

Freie Universität Berlin

Doctoral Thesis

---

**What is in my Sample? – Challenges and  
Approaches for Unveiling the Hidden  
Diversity in Plankton Samples**

---

*Dissertation zur Erlangung des Grades eines Doktors der  
Naturwissenschaften (Dr. rer. nat.) vorgelegt von*

Marie Hoffmann

*am Fachbereich*

Mathematik und Informatik  
der Freien Universität Berlin

Berlin, 2022

*Tag der Disputation: 17.05.2022*

*Erstgutachter: **Prof. Dr. Knut Reinert**, Freie Universität Berlin*

*Zweitgutachter: **Prof. Dr. Michael T. Monaghan**, Freie Universität  
Berlin und Leibniz-Institut für Gewässerökologie und Binnenfischerei*

FREIE UNIVERSITÄT BERLIN

## *Abstract*

Mathematik und Informatik  
der Freien Universität Berlin

Doctor rerum naturalium

### **What is in my Sample? – Challenges and Approaches for Unveiling the Hidden Diversity in Plankton Samples**

by Marie Hoffmann

#### **English**

The goal of this work is to deploy a primer search tool (PriSeT) suitable for taxonomically broad, sparse, and uncurated datasets of reference sequences. I discuss the theoretical and practical challenges when sequence datasets are large. Uncurated online reference databases are often the only source available when designing new primer sequences, studying the effectiveness, and for species identification. As a case study, two different identification methods for planktonic microorganisms from freshwater samples are presented: identification via light microscopy and DNA sequencing. The sequencing approach will replace the manual method to a large extent, but still needs to be improved and requires many costly trial-and-error iterations. The robust primer search tool PriSeT is here developed and designed to shorten the optimization time and to facilitate new types of *in silico* analyses. I evaluate PriSeT on 18S rRNA genes from all major plankton clades and on whole RNA genomes. The resulting primer sequences are compared to published primer pairs. Finally, the workflow of an academic research group planning and conducting metabarcoding experiments is critically reviewed. I present a database schema designed to summarize key information and enable researchers to be more productive in less time. The scheme also alleviates new types of meta-analysis that are not possible when data are scattered, such as quantitative and qualitative comparisons between different studies.

#### **Deutsch**

In diesem Beitrag entwerfe ich ein Primer-Suchwerkzeug (PriSeT), das sich für taxonomisch breite, aber unkuratierte Sequenzdatensätze eignet. Ich erörtere die theoretischen und praktischen Herausforderungen, die damit verbunden sind. Große und unkuratierte Online-Sequenzdatenbanken sind oft die einzige verfügbare Quelle um neue Primer-Sequenzen zu entwickeln, ihre Wirksamkeit zu untersuchen oder Arten zu identifizieren. In einer Fallstudie betrachte ich zwei verschiedene Identifizierungsmethoden für Plankton aus Süßwasserproben: Identifizierung mit dem Lichtmikroskop und DNA-Sequenzierung. Der letztere Ansatz wird die manuelle Methode weitgehend ersetzen, erfordert aber viele kostspielige Iterationen. Das robuste Primersuchwerkzeug PriSeT wurde entwickelt, um die Zeit für die Sequenzoptimierung zu verkürzen und neue Arten von *in silico* Analysen zu ermöglichen. Ich evaluiere PriSeT an 18S rRNA-Genen aus allen wichtigen Planktonkladen und an ganzen RNA-Genomen. Die berechneten Primer-Sequenzen werden mit veröffentlichten Primer-Paaren verglichen. Schließlich wird der Arbeitsablauf einer akademischen Forschungsgruppe, die Metabarcoding-Experimente plant und durchführt, kritisch betrachtet. Ich stelle ein Datenbankschema vor, das die wichtigsten Informationen zusammenfasst und es den Forschern ermöglicht, in kürzerer Zeit produktiver zu arbeiten. Das Schema erleichtert auch neue Arten von Meta-Analysen, die nicht möglich sind, wenn die Daten verstreut sind.





## *Acknowledgements*

I would like to thank both supervisors for giving me the freedom to venture into waters that were closest to my heart. I thank Michael Monaghan for the time we spent in fruitful conversations, and Knut Reinert for his hospitality and algorithmic advice to make PriSeT fast and cool. Tatiana Semenova-Nelson inspired me to tackle the problem of primer search for metabarcoding experiments, Felix Heeger introduced me to the mysterious world of metabarcoding, Camilla Mazzoni, Justyna Wollinska, and Rita Adrian shared their expertise, Ursula Newen explained in great detail the sampling protocol for Lake Müggelsee and River Spree. In addition, I would like to thank all members of the SeqAn team for further developing my C++ coding skills.



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Definition . . . . .	1
1.2 Contribution . . . . .	3
1.3 Thesis Outline . . . . .	3
<b>2 Species Identification in Environmental Samples</b>	<b>5</b>
2.1 Introduction . . . . .	5
2.2 Tree of Life . . . . .	6
2.2.1 Identification . . . . .	7
2.2.2 Tree of Life Construction	9
Distance-Matrix Methods . . . . .	10
Maximum Parsimony . . . . .	10
Maximum Likelihood . . . . .	10
2.2.3 About Sequence Distances . . . . .	10
2.2.4 Multiple Sequence Alignment (MSA) . . . . .	11
2.2.5 NP-Completeness of $MSA_{Sp}$ . . . . .	16
2.3 Polymerase Chain Reaction (PCR) . . . . .	21
2.3.1 Principle . . . . .	21
2.3.2 Primer Design . . . . .	23
2.3.3 Reverse-Transcriptase qPCR . . . . .	25
2.4 Challenges in Species Identification of Plankton Samples . . . . .	27
2.4.1 Microscopic Method . . . . .	27
2.4.2 Metabarcoding Method . . . . .	29
2.5 The Lake Müggelsee Long-Term Monitoring Project . . . . .	31
2.5.1 Study Motivation . . . . .	31
2.5.2 Materials and Methods . . . . .	32
2.5.3 Results . . . . .	38
<b>3 Primer Discovery in Large Datasets</b>	<b>53</b>
3.1 Problem Definition . . . . .	53
3.2 Introduction . . . . .	54
3.3 Starting with an MSA . . . . .	56
3.4 Limitations of Existing Primer Search Tools . . . . .	58
3.5 Idea . . . . .	59
3.6 Indexing Data Structure . . . . .	61
3.6.1 FM-Index . . . . .	62
3.7 Space-Efficient $K$ -Mer Representation . . . . .	67
3.8 Bit Vectors for $K$ -Mer Location Encoding . . . . .	69
3.9 Algorithm . . . . .	71

3.9.1	FM-Index Step	71
3.9.2	FM Frequency Step	71
3.9.3	K-Mer Transform and Filter Step	72
3.9.4	Combining K-mers into Pairs	78
3.10	Theoretical Runtimes and Space Occupation	80
3.11	Example Applications	81
3.12	Plankton	81
3.12.1	Data Set for Plankton	81
3.12.2	Verification of Published Primer Pairs	81
3.12.3	<i>De Novo</i> Computation	84
3.12.4	Performance on Plankton Data Set	86
3.13	SARS-CoV-2	89
3.13.1	Genome and Functional Units	89
3.13.2	Dataset for SARS-CoV-2	90
3.13.3	<i>De Novo</i> Computation	90
<b>4</b>	<b>A Database for Metabarcoding Experiments</b>	<b>93</b>
4.1	Problem Statement	93
4.2	Motivation	95
4.2.1	Short- and Long-Term Staff	95
4.2.2	Text Documents	97
4.2.3	Workflow and Goal	97
4.3	Aspects and Principles	99
4.4	Entities and Workflow	99
4.4.1	Samples	99
4.4.2	Morphological Analysis	100
4.4.3	Metabarcoding Analysis	100
4.4.4	Bioinformatics Pipeline	101
4.5	Data and Knowledge Flow	101
4.5.1	Organizational Structure	101
4.5.2	Sample Data	101
4.5.3	Morphological Analysis	102
4.5.4	Metabarcoding Analysis	103
4.5.5	Bioinformatics Pipeline	104
4.5.6	Additional Requirements	104
4.5.7	Why a Relational Database System?	104
4.6	Database Schema	107
4.6.1	Table Definitions and Relations	109
4.6.2	Roles	127
<b>5</b>	<b>Discussion</b>	<b>129</b>
5.1	Metabarcoding	129
5.1.1	Taxonomic Sparseness of Sequence Libraries	129
5.1.2	Tree of Life Construction	130
5.2	PriSeT	132
5.2.1	Open Source Code	133
5.2.2	Performance	133
5.3	A Database for Metabarcoding Experiments	135
5.3.1	What to Keep	136
<b>6</b>	<b>Conclusion</b>	<b>137</b>

<b>A</b>	<b>Species Identification in Environmental Samples</b>	<b>139</b>
A.1	Abundance Plots . . . . .	139
A.2	R Code for Statistical Analyses . . . . .	140
A.2.1	NMDS in R . . . . .	141
A.2.2	MRPP in R . . . . .	143
<b>B</b>	<b>Primer Discovery in Large Datasets</b>	<b>147</b>
B.1	One-Letter Encodings of Nucleotides . . . . .	147
B.2	Runtimes for Plankton Datasets . . . . .	148
B.3	Published SARS-CoV-2 Real-time RT-PCR Primers . . . . .	149
B.4	Primer Pairs computed by PriSeT for SARS-CoV-2 . . . . .	150
<b>C</b>	<b>A Database for Metabarcoding Experiments</b>	<b>153</b>
C.1	Trigger Definition . . . . .	153
C.2	Example Queries in SQL . . . . .	153
	<b>Bibliography</b>	<b>159</b>
	<b>Selbstständigkeitserklärung</b>	<b>169</b>



# List of Figures

2.1	Complementing sample processing pipelines for monitoring projects .	6
2.2	Tree of life of Bacteria, Archaea, and Eukaryota based on ribosomal RNA genes . . . . .	7
2.3	Eukaryotic tree of life based on phylogeny . . . . .	9
2.4	Example of a multiple sequence alignment . . . . .	12
2.5	Possible events supported by the sequences . . . . .	13
2.6	Thermocycler . . . . .	21
2.7	First cycle of a PCR . . . . .	22
2.8	Critical self-annealing patterns . . . . .	25
2.9	Nikon Diaphot 200/300 Inverted Microscope . . . . .	27
2.10	Utermöhl counting chambers . . . . .	28
2.11	Various genera of Cladocera . . . . .	28
2.12	Integrated water sampler . . . . .	33
2.13	Rarefaction curves obtained for plankton OTUs for the three primer pairs (EUK15, EUK14, DIV4) . . . . .	37
2.14	Abundances by method, location, and higher-order groups . . . . .	41
2.15	Abundance plots of four of the most abundant species according to the morphological identification . . . . .	45
2.16	The 25 most abundant organisms and OTUs for each method: morphological identification and metabarcoding with one of the three markers (EUK15, EUK14, and DIV4) . . . . .	48
2.17	Statistical analysis of sample composition and sampling site . . . . .	49
3.1	Example of 6-mers contained in a text . . . . .	55
3.2	K-mer compression scheme . . . . .	68
3.3	Encoding of TKMerID locations . . . . .	70
3.4	Possible combinations of <i>k</i> -mers encoded in two TKMerIDs . . . . .	78
3.5	Runtimes for all clade data sets broken down to FM frequency computation, filter, and combine step . . . . .	88
3.6	K-Mers for all clade data sets counted after the FM frequency computation, filter, and combine steps . . . . .	89
3.7	Taxonomy of Orthocoronavirinae to subgenus level . . . . .	90
3.8	Top: Genome organization and functional domains of SARS-CoV-2. Bottom: Transcript positions for <i>de novo</i> primers . . . . .	92
4.1	Data transfer between lab operators and bioinformaticians . . . . .	98
4.2	Example of how sample metadata is documented and communicated .	102
4.3	Working document for storing counting results . . . . .	103
4.4	Data temperature and storage media . . . . .	105
4.5	Components of a full RDBMS . . . . .	106
4.6	Database schema for storing and relating experiments on plankton samples . . . . .	110

5.1	Growth of nucleotides and sequences in GenBank between 1982 and 2021 . . . . .	130
5.2	Critical annealing patterns missed by two online tools . . . . .	133
A.1	Species abundances as individual (Morph) or read count abundances (part 1) . . . . .	139
A.2	Species abundances as individual (Morph) or read count abundances (part 2) . . . . .	140



# List of Tables

2.1	A supersequence of $B_1$ , $B_2$ , and $B_3$ .	17
2.2	Scoring scheme for proof of Theorem 2.1.	19
2.3	Chemical constraints for conventional PCR	25
2.4	Chemical constraints for RT-PCR	26
2.5	Samples collected in autumn 2014 from lake and river sites	33
2.6	List of high-abundant species	34
2.7	Primers for metabarcoding on freshwater plankton samples	35
2.8	Number of taxa (Morph ID) and OTUs (molecular ID) found and identified to genus or species level	38
2.9	Results of <i>in silico</i> PCR for each primer set tested	39
2.10	Plankton community characterization using morphological assessment and DNA metabarcoding with three different primer sets (EUK15, EUK14, DIV4)	43
2.11	Most abundant species as seen under the light microscope and indication whether one of the marker-based approaches identified the species or other species of the same genus	44
3.1	Theoretical runtimes for an MSA-based primer search and barcode suitability check	57
3.2	Summary of available primer search tools	59
3.3	Example of a suffix array $A$ over the text $T = \text{ACGTACGT}$	64
3.4	Derivation of FM-index helper structures	65
3.5	Two-bit encoding of single nucleotides	68
3.6	Bit-parallelized counting of cytosine and guanine bases on the sequence ACGT	73
3.7	Closed or connected annealing patterns produce blocks of set bits	76
3.8	Bit-parallelized counting of self- or cross-annealing basepairs	76
3.9	Runtime classes and space occupation	80
3.10	Data set used for the primer verification test, the <i>de novo</i> search, and the runtime analysis	82
3.11	Selected primer sequences for the verification experiment	83
3.12	Filter settings in PriSeT for the primer verification experiment and the <i>de novo</i> primer discovery on the plankton data sets	84
3.13	Primer presence test for established primer pairs	85
3.14	<i>De novo</i> computation and statistical comparison with published primers ranked by reference coverage (left) or amplicon variation (right)	87
3.15	Filter Settings in PriSeT for <i>de novo</i> primer discovery on SARS-CoV-2 genomes	91
4.1	Categories of SQL commands and associated user roles	106
4.2	Unnormalized experiment table with primer and experiment names as a composite primary key	107
4.3	A table storing primer sequences	108

4.4	PostgreSQL datatypes used in the schema definition . . . . .	111
4.5	PostgreSQL constraints used in the schema definition . . . . .	111
4.6	Within the eukaryotic empire, the species <i>Closterium acerosum</i> has different lineages, because of different taxonomies in each database .	122
4.7	Proposed role assignment of lab members . . . . .	127
A.1	Layout for statistical analyses . . . . .	140
B.1	One-letter encodings for (ambiguous) DNA bases . . . . .	147
B.2	Runtimes separated by major computation steps . . . . .	148
B.3	Published Real-time RT-PCR primers for SARS-CoV-2 . . . . .	149

# List of Abbreviations

<b>ACID</b>	Atomicity, consistency, isolation, and durability
<b>AT</b>	Adenine or tyrosine occurrences in a DNA molecule
<b>AT-count</b>	Amount of adenine and tyrosine in a DNA molecule
<b>BLAST</b>	Basic Local Alignment Search Tool
<b>BW</b>	Burrows-Wheeler transform
<b>DNA</b>	Deoxyribonucleic acid
<b>eDNA</b>	Environmental deoxyribonucleic acid
<b>esample</b>	Environmental sample
<b>eTOL</b>	Eukaryotic tree of life
<b>FM-index</b>	Full-text substring index
<b>GC</b>	Guanine or cytosine occurrences in a DNA molecule
<b>GC-count</b>	Amount of guanine and cytosine in a DNA molecule
<b>GC-content</b>	Percentage of guanine and cytosine in a DNA molecule
<b>GenBank</b>	An open access sequence database
<b>GF</b>	Glass fiber
<b>GPU</b>	Graphics processing unit
<b>HDD</b>	Hard disk drive
<b>k-mer</b>	Sequence of fixed length $k \in \mathbb{N}$ over an arbitrary alphabet
<b>LAMP</b>	Loop-mediated isothermal amplification
<b>LCA</b>	Lowest common ancestor
<b>LIMS</b>	Laboratory information management system
<b>LM</b>	Light microscopy
<b>LUCA</b>	Last universal common ancestor
<b>Morph ID</b>	Morphological identification
<b>MPS, MPS, MS3, MPO, MPU</b>	Sampling sites at Lake Müggelsee
<b>MRPP</b>	Multi-response permutation procedure
<b>MS Excel, MS Word</b>	Microsoft Excel and Microsoft Word
<b>MSA</b>	Multiple sequence alignment
<b>NCBI</b>	National Center for Biotechnology Information
<b>NGS</b>	Next-generation sequencing
<b>NMDS</b>	Non-metric multidimensional scaling
<b>nt</b>	Nucleotide
<b>NP</b>	Nondeterministic polynomial-time (complexity class)
<b>OTU</b>	Operational taxonomic unit
<b>PCR</b>	Polymerase chain reaction
<b>PriSeT</b>	<i>Primer Search Tool</i>
<b>qPCR</b>	Real-time polymerase chain reaction
<b>RAM</b>	Random-access memory
<b>(R)DBMS</b>	(Relational) database management system
<b>rRNA</b>	Ribosomal ribonucleic acid
<b>RT-PCR</b>	Reverse transcription polymerase chain reaction
<b>SA</b>	Suffix array

<b>SARS-CoV-2</b>	Severe acute respiratory syndrome coronavirus 2
<b>SCS</b>	Shortest common supersequence problem
<b>SGT, SNZ</b>	Two sampling sites at River Spree
<b>SQL</b>	Structured query language
<b>SSD</b>	Solid-state drive
<b>SSU</b>	Small subunit (ribosome)
<b>TOL</b>	Tree of life
<b>URL</b>	Uniform resource locator
<b>WGS</b>	Whole-genome sequencing
<b>WMGS</b>	Whole metagenome shotgun (sequencing)

# List of Symbols

$\mathcal{B}$	set of bit vectors	$\{\{0,1\}^*\}^+$
$\mathcal{S}$	set of sequences	$\{\Sigma^+\}^+$
$\sigma_i$	$i$ -th symbol from alphabet $\Sigma$	
$C_s, C_p$	set of constraints for $k$ -mers ( $s$ ) and pairs ( $p$ )	
$Z, \zeta$	absolute and relative frequency cutoff for $k$ -mer occurrences	$\mathbb{N}, \mathbb{Q}$
$\kappa_{\min}$	minimum $k$ -mer length	nt
$\kappa_{\max}$	maximum $k$ -mer length	nt
$\mathcal{O}(\cdot)$	big O notation	
$T_{\min}$	minimum melting temperature	$^{\circ}\text{C}$
$T_{\max}$	maximum melting temperature	$^{\circ}\text{C}$
$\Delta T_m$	difference in melting temperatures	K
$\tau_{\min}$	minum amplicon length	nt
$\tau_{\max}$	maximum amplicon length	nt



*À mon petit singe qui a le pouvoir de resoudre les nœuds par  
l'épée d'humour.*





## Chapter 1

# Introduction

Quis custodiet ipsos custodes?

*Juvenal (Satire VI, lines 347-348)*

More than one quarter of all investigated animal and plant groups are endangered according to the IPBES<sup>1</sup> assessment report from 2019. It is estimated that within the next decades, about one million species will become extinct. Economically growth-driven, humans massively intervene in global ecosystems by agriculture, deforestation, mining, interruption of migration paths, direct exploitation of organisms, climate change, pollution, and the introduction of invasive alien species. The ongoing biological annihilation is labeled as the Anthropocene or sixth mass extinction. Ceballos, Ehrlich, and Dirzo, 2017 studied the population sizes of 27,600 vertebrate species (about half of the known vertebrate species) and found that 32 % are decreasing in population size and geographic range. All 177 analyzed mammals have lost at least 30 % of their geographic ranges, and more than 40 % of these mammal species have experienced a population shrinkage of 80 % or more. Population numbers and geographical ranges are only one aspect. As most organisms, including humans, live in communities, there are adverse cascading effects on the shared ecosystem, promoting further decline. Vos et al., 2014 estimated that the current extinction rates are 1,000 times higher than natural background rates of extinction and that future rates are likely to be 10,000 times higher.

In contrast, our effort to understand the ecosystem, its components, and dynamics is much too slow-paced because of a chronic shortage in funding of non-commercial research projects. Monitoring, analyzing, and understanding what covers the scientific aspect and is needed to convince decision-makers. So far, proportionally less attention has been paid to study the diversity of life – our resource.

As Edward O. Wilson put it at a commencement speech in 2011<sup>2</sup>:

*“It is that the 21st Century is going to be the Century of the Environment worldwide, and in science it is going to be the Century of Biology. The reason is simply that this is the time we either will settle down as a species or completely wreck the planet.”*

## 1.1 Problem Definition

Metabarcoding of environmental DNA (eDNA) is a procedure to determine the species composition of a sample. The focus is not on a single species, but the simultaneous

<sup>1</sup>Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services

<sup>2</sup>The speech was held at University of North Carolina, Chapel Hill (2011). Source: <https://whatrocks.github.io/commencement-db/2011-edward-o-wilson-university-of-north-carolina,-chapel-hill/>

identification of many species. It is becoming the predominant tool in the set of environmental monitoring methods. Water, soil, and even air can be sampled, the DNA extracted, amplified via polymerase chain reaction (PCR), sequenced using next-generation sequencing (NGS) methods, and computationally processed. The metabarcoding method allows for quick assessment of species compositions *in situ* and is the foundation for modeling dynamic processes of the environment over time and space.

Environmental samples (*esamples*) can comprise hundreds of organisms that may be closely or very distantly related in the taxonomic tree of life. DNA-based identification techniques of species mixtures encounter difficulties known to single organisms DNA analysis like repeats or high intra-species variations and problems related to the taxonomic heterogeneity of the sample and a missing *ground truth*<sup>3</sup>.

Currently, we are far from having solved these challenges - human experts for plankton identification are still irreplaceable. These experts are not only consulted for plain identification, but abundance and biomass estimation, or discovery of teratological forms<sup>4</sup>, which are indicators for stress and altered environmental conditions.

If the task of identification could be accomplished by an established metabarcoding protocol, the few available experts would have more resources to work on tasks that will remain infeasible for metabarcoding. The identifiability of species via metabarcoding is inherently linked to

- (i) the choice of the DNA *barcode*<sup>5</sup> and therefore the primer sequences,
- (ii) the effectiveness of the polymerase chain reaction (PCR),
- (iii) the tree of life (TOL) model as any identification method follows the hierarchical tree structure, and
- (iv) the availability of reference sequences for assigning operational taxonomic units (OTUs) as a product of a sequence processing pipeline to a taxonomic node.

To allow a high phylogenetic resolution, ideally, each species of interest would have a unique DNA barcode flanked by highly conserved regions that can serve as primer binding sites. Due to the species richness, this is impossible to reach. In a sample, we will find species being genetically indistinguishable in their barcodes and species being so distant that no single marker-based method would be able to capture both. So far, barcodes or new primer sequences are determined by analyzing a few genomes or sequences manually. Alternatively, already published primer pairs are taken that have been evaluated on similar *esamples* and modified. There exists no tool that operates fully automatically on an uncurated reference data set to propose new primer sequences occurring at high frequency.

Metabarcoding experiments are carried out by multiple groups of people for months and even years: lab processors, department and group heads, students, and researchers. It needs many iterations of evaluation and feedback to find a set of primer pairs that capture best a sample's composition. As the diversity is highly dynamic and undergoes seasonal cycles, but is as well driven by environmental changes, this process may always be ongoing. It is, therefore, of utmost importance to keep feedback loops short. A significant obstacle for data analyzing researchers to

<sup>3</sup>With *ground truth* we delineate information provided by empirical evidence in contrast to inferred information.

<sup>4</sup>abnormalities in the physiological development

<sup>5</sup>A DNA barcode is a short DNA segment that uniquely assigns a taxon name to an organism. A more detailed explanation follows in Section 2.1.

get into a productive state and deliver the urgently needed feedback is the difficulty of overseeing a metabarcoding setting and accessing all relevant information at once. Senior staff is involved in the Lake Müggelsee monitoring project presented in Section 2.5 for up to 20 years. In contrast, data analysts are relatively shortly involved (a few months to a few years) and have difficulty gathering information that seems to be obvious.

## 1.2 Contribution

The thesis at hand is dedicated to the challenges centered around the evaluation and management of metabarcoding experiments on plankton samples, and the *in silico* discovery of new primer sequences. Introductory, we discuss the difficulty to find a consensus of the many species concepts, and to construct a tree of life that reflects biological relatedness. A comparative study (Chapter 2) aims to determine how well metabarcoding performs compared to the standard approach and where it fails. Overall, metabarcoding is more sensitive, but it fails to identify some clades known to be present. Motivated by the identification failure of a specific clade in freshwater samples (Rotifera), a tool was written that allows to compute chemically suitable primer pairs given reference sequences for that clade (Chapter 3). It is fast enough to deal with large and permanently updated reference databases and robust against low quality or mislabeled sequences – a requirement that is implied by the sparse population of reference databases and the impossibility to curate sequences for thousands of taxa. Finally, a database schema is presented to consolidate and track data centered around metabarcoding experiments. It is intended to allow querying past experiments and gain an overview independently of the availability of dedicated staff. All herein presented tools are open-source and available on GitHub.

## 1.3 Thesis Outline

### Chapter 2 – Species Identification in Environmental Samples

In Chapter 2, we introduce the problem of species identification via metabarcoding. We review the problem of the tree of life (TOF) construction and how multiple sequence alignments (MSAs) guide tree construction. We look at the problem of primer search for metabarcoding experiments where specificity has to be balanced against primer effectiveness. Conventional approaches are evaluated with respect to the possibility to scale up the reference dataset in which primer pair candidates are searched. Two distinct identification methods are presented – the historically older and indispensable method of binocular reading and metabarcoding, a method well known from gut microbiome studies or pathogen detection. A study, conducted at the Leibniz Institute for Freshwater Ecology and Inland Fisheries (IGB), is presented that compares both identification methods qualitatively on freshwater plankton samples.

### Chapter 3 – Primer Discovery in Large Datasets

The primer search tool PriSeT is presented whose initial motivation emerged from the personal involvement in the ongoing monitoring project outlined in Chapter 2. When analyzing species that had not been detected via metabarcoding, but were known to be present in the sample, it turned out that there exists *in silico* tools for simulating the effectiveness of a known primer pair, but no tools for discovering

new primer pairs given arbitrary, more extensive reference libraries. The tool gains computational efficiency by using a FM-frequency computation on the reference dataset, bit vectors with constant *rank* and *select* support and bit-parallelism for sequence property checks. PriSeT is here evaluated for two very different scenarios – taxonomically sparse plankton data sets and whole genomes of pathogenic viruses.

## **Chapter 4 – A Database Schema for Metabarcoding Experiments**

Chapter 4 is dedicated to analyzing the workflow and data organization based on my experiences as a member of an institute working with genetic material. A database schema is presented that summarizes essential information that is otherwise spread across multiple computational units or inaccessible online. Implementation of the scheme would facilitate searching for previous experimental data, shorten the training period of new group members, and enable new types of metastudies.

## **Chapter 5 – Discussion**

Chapter 5 summarizes and discusses the results of the metabarcoding study (Chapter 2), the primer search tool (Chapter 3), and the proposed data organization for metagenome studies (Chapter 4).

## **Chapter 6 – Conclusion**

Chapter 6 concludes with a summary of the contributions. We project what challenges will persist in meta-barcoding and how software tools should address them.

## Chapter 2

# Species Identification in Environmental Samples

Die Natur kommt auf dem kürzesten  
Weg zu ihrem Ziel.

---

*Georg Wilhelm Friedrich Hegel*

### 2.1 Introduction

Metabarcoding is a molecular batch-processing method where DNA is extracted from an environmental sample (*esample*), amplified with the polymerase chain reaction (PCR) method with a primer pair that targets a short marker region (e.g., small subunit (SSU) of rRNA), also called *barcode*, and sequenced on next-generation sequencing (NGS) platforms. The raw reads as output by the sequencer machine are then analyzed and processed into operational taxonomic units (OTUs) using their barcodes. An OTU represents a set of sequences that are identical or highly similar. Instead of applying the identification process to each individual read, the OTU represented by a common-sense sequence (and a read counter) is used instead. An OTU can be successfully classified if there exists a highly similar and taxonomically unique match against a labeled reference sequence in a database. Ambiguous or low-quality matches are tackled by inferring the most probable ancestor.

Metabarcoding is the most affordable method to survey taxonomically heterogeneous communities quickly, but is also used in long-term monitoring projects where regular sampling is required. In the case of environmental monitoring, metabarcoding is complemented by morphological identification under the microscope (see Figure 2.1) as both methods encounter limitations as discussed in Section 2.5.3. The two methods depend on a taxonomy that might be erroneous or is more suitable for one identification method, but not for the other (see Section 2.2). The morphological identification for important indicator species<sup>1</sup> may fail when they are present in larvae or juvenile state due to indistinguishable features (see Section 2.5.2). On the other hand, metabarcoding cannot resolve an OTU with a barcode that matches to references assigned to more than one organism group. The inability to culture and clone most microorganisms for whole genome sequencing complicates the search for barcodes that would allow higher detection rates.

Overall metabarcoding detects a broader range of organisms (diversity) compared to morphology-based identification (Morph ID), but still may under- or overestimate

---

<sup>1</sup>i.e. species that indicate special environmental conditions

a sample's richness<sup>2</sup>. Overestimation occurs when there is intraspecific and intragenomic variability, i.e., individuals of the same species exhibit significant variations in their barcodes; without that there is a morphological manifestation. This becomes problematic if OTU richness is taken as a proxy for species richness (Ritter et al., 2019). This phenomenon is called *cryptic diversity*. An indicator species frequently found in freshwater samples which exhibits cryptic diversity is the diatom *Nitzschia palea* (Rimet et al., 2014).

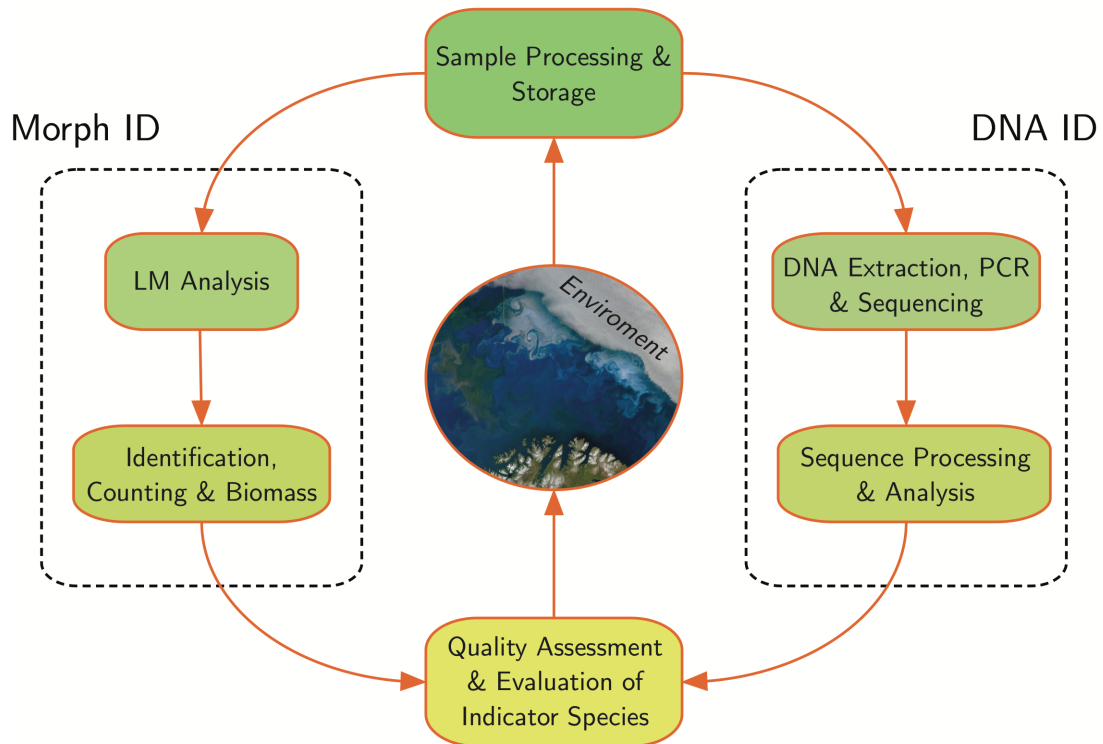


Figure 2.1: Complementing sample processing pipelines for monitoring projects. The NASA satellite image (center) is license-free.

## 2.2 Tree of Life

Identification means that we assign an organism or piece of DNA to the most probable taxonomic rank and name. In case of significant uncertainty, we assign it to a higher taxonomic group containing the unknown species. Taxonomies can be seen as trees grouping hierarchically the set of classified organisms. Depending on the underlying concept, an inner node may correspond to an extinct ancestor or is a virtual placeholder in the face of lacking information. A lineage is a path from the root node – the last universal common ancestor (LUCA) – to a species. The three domains of life are Bacteria, Archaea, and Eukaryota (see Figure 2.2). The ranks that typically constitute the lineage of an individual organism are *domain*, *kingdom*, *phylum*, *class*, *order*, *family*, *genus*, and *species*. Many lineages are refined further by subdividing existing ranks and indicate their relative positions with prefixes like *giga-*, *grand-*, *hyper-*, *infra-*, *magn-*, *micro-*, *min-*, *mir-*, *parv-*, *sub-*, *nan-*, or *super-*. Non-common ranks include *division*, *legion*, *cohort*, *series*, *section*, *tribe*, *varietas*, or *form*. The complete

<sup>2</sup>Richness refers here to the entirety of all sample-contained species, which represents the unknown ground truth.

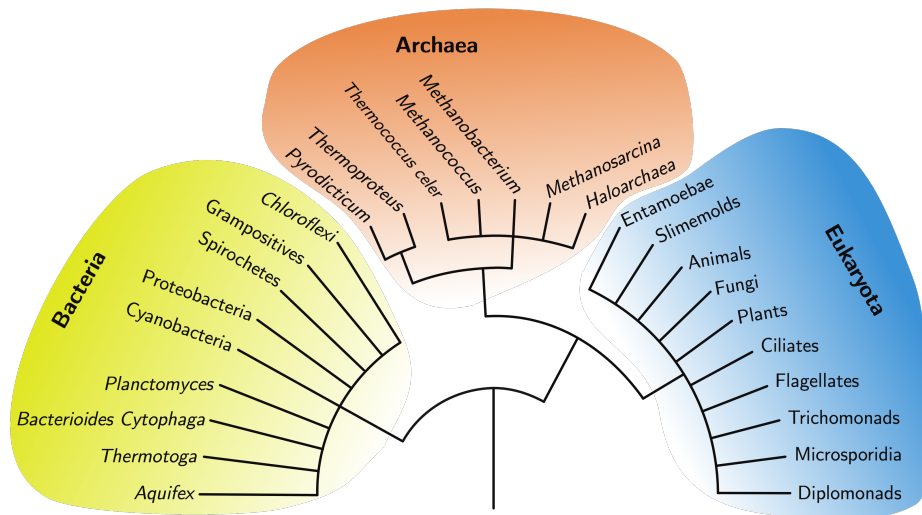


Figure 2.2: Dendrogram of the tree of life showing the three domains Bacteria, Archaea, and Eukaryota based on ribosomal RNA genes, and linked to the last universal common ancestor. The figure is adapted from Woese, Kandler, and Wheelis, 1990. Organisms of separate domains show more profound differences compared to organisms of separate kingdoms within the same domain.

list contains 73 different taxonomic rank names. The taxonomies<sup>3</sup> differ significantly from the original five-rank Linnaean taxonomy published by Carl Linnaeus in 1758. However, the principles are the same - the ranks are nested, and organisms of the same group have a greater number of similarities than organisms of different groups.

Attempts to classify species go back hundreds of years, when criteria were limited to what was visible to the eye and observable within the lifespan of a human. Therefore, the decision to call a particular group<sup>4</sup> a phylum is highly anthropogenic. Particularly formative, especially for higher taxonomic levels, are the differences in body plan. The phylum of mollusks (Mollusca), for example, consists mainly of invertebrates with shells and includes more than 100,000 species. Mollusks have a bilaterally symmetric body plan, a complete gut, an undifferentiated main body cavity (*coelom*), and are externally shelled<sup>5</sup>. The full set then of characteristics is supposed to define each phylum uniquely, whereas single features may be shared between phyla.

### 2.2.1 Identification

Most non-sequence-based identification methods are guided by a particular taxonomy until no further distinction is possible. For example, we would first look for the significant features of the body structure and reproductive system before examining the shape of the mouthparts.

Taxonomic ranks suggest that there is something in common across all groups. Nevertheless, even for domains and phyla, some propose six, others up to 32 kingdoms

<sup>3</sup>there are several taxonomies that are not identical, e.g., the GenBank taxonomy differs from Silva's taxonomy

<sup>4</sup>A group is a branch in a tree of life that groups organisms thought to include all evolutionary descendants of a common ancestor

<sup>5</sup>from [https://global.oup.com/us/companion.websites/9780195326949/student\\_resources/facts/#mollusca](https://global.oup.com/us/companion.websites/9780195326949/student_resources/facts/#mollusca) on 09.12.2021.



(Tedersoo, 2017), there is disagreement, and for the rank of species, there are a dozen concepts. In an overview De Queiroz, 2007 lists 14 major classes of contemporary species definitions. A primary species concept can be based on biology (interbreeding is possible and results in fertile offspring), ecology (sharing of the same niche), on evolution, or phylogeny, to name a few. Mayden, 1997 states that after evaluating the many concepts for their theoretical and operational qualities, it is essential to have a monistic, primary concept of species, applicable to everything that is believed to be a species. Whereas, secondary concepts should only be used as operational tools to identify new species and employ differences in morphology, genetics, or behavior. The primary concept, according to Mayden, 1997 is the *evolutionary species concept*, which he describes as “*an entity composed of organisms which maintains its identity from other such entities through time and over space, and which has its own independent evolutionary fate and historical tendencies*” (Mayden, 1997). This concept challenges scientists to adequately capture the temporal and spatial components of an entity, as they cannot be experienced within a human lifespan. Similarly, evolutionary fate can only be assessed retrospectively by examining genetic similarities and differences.

Defining microbial species is particularly difficult because some have the same body type but genetic markers show high sequence variability, or vice versa. In such a case, mating data would provide an indication of whether the variance is interspecific or reflects a species difference, but they are not always available. Rimet et al., 2014 found that there is no objective criteria for species separation and that instead barcoding will need a consensual approach to molecular species limits.

Current models synthesize biological and phylogenetic information: subdivisions are either based on visible biological features or genetic distances between a set of ribosomal protein sequences or complete genomes. Hug et al., 2016 are rendering a taxonomy based on RNA sequences or genomes and yield a different taxonomy than those currently in use. There is also a difference in whether SSU rRNA is used or a set of protein sequences combined with complete genomes. A preceding study from 2016 rendered a tree of life for all three domains by aligning 16 ribosomal protein sequences from each organism for which a high-quality draft or complete genome was available (3,000 in total). Large parts are congruent with an SSU rRNA based tree of life, but as expected, it yields a higher phylogenetic resolution and is not prone to artifacts (compared to using only one gene). In this tree of life, the domain of Bacteria contains 92 named phyla, the domain of Archaea 26 phyla, and Eukaryota five supergroups (Hug et al., 2016). Since more sequence data have become available, Burki et al., 2020 has presented a new eukaryotic phylogeny of life based solely on molecular phylogenies. In this new eukaryotic tree of life, the authors identify 13 supergroups (see Figure 2.3). The former supergroup of Chromalveolata (Alveolata, Stramenopila, Haptophyta, and Cryptophyta) was formed by the assumption that secondary plastids were once acquired from a common algal ancestor. Given more molecular evidence, this supergroup turned out to be *polyphyletic*<sup>6</sup>. In the proposed tree of life by Burki et al., 2020, Stramenopila and Alveolata are assigned to the supergroup TSAR (an acronym of its members: telonemids, stramenopiles, alveolates, and Rhizaria), Haptophyta to the supergroup Haptista, and Cryptophyta to the supergroup Cryptista. The new sister group of SAR is the flagellate taxon Telonemia, currently consisting of only two species: *Lateronema antarctica* and *Telonema subtile*. It is estimated that SAR contains half of the eukaryote species: major groups of microbial algae like diatoms and dinoflagellates, seaweeds, free-living protozoa like ciliates or foraminiferans, and protozoan parasites like oomycetes.

<sup>6</sup>composed of phyla that do not share an immediate common ancestor



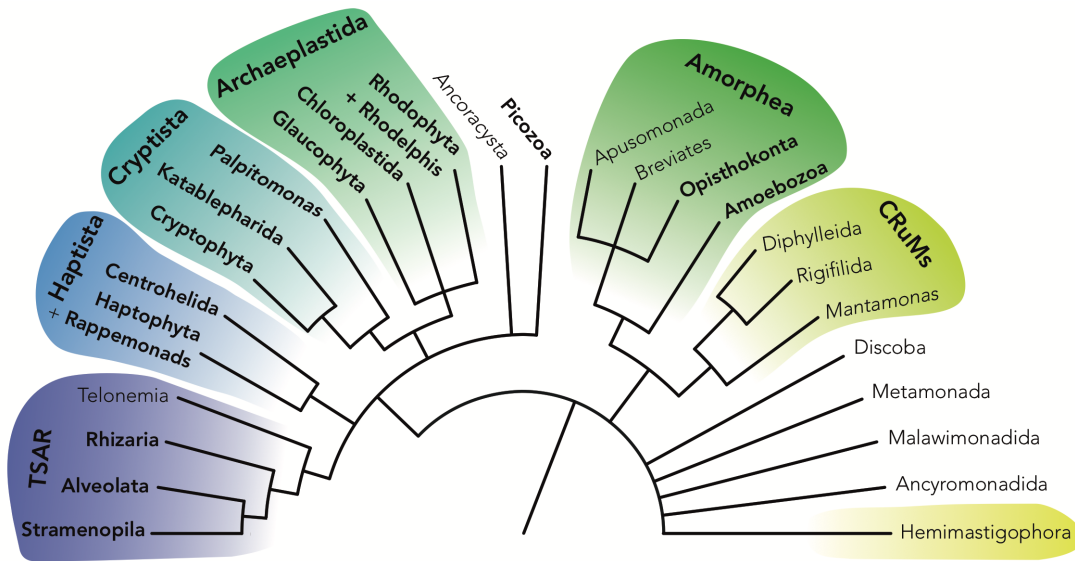


Figure 2.3: A dendrogram of the eukaryote tree of life proposed by Burki et al., 2020. It is based on a consensus of recent phylogenomic studies. Current supergroups are shown colored. Unresolved branching orders are shown as multifurcations. Individuals from half of the supergroups (marked bold) can be found in freshwater samples.

Both studies have in common that the phylogenetic placement uncovers the existence of new lineages and, therefore, a higher diversity as stated so far. Large parts of this diversity are only accessible via cultivation-independent genome resolution. There is ongoing work to draft and ratify rules for a phylogenetic nomenclature, which will then be regulated by the *International Code of Phylogenetic Nomenclature* (Cantino, De Queiroz, et al., 2020). The challenge is to pose a taxonomic model that accurately reflects the phylogenetic distances between species but is also suitable for traditional identification methods. Next, we will look at the methods for constructing a tree of life.

### 2.2.2 Tree of Life Construction

The idea is to represent the tree of life as a graphic structure with a root node, inner nodes and leaf nodes. The inner nodes represent the ancestors and the leaves represent the species. When classifying an unknown sequence, we would walk downstream from the root to the leaves until no further distinction is possible. In the case of ambiguous assignment, i.e., high sequence similarity to multiple taxa, we determine the lowest common ancestor (LCA), i.e., the ancestor that encompasses both species and has a maximal distance to the root. Whereas binocular identification, as used for species identification of microplankton samples, is guided by phenotypical features like shape or ornaments.

In DNA- and protein-based classification the times of separation from a common ancestor are determined from sequence differences, concretely their *genetic distance*. Best results are achieved when combining sequence data from different genomic sources or loci.

For phylogenetic tree construction, identity scoring schemes should account for all the levels of evolutionary changes that apply. A change in allele frequencies is a *microevolutional* change occurring *within* a species. It is driven by gene flow, genetic

drift, mutation, or natural selection. Parasite mediated selection is an example of a microevolutionary change. In contrast, *macroevolutionary* changes are guided by the selection of interspecific variations (Hautmann, 2020).

### Distance-Matrix Methods

A simple and computationally fast approach is to compute a multisequence alignment (see 2.2.4) and express pairwise distances in dependence of mismatched positions. Sequences with small distances are arranged under the same interior node; branch lengths reflect the genetic distances. The complete tree can be constructed via *neighbor joining* – a bottom-up clustering algorithm (Saitou and Nei, 1987). MSA computation is not suitable for all types of genomic sources, e.g., whole genomes. Pairwise distances can be computed in other ways like using the Lempel-Ziv complexity as the relative information between sequences (Otu and Sayood, 2003).

### Maximum Parsimony

Among the set of possible phylogenetic trees, the one minimizing the amount of evolutionary change required is maximizing parsimony (Fitch, 1971). Alternatively stated, a tree constructed under the maximum parsimony criterion minimizes the number of similarities that cannot be explained by inheritance. Thereby, an optimal tree minimizes homoplasy<sup>7</sup> and is the shortest possible tree (Farris, 2008). As there exists no algorithm to generate optimal parsimonious trees, the complete tree space must be searched. Exhaustive search is therefore only feasible for a handful of sequences. Otherwise, branch-and-bound or heuristic approaches are applied. A most-parsimonious tree may underestimate the true number of evolutionary changes.

### Maximum Likelihood

Another computationally expensive method is the maximum likelihood method (Felsenstein, 1981) which estimates the overall likelihood of all trees and selects the maximizing one. For a given configuration and nucleotide position in a branch, its likelihood depends on whether the nucleotide is present or absent in the ancestor. The cumulative probabilities are computed for each component independently and summed to yield the final score. The underlying concept assumes that each nucleotide site evolves independently, which is certainly not always true. E.g., when a locus encodes a protein whose higher dimensional structure preservation is crucial, we observe that remote positions co-evolve, i.e., change simultaneously. Another restriction arises from its computational expensiveness – it is only feasible for a few sequences. For larger sequence sets, first, a solution is calculated on a subset. Secondly, the solution is adjusted by adding the remaining sequences one by one.

### 2.2.3 About Sequence Distances

Computing a distance score based on MSAs is only one out of many ways of how to determine the phylogenetic distance between two taxa. In order to yield stable results for each taxon not only multiple individuals per species should be sampled, but also multiple loci be evaluated. A survey study by Dogan and Dogan, 2016 compared 26 different scoring schemes. For macroevolution level comparison the authors conclude

<sup>7</sup>independent loss or gain of a trait in separate lineages

that the scheme proposed by Nei, 1972 would be best (see Equation Abschnitt 2.2.3). Nei, 1972 allows the expression of differences by genetic drift *and* mutations. Given two randomly mating diploid populations,  $X$  and  $Y$ , and a set of  $L$  investigated loci,  $x_i$  and  $y_i$  denote the frequencies of the  $i$ -th allele in population  $X$  and  $Y$ , respectively. The probability for two randomly chosen loci to be identical is  $j_X = \sum x_i^2$  for population  $X$ , and  $j_Y = \sum y_i^2$  for population  $Y$ . Likewise, the probability for identity when choosing from both populations is  $j_{XY} = \sum x_i y_i$ . Nei, 1972 then define the distance  $D^{\text{Nei}}$  as the normalized arithmetic mean of the identity as shown in Equation (2.1).

$$\begin{aligned} D^{\text{Nei}} &:= -\ln I & (2.1) \\ I &:= \frac{J_{XY}}{\sqrt{J_X J_Y}} \\ J_X, J_Y, J_{XY} &:= \frac{1}{L} \sum j_X, \frac{1}{L} \sum j_Y, \frac{1}{L} \sum j_{XY} \end{aligned}$$

For microevolutional comparisons Dogan and Dogan, 2016 recommend distance measures by Sanghvi, 1953 or Edwards, 1971 depending on sample size and allele frequency ratios.

#### 2.2.4 Multiple Sequence Alignment (MSA)

Multiple sequence alignments remain an important operation in character-based TOL construction. Given a set of sequences  $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$ , it constructs an alignment of two or more sequences, such that a penalty function is minimized. Biologically plausible solutions exist only for similar sequences, as explained in the next sections. MSAs help to tackle two related areas – the detection of highly conserved regions and the inference of an evolutionary past of their carriers. It is unnecessary to align complete genomes, but a set of informative sequences is sufficient to construct phylogenetic trees with high confidence.

##### Multiple Sequence Alignment (MSA)

**Definition 2.1.** A sequence  $S_i$  is a string of characters from an alphabet  $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_l\}$ . We can obtain an alignment between two or more sequences by inserting an additional gap symbol, denoted as ‘–’, such that all sequences have the same length. For an alignment to be meaningful columns containing only gap symbols are not allowed.

Identical non-gap symbols at an alignment position are a *match*, and distinct non-gap symbols are a *mismatch*. Gap costs are usually assessed separately and are often further distinguished according to whether a gap is opened or extended (*affine gap costs*). Character matches are rewarded, mismatches and the presence of gap symbols are penalized. An *optimal MSA* is an alignment that minimizes the total costs with respect to a scoring scheme. An example of a scoring scheme is the sum-of-pairs shown in Equation (2.3).

For DNA sequences with unambiguously encoded nucleotides we have  $\Sigma_{DNA} = \{A, C, G, T\}$ . Without restricting generality, we use the alphabet  $\Sigma_{DNA}$  in all examples.

## MSA Construction

When constructing a multiple sequence alignment of two or more sequences with various lengths, we typically seek for the optimal alignment. As the score function is decisive for the alignment, it has to be chosen such that the biologically most plausible solution is also in the set of optimal solutions. We will see later that the larger the sequence dissimilarities are, the more solutions exist that are equally good, and the choice becomes an arbitrary one. The operational set for transforming sequences into aligned and annotated sequences is the insertion of gap symbols and mismatches, corresponding to the evolutionary events insertion, deletion, and mutation. More complex events like sequence rearrangement, replication, or inversion cannot be considered. Instead, solutions are greedily formed from local matches or dynamic programming approaches are used that extend prefix solutions. MSAs are best suited for aligning relatively short and related genomic regions or closely related genomes.

There are infinitely many ways to transform one sequence into another. However, since there is a finite amount of time passed between a population separation, and modifications need to preserve metabolic functioning, those modifications are more probable that require the lowest amount of changes and have intermediate genotypes that are not fatal.

The result of an MSA is typically displayed as a matrix  $M \in (\Sigma \cup \{-\})^{n \times \mathcal{O}(\bar{m})}$  where  $n$  is the number of sequences and  $\bar{m}$  the expected sequence length. Rows correspond to input sequences and columns to global alignment positions containing up to  $|\Sigma| + 1$  distinct characters. Apart from the alignment method, the scores for mismatches, insertions, deletions, or gap opening versus extension<sup>8</sup>, significantly influence the final shape of the alignment. For similar sequences, we expect that we find few variations per column. Figure 2.4 shows an alignment excerpt of cytochrome oxidase subunit I (COI) sequences computed by PRANK (Löytynoja, 2014).

pos	526	527	528	529	530	531	532	533	534
$S_1$	A	C	T	G	C	A	-	-	-
$S_2$	A	C	T	G	-	T	G	T	C
$S_3$	A	C	T	G	C	T	G	T	C

Figure 2.4: A multiple sequence alignment computed with Wasabi's PRANK (Löytynoja, 2014) of sequences  $S_1$ ,  $S_2$ , and  $S_3$  from three different species.

Given an MSA, we can construct a phylogenetic tree (see Section 2.2.2) with branches expressing spatial separation of populations, and branch lengths the time since their separation. The series of DNA modifying events that could be causal for our observed sequences is shown in Figure 2.5. Though phylogenetic tree construction (with default parameter settings) would arrange all three species as direct ancestors of  $S_0$ .

With the number of sequence variations, the number of possible insertion, deletion, and substitution operations grows exponentially, thereby generating a set of possible operations that are equally scored and equally probable from a biological point of view. As a result, the reported alignment is only one of many, or worse, biologically meaningless. Therefore, it is not advisable to perform MSA calculations

<sup>8</sup>Intuitively, a gap opening is less probable than extending an already existing gap and should, therefore, be higher penalized.

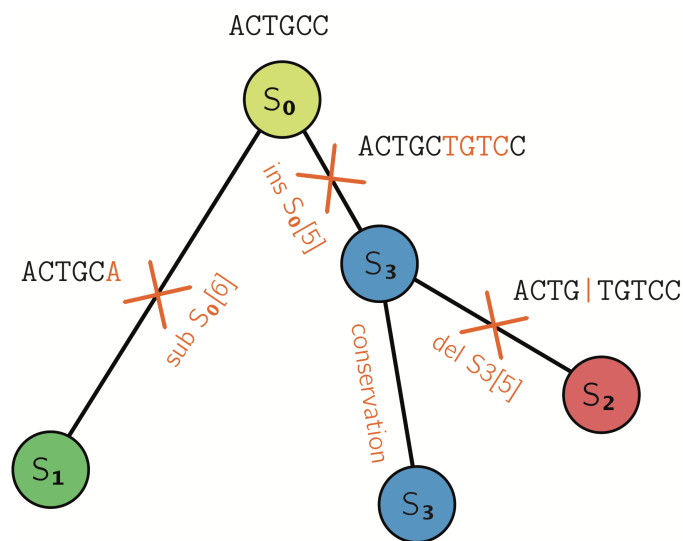


Figure 2.5: Possible events supported by the sequences  $S_1$ ,  $S_2$ , and  $S_3$ , transformed into a phylogenetic tree given their sequence alignment (see Figure 2.4). Here we assume that the sequence information is sufficient to compute the phylogenetic tree and that the events are significant enough to support speciation. Sequence  $S_0$  is the sequence of a common, hypothetical ancestor. One explanation is that at some point the population carrying  $S_0$  got separated and a substitution of C to A at position six took place resulting in the observed sequence  $S_1$ . For the second group TGTC got inserted after position five resulting in the observed sequence  $S_3$ . After a subpopulation split from carriers of  $S_3$ , a deletion event occurred at position five resulting in  $S_2$ . Whether,  $S_1$  to  $S_3$  are considered as distinct species is dependent on the concrete model and scoring.

for sequences that are genetically very distant.

### Sum-of-Pairs Score

Assuming a length of  $m$  of the final alignment, the quality of an alignment can be assessed straight forward by scoring each of the  $m$  sites of the alignment independently and pair-wise – same characters are scored with 0 and different characters, including the gap symbol, are penalized with a fixed value  $\lambda$  (typically  $\lambda = 1$ ). This approach is called *sum-of-pairs* (SP) score:

$$SP := \sum_{j \in [1:m]} \sum_{\substack{i_1, i_2 \in [1:m] \\ i_1 \neq i_2}} \text{score}(S_{i_1}[j], S_{i_2}[j]) \quad (2.2)$$

$$\text{score}(\sigma_1, \sigma_2) := \begin{cases} 0, & \text{if } \sigma_1 == \sigma_2 \\ \gamma \in \mathbb{R}^+, & \text{else} \end{cases} \quad (2.3)$$

For a scoring scheme to be a metric, the axioms of *identity of indiscernibles* (Equation 2.4), *symmetry* (Equation 2.4), and *triangle inequality* (Equation 2.6) have to hold for all  $S_i, S_j, S_k \in \Sigma^+$ .

$$\text{metric}(S_i, S_j) = 0 \iff S_i = S_j \quad (2.4)$$

$$\text{metric}(S_i, S_j) = \text{metric}(S_j, S_i) \quad (2.5)$$

$$\text{metric}(S_i, S_j) \leq \text{metric}(S_i, S_k) + \text{metric}(S_k, S_j) \quad (2.6)$$

The triangle inequality ensures the transitivity of closeness in the distance measure. Its introduction was motivated by the possibility of speeding up similarity searches in large sequence databases (Stojmirović and Yu, 2009). The SP score satisfies the triangle inequality and is widely used due to its ease of calculation.

A straight-forward approach to compute an MSA precisely on sequences  $S_1, S_2, \dots, S_n$  is via dynamic programming (DP). Each axis of the  $n$ -dimensional table corresponds to one of the aligned sequences. Any point  $(i_1, i_2, \dots, i_n)$  in the table stores the costs (and backtracking information) of the best alignment of all prefixes  $S_1[1 : i_1], S_2[1 : i_2], \dots, S_n[1 : i_n]$ . The edges are initialized with accumulated gap insertion costs. Starting with cell  $(1, 1, \dots, 1)$ , the computation is carried on to neighboring cells along each dimension by considering preceding optima and the costs to align, insert, or delete the new character against all other. The algorithm was first described for two sequences by Vintsyuk, 1968 – a pioneer in speech recognition, but gained more popularity among Bioinformaticians as the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970). It can easily be expanded to be used with more than two sequences. Below is a code snippet for three sequences and the sum-of-pairs scoring scheme with  $\lambda = 1$ .

For each additional sequence, another nested for-loop has to be created, resulting in  $\mathcal{O}(m^n)$  computation steps. In each iteration step  $2^n - 1$  neighboring cells are looked up to compute in  $\mathcal{O}(1)$  the costs of an optimal alignment passing through one of the neighboring cells. The accumulated runtime costs are  $\mathcal{O}(2^n m^n)$ . There is no algorithm known that *substantially* breaks down the exponential runtime. Practical runtime improvements have been achieved by restricting the considered regions (Carrillo and Lipman, 1988, Lermen and Reinert, 2000), parallelization, or GPU acceleration. There exists heuristical approaches that build the alignment progressively like MAFFT (Nakamura et al., 2018), iteratively like MUSCLE (Edgar, 2004), are phylogeny-aware like PAGAN (Löytynoja, Vilella, and Goldman, 2012), by simulating annealing (Kim,

---

**Algorithm 1** Optimal alignment of three sequences via dynamic programming. Mismatch, deletion, or insertion scores follow the scoring scheme in Equation (2.3) with  $\lambda = 1$ . At each step the optimum of  $2^3 - 1$  adjacent cells including insertion/deletion and mismatch costs is carried on to cell  $(i, j, k)$ .

---

```

1: procedure AlignThree( $S_1, S_2, S_3$ )
2:    $dp \leftarrow \text{int}[|S_1| + 1][|S_2| + 1][|S_3| + 1]$ 
3:    $dp[0][0][0] \leftarrow 0$ 
4:    $dp[i][0][0] \leftarrow 2i \forall i \in [1 : |S_1|]$ 
5:    $dp[0][j][0] \leftarrow 2j \forall j \in [1 : |S_2|]$ 
6:    $dp[0][0][k] \leftarrow 2k \forall k \in [1 : |S_3|]$ 
7:   for all  $i \leftarrow [1 : |S_1|]$  do
8:     for all  $j \leftarrow [1 : |S_2|]$  do
9:       for all  $k \leftarrow [1 : |S_3|]$  do
10:         $\text{score}_{i,j} \leftarrow (S_1[i] == S_2[j]) ? 0 : 1$ 
11:         $\text{score}_{i,k} \leftarrow (S_1[i] == S_2[k]) ? 0 : 1$ 
12:         $\text{score}_{j,k} \leftarrow (S_1[j] == S_2[k]) ? 0 : 1$ 
13:         $d_{ijk} \leftarrow dp[i-1][j-1][k-1] + \text{score}_{i,j} + \text{score}_{i,k} + \text{score}_{j,k}$ 
14:         $d_{ij} \leftarrow dp[i-1][j-1][k] + \text{score}_{i,j} + 2$ 
15:         $d_{ik} \leftarrow dp[i-1][j][k-1] + \text{score}_{i,k} + 2$ 
16:         $d_{jk} \leftarrow dp[i][j-1][k-1] + \text{score}_{j,k} + 2$ 
17:         $d_i \leftarrow dp[i-1][j][k] + 2$ 
18:         $d_j \leftarrow dp[i][j-1][k] + 2$ 
19:         $d_k \leftarrow dp[i][j][k-1] + 2$ 
20:         $dp[i][j][k] \leftarrow \min(d_{ijk}, d_{ij}, d_{ik}, d_{jk}, d_i, d_j, d_k)$ 
21:   return  $dp[|S_1|][|S_2|][|S_3|]$ 

```

---



Pramanik, and Chung, 1994), or by simulating quantum computing (Nuin, Wang, and Tillier, 2006). All of these efforts demonstrate the importance of multiple sequence alignments as a method of analysis that allows inference of phylogeny, prediction of protein structure (Jumper et al., 2021), or primer design in the biological domain alone.

### 2.2.5 NP-Completeness of $MSA_{SP}$

In this section we elucidate the NP-completeness of MSAs using the SP-score ( $MSA_{SP}$ ). Even though there exist runtime improvements, an MSA remains computationally expensive. Greedy approaches often exploit sequence similarities, which are less given for phylogenetically distant organisms.

The complexity class of the MSA problem depends on the score function. But even for the simple SP-score, the MSA problem is NP-complete, i.e.,  $MSA_{SP} \in NP$  and any other NP-complete problem can be reduced to  $MSA_{SP}$ . The complexity class NP describes the set of problems that are solvable by a non-deterministic Turing machine in polynomial time (machine-definition). An equivalent way to define NP containment is to require that a proposed solution can be verified by a deterministic, polynomial-time algorithm (verifier-based definition). In contrast to the Halting problem<sup>9</sup>, NP-complete problems are decidable.

NP-completeness can be proved by reducing a version of the Shortest Common Supersequence problem (SCS) to the MSA problem. Maier, 1978 showed that the Vertex Cover problem<sup>10</sup> reduces to SCS over an arbitrary alphabet and Middendorf, 1994 that SCS reduces to  $SCS_2$  – the common supersequence problem over an alphabet of size two. It is advisable to reduce  $SCS_2$  to  $MSA_{SP}$  since their representations are very similar, which facilitates the translation between the two inputs.

Given a set of binary encoded sequences  $\mathcal{B} = \{B_1, B_2, \dots, B_m\}$  with  $B_i \in \{0, 1\}^*$ ,  $SCS_2$  poses the question of what is the shortest supersequence  $\tilde{S}$  with length constraint  $|\tilde{S}| \leq m$  ( $m \in \mathbb{N}$ ), such that every sequence  $B_i \in \mathcal{B}$  is a subsequence of  $\tilde{S}$ ? In other words we search a sequence from which we can construct every sequence from  $\mathcal{B}$  by removing symbols in  $\tilde{B}$ . In contrast to substrings, subsequences must not be embedded consecutively in another sequence.

The equation below shows formally the supersequence  $\tilde{B}$  over an arbitrary sequence set  $\mathcal{B}$ . Let  $s_j \in \mathcal{B}$  be a set of disjoint substrings of  $\tilde{B} \cup \{\emptyset\}$ , then a superstring is:

$$\forall_{B_i \in \mathcal{B}} \exists_{s_j \in \tilde{S}} \text{ s.t. } \tilde{S} = s_1 \circ B_i[1] \circ s_2 \circ B_i[2] \circ \dots \circ s_{|B_i|} \circ B_i[|B_i|] \circ s_{|B_i|+1} \quad (2.7)$$

The shortest supersequence minimizes the total substring length while still fulfilling containment of  $\mathcal{B}$  in  $\tilde{B}$ . To give an example, assume we have  $B_1 = 00$ ,  $B_2 = 11$ , and  $B_3 = 0010$ . A supersequence of  $\mathcal{B}$  is  $\tilde{B} = 01010$ . By crossing out non-relevant letters from the supersequence all sequences can be recovered. The tabular layout (Table 2.1) resembles already a multiple sequence alignment with the difference that substitutions are missing – in each column we either have  $\{0, -\}$  or  $\{1, -\}$ . As we will see later, this fact complicates the translation between  $MSA_{SP}$  and  $SCS_2$ . Using the term *translate* is not chosen by chance, formally, we express inputs as words of a language  $L$ , which can either be accepted ( $w \in L$ ) or rejected ( $w \notin L$ ). We can think of language  $L(MSA_{SP})$  as the set of all accepted inputs in the format  $w_{MSA} =$

<sup>9</sup>Given a description of an arbitrary computer program, decide whether the program will terminate or run forever.

<sup>10</sup>Given an undirected graph, find a minimal node set such that at least one endpoint of each edge is covered.



$\tilde{B}$	0	1	0	1	0
$B_1$	0	-	0	-	0
$B_2$	-	1	-	1	-
$B_3$	0	-	0	1	0

Table 2.1: A supersequence of  $B_1$ ,  $B_2$ , and  $B_3$ .

$(\mathcal{S}, M, c_{MSA})$  where  $\mathcal{S}$  is the set of sequences<sup>11</sup>,  $M$  represents an optimal alignment with score not larger than  $c_{MSA}$  (according to definition (2.2)), and  $L(SCS_2)$  as the set of accepted inputs in the format  $w_{SCS} = (\mathcal{B}, \tilde{B}, c_{SCS})$  where  $\tilde{B}$  is a supersequence of  $\mathcal{B}$  with score  $c_{SCS}$ .

The computation of valid solutions is delegated to an oracle. The best known exact algorithm for computing valid solutions runs in exponential time relative to the input size. It suffices to verify an input in polynomial time and to find an invertible translation function, called *encoder* from hereon, that maps words from one to words for the other language in polynomial runtime w.r.t. to the input size, and yields the same (non-)acceptance answers. We call the inverse function of an encoder a *decoder* from hereon.

Suppose we want to show that problem  $Q$  is NP-complete given that another problem  $R$  has been proved to be NP-complete. A typical proof by reduction follows four steps.

- (i) Show that  $Q \in NP$ , i.e. a solution or word can be verified in polynomial time.
- (ii) Select a problem  $R$  that has been proved to be NP-complete.
- (iii) Construct a polynomial-time encoder that translates words for  $R$  into words for  $Q$  and show  $w_R \in L(R) \Leftrightarrow \text{encode}(w_R) \in L(Q)$ .
- (iv) Prove that the encoder runs in polynomial time.

We are free to choose a scoring scheme satisfying triangular inequality. Table 2.1 suggests that forward and reverse mapping is straightforward. However, when mapping alignments with columns containing substitutions, we could expand such columns (transform a mixed column into two columns containing either  $\{0, -\}$  or  $\{1, -\}$  in the alignment) without that the optimal score is changed. This would result in a group of alignments that have only one supersequence correspondence – the encoder would not be a bijective function anymore (see Example 2.1). The Berman-Hartmanis conjecture states that the encoder has to be a bijective, invertible function and that its runtime is polynomial in both directions (see conclusions in Berman and Hartmanis, 1977). There is evidence for and against the conjecture. All known NP-complete proofs, so far, respect the conjecture, and in fact, the proof of the Berman-Hartmanis conjecture would implicate  $P \neq NP$ .

The need for the encoder to be bijective poses a challenge when reducing  $SCS_2$  to  $MSA_{SP}$ , because we have to enforce optimal alignments to contain *no columns* with mixed characters. The bad news is that when using the binary alphabet alone, there exists no scoring scheme satisfying triangular inequality that can penalize  $0 - 1$  alignments in a way that mixed columns are suppressed, which can be easily seen

<sup>11</sup>identical to  $\mathcal{B}$

in an example where we give the highest possible score for mismatches (without violating the triangular property):

$$\text{score}(\sigma_1, \sigma_2) = \begin{cases} B_1 = 0 \\ B_2 = 1 \\ 0, & \text{if } \sigma_1 = \sigma_2 \\ 2, & \text{else if } \sigma_1 \neq \sigma_2 \wedge \sigma_1, \sigma_2 \in \{0, 1\} \\ 1, & \text{else} \end{cases}$$

resulting in two optimal alignments of cost 2. Only the second one is directly translatable into a superstring.

### Multiple Optimal Solutions

**Example 2.1.** The sequences  $S_1 = 0$  and  $S_2 = 1$  can be aligned in three different ways with an optimal score of 2.  $\mathcal{A}_{\text{opt}}^2$  and  $\mathcal{A}_{\text{opt}}^3$  can be unambiguously translated into the supersequences  $S^2 = 01$ , and  $S^3 = 10$ , respectively. It is not evident how to translate  $\mathcal{A}_{\text{opt}}^1$  into a supersequence.

$$\mathcal{A}_{\text{opt}}^1 = \begin{array}{c} 0 \\ | \\ 1 \end{array}, \mathcal{A}_{\text{opt}}^2 = \begin{array}{c} 0 \quad - \\ | \quad | \\ - \quad 1 \end{array}, \mathcal{A}_{\text{opt}}^3 = \begin{array}{c} - \quad 0 \\ | \quad | \\ 1 \quad - \end{array}$$

One way to satisfy triangular inequality and the Berman-Hartmanis conjecture at the same time is to add two new symbols and to fix the scoring, as shown in the subsequent proof based on Wang and Jiang, 1994. Noteworthy to mention is Elias, 2006, who also proved the intractability of the MSA with SP-score by reducing from the *Independent R3 Set*<sup>12</sup> problem. In addition, he proved that two related problems are NP-hard: *Star Alignment* and *Tree Alignment*.

### MSA<sub>SP</sub> is NP-complete

**Theorem 2.1.** The decision version of the multisequence alignment problem with the SP-score is NP-complete.

*Proof.* We will show that  $\mathcal{B}$  has a supersequence  $\tilde{B}$  of length  $c_{\text{SCS}}$  if and only if  $\mathcal{S}$  has an alignment with a score of at most  $c_{\text{MSA}}$ . For  $\mathcal{B}$  and  $\mathcal{S}$  we assume the same underlying alphabet  $\Sigma = \{0, 1\}$ .

- (i)  $\text{MSA} \in \text{NP}$ , i.e. a solution  $w_{\text{MSA}} = (\mathcal{S}, \mathcal{A}, c_{\text{MSA}})$  can be verified in polynomial time by applying Equation (2.2) to the alignment  $\mathcal{A}$  and compare the resulting score to  $c_{\text{MSA}}$ . For each position of the alignment of length  $m$  we have to compare  $\binom{m}{2}$  characters. The length of the alignment is at most  $\|\mathcal{S}\|$  – the length of the concatenation of all sequences, resulting in a worst-case runtime of  $\mathcal{O}(\|\mathcal{S}\| m^2)$ .
- (ii) The  $\text{SCS}_2$  has been proved to be NP-complete by Middendorf, 1994 and we will reduce it to  $\text{MSA}_{\text{SP}}$ .
- (iii) We alter the  $\text{MSA}_{\text{SP}}$  problem by adding two new characters  $\alpha, \beta \notin \Sigma$  to the alphabet, and form two sequences  $\alpha^i$  in the length of the number of zeros found in the supersequence, and  $\beta^j$  in the length of the number of ones. As addressed

<sup>12</sup>*Independent Set* problem with degree bounded by 3

previously, the purpose is to suppress alignments that have ambiguous super-sequence analoga. We show  $w_{SCS} \in L(SCS_2) \Leftrightarrow \text{encoder}(w_{SCS}) \in L(MSA_{SP})$ . Specifically,  $\mathcal{B}$  has a supersequence  $\tilde{B}$  of length  $c_{SCS}$  iff  $X_{i,j} = \mathcal{S} \cup \{\alpha^i, \beta^j\}$  has an alignment with a score of at most  $c_{MSA}$ . ' $\Leftarrow$ ' describes the encoding of  $w_{MSA}$  to  $w_{SCS}$ , and ' $\Rightarrow$ ' vice versa.

' $\Leftarrow$ ' We form sequences  $\alpha^i$  and  $\beta^j$  for some  $i, j$ , such that  $i + j = m$ . Our new sequence set is  $X = \mathcal{S} \cup \{\alpha^i, \beta^j\}$ <sup>13</sup>. We define a scoring scheme, such that among all solutions, there will be one optimal alignment in which  $\alpha$  will be aligned with zeros, and  $\beta$  with ones (see Table 2.2).

	0	1	$\alpha$	$\beta$	-
0	2	2	1	2	1
1	2	2	2	1	1
$\alpha$	1	2	0	2	1
$\beta$	2	1	2	0	1
-	1	1	1	1	0

Table 2.2: Scoring scheme for proof of Theorem 2.1.

$$\begin{aligned} \mathcal{S}' &= \mathcal{S} \cup \{\alpha^i, \beta^j\} \text{ with } i + j = c_{SCS} \\ \mathcal{A} &= \text{construct\_alignment}(\mathcal{S}') \\ c_{MSA} &= (m - 1)|\mathcal{S}'| + (2m + 1)c_{SCS} \end{aligned}$$

No matter how we align the sequence set  $\mathcal{S}$ , the score will always be  $(m - 1)|\mathcal{S}'|$ <sup>14</sup>. For example, introducing gaps (new columns) does not change the costs as their alignment costs halfens and  $\text{score}(-, -)$  is zero. There rests a contribution of  $\alpha^i$  and  $\beta^j$  of at most  $(2m + 1)(i + j)$ . Consequently, one possible optimal alignment contains zeros aligned with an  $\alpha$  and ones with a  $\beta$  symbol as aligning  $\alpha - \beta$  worsens the score. From such an alignment we obtain the supersequence by simply setting  $\tilde{B}[i] = 0$  if column  $i$  contains  $\alpha$  and  $\tilde{B}[i] = 1$  if column  $i$  contains  $\beta$ . Its length will be  $i + j = c_{SCS}$ . The sequence set  $\mathcal{B}$  equals  $\mathcal{S}$ .

' $\Rightarrow$ ' Given a shortest common supersequence  $\tilde{B}$  over a set of binary sequences  $\mathcal{B} = \{B_1, B_2, \dots, B_m\}$ , we construct an encoder function formatting input of  $SCS_2$  to input of  $MSA_{SP}$ :

$$\text{encoder} : (\mathcal{B}, \tilde{B}, c_{SCS}) \mapsto (\mathcal{S}, \mathcal{A}, \text{score}, c_{MSA})$$

The encoder places each sequence from  $\mathcal{B}$  into a new row and its symbols are aligned such that letters of the sequence match letters of the supersequence. Skipped characters relative to the supersequence are filled with gap symbols. Two additional sequences  $\alpha^i$  and  $\beta^j$  with  $i + j = |\tilde{B}| = c_{SCS}$  are aligned with  $\alpha$  where supersequence contains zero, and  $\beta$  where it contains a one. We now have  $c_{SCS}$  columns that either contain  $\{\alpha, 0, -\}$  or  $\{\beta, 1, -\}$  resulting in an alignment score of  $c_{MSA}$ .

<sup>13</sup>We do not fix the values for  $i$  and  $j$ , because the only invariant known about the corresponding supersequence is that 0s and 1s sum up to  $m$ .

<sup>14</sup> $S$  denotes the length sum of all contained sequences

- (iv) It can easily be seen that encoder and decoder instructions operate in polynomial time w.r.t. their number of sequences.

□

## Conclusion

The NP-completeness of the  $MSA_{SP}$  problem also has consequences for boundaries of approximate, heuristic approaches mentioned in the previous section. Where possible parallelization and GPU acceleration are utilized to speed up alignment computations. MSA and phylogenetic tree computations are related: an MSA aids in constructing a tree, an existing phylogenetic tree can on the other hand guide the alignment construction as done in PRANK and PAGAN (Löytynoja, 2014) and Canopy (Li, Medlar, and Löytynoja, 2016). MSAs always conserve the sequence ordering. Consequently, they are not able to discover optimal solutions that include larger chromosomal rearrangements like inversion, translocation, or replication. From this point of view, it is not advisable to align distantly related genomes, but rather a subset of short and informative sequences.

In practice, sets of functional regions, i.e., protein-encoding sequences are concatenated and aligned. Hug et al., 2016 used a set of 16S ribosomal protein sequences. Using functionally related proteins reduces the chance for phylogenetic artifacts arising when genes are subject to different evolutionary processes. For an in-depth introduction and overview of phylogenetic tree construction, we refer to the book "Inferring Phylogenies" by Joseph Felsenstein (Felsenstein, 2004).

Care needs to be taken when using regions of high similarity as a guidance for a phylogenetic tree construction because sequence similarity does not automatically induce homology. An example are protein-coding genes that could have evolved independently and converged.

## 2.3 Polymerase Chain Reaction (PCR)


### 2.3.1 Principle

Given an environmental sample, the DNA is extracted through cell lysing and chemical separation from macromolecules, lipids, proteins, or RNA. For the polymerase chain reaction method the double-stranded DNA is mixed with DNA polymerase, primer sequences, a buffer of a salt solution, and free deoxynucleoside triphosphates (dNTPs). The mixture runs through a series of 25-35 thermal cycles. Each cycle undergoes three phases. The free dNTPs serve as building blocks for synthesis. First, the reagents are heated for denaturation, cooled down for primer annealing, and slightly heated again for amplicon elongation. During the denaturation phase, at 94-98 °C for 20-30 seconds, the hydrogen bonds of double-stranded DNA are broken. The now single-stranded DNA templates can serve as a primer binding site. When cooling down to 50-65°C for 20-40 seconds, primers will bind to complementary regions on the template DNA as this represents an energetically lower state. The optimal annealing temperature should be 3-5 °C below the *melting temperature* ( $T_m$ ) of the primers, which is the state when half of the primer sequences are dissociated. It can be altered by adding salt or changing the pH value. For a PCR to be effective, the melting temperatures of both primer sequences should be close to each other ( $\Delta T_m \leq 5$  Kelvin). The sequence synthesis is driven by DNA polymerase for which the hybridized primer acts as a substrate. DNA polymerase promotes sequential agglomeration of freely available dNTPs (see Figure 2.7) from 5' to 3'<sup>15</sup>. The optimal annealing temperature depends on the type of polymerase - *Taq polymerase*<sup>16</sup> works best between 72 - 78 °C. The duration of the elongation phase is chosen in dependence on the estimated transcript length and synthesis speed (about 1-1.5 kb/min). The programmable apparatus that facilitates the temperature increase and decrease is called *thermal cycler* (see Figure 2.6).

The final elongation phase is prolonged to 5-15 min to ensure that single-stranded templates are fully extended. The reaction is stopped by cooling down the mixture to 4-15 °C. At this temperature; it can be stored for a more extended amount of time until sequencing.

PCR products are usually checked by *agarose gel electrophoresis*. The DNA fragments are mixed with running buffer and a fluorescent tag such as ethidium bromide, which binds by intercalating between DNA base pairs. The mixture is inserted into pockets of the viscous gel, placed in a tray and exposed to an electric field. The negatively charged DNA fragments run through the gel towards the anode. Their velocity is inverse logarithmic to the sequence length - smaller fragments travel faster. Into one of the pockets, a ladder mixture with known fragment lengths (e.g., 100 bp, 200 bp, 300 bp, and so forth) runs in parallel and allows length estimation of the PCR products



Figure 2.6: Thermocycler “Baby Blue”, ca. 1986. Credits: [Science Museum London](#) / [Science and Society Picture Library](#). Licensed under .

<sup>15</sup>or 3' to 5' relative to the template

<sup>16</sup>a thermostable *DNA polymerase I* originally isolated by Chien, Edgar, and Trela, 1976

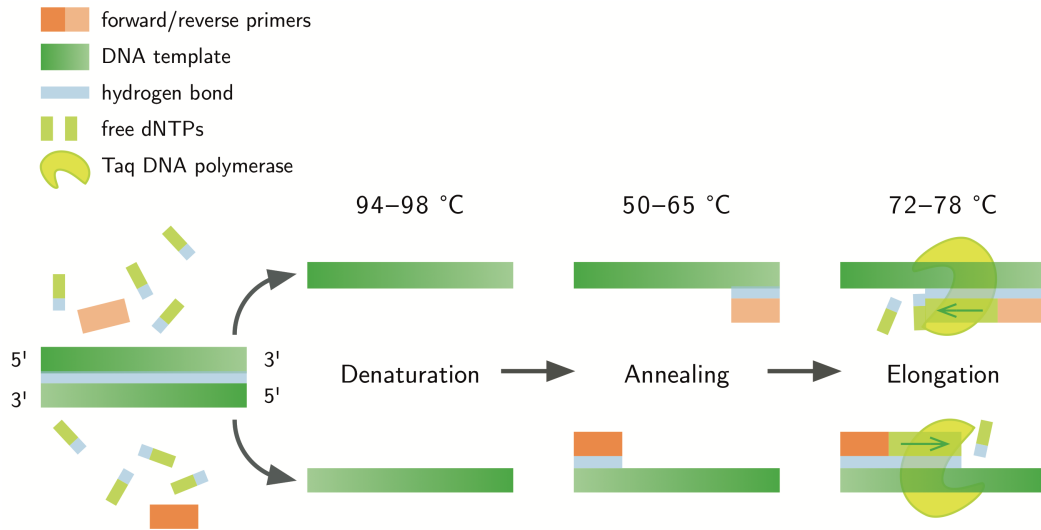


Figure 2.7: First cycle of a PCR. The denaturation leads to hydrogen bond breakage, allowing the primer sequences to bind to the single-stranded template during annealing. For an effectively working polymerase, the temperature is slightly increased. The elongation time depends on the targeted transcript length. It is interrupted by increasing the temperature again to 94–98 °C for another round. In the second round, original DNA and transcripts serve as copy templates, such that we have nearly a duplication after each cycle.

via comparison with the ladder bands. Ideally, the mixture should form one well-defined band under UV light. A smeared band or multiple bands indicate secondary amplification products, DNA degradation, or the presence of PCR chimeras.

Finally, the PCR transcripts are digitized by a sequencer machine. Sequencing is a stochastic process; confidence in a particular base is expressed by a quality score that is output along with the sequences in what is known as FASTQ format. Most common is the Phred score to describe the confidence of each single base with  $e$  being the probability for a wrong base call:

$$Q_{\text{phred}} = -10 \log_{10} e$$

The lower the error probability, the higher the score  $Q_{\text{phred}}$ . The quality scores are encoded with ASCII symbols between '!' and 'I' (Illumina 1.8+ format).

The most applied type of PCR for metabarcoding experiments is a PCR with paired primers as depicted in Figure 2.7. Two distinct and chemically not interfering primer sequences are added as reagents. They are located upstream and downstream relative to the target region. Accordingly, the upstream located primer is called *forward primer*, its counterpart the *reverse primer*. At the end of a PCR, we obtain overlapping transcripts.

One of the first steps in DNA read processing is the identification and merging of overlapping reads. Read merging improves the confidence of sequence quality, especially near the ends of reads where errors accumulate. The degree of overlap must be weighed against the length of the barcode. Statistically, a longer barcode offers a higher chance of being species-specific.



### 2.3.2 Primer Design

When designing primers we need to consider the chemical nature of DNA which induces properties that must hold for all candidate primers and can be computed independently from the DNA template(s). We would like to achieve specificity<sup>17</sup> and stable template binding.

From a software engineering point of view it is helpful to split the set of constraints into two sets: one that must hold for single primer sequences ( $C_s$  hereafter) and one that arise from fixing the strand orientation and other primer sequence to form a pair ( $C_p$  hereafter). The point of this separation is that on the one hand it gives us an additional way to parallelize constraint checks, but on the other hand it also covers other use cases like primer calculation for single-ended PCR.

While there are patterns that lead to PCR failure (e.g., TATA boxes, self-annealing of 6-mers), there are constraints that are soft and can be partially controlled by buffer composition, temperature settings, or cycle lengths.

The PriSeT algorithm described later in Section 3.9 will apply  $C_s$  in an early step to reduce the tremendous amount of  $k$ -mers per sequence and avoid unnecessary computations. Whereas  $C_p$  is checked in a subsequent combination step (see Table 2.3). Table 2.3 summarizes their recommended settings for a standard PCR. The subsequent sections describe the chemical constraint checks implemented in PriSeT.

#### Primer Length

Independent of the PCR variant, the length parameter controls the specificity and the capability to bind easily in the annealing step. A range that is considered to be optimal is 18 to 22 bases. Though the recommendations slightly vary between different sources and protocol variants. For example, for the reverse transcriptase PCR a range of 18 to 24 bases is recommended (Thornton and Basu, 2011).

High specificity for a target can be achieved by choosing longer sequences. However, longer primers will take more time to hybridize and dehybridize, and therefore produce less amplicon. The chance for secondary structures, dimerization, and interaction with other reagents inflates likewise.

In a heterogeneous mixture for which we would like to capture the target fragment from as many genomes as possible, we are likely to find sequence variations even in conserved regions. Therefore, choosing a rather short primer will allow us to amplify more genomes, while increasing the chance for being unspecific to the target region. One other reason for not too short primers is that the enzymes require a minimum working temperature which must fit the melting temperature of the primer. As we will see in the next section, shorter primers generally have lower melting temperatures because there are thresholds for maximum CG content, which contributes about twice as much to the melting temperature compared to AT.

#### Melting Temperature

The melting temperature in a PCR is the temperature at which half of the DNA duplexes will be dissociated. Only when template DNA is single-stranded, primer sequences will bind, and the enzymatically driven elongation will take place. The temperature for the denaturation phase can be adjusted, but too high temperatures will lead to irreversible damage to the molecular components.

<sup>17</sup>which is improved in expectation by choosing longer sequences

The simplest and oldest method to compute melting temperatures is the Wallace rule (Wallace et al., 1979). Irrespective of the nucleotide positions, it counts the occurrence of GC and AT and weighs them with 2 and 4, respectively. The different weights account for the stronger bond between cytosine and guanine (three hydrogen bonds) compared to adenine and thymine (two hydrogen bonds).

$$T_m = 2AT + 4GC \quad (2.8)$$

Denaturation and annealing phase temperatures can be chemically adjusted. However, we need to ensure that roughly the same amount of forward and reverse primers are dissociated to produce similar amounts of transcripts by allowing not more than five Kelvin difference in their melting temperatures. Note that, some published primer pairs for 18S in metabarcoding experiments have excessively diverging melting temperatures (see Table 3.11 in Section 3.12.2).

### GC-Content

Another criterion for primers being not too associative to the template is their GC-content. Not only are C-G bonds twice as strong as A-T bonds, but they are also more prone to mispairing. The recommended range is 40 - 60 % in proportion to the sequence length.

### Mono- and Dinucleotide Runs

Mono- or dinucleotide runs are patterns where a single nucleotide or dinucleotide is consecutively repeated, i.e.  $\sigma_x$  (mononucleotide run) or  $(\sigma_1\sigma_2)_x$  (dinucleotide run) with  $\sigma \in \{A, C, G, T\}$ ,  $\sigma_1 \neq \sigma_2$ , and  $x \in [2 : k_{\max}]$ . PCR amplification tends to alter the mononucleotide or dinucleotide repeat length (Clarke et al., 2001). A maximal accepted number is four. Some primer design manuals even recommend to restrict serial runs to three.

### Self- and Cross-Annealing

Self-annealing (or self-dimerization) occurs when copies of the same primer sequence form stable dimers, and cross-annealing when two distinct primer sequences dimerize. If their binding energy is relatively high, a significant amount will not be available for the template and thereby derate template amplification. The Gibb's free energy is a measure of the bond strength. It relates to the energy released during bond formation and should not be below  $-6 \text{ Jmol}^{-1}$ . The amount depends on the positions and types of nucleotides involved, and can only be determined precisely via physicochemical experiments.

When inspecting an alignment of two oligomers, as a rule of a thumb, there should be no more than four self- or cross-annealing nucleotides in a row (connected annealing pattern) as shown in Figure 2.8a and 2.8b or not more than 50 % of the sequence (disconnected annealing pattern) be involved in bonding as shown in Figure 2.8c and 2.8d.

### GC Clamps and AT Tails

GC clamp refers to the amount of GC in the 5'-end of a primer. Especially at the 5'-end where the elongation occurs, a primer should bind stably, but not too strongly.



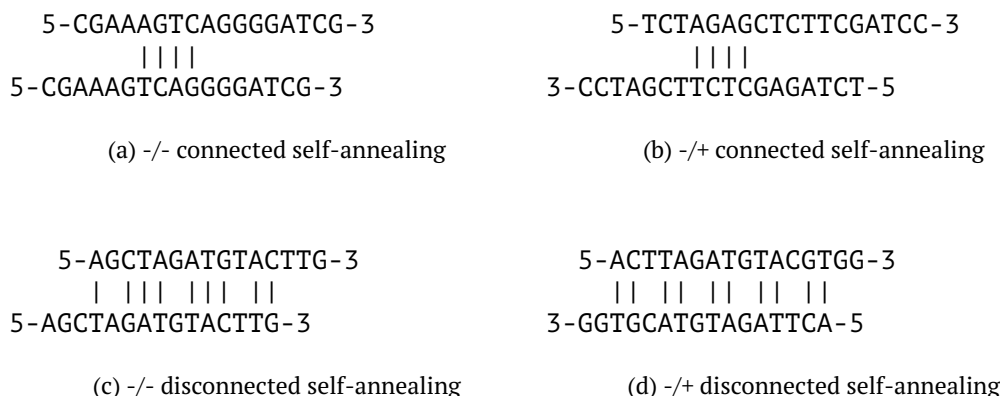


Figure 2.8: Critical self-annealing patterns: oligomers in a) and b) have four self-annealing nucleotides in a row. In c) and d), more than half of the nucleotides participate in bonds. The oligomer orientation can be the same (-/-) or opposite (-/+).

Since C and G bind twice as strongly as A or T, ideally one or two nucleotides should be C or G, but the last three bases at the 3'-end should not be exclusively A or T, also called AT tail. For example, CGATA-3 and CGTCA-3 are not suitable, whereas AGTCA-3 is a suitable primer tail.

Constraint Set	Property	Recommendation
$C_s$	Primer Length	[16:25]
	Melting Temperature	[52:58]
	GC-Content	[0.4:0.6]
	Mononucleotide Runs	not more than 4 same nt in a row
	Dinucleotide Runs	not more than 4 dinucleotides
	Self-Annealing	less than four consecutive nts and less than 50 % bond participation
$C_p$	$\Delta T_m$	$\leq 5$ Kelvin
	AT Tails	avoid (A T) <sub>3</sub> tails
	GC Clamp	not more than 3 out of 5 nts at 3' end are GC
	Cross-Annealing	less than four consecutive nts and less than 50 % bond participation
	Transcript Length	[30:800]

Table 2.3: Chemical constraints for conventional PCR. The first set of constraints  $C_s$  applies to  $k$ -mers irrespective of their later orientation (forward or reverse) and can be applied directly when looking up a  $k$ -mer sequence. The second set of constraints  $C_p$  can be computed as soon as the second primer sequence is determined.  $C_p$  checks are applied in the final combination step.

### 2.3.3 Reverse-Transcriptase qPCR

Reverse-transcriptase PCR (RT-PCR) in combination with real-time PCR (qPCR) plays an increasing role in clinical diagnostics of viral pathogens as it is not only carried out faster, but allows for a more precise quantification, an improved sensitivity and

reproducibility compared to the conventional approach, and also tolerates partial RNA degradation (Bustin, 2002). The thermal cycler itself produces qualitative and quantitative results. As additional sequencing can introduce contamination and setting up a new bioinformatics pipeline may take days, this has a massive impact on the quality and number of analyses that can be carried out in the same amount of time.

The first step of a RT-PCR is to translate RNA into its DNA complement. In the second step, real-time PCR is applied similarly to a conventional PCR, except that the polymerization phase can be dropped due to the shorter product lengths (60-150 bp). An additional sequence recognition step is carried out at the end of each cycle. There currently exist two methods based on fluorescence sensing – the addition of fluorescent dye that binds unspecific to double-stranded DNA or fluorescent reporter probes. Classical sequencing for example with Illumina is not involved.

Apart from viral pathogen detection, other application fields of RT-PCR are the quantification of gene expression levels, e.g., in the field of oncology to detect circulating cancer cells, assignment of genotypes of pathogens, or detection of genetically modified organisms, as the modification often leads to a replicated insertion of the transgene and therefore provides a higher level.

The shortening of cycles and reduction of unspecific binding has implications for the optimal PCR conditions. Therefore, more extended and GC-richer primer sequences are favored. From the conventional PCR deviating parameter recommendations are listed in Table 2.4.

Property	Recommendation
Primer Length	[18:24]
Melting Temperature	[52:58]
GC-Content	[0.5:0.6]
Transcript Length	[60:150]

Table 2.4: Chemical constraints for RT-PCR primers deviating from standard settings.

## 2.4 Challenges in Species Identification of Plankton Samples

We will now focus specifically on microplankton, also called net plankton, which comprises plankton organisms with body sizes between 0.05 and 1 mm. For historic reasons plankton was characterized by its inability to propel against a current, which led to their name *plankton* from Greek  $\pi\lambda\alpha\gamma\kappa\tau\acute{o}\zeta$  – wanderer or drifter (Thurman and Trujillo, 2004). Despite many zooplankters having appendices that allow them to propel, their ability to swim against currents is limited. Newer definitions include aeroplankton to account for wind spreading of seeds, spores, and pollen (Smith, 2013). As plankters are not defined by their phylogeny, but their niche and level of motility, they comprise closely related, and as well very distant clades (see Section 2.5).

Given an environmental sample from an aquatic biotope, we would like to assess the many contained species, i.e., to generate a taxonomic profile. The species are first mechanically separated by size and either subsampled into defined volumes for identification and counting by eye or batch-processed by extracting and processing their DNA. Although both methods are fundamentally different, they rely on the same taxonomic tree, which may be constructed based on genetic data or, more commonly, is a hybrid model merging phenotypical feature classifications and phylogenetic information. The challenges of both identification methods are described briefly in the next sections. An example of a concrete sampling, identification, and counting protocol is given in the study presented in Section 2.5.

### 2.4.1 Microscopic Method

While the first classified animals could be examined by eye, it took a dramatic improvement in microscopes at the end of the 19th century to enter the realm of microbiomes. When combined with an artificial light source, light microscopes illuminate the sample and allow researchers to work closer to theoretical limits (Murphy and Davidson, 2012). Today's microscopes for plankton identification are equipped with cameras that facilitate inspection and allow human sample processors to archive image data instead of physical samples. At IGB, two microscopes are in use: the Nikon Diaphot 300 (see Figure 2.9) and Axiovert 135.

Classification of organisms seen under the light microscope relies solely on their body morphology. It needs year long training of the sample processors to allow for morphology-based identification. The two processors at IGB specialize in either phytoplankton or zooplankton identification. They continuously extend their knowledge base by consulting reference literature and colleagues, participating in specialized seminars, or collecting feedback from DNA-based analyses. The two preceding phytoplankton processors were in office for about 18 years each. Before retiring, each plankton processor trains his or her successor



Figure 2.9: Nikon Diaphot 200/300 Inverted Microscope. ©Nikon.

for one year until the variation in identification is in the same range as the parallel count of the same sample by the current processor. This procedure minimizes knowledge-based bias. The actual task is to estimate abundances and biomasses of the most relevant organisms.

The first step of the identification routine is to create a list of all species or higher order organisms that can be seen under the microscope. The species list simplifies the counting step, as the counted organisms only need to be matched with the confirmed species.

The samples are retrieved by filtering a specified water volume from a lake or river site with pressure<sup>18</sup>. The samples are enriched with water and fixed to stop the movement of the organisms. Subsamples of dedicated volumes are transferred into Utermöhl chambers (phytoplankton, see Figure 2.10) or to Sedgewick-Rafter counting chambers (zooplankton) for subsequent counting.

Organism groups are considered relevant if they contribute significantly to biomass or play a role as indicators of water quality. Not all organisms can be determined to species level. Some rotifer genera like *Collotheca* and *Syncheata* can only be identified to the genus level because some important taxonomic features are not visible after fixation. The ability to identify to species level also depends on the sex or juvenile state – in the order of Cladocera (see Figure 2.11), females are determined to species level, whereas interspecific hybrids, males, or juveniles without brood chamber are identified only to genus level. Adult specimens of copepodites are identified to species level and sex, whereas juveniles stages (nauplii and copepodites) are distinguished no further than into cyclopoid and calanoid.

The biomass is calculated as the product of individual counts and biovolume by assuming that one mm<sup>3</sup> corresponds to a plankton mass of one  $\mu\text{m}$  (Mischke and Behrendt, 2007). Limitations are encountered for species that form colonies – cell numbers are then estimated, or for filamentous algae spanning two transects.



Figure 2.10: Utermöhl counting chambers by Panek. Licensed under [CC](#) [i](#).



Figure 2.11: Various genera of Cladocera by Andrei Savitsky. Licensed under [CC](#) [i](#).

<sup>18</sup>detailed description of sampling protocol in method Section 2.5.2

### 2.4.2 Metabarcoding Method

Metagenomics is the study of genetic material that is directly obtained from heterogeneous environmental samples. Asking the obvious question about the species composition is only one out of many. Often specific biomes are insufficiently backed up by a reference database, and if backed up, then mostly by regions that are known to serve as molecular clocks.

DNA *metabarcoding* combines polymerase chain reaction (PCR, see Section 2.3), next-generation sequencing (NGS), and identification via DNA barcodes, i.e. amplicons (300-800 bp) that are matched to a reference database. A typical sample processing pipeline is given later in Section 2.5. In contrast, single DNA probing with genome assembly would be more precise but is for many reasons not feasible, e.g., the majority of microorganisms cannot be clonally cultured<sup>19</sup> – a necessity for assembling sequences (Rappé and Giovannoni, 2003).

The evolutionary diversity of planktonic organisms presents a major challenge for metabarcoding: On the one hand, plankton account for more than half of the recognized supergroups in the eukaryotic tree of life, as shown in Section 2.2 and Figure 2.3, and at the same time many closely related species co-occur. A marker-based approach faces the challenge of finding a region that is sufficiently conserved to provide primer binding sites but variable enough to distinguish taxa, ideally down to the species level. Barcodes of distinct species showing a high sequence similarity, e.g. more than 97 %, manifest in a unifying *operational taxonomic unit* (OTU) as a product of the bioinformatics pipeline. In such a case the OTU represents organisms with taxonomically distinct lineages. In combination with a sparsely populated reference database, which is subsequently consulted for OTU labeling, the result can be an overestimated<sup>20</sup> taxon or a diminished resolution.

An added challenge is that most NGS platforms suitable for the analysis of large numbers of samples have a read-length limit of ca. 300-450 bp. The read-length limit reduces the number of nucleotides available to distinguish among closely related lineages and hampers the application of DNA metabarcoding for studies in aquatic biodiversity (Mohrbeck et al., 2015). This fueled the development of two approaches: using taxon-specific primers to examine a restricted number of lineages (e.g., zooplankton, diatoms) or *universal* primers that allow for the examination of a broader range of taxa but that are often not able to provide taxonomic resolution to the genus or species level (Wurzbacher et al., 2017).

The majority of DNA metabarcoding studies of plankton target the SSU RNA gene<sup>21</sup>. The gene is present in all known eukaryotic lineages, with sufficient copy numbers per genome to make PCR amplification feasible, and with a combination of highly variable (i.e., species-specific) regions and highly conserved regions (i.e., for *universal* primer binding sites). Plenty of studies support the suitability of SSU RNA as a marker: Schmidt, Rodrigues, and Mering, 2014 demonstrated that OTUs from 16S/18S rRNA reflect the underlying ecological diversity consistently across habitats. Even if OTUs cannot be resolved, their counts correlate to the sample's diversity and are roughly independent of the species composition. As a result, SSU RNA gene sequences are well represented in reference databases (Stoeck et al., 2010, Hadziavdic et al., 2014; Albaina et al., 2016).

<sup>19</sup>It is estimated that only 1% of all species are cultivatable.

<sup>20</sup>in terms of read abundance

<sup>21</sup>also referred to as 16S for prokaryotes and mitochondrial ribosomes in eukaryotes, or 18S for eukaryotes



Metabarcoding is especially prone to the non-standardization of taxonomic systems. In practice, it is not uncommon to consult more than one reference database for OTU resolution because they specialize in different groups of organisms. In such a case, OTUs are associated with lineages of different taxonomies that cannot be harmonized, making comparative analyses difficult.

Finding the optimal set of primers w.r.t. coverage and resolution takes many costly iterations of trial and error (Elbrecht et al., 2019). Scientists encounter problems like missing ground truth and sparsely populated reference databases. Often they would start with primer sequences published in previous studies on similar data sets and then compare OTU diversity between different primer sets.

However, some of these challenges may be overcome as the cost of NGS methods continues to decline, making multi-marker approaches more attractive, and sequence databases continue to be replenished. Altogether, metabarcoding of environmental DNA (eDNA) is a far more sensitive method for species detection than the traditional methods (Smart et al., 2015) provided that a suitable marker is chosen.

However, some tasks apart from identification cannot be solved by molecular methods, but require microscopy. These tasks are description of new taxa, abundance estimation<sup>22</sup>, differentiation of life stages, or observation of teratological forms as indicators for environmental pollution.

A very different DNA-based method is the *whole metagenome shotgun* (WMGS) sequencing. DNA is randomly sheared, sequenced, and reconstructed into consensus sequences. However, low abundant organisms may remain unnoticed, as read assembly requires a minimum of read coverage per genome position. The heterogeneity of esamples does not allow WMGS to reconstruct whole genomes. Instead, assemblers construct the largest reliable *contigs*<sup>23</sup>. Its ability to identify species is not as precise as DNA metabarcoding, but WMGS is continuously becoming better (see Segata et al., 2013 for further reading). A considerable advantage of WMGS is that it is useful for species identification and reveals metabolic processes that are possible in a community<sup>24</sup>. WMGS is not currently used as an alternative to metabarcoding in monitoring projects due to its inferiority in species identification and high cost.

---

<sup>22</sup>Correlating read and individual counts in heterogeneous mixtures is currently not feasible.

<sup>23</sup>set of overlapping DNA segments representing a consensus region

<sup>24</sup>Note: only genome, not transcriptome is sequenced

## 2.5 The Lake Müggelsee Long-Term Monitoring Project

The original motivation for the primer search tool (PriSeT<sup>25</sup>) presented in Chapter 3 resulted from participation in a study as part of an ongoing monitoring project conducted by the Leibniz Institute of Freshwater Ecology and Inland Fisheries in Berlin, Germany. The monitoring project was initiated in the 1970s and led to numerous findings on the plankton dynamics of nearby Lake Müggelsee and the River Spree.

Freshwater biomes are especially diverse – they account for 0.8 % of the earth's surface (Gleick, 1996) and about 6 % of the global species diversity, i.e., 100,000 species out of approximately 1.8 million (Dudgeon et al., 2006). In a single sample, we can find species originating from more than half of the supergroups depicted in Figure 2.3.

Freshwater biomes follow distinct seasonal patterns, driven by environmental conditions (Adrian et al., 2009), but vary between years in the same habitat and across different ecosystems. Besides having species-specific demands concerning abiotic resources and co-existence, plankton species also affect trophic interactions within food webs, and thereby affect the species pool within metacommunities. For understanding the connectedness of the ecosystem, a first and primary goal is to track the species composition over time and space. The global decline in freshwater biodiversity also calls for new ways to determine species distributions at a reliable taxonomic resolution on broad spatial and temporal scales. There are plenty of organisms that are biological indicators for specific environmental conditions like diatoms for eutrophication caused by runoff of agricultural fertilizer or sewage, or even sea-level change.

Identifying plankton to species level is time-consuming and requires years of training as there are only a few non-homoplastic features visible under the light microscope. Resolution must be weighed against spectrum width, since there is only one human operator and new samples are collected at high frequency.<sup>26</sup> Motivated by the high sensitivity, and speed that could be gained by a metagenomic approach, the study intends to compare morphological identification under the light microscope and metabarcoding.

Involved in this study are Katrin Preuss and Ursula Newen (plankton identification, sample processing), and Susan Mbedi, and Sarah Sparmann at the Berlin Center for Genomics in Biodiversity Research for library preparation and next-generation sequencing. Tatiana Semenova-Nelson, Michael Monaghan, and Rita Adrian initiated, guided, and contributed significantly to this study. Justyna Wolinska, Christian Wurzbacher, Jana Kulichová, Jan Köhler, and Sabine Hilt contributed information about the study sites, DNA extraction, and diatoms.

### 2.5.1 Study Motivation

Ecological studies on plankton use mostly microscopical methods for taxonomic identification (see Section 2.4). The resolution varies between order and species level as for some clades features are morphologically indistinguishable.

Next-generation sequencing platforms offer an alternative method to characterize plankton communities, whereby taxa are identified using specific regions of their genomes (see Metabarcoding Method in Section 2.4.2). The batch processing of environmental samples at a relatively low cost offers the chance to conduct elaborate

<sup>25</sup><https://github.com/mariehoffmann/PriSeT>

<sup>26</sup>weekly to biweekly

analyses that otherwise could not be implemented at a larger scale or conducted at high frequency. As we are in the transition from traditional analysis methods to molecular identification, it is crucial to understand in what aspects metabarcoding is underperforming and how we compensate for it.

In the study, that is presented in the following, the compositional data of phyto- and zooplankton species from lake and river as produced by light microscopy (LM) or metabarcoding identification were qualitatively compared. For the metabarcoding analysis, three different primer sets were used on the same samples. All primer sets aim for the V4 region of 18S rRNA – a region that provides many species-specific barcodes and reflects well the diversity of samples (Schmidt, Rodrigues, and Mering, 2014). Two of the three primer pairs were designed to assay a broad taxonomic range of eukaryotes and one to assay diatoms, which is expected to be the most diverse group based on morphological studies from previous years. In particular, we were interested in the following questions:

1. How do both methods differ in their number of identifications on the genus and species level?
2. How do species-level resolutions compare for major plankton groups?
3. How do both methods compare in their ability to differentiate lake and river sites?

Here, we expected to recover about 68 % of the taxa that were morphologically identified to species level and had database references, based on an *in silico* PCR<sup>27</sup> of publically available plankton sequences using our primers. We additionally used bioinformatics tools to search for explanations of diverging results. By answering these questions, we gain insight into missed identifications. The underlying reasons may be the choice of barcode, lack of reference sequences, or the need to adjust and fine-tune the metabarcoding protocol and bioinformatics pipeline.

## 2.5.2 Materials and Methods

### Study Sites

Lake Müggelsee is a *polymictic*<sup>28</sup>, *eutrophic*<sup>29</sup> lake located on the eastern edge of Berlin, Germany (52.4369°N, 13.6357°E). The lake has a mean depth of 4.9 m, a maximum depth of 8.0 m, a surface area of 7.4 km<sup>2</sup>, and a water retention time of circa 100 days (Driescher et al., 1993). Plankton communities were collected on multiple dates from September to November 2014 (see Table 2.5). On each date, samples were collected from five locations within Lake Müggelsee and combined for further processing. The River Spree flows into and out of Lake Müggelsee and was sampled multiple times at two locations ca. 40 km upstream of Lake Müggelsee (Spree Große Tränke, hereafter SGT; 52.368603°N, 13.997045°E) and ca. 10 km upstream (Spree New Zittau; hereafter SNZ; 52.392355 °N, 13.744298°E). River water travels between these sites in about one day (in winter) to two days (in summer). At mean discharge, the river is about 20-25 m wide and 1.4 m deep.

<sup>27</sup>In an *in silico* PCR, we search for primer sequences matches in the reference dataset without performing a real PCR. In case forward and reverse primer match for at least one reference sequence, we expect the PCR to amplify the species' barcode successfully. However, it remains open if the barcode is sufficiently distinct for taxonomic resolution.

<sup>28</sup>A polymictic lake has mixed waters because it is too shallow to develop thermal stratification.

<sup>29</sup>with high nutrient levels



Location	Lake Müggelsee					River SGT			River SNZ			
Date	Sep-15	Sep-29	Oct-13	Oct-27	Nov-10	Sept-02	Oct-14	Nov-25	Sep-02	Sep-16	Oct-14	Nov-25
Sample ID	S23	S4	S6	S21	S24	S31	S7	S17	S32	S38	S8	S18

Table 2.5: Samples collected in autumn 2014 from lake and river sites.

### Sampling of Plankton Communities

Plankton was collected from surface water using a 5-L Friedinger sampler (HYDRO-BIOS Apparatebau GmbH, Kiel, Germany) as shown in Figure 2.12.

For morphological identification, phytoplankton was subsampled (1-3 L, depending on plankton density), or accumulated for zooplankton (20 L). The water samples were filtered (5  $\mu\text{m}$  mesh size for phytoplankton and 30  $\mu\text{m}$  mesh size for zooplankton) before identification. For counting and biomass estimation of phytoplankton, smaller subsamples (50 ml) were collected and fixed with Lugol's solution (50 g Potassium Iodide p.a. in 100 ml distilled water, 25 g double sublimated Iodine p.a., 250 ml distilled water, 5 ml pure acetic acid until the final concentration is 10 %), placed in a dark bottle, and kept cold. Zooplankton samples for counting biomass were fixed by adding formaldehyde to a final concentration of 4 %.

### Sample Processing for Morphological Identification

Phytoplankton samples were processed by transferring the Lugol-fixed samples to an Utermöhl chamber and leaving them undisturbed until plankton settled to the bottom. The content of the chamber was then evaluated under an inverted microscope according to the EU standardized Utermöhl procedure (DIN EN 15204). For the biomass estimation, cell numbers per taxon were counted along transects using a Nikon Diaphot 300 (see Figure 2.9) or an Axiovert 135 (Zeiss, Jena, Germany) light microscope with magnification factors 200, 400, or 1000. One transect was counted across the middle of the chamber; next, the chamber was rotated 180° for the second transect analysis. A minimum of 400 individuals were counted in each sample. The entire chamber area was screened for large (e.g., Ceratium) and rare species. Cell dimensions were determined via an ocular micrometer or a microscope camera.

Zooplankton samples were processed by removing formaldehyde from the fixed samples by filtration (mesh size 30  $\mu\text{m}$ ). Depending on zooplankton density, different aliquots (25 ml, 50 ml, or 100 ml) of the sample were



Figure 2.12: Integrated water sampler. ©HYDRO-BIOS Apparatebau GmbH

transferred to the Sedgewick Rafter Counting Chamber. Identification and counting were done under a light microscope (Zeiss Axio Scope.A1) at magnification factor 50 (crustaceans) or 100 (rotifers). Multiple parallel chambers were inspected until at least 100 individuals of the most abundant species were counted. Female cladocerans were determined to species level where possible. Males and juveniles were identified to genus level or higher. The abundance of large and rare cladocerans (e.g., *Leptodora kindtii*) was determined by flushing the complete sample into a petri dish and examining it under a dissecting microscope. Adult copepods were mostly identified to the species level. Individuals in early development phases were distinguished into cyclopoid and calanoid nauplii or copepodites, but not identified to species level. Other groups that were counted include nematodes, tardigrades, Chironomidae, Ostracoda, and *Chaoborus* larvae, *Dreissena polymorpha* larvae, and *Diffugia*.

The most abundant species are given in Table 2.6. The individual abundance was considered as abundant if at least at one date, a sample contained more than 2,000 individuals. The strong fluctuations are noteworthy: Within 14 days, some species disappear completely or are suddenly highly abundant.

Species	Lake Müggelsee					River SGT			River SNZ			
	Sep-15	Sep-29	Oct-13	Oct-27	Nov-10	Sept-02	Oct-14	Nov-25	Sep-02	Sep-16	Oct-14	Nov-25
<i>Ankyra</i> sp.	H	H			L							
<i>Asterionella formosa</i>			L	L		H	L	L	L	L	L	L
<i>Aulacoseira granulata</i>	L	H		H	H	H	H	H	H	H	H	H
<i>Chlamydomonas</i> sp.		L			H	H	L	H	H	H	H	
<i>Fragilaria acus</i>				L		H	L	L	L	L		H
<i>Fragilaria ulna</i> f. <i>angustissima</i>						H	L	H	H	H	L	L
<i>Kephyrion</i> sp.			H									
<i>Nitzschia fonticula</i>	H	H										
<i>Oocystis</i> sp.		L	L			H	H	L	L			
<i>Raphidocelis</i> sp.				H								
<i>Scenedesmus</i> sp.						H		H	H	H	H	H
<i>Skeletonema</i> sp.		L		H	H	H	H		H	H	L	L
<i>Synura</i> sp.							H					

Table 2.6: List of species occurring in at least one sample in *high* abundance. Abundance was designated as high if more than 2,000 individuals were found in a single sample, otherwise it was designated as low, i.e., 1 to 1,999 individuals, or left blank, i.e., zero individuals.

### Sample Processing for Metabarcoding

Samples for DNA metabarcoding analysis were collected from the same plankton samples as used for the morphological analysis. Phytoplankton DNA was obtained by filtering a 50-500-ml subsample of water (depending on plankton density, as estimated by eye) through a glass fiber filter (GF/F 25 mm diameter) using a vacuum filtration at 200 mbar. Zooplankton intended for DNA metabarcoding analysis were collected from 10 L of water by vacuum filtration at 200 mbar (30  $\mu$ m mesh) followed by filtering through a glass-fiber filter using vacuum filtration at 200 mbar. The glass-fiber filters with the residual plankton were then freeze-dried (Alpha 1-4, Martin Christ Gefriertrocknungsanlagen GmbH, Osterode am Harz, Germany) for 8 hours at -45 °C and then stored at -20 °C until DNA extraction.

## Primers

We PCR-amplified all samples with three sets of primers targeting the V4 region of the 18S rRNA gene (see Table 2.7). Two primer sets target a broad range of eukaryotes: EUK15 by Stoeck et al., 2010 primers, which has been shown to target all groups of Eukaryota except Excavata and Microsporidia in pelagic marine samples and has an expected amplicon length of ~380 nt. The primer pair EUK14 by Hadziavdic et al., 2014 amplifies a larger fragment (~630 nt) and has previously been used to target marine sediment communities. The third set was designed to target diatoms (Bacillariophyta) in freshwater ecosystems (Visco et al., 2015) and is referred to as DIV4 with an expected amplicon length of ~280 nt.

Marker ID	Name	Sequence (5' - 3')	Reference
EUK15	TAREuk454FWD1 TAREukREV3	CCAGCASCYCGGGTAATTCC ACTTTCGTTCTTGATYRA	Stoeck et al., 2010
EUK14	F-566a R-1200	CAGCAGCCGCGGTAATTCC CCCGTGTGAGTCAAATTAAGC	Hadziavdic et al., 2014
DIV4	DIV4for DIV4rev3	GCGGTAATTCAGCTCCAATAG CTCTGACAATGGAATACGAATA	Visco et al., 2015

Table 2.7: Primers for metabarcoding on freshwater plankton samples. For the EUK15 primers multiple sequence variants were deployed; the ambiguous symbols are IUPAC-encoded (see Table B.1).

## *In Silico* PCR

We first investigated the feasibility of DNA metabarcoding for the detection of species present in Lake Müggelsee and River Spree using an *in silico* PCR. The goal was to verify that a given morphologically identified species could be detected using our primer sets (see Table 2.7) when using the reference database to assign taxonomic names to OTUs. We considered *in silico* PCR for a species to be successful if *all* of the following three conditions were met:

- (i) three or fewer mismatches between primer and database sequence.
- (ii) a relative primer-template match of more than 80 %.
- (iii) the resulting amplicon length was 30-1000 bp

We built a locally searchable database from GenBank's nucleotide collection<sup>30</sup> (*nt* dataset), and the SSU subset of SILVA<sup>31</sup> using the `makeblastdb` command (`ncbi-blast` tool suite version 2.6.0+). Our database was indexed by accession numbers because indexing by species names or taxonomic IDs is not supported. In a pre-processing step, we first collected the accession numbers for each candidate species and then ran `blastn` queries for forward and reverse primer sequences. We used the BLAST output format 6 which gives the start and end positions and the number of mismatches. The script and setup instructions are provided on GitHub<sup>32</sup>.

<sup>30</sup><ftp://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/nt.gz>, downloaded on 5 April 2017

<sup>31</sup><https://www.arb-silva.de/browser>, Eukaryota branch, downloaded 28 April 2017

<sup>32</sup><https://github.com/mariehoffmann/greentools>

### DNA Extraction, PCR, Library Preparation, and Sequencing

Filters were loaded into Eppendorf tubes containing sterile metal beads, covered with TissueLyser (Qiagen GmbH, Hideln, Germany), and shaken three times for 5 minutes and 30 seconds each. The tubes were then spun briefly to collect the cells at the bottom. DNA was extracted from 0.5 g of the homogenized cells using a NucleoSplin®Plant II extraction kit (Macherey-Nagel GmbH & Co. KG, Düren, Germany) according to the manufacturer's protocol. Extracted DNA was stored in TE buffer at 20 °C until further analysis.

DNA from phyto- and zooplankton samples were first combined equimolar for each sample. For EUK15 and DIV4 reactions, template DNA (5 ng) was combined with 5  $\mu$ l reaction buffer (Q5, New England Biolabs, Ipswich, MA, USA), 0.625  $\mu$ l dNTP mixture (New England Biolabs), 1.25  $\mu$ l of each primer (see Table 2.7), 0.125  $\mu$ l proof-reading polymerase (Q5 High Fidelity, New England Biosystems) and RO-filtered water to yield a total reaction volume of 25  $\mu$ l. PCR (98°C for 30 s; 25 cycles of 98°C for 10s, 57 °C for 30 s, 72 °C for 30 s; 72 °C for 2 min) products were checked by eye on 2 % agarose gels to ensure successful amplification. Products were cleaned with a magnetic bead protocol (Agencourt AMPure XP, Beckman Coulter, Indianapolis, IN, USA) in accordance with the manufacturer's instructions. A second PCR reaction attached unique 12-bp inline sequence barcodes (Nextera Index Kit, Illumina, San Diego, CA, USA) to each sample. PCR was performed as above except that 5  $\mu$ l of PCR product was used as a template, and 8 PCR cycles were used. PCR products were purified as above, and the DNA concentration was determined using a Qubit 2.0 fluorometer (Thermo Fisher Scientific, Waltham, MA, USA). All samples were then pooled equimolar and sequenced on a MiSeq (Illumina) using v. 3 chemistry and 600 cycles.

For EUK14 reactions, PCR was applied as written above except that we used 20 ng of the template and 25 cycles of amplification. The biotinylated PCR products were enzymatically fragmented by the transposase provided by the Nextera XT Kit (Illumina), and then all fragments with intact forward ends (containing biotin) were recaptured by Streptavidin beads (DYNAL Dynabeads™ M-270 Streptavidin, Invitrogen™). These fragments were purified and subsequently amplified by the second index PCR as described in the Nextera XT protocol, which also introduces barcodes and sequencing adapters, although with four additional cycles (16 cycles in total). A final size-selection step was carried out to define the desired fragment length (430-530 bp, including ~130 bp adaptors) using an automated fragment size extraction (BluePippin, Sage Science, Beverly MA, USA). Raw sequencing data (FASTQ files) are available on the Sequence Read Archive (BioProject accession number PRJNA526363).

### Bioinformatic Analyses

Raw sequence data were processed using Galaxy v0.0.5 with the Genome Space Exporter tool<sup>33</sup>. Sequences were sorted by sample, and reads were merged using PEAR v 0.9.10 (Zhang et al., 2013). Poor-quality ends were trimmed (based on a probability limit of 0.02) using Geneious Pro v5.6.1 (BioMatters, Auckland, New Zealand). Sequences were then quality filtered using USEARCH v 8.0 (Edgar, 2010) and reads with expected error greater than 0.5 were removed (`fastq_maxee 0.5`). Sequences were then trimmed to a length of 270 bp (EUK15), 290 (EUK14) or 280 (DIV4) bp (`fastq_trunclen`). Reads were dereplicated (`derep_prefix`), merged into a single file, and renamed according to sample ID in Geneious Pro v5.6.1. Singleton

<sup>33</sup><https://usegalaxy.org>

sequences were removed, and the remaining sequences were clustered into OTUs by the UPARSE algorithm of USEARCH with a 97 % sequence similarity threshold. Putatively chimeric sequences were removed by the *de novo* and reference-based filtering algorithm of USEARCH with the SILVA (Quast et al., 2012) dataset as a reference. Taxonomic identities were initially assigned using USEARCH with the SILVA database as a reference database. The OTUs obtained were additionally checked manually using `blastn` queries against the NCBI reference database (queried in May 2017) to ensure that the top 10 sequence hits confirm the taxonomic identity. Following Lindeque et al., 2013, OTUs were assigned species names when they were >97 % similar to a reference sequence and were assigned genus names when they were >95 % similar to the reference sequence. When OTUs were >97 % similar to more than one reference sequence, the RDP (<http://rdp.cme.msu.edu>) classifier (Wang et al., 2007) was used to provide a consensus taxonomic identification.

### Statistical Analyses

To visualize how well the PCR captured the sample's diversity, we carried out a rarefaction analysis, which plots the number of reads against the obtained OTUs as a representative for species richness. A curve that reaches a plateau correlates to the situation where adding more sequence data, does not add to the species richness, whereas a slope significantly larger than zero indicates insufficient DNA yield to capture its diversity adequately.

The rarefaction analysis was carried out by Tatiana Semenova-Nelson<sup>34</sup> using the R package `vegan` (v2.5-6) by Oksanen et al., 2012 for R v3.1.2 (R Core Team 2018). As can be seen in Figure 2.13, all curves reach a plateau, which confirms that a sufficient amount of DNA has been extracted.

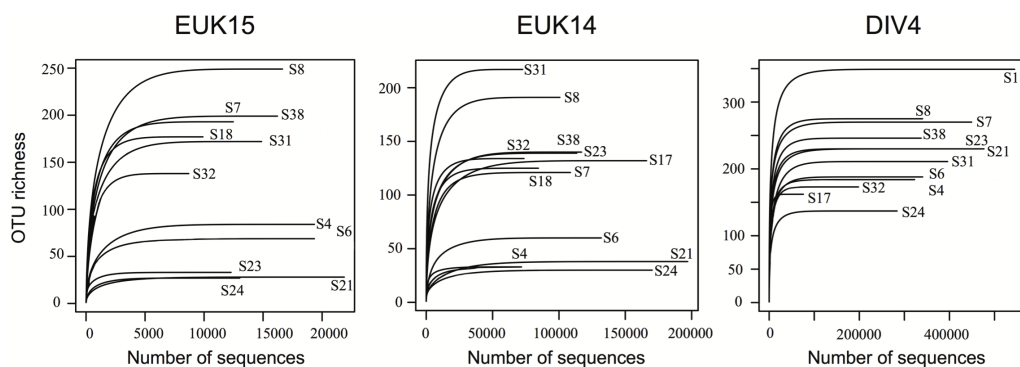


Figure 2.13: Rarefaction curves obtained for plankton OTUs for the three primer pairs (EUK15, EUK14, DIV4) at three sampling sites (SNZ, SNT, Lake Müggelsee). The curves reached a plateau suggesting that most of the plankton diversity has been captured by sequencing.

To answer the third question about the ability of each method and marker to differentiate the sampling sites based on the samples' compositions, we split the data sets into three major groups – diatoms, green algae, and zooplankton. Each site represents a class and OTUs its feature set.

Non-metric multidimensional scaling (NMDS) is a suitable method to embed the high-dimensional samples in a low-dimensional space. If the samples are separable by their OTU composition, they are likely to form clusters even in low-dimensional space. Similar to the principal component analysis (PCA), the high-dimensional

<sup>34</sup>co-author of study



feature space is represented by fewer linear combined axes. While a PCA ranks its linearly combined features by eigenvalues and requires numerical constraints that are not met<sup>35</sup>, NMDS attempts to place a few axes (typically two or three) in the feature space, such that stress<sup>36</sup> is minimal. The initial orientations are chosen randomly, and axes are rotated in the direction of less stress. If multiple reruns converge to similar solutions, the procedure stops. A further taxonomic breakdown was not possible as it would yield group sizes with low statistical significance. For example, the zooplankton discovered by DIV4 (29 taxa) is already too small for the NMDS procedure to converge to a solution.

To avoid false positives, OTUs with less than four reads in all samples were removed from NMDS analysis (see Lindahl et al., 2013). The dataset was subjected to 200 iterations per analysis using the Bray-Curtis dissimilarity, two axes, and a random starting number. The final stress values are indicated in each plot (see Figure 2.17).

The statistical difference between plankton communities of the lake and the two river locations was assessed using a multi-response permutation procedure (MRPP) performed pairwise on all sampling sites. Null rows were deleted before subjecting the subsets to 500 permutations with the Bray-Curtis index as a similarity measure. We reported the statistical significance (p-value), i.e., the fraction of shuffled data sets with similarity scores below the originally labeled one. NMDS and MRPP were carried out using the `metaMDS` and `mrpp` functions in the `vegan` package for R (see Figure 2.17) and code in Appendix A.2.

### 2.5.3 Results

#### Morphological Identifications

There were 235 eukaryotic taxa (170 phytoplankton, 65 zooplankton) in our morphological data set (see Table 2.8 and Table S2\_Morph.csv<sup>37</sup>). Of these, 146 (62 %) were identified to species level, 60 to genus level, and the remaining 29 were identified to higher taxonomic levels, to non-taxonomic, operational classifications, or different juvenile life stages. The four most diverse groups were Chlorophyta with 82 taxa, Bacillariophyta with 51 taxa, crustaceans with 36 taxa, and rotifers with 22 taxa (see Table S2\_Morph.csv).

	Taxa/OTU	Genus Level ID	Species Level ID
Morph ID	235	206 (88 %)	146 (62 %)
EUK15 ID	325	277 (69 %)	128 (40 %)
EUK14 ID	506	268 (53 %)	166 (33 %)
DIV4 ID	543	344 (63 %)	231 (43 %)

Table 2.8: Number of taxa (Morph ID) and OTUs (molecular ID) found and identified to genus or species level. Note that the counters are hierarchical, i.e. genera counts include species.

<sup>35</sup>like normalized OTU counts, low variance, or unique rows.

<sup>36</sup>Stress is here defined as differences in distances between data points before and after the transformation.

<sup>37</sup>available under <https://github.com/mariehoffmann/aquamarine>

The phytoplankton communities from Lake Müggelsee and the River Spree differed in composition when considering the most abundant taxa, i.e., at least 2,000 individuals had been counted in a single sample (see Table 2.6). *Ankyra* sp., *Asterionella formosa*, *Nitzschia fonticula* (Bacillariophyta), *Kephyrion* sp. (Chrysophyta), and *Raphidocelis* sp. (Chlorophyta) were all abundant in one or more samples from Lake Müggelsee, whereas *Chlamydomonas* sp. (Chlorophyta), *Fragilaria* spp. (Bacillariophyta), *Oocystis* sp., *Scenedesmus* sp. (Chlorophyta), and *Synura* sp. (Ochrophyta) were abundant in one or more samples from River Spree. Only *Aulacoseira granulata* and *Skeletonema* sp. (Bacillariophyta) were abundant in at least one river and lake sample.

### *In Silico* PCR

We used *in silico* PCR to examine whether the 141 morphologically identified species in our data set could have been recovered using DNA metabarcoding with our chosen primer sets. Specifically, we tested whether these species had a reference sequence in the NCBI or SILVA database, contained suitable primer-binding sites that matched at least one of our chosen primer sets, and met our fragment-length criteria. For 71 of the 144 morphological species identified, there were no entries in either database for the 18S rRNA gene based on a search using the binomial name (see Table 2.9). For the remaining 70 species that could be tested, the success rate of *in silico* PCR was high ( $\geq 96\%$ ) for both EUK14 and EUK15 primer sets, and the DIV4 primer set performed equally well for diatoms (95%). DIV4 primers also recovered a large majority (85%) of the Chlorophyta in our morphological data set. DIV4 had a low performance when attempting to recover all taxa (68%), but this was to be expected for primers developed specifically for diatoms (Bacillariophyta).

Marker ID	Species		<i>in silico</i> PCR	
	Morphologically identified	Reference available/missing	Success	Failure
EUK15	144	70/74	68 (97 %)	2 (3 %)
EUK14	144	70/74	67 (96 %)	3 (4 %)
DIV4	144	70/74	48 (68 %)	22 (32 %)

Table 2.9: Results of *in silico* PCR for each primer set tested. Of the 144 taxa identified morphologically to species-level (without hybrids), 70 had database entries in either SILVA or NCBI's *nt* dataset. *In silico* PCR success and failure rates were therefore calculated as % of the 70 species that could have been recovered. Primer DIV4 exhibits a lower success rate on the complete species set (68%), but performs well on important phytoplankton groups.

### DNA Metabarcoding

Sequencing yielded 20,285,523 reads, of which 6,655,909 passed our quality controls. Clustering at 97% similarity yielded 325 (EUK15), 506 (EUK14) and 543 (DIV4) OTUs. The rarefaction curves reached a plateau for all samples (see Figure 2.13), suggesting that all OTUs present in the samples were sequenced. When compared to the 235 taxa identified morphologically, OTUs resulted in increases of 1.4-fold to 2.3-fold depending on the primer set (see Table 2.8). More than half of the OTUs were identified at least to genus, and more than one-third to species level (see Table 2.8).

We investigated further how relative abundances are represented by each method, and accumulated read counts hierarchically until level<sup>38</sup> four for phytoplankton, level eight for zooplankton, and level five for fungi (see Figure 2.14). Note that many high-abundant OTUs cannot be resolved to lower levels than indicated by the taxonomic name (e.g., Chlorophyta, or Oomycota). Abundances below 4 % were dropped for the sake of readability. The discs' diameters correspond to the number of individuals (Morph ID) or read abundances and are  $\log_{10}$ -scaled. For example, from the class of Cryptophyta, the most dominant family is Pyrenomonadaceae, according to the LM identification. Whereas the family of Cryptomonadaceae is most dominant for PCR amplification primer sets EUK15 and EUK14. The diatom-specific primer set outperforms the other two markers on the phytoplankton subset in terms of read count number and captures well Chlorophyta and Coscinodiscophyceae. In contrast, in terms of diversity, EUK14 recovered a broader diversity of taxa than any other method (morphology, DIV4, EUK15) (see Figure 2.14). The EUK14 primer set also recovered a higher richness of fungi compared to the other two markers (see Figure 2.14) and a high diversity of ciliates (89 OTUs in 42 genera) making them the group of zooplankton with the highest taxonomic richness in our OTU dataset.

When considering the zooplankton, morphological analysis revealed that the Panarthropoda represent the largest group in terms of biomass, as do the Rotifera and Tintinnidae. Molecular identification revealed mostly Eumetazoa but lacked further resolution either due to missing references or indistinct barcodes.

Fungi remain completely undetected in LM because the protocol is designed for phyto- and zooplanktoners - magnification factors above 1,000 are not applied. PCR amplification leads inevitably to the emerge of large amounts of fungi reads. It is remarkable that all three primer pairs have matches in Oomycota and fungi, but produce differently distinctive transcripts. DIV4 seems to be very sensitive and effective for fungi. However, the barcode is identical for all individuals in this kingdom. EUK14, on the other hand, provides better resolution and identified 10 oomycota and fungi down to species level.

Besides, we accumulated individual, and OTU read counts for relevant plankton groups like diatoms or Chlorophyta irrespective of their level in the taxonomic hierarchy (see Table 2.10). The DIV4 primer set recovered a ca. 1.5-fold increase in diatom diversity and a 1.6-fold increase in green algal diversity compared to morphological analysis (see Table 2.10). The DIV4 primer set also recovered more OTUs in these groups compared to EUK15 and EUK14 primer sets (see Table 2.10) and also a greater diversity of higher taxa in the groups Chlorophyta, Coscinodiscophyceae, Chrysophyceae, and Synurales (see Figure 2.14). Interestingly, fewer genera of diatoms were identified using DIV4 primers (10) compared to morphology (12); however, many more diatom OTUs could be identified to species level (see Table 2.10). In contrast, most of the Chlorophyta OTUs could not be identified to species, and there was only a 13 % increase in the number of species identified with DNA compared to morphology (see Table 2.10). In diatoms, 20 species in 13 different genera were found only with DIV4, while only three taxa that were morphologically identified to genus (*Diploneis* sp., *Rhizosolenia longiseta*, and *Rhopalodia gibba*) were missing, all of which were found in low abundances in the morphological data set.

The most substantial increase in diversity when comparing morphology to DNA was observed in the golden-brown algae (Chrysophyta), where the DIV4 primers recovered a 16-fold increase in the number of OTUs (see Table 2.10). Only 18 of 158 taxa could be determined to species and seven others to genus (see Table 2.10). DIV4

<sup>38</sup>given NCBI's taxonomy in April 2019, with Eukaryota assigned to level one



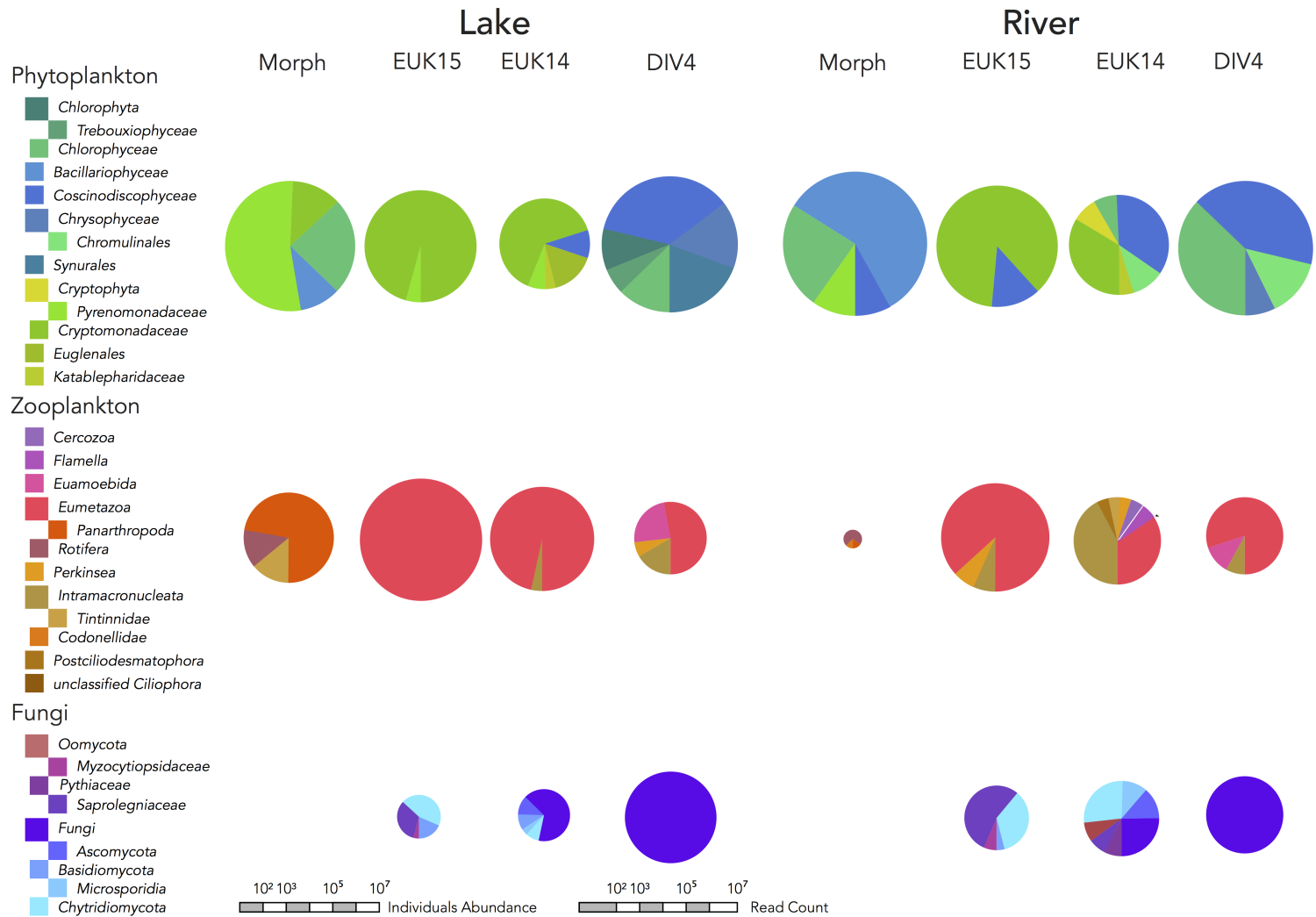


Figure 2.14: Abundances by method, location, and higher-order groups. Counts were conducted at taxonomic depth four (phytoplankton), five (fungi), or eight (zooplankton). Group abundances below 4 % were dropped for the sake of readability. The diameter corresponds to  $\log_{10}$ -scaled abundance (Morph ID) or read counts (EUK15, EUK14, and DIV4 ID). Note how strongly each marker is targeted to specific taxa. Metabarcoding revealed a considerable abundance of fungi and oomycota.

sequences recovered the three species of golden-brown algae in our morphological data set (*Synura uvella*, *Mallomonas akromonas*, and *Dinobryon divergens*), as well as 15 other OTUs, identified to the species level. The other primer sets uncovered more diversity for cryptomonads (EUK15), euglenids, and dinoflagellates (EUK14), although DIV4 recovered the highest diversity of Charophyta and Eustigmatophyta (see Table 2.10).

Not surprisingly, the marker-based approach yielded greater diversity. But how are the most common species identified by morphology represented in the marker-based approaches? In case a species showed up in the identification set of one of the markers, we would like to know if their abundance patterns are comparable. In the event that a species appears in the identification group of one of the markers, we would like to know if their abundance patterns are comparable. Eight of the 12 most abundant species could be identified with at least one marker. In the remaining three cases, there was insufficient database backup or an indication that the reverse primer did not match the template (see Table 2.11).

Four co-identified species could be determined by their binomial name (*Asterionella formosa*, *Aulacoseira granulata*, *Fragilaria acus*, and *Fragilaria ulna f. angustissima*) and we plotted their abundances plotted over time (see Table 2.6). For species without binomial names, counts were accumulated whenever a marker identified a species in the same genus. The full set of charts can be found in the appendix A.1.

The individual counts (Morph ID) and OTU read counts (EUK15, EUK14, and DIV4 ID) are not comparable *per se*, but similar shapes of their abundance distributions would suggest a correlation and the possibility that metabarcoding could be suitable for abundance estimation. We would expect that high abundant species with complete binomial names are less prone to PCR biases, and counts, therefore, more suitable for method-wise comparison.

As can be seen in Figure 2.15, the markers EUK15 and DIV4 which identified *Asterionella formosa* follow the upward trend in the first half of November. At the river site SGT, all markers show a downward trend, which is not given by the morphological identification, because only one individual was counted for the specified volumes. More water would need to be sampled to reveal the actual distribution. When looking at *Aulacoseira granulata*, there is an upward trend towards the 10th of November, whereas the markers EUK15 and DIV4 have peaks in mid of September and the 1st of November and then start to decline. All markers exhibit a slight decrease for the river sites, which cannot be seen for the morphological identification due to the too-small individual count. For *Fragilaria acus*, DIV4 follows the shape in November; for previously collected samples, the individual counts were too small. At the SGT river site, marker DIV4 followed the decline in individuals but with a lag, and at the SNZ river site the trends are slightly opposite. *Fragilaria ulna f. angustissima* occurs in high abundances at both river sites in September and almost disappears around mid-October. No individuals were found in the lake. However, marker EUK15 must have amplified some spurious DNA of the same species as the marker exhibits the same trend. The marker largely agrees in the species decline between September and December (see Figure 2.15).

### Classification Failures of DNA Metabarcoding

Under the assumption that a species is known to be present in a sample and has 18S sequences in the database, there are two frequent causes for DNA metabarcoding to miss the identification. Firstly, the genome may not provide sufficiently matching primer binding sites. Secondly, the clustering may lead to the absorption of the

Taxon	Identification Method	Taxa/OTU	Genus Level ID	Species Level ID
Bacillariophyta (Diatoms)	Morph	51	12	33
	EUK15	18	5	15
	EUK14	22	3	15
	<b>DIV4</b>	101	10	72
Charophyta	Morph	9	4	5
	EUK15	3	2	1
	EUK14	0	0	0
	<b>DIV4</b>	11	3	8
Chlorophyta	Morph	82	12	58
	EUK15	40	7	29
	EUK14	23	18	20
	<b>DIV4</b>	128	20	66
Chrysophyta	Morph	9	3	1
	EUK15	41	5	7
	EUK14	55	6	7
	<b>DIV4</b>	158	7	18
Cryptophyta	Morph	3	2	1
	EUK15	26	4	9
	EUK14	23	3	9
	DIV4	7	3	3
Dinoflagellata	Morph	5	2	2
	<b>EUK15</b>	14	1	8
	EUK14	12	2	7
	DIV4	7	1	3
Euglenida	Morph	6	3	1
	EUK15	0	0	0
	<b>EUK14</b>	6	1	1
	DIV4	0	0	0
Eustigmatophyta	Morph	4	2	2
	EUK15	0	0	0
	EUK14	1	0	1
	<b>DIV4</b>	8	2	4

Table 2.10: Plankton community characterization using morphological assessment and DNA metabarcoding with three different primer sets (EUK15, EUK14, DIV4). For each taxonomic group, the total number of entries (taxa for morphology and OTUs for DNA metabarcoding) is given. In addition, we counted the number of entries that are identified to the genus- and species-level. We considered species-level identifications for OTUs when a BLAST search of NCBI yielded a match of  $\geq 97\%$  and genus-level identifications for matches of  $\geq 95\%$ . Methods written boldly indicate the method resulting in the most entries/OTUs and highest count of species-level identifications. Tatiana Semenova-Nelson originally compiled this table.

Species	EUK15	EUK14	DIV4	Comment
<i>Ankyra</i> sp.	1	1	1	All markers identified <i>Ankyra judayi</i> .
<i>Asterionella formosa</i>	1	0	1	EUK14 match to unidentified <i>Asterionella</i> sp.
<i>Aulacoseira granulata</i>	1	0	1	EUK14 match to unidentified <i>Aulacoseira</i> sp.
<i>Chlamydomonas</i> sp.	1	1	1	All markers identified multiple species.
<i>Fragilaria acus</i>	0	0	1	Only marker DIV4 identified this species.
<i>Fragilaria ulna</i> <i>f. angustissima</i>	1	-	-	EUK15 identified <i>Fragilaria ulna</i> .
<i>Kephyrion</i> sp.	0	0	0	There exists only a single partial 18S sequence in GenBank of an unclassified <i>Kephyrion</i> . All forward, but non of the reverse primers matched.
<i>Nitzschia fonticola</i>	0	0	0	There were four 18S sequences marked as partial. A dozen other <i>Nitzschia</i> species were identified by the markers, e.g. <i>N. acicularis</i> or <i>N. longissima</i> .
<i>Oocystis</i> sp.	1	1	1	All markers identified <i>O. heteromucosa</i> and <i>O. marssonii</i> .
<i>Raphidocelis</i> sp.	0	0	0	More than 100 accessions are registered to the genus <i>Raphidocelis</i> . In a randomly chosen accession all forward primer sequences were found, but none of the reverse primers.
<i>Skeletonema</i> sp.	1	1	1	All markers identified at least one species.
<i>Synura</i> sp.	1	1	1	All markers identified multiple species.

Table 2.11: Most abundant species as seen under the light microscope and indication (1: yes, 0: no) whether one of the marker-based approaches identified the species or other species of the same genus.

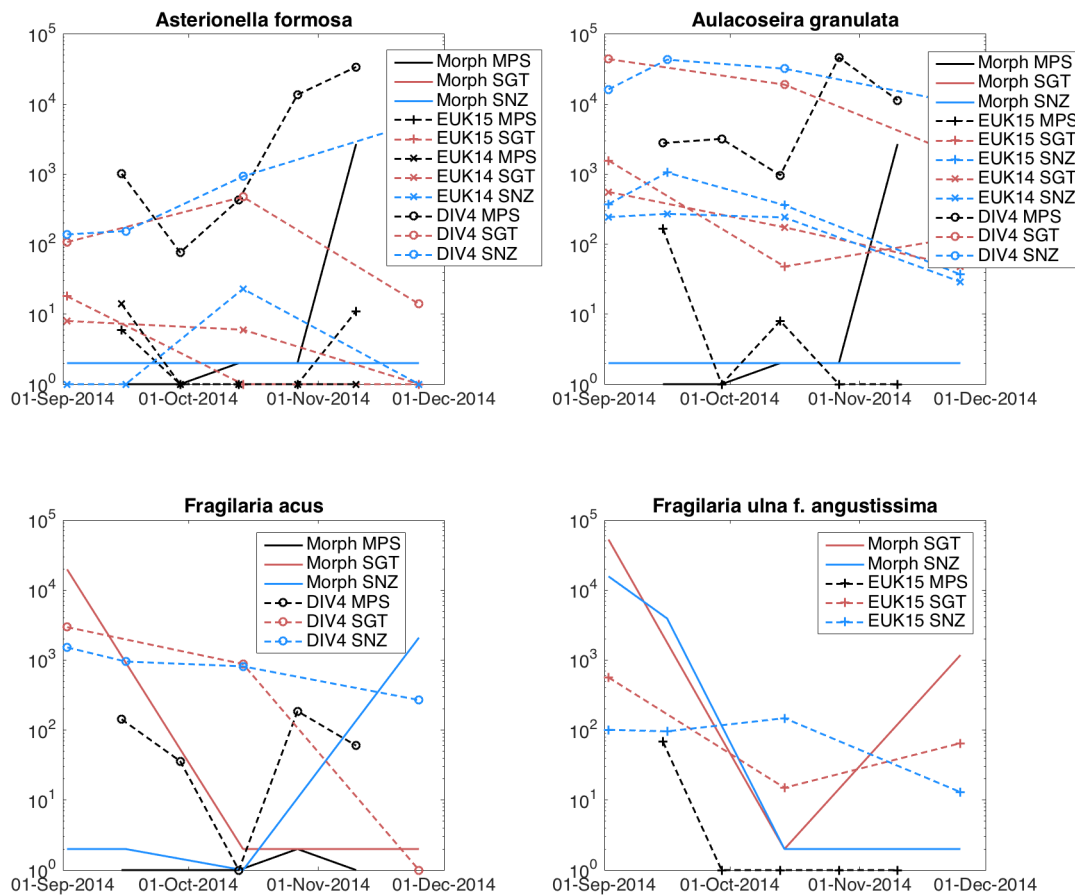


Figure 2.15: Abundance plots of four of the most abundant species according to the morphological identification. Identical colors refer to the same sampling site (black for the lake, red for river SGT, and blue for river SNZ). A solid line represents individual counts from the morphological analysis, otherwise, OTU sizes (number of reads per OTU), i.e., a dashed line with a plus symbol to EUK15, with an x to EUK14, and with a circle to DIV4.

barcode carrying read by an OTU that has higher proximity to a different species or cannot be resolved at all.

Whereas the first cause can only be solved by designing new primer sequences, the second one may be triggered by the sparseness of the reference database, or by the bioinformatics protocol. We were interested in reasons that lead to failure despite 18S sequences being available with primers matching. Concretely, we looked at the 35 species that had been morphologically identified, had at least one 18S sequence in the database, but had not been recovered by metabarcoding. *In silico* PCR indicated these taxa could be PCR-amplified with at least one of the primer sets (see Table 2.9). We ensured that at least one of the three primer pairs matched with 100 % sequence identity.

We collected reference sequences for a subset (17 randomly chosen species out of the 35) and created for each species an MSA using PRANK v.170427 (Löytynoja, 2014). A manual comparison of interspecific and intra-specific differences revealed that nine of these 17 species would have been identified had we used a higher clustering threshold for OTU formation. For example, a 97.1 % threshold would have distinguished *Rhodomonas lens* from its close relatives, and 99.1 % would have distinguished *Monoraphidium contortum*. For the diatom *Staurosira construens* (also known as *Fragilaria construens*), there were species-specific differences at the 5' end of the sequenced fragment, but a threshold approach that uses mean sequence difference failed to separate *F. construens* from other species of the same genus (*F. crotonensis*, *F. bidens*, *F. capucina*, *F. nanana*, *F. vaucheriae*, *F. ulna*). Another six of the 17 species also had sequences highly similar to other taxa in the database (*Eudiatomus gracilis*, *Scenedesmus acuminatus*, *S. armatus*, *S. dimorphus*, *S. subspicatus*, *S. ovalternus*). Setting the similarity threshold for clustering even higher (e.g., > 99 %) would allow for improved resolution of *Scenedesmus*. For example, *Scenedesmus dimorphus/acutus* would be distinguishable from *Scenedesmus communis/acuminatus*. Two of the 17 species contained poor quality or potentially misclassified sequences (*Ceratium hirundinella*, *Rhodomonas lens*).

Among the 71 morphological species without references (see Table 2.9), there were 39 cases in which an OTU was assigned to the same genus and a different species. This finding suggests that identification at the genus level would be possible in > 70 % of the cases in our data. Finally, there were 32 species with no reference and no OTUs assigned to the genus. In practice, these 32 species are likely to be assigned to a higher taxonomic level, but could potentially be better studied with more reference data.

### Community Composition in Lake Müggelsee using Multiple Methods

We ranked the top 25 taxa by total abundance over all samples from Lake Müggelsee (individual counts for Morph ID, otherwise OTU read counts). The ranking yielded a surprisingly similar overview of the lake community, especially when comparing morphological data and EUK15 OTUs, where both methods yielded similar numbers of OTUs from most of the same higher taxa (see Figure 2.16). In many cases, the data were not directly comparable because the classifications differ and are often less precise for the morphological data (see Figure 2.16). According to the morphological data, the cryptophytes *Rhodomonas* and *Cryptomonas*, two chlorophytes (*Ankyra* sp. and an unclassified Chlorophyceae), and unclassified centric diatoms were the most abundant taxa. Using the EUK15 primer set, the second and fourth most abundant taxa were also *Cryptomonas* species (*C. curvata* and *C. marssonii*). One abundant OTU could not be classified (see OTU-15 in Figure 2.16). The dominant OTU based on

EUK15 sequencing reads was the copepod *Eudiaptomus vulgaris*, and for EUK14 was the copepod *Sinodiaptomus sarsi*. A subsequent comparison using a multisequence alignment of 18S rRNA V4 reference sequences (*E. vulgaris* NCBI accession JX945121; *S. sarsi* NCBI accession KR048711) revealed only a 1-bp difference between these two taxa (i.e., > 99 % sequence similarity). Other dominant groups for the EUK14 primer set were also the Cryptophyta *Cryptomonas curvata*, the crustaceans *Daphnia galeata* and *Sinodiaptomus sarsi*, the Ciliophora *Tintinnidium fluviatile*, and *Colacium* spp. from the phylum Euglenozoa (discovered exclusively by EUK14). As expected, the DIV4 primer set was biased towards phytoplankton groups. One-fourth of the reads were assigned to *Stephanodiscus* sp. (Bacillariophyta), and one-third of the reads were assigned to Chrysophyceae, most notably *Mallomonas caudata*. Just like the EUK15 primer set, DIV4 detected the Chlorophyta *Ankyra judayi* and *Desmodesmus communis*, but also *Pseudopediastrum alternans*, and seven more Chlorophyta within the top 25 taxa. A notable exception to the similarity of methods was the absence of Rotifera from all three DNA metabarcoding data sets, despite being an important component of samples based on morphological identifications (see Figure 2.16).

### Comparison of Sampling Sites

It was possible to distinguish samples from the lake and the river for at least one major organism group (diatoms, green algae, and zooplankton) by statistical tests. Distinguishability is indicated by spatial separation in the non-metric multidimensional scaling (NMDS, see Figure 2.17 and code in Appendix A.1) or significantly small p-values ( $P < 0.05$ ) computed pair-wise via the multi-response permutation procedure (MRPP, see code in Appendix A.2). In some cases, p-values were insignificant, although the NMDS procedure succeeded in spatially separating the lake and river samples (Morph ID for diatoms and green algae and EUK15 ID for diatoms, Figure 2.17), as indicated by stress values below 0.08 (with the exception of DIV4 for the green algae and zooplankton subsets).

The two river sites (Spree SNZ, Spree SGT), which are 30 km away from each other, were inseparable for all markers and subgroups when consulting the p-values, and separable by NMDS only in two marker combinations: DIV4 on the diatom and EUK14 on the zooplankton subset. The degenerated figure (top left) from the zooplankton data set (Morph ID) can be explained by an insufficient dataset size causing a non-termination of the NMDS procedure.

### Conclusion

The study aimed to evaluate the feasibility of DNA metabarcoding to delineate the taxonomic richness of plankton communities in two freshwater ecosystems that are the focus of long-term research programs. A first step in evaluating DNA metabarcoding data is to compare the results with data obtained from the morphological analysis. While the present study system is well suited for this purpose because of the exceptional taxonomic expertise of the data processors. It should be noted that morphological identification does not attempt to identify all species in a sample. Therefore, direct comparisons of overall diversity are not meaningful, but rather comparisons within specific groups.

In addition, identification by LM is unsuitable for some algal species. For example, minute centric diatoms and chrysophytes (such as *Paraphysomonas*, *Mallomonas*, and *Synura*) must be examined by electron microscopy. For some cryptomonad genera, species identification is possible only with spectrophotometric or molecular sequence



Higher taxon	Morphological ID	%	EUK15 ID	%	EUK14 ID	%	DIV4 ID	%
Chlorophyta	Chlorophyceae sp	*15.88	<i>Ankyra judayi</i>	0.11	<i>Mallomonas akrokomos</i>	0.37	<i>Ankyra judayi</i>	*10.29
	<i>Ankyra</i> sp	*7.80	<i>Desmodesmus communis</i>	0.05	Ciliophora	0.27	<i>Apatococcus</i> sp	2.19
	<i>Raphidocelis</i> sp	0.08	<i>Mallomonas akrokomos</i>	0.02	<i>Rimostrombidium lacustris</i>	0.34	<i>Chlamydomonas</i> sp	0.37
Chrysophyta	Chrysoflagellata sp	2.24	<i>Perkinsea</i> sp	0.04	<i>Strobilidium</i>	0.38	<i>Choricystis minor</i>	2.66
	<i>Kephyrion</i> sp	0.59	Perkinsidae sp	0.02	<i>Tintinnidium fluviatile</i>	*1.25	Chrysophyceae sp	2.01
Ciliophora	<i>Tintinnidium fluviatile</i>	0.24	<i>Strobilidium</i> sp	0.04	<i>Bosmina longirostris</i>	0.24	<i>Desmodesmus communis</i>	0.55
	<i>Codonella cratera</i>	0.07	<i>Tintinnidium fluviatile</i>	0.45	<i>Daphnia galeata</i>	*7.77	<i>Pseudopediastrum alternans</i>	*7.26
Crustacea	<i>Bosmina</i>	0.06	<i>Rimostrombidium lacustris</i>	0.08	<i>Eurytemora affinis</i>	*1.19	<i>Scenedesmus</i> sp	1.71
	<i>Chydorus sphaericus</i>	0.03	<i>Cypridopsis uenoi</i>	0.09	<i>Mesocyclops leuckarti</i>	0.45	<i>Spermatozopsis similis</i>	0.44
	Copepoda (cyclopoid)	0.42	<i>Daphnia pulex</i>	0.14	<i>Sinodiaptomus sarsi</i>	*66.51	<i>Trebouxia</i> sp	0.68
	Copepoda (calanoid)	0.25	<i>Eudiaptomus vulgaris</i>	*69.99	<i>Thermocyclops</i>	1.04	Chrysophyceae sp	*10.73
Cryptophyta	<i>Daphnia spp</i>	0.33	<i>Mesocyclops leuckarti</i>	0.06	<i>Cryptomonas curvata</i>	*11.03	<i>Mallomonas akrokomos</i>	0.48
	<i>Eubosmina longicornis berlinensis</i>	0.12	<i>Temora</i> sp	*0.56	<i>Cryptomonas marssonii</i>	0.66	<i>Mallomonas caudata</i>	*18.05
	<i>Cryptomonas spp</i>	*12.22	<i>Thermocyclops</i> sp	0.32	Kathablepharidae	0.78	<i>Mallomonas tonsurata</i>	0.92
	<i>Rhodomonas minuta lacustris</i>	*49.81	<i>Cryptomonas curvata</i>	*22.57	<i>Plagioselmis nannoplantica</i>	0.26	<i>Paraphysomonas</i> sp	2.38
Bacillariophyta	<i>Asterionella formosa</i>	0.05	<i>Cryptomonas marssonii</i>	*1.28	<i>Plagioselmis</i>	0.29	<i>Spumella danica</i>	1.46
	<i>Aulacoseira granulata</i>	0.34	<i>Cryptomonas</i> sp	0.03	<i>Rhodomonas minuta</i>	1.25	<i>Actinocyclus actinochilus</i>	1.34
	Centric diatoms	*5.24	<i>Gemingera</i> sp	0.12	Cryptophyceae	0.40	<i>Asterionella formosa</i>	2.66
Mollusca	<i>Nitzschia fonticola</i>	4.09	<i>Rhodomonas minuta</i>	0.49	<i>Actinocyclus</i>	0.46	<i>Aulacoseira granulata</i>	3.17
	<i>Dreissena polymorpha</i> larvae	0.04	<i>Rhodomonas</i> sp	0.48	<i>Cyclostephanos tholiformis</i>	0.58	<i>Nitzschia sigma</i>	2.26
Rotifera	<i>Keratella cochlearis cochlearis</i>	0.02	<i>Actinocyclus actinochilus</i>	0.09	Colacium (Euglenozoa)	*2.85	<i>Skeletonema potamos</i>	1.95
	<i>Polyarthra major euryptera</i>	0.06	<i>Cyclostephanos tholiformis</i>	0.13	Bactrodes (Insecta)	0.59	<i>Stephanodiscus</i> sp	*25.01
	<i>Polyarthra vulgaris dolichoptera</i>	0.03	<i>Skeletonema costatum</i>	0.03	Unclassified OTU-15	0.65	<i>Synedra berlinensis</i>	0.36
	<i>Synchaeta oblonga</i>	0.03	Unclassified OTU-15	*2.80	Unclassified OTU-65	0.55	<i>Nannochloropsis</i> sp <sup>a</sup>	0.69
	<i>Synchaeta</i> sp	0.04	Unclassified OTU-65	0.04	Unclassified OTU-15	0.35	Fungi	1.25

<sup>a</sup>Eustigmataceae

Figure 2.16: The 25 most abundant organisms and OTUs for each method: morphological identification and metabarcoding with one of the three markers EUK15, EUK14, and DIV4. We placed organisms of the same supergroup adjacently. The figure shows that all four methods detected organisms assigned to Chlorophyta, Chrysophyta, and Bacillariophyta. We identified two more groups via morphological identification and the eukaryote targeting primers: Crustacea and Cryptophyta. Some top 25 organisms, however, are present in only one of the methods: Mollusca, Rotifera in the morphological column, unclassified OTUs for EUK15, and EUK14, *Colacium* (Euglenozoa) and *Bactrodes* (Insecta) exclusively by EUK14, and *Nannochloropsis* sp. (Eustigmataceae) by DIV4. Michael Monaghan originally compiled the table.



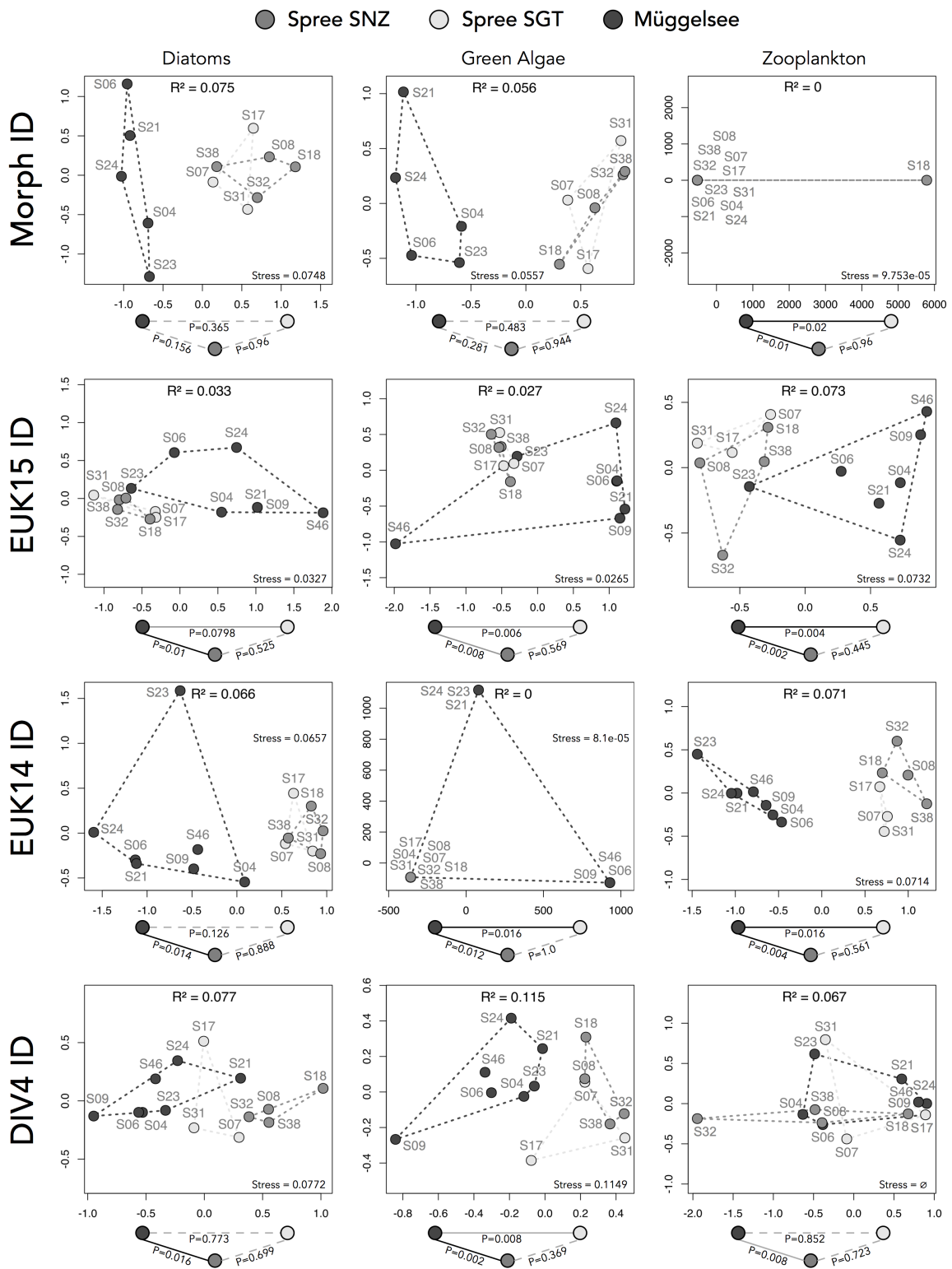


Figure 2.17: Statistical analysis of sample composition and sampling site subset by three major organism groups (diatoms, green algae, and zooplankton). The box plots show the results of non-metric multidimensional scaling (NMDS) with two axes. Final stress values are noted down in the plots. The p-values of the multiresponse-permutation procedure (MRPP) were computed pair-wise for each of the three sampling sites and are shown below each plot.

analyses. Many diatoms and green algae differ in their DNA sequences but are cryptic in their morphology. Some taxa remain undescribed or occur at a life cycle stage that cannot be revealed. These organisms are then grouped based on a few common morphological characteristics under the morphological identification protocol. Some of these morphological groups are polyphyletic, such as green flagellates, calanoid and cyclopoid nauplii, or calanoid and cyclopoid copepodites.

We used three primer sets to examine the difference in coverage and potential taxonomic resolution of different markers for plankton (Groendahl, Kahlert, and Fink, 2017; Clarke et al., 2017). DNA metabarcoding resulted in between 325 and 543 OTUs compared to 235 taxa of the morphological screening. More than half could be identified to genus level (one third to species level), and the diversity of most major groups increased almost 2-fold compared to morphological identifications.

No single marker was optimal for all taxonomic groups. DIV4 showed significantly increased diversity in diatoms, green algae and golden brown algae (with the exception of Cryptophyta). Both EUK14 and EUK15 recovered Dinoflagellata, Cryptophyta, Amoebae, Bicosoecida, Cercozoa, Ciliophora, Nematoda, and representatives of the Perkinsus clade. EUK14 recovered more OTUs and representatives of all supergroups of Eukaryota but encounters some challenges. First, the long fragment length did not allow for merging of reads in many cases and reads needed to be trimmed. Agreement based on *in silico* PCR was comparable with EUK15. A significant limitation was that EUK14 identified the lowest proportion of OTUs to genus and species, and failed to detect many OTUs within the Chlorophyta, an ecologically important group in the lake. We did not control the read number, which varied among markers and, to a lesser extent, sampling sites (see Figure 2.14), although rarefaction analysis suggested that OTUs were fully sampled for each primer set. The data set from the EUK15 primer also was most similar to the morphological community composition when considering the 25 most abundant taxa.

Metabarcoding captured the compositional separation of the lake and river communities at least as well as morphological screening as suggested by the results of the NMDS and the p-values from the MRPP. Half of the species that were exclusively identified by LM were not backed up by sequences in the reference database. Despite all three primer pairs produced more OTUs identified to species level than the morphological identification, there remain unidentified groups.

The barcode analysis of a selected set of organisms showed that the barcode differences between distinct species could be as low as one base. A bioinformatics pipeline must account for this. Using a 97 % sequence similarity threshold for OTU formation is standard, but inappropriate in metabarcoding. We hypothesize that for capturing the diversity of plankton, a denoising step combined with *dereplication* (or clustering with 100 % sequence identity) allows for even more taxa to be identified.

The ranking of the most abundant species (see Figure 2.16) and grouping by their phyla revealed commonalities between the morphological counting and the metabarcoding method. The fact that some species of the same genus have references in the database and others do not, occasionally manifest in OTUs that are resolved to particular species but may represent for real a different species that is not backed up. When comparing organisms on a higher taxonomic level, such errors blur out.

Comparing abundances on species level (see Table 2.11, Figure 2.15) remains difficult. We found occasionally similar trends between the binocular and molecular methods. An open question is when the method of morphological counting underestimates the number of individuals due to life stages that obscure morphological characteristics. For metabarcoding the question comes up when does the method

overestimates abundances due to missing references of species with identical barcodes? Such a question can only be answered in a probabilistic manner and requires a TOL that reflects phylogenetic distances well.

The enormous phylogenetic breadth of planktonic organisms requires the use of multiple DNA markers as the only way to fully characterize diverse planktonic communities (Pawlowski et al., 2012) as well as to compensate for the problem of incomplete reference databases for a given marker. For example, at the time of writing, no 18S sequences were available in GenBank that could be assigned to the common rotifer genera *Kellicottia*, *Keratella*, *Synchaeta*, and *Trichocerca*. However, identification of these taxa could still be possible using the CO1 barcode (Makino et al., 2017). Sequencing of the *tufA* region in addition to 18S would result in better species identification of the green algae *Scenedesmus* and *Schroederia* (Vieira et al., 2016), while primers targeting the 26S region could effectively delimit species of another green algae genus *Tetraedron* (Buchheim et al., 2005). Due to the continually growing number of markers needed to identify species in diverse plankton groups better, sequencing costs may rapidly accumulate. Therefore, whenever morphological assessments of the plankton are available, we recommend that *in silico* PCR tests be carried out prior to next-generation sequencing, to select the optimal combination of markers for cost-effective identification of the majority of present plankton species. The *in silico* PCR conducted in this study indicates that all but a few of our morphologically identified species could, in principle, be recovered using DNA metabarcoding. Reasons for the few missing taxa include, in part, clustering thresholds that were too low, perhaps a low abundance of DNA due to extraction bias, and PCR stochasticity that could not be assessed here.



## Chapter 3

# Primer Discovery in Large Datasets

Neue Wege entstehen, indem wir sie gehen.

---

*Friedrich Nietzsche*

### 3.1 Problem Definition

We have seen in the previous chapter and especially in Section 2.5 that one of the most significant challenges in metabarcoding is the choice of the barcode. In the search for a new barcode, we know that evolutionary pressure is not consistent within genomes. Some regions vary even within species, or on the other extreme, are conserved for whole clades. For example, functional regions are under higher evolutionary pressure and tend to be more conserved. When known, these regions can serve as markers to catch a wider taxonomic variety at the cost of a lower resolution, i.e., we may not be able to assign OTUs to species level. In contrast, non-coding regions expose more sequence variation and are better suited for delineating closely related species.

A barcode needs to be surrounded by conserved regions that contain suitable binding sites for primers (see Section 2.3). Flanking conservation is needed because we batch-process all extracted DNA at once and add only one primer pair per PCR run<sup>1</sup>.

High variant regions seem to be good candidates for barcodes at first glance but are unlikely to be backed up sufficiently by reference sequences. The immense diversity and presence of closely and very distantly related organisms in combination with a lack of complete genomes (see Chapter 2) makes it hard to find a marker (or a set of markers) that solves two opposing goals: sensitivity and specificity. We need markers that are effective within a broad taxonomic range for surveying and not being biased towards a few taxa and markers that accurately resolve important indicator species.

This problem is traditionally tackled by using a combination of primer pairs that have already been evaluated experimentally, and to alter them iteratively with the aid of bioinformatics analysis tools. A common approach is to select a few sequences of the clade of interest and to compute a multisequence alignment (MSA) as it helps to identify conserved regions (see Section 2.2.4). Conserved regions are then examined to determine whether they represent chemically suitable primer targets, and the region in between is examined for high similarity within a species. For single

---

<sup>1</sup>An exception is the multiplex PCR where multiple non-interfering primer pairs are added.

sequences there exists applications like Primer3 (Untergasser et al., 2012) to detect chemically suitable primers.

As remarked in more depth in Chapter 2, the MSA approach firstly fails when applied to phylogenetically distant sequences, and secondly does not scale linearly with the number of sequences. A 10-Liter freshwater sample from Lake Müggelsee (see Section 2.5) contains hundreds of eukaryotic plankton species from all major groups (see Figure 2.3) of the eukaryotes and besides bacteria, and fungi. The zooplankton clade Crustacea has more than 50,000 sequences in GenBank. Searching for a primer pair that aims to amplify a maximum of crustaceans is currently infeasible. In summary, there exists no tool that

1. accepts thousands of uncurated reference sequences
2. optimizes a primer candidate towards frequency which implies
  - (a) a broader effectiveness when applied to high-level taxa
  - (b) a higher resolution when applied to low-level taxa in combination with transcript analysis

The survey-like character of metabarcoding does not permit the restriction to a few, but curated databases like PR<sup>2</sup> (Guillou et al., 2012) or SILVA (Quast et al., 2012). These databases exhibit much lower taxonomic coverages than GenBank's nucleotide collection (*nt*) and would result in more OTUs remaining unidentified. Merging of reference libraries is also not possible, because they rely on different taxonomies. For an optimal OTU resolution we, therefore, rely on the largest publicly available sequence database, which yet has a relatively low coverage for microplankton taxa: thousands of taxa contain only one or two (mostly SSU rRNA) reference sequences assigned to it<sup>2</sup>. In the case of metabarcoding, it needs to be possible to consult the largest publicly available data sets in the search for frequent primers and markers.

Summarized, given an arbitrarily-sized set of references  $\mathcal{R} = \{R_1, R_2, \dots, R_m\}$ , the problem is to find a primer set  $P$  balancing both: high taxonomic coverage and high resolution. This goal can be captured by filtering for frequently occurring primers and ranking by coverage or variation, i.e., the number of unique barcodes. This chapter presents the software PriSeT (**P**ri**S**e**T** (**P**ri**S**e**T** **T**ool)), an offline primer discovery tool that is capable of processing large libraries and is robust against mislabeled, and low-quality sequences as an input source. PriSeT tackles the computationally expensive steps with linear runtimes for primer checks and space-efficient encodings. PriSeT was presented at the BCB'21 conference (Hoffmann, Monaghan, and Reinert, 2021) and briefly described in a cumulative publication on strategies to combat COVID-19 (Hufsky et al., 2020).

## 3.2 Introduction

The search for frequently occurring primer pairs conflicts with the search for marker regions that exhibit a high variation. Solving both tasks greedily in a sequential way is more feasible than solving them simultaneously. The two sequential approaches are:

- (a) First marker search, then primer search in the neighborhood
- (b) First primer search, then marker property verification of enclosed region

<sup>2</sup>as indicated by low mean accession counts per taxon in Table 3.10

In environmental genomics, we have to deal with missing ground truth, i.e., unidentified and unknown species, and a lack of complete genomes. The marker constraints have to deal with these uncertainties. We can, at most, require that a marker is rare or unique to a taxon given the incomplete state of a reference database. There is no guarantee that a marker has no co-occurrences outside the backed-up region within the same or other species. This chance, however, decreases with the length of the marker. To give an example of the known sequence proportion: the estimated genome size of the diatom *Skeletonema costatum* is about 51 Mbp (Ogura et al., 2018), whereas a typical reference sequence of 18S rRNA is 1,700 bp long, which corresponds to only 0.003 % of the complete genome. As stated in Section 2.1 the increase of available microplankton genomes is steadily increasing, but slow due to difficulties in cloning.

Approach (a) is less feasible due to the database's incompleteness. PriSeT, therefore, supports approach (b) and searches first for short, conserved regions serving as primer binding sites. How the proposed primer pairs are ranked is up to the user.

### $k$ -mer

**Definition 3.1.** With  $k$ -mer we refer to a substring of length  $k \in \mathbb{N}$  as part of a biological text  $T$ . A text of length  $n$  contains  $n - k + 1$   $k$ -mers (see Figure 3.1 for illustration). For an arbitrary alphabet  $\Sigma$ , there exists  $|\Sigma|^k$  different  $k$ -mers, i.e. for DNA  $4^k$  unique  $k$ -mers. Since DNA is inherently structured, we do not expect  $k$ -mers to occur with equal probability.

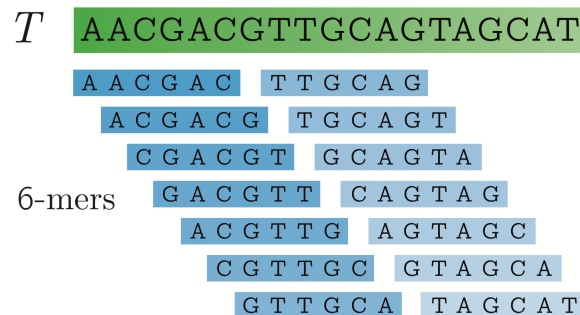


Figure 3.1: Example of 6-mers contained in a text  $T$ . The text (green) contains  $n - k + 1 = 19 - 6 + 1 = 14$   $k$ -mers (blue). The last  $k - 1$  symbols do not form full-length  $k$ -mers.

Formally, we can describe primer candidates as  $k$ -mers - DNA textitwords of length  $k$  - that satisfy a set of sequence conditions essential for the success of a PCR. Some of these constraints apply regardless of the second sequence of the primer pair. Examples are range limits for melting temperature ( $[\kappa_{\min} : \kappa_{\max}]$ ), GC-content, or low probabilities for self-annealing. We will abbreviate the set of constraints counting independently with  $C_s$  as done in Section 2.3. When combining two  $k$ -mers to produce a primer pair, both have to fulfill secondarily a set of constraints arising from their strand orientation and fitting like avoidance of GC clamps or low probability of cross-annealing, abbreviated with  $C_p$ .

Note that there are additional recommendations for PCR primers and transcripts like the spanning of exon or intron regions, which we do not consider here due to the lack of longer, annotated sequences for microplankton clades.



### 3.3 Starting with an MSA

A straightforward and naïve approach is to compute a multisequence alignment first as it tries best to align conserved regions (see Section 2.2.4). We can calculate the amount of conservation by measuring the entropy within a sliding window of variable length, which corresponds to the range of possible primer lengths. In case some entropy threshold is not exceeded within the block covered by the sliding window, and the sequence is chemically suitable as a primer sequence, we memorize its location and length.

In case multiple highly conserved blocks are detected, those blocks are combined, which satisfy a distance range<sup>3</sup>. If their sequences allow the formation of a proper primer pair, we output the primer pair as a candidate<sup>4</sup>.

---

**Algorithm 2** Searching primer pairs by exploiting the structure of an MSA. Blocks in the length range  $[\kappa_{\min} : \kappa_{\max}]$  with low entropy are verified to satisfy constraint set  $C_s$ . Pairs are then formed, and the enclosed regions are scored to assess marker suitability.  $\delta_{\text{ent}}$  is the entropy threshold,  $\delta_{\text{sim}}$  the sequence similarity threshold, and  $\theta$  a threshold for the fraction of same-labeled OTUs.

---

```

1: procedure PrimerSearch( $\mathcal{R}, C_s, C_p, \delta_{\text{ent}}, \delta_{\text{sim}}, \theta$ )
2:   MSA  $\leftarrow$  msa( $\mathcal{R}$ )
3:    $[\kappa_{\min}, \kappa_{\max}] \leftarrow C_s.\text{get\_kappa}()$ 
4:   Primers  $\leftarrow []$  ▷ accumulator for primer locations
5:   for all  $i \leftarrow [1 : |M| - \kappa_{\min}]$  do
6:     for all  $k \leftarrow [\kappa_{\min} : \kappa_{\max}]$  do
7:        $S \leftarrow \text{entropy}(\text{MSA}[:, i : i + k])$ 
8:       if  $S \leq \delta_{\text{ent}}$  then
9:         seq  $\leftarrow \text{common}(\text{MSA}[:, i : i + k])$  ▷ common sense sequence
10:        if primer_check(seq,  $C_s$ ) then
11:          Primers +=  $[(i, k)]$ 
12:   Pairs  $\leftarrow []$  ▷ accumulator for pairs
13:   for all  $(i_1, k_1) \leftarrow \text{Primers}$  do
14:     for all  $(i_2, k_2) \leftarrow \text{Primers}$  do
15:       if  $i_1 < i_2$  and  $i_2 - i_1 \in [\tau_{\min} : \tau_{\max}]$  then
16:         seqfwd  $\leftarrow \text{common}(\text{MSA}[:, i_1 : i_1 + k_1])$ 
17:         seqrev  $\leftarrow \text{common}(\text{MSA}[:, i_2 : i_2 + k_2])$ 
18:         if pair_check(seqfwd, seqrev,  $C_p$ ) then
19:           barcode  $\leftarrow \text{MSA}[:, i_1 + k_1 : i_2 - 1]$ 
20:           if BarcodeCheck(barcode,  $\mathcal{R}.\text{get\_labels}()$ ,  $\delta_{\text{sim}}, \theta$ ) then
21:             Pairs +=  $[(\text{seq}_{\text{fwd}}, \text{seq}_{\text{rev}})]$ 
22:   return Pairs

```

---

To check if the transcript allows sufficient species distinction, we can repeat the last step of a bioinformatics pipeline: sequence clustering based on a similarity threshold  $\delta_{\text{sim}}$ . A suitable barcode would result in clusters that unite transcripts from identical taxa. In practice, we would also get clusters with mixed labels, meaning that two or more taxa are indistinguishable. We would then use a soft approach by limiting the fraction  $\theta$  of mixed-labeled clusters with  $\theta \in (0, 1]$ .

The advantages of the MSA-based approach (Algorithm 2) are:

<sup>3</sup>a constraint that arises from the transcript length restriction of the PCR (see Section 2.3)

<sup>4</sup>Note, *primer pair* and *primer set* are synonymous.



**Algorithm 3** Barcode check by using agglomerative clustering on a set of transcripts  $T$  to form OTUs. A dictionary tracks the centroids' labels. The labels are the assigned taxa of the transcripts. The hierarchical clustering routine stops when there are no more centroids closer than the threshold parameter  $\delta_{\text{sim}}$ . A cluster can be represented by a *common sense* sequence and the distance between two clusters as the Hamming distance between their common sense sequences. This test returns true if a sufficiently large fraction (more than  $\theta$ ) of OTUs contain sequences of the same label.

---

```

1: procedure BarcodeCheck( $T$ , labels,  $\delta_{\text{sim}}$ ,  $\theta$ )
2:    $C \leftarrow [\tau_i]_{i \in [1:|T|]}$  ▷ centroids
3:    $L \leftarrow \{c_i : \{\text{label}_i\}\}_{i \in [1:|T|]}$  ▷ label dictionary
4:    $i, j \leftarrow \underset{c_i/j \in C, c_i \neq c_j}{\text{argmin}} (||c_i - c_j||)$ 
5:   while  $|C| > 1$  and  $||c_i - c_j|| \leq \delta_{\text{sim}}$  do
6:      $c_{\text{new}} \leftarrow \text{merge}(c_i, c_j)$ 
7:      $C \leftarrow C \setminus \{c_i, c_j\} \cup \{c_{\text{new}}\}$ 
8:      $L \leftarrow L \setminus \{c_i, c_j\} \cup \{c_{\text{new}} : L[c_i] \cup L[c_j]\}$ 
9:      $i, j \leftarrow \underset{c_i/j \in C, c_i \neq c_j}{\text{argmin}} (||c_i - c_j||)$ 
10:  purity  $\leftarrow |[1 | L[c_i].\text{size}() == 1]_{i \in [1:|C|]}|$ 
11:  if purity/ $|C| \leq \theta$  then
12:    return True
13:  return False

```

---

Procedure	PrimerSearch		
Subroutine	MSA	Low-Entropic Regions	Pair Formation
Runtime	$\mathcal{O}(n^2 l^2) + \mathcal{O}(n^3 l)$	$\mathcal{O}(nl)$	$\mathcal{O}(l^2(\kappa_{\text{max}} - \kappa_{\text{min}} + 1)\bar{k})$

Procedure	BarcodeCheck	
	per block	over all blocks
Runtime	$\mathcal{O}(n^3)$	$\mathcal{O}(l^2(\kappa_{\text{max}} - \kappa_{\text{min}} + 1)n^3)$

Table 3.1: Theoretical runtimes for an MSA-based primer search and barcode suitability check. Assuming  $n$  sequences and an upper limit  $l$  of the sequence length, e.g. T-Coffee for MSA computations consumes  $\mathcal{O}(n^2 l^2) + \mathcal{O}(n^3 l)$  (Notredame, Higgins, and Heringa, 2000). We represent each block through a single, common sense sequence. As we can check chemical fitness in linear time, we need  $\mathcal{O}(\bar{k})$  time for each block combination.  $\bar{k}$  denotes the block length mean. The barcode check for a single block of transcripts  $T \in \text{MSA}$  runs in  $\mathcal{O}(n^3)$  due to the need for repeated pair-wise distance computations when using a standard clustering approach as described in Algorithm 3.

- (i) MSA aligns most conserved columns in a stacked arrangement.
- (ii) The entropy calculation can be performed in one pass for all possible block lengths in  $[\kappa_{\min} : \kappa_{\max}]$  due to the linear independence of the columns.

The disadvantages are related to the nature of MSAs and a costly barcode check routine:

- (i) The MSA calculation is interrupted if sequences of low quality (ambiguous or wrong bases) are present or if the sequence length varies, which is the case for most database sequences.
- (ii) MSA is challenging to solve for thousands of sequences simultaneously.
- (iii) MSA is meaningless for phylogenetically distant sequences (see Section 2.2.4).

Phylogenetically distant sequences underwent DNA insertions and deletions over time, such that the relative distance between two conserved regions is altered. Hence, we need a much more robust strategy that primarily ignores relative distances of conserved regions, but still detects their similarity, and allows individual (i.e., reference-wise) testing of constrained range limits.

### 3.4 Limitations of Existing Primer Search Tools

There exist computer-assisted approaches, in which a manageable subset of references of organisms that are expected to show up, is collected and serves as input to compute a *multiple sequence alignment* (MSA). Then a variation or entropy score is calculated given the nucleotide distribution for each position of the alignment. Low entropy regions are then analyzed for serving as primer templates and regions with high inter-species entropy as barcodes (Hadziavdic et al., 2014). As we have seen in Section 2.2.4, MSA-based approaches are unsuitable for the planning of metabarcoding experiments. In this section, we look at the available primer search tools and discuss why they are not able to handle reference datasets for metabarcoding.

Most popular is the online tool Primer Blast/Primer3<sup>5</sup> (Ye et al., 2012). Users provide as input a GenBank accession or a FASTA file. NCBI's Primer-BLAST uses Primer3 to design primer sequences and performs a BLAST search against the specified sequence or user-supplied FASTA file. Surprisingly, Primer-BLAST *does not handle files with multiple references* and is therefore not applicable in a metabarcoding setup.

The Primer Search Tool<sup>6</sup> by Tusnády et al., 2005 uses a dynamic programming approach inspired by Kämpke, Kieninger, and Mecklenburg, 2001. When combining candidate sequences, Kämpke, Kieninger, and Mecklenburg, 2001 makes use of the sequence overlap and re-use computations of the shared substring. The Primer Search Tool is not able to discover new sequences. Instead, the user has to propose primer sequences that are searched in about two dozen species – plankton taxa are not covered. The tool outputs primer pairs scored by chemical fitness but does not optimize towards coverage or amplicon variation as only single genomes are parsed. About two dozen provided genomes can be selected for primer search. The Primer Search Tool is therefore not a primer discovery tool for user-defined input.

PRIMEX by Lexa and Valle, 2004 is intended to provide querying as an online service. The tool looks up  $k$ -mers derived from the query string in a prepared reference

<sup>5</sup>[www.ncbi.nlm.nih.gov/tools/primer-blast](http://www.ncbi.nlm.nih.gov/tools/primer-blast)

<sup>6</sup><http://bisearch.enzim.hu/?m=genomsearch>

Tool	Input	Remarks
FastPCR GLAPD	sequences genomes	does not identify frequent pairs LAMP (not PCR!) primer design for pathogen detection, not optimized for surveying
Primer Search Tool	primers	no input of multiple sequences possible, no primer discovery
Primer3 GenScript	one sequence one sequence	no support for multiple sequences no identification of frequent primers

Table 3.2: Summary of some available primer search tools w.r.t. their suitability for metabarcoding in the context of environmental monitoring where the reference data set is typically composed of thousands of short sequences.

dataset and allows for mismatches. Unfortunately, PRIMEX is currently not available, and it is not capable of discovering new primer sequences<sup>7</sup>.

One of the few standalone tools is FastPCR by Kalendar et al., 2017. It supports a variety of PCR protocols and chemical checks. FastPCR first computes a hash table with  $k$ -mers grouped by overlap into so-called  $k$ -tuples with  $k$  being 7, 9, or 12 bp long. The derivation of these  $k$ -tuples is not described. In a second step, FastPCR uses a sliding window to search for matches between  $k$ -tuples and reference sequences. Upon a match, individual  $k$ -tuples are extended in both directions. FastPCR has no preference for frequent  $k$ -mers – it is designed to work on single genomes. In addition, the Java Applet does not meet the Security standards and is blocked by default. After having added an exception and launched the applet with the clade Rotifera (taxonomic ID 10190) dataset (1.6 MB) the program did not terminate after 70 min (compared to 40 s with PriSeT, see Appendix B.2).

Most recently, GLAPD by Jia et al., 2019 has been published. The tool searches for primers for loop-mediated isothermal amplification (LAMP) – a type of PCR performed in a single tube with 4-6 different primer sets to amplify 6-8 different regions of the target gene (Notomi et al., 2000). LAMP is more robust and produces larger amounts of DNA compared to conventional PCR. The multitude of amplified regions per genome allows for higher precision. LAMP was primarily developed for the detection of pathogens in medical laboratories. As the amplified regions need to be in a close neighborhood, it is relatively difficult to identify optimal primer combinations. GLAPD aims to support the otherwise manual search and requires whole genomes as input to identify a combination that differentiates pathogenic from non-pathogenic groups with high certainty. LAMP lacks much of the versatility that a conventional PCR has, which is required for metabarcoding to be **survey-like**.

Most available primer search tools can only process a single sequence. We are not aware of any primer discovery tool that targets high taxonomic coverage and barcode variation.

### 3.5 Idea

We have seen in Section 2.2.4 that MSAs become arbitrary and thus meaningless for phylogenetically very distant sequences. On top of that, even the fastest, heuristic MSA computation algorithms cannot deal with tens of thousands of sequences. We

<sup>7</sup>attempt of accession on 24th February 2020 <http://bioinformatics.cribi.unipd.it/primex>

have to tackle the primer discovery problem in a very different way and think of conservation as short regions that occur repeatedly. Primer lengths are restricted to a relatively small range. A good starting point would be a data structure that allows us to collect variable-length words ( $k$ -mers) so that we memorize only those  $k$ -mers that cover a minimum set of reference sequences.

A well-known and highly optimized *library*<sup>8</sup> transformation that supports  $k$ -mer querying and frequency computation is the *FM-index* presented in detail in Section 3.6. An *index* is an auxiliary data structure built on top of a text corpus. We use the index to filter for frequent  $k$ -mers first as computed by the submodule (GenMap by Pockrandt et al., 2020). To obtain the actual sequences, we need to look them up in the original library. Storing each  $k$ -mer as a character sequence (1 byte per character) consumes unnecessarily much space and quickly limits the size of input libraries. For example, the smallest group evaluated later in this chapter (see Section 3.11) is Perkinsidae (taxonomic ID 27999) with a library of only 114 sequences (0.13 MB). PriSeT identifies about  $2^{18}$  frequent  $k$ -mers for  $k$  in  $[\kappa_{\min} : \kappa_{\max}] = [16 : 25]$ , of which  $2^{12}$  are chemically suitable primer sequences. Assuming a  $k$ -mer's average length of 20, we need to store and handle  $20 \cdot 2^{18}$  bytes = 5.2 MB of  $k$ -mer sequences. We will reduce the storage size by using a 2-bit compression scheme and storing up to 10  $k$ -mers in a single 64-bit built-in datatype (64ULL). Compression is possible because  $k$ -mers starting at the same position also share a common prefix (see Section 3.7). The 2-bit encoding scheme also allows for bit parallelizing of chemical tests that require parsing, accumulation, or pattern search. In short, PriSeT operates in four steps:

1. **FM-Index Step.** As a preprocessing, the FM-index is computed on the complete corpus of all references. Recomputation is only necessary when the references change.
2. **FM Frequency Step.** Given the FM-index, PriSeT computes the FM frequency for  $k$  in range  $[\kappa_{\min} : \kappa_{\max}]$  and reports all those  $k$ -mers exceeding a threshold  $Z$ .
3. **Filter Step.** For each  $k$ -mer, PriSeT checks the first set of constraints  $C_s$ , which must hold independently.
4. **Combine Step.** The remaining  $k$ -mers are combined and checked for pair-wise fitting of constraint set  $C_p$ .

The distinct steps are presented in detail in Section 3.9. The data structures that facilitate fast  $k$ -mer acquisition and sequence checks are presented in the following section.

---

<sup>8</sup>Note, that in this chapter, *library* refers to the set of reference sequences, typically obtained from an online database, as opposed to library preparation as the first step of NGS.

## 3.6 Indexing Data Structure

Index computation is the primary technique to accelerate queries over a large corpus that is rarely updated. The index presented here supports the quick assessment of all occurrences of a given  $k$ -mer. PriSeT uses this quantitative information to capture only those who exceed a user-defined frequency threshold.

From hereon we denote a text over an alphabet  $\Sigma$  with  $T$ . An arbitrary sequence of alphabet symbols, including the empty string, is denoted as  $\Sigma^*$ , and the set of all fixed-length ( $k$ ) words as  $\Sigma^k$ . The text  $T \in \Sigma^*$  supports the random access operator  $[\cdot] : \mathcal{N}^+ \mapsto \Sigma$  which maps a position index or range of indices to a slice in  $T$ . A word  $w \in \Sigma^*$  (or  $k$ -mer) has an occurrence at position  $i$  if  $T[i : i + |w|] == w$ .

Frequent  $k$ -mers are words whose particular lengths and sequences are unknown yet. However, we limit the lengths to a reasonable range, i.e.  $k \in [\kappa_{\min} : \kappa_{\max}]$ . Since we are interested in the occurrence of  $k$ -mers in all input sequences, we treat them as a single text corpus and concatenate the sequences:

$$T = R_1 \circ R_2 \circ \dots \circ R_m$$

To operate on the index, we need the provisioning of two functionalities:

1. *locate*( $T, k$ -mer) which returns the set of positions where the given  $k$ -mer occurs as a list of pairs (SeqID, SeqPos), where SeqID is the rank of the host sequence and SeqPos the relative position, s.t.  $R_{\text{SeqID}}[\text{SeqPos} : \text{SeqPos} + k]$  is equal to the searched  $k$ -mer.
2. *frequency*( $T, k$ ) which returns a list  $F$  denoting for each text position  $i$  the number of occurrences of substring  $T[i : i + k - 1]$  in  $T$ .

We expect to have millions of queries on the index, both operations should be therefore very time-efficient. We will use a more general definition allowing up to  $e$  errors for a  $k$ -mer:

### $(k, e)$ -Frequency

**Definition 3.2.** The  $(k, e)$ -frequency counts for each of the  $n - k + 1$   $k$ -mers in a text  $T$  of length  $n$ , their frequencies in the same text  $T$  with up to  $e$  errors.

For example, the  $(4, 0)$ - and  $(4, 1)$ -frequencies of  $T = \text{AACGACGTTGCAGTAGCAT}$  over  $\Sigma = \{\text{A,C,G,T}\}$  are:

$i$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$T[i]$	A	A	C	G	A	C	G	A	T	G	C	A	G	T	A	C	G	A	T
$F_{4,0}[i]$	1	3	1	1	3	2	1	1	1	1	1	1	1	1	3	2			
$F_{4,1}[i]$	3	3	3	3	3	3	2	1	1	1	1	1	1	3	3	3			

We will later use the frequency vector  $F_{k,e}$  to filter for those  $k$ -mers that exceed the occurrence threshold parameter  $Z$ . Note that with the inclusion of more errors, the number of  $k$ -mers reported increases dramatically (35 for  $F_{4,1}$  compared to 24 for  $F_{4,0}$ ), since  $k$ -mers are counted not only for identical matches but for all  $k$ -mers with a *Hamming distance*<sup>9</sup> of one. Before we derive an index representation that allows time- and space-efficient *locate* and *frequency* operations, there are two more important functionalities that are used for index computation and in PriSeT's combination step:

<sup>9</sup>The Hamming distance is the number of positions for which two sequences of equal length differ

*rank* and *select* support. The support structures answer questions like given a text index  $i$  how often has symbol  $\sigma$  occurred in the prefix  $T[:i]$  ( $\text{rank}_1$ ), or inversely what is the index position until which a symbol  $\sigma$  has occurred a given number of times ( $\text{select}_\sigma$ -support)?

If the underlying sequence remains unchanged, the rank and select support structures can be built to allow query answering in  $\mathcal{O}(1)$  time. Concretely, we employ  $\text{rank}_1$  and  $\text{select}_1$  support for querying efficiently  $k$ -mers located in sliding windows (see Section 3.9.4).

### rank Support

#### Definition 3.3.

Given a text index  $i$ , the *rank* of a symbol  $\sigma \in \Sigma$ , outputs the number of letters equal to  $\sigma$  until position  $i$ , including the text position  $i$  itself.

$$\text{rank}_\sigma(i) := |\{j \mid T[j] = \sigma, j \in [0 : i]\}|$$

For example, the  $\text{rank}_1$  support on  $T = 100101111010001$  over  $\Sigma = \{0, 1\}$  for all  $i \in [1 : |T|]$  is:

$i$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$T[i]$	1	0	0	1	0	1	1	1	1	0	1	0	0	0	1
$\text{rank}_1[i]$	1	1	1	2	2	3	4	5	6	6	7	7	7	7	8

### select Support

#### Definition 3.4.

Inverse to the  $\text{rank}_\sigma$  function,  $\text{select}_\sigma$  outputs the index of the  $k$ -th occurrence of symbol  $\sigma$  in the text.

$$\text{select}_\sigma(k) := \underset{i}{\operatorname{argmax}} \{i \mid \text{rank}_\sigma(i) = k, i \in [0 : n]\}$$

For example,  $\text{select}_1$  on  $T = 100101111010001$  over  $\Sigma = \{0, 1\}$  gives us:

$i$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$T[i]$	1	0	0	1	0	1	1	1	1	0	1	0	0	0	1
$k$	1		2		3	4	5	6		7		8			
$\text{select}_1[k]$	0		3		5	6	7	8		10		14			

## 3.6.1 FM-Index

Arguably the most important index data structure in bioinformatics applications is the FM index, a substring index over a text. The name stands for **F**ull-text index in **M**inute space (Ferragina and Manzini, 2005). It supports the needed *locate* and *frequency*<sup>10</sup> functionalities while representing the library in a compressed form.

Concretely, the FM-Index combines the Burrows-Wheeler compression algorithm (BW, see Burrows and Wheeler, 1994) and suffix arrays (SAs) to obtain a compressible representation.

<sup>10</sup>called *count* in the original paper

Nong, Zhang, and Chan, 2009 showed that the suffix array can be constructed in linear time. The same authors also presented one of the most influential implementations. Its practical runtime has been improved further by using *DivSufSort* as a sorting routine by Yuri Mori (not documented). Despite his approach being in a slower runtime class ( $\mathcal{O}(n \log n)$ ), it makes better usage of the main memory – a bottleneck during index construction – and yields faster runtimes in practice as analyzed by Fischer and Kurpicz, 2017.

We will now derive a basic form of the FM-index, starting with the well-known suffix array. The notation follows the one in “*Compact Data Structures*” by Navarro, 2016.

### Suffix Array $A$

Suffix arrays (SAs) have been described as an alternative to suffix trees independently by Manber and Myers, 1990 and Gonnet and Baeza-Yates, 1992 (PAT arrays). A suffix tree is constructed by inserting each of the  $|T|$  suffixes, i.e.  $T[i : n]_{i \in [1:n]}$  plus a terminal character (most often denoted as  $\$$  with  $\$ \notin \Sigma$ ) into a tree where edges correspond to substrings and leaves to a terminated suffix. Patterns are then searched, starting at the root and following edges that match the next symbol. Space consumption and tree structure overhead are unnecessarily large. Spatially more efficient is to use an index array in the length of the text denoting the suffix starts relative to the original text. The array gets sorted lexicographically to facilitate searches.

Table 3.3 shows an example of an SA constructed from the text  $T = \text{ACGTACGT}$ . The terminal symbol  $\$$  is defined as lexicographically smaller than any other symbol in  $\Sigma$ . Consequently, the last suffix is “ $\$$ ” and will always be the first one in the sorted list. The next smallest suffix is “ACGT $\$$ ” followed by “ACGTACGT $\$$ ” and so forth.  $A[i]$  gives us the suffix start index that has rank  $i$  in the sorted list of suffixes. We can also ask the other way around: what is the rank of the suffix  $T[i : n]$  in the sorted list? We denote the inverse of  $A$  that answers such questions with  $A^{-1}$ .

Searching for a word ( $k$ -mer) in  $T$  is equivalent to searching for all suffixes starting with that word. The lexicographical sorting of  $A$  implies that all occurrences of a word are coalescent in  $A$ . For example, the word ACGT is the prefix of  $T[1 : n]$  and  $T[5 : n]$  whose indices are found in direct juxtaposition: at  $A[2]$  and  $A[3]$ . If we are able to determine the left ( $l$ ) and right ( $r$ ) boundaries of a  $k$ -mer occurrence in  $A$ , we can answer *locate* and *frequency* queries very easily:

1.  $locate(T, kmer) := A[l : r]$
2.  $frequency(T, k) := [r_i - l_i + 1]_{i \in [1:|T|]}$  with  $l_i, r_i$  denoting the ranges for  $k$ -mer  $T[i : i + k - 1]$

### A Compressible Representation $\Psi$ of $A$

We will use a different ordering of the indices in  $A$ , such that we can iterate over the suffix ranks in the order of the original text, i.e.,

$$\Psi := A^{-1}[(A[i] \% n) + 1] \forall i \in [1 : n] \quad (3.1)$$



Index	1	2	3	4	5	6	7	8	9
$T$	A	C	G	T	A	C	G	T	\$
Sorted Suffixes	\$								
	A	C	G	T	\$				
	A	C	G	T	A	C	G	T	\$
	C	G	T	\$					
	C	G	T	A	C	G	T	\$	
	G	T	\$						
	G	T	A	C	G	T	\$		
	T	\$							
T	A	C	G	T	\$				
$A$	9	5	1	6	2	7	3	8	4
$T[A[i]]$	\$	A	A	C	C	G	G	T	T
$A^{-1}$	3	5	7	9	2	4	6	8	1
$\Psi$	3	4	5	6	7	8	9	1	2

Table 3.3: Example of a suffix array  $A$  over the text  $T = \text{ACGTACGT}$ . When sorting all suffixes, the index order is permuted and stored in  $A$ .  $A[i]$  gives the start index of the suffix, which has rank  $i$  in the list of sorted suffixes. Whereas its inverse  $A^{-1}$  gives the rank of suffix  $T[i : n]$ . The last row displays the re-ordering  $\Psi$  of  $A$ .

Table 3.3 shows exemplarily the re-ordering  $\Psi$  given the suffix array  $A$  over  $T = \text{ACGTACGT}$ . Applying  $\Psi$  iteratively on its input, i.e.,

$$\Psi^k[i] := \underbrace{\Psi \circ \dots \circ \Psi}_{k \text{ times}}[i]$$

where  $i$  is the query rank,  $A[\Psi^k[i]]$  actually returns the corresponding suffix position in  $T$  with offset  $k$ , i.e.,

$$A[i] = j \Leftrightarrow A[\Psi^k[i]] = j + k \quad (3.2)$$

Let us iterate, for example, over three consecutive text positions starting with  $T[5]$ :

$$\begin{aligned} T[A[2]] &= \text{A} \\ T[A[\underbrace{\Psi[2]}_{=4}]] &= \text{C} \\ T[A[\underbrace{\Psi^2[2]}_{=6}]] &= \text{G} \\ T[A[\underbrace{\Psi^3[2]}_{=8}]] &= \text{T} \end{aligned}$$

$\Psi$  is a permutation of  $A$  that allows us to parse the text and retrieve each suffix's rank. We use  $A$  only for querying the corresponding symbols in  $T$ . As you can see in Table 3.4 (row  $T[A]$ ) the starting symbols of each sorted suffix are clustered: first the  $\text{\$}$ -symbol, then a series of As, followed by Cs, and so forth. We denote with  $\Psi_\sigma$  the sub-range of  $\Psi$  with same starting symbol  $\sigma \in \{\text{\$}, \text{A}, \text{C}, \text{G}, \text{T}\}$ . It is sufficient to store locations of symbol changes either as an equally sized sparse bit vector, which



supports rank operations in  $\mathcal{O}(1)$  or simply the start indices of each symbol range (see array  $C$  in the last row of Table 3.3).

Index	1	2	3	4	5	6	7	8	9
$T$	A	C	G	T	A	C	G	T	\$
$A$	9	5	1	6	2	7	3	8	4
$T[A]$	\$	A	A	C	C	G	G	T	T
$\Psi$	3	4	5	6	7	8	9	1	2
$\sigma$	\$	$\Psi_A$	$\Psi_C$	$\Psi_G$	$\Psi_T$				
$C[\sigma]$	0	1	3	5	7				

i	1	2	3	4	5	6	7	8	9
$B_\$$	0	0	1	0	0	0	0	0	0
$B_A$	0	0	0	1	1	0	0	0	0
$B_C$	0	0	0	0	0	1	1	0	0
$B_G$	0	0	0	0	0	0	0	1	1
$B_T$	1	1	0	0	0	0	0	0	0
$L$	T	T	\$	A	A	C	C	G	G

Table 3.4: Derivation of FM-index helper structures. The range indices for suffix start symbols are stored in the array  $C$  (left). When sorting symbols relative to their  $\Psi$  values, we get a symbol representation of  $\Psi$  either as a set of bit-vectors  $B_\sigma$  or as a compact list  $L$  (right). The subranges of  $\Psi$  corresponding to identical text symbols are called  $\Psi_\sigma$ . Array  $C$  delimits the subarrays  $\Psi_\sigma$ .

Here, we will use array  $L$  – the compact list representation as depicted in Table 3.4 (instead of bit-vectors  $B_\sigma$ ) with  $\text{rank}_\sigma$ -support. When searching for words, it turns out that searching a word backward is in practice faster. Assume, we know the occurrence range for a symbol  $\sigma_{m+1}$ , say  $[l_{m+1}, r_{m+1}]$ , we do not need to carry out binary search for the preceding symbol  $\sigma_m$  in  $\Psi$ , but can restrict to values of  $\Psi_\sigma$  that are contained in the interval  $[l_{m+1}, r_{m+1}]$ . In other words, we are narrowing down the interval of candidate locations in each iteration. Now we have all the tools to describe a fast search procedure. We will make use of the  $\text{rank}_\sigma$ -support, which augments  $L$  instead of carrying out a binary search.

### Backward Search

Given the representation  $L$  of  $\Psi$ , the symbol boundaries  $C$ , and a concrete  $k$ -mer, Algorithm 4 demonstrates the narrowing of the candidates' range.

**Algorithm 4** Backward search of a  $k$ -mer on an FM-index. The search is narrowing down the left and right boundaries with each iteration. Here we use another representation for  $\Psi$ , i.e.,  $L$  with rank support for each symbol.

---

```

1: procedure BackwardSearch( $L, C, \text{kmer}$ )
2:    $m \leftarrow |\text{kmer}|$ 
3:    $[l, r] \leftarrow [C[\text{kmer}[m]] + 1, C[\text{kmer}[m] + 1]]$ 
4:   for all  $k \leftarrow [m - 1 : -1 : 1]$  do
5:     if  $l > r$  then
6:       return  $\emptyset$ 
7:      $\sigma \leftarrow \text{kmer}[m]$ 
8:      $l \leftarrow C[\sigma] + \text{rank}_\sigma(L, l - 1) + 1$ 
9:      $r \leftarrow C[\sigma] + \text{rank}_\sigma(L, e)$ 
10:  return  $[l, r]$ 

```

---

For example, searching for the 3-mer GTA in  $T$  gives us the initial range indices  $[l_3, r_3] = [2, 3]$  (line 3), then  $[l_2, r_2] = [9, 9]$ , and finally  $[l_1, r_1] = [7, 7]$ .

### Locate Computation

Equipped with a memory-efficient  $k$ -mer search algorithm, we can specify the location retrieval by collecting the start indices in the computed range as outlined in Algorithm 5.

---

**Algorithm 5** Location computation for a  $k$ -mer using BackwardSearch (see Algorithm 4).

---

```

1: procedure Locate( $L, C, kmer$ )
2:    $[l, r] \leftarrow$  BackwardSearch( $L, C, kmer$ )
3:   Locations  $\leftarrow$  []
4:   for all  $i \leftarrow [l : r]$  do
5:     Locations  $\leftarrow$  Locations + [ $A[i]$ ]
6:   return Locations

```

---

Instead of operating on the complete suffix array  $A$  in constant time, we can sample  $A$  and use  $\Psi$  to navigate to the next sampled position  $i' \geq i$  in case of a miss. From the equivalence relationship described in Equation (3.2) follows that the searched index can be computed by subtracting the number of steps it takes until we find a sampled  $A[i']$ . We choose a sampling factor  $\rho$  and sample equidistantly at positions of  $A$  that are a multiple of  $\rho$ . The indices are stored densely in  $A_S[1 : \lceil \frac{n}{\rho} \rceil]$  with presence indicated by a set bit in an additional bit array  $B_S$  of length  $n$  and size  $\log_2 n$ .

The sampling factor allows us to trade off space consumption versus retrieval time. Its adjustment should account for the computing architecture, text size, and function call frequency.

---

**Algorithm 6** Location computation for a  $k$ -mer with sampled suffix array  $A_S$ . Instead of accessing  $A$  directly, we compute  $A[i]$  via the offset distance to the next sampling point.

---

```

1: procedure LocateWithSampling( $\Psi, A_S, B_S, L, C, kmer$ )
2:    $[l, r] \leftarrow$  BackwardSearch( $L, C, kmer$ )
3:   Locations  $\leftarrow$  []
4:   for all  $i \leftarrow [l : r]$  do
5:     Locations  $\leftarrow$  Locations + [QuerySuffixArray( $\Psi, A_S, B_S, i$ )]
6:   return Locations
7: procedure QuerySuffixArray( $\Psi, A_S, B_S, i$ )
8:    $k \leftarrow 0$ 
9:   while  $B_S[i] == 0$  do
10:     $i \leftarrow \Psi[i]$ 
11:     $k \leftarrow k + 1$ 
12:   return  $A_S[\text{rank}(B_S, i)] - k$ 

```

---

### Frequency Computation

We generate  $k$ -mers by selecting substrings of length  $k$  at each text position from start to end. For each distinct  $k$ -mer, we then call the backward search to retrieve its occurrence range. Reducing the search to distinct  $k$ -mers can be achieved without memorization. Whenever a first co-occurring  $k$ -mer is located before the

currently processed one, we abort. The reason for this is that for the preceding copy we have already collected all simultaneous occurrences including the current one. Alternatively, we can memorize already seen  $k$ -mers as done below.

---

**Algorithm 7** Frequency computation for all  $k$ -mers using BackwardSearch (see Algorithm 4). As a simple optimization, we memorize already processed  $k$ -mers. The frequency computation deployed by PriSeT uses additional optimizations as described in 3.9.2.

---

```

1: procedure Frequency( $T, L, C, k$ )
2:    $F \leftarrow []$ 
3:    $mem \leftarrow \{\}$ 
4:   for all  $i \leftarrow [1 : |T| - k + 1]$  do
5:      $kmer \leftarrow T[i : i + k - 1]$ 
6:     if  $kmer \notin mem$  then
7:        $[l, r] \leftarrow \text{BackwardSearch}(L, C, kmer)$ 
8:        $F[i] \leftarrow r - l + 1$ 
9:        $mem.insert(kmer)$ 
10:  return  $F$ 

```

---

### 3.7 Space-Efficient $K$ -Mer Representation

In some applications, DNA sequences contain ambiguous nucleotides (see encoding in Appendix B.1). The ambiguity may stem from a base call uncertainty of the sequencer machine, or serves as a shortcut notation for a set of sequences. The latter is common for the notation of primer sequences, for example, the primer CGCGGTAATTCAGCTYC (SSU556F by Smith et al., 2017) stands for two sequences, namely CGCGGTAATTCAGCTCC and CGCGGTAATTCAGCTTC. Both variants would be added as reagents to a PCR. However, when searching for primer sequences in a library, substrings with ambiguous nucleotides will be ignored by PriSeT to have full control of the error or mismatch rate between primer and template. Primer-template mismatches are critical and may lead to an amplification failure.

When computing the  $k$ -mer frequency with varying values for  $k$ , one notices that for a specific position  $i$  in a reference  $R$ , it is likely that we will yield many  $k$ -mers of different lengths which will successfully pass the frequency filter and start at the same index. In fact there are up to  $\kappa_{max} - \kappa_{min} + 1$  many  $k$ -mers per sequence and position.

We introduce the TKMerID data type which is an alias for an unsigned 64 bit integer (uint64\_t or 64ULL), which primarily encodes the longest  $k$ -mer found at a specific text position (see Figure 3.2) via the two-bit encoding scheme shown in Table 3.5. The two-bit encoding scheme has the advantage that taking the complement of the integer produces its DNA complement. As shown later, we will exploit this property, combined with bit parallelism, to implement a linear-time annealing test.

Given the ubiquitous word length of 64 bits, we have to trade-off between the maximum sequence length we want to capture and its range. Given that optimal primer sequence lengths are in  $[18 : 22]$ , we can encode an even larger range, which is here  $[16 : 26]$ .

The DNA sequence encoding consumes two bits per nucleotide. A word size of 64 bits allows us to encode  $k$ -mers with up to  $\kappa_{max} = 26$  nucleotides. The part reserved

Nucleotide	Encoding	2-Bit Complement	Nucleotide Complement
A	↦ 00	11	T
C	↦ 01	10	G
G	↦ 10	01	C
T	↦ 11	00	A

Table 3.5: Two-bit encoding of single nucleotides. Note that the bit-wise complement operator produces the complement of the nucleotide likewise.

for the sequence consumes  $2 \times 26$  bits stored in little-endian ordering. We use the term *suffix* in this context to denote the DNA code part.

To unambiguously encode the beginning of the DNA sequence, we use one extra bit (= *closure bit*). Without the closure bit, preceding As ( $00_x$ ) would be ignored (or over-counted) otherwise. The various lengths represented by a single TKMerID are stored in the preceding 11 bits (= prefix). The  $j$ -th highest bit encodes length  $k = \kappa_{min} + j - 1$  (see Figure 3.2). We use masking to 0 of the fixed-length prefix to perform chemical property checks in a bit-parallel manner. If a chemical constraint is not satisfied for some length  $k'$ , we only delete the corresponding prefix bit. Truncation of the actual sequence located in the suffix is not necessary. The idea is to continue processing a TKMerID as a container with primer candidates as long as the prefix is not set to zero.

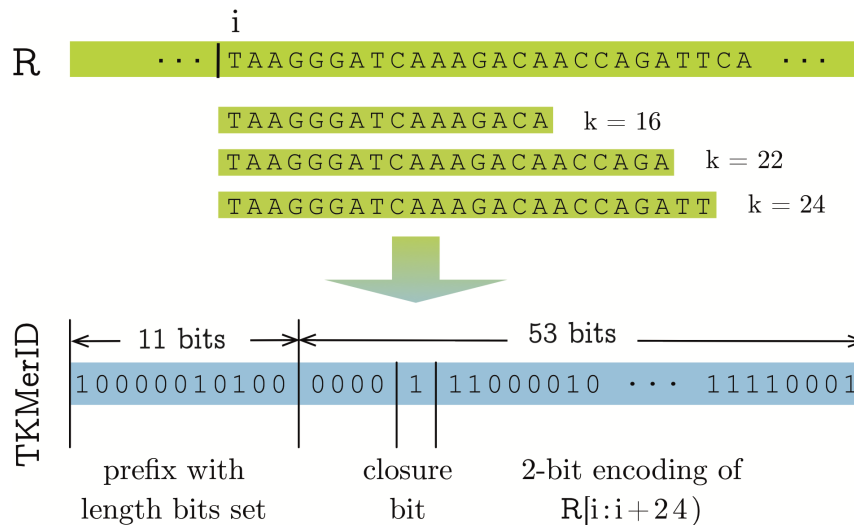


Figure 3.2:  $K$ -mer compression scheme. Top: All  $k$ -mers with the same starting index are prefixes of longer  $k$ -mers. Redundancy is reduced by encoding the longest  $k$ -mer only. Bottom: In this example, we encode the oligomer TAAGGGATCAAAGACAACCAGA with length bits set for 16, 18, and 22 in the remaining prefix bits. Instead of handling three sequences, we use only one integer that fits into a single 64-bit register.

## Macros

We can access prefix and suffix properties in constant time with the help of library

functions available on all C++ compilers. In combination with masking the prefix or suffix reserved parts in an `uint64_t`, we will namely use:

- `ffsll` for finding the first set bit and thereby the maximal encoded *k*-mer
- `__builtin_clzll` for counting leading zeros and identifying the sequence beginning
- `__builtin_popcountll` for counting the number set bits in order to facilitate bit-parallelism for sequence checks<sup>11</sup>

We will need functionalities like splitting a `TKMerID` into its prefix and code part, determining the encoded length, or resetting a specific length bit. The least error-prone and computationally most efficient way in C++ is to define these operations once as macros (see Listing 3.1). Macros are preprocessor directives that are named code fragments. The compiler replaces the macro with the content wherever it is used in the code.

For example, to identify the longest encoded length in a `TKMerID`, we define a function-like macro that drops the suffix and identifies the first set bit counted from the left end (see `encoded_length` in Listing 3.1). A length bit is reset by XORing a `TKMerID` with a single bit shifted to the corresponding position in the prefix (see `reset_length` in Listing 3.1).

Listing 3.1: Macro definitions.

```

1 #define WORD_SIZE 64
2 #define PREFIX_SIZE 11
3 #define PREFIX_SELECTOR ~(1 << (WORD_SIZE - PREFIX_SIZE)) - 1)
4
5 #define encoded_length(kmerID) WORD_SIZE - 1 -
6     __builtin_clzll(kmerID & ~PREFIX_SELECTOR)
7
8 #define reset_length(kmerID, l) kmerID ^
9     (1ULL << (WORD_SIZE - 1 - (l >> 1) + KAPPA_MIN))

```

### 3.8 Bit Vectors for *K*-Mer Location Encoding

After the FM-index computation (step 1), the original reference sequences are only needed once to lookup and encode frequent *k*-mers (step 2). However, when combining *k*-mer candidates to form pairs (step 4), we need location information – because *k*-mers need to refer to the same reference and have to be in an offset range of  $\tau_{min}$  to  $\tau_{max}$  as the transcript lengths are constrained. A naïve approach is to parse for each `TKMerID` occurrence a window of length  $\tau_{max} - \tau_{min}$  and test at each window position if there exists another `TKMerID` it can be combined with. If, on the other hand, we would know the exact locations of `TKMerIDs` in the search window, we could reduce the number of queries from  $\tau_{max} - \tau_{min} + 1$  to only  $|\{j \in [i + \kappa_{min} + \tau_{min} : i + \kappa_{max} + \tau_{max}] \wedge R_j = 1\}|$  – the actual number of `TKMerIDs` in the search window. This strategy takes full effect when there are only a few candidates in the search window.

To accomplish a lower number of queries, we use two data structures to represent the original reference sequence: a list to store `TKMerIDs` in order of occurrence and a *compact* data structure, namely a bit vector *B* in the length of the last *k*-mer

<sup>11</sup>e.g., counting the number of C or G bases

occurrence. A set bit indicates the presence of a TKMerID and is also the rank of the TKMerID in the associated list.

The *compact* data structure is augmented with  $\text{rank}_1$  and  $\text{select}_1$  support data structures, which both permit querying in constant time. Concretely, we use the `sdsl::bit_vector` by Gog et al., 2014 combined with `sdsl::rank_support_v5` and `sdsl::select_support_mcl`. Both support data structures occupy at most  $0.0625n$ , and  $0.2n$  extra bits, respectively (Clark, 1998, Vigna, 2008).

When iterating over TKMerIDs,  $\text{select}_1$  gives us their positions relative to the reference sequence. The search window of combinable reverse primers is then accessed by adding the transcript length range. Again we apply  $\text{rank}_1$  on the window indices to address the associated TKMerIDs in the list (see Algorithm 11). Pairs satisfying the constraint set  $C_p$  are collected and can be ranked by coverage or compactness. Figure 3.3 illustrates the later described  $k$ -mer combination step (see Algorithm 11) over the transformed reference.

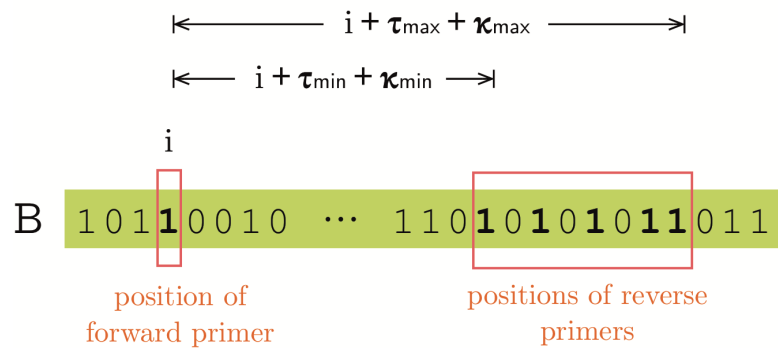


Figure 3.3: Encoding of TKMerID locations. In this example the bit-vector represents a reference sequences with  $k$ -mers starting at the indices 0, 2, 3, 6, and so forth. When searching for possible  $k$ -mer combinations to produce forward and reverse primer pairs, we have to parse for each forward TKMerID at a position  $i$  the window  $[i + \kappa_{\min} + \tau_{\min} : i + \kappa_{\max} + \tau_{\max}]$ . We directly access set bits in the window utilizing  $\text{select}$  support on bit-vector  $B$ , and  $\text{rank}$  support to address the associated TKMerIDs stored in an extra container.

## 3.9 Algorithm

We have seen in Section 3.6.1 how an FM-index over the set of references  $\mathcal{R}$  achieves fast runtimes for *locate* and *frequency* computations, in Section 3.7 how we use shared prefixes of  $k$ -mers starting at same text positions to store up to 11  $k$ -mers in a single 64-bit integer, and in Section 3.8 how  $k$ -mer candidates for primer pair formation can be addressed using  $\text{rank}_1$  and  $\text{select}_1$  support. We now have sufficient tools to describe the four steps of the primer discovery algorithm (see Section 3.5) in more detail. Exemplary, we demonstrate how sequence property checks (fulfillment of  $C_s$ ) is carried out on the TKMERID datatype.

### 3.9.1 FM-Index Step

For indexing and  $k$ -mer report PriSeT makes use of the recently published GenMap v1.0.1 by Pockrandt et al., 2020. As library input GenMap accepts a FASTA file or directory of FASTA files. Given the library length in number of symbols ( $N$ ), the index is computed in  $O(N)$  using the Skew7 algorithm by Weese, 2006 based on the known Skew algorithm (Kärkkäinen, Sanders, and Burkhardt, 2006). For index storage and bidirectional search GenMap uses a slightly different approach than presented in Section 3.6.1, concretely, enhanced prefix sum rank dictionaries (EPR) occupying  $\mathcal{O}(\log \sigma n) + o(\log \sigma^2 n)$  bits space (Pockrandt, Ehrhardt, and Reinert, 2017). Algorithm 8 illustrates the interface call. The procedure also calculates the absolute frequency threshold  $Z$  for  $k$ -mers given a user-defined percentual cutoff  $\zeta$ .  $Z$  will be applied later in step 2.

---

**Algorithm 8** FM-index transformation of the library (step 1). Based on a user-defined percentual frequency cutoff  $\zeta$ , an absolute cutoff ( $Z$ ) for  $k$ -mer occurrences is computed.  $Z$  will be applied when locating  $k$ -mers in the index (see Algorithm 9).

---

```

1: procedure FMIndexing(Library,  $\zeta$ )
2:    $Z \leftarrow \zeta \cdot |\text{Library}|$ 
3:   FMIndex  $\leftarrow$  genmap.index(Library)
4:   return FMIndex,  $Z$ 

```

---

### 3.9.2 FM Frequency Step

Having computed the FM-index over the library, *frequency* queries are submitted with values of  $k$  in the range  $[\kappa_{\min} : \kappa_{\max}]$ . GenMap's frequency computation of  $k$ -mers is based on an algorithm described by Derrien et al., 2012. However, it introduces runtime improvements by cutting redundant searches (Pockrandt et al., 2020) related to the optimization in Algorithm 4, usage of optimal search schemes, and skipping of mismatching positions. We modified GenMap's original frequency functionality to accept a frequency threshold parameter  $Z$  to avoid temporary storage of low frequent  $k$ -mers (see GenMap fork<sup>12</sup>). Algorithm 8 describes the call to GenMap's frequency interface.

Location information of each  $k$ -mer is provided by returning the sequence identifier SeqID referring to the library sequence where a  $k$ -mer has been found and a relative position index SeqPos. These locations are collected in a Locations map (see Algorithm 9) with  $k$ -mers as keys and an occurrence vector as value.

<sup>12</sup><https://github.com/mariehoffmann/genmap>



---

**Algorithm 9**  $K$ -mer frequency computation given the FM-index as lists  $C$ ,  $L$  (see Section 3.6.1), and a frequency cutoff (step 2).

---

```

1: procedure FMFreq(FMIndex,  $Z$ )
2:   Locations  $\leftarrow$  []
3:   for all  $k \leftarrow [16:25]$  do
4:     Locations  $\leftarrow$  Locations + genmap.freq(FMIndex,  $k$ ,  $Z$ )
5:   return Locations

```

---

### 3.9.3 $K$ -Mer Transform and Filter Step

Each  $k$ -mer has to pass a chemical filter that checks molecular property constraints as listed in Table 2.3 (step 3), which have to hold for single primers independent of the second one. Applying the filter directly after the frequency step reduces the number of  $k$ -mers by orders of magnitudes before running the combination procedure. The constraints ( $C_s$ ) that are currently checked by PriSeT are melting temperature, GC-content, mono- or dinucleotide runs, and self-annealing patterns (see Table 2.3). The other set of constraints ( $C_p$ ) is postponed to the last step (see Section 3.9.4).

Before frequent  $k$ -mers undergo chemical filtering, we encode their DNA sequences into the space-efficient data structure presented in Section 3.7 which exploits the length restriction of PCR primers. Per sequence position, we reduce space occupancy from  $\sum_{k=\kappa_{\min}}^{\kappa_{\max}} k = \mathcal{O}(\kappa_{\max}^2)$  bytes with the naive approach to  $4 = \mathcal{O}(1)$  bytes per location.

The chemical filters  $C_s$  are deployed on all  $k$ -mers encoded in a single TKMERID in parallel, and each single filter processes a TKMERID in  $\mathcal{O}(k)$ . For one category of filters, we will use optimization by stopping the processing of subsequent  $k$ -mer positions and immediately deleting length encoding bits. Assume an encoded  $k$ -mer fails a chemical test; then the following checks imply that all extensions will also fail to pass the test:

1. Excess of melting temperature  $T_m$
2. Runs and dinucleotide repeats
3. Self-annealing patterns
4. TATA-box

A test for the GC-content cannot be aborted, since an undercutting or exceeding GC ratio can be compensated for after the subsequent bases have been taken into account.

#### Filter for GC-content

The relative CG-content of a  $k$ -mer is determined by the number of cytosine and guanine bases in proportion to its length  $k$ . In the 2-bit encoding format we need to determine the number of 01 and 10 patterns subtracted by the closure bit (01). We can filter cytosine bases by computing the bit-wise AND of the prefix-eliminated code and the pattern  $(01)_{26}$ , and guanine bases with  $(01)_{26}$ . When looking at the example in Table 3.6 we see that selecting odd and even set bits is insufficient, since thymine (encoded as 11) occurs in  $p$  and  $q$ . To eliminate both bits introduced by T, we simply shift  $p$  left by one position and take the bit-wise XOR on  $p \ll 1$  and  $q$ . The C++-code is shown in Listing 3.2. As leading zero count (`__builtin_clzll`)



and `popcount(__builtin_popcountll)` also operation in  $\mathcal{O}(1)$ , the total runtime to determine the CG-content of a single  $k$ -mer is in  $\mathcal{O}(1)$ .

sequence with closure bit	0	1	A	C	G	T
code	0	1	0	0	1	1
$p = \text{code} \& (01)_4$	0	0	0	0	1	0
$p = p \ll 1$	1	0	0	0	0	0
$q = \text{code} \& (10)_4$	0	0	0	0	0	1
$x = p \text{ XOR } q$	0	0	0	0	1	0
<code>popcount(x)</code>	2					

Table 3.6: Bit-parallelized counting of cytosine and guanine bases on the sequence ACGT.

Listing 3.2: GC-content computation in  $\mathcal{O}(1)$ .

```

1 float GC(TKmerID const kmerID, uint64_t const mask)
2 {
3     auto [prefix, code] = split_kmerID(kmerID);
4     uint8_t enc_l = encoded_length(kmerID);
5     uint8_t target_l = KAPPA_MIN;
6     target_l += (prefix & !mask) ? __builtin_clzll(prefix) :
7         __builtin_clzll(mask);
8     code >>= (enc_l - (target_l << 1));
9     uint64_t p = 0x55555555555555ULL;
10    uint64_t q = 0xAAAAAAAAAAAAULL;
11    uint64_t x = ((code & p) << 1) ^ (code & q);
12    return __builtin_popcountll(x) - 1;
13 }

```

The relative amount of cytosine and guanine is then retrieved by dividing the absolute count by the sequence length (see Listing 3.3).

Listing 3.3: GC-filter based on the GC-content computation in Listing 3.2.

```

1 float GC_percent(TKmerID const kmerID, uint64_t const mask)
2 {
3     uint8_t target_l = KAPPA_MIN;
4     auto [prefix, code] = split_kmerID(kmerID);
5     target_l += (prefix & !mask) ? __builtin_clzll(prefix) :
6         __builtin_clzll(mask);
7     return float(GC(kmerID, mask))/float(target_l);
8 }

```

In order to filter a TKMERID encoding up to  $(\kappa_{\max} - \kappa_{\min} + 1)$   $k$ -mers, we simply iterate over set bits in the prefixing length mask as shown in Listing 3.4.

Listing 3.4: GC-filter based on the GC-content computation in Listing 3.2.

```

1 void GC_filter(TKmerID & kmerID, float const GC_min,
2     float const GC_max, uint8_t const kappa_min,
3     uint8_t const kappa_max)
4 {
5     uint64_t mask = 1ULL << 63;
6     float GC_content;
7     for (uint8_t k = kappa_min; k <= kappa_max; ++k)
8     {
9         if (mask & kmerID)

```

```

10     {
11         GC_content = GC_percent(kmerID, mask);
12         if (GC_content < GC_min || GC_content > GC_max)
13             kmerID ^= mask;
14     }
15     mask >>= 1;
16 }
17 }

```

### Filter for Melting Temperature $T_m$

A simple and approximate method for calculating the melting temperature is the linear combination of AT- and CG-counts according to Wallace's rule (Wallace et al., 1979):

$$T_m := 2AT + 4GC$$

We employ the GC-count function (Listing 3.2) to determine the number of guanine and cytosine bases. The difference between currently processed  $k$ -mer length and GC content will give us the number of adenine and thymine bases. If for a given  $k$ -mer length, say  $k'$ , the melting temperature falls below  $T_{m_{\min}}$ , the corresponding length bit in the TKMerID is reset. In case  $T_{m_{\max}}$  is exceeded, PriSeT resets all length bits corresponding to lengths larger or equal to  $k'$  because  $T_m$  grows monotonously with the length of the sequence. Hence,  $T_m$  will remain in an exceeding range.

Listing 3.5: Melting temperature computation in  $\mathcal{O}(1)$  according to the Wallace rule.

```

1 uint8_t Tm(TKmerID const kmerID, uint64_t const mask)
2 {
3     uint8_t target_l = KAPPA_MIN;
4     target_l += (prefix & !mask) ? __builtin_clzll(prefix) :
5         __builtin_clzll(mask);
6     uint8_t GC_content = GC(kmerID, mask);
7     return ((target_l - GC) << 1) + (GC << 2);
8 }

```

Listing 3.6:  $T_m$  filter based on the  $T_m$  computation in Listing 3.6. Note the short-cut when  $T_m$  exceeds  $T_{m_{\max}}$  in lines 13-17. `reset_length_leq` is a macro.

```

1 void Tm_filter(TKmerID & kmerID, uint8_t const Tm_min,
2     uint8_t const Tm_max, uint8_t const kappa_min, uint8_t const kappa_max)
3 {
4     uint64_t mask = 1ULL << 63;
5     uint8_t Tm_wallace;
6     for (uint8_t k = kappa_min; k <= kappa_max; ++k)
7     {
8         if (mask & kmerID)
9         {
10             Tm_wallace = Tm(kmerID, mask);
11             if (Tm_wallace < Tm_min)
12                 kmerID ^= mask;
13             else if (Tm_wallace > Tm_max)
14             {
15                 reset_length_leq(kmerID, encoded_length_mask(mask));
16                 return;
17             }
18         }
19         mask >>= 1;
20     }
21 }

```

### Filter for Runs and Dinucleotide Repeats

PriSeT discovers patterns like mono- or dinucleotide runs by comparing tailing bits of the code with the two-bit encoded patterns. These prohibited patterns can be infixes of a  $k$ -mer at any position. We, therefore, apply a right shift and comparison iteratively until 10 bits remain. Similarly, we proceed for dinucleotide runs (see Listing 3.7).

Listing 3.7: Filter for runs and dinucleotide repeats.

```

1 void filter_repeats_runs(TKmerID & kmerID)
2 {
3     auto [prefix, code] = split_kmerID(kmerID);
4     uint64_t const tail_selector_10 = (1 << 10) - 1;
5     uint64_t const tail_selector_20 = (1 << 20) - 1;
6     kmerID = code; // save trimmed kmer-code part
7     uint64_t const k = (63 - __builtin_clz(code)) >> 1;
8     uint64_t offset;
9     for (uint64_t i = 0; i < k - 4; ++i)
10    {
11        uint64_t tail_10 = tail_selector_10 & code;
12        if (    (tail_10 == 0b00000000) ||
13            (tail_10 == 0b01010101) ||
14            (tail_10 == 0b10101010) ||
15            (tail_10 == 0b11111111))
16        {
17            offset = 64 - (max((uint64_t)KAPPA_MIN, k - i) - KAPPA_MIN);
18            prefix = (offset == 64) ? 0 : (prefix >> offset) << offset;
19        }
20        if (k - i > 9)
21        {
22            uint64_t tail_20 = code & tail_selector_20;
23            if (    (tail_20 == 0b00110011001100110011) ||
24                (tail_20 == 0b11001100110011001100) ||
25                (tail_20 == 0b00010001000100010001) ||
26                (tail_20 == 0b01000100010001000100) ||
27                (tail_20 == 0b00100010001000100010) ||
28                (tail_20 == 0b10001000100010001000) ||
29                (tail_20 == 0b01100110011001100110) ||
30                (tail_20 == 0b10011001100110011001) ||
31                (tail_20 == 0b01110111011101110111) ||
32                (tail_20 == 0b11011101110111011101) ||
33                (tail_20 == 0b10111011101110111011) ||
34                (tail_20 == 0b11101110111011101110))
35            {
36                offset = 64 - (max((uint64_t)KAPPA_MIN, k - i) -
37                    KAPPA_MIN);
38                prefix = (offset == 64) ? 0 : (prefix >> offset) << offset;
39            }
40        }
41        if (!prefix)
42            break;
43        code >>= 2;
44    }
45    kmerID |= prefix;
46 }

```

### Filter for Self-Annealing

Self-annealing, as described in Section 2.3.2, may occur if there exists an alignment of a DNA oligomer against a same-sequence copy, which may lead to the formation of a stable dimer. We consider an annealing pattern as critical if more than four

bases in a row anneal (*connected* annealing pattern), or more than 50 % of the bases participate in bonds (*disconnected* annealing pattern). PriSeT determines the presence of annealing patterns by checking the sequence's shiftings against itself (-/-) and its reversed copy (-/+). Cross-annealing is handled analogously, except that PriSeT has to test more shiftings due to lack of symmetry. Given the rule of thumb, there are at most  $2\kappa_{\max} - 2 \cdot 8$  possible alignments, and for each alignment, we apply the bit-wise XOR-operator to translate complementary nucleotides into 0b11-blocks (see Table 3.7 and Table 3.8).

p	...	T	<b>A</b>	<b>C</b>	<b>G</b>	<b>T</b>	C	...
		11	00	01	10	11	01	
		01	11	10	01	00	01	
q	...	C	<b>T</b>	<b>G</b>	<b>C</b>	<b>A</b>	C	...
p XOR q		01	<b>11</b>	<b>11</b>	<b>11</b>	<b>11</b>	00	

Table 3.7: Closed or connected annealing patterns produce blocks of set bits for some alignment position when XORing the sequence against itself (see self-annealing in Figure 2.8) or against another (cross-annealing). A block of eight set bits or more corresponds to a critical pattern.

Connected self-annealing patterns correspond to eight consecutive set bits starting at an even position (see Table 3.7), and disconnected patterns to occurrence counts of 0b11-blocks exceeding  $k/2$ . We can count 0b11-blocks at even positions in parallel with a combination of masking, right-shift and subtraction. The goal is to transform 0b11-blocks into single bits and to eliminate 01- and 10-blocks which are the result from XORing T with C or G. An example is shown in Table 3.8 and a code excerpt in Listing 3.8. The check has to be done with same and opposite strand orientations, i.e. we also have to reverse one sequence and process it the same way.

p	...	G	<b>G</b>	<b>A</b>	<b>T</b>	C	<b>G</b>	...
		10	10	00	11	01	10	
		00	01	11	00	00	01	
q	...	A	<b>C</b>	<b>T</b>	<b>A</b>	A	<b>C</b>	...
x = p XOR q		10	<b>11</b>	<b>11</b>	<b>11</b>	01	<b>11</b>	
x' = (x » 1) & (01) <sub>26</sub>		01	01	01	01	00	01	
x -= x'		01	10	10	10	01	10	
x &= (10) <sub>26</sub>		00	<b>10</b>	<b>10</b>	<b>10</b>	00	<b>10</b>	
popcount(x)							4	

Table 3.8: Bit-parallelized counting of self- or cross-annealing base-pairs in two fragments.

Listing 3.8: Code fragment for bit-parallelized counting of annealing bases. Note that we need to mask out bits outside of the overlap region.

```

1 x = (code1 ^ code2) & overlap_mask;
2 x = x - ((x >> 1) & 0x15555555555555);
3 x &= 0xAAAAAAAAAAAA & overlap_mask;
4 auto annealings_x = __builtin_popcountll(x);

```

### Encoding and Filter Procedure

Algorithm 10 puts together the encodings of  $k$ -mers and their locations. The procedure builds for each reference sequence a list of TKMERIDs in order of occurrence and a bit vector  $B$  in the length of the last TKMERID occurrence (see Section 3.8). We first have to collect positional indices of  $k$ -mers (set bits in  $B$ ), and lengths occurring at a specific position (encoded in a prefix as described in Section 3.7). Finally, the sequence of the longest  $k$ -mer per location is looked up, 2-bit encoded, and the consolidated TKMERID checked for primer acceptability (see  $C_s$  in Table 2.3). In Section 3.9.4, it is explained how the location encoding structure  $B$  is used to address TKMERIDs efficiently.

---

**Algorithm 10** Lookup and encoding of DNA  $k$ -mers and references (step 3). First, the last  $k$ -mer occurrence for each reference sequence is determined, and a new bit vector instance is resized accordingly. Each location  $L$  from the input map is associated with a fixed value for  $k$  and a list of occurrences represented as tuples of sequence identifiers and positions. We refer to the number of bit vector transformed references as  $n$ . In the first loop (lines 4-6), PriSeT sets bits for each  $k$ -mer occurrence. The second loop (lines 8-12) composes the prefix by accumulating the length bits. In the third loop (lines 13-24), PriSeT looks up sequences for the longest  $k$  occurring at a specific position. If at least one  $k$ -mer encoded in a KMERID passes the filtering, the KMERID is appended to the list associated with the bit vector  $B$ . Otherwise, the related bit in  $B$  has to be reset.

---

```

1: procedure Filter(Locations, Text)
2:    $B \leftarrow [\vec{0}]_n$  ▷ initialize bit vectors
3:   KMerIDs  $\leftarrow [ [] ]_n$ 
4:   for all  $L \leftarrow$  Locations do
5:     for all SeqID, SeqPos  $\leftarrow$  L.occurences() do
6:        $B[\text{SeqID}][\text{SeqPos}] \leftarrow 1$  ▷ set bit for  $k$ -mer occurrence
7:   Loc2k  $\leftarrow \{ \}$  ▷ dictionary to collect values of  $k$  per position
8:   for all  $L \leftarrow$  Locations do
9:      $k \leftarrow$  get_k( $L$ )
10:    for all SeqID, SeqPos  $\leftarrow$  L.occurences() do
11:      prefix  $\leftarrow$  Loc2k[(SeqID, SeqPos)]|(1  $\ll$  (63 -  $k$  +  $\kappa_{min}$ ))
12:      Loc2k[(SeqID, SeqPos)]  $\leftarrow$  prefix ▷ update length bits
13:    for all SeqID  $\leftarrow$  [1 : | $B$ |] do ▷ lookup when bit set
14:      for all  $r \leftarrow$  [1 : rank1( $B$ , | $B$ [SeqID]|)] do
15:        SeqPos  $\leftarrow$  select1( $B$ ,  $r$ )
16:        prefix  $\leftarrow$  Loc2k[(SeqID, SeqPos)]
17:         $k_{max} \leftarrow \kappa_{max} - \text{ffs}(\text{prefix} \gg 54) + 1$  ▷ identify largest  $k$ 
18:        dna  $\leftarrow$  lookup(Text, SeqID, SeqPos,  $k_{max}$ )
19:        code  $\leftarrow$  encode(dna) ▷ encode DNA seq
20:        KMerID  $\leftarrow$  prefix | code ▷ concatenate prefix and code
21:        if  $C_s$ (KMerID) then
22:          KMerIDs  $\leftarrow$  KMerIDs + [KMerID] ▷ store
23:        else
24:           $B[\text{SeqPos}] \leftarrow 0$  ▷ drop KMerID and reset bit if not passing
25: return  $B$ , KMerIDs

```

---

### 3.9.4 Combining $K$ -mers into Pairs

In the final combining step, we form pairs of two  $k$ -mers if they satisfy the second chemical constraint set  $C_p$  (see Table 2.3). In a PCR with paired primers, we need to restrict the transcript regions to ensure overlap of reads and guarantee a common minimum length for all transcripts. This is because reads are usually trimmed to the same length in the bioinformatics pipeline - a requirement for most OTU clustering algorithms.

The strategy for pair-forming is to determine one  $k$ -mer to be the forward primer on the minus strand from 5' to 3' direction, and the second  $k$ -mer the reverse primer on the plus strand. Since  $k$ -mers with the same starting positions are encoded in a single TKMerID, for each *forward* and *reverse* TKMerID pair we have to iterate over all encoded  $k$ -mer combinations as shown in Figure 3.4 and Algorithm 11.

Note that FASTA libraries usually store sequences from 5' to 3' direction. Therefore, when choosing a second primer to be the reverse primer, the GC clamp and AT tail tests have to be applied to the prefix of a  $k$ -mer. A translation into the DNA complement is not necessary, since all chemical tests, except for the annealing, treat A and T, and C and G substitutionally.

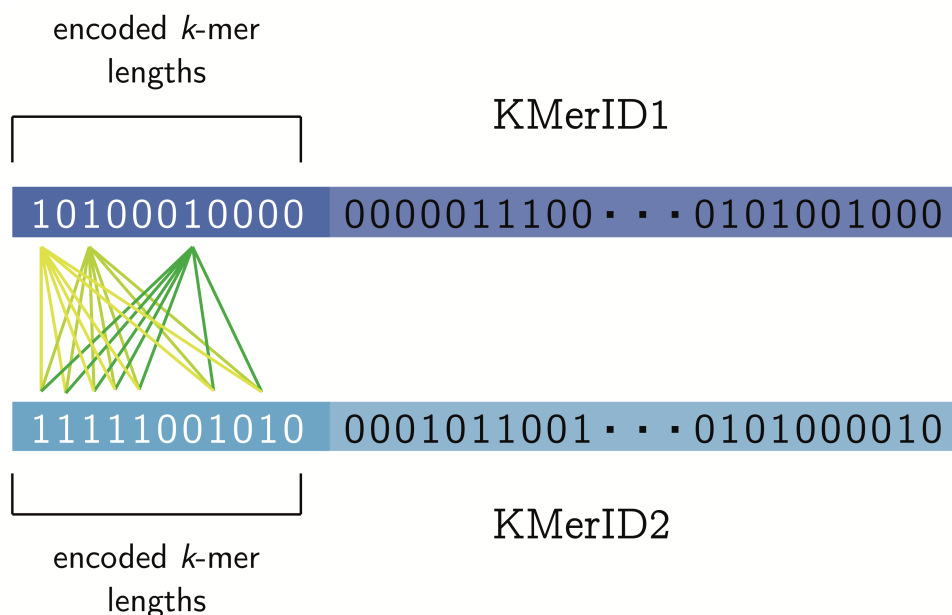


Figure 3.4: Possible combinations of  $k$ -mers encoded in two TKMerIDs. A single TKMerID encodes up to 11 different  $k$ -mers. When forming pairs at most, 121 unique combinations have to be considered, which can be stored in a fixed-length `std::bitset<121>` as part of a primer pair data structure (see `CombinePattern.hpp` in source code).

#### Filters for Primer Pairs

Two  $k$ -mers encoded in two distinct TKMerIDs represent a sound primer pair if in addition to  $C_s$  they satisfy the constraint set  $C_p$ . Concretely, we have to check for cross-annealing, GC clamps,  $(A|T)_3$  tails, and  $\Delta T_m$  (see Table 2.3). We can re-use the code of the self-annealing test for testing cross-annealing, except that we cannot exploit symmetry and have to test almost  $2k$  offset positions. GC clamps and  $(A|T)_3$  tail tests are carried out by comparing the tails against a fixed number of possible

---

**Algorithm 11** Combine  $k$ -mers sequence-wise (step 4). For each position referring to a TKMerID in the bit vector transformed reference, we fix the search window indices using  $\text{select}_1$ . Then for each forward and reverse  $k$ -mer combination encoded by TKMerID1 and TKMerID2, we evaluate chemical fitting and report them.

---

```

1: procedure Combine( $\mathcal{B}$ , KMerIDs)
2:   Pairs  $\leftarrow$  [[]] $n$ 
3:   for all  $B \leftarrow \mathcal{B}$  do
4:     for all  $r1 \leftarrow [1:\text{rank}_1(B, |B|)]$  do
5:        $\text{idx1} \leftarrow \text{select}_1(B, r1)$ 
6:        $w\_beg \leftarrow \text{idx1} + \kappa_{min} + \tau_{min}$ 
7:        $w\_end \leftarrow \text{idx1} + \kappa_{max} + \tau_{max}$ 
8:       for all  $r2 \leftarrow [\text{rank}_1(B, w\_beg) : \text{rank}_1(B, w\_end)]$  do
9:         TKMerID1  $\leftarrow$  TKMerIDs[ $r1$ ]
10:        TKMerID2  $\leftarrow$  TKMerIDs[ $r2$ ]
11:        for all (kmer1, kmer2)  $\leftarrow$  [TKmerID1, TKMerID2] do
12:          if  $C_p(\text{kmer1}, \text{kmer2})$  then
13:            Pairs  $\leftarrow$  Pairs + [(kmer1, kmer2)]
14:   return Pairs

```

---

tail patterns, and for computing  $\Delta T_m$ , we re-use the bit-parallel GC count function shown in Listing 3.2.

### 3.10 Theoretical Runtimes and Space Occupation

For the FM-index transformation of a library with a total size of  $N$  symbols we need  $\mathcal{O}(N)$  time and  $\mathcal{O}(\log |\Sigma|n) + o(\log |\Sigma|^2n)$  bits space as described in more detail in Section 3.9.1. The FM *frequency* computation performs a single  $k$ -mer look-up in  $\mathcal{O}(k)$  where  $k$  is the length of the  $k$ -mer to be looked up. Additionally, all occurrence locations need to be gathered, which depend on the number of occurrences  $occ$  and the total library size  $N$ . This can be done in  $\mathcal{O}(k + occ)$  by exploiting the lexicographical ordering (Ferragina and Manzini, 2000) and subsampling the suffix array. The space thereby occupied is  $\mathcal{O}(N)$  thanks to the  $k$ -mer compression scheme described in section 3.7 compared to  $\mathcal{O}(N(\kappa_{\max} - \kappa_{\min} + 1))$  when storing all  $k$ -mers separately. Taking into account at most  $N - k + 1$  different  $k$ -mers, the runtime is in  $\mathcal{O}(N(k + occ))$  (see Table 3.9).

All chemical filters in the filtering step analyze the encoded sequences (TKMerID) in a bit-parallel or single-pass fashion. Some require simple counting (Tm, GC-content) or pattern match (dinucleotide runs, (A/T)<sub>3</sub> tails). The most complex one is the annealing filter in which a  $k$ -mer is shifted at most  $2k$  times and XORed against its complement or reverse complement. The identification of a connected self-annealing pattern of size four, which corresponds to a  $\text{\textcircled{0}b11111111}$  pattern, can be made in constant time per shift position by using bit parallelism (see Listing 3.8). Filtering is applied to at most  $N$   $k$ -mers, giving us a total runtime of  $\mathcal{O}(\kappa_{\max}N)$ . For storing the bit vector transformed sequences, and the ranked TKmerIDs PriSeT requires  $\mathcal{O}(N)$  space.

Pairing information of  $k$ -mers needs to be stored in addition to the list of TKMerIDs and the associated bit vector. A single reference has an expected length of  $\frac{N}{n}$  bases. For each forward  $k$ -mer we search with an offset of  $\tau_{\min}$  a candidates' window of size  $\tau_{\max} - \tau_{\min}$ . Hence, we have at most  $\mathcal{O}((\frac{N}{n} - \tau_{\min})(\tau_{\max} - \tau_{\min}))$  pairs per reference<sup>13</sup>. For each pair, a match test is performed that is linear to the expected length of the two  $k$ -mers<sup>14</sup>, i.e., in  $\mathcal{O}(k)$ . In total, we have  $n$  references, giving us a combination runtime of  $\mathcal{O}(nk(\frac{N}{n} - \tau_{\min})(\tau_{\max} - \tau_{\min}))$ . The theoretical upper bounds are summarized in Table 3.9.

	FM-Index	FM Frequency
Runtime	$\mathcal{O}(N)$	$\mathcal{O}(N(\kappa_{\max} + occ))$
Space	$\mathcal{O}(\log  \Sigma N) + o(\log  \Sigma ^2N)$	$\mathcal{O}(N)$
	Filter & Transform	Combine
Runtime	$\mathcal{O}(\kappa_{\max}N)$	$\mathcal{O}(n\kappa_{\max}(\frac{N}{n} - \tau_{\min})(\tau_{\max} - \tau_{\min}))$
Space	$\mathcal{O}(N)$	$\mathcal{O}(n(\frac{N}{n} - \tau_{\min})(\tau_{\max} - \tau_{\min}))$

Table 3.9: Runtime classes and space occupation module-wise with  $N$  as the total library size,  $|\Sigma|$  is four, because of the underlying alphabet being  $\{A, C, G, T\}$ ,  $n$  the number of references per library,  $\kappa_{\max}$  the largest  $k$ -mer length,  $\omega$  the window width, and  $\tau_{\min/\max}$  the amplicon length limits, and  $occ$  the expected number of  $k$ -mer occurrences.

<sup>13</sup> $\kappa_{\min}, \kappa_{\max}$  are dropped here due to their relatively smaller size

<sup>14</sup>i.e. application of filter set  $C_p$



## 3.11 Example Applications

In this section, we evaluate two different scenarios. In the first use case the library is an explicitly *uncurated* plankton data set. The sequences are of short length and cover primarily 18S. The plankton dataset is taxonomically broad as outlined in the study of Section 2.5. The task is to compute *de novo* primer pairs exhibiting a broad coverage in terms of taxa. The second scenario is diametrically opposite. We are given a few complete viral genomes (SARS-CoV-2) to compute primer pairs producing species-distinctive amplicons.

Put in front is a proof of concept test, in which the primer pairs computed by PriSeT are searched for primer pairs that have been published in previous metabarcoding studies. Lastly, the experimental runtime is measured on the plankton data set.

## 3.12 Plankton

Plankton clades are taxonomically very heterogeneous, as shown in Figure 2.3. The phylogenetic diversity is challenging for identifying new primer pairs in common, and in addition, the sparsity of the reference database limits the search options severely. In the subsequent experiments, we apply PriSeT to identify frequent pairs and compare the sequences with published primer pairs. We further compared taxonomic coverage and barcode suitability.

### 3.12.1 Data Set for Plankton

As a reference library, NCBI GenBank's nucleotide collection (Benson et al., 2012) was sampled<sup>15</sup> with *tactac*<sup>16</sup>, which contains non-human sequences from various sources. The prevalent sequence length range is between 400 to 2500 bases. We picked nineteen clades that include eukaryotic groups typically found in freshwater plankton samples ranging from phytoplankton to zooplankton and fungi. For each taxon within a clade that contained at least one accession assigned to it, we sampled at most three accessions to remove the sequence bias introduced by highly populated taxa. Table 3.10 lists the numerical identifier, clade names, their number of taxa, taxa with at least one accession (Covered), total number of accessions, and library sizes in megabytes (MB). We sampled between 1.38 (Rotifera) to 2.5 (Charophyceae) accessions per covered taxon (Accs/Covered), which illustrates the sparseness of plankton clades.

### 3.12.2 Verification of Published Primer Pairs

Bacillariophyta (diatoms) and green algae (mostly Chlorophyta) are among the most diverse and abundant organisms in freshwater and marine plankton communities. Bacillariophyta are small (2 - 200  $\mu\text{m}$ ), and their characteristic silica cell walls allow morphological identification by trained experts. They contribute approximately 20 % of global oxygen production and represent nearly half of the organic material in the oceans. We use the DIV4 primer pair described in Section 2.5.2 which was specifically designed for Bacillariophyta by Visco et al., 2015 (see Table 2.7). Hadziavdic et al., 2014 optimized primers towards a broad coverage of Eukaryota (*universal* eukaryotic primers) by aligning all sequences from the SILVA database and computing the entropy

<sup>15</sup>accessed on 29.03.2019

<sup>16</sup><https://github.com/mariehoffmann/tactac>

	Clade	Name	Taxa	Covered	Accs	Lib Size
Phytoplankton	33849	Bacillariophyta	2,060	1,724	3,474	4.98 MB
	304574	Charophyceae	153	138	350	0.42 MB
	3041	Chlorophyta	10,490	9,466	15,377	31.79 MB
	2825	Chrysophyceae	428	339	507	0.89 MB
	3027	Cryptophyta	396	344	653	1.97 MB
	2864	Dinophyceae	6,147	4,630	7,151	6.24 MB
	33682	Euglenozoa	1,912	1,710	3,254	19.29 MB
	5747	Eustigmatophyceae	250	215	344	1.4 MB
Zooplankton	554915	Amoebozoa	3,211	2,817	3,898	4.63 MB
	33651	Bicosoecida	101	79	119	0.15 MB
	28009	Choanoflagellata	131	88	186	0.28 MB
	136419	Cercozoa	1,221	953	1,562	2.34 MB
	5878	Ciliophora	4,101	2,977	4,868	6.94 MB
	6657	Crustacea	45,058	25,643	50,163	38.85 MB
	6231	Nematoda	13,954	12,086	20,975	82.44 MB
	27999	Perkinsidae	81	75	114	0.13 MB
10190	Rotifera	1,429	1,254	1,727	1.57 MB	
Fungi	451864	Dikarya	141,097	129,254	209,449	518.44 MB
	112252	Fungi i. s.	7,169	5,948	9,149	10.21 MB

Table 3.10: Data set used for the primer verification test (Section 3.12.2), the *de novo* search (Section 3.12.3), and the runtime analysis (Section 3.12.4). Taxonomic identifiers in the clade column follow NCBI's nomenclature, *Taxa* refers to the total number of nodes (including virtual ancestors), *Covered* to the number of taxa having at least one accession assigned to it, *Accs* to the total number of collected accessions and *Lib Size* to the size of the FASTA file containing all accessions.

at each alignment position. The authors identified eight forward and six reverse primer candidates from the low-entropic regions. Here we used F-566a as forward, and R-1200 as a reverse primer (EUK14 from Table 2.7 and E14 hereafter).

Other tested primers are 23S by Yoon et al., 2016 developed for marine phytoplankton (23S hereafter), ChloroF/R by Moro et al., 2009 for Chlorophyceae and Bacillariophyceae (CHL hereafter), CVfor/rev by Boscaro et al., 2017 for freshwater ciliates (CV hereafter), D512for/D978rev by Zimmermann, Jahn, and Gemeinholzer, 2011 for diatoms (DIA hereafter), TAReuk454FWD1/TAReukREV3 by Stoeck et al., 2010 (EUK15 from Section 2.5 and E15 hereafter) for the V4 rRNA region of marine Eukaryota, EUKAF/R by Moreno et al., 2018 for the 18S region of Protozoa (EA hereafter), G18S4/22R by Blaxter et al., 1998 for Nematoda (nSSU hereafter), and SSU556F/SSU911R by Smith et al., 2017 for Dinoflagellata (SSU hereafter). Some of these primers had been designed for a specific organism group. However, we expect that they are also effective in other clades.

Primer ID	Name	Sequence (5' - 3')	Tm [°C]	GC [%]
23S	A23SrVF1	GGAC <b>ARAA</b> AGACCCTATG <sup>β</sup>	54.9	47.2
	A23SrVR1	AGATCAGCCTGTTATCC <sup>α</sup>	52.6	47.1
CHL	ChloroF	TGGCCTATCTTGTGGTCTGT <sup>α</sup>	63.8	47.6
	ChloroR	GAATCAACCTGACAAGGCAAC	63.8	47.6
CV	CVfor	CCAGCASC CGCGGTAATWCC	<b>71.6</b> <sup>δ</sup>	<b>65.0</b> <sup>γ</sup>
	CVrev	TCTGRTYGTCTTTGATCCCYTA	<b>62.8</b> <sup>δ</sup>	43.2
DIA	D512for	ATTCCAGCTCCAATAGCG <sup>α</sup>	<b>60.9</b> <sup>δ</sup>	50.0
	D978rev	GACTACGATGGTATCTAATC	<b>50.7</b> <sup>δ</sup>	40.0
DIV4	DIV4for	GCGGTAATTCAGCTCCAATAG <sup>α</sup>	<b>65.8</b> <sup>δ</sup>	50.0
	DIV4rev3	CTCTGACAATGGAATACGAATA	<b>58.7</b> <sup>δ</sup>	<b>36.4</b> <sup>γ</sup>
E14	F-566a	CAGCAGCCGCGGTAATTCC <sup>α</sup>	<b>70.2</b> <sup>δ</sup>	<b>63.2</b> <sup>γ</sup>
	R-1200	<b>CCCGT</b> GTGAGTCAAATTAAGC <sup>γ</sup>	<b>64.7</b> <sup>δ</sup>	45.5
E15	TAReuk454FWD1	CCAGCASCYCGGTAATTCC <sup>α</sup>	<b>70.8</b> <sup>δ</sup>	<b>62.5</b> <sup>γ</sup>
	TAReukREV3	ACTTTCGTTCTTGATYRA	<b>53.3</b> <sup>δ</sup>	<b>33.3</b> <sup>γ</sup>
EA	EUKAF	GCCGCGGTAATTCCAGCTC <sup>α</sup>	<b>69.2</b> <sup>δ</sup>	<b>63.2</b> <sup>γ</sup>
	EUKAR	CYTTCGYCTTGATTRA	<b>55.2</b> <sup>δ</sup>	41.2
nSSU	G18S4	GCTTGTCTCAAAGATTAAGCC <sup>α</sup>	<b>60.0</b> <sup>δ</sup>	42.9
	22R	<b>GCCTGCT</b> GCCTTCCTTGGA <sup>γ</sup>	<b>70.3</b> <sup>δ</sup>	<b>63.2</b> <sup>γ</sup>
SSU	SSU556F	CGCGGTAATTCCAG <b>CTYC</b> <sup>αγ</sup>	64.8	58.3
	SSU911R	<b>ATYCA</b> AGAATTTACCTCTGAC <sup>αε</sup>	60.2	<b>38.6</b> <sup>γ</sup>

Table 3.11: Selected primer sequences for the verification experiment. Sequences are noted in 5' to 3' direction. Melting temperatures were computed using a modified nearest-neighbor method described by Breslauer et al., 1986, assuming a primer concentration of 0.5 μM and salt 50 mM. For ambiguous encodings, the average of both extrema was taken. Sequence parameters violating primer design recommendations are written boldly. Critical structures are marked: self-annealing (α), mononucleotide runs (β), GC clamps (γ), exceeding ΔTm (δ), (A|T)<sub>3</sub> tails (ε), and exceeding CG-content ranges (γ).

Table 3.11 lists the ten selected primer pairs targeting 18S that we searched for in the reference library. Remarkably, not a single pair has chemically optimal properties:

- At least one sequence of each primer pair shows a self-annealing pattern.

Parameter	Verification	<i>De Novo</i>
$k$	[16 : 25]	[16 : 25]
$\tau$ [nt]	[30 : 800]	[30 : 800]
Tm [°C]	[50 : 60]	[52 : 58]
GC [%]	[35 : 65]	[40 : 60]
4-Runs of C or G	no	no
Self-Annealing	off	on
$\Delta T_m$ [K]	10	5
Cross-Annealing	off	on

Table 3.12: Filter settings in PriSeT for the primer verification experiment and the *de novo* primer discovery on the plankton data sets.

- Seven pairs differ significantly in melting temperatures (independent of the computation method, i.e., Wallace rule or nearest-neighbor method).
- Three sequences have GC clamps at their 3' ends.
- Eight sequences have a GC-content that exceeds the recommended range.
- Primer A23SrVF1 contains a run of five adenine bases (R substitutes A or G).
- Primer SSU911R has an (A|T)<sub>3</sub> tail.

Therefore, we relaxed the chemical constraints for the verification experiment by allowing a broader melting temperature range and difference  $\Delta T_m$ , a larger GC-content range, and we deactivated the self-annealing filter (see settings in Table 3.12).

Given the sampled library, we first computed the ground truth by searching the library for the known primer sequences via a linear text search. All data sets were marked that had at least one accession with forward and reverse primers matching as indicated by a presence (1) or absence (0) bit in the denominator of each cell in Table 3.13. Then PriSeT was run with the relaxed constraints listed in the Verification column of Table 3.12, and we searched the result set for the published primer sequences listed in Table 3.11. Presence or absence in the result set is indicated in the numerator.

As a result, we could recover all those primer sequences that satisfied the relaxed constraints. The forward primer of DIV4 and both primers of CV did not pass the relaxed Tm filter setting ( $T_m > 65$  °C). The primer pairs E15 and EA failed for  $C_p$  because of their excessive differences in melting temperatures (more than 8 Kelvin).

### 3.12.3 *De Novo* Computation

We reset the primer constraints to the recommended ranges (see Table 2.3 in Section 2.3.2) to yield chemically uncritical primer pairs. The 50 most frequent  $k$ -mer pairs of each clade were then reported<sup>17</sup>. Moreover, to compare PriSeT's *de novo* pairs with the published ones, we computed frequencies, coverage rates, and amplicon variations. The computation was performed via text search (see Table 3.14).

The *de novo* and published primer result sets were either ranked by taxon coverage (see left side of Table 3.14) or by amplicon variation (see right part of Table 3.14). From

<sup>17</sup>labeled by hashing forward and reverse sequence

	Clade	23S	CHL	CV*	DIA	DIV4*	E14	E15**	EA**	nSSU	SSU
Phytoplankton	33849	1/1	0/0	0/0	1/1	0/1	1/1	0/1	0/1	1/1	1/1
	304574	1/1	0/0	0/0	0/0	0/0	1/1	0/1	0/1	0/0	1/1
	3041	1/1	1/1	0/0	1/1	0/0	1/1	0/1	0/1	1/1	1/1
	2825	0/0	0/0	0/0	1/1	0/1	1/1	0/1	0/1	1/1	1/1
	3027	1/1	0/0	0/0	0/0	0/0	1/1	0/1	0/1	1/1	1/1
	2864	1/1	0/0	0/1	1/1	0/1	1/1	0/1	0/1	1/1	1/1
	33682	1/1	0/0	0/0	0/0	0/0	1/1	0/1	0/0	0/0	0/0
	5747	1/1	1/1	0/0	1/1	0/0	1/1	0/1	0/1	1/1	1/1
Zooplankton	554915	0/0	0/0	0/1	0/0	0/0	1/1	0/1	0/1	1/1	1/1
	33651	0/0	0/0	0/0	0/0	0/0	1/1	0/1	0/1	1/1	1/1
	28009	0/0	0/0	0/0	0/0	0/0	1/1	0/1	0/1	1/1	1/1
	136419	0/0	0/0	0/0	0/0	0/0	1/1	0/1	0/1	1/1	1/1
	5878	0/0	1/1	0/1	0/0	0/0	1/1	0/1	0/1	1/1	1/1
	6657	0/0	0/0	0/0	0/0	0/0	1/1	0/1	0/1	1/1	1/1
	6231	0/0	0/0	0/0	0/0	0/0	1/1	0/1	0/1	1/1	0/0
	27999	0/0	0/0	0/0	0/0	0/0	1/1	0/1	0/1	0/0	1/1
	10190	0/0	0/0	0/0	0/0	0/0	1/1	0/1	0/1	0/0	0/0
Fungi	451864	0/0	0/0	0/1	0/0	0/1	1/1	0/1	0/1	1/1	1/1
	112252	0/0	0/0	0/1	1/1	0/0	1/1	0/1	0/1	1/1	1/1

Table 3.13: Primer presence test for established primer pairs.  $x/y$  refers to whether a pair was found by PriSeT ( $x = 1$ ) or not ( $x = 0$ ) versus the pair was found in the library ( $y = 1$ ) or not ( $y = 0$ ). Primer pairs that were not discovered by PriSeT either did not pass the first chemical filter set  $C_s$  (labeled with \*) or the pair filter set  $C_p$  (labeled with \*\*).

the *de novo* set and the published primers set, we reported only the top-ranked ones. For example, for Charophyceae (clade 304574), the top-ranked primer by coverage is d8d47dc9b873d02b – a *de novo* computed primer, and the highest-ranked published primer is EUK14. In the common set, however, there are more *de novo* primer pairs that have a higher rank than EUK14 but are not reported here for the sake of brevity. When ranking by coverage, for 11 out of 19 clades a *de novo* primer outperformed all published primers.

Since coverage optimizes towards broadness and variation towards amplicon distinguishability, one expects that the top primers for the two ranking methods may differ. In 14 out of 38 cases, the top primer pairs of the *de novo* or published sets were identical, i.e., the ranking method had no influence. In 24 cases, a higher variation was traded against a lower coverage and *vice versa*. When ranking by number of unique amplicons (variation), we identified seven *de novo* primers that performed equally well (see Table 3.14).

The result is surprising in light of the well-known sparsity problem in planktonic clades and the apparent fact that the GenBank reference sequences are from PCR amplicons with previously published primer sets. However, with PriSeT we were able to identify new primer pairs having a broader coverage or variation for some clades.

Published primers outperformed *de novo* primers by at most 7 %, and *de novo* the published ones by at most 70 % for coverage. Nonetheless, other performant primer pairs may exist for specific clades that the author is not aware of – the most promising ones were chosen from the PR2 database<sup>18</sup>. Furthermore, EUKA/B from Medlin et al., 1988, Cerc479F/Cerc750R by Harder et al., 2016, and DimA/DimB by Cannon et al., 2018 were tested, which either did not show up or occurred with very low frequencies.

From the published primers E14 and EUKA are certainly versatile primer pairs. However, they do not satisfy  $C_s$  or  $C_p$  and might require more lab experience when applied in a PCR. The complete list of top-performing primers is available on GitHub<sup>19</sup>.

### 3.12.4 Performance on Plankton Data Set

For each clade, we measured the step-wise runtimes excluding the FM-index computation (see Figure 3.5). The index computation needs to be done only once and upon updates of the original library. The FM-index on the plankton data set consumed about 4.2 times more space than the library it is computed on<sup>20</sup>.

We set the relative frequency cutoff for  $k$ -mers to 5 %. The number of  $k$ -mer locations grows, therefore, linearly with the library size. We have chosen not to show the performance on a synthetic data set because  $k$ -mer frequencies and dropout rates are defined by inherent sequence properties (entropy, repeats, and other), which are not homogeneous for all clades. A single clade or synthetic data set would produce non-representative, and even misleading results. For example, clade 6657 has a library which is 7 MB larger than the one of clade 3041. Surprisingly, clade 6657 produces only 4.83 million  $k$ -mers, compared to 43,9 million  $k$ -mers of clade 3041 (see Figure 3.6). For the largest data set (clade 451864 of Dikarya), the frequency cutoff had to be raised to 10 %. Otherwise, the vast amount of  $k$ -mers causes memory issues on a laptop with 16 GB RAM (see Discussion).

Data sets sizes (abscissa) are plotted against the runtimes (ordinate). Both axes are  $\log_2$ -scaled for readability. The total runtime for the smallest data set Perkinsidae

<sup>18</sup><https://pr2-database.org>

<sup>19</sup>[https://github.com/mariehoffmann/PriSeT\\_denovo](https://github.com/mariehoffmann/PriSeT_denovo)

<sup>20</sup>theoretical complexity class for space occupation is  $\mathcal{O}(N \log |\Sigma|)$

Clade	Primer	Frequency	Coverage ↑	Variation	Primer	Frequency	Coverage	Variation ↑
33849	SSU	768	0.43 (734/1724)	631	EUK14	755	0.42	651
	b099967d0f5ac180	750	0.39 (673/1724)	27	33c14haf2ac76276	675	0.38	591
304574	eb59a790ecec1766	35	0.23 (32/138)	13	d8d47dc9b873d02b	31	0.22	25
	EUK14	30	0.21 (29/138)	25	EUK14	30	0.21	25
3041	SSU	3615	0.36 (3404/9466)	1715	EUK14	3217	0.32	1835
	b7488789aaf96d7b	3051	0.3 (2824/9466)	902	1.426f6f9b97f501a	2572	0.26	1425
3027	412d47502c9c1b	248	0.66 (226/344)	124	EUK14	237	0.65	153
	EUK14	237	0.65 (223/344)	153	8d14274d691d15e5	234	0.64	152
2825	SSU	241	0.69 (235/339)	181	SSU	241	0.69	181
	96c04483d9557b08	234	0.65 (222/339)	66	cce22a01d74086cd	197	0.58	168
2864	EUK15	848	0.16 (756/4630)	608	EUK14	821	0.16	634
	24555f6837b9651f	860	0.16 (742/4630)	408	d90e133cfe247ad5	693	0.14	564
33682	b805fd5bf8167cc7	539	0.27 (465/1710)	25	af8f7968a5ee777e	367	0.20	129
	235	114	0.07 (113/1710)	107	235	114	0.07	107
5747	EUKA	140	0.61 (132/215)	73	EUK14	138	0.61	84
	35050c2634d666c7	145	0.61 (131/215)	3	fc89901444ed4b0	141	0.61	13
554915	6f74c4dbf4ceae12	940	0.31 (885/2817)	57	EUKA	676	0.23	523
	EUKA	676	0.23 (650/2817)	523	4dbdfed2605e324	788	0.27	116
33651	f83f6dae50175268	64	0.73 (58/79)	23	EUKA	59	0.71	47
	EUKA	59	0.71 (56/79)	47	f83f6dae50175268	64	0.73	23
28009	2afb9ade5ef548f	49	0.55 (48/88)	47	2afb9ade5ef548f	49	0.55	47
	EUKA	48	0.53 (47/88)	47	EUK14	47	0.53	47
136419	EUKA	503	0.50 (479/953)	412	EUKA	503	0.50	412
	24555f6837b9651f	440	0.43 (411/953)	298	cb6c82029453de6a	378	0.38	326
5878	EUKA	2056	0.64 (1890/2976)	1501	EUKA	2056	0.64	1501
	5f5fd94bf3f7db4	1901	0.57 (1675/2976)	1087	c2ee8143ec040d5a	1494	0.47	1188
6657	38f9227a340f05e	2696	0.10 (2617/25643)	1651	38f9227a340f05e	2696	0.10	1651
	EUK14	1745	0.07 (1719/25643)	1458	EUK14	1745	0.07	1458
6231	edcd39ccccc9d72a	3158	0.24 (2883/12086)	2133	edcd39ccccc9d72a	3158	0.24	2133
	EUKA	2452	0.19 (2350/12086)	1949	EUKA	2452	0.19	1949
27999	ecb3cb9fe242b24e	63	0.71 (53/75)	15	7aab6403d6856205	61	0.69	22
	EUK15	6	0.08 (6/75)	5	EUK15	6	0.08	5
10190	57bc43fe1080644d	224	0.18 (224/1254)	1	EUKA	69	0.05	66
	EUKA	69	0.05 (66/1254)	66	bbcb9dc15a5fc34c	216	0.17	40
451864	3f32736b9d094915	24868	0.17 (22387/129253)	12378	3f32736b9d094915	24868	0.17	12378
	SSU	5533	0.04 (5255/129253)	3007	SSU	5533	0.04	3007
112252	EUKA	1017	0.17 (990/5948)	882	EUKA	1017	0.17	882
	7e079b504409f8c0	979	0.15 (908/5948)	822	4dbdfed2605e324	788	0.27	116

Table 3.14: *De novo* computation and statistical comparison with published primers ranked by reference coverage (left) or amplicon variation (right). From the sets of *de novo* and published primers, the top-performing primer pair in terms of coverage or variation were picked. Frequency reflects the raw number of occurrences in the clade, whereas coverage computes how many taxa of each clade were covered in proportion to taxa with references, and variation the total number of amplicons. Note, that PriSeT computed primer pairs with narrowed constraints – primer pairs like SSU, EUKA, and others, would not emerge in the result set. When ranking by coverage, for 11 out of 19 clades, PriSeT identifies at least one new primer pair with a higher coverage rate than the published primers. Whereas when ranking by amplicon variation for seven out of 19 clades, PriSeT found at least one more or equally performant primer pair.

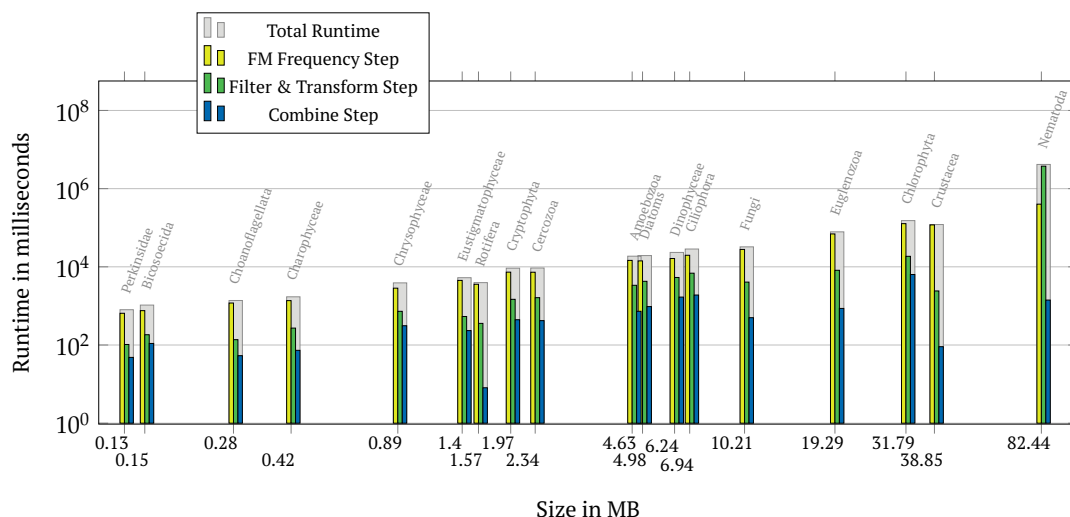


Figure 3.5: Runtimes for all clade data sets broken down to FM frequency computation, filter, and combine step. We set the  $k$ -mer frequency cutoff was set to 5 % w.r.t. the number of references per clade. The results for Fungi (clade 451864, 500 MB large) are omitted here due to the necessity of setting the cutoff to 10 %, which results in runtimes comparable with clade 6231 (82 MB). Both axes are log-scaled. The runtimes are listed in tabular format in Table B.2.

(clade 27999 with 0.13 MB) is  $\leq 1$  second, for fungi (clade 112252 with 10.21 MB) 33 seconds, and a large dataset like Nematoda (clade 6231 with 82.55 MB) 70 minutes.

The FM frequency computation contributes the most to the total runtime. This is due to the large number of possible  $k$ -mers within a library. For each separate call of FM frequency with a fixed value for  $k$ , all  $\mathcal{O}(N)$   $k$ -mers have to be held in main memory until the location gathering has terminated and low-frequency  $k$ -mers can be permanently dropped. The low-frequency cutoff reduces the number of  $k$ -mers drastically, such that the contribution of the expensive combine step remains relatively low.

The dropout rate during the filter and combination steps depends strongly on the sequence structure within the clades, as shown by the strongly varying run times of the filter and combination steps in proportion to the original library size. Theoretical runtimes were analysed in Section 3.10 (see Table 3.9).

In general, if the number of suitable primers is sufficiently low, it may be worthwhile to pre-calculate primer sequences and use the FM index only to search for location and abundance. The  $k$ -mer counts in Figure 3.6 suggest that the remaining amount is infeasible to precompute: about one-third of the frequent  $k$ -mers (FM Frequency Step) is chemically suitable. Between  $2^{12}$  and  $2^{18}$   $k$ -mers remain as candidates (Transform & Filter Step) for pairing.



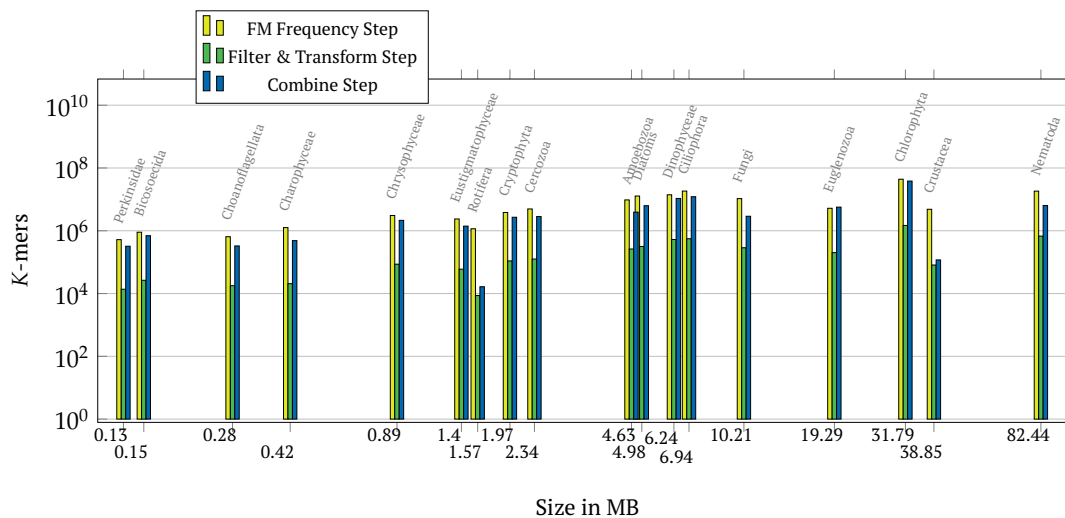


Figure 3.6: K-Mers for all clade data sets counted after the FM frequency computation, filter, and combine steps. For the combining step, we counted pairs, not  $k$ -mers. The settings were the same as in Table 3.5.

### 3.13 SARS-CoV-2

The problem of data sparseness of plankton clades does not hold for viral pathogens. Contrarily, a lot of effort is spent to sequence as many genomic variants as possible. The data abundance allows for correct phylogenetic placement and understanding of their evolution or functional deviation from related species. Now, given a set of complete genomes, we wish to obtain primers that produce sequences that are guaranteed to have no matches outside a specific taxon. When looking at the current SARS-CoV-2 outbreak, such primers can be used for processing saliva samples of patients with flu-like symptoms. These samples contain hundreds of bacteria, fungi, and virus species. Needless to say that a recall of 100 % is of uttermost importance – an unrecognized SARS-CoV-2 infection will impede efforts to contain the outbreak. On the other hand, a false-positive diagnosis imposes an unnecessary burden on the patient and the healthcare system.

For this experiment, we searched for primers on complete SARS-CoV-2 genomes<sup>21</sup>. We then filtered for those having distinct transcripts from their closest relatives within the Orthocoronavirinae, a subfamily within the Coronaviridae. Online BLAST searches against the complete *nt/nr* dataset from GenBank showed that in addition, no sequence matches outside the coronavirus family occurred.

#### 3.13.1 Genome and Functional Units

The SARS-CoV-2 virus belongs to the family of coronaviruses. Its genome is a single linear, positive-sense RNA sequence (+ssRNA) with a length of 29,727 bases. It is the seventh known coronavirus to infect humans (Zhu et al., 2020). Notable outbreaks of other coronaviruses occurred in 2002-2004 of SARS-CoV (Forgie and Marrie, 2009); MERS-CoV<sup>22</sup> in 2012, 2015, 2018.

The known open reading frames (ORFs) are shown in Figure 3.8. SARS-CoV-2 viruses can enter cells by binding to two types of cellular receptors, which are

<sup>21</sup>A virologist might exclude some regions.

<sup>22</sup><https://www.who.int/csr/don/24-july-2019-mers-saudi-arabia/en/>

ACE2 (angiotensin-converting enzyme 2) and DPP4 (dipeptidyl peptidase 4) via the S protein (Song et al., 2019). In the cell's cytoplasm, the ORFs 1a and 1b are translated and cleaved into an RNA replicase-transcriptase complex, which promotes replication and transcription of more RNAs, first into negative-sense, then into positive-sense RNA. The copies are encapsulated by N proteins (from ORF10) and released through exocytosis.

### 3.13.2 Dataset for SARS-CoV-2

19 complete SARS-CoV-2 genomes<sup>23</sup> were selected from the subgenus Sarbecovirus for PriSeT to compute primer pairs satisfying the RT-PCR constraints listed in Table 3.15. To filter for primer pairs producing amplicons with no co-occurrences in other Orthocoronavirinae genomes, we downloaded all available genomes from GenBank – 24 Alphacoronavirus, 5 Betacoronavirus (excluding *Sarbecovirus*), and 2 Gammacoronavirus genomes (see Figure 3.7).

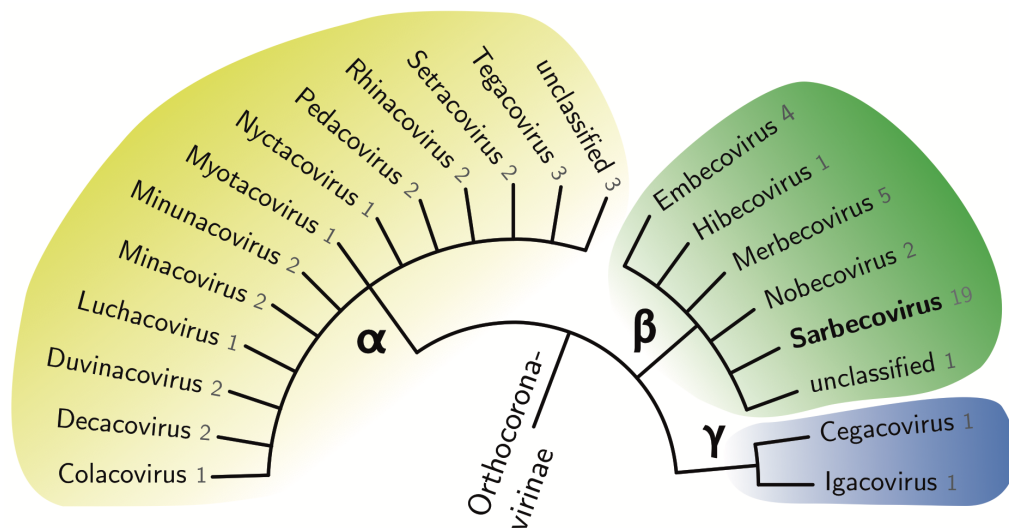


Figure 3.7: Taxonomy of Orthocoronavirinae to subgenus level. The numbers indicate how many complete genomes we used for the primer *de novo* search of each subgenus. From the *Sarbecovirus* subgenus, all 19 genomes are assigned to the species SARS-CoV-2. Their proximity to Bat SARSr-CoV (another *Sarbecovirus*) was revealed by a BLAST search of amplicons of *de novo* primer pair candidates.

### 3.13.3 De Novo Computation

Since we have RNA as a template, a different protocol needs to be applied that first produces a complementary DNA sequence with reverse transcriptase, which is then PCR-amplified (see RT-PCR in Section 2.3.3). The settings are narrower compared to a PCR for plankton samples. The optimal primer length is 20 bp, the amplicon length between 60 to 150 bp, and the GC-content between 50 to 60 %.

PriSeT produced 286 primer pairs given the settings listed in Table 3.15. We filtered for pairs producing amplicon sequences that have no co-occurrences in one of the other 39 coronavirus genomes (based on a 100 % sequence identity). For the 114 remaining primer pairs, we ensured amplicon distinction by launching two BLAST

<sup>23</sup>Downloaded from GenBank on 3rd April 2020. Accessions can be requested from the author.

Parameter	Settings SARS-CoV-2
$k$	[18 : 24]
$\tau$ [nt]	[60 : 150]
Tm [°C]	[55 : 63]
GC [%]	[50 : 60]
4-Runs of C or G	yes
Self-Annealing	on
$\Delta Tm$ [K]	5
Cross-Annealing	on

Table 3.15: Filter Settings in PriSeT for *de novo* primer discovery on SARS-CoV-2 genomes. Properties not listed follow the properties of the standard protocol.

queries for each of the 114 amplicons against GenBank’s nucleotide collection (*nt/nr*) online<sup>24</sup>. We ran the first query on the complete *nt/nr* data set and the second on the complete *nt/nr* data set except SARS-CoV-2 (taxonomic ID 2697049) to ensure that we did not miss relevant matches with non-SARS-CoV-2 entries.

None of the primer pairs produced amplicons with 100 % identity and 100 % coverage simultaneously for non-*Sarbecoviruses*. Of the 114 primer pairs, five had 100 % sequence identity with a single accession of a *Sarbecovirus* isolated from the pangolin, but no other relevant matches. There were 109 primer pairs producing amplicons with no co-occurrences. Out of the 109 primer pairs, we found 12 with amplicons distant from even closely related viruses. Concretely, the sequence identity was below 97 %, and 97 had proximity (but not identity) to at most two other accessions, namely, two *Sarbecovirus* species isolated from a bat and the pangolin. The first one is associated with the recent pneumonia outbreak (Zhou et al., 2020).

The complete list of primer sequences and amplicons can be found in the Appendix B.4. None of the primer sequences is identical to the published coronavirus (2019-nCoV) real-time RT-PCR primers<sup>25</sup>, which are 2019-nCoV\_N1-F/R, 2019-nCoV\_N2-F/R, 2019-nCoV\_N3-F/R, and RNase P (RP-F/R) (see sequences in Appendix B.3). However, the forward primer 2019-nCoV\_N2-F occurred in transcripts of three *de novo* primer pairs computed by PriSeT.

The approximate amplicon locations relative to the genome are shown in Figure 3.8. Most notable is the cluster of more than 60 barcodes around the 3’ end. Each cluster corresponds to a region that contains changes unique to the newly evolved SARS-CoV-2 virus.

<sup>24</sup>on 3rd of April 2020

<sup>25</sup><https://www.cdc.gov/coronavirus/2019-ncov/lab/rt-pcr-panel-primer-probe.html>

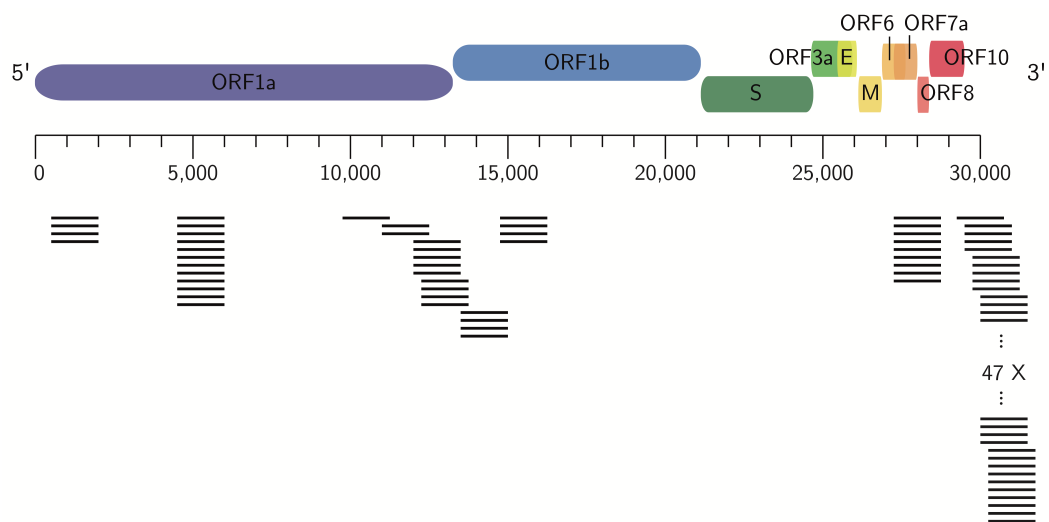


Figure 3.8: Top: Genome organization and functional domains of SARS-CoV-2 based on GenBank MN908947.3 and NCBI's ORFfinder. Bottom: Transcript positions for *de novo* primers. Note that transcript lengths are scaled up relative to genome length for readability.

## Chapter 4

# A Database for Metabarcoding Experiments

Ordnung ist die Verbindung des Vielen  
nach einer Regel.

---

*Immanuel Kant*

### 4.1 Problem Statement

Metabarcoding experiments are highly sophisticated procedures involving many actors and technical components. They are usually conducted over months or even decades, as in the long-term monitoring project presented in the study (see Section 2.5). Here we look at the workflow of a research group with access to lab services like sample collection, microscopic analyses, and molecular services like DNA extraction, PCR, and sequencing. The fundamental task to advance research in environmental monitoring and understanding freshwater ecosystems is the iterative improvement of all methods that contribute to the species identification of heterogeneous mixtures.

The current state at the institute's lab is that data and information are physically scattered over servers, workstations, email accounts, and individuals. In order for new entrants to reach an informed and contributing state, they must contact a list of individuals, require access to at least one server, and must evaluate multiple documents. If entrants intend to reproduce a metabarcoding pipeline, they have to comprehend the digital remains in terms of scripts and intermediate data products. There are no standards for protocolling and documentation. Individual group members who have key information are a bottleneck because they may be vacant or ill.

When looking at typical digital service providers that are being faced with workflow issues for decades, we can identify and transfer some of their methods to a lab environment to alleviate data management. Service providers have three major concerns:

1. Reliability
2. Scalability
3. Maintainability

A service is reliable if it tolerates the faults of components. It is scalable if the system can deal with growing data volumes, network traffic, or more complex computations, and it is maintainable if engineers can work *productively* over multiple generations. Maintainability is especially an issue for software that is in use 24 hours.

Bug fixing and features development are handled by a globally distributed team (Follow-the-sun model, see Carmel, Dubinsky, and Espinosa, 2009). A remote developer can take over the task of his predecessor immediately. Something impossible without strict coding style guidelines, and documentation.

Maintainability and scalability are major concerns of research labs but are solved insufficiently. While missing maintainability is pressing, scalability will become an issue soon as sequencing costs drop faster than storing a base on a hard disk. In this section, we address maintainability by proposing a database scheme that captures relevant components and relationships. When given access to the hosting database server, the entrant reaches a productive state faster, is admonished to document, and stores a result in a consistent way for co-workers and successors. The schema intends to answer typical entrants' questions like

- How frequently are all sites sampled?
- Which primer pairs have been tested so far?
- Where is the data set used in study X?
- Which tools have been used in previous studies to build a bioinformatics pipeline?
- How many different DNA extraction kits have been in use since last year?

Also, the consolidation of data and analysis results allows for new types of meta-analyses that are, at the moment, hard to solve, or even infeasible. However, these meta-analyses are important for iterative method improvement as they provide a form of feedback both for sampling and bioinformatics analysis. For example, having experimental data consolidated, we will be able to answer questions like:

- What was the average number of OTUs a primer pair produced given identical sample treatment?
- Which primer pair is most effective on Cryptophyta's group?
- Which plankton samples underwent both microscopic and metabarcoding analysis?
- What is the average number of reads produced when using freeze-drying versus airdrying as a sample treatment?

Consolidating trial data allows users to analyze results from previous trials, calculate statistics, identify study-related records without delay in human-to-human communication, or restart entire metabarcoding pipelines. A data schema can also be used for much simpler but common tasks to identify institution-specific terminology that cannot be looked up via web search. The concern of scalability is addressed in the final discussion (Section 5.3).

In this section, we refer to *data* as raw, unorganized facts, as opposed to *information* that refers to processed, organized, and structured data. In the following, we will use the more general term *data* to refer to unstructured and structured data for the sake of brevity.

## 4.2 Motivation

For lake monitoring, researchers at the IGB apply more and more DNA metabarcoding as it allows for higher precision (see Section 2.4.2) and can be executed at a higher pace compared to manual sample evaluation. The constant improvement of marker efficiency and bioinformatics pipelines contributes to this trend. The cost of sequencing a base has halved roughly every five months, whereas the costs for storing a byte are only halving every 14 months (Stein, 2010). According to Stein, 2010, the prediction is that in a not too distant future, sequencing a base will cost less than storing it on a hard disk. The accumulation of sequence and experimental data is merely beginning.

We want to structure trial-related data by defining a database schema that captures the relevant and frequently queried data. Structured and merged data would allow entrants to work more self-independent in contrast to waiting for documents to be shared. Worse, there were situations where scripts and analysis results were lost because the researcher had left the lab.

Another motivation is the increase in remote work in light of the ongoing pandemic. Before the pandemic, about 5 % of North Americans worked remotely; by 2020, 37 % were working full-time from home (Yang et al., 2022). Those not directly involved in sampling and sequencing are particularly affected. Remote work does not only change the communication habits, but also uncouples the time windows in which work is done. Yang et al., 2022 found a decrease in synchronous communication and an increase in asynchronous communication. Without a compensation, it is more difficult to exchange information over the network. A database management system for experimental data could provide a minimal basis to accommodate this change.

While working out a database schema, it became evident that structuring the data and information would also enable new kinds of meta-analyses. The IGB has the unique possibility to sample the same sites over decades. It is of increasing importance that lab operators and researchers query past and new data sets without memorizing server locations and folder structures. Soon, the only way to manage massive amounts of data will be a data store that structures, relates, and minimizes data.

A normalized database schema tackles many of the herein described problems – the reduction of data redundancy and reflection of data relationships. It allows the formulation of constraints and enforces new data entries to comply with these rules. This chapter describes a schema for a relational database management system (RDBMS, or short DBMS) that allows consistent, non-redundant, and persistent data storage for metabarcoding experiments. The data and data relationships are accessible via virtualization and require just a user account and intranet access. An administrator assigns user roles that allow graduations in their rights to read or write to specific data partitions. Schema definitions are available on GitHub<sup>1</sup>.

### 4.2.1 Short- and Long-Term Staff

The particular situation for researchers at the institute's lab poses challenges on top of the organization of experimental data. Researchers like doctoral students, postdocs, guests, or interns stay relatively short compared to lab operators and department heads: typically three months to four years. These are the ones analyzing sequence data or building tools that improve the workflow in one way or another.

<sup>1</sup>[https://github.com/mariehoffmann/metabarcoding\\_database](https://github.com/mariehoffmann/metabarcoding_database)



Whereas department heads, group leaders, senior researchers, and lab operators stay for many years and even decades. They are so familiar with the lab organization that many details are self-evident and insufficiently communicated. On the other side, it is tedious for long-term staff to communicate repeatedly the same pieces of information.

The communication situation is complex through the geographical remoteness of the data analysts. They are located remotely and visit the lab facilities only on occasion. In a non-remote situation, open questions are often solved effortlessly during coffee breaks with colleagues or stand-up meetings. Instead, questions must be collected and communicated via emails, which may be overwhelming to the interviewee. From our own experience, we know that when multiple questions are posed in the same email, rarely all get answered, and often answers raise new questions. The whole interview process may take days and weeks. If parts of the *trivial facts* would be available in a consolidated, digitalized form, confusion and communication overhead is reduced drastically. The goal is not to avoid human-to-human interaction, but to reduce them for the sake of high-level questions.

Apart from the difficulty of knowledge transfer between the various staff groups, data, and knowledge itself is cluttered over servers, accounts, and colleagues. Many steps have to be taken until the actual research work can begin. A data analyst has to reconstruct sample origin, molecular procedures, and previous pipelines and their outcomes to create a starting point from which to improve the *status quo*. For example, the current system does not allow querying all experiments with some primer pair X. All these relationships have to be decrypted from file names, stored scripts, and by inquiring tabular sheets from a few technicians available. It is a fragile system with many points of failure, like erroneous data deletion, language barriers, and inconsistent data copies. There exists no digital description of how data sets are related. Concretely, I experienced the following workflow blocking situations:

- (i) *Important information was known only to one co-worker*
- (ii) *Redundantly stored data resulting in inconsistent data copies*
- (iii) *Information residing on personal accounts*
- (iv) *Software dependencies and updates*
- (v) *Unfamiliar terms and abbreviations*
- (vi) *Language barrier*

Environmental studies are so complex that they are build-up and improved iteratively. There exists no one-fits-all pipeline. The iterative approach relies on constructive feedback. It is challenging to establish an ongoing feedback loop because lab operators rely on the qualitative results of subsequently involved operators and analysts. Many analyses are performed by researchers staying as short as for a single experiment. They cannot compare two data sets with the same bioinformatics pipeline. From a laboratory operator's point of view, it is not transparent who has just joined the group and what their responsibilities are. This makes the communication barrier for lab operators higher than the other way around.

With the departure of researchers contracted on short term, insights and experience are lost, slowing the optimization and refinement of a metabarcoding analysis pipeline. Most often, the only work document is a scientific publication. The short stay of PhD students, trainees and postdocs will remain a fact, but a simplified access to information would facilitate new research.



### 4.2.2 Text Documents

Text documents are the primary communication tool that implies the copying, modification, and content divergence of documents. Synchronizing content that diverged is time-consuming, and most often omitted. As a result, researchers lose work.

With data cluttered over many types of documents in various formats (free text, tabular sheets, emails, etc.) also compatibility issues come into play. Some are commercial, exclude important operating systems like macOS or Linux, or are not backward compatible. For example, not all features of an MS Excel sheet are convertible into OpenOffice Calc<sup>2</sup>. The macOS text edit software Pages is not downward compatible, but a software update is often enforced<sup>3</sup>. With more data transferred into a single system that undergoes low-frequent update cycles and speaks a single language, incompatibilities are less frequent.

Entrants are confronted with plenty of abbreviations and technical terms that are specific in their semantics to the institute. For example, one of the river sampling sites *Spree Neu-Zittau* is abbreviated differently in protocols or label naming: *SNZ* and *NZ*. It is not evident that those are aliases for the same site. In another situation, a bioinformatician has to evaluate how many organisms were identified to species level. When given a list of hundreds of unknown organisms, it is infeasible to look them all up to gather taxonomic information. Instead, one is guided by binomial names<sup>4</sup>. However, some organism groups comprise multiple taxa and carry collective names written in the same format<sup>5</sup>. They are then easily counted as species falsifying any statistics based on species count. More researchers are to come and will need species lists augmented with taxonomic information. For example, it would save a lot of work and avoid misunderstandings if organism names and taxonomies were linked once.

### 4.2.3 Workflow and Goal

Figure 4.1 illustrates on a high level the steps where data is generated and further processed. A technician takes an esample from a sampling site and prepares it for storage and analysis. Samples can be analyzed by light microscopy or DNA metabarcoding. The results are forwarded to the study designer or data analyst. The analyst may request more documents or give feedback. Analysis results are consolidated as OTU resolutions, statistical tests, or scientific publications.

---

<sup>2</sup>Some macros and formulas need to be rewritten.

<sup>3</sup>see discussion thread here: <https://discussions.apple.com/thread/8627934>, accessed on 03.01.2022

<sup>4</sup>e.g., *Scenedesmus maximus*

<sup>5</sup>E.g., Cyclopoid nauplii

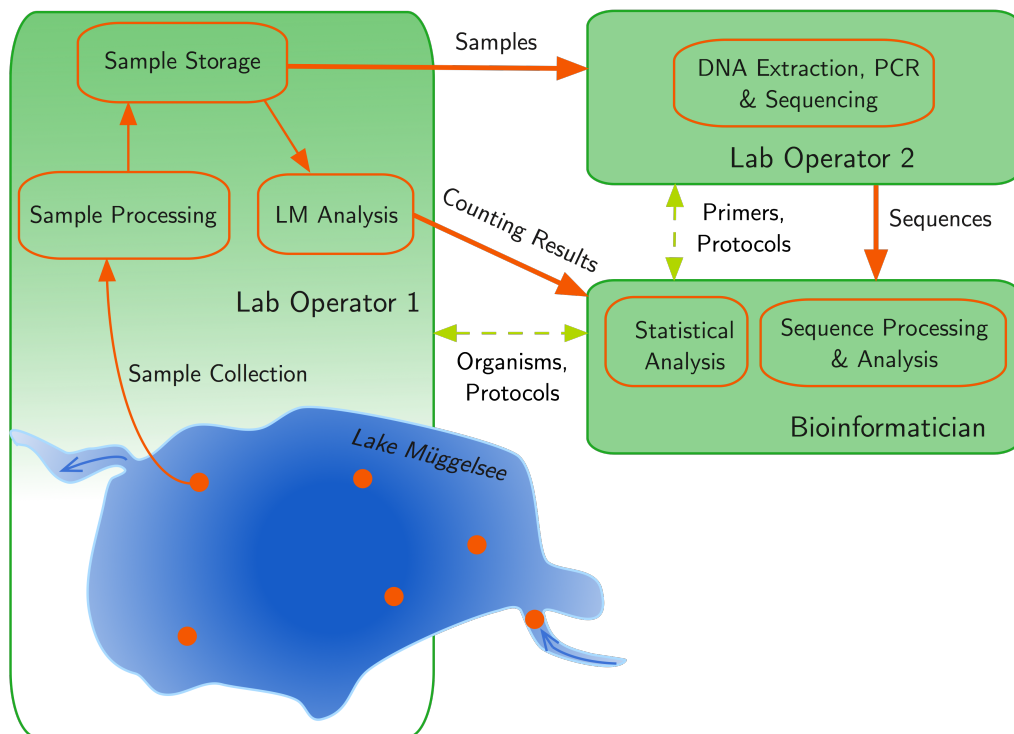


Figure 4.1: Data transfer between lab operators and bioinformaticians. Samples are collected from dedicated locations, processed, and prepared for storage by the first lab operator. Some samples undergo light-microscopic (LM) analysis, and the results are forwarded to the study conductor or bioinformatician. Some samples undergo PCR and sequencing conducted by the second lab operator. Sequences are sent to the bioinformatician. There is plenty of bidirectional exchange (light green arrows) about protocol details or biological questions on organism groups.

### 4.3 Aspects and Principles

In the next sections, we will describe how data and information are stored and communicated. We highlight those parts of the communication steps that we consider inefficient and error-prone and could profit from a data consolidation. The aspects we examined are *immediacy* of data, *consistency*, and *consolidation*. The immediacy of data transmission describes how many steps are necessary before information reaches the recipient. Data inconsistency occurs when data is replicated and changed, for example, by sending and changing result sheets at the same time. Write access to sensitive data should only be granted to a few maintainers to reduce the chance for erroneous modifications. Another aspect is the control over data changes, e.g. via *version* control.

Efficient retrieval of records is only possible if they are consolidated in the same format and managed by the same system. Furthermore, it should be impossible to store inconsistent data and it should be possible to correct human-made errors.

Principle	Example
Immediacy	Data should be communicated as directly as possible.
Consistency	Data should be self-consistent.
Consolidation	Data that is linked should be consolidated in the same format and management system.
Access and Versioning	Data and information should be modifyable by an exclusive group of people, and be revertible.

### 4.4 Entities and Workflow

We now describe in continuous text format the most relevant artifacts (entities) that are handled in the lab, their representation, and how they relate to each other: *samples*, *morphological* and *metabarcoding analyses*, and the *bioinformatics pipeline*. Entity properties relevant for documentation and analysis steps are displayed underlined. We also mark pieces of information that allow us to constrain their data type.

#### 4.4.1 Samples

Temporal patterns of species occurrence are recorded by sampling throughout the year. The frequency of sampling is adjusted to the seasonal turnover rates of the lake and river. At most one sample is taken per site and date. Freshwater sampling procedures are described in protocols intended for either phyto- or zooplankton. The set of sampling sites is fixed and can be described by their GPS coordinates. In protocols, the sample collectors use various names for the sampling sites. E.g., the site *Spree Neu-Zittau* is occasionally abbreviated with *SNZ* or simply *NZ*. Depending on what to catch primarily, different filter types are used – glass fiber (GF) or cellulose nitrate (CN). Glass fiber filters have a mesh width or pore size of 0.7  $\mu\text{m}$  and are used for catching phytoplankton. The mesh width is large enough to exclude most picoplankton. Cellulose nitrate filters have a pore size of 3  $\mu\text{m}$  and are used for catching zooplankton. This filter excludes smaller phytoplankter and larvae below 3  $\mu\text{m}$ ; larger plant parts are removed manually from the filter. The sample volume

is chosen based on the seasonal plankton density and the plankton type. Samples are labeled and stored until they undergo further analysis via light microscopy or metabarcoding.

#### 4.4.2 Morphological Analysis

Some samples are destined to be analyzed under a light microscope with different magnification factors by a trained plankton specialist. Phyto- and zooplankton are treated differently in terms of chemicals and subsampling for identification or counting, as described in more detail in Section 2.5.2. In a first screening all identifiable species or organism groups<sup>6</sup> are noted down. In a second step defined volumes are subsampled and the most abundant groups are counted in Utermöhl or Sedgewick-Rafter chambers. In addition, the biomass is estimated given the individual count and the geometry of an individual. The analyst notes the results in MS Excel sheets. MS Excel is used not only to store information, but also to visually organize data and create simple statistics, such as accumulation of counters. The analyst uploads the sheets to a server for long-term storage and sends copies or grants access to researchers who request them.

#### 4.4.3 Metabarcoding Analysis

Samples that undergo metabarcoding are cooled until DNA extraction. The DNA extracts then become labeled like MPS\_15\_9\_Phy\_300\_GF\_fd\_Q, which can be decoded into:

- Sampling site: Müggelsee MPS
- Sampling date: 15th of December
- Plankton type: Pythoplankton
- Volumina: 300 ml
- Filter type: glass fibre
- Drying method: freeze-drying
- DNA extraction kit: Qiagen (manufacturer)

We can recover the most relevant details from the label to relate sequenced DNA and the original freshwater sample. However, these details are incomplete: the year is missing, the mesh width of the glass fiber unknown, and Qiagen is a manufacturer of NGS-related products that produces many different types of extraction kits. DNA from phyto- and zooplankton samples are mixed equimolar and split depending on how many PCRs are scheduled (one for each primer pair). The technician adjusts the PCR cycler settings to the primer sequences. The PCR products are then sequenced and stored on a dedicated server under the project name in compressed FASTQ format. One specific lab member conducts the PCR who must not be identical with the sample collector. A PCR is usually completed within one day.

---

<sup>6</sup>which comprise multiple taxa

#### 4.4.4 Bioinformatics Pipeline

When being notified about sequenced samples, the first goal is to transform the raw reads into a few OTUs and possibly assign them to organisms or higher-order taxa. Typical read processing steps involve quality filtering guided by the quality scores given per base in the originating FASTQ file, merging of forward and reverse reads, denoising, trimming, and clustering based on a defined sequence similarity threshold. A cluster is represented by a common sense sequence (algorithm-dependent) and a read count size. OTUs can be resolved if there exist labeled references in the database that are sufficiently similar. For later inquiry, it is essential to associate an owner to the pipeline.

These first results are the foundation for further studies that, e.g., compare efficiency between different primers or identification methods (as shown in Section 2.5), reveal statistical correlation of sampling sites, or reveal seasonal patterns. These further findings may result in a publication.

### 4.5 Data and Knowledge Flow

#### 4.5.1 Organizational Structure

Directly involved in a study are three groups, each having their separate area of operation: lab operators for sample collection and morphological identification, lab operators for NGS methods, and researchers that conduct the study, process and evaluate the sequence data. The first two groups work directly at the institute near the sampling sites, whereas the last group works in part remotely. In practice, more people contribute to the forthcoming of an experiment: colleagues, including all types of students and guest researchers, and former members. In total, we have:

1. Group or department heads
2. Researchers, interns, students, visiting researchers
3. Lab Operators
4. Former members

The group and department heads oversee the long-term study goals, have a vast knowledge base covering all significant processing steps, and, most importantly, know how to redirect more detailed inquiries. Colleagues like students or postdocs are consulted regularly to advise on specific tools or common pitfalls. Lab operators are contacted mainly for protocol requests and sometimes ask for feedback like the quality of a sequence data set<sup>7</sup> or to confirm the presence of a particular organism. The most critical point is the necessity to contact members who are no longer part of the lab. In my experience, former members were usually contacted to ask for details about the processing of samples or sequences that were not adequately documented.

#### 4.5.2 Sample Data

The metadata (date, location, plankton type, and other) is encoded in a sample label and listed additionally in an MS Excel sheet (see Figure 4.2a) by the lab operator in charge of sample collection and preprocessing. They send the sheets upon request.

<sup>7</sup>as they try out different sampling processes

There are text documents (MS Word) describing detailed sampling protocols, tabular sheets (CSV or MS Excel) listing sample assignments, or primer sequences. For each piece of information, the sequence analyst has to maintain an additional document.

	A	B	C	D	E	F	G	H
1	DNA-Extraktionen Müggelsee/Spree 2014							
2								
3	a) Phytoplankton							
4	Nr.	Probenahmedatum	Probenahmestelle	Volumen (ml)	Konservierung	Homogenisation	Extraktion	Vol. Eluat(µl)
5	1	08/19/2014	Spree Grosse Tränke	50	GF/freezedrying	Tissuelyser/Metallball	Qiagen	70
6	2	08/19/2014	Spree Neu Zittau	50	GF/freezedrying	Tissuelyser/Metallball	Qiagen	70
7	3	08/25/2014	MüSee MPS	100	GF/freezedrying	Tissuelyser/Metallball	Qiagen	70
8	4	08/25/2014	MüSee MS3	100	GF/freezedrying	Tissuelyser/Metallball	Qiagen	70
9	5	09/01/2014	MüSee MPS	100	GF/freezedrying	Tissuelyser/Metallball	Qiagen	70
10	6	09/01/2014	MüSee MS3	100	GF/freezedrying	Tissuelyser/Metallball	Qiagen	70
11	7	09/02/2014	Spree Grosse Tränke	100	GF/freezedrying	Tissuelyser/Metallball	Qiagen	70
12	8	09/02/2014	Spree Neu Zittau	100	GF/freezedrying	Tissuelyser/Metallball	Qiagen	70
13	9	09/08/2014	MüSee MPO	200	GF/freezedrying	Tissuelyser/Metallball	Qiagen	70
14	10	09/08/2014	MüSee MPO	400	CN/freezedrying	Tissuelyser/Metallball	Qiagen	70
15	11	09/08/2014	MüSee MPU	200	GF/freezedrying	Tissuelyser/Metallball	Macherey-Nagel	30

(a) Sample metadata captured in an MS Excel sheet.

	A			B	C
1	Name file			Sample	Place
2	stability.trim.contigs.good.All_K_PI_rub_GF_fd_Q.short.fasta			Culture of P rubescens	
3	stability.trim.contigs.good.All_M_MPO_8_9_Phy_200_GF_fd_Q.short.fasta			Müggelsee	MPO
4	stability.trim.contigs.good.All_M_MPO_8_9_Phy_400_CN_fd_Q.short.fasta			Müggelsee	MPO
5	stability.trim.contigs.good.All_M_MPS_10_11_Phy_500_GF_fd_Q1_II.short.fasta			Müggelsee	MPS
6	stability.trim.contigs.good.All_M_MPS_10_11_Phy_500_GF_fd_Q1.short.fasta			Müggelsee	MPS
7	stability.trim.contigs.good.All_M_MPS_1_9_Phy_100_GF_fd_Q.short.fasta			Müggelsee	MPS
8	stability.trim.contigs.good.All_M_MPS_22_9_Phy_200_CN_fd_Q_1.short.fasta			Müggelsee	MPS
9	stability.trim.contigs.good.All_M_MPS_22_9_Phy_200_GF_fd_Q.short.fasta			Müggelsee	MPS

(b) Filenames of processed sequences encode metadata of the original sample.

Figure 4.2: Example of how sample metadata is documented and communicated. Notice the mixture of languages: the first document was created by a native German speaker and contains column names written in German, whereas a foreign researcher created the second document in English. The same researcher adopted the sample encodings for naming the FASTA files but is not forced to do so.

Since samples from different sites are not pooled for sequencing, their names are used as identifiers for the sequence data and the data intermediates (see Figure 4.2b). Other possibilities are to take the sample number listed in column A of Figure 4.2a or to introduce a new key. Note the repetitive column entries in the second document. The researcher has redundantly noted all the metadata of the samples – they are already included in the first file and its naming. Manual replication of metadata is an often observed pattern. It is time-consuming and error-prone, and should, therefore, be omitted. All text documents are permanent and are used by researchers for further studies. The files are neither read-only nor versioned. Later it will be impracticable to handle and remove ambiguities, if they are detected at all.

As shown in Figure 4.2a, the label gives us incomplete information (e.g., filter pore width unknown). The current state is that usually, only a single lab operator can complete the missing information. Based on personal experience, co-workers not directly involved in an analysis redirect to the colleague in charge. Hence, in case of further inquiry, one is dependent on the availability of the lab operator.

### 4.5.3 Morphological Analysis

Analysts store organism identification and counting results in MS Excel sheets. Results of multiple samples are grouped on the same sheet (see Figure 4.3). The graphical arrangement serves to orient the reader, but makes it impossible to rearrange the rows and columns for statistical analysis because they contain data from different experiments.



For each morphological experiment, the sample processor lists a sample name, metadata, lab operators' names, and finally, the organism groups and their sizes (individuals per liter), and biomasses. Percentages are listed in the last two columns (after the counting results), and below (not shown) as group accumulations. Most likely, all percentages and group totals were calculated manually because the cells did not contain formulas. There are at least two significant drawbacks:

1. Impossibility to detect wrong calculations, or copy-paste errors if not redone given the original document
2. Corrections or updates on counting data requires recalculation of all affected statistics (here: row-wise and group-wise percentages)

The first type of error often remains unnoticed; the second one allows easily introduction of new errors when not all affected statistics are updated. Whereas, a registered formula can be double-checked, and updates its associated tabular cell value automatically.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
1	Name	MPS	Nr.	Probenname	Index1	Index2		MS3		Nr.	Probenname	Index1	Index2			
2	Nr. Seq.	25/08/2014	3	Plankton_IGB_2014-3	704	507		25/08/2014		4	Plankton_IGB_2014-4	706	508			
3	Index		23	Plankton_IGB_2014-23	707	508		Bearbeiter:X			18s barcode2		16s			
4		Bearbeiter:XY/Z						Müsee MS3/ 25.08.2014/	Phytopl/ 100ml/ GF Filter/				freezedrying/ Extr. Qiag.Kit			
5		704	507	Phytopl/ 100ml/ GF Filter/	freezedrying/ Extr. Qiag.Kit											
6		707	508	Zoopl/10/ Ethanolfixed/ Extr. Qiag.Kit												
7				Vergleich Verhältnis Zooplanktonbiomasse klein												
8																
9	Zählung	Gruppe	Taxon	Ind/L	Biom(mg/m³)	%Ind	%Biom	Gruppe	Taxon	Ind/L	Biom(mg/m³)	%Ind	%Biom			
10		CYANOBACT	Chroococcus spec	7854	4.4783	0.06	0.16	CYANOBA	Aphanizomenon issatschenkii	137445	81.6984	0.47	1.07			
11		CYANOBACT	Microcystis spec	94248	92.9347	0.78	3.28	CYANOBA	Heterocyste api	19635	0	0.07	0.00			
12		CYANOBACT	Microcystis wesenbergii	39270	331.557	0.32	11.69	CYANOBA	Limnothrix spec	157080	33.0755	0.54	0.43			
13		CYANOBACT	Mikrocystis flos-aquae	19635	277.5826	0.16	9.78	CYANOBA	Planktothrix agardhii	49087	56.0594	0.17	0.73			
14		CYANOBACT	Pseudanabaena mucicola	883573	19.241	7.30	0.68	CYANOBA	Anabaena flos-aquae	39270	133.2397	0.13	1.74			
15		CYANOBACT	Cyanophyceae Zellen, spha	10426161	349.384	86.15	12.31	CYANOBA	Heterocyste anf	39270	0	0.13	0.00			
16		CHLOROPHY	Ankyra spec	39270	0.8032	0.32	0.03	CYANOBA	Cyanophyceae Zellen, spha	1521709	50.993	5.20	0.67			
17		CRYPTOPHY	Cryptomonas	30	31416	111.0331		CHLOROPH	Dicryosphae spec	49087	59.2176	0.17	0.77			
18		CRYPTOPHY	Cryptomonas	20	66759	69.9087		CHLOROPH	Chlorophyceae	5	834486	54.617				
19		CRYPTOPHY	Cryptomonas	35	7854	44.0791		CHLOROPH	Chlorophyceae	10	196350	102.8084	3.52	2.05		
20		CRYPTOPHY	Cryptomonas	25	58905	120.4786		CRYPTOPH	Cryptomonas	20	19635	20.5617				
21		CRYPTOPHY	Cryptomonas	15	11781	5.2047	1.46	12.36	CRYPTOPH	Cryptomonas	25	9817	20.0798			
22		CRYPTOPHY	Rhodomonas minuta/lacua	157080	7.7106	1.30	0.27	CRYPTOPH	Cryptomonas	15	19635	8.6745	0.17	0.64		
23		DIATOMS	Aulacoseira	5	3927	199.8595		CRYPTOPH	Rhodomonas minuta/lacua	706858	37.011	2.41	0.48			
24		DIATOMS	Aulacoseira	8	3927	167.7833	0.06	12.96	DINOPHY	Ceratium furcoides	3500	90.6675	0.01	1.17		
25		DIATOMS	Fragilaria crotonensis	133518	1024.3687	1.10	36.11	DIATOMS	Skeletonema spec	78540	42.6783	0.27	0.56			
26		DIATOMS	Nitzschia fonticola	117810	10.7796	0.97	0.38	DIATOMS	Aulacoseira	3	9817	2.7758				
27			TOT	CNT	12102988	2837.1877	100	100	DIATOMS	Aulacoseira	5	98175	277.5634	0.37	3.66	
28								DIATOMS	Asterionella formosa	39270	15.708	0.13	0.20			
29								DIATOMS	Fragilaria langustissii	510509	1989.464	1.74	26.09			
30								DIATOMS	Centrales d20	176715	536.0635					
31								DIATOMS	Centrales d25	29452	159.0317					
32								DIATOMS	Centrales d15	736311	780.7011					
33								DIATOMS	Centrales d10	2601632	744.7889					
34								DIATOMS	Centrales d05	21205750	2448.2788	84.51	60.91			
35									TOT	CNT	29289035	7755.757	100.00	100.00		

Figure 4.3: Working document for storing counting results. The sample names (highlighted yellow) contain the project name and a sample number, as listed in Figure 4.2a. Repeatedly, metadata of the samples are noted down. Below are the taxa and their counts per liter and biomass estimates. Percentages are computed manually per taxa and accumulated per group (not shown).

Documents, as shown in Figure 4.3 are sent without write protection to researchers upon their requests. The recipient either edits the received copy or copies content into a new sheet for further processing. Typically, more than one researcher holds a copy of the original sheets and processes it in his way. Regularly, typos are introduced (or corrected), leading once more to inconsistent copies. In the best case, work is done redundantly, but often it is infeasible to synchronize changes in the many copies leading to data inconsistencies.

#### 4.5.4 Metabarcoding Analysis

A workflow implication is that the lab operators are closer to the sample collectors than the analysts are. Lab operators and collectors have well-established workflows that should not be interfered with. The primary task of lab workers is to conduct the

PCR and provide the sequencing results. Some PCRs are non-standard like multiplex PCR where multiple primers run in a single PCR or the usage of biotinylated primers (like the forward primer of EUK14 used in the study of Section 2.5.2). Such information needs to be forwarded to the analysts, as it cannot be guessed given the raw data. Read data from experiments with biotinylated primers must be processed differently. The biotinylation information should therefore be stored together with the primer sequences themselves. PCR multiplexing is another variation of PCR experiments and should be associated with the experiment itself.

Two types of DNA extraction kits are currently in use: the NucleoSpln Plant II Extraction Kit from Macherey-Nagel GmbH & Co. KG, and the Qiagen DNeasy Plant Mini Kit. For this reason, the labels of the treated samples only bear the abbreviations MN for the first and Qiagen for the second kit. However, this is only obvious to initiated personnel. Detailed kit information is often requested, e.g. when a researcher wants to compare or publish a study.

#### 4.5.5 Bioinformatics Pipeline

There are a variety of tools for each single processing step in a bioinformatics pipeline. Indeed, one of the most time-consuming tasks is the translation between the different input and output formats.

Many researchers working in ecology are somewhat self-taught programmers. It is not advisable to fix a scripting language or specific toolchain for the sake of reproducibility. On the contrary, it is beneficial for researchers to start with tools they are familiar with to reach a *productive state* faster. This adds uniqueness to each analysis. In addition, there are multiple versions of each tool and a large number of possible parameter combinations, noteworthy to be documented for the purpose of reproduction.

#### 4.5.6 Additional Requirements

We can solve most of the herein described conflicts by a consolidated data scheme that enforces constraints where it is meaningful and beneficial. Apart from the data availability, we would like to devise queries over data portions that are inherently related to each other. Therefore, we need an intuitive and simple model and a query language that is easy to learn.

To maintain privacy and consistency, we would like to provide users with tiered permissions. For example, an intern does not need to have write access if their task is to analyze historical data. In the event that a supervisor with write privileges makes a mistake, it would be helpful to roll back the database to a previous checkpoint. The database should also be accessible online and connected to the IT infrastructure. An obvious solution is to install a relational database management system that provides all the means to formulate a logical model, grows automatically as more data is entered, and processes queries asynchronously.

#### 4.5.7 Why a Relational Database System?

For decades *relational database management systems* (RDBMS) had been the working horses for data-intensive applications. RDBMS undergo fewer release cycles and are one of the most sophisticated software systems available. A key feature is its *declarative* query language SQL, which follows the structure of relational algebra closely, and has an English-sentence style. In a *declarative* language, we specify the pattern that



a piece of data has to satisfy<sup>8</sup>. In contrast, an *imperative* language describes *how* to compute the result. The *declarative* property of SQL makes it independent of physical schema and automatic query automation; implementation details of the database engine and optimizer remain hidden.

Besides data consolidation, a DBMS ensures data consistency by making it impossible to insert mistyped or incomplete entries. The key is to capture only relevant information and eliminate redundancy by moving attributes with small and limited domain sizes into separate tables. This process is called *normalization* (Codd, 1970).

Since we expect the number of queries and insertion requests to be low over time, a solid-drive stored relational database is sufficient. Figure 4.4 relates *data temperature* and media for data storage. Only the top-tier online services that receive thousands of requests per second have to move to in-memory databases or distributed key-value stores.

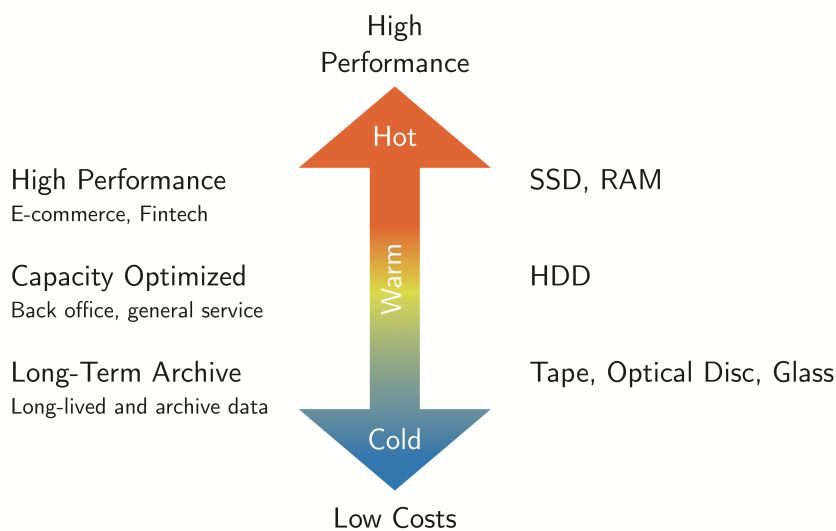


Figure 4.4: Data temperature and storage media. Storage medium costs are traded off against low latency. High-performance applications require data to be stored on low-latent, but costly media. Data that is accessed infrequently is stored on less expensive hard disk drives (HDD), optical media, magnetic tapes, or glass libraries. The expected number of I/O operations, the amount of data and the location of the users determine which medium is optimal.

The internal organization of a database management system is shown in Figure 4.5. There are two types of clients: remote and self-service. A remote client accesses the database through the server process. Whereas a self-service client runs on the same machine as the master database process (broker) and accesses the database directly through shared memory. The broker coordinates all the requests, manages the shared memory, and blocks the database. The shared memory is used for physically storing process handles, locks, and other data structures needed for interprocess communication. The actual transaction is performed by background processes (writer, reader, cleaning processes) with access to the physical database.

A relational database organizes its data into *relations* (also called *tables*). A table reflects the abstract concept of an entity, a location, or primer pair, for example, that exists independently. Columns of a table correspond to attributes of an entity. A relation is an unordered set of *tuples* (also called *rows*). The relational model for

<sup>8</sup>and possible data transformation like sorting or agglomeration

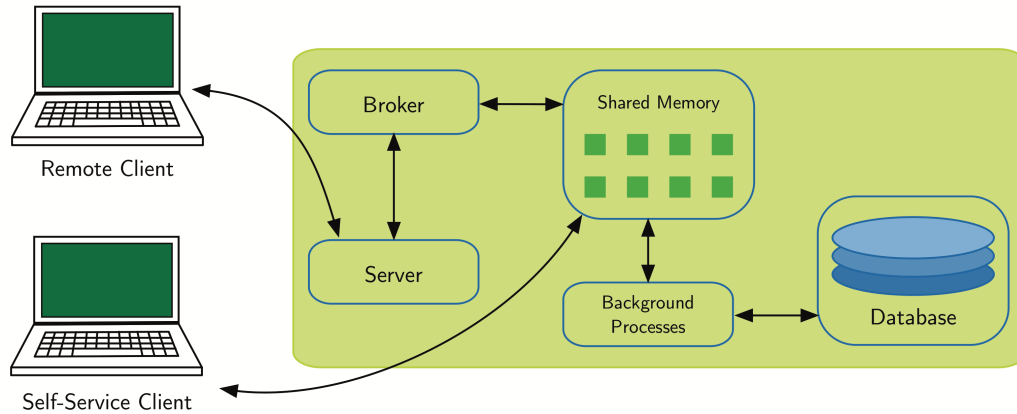


Figure 4.5: Components of a full RDBMS. A full database management system is a multi-threaded management system with at least one server process for managing access, and a client software to allow users to sign in for access.

database management was first described by Codd, 1970. Rapid implementations came already into existence in the early 1980s.

Furthermore, each row must be uniquely addressable by declaring a column or set of columns as the *primary key* (PK); alternatively, we can introduce an enumerator to generate a series of numerical identifiers. The value or value combination of the key column(s) forms a unique identifier. It is crucial to link rows of different tables. For faster row access, an index is created on primary keys if not defined otherwise.

A relational database would use SQL as a primary programming language. The command set is divided into *data definition* (DDL), *data control* (DCL), *data manipulation* (DML), and *data querying* (DQL) languages matching the different roles that users can be assigned to (see Table 4.1).

Syntax Set	Description	Commands	Role
DCL	Allow users to perform specific tasks or remove their access	GRANT REVOKE	Admin A
DDL	Create, delete, and alter tables; define column constraints and trigger rules	CREATE DROP ALTER	Admin B
DML	Insert, modify or delete rows of a table without altering the schema	INSERT UPDATE DELETE	User A
DQL	Query and analyse stored data	SELECT	User B

Table 4.1: Categories of SQL commands and associated user roles.

The purpose of roles is to restrict access for specific groups of users in order to maintain integrity or hide sensitive or irrelevant data. It is usually an administrator (Admin A) who is responsible for granting or revoking access to users. Changes that affect the schema structure should only be made by users (Admin B) who understand the dependencies between tables or functions, as the privileges of this user group allow the database to be left in an inconsistent state. Insertion rights may be granted to collaborators generating data such as protocol results, a new primer pair for a

PCR experiment, or renaming of species due to changes in official nomenclature. Researchers who work on a bioinformatics pipeline or statistical analysis may not produce data to be captured in the database but rely on detailed information about previous experiments. The last user group would primarily need read access to tables storing past experiments and be able to link information from different tables to pose complex queries.

In contrast to other programming languages, SQL has various dialects like MySQL (Widenius, Axmark, and DuBois, 2002), PostgreSQL (Stonebraker and Rowe, 1986), or DB2 (Chamberlin, 1998). As RDBMs had originally been intended for consistency and long-term storage, update cycles are in the range of years and never enforced.

For every read or write transaction on a database, the DBMS guarantees the *ACID* properties, which are *atomicity*, *consistency*, *isolation*, and *durability* (Gray, 1981). A command set to be executed on the database is grouped into atomic *transactions*. The reason for this is that the interweaving of concurrent processes could otherwise lead to data inconsistencies.

*Atomicity* guarantees that all commands within the same transaction are executed as if they were one. Either all commands or none gets executed. *Consistency* describes that the data constraints are respected and that it is impossible to enter data that violates these rules. The *isolation* property guarantees that concurrently executed transactions leave the database in the same state as if they were executed sequentially. The *durability* property guarantees that each successfully executed transaction remains once and forever executed even in case of a blackout. The last property favors non-volatile storage media. Conclusively, DBMS, which are not explicitly in-memory, have relatively slow response times. For e-commerce or social media applications, this may be prohibitive, but for the lab described here, we want consistency and durability above all else.

## 4.6 Database Schema

When the process of deciding what to store is complete, the database schema is *normalized*. The goal is to reduce redundant attributes and ensure data integrity where appropriate. For example, assume metabarcoding experiments have three properties we would like to store: primer pair, experiment, and date. We could store them in a single table with three columns, and use a combination of primer and experiment name as a unique identifier. The state shown in Table 4.2 would be legal: the composite keys (DIV4, exp1), (EUK15, exp1), (EUK-15, exp1) are unique. One of the primer names is mistyped, or has an alias name:

Primer_Name (PK)	Experiment_Name (PK)	Datetime
DIV4	exp1	2020-01-09 12:11:04
EUK15	exp1	2020-01-09 12:51:42
EUK-15	exp1	2020-01-09 12:51:42

Table 4.2: Unnormalized experiment table with primer and experiment names as a composite primary key.

By putting primer pairs in their own table and define a foreign key constraint on the primer name (syntax shown later), we can avoid the inconsistent state shown in Table 4.2. Trying to insert the tuple (EUK-15, exp1, '2020-01-09 12:51:42') into

the table would fail as EUK-15 is not known as a primary key in the table `Primer` (see Table 4.3).

Primer_Name (PK)	Forward_Sequence	Reverse_Sequence
DIV4	GCGGTAATTCCAGCTCCAATAG	CTCTGACAATGGAATACGAATA
EUK15	CCAGCASCYGGGTAATTCC	ACTTTCGTTCTTGATYRA

Table 4.3: A table storing primer sequences.

The challenge is to balance constraints and degrees of freedom. Values that are important for linking data across multiple tables (thus allowing more complex queries) should be restricted. Restricting free-text fields that contain human-readable information, on the other hand, is not useful. Columns (or combinations of columns) that are highly redundant or require naming enforcement could potentially be normalized. The following entities should go in separate tables:

- `Experiment`
- `PCR experiments` as a specialization of `Experiment` and with redundant attributes in their own tables:
  - `Primer Pair` for storing name, sequences and reference publication
  - `Sequences` for storing meta data like server address of the sequenced amplicons
  - `DNA Extraction Kit` to store the manufacturer and product version of the kit
- `Morphological Experiments` as a specialization of `Experiments`
  - `Organism Group` to capture polyphyletic groups corresponding to a set of taxa
- `Samples` with redundant features in their own tables:
  - `Location` to name the fixed set of sampling sites and add geographic information
  - `Dry Method` to name one of the few methods (freeze-drying, oven-drying, bluegel, air-drying)
  - `Filter Method` to describe the filters applied during sampling
  - `Plankton Type` to name the major group that was caught
- `Pipeline` which processes a sequence data set
- `Publication` which refers to an experiment analysis
- `OTU` as a result of a pipeline
- `Taxon` as a possible assignment to an OTU with redundant attributes in their own columns:
  - `Rank name`
  - `Taxon Names` to store synonyms
  - `Lineage` from associated taxon to root node of the registered taxonomy

- Taxonomy Source for naming the origin (e.g. NCBI GenBank) and give an URL

In the center of the schema, we have the `Experiment` table. An experiment can be a morphological or a PCR experiment. The results of a morphological experiment are expressed as a list of taxa of all organisms seen (without counts). Optionally, there is a census table that contains individual counts and body mass estimates for indicator groups. Some organism groups are categorized into size groups. The introduced table `organism_group_count` captures this additional information and is tied to a one-to-one correspondence with an already registered organism group.

Within an experiment, at least one environmental sample is processed. The composition (primarily phyto- or zooplankton) is controlled via the filter type and its mesh width (stored in table `Filter Method`). The sample is taken at a special sampling point assigned to a stratum and a geographic location (table `Location`). It is then conserved for later processing (table `Dry Method`). When the sample is subjected to PCR analysis, it can be divided into subsamples and treated with different sets of primers. There are relatively few primer pairs, so extraction into a separate table is useful. The sequenced PCR products are output as compressed FASTQ files. These files are stored on an external server for the long term. The location (server address) and sequence metadata are noted in the `Sequences` table, but not the raw sequence files, since these contain highly redundant reads and are several gigabytes in size.

In contrast, it would be advantageous to store the instructions, i.e., the processing steps and utilities of the bioinformatics pipeline in an additional table (`Pipeline`). A pipeline can be stored as a free text description, or if serializable, in a serialized format.

One of the final steps in a bioinformatics pipeline is to cluster reads to form OTUs which hypothetically have a corresponding taxon. OTUs are described by at least one representative sequence and a read count. The most likely taxon can be determined using BLAST searches and tools that estimate the lowest common ancestor in cases of ambiguous matches. A taxon (table `Taxon`) has a unique scientific name and aliases, a taxonomic rank, a lineage, and metadata about the origin of the taxonomy<sup>9</sup>.

#### 4.6.1 Table Definitions and Relations

The following subsections detail the properties of the tables shown in the overview Figure 4.6 and provide PostgreSQL table definitions. We define the tables and functions as part of a schema that we will call `lake_monitor`. In a relational database, each row of a table must be uniquely identifiable, that is, each row of a table has an attribute or combination of attributes that is unique. We declare a column or a combination of columns as a key by adding the constraint `PRIMARY KEY (PK)`. By default, indices are built on primary keys.

When a column should have unique values, but not necessarily serve as an identifier, we use the `UNIQUE` constraint (U). It is not recommended to use an attribute of the `VARCHAR` type – a character sequence of variable length – as a primary key. The reason for this is that the `VARCHAR` type is used to add information that is significant outside the database system. This type of attribute tends to be changed more frequently in the future, leading to updates in all copies and references. Such actions cause write overhead and carry the risk of inconsistencies (see p. 33 in Kleppmann, 2017).

<sup>9</sup>For example, the taxonomies provided by GenBank and Silva are not identical.

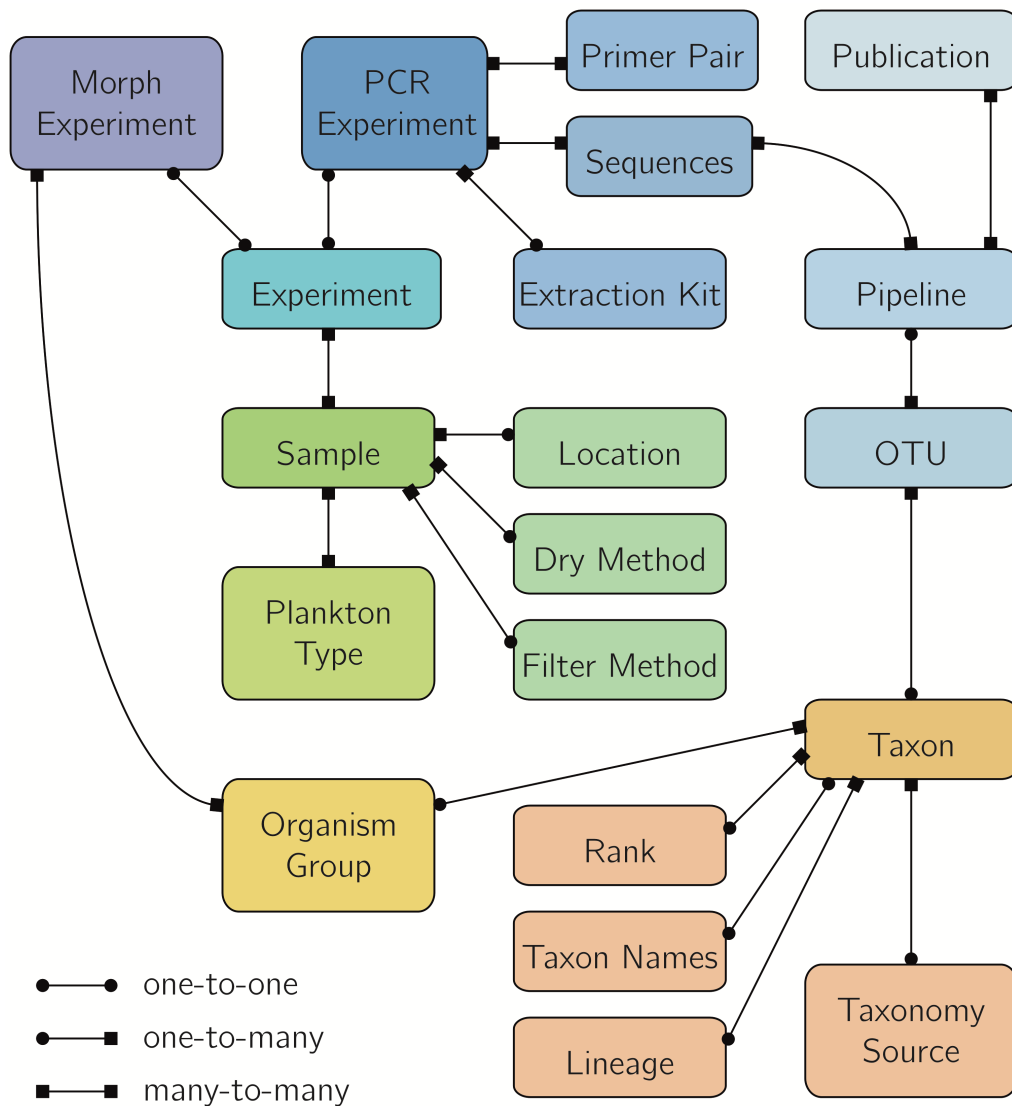


Figure 4.6: Database schema for storing and relating experiments on plankton samples. Fields about small domains such as sample location or drying method are part of the sample metadata, but are stored in separate tables to avoid redundancy and allow constraints to be defined. For example, a new sample cannot be registered that is associated with a non-existent location.

Also, the VARCHAR data type allows spaces, which can be crucial for some indexing algorithms. Instead of using an attribute VARCHAR as a primary key, we add an identifier of type SERIAL and add uniqueness constraints to the column VARCHAR.

Columns that refer to primary keys of other tables will have a FOREIGN KEY constraint that forces an inserted identifier to refer to an existing entry. Another useful PostgreSQL constraint is NOT NULL, which specifies that the field must not be empty when inserted. All data types and constraints applied to the lake database schema are listed in Table 4.4 and Table 4.5, respectively.

Type	Range/Size	Description
INTEGER	$[-2^{31} : 2^{31} - 1]$	Integer stored in four bytes Variants: SMALLINT (two bytes), BIGINT (eight bytes)
SERIAL	$[1 : 2^{31} - 1]$	Autoincrementing integer for primary key generation Variants: SMALLSERIAL (two bytes), BIGSERIAL (eight bytes)
REAL	$[-2^{31} : 2^{31} - 1]$	Floating point stored in four bytes Variable precision
POINT	(REAL, REAL)	Geometric type storing two floating points
CHAR(k)	up to 1 GB	Fixed-length character string Trailing and unused positions are padded
VARCHAR	up to 1 GB	Variable-length character string, no padding Optional check for length restriction
TEXT	up to 1 GB	Variable-length character string No length restriction check
DATE	[4713 BC : 5874897 AD]	Stores date in eight bytes Formattable into, e.g., dd/mm/yyyy Resolution: 1 day
TIMESTAMP	[4713 BC : 294276 AD]	Stores date and time in four bytes Formattable into yyyy-mm-dd hh:mm:ss Resolution: 1 $\mu$ s

Table 4.4: PostgreSQL datatypes used in the schema definition.

Command	Description
PRIMARY KEY	Unique record identifier not null
FOREIGN KEY	Value matches primary key of record in another table
UNIQUE	Value must be unique
NOT NULL	Constrain record value to be not empty

Table 4.5: PostgreSQL constraints used in the schema definition.

The subsequent sections address table details of the schema shown in Figure 4.6. Tables or entities are depicted as rounded boxes, and edges between them represent a relation. Concretely, a row of table PCR Experiment can be related to a row in table Primer Pair by having an extra field in PCR Experiment that stores the identifier of a specific primer pair. The relationships are specified into:

- (i) one-to-one
- (ii) one-to-many
- (iii) many-to-many



In a *one-to-one* relationship between two tables A, B, a row of table A refers to at most one row of table B. As PCR experiments are a specialization of an experiment, their relationship is one-to-one. In such a case, we can transfer the primary key of Experiment to PCR\_Experiment. In a *one-to-many* relationship, a single row of table A can be associated with many rows in table B. For example, a single OTU is resolved to at most one taxon, but the same taxon can be assigned to many OTUs. The foreign key has to be stored in table OTU to avoid redundancy. In a *many-to-many* relationship, a single row of table A can relate to multiple rows in table B and *vice versa*. For example, in an experiment, one or more samples are processed. On the other hand, a single sample may be assayed for more than one experiment. In this case, we can put the foreign key in either table.

We initialize a database schema as shown in Listing 4.1. As a name we choose lake\_monitor. Multiple schemas can live on the same database server. We address a table by first calling the schema name followed by a dot and the table name.

Listing 4.1: Define a database schema named 'lake\_monitor'.

```
1 CREATE DATABASE lake_monitor;
```

### Table Sample

Experiments process environmental samples. These samples are collected on a specific date and location. Since sampling locations are fixed to a few and should be supplemented with additional information such as GPS coordinates and a location description, we store only a reference to the location in the table. We require that its identifier refers to a location already registered in a special location table. When samples undergo PCR treatment, they get labeled and listed with a generic name: Plankton\_IGB\_<year>-<no>, e.g., Plankton\_IGB\_2014-14, which states that the sample was collected in 2014 and is the 14th in a series of PCR treated samples. The actual sampling location, precise date, volumes, PCR extraction kit, and others, are listed in the working sheet. These identifiers are not sufficient to serve as the primary key for all samples, regardless of their subsequent treatment. Therefore, we choose to auto-generate a primary key (Sample.id) but enforce the uniqueness of sample names that can follow the previously described format.

A plankton sample is the solid parts that remain after filtering a specific water volume with dedicated filters. We link this information by storing only the references to entries in specific tables for filtering methods, drying methods, and plankton species. In contrast, sampled volumes and collection dates are stored directly in the table.

Sample		
PK	id	SERIAL
U	sample_name	VARCHAR
FK	location_id	
FK	filter_method	
FK	dry_method	
FK	plankton_type	
	volume	REAL
	collection_date	DATE



In PostgreSQL, we can auto-generate the primary keys of an integer type by declaring a column type as SERIAL. A SERIAL is stored in four bytes and can represent identifiers between 1 and  $2^{31} - 1$ . In case a much smaller range is sufficient or a larger range required, PostgreSQL offers SMALLSERIAL (two bytes) or BIGSERIAL (eight bytes). Since the Sample table is the first one to be created, we can use the foreign key constraints earliest after the referred tables have been created as well. Concretely, we have to postpone the foreign key declarations for location\_id, filter\_method, dry\_method, and plankton\_type. However, it is mandatory to specify a column type for foreign keys as well, which must be identical to the type of the primary key they refer to. In the case of SERIALs, we have to depict the alias INTEGER type.

Listing 4.2: Define table Sample.

```

1 CREATE TABLE Sample(
2     id SERIAL PRIMARY KEY,
3     sample_name VARCHAR NOT NULL,
4     location_id CHAR(3) NOT NULL,
5     filter_method SMALLINT NOT NULL,
6     dry_method SMALLINT NOT NULL,
7     plankton_type SMALLINT NOT NULL,
8     volume REAL,
9     collection_date DATE NOT NULL
10 );

```

### Table Location

Samples are taken from the River Spree at Große Tränke (SGT), Neu-Zittau (SNZ), and at several sites of the Lake Müggelsee (MPS, MS3, MPO, and MPU). For all sites, there exists a three-letter abbreviation, which we will use as an identifier. There are more abbreviations: for example, the sampling site in Spree Neu-Zittau may be referred to as Spree NZ, SNZ, or NZ. The aliases column captures the synonyms. Each site has known GPS coordinates and prescribes seasonal sampling of specific strata. The description field allows a free text description of how the freshwater sampling will be approached and performed.

Location		
PK	short_name	CHAR(3)
U, NN	long_name	VARCHAR
	aliases	VARCHAR[]
NN	gps_coordinates	POINT
	stratification_layer	VARCHAR
	description	TEXT

The Location table has no columns referring to other tables but is used by the above-defined Sample table. We can now add the foreign key constraint on the location\_id of a sample to refer to the primary key of a location.

Listing 4.3: Define a Location table and add foreign key constraint on Sample.location\_id.

```

1 CREATE TABLE Location(
2     short_name CHAR(3) PRIMARY KEY,

```

```

3     long_name VARCHAR UNIQUE NOT NULL,
4     aliases VARCHAR[],
5     gps_coordinates POINT NOT NULL,
6     stratification_layer VARCHAR,
7     description TEXT
8 );
9
10 ALTER TABLE Sample ADD FOREIGN KEY (location_id)
11 REFERENCES Location (short_name);

```

### Table Plankton Type

Standard abbreviations in result sheets or sample labels for the two types of plankton are phyto, and zoo, respectively. We enforce unique naming by adding the UNIQUE constraint.

Plankton_Type		
PK	id	SMALLSERIAL
U, NN	short_name	CHAR(10)
	designation	VARCHAR

Listing 4.4: Define table Plankton\_Type and add foreign key constraint on Sample.plankton\_type.

```

1 CREATE TABLE Plankton_Type(
2     id SMALLSERIAL PRIMARY KEY,
3     short_name CHAR(20) UNIQUE NOT NULL,
4     designation VARCHAR
5 );
6
7 ALTER TABLE Sample ADD FOREIGN KEY (plankton_type)
8 REFERENCES Plankton_type (id);

```

### Table Filter Method

For the mechanical separation of phyto- and zooplankton we store a filter method name, the mesh width, and a free text field for a more detailed description. The description field allows to store information about the material or the manufacturer, which is needed for study reports.

Filter_Method		
PK	id	SMALLSERIAL
U, NN	name	VARCHAR
NN	mesh_width	REAL
	description	TEXT

After defining the Filter\_Method table we can add a foreign key constraint for the filter method column of the Sample table.

Listing 4.5: Define table `Filter_Method` and add foreign key constraint on `Sample.filter_method`.

```

1 CREATE TABLE Filter_Method(
2     id SMALLSERIAL PRIMARY KEY,
3     name VARCHAR UNIQUE NOT NULL,
4     mesh_width REAL NOT NULL,
5     description TEXT
6 );
7
8 ALTER TABLE Sample ADD FOREIGN KEY (filter_method)
9     REFERENCES Filter_Method (id);

```

### Table Dry Method

Several drying or preservation methods are being tested by laboratory operators. Using feedback from PCR analysis, they can evaluate the relative efficiency of a drying method. The goal is to obtain as much DNA as possible, resulting in a larger number of PCR reads. We use an automatically generated identifier of type `SMALLSERIAL` since there are only a handful of preservation methods, a short name common on sample labels, and a free text field to describe the drying method in more detail. The most common method to date is freeze-drying for phytoplankton and ethanol fixation for zooplankton. Other tested methods are air drying, blue gel drying, or oven drying. Other parameters that can be varied are the temperature and the duration.

Dry_Method		
PK	id	SMALLSERIAL
U	short_name	CHAR(10)
	temperature	REAL
	duration_hours	REAL
NN	description	TEXT

Since some methods are performed at room temperature, the temperature and duration fields may be left blank for new entries. We will also add the last foreign key constraint on the `Sample` table.

Listing 4.6: Define table `Dry_Method` for sample conservation methods and add foreign key constraint on `Sample.dry_method`.

```

1 CREATE TABLE Dry_Method (
2     id SMALLSERIAL PRIMARY KEY,
3     short_name CHAR(10) UNIQUE NOT NULL,
4     temperature REAL,
5     duration_hours REAL,
6     description TEXT NOT NULL
7 );
8
9 ALTER TABLE Sample ADD FOREIGN KEY (dry_method) REFERENCES Dry_Method (id);

```

### Table Experiment

Two types of analyses can be conducted on a set of plankton samples: identification and counting based on morphological features under the light microscope, or DNA

extraction for a metabarcoding experiment. Both have common features but later differ in their protocols and types of results. Thinking object-oriented, the table `experiment` concentrates common features, and the morphological and metabarcoding experiments are specializations of a generic experiment. Common to all experiments is the processing of a set of samples, a start date, a principal investigator (or collaborator), and an optional description of the objective.

Note that an array data type (`INTEGER[]`) is used here for storing the associated samples, which cannot be augmented with a foreign key constraint, so referential integrity is not automatically maintained. There are two ways to implement a foreign key constraint for a list: either by normalizing, i.e. using only the primitive data type `INTEGER` and inserting as many rows as there are different samples, or by adding a function that is triggered when new experiments are inserted. This trigger checks whether each identifier listed in the `sample_ids` field matches an existing primary key in the `sample` table. Upon success, the insertion is completed, otherwise aborted (see Appendix C.1). Since 2012 there exists a patch that adds syntax to declare foreign key constraints on array datatypes<sup>10</sup>, but its acceptance for PostgreSQL (version 11) is pending.

Experiment		
PK	id	SERIAL
NN	sample_ids	INTEGER[]
NN	date	TIMESTAMP
NN	lab_staff	VARCHAR
NN	description	TEXT

Listing 4.7: Define table `Experiment` for general experiments. The trigger on `sample_ids` is listed in Appendix C.1

```

1 CREATE TABLE Experiment (
2   id SERIAL PRIMARY KEY,
3   sample_ids INTEGER[] NOT NULL,
4   date TIMESTAMP NOT NULL,
5   lab_staff VARCHAR NOT NULL,
6   description TEXT NOT NULL
7 );

```

### Table Morphological Experiment

One specialization of an experiment is the morphological experiment, in which species are identified under the microscope, counted in Utermöhl or Sedgewick-Rafter chambers, and biomass is estimated for some common groups of organisms. Because abundance and biomass estimates are not made for all organism groups, the table definition does not enforce a `NOT NULL` constraint.

<sup>10</sup><https://commitfest.postgresql.org/17/1252/>

<b>Morph_Experiment</b>			
<b>PK, FK</b>	experiment_id		
<b>FK, NN</b>	organism_group_id		
	individuals_liter		INTEGER
	biomass_mg_m3		REAL
	size_category		REAL

Listing 4.8: Define table Morph\_Experiment for morphological experiments as a specialization of Experiment and reference its primary key to Experiment.

```

1 CREATE TABLE Morph_Experiment(
2   experiment_id INTEGER PRIMARY KEY,
3   organism_group_id INTEGER NOT NULL,
4   individuals_liter INTEGER,
5   biomass_mg_m3 REAL,
6   size_category VARCHAR
7 );
8
9 ALTER TABLE Morph_Experiment ADD FOREIGN KEY (experiment_id)
10  REFERENCES Experiment (id);

```

### Table PCR Experiment

The other type of experiment is a metabarcoding experiment with all the properties listed in Experiment and data about the PCR: primer pairs, DNA extraction kit, and optional indices. As only a handful of different primer pairs and extraction kits are in use, we extract these into dedicated tables and relate here only their keys.

<b>PCR_Experiment</b>			
<b>PK, FK</b>	experiment_id		
<b>FK, NN</b>	primer_pair		
<b>FK, NN</b>	extraction_kit		
	description		TEXT

Listing 4.9: Define table PCR\_Experiment for metabarcoding experiments and reference its primary key to Experiment.

```

1 CREATE TABLE PCR_Experiment(
2   experiment_id INTEGER PRIMARY KEY,
3   primer_pair SMALLINT NOT NULL,
4   extraction_kit SMALLINT NOT NULL,
5   description TEXT
6 );
7
8 ALTER TABLE PCR_Experiment ADD FOREIGN KEY (experiment_id)
9  REFERENCES Experiment (id);

```

### Table Extraction Kit

The DNA extraction kit is another element that affects the overall efficiency of PCR and must be mentioned in linked publications in the methods section. There are mainly two manufacturers in use: the NucleoSpin Plant II extraction kit and one from Qiagen. The extraction kit names are frequently abbreviated for sample labels with MN or Qiag. We use these as identifiers instead of auto-generating keys. The full name of the extraction kit, as specified by the manufacturer, can be noted in the designation field.

Extraction_Kit		
PK	id	SMALLSERIAL
U, NN	short_name	CHAR(10)
NN	manufacturer	VARCHAR
	designation	VARCHAR

Listing 4.10: Define table `Extraction_Kit` for storing DNA extraction kits and add foreign key constraint to `PCR_Experiment`.

```

1 CREATE TABLE Extraction_Kit(
2   id SMALLSERIAL PRIMARY KEY,
3   short_name CHAR(10) UNIQUE NOT NULL,
4   manufacturer VARCHAR NOT NULL,
5   designation VARCHAR
6 );
7
8 ALTER TABLE PCR_Experiment ADD FOREIGN KEY (extraction_kit)
9   REFERENCES Extraction_Kit(id);

```

### Table Primer Pair

A primer pair is unique by its combination of forward and reverse primer sequences. Often there are no consistent designations across multiple publications. Here we allow to define a ten-character identifier as the name for the pair and to require names for the forward and backward sequences. In the metabarcoding study presented in Section 2.5, we used EUK15 to address the pair, but its constituting sequences are TAREuk454FWD1 for the forward primer, and TAREukREV3 for the reverse primer as named in the original paper by Stoeck et al., 2010. When combining the forward primer with another reverse primer for a study, we force the registration of a new pair.

Usually, primers are either designed and experimentally evaluated for a specific group of organisms, such as the DIV4 primer used in the study (section 2.5), or they are universally applicable, i.e. they cover a broad range of phylogenetically diverse groups. The target group and the region are therefore known in advance and must be specified for new entries. Whereas, the product length may be unknown due to a lack of referential data or vary a lot due to the universality of the primer pair. The reference field allows the specification of the DOI of the originating publication, and a free text field (`description`) further remarks.

Primer_Pair		
PK	id	SMALLSERIAL
U	pair_name	CHAR(20)
NN	name_fwd	VARCHAR
NN	name_rev	VARCHAR
NN	sequence_fwd	CHAR(40)
NN	sequence_rev	CHAR(40)
NN	target_group	VARCHAR
NN	target_region	VARCHAR
	product_length	INTEGER
	reference_doi	VARCHAR
	description	TEXT

A single primer sequence rarely exceeds 25 base pairs – we can restrict the sequence type to hold at most 32 characters. If known, we allow the addition of target groups (e.g., diatoms) or target regions (e.g., 18S rRNA), and an expected PCR product length. This data comes from related publications or previous studies and is extremely useful when evaluating the PCR efficiency and building the bioinformatics pipeline.

Listing 4.11: Define table `Primer_Pair` for primer pairs and add foreign key constraint to `Experiment`.

```

1 CREATE TABLE Primer_Pair(
2   id SMALLSERIAL PRIMARY KEY,
3   name CHAR(20) UNIQUE,
4   name_fwd VARCHAR NOT NULL,
5   name_rev VARCHAR NOT NULL,
6   sequence_fwd CHAR(40) NOT NULL,
7   sequence_rev CHAR(40) NOT NULL,
8   target_group VARCHAR NOT NULL,
9   target_region VARCHAR NOT NULL,
10  product_length INTEGER,
11  reference_doi VARCHAR,
12  description TEXT
13 );
14
15 ALTER TABLE PCR_Experiment ADD FOREIGN KEY (primer_pair)
16 REFERENCES Primer_Pair (id);

```

### Table Sequences

Storing raw DNA sequences as output by the sequencer machine in a database schema would be wasteful as they are accessed only 1-2 times in their lifetimes. They are large (several gigabytes), compressed, and contain highly redundant data. In the event that we need to repeat a bioinformatics pipeline using a different toolchain, we need to be able to query the dataset at the location. As can be seen in Figure 4.6, an instance of a bioinformatics pipeline is related to a sequence data set and a publication. By joining the three tables `Publication`, `Pipeline`, and `Sequences`, we get all publications related to a specific sequence data set and *vice versa*.

Sequences		
PK	experiment_id	SERIAL
U	data_set_name	VARCHAR
NN	read_count	INTEGER
NN	server	VARCHAR
NN	path	VARCHAR
	description	TEXT

Sequences are the product of a single PCR experiment. We use its primary key directly to identify sequence data sets.

Listing 4.12: Define table Sequences for locating sequence data sets and add foreign key constraint on primary key.

```

1 CREATE TABLE Sequences(
2     experiment_id INTEGER PRIMARY KEY,
3     data_set_name VARCHAR UNIQUE,
4     read_count INTEGER NOT NULL,
5     server VARCHAR NOT NULL,
6     path VARCHAR NOT NULL,
7     description TEXT
8 );
9
10 ALTER TABLE Sequences ADD FOREIGN KEY experiment_id
11 REFERENCES PCR_Experiment (experiment_id);

```

### Table Pipeline

There is no standard pipeline for processing sequence data. The PCR is a stochastic method, and therefore, leads to different results in terms of efficiency and read quality. The sequence intermediates are monitored with tools like FastQC and MultiQC that collect statistics. These statistics impact the tooling of the subsequent steps. Therefore, it is plausible to allow the registration of more than one pipeline for the same set of sequences. We force a sequence record to be registered in the Sequence table before it is referenced. In this version, the log is stored as free text. We chose to store the resulting OTU data as this allows for elaborate meta-studies such as calculating common OTUs between two different pipelines or datasets.

Pipeline		
PK	id	SMALLSERIAL
FK	sequences_id	
NN	protocol	TEXT
NN	lab_staff	VARCHAR
	otu_ids	SERIAL[]



Listing 4.13: Define Pipeline table for associating sequence data sets and bioinformatics pipelines.

```

1 CREATE TABLE Pipeline (
2     id SMALLSERIAL PRIMARY KEY,
3     sequences_id INTEGER,
4     protocol TEXT NOT NULL,
5     lab_stuff VARCHAR NOT NULL,
6     otu_ids INTEGER[]
7 );
8
9 ALTER TABLE Pipeline ADD FOREIGN KEY (sequences_id)
10 REFERENCES Sequences (experiment_id);

```

### Table Publication

It is common practice that researchers tasked to build a pipeline would ask for previous studies conducted at the lab on the same type of samples. Because the individuals who conducted the earlier study may no longer be part of the laboratory, it is useful to correlate a publication and the underlying pipeline. We are aware that the methods section of a paper is intended to name all tools, parameters, and versions, but it is often incomplete because some parameter settings may not be relevant to the reader, and it is not forced by publishers to be complete or in a particular format. The purpose of assigning a publication to a specific pipeline is to thereby uniquely assign an experiment and the data.

Publication		
PK	id	SERIAL
FK	pipeline_id	
U	doi	VARCHAR
	date	DATE
N	description	TEXT

Listing 4.14: Define table Publication to associate publications and analyses.

```

1 CREATE TABLE Publication(
2     id SERIAL PRIMARY KEY,
3     pipeline_id INTEGER NOT NULL,
4     doi VARCHAR UNIQUE,
5     publication_date DATE NOT NULL,
6     description TEXT
7 );
8
9 ALTER TABLE Publication ADD FOREIGN KEY (pipeline_id)
10 REFERENCES Pipeline (id);

```

### Table OTU

OTUs are the end product of a specific bioinformatics pipeline, and can therefore be identified via the field `pipeline_id`. A single OTU is a group of sequences that have high sequence similarities and can be represented with very little data. Namely, a common-sense sequence to clarify their identity, and a reading count. These two data

DB	Lineage				
AlgaeBase	Plantae	Viridiplantae	Streptophyta	Charophyta	Zygnematomyceae
GenBank	Viridiplantae	Streptophyta	Streptophytina	Zygnematomyceae	Zygnematomycidae
Silva	Archaeplastida	Chloroplastida	Charophyta	Phragmoplastophyta	Zygnematomyceae

Table 4.6: Each of the three databases uses a different taxonomy. As a result the species *Closterium acerosum* has three distinct lineages. For the sake of brevity, only the first five levels below Eukaryota are shown.

points are sufficient, for example, to investigate the efficiency of different toolchains or to execute the OTU resolution step on a different library.

OTU		
PK	id	SERIAL
FK	pipeline_id	
	read_count	INTEGER
	sequence	VARCHAR
FK	taxon_id	

Listing 4.15: Define table OTU to store computed OTUs of a bioinformatics pipeline.

```

1 CREATE TABLE OTU(
2   id SERIAL PRIMARY KEY,
3   pipeline_id INTEGER,
4   read_count INTEGER,
5   sequence VARCHAR,
6   taxon_id INTEGER
7 );
8
9 ALTER TABLE OTU ADD FOREIGN KEY (pipeline_id) REFERENCES Pipeline (id);

```

### Table Taxon

The biggest challenge in metabarcoding on plankton samples lies in the immense biodiversity. As plankton is part of more than half of the supergroups, taxonomic rearrangements likely affect the OTU resolution. The Taxon table allows the capture of different taxonomies using the taxonomic identifier in combination with the taxonomy source as a composite key. This way, taxonomies are separable despite being stored in the same table. To give an example, the species *Closterium acerosum* has different lineages in two important databases: Silva and GenBank.

Not only does the taxonomic structure vary between different sequence databases, but the naming of taxa is also ambiguous. For example, Chloroplastida and Viridiplantae are synonyms. The group of freshwater green algae (Charophyta) is sometimes

treated as a division, a superdivision, or an unranked group.

To clarify the ancestry, we can store either only the parent taxon or the full list of ancestors. In the first case, lineage resolution requires a recursive query of the taxon table; in the second case (as implemented here), we offload the unrolled lineage at the expense of more memory.

Taxa have at least one scientific name and often many aliases, and since taxonomic nodes of different taxonomies usually refer to the same organism (group), we extract the taxon name into another table (see Taxon\_Names below). Currently, there are at most 73 rank designations, of which only a dozen are frequently used. Again, incorrect input is prevented by forcing the use of a registered rank from the rank table.

Taxon		
PK	id	SERIAL
CU	taxid	INTEGER
FK, CU	taxon_src_name	
FK	taxon_names_id	
FK	lineage_id	
FK	rank_id	

Listing 4.16: Define table Taxon as a node of a specific taxonomy with rank and lineage associations.

```

1 CREATE TABLE Taxon(
2   id SERIAL PRIMARY KEY,
3   taxid INTEGER NOT NULL,
4   tax_src_id INTEGER NOT NULL,
5   tax_names_id INTEGER NOT NULL,
6   rank_id INTEGER NOT NULL,
7   lineage_id INTEGER NOT NULL,
8   UNIQUE(taxid, tax_src_id)
9 );
10
11 ALTER TABLE Lineage ADD FOREIGN KEY (parent_taxon_id) REFERENCES Taxon(id);
12 ALTER TABLE OTU ADD FOREIGN KEY (taxon_id) REFERENCES Taxon(id);

```

### Table Rank

Rank names like *phylum*, *order*, *family*, or *species* are stored separately to avoid misspelling. An alias field allows the listing of synonyms or Latin designations.

Rank		
PK	id	SMALLSERIAL
U, NN	name	VARCHAR
	aliases	VARCHAR[]

Listing 4.17: Define the taxonomic Rank table and add foreign key constraints to a taxon.

```

1 CREATE TABLE Rank(
2     id SMALLINT PRIMARY KEY,
3     name VARCHAR UNIQUE NOT NULL,
4     aliases VARCHAR[]
5 );
6
7 ALTER TABLE Taxon ADD FOREIGN KEY (rank_id) REFERENCES Rank (id);

```

### Table Taxon Names

Storing multiple taxa in the Taxon table will introduce redundant taxon namings. For example, the species *Closterium acerosum* has the taxonomic id 130971 in GenBank, but a different one in Silva. We add a field for listing alias names.

Taxon_Names		
PK	id	SERIAL
NN	taxon_id	INTEGER
NN	taxon_name	VARCHAR
	aliases	VARCHAR[]

Listing 4.18: Define table Taxon\_Names for scientific names and aliases of taxonomic nodes and add a foreign key constraint to Taxon table.

```

1 CREATE TABLE Taxon_Names(
2     id SERIAL PRIMARY KEY,
3     taxon_id INTEGER NOT NULL,
4     taxon_name VARCHAR NOT NULL,
5     aliases VARCHAR[]
6 );
7
8 ALTER TABLE Taxon ADD FOREIGN KEY (tax_names_id)
9     REFERENCES Taxon_Names (id);

```

### Table Lineage

As described above, we decided to unroll lineages and store them as a list of taxa. Taxonomic trees are widely and flatly organized, meaning that hundreds of organisms share the same parental lineages.

Lineage		
PK	id	
FK	parent_taxon_id	
	grand_taxon_ids	SERIAL[]

Listing 4.19: Define table Lineage for taxonomic lineages and add foreign key constraints to Taxon.

```

1 CREATE TABLE Lineage(
2   id INTEGER PRIMARY KEY,
3   parent_taxon_id INTEGER,
4   grand_taxon_ids INTEGER[]
5 );
6
7 ALTER TABLE Taxon ADD FOREIGN KEY (lineage_id) REFERENCES Lineage (id);

```

### Table Taxonomy Source

The second part of the composite key in the Taxon table is the taxonomy source identifier that we store here. The main non-synchronized taxonomies are from GenBank or Silva, but phylogenetic trees based on alignment studies from Hug et al., 2016 or Burki et al., 2020 can also be entered. We use a small integer to auto-generate an identifier and enforce a named source and an URL. Since taxonomies are periodically updated based on recent findings or committee decisions, we add a field for the last update performed in the format DATE<sup>11</sup> and an optional field description.

Taxonomy_Source		
PK	id	SMALLSERIAL
NN	name	VARCHAR
NN	url	VARCHAR
NN	last_update	DATE
	description	TEXT

Listing 4.20: Define table Taxonomy\_Source for storing metadata of taxonomic source files.

```

1 CREATE TABLE Taxonomy_Source(
2   id SERIAL PRIMARY KEY,
3   name VARCHAR UNIQUE NOT NULL,
4   url VARCHAR NOT NULL,
5   last_update DATE NOT NULL,
6   description TEXT
7 );
8
9 ALTER TABLE Taxon ADD FOREIGN KEY tax_src_id
10  REFERENCES Taxonomy_Sources (id);

```

### Table Organism Group

The study described in section 2.5 has shown that resolving the composition of a sample to species level is not feasible for all clades and not always useful. Organisms are therefore grouped by life forms, life stages, or lowest common ancestors by either method. The most narrowing taxonomic description is to assign a common ancestor. However, there are examples of polyphyletic groups. We therefore enforce the listing of taxa that belong entirely to this group.

<sup>11</sup>one-day resolution is sufficient since taxonomies are never updated more than once per day

Organism_Group		
PK	id	SERIAL
U	name	VARCHAR
	aliases	VARCHAR[]
NN	taxon_ids	INTEGER[]
	description	TEXT

To ensure that listed taxa in `taxon_ids` correspond to existing taxa in table `Taxon`, we can add a trigger as done for the `Experiment` table (see Appendix C.1).

Listing 4.21: Define table `Organism_Group` for storing collective groups.

```

1 CREATE TABLE Organism_Group(
2   id SERIAL PRIMARY KEY,
3   name VARCHAR UNIQUE NOT NULL,
4   aliases VARCHAR[],
5   taxon_ids INTEGER[] NOT NULL,
6   description TEXT
7 );
8
9 ALTER TABLE Morph_Experiment ADD FOREIGN KEY (organism_group_id)
10  REFERENCES Organism_Group (id);

```

## 4.6.2 Roles

In a lab environment, there are three main roles with different privileges: a database administrator with the dual ability to create a schema, but also to grant or revoke access to users. Account management can be separated from schema modification tasks, as it does not require a deeper understanding of the schema. Laboratory staff and researchers make up the group that generates experimental data or processes data. All their introduced modifications must comply with the rules to ensure the sound state. For visiting scientists, interns, or students who access only historical data, it is sufficient to grant read access as discussed in more detail in Section 4.5.7.

Role Name	Actors	Rights
Admin	Long-term employers	Create and modify the schema for structuring data; create user accounts and granting access with different sets of privileges
User A	Lab operator, long-term researcher	Needs to manipulate data by inserting or correcting result
User B	Visiting researcher, student	Needs to query historical data, but does not need or is not trusted to manipulate data

Table 4.7: Proposed role assignment of lab members.

With the database schema in hand, we can answer our initial questions. See C.2 how these translate into SQL queries. A broader discussion follows in Section 5.3.

## Chapter 5

# Discussion

### 5.1 Metabarcoding

In Chapter 2, we have discussed the fundamental challenges that metabarcoding is facing: a missing consensus among biologists about the tree of life structure, computational challenges when deducing phylogenies from a set of DNA sequences, and the unavoidable trade-off between taxonomic breadth and low-level resolution of barcode-based identification. These problems have not yet been solved and complicate identification because it is impossible to consult separately curated sequence databases.

#### 5.1.1 Taxonomic Sparseness of Sequence Libraries

A well-covered database allows statistically more reliable identification, and the discovery of new barcodes and primer sequences. For many taxonomically described species, there is no or only a single reference sequence from a short genomic region. DNA metabarcoding must capture species diversity that spans more than half of the eukaryotic supergroups, as shown in Figure 2.3 of Section 2.2. The difficulty in cloning and whole-genome sequencing microplankton challenges the search for new barcodes additionally.

In practice, primers are searched on a few, taxonomically related sequences. How large the actual primer coverage is can only be determined by practical experiments. The primer pair DIV4, for example, was originally designed on a set of *Sellaphora*<sup>1</sup> genomes. Zimmermann, Jahn, and Gemeinholzer, 2011 discovered a 400 bp segment of the 18S rRNA that would allow distinction of the genus *Sellaphora* from closely related genera. For distinction they tested the primer set on 123 sampled diatom sequences<sup>2</sup>. However, practical tests on samples revealed an exceptionally high discrimination and detection rate for other phytoplankton clades. OTU resolution was only possible because most reference sequences cover the ribosomal 18S region. To make barcode searches truly unbiased, more genomic regions or even whole genomes need to be sequenced.

Ribosomal subunits like 16S or 18S have long been considered important marker regions. The decreasing cost of sequencing, the advent of affordable 454 pyrosequencing (Margulies et al., 2005), and PhyloChip (a microarray for 16S surveys) are leading to a flood of 16S data (DeSantis et al., 2007) and therefore contribute to the majority of online submissions.

Nevertheless, the standard for classifying new lineages is the clone and sequence approach. Only complete genomes allow accurate phylogenetic placement in the

---

<sup>1</sup>a diatom genus

<sup>2</sup>via *in silico* PCR

tree of life and enable comprehensive searches for unique barcodes. The cost of sequencing and assembling whole genomes is slowing the pace of taxonomy refinement. Nevertheless, the number of additions to GenBank is increasing daily. Figure 5.1 shows the growth in terms of number of nucleotides and sequences since their introduction. The number of nucleotides last doubled four years ago, and the number of sequences ten years ago (GenBank). Since 2005, even more data, in terms of number of bases, has come from whole-genome sequencing (WGS).

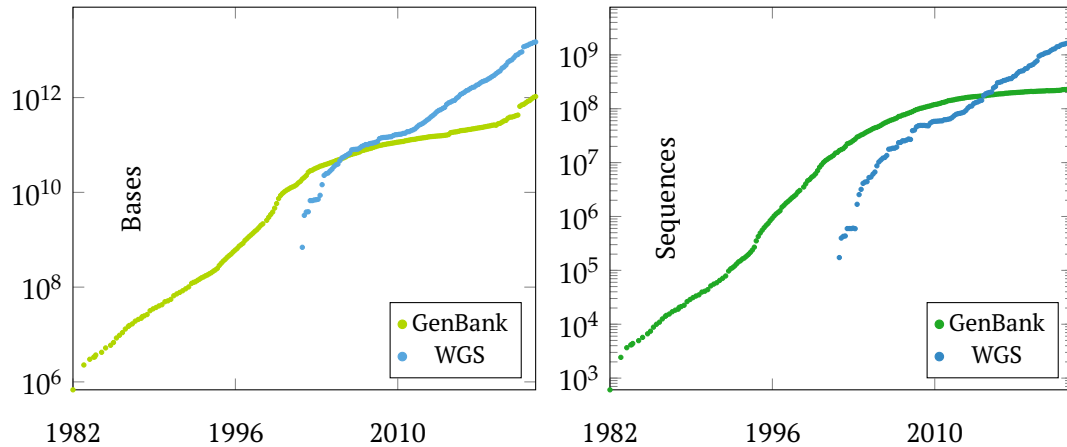


Figure 5.1: Growth of nucleotides and sequences in GenBank and its WGS component between 1982 and 2021. Accumulated release statistics retrieved from <https://www.ncbi.nlm.nih.gov/genbank/statistics> on 14.02.2022.

We need tools that allow *in silico* primer search with the few genomes and sequence fragments available. It is questionable whether clades that have no direct impact on the human species will ever achieve such sequence coverage that they can be studied by more holistic approaches. Scalable *in silico* PCR tools would allow us to predict the universal effectiveness of a primer pair. Thereby, the costs for exploring a potential universal primer are reduced. *K*-mer based approaches, in contrast to MSA-based approaches, are capable of solving the primer discovery problem because *k*-mers are searched location-independent and do not require similarity between sequences. Frequent *k*-mers as potential primer sequences can be identified quickly in FM-indices.

### 5.1.2 Tree of Life Construction

Another difficulty for barcode-based identification is due to the history of taxonomic placement. Large parts of the phylogenetic tree of life are structured based on features that are observable by the eye, such as morphology or other biological characteristics. These commonalities may not be reflected in the chosen barcode. The arbitrary nature of the classification means that there are no uniform division criteria for all groups of life, and is the cause of disagreement. For the most crucial rank – species – there exist plenty of concepts. Apart from the reproduction- or phylogenetic-based criteria, there exist concepts based on isolation (no interbreeding), recognition (of mates), occupation of ecological niches, evolution, genealogy, genotypic clustering, etc. (see Table 1 in De Queiroz, 2007). All those concepts represent important abstractions and reductions of the many underlying DNA changes. Some only measure the biological and behavioral impact, what mostly matters for survival, and may also explain the pressure and direction of changes.



Most organisms that are visible to the human eye already have their place in the family tree of life. In the era of NGS, we can identify previously unseen microorganisms and classify them primarily based on their molecular properties. This trend is opportune for barcode-based identification and will make metabarcoding an even more meaningful and robust tool. It also has a corrective impact on the existing taxonomy. Just recently major clades have been rearranged given new molecular evidence as proposed by Burki et al., 2020.

As reported by Nakov, Beaulieu, and Alverson, 2018, genome comparison of the 20 most abundant marine planktonic diatom genera revealed that their ages ranged from 4 to 134 million years. In other words, some same ranked diatom genera have, in fact, an evolutionary lead of 130 million years. When looking at the diatom order Thalassiosirales, one of its contained genera, Thalassiosira, is more a polyphyletic set of species. Besides, four of the eight Thalassiosirales genera are nested within Thalassiosira with a common ancestor that dates back more than 63 million years.

## 5.2 PriSeT

In Chapter 3, we presented PriSeT, a primer discovery tool that reliably identifies pairs of sequences suitable as PCR primers. PriSeT is robust concerning the quality of the sequence sources. The  $k$ -mer-based approach overcomes computationally expensive multiple sequence alignments and makes PriSeT independent of the sequence quality.

Given the current state of the largest publicly available sequence database, GenBank has low sequence coverage for plankton taxa. It was more than surprising that PriSeT found for some clades new primer pairs offering a broader coverage or barcode variation than published ones and are at the same time chemically suitable for a classical PCR (see Table 2.3). PriSeT correctly output primer pairs known to be present in the library if and only if they pass the constraint sets  $C_s$  and  $C_p$  (see Table 3.13). When having complete genomes available for primer discovery too many candidates are produced, and it is necessary to narrow down the primer sequence constraints or filter in a post-processing step, e.g., for pairs producing amplicons that are distinctive or span exons in case annotations are available.

The experiments in Section 3.11 have shown that when searching for primer pairs for metabarcoding experiments, it is appropriate to use frequency as an initial filtering heuristic. Only  $k$ -mers occurring with a minimum frequency will later satisfy sufficient coverage or amplicon variation. The FM-index is a transformation that supports frequency queries with lower costs than a seed-and-extend approach as used in FastPCR by Kalendar et al., 2017, or an MSA-based approach, which requires manageable data sets to identify conserved regions serving as primer binding sites.

None of the existing primer search tools we examined was capable of processing multi-sequence libraries and optimizing for high-frequent primer pairs simultaneously. We built PriSeT to satisfy this need. Given the declining cost of NGS, databases are getting larger, making curation impossible at all, and given the low coverage of plankton clades, we cannot afford to exclude resources.

GenBank does not stipulate the annotation format for labeling sequences by their origin (e.g., as 18S or COI). Conclusively, the sampling approach also collected non-18S sequences, explaining the relatively low values for coverage and amplicon variation. When users evaluate PriSeT's computed primer pairs, they have to consider the heterogeneity of the originating regions.

The PriSeT version at hand does not include coverage or amplicon variation criteria into the filtering for not limiting the users' options – the benefit of a higher coverage is in many cases paid with a lower number of distinct reads (see results in Table 3.14). It should remain in the users' hands to decide when coverage is favored over amplicon variation. The recent SARS-CoV-2 outbreak has demonstrated the importance of scenarios where the goal is to recover barcodes as distinguishing clades. In the case of SARS-CoV-2, it was sufficient to ensure uniqueness from its closest relatives in the same subfamily (Orthocoronavirinae); the identified barcodes had no matches outside its group. Yet, because many taxa were not classified according to phylogenetic criteria, these findings cannot be generalized.

By then, we mostly have to rely on partial sequences for primer discovery. PriSeT can handle both situations: taxonomically wide-spread and sparse sequence libraries, but yields more primer candidates when longer sequences are available as in the case of pathological complete viral genomes. With more candidates available, additional requirements can be implemented, like barcode specificity.

### 5.2.1 Open Source Code

It is common knowledge that software systems improve substantially upon publication of underlying algorithms and source code. The exposure allows a large community of hackers and enthusiasts to depict errors. For example, the code exposure of widely used cryptographic schemes made them robust over time. When looking at other areas, it is not common to publish code. For example, some of the primer check tools used for debugging purposes displayed occasionally faulty behaviors.

Users tend to be more confident with online tools than with unhandy command-line tools. For none of the online primer search tools, their source code is provided, and even worse, there is no description available of the underlying algorithms (see Blast/Primer3 by Ye et al., 2012, Primer Search Tool by Tusnády et al., 2005, Multiple Primer Analyzer by Thermo Fisher Scientific, or the standalone tool FastPCR by Kalendar et al., 2017). All associated publications remain vague in their algorithmic sections. An undetected self-annealing pattern in one of the primer sequences may result in a costly and ineffective PCR.

To give an example PCR Primer Stats<sup>3</sup> failed to identify a self-annealing or cross-annealing 4-mer pattern and Thermo Fisher Scientific's online tool Multiple Primer Analyzer<sup>4</sup> detects self-annealing 4-mers, but fails to discover a disconnected self-annealing pattern and does not check for cross-annealing when providing multiple sequences (see Figure 5.2).

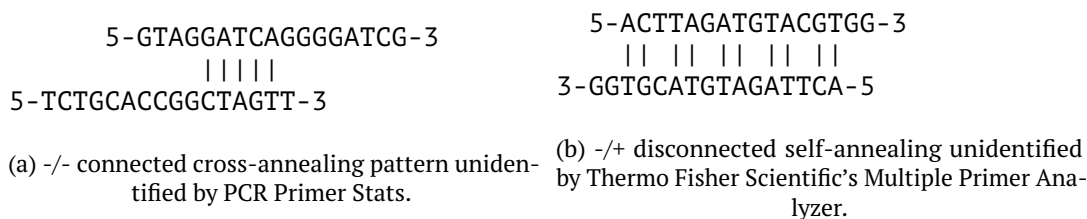


Figure 5.2: Critical annealing patterns missed by two online tools.

An openly available specification clarifies whether a behavior is faulty or intended. In this spirit, we host PriSeT in a public repository such that users pinpoint errors or request additional features.

### 5.2.2 Performance

The sequence data set sizes of important plankton clades do not exceed a half gigabyte<sup>5</sup>. For very large datasets we recommend not to create an FM-index over the whole library, but to split the library and calculate FM-indices independently. This also impacts the runtime of the  $k$ -mer frequency computation as it does not grow linearly with the library size (see theoretical runtimes in Table 3.9 of Section 3.10).

PriSeT operates step-wise, i.e., all  $k$ -mers with frequencies exceeding a threshold are collected at once into a single data structure, filtered, and combined reference-wise. To give an example, Dikarya from the fungal kingdom (clade 451864) produces 119.4 million  $k$ -mers. The main memory occupation of the location map received from GenMap represents the current bottleneck of PriSeT. Processing libraries beyond

<sup>3</sup>[https://www.bioinformatics.org/sms2/pcr\\_primer\\_stats.html](https://www.bioinformatics.org/sms2/pcr_primer_stats.html) accessed in January 2020.

<sup>4</sup><https://tinyurl.com/ybj4aqbc>

<sup>5</sup>An FM-index is by a factor of 4.2 larger than the original library.

500 MB is currently only feasible when increasing the  $k$ -mer frequency cutoff, s.t. not more than roughly 120 million  $k$ -mers ( $\approx 1$  GB) are produced<sup>6</sup>.

The temporary space occupation can be tackled by interweaving  $k$ -mer frequency and filtering: a frequent  $k$ -mer immediately undergoes filtering and is only collected when satisfying the frequency threshold and constraint set  $C_s$ . Input libraries composed of multiple reference sequences would additionally profit from the reference-wise partitioning approach. We can apply this strategy to the combination step since  $k$ -mers are pairable only if they refer to the same sequence; each of the references can be processed in parallel. The current version of PriSeT is v1.0 and does not use any sort of thread or process parallelism. Some obvious parallelization options are:

- $k$ -wise for FM frequency calls: given sufficient main memory, all calls for  $k \in [\kappa_{\min} : \kappa_{\max}]$  are executed in parallel
- sequence-wise for  $k$ -mer pairing: multiple threads process distinct sequences
- position-wise for  $k$ -mer pairing: multiple threads process distinct search windows

The computationally most expensive step is the combination step. Its time complexity is dominated by the product of the library size and the search window, which is the targeted transcript length range (see Table 3.9 in Section 3.10). It is therefore critical to constrain free parameters as much as possible.

Stable dimerization of primer and template DNA is crucial for the success of a PCR. A single mismatch, especially at the 3'-end, may result in an ineffective PCR. On that account PriSeT is using the  $(k, 0)$ -frequency to gather only  $k$ -mer locations with 100 % sequence identity. Users can configure PriSeT to allow for up to four errors  $e$  (mismatches) for primer sequences. When allowing errors, a single  $k$ -mer occurrence adds to all location collections referring to  $k$ -mers with Hamming distances less or equal to  $e$ . The relaxation of sequence identity impacts the size of the candidate set and must be carefully selected.

---

<sup>6</sup>exemplary for a desktop computer with 16 GB RAM

## 5.3 A Database for Metabarcoding Experiments

In Chapter 4, we analyzed the workflow of a research laboratory that performs DNA metabarcoding for monitoring freshwater biomes. We identified stumbling blocks that slow down productivity and iterative feedback through inefficient and error-prone data transmission. We proposed the implementation of a database management system as a laboratory-specific solution. We hope that the proposal will encourage biological laboratories to adapt and optimize their data management and workflows early enough.

An RDBMS reduces data movement and complex infrastructure maintenance. It allows data consistency to be enforced as needed through user-defined restrictions and allows roles to be assigned with graduated permissions. The customized scheme does not cause additional costs, unlike commercial laboratory management software. Set up in a minimal form, it does not disrupt the usual workflows of long-time laboratory operators. Existing laboratory data management software (Laboratory Information Management System, LIMS) includes workflows, data tracking, and interfaces for data exchange. Companies that sell LIMS as software are Accelrys, Illumina, Siemens, ThermoFisher Scientific, etc. The reasons why such systems are not considered by research groups in general are

- Critically small number of group members
- High initial and constant maintenance costs
- Data privacy

LIMs are laid-out for fast-paced, commercial environments like manufacturing or medical laboratories, where tracking needs to fulfill a minimum set of standards like ISO/IEC 17025<sup>7</sup> or ISO 15189<sup>8</sup> for Medical laboratories. Such management systems are too elaborate for institutional research groups, and have high purchase and maintenance costs. Many of these systems are cloud-based, which would require additional contracting because of data compliance. Research groups are typically small<sup>9</sup> and so is the expected number of queries or modifications per day.

Graphical user interfaces or automatized routines for data input and querying can be implemented in any language that provides interfacing libraries like MS Excel readers or database connectors. The proposed scheme is intended as a starting point, e.g., the protocol steps in the Pipeline table are currently a free text field. Alternatively, we could require that only registered sequence processing tools are listed or that a serialized pipeline is uploaded. A serialized pipeline does not have to be reconstructed from a text description but can be drawn, deserialized, and executed.

In fact, with the implementation of the database scheme proposed here, the workflows of the laboratory technicians can remain untouched, as they have been part of the institute for many years and have developed and refined their methods of logging. The content of counting results, for example, can easily be transferred in an automatized manner into a database – Python offers libraries to parse MS Excel sheets<sup>10</sup> and access PostgreSQL databases<sup>11</sup>.

<sup>7</sup>General requirements for the competence of testing and calibration laboratories

<sup>8</sup>Requirements for quality and competence

<sup>9</sup>A study about group sizes of biology research groups in the UK revealed an average group size of 7.3 with a standard deviation of 4.5. The average composition is 3.0 doctoral students, 2.1 postdocs, 0.5 technicians, and 0.68 other staff like research associates (Cook, Grange, and Eyre-Walker, 2015).

<sup>10</sup>MS Excel and OpenOffice supporting libraries: `openpyxl`, `xlswriter`, or `xlwt`

<sup>11</sup>PostgreSQL supporting libraries: `psycopg2` or `py-postgresql`

The schema definitions are provided on GitHub<sup>12</sup>. We recommend refining or simplifying the schema where needed. The workflow difficulties examined here are from the perspective of a sequence data analyst – a more comprehensive requirements analysis is needed for all user groups involved.

### 5.3.1 What to Keep

The purpose of a database is not to store all the data generated in the laboratory for all time, but to capture the essentials to understand experimental steps and enable new types of analyses. The goal is to reduce the inherent complexity of metabarcoding studies, allow participants to reach a productive state more quickly, have assured data and pipelines for replication, and enable cumulative surveys.

The here presented database scheme scales perfectly with the number of experiments. It is not intended for storing raw data, which is the only unit of considerable size. However, the physical location should be reasonably static. In section 4.2 we indicated that sequencing a base is faster than storing a byte, and we, therefore, expect to encounter memory problems. Currently, server capacities are bought in when needed. The costs are relatively high relative to the lab's total budget – alone the management service for a server is €3,000 as offered by the university's computing center<sup>13</sup>.

A first and frequently used approach to reduce the extra storage space for sequencing data is lossless text compression. On read access, the compressed files are unzipped and later deleted. A zipped FASTQ file consumes about 25 % of the original file size. Another technique is *referential compression*. This technique can be applied to large, highly similar sequences such as genomes of the same species (e.g., Liu et al., 2017). For example, in the human genome, there are about 4 million variations from a reference genome that could be represented by only a few megabytes (Baker, 2010).

We can reduce the storage size by orders of magnitude by saving only dereplicated high-quality reads. However, this would demand agreeing on a minimal data processing pipeline. A single cluster can then be represented by a unique sequence and the size of the cluster. Additionally, compression would be possible.

At the IGB, steps have been taken to digitize all analytical results in the form of microscopic images or spreadsheets. Spreadsheets can be uploaded to an IGB internal server, while sequencing data are often stored on servers that are part of the university infrastructure. These systems are physically and digitally separated, and many users only have access to a single system.

The most radical approach is to delete the reads from the lab's server and keep only the OTU data. It is a *de facto* standard to archive sequencing data associated with publications databases such as GenBank at the National Center for Biotechnology Information (NCBI, Benson et al., 2012), the EMBL database of the European Bioinformatics Institute (Catherine, Cameron, and Janet, 2010), or the DNA Data Bank of Japan (DDBJ, Sugawara et al., 2008). In such a situation, laboratory copies are redundant, but research groups are reluctant to delete them.

Instead, laboratories tend to retain sequencing data in the form of assembled genomes, raw reads, and data intermediates. Data products quickly occupy terabytes of storage space, for which additional servers are rented. Some labs may have already reached the limit of what they can afford. Deciding what to keep and how to compress what needs to be kept is becoming unavoidable.

<sup>12</sup>[https://github.com/mariehoffmann/plankton\\_database](https://github.com/mariehoffmann/plankton_database)

<sup>13</sup><https://www.zedat.fu-berlin.de/ManagedServer>, (accessed on 01.06.2020)



## Chapter 6

# Conclusion

With a background in bioinformatics and computer science, I started working at an ecology working group with the primary goal of analyzing sequencing results. The originating samples were freshwater plankton samples, which exhibit an enormous diversity. I identified multiple stumbling blocks that we need to address. A profound one was the consolidation of data centered around environmental experiments to enable efficient and timely analyses. Chapter 4 presents the outcome – a no-budget solution in the form of a database schema for PostgreSQL, which consolidates the most relevant parts.

The other challenge was to identify species in heterogeneous samples using DNA barcodes. The computationally ideal solution would require that the genomes of all indicator species are complete. We could, for example, compute a combination of a few barcodes that guarantee unambiguous species assignment. However, whole-genome sequencing is not feasible for most plankton species (see Chapter 2). Therefore, we have to deal with the available sequences that mainly cover the SSU region. In addition to a missing *ground truth*<sup>1</sup>, we encounter underrepresented taxa even in the largest available sequence database: GenBank (Benson et al., 2012). Primer pairs have to be optimized iteratively, with only partial knowledge of the ground truth. At the same time, primer pairs must be sensitive and unbiased to well-covered taxa. The reason is that there might be yet unidentified species in the sample, or species exhibiting volatile occurrence patterns. A combination of universal and clade-specific markers is usually chosen.

Practical challenges aside, biologists disagree on a unifying species concept, resulting in noncongruent trees of life (TOLs) associated with different databases. The availability of more sequence data fosters a particular concept: that of phylogenetic distances. Currently, TOLs are synthesized from taxonomic and phylogenetic information (Chapter 2). Nevertheless, metabarcoding needs complementation by light microscopy (LM) for cryptic or unknown species or measurement of phenotypical parameters.

Most indices or metrics that describe the quality status of a biotope use species richness, composition, and abundance of indicator species (see Birk et al., 2012). While richness could be well captured via metabarcoding, an abundance estimation of high-abundant organisms is currently only feasible via light microscopy. The direct comparison of individual counts and read abundances of high abundant and well-defined species demonstrated (see Figure 2.15) that metabarcoding does not replace the manual method. Section 2.4.1 described the limitations of LM for some critical and high abundant organisms that are indistinguishable when present in a specific state (larvae, juvenile, male) and can, therefore, be only assigned to some higher-order taxon. In such a case LM underestimates these species.

---

<sup>1</sup>here: knowledge of all contained species in the sample

On the contrary, metabarcoding measures DNA abundance independent of whether the originating material is dead, alive, or in a morphologically indistinguishable state. Therefore, we would expect species records from metabarcoding before and possibly after they are seen under the LM in a time series. Dead organisms tend to sink and get enriched in marine sediments. Even though all samples were taken from surface water, Lake Müggelsee is a shallow lake with boat traffic and layers most often not stratified. Water is whirled up, and a significant amount of dead material may also be contained in surface water.

The iterative optimization of primer pairs for a better sample resolution needs to be supported by search tools that deal with hundreds of thousands of reference sequences that are phylogenetically close or distant. Chapter 3 presented the C++-tool PriSeT, which discovers new primer sequences exceeding a frequency threshold. PriSeT uses a  $k$ -mer-based approach to tackle scalability and robustness towards the reference data set. Concretely, PriSeT employs a bidirectional FM-index based on EPR<sup>2</sup>-dictionaries (see Pockrandt, Ehrhardt, and Reinert, 2017) and optimal search schemes for  $k$ -mer retrieval. PriSeT is fast enough in practice to process millions of primer candidates within seconds or minutes (see Section 3.12.4). In combination with transcript information, primers can be determined that not only serve as barcodes but also generate unique barcodes. Clade discrimination is particularly important for pathogen detection.

We evaluated PriSeT on reference sequences (mostly 18S rRNA genes) from 19 clades covering eukaryotic organisms typical for freshwater plankton samples. PriSeT has found several published primer sets as well as additional primer sets that are more chemically suitable. We compared frequency, taxon coverage, and amplicon variation with published primer sets for these new sets. One result was that for 11 clades, we found *de novo* primer pairs that cover more taxa than the published ones, and for six clades *de novo*, primers resulted in higher sequence (i.e., DNA barcode) variation (Table 3.14). Second, we applied PriSeT to 19 SARS-CoV-2 genomes and 114 new primer pairs were calculated, with sequences not allowed to occur in other taxa to avoid reporting false-positive results (Figure 3.8). These primer sets would be suitable for empirical testing. These findings show that although the search space is very limited, it still contains unknown primer pairs that could potentially improve the sensitivity or specificity.

PriSeT is available under <https://github.com/mariehoffmann/PriSeT>. The repository contains example applications and documentation. Supplementary result data shown in Section 3.11 can be also found in tabular format under [https://github.com/mariehoffmann/PriSeT\\_denovo](https://github.com/mariehoffmann/PriSeT_denovo).

---

<sup>2</sup>Enhanced Prefixsum Rank



## Appendix A

# Species Identification in Environmental Samples

### A.1 Abundance Plots

Figures A.1 and A.2 show the set of high abundant species identified under the microscope *and* by at least one marker (EUK15, EUK14, or DIV4). In case a morphological species could only be given the genus name (sp.), we accumulated OTU counters whenever an OTU could be resolved to that genus or a related species.

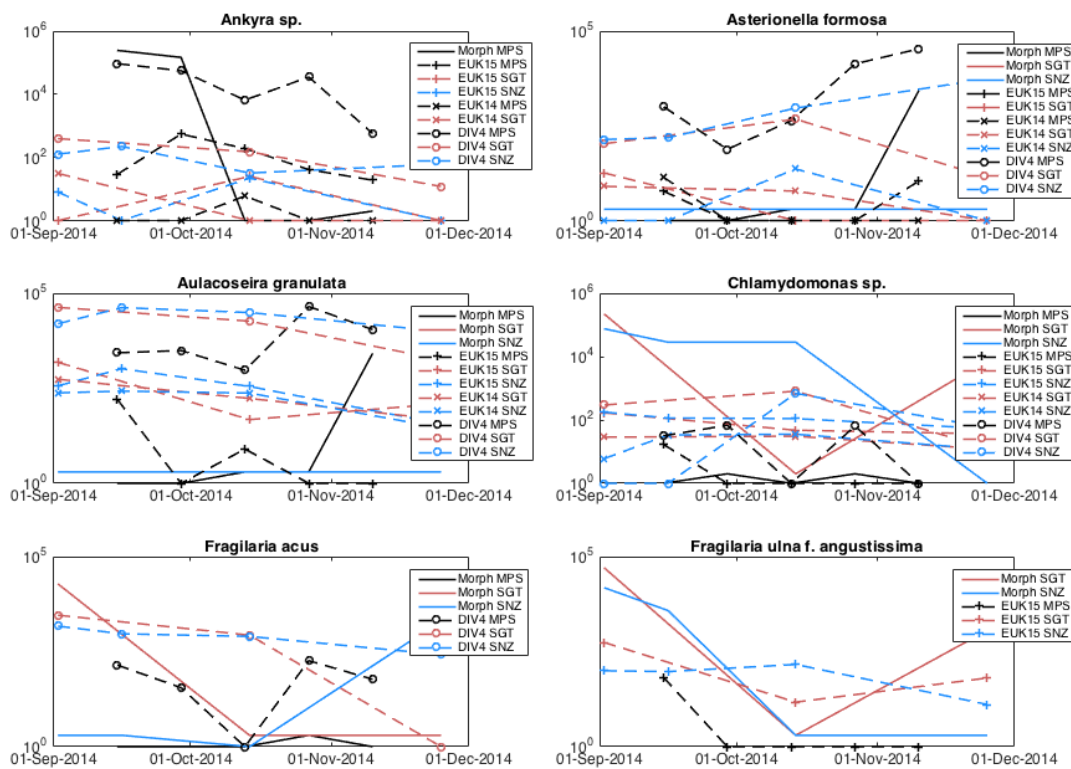


Figure A.1: Species abundances as individual (Morph) or read count abundances (part 1).

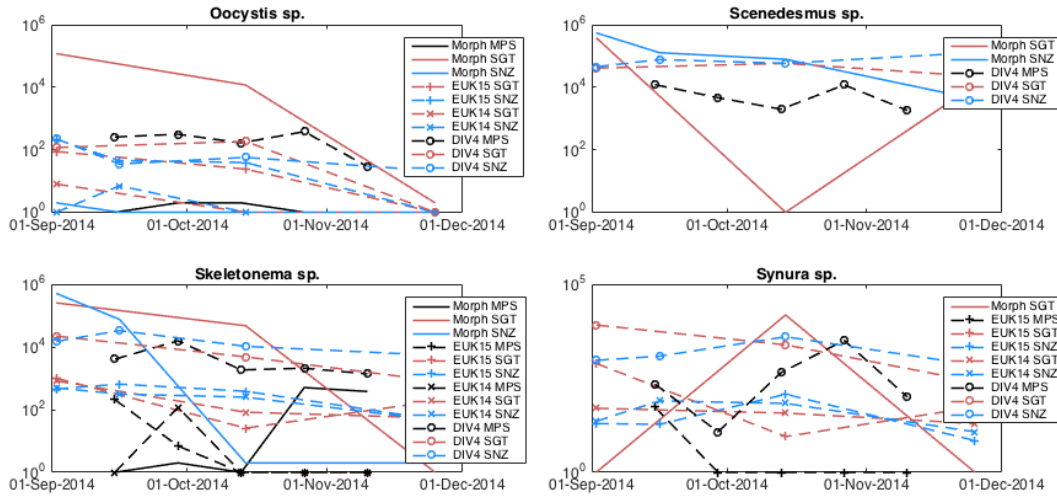


Figure A.2: Species abundances as individual (Morph) or read count abundances (part 2).

## A.2 R Code for Statistical Analyses

The scripts and data sheet (Plankton\_ID.ods) are available under <https://github.com/mariehoffmann/aquamarine>. Each analysis is done on three major groups:

1. Diatoms
2. Green Algae
3. Zooplankton

The data is read out from the tabular file using the R library readODS. An extra column contains the above group labels for subsetting. In a first step, the OTU data is transformed into the following layout: for each sample from one of the three sites (Lake MPS, or River SGT/SNZ)  $\text{cnt}_{x,y} \in \mathbb{N}_0$  denotes the OTU size (or the number of individuals for morphological identification) in terms of read counts (see Table A.1). Only OTUs of the same identification method and marker are comparable. That is why

Site	OTU <sub>1</sub>	OTU <sub>2</sub>	⋯	OTU <sub>n</sub>
MPS <sub>1</sub>	cnt <sub>1,1</sub>	cnt <sub>1,2</sub>	⋯	cnt <sub>1,n</sub>
MPS <sub>2</sub>	cnt <sub>2,1</sub>	cnt <sub>2,2</sub>	⋯	cnt <sub>2,n</sub>
⋮	⋮	⋮	⋮	⋮
MPS <sub>5</sub>	cnt <sub>5,1</sub>	cnt <sub>5,2</sub>	⋯	cnt <sub>5,n</sub>
SGT <sub>1</sub>	cnt <sub>6,1</sub>	cnt <sub>6,2</sub>	⋯	cnt <sub>6,n</sub>
SGT <sub>2</sub>	cnt <sub>7,1</sub>	cnt <sub>7,2</sub>	⋯	cnt <sub>7,n</sub>
SGT <sub>3</sub>	cnt <sub>8,1</sub>	cnt <sub>8,2</sub>	⋯	cnt <sub>8,n</sub>
SNZ <sub>1</sub>	cnt <sub>9,1</sub>	cnt <sub>9,2</sub>	⋯	cnt <sub>9,n</sub>
⋮	⋮	⋮	⋮	⋮
Site SNZ <sub>4</sub>	cnt <sub>12,1</sub>	cnt <sub>12,2</sub>	⋯	cnt <sub>12,n</sub>

Table A.1: Layout for statistical analyses.

the non-metric multidimensional scaling (NMDS) and multi-response permutation

procedure (MRPP) are carried out marker-wise. The transformed data is then subset into one of the three organism groups listed above.

### A.2.1 NMDS in R

Listing A.1: NMDS on all subsets and identification methods.

```

1 library(dplyr)
2 library(readODS)
3 library(paperR)
4 library(pracma)
5 library(stringr)
6 library(vegan)
7
8 fname <- "Plankton_IDs.ods"
9
10 sheets <- list(list(method = "Morph_ID", table_idx = 2,
11                    last_row_idx = 235),
12               list(method = "EUK15_ID", table_idx = 3, last_row_idx = 325),
13               list(method = "EUK14_ID", table_idx = 4, last_row_idx = 506),
14               list(method = "DIV4_ID_", table_idx = 6, last_row_idx = 543))
15
16
17 cols_MPS_Morph_ID <- c("S23_MPS_140915",
18                       "S04_MPS_140929",
19                       "S06_MPS_141013",
20                       "S21_MPS_141027",
21                       "S24_MPS_141110")
22
23 cols_MPS_NGS_ID <- c(
24   cols_MPS_Morph_ID, c("S09_MPS_141027",
25                       "S46_MPS_141110"))
26
27 cols_SGT <- c("S31_SGT_140902",
28              "S07_SGT_141014",
29              "S17_SGT_141125")
30
31 cols_SNZ <- c("S32_SNZ_140902",
32              "S38_SNZ_140916",
33              "S08_SNZ_141014",
34              "S18_SNZ_141125")
35
36 organism_groups <- c("Diatoms", "Green_Algae", "Zooplankton")
37
38 # Group label colors
39 colvec <- c("gray27", "gray90", "gray57")
40 pchvec <- c(21, 21, 21)
41
42 # for plot display
43 sysname = (Sys.info()[ 'sysname' ])
44 switch(Sys.info()[ 'sysname' ]),
45 Darwin = {quartz()}, # X11()
46 Linux = {quartz()},
47 Windows = {windows()}
48
49 for (sheet in sheets)
50 {
51   print(sheet)
52   cols_MPS <- cols_MPS_NGS_ID
53   if (sheet$table_idx == 2)
54     {
55       cols_MPS <- cols_MPS_Morph_ID
56     }
57 }

```

```

55 # Subset of columns to select per sheet given by header name
56 cols <- c(c("Subset_MRPP_NMDS"), cols_MPS, cols_SGT, cols_SNZ)
57
58 # iterate over table indices
59 data <- read_ods(  fname,
60                  sheet$table_idx,
61                  col_names = TRUE,
62                  formula_as_formula = FALSE)
63 # Crop until last row
64 data <- data[1:sheet$last_row_idx, cols]
65 for (organism_group in organism_groups)
66 {
67   # Delete rows not matching group labels and discard first column
68   data.organism <- data[data$Subset_MRPP_NMDS == organism_group,]
69   print(data.organism)
70   data.organism <- data.organism[, -1]
71
72   # transpose and accumulate sampling sites into MPS, SGT, or SNZ
73   data.organism <- t(data.organism)
74
75   #print(head(data.organism))
76   data.organism <- transform(data.organism, site =
77     ifelse(str_detect(row.names(data.organism), "MPS"), "MPS",
78     ifelse(str_detect(row.names(data.organism), "SGT"), "SGT",
79     "SNZ")))
80
81   # delete rows containing only null values
82   data.organism <- data.organism[rowSums(data.organism[,
83     1:ncol(data.organism)-1]) != 0, ]
84
85   # extract row names and site IDs as columns
86   data.sites <- as.data.frame(  data.organism$site,
87                               rownames(data.organism))
88   colnames(data.sites)[1] <- "site"
89   data.sites <- tibble::rownames_to_column(data.sites, "sample")
90   data.sites <- transform(data.sites,
91     sampleID = unlist(
92       strsplit(data.sites$sample, c('_'))
93       [seq(1, 3*nrow(data.sites), 3)])
94
95   # delete tailing 'sites' column
96   data.organism = data.organism[, 1:ncol(data.organism) - 1]
97   data.rel <- decostand(data.organism, method = "log")
98   print(data.sites)
99
100  # NMDS with bray-curtis index as distance measure
101  data.nmnds = metaMDS(data.rel, distance = "bray", k = 2)
102  stress <- paste("Stress:_", sheet$method, "_",
103    organism_group, ":", data.nmnds$stress, sep = "")
104  print(stress)
105  goodness(data.nmnds)
106
107  # stress plot
108  png(filename = paste("./nmnds_plots/", sheet, "_", organism_group,
109    "_stress.png", sep = ""))
110  stressplot(data.nmnds)
111  dev.off()
112
113  # nmnds sites plot with labels
114  # save as svg to edit label positions later with graphic program
115  svg(filename = paste("./nmnds_plots/", sheet, "_", organism_group,
116    ".svg", sep = ""), width = 5, height = 5) #, units = 'in',

```

```

117                                     res = 300)
118   plot(data.nmds, display = "sites", ann=FALSE)
119
120   # plot convex hull
121   ordihull(
122     data.nmds,
123     data.sites$site,
124     display = "sites",
125     draw = c("polygon"),
126     col = NULL,
127     border = colvec,
128     lty = c(3, 3, 3),
129     lwd = 2.5
130   )
131
132   # plot colored
133   with(data.sites,
134     points(data.nmds,
135           display = "sites",
136           col = "black",
137           pch = pchvec[data.sites$site],
138           bg = colvec[data.sites$site],
139           cex = 2))
140
141   # annotate sites
142   text(data.nmds, "sites", labels = data.sites$sampleID, cex = 1.2,
143        pos = 3, offset = 0.8, col = "gray50")
144
145   # display stress value
146   text(0, 0, labels = c(paste("R^2_=", round(data.nmds$stress, 3),
147     sep = "")), cex = 1.25)
148   dev.off()
149 }
150 }
151
152 # Reduce digits after comma for terminal display
153 digits <- function(nvec, precision)
154 {
155   return(lapply(nvec, "format", digits = precision, nsmall = precision))
156 }

```

## A.2.2 MRPP in R

Listing A.2: MRPP on all subsets and identification methods.

```

1  library(dplyr)
2  library(readODS)
3  library(pracma)
4  library(stringr)
5  library(vegan)
6
7  fname <- "Plankton_IDs.ods"
8
9  # Sheets to read, 2: "Morph_ID", 3: "EUK15_ID", 4: "EUK14_ID", 5: "DIV4_ID"
10 sheets <- list(
11   list(method = "Morph_ID", table_idx = 2, last_row_idx = 235),
12   list(method = "EUK15_ID", table_idx = 3, last_row_idx = 325),
13   list(method = "EUK14_ID", table_idx = 4, last_row_idx = 506),
14   list(method = "DIV4_ID_", table_idx = 5, last_row_idx = 543))
15
16 # columns to pool for MPS

```

```

17 cols_MPS_Morph_ID <- c( "S23_MPS_140915",
18                        "S04_MPS_140929",
19                        "S06_MPS_141013",
20                        "S21_MPS_141027",
21                        "S24_MPS_141110")
22
23 # add technical replicates for NGS
24 cols_MPS_NGS_ID <- c(cols_MPS_Morph_ID,
25                     c( "S09_MPS_141027",
26                       "S46_MPS_141110"))
27
28 cols_SGT <- c( "S31_SGT_140902",
29              "S07_SGT_141014",
30              "S17_SGT_141125")
31
32 cols_SNZ <- c( "S32_SNZ_140902",
33              "S38_SNZ_140916",
34              "S08_SNZ_141014",
35              "S18_SNZ_141125")
36
37 # we apply mrpp pairwise
38 site_combinations <- c('MPS|SGT|SNZ', 'MPS|SGT', 'MPS|SNZ', 'SNZ|SGT')
39
40 organism_groups <- c("Diatoms", "Green_Algae", "Zooplankton")
41
42 results <- list()
43
44 for (sheet in sheets)
45 {
46   print(sheet)
47   cols_MPS <- cols_MPS_NGS_ID
48   if (sheet$table_idx == 2)
49   {
50     cols_MPS <- cols_MPS_Morph_ID
51   }
52
53   # Subset of columns to select per sheet given by header name
54   cols <- c(c("Subset_MRPP_NMDS"), cols_MPS, cols_SGT, cols_SNZ)
55
56   # iterate over table indices
57   data <- read_ods(fname, sheet$table_idx, col_names = TRUE,
58                  formula_as_formula = FALSE)
59
60   # Crop until last row
61   data <- data[1:sheet$last_row_idx, cols]
62   for (organism_group in organism_groups)
63   {
64     print(paste("organism_group_=", organism_group))
65
66     # Delete rows not matching group labels and discard first column
67     data_o <- data[data$Subset_MRPP_NMDS == organism_group,]
68     data_o <- data_o[, -1]
69
70     # transpose and accumulate sampling sites into MPS, SGT, SNZ
71     data_o <- t(data_o)
72     data_o <- transform(data_o, sites =
73                        ifelse(str_detect(row.names(data_o), "MPS"), "MPS",
74                              ifelse(str_detect(row.names(data_o), "SGT"), "SGT", "SNZ")))
75
76     # iterate and filter over all combination sets > 1
77     for (site_combination in site_combinations)
78     {

```

```

79     data_o_s <- dplyr::filter(data_o,
80       grepl(site_combination, sites))
81     # delete rows containing only null values
82     data_o_s <-
83       data_o_s[rowSums(data_o_s[,1:ncol(data_o_s)-1])!=0, ]
84     sites_factor <- as.factor(data_o_s[, ncol(data_o_s)])
85     mrpp_list <- mrpp( data_o_s[, 1:ncol(data_o_s) - 1],
86       sites_factor,
87       permutations = 500,
88       distance = "bray")
89     result <- list(sheet$method, organism_group,
90       site_combination, mrpp_list$n,
91       mrpp_list$classdelta, mrpp_list$Pvalue,
92       mrpp_list$E.delta, mrpp_list$delta)
93     results <- append(results, list(result))
94   }
95 }
96 }
97
98 # Reduce digits after comma for terminal display
99 digits <- function(nvec, precision)
100 {
101   return(lapply(nvec, "format", digits = precision, nsmall = precision))
102 }
103
104 # Display results
105 cat("/n-----\tRESULTS\t-----\n\n")
106 cat("_Method_ \t Organism_Group_ \t Sites_ \t \t Num_Observ_ \t Class_delta_ |")
107 cat("\t \t \t \t \t p-value_ \t \t E.delta_ \t delta_ | \n")
108 cat(" |--|--|--|--|--|--|--|")
109 cat("\n")
110 for (result in results)
111 {
112   cat("_")
113   cat(paste(result[1]))
114   cat("_|\t")
115   cat(paste(result[2]))
116   cat("_|\t\t")
117   s <- paste(result[3])
118   s <- str_replace_all(s, "\\|", ",")
119   cat(s)
120   cat("_|\t")
121   for (site in result[4])
122   {
123     cat(paste(site))
124   }
125   cat("_|\t\t")
126   cat(paste(digits(result[5], 2)))
127   cat(ifelse(lengths(result[5], use.names = TRUE) == 3,
128     "_|\t", "_|\t\t\t"))
129   cat(paste(digits(result[6], 4)))
130   cat("_|\t\t")
131   cat(paste(digits(result[7], 2)))
132   cat("_|\t\t")
133   cat(paste(digits(result[8], 2)))
134   cat("\n")
135 }

```





## Appendix B

# Primer Discovery in Large Datasets

### B.1 One-Letter Encodings of Nucleotides

Symbol	Base Representation			
A	A			
C		C		
G			G	
T				T
U				U
M	A	C		
R	A		G	
W	A			T
S		C	G	
Y		C		T
K			G	T
V	A	C	G	
D	A		G	T
H	A	C		T
B		C	G	T
N	A	C	G	T

Table B.1: One-letter encodings for (ambiguous) DNA bases.

## B.2 Runtimes for Plankton Datasets

Clade Taxid	Name	Size [MB]	FM Freq	Runtimes [ms]	
				Transform & Filter	Combine
304574	Charophyceae	0.42	1366	271	73
3041	Chlorophyta	31.79	127249	18575	6362
2825	Chrysophyceae	0.89	2844	728	311
3027	Cryptophyta	1.97	7309	1475	443
33849	Diatoms	4.98	14189	4259	961
2864	Dinophyceae	6.24	16321	5341	1682
33682	Euglenozoa	19.29	69457	8157	862
5747	Eustigmatophyceae	1.40	4498	536	234
554915	Amoebozoa	4.63	14609	3369	729
33651	Bicosoecida	0.15	757	183	110
28009	Choanoflagellata	0.28	1186	137	53
136419	Cercozoa	2.34	7276	1627	420
5878	Ciliophora	6.94	19749	6877	1899
6657	Crustacea	38.85	118194	2415	91
6231	Nematoda	82.44	400836	3756957	1411
27999	Perkinsidae	0.13	648	103	48
10190	Rotifera	1.57	3579	357	8
451864	Dikarya	518.44	3052423	98212	3191
112252	Fungi	10.21	27853	4066	500

Table B.2: Runtimes separated by major computation steps: FM frequency, transform and filter of  $k$ -mers, and pair formation.

### B.3 Published SARS-CoV-2 Real-time RT-PCR Primers

Primer Name	Sequence (5' - 3')
2019-nCoV_N1-F	GACCCCAAATCAGCGAAAT
2019-nCoV_N1-R	TCTGGTACTGCCAGTTGAATCTG
2019-nCoV_N2-F	TTACAAACATTGGCCGCAA
2019-nCoV_N2-R	GCGCGACATTCCGAAGAA
2019-nCoV_N3-F	GGGAGCCTTGAATACACCAAAA
2019-nCoV_N3-R	TGTAGCACGATTGCAGCATTG
RNAse P (RP-F)	AGATTGGACCTGCGAGCG
RNAse P (RP-R)	GAGCGGCTGTCTCCACAAGT

Table B.3: Published Real-time RT-PCR primers for SARS-CoV-2 (Reijns et al., 2020).

## B.4 Primer Pairs computed by PriSeT for SARS-CoV-2

primer ID	primer fwd	primer rev
4b81c574f92f5b6b	TGTGGGCTCAATGTGTCC	AGAAATGCTGGACAACAGGG
fb9e459a5fd403d0	TGTGGGCTCAATGTGTCC	CAACAGGGCAACCTTACAAG
d77312094faec94a	TGTTGGGTGTTGGTGGCA	TGTGGGCTCAATGTGTCCA
20cdc3951a8444e3	AAGGCTGGTGGCACTACTG	CACTGTAGAGGAGGCAAAGA
fd353b662018da72	AAGGCTGGTGGCACTACTG	CACTGTAGAGGAGGCAAAG
8172266d6ee0d285	ATTTCGTGGTGGTACGGT	CTGGACTTCCCTATGGTG
11d43e7c3ecd6a99	TTGTTGGGTGTTGGTGGCA	TGTGGGCTCAATGTGTCC
5995c9f26e0676d3	TTGTTGGGTGTTGGTGGCA	TGTGGGCTCAATGTGTCCA
29b1f4099511af84	TTGTTGGGTGTTGGTGGCA	TGTGGGCTCAATGTGTCCAG
48a672e138679875	TGTTGGGTGTTGGTGGCA	TGTGGGCTCAATGTGTCC
1892c93190392948	ATTCCCACCAACAGAGCCT	CTGTGACTCTTCTTCCTGC
159f2e139785ffbe	ATTCCCACCAACAGAGCCT	GTGACTCTTCTTCCTGCTGC
96805f79c2a0e052	ATTCCCACCAACAGAGCCT	TGACTCTTCTTCCTGCTGC
7aaa108589efa2b2	TCCCACCAACAGAGCCT	CTGTGACTCTTCTTCCTGC
1206442a59c1402e	TCCCACCAACAGAGCCT	GTGACTCTTCTTCCTGCTG
bc18299b71ff09b3	TCCCACCAACAGAGCCT	TGACTCTTCTTCCTGCTGC
2a9b62341d175dd3	TGTGGGCTCAATGTGTCC	ACAGGGCAACCTTACAAGC
fde205d091dbf5c6	AAGGCTGGTGGCACTACTG	ACTGTAGAGGAGGCAAAGAC
4baddf7c639182bd	TGTGGGCTCAATGTGTCC	AACAGGGCAACCTTACAAGC
6f50a4785eea571e	ATTTCGTGGTGGTACGGT	CTGGACTTCCCTATGGTGC
401ba718965a2358	AAGGCTGGTGGCACTACTG	AGGAGGCAAAGACAGTGCTT
808a6ec19b1fa754	AAGGCTGGTGGCACTACTG	AGGAGGCAAAGACAGTGCT
39f016cd1cf70709	ATTTCGTGGTGGTACGGT	ATGGTGCTAACAAAGACGGC
cf8f8ca6678946c0	ATTTCGTGGTGGTACGGT	TGGACTTCCCTATGGTGCTA
aad2a480277c9caa	ATTTCGTGGTGGTACGGT	TGGACTTCCCTATGGTGCT
46ba4fda4b07809f	ATTTCGTGGTGGTACGGT	TGGACTTCCCTATGGTGCT
aaccdd3959e4527	AAGGCTGGTGGCACTACTG	GAGGAGGCAAAGACAGTGCT
913be450fe26d09c	AAGGCTGGTGGCACTACTG	GAGGAGGCAAAGACAGTGCT
f55da99ba5adaf67	GCTGCTGCTTGACAGATTG	GAAATCTGCTGCTGAGGC
25eedfb2945a69ab	ATTTCGTGGTGGTACGGT	TGGTGCTAACAAAGACGGC
6942e5c76ceb030f	AAGGCTGGTGGCACTACTG	AGAGGAGGCAAAGACAGTGCT
2153ca4cdd8250d5	AAGGCTGGTGGCACTACTG	AGAGGAGGCAAAGACAGTG
3861c9f4dc028f97	GTCCAGAACAACCCAAGG	GCTTCAGCGTTCTTCGGA
596418188191cecc	AAGGCTGGTGGCACTACTG	TAGAGGAGGCAAAGACAGTG
995b13e0fc05d365	GTCCAGAACAACCCAAGG	GCTTCAGCGTTCTTCGGA
d5a535fd1a44041a	AAGGCTGGTGGCACTACTG	GTAGAGGAGGCAAAGACAGT
1ffc143ae28fdb18	AAGGCTGGTGGCACTACTG	GTAGAGGAGGCAAAGACAG
785db585374bf1c5	GTCCAGAACAACCCAAGG	GCTTCAGCGTTCTTCGGAAT
f44b6842efd9edbb	ACAGGCAAACAGCACAAGC	GACAGGTGGTTTCTCAATCG
120b36b16368d3e5	GCTTGTGTTTTGGCTGCTG	GAAAGTTTACGCCCTGACAC
3de2c61dcbe9ef19	ATTCCCACCAACAGAGCCT	GTGACTCTTCTTCCTGCTG
d3c5486ddf1d1cec	CAGGCAAACAGCACAAGC	GACAGGTGGTTTCTCAATCG
b71cdb9247230ee7	ATTTCGTGGTGGTACGGT	AGGGAGCCTTGAATACACCA
13c180b2898140fc	ATTTCGTGGTGGTACGGT	AGGGAGCCTTGAATACACC
5a60bfd1673126f7	ATTTCGTGGTGGTACGGT	TGGTGCTAACAAAGACGGCA
9603b7faf904bfab	CTTGCTTTGCTGCTGCTTG	AAGAAATCTGCTGCTGAGGC
ec5a867fc3a7b57b	CTTGCTTTGCTGCTGCTTG	AGAAATCTGCTGCTGAGGC
ab4d63965beacadd	CTTGCTTTGCTGCTGCTTG	AGAAATCTGCTGCTGAGGCT
d8d6d128b0fbbfa8	CTTGCTTTGCTGCTGCTTG	GAAATCTGCTGCTGAGGC

primer ID	primer fwd	primer rev
bc11be2f542428d3	CCTTACCGCAGAGACAGA	AGACCACACAAGGCAGATG
2f8186519877510	GCCTTGTCCCTGGTTTCA	TCCGTGGAGGAGGTCTTA
47547a832a425d71	CCTTACCGCAGAGACAGA	CAGACCACACAAGGCAGA
753a121c64e12362	CCTTACCGCAGAGACAGA	CAGACCACACAAGGCAGAT
c79a3c0bca11e8c	GCCTTGTCCCTGGTTTCA	TCCGTGGAGGAGGTCTTAT
ae069f7176fa016e	CTACAGTGTTCACCTAC	TGGATAACCACTTCAGAGAGC
1b6899a98e69cbeb	CCGCAGAGACAGAAGAAAC	CACAAGGCAGATGGGCTA
da0fb36d836d8cd9	CTACAGTGTTCACCTAC	GGATAACCACTTCAGAGAGC
46676cb30e959c1b	CCTTACCGCAGAGACAGA	GACCACACAAGGCAGATG
a783fac1bd05c637	CCTTACCGCAGAGACAGA	GACCACACAAGGCAGATGG
3a10f162d16e7001	CCTTACCGCAGAGACAGA	ACCACACAAGGCAGATGG
f065169979c313af	CAAGCCTTACCGCAGAGAC	AGACCACACAAGGCAGATG
e94ddb81c38e38f6	CAAGCCTTACCGCAGAGAC	AGACCACACAAGGCAGATGG
fdabc47cd437dab4	CTCTACAGTGTTCACCT	TGGATAACCACTTCAGAGAGC
74cfaed3fd8a877a	CTCTACAGTGTTCACCT	GGATAACCACTTCAGAGAGC
f4074f98460c112e	CAAGCCTTACCGCAGAGAC	GACCACACAAGGCAGATGG
818cc24f947313f7	CCTTACCGCAGAGACAGA	ACCACACAAGGCAGATGGG
56a49f74320e52ab	CAAGCCTTACCGCAGAGAC	ACCACACAAGGCAGATGGG
bc45f67fe01820c5	CCGCAGAGACAGAAGAAAC	CACAAGGCAGATGGGCTAT
8bc807ff2b7245e5	CAAGCCTTACCGCAGAGAC	CACAAGGCAGATGGGCTAT
f8e5c9a2c1cf4de4	CCTTACCGCAGAGACAGA	GCAGACCACACAAGGCAGA
8381ab5256f9c4a	CCTTCGTGGACATCTTCGT	GTAGCAGGTGACTCAGGTT
f8c4c2eadd355907	CCGCAGAGACAGAAGAAAC	ACCACACAAGGCAGATGG
d8dc5c165aa3e9b9	CCGCAGAGACAGAAGAAAC	GACCACACAAGGCAGATGG
3739c465ddef266a	CCTTCGTGGACATCTTCGT	GTAGCAGGTGACTCAGGT
42939198c8de99f1	CCGCAGAGACAGAAGAAAC	ACCACACAAGGCAGATGGG
df6cd6ebf5846227	CCGCAGAGACAGAAGAAAC	GCAGACCACACAAGGCAGA
b6cebc57418e2c96	CCGCAGAGACAGAAGAAAC	GACCACACAAGGCAGATG
5fb65bd3e1e24571	CCGCAGAGACAGAAGAAAC	GCAGACCACACAAGGCAGAT
4d6e1bd61daee5d9	CCGCAGAGACAGAAGAAAC	CAGACCACACAAGGCAGA
13b955fb4a5f939e	CCGCAGAGACAGAAGAAAC	CAGACCACACAAGGCAGAT
5561fe18bab71084	CCGCAGAGACAGAAGAAAC	CAGACCACACAAGGCAGATG
3b234e91b23a2bda	CCGCAGAGACAGAAGAAAC	AGACCACACAAGGCAGATG
9fe99fafb93761e9	CCGCAGAGACAGAAGAAAC	AGACCACACAAGGCAGATGG
2f1d6e5af859ccba	GGTGTGACCGAAAGGTAAG	TCCGTGGAGGAGGTCTTATC
e0d0082a500aa4ef	GGTGTGACCGAAAGGTAAG	TCCGTGGAGGAGGTCTTAT
cca886c3334684c6	GGTGTGACCGAAAGGTAAG	TCCGTGGAGGAGGTCTTA
c811b6a14f804d28	CCGCAGAGACAGAAGAAAC	ACACAAGGCAGATGGGCTAT
5d1d86286104193e	CCGCAGAGACAGAAGAAAC	ACACAAGGCAGATGGGCTA
ceef8450fb2dca51	CCGCAGAGACAGAAGAAAC	ACACAAGGCAGATGGGCT
ca93d3c4faa1b9b4	TCAAGCCTTACCGCAGAG	AGACCACACAAGGCAGATG
5ad419d2177a1aa8	TCAAGCCTTACCGCAGAG	CAGACCACACAAGGCAGAT
8a99ac42437987df	TCAAGCCTTACCGCAGAG	CAGACCACACAAGGCAGA
77e26c50492741a5	TACTGCCGTTGCCACATAG	TCTGCGGTATGTGGAAAGG
d9351aa64f98162	TACTGCCGTTGCCACATAG	CGTCTGCGGTATGTGGAAA
48d9a9de8128f216	TCAAGCCTTACCGCAGAG	GCAGACCACACAAGGCAGA
b73b0135194882b3	TACTGCCGTTGCCACATAG	CGTCTGCGGTATGTGGAA
b9ceaeac1bc8b0ed	TACTGCCGTTGCCACATAG	TTGTGCTAATGACCCTGTGG
f558e43b7cabd547	CAAGCCTTACCGCAGAGAC	CAGACCACACAAGGCAGAT

primer ID	primer fwd	primer rev
8896eef4f7641d17	CTTCGTATTGCTGGACACC	AGCGTGTAGCAGGTGACTCA
fd2cedb6abf2db7e	CTTCGTATTGCTGGACACC	AGCGTGTAGCAGGTGACTC
89624fee01dbb45b	CTTCGTATTGCTGGACACC	AGCGTGTAGCAGGTGACT
3383b86f2565d058	CCTTCGTGGACATCTTCGT	GTGACTCAGGTTTTGCTGC
3250b5e3a3495748	CAAGCCTTACCGCAGAGAC	CAGACCACACAAGGCAGATG
75d8b0106f385015	CCTTCGTGGACATCTTCGT	GTAGCAGGTGACTCAGGTTT
f31935d74b63fcb6	CAAGCCTTACCGCAGAGAC	GCAGACCACACAAGGCAGAT
bf78371688c47748	CAAGCCTTACCGCAGAGAC	GCAGACCACACAAGGCAGA
2a07aeaac84c9576	CCTTCGTGGACATCTTCGT	GGTGACTCAGGTTTTGCTG
3dddc7fb9f8e35a4	TCAAGCCTTACCGCAGAG	CACAAGGCAGATGGGCTAT
8bde0e8917e7305f	CCTTCGTGGACATCTTCGT	GGTGACTCAGGTTTTGCTGC
31b0c856466f7d5f	TCAAGCCTTACCGCAGAG	GACCACACAAGGCAGATG
d11ecc9c7bdc5645	TCAAGCCTTACCGCAGAG	GACCACACAAGGCAGATGG
bfd7573827850f81	TCAAGCCTTACCGCAGAG	ACCACACAAGGCAGATGG
10ca2d9422ab2af5	TCAAGCCTTACCGCAGAG	ACCACACAAGGCAGATGGG
91337f498097e9d9	TCAAGCCTTACCGCAGAG	CACAAGGCAGATGGGCTA

## Appendix C

# A Database for Metabarcoding Experiments

### C.1 Trigger Definition

Listing C.1: Trigger on insertion into table Experiment.

```

1 CREATE OR REPLACE FUNCTION sample_check() RETURNS trigger
2 AS $sample_check$
3 DECLARE
4     sample_id int;
5 BEGIN
6     FOREACH sample_id IN ARRAY NEW.sample_ids LOOP
7         IF sample_id NOT IN (SELECT id FROM Sample) THEN
8             RAISE EXCEPTION
9                 'sample_id_%_not_registered_in_Samples', sample_id;
10            END IF;
11        END LOOP;
12 END;
13 $sample_check$ LANGUAGE PLPGSQL;
14
15 CREATE TRIGGER sample_check
16 BEFORE INSERT OR UPDATE
17 ON Experiment
18 FOR EACH ROW EXECUTE PROCEDURE sample_check();

```

### C.2 Example Queries in SQL

**How frequently are all sites sampled?**

```

1 SELECT location.long_name as site, collection_date as date
2 FROM sample, location
3 WHERE sample.location_id = location.short_name
4 ORDER BY site ASC

```

**Which primer pairs have been tested so far?**

```

1 SELECT * FROM primer_pair, pcr_experiment
2 WHERE primer_pair.id = pcr_experiment.primers

```

**Where is the data set located that was used in study X?**

```

1 SELECT sequences.server, sequences.path
2 FROM publication, pipeline, sequences
3 WHERE publication.pipeline_id = pipeline.id
4 AND pipeline.sequences_id = X

```

### Which tools have been used in previous studies to build a bioinformatics pipeline?

```

1 SELECT publication.description, publication.doi, pipeline.protocol
2 FROM publication, pipeline
3 WHERE publication.pipeline_id = pipeline.id

```

### How many different DNA extraction kits are in use since last year?

```

1 SELECT DISTINCT(T.exkit) FROM
2 (
3 SELECT extraction_kit.id as exkit, date_part('year', experiment.date)
4     as year
5 FROM experiment, extraction_kit, pcr_experiment
6 WHERE experiment.id = pcr_experiment.experiment_id
7 AND experiment.id = pcr_experiment.experiment_id
8 ) as T
9 WHERE T.year >= 2019

```

### What was the average number of OTUs a primer pair X produced given identical sample treatment?

```

1 SELECT sample.filter_method, sample.dry_method,
2     AVG(CARDINALITY(pipeline.otu_ids))
3 FROM sample, experiment, pcr_experiment, sequences, pipeline
4 WHERE sample.id = ANY(experiment.sample_ids)
5 AND pcr_experiment.experiment_id = experiment.id
6 AND pcr_experiment.primers_pair = X
7 AND sequences.experiment_id = experiment.id
8 AND pipeline.id = sequences.experiment_id
9 GROUP BY sample.filter_method, sample.dry_method

```

### Which primer pair is most effective on the group of Cryptophyta in terms of the number of OTUs?

```

1 SELECT primer_pair.name, COUNT(otu.id) as otu_count
2 FROM pcr_experiment, primer_pair, sequences, pipeline, otu, taxon, lineage
3 WHERE pcr_experiment.primers_pair = primer_pair.id
4 AND sequences.experiment_id = pcr_experiment.experiment_id
5 AND pipeline.sequences_id = sequences.experiment_id
6 AND otu.id = ANY(pipeline.otu_ids)
7 AND otu.taxon_id = taxon.id
8 AND 3027 = ANY (lineage.grand_taxon_ids)
9 GROUP BY pcr_experiment.experiment_id, primer_pair.id
10 ORDER BY otu_count

```



**Which plankton samples underwent both microscopic and metabarcoding analysis?**

```
1 SELECT sample.id
2 FROM sample, experiment, pcr_experiment
3 WHERE sample.id = ANY(experiment.sample_ids)
4 AND experiment.id = pcr_experiment.experiment_id
5 AND sample.id = ANY
6 (
7     SELECT sample.id
8     FROM sample, experiment, morph_experiment
9     WHERE sample.id = ANY(experiment.sample_ids)
10    AND experiment.id = morph_experiment.experiment_id
11 )
```

**What is the average number of reads produced when using freeze-drying versus airdrying as a sample treatment?**

```
1 SELECT sample.dry_method, AVG(CARDINALITY(pipeline.otu_ids)) as avg
2 FROM sample, experiment, pcr_experiment, sequences, pipeline, dry_method
3 WHERE sample.id = ANY(experiment.sample_ids)
4 AND sequences.experiment_id = experiment.id
5 AND pipeline.sequences_id = sequences.experiment_id
6 AND dry_method.id = sample.dry_method
7 AND dry_method.short_name = ANY('{fd,_ad}')
8 GROUP BY sample.dry_method
```



# Zusammenfassung

Die vorliegende Arbeit behandelt einige wichtige Herausforderungen von Metabarcoding-Experimenten von Umweltproben. Diese haben ihren Ursprung in der phylogenetischen Heterogenität der Probe, der grundsätzlichen Unwissenheit der darin enthaltenen Organismen und der nur mäßigen Verfügbarkeit von Referenzsequenzen zur Identifizierung. Grundsätzliches Ziel ist es möglichst viele Organismen genau zu identifizieren, d.h. bis auf die Ebene von Spezies.

Konkret handelt Kapitel 2 von der Schwierigkeit eine verlässliche Taxonomie aufzustellen, sowohl auf der Grundlage von morphologischen Merkmalen, als auch auf der Grundlage von DNA-Sequenz-Ähnlichkeiten. Dies hat Implikationen für jede Art von Identifizierungsmethode, da sie entlang der hierarchisch organisierten Taxonomie stattfindet. Es wird beispielhaft eine Studie ausgeführt, die im Rahmen eines Monitoring-Projektes entstanden ist. Zu den Aufgaben des Projektes gehört, die Plankton-Zusammensetzung eines nahe gelegenen Frischwasser-Biotops zu bestimmen.

In Kapitel 3 wird das Primer Search Tool PriSeT vorgestellt, für welches die einführende Studie der Katalysator war. PriSeT berechnet neue Primerpaare auf beliebigen, nicht-kuratierten DNA-Referenz-Biobibliotheken, indem es häufig vorkommende  $k$ -mere auf Primer-Tauglichkeit testet und diese zu Paaren kombiniert. Die errechneten Primerpaare sind sortierbar nach dem Kriterium taxonomischer Abdeckungsrate oder Barcode-Variabilität. Es gibt aktuell kein quelloffenes Programm, was hierzu in der Lage wäre. Alle bekannten Primersuch-Werkzeuge sind ungeeignet für ein Metabarcoding-Setting, d.h. sie sind beschränkt auf eine oder sehr wenige Referenzen als Eingabe und in keinem Fall in der Lage hochfrequente Paare erkennen – eine erste Notwendigkeit, um eine hohe taxonomische Abdeckungsrate zu erzielen. PriSeT ist derart robust, dass es nicht-kuratierte Datensätze bestehend aus mehreren hunderttausend Referenzen oder auch komplette Genome zur Suche verwenden kann. Die hohe Rechen- und Speicherintensität wird reduziert durch Indizierung der Bibliothek, Speicherung von mehreren  $k$ -meren in einem einzigen 64-bit Datentypen, Bit-Parallelisierung bei der Verifizierung chemischer Eigenschaften, und Transformation der Referenzen zu Bitvektoren mit Rank- und Select-Support in  $\mathcal{O}(1)$ .

Kapitel 4 stellt einen Lösungsvorschlag zur Optimierung des Arbeitsflusses zwischen den an Metagenomik-Experimenten teilnehmenden Forschern vor. Es werden die Schwierigkeiten aus der Sicht eines Datenanalytisten analysiert, der vergleichsweise kurz in ein Projekt involviert ist und schnell einen Überblick gewinnen muss, um produktiv arbeiten zu können. Ein Datenbank-Schema wird vorgestellt, das die wichtigsten Daten konsolidiert und Meta-Analysen erlaubt, die bisher nahezu unmöglich waren. Es löst die häufigsten Probleme im Zusammenhang mit Daten-Beschaffung, Redundanz, Nicht-Versionierung und Übergabe von Projekte an nachfolgende Wissenschaftler. Ein Prototyp wurde den involvierten Arbeitsgruppen vorgestellt.



# Bibliography

- Adrian, Rita et al. (2009). “Lakes as sentinels of climate change”. In: *Limnology and Oceanography* 54.6, pp. 2283–2297. doi: [10.4319/lo.2009.54.6](https://doi.org/10.4319/lo.2009.54.6).
- Albaina, Aitor et al. (2016). “18S rRNA V9 metabarcoding for diet characterization: a critical evaluation with two sympatric zooplanktivorous fish species”. In: *Ecology and Evolution* 6.6, pp. 1809–1824. doi: [10.1002/ece3.1986](https://doi.org/10.1002/ece3.1986).
- Baker, Monya (2010). “Next-generation sequencing: adjusting to data overload”. In: *Nature Methods* 7, pp. 495–499. doi: [10.1038/nmeth0710-495](https://doi.org/10.1038/nmeth0710-495).
- Benson, Dennis A. et al. (Nov. 2012). “GenBank”. In: *Nucleic Acids Research* 41.D1, pp. D36–D42. issn: 0305-1048. doi: [10.1093/nar/gks1195](https://doi.org/10.1093/nar/gks1195).
- Berman, L. and J. Hartmanis (1977). “On Isomorphisms and Density of NP and Other Complete Sets”. In: *SIAM Journal on Computing* 6.2, pp. 305–322. doi: [10.1137/0206023](https://doi.org/10.1137/0206023).
- Birk, Sebastian et al. (2012). “Three hundred ways to assess Europe’s surface waters: An almost complete overview of biological methods to implement the Water Framework Directive”. In: *Ecological Indicators* 18, pp. 31–41. doi: [10.1016/j.ecolind.2011.10.009](https://doi.org/10.1016/j.ecolind.2011.10.009).
- Blaxter, Mark et al. (Mar. 1998). “A molecular evolutionary framework for the phylum Nematoda”. In: *Nature* 392, pp. 71–75. doi: [10.1038/32160](https://doi.org/10.1038/32160).
- Boscaro, Vittorio et al. (2017). “Strengths and Biases of High-Throughput Sequencing Data in the Characterization of Freshwater Ciliate Microbiomes”. In: *Microbial Ecology* 73 (4), pp. 865–875. doi: [10.1007/s00248-016-0912-8](https://doi.org/10.1007/s00248-016-0912-8).
- Breslauer, Kenneth et al. (July 1986). “Predicting DNA Duplex Stability from the Base Sequence”. In: *Proceedings of the National Academy of Sciences of the United States of America* 83, pp. 3746–3750. doi: [10.1073/pnas.83.11.3746](https://doi.org/10.1073/pnas.83.11.3746).
- Buchheim, Mark et al. (Sept. 2005). “Phylogeny of the Hydrodictyaceae (Chlorophyceae): inferences from rDNA data”. In: *Journal of Phycology* 41, pp. 1039–1054. doi: [10.1111/j.1529-8817.2005.00129.x](https://doi.org/10.1111/j.1529-8817.2005.00129.x).
- Burki, Fabien et al. (2020). “The New Tree of Eukaryotes”. In: *Trends in Ecology & Evolution* 35 (1), pp. 43–55. doi: [10.1016/j.tree.2019.08.008](https://doi.org/10.1016/j.tree.2019.08.008).
- Burrows, Michael and David Wheeler (1994). *A Block-Sorting Lossless Data Compression Algorithm*. Tech. rep. Digital Equipment Corporation. doi: [10.1.1.37.6774](https://doi.org/10.1.1.37.6774).
- Bustin, S. A. (2002). “Quantification of mRNA using real-time reverse transcription PCR (RT-PCR): trends and problems”. In: *Journal of Molecular Endocrinology* 29.1, pp. 23–39. doi: [10.1677/jme.0.0290023](https://doi.org/10.1677/jme.0.0290023).
- Cannon, Matthew et al. (Dec. 2018). “A high-throughput sequencing assay to comprehensively detect and characterize unicellular eukaryotes and helminths from biological and environmental samples”. In: *Microbiome* 6. doi: [10.1186/s40168-018-0581-6](https://doi.org/10.1186/s40168-018-0581-6).
- Cantino, Philip D., Kevin De Queiroz, et al. (2020). *PhyloCode: a phylogenetic code of biological nomenclature*. doi: [10.1201/9780429446320](https://doi.org/10.1201/9780429446320).
- Carmel, Erran, Yael Dubinsky, and Alberto J. Espinosa (2009). “Follow The Sun Software Development: New Perspectives, Conceptual Foundation, and Exploratory

- Field Study". In: *2009 42nd Hawaii International Conference on System Sciences*, pp. 1–9. doi: [10.1109/HICSS.2009.218](https://doi.org/10.1109/HICSS.2009.218).
- Carrillo, Humberto and David Lipman (Oct. 1988). "The Multiple Sequence Alignment Problem in Biology". In: *SIAM J. Appl. Math.* 48.5, pp. 10731082. issn: 0036-1399. doi: [10.1137/0148063](https://doi.org/10.1137/0148063).
- Catherine, Brooksbank, Graham Cameron, and Thornton Janet (2010). "The European Bioinformatics Institute's data resources". In: *Nucleic Acids Research* 38, pp. D17–D25. doi: [10.1093/nar/gkp986](https://doi.org/10.1093/nar/gkp986).
- Ceballos, Gerardo, Paul R. Ehrlich, and Rodolfo Dirzo (2017). "Biological annihilation via the ongoing sixth mass extinction signaled by vertebrate population losses and declines". In: *Proceedings of the National Academy of Sciences* 114.30, E6089–E6096. doi: [10.1073/pnas.1704949114](https://doi.org/10.1073/pnas.1704949114).
- Chamberlin, D. (1998). *A Complete Guide to DB2 Universal Database*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science. isbn: 9781558604827.
- Chien, A. L., D.B. Edgar, and J. M. Trela (1976). "Deoxyribonucleic acid polymerase from the extreme thermophile *Thermus aquaticus*". In: *Journal of Bacteriology* 127.3, pp. 1550–1557. doi: [10.1128/jb.127.3.1550-1557.1976](https://doi.org/10.1128/jb.127.3.1550-1557.1976).
- Clark, David Richard (1998). "Compact Pat Trees". PhD thesis. Ontario, Canada.
- Clarke, L. A. et al. (Oct. 2001). "PCR amplification introduces errors into mononucleotide and dinucleotide repeat sequences". In: *Molecular Pathology* 54, pp. 351–353. doi: [10.1136/mp.54.5.351](https://doi.org/10.1136/mp.54.5.351).
- Clarke, Laurence J. et al. (2017). "Effect of marker choice and thermal cycling protocol on zooplankton DNA metabarcoding studies". In: *Ecology and Evolution* 7.3, pp. 873–883. doi: [10.1002/ece3.2667](https://doi.org/10.1002/ece3.2667).
- Codd, Edgar F. (June 1970). "A Relational Model of Data for Large Shared Data Banks". In: *Commun. ACM* 13.6, pp. 377387. issn: 0001-0782. doi: [10.1145/362384.362685](https://doi.org/10.1145/362384.362685).
- Cook, Isabelle, Sam Grange, and Adam Eyre-Walker (2015). "Research groups: How big should they be?" In: *PeerJ* 3, e989. doi: [10.7717/peerj.989](https://doi.org/10.7717/peerj.989).
- De Queiroz, Kevin (Dec. 2007). "Species Concepts and Species Delimitation". In: *Systematic Biology* 56.6, pp. 879–886. issn: 1063-5157. doi: [10.1080/10635150701701083](https://doi.org/10.1080/10635150701701083).
- Derrien, Thomas et al. (Jan. 2012). "Fast Computation and Applications of Genome Mappability". In: *PLOS ONE* 7.1, pp. 1–16. doi: [10.1371/journal.pone.0030377](https://doi.org/10.1371/journal.pone.0030377).
- DeSantis, Todd Z. et al. (2007). "High-density universal 16S rRNA microarray analysis reveals broader diversity than typical clone library when sampling the environment". In: *Microbial Ecology* 53, pp. 371383. doi: [10.1007/s00248-006-9134-9](https://doi.org/10.1007/s00248-006-9134-9).
- Dogan, Ismet and Nurhan Dogan (Apr. 2016). "Genetic Distance Measures: Review". In: *Turkiye Klinikleri Journal of Biostatistics* 8, pp. 87–93. doi: [10.5336/biostat.2015-49517](https://doi.org/10.5336/biostat.2015-49517).
- Driescher, Eva et al. (1993). "Lake Müggelsee and its Environment Natural Conditions and Anthropogenic Impacts". In: *Internationale Revue der gesamten Hydrobiologie und Hydrographie* 78.3, pp. 327–343. doi: [10.1002/iroh.19930780303](https://doi.org/10.1002/iroh.19930780303).
- Dudgeon, David et al. (2006). "Freshwater biodiversity: importance, threats, status and conservation challenges". In: *Biological Reviews* 81.2, pp. 163–182. doi: [10.1017/S1464793105006950](https://doi.org/10.1017/S1464793105006950).
- Edgar, Robert C. (Mar. 2004). "MUSCLE: multiple sequence alignment with high accuracy and high throughput". In: *Nucleic Acids Research* 32.5, pp. 1792–1797. doi: [10.1093/nar/gkh340](https://doi.org/10.1093/nar/gkh340).

- (Aug. 2010). “Search and clustering orders of magnitude faster than BLAST”. In: *Bioinformatics* 26.19, pp. 2460–2461. issn: 1367-4803. doi: [10.1093/bioinformatics/btq461](https://doi.org/10.1093/bioinformatics/btq461).
- Edwards, A. W. (1971). “Distances between populations on the basis of gene frequencies”. In: *Biometrics* 27 (4), pp. 873–881.
- Elbrecht, Vasco et al. (2019). “Validation of COI metabarcoding primers for terrestrial arthropods”. In: *PeerJ* 7. doi: [10.7717/peerj.7745](https://doi.org/10.7717/peerj.7745).
- Elias, Isaac (Oct. 2006). “Settling the Intractability of Multiple Alignment”. In: *Journal of computational biology : a journal of computational molecular cell biology* 13, pp. 1323–39. doi: [10.1089/cmb.2006.13.1323](https://doi.org/10.1089/cmb.2006.13.1323).
- Farris, James S. (2008). “Parsimony and explanatory power”. In: *Cladistics* 24.5, pp. 825–847. doi: [10.1111/j.1096-0031.2008.00214.x](https://doi.org/10.1111/j.1096-0031.2008.00214.x).
- Felsenstein, Joseph (1981). “Evolutionary trees from DNA sequences: a maximum likelihood approach”. In: *Journal of molecular evolution* 17.6, pp. 368–376.
- (May 2004). *Inferring Phylogenies*. Vol. 2. Sinauer associates Sunderland, MA.
- Ferragina, Paolo and Giovanni Manzini (Feb. 2000). “Opportunistic Data Structures with Applications”. In: vol. 2000, pp. 390–398. isbn: 0-7695-0850-2. doi: [10.1109/SFCS.2000.892127](https://doi.org/10.1109/SFCS.2000.892127).
- (July 2005). “Indexing Compressed Text”. In: *J. ACM* 52.4, pp. 552–581. doi: [10.1145/1082036.1082039](https://doi.org/10.1145/1082036.1082039).
- Fischer, Johannes and Florian Kurpicz (2017). “Dismantling DivSufSort”. In: *ArXiv abs/1710.01896*.
- Fitch, Walter M. (Dec. 1971). “Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology”. In: *Systematic Biology* 20.4, pp. 406–416. issn: 1063-5157. doi: [10.1093/sysbio/20.4.406](https://doi.org/10.1093/sysbio/20.4.406).
- Forgie, Sarah and Thomas J. Marrie (2009). “Healthcare-Associated Atypical Pneumonia”. In: *Seminars in Respiratory and Critical Care Medicine* 30 (1), pp. 67–85. doi: [10.1055/s-0028-1119811](https://doi.org/10.1055/s-0028-1119811).
- Gleick, Peter H. (1996). “Basic Water Requirements for Human Activities: Meeting Basic Needs”. In: *Water International* 21.2, pp. 83–92. doi: [10.1080/02508069608686494](https://doi.org/10.1080/02508069608686494).
- Gog, Simon et al. (2014). “From Theory to Practice: Plug and Play with Succinct Data Structures”. In: *13th International Symposium on Experimental Algorithms, (SEA 2014)*, pp. 326–337. doi: [10.1007/978-3-319-07959-2\\_28](https://doi.org/10.1007/978-3-319-07959-2_28).
- Gonnet, Gaston H and Ricardo A Baeza-Yates (1992). “New Indices for Text: Pat Trees and Pat Arrays.” In: *Information Retrieval: Data Structures & Algorithms* 66, p. 82.
- Gray, Jim (Sept. 1981). “The Transaction Concept: Virtues and Limitations”. In: *Seventh International Conference on Very Large Data Bases*, pp. 144–154.
- Groendahl, Sophie, Maria Kahlert, and Patrick Fink (Feb. 2017). “The best of both worlds: A combined approach for analyzing microalgal diversity via metabarcoding and morphology-based methods”. In: *PLOS ONE* 12.2, pp. 1–15. doi: [10.1371/journal.pone.0172808](https://doi.org/10.1371/journal.pone.0172808).
- Guillou, Laure et al. (Nov. 2012). “The Protist Ribosomal Reference database (PR2): a catalog of unicellular eukaryote Small Sub-Unit rRNA sequences with curated taxonomy”. In: *Nucleic Acids Research* 41.D1, pp. D597–D604. doi: [10.1093/nar/gks1160](https://doi.org/10.1093/nar/gks1160).
- Hadziavdic, Kenan et al. (Feb. 2014). “Characterization of the 18S rRNA Gene for Designing Universal Eukaryote Specific Primers”. In: *PLOS ONE* 9.2, pp. 1–10. doi: [10.1371/journal.pone.0087624](https://doi.org/10.1371/journal.pone.0087624).

- Harder, Christoffer et al. (Mar. 2016). “Local diversity of heathland Cercozoa explored by in-depth sequencing”. In: *The ISME Journal* 10. doi: [10.1038/ismej.2016.31](https://doi.org/10.1038/ismej.2016.31).
- Hautmann, Michael (2020). “What is macroevolution?” In: *Palaeontology* 63.1, pp. 1–11. doi: [10.1111/pala.12465](https://doi.org/10.1111/pala.12465).
- Hoffmann, Marie, Michael T. Monaghan, and Knut Reinert (2021). “PriSeT: Efficient de Novo Primer Discovery”. In: *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. BCB '21. New York, NY, USA: Association for Computing Machinery. isbn: 9781450384506. doi: [10.1145/3459930.3469546](https://doi.org/10.1145/3459930.3469546).
- Hufsky, Franziska et al. (Nov. 2020). “Computational Strategies to Combat COVID-19: Useful Tools to Accelerate SARS-CoV-2 and Coronavirus Research”. In: *Briefings in Bioinformatics* 22.2, pp. 642–663. issn: 1477-4054. doi: [10.1093/bib/bbaa232](https://doi.org/10.1093/bib/bbaa232).
- Hug, Laura A. et al. (2016). “A new view of the tree of life”. In: *Nature Microbiology* 1 (5). doi: [10.1038/nmicrobiol.2016.48](https://doi.org/10.1038/nmicrobiol.2016.48).
- Jia, Ben et al. (2019). “GLAPD: Whole Genome Based LAMP Primer Design for a Set of Target Genomes”. In: *Frontiers in Microbiology* 10, p. 2860. issn: 1664-302X. doi: [10.3389/fmicb.2019.02860](https://doi.org/10.3389/fmicb.2019.02860).
- Jumper, J. et al. (2021). “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596, pp. 583–589. doi: [10.1038/s41586-021-03819-2](https://doi.org/10.1038/s41586-021-03819-2).
- Kalendar, Ruslan et al. (2017). “FastPCR: An in silico tool for fast primer and probe design and advanced sequence analysis”. In: *Genomics* 109.3, pp. 312–319. issn: 0888-7543. doi: [10.1016/j.ygeno.2017.05.005](https://doi.org/10.1016/j.ygeno.2017.05.005).
- Kärkkäinen, Juha, Peter Sanders, and Stefan Burkhardt (2006). “Linear Work Suffix Array Construction”. In: *J. ACM* 53.6, pp. 918–936. issn: 0004-5411. doi: [10.1145/1217856.1217858](https://doi.org/10.1145/1217856.1217858).
- Kim, Jin, Sakti Pramanik, and Moon Jung Chung (July 1994). “Multiple sequence alignment using simulated annealing”. In: *Bioinformatics* 10.4, pp. 419–426. issn: 1367-4803. doi: [10.1093/bioinformatics/10.4.419](https://doi.org/10.1093/bioinformatics/10.4.419).
- Kleppmann, Martin (2017). *Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems*. O’Reilly Media, Inc., p. 569. isbn: 9781491903063.
- Kämpke, Thomas, Markus Kieninger, and Michael Mecklenburg (Apr. 2001). “Efficient primer design algorithms”. In: *Bioinformatics (Oxford, England)* 17, pp. 214–25. doi: [10.1093/bioinformatics/17.3.214](https://doi.org/10.1093/bioinformatics/17.3.214).
- Lermen, Martin and Knut Reinert (2000). “The Practical Use of the A\* Algorithm for Exact Multiple Sequence Alignment”. In: *Journal of Computational Biology* 7.5. PMID: 11153092, pp. 655–671. doi: [10.1089/106652701446134](https://doi.org/10.1089/106652701446134).
- Lexa, Matej and Giorgio Valle (Jan. 2004). “PRIMEX: Rapid identification of oligonucleotide matches in whole genomes”. In: *Bioinformatics (Oxford, England)* 19, pp. 2486–8. doi: [10.1093/bioinformatics/btg350](https://doi.org/10.1093/bioinformatics/btg350).
- Li, Chunxiang, Alan Medlar, and Ari Löytynoja (2016). “Co-estimation of Phylogeny-aware Alignment and Phylogenetic Tree”. In: doi: [10.1101/077503](https://doi.org/10.1101/077503).
- Lindahl, Björn D. et al. (2013). “Fungal community analysis by high-throughput sequencing of amplified markers a user’s guide”. In: *New Phytologist* 199.1, pp. 288–299. doi: [10.1111/nph.12243](https://doi.org/10.1111/nph.12243).
- Lindeque, Penelope K. et al. (Nov. 2013). “Next Generation Sequencing Reveals the Hidden Diversity of Zooplankton Assemblages”. In: *PLOS ONE* 8.11. doi: [10.1371/journal.pone.0081327](https://doi.org/10.1371/journal.pone.0081327).



- Liu, Yuansheng et al. (June 2017). “High-speed and high-ratio referential genome compression”. In: *Bioinformatics* 33.21, pp. 3364–3372. doi: [10.1093/bioinformatics/btx412](https://doi.org/10.1093/bioinformatics/btx412).
- Löytynoja, Ari (2014). “Phylogeny-aware alignment with PRANK”. In: *Methods in Molecular Biology* (1079), pp. 155–170. doi: [10.1007/978-1-62703-646-7\\_10](https://doi.org/10.1007/978-1-62703-646-7_10).
- Löytynoja, Ari, Albert J. Vilella, and Nick Goldman (Apr. 2012). “Accurate extension of multiple sequence alignments using a phylogeny-aware graph algorithm”. In: *Bioinformatics* 28.13, pp. 1684–1691. issn: 1367-4803. doi: [10.1093/bioinformatics/bts198](https://doi.org/10.1093/bioinformatics/bts198).
- Maier, David (1978). “The Complexity of Some Problems on Subsequences and Supersequences”. In: *Journal of the ACM* 25, pp. 322–336. doi: [10.1145/322063.322075](https://doi.org/10.1145/322063.322075).
- Makino, Wataru et al. (Apr. 2017). “DNA barcoding of freshwater zooplankton in Lake Kasumigaura, Japan”. In: *Ecological Research* 32. doi: [10.1007/s11284-017-1458-z](https://doi.org/10.1007/s11284-017-1458-z).
- Manber, Udi and Gene Myers (1990). “Suffix Arrays: A New Method for on-Line String Searches”. In: *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA 90. San Francisco, California, USA: Society for Industrial and Applied Mathematics, pp. 319327. isbn: 0898712513.
- Margulies, Marcel et al. (Oct. 2005). “Genome Sequencing in Microfabricated High-Density Picolitre Reactors”. In: *Nature* 437, pp. 376–80. doi: [10.1038/nature03959](https://doi.org/10.1038/nature03959).
- Mayden, Richard L. (1997). “A hierarchy of species concepts: the denouement in the saga of the species problem”. In: *Species: the units of biodiversity*. London: Chapman & Hall. Chap. 19.
- Medlin, Linda et al. (1988). “The characterization of enzymatically amplified eukaryotic 16S-like rRNA-coding regions”. In: *Gene* 71.2, pp. 491–499. issn: 0378-1119. doi: [10.1016/0378-1119\(88\)90066-2](https://doi.org/10.1016/0378-1119(88)90066-2).
- Middendorf, Martin (1994). “More on the complexity of common superstring and supersequence problems”. In: *Theoretical Computer Science* 125.2, pp. 205–228.
- Mischke, Ute and Horst Behrendt (Mar. 2007). *Handbuch zum Bewertungsverfahren von Fließgewässern mittels Phytoplankton zur Umsetzung der EU-WRRL in Deutschland*. Stuttgart, Germany: Schweizerbart Science Publishers.
- Mohrbeck, Inga et al. (Oct. 2015). “High-Throughput Sequencing The Key to Rapid Biodiversity Assessment of Marine Metazoa?” In: *PLOS ONE* 10.10, pp. 1–24. doi: [10.1371/journal.pone.0140342](https://doi.org/10.1371/journal.pone.0140342).
- Moreno, Y. et al. (2018). “Multiple identification of most important waterborne protozoa in surface water used for irrigation purposes by 18S rRNA amplicon-based metagenomics”. In: *International Journal of Hygiene and Environmental Health* 221.1, pp. 102–111. issn: 1438-4639. doi: [10.1016/j.ijheh.2017.10.008](https://doi.org/10.1016/j.ijheh.2017.10.008).
- Moro, Claire Valiente et al. (2009). “New Design Strategy for Development of Specific Primer Sets for PCR-Based Detection of Chlorophyceae and Bacillariophyceae in Environmental Samples”. In: *Applied and Environmental Microbiology* 75.17, pp. 5729–5733. issn: 0099-2240. doi: [10.1128/AEM.00509-09](https://doi.org/10.1128/AEM.00509-09).
- Murphy, Douglas B. and Michael W. Davidson (2012). “Fundamentals of Light Microscopy”. In: *Fundamentals of Light Microscopy and Electronic Imaging*. John Wiley & Sons, Ltd. Chap. 1, pp. 1–19. isbn: 9781118382905. doi: <https://doi.org/10.1002/9781118382905.ch1>.

- Nakamura, Tsukasa et al. (Mar. 2018). “Parallelization of MAFFT for large-scale multiple sequence alignments”. In: *Bioinformatics* 34.14, pp. 2490–2492. issn: 1367-4803. doi: [10.1093/bioinformatics/bty121](https://doi.org/10.1093/bioinformatics/bty121).
- Nakov, Teofil, Jeremy M. Beaulieu, and Andrew J. Alverson (2018). “Insights into global planktonic diatom diversity: The importance of comparisons between phylogenetically equivalent units that account for time”. In: *The ISME Journal* 12 (11), pp. 2807–2810. doi: [10.1038/s41396-018-0221-y](https://doi.org/10.1038/s41396-018-0221-y).
- Navarro, Gonzalo (2016). *Compact Data Structures: A Practical Approach*. Cambridge University Press. doi: [10.1017/CBO9781316588284](https://doi.org/10.1017/CBO9781316588284).
- Needleman, Saul B. and Christian D. Wunsch (1970). “A general method applicable to the search for similarities in the amino acid sequence of two proteins”. In: *Journal of Molecular Biology* 48.3, pp. 443–453. issn: 0022-2836. doi: [10.1016/0022-2836\(70\)90057-4](https://doi.org/10.1016/0022-2836(70)90057-4).
- Nei, Masatoshi (1972). “Genetic Distance between Populations”. In: *The American Naturalist* 106.949, pp. 283–292. doi: [10.1086/282771](https://doi.org/10.1086/282771).
- Nong, Ge, Sen Zhang, and Daricks Wai Hong Chan (Mar. 2009). “Linear Suffix Array Construction by Almost Pure Induced-Sorting”. In: pp. 193–202. doi: [10.1109/DCC.2009.42](https://doi.org/10.1109/DCC.2009.42).
- Notomi, Tsugunori et al. (June 2000). “Loop-mediated isothermal amplification of DNA”. In: *Nucleic Acids Research* 28.12, e63–e63. issn: 0305-1048. doi: [10.1093/nar/28.12.e63](https://doi.org/10.1093/nar/28.12.e63).
- Notredame, Cédric, Desmond G. Higgins, and Jaap Heringa (2000). “T-coffee: a novel method for fast and accurate multiple sequence alignment”. In: *Journal of Molecular Biology* 302.1, pp. 205–217. issn: 0022-2836. doi: [10.1006/jmbi.2000.4042](https://doi.org/10.1006/jmbi.2000.4042).
- Nuin, Paulo A. S., Zhouzhi Wang, and Elisabeth R. M. Tillier (2006). “The accuracy of several multiple sequence alignment programs for proteins”. In: *BMC bioinformatics* 7 (1), p. 471. doi: [10.1186/1471-2105-7-471](https://doi.org/10.1186/1471-2105-7-471).
- Ogura, Atsushi et al. (2018). *Comparative genome and transcriptome analysis of diatom, Skeletonema costatum, reveals evolution of genes for harmful algal bloom*. doi: [10.6084/m9.figshare.c.4275500.v1](https://doi.org/10.6084/m9.figshare.c.4275500.v1).
- Oksanen, J. et al. (2012). *vegan: Community Ecology Package*. Version 2.5-6.
- Otu, Hasan and K. Sayood (Dec. 2003). “A New Sequence Distance Measure for Phylogenetic Tree Construction”. In: *Bioinformatics (Oxford, England)* 19, pp. 2122–30. doi: [10.1093/bioinformatics/btg295](https://doi.org/10.1093/bioinformatics/btg295).
- Pawlowski, Jan et al. (Nov. 2012). “CBOL Protist Working Group: Barcoding Eukaryotic Richness beyond the Animal, Plant, and Fungal Kingdoms”. In: *PLOS Biology* 10.11, pp. 1–5. doi: [10.1371/journal.pbio.1001419](https://doi.org/10.1371/journal.pbio.1001419).
- Pockrandt, Christopher, Marcel Ehrhardt, and Knut Reinert (2017). “EPR-Dictionaries: A Practical and Fast Data Structure for Constant Time Searches in Unidirectional and Bidirectional FM Indices”. In: *21st Annual International Conference on Research in Computational Molecular Biology - Volume 10229*. RECOMB 2017. Berlin, Heidelberg: Springer-Verlag, pp. 190–206. isbn: 9783319569697. doi: [10.1007/978-3-319-56970-3\\_12](https://doi.org/10.1007/978-3-319-56970-3_12).
- Pockrandt, Christopher et al. (Apr. 2020). “GenMap: ultra-fast computation of genome mappability”. In: *Bioinformatics* 36.12, pp. 3687–3692. issn: 1367-4803. doi: [10.1093/bioinformatics/btaa222](https://doi.org/10.1093/bioinformatics/btaa222).
- Quast, Christian et al. (2012). “The SILVA ribosomal RNA gene database project: improved data processing and web-based tools”. In: *Nucleic Acids Research* 41.D1, pp. D590–D596. issn: 0305-1048. doi: [10.1093/nar/gks1219](https://doi.org/10.1093/nar/gks1219).

- Rappé, Michael S. and Stephen J. Giovannoni (2003). "The Uncultured Microbial Majority". In: *Annual Review of Microbiology* 57.1. PMID: 14527284, pp. 369–394. doi: [10.1146/annurev.micro.57.030502.090759](https://doi.org/10.1146/annurev.micro.57.030502.090759).
- Reijns, Martin A. M. et al. (Dec. 2020). "A sensitive and affordable multiplex RT-qPCR assay for SARS-CoV-2 detection". In: *PLOS Biology* 18.12, pp. 1–20. doi: [10.1371/journal.pbio.3001030](https://doi.org/10.1371/journal.pbio.3001030).
- Rimet, Frédéric et al. (May 2014). "When is sampling complete? The effects of geographical range and marker choice on perceived diversity in *Nitzschia palea* (Bacillariophyta)". In: *Protist* 165.3, pp. 245–259. doi: [10.1016/j.protis.2014.03.005](https://doi.org/10.1016/j.protis.2014.03.005).
- Ritter, Camila D. et al. (2019). "The pitfalls of biodiversity proxies: Differences in richness patterns of birds, trees and understudied diversity across Amazonia". In: *Scientific Reports* 9. doi: [10.1038/s41598-019-55490-3](https://doi.org/10.1038/s41598-019-55490-3).
- Saitou, N and M Nei (July 1987). "The neighbor-joining method: a new method for reconstructing phylogenetic trees." In: *Molecular Biology and Evolution* 4.4, pp. 406–425. issn: 0737-4038. doi: [10.1093/oxfordjournals.molbev.a040454](https://doi.org/10.1093/oxfordjournals.molbev.a040454).
- Sanghvi, L. D. (1953). "Comparison of genetical and morphological methods for a study of biological differences". In: *American Journal of Physical Anthropology* 11 (3), pp. 385–404. doi: [doi:10.1002/ajpa.1330110313](https://doi.org/10.1002/ajpa.1330110313).
- Schmidt, Thomas, João Rodrigues, and Christian von Mering (Apr. 2014). "Ecological Consistency of SSU rRNA-Based Operational Taxonomic Units at a Global Scale". In: *PLoS computational biology* 10, e1003594. doi: [10.1371/journal.pcbi.1003594](https://doi.org/10.1371/journal.pcbi.1003594).
- Segata, Nicola et al. (2013). "Computational metaomics for microbial community studies". In: 9, pp. 666–666. doi: [10.1038/msb.2013.22](https://doi.org/10.1038/msb.2013.22).
- Smart, Adam S. et al. (2015). "Environmental DNA sampling is more sensitive than a traditional survey technique for detecting an aquatic invader". In: *Ecological Applications* 25.7, pp. 1944–1952. doi: [10.1890/14-1751.1](https://doi.org/10.1890/14-1751.1).
- Smith, David J. (July 2013). "Aeroplankton and the Need for a Global Monitoring Network". In: *BioScience* 63.7, pp. 515–516. issn: 0006-3568. doi: [10.1525/bio.2013.63.7.3](https://doi.org/10.1525/bio.2013.63.7.3).
- Smith, Kirsty F. et al. (2017). "Assessment of the metabarcoding approach for community analysis of benthic-epiphytic dinoflagellates using mock communities". In: *New Zealand Journal of Marine and Freshwater Research* 51.4, pp. 555–576. doi: [10.1080/00288330.2017.1298632](https://doi.org/10.1080/00288330.2017.1298632).
- Song, Zhiqi et al. (2019). "From SARS to MERS, Thrusting Coronaviruses into the Spotlight". In: *Viruses* 11. doi: [10.3390/v11010059](https://doi.org/10.3390/v11010059).
- Stein, Lincoln D. (2010). "The case for cloud computing in genome informatics". In: *Genome Biology* 11 (5). doi: [10.1186/gb-2010-11-5-207](https://doi.org/10.1186/gb-2010-11-5-207).
- Stoeck, Thorsten et al. (2010). "Multiple marker parallel tag environmental DNA sequencing reveals a highly complex eukaryotic community in marine anoxic water". In: *Molecular Ecology* 19.s1, pp. 21–31. doi: [10.1111/j.1365-294X.2009.04480.x](https://doi.org/10.1111/j.1365-294X.2009.04480.x).
- Stojmirović, Aleksandar and Yi-Kuo Yu (2009). "Geometric Aspects of Biological Sequence Comparison". In: *Journal of Computational Biology* 16.4. PMID: 19361329, pp. 579–610. doi: [10.1089/cmb.2008.0100](https://doi.org/10.1089/cmb.2008.0100).
- Stonebraker, Michael and Lawrence A. Rowe (1986). "The design of POSTGRES". In: *SIGMOD '86*.
- Sugawara, H. et al. (2008). "DDBJ with new system and face". In: *Nucleic Acids Research* 36, pp. D22–24. doi: [10.1093/nar/gkm889](https://doi.org/10.1093/nar/gkm889).

- Tedersoo, Leho (2017). "Proposal for practical multi-kingdom classification of eukaryotes based on monophyly and comparable divergence time criteria". In: *bioRxiv*. doi: [10.1101/240929](https://doi.org/10.1101/240929).
- Thornton, Brenda and Chhandak Basu (2011). "Real-time PCR (qPCR) primer design using free online software". In: *Biochemistry and Molecular Biology Education* 39.2, pp. 145–154. doi: [10.1002/bmb.20461](https://doi.org/10.1002/bmb.20461).
- Thurman, Harald V. and Alan P. Trujillo (2004). *Introductory Oceanography*. Introductory Oceanography. Prentice Hall. isbn: 9780131438880.
- Tusnády, Gábor E. et al. (Jan. 2005). "BiSearch: primer-design and search tool for PCR on bisulfite-treated genomes". In: *Nucleic Acids Research* 33.1, e9–e9. issn: 0305-1048. doi: [10.1093/nar/gni012](https://doi.org/10.1093/nar/gni012).
- Untergasser, Andreas et al. (June 2012). "Primer3new capabilities and interfaces". In: *Nucleic Acids Research* 40.15, e115–e115. doi: [10.1093/nar/gks596](https://doi.org/10.1093/nar/gks596).
- Vieira, Helena Henriques et al. (2016). "tufA gene as molecular marker for freshwater Chlorophyceae". In: *ALGAE* 31.2, pp. 155–165. doi: [10.4490/algae.2016.31.4.14](https://doi.org/10.4490/algae.2016.31.4.14).
- Vigna, Sebastiano (2008). "Broadword Implementation of Rank/Select Queries". In: *Experimental Algorithms*. Ed. by Catherine C. McGeoch. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 154–168. isbn: 978-3-540-68552-4.
- Vintsyuk, Taras K. (1968). "Speech discrimination by dynamic programming". In: *Cybernetics* 4.1. Russian Kibernetika 4(1):81-88 (1968), pp. 52–57.
- Visco, Joana Amorim et al. (2015). "Environmental Monitoring: Inferring the Diatom Index from Next-Generation Sequencing Data". In: *Environmental Science & Technology* 49.13, pp. 7597–7605. doi: [10.1021/es506158m](https://doi.org/10.1021/es506158m).
- Vos, Jurriaan et al. (Aug. 2014). "Estimating the Normal Background Rate of Species Extinction." In: *Conservation biology : the journal of the Society for Conservation Biology* 29. doi: [10.1111/cobi.12380](https://doi.org/10.1111/cobi.12380).
- Wallace, R. Bruce et al. (Aug. 1979). "Hybridization of synthetic oligodeoxyribonucleotides to X 174 DNA: the effect of single base pair mismatch". In: *Nucleic Acids Research* 6.11, pp. 3543–3558. issn: 0305-1048. doi: [10.1093/nar/6.11.3543](https://doi.org/10.1093/nar/6.11.3543).
- Wang, Lusheng and Tao Jiang (1994). "On the Complexity of Multiple Sequence Alignment". In: *Journal of Computational Biology* 1.4, pp. 337–348. doi: [10.1089/cmb.1994.1.337](https://doi.org/10.1089/cmb.1994.1.337).
- Wang, Qiong et al. (2007). "Naïve Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy". In: *Applied and Environmental Microbiology* 73.16, pp. 5261–5267. issn: 0099-2240. doi: [10.1128/AEM.00062-07](https://doi.org/10.1128/AEM.00062-07).
- Weese, David (2006). "Entwurf und Implementierung eines generischen Substring-Index". diploma thesis. Humboldt-Universität.
- Widenius, Michael, Davis Axmark, and Paul DuBois (2002). *Mysql Reference Manual*. 1st. USA: O'Reilly & Associates, Inc. isbn: 0596002653.
- Woese, C. R., O. Kandler, and M. L. Wheelis (1990). "Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya". In: *Proceedings of the National Academy of Sciences* 87.12, pp. 4576–4579. issn: 0027-8424. doi: [10.1073/pnas.87.12.4576](https://doi.org/10.1073/pnas.87.12.4576).
- Wurzbacher, Christian et al. (2017). "DNA metabarcoding of unfractionated water samples relates phyto-, zoo- and bacterioplankton dynamics and reveals a single-taxon bacterial bloom". In: *Environmental Microbiology Reports* 9.4, pp. 383–388. doi: [10.1111/1758-2229.12540](https://doi.org/10.1111/1758-2229.12540).

- Yang, Longqi et al. (2022). “The effects of remote work on collaboration among information workers”. In: *Nature Human Behaviour* 6 (1), pp. 43–54. doi: [10.1038/s41562-021-01196-4](https://doi.org/10.1038/s41562-021-01196-4).
- Ye, Jian et al. (June 2012). “Primer-BLAST: a tool to design target-specific primers for polymerase chain reaction. BMC Bioinform 13:134”. In: *BMC bioinformatics* 13, p. 134. doi: [10.1186/1471-2105-13-134](https://doi.org/10.1186/1471-2105-13-134).
- Yoon, Tae-Ho et al. (June 2016). “Development of a cost-effective metabarcoding strategy for analysis of the marine phytoplankton community”. In: *PeerJ* 4, e2115. doi: [10.7717/peerj.2115](https://doi.org/10.7717/peerj.2115).
- Zhang, Jiajie et al. (Oct. 2013). “PEAR: a fast and accurate Illumina Paired-End read mergeR”. In: *Bioinformatics* 30.5, pp. 614–620. issn: 1367-4803. doi: [10.1093/bioinformatics/btt593](https://doi.org/10.1093/bioinformatics/btt593).
- Zhou, Peng et al. (Mar. 2020). “A pneumonia outbreak associated with a new coronavirus of probable bat origin”. In: *Nature* 579, pp. 270–273. doi: [10.1038/s41586-020-2012-7](https://doi.org/10.1038/s41586-020-2012-7).
- Zhu, Na et al. (Jan. 2020). “A Novel Coronavirus from Patients with Pneumonia in China, 2019”. In: *New England Journal of Medicine* 382. doi: [10.1056/NEJMoa2001017](https://doi.org/10.1056/NEJMoa2001017).
- Zimmermann, Jonas, Regine Jahn, and Birgit Gemeinholzer (July 2011). “Barcoding diatoms: Evaluation of the V4 subregion on the 18S rRNA gene, including new primers and protocols”. In: *Organisms Diversity & Evolution* 11, pp. 173–192. doi: [10.1007/s13127-011-0050-6](https://doi.org/10.1007/s13127-011-0050-6).



# Selbstständigkeitserklärung

Name: Hoffmann

Vorname: Marie

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht.

Datum:

Unterschrift:

