FinalProject

Alejandra

2024-05-11

#load required packages

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

```
## Warning: package 'Biostrings' was built under R version 4.3.3
```

```
## Warning: package 'GenomeInfoDb' was built under R version 4.3.3
```

```
library(ShortRead)
library(phyloseq)
library(dplyr)
library(BiMiCo)
library(ggplot2)
library(devtools)
library(MicEco)
library(vegan)
```

#load sequences

```
path <- "Sequences"
list.files(path)
```

```
##   [1] "103_S309_L001_R1_001.fastq" "103_S309_L001_R2_001.fastq"
##   [3] "120_S5_L001_R1_001.fastq"   "120_S5_L001_R2_001.fastq"
##   [5] "123_S332_L001_R1_001.fastq" "123_S332_L001_R2_001.fastq"
##   [7] "150_S52_L001_R1_001.fastq"  "150_S52_L001_R2_001.fastq"
##   [9] "152_S68_L001_R1_001.fastq"  "152_S68_L001_R2_001.fastq"
##  [11] "156_S21_L001_R1_001.fastq"  "156_S21_L001_R2_001.fastq"
##  [13] "160_S78_L001_R1_001.fastq"  "160_S78_L001_R2_001.fastq"
##  [15] "169_S20_L001_R1_001.fastq"  "169_S20_L001_R2_001.fastq"
##  [17] "173_S88_L001_R1_001.fastq"  "173_S88_L001_R2_001.fastq"
##  [19] "178_S39_L001_R1_001.fastq"  "178_S39_L001_R2_001.fastq"
##  [21] "180_S346_L001_R1_001.fastq" "180_S346_L001_R2_001.fastq"
##  [23] "184_S298_L001_R1_001.fastq" "184_S298_L001_R2_001.fastq"
##  [25] "186_S370_L001_R1_001.fastq" "186_S370_L001_R2_001.fastq"
##  [27] "195_S18_L001_R1_001.fastq"  "195_S18_L001_R2_001.fastq"
##  [29] "198_S29_L001_R1_001.fastq"  "198_S29_L001_R2_001.fastq"
##  [31] "2_S297_L001_R1_001.fastq"   "2_S297_L001_R2_001.fastq"
##  [33] "20_S369_L001_R1_001.fastq"  "20_S369_L001_R2_001.fastq"
##  [35] "200_S62_L001_R1_001.fastq"  "200_S62_L001_R2_001.fastq"
##  [37] "203_S6_L001_R1_001.fastq"   "203_S6_L001_R2_001.fastq"
##  [39] "205_S9_L001_R1_001.fastq"   "205_S9_L001_R2_001.fastq"
##  [41] "207_S357_L001_R1_001.fastq" "207_S357_L001_R2_001.fastq"
##  [43] "214_S322_L001_R1_001.fastq" "214_S322_L001_R2_001.fastq"
```

```
##  [45] "216_S38_L001_R1_001.fastq"  "216_S38_L001_R2_001.fastq"
##  [47] "22_S358_L001_R1_001.fastq"  "22_S358_L001_R2_001.fastq"
##  [49] "237_S333_L001_R1_001.fastq" "237_S333_L001_R2_001.fastq"
##  [51] "238_S91_L001_R1_001.fastq"  "238_S91_L001_R2_001.fastq"
##  [53] "24_S8_L001_R1_001.fastq"    "24_S8_L001_R2_001.fastq"
##  [55] "246_S54_L001_R1_001.fastq"  "246_S54_L001_R2_001.fastq"
##  [57] "262_S320_L001_R1_001.fastq" "262_S320_L001_R2_001.fastq"
##  [59] "264_S56_L001_R1_001.fastq"  "264_S56_L001_R2_001.fastq"
##  [61] "268_S19_L001_R1_001.fastq"  "268_S19_L001_R2_001.fastq"
##  [63] "283_S381_L001_R1_001.fastq" "283_S381_L001_R2_001.fastq"
##  [65] "284_S15_L001_R1_001.fastq"  "284_S15_L001_R2_001.fastq"
##  [67] "287_S16_L001_R1_001.fastq"  "287_S16_L001_R2_001.fastq"
##  [69] "297_S31_L001_R1_001.fastq"  "297_S31_L001_R2_001.fastq"
##  [71] "298_S356_L001_R1_001.fastq" "298_S356_L001_R2_001.fastq"
##  [73] "31_S3_L001_R1_001.fastq"    "31_S3_L001_R2_001.fastq"
##  [75] "319_S310_L001_R1_001.fastq" "319_S310_L001_R2_001.fastq"
##  [77] "32_S79_L001_R1_001.fastq"   "32_S79_L001_R2_001.fastq"
##  [79] "326_S334_L001_R1_001.fastq" "326_S334_L001_R2_001.fastq"
##  [81] "327_S64_L001_R1_001.fastq"  "327_S64_L001_R2_001.fastq"
##  [83] "329_S382_L001_R1_001.fastq" "329_S382_L001_R2_001.fastq"
##  [85] "342_S77_L001_R1_001.fastq"  "342_S77_L001_R2_001.fastq"
##  [87] "346_S87_L001_R1_001.fastq"  "346_S87_L001_R2_001.fastq"
##  [89] "348_S30_L001_R1_001.fastq"  "348_S30_L001_R2_001.fastq"
##  [91] "35_S42_L001_R1_001.fastq"   "35_S42_L001_R2_001.fastq"
##  [93] "350_S41_L001_R1_001.fastq"  "350_S41_L001_R2_001.fastq"
##  [95] "352_S89_L001_R1_001.fastq"  "352_S89_L001_R2_001.fastq"
##  [97] "357_S7_L001_R1_001.fastq"   "357_S7_L001_R2_001.fastq"
##  [99] "363_S43_L001_R1_001.fastq"  "363_S43_L001_R2_001.fastq"
## [101] "365_S32_L001_R1_001.fastq"  "365_S32_L001_R2_001.fastq"
## [103] "371_S66_L001_R1_001.fastq"  "371_S66_L001_R2_001.fastq"
## [105] "372_S345_L001_R1_001.fastq" "372_S345_L001_R2_001.fastq"
## [107] "373_S92_L001_R1_001.fastq"  "373_S92_L001_R2_001.fastq"
## [109] "377_S321_L001_R1_001.fastq" "377_S321_L001_R2_001.fastq"
## [111] "39_S368_L001_R1_001.fastq"  "39_S368_L001_R2_001.fastq"
## [113] "42_S53_L001_R1_001.fastq"   "42_S53_L001_R2_001.fastq"
## [115] "48_S55_L001_R1_001.fastq"   "48_S55_L001_R2_001.fastq"
## [117] "54_S90_L001_R1_001.fastq"   "54_S90_L001_R2_001.fastq"
## [119] "55_S26_L001_R1_001.fastq"   "55_S26_L001_R2_001.fastq"
## [121] "63_S80_L001_R1_001.fastq"   "63_S80_L001_R2_001.fastq"
## [123] "66_S51_L001_R1_001.fastq"   "66_S51_L001_R2_001.fastq"
## [125] "70_S74_L001_R1_001.fastq"   "70_S74_L001_R2_001.fastq"
## [127] "77_S44_L001_R1_001.fastq"   "77_S44_L001_R2_001.fastq"
## [129] "8_S344_L001_R1_001.fastq"   "8_S344_L001_R2_001.fastq"
## [131] "87_S75_L001_R1_001.fastq"   "87_S75_L001_R2_001.fastq"
## [133] "88_S28_L001_R1_001.fastq"   "88_S28_L001_R2_001.fastq"
## [135] "93_S380_L001_R1_001.fastq"  "93_S380_L001_R2_001.fastq"
## [137] "95_S65_L001_R1_001.fastq"   "95_S65_L001_R2_001.fastq"
## [139] "97_S67_L001_R1_001.fastq"   "97_S67_L001_R2_001.fastq"
## [141] "filtered"
```

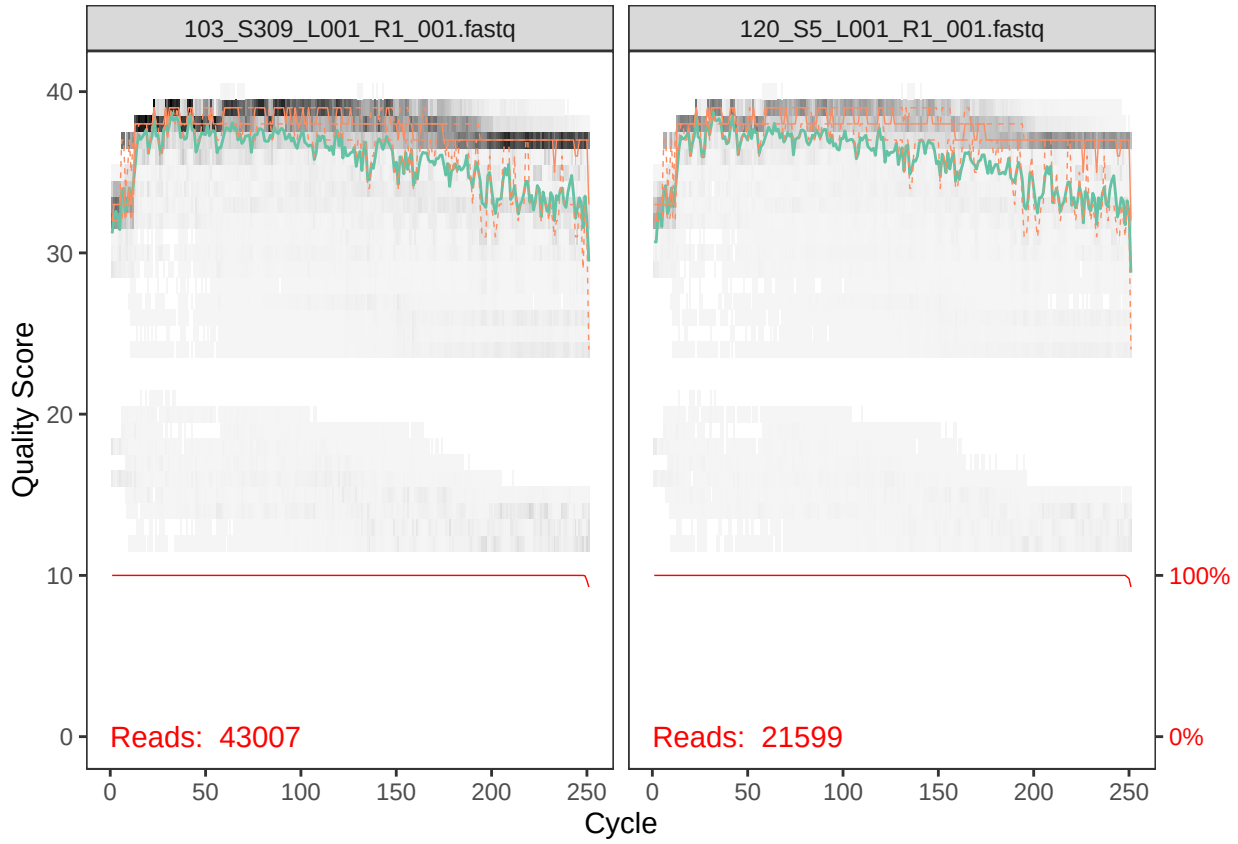
#read file names

```
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 file names
```

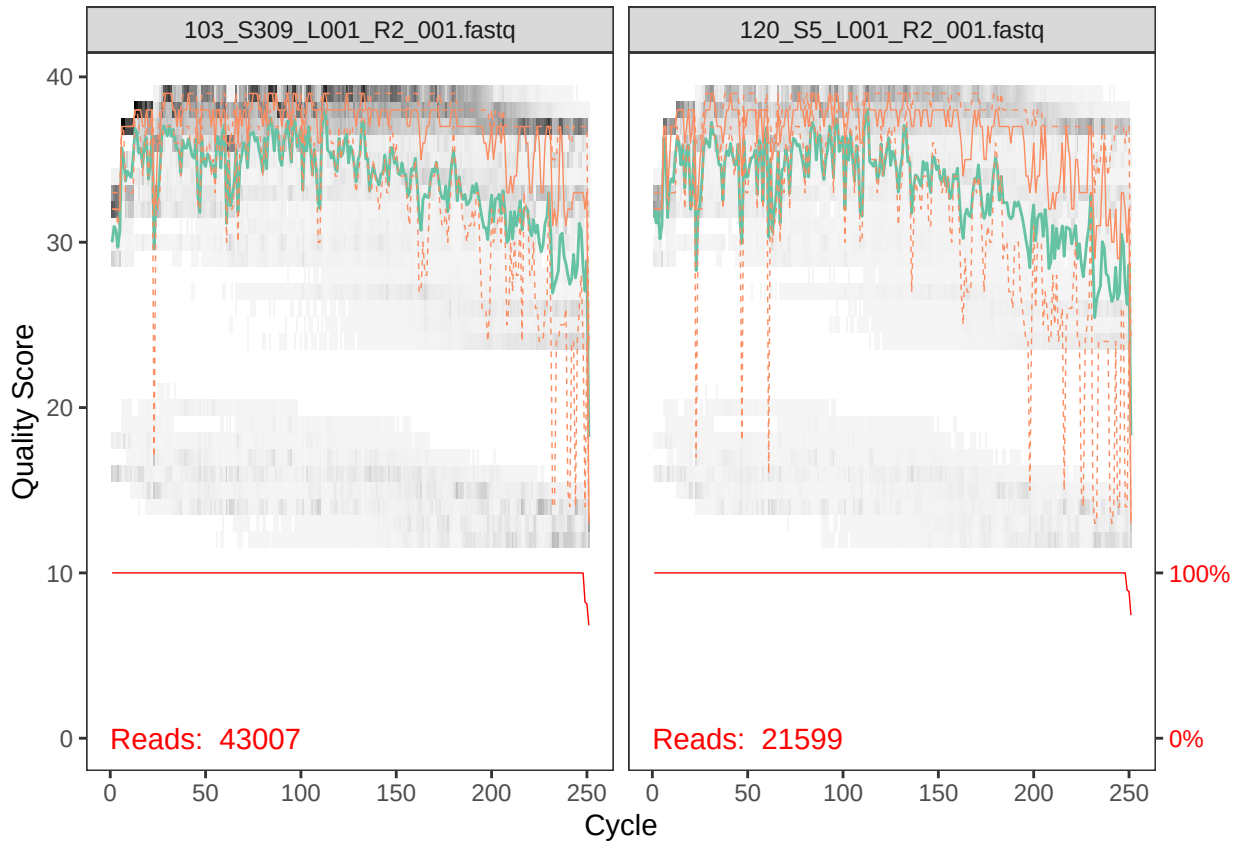
```
sample.names <- sapply(strsplit(basename(fnFs), "_"), `[`, 1)
```

#inspect file quality of forward and reverse reads

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




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



#filter and trim

```
#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
out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs, truncLen=c(220,220),
              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     37394
## 120_S5_L001_R1_001.fastq      21599     18124
## 123_S332_L001_R1_001.fastq     3984      3467
## 150_S52_L001_R1_001.fastq      6944      6040
## 152_S68_L001_R1_001.fastq     11697      9959
## 156_S21_L001_R1_001.fastq     51180     45398
```

#learn error rates of reads

```
##learn error rates of forward and reverse reads
errF <- learnErrors(filtFs, multithread=TRUE)
```

```
## 107390580 total bases in 488139 reads from 17 samples will be used for learning the error rates.
```

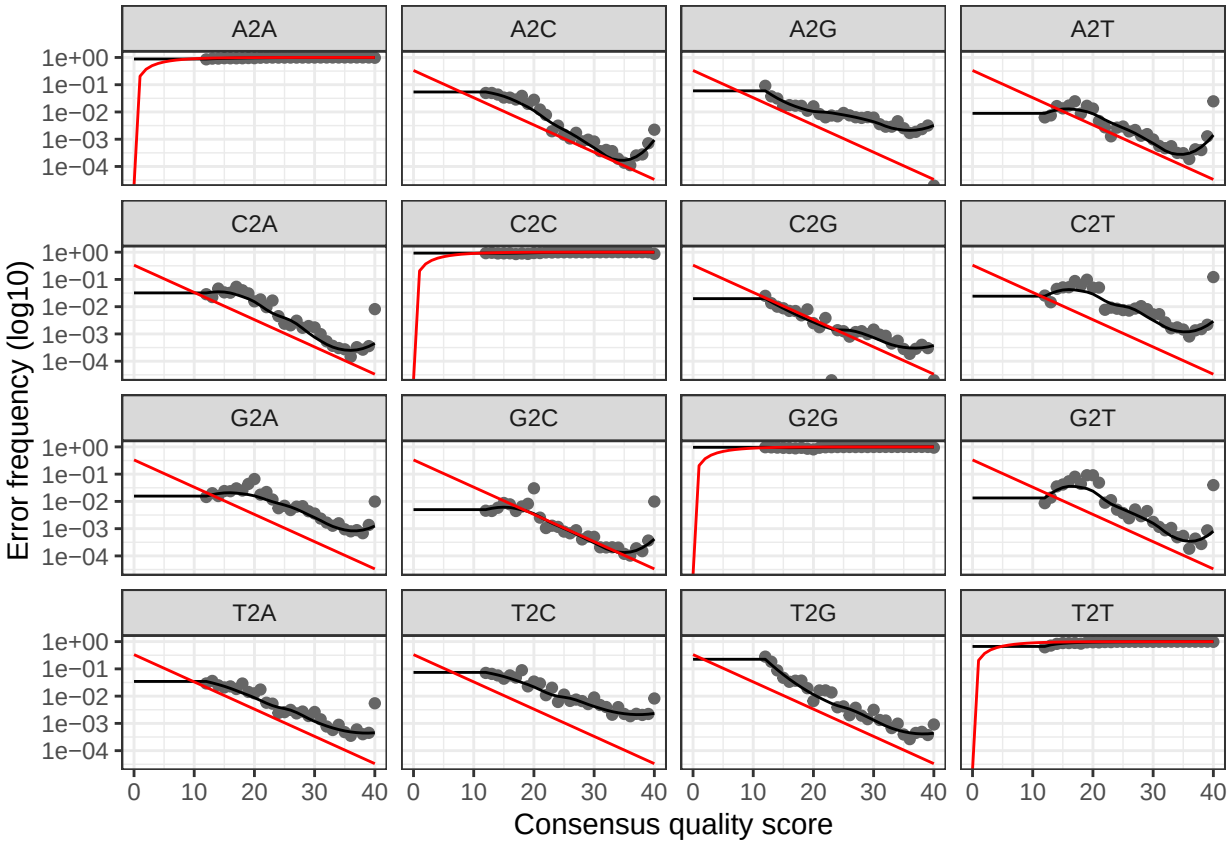
```
errR <- learnErrors(filtRs, multithread=TRUE)
```

```
## 107390580 total bases in 488139 reads from 17 samples will be used for learning the error rates.
```

#visulaize error rate

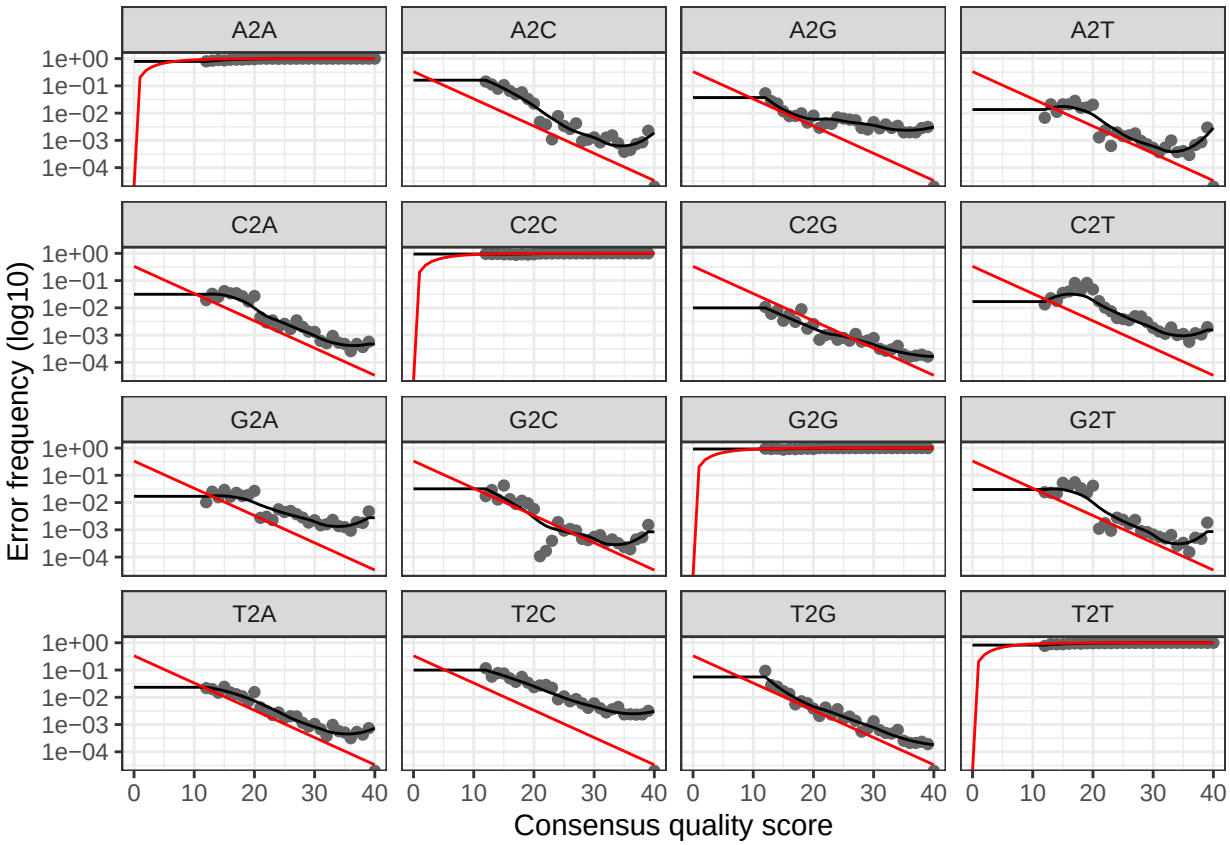
```
plotErrors(errF, nominalQ=TRUE)
```

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




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

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



#will take reads and show how many sequences/species are in the sample

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

```
## Sample 1 - 37394 reads in 11127 unique sequences.
## Sample 2 - 18124 reads in 5378 unique sequences.
## Sample 3 - 3467 reads in 1233 unique sequences.
## Sample 4 - 6040 reads in 1466 unique sequences.
## Sample 5 - 9959 reads in 2863 unique sequences.
## Sample 6 - 45398 reads in 9580 unique sequences.
## Sample 7 - 36254 reads in 5768 unique sequences.
## Sample 8 - 21241 reads in 6434 unique sequences.
## Sample 9 - 38528 reads in 8860 unique sequences.
## Sample 10 - 13949 reads in 4485 unique sequences.
## Sample 11 - 96304 reads in 10949 unique sequences.
## Sample 12 - 47143 reads in 12715 unique sequences.
## Sample 13 - 23938 reads in 6641 unique sequences.
## Sample 14 - 2376 reads in 887 unique sequences.
## Sample 15 - 10576 reads in 3106 unique sequences.
## Sample 16 - 34833 reads in 11128 unique sequences.
## Sample 17 - 42615 reads in 11774 unique sequences.
## Sample 18 - 20858 reads in 5396 unique sequences.
## Sample 19 - 2122 reads in 761 unique sequences.
## Sample 20 - 42204 reads in 12898 unique sequences.
## Sample 21 - 6873 reads in 2145 unique sequences.
## Sample 22 - 17590 reads in 3620 unique sequences.
## Sample 23 - 195 reads in 86 unique sequences.
```

```
## Sample 24 - 51451 reads in 8937 unique sequences.
## Sample 25 - 33123 reads in 9679 unique sequences.
## Sample 26 - 846 reads in 276 unique sequences.
## Sample 27 - 40717 reads in 10711 unique sequences.
## Sample 28 - 28265 reads in 7015 unique sequences.
## Sample 29 - 20340 reads in 4893 unique sequences.
## Sample 30 - 14277 reads in 4485 unique sequences.
## Sample 31 - 750 reads in 305 unique sequences.
## Sample 32 - 1404 reads in 553 unique sequences.
## Sample 33 - 42084 reads in 10080 unique sequences.
## Sample 34 - 41278 reads in 14515 unique sequences.
## Sample 35 - 398 reads in 142 unique sequences.
## Sample 36 - 9046 reads in 1263 unique sequences.
## Sample 37 - 76834 reads in 9219 unique sequences.
## Sample 38 - 18193 reads in 5709 unique sequences.
## Sample 39 - 21538 reads in 6134 unique sequences.
## Sample 40 - 39491 reads in 9460 unique sequences.
## Sample 41 - 9399 reads in 2872 unique sequences.
## Sample 42 - 35074 reads in 9536 unique sequences.
## Sample 43 - 26239 reads in 6816 unique sequences.
## Sample 44 - 19514 reads in 5376 unique sequences.
## Sample 45 - 28000 reads in 7623 unique sequences.
## Sample 46 - 8583 reads in 2972 unique sequences.
## Sample 47 - 26943 reads in 8642 unique sequences.
## Sample 48 - 1904 reads in 754 unique sequences.
## Sample 49 - 43654 reads in 11347 unique sequences.
## Sample 50 - 1119 reads in 456 unique sequences.
## Sample 51 - 55236 reads in 19987 unique sequences.
## Sample 52 - 2714 reads in 1048 unique sequences.
## Sample 53 - 41159 reads in 8734 unique sequences.
## Sample 54 - 35252 reads in 10990 unique sequences.
## Sample 55 - 51946 reads in 10472 unique sequences.
## Sample 56 - 8867 reads in 2654 unique sequences.
## Sample 57 - 16711 reads in 5007 unique sequences.
## Sample 58 - 23952 reads in 7003 unique sequences.
## Sample 59 - 26869 reads in 5282 unique sequences.
## Sample 60 - 45370 reads in 14713 unique sequences.
## Sample 61 - 37525 reads in 12923 unique sequences.
## Sample 62 - 18588 reads in 5637 unique sequences.
## Sample 63 - 22286 reads in 5956 unique sequences.
## Sample 64 - 42490 reads in 14903 unique sequences.
## Sample 65 - 47048 reads in 13825 unique sequences.
## Sample 66 - 16150 reads in 4697 unique sequences.
## Sample 67 - 19661 reads in 6082 unique sequences.
## Sample 68 - 9663 reads in 3109 unique sequences.
## Sample 69 - 2077 reads in 746 unique sequences.
## Sample 70 - 4788 reads in 1572 unique sequences.
```

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

```
## Sample 1 - 37394 reads in 15434 unique sequences.
## Sample 2 - 18124 reads in 8388 unique sequences.
## Sample 3 - 3467 reads in 1659 unique sequences.
## Sample 4 - 6040 reads in 2168 unique sequences.
## Sample 5 - 9959 reads in 4152 unique sequences.
```

```
## Sample 6 - 45398 reads in 15078 unique sequences.
## Sample 7 - 36254 reads in 12918 unique sequences.
## Sample 8 - 21241 reads in 9098 unique sequences.
## Sample 9 - 38528 reads in 13857 unique sequences.
## Sample 10 - 13949 reads in 6543 unique sequences.
## Sample 11 - 96304 reads in 23613 unique sequences.
## Sample 12 - 47143 reads in 18858 unique sequences.
## Sample 13 - 23938 reads in 11928 unique sequences.
## Sample 14 - 2376 reads in 1268 unique sequences.
## Sample 15 - 10576 reads in 5003 unique sequences.
## Sample 16 - 34833 reads in 15163 unique sequences.
## Sample 17 - 42615 reads in 21576 unique sequences.
## Sample 18 - 20858 reads in 8304 unique sequences.
## Sample 19 - 2122 reads in 1104 unique sequences.
## Sample 20 - 42204 reads in 18205 unique sequences.
## Sample 21 - 6873 reads in 2991 unique sequences.
## Sample 22 - 17590 reads in 5778 unique sequences.
## Sample 23 - 195 reads in 113 unique sequences.
## Sample 24 - 51451 reads in 15339 unique sequences.
## Sample 25 - 33123 reads in 14444 unique sequences.
## Sample 26 - 846 reads in 391 unique sequences.
## Sample 27 - 40717 reads in 14255 unique sequences.
## Sample 28 - 28265 reads in 10804 unique sequences.
## Sample 29 - 20340 reads in 8031 unique sequences.
## Sample 30 - 14277 reads in 5860 unique sequences.
## Sample 31 - 750 reads in 424 unique sequences.
## Sample 32 - 1404 reads in 756 unique sequences.
## Sample 33 - 42084 reads in 15869 unique sequences.
## Sample 34 - 41278 reads in 19533 unique sequences.
## Sample 35 - 398 reads in 202 unique sequences.
## Sample 36 - 9046 reads in 2625 unique sequences.
## Sample 37 - 76834 reads in 19413 unique sequences.
## Sample 38 - 18193 reads in 7839 unique sequences.
## Sample 39 - 21538 reads in 11353 unique sequences.
## Sample 40 - 39491 reads in 14042 unique sequences.
## Sample 41 - 9399 reads in 4195 unique sequences.
## Sample 42 - 35074 reads in 14099 unique sequences.
## Sample 43 - 26239 reads in 12690 unique sequences.
## Sample 44 - 19514 reads in 7905 unique sequences.
## Sample 45 - 28000 reads in 12185 unique sequences.
## Sample 46 - 8583 reads in 4405 unique sequences.
## Sample 47 - 26943 reads in 13159 unique sequences.
## Sample 48 - 1904 reads in 1032 unique sequences.
## Sample 49 - 43654 reads in 17311 unique sequences.
## Sample 50 - 1119 reads in 611 unique sequences.
## Sample 51 - 55236 reads in 27448 unique sequences.
## Sample 52 - 2714 reads in 1259 unique sequences.
## Sample 53 - 41159 reads in 12825 unique sequences.
## Sample 54 - 35252 reads in 16195 unique sequences.
## Sample 55 - 51946 reads in 16203 unique sequences.
## Sample 56 - 8867 reads in 4796 unique sequences.
## Sample 57 - 16711 reads in 7367 unique sequences.
## Sample 58 - 23952 reads in 9575 unique sequences.
## Sample 59 - 26869 reads in 8507 unique sequences.
```

```
## Sample 60 - 45370 reads in 21203 unique sequences.
## Sample 61 - 37525 reads in 21414 unique sequences.
## Sample 62 - 18588 reads in 7608 unique sequences.
## Sample 63 - 22286 reads in 11057 unique sequences.
## Sample 64 - 42490 reads in 21276 unique sequences.
## Sample 65 - 47048 reads in 19120 unique sequences.
## Sample 66 - 16150 reads in 7968 unique sequences.
## Sample 67 - 19661 reads in 8742 unique sequences.
## Sample 68 - 9663 reads in 4534 unique sequences.
## Sample 69 - 2077 reads in 1118 unique sequences.
## Sample 70 - 4788 reads in 2186 unique sequences.
```

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

```
## dada-class: object describing DADA2 denoising results
## 964 sequence variants were inferred from 11127 input unique sequences.
## Key parameters: OMEGA_A = 1e-40, OMEGA_C = 1e-40, BAND_SIZE = 16
```

#merge paired reads

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

```
## 35572 paired-reads (in 859 unique pairings) successfully merged out of 36566 (in 1089 pairings) input
## 16909 paired-reads (in 476 unique pairings) successfully merged out of 17638 (in 615 pairings) input
## 3193 paired-reads (in 155 unique pairings) successfully merged out of 3306 (in 178 pairings) input.
## 5824 paired-reads (in 152 unique pairings) successfully merged out of 5890 (in 167 pairings) input.
## 9506 paired-reads (in 303 unique pairings) successfully merged out of 9746 (in 361 pairings) input.
## 43190 paired-reads (in 808 unique pairings) successfully merged out of 44667 (in 987 pairings) input
## 34742 paired-reads (in 176 unique pairings) successfully merged out of 35407 (in 324 pairings) input
## 19928 paired-reads (in 602 unique pairings) successfully merged out of 20687 (in 758 pairings) input
## 36264 paired-reads (in 660 unique pairings) successfully merged out of 37595 (in 901 pairings) input
## 12870 paired-reads (in 442 unique pairings) successfully merged out of 13517 (in 556 pairings) input
## 86963 paired-reads (in 432 unique pairings) successfully merged out of 95163 (in 632 pairings) input
## 44519 paired-reads (in 821 unique pairings) successfully merged out of 45857 (in 1091 pairings) input
## 22384 paired-reads (in 535 unique pairings) successfully merged out of 23275 (in 706 pairings) input
## 2057 paired-reads (in 124 unique pairings) successfully merged out of 2176 (in 149 pairings) input.
## 9684 paired-reads (in 380 unique pairings) successfully merged out of 10190 (in 458 pairings) input.
## 32182 paired-reads (in 823 unique pairings) successfully merged out of 33518 (in 1088 pairings) input
## 38012 paired-reads (in 976 unique pairings) successfully merged out of 40937 (in 1539 pairings) input
## 19719 paired-reads (in 439 unique pairings) successfully merged out of 20392 (in 560 pairings) input
## 1909 paired-reads (in 117 unique pairings) successfully merged out of 2009 (in 128 pairings) input.
## 37378 paired-reads (in 1127 unique pairings) successfully merged out of 40327 (in 1601 pairings) input
## 6398 paired-reads (in 195 unique pairings) successfully merged out of 6622 (in 242 pairings) input.
## 17127 paired-reads (in 215 unique pairings) successfully merged out of 17362 (in 259 pairings) input
## 156 paired-reads (in 14 unique pairings) successfully merged out of 166 (in 16 pairings) input.
```

```
## 49408 paired-reads (in 560 unique pairings) successfully merged out of 50340 (in 739 pairings) input
## 31066 paired-reads (in 689 unique pairings) successfully merged out of 32254 (in 897 pairings) input
## 828 paired-reads (in 27 unique pairings) successfully merged out of 828 (in 27 pairings) input.
## 38135 paired-reads (in 706 unique pairings) successfully merged out of 39532 (in 999 pairings) input
## 26812 paired-reads (in 521 unique pairings) successfully merged out of 27514 (in 677 pairings) input
## 19316 paired-reads (in 548 unique pairings) successfully merged out of 19896 (in 643 pairings) input
## 13537 paired-reads (in 405 unique pairings) successfully merged out of 13898 (in 493 pairings) input
## 636 paired-reads (in 51 unique pairings) successfully merged out of 653 (in 54 pairings) input.
## 1262 paired-reads (in 92 unique pairings) successfully merged out of 1296 (in 104 pairings) input.
## 40197 paired-reads (in 720 unique pairings) successfully merged out of 41315 (in 945 pairings) input
## 35897 paired-reads (in 1420 unique pairings) successfully merged out of 38850 (in 2023 pairings) inpu
## 356 paired-reads (in 20 unique pairings) successfully merged out of 368 (in 23 pairings) input.
## 8912 paired-reads (in 49 unique pairings) successfully merged out of 8934 (in 53 pairings) input.
## 68331 paired-reads (in 507 unique pairings) successfully merged out of 75395 (in 815 pairings) input
## 16819 paired-reads (in 522 unique pairings) successfully merged out of 17667 (in 675 pairings) input
## 19365 paired-reads (in 517 unique pairings) successfully merged out of 20779 (in 719 pairings) input
## 36430 paired-reads (in 689 unique pairings) successfully merged out of 38172 (in 991 pairings) input
## 8724 paired-reads (in 302 unique pairings) successfully merged out of 9070 (in 368 pairings) input.
## 33416 paired-reads (in 715 unique pairings) successfully merged out of 34438 (in 902 pairings) input
## 23973 paired-reads (in 543 unique pairings) successfully merged out of 25291 (in 825 pairings) input
## 18657 paired-reads (in 424 unique pairings) successfully merged out of 19185 (in 509 pairings) input
## 25798 paired-reads (in 779 unique pairings) successfully merged out of 27103 (in 1045 pairings) input
## 7708 paired-reads (in 347 unique pairings) successfully merged out of 8153 (in 450 pairings) input.
## 24293 paired-reads (in 886 unique pairings) successfully merged out of 25836 (in 1175 pairings) input
## 1479 paired-reads (in 71 unique pairings) successfully merged out of 1648 (in 98 pairings) input.
## 41533 paired-reads (in 845 unique pairings) successfully merged out of 42850 (in 1035 pairings) input
## 966 paired-reads (in 59 unique pairings) successfully merged out of 1020 (in 75 pairings) input.
## 48018 paired-reads (in 1644 unique pairings) successfully merged out of 52205 (in 2461 pairings) inpu
## 2537 paired-reads (in 95 unique pairings) successfully merged out of 2601 (in 115 pairings) input.
## 38957 paired-reads (in 572 unique pairings) successfully merged out of 40219 (in 786 pairings) input
## 33025 paired-reads (in 776 unique pairings) successfully merged out of 34386 (in 1024 pairings) input
## 48895 paired-reads (in 709 unique pairings) successfully merged out of 50345 (in 1030 pairings) input
## 8039 paired-reads (in 249 unique pairings) successfully merged out of 8375 (in 334 pairings) input.
## 15272 paired-reads (in 549 unique pairings) successfully merged out of 16088 (in 712 pairings) input
## 22608 paired-reads (in 552 unique pairings) successfully merged out of 23379 (in 692 pairings) input
## 25800 paired-reads (in 273 unique pairings) successfully merged out of 26280 (in 376 pairings) input
```

```
## 40661 paired-reads (in 1434 unique pairings) successfully merged out of 43266 (in 1952 pairings) inpu
## 31746 paired-reads (in 1066 unique pairings) successfully merged out of 35359 (in 1797 pairings) inpu
## 17488 paired-reads (in 511 unique pairings) successfully merged out of 18095 (in 635 pairings) input
## 20858 paired-reads (in 443 unique pairings) successfully merged out of 21790 (in 594 pairings) input
## 37463 paired-reads (in 1275 unique pairings) successfully merged out of 40250 (in 1874 pairings) inpu
## 42749 paired-reads (in 1223 unique pairings) successfully merged out of 45210 (in 1722 pairings) inpu
## 15149 paired-reads (in 396 unique pairings) successfully merged out of 15690 (in 502 pairings) input
## 18394 paired-reads (in 591 unique pairings) successfully merged out of 19103 (in 736 pairings) input
## 8903 paired-reads (in 309 unique pairings) successfully merged out of 9346 (in 393 pairings) input.
## 1811 paired-reads (in 95 unique pairings) successfully merged out of 1964 (in 112 pairings) input.
## 4386 paired-reads (in 191 unique pairings) successfully merged out of 4584 (in 232 pairings) input.
```

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

```
##
## 1 TACAGAGGATGCAAGCGTTATCCGGAATGATTGGGCGTAAAGCGTCTGTAGGTGGCTTTTTAAGTCCGCCGTCAAATCCCAGGGCTCAACTCTGGACAG
## 2 TACGAAGGGGGCTAGCGTTGTTCGGATTTACTGGGCGTAAAGCGCGCGTAGGCGGGTAAATAAGTTAGAGGTGAAATCCCAGGGCTCAACCCTGGAACT
## 3 TACGAAGGGGGCTAGCGTTGTTCGGATTTACTGGGCGTAAAGCGCGCGTAGGCGGATATTTAAGTCAGAGGTGAAATCCCAGGGCTCAACCCTGGAACT
## 4                                 TACGTAAAAGACTAGTGTTAGTCATCTTTATTAGGTTTAAAGGGTACCTAGACGGTAAATTAAACTG
## 5 TACGGAGGGAGCTAGCGTTATTCGGAATTACTGGGCGTAAAGCGCACGTAGGCGGCTTTGTAAGTAAGAGGTGAAAGCCCAGAGCTCAACTCTGGAATT
## 6 GACAGAGGATGCAAGCGTTATCCGGAATGATTGGGCGTAAAGCGTCTGTAGGTGGCTTTTTAAGTCTGCCGTTAAATCCCAGGGCTTAACCCTGGATAA
##   abundance forward reverse nmatch nmismatch nindel prefer accept
## 1      2698       1       1    187         0      0      1   TRUE
## 2      1791       3       2    187         0      0      1   TRUE
## 3      1763       2       3    187         0      0      1   TRUE
## 4      1244       4       4    219         0      0      1   TRUE
## 5       742       5       5    187         0      0      1   TRUE
## 6       429       6       6    187         0      0      1   TRUE
```

#construct sequence table to see how many sequences are present and length

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

```
## [1]    70 17824
```

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

```
##
##   220   221   222   223   224   225   226   227   228   229   230   231   232
##    68   120    84   272    91    29    41   161    25    32     8     9     7
##   233   234   235   236   237   238   239   240   241   242   243   244   245
##     5    12    13     6    48     6     9     3     5     1     4    57    10
##   246   247   248   249   250   251   252   253   254   255   256   257   258
##     2     5     8     7    16    34   412 14786  1078    76    27    33     8
##   259   260   261   262   263   264   265   266   267   268   270   272   274
##     5    18     1     5     4     3     7     2     4     1     1     3     1
##   275   279   280   282   283   284   285   286   287   288   290   293   294
##     4     1     1     1     2     2     3     6     3     1     1     5     1
##   298   299   300   302   303   304   311   313   315   316   317   318   319
```

```
##     1     1     2     2     1     6     1     3     2     4     4     4     4
##   321   322   324   325   327   328   330   332   333   334   335   336   337
##     2     2     1     1     5     1     1     1     1     1     5     1     2
##   339   340   343   344   347   349   350   352   354   355   356   357   358
##     1     2     1     1     1     1     2     1     3     1     2     1     2
##   359   362   365   367   368   370   371   373   377   378   379   385   386
##     4     1     5     1     1     2     1     1     1     2     2     1     1
##   389   390   393   399   403   405   414   415   417   418   423   427
##     2     1     2     1     1     3     2     1     1     1     1     3
```

#remove chimeras (two sperate reads that got smashed together)

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

```
## Identified 96 bimeras out of 17824 input sequences.
```

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

```
## [1]    70 17728
```

```
sum(seqtab.nochim)/sum(seqtab)
```

```
## [1] 0.9969481
```

#track reads (which step lost reads)

```
getN <- function(x) sum(getUniques(x))
track <- cbind(out, sapply(dadaFs, getN), sapply(dadaRs, getN), sapply(mergers, getN), rowSums(seqtab.n
colnames(track) <- c("input", "filtered", "denoisedF", "denoisedR", "merged", "nonchim")
rownames(track) <- sample.names
head(track)
```

```
##     input filtered denoisedF denoisedR merged nonchim
## 103 43007    37394     36900     36857  35572   35572
## 120 21599    18124     17890     17785  16909   16860
## 123  3984     3467      3376      3358   3193    3193
## 150  6944     6040      5934      5942   5824    5824
## 152 11697     9959      9835      9826   9506    9506
## 156 51180    45398     44958     45013  43190   43108
```

#save setab.nochim as an R file

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

#load seqtab.nochim

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

#asign taxonomy

```
taxa <- assignTaxonomy(seqtab.nochim, "silva_nr99_v138.1_wSpecies_train_set.fa.gz", multithread=TRUE)
```

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

#load taxa and seqtab.nochim

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

#import metadata

```
metadata <- read.csv("metadatanocontrol.csv", header=TRUE, row.names = 1)
```

#create physeq object

```
physeq <- phyloseq(otu_table(seqtab.nochim, taxa_are_rows = FALSE),
                   sample_data(metadata),
                   tax_table(taxa))
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 17728 taxa and 70 samples ]
## sample_data() Sample Data:       [ 70 samples by 5 sample variables ]
## tax_table()   Taxonomy Table:    [ 17728 taxa by 7 taxonomic ranks ]
```

```
#convert from raw to abundance so its easier to comapre
physeq <- transform_sample_counts(physeq, function(abund) 1*(abund>0))

#visualize to data
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 17728 taxa and 70 samples ]
## sample_data() Sample Data:       [ 70 samples by 5 sample variables ]
## tax_table()   Taxonomy Table:    [ 17728 taxa by 7 taxonomic ranks ]
```

#remove the sequence itselt and replace with ASV

```
##this allows it to be easier to read, replaces the raw data
dna <- Biostrings::DNAStringSet(taxa_names(physeq))
names(dna) <- taxa_names(physeq)
physeq <- merge_phyloseq(physeq, dna)
taxa_names(physeq) <- paste0("ASV", seq(ntaxa(physeq)))
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 17728 taxa and 70 samples ]
## sample_data() Sample Data:       [ 70 samples by 5 sample variables ]
## tax_table()   Taxonomy Table:    [ 17728 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 17728 reference sequences ]
```

#remove mitochondria and phloroplast mathces, remove all non bacterial sequences

```
#stictly use bacteria 16S rRNA,
physeq <- physeq %>% subset_taxa( Family!= "Mitochondria" | is.na(Family) & Order!="Chloroplast" | is.na
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 15995 taxa and 70 samples ]
## sample_data() Sample Data:       [ 70 samples by 5 sample variables ]
## tax_table()   Taxonomy Table:    [ 15995 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 15995 reference sequences ]
```

#remove all non bacterial sequences

```
physeq<-rm_nonbac(physeq)
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 15951 taxa and 70 samples ]
```

```
## sample_data() Sample Data:       [ 70 samples by 5 sample variables ]
## tax_table()   Taxonomy Table:    [ 15951 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 15951 reference sequences ]
```

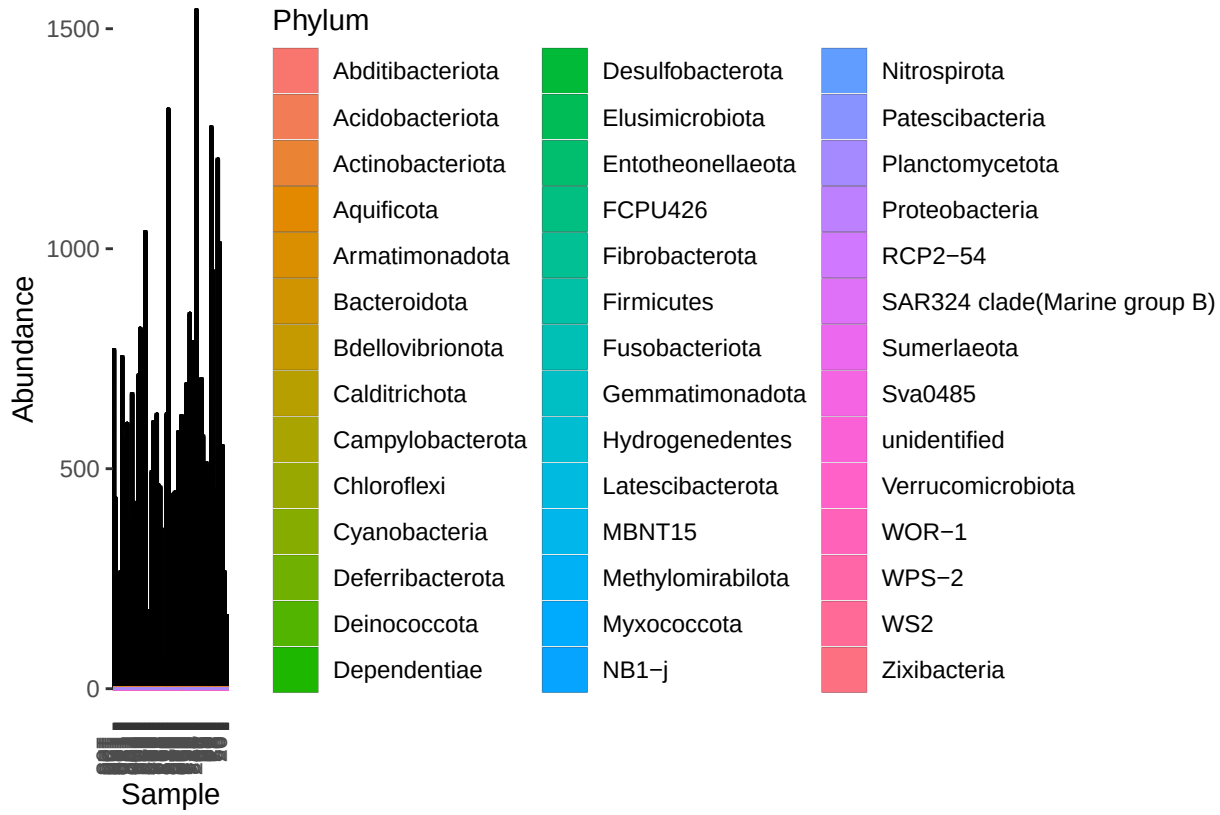
#save physeq objects and load

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

```
load("RData/physeq.RData")
```

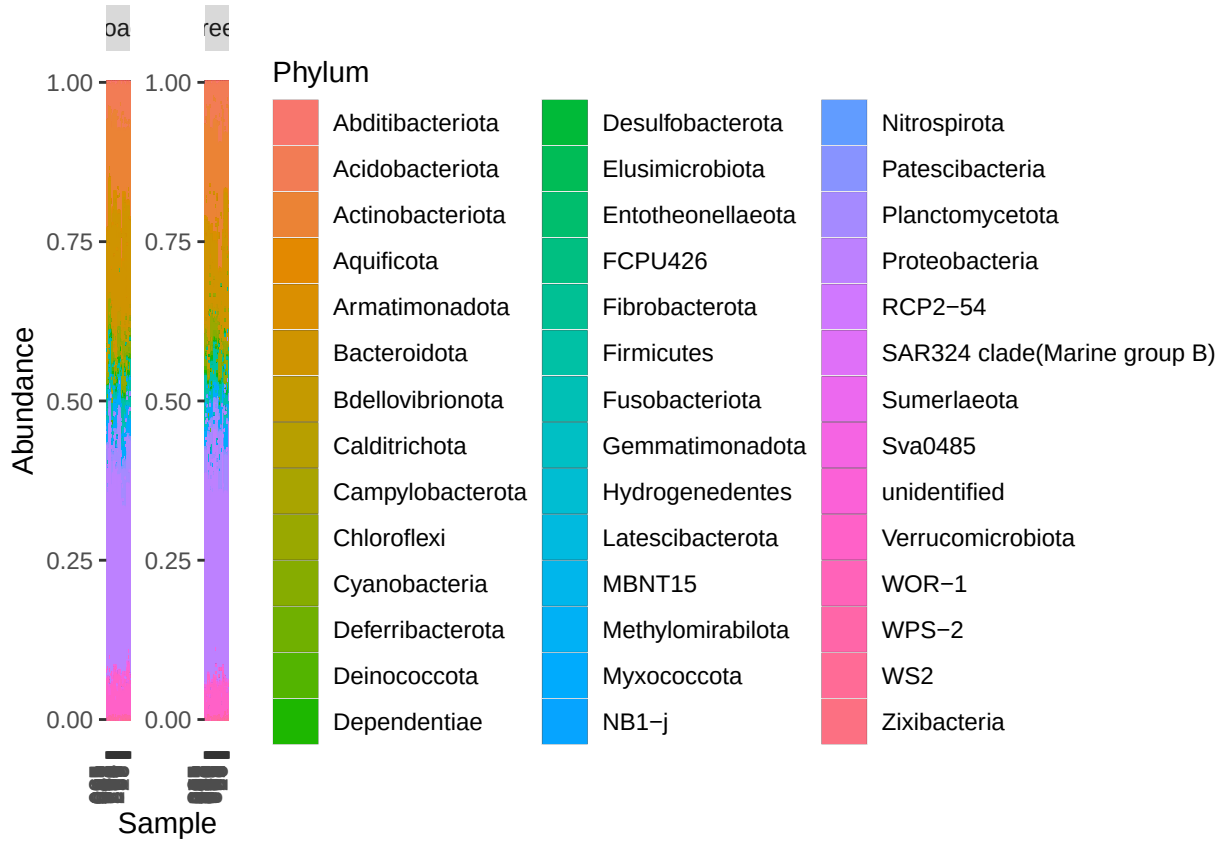
#plot bar grpah based on phylum

```
plot_bar(physeq, fill = "Phylum") + geom_bar(aes(color=Phylum, fill=Phylum), stat="identity", position="
```



#create a barplot of relative abundance

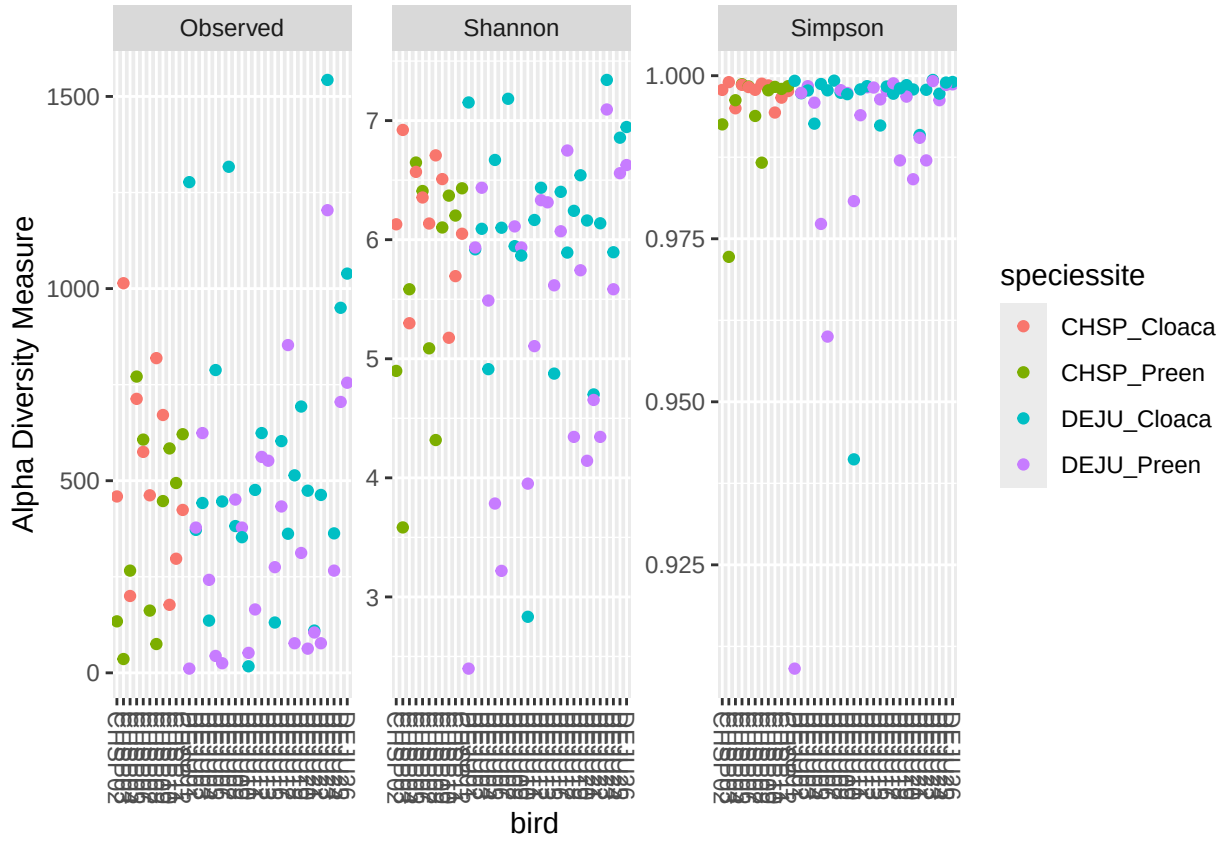
```
#convert to relative abundance
physeq_relabund <- transform_sample_counts(physeq, function(x) x / sum(x))

#barplot
plot_bar(physeq_relabund, fill = "Phylum") + geom_bar(aes(color=Phylum, fill=Phylum), stat="identity", p
```




```
##can change based on the column name in metadata in facet_wrap(~columnName)
```

#plot alpha diversity based on bird

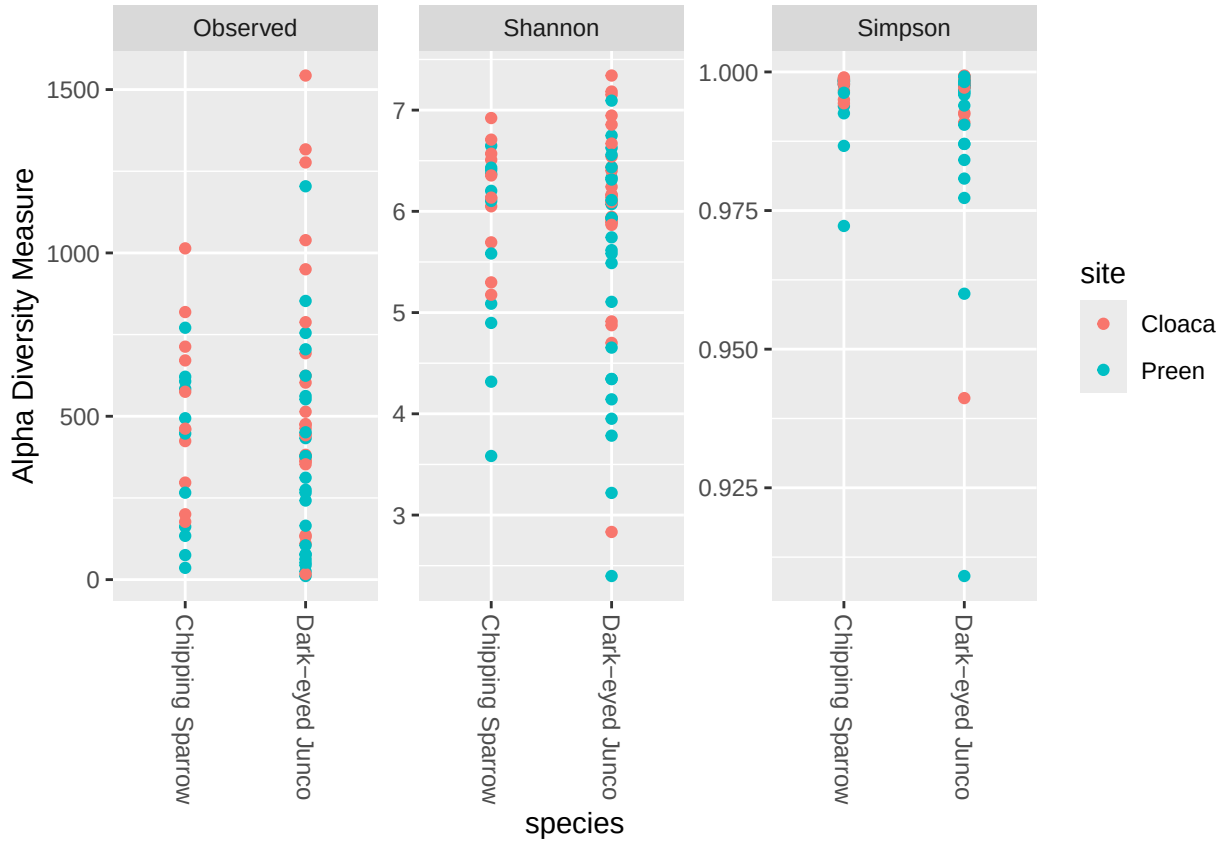
```
plot_richness(physeq, x="bird", color= "speciessite", measures=c("Observed", "Simpson", "Shannon"))
```




```
##Simpson(less sensitive, will be more custered together) and Shannon(more sensitive to rare taxa) take
```

#plot alpha diversity based on site

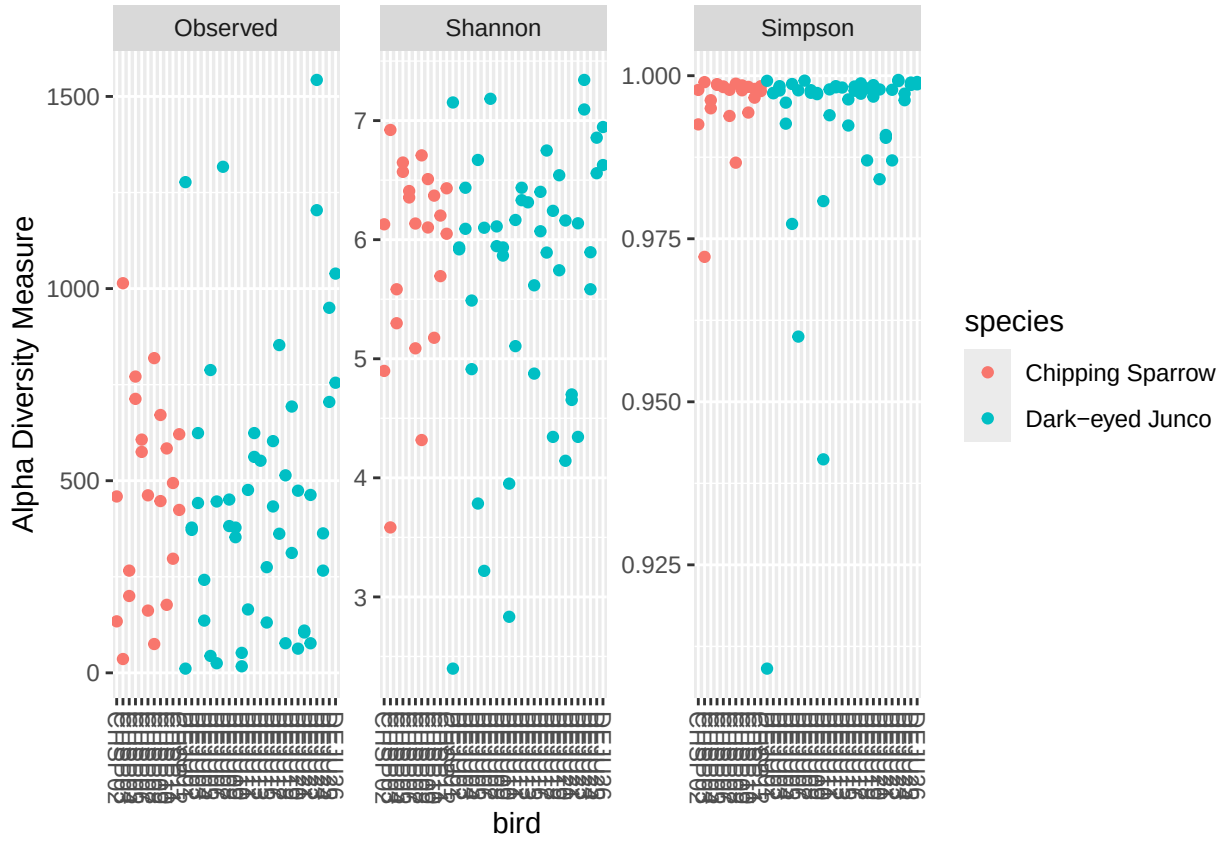
```
plot_richness(physeq, x="species", color= "site", measures=c("Observed", "Simpson", "Shannon"))
```




```
##Simpson(less sensitive, will be more custered together) and Shannon(more sensitive to rare taxa) take
```

#plot alpha diversity based on sex

```
plot_richness(physeq, x="bird", color= "species", measures=c("Observed", "Simpson", "Shannon"))
```




```
##Simpson(less sensitive, will be more custered together) and Shannon(more sensitive to rare taxa) take
```

#creating file with statistics of alpha diversity

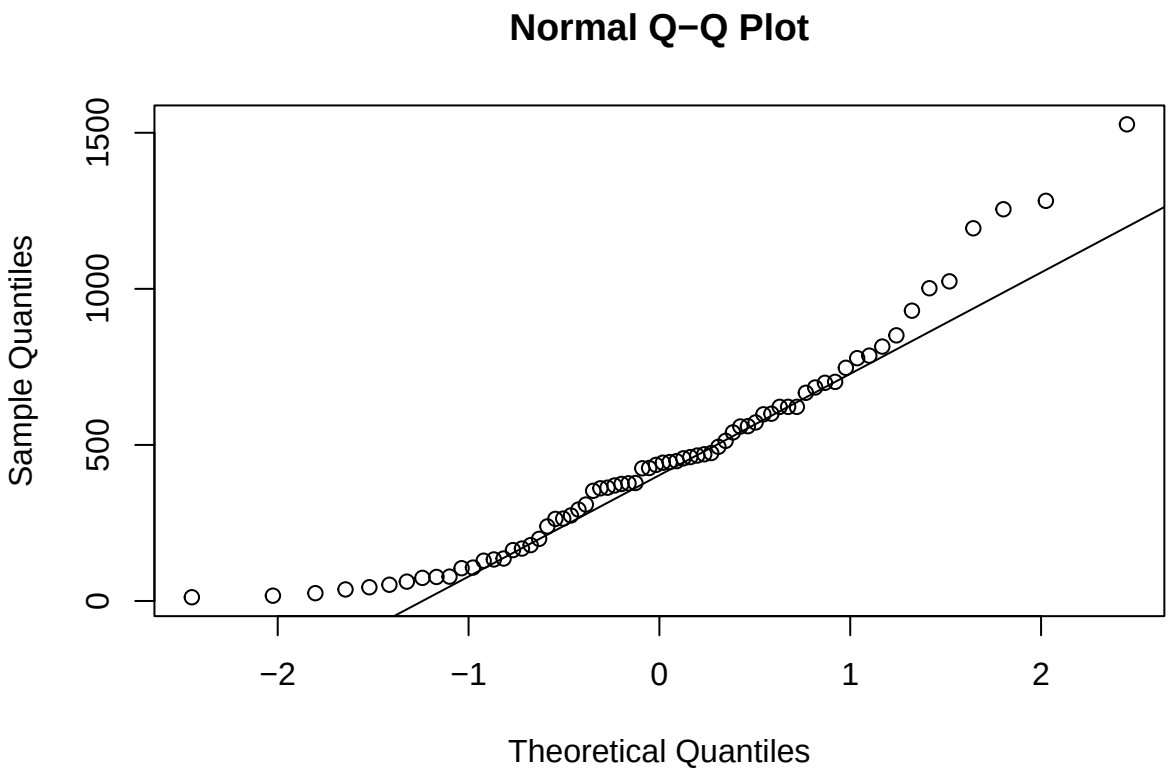
```
alphadiv <- estimate_richness(physeq, measures=c("Observed", "Simpson", "Shannon"))
write.csv(alphadiv, "alpha.csv")
```

#load file that was manually made that has metadata and alpha diversity statistics

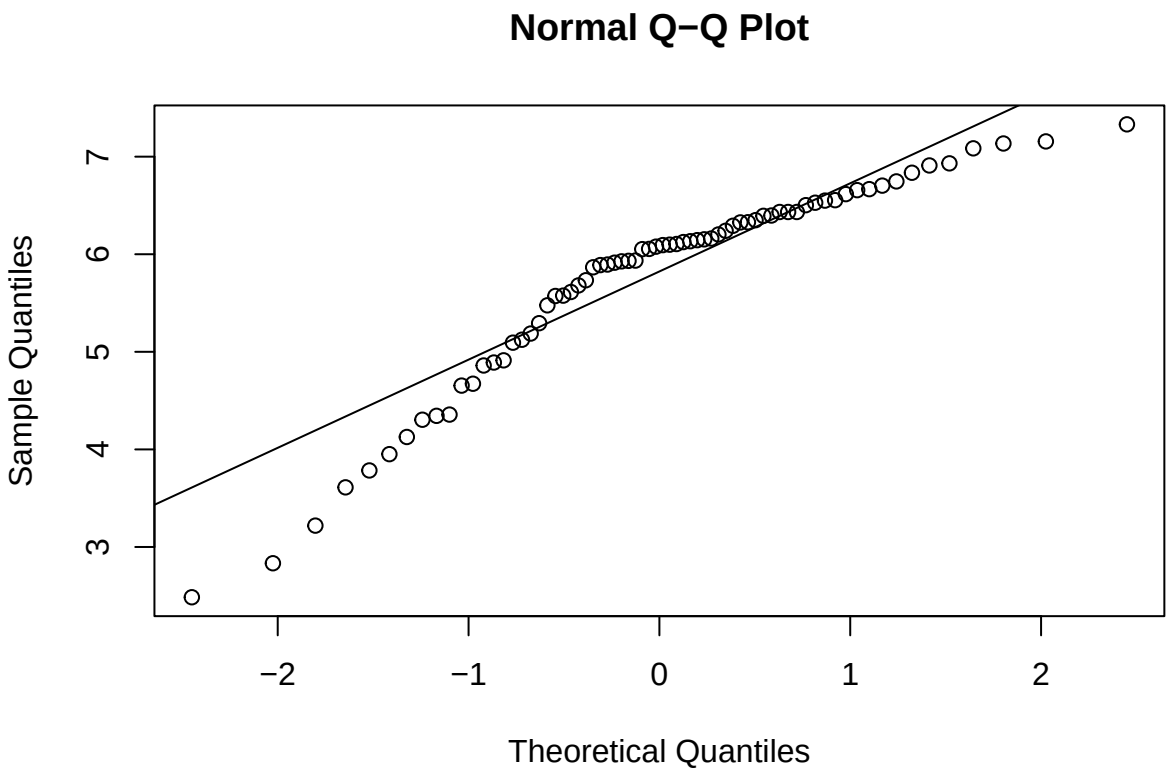
```
alphameta <- read.csv("alphameta.csv")
```

#perform normaility test on diversity index using qq plot and shapiro

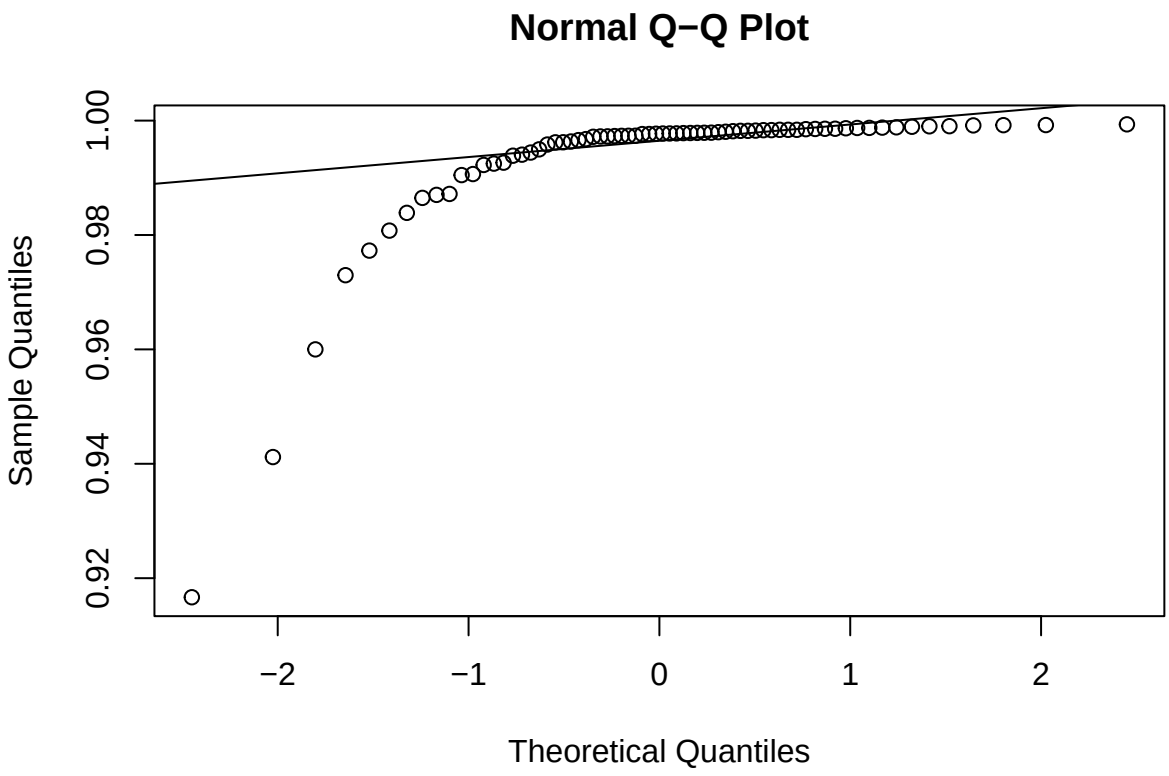
```
#observed stats
qqnorm(alphameta$Observed)
qqline(alphameta$Observed)
```




```
#shannon index stats
qqnorm(alphameta$Shannon)
qqline(alphameta$Shannon)
```




```
#simpson stats
qqnorm(alphameta$Simpson)
qqline(alphameta$Simpson)
```




```
#shaprio test
observed <- shapiro.test(alphameta$Observed)
shannon <- shapiro.test(alphameta$Shannon)
print(observed)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alphameta$Observed
## W = 0.93244, p-value = 0.0009703
```

```
print(shannon)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  alphameta$Shannon
## W = 0.90007, p-value = 3.9e-05
```

#performe statics on site

```
#perform wilcox tests for each biodiversity index
test_observed<- wilcox.test(alphameta$Observed ~ alphameta$site)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
test_shannon <- wilcox.test(alphameta$Shannon ~ alphameta$site)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
test_simpson <- wilcox.test(alphameta$Simpson ~ alphameta$site)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
print(test_observed)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  alphameta$Observed by alphameta$site
## W = 814, p-value = 0.01822
## alternative hypothesis: true location shift is not equal to 0
```

```
print(test_shannon)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  alphameta$Shannon by alphameta$site
## W = 814, p-value = 0.01822
## alternative hypothesis: true location shift is not equal to 0
```

```
print(test_simpson)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  alphameta$Simpson by alphameta$site
## W = 814, p-value = 0.01822
## alternative hypothesis: true location shift is not equal to 0
```

#performe statics on species

```
#perform wilcox tests for each biodiversity index
test_observed<- wilcox.test(alphameta$Observed ~ alphameta$species)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
test_shannon <- wilcox.test(alphameta$Shannon ~ alphameta$species)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
print(test_observed)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  alphameta$Observed by alphameta$species
## W = 566, p-value = 0.6352
## alternative hypothesis: true location shift is not equal to 0
```

```
print(test_shannon)
```

```
##
```

```
##  Wilcoxon rank sum test with continuity correction
##
## data:  alphameta$Shannon by alphameta$species
## W = 566, p-value = 0.6352
## alternative hypothesis: true location shift is not equal to 0
```

#performe statics on speciessite

```
#perform kruskal tests for each biodiversity index
test_observed<- kruskal.test(alphameta$Observed ~ alphameta$speciessite)
test_shannon <- kruskal.test(alphameta$Shannon ~ alphameta$speciessite)
print(test_observed)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  alphameta$Observed by alphameta$speciessite
## Kruskal-Wallis chi-squared = 5.923, df = 3, p-value = 0.1154
```

```
print(test_shannon)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  alphameta$Shannon by alphameta$speciessite
## Kruskal-Wallis chi-squared = 5.923, df = 3, p-value = 0.1154
```

#doing a kruskal-wallis on birds and diversity indices

```
test_observed <- kruskal.test(alphameta$Observed ~ alphameta$bird)
test_shannon <- kruskal.test(alphameta$Shannon ~ alphameta$bird)
print(test_observed)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  alphameta$Observed by alphameta$bird
## Kruskal-Wallis chi-squared = 37.525, df = 35, p-value = 0.3541
```

```
print(test_shannon)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  alphameta$Shannon by alphameta$bird
## Kruskal-Wallis chi-squared = 37.525, df = 35, p-value = 0.3541
```

#remove taxa with relative abundance <0.05%

```
minTotRelAbun = .00005
x = taxa_sums(physeq)
keepTaxa = (x / sum(x)) > minTotRelAbun
physeqprune = prune_taxa(keepTaxa, physeq)
physeqprune
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 4503 taxa and 70 samples ]
## sample_data() Sample Data:       [ 70 samples by 5 sample variables ]
## tax_table()   Taxonomy Table:    [ 4503 taxa by 7 taxonomic ranks ]
```

```
## refseq()      DNAStringSet:      [ 4503 reference sequences ]
```

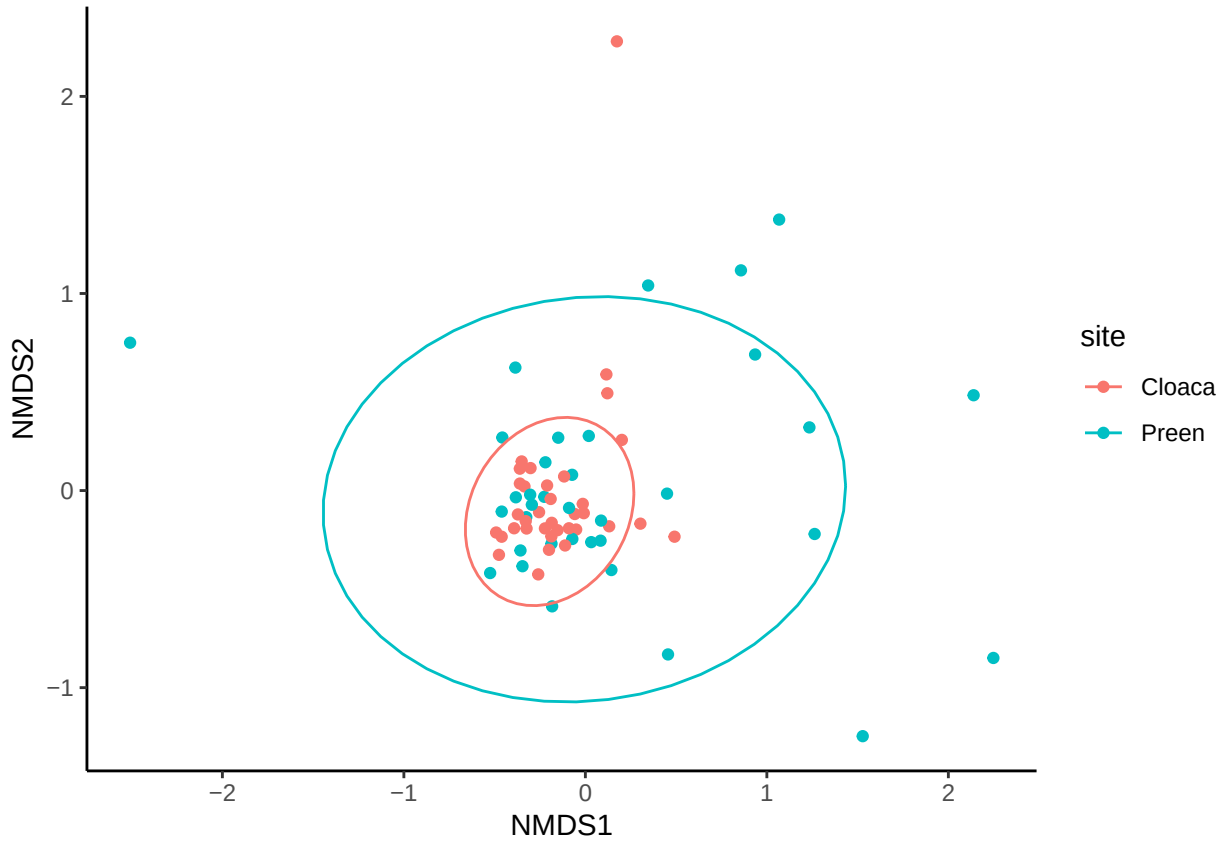
#bray curtis caculation, 0; exactly the same, 1; very diverse

```
set.seed(666)
dist = phyloseq::distance(physeqprune, method="bray", weighted=TRUE)
ordination = ordinate(physeqprune, method="NMDS", distance=dist)
```

```
## Run 0 stress 0.1733585
## Run 1 stress 0.172396
## ... New best solution
## ... Procrustes: rmse 0.09856416  max resid 0.4428449
## Run 2 stress 0.1790348
## Run 3 stress 0.1764735
## Run 4 stress 0.1707124
## ... New best solution
## ... Procrustes: rmse 0.0904237  max resid 0.391689
## Run 5 stress 0.1765716
## Run 6 stress 0.1714972
## Run 7 stress 0.1813422
## Run 8 stress 0.178451
## Run 9 stress 0.1672896
## ... New best solution
## ... Procrustes: rmse 0.04928088  max resid 0.3130497
## Run 10 stress 0.1752438
## Run 11 stress 0.1705463
## Run 12 stress 0.1724554
## Run 13 stress 0.1810925
## Run 14 stress 0.1821154
## Run 15 stress 0.183209
## Run 16 stress 0.1667074
## ... New best solution
## ... Procrustes: rmse 0.09073579  max resid 0.3991868
## Run 17 stress 0.1834809
## Run 18 stress 0.1684114
## Run 19 stress 0.1660158
## ... New best solution
## ... Procrustes: rmse 0.06977916  max resid 0.3425165
## Run 20 stress 0.1773843
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      8: no. of iterations >= maxit
##     12: stress ratio > sratmax
```

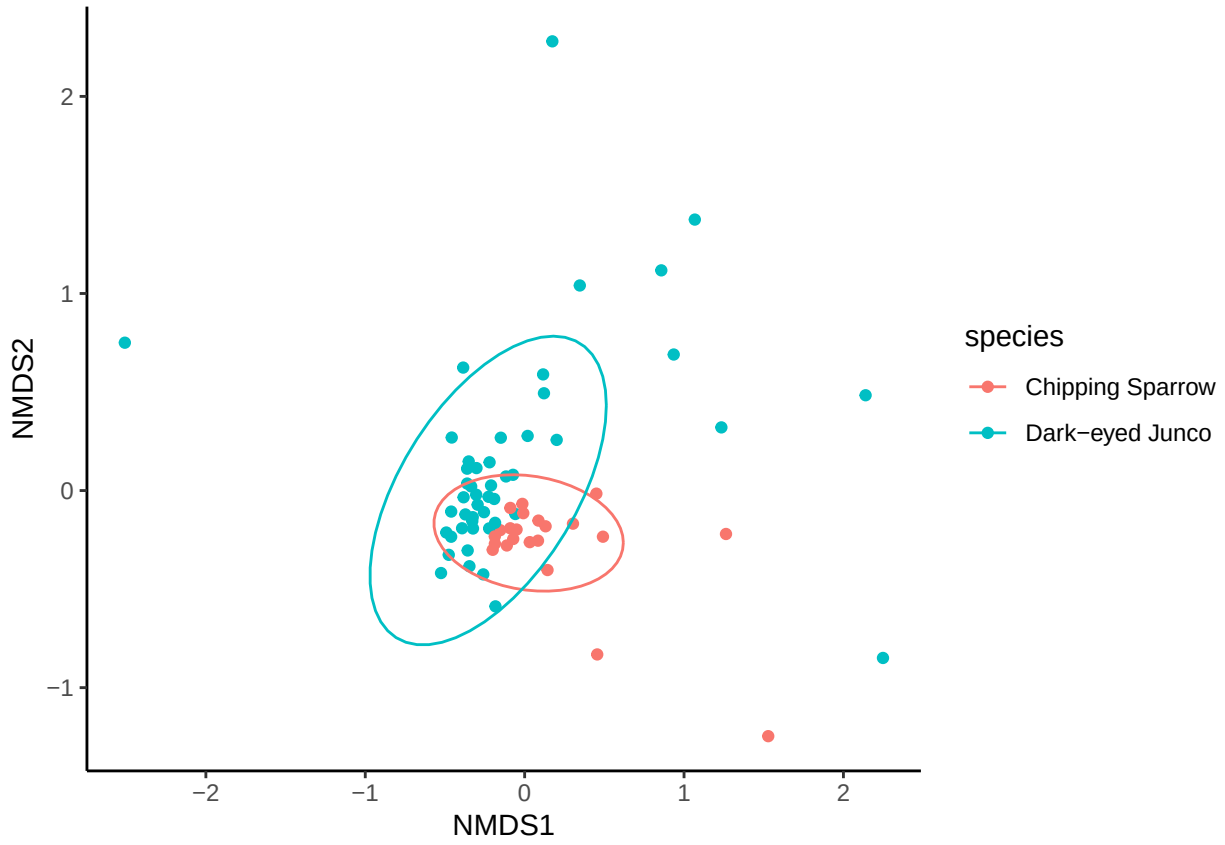
#bray curtis site plot

```
braysite=plot_ordination(physeqprune, ordination, color="site") +
  theme_classic() +
  theme(strip.background = element_blank()) + stat_ellipse(aes(group=site))
braysite
```



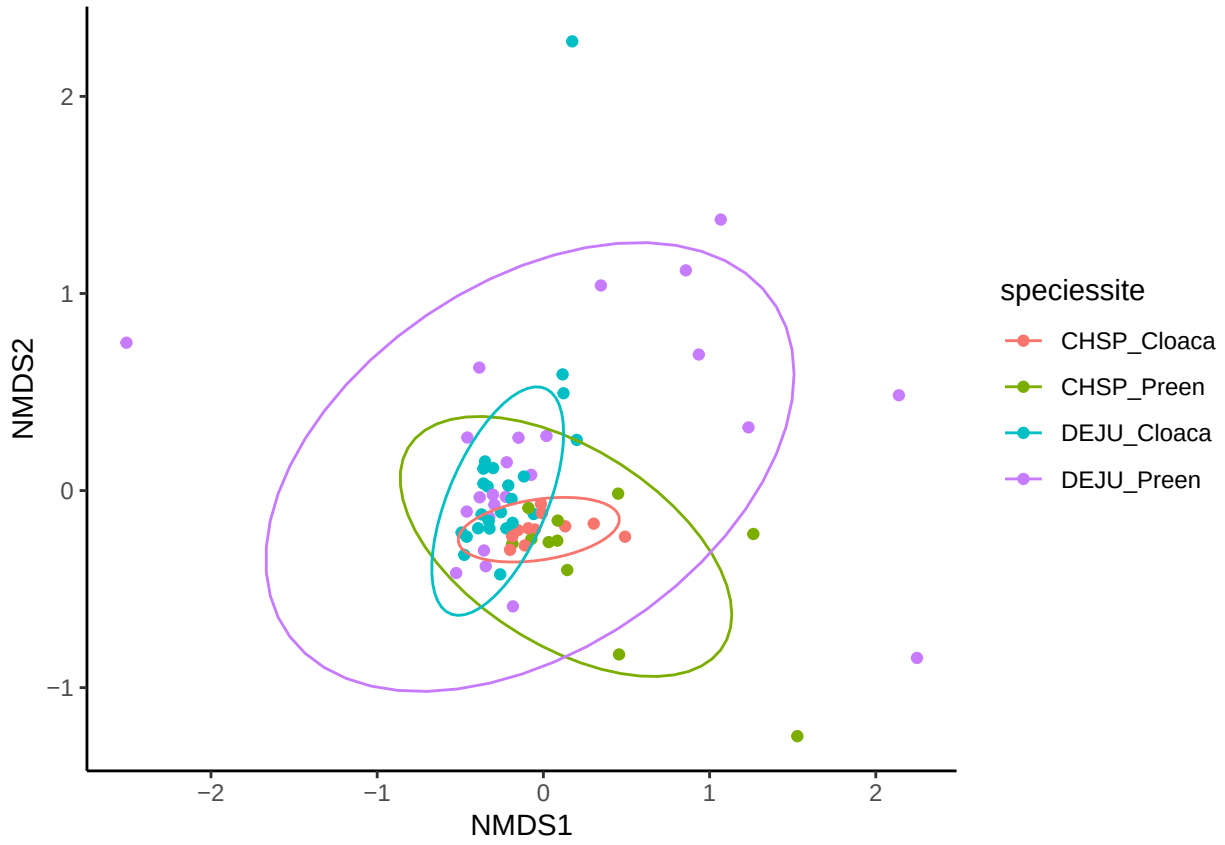
#bray curtis species plot

```
brayspecies=plot_ordination(physeqprune, ordination, color="species") +
  theme_classic() +
  theme(strip.background = element_blank()) + stat_ellipse(aes(group=species))
brayspecies
```



#bray curtis speicessite plot

```
brayspeciessite=plot_ordination(physeqprune, ordination, color="speciessite") +
  theme_classic() +
  theme(strip.background = element_blank()) + stat_ellipse(aes(group=speciessite))
brayspeciessite
```



#bray curits statistics

```
#on species
adonis2(dist ~ sample_data(physeqprune)$species)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = dist ~ sample_data(physeqprune)$species)
##                                 Df SumOfSqs      R2      F Pr(>F)
## sample_data(physeqprune)$species  1   0.9446 0.03576 2.5218  0.001 ***
## Residual                         68  25.4718 0.96424
## Total                            69  26.4165 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#on site
adonis2(dist ~ sample_data(physeqprune)$site)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = dist ~ sample_data(physeqprune)$site)
##                              Df SumOfSqs      R2      F Pr(>F)
```

```
## sample_data(physeqprune)$site  1   0.5755 0.02179 1.5144  0.002 **
## Residual                      68  25.8410 0.97821
## Total                         69  26.4165 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#bray curtis species and site stats

```
adonis2(dist ~ sample_data(physeqprune)$speciessite)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = dist ~ sample_data(physeqprune)$speciessite)
##                                    Df SumOfSqs      R2      F Pr(>F)
## sample_data(physeqprune)$speciessite  3   1.8442 0.06981 1.6512  0.001 ***
## Residual                             66  24.5722 0.93019
## Total                                69  26.4165 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ps.disper<-betadisper(dist, sample_data(physeqprune)$speciessite)
permutest(ps.disper, pair=TRUE)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##           Df   Sum Sq  Mean Sq      F N.Perm Pr(>F)
## Groups     3 0.071715 0.023905 7.6741    999  0.002 **
## Residuals 66 0.205593 0.003115
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##             CHSP_Cloaca CHSP_Preen DEJU_Cloaca DEJU_Preen
## CHSP_Cloaca             5.0000e-02  6.8000e-02      0.001
## CHSP_Preen   3.9331e-02             6.1700e-01      0.052
## DEJU_Cloaca  4.8232e-02 6.2133e-01                 0.005
## DEJU_Preen   5.8381e-05 5.6283e-02  2.6422e-03
```