

```
---
title: "ADM-TD3"
output: github_document
date: "2023-11-29"
---
```

```
``{r library, include=FALSE}
```

```
library(stringr)
library(rmarkdown)
library(knitr)
library(phyloseq)
library(dada2)
library(DECIPHER)
library(phangorn)
library(ggplot2)
library(gridExtra)
library(shiny)
library(miniUI)
library(caret)
library(pls)
library(e1071)
library(ggplot2)
library(randomForest)
library(dplyr)
library(ggrepel)
#library(nlme)
library(devtools)
library(reshape2)
library(PMA)
#library(structSSI)
library(ade4)
library(ggnetwork)
library(intergraph)
library(scales)
library(phyloseqGraphTest)
library(Biostrings)
library(RSQLite)
library(parallel)
library(ape)
```

```
---
```

```
``{r}
```

```
refdb_folder <- here::here("TD3","data")
refdb_folder
if (!dir.exists(refdb_folder)) dir.create(refdb_folder, recursive = TRUE)
---
```

```
``{bash, eval=FALSE}
```

```
cp -R course-material-main/data/raw ./data
```

```
'''
```

```
'''{r}
```

```
# so we change timeout to be 20 minutes
```

```
options(timeout = 1200)
```

```
# we save in variable the path to the refdb
```

```
# in the working space
```

```
silva_train_set <- file.path (refdb_folder, "silva_nr99_v138.1_train_set.fa.gz")
```

```
silva_species_assignment <- file.path(refdb_folder,  
                                       "silva_species_assignment_v138.1.fa.gz")
```

```
# then we download the files if they don't already exist
```

```
if (!file.exists(silva_train_set)) {
```

```
  download.file(
```

```
    "https://zenodo.org/record/4587955/files/silva_nr99_v138.1_train_set.fa.gz",
```

```
    silva_train_set,
```

```
    quiet = TRUE
```

```
  )
```

```
}
```

```
if (!file.exists(silva_species_assignment)) {
```

```
  download.file(
```

```
    "https://zenodo.org/record/4587955/files/silva_species_assignment_v138.1.fa.gz",
```

```
    silva_species_assignment,
```

```
    quiet = TRUE
```

```
  )
```

```
}
```

```
'''
```

```
'''{r}
```

```
path_to_fastqs <- "data"
```

```
list.files(path_to_fastqs)
```

```
'''
```

```
'''{r}
```

```
fnFs <- sort(list.files(path_to_fastqs,  
                        pattern = "_R1.fastq.gz",  
                        full.names = TRUE))
```

```

fnRs <- sort(list.files(path_to_fastqs,
                        pattern = "_R2.fastq.gz",
                        full.names = TRUE))
...

```{r}

sample.names <- sapply(strsplit(basename(fnFs), "_"), `[`, 1)
...

```{r}
sample.names <- basename(fnFs) |>
  strsplit(split = "_") |> # diviser la chaîne de caractères selon le modèle mis entre ""
  sapply(head, 1) # appliquer une fonction à chaque élément d'une liste
...

```{r}
basename(fnFs) |>
  strsplit(split = "_") |>
  head()
...

```{r}
gsub ("^./|_.$", "", fnFs) |> head()
...

```{r}
file.exists(fnFs)
file.exists(fnRs)
...

...

...

# Filter and trim

```{r}
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
...

```{r}
out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs, trimLeft=c(50,55), truncLen =c(145,130),
  maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,
  compress=TRUE, multithread=TRUE)
head(out)
...

```