Eco-AlpsWater

Innovative Ecological Assessment and Water Management Strategy for the Protection of Ecosystem Services in Alpine Lakes and Rivers

Priority 3: Liveable Alpine Space. SO3.2 - Enhance the protection, the conservation and the ecological connectivity of Alpine Space

Project Eco-AlpsWater
Work Package WPT1
Activity A.T1.1
Deliverable D.T1.1.3. – 1
Version 1.0
Date December 2018
Coordination: I Domaizon

Deliverable D.T1.1.3

Diatom DNA metabarcoding bioinformatics pipeline *"Mothur" software, Miseq, rbcL 312 bp*

Interreg Alpine Space - Eco-AlpsWater project – WP1

Author : Valentin Vasselon[1,2].

[1] *AFB, Aix en Provence, France*
[2] *INRA, CARRTEL, Thonon les bains, France*

ABSTRACT

This protocol describes in details the main steps of the bioinformatics process applied to treat high throughput sequencing (HTS) data, in particular for Diatoms metabarcoding.
This protocol is one of those proposed by the EcoAlpsWater consortium to promote the implementation of HTS of environmental DNA in the biomonitoring and ecological assessment of water bodies (lakes and rivers).

Diatom DNA metabarcoding bioinformatics pipeline

"Mothur" software, Miseq, rbcL 312 bp

The following bioinformatical pipeline uses the "Mothur" software (Schloss *et al.*, 2009) to process DNA reads produced by High-Throughput Sequencing technologies (Illumina MiSeq), from raw data to final OTU/Taxonomic inventories. A standard pipeline for 16S rRNA metabarcoding data is proposed online (Kozich *et al.*, 2013, MiSeq SOP, https://www.mothur.org/wiki/MiSeq_SOP) and is regularly updated. Here we present an alternative pipeline adapted to diatom DNA metabarcoding and already applied in different studies targeting benthic diatom communities from lakes (e.g. Rimet *et al.*, 2018; Rivera *et al.*, 2018) and rivers (Vasselon *et al.*, 2017a b; Keck *et al.*, 2018a). Description of the different commands and the parameters used are also found in the Mothur wiki (https://www.mothur.org/wiki/).

1 Demultiplexing and Contig steps

After MiSeq paired-end sequencing, 2 *fastq* files are provided with R1.fastq including forward DNA reads and with R2.fastq corresponding to the reverse DNA reads. First bioinformatical steps consist i) to contig the forward and reverse DNA read in order to only keep one consensus sequence with a good overlapping score, ii) to demultiplex DNA reads in order to sort DNA reads according to their sample origin using dual indices included during HTS library preparation. Those steps can be performed using Mothur using the following commands with an oligos file including the *rbc*L primer sequences and the sample dual indices (https://www.mothur.org/wiki/Oligos_File):

```
make.file(inputdir=D:\Example, type=fastq, prefix=stability)
make.contigs(file=stability.files, checkorient=t)
screen.seqs(fasta=stability.trim.contigs.fasta, qfile=stability.trim.contigs.qual,
contigsreport=stability.contigs.report, maxambig=0, minoverlap=140)
```

As the quality of the sequencing may vary according to the chemistry, sequencing machine, sequencing platform or the samples used, we preferred to let the sequencing platform perform those steps. They can easily adjust filtering parameters of the contig step (e.g. minimum length for the overlapping region, % of mismatches between forward and reverse reads…) according to the quality of HTS data they obtained.

2 Quality trimming of DNA reads

At this point, the sequencing platform provided one *fastq* file per sample which correspond to DNA reads already contiged and demultiplexed. The following commands will be applied for each sample to perform the trimming steps:

```
#Extract the fasta and the qual (PHRED quality score) files from the fastq file
fastq.info(fastq=1.fastq)
#Trim DNA reads according to different criteria (read quality, length, "N", homopolymers, forward primer)
trim.seqs(fasta=1.fasta, qfile=1.qual, qwindowaverage=23, qwindowsize=25, minlength=280, maxlength=340,
maxambig=0, maxhomop=8, keepforward=t, checkorient=t, oligos=forward_new.oligos, pdiffs=1, processors=1)
#Successfully trimmed DNA reads are then trimmed focusing the reverse primer
```

trim.seqs(fasta=1.trim.fasta, qfile=1.trim.qual, qwindowaverage=23, qwindowsize=25, minlength=280, maxlength=340, maxambig=0, maxhomop=8, keepforward=t, checkorient=t, oligos=reverse_new.oligos, pdiffs=1, processors=1)

#Successfully trimmed DNA reads are lists and extracted from the original *fasta* file (1.fasta) to keep integrity of DNA reads name (trim.seqs command add information in DNA reads name)

list.seqs(fasta=1.trim.trim.fasta)

get.seqs(accnos=1.trim.trim.accnos, fasta=1.fasta, qfile=1.qual)

rename.file(input=1.pick.fasta, new=barcode1.fasta)

Those steps are applied to each *fastq* file i) individually by repeating this script for each sample and changing the files names, ii) or parallelized automatically using appropriate code on Linux for example. Here is an example to perform individual trimming on 4 samples:

####Trimming sample 1

fastq.info(fastq=1.fastq)

trim.seqs(fasta=1.fasta, qfile=1.qual, qwindowaverage=23, qwindowsize=25, minlength=280, maxlength=340, maxambig=0, maxhomop=8, keepforward=t, checkorient=t, oligos=forward_new.oligos, pdiffs=1, processors=1)

trim.seqs(fasta=1.trim.fasta, qfile=1.trim.qual, qwindowaverage=23, qwindowsize=25, minlength=280, maxlength=340, maxambig=0, maxhomop=8, keepforward=t, checkorient=t, oligos=reverse_new.oligos, pdiffs=1, processors=1)

list.seqs(fasta=1.trim.trim.fasta)

get.seqs(accnos=1.trim.trim.accnos, fasta=1.fasta, qfile=1.qual)

rename.file(input=1.pick.fasta, new=barcode1.fasta)

####Trimming sample 2

fastq.info(fastq=2.fastq)

trim.seqs(fasta=2.fasta, qfile=2.qual, qwindowaverage=23, qwindowsize=25, minlength=280, maxlength=340, maxambig=0, maxhomop=8, keepforward=t, checkorient=t, oligos=forward_new.oligos, pdiffs=1, processors=1)

trim.seqs(fasta=2.trim.fasta, qfile=2.trim.qual, qwindowaverage=23, qwindowsize=25, minlength=280, maxlength=340, maxambig=0, maxhomop=8, keepforward=t, checkorient=t, oligos=reverse_new.oligos, pdiffs=1, processors=1)

list.seqs(fasta=2.trim.trim.fasta)

get.seqs(accnos=2.trim.trim.accnos, fasta=2.fasta, qfile=2.qual)

rename.file(input=2.pick.fasta, new=barcode2.fasta)

####Trimming sample 3

fastq.info(fastq=3.fastq)

trim.seqs(fasta=3.fasta, qfile=3.qual, qwindowaverage=23, qwindowsize=25, minlength=280, maxlength=340, maxambig=0, maxhomop=8, keepforward=t, checkorient=t, oligos=forward_new.oligos, pdiffs=1, processors=1)

trim.seqs(fasta=3.trim.fasta, qfile=3.trim.qual, qwindowaverage=23, qwindowsize=25, minlength=280, maxlength=340, maxambig=0, maxhomop=8, keepforward=t, checkorient=t, oligos=reverse_new.oligos, pdiffs=1, processors=1)

list.seqs(fasta=3.trim.trim.fasta)

get.seqs(accnos=3.trim.trim.accnos, fasta=3.fasta, qfile=3.qual)

rename.file(input=3.pick.fasta, new=barcode3.fasta)

####Trimming sample 4

fastq.info(fastq=4.fastq)

trim.seqs(fasta=4.fasta, qfile=4.qual, qwindowaverage=23, qwindowsize=25, minlength=280, maxlength=340, maxambig=0, maxhomop=8, keepforward=t, checkorient=t, oligos=forward_new.oligos, pdiffs=1, processors=1)

trim.seqs(fasta=4.trim.fasta, qfile=4.trim.qual, qwindowaverage=23, qwindowsize=25, minlength=280, maxlength=340, maxambig=0, maxhomop=8, keepforward=t, checkorient=t, oligos=reverse_new.oligos, pdiffs=1, processors=1)

list.seqs(fasta=4.trim.trim.fasta)

get.seqs(accnos=4.trim.trim.accnos, fasta=4.fasta, qfile=4.qual)

rename.file(input=4.pick.fasta, new=barcode4.fasta)

3 Dereplication into Individual Sequence Unit (ISU)

Prior to the dereplication step and the subsequent bioinformatic steps, all the individual sample *fasta* files resulting from the trimming steps are merged together. Commands provided below are based on the 4 samples example dataset:

#Merge of the 4 samples *fasta* files which contained trimmed DNA reads

merge.files(input=barcode1.fasta-barcode2.fasta-barcode3.fasta-barcode4.fasta, output=barcodes.fasta)

#Creation of a group file which indicate the sample origin of each DNA read

make.group(fasta=barcode1.fasta-barcode2.fasta-barcode3.fasta-barcode4.fasta, groups=01-02-03-04)

rename.file(input=merge.groups, new=barcodes.groups, deleteold=true)

As the *fasta* file may contain DNA reads that are identical, we use the dereplication step to conserve only one DNA read sequence called Individual Sequence Unit (ISU). Mothur will create the *names* file in order to indicate how many DNA reads are identical to the ISU.

 # Dereplication step: creation of a *fasta* file with ISUs and the *names*

unique.seqs(fasta=barcodes.fasta)

summary.seqs(fasta=barcodes.unique.fasta, name=barcodes.names)

```
mothur > summary.seqs(fasta=barcodes.unique.fasta, name=barcodes.nar


Using 1 processors.

                Start   End     NBases  Ambigs  Polymer NumSeqs
Minimum:        1       280     280     0       3       1
2.5%-tile:      1       311     311     0       4       4158
25%-tile:       1       312     312     0       7       41574
Median:         1       312     312     0       7       83147
75%-tile:       1       312     312     0       7       124720
97.5%-tile:     1       312     312     0       7       162135
Maximum:        1       354     354     0       10      166292
Mean:   1       312     312     0       6
# of unique seqs:       76649
total # of seqs:        166292
```

After dereplication we obtained 76,649 ISUs corresponding to 166,292 DNA reads

As we organized our data into ISUs, which can be considered as OTUs at 100% similarity threshold, we can remove ISUs represented by only 1 DNA reads (similar to singleton removal). This step is performed to denoise our data by removing DNA reads which certainly correspond to erroneous sequences and artefacts produced during PCR amplification and the sequencing. If the aim of the study is to work on low abundant taxa, this step should be avoided in order to keep singleton. In the case of diatom metabarcoding and for biomonitoring purposes, previous study set this threshold to 10 reads; because only abundant taxa have a real impact during calculation of diatom water quality indices (Apothéloz-Perret-Gentil *et al.*, 2017) we can remove rare signals. In this example, we will remove ISUs represented by only 1 DNA reads.

#Removal of ISU with represented only 1 read

```
split.abund(fasta=barcodes.unique.fasta, name=barcodes.names, group=barcodes.groups, cutoff=1, accnos=T)
summary.seqs(fasta=barcodes.unique.abund.fasta, name=barcodes.abund.names)
count.groups(group= barcodes.abund.groups)
```

```
mothur > summary.seqs(fasta=barcodes.unique.abund.fasta, name=barcodes.abund
names)

Using 1 processors.

                Start   End     NBases  Ambigs  Polymer NumSeqs
Minimum:        1       302     302     0       3       1
2.5%-tile:      1       312     312     0       4       2627
25%-tile:       1       312     312     0       7       26261
Median:         1       312     312     0       7       52521
75%-tile:       1       312     312     0       7       78781
97.5%-tile:     1       312     312     0       7       102415
Maximum:        1       338     338     0       8       105040
Mean:   1       311     311     0       6
# of unique seqs:       15397
total # of seqs:        105040
```

After removal of ISUs represented by 1 read, we obtain 15,397 ISUs corresponding to 105,040 DNA reads

4 Alignment of DNA reads

Prior to the alignment step, primers are removed from the DNA reads nucleotide sequence and the expected length of the *rbc*L barcode without primers is 263 bp.

#Primers removal

```
pcr.seqs(fasta=barcodes.unique.abund.fasta, group=barcodes.abund.groups, name=barcodes.abund.names,
oligos=oligos_primer_removal.oligos, keepprimer=f, pdiffs=1, rdiffs=1, keepdots=f)
```

For downstream bioinformatics steps, Mothur will need an alignment of our DNA reads. As classical alignment software (Muscle, Clustalw, Maaft) would be time-consuming to align millions of DNA reads, Mothur propose a different approach : “The general approach is to i) find the closest template for each candidate using kmer searching; ii) to make a pairwise alignment between the candidate and de-gapped template sequences using the Needleman-Wunsch algorithms (Needleman & Wunsch, 1970) ; and iii) to re-insert gaps to the candidate and template pairwise alignments using the NAST algorithm so that the candidate sequence alignment is compatible with the original template alignment”

(https://www.mothur.org/wiki/Align.seqs). Thus, we use a reference alignment created with the *rbc*L diatom database as a template to align the DNA reads.

#Alignment of DNA reads

```
align.seqs(fasta=barcodes.unique.abund.pcr.fasta, reference=312bp_rbcL_(11_15)_980seq_align.fasta, processors=1)
```

The alignment produced is optimized automatically in order to find the best "start" and "end" positions shared by 90% of the DNA reads. Poorly aligned sequences will be removed.

#Optimization and curation of the alignment

```
screen.seqs(fasta=barcodes.unique.abund.pcr.align, name=barcodes.abund.pcr.names, group=barcodes.abund.pcr.groups, optimize=start-end, criteria=90, processors=1)
```

#Useless "_"and "." are removed from the alignment

```
filter.seqs(fasta=barcodes.unique.abund.pcr.good.align, trump=., vertical=T, processors=1)
```

#As we removed primers, a new dereplication step can be performed to detect new ISUs

```
unique.seqs(fasta=barcodes.unique.abund.pcr.good.filter.fasta, name=barcodes.abund.pcr.good.names)
```

5 Detection of chimeras

Prior to the detection of chimeras, Mothur recommend to pre-cluster the DNA reads in order to de-noise our data. This is performed using a pseudo-single linkage algorithm that will split the sequence sample by sample and sort them by abundance. The algorithm will then identify DNA reads that are within 1 nt similarity threshold and merge low abundant DNA read into the most abundant ones (https://mothur.org/wiki/Pre.cluster). As for singleton removal, this step can be avoided (or diffs set to 0) to enable low abundant taxa detection.

#Denoising using pre-cluster approach

```
pre.cluster(fasta=barcodes.unique.abund.pcr.good.filter.unique.fasta, name=barcodes.unique.abund.pcr.good.filter.names, group=barcodes.abund.pcr.good.groups, diffs=1, processors=1)
```

#Chimera detection software as implemented in vsearch (Rognes *et al.*, 2016)

```
chimera.vsearch(fasta=barcodes.unique.abund.pcr.good.filter.unique.precluster.fasta, name=barcodes.unique.abund.pcr.good.filter.unique.precluster.names, group=barcodes.abund.pcr.good.groups, dereplicate=T, processors=1)
```

#Chimera are removed from *fasta*, *names*, *groups* files according to chimera.vsearch results

```
remove.seqs(accnos=barcodes.unique.abund.pcr.good.filter.unique.precluster.denovo.vsearch.accnos, fasta=barcodes.unique.abund.pcr.good.filter.unique.precluster.fasta, name=barcodes.unique.abund.pcr.good.filter.unique.precluster.names, group=barcodes.abund.pcr.good.groups)
```

#Change the name of the *groups*, *fasta*, *names* files

```
rename.file(input=barcodes.abund.pcr.good.pick.groups, new=barcodes.final.groups, deleteold=true)
```

```
rename.file(input=barcodes.unique.abund.pcr.good.filter.unique.precluster.pick.fasta, new=barcodes.final.fasta, deleteold=true)
```

```
rename.file(input=barcodes.unique.abund.pcr.good.filter.unique.precluster.pick.names, new=barcodes.final.names, deleteold=true)
```

6 Selection of diatom DNA reads (*Bacillariophyta*)

Mothur implemented the Naïve Bayesian classifier developed by (Wang *et al.*, 2007) that allowed a rapid assignment of DNA reads. The Diat.barcode reference database (previously named R-Syst::diatom, Rimet *et al.*, 2016) is used as template for the assignment and is freely available (https://www6.inra.fr/carrtel-collection_eng/Barcoding-database/Aligned-and-trimed-database).

#DNA reads taxonomic assignment (wang method, bootstrap cutoff=75%)

```
classify.seqs(fasta=barcodes.final.fasta, group=barcodes.final.groups, name=barcodes.final.names,
template=Rsyst_230218_align_1401seqs_312bp_vf.fasta,
taxonomy=Rsyst_230218_align_1401seqs_312bp_vf.tax, cutoff=75, processors=1)
```

#Selection of DNA reads belonging to Bacillariophyta

```
get.lineage(taxonomy=barcodes.final.Rsyst_230218_align_1401seqs_312bp_vf.wang.taxonomy,
taxon='Bacillariophyta;Fragilariophyceae;-Bacillariophyta;Bacillariophyceae;-Bacillariophyta;Mediophyceae;-
Bacillariophyta;Coscinodiscophyceae;', group=barcodes.final.groups, name=barcodes.final.names,
fasta=barcodes.final.fasta)
```

7 Similarity Distance Matrix

Prior performing OTU clustering, we need to create a similarity distance matrix. To do so, "the dist.seqs command will calculate uncorrected pairwise distances between aligned DNA sequences". This approach presents several advantages: i) distances are not stored in RAM, ii) we can obtain a column-formatted matrix instead of lower triangle or square matrices to reduce the size of the file generated, iii) we can set a cutoff to only conserve distances we are interested in. As we will used a distance suimilarity threshold of 95% to create OTUs, we can set a cutoff of 0.06 (94% similarity) to generate the distance matrix. However, it considers that caution should be exercised when applying this command, it can generate large file (>100 Go) if the dataset contains a lot DNA reads or if the cutoff is not adapted.

#Creation of the distance matrix based on DNA reads similarity

```
dist.seqs(fasta=barcodes.final.pick.fasta, cutoff=0.06, processors=1)
```

8 OTU Clustering

Several clustering algorithms are proposed by Mothur to cluster DNA reads into OTUs, the default method proposed corresponding to the Opticlust algorithm which has been identified as the most adapted (Westcott & Schloss, 2017). Despite good results obtained with this approach for diatom DNA metabarcoding (Mortagua et al. in review), we will use the furthest neighbor algorithm as reported in previous studies (e.g. Vasselon *et al.*, 2017b; Rimet *et al.*, 2018; Rivera *et al.*, 2018; Keck *et al.*, 2018b). Even if this algorithm inflates the number of OTUs created, their consistency is well preserved allowing a better reliability of the taxonomic assignment. We use a 95% similarity threshold to define OTUSs with this clustering methods and the diatom *rbc*L barcode as we considered it as a good compromise between a correct taxonomic resolution and limitation of DNA metabarcoding biases (see Tapolczai *et al.*, 2019 for a better insight of "the impact of OTU sequence similarity threshold on diatom-based bioassessment").

#OTU Clustering with the furthest neighbor method. The cutoff set to 0.06 means that the command will automatically create OTUs lists at 100, 99, 98, 97, 96, 95, 94% similarity thresholds

```
cluster(column=barcodes.final.pick.dist, name=barcodes.final.pick.names, method=furthest, cutoff=0.06)
```

If the distance matrix is too large, your computer may not be able to use the *cluster* command to create OTUs. In this case, the *cluster.split* command can be used to proceed as it will split your distance matrix based on the DNA read taxonomy you obtained at the step 6 prior performing the clustering (https://www.mothur.org/wiki/Cluster.split).

#Use *cluster.split* command if *cluster* command failed with distance matrix

```
#cluster.split(column=barcodes.final.pick.dist, name=barcodes.final.pick.names,
taxonomy=barcodes.final.Rsyst_230218_align_1401seqs_312bp_vf.wang.pick.taxonomy, splitmethod=classify,
taxlevel=7, method=furthest, cutoff=0.06)
```

#Reorganize the OTU table and select only the 95% OTU table

```
make.shared(list=barcodes.final.pick.fn.list, group=barcodes.final.pick.groups, label=0.05)
```

```
make.table(shared=barcodes.final.pick.fn.shared)
```

9 Taxonomic assignment of OTUs

A taxonomy is given to each OTU based on the consensus taxonomy of DNA reads belonging to each OTU using the DNA read taxonomy obtained at the step 6. A cutoff is provided and correspond to the minimum confidence threshold expected before validating a taxonomy.

#Taxonomic assignment of OTUs

```
classify.otu(list=barcodes.final.pick.fn.list, name=barcodes.final.pick.names, group=barcodes.final.pick.groups,
taxonomy=barcodes.final.Rsyst_230218_align_1401seqs_312bp_vf.wang.pick.taxonomy, cutoff=80,
label=0.05)
```

#Selection of a DNA representative sequence for each OTU

```
get.oturep(list=barcodes.final.pick.fn.list, name=barcodes.final.pick.names, group=barcodes.final.pick.groups,
label=0.05, method=abundance, fasta=barcodes.final.pick.fasta)
```

10 Data normalization

Before performing community structure analyses, the data must be normalized. We perform a random subsampling of DNA reads in order to set all the samples at the same DNA read number corresponding to the lowest read number found within one sample. For diatom DNA metabarcoding, we expect to have at least 5000 DNA read per sample and if a sample is below this number, he will be removed from the analyses before performing the subsampling.

#Sub sampling and creation of the normalized OTU table

```
sub.sample(shared=barcodes.final.pick.fn.shared, name=barcodes.final.pick.names,
group=barcodes.final.pick.groups, fasta=barcodes.final.pick.fasta, label=0.05)
```

```
make.table(shared=barcodes.final.pick.fn.0.05.subsample.shared)

#Selection of the subsampled OTUs within the OTU taxonomy file

list.otulabels(shared=barcodes.final.pick.fn.0.05.subsample.shared)

get.otus(accnos=barcodes.final.pick.fn.0.05.subsample.0.05.otulabels,
constaxonomy=barcodes.final.pick.fn.0.05.cons.taxonomy)
```

Reference

Apothéloz-Perret-Gentil L., Cordonier A., Straub F., Iseli J., Esling P. & Pawlowski J. (2017). Taxonomy-free molecular diatom index for high-throughput eDNA biomonitoring. *Molecular Ecology Resources* **17**, 1231–1242. https://doi.org/10.1111/1755-0998.12668

Keck F., Vasselon V., Rimet F., Bouchez A. & Kahlert M. (2018a). Boosting DNA metabarcoding for biomonitoring with phylogenetic estimation of operational taxonomic units' ecological profiles. *Molecular Ecology Resources* **18**, 1299–1309. https://doi.org/10.1111/1755-0998.12919

Keck F., Vasselon V., Rimet F., Bouchez A. & Kahlert M. (2018b). Boosting DNA metabarcoding for biomonitoring with phylogenetic estimation of operational taxonomic units' ecological profiles. *Molecular Ecology Resources*. https://doi.org/10.1111/1755-0998.12919

Kozich J.J., Westcott S.L., Baxter N.T., Highlander S.K. & Schloss P.D. (2013). Development of a dual-index sequencing strategy and curation pipeline for analyzing amplicon sequence data on the MiSeq Illumina sequencing platform. *Applied and environmental microbiology* **79**, 5112–20. https://doi.org/10.1128/AEM.01043-13

Needleman S.B. & Wunsch C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* **48**, 443–453. https://doi.org/10.1016/0022-2836(70)90057-4

Rimet F., Chaumeil P., Keck F., Kermarrec L., Vasselon V., Kahlert M., *et al.* (2016). R-Syst::diatom: an open-access and curated barcode database for diatoms and freshwater monitoring. *Database* **2016**, baw016. https://doi.org/10.1093/database/baw016

Rimet F., Vasselon V., A.-Keszte B. & Bouchez A. (2018). Do we similarly assess diversity with microscopy and high-throughput sequencing? Case of microalgae in lakes. *Organisms Diversity and Evolution* **18**. https://doi.org/10.1007/s13127-018-0359-5

Rivera S.F., Vasselon V., Jacquet S., Bouchez A., Ariztegui D. & Rimet F. (2018). Metabarcoding of lake benthic diatoms: from structure assemblages to ecological assessment. *Hydrobiologia* **807**, 37–51. https://doi.org/10.1007/s10750-017-3381-2

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

Schloss P.D., Westcott S.L., Ryabin T., Hall J.R., Hartmann M., Hollister E.B., *et al.* (2009). Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology* **75**, 7537–7541. https://doi.org/10.1128/AEM.01541-09

Tapolczai K., Vasselon V., Bouchez A., Stenger-Kovács C., Padisák J. & Rimet F. (2019). The impact of OTU sequence similarity threshold on diatom-based bioassessment: A case study of the rivers of Mayotte (France, Indian Ocean). *Ecology and Evolution* **9**, 166–179. https://doi.org/10.1002/ece3.4701

Vasselon V., Domaizon I., Rimet F., Kahlert M. & Bouchez A. (2017a). Application of high-throughput sequencing (HTS) metabarcoding to diatom biomonitoring: Do DNA extraction methods matter? *Freshwater Science* **36**, 162–177. https://doi.org/10.1086/690649

Vasselon V., Rimet F., Tapolczai K. & Bouchez A. (2017b). Assessing ecological status with diatoms DNA metabarcoding: Scaling-up on a WFD monitoring network (Mayotte island, France).

Ecological Indicators **82**, 1–12. https://doi.org/10.1016/j.ecolind.2017.06.024

Wang Q., Garrity G.M., Tiedje J.M. & Cole J.R. (2007). Naïve Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology* **73**, 5261–5267. https://doi.org/10.1128/AEM.00062-07

Westcott S.L. & Schloss P.D. (2017). OptiClust, an Improved Method for Assigning Amplicon-Based Sequence Data to Operational Taxonomic Units. *mSphere* **2**, e00073-17. https://doi.org/10.1128/mSphereDirect.00073-17