Metabarcoding protocol – Analysis of Bacteria (including Cyanobacteria) using the 16S rRNA gene and a DADA2 pipeline

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

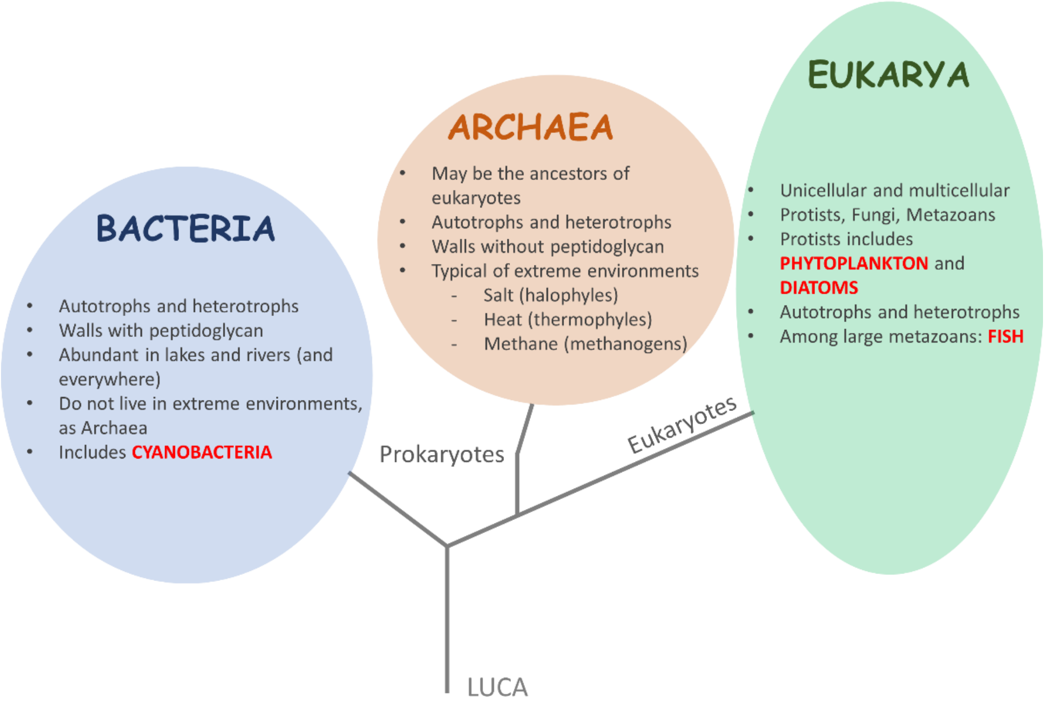
[1] Research and Innovation Centre, Fondazione Edmund Mach, San Michele all'Adige, Italy
[2] Scimabio Interface SAS, Thonon-les-Bains, France
[3] French National Institute for Agriculture, Food and Environment, Thonon les Bains, France
[4] University of Innsbruck, Mondsee, Austria

[*]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 – 1, https://www.alpine-space.eu/projects/eco-alpswater/en/home

1. Introduction

Prokaryotes are represented by two Kingdoms, i.e. Bacteria (Eubacteria) and Archaea (Fig. 1). In contrast to eukaryotic organisms, prokaryotes do not contain a distinct nucleus bounded by a nuclear envelope; moreover, they lack membrane-bound organelles, such as chloroplasts and mitochondria.



BACTERIA

- Autotrophs and heterotrophs
- Walls with peptidoglycan
- Abundant in lakes and rivers (and everywhere)
- Do not live in extreme environments, as Archaea
- Includes **CYANOBACTERIA**

ARCHAEA

- May be the ancestors of eukaryotes
- Autotrophs and heterotrophs
- Walls without peptidoglycan
- Typical of extreme environments
  - Salt (halophyles)
  - Heat (thermophyles)
  - Methane (methanogens)

EUKARYA

- Unicellular and multicellular
- Protists, Fungi, Metazoans
- Protists includes **PHYTOPLANKTON** and **DIATOMS**
- Autotrophs and heterotrophs
- Among large metazoans: **FISH**

Fig. 1 – The three Domains of life. The biological elements included in the monitoring plans by the project EAW are among those considered in the Water Framework Directive (2000), i.e. Cyanobacteria (Bacteria), and phytoplankton/diatoms and fish (Eukarya), highlighted in red. Archaea, being mostly segregated to extreme environments, are not included among the biological elements used to evaluate water quality in lakes and rivers. Nevertheless, among Bacteria, only the oxygenic photosynthetic Cyanobacteria are included in the WFD monitoring plans. Similarly, among eukaryotic planktic microorganisms (protists), only the photosynthetic/mixotrophic microalgae (phytoplankton and periphitic algae) are considered. In classical ecological investigations and monitoring, eukaryotic pelagic microalgae and Cyanobacteria are usually put together to form a comprehensive “phytoplankton” ecological group (Reynolds, 2006). Traditionally, phytoplankton has been studied using light microscopy (LM).Nonetheless , owing to the microscopic size of organisms and the limited number of morphological diacrytical features, LM is able to provide reliable identification only for the largest organisms (Canter-Lund and Lund, 1995). LUCA, last universal common ancestor (Theobald, 2010).

Differences between Bacteria and Archaea are both physiological and ecological (Campbell et al., 2008). Bacteria include the majority of prokaryotes that drive ecosystemic processes, quality of ecosystems (including water quality) and human life. Bacteria include many pathogenic species that can cause serious diseases, as well as many beneficial species responsible of important biogeochemical processes (e.g. nitrogen, carbon, and sulphur cycles) or living in symbiosis with higher organisms, including man. Among bacteria, photosynthetic Cyanobacteria (oxyphotobacteria) have a particular importance for the impact they can have on water quality and human health. In eutrophic water bodies, these organisms can develop massive water blooms, inducing discoloration of water and accumulation of organic substance at the surface of lakes and rivers (Reynolds and Walsby, 1975; Whitton, 2012). Moreover, Cyanobacteria include many taxa able to produce a wide variety of toxins (cyanotoxins), which are harmful to humans and many other aquatic organisms or animals drinking contaminated waters (Bernard et al., 2017; Kurmayer et al., 2017; Meriluoto et al., 2017). Conversely, Archaea are abundant in extreme environments, and are adapted e.g. to high levels of salinity and heat, and involved in specific biogeochemical processes (e.g. methanogenesis) (Fig. 1). For these reasons, and also considering that the most dangerous toxigenic cyanobacteria can be identified by LM, only this group of oxygenic photosynthetic bacteria have been included in the water quality monitoring plans by the WFD. Though recognizing the importance of the whole community of pathogenic bacteria (Pandey et al., 2014), generally only a fraction of selected indicator pathogens are included in health monitoring plans (Quilliam et al., 2011).

This deliverable provides the basic elements that have been used for the identification of bacteria and cyanobacteria within the project Eco-AlpsWater using a pipeline adapted for the identification of amplicon sequence variants (ASVs) using the DADA2 protocol (Callahan et al., 2016a, 2016b). The bioinformatic analysis has been applied to DNA extracted from samples collected in the water column of lakes and biofilms collected in rivers and lakeshores; for sampling and laboratory methods, see Domaizon et al. (2019), Rimet et al. (2020; 2021), Vautier et al. (2020, 2021). An overview of the analyses carried out using the 16S and 18S rRNA gene, *rbcL* (diatoms) and 12S rRNA gene (fish) markers in the context of the project Eco-AlpsWater is reported in Salmaso et al. (2021b).

2. Selection of primers

PCR amplification of the 16S rRNA genes has been performed by targeting a ~ 460-bp fragment in the variable regions V3–V4 of the total genomic DNA extracted from environmental samples. The bacterial primer set includes:

341F (5′ CCTACGGGNGGCWGCAG 3′) (Herlemann et al., 2011; Klindworth et al., 2013), and 805Rmod (5′ GACTACNVGGGTWTCTAATCC 3′) (Apprill et al., 2015) with overhang Illumina adapters.

This pair of primers, or primers with slight modifications, has been widely used in the assessment of bacterial biodiversity in aquatic environments, including the large lakes south of the Alps investigated in the project Eco-AlpsWater (Salmaso et al., 2018; Salmaso, 2019). A general evaluation of 16S rRNA gene primer sets has been carried out in the comprehensive study by Klindworth et al. (2013).

3. Wet lab, amplification and HTS

The preparation of library followed a standard procedure in use at the FEM sequencing facility directed by M. Pindo, as described e.g. in (Salmaso et al., 2018, 2020).

4. Bioinformatic pipeline

In general, different approaches can be adopted for the analysis of HTS 16S rRNA gene reads. These can be based 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, which identify amplicon sequence variants, ASVs, also known as exact sequence variants, ESVs. As for the latter approach

(ASVs), a number of methods have been proposed, including DADA2 (Callahan et al., 2016a), DEBLUR (Amir et al., 2017), UNOISE 2 and 3 (Edgar, 2016); a few of them have been implemented in QIIME2 (DADA2 and DEBLUR; Bolyen et al., 2019) or adapted in VSEARCH (such as UNOISE3) (Rognes et al., 2016). The usefulness of denoising approaches compared to OTUs methods has been substantiated in a number of investigations using bacterial (Glassman and Martiny, 2018; Nearing et al., 2018; Caruso et al., 2019; Prodan et al., 2020) and fungal (Pauvert et al., 2019) mock communities.

This deliverable reports a pipeline (DADA2 Pipeline Tutorial 1.18), based on DADA2 v. 1.20.0, 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., 2016a, 2018). The pipeline has been tested on a machine equipped with an i7 9700K and 64 Gb of RAM, under Linux Ubuntu 20.10 and R 4.1.0[1]. The analysis of larger datasets (>100 pairs of F and R FASTQ files) would require the use of High Performance Computing (HPC) facilities equipped with multicore 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).

4.1 Download of FASTQ files, and preliminary processing

A selection of 6 samples, with 6 Forward (R1) and 6 Reverse (R2) files will be 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 the16S rRNA gene reads obtained from the analyses carried out on the samples collected and filtered (Sterivex$^{\text{TM}}$ 0.22 µm) in different stations of Lake Garda on September 2018 (Fig. 2).

Forward (R1)	Reverse (R2)	Code (Fig. 2)
• ECOALPSWATER-16S-386-09-18-0stv_S131_L001_R1_001.fastq.gz	• ECOALPSWATER-16S-386-09-18-0stv_S131_L001_R2_001.fastq.gz	S0
• ECOALPSWATER-16S-386-09-18-4stv_S132_L001_R1_001.fastq.gz	• ECOALPSWATER-16S-386-09-18-4stv_S132_L001_R2_001.fastq.gz	S4
• ECOALPSWATER-16S-B0918D1stv_S127_L001_R1_001.fastq.gz	• ECOALPSWATER-16S-B0918D1stv_S127_L001_R2_001.fastq.gz	C0
• ECOALPSWATER-16S-B0918D4stv_S128_L001_R1_001.fastq.gz	• ECOALPSWATER-16S-B0918D4stv_S128_L001_R2_001.fastq.gz	C100
• ECOALPSWATER-16S-B0918D5stv_S129_L001_R1_001.fastq.gz	• ECOALPSWATER-16S-B0918D5stv_S129_L001_R2_001.fastq.gz	C300
• ECOALPSWATER-16S-Porto0918stv_S130_L001_R1_001.fastq.gz	• ECOALPSWATER-16S-Porto0918stv_S130_L001_R2_001.fastq.gz	H0

The 12 files are stored in the Zenodo archives (Salmaso et al., 2021; https://zenodo.org/record/5215815#.YSHzg0txeHs). Download the files in a working directory, under your home dir, e.g., `~/EAW16S`.

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 are trimmed before the application of the DADA2 bioinformatic pipeline using Cutadapt[2] (see below). In the second case, primers are trimmed by using an internal procedure implemented in DADA2 (argument `trimLeft` in the function `filterAndTrim`); this last procedure is presented in the **Appendix 1**.

[1] This pipeline can be used under Windows 10 only adapting the working directories, i.e. using, e.g.,:
path <- "c:/EAW16S/" (see below)

[2] Cutadapt is only one among several options that can be used to trim primers. Results obtained with different tools are often not comparable (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.

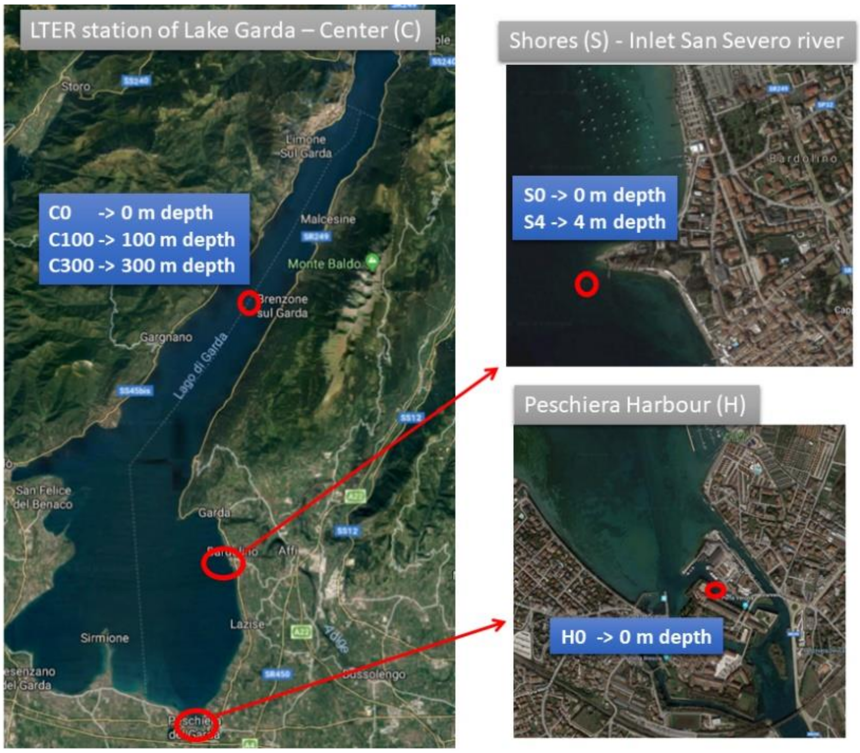


Fig. 2 – Location of the sampling stations considered in this protocol.

A 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; besides single couples of paired files, the application works also on multiple datasets. The Galaxy platform runs under Windows and UNIX-like operating systems. Nevertheless, all the operations have to be completed using menus, and the computations can be slow.

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 341F and 805Rmod used in this protocol. After opening the terminal, enter the directory `~/EAW16S`, and download rmprim.sh:

```
cd ~/EAW16S
wget https://raw.githubusercontent.com/hts-tools/metatools/master/rmprim/rmprim.sh
```

Primers can be removed using the script[3]:

```
bash rmprim.sh -f CCTACGGGNGGCWGCAG -r GACTACNVGGGTWTCTAATCC -n TRUE -t TRUE -d ~/EAW16S
```

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

4.2 Installation of DADA2 and loading of the package in R

For the installation of DADA2, follow the instructions reported in the dedicated bioconductor webpage, https://www.bioconductor.org/packages/release/bioc/html/dada2.html ; see also Salmaso et al. (2021b). Other required packages are tidyverse, openxlsx and vegan.

Start R and load the package dada2:

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

In the following steps, the input directory ~/EAW16S is saved in the variable “path”. Moreover, to keep things in order, other sub-directories are created under ~/EAW16S with the following lines:

```
path <- "~/EAW16S"
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[4]:

```
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)
sample.names
```

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

[3] 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`”

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

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

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 bases with Q-scores < 30. Truncation, however, should allow overlapping of R1 and R2 reads in successive analyses. In this exercise, R1 and R2 reads are truncated at 258 and 205, respectively, allowing a final overlap of around >35 bp bewteen R1 and R2 reads[5].

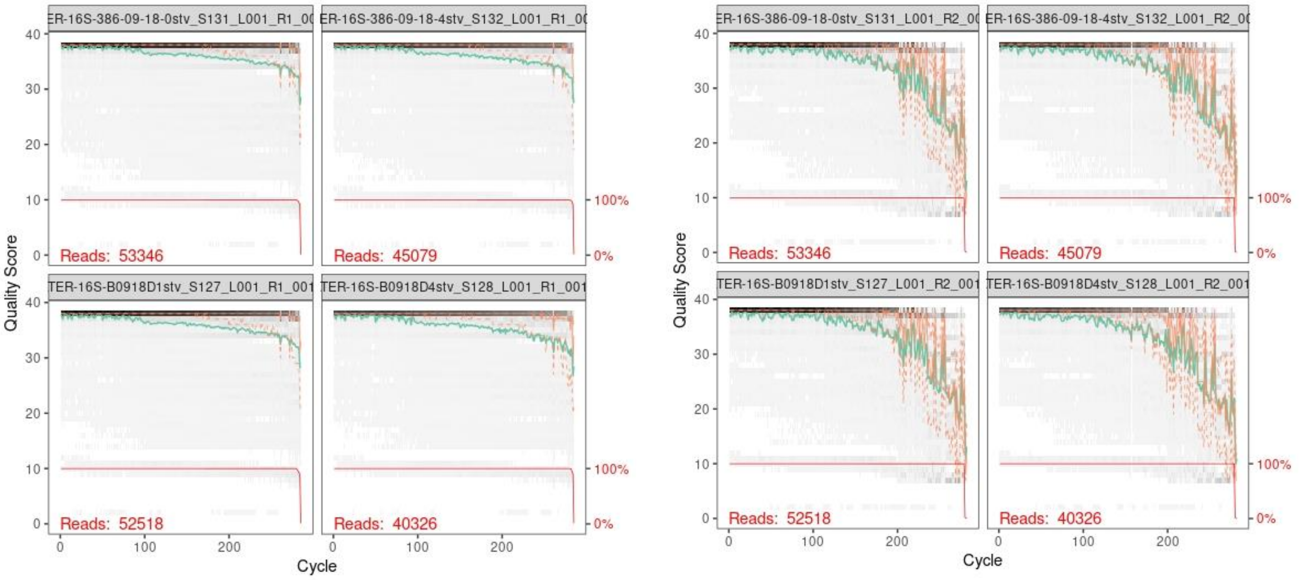


Fig. 3 – Quality profiles of the forward (R1, left) and reverse (R2, rigth) reads (primers trimmed). Quality scores are encoded in the FASTQ files (fourth line of each single read). The bases are along the horizontal axis, whereas the quality scores are reported on the vertical axis. The gray-scale is 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). For a complete quality assessment of FASTQ files, see the functions *qa* and *report* in the package ShortRead (Morgan et al., 2009)

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 `"~/EAW16S/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

[5] 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; https://benjjneb.github.io/dada2/tutorial.html.

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(258,205),
maxEE=c(1,1), matchIDs=TRUE, maxN=0, rm.phix=TRUE, multithread=TRUE, verbose=TRUE)
out # On Windows, multithread is not supported
```

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])
```

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., 2016a). 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)
```

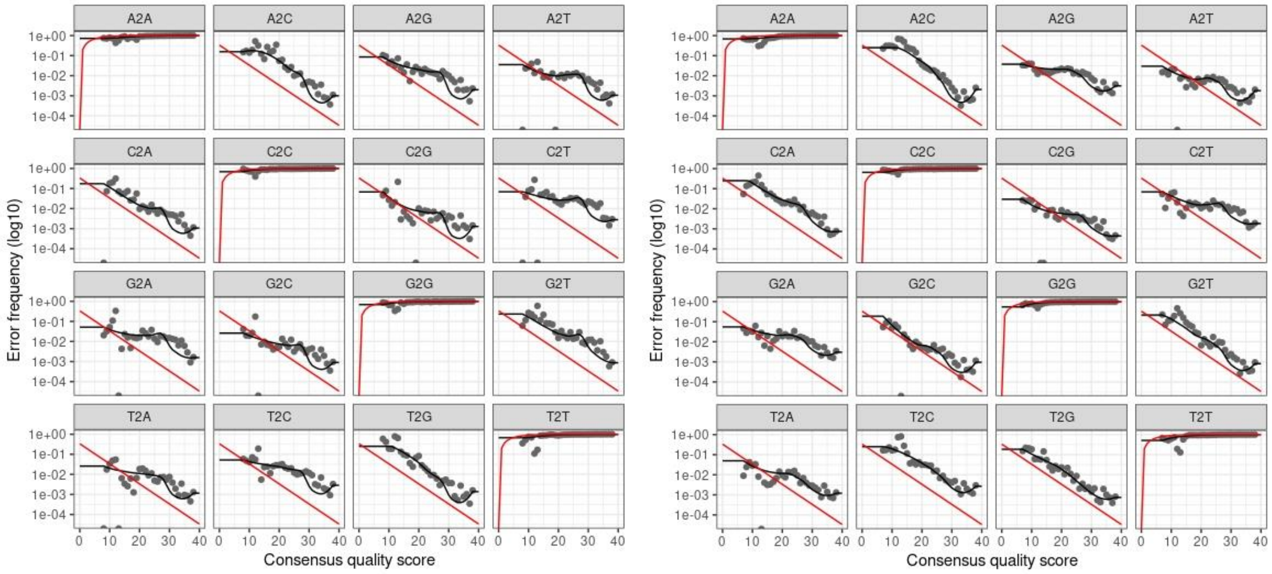


Fig. 4 – Visualization of the estimated error rates. Each single graph shows the error rates for each possible transition (e.g. A→C, A→G, … T→G). The red line is what is expected based on the quality score; the black line is the estimate, whereas the black dots are the observed. Overall, the observed,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

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 biological origin), or spurious (Callahan et al., 2016a). 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` (default), 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. Nevertheless, running all the samples together can be computationally not feasible on common computers when datasets are large. In those cases, set `pool = FALSE` or `“pseudo”`. Estimated error rates are reported in Fig. 4.

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. Also considering the fraction of merged reads (see table below), in this dataset, the minimum overlap has been fixed at 35. No mismatches are allowed in the overlap region.

```
merged_reads <- mergePairs(dadaFs, filtFs, dadaRs, filtRs, verbose=TRUE, minOverlap=35,
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 (2349).

```
plot(table(nchar(getSequences(seqtaball))), xlab="Reads R1+R2 merged length")
```

A fraction of lengths in the merged sequences do not fall within the expected range for this V3-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.

```
seqtab <- seqtaball[,nchar(colnames(seqtaball)) %in% seq(400,430)]
dim(seqtab)
table(nchar(getSequences(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.

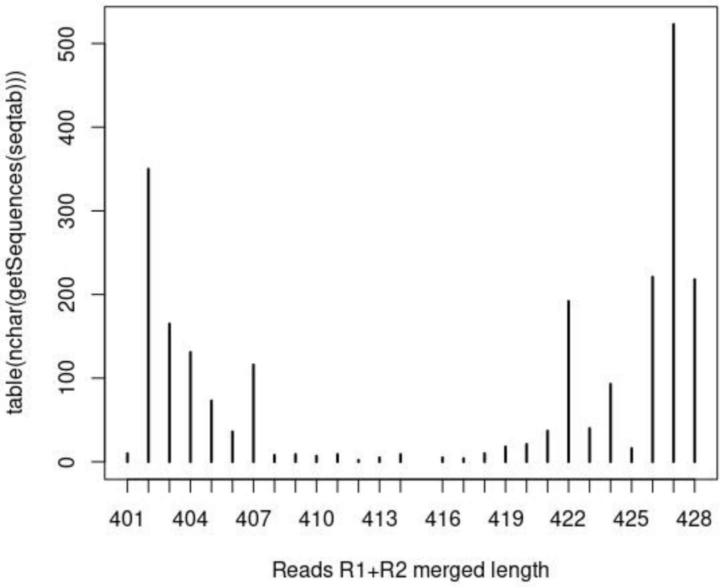


Fig. 5 – Number of reads (y) with specific amplicon lengths (x) in the V3-V4 sequenced region. The bimodal distribution in the V3-V4 region is expected, and is due to variation in the lengths of amplicons in the V3 region, which is composed of 2 main peaks at 161 and 186 nt, and minor peaks at 166 and 181 nt (whole V3 length); conversely, variation in the V4 region is very limited (282-284 nt, whole length) (Vargas-Albores et al., 2017).

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-16S-386-09-18-0stv	53346	42636	39741	40335	37091	36818
ECOALPSWATER-16S-386-09-18-4stv	45079	35784	34351	34862	33224	32766
ECOALPSWATER-16S-B0918D1stv	52518	41841	41015	41395	39971	39654
ECOALPSWATER-16S-B0918D4stv	40326	31009	30226	30437	29216	28048
ECOALPSWATER-16S-B0918D5stv	55785	43871	42904	43025	41385	39937
ECOALPSWATER-16S-Porto0918stv	60798	48923	47382	47841	46018	45550

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). A selection of taxonomic databases have been adapted for the use with DADA2. Here we will rely on the last version of the SILVA database formatted for DADA2, v. 138.1[6] (March 2021). Download the files `silva_nr99_v138.1_train_set.fa.gz` and `silva_species_assignment_v138.1.fa.gz`, saving them in the directory `~/EAW16S/Taxonomy/` (skip this step if the taxonomy file was already downloaded):

```
download.file("https://zenodo.org/record/4587955/files/silva_nr99_v138.1_train_set.fa.gz"
, paste0(pathtax, "silva_nr99_v138.1_train_set.fa.gz"))
download.file("https://zenodo.org/record/4587955/files/silva_species_assignment_v138.1.fa
.gz", paste0(pathtax, "silva_species_assignment_v138.1.fa.gz"))
```

The minimum bootstrap confidence for assigning a taxonomic level has been set to 95 (default=50). Depending on the number of sequences, this step could be computationally demanding, requiring a high amount of RAM memory.

```
taxa138a <- assignTaxonomy(seqtab.nochim, paste0(pathtax,
"silva_nr99_v138.1_train_set.fa.gz"), multithread=TRUE, minBoot = 95, verbose = TRUE)
```

The successive step is the taxonomic assignment at the species level, based on exact matching between ASVs and reference strains. The option allowMultiple allows obtaining a concatenated string of all exactly matched species.

```
taxa138 <- addSpecies(taxa138a, paste0(pathtax, "silva_species_assignment_v138.1.fa.gz"),
allowMultiple = TRUE)
```

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

```
save.image(paste0(pathana, "EAW16S_analysis.RData"))
```

4.9 Collecting dada2 results: saving tables for downstream statistical analyses

[6] Previous analyses of the EAW 16S rRNA gene sequences made in December 2020 were performed on the DADA2 v. 138 (August 2020) reference databases, i.e. “silva_nr99_v138_train_set.fa.gz” and “silva_species_assignment_v138.fa.gz”. In case of updating the taxonomic databases, their name in the scripts will have to be changed accordingly.

```
# 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 <- taxa138
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_16S", firstRow = TRUE, zoom = 90,
keepNA = TRUE)
```

The above files can be imported/read 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,
check.names=FALSE)
head(counts)
counts2 <- t(counts)^0.25
bc <- vegdist(counts2, method="bray")
plot(hclust(bc))
```

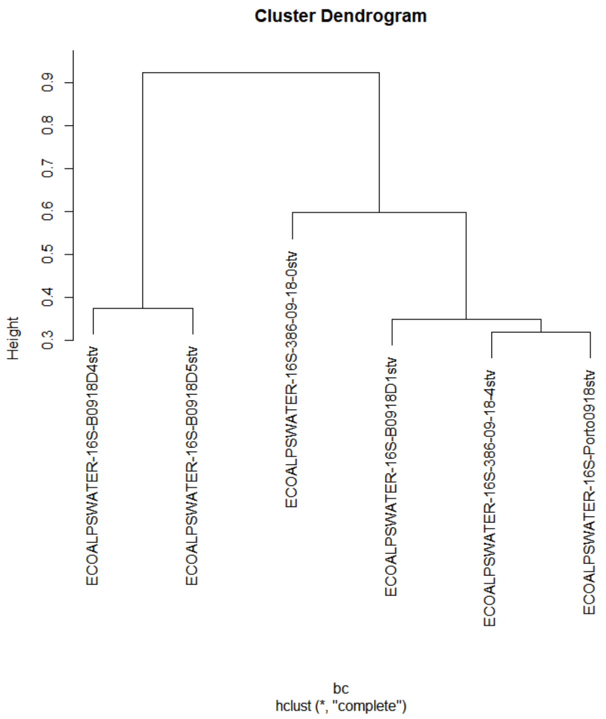


Fig. 6 - Quick and dirty cluster analysis made importing the file counts.tsv in R. Data preliminarily transformed by double square root. Note how the deep samples are isolated from the surface samples.

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_16S_metadata.ods” (in Zenodo, https://doi.org/10.5281/zenodo.5215815), delete the first line, and save in .CSV format under the dir “~/EAW16S”. Import the metadadata in R (check the parameter “sep”: it can be either “;” or ”,”):

```
ambio <- read.csv(file="EAW_2018_16S_metadata.csv", header=T, row.names=1, sep = ";")
ambio
```

Create a phyloseq object, and save it under the dir “~/EAW16S/Analysis”,

```
taxtable_ps <- taxa138
row.names(taxtable_ps) <- SeqName
eawps16S <- phyloseq(otu_table(counts, taxa_are_rows=TRUE), sample_data(ambio),
tax_table(taxtable_ps))
eawps16S
saveRDS(eawps16S, 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):

```
eawps16S <- readRDS(file = paste0(pathana, "EAW16S_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. In more complex situations, for example when the F and R reads extend into the opposite primers which will appear in their reverse complement form towards the ends of reads, the option `filterAndTrim` cannot be used; the typical case is the ITS region, for which specific tools, such as cutadapt (https://cutadapt.readthedocs.io), can be used; see also http://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 from https://zenodo.org/record/5215815#.YSE540txeHs, save them to the directory ~/EAW16S 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. 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 bases with Q-scores < 30. Truncation, however, should allow overlapping of R1 and R2 reads in successive analyses. In this exercise, R1 and R2 reads will be truncated at 275 and 226, respectively, allowing a final overlap of around >35 bp bewteen R1 and R2 reads.

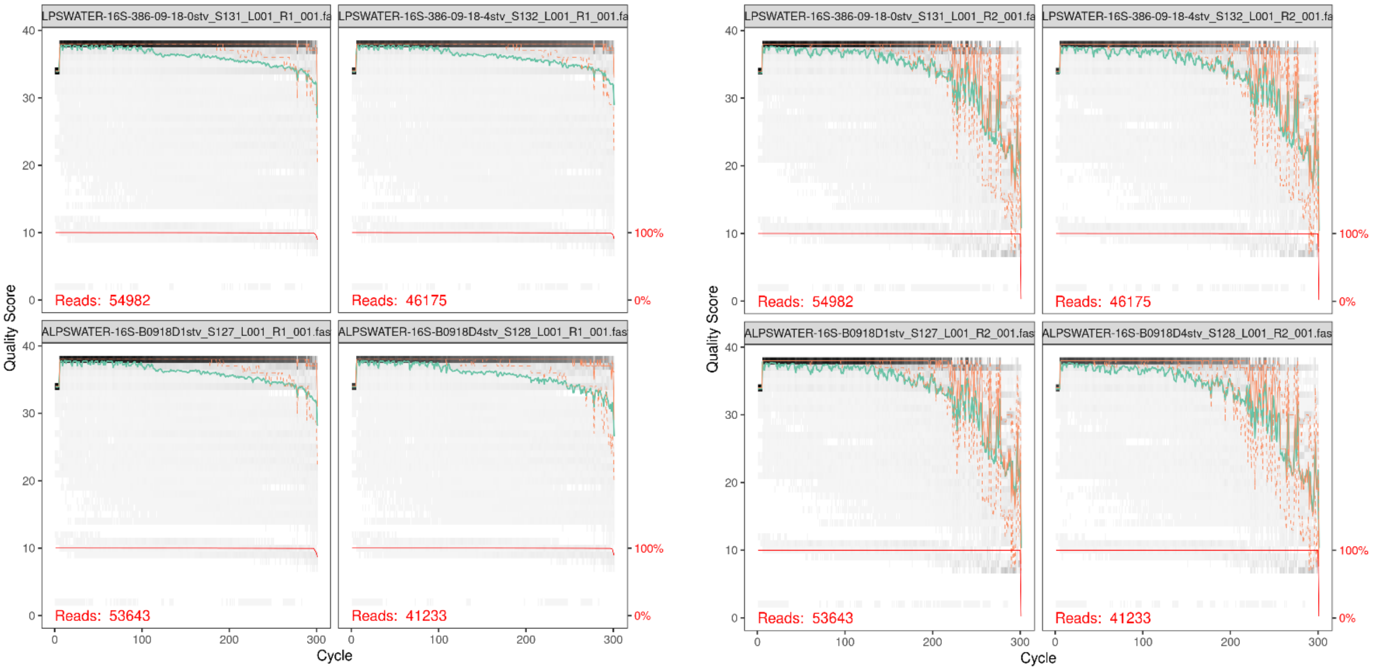


Fig. 7 – Quality profiles of the forward (R1, left) and reverse (R2, rigth) reads. These reads still include the primers.

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, for common primer design, where primers of fixed length (in this exercise 17 and 21 nt long, respectively, for R1 and R2) are at the start of the R1 and R2 reads, primers can be removed in the `filterAndTrim` step using the argument `trimLeft`.[7] The new filtered fastq are saved in the directory `"~/EAW16S/filtered/"`.

```
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(275,226),
trimLeft=c(17, 21), maxEE=c(1,1), matchIDs=TRUE, maxN=0, rm.phix=TRUE, multithread=TRUE,
verbose=TRUE)
out # On Windows, set multithread=FALSE
```

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.

***

[7] If both truncLen and trimLeft arguments are used, the resulting filtered reads will have length=truncLen-trimLeft

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.

Apprill, A., McNally, S., Parsons, R., and Laura Weber (2015). Minor revision to V4 region SSU rRNA 806R gene primer greatly increases detection of SAR11 bacterioplankton. *Aquat. Microb. Ecol.* 75, 129–137. Available at: http://www.int-res.com/abstracts/ame/v75/n2/p129-137/ [Accessed March 14, 2016].

Bernard, C., Ballot, A., Thomazeau, S., Maloufi, S., Furey, A., Mankiewicz-Boczek, J., et al. (2017). "Appendix 2. Cyanobacteria associated with the production of cyanotoxins," in *Handbook on Cyanobacterial Monitoring and Cyanotoxin Analysis*, eds. J. Meriluoto, L. Spoof, and G. A. Codd (Wiley, Chichester), 501–525.

Bolyen, E., Rideout, J. R., Dillon, M. R., Bokulich, N. A., Abnet, C. C., Al-Ghalith, G. A., et al. (2019). Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. *Nat. Biotechnol.* 37, 852–857. doi:10.1038/s41587-019-0209-9.

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

Callahan, B. J., Sankaran, K., Fukuyama, J. A., McMurdie, P. J., and Holmes, S. P. (2016b). Bioconductor Workflow for Microbiome Data Analysis: from raw reads to community analyses. *F1000Research* 5, 1492. doi:10.12688/f1000research.8986.2.

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

Campbell, N. A., Reece, J. B., Urry, L. A., Cain, M. L., A., W. S., Minorsky, P. V., et al. (2008). *Biology*. Eighth. San Francisco: Pearson Benjamin Cummings.

Canter-Lund, H., and Lund, J. W. G. (1995). *Freshwater algae, their microscopic world explored*. Bristol: Biopress Ltd.

Caruso, V., Song, X., Asquith, M., and Karstens, L. (2019). Performance of Microbiome Sequence Inference Methods in Environments with Varying Biomass. *mSystems* 4. doi:10.1128/msystems.00163-18.

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.

Glassman, S. I., and Martiny, J. B. H. (2018). Broadscale Ecological Patterns Are Robust to Use of Exact Sequence Variants versus Operational Taxonomic Units. *mSphere* 3. doi:10.1128/msphere.00148-18.

Herlemann, D. P., Labrenz, M., Jürgens, K., Bertilsson, S., Waniek, J. J., and Andersson, A. F. (2011). Transitions in bacterial communities along the 2000 km salinity gradient of the Baltic Sea. *ISME J.* 5, 1571–9. doi:10.1038/ismej.2011.41.

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.

Klindworth, A., Pruesse, E., Schweer, T., Peplies, J., Quast, C., Horn, M., et al. (2013). Evaluation of general 16S ribosomal RNA gene PCR primers for classical and next-generation sequencing-based diversity studies. *Nucleic Acids Res.* 41, e1. doi:10.1093/nar/gks808.

Kurmayer, R., Sivonen, K., Wilmotte, A., and Salmaso, N. (2017). *Molecular Tools for the Detection and Quantification of Toxigenic Cyanobacteria*. Wiley, Chichester.

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

McMurdie, P. J., and Holmes, S. (2013). Phyloseq: An R Package for Reproducible Interactive Analysis and Graphics of Microbiome Census Data. *PLoS One* 8, e61217. doi:10.1371/journal.pone.0061217.

Meriluoto, J., Blaha, L., Bojadzija, G., Bormans, M., Brient, L., Codd, G. A., et al. (2017). Toxic cyanobacteria and cyanotoxins in European waters – recent progress achieved through the CYANOCOST Action and challenges for further research. *Adv. Oceanogr. Limnol.* 8, 161–178. doi:10.4081/aiol.2017.6429.

Morgan, M., Anders, S., Lawrence, M., Aboyoun, P., Pagès, H., and Gentleman, R. (2009). ShortRead: A bioconductor package for input, quality assessment and exploration of high-throughput sequence data. *Bioinformatics* 25, 2607–2608. doi:10.1093/bioinformatics/btp450.

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.

Pandey, P. K., Kass, P. H., Soupir, M. L., Biswas, S., and Singh, V. P. (2014). Contamination of water resources by pathogenic bacteria. *AMB Express* 4, 1–16. doi:10.1186/s13568-014-0051-x.

Pauvert, C., Buée, M., Laval, V., Edel-Hermann, V., Fauchery, L., Gautier, A., et al. (2019). Bioinformatics matters: The accuracy of plant and soil fungal community data is highly dependent on the metabarcoding pipeline. *Fungal Ecol.* 41, 23–33. doi:10.1016/j.funeco.2019.03.005.

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.

Quilliam, R. S., Clements, K., Duce, C., Cottrill, S. B., Malham, S. K., and Jones, D. L. (2011). Spatial variation of waterborne Escherichia coli - Implications for routine water quality monitoring. *J. Water Health* 9, 734–737. doi:10.2166/wh.2011.057.

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

Reynolds, C. S., and Walsby, A. E. (1975). Water-Blooms. *Biol. Rev.* 50, 437–481. doi:10.1111/j.1469-185X.1975.tb01060.x.

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.

Rognes, T., Flouri, T., Nichols, B., Quince, C., and Mahé, F. (2016). VSEARCH: a versatile open source tool for metagenomics. *PeerJ* 4, e2584. doi:10.7717/peerj.2584.

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 (16S rRNA genes, 2018). doi:10.5281/ZENODO.5215815.

Salmaso, N., Riccioni, G., Pindo, M., Kurmayer, R., Vasselon, V., and Domaizon, I. (2021b). Metabarcoding protocol – Analysis of protists using the 18S rRNA gene and a DADA2 pipeline. *Zenodo*. doi:10.5281/zenodo.5233527.

Theobald, D. L. (2010). A formal test of the theory of universal common ancestry. *Nature* 465, 219–222. doi:10.1038/nature09014.

Vargas-Albores, F., Ortiz-Suárez, L. E., Villalpando-Canchola, E., and Martínez-Porchas, M. (2017). Size-variable zone in V3 region of 16S rRNA. *RNA Biol.* 14, 1514–1521. doi:10.1080/15476286.2017.1317912.

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, 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.

Whitton, B. A. ed. (2012). *Ecology of Cyanobacteria II Their Diversity in Space and Time*. 2nd ed. Springer Dordrecht doi:DOI 10.1007/978-94-007-3855-3.