Microbiology Conference Paper

Fiona Dubay

5/13/2022

Contents

Introduction

Personal Note

The conference work I did this semester for Microbiology took on many forms, most of which involved steeper learning curves than I had initially anticipated. I'm proudest of the skills I gained this semester, and the time I spent cultivating them.

Stages of Conferende Work

All of my work this semester evolved from an interest in data visualization. The opportunity to make visualizations from water quality sampling data presented a chance to learn R and RStudio in an applied setting, which is something I had been hoping to do for a while. In addition to that, it was a chance to have give support to important research. The opportunity to make an animation that accessibly presented CURB's water quality data offered the same exciting possibilities.

CURB Enteroroccus Data Animation

The Center for the Urban River at Beczak (CURB) in Yonkers has a mission to increase the environmental knowledge of communities within the Hudson Watershed by collecting and distributing information on water quality and the safety of their river networks. Formatting data in a way that is accessible and understandable is extremely important to CURB and other organizations like them (Groundworks, Riverkeeper, etc.), as these data directly impact the health and safety of Yonkers. CURB's Saw Mill River water quality program has created one such dataset of Enterococcus counts over the past 8 years.

Enterococci are fecal indicator bacteria (FIB) used to determine water safety levels by the New York State Environmental Protection Agency (EPA). They inhabit the gastrointestinal tracts of animals, and are typically found in high concentrations in human feces, between 10^4 and 10^6 per gram wet weight (Boehm et al. 2014). While using Enterococci is problematic for finding evidence of solely human fecal matter, their use as a FIB is very widespread because of their culturability, and their correlation with human health coutcomes in fresh and marine waters (Pyappanahalli et al. 2012). The EPA has dictated that there should be no more than 104 colony-forming unite (CFUs) per 100-mL within a water sample. Sources of fecal material can enter waterways in many ways, including wastewater treatment plant effluent, leaking septic systems, domestic animal and wildlife waste, and Combined Sewage Overfloe (CSO) events which are common in New York City and greatly increase levels of contamination in local waterways. CURB's Enterococcus count dataset contains counts from 16 different sampling sites along the Saw Mill River. With such a large span over both time and location, CURB needs a way to visualize these data that's meaningful. I attempted to create an animation for that purpose.

Python work

The first step I took was importing the master spreadsheet into Python so that I could clean it up a bit. The following code chunk contains my cleaning process:

```python
# import the pandas library and call it in the future as pd
import pandas as pd

# import the master CURB entero count csv as a dataframe
df = pd.read_csv('saw_miller_master_CURB.csv')

# drop rows that have no Site ID or River Mile entries
df = df.dropna(subset=['River Mile', 'Site ID'])

# exclude the sites: JFK Marina Launch and Yonkers Paddling and Rowing Club
to_drop = ['Yonkers- JFK Marina boat launch', 'Yonkers Paddling and Rowing Club']
df = df[df['Site Name'].isin(to_drop) == False]

# replace 'Entero. Count' count values of '<10' to 9 and '>24196' to 24197
df['Entero. Count'] = df['Entero. Count'].replace(to_replace=['<10', '>24196'], value=[9, 24197])

# force non-numeric 'Entero. Count' entries to become NaN
df['Entero. Count'] = pd.to_numeric(df['Entero. Count'], errors='coerce')
# now drop all rows with NaN in the 'Entero. Count' column
df = df.dropna(subset='Entero. Count')

# change data type of 'Entero. Count' to numeric
df['Entero. Count'] = df['Entero. Count'].astype('float')

# drop columns that have no data
```

```
df = df.dropna(axis=1, how="any")

# export dataframe as a csv
df.to_csv('cleaned_saw_miller_CURB.csv')
```

I was given the site locations as points in a Google Maps folder, which gave me some difficulty exporting as coordinates. Once I did that, I had two datasheets: one with the Enterococcus counts and one with the site coordinates. After correcting some Site ID values by hand, I merged the two sheets with the following code:

```
import pandas as pd

# import data as dataframes using pandas function read_csv()
# cleaned_saw_miller_CURB.csv contains all entero counts
# site_coordinates.csv contains the site names and coordinates for each site
# the parse_dates argument scans the 'Sample Date' column and turns any date strings into datetime
counts = pd.read_csv('cleaned_saw_miller_CURB.csv', parse_dates=['Sample Date'])
sites = pd.read_csv('site_coordinates.csv')

# merge the two dataframes into one
# on = specifies the column used to combine the two dataframes, in this case we're combining by site ID
# how ="left" specifies that sites is merged onto counts, not the other way around
# this way we keep all the rows in the entero counts and simply add the site coordinates to them
merged = pd.DataFrame(pd.merge(counts, sites, on="Site ID", how="left"))

# created a cleaned version with only the most important columns for QGIS
# only keep the columns listed within merged
# (technically I'm making a copy of just the needed columns from the merged dataframe
# and assigned it to a new dataframe name)
merged_simple = merged[['Site ID', 'Sample Date', 'X', 'Y', 'Entero. Count']].copy()

# export this new merged dataframe as a csv to use in QGIS
merged.to_csv('merged_sites_counts.csv')
```

Once this formatting was complete, the data were ready to be used in QGIS.

QGIS Animation Troubles

Animating maps in QGIS turned out to be a lot more challenging than I expected. I used QGIS version 3.22 which has a new 'Temporal Controller' tool that ended up being more unwieldy than useful. The general steps I took to produce an animation draft involved importing the data, adjusting the symbology to change color and size depending on the Enterococcus count, and temporalizing the data for time series animation. I'm not happy with the animation draft I made, and it took me far too long to make for its quality level. The next step I see myself taking for this project is starting over in with a different software.

Visualizations with *R* and RStudio

My introduction to R and RStudio began with the book **Getting Started with R An Introduction For Biologists (Second Edition)**. Once I got comfortable visualizing with *ggplot2*, I followed the DADA2 Pipeline Tutorial (1.16) with the given example data. Then in class, I ran the dada2 pipeline with data from our water quality lab. These exercises gave me a strong enough understanding visualizing Illumina-sequenced paired-end samples to begin working with data from the daylighting paper.

Daylighting paper

Sarah Fiordaliso, La Zhen Han, George A. Scott, and Michelle H. Hersh are drafting a paper on their research of changes in bacterial community structure within the Saw Mill River caused by sunlight exposure from daylighting. To visualize the bacterial communities in their samples, collected and sequenced back in 2015, I had to run them through the dada2 pipeline. Using the methods from lab, I made a few changes to the pipeline, starting with assigning taxonomy.

Taxonomy

In order to make sure taxonomy assignments were as accurate as possible, I followed the taxonomy extension instructions detailed in the dada2 tutorial. This made species level assignments based on exact matching between ASVs and sequenced reference strains. I used the Silva species assignment database (version 138.1) to assign species to 16S gene fragments.

```
taxa <- assignTaxonomy(seqtab.nochim2, "../data_for_paper/tax/silva_nr99_v138.1_train_set.fa.gz",
                       multithread=TRUE)
taxa <- addSpecies(taxa, "../data_for_paper/tax/silva_species_assignment_v138.1.fa.gz")
taxa.print <- taxa # remove sequence rownames for display only
rownames(taxa.print) <- NULL
head(taxa.print)
```

```
##      Kingdom    Phylum           Class                 Order
## [1,] "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Burkholderiales"
## [2,] "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Burkholderiales"
## [3,] "Bacteria" "Bacteroidota"   "Bacteroidia"         "Sphingobacteriales"
## [4,] "Bacteria" "Cyanobacteria"  "Cyanobacteriia"      "Chloroplast"
## [5,] "Bacteria" "Proteobacteria" "Gammaproteobacteria" "Burkholderiales"
## [6,] "Bacteria" "Cyanobacteria"  "Cyanobacteriia"      "Chloroplast"
##      Family               Genus        Species
## [1,] "Comamonadaceae"     "Rhodoferax" NA
## [2,] "Comamonadaceae"     "Rhodoferax" NA
## [3,] "Sphingobacteriaceae" "Solitalea"  NA
## [4,] NA                   NA           NA
## [5,] "Comamonadaceae"     "Rhodoferax" NA
## [6,] NA                   NA           NA
```

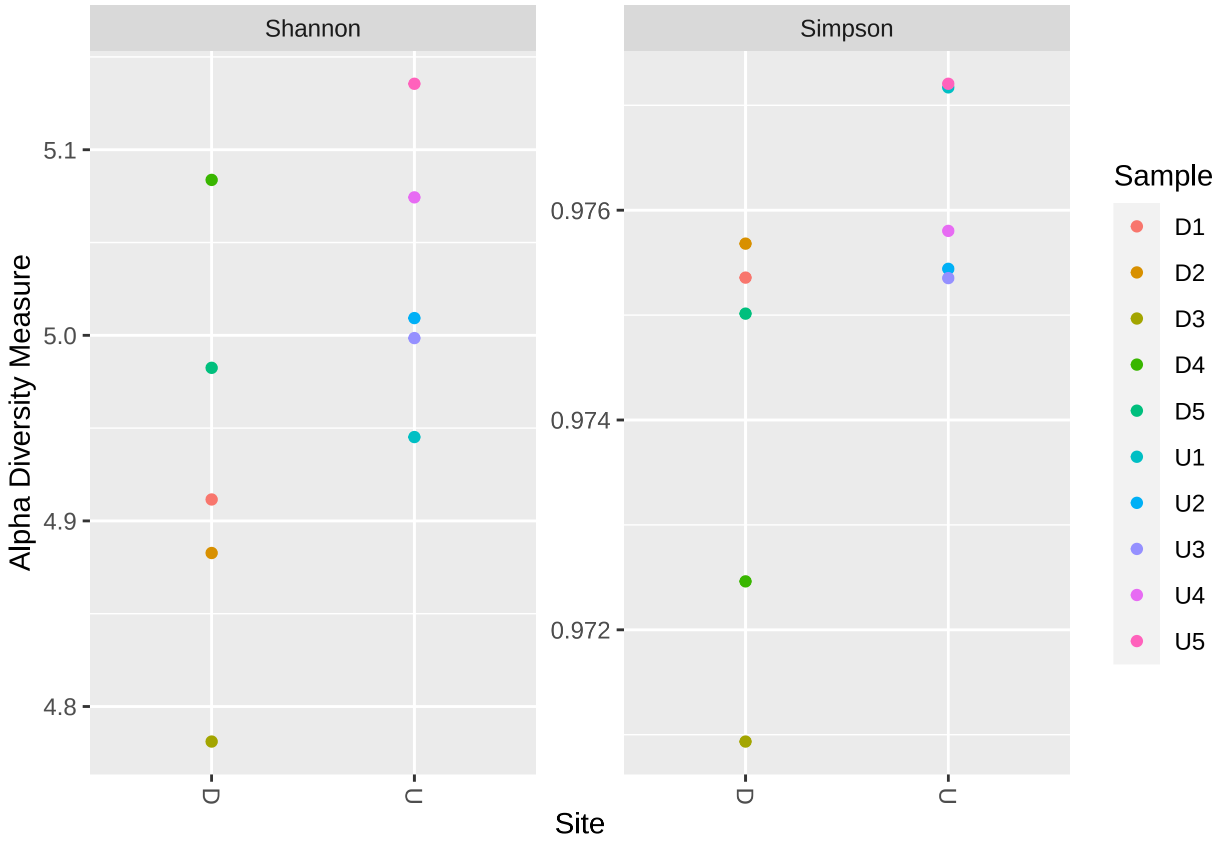
Phyloseq

The phyloseq formatting for these data was different than previous formatting I had done on our class data from our bioinformatics lab. In this case, the most important part was labeling the samples so they could, later on, be easily separated by downstream and upstream sites.

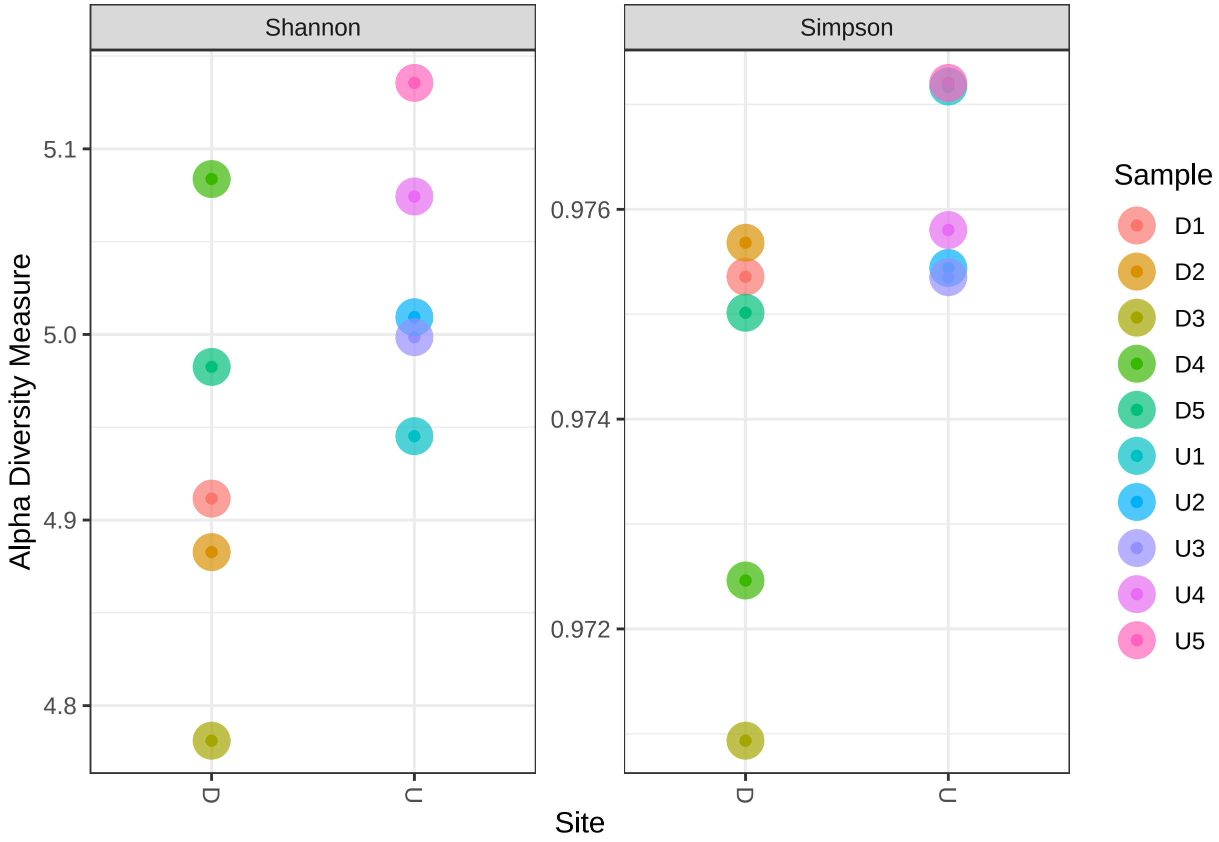
```
# make a data frame for covariates
samples.out <- rownames(seqtab.nochim2)
siteloc <- substr(samples.out,1,1) # create siteloc with first letter of sample name: 'U' or 'D'
# create labels
loclabel <- c("Downstream","Downstream","Downstream","Downstream","Downstream","Upstream","Upstream","Up
# create the dataframe
samdf <- data.frame(Sample=samples.out, Site=siteloc, Label=loclabel)
rownames(samdf) <- samples.out
```

Alpha Diversity

Here's the first alpha diversity plot I made.

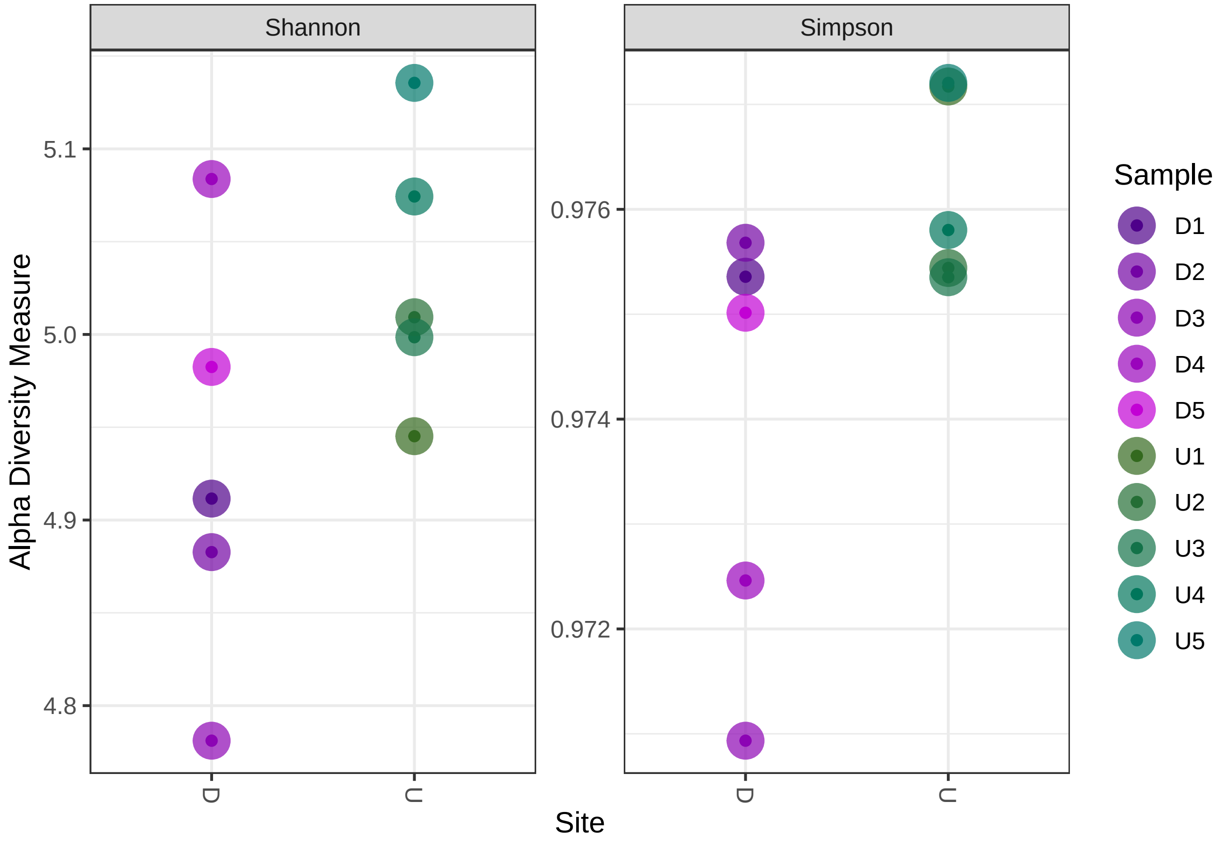


Since it was the first plot I made with these data, it was very exciting to see! But, of course, it had a long way to go. For starters, the points were too small and the grey background distracted from the colors and made them harder to distinguish. Here's the second iteration of my diversity plots, which I brought to conference:



In conference we discussed changing the color scheme to create more of a distinction between the two sites. Here's the third iteration after the color change, with code included:

```
# set the theme so the plot doesn't have a gross grey background, I'll keep that in future plots
theme_set(theme_bw())
# create base richness plot
rich <- plot_richness(ps_good, x="Site", measures=c("Shannon", "Simpson"), color="Sample")
# adjust size of points and alpha (opacity)
rich + geom_point(size=6, alpha=0.7) +
  # I got these hex colors by using an online gradient generator
  scale_color_manual(values = c('#4e018a', '#7303a4', '#8c04b4', '#9a04bc', '#c201d4','#33691e', '#246e3
```

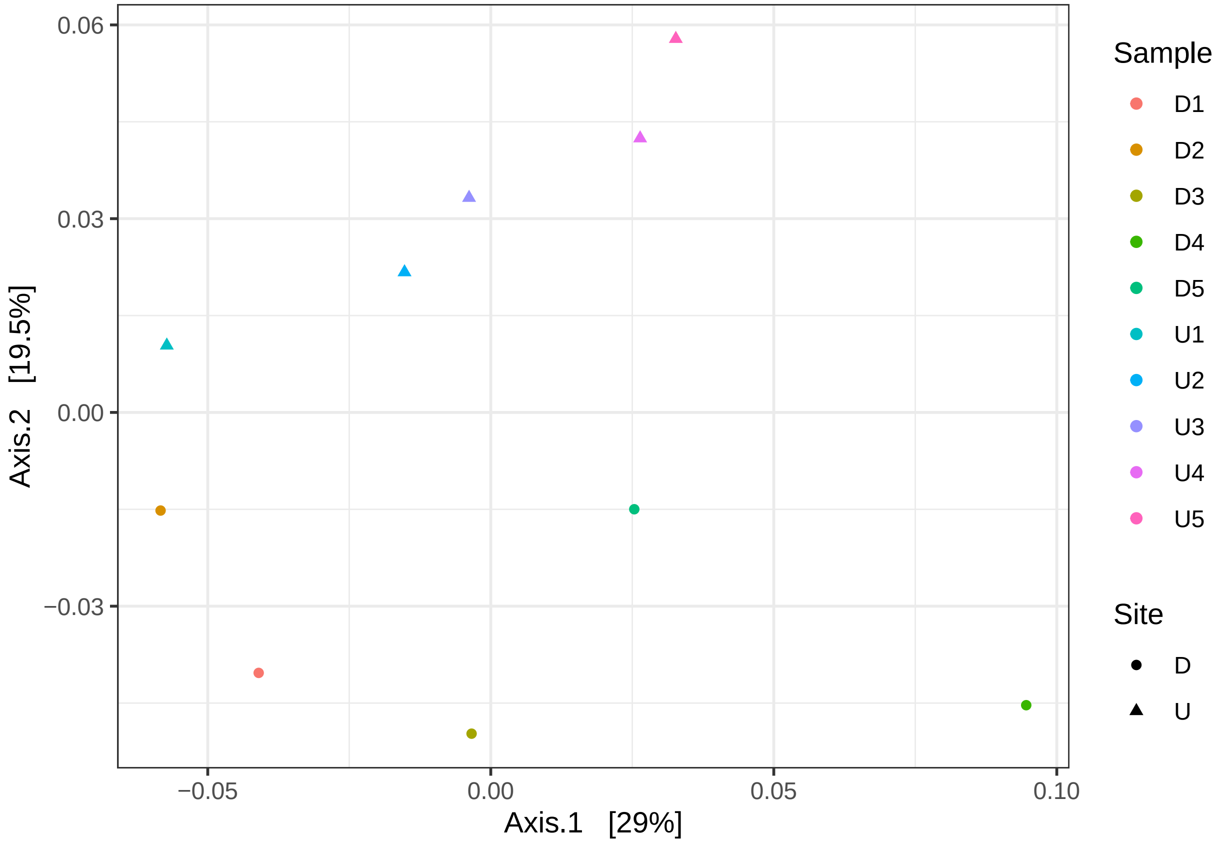


Ordination

When making ordination plots previously, I had used the NMDS ordination method. The draft of the daylighting paper used PCoA so I figured it would make sense for me to learn both. After a bit of research, I learned from this website that the algorithms used with NMDS and PCoA are very different. NMDS is an iterative method that can give a different result on re-analyses of the same data, however, the number of ordination axes can be fixed by the programmer. PCoA has a "unique analytical solution" and will return the same solution every time.

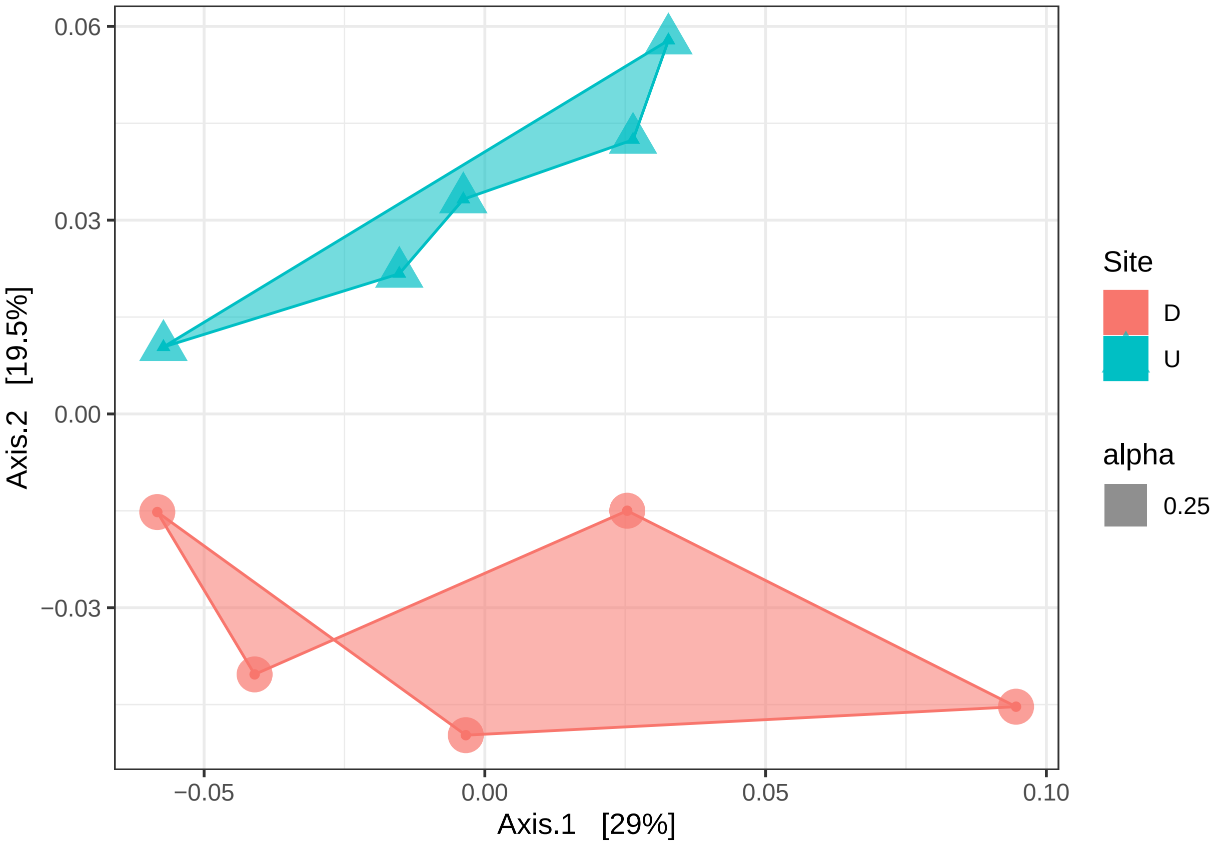
PCoA ordination:

```
# transform data to proportions as appropriate for Bray-Curtis distances
ps.prop <- transform_sample_counts(ps_good, function(otu) otu/sum(otu))
# ordinate with PCoA
ordu <- ordinate(ps.prop, "PCoA", "bray")
# make plot
plot_ordination(ps.prop, ordu, color="Sample", shape="Site")
```



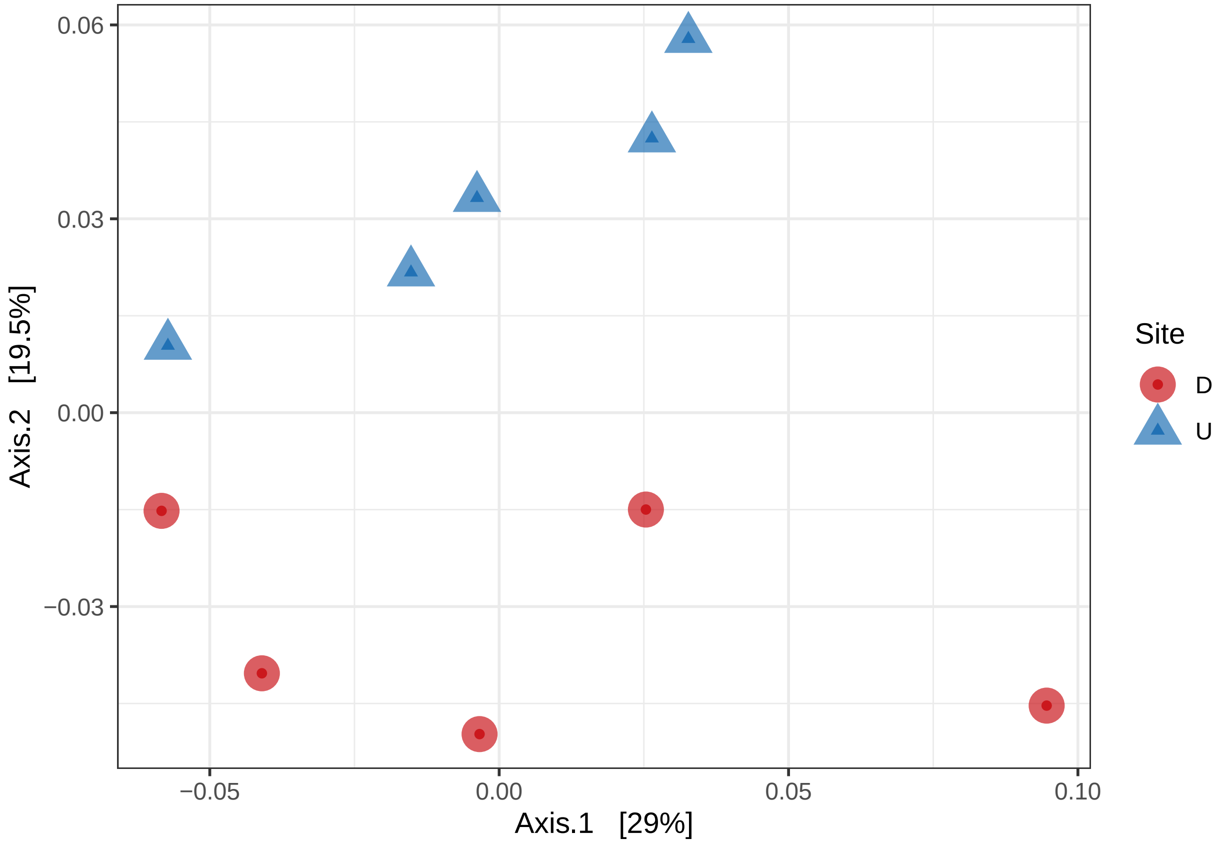
The first ordination plot I made I gave each individual sample its own color, but that ended up being a useless addition which made the plot too busy. In the next iteration I used both color and shape to distinguish between site locations. Following this Joey tutorial, I filled in the space between samples according to site.

```
ordplot <- plot_ordination(ps.prop, ordu, type='samples', color="Site", shape="Site")
# increase the size of points, lower opacity to 75%
ordplot + geom_point(size=6, alpha=.7) +
  # fill the space in between samples of each site, lower opacity to 25%
  geom_polygon(aes(fill=Site, alpha=0.25))
```



After discussing that version in conference I learned that, despite how cool I thought the shapes looked on the plot, they didn't represent anything. They revealed no new information. After removing them, I had the final version:

```
ordplot <- plot_ordination(ps.prop, ordu, type='samples', color="Site", shape="Site")
ordplot + geom_point(size=6, alpha=.7) +
  scale_color_manual(values =c('#cb181d', '#2171b5'))
```

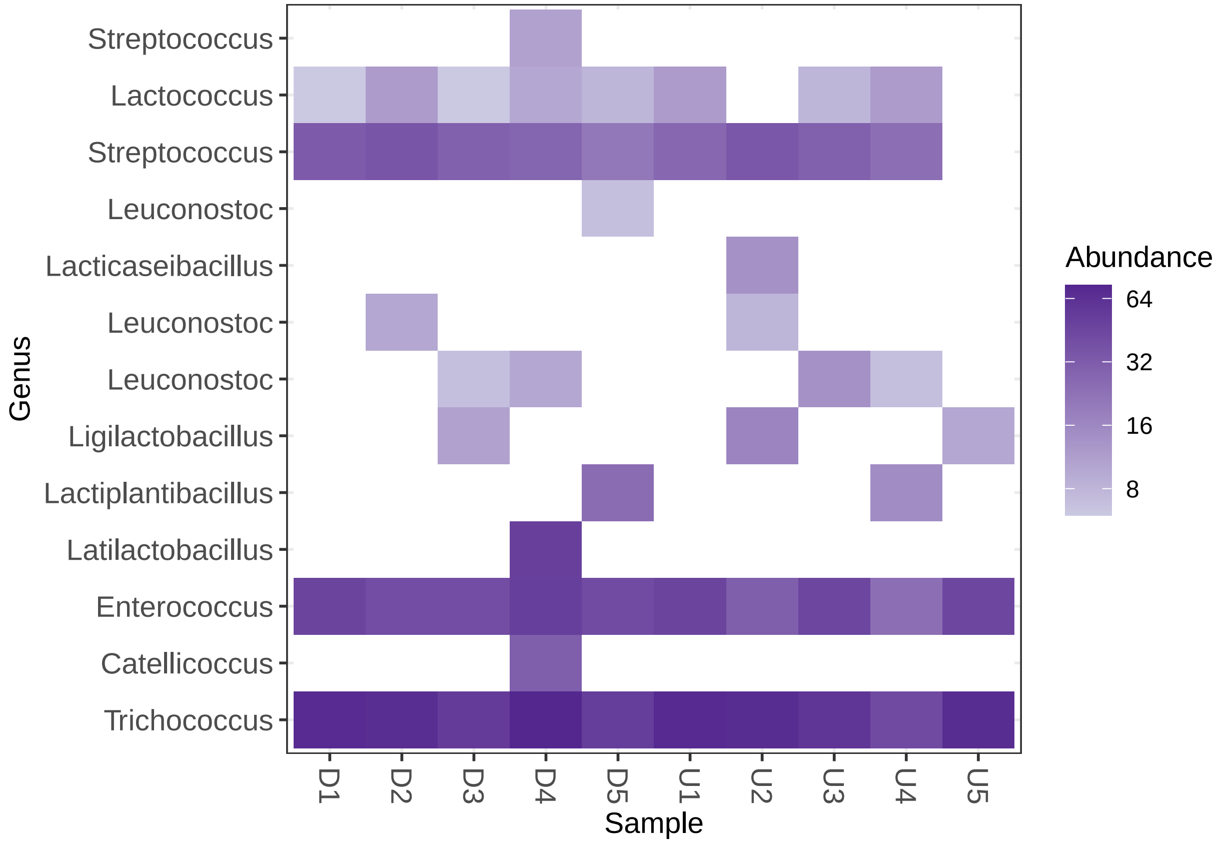


First Heatmaps

Instead of using basic bar plots (boring!) to represent bacterial community structure, my goal was to create either relative abundance bar plots or heatmaps. Figuring relative abundance bar plots would be easier I started with those. I found this Joey tutorial which used the plot_heatmap() function, documented here. With some trial and error, I was able to make a couple satisfactory heatmaps! I brought several intial plots to conference where we narrowed them down to the useful visualizations.

***Lactobacillales* Order Heatmap** The first heatmap I attempted with this method was genera within the *Lactobacillales* order. Here's the latest version, with code:

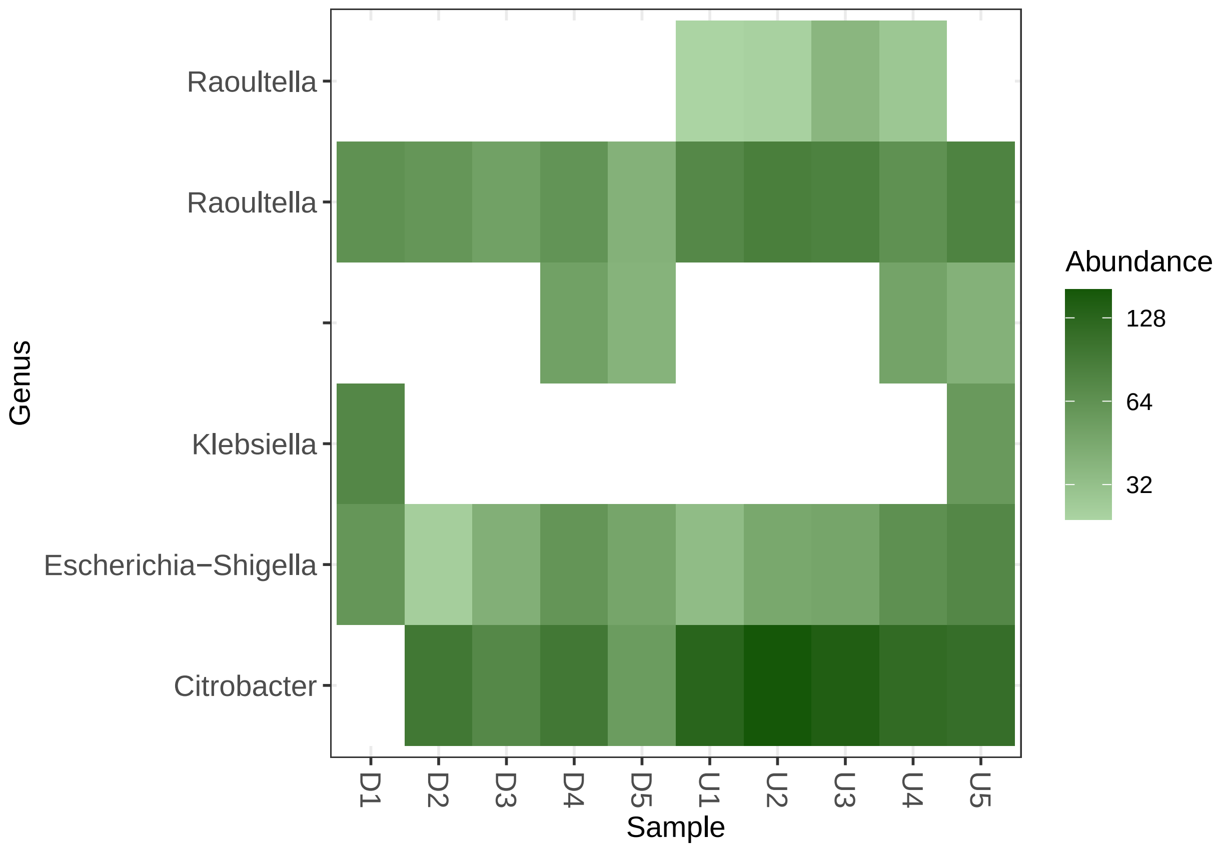
```r
# first plot: genera within the Lactobacillales order
# grab subset of taxa within Lactobacillales order from phyloseq object
tax1 <- subset_taxa(ps_good, Order=="Lactobacillales")
# plot_heatmap() function using PCoA ordination and Bray-Curtis distances:
# sample.label dictates the x-axis values and sample.order dicates the order of x ticks
# taxa.label dictates the taxonomic rank used for plotting and
# taxa.order dictates the order in which taxa are listed
# represent NA value with white
hm1 <- plot_heatmap(tax1, method="PCoA", distance="bray",
                    sample.label="Sample", sample.order="Sample",
                    taxa.label="Genus", taxa.order="Family",
                    low="#cbc9e2", high="#54278f", na.value="white")
print(hm1)
```



There's a couple issues with this plot, one of which being the y-axis labels. Without the full taxonomic names, there's a few repetitions, like with *Leuconostoc* and *Streptococcus*. The other issue is that even though the x-axis is ordered by site, there isn't a clear enough distinction between downstream and upstream samples. Having spent a couple hours on this plot already, I decided to leave it and come back if I had time later.

***Enterbacteriaceae* Family Heatmap** Moving on to a new heatmap, I plotted genera within the *Enterobacteriaceae* family, as that was of great importance to the study. Here's the latest version, with code:

```
# second plot: genera within the Enterobacteriaceae
# grab subset of taxa within Enterobacteriacea family from phyloseq object
tax2 <- subset_taxa(ps_good, Family=='Enterobacteriaceae')
# create heatmap
# once again, represent NA value with white
hm2 <- plot_heatmap(tax2, method="PCoA", distance="bray",
                    sample.label="Sample", sample.order="Sample",
                    taxa.label="Genus", taxa.order="Genus",
                    low="#abd4a3", high="#155708", na.value="white")
print(hm2)
```



One again, this is a cool visualization with a couple problems! Like the *Lactobacillales* heatmap, there should be a clearer distinction between data from the downstream and upstream sampling sites. An unfortunate problem with this plot is that one of the species within *Enterbacteriacea* that was found in some samples is unknown. The unknown genus is a blank space, but it should be notated as "unknown taxon" or something similar.

An big issue with both of these heatmaps is that their metric is "Abundance" and not relative abundance. Relative abundance heatmaps would be much more useful for comparing community structure between different samples.

After a lot of trial and equal amounts of error, I moved on from fixing these heatmaps and tried once more to make a relative abundance bar plot.

Relative Abundance Bar Plots and More Heatmaps

In my search for examples of attractive relative abundance bar plots (or, hot plots, as I like to call them), I found this mysterious tutorial from an open-source microbiome data analysis class that contained a very approachable and understandable walkthrough on how to create **both** relative abundance bar plots ***and*** relative abundance heat maps!

Top 20 Taxa Relative Abundance Bar Plot The process involved loading a few more packages:

```
library(RColorBrewer)
library(ggpubr)
library(microbiome)
```

I decided to only use the top 20 taxa:

```
top20 <- names(sort(taxa_sums(ps_good), decreasing=TRUE))[1:20]
ps.top20 <- transform_sample_counts(ps_good, function(OTU) OTU/sum(OTU))
ps.top20 <- prune_taxa(top20, ps.top20)
```

Further formatting and setup involved selecting a good color palette, labeling unclassified taxa, and changing taxa names to italic for the figure:

```
# create a new object that's the same as ps.top20 to alter
top20 <- ps.top20
taxic <- as.data.frame(top20@tax_table)
# define number of variable colors based on number of Family (change the level accordingly to phylum/cl
colorCount = length(unique(taxic$Family))
# change the palette as well as the number of colors will change according to palette.
getPalette = colorRampPalette(brewer.pal(7, "Paired"))
# add the OTU ids from OTU table into the taxa table at the end
taxic$OTU <- rownames(taxic)
# check the column names, now OTUs are included
colnames(taxic)
```

```
## [1] "Kingdom" "Phylum"  "Class"   "Order"   "Family"  "Genus"   "Species"
## [8] "OTU"
```

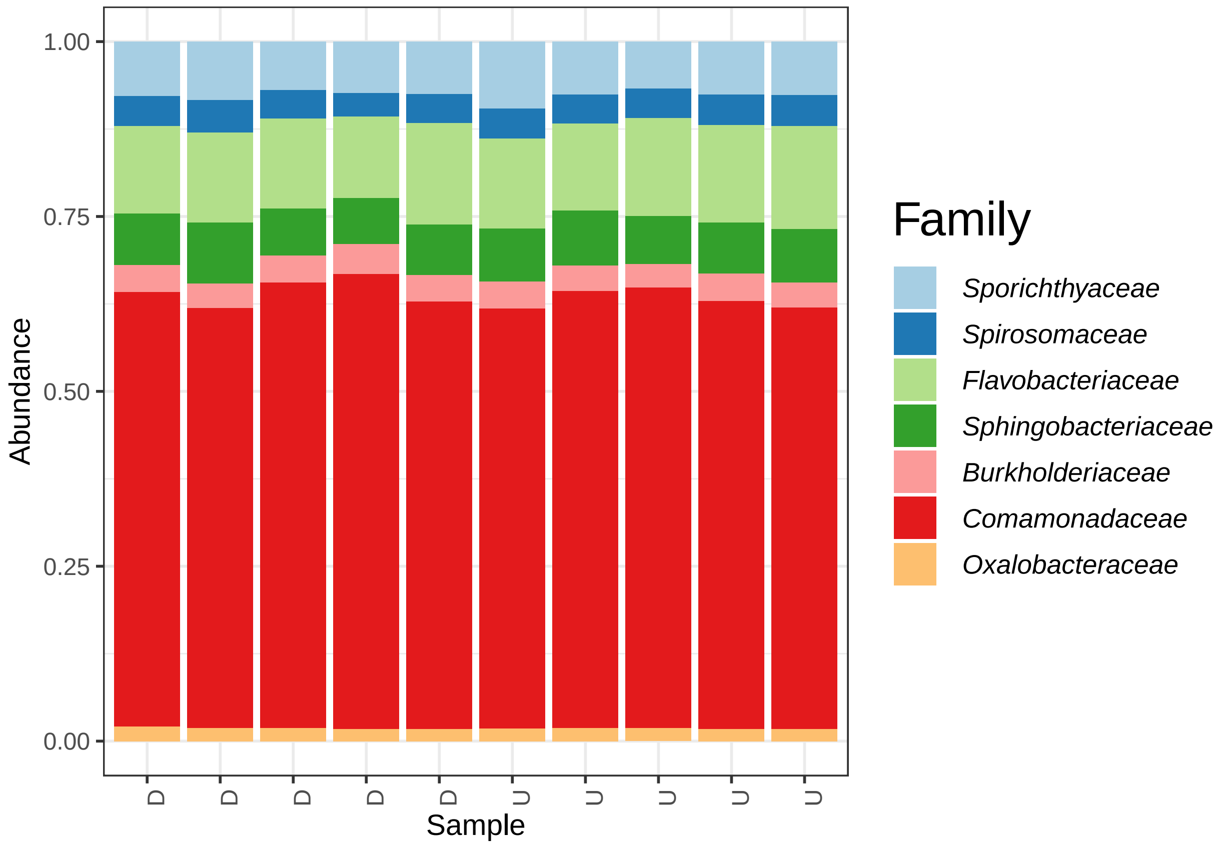
```
# label unclassified taxa as unclassified!
tax_table(top20)[tax_table(top20)[, "Family"] == "", "Family"] <- "Unclassified family"

# change the taxa names to italic
guide_italics <- guides(fill = guide_legend(label.theme = element_text(
  size = 10,
  face = "italic", colour = "Black", angle = 0
)))
```

Then, I aggregated taxa at the family level and plotted it! The tutorial included some great formatting steps, which I used and altered a bit.

```
# first remove the phy_tree
top20@phy_tree <- NULL

# merge at the family level using aggregate_taxa()
# the tutorial uses aggregate_top_taxa() which has since been deprecated
top20.fam <- microbiome::aggregate_taxa(top20, "Family")
# use the transform() function to convert to relative abundance
top20.fam.rel <- microbiome::transform(top20.fam, "compositional")
# plot!
plot.top20.fam.relAbun <- plot_composition(top20.fam.rel,
                                           sample.sort = "Sample",
                                           x.label = "Site")
# formatting
plot.top20.fam.relAbun <- plot.top20.fam.relAbun + theme(legend.position = "bottom")
plot.top20.fam.relAbun <- plot.top20.fam.relAbun + scale_fill_brewer("Family", palette = "Paired") + the
plot.top20.fam.relAbun <- plot.top20.fam.relAbun + theme(axis.text.x = element_text(angle = 90))
plot.top20.fam.relAbun <- plot.top20.fam.relAbun + guide_italics + theme(legend.title = element_text(si
print(plot.top20.fam.relAbun)
```



This plot tells such a great story about the community structure of the Saw Mill River. When looking at the old abundance bar plot, used in the draft of the daylighting paper, it appears that *Burkholderiaceae* are the dominant family present in both sampling locations. This new relative abundance chart shows that it's actually *Comamondaceae* that are the most prominant family. Now, this change could be in part due to it being a relative abundance plot rather than an abundance plot, but the difference between the two figures is too large for it to simply be a visualization change. It seems to me that a combination of additional dataset trimming and using the more thorough taxonomy assignment method is the cause of this change. *Comamonadaceae* and *Burkholderiaceae* are families within *Burkholderiales*, and order of *Pseudomonadota*. Both families are home to some opportunistic pathogens.

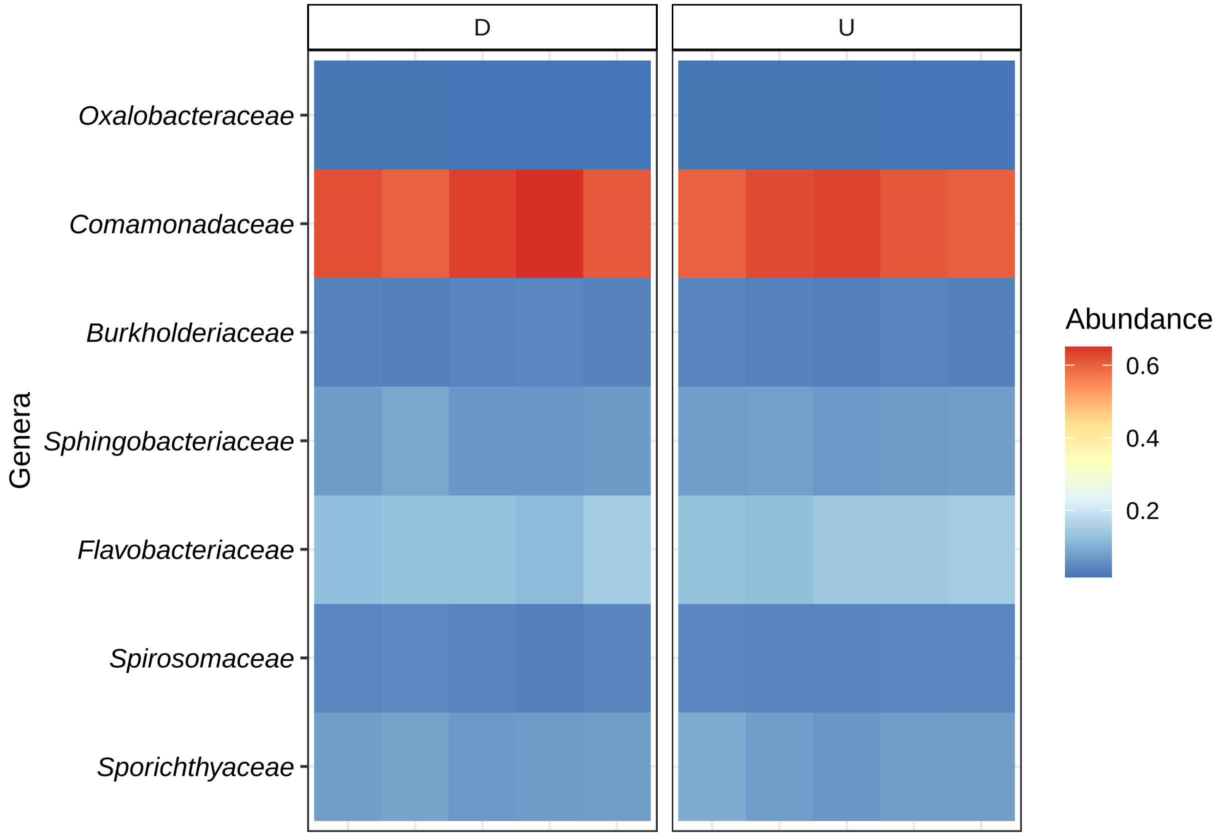
Top 20 Taxa Relative Abundance Heatmap Using the relative abundance data from the bar plot, I made a heatmap, just like in the tutorial:

```
# grab data from relative abundance plot and place it in a new object
data.top20 <- plot.top20.fam.relAbun$data
# check column names
colnames(data.top20)
```

```
## [1] "Tax"       "Sample"    "Abundance" "xlabel"
```

```r
# create the base plot
top20.heat <- ggplot(data.top20, aes(x = Sample, y = Tax)) + geom_tile(aes(fill = Abundance))
# choose color palette and set theme
top20.heat <- top20.heat + scale_fill_distiller("Abundance", palette = "RdYlBu") + theme_bw()
# make bacterial names italics
top20.heat <- top20.heat + theme(axis.text.y = element_text(colour = 'black',
                                                  size = 10,
                                                  face = 'italic'))
# make separate samples based on the main y-variable
top20.heat <- top20.heat + facet_grid(~xlabel,
                              scales = "free") + rremove("x.text")
top20.heat <- top20.heat + ylab("Genera")

# clean up the x-axis
top20.heat <- top20.heat + theme(axis.title.x=element_blank(),
                         axis.text.x=element_blank(),
                         axis.ticks.x=element_blank())
# clean up the facet label box
top20.heat <- top20.heat + theme(legend.key = element_blank(),
                         strip.background = element_rect(colour="black", fill="white"))
print(top20.heat)
```



Amazing! This method fixes the issues with my previous heatmaps. There's a clear distinction between downstream and upstream samples, and the unknown taxa are clearly labeled. Using relative abundance instead of just abundance reveals a lot more about the community structure of the Saw Mill. This figure shows the dominance of *Comamondaceae*, like the relative abundance bar plot but in a new way. Now, because this is a heatmap for the top 20 taxa, there obviously isn't much variation between the other genera.

Next, I performed the same steps for visualizing the *Lactobacillales* order and *Enterobacteriaceae* family with relative abundance plots and heatmaps.

```
# try second heatmap method but with Lactobacillales family
# tax1 is subsetted phyloseq object containing only taxa from Lactobacillales
lacto <- tax1
taxic2 <- as.data.frame(lacto@tax_table)
# set colors
colorCount = length(unique(taxic2$Genus))
getPalette = colorRampPalette(brewer.pal(10, "Paired"))

# add the OTU ids from OTU table into the taxa table at the end
taxic2$OTU <- rownames(taxic2)
colnames(taxic2)
```

Lactobacillales Relative Abundance Bar Plot

```
## [1] "Kingdom" "Phylum"  "Class"   "Order"   "Family"  "Genus"   "Species"
## [8] "OTU"
```

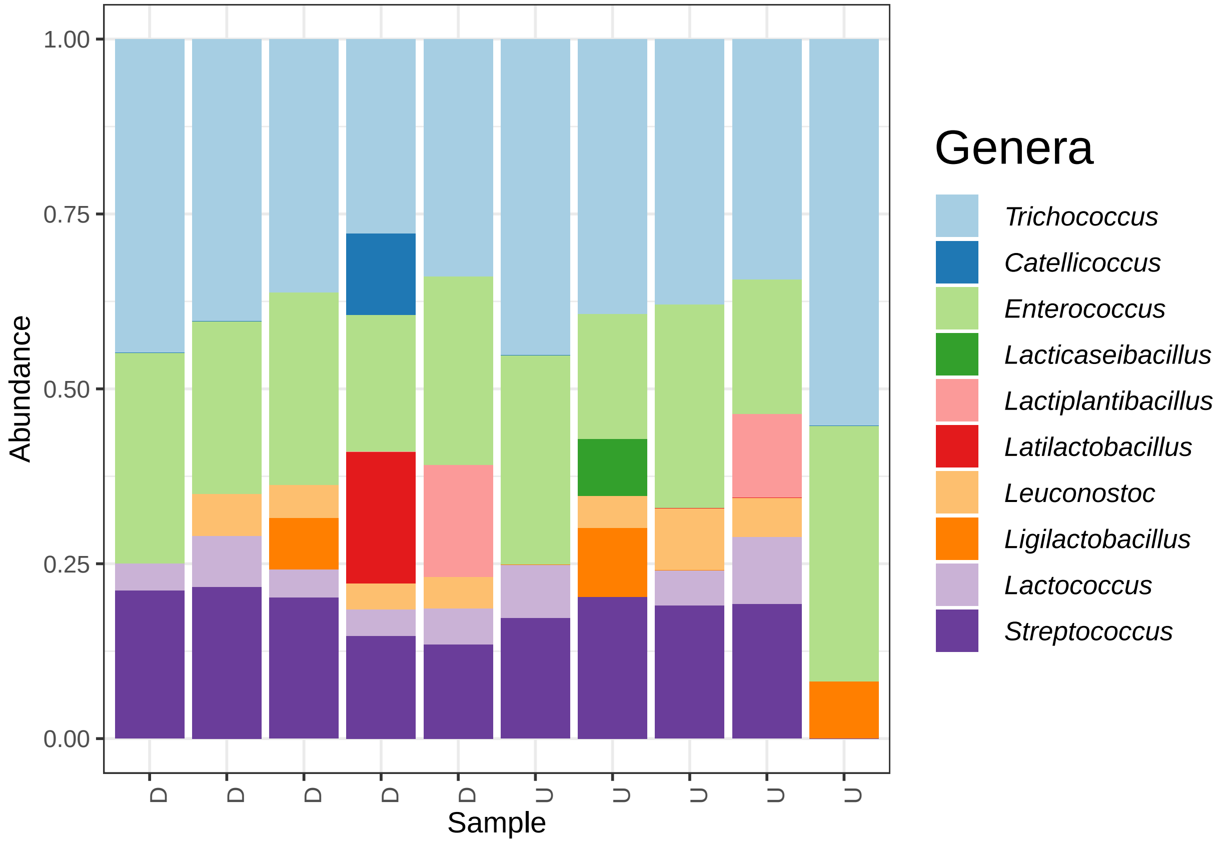
```
# edit the unclassified genera
tax_table(lacto)[tax_table(lacto)[, "Genus"] == "", "Genus"] <- "Unclassified genus"

# put taxonomic names in italics
guide_italics <- guides(fill = guide_legend(label.theme = element_text(
  size = 10,
  face = "italic", colour = "Black", angle = 0
)))

ps1.com2@phy_tree <- NULL

# merge at genus level
lacto.gen <- microbiome::aggregate_taxa(lacto, "Genus")
# transform() for relative abundance
lacto.gen.rel <- microbiome::transform(lacto.gen, "compositional")
# plot!
plot.lacto.relAbun <- plot_composition(lacto.gen.rel,
                                        sample.sort = "Sample",
                                        x.label = "Site")
plot.lacto.relAbun <- plot.lacto.relAbun + theme(legend.position = "bottom")
plot.lacto.relAbun <- plot.lacto.relAbun + scale_fill_brewer("Genera", palette = "Paired") + theme_bw()
plot.lacto.relAbun <- plot.lacto.relAbun + theme(axis.text.x = element_text(angle = 90))
plot.lacto.relAbun <- plot.lacto.relAbun + guide_italics + theme(legend.title = element_text(size = 18)

print(plot.lacto.relAbun)
```



```
# grab data from rel abundance bar plot
data.lacto <- plot.lacto.relAbun$data
# check column names
colnames(data.lacto)
```

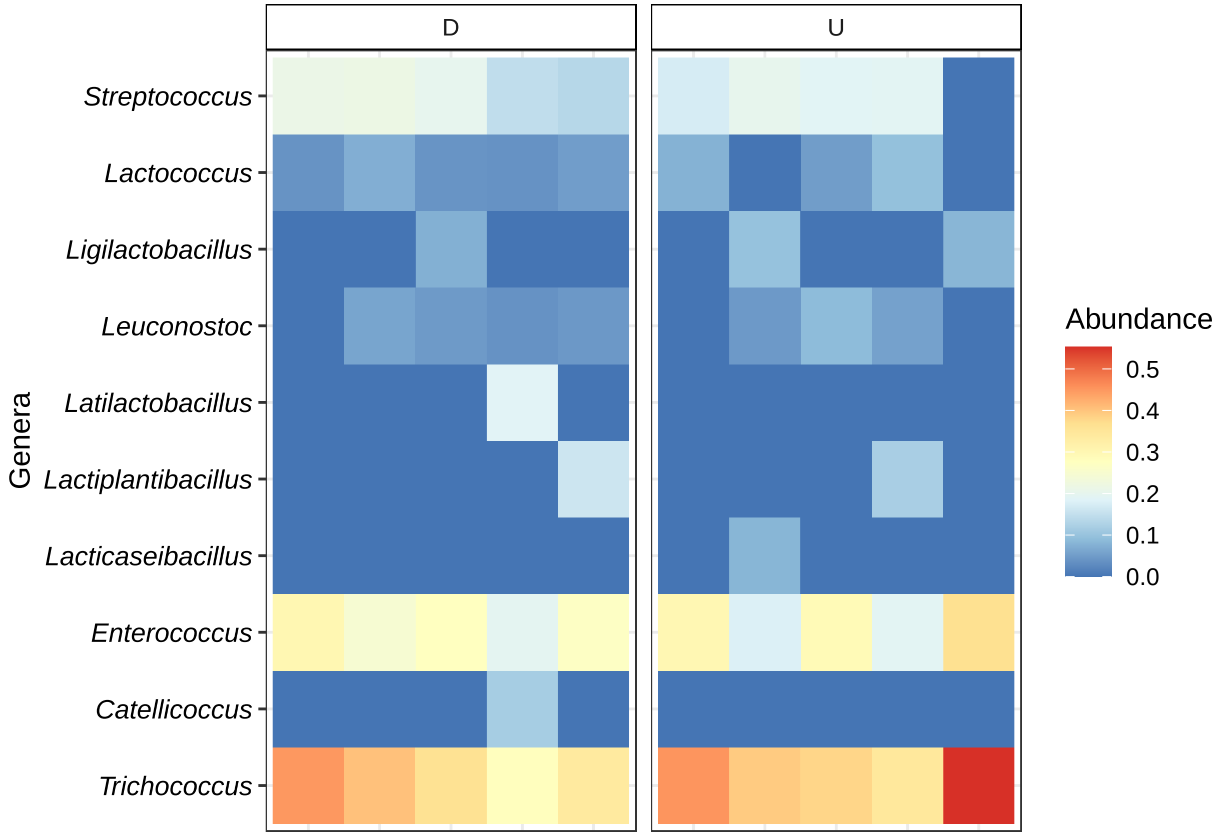
Lactobacillales Relative Abundance Heat Map

```
## [1] "Tax"       "Sample"    "Abundance" "xlabel"
```

```
# base plot
lacto.heat <- ggplot(data.lacto, aes(x = Sample, y = Tax)) + geom_tile(aes(fill = Abundance))
# choose palette and theme
lacto.heat <- lacto.heat + scale_fill_distiller("Abundance", palette = "RdYlBu") + theme_bw()
# taxonomic name to italics
lacto.heat <- lacto.heat + theme(axis.text.y = element_text(colour = 'black',
                                                          size = 10,
                                                          face = 'italic'))
# make separate samples based on y-variable
lacto.heat <- lacto.heat + facet_grid(~xlabel,
                                      scales = "free") + rremove("x.text")
lacto.heat <- lacto.heat + ylab("Genera")

# clean up plot
```

```
lacto.heat <- lacto.heat + theme(axis.title.x=element_blank(),
                         axis.text.x=element_blank(),
                         axis.ticks.x=element_blank())
lacto.heat <-lacto.heat + theme(legend.key = element_blank(),
                         strip.background = element_rect(colour="black", fill="white"))
print(lacto.heat)
```



```
# create new object with Enterobacteriaceae subset from before
entero <- tax2

# set palette and other color formatting
taxic3 <- as.data.frame(entero@tax_table)
colorCount = length(unique(taxic3$Genera))
getPalette = colorRampPalette(brewer.pal(12, "Paired"))
# add OTUs
taxic3$OTU <- rownames(taxic3)
# check column names
colnames(taxic3)
```

Enterobacteriacaea Relative Abundance Bar Plot

```
## [1] "Kingdom" "Phylum"  "Class"   "Order"   "Family"  "Genus"   "Species"
## [8] "OTU"
```

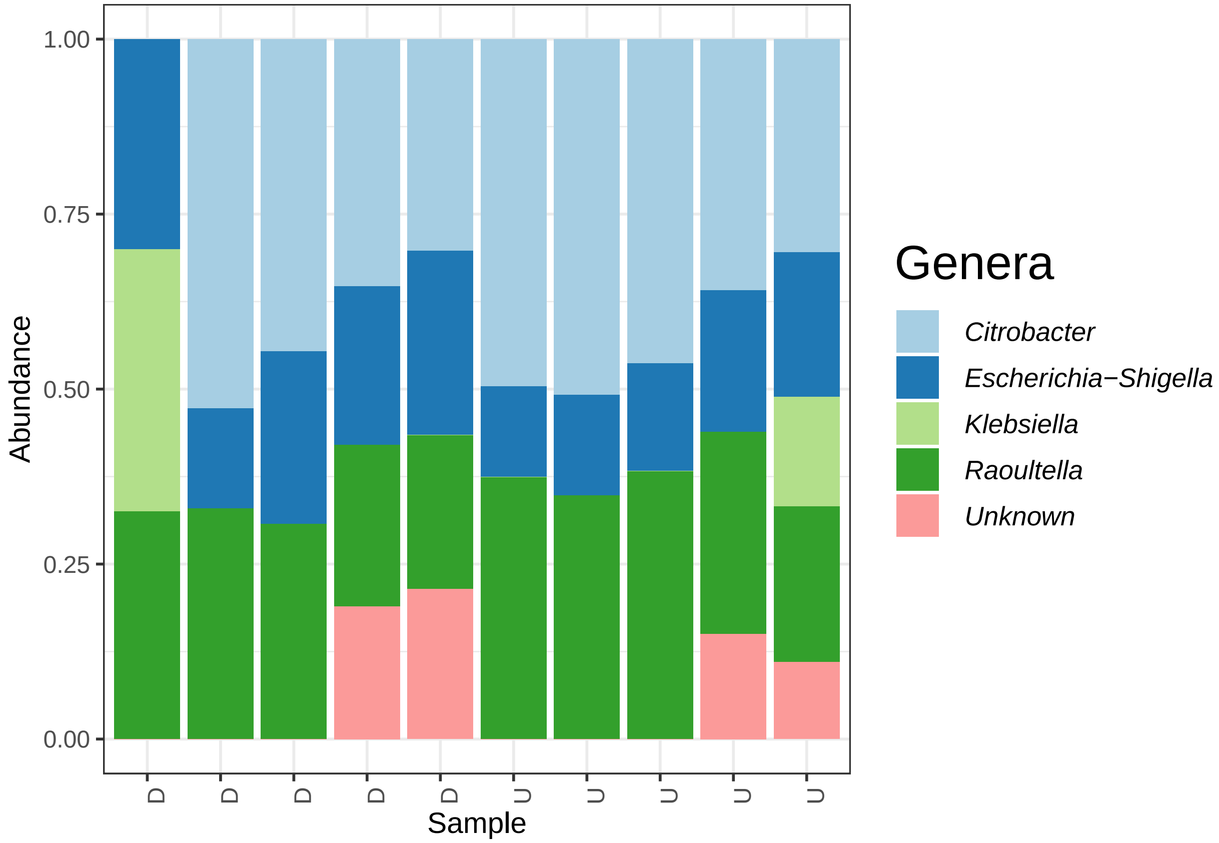
```r
# edit unclassified genera
tax_table(entero)[tax_table(entero)[, "Genus"] == "", "Genus"] <- "Unclassified genus"

# taxonomic names in italic
guide_italics <- guides(fill = guide_legend(label.theme = element_text(
  size = 10,
  face = "italic", colour = "Black", angle = 0
)))

# plot at genus level
# remove phy tree
entero@phy_tree <- NULL

# merge at genus level
entero.gen <- microbiome::aggregate_taxa(entero, "Genus")
# make relative abundance
entero.gen.rel <- microbiome::transform(entero.gen, "compositional")
# plot!
plot.entero.relAbun <- plot_composition(entero.gen.rel,
                                         sample.sort = "Sample",
                                         x.label = "Site")
plot.entero.relAbun <- plot.entero.relAbun + theme(legend.position = "bottom")
plot.entero.relAbun <- plot.entero.relAbun + scale_fill_brewer("Genera", palette = "Paired") + theme_bw
plot.entero.relAbun <- plot.entero.relAbun + theme(axis.text.x = element_text(angle = 90))
plot.entero.relAbun <- plot.entero.relAbun + guide_italics + theme(legend.title = element_text(size = 18

print(plot.entero.relAbun)
```



```
# grab data from bar plot
data.entero <- plot.entero.relAbun$data
# check column names
colnames(data.entero)
```

Enterobacteriaceae Relative Abundance Heat Map

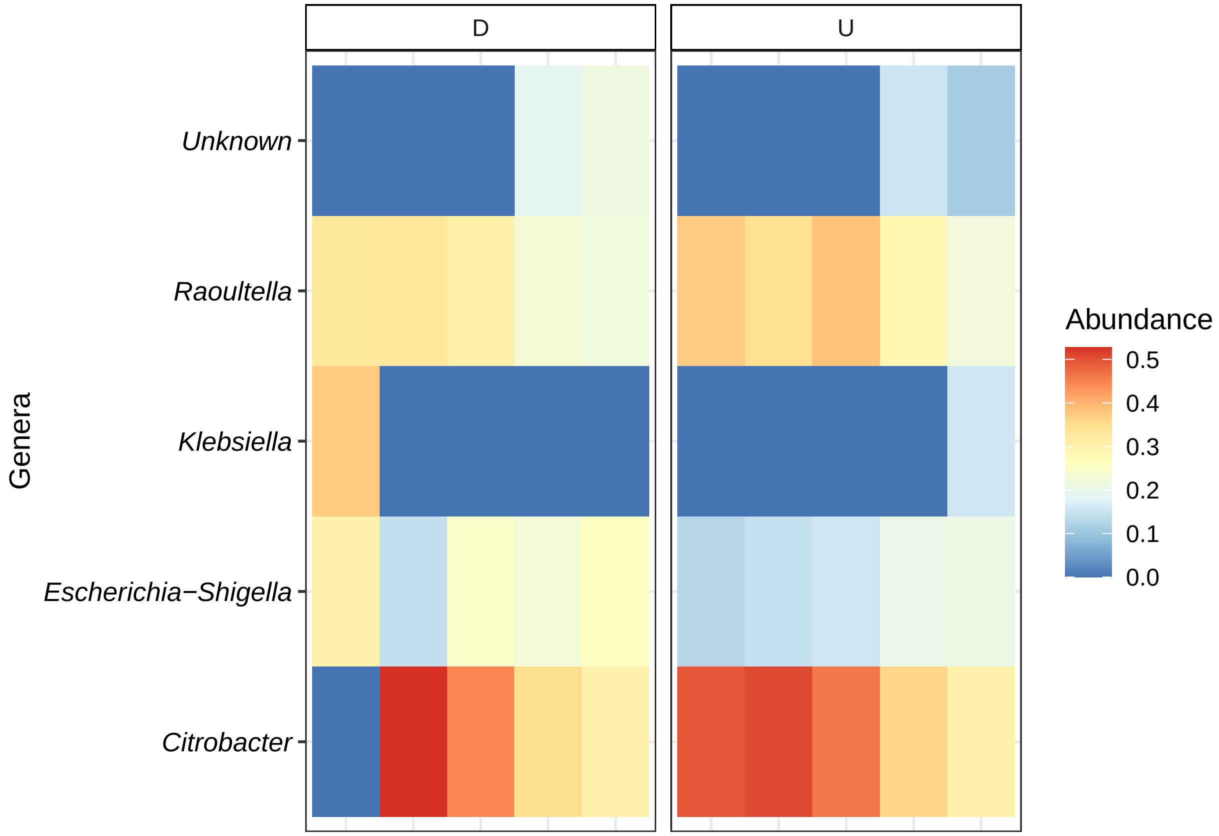
```
## [1] "Tax"       "Sample"    "Abundance" "xlabel"
```

```
# base plot
entero.heat <- ggplot(data.entero, aes(x = Sample, y = Tax)) + geom_tile(aes(fill = Abundance))
# choose palette and theme
entero.heat <- entero.heat + scale_fill_distiller("Abundance", palette = "RdYlBu") + theme_bw()
# taxonomic names in italic
entero.heat <- entero.heat + theme(axis.text.y = element_text(colour = 'black',
                                                   size = 10,
                                                   face = 'italic'))
# make separate samples based on main v-varaible
entero.heat <- entero.heat + facet_grid(~xlabel,
                                        scales = "free") + rremove("x.text")
entero.heat <- entero.heat + ylab("Genera")

# clean up plot
```

```
entero.heat <- entero.heat + theme(axis.title.x=element_blank(),
                                   axis.text.x=element_blank(),
                                   axis.ticks.x=element_blank())
entero.heat <- entero.heat + theme(legend.key = element_blank(),
                                   strip.background = element_rect(colour="black", fill="white"))

print(entero.heat)
```



And with relative abundance we can observe that *Citrobacter* was usually the dominant genus within the *Enterobacteriaceae* family!

Conclusion & Future Work

Bioinformatics and next-generation sequencing technology are incredible resources for the field of environmental metagenomics! Being able to detect bacterial abundance and evaluate bacterial diversity of different environments is an incredible ability. But, like all other research, amplicon sequencing data needs to be visualized in a way that gives it meaning. The biggest lesson I learned from my work is that expression and communication of findings is the most important part of any research. Going forward, my goal is to continue improving upon the visualizations I've made this semester, and to keep finding ways to present data in a way that is meaningful, accessible, and beautiful.

Sources

Boehm AB, Sassoubre LM. 2014. Enterococci as Indicators of Environmental Fecal Contamination. In: Gilmore MS, Clewell DB, Ike Y, Shankar N, editors.

Enterococci: From Commensals to Leading Causes of Drug Resistant Infection. Boston: Massachusetts Eye and Ear Infirmary. [accessed 2022 May 13]. http://www.ncbi.nlm.nih.gov/books/NBK190421/.

Byappanahalli MN, Nevers MB, Korajkic A, Staley ZR, Harwood VJ. 2012. Enterococci in the Environment. Microbiology and Molecular Biology Reviews. 76(4):685–706. doi:10.1128/MMBR.00023-12.

Childs DZ, Beckerman AP, Petchey OL. 2017. Getting Started with R, An Introduction for Biologists. Second. Oxford University Press.

Indicators: Enterococci | US EPA. [accessed 2022 May 13]. https://www.epa.gov/national-aquatic-resource-surveys/indicators-enterococci.