

# ADM\_2024\_plastique

2024-01-11

```
library(dada2)
```

```
## Le chargement a nécessité le package : Rcpp
```

```
## Warning: le package 'Rcpp' a été compilé avec la version R 4.3.2
```

```
path_to_fastqs <- "/Users/Anna/OneDrive/Cours/Master/M1/Semestre 8/stage/DADA2/dezippes"  
# définir le chemin pour aller aux Fastas
```

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

```
## [1] "ERR4008961_1.fastq" "ERR4008961_2.fastq" "ERR4008963_1.fastq"  
## [4] "ERR4008963_2.fastq" "ERR4008967_1.fastq" "ERR4008967_2.fastq"  
## [7] "ERR4008970_1.fastq" "ERR4008970_2.fastq" "filtered"
```

```
fnFs <- sort(list.files(path_to_fastqs,  
                        pattern = "_1.fastq",  
                        full.names = TRUE))
```

```
fnRs <- sort(list.files(path_to_fastqs,  
                        pattern = "_2.fastq",  
                        full.names = TRUE))
```

```
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  
sample.names
```

```
## [1] "ERR4008961" "ERR4008963" "ERR4008967" "ERR4008970"
```

```
basename(fnFs) |>  
  strsplit(split = "_") |> # séparer les noms des fichiers en deux vecteurs  
  head()
```

```
## [[1]]  
## [1] "ERR4008961" "1.fastq"  
##  
## [[2]]  
## [1] "ERR4008963" "1.fastq"  
##
```

```
## [[3]]
## [1] "ERR4008967" "1.fastq"
##
## [[4]]
## [1] "ERR4008970" "1.fastq"
```

```
filtFs <- file.path(path_to_fastqs, "filtered", paste0(sample.names, "_1_filt.fastq"))
filtRs <- file.path(path_to_fastqs, "filtered", paste0(sample.names, "_2_filt.fastq"))
names(filtFs) <- sample.names
names(filtRs) <- sample.names
```

```
out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs, trimLeft = c(20,20), truncLen=c(240,200),
                    maxN=0, maxEE = c(2,2),truncQ=2, rm.phix=TRUE,
                    compress=FALSE, multithread=FALSE)
head(out)
```

```
##                reads.in reads.out
## ERR4008961_1.fastq  132705   118355
## ERR4008963_1.fastq   97235    81283
## ERR4008967_1.fastq   75637    61389
## ERR4008970_1.fastq  114463   101583
```

```
errF <- learnErrors(filtFs, multithread=FALSE)
```

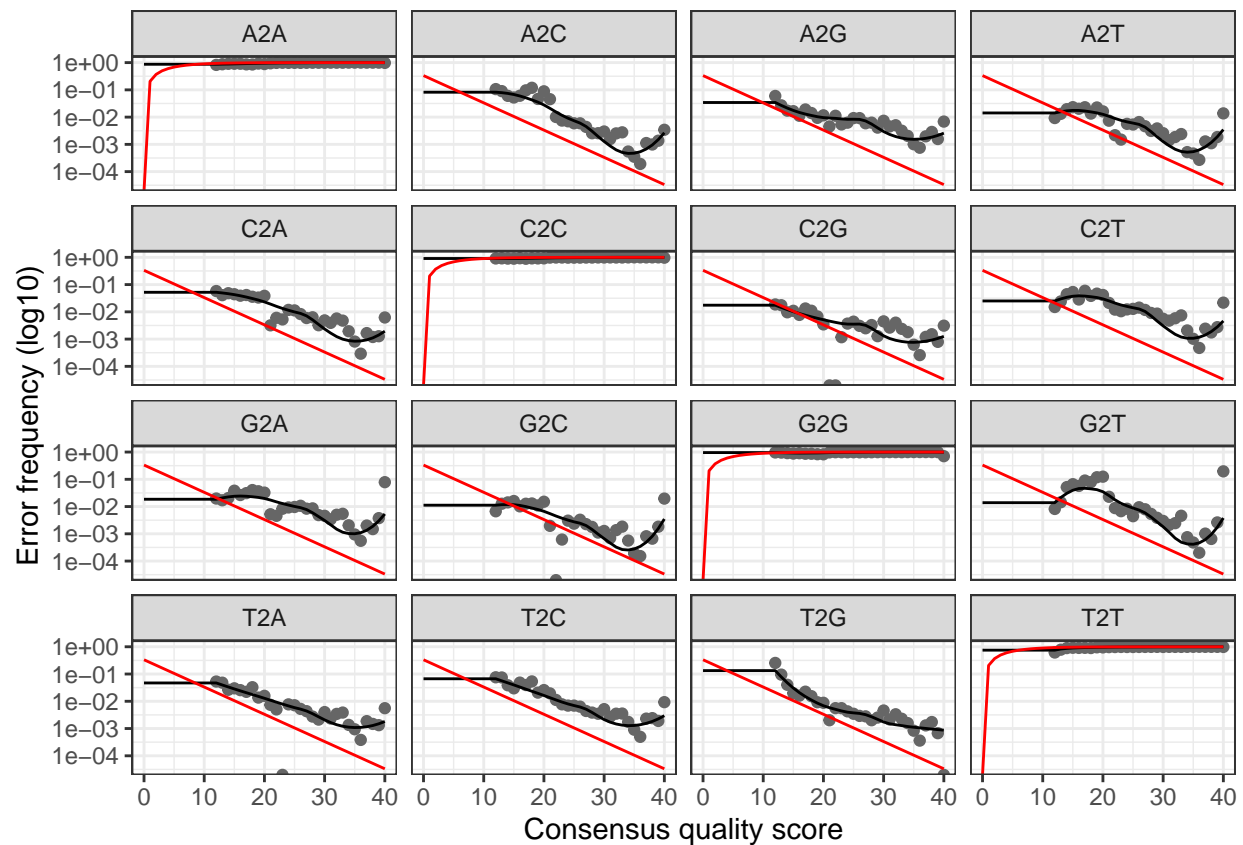
## 79774200 total bases in 362610 reads from 4 samples will be used for learning the error rates.

```
errR <- learnErrors(filtRs, multithread = FALSE)
```

## 65269800 total bases in 362610 reads from 4 samples will be used for learning the error rates.

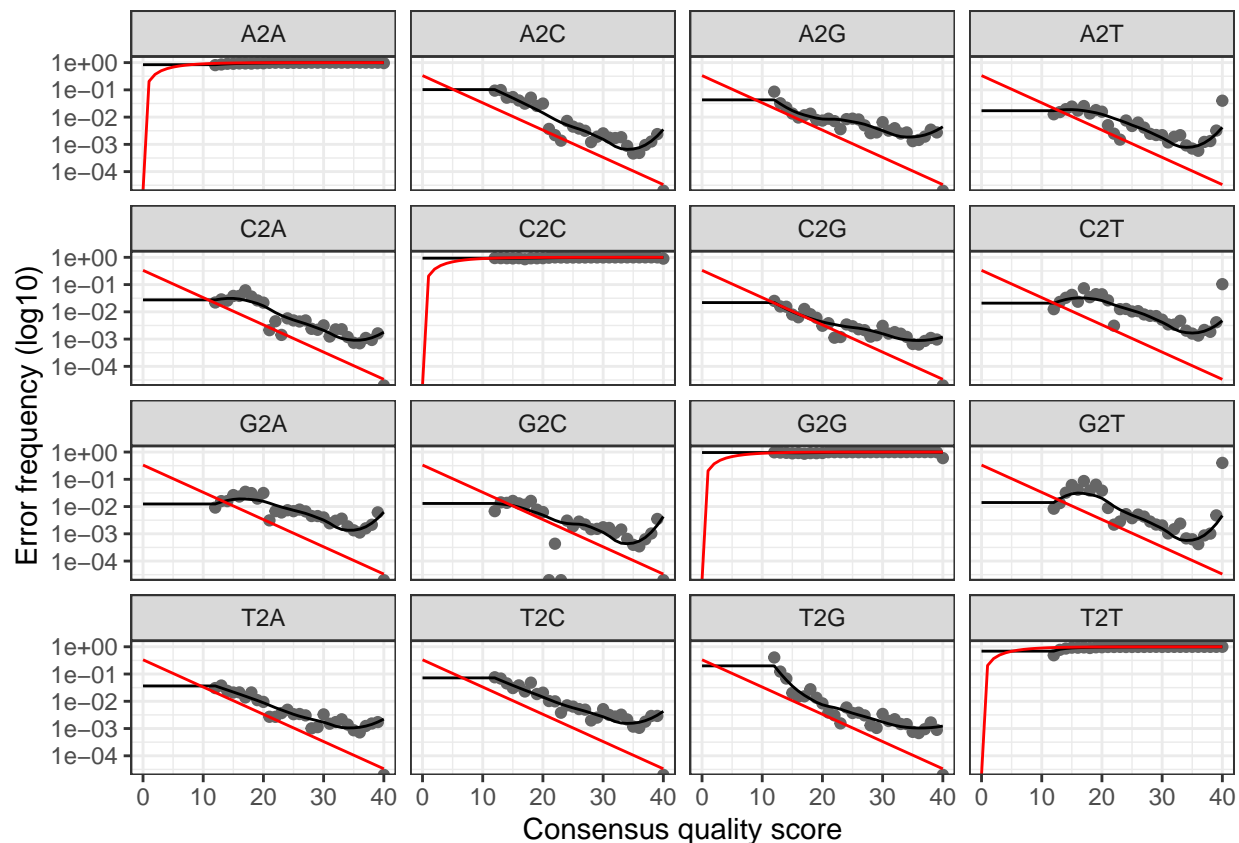
```
plotErrors(errF, nominalQ=TRUE)
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis
```



```
plotErrors(errR, nominalQ=TRUE)
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis
```



```
dadaFs <- dada(filtFs, err=errF, multithread=FALSE)
```

```
## Sample 1 - 118355 reads in 21700 unique sequences.
## Sample 2 - 81283 reads in 15103 unique sequences.
## Sample 3 - 61389 reads in 11563 unique sequences.
## Sample 4 - 101583 reads in 17475 unique sequences.
```

```
dadaRs <- dada(filtRs, err=errR, multithread=FALSE)
```

```
## Sample 1 - 118355 reads in 24967 unique sequences.
## Sample 2 - 81283 reads in 19709 unique sequences.
## Sample 3 - 61389 reads in 15395 unique sequences.
## Sample 4 - 101583 reads in 22124 unique sequences.
```

```
dadaFs[[1]]
```

```
## dada-class: object describing DADA2 denoising results
## 2284 sequence variants were inferred from 21700 input unique sequences.
## Key parameters: OMEGA_A = 1e-40, OMEGA_C = 1e-40, BAND_SIZE = 16
```

```
mergers <- mergePairs(dadaFs, filtFs, dadaRs, filtRs, verbose=TRUE)
```

```
## 106951 paired-reads (in 2709 unique pairings) successfully merged out of 113328 (in 5374 pairings) in
```

```
## 75152 paired-reads (in 1806 unique pairings) successfully merged out of 78369 (in 2913 pairings) input
## 55626 paired-reads (in 1559 unique pairings) successfully merged out of 59118 (in 3141 pairings) input
## 94264 paired-reads (in 1878 unique pairings) successfully merged out of 97544 (in 3808 pairings) input
```

```
head(mergers[[1]])
```

```
##
## 1 TAACCTTGC GGCCGTACTCCCCAGGCGGTCAACTTATTGCGTTAGCTGCGCCACTAAATCCTCAAGGAATCCAACGGCTAGTTGACATCGTTACGG
## 2 ACGGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCGCTAGGCGGTTTGTTAAGCGAGATGTGAAAGCCCAGGGCTCAACCTTGAAT
## 3 TAATCTTGCGACCGTACTCCCCAGGCGGAGAGCTTAATGCGTTAACTGCGCCACCGAACAGTATACTGTCCGACGGCTAGCTCTCATCGTTACGG
## 4 ACGGAGGGGGCAAGCGTTGTTTCGGAATTACTGGGCGTAAAGCGCACGTAGGCGGACTGATCAGTTGGGGGTGAAATCCCGAGGCTTAACCTCGGA
## 5 TAGTCTTGCGACCGTACTTCCCAGGCGGTACACTTATCGCGTTAGCTTAGACACCGAAGGGGCTAATCCCCCGACATCGAGTGACAACGTTACGG
## 6 TAATCTTGCGACCGTACTCCCCAGGCGGTCAACTTATCGCGTTAGCTGCGCCACTAAGTCCTTAAATGAACCCAACGGCTAGTTGACATCGTTAGG
## abundance forward reverse nmatch nmismatch nindel prefer accept
## 1 9438 1 1 28 0 0 2 TRUE
## 2 8897 2 2 28 0 0 1 TRUE
## 3 6381 3 3 29 0 0 2 TRUE
## 4 5291 4 4 29 0 0 1 TRUE
## 5 1883 6 5 27 0 0 2 TRUE
## 6 1823 5 6 27 0 0 1 TRUE
```

```
seqtab <- makeSequenceTable(mergers)
dim(seqtab)
```

```
## [1] 4 6460
```

```
table(nchar(getSequences(seqtab)))
```

```
##
## 220 222 223 224 225 226 230 231 232 234 235 236 237 240 241 242
## 37 3 4 3 1 1 2 1 1 1 1 1 1 7 2 2
## 243 244 245 247 248 249 250 251 253 254 257 259 261 262 263 264
## 3 1 2 2 118 3 2 1 3 2 2 2 2 1 1 1
## 265 267 269 271 273 277 280 284 287 289 295 300 301 302 303 305
## 1 2 2 1 1 1 2 2 1 3 3 3 1 2 2 3
## 306 313 314 315 316 317 318 319 321 322 324 325 327 329 331 335
## 1 2 2 3 2 1 2 1 3 1 3 2 1 3 1 1
## 336 337 338 339 340 341 343 344 345 346 347 348 349 350 351 353
## 2 2 2 1 2 4 2 1 2 3 4 1 1 2 1 2
## 354 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372
## 3 14 3 3 13 11 8 9 45 17 104 82 605 281 1481 1116
## 373 374 375 376 377 378 379 380 381 382 383 385 386
## 1644 507 90 32 27 9 12 9 10 7 5 7 1
```

```
seqtab.nochim <- removeBimeraDenovo(seqtab, method="consensus", multithread=FALSE, verbose=TRUE)
```

```
## Identified 1408 bimeras out of 6460 input sequences.
```

```
dim(seqtab.nochim)
```

```
## [1] 4 5052
```

```
getN <- function(x) sum(getUniques(x))
track <- cbind(out, sapply(dadaFs, getN), sapply(dadaRs, getN), sapply(mergers, getN), rowSums(seqtab.nochim))
# If processing a single sample, remove the sapply calls: e.g. replace sapply(dadaFs, getN) with getN(dadaFs)
colnames(track) <- c("input", "filtered", "denoisedF", "denoisedR", "merged", "nonchim")
rownames(track) <- sample.names
head(track)
```

```
##           input filtered denoisedF denoisedR merged nonchim
## ERR4008961 132705   118355   115192   115469 106951 101516
## ERR4008963  97235    81283    79450    79509  75152  74463
## ERR4008967  75637    61389    59984    60003  55626  55172
## ERR4008970 114463   101583    99088    99163  94264  92746
```

```
taxa <- assignTaxonomy(seqtab.nochim, "C:/Users/Anna/OneDrive/Cours/Master/M1/Semestre 8/stage/DADA2/silva_species_assignment_v4.11.fasta")
```

```
taxa <- addSpecies(taxa, "C:/Users/Anna/OneDrive/Cours/Master/M1/Semestre 8/stage/DADA2/silva_species_assignment_v4.11.fasta")
```

```
taxa.print <- taxa # Removing sequence rownames for display only
rownames(taxa.print) <- NULL
head(taxa.print)
```

```
##      Kingdom      Phylum      Class      Order
## [1,] "Bacteria" NA          NA          NA
## [2,] "Bacteria" NA          NA          NA
## [3,] "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Pseudomonadales"
## [4,] "Bacteria" "Proteobacteria" "Alphaproteobacteria" "Rhizobiales"
## [5,] "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Pseudomonadales"
## [6,] "Bacteria" "Proteobacteria" NA          NA
##      Family      Genus      Species
## [1,] NA          NA          NA
## [2,] NA          NA          NA
## [3,] "Cellvibrionaceae" "Eionea" NA
## [4,] "Methylobacteriaceae" NA      NA
## [5,] "Cellvibrionaceae" "Agaribacter" NA
## [6,] NA          NA          NA
```

```
export_folder <- ("export")
```

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

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

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

```
write.csv(taxa.print, "tableASVfinale.csv")
```