Fastq analysis script

Tian You

2025-03-12

#Introduction

The following script is the 16S rRNA gene amplicon sequencing analysis for our paper "Evaluating Water-Vapor Treatment for Effective Control of Legionella pneumophila". A total of 31 samples were analyzed

#First, load library.

```
library(Rcpp)#For improving performance
library(dada2)#For correcting sequencing errors to generate ASVs
library(ShortRead)#For correcting sequencing errors to generate ASVs
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, table,
##     tapply, union, unique, unsplit, which.max, which.min
```

```
## Loading required package: BiocParallel
```

```
## Loading required package: Biostrings
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:utils':
##
##     findMatches
```

```
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
```

```
## Loading required package: IRanges
```

```
##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: XVector

## Loading required package: GenomeInfoDb

##
## Attaching package: 'Biostrings'

## The following object is masked from 'package:base':
##
##     strsplit

## Loading required package: Rsamtools

## Loading required package: GenomicRanges

## Loading required package: GenomicAlignments

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'MatrixGenerics'

## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'
```

```
## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians
## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
library(ggplot2)#For visualizing the mass fraction distribution of sequencing data
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
library(Biostrings)#For processing sequence data in FASTA and FASTQ files
```

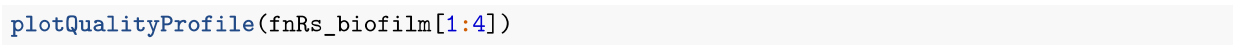
#Second, load the files

```
# Set file path
path_biofilm <- "C:\\Users\\ASUS\\Documents\\project1\\data"

# Adding files
fnFs_biofilm <- sort(list.files(path_biofilm, pattern="_R1.fastq.gz", full.names = TRUE))
fnRs_biofilm <- sort(list.files(path_biofilm, pattern="_R2.fastq.gz", full.names = TRUE))

# Extract sample names
sample.namesF_biofilm <- sapply(strsplit(basename(fnFs_biofilm), "_R1"), `[`, 1)
sample.namesR_biofilm <- sapply(strsplit(basename(fnRs_biofilm), "_R2"), `[`, 1)

# Generate quality profile plots for the first four reverse read files
plotQualityProfile(fnFs_biofilm[1:4])
```



```
plotQualityProfile(fnRs_biofilm[1:4])
```


#Third Filter the sequence

```
# Set paths for filtered forward (R1) and reverse (R2) FASTQ files.
filtFs_biofilm <- file.path(path_biofilm, "dada2_filtered",
                            paste0(sample.namesF_biofilm, "_R1_filt.fastq.gz"))
filtRs_biofilm <- file.path(path_biofilm, "dada2_filtered",
                            paste0(sample.namesR_biofilm, "_R2_filt.fastq.gz"))

# Name filtered file paths with corresponding sample names.
names(filtFs_biofilm) <- sample.namesF_biofilm
names(filtRs_biofilm) <- sample.namesR_biofilm

# Filter and trim raw FASTQ files for quality control.
filtering_results <- filterAndTrim(
  fnFs_biofilm, filtFs_biofilm,     # Input/output forward FASTQ files.
  fnRs_biofilm, filtRs_biofilm,     # Input/output reverse FASTQ files.
  trimLeft = c(5, 5),               # Trim 5 bases from 5' end of reads.
  maxEE = c(1.5, 1.5),              # Max expected errors: 1.5.
  truncLen = c(250, 250),           # Truncate reads to 250 bases.
  compress = TRUE,                  # Gzip output files.
  matchIDs = TRUE,                  # Ensure paired-end read ID matching.
  multithread = FALSE               # Disable multithreading.
```

```
)

# Show filtering results (reads in/out per sample).
filtering_results
```

```
##                       reads.in reads.out
## Bu1C4-1_R1.fastq.gz      60503     60362
## Bu1C4-2_R1.fastq.gz      61459     61339
## Bu1C4-3_R1.fastq.gz      83664     83518
## Bu1CT4-1_R1.fastq.gz        82        80
## Bu1CT4-2_R1.fastq.gz       116       112
## Bu1CT4-3_R1.fastq.gz     91702     91598
## Bu1HW4-1_R1.fastq.gz     33346     32990
## Bu1HW4-2_R1.fastq.gz        61        60
## Bu1HW4-3_R1.fastq.gz      5663      5454
## Bu7C1_R1.fastq.gz        80143     74201
## Bu7C5-1_R1.fastq.gz      67536     65996
## Bu7C5-2_R1.fastq.gz      41762     36103
## Bu7CT1_R1.fastq.gz       64601     64228
## Bu7CT5-1_R1.fastq.gz     55812     55436
## Bu7CT5-2_R1.fastq.gz      2957      2738
## BU7HW1_R1.fastq.gz       99624     97574
## BU7HW5-1_R1.fastq.gz     77520     77193
## BU7HW5-2_R1.fastq.gz     57873     57584
## NiC1_R1.fastq.gz        122362    120860
## NiCT1_R1.fastq.gz        39428     39306
## NoC1_R1.fastq.gz        185364    181945
## NoCT1_R1.fastq.gz       150821    150352
## PrC1-1_R1.fastq.gz       78293     76945
## PrC1-2_R1.fastq.gz       82784     80141
## PrCT1-1_R1.fastq.gz     145506    123913
## PrCT1-2_R1.fastq.gz     104893    104494
## SE511_R1.fastq.gz           35        34
## SE512_R1.fastq.gz        20201     20173
## SE513_R1.fastq.gz           30        30
## SE612_R1.fastq.gz           31        26
## SE613_R1.fastq.gz        40710     40658
```

#Forth,inferring ASVs

```
# 1. Learn error rate
errF <- learnErrors(filtFs_biofilm, multithread = FALSE)
```

```
## 100379930 total bases in 409714 reads from 10 samples will be used for learning the error rates.
```

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

```
## 100379930 total bases in 409714 reads from 10 samples will be used for learning the error rates.
```

```
# 2.Denoise sequences using DADA2 algorithm.
dadaFs <- dada(filtFs_biofilm, err = errF, multithread = FALSE)
```

```
## Sample 1 - 60362 reads in 11675 unique sequences.
## Sample 2 - 61339 reads in 11976 unique sequences.
## Sample 3 - 83518 reads in 12575 unique sequences.
## Sample 4 - 80 reads in 53 unique sequences.
## Sample 5 - 112 reads in 69 unique sequences.
```

```
## Sample 6 - 91598 reads in 13025 unique sequences.
## Sample 7 - 32990 reads in 4885 unique sequences.
## Sample 8 - 60 reads in 47 unique sequences.
## Sample 9 - 5454 reads in 1564 unique sequences.
## Sample 10 - 74201 reads in 15221 unique sequences.
## Sample 11 - 65996 reads in 13975 unique sequences.
## Sample 12 - 36103 reads in 9918 unique sequences.
## Sample 13 - 64228 reads in 13959 unique sequences.
## Sample 14 - 55436 reads in 8548 unique sequences.
## Sample 15 - 2738 reads in 1138 unique sequences.
## Sample 16 - 97574 reads in 17859 unique sequences.
## Sample 17 - 77193 reads in 14785 unique sequences.
## Sample 18 - 57584 reads in 10997 unique sequences.
## Sample 19 - 120860 reads in 27197 unique sequences.
## Sample 20 - 39306 reads in 5647 unique sequences.
## Sample 21 - 181945 reads in 34426 unique sequences.
## Sample 22 - 150352 reads in 19110 unique sequences.
## Sample 23 - 76945 reads in 18380 unique sequences.
## Sample 24 - 80141 reads in 20623 unique sequences.
## Sample 25 - 123913 reads in 24785 unique sequences.
## Sample 26 - 104494 reads in 14335 unique sequences.
## Sample 27 - 34 reads in 31 unique sequences.
## Sample 28 - 20173 reads in 3874 unique sequences.
## Sample 29 - 30 reads in 24 unique sequences.
## Sample 30 - 26 reads in 23 unique sequences.
## Sample 31 - 40658 reads in 6609 unique sequences.
```

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

```
## Sample 1 - 60362 reads in 12893 unique sequences.
## Sample 2 - 61339 reads in 13128 unique sequences.
## Sample 3 - 83518 reads in 14115 unique sequences.
## Sample 4 - 80 reads in 59 unique sequences.
## Sample 5 - 112 reads in 73 unique sequences.
## Sample 6 - 91598 reads in 14476 unique sequences.
## Sample 7 - 32990 reads in 5967 unique sequences.
## Sample 8 - 60 reads in 49 unique sequences.
## Sample 9 - 5454 reads in 1840 unique sequences.
## Sample 10 - 74201 reads in 18530 unique sequences.
## Sample 11 - 65996 reads in 17193 unique sequences.
## Sample 12 - 36103 reads in 11673 unique sequences.
## Sample 13 - 64228 reads in 14425 unique sequences.
## Sample 14 - 55436 reads in 10802 unique sequences.
## Sample 15 - 2738 reads in 1279 unique sequences.
## Sample 16 - 97574 reads in 20639 unique sequences.
## Sample 17 - 77193 reads in 17318 unique sequences.
## Sample 18 - 57584 reads in 12459 unique sequences.
## Sample 19 - 120860 reads in 28586 unique sequences.
## Sample 20 - 39306 reads in 6623 unique sequences.
## Sample 21 - 181945 reads in 37932 unique sequences.
## Sample 22 - 150352 reads in 21797 unique sequences.
## Sample 23 - 76945 reads in 20548 unique sequences.
## Sample 24 - 80141 reads in 22466 unique sequences.
## Sample 25 - 123913 reads in 27947 unique sequences.
## Sample 26 - 104494 reads in 16767 unique sequences.
```

```
## Sample 27 - 34 reads in 29 unique sequences.
## Sample 28 - 20173 reads in 4269 unique sequences.
## Sample 29 - 30 reads in 23 unique sequences.
## Sample 30 - 26 reads in 26 unique sequences.
## Sample 31 - 40658 reads in 7175 unique sequences.
```

```
# 3. Merge paired-end reads.
mergers <- mergePairs(dadaFs, filtFs_biofilm, dadaRs, filtRs_biofilm, verbose = TRUE)
```

```
## 60020 paired-reads (in 765 unique pairings) successfully merged out of 60080 (in 808 pairings) input
## 61025 paired-reads (in 939 unique pairings) successfully merged out of 61150 (in 1051 pairings) inpu
## 83142 paired-reads (in 648 unique pairings) successfully merged out of 83211 (in 684 pairings) input
## 56 paired-reads (in 5 unique pairings) successfully merged out of 56 (in 5 pairings) input.
## 79 paired-reads (in 4 unique pairings) successfully merged out of 83 (in 6 pairings) input.
## 91245 paired-reads (in 530 unique pairings) successfully merged out of 91347 (in 587 pairings) input
## 32813 paired-reads (in 299 unique pairings) successfully merged out of 32837 (in 310 pairings) input
## 34 paired-reads (in 4 unique pairings) successfully merged out of 34 (in 4 pairings) input.
## 5380 paired-reads (in 159 unique pairings) successfully merged out of 5383 (in 162 pairings) input.
## 73736 paired-reads (in 997 unique pairings) successfully merged out of 73931 (in 1158 pairings) inpu
## 65543 paired-reads (in 1343 unique pairings) successfully merged out of 65647 (in 1431 pairings) inp
## 35601 paired-reads (in 1012 unique pairings) successfully merged out of 35668 (in 1071 pairings) inp
## 62962 paired-reads (in 1205 unique pairings) successfully merged out of 63257 (in 1352 pairings) inp
## 55095 paired-reads (in 402 unique pairings) successfully merged out of 55245 (in 497 pairings) input
## 2479 paired-reads (in 135 unique pairings) successfully merged out of 2507 (in 141 pairings) input.
## 96622 paired-reads (in 1496 unique pairings) successfully merged out of 96848 (in 1688 pairings) inp
## 76538 paired-reads (in 1001 unique pairings) successfully merged out of 76890 (in 1274 pairings) inp
## 57054 paired-reads (in 903 unique pairings) successfully merged out of 57232 (in 1032 pairings) inpu
## 119602 paired-reads (in 1687 unique pairings) successfully merged out of 120002 (in 1914 pairings) i
## 38942 paired-reads (in 543 unique pairings) successfully merged out of 39125 (in 656 pairings) input
## 180665 paired-reads (in 2252 unique pairings) successfully merged out of 181156 (in 2653 pairings) i
## 148888 paired-reads (in 1159 unique pairings) successfully merged out of 149704 (in 1497 pairings) i
## 75357 paired-reads (in 1998 unique pairings) successfully merged out of 75663 (in 2184 pairings) inp
## 78340 paired-reads (in 1977 unique pairings) successfully merged out of 78859 (in 2190 pairings) inp
## 122143 paired-reads (in 1911 unique pairings) successfully merged out of 122760 (in 2281 pairings) i
## 104012 paired-reads (in 771 unique pairings) successfully merged out of 104149 (in 836 pairings) inp
## 5 paired-reads (in 1 unique pairings) successfully merged out of 5 (in 1 pairings) input.
## 20071 paired-reads (in 177 unique pairings) successfully merged out of 20071 (in 177 pairings) input
## 21 paired-reads (in 2 unique pairings) successfully merged out of 21 (in 2 pairings) input.
## 0 paired-reads (in 0 unique pairings) successfully merged out of 5 (in 1 pairings) input.
## 40509 paired-reads (in 176 unique pairings) successfully merged out of 40523 (in 183 pairings) input
```

```r
# 4. Create sequence table
seqtab <- makeSequenceTable(mergers)

# 5. Remove chimeric sequences.
seqtab.nochim <- removeBimeraDenovo(seqtab, method = "consensus", multithread = FALSE, verbose = TRUE)
```

```
## Identified 9931 bimeras out of 11385 input sequences.
```

```r
#save asv table
file_path <- "C:\\Users\\ASUS\\Documents\\project1\\ASV_TABLE.csv"
write.csv(seqtab.nochim, file = file_path, quote = FALSE)
```

#Fifth,assign taxanomy

```r
# 1. Set path to SILVA database for taxonomy assignment.
silva_db <- "C:\\Users\\ASUS\\Documents\\project1\\silva_nr99_v138.1_train_set.fa.gz"

# 2. Assign taxonomy to ASVs using SILVA database.
taxa <- assignTaxonomy(
  seqtab.nochim,          # Input ASV sequence table.
  refFasta = silva_db,    # Reference SILVA database.
  multithread = FALSE     # Disable multithreading.
)

# 3. Add species-level assignments using SILVA species database.
silva_species_db <- "C:\\Users\\ASUS\\Documents\\project1\\silva_species_assignment_v138.1.fa.gz"
taxa <- addSpecies(taxa, silva_species_db)

# 4. Preview taxonomy results and create a data frame.
head(taxa)
```

```
##
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATCTTGCGCAATGGGCGAAAGCCTGACGCAGCCATGCCGCGTGAATGATGAAGGTCTTAGGATTGTAAAATTCTTTC
##
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATCTTGCGCAATGGGCGAAAGCCTGACGCAGCCATGCCGCGTGAATGATGAAGGTCTTAGGATTGTAAAATTCTTTC
##
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATCTTGCGCAATGGGCGAAAGCCTGACGCAGCCATGCCGCGTGAATGATGAAGGTCTTAGGATTGTAAAATTCTTTC
##
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
```

```
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATCTTGCGCAATGGGCGAAAGCCTGACGCAGCCATGCCGCGTGAATGATGAAGGTCTTAGGATTGTAAAATTCTTTC
##
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATCTTGCGCAATGGGCGAAAGCCTGACGCAGCCATGCCGCGTGAATGATGAAGGTCTTAGGATTGTAAAATTCTTTC
##
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATCTTGCGCAATGGGCGAAAGCCTGACGCAGCCATGCCGCGTGAATGATGAAGGTCTTAGGATTGTAAAATTCTTTC
##
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGTGGCTGCAGTGGGGAATTTTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGCAGGATGAAGGCCTTCGGGTTGTAAACTGCTTTT
## CCTACGGGGGGCTGCAGTGGGGAATCTTGCGCAATGGGCGAAAGCCTGACGCAGCCATGCCGCGTGAATGATGAAGGTCTTAGGATTGTAAAATTCTTTC
```

```
# 5. Save taxanomy table
asv_sequences <- colnames(seqtab.nochim)
taxa_df <- data.frame(
  ASV_ID = paste0("ASV", seq_along(asv_sequences)),
  Sequence = asv_sequences,
  taxa,
  row.names = NULL
)
write.csv(taxa_df, file = "ASV_Taxonomy.csv", quote = FALSE, row.names = FALSE)
```