MicrobiomeSequence

Alejandra

2024-04-21

#load required packages

```
library(dada2)
```

```
## Loading required package: Rcpp
```

```
library(Biostrings)
```

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

## 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, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min

## 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

## Loading required package: XVector

## Loading required package: GenomeInfoDb

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

```
##
## Attaching package: 'Biostrings'

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

```
library(ShortRead)
```

```
## Loading required package: BiocParallel

## 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
```

```r
library(phyloseq)
```

```
##
## Attaching package: 'phyloseq'

## The following object is masked from 'package:SummarizedExperiment':
##
##     distance

## The following object is masked from 'package:Biobase':
##
##     sampleNames

## The following object is masked from 'package:GenomicRanges':
##
##     distance

## The following object is masked from 'package:IRanges':
##
##     distance
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:ShortRead':
##
##     id

## The following objects are masked from 'package:GenomicAlignments':
##
##     first, last

## The following object is masked from 'package:Biobase':
##
##     combine

## The following object is masked from 'package:matrixStats':
##
##     count

## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union

## The following objects are masked from 'package:Biostrings':
##
##     collapse, intersect, setdiff, setequal, union

## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect

## The following object is masked from 'package:XVector':
##
##     slice

## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
## The following objects are masked from ’package:S4Vectors’:
##
##     first, intersect, rename, setdiff, setequal, union
```

```
## The following objects are masked from ’package:BiocGenerics’:
##
##     combine, intersect, setdiff, union
```

```
## The following objects are masked from ’package:stats’:
##
##     filter, lag
```

```
## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

```
library(BiMiCo)
library(ggplot2)
library(devtools)
```

```
## Loading required package: usethis
```

```
library(MicEco)
library(vegan)
```

```
## Loading required package: permute
```

```
##
## Attaching package: ’permute’
```

```
## The following object is masked from ’package:devtools’:
##
##     check
```

```
## Loading required package: lattice
```

```
## This is vegan 2.6-4
```

#load sequences

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

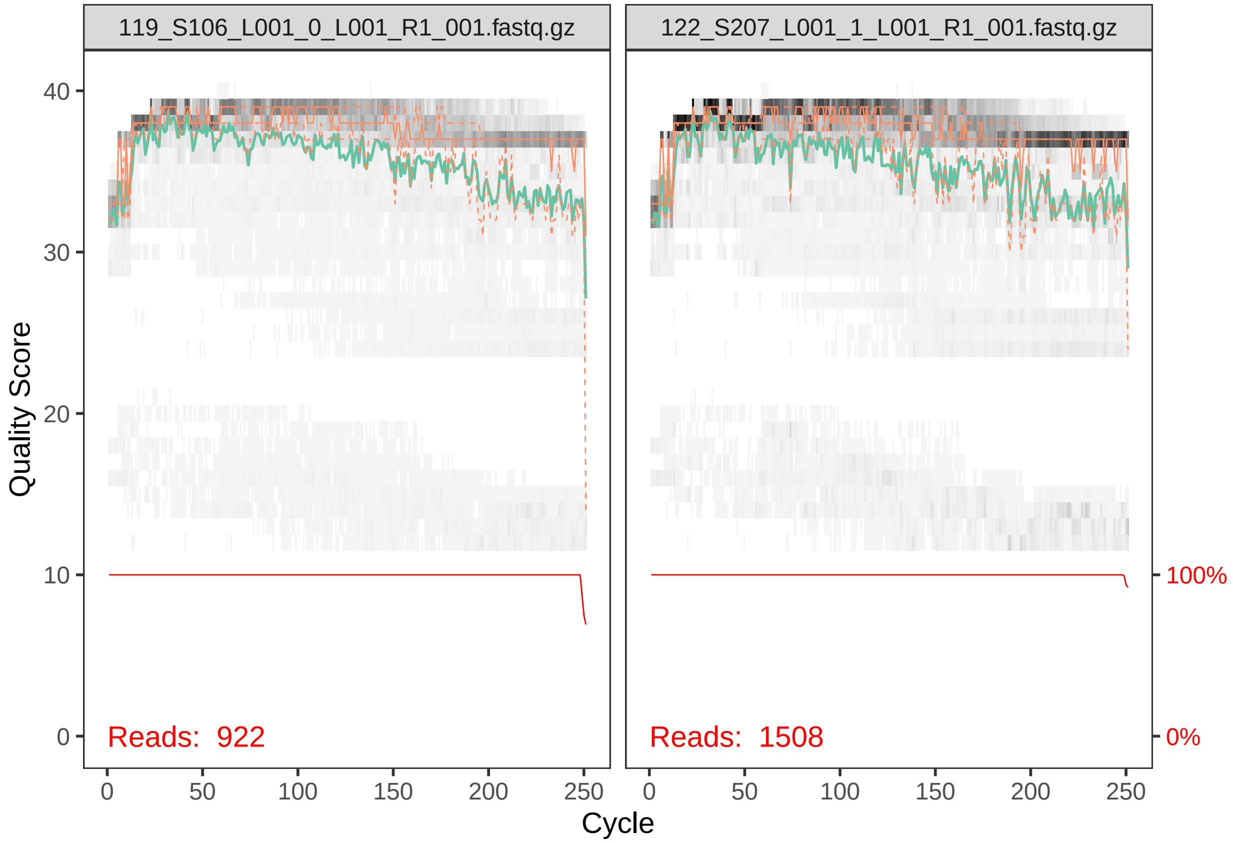
```
##  [1] "119_S106_L001_0_L001_R1_001.fastq.gz"
##  [2] "119_S106_L001_24_L001_R2_001.fastq.gz"
##  [3] "122_S207_L001_1_L001_R1_001.fastq.gz"
##  [4] "122_S207_L001_25_L001_R2_001.fastq.gz"
##  [5] "133_S265_L001_2_L001_R1_001.fastq.gz"
##  [6] "133_S265_L001_26_L001_R2_001.fastq.gz"
##  [7] "165_S230_L001_27_L001_R2_001.fastq.gz"
##  [8] "165_S230_L001_3_L001_R1_001.fastq.gz"
##  [9] "176_S154_L001_28_L001_R2_001.fastq.gz"
## [10] "176_S154_L001_4_L001_R1_001.fastq.gz"
## [11] "208_S177_L001_29_L001_R2_001.fastq.gz"
## [12] "208_S177_L001_5_L001_R1_001.fastq.gz"
## [13] "210_S336_L001_30_L001_R2_001.fastq.gz"
## [14] "210_S336_L001_6_L001_R1_001.fastq.gz"
## [15] "220_S155_L001_31_L001_R2_001.fastq.gz"
## [16] "220_S155_L001_7_L001_R1_001.fastq.gz"
## [17] "236_S241_L001_32_L001_R2_001.fastq.gz"
## [18] "236_S241_L001_8_L001_R1_001.fastq.gz"
## [19] "252_S179_L001_33_L001_R2_001.fastq.gz"
## [20] "252_S179_L001_9_L001_R1_001.fastq.gz"
## [21] "260_S178_L001_10_L001_R1_001.fastq.gz"
## [22] "260_S178_L001_34_L001_R2_001.fastq.gz"
## [23] "281_S130_L001_11_L001_R1_001.fastq.gz"
## [24] "281_S130_L001_35_L001_R2_001.fastq.gz"
## [25] "282_S217_L001_12_L001_R1_001.fastq.gz"
## [26] "282_S217_L001_36_L001_R2_001.fastq.gz"
## [27] "306_S120_L001_13_L001_R1_001.fastq.gz"
## [28] "306_S120_L001_37_L001_R2_001.fastq.gz"
## [29] "331_S131_L001_14_L001_R1_001.fastq.gz"
## [30] "331_S131_L001_38_L001_R2_001.fastq.gz"
## [31] "332_S105_L001_15_L001_R1_001.fastq.gz"
## [32] "332_S105_L001_39_L001_R2_001.fastq.gz"
## [33] "361_S168_L001_16_L001_R1_001.fastq.gz"
## [34] "361_S168_L001_40_L001_R2_001.fastq.gz"
## [35] "368_S129_L001_17_L001_R1_001.fastq.gz"
## [36] "368_S129_L001_41_L001_R2_001.fastq.gz"
## [37] "41_S254_L001_18_L001_R1_001.fastq.gz"
## [38] "41_S254_L001_42_L001_R2_001.fastq.gz"
## [39] "50_S144_L001_19_L001_R1_001.fastq.gz"
## [40] "50_S144_L001_43_L001_R2_001.fastq.gz"
## [41] "57_S153_L001_20_L001_R1_001.fastq.gz"
## [42] "57_S153_L001_44_L001_R2_001.fastq.gz"
## [43] "72_S206_L001_21_L001_R1_001.fastq.gz"
## [44] "72_S206_L001_45_L001_R2_001.fastq.gz"
## [45] "90_S107_L001_22_L001_R1_001.fastq.gz"
## [46] "90_S107_L001_46_L001_R2_001.fastq.gz"
## [47] "94_S278_L001_23_L001_R1_001.fastq.gz"
## [48] "94_S278_L001_47_L001_R2_001.fastq.gz"
## [49] "filtered"
## [50] "MANIFEST"
## [51] "metadata.yml"
## [52] "RData"
```

#read file names

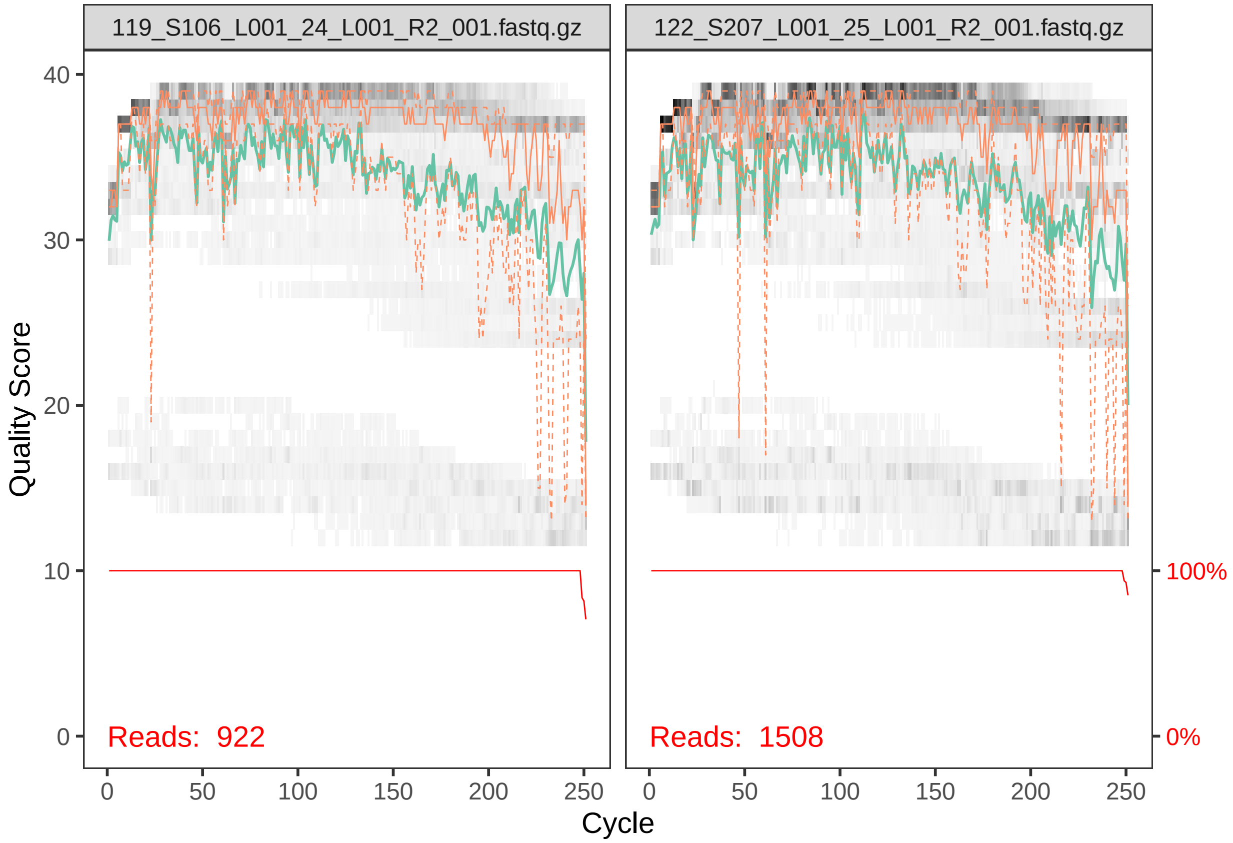
```r
fnFs <- sort(list.files(path, pattern="_R1_001.fastq", full.names = TRUE))
fnRs <- sort(list.files(path, pattern="_R2_001.fastq", full.names = TRUE))
#extract file names
sample.names <- sapply(strsplit(basename(fnFs), "_"), `[`, 1)
```

#inspect file quality of forward and reverse reads

```r
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(200,200),
              maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,
              compress=TRUE, multithread=TRUE)
head(out)
```

```
##                                         reads.in reads.out
## 119_S106_L001_0_L001_R1_001.fastq.gz       922       837
## 122_S207_L001_1_L001_R1_001.fastq.gz      1508      1338
## 133_S265_L001_2_L001_R1_001.fastq.gz      2072      1809
## 165_S230_L001_3_L001_R1_001.fastq.gz     34066     31511
## 176_S154_L001_4_L001_R1_001.fastq.gz     32573     29451
## 208_S177_L001_5_L001_R1_001.fastq.gz      8877      8054
```

#learn error rates of reads

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

```
## 65004600 total bases in 325023 reads from 24 samples will be used for learning the error rates.
```

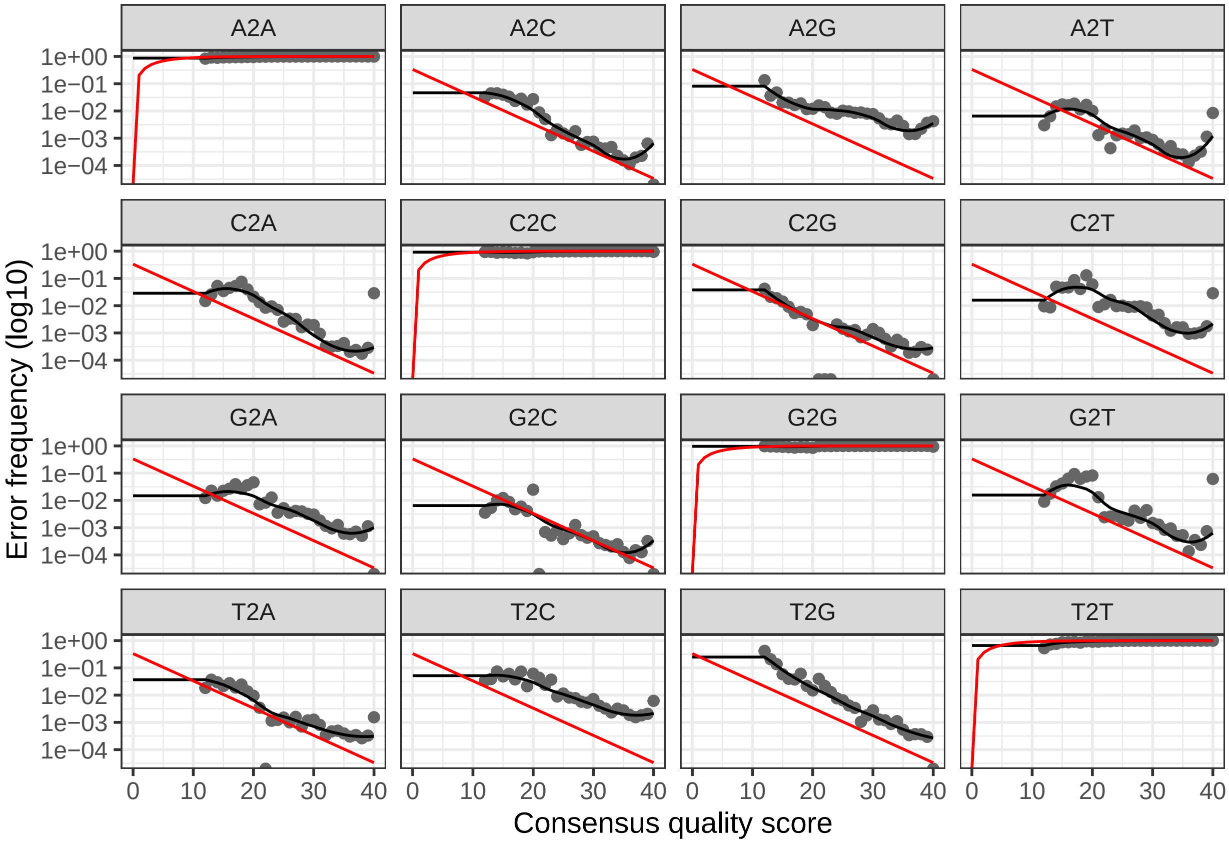
```
errR <- learnErrors(filtRs, multithread=TRUE)
```

```
## 65004600 total bases in 325023 reads from 24 samples will be used for learning the error rates.
```

#visulaize error rate

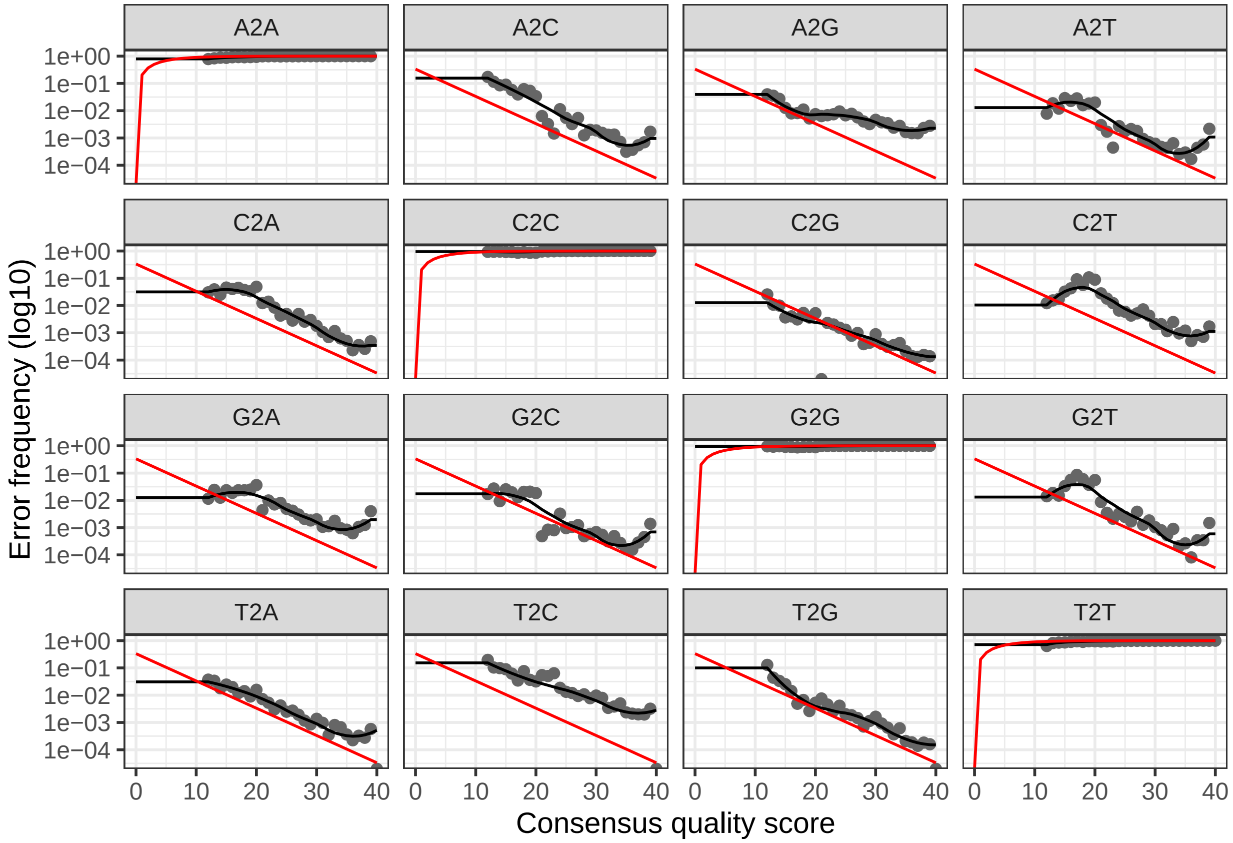
```
plotErrors(errF, nominalQ=TRUE)
```

```
## 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 - 837 reads in 288 unique sequences.
## Sample 2 - 1338 reads in 521 unique sequences.
## Sample 3 - 1809 reads in 613 unique sequences.
## Sample 4 - 31511 reads in 7794 unique sequences.
## Sample 5 - 29451 reads in 7552 unique sequences.
## Sample 6 - 8054 reads in 2135 unique sequences.
## Sample 7 - 5098 reads in 1619 unique sequences.
## Sample 8 - 41388 reads in 9435 unique sequences.
## Sample 9 - 28495 reads in 6303 unique sequences.
## Sample 10 - 681 reads in 218 unique sequences.
## Sample 11 - 2786 reads in 793 unique sequences.
## Sample 12 - 21127 reads in 5613 unique sequences.
## Sample 13 - 12933 reads in 3517 unique sequences.
## Sample 14 - 2674 reads in 853 unique sequences.
## Sample 15 - 7939 reads in 2302 unique sequences.
## Sample 16 - 968 reads in 342 unique sequences.
## Sample 17 - 64147 reads in 12869 unique sequences.
## Sample 18 - 6961 reads in 2094 unique sequences.
## Sample 19 - 567 reads in 216 unique sequences.
## Sample 20 - 13772 reads in 3683 unique sequences.
## Sample 21 - 9094 reads in 2719 unique sequences.
## Sample 22 - 20612 reads in 4716 unique sequences.
## Sample 23 - 6037 reads in 1628 unique sequences.
## Sample 24 - 6744 reads in 1906 unique sequences.
```

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

```
## Sample 1 - 837 reads in 354 unique sequences.
## Sample 2 - 1338 reads in 619 unique sequences.
## Sample 3 - 1809 reads in 913 unique sequences.
## Sample 4 - 31511 reads in 10707 unique sequences.
## Sample 5 - 29451 reads in 11673 unique sequences.
## Sample 6 - 8054 reads in 3086 unique sequences.
## Sample 7 - 5098 reads in 2279 unique sequences.
## Sample 8 - 41388 reads in 15824 unique sequences.
## Sample 9 - 28495 reads in 9626 unique sequences.
## Sample 10 - 681 reads in 372 unique sequences.
## Sample 11 - 2786 reads in 1168 unique sequences.
## Sample 12 - 21127 reads in 7792 unique sequences.
## Sample 13 - 12933 reads in 5314 unique sequences.
## Sample 14 - 2674 reads in 1200 unique sequences.
## Sample 15 - 7939 reads in 3304 unique sequences.
## Sample 16 - 968 reads in 439 unique sequences.
## Sample 17 - 64147 reads in 19827 unique sequences.
## Sample 18 - 6961 reads in 3020 unique sequences.
## Sample 19 - 567 reads in 287 unique sequences.
## Sample 20 - 13772 reads in 5336 unique sequences.
## Sample 21 - 9094 reads in 4062 unique sequences.
## Sample 22 - 20612 reads in 6350 unique sequences.
## Sample 23 - 6037 reads in 2495 unique sequences.
## Sample 24 - 6744 reads in 2580 unique sequences.
```

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

```
## dada-class: object describing DADA2 denoising results
## 47 sequence variants were inferred from 288 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)
```

```
## 762 paired-reads (in 43 unique pairings) successfully merged out of 799 (in 49 pairings) input.

## 1172 paired-reads (in 55 unique pairings) successfully merged out of 1252 (in 72 pairings) input.

## 1629 paired-reads (in 75 unique pairings) successfully merged out of 1717 (in 102 pairings) input.

## 29841 paired-reads (in 403 unique pairings) successfully merged out of 30767 (in 622 pairings) input

## 28295 paired-reads (in 309 unique pairings) successfully merged out of 29025 (in 440 pairings) input

## 7706 paired-reads (in 121 unique pairings) successfully merged out of 7883 (in 171 pairings) input.

## 4858 paired-reads (in 90 unique pairings) successfully merged out of 4992 (in 121 pairings) input.

## 40328 paired-reads (in 291 unique pairings) successfully merged out of 41118 (in 445 pairings) input

## 27535 paired-reads (in 214 unique pairings) successfully merged out of 28033 (in 340 pairings) input

## 611 paired-reads (in 28 unique pairings) successfully merged out of 633 (in 36 pairings) input.

## 2702 paired-reads (in 64 unique pairings) successfully merged out of 2720 (in 73 pairings) input.

## 20425 paired-reads (in 267 unique pairings) successfully merged out of 20789 (in 347 pairings) input
## 12207 paired-reads (in 251 unique pairings) successfully merged out of 12657 (in 322 pairings) input

## 2518 paired-reads (in 87 unique pairings) successfully merged out of 2576 (in 99 pairings) input.

## 7466 paired-reads (in 149 unique pairings) successfully merged out of 7763 (in 198 pairings) input.

## 911 paired-reads (in 47 unique pairings) successfully merged out of 920 (in 50 pairings) input.

## 63058 paired-reads (in 347 unique pairings) successfully merged out of 63747 (in 496 pairings) input

## 6537 paired-reads (in 210 unique pairings) successfully merged out of 6733 (in 243 pairings) input.

## 487 paired-reads (in 14 unique pairings) successfully merged out of 489 (in 15 pairings) input.

## 13315 paired-reads (in 229 unique pairings) successfully merged out of 13574 (in 273 pairings) input

## 8632 paired-reads (in 196 unique pairings) successfully merged out of 8869 (in 263 pairings) input.

## 20061 paired-reads (in 225 unique pairings) successfully merged out of 20358 (in 315 pairings) input

## 5848 paired-reads (in 110 unique pairings) successfully merged out of 5957 (in 129 pairings) input.

## 6397 paired-reads (in 104 unique pairings) successfully merged out of 6603 (in 148 pairings) input.
```

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

```
##
## 1                                TACGTAAAAGACAAGTGTTATTCATCTTTAATAGGTTTAAAGGGTACCTAGACGGTATTATTAGCCCA
## 2                           CACAAGTAAGATTAGTGTTATTCATCTTTATTAGGTTTAAAGGGTACCTAGACGGCAAAAGCAACTTCTAAAA
## 3 TACGAAGGGGGCTAGCGTTGCTCGGAATCACTGGGCGTAAAGGGCGCGTAGGCGGCCGTTTAAGTCGGGGGTGAAAGCCTGTGGCTCAACCACAGAATT
## 4                                TACGTAAAAGACAAGTGTTATTCATCTTTAATAGGTTTAAAGGGTACCTAGACGGTATTATTAGCCCA
## 5 TACGTAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGTGCGCAGGCGGTTTTGTAAGTTTGTCGTGAAATCCCCGGGCTCAACCTGGGAATC
## 6 TACGAAGGGGGCTAGCGTTGCTCGGAATCACTGGGCGTAAAGGGCGCGTAGGCGGCGTTTTAAGTCGGGGGTGAAAGCCTGTGGCTCAACCACAGAATC
##   abundance forward reverse nmatch nmismatch nindel prefer accept
## 1        63       2       3    178         0      0      1   TRUE
## 2        60       1       1    173         0      0      1   TRUE
## 3        55       4       4    147         0      0      1   TRUE
## 4        51       3       2    178         0      0      1   TRUE
## 5        47       9       7    147         0      0      1   TRUE
## 6        43       5      34    147         0      0      1   TRUE
```

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

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

```
## [1]   24 2229
```

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

```
##
##  201  203  204  216  220  221  222  223  224  225  226  227  228  229  231  233
##    1    2    1    1   19   30   17   33   16    7    9   35    5    8    1    1
##  235  236  237  238  239  240  244  247  249  250  251  252  253  254  255  256
##    1    1    3    1    1    2   14    1    2    1    4   60 1811  109    6    4
##  257  260  265  266  274  275  293  304  313  325  335  336  359  362  363  365
##    2    2    1    2    1    1    2    1    1    1    3    1    1    1    1    1
```

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

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

```
## Identified 12 bimeras out of 2229 input sequences.
```

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

```
## [1]   24 2217
```

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

```
## [1] 0.9975998
```

#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
## 119   922      837       814       802    762     762
## 122  1508     1338      1279      1278   1172    1172
## 133  2072     1809      1747      1752   1629    1629
## 165 34066    31511     31066     31004  29841   29841
## 176 32573    29451     29246     29149  28295   28295
## 208  8877     8054      7948      7958   7706    7706
```

#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("sample-metadata.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:         [ 2217 taxa and 24 samples ]
## sample_data() Sample Data:       [ 24 samples by 6 sample variables ]
## tax_table()   Taxonomy Table:    [ 2217 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:         [ 2217 taxa and 24 samples ]
## sample_data() Sample Data:       [ 24 samples by 6 sample variables ]
## tax_table()   Taxonomy Table:    [ 2217 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 2217 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:         [ 1929 taxa and 24 samples ]
## sample_data() Sample Data:       [ 24 samples by 6 sample variables ]
## tax_table()   Taxonomy Table:    [ 1929 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 1929 reference sequences ]
```

#remove all non bacterial sequences

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

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 1929 taxa and 24 samples ]
## sample_data() Sample Data:       [ 24 samples by 6 sample variables ]
## tax_table()   Taxonomy Table:    [ 1929 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 1929 reference sequences ]
```

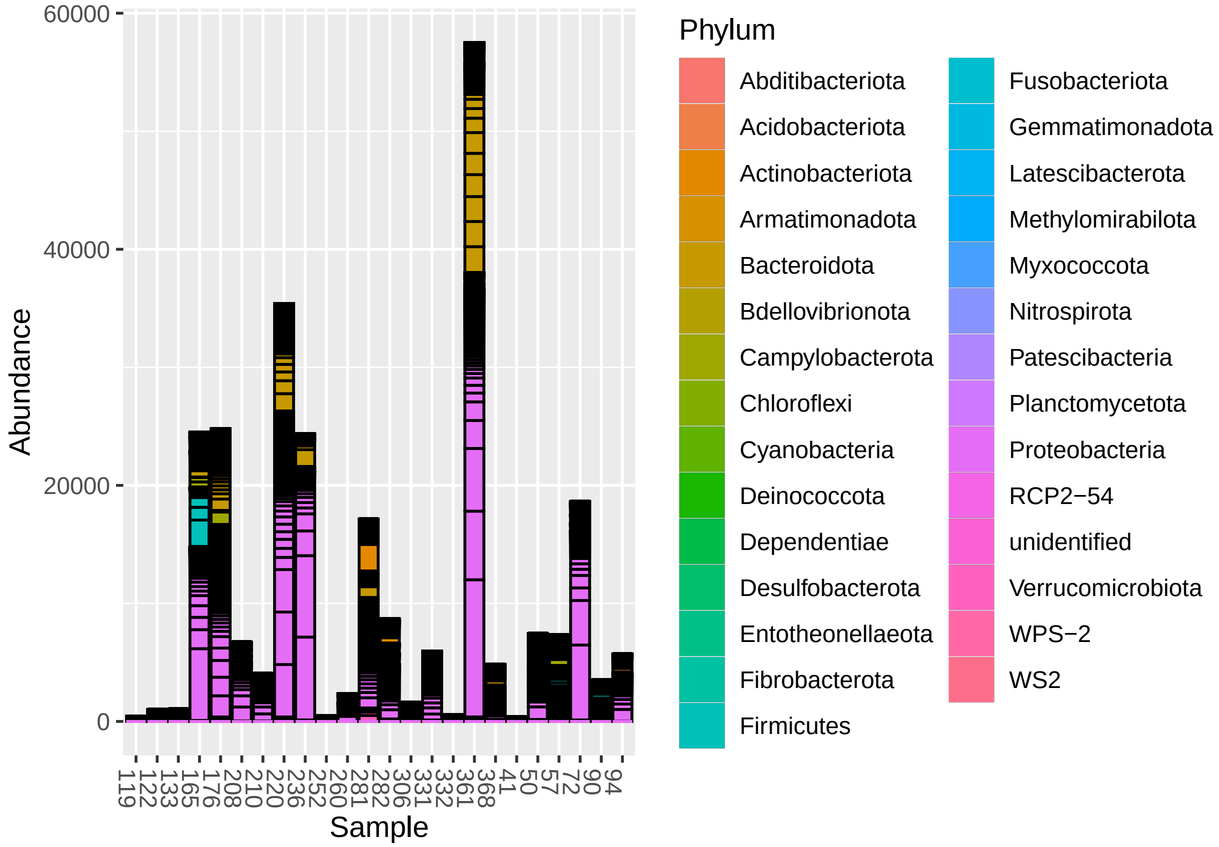
#save physeq objects and load

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

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

#plot bar grpah based on phylum

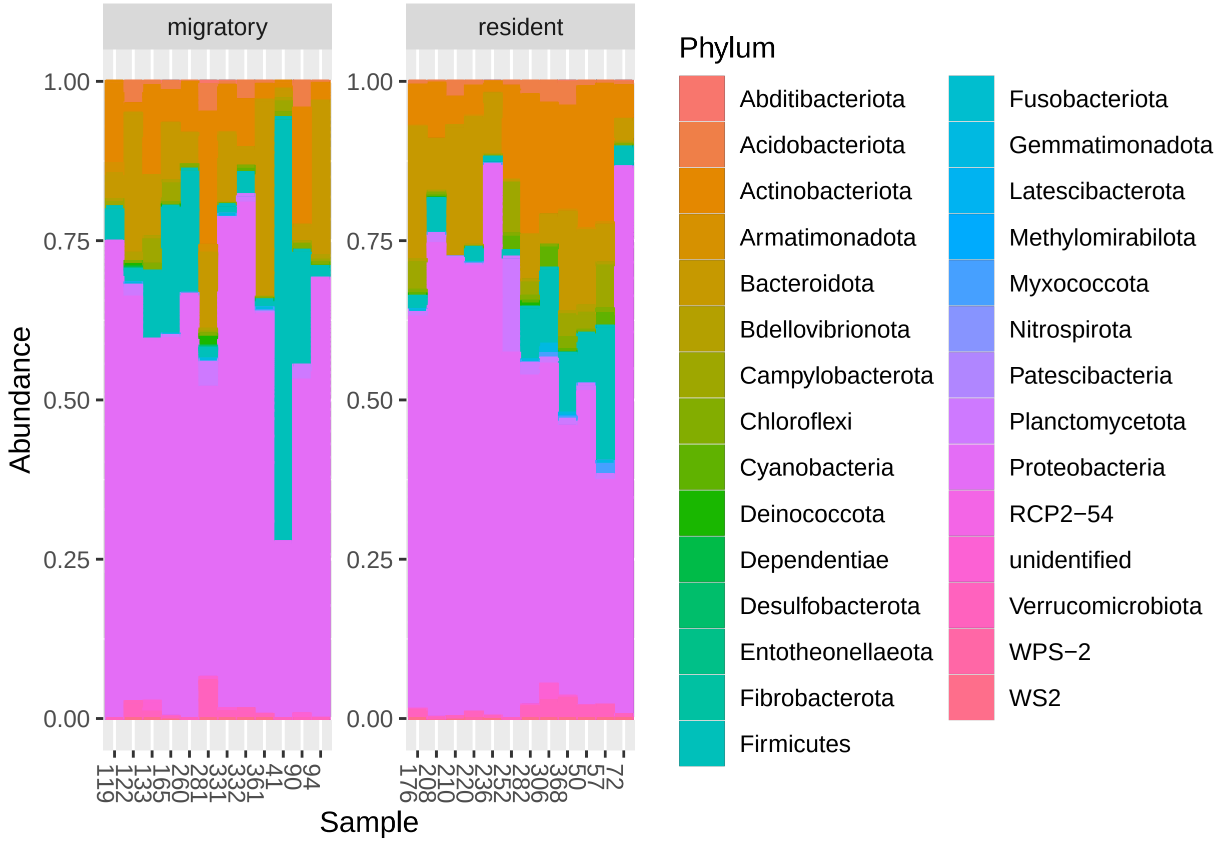
```
plot_bar(physeq, fill = "Phylum") + geom_bar(aes(color=Phylum, fill=Phylum), stat="identity", position=
```



#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",
```

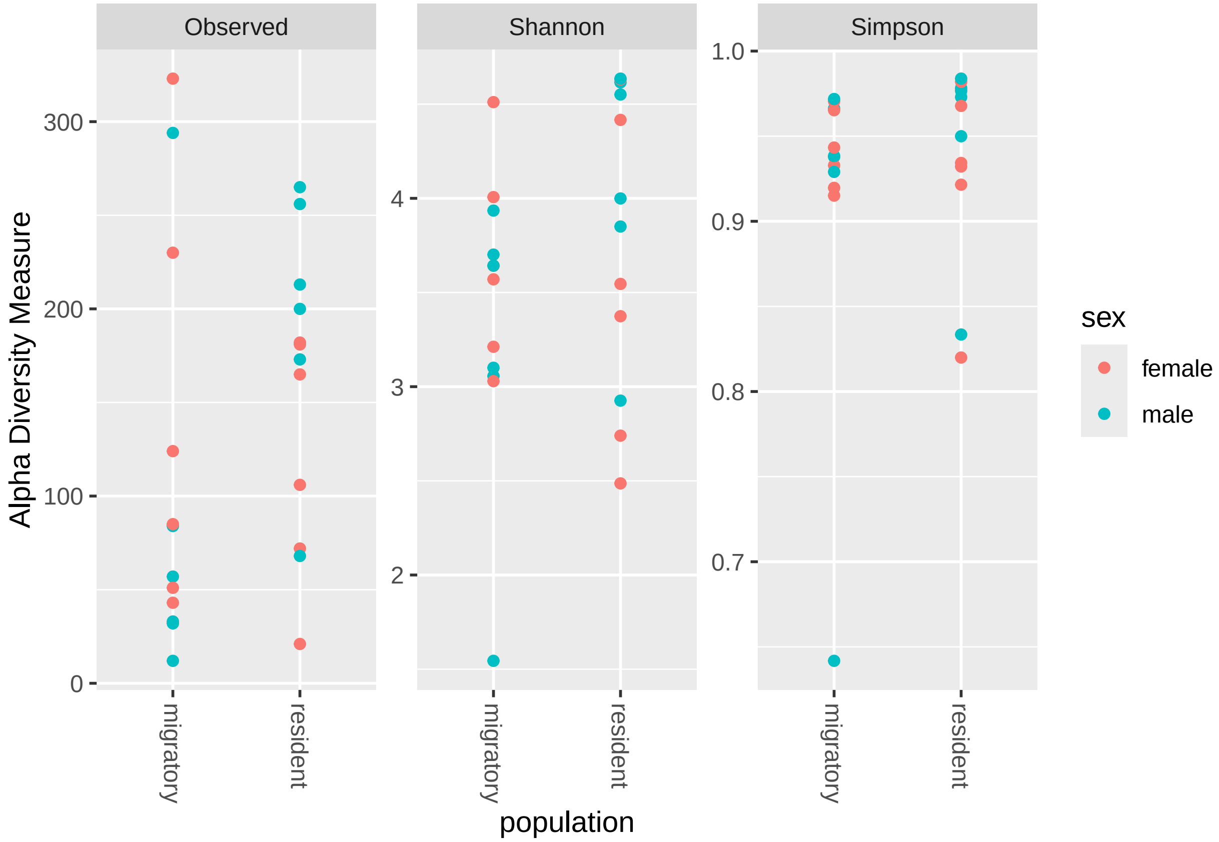


```
##can change based on the column name in metadata in facet_wrap(~columnName)
```

#plot alpha diversity based on bird

```
plot_richness(physeq, x="population", color= "sex", measures=c("Observed", "Simpson", "Shannon"))
```

```
## Warning in estimate_richness(physeq, split = TRUE, measures = measures): The data you have provided
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
```

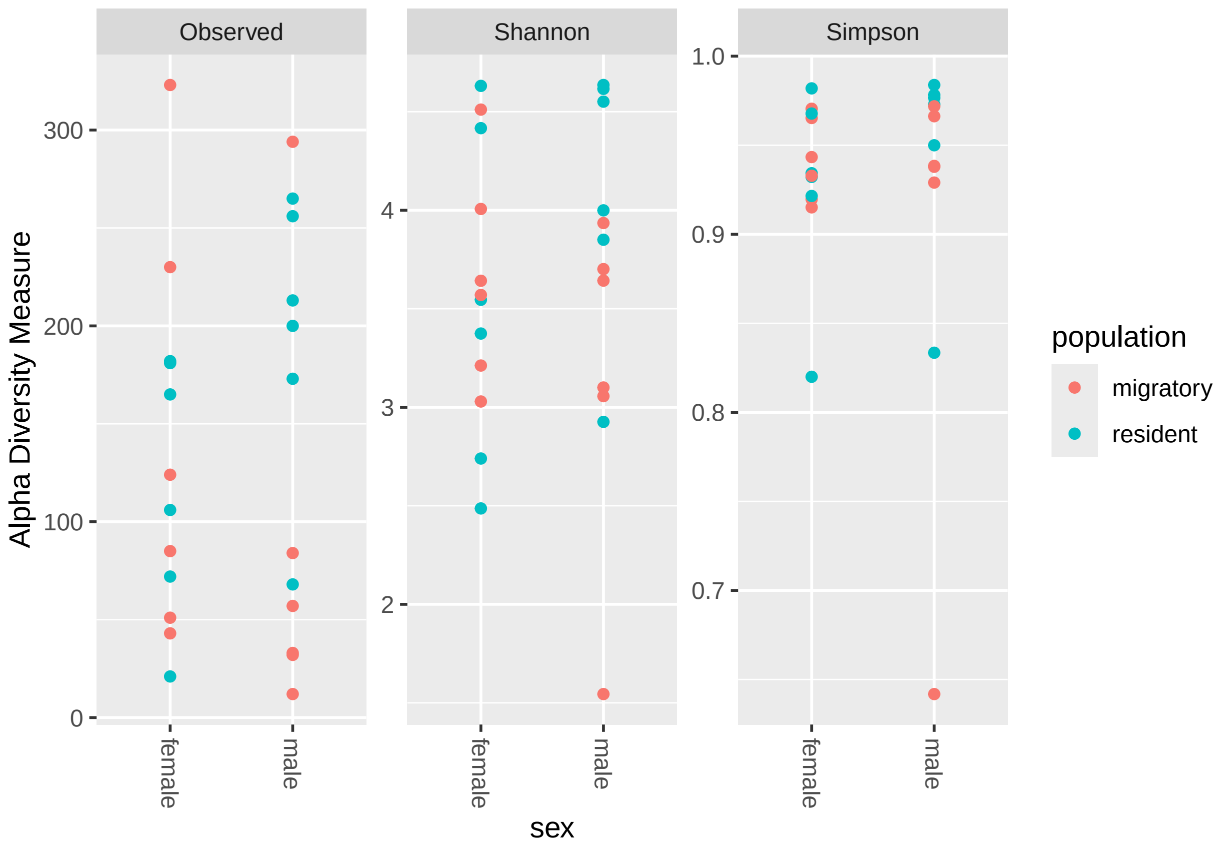


```
##Simpson(less sensitive, will be more custered together) and Shannon(more sensitive to rare taxa) take
```

#plot alpha diversity based on sex

```
plot_richness(physeq, x="sex", color= "population", measures=c("Observed", "Simpson", "Shannon"))
```

```
## Warning in estimate_richness(physeq, split = TRUE, measures = measures): The data you have provided
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
```

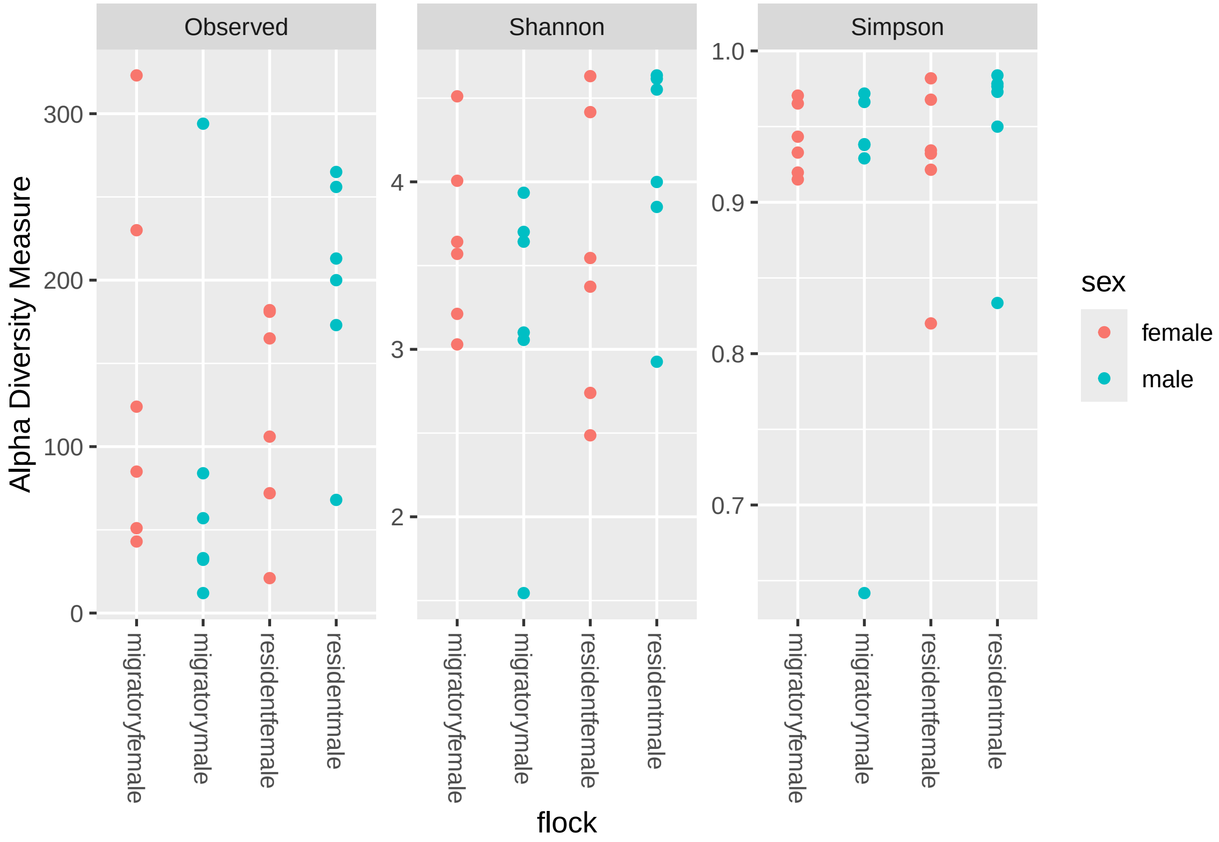


```
##Simpson(less sensitive, will be more custered together) and Shannon(more sensitive to rare taxa) take
```

#plot alpha diversity based on sex

```
plot_richness(physeq, x="flock", color= "sex", measures=c("Observed", "Simpson", "Shannon"))
```

```
## Warning in estimate_richness(physeq, split = TRUE, measures = measures): The data you have provided d
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
```



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

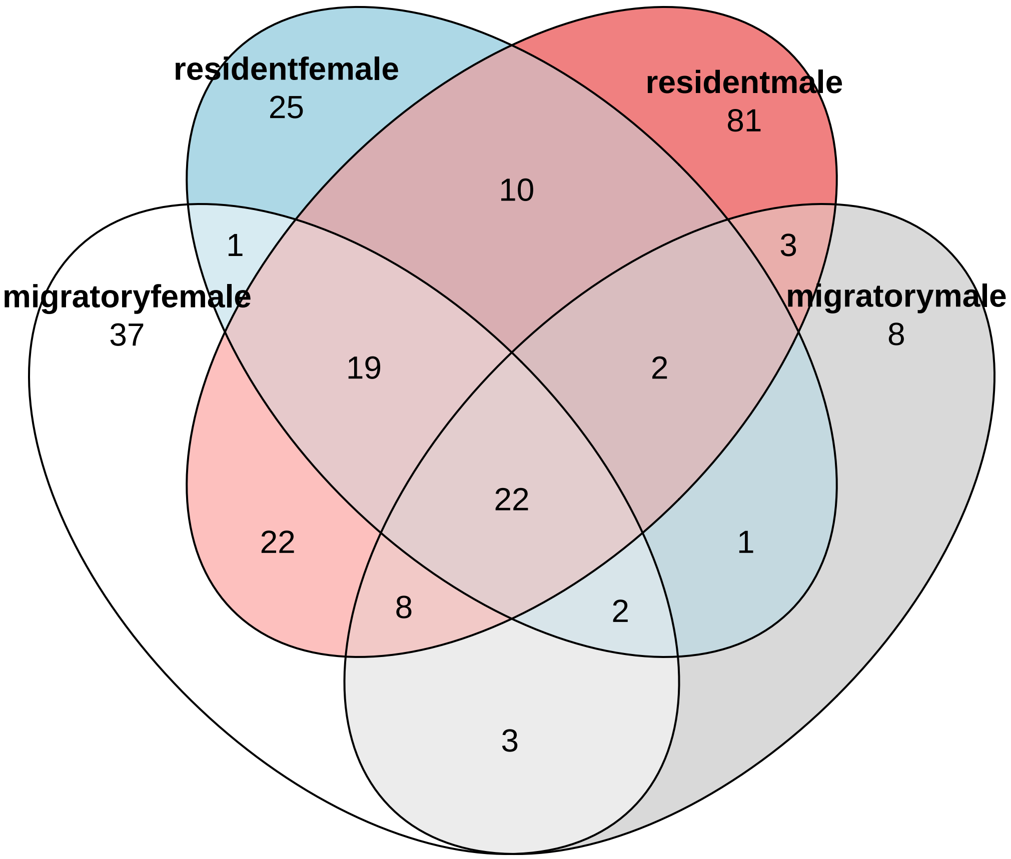
#remove taxa with relative abundance <0.05%

```
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:         [ 1182 taxa and 24 samples ]
## sample_data() Sample Data:       [ 24 samples by 6 sample variables ]
## tax_table()   Taxonomy Table:    [ 1182 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 1182 reference sequences ]
```

#number of shared ASVs birds (found in 25% or more)

```
#create a venn diagram showing the different categories and what they share
flock=ps_venn(
physeqprune,
"flock",
fraction = .25,
weight = FALSE,
relative = TRUE,
plot = TRUE)
flock
```



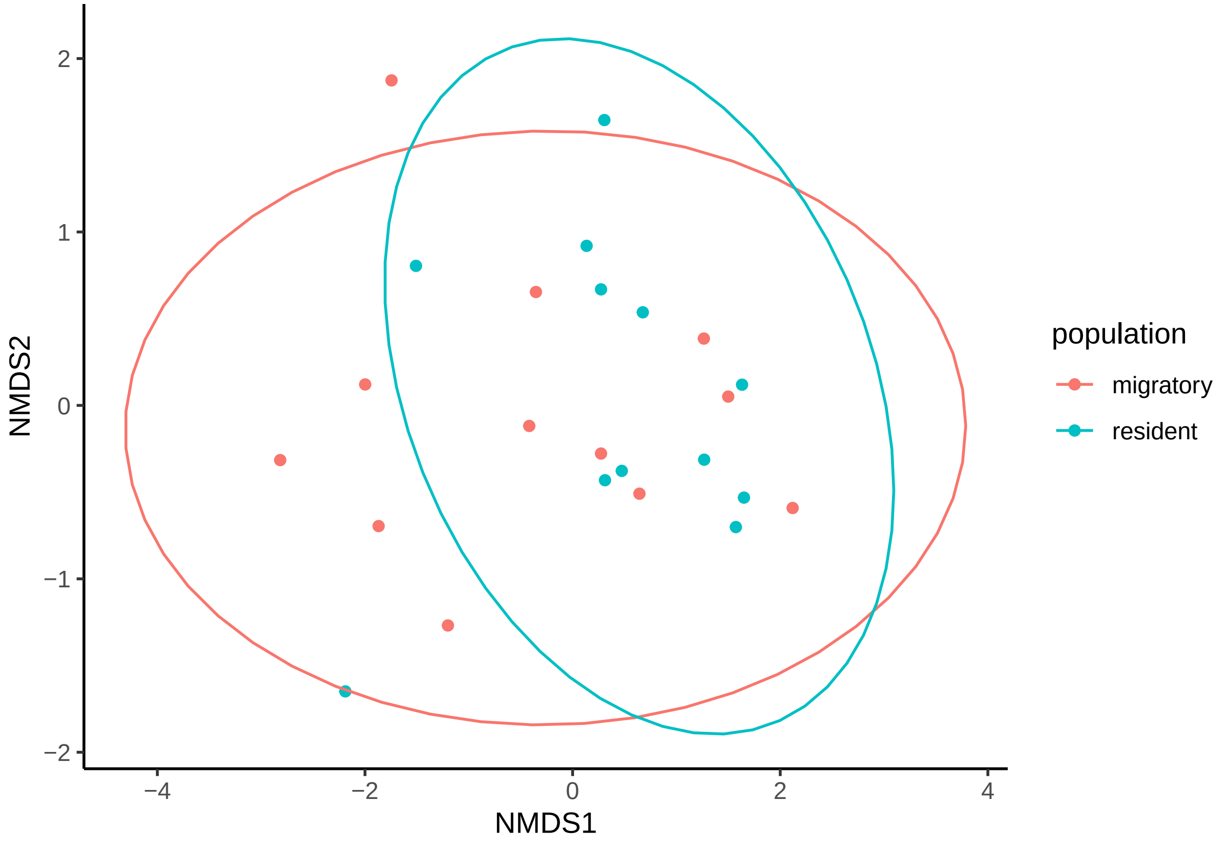
#bray curtis caculation, 0; exactly the same, 1; very diverse

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

```
## Run 0 stress 0.1428942
## Run 1 stress 0.146969
## Run 2 stress 0.1594356
## Run 3 stress 0.1513841
## Run 4 stress 0.14457
## Run 5 stress 0.1428942
## ... Procrustes: rmse 0.0002100465  max resid 0.0006044241
## ... Similar to previous best
## Run 6 stress 0.1525717
## Run 7 stress 0.1588996
## Run 8 stress 0.1573144
## Run 9 stress 0.154528
## Run 10 stress 0.1733832
## Run 11 stress 0.143154
## ... Procrustes: rmse 0.03002238  max resid 0.1221175
## Run 12 stress 0.1574675
## Run 13 stress 0.1500478
## Run 14 stress 0.1602027
## Run 15 stress 0.1431865
## ... Procrustes: rmse 0.02878722  max resid 0.1212331
## Run 16 stress 0.1451435
## Run 17 stress 0.1511038
## Run 18 stress 0.1587388
## Run 19 stress 0.1430943
## ... Procrustes: rmse 0.03226095  max resid 0.1232675
## Run 20 stress 0.1513171
## *** Best solution repeated 1 times
```

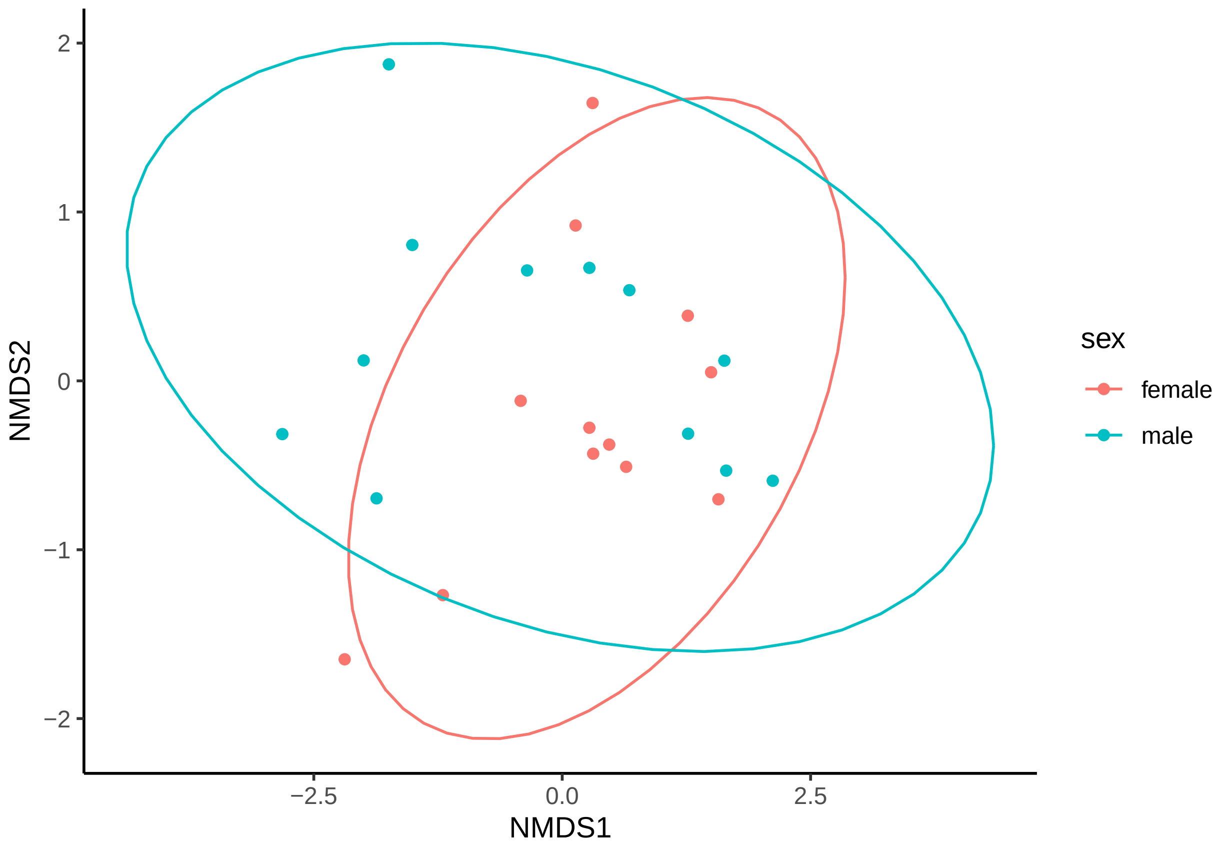
#bray curtis population plot

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



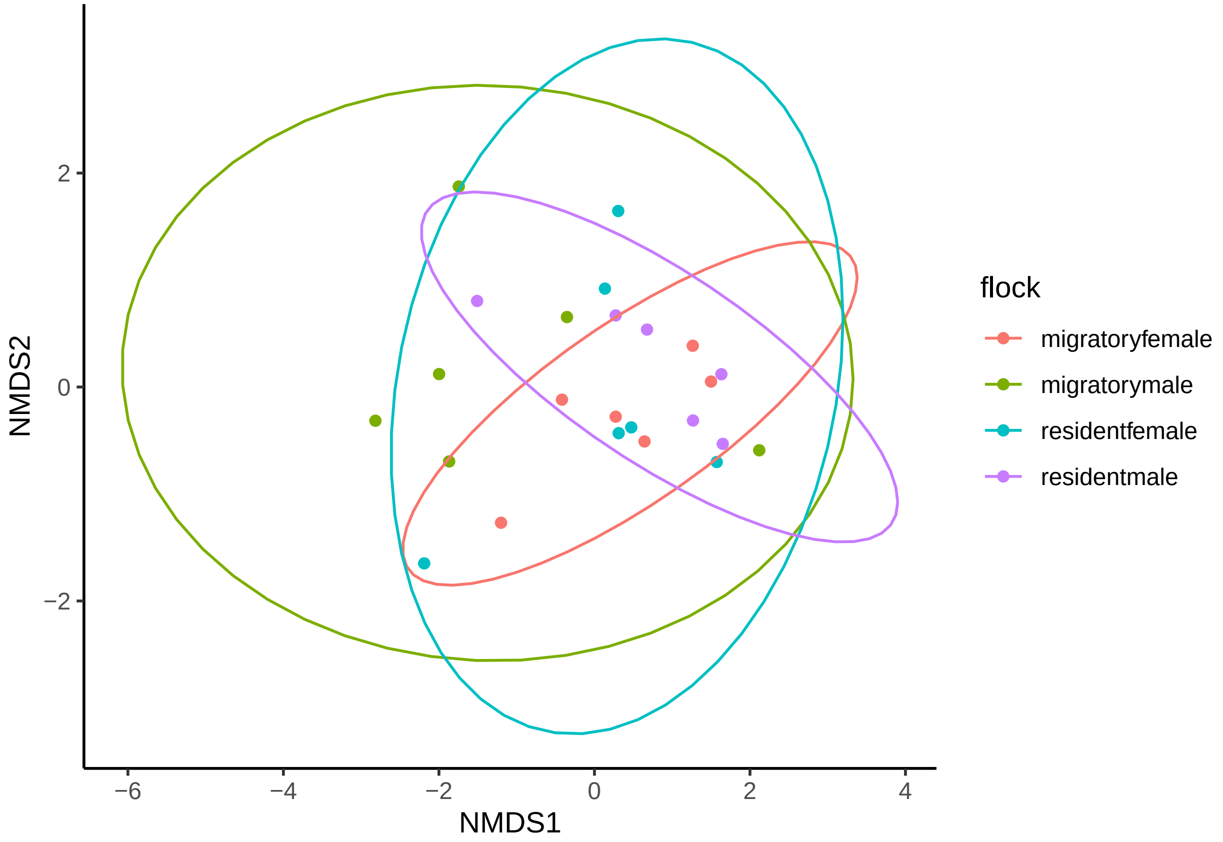
#bray curtis sex plot

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



#bray curtis flock plot

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



#bray curits statistics

```
#population
adonis2(dist ~ sample_data(physeqprune)$population)
```

```
## 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)$population)
##                                     Df SumOfSqs     R2      F Pr(>F)
## sample_data(physeqprune)$population  1   0.4198 0.0445 1.0246  0.372
## Residual                            22   9.0141 0.9555
## Total                               23   9.4339 1.0000
```

```
ps.disper <-betadisper(dist, sample_data(physeqprune)$population)
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     1 0.002974 0.0029738 1.105    999  0.321
## Residuals 22 0.059204 0.0026911
```

```
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##           migratory resident
## migratory               0.31
## resident    0.30457
```

```r
#flock
adonis2(dist ~ sample_data(physeqprune)$flock)
```

```
## 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)$flock)
##                                Df SumOfSqs      R2      F Pr(>F)
## sample_data(physeqprune)$flock  3   1.3074 0.13858 1.0725  0.255
## Residual                       20   8.1265 0.86142
## Total                          23   9.4339 1.00000
```

```r
ps.disper<-betadisper(dist, sample_data(physeqprune)$flock)
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 0.010145 0.0033816 0.7328    999  0.579
## Residuals 20 0.092290 0.0046145
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##                  migratoryfemale migratorymale residentfemale residentmale
## migratoryfemale                        0.17400        0.64800        0.824
## migratorymale            0.16662                      0.37400        0.219
## residentfemale           0.62888       0.38315                       0.772
## residentmale             0.80975       0.19871        0.77911
```

```r
#flock
adonis2(dist ~ sample_data(physeqprune)$flock)
```

```
## 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)$flock)
##                                Df SumOfSqs      R2      F Pr(>F)
## sample_data(physeqprune)$flock  3   1.3074 0.13858 1.0725  0.245
## Residual                       20   8.1265 0.86142
## Total                          23   9.4339 1.00000
```

```
ps.disper<-betadisper(dist, sample_data(physeqprune)$flock)
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 0.010145 0.0033816 0.7328    999  0.541
## Residuals 20 0.092290 0.0046145
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##                 migratoryfemale migratorymale residentfemale residentmale
## migratoryfemale                       0.14700        0.61100        0.802
## migratorymale           0.16662                      0.38700        0.201
## residentfemale          0.62888       0.38315                       0.767
## residentmale            0.80975       0.19871        0.77911
```

##Question 10; Alpha diversity visualizes the relative abundances of taxas where as beta diversity observes the identity of each taxa in sample.

##Question 12; alpha diversity was plotted for variation in taxa between population, sex, and flock. Resident male showed to have the most alpha diversity when comparing relative abundance between flocks. Only 22 taxas where shared between flocks. Observed features measures the number of bacterial species present in the sample where the Shannon index takes into account the the relative abundance of each species present in the sample.

##Question 13; The bay curtis plot displayed that not much diversity was unique between population, sex, and flock. This indicates that taxa identiy were similair when taking into account these variables tested.

##Question 14; Performing the statistical analysis, this confirmed no significance between betaa diversity due to p values > 0.05.