-compute the PhyloMap on demand. PhyloMap can also be applied to other types of data, such as the beta-diversity calculated by QIIME [18].

9.2.3 PEAR

PEAR is already well optimized, so there is little room for improvement. However, the memory buffer has to be set manually by the user, thus we intend to implement an automatic buffer size tuning routine in PEAR to maximize performance without user intervention.

Currently, PEAR can only perform very limited post-processing after merging. The post processing steps such as collapsing identical reads, fast clustering of similar reads, removing adapter sequences and splitting according to barcode tags are algorithmically trivial, but are essential steps for further analysis. Therefore, we are also planning to add post processing routines to PEAR.

9.2.4 EPA-PTP Pipeline

The EPA-PTP pipeline can only identify known species and delimit new species. It lacks the ability to assign new species to a higher taxonomic rank. Thus, future work should focus on an integrated approach for species delimitation and taxonomic assignment under the EPA framework. However,

this can be quite challenging. To start with, compiling a good reference data set is not trivial and well-studied references are rare. More importantly, the current taxonomy is not aways consistent with the phylogeny. This situation is unlikely to change because some of the taxonomic ranks were not defined based on molecular sequences. However, there exist some approaches that try to correct the taxonomy based on phylogenetic markers [29, 105].

Compared to OTU-picking methods, EPA-PTP requires substantially more CPU time. While most OTU-picking methods can run on an off-the-shelf desktop computer, the EPA-PTP pipeline requires a multi-core server for analyzing large metagenetic datasets. There are several ways to improve EPA-PTP performance. First, the input reads can be pre-clustered to remove or combine very similar reads which might be due to sequencing errors. Second, the full reference tree may be divided into several smaller subtrees for a divide and conquer approach. Finally, as we have described in subsection 9.2.1, PTP can be further improved and re-implemented in C/C++.

List of Figures

List of Tables

List of Acronyms

AA - Amino Acid
AS - Assembly Score
BS - Bootstrap Support
BDP - Birth-Death Process
CPU - Central Processing Unit
DNA - Deoxyribonucleic Acid
EPA - Evolutionary Placement Algorithm
FPR - False-Positive Rates
GTR - General Time Reversible
GMYC - General Mixed Yule Coalescent
GUI - Graphical User Interface
MAP - Maximal Accepted Probability
MI - Mutual Information
NGS - Next Generation Sequencing
NMI - Normalized Mutual Information
NNI - Nearest Neighbor Interexchange
MCMC - Markov Chain Monte Carlo
OES - Observed Expected alignment Score
OTU - Operational Taxonomic Unit
PTP - Poisson Tree Processes
PSC - Phylogenetic Species Concept
RNA - Ribonucleic Acid
SPR - Subtree Pruning and Regrafting
TBR - Tree Bisection and Reconnection

Bibliography

[1] 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, Aug. 2014.

[2] S. Altschul and W. Gish. Local alignment statistics. *Methods in Enzymology*, 266:460–480, 1996.

[3] 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):403–10, Oct. 1990.

[4] H. Amrine-Madsen, K.-P. Koepfli, R. K. Wayne, and M. S. Springer. A new phylogenetic marker, apolipoprotein B, provides compelling evidence for eutherian relationships. *Molecular Phylogenetics and Evolution*, 28(2):225–240, Aug. 2003.

[5] C. Andrieu, N. D. Freitas, A. Doucet, and M. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

[6] T. G. Barraclough and S. Nee. Phylogenetics and speciation. *Trends in Ecology & Evolution*, 16(7):391–399, July 2001.

[7] A. K. Bartram, M. D. J. Lynch, J. C. Stearns, G. Moreno-Hagelsieb, and J. D. Neufeld. Generation of multimillion-sequence 16S rRNA gene libraries from complex microbial communities by assembling paired-end illumina reads. *Applied and Environmental Microbiology*, 77(11):3846–52, June 2011.

[8] D. Baum and M. Donoghue. Choosing among alternative phylogenetic species concepts. *Systematic Botany*, 20(4):560–573, 1995.

[9] D. Baum and K. Shaw. Genealogical perspectives on the species problem. *Experimental and Molecular Approaches to Plant Biosystematics*, 53:289–303, 1995.

[10] S. A. 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, May 2011.

[11] S. a. Berger and A. Stamatakis. Aligning short reads to reference alignments and trees. *Bioinformatics (Oxford, England)*, 27(15):2068–75, Aug. 2011.

[12] H. M. Bik, D. L. Porazinska, S. Creer, J. G. Caporaso, R. Knight, and W. K. Thomas. Sequencing our way towards understanding global eukaryotic biodiversity. *Trends in Ecology & Evolution*, 27(4):233–43, Apr. 2012.

[13] L. Blanco-Bercial, A. Cornils, N. Copley, and A. Bucklin. DNA barcoding of marine copepods: assessment of analytical approaches to species identification. *PLoS Currents*, 6, Jan. 2014.

[14] N. a. Bokulich, S. Subramanian, J. J. Faith, D. Gevers, J. I. Gordon, R. Knight, D. a. Mills, and J. G. Caporaso. Quality-filtering vastly improves diversity estimates from Illumina amplicon sequencing. *Nature Methods*, 10(1):57–9, Jan. 2013.

[15] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, Sept. 1995.

[16] Y. Cai and Y. Sun. ESPRIT-Tree: hierarchical clustering analysis of millions of 16S rRNA pyrosequences in quasilinear computational time. *Nucleic Acids Research*, 39(14):e95, Aug. 2011.

[17] S. Capella-Gutierrez, F. Kauff, and T. Gabaldón. A phylogenomics approach for selecting robust sets of phylogenetic markers. *Nucleic Acids Research*, 42(7):e54, Apr. 2014.

[18] J. G. Caporaso, J. Kuczynski, J. Stombaugh, K. Bittinger, F. D. Bushman, E. K. Costello, N. Fierer, A. G. Peña, J. K. Goodrich, J. I. Gordon, G. A. Huttley, S. T. Kelley, D. Knights, J. E. Koenig, R. E. Ley, C. A. Lozupone, D. McDonald, B. D. Muegge, M. Pirrung, J. Reeder, J. R. Sevinsky, P. J. Turnbaugh, W. A. Walters, J. Widmann, T. Yatsunenko, J. Zaneveld, and R. Knight. QIIME allows analysis of high-throughput community sequencing data. *Nature Methods*, 7(5):335–6, May 2010.

[19] J. G. Caporaso, C. L. Lauber, W. A. Walters, D. Berg-Lyons, C. A. Lozupone, P. J. Turnbaugh, N. Fierer, and R. Knight. Global patterns of 16S rRNA diversity at a depth of millions of sequences per sample. *Proceedings of the National Academy of Sciences of the United States of America*, 108 Suppl(Supplement_1):4516–22, Mar. 2011.

[20] B. C. Carstens and T. a. Dewey. Species delimitation using a combined coalescent and information-theoretic approach: an example from North American Myotis bats. *Systematic Biology*, 59(4):400–14, July 2010.

[21] G. Castel, M. Razzauti, E. Jousselin, G. J. Kergoat, and J.-F. Cosson. Changes in diversification patterns and signatures of selection during the evolution of murinae-associated hantaviruses. *Viruses*, 6(3):1112–34, Jan. 2014.

[22] S. Chang, J. Zhang, X. Liao, X. Zhu, D. Wang, J. Zhu, T. Feng, B. Zhu, G. F. Gao, J. Wang, H. Yang, J. Yu, and J. Wang. Influenza Virus Database (IVDB): an integrated information resource and analysis platform for influenza virus research. *Nucleic Acids Research*, 35(Database issue):D376–80, Jan. 2007.

[23] J.-M. Chen, Y.-X. Sun, J.-W. Chen, S. Liu, J.-M. Yu, C.-J. Shen, X.-D. Sun, and D. Peng. Panorama phylogenetic diversity and distribution of type A influenza viruses based on their six internal gene sequences. *Virology Journal*, 6:137, Jan. 2009.

[24] H. Christensen, P. Kuhnert, J. E. Olsen, and M. Bisgaard. Comparative phylogenies of the housekeeping genes atpD, infB and rpoB and the 16S rRNA gene within the Pasteurellaceae. *International Journal of Systematic and Evolutionary Microbiology*, 54(Pt 5):1601–9, Sept. 2004.

[25] 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):1767–71, Apr. 2010.

[26] E. Coissac, T. Riaz, and N. Puillandre. Bioinformatic challenges for DNA metabarcoding of plants and animals. *Molecular Ecology*, 21(8):1834–47, Apr. 2012.

[27] J. R. Cole, Q. Wang, E. Cardenas, J. Fish, B. Chai, R. J. Farris, A. S. Kulam-Syed-Mohideen, D. M. McGarrell, T. Marsh, G. M. Garrity, and

J. M. Tiedje. The Ribosomal Database Project: improved alignments and new tools for rRNA analysis. *Nucleic Acids Research*, 37(Database issue):D141–5, Jan. 2009.

[28] 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(Database issue):D633–42, Jan. 2014.

[29] 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(Database issue):D633–42, Jan. 2014.

[30] M. P. Cox, D. A. Peterson, and P. J. Biggs. SolexaQA: At-a-glance quality assessment of Illumina second-generation sequencing data. *BMC Bioinformatics*, 11(1):485, Jan. 2010.

[31] S. C. Cox, S. Carranza, and R. P. Brown. Divergence times and colonization of the Canary Islands by Gallotia lizards. *Molecular Phylogenetics and Evolution*, 56(2):747–57, Aug. 2010.

[32] J. Cracraft. Species concepts and speciation analysis. *Current Ornithology*, 1:159–187, 1983.

[33] D. Dao. *Automated Plausibility Analysis of Large Phylogenies*. Bachelor thesis, Karlsruhe Institute of Technology, Germany, 2014.

[34] J. I. Davis and K. C. Nixon. Populations, Genetic Variation, and the Delimitation of Phylogenetic Species. *Systematic Biology*, 41(4):421–435, Dec. 1992.

[35] K. De Queiroz. Species concepts and species delimitation. *Systematic Biology*, 56(6):879–86, Dec. 2007.

[36] P. H. Degnan and H. Ochman. Illumina-based analysis of microbial community diversity. *The ISME Journal*, 6(1):183–94, Jan. 2012.

[37] T. Z. DeSantis, P. Hugenholtz, K. Keller, E. L. Brodie, N. Larsen, Y. M. Piceno, R. Phan, and G. L. Andersen. NAST: a multiple sequence alignment server for comparative analysis of 16S rRNA genes. *Nucleic Acids Research*, 34(Web Server issue):W394–9, July 2006.

[38] J. W. Drake, B. Charlesworth, D. Charlesworth, and J. F. Crow. Rates of spontaneous mutation. *Genetics*, 148(4):1667–86, Apr. 1998.

[39] A. J. Drummond and A. Rambaut. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evolutionary Biology*, 7(1):214, Jan. 2007.

[40] R. Durrett. *Essentials of Stochastic Processes*. Springer, 2nd edition, 2012.

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

[42] S. R. Eddy. A new generation of homology search tools based on probabilistic inference. *Genome informatics. International Conference on Genome Informatics*, 23(1):205–11, Oct. 2009.

[43] R. C. Edgar. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–7, Jan. 2004.

[44] R. C. Edgar. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics (Oxford, England)*, 26(19):2460–1, Oct. 2010.

[45] R. C. Edgar. UPARSE: highly accurate OTU sequences from microbial amplicon reads. *Nature Methods*, 10(10):996–8, Oct. 2013.

[46] R. C. Edgar and S. Batzoglou. Multiple sequence alignment. *Current Opinion in Structural Biology*, 16(3):368–73, June 2006.

[47] R. C. Edgar, B. J. Haas, J. C. Clemente, C. Quince, and R. Knight. UCHIME improves sensitivity and speed of chimera detection. *Bioinformatics (Oxford, England)*, 27(16):2194–200, Aug. 2011.

[48] N. Eldredge and J. Cracraft. *Phylogenetic patterns and the evolutionary process: Method and theory in comparative biology*. Columbia Univ Press, New York, 1980.

[49] I. Elias. Settling the intractability of multiple alignment. *Journal of Computational Biology*, 13(7):1323–39, Sept. 2006.

[50] D. D. Ence and B. C. Carstens. SpedeSTEM: a rapid and accurate method for species delimitation. *Molecular Ecology Resources*, 11(3):473–80, May 2011.

[51] J. a. Esselstyn, B. J. Evans, J. L. Sedlock, F. A. Anwarali Khan, and L. R. Heaney. Single-locus species delimitation: a test of the mixed Yule-coalescent model, with an empirical application to Philippine round-leaf bats. *Proceedings. Biological sciences / The Royal Society*, 279(1743):3678–86, Sept. 2012.

[52] C. Feldman, Richard M., Valdez-Flores. *Applied Probability and Stochastic Processes*. Springer, 2nd edition, 2010.

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

[54] J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Sunderland, MA, 2004.

[55] R. Fleissner, D. Metzler, and A. von Haeseler. Simultaneous statistical multiple alignment and phylogeny reconstruction. *Systematic Biology*, 54(4):548–61, Aug. 2005.

[56] W. Fletcher and Z. Yang. INDELible: a flexible simulator of biological sequence evolution. *Molecular Biology and Evolution*, 26(8):1879–88, Aug. 2009.

[57] R. Floyd, E. Abebe, A. Papert, and M. Blaxter. Molecular barcodes for soil nematode identification. *Molecular Ecology*, 11(4):839–850, Apr. 2002.

[58] D. Fontaneto, E. a. Herniou, C. Boschetti, M. Caprioli, G. Melone, C. Ricci, and T. G. Barraclough. Independently evolving species in asexual bdelloid rotifers. *PLoS Biology*, 5(4):e87, Apr. 2007.

[59] D. N. Frank, C. E. Robertson, C. M. Hamm, Z. Kpadeh, T. Zhang, H. Chen, W. Zhu, R. B. Sartor, E. C. Boedeker, N. Harpaz, N. R. Pace, and E. Li. Disease phenotype and genotype are associated with shifts in intestinal-associated microbiota in inflammatory bowel diseases. *Inflammatory Bowel Diseases*, 17(1):179–84, Jan. 2011.

[60] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics (Oxford, England)*, 28(23):3150–2, Dec. 2012.

[61] T. Fujisawa and T. G. Barraclough. Delimiting Species Using Single-locus Data and the Generalized Mixed Yule Coalescent (GMYC) Approach: A Revised Method and Evaluation on Simulated Datasets. *Systematic Biology*, 0(0):1–18, May 2013.

[62] M. K. Fujita, A. D. Leaché, F. T. Burbrink, J. a. McGuire, and C. Moritz. Coalescent-based species delimitation in an integrative taxonomy. *Trends in Ecology & Evolution*, 27(9):480–8, Sept. 2012.

[63] R. J. Garten, C. T. Davis, C. a. Russell, B. Shu, S. Lindstrom, A. Balish, W. M. Sessions, X. Xu, E. Skepner, V. Deyde, M. Okomo-Adhiambo, L. Gubareva, J. Barnes, C. B. Smith, S. L. Emery, M. J. Hillman, P. Rivailler, J. Smagala, M. de Graaf, D. F. Burke, R. a. M. Fouchier, C. Pappas, C. M. Alpuche-Aranda, H. López-Gatell, H. Olivera, I. López, C. a. Myers, D. Faix, P. J. Blair, C. Yu, K. M. Keene, P. D. Dotson, D. Boxrud, A. R. Sambol, S. H. Abid, K. St George, T. Bannerman, A. L. Moore, D. J. Stringer, P. Blevins, G. J. Demmler-Harrison, M. Ginsberg, P. Kriner, S. Waterman, S. Smole, H. F. Guevara, E. a. Belongia, P. a. Clark, S. T. Beatrice, R. Donis, J. Katz, L. Finelli, C. B. Bridges, M. Shaw, D. B. Jernigan, T. M. Uyeki, D. J. Smith, A. I. Klimov, and N. J. Cox. Antigenic and genetic characteristics of swine-origin 2009 A(H1N1) influenza viruses circulating in humans. *Science*, 325(5937):197–201, July 2009.

[64] A. GEORGES, M. ADAMS, and W. McCORD. Electrophoretic delineation of species boundaries within the genus Chelodina (Testudines: Chelidae) of Australia, New Guinea and Indonesia. *Zoological Journal of the Linnean Society*, 134(4):401–421, Apr. 2002.

[65] G. B. Gloor, R. Hummelen, J. M. Macklaim, R. J. Dickson, A. D. Fernandes, R. MacPhee, and G. Reid. Microbiome profiling by illumina sequencing of combinatorial sequence-tagged PCR products. *PloS One*, 5(10):e15406, Jan. 2010.

[66] P. Goldstein and R. Desalle. Conservation genetics at the species boundary. *Conservation Biology*, 14(1):120–131, 2000.

[67] P. Z. Goldstein and R. DeSalle. Integrating DNA barcode data and taxonomic practice: determination, discovery, and description. *BioEssays*, 33(2):135–47, Feb. 2011.

[68] J. C. GOWER. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3-4):325–338, Dec. 1966.

[69] A. Guiller, A. Bellido, and L. Madec. Genetic distances and ordination: the land snail Helix aspersa in north Africa as a test case. *Systematic Biology*, 47(2):208–27, June 1998.

[70] X. Hao, R. Jiang, and T. Chen. Clustering 16S rRNA for OTU prediction: a method of unsupervised Bayesian clustering. *Bioinformatics (Oxford, England)*, 27(5):611–8, Mar. 2011.

[71] 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, Oct. 1985.

[72] X. He, S. Chang, J. Zhang, Q. Zhao, H. Xiang, K. Kusonmano, L. Yang, Z. S. Sun, H. Yang, and J. Wang. MethyCancer: the database of human DNA methylation and cancer. *Nucleic Acids Research*, 36(Database issue):D836–41, Jan. 2008.

[73] T. A. Heath, M. T. Holder, and J. P. Huelsenbeck. A dirichlet process prior for estimating lineage-specific substitution rates. *Molecular Biology and Evolution*, 29(3):939–55, Mar. 2012.

[74] D. G. Higgins. Sequence ordinations: a multivariate analysis approach to analysing large sequence data sets. *Computer Applications in the Biosciences : CABIOS*, 8(1):15–22, Feb. 1992.

[75] M. S. Hill, A. L. Hill, J. Lopez, K. J. Peterson, S. Pomponi, M. C. Diaz, R. W. Thacker, M. Adamska, N. Boury-Esnault, P. Cárdenas, A. Chaves-Fonnegra, E. Danka, B.-O. De Laine, D. Formica, E. Hajdu, G. Lobo-Hajdu, S. Klontz, C. C. Morrow, J. Patel, B. Picton, D. Pisani, D. Pohlmann, N. E. Redmond, J. Reed, S. Richey, A. Riesgo, E. Rubin, Z. Russell, K. Rützler, E. a. Sperling, M. di Stefano, J. E. Tarver, and A. G. Collins. Reconstruction of family-level phylogenetic relationships within Demospongiae (Porifera) using nuclear encoded housekeeping genes. *PloS One*, 8(1):e50437, Jan. 2013.

[76] D. E. Holmes, K. P. Nevin, and D. R. Lovley. Comparison of 16S rRNA, nifD, recA, gyrB, rpoB and fusA genes within the family Geobacteraceae fam. nov. *International Journal of Systematic and Evolutionary Microbiology*, 54(Pt 5):1591–9, Sept. 2004.

[77] W. Huang, L. Li, J. R. Myers, and G. T. Marth. ART: a next-generation sequencing read simulator. *Bioinformatics (Oxford, England)*, 28(4):593–4, Feb. 2012.

[78] R. R. Hudson. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics (Oxford, England)*, 18(2):337–338, Feb. 2002.

[79] J. Huerta-Cepas, J. Dopazo, and T. Gabaldón. ETE: a python Environment for Tree Exploration. *BMC Bioinformatics*, 11:24, Jan. 2010.

[80] R. Hughey and A. Krogh. Hidden Markov models for sequence analysis: extension and analysis of the basic method. *Bioinformatics*, 12(2):95–107, Apr. 1996.

[81] S. Joly and A. Bruneau. Delimiting Species Boundaries in Rosa Sect. Cinnamomeae (Rosaceae) in Eastern North America. *Systematic Botany*, 32(4):819–836, Oct. 2007.

[82] C. R. Jukes, T. H. and Cantor. Evolution of protein molecules. In M. N. Munro, editor, *Mammalian protein metabolism*, pages 21–132. Academic Press, New York, 1969.

[83] U. Kõljalg, R. H. Nilsson, K. Abarenkov, L. Tedersoo, A. F. S. Taylor, M. Bahram, S. T. Bates, T. D. Bruns, J. Bengtsson-Palme, T. M. Callaghan, B. Douglas, T. Drenkhan, U. Eberhardt, M. Dueñas, T. Grebenc, G. W. Griffith, M. Hartmann, P. M. Kirk, P. Kohout, E. Larsson, B. D. Lindahl, R. Lücking, M. P. Martín, P. B. Matheny, N. H. Nguyen, T. Niskanen, J. Oja, K. G. Peay, U. Peintner, M. Peterson, K. Põldmaa, L. Saag, I. Saar, A. Schüß ler, J. A. Scott, C. Senés, M. E. Smith, A. Suija, D. L. Taylor, M. T. Telleria, M. Weiss, and K.-H. Larsson. Towards a unified paradigm for sequence-based identification of fungi. *Molecular Ecology*, 22(21):5271–7, Nov. 2013.

[84] D. A. Keim. Visual exploration of large data sets. *Communications of the ACM*, 44(8):38–44, Aug. 2001.

[85] C. T. Kelleher, T. R. Hodkinson, G. C. Douglas, and D. L. Kelly. Species distinction in Irish populations of Quercus petraea and Q. robur: morphological versus molecular analyses. *Annals of Botany*, 96(7):1237–46, Dec. 2005.

[86] 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, June 1980.

[87] 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, June 2001.

[88] J. Kuczynski, C. L. Lauber, W. A. Walters, L. W. Parfrey, J. C. Clemente, D. Gevers, and R. Knight. Experimental and analytical tools for studying the human microbiome. *Nature reviews. Genetics*, 13(1):47–58, Jan. 2012.

[89] M. Kullberg, M. a. Nilsson, U. Arnason, E. H. Harley, and A. Janke. Housekeeping genes for phylogenetic analysis of eutherian relationships. *Molecular Biology and Evolution*, 23(8):1493–503, Aug. 2006.

[90] B. Langmead and S. L. Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–9, Apr. 2012.

[91] G. F. Lawler. *Introduction to Stochastic Processes*. CRC Press, 2nd edition, 2006.

[92] F. Leliaert, H. Verbruggen, P. Vanormelingen, F. Steen, J. M. López-Bautista, G. C. Zuccarello, and O. De Clerck. DNA-based species delimitation in algae. *European Journal of Phycology*, 49(2):179–196, June 2014.

[93] R. E. Ley, F. Bäckhed, P. Turnbaugh, C. A. Lozupone, R. D. Knight, and J. I. Gordon. Obesity alters gut microbial ecology. *Proceedings of the National Academy of Sciences of the United States of America*, 102(31):11070–5, Aug. 2005.

[94] B. Liu, J. Yuan, S.-M. Yiu, Z. Li, Y. Xie, Y. Chen, Y. Shi, H. Zhang, Y. Li, T.-W. Lam, and R. Luo. COPE: an accurate k-mer-based pair-end reads connection tool to facilitate genome assembly. *Bioinformatics (Oxford, England)*, 28(22):2870–4, Nov. 2012.

[95] K. Liu, T. J. Warnow, M. T. Holder, S. M. Nelesen, J. Yu, A. P. Stamatakis, and C. R. Linder. SATe-II: very fast and accurate simultaneous estimation of multiple sequence alignments and phylogenetic trees. *Systematic Biology*, 61(1):90–106, Jan. 2012.

[96] S. Liu, K. Ji, J. Chen, D. Tai, W. Jiang, G. Hou, J. Chen, J. Li, and B. Huang. Panorama phylogenetic diversity and distribution of Type A influenza virus. *PloS One*, 4(3):e5022, Jan. 2009.

[97] Z. Liu, T. Z. DeSantis, G. L. Andersen, and R. Knight. Accurate taxonomy assignments from 16S rRNA sequences produced by highly parallel pyrosequencers. *Nucleic Acids Research*, 36(18):e120, Oct. 2008.

[98] A. Löytynoja, A. J. Vilella, and N. Goldman. Accurate extension of multiple sequence alignments using a phylogeny-aware graph algorithm. *Bioinformatics (Oxford, England)*, 28(13):1684–91, July 2012.

[99] I. Maccallum, D. Przybylski, S. Gnerre, J. Burton, I. Shlyakhter, A. Gnirke, J. Malek, K. McKernan, S. Ranade, T. P. Shea, L. Williams, S. Young, C. Nusbaum, and D. B. Jaffe. ALLPATHS 2: small genomes assembled accurately and with high continuity from short paired reads. *Genome Biology*, 10(10):R103, Jan. 2009.

[100] T. Magoč and S. L. Salzberg. FLASH: fast length adjustment of short reads to improve genome assemblies. *Bioinformatics (Oxford, England)*, 27(21):2957–63, Nov. 2011.

[101] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten. Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4):558–69, Jan. 1993.

[102] A. P. Masella, A. K. Bartram, J. M. Truszkowski, D. G. Brown, and J. D. Neufeld. PANDAseq: paired-end assembler for illumina sequences. *BMC Bioinformatics*, 13(1):31, Jan. 2012.

[103] 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, Jan. 2010.

[104] R. L. Mayden. A hierarchy ofspecies concepts: the denouement in the saga of the species problem. In H. D. M.F. Oaridge and M. Wilson., editors, *Species: The Units of Biodiversity*, pages 381–424. Chapman and Hall, London, 1997.

[105] D. McDonald, M. N. Price, J. Goodrich, E. P. Nawrocki, T. Z. DeSantis, A. Probst, G. L. Andersen, R. Knight, and P. Hugenholtz. An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *The ISME journal*, 6(3):610–8, Mar. 2012.

[106] T. C. Mendelson and K. L. Shaw. Sexual behaviour: rapid speciation in an arthropod. *Nature*, 433(7024):375–6, Jan. 2005.

[107] C. P. Meyer and G. Paulay. DNA barcoding: error rates based on comprehensive sampling. *PLoS Biology*, 3(12):e422, Dec. 2005.

[108] M. V. Modica, N. Puillandre, M. Castelin, Y. Zhang, and M. Holford. A good compromise: rapid and robust species proxies for inventorying biodiversity hotspots using the terebridae (gastropoda: conoidea). *PloS One*, 9(7):e102160, Jan. 2014.

[109] M. T. Monaghan, R. Wild, M. Elliot, T. Fujisawa, M. Balke, D. J. G. Inward, D. C. Lees, R. Ranaivosolo, P. Eggleton, T. G. Barraclough, and A. P. Vogler. Accelerated species inventory on Madagascar using coalescent-based models of species delineation. *Systematic Biology*, 58(3):298–311, June 2009.

[110] J. L. Morales and J. Nocedal. Remark on "algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization". *ACM Transactions on Mathematical Software*, 38(1):1–4, Nov. 2011.

[111] T. Munzner, F. Guimbretiere, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: scalable tree comparison using Focus+ Context with guaranteed visibility. *ACM Transactions on Graphics*, 22(3):453–462, 2003.

[112] R. G. Mustapha, M. N. Sulaiman, H. Ibrahim, and Norwati. A Survey: Clustering Ensembles Techniques. *World Academy of Science, Engineering and Technology*, 38, 2009.

[113] M. W. Nachman and S. L. Crowell. Estimate of the mutation rate per nucleotide in humans. *Genetics*, 156(1):297–304, Sept. 2000.

[114] K. Nakamura, T. Oshima, T. Morimoto, S. Ikeda, H. Yoshikawa, Y. Shiwa, S. Ishikawa, M. C. Linak, A. Hirai, H. Takahashi, M. Altaf-Ul-Amin, N. Ogasawara, and S. Kanaya. Sequence-specific error profile of Illumina sequencers. *Nucleic Acids Research*, 39(13):e90, July 2011.

[115] S. Nee. Inferring speciation rates from phylogenies. *Evolution.*, 55(4):661–8, Apr. 2001.

[116] 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):443–53, Mar. 1970.

[117] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, Jan. 1965.

[118] 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):4327–41, June 2011.

[119] K. C. Nixon and Q. D. Wheeler. An amplification of the phylogenetic species concept. *Cladistics*, 6(3):211–223, Sept. 1990.

[120] C. Notredame, D. G. Higgins, and J. Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 302(1):205–17, Sept. 2000.

[121] J. Padial, A. Miralles, I. D. la Riva, and M. Vences. Review: The integrative future of taxonomy. *Front Zool.*, pages 1–14, 2010.

[122] A. Papadopoulou, I. Anastasiou, and A. P. Vogler. Revisiting the insect mitochondrial molecular clock: the mid-Aegean trench calibration. *Molecular Biology and Evolution*, 27(7):1659–72, July 2010.

[123] G. a. Pavlopoulos, T. G. Soldatos, A. Barbosa-Silva, and R. Schneider. A reference guide for tree analysis and visualization. *BioData Mining*, 3(1):1, Jan. 2010.

[124] J. Pons, T. Barraclough, J. Gomez-Zurita, A. Cardoso, D. Duran, S. Hazell, S. Kamoun, W. Sumlin, and A. Vogler. Sequence-Based Species Delimitation for the DNA Taxonomy of Undescribed Insects. *Systematic Biology*, 55(4):595–609, Aug. 2006.

[125] J. R. Powell. Accounting for uncertainty in species delineation during the analysis of environmental DNA sequence data. *Methods in Ecology and Evolution*, 3(1):1–11, Feb. 2012.

[126] J. B. Procter, J. Thompson, I. Letunic, C. Creevey, F. Jossinet, and G. J. Barton. Visualization of multiple alignments, phylogenies and gene family evolution. *Nature Methods*, 7(3 Suppl):S16–25, Mar. 2010.

[127] E. Pruesse, J. Peplies, and F. O. Glöckner. SINA: accurate high-throughput multiple sequence alignment of ribosomal RNA genes. *Bioinformatics (Oxford, England)*, 28(14):1823–9, July 2012.

[128] N. Puillandre, a. Lambert, S. Brouillet, and G. Achaz. ABGD, Automatic Barcode Gap Discovery for primary species delimitation. *Molecular Ecology*, 21(8):1864–77, Apr. 2012.

[129] J. Qin, Y. Li, Z. Cai, S. Li, J. Zhu, F. Zhang, S. Liang, W. Zhang, Y. Guan, D. Shen, Y. Peng, D. Zhang, Z. Jie, W. Wu, Y. Qin, W. Xue, J. Li, L. Han, D. Lu, P. Wu, Y. Dai, X. Sun, Z. Li, A. Tang, S. Zhong, X. Li, W. Chen, R. Xu, M. Wang, Q. Feng, M. Gong, J. Yu, Y. Zhang, M. Zhang, T. Hansen, G. Sanchez, J. Raes, G. Falony, S. Okuda, M. Almeida, E. LeChatelier, P. Renault, N. Pons, J.-M. Batto, Z. Zhang, H. Chen, R. Yang, W. Zheng, S. Li, H. Yang, J. Wang, S. D. Ehrlich, R. Nielsen, O. Pedersen, K. Kristiansen, and J. Wang. A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, 490(7418):55–60, Oct. 2012.

[130] C. Quast, E. Pruesse, P. Yilmaz, J. Gerken, T. Schweer, P. Yarza, J. Peplies, and F. O. Glöckner. The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Research*, 41(Database issue):D590–6, Jan. 2013.

[131] S. Ratnasingham and P. D. N. Hebert. bold: The Barcode of Life Data System (http://www.barcodinglife.org). *Molecular Ecology Notes*, 7(3):355–364, May 2007.

[132] J. Ravel, P. Gajer, Z. Abdo, G. M. Schneider, S. S. K. Koenig, S. L. McCulle, S. Karlebach, R. Gorle, J. Russell, C. O. Tacket, R. M. Brotman, C. C. Davis, K. Ault, L. Peralta, and L. J. Forney. Vaginal microbiome of reproductive-age women. *Proceedings of the National Academy of Sciences of the United States of America*, 108 Suppl(Supplement_1):4680–4067, Mar. 2011.

[133] B. D. Redelings and M. a. Suchard. Joint Bayesian estimation of alignment and phylogeny. *Systematic Biology*, 54(3):401–18, June 2005.

[134] N. M. Reid and B. C. Carstens. Phylogenetic estimation error can decrease the accuracy of species delimitation: a Bayesian implementation of the general mixed Yule-coalescent model. *BMC Evolutionary Biology*, 12(1):196, Jan. 2012.

[135] L. J. Revell, L. J. Harmon, and D. C. Collar. Phylogenetic signal, evolutionary process, and rate. *Systematic Biology*, 57(4):591–601, Aug. 2008.

[136] G. M. Richard Durbin, Sean R. Eddy, Anders Krogh. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[137] S. Rodrigue, A. C. Materna, S. C. Timberlake, M. C. Blackburn, R. R. Malmstrom, E. J. Alm, and S. W. Chisholm. Unlocking short read sequencing for metagenomics. *PloS One*, 5(7):e11840, Jan. 2010.

[138] a. J. Roger, O. Sandblom, W. F. Doolittle, and H. Philippe. An evaluation of elongation factor 1 alpha as a phylogenetic marker for eukaryotes. *Molecular Biology and Evolution*, 16(2):218–33, Feb. 1999.

[139] F. Ronquist, M. Teslenko, P. van der Mark, D. L. Ayres, A. Darling, S. Höhna, B. Larget, L. Liu, M. A. Suchard, and J. P. Huelsenbeck. MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. *Systematic Biology*, 61(3):539–42, May 2012.

[140] S. T. Roweis. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, Dec. 2000.

[141] J. W. Sahl, M. N. Matalka, and D. a. Rasko. Phylomark, a tool to identify conserved phylogenetic markers from whole-genome alignments. *Applied and Environmental Microbiology*, 78(14):4884–92, July 2012.

[142] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–25, July 1987.

[143] J. Sammon Jr. A nonlinear mapping for data structure analysis. *Computers, IEEE Transactions on computers*, 100(5):401–409, 1969.

[144] M. J. Sanderson. r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. *Bioinformatics (Oxford, England)*, 19(2):301–302, Jan. 2003.

[145] R. Santamaría and R. Therón. Treevolution: visual analysis of phylogenetic trees. *Bioinformatics (Oxford, England)*, 25(15):1970–1, Aug. 2009.

[146] J. Sauer and B. Hausdorf. A comparison of DNA-based methods for delimiting species in a Cretan land snail radiation reveals shortcomings of exclusively molecular taxonomy. *Cladistics*, 28:300–316, 2012.

[147] V. Savolainen, M. W. Chase, S. B. Hoot, C. M. Morton, D. E. Soltis, C. Bayer, M. F. Fay, a. Y. de Bruijn, S. Sullivan, and Y. L. Qiu. Phylogenetics of flowering plants based on combined analysis of plastid atpB and rbcL gene sequences. *Systematic Biology*, 49(2):306–62, June 2000.

[148] 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, Oct. 2005.

[149] P. D. Schloss and S. L. Westcott. Assessing and improving methods used in operational taxonomic unit-based approaches for 16S rRNA gene sequence analysis. *Applied and Environmental Microbiology*, 77(10):3219–26, May 2011.

[150] P. D. Schloss, S. L. Westcott, T. Ryabin, J. R. Hall, M. Hartmann, E. B. Hollister, R. A. Lesniewski, B. B. Oakley, D. H. Parks, C. J. Robinson, J. W. Sahl, B. Stres, G. G. Thallinger, D. J. Van Horn, and C. F. Weber. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology*, 75(23):7537–41, Dec. 2009.

[151] S. Schneider and L. Excoffier. Estimation of past demographic parameters from the distribution of pairwise differences when the mutation rates vary among sites: application to human mitochondrial DNA. *Genetics*, 152(3):1079–89, July 1999.

[152] J. W. Sites and J. C. Marshall. Delimiting species: a Renaissance issue in systematic biology. *Trends in Ecology & Evolution*, 18(9):462–470, Sept. 2003.

[153] J. W. Sites and J. C. Marshall. Operational Criteria for Delimiting Species. *Annual Review of Ecology, Evolution, and Systematics*, 35(1):199–227, Dec. 2004.

[154] G. J. D. Smith, D. Vijaykrishna, J. Bahl, S. J. Lycett, M. Worobey, O. G. Pybus, S. K. Ma, C. L. Cheung, J. Raghwani, S. Bhatt, J. S. M. Peiris, Y. Guan, and A. Rambaut. Origins and evolutionary genomics of the 2009 swine-origin H1N1 influenza A epidemic. *Nature*, 459(7250):1122–5, June 2009.

[155] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–7, Mar. 1981.

[156] P. H. A. Sokal, R. R. and Sneath. *Numerical Taxonomy*. W.H. Freeman and Co., San Francisco, CA, 1963.

[157] L. Soldati, G. J. Kergoat, A.-L. Clamens, H. Jourdan, R. Jabbour-Zahab, and F. L. Condamine. Integrative taxonomy of New Caledonian beetles: species delimitation and definition of the Uloma isoceroides species group (Coleoptera, Tenebrionidae, Ulomini), with the description of four new species. *ZooKeys*, (415):133–67, Jan. 2014.

[158] E. STACKEBRANDT and B. M. GOEBEL. Taxonomic Note: A Place for DNA-DNA Reassociation and 16S rRNA Sequence Analysis in the Present Species Definition in Bacteriology. *International Journal of Systematic Bacteriology*, 44(4):846–849, Oct. 1994.

[159] J. E. Stajich, D. Block, K. Boulez, S. E. Brenner, S. A. Chervitz, C. Dagdigian, G. Fuellen, J. G. R. Gilbert, I. Korf, H. Lapp, H. Lehväslaiho, C. Matsalla, C. J. Mungall, B. I. Osborne, M. R. Pocock, P. Schattner, M. Senger, L. D. Stein, E. Stupka, M. D. Wilkinson, and E. Birney. The Bioperl toolkit: Perl modules for the life sciences. *Genome Research*, 12(10):1611–8, Oct. 2002.

[160] A. Stamatakis. RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics (Oxford, England)*, 22(21):2688–90, Nov. 2006.

[161] A. Stamatakis. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics (Oxford, England)*, pages btu033–, Feb. 2014.

[162] A. Stamatakis, P. Hoover, and J. Rougemont. A rapid bootstrap algorithm for the RAxML Web servers. *Systematic Biology*, 57(5):758–71, Oct. 2008.

[163] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[164] Y. Sun, Y. Cai, S. M. Huse, R. Knight, W. G. Farmerie, X. Wang, and V. Mai. A large-scale benchmark study of existing algorithms for taxonomy-independent microbial community analysis. *Briefings in Bioinformatics*, 13(1):107–21, Jan. 2012.

[165] B. Surina and I. Dakskobler. Delimitation of the alliances Caricion firmae (Seslerietalia albicantis) and Seslerion juncifoliae (Seslerietalia juncifoliae) in the southeastern Alps and Dinaric mountains. *Plant Biosystems*, 139(3):399–410, Nov. 2005.

[166] L. Tancioni, T. Russo, S. Cataudella, V. Milana, A. K. Hett, E. Corsi, and A. R. Rossi. Testing species delimitations in four Italian sympatric leuciscine fishes in the Tiber River: a combined morphological and molecular approach. *PloS One*, 8(4):e60392, Jan. 2013.

[167] D. Tautz, P. Arctander, A. Minelli, R. H. Thomas, and A. P. Vogler. A plea for DNA taxonomy. *Trends in Ecology & Evolution*, 18(2):70–74, Feb. 2003.

[168] H. R. Taylor and W. E. Harris. An emergent science on the brink of irrelevance: a review of the past 8 years of DNA barcoding. *Molecular Ecology Resources*, 12(3):377–88, May 2012.

[169] Y. I. Tekle. DNA Barcoding in Amoebozoa and Challenges: The Example of Cochliopodium. *Protist*, 165(4):473–484, May 2014.

[170] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–23, Dec. 2000.

[171] J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–80, Nov. 1994.

[172] S. S. Tobe, A. C. Kitchener, and A. M. T. Linacre. Reconstructing mammalian phylogenies: a detailed comparison of the cytochrome B and cytochrome oxidase subunit I mitochondrial genes. *PloS One*, 5(11):e14156, Jan. 2010.

[173] A. Velasco-Castrillón, T. J. Page, J. A. E. Gibson, and M. I. Stevens. Surprisingly high levels of biodiversity and endemism amongst Antarctic rotifers uncovered with mitochondrial DNA. *Biodiversity*, pages 1–13, July 2014.

[174] N. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854, 2010.

[175] a. P. Vogler and M. T. Monaghan. Recent advances in DNA taxonomy. *Journal of Zoological Systematics and Evolutionary Research*, 45(1):1–10, Feb. 2007.

[176] D. V. Volokhov, A. a. Neverov, J. George, H. Kong, S. X. Liu, C. Anderson, M. K. Davidson, and V. Chizhikov. Genetic analysis of housekeeping genes of members of the genus Acholeplasma: phylogeny and complementary molecular markers to the 16S rRNA gene. *Molecular Phylogenetics and Evolution*, 44(2):699–710, Aug. 2007.

[177] L. Vuataz, M. Sartori, A. Wagner, and M. T. Monaghan. Toward a DNA taxonomy of Alpine Rhithrogena (Ephemeroptera: Heptageniidae) using a mixed Yule-coalescent analysis of mitochondrial and nuclear DNA. *PloS One*, 6(5):e19728, Jan. 2011.

[178] J. Wakeley. *Coalescent Theory: An Introduction*. Roberts and Company Publishers, 2009.

[179] Q. Wang, G. M. Garrity, J. M. Tiedje, and J. R. Cole. Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology*, 73(16):5261–7, Aug. 2007.

[180] Q. Wang, G. M. Garrity, J. M. Tiedje, and J. R. Cole. Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology*, 73(16):5261–7, Aug. 2007.

[181] Z. Wang, B. Fang, J. Chen, X. Zhang, Z. Luo, L. Huang, X. Chen, and Y. Li. De novo assembly and characterization of root transcriptome using Illumina paired-end sequencing and development of cSSR markers in sweet potato (Ipomoea batatas). *BMC Genomics*, 11:726, Jan. 2010.

[182] J. J. Werner, D. Zhou, J. G. Caporaso, R. Knight, and L. T. Angenent. Comparison of Illumina paired-end and single-direction sequencing for microbial 16S rRNA gene amplicon surveys. *The ISME Journal*, 6(7):1273–6, July 2012.

[183] 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, Nov. 1977.

[184] D. S. W.R. Gilks, S. Richardson. *Markov Chain Monte Carlo in Practice*. CRC Press, 1995.

[185] Z. Yang. Estimating the pattern of nucleotide substitution. *Journal of Molecular Evolution*, 39(1), July 1994.

[186] Z. Yang. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *Journal of Molecular Evolution*, 39(3):306–314, Sept. 1994.

[187] Z. Yang. *Computational Molecular Evolution*. Oxford University Press, 2006.

[188] Z. Yang and B. Rannala. Bayesian species delimitation using multilocus sequence data. *Proceedings of the National Academy of Sciences of the United States of America*, 107(20):9264–9, May 2010.

[189] D. W. Yu, Y. Ji, B. C. Emerson, X. Wang, C. Ye, C. Yang, and Z. Ding. Biodiversity soup: metabarcoding of arthropods for rapid biodiversity assessment and biomonitoring. *Methods in Ecology and Evolution*, 3(4):613–623, Aug. 2012.

[190] L. Zaslavsky, Y. Bao, and T. a. Tatusova. Visualization of large influenza virus sequence datasets using adaptively aggregated trees with sampling-based subscale representation. *BMC Bioinformatics*, 9:237, Jan. 2008.

[191] J. Zhang, P. Kapli, P. Pavlidis, and A. Stamatakis. A general species delimitation method with applications to phylogenetic placements. *Bioinformatics (Oxford, England)*, 29(22):2869–76, Nov. 2013.

[192] J. Zhang, K. Kobert, T. Flouri, and A. Stamatakis. PEAR: a fast and accurate Illumina Paired-End reAd mergeR. *Bioinformatics (Oxford, England)*, pages 1–7, Nov. 2013.

[193] J. Zhang, A. M. Mamlouk, T. Martinetz, S. Chang, J. Wang, and R. Hilgenfeld. PhyloMap: an algorithm for visualizing relationships of large sequence data sets and its application to the influenza A virus genome. *BMC Bioinformatics*, 12:248, Jan. 2011.

[194] J. Zhang and A. Stamatakis. The Multi-Processor Scheduling Problem in Phylogenetics. *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, pages 691–698, May 2012.

[195] H.-W. Zhou, D.-F. Li, N. F.-Y. Tam, X.-T. Jiang, H. Zhang, H.-F. Sheng, J. Qin, X. Liu, and F. Zou. BIPES, a cost-effective high-throughput method for assessing microbial diversity. *The ISME Journal*, 5(4):741–9, Apr. 2011.

[196] E. a. Zimmer and J. Wen. Using nuclear gene data for plant phylogenetics: progress and prospects. *Molecular Phylogenetics and Evolution*, 65(2):774–85, Nov. 2012.