DADA2CHSP

Tricia

2024-05-25

##Load required packages

```
library(dada2)
library(Biostrings)
```

##Provide path to sequences

```
path <- "../sequences"
list.files(path)
```

```
##  [1] "103_S309_L001_R1_001.fastq"       "103_S309_L001_R2_001.fastq"
##  [3] "123_S332_L001_R1_001.fastq"       "123_S332_L001_R2_001.fastq"
##  [5] "180_S346_L001_R1_001.fastq"       "180_S346_L001_R2_001.fastq"
##  [7] "184_S298_L001_R1_001.fastq"       "184_S298_L001_R2_001.fastq"
##  [9] "186_S370_L001_R1_001.fastq"       "186_S370_L001_R2_001.fastq"
## [11] "2_S297_L001_R1_001.fastq"         "2_S297_L001_R2_001.fastq"
## [13] "20_S369_L001_R1_001.fastq"        "20_S369_L001_R2_001.fastq"
## [15] "207_S357_L001_R1_001.fastq"       "207_S357_L001_R2_001.fastq"
## [17] "214_S322_L001_R1_001.fastq"       "214_S322_L001_R2_001.fastq"
## [19] "22_S358_L001_R1_001.fastq"        "22_S358_L001_R2_001.fastq"
## [21] "237_S333_L001_R1_001.fastq"       "237_S333_L001_R2_001.fastq"
## [23] "262_S320_L001_R1_001.fastq"       "262_S320_L001_R2_001.fastq"
## [25] "283_S381_L001_R1_001.fastq"       "283_S381_L001_R2_001.fastq"
## [27] "298_S356_L001_R1_001.fastq"       "298_S356_L001_R2_001.fastq"
## [29] "319_S310_L001_R1_001.fastq"       "319_S310_L001_R2_001.fastq"
## [31] "326_S334_L001_R1_001.fastq"       "326_S334_L001_R2_001.fastq"
## [33] "329_S382_L001_R1_001.fastq"       "329_S382_L001_R2_001.fastq"
## [35] "372_S345_L001_R1_001.fastq"       "372_S345_L001_R2_001.fastq"
## [37] "377_S321_L001_R1_001.fastq"       "377_S321_L001_R2_001.fastq"
## [39] "39_S368_L001_R1_001.fastq"        "39_S368_L001_R2_001.fastq"
## [41] "8_S344_L001_R1_001.fastq"         "8_S344_L001_R2_001.fastq"
## [43] "93_S380_L001_R1_001.fastq"        "93_S380_L001_R2_001.fastq"
## [45] "Blk1_S108_L001_R1_001.fastq"      "Blk1_S108_L001_R2_001.fastq"
## [47] "Blk10_S1_L001_R1_001.fastq"       "Blk10_S1_L001_R2_001.fastq"
## [49] "Blk11_S12_L001_R1_001.fastq"      "Blk11_S12_L001_R2_001.fastq"
## [51] "Blk12_S373_L001_R1_001.fastq"     "Blk12_S373_L001_R2_001.fastq"
## [53] "Blk2_S97_L001_R1_001.fastq"       "Blk2_S97_L001_R2_001.fastq"
## [55] "Blk3_S85_L001_R1_001.fastq"       "Blk3_S85_L001_R2_001.fastq"
## [57] "Blk4_S193_L001_R1_001.fastq"      "Blk4_S193_L001_R2_001.fastq"
## [59] "Blk5_S277_L001_R1_001.fastq"      "Blk5_S277_L001_R2_001.fastq"
## [61] "Blk6_S204_L001_R1_001.fastq"      "Blk6_S204_L001_R2_001.fastq"
## [63] "Blk7_S181_L001_R1_001.fastq"      "Blk7_S181_L001_R2_001.fastq"
## [65] "Blk8_S300_L001_R1_001.fastq"      "Blk8_S300_L001_R2_001.fastq"
## [67] "Blk9_S289_L001_R1_001.fastq"      "Blk9_S289_L001_R2_001.fastq"
```

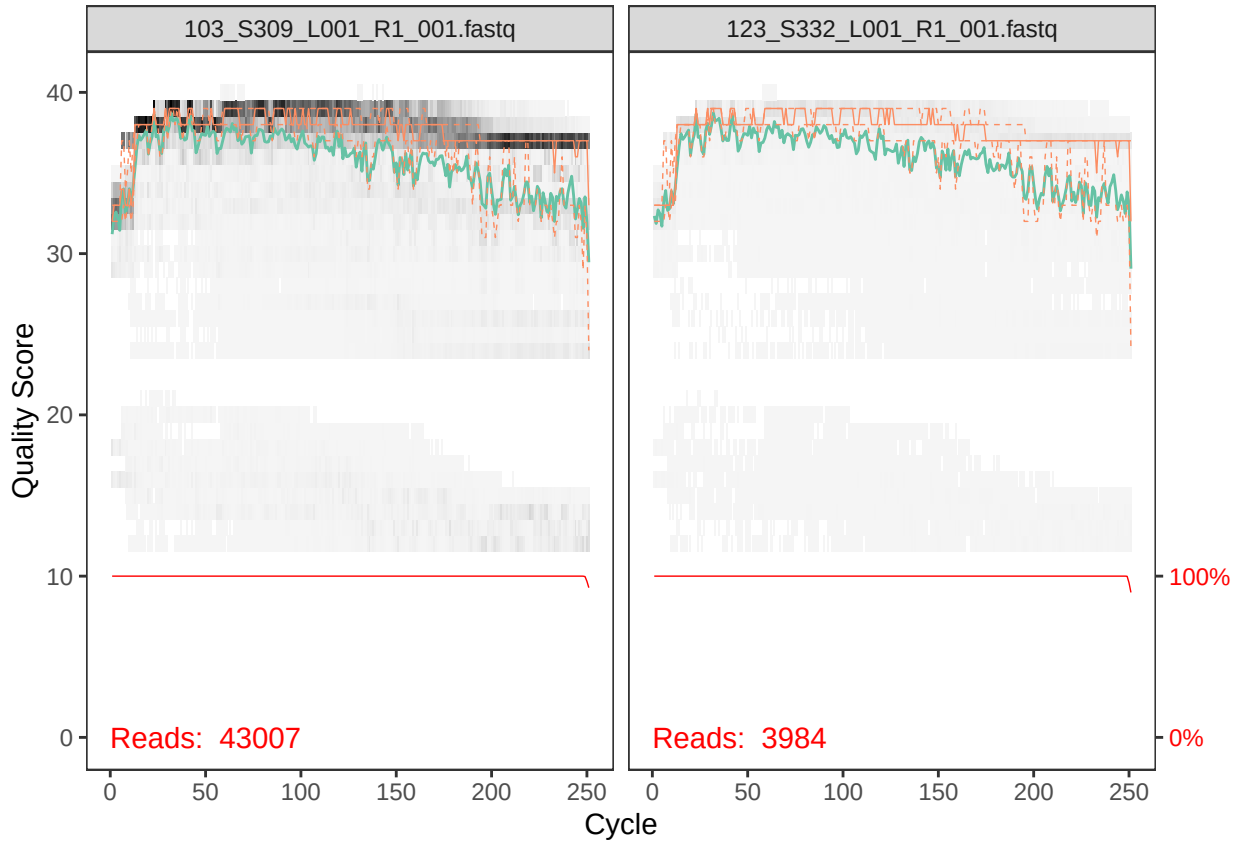
```
## [69] "filtered"                           "WaterNegA_S96_L001_R1_001.fastq"
## [71] "WaterNegA_S96_L001_R2_001.fastq"   "WaterNegB_S192_L001_R1_001.fastq"
## [73] "WaterNegB_S192_L001_R2_001.fastq" "WaterNegC_S288_L001_R1_001.fastq"
## [75] "WaterNegC_S288_L001_R2_001.fastq" "WaterNegD_S384_L001_R1_001.fastq"
## [77] "WaterNegD_S384_L001_R2_001.fastq" "ZymoPosA_S84_L001_R1_001.fastq"
## [79] "ZymoPosA_S84_L001_R2_001.fastq"    "ZymoPosB_S180_L001_R1_001.fastq"
## [81] "ZymoPosB_S180_L001_R2_001.fastq"  "ZymoPosC_S276_L001_R1_001.fastq"
## [83] "ZymoPosC_S276_L001_R2_001.fastq"  "ZymoPosD_S372_L001_R1_001.fastq"
## [85] "ZymoPosD_S372_L001_R2_001.fastq"
```

##Import file names and make matched list

```
# Forward and reverse fastq filenames have format: SAMPLENAME_R1_001.fastq and SAMPLENAME_R2_001.fastq
fnFs <- sort(list.files(path, pattern="_R1_001.fastq", full.names = TRUE))
fnRs <- sort(list.files(path, pattern="_R2_001.fastq", full.names = TRUE))
# Extract sample names, assuming filenames have format: SAMPLENAME_XXX.fastq
sample.names <- sapply(strsplit(basename(fnFs), "_"), `[`, 1)
```

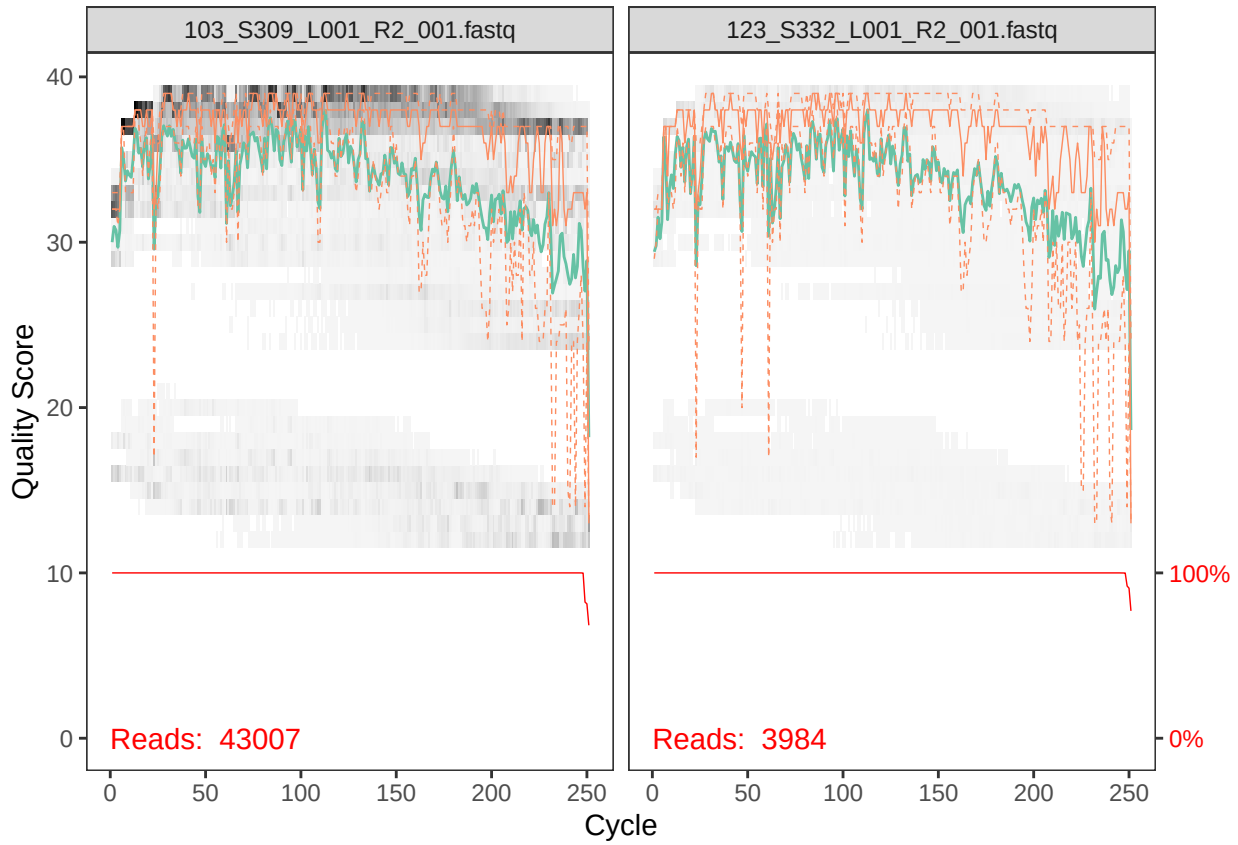
##Inspect forward read quality

```
plotQualityProfile(fnFs[1:2])
```



##Inspect reverse read quality

```
plotQualityProfile(fnRs[1:2])
```



##Assign file names for filtered reads

```
# Place filtered files in filtered/ subdirectory
filtFs <- file.path(path, "filtered", paste0(sample.names, "_F_filt.fastq.gz"))
filtRs <- file.path(path, "filtered", paste0(sample.names, "_R_filt.fastq.gz"))
names(filtFs) <- sample.names
names(filtRs) <- sample.names
```

##Filter reads

```
# Filter based on quality plots above. for this work, trim first 10 from F and R, trunc 240 F 210 R
out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs, truncLen=c(240,200), trimLeft=c(10, 10),
              maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,
              compress=TRUE, multithread=TRUE)
head(out)
```

```
##                             reads.in reads.out
## 103_S309_L001_R1_001.fastq    43007     38430
## 123_S332_L001_R1_001.fastq     3984      3585
## 180_S346_L001_R1_001.fastq   112134    100851
## 184_S298_L001_R1_001.fastq    55020     48926
## 186_S370_L001_R1_001.fastq    29577     25491
## 2_S297_L001_R1_001.fastq      40953     35685
```

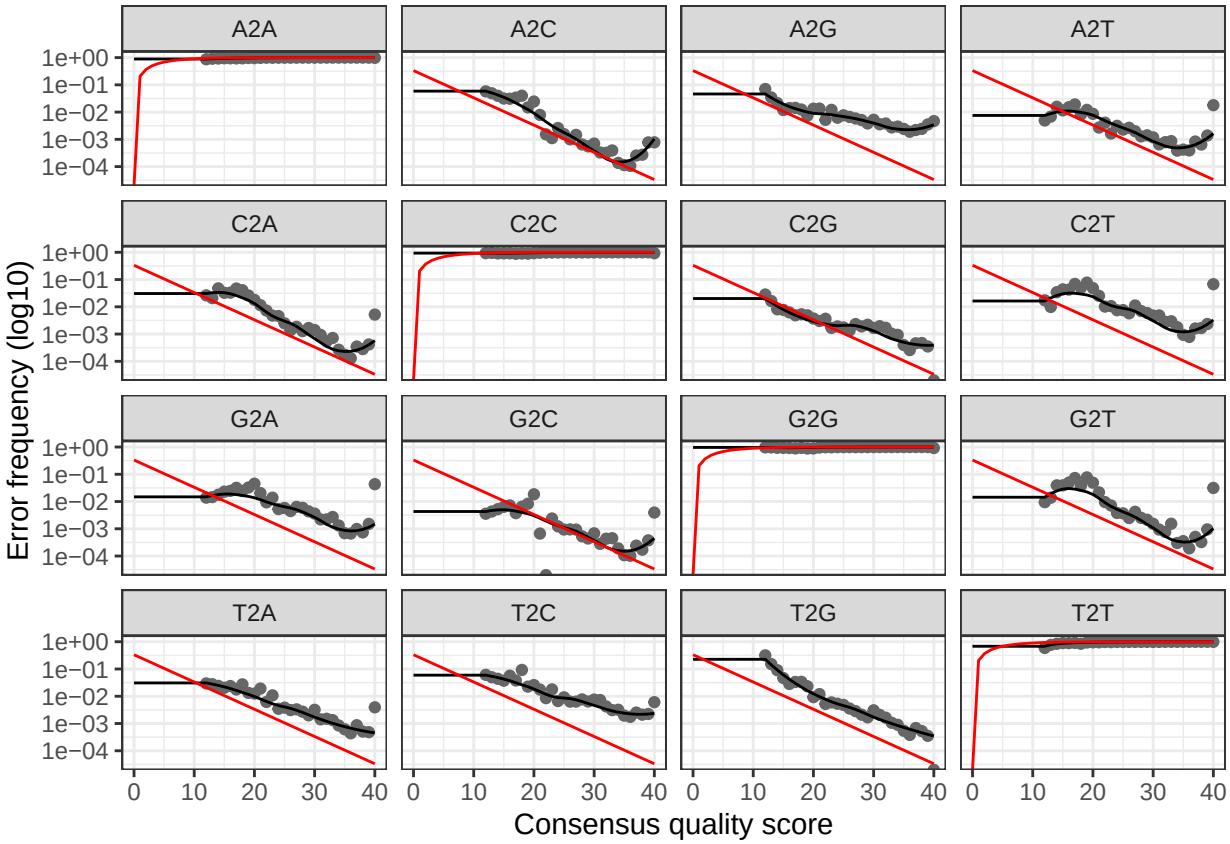
##learn error rates forward reads

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

```
## 102022250 total bases in 443575 reads from 14 samples will be used for learning the error rates.
```

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

```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
## log-10 transformation introduced infinite values.
```



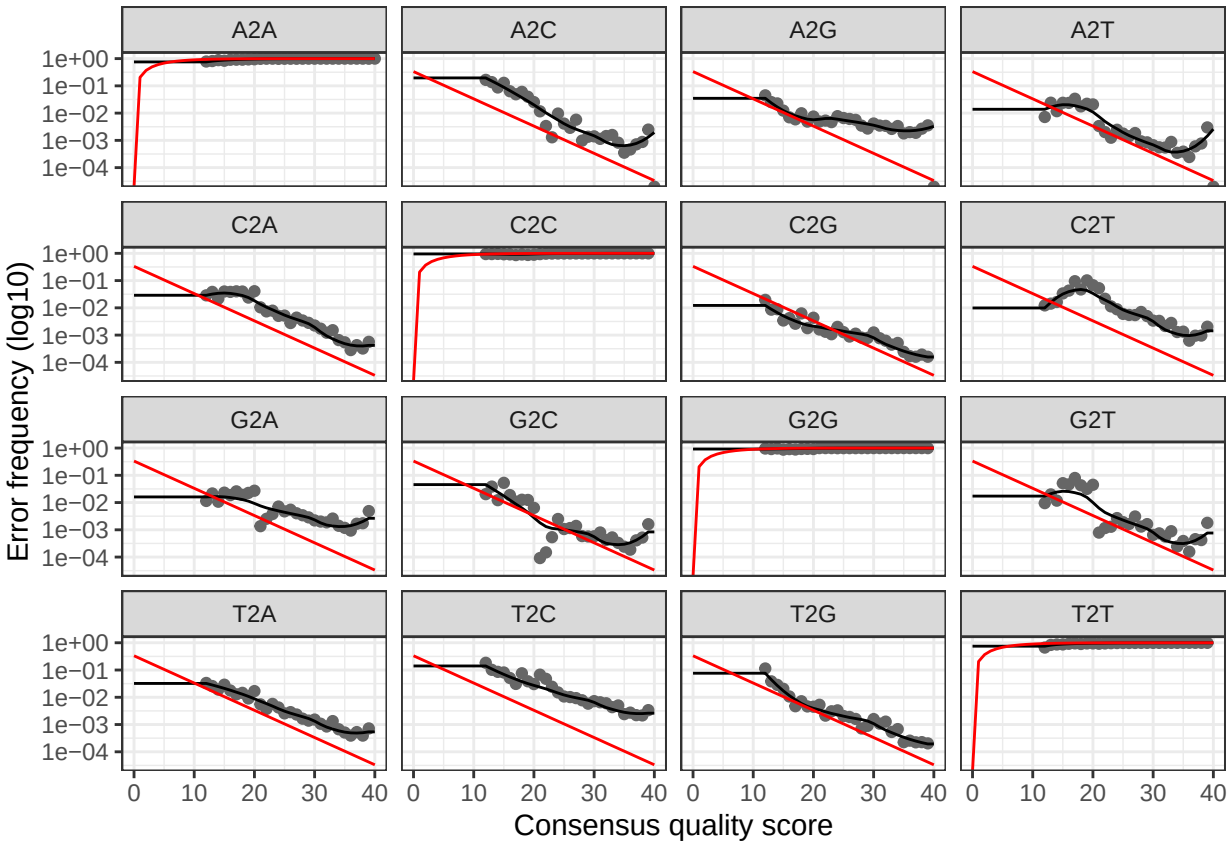
##learn error rates reverse reads

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

```
## 102395370 total bases in 538923 reads from 17 samples will be used for learning the error rates.
```

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

```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
## log-10 transformation introduced infinite values.
```



##Dereplicate

```
derepFs <- derepFastq(filtFs)
derepRs <- derepFastq(filtRs)
# Name the derep-class objects by the sample names
names(derepFs) <- sample.names
names(derepRs) <- sample.names
```

##Sample Inference Forward reads

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

```
## Sample 1 - 38430 reads in 12381 unique sequences.
## Sample 2 - 3585 reads in 1397 unique sequences.
## Sample 3 - 100851 reads in 12980 unique sequences.
## Sample 4 - 48926 reads in 14396 unique sequences.
## Sample 5 - 25491 reads in 8078 unique sequences.
## Sample 6 - 35685 reads in 12287 unique sequences.
## Sample 7 - 45840 reads in 13992 unique sequences.
## Sample 8 - 7037 reads in 2402 unique sequences.
## Sample 9 - 18049 reads in 4129 unique sequences.
## Sample 10 - 53469 reads in 10053 unique sequences.
## Sample 11 - 34316 reads in 10932 unique sequences.
## Sample 12 - 21166 reads in 5915 unique sequences.
## Sample 13 - 1461 reads in 631 unique sequences.
## Sample 14 - 9269 reads in 2448 unique sequences.
## Sample 15 - 18642 reads in 6375 unique sequences.
## Sample 16 - 40618 reads in 10402 unique sequences.
```

```
## Sample 17 - 36088 reads in 10585 unique sequences.
## Sample 18 - 42088 reads in 9745 unique sequences.
## Sample 19 - 53359 reads in 11712 unique sequences.
## Sample 20 - 9649 reads in 3283 unique sequences.
## Sample 21 - 48310 reads in 15374 unique sequences.
## Sample 22 - 10028 reads in 3568 unique sequences.
## Sample 23 - 89 reads in 37 unique sequences.
## Sample 24 - 1588 reads in 434 unique sequences.
## Sample 25 - 229 reads in 86 unique sequences.
## Sample 26 - 87 reads in 51 unique sequences.
## Sample 27 - 147 reads in 58 unique sequences.
## Sample 28 - 378 reads in 158 unique sequences.
## Sample 29 - 57 reads in 35 unique sequences.
## Sample 30 - 16 reads in 16 unique sequences.
## Sample 31 - 141 reads in 72 unique sequences.
## Sample 32 - 275 reads in 78 unique sequences.
## Sample 33 - 99 reads in 47 unique sequences.
## Sample 34 - 306 reads in 109 unique sequences.
## Sample 35 - 33 reads in 25 unique sequences.
## Sample 36 - 18 reads in 16 unique sequences.
## Sample 37 - 22 reads in 22 unique sequences.
## Sample 38 - 24 reads in 21 unique sequences.
## Sample 39 - 53009 reads in 12309 unique sequences.
## Sample 40 - 85992 reads in 18528 unique sequences.
## Sample 41 - 34133 reads in 9068 unique sequences.
## Sample 42 - 50636 reads in 12432 unique sequences.
```

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

```
## dada-class: object describing DADA2 denoising results
## 975 sequence variants were inferred from 12381 input unique sequences.
## Key parameters: OMEGA_A = 1e-40, OMEGA_C = 1e-40, BAND_SIZE = 16
```

##Sample Inference Reverse reads

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

```
## Sample 1 - 38430 reads in 15070 unique sequences.
## Sample 2 - 3585 reads in 1621 unique sequences.
## Sample 3 - 100851 reads in 23118 unique sequences.
## Sample 4 - 48926 reads in 18344 unique sequences.
## Sample 5 - 25491 reads in 12276 unique sequences.
## Sample 6 - 35685 reads in 14603 unique sequences.
## Sample 7 - 45840 reads in 22582 unique sequences.
## Sample 8 - 7037 reads in 2884 unique sequences.
## Sample 9 - 18049 reads in 5527 unique sequences.
## Sample 10 - 53469 reads in 14997 unique sequences.
## Sample 11 - 34316 reads in 14311 unique sequences.
## Sample 12 - 21166 reads in 7966 unique sequences.
## Sample 13 - 1461 reads in 753 unique sequences.
## Sample 14 - 9269 reads in 2463 unique sequences.
## Sample 15 - 18642 reads in 7560 unique sequences.
## Sample 16 - 40618 reads in 13451 unique sequences.
## Sample 17 - 36088 reads in 13710 unique sequences.
## Sample 18 - 42088 reads in 12331 unique sequences.
## Sample 19 - 53359 reads in 15910 unique sequences.
```

```
## Sample 20 - 9649 reads in 5039 unique sequences.
## Sample 21 - 48310 reads in 18504 unique sequences.
## Sample 22 - 10028 reads in 4491 unique sequences.
## Sample 23 - 89 reads in 43 unique sequences.
## Sample 24 - 1588 reads in 606 unique sequences.
## Sample 25 - 229 reads in 117 unique sequences.
## Sample 26 - 87 reads in 52 unique sequences.
## Sample 27 - 147 reads in 67 unique sequences.
## Sample 28 - 378 reads in 189 unique sequences.
## Sample 29 - 57 reads in 39 unique sequences.
## Sample 30 - 16 reads in 16 unique sequences.
## Sample 31 - 141 reads in 81 unique sequences.
## Sample 32 - 275 reads in 129 unique sequences.
## Sample 33 - 99 reads in 60 unique sequences.
## Sample 34 - 306 reads in 153 unique sequences.
## Sample 35 - 33 reads in 28 unique sequences.
## Sample 36 - 18 reads in 17 unique sequences.
## Sample 37 - 22 reads in 22 unique sequences.
## Sample 38 - 24 reads in 22 unique sequences.
## Sample 39 - 53009 reads in 17233 unique sequences.
## Sample 40 - 85992 reads in 21901 unique sequences.
## Sample 41 - 34133 reads in 10663 unique sequences.
## Sample 42 - 50636 reads in 17761 unique sequences.
```

```
dadaRs[[1]]
```

```
## dada-class: object describing DADA2 denoising results
## 898 sequence variants were inferred from 15070 input unique sequences.
## Key parameters: OMEGA_A = 1e-40, OMEGA_C = 1e-40, BAND_SIZE = 16
```

##Merge paired reads

```
mergers <- mergePairs(dadaFs, derepFs, dadaRs, derepRs, verbose=TRUE)
```

```
## 36601 paired-reads (in 884 unique pairings) successfully merged out of 37506 (in 1114 pairings) input
## 3244 paired-reads (in 153 unique pairings) successfully merged out of 3397 (in 182 pairings) input.
## 89998 paired-reads (in 413 unique pairings) successfully merged out of 99686 (in 605 pairings) input
## 46358 paired-reads (in 843 unique pairings) successfully merged out of 47662 (in 1109 pairings) input
## 23672 paired-reads (in 548 unique pairings) successfully merged out of 24731 (in 761 pairings) input
## 33020 paired-reads (in 827 unique pairings) successfully merged out of 34430 (in 1100 pairings) input
## 40451 paired-reads (in 1002 unique pairings) successfully merged out of 43814 (in 1628 pairings) input
## 6591 paired-reads (in 197 unique pairings) successfully merged out of 6763 (in 241 pairings) input.
## 17483 paired-reads (in 215 unique pairings) successfully merged out of 17708 (in 267 pairings) input
## 48990 paired-reads (in 555 unique pairings) successfully merged out of 52615 (in 734 pairings) input
## 32041 paired-reads (in 709 unique pairings) successfully merged out of 33289 (in 934 pairings) input
## 19894 paired-reads (in 568 unique pairings) successfully merged out of 20543 (in 676 pairings) input
## 1308 paired-reads (in 92 unique pairings) successfully merged out of 1340 (in 101 pairings) input.
## 9144 paired-reads (in 68 unique pairings) successfully merged out of 9165 (in 72 pairings) input.
## 17466 paired-reads (in 527 unique pairings) successfully merged out of 18070 (in 678 pairings) input
```

```
## 37751 paired-reads (in 700 unique pairings) successfully merged out of 39290 (in 996 pairings) input
## 34274 paired-reads (in 720 unique pairings) successfully merged out of 35380 (in 913 pairings) input
## 40076 paired-reads (in 582 unique pairings) successfully merged out of 41133 (in 808 pairings) input
## 50307 paired-reads (in 717 unique pairings) successfully merged out of 51772 (in 1046 pairings) inpu
## 8763 paired-reads (in 256 unique pairings) successfully merged out of 9167 (in 349 pairings) input.
## 43746 paired-reads (in 1222 unique pairings) successfully merged out of 46294 (in 1722 pairings) inp
## 9332 paired-reads (in 322 unique pairings) successfully merged out of 9753 (in 404 pairings) input.
## 71 paired-reads (in 2 unique pairings) successfully merged out of 71 (in 2 pairings) input.
## 1545 paired-reads (in 17 unique pairings) successfully merged out of 1554 (in 20 pairings) input.
## 185 paired-reads (in 9 unique pairings) successfully merged out of 213 (in 10 pairings) input.
## 75 paired-reads (in 4 unique pairings) successfully merged out of 75 (in 4 pairings) input.
## 129 paired-reads (in 5 unique pairings) successfully merged out of 129 (in 5 pairings) input.
## 352 paired-reads (in 19 unique pairings) successfully merged out of 352 (in 19 pairings) input.
## 37 paired-reads (in 4 unique pairings) successfully merged out of 37 (in 4 pairings) input.
## No paired-reads (in ZERO unique pairings) successfully merged out of 16 pairings) input.
## 121 paired-reads (in 8 unique pairings) successfully merged out of 121 (in 8 pairings) input.
## 264 paired-reads (in 3 unique pairings) successfully merged out of 264 (in 3 pairings) input.
## 75 paired-reads (in 3 unique pairings) successfully merged out of 75 (in 3 pairings) input.
## 287 paired-reads (in 6 unique pairings) successfully merged out of 287 (in 6 pairings) input.
## 18 paired-reads (in 2 unique pairings) successfully merged out of 18 (in 2 pairings) input.
## 2 paired-reads (in 1 unique pairings) successfully merged out of 2 (in 1 pairings) input.
## No paired-reads (in ZERO unique pairings) successfully merged out of 22 pairings) input.
## 4 paired-reads (in 1 unique pairings) successfully merged out of 4 (in 1 pairings) input.
## 50506 paired-reads (in 118 unique pairings) successfully merged out of 52708 (in 532 pairings) input
## 82813 paired-reads (in 155 unique pairings) successfully merged out of 85491 (in 631 pairings) input
## 32044 paired-reads (in 80 unique pairings) successfully merged out of 33750 (in 365 pairings) input.
## 47506 paired-reads (in 88 unique pairings) successfully merged out of 50230 (in 498 pairings) input.
```

##Construct the sequence table

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

```
## [1]   42 7948
```

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

```
##
##  230  231  232  233  234  235  236  237  238  239  240  242  243  244  245  246
##  878   19  179 6259  397   29   17   34    7    4    3    4    3    2    2    1
##  249  255  260  263  264  265  266  267  268  270  273  274  278  279  280  282
##    1    1    5    1    1    2    3    3    1    1    5    1    1    1    2    1
```

```
##  284  293  295  296  297  298  301  302  304  305  307  310  312  313  314  315
##    2    2    1    3    3    4    2    1    1    1    2    1    1    1    1    2
##  316  317  319  320  323  324  327  330  332  334  336  337  338  339  342  345
##    1    2    1    1    1    1    1    1    1    3    2    1    1    3    1    5
##  346  347  348  350  351  353  357  358  359  365  366  369  373  379  385  395
##    1    1    1    1    1    1    1    2    2    1    1    2    1    1    2    1
##  397  398  399  407
##    1    1    1    3
```

##Remove sequences that are too long or too short

```
seqtab <- seqtab[,nchar(colnames(seqtab)) %in% 230:237]
#check new sequence length
dim(seqtab)
```

```
## [1]   42 7812
```

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

```
##
##  230  231  232  233  234  235  236  237
##  878   19  179 6259  397   29   17   34
```

##Remove chimeras

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

```
## Identified 287 bimeras out of 7812 input sequences.
```

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

```
## [1]   42 7525
```

```
#Determine % chimeric abundance
sum(seqtab.nochim)/sum(seqtab)
```

```
## [1] 0.9634597
```

##Save seqtab.nochim as an R file

```
save(seqtab.nochim, file="../RData/seqtab.nochim.RData")
```

##Track reads through pipeline

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

```
##      input filtered denoisedF denoisedR merged nonchim
## 103  43007    38430     37808     37882  36601   35407
## 123   3984     3585      3452      3488   3244    3202
## 180 112134   100851    100044    100117  89998   88560
## 184  55020    48926     48163     48153  46358   44916
## 186  29577    25491     25041     25060  23672   22923
## 2    40953    35685     34840     34926  33020   31999
```

##load seqtab.nochim to start here

```
load("../RData/seqtab.nochim.RData")
```

##Assign taxonomy to seqtab.nochim

```
#Download taxonomy file from https://zenodo.org/record/4587955 and place it in working directory

#assign taxonomy. make sure file name corresponds with downloaded file
taxa <- assignTaxonomy(seqtab.nochim, "silva_nr99_v138.1_wSpecies_train_set.fa.gz", multithread=TRUE)

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

##Save taxa as an R file

```
save(taxa, file="../RData/taxa.RData")
```