

# deseq2\_analysis.R

*arisely*

*Thu Oct 26 11:01:45 2017*

```
## R script for publication: "Active migration associated with repeatable and specific changes
  to gut microbiota in migratory shorebirds"
# Alice Risely, David Waite, Beata Ujvari, Bethany Hoyer, & Marcel Klaassen

##R version 3.4.1

##workflows:
#https://f1000research.com/articles/5-1492/v1
#http://deneflab.github.io/MicrobeMiseq/demos/mothur_2_phyloseq.html
#DESeq2
#https://www.bioconductor.org/packages/devel/bioc/vignettes/phyloseq/inst/doc/phyloseq-mixture
-models.html

##good threads on data transformation
#https://github.com/joey711/phyloseq/issues/492
#https://github.com/joey711/phyloseq/issues/283

#To download phyloseq and DESeq2 from bioconductor run following two lines:

#source('http://bioconductor.org/biocLite.R')
#biocLite('phyloseq')
#biocLite("DESeq2")

#It might mess up some other packages, so packages may need reinstalling (either from CRAN or
Bioconductor) as errors come up when loading phyloseq and DESeq2 into library
#I had to reinstall ~10 packages

##setwd() somewhere and download all github files to this location

library(phyloseq)
library(DESeq2)
```

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

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

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

```
## Loading required package: parallel
```

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

```
## The following objects are masked from 'package:parallel':  
##  
##   clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,  
##   clusterExport, clusterMap, parApply, parCapply, parLapply,  
##   parLapplyLB, parRapply, parSapply, parSapplyLB
```

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

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

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

```
## The following object is masked from 'package:base':  
##  
##   expand.grid
```

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

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

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

```
## Loading required package: GenomicRanges
```

```
## Loading required package: GenomeInfoDb
```

```
## Loading required package: SummarizedExperiment
```

```
## 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:phyloseq':
##
## sampleNames
```

```
## Loading required package: DelayedArray
```

```
## Loading required package: matrixStats
```

```
## Warning: package 'matrixStats' was built under R version 3.4.2
```

```
##
## Attaching package: 'matrixStats'
```

```
## The following objects are masked from 'package:Biobase':
##
## anyMissing, rowMedians
```

```
##
## Attaching package: 'DelayedArray'
```

```
## The following objects are masked from 'package:matrixStats':
##
## colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
```

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

```
##other packages you may need, althoguh I think a few are included with the phyloseq package s
o you may not need to load them
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```
library(vegan)
```

```
## Warning: package 'vegan' was built under R version 3.4.2
```

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

```
## Warning: package 'permute' was built under R version 3.4.2
```

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

```
## This is vegan 2.4-4
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

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

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

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

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

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

```
## 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(scales)
```

```
## Warning: package 'scales' was built under R version 3.4.2
```

```
library(grid)  
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 3.4.2
```

```
library(ape)  
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.4.2
```

```
##  
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':  
##  
##      combine
```

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

```
## The following object is masked from 'package:BiocGenerics':  
##  
##      combine
```

```
library(ade4)
```

```
## Warning: package 'ade4' was built under R version 3.4.2
```

```
##
```

```
## Attaching package: 'ade4'
```

```
## The following object is masked from 'package:GenomicRanges':  
##  
##      score
```

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

```
## The following object is masked from 'package:BiocGenerics':  
##  
##      score
```

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.4.2
```

```
## -----
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)
```

```
## -----
```

```
##  
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':  
##  
##      arrange, count, desc, failwith, id, mutate, rename, summarise,  
##      summarize
```

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

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

```
## The following object is masked from 'package:S4Vectors':  
##  
##      rename
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.4.2
```

```
##  
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:reshape2':  
##  
## smiths
```

```
## The following object is masked from 'package:S4Vectors':  
##  
## expand
```

```
library(data.table)
```

```
##  
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:reshape2':  
##  
## dcast, melt
```

```
## The following objects are masked from 'package:dplyr':  
##  
## between, first, last
```

```
## The following object is masked from 'package:SummarizedExperiment':  
##  
## shift
```

```
## The following object is masked from 'package:GenomicRanges':  
##  
## shift
```

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

```
## The following objects are masked from 'package:S4Vectors':  
##  
## first, second
```

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.4.2
```

```
source("miseqR.R")
```

```
theme_set(theme_bw())
```

```
##### ONLY OTUS WITH OVER 10 SEQUENCES (AND WITH PHYLO TREE) #####
#####
```

```
setwd("C:\\Users\\arisely\\Dropbox\\PhD\\Microbiome\\CHAPTER 3 - PHYSIOLOGY\\ANALYSIS\\R\\Phyl
oseq analysis")
```

```
# Assign variables for imported data
```

```
sharedfile = "shorebird.microbiome.shared"
```

```
taxfile = "shorebird.microbiome.taxonomy"
```

```
##import mothur shared and tax files
```

```
mothur_data <- import_mothur(mothur_shared_file = sharedfile,
                             mothur_constaxonomy_file = taxfile)
```

```
##import metadata
```

```
map <- read.csv("shorebird.metadata.csv", header=T, row.names=1)
```

```
##make meta data into phyloseq format
```

```
map <- sample_data(map)
```

```
str(map)
```

```
## 'data.frame': 188 obs. of 21 variables:
```

```
## Formal class 'sample_data' [package "phyloseq"] with 4 slots
```

```
## ..@ .Data :List of 21
```

```
## .. ..$ : Factor w/ 180 levels "8546","8548",...: 1 2 3 4 5 6 7 8 9 10 ...
```

```
## .. ..$ : Factor w/ 6 levels "Curlew sandpiper",...: 4 4 4 4 4 4 4 4 4 4 ...
```

```
## .. ..$ : Factor w/ 5 levels "", "Broome", "Flinders",...: 2 2 2 2 2 2 2 2 2 2 ...
```

```
## .. ..$ : Factor w/ 7 levels "", "August", "December",...: 2 2 2 2 2 2 2 2 2 2 ...
```

```
## .. ..$ : Factor w/ 6 levels "", "Adult", "Juvenile",...: 5 5 2 5 5 5 5 5 5 2 ...
```

```
## .. ..$ : Factor w/ 4 levels "", "one", "three",...: 4 4 4 4 4 4 4 4 4 4 ...
```

```
## .. ..$ : Factor w/ 161 levels "", "3611876", "3648481",...: 20 21 10 22 23 11 12 13 14 15 ..
```

```
.
```

```
## .. ..$ : Factor w/ 3 levels "bird","env","neg": 1 1 1 1 1 1 1 1 1 1 ...
```

```
## .. ..$ : Factor w/ 3 levels "na","No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
```

```
## .. ..$ : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

```
## .. ..$ : int NA NA NA NA NA NA NA NA NA NA ...
```

```
## .. ..$ : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

```
## .. ..$ : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

```
## .. ..$ : int NA NA NA NA NA NA NA NA NA NA ...
```

```
## .. ..$ : num NA NA NA NA NA NA NA NA NA NA ...
```



```
## .. ..$ : num NA NA NA NA NA NA NA NA NA NA NA ...
## .. ..$ : Factor w/ 3 levels "", "sink", "source": 1 1 1 1 1 1 1 1 1 1 1 ...
## .. ..$ : Factor w/ 4 levels "", "env", "gut", ...: 1 1 1 1 1 1 1 1 1 1 1 ...
## .. ..$ : Factor w/ 10 levels "1/04/2017", "1/10/2016", ...: 6 6 6 6 6 6 6 6 6 6 6 ...
## .. ..$ : int 2 2 2 2 2 2 2 2 2 2 2 ...
## .. ..$ : Factor w/ 9 levels "", "Broome-curlewsand-adult", ...: 5 5 4 5 5 5 5 5 5 4 ...
## ..@ names : chr "Group" "species" "site" "month" ...
## ..@ row.names: chr "8546" "8548" "8549" "8551" ...
## ..@ .S3Class : chr "data.frame"
```

head(map)

```
##      Group      species  site month      age group bird_id type
## 8546  8546 Red-necked stint Broome August Second year    two 3674227 bird
## 8548  8548 Red-necked stint Broome August Second year    two 3674229 bird
## 8549  8549 Red-necked stint Broome August      Adult    two 3671906 bird
## 8551  8551 Red-necked stint Broome August Second year    two 3674231 bird
## 8552  8552 Red-necked stint Broome August Second year    two 3674232 bird
## 8556  8556 Red-necked stint Broome August Second year    two 3674214 bird
##      Recap recap_delete Recap_num replicate remove_replicate weight wing
## 8546    No              No        NA        No              No    NA    NA
## 8548    No              No        NA        No              No    NA    NA
## 8549    No              No        NA        No              No    NA    NA
## 8551    No              No        NA        No              No    NA    NA
## 8552    No              No        NA        No              No    NA    NA
## 8556    No              No        NA        No              No    NA    NA
##      hb SourceSink Env      Date Plate      migration
## 8546 NA              22/08/2015    2    Broome-stint-SY
## 8548 NA              22/08/2015    2    Broome-stint-SY
## 8549 NA              22/08/2015    2 Broome-stint-adult
## 8551 NA              22/08/2015    2    Broome-stint-SY
## 8552 NA              22/08/2015    2    Broome-stint-SY
## 8556 NA              22/08/2015    2    Broome-stint-SY
```

tail(map)

```
##      Group species  site  month age group bird_id type Recap
## F1    Flind_env  env Flinders  March  na      na  env    No
## F2    Flind_env  env Flinders  March  na      na  env    No
## Neg      Neg1   neg    na      na  na      na  neg    na
## Neg2      Neg2   neg              neg    No
## WTP1    WTP_env  env      WTP December  na      na  env    No
## WTP2    WTP_env  env      WTP December  na      na  env    No
##      recap_delete Recap_num replicate remove_replicate weight wing hb
## F1              No        NA      Yes              No    NA    NA NA
## F2              No        NA      Yes              No    NA    NA NA
## Neg              No        NA      No              No    NA    NA NA
## Neg2             No        NA      No              No    NA    NA NA
## WTP1             No        NA      No              No    NA    NA NA
## WTP2             No        NA      No              No    NA    NA NA
##      SourceSink Env      Date Plate migration
```

```
## F1      sink env 11/03/2016      1
## F2      sink env 11/03/2016      1
## Neg     sink neg  1/10/2016      1
## Neg2    1/04/2017      2
## WTP1    sink env 29/12/2015      1
## WTP2    sink env 29/12/2015      1
```

```
##merge the metadata into the phyloseq object

moth_merge <- merge_phyloseq(mothur_data, map)

##import phylogenetic tree

tree<-read.tree("shorebird.microbiome.tree")

##make tree into phyloseq format

tree<-phy_tree(tree)

##merge tree into current phyloseq object so all object now contained in moth_merge

moth_merge <-merge_phyloseq(moth_merge, tree)

moth_merge
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:           [ 8791 taxa and 188 samples ]
## sample_data() Sample Data:        [ 188 samples by 21 sample variables ]
## tax_table()   Taxonomy Table:      [ 8791 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree:  [ 8791 tips and 8789 internal nodes ]
```

```
##everything looks fine and no errors!

##rename columns to relevant taxonomic group

colnames(tax_table(moth_merge))
```

```
## [1] "Rank1" "Rank2" "Rank3" "Rank4" "Rank5" "Rank6"
```

```
colnames(tax_table(moth_merge)) <- c("Kingdom", "Phylum", "Class",
                                     "Order", "Family", "Genus")

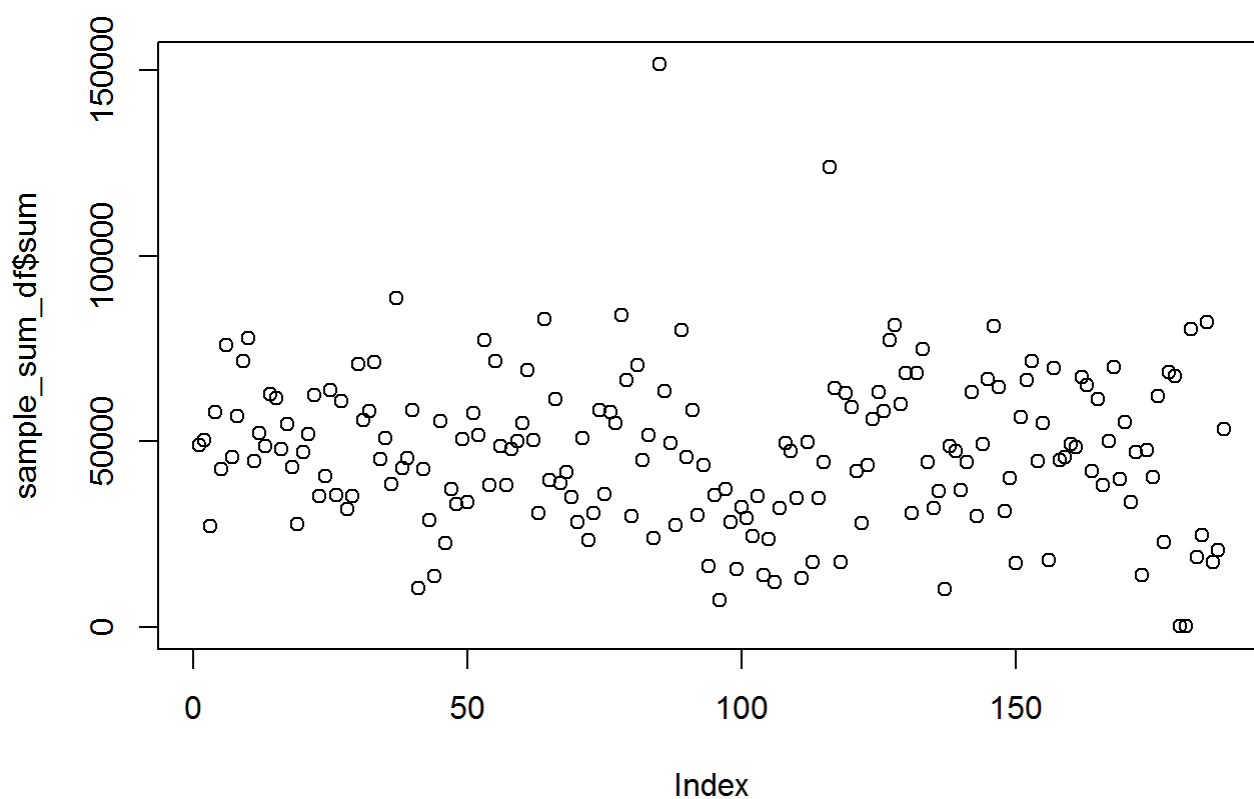
data_all <- moth_merge %>%
  subset_taxa(
    Kingdom == "Bacteria" &
    Family  != "mitochondria" &
    Class   != "Chloroplast"
  )

data_all
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:             [ 8791 taxa and 188 samples ]
## sample_data() Sample Data:          [ 188 samples by 21 sample variables ]
## tax_table()   Taxonomy Table:        [ 8791 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree:     [ 8791 tips and 8789 internal nodes ]
```

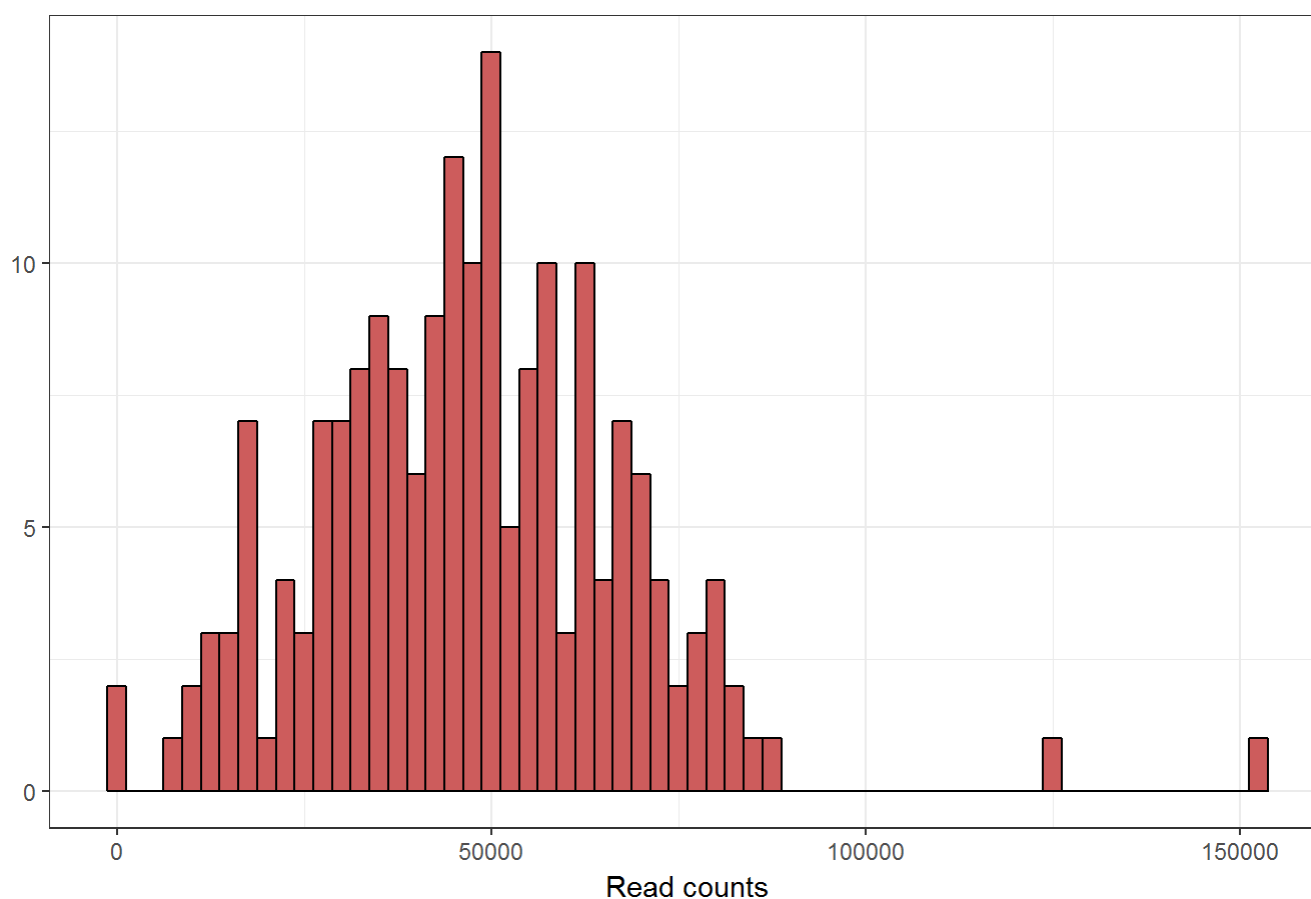
```
#####look at read library
```

```
sample_sum_df <- data.frame(sum = sample_sums(data_all))
plot(sample_sum_df$sum)
```



```
ggplot(sample_sum_df, aes(x = sum)) +
  geom_histogram(color = "black", fill = "indianred", binwidth = 2500) +
  ggtitle("Distribution of sample sequencing depth") +
  xlab("Read counts") +
  ylab("Frequency")+
  theme(axis.title.y = element_blank())
```

## Distribution of sample sequencing depth



```
#####
#####
```

```
##subset samples to just birds (there are environmental samples in here too)
```

```
data_all<-data_all %>%
  subset_samples(Group != "Env_Broome_A")
```

```
data_all<-data_all %>%
  subset_samples(Group != "Env_Broome_E")
```

```
data_all<-prune_taxa(taxa_sums(data_all)>0, data_all)
```

```
## remove OTUs that are common in the neg controls (Neg = just lab control, Neg2 = field and 1
ab control)
```

```
neg_control<-data_all %>% subset_samples(type == "neg")
```

```
neg_control <- prune_taxa(taxa_sums(neg_control) > 5, neg_control)
neg_control
```

```
## phyloseq-class experiment-level object
```

```
## otu_table() OTU Table: [ 97 taxa and 2 samples ]
```

```
## sample_data() Sample Data: [ 2 samples by 21 sample variables ]
```

```
## tax_table() Taxonomy Table: [ 97 taxa by 6 taxonomic ranks ]
```

```
## phy_tree()      Phylogenetic Tree: [ 97 tips and 96 internal nodes ]
```

```
## 97 OTUs which have over 1 reads

## we should get rid of these

badtaxa<-taxa_names(neg_control)
alltaxa<-taxa_names(data_all)
alltaxa1 <- alltaxa[!(alltaxa %in% badtaxa)]

data_all = prune_taxa(alltaxa1, data_all)

##get rid of negative control altogether

data_all<-data_all%>%subset_samples(species!="Neg")

#delete replicates

data_all<-data_all%>%subset_samples(remove_replicate=="No")

## just birds

birds<-data_all%>%subset_samples(type=="bird")
birds <- prune_taxa(taxa_sums(birds) > 0, birds)

#####

## data has more samples than we need. We are going to prune the dataset to just the relevant
samples for this analysis

## for migration analysis (migrant/resident comparison)

migration<-birds%>%subset_samples(species=="Red-necked stint"|species=="Curlew sandpiper")
migration<-migration%>%subset_samples(month=="August"|month=="September")
migration <- prune_taxa(taxa_sums(migration) > 0, migration)
migration
```

```
## phyloseq-class experiment-level object
## otu_table()      OTU Table:          [ 5262 taxa and 77 samples ]
## sample_data()    Sample Data:       [ 77 samples by 21 sample variables ]
## tax_table()      Taxonomy Table:     [ 5262 taxa by 6 taxonomic ranks ]
## phy_tree()      Phylogenetic Tree: [ 5262 tips and 5260 internal nodes ]
```

```
## 77 samples, 5262 OTUs

##### DESeq 2 #####

## Will do this analysis first (for Fig2 and Table S3) before rarefying data

## Analyse differential abundances between migrants and residents for each group
```

```
## subset RNS (Red-necked stint) in Broome

rns_br<-migration%>%subset_samples(species=="Red-necked stint")
rns_br<-rns_br%>%subset_samples(site=="Broome")
rns_br <- prune_taxa(taxa_sums(rns_br) > 0, rns_br)

#####

##the following analysis is from the Phyloseq-DESeq2 vignette, linked at top of script.

diagdds = phyloseq_to_deseq2(rns_br, ~ age)
```

```
## converting counts to integer mode
```

```
# calculate geometric means prior to estimate size factors
gm_mean = function(x, na.rm=TRUE){
  exp(sum(log(x[x > 0])), na.rm=na.rm) / length(x))
}
geoMeans = apply(counts(diagdds), 1, gm_mean)
diagdds = estimateSizeFactors(diagdds, geoMeans = geoMeans)
diagdds = DESeq(diagdds, fitType="local")
```

```
## using pre-existing size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
## -- replacing outliers and refitting for 985 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
```

```
## estimating dispersions
```

```
## fitting model and testing
```

```
#diagdds = DESeq(diagdds, fitType="parametric")

res = results(diagdds, cooksCutoff = FALSE)
```

```
alpha = 0.01
sigtab = res[which(res$padj < alpha), ]
sigtab_rns.br = cbind(as(sigtab, "data.frame"), as(tax_table(rns_br)[rownames(sigtab), ], "matrix"))
head(sigtab_rns.br)
```

##	baseMean	log2FoldChange	lfcSE	stat	pvalue
## Otu000765	8.943807	6.451209	1.473355	4.378585	1.194523e-05
## Otu000279	12.561703	6.487397	1.733635	3.742078	1.825051e-04
## Otu000053	448.890074	4.714085	1.207552	3.903836	9.468006e-05
## Otu000013	1452.245588	-10.263594	1.174882	-8.735852	2.418274e-18
## Otu000117	43.826023	-8.519601	1.381634	-6.166322	6.989664e-10
## Otu000009	451.397495	-11.616548	1.627590	-7.137267	9.520456e-13
##	padj	Kingdom	Phylum	Class	
## Otu000765	5.492657e-04	Bacteria	Proteobacteria	Gammaproteobacteria	
## Otu000279	4.994002e-03	Bacteria	Proteobacteria	Gammaproteobacteria	
## Otu000053	3.562337e-03	Bacteria	Bacteroidetes	Sphingobacteriia	
## Otu000013	3.639502e-16	Bacteria	Actinobacteria	Actinobacteria	
## Otu000117	5.259722e-08	Bacteria	Actinobacteria	Actinobacteria	
## Otu000009	9.552191e-11	Bacteria	Actinobacteria	Actinobacteria	
##		Order		Family	
## Otu000765		Pseudomonadales		Moraxellaceae	
## Otu000279		Pseudomonadales		Moraxellaceae	
## Otu000053		Sphingobacteriales		Chitinophagaceae	
## Otu000013		Corynebacteriales		Corynebacteriaceae	
## Otu000117		Corynebacteriales		Corynebacteriaceae	
## Otu000009		Corynebacteriales	Corynebacteriales_unclassified		
##		Genus			
## Otu000765		Acinetobacter			
## Otu000279		Acinetobacter			
## Otu000053		Vibrionimonas			
## Otu000013		Corynebacteriaceae_unclassified			
## Otu000117		Corynebacteriaceae_unclassified			
## Otu000009		Corynebacteriales_unclassified			

```
sigtab_rns.br<- sigtab_rns.br[order(sigtab_rns.br$log2FoldChange),]

#add variable to end
sigtab_rns.br$Group<-"Red-necked stint Broome"
#####

#####
#

##RNS in Vic

rns_vic<-migration%>%subset_samples(species=="Red-necked stint")
rns_vic<-rns_vic%>%subset_samples(month=="September")
rns_vic <- prune_taxa(taxa_sums(rns_vic) > 0, rns_vic)

#####
```

```
diagdds = phyloseq_to_deseq2(rns_vic, ~ age)
```

```
## converting counts to integer mode
```

```
# calculate geometric means prior to estimate size factors
gm_mean = function(x, na.rm=TRUE){
  exp(sum(log(x[x > 0])), na.rm=na.rm) / length(x))
}
geoMeans = apply(counts(diagdds), 1, gm_mean)
diagdds = estimateSizeFactors(diagdds, geoMeans = geoMeans)
diagdds = DESeq(diagdds, fitType="local")
```

```
## using pre-existing size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
## -- replacing outliers and refitting for 1034 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
```

```
## estimating dispersions
```

```
## fitting model and testing
```

```
#diagdds = DESeq(diagdds, fitType="parametric")

res = results(diagdds, cooksCutoff = FALSE)
alpha = 0.01
sigtab = res[which(res$padj < alpha), ]
sigtab_rns.vic = cbind(as(sigtab, "data.frame"), as(tax_table(rns_vic)[rownames(sigtab), ], "matrix"))
head(sigtab_rns.vic)
```

```
##           baseMean log2FoldChange      lfcSE      stat      pvalue
## Otu000013   310.74839        -7.603356 0.9299704 -8.175912 2.936371e-16
## Otu000117   11.77106         -6.999681 1.2864956 -5.440890 5.301508e-08
## Otu000009  6020.41474        -9.169087 0.9653096 -9.498597 2.127372e-21
```



```
## Otu000575      17.49598      -7.299684 1.2980066 -5.623765 1.868398e-08
## Otu000265      12.17622      -7.131328 1.6612481 -4.292753 1.764711e-05
## Otu000757      13.11934      -6.617123 1.3422332 -4.929936 8.225670e-07
##                padj  Kingdom                Phylum                Class
## Otu000013 1.642900e-13 Bacteria Actinobacteria Actinobacteria
## Otu000117 1.483097e-05 Bacteria Actinobacteria Actinobacteria
## Otu000009 2.380529e-18 Bacteria Actinobacteria Actinobacteria
## Otu000575 6.969125e-06 Bacteria Actinobacteria Actinobacteria
## Otu000265 2.194124e-03 Bacteria Actinobacteria Actinobacteria
## Otu000757 1.534087e-04 Bacteria Actinobacteria Actinobacteria
##                Order                Family
## Otu000013 Corynebacteriales          Corynebacteriaceae
## Otu000117 Corynebacteriales          Corynebacteriaceae
## Otu000009 Corynebacteriales Corynebacteriales_unclassified
## Otu000575 Corynebacteriales          Corynebacteriaceae
## Otu000265 Corynebacteriales          Corynebacteriaceae
## Otu000757 Corynebacteriales          Corynebacteriaceae
##                Genus
## Otu000013 Corynebacteriaceae_unclassified
## Otu000117 Corynebacteriaceae_unclassified
## Otu000009 Corynebacteriales_unclassified
## Otu000575 Corynebacteriaceae_unclassified
## Otu000265 Corynebacteriaceae_unclassified
## Otu000757 Corynebacterium
```

```
sigtab_rns.vic<- sigtab_rns.vic[order(sigtab_rns.vic$log2FoldChange),]

sigtab_rns.vic$Group<-"Red-necked stint Flinders"

#####

#curlew sandpiper

cs<-migration%>%subset_samples(species=="Curlew sandpiper")
cs <- prune_taxa(taxa_sums(cs) > 0, cs)

#####

diagdds = phyloseq_to_deseq2(cs, ~ age)
```

```
## converting counts to integer mode
```

```
# calculate geometric means prior to estimate size factors
gm_mean = function(x, na.rm=TRUE){
  exp(sum(log(x[x > 0])), na.rm=na.rm) / length(x))
}
geoMeans = apply(counts(diagdds), 1, gm_mean)
diagdds = estimateSizeFactors(diagdds, geoMeans = geoMeans)
diagdds = DESeq(diagdds, fitType="local")
```

```
## using pre-existing size factors
```

```
## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## -- replacing outliers and refitting for 509 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing

#diagdds = DESeq(diagdds, fitType="parametric")

res = results(diagdds, cooksCutoff = FALSE)
alpha = 0.01
sigtab = res[which(res$padj < alpha), ]
sigtab_cs = cbind(as(sigtab, "data.frame"), as(tax_table(cs)[rownames(sigtab), ], "matrix"))
head(sigtab_cs)

##          baseMean log2FoldChange    lfcSE      stat      pvalue
## Otu000319   36.83220      -23.122692 3.172221 -7.289117 3.119939e-13
## Otu000017  211.66346       6.298750 1.498019  4.204720 2.614058e-05
## Otu000023  648.40735      -8.741014 1.963501 -4.451749 8.517358e-06
## Otu000134   17.35786     -22.100447 3.161498 -6.990498 2.739122e-12
## Otu000157   17.97271     -22.155554 3.172759 -6.983056 2.888269e-12
## Otu000028  177.39850      -7.319617 2.008101 -3.645044 2.673464e-04
##                padj   Kingdom          Phylum          Class
## Otu000319 1.164777e-11  Bacteria  Proteobacteria  Betaproteobacteria
## Otu000017 4.182493e-04  Bacteria  Proteobacteria  Gammaproteobacteria
## Otu000023 1.467606e-04  Bacteria  Actinobacteria    Actinobacteria
## Otu000134 7.188580e-11  Bacteria  Actinobacteria    Actinobacteria
## Otu000157 7.188580e-11  Bacteria  Actinobacteria    Actinobacteria
## Otu000028 3.522682e-03  Bacteria  Actinobacteria    Actinobacteria
##                Order          Family
## Otu000319   Rhodocyclales    Rhodocyclaceae
## Otu000017   Aeromonadales    Succinivibrionaceae
## Otu000023   Corynebacteriales Corynebacteriaceae
## Otu000134   Corynebacteriales Corynebacteriaceae
## Otu000157   Corynebacteriales Corynebacteriaceae
## Otu000028   Corynebacteriales Corynebacteriaceae
```

```
##                               Genus
## Otu000319                     Methyloversatilis
## Otu000017 Succinivibrionaceae_unclassified
## Otu000023 Corynebacteriaceae_unclassified
## Otu000134 Corynebacteriaceae_unclassified
## Otu000157                     Corynebacterium
## Otu000028                     <NA>
```

```
sigtab_cs<- sigtab_cs[order(sigtab_cs$log2FoldChange),]

sigtab_cs$Group<-"Curlew sandpiper Broome"

#####

##plot all together
##bind objects

deseq2<-rbind(sigtab_rns.br, sigtab_rns.vic, sigtab_cs)

#edit small details for figure

deseq2[3,11]<-"Corynebacteriaceae"
deseq2[12,11]<-"Corynebacteriaceae"
revalue(deseq2$Family, c("Gammaproteobacteria_unclassified"="Gammaproteobacteria uncl."))
```

```
##           Otu000118           Otu000028
##   Corynebacteriaceae   Corynebacteriaceae
##           Otu000009           Otu000013
##   Corynebacteriaceae   Corynebacteriaceae
##           Otu000117           Otu000011
##   Corynebacteriaceae   Helicobacteraceae
##           Otu000053           Otu000106
##   Chitinophagaceae     Lachnospiraceae
##           Otu000765           Otu000279
##   Moraxellaceae        Moraxellaceae
##           Otu000358           Otu000009
##   Gammaproteobacteria uncl.   Corynebacteriaceae
##           Otu000013           Otu000575
##   Corynebacteriaceae   Corynebacteriaceae
##           Otu000265           Otu000117
##   Corynebacteriaceae   Corynebacteriaceae
##           Otu000757           Otu000769
##   Corynebacteriaceae   Corynebacteriaceae
##           Otu000136           Otu000028
##   Corynebacteriaceae   Corynebacteriaceae
##           Otu000020           Otu000006
##   Enterococcaceae      Helicobacteraceae
##           Otu000319           Otu000157
##   Rhodocyclaceae       Corynebacteriaceae
##           Otu000134           Otu000023
```

```
##      Corynebacteriaceae      Corynebacteriaceae
##      Otu000028              Otu000044
##      Corynebacteriaceae      Ruminococcaceae
##      Otu000017              Otu000070
##      Succinivibrionaceae      Lachnospiraceae
##      Otu000124              Otu000126
##      Lachnospiraceae      Ruminococcaceae
##      Otu000227              Otu000208
##      Peptostreptococcaceae      Clostridiaceae_1
##      Otu000078              Otu000628
##      Campylobacteraceae      Peptostreptococcaceae
##      Otu000358              Otu000343
##      Gammaproteobacteria uncl.      Lachnospiraceae
##      Otu000094              Otu000316
##      Mycoplasmataceae      Lachnospiraceae
##      Otu000183
##      Clostridiaceae_1
## 15 Levels: Chitinophagaceae ... Succinivibrionaceae
```

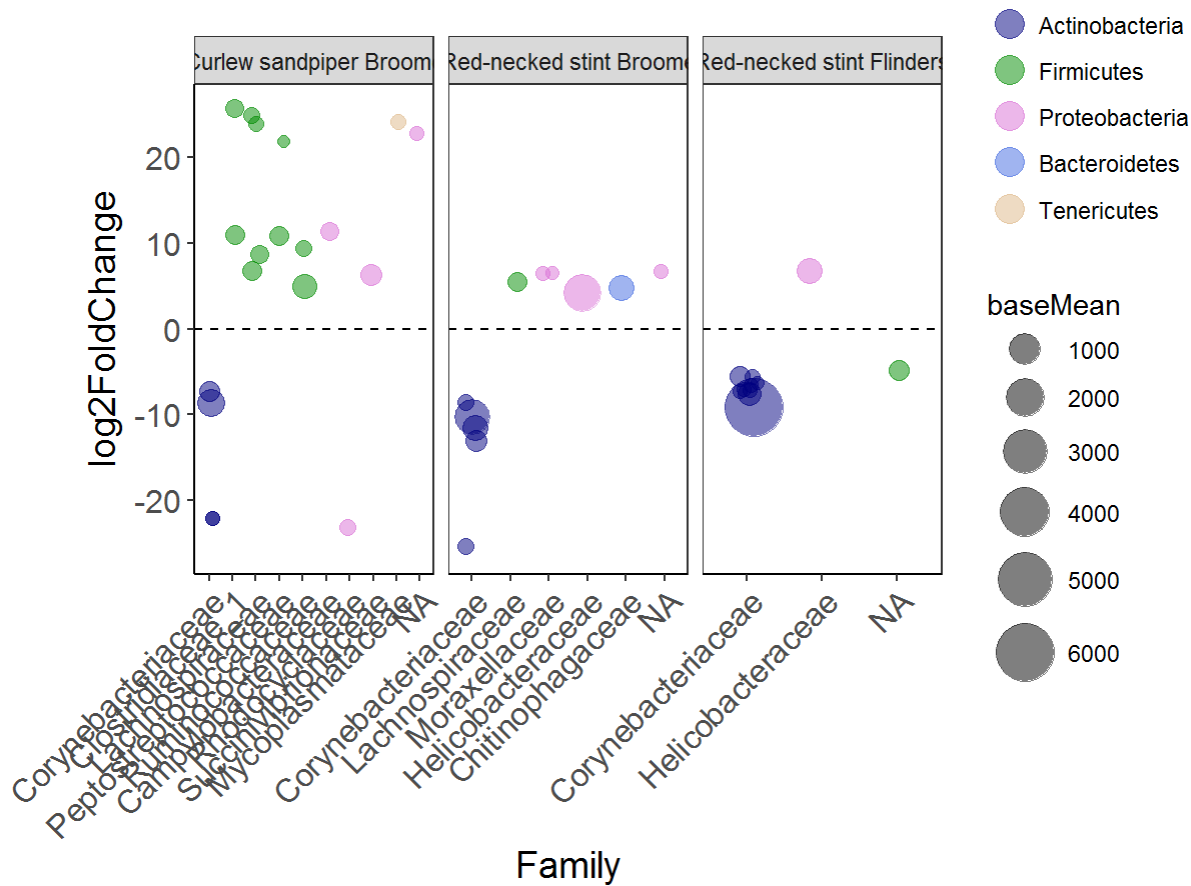
```
deseq2$Family<-factor(deseq2$Family, levels = c("Corynebacteriaceae",
                                                "Clostridiaceae_1","Lachnospiraceae" ,"Peptos
treptococcaceae", "Ruminococcaceae",

                                                "Moraxellaceae",
                                                "Campylobacteraceae" ,
                                                "Helicobacteraceae" ,
                                                "Rhodocyclaceae" ,
                                                "Gammaproteobacteria uncl.",
                                                "Succinivibrionaceae",
                                                "Chitinophagaceae" ,
                                                "Mycoplasmataceae"))
```

```
deseq2$Phylum<-factor(deseq2$Phylum, levels = c("Actinobacteria","Firmicutes", "Proteobacteria
","Bacteroidetes","Tenericutes"))
```

#Figure 2

```
ggplot(deseq2, aes(x=Family, y=log2FoldChange, color=Phylum)) +
  geom_jitter(aes(size = baseMean), width = 0.2, alpha = 0.5)+ scale_size(range = c(2,10))+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  geom_hline(yintercept = 0, linetype = 2)+facet_grid(.~Group, scales = "free")+ylim(-26,26)+
  scale_color_manual(values = c("navyblue","green4","orchid","royalblue","burlywood"))+
  theme( axis.text = element_text(size=12), axis.title=element_text(size=14))+
  theme(plot.margin=unit(c(0.5,1,1,2),"cm"))+guides(color = guide_legend(override.aes = list(s
ize=5)))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"))
```



```
#####

##Now rest of analyses

migration1 <-migration

##rarefy
migration <-rarefy_even_depth(migration1, sample.size = min(sample_sums(migration1)),
                               rngseed = 3, replace = TRUE, trimOTUs = TRUE, verbose = TRUE)
```

```
## `set.seed(3)` was used to initialize repeatable random subsampling.
```

```
## Please record this for your records so others can reproduce.
```

```
## Try `set.seed(3); .Random.seed` for the full vector
```

```
## ...
```

```
## 8560TUs were removed because they are no longer
## present in any sample after random subsampling
```

```
## ...
```

```
migration
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:           [ 4406 taxa and 77 samples ]
## sample_data() Sample Data:        [ 77 samples by 21 sample variables ]
## tax_table()   Taxonomy Table:      [ 4406 taxa by 6 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree:  [ 4406 tips and 4404 internal nodes ]
```

```
migration2 <- migration1 %>%
  scale_reads(n=5815) ##scale to smallest library size

#migration1 = unrarefied
#migration = rarefied
#migration2 = scaled

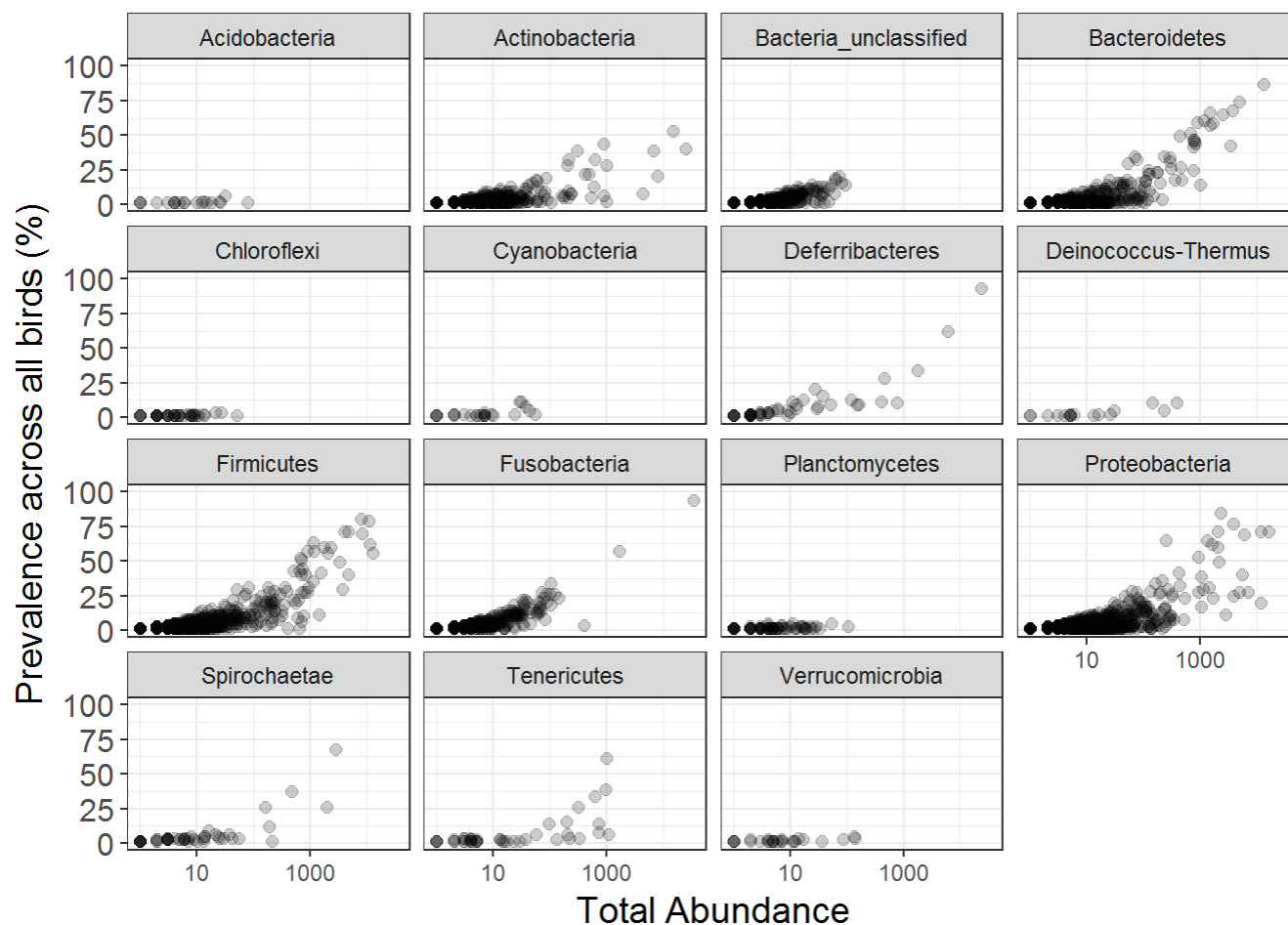
#using rarefied data, but can test with others - makes no difference

##### Prevalence #####
#####
##prevance whole group

prev0 = apply(X = otu_table(migration),
              MARGIN = ifelse(taxa_are_rows(migration), yes = 1, no = 2),
              FUN = function(x){sum(x > 0)})
prevdf = data.frame(Prevalence = prev0,
                    TotalAbundance = taxa_sums(migration),
                    tax_table(migration))
keepPhyla = table(prevdf$Phylum)[(table(prevdf$Phylum) > 10)]
prevdf1 = subset(prevdf, Phylum %in% names(keepPhyla))

prevdf1$Prevalence<-(prevdf1$Prevalence/77)*100
prevdf$Prevalence<-(prevdf$Prevalence/77)*100

ggplot(prevdf1, aes(TotalAbundance, Prevalence)) +
  geom_point(size = 2, alpha = 0.2) +
  scale_x_log10() +
  xlab("Total Abundance") + ylab("Prevalence across all birds (%)")+
  facet_wrap(~Phylum)+theme_bw() +
  theme( axis.text.y = element_text(size=12), axis.title=element_text(size=14))+ylim(c(0,100)
)
```



```
mean(prevdf$Prevalence)
```

```
## [1] 4.638008
```

```
#write.csv(prevdf, "prevalence_all_samples.csv")

#####

##table S2

#Subset each group seperately to get top 10 OTUs
# b = broome, f = flinders, a = adult/migrants, sy = second year/resident

curlew.b<-migration%>%subset_samples(species=="Curlew sandpiper"& site=="Broome")
curlew.b<- prune_taxa(taxa_sums(curlew.b) > 0, curlew.b)
curlew.b.a<-curlew.b%>%subset_samples(age=="Adult")
curlew.b.a<- prune_taxa(taxa_sums(curlew.b.a) > 0, curlew.b.a)
curlew.b.sy<-curlew.b%>%subset_samples(age=="Second year")
curlew.b.sy<- prune_taxa(taxa_sums(curlew.b.sy) > 0, curlew.b.sy)

stint.b<-migration%>%subset_samples(site=="Broome" & species=="Red-necked stint")
stint.b<- prune_taxa(taxa_sums(stint.b) > 0, stint.b)
stint.b.a<-stint.b%>%subset_samples(age=="Adult")
stint.b.a<- prune_taxa(taxa_sums(stint.b.a) > 0, stint.b.a)
stint.b.sy<-stint.b%>%subset samples(age=="Second year")
```

```
stint.b.sy<- prune_taxa(taxa_sums(stint.b.sy) > 0, stint.b.sy)

stint.f<-migration%>%subset_samples(month=="September")
stint.f<-stint.f%>%subset_samples(species=="Red-necked stint")
stint.f<- prune_taxa(taxa_sums(stint.f) > 0, stint.f)
stint.f.sy<-stint.f%>%subset_samples(age=="Second year")

stint.f.sy<- prune_taxa(taxa_sums(stint.f.sy) > 0, stint.f.sy)
stint.f.a<-stint.f%>%subset_samples(age=="Adult")
stint.f.a<- prune_taxa(taxa_sums(stint.f.a) > 0, stint.f.a)

##### TABLE S2A)

# broome curlew sand migrants

prev0 = apply(X = otu_table(curlew.b.a),
              MARGIN = ifelse(taxa_are_rows(curlew.b.a), yes = 1, no = 2),
              FUN = function(x){sum(x > 0)})
prevdf = data.frame(Prevalence = prev0,
                    TotalAbundance = taxa_sums(curlew.b.a),
                    tax_table(curlew.b.a))

prevdf$RelAbund<-prevdf$TotalAbundance/sum(prevdf$TotalAbundance)
prevdf$RelPrev<-prevdf$Prevalence/12
prevdf<-prevdf[order(-prevdf$RelAbund),]
head(prevdf)
```

##	Prevalence	TotalAbundance	Kingdom	Phylum
## Otu000023	9	7688	Bacteria	Actinobacteria
## Otu000027	8	5928	Bacteria	Proteobacteria
## Otu000002	11	4683	Bacteria	Fusobacteria
## Otu000058	4	4295	Bacteria	Actinobacteria
## Otu000010	10	3326	Bacteria	Deferribacteres
## Otu000028	7	2829	Bacteria	Actinobacteria
##	Class	Order	Family	
## Otu000023	Actinobacteria	Corynebacteriales	Corynebacteriaceae	
## Otu000027	Gammaproteobacteria	Aeromonadales	Succinivibrionaceae	
## Otu000002	Fusobacteriia	Fusobacteriales	Fusobacteriaceae	
## Otu000058	Actinobacteria	Corynebacteriales	Corynebacteriaceae	
## Otu000010	Deferribacteres	Deferribacterales	Deferribacteraceae	
## Otu000028	Actinobacteria	Corynebacteriales	Corynebacteriaceae	
##	Genus	RelAbund	RelPrev	
## Otu000023	Corynebacteriaceae_unclassified	0.11017484	0.7500000	
## Otu000027	Anaerobiospirillum	0.08495271	0.6666667	
## Otu000002	Cetobacterium	0.06711092	0.9166667	
## Otu000058	Corynebacteriaceae_unclassified	0.06155059	0.3333333	
## Otu000010	Mucispirillum	0.04766409	0.8333333	
## Otu000028	<NA>	0.04054170	0.5833333	

```
#write.csv(prevdf, "curlew.b.a.csv")
```



```
#####

# broome curlew sand residents

prev0 = apply(X = otu_table(curlew.b.sy),
              MARGIN = ifelse(taxa_are_rows(curlew.b.sy), yes = 1, no = 2),
              FUN = function(x){sum(x > 0)})
prevdf = data.frame(Prevalence = prev0,
                    TotalAbundance = taxa_sums(curlew.b.sy),
                    tax_table(curlew.b.sy))

prevdf$RelAbund<-prevdf$TotalAbundance/sum(prevdf$TotalAbundance)
prevdf$RelPrev<-prevdf$Prevalence/6
prevdf<-prevdf[order(-prevdf$RelAbund),]
head(prevdf)
```

##	Prevalence	TotalAbundance	Kingdom	Phylum
## Otu000002	5	4223	Bacteria	Fusobacteria
## Otu000010	5	3457	Bacteria	Deferribacteres
## Otu000018	6	2711	Bacteria	Firmicutes
## Otu000016	6	2427	Bacteria	Bacteroidetes
## Otu000034	6	2238	Bacteria	Firmicutes
## Otu000008	5	1973	Bacteria	Firmicutes
##	Class	Order	Family	
## Otu000002	Fusobacteriia	Fusobacteriales	Fusobacteriaceae	
## Otu000010	Deferribacteres	Deferribacterales	Deferribacteraceae	
## Otu000018	Clostridia	Clostridiales	Lachnospiraceae	
## Otu000016	Bacteroidia	Bacteroidales	Bacteroidaceae	
## Otu000034	Erysipelotrichia	Erysipelotrichales	Erysipelotrichaceae	
## Otu000008	Bacilli	Lactobacillales	Enterococcaceae	
##	Genus	RelAbund	RelPrev	
## Otu000002	Cetobacterium	0.12103755	0.8333333	
## Otu000010	Mucispirillum	0.09908283	0.8333333	
## Otu000018	Tyzzerella_3	0.07770135	1.0000000	
## Otu000016	Bacteroides	0.06956148	1.0000000	
## Otu000034	Erysipelotrichaceae_unclassified	0.06414445	1.0000000	
## Otu000008	Catelllicoccus	0.05654915	0.8333333	

```
#write.csv(prevdf, "curlew.b.sy.csv")

#####

##table S2b)

# broome stint migrants

prev0 = apply(X = otu_table(stint.b.a),
              MARGIN = ifelse(taxa_are_rows(stint.b.a), yes = 1, no = 2),
              FUN = function(x){sum(x > 0)})
prevdf = data.frame(Prevalence = prev0,
                    TotalAbundance = taxa_sums(stint.b.a),
                    tax table(stint.b.a))
```

```
prevdf$RelAbund<-prevdf$TotalAbundance/sum(prevdf$TotalAbundance)
prevdf$RelPrev<-prevdf$Prevalence/13
prevdf<-prevdf[order(-prevdf$RelAbund),]
head(prevdf)
```

##	Prevalence	TotalAbundance	Kingdom	Phylum
## Otu000013	11	8616	Bacteria	Actinobacteria
## Otu000002	13	7614	Bacteria	Fusobacteria
## Otu000017	12	5778	Bacteria	Proteobacteria
## Otu000009	6	5195	Bacteria	Actinobacteria
## Otu000010	13	3510	Bacteria	Deferribacteres
## Otu000016	12	3106	Bacteria	Bacteroidetes
##	Class	Order		
## Otu000013	Actinobacteria	Corynebacteriales		
## Otu000002	Fusobacteriia	Fusobacteriales		
## Otu000017	Gammaproteobacteria	Aeromonadales		
## Otu000009	Actinobacteria	Corynebacteriales		
## Otu000010	Deferribacteres	Deferribacterales		
## Otu000016	Bacteroidia	Bacteroidales		
##		Family		Genus
## Otu000013		Corynebacteriaceae	Corynebacteriaceae_unclassified	
## Otu000002		Fusobacteriaceae		Cetobacterium
## Otu000017		Succinivibrionaceae	Succinivibrionaceae_unclassified	
## Otu000009	Corynebacteriales_unclassified		Corynebacteriales_unclassified	
## Otu000010		Deferribacteraceae		Mucispirillum
## Otu000016		Bacteroidaceae		Bacteroides
##	RelAbund	RelPrev		
## Otu000013	0.11397579	0.8461538		
## Otu000002	0.10072095	1.0000000		
## Otu000017	0.07643363	0.9230769		
## Otu000009	0.06872148	0.4615385		
## Otu000010	0.04643164	1.0000000		
## Otu000016	0.04108737	0.9230769		

```
#write.csv(prevdf, "stint.b.a.csv")
#####

## broome stint resident

prev0 = apply(X = otu_table(stint.b.sy),
              MARGIN = ifelse(taxa_are_rows(stint.b.sy), yes = 1, no = 2),
              FUN = function(x){sum(x > 0)})
prevdf = data.frame(Prevalence = prev0,
                    TotalAbundance = taxa_sums(stint.b.sy),
                    tax_table(stint.b.sy))

prevdf$RelAbund<-prevdf$TotalAbundance/sum(prevdf$TotalAbundance)
prevdf$RelPrev<-prevdf$Prevalence/16
prevdf<-prevdf[order(-prevdf$RelAbund),]
```

```
head(prevdf)
```

##	Prevalence	TotalAbundance	Kingdom	Phylum
## Otu000010	16	9725	Bacteria	Deferribacteres
## Otu000011	15	7912	Bacteria	Proteobacteria
## Otu000020	10	5754	Bacteria	Firmicutes
## Otu000002	15	4614	Bacteria	Fusobacteria
## Otu000018	9	4421	Bacteria	Firmicutes
## Otu000036	15	4284	Bacteria	Proteobacteria
##	Class	Order	Family	
## Otu000010	Deferribacteres	Deferribacterales	Deferribacteraceae	
## Otu000011	Epsilonproteobacteria	Campylobacterales	Helicobacteraceae	
## Otu000020	Bacilli	Lactobacillales	Enterococcaceae	
## Otu000002	Fusobacteriia	Fusobacteriales	Fusobacteriaceae	
## Otu000018	Clostridia	Clostridiales	Lachnospiraceae	
## Otu000036	Alphaproteobacteria	Rhizobiales	Methylobacteriaceae	
##	Genus	RelAbund	RelPrev	
## Otu000010	Mucispirillum	0.10452494	1.0000	
## Otu000011	Helicobacter	0.08503869	0.9375	
## Otu000020	Catellibacillus	0.06184437	0.6250	
## Otu000002	Cetobacterium	0.04959157	0.9375	
## Otu000018	Tyzzerella_3	0.04751720	0.5625	
## Otu000036	Methylobacterium	0.04604471	0.9375	

```
#write.csv(prevdf, "stint.b.sy.csv")

#####

#Table S2c)

## vic stint migrants

prev0 = apply(X = otu_table(stint.f.a),
              MARGIN = ifelse(taxa_are_rows(stint.f.a), yes = 1, no = 2),
              FUN = function(x){sum(x > 0)})
prevdf = data.frame(Prevalence = prev0,
                    TotalAbundance = taxa_sums(stint.f.a),
                    tax_table(stint.f.a))

prevdf$RelAbund<-prevdf$TotalAbundance/sum(prevdf$TotalAbundance)
prevdf$RelPrev<-prevdf$Prevalence/15
prevdf<-prevdf[order(-prevdf$RelAbund),]

head(prevdf)
```

##	Prevalence	TotalAbundance	Kingdom	Phylum
## Otu000009	14	19124	Bacteria	Actinobacteria
## Otu000002	14	6743	Bacteria	Fusobacteria
## Otu000011	13	5365	Bacteria	Proteobacteria
## Otu000020	10	4261	Bacteria	Firmicutes
## Otu000013	14	3865	Bacteria	Actinobacteria
## Otu000014	15	2560	Bacteria	Firmicutes

##	Class	Order
## Otu000009	Actinobacteria	Corynebacteriales
## Otu000002	Fusobacteriia	Fusobacteriales
## Otu000011	Epsilonproteobacteria	Campylobacterales
## Otu000020	Bacilli	Lactobacillales
## Otu000013	Actinobacteria	Corynebacteriales
## Otu000014	Bacilli	Lactobacillales
##	Family	Genus
## Otu000009	Corynebacteriales_unclassified	Corynebacteriales_unclassified
## Otu000002	Fusobacteriaceae	Cetobacterium
## Otu000011	Helicobacteraceae	Helicobacter
## Otu000020	Enterococcaceae	Catelllicoccus
## Otu000013	Corynebacteriaceae	Corynebacteriaceae_unclassified
## Otu000014	Enterococcaceae	Catelllicoccus
##	RelAbund	RelPrev
## Otu000009	0.21924907	0.9333333
## Otu000002	0.07730582	0.9333333
## Otu000011	0.06150760	0.8666667
## Otu000020	0.04885067	0.6666667
## Otu000013	0.04431069	0.9333333
## Otu000014	0.02934938	1.0000000

```
#write.csv(prevdf, "stint.f.a.csv")

#####

##vic stint resident

prev0 = apply(X = otu_table(stint.f.sy),
              MARGIN = ifelse(taxa_are_rows(stint.f.sy), yes = 1, no = 2),
              FUN = function(x){sum(x > 0)})
prevdf = data.frame(Prevalence = prev0,
                    TotalAbundance = taxa_sums(stint.f.sy),
                    tax_table(stint.f.sy))

prevdf$RelAbund<-prevdf$TotalAbundance/sum(prevdf$TotalAbundance)
prevdf$RelPrev<-prevdf$Prevalence/15
prevdf<-prevdf[order(-prevdf$RelAbund),]
head(prevdf)
```

##	Prevalence	TotalAbundance	Kingdom	Phylum
## Otu000006	7	11596	Bacteria	Proteobacteria
## Otu000002	14	6271	Bacteria	Fusobacteria
## Otu000008	15	4866	Bacteria	Firmicutes
## Otu000031	8	4102	Bacteria	Firmicutes
## Otu000039	3	3774	Bacteria	Proteobacteria
## Otu000014	13	2963	Bacteria	Firmicutes
##	Class	Order	Family	
## Otu000006	Epsilonproteobacteria	Campylobacteriales	Helicobacteraceae	
## Otu000002	Fusobacteriia	Fusobacteriales	Fusobacteriaceae	
## Otu000008	Bacilli	Lactobacillales	Enterococcaceae	
## Otu000031	Erysipelotrichia	Erysipelotrichales	Erysipelotrichaceae	
## Otu000039	Gammaproteobacteria	Enterobacteriales	Enterobacteriaceae	

##	Otu000014	Bacilli	Lactobacillales	Enterococcaceae
##			Genus	RelAbund
##	Otu000006	Helicobacteraceae_unclassified	0.13294354	0.4666667
##	Otu000002	Cetobacterium	0.07189453	0.9333333
##	Otu000008	Catelllicoccus	0.05578676	1.0000000
##	Otu000031	Erysipelotrichaceae_unclassified	0.04702780	0.5333333
##	Otu000039	Escherichia-Shigella	0.04326741	0.2000000
##	Otu000014	Catelllicoccus	0.03396962	0.8666667

```
#write.csv(prevdf, "stint.f.sy.csv")

#####stacked barplots #####

#Figure 1)

##### migration

migration_phylum <- migration %>%
  tax_glom(taxrank = "Phylum") %>% # agglomerate at phylum level
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel. abundance
  psmelt() %>% # Melt to long format
  arrange(Phylum) # Sort data frame alphabetically by phylum

#filter(Abundance > 0.05) %>% # Filter out low abundance taxa
head(migration_phylum)
```

##	OTU	Sample	Abundance	Group	species	site	month
##	1	Otu000944	8680	0.016852966	8680 Red-necked stint	Flinders	September
##	2	Otu000944	8607	0.004815133	8607 Red-necked stint	Broome	August
##	3	Otu000944	8667	0.004815133	8667 Red-necked stint	Flinders	September
##	4	Otu000944	8639	0.003955288	8639 Red-necked stint	Flinders	September
##	5	Otu000944	8560	0.003439381	8560 Red-necked stint	Broome	August
##	6	Otu000944	8566	0.002063629	8566 Red-necked stint	Broome	August
##		age	group	bird_id	type	Recap	recap_delete
##	1	Adult	three	3685834	bird	No	No
##	2	Adult	two	3678451	bird	No	No
##	3	Second year	three	3686272	bird	No	No
##	4	Adult	three	3686361	bird	Yes	Yes
##	5	Adult	two	3674218	bird	No	No
##	6	Second year	two	3674234	bird	No	No
##		remove_replicate	weight	wing	hb	SourceSink	Env
##	1	No	29	NA	NA	source	gut
##	2	No	NA	NA	NA		
##	3	No	28	NA	NA	source	gut
##	4	No	31	NA	NA	source	gut
##	5	No	NA	NA	NA		
##	6	No	NA	NA	NA		
##		migration	Kingdom		Phylum		
##	1	Flinders-stint-adult	Bacteria	Acidobacteria			

```
## 2   Broome-stint-adult Bacteria Acidobacteria
## 3     Flinders-stint-SY Bacteria Acidobacteria
## 4 Flinders-stint-adult Bacteria Acidobacteria
## 5   Broome-stint-adult Bacteria Acidobacteria
## 6     Broome-stint-SY Bacteria Acidobacteria
```

```
#####

migration_phylum$Phylum<-factor(migration_phylum$Phylum)
#broome_phylum$Order<-factor(broome_phylum$Order)
unique(migration_phylum$Phylum )
```

```
## [1] Acidobacteria          Actinobacteria
## [3] Armatimonadetes          Bacteria_unclassified
## [5] Bacteroidetes            Chloroflexi
## [7] Cyanobacteria            Deferribacteres
## [9] Deinococcus-Thermus      FBP
## [11] Fibrobacteres            Firmicutes
## [13] Fusobacteria             Gemmatimonadetes
## [15] Gracilibacteria          Lentisphaerae
## [17] Microgenomates           Parcubacteria
## [19] Peregrinibacteria        Planctomycetes
## [21] Proteobacteria           Saccharibacteria
## [23] Spirochaetae             SR1_(Absconditabacteria)
## [25] Tenericutes              TM6_(Dependentiae)
## [27] Verrucomicrobia
## 27 Levels: Acidobacteria Actinobacteria ... Verrucomicrobia
```

```
migration_phylum$Phylum<-factor(migration_phylum$Phylum)

unique(migration_phylum$Phylum)
```

```
## [1] Acidobacteria          Actinobacteria
## [3] Armatimonadetes          Bacteria_unclassified
## [5] Bacteroidetes            Chloroflexi
## [7] Cyanobacteria            Deferribacteres
## [9] Deinococcus-Thermus      FBP
## [11] Fibrobacteres            Firmicutes
## [13] Fusobacteria             Gemmatimonadetes
## [15] Gracilibacteria          Lentisphaerae
## [17] Microgenomates           Parcubacteria
## [19] Peregrinibacteria        Planctomycetes
## [21] Proteobacteria           Saccharibacteria
## [23] Spirochaetae             SR1_(Absconditabacteria)
## [25] Tenericutes              TM6_(Dependentiae)
## [27] Verrucomicrobia
## 27 Levels: Acidobacteria Actinobacteria ... Verrucomicrobia
```

```
migration_phylum$Phylum<-factor(migration_phylum$Phylum, level = c("Actinobacteria",
                                                                              "Firmicutes",
```

```
"Proteobacteria",
"Bacteroidetes",
"Deferribacteres",
"Fusobacteria",
"Tenericutes",
"Spirochaetae" ,
"Deinococcus-Thermus",
"Acidobacteria",
"Armatimonadetes",
"Bacteria_unclassified",
"Chloroflexi",
"Cyanobacteria",
"FBP",
"Fibrobacteres",
"Gemmatimonadetes",
"Gracilibacteria",
"Ignavibacteriae",
"Lentisphaerae",
"Microgenomates",
"Parcubacteria",
"Peregrinibacteria",
"Planctomycetes",
"Saccharibacteria",

"SR1_(Absconditabacteria)",
"TM6_(Dependentiae)",
"Verrucomicrobia"))
```

```
unique(migration_phylum$Sample)
```

```
## [1] "8680" "8607" "8667" "8639" "8560" "8566" "8645" "8662" "8558" "8559"
## [11] "8610" "8632" "8628" "8563" "8581" "8671" "8549" "8552" "8548" "8660"
## [21] "8601" "8604" "8546" "8605" "8612" "8658" "8556" "8570" "8551" "8659"
## [31] "8657" "8646" "8665" "8629" "8655" "8627" "8633" "8609" "8616" "8649"
## [41] "8589" "8606" "8586" "8575" "8644" "8557" "8571" "8677" "8672" "8663"
## [51] "8630" "8689" "8565" "8569" "8615" "8564" "8590" "8561" "8682" "8583"
## [61] "8634" "8683" "8685" "8588" "8675" "8674" "8613" "8626" "8688" "8686"
## [71] "8578" "8625" "8573" "8618" "8684" "8679" "8638"
```

```
colors <- c("navyblue","green4","orchid","royalblue","black","red","brown","grey","bisque",
           "white","white","white","white","white","white","white","white","white","white","white",
           "white","white","white","white","white","white","white","white","white")

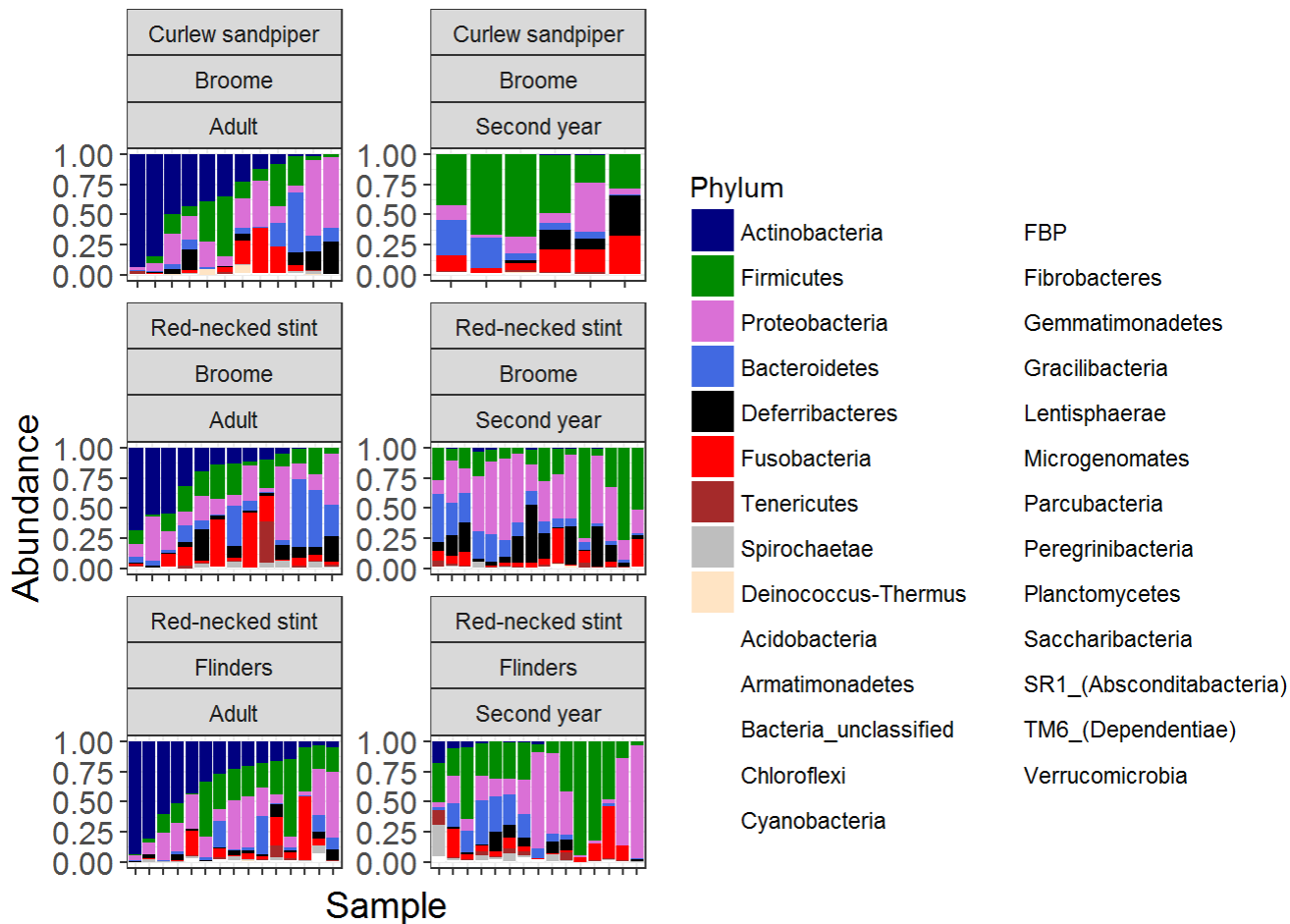
names(colors) <- levels(migration_phylum$Phylum)
colScale <- scale_fill_manual(name = "Phylum",values = colors)
```

```
migration_phylum$Sample<-factor(migration_phylum$Sample, level = c("8682", "8629", "8615", "8685", "8610", "8662", "8560", "8607", "8671", "8589", "8657", "8573", "8625", "8627", "8639", "8549", "8645", "8632", "8660", "8649", "8601", "8658", "8638", "8646", "8634", "8606", "8612", "8581", "8618", "8609", "8628", "8672", "8605", "8689", "8659", "8616", "8626", "8604", "8569", "8677", "8588", "8675", "8551", "8613", "8546", "8683", "8590", "8557", "8558", "8674", "8559", "8680", "8630", "8563", "8586", "8556", "8561", "8644", "8667", "8566", "8665", "8633", "8564", "8583", "8575", "8571", "8565", "8578", "8570", "8684", "8688", "8655", "8552", "8548", "8679", "8663", "8686"))

##Figure 1a

ggplot(migration_phylum, aes(x = Sample, y = Abundance, fill = Phylum)) +

  geom_bar(stat = "identity") +
  theme(axis.text.x = element_blank(), axis.text.y = element_text(size=12), axis.title=element_text(size=14))+
  facet_wrap(species~site~age, scales="free", nrow=3, ncol=2)+colScale
```



```
#####repeat but for Family Fig 1b #####

migration_family <- migration %>%
  tax_glom(taxrank = "Family") %>% # agglomerate at phylum level
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel. abundance
```



```
psmelt() %>% # Melt to long format
  # Filter out low abundance taxa
  arrange(Family) # Sort data frame alphabetically by family
#
head(migration_family)
```

##	OTU	Sample	Abundance	Group	species	site	month	
## 1	Otu000437	8558	0.003439381	8558	Red-necked stint	Broome	August	
## 2	Otu000437	8563	0.002407567	8563	Red-necked stint	Broome	August	
## 3	Otu000437	8552	0.001031814	8552	Red-necked stint	Broome	August	
## 4	Otu000437	8564	0.001031814	8564	Red-necked stint	Broome	August	
## 5	Otu000437	8556	0.000000000	8556	Red-necked stint	Broome	August	
## 6	Otu000437	8583	0.000000000	8583	Curlew sandpiper	Broome	August	
##	age	group	bird_id	type	Recap	recap_delete	Recap_num	replicate
## 1	Second year	two	3674216	bird	No	No	NA	No
## 2	Second year	two	3674221	bird	No	No	NA	No
## 3	Second year	two	3674232	bird	No	No	NA	No
## 4	Second year	two	3674222	bird	No	No	NA	No
## 5	Second year	two	3674214	bird	No	No	NA	No
## 6	Second year	one	4267072	bird	No	No	NA	No
##	remove_replicate	weight	wing	hb	SourceSink	Env	Date	Plate
## 1		No	NA	NA	NA		22/08/2015	2
## 2		No	NA	NA	NA		22/08/2015	2
## 3		No	NA	NA	NA		22/08/2015	2
## 4		No	NA	NA	NA		22/08/2015	2
## 5		No	NA	NA	NA		22/08/2015	2
## 6		No	NA	NA	NA		22/08/2015	2
##	migration	Kingdom	Phylum	Class				
## 1	Broome-stint-SY	Bacteria	Proteobacteria	Deltaproteobacteria				
## 2	Broome-stint-SY	Bacteria	Proteobacteria	Deltaproteobacteria				
## 3	Broome-stint-SY	Bacteria	Proteobacteria	Deltaproteobacteria				
## 4	Broome-stint-SY	Bacteria	Proteobacteria	Deltaproteobacteria				
## 5	Broome-stint-SY	Bacteria	Proteobacteria	Deltaproteobacteria				
## 6	Broome-curlewsand-SY	Bacteria	Proteobacteria	Deltaproteobacteria				
##	Order	Family						
## 1	Oligoflexales	0319-6G20						
## 2	Oligoflexales	0319-6G20						
## 3	Oligoflexales	0319-6G20						
## 4	Oligoflexales	0319-6G20						
## 5	Oligoflexales	0319-6G20						
## 6	Oligoflexales	0319-6G20						

```
migration_family$Family<-factor(migration_family$Family)
#broome_phylum$Order<-factor(broome_phylum$Order)
unique(migration_family$Family)
```

##	[1]	0319-6G20	Acetobacteraceae
##	[3]	Acidaminococcaceae	Acidimicrobiaceae
##	[5]	Acidimicrobiales_unclassified	Acidobacteriaceae_(Subgroup_1)
##	[7]	Actinobacteria_unclassified	Actinomycetaceae
##	[9]	Aeromonadaceae	Aeromonadales_unclassified

## [11]	Alcaligenaceae	Alphaproteobacteria_unclassified
## [13]	Alteromonadaceae	Anaerolineaceae
## [15]	Anaeroplasmataceae	Anaplasmataceae
## [17]	Archangiaceae	Ardenticatenales_fa
## [19]	Arenicellaceae	Armatimonadales_unclassified
## [21]	Aurantimonadaceae	Bacillaceae
## [23]	Bacilli_unclassified	Bacteria_unclassified
## [25]	Bacteriovoracaceae	Bacteroidaceae
## [27]	Bacteroidales_S24-7_group	Bacteroidales_unclassified
## [29]	Bacteroidetes_unclassified	BD2-11_terrestrial_group_fa
## [31]	BD7-8_marine_group_fa	Bdellovibrionaceae
## [33]	Beijerinckiaceae	Bifidobacteriaceae
## [35]	Blastocatellaceae_(Subgroup_4)	Brachyspiraceae
## [37]	Bradymonadaceae	Bradyrhizobiaceae
## [39]	Brevibacteriaceae	Brucellaceae
## [41]	Burkholderiaceae	Burkholderiales_unclassified
## [43]	Caldilineaceae	Campylobacteraceae
## [45]	Campylobacterales_unclassified	Cardiobacteriaceae
## [47]	Carnobacteriaceae	Caulobacteraceae
## [49]	Cellulomonadaceae	Cellvibrionaceae
## [51]	Cellvibrionales_unclassified	Chitinophagaceae
## [53]	Chloroflexaceae	Chloroflexi_unclassified
## [55]	Chromatiaceae	Clostridia_unclassified
## [57]	Clostridiaceae_1	Clostridiaceae_4
## [59]	Clostridiales_unclassified	Clostridiales_vadinBB60_group
## [61]	Comamonadaceae	Coriobacteriaceae
## [63]	Corynebacteriaceae	Corynebacteriales_unclassified
## [65]	Coxiellaceae	Cryomorphaceae
## [67]	CS-B046_fa	Cyanobacteria_unclassified
## [69]	Cyclobacteriaceae	Cytophagaceae
## [71]	Cytophagales_unclassified	dl142_fa
## [73]	DA111	DBS1
## [75]	Deferribacteraceae	Deinococcaceae
## [77]	Deltaproteobacteria_unclassified	Dermabacteraceae
## [79]	Dermacoccaceae	Dermatophilaceae
## [81]	Desulfo bacteraceae	Desulfo bulbaceae
## [83]	Desulfovibrionaceae	Desulfovibrionales_unclassified
## [85]	Desulfuromonadaceae	DEV007
## [87]	Dietziaceae	Ectothiorhodospiraceae
## [89]	Eel-36e1D6	Elev-16S-1332
## [91]	Enterobacteriaceae	Enterococcaceae
## [93]	Entomoplasmatales_Incertae_Sedis	Epsilonproteobacteria_unclassified
## [95]	Erysipelotrichaceae	Erythrobacteraceae
## [97]	Eubacteriaceae	Euzebyaceae
## [99]	EV818SWSAP88	Family_XI
## [101]	Family_XII	Family_XIII
## [103]	Family_I	Family_II
## [105]	FBP_fa	FD035
## [107]	Firmicutes_unclassified	Flammeovirgaceae
## [109]	Flavobacteriaceae	Flavobacteriales_unclassified
## [111]	Fusobacteriaceae	Fusobacteriales_unclassified
## [113]	Gammaproteobacteria_unclassified	Gastranaerophilales_fa
## [115]	Gemmatimonadaceae	Geodermatophilaceae
## [117]	Gitt-GS-136_fa	Gracilibacteria_fa

## [119] Granulosicoccaceae	Hahellaceae
## [121] Halanaerobiales_unclassified	Haliangiaceae
## [123] Halieaceae	Halomonadaceae
## [125] Helicobacteraceae	Herpetosiphonaceae
## [127] HOC36_fa	Holosporaceae
## [129] Hydrogenophilaceae	Hyphomicrobiaceae
## [131] Hyphomonadaceae	Intrasporangiaceae
## [133] JG30-KF-CM45_fa	JG34-KF-361
## [135] JTB255_marine_benthic_group	KD4-96_fa
## [137] KF-JG30-B3	Kineosporiaceae
## [139] Lachnospiraceae	Lactobacillaceae
## [141] Lactobacillales_unclassified	LD29
## [143] Legionellaceae	Lentisphaeraceae
## [145] Leptospiraceae	Leptotrichiaceae
## [147] Leuconostocaceae	Listeriaceae
## [149] Longimicrobiaceae	Methylobacteriaceae
## [151] Methylophilaceae	MgMjR-022
## [153] Microbacteriaceae	Micrococcaceae
## [155] Micrococcales_unclassified	Microgenomates_unclassified
## [157] Milano-WF1B-44_fa	Mollicutes_unclassified
## [159] Moraxellaceae	MSB-1E8
## [161] Mycobacteriaceae	Mycoplasmataceae
## [163] NBl-n_fa	Neisseriaceae
## [165] Nitriliruptoraceae	Nitrosomonadaceae
## [167] Nocardiaceae	Nocardiodaceae
## [169] NS9_marine_group	Obscuribacterales_fa
## [171] Oceanospirillaceae	Oceanospirillales_unclassified
## [173] Oligoflexaceae	OM1_clade
## [175] OM182_clade	OM190_fa
## [177] Oxalobacteraceae	Paenibacillaceae
## [179] Parcubacteria_fa	Parvularculaceae
## [181] Pasteurellaceae	PeM15_fa
## [183] Peptostreptococcaceae	Peregrinibacteria_fa
## [185] Phycisphaeraceae	Phyllobacteriaceae
## [187] Pla3_lineage_fa	Planctomycetaceae
## [189] Planococcaceae	Porphyromonadaceae
## [191] Prevotellaceae	Propionibacteriaceae
## [193] Proteobacteria_unclassified	Pseudoalteromonadaceae
## [195] Pseudomonadaceae	Pseudonocardiaceae
## [197] Psychromonadaceae	Puniceicoccaceae
## [199] PYR10d3_fa	Rhizobiaceae
## [201] Rhizobiales_Incertae_Sedis	Rhizobiales_unclassified
## [203] Rhodobacteraceae	Rhodobiaceae
## [205] Rhodocyclaceae	Rhodospirillaceae
## [207] Rhodospirillales_Incertae_Sedis	Rhodospirillales_unclassified
## [209] Rhodothermaceae	Rickettsiaceae
## [211] Rickettsiales_Incertae_Sedis	Rickettsiales_unclassified
## [213] Rikenellaceae	Rs-D42
## [215] Rubritaleaceae	Ruminococcaceae
## [217] S0134_terrestrial_group_fa	S085_fa
## [219] Saccharibacteria_fa	Sandaracinaceae
## [221] Saprospiraceae	SAR116_clade
## [223] SAR324_clade(Marine_group_B)_fa	Shewanellaceae
## [225] Solibacteraceae_(Subgroup_3)	Solirubrobacteraceae

```
## [227] Sphingobacteriaceae      Sphingobacteriales_unclassified
## [229] Sphingomonadaceae         Sphingomonadales_unclassified
## [231] Spirochaetaceae          Spirochaetales_unclassified
## [233] Sphingobacteriaceae       SR1_(Absconditabacteria)_fa
## [235] Staphylococcaceae         Streptococcaceae
## [237] Streptomyetaceae         Subgroup_21_fa
## [239] Subgroup_22_fa           Subgroup_23_fa
## [241] Subgroup_6_fa            Subgroup_7_fa
## [243] Succinivibrionaceae      Surface_1
## [245] Sva0071_fa               Sva0725
## [247] Sva0996_marine_group     Sva1033
## [249] Syntrophaceae            T9d
## [251] Tepidispheeraceae        Thermoactinomycetaceae
## [253] Thiotrichaceae           TM6_(Dependentiae)_fa
## [255] TRA3-20_fa              Trueperaceae
## [257] uncultured               uncultured_fa
## [259] Unknown_Family          Veillonellaceae
## [261] Verrucomicrobia_fa       Verrucomicrobia_unclassified
## [263] Verrucomicrobiaceae     Verrucomicrobiales_unclassified
## [265] Vibrionaceae             X35_fa
## [267] Xanthobacteraceae        Xanthomonadaceae
## [269] Xanthomonadales_Incertae_Sedis Xanthomonadales_unclassified
## 270 Levels: 0319-6G20 Acetobacteraceae ... Xanthomonadales_unclassified
```

```
#write.csv(migration_family,"family.csv")
```

```
##all common Corynebacteriales_unclassified are genus Corynebacterium (we ran sequences through ARB):
```

```
migration_family$Family[migration_family$Family=="Corynebacteriales_unclassified"]<-"Corynebacteriaceae"
```

```
#####plot
```

```
migration_family$Family<-factor(migration_family$Family, level = c("Corynebacteriaceae",
                                                                    "Acidaminococcaceae", "Clostridiaceae_1", "Clostridiales_unclassified", "Enterococcaceae", "Erysipelotrichaceae", "Firmicutes_unclassified", "Lachnospiraceae", "Peptostreptococcaceae", "Ruminococcaceae", "Staphylococcaceae",
                                                                    "Comamonadaceae", "Desulfovibrionaceae", "Enterobacteriaceae", "Gammaproteobacteria_unclassified", "Helicobacteraceae", "Methylobacteriaceae", "Oxalobacteraceae", "Rhizobiales_unclassified", "Rhodobacteraceae", "Sphingomonadaceae", "Succinivibrionaceae",
                                                                    "Bacteroidaceae", "Bacteroidales_unclassified", "Chitinophagaceae", "Flavobacteriaceae", "Porphyromonadaceae", "Rikenellaceae",
                                                                    "Deferribacteraceae", "Fusobacteriaceae", "Anaeroplasmataceae", "Mycoplasmataceae",
                                                                    "Brachyspiraceae", "Deinococcaceae",
```

	"0319-6G20",
	"Acetobacteraceae",
	"Acidimicrobiaceae",
	"Acidimicrobiales_unclassif
ied",	
	"Acidobacteriaceae_(Subgrou
p_1)",	
	"Actinobacteria_unclassifie
d",	
	"Actinomycetaceae",
	"Aeromonadaceae",
	"Aeromonadales_unclassified
",	
	"Alcaligenaceae",
	"Alphaproteobacteria_unclas
sified",	
	"Alteromonadaceae",
	"Anaerolineaceae",
	"Anaplasmataceae",
	"Ardenticatenales_fa",
	"Arenicellaceae",
	"Armatimonadales_unclassifi
ed",	
	"Aurantimonadaceae",
	"Bacillaceae",
	"Bacilli_unclassified",
	"Bacteria_unclassified",
	"Bacteriovoracaceae",
	"Bacteroidales_S24-7_group"
,	
	"Bacteroidetes_unclassified
",	
	"Bartonellaceae",
a",	"BD2-11_terrestrial_group_f
	"BD7-8_marine_group_fa",
	"Bdellovibrionaceae",
	"Beijerinckiaceae",
	"Bifidobacteriaceae",
	"Blastocatellaceae_(Subgrou
p_4)",	
	"Bradymonadaceae",
	"Bradymonadales_fa",
	"Bradyrhizobiaceae",
	"Brevibacteriaceae",
	"Brucellaceae",
	"Burkholderiaceae",
	"Burkholderiales_unclassifi
ed",	

fied",	"Caldilineaceae",
	"Campylobacteraceae",
	"Campylobacterales_unclassi
	"Cardiobacteriaceae",
	"Carnobacteriaceae",
	"Caulobacteraceae",
	"Cellulomonadaceae",
	"Cellvibrionaceae",
ed",	"Cellvibrionales_unclassifi
	"Chloroflexaceae",
	"Chloroflexi_unclassified",
	"Chromatiaceae",
	"Clostridia_unclassified",
	"Clostridiaceae_4",
oup",	"Clostridiales_vadinBB60_gr
	"Coriobacteriaceae",
	"Coxiellaceae",
	"Cryomorphaceae",
	"CS-B046_fa",
",	"Cyanobacteria_unclassified
	"Cyclobacteriaceae",
	"Cytophagaceae",
,	"Cytophagales_unclassified"
	"d142_fa",
	"DA111",
	"DBS1",
sified",	"Deltaproteobacteria_unclas
	"Dermabacteraceae",
	"Dermacoccaceae",
	"Dermatophilaceae",
	"Desulfobacteraceae",
	"Desulfobulbaceae",
ified",	"Desulfovibrionales_unclass
	"Desulfuromonadaceae",
ified",	"Desulfuromonadales_unclass
	"DEV007",

	"Dietziaceae",
	"Eel-36e1D6",
	"Elev-16S-1332",
	"Entomoplasmales_Incertae
_Sedis",	
	"Epsilonproteobacteria_uncl
assified",	
	"Erythrobacteraceae",
	"Eubacteriaceae",
	"Euzebyaceae",
	"Family_XI",
	"Family_XII",
	"Family_XIII",
	"FamilyI",
	"FamilyII",
	"FBP_fa",
	"FD035",
	"Flammeovirgaceae",
	"Flavobacteriales_unclassif
ied",	
	"Fusobacteriales_unclassifi
ed",	
	"Gastranaerophilales_fa",
	"Gemmatimonadaceae",
	"Geodermatophilaceae",
	"Gitt-GS-136_fa",
	"Gracilibacteria_fa",
	"Granulosicoccaceae",
	"Hahellaceae",
	"Halanaerobiales_unclassifi
ed",	
	"Haliangiaceae",
	"Halieaceae",
	"Halomonadaceae",
	"Herpetosiphonaceae",
	"HOC36_fa",
	"Holosporaceae",
	"Hydrogenophilaceae",
	"Hyphomicrobiaceae",
	"Hyphomonadaceae",
	"Iamiaceae",
	"Intrasporangiaceae",
	"JG30-KF-CM45_fa",
	"JG34-KF-361",
	"JTB255_marine_benthic_grou

```
p" ,
"KD4-96_fa" ,
"KF-JG30-B3" ,
"Kineosporiaceae" ,

"Lactobacillaceae" ,
"Lactobacillales_unclassifi

ed" ,

"LD29" ,
"Legionellaceae" ,
"Lentisphaeraceae" ,
"Leptospiraceae" ,
"Leptotrichiaceae" ,
"Leuconostocaceae" ,
"Listeriaceae" ,
"Longimicrobiaceae" ,

"Methylophilaceae" ,
"MgMjR-022" ,
"Microbacteriaceae" ,
"Micrococcaceae" ,
"Micrococcales_unclassified

",
"Microgenomates_unclassifie

d" ,

"Milano-WF1B-44_fa" ,
"Mollicutes_unclassified" ,

"Moraxellaceae" ,
"MSB-1E8" ,
"Mycobacteriaceae" ,

"NB1-n_fa" ,
"Neisseriaceae" ,
"Nitriliruptoraceae" ,
"Nitrosomonadaceae" ,
"Nocardiaceae" ,
"Nocardioidaceae" ,
"NS9_marine_group" ,
"Obscuribacterales_fa" ,
"Oceanospirillaceae" ,
"Oceanospirillales_unclassi

fied" ,

"Oligoflexaceae" ,
"OM1_clade" ,
"OM182_clade" ,
"OM190_fa" ,

"Paenibacillaceae" ,
"Parcubacteria_fa" ,
"Parvularculaceae" ,
"Pasteurellaceae" ,
"PeM15_fa" ,
```



	"Peregrinibacteria_fa",
	"Phycisphaeraceae",
	"Phyllobacteriaceae",
	"Pla3_lineage_fa",
	"Planctomycetaceae",
	"Planococcaceae",
	"Prevotellaceae",
	"Propionibacteriaceae",
	"Proteobacteria_unclassifie
d",	
	"Pseudoalteromonadaceae",
	"Pseudomonadaceae",
	"Pseudonocardiaceae",
	"Psychromonadaceae",
	"Puniceicoccaceae",
	"PYR10d3_fa",
	"Rhizobiaceae",
	"Rhizobiales_Incertae_Sedis
",	
	"Rhodobiaceae",
	"Rhodocyclaceae",
	"Rhodospirillaceae",
	"Rhodospirillales_Incertae_
Sedis",	
	"Rhodospirillales_unclassif
ied",	
	"Rhodothermaceae",
	"Rickettsiaceae",
	"Rickettsiales_Incertae_Sed
is",	
	"Rickettsiales_unclassified
",	
	"Rs-D42",
	"Rubritaleaceae",
	"S0134_terrestrial_group_fa
",	
	"S085_fa",
	"Saccharibacteria_fa",
	"Sandaracinaceae",
	"Saprospiraceae",
	"SAR116_clade",
	"SAR324_clade(Marine_group_
B)_fa",	
	"Shewanellaceae",
	"Solibacteraceae_(Subgroup_
3)",	
	"Solirubrobacteraceae",
	"Sphingobacteriaceae",

ified",	"Sphingobacteriales_unclass
ied",	"Sphingomonadales_unclassif
d",	"Spirochaetaceae", "Spirochaetales_unclassifie
a",	"Spongiibacteraceae", "SR-FBR-L83", "SR1_(Absconditabacteria)_f
	"Streptococcaceae", "Streptomycetaceae", "Subgroup_21_fa", "Subgroup_22_fa", "Subgroup_23_fa", "Subgroup_6_fa", "Subgroup_7_fa",
	"Surface_1", "Sva0071_fa", "Sva0725", "Sva0996_marine_group", "Sva1033", "Syntrophaceae", "T9d", "Tepidisphaeraceae", "Thermoactinomycetaceae",
	"Thiotrichaceae", "TM146", "TM6_(Dependentiae)_fa", "TRA3-20_fa", "Trueperaceae", "uncultured", "uncultured_fa", "Unknown_Family", "Veillonellaceae", "Verrucomicrobia_fa", "Verrucomicrobia_unclassifi
ed",	"Verrucomicrobiaceae", "Verrucomicrobiales_unclass
ified",	"Vibrionaceae", "X35_fa", "Xanthobacteraceae", "Xanthomonadaceae", "Xanthomonadales_Incertae_S
edis",	
ed" ) )	"Xanthomonadales_unclassifi

```
##only families that make up more than 5% total abundance are coloured. The rest are coloured white

colors <- c("navy",
            "olivedrab4","olivedrab2","forestgreen","darkgreen","yellow3","yellow1","yellowgre
en","lawngreen","springgreen3","aquamarine",
            "mistyrose","pink","palevioletred1","palevioletred4", "orchid1","orchid4", "maroon
1","magenta","deeppink3","red4","purple",

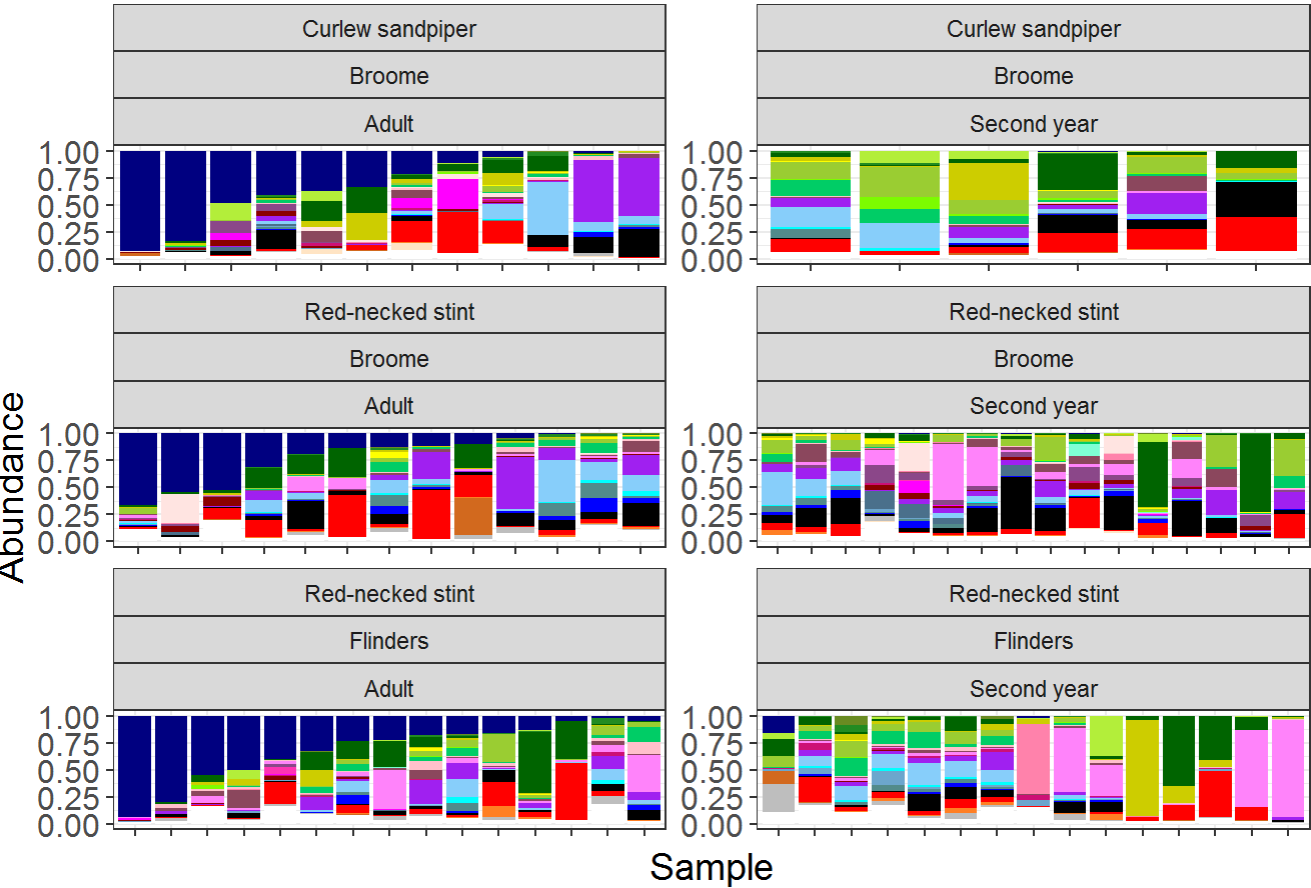
            "lightskyblue", "cyan","skyblue4", "skyblue3","darkslategray4","blue",
            "black",
            "red",
            "chocolate1","chocolate",
            "grey",
            "bisque",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white",
            "white","white","white","white","white","white","white","white","white","white")

names(colors) <- levels(migration_family$Family)
colScale <- scale_fill_manual(name = "Family",values = colors)

migration_family$Sample<-factor(migration_family$Sample, level = c("8682", "8629", "8615", "86
85", "8610", "8662", "8560", "8607", "8671", "8589", "8657", "8573", "8625", "8627", "8639", "
```

```
8549", "8645", "8632", "8660", "8649", "8601",  
"8658", "8638", "8646", "86  
34", "8606", "8612", "8581", "8618", "8609", "8628", "8672", "8605", "8689", "8659", "8616", "  
8626", "8604", "8569", "8677", "8588", "8675",  
"8551", "8613", "8546", "86  
83", "8590", "8557", "8558", "8674", "8559", "8680", "8630", "8563", "8586", "8556", "8561", "  
8644", "8667", "8566", "8665", "8633", "8564",  
"8583", "8575", "8571", "85  
65", "8578", "8570", "8684", "8688", "8655", "8552", "8548", "8679", "8663", "8686"))  
  
ggplot(migration_family, aes(x = Sample, y = Abundance, fill = Family)) +  
  
#Fig 1b)  
geom_bar(stat = "identity", position="fill") +  
theme(axis.text.x = element_blank(), axis.text.y = element_text(size=12), axis.title=element  
_text(size=14))+  
ggtitle("Family abundance per sample")+facet_wrap(species~site~age, scales="free", nrow=3, n  
col=2)+colScale + theme(legend.position="none")
```

Family abundance per sample



##### ORDINATION ANALYSIS#####  
#####

```
##Figure S4)
```

```
set.seed(1)
```

```
migration_unifrac <- ordinate(
  physeq = migration,
  method = "NMDS",
  distance = "unifrac"
)
```

```
## Warning in UniFrac(physeq, ...): Randomly assigning root as -- Otu000566 --
## in the phylogenetic tree in the data you provided.
```

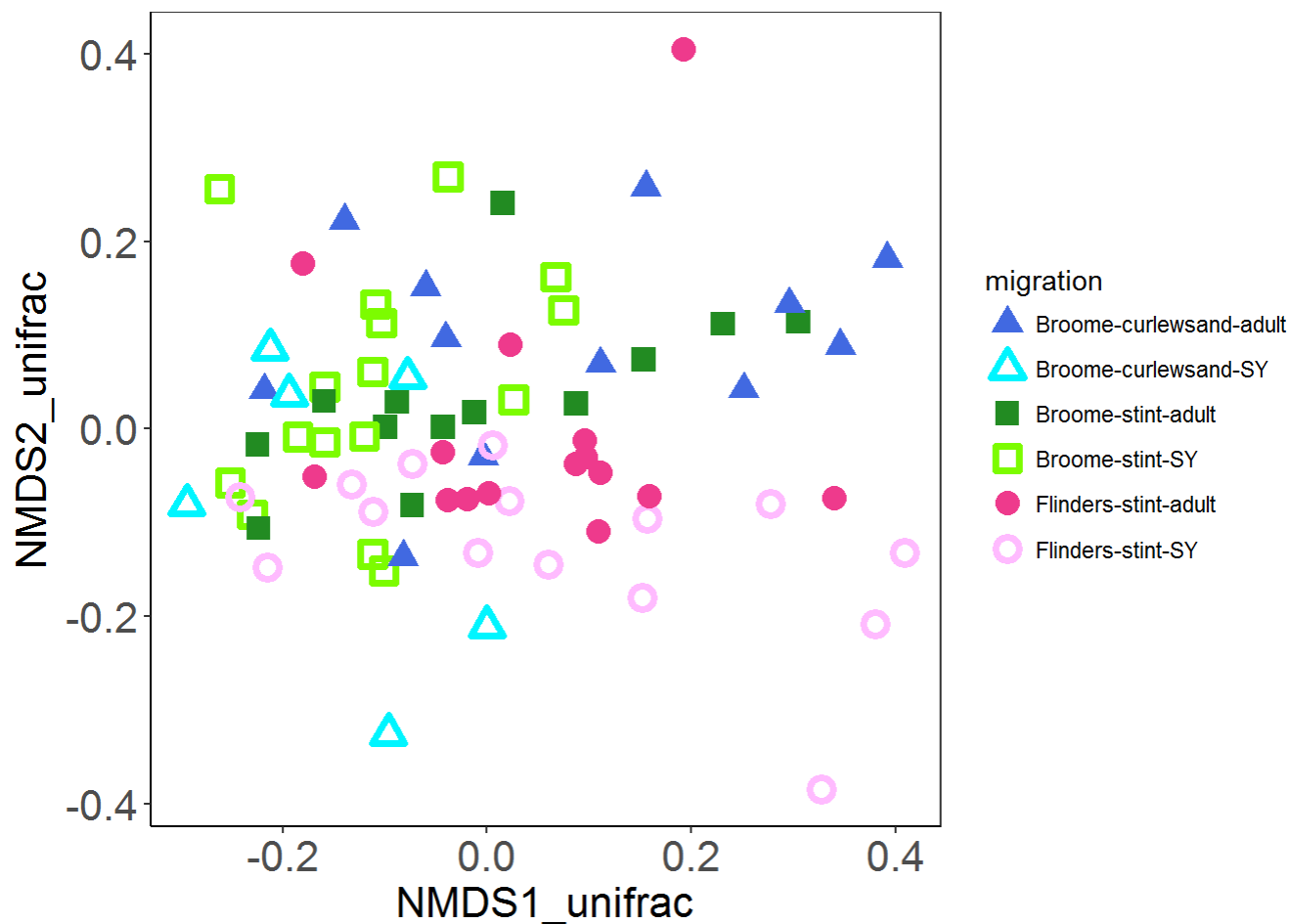
```
## Run 0 stress 0.2178487
## Run 1 stress 0.2322504
## Run 2 stress 0.2155708
## ... New best solution
## ... Procrustes: rmse 0.06628759  max resid 0.2890222
## Run 3 stress 0.2271085
## Run 4 stress 0.2286939
## Run 5 stress 0.2196326
## Run 6 stress 0.2349434
## Run 7 stress 0.2226841
## Run 8 stress 0.2133707
## ... New best solution
## ... Procrustes: rmse 0.02225117  max resid 0.1186602
## Run 9 stress 0.2190638
## Run 10 stress 0.2108544
## ... New best solution
## ... Procrustes: rmse 0.06605822  max resid 0.2885335
## Run 11 stress 0.2209799
## Run 12 stress 0.2172837
## Run 13 stress 0.22129
## Run 14 stress 0.2105399
## ... New best solution
## ... Procrustes: rmse 0.02824563  max resid 0.1835601
## Run 15 stress 0.2231375
## Run 16 stress 0.2185725
## Run 17 stress 0.2123678
## Run 18 stress 0.2131853
## Run 19 stress 0.2172729
## Run 20 stress 0.2181311
## *** No convergence -- monoMDS stopping criteria:
##      20: stress ratio > sratmax
```

```
## add unifrac ordination axes to dataframe and plot
```

```
unifrac<-data.frame(migration_unifrac$points)
sample_data(migration)$NMDS1_unifrac<-unifrac$MDS1
sample_data(migration)$NMDS2_unifrac<-unifrac$MDS2

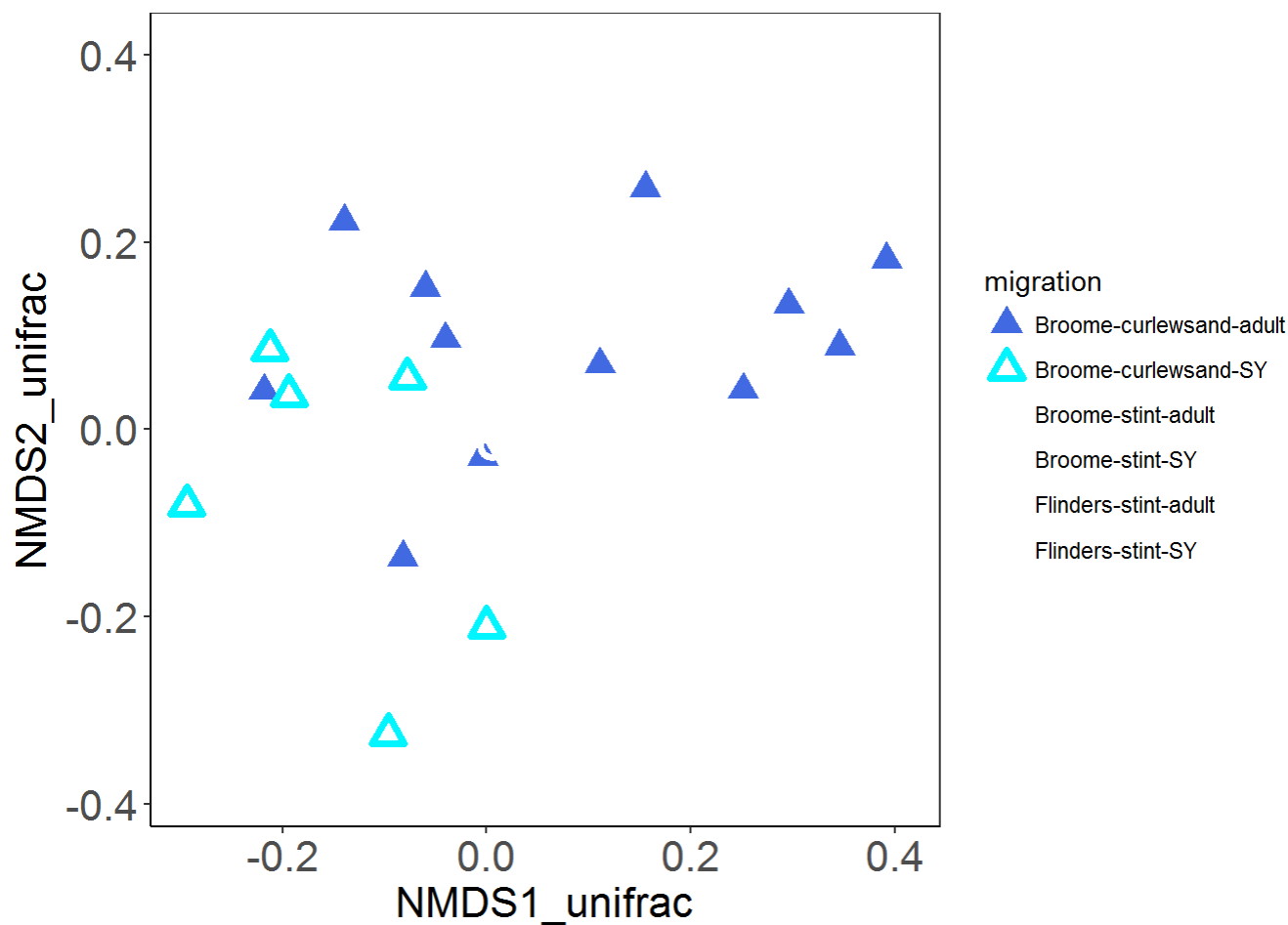
migration.df<-data.frame(sample_data(migration))
```

```
ggplot(data=migration.df, aes(x = NMDS1_unifrac, y = NMDS2_unifrac))+ geom_point(aes(colour =
migration, shape = migration), alpha = 1, size = 3, stroke = 2) +
  theme_bw()+scale_color_manual(values=c("royalblue","turquoise1",
                                         "forestgreen","lawngreen","violetred2","plum1"))+
  scale_shape_manual(values=c(17,2,15,0,16,1))+theme(panel.grid.major = element_blank(), panel
.grid.minor = element_blank(),
                                                    panel.background = element_blank(), axis.
line = element_line(colour = "black"))+
  theme(axis.text=element_text(size=16), axis.title=element_text(size=16))
```



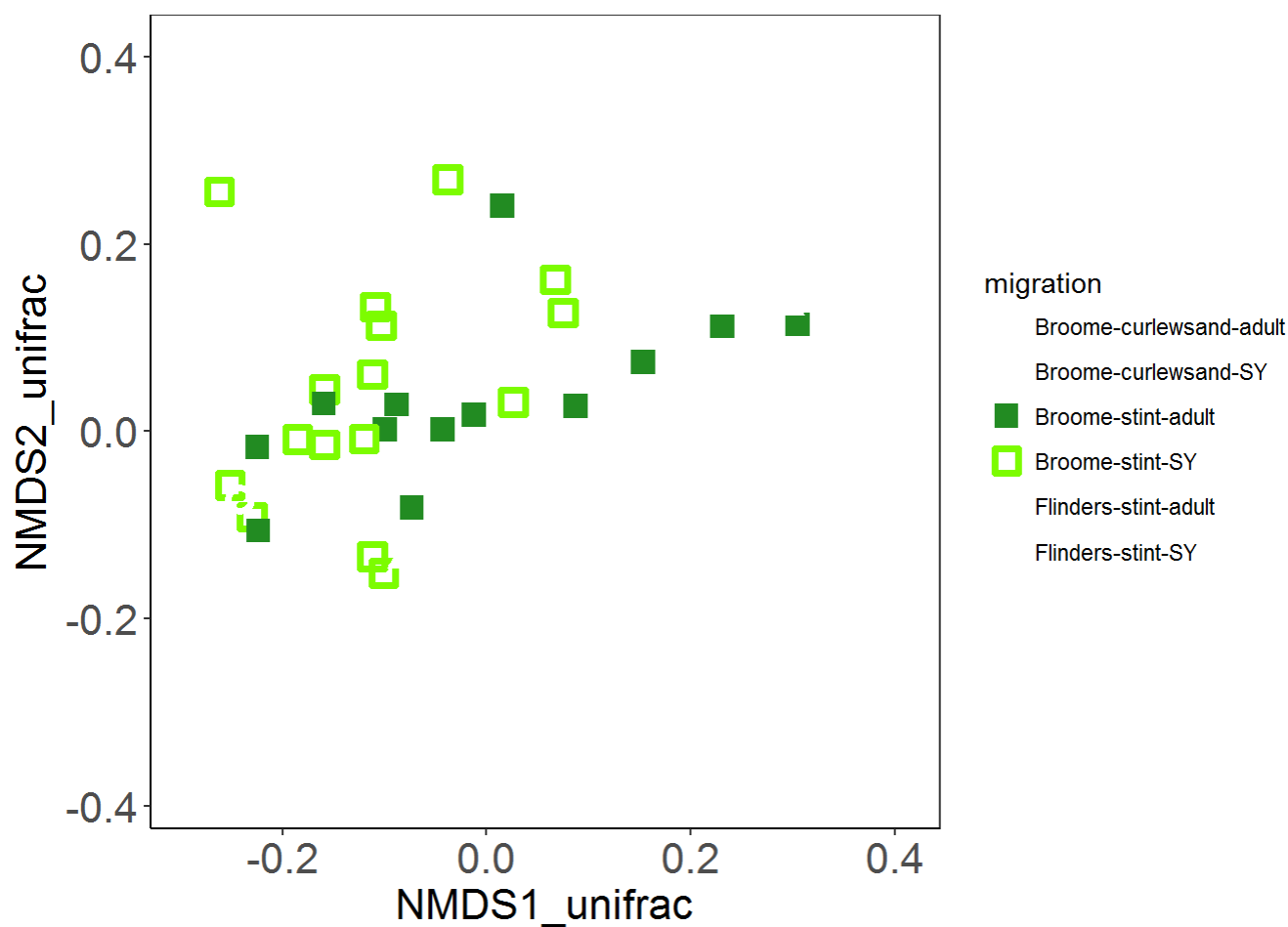
```
#####
```

```
ggplot(data=migration.df, aes(x = NMDS1_unifrac, y = NMDS2_unifrac))+ geom_point(aes(colour =
migration, shape = migration), alpha = 1, size = 3, stroke = 2) +
  theme_bw()+scale_color_manual(values=c("royalblue","turquoise1",
                                         "white","white","white","white"))+
  scale_shape_manual(values=c(17,2,15,0,16,1))+theme(panel.grid.major = element_blank(), panel
.grid.minor = element_blank(),
                                                    panel.background = element_blank(), axis.
line = element_line(colour = "black"))+
  theme(axis.text=element_text(size=16), axis.title=element_text(size=16))
```



```
#####
```

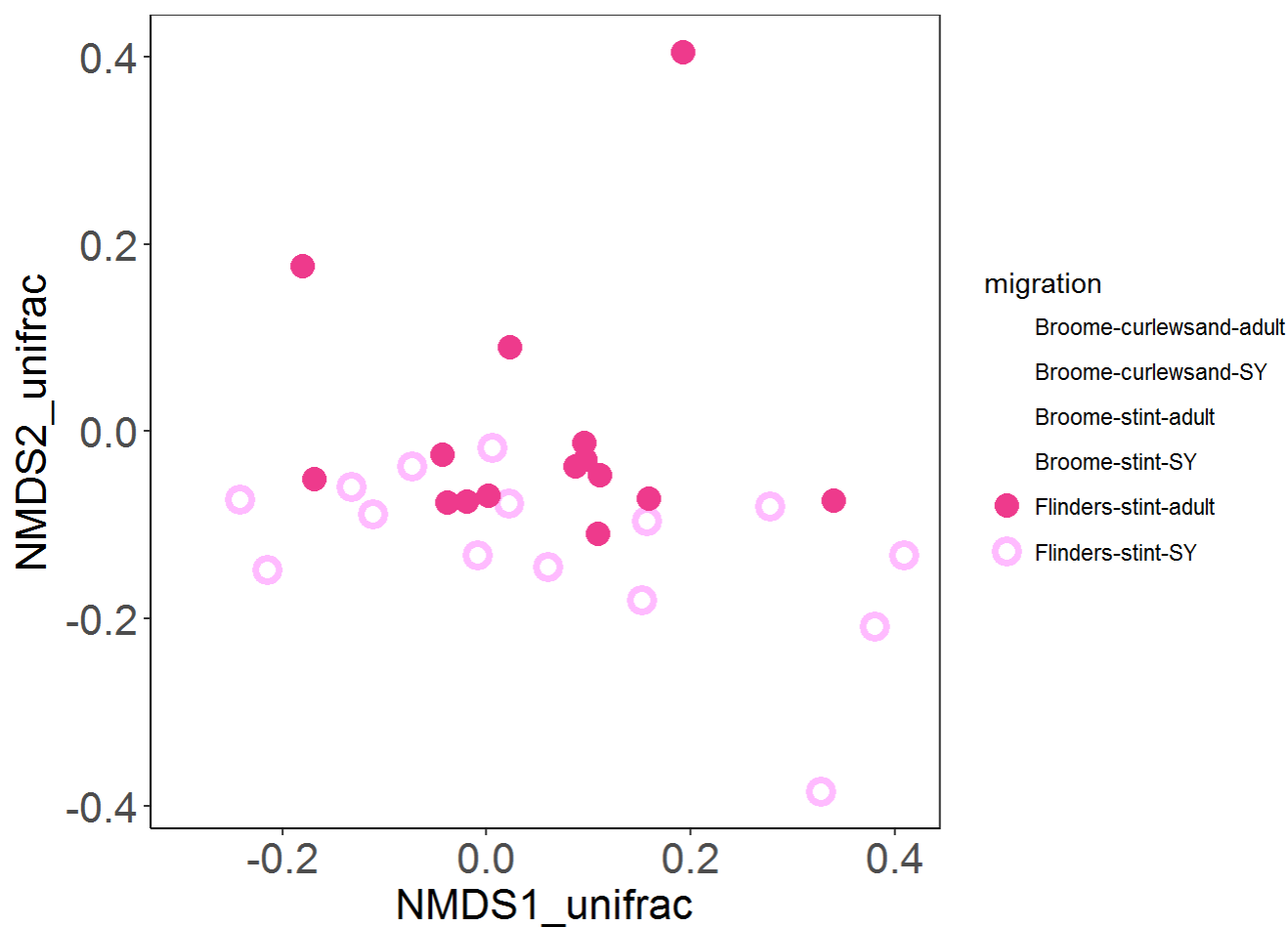
```
ggplot(data=migration.df, aes(x = NMDS1_unifrac, y = NMDS2_unifrac))+ geom_point(aes(colour =
migration, shape = migration), alpha = 1, size = 3, stroke = 2) +
  theme_bw()+scale_color_manual(values=c("white","white",
                                         "forestgreen","lawngreen","white","white"))+
  scale_shape_manual(values=c(17,2,15,0,16,1))+theme(panel.grid.major = element_blank(), panel
.grid.minor = element_blank(),
                                                    panel.background = element_blank(), axis.
line = element_line(colour = "black"))+
  theme(axis.text=element_text(size=16), axis.title=element_text(size=16))
```



```
#####
```

```
ggplot(data=migration.df, aes(x = NMDS1_unifrac, y = NMDS2_unifrac))+ geom_point(aes(colour =
migration, shape = migration), alpha = 1, size = 3, stroke = 2) +
  theme_bw()+scale_color_manual(values=c("white","white",
                                         "white","white","violetred2","plum1"))+
  scale_shape_manual(values=c(17,2,15,0,16,1))+theme(panel.grid.major = element_blank(), panel
.grid.minor = element_blank(),
                                                    panel.background = element_blank(), axis.
line = element_line(colour = "black"))+
  theme(axis.text=element_text(size=16), axis.title=element_text(size=16))
```





```
#####
```

```
##stats
```

```
migration_unifrac <- phyloseq::distance(migration, method = "unifrac")
```

```
## Warning in UniFrac(physeq, ...): Randomly assigning root as -- Otu004229 --
## in the phylogenetic tree in the data you provided.
```

```
# make a data frame from the sample_data
sampledf <- data.frame(sample_data(migration))
```

```
# Adonis test
```

```
adonis(migration_unifrac ~ site+species+age, data = sampledf)
```

```
##
## Call:
## adonis(formula = migration_unifrac ~ site + species + age, data = sampledf)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
```

```
## site      1      0.8129 0.81295  3.3509 0.04143  0.001 ***
## species   1      0.5368 0.53678  2.2126 0.02736  0.001 ***
## age       1      0.5606 0.56056  2.3106 0.02857  0.001 ***
## Residuals 73     17.7102 0.24261          0.90264
## Total     76     19.6205          1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#####
```

```
#####bray curtis
```

```
##Figure 3)
```

```
set.seed(1)
```

```
migration_bray <- ordinate(
  physeq = migration,
  method = "NMDS",
  distance = "bray"
)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.2453602
## Run 1 stress 0.2607537
## Run 2 stress 0.255154
## Run 3 stress 0.2561769
## Run 4 stress 0.2518526
## Run 5 stress 0.2718302
## Run 6 stress 0.2533695
## Run 7 stress 0.2584876
## Run 8 stress 0.2453605
## ... Procrustes: rmse 0.0003253219  max resid 0.002527825
## ... Similar to previous best
## Run 9 stress 0.261244
## Run 10 stress 0.2517643
## Run 11 stress 0.2439716
## ... New best solution
## ... Procrustes: rmse 0.04242624  max resid 0.3241008
## Run 12 stress 0.2649426
## Run 13 stress 0.2657925
## Run 14 stress 0.2448462
## Run 15 stress 0.243968
## ... New best solution
## ... Procrustes: rmse 0.0320396  max resid 0.2619141
## Run 16 stress 0.26698
## Run 17 stress 0.2443094
## ... Procrustes: rmse 0.04255965  max resid 0.2075804
## Run 18 stress 0.2705507
## Run 19 stress 0.2493743
## Run 20 stress 0.2500642
## *** No convergence -- monoMDS stopping criteria:
```

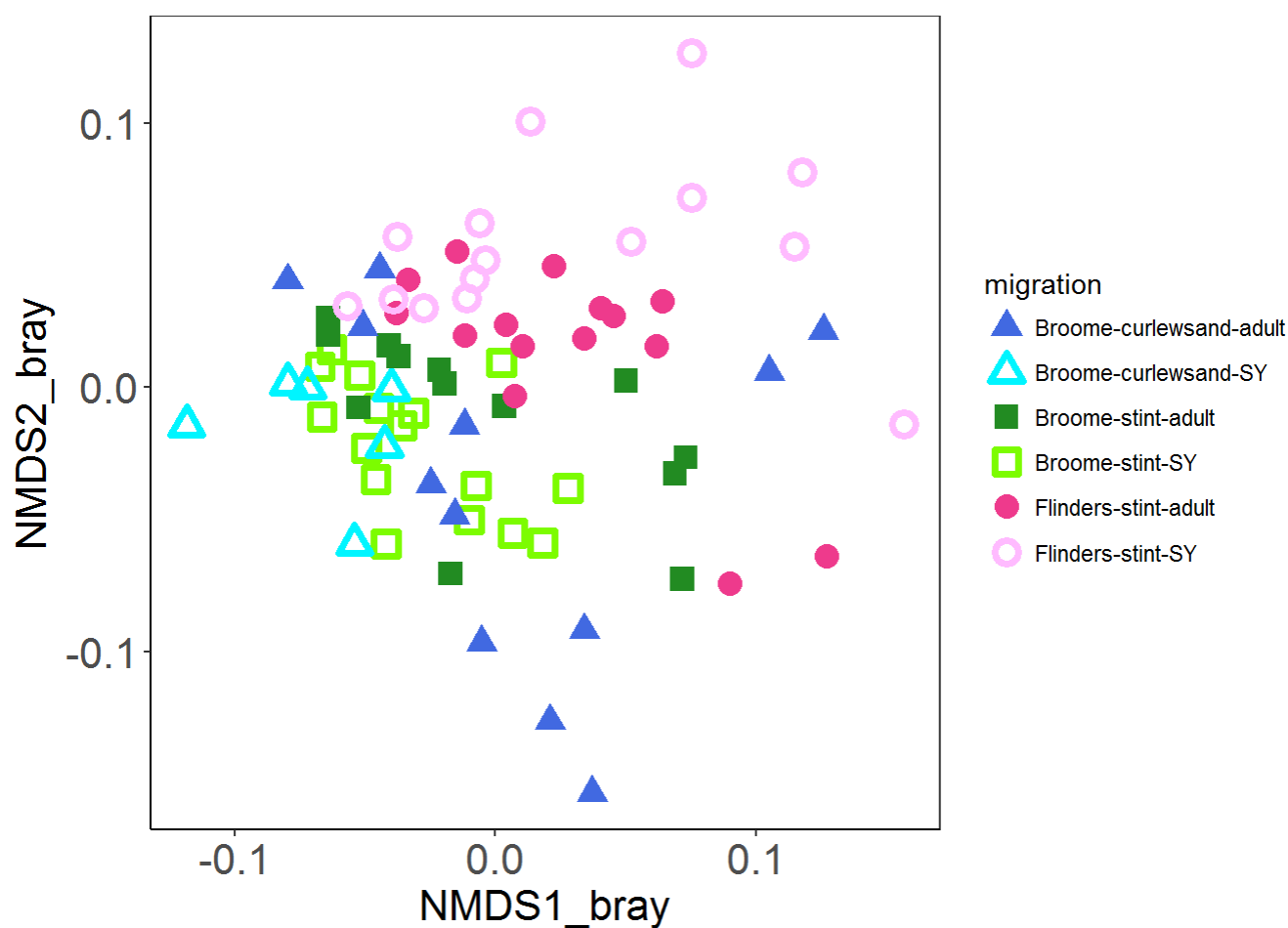
```
##      20: stress ratio > sratmax
```

```
bray<-data.frame(migration_bray$points)
sample_data(migration)$NMDS1_bray<-bray$MDS1
sample_data(migration)$NMDS2_bray<-bray$MDS2

migration.df<-data.frame(sample_data(migration))

##plot

ggplot(data=migration.df, aes(x = NMDS1_bray, y = NMDS2_bray))+
  geom_point(aes(colour = migration, shape = migration), alpha = 1, size = 3, stroke = 2) +
  theme_bw()+scale_color_manual(values=c("royalblue", "turquoise1",
                                          "forestgreen", "lawngreen", "violetred2", "plum1"))+
  scale_shape_manual(values=c(17, 2, 15, 0, 16, 1))+theme(panel.grid.major = element_blank(),
                                                         panel.grid.minor = element_blank(),
                                                         panel.background = element_blank(),
                                                         axis.line = element_line(colour = "black"
                                                         ))+
  theme(axis.text=element_text(size=16), axis.title=element_text(size=16))
```



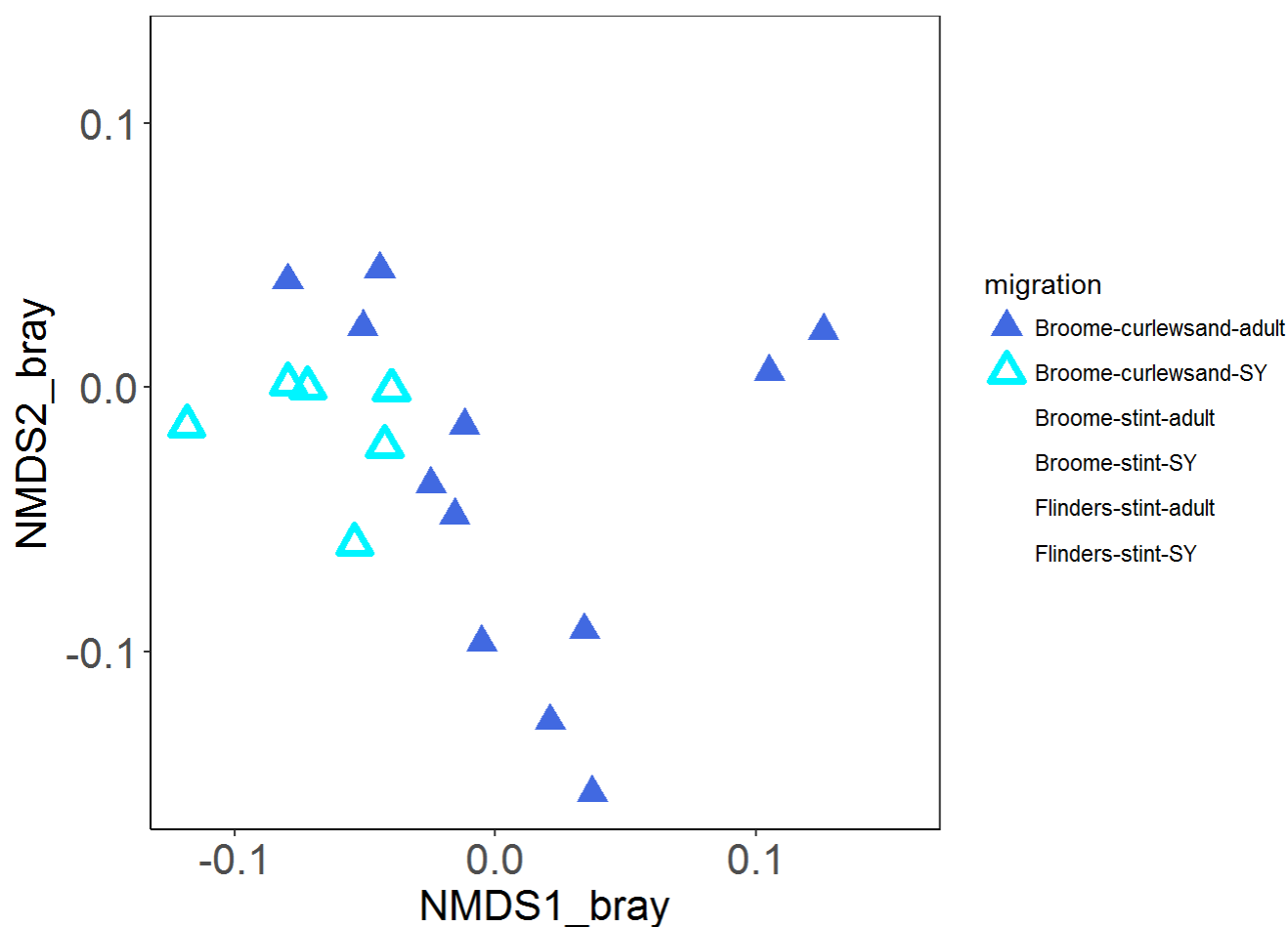
```
#####
#####

ggplot(data=migration.df, aes(x = NMDS1_bray, y = NMDS2_bray))+
```

```

geom_point(aes(colour = migration, shape = migration), alpha = 1, size = 3, stroke = 2) +
theme_bw()+scale_color_manual(values=c("royalblue","turquoise1",
                                         "white","white","white","white"))+
scale_shape_manual(values=c(17,2,15,0,16,1))+theme(panel.grid.major = element_blank(),
                                                    panel.grid.minor = element_blank(),
                                                    panel.background = element_blank(),
                                                    axis.line = element_line(colour = "black"
)))+
theme(axis.text=element_text(size=16), axis.title=element_text(size=16))

```

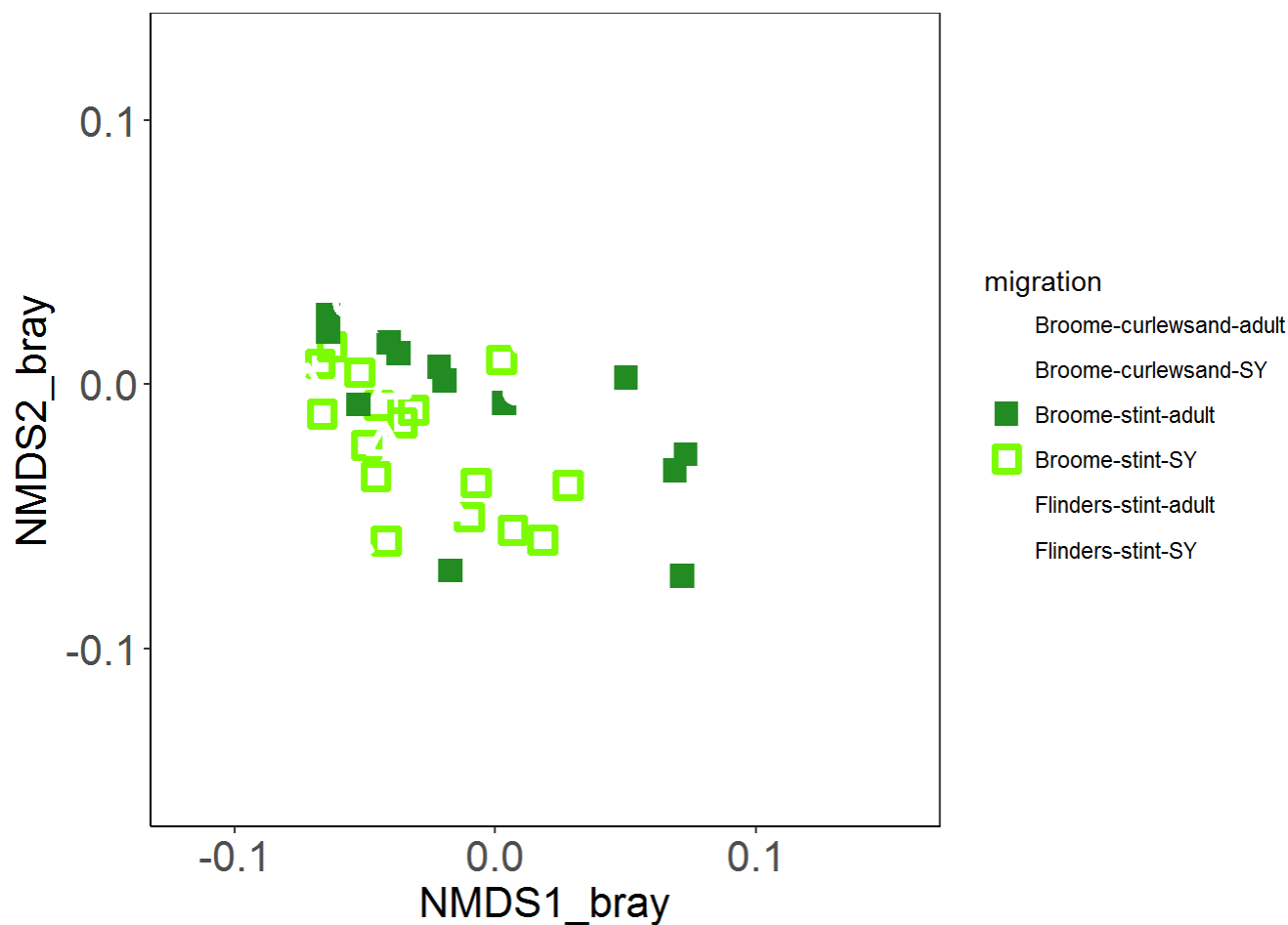


```
#####
```

```

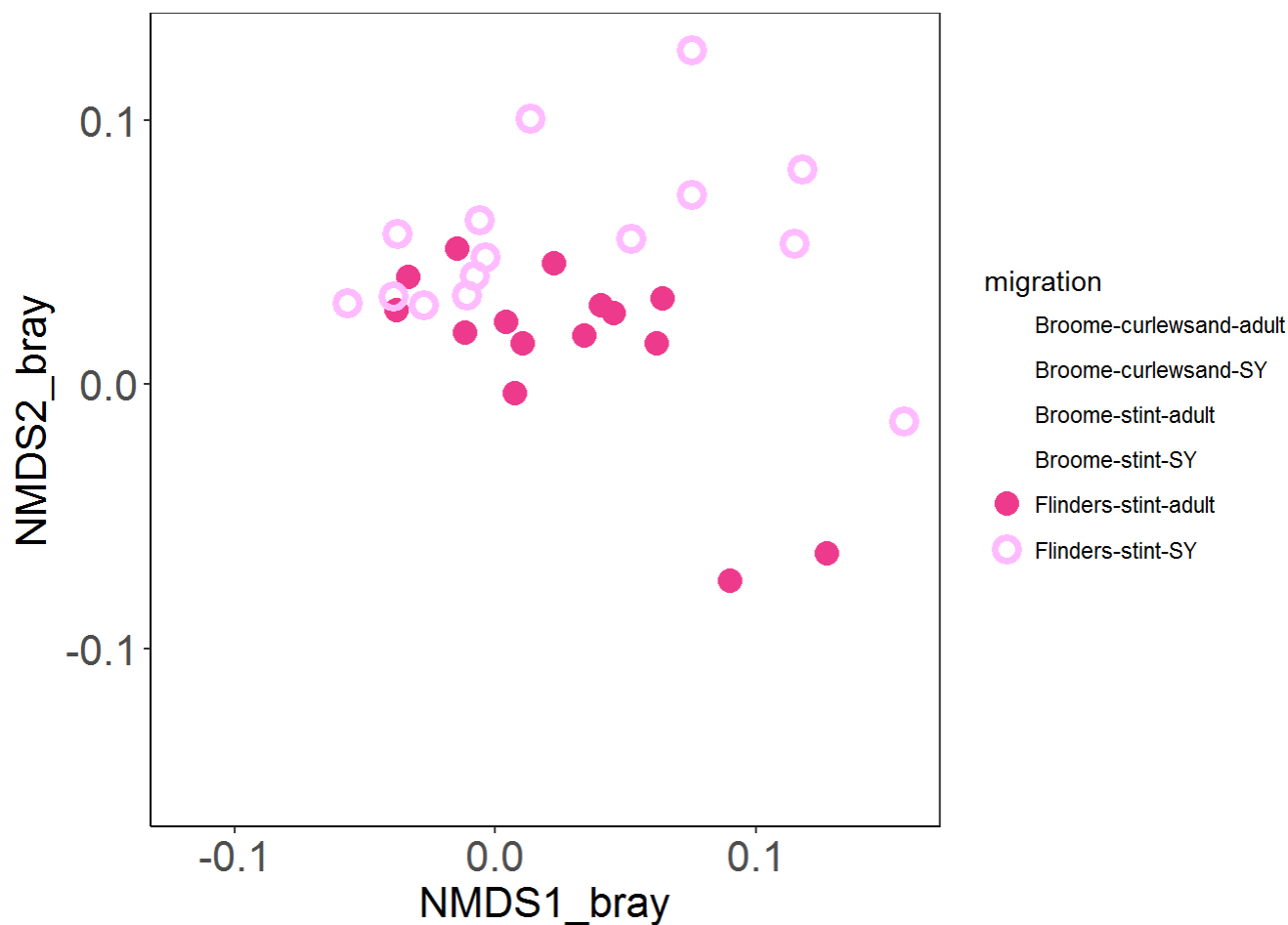
ggplot(data=migration.df, aes(x = NMDS1_bray, y = NMDS2_bray))+
geom_point(aes(colour = migration, shape = migration), alpha = 1, size = 3, stroke = 2) +
theme_bw()+scale_color_manual(values=c("white","white",
                                         "forestgreen","lawngreen","white","white"))+
scale_shape_manual(values=c(17,2,15,0,16,1))+theme(panel.grid.major = element_blank(),
                                                    panel.grid.minor = element_blank(),
                                                    panel.background = element_blank(),
                                                    axis.line = element_line(colour = "black"
)))+
theme(axis.text=element_text(size=16), axis.title=element_text(size=16))

```



```
#####
```

```
ggplot(data=migration.df, aes(x = NMDS1_bray, y = NMDS2_bray))+
  geom_point(aes(colour = migration, shape = migration), alpha = 1, size = 3, stroke = 2) +
  theme_bw()+scale_color_manual(values=c("white","white",
                                         "white","white","violetred2","plum1"))+
  scale_shape_manual(values=c(17,2,15,0,16,1))+theme(panel.grid.major = element_blank(),
                                                    panel.grid.minor = element_blank(),
                                                    panel.background = element_blank(),
                                                    axis.line = element_line(colour = "black"
                                                    ))+
  theme(axis.text=element_text(size=16), axis.title=element_text(size=16))
```



```
#####
```

```
##stats
```

```
migration_bray <- phyloseq::distance(migration, method = "bray")
```

```
# make a data frame from the sample_data
sampledf <- data.frame(sample_data(migration))
```

```
# Adonis test
adonis(migration_bray ~ site+species+age, data = sampledf)
```

```
##
## Call:
## adonis(formula = migration_bray ~ site + species + age, data = sampledf)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model    R2 Pr(>F)
## site      1    1.1939 1.19392   3.4024 0.04114 0.001 ***
## species   1    0.9877 0.98765   2.8146 0.03403 0.001 ***
```

```
## age      1      1.2225 1.22247  3.4837 0.04212  0.001 ***
## Residuals 73      25.6163 0.35091           0.88270
## Total     76      29.0203           1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#####
```

```
##remove corynebacteria to see if any other differences
```

```
migration1<-subset_taxa(migration, Order != "Corynebacteriales")
migration1<- prune_taxa(taxa_sums(migration1)>0, migration1)
```

```
set.seed(1)
```

```
migration_bray <- ordinate(
  physeq = migration1,
  method = "NMDS",
  distance = "bray"
)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.2434372
## Run 1 stress 0.2520435
## Run 2 stress 0.2505532
## Run 3 stress 0.2414003
## ... New best solution
## ... Procrustes: rmse 0.05723422  max resid 0.3325888
## Run 4 stress 0.2407164
## ... New best solution
## ... Procrustes: rmse 0.04558418  max resid 0.3847929
## Run 5 stress 0.2636338
## Run 6 stress 0.2525901
## Run 7 stress 0.2554168
## Run 8 stress 0.2445606
## Run 9 stress 0.2592545
## Run 10 stress 0.2590405
## Run 11 stress 0.2401147
## ... New best solution
## ... Procrustes: rmse 0.04837525  max resid 0.3577633
## Run 12 stress 0.2539493
## Run 13 stress 0.248226
## Run 14 stress 0.2423354
## Run 15 stress 0.2504476
## Run 16 stress 0.2600155
## Run 17 stress 0.2488418
## Run 18 stress 0.2526451
## Run 19 stress 0.240373
## ... Procrustes: rmse 0.03543252  max resid 0.2868491
## Run 20 stress 0.2567804
## *** No convergence -- monoMDS stopping criteria:
```

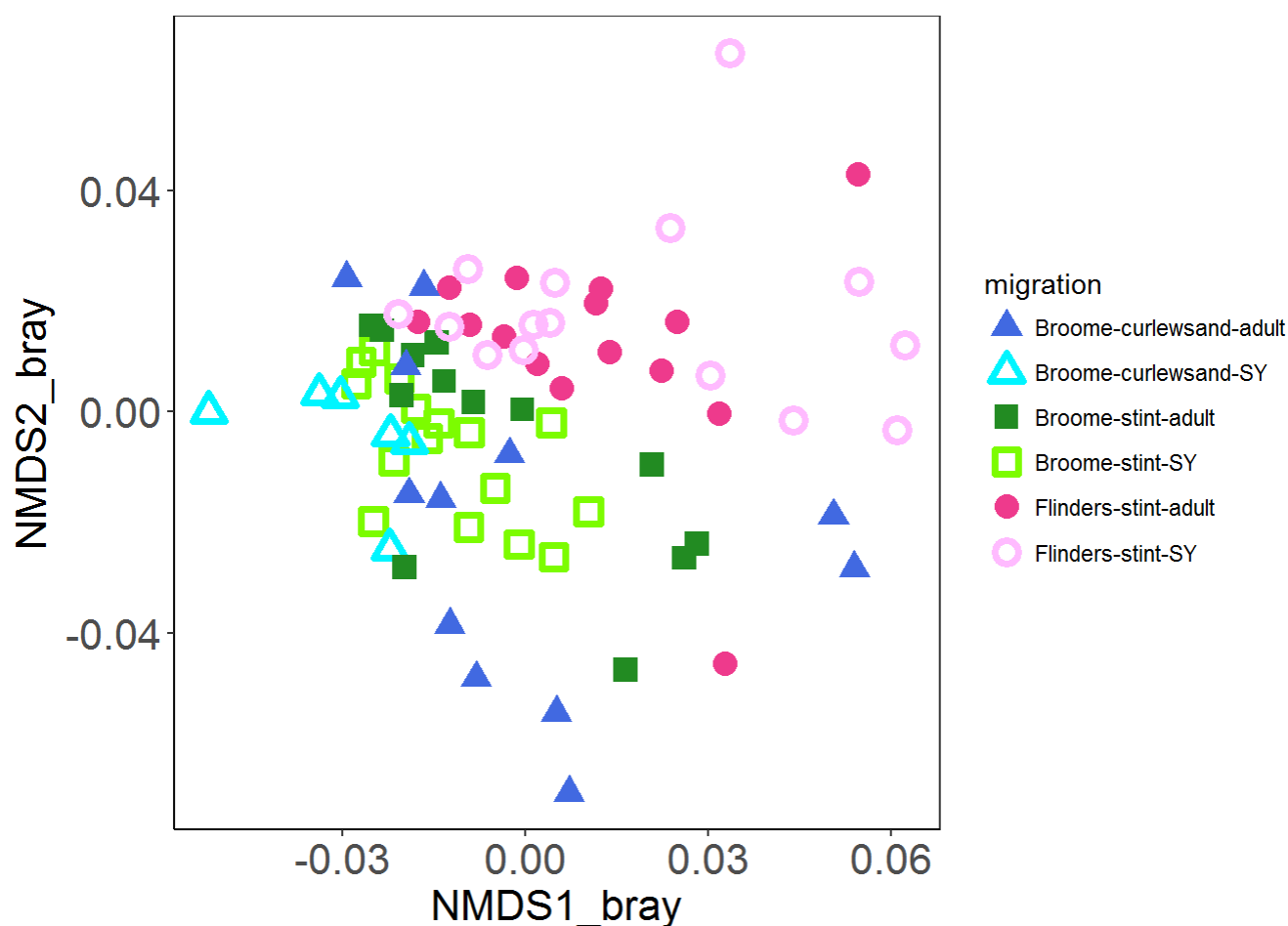
```
##      20: stress ratio > sratmax
```

```
bray<-data.frame(migration_bray$points)
sample_data(migration1)$NMDS1_bray<-bray$MDS1
sample_data(migration1)$NMDS2_bray<-bray$MDS2

migration1.df<-data.frame(sample_data(migration1))

##plot

ggplot(data=migration1.df, aes(x = NMDS1_bray, y = NMDS2_bray))+
  geom_point(aes(colour = migration, shape = migration), alpha = 1, size = 3, stroke = 2) +
  theme_bw()+scale_color_manual(values=c("royalblue","turquoise1",
                                          "forestgreen","lawngreen","violetred2","plum1"))+
  scale_shape_manual(values=c(17,2,15,0,16,1))+theme(panel.grid.major = element_blank(),
                                                    panel.grid.minor = element_blank(),
                                                    panel.background = element_blank(),
                                                    axis.line = element_line(colour = "black"
                                                    ))+
  theme(axis.text=element_text(size=16), axis.title=element_text(size=16))
```



```
##stats
```

```
migration_bray <- phyloseq::distance(migration1, method = "bray")
#migration_bray <- phyloseq::distance(migration1, method = "unifrac")
```



```
# make a data frame from the sample_data
sampledf <- data.frame(sample_data(migration1))

# Adonis test
adonis(migration_bray ~ site+species+age, data = sampledf)
```

```
##
## Call:
## adonis(formula = migration_bray ~ site + species + age, data = sampledf)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##              Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## site          1    1.0202 1.02019   2.8284 0.03539  0.001 ***
## species        1    0.9369 0.93687   2.5974 0.03250  0.001 ***
## age            1    0.5401 0.54012   1.4974 0.01874  0.042 *
## Residuals     73    26.3305 0.36069           0.91338
## Total          76    28.8277           1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#####

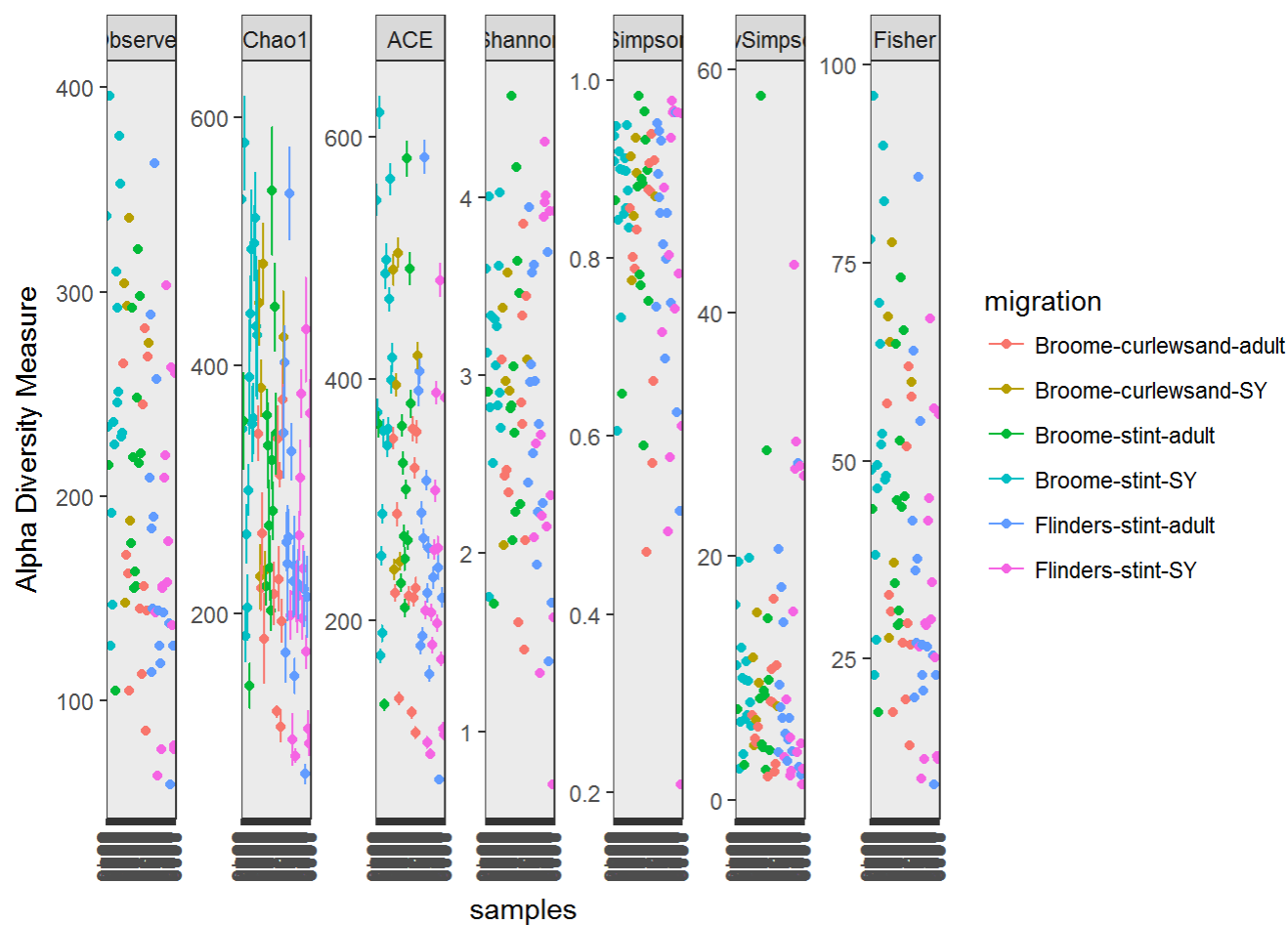
#####

#####
#####

####alpha diversity
### Figure 3c)

plot_richness(migration, color = "migration")
```

```
## Warning: Removed 385 rows containing missing values (geom_errorbar).
```



```

min_lib <- min(sample_sums(migration))

# Initialize matrices to store richness and evenness estimates
nsamp = nsamples(migration)
trials = 100

richness <- matrix(nrow = nsamp, ncol = trials)
row.names(richness) <- sample_names(migration)

evenness <- matrix(nrow = nsamp, ncol = trials)
row.names(evenness) <- sample_names(migration)

###

set.seed(3)

for (i in 1:100) {
  # Subsample
  r <- rarefy_even_depth(migration, sample.size = min_lib, verbose = FALSE, replace = TRUE)

  # Calculate richness
  rich <- as.numeric(as.matrix(estimate_richness(r, measures = "Observed")))
  richness[,i] <- rich

  # Calculate evenness

```

```
even <- as.numeric(as.matrix(estimate_richness(r, measures = "InvSimpson")))
evenness[,i] <- even
}

SampleID <- row.names(richness)
mean <- apply(richness, 1, mean)
sd <- apply(richness, 1, sd)
measure <- rep("Richness", nsamp)
rich_stats <- data.frame(SampleID, mean, sd, measure)

SampleID <- row.names(evenness)
mean <- apply(evenness, 1, mean)
sd <- apply(evenness, 1, sd)
measure <- rep("Inverse Simpson", nsamp)
even_stats <- data.frame(SampleID, mean, sd, measure)

s <- data.frame(sample_data(migration))

alphadiv_rich_observed <- merge(rich_stats, s, by = "row.names")
alphadiv_even_inverse <- merge(even_stats, s, by = "row.names")

#summary stats

tapply(alphadiv_rich_observed$mean, alphadiv_rich_observed$migration, summary)
```

```
## $`Broome-curlewsand-adult`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    74.05  116.02  132.22  152.53  210.12  251.35
##
## $`Broome-curlewsand-SY`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    118.6   176.5   236.0   212.4   248.5   274.6
##
## $`Broome-stint-adult`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    91.52  139.22  176.77  179.97  215.02  262.04
##
## $`Broome-stint-SY`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    107.2   185.0   206.9   216.7   257.8   324.4
##
## $`Flinders-stint-adult`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    52.94  103.52  118.65  145.55  162.16  294.01
##
## $`Flinders-stint-SY`
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    54.81   92.65  141.46  141.66  189.76  256.55
```

```
unique(alphadiv_rich_observed$migration)
```

```
## [1] Broome-stint-SY          Broome-stint-adult        Broome-curlewsand-adult
```

```
## [4] Broome-curlewsand-SY      Flinders-stint-adult      Flinders-stint-SY
## 6 Levels: Broome-curlewsand-adult ... Flinders-stint-SY
```

```
group1a<-subset(alphadiv_rich_observed, migration=="Broome-curlewsand-adult")
sd(group1a$mean)
```

```
## [1] 59.45963
```

```
group1b<-subset(alphadiv_rich_observed, migration=="Broome-curlewsand-SY")
sd(group1b$mean)
```

```
## [1] 60.1231
```

```
group2a<-subset(alphadiv_rich_observed, migration=="Broome-stint-adult")
sd(group2a$mean)
```

```
## [1] 53.34191
```

```
group2b<-subset(alphadiv_rich_observed, migration=="Broome-stint-SY")
sd(group2b$mean)
```

```
## [1] 62.84926
```

```
group3a<-subset(alphadiv_rich_observed, migration=="Flinders-stint-adult")
sd(group3a$mean)
```

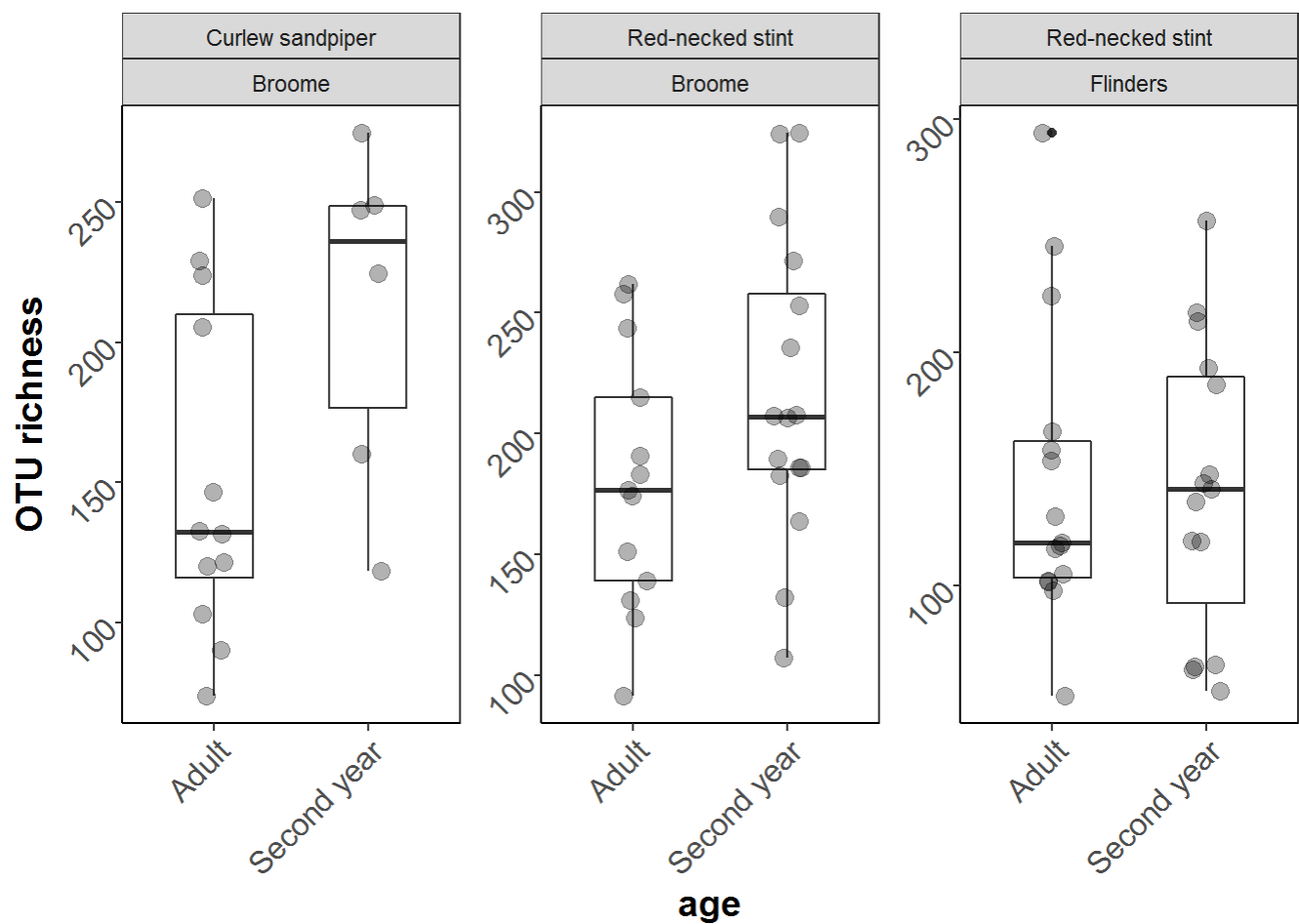
```
## [1] 64.27512
```

```
group3b<-subset(alphadiv_rich_observed, migration=="Flinders-stint-SY")
sd(group3b$mean)
```

```
## [1] 62.57409
```

```
##observed richness
```

```
ggplot(alphadiv_rich_observed, aes(x = age, y = mean))+geom_boxplot(width=0.5)+
  geom_jitter(width=0.1, alpha = 0.3, size=3)+ylab("OTU richness")+
  theme(axis.text=element_text(size=12, angle=45, hjust=1), axis.title=element_text(size=14, fa
ce="bold"))+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"))+
  facet_wrap(species~site, scales="free", nrow=1, ncol=3)
```



```
#####SHANNON

set.seed(3)

for (i in 1:100) {
  # Subsample
  r <- rarefy_even_depth(migration, sample.size = min_lib, verbose = FALSE, replace = TRUE)

  # Calculate richness
  rich <- as.numeric(as.matrix(estimate_richness(r, measures = "Shannon")))
  richness[ ,i] <- rich

  # Calculate evenness
  even <- as.numeric(as.matrix(estimate_richness(r, measures = "InvSimpson")))
  evenness[ ,i] <- even
}

SampleID <- row.names(richness)
mean <- apply(richness, 1, mean)
sd <- apply(richness, 1, sd)
measure <- rep("Richness", nsamp)
rich_stats <- data.frame(SampleID, mean, sd, measure)

SampleID <- row.names(evenness)
mean <- apply(evenness, 1, mean)
sd <- apply(evenness, 1, sd)
```

```
measure <- rep("Inverse Simpson", nsamp)
even_stats <- data.frame(SampleID, mean, sd, measure)

s <- data.frame(sample_data(migration))

alphadiv_rich_observed <- merge(rich_stats, s, by = "row.names")
alphadiv_even_inverse <- merge(even_stats, s, by = "row.names")

#summary stats

tapply(alphadiv_rich_observed$mean, alphadiv_rich_observed$migration, summary)
```

```
## $`Broome-curlewsand-adult`
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      1.449   2.263   2.585   2.624   3.128   3.824
##
## $`Broome-curlewsand-SY`
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      2.032   2.908   3.000   2.971   3.279   3.545
##
## $`Broome-stint-adult`
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      1.708   2.257   2.814   2.936   3.433   4.546
##
## $`Broome-stint-SY`
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      1.744   2.803   3.176   3.112   3.381   3.990
##
## $`Flinders-stint-adult`
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      1.392   2.244   2.717   2.723   3.307   3.918
##
## $`Flinders-stint-SY`
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      0.6943  2.1060  2.6075   2.7683  3.8974  4.2952
```

```
unique(alphadiv_rich_observed$migration)
```

```
## [1] Broome-stint-SY      Broome-stint-adult      Broome-curlewsand-adult
## [4] Broome-curlewsand-SY    Flinders-stint-adult    Flinders-stint-SY
## 6 Levels: Broome-curlewsand-adult ... Flinders-stint-SY
```

```
group1a<-subset(alphadiv_rich_observed, migration=="Broome-curlewsand-adult")
sd(group1a$mean)
```

```
## [1] 0.720232
```

```
group1b<-subset(alphadiv_rich_observed, migration=="Broome-curlewsand-SY")
sd(group1b$mean)
```

```
## [1] 0.5237586
```

```
group2a<-subset(alphadiv_rich_observed, migration=="Broome-stint-adult")  
sd(group2a$mean)
```

```
## [1] 0.8243023
```

```
group2b<-subset(alphadiv_rich_observed, migration=="Broome-stint-SY")  
sd(group2b$mean)
```

```
## [1] 0.5628283
```

```
group3a<-subset(alphadiv_rich_observed, migration=="Flinders-stint-adult")  
sd(group3a$mean)
```

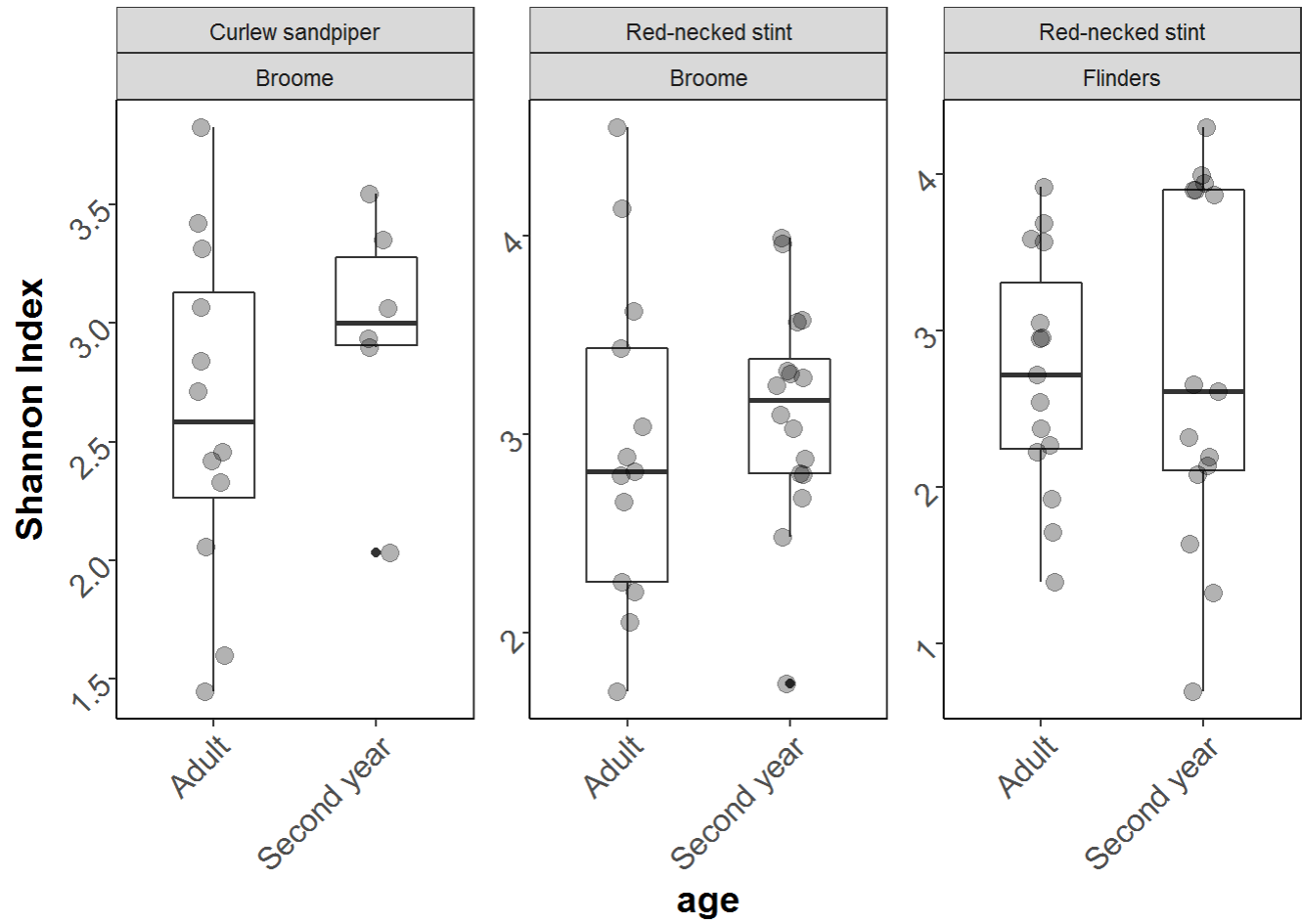
```
## [1] 0.7589452
```

```
group3b<-subset(alphadiv_rich_observed, migration=="Flinders-stint-SY")  
sd(group3b$mean)
```

```
## [1] 1.137082
```

```
##observed richness
```

```
ggplot(alphadiv_rich_observed, aes(x = age, y = mean))+geom_boxplot(width=0.5)+  
  geom_jitter(width=0.1, alpha = 0.3, size=3)+ylab("Shannon Index")+  
  theme(axis.text=element_text(size=12, angle=45, hjust=1), axis.title=element_text(size=14, fa  
ce="bold"))+  
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),  
        panel.background = element_blank(), axis.line = element_line(colour = "black"))+  
  facet_wrap(species~site, scales="free", nrow=1, ncol=3)
```



##### END #####