Metabarcoding protocol – Analysis of protists using the 18S rRNA gene and a DADA2 pipeline

Nico Salmaso[1*], Giulia Riccioni[1], Massimo Pindo[1], Rainer Kurmayer[2], Valentin Vasselon[3], Isabelle Domaizon[4]

[1] Research and Innovation Centre, Fondazione Edmund Mach, San Michele all'Adige, Italy
[2] University of Innsbruck, Mondsee, Austria
[3] Scimabio Interface SAS, Thonon-les-Bains, France
[4] French National Institute for Agriculture, Food and Environment, Thonon les Bains, France

[*]Corresponding author, nico.salmaso@fmach.it

This protocol has been prepared as part of the Interreg Alpine Space project Eco-AlpsWater (ASP569) - *Innovative Ecological Assessment and Water Management Strategy for the Protection of Ecosystem Services in Alpine Lakes and Rivers*, Activity A.T1.3, Deliverable D.T1.3.2 – 2, https://www.alpine-space.eu/projects/eco-alpswater/en/home

1. Introduction

Among Eukaryotes, microscopic single celled protists share the simpler level of organization. Besides this property, this large polyphyletic assemblage of organisms includes many groups that are more closely related to plants, fungi or animals than they are to other protists. The majority of protist diversity is distinguished into a number of comprehensive monophyletic groups, which are usually referred to by the informal name "supergroups" (Guillou et al., 2013). Besides heterotrophic protists and microscopic fungi, photosynthetic and mixotrophic protists, or "algae", are scattered within many supergroups along with many other protozoans, with the exception of Archaeplastida, which form a group of their own (Simpson et al., 2017). Photosynthetic groups contribute to the primary production in oceans and inland waters, playing a fundamental role in the global CO_2/O_2 balance (Flombaum et al., 2013). Many other groups are mostly involved in the recycling of organic matter, nutrient cycling and grazing (Weisse et al., 2016), and parasitism (Schwelm et al., 2018).

In freshwater environments, the majority of the investigations were historically addressed towards the study of microalgae, which include both pelagic organisms (phytoplankton and cyanobacteria) (Reynolds, 2006; Oliver et al., 2012) and organisms attached to substrata, such as diatoms (Rimet et al., 2015) and other periphytic algae, either eukaryotic (Stevenson et al., 1996; Wehr and Sheath, 2003) or prokaryotic (Quiblier et al., 2013). Overall, these groups are composed of a wide variety of photosynthetic, mixotrophic, and even heterotrophic (Moestrup and Calado, 2018) organisms that show exclusive adaptations to specific lake typologies and trophic status.

Besides cyanobacteria, which, in this project, have been evaluated separately with the other prokaryotic species (Salmaso et al., 2021b), phytoplankton is one of the main biological elements included in the Water Framework Directive for the evaluation of lake water quality (Water Framework Directive, 2000; Carvalho et al., 2013; Pasztaleniec, 2016). The use of phytoplankton and cyanobacteria in the assessment of water quality has been fostered by a long tradition of investigations based on the identification of species by light microscopy (LM) (Sournia, 1978; Soares et al., 2011) and polyphasic approaches, supplementing LM with genetic methods (Krienitz and Bock, 2012; Kurmayer et al., 2015; Shams, 2015; Wilmotte et al., 2017) based on the use of rRNA gene markers (16S and 18S) and many other more selective genetic markers, as in the case of diatoms (Rimet et al., 2015; Vasselon et al., 2017).

Compared to phytoplankton, the study of non-photosynthetic protists in inland waters was mostly focused on taxonomic and broad ecological aspects. The knowledge of their ecological key roles was slow down by an insufficient understanding of their diversity, which, due to methodological limitations, up to recently was limited to a few more or less abundant taxa (Cotterill et al., 2008; Grossmann et al., 2016).

This deliverable provides the basic elements that are used for the identification of protists using high throughput sequencing (HTS) methods within the project Eco-AlpsWater. An overview of the analyses carried out using the 16S and 18S rRNA gene, *rbcL* (diatoms) and 12S rRNA gene (fish) gene markers in the context of the project Eco-AlpsWater is reported in Fig. 1.

2. Selection of primers

PCR amplification of the 18S rRNA genes are performed by targeting a ~380-bp fragment of the 18S rRNA gene variable region V4 using the specific primer set:

- TAReuk454FWD1 (5' CCAGCASCYGCGGTAATTCC 3') (Stoeck et al., 2010), and
- TAReukREV3_modified (5' ACTTTCGTTCTTGATYRATGA 3') (Stoeck et al., 2010; Piredda et al., 2017).

This pair of primers has been widely used in the assessment of microeukaryotic biodiversity in aquatic environments (e.g., Piredda et al., 2017; Armeli Minicante et al., 2019; Salmaso et al., 2020). In a recent comparison of the performance of primers targeting different hypervariable regions in the 18S rRNA gene, Tragin et al. (2018) found that the V4 and V9 regions provided similar images of alpha diversity and ecological patterns; though V9 was able to provide more OTUs (object taxonomic units) built at 97% identity than V4, the V9 dataset failed to describe the diversity of specific chlorophycean groups (Dolichomastigales), emphasizing the lack of sequences in this hypervariable region, and the importance of the reference database for metabarcode analysis. Moreover, owing to its short length (< 200 nt), the V9 marker provides limited phylogenetic information, whereas the V4 region allows phylogenetic/taxonomic resolution often to species-or genus-level, enabling more accurate taxonomic placements of unassigned HTS amplicon sequences (Geisen et al., 2019).

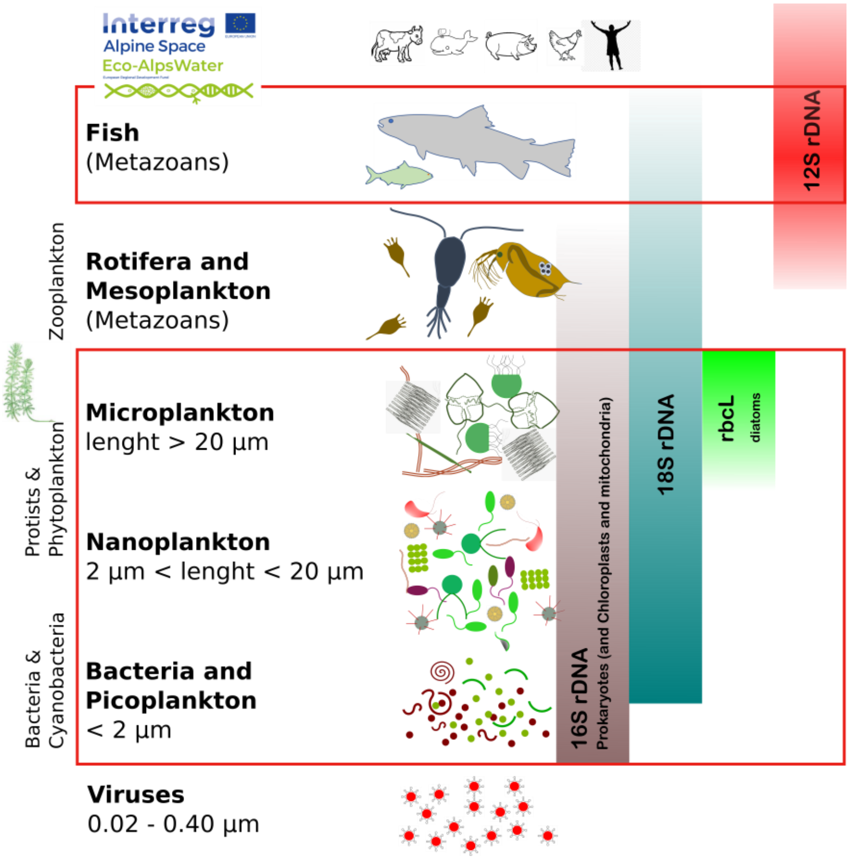

Fig. 1 – Schematic representation of planktic organisms and nekton in freshwater bodies. The biological elements included in the monitoring activities of the project Eco-AlpsWater are enclosed within red squares; these include bacteria and cyanobacteria, protists (including photosynthetic and mixotrophic microalgae, and pelagic and benthic diatoms), and fish. Macrobenthos is not represented. The specific genes used in the project are intended to target bacteria/cyanobacteria (16S rRNA gene), unicellular protists (18S rRNA gene), diatoms (*rbcL*), and fish (12S rRNA gene). Nevertheless, though designed to amplify genetic regions belonging to these intended organisms, the generality of primers is such as to amplify also “unintended” biological elements, such as, e.g., chloroplasts and mitochondria (16S rRNA gene), metazoans (mostly zooplankton, 18S rRNA gene), and higher organisms (such as mammals, 12S rRNA gene). Microorganisms (as well as other organisms) are not in scale.

3. Wet lab, amplification and HTS

DNA extraction was performed to a set of samples collected in a variety of habitats, including lakes (open waters and biofilm) and rivers (biofilm). Methods have been described in detail in Domaizon et al. (2019), Rimet et al. (2020; 2021), and Vautier et al. (2020, 2021). PCR amplification and library construction are performed as described in (Salmaso et al., 2018, 2020). All barcoded libraries are pooled in equimolar concentrations by qPCR in a final library and checked on a Typestation 2200 platform (Agilent Technologies, Santa Clara, CA, USA). The final library is sequenced on an lllumina® MiSeq (PE300) platform (MiSeq Control Software 2.6.2.1 and Real-Time Analysis software 1.18.54).

4. Bioinformatic pipeline

18S rRNA gene reads are analysed using standardized bioinformatic pipelines. Different approaches can be adopted, based essentially on the identification of OTUs built at specific levels of identity (generally 97%) (Edgar, 2018) or, as more recently proposed, on the identification of individual variants using oligotyping approaches (Eren et al., 2013, 2015) and denoising methods (amplicon sequence variants, ASVs, also known as exact sequence variants, ESVs) (Callahan et al., 2016; Edgar, 2016; Amir et al., 2017). Compared to OTUs, the use of ASVs has several advantages, including the ability to detect species at the level of strains. Moreover, representing a cloud of divergent sequences clustered ad different levels of similarity, OTUs are invalid outside of the data set in which they are defined. Conversely, representing exact sequences with consistent taxonomic labels, ASVs can be compared among different datasets (Callahan et al., 2016, 2017). Nevertheless, as in the case of OTUs (Prodan et al., 2020), although different ASVs pipelines are able to produce similar microbial compositions based on relative abundance, the approaches can provide different numbers of ASVs that significantly impact alpha diversity metrics (Nearing et al., 2018; Prodan et al., 2020). Further, high caution should be adopted in the evaluation and interpretation of ASVs diversity, due to the different 18S rRNA gene copies in the microeukaryotic cells (from less than a hundred, to well over half a million in ciliates), which can affect intragenomic heterogeneity and ASVs diversity (Wang et al., 2017; Salmaso et al., 2020).

This protocol reports a pipeline, based on DADA2, under R, for the identification of ASVs. The pipeline has been adapted from those continuously updated from the WEB site of DADA2 (https://benjjneb.github.io/dada2/index.html) (Callahan et al., 2016, 2018).

For the analyses of bacterial and microeukaryotic communities in a selection of lakes analyzed within the Eco-AlpsWater project, these pipelines have been adapted, described and applied in Salmaso (2019) and (Salmaso et al., 2020), respectively. With this protocol, the pipeline has been tested on a machine equipped with an i7 9700K and 64 Gb of RAM, under Linux Ubuntu 20.10 LTS[1] and R 4.1.0, and with the latest DADA2 version (1.20.0). The analysis of larger datasets (>50-100 pairs of F and R FASTQ files) would require the use of High Performance Computing (HPC) facilities equipped with multithread processors and high RAM (≥64 Gb). Generally, even the analysis of limited datasets could be unreliable with basic laptops (e.g. dual core and ≤ 8Gb RAM).

[1] This pipeline can be used in Windows 10 only adapting the working directories, i.e. using, e.g.,:
path <- "c:/EAW18S/"

4.1 Download of FASTQ files, and preliminary processing

A selection of 6 samples, with 6 Forward (R1) and 6 Reverse (R2) files are used in this tutorial. R1 and R2 reads are 300 bp long, and were obtained from Illumina MiSeq technologies, at the FEM facility sequence (sections 2 and 3). The files refer to the18S rRNA gene reads obtained from the analyses carried out on the samples collected and filtered (Sterivex^{TM} 0.22 µm) in different stations of Lake Garda on September, 2018 (Fig. 2)[2].

Forward (R1)	Reverse (R2)	Code (Fig. 2)
• ECOALPSWATER-18S-386-09-18-0stv_S145_L001_R1_001.fastq	• ECOALPSWATER-18S-386-09-18-0stv_S145_L001_R2_001.fastq	S0
• ECOALPSWATER-18S-386-09-18-4stv_S146_L001_R1_001.fastq	• ECOALPSWATER-18S-386-09-18-4stv_S146_L001_R2_001.fastq	S4
• ECOALPSWATER-18S-B0918D1stv_S141_L001_R1_001.fastq	• ECOALPSWATER-18S-B0918D1stv_S141_L001_R2_001.fastq	C0
• ECOALPSWATER-18S-B0918D4stv_S142_L001_R1_001.fastq	• ECOALPSWATER-18S-B0918D4stv_S142_L001_R2_001.fastq	C100
• ECOALPSWATER-18S-B0918D5stv_S143_L001_R1_001.fastq	• ECOALPSWATER-18S-B0918D5stv_S143_L001_R2_001.fastq	C300
• ECOALPSWATER-18S-Porto0918stv_S144_L001_R1_001.fastq	• ECOALPSWATER-18S-Porto0918stv_S144_L001_R2_001.fastq	H0

The 12 files are stored in the Zenodo archives (Salmaso et al., 2021b; https://zenodo.org/record/5215919#.YSJyHEtxeHs). Download the files in a working directory, under your home dir, `~/EAW18S`.
Either before or during data processing with DADA2, primers at the beginning of the F and R reads will have to be trimmed. In the first case, primers will be trimmed before the application of the DADA2 bioinformatic pipeline using Cutadapt[3] (see below). In the second case, primers will be trimmed by using an internal procedure implemented in DADA2 (argument `trimLeft` in the function `filterAndTrim`); this last procedure is presented in the **Appendix 1**.

A first option to trim primers from FASTQ and zipped FASTQ files, which does not require a native installation, is to use the version of Cutadapt included in the Galaxy web-based platform (https://usegalaxy.org/), following the links: Genomic File Manipulation → FASTA/FASTQ → Cutadapt; the application works on single paired reads and multiple datasets. The Galaxy platform is runs under Windows and UNIX operating systems. All the operations have to be completed using menus.

[2] The same set of samples and DNA extracts were used to analyze the 16S rRNA gene profiles (Salmaso et al., 2021a, 2021c).

[3] Cutadapt is only one among several options that can be used to trim primers. Results obtained with different tools can differ (Lindgreen, 2012; Kechin et al., 2017). Moreover, final results depend, for every single tools, from the choice of parameters. In this protocol, Cutadapt uses default options, with the exception of -t (TRUE), which allows discarding reads that do not contain the primers.

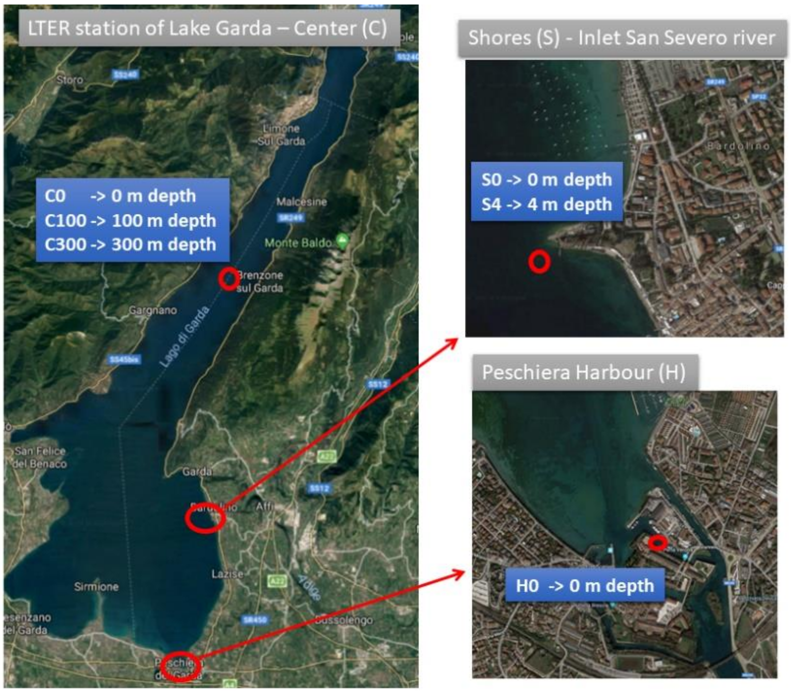

Fig. 2 – Location of the sampling stations considered in this protocol.

A second straightforward option, is to use Cutadapt natively, under UNIX operating systems. Under Linux Debian operating systems (e.g. Ubuntu), Cutadapt can be installed, in the terminal, using:

```
sudo apt install cutadapt
```

To assure the installation of the most recent versions of Cutadapt, the machine should be equipped with the latest LINUX versions. For other installation options see https://cutadapt.readthedocs.io/en/stable/installation.html.

Since Cutadapts works on single FASTQ or single paired FASTQ files, the application on multiple datasets (samples) requires the use of specific wrappers. Here we will use the bash script rmprim.sh (https://github.com/hts-tools/metatools). The script works when the F and R reads do not extend into the opposite primers, as in the case of TAReuk454FWD1 and TAReukREV3_modified used in this protocol. After opening the terminal, enter the directory `~/EAW18S`, and download rmprim.sh:

```
cd ~/EAW18S
wget https://raw.githubusercontent.com/hts-tools/metatools/master/rmprim/rmprim.sh
```

Primers can be removed using the script[4]:

```
bash rmprim.sh -f CCAGCASCYGCGGTAATTCC -r ACTTTCGTTCTTGATYRATGA -n TRUE -t TRUE -d
~/EAW18S
```

The arguments -f and -r indicate the F and R primers; -n TRUE indicates anchored primers (at the beginning of reads); if TRUE, -t discard reads that do not contain the adapter; -d indicates the

[4] rprim requires zipped FASTQ files (fastq.gz). If the fastq files are not zipped, move to the FASTQ dir, and use the bash script: “`for file in *; do gzip -k "$file"; done`”

directory where the FASTQ files are stored. The result is a corresponding number of trimmed files, with extension *trim.fastq.gz. These files are used in the DADA2 pipeline, below.

4.2 Installation of DADA2

DADA2 can be installed in the R environment under Linux (e.g. Debian/Ubuntu), Mac OS, or Windows. In the following, the pipeline describes the utilization under Linux, but the script, after adapting the directories, can be easily run also under Windows. In this context, UNIX environments have the advantage to use multicores/multithreading. The pipeline assumes that one of the latest versions of R (>= 4.1.0) is already installed in the machine. Binaries can be downloaded and installed from Bioconductor using the BiocManager. In case of installation problems, see https://www.bioconductor.org/install/. Different versions of R, bioconductor and DADA2 can cause conflicts due to incompatibility. Before every project analysis, a good strategy is to evaluate the requirements of the last DADA2 version, and to install R and Bioconductor packages accordingly.

After launching R, if BiocManager is not installed, run:

```
chooseCRANmirror()
install.packages("BiocManager")
```

Then install the package dada2:

```
chooseCRANmirror()
BiocManager::install("dada2")
```

Other required packages are tidyverse and openxlsx; if not installed, run:

```
install.packages("openxlsx", dependencies=TRUE)
install.packages(vegan)
install.packages("tidyverse")
```

Load dada2 and other packages into memory:

```
rm(list=ls(all=TRUE))
library(dada2)
library(openxlsx)
library(vegan)
library(tidyverse)
packageVersion("dada2")
```

In the following steps, the input directory `~/EAW18S` is saved in the variable “path”. Moreover, to keep things in order, other sub-directories are created under `~/EAW18S` with the following lines:

```
path <- "~/EAW18S"
setwd(path)
list.files(path)
# prepare the directories for tables and analyses
pathtab <- paste0(path, '/', 'Tables/')
pathana <- paste0(path, '/', 'Analysis/')
pathtax <- paste0(path, '/', 'Taxonomy/')
dir.create(pathtab)
dir.create(pathana)
dir.create(pathtax)
```

4.3. Evaluation of quality profiles

Read the names of files, and obtain R1 and R2 fastq files in matched order[5]:

```
fnFs <- sort(list.files(path, pattern="_R1_001_trim.fastq.gz", full.names = TRUE))
fnRs <- sort(list.files(path, pattern="_R2_001_trim.fastq.gz", full.names = TRUE))
sample.names <- sapply(strsplit(basename(fnFs), "_"), '[', 1)
```

Visualize the quality profiles of the forward and reverse reads (here, only the first four will be shown) (Fig. 3):

```
plotQualityProfile(fnFs[1:4])
plotQualityProfile(fnRs[1:4])
```

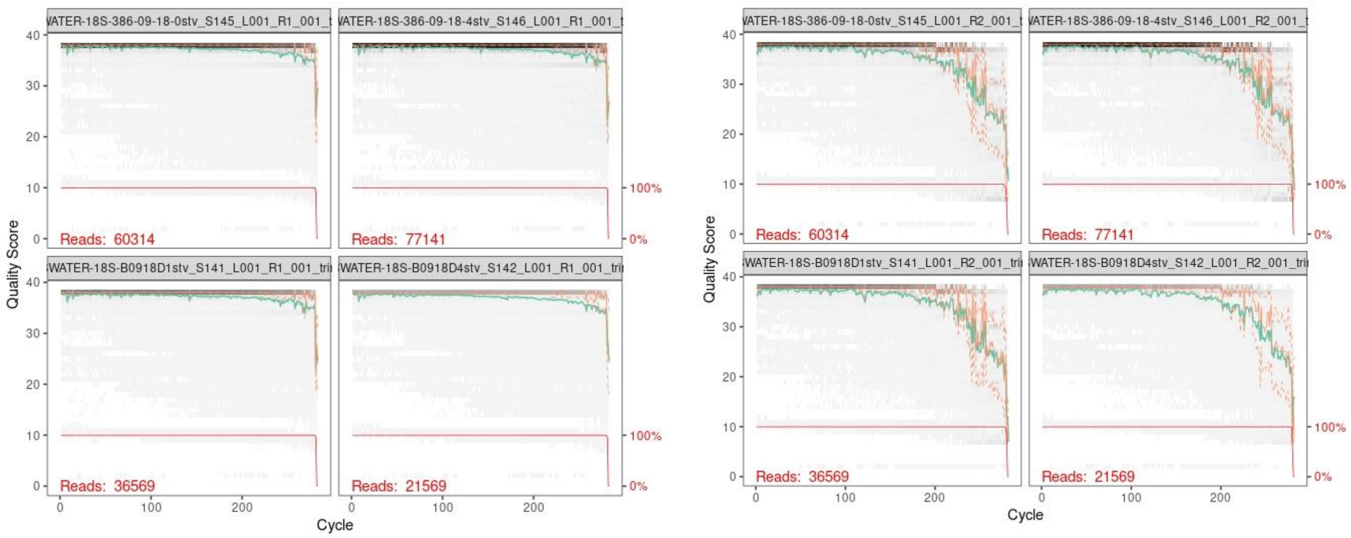

Fig. 3 – Quality profiles of the forward (R1, left) and reverse (R2, rigth) reads (primers trimmed). Quality scores are encoded in the FASTQ files (fourth line of each single read). The bases are along the horizontal axis, whereas the quality scores are reported on the vertical axis. The gray-scale is a heat map of the frequency of each Q-score at each base position; darker colors correspond to higher frequency. The green line is the mean quality score at each base position, and the three orange lines show the quartiles (median, 50th continuous; 25th and 75th dashed). The evaluation of quality profiles of all the samples can be facilitated by averaging the analysis (argument `aggregate=TRUE`, which computes an aggregate quality profile for all fastq files provided). A red line is plotted when the sequences vary in length, indicating the percentage of reads (rigth y-axis) that extend to at least that position (on the x-axis).

These plots allow deciding which range of bases to include in the analysis. Q-scores of 40, 30 and 20 indicates an expected error rate of 1 in 10000, 1 in 1000, and 1 in 100, respectively. As a rule of thumb, truncation should exclude average qualities Q-scores read areas < 30. Truncation, however, should allow overlapping of R1 and R2 reads in successive analyses. In this exercise, R1 and R2

[5] Besides fastq.gz files, fastq files can be analysed. In that case, `pattern="_R1_001.fastq"`…

reads will be truncated at 255 and 220, respectively, allowing a final overlap of around >75 bp bewteen R1 and R2 reads[6].

The quality-filtering step is done with the `filterAndTrim()` function. The argument `truncLen` allows truncating the R1 and R2 reads at the desired length. The new filtered fastq are saved in the directory `"~/EAW18S/filtered/"` . The parameter `truncQ` truncate reads at the first instance of a quality score less than or equal to truncQ. After truncation with `truncLen`, reads with higher than `maxEE` "expected errors" will be discarded; `maxEE` (1, default 2) sets the maximum number of expected errors allowed in each read; $EE = \Sigma_i 10^{-Q_i/10}$ (Edgar and Flyvbjerg, 2015). `matchIDs=TRUE` enforces matching between the id-line sequence identifiers of the R1 and R2 fastq files. All the other arguments in `filterAndTrim()` are set to default values.

```
filtFs <- file.path(path, "filtered", paste0(sample.names, "_F_filt.fastq.gz"))
filtRs <- file.path(path, "filtered", paste0(sample.names, "_R_filt.fastq.gz"))
names(filtFs) <- sample.names
names(filtRs) <- sample.names
out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs, truncQ=5, truncLen=c(255,220),
maxEE=c(1,1), matchIDs=TRUE, maxN = 0, rm.phix=TRUE, multithread=TRUE, verbose = TRUE)
out # On Windows, multithread is not supported
mean(out[,2])/mean(out[,1])
```

The output shows the fraction of reads retained and discarded. The quality of the filtered filed can be also cheked (figures not shown):

```
plotQualityProfile(filtFs[1:4])
plotQualityProfile(filtRs[1:4])
```

4.4 Learn the Error Rates and Sample Inference

In this step, DADA2 removes all sequencing errors to reveal the members of the sequenced community. For details, see https://rdrr.io/github/benjjneb/dada2/man/dada.html, and (Callahan et al., 2016). The error profiles are used in a successive step to correct errors.

```
set.seed(123)
errF <- learnErrors(filtFs, multithread=TRUE, verbose = TRUE, nbases = 2e+08, MAX_CONSIST
= 15)
errR <- learnErrors(filtRs, multithread=TRUE, verbose = TRUE, nbases = 2e+08, MAX_CONSIST
= 15)
plotErrors(errF, nominalQ=TRUE)
plotErrors(errR, nominalQ=TRUE)
```

In the sample inference step, the sample inference algorithm is applied to the filtered and trimmed sequence data, with the aim to infer true biological sequences. This is done by incorporating the quality profiles and abundances of each unique sequence, deciding if sequences are “true” (of

[6] The truncation values must be decided each time. The values used in this protocol were defined after examining the quality profiles, in Fig. 3, and do not necessarily apply to other data sets. If reads are of good quality, the value of the trunclen parameter can be increased, allowing a higher number of bases overlapping between R1 and R2. Viceversa, if reads are of bad quality, try decrease the value of the trunclen parameter, but maintaining a final overlap between R1 and R2 reads of at least 20 bp + biological length variation (see https://benjjneb.github.io/dada2/tutorial.html).

biological origin), or spurious (Callahan et al., 2016). A dereplication step (as in previous versions of DADA2), is no more necessary.

```
dadaFs <- dada(filtFs, err=errF, pool=FALSE, multithread=TRUE)
dadaRs <- dada(filtRs, err=errR, pool=FALSE, multithread=TRUE)
```

If `pool = TRUE`, the algorithm will pool together all samples prior to sample inference. If `pool = FALSE` (<u>default</u>), sample inference is performed on each sample individually. If `pool = "pseudo"`, the algorithm will perform pseudo-pooling between individually processed samples. Pooling samples together increases the ability to identify low-abundance ASVs but can be computationally not feasible on common computers when datasets are large. Estimated error rates are reported in Fig. 4.

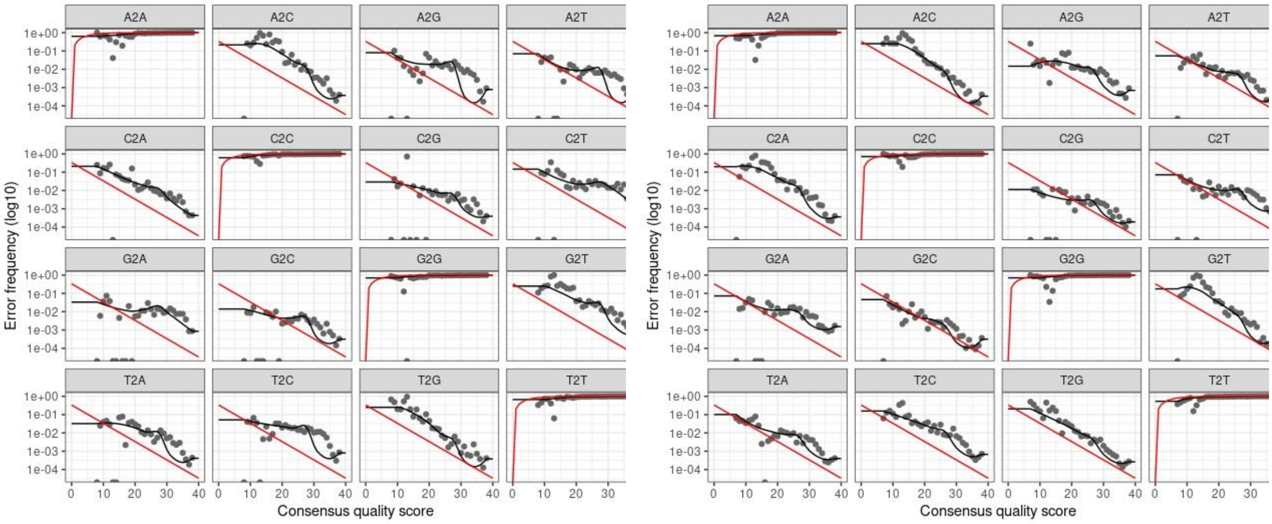

Fig. 4 – Visualization of the estimated error rates. Each single graph shows the error rates for each possible transition (e.g. A→C, A→G, … T→G). The red line is what is expected based on the quality score; the black line is the estimate, whereas the black dots are the observed. Overall, the observed, black dots, should track well the estimated errors (black line).

4.5 Merging forward (R1) and reverse (R2) reads

Reconstruction of target amplicons requires the overlapping region of F and R reads to be identical. The function `mergePairs` requires as default a minimum of 12 bp. In this dataset, after cutting the primers, and truncating the R1 and R2 reads, we expect an amplicon size of around 380 bp, and an overlap of ca. >75 bp. As a conservative measure, and allowing for natural biological variation, the minimum overlap in this dataset can be fixed at 65, which is still considerably large. On a practical ground, this value has been checked also considering the outputs and the fraction of merged reads (see table below). No mismatches are allowed in the overlap region.

```
merged_reads <- mergePairs(dadaFs, filtFs, dadaRs, filtRs, verbose=TRUE, minOverlap=65,
maxMismatch=0)
length(merged_reads)
head(merged_reads[[1]])
```

4.6 Generate the count table (ASV matrix, abundance table)

```
seqtaball <- makeSequenceTable(merged_reads)
dim(seqtaball)
```

The table is a matrix, in which rows correspond to samples (6), and columns to the sequence variants (908).

```
plot(table(nchar(getSequences(seqtaball))))
```

A fraction of lengths in the merged sequences do not fall within the expected range for this V4 amplicon, possibly because of non-specific priming. These sequences could be removed. This is conservative, and it could be worth always a try to inspect the nature of the discarded sequences. Actually, at sequence length between 300 and 340 bp, a number of ciliates has been identified. Therefore, a lower limit of 300 bp allows to retain all the taxonomic information, whereas further checking and taxonomic filtering can be made on the downstream analyses of data.

```
seqtab <- seqtaball[,nchar(colnames(seqtaball)) %in% seq(300,405)]
dim(seqtab)
plot(table(nchar(getSequences(seqtab))), xlab = "Reads R1+R2 merged length")
```

The final result, after discarding sequences outside the expected range, is given in Fig. 5.

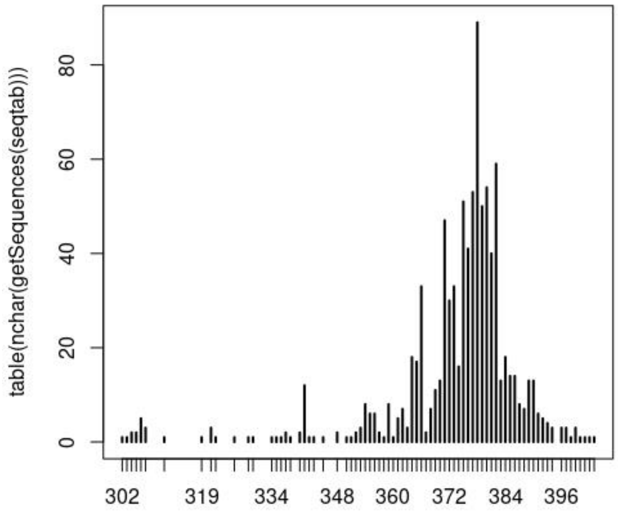

Fig. 5 – Number of reads (y) with specific amplicon lengths (x) in the 18S rRNA gene V4 region.

4.7 Chimera identification and removal, and track reads through the pipeline

Chimeric sequences are identified and then removed if they are formed by the left-segment and a right-segment belonging to two of the more abundant sequences.

```
seqtab.nochim <- removeBimeraDenovo(seqtab, method="consensus", multithread=TRUE,
verbose=TRUE)
dim(seqtab.nochim)
sum(seqtab.nochim)/sum(seqtab)*100
```

When accounting for the abundances of the merged sequence variants, chimeras account for less than 2% of the merged sequence reads.

As a further computational check, it is worth to know how many reads were discarded at various points of the pipeline:

```
getN <- function(x) sum(getUniques(x))
track <- cbind(out, sapply(dadaFs, getN), sapply(dadaRs, getN), sapply(merged_reads,
getN), rowSums(seqtab.nochim))
colnames(track) <- c("input", "filtered", "denoisedF", "denoisedR", "merged", "nonchim")
rownames(track) <- sample.names
head(track)
```

	input	filtered	denoisedF	denoisedR	merged	nonchim
ECOALPSWATER-18S-386-09-18-0stv	60314	54383	53835	53834	51151	50591
ECOALPSWATER-18S-386-09-18-4stv	77141	69917	69381	69497	67462	66415
ECOALPSWATER-18S-B0918D1stv	36569	33202	32983	33017	32152	31962
ECOALPSWATER-18S-B0918D4stv	21569	19441	19243	19295	17529	17055
ECOALPSWATER-18S-B0918D5stv	19279	17572	17415	17468	16874	16782
ECOALPSWATER-18S-Porto0918stv	33337	30151	29862	29907	28622	28339

Results are quite good. After the filtering step, the majority of reads should be retained.

4.8 Taxonomy assignement

Taxonomy assignement is implemented using the naive Bayesian classifier method (Wang et al., 2007)and the PR2 database (PR2 version 4.14.0, June 2021[7]), a curated list containing only eukaryotic taxa (Guillou et al., 2013). Download the file `pr2_version_4.14.0_SSU_dada2.fasta.gz`, saving it in the directory `~/EAW18S/Taxonomy/` (skip this step if the taxonomy file was already downloaded):

```
download.file("https://github.com/pr2database/pr2database/releases/download/v4.14.0/pr2_v
ersion_4.14.0_SSU_dada2.fasta.gz", paste0(pathtax,
"pr2_version_4.14.0_SSU_dada2.fasta.gz"))
```

The minimum bootstrap confidence for assigning a taxonomic level has been set to 95 (default=50). This step is computationally demanding, requiring a high amount of RAM memory.

[7] Previous analyses of the EAW 18S rRNA sequences made in December 2020 were done on the PR2 version 4.12.0 (August 2020) reference database, i.e. “pr2_version_4.12.0_18S_dada2.fasta.gz”. In case of updating the taxonomic databases, their name in the scripts will have to be changed accordingly.

```
taxaPR2 <- assignTaxonomy(seqtab.nochim, paste0(pathtax,
"pr2_version_4.14.0_SSU_dada2.fasta.gz"), multithread=TRUE, minBoot = 95, verbose = TRUE,
taxLevels=c("Kingdom", "Supergroup", "Division", "Class", "Order", "Family", "Genus",
"Species"))
```

Save the session. Results can be successively loaded in R with the function `load`:

```
save.image(paste0(pathana, "EAW18S_analysis.RData"))
```

4.9 Collecting DADA2 results: saving tables for downstream statistical analyses

```
# clean the “Tables” dir of previous (if any) files
setwd(pathtab)
file.remove(list.files())
setwd(path)

# save sequences with original headers
write.csv(t(seqtab.nochim), paste0(pathtab, "seqtab-nochim.csv"), quote=FALSE)

# simplify names to sequence headers(seq1, seq2...seq100...)
# adapted from: https://github.com/benjjneb/dada2/issues/655
seqs <- colnames(seqtab.nochim)
SeqName <- vector(dim(seqtab.nochim)[2], mode="character")
SeqName_ft <- vector(dim(seqtab.nochim)[2], mode="character")
for (i in 1:dim(seqtab.nochim)[2]) {
  SeqName[i] <- paste("seq", i, sep="")
  SeqName_ft[i] <- paste(">seq", i, sep="")
}

# write sequences in a FASTA file
fastaseqs_ft <- rbind(SeqName_ft, seqs)
write(fastaseqs_ft, paste0(pathtab,"fastaseqs.fasta"))
# write sequences in a FASTA file (tabula)
fastaseqs <- cbind(SeqName, seqs)
write.csv(fastaseqs, paste0(pathtab,"fastaseqs.csv"), quote=FALSE, row.names = FALSE)

# write count table
seqtab.nochim.t <- t(seqtab.nochim)
row.names(seqtab.nochim.t) <- SeqName
seqtab.nochim.t <- tibble::rownames_to_column(as.data.frame(seqtab.nochim.t), "SeqName")
write.csv(seqtab.nochim.t, paste0(pathtab, "counts.csv"), quote=FALSE, row.names = FALSE)

# write taxonomy table
taxtable <- taxaPR2
row.names(taxtable) <- SeqName
taxtable <- tibble::rownames_to_column(as.data.frame(taxtable), "SeqName")
write.csv(taxtable, paste0(pathtab, "taxtable.csv"), quote=FALSE, row.names = FALSE)

# Join results and save in spreadsheet (excel) format
tax_counts <- left_join(taxtable, seqtab.nochim.t, by = "SeqName", keep = TRUE)
tax_counts_fasta <- left_join(tax_counts, as.data.frame(fastaseqs),  by = c("SeqName.x" =
"SeqName"), keep = TRUE)
openxlsx::write.xlsx(tax_counts_fasta, file = paste0(pathtab, "tax_counts_fasta.xlsx"),
overwrite = TRUE, asTable = FALSE, sheetName = "EAW_18S", firstRow = TRUE, zoom = 90,
keepNA = TRUE)
```

The above files can be imported in spreadsheet and/or statistical programs, merged, and analyzed in downstream statistical analyses. Here a quick example (Fig. 6), using the package vegan (Oksanen et al., 2020):

```
library(vegan)
counts <-read.csv(file=paste0(pathtab, "counts.csv"), header=T, row.names=1)
head(counts)
counts2 <- t(counts)^0.25
bc <- vegdist(counts2, method="bray")
plot(hclust(bc))
```

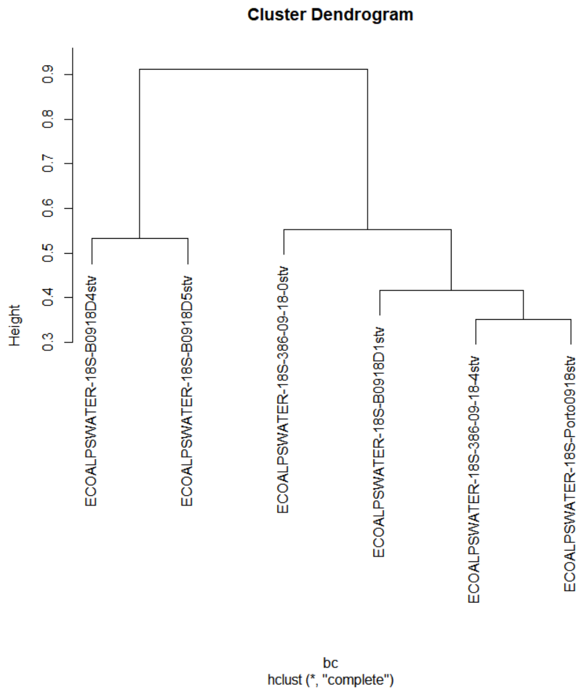

Fig. 6 - Quick and dirty cluster analysis made importing the file counts.tsv in R. Data preliminarily transformed by double square root. Note how the deep samples are isolated from the surface samples.

4.10 Export data to phyloseq

Install the package phyloseq (McMurdie and Holmes, 2013), following the instructions provided in bioconductor, https://www.bioconductor.org/packages/release/bioc/html/phyloseq.html ,
and load the package:

```
library(phyloseq)
```

Open the metadata spreadsheet file “EAW_2018_18S_metadata.ods” (in Zenodo, https://doi.org/10.5281/zenodo.5215919), delete the first line, and save in .CSV format under the dir “~/EAW18S”. Import the metadadata in R (check the parameter “sep”: it can be either “;” or ”,”):

```
ambio <- read.csv(file="EAW_2018_18S_metadata.csv", header=T, row.names=1, sep = ";")
ambio
```

Create a phyloseq object, and save it under the dir “~/EAW18S/Analysis”,

```
taxtable_ps <- taxaPR2
row.names(taxtable_ps) <- SeqName
eawps18S <- phyloseq(otu_table(counts, taxa_are_rows=TRUE), sample_data(ambio),
tax_table(taxtable_ps))
eawps18S
saveRDS(eawps18S, file = paste0(pathana, "EAW18S_ps.rds"))
```

eawps can be successively loaded in new R sessions, and data analyzed with phyloseq (https://joey711.github.io/phyloseq/index.html):

```
eawps18S <- readRDS(file = paste0(pathana, "EAW18S_ps.rds"))
```

APPENDIX 1

As indicated by the authors of DADA2 (http://benjjneb.github.io/dada2/faq.html), if primers are at the start of reads and are a constant length, the argument trimLeft = c(FWD_PRIMER_LEN, REV_PRIMER_LEN) in the filtering function `filterAndTrim` can be used to remove the primers. For more complex situations, see https://benjjneb.github.io/dada2/ITS_workflow.html

The trimming of primers using DADA2 is described below, using a sligth modification of section 4.3. If not already done, download the test files to the directory ~/EAW18S and go through Section 4.2, and then follow 4.3.A, below.

4.3.A Evaluation of quality profiles

Read the names of untrimmed files, and obtain R1 and R2 fastq files in matched order:

```
fnFs <- sort(list.files(path, pattern="_R1_001.fastq.gz", full.names = TRUE))
fnRs <- sort(list.files(path, pattern="_R2_001.fastq.gz", full.names = TRUE))
sample.names <- sapply(strsplit(basename(fnFs), "_"), '[', 1)
```

Visualize the quality profiles of the forward and reverse reads (here, only the first four will be shown) (Fig. 7):

```
plotQualityProfile(fnFs[1:4])
plotQualityProfile(fnRs[1:4])
```

These plots allow deciding which range of bases to include in the analysis. In this exercise, R1 and R2 reads will be truncated at 275 and 241, respectively, allowing a final overlap of around >75 bp bewteen R1 and R2 reads.

The quality-filtering step is done with the `filterAndTrim()` function. The argument `truncLen` allows truncating the R1 and R2 reads at the desired length. At the same time, primers are removed using the argument `trimLeft`. For the other arguments in `filterAndTrim()` see 4.3. The new filtered fastq files are saved in the directory `"~/EAW18S/filtered/"`. .

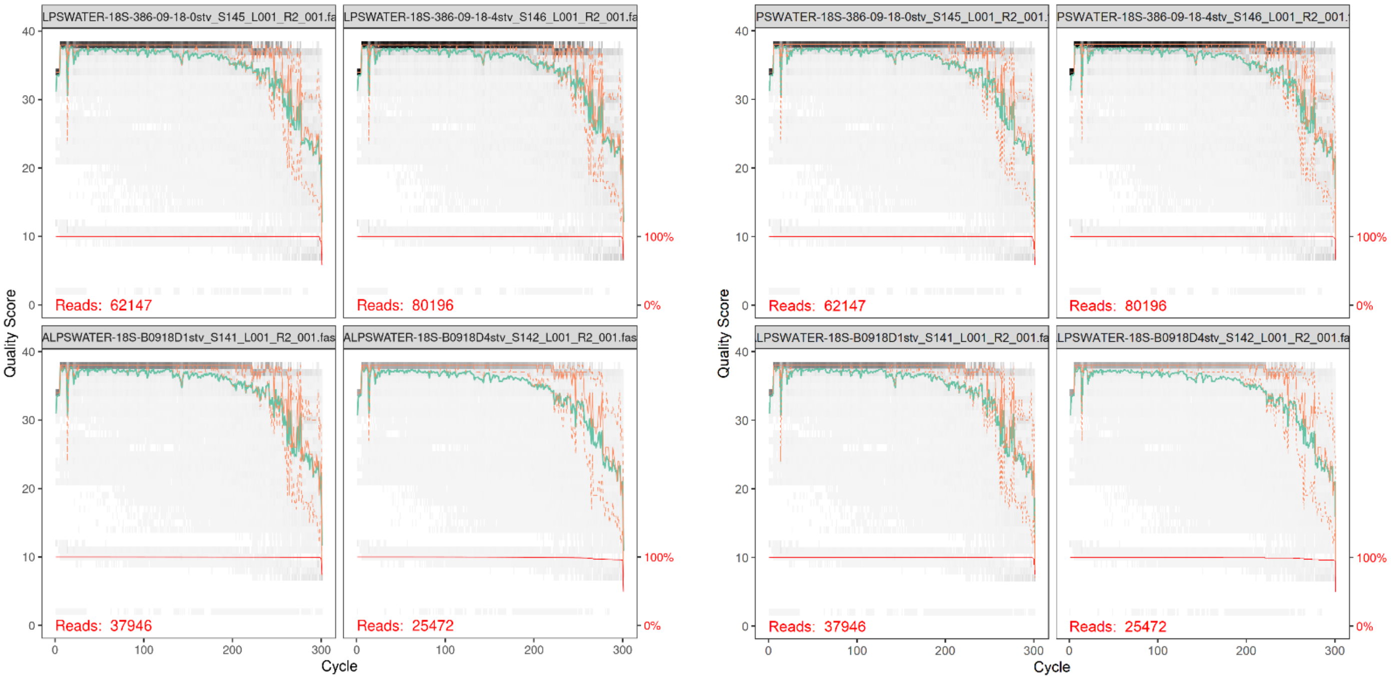

Fig. 7 – Quality profiles of the forward (R1, left) and reverse (R2, rigth) reads. These reads still include the primers.

```
filtFs <- file.path(path, "filtered", paste0(sample.names, "_F_filt.fastq.gz"))
filtRs <- file.path(path, "filtered", paste0(sample.names, "_R_filt.fastq.gz"))
names(filtFs) <- sample.names
names(filtRs) <- sample.names
out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs, truncLen=c(275,241), trimLeft=c(20, 21),
maxEE=c(2,2), multithread=TRUE, matchIDs=TRUE)
out # On Windows, multithread is not supported
mean(out[,2])/mean(out[,1])
```

The output shows the fraction of reads discarded. The quality of the filtered filed can be also cheked (figures not shown):

```
plotQualityProfile(filtFs[1:4])
plotQualityProfile(filtRs[1:4])
```

Continue with Section 4.4.

***

5. References

Amir, A., McDonald, D., Navas-Molina, J. A., Kopylova, E., Morton, J. T., Zech Xu, Z., et al. (2017). Deblur Rapidly Resolves Single-Nucleotide Community Sequence Patterns. *mSystems* 2. doi:10.1128/msystems.00191-16.

Armeli Minicante, S., Piredda, R., Quero, G. M., Finotto, S., Bernardi Aubry, F., Bastianini, M., et al. (2019). Habitat Heterogeneity and Connectivity: Effects on the Planktonic Protist Community Structure at Two Adjacent Coastal Sites (the Lagoon and the Gulf of Venice, Northern Adriatic Sea, Italy) Revealed by Metabarcoding. *Front. Microbiol.* 10, 2736. doi:10.3389/fmicb.2019.02736.

Callahan, B. J., McMurdie, P. J., and Holmes, S. P. (2017). Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. *ISME J.* 11, 2639–2643. doi:10.1038/ismej.2017.119.

Callahan, B. J., McMurdie, P. J., Rosen, M. J., Han, A. W., Johnson, A. J. A., and Holmes, S. P. (2016). DADA2: High-resolution sample inference from Illumina amplicon data. *Nat. Methods* 13, 581–583. doi:10.1038/nmeth.3869.

Callahan, B., McMurdie, P. J., and Holmes, S. (2018). Package “dada2”. Accurate, high-resolution sample inference from amplicon sequencing data.

Carvalho, L., Poikane, S., Lyche Solheim, A., Phillips, G., Borics, G., Catalan, J., et al. (2013). Strength and uncertainty of phytoplankton metrics for assessing eutrophication impacts in lakes. *Hydrobiologia* 704, 127–140. doi:10.1007/s10750-012-1344-1.

Cotterill, F. P. D., Al-Rasheid, K., and Foissner, W. (2008). Conservation of protists: is it needed at all? *Biodivers. Conserv.* 17, 427–443. doi:10.1007/s10531-007-9261-8.

Domaizon, I., Kurmayer, R., Capelli, C., Chardon, C., Hufnagl, P., Vautier, M., et al. (2019). Lake plankton sample collection from the field for downstream molecular analysis. *protocols.io*. doi:dx.doi.org/10.17504/protocols.io.xn6fmhe.

Edgar, R. C. (2016). UNOISE2: improved error-correction for Illumina 16S and ITS amplicon sequencing. *bioRxiv*, 081257. doi:10.1101/081257.

Edgar, R. C. (2018). Updating the 97% identity threshold for 16Sribosomal RNA OTUs. *Bioinformatics* 34, 2371–2375.

Edgar, R. C., and Flyvbjerg, H. (2015). Error filtering, pair assembly and error correction for next-generation sequencing reads. *Bioinformatics* 31, 3476–3482. doi:10.1093/bioinformatics/btv401.

Eren, A. M., Maignien, L., Sul, W. J., Murphy, L. G., Grim, S. L., Morrison, H. G., et al. (2013). Oligotyping: Differentiating between closely related microbial taxa using 16S rRNA gene data. *Methods Ecol. Evol.* 4, 1111–1119. doi:10.1111/2041-210X.12114.

Eren, A. M., Morrison, H. G., Lescault, P. J., Reveillaud, J., Vineis, J. H., and Sogin, M. L. (2015). Minimum entropy decomposition: Unsupervised oligotyping for sensitive partitioning of high-throughput marker gene sequences. *ISME J.* 9, 968–979. doi:10.1038/ismej.2014.195.

Flombaum, P., Gallegos, J. L., Gordillo, R. A., Rincón, J., Zabala, L. L., Jiao, N., et al. (2013). Present and future global distributions of the marine Cyanobacteria Prochlorococcus and Synechococcus. *Proc. Natl. Acad. Sci. U. S. A.* 110, 9824–9. doi:10.1073/pnas.1307701110.

Geisen, S., Vaulot, D., Mahé, F., Lara, E., Vargas, C. de, Bass, D., et al. (2019). A user guide to environmental protistology: primers, metabarcoding, sequencing, and analyses. *bioRxiv*, 850610. doi:10.1101/850610.

Grossmann, L., Jensen, M., Heider, D., Jost, S., Glücksman, E., Hartikainen, H., et al. (2016). Protistan community analysis: Key findings of a large-scale molecular sampling. *ISME J.* 10, 2269–2279. doi:10.1038/ismej.2016.10.

Guillou, L., Bachar, D., Audic, S., Bass, D., Berney, C., Bittner, L., et al. (2013). The Protist

Ribosomal Reference database (PR2): A catalog of unicellular eukaryote Small Sub-Unit rRNA sequences with curated taxonomy. *Nucleic Acids Res.* 41, D597–D604. doi:10.1093/nar/gks1160.

Kechin, A., Boyarskikh, U., Kel, A., and Filipenko, M. (2017). CutPrimers: A New Tool for Accurate Cutting of Primers from Reads of Targeted Next Generation Sequencing. *J. Comput. Biol.* 24, 1138–1143. doi:10.1089/cmb.2017.0096.

Krienitz, L., and Bock, C. (2012). Present state of the systematics of planktonic coccoid green algae of inland waters. *Hydrobiologia* 698, 295–326. doi:10.1007/s10750-012-1079-z.

Kurmayer, R., Blom, J. F., Deng, L., and Pernthaler, J. (2015). Integrating phylogeny, geographic niche partitioning and secondary metabolite synthesis in bloom-forming Planktothrix. *ISME J.* 9, 909–21. doi:10.1038/ismej.2014.189.

Lindgreen, S. (2012). AdapterRemoval: Easy cleaning of next-generation sequencing reads. *BMC Res. Notes* 5, 337. doi:10.1186/1756-0500-5-337.

Moestrup, Ø., and Calado, A. J. (2018). “Dinophyceae.,” in *Süßwasserflora von Mitteleuropa*, eds. B. Büdel, G. Gärtner, M. Schagerl, and L. Krienitz (Berlin: Springer Spektrum), 1–560. doi:https://doi.org/10.1007/978-3-662-56269-7.

Nearing, J. T., Douglas, G. M., Comeau, A. M., and Langille, M. G. I. (2018). Denoising the Denoisers: an independent evaluation of microbiome sequence error-correction approaches. *PeerJ* 6, e5364. doi:10.7717/peerj.5364.

Oksanen, J., Blanchet, F. G., Friendly, M., Kindt, R., Legendre, P., McGlinn, D., et al. (2020). vegan: Community Ecology Package. 285. Available at: https://cran.r-project.org/package=vegan.

Oliver, R. L., Hamilton, D. P., Brookes, J. D., and Ganf, G. G. (2012). “Physiology, Blooms and Prediction of Planktonic Cyanobacteria,” in *Ecology of Cyanobacteria II Their Diversity in Space and Time*, ed. B. A. Whitton (Dordrecht, The Netherlands: Springer), 155–194. Available at: http://www.springerlink.com/index/10.1007/978-94-007-3855-3.

Pasztaleniec, A. (2016). Phytoplankton in the ecological status assessment of European lakes - Advantages and constraints. *Ochr. Sr. i Zasobow Nat.* 27, 26–36. doi:10.1515/OSZN-2016-0004.

Piredda, R., Tomasino, M. P., D’Erchia, A. M., Manzari, C., Pesole, G., Montresor, M., et al. (2017). Diversity and temporal patterns of planktonic protist assemblages at a Mediterranean Long Term Ecological Research site. *FEMS Microbiol. Ecol.* 93, fiw200. doi:10.1093/femsec/fiw200.

Prodan, A., Tremaroli, V., Brolin, H., Zwinderman, A. H., Nieuwdorp, M., and Levin, E. (2020). Comparing bioinformatic pipelines for microbial 16S rRNA amplicon sequencing. *PLoS One* 15, 1–19. doi:10.1371/journal.pone.0227434.

Quiblier, C., Wood, S., Echenique-Subiabre, I., Heath, M., Villeneuve, A., and Humbert, J.-F. (2013). A review of current knowledge on toxic benthic freshwater cyanobacteria - ecology, toxin production and risk management. *Water Res.* 47, 5464–79. doi:10.1016/j.watres.2013.06.042.

Reynolds, C. S. (2006). *The ecology of phytoplankton*. Cambridge University Press doi:10.1017/CBO9780511542145.

Rimet, F., Bouchez, A., and Montuelle, B. (2015). Benthic diatoms and phytoplankton to assess nutrients in a large lake: Complementarity of their use in Lake Geneva (France–Switzerland). *Ecol. Indic.* 53, 231–239. doi:10.1016/J.ECOLIND.2015.02.008.

Rimet, F., Kurmayer, R., Salmaso, N., Capelli, C., Chardon, C., Vautier, M., et al. (2021). updated version- Lake biofilms sampling for both downstream DNA analysis and microscopic counts. *protocols.io*. doi:dx.doi.org/10.17504/protocols.io.br2xm8fn.

Rimet, F., Vautier, M., Kurmayer, R., Salmaso, N., Capelli, C., Bouchez, A., et al. (2020). River biofilms sampling for both downstream DNA analysis and microscopic counts. *protocols.io*. doi:dx.doi.org/10.17504/protocols.io.ben6jdhe.

Salmaso, N. (2019). Effects of habitat partitioning on the distribution of bacterioplankton in deep lakes. *Front. Microbiol.* 10, 2257. doi:10.3389/fmicb.2019.02257.

Salmaso, N., Albanese, D., Capelli, C., Boscaini, A., Pindo, M., and Donati, C. (2018). Diversity and Cyclical Seasonal Transitions in the Bacterial Community in a Large and Deep Perialpine Lake. *Microb. Ecol.* 76, 125–143. doi:10.1007/s00248-017-1120-x.

Salmaso, N., Boscaini, A., and Pindo, M. (2020). Unraveling the diversity of eukaryotic microplankton in a large and deep perialpine lake using a high throughput sequencing approach. *Front. Microbiol.* 11, 789. doi:10.3389/fmicb.2020.00789.

Salmaso, N., Boscaini, A., and Pindo, M. (2021a). EAW FASTQ files for bioinformatic courses (18S rRNA genes, 2018). doi:10.5281/zenodo.5215919.

Salmaso, N., Riccioni, G., Pindo, M., Vasselon, V., Domaizon, I., and Kurmayer, R. (2021b). Metabarcoding protocol – Analysis of Bacteria (including Cyanobacteria) using the 16S rRNA gene and a dada2 pipeline. *Zenodo*. doi:10.5281/zenodo.5232772.

Schwelm, A., Badstöber, J., Bulman, S., Desoignies, N., Etemadi, M., Falloon, R. E., et al. (2018). Not in your usual Top 10: protists that infect plants and algae. *Mol. Plant Pathol.* 19, 1029–1044. doi:10.1111/mpp.12580.

Shams, S. (2015). Diversity, impact and fate of cyanobacterial toxins in freshwater ecosystems.

Simpson, A. G. B., Slamovits, C. H., and Archibald, J. M. (2017). “Protist Diversity and Eukaryote Phylogeny,” in *Handbook of the Protists*, eds. J. M. A. Simpson, A. G. B., and C. H. Slamovits (Cham, Switzerland: Springer International Publishing AG), 1–21.

Soares, M. C. S., Lobão, L. M., Vidal, L. O., Noyma, N. P., Barros, N. O., Cardoso, S. J., et al. (2011). Light microscopy in aquatic ecology: methods for plankton communities studies. *Methods Mol. Biol.* 689, 215–227. doi:10.1007/978-1-60761-950-5_13.

Sournia, A. ed. (1978). “Phytoplankton manual,” in *Monographs on oceanographic methodology* (Paris: UNESCO), 1-337.

Stevenson, J. R., Bothwell, M. L., and Lowe, R. L. eds. (1996). *Algal Ecology - Freshwater Benthic Ecosytems*. San Diego, USA: Academic Press, Elsevier.

Stoeck, T., Bass, D., Nebel, M., Christen, R., Jones, M. D. M., Breiner, H. W., et al. (2010). Multiple marker parallel tag environmental DNA sequencing reveals a highly complex eukaryotic community in marine anoxic water. *Mol. Ecol.* 19, 21–31. doi:10.1111/j.1365-294X.2009.04480.x.

Tragin, M., Zingone, A., and Vaulot, D. (2018). Comparison of coastal phytoplankton composition estimated from the V4 and V9 regions of the 18S rRNA gene with a focus on photosynthetic groups and especially Chlorophyta. *Environ. Microbiol.* 20, 506–520. doi:10.1111/1462-2920.13952.

Vasselon, V., Rimet, F., Tapolczai, K., and Bouchez, A. (2017). Assessing ecological status with diatoms DNA metabarcoding: Scaling-up on a WFD monitoring network (Mayotte island, France). *Ecol. Indic.* 82, 1–12. doi:10.1016/j.ecolind.2017.06.024.

Vautier, M., Chardon, C., Capelli, C., Kurmayer, R., Salmaso, N., and Domaizon, I. (2021). Plankton DNA extraction from Sterivex filter units. *protocols.io*. doi:dx.doi.org/10.17504/protocols.io.bvgzn3x6.

Vautier, M., Vasselon, V., Chardon, C., Rimet, F., Bouchez, A., and Domaizon, I. (2020). DNA extraction from environmental biofilm using the NucleoSpin® Soil kit (MACHEREY-NAGEL). *protocols.io*. doi:dx.doi.org/10.17504/protocols.io.bd52i88e.

Wang, C., Zhang, T., Wang, Y., Katz, L. A., Gao, F., and Song, W. (2017). Disentangling sources of variation in SSU rDNA sequences from single cell analyses of ciliates: Impact of copy number variation and experimental error. *Proc. R. Soc. B Biol. Sci.* 284, 20170425.

doi:10.1098/rspb.2017.0425.

Wang, Q., Garrity, G. M., Tiedje, J. M., and Cole, J. R. (2007). Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Appl. Environ. Microbiol.* 73, 5261–7. doi:10.1128/AEM.00062-07.

Water Framework Directive (2000). Directive 2000/60/EC of the European Parliament and of the Council of 23 October 2000 establishing a framework for Community action in the field of water policy. *Off. J. Eur. Parliam.* doi:10.1039/ap9842100196.

Wehr, J. D., and Sheath, R. G. (2003). *Freshwater Algae of North America - Ecology and Classification*. Academic Press, Elsevier.

Weisse, T., Anderson, R., Arndt, H., Calbet, A., Hansen, P. J., and Montagnes, D. J. S. (2016). Functional ecology of aquatic phagotrophic protists – Concepts, limitations, and perspectives. *Eur. J. Protistol.* 55, 50–74. doi:10.1016/j.ejop.2016.03.003.

Wilmotte, A., Laughinghouse, H. D. I., Capelli, C., Rippka, R., and Salmaso, N. (2017). “Taxonomic identification of cyanobacteria by a polyphasic approach,” in *Molecular tools for the detection and quantification of toxigenic cyanobacteria*, eds. R. Kurmayer, K. Sivonen, A. Wilmotte, and N. Salmaso (John Wiley), 79–119.