APPENDIX A

R Code Notebook for Data Analyses in R version 4.1.2

Investigating how bat ectoparasites influence the skin microbiome diversity and composition of Washington state bats

The objectives of this study are 1.) to investigate whether ectoparasite infestation in bats influences the skin microbiome diversity of bats, and 2.) to isolate and identify the most abundant bacteria from the skin of bats to compare them to the most abundant taxa from the sequencing data.

We hypothesized that bats with ectoparasites will have decreased skin microbiome diversity and altered composition compared to bats without ectoparasites, placing bats at a higher risk of Pd infection. Since culture methods are highly selective and many bacteria in a microbiome are unable to be cultured, we also hypothesized that the culturable bacterial isolates from western bats be in low relative abundances in our sequencing data.

1.) Cleaning up sampling metadata files in R and running preliminary data analyses:

- Basic information from the dataset: how many bats total, how many different bat species, sexes, how many ectoparasites by the different bat species?
- Run logistic regression (general linear mixed modeling) and an ANOVA to determine how ectoparasite presence differs by bat species (are some bat species more likely to have ectoparasites than others?)
- No microbiome data, just sampling data
- All outputs of statistical tests have been omitted in the code for sake of space.

```
#Set working directory to file with data you want to use
#Load in entire bat dataset - focusing on sampling, not microbiome
fullbat=read.csv("WDFW_EWU_Bat_Sampling_Data_Spring_2021.csv",header=T)

#Load in bat microbiome dataset
batmicrobiome=read.csv("Microbiome_WDFW_EWU_Bat_Sampling_Data_Spring_2021.csv
",header=T)

#How many different spp?
length(levels(fullbat$Species))

## [1] 0

#Set as a factor
fullbat$Species=factor(fullbat$Species)
length(levels(fullbat$Species))

## [1] 6
```

```r
#Look at frequency of diff spp, needs to be a data frame
speciestable=as.data.frame(table(fullbat$Species))


#Make a graph of bat spp observed
#load in ggplot2 to make pretty graphs
library(ggplot2)

#Bar graph with counts of each bat spp
ggplot(fullbat,aes(x=Species))+geom_bar()+theme_classic()+labs(y="Number of B
ats",title="Bat Species Observed",)

#grouped barplot, bat spp across the different roost sites (counties)
ggplot(fullbat,aes(x=Species,fill=County))+geom_bar(position="dodge")+theme_c
lassic()+labs(y="Number of Bats",title="Bat Species Observed")

#Look at ectoparasite presence across the bat spp
ggplot(fullbat,aes(x=Species,fill=Ectoparasites_.Y.N.))+geom_bar(position="do
dge")+theme_classic()+labs(y="Number of Bats",title="Bat Species Observed")

#Make a table of ectoparasite
speciesectotable=(table(fullbat$Ectoparasites_.Y.N.,fullbat$Species))
speciesectotable
##
##     COTO EPFU MYLU MYVO MYYU MYYU/LU
##   N    6    4   36    4   50      11
##   Y    0    8    6   12    7       3

ggplot(fullbat,aes(x=Ectoparasites_.Y.N.,fill=Species))+geom_bar(position="do
dge")+theme_classic()+labs(y="Number of Bats",title="Bat Species Observed")

#Run Chi-Square on Spp/Ecto table, null=no association btw spp and EP = same
proportion
chisq.test(speciesectotable)

#some #s really small so approx may be incorrect, less than 20% of #s in tabl
e should be less than 5

#run Fisher's, no assumption!
fisher.test(speciesectotable)

#some species have sig. diff portions of EPs, assuming those are representati
ve of the spp

#some spp more likely to have Eps than others

#which species specifically? Chi-sq table, big ones = diff; pairwise testing
with chisq.; turn into logistic regression, make ep the response var., sp as
predictor
#probability that bat has EP depending on what spp

#logistic regression
```

```
#make a new variable
fullbat$ecto=1
#finds bat w/o ecto, those = to 0
fullbat$ecto[fullbat$Ectoparasites_.Y.N.=="N"]=0
#run the logistic regression
model=glm(ecto~Species,data=fullbat,family="binomial")
summary(model)

#natural log of the odds (probability/not probability)in results table, coto
is intercept
#relative to coto, how sig diff?

#load in car library, run an anova
library(car)

## Loading required package: carData

Anova(model)

#make an effects plot
library(effects)

## lattice theme set by effectsTheme()
## See ?effectsTheme for details.

plot(allEffects(model),type="response")

#predicted probabilites of finding EP for each spp and 95% CI
#no variation in observed coto data, infinite uncertainty! Just b/c we didn't
find any?

#load in emmeans library (estimated means), will give us the p-values for eac
h comparison
library(emmeans)

#runs Tukey hsd for logistic regression, pairwise comparison
emmeans(model,pairwise~Species)

#compares spp to each other, gives p-vales, should match up with effects plot
#could be confounded by time and site, site as random effect

#mixed model, site as random effect? MYVO only on one site

#including site as a random effect
library(lme4)

## Loading required package: Matrix

#allows differences between spp to be different in diff counties
#making the least assumptions
model2=glmer(ecto~Species+(Species|County),data=fullbat,family=binomial)
```

```r
summary(model2)

#random effects, how much differences between counties, how much variation is
explained by the differences in counties
#similar numbers (lrg) = differences between counties

#run anova, makes 95% CI bigger, accounts for sites, use this for results!
#Uses proportions of ectoparasites, not counts = better representation
Anova(model2)
#Effects plot, ectoparasite presence by bat spp
plot(allEffects(model2),type="response")
#gives biger CI's
emmeans(model2,pairwise~Species)

#Number of sexes
table(fullbat$Sex_.M.F.)

##
##   F   M
## 142   5

#Number of sexes in each bat spp
table(fullbat$Sex_.M.F.,fullbat$Species)
##
##     COTO EPFU MYLU MYVO MYYU MYYU/LU
##   F    5   10   41   16   57      13
##   M    1    2    1    0    0       1

table(fullbat$Species)

##
##    COTO    EPFU    MYLU    MYVO    MYYU MYYU/LU
##       6      12      42      16      57      14

#Center the title on the graphs, add gg-title at the end of the code
#Make publication-worthy figures with centered titles

#Bar plot with number of bat species observed
ggplot(fullbat,aes(x=Species))+geom_bar()+theme_classic()+labs(y="Number of B
ats",title="Bat Species Observed")+ggtitle("Bat Species Observed") +
theme(plot.title = element_text(hjust = 0.5))

#Barplot with ectoparasites by bat species
ggplot(fullbat,aes(x=Species,fill=Ectoparasites_.Y.N.))+geom_bar(position="do
dge")+theme_classic()+labs(y="Number of Bats",title="Bat Species Observed")+g
gtitle("Ectoparasites Across Bat Species") +
  theme(plot.title = element_text(hjust = 0.5))

#Grouped bar plot, number of bats across counties
ggplot(fullbat,aes(x=Species,fill=County))+geom_bar(position="dodge")+theme_c
lassic()+labs(y="Number of Bats",title="Bat Species Observed")+ggtitle("Bat S
```

```
pecies Across Washington Counties") +
  theme(plot.title = element_text(hjust = 0.5))
```

2.) Calculating summary statistics from alpha diversity metrics.

- Dealing with our sequenced skin microbiome data.
- What are the means and standard deviations for the raw alpha diversity metric data?
    - Shannon's diversity, Faith's phylogenetic diversity, evenness, observed features
- I downloaded the raw TSV data files from the QIIME2 view website as I visualized the alpha diversity metrics. We then used those data files to calculate the means and standard deviations for all alpha diversity metrics for all variables (ectoparasite presence, bat species, roost location)

```
#Summary statistics for Shannon diversity results
#Read in the Shannon diversity alpha diversity results dataset
Shannon_bat=read.csv("Shannon_alpha_metadata.csv", header=T)

#Polish up the Shannon diversity dataset, remove unnecessary rows
names(Shannon_bat)=Shannon_bat[1,] #makes header names the true headers
Shannon_bat=Shannon_bat[-1,] #removes 1st row
Shannon_bat=Shannon_bat[-1,]

#Convert shannon entropy to numbers
Shannon_bat$shannon_entropy=as.numeric(Shannon_bat$shannon_entropy)

#Histogram
hist(Shannon_bat$shannon_entropy)

#ANOVA on Shannon entropy values
#Shannon entropy values by bat species
model1=aov(shannon_entropy~Species,data=Shannon_bat)
summary(model1)

#which bat species are different? TukeyHSD
TukeyHSD(model1)

library(effects)
plot(allEffects(model1)) #effects plot
plot(model1) #See all the different types of effects plots to check that the
data is normally distributed before moving on

#Shannon entropy values by roost location
model2=aov(shannon_entropy~Site,data=Shannon_bat)
summary(model2)

#which one, TukeyHSD
TukeyHSD(model2)
```

```
library(effects)
plot(allEffects(model2))
plot(model2)

#Shannon entropy values by ectoparasites
model3=aov(shannon_entropy~Ectoparasites,data=Shannon_bat)
summary(model3)

#which one, TukeyHSD
TukeyHSD(model3)

library(effects)
plot(allEffects(model3))
plot(model3)

summary(allEffects(model3)) #provides means, 95% CI not st.dev



#Shortcut - to just get means and standard deviations, us tapply
#Shannon summary stats by group, ectoparasites
tapply(Shannon_bat$shannon_entropy,Shannon_bat$Ectoparasites,summary)
tapply(Shannon_bat$shannon_entropy,Shannon_bat$Ectoparasites,sd)


#Shannon summary stats for bat species
tapply(Shannon_bat$shannon_entropy,Shannon_bat$Species,mean)
tapply(Shannon_bat$shannon_entropy,Shannon_bat$Species,sd)

#summary stats for site
tapply(Shannon_bat$shannon_entropy,Shannon_bat$Site,mean)
tapply(Shannon_bat$shannon_entropy,Shannon_bat$Site,sd)


#Evenness summary stats
#Read in evenness metadata - alpha diversity
Evenness_bat=read.csv("Evenness_alpha_metadata.csv", header=T)

#Polish evenness data
names(Evenness_bat)=Evenness_bat[1,]
Evenness_bat=Evenness_bat[-1,] #removes 1st row
Evenness_bat=Evenness_bat[-1,]

#Convert pielou's evenness to numbers so you can use it
Evenness_bat$pielou_evenness=as.numeric(Evenness_bat$pielou_evenness)

#Histogram
hist(Evenness_bat$pielou_evenness)

#Evenness summary stats by group, ectoparasites
tapply(Evenness_bat$pielou_evenness,Evenness_bat$Ectoparasites,mean)
```

```
tapply(Evenness_bat$pielou_evenness,Evenness_bat$Ectoparasites,sd)

#Evennesssummary stats for species
tapply(Evenness_bat$pielou_evenness,Evenness_bat$Species,mean)
tapply(Evenness_bat$pielou_evenness,Evenness_bat$Species,sd)
#Evenness summary stats for site
tapply(Evenness_bat$pielou_evenness,Evenness_bat$Site,mean)
tapply(Evenness_bat$pielou_evenness,Evenness_bat$Site,sd)


#Faith's phylogenetic diversity summary stats
#Read in faith's pd metadata - alpha diversity
Faith_pd_bat=read.csv("Faith_pd_alpha_metadata.csv", header=T)

#Clean up faith’s pd data
names(Faith_pd_bat)=Faith_pd_bat[1,]
Faith_pd_bat=Faith_pd_bat[-1,] #removes 1st row
Faith_pd_bat=Faith_pd_bat[-1,]

#Convert faith's pd to numbers
Faith_pd_bat$faith_pd=as.numeric(Faith_pd_bat$faith_pd)

#Histogram
hist(Faith_pd_bat$faith_pd)

#Faith pd summary stats by group, ectoparasites
tapply(Faith_pd_bat$faith_pd,Faith_pd_bat$Ectoparasites,mean)
tapply(Faith_pd_bat$faith_pd,Faith_pd_bat$Ectoparasites,sd)

#summary stats for species
tapply(Faith_pd_bat$faith_pd,Faith_pd_bat$Species,mean)
tapply(Faith_pd_bat$faith_pd,Faith_pd_bat$Species,sd)

#summary stats for site
tapply(Faith_pd_bat$faith_pd,Faith_pd_bat$Site,mean)
tapply(Faith_pd_bat$faith_pd,Faith_pd_bat$Site,sd)



#Observed features summary stats
#Read in obs. feat metadata - alpha diversity
Obs_feat_bat=read.csv("Observed_featured_alpha_metadata.csv", header=T)


#Polish observed features data
names(Obs_feat_bat)=Obs_feat_bat[1,]
Obs_feat_bat=Obs_feat_bat[-1,] #removes 1st row
Obs_feat_bat=Obs_feat_bat[-1,]
```

```
#Convert observed features to numbers
Obs_feat_bat$observed_features=as.numeric(Obs_feat_bat$observed_features)

#Histogram
hist(Obs_feat_bat$observed_features)

#summary stats by group, ectoparasites
tapply(Obs_feat_bat$observed_features,Obs_feat_bat$Ectoparasites,mean)
tapply(Obs_feat_bat$observed_features,Obs_feat_bat$Ectoparasites,sd)

#summary stats for species
tapply(Obs_feat_bat$observed_features,Obs_feat_bat$Species,mean)
tapply(Obs_feat_bat$observed_features,Obs_feat_bat$Species,sd)

#summary stats for site
tapply(Obs_feat_bat$observed_features,Obs_feat_bat$Site,mean)
tapply(Obs_feat_bat$observed_features,Obs_feat_bat$Site,sd)
```

3.) Running indicspecies analysis in R to determine how relative abundance of bacterial taxa differ by different variables (ectoparasite presence, bat species, roost location)

- How do the relative abundance of bacterial taxa differ between different groups (ectoparasites, bat species, roost location)?
- Using microbiome data, specifically the feature table from QIIME
- Be careful, make sure that you get the correct group for the correct variable
- We built boxplots for all of the indicspecies taxa since they were in such low abundances that they couldn't be seen on the grouped taxa bar plot

```
#Indicspecies Analysis
#Use the feature table from QIIME2, make sure you export it from QIIME2 and t
hen convert it to a .csv file
Bat = read.csv("Bat Feature Table_Excel.csv", header=T)

#Polish frequency table
#Trim extra taxonomy column from data table
Bat = Bat[,-92]
names(Bat)=Bat[2,]
Bat=Bat[-2,]
Ectoparasite=Bat[1,]#stores this data
Bat=Bat[-1,]
row.names(Bat)=Bat[,1]
Bat=Bat[,-1]#removes 1st row

for (i in 1:90) Bat[,i]=as.numeric(Bat[,i])#convert that one column


#Transpose dataset
```

```
Bat=t(Bat)

#Trim ectoparasite dataset
#Convert ectoparasite to character
Ectoparasite=as.character(Ectoparasite)#not a data frame
Ectoparasite=Ectoparasite[-1]
#Convert ectoparasite to 0s and 1s
#Convert to factor, then convert to numeric
Ectoparasite=as.factor(Ectoparasite)
Ectoparasite=as.numeric(Ectoparasite)
#2 = Y, 1 = N

#load in the indicspecies package
library(indicspecies)
#Run indicspecies analysis by ectoparasite presence (yes/no)

indval=multipatt(Bat,Ectoparasite)
summary(indval)

##  List of species associated to each combination:
##
##  Group 1  #sps.  1 #make sure you know which group corresponds to what
##
##  Group 2  #sps.  4
#Trying to figure out which groups of bacteria are associated with each group
, which bacteria associated with which group
#mixes up group randomly, compares randomly to actual, which is not like the
random
#only 5 spp that are associated with EP/non-EP group
#permutational test, larger = better?

#Indicspecies analysis by site (roost location)
#The frequency table file name is called species but it actually has site dat
a! Whoopsies.
Site = read.csv("Bat Feature Table_Species.csv", header=T)

#Polish bat species dataset
#Need to put bat species names to the proper sample, forgot to put this in th
e frequency table
Site = Site[,-92]
names(Site)=Site[2,]
Site=Site[-2,]
County=Site[1,]#stores this data
Site=Site[-1,]
row.names(Site)=Site[,1]
Site=Site[,-1]#removes 1st row #2 = EP, associated with EP bats! Just shows t
he sig. groups

for (i in 1:90) Site[,i]=as.numeric(Site[,i])#convert that one column
```

```
#Transpose dataset
Site=t(Site)

#Trim site dataset
#Convert site to character
County=as.character(County)#not a data frame
County=County[-1]

#Convert to factor, then convert to numeric
County=as.factor(County)
levels(County) #important so you know what group number corresponds to what c
ounty!
## [1] "Chelan"       "Klickitat"    "Lewis"        "Lincoln "     "Mason"
## [6] "Okanogan"     "Pend Oreille" "Spokane"      "Thurston"

County=as.numeric(County)
#1 = Chelan; 2 = Klickitat; 3 = Lewis; 4 = Lincoln; 5 = Mason; 6 = Okan. 7 =
PO; 8=Spokane; 9 = Thurston

#load in the indicspecies package
library(indicspecies)
indval_County=multipatt(Site,County)
summary(indval_County)



#Indicspecies analysis by bat species

#Polish frequency table with species info
#This freuqnecy table file actually ahs species data included
Species = read.csv("Bat_Species(Real).csv", header=T)
#compare sample IDs, which samples were dropped?
Bat = read.csv("Bat Feature Table_Excel.csv", header=T)

#Trim extra taxonomy column from data table
Bat = Bat[,-92]
Not_in=which(!Species$X.SampleID %in% Bat[2,])
Species=Species[-Not_in,]
names(Species)[1]="SampleID"

Bat=Bat[-1,]
row.names(Bat)=Bat[,1]
Bat=Bat[,-1]#removes 1st row

#Transpose dataset
Bat=t(Bat)
Bat=as.data.frame(Bat)
```

```
#Need to add in the proper bat species info with each sample in R since I for
got to put it in the frequency table
Bat_w_species=merge(Bat,Species,by="SampleID")



Species=Bat_w_species$Species
Bat_w_species=Bat_w_species[,-c(1,252)]

for (i in 1:250) Bat_w_species[,i]=as.numeric(Bat_w_species[,i])#convert that
one column

Species=as.factor(Species)
levels(Species)
## [1] "COTO"    "EPFU"    "MYLU"    "MYVO"    "MYYU"    "MYYU.LU"

Species=as.numeric(Species)


#Indicspecies analysis by bat species
indval_Species=multipatt(Bat_w_species,Species)
summary(indval_Species)



#Individual taxa barplots
#Just for species

#Load in ggplot library
library(ggplot2)
Species2=as.data.frame(Species)
Species2$Species=as.factor(Species2$Species)

#ggplot(Bat_w_species,aes_string(x=names(Bat_w_species)[1],y=names(Bat_w_spec
ies)[which(names(Bat_w_species)=="d__Bacteria;p__Bacteroidota;c__Bacteroidia;
o__Flavobacteriales;f__Crocinitomicaceae;g__Brumimicrobium")]))+geom_point()
boxplot(Bat_w_species[,which(names(Bat_w_species)=="d__Bacteria;p__Bacteroido
ta;c__Bacteroidia;o__Flavobacteriales;f__Crocinitomicaceae;g__Brumimicrobium"
)]~Species,ylab="Brumimicrobium")

#could change the column name for that species so easier to work with
#Polish bat species data
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Bacteroidota
;c__Bacteroidia;o__Flavobacteriales;f__Crocinitomicaceae;g__Brumimicrobium")]
="Brumimicrobium"

Species2$Species=as.character(Species2$Species)
Species2$Species[which(Species2$Species=="5")]="MYYU"
Species2$Species[which(Species2$Species=="4")]="MYVO"
```

```
Species2$Species[which(Species2$Species=="3")]="MYLU"
Species2$Species[which(Species2$Species=="2")]="EPFU"
Species2$Species[which(Species2$Species=="1")]="COTO"
Species2$Species[which(Species2$Species=="6")]="MYYU.LU"



#COTO Bacteria

#Check in frequency table with CTRL+F
#Klenkia spp.
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Actinobacter
iota;c__Actinobacteria;o__Frankiales;f__Geodermatophilaceae;g__Klenkia")]="Kl
enkia"

ggplot(Bat_w_species,aes(x=Species2$Species,y=Klenkia))+geom_boxplot()+theme_
classic()+labs(y=expression(paste("Frequency of ",italic("Klenkia"),"
spp.")),x="Bat Species")+ggtitle(expression(paste("Frequency of
",italic("Klenkia"),"  spp. Across Bat Species"))) +
  theme(plot.title = element_text(hjust = 0.5))


#Family Stappiaceae
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Proteobacter
ia;c__Alphaproteobacteria;o__Rhizobiales;f__Stappiaceae;g__Stappia")]="Stappi
aceae"

ggplot(Bat_w_species,aes(x=Species2$Species,y=Alcaligenaceae))+geom_boxplot()
+theme_classic()+labs(y="Frequency of Family Stappiaceae",x="Bat
Species")+ggtitle("Frequency of Family Stappiaceae Across Bat Species") +
  theme(plot.title = element_text(hjust = 0.5))

#Repeat for other abundant COTO bacteria from indicspecies analysis

#EPFU Bacteria
#Bergeyella spp.
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Bacteroidota
;c__Bacteroidia;o__Flavobacteriales;f__Weeksellaceae;g__Bergeyella")]="Bergey
ella"

ggplot(Bat_w_species,aes(x=Species2$Species,y=Bergeyella))+geom_boxplot()+the
me_classic()+labs(y=expression(paste("Frequency of ",italic("Bergeyella"),"
spp.")),x="Bat Species")+ggtitle(expression(paste("Frequency of
",italic("Bergeyella"),"  spp. Across Bat Species"))) +
  theme(plot.title = element_text(hjust = 0.5))

#Order d__Bacteria;p__Firmicutes;c__Bacilli;o__Bacillales
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Firmicutes;c
__Bacilli;o__Bacillales;__;__")]="Bacillales"
```

```
ggplot(Bat_w_species,aes(x=Species2$Species,y=Bacillales))+geom_boxplot()+the
me_classic()+labs(y="Frequency of Order Bacillales",x="Bat
Species")+ggtitle("Frequency of Order Bacillales Across Bat Species") +
  theme(plot.title = element_text(hjust = 0.5))

#Repeat for other abundant EPFU bacteria from indicspecies analysis

#MYYU

#d__Bacteria;p__Actinobacteriota;c__Actinobacteria;o__Micrococcales;f__Dermab
acteraceae;g__Brachybacterium
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Actinobacter
iota;c__Actinobacteria;o__Micrococcales;f__Dermabacteraceae;g__Brachybacteriu
m")]="Brachybacterium"

ggplot(Bat_w_species,aes(x=Species2$Species,y=Brachybacterium))+geom_boxplot(
)+theme_classic()+labs(y=expression(paste("Frequency of
",italic("Brachybacterium")," spp.")),x="Bat
Species")+ggtitle(expression(paste("Frequency of
",italic("Brachybacterium")," spp. Across Bat Species"))) +
  theme(plot.title = element_text(hjust = 0.5))

#d__Bacteria;p__Firmicutes;c__Bacilli;o__Staphylococcales;f__Staphylococcacea
e;__
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Firmicutes;c
__Bacilli;o__Staphylococcales;f__Staphylococcaceae;__")]="Staphylococcaceae"

ggplot(Bat_w_species,aes(x=Species2$Species,y=Staphylococcaceae))+geom_boxplo
t()+theme_classic()+labs(y="Frequency of Family Staphylococcaceae",x="Bat
Species")+ggtitle("Frequency of Family Staphylococcaceae Across Bat Species")
+
  theme(plot.title = element_text(hjust = 0.5))



#MYYU/LU

#d__Bacteria;p__Proteobacteria;c__Gammaproteobacteria;o__Enterobacterales;f__
Morganellaceae;g__Candidatus_Blochmannia
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Proteobacter
ia;c__Gammaproteobacteria;o__Enterobacterales;f__Morganellaceae;g__Candidatus
_Blochmannia")]="Candidatus_Blochmannia"

ggplot(Bat_w_species,aes(x=Species2$Species,y=Candidatus_Blochmannia))+geom_b
oxplot()+theme_classic()+labs(y=expression(paste("Frequency of
",italic("Candidatus_Blochmannia")," spp.")),x="Bat
Species")+ggtitle(expression(paste("Frequency of
",italic("Candidatus_Blochmannia")," spp. Across Bat Species"))) +
```

```
  theme(plot.title = element_text(hjust = 0.5))

#d__Bacteria;p__Verrucomicrobiota;c__Verrucomicrobiae;o__Verrucomicrobiales;f
__Verrucomicrobiaceae;g__uncultured
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Verrucomicro
biota;c__Verrucomicrobiae;o__Verrucomicrobiales;f__Verrucomicrobiaceae;g__unc
ultured")]="Verrucomicrobiaceae"

ggplot(Bat_w_species,aes(x=Species2$Species,y=Verrucomicrobiaceae))+geom_boxp
lot()+theme_classic()+labs(y="Frequency of Family Verrucomicrobiaceae",x="Bat
Species")+ggtitle("Frequency of Family Verrucomicrobiaceae Across Bat
Species") +
  theme(plot.title = element_text(hjust = 0.5))

#Repeat for other abundant MYYU/LU bacteria from indicspecies analysis


#Boxplots by ectoparasite presence

Bat = read.csv("Bat Feature Table_Excel_Site.csv", header=T)

#Trim extra taxonomy column from data table
Bat = Bat[,-92]

Bat=Bat[-1,]
row.names(Bat)=Bat[,1]
Bat=Bat[,-1]#removes 1st row
Bat=Bat[-1,]#removes 1st row


for (i in 1:90) Bat[,i]=as.numeric(Bat[,i])#convert that one column


#Transpose dataset
Bat=t(Bat)
Bat=as.data.frame(Bat)
Bat_w_species=Bat

#no ectoparasites
#d__Bacteria;p__Actinobacteriota;c__Actinobacteria;o__Micrococcales;f__Microc
occaceae;g__Micrococcus
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Actinobacter
iota;c__Actinobacteria;o__Micrococcales;f__Micrococcaceae;g__Micrococcus")]="
Micrococcus"

ggplot(Bat_w_species,aes(x=ectoparasite,y=Micrococcus))+geom_boxplot()+theme_
classic()+labs(y=expression(paste("Frequency of ",italic("Micrococcus"),"
spp.")),x="Ectoparasite Presence")+ggtitle(expression(paste("Frequency of
",italic("Micrococcus")," spp. by Ectoparasite Presence Status"))) +
```

```
  theme(plot.title = element_text(hjust = 0.5))


#ectoparasites
#d__Bacteria;p__Actinobacteriota;c__Actinobacteria;o__Propionibacteriales;f__
Nocardioidaceae;g__Nocardioides
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Actinobacter
iota;c__Actinobacteria;o__Propionibacteriales;f__Nocardioidaceae;g__Nocardioi
des")]="Nocardioides"

ggplot(Bat_w_species,aes(x=ectoparasite,y=Nocardioides))+geom_boxplot()+theme
_classic()+labs(y=expression(paste("Frequency of ",italic("Nocardioides"),"
spp.")),x="Ectoparasite Presence")+ggtitle(expression(paste("Frequency of
",italic("Nocardioides"),"  spp. by Ectoparasite Presence Status"))) +
  theme(plot.title = element_text(hjust = 0.5))

#Repeat for other abundant ectoparsite bacteria from indicspecies analysis


#By site
Bat_Site = read.csv("Bat Feature Table_Species.csv", header=T)

as.character(Bat_Site[1,2:91])

#Klickitat County
#d__Bacteria;p__Firmicutes;c__Clostridia;o__Clostridiales;f__Clostridiaceae;g
__Clostridium_sensu_stricto_1
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Firmicutes;c
__Clostridia;o__Clostridiales;f__Clostridiaceae;g__Clostridium_sensu_stricto_
1")]="Clostridium_sensu_stricto_1"

ggplot(Bat_w_species,aes(x=as.character(Bat_Site[1,2:91]),y=Clostridium_sensu
_stricto_1))+geom_boxplot()+theme_classic()+labs(y=expression(paste("Frequenc
y of ",italic("Clostridium_sensu_stricto_1"),"  spp.")),x="Roost
Location")+ggtitle(expression(paste("Frequency of
",italic("Clostridium_sensu_stricto_1"),"  spp. Across Roost Locations"))) +
  theme(plot.title = element_text(hjust = 0.5))

#Linoln County
#d__Bacteria;p__Actinobacteriota;c__Actinobacteria;__;__;__
#labs(y="Frequency of Family Staphylococcaceae",x="Bat
Species")+ggtitle("Frequency of Family Staphylococcaceae Across Bat Species")
names(Bat_w_species)[which(names(Bat_w_species)=="d__Bacteria;p__Actinobacter
iota;c__Actinobacteria;__;__;__")]="Actinobacteria"

ggplot(Bat_w_species,aes(x=as.character(Bat_Site[1,2:91]),y=Actinobacteria))+
geom_boxplot()+theme_classic()+labs("Frequency of Order
Actinobacteria",x="Roost Location",y="Frequency of Order
```

```
Actinobacteria")+ggtitle("Frequency of Order Actinobacteria Across Roost
Locations") +
  theme(plot.title = element_text(hjust = 0.5))

#Repeat for other abundant bacteria in different counties from indicspecies
analysis
```

4.) Creating ordination plots in R using our data output from QIIME2. This is a way to interface QIIME in R, which can be useful for future QIIME2 data analyses.

- A little tricky!
- We used the rarefied sequence table from QIIME2, in addition to the rooted-tree file and taxonomy file. We were able to use the straight .qza files from QIIME, no exporting necessary.
- All ordination plots are already set up to have the ellipses and centered titles.

```
#PCoA plots in R using microbiome data from QIIME2

#Run an NMDS, need vegan library
library(vegan)
library(ggord)

#Install qiime2R with the commented chunks below
#if (!requireNamespace("devtools", quietly =
TRUE)){install.packages("devtools")}
#devtools::install_github("jbisanz/qiime2R") # current version is 0.99.20

#Load in qiime2R package
library(qiime2R)
library (devtools)

#Read in your rarefied sequence table directly from QIIME2
SVs<-read_qza("Bat_2021_rarefied-table.qza")
names(SVs) #this was supposed to read "ASVs", not "SVs"
## [1] "uuid"       "type"       "format"     "contents"   "version"
## [6] "data"       "provenance"

#Load in gplot2 and readr packages
library(ggplot2)
library(readr)

#Read in your .tsv metadata file with read_tsv
metadata=read_tsv("Bat_Metadata_R.tsv")
taxonomy=read_qza("Bat_2021_taxonomy.qza")
head(taxonomy$data) #make sure everything looks OK like it does below
##                          Feature.ID
## 1 a4a5cc927e391a59011c7e017f949dfd
```

```
## 2 9518639833147ad82db09d616f535f2b
## 3 0f268443b2b455b7c1f9dc9874a4c05f
## 4 2bf439a3c30945a7ef125b94d6d95929
## 5 3d09cd886d2a8b01c6ac64d2da2ebbfa
## 6 40f73c7caeb05aaed60dcfce2a90b099
##
Taxon
## 1 d__Bacteria; p__Proteobacteria; c__Gammaproteobacteria;
o__Alteromonadales; f__Pseudoalteromonadaceae; g__Pseudoalteromonas
## 2                          d__Bacteria; p__Proteobacteria;
c__Gammaproteobacteria; o__Vibrionales; f__Vibrionaceae; g__Vibrio

#use qza_to_phloseq to load in more files that you made with or for QIIME
physeq<-qza_to_phyloseq(
  features="Bat_2021_rarefied-table.qza", #rarefied table
  tree="Bat_2021_rooted-tree.qza", #rooted tree
  "Bat_2021_taxonomy.qza", #taxonomy file
  metadata = "Header_No_slashes_With_Spp_Edited_Updated_Bat_2021_Mapping_File
- Sheet1.tsv" #metadata file
)
physeq
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 692 taxa and 90 samples ]
## sample_data() Sample Data:       [ 90 samples by 23 sample variables ]
## tax_table()   Taxonomy Table:    [ 692 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 692 tips and 691 internal nodes ]
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 759 taxa and 34 samples ]
## sample_data() Sample Data:       [ 34 samples by 10 sample variables ]
## tax_table()   Taxonomy Table:    [ 759 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 759 tips and 757 internal nodes ]

#Load in phloseq library
library(phyloseq)

#Run ordination, specify what type of UniFrac distance you want
#We used weighted UniFrac because it's the most comprehensive
physeq.ord.wuni <- ordinate(physeq, "PCoA", "unifrac", weighted=T)

#Weighted unifrac PCoA plot for ectoparasite presence
#Ellipse is 95% CI
b.div.wuni <- plot_ordination(physeq, physeq.ord.wuni, type= "samples",
color= "Ectoparasites") + geom_point(size=3)
b.div.wuni <- b.div.wuni + stat_ellipse() + ggtitle("Weighted UniFrac
Distances by Ectoparasite Presence")  + theme_classic() +
scale_color_brewer("Ectoparasites", palette = "Set1")+theme(plot.title =
element_text(hjust = 0.5))
print(b.div.wuni)
```

```
#Ordination by bat species
b.div.wuni <- plot_ordination(physeq, physeq.ord.wuni, type= "samples",
color= "Species") + geom_point(size=3)
b.div.wuni <- b.div.wuni + stat_ellipse() + ggtitle("Weighted UniFrac
Distances by Bat Species")  + theme_classic() + scale_color_brewer("Species",
palette = "Set1")+theme(plot.title = element_text(hjust = 0.5))
print(b.div.wuni)



#Ordination by roost location
#If you have more samples than colors in a color-scheme, use a different set,
check out this website for different color palettes: https://r-graph-
gallery.com/38-rcolorbrewers-palettes.html
b.div.wuni <- plot_ordination(physeq, physeq.ord.wuni, type= "samples",
color= "Site") + geom_point(size=3)
b.div.wuni <- b.div.wuni + stat_ellipse() + ggtitle("Weighted UniFrac
Distances by Roost Location")  + theme_classic() + scale_color_brewer("Site",
palette = "Set1")+theme(plot.title = element_text(hjust = 0.5))
print(b.div.wuni)
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```