Mothur Illumina Tutorial

Daniel Vaulot

17 janvier 2018

Contents

1 Aim of tutorial

This tutorial explain how to process Illumina sequences.

- The first part of the tutorial makes use of R to obtain information on the number and quality of sequences.
- The second part uses mothur to process the sequences and compute the final abundance table.

2 Directory structure

- **/fastq_carbom** : fastq files from the carbom cruise
- **/databases** : Silva alignement and PR2 database files (see Prerequisite above)
- **/mothur/illumina** : Tutorial for Illumina files (carbom cruise)
- **/mothur/454** : Tutorial with 454 files

3 Downloads

Install the following software :

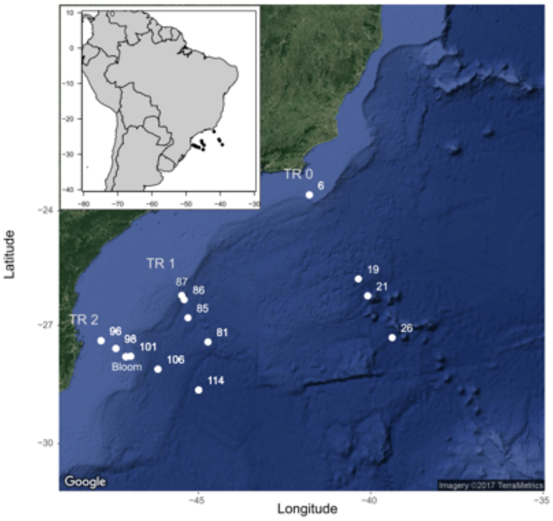
- Mothur : https://github.com/mothur/mothur/releases/tag/v1.39.5
- Terminal program. For Windows MobaXterm is highly recommended : https://mobaxterm.mobatek.net/
- R : https://pbil.univ-lyon1.fr/CRAN/
- R studio : https://www.rstudio.com/products/rstudio/download/#download
- Download and install the following libraries by running under R studio the following lines

```r
install.packages("dplyr")     # To manipulate dataframes
install.packages("stringr")   # To strings

install.packages("ggplot2")   # for high quality graphics

source("https://bioconductor.org/biocLite.R")
biocLite("Biostrings")        # manipulate sequences
biocLite('dada2')             # metabarcode data analysis
```

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

- PR2 database : https://github.com/vaulot/pr2_database/releases/download/4.7.2/pr2_version_4.7.2_mothur.zip .

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 Pre visualization of the fastq files with R

- Load the script file from **/mothur/illumina/R_analyze_fastq.R**

Load the necessary libraries

```
library("dada2")
library("Biostrings") # To manipulate DNA sequences

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

(1) **Set up the directories for the analysis**

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

# set up working directory
  working_dir <- paste0( tutorial_dir, "metabarcodes_tutorials/mothur/illumina")
  setwd(working_dir)

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

# 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")]
```

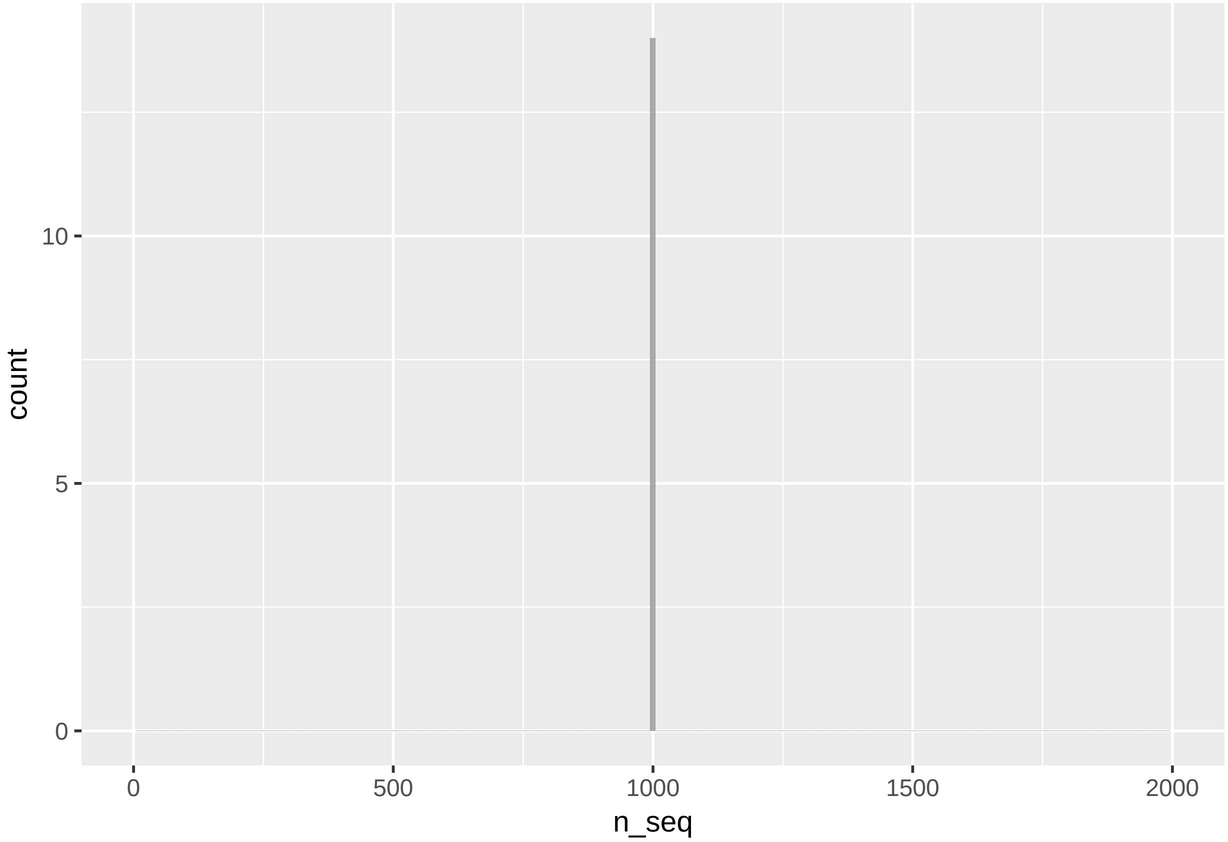
(2) **Compute number of paired reads in each fastq file**

Note that the data have been sub-sampled at 1000 reads per file.

```
# 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
  df
```

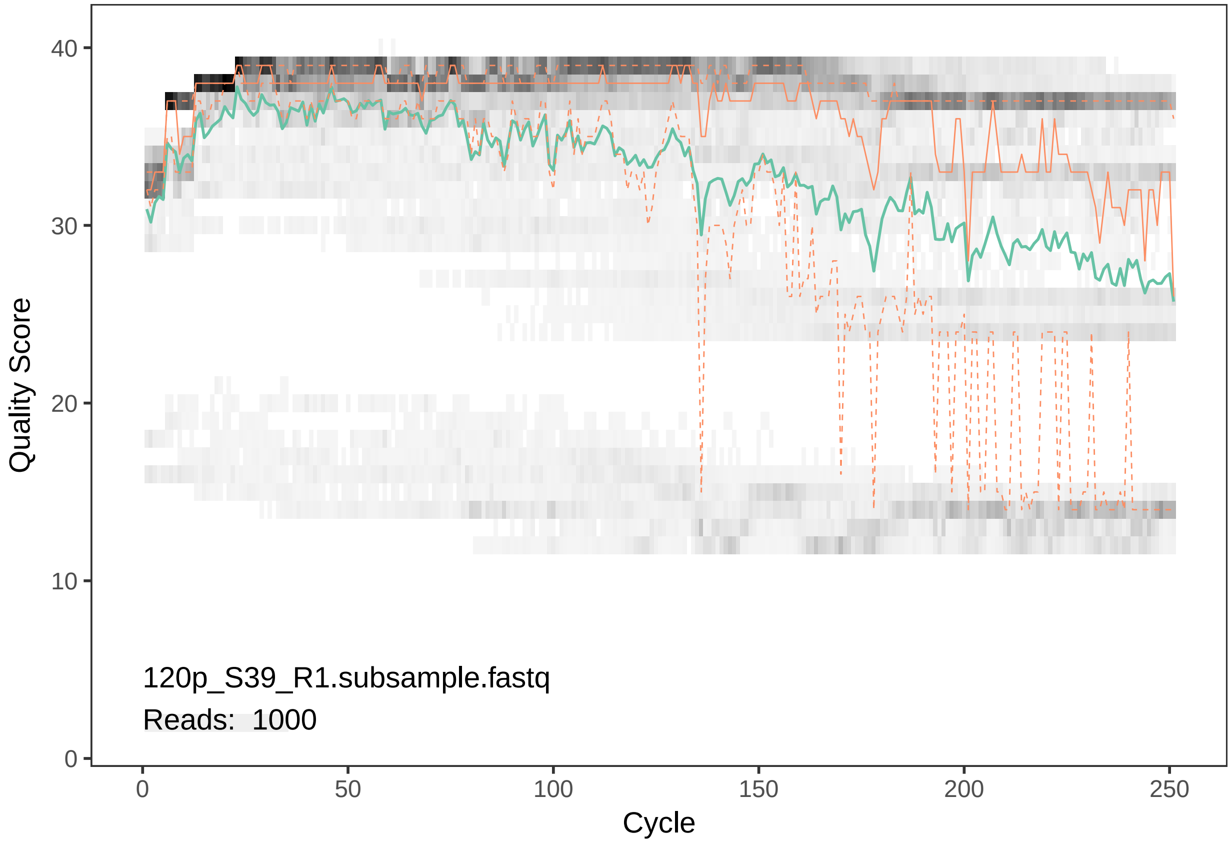
```
##    n_seq                    file_name
## 1   1000 120p_S39_R1.subsample.fastq
## 2   1000 120p_S39_R2.subsample.fastq
## 3   1000 121p_S57_R1.subsample.fastq
## 4   1000 121p_S57_R2.subsample.fastq
## 5   1000  122p_S4_R1.subsample.fastq
## 6   1000  122p_S4_R2.subsample.fastq
## 7   1000 125p_S22_R1.subsample.fastq
## 8   1000 125p_S22_R2.subsample.fastq
## 9   1000 126p_S40_R1.subsample.fastq
## 10  1000 126p_S40_R2.subsample.fastq
## 11  1000  140p_S5_R1.subsample.fastq
## 12  1000  140p_S5_R2.subsample.fastq
## 13  1000 141p_S23_R1.subsample.fastq
## 14  1000 141p_S23_R2.subsample.fastq
```

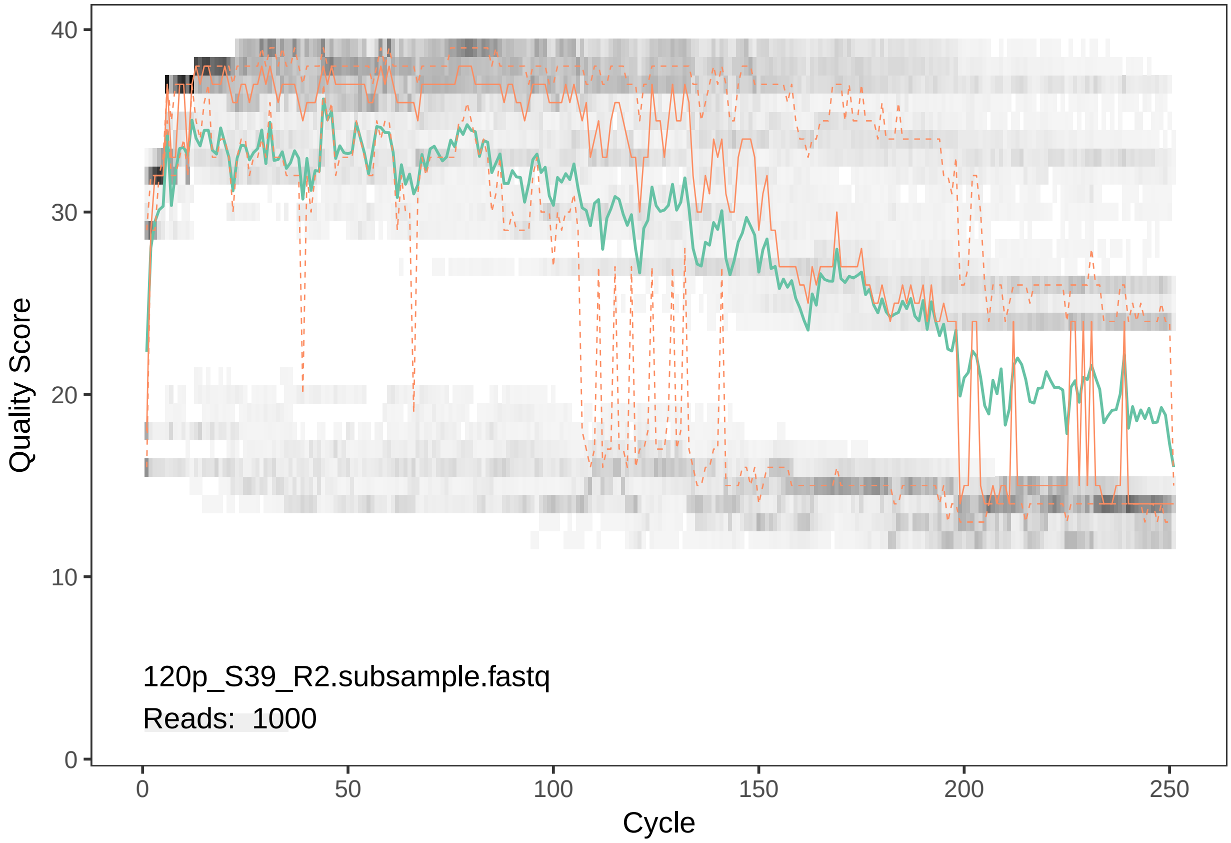
```
  write.table(df, file = paste0(working_dir,"/n_seq.txt"), sep="\t", row.names = FALSE, na="", quote=F

# 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)
```



(3) **Plot the quality fo each fastq file**

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

 for(i in 1:2) {

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

  # Only plot on screen for first 2 files
    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")
  }
```





(4) **Clean up memory**

It is necessary to clean up the memory because the fastq files are quite big and occupy a lot of memory during processing

```
rm(list=ls())
```

6 Analysis with mothur

Two files containing all the commands are provided
* mothur_carbom_linux.sh : use on a server (not tested on Mac) * mothur_carbom_windows.cmd : use on windows

Note that some of the steps have been removed for simplicity.

The major steps of the processing are :

- Build the contigs from the R1 and R2 reads
- Extract the sequences that contain the 2 primers
- Remove sequences in low abundance (singletons in particular)
- Align sequences to a reference alignment
- Remove chimeras
- Assign taxonomy based on PR2
- Compute sequence distance
- Cluster sequences at a given threhold (make OTUs)
- Create a final file with all the information

(1) **First define a few constants to make the script independant of the files**

```
# Change the DIR_DATA below to the path where you have downloaded the different files
DIR_DATA=".... /metabarcodes_tutorials/fastq_carbom"

FILE_PR2_TAX="../databases/pr2_version_4.72_mothur.tax"
FILE_PR2_FASTA="../databases/pr2_version_4.72_mothur.fasta"
FILE_SILVA="../databases/silva.seed_v123.euk.fasta"
FILE_PR2_END="72"

FILE_OLIGOS = "../databases/oligos18s_V4_Zingone.oligos"

MOTHUR="mothur"
PROJECT="carbom"
```

(2) **Change directory to where the fastq files are located**

```
cd $DIR_DATA
```

(3) **Make the contigs using the file $PROJECT.txt (= carbom.txt).**

This file has the following structure :

Sample	R1 file	R2 file
120p	120p_S39_R1.subsample.fastq	120p_S39_R2.subsample.fastq
121p	121p_S57_R1.subsample.fastq	121p_S57_R2.subsample.fastq
122p	122p_S4_R1.subsample.fastq	122p_S4_R2.subsample.fastq

```
$MOTHUR "#make.contigs(file=$PROJECT.txt, processors=32)"
```

(4) **Remove sequences that do not satisfy the following conditions:**

- Number of ambiguities = 0
- Minlength=350
- Maxlength=450

```
$MOTHUR "#screen.seqs(fasta=$PROJECT.trim.contigs.fasta,group=$PROJECT.contigs.groups,
                      maxambig=0,minlength=350, maxlength=450, processors=32)"
```

(5) Extract the sequences based on the presence of forward and reverse primers

- Mismatches allowed on the forward primer - pdiffs=2,
- Mismatches allowed on the reverse primer - rdiffs=2
- Oligo file : oligos18s_V4_Zingone.oligos

Keyword	Primer forward	Primer reverse	Name of primer
primer	CCAGCASCYGCGGTAATTCC	ACTTTCGTTCTTGATYRATGA	18S_V4_Zingone

```
$MOTHUR "#pcr.seqs(fasta=$PROJECT.trim.contigs.good.fasta,
                   group=$PROJECT.contigs.good.groups,
                   oligos=$FILE_OLIGOS,
                   pdiffs=2, rdiffs=2,
                   processors=32)"
```

(6) Shorten file names and indicate gene name

```
cp $PROJECT.trim.contigs.good.pcr.fasta $PROJECT_18S.fasta
cp $PROJECT.contigs.good.pcr.groups $PROJECT_18S.groups
```

(7) Dereplicate unique sequences

```
$MOTHUR "#unique.seqs(fasta=$PROJECT_18S.fasta)"
```

(8) Create a count file

This file create a table which as the following structure. For each unique sequence, it provides the total number of sequences and the number of sequences in each sample.

```
Representative_Sequence total   120p    121p    122p    125p    126p
M02439_22_000000000-ADOLA_1_1101_14247_1437 277 46  35  0   12  20
M02439_22_000000000-ADOLA_1_1101_12787_1647 2   2   0   0   0   0
M02439_22_000000000-ADOLA_1_1101_17899_1772 2   2   0   0   0   0
M02439_22_000000000-ADOLA_1_1101_13893_1778 1   1   0   0   0   0
```

This step saves disk space and speed up analysis

```
$MOTHUR "#count.seqs(name=$PROJECT_18S.names,
                     group=$PROJECT_18S.groups, processors=32)"
```

(9) Remove singletons

One can change the settings with the cutoff parameter.

```
$MOTHUR "#split.abund(count=$PROJECT_18S.count_table,
                      fasta=$PROJECT_18S.unique.fasta,
                      cutoff=1, accnos=true)"
```

(10) Align sequences to reference alignement

The file to be used can be downloaded from the mothur web site : https://www.mothur.org/w/images/a/a4/Silva.seed_v128.tgz. It is best to :

- extract only the eukaryotes using mothur command:`get.lineage(taxonomy=$SILVA.tax, taxon=Eukaryota, fasta=$SILVA.align)`

- remove all the gaps that are common to all sequences with mothur command `filter.seqs` (see next line)

```
$MOTHUR "#align.seqs(fasta=$PROJECT_18S.unique.abund.fasta,
                     reference=$FILE_SILVA,
                     flip=T, processors=32)"
```

(11) Remove all the gaps that are common to all sequences

```
$MOTHUR "#filter.seqs(fasta=$PROJECT_18S.unique.abund.align, processors=32)"
```

(12) Precluster the sequences

The number of differences taken into account can be changed. In general use `diffs=2`. However if one does not want to make OTUS for example to look at fine genetic variation, it is necessary to remove this step.

```
$MOTHUR "#pre.cluster(fasta=$PROJECT_18S.unique.abund.filter.fasta,
                      count=$PROJECT_18S.abund.count_table,
                      diffs=1, processors=32)"
```

(13) Remove chimeras

```
$MOTHUR "#chimera.uchime(fasta=$PROJECT_18S.unique.abund.filter.precluster.fasta,
                         count=$PROJECT_18S.unique.abund.filter.precluster.count_table,
                         processors=32)"

$MOTHUR "#remove.seqs(fasta=$PROJECT_18S.unique.abund.filter.precluster.fasta,
                      accnos=$PROJECT_18S.unique.abund.filter.precluster.denovo.uchime.accnos,
                      count=$PROJECT_18S.unique.abund.filter.precluster.count_table)"
```

(14) Remove sequences in low abundance (here cutoff=2)

It is critical to remove the sequences in low abundance to speed up processing. In general use `cutoff = 10`.

```
$MOTHUR "#split.abund(count=$PROJECT_18S.unique.abund.filter.precluster.pick.count_table,
                      fasta=$PROJECT_18S.unique.abund.filter.precluster.pick.fasta,
                      cutoff=2, accnos=true)"
```

(15) Remove sequences that are too short or too long (here minlength=200)

```
$MOTHUR "#screen.seqs(fasta=$PROJECT_18S.unique.abund.filter.precluster.pick.abund.fasta,
                      count=$PROJECT_18S.unique.abund.filter.precluster.pick.abund.count_table,
                      minlength=200, processors=32)"
```

(16) Rename files to remember that sequences in low abundance where removed

```
cp $PROJECT_18S.unique.abund.filter.precluster.pick.abund.good.fasta
   $PROJECT_18S.uniq.preclust.no_chim.more_than_2.fasta
cp $PROJECT_18S.unique.abund.filter.precluster.pick.abund.good.count_table
   $PROJECT_18S.uniq.preclust.no_chim.more_than_2.count_table
```

(17) Classify the sequences using the PR2 database

Two files are required

- pr2.fasta
- pr2.taxo

```
$MOTHUR "#classify.seqs(fasta=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.fasta,
            count=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.count_table,
            reference=$FILE_PR2.fasta, taxonomy=$FILE_PR2.tax,
            processors=32,
            probs=T)"
```

(18) Compute distance matrix

It is critical to have as few sequences as possible at this step because the computation time is proportionnal to the **square** of the number of sequences.

```
$MOTHUR "#dist.seqs(fasta=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.fasta, processors=32)"
```

(19) Cluster the sequences to create the OTUs

Here we use a 0.02 cutoff corresponding to 98% similarity.

```
$MOTHUR "#cluster(column=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.dist,
                  count=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.count_table,
                  cutoff=0.02, processors=32)"
```

(20) Classify the OTUs based on the classification of the sequences (see above)

```
$MOTHUR "#classify.otu(taxonomy=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.$FILE_PR2_END.wang.taxonomy,
                       count=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.count_table,
                       list=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.opti_mcc.list,
                       label=0.02, probs=F, basis=sequence)"
```

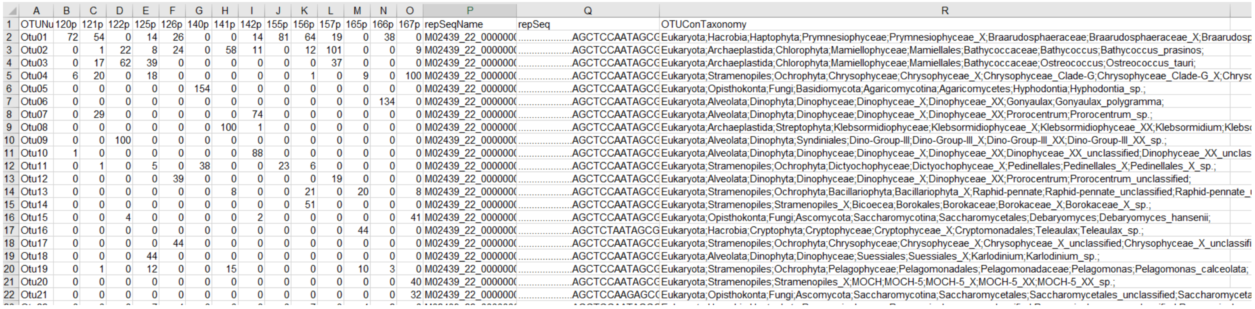
(21) Get sequences represnetative of each OTU

```
$MOTHUR "#get.oturep(fasta=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.fasta,
                     column=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.dist,
                     count=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.count_table,
                     list=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.opti_mcc.list,
                     method=abundance,
                     cutoff=0.02)"
```

(22) Format the final result in a single synthetic file

- otu id
- abundance in each sample
- representative sequence
- taxonomy

```
$MOTHUR "#create.database(list=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.opti_mcc.list,
          count=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.opti_mcc.0.02.rep.count_table,
          label=0.02,
          repfasta=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.opti_mcc.0.02.rep.fasta ,
          constaxonomy=$PROJECT_18S.uniq.preclust.no_chim.more_than_2.opti_mcc.0.02.cons.taxonomy)"
```



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	OTUNu	120p	121p	122p	125p	126p	140p	141p	142p	155p	156p	157p	165p	166p	167p	repSeqName	repSeq	OTUConTaxonomy
2	Otu01	72	54	0	14	26	0	0	14	81	64	19	0	38	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Hacrobia;Haptophyta;Prymnesiophyceae;Prymnesiophyceae_X;Braarudosphaeraceae;Braarudosphaeraceae_X;Braarudosp
3	Otu02	0	1	22	8	24	0	58	11	0	12	101	0	0	9	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Archaeplastida;Chlorophyta;Mamiellophyceae;Mamiellales;Bathycoccaceae;Bathycoccus;Bathycoccus_prasinos;
4	Otu03	0	17	62	39	0	0	0	0	0	0	37	0	0	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Archaeplastida;Chlorophyta;Mamiellophyceae;Mamiellales;Bathycoccaceae;Ostreococcus;Ostreococcus_tauri;
5	Otu04	6	20	0	18	0	0	0	0	0	1	0	9	0	100	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Stramenopiles;Ochrophyta;Chrysophyceae;Chrysophyceae_X;Chrysophyceae_Clade-G;Chrysophyceae_Clade-G_X;Chryso
6	Otu05	0	0	0	0	0	154	0	0	0	0	0	0	0	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Opisthokonta;Fungi;Basidiomycota;Agaricomycotina;Agaricomycetes;Hyphodontia;Hyphodontia_sp.;
7	Otu06	0	0	0	0	0	0	0	0	0	0	0	0	134	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Alveolata;Dinophyta;Dinophyceae;Dinophyceae_X;Dinophyceae_XX;Gonyaulax;Gonyaulax_polygramma;
8	Otu07	0	29	0	0	0	0	0	74	0	0	0	0	0	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Alveolata;Dinophyta;Dinophyceae;Dinophyceae_X;Dinophyceae_XX;Prorocentrum;Prorocentrum_sp.;
9	Otu08	0	0	0	0	0	0	100	1	0	0	0	0	0	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Archaeplastida;Streptophyta;Klebsormidiophyceae;Klebsormidiophyceae_X;Klebsormidiophyceae_XX;Klebsormidium;Klebso
10	Otu09	0	0	100	0	0	0	0	0	0	0	0	0	0	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Alveolata;Dinophyta;Syndiniales;Dino-Group-III;Dino-Group-III_X;Dino-Group-III_XX;Dino-Group-III_XX_sp.;
11	Otu10	1	0	0	0	0	0	0	88	0	0	0	0	0	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Alveolata;Dinophyta;Dinophyceae;Dinophyceae_X;Dinophyceae_XX;Dinophyceae_XX_unclassified;Dinophyceae_XX_unclas
12	Otu11	0	1	0	5	0	38	0	0	23	6	0	0	0	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Stramenopiles;Ochrophyta;Dictyochophyceae;Dictyochophyceae_X;Pedinellales;Pedinellales_X;Pedinellales_X_sp.;
13	Otu12	0	0	0	0	39	0	0	0	0	0	19	0	0	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Alveolata;Dinophyta;Dinophyceae;Dinophyceae_X;Dinophyceae_XX;Prorocentrum;Prorocentrum_unclassified;
14	Otu13	0	0	0	0	0	0	8	0	0	21	0	20	0	8	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Stramenopiles;Ochrophyta;Bacillariophyta;Bacillariophyta_X;Raphid-pennate;Raphid-pennate_unclassified;Raphid-pennate_(
15	Otu14	0	0	0	0	0	0	0	0	0	51	0	0	0	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Stramenopiles;Stramenopiles_X;Bicoecea;Borokales;Borokaceae;Borokaceae_X;Borokaceae_X_sp.;
16	Otu15	0	0	4	0	0	0	0	2	0	0	0	0	0	41	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Opisthokonta;Fungi;Ascomycota;Saccharomycotina;Saccharomycetales;Debaryomyces;Debaryomyces_hansenii;
17	Otu16	0	0	0	0	0	0	0	0	0	0	0	44	0	0	M02439_22_000000(	..................AGCTCTAATAGCG	Eukaryota;Hacrobia;Cryptophyta;Cryptophyceae;Cryptophyceae_X;Cryptomonadales;Teleaulax;Teleaulax_sp.;
18	Otu17	0	0	0	0	44	0	0	0	0	0	0	0	0	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Stramenopiles;Ochrophyta;Chrysophyceae;Chrysophyceae_X;Chrysophyceae_X_unclassified;Chrysophyceae_X_unclassifi
19	Otu18	0	0	0	44	0	0	0	0	0	0	0	0	0	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Alveolata;Dinophyta;Dinophyceae;Suessiales;Suessiales_X;Karlodinium;Karlodinium_sp.;
20	Otu19	0	1	0	12	0	0	15	0	0	0	0	10	3	0	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Stramenopiles;Ochrophyta;Pelagophyceae;Pelagomonadales;Pelagomonadaceae;Pelagomonas;Pelagomonas_calceolata;
21	Otu20	0	0	0	0	0	0	0	0	0	0	0	0	0	40	M02439_22_000000(	..................AGCTCCAATAGCG	Eukaryota;Stramenopiles;Stramenopiles_X;MOCH;MOCH-5;MOCH-5_X;MOCH-5_XX;MOCH-5_XX_sp.;
22	Otu21	0	0	0	0	0	0	0	0	0	0	0	0	0	32	M02439_22_000000(	..................AGCTCCAAGAGCC	Eukaryota;Opisthokonta;Fungi;Ascomycota;Saccharomycotina;Saccharomycetales;Saccharomycetales_unclassified;Saccharomyceta

7 What is next ?

- It is a good practice to confirm the phylogeny of at least the major OTUs by BLAST
- The database format can be easily used by the phyloseq package. A short tutorial can be found here : https://github.com/vaulot/R_tutorials

8 Alternative strategies

- Use the R dada2 package : https://benjjneb.github.io/dada2/tutorial.html
- Use vsearch : https://github.com/torognes/vsearch/wiki/VSEARCH-pipeline