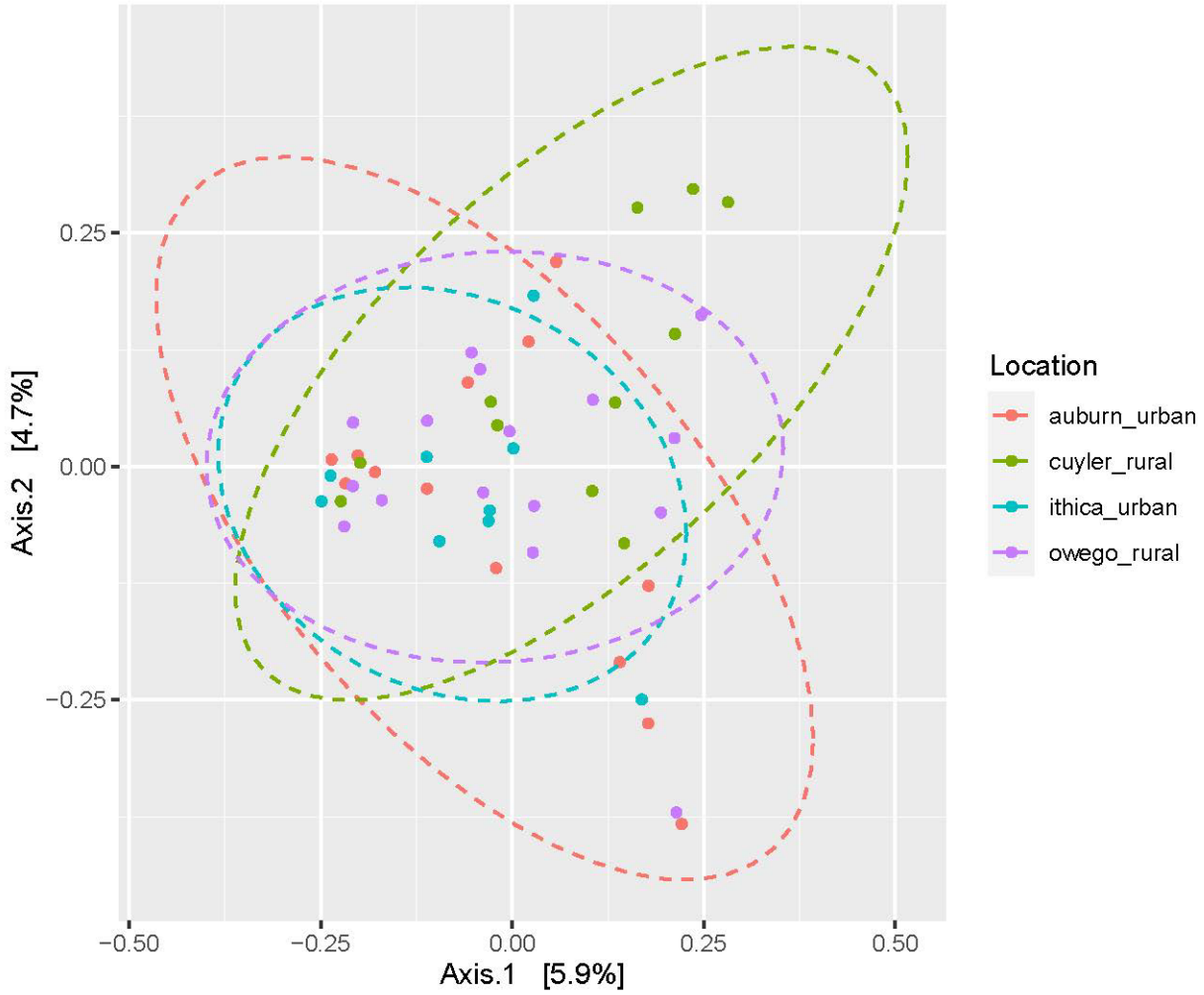
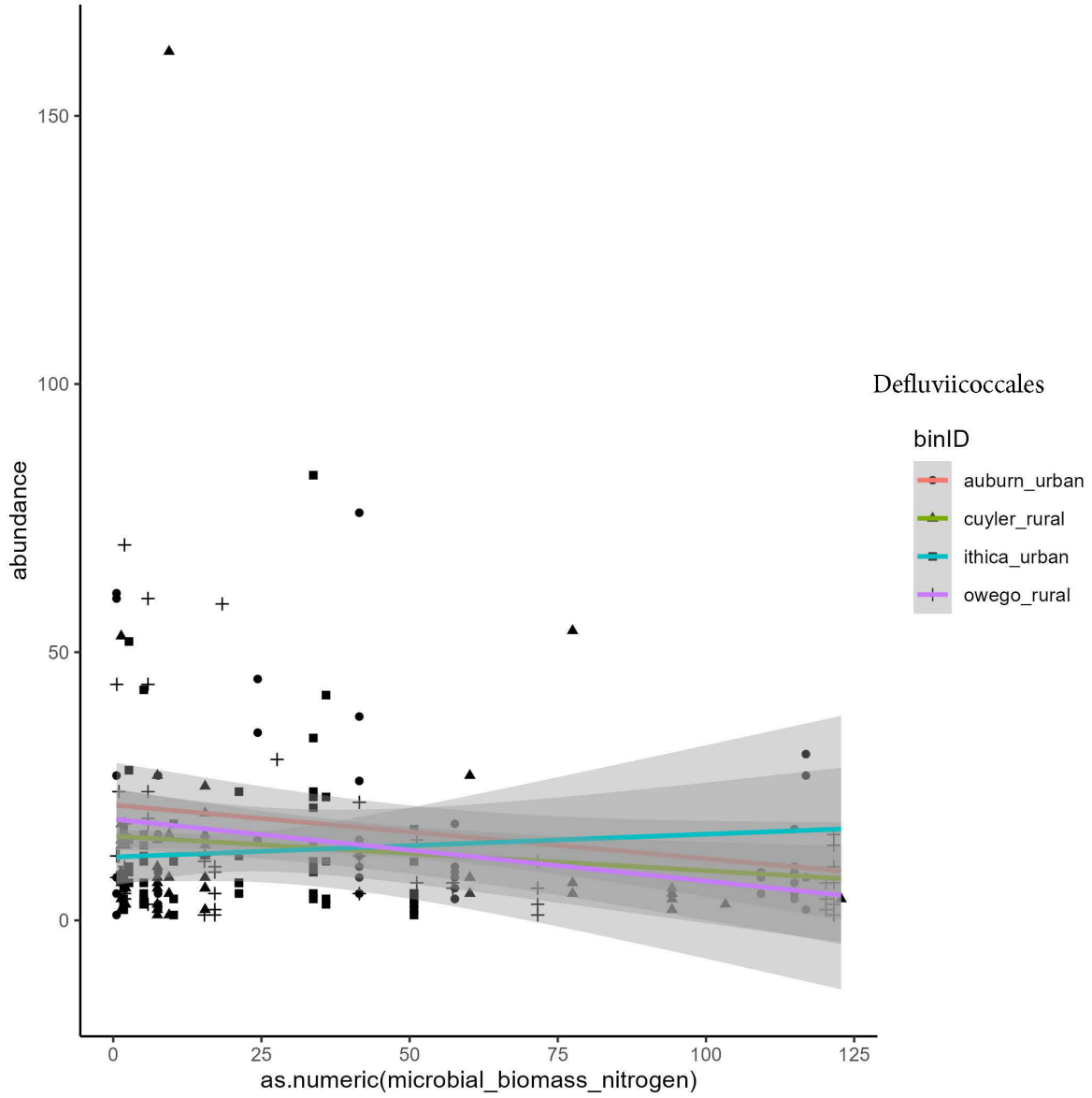
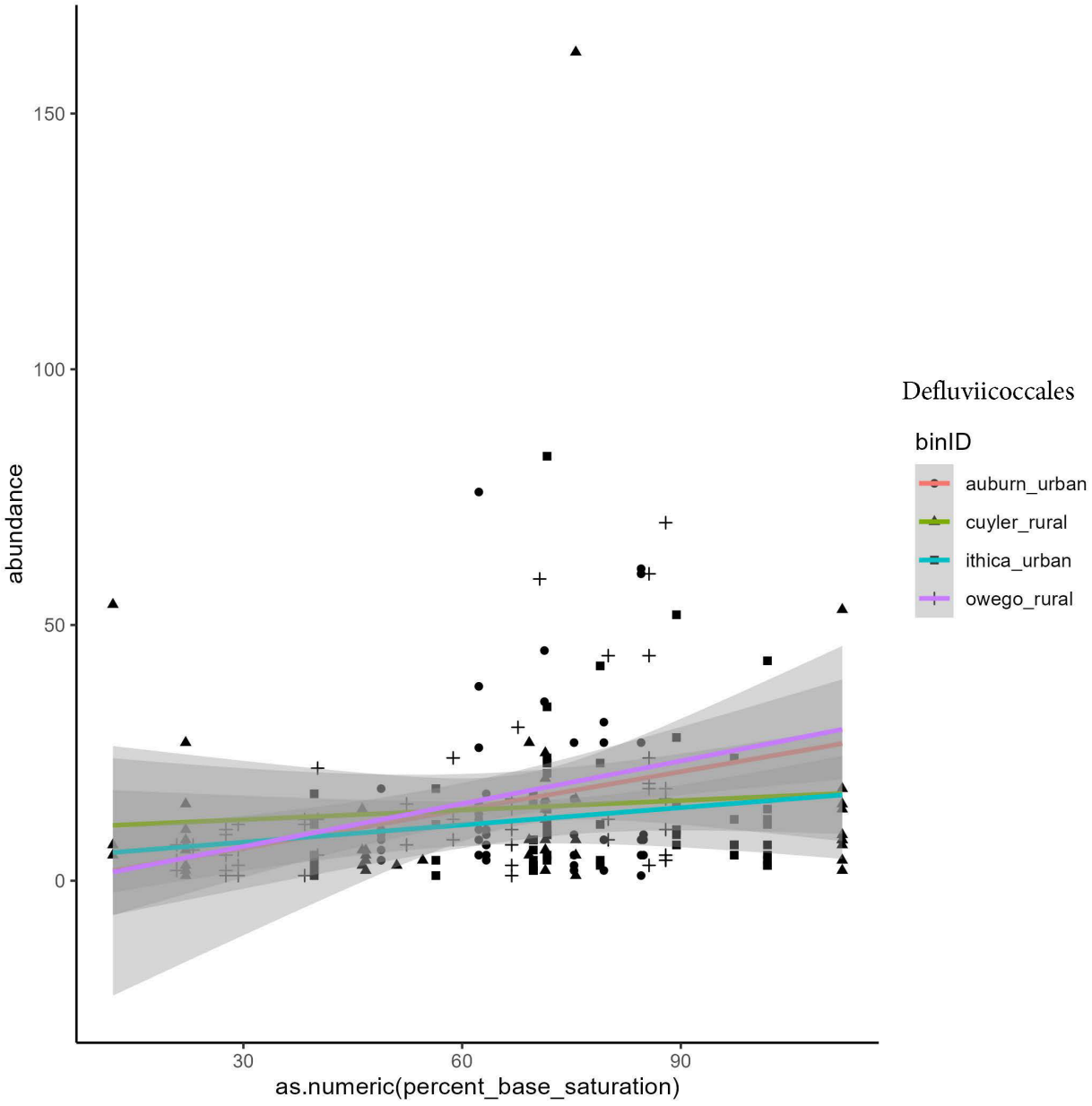
SUPPLEMENTAL

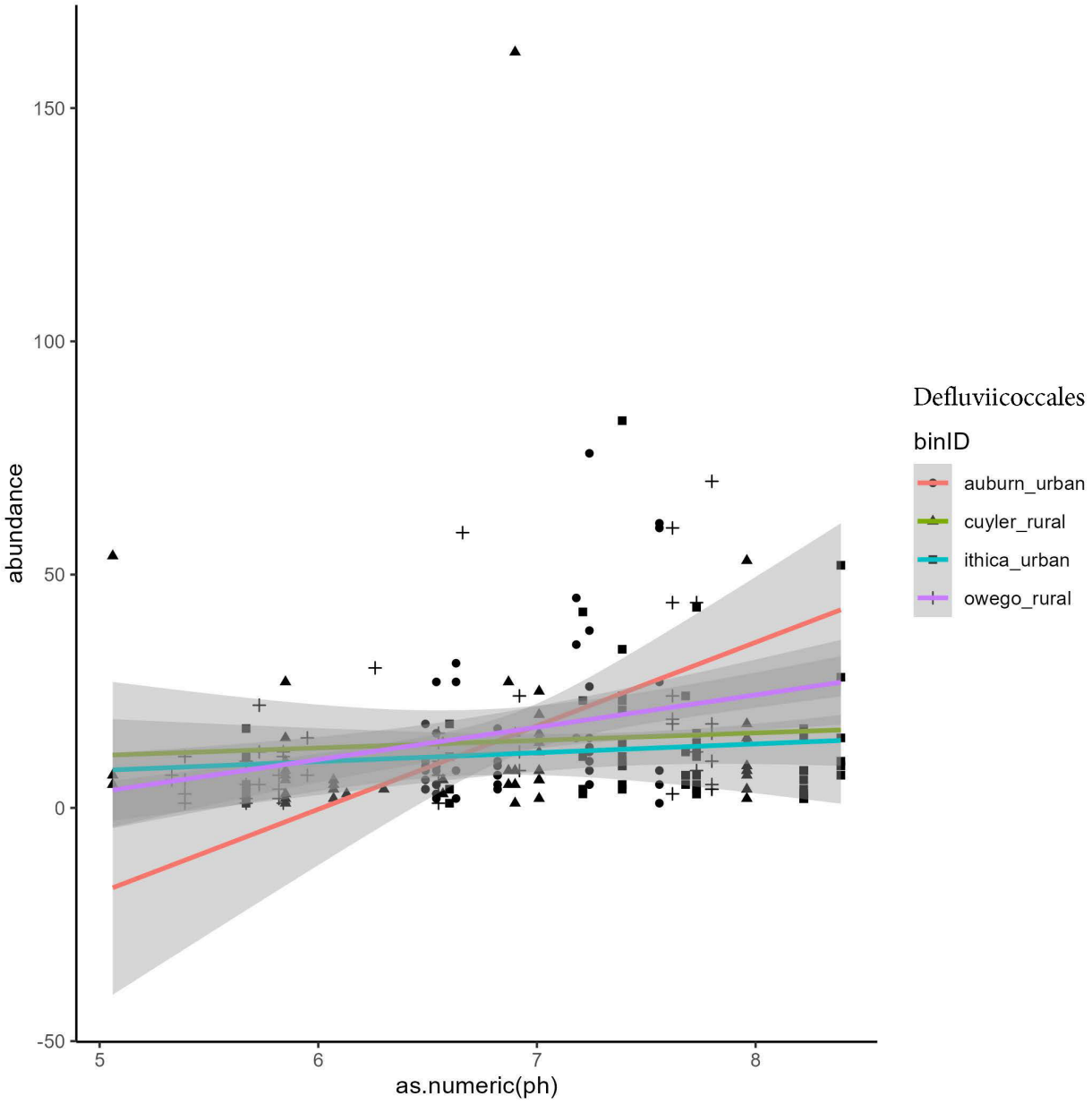
Supplemental 1: PCoA plot exhibiting insignificant beta diversity results between rural and urban wetland samples. Unifrac, Bray-Curtis, and Jaccard diversity statistics were used to compare samples based on binID. Adonis2 was then used to measure the statistical significance of the results comparing Auburn and Cuyler, and Ithaca and Owego. As shown, no clustering was observed between both rural sites (Owego and Cuyler) and urban sites (Ithaca and Auburn).

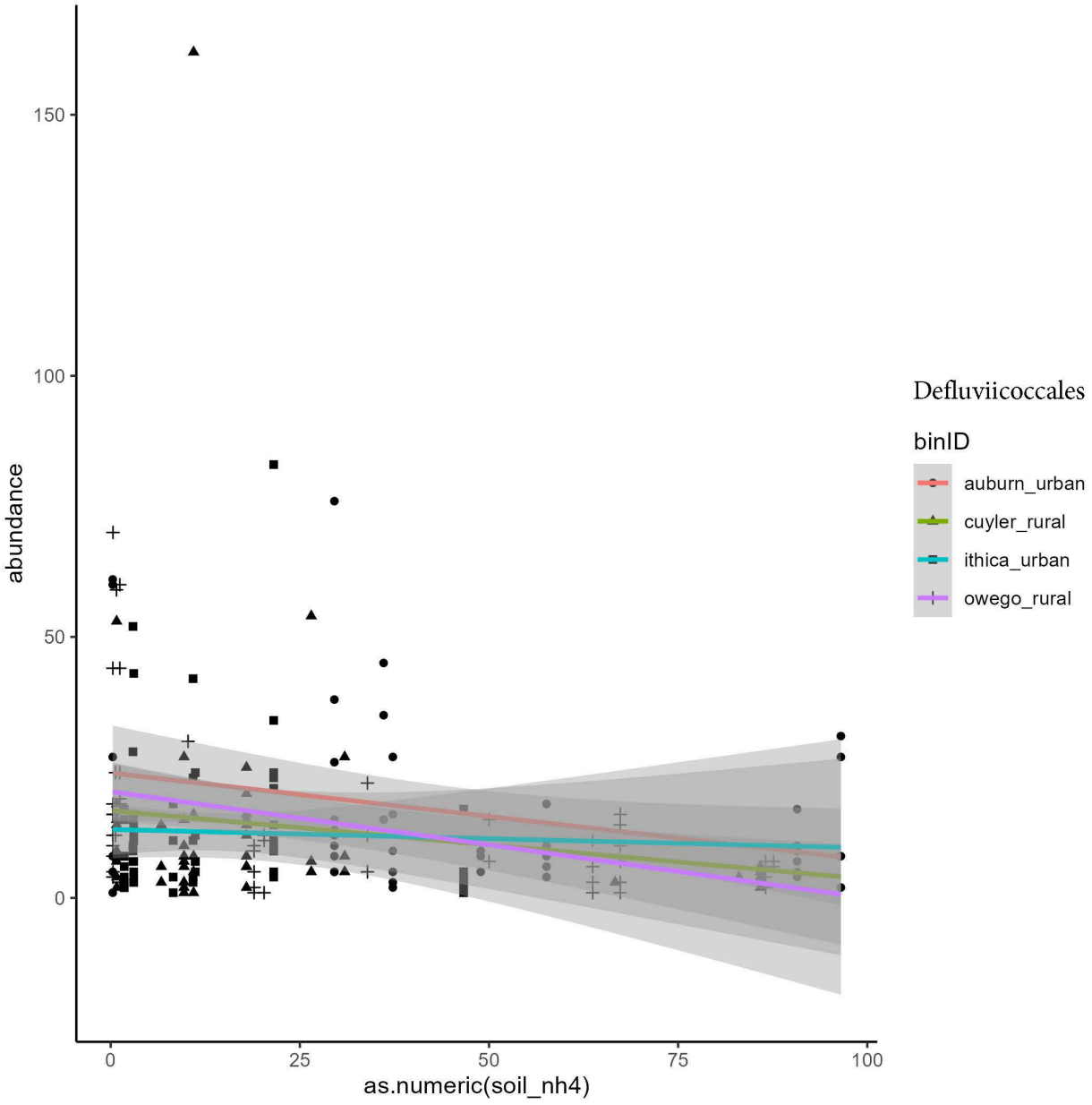


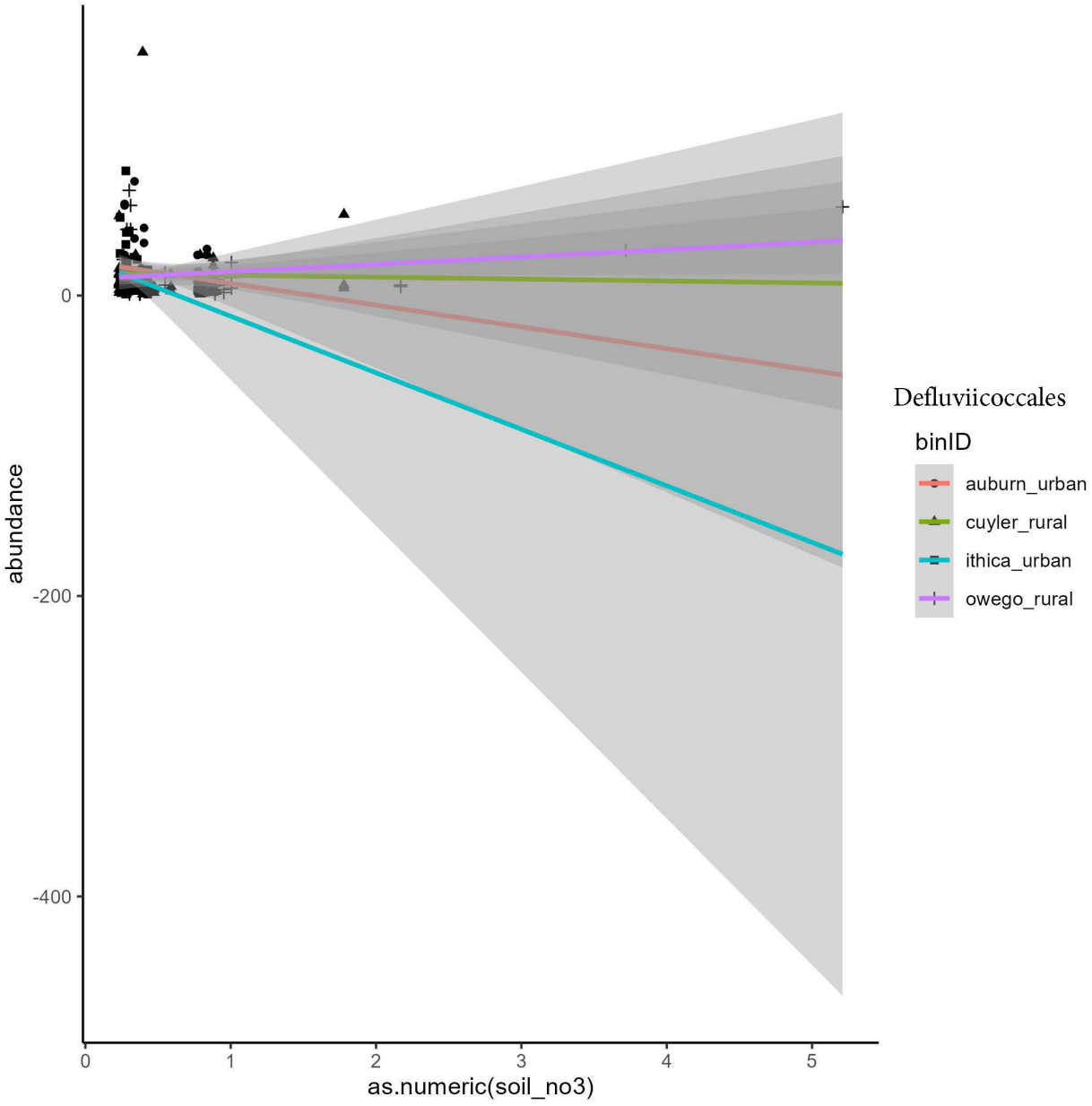
Supplemental 2: Correlation curves between environmental variables measured at binning locations and bacterial Orders. All Orders herein have significant correlations with cell abundance and the specified environmental variables (see results).

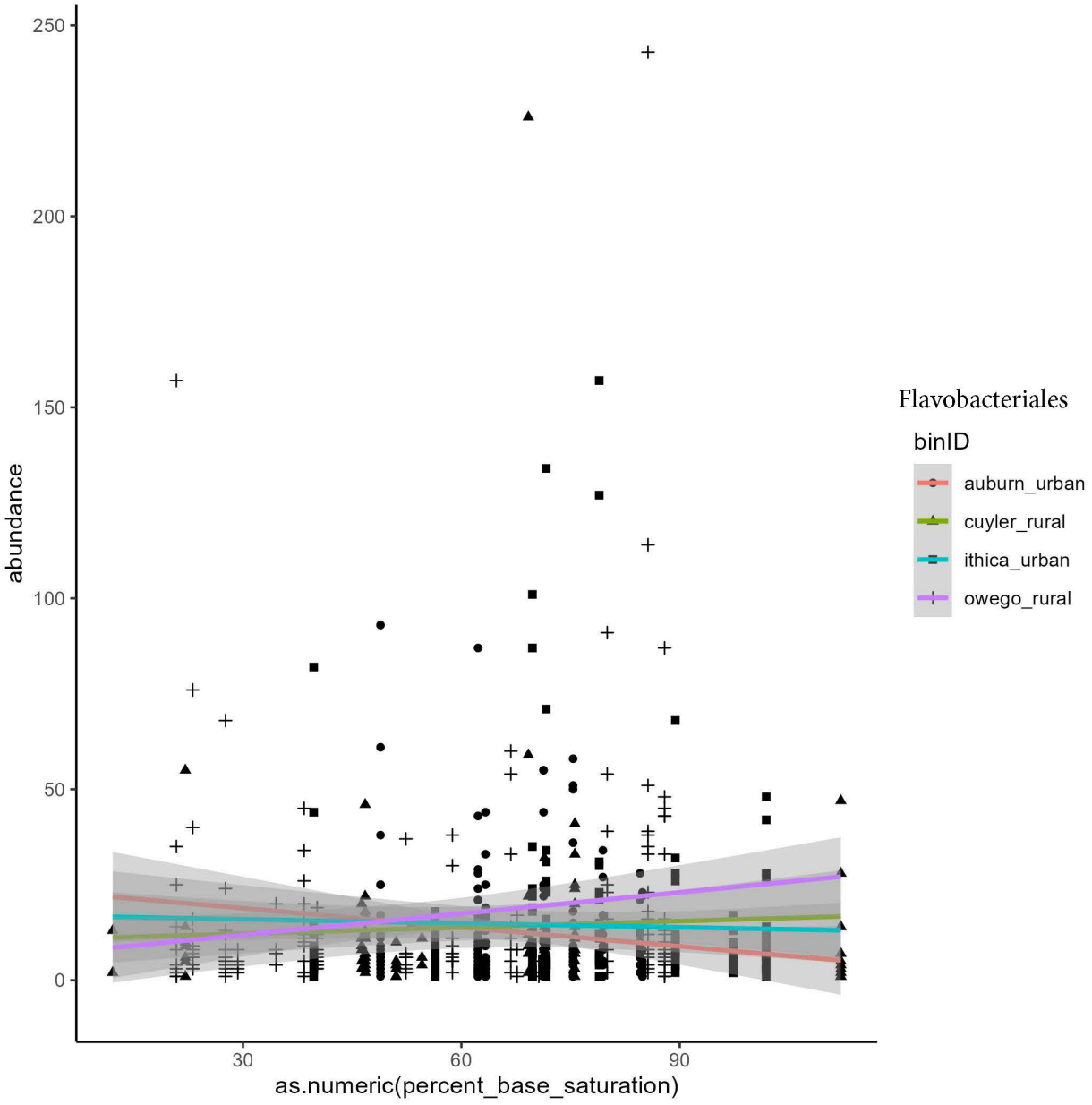


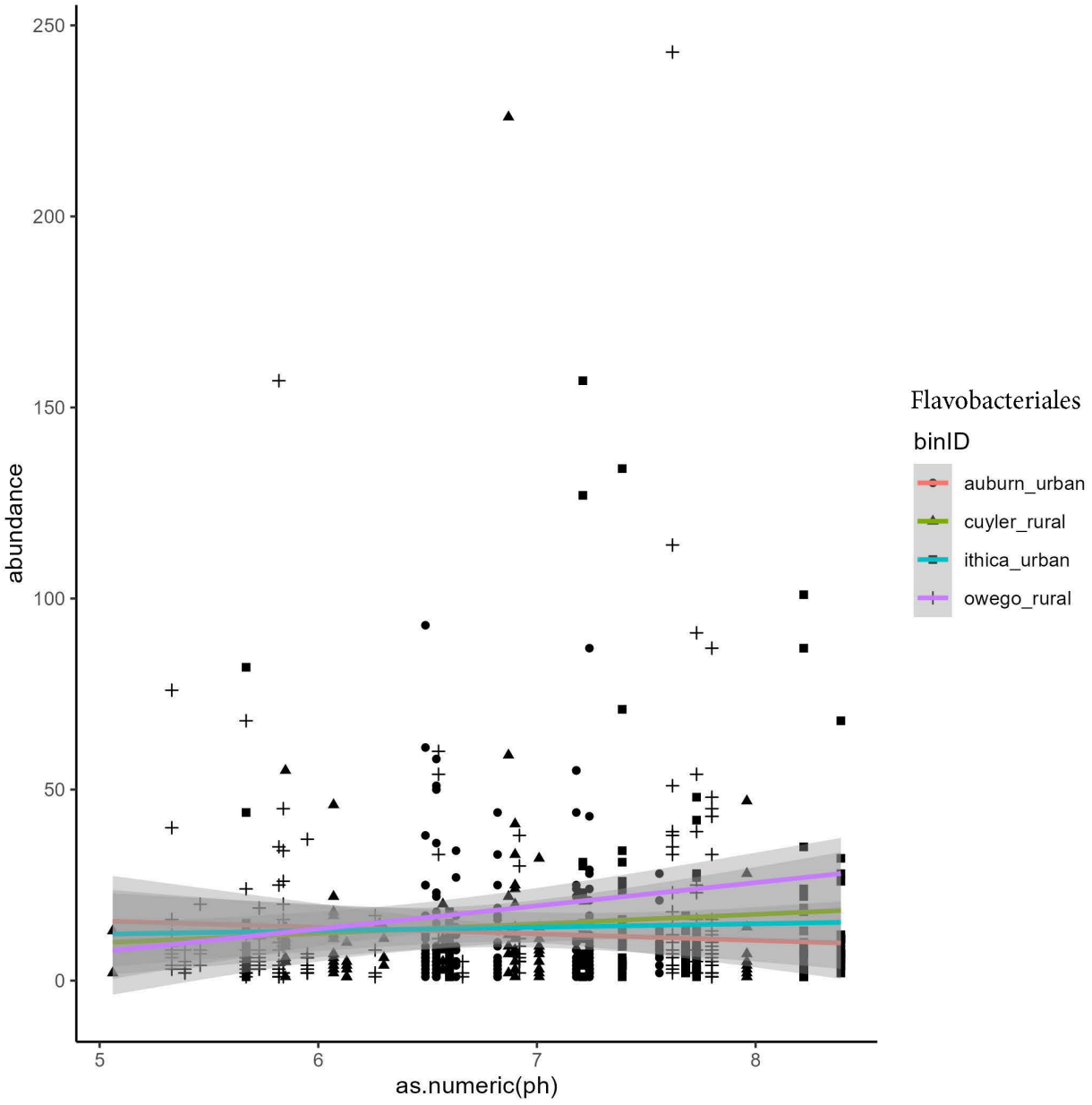

150
100
50
0
abundance
30
60
90
as.numeric(percent_base_saturation)
Defluviicoccales
binID
auburn_urban
cuyler_rural
ithica_urban
owego_rural

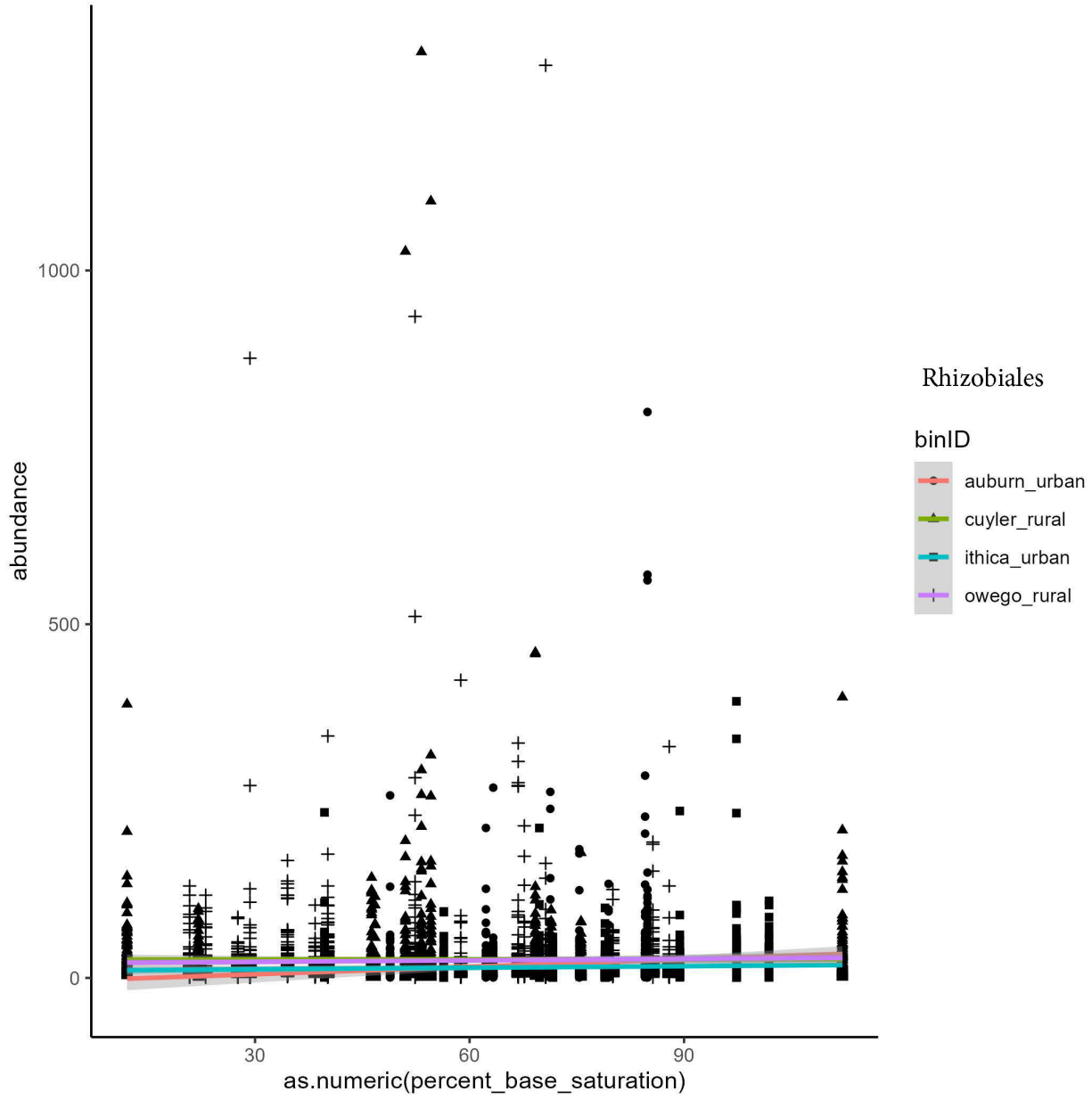

150
100
50
0
-50
abundance
5
6
7
8
as.numeric(ph)
Defluviicoccales
binID
auburn_urban
cuyler_rural
ithica_urban
owego_rural

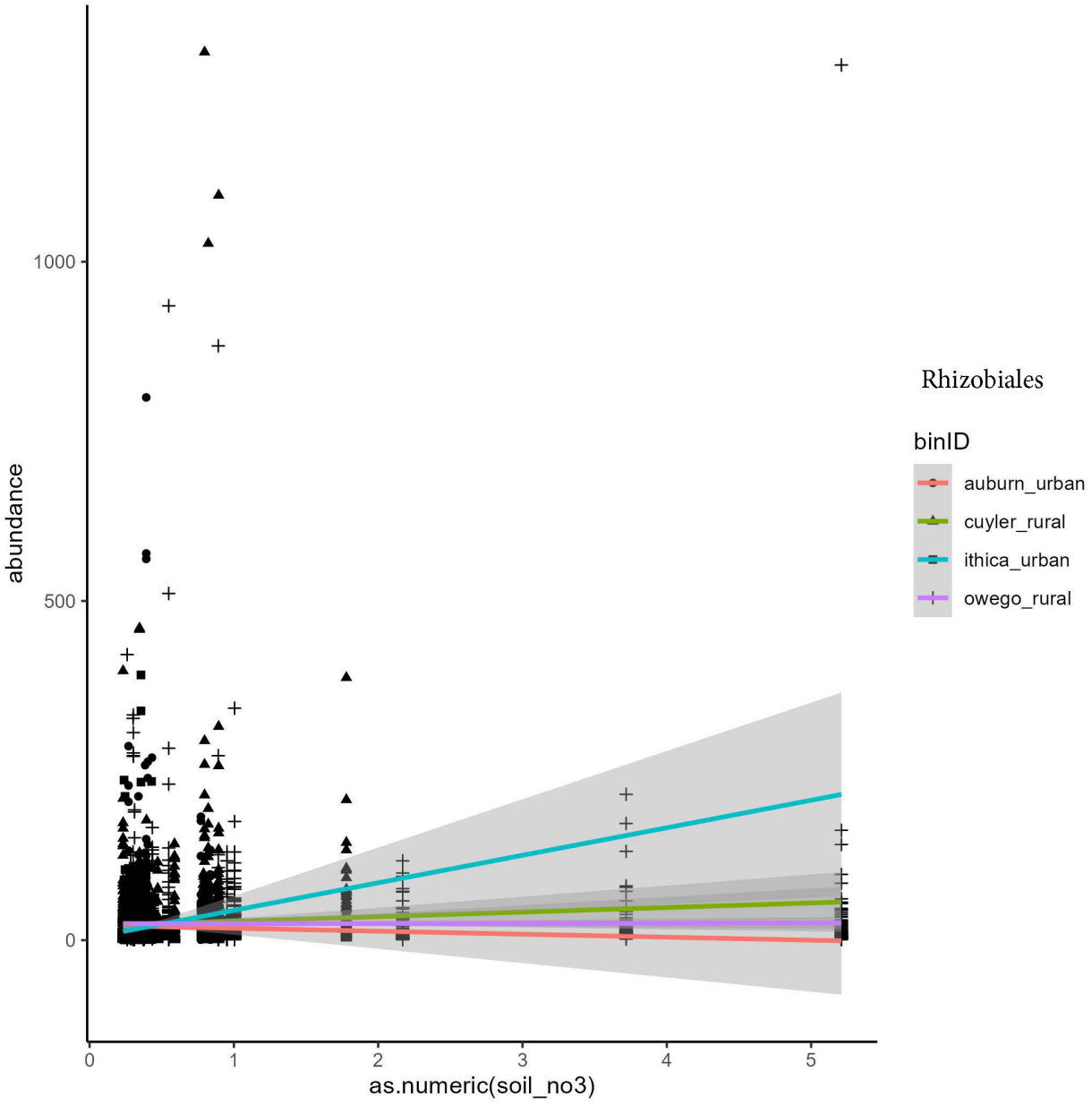

Defluviicoccales
binID
auburn_urban
cuyler_rural
ithica_urban
owego_rural
abundance
150
100
50
0
0
25
50
75
100
as.numeric(soil_nh4)

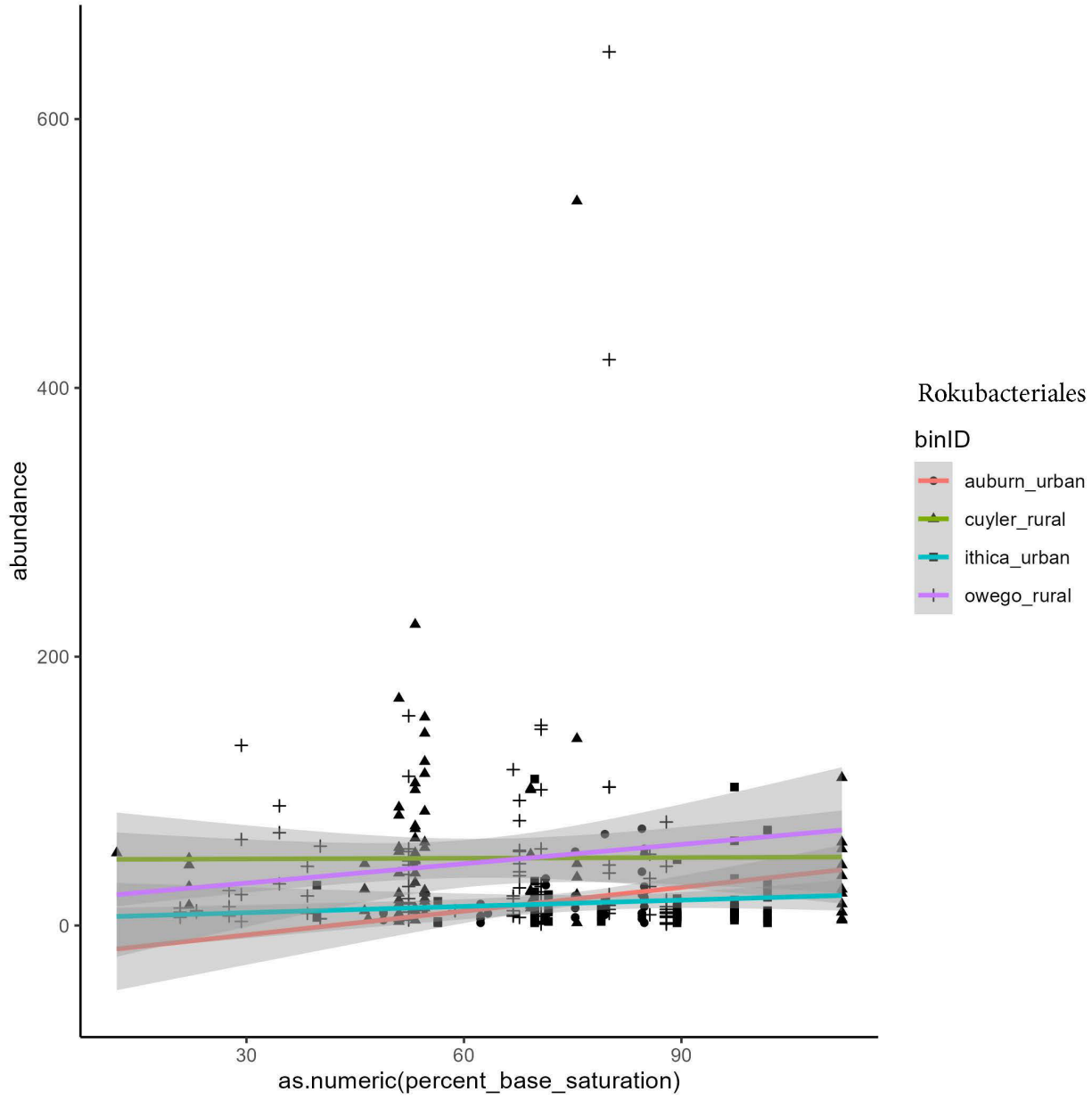

abundance
0
-200
-400
0
1
2
3
4
5
as.numeric(soil_no3)
Defluviicoccales
binID
auburn_urban
cuyler_rural
ithica_urban
owego_rural

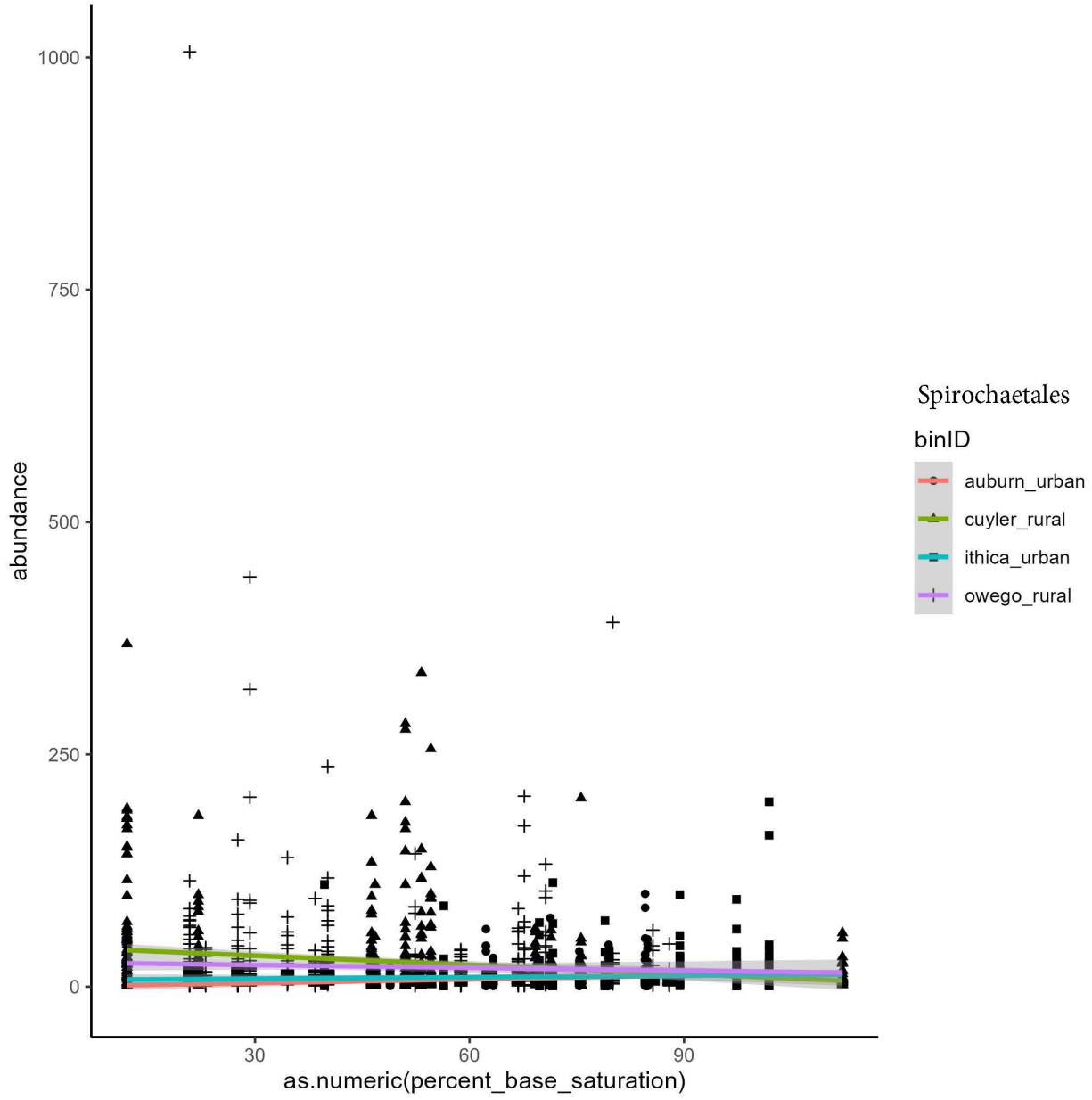

Flavobacteriales
binID
auburn_urban
cuyler_rural
ithica_urban
owego_rural
abundance
250
200
150
100
50
0
30
60
90
as.numeric(percent_base_saturation)

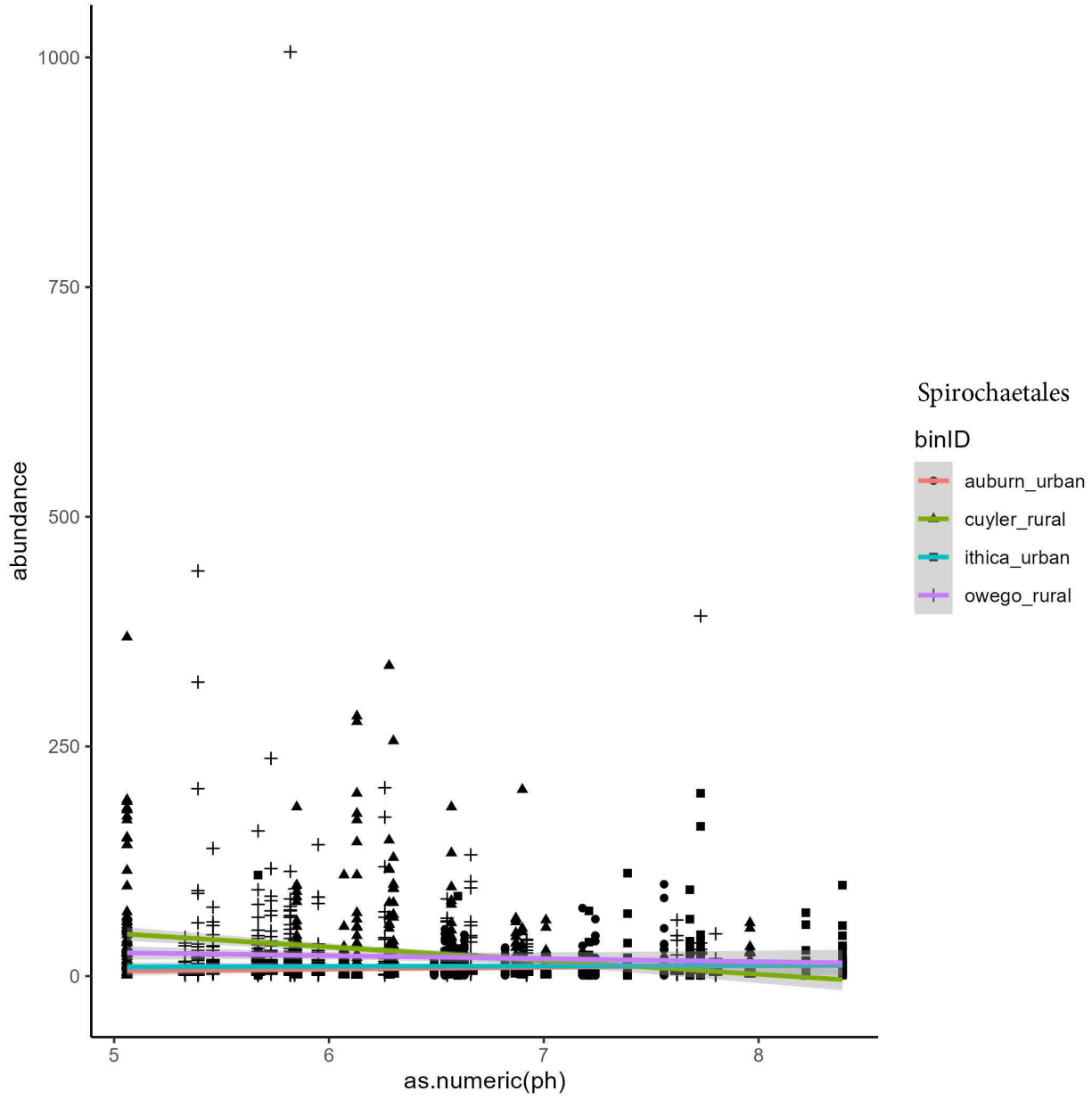

250
200
150
100
50
0
abundance
5
6
7
8
as.numeric(ph)
Flavobacteriales
binID
auburn_urban
cuyler_rural
ithica_urban
owego_rural

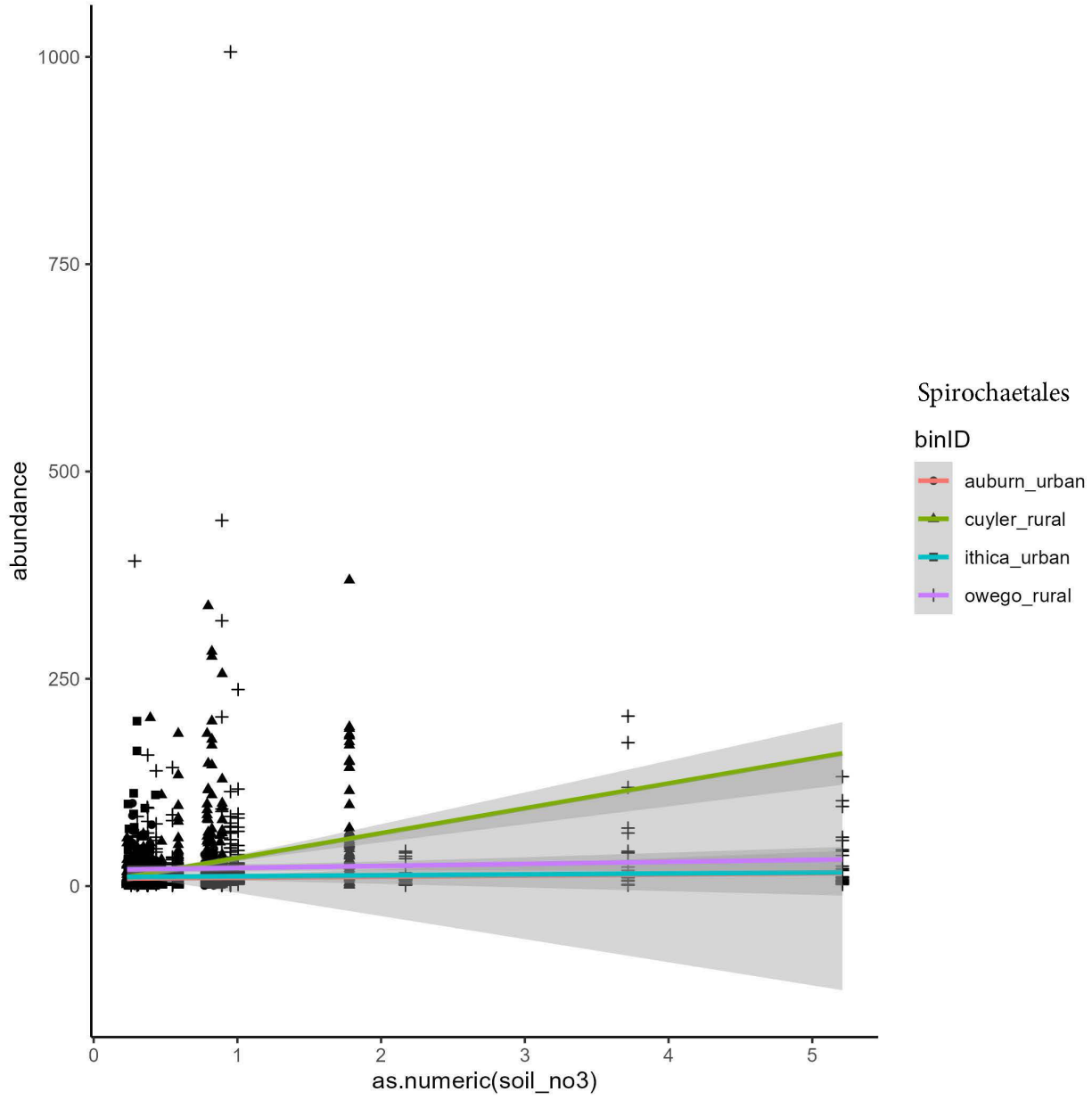

Rhizobiales
binID
auburn_urban
cuyler_rural
ithica_urban
owego_rural
abundance
1000
500
0
30
60
90
as.numeric(percent_base_saturation)


Rhizobiales
binID
auburn_urban
cuyler_rural
ithica_urban
owego_rural
abundance
1000
500
0
0
1
2
3
4
5
as.numeric(soil_no3)


Rokubacteriales
binID
auburn_urban
cuyler_rural
ithica_urban
owego_rural
abundance
600
400
200
0
30
60
90
as.numeric(percent_base_saturation)


Spirochaetales
binID
auburn_urban
cuyler_rural
ithica_urban
owego_rural
abundance
1000
750
500
250
0
30
60
90
as.numeric(percent_base_saturation)


1000
750
500
250
0
abundance
5
6
7
8
as.numeric(ph)
Spirochaetales
binID
auburn_urban
cuyler_rural
ithica_urban
owego_rural



Supplemental 3: LINUX script used to import dataset.

```
#!/bin/bash/tanyabalaji/qiime2-analaysis

#Importing data within working directory (using manifest file)

(1) qiime tools import \           #calls on the QIIME2 tool called
"import"
       --type "SampleData[SequencesWithQuality]" \ #specifies what type of
data we have (no need to change this line)
       --input-format SingleEndFastqManifestPhred33V2 \ #helping locate
all relevant files
```

```
       --input-path /data/Final-Project/wetlands_manifest.txt \
#dictate absolute pathway to the manifest file!
       --output-path /data/Final-Project/demux_wetlandseqs.qza

#Visualization of demultiplexed files + Code for Moving File to Local
Laptop

(2) qiime demux summarize --i-data demux_wetlandseqs.qza --o-visualization
demux_wetlandseqs.qzv

#Denoising/Clustering

(3) qiime dada2 denoise-single \ #calling on the DADA2 tool to Denoise
single end sequences
       --i-demultiplexed-seqs demux_wetlandseqs.qza \ #input file is your
demultiplexed file
       --p-trim-left 0 \  #your trimming parameters, how many you want to
trim on left side
               --p-trunc-len 125 \ #truncate it to 125 bps on the right
side
       --o-representative-sequences rep-seqs.qza \  #reference sequences
       --o-table table.qza \ #what it will output
       --o-denoising-stats stats.qza #generates 3 output files, the stats
file is irrelevant

#Data Visualization

#visualizing DADA2 stats
(4) qiime metadata tabulate \
       --m-input-file wetlandstats.qza \
       --o-visualization wetlandstats.qzv

# Visualize ASVs stats
(5) qiime feature-table summarize \
       --i-table wetlandtable.qza \
       --o-visualization wetlandtable.qzv \
       --m-sample-metadata-file /data/Final-Project/wetlands_metadata.txt

(6) qiime feature-table tabulate-seqs \
       --i-data wetlandrep-seqs.qza \
       --o-visualization wetlandrep-seqs.qzv

#Extracting your amplicon of interest from the reference database

(7) qiime feature-classifier extract-reads
       --i-sequences /mnt/datasets/silva_ref_files/silva-138-99-seqs.qza
       --p-f-primer CACGGTCGKCGGCGCCATT
       --p-r-primer GGACTACHVGGGTWTCTAAT
       --p-trunc-len 125
       --o-reads ref-seqs-trimmed.qza

#Training classifier with your new ref-seq file

(8) qiime feature-classifier fit-classifier-naive-bayes
```

```
       --i-reference-reads ref-seqs-trimmed.qza
       --i-reference-taxonomy /mnt/datasets/silva_ref_files/silva-138-99-
tax.qza
       --o-classifier classifier.qza

#Using the trained classifier to assign taxonomy to our reads

(9) qiime feature-classifier classify-sklearn
       --i-classifier classifier.qza
       --i-reads wetlandrep-seqs.qza
       --o-classification wetlandtaxonomy.qza

#Filtering our Data

#filtering – only included taxonomy-based filtering was conducted here
#removing mitochondria and chloroplast sequences from the data
(10) qiime taxa filter-table
       --i-table wetlandtable.qza
       --i-taxonomy wetlandtaxonomy.qza
       --p-exclude mitochondria,chloroplast
       --o-filtered-table wetlandtable-no-mitochondria-no-chloroplast.qza

#visualizing the table with filtered data
(11) qiime feature-table summarize
       --i-table wetlandtable-no-mitochondria-no-chloroplast.qza
       --o-visualization wetlandtable-no-mitochondria-no-chloroplast.qzv
       --m-sample-metadata-file /data/Final-Project/wetlands_metadata.txt

#Generating the phylogenetic tree

(12) qiime phylogeny align-to-tree-mafft-fasttree
       --i-sequences wetlandrep-seqs.qza
       --o-alignment aligned-rep-seqs.qza
       --o-masked-alignment masked-aligned-rep-seqs.qza
       --o-tree unrooted-tree.qza
       --o-rooted-tree rooted-tree.qza
```

Supplemental 4: Rstudio script used for data processing and analysis.

Group 16 Wetlands Script

library(tidyverse)

library(mapview)

library(readxl)

library(rfishnet2)

library(ggplot2)

library(phyloseq)

```
library(ape)
library(vegan)
library(DESeq2)

setwd("~/Desktop/Final-Project/phyloseq_obj")

###### Data Processing ######

#importing raw .xlx file
#filtering unusable metadata out

wetlands_metadata <- read_excel("wetlands_metadata.xlsx")
View(wetlands_metadata)

#quick mapping of sample points
mapview(wetlands_metadata, xcol = 'longitude', ycol = 'latitude', crs = 4269, grid = FALSE)

#filtering data
wetland_data_keep <- filter(wetlands_metadata, wetlands_metadata$ch4_flux != "Missing: Not provided")

#mapping the filtered data
mapview(wetland_data_keep, xcol = 'longitude', ycol = 'latitude', crs = 4269, grid = FALSE)

#exporting data to keep
write.csv(wetland_data_keep, file = 'wetlands_data2use.csv')

#binning based on proximity to pop densities
```

```
data <- data.frame(Longitude = wetland_data_keep$longitude,
                   Latitude = wetland_data_keep$latitude,
                   sample_ID = wetland_data_keep$SampleID,
                   binID = wetland_data_keep$binID)

centers = data.frame(longitude = c(-76.1474,-76.5661,-76.5018),
                     latitude = c(43.0481,42.9317,42.4439),
                     City = c("Syracuse", "Auburn", "Ithica"))

#filter for points close to auburn
Auburn_urban <- spatial_search(data, lat = 42.9327,lon = -76.5661, 25)

#filter for points close to Ithica
Ithica_urban <- spatial_search(data, lat = 42.4439,lon = -76.5018, 25)

#filtering for rural points

Owego_rural <- spatial_search(data, lat = 42.09415,lon = -76.26031, 30)
Cuyler_rural <- spatial_search(data, lat = 42.73592,lon = -75.83425, 30)

#mapping points to check
mapview(Cuyler_rural, xcol = 'Longitude', ycol = 'Latitude', crs = 4269, grid = FALSE)

#adding binning ID to df
wetland_data_keep$binID <- 'N/A'

auburn_urban_list <- c(Auburn_urban$sample_ID)
ithica_urban_list <- c(Ithica_urban$sample_ID)
```

```
owego_rural_list <- c(Owego_rural$sample_ID)
cuyler_rural_list <- c(Cuyler_rural$sample_ID)

for (i in 1: nrow(wetland_data_keep)){
  if (wetland_data_keep$SampleID[i] %in% auburn_urban_list){
    wetland_data_keep$binID[i] <- 'auburn_urban'
  }
  if (wetland_data_keep$SampleID[i] %in% ithica_urban_list){
    wetland_data_keep$binID[i] <- 'ithica_urban'
  }
  if (wetland_data_keep$SampleID[i] %in% owego_rural_list){
    wetland_data_keep$binID[i] <- 'owego_rural'
  }
  if (wetland_data_keep$SampleID[i] %in% cuyler_rural_list){
    wetland_data_keep$binID[i] <- 'cuyler_rural'
  }
}

#filter data with binIDs
wetland_data_binIDs <- filter(wetland_data_keep, wetland_data_keep$binID != "N/A")



###### Alpha/Beta Diversity Metrics ######

#use the full metadata file with binIDs
meta <- wetland_data_binIDs

otufp <- "feature-table.txt"
```

```
otu <- read_delim(file = otufp, delim="\t", skip=1)

taxfp <- "taxonomy.tsv"
tax <- read_delim(taxfp, delim="\t")

phylotreefp <- "data/tree.nwk"
phylotree <- read.tree(phylotreefp)

#converting all data to phyloseq objects
#Format OTU table

# save everything except first column (OTU ID) into a matrix
otu_mat <- as.matrix(otu[,-1])

# Make first column (#OTU ID) the rownames of the new matrix
rownames(otu_mat) <- otu$`#OTU ID`

# Use the "otu_table" function to make an OTU table
OTU <- otu_table(otu_mat, taxa_are_rows = TRUE)
class(OTU)

#Format sample metadata

# Save everything except sampleid as new data frame
#the second line is random ID#s that we can get rid of
samp_df <- as.data.frame(meta[,-2])

# Make sampleids the rownames
```

```
rownames(samp_df)<- meta$SampleID

# Make phyloseq sample data with sample_data() function
SAMP <- sample_data(samp_df)
class(SAMP)

#Formatting taxonomy

# Convert taxon strings to a table with separate taxa rank columns
tax_mat <- tax %>% select(-Confidence) %>%
  separate(col=Taxon, sep="; "
           , into = c("Domain","Phylum","Class","Order","Family","Genus","Species")) %>%
  as.matrix() # Saving as a matrix

# Save everything except feature IDs
tax_mat <- tax_mat[,-1]

# Make sampleids the rownames
rownames(tax_mat) <- tax$`Feature ID`

# Make taxa table
TAX <- tax_table(tax_mat)
class(TAX)

#Create phyloseq object

# Merge all into a phyloseq object
wetlands_phyloseq <- phyloseq(OTU, SAMP, TAX, phylotree)
```

```
##filtering

wetlands_filt <- subset_taxa(wetlands_phyloseq,  Domain == "d__Bacteria" & Class!="c__Chloroplast" & Family !="f__Mitochondria")

wetlands_filt_nolow <- filter_taxa(wetlands_filt, function(x) sum(x)>2, prune = TRUE)


#wetlands_filt_nolow_samps <- prune_samples(sample_sums(wetlands_filt_nolow)>10000, wetlands_filt_nolow)

wetlands_final <- wetlands_filt_nolow_samps


#Dont need this step unless we want to filter more

#wetlands_final <- subset_samples(wetlands_filt_nolow_samps, !is.na(ph) )


#forcing the rarefaction curve

rarecurve(t(as.data.frame(otu_table(wetlands_final))), cex = 0.1)


#used rarefaction curve to decide on depth 25000

wetlands_rare <- rarefy_even_depth(wetlands_final, rngseed = 1, sample.size = 25000)


#plotting alpha diversity


plot_richness(wetlands_rare, measures = c("Shannon","Chao1"))

# Add ggplot layers if desired to adjust visuals

plot_richness(wetlands_rare, x = "binID", measures = c("Shannon","Chao1")) +

  xlab("sample location") +

  geom_boxplot()


alpha = estimate_richness(wetlands_phyloseq)
```

```
meta <- sample_data(wetlands_phyloseq) #extracting meta from phyloseq obj
df <-cbind(alpha,meta)

#alpha diversity stats using kruskal wallis test

kruskal.test(Shannon ~ binID, data = df)
kruskal.test(Chao1 ~ binID, data = df)
kruskal.test(ACE ~ binID, data = df)
kruskal.test(Simpson ~ binID, data = df)
kruskal.test(InvSimpson ~ binID, data = df)
kruskal.test(Fisher ~ binID, data = df)

#beta diversity stats

#bray = looks at abundance and presence
clr_dist_matrix <- phyloseq::distance(wetlands_phyloseq, method = "bray") #building the distance matrix based on bray curtis

#adonis is stat tool to determine if matrix is significant; requires matrix generated above
adonis2(clr_dist_matrix ~ binID, data=data.frame(meta), permutations= 10000) #running adonis2 with 10,000 permutations

#jaccard only looks at presence and absence
clr_dist_matrix <- phyloseq::distance(wetlands_phyloseq, method = "jaccard") #building distance matrix based on jaccard

#adonis is stat tool to determine if matrix is significant; requires matrix generated above
adonis2(clr_dist_matrix ~ binID, data=data.frame(meta), permutations= 10000)
```

```
clr_dist_matrix <- phyloseq::distance(wetlands_phyloseq, method = "unifrac") #building distance matrix based on unifrac

#adonis is stat tool to determine if matrix is significant; requires matrix generated above
adonis2(clr_dist_matrix ~ binID, data=data.frame(meta), permutations= 10000)

#used bray to generate this
pcoa_bc <- ordinate(wetlands_phyloseq, method="PCoA", distance= clr_dist_matrix)

plot_ordination(wetlands_phyloseq, pcoa_bc, color = "binID")+
  labs(pch="", col = "Location")+
  stat_ellipse(aes(group= binID), linetype = 2)



###### Taxa Bar Plots ######

#function for combining pvalues
combine_pvalues = function(p){
  return(1-pchisq(-2*sum(log(p),na.rm=T),2*sum(!is.na(p))))
}

#Extracting OTU data
otu_table = data.frame(otu_table(wetlands_phyloseq))
correct_ASV_names= rownames(otu_table)
otu_table = data.frame(t(otu_table))
colnames(otu_table) = correct_ASV_names
otu_table$ID = rownames(otu_table)
otu_table$ID = gsub("^X","",otu_table$ID)
```

```
#Extracting metadata
metadata = data.frame(sample_data(wetlands_phyloseq))
metadata$ID = rownames(metadata)

TAX$ASV = rownames(TAX)

#joining OTU and metadata
otu_meta = inner_join(metadata,otu_table , by = "ID")

#Transforming the OTU matrix to a single column called abundance. 45 represents the number of metadata columns we want to exclude.

grouped = gather(otu_meta, key = "ASV", value = "abundance", -(1:45) )
grouped$ASV = gsub("^X","",grouped$ASV)

#joining the taxa information to this transformed dataframe
grouped_taxa = inner_join(TAX, grouped, by = "ASV", multiple = "all")

#collecting the list of unique severity names for the loop (control, mild, moderate, severe)
vars = unique(grouped_taxa$binID)

#Creating an empty dataframe that the loop will fill. Re run this line before re running the loop.
data_rel = data.frame()

#Generating relative abundance

#looping though each severity index to create a relative abundance measure.
#can change the classification system to generate different plots; just replace "Phylum" with x
```

```
for (i in vars){
  #i = "ithica_urban"
  df = grouped_taxa %>%
    filter(binID == i)

  df_sum = df %>%
    group_by(binID, Phylum) %>%
    summarize(rel_abs = sum(abundance))

  df_sum$rel_abs = as.factor(df_sum$rel_abs)
  row_sub = apply(df_sum, 1, function(row) all(row !=0 ))

  df_sum = df_sum[row_sub,]
  df_sum$rel_abs = as.numeric(as.character(df_sum$rel_abs))
  df_sum = na.omit(df_sum)
  count = sum(df_sum$rel_abs)
  df_sum$rel_abs = df_sum$rel_abs/count*100

  data_rel = rbind(data_rel, df_sum)
  data_rel = na.omit(data_rel)
}

#plotting the results at the phylum level

ggplot(data = data_rel, aes(binID,rel_abs, fill = Phylum))+
  geom_col(color = "black")+
  theme(axis.text.x = element_text(angle = -90))+
  labs(y = "Relative abundance", x = "Location")+
```

```
  theme_classic()+
  theme(axis.text = element_text(size = 15, face = "bold"),
        axis.title = element_text(size = 15,face = "bold"),
        legend.position = "none")

###### Correlation Analysis ######

#Extracting OTU data info and a bit of data manipulation to allow inner_join later on
otu_table = data.frame(otu_table(wetlands_rare))
correct_ASV_names= rownames(otu_table)
otu_table = data.frame(t(otu_table))
colnames(otu_table) = correct_ASV_names
otu_table$ID = rownames(otu_table)
otu_table$ID = gsub("^X","",otu_table$ID)

#Extracting metadata
metadata = data.frame(sample_data(wetlands_rare))
metadata$ID = rownames(metadata)

#cleaning up the taxa names
tax_mat = data.frame(tax_table(wetlands_rare))
tax_mat$Phylum = gsub("^...","",tax_mat$Phylum)
tax_mat$Class = gsub("^...","",tax_mat$Class)
tax_mat$Order = gsub("^...","",tax_mat$Order)
tax_mat$Family = gsub("^...","",tax_mat$Family)
tax_mat$Genus = gsub("^...","",tax_mat$Genus)
tax_mat$Species = gsub("^...","",tax_mat$Species)
```

```
#adding column of ASV to join it with meta/otu info
tax_mat$ASV = rownames(tax_mat)

#joining OTU and metadata
otu_meta = inner_join(metadata,otu_table , by = "ID")

#Transforming the OTU matrix to a single column called abundance. 53 represents the number of metadata columns we want to exclude.
grouped = gather(otu_meta, key = "ASV", value = "abundance", -(1:45) )

#joining the taxa information to this transformed dataframe
grouped_taxa = inner_join(tax_mat, grouped, by = "ASV", multiple = "all")

#list of significant Order

list_urb = c("Rokubacteriales",
             "Spirochaetales",
             "vicinamibacterales",
             "Rhizobiales",
             "Nb1-j")
list_rur = c("Defluviicoccales",
             "Methanosarciniales",
             "Bacteroidetes_VC2.1_Bac22",
             "Flavobacteriales",
             "Cytophagales",
             "sphingobacteriales")

#loop through diff env. var with each list
```

```
for (i in list_rur){
  stats_table = list()
  #i = "Sulfuritalea"
  df = grouped_taxa %>%
    filter(Order == i)
  df = df%>%
    filter(abundance != 0)
  #need to change x=as.numeric(env. var)
  ggplt <-
ggplot(df,aes(x=as.numeric(microbial_biomass_nitrogen),y=abundance,shape=binID))+
    geom_point()+
    theme_classic()+
    geom_smooth(method=lm,fullrange=TRUE,
                aes(color=binID))
  #rename .png to reflect env. var
  ggsave(paste(i,"_microbial_biomass_nitrogen.png"),plot = ggplt, path=
"correlation_curves/Final_correlation_curves/")


  ####################################
  location = unique(df$binID)
  for (j in location){
    #j = "auburn_urban"
    site = df %>%
      filter(binID == j)
    #need to change lm(env. var ~abundance . . .)
    fit.lm <- lm(microbial_biomass_nitrogen ~ abundance, data = site)
    summary(fit.lm)


    stats = table(paste("Stats for ",i,"in ",j,"Adj R2 = ",signif(summary(fit.lm)$adj.r.squared, 5),
```

```
                    "Intercept =",signif(fit.lm$coef[[1]],5 ),
                    " Slope =",signif(fit.lm$coef[[2]], 5),
                    " P =",signif(summary(fit.lm)$coef[2,4], 5)))
    stats_table = append(stats_table,stats)



  }
  #exporting the stats table, change name to reflect env. var
  capture.output(summary(stats_table), file =
(paste(i,"microbial_biomass_nitrogen_Stats_results.txt")))
}



###### Graphing Environmental Variables######

#grouping environemental conditions
library(ggplot2)

#ph
ggplot(SAMP, aes(x = binID, y = as.numeric(ph), fill = binID)) +
  geom_boxplot(alpha = 0.5) +
  geom_point(position = position_jitter(seed = 1, width = 0.2))+
  stat_summary(fun.y=mean, geom="point", shape=23, size=4)

#biomass
ggplot(SAMP, aes(x = binID, y = as.numeric(microbial_biomass_nitrogen), fill = binID)) +
  geom_violin(alpha = 0.5) +
  geom_point(position = position_jitter(seed = 1, width = 0.2))+
  stat_summary(fun.y=mean, geom="point", shape=23, size=4)
```

```
#percent_base_saturation
ggplot(SAMP, aes(x = binID, y = as.numeric(percent_base_saturation), fill = binID)) +
  geom_violin(alpha = 0.5) +
  geom_point(position = position_jitter(seed = 1, width = 0.2))+
  stat_summary(fun.y=mean, geom="point", shape=23, size=4)

#no3
ggplot(SAMP, aes(x = binID, y = as.numeric(soil_no3), fill = binID)) +
  geom_violin(alpha = 0.5) +
  geom_point(position = position_jitter(seed = 1, width = 0.2))+
  stat_summary(fun.y=mean, geom="point", shape=23, size=4)

#nh4
ggplot(SAMP, aes(x = binID, y = as.numeric(soil_nh4), fill = binID)) +
  geom_violin(alpha = 0.5) +
  geom_point(position = position_jitter(seed = 1, width = 0.2))+
  stat_summary(fun.y=mean, geom="point", shape=23, size=4)



#Group 16 DESeq2 Differential Abundance Script

#Script is adapted from Christopher Lee of the University of British Columbia's
#Microbiology and Immunology Dept.

#Differential abundance analysis in DESeq2
#This script compares the microbial orders between the urban site Ithaca and the
```

```
# rural site Owego


#Loading libraries necessary for DESeq2 analysis
library(phyloseq)
library(DESeq2)
library(dplyr)

combine_pvalues = function(p){
  return(1-pchisq(-2*sum(log(p),na.rm=T),2*sum(!is.na(p))))
}


#Constructing the phyloseq object again
load("TAX_phyloseq-obj.RData")
TAX = data.frame(TAX)
TAX$ASV = rownames(TAX)
load("OTU_phyloseq-obj.RData")
load("metadata_phyloseq-obj.RData")
load("phylotree.RData")

#The next portion describes filtering steps:
wetlands_phyloseq <- phyloseq(OTU, SAMP, TAX, phylotree)
#Removing any bacteria that have a total count >2 across all sites
wetlands_phyloseq <- filter_taxa(wetlands_phyloseq, function(x) sum(x)>2, prune = TRUE)
#Adding 1 pseudocount to meet DEseq2's negative binomial distribution assumption
otu_table(wetlands_phyloseq) = otu_table(wetlands_phyloseq)+1
#Filtering the data to a single comparison between the urban and rural sites of interest
```

```
wetlands_filtabrum_vs_cuyler<- subset_samples(wetlands_phyloseq, binID%in%
c("ithica_urban", "owego_rural"))

#The next segment shows how to construct a DESeq2 object for differential abundance
#result analysis
#Converting phyloseq to DEseq object
deseq_filtabrum_vs_cuyler<- phyloseq_to_deseq2(wetlands_filtabrum_vs_cuyler,~binID)
#Running DESeq2
wetlands_phyloseq <- DESeq(deseq_filtabrum_vs_cuyler,test="Wald",fitType="parametric")
#Getting the results!
wetlands_phyloseq_results <- results(wetlands_phyloseq)
#Converting the results to a dataframe
wetlands_phyloseq_results = data.frame(wetlands_phyloseq_results)
#Adding a colunm to the results called "ASV"
wetlands_phyloseq_results$ASV = rownames(wetlands_phyloseq_results)

#By combining the DESeq2 results with the taxanomic information, we can
#observe where microbes change the most signifigantly between sites.
wetlands_phyloseq_results_taxa = inner_join(TAX,wetlands_phyloseq_results, by = "ASV" )

#We then continue our filtering efforts to include signifigant changse with a
# p < 0.001 for statistically signifigant results
significant_res = wetlands_phyloseq_results_taxa %>%
  filter(padj < 0.001) %>%
  filter(baseMean > 10)

#This next segment of code allows us to analyze our samples at different
#taxanomic levels
wetlands_phyloseq_results_genus_summary<-significant_res %>%
```

```
  group_by(Phylum,Class,Order,Family,Genus) %>%
  select(Phylum,Class,Order,Family,Genus,log2FoldChange,padj,baseMean,pvalue) %>%

dplyr::summarise(mean_2_fold_change=mean(log2FoldChange),padj=combine_pvalues(padj),c
ountmean = sum(baseMean),pvalue=combine_pvalues(pvalue))%>%
  filter(Order != "g__uncultured")



wetlands_phyloseq_results_family_summary<-significant_res %>%
  group_by(Phylum,Class,Order,Family) %>%
  select(Phylum,Class,Order,Family,log2FoldChange,padj,baseMean,pvalue) %>%

dplyr::summarise(mean_2_fold_change=mean(log2FoldChange),padj=combine_pvalues(padj),c
ountmean = sum(baseMean),pvalue=combine_pvalues(pvalue))



wetlands_phyloseq_results_order_summary<-significant_res %>%
  group_by(Phylum,Class,Order) %>%
  select(Phylum,Class,Order,log2FoldChange,padj,baseMean,pvalue) %>%

dplyr::summarise(mean_2_fold_change=mean(log2FoldChange),padj=combine_pvalues(padj),c
ountmean = sum(baseMean),pvalue=combine_pvalues(pvalue))


wetlands_phyloseq_results_class_summary<-significant_res %>%
  group_by(Phylum,Class) %>%
  select(Phylum,Class,log2FoldChange,padj,baseMean,pvalue) %>%

dplyr::summarise(mean_2_fold_change=mean(log2FoldChange),padj=combine_pvalues(padj),c
ountmean = sum(baseMean),pvalue=combine_pvalues(pvalue))


wetlands_phyloseq_results_phylum_summary<-significant_res %>%
```

```
  group_by(Phylum) %>%
  select(Phylum,log2FoldChange,padj,baseMean,pvalue) %>%

dplyr::summarise(mean_2_fold_change=mean(log2FoldChange),padj=combine_pvalues(padj),c
ountmean = sum(baseMean),pvalue=combine_pvalues(pvalue))


#The plot generated below is filtered to show differential abundance between
#the sites of interest at the 'Order' level


#To further clean up results,the following line may be uncommented to
#remove the '_____g' tag present at the genus level, or it may be adapted to
#remove tags of other taxanomic levels such as order in which case we would remove
# '____o'
#wetlands_phyloseq_results_genus_summary$Order =
gsub("^...","",wetlands_phyloseq_results_genus_summary$Order)


#We now are filtering our data so that only ordders which demonstrate a
#log2fold change of 2 and are statistically signifigant ( p > 0.001) are shown
wetlands_phyloseq_results_genus_summary = wetlands_phyloseq_results_genus_summary
%>% filter(abs(mean_2_fold_change) > 2)



#This segment now creates the plot and colours the orders based on their phyla.
DEseqafterfiltergenus <-
ggplot(na.omit(wetlands_phyloseq_results_genus_summary),aes(x=reorder(Order, -
mean_2_fold_change),y=mean_2_fold_change,color = Phylum))+
  geom_point(size=3) +
  geom_hline(yintercept = 0,size=1)+
  theme_bw()+
  theme(axis.text.x = element_text(angle = -90,hjust = 0,vjust = 0.5),
```

```
      text = element_text(face = "bold",size=7),
      axis.text.x.bottom  = element_text(size=7,angle = -90))+
  labs(x="Order",y="Log2 Fold Change",color= "Phylum")


#Saving a table of the results for order at the comparison between the
#Urban Ithaca site and the Rural Owego site


write.table(wetlands_phyloseq_results_genus_summary,file = "Ithaca urban vs Owego rural Order level.txt")
#Saving the plot to your current working directory. Run getwd() to find out where the plot was saved
ggsave("Ithaca urban vs Owego rural order.png",plot= DEseqafterfiltergenus)
```