Version: v1.3
Date: 27/05/2024

Workflow for the analysis of e-DNA metabarcoding data:

1. Step 0 – Quality control of the raw sequencing data 2
2. Step 1 – Prepare data for demultiplexing 3
   2.1. Step1a – Check number of reads in the paired R1 and R2 file 3
   2.2. Step1b – Create Barcode files in fasta format used for demultiplexing 4
3. Step2 – Demultiplexing and concatenation of the PCR replicates per sample 5
   3.1. Step2a – Demultiplexing 5
   3.2. Step2b – Concatenation of the three PCR-replicates 7
4. Step 3 – DADA2 to produce amplicon sequence variants (ASVs) and for the taxonomic assignment 8
5. Step 4 – nucleotide BLAST (BLASTn) for taxonomic assignment of ASVs that remained unassigned in DADA2 9
6. Step 5 – Automatically merge taxonomic assignment by DADA2 and BLASTn 11
7. Step 6 – Decontamination 12
8. Step 7 – Data Analysis 13
   8.1. Step 7a – Check the origin of the ASVs and reads 13
   8.2. Step 7b – Heatmap 13
   8.3. Step 7c – Alpha-diversity 13
   8.4. Step 7d – Permanova 14
   8.5. Step 7e – Analysis of the species community structures 14

1. Step 0 – Quality control of the raw sequencing data

Use script: Step0_fastqc.sh

What does the script do?

This script is used to check the quality of the raw sequencing data. The RawData is downloaded from GenHub and will be saved in a folder at:
/home/genomics/ALL_RAW_SEQ_DATA/03_eDNA/Taxonomic_profiling/

How to run the script in bash?

- To run the script in bash multiqc must be activated through the bash command:
  ```
  conda activate multiqc
  ```
- Three parameters are needed to run the script:
  - Path of the folder with fastq.gz files
    eg. `/home/genomics/icornelis/01_Raw_sequencing_Data/12S-MiFish_UE/RawData_18094-15`
  - Path of the output folder
    eg. `/home/genomics/icornelis/01_Raw_sequencing_Data/12S-MiFish_UE/RawData_18094-15/FastQC`
  - Number of threads to use
    eg. `4`

  The command should look like this example:
  ```
  /home/genomics/icornelis/03_RawScripts/Step0_fastqc.sh
  /home/genomics/icornelis/01_Raw_sequencing_Data/12S-
  MiFish_UE/RawData_18094-15
  /home/genomics/icornelis/01_Raw_sequencing_Data/12S-
  MiFish_UE/RawData_18094-15/FastQC 4
  ```
- Run `multiqc .`
- After running the script multiqc can be deactivated:
  ```
  conda deactivate
  ```

2. Step 1 – Prepare data for demultiplexing

2.1. Step1a – Check number of reads in the paired R1 and R2 file

Use command: zgrep -Ec “@M0”

What does the command do?

In order to continue with the Demultiplexing step, the raw reads in the forward file (R1) and reverse file (R2) must be paired. If both files are paired, the number of raw reads in both files will be equal. In order to check this the command zgrep in bash will be used.

How to run the command in bash?

Type the command zgrep -Ec “@M0” in Bash and then add the path to your folder with the raw reads in fastq.gz format and to the folder that should contain the results from the count

The command should look like this example (haakjes rond @M0 apart ingeven in Bash script):

```
zgrep -Ec ‘@M0’ /home/genomics/icornelis/Raw_sequencing_Data/12S-
MiFish_UE/RawData_18094-15
1>/home/genomics/icornelis/Raw_sequencing_Data/12S-MiFish_UE/ RawData_18094-
15/output_count_raw_reads.txt
```

How to continue after running the script?

If the **number of raw reads** in the paired R1 and R2 files are **equal**, continue to **Step1b**
If the **number of raw reads** in the paired R1 and R2 file **differ**, use the script in the folder **Step1a_Pairfq-master/scripts** (more information: https://github.com/sestaton/Pairfq/wiki)

Use script: pairfq_lite.pl

- To run the script you need to define what you want to do:
    - `addinfo` - Add the pair info back to the FASTA/Q header
    - `makepairs` - Pair the forward and reverse reads and write singletons in a separate file
    - `joinpairs` - Interleave the paired forward and reverse files
    - `splitpairs` - Split the interleaved file into separate files for the forward and reverse reads
- To pair the data use the **makepairs** function and add the following parameters to run makepairs:
    - -f : File with the forward reads in fastq.gz format
      eg. `/home/genomics/icornelis/01_Raw_sequencing_Data/12S-MiFish_UE/RawData_18094-15/18094D-15-01_S1_L001_R1_001.fastq.gz`
    - -r : File with the reverse reads in fastq.gz format
      eg. `/home/genomics/icornelis/01_Raw_sequencing_Data/12S-MiFish_UE/RawData_18094-15/18094D-15-01_S1_L001_R2_001.fastq.gz`
    - -fp : Name of the fastq.gz file that will contain the paired forward reads
      eg. `/home/genomics/icornelis/01_Raw_sequencing_Data/12S-MiFish_UE/RawData_18094-15/S1_R1_paired.fastq.gz`
    - -rp : Name of the fastq.gz file that will contain the paired reverse reads
      eg. `/home/genomics/icornelis/01_Raw_sequencing_Data/12S-MiFish_UE/RawData_18094-15/S1_R2_paired.fastq.gz`
    - -fs : Name of the fastq.gz file that will contain the singleton forward reads
      eg. `/home/genomics/icornelis/01_Raw_sequencing_Data/12S-MiFish_UE/RawData_18094-15/S1_R1_singleton.fastq.gz`

  - -rs : Name of the fastq.gz file that will contain the singleton reverse reads eg. `/home/genomics/icornelis/01_Raw_sequencing_Data/12S-MiFish_UE/RawData_18094-15/S1_R2_singleton.fastq.gz`
- The final command should look like this example:

```
/home/genomics/icornelis/03_Raw_scripts/Step1a_Pairfq-
master/scripts/pairfq_lite.pl -f
/home/genomics/icornelis/01_Raw_sequencing_Data/12S-
MiFish_UE/RawData_18094-15/18094D-15-01_S1_L001_R1_001.fastq.gz -r
/home/genomics/icornelis/01_Raw_sequencing_Data/12S-
MiFish_UE/RawData_18094-15/18094D-15-01_S1_L001_R2_001.fastq.gz -fp
/home/genomics/icornelis/01_Raw_sequencing_Data/12S-
MiFish_UE/RawData_18094-15/S1_R1_paired.fastq.gz -rp
/home/genomics/icornelis/01_Raw_sequencing_Data/12S-
MiFish_UE/RawData_18094-15/S1_R2_paired.fastq.gz -fs
/home/genomics/icornelis/01_Raw_sequencing_Data/12S-
MiFish_UE/RawData_18094-15/S1_R1_singleton.fastq.gz -rs
/home/genomics/icornelis/01_Raw_sequencing_Data/12S-
MiFish_UE/RawData_18094-15/S1_R2_singleton.fastq.gz
```

2.2. Step1b – Create Barcode files in fasta format used for demultiplexing

In R run script: Step1b_Script_Barcodes_text-fasta.R

What does the script do?

For demultiplexing, the unique tags (Barcodes) used for each of the samples must be available in fasta format. Furthermore, the barcodes for the F- and R- primers must be available in a separate document.

How to run the script in RStudio?

- **Save As…** the script to your folder of interest
- Before using the script adjust the following parameters for your dataset in:
  - in RStudio – set working directory to your folder of interest
  - **line 10:** adjust proj.path to your project folder which contains the files needed for demultiplexing
  - **line 12:** file = give the name of your Excel file and sheet with the information about the sample tags
  - **lines 13 – 16:** file = give a name to the fasta-file containing the tags for each sample
- The input file containing the information about the barcodes used for each sample must be in xlsx-format and must contain the **Final** Name of the sample (**Demultiplex name**), the barcode for the forward primer (**F-primer tag**) and the barcode for the reverse primer (**R-primer tag**).

3. Step2 – Demultiplexing and concatenation of the PCR replicates per sample

3.1. Step2a – Demultiplexing

Use script: Step2a_Script_demultiplex_Liu.sh

What does the script do?

The script will demultiplex the raw reads from the paired R1 and R2 files based on the primer tags used for each sample. For each sample a separate file will be made, that will only contain the raw reads from that sample. After demultiplexing the raw reads will be trimmed by removing the primer- and tag-sequences from the 5'-end and 3'-end of the read. In addition, trimmed sequences that are shorter than a set length will be removed from the file.

How to run the script in bash?

- **Save As…** the script to your folder of interest
- Before using the script adjust the following parameters for your dataset in:
  - **line 22:** DIR, set path directory to your folder of interest
  - **line 24:** TRUCVAL, minimum length of the sequences kept in the dataset
  - **line 75:** add the fasta file with the forward reads and reverse primer tags
    - **-g** file:"$DIR"/ "fasta file with the **F-tags** used for each sample"
    - **-G** file:"$DIR"/ "fasta file with the **R-tags** used for each sample"
  - **line 78:** add the fasta file with the forward reads and reverse primer tags
    - **-g** file:"$DIR"/ "fasta file with the **R-tags** used for each sample"
    - **-G** file:"$DIR"/ "fasta file with the **F-tags** used for each sample"
  - **line 104:** add primersequence in # trim **3' SENSE** with cutadapt
    - **-a** "ReverseComplement of the **R-primer**"
    - **-A** "ReverseComplement of the **F-primer**"
  - **line 111:** add primersequences in # trim **3' ANTISENSE** with cutadapt
    - **-a** "ReverseComplement of the **F-primer**"
    - **-A** "ReverseComplement of the **R-primer**"
- Add the following parameters to run the demultiplex script:
  - -p: the name of the primerset
    eg. `MiFish_UE`
  - -l: the name of the library or PCR-pool
    eg. `S1`
    → -p and -l are also used in the final folder name
    → -l is also used in the final file name for each sample
  - -a: the input file with the paired R1 sequences in fastq format
    eg.`/home/genomics/icornelis/ZEROimpact/01_12S/NJ2021/MiFish-UE_run2/18094D-12-01_S1_L001_R1_001.fastq`
  - -b: the input file with the paired R2 sequences in fastq format
    eg.`/home/genomics/icornelis/ZEROimpact/01_12S/NJ2021/MiFish-UE_run2/18094D-12-01_S1_L001_R2_001.fastq`
  - -f: the sequence of the forward primer without tag
    eg. `GTYGGTAAAWCTCGTGCCAGC`
  - -r: the sequence of the reverse primer without tag
    eg. `CATAGTGGGGTATCTAATCCYAGTTTG`
  - -t: the number of threads
    eg. `8`
  - -m: the length of the shortest primer without tag

eg. `21` in case of the MiFish_UE primers

- The final command should look like this example:

```
/home/genomics/icornelis/ZEROimpact/01_12S/NJ2021/MiFish-UE_run2/
Step2_Script_demultiplex_Liu.sh -p MiFish_UE -l S1 -a
/home/genomics/icornelis/ZEROimpact/01_12S/NJ2021/MiFish-UE_run2/18094D-
12-01_S1_L001_R1_001.fastq -b
/home/genomics/icornelis/ZEROimpact/01_12S/NJ2021/MiFish-UE_run2/18094D-
12-01_S1_L001_R2_001.fastq -f GTYGGTAAAWCTCGTGCCAGC -r
CATAGTGGGGTATCTAATCCYAGTTTG -t 8 -m 21
```

- After running the script the demultiplexed and trimmed sequences will be sorted into 4 folder: 2 folders in the sense folder, trimmed-R1 and trimmed-R2, and 2 folders in the antisense folder, trimmed-R1 and trimmed-R2. This is important for further processing in DADA2.

How does the script work?

Part 1: The raw sequences in the paired R1 and R2 file are **orientated** into sense and antisense sequences, and are **sorted** into the respective **sense or antisense** folder. First the script will look for the forward primer in the sequences from the R1-file (sense) and the reverse primer in the sequences from the R2-file (sense). Sequences were no forward or reverse primer was found are stored into the untrimmed file, which will be used during the 2nd round. During the 2nd round the script searches for the reverse primer in the sequences from the R1-file (antisense) and the forward primer in sequences from the R2-file (antisense). Sequences where no primers were found will be moved to the trash folder (filename: noprimer.R1.fastq.gz)

1st round -g is used to look for the forward primer in the R1 file and -G is used to look for the reverse primer in the paired R2 file → these are all the sequences orientated in sense

2nd round -g is used look for the reverse primer in the R1 file and -G is used to look for the forward primer in the paired R2 file → these are all the sequences orientated in antisense

To increase the number of sequences that are kept for further processing, the number of mismatches allowed was set to -e 0.15 → max 3 mismatches in the forward primer (21 bp) and max 4 mismatches in the reverse primer (27 bp) in case of the MiFish_UE primers

Part 2: Is a **visualization step**, to check for the presence of the forward or reverse primer in the first 20 sequences in the R1 and R2 files from the sense folder. The primer sequence is colored for visualization.

Part 3: Is the **demultiplexing step**, the sequences will only be sorted if the tag is found in both paired sequences through the command --pair adapter, which means that for one sequence the tag is found in both the R1 and R2 file.

-g is used by cutadapt to look for the forward tag in the R1 file (sense-folder) and for the reverse tag in the R1 file (antisense-folder)

-G is used by cutadapt to look for the reverse tag in the R2 file (sense-folder) and for the forward tag in the R2 file (antisense-folder)

Thus cutadapt will only keep sequences in the sense folder when the forward tag is found in the R1 file and the reverse tag is found in the reverse complement of that sequence in the paired R2 file, and in the antisense folder when the reverse tag is found in the R1 file and the forward tag is found in the reverse complement of that sequence in the paired R2 file

Part 4: Is the **trimming step** to remove the primer sequences and their tag. This is first done at the 3'-end of the sequences using the reverse complement of the two primer sequences and then at the 5'-end of the sequences. In addition, all sequences with less nucleotides than the set TRUCVAL-value will be removed.

3.2. Step2b – Concatenation of the three PCR-replicates

Use script: Step2b_Script_Concatenate_PCRseparate.sh or
Step2b_Script_Concatenate_PCRconcatenated.sh

What does the script do?

After demultiplexing the sequences of the 3 PCR replicates will be concatenated into one folder. This folder contains a processed-reads folder with the same structure as the folders for the separate PCR replicates. You can either choose to keep the PCR replicates for each sample separated (*_PCRseparate.sh) or to concatenated the 3PCR replicates into one file for each sample (*_PCRconcatenated.sh).

How to run the script in bash?

- When activating the script Step2b_Script_Concatenate_PCRseparate.sh, bash will ask you to enter your:

  Input folder: the path to the folder that contains the folders with the sequences

  RUNNAME: eg. MiFish_UE-S, the runname will be used to name the final folder containing the concatenated files of all 3 PCR runs
- The folder MiFish_UE-S_concatenated/processed-reads/ has 4 subfolders (sense/trimmed-R1, sense/trimmed-R2, antisense/trimmed-R1 and antisense/trimmed-R2). In each folder the demultiplexed raw sequences of the three PCR replicates are kept in separate files for each sample as *_S1.R1.fatsq.gz, *_S2.R1.fatsq.gz, and *_S3.R1.fatsq.gz
- To check the number of reads per sample use command:
  ```
  zgrep -Ec "@M0" MiFish_UE-S_concatenated/processed-reads/*/trimmed-R*/*.fastq.gz 1>output_count_concatenated_reads.txt
  ```

- When activating the script Step2b_Script_Concatenate_PCRconcatenated.sh, bash will ask you to enter your:

  Input folder: the path to the folder that contains the folders with the sequences

  RUNNAME: eg. MiFish_UE-S, the runname will be used to name the final folder containing the concatenated files of all 3 PCR runs
- The folder MiFish_UE-S_concatenated/processed-reads/ has 4 subfolders (sense/trimmed-R1, sense/trimmed-R2, antisense/trimmed-R1 and antisense/trimmed-R2). In each folder the demultiplexed raw sequences of the three PCR replicates of each sample are kept in one file per sample
- To check the number of reads per sample use command:
  ```
  zgrep -Ec "@M0" MiFish_UE-S_concatenated/processed-reads/*/trimmed-R*/*.fastq.gz 1>output_count_FullConcatenated_reads.txt
  ```

4. Step 3 – DADA2 to produce amplicon sequence variants (ASVs) and for the taxonomic assignment

In R run script: Step3_Script_dada2_PCRseparate.R or
Step3_Script_dada2_PCRconcatenated.R

What does the script do?

Through seven different steps, the DADA2 pipeline will use the demultiplexed fastq files to produce amplicon sequence variants (ASVs). The output is a matrix containing the samples (rows) and ASVs (columns), in which the value of each entry is the number of times that ASV was observed in that sample (number of reads). More information about the different steps can be found in the Introduction to dada2 (bioconductor.org).

After receiving the matrix, the samples will be concatenated and these concatenated samples will be rarefied in order to remove differences in the sequencing depth. After rarefaction, the script will use the DADA2 function assignTaxonomy for the taxonomic assignment of the ASVs. It uses the naive Bayesian classifier method and makes taxonomic assignments by comparing the ASV sequences to a custom reference database. The function assignTaxonomy will only return ASVs at species level when more than 80 of 100 bootstraps result in the same species level. This is also true for the higher taxonomic levels.

How to run the script in RStudio?

- **Save As…** the script to your folder of interest
- Before using the script adjust parameters for your dataset in:
  - Script funs-libs.R

    Cpath → if concatenated folder has the structure /processed-reads/sense and /processed-reads/antisense → adjust cpath in funs-libs.R to

    ```
    cpath <- function(sense,step,r){
        path <- paste0(proj.path,"/processed-reads/",sense,"/",step,"-",r)
        return(path)
        }
    ```
  - Step3_Script_dada2_PCRseparate.R or Step3_Script_dada2_PCRconcatenated.R
    - **line 32:** adjust proj.path to your project folder which contains the demultiplexed files with the reads and in the project folder create a folder results where all the data generated in DADA2 will be stored
    - **line 43:** set trucVal for the gene of interest
    - **line 152 + 159:** change MiFish_UE with the name of your primerset
    - The **taxonomic assignment** will be done with the **unrarefied data**, in order to keep all the information
    - After running the script you will have one important table used for further analysis:
      - table_unrarefied → contains the raw data and the taxonomic assignment of all ASVs

5. Step 4 – nucleotide BLAST (BLASTn) for taxonomic assignment of ASVs that remained unassigned in DADA2

Use command: blastn

What does the command do?

The nucleotide BLAST is used for the taxonomic assignment of the ASVs that remained unassigned after running DADA2. The cause may be the absence of the species from the custom reference database or a low taxonomic resolution between closely related species. BLAST uses a heuristic method, which first starts to search for short matches, seeds, between two sequences. During the second step, local alignments are made between the matches found during the first search. However, this means that optimal alignments are not guaranteed and that some hits may be missed. Although the results from the nucleotide BLAST will only be used for the taxonomic assignment of the unassigned ASVs, BLASTn will be run on the asvs.fa file that contains all the ASVs. This results in a extra check for the results from assignTaxonomy in DADA2.

How to run the command in bash?

To run the command in bash use the following commands:

- Command 1 for nucleotide BLAST against **GenBank** database:
```
/usr/local/bin/blastn -db /home/genomics/bioinf_databases/genbank/nt/nt -
num_threads 8 -query /home/genomics/icornelis/Parkwind/12S/MiFish_UE-
S_concatenated/results/asvs.fa -out
/home/genomics/icornelis/Parkwind/12S/blastn_GenBank_20220928.txt -
max_hsps 1 -max_target_seqs 1 -qcov_hsp_perc 75 -perc_identity 90 -outfmt
"6 qseqid sseqid staxids stitle pident qlen length mismatch gapopen evalue
bitscore qcovs"
```
- Command 2 for nucleotide BLAST against **own reference** database top10 hits**:**
```
blastn -db
/home/genomics/icornelis/03_RawScripts/Step4_12S_ReferenceDB_TaxonomicAss
ignment/12S_references_20220906.fa -query
/home/genomics/icornelis/Parkwind/12S/MiFish_UE-
S_concatenated/results/asvs.fa -max_hsps 1 -max_target_seqs 10 -
qcov_hsp_perc 75 -perc_identity 75 -outfmt "6 qseqid sseqid staxids stitle
pident qlen length mismatch gapopen evalue bitscore qcovs" -out
/home/genomics/icornelis/Parkwind/12S/blastn_ref_20221005.txt
```
- Command 2 for nucleotide BLAST against **own reference** database with tophits only**:**
```
blastn -db
/home/genomics/icornelis/03_RawScripts/Step4_12S_ReferenceDB_TaxonomicAss
ignment/12S_references_20220906.fa -query
/home/genomics/icornelis/Parkwind/12S/MiFish_UE-
S_concatenated/results/asvs.fa -max_hsps 1 -max_target_seqs 1 -
qcov_hsp_perc 75 -perc_identity 90 -outfmt "6 qseqid sseqid staxids stitle
pident qlen length mismatch gapopen evalue bitscore qcovs" -out
/home/genomics/icornelis/Parkwind/12S/blastn_ref_tophits_20221005.txt
```
- To run the command for your data adjust:
  - -db (only in Command 2 and 3): add your custom reference database
  - -query: add your file in fasta format that contains the ASVs
  - -out: adjust to your output file of interest
- **Important:** before blasting against the custom reference database convert the fasta file containing the reference sequences into a NCBI Blast Database through the command
```
makeblastdb -in /home/genomics/icornelis/
/03_RawScripts/Step4_12S_ReferenceDB_TaxonomicAssignment/12S_references_2
0220906.fas  -input_type fasta -dbtype nucl -out /home/genomics/icornelis/
```

```
/03_RawScripts/Step4_12S_ReferenceDB_TaxonomicAssignment/12S_references_2
0220906.fa
```

After running the command three files will be made: 12S_references_20220906.fa.nhr, 12S_references_20220906.fa.nin, 12S_references_20220906.fa.nsq

- After running blastn the results from both GenBank and our own reference list can be opened in Excel, using the tab as delimiter. For further processing, add column names to the first row in Excel.
  - Blastn_GenBank:
    ASV, sseqid, staxids, Species, pident, qlen, length, mismatch, gapopen, evalue, bitscore, qcovs
  - Blastn_own_references:
    Split the data in the 4th column: Select Data -> Text to Columns -> select Delimited -> Next, select the delimiter for your data -> Next -> Select the destination in your worksheet
    ASV, sseqid, staxids, Kingdom, Phylum, Class, Order, Family, Genus, Species, pident, qlen, length, mismatch, gapopen, evalue, bitscore, qcovs
  - For both files: Save As Excel Workbook (*.xlsx)

6. Step 5 – Automatically merge taxonomic assignment by DADA2 and BLASTn

In R run script: Step5_Script_SequenceTable_FullTaxonomicAssignment.R

What does the script do?

The script will be used to combine the taxonomic information gathered from DADA2 and BLASTn against the custom reference database and GenBank (tophits only). The three Excel files are uploaded into R and the column Species from each table is merged into a new data frame based on the column ASV.

During the first step the results from the taxonomic assignment in DADA2 are then copied into a new column Full. During the second step, we add the results from the nucleotide BLAST against the custom reference database to all the ASVs that remained unassigned by the function assignTaxonomy in DADA2. During the final step, the results from BLASTn against GenBank will be used to add a taxonomic assignment to the ASVs that still remain unassigned after the second step. Before getting the full taxonomic assignment all scientific names are check with WoRMS, and unaccepted scientific names are replaced by accepted scientific names. After getting the full taxonomic assignment the Classification received by the taxonomic assignment in DADA2 is replaced by the Classification from WoRMS and NCBI (for species that are not found in WoRMS).

For some ASVs taxonomic assignment to species level is not possible (due to it being absent from the reference database or due to identical barcode sequences). For these ASVs, the script uses a higher classification level (until Familiy-level) and adds sp. to it. If the Family is unknown the taxonomic assignment remains NA.

This script results in 2 datasets will be used for further analysis:

- table_unrarefied_raw_Full_TaxAss_WoRMS.xlsx
  → The raw dataset with full taxonomic assignment that will be used in the next step to remove the contaminant ASVs with Decontam.
- Fish_classes.rds
  ➔ The rds file that is used to select the ASVs assigned to fish species

How to run the script in RStudio?

- **Save As…** the script to your folder of interest
- Before using the script adjust parameters for your dataset in:
  - **line 2:** adjust proj.path to your project folder which contains the results from DADA2 and BLASTn against GenBank and the custom reference database
  - **line 29:** For 12S data, the 16S sequence of Skeletonema pseudocostatum (Chromista) was added to improve the resolution of the taxonomic assignment in DADA2. In the R-script, this information can be either removed or adjusted to your species of interest
  - **line 35:** These are species that share identical 12S sequences, which makes it impossible to detect them correctly based on eDNA metabarcoding with MiFish_UE primers against the 12S gene. You can detect these species by checking the results in the Excel-file that contains the results from the nucleotide BLAST, against the custom reference list, that contains the top10 hits (two species with a the same pident for the same ASV). This step can either be omitted or adjusted to species that cause problems in your dataset.

7. Step 6 – Decontamination

In R run script: Step6_Script_Decontam.R or Step6_Script_microDecon.R

After combining all taxonomic assignment results, the raw data with the full taxonomic assignment will be cleaned using the prevalence method in Decontam or microDecon.
After identifying the contaminants the PCR replicates are concatenated through summation. After concatenation, the data can be rarefied if necessary. Raremax should be low enough (min value = 10.000) to avoid that too many samples will get lost, but high enough to represent the biodiversity. The total number of reads in the negative controls is not considered. Create the rarecurve, using the chosen raremax as sample. In the rarecurve, raremax should be present at the plateau, after the steep part of the curve. Rarefy all samples with raremax. After rarefaction all the samples have the same total number of reads (= raremax). Samples with a total number of reads < raremax, will be removed from the data.
After running the script for Decontam you will get five files

- table_raw_clean
  → the raw but cleaned data-set with all samples
- table_unrarefied_concatenated_clean
  → the unrarefied cleaned data-set that still contains all the samples
- table_rarefied_cleaned_allSamples
  → the rarefied and cleaned data-set that still contains all the samples, the ASVs with high reads in negatives controls and without reads after rarefaction are removed
- table_rarefied_clean
  → the rarefied and cleaned data-set of which the samples with less reads than raremax are removed
- env_unrarefied.rds
  → contains environmental information about the locations at which the samples were collected
- The final **table_unrarefied_concatenated_clean** or **table_rarefied_clean** will be used for **further analysis**.

After running the script for microDecon you will get four files

- env_AllSample.rds
  → contains metadata about the samples (replicate, environmental zone, …)
- env_order.rds
  → contains metadata about the samples (replicate, environmental zone, …) but ordered according to the zone
- table_raw_clean
  → the raw but cleaned data-set with all samples
- The final **table_unrarefied_concatenated_clean** will be used for **further analysis**.

How to run the script in RStudio?

- **Save As…** the script to your folder of interest
- Before using the script adjust parameters for your dataset in:
  - **line 18:** adjust proj.path to your project folder which contains the results from

8. Step 7 – Data Analysis

8.1. Step 7a – Check the origin of the ASVs and reads

In R run script: Step7a_StackedBarplot_NegativeControlSamples_ColorZone.R

What does the script do?

The script will provide a stack barplot with the ASV-count and read-count in each sample (y-axis). The colors of the bars represent the species origin of the ASVs/reads. The labels will be colored according to the zone in which the sample was collected (x-axis). Input file is the ASV-table with full taxonomic assignment before concatenation. The technical replicates (PCR-replicates) should already be concatenated by summation.

How to run the script in RStudio?

- **Save As…** the script to your folder of interest
- Before using the script adjust parameters for your dataset in:
  - **line 9:** adjust proj.path to your project folder which contains the results from

8.2. Step 7b – Heatmap

In R run script: Step7b_Script_Heatmap_microDecon.R

What does the script do?

The script will use the un-rarefied or rarefied data after taxonomic assignment to create a:

- a heatmap with the ASVs assigned to species level and all samples. The species are sorted with a dendrogram to determine which species occur together.
  - Clustering is based on the ward.D method using the double transformed data
- a heatmap with the ASVs assigned to humans in all samples (optional)

How to run the script in RStudio?

- **Save As…** the script to your folder of interest
- Before using the script adjust parameters for your dataset in:
  - **line 43:** adjust proj.path to your project folder which contains the results from

8.3. Step 7c – Alpha-diversity

In R run script: Step7c_Script_AlphaDiversity_COIvs12S_CoverageBasedRarefaction.R

What does the script do?

The script will use the unrarefied data after decontamination to create:

- a boxplot with the observed diversity in each environmental zone
- a lineplot with the observed diversity in each environmental zone
- Statistical analysis to determine the difference in species richness between the environmental zones

How to run the script in RStudio?

- **Save As…** the script to your folder of interest
- Before using the script adjust parameters for your dataset in:
  - **line 47-48:** adjust proj.path to your project folder which contains the results from

8.4. Step 7d – Permanova

In R run script: Step7d_DNA_Permanova.Rmd

What does the script do?

The script will run a statistical analysis on the unrarefied data after decontamination.

How to run the script in RStudio?

- **Save As…** the script to your folder of interest
- Before using the script adjust parameters for your dataset in:
  - **line 29:** adjust proj.path to your project folder which contains the results from
  - **2nd Block until end:** adjust where needed to sort the data according to your environmental data

8.5. Step 7e – Analysis of the species community structures

In R run script: Step7e_Script_nMDS_COIvs12S_eDNAvsMorphology_microDecon_unrarefied.R

What does the script do?

The script will use the unrarefied after decontamination to create a nMDS-plot using the double transformed data:

- nMDS plot with the ASVs assigned to species level focusing on the three zones
- nMDS plot with the ASVs assigned to species level focusing on the samples taken outside and inside the offshore windfarms

All nMDS plots are created using Bray-Curtis dissimilarities

How to run the script in RStudio?

- **Save As…** the script to your folder of interest
- Before using the script adjust parameters for your dataset in:
  - **line 19-20:** adjust proj.path to your project folder which contains the results from