
title: "R Notebook"

output:

pdf_document: default

html_notebook: default

word_document: default

```{r}

library(gitcreds)

gitcreds\_set()

...

```{r}

fichier_texte <- "Home/Rstudio/26589611"

...

```{r}

refdb\_folder <- here::here("data", "refdb")

refdb\_folder

...

```{r, eval=FALSE}

if (!dir.exists(refdb_folder))

dir.create(refdb_folder, recursive = TRUE)

...

```{bash}

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

...

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

```
getOption("timeout")
```

```
```
```

# on défini une nouvelle variable qui reçoit le chemin dans fdb folder.

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

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

```
```
```

# On créé une nouvelle variable

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

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

```
```
```

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

```
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  
  )  
}
```

```
...
```

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

```
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, echo=FALSE}
```

```
devtools::load_all(  
  path = "Home/Rstudio/26589611/")  
...
```

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

```
path_to_fastqs <- here::here("data", "raw")
...
```

```
path_to_fastqs permet de montrer les chemins qui vont vers les fichiers
```

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

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

```
``{r}
```

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

```
``
```

```
``{r}
```

```
sample_names <- basename(fnFs) |>  
  strsplit(split = "_") |>  
  sapply(head, 1)
```

```
``
```

```
``{r}
```

```
basename(fnFs) |>  
  head()
```

```
``
```

```
## coupe/sépare au niveau des tirets (_)
```

```
``{r}
```

```
basename(fnFs) |>  
  strsplit(split = "_") |>  
  head()
```

```
``
```

```
## On ne prend seulement que le premier élément de cette liste.
```

```
``{r}
```

```
basename(fnFs) |>
```

```
strsplit(split = "_") |>
  sapply(head, 1) |>
  head()
...

```

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

```

```
```{r}
# create a directory for the outputs
quality_folder <- here::here("outputs",
                             "dada2",
                             "quality_plots")

if (!dir.exists(quality_folder)) {
  dir.create(quality_folder, recursive = TRUE)
}

```

```
...

```

```
```{r}
path_to_trimmed_reads <- here::here(
 "outputs",
 "dada2",

```

```
"trimmed"
)
```

```
if (!dir.exists(path_to_trimmed_reads)) dir.create(path_to_trimmed_reads, recursive = TRUE)
...
```

```
```{r}  
primer_fwd <- "CCTACGGGNBGCASCAG"  
primer_rev <- "GACTACNVGGGTATCTAAT"  
...
```

```
```{r}  
Biostrings::readDNASTringSet(
 fnFs[1],
 format = "fastq",
 nrec = 10
)
...
```

```
```{r}  
Biostrings::readDNASTringSet(  
  fnRs[1],  
  format = "fastq",  
  nrec = 10  
)  
...
```

```
```{bash}
```

```
pwd
```

```
cp -R /home/rstudio/ADM2023_tutoriel/course-material-main/bash .
```

```
...
```

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

```
nopFw <- sort(list.files(path_to_trimmed_reads, pattern = "R1", full.names = TRUE))
```

```
nopRv <- sort(list.files(path_to_trimmed_reads, pattern = "R2", full.names = TRUE))
```

```
print(nopRv)
```

```
print(nopFw)
```

```
...
```

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

```
nopFw <- sort(list.files(path_to_trimmed_reads, pattern = "R1", full.names = TRUE))
```

```
nopRv <- sort(list.files(path_to_trimmed_reads, pattern = "R2", full.names = TRUE))
```

```
print(nopRv)
```

```
print(nopFw)
```

```
...
```

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

```
path_to_filtered_reads <- here::here("outputs", "dada2", "filtered")
```

```
if (!dir.exists(path_to_filtered_reads)) dir.create(path_to_filtered_reads, recursive = TRUE)
```

```
...
```

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

```
filtFs <- file.path(path_to_filtered_reads, basename(fnFs))
```

```
filtRs <- file.path(path_to_filtered_reads, basename(fnRs))
```

```
...
```

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

```
names(filtFs) <- sample_names
```

```
names(filtRs) <- sample_names
```

```
...
```

```
``{r}
```

```
(out <- dada2::filterAndTrim(
```

```
  fwd = nopFw,
```

```
  filt = filtFs,
```

```
  rev = nopRv,
```

```
  filt.rev = filtRs,
```

```
  minLen = 150,
```

```
  matchIDs = TRUE,
```

```
  maxN = 0,
```

```
  maxEE = c(3, 3),
```

```
  truncQ = 2
```

```
))
```

```
...
```

```
``{r}
```

```
errF <- dada2::learnErrors(filtFs,
```

```
  randomize = TRUE,
```

```
  multithread = TRUE)
```

```
...
```

```
``{r}
```

```
errR <- dada2::learnErrors(filtRs,
```

```
  randomize = TRUE,
```

```
  multithread = TRUE)
```

```
...
```



```
``{r}
```

```
dada2::plotErrors(errF, nominalQ=TRUE)
```

```
``
```

```
``{r}
```

```
derepFs <- dada2::derepFastq(filtFs, verbose = TRUE)
```

```
derepRs <- dada2::derepFastq(filtRs, verbose = TRUE)
```

```
``
```

```
``{r}
```

```
dadaFs <- dada2::dada(derepFs, err = errF, multithread = TRUE)
```

```
``
```

```
``{r}
```

```
dadaRs <- dada2::dada(derepRs, err = errR, multithread = TRUE)
```

```
``
```

```
``{r}
```

```
mergers <- dada2::mergePairs(
```

```
  dadaF = dadaFs,
```

```
  derepF = derepFs,
```

```
  dadaR = dadaRs,
```

```
  derepR = derepRs,
```

```
  maxMismatch = 0,
```

```
  verbose = TRUE
```

```
)
```

```
...
```

```
``{r}
```

```
seqtab <- dada2::makeSequenceTable(mergers)
```

```
...
```

```
``{r}
```

```
seqtab_nochim <- dada2::removeBimeraDenovo(seqtab,  
                                             method = "consensus",  
                                             multithread = TRUE,  
                                             verbose = TRUE)
```

```
...
```

```
``{r}
```

```
export_folder <- here::here("outputs", "dada2", "asv_table")
```

```
if (!dir.exists(export_folder)) dir.create(export_folder, recursive = TRUE)
```

```
saveRDS(object = seqtab_nochim,  
         file = file.path(export_folder, "seqtab_nochim.rds"))
```

```
...
```

```
``{r}
```

```
asv_seq <- colnames(seqtab_nochim)
```

```
...
```

```
``{r}
```

```
ndigits <- nchar(length(asv_seq))
```

```
asv_id <- sprintf(paste0("ASV_%0", ndigits, "d"), seq_along(asv_seq))
```

```
````
```

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

```
seqtab_nochim_export <- t(seqtab_nochim)
```

```
````
```

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

```
cat(paste0(">", names(asv_seq), "\n", asv_seq),
```

```
  sep = "\n",
```

```
  file = file.path(export_folder, "asv.fasta"))
```

```
````---
```

```
title: "R Notebook"
```

```
output:
```

```
 pdf_document: default
```

```
 html_notebook: default
```

```
 word_document: default
```

```

```

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

```
library(gitcreds)
```

```
gitcreds_set()
```

```
````
```

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

```
fichier_texte <- "Home/Rstudio/26589611"
```

```
...
```

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

```
refdb_folder <- here::here("data", "refdb")
```

```
refdb_folder
```

```
...
```

```
```{r, eval=FALSE}
```

```
if (!dir.exists(refdb_folder))
```

```
  dir.create(refdb_folder, recursive = TRUE)
```

```
...
```

```
```{bash}
```

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

```
...
```

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

```
getOption("timeout")
```

```
...
```

on défini une nouvelle variable qui reçoit le chemin dans fdb folder.

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

```
silva_train_set <- file.path(fichier_texte,
```

```
 "silva_nr99_v138.1_train_set.fa.gz")
```

```
...
```

# On créé une nouvelle variable

```
``{r}
```

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

```
``
```

```
``{r}
```

```
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
)
}
```

```
``
```

```
``{r}
```

```
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, echo=FALSE}
```

```
devtools::load_all(
```

```
 path = "Home/Rstudio/26589611/")
```

```
``
```

```
``{r}
```

```
path_to_fastqs <- here::here("data", "raw")
```

```
``
```

```
path_to_fastqs permet de montrer les chemins qui vont vers les fichiers
```

```
``{r}
```

```
fnFs <- sort(list.files(path_to_fastqs,
```

```
 pattern = "_R1.fastq.gz",
```

```
 full.names = TRUE))
```

```
``
```

```
``{r}
```

```
fnRs <- sort(list.files(path_to_fastqs,
```

```
 pattern = "_R2.fastq.gz",
```

```
 full.names = TRUE))
```

```
``
```

```
``{r}
```

```
sample_names <- basename(fnFs) |>
```

```
 strsplit(split = "_") |>
```

```
 sapply(head, 1)
```

```
``
```

```
``{r}
```

```
basename(fnFs) |>
```

```
head()
```

```
``
```

```
coupe/sépare au niveau des tirets (_)
```

```
``{r}
```

```
basename(fnFs) |>
```

```
strsplit(split = "_") |>
```

```
head()
```

```
``
```

```
On ne prend seulement que le premier élément de cette liste.
```

```
``{r}
```

```
basename(fnFs) |>
```

```
strsplit(split = "_") |>
```

```
sapply(head, 1) |>
```

```
head()
```

```
``
```

```
``{r}
```

```
gsub("^.+/_+.$", "", fnFs) |> head()
```

```
``
```

```
``{r}
```

```
create a directory for the outputs
```

```
quality_folder <- here::here("outputs",
 "dada2",
 "quality_plots")
```

```
if (!dir.exists(quality_folder)) {
 dir.create(quality_folder, recursive = TRUE)
}
```

```
...
```

```
``{r}
```

```
path_to_trimmed_reads <- here::here(
 "outputs",
 "dada2",
 "trimmed"
)
```

```
if (!dir.exists(path_to_trimmed_reads)) dir.create(path_to_trimmed_reads, recursive = TRUE)
...
```

```
``{r}
```

```
primer_fwd <- "CCTACGGGNBGCASCAG"
primer_rev <- "GACTACNVGGGTATCTAAT"
...
```

```
``{r}
```



```
Biostrings::readDNASTringSet(
 fnFs[1],
 format = "fastq",
 nrec = 10
)
...
```

```
``{r}
```

```
Biostrings::readDNASTringSet(
 fnRs[1],
 format = "fastq",
 nrec = 10
)
...
```

```
``{bash}
```

```
pwd
cp -R /home/rstudio/ADM2023_tutoriel/course-material-main/bash .
...
```

```
``{r}
```

```
nopFw <- sort(list.files(path_to_trimmed_reads, pattern = "R1", full.names = TRUE))
nopRv <- sort(list.files(path_to_trimmed_reads, pattern = "R2", full.names = TRUE))
print(nopRv)
print(nopFw)
...
```

```
``{r}
```

```
nopFw <- sort(list.files(path_to_trimmed_reads, pattern = "R1", full.names = TRUE))
nopRv <- sort(list.files(path_to_trimmed_reads, pattern = "R2", full.names = TRUE))
```

```

print(nopRv)
print(nopFw)
...

```{r}

path_to_filtered_reads <- here::here("outputs", "dada2", "filtered")
if (!dir.exists(path_to_filtered_reads)) dir.create(path_to_filtered_reads, recursive = TRUE)
...


```{r}

filtFs <- file.path(path_to_filtered_reads, basename(fnFs))
filtRs <- file.path(path_to_filtered_reads, basename(fnRs))
...


```{r}

names(filtFs) <- sample_names
names(filtRs) <- sample_names
...


```{r}

(out <- dada2::filterAndTrim(
 fwd = nopFw,
 filt = filtFs,
 rev = nopRv,
 filt.rev = filtRs,
 minLen = 150,
 matchIDs = TRUE,
 maxN = 0,
 maxEE = c(3, 3),

```

```

 truncQ = 2
))
 ...

  ```{r}
  errF <- dada2::learnErrors(filtFs,
    randomize = TRUE,
    multithread = TRUE)
  ...

  ```{r}
 errR <- dada2::learnErrors(filtRs,
 randomize = TRUE,
 multithread = TRUE)
 ...

  ```{r}
  dada2::plotErrors(errF, nominalQ=TRUE)
  ...

  ```{r}
 derepFs <- dada2::derepFastq(filtFs, verbose = TRUE)

 derepRs <- dada2::derepFastq(filtRs, verbose = TRUE)
 ...

  ```{r}

```

```
dadaFs <- dada2::dada(derepFs, err = errF, multithread = TRUE)
```

```
...
```

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

```
dadaRs <- dada2::dada(derepRs, err = errR, multithread = TRUE)
```

```
...
```

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

```
mergers <- dada2::mergePairs(
```

```
  dadaF = dadaFs,
```

```
  derepF = derepFs,
```

```
  dadaR = dadaRs,
```

```
  derepR = derepRs,
```

```
  maxMismatch = 0,
```

```
  verbose = TRUE
```

```
)
```

```
...
```

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

```
seqtab <- dada2::makeSequenceTable(mergers)
```

```
...
```

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

```
seqtab_nochim <- dada2::removeBimeraDenovo(seqtab,
```

```
  method = "consensus",
```

```
  multithread = TRUE,
```

```
  verbose = TRUE)
```

```
...
```

```

```{r}
export_folder <- here::here("outputs", "dada2", "asv_table")

if (!dir.exists(export_folder)) dir.create(export_folder, recursive = TRUE)

saveRDS(object = seqtab_nochim,
 file = file.path(export_folder, "seqtab_nochim.rds"))

...

```{r}
asv_seq <- colnames(seqtab_nochim)
...

```{r}
ndigits <- nchar(length(asv_seq))
asv_id <- sprintf(paste0("ASV_%0", ndigits, "d"), seq_along(asv_seq))
...

...

```{r}

seqtab_nochim_export <- t(seqtab_nochim)

...

```{r}

```

```
cat(paste0(">", names(asv_seq), "\n", asv_seq),
 sep = "\n",
 file = file.path(export_folder, "asv.fasta"))
...
```