```
#AdditionalFile16_Dineretal2021_AmpliconAnalysis_figures

library(ggplot2)
library(phyloseq)
library(RColorBrewer)
library(dplyr)
library(corrplot)
library(vegan)
library(Hmisc)
library(pairwiseAdonis)

#Importing! Example is of 16S sequences
#read in table
otu_table16S=read.csv("Vibrio16S_OTUtable.csv", sep=",",row.names=1)
otu_table16S=as.matrix(otu_table16S)

#read in taxonomy
#separated by Kingdom, phylum, class, order, family, genus, species
taxonomy16S=read.csv("Vibrio16S_taxonomy_parsed.csv",sep=",",row.names=1)
taxonomy16S=as.matrix(taxonomy16S)

#read in metadata, fill=true accounts for the NAs in the sheet (?)
metadata=read.table(file = "DEA2020_Vibrio_Metadata.txt", row.names=1, header=TRUE, fill=TRUE)
metadata

#read in tree. Also generated new tree below using ape.
phy_tree16S=read_tree("Vibrio16S_rooted_tree.nwk")

#IMPORT as phyloseq objects:
OTU16S=otu_table(otu_table16S,taxa_are_rows=TRUE)
TAX16S=tax_table(taxonomy16S)
META16S= sample_data(metadata)
TREE16S=phy_tree16S

#confirm that OTU names and sample names/numbers are consistent across objects. If they aren't consistent,
objects will not merge into phyloseq object
sample_names(OTU16S)
sample_names(META16S)

#merge files into single phyloseq object
Amp16S = merge_phyloseq(OTU16S, TAX16S, META16S, phy_tree16S)
Amp16S
#Check the phyloseq object sets to make sure they make sense

#otu_table()   OTU Table:         [ 31928 taxa and 60 samples ]
#sample_data() Sample Data:       [ 60 samples by 12 sample variables ]
#tax_table()   Taxonomy Table:    [ 31928 taxa by 8 taxonomic ranks ]
#phy_tree()    Phylogenetic Tree: [ 31928 tips and 31836 internal nodes ]

#Filter out sequences non bacterial/archael sequences (i.e. Eukaryotic sequences, which we get in our
dataset due to the amplicon region, mitochondrial sequences, and chloroplast sequences)
#For 18S and HSP60, no filtering in phyloseq.
Amp16S.euks <- subset_taxa(Amp16S, Kingdom=="Eukaryota")
Amp16S.noeuk <- subset_taxa(Amp16S, Kingdom!="Eukaryota")
Amp16S.chlor <- subset_taxa(Amp16S.noeuk, Order=="Chloroplast")
Amp16S.noeuk.nochlor <- subset_taxa(Amp16S.noeuk, Order!="Chloroplast")
Amp16S.mito <-subset_taxa(Amp16S.noeuk.nochlor, Family=="Mitochondria")
Amp16S.noeuk.nochlor.nomito <- subset_taxa(Amp16S.noeuk.nochlor, Family!="Mitochondria")
Amp16S.filt <- Amp16S.noeuk.nochlor.nomito

Amp16S.filt

#Looking at Vibrio genus: 116 taxa across 60 samples
vibgenus <- subset_taxa(Amp16S.filt, Genus=="Vibrio")
vibgenus
ntaxa(vibgenus)
nsamples(vibgenus)

#Taxonomy Bar plots for whole community (by class level)

#Creating a taxonomy barplot of top classes
Amp16S.filt1 <- tax_glom(Amp16S.filt, "Class") # agglomerate taxa by class, may take awhile
Amp16S.filt2 <- transform_sample_counts(Amp16S.filt1, function(x) x/sum(x)) #get relative abundance in %
```

```
Amp16S.filt3 <- psmelt(Amp16S.filt2) # create dataframe from phyloseq object
Amp16S.filt3$Class <- as.character(Amp16S.filt3$Class) #convert to character
Amp16S.filt3$Class[Amp16S.filt3$Abundance < 0.01] <- "< 1% Abundance" #group and rename genera with < 1%
abundance

#exporting this dataframe for other uses, for example, pulling out the top X most abundant classes to
combine with metadata for correlogram (done manually in excel)
write.csv(Amp16S.filt3, file = "Amp16S.relabund.class.csv")

#set color palette to accommodate the number of genera
colourCount = length(unique(Amp16S.filt3$Class))
getPalette = colorRampPalette(brewer.pal(6, "Set1"))

#plot 16S classes
p <- ggplot(data=Amp16S.filt3, aes(x=Month, y=Abundance, fill=Class))
p + geom_bar(aes(), stat="identity", position="stack") +
  scale_fill_manual(values=getPalette(colourCount)) +
  scale_x_discrete(name ="Month",
limits=c("Dec","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov")) +
  theme(axis.text = element_text(colour = "black", size = 10, family = "Helvetica"), axis.text.x =
element_text(angle=90), axis.title = element_text(colour = "black", size = 10, family = "Helvetica")) +
  theme(legend.position="right") + guides(fill=guide_legend(ncol=1)) +
  facet_wrap(~ Site, ncol = 5)

#Vibrio 16S composition and percent community
vibgenus <- subset_taxa(Amp16.filt, Genus=="Vibrio")
VGr  = transform_sample_counts(vibgenus, function(x) x / sum(x) )
vibgenus_VGr <- psmelt(VGr) # create dataframe from phyloseq object

#set color palette to accommodate the number of genera
colourCount = length(unique(vibgenus_VGr))
getPalette = colorRampPalette(brewer.pal(6, "Set1"))

#plot
p <- ggplot(data=vibgenus_VGr, aes(x=Month, y=Abundance, fill=Species))
p + geom_bar(aes(), stat="identity", position="stack") +
  scale_fill_manual(values=getPalette(colourCount)) +
  scale_x_discrete(name ="Month",
limits=c("Dec","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov")) +
  theme(axis.text = element_text(colour = "black", size = 6, family = "Helvetica"), axis.title =
element_text(colour = "black", size = 10, family = "Helvetica", vjust=1)) +
  facet_wrap(~ Site, ncol = 5)

#Alpha Diversity
#filter out taxa with 0 reads, keeping singletons and all other samples
Amp16S_alpha <- prune_taxa(taxa_sums(Amp16.filt) > 0, Amp16.filt)
AlphaDiv16S <- estimate_richness(Amp16S_alpha, split = TRUE, measures = NULL)
write.csv(AlphaDiv16S, file = "AlphaDiv16S.csv")
plot_richness(Amp16S_alpha, x="Month", measures=c("Observed", "Shannon")) + geom_boxplot() +
scale_x_discrete(name ="Month",
limits=c("Dec","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov"))
plot_richness(Amp16S_alpha, x="Site", measures=c("Observed", "Shannon")) + geom_boxplot() +
scale_x_discrete(name ="Site", limits=c("LPL","SDR1","SDR2","TJ1","TJ2"))
plot_richness(Amp16S_alpha, x="Month") + geom_boxplot() + scale_x_discrete(name ="Month",
limits=c("Dec","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov"))
plot_richness(Amp16S_alpha, x="Site") + geom_boxplot() + scale_x_discrete(name ="Site",
limits=c("LPL","SDR1","SDR2","TJ1","TJ2"))

#vibrio alpha diversity
vibgenus_alpha <- prune_taxa(taxa_sums(vibgenus) > 0, vibgenus)
plot_richness(vibgenus_alpha, x="Month", measures=c("Observed", "Shannon")) + geom_boxplot() +
scale_x_discrete(name ="Month",
limits=c("Dec","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov"))
plot_richness(vibgenus_alpha, x="Site", measures=c("Observed", "Shannon")) + geom_boxplot() +
scale_x_discrete(name ="Site", limits=c("LPL","SDR1","SDR2","TJ1","TJ2"))

#Testing difference between groups and influence of metadata variables
#note the .csv I am reading in contains the diversity metrics generated in estimate_richness along with the
metadata corresponding to each sample
AlphaDiv16S_meta=read.csv("AlphaDiv16S_meta.csv", sep=",",row.names=1)
AlphaDiv_obs_Kruskal16S_site <- kruskal.test(Observed ~ Site, data=AlphaDiv16S_meta)
AlphaDiv_obs_Kruskal16S_month <- kruskal.test(Observed ~ Month, data=AlphaDiv16S_meta)
AlphaDiv_shannon_Kruskal16S_site <- kruskal.test(Shannon~ Site, data=AlphaDiv16S_meta)
AlphaDiv_shannon_Kruskal16S_month <- kruskal.test(Shannon ~ Month, data=AlphaDiv16S_meta)
```

```
rbind(AlphaDiv_obs_Kruskal16S_site,
AlphaDiv_obs_Kruskal16S_month,AlphaDiv_shannon_Kruskal16S_site,AlphaDiv_shannon_Kruskal16S_month)

#visualizing spearman correlations between diversity metrics and continuous env metadata using corRplot.
#question: are any of the alpha diversity metrics correlated with the environmental variables or vibrio
abundances?
AlphaDiv16S_meta_spear=read.table("AlphaDiv16S_meta_spear.txt", sep="\t", header=TRUE)
Alpha16Spear=as.matrix(AlphaDiv16S_meta_spear[,-1])
correlations16SALphadiv <- rcorr((Alpha16Spear), type = c("spearman"))
AlphaDiv16S_Spearman_coefficients <- correlations16SALphadiv$r
AlphaDiv16S_Spearman_Pvalues <- correlations16SALphadiv$P
corrplot(AlphaDiv16S_Spearman_coefficients, method = "color", type="upper", p.mat =
AlphaDiv16S_Spearman_Pvalues, insig = "label_sig", sig.level = 0.05, pch.cex=1.5, pch.col = "black", tl.col
= "black", tl.srt = 45, tl.cex = .8)

#Beta Diversity
#filter low-occurrence, poorly-represented OTUs from the data (as in phyloseq tutorial:
https://joey711.github.io/phyloseq/preprocess.html)
Amp16S_gt3 = genefilter_sample(Amp16S.filt, filterfun_sample(function(x) x > 3),
A=0.2*nsamples(Amp16S.filt))
Amp16S.filt_beta = prune_taxa(Amp16S_gt3, Amp16S.filt)
#Transform to even sampling depth.
Amp16S.filt_beta = transform_sample_counts(Amp16S.filt_beta, function(x) 1E6 * x/sum(x))

#PCoA ordination and plots
Amp16S.PCoA <- ordinate(Amp16S.filt_beta, "PCoA", "bray")
plot_ordination(Amp16S.filt_beta, Amp16S.PCoA, shape="Site", color="Month") + geom_point(size=5) +
  scale_color_discrete(name ="Month",
limits=c("Dec","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov"))

#Testing similarity of groups, clustering, influences on diversity
Amp16S.filt_bray <- phyloseq::distance(Amp16S.filt_beta, method = "bray")
Amp16S.filt_bray

# make a data frame from the sample_data
sampledf <- data.frame(sample_data(Amp16S.filt_beta))

#First, Betadisper tests the dispersion of your data (distance from centroid) to confirm they are the same.
Ideally this result is insignificant, but not always

#Site: Betadisper = significant (not even dipersion)- Tukey posthoc indicates significance is between SDR2
and LPL, Adonis PERMANOVA (failed assumption, so not really relevant)
sitebeta16S <- betadisper(Amp16S.filt_bray, sampledf$Site)
sitebeta16S
plot(sitebeta16S)
boxplot(sitebeta16S)
sitebeta16Spermutest <- permutest(sitebeta16S)
sitebeta16Spermutest
sitebeta16Stukey <- TukeyHSD(sitebeta16S)
sitebeta16Stukey
siteadonis16S <- adonis(Amp16S.filt_bray ~ Site, data = sampledf)
siteadonis16S
pairwise.adonis(Amp16S.filt_bray, sampledf$Site, sim.method = "bray", p.adjust.m = "fdr")

#Month: Betadisper = insignificant (even dipersion)
monthbeta16S <- betadisper(Amp16S.filt_bray, sampledf$Month)
monthbeta16S
plot(monthbeta16S)
boxplot(monthbeta16S)
monthbeta16Spermutest <- permutest(monthbeta16S)
monthbeta16Spermutest
monthbeta16Stukey <- TukeyHSD(monthbeta16S)
monthbeta16Stukey
monthadonis16S <- adonis(Amp16S.filt_bray ~ Month, data = sampledf)
monthadonis16S
pairwise.adonis(Amp16S.filt_bray, sampledf$Month, sim.method = "bray", p.adjust.m = "fdr")

#Also conducting adonis on continuous variables, including Temperature, Salinity, and Cholorophyll A
adonis(Amp16S.filt_bray ~ Temperature, data = sampledf)
adonis(Amp16S.filt_bray ~ Salinity, data = sampledf)
adonis(Amp16S.filt_bray ~ ChlA, data = sampledf)

#top10 classes for correlogram (see below), note: manually created import file in excel
#note, other correllograms created using corrplot as well, here is a useful tutorial:
```

```
http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram
Amp16S.classtop10=read.table("SpearmanCorr_Top1016S_metadata.txt", sep="\t", header=TRUE)
Amp16Sclass10=as.matrix(Amp16S.classtop10[,-1])
cor(Amp16Sclass10, method = c("spearman"))
Amp16Sclass10spearsig=rcorr(Amp16Sclass10, type = c("spearman"))
corrplot(Amp16Sclass10spearsig$r, method = "color", type="upper", p.mat = Amp16Sclass10spearsig$P,
sig.level = 0.05, insig = "label_sig", tl.col = "black", tl.srt = 45)

#18S analyses: Same process as above, class-specific plots for diatoms and copepods are below:

#Making class-specific plots for Bacillariophyta (diatoms) and arthropods (arthropoda)
bacillario <- subset_taxa(Amp18S, Class=="Bacillariophyta")
bacillario.genus <- tax_glom(bacillario, "Genus")
bacillario.genus.rel <- transform_sample_counts(bacillario, function(x) x/sum(x))
bacillario.genus.rel.df <- psmelt(bacillario.genus.rel)
bacillario.genus.rel.df$Genus <- as.character(bacillario.genus.rel.df$Genus)
bacillario.genus.rel.df$Genus[bacillario.genus.rel.df$Abundance < 0.01] <- "< 1% Abundance"

colourCount = length(unique(bacillario.genus.rel.df$Genus))
getPalette = colorRampPalette(brewer.pal(6, "Paired"))

#plot
p <- ggplot(data=bacillario.genus.rel.df, aes(x=Month, y=Abundance, fill=Genus))
p + geom_bar(aes(), stat="identity", position="stack") +
  scale_fill_manual(values=
c("chocolate1","chartreuse2","darkgreen","darkslategrey","#4363d8","cadetblue3","coral",'#3cb44b','#e6194b',
 'darkseagreen', '#ffe119', 'azure2', '#f58231', '#911eb4', '#46f0f0', '#f032e6', '#bcf60c', '#fabebe',
'#008080', '#e6beff', '#9a6324', '#fffac8', '#800000', '#aaffc3', '#808000', '#ffd8b1', '#000075',
'#808080', '#ffffff', '#000000')) +
  scale_x_discrete(name ="Month",
limits=c("Dec","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov")) +
  theme(axis.text = element_text(colour = "black", size = 12, family = "Helvetica"), axis.text.x =
element_text(angle=90), axis.title = element_text(colour = "black", size = 10, family = "Helvetica")) +
  theme(legend.position="bottom") + guides(fill=guide_legend(nrow=6)) +
  facet_wrap(~ Site, ncol = 5)

arthropoda <- subset_taxa(Amp18S, Class=="Arthropoda")
arthropoda.genus <- tax_glom(arthropoda, "Genus")
arthropoda.genus.rel <- transform_sample_counts(arthropoda, function(x) x/sum(x))
arthropoda.genus.rel.df <- psmelt(arthropoda.genus.rel)
arthropoda.genus.rel.df$Genus <- as.character(arthropoda.genus.rel.df$Genus)
arthropoda.genus.rel.df$Genus[arthropoda.genus.rel.df$Abundance < 0.01] <- "< 1% Abundance"

#set color palette to accommodate the number of genera
colourCount = length(unique(arthropoda.genus.rel.df$Genus))
getPalette = colorRampPalette(brewer.pal(6, "Paired"))

#plot
p <- ggplot(data=arthropoda.genus.rel.df, aes(x=Month, y=Abundance, fill=Genus))
p + geom_bar(aes(), stat="identity", position="stack") +
  scale_fill_manual(values=
c("chocolate1","chartreuse2","darkgreen","black","#4363d8","cadetblue3","coral",'#3cb44b','#e6194b',
'darkseagreen', '#ffe119', 'azure2', '#f58231', '#911eb4', '#46f0f0', '#f032e6', '#bcf60c', '#fabebe',
'#008080', '#e6beff', '#9a6324', '#fffac8', '#800000', '#aaffc3', '#808000', '#ffd8b1', '#000075',
'#808080', '#ffffff', '#000000',"blue2","darkgoldenrod2","darkorchid1","deeppink")) +
  scale_x_discrete(name ="Month",
limits=c("Dec","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov")) +
  theme(axis.text = element_text(colour = "black", size = 12, family = "Helvetica"), axis.text.x =
element_text(angle=90), axis.title = element_text(colour = "black", size = 10, family = "Helvetica")) +
  theme(legend.position="bottom") + guides(fill=guide_legend(nrow=6)) +
  facet_wrap(~ Site, ncol = 5)
```