STANDARD OPERATING PROTOCOL · EnGeL-BIOINF-OBITOOLS4

Metabarcoding bioinformatics with OBITools4, VSEARCH, LULU and BOLDigger3 (COI amplicons; 18S nematode variant)

Protocol ID	EnGeL-BIOINF-OBITOOLS4
Version	1.2.0
Date	2026-11-20
Status	draft
Area	Bioinformatics (BIOINF)
Authors	Luísa Fraga (lead; pipeline script; ORCID 0000-0003-3153-4593); Andreia Quaresma (ORCID 0000-0002-8678-5800); Luís Cunha (pipeline script; review and approval; ORCID 0000-0002-5870-2537)
Reviewed and approved by	Luís Cunha
Laboratory	Environmental Genomics Laboratory (EnGeL-UC), Environmental Ecology & Ecotoxicology Group (E3G), Centre for Functional Ecology, Department of Life Sciences, University of Coimbra · Laboratório Associado TERRA
Contact	luis.cunha@uc.pt · ewormuc@gmail.com
Licence	Text: CC BY 4.0 · Code: MIT
Main hazards	None: computational protocol

1 Purpose and scope

This protocol turns demultiplexed paired-end Illumina reads from COI metabarcoding (BF3/BR2; EnGeL-MOLB-MB) into a curated OTU table with taxonomic assignments. Steps: read merging, primer removal and sample assignment (OBITools4), dereplication and singleton removal, denoising, chimera removal and clustering (VSEARCH), curation of erroneous OTUs and NUMTs (LULU), and identification against BOLD (BOLDigger3). Length filters and identity thresholds are set for the COI fragment. A variant for 18S nematode amplicons (NF1/18Sr2b), identified against 18S-NemaBase instead of BOLD, is given in step 10. The pipeline writes a run report with the number of reads and sequences kept after each step.

2 Principle

Forward and reverse reads are aligned into full-length sequences; reads that do not overlap well are discarded. Primers are removed and each sequence is annotated with its sample. Identical sequences are collapsed (dereplicated) and sequences seen only once (singletons), most likely PCR or sequencing errors, are removed. Across all samples, sequences of the expected length are denoised (UNOISE), chimeras are removed (UCHIME3 de novo)

and the remaining sequences are clustered into OTUs. LULU merges OTUs that are probably errors or nuclear copies of mitochondrial DNA (NUMTs) of more abundant OTUs, and BOLDigger3 assigns taxonomy against the BOLD database.

3 Before you start

Item	Details
Input	Demultiplexed paired fastq files (R1, R2) per sample, as delivered by the sequencing provider (Illumina adapters already removed)
Sample sheet	Tab-delimited text file: experiment, sample, tags (F:R; type ":" if there are no tags), primer F, primer R, F/T (both primers or forward only), "@", other information, and the plate-well index combination or fastq file code
Computing	Workstation or personal computer; hands-on and run time [TO CONFIRM]
Prepare in advance	OBITools4, VSEARCH, R packages and BOLDigger3 installed; BOLD database downloaded

4 Materials

4.1 Computer

- Lab workstation (see the lab's workstation guidance; conda environment "boldigger3") or a personal computer with a Unix shell
- Local copy of the BOLD database for BOLDigger3 (e.g. ~/Bold_db)
- For 18S nematode data: local copy of the 18S-NemaBase reference database (FASTA) [TO CONFIRM: version or download date]

4.2 Consumables

Not applicable.

4.3 Software

- OBITools4 (obipairing, obigrep, obimultiplex, obiuniq, obiannotate, obicount) [TO CONFIRM: version]
- VSEARCH, run from the conda environment "qiime2-amplicon-2023.9" [TO CONFIRM: VSEARCH version]
- R with Biostrings (Bioconductor), tidyverse, lulu (tobiasgf/lulu), seqinr, ggplot2, ggpubr, dplyr, jsonify, devtools [TO CONFIRM: R and package versions]
- BOLDigger3 (pip package boldigger3) [TO CONFIRM: version]
- Helper script ngs_filter.py (splits the sample sheet per sample) [TO CONFIRM: source and location in the repository]

5 Reagent preparation

Not applicable. Software installation (as in the source script):

```
curl -L https://metabarcoding.org/obitools4/install.sh | bash        # OBITools4
pip install boldigger3                                               # BOLDigger3 (pip install --upgrade
boldigger3 to update)
boldigger3 download_db ~/Bold_db                                     # local BOLD database
```

```
install.packages(c("tidyverse", "ggpubr", "seqinr", "jsonify", "devtools"))
if (!require("BiocManager", quietly = TRUE)) install.packages("BiocManager")
BiocManager::install("Biostrings")
devtools::install_github("tobiasgf/lulu", force = TRUE)
```

6 Procedure

6.1 1. Working folders and raw data

1. Set the working directory and create one folder per step:

```
mkdir files 1.raw 2.pairedend 3.obimultiplex 4.uniq 5.simplified 6.nosingletons 7.clean 8.vsearch 9.lulu
10.stats samplesheets reports
```

2. Move all raw .gz files (usually in one folder per sample) into `1.raw` and decompress them:

```
find ./All_Data_Raw/* -type f \( -name "PC*" -o -name "*raw*" \) -exec mv {} ./1.raw/ \;
gunzip -r ./1.raw
```

3. Rename the files to `<sample>_R1.fastq` and `<sample>_R2.fastq` (the source script adds a missing “R” and shortens names; adapt the patterns to the provider’s file names; Novogene delivers `<sample>.raw_1.fastq.gz` and `<sample>.raw_2.fastq.gz`) [TO CONFIRM: general renaming rule].

4. Start the run report, one row per metric, and define a short helper to add rows to it:

```
echo -e "section\tstep\tmetric\tvalue\tunit\tnotes" > reports/report_info.tsv
report() { echo -e "$1\t$2\t$3\t$4\t$5\t$6" >> reports/report_info.tsv; }
nreads() { echo $(( $(wc -l < "$1") / 4 )); }     # reads in a fastq file
nseqs()  { grep -c "^>" "$1"; }                   # sequences in a fasta file
```

The helpers exist only in the current terminal session; define them again if you open a new one. In the source script each row is written with its own `echo` command; the result is the same.

6.2 2. Sample sheets

5. Put the master sample sheet (Before you start) and the helper script in `files`, and split the sheet into one file per sample in `samplesheets`. Keep the master sheet out of `samplesheets`: every `.txt` file there is treated as one sample.

```
python3 -i ./files/ngs_filter.py -s ./files/samplesheet.txt -o ./samplesheets/
```

6. Report the number of samples and the raw reads:

```
report "Raw data" "Input" "Number of biological samples" $(awk 'NR>1' ./files/samplesheet.txt | wc -l)
samples "From master sample sheet"
TOTAL=0
for SAMPLE_SHEET in ./samplesheets/*.txt; do
  SAMPLE_NAME=$(basename "$SAMPLE_SHEET" .txt)
  TOTAL=$(( TOTAL + $(nreads ./1.raw/"${SAMPLE_NAME}"_R1.fastq) ))
done
report "Raw" "Output" "Read pairs before merging" $TOTAL reads "Sum across samples"
```

6.3 3. Merge paired-end reads

7. Align forward and reverse reads. With these settings, pairs that overlap by less than 10 bp or have less than 80% identity in the overlap are concatenated instead of aligned (`mode` = joined):

```
for SAMPLE_SHEET in ./samplesheets/*.txt; do
  SAMPLE_NAME=$(basename "$SAMPLE_SHEET" .txt)
  obipairing --min-identity=0.8 --min-overlap=10 \
    -F ./1.raw/"${SAMPLE_NAME}"_R1.fastq -R ./1.raw/"${SAMPLE_NAME}"_R2.fastq \
    > ./2.pairedend/"${SAMPLE_NAME}"_paired.fastq
done
```

8. Remove the unaligned (joined) records:

```
for SAMPLE_SHEET in ./samplesheets/*.txt; do
  SAMPLE_NAME=$(basename "$SAMPLE_SHEET" .txt)
  obigrep -p 'annotations.mode != "join"' ./2.pairedend/"${SAMPLE_NAME}"_paired.fastq \
    > ./2.pairedend/"${SAMPLE_NAME}"_paired.ali.fastq
done
```

9. Report the pairing settings and the reads kept and removed:

```
report "Pairing" "obipairing" "Min overlap" 10 bp ""
report "Pairing" "obipairing" "Min identity" 0.8 proportion ""
MERGED=0; ALIGNED=0
for SAMPLE_SHEET in ./samplesheets/*.txt; do
  SAMPLE_NAME=$(basename "$SAMPLE_SHEET" .txt)
  [ -f ./2.pairedend/"${SAMPLE_NAME}"_paired.fastq ] || continue
  M=$(nreads ./2.pairedend/"${SAMPLE_NAME}"_paired.fastq);
A=$(nreads ./2.pairedend/"${SAMPLE_NAME}"_paired.ali.fastq)
  report "Filtering" "Remove unaligned" "Reads kept" $A reads "$SAMPLE_NAME"
  report "Filtering" "Remove unaligned" "Reads removed" $(( M - A )) reads "$SAMPLE_NAME"
  MERGED=$(( MERGED + M )); ALIGNED=$(( ALIGNED + A ))
done
report "Pairing" "Output" "Reads after merging" $MERGED reads "Sum across samples"
report "Filtering" "Remove unaligned" "Total reads removed" $(( MERGED - ALIGNED )) reads "All samples"
```

6.4 4. Primer removal and sample assignment

10. Remove primers and annotate each sequence with its sample information, allowing four mismatches in the primers (`-e 4`); unassigned sequences go to a separate file:

```
for SAMPLE_SHEET in ./samplesheets/*.txt; do
  SAMPLE_NAME=$(basename "$SAMPLE_SHEET" .txt)
  obimultiplex -t ./samplesheets/"${SAMPLE_NAME}".txt -e 4 \
    -u ./3.obimultiplex/"${SAMPLE_NAME}"_unidentified.fastq \
    ./2.pairedend/"${SAMPLE_NAME}"_paired.ali.fastq \
    > ./3.obimultiplex/"${SAMPLE_NAME}"_paired.ali.assigned.fastq
done
```

11. Report the reads assigned and not assigned to a sample:

```
AS=0; UN=0
for f in ./3.obimultiplex/*_paired.ali.assigned.fastq; do AS=$(( AS + $(nreads "$f") )); done
for f in ./3.obimultiplex/*_unidentified.fastq; do UN=$(( UN + $(nreads "$f") )); done
report "Filtering" "obimultiplex" "Total reads assigned" $AS reads "All samples"
report "Filtering" "obimultiplex" "Total reads unassigned" $UN reads "All samples"
```

6.5 5. Dereplication and singleton removal

12. Collapse identical reads into exact sequence variants and keep only the `count` and `sample` annotations:

```
for SAMPLE_SHEET in ./samplesheets/*.txt; do
  SAMPLE_NAME=$(basename "$SAMPLE_SHEET" .txt)
  obiuniq -O ./3.obimultiplex/"${SAMPLE_NAME}"_paired.ali.assigned.fastq \
    > ./4.uniq/"${SAMPLE_NAME}"_paired.ali.assigned.uniq.fasta
  obiannotate -k count -k sample ./4.uniq/"${SAMPLE_NAME}"_paired.ali.assigned.uniq.fasta \
    > ./5.simplified/"${SAMPLE_NAME}"_paired.ali.assigned.simple.fasta
done
```

13. Record statistics per sample in `10.stats` (merged reads at the start, unique sequences, singletons) with `obicount`, and report the totals:

```
for SAMPLE_SHEET in ./samplesheets/*.txt; do
  SAMPLE_NAME=$(basename "$SAMPLE_SHEET" .txt)
  F=./5.simplified/"${SAMPLE_NAME}"_paired.ali.assigned.simple.fasta
  [ -f "$F" ] || continue
  obicount "$F" > ./10.stats/"${SAMPLE_NAME}".final_metrics.txt
  obigrep -p 'sequence.Count() == 1' "$F" | obicount > ./10.stats/"${SAMPLE_NAME}".singleton_metrics.txt
```

```
done
report "Dereplication" "QC" "Total unique sequences (sum of samples)" $(cat ./10.stats/
*.final_metrics.txt | awk '{s+=$1} END {print s}') sequences "All samples"
report "Filtering" "Singleton removal" "Total singletons" $(cat ./10.stats/*.singleton_metrics.txt | awk
'{s+=$1} END {print s}') sequences "All samples"
```

> Check the first lines of each `obicount` output once: the column used for the sum must be the number of sequences (or reads, with `--reads`) for the OBITools4 version installed.

14. Remove singletons and check that no sequence with count 1 remains:

```
for SAMPLE_SHEET in ./samplesheets/*.txt; do
  SAMPLE_NAME=$(basename "$SAMPLE_SHEET" .txt)
  obigrep -p 'sequence.Count() > 1' ./5.simplified/"${SAMPLE_NAME}"_paired.ali.assigned.simple.fasta \
    > ./6.nosingletons/"${SAMPLE_NAME}"_paired.ali.assigned.simple.nosingle.fasta
done
KEPT=0; for f in ./6.nosingletons/*.fasta; do KEPT=$(( KEPT + $(nseqs "$f") )); done
report "Filtering" "Singleton removal" "Sequences kept after singleton removal" $KEPT sequences "All
samples"
```

6.6 6. Merge samples and inspect

15. Concatenate all samples and report the total: `cat ./6.nosingletons/*.fasta > all.fasta`, then `report "Sanity" "Merge samples" "Sequences after merging all samples" $(nseqs all.fasta) sequences "All samples"`.

16. In R, plot the number of occurrences per variant and the sequence length distribution, and add the summary to the report. The same function is used again after denoising (step 7) and after chimera removal:

```
library(Biostrings); library(tidyverse); library(ggpubr)

inspect <- function(fasta, count_pattern, png, min_len, max_len, binwidth = 10) {
  seqs <- readDNAStringSet(fasta)
  counts <- as.numeric(str_extract(names(seqs), count_pattern))
  df <- tibble(count = counts, length = width(seqs))
  a <- ggplot(df, aes(count)) + geom_histogram(bins = 100) + scale_x_sqrt() + scale_y_sqrt() +
    geom_vline(xintercept = 10, colour = "red", linetype = 2) + xlab("Occurrences of a variant")
  b <- ggplot(df, aes(length)) + geom_histogram(binwidth = binwidth) + scale_y_log10() +
    geom_vline(xintercept = c(min_len, max_len), colour = "red", linetype = 2) + xlab("Sequence length
(bp)")
  ggsave(png, ggarrange(a, b, labels = c("A", "B"), ncol = 2), width = 10, height = 5, dpi = 300)
  rows <- tibble(section = "QC", step = basename(fasta),
                 metric = c("Variants", "Min count", "Median count", "Max count", "Min length", "Median
length", "Max length"),
                 value = c(length(seqs), min(counts), median(counts), max(counts), min(df$length),
median(df$length), max(df$length)),
                 unit = c("sequences", rep("reads", 3), rep("bp", 3)), notes = "All samples")
  write_tsv(rows, "reports/report_info.tsv", append = TRUE)
  invisible(df)
}

inspect("all.fasta", '(?<="count":)\\d+', "1_sequence_qc_plots.png", 400, 450)   # COI limits
```

6.7 7. VSEARCH: denoising, chimeras and clustering

17. Convert the headers to VSEARCH format (`count` → `size`):

```
sed '/^>/ s/{"count":\([0-9]\+\),"sample":"\([^"]\+\)"}/count=\1; sample=\2/g' all.fasta > all_sa.fasta
sed 's/\;\ count=/\;size=/g;s/\ count=/\;size=/g' all_sa.fasta | sed 's/\;\ /\;/g;s/\ /\;/g'
> ./8.vsearch/file_vsearch.fasta
grep '^>' ./8.vsearch/file_vsearch.fasta | head    # expected: >read_id;size=4;sample=P1.10
```

The first command accepts any sample name, including names with dots (e.g. P1.10); a pattern limited to letters and digits leaves such headers unconverted.

18. Dereplicate across the whole dataset and relabel:

```
vsearch --derep_fulllength ./8.vsearch/file_vsearch.fasta --output ./8.vsearch/file_dereplicate.fasta \
  --sizein --sizeout --relabel otu --uc 10.stats/dereplicate.txt
```

19. Denoise with UNOISE, keeping sequences of the expected length (COI: 400–450 bp; target 418 bp):

```
vsearch --cluster_unoise ./8.vsearch/file_dereplicate.fasta --minseqlength 400 --maxseqlength 450 \
  --sizein --sizeout --centroids ./8.vsearch/file_centroids.fasta --uc ./10.stats/centroids.txt
```

20. Remove chimeras (UCHIME3 de novo; by default a parent must be at least 16 times more abundant than its chimera):

```
vsearch --uchime3_denovo ./8.vsearch/file_centroids.fasta --sizein --notrunclabels --fasta_width 0 \
  --nonchimeras ./8.vsearch/file_nonchim.fasta --chimeras ./8.vsearch/file_chim.fasta \
  --uchimeout ./10.stats/chimera.txt
```

21. Report the sequences after each VSEARCH step and inspect the denoised and non-chimeric sequences with the R function of step 6 (count pattern `'(?<=size=)\\d+'`, `binwidth = 1`):

```
report "VSEARCH" "Dereplication" "Unique sequences" $(nseqs ./8.vsearch/file_dereplicate.fasta) sequences
"All samples"
report "VSEARCH" "UNOISE" "Centroids" $(nseqs ./8.vsearch/file_centroids.fasta) sequences "All samples"
report "VSEARCH" "Chimera removal" "Non-chimeric sequences" $(nseqs ./8.vsearch/file_nonchim.fasta)
sequences "All samples"
report "VSEARCH" "Chimera removal" "Chimeras removed" $(nseqs ./8.vsearch/file_chim.fasta) sequences "All
samples"
```

22. Cluster into OTUs at 96% similarity. Adapt the threshold to the intraspecific genetic variability expected for the target group:

```
vsearch --cluster_size ./8.vsearch/file_nonchim.fasta --id 0.96 --centroids ./8.vsearch/file_otus.fasta \
  --sizein --relabel otu -uc ./10.stats/cluster_uc.txt
report "VSEARCH" "OTU clustering" "OTUs" $(nseqs ./8.vsearch/file_otus.fasta) OTUs "All samples"
```

> **CRITICAL** 96% is the lab rule: 3–4% divergence is acceptable within an invertebrate (e.g. arthropod) species. Use 99% (`--id 0.99`) only for groups known to have very low intraspecific variability, and record the threshold used. (The December 2025 scripts used 99%.)

6.8 8. LULU curation (erroneous OTUs and NUMTs)

23. Build the OTU table (OTU frequency per sample) and the match list (pairwise similarity among OTUs):

```
vsearch --usearch_global ./8.vsearch/file_vsearch.fasta -db ./8.vsearch/file_otus.fasta -id .99 \
  --relabel "sample_#sample" -otutabout ./8.vsearch/file_otutab.tsv
vsearch --usearch_global ./8.vsearch/file_otus.fasta --db ./8.vsearch/file_otus.fasta --self --id .84 \
  --iddef 1 --userout ./8.vsearch/file_matchlist.txt -userfields query+target+id --maxaccepts 0 \
  --query_cov .9 --maxhits 10
```

24. Run LULU in R and write the curated table and the curated OTU sequences:

```
otutab <- read.delim("./8.vsearch/file_otutab.tsv", row.names = 1, as.is = TRUE, header = TRUE)
matchlist <- read.delim("./8.vsearch/file_matchlist.txt", header = FALSE, stringsAsFactors = FALSE, as.is
= TRUE)
lulu.output <- lulu(otutab, matchlist, minimum_ratio_type = "min", minimum_ratio = 1,
                    minimum_match = 84, minimum_relative_cooccurence = 0.95)
write.table(lulu.output$curated_table, file = "./9.lulu/otutable_curated.tsv", sep = "\t")
otufasta <- read.fasta(file = "./8.vsearch/file_otus.fasta")
otu.curated <- otufasta[which(names(otufasta) %in% lulu.output$curated_otus)]
write.fasta(otu.curated, names(otu.curated), file.out = "./9.lulu/curated_OTUs.fasta", nbchar = 500)
```

25. Copy the curated OTU table to `reports`: `cp ./9.lulu/otutable_curated.tsv ./reports/otutable_curated.txt`, and report the OTU table size (OTUs, samples, reads assigned) and the number of curated OTUs:

```
T=./8.vsearch/file_otutab.tsv
report "VSEARCH" "OTU table" "OTUs" $(( $(wc -l < $T) - 1 )) OTUs "All samples"
report "VSEARCH" "OTU table" "Samples" $(head -1 $T | awk -F'\t' '{print NF-1}') samples "All samples"
report "VSEARCH" "OTU table" "Reads assigned" $(awk 'NR>1{for(i=2;i<=NF;i++) s+=$i} END{print s}' $T)
reads "All samples"
report "LULU" "Curation" "Curated OTUs" $(nseqs ./9.lulu/curated_OTUs.fasta) OTUs "All samples"
```

6.9 9. Taxonomic assignment (BOLDigger3)

26. Identify the curated OTUs against the local BOLD database:

```
boldigger3 identify ./9.lulu/curated_OTUs.fasta ~/Bold_db --db 3 --mode 1 --thresholds 95 90 85 80
```

6.10 10. Variant: 18S nematode amplicons (NF1/18Sr2b)

NF1/18Sr2b is the lab's routine 18S primer pair (amplicon about 320 bp without primers). Steps 1–8 are the same, with these changes.

27. Step 3: only merged (aligned) read pairs are used. Pairs that do not merge well are discarded for now; how to use unmerged pairs is under development in the lab and not part of this protocol.

28. Step 7, UNOISE: keep sequences of 280–430 bp (`--minseqlength 280 --maxseqlength 430`), and use the same limits in the R inspection. This range comes from the December 2025 primer test, which also included TAReukFWD1/TAReukREV3 (about 390 bp); it is provisional for NF1/18Sr2b alone [TO CONFIRM: final range]. Amplicon length varies a little between species.

29. Step 7, clustering: 96%, as for COI, unless the group is known to have very low intraspecific variability.

30. Instead of step 9, identify the curated OTUs against 18S-NemaBase with VSEARCH (global alignment, at least 84% identity over at least 90% of the query; all hits up to 10 are kept, sorted by identity):

```
vsearch --usearch_global ./9.lulu/curated_OTUs.fasta --db ./18S_NemaBase/18S_NemaBase.fasta   --
userout ./reports/NemaBase_hits.txt --userfields query+target+id   --id 0.84 --query_cov 0.9 --maxaccepts
0 --maxhits 10
awk 'BEGIN{OFS="\t"} !seen[$1]++' ./reports/NemaBase_hits.txt > ./reports/NemaBase_best_hits.txt
```

31. Split the taxonomy of the best hit into ranks in R. Each 18S-NemaBase header holds the accession followed by 16 ranks separated by semicolons (domain to species); missing ranks are written as `NA_<rank>`:

```
hits <- read.delim("reports/NemaBase_best_hits.txt", header = FALSE, col.names = c("OTU", "Target",
"Identity"))
ranks <- c("Domain", "Kingdom", "Phylum", "Subphylum", "Superclass", "Class", "Subclass", "Superorder",
           "Order", "Suborder", "Infraorder", "Superfamily", "Family", "Subfamily", "Genus", "Species")
tax <- do.call(rbind, lapply(strsplit(sub("^[^_]+_", "", hits$Target), ";"), function(x) { length(x) <-
16; x }))
colnames(tax) <- ranks
tax[grepl("^NA_", tax)] <- NA
taxonomy_table <- data.frame(OTU = hits$OTU, Identity = hits$Identity, tax)
write.table(taxonomy_table, "reports/curated_OTUs_taxonomy.tsv", sep = "\t", quote = FALSE, row.names =
FALSE)
```

32. Accept a name only at the rank its identity supports: species at ≥ 99%, genus at ≥ 97%, family at ≥ 95%. Below these, report the next higher rank that is supported (for example the family, or only "Nematoda"); never force a genus or species name from a weak match:

```
taxonomy_table$Species[taxonomy_table$Identity < 99] <- NA
taxonomy_table$Genus[taxonomy_table$Identity < 97]   <- NA
taxonomy_table$Family[taxonomy_table$Identity < 95]  <- NA
write.table(taxonomy_table, "reports/curated_OTUs_taxonomy_accepted.tsv", sep = "\t", quote = FALSE,
row.names = FALSE)
```

CRITICAL The search keeps matches from 84% so that distant OTUs are still placed in Nematoda; 84% does not support a genus or species name. Record the thresholds used. 18S reference coverage for nematodes is incomplete, so many OTUs will stop at family or above.

6.11 11. Read losses per step and per sample

33. Collect the reads of every sample at each step into one table (long format: sample, step, reads):

```
OUT=reports/read_tracking.tsv
echo -e "sample\torder\tstep\treads" > $OUT
sumcount() { grep -o '"count":[0-9]*' "$1" | cut -d: -f2 | awk '{s+=$1} END {print s+0}'; }
for SAMPLE_SHEET in ./samplesheets/*.txt; do
  S=$(basename "$SAMPLE_SHEET" .txt)
  echo -e "$S\t1\traw read pairs\t$(nreads ./1.raw/${S}_R1.fastq)" >> $OUT
  echo -e "$S\t2\tpaired\t$(nreads ./2.pairedend/${S}_paired.fastq)" >> $OUT
  echo -e "$S\t3\taligned (unaligned removed)\t$(nreads ./2.pairedend/${S}_paired.ali.fastq)" >> $OUT
  echo -e "$S\t4\tassigned to sample (primers found)\t$(nreads ./3.obimultiplex/${S}
_paired.ali.assigned.fastq)" >> $OUT
  echo -e "$S\t5\treads in non-singleton variants\t$(sumcount ./6.nosingletons/${S}
_paired.ali.assigned.simple.nosingle.fasta)" >> $OUT
done
# reads per sample in the OTU table and in the curated table (column sums)
awk -F'\t' 'NR==1{for(i=2;i<=NF;i++) n[i]=$i; next} {for(i=2;i<=NF;i++) s[i]+=$i} END{for(i in n) print
n[i]"\t6\tin OTU table\t"s[i]}' ./8.vsearch/file_otutab.tsv >> $OUT
```

34. In R, compute for each sample and step the reads in, reads out, reads lost and percentage lost, flag any step that loses more than half of its input, and add dataset totals:

```
library(tidyverse)
trk <- read_tsv("reports/read_tracking.tsv", show_col_types = FALSE)
loss <- trk |>
  arrange(sample, order) |>
  group_by(sample) |>
  mutate(reads_in = lag(reads), reads_out = reads, lost = reads_in - reads_out,
         pct_lost = round(100 * lost / reads_in, 1),
         flag = if_else(!is.na(pct_lost) & pct_lost > 50, "CHECK", "")) |>
  ungroup() |> filter(!is.na(reads_in)) |>
  select(sample, step, reads_in, reads_out, lost, pct_lost, flag)
totals <- loss |> group_by(step) |>
  summarise(sample = "ALL", reads_in = sum(reads_in), reads_out = sum(reads_out), .groups = "drop") |>
  mutate(lost = reads_in - reads_out, pct_lost = round(100 * lost / reads_in, 1),
         flag = if_else(pct_lost > 50, "CHECK", ""))
write_tsv(bind_rows(loss, totals), "reports/read_losses.tsv")
filter(bind_rows(loss, totals), flag == "CHECK")      # steps to look at
```

The sample names in the OTU table come from the `sample` annotation; if they differ from the sample-sheet file names, the OTU-table rows will not line up and must be matched by hand. The table is added to as the run progresses; rerun both blocks after any step is repeated.

35. Check every flagged step before going on. A large loss at merging usually means short overlaps (amplicon too long for the reads), at primer assignment a sample-sheet or primer problem, and at singleton removal a sample with very few reads.

7 Calculations

Not applicable.

8 Quality control

Check	Where	Acceptance criterion
Reads per sample in the final table	`reports/read_losses.tsv`	Judged against the whole data set. < 1 000 reads: re-sequence. < 500 reads:

Check	Where	Acceptance criterion
		not used by default, but always checked first (one individual in a tube can give few reads); no sample is dropped automatically
Reads lost at each step, per sample	`reports/read_losses.tsv` (step 11)	Any step losing more than 50% of its input is checked before going on
Singletons removed	`6.nosingletons`	No sequence with count 1
Length distribution	R plots (steps 6 and 7)	Main peak within 400–460 bp (COI); within 280–430 bp (18S, provisional)
Negative controls (extraction blanks, PCR negatives)	OTU table	First identify what the OTUs in the blanks are and whether they point to cross-contamination between samples (e.g. a sample's dominant OTU in a blank); then subtract the reads found in the negative controls from the samples, OTU by OTU [TO CONFIRM: subtract the highest count in any blank, or the sum]
Mock community (if sequenced)	OTU table	[TO CONFIRM]

9 Data and metadata

- **Outputs:** `reports/otutable_curated.txt` (curated OTU table), `9.lulu/curated_OTUs.fasta`, BOLDigger3 identification table (COI) or `reports/curated_OTUs_taxonomy.tsv` (18S), the run report `reports/report_info.tsv`, the inspection plots, per-step statistics in `10.stats`.
- **Code:** the source script is kept with the protocol sources; it is to be added to this repository (`code/`) once personal paths are removed [TO CONFIRM].
- **Where:** project data folder [TO CONFIRM] raw reads deposited in [TO CONFIRM: public archive].

Minimum to record

- Run and sample sheet used; sequencing run ID
- Software versions (OBITools4, VSEARCH, R and packages, BOLDigger3) and BOLD database download date
- All non-default parameters (as in this protocol)
- Reads per sample at each step (`10.stats`) and the run report (`reports/report_info.tsv`)
- Reference database and version (BOLD download date; 18S-NemaBase version)

10 Safety and waste

Not applicable (computational protocol).

11 Troubleshooting

Problem	Likely cause	Solution
Many reads in `_unidentified.fastq`	Primers or sample sheet do not match the reads	Check primer sequences and tags in the sample sheet.
Many joined (unaligned) pairs	Amplicon longer than the combined reads (e.g. long 18S amplicons)	Check marker length against the read length (EnGeL-MOLB-MB, Primer table).
Sequences with count 1 after step 5	Singleton filter not applied	Re-run the `obigrep -p 'sequence.Count() > 1'` step.

Problem	Likely cause	Solution
Headers still in `{"count":…}` form after conversion	Sample names with characters the pattern did not expect	Use the conversion in step 7, which accepts any sample name; check with `grep '^>' … | head`.
`report: command not found`	Helpers not defined in this terminal session	Run the helper definitions of step 1 again.

12 Remarks

Parameters are written for COI (BF3/BR2; amplicon 458 bp with primers, target 418 bp). The source script also mentions primers III_B_F/Fol-degen-rev (464 bp) with the same length filter. The clustering threshold (96%, the lab rule) follows the intraspecific genetic variability expected for the target group; 99% only for groups with very low variability. Over-denoising can remove true variability and under-denoising keeps erroneous OTUs; denoising settings were developed for bacterial 16S and need care for COI.

13 References

[TO CONFIRM: OBITools4, VSEARCH, UNOISE, UCHIME3, LULU, BOLDigger3 and Biostrings references.]

Gattoni, K., Gendron, E.M.S., Sandoval-Ruiz, R., Borgemeier, A., McQueen, J.P., Shepherd, R.M., Slos, D., Powers, T.O., & Porazinska, D.L. (2023). 18S-NemaBase: curated 18S rRNA database of nematode sequences. *Journal of Nematology*, 55(1), e2023-0006. https://doi.org/10.2478/jofnem-2023-0006

Elbrecht, V., Braukmann, T.W.A., Ivanova, N.V., et al. (2019). Validation of COI metabarcoding primers for terrestrial arthropods. PeerJ, 7, e7745. https://doi.org/10.7717/peerj.7745

14 Version history and citation

Version	Date	Change	By
1.0.0	2026-11-16	First EnGeL-UC version, from the Quarto script "Obitools4 – Script" by Luísa F. Dornellas and Luís Cunha (2026).	Luísa Fraga, Luís Cunha
1.0.1	2026-11-16	Clustering threshold confirmed at 96% (adapt to the intraspecific variability expected for the group).	Luís Cunha
1.0.2	2026-11-17	Contributor: Andreia Quaresma.	Luís Cunha
1.0.3	2026-11-17	Authorship: all who contributed are authors, with their roles; lead author first where identified, reviewer (Luís Cunha) last.	Luís Cunha
1.1.0	2026-11-20	From Luísa F. Dornellas's scripts of December 2025: run report (reads and sequences after each step in reports/report_info.tsv); R inspection with Biostrings instead of ROBIFastread, also after denoising and af-	Luísa Fraga, Luís Cunha

Version	Date	Change	By
		ter chimera removal; header conversion that accepts sample names with dots; sample-sheet loops over all samples; 18S nematode variant (length filter and identification against 18S-NemaBase with VSEARCH).	
1.2.0	2026-11-20	Values confirmed by Luís Cunha: clustering at 96% as the lab rule (99% only for groups with very low intraspecific variability); NF1/18Sr2b is the routine 18S pair, merged reads only, 280–430 bp provisional; 18S-NemaBase names accepted at ≥ 99% (species), ≥ 97% (genus), ≥ 95% (family), otherwise the next higher rank; read-loss table per step and sample, steps losing more than half the reads flagged; samples judged against the whole data set (< 1 000 reads: resequence; < 500: not used by default, always checked first); negative controls: identify, then subtract their reads.	Luís Cunha

How to cite

Fraga L, Quaresma A, Cunha L (2026). Metabarcoding bioinformatics with OBITools4, VSEARCH, LULU and BOLDigger3 (COI amplicons; 18S nematode variant). EnGeL-UC protocol EnGeL-BIOINF-OBITOOLS4, version 1.2.0. Zenodo. DOI [TO CONFIRM].

Licence

Text: CC BY 4.0. Code: MIT. Logos are the property of their institutions and are not covered by the licence.

15 Open questions

1. Software versions (OBITools4, VSEARCH, R and packages, BOLDigger3); BOLD database version or date; 18S-NemaBase version (ask Luísa Fraga, and which identity she accepted in practice).
2. 18S: final length range for NF1/18Sr2b alone.
3. Source and location of `ngs_filter.py`; general file-renaming rule.
4. Negative controls: subtract the highest read count of an OTU in any blank, or the sum over blanks?
5. Add the script (without personal paths) to the repository under `code/`? Licence MIT.
6. Where outputs are kept; public archive for raw reads.
7. References for the tools; DOI.
8. Mock community: is one sequenced, and what is the criterion?