x Tutorial

Daniel Vaulot

24 01 2018

Contents

1 Aim

This tutorial explain how to process Illumina data with the Dada2 suite as implemented in R (dada2 is also implemented in Qiime). It is adapted from : https://benjjneb.github.io/dada2/tutorial.html

2 Directory structure

- **/fastq_carbom** : fastq files from the carbom cruise
- **/databases** : PR2 database files (see Prerequisite belwo)
- **/dada2** : This tutorial for Illumina files

3 Downloads

Install the following software :

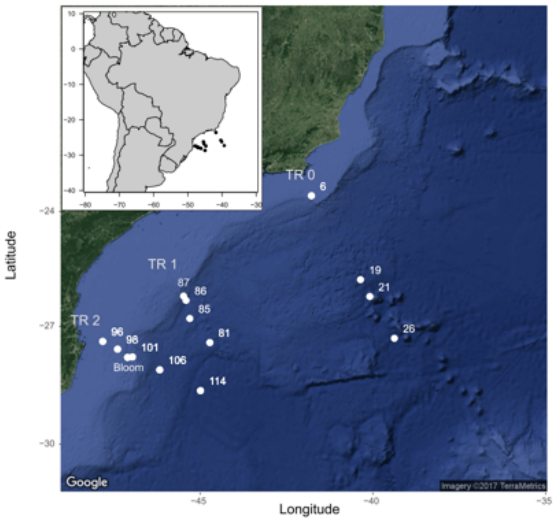
- R : https://pbil.univ-lyon1.fr/CRAN/
- R studio : https://www.rstudio.com/products/rstudio/download/#download
- Download and install the following libraries by running under R studio the following lines

```
install.packages("dplyr")      # To manipulate dataframes
install.packages("stringr")    # To strings
install.packages("ggplot2")    # To do plots
install.packages("readxl")     # To read excel files
install.packages("tibble")     # To work with data frames
install.packages("tidyr")     # To work with data frames

source("https://bioconductor.org/biocLite.R")
biocLite('dada2')              # metabarcode data analysis
biocLite('phyloseq')           # metabarcode data analysis
biocLite('Biostrings')         # needed for fastq.geometry
```

Download and install in the **/databases directory**

- PR2 database formatted for dada2 : https://github.com/vaulot/pr2_database/releases/

4 Data used



The samples originate from the CARBOM cruise (2013) off Brazil.

Samples have been sorted by flow cytometry and 3 genes have been PCR amplified :

- 18S rRNA - V4 region
- 16S rNA with plastid
- nifH

The PCR products have been sequenced by 1 run of Illumina 2*250 bp. The data consist of the picoplankton samples from one transect and fastq files have been subsampled with 1000 sequences per sample.

4.1 References

- Gérikas Ribeiro C, Marie D, Lopes dos Santos A, Pereira Brandini F, Vaulot D. (2016). Estimating microbial populations by flow cytometry: Comparison between instruments. Limnol Oceanogr Methods 14:750–758.
- Gérikas Ribeiro C, Lopes dos Santos A, Marie D, Brandini P, Vaulot D. (2018). Relationships between photosynthetic eukaryotes and nitrogen-fixing cyanobacteria off Brazil. ISME J in press.
- Gérikas Ribeiro C, Lopes dos Santos A, Marie D, Helena Pellizari V, Pereira Brandini F, Vaulot D. (2016). Pico and nanoplankton abundance and carbon stocks along the Brazilian Bight. PeerJ 4:e2587.

5 Tutorial description

5.1 Load the necessary libraries**

```
library("dada2")
library("phyloseq")
library("Biostrings")

library("ggplot2")
library("stringr")
library("dplyr")
library("tidyr")
```

```
  library("readxl")
  library("tibble")

  library("kableExtra") # necessary for nice table formatting with knitr
```

5.2 Set up directories

```
# change the following line to the path where you unzipped the tutorials
  tutorial_dir <- "C:/Users/vaulot/Google Drive/Scripts/"

# working directory in R_dada2
  working_dir <- paste0( tutorial_dir, "metabarcodes_tutorials/R_dada2")
  setwd(working_dir)

# ngs directory
  ngs_dir <- paste0( tutorial_dir, "metabarcodes_tutorials/fastq_carbom")
```

5.3 Primers

Note that the primers are degenerated. Dada2 has an option to remove primers (FilterandTrim) but this function will not accept degeneracy.

```
  primer_set_fwd = c("CCAGCAGCCGCGGTAATTCC", "CCAGCACCCGCGGTAATTCC",
                     "CCAGCAGCTGCGGTAATTCC", "CCAGCACCTGCGGTAATTCC")
  primer_set_rev = c("ACTTTCGTTCTTGATYRATGA")
  primer_length_fwd <- str_length(primer_set_fwd[1])
  primer_length_rev <- str_length(primer_set_rev[1])
```

5.4 PR2 tax levels

```
  PR2_tax_levels <- c("Kingdom", "Supergroup","Division", "Class",
                      "Order", "Family", "Genus", "Species")
```

5.5 Examine the fastQ files

5.5.1 Construct a list of the fastq files and extract the sample names (start of file name separated by _)

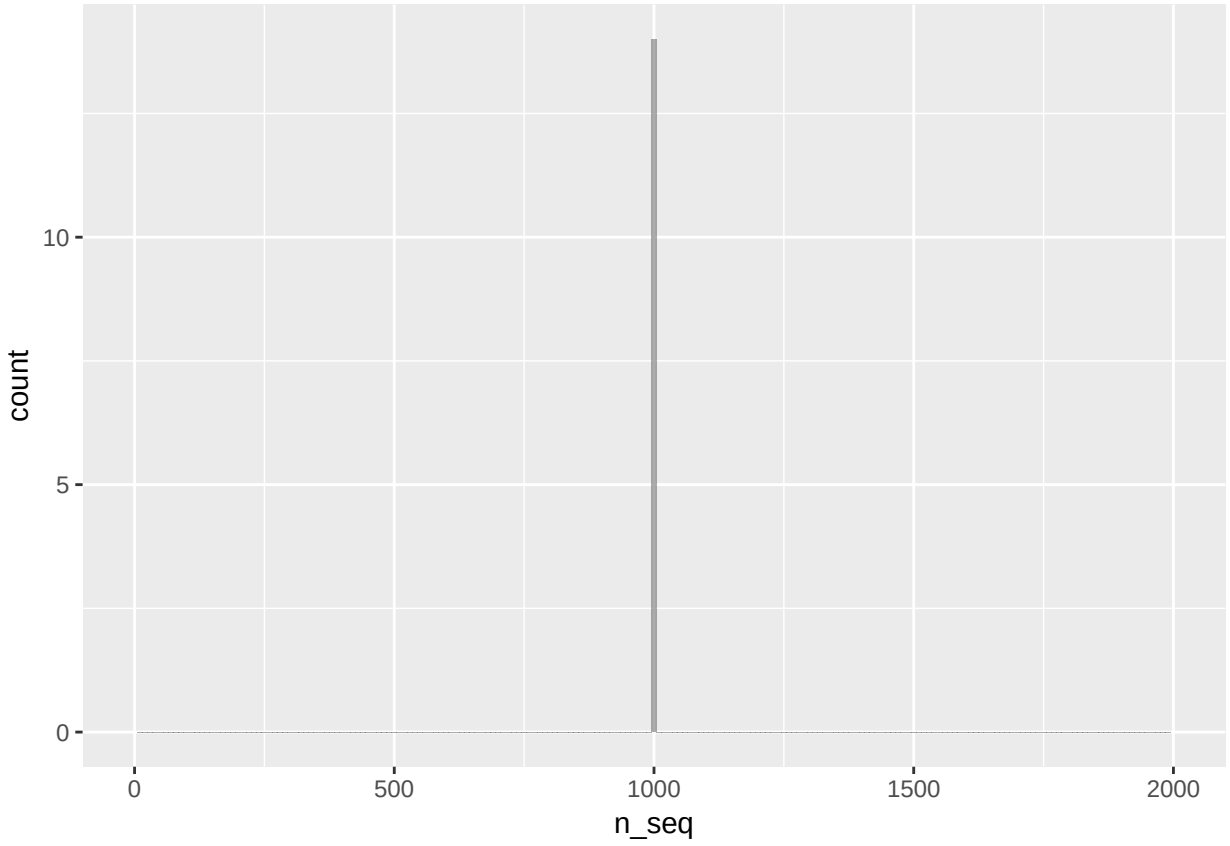
```
# get a list of all fastq files in the ngs directory and separate R1 and R2
  fns <- sort(list.files(ngs_dir, full.names = TRUE))
  fns <- fns[str_detect( basename(fns),".fastq")]
  fns_R1 <- fns[str_detect( basename(fns),"R1")]
  fns_R2 <- fns[str_detect( basename(fns),"R2")]

# Extract sample names, assuming filenames have format: SAMPLENAME_XXX.fastq
  sample.names <- str_split(basename(fns_R1), pattern = "_", simplify = TRUE)
  sample.names <- sample.names[,1]
```

5.5.2 Compute number of paired reads

```
# create an empty data frame
  df <- data.frame()

# loop throuh all the R1 files (no need to go through R2 which should be the same)

  for(i in 1:length(fns_R1)) {

    # use the dada2 function fastq.geometry
      geom <- fastq.geometry(fns_R1[i])

    # extract the information on number of sequences and file name
      df_one_row <- data.frame (n_seq=geom[1], file_name=basename(fns[i]) )

    # add one line to data frame
      df <- bind_rows(df, df_one_row)
 }
# display number of sequences and write data to small file
  knitr::kable(df)
```

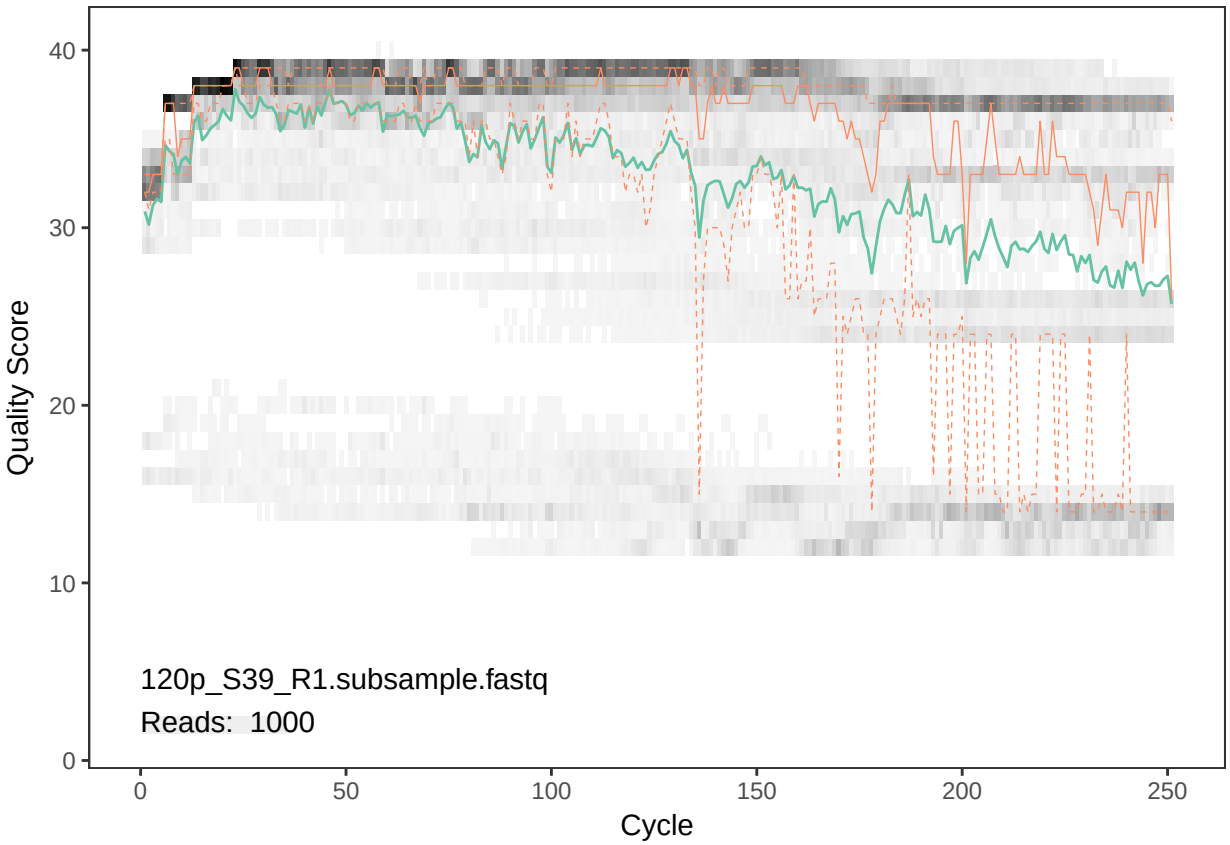
n_seq	file_name
1000	120p_S39_R1.subsample.fastq
1000	120p_S39_R2.subsample.fastq
1000	121p_S57_R1.subsample.fastq
1000	121p_S57_R2.subsample.fastq
1000	122p_S4_R1.subsample.fastq
1000	122p_S4_R2.subsample.fastq
1000	125p_S22_R1.subsample.fastq
1000	125p_S22_R2.subsample.fastq
1000	126p_S40_R1.subsample.fastq
1000	126p_S40_R2.subsample.fastq
1000	140p_S5_R1.subsample.fastq
1000	140p_S5_R2.subsample.fastq
1000	141p_S23_R1.subsample.fastq
1000	141p_S23_R2.subsample.fastq

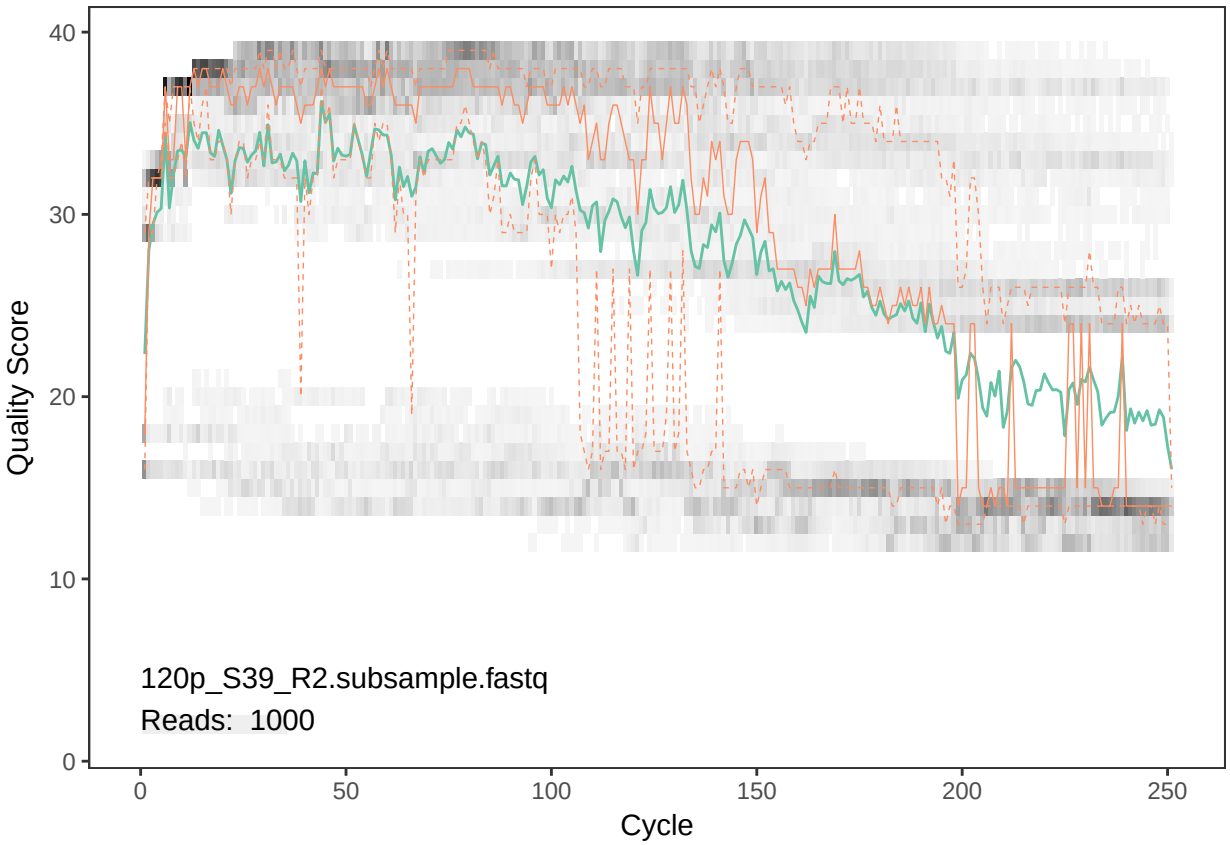
```
  # write.table(df, file = paste0(working_dir,"/n_seq.txt"),
  #             sep="\t", row.names = FALSE, na="", quote=FALSE)

# plot the histogram with number of sequences
  ggplot(df, aes(x=n_seq)) +
        geom_histogram( alpha = 0.5, position="identity", binwidth = 10) +
        xlim(0, 2000)
```



5.5.3 Plot quality fo reads

```
for(i in 1:length(fns)) {

 # Use dada2 function to plot quality
   p1 <- plotQualityProfile(fns[i])

 # Only plot on screen for first 2 files
   if (i <= 2) {print(p1)}

 # save the file as a pdf file
   p1_file <- paste0(ngs_dir,"/qual/",basename(fns[i]),".pdf")
   ggsave( plot=p1, filename= p1_file,
             device = "pdf", width = 15, height = 15, scale=1, units="cm")
 }
```


Quality Score
40
30
20
10
0
0
50
100
150
200
250
Cycle
120p_S39_R1.subsample.fastq
Reads: 1000



5.6 Filter and Trim the reads

5.6.1 Create names for the filtered files in filtered/ subdirectory of the fastq carbom

```
filt_dir <- file.path(ngs_dir, "filtered")

filt_R1 <- file.path(filt_dir, paste0(sample.names, "_R1_filt.fastq"))
filt_R2 <- file.path(filt_dir, paste0(sample.names, "_R2_filt.fastq"))
```

5.6.2 Removing the primers by sequence

The dada2 algorithm requires primers to be removed prior to processing.

The next piece of code could be used to remove the primers by **sequence**. The dada2 package does not allow for primer degeneracy. Since our forward primer is degenerated at two positions, all four combinations need to be tested. However it will be necessary to re-assemble after that the 4 fastQ files created (which has not to done). So the better strategy is to remove primer by trunction (see next step).

```
# On Windows set multithread=FALSE

out_all <-data.frame(id=length(fns_R1))
for (i in 1:4) {
  out <- filterAndTrim(fns_R1, filt_R1, fns_R2, filt_R2, truncLen=c(250,240), trimLeft = c(0,0),
            maxN=0, maxEE=c(Inf, Inf), truncQ=10, rm.phix=TRUE, primer.fwd = primer_set_fwd[i],
```

```
                compress=FALSE, multithread=FALSE)
    out_all <- cbind(out_all, out)

  }
```

```
  knitr::kable(out_all)
```

	id	reads.in	reads.out	reads.in	reads.out	reads.in	reads.out	reads.in	reads.ou
120p_S39_R1.subsample.fastq	14	1000	204	1000	171	1000	47	1000	3
121p_S57_R1.subsample.fastq	14	1000	294	1000	205	1000	64	1000	7
122p_S4_R1.subsample.fastq	14	1000	275	1000	191	1000	49	1000	4
125p_S22_R1.subsample.fastq	14	1000	306	1000	224	1000	81	1000	6
126p_S40_R1.subsample.fastq	14	1000	360	1000	241	1000	90	1000	7
140p_S5_R1.subsample.fastq	14	1000	276	1000	217	1000	57	1000	5
141p_S23_R1.subsample.fastq	14	1000	305	1000	221	1000	86	1000	5
142p_S41_R1.subsample.fastq	14	1000	307	1000	245	1000	96	1000	8
155p_S59_R1.subsample.fastq	14	1000	200	1000	166	1000	42	1000	1
156p_S6_R1.subsample.fastq	14	1000	284	1000	224	1000	82	1000	6
157p_S24_R1.subsample.fastq	14	1000	302	1000	248	1000	83	1000	5
165p_S42_R1.subsample.fastq	14	1000	246	1000	172	1000	92	1000	3
166p_S60_R1.subsample.fastq	14	1000	254	1000	206	1000	78	1000	7
167p_S7_R1.subsample.fastq	14	1000	263	1000	214	1000	69	1000	5

The table shows the number of sequences recognized by each primer combination.

5.6.3 Remove primers by truncation and filter

Filter all sequences with N, truncate R2 to 240 bp

```
  out <- filterAndTrim(fns_R1, filt_R1, fns_R2, filt_R2,
                       truncLen=c(250,240), trimLeft = c(primer_length_fwd,primer_length_rev),
                       maxN=0, maxEE=c(2, 2), truncQ=10, rm.phix=TRUE,
                       compress=FALSE, multithread=FALSE)
```

5.7 Dada2 processing

5.7.1 Learn error rates

The error rates are plotted.

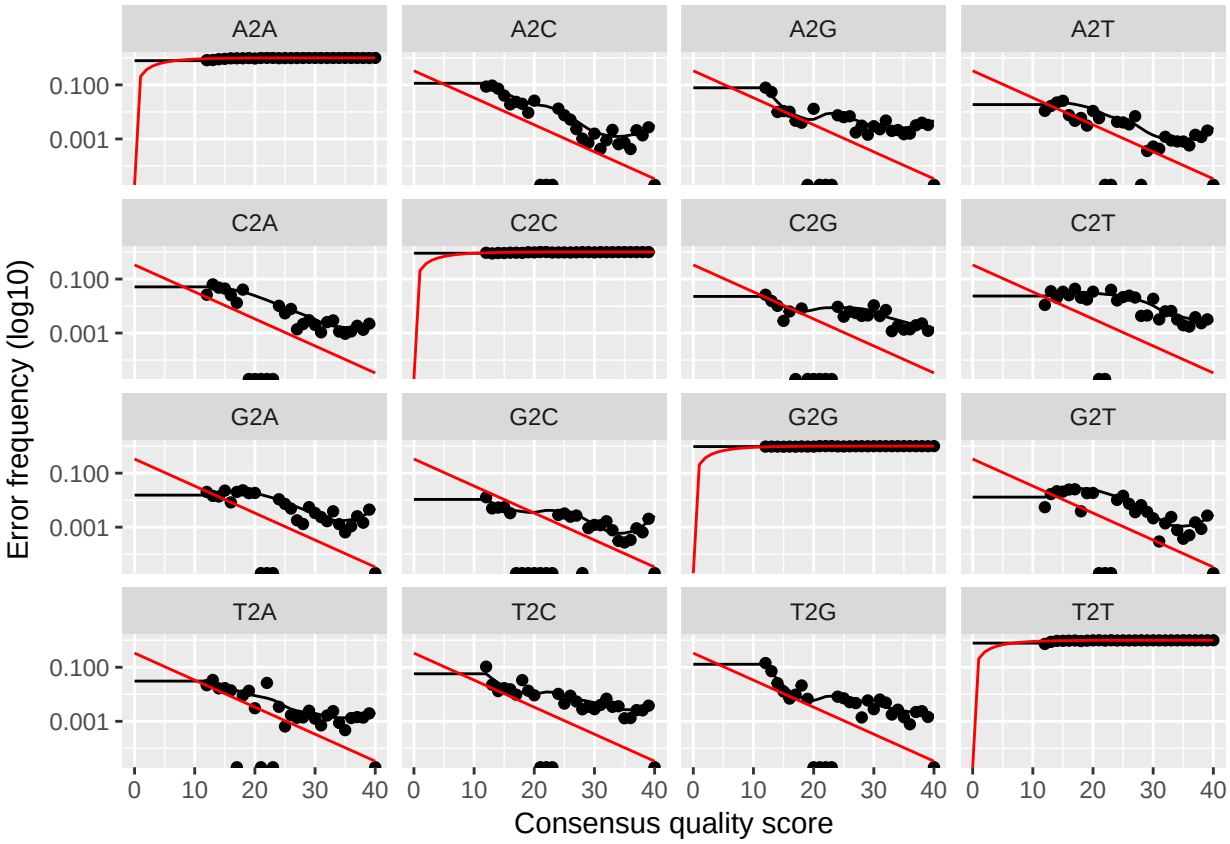
```
  err_R1 <- learnErrors(filt_R1, multithread=FALSE)
```

```
## Initializing error rates to maximum possible estimate.
## Sample 1 - 256 reads in 93 unique sequences.
## Sample 2 - 457 reads in 166 unique sequences.
## Sample 3 - 407 reads in 128 unique sequences.
## Sample 4 - 553 reads in 220 unique sequences.
## Sample 5 - 508 reads in 219 unique sequences.
## Sample 6 - 456 reads in 147 unique sequences.
## Sample 7 - 473 reads in 180 unique sequences.
## Sample 8 - 583 reads in 211 unique sequences.
## Sample 9 - 528 reads in 172 unique sequences.
```

```
## Sample 10 - 530 reads in 211 unique sequences.
## Sample 11 - 513 reads in 177 unique sequences.
## Sample 12 - 521 reads in 199 unique sequences.
## Sample 13 - 519 reads in 172 unique sequences.
## Sample 14 - 572 reads in 170 unique sequences.
##    selfConsist step 2
##    selfConsist step 3
## Convergence after  3  rounds.
## Total reads used:  6876
```

```
plotErrors(err_R1, nominalQ=TRUE)
```

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

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



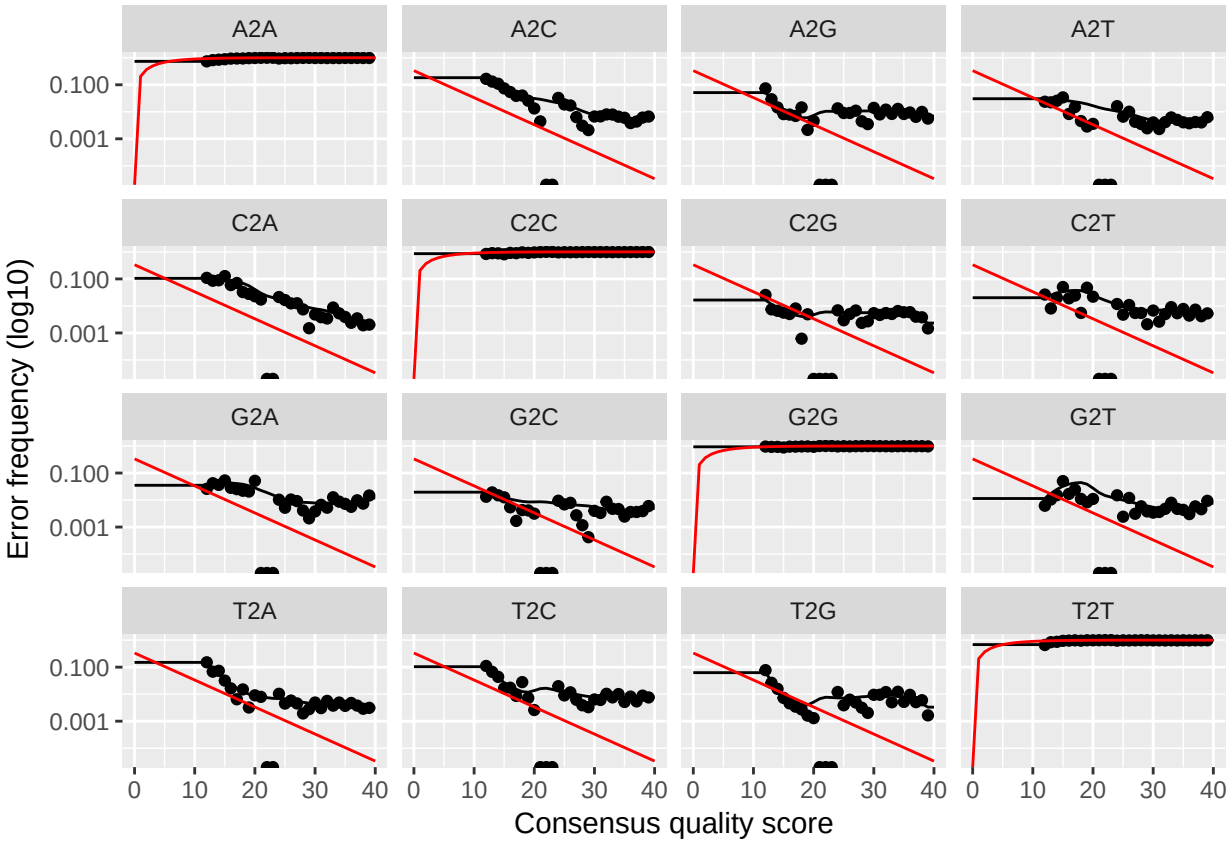

```
err_R2 <- learnErrors(filt_R2, multithread=FALSE)
```

```
## Initializing error rates to maximum possible estimate.
## Sample 1 - 256 reads in 236 unique sequences.
## Sample 2 - 457 reads in 391 unique sequences.
## Sample 3 - 407 reads in 299 unique sequences.
## Sample 4 - 553 reads in 409 unique sequences.
## Sample 5 - 508 reads in 451 unique sequences.
## Sample 6 - 456 reads in 335 unique sequences.
## Sample 7 - 473 reads in 347 unique sequences.
## Sample 8 - 583 reads in 458 unique sequences.
## Sample 9 - 528 reads in 418 unique sequences.
```

```
## Sample 10 - 530 reads in 404 unique sequences.
## Sample 11 - 513 reads in 384 unique sequences.
## Sample 12 - 521 reads in 395 unique sequences.
## Sample 13 - 519 reads in 396 unique sequences.
## Sample 14 - 572 reads in 400 unique sequences.
##    selfConsist step 2
##    selfConsist step 3
##    selfConsist step 4
## Convergence after  4  rounds.
## Total reads used:  6876
```

```
  plotErrors(err_R2, nominalQ=TRUE)
```

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

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



5.7.2 Dereplicate the reads

```
  derep_R1 <- derepFastq(filt_R1, verbose=FALSE)
  derep_R2 <- derepFastq(filt_R2, verbose=FALSE)

# Name the derep-class objects by the sample names
  names(derep_R1) <- sample.names
  names(derep_R2) <- sample.names
```

5.7.3 Sequence-variant inference algorithm to the dereplicated data

```
    dada_R1 <- dada(derep_R1, err=err_R1, multithread=FALSE)
```

```
## Sample 1 - 256 reads in 93 unique sequences.
## Sample 2 - 457 reads in 166 unique sequences.
## Sample 3 - 407 reads in 128 unique sequences.
## Sample 4 - 553 reads in 220 unique sequences.
## Sample 5 - 508 reads in 219 unique sequences.
## Sample 6 - 456 reads in 147 unique sequences.
## Sample 7 - 473 reads in 180 unique sequences.
## Sample 8 - 583 reads in 211 unique sequences.
## Sample 9 - 528 reads in 172 unique sequences.
## Sample 10 - 530 reads in 211 unique sequences.
## Sample 11 - 513 reads in 177 unique sequences.
## Sample 12 - 521 reads in 199 unique sequences.
## Sample 13 - 519 reads in 172 unique sequences.
## Sample 14 - 572 reads in 170 unique sequences.
```

```
    dada_R2 <- dada(derep_R2, err=err_R2, multithread=FALSE)
```

```
## Sample 1 - 256 reads in 236 unique sequences.
## Sample 2 - 457 reads in 391 unique sequences.
## Sample 3 - 407 reads in 299 unique sequences.
## Sample 4 - 553 reads in 409 unique sequences.
## Sample 5 - 508 reads in 451 unique sequences.
## Sample 6 - 456 reads in 335 unique sequences.
## Sample 7 - 473 reads in 347 unique sequences.
## Sample 8 - 583 reads in 458 unique sequences.
## Sample 9 - 528 reads in 418 unique sequences.
## Sample 10 - 530 reads in 404 unique sequences.
## Sample 11 - 513 reads in 384 unique sequences.
## Sample 12 - 521 reads in 395 unique sequences.
## Sample 13 - 519 reads in 396 unique sequences.
## Sample 14 - 572 reads in 400 unique sequences.
```

```
    dada_R1[[1]]
```

```
## dada-class: object describing DADA2 denoising results
## 5 sample sequences were inferred from 93 input unique sequences.
## Key parameters: OMEGA_A = 1e-40, BAND_SIZE = 16, USE_QUALS = TRUE
```

```
    dada_R2[[1]]
```

```
## dada-class: object describing DADA2 denoising results
## 3 sample sequences were inferred from 236 input unique sequences.
## Key parameters: OMEGA_A = 1e-40, BAND_SIZE = 16, USE_QUALS = TRUE
```

5.7.4 Merge sequences

```
    mergers <- mergePairs(dada_R1, derep_R1, dada_R2, derep_R2, verbose=TRUE)
```

```
## 237 paired-reads (in 3 unique pairings) successfully merged out of 256 (in 6 pairings) input.
## 404 paired-reads (in 7 unique pairings) successfully merged out of 457 (in 16 pairings) input.
```

```
## 368 paired-reads (in 5 unique pairings) successfully merged out of 407 (in 10 pairings) input.
## 466 paired-reads (in 11 unique pairings) successfully merged out of 553 (in 25 pairings) input.
## 343 paired-reads (in 7 unique pairings) successfully merged out of 508 (in 19 pairings) input.
## 441 paired-reads (in 4 unique pairings) successfully merged out of 456 (in 6 pairings) input.
## 396 paired-reads (in 6 unique pairings) successfully merged out of 473 (in 12 pairings) input.
## 508 paired-reads (in 7 unique pairings) successfully merged out of 583 (in 18 pairings) input.
## 449 paired-reads (in 4 unique pairings) successfully merged out of 528 (in 7 pairings) input.
## 429 paired-reads (in 9 unique pairings) successfully merged out of 530 (in 13 pairings) input.
## 439 paired-reads (in 9 unique pairings) successfully merged out of 513 (in 19 pairings) input.
## 449 paired-reads (in 7 unique pairings) successfully merged out of 521 (in 12 pairings) input.
## 487 paired-reads (in 5 unique pairings) successfully merged out of 519 (in 10 pairings) input.
## 563 paired-reads (in 9 unique pairings) successfully merged out of 572 (in 14 pairings) input.
```

```
  # Inspect the merger data.frame from the first sample
    knitr::kable(head(mergers[[1]]) )
```

sequence
AGCTCCAATAGCGTATATTAAAGTTGTTGCAGTTAAAACGCTCGTAGTCGGATTTCGGGGCAGGTTTGCCGGTC
CACACGTCTAATGTTGCATTGTAAAGCACAAACCACTGTTTTACATTTAGCTGCAGAAAGAGGAACTGTAGAAG
AGCTCCAATAGCGTATACTAAAGTTGTTGCAGTTAAAAAGCTCGTAGTTGGATTTCTGGTCGAAGCAGCCGCTC

5.7.5 Make sequence table

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

```
## The sequences being tabled vary in length.
```

```
    dim(seqtab)
```

```
## [1] 14 58
```

```
  # Make a transposed of the seqtab to make it be similar to mothur database
    t_seqtab <- t(seqtab)

  # Inspect distribution of sequence lengths
    table(nchar(getSequences(seqtab)))
```

```
##
## 318 351 360 363 367 369 371 372 373 375 376 377 378 379 380 382 383 384
##   3   1   1   3   1   1   2   1   2   1   3   6  12   4   4   2   5   2
## 387 388
##   1   3
```

5.7.6 Remove chimeras

Note that remove chimeras will produce spurious results if primers have not be removed. The parameter `methods` can be `pooled` or `consensus`

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

```
## Identified 0 bimeras out of 58 input sequences.
```

```
  # Compute % of non chimeras
    paste0("% of non chimeras : ",sum(seqtab.nochim)/sum(seqtab))
```

```
## [1] "% of non chimeras : 1"
```

```
    paste0("total number of sequences : ",sum(seqtab.nochim))
```

```
## [1] "total number of sequences : 5979"
```

In our case there were no chimeras found. It is noteworthy that the total number of sequences is almost twice that what is recovered with **mothur** which is **2573**

5.7.7 Track number of reads at each step

```
  # define a function
    getN <- function(x) sum(getUniques(x))

    track <- cbind(out, sapply(dada_R1, getN), sapply(mergers, getN), rowSums(seqtab), rowSums(seqtab.n

    colnames(track) <- c("input", "filtered", "denoised", "merged", "tabled", "nonchim")
    rownames(track) <- sample.names

    knitr::kable(track)
```

	input	filtered	denoised	merged	tabled	nonchim
120p	1000	256	256	237	237	237
121p	1000	457	457	404	404	404
122p	1000	407	407	368	368	368
125p	1000	553	553	466	466	466
126p	1000	508	508	343	343	343
140p	1000	456	456	441	441	441
141p	1000	473	473	396	396	396
142p	1000	583	583	508	508	508
155p	1000	528	528	449	449	449
156p	1000	530	530	429	429	429
157p	1000	513	513	439	439	439
165p	1000	521	521	449	449	449
166p	1000	519	519	487	487	487
167p	1000	572	572	563	563	563

5.7.8 Assigning taxonomy

```
    pr2_file <- paste0(tutorial_dir, "metabarcodes_tutorials/databases/pr2_version_4.72_dada2_Eukaryota
    taxa <- assignTaxonomy(seqtab.nochim, refFasta=pr2_file,
                           taxLevels = PR2_tax_levels,
                           minBoot = 0, outputBootstraps = TRUE,
                           verbose = TRUE)
```

```
## Finished processing reference fasta.
```

5.7.9 Export data as produced by Dada2

```
    write.table(taxa$tax, file = paste0(working_dir,"/taxa.txt"), sep="\t", row.names = TRUE, na="", qu
    write.table(taxa$boot, file = paste0(working_dir,"/taxa_boot.txt"), sep="\t", row.names = TRUE, na=
    write.table(seqtab.nochim, file = paste0(working_dir,"/seqtab.txt"), sep="\t", row.names = TRUE, na=
```

5.7.10 Changing OTU names from sequence to Otuxxxx

In the OTU put of dada2, otu names are the sequences. We change to give a Otuxxx name and the sequences are stored in the taxonomy table.

```
    taxa_tax <- as.data.frame(taxa$tax)
    taxa_boot <- as.data.frame(taxa$boot)

    taxa_tax <- taxa_tax %>%  rownames_to_column(var = "sequence") %>%
                              rowid_to_column(var = "OTUNumber") %>%
                              mutate(OTUNumber = sprintf("otu%04d", OTUNumber))
    row.names(taxa_tax) <- taxa_tax$OTUNumber
    row.names(taxa_boot) <- taxa_tax$OTUNumber

  # Transpose matrix of abundance
    seqtab.nochim_trans <- as.data.frame(t(seqtab.nochim))
    row.names(seqtab.nochim_trans) <- taxa_tax$OTUNumber
```

5.7.11 Filter for 18S

Remember that we sequenced 3 genes (18S, 16S plastid and nifH. We remove the sequences are not 18S by selecting only bootstrap value for Supergroup in excess of 80.

```
    bootstrap_min <- 80

  # Filter based on the bootstrap
    taxa_tax_18S <- taxa_tax[taxa_boot$Supergroup >= bootstrap_min,]
    taxa_boot_18S <- taxa_boot[taxa_boot$Supergroup >= bootstrap_min,]



  # Filter matrix of abundance by removing row for which Supergroup bootstrap < min
    seqtab.nochim_18S <- seqtab.nochim_trans[taxa_boot$Supergroup >= bootstrap_min,]

  # Create a database like file for dada2
    dada2_database <-   cbind(taxa_tax_18S, seqtab.nochim_18S)
```

5.8 Phyloseq

5.8.1 Create a phyloseq object for dada2

```
samdf <- data.frame(sample_name=sample.names)
rownames(samdf) <- sample.names

ps_dada2 <- phyloseq(otu_table(as.matrix(seqtab.nochim_18S), taxa_are_rows=TRUE),
```

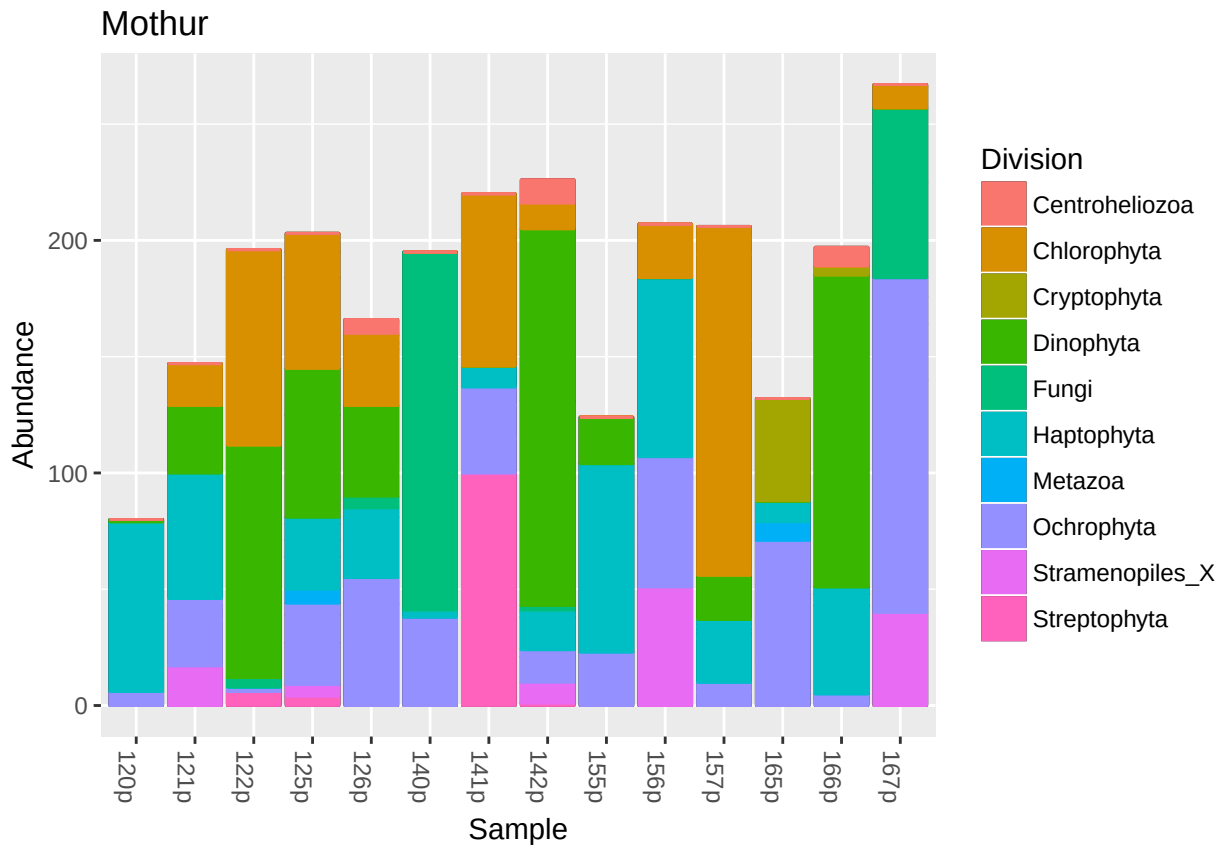
```
               sample_data(samdf),
               tax_table(as.matrix(select(taxa_tax_18S, - sequence, -OTUNumber))))
```

5.8.2 Create a phyloseq object for the mothur results

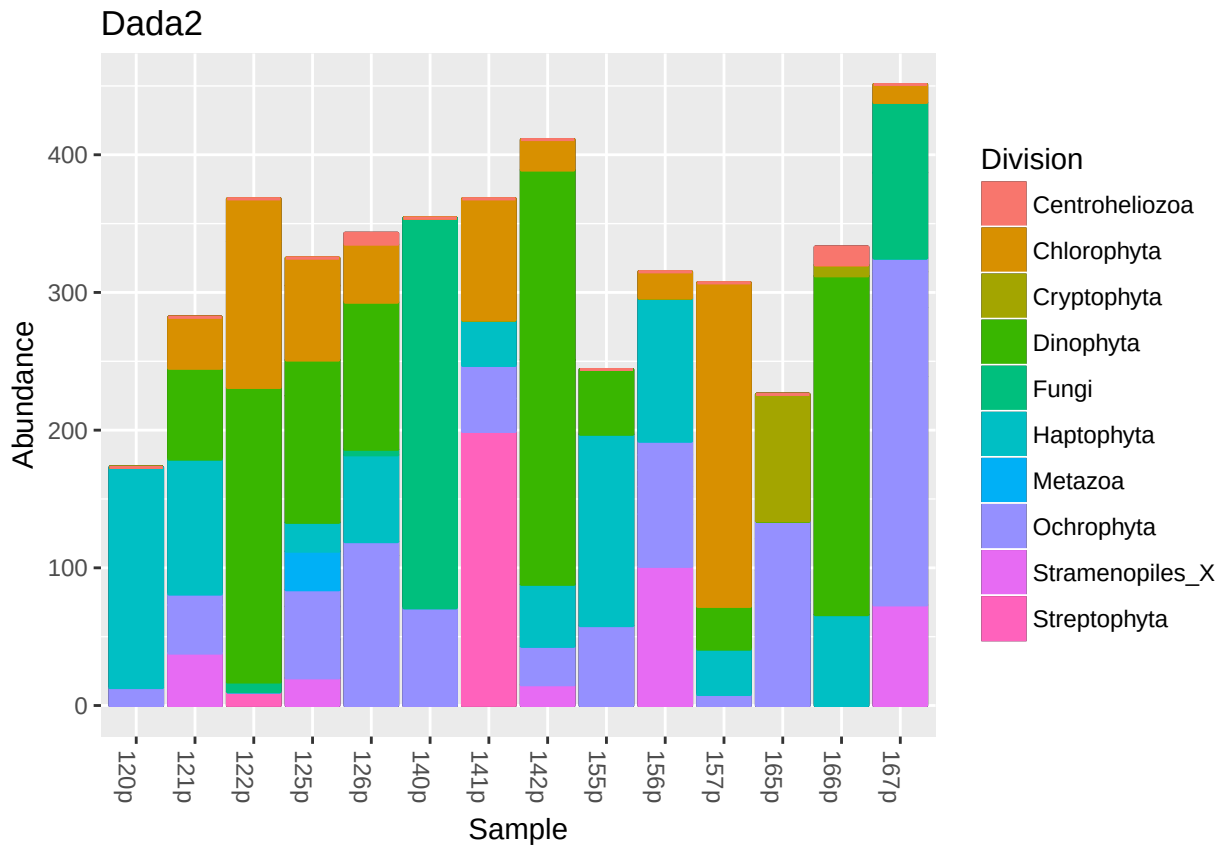
```
# read the mothur database from Excel file
  mothur_file <- paste0( tutorial_dir, "metabarcodes_tutorials/Comparison different methods 1.0.xlsx")
  mothur_database <- read_excel(mothur_file, sheet="mothur")
# need to remove empty rows
  mothur_database <- mothur_database %>% filter(!is.na(OTUNumber))

# create the taxonomy matrix
  mothur_tax <- str_split(mothur_database$OTUConTaxonomy, ";", 9, simplify = TRUE)
# the last column is empty so remove
  mothur_tax <- mothur_tax[,1:8 ]

# give row names and column names
  colnames(mothur_tax) <- PR2_tax_levels
  rownames(mothur_tax) <- mothur_database$OTUNumber

# replace in the database, the unique taxonomy column by 8 columns
  mothur_database <- cbind(mothur_database, mothur_tax)
  mothur_database <- mothur_database %>% select(-OTUConTaxonomy)

# create the otu_table
  mothur_otu <- select(mothur_database, ends_with("p"),-Supergroup)
  rownames(mothur_otu) <- mothur_database$OTUNumber
  mothur_otu <- as.matrix(mothur_otu)

# create the phyloseq for mothur
  ps_mothur <- phyloseq(otu_table(mothur_otu, taxa_are_rows=TRUE),
               sample_data(samdf),
               tax_table(mothur_tax))
```

5.8.3 Compare at the division level

```
  # remove Lobosa only found in Mothur
  ps_mothur_common <- subset_taxa(ps_mothur, !(Division %in% c("Lobosa")))

  plot_bar(ps_mothur_common, fill = "Division") +
    geom_bar(aes(color=Division, fill=Division), stat="identity", position="stack") +
    ggtitle("Mothur")
```




```
plot_bar(ps_dada2, fill = "Division") +
  geom_bar(aes(color=Division, fill=Division), stat="identity", position="stack") +
  ggtitle("Dada2")
```



Compare by aggregation

5.8.4 Transform the database files into the long version

```
dada2_long <- dada2_database %>% gather(key = "sample", value="n_seq", num_range(120:200,"p"))  %>%
                                 select(-sequence)
knitr::kable(head(dada2_long), "latex") %>%
       kable_styling(bootstrap_options = "striped", font_size = 7)
```

OTUNumber	Kingdom	Supergroup	Division	Class	Order	Family	Genus
otu0002	Eukaryota	Hacrobia	Haptophyta	Prymnesiophyceae	Prymnesiophyceae_X	Braarudosphaeraceae	Braarudosph
otu0003	Eukaryota	Archaeplastida	Chlorophyta	Mamiellophyceae	Mamiellales	Bathycoccaceae	Bathycoccus
otu0004	Eukaryota	Opisthokonta	Fungi	Basidiomycota	Agaricomycotina	Agaricomycetes	Hyphodontia
otu0005	Eukaryota	Stramenopiles	Ochrophyta	Chrysophyceae	Chrysophyceae_X	Chrysophyceae_Clade-G	Chrysophyce
otu0006	Eukaryota	Alveolata	Dinophyta	Dinophyceae	Dinophyceae_X	Dinophyceae_XX	Gonyaulax
otu0007	Eukaryota	Alveolata	Dinophyta	Syndiniales	Dino-Group-III	Dino-Group-III_X	Dino-Group-

```
mothur_long <- mothur_database %>% gather(key = "sample", value="n_seq", num_range(120:200,"p"))  %>%
                                 select(-contains("repSeq"), -nseq)
knitr::kable(head(mothur_long), "latex")  %>%
       kable_styling(bootstrap_options = "striped", font_size = 7)
```

OTUNumber	Kingdom	Supergroup	Division	Class	Order	Family	Genus
Otu01	Eukaryota	Hacrobia	Haptophyta	Prymnesiophyceae	Prymnesiophyceae_X	Braarudosphaeraceae	Braarudosph
Otu02	Eukaryota	Archaeplastida	Chlorophyta	Mamiellophyceae	Mamiellales	Bathycoccaceae	Bathycoccus
Otu03	Eukaryota	Archaeplastida	Chlorophyta	Mamiellophyceae	Mamiellales	Bathycoccaceae	Ostreococcus
Otu04	Eukaryota	Stramenopiles	Ochrophyta	Chrysophyceae	Chrysophyceae_X	Chrysophyceae_Clade-G	Chrysophyce
Otu05	Eukaryota	Opisthokonta	Fungi	Basidiomycota	Agaricomycotina	Agaricomycetes	Hyphodontia
Otu06	Eukaryota	Alveolata	Dinophyta	Dinophyceae	Dinophyceae_X	Dinophyceae_XX	Gonyaulax

5.8.5 Aggregate by Division, Class, Genus, Species

```
dada2_species <- dada2_long %>% group_by(Division, Class, Genus, Species) %>%
                                summarize (n_seq = sum(n_seq))
knitr::kable(head(dada2_species))
```

Division	Class	Genus	Species	n_seq
Centroheliozoa	Centroheliozoa_X	Pterocystida_XX	Pterocystida_XX_sp.	21
Chlorophyta	Mamiellophyceae	Bathycoccus	Bathycoccus_prasinos	368
Chlorophyta	Mamiellophyceae	Micromonas	Micromonas_Clade-B..4	19
Chlorophyta	Mamiellophyceae	Micromonas	Micromonas_Clade-B.E.3	20
Chlorophyta	Mamiellophyceae	Ostreococcus	Ostreococcus_lucimarinus	37
Chlorophyta	Mamiellophyceae	Ostreococcus	Ostreococcus_tauri	223

```
mothur_species <- mothur_long %>% group_by(Division, Class, Genus, Species) %>%
                                  summarize (n_seq = sum(n_seq))
knitr::kable(head(mothur_species))
```

Division	Class	Genus	Species	n_seq
Centroheliozoa	Centroheliozoa_X	Pterocystida_XX	Pterocystida_XX_sp.	24
Chlorophyta	Mamiellophyceae	Bathycoccus	Bathycoccus_prasinos	249
Chlorophyta	Mamiellophyceae	Dolichomastix	Dolichomastix_tenuilepis	3
Chlorophyta	Mamiellophyceae	Micromonas	Micromonas_Clade-A.ABC.1-2	21
Chlorophyta	Mamiellophyceae	Micromonas	Micromonas_Clade-B..4	10
Chlorophyta	Mamiellophyceae	Micromonas	Micromonas_Clade-B.E.3	11

5.8.6 Merge the two lists to comute the relation between mothur and dada2 estimates

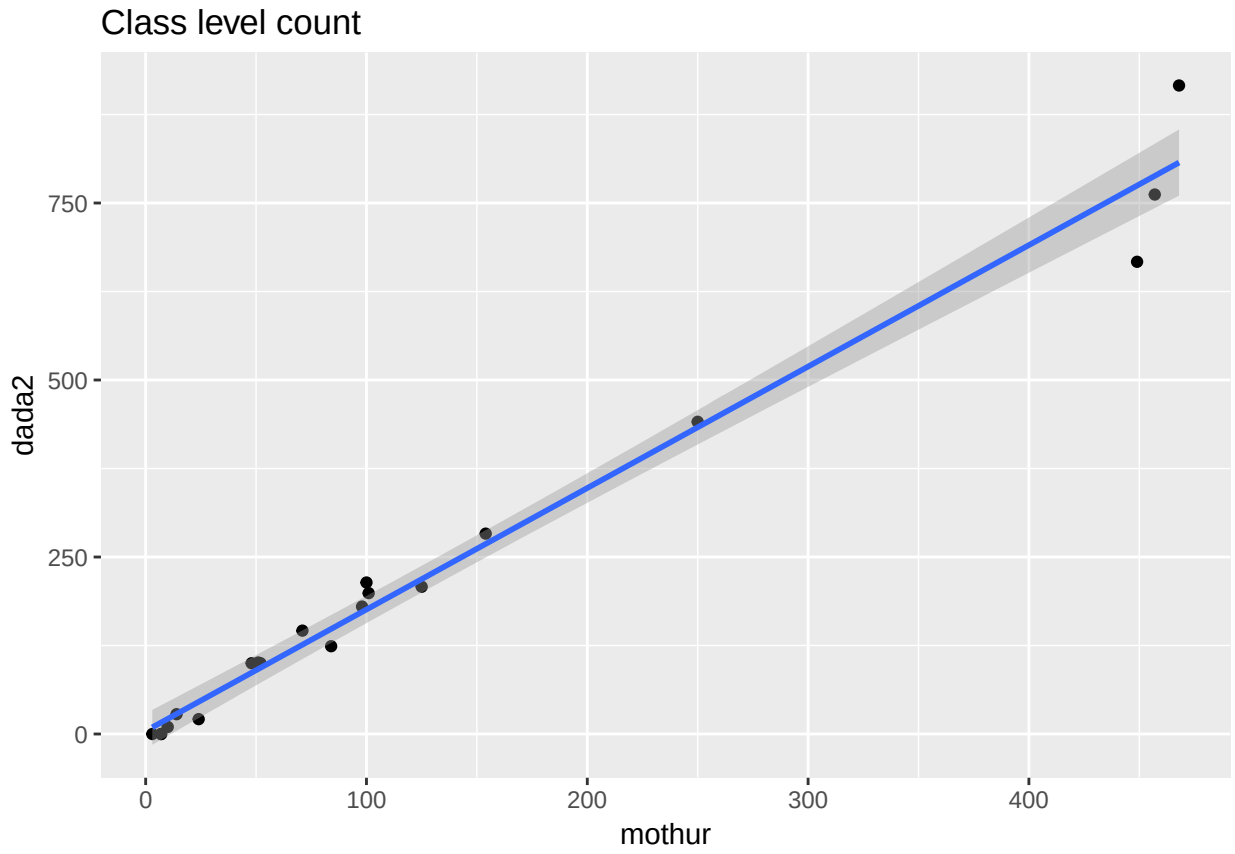
```
both_species <- rbind(dada2_species, mothur_species) %>%
                      group_by(Division, Class, Genus, Species) %>%
                      summarize (n_methods = n()) %>%
                      left_join(dada2_species) %>%
                      dplyr::rename (dada2 = n_seq) %>%
                      left_join(mothur_species) %>%
                      dplyr::rename (mothur = n_seq) %>%
                      arrange(desc(dada2))
```

```
## Joining, by = c("Division", "Class", "Genus", "Species")
## Joining, by = c("Division", "Class", "Genus", "Species")
```

```
both_class <- both_species %>%  group_by(Division, Class) %>%
                                summarise(dada2=sum(dada2, na.rm = TRUE), mothur=sum(mothur, na.rm = T
                                arrange(desc(dada2))
```

5.8.7 Plot at Class level

```
ggplot(both_class) + geom_point(aes(mothur,dada2)) +
                        geom_smooth(aes(mothur,dada2), method = "lm", show.legend = TRUE) +
                        ggtitle ("Class level count")
```




```
summary(lm(both_class$dada2 ~ both_class$mothur))
```

```
##
## Call:
## lm(formula = both_class$dada2 ~ both_class$mothur)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -107.598  -16.295    3.099   13.651  108.806
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         4.2860    11.8432   0.362    0.722
## both_class$mothur   1.7156     0.0599  28.641   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 40.22 on 18 degrees of freedom
```

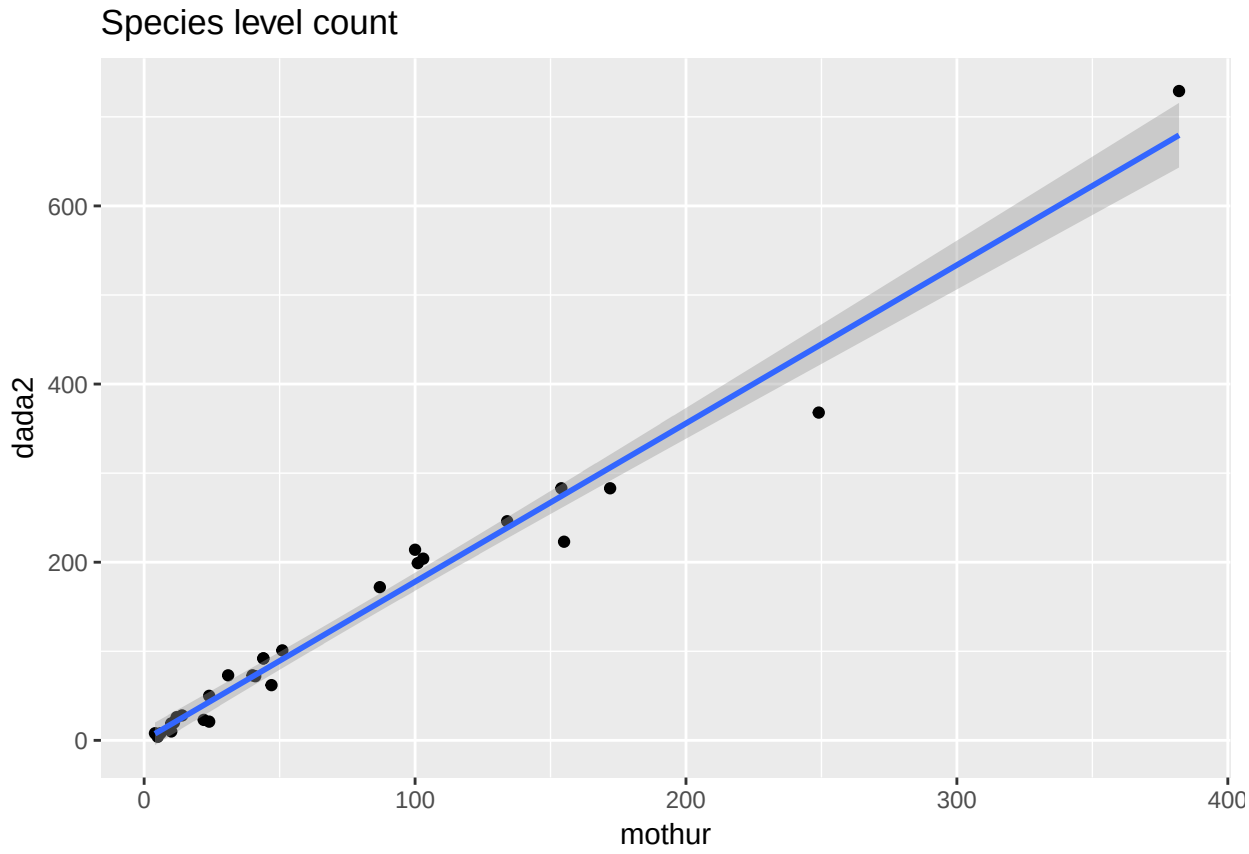
```
## Multiple R-squared:  0.9785, Adjusted R-squared:  0.9773
## F-statistic: 820.3 on 1 and 18 DF,  p-value: < 2.2e-16
```

5.8.8 Plot at Species level

```
ggplot(both_species) + geom_point(aes(mothur,dada2)) +
                       geom_smooth(aes(mothur,dada2), method = "lm", show.legend = TRUE) +
                       ggtitle ("Species level count")
```

```
## Warning: Removed 35 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 35 rows containing missing values (geom_point).
```




```
summary(lm(both_species$dada2 ~ both_species$mothur))
```

```
##
## Call:
## lm(formula = both_species$dada2 ~ both_species$mothur)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.001  -6.005   2.111  13.393  49.587
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.39515    6.23766   0.063     0.95
```

```
## both_species$mothur  1.77753    0.05525  32.172   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.88 on 26 degrees of freedom
##   (35 observations deleted due to missingness)
## Multiple R-squared:  0.9755, Adjusted R-squared:  0.9746
## F-statistic:  1035 on 1 and 26 DF,  p-value: < 2.2e-16
```

```
species_one_method <- both_species %>% filter(n_methods==1)
knitr::kable(species_one_method, "latex")%>%
       kable_styling(bootstrap_options = "striped", font_size = 7)
```

Division	Class	Genus	Species	n_methods	dada2	mothur
Dinophyta	Dinophyceae	Dinophyceae_XXX	Dinophyceae_XXX_sp.	1	241	NA
Ochrophyta	Chrysophyceae	Chrysophyceae_Clade-C_X	Chrysophyceae_Clade-C_X_sp.	1	109	NA
Dinophyta	Dinophyceae	Prorocentrum	Prorocentrum_shikokuense	1	107	NA
Ochrophyta	Bacillariophyta	Bacillaria	Bacillaria_paxillifer	1	89	NA
Fungi	Ascomycota	Pichia	Pichia_burtonii	1	51	NA
Ochrophyta	Bacillariophyta	Raphid-pennate_X	Raphid-pennate_X_sp.	1	47	NA
Chlorophyta	Mamiellophyceae	Ostreococcus	Ostreococcus_lucimarinus	1	37	NA
Haptophyta	Prymnesiophyceae	Oolithus	Oolithotus_fragilis	1	33	NA
Ochrophyta	Pelagophyceae	Sarcinochrysis	Sarcinochrysis_sp.	1	28	NA
Dinophyta	Dinophyceae	Gyrodinium	Gyrodinium_spirale	1	26	NA
Ochrophyta	Bacillariophyta	Nitzschia	Nitzschia_linearis	1	12	NA
Ochrophyta	Bacillariophyta	Cymbella	Cymbella_bruyanti	1	10	NA
Fungi	Ascomycota	Meyerozyma	Meyerozyma_guilliermondii	1	7	NA
Chlorophyta	Mamiellophyceae	Dolichomastix	Dolichomastix_tenuilepis	1	NA	3
Chlorophyta	Mamiellophyceae	Micromonas	Micromonas_Clade-A.ABC.1-2	1	NA	21
Chlorophyta	Prasino-Clade-VII	Prasino-Clade-VII-A-4	Prasino-Clade-VII-A-4_sp.	1	NA	7
Chlorophyta	Pyramimonadales	Halosphaera	Halosphaera_sp.	1	NA	3
Dinophyta	Dinophyceae	Dinophyceae_XX_unclassified	Dinophyceae_XX_unclassified	1	NA	109
Dinophyta	Dinophyceae	Gyrodinium	Gyrodinium_fusiforme	1	NA	20
Dinophyta	Dinophyceae	Prorocentrum	Prorocentrum_unclassified	1	NA	58
Fungi	Ascomycota	Saccharomycetales_unclassified	Saccharomycetales_unclassified	1	NA	32
Haptophyta	Prymnesiophyceae	Braarudosphaera	Braarudosphaera_bigelowii	1	NA	4
Haptophyta	Prymnesiophyceae	Calcidiscaceae_unclassified	Calcidiscaceae_unclassified	1	NA	5
Haptophyta	Prymnesiophyceae	Chrysochromulina	Chrysochromulina_sp.	1	NA	3
Haptophyta	Prymnesiophyceae	Noelaerhabdaceae_unclassified	Noelaerhabdaceae_unclassified	1	NA	9
Haptophyta	Prymnesiophyceae	Prymnesiaceae_unclassified	Prymnesiaceae_unclassified	1	NA	7
Haptophyta	Prymnesiophyceae	Prymnesiophyceae_Clade_B4_X	Prymnesiophyceae_Clade_B4_X_sp.	1	NA	5
Haptophyta	Prymnesiophyceae	Prymnesiophyceae_Clade_D_XX	Prymnesiophyceae_Clade_D_XX_sp.	1	NA	12
Haptophyta	Prymnesiophyceae	Prymnesiophyceae_unclassified	Prymnesiophyceae_unclassified	1	NA	30
Lobosa	Tubulinea	Hartmannella	Hartmannella_unclassified	1	NA	7
Ochrophyta	Bacillariophyta	Bacillariophyta_X_unclassified	Bacillariophyta_X_unclassified	1	NA	6
Ochrophyta	Bacillariophyta	Raphid-pennate_unclassified	Raphid-pennate_unclassified	1	NA	95
Ochrophyta	Chrysophyceae	Chrysophyceae_X_unclassified	Chrysophyceae_X_unclassified	1	NA	44
Ochrophyta	Dictyochophyceae	Dictyochales_X	Dictyochales_X_sp.	1	NA	5
Ochrophyta	Pelagophyceae	Pelagophyceae_unclassified	Pelagophyceae_unclassified	1	NA	11

6 Conclusion on the dada2 pipeline

- The dada2 pipeline yieds 1.7 more reads than mothur
- The number of reads at the species and class levels are correlated
- It is very fast, the longest step is the taxonomy assignement
- It offers the advantage of having everything performed under R.