Montana Statewide 16s Analysis

Alexander Alleman

January 29, 2019

```
## [1] "2019-10-30-17-16-45"
```

```
## [1] "/home/peterslab/Alex Alleman/Statewide Microbiome Analysis/Statewide analysis"
```

```
set.seed(8765)
```

Load packages

```
library(ggplot2)
library(data.table)
library(vegan)
library(dplyr)
library(scales)
library(grid)
library(reshape2)
library(ggpubr)
library(RColorBrewer)
library(ape)
library(grid)
#library(knitr)
library(ggrepel)
library(igraph)
library(Hmisc)
library(Matrix)
library(ggnetwork)
library(intergraph)
library(parallel)
library(tinytex)
library(phyloseq)
```

Colors

```
farm_col<-(c("#8c510a", "#d8b365", "#f6e8c3", "#f5f5f5", "#c7eae5", "#5ab4ac", "#01665e"))
farm_col_dark<-brewer.pal(7, "Dark2")
farm_col_paired<-(c('#fdbf6f','#ff7f00','#b2df8a','#33a02c','#fb9a99','#e31a1c','#cab2d6','#a6cee3','#1
```

Load OTU, Taxa, and Meta data

Add OTU table with sample names on top and OTU names as row names

```r
OTU_16s<- read.delim("~/Alex Alleman/Statewide Microbiome Analysis/Statewide analysis/16s_OTUall2016_fi
head(OTU_16s)[,1:10]
```

	JZ017	JZ018	JZ019	JZ020	JZ021	JZ022	JZ023	JZ024	JZ025	JZ026
OTU2	130	167	22	27	384	105	124	86	72	204
OTU7	76	44	17	39	16	18	22	30	65	50
OTU8	789	173	184	625	190	256	79	78	367	141
OTU10	306	26	17	34	13	26	342	508	20	563
OTU11	185	233	200	411	345	266	212	258	160	231
OTU12	80	168	522	284	262	305	27	75	339	213

```r
OTU_16s <- subset(OTU_16s, select = -c(102))
```

Add taxa data from Mr. DNA column seperated into 8 columns down to the strain level. This taxa assignment was done by Mr. DNA and assigned from the database greengenes.

```r
tax_16s<- read.delim("~/Alex Alleman/Statewide Microbiome Analysis/Statewide analysis/16s_OTU_ids_2016_
head(tax_16s)[,1:8]
```

	kingdom	phylum	class	order	family	genus
OTU2	k___bacteria	p___actinobacteria	c___actinobacteria	o___gaiellales	f___gaiellaceae	g___g
OTU7	k___bacteria	p___acidobacteria	c___acidobacteriia	o___acidobacteriales	f___acidobacteriaceae	g___a
OTU8	k___bacteria	p___actinobacteria	c___actinobacteria	o___actinomycetales	f___geodermatophilaceae	g___b
OTU10	k___bacteria	p___actinobacteria	c___actinobacteria	o___actinomycetales	f___geodermatophilaceae	g___g
OTU11	k___bacteria	p___actinobacteria	c___actinobacteria	o___actinomycetales	f___mycobacteriaceae	g___m
OTU12	k___bacteria	p___actinobacteria	c___actinobacteria	o___actinomycetales	f___microbacteriaceae	g___a

Meta data set has be placed together from all the spring and summer data with excel

```r
meta<- read.delim("~/Alex Alleman/Statewide Microbiome Analysis/Statewide analysis/all_metadata_summer.
                  colClasses = c(rep('factor', 7), rep('numeric', 3), rep('factor', 4), 'numeric',
                                 rep('factor', 3), rep('numeric', 28) ) )
head(meta)[,1:5]
```

	Site	ARC	Season	Sample_dates	Pea_variety
JZ032	Kalispell	NWARC	Summer	2016-summer	Delta
JZ031	Kalispell	NWARC	Summer	2016-summer	CDC Saffron
JZ030	Kalispell	NWARC	Summer	2016-summer	AC Earlystar
JZ034	Kalispell	NWARC	Summer	2016-summer	Majoret
JZ033	Kalispell	NWARC	Summer	2016-summer	DS Admiral
JZ035	Kalispell	NWARC	Summer	2016-summer	Navarro

Removed all Havre for analysis

```
meta2 <- meta[-c(19:48),]
sapply(meta2, class)
```

```
##             Site              ARC           Season     Sample_dates
##         "factor"         "factor"         "factor"         "factor"
##      Pea_variety             Plot    season_precip        irrgation
##         "factor"         "factor"        "numeric"        "numeric"
## total_precip_irr     sample_depth             Date          Tillage
##        "numeric"         "factor"         "factor"         "factor"
##        prev_crop      grain_yield        elevation              lat
##         "factor"        "numeric"         "factor"         "factor"
##              lon   Organic_Matter Moisture_Content  Nitrate_Nitrite
##         "factor"        "numeric"        "numeric"        "numeric"
##          Ammonia    Av_Phosphorus     Av_Potassium   Sulfate_Sulfur
##        "numeric"        "numeric"        "numeric"        "numeric"
##               pH            Boron          Arsenic           Barium
##        "numeric"        "numeric"        "numeric"        "numeric"
##          Cadmium          Calcium         Chromium           Cobalt
##        "numeric"        "numeric"        "numeric"        "numeric"
##           Copper             Iron             Lead        Magnesium
##        "numeric"        "numeric"        "numeric"        "numeric"
##        Manganese       Molybdenum           Nickel       Phosphorus
##        "numeric"        "numeric"        "numeric"        "numeric"
##        Potassium           Sodium           Sulfur         Vanadium
##        "numeric"        "numeric"        "numeric"        "numeric"
##             Zinc
##        "numeric"
```

Convert to matrix

```
OTU_16s_m<-as.matrix(OTU_16s)
tax_16s_m<-as.matrix(tax_16s)
meta_m<-as.matrix(meta2)

class(OTU_16s_m)
```

```
## [1] "matrix"
```

```
class(tax_16s_m)
```

```
## [1] "matrix"
```

```
class(meta_m)
```

```
## [1] "matrix"
```

Make phyloseq object

```
OTU16s = otu_table(OTU_16s_m, taxa_are_rows = TRUE)
TAX16s = tax_table(tax_16s_m)
physeq_16s = phyloseq(OTU16s, TAX16s)
```

Get physeq info

```
physeq_16s
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 36023 taxa and 101 samples ]
## tax_table()   Taxonomy Table:    [ 36023 taxa by 8 taxonomic ranks ]
```

Add meta data to both phyoseq

```
meta_phy <- sample_data(meta2)
sample_names(meta_phy)
```

```
##  [1] "JZ032" "JZ031" "JZ030" "JZ034" "JZ033" "JZ035" "JZ040" "JZ046"
##  [9] "JZ042" "JZ044" "JZ038" "JZ036" "JZ041" "JZ047" "JZ043" "JZ045"
## [17] "JZ039" "JZ037" "JZ081" "JZ078" "JZ082" "JZ079" "JZ080" "JZ083"
## [25] "JZ105" "JZ107" "JZ103" "JZ102" "JZ106" "JZ104" "JZ084" "JZ085"
## [33] "JZ086" "JZ087" "JZ088" "JZ089" "JZ090" "JZ091" "JZ092" "JZ093"
## [41] "JZ094" "JZ095" "JZ096" "JZ097" "JZ098" "JZ099" "JZ100" "JZ101"
## [49] "JZ112" "JZ109" "JZ110" "JZ111" "JZ108" "JZ113"
```

```
physeq_16s<-merge_phyloseq(physeq_16s, meta_phy)
physeq_16s
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 36023 taxa and 54 samples ]
## sample_data() Sample Data:       [ 54 samples by 45 sample variables ]
## tax_table()   Taxonomy Table:    [ 36023 taxa by 8 taxonomic ranks ]
```

Make sure the samples match 54-54, 36023-36023, and 45 measured variables

Rarefy data

```
physeq_16s<-rarefy_even_depth(physeq_16s)
```

```
## You set `rngseed` to FALSE. Make sure you've set & recorded
##  the random seed of your session for reproducibility.
## See `?set.seed`
```

```
## ...
```

```
## 5056OTUs were removed because they are no longer
## present in any sample after random subsampling
## ...
```

```
physeq_16s
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 30967 taxa and 54 samples ]
## sample_data() Sample Data:       [ 54 samples by 45 sample variables ]
## tax_table()   Taxonomy Table:    [ 30967 taxa by 8 taxonomic ranks ]
```

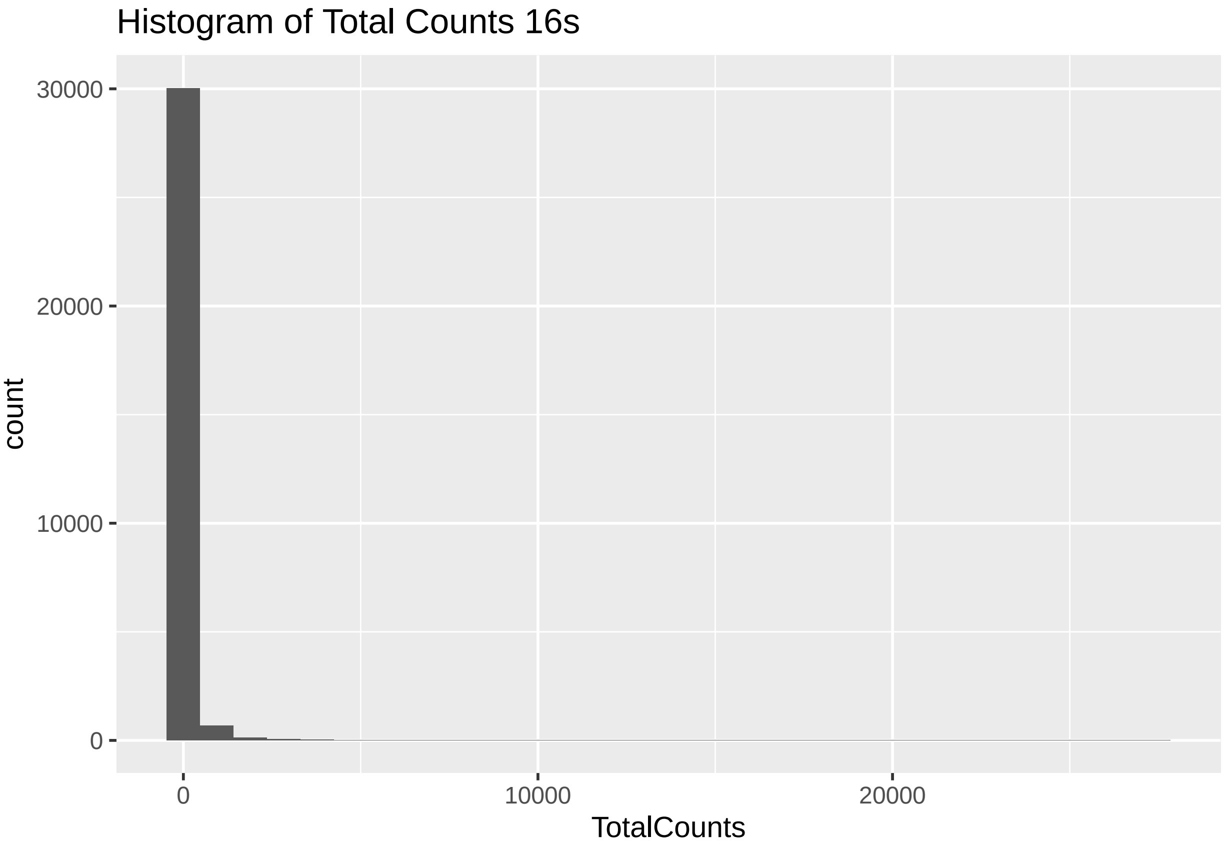
Trim data

Trim data to exclude OTUs that are not in any samples

Source for trimming code: http://evomics.org/wp-content/uploads/2016/01/phyloseq-Lab-01-Answers.html#taxa-total-counts-histogram

```
tdt_16s = data.table(tax_table(physeq_16s),
                 TotalCounts = taxa_sums(physeq_16s),
                 OTU = taxa_names(physeq_16s))
ggplot(tdt_16s, aes(TotalCounts)) +
  geom_histogram() +
  ggtitle("Histogram of Total Counts 16s")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



How many OTUS have low count (Rare)?

```
tdt_16s[(TotalCounts <= 0), .N]#zero count
```

```
## [1] 0
```

```
tdt_16s[(TotalCounts <= 1), .N]#single count
```

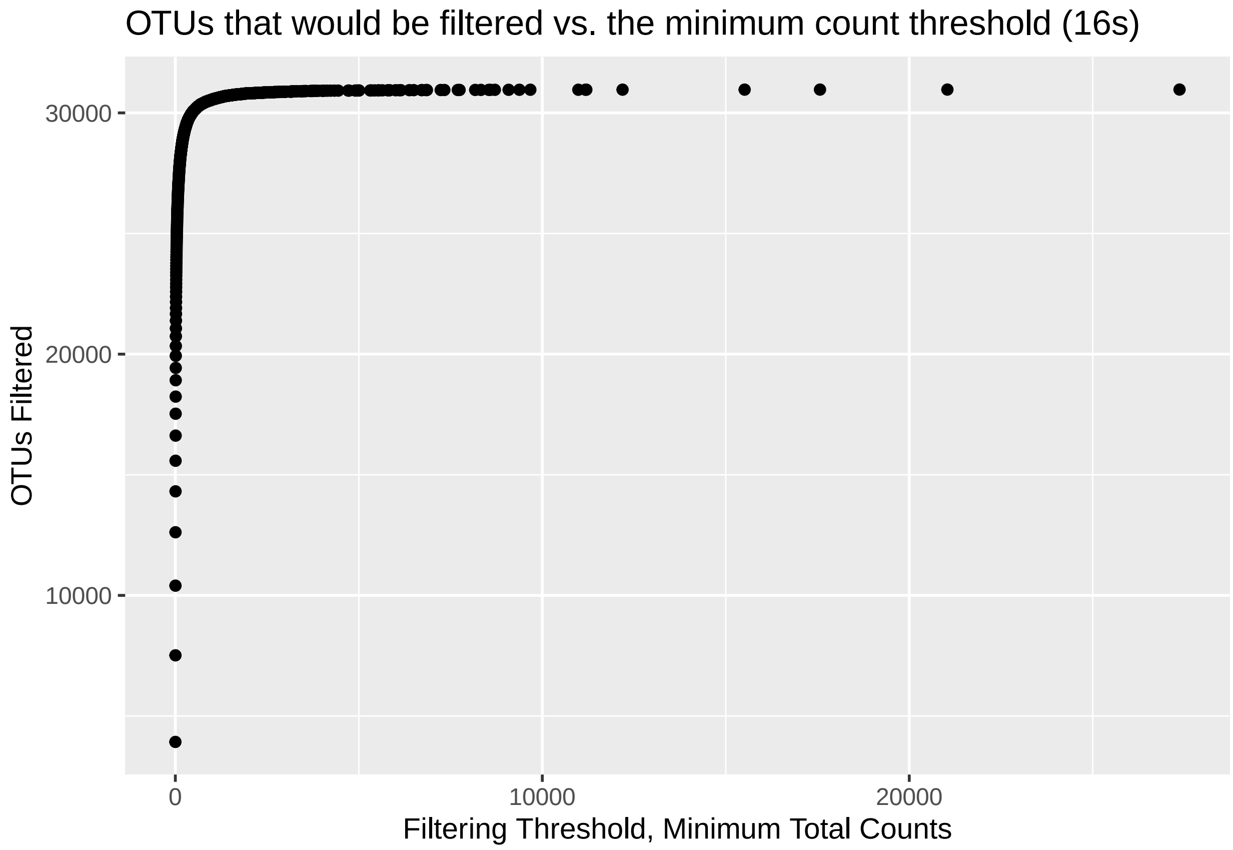
```
## [1] 3927
```

```
tdt_16s[(TotalCounts <= 2), .N]#double count
```

```
## [1] 7516
```

We have many OTUs with no counts or less than two we will trim the data to remove these OTUs. But how much to remove?

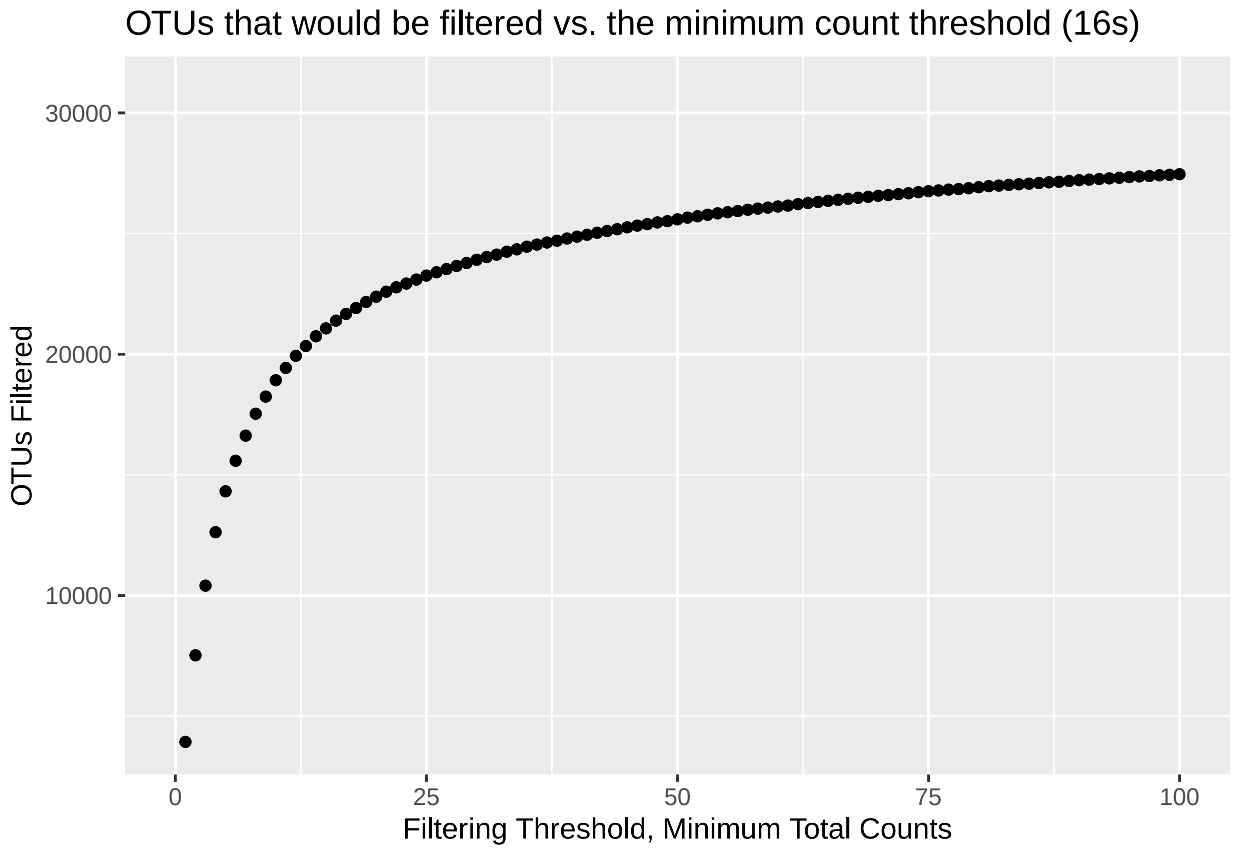
```
# taxa cumulative sum
taxcumsum_16s = tdt_16s[, .N, by = TotalCounts]
setkey(taxcumsum_16s, TotalCounts)
taxcumsum_16s[, CumSum := cumsum(N)]
# Define the plot
pCumSum_16s = ggplot(taxcumsum_16s, aes(TotalCounts, CumSum)) +
  geom_point() +
  xlab("Filtering Threshold, Minimum Total Counts") +
  ylab("OTUs Filtered") +
  ggtitle("OTUs that would be filtered vs. the minimum count threshold (16s)")
pCumSum_16s
```



Zoom-in find threshold

```
pCumSum_16s + xlim(0, 100)
```

```
## Warning: Removed 1051 rows containing missing values (geom_point).
```



Sort by prevlance (number of times an OTU is observed at least once)

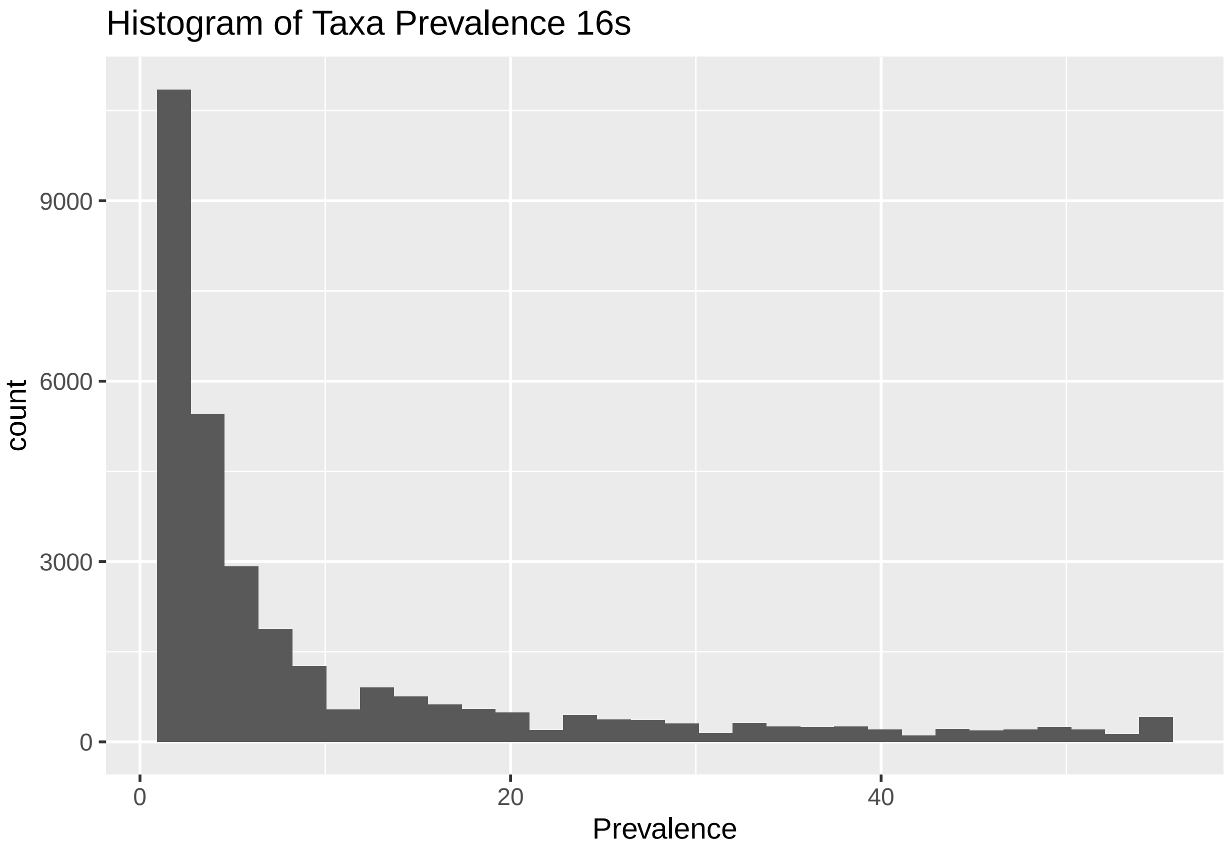
melt function for taxa prevelance filtering, Source: http://evomics.org/wp-content/uploads/2016/01/phyloseq-Lab-01-Answers.html#taxa-total-counts-histogram

fast_melt function

```
mdt_16s = fast_melt(physeq_16s)
prevdt16s = mdt_16s[, list(Prevalence = sum(count > 0),
                    TotalCounts = sum(count)),
             by = TaxaID]
```

```
ggplot(prevdt16s, aes(Prevalence)) +
  geom_histogram() +
  ggtitle("Histogram of Taxa Prevalence 16s")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# How many OTUS have low prevelance (Rare)?
prevdt16s[(Prevalence <= 0), .N]#zero
```

```
## [1] 0
```

```
prevdt16s[(Prevalence <= 1), .N]#single
```

```
## [1] 5986
```

```
prevdt16s[(Prevalence <= 2), .N]#double
```

```
## [1] 10849
```

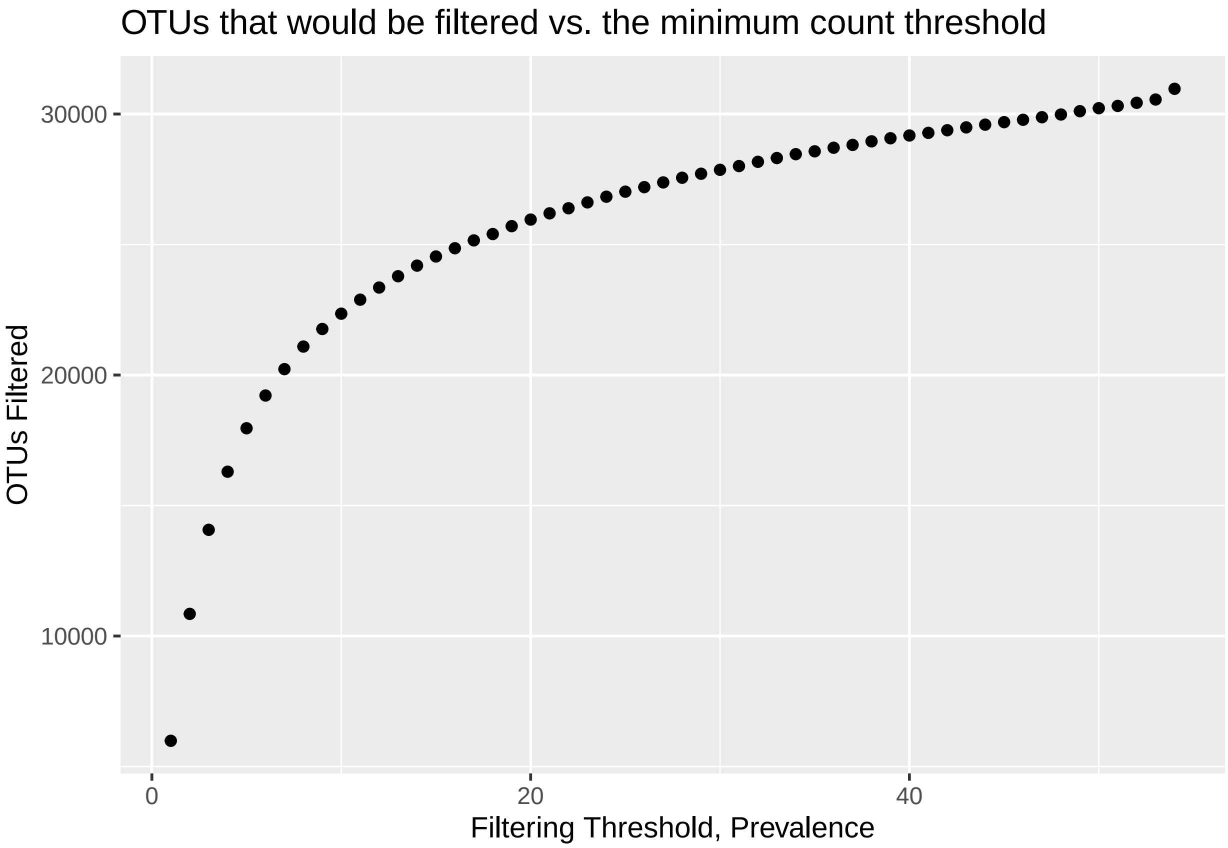
```
prevdt16s[(Prevalence >= 54), .N]#how many OTUs are in all samples
```

```
## [1] 410
```

So our samples have low abundance and low prevelance

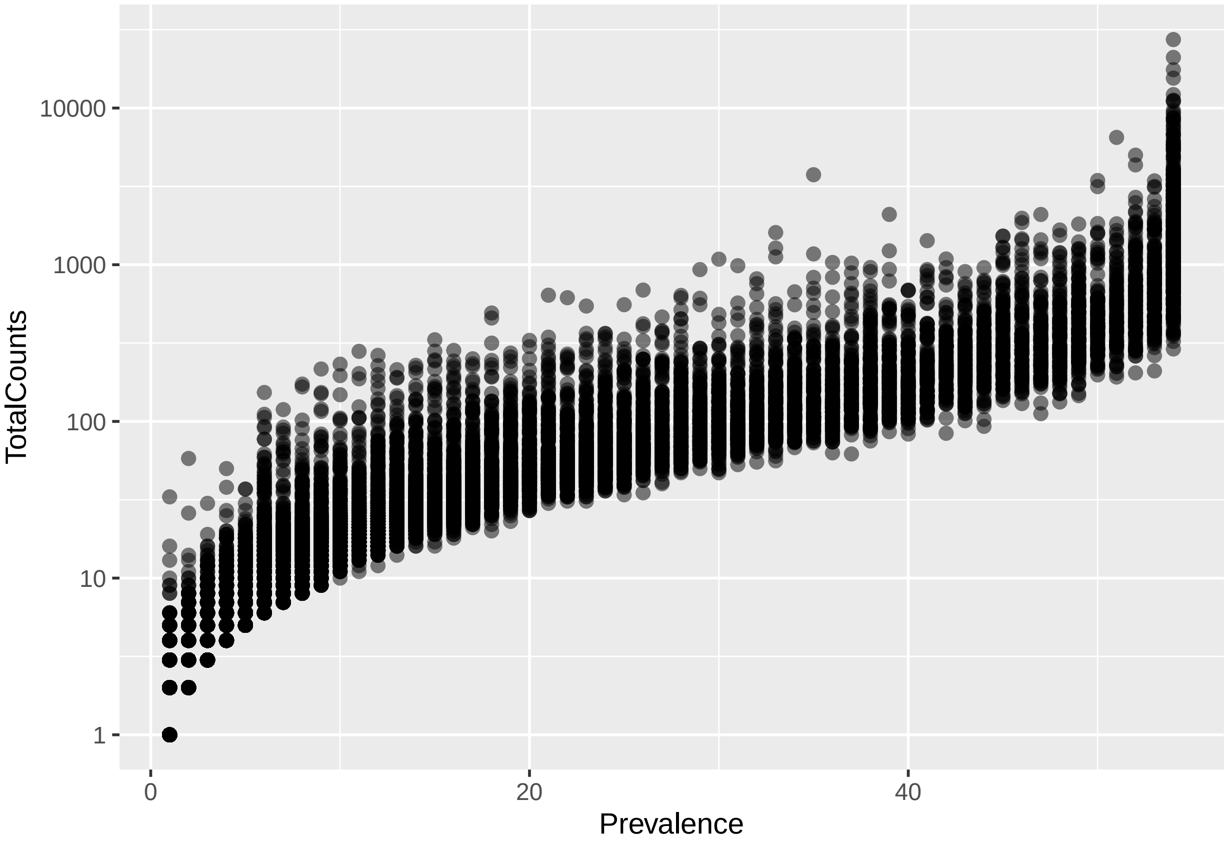
Zoomed in scater plots for prevalence

```
prevcumsum16s = prevdt16s[, .N, by = Prevalence]
setkey(prevcumsum16s, Prevalence)
prevcumsum16s[, CumSum := cumsum(N)]
pPrevCumSum = ggplot(prevcumsum16s, aes(Prevalence, CumSum)) +
  geom_point() +
  xlab("Filtering Threshold, Prevalence") +
  ylab("OTUs Filtered") +
  ggtitle("OTUs that would be filtered vs. the minimum count threshold")
pPrevCumSum
```



Prevalence vs total count scatter plot

```
ggplot(prevdt16s, aes(Prevalence, TotalCounts)) +
  geom_point(size = 2, alpha = 0.50) +
  scale_y_log10()
```



So we have a good distrubution of low abundnace and low prevlance OTUS and high abundance and high prevelance OTUs

---

Trimming

Remove less than triplets in data and prevlant in 20% of the sample

```
physeq_16s_trim = filter_taxa(physeq_16s, function(x) sum(x > 3) > (0.2*length(x)), TRUE)
physeq_16s_trim
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 3277 taxa and 54 samples ]
## sample_data() Sample Data:       [ 54 samples by 45 sample variables ]
## tax_table()   Taxonomy Table:    [ 3277 taxa by 8 taxonomic ranks ]
```

We have removed the majorty of the low abundance data with a remaining 3200 taxa which make the data analysis much more managable.

Analysis

Alpha Analysis

Bar plots

Batch all phylum that do not have more than 5% abundance in the total abundance and group together and call "<5% abundance"

```
physeq_16s_ord_1 = transform_sample_counts(physeq_16s_trim, function(x) x / sum(x) )
physeq_16s_ord_phylum <- tax_glom(physeq_16s_ord_1, "phylum")
data_16s_phylum <- psmelt(physeq_16s_ord_phylum)
data_16s_phylum$phylum<-as.character(data_16s_phylum$phylum)
data_16s_phylum$phylum[data_16s_phylum$Abundance<0.05]<-"<5% abdund"
count <- length(unique(data_16s_phylum$phylum))
count
```

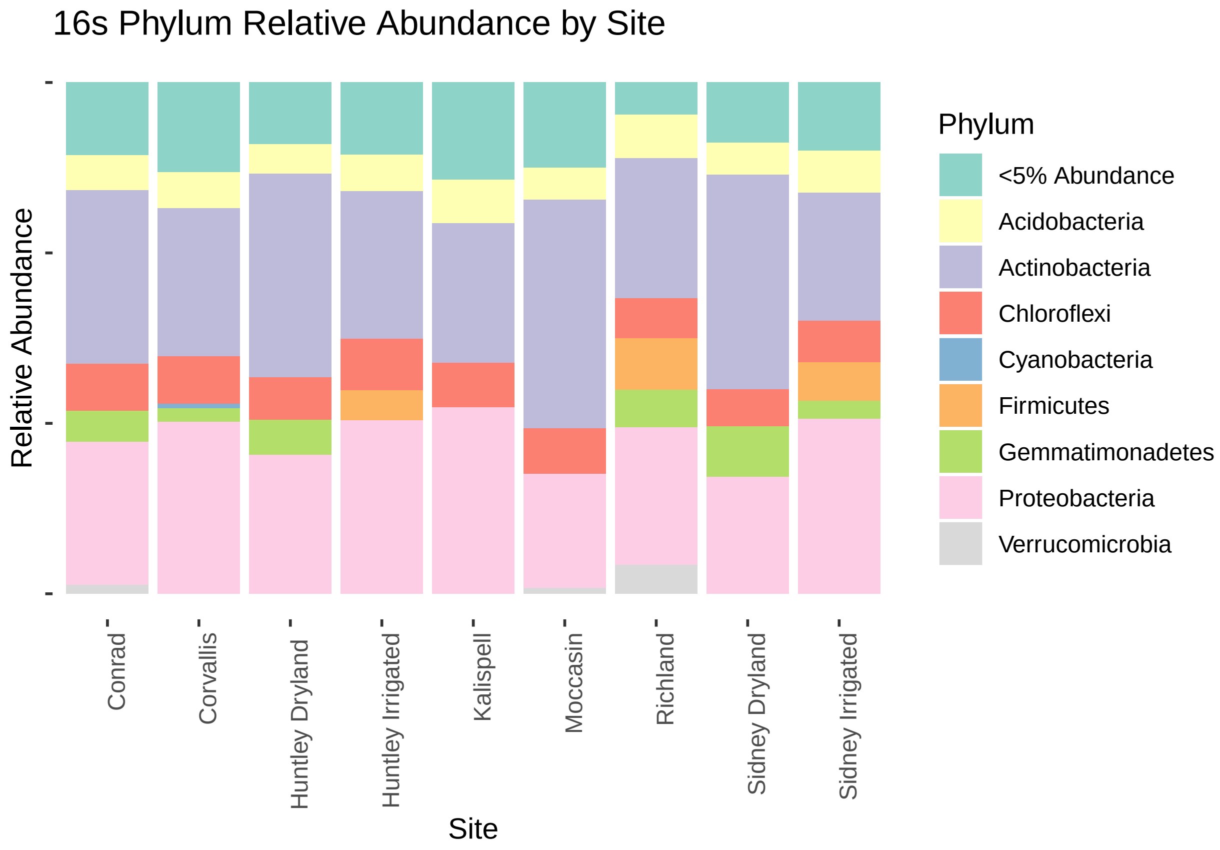
```
## [1] 9
```

```
unique(data_16s_phylum$phylum)
```

```
## [1] "p__actinobacteria"   "p__proteobacteria"   "p__firmicutes"
## [4] "p__chloroflexi"      "p__gemmatimonadetes" "p__acidobacteria"
## [7] "p__verrucomicrobia"  "p__cyanobacteria"    "<5% abdund"
```

We have 9 Phylum that are more the 5% of the total abundance in all samples this simplifies the plot to a readable format

```
ggplot(data = data_16s_phylum, aes(x = Site, y = Abundance, fill = phylum))+
geom_bar(aes(fill = phylum), stat = "identity",  position = "stack", show.legend = TRUE)+
scale_fill_manual(name = "Phylum",
                   #values=c('#a6cee3','#1f78b4','#b2df8a','#33a02c','#984ea3','#e31a1c','#fdbf6f','#ff
                  values = c('#8dd3c7','#ffffb3','#bebada','#fb8072','#80b1d3','#fdb462','#b3de69','#fc
                   breaks = c( "<5% abdund" , "p__acidobacteria", "p__actinobacteria","p__bacteroidetes
                   labels = c("<5% Abundance", "Acidobacteria", "Actinobacteria", "Bacteroidetes", "Chl
              guide = guide_legend(reverse = FALSE)
               )+
ggtitle("16s Phylum Relative Abundance by Site")+
ylab("Relative Abundance")+
#scale_x_discrete(labels = c("Conrad", "Corvallis", "Huntley Dryland", "Huntley Irrigated", "Kalispell"
  #"Moccasin", "Sidney Dryland", "Sidney Irrigated", "Richland"))+
theme(axis.text.x = element_text(angle = 90, hjust = 1), axis.text.y = element_blank(), panel.backgroun
```



There is a nice distrubution of Taxa among sites with some phyla being present or most likely in the <5% abundance portion. The taxa are dominated by Acintobacteria and Proteobacteria as expected

Publish figure as a tiff

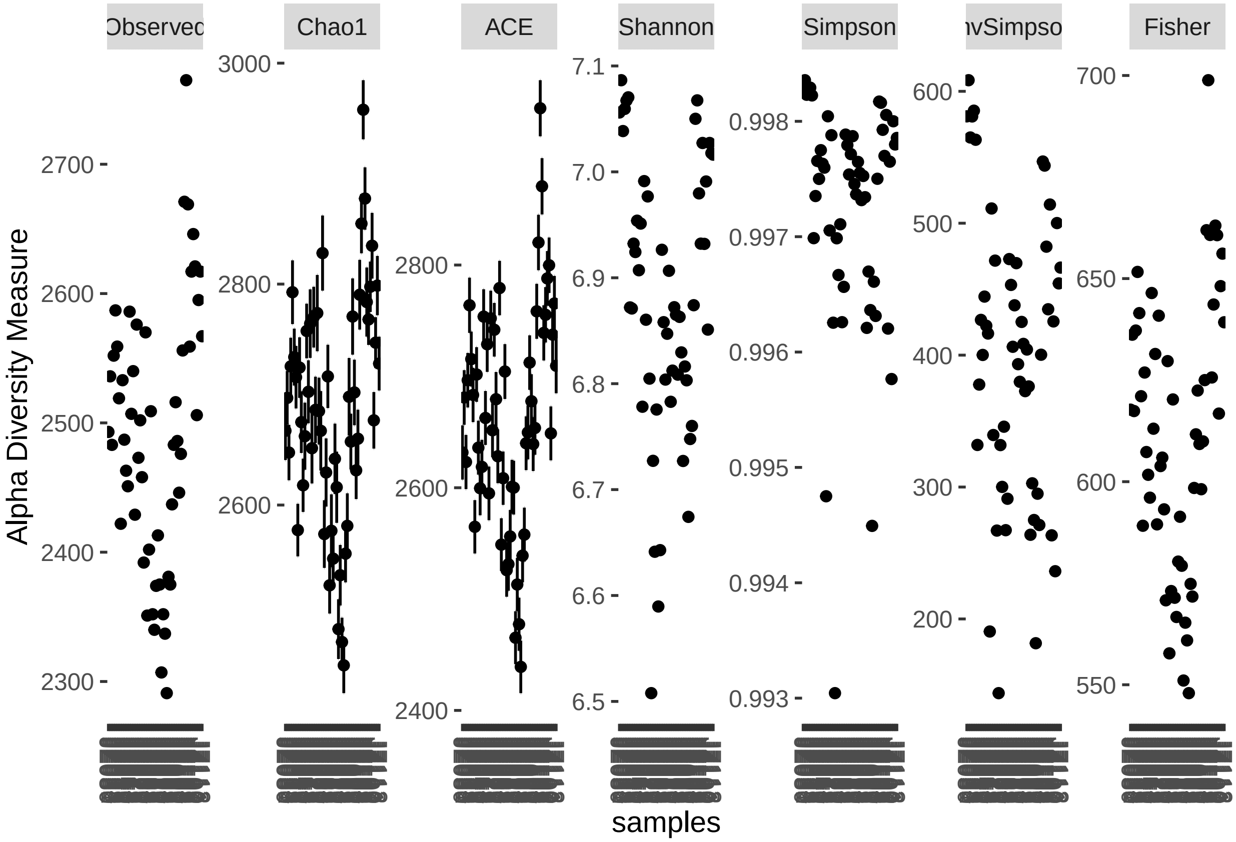
```
tiff("16s_barplot.tiff", width = 6, height = 4, units = 'in', res = 600)
ggplot(data = data_16s_phylum, aes(x = Site, y = Abundance, fill = phylum))+
geom_bar(aes(fill = phylum), stat = "identity",  position = "stack", show.legend = TRUE)+
scale_fill_manual(name = "Phylum",
                   values=c('#a6cee3','#1f78b4','#b2df8a','#33a02c','#984ea3','#e31a1c','#fdbf6f','#ff7
                   breaks = c( "<5% abdund" , "p__acidobacteria", "p__actinobacteria","p__bacteroidetes
                   labels = c("<5% Abundance", "Acidobacteria", "Actinobacteria", "Bacteroidetes", "Chl
              guide = guide_legend(reverse = FALSE)
               )+
ggtitle("16s Phylum Relative Abundance by Site")+
ylab("Relative Abundance")+
scale_x_discrete(labels = c("Conrad", "Corvallis", "Huntley Dryland", "Huntley Irrigated", "Kalispell",
theme(axis.text.x = element_text(angle = 90, hjust = 1), axis.text.y = element_blank(), panel.backgroun
dev.off()
```

```
## pdf
##   2
```

Alpha diversity metrics

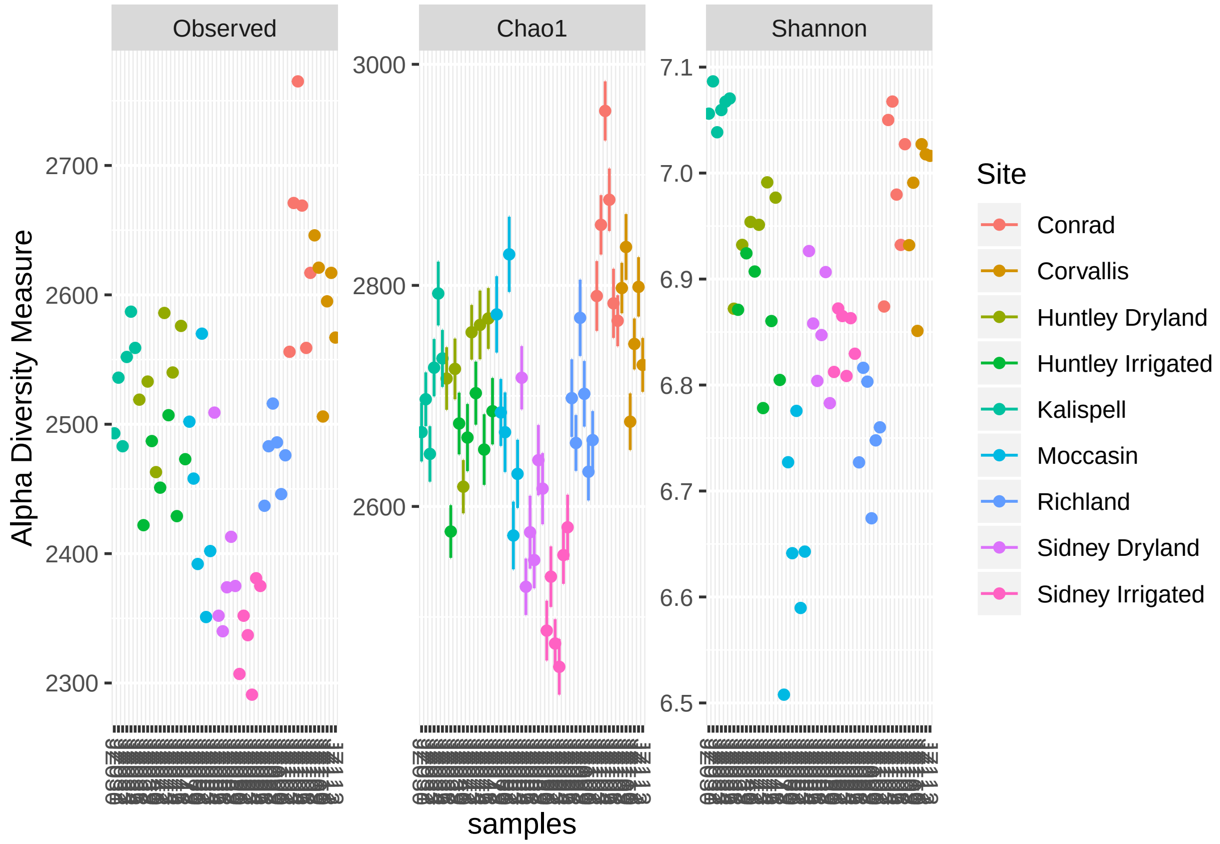
Use phyloseq internal packages to calculate the alpha diversity

```
plot_richness(physeq_16s_trim)
```



Simplify to just observed and Chao1 and Shannon

```
plot_richness(physeq_16s_trim, measures = c("Observed","Chao1", "Shannon"), color = "Site")
```



Make a table of the alpha and write table to folder

	Observed	Chao1	se.chao1	ACE	se.ACE	Shannon	Simpson	InvSimpson	Fisher
JZ030	2493	2667.224	26.49658	2631.680	24.37015	7.056065	0.9982789	581.0297	617.7272
JZ031	2536	2696.972	24.32243	2681.038	24.55532	7.086543	0.9983564	608.4244	636.1814
JZ032	2483	2647.509	25.08450	2623.104	24.34196	7.038507	0.9982301	564.9991	617.3889
JZ033	2552	2725.533	25.92421	2696.558	24.48886	7.059469	0.9982786	580.9361	637.2114
JZ034	2587	2792.640	28.84816	2763.867	24.81205	7.067370	0.9982915	585.3157	651.6376
JZ035	2559	2733.900	25.78371	2715.528	24.69299	7.070364	0.9982246	563.2674	641.4945

Just make a shannon table for further analysis

```
statewide_16s_shannon<-estimate_richness(physeq_16s_trim, split = TRUE, measures = "Shannon")

write.table(statewide_16s_shannon, file = "statewide_16s_shannon.text", sep = "\t")

head(statewide_16s_shannon)
```

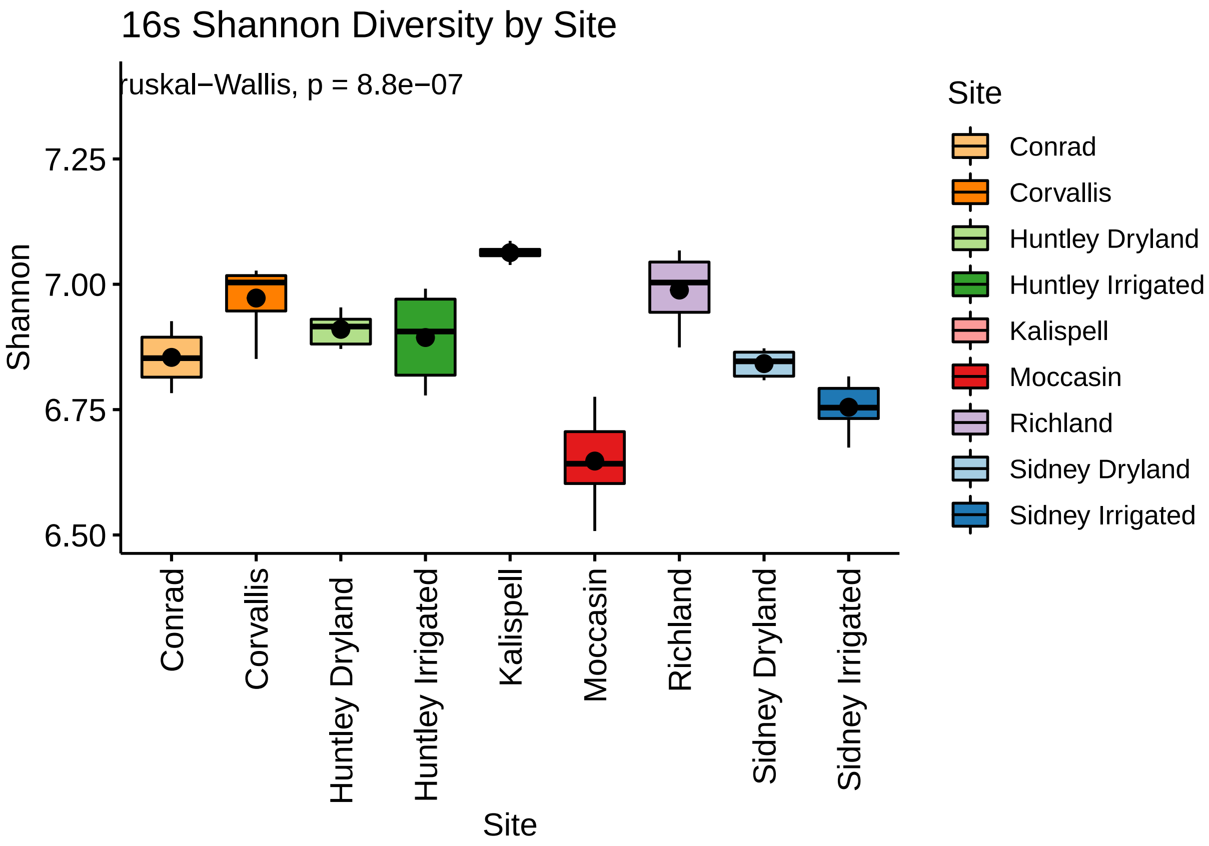
	Shannon
JZ030	7.056065
JZ031	7.086543
JZ032	7.038507
JZ033	7.059469

	Shannon
JZ034	7.067370
JZ035	7.070364

Plot Shannon diversity boxplot using ggpubr

```r
#make new DF with just shannon and site
statewide_16s_shannon$Site<-meta2$Site

#colors


#use ggpubr for plot
ggboxplot(statewide_16s_shannon, x = "Site", y = "Shannon",
   add = "mean", rug = TRUE,
   fill = "Site",
   title = "16s Shannon Diversity by Site", palette = farm_col_paired, legend = "right")+
  stat_compare_means(label.y = 7.4,  p.adjust.method = "bonferroni")+
  rotate_x_text()
```

```
## Warning: Ignoring unknown parameters: p.adjust.method
```

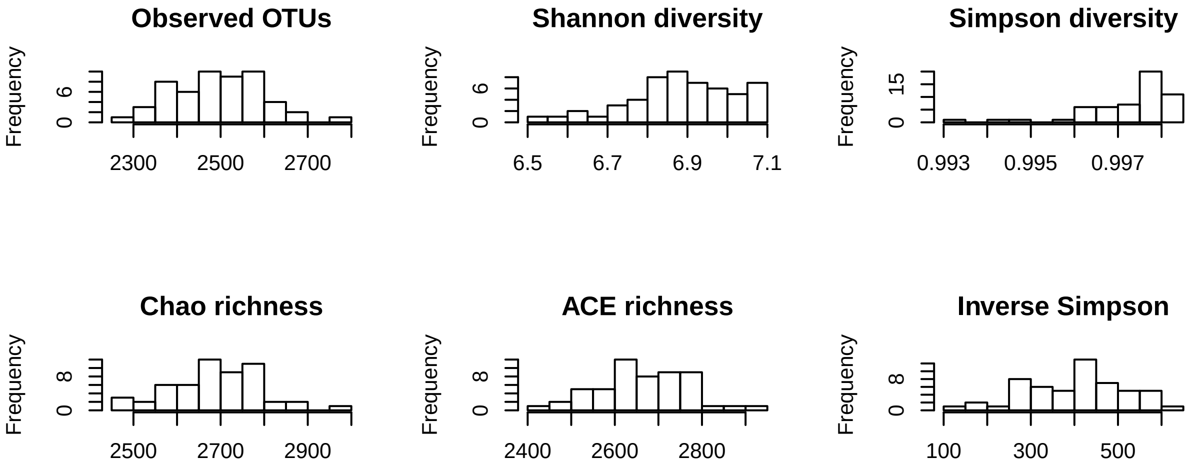


We see there is a significant difference between sites using the Kruskal-Wallis test. From the above graph we see the plots look like normal distribution with Shannon but lets check if the data is normal in all alpha metrics.

Used the following protocol

https://rpubs.com/dillmcfarlan/R_microbiotaSOP

```
#Create 2x2 plot environment so that we can see all 4 metrics at once.
par(mfrow = c(3, 3))

#Then plot each metric.
hist(rich_16s$Observed, main="Observed OTUs", xlab="", breaks=10)
hist(rich_16s$Shannon, main="Shannon diversity", xlab="", breaks=10)
hist(rich_16s$Simpson, main="Simpson diversity", xlab="", breaks=10)
hist(rich_16s$Chao1, main="Chao richness", xlab="", breaks=15)
hist(rich_16s$ACE, main="ACE richness", xlab="", breaks=15)
hist(rich_16s$InvSimpson, main="Inverse Simpson", xlab="", breaks=15)
```



Test for normalcy using the Shapiro test. The null hypothesis for this test is that the data are normally distributed, if the p-value is greater than 0.05, then the null hypothesis is not rejected.

```
shapiro.test(rich_16s$Observed)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rich_16s$Observed
## W = 0.98713, p-value = 0.8279
```

```
shapiro.test(rich_16s$Shannon)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rich_16s$Shannon
## W = 0.96413, p-value = 0.1056
```

```
shapiro.test(rich_16s$InvSimpson)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rich_16s$InvSimpson
## W = 0.98005, p-value = 0.5025
```

```
shapiro.test(rich_16s$Chao1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rich_16s$Chao1
## W = 0.99134, p-value = 0.9636
```

```
shapiro.test(rich_16s$ACE)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rich_16s$ACE
## W = 0.99177, p-value = 0.9715
```

```
shapiro.test(rich_16s$InvSimpson)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rich_16s$InvSimpson
## W = 0.98005, p-value = 0.5025
```

So we cannot reject the null for InvSimpson CHao1 and Ace because the are all normal and we can use the standard ANOVA and t-test to test our hypothesis. The Shannon index is just barley not normal but that just means we use other test like the above Kruskal-Wallis and Wilcoxon Rank sum to test the hypothesis against Shannon. But since it is so close to normalcy we will run anova with all.

Merge the meta data with the richness data and add back to the phyloseq data

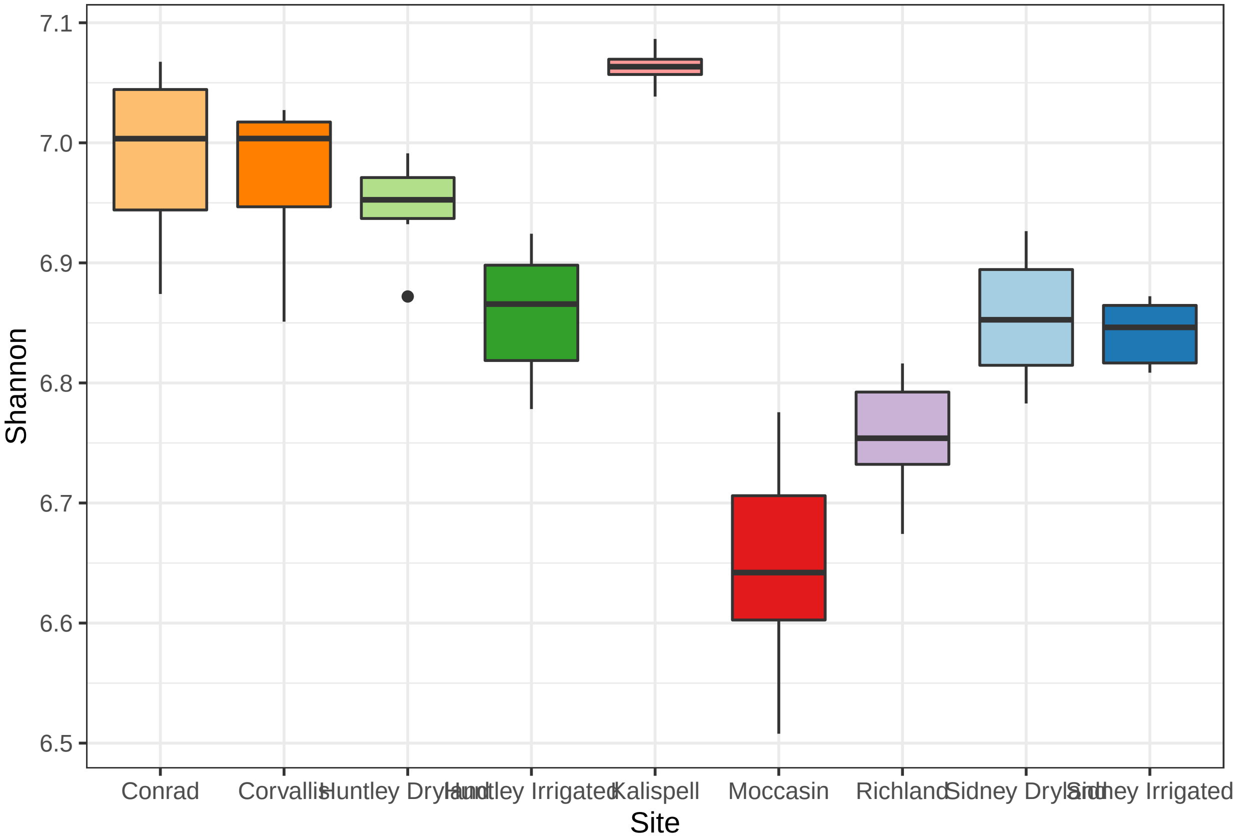
```
#First merge data sets with meta2
meta2$sample_names<-rownames(meta2)
rich_16s$sample_names<-rownames(rich_16s)
meta_16s<-merge(meta2, rich_16s, by = "sample_names")
rownames(meta_16s)<-meta_16s$sample_names
meta_16s<-meta_16s[,-1]
head(meta_16s)
```

	Site	ARC	Season	Sample_dates	Pea_variety	Plot	season_precip	irrgation	total_pr
JZ030	Kalispell	NWARC	Summer	2016-summer	AC Earlystar	Irrigated	9.33	0	
JZ031	Kalispell	NWARC	Summer	2016-summer	CDC Saffron	Irrigated	9.33	0	
JZ032	Kalispell	NWARC	Summer	2016-summer	Delta	Irrigated	9.33	0	
JZ033	Kalispell	NWARC	Summer	2016-summer	DS Admiral	Irrigated	9.33	0	
JZ034	Kalispell	NWARC	Summer	2016-summer	Majoret	Irrigated	9.33	0	
JZ035	Kalispell	NWARC	Summer	2016-summer	Navarro	Irrigated	9.33	0	

```r
mean(meta_16s$Observed)
```

```
## [1] 2490.926
```

```r
ggplot(meta_16s, aes( x= Site, y = Shannon, fill = Site)) +
  geom_boxplot(position = position_dodge(0.1),fill = farm_col_paired) +
  #geom_jitter(size = 0.5, alpha = 0.8)+
  theme_bw()
```



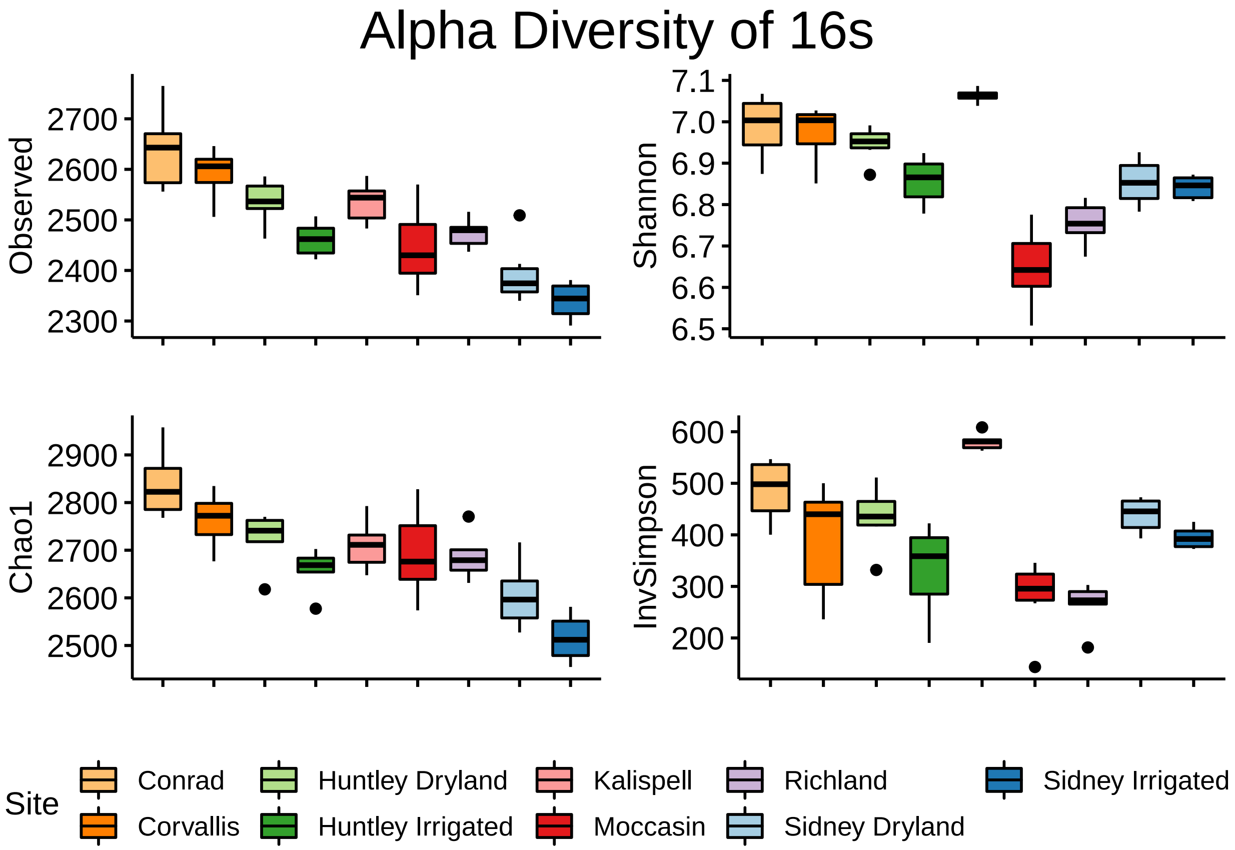
```r
shannon_16s<-ggboxplot(meta_16s, x = "Site", y = "Shannon",
   rug = TRUE,
   fill = "Site", xlab = " ", width = 0.4, title = "16s",
   palette = farm_col_paired,
   legend = "right"
   )+
  rremove("x.text")
```

```r
tiff("shannon_16s.tiff", width = 5, height = 8, units = 'in', res = 600)
shannon_16s

dev.off()
```

```
## pdf
##   2
```

Make multiple grid plot with observed, shannon simpson and chao1 diveristy

```r
#colors

#use ggpubr for plot
s16_Observ<-ggboxplot(meta_16s, x = "Site", y = "Observed",
   rug = TRUE,
   fill = "Site", xlab = " ",
   palette = farm_col_paired)+
  rremove("x.text")

s16_Shannon<-ggboxplot(meta_16s, x = "Site", y = "Shannon",
   rug = TRUE,
   fill = "Site", xlab = " ",
   palette = farm_col_paired)+
  rremove("x.text")

s16_Chao<- ggboxplot(meta_16s, x = "Site", y = "Chao1",
    rug = TRUE,
   fill = "Site", xlab = " ",
   palette = farm_col_paired)+
  rremove("x.text")

s16_InvSim<- ggboxplot(meta_16s, x = "Site", y = "InvSimpson",
   rug = TRUE,
   fill = "Site", xlab = " ",
   palette = farm_col_paired)+
  rremove("x.text")

alpha_nifH_fig<-ggarrange(s16_Observ, s16_Shannon, s16_Chao, s16_InvSim, ncol = 2, nrow = 2, common.lege

annotate_figure(alpha_nifH_fig, top = text_grob("Alpha Diversity of 16s", size = 20))
```



```
s16_Observ_stats<-ggplot_build(s16_Observ)
s16_Observ_stats$data
```

```
## [[1]]
##      fill ymin   lower middle   upper ymax outliers notchupper notchlower
## 1 #fdbf6f 2556 2573.50 2643.0 2670.50 2765            2705.568   2580.432
## 2 #ff7f00 2506 2574.00 2606.0 2620.00 2646            2635.671   2576.329
## 3 #b2df8a 2463 2522.50 2536.5 2567.00 2586            2565.204   2507.796
## 4 #33a02c 2422 2434.50 2462.0 2483.50 2507            2493.607   2430.393
## 5 #fb9a99 2483 2503.75 2544.0 2557.25 2587            2578.509   2509.491
## 6 #e31a1c 2351 2394.50 2430.0 2491.00 2570            2492.246   2367.754
## 7 #cab2d6 2437 2453.50 2479.5 2485.25 2516            2499.980   2459.020
## 8 #a6cee3 2340 2357.50 2374.5 2403.50 2413     2509   2404.171   2344.829
## 9 #1f78b4 2291 2314.50 2344.5 2369.25 2381            2379.816   2309.184
##   x PANEL group ymin_final ymax_final xmin xmax weight colour size alpha
## 1 1     1     1       2556       2765 0.65 1.35      1  black  0.5    NA
## 2 2     1     2       2506       2646 1.65 2.35      1  black  0.5    NA
## 3 3     1     3       2463       2586 2.65 3.35      1  black  0.5    NA
## 4 4     1     4       2422       2507 3.65 4.35      1  black  0.5    NA
## 5 5     1     5       2483       2587 4.65 5.35      1  black  0.5    NA
## 6 6     1     6       2351       2570 5.65 6.35      1  black  0.5    NA
## 7 7     1     7       2437       2516 6.65 7.35      1  black  0.5    NA
## 8 8     1     8       2340       2509 7.65 8.35      1  black  0.5    NA
## 9 9     1     9       2291       2381 8.65 9.35      1  black  0.5    NA
##   shape linetype
## 1    19    solid
## 2    19    solid
## 3    19    solid
## 4    19    solid
## 5    19    solid
## 6    19    solid
## 7    19    solid
## 8    19    solid
## 9    19    solid
```

```
s16_shannon_stats<-ggplot_build(s16_Shannon)
s16_shannon_stats$data
```

```
## [[1]]
##      fill     ymin    lower   middle    upper     ymax outliers notchupper
## 1 #fdbf6f 6.874101 6.944082 7.003472 7.044393 7.067491            7.068176
## 2 #ff7f00 6.851058 6.946720 7.003582 7.017327 7.027266            7.049126
## 3 #b2df8a 6.932241 6.936974 6.952528 6.971068 6.991249 6.872002   6.974520
## 4 #33a02c 6.778248 6.818674 6.865689 6.898073 6.924291            6.916903
## 5 #fb9a99 7.038507 7.056916 7.063419 7.069615 7.086543            7.071611
## 6 #e31a1c 6.507810 6.602550 6.642070 6.706073 6.775595            6.708846
## 7 #cab2d6 6.674251 6.732204 6.753930 6.792412 6.816231            6.792767
## 8 #a6cee3 6.782895 6.814687 6.852601 6.894445 6.926426            6.904048
## 9 #1f78b4 6.808538 6.816632 6.846298 6.864581 6.872218            6.877227
##   notchlower x PANEL group ymin_final ymax_final xmin xmax weight colour
## 1   6.938768 1     1     1   6.874101   7.067491 0.65 1.35      1  black
## 2   6.958039 2     1     2   6.851058   7.027266 1.65 2.35      1  black
## 3   6.930536 3     1     3   6.872002   6.991249 2.65 3.35      1  black
## 4   6.814474 4     1     4   6.778248   6.924291 3.65 4.35      1  black
## 5   7.055228 5     1     5   7.038507   7.086543 4.65 5.35      1  black
## 6   6.575295 6     1     6   6.507810   6.775595 5.65 6.35      1  black
## 7   6.715094 7     1     7   6.674251   6.816231 6.65 7.35      1  black
## 8   6.801154 8     1     8   6.782895   6.926426 7.65 8.35      1  black
## 9   6.815370 9     1     9   6.808538   6.872218 8.65 9.35      1  black
##   size alpha shape linetype
## 1  0.5    NA    19    solid
## 2  0.5    NA    19    solid
## 3  0.5    NA    19    solid
## 4  0.5    NA    19    solid
## 5  0.5    NA    19    solid
## 6  0.5    NA    19    solid
## 7  0.5    NA    19    solid
## 8  0.5    NA    19    solid
## 9  0.5    NA    19    solid
```

Save to .tiff

```
## pdf
##   2
```

Explanation of alpha diversity metrics:

Observed- total observed OTUs ***Chao1***- estimate diversity and assumes that the number of observations for a taxa has a Poisson distribution and corrects for variance ***Shannon***- # of OTUs (richness) scaled to the evenness ***Simpson-*** scale of dominance probability of any two individuals drawn at random belonging to the same species

Use ANOVA on alpha diversity metrics for main variables

Shannon first

```
aov_shannon_site_16s <- aov(Shannon ~ Site, meta_16s)
summary(aov_shannon_site_16s)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## Site         8 0.7838 0.09798   28.16 4.41e-15 ***
## Residuals   45 0.1566 0.00348                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Correct for multiple comparisons

```
shannon_16s_site<- TukeyHSD(aov_shannon_site_16s, "Site", ordered = TRUE)
shannon_16s_site
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##     factor levels have been ordered
## 
## Fit: aov(formula = Shannon ~ Site, data = meta_16s)
## 
## $Site
##                                          diff          lwr       upr
## Richland-Moccasin                  0.10736910 -0.003557796 0.2182960
## Sidney Irrigated-Moccasin          0.19440683  0.083479936 0.3053337
## Sidney Dryland-Moccasin            0.20677243  0.095845537 0.3176993
## Huntley Irrigated-Moccasin         0.21024291  0.099316017 0.3211698
## Huntley Dryland -Moccasin          0.29883563  0.187908735 0.4097625
## Corvallis-Moccasin                 0.32513913  0.214212240 0.4360660
## Conrad-Moccasin                    0.34108728  0.230160383 0.4520142
## Kalispell-Moccasin                 0.41566449  0.304737597 0.5265914
## Sidney Irrigated-Richland          0.08703773 -0.023889163 0.1979646
## Sidney Dryland-Richland            0.09940333 -0.011523561 0.2103302
## Huntley Irrigated-Richland         0.10287381 -0.008053081 0.2138007
## Huntley Dryland -Richland          0.19146653  0.080539637 0.3023934
## Corvallis-Richland                 0.21777004  0.106843142 0.3286969
## Conrad-Richland                    0.23371818  0.122791284 0.3446451
## Kalispell-Richland                 0.30829539  0.197368499 0.4192223
## Sidney Dryland-Sidney Irrigated    0.01236560 -0.098561292 0.1232925
## Huntley Irrigated-Sidney Irrigated 0.01583608 -0.095090812 0.1267630
## Huntley Dryland -Sidney Irrigated  0.10442880 -0.006498095 0.2153557
## Corvallis-Sidney Irrigated         0.13073230  0.019805411 0.2416592
## Conrad-Sidney Irrigated            0.14668045  0.035753553 0.2576073
## Kalispell-Sidney Irrigated         0.22125766  0.110330768 0.3321846
## Huntley Irrigated-Sidney Dryland   0.00347048 -0.107456414 0.1143974
## Huntley Dryland -Sidney Dryland    0.09206320 -0.018863697 0.2029901
## Corvallis-Sidney Dryland           0.11836670  0.007439809 0.2292936
## Conrad-Sidney Dryland              0.13431485  0.023387951 0.2452417
## Kalispell-Sidney Dryland           0.20889206  0.097965166 0.3198190
## Huntley Dryland -Huntley Irrigated 0.08859272 -0.022334177 0.1995196
## Corvallis-Huntley Irrigated        0.11489622  0.003969329 0.2258231
## Conrad-Huntley Irrigated           0.13084437  0.019917471 0.2417713
## Kalispell-Huntley Irrigated        0.20542158  0.094494686 0.3163485
## Corvallis-Huntley Dryland          0.02630351 -0.084623389 0.1372304
## Conrad-Huntley Dryland             0.04225165 -0.068675246 0.1531785
## Kalispell-Huntley Dryland          0.11682886  0.005901969 0.2277558
## Conrad-Corvallis                   0.01594814 -0.094978752 0.1268750
## Kalispell-Corvallis                0.09052536 -0.020401537 0.2014523
## Kalispell-Conrad                   0.07457721 -0.036349679 0.1855041
##                                        p adj
## Richland-Moccasin                  0.0646275
## Sidney Irrigated-Moccasin          0.0000282
## Sidney Dryland-Moccasin            0.0000083
## Huntley Irrigated-Moccasin         0.0000059
## Huntley Dryland -Moccasin          0.0000000
## Corvallis-Moccasin                 0.0000000
## Conrad-Moccasin                    0.0000000
## Kalispell-Moccasin                 0.0000000
## Sidney Irrigated-Richland          0.2341315
## Sidney Dryland-Richland            0.1112375
## Huntley Irrigated-Richland         0.0882958
## Huntley Dryland -Richland          0.0000376
## Corvallis-Richland                 0.0000028
## Conrad-Richland                    0.0000006
## Kalispell-Richland                 0.0000000
## Sidney Dryland-Sidney Irrigated    0.9999894
## Huntley Irrigated-Sidney Irrigated 0.9999293
## Huntley Dryland -Sidney Irrigated  0.0793879
## Corvallis-Sidney Irrigated         0.0104970
## Conrad-Sidney Irrigated            0.0026273
## Kalispell-Sidney Irrigated         0.0000020
## Huntley Irrigated-Sidney Dryland   1.0000000
## Huntley Dryland -Sidney Dryland    0.1757617
## Corvallis-Sidney Dryland           0.0285100
## Conrad-Sidney Dryland              0.0077557
## Kalispell-Sidney Dryland           0.0000067
## Huntley Dryland -Huntley Irrigated 0.2147727
## Corvallis-Huntley Irrigated        0.0372016
## Conrad-Huntley Irrigated           0.0103989
## Kalispell-Huntley Irrigated        0.0000095
## Corvallis-Huntley Dryland          0.9971100
## Conrad-Huntley Dryland             0.9426445
## Kalispell-Huntley Dryland          0.0321049
## Conrad-Corvallis                   0.9999254
## Kalispell-Corvallis                0.1923403
## Kalispell-Conrad                   0.4293812
```

not all groups have Significant difference must plot large plot with all comparisons

Write to table

irrigation

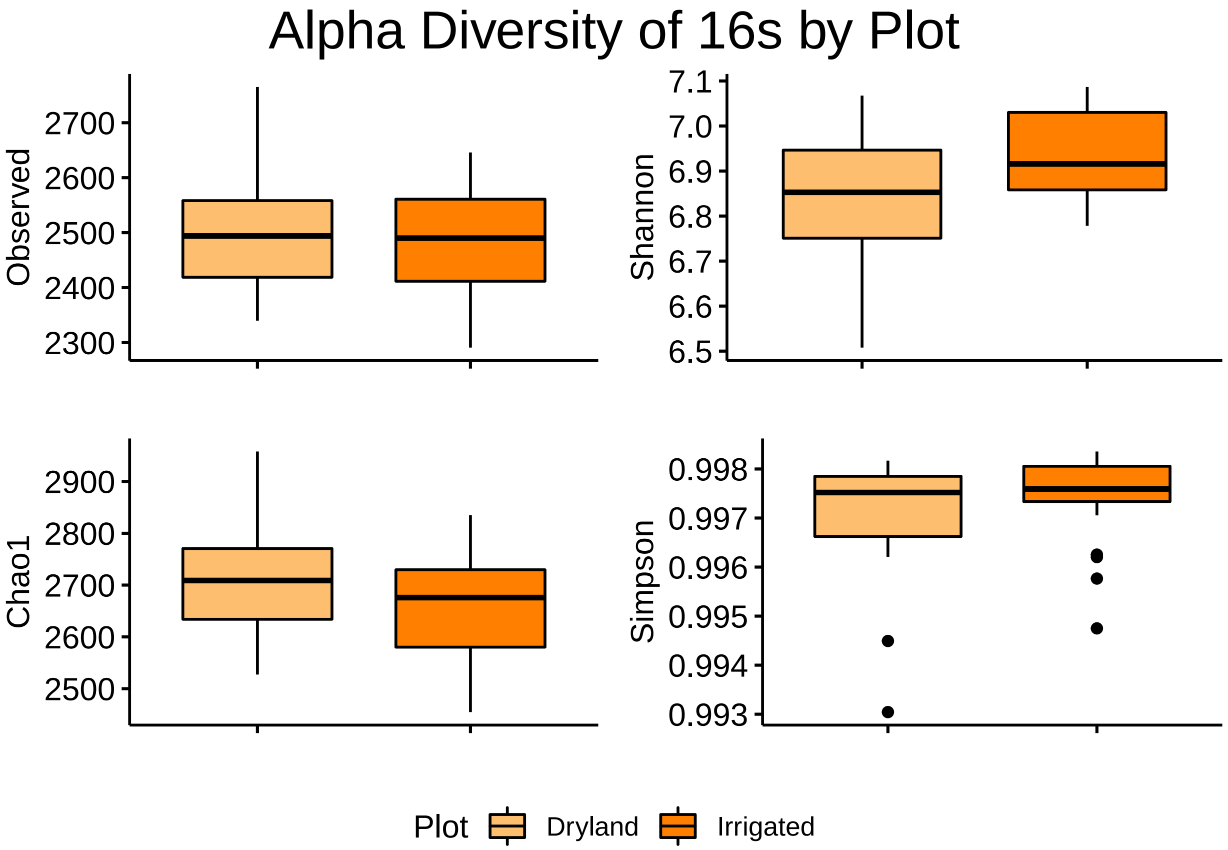
```
aov_shannon_irr <- aov(Shannon ~ Plot, meta_16s)
summary(aov_shannon_irr)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)   
## Plot         1 0.1217 0.12173   7.732 0.00754 **
## Residuals   52 0.8187 0.01574                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

irrigation is not sigficant driver of alpha diversity across the state

Plot Irrigation

```
#use ggpubr for plot
s16_Observ<-ggboxplot(meta_16s, x = "Plot", y = "Observed",
   rug = TRUE,
   fill = "Plot", xlab = " ",
   palette = farm_col_paired)+
  rremove("x.text")

s16_Shannon<-ggboxplot(meta_16s, x = "Plot", y = "Shannon",
   rug = TRUE,
   fill = "Plot", xlab = " ",
   palette = farm_col_paired)+
  rremove("x.text")

s16_Chao<- ggboxplot(meta_16s, x = "Plot", y = "Chao1",
    rug = TRUE,
   fill = "Plot", xlab = " ",
   palette = farm_col_paired)+
  rremove("x.text")

s16_InvSim<- ggboxplot(meta_16s, x = "Plot", y = "Simpson",
   rug = TRUE,
   fill = "Plot", xlab = " ",
   palette = farm_col_paired)+
  rremove("x.text")

alpha_16s_irr_fig<-ggarrange(s16_Observ, s16_Shannon, s16_Chao, s16_InvSim, ncol = 2, nrow = 2, common.l

annotate_figure(alpha_16s_irr_fig, top = text_grob("Alpha Diversity of 16s by Plot", size = 20))
```



Looks like there is significance in the other diverisities to for irrgation (Dryland vs irrigated)

```
aov_observed_irr <- aov(Observed ~ Plot, meta_16s)
summary(aov_observed_irr)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Plot         1   3252    3252   0.304  0.583
## Residuals   52 555466   10682
```

No difference in observed

```
aov_shannon_irr <- aov(Shannon ~ Plot, meta_16s)
summary(aov_shannon_irr)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Plot         1 0.1217 0.12173   7.732 0.00754 **
## Residuals   52 0.8187 0.01574
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
aov_shannon_irr_site <- aov(Shannon ~ Plot/Site, meta_16s)
summary(aov_shannon_irr_site)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## Plot          1 0.1217 0.12173   34.98 4.18e-07 ***
## Plot:Site     7 0.6621 0.09459   27.18 3.60e-14 ***
## Residuals    45 0.1566 0.00348                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
capture.output(aov_shannon_irr_site,file="aov_16s_plot_state.txt")
```

```
aov_Chao1_irr <- aov(Chao1 ~ Plot/Site, meta_16s)
summary(aov_Chao1_irr)
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)    
## Plot          1  29946   29946   7.745 0.00784 ** 
## Plot:Site     7 377343   53906  13.943 1.9e-09 ***
## Residuals    45 173984    3866                    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Just significant in chao1

```
aov_Simpson_irr <- aov(Simpson ~ Plot, meta_16s)
summary(aov_Simpson_irr)
```

```
##              Df   Sum Sq   Mean Sq F value Pr(>F)
## Plot          1 1.42e-06 1.418e-06    1.36  0.249
## Residuals    52 5.42e-05 1.042e-06               
```

Simpson is not significant

Shannon is the only metric that is significant with irrigation but it is nested within the Site so we have location effect with the alpha.

```
aov_shannon_tillage <- aov(Shannon ~ Tillage, meta_16s)
summary(aov_shannon_tillage)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)  
## Tillage       2 0.1304 0.06518   4.104 0.0223 *
## Residuals    51 0.8101 0.01588                 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
aov_shannon_site_tillage <- aov(Shannon ~ Tillage/Site, meta_16s)
summary(aov_shannon_site_tillage)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)    
## Tillage        2 0.1304 0.06518   18.73 1.21e-06 ***
## Tillage:Site   6 0.6535 0.10891   31.30 1.66e-14 ***
## Residuals     45 0.1566 0.00348                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Lets test another important facotral variable previous crop

```
aov_shannon_prevcrop <- aov(Shannon ~ prev_crop, meta_16s)
summary(aov_shannon_prevcrop)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## prev_crop    3 0.4625 0.15415   16.13 1.84e-07 ***
## Residuals   50 0.4780 0.00956                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
aov_shannon_prevcrop <- aov(Shannon ~ prev_crop/Site, meta_16s)
summary(aov_shannon_prevcrop)
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)    
## prev_crop       3 0.4625 0.15415   44.30 1.75e-13 ***
## prev_crop:Site  5 0.3214 0.06428   18.47 6.16e-10 ***
## Residuals      45 0.1566 0.00348                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Previous crop is but most likely nested within site

```
aov_shannon_site_prevcrop <- aov(Shannon ~ Site + prev_crop + Site:prev_crop, meta_16s)
summary(aov_shannon_site_prevcrop)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## Site         8 0.7838 0.09798   28.16 4.41e-15 ***
## Residuals   45 0.1566 0.00348                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see which is what we will see with most of the farm management factors that they are nested within the site variables due to experimental design.

Plot ordination

To simplfy ordination and save time we will trim the OTUs more

Remove OTUs that do not show appear more than 5 times in more than 10th of the samples

Source of Code: phyloseq protocol https://joey711.github.io/phyloseq/plot_ordination-examples.html

```
wh0 = genefilter_sample(physeq_16s_trim, filterfun_sample(function(x) x > 5), A=0.1*nsamples(physeq_16s_
physeq_16s_ord = prune_taxa(wh0, physeq_16s_trim)
physeq_16s_ord
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 3051 taxa and 54 samples ]
## sample_data() Sample Data:       [ 54 samples by 45 sample variables ]
## tax_table()   Taxonomy Table:    [ 3051 taxa by 8 taxonomic ranks ]
```

Transform to even sampling depth

```
physeq_16s_ord = transform_sample_counts(physeq_16s_ord, function(x) 1E6 * x/sum(x))
physeq_16s_ord
```

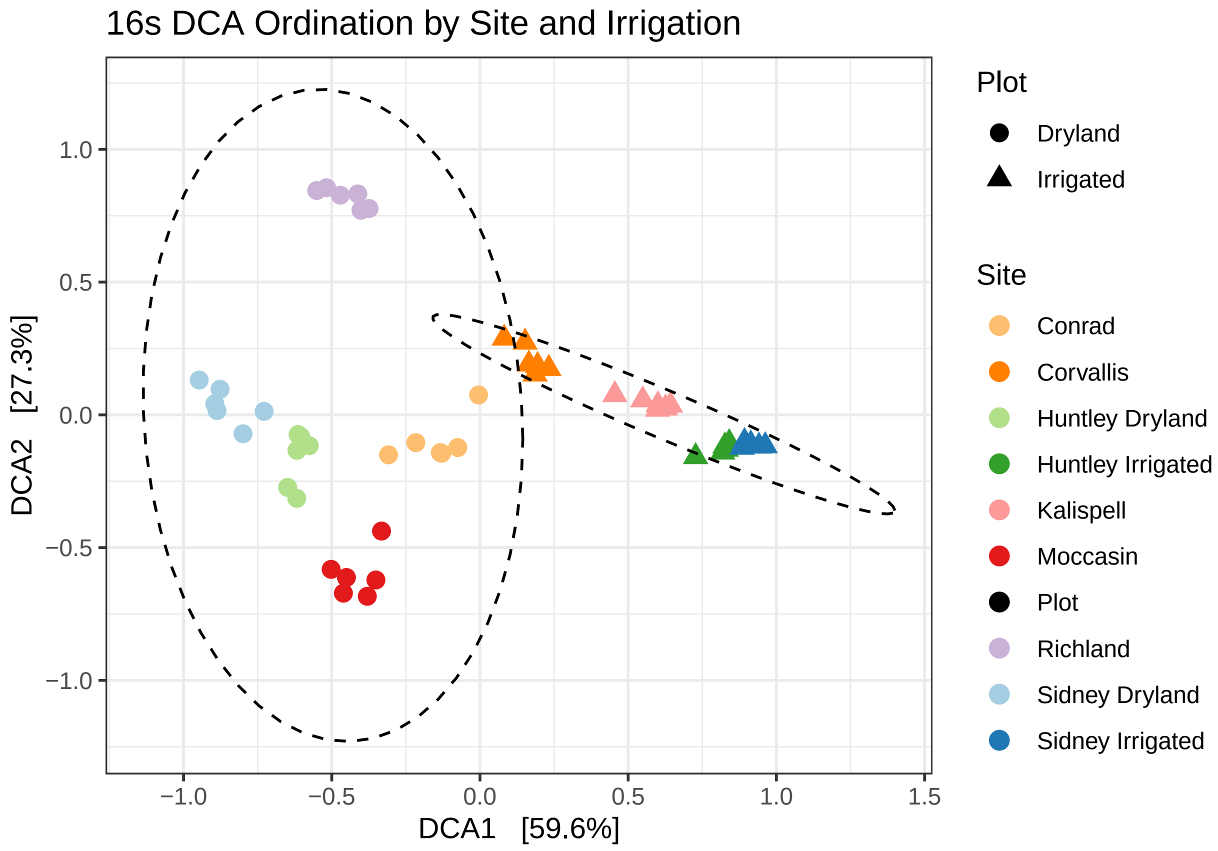
```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 3051 taxa and 54 samples ]
## sample_data() Sample Data:       [ 54 samples by 45 sample variables ]
## tax_table()   Taxonomy Table:    [ 3051 taxa by 8 taxonomic ranks ]
```

DCA Ordination

New colors:

```
farm_col_paired<-(c('#fdbf6f','#ff7f00','#b2df8a','#33a02c','#fb9a99','#e31a1c',"black",'#cab2d6','#a6c
```

```
phy16s_ord_DCA<- ordinate(physeq_16s_ord, "DCA", "bray")
plot_ordination(physeq_16s_ord, phy16s_ord_DCA, color = "Site", shape = "Plot")+
    geom_point(size = 3)+
  stat_ellipse(type = "norm", linetype = 2, aes(color ="Plot"), show.legend = F) +
  scale_color_manual(values =  farm_col_paired)+
  ggtitle("16s DCA Ordination by Site and Irrigation")+
  theme_bw()
```



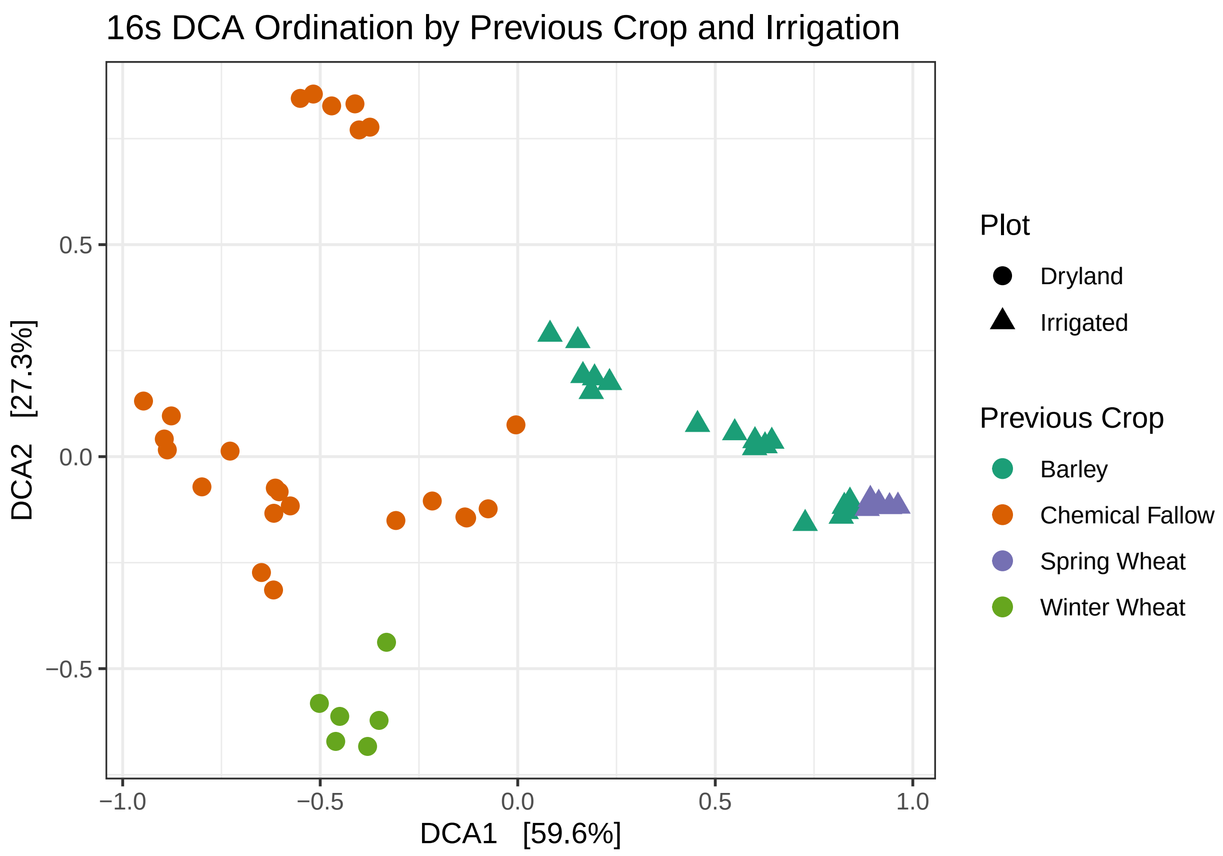
Irrigation is influencing the ordination of the principle components (DCA1 is most likely comprsied of irrigation / other farm Managment)

Make tiff

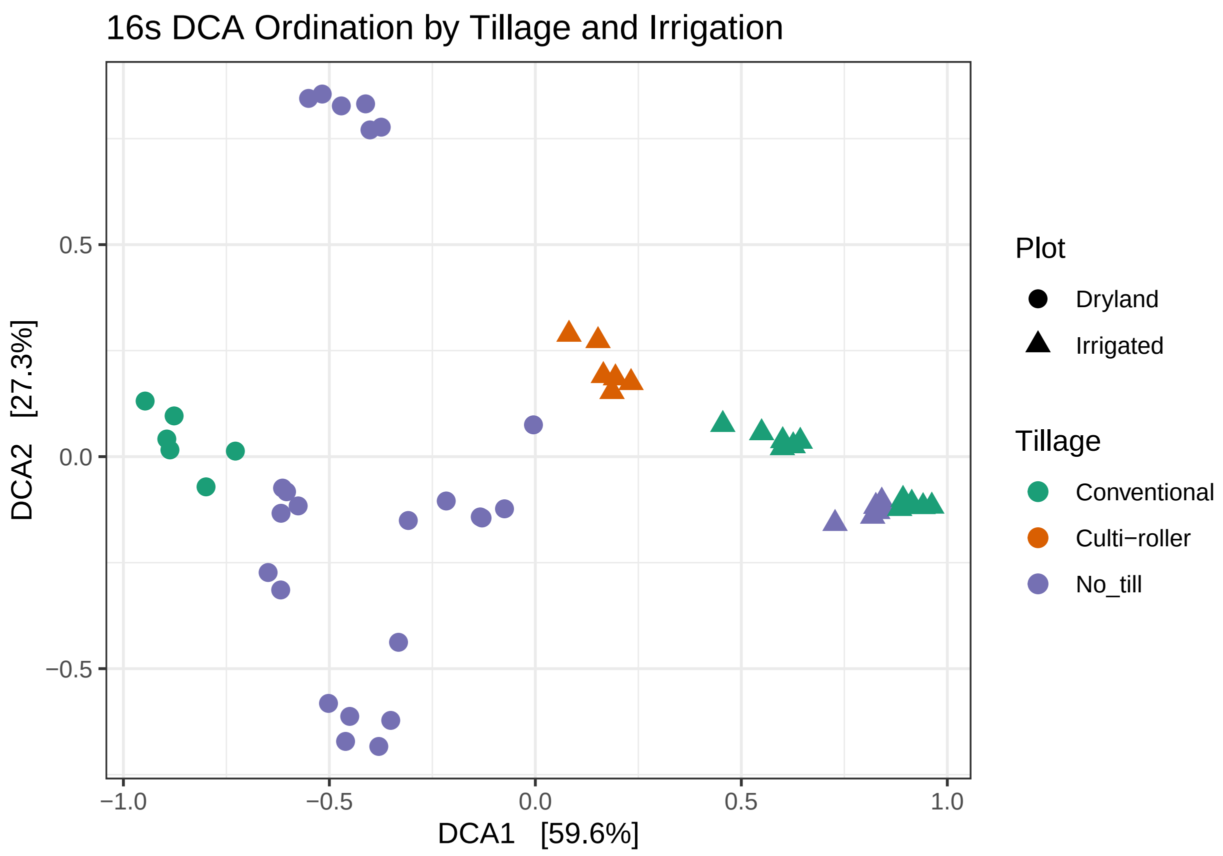
```
## pdf
##   2
```

Will color with other farm managment to see if anything is interesting.

```
phy16s_ord_DCA <- ordinate(physeq_16s_ord, "DCA", "bray")
plot_ordination(physeq_16s_ord, phy16s_ord_DCA, color = "prev_crop", shape = "Plot")+
    geom_point(size = 3)+
  scale_color_manual(values =  c("#1B9E77", "#D95F02", "#7570B3", "#66a61E"),
                     name = "Previous Crop",
                     breaks=c("barley",  "Chem_fallow", "Spring_wheat", "winter_wheat"),
                     labels=c("Barley", "Chemical Fallow", "Spring Wheat", "Winter Wheat"))+
  ggtitle("16s DCA Ordination by Previous Crop and Irrigation")+
  theme_bw()
```



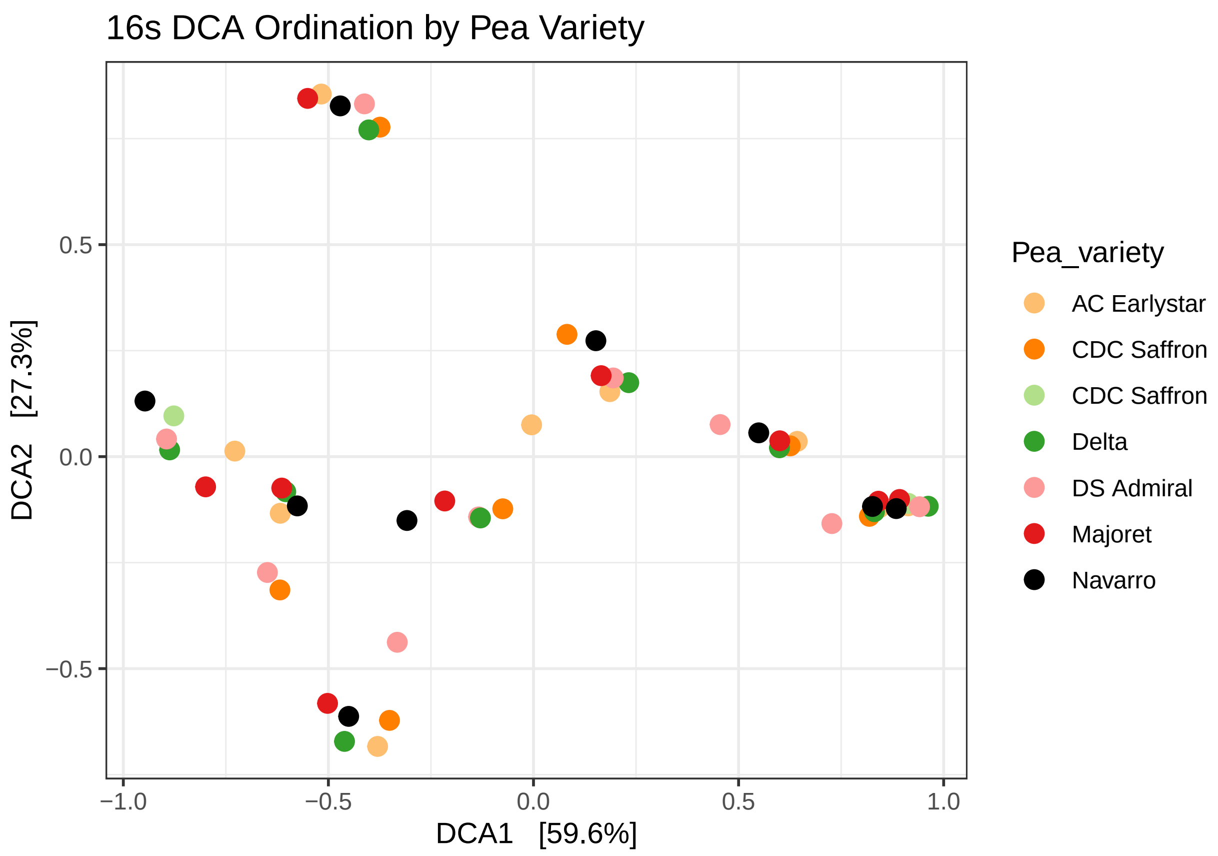
```r
phy16s_ord_DCA <- ordinate(physeq_16s_ord, "DCA", "bray")
plot_ordination(physeq_16s_ord, phy16s_ord_DCA, color = "Tillage", shape = "Plot")+
    geom_point(size = 3)+
  scale_color_manual(values =  farm_col_dark)+
  ggtitle("16s DCA Ordination by Tillage and Irrigation")+
  theme_bw()
```



Tillage has not much patterns neither compared to previous crop. The design of the study doesn't allow us to pull apart the Farm Management strategies due to the fact that many factors are overlapping (ie most Irrigated plots were prev-crop barley and most dryland plots are no till plots).

Pea variety has no correlation or ordination to bacterial community bray-curtis distance

```
plot_ordination(physeq_16s_ord, phy16s_ord_DCA, color = "Pea_variety")+
    geom_point(size = 3)+
  scale_color_manual(values =  farm_col_paired)+
  ggtitle("16s DCA Ordination by Pea Variety")+
  theme_bw()
```



DCA is good but we can also look in nonmetric multidimensional scaling

Contrast between DCA and NMDS

> DCA and NMDS are the two most popular methods for indirect gradient analysis. The reason they have remained side-by-side for so long is because, in part, they have different strengths and weaknesses… Some of the issues are relatively minor: for example, computation time is rarely an important consideration, except for the hugest data sets. Some issues are not entirely resolved: the degree to which noise affects NMDS, and the degree to which NMDS finds local rather than global options still need to be determined … Since NMDS is a distance-based method, all information about species identities is hidden once the distance matrix is created. For many, this is the biggest disadvantage of NMDS… perhaps the biggest difference between the two methods: DCA is based on an underlying model of species distributions, the unimodal model, while NMDS is not. Thus, DCA is closer to a theory of community ecology. However, NMDS may be a method of choice if species composition is determined by factors other than position along a gradient: For example, the species present on islands may have more to do with vicariance biogeography and chance extinction events than with environmental preferences – and for such a system, NMDS would be a better a priori choice. As De'ath (1999) points out, there are two classes of ordination methods - 'species composition restoration' (e.g. NMDS) and 'gradient analysis' (e.g. DCA). The choice between the methods should ultimately be governed by this philosophical distinction. - http://ordination.okstate.edu/overview.htm#Principal_Components_Analysis

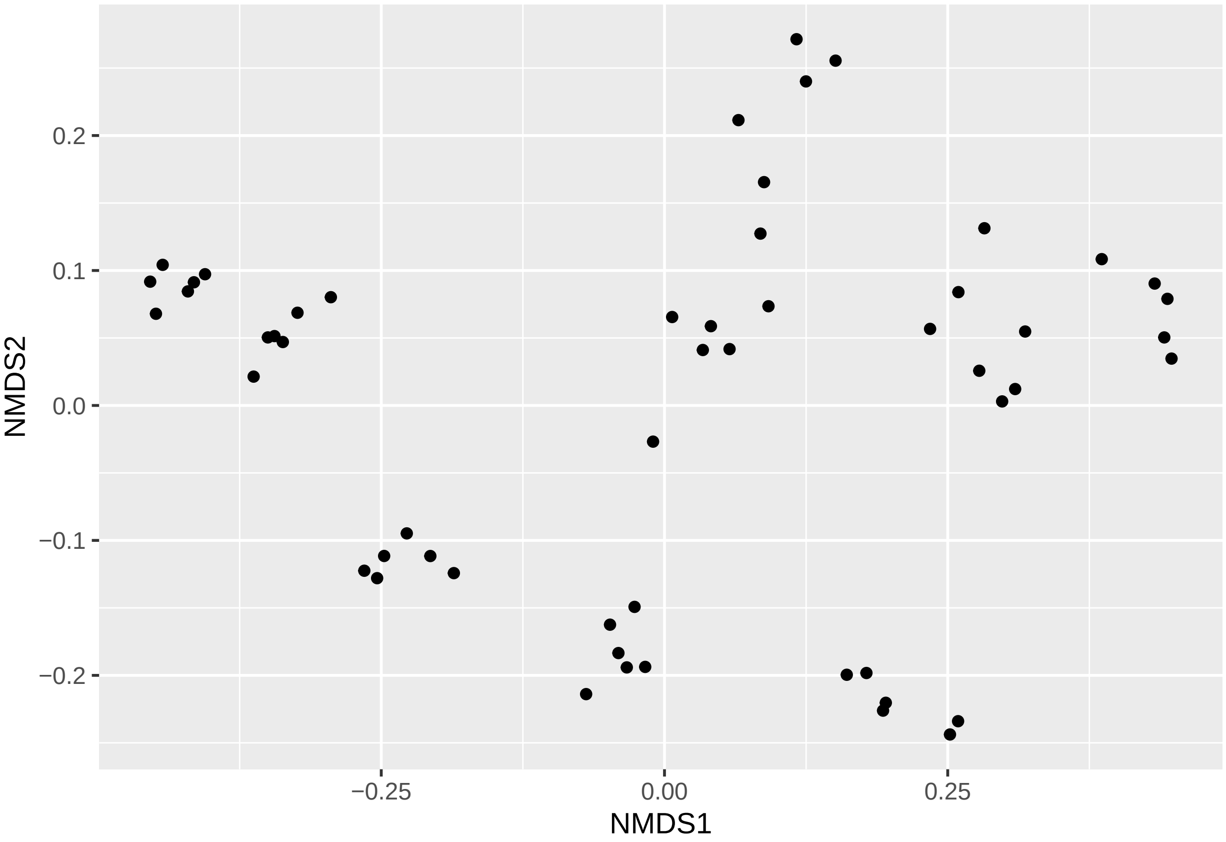
NMDS might be a better choice since we have non gradient determining facotrs site and farm managment effecting the bacteria community

NMDS Ordination

```r
phy16s_ord_NMDS <- ordinate(physeq_16s_ord, "NMDS", "bray")
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.0707239
## Run 1 stress 0.07072393
## ... Procrustes: rmse 1.537255e-05  max resid 5.868764e-05
## ... Similar to previous best
## Run 2 stress 0.07072337
## ... New best solution
## ... Procrustes: rmse 0.0005246964  max resid 0.002791055
## ... Similar to previous best
## Run 3 stress 0.07072343
## ... Procrustes: rmse 1.757084e-05  max resid 5.586827e-05
## ... Similar to previous best
## Run 4 stress 0.07072343
## ... Procrustes: rmse 3.854813e-05  max resid 0.0002420779
## ... Similar to previous best
## Run 5 stress 0.07072337
## ... New best solution
## ... Procrustes: rmse 4.161409e-06  max resid 1.104686e-05
## ... Similar to previous best
## Run 6 stress 0.07072337
## ... New best solution
## ... Procrustes: rmse 4.046738e-06  max resid 1.24879e-05
## ... Similar to previous best
## Run 7 stress 0.07072337
## ... Procrustes: rmse 2.433801e-06  max resid 7.992474e-06
## ... Similar to previous best
## Run 8 stress 0.07072337
## ... Procrustes: rmse 2.837782e-06  max resid 1.732339e-05
## ... Similar to previous best
## Run 9 stress 0.0707239
## ... Procrustes: rmse 0.0005247064  max resid 0.002792583
## ... Similar to previous best
## Run 10 stress 0.07072347
## ... Procrustes: rmse 3.100061e-05  max resid 0.0001109773
## ... Similar to previous best
## Run 11 stress 0.07072337
## ... Procrustes: rmse 5.69002e-06  max resid 2.90601e-05
## ... Similar to previous best
## Run 12 stress 0.07072337
## ... Procrustes: rmse 6.388705e-06  max resid 2.221169e-05
## ... Similar to previous best
## Run 13 stress 0.0707239
## ... Procrustes: rmse 0.0005246925  max resid 0.002792089
## ... Similar to previous best
## Run 14 stress 0.0707239
## ... Procrustes: rmse 0.000524513  max resid 0.002792986
## ... Similar to previous best
## Run 15 stress 0.07072337
## ... Procrustes: rmse 4.430124e-06  max resid 1.41367e-05
## ... Similar to previous best
## Run 16 stress 0.0707239
## ... Procrustes: rmse 0.0005241659  max resid 0.002792631
## ... Similar to previous best
## Run 17 stress 0.07072337
## ... Procrustes: rmse 2.107846e-06  max resid 6.354439e-06
## ... Similar to previous best
## Run 18 stress 0.07072404
## ... Procrustes: rmse 0.0005270439  max resid 0.002791887
## ... Similar to previous best
## Run 19 stress 0.07072337
## ... Procrustes: rmse 9.303553e-06  max resid 2.548285e-05
## ... Similar to previous best
## Run 20 stress 0.0707239
## ... Procrustes: rmse 0.0005249115  max resid 0.002792383
## ... Similar to previous best
## *** Solution reached
```

```
plot_ordination(physeq_16s_ord, phy16s_ord_NMDS)
```



Data has good ordination with NMDS must see stress to make sure the algorithum didnt force fit any ordination.

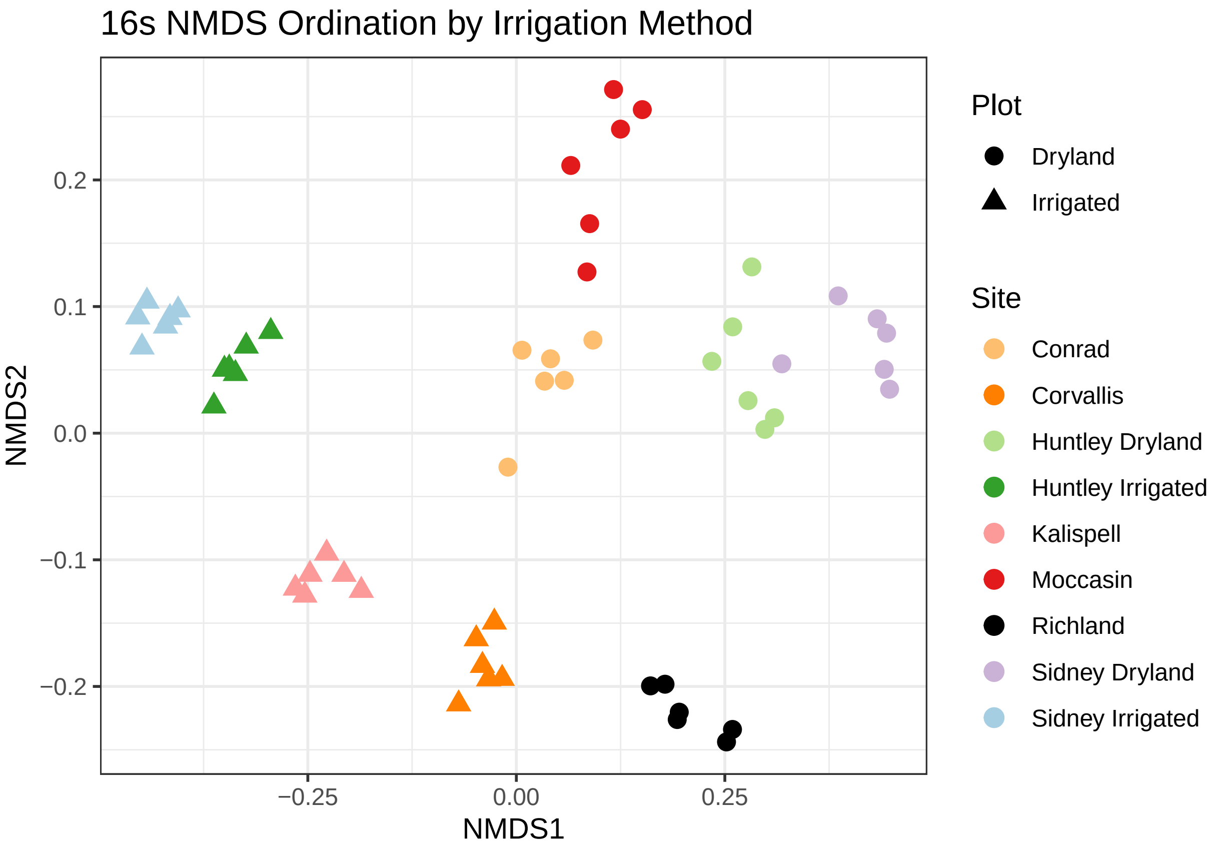
```
phy16s_ord_NMDS
```

```
##
## Call:
## metaMDS(comm = veganifyOTU(physeq), distance = distance)
##
## global Multidimensional Scaling using monoMDS
##
## Data:     wisconsin(sqrt(veganifyOTU(physeq)))
## Distance: bray
##
## Dimensions: 2
## Stress:     0.07072337
## Stress type 1, weak ties
## Two convergent solutions found after 20 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on 'wisconsin(sqrt(veganifyOTU(physeq)))'
```

After stress test run, we get a value of 0.07 which is considered good, anything below 0.2 is acceptable.

Plot NMDS with Site and Irrigation

```
plot_ordination(physeq_16s_ord, phy16s_ord_NMDS, shape = "Plot", color = "Site")+
  geom_point(size = 3)+
  scale_color_manual(values =  farm_col_paired)+
  #stat_ellipse(type = "norm", linetype = 2, aes(color ="Plot"), show.legend = TRUE) +
  ggtitle("16s NMDS Ordination by Irrigation Method")+
  theme_bw()
```



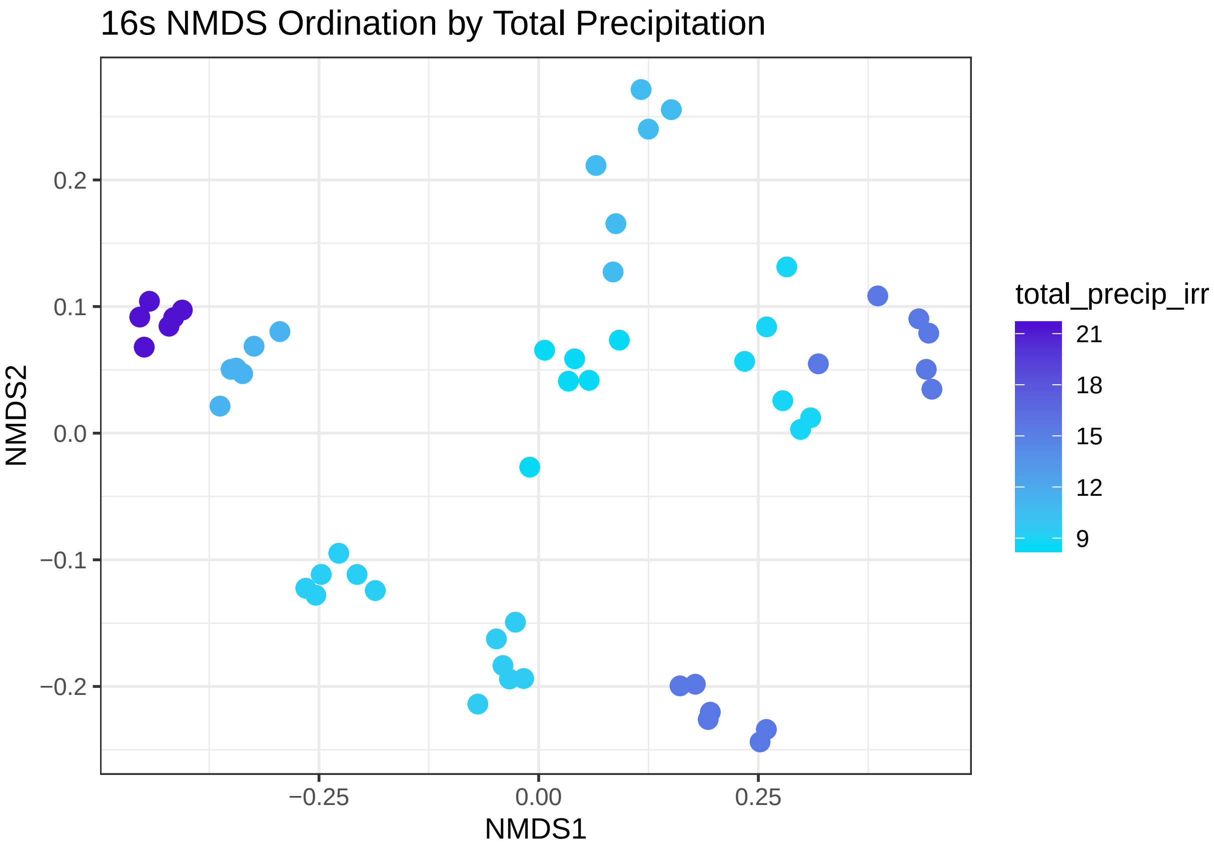
```
NMDS_16s<-plot_ordination(physeq_16s_ord, phy16s_ord_NMDS, shape = "Plot", color = "Site")+
  geom_point(size = 3)+
  scale_color_manual(values =  farm_col_paired)+
  #stat_ellipse(type = "norm", linetype = 2, aes(color ="Plot"), show.legend = TRUE) +
  #ggtitle("16s NMDS Ordination by Irrigation Method")+
  theme_bw()
```

Print to tiff

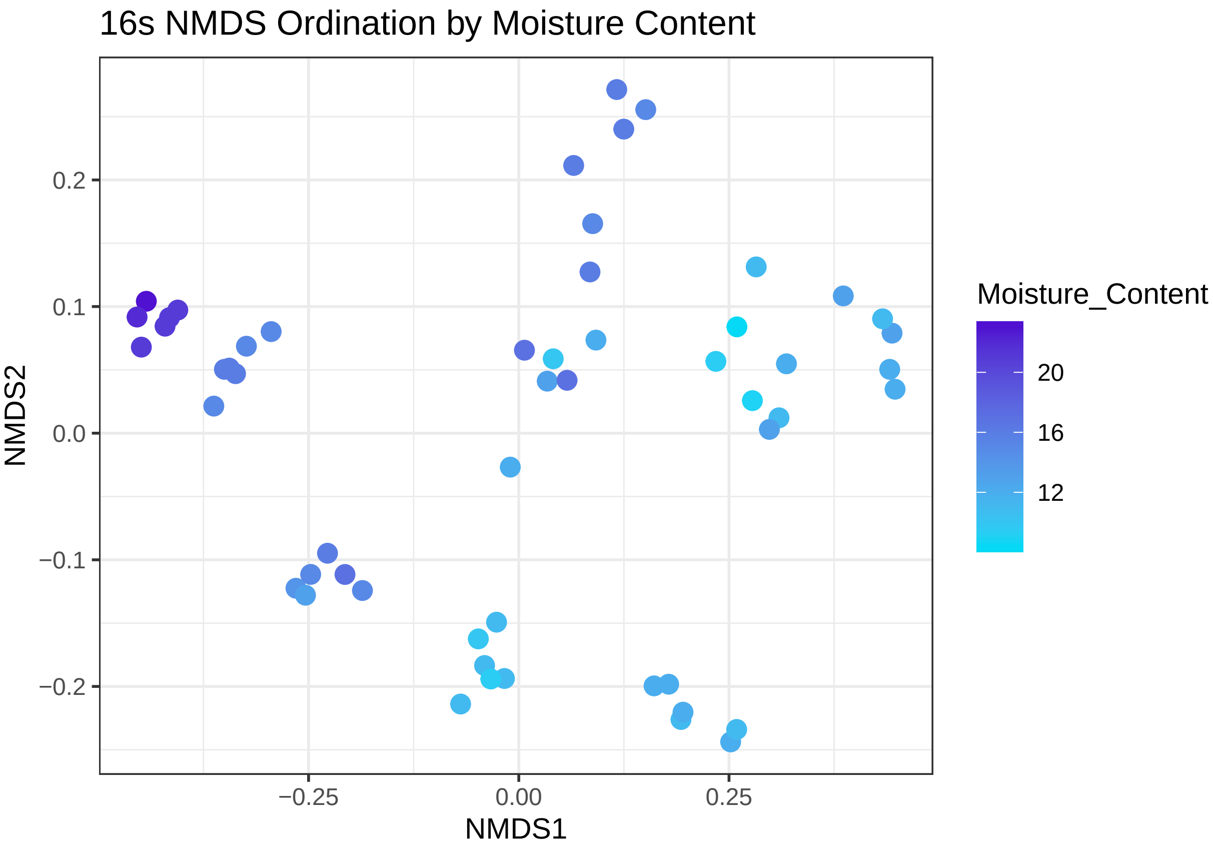
```
## pdf
##   2
```

Lets explore the other farm variables and factors with the NMDS ordination to visually see if there are more patterns.

```
plot_ordination(physeq_16s_ord, phy16s_ord_NMDS, color = "total_precip_irr")+
  geom_point(size = 3)+
  scale_color_gradient(low='#05D9F6', high='#5011D1')+
  ggtitle("16s NMDS Ordination by Total Precipitation")+
  theme_bw()
```

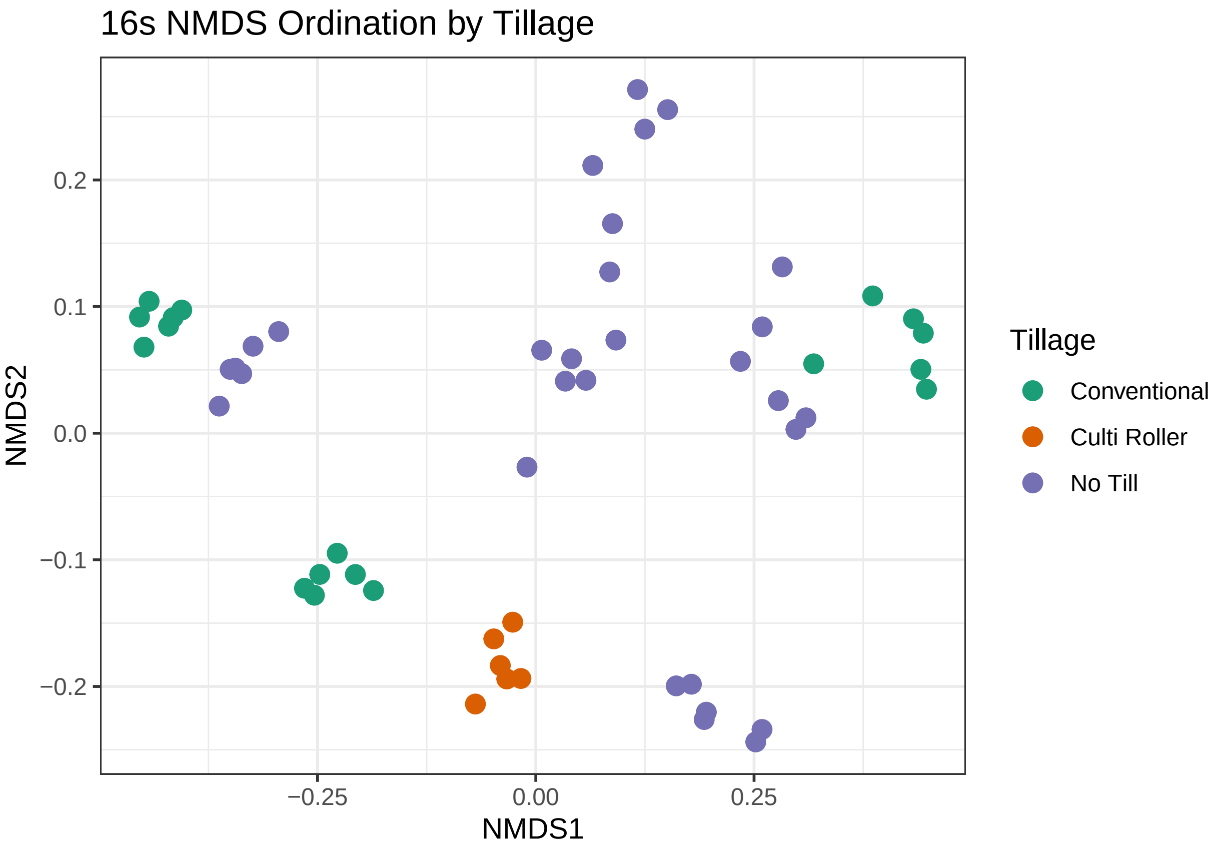


```
plot_ordination(physeq_16s_ord, phy16s_ord_NMDS, color = "Moisture_Content")+
  geom_point(size = 3)+
  scale_color_gradient(low='#05D9F6', high='#5011D1')+
  ggtitle("16s NMDS Ordination by Moisture Content")+
  theme_bw()
```



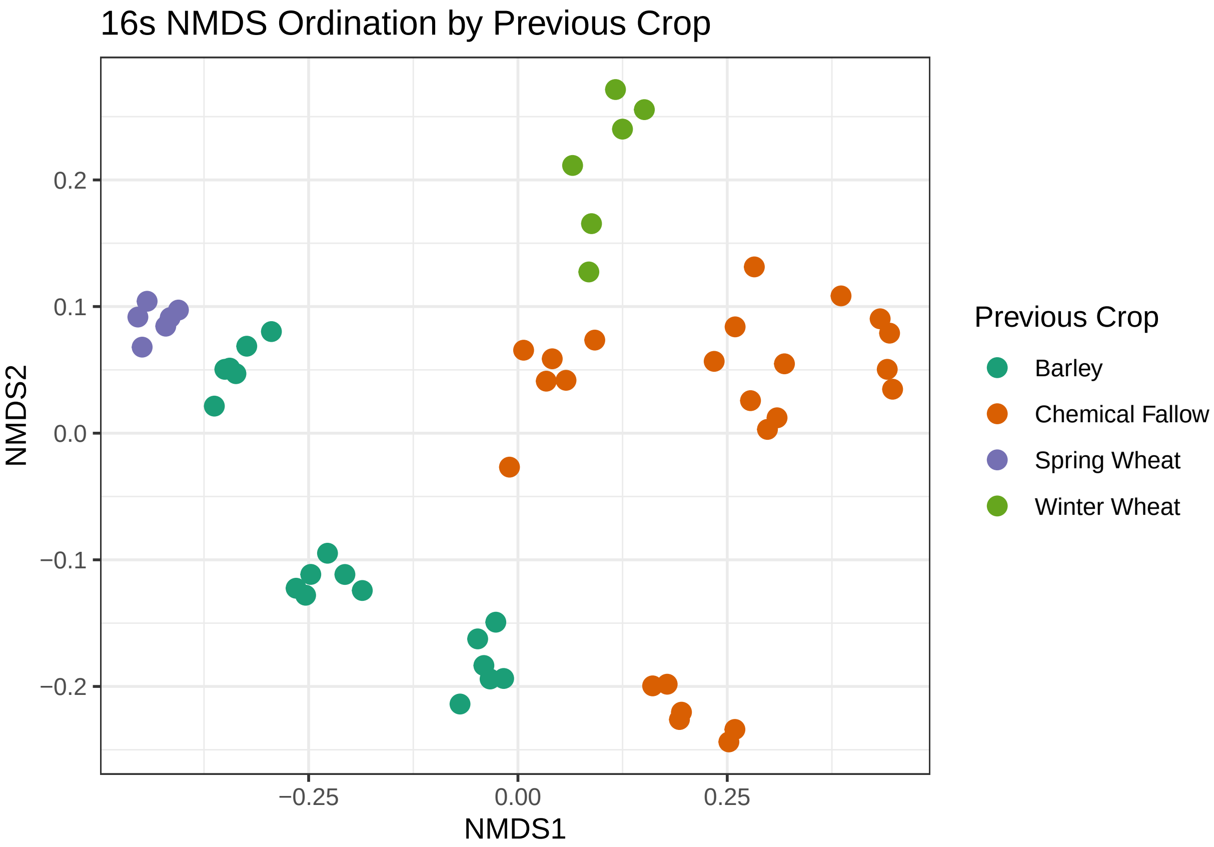
Total mositure content is present but hard to see with the naked eye.

```
plot_ordination(physeq_16s_ord, phy16s_ord_NMDS, color = "Tillage")+
  geom_point(size = 3)+
  scale_color_manual(values =  farm_col_dark, breaks=c("Conventional", "Culti-roller", "No_till"),
                   labels=c("Conventional", "Culti Roller", "No Till"))+
  ggtitle("16s NMDS Ordination by Tillage")+
  theme_bw()
```



```
## pdf
##   2
```

```
plot_ordination(physeq_16s_ord, phy16s_ord_NMDS, color = "prev_crop")+
  geom_point(size = 3)+
  scale_color_manual(values =  c("#1B9E77", "#D95F02", "#7570B3", "#66a61E"),
                    name = "Previous Crop",
                    breaks=c("barley",  "Chem_fallow", "Spring_wheat", "winter_wheat"),
                    labels=c("Barley", "Chemical Fallow", "Spring Wheat", "Winter Wheat"))+
  ggtitle("16s NMDS Ordination by Previous Crop")+
  theme_bw()
```



```
## pdf
##   2
```

Beta dispersions

Test the diffrences in group homogeneities. Do our farm managment factors effect the homogeneitiey of the bray curtis distance?

If a group (Site) in the MDS space are close but have diffrenent dispersion you could have a signifcant results when it is only a diffrence in dispersion.

Anderson (2006)-https://www.ncbi.nlm.nih.gov/pubmed/16542252

https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1461-0248.2006.00926.x

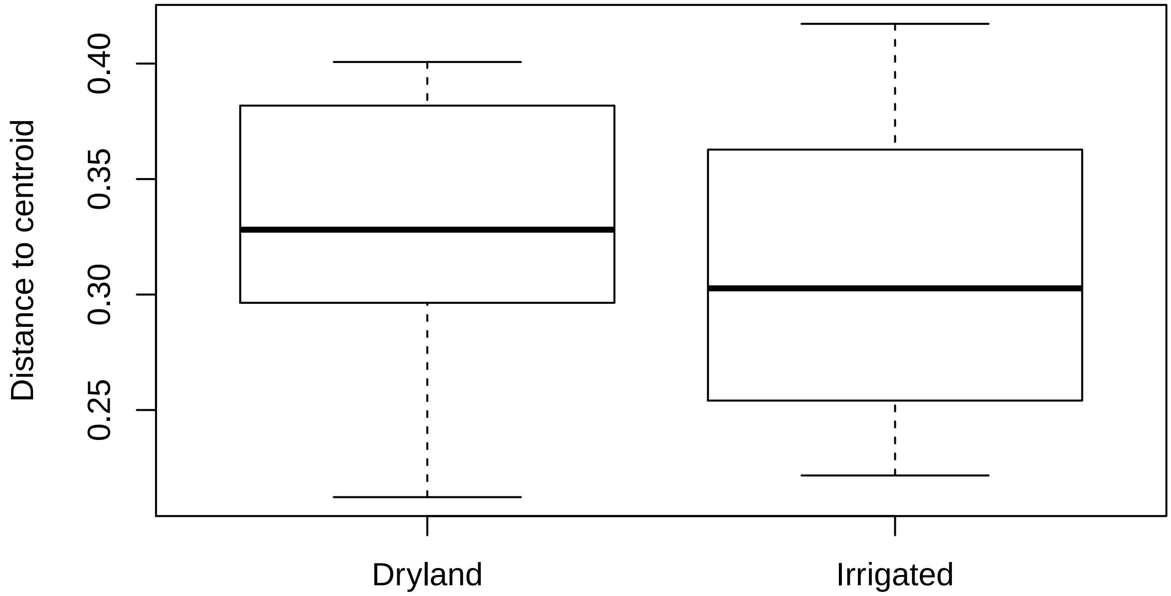
Irrigation beta dispersion

```
disp_16s_plot <- betadisper(distance(physeq_16s_ord, method = "bray"), meta2$Plot)
permutest(disp_16s_plot, pairwise=TRUE, permutations=1000)
```

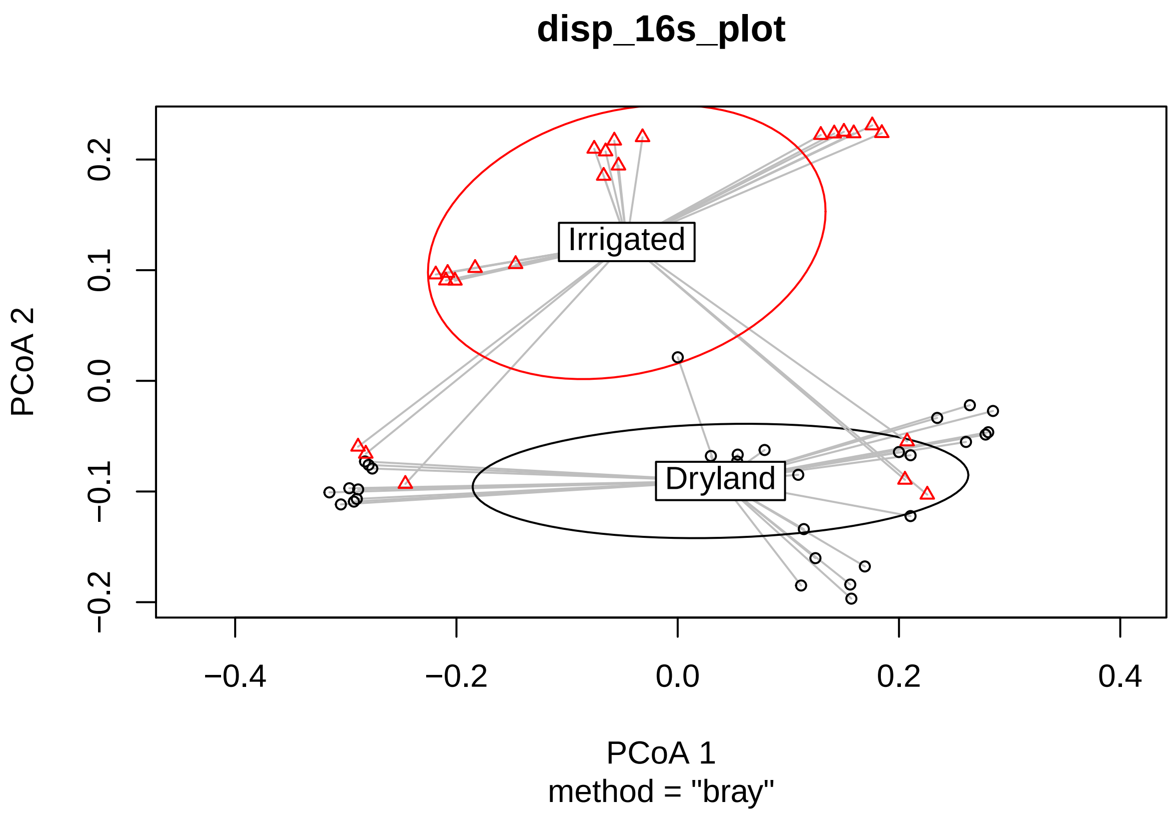
```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 1000
##
```

```
## Response: Distances
##           Df   Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups     1 0.003837 0.0038372 1.0339   1000 0.2987
## Residuals 52 0.192984 0.0037112
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##           Dryland Irrigated
## Dryland              0.3017
## Irrigated 0.31394
```

```
boxplot(disp_16s_plot)
```



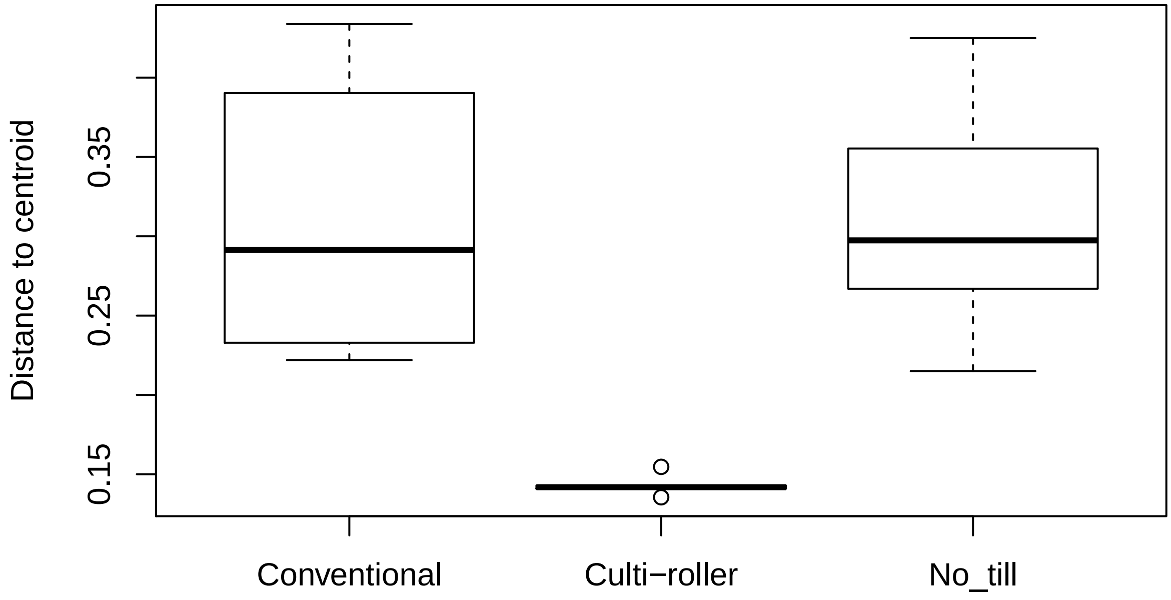
```
plot(disp_16s_plot, hull = FALSE, ellipse = TRUE)
```



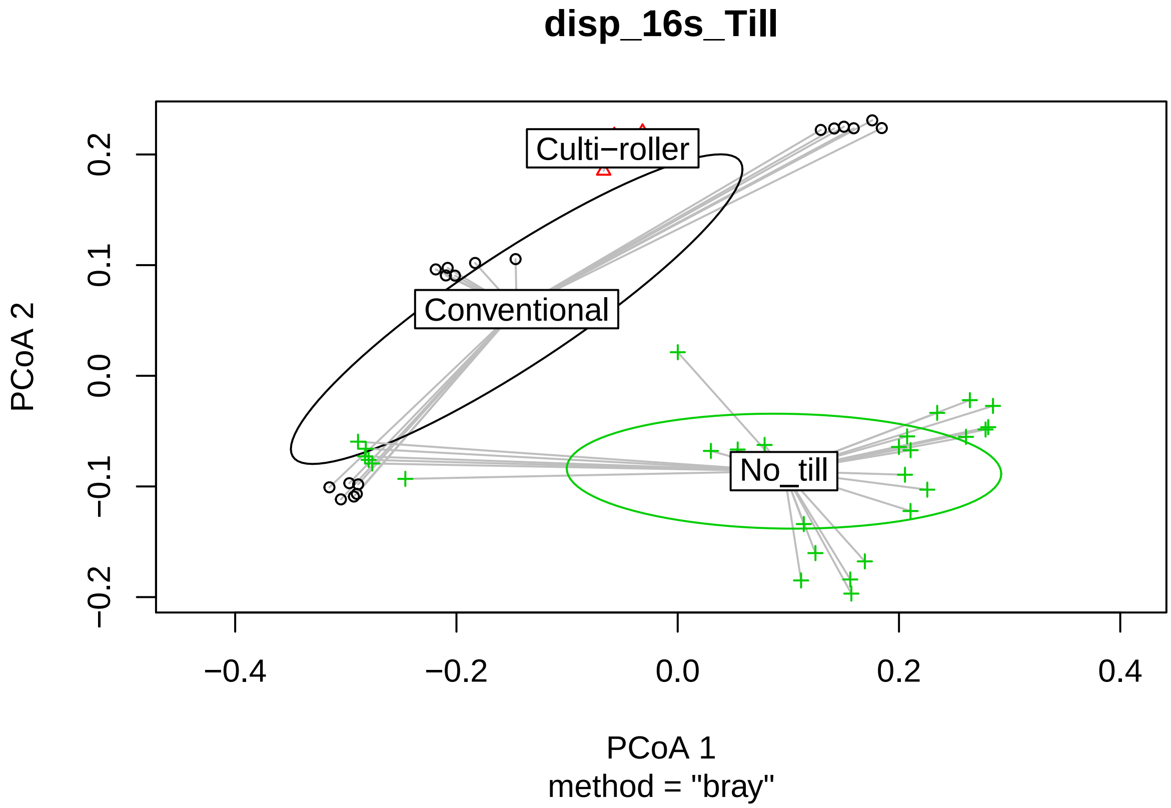
```
disp_16s_Till <- betadisper(distance(physeq_16s_ord, method = "bray"), meta2$Tillage)
permutest(disp_16s_Till, pairwise=TRUE, permutations=1000)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 1000
##
## Response: Distances
##           Df  Sum Sq  Mean Sq      F N.Perm   Pr(>F)
## Groups     2 0.15179 0.075893 17.177   1000 0.000999 ***
## Residuals 51 0.22533 0.004418
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##              Conventional Culti-roller No_till
## Conventional                9.9900e-04   0.977
## Culti-roller   3.4470e-05                0.001
## No_till        9.7962e-01   3.1559e-07
```

```
boxplot(disp_16s_Till)
```



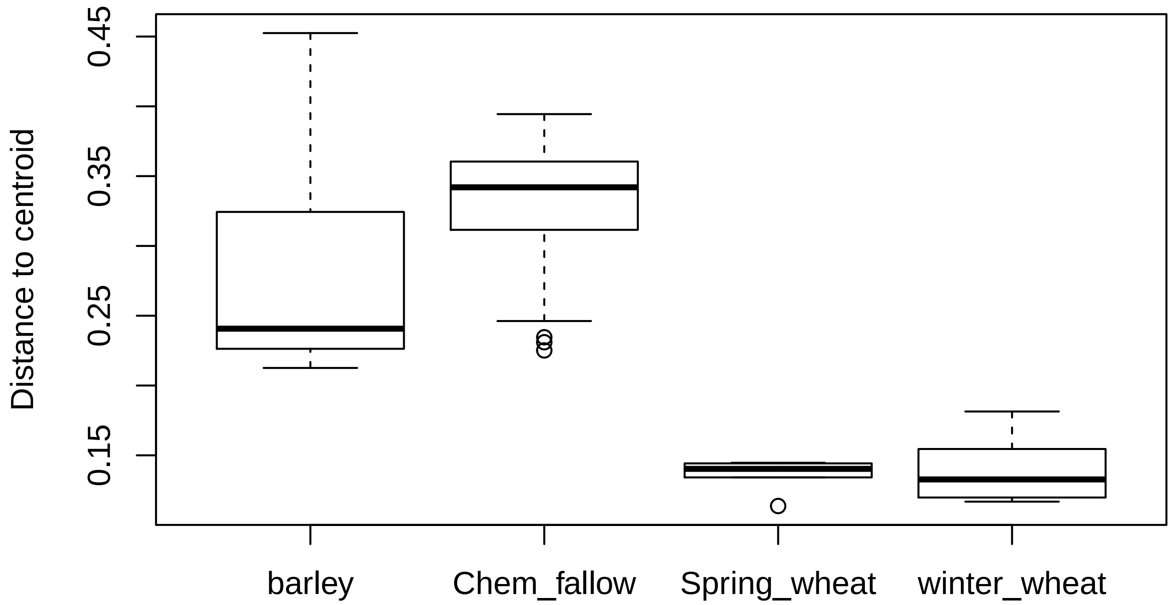
```
plot(disp_16s_Till, hull = FALSE, ellipse = TRUE)
```



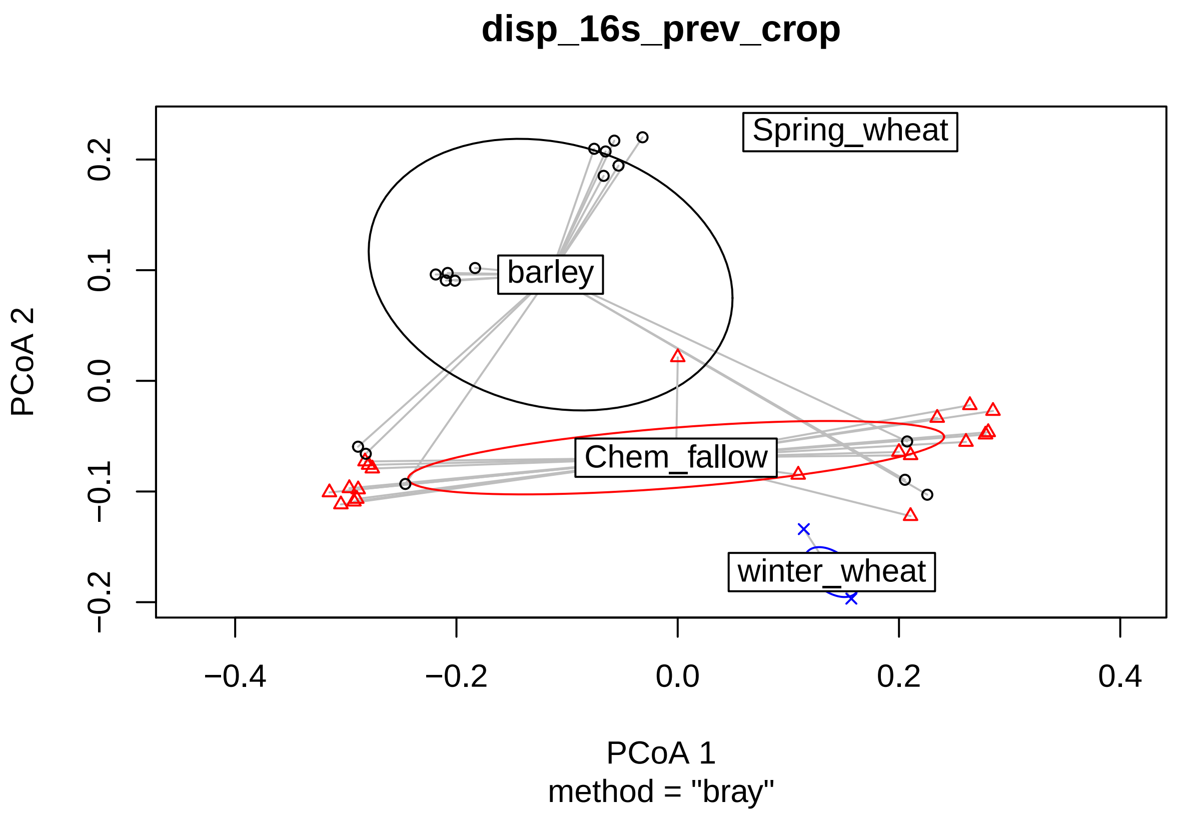
```
disp_16s_prev_crop <- betadisper(distance(physeq_16s_ord, method = "bray"), meta2$prev_crop)
permutest(disp_16s_prev_crop, pairwise=TRUE, permutations=1000)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 1000
##
## Response: Distances
##           Df  Sum Sq  Mean Sq      F N.Perm   Pr(>F)
## Groups     3 0.29014 0.096712 26.821   1000 0.000999 ***
## Residuals 50 0.18029 0.003606
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##                  barley Chem_fallow Spring_wheat winter_wheat
## barley                   5.1948e-02   9.9900e-04       0.0010
## Chem_fallow  4.3211e-02               9.9900e-04       0.0010
## Spring_wheat 3.3222e-04  7.0154e-10                    0.7722
## winter_wheat 4.6584e-04  1.5388e-09   7.6209e-01
```

```
boxplot(disp_16s_prev_crop)
```



```
plot(disp_16s_prev_crop, hull = FALSE, ellipse = TRUE)
```



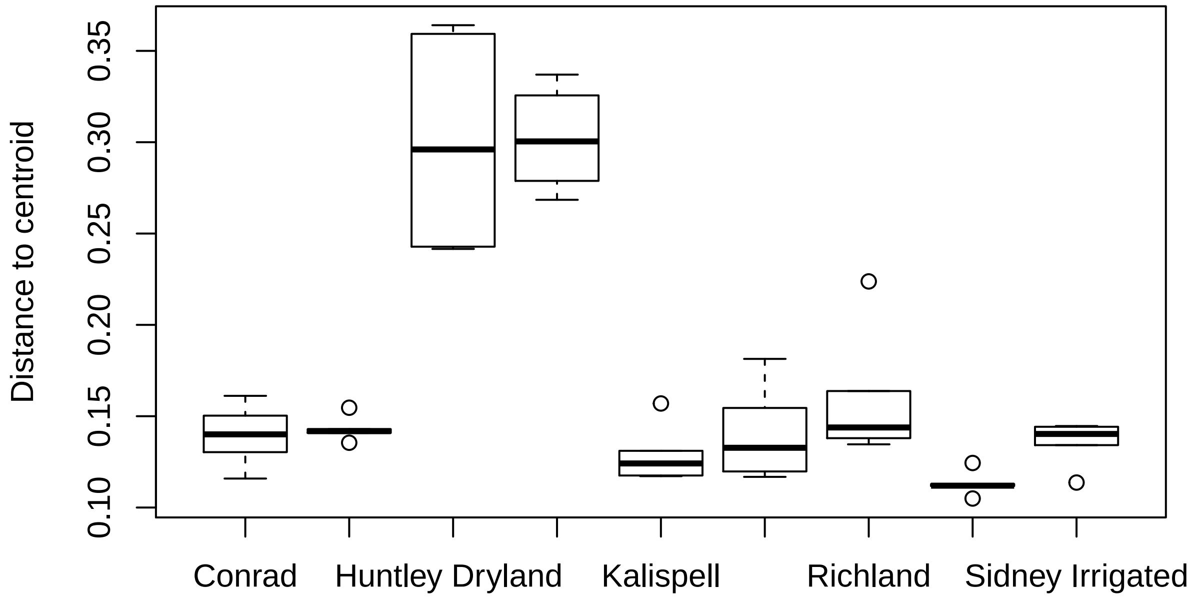
```
TukeyHSD(disp_16s_prev_crop)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##                                   diff          lwr         upr     p adj
## Chem_fallow-barley         0.043262876 -0.006496561  0.09302231 0.1092488
## Spring_wheat-barley       -0.147348629 -0.222577826 -0.07211943 0.0000212
## winter_wheat-barley       -0.143910802 -0.219139999 -0.06868160 0.0000323
## Spring_wheat-Chem_fallow  -0.190611505 -0.263451862 -0.11777115 0.0000000
## winter_wheat-Chem_fallow  -0.187173678 -0.260014035 -0.11433332 0.0000001
## winter_wheat-Spring_wheat  0.003437827 -0.088698747  0.09557440 0.9996460
```

```
disp_16s_site <- betadisper(distance(physeq_16s_ord, method = "bray"), meta2$Site)
permutest(disp_16s_site, pairwise=TRUE, permutations=1000)
```

```
##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 1000
##
## Response: Distances
##           Df  Sum Sq Mean Sq      F N.Perm   Pr(>F)
## Groups     8 0.25800 0.03225 40.302   1000 0.000999 ***
## Residuals 45 0.03601 0.00080
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##                       Conrad  Corvallis Huntley Dryland  Huntley Irrigated
## Conrad                       6.3437e-01      9.9900e-04        9.9900e-04
## Corvallis         6.4638e-01                 9.9900e-04        9.9900e-04
## Huntley Dryland   1.2635e-04 1.2167e-04                        9.5504e-01
## Huntley Irrigated 2.0623e-07 8.3650e-08      9.4967e-01
## Kalispell         2.4187e-01 5.7211e-02      7.1876e-05        1.0017e-07
## Moccasin          9.9884e-01 7.5736e-01      1.7285e-04        8.1317e-07
## Richland          2.5817e-01 3.1099e-01      6.7079e-04        1.1330e-05
## Sidney Dryland    3.2976e-03 9.3105e-06      2.8677e-05        1.5922e-08
## Sidney Irrigated  6.7891e-01 2.4406e-01      9.5480e-05        9.6493e-08
##                    Kalispell   Moccasin   Richland Sidney Dryland
## Conrad            2.3177e-01 9.9900e-01 2.7772e-01     1.9980e-03
## Corvallis         4.4955e-02 7.9221e-01 3.3467e-01     9.9900e-04
## Huntley Dryland   9.9900e-04 9.9900e-04 1.9980e-03     9.9900e-04
## Huntley Irrigated 9.9900e-04 9.9900e-04 9.9900e-04     9.9900e-04
## Kalispell                    3.8861e-01 6.7932e-02     2.5974e-02
## Moccasin          3.6467e-01            3.2967e-01     9.9900e-03
## Richland          8.1075e-02 3.0855e-01                4.9950e-03
## Sidney Dryland    4.3340e-02 2.7003e-02 9.6827e-03
## Sidney Irrigated  3.4726e-01 7.6209e-01 1.6866e-01     1.5929e-03
##                   Sidney Irrigated
## Conrad                      0.6843
## Corvallis                   0.2637
## Huntley Dryland             0.0010
## Huntley Irrigated           0.0010
## Kalispell                   0.3846
## Moccasin                    0.8012
## Richland                    0.1608
## Sidney Dryland              0.0010
## Sidney Irrigated
```

```r
boxplot(disp_16s_site)
```

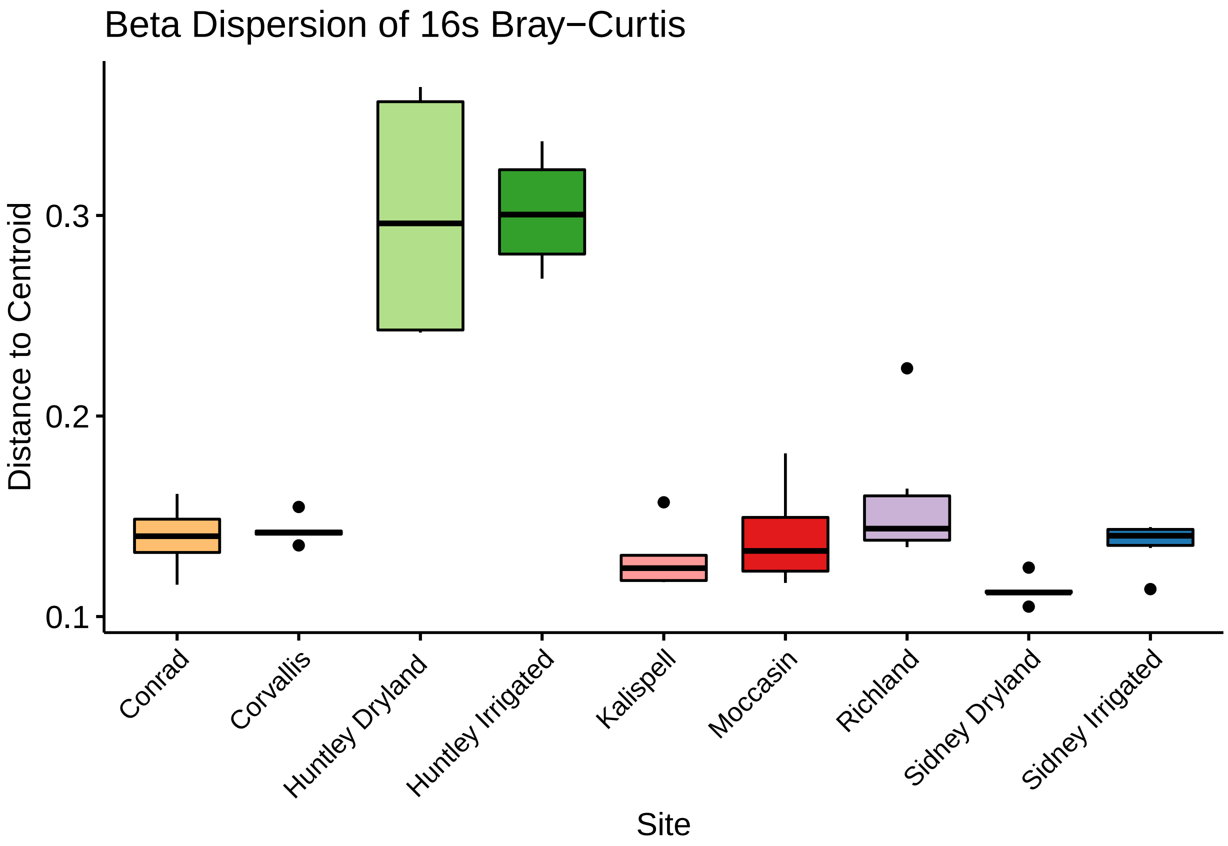


```
TukeyHSD(disp_16s_site)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##                                            diff         lwr          upr
## Corvallis-Conrad                   3.287453e-03 -0.04990884  0.056483741
## Huntley Dryland -Conrad            1.603433e-01  0.10714702  0.213539598
## Huntley Irrigated-Conrad           1.621654e-01  0.10896914  0.215361717
## Kalispell-Conrad                  -1.112795e-02 -0.06432424  0.042068339
## Moccasin-Conrad                    1.763081e-05 -0.05317866  0.053213919
## Richland-Conrad                    1.834374e-02 -0.03485255  0.071540026
## Sidney Dryland-Conrad             -2.663411e-02 -0.07983040  0.026562176
## Sidney Irrigated-Conrad           -3.420196e-03 -0.05661648  0.049776093
## Huntley Dryland -Corvallis         1.570559e-01  0.10385957  0.210252146
## Huntley Irrigated-Corvallis        1.588780e-01  0.10568169  0.212074264
## Kalispell-Corvallis               -1.441540e-02 -0.06761169  0.038780887
## Moccasin-Corvallis                -3.269822e-03 -0.05646611  0.049926467
## Richland-Corvallis                 1.505628e-02 -0.03814000  0.068252573
## Sidney Dryland-Corvallis          -2.992157e-02 -0.08311785  0.023274723
## Sidney Irrigated-Corvallis        -6.707649e-03 -0.05990394  0.046488640
## Huntley Irrigated-Huntley Dryland  1.822119e-03 -0.05137417  0.055018407
## Kalispell-Huntley Dryland         -1.714713e-01 -0.22466755 -0.118274970
## Moccasin-Huntley Dryland          -1.603257e-01 -0.21352197 -0.107129390
## Richland-Huntley Dryland          -1.419996e-01 -0.19519586 -0.088803284
## Sidney Dryland-Huntley Dryland    -1.869774e-01 -0.24017371 -0.133781134
## Sidney Irrigated-Huntley Dryland  -1.637635e-01 -0.21695979 -0.110567217
## Kalispell-Huntley Irrigated       -1.732934e-01 -0.22648967 -0.120097089
## Moccasin-Huntley Irrigated        -1.621478e-01 -0.21534409 -0.108951509
## Richland-Huntley Irrigated        -1.438217e-01 -0.19701798 -0.090625403
## Sidney Dryland-Huntley Irrigated  -1.887995e-01 -0.24199583 -0.135603253
## Sidney Irrigated-Huntley Irrigated -1.655856e-01 -0.21878191 -0.112389336
## Moccasin-Kalispell                   1.114558e-02 -0.04205071  0.064341869
## Richland-Kalispell                   2.947169e-02 -0.02372460  0.082667975
## Sidney Dryland-Kalispell            -1.550616e-02 -0.06870245  0.037690125
## Sidney Irrigated-Kalispell           7.707753e-03 -0.04548854  0.060904042
## Richland-Moccasin                    1.832611e-02 -0.03487018  0.071522395
## Sidney Dryland-Moccasin             -2.665174e-02 -0.07984803  0.026544545
## Sidney Irrigated-Moccasin           -3.437827e-03 -0.05663412  0.049758462
## Sidney Dryland-Richland             -4.497785e-02 -0.09817414  0.008218439
## Sidney Irrigated-Richland           -2.176393e-02 -0.07496022  0.031432356
## Sidney Irrigated-Sidney Dryland      2.321392e-02 -0.02998237  0.076410205
##                                         p adj
## Corvallis-Conrad                    0.9999999
## Huntley Dryland -Conrad             0.0000000
## Huntley Irrigated-Conrad            0.0000000
## Kalispell-Conrad                    0.9988053
## Moccasin-Conrad                     1.0000000
## Richland-Conrad                     0.9674829
## Sidney Dryland-Conrad               0.7829947
## Sidney Irrigated-Conrad             0.9999999
## Huntley Dryland -Corvallis          0.0000000
## Huntley Irrigated-Corvallis         0.0000000
## Kalispell-Corvallis                 0.9928508
## Moccasin-Corvallis                  0.9999999
## Richland-Corvallis                  0.9904839
## Sidney Dryland-Corvallis            0.6613908
## Sidney Irrigated-Corvallis          0.9999726
## Huntley Irrigated-Huntley Dryland   1.0000000
## Kalispell-Huntley Dryland           0.0000000
## Moccasin-Huntley Dryland            0.0000000
## Richland-Huntley Dryland            0.0000000
## Sidney Dryland-Huntley Dryland      0.0000000
## Sidney Irrigated-Huntley Dryland    0.0000000
## Kalispell-Huntley Irrigated         0.0000000
## Moccasin-Huntley Irrigated          0.0000000
## Richland-Huntley Irrigated          0.0000000
## Sidney Dryland-Huntley Irrigated    0.0000000
## Sidney Irrigated-Huntley Irrigated 0.0000000
## Moccasin-Kalispell                  0.9987917
## Richland-Kalispell                  0.6789891
## Sidney Dryland-Kalispell            0.9884846
## Sidney Irrigated-Kalispell          0.9999209
## Richland-Moccasin                   0.9676667
## Sidney Dryland-Moccasin             0.7823968
## Sidney Irrigated-Moccasin           0.9999999
## Sidney Dryland-Richland             0.1584632
## Sidney Irrigated-Richland           0.9159075
## Sidney Irrigated-Sidney Dryland     0.8835899
```

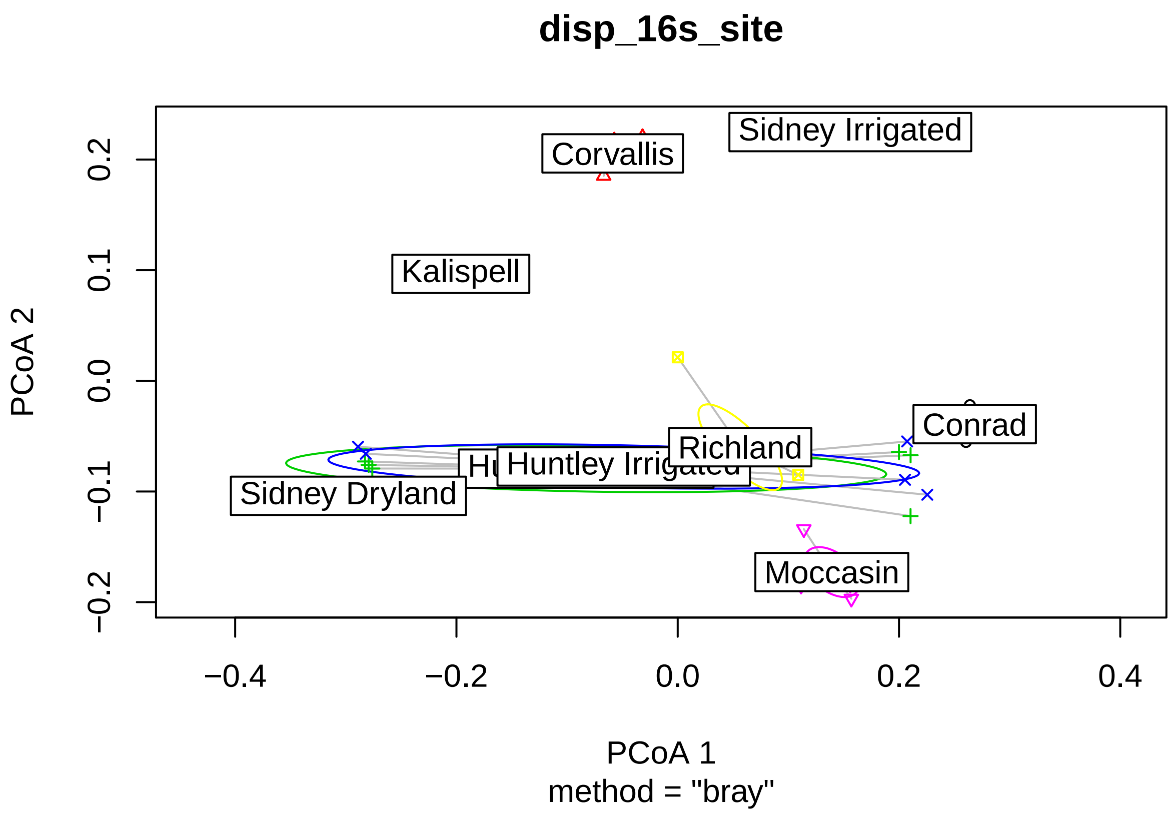
```
farm_col_paired<-(c('#fdbf6f','#ff7f00','#b2df8a','#33a02c','#fb9a99','#e31a1c','#cab2d6','#a6cee3','#1
```

```
dispersion_16s_site<-data.frame(Distance_to_centroid=disp_16s_site$distances,
                                Site=disp_16s_site$group)
ggboxplot(dispersion_16s_site, x = "Site", y = "Distance_to_centroid",
   rug = TRUE,
   fill = "Site", ylab = "Distance to Centroid",
   legend = "none",
   title = "Beta Dispersion of 16s Bray-Curtis",
   palette = farm_col_paired)+
  rotate_x_text(45, size = 10)
```

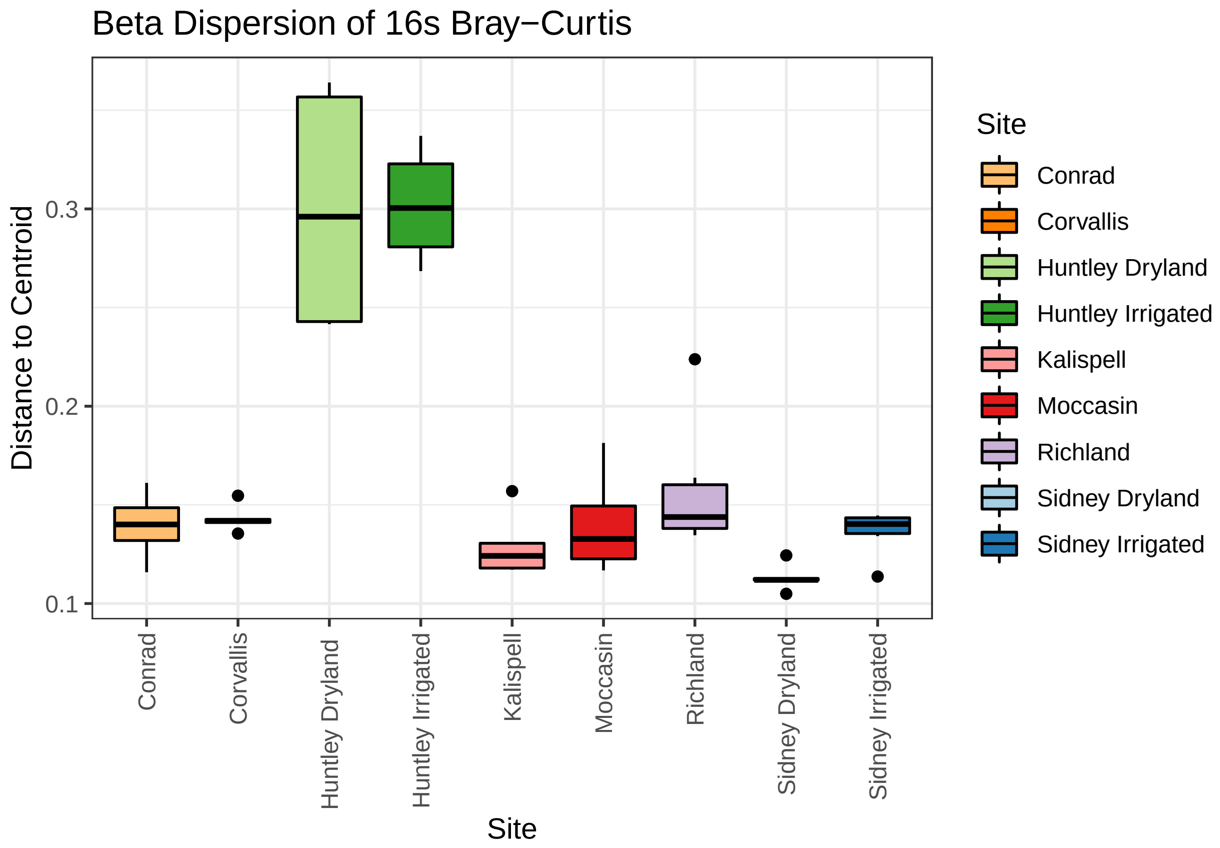


```
## pdf
##   2
```

```
plot(disp_16s_site, hull = FALSE, ellipse = TRUE)
```



```
ggboxplot(dispersion_16s_site, x = "Site", y = "Distance_to_centroid",
   rug = TRUE,
   fill = "Site", ylab = "Distance to Centroid", title = "Beta Dispersion of 16s Bray-Curtis",
   palette = farm_col_paired)+
  theme_bw()+
  rotate_x_text()
```



```r
TukeyHSD(disp_16s_site)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##                                             diff         lwr          upr
## Corvallis-Conrad                    3.287453e-03 -0.04990884  0.056483741
## Huntley Dryland -Conrad             1.603433e-01  0.10714702  0.213539598
## Huntley Irrigated-Conrad            1.621654e-01  0.10896914  0.215361717
## Kalispell-Conrad                   -1.112795e-02 -0.06432424  0.042068339
## Moccasin-Conrad                     1.763081e-05 -0.05317866  0.053213919
## Richland-Conrad                     1.834374e-02 -0.03485255  0.071540026
## Sidney Dryland-Conrad              -2.663411e-02 -0.07983040  0.026562176
## Sidney Irrigated-Conrad            -3.420196e-03 -0.05661648  0.049776093
## Huntley Dryland -Corvallis          1.570559e-01  0.10385957  0.210252146
## Huntley Irrigated-Corvallis         1.588780e-01  0.10568169  0.212074264
## Kalispell-Corvallis                -1.441540e-02 -0.06761169  0.038780887
## Moccasin-Corvallis                 -3.269822e-03 -0.05646611  0.049926467
## Richland-Corvallis                  1.505628e-02 -0.03814000  0.068252573
## Sidney Dryland-Corvallis           -2.992157e-02 -0.08311785  0.023274723
## Sidney Irrigated-Corvallis         -6.707649e-03 -0.05990394  0.046488640
## Huntley Irrigated-Huntley Dryland   1.822119e-03 -0.05137417  0.055018407
## Kalispell-Huntley Dryland          -1.714713e-01 -0.22466755 -0.118274970
## Moccasin-Huntley Dryland           -1.603257e-01 -0.21352197 -0.107129390
## Richland-Huntley Dryland           -1.419996e-01 -0.19519586 -0.088803284
## Sidney Dryland-Huntley Dryland     -1.869774e-01 -0.24017371 -0.133781134
## Sidney Irrigated-Huntley Dryland   -1.637635e-01 -0.21695979 -0.110567217
## Kalispell-Huntley Irrigated        -1.732934e-01 -0.22648967 -0.120097089
## Moccasin-Huntley Irrigated         -1.621478e-01 -0.21534409 -0.108951509
## Richland-Huntley Irrigated         -1.438217e-01 -0.19701798 -0.090625403
## Sidney Dryland-Huntley Irrigated   -1.887995e-01 -0.24199583 -0.135603253
## Sidney Irrigated-Huntley Irrigated -1.655856e-01 -0.21878191 -0.112389336
## Moccasin-Kalispell                  1.114558e-02 -0.04205071  0.064341869
## Richland-Kalispell                  2.947169e-02 -0.02372460  0.082667975
## Sidney Dryland-Kalispell           -1.550616e-02 -0.06870245  0.037690125
## Sidney Irrigated-Kalispell          7.707753e-03 -0.04548854  0.060904042
## Richland-Moccasin                   1.832611e-02 -0.03487018  0.071522395
## Sidney Dryland-Moccasin            -2.665174e-02 -0.07984803  0.026544545
## Sidney Irrigated-Moccasin          -3.437827e-03 -0.05663412  0.049758462
## Sidney Dryland-Richland            -4.497785e-02 -0.09817414  0.008218439
## Sidney Irrigated-Richland          -2.176393e-02 -0.07496022  0.031432356
## Sidney Irrigated-Sidney Dryland     2.321392e-02 -0.02998237  0.076410205
##                                        p adj
## Corvallis-Conrad                   0.9999999
## Huntley Dryland -Conrad            0.0000000
## Huntley Irrigated-Conrad           0.0000000
## Kalispell-Conrad                   0.9988053
## Moccasin-Conrad                    1.0000000
## Richland-Conrad                    0.9674829
## Sidney Dryland-Conrad              0.7829947
## Sidney Irrigated-Conrad            0.9999999
## Huntley Dryland -Corvallis         0.0000000
## Huntley Irrigated-Corvallis        0.0000000
## Kalispell-Corvallis                0.9928508
## Moccasin-Corvallis                 0.9999999
## Richland-Corvallis                 0.9904839
## Sidney Dryland-Corvallis           0.6613908
## Sidney Irrigated-Corvallis         0.9999726
## Huntley Irrigated-Huntley Dryland  1.0000000
## Kalispell-Huntley Dryland          0.0000000
## Moccasin-Huntley Dryland           0.0000000
## Richland-Huntley Dryland           0.0000000
## Sidney Dryland-Huntley Dryland     0.0000000
## Sidney Irrigated-Huntley Dryland   0.0000000
## Kalispell-Huntley Irrigated        0.0000000
## Moccasin-Huntley Irrigated         0.0000000
## Richland-Huntley Irrigated         0.0000000
## Sidney Dryland-Huntley Irrigated   0.0000000
## Sidney Irrigated-Huntley Irrigated 0.0000000
## Moccasin-Kalispell                 0.9987917
## Richland-Kalispell                 0.6789891
## Sidney Dryland-Kalispell           0.9884846
## Sidney Irrigated-Kalispell         0.9999209
## Richland-Moccasin                  0.9676667
## Sidney Dryland-Moccasin            0.7823968
## Sidney Irrigated-Moccasin          0.9999999
## Sidney Dryland-Richland            0.1584632
## Sidney Irrigated-Richland          0.9159075
## Sidney Irrigated-Sidney Dryland    0.8835899
```

The results from the beta dispersion show that we have a significant difference in the heterogeneity of our sites due to each of the farm management factors. We see that there are sites that do not have significant differences and others that do. Beta dispersion cannot except models so we cannot detect if the differences are due to nestedness (prev_crop/Site).

Due to the design of the experiment it will be hard to determine if the farm management practices are responsible for the variation in bacterial population.

PERMANOVA (adoins)

To test if any of the farm management factors are statistically significant will use adonis from vegan to perform a PERMANOVA on the bray curtis distance. This will be able to tell the nestedness of the site and farm management.

First try univariate farm management

```
adonis(distance(physeq_16s_ord, method = "bray") ~Plot, data = meta2, permutations = 1000)
```

```
##
## Call:
## adonis(formula = distance(physeq_16s_ord, method = "bray") ~      Plot, data = meta2, permutations =
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2   Pr(>F)
## Plot       1    0.6443 0.64433  5.8993 0.10189 0.000999 ***
## Residuals 52    5.6795 0.10922         0.89811
## Total     53    6.3238                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(distance(physeq_16s_ord, method = "bray") ~Plot*Site, data = meta2, permutations = 1000, strata
```

```
##
## Call:
## adonis(formula = distance(physeq_16s_ord, method = "bray") ~      Plot * Site, data = meta2, permuta
##
## Blocks:  strata
## Permutation: free
## Number of permutations: 1000
##
```

```
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## Plot       1    0.6443 0.64433  15.307 0.10189      1
## Site       7    3.7853 0.54076  12.847 0.59858      1
## Residuals 45    1.8942 0.04209         0.29953
## Total     53    6.3238                 1.00000
```

Data is nested within Site (Location effect) so the signficance in the bray-curtis dissimilarity with respect to plot is not sinifcant with the data nested due to the lack of reproduction of conditions at each plot.

Since there is issues with doing permutated anova over multivatrite data lets try to fit the chemical and farm managment data the NMDS orientation space using the envfit function in vegan

---

Model Selection

ENVFIT

Envfit does not like single variable values so we remove them

```
meta3<-meta2[,-c(3,4,10,11,27,29,35,38,42,46)]
```

Will remove Site category like elevation, lat, long etc that do not differentiate between site we can call these all geographical factors as they do not change between sites.

```
meta3<-meta3[,-c(2,11:13)]
```

Model fitting will be biased by chemical outliers that are in some plots the best way to avoid this is to determine the outliers (See chemical_analysis.Rmd) and remove the whole variable since functions ENVFIT and BIOENV will remove it if there are any n/a values.

Removing Sulfate_Sulfur, Boron, Molybdenum, Potassium, Vanadium, Chromium and Sodium

```
meta3<-meta3[,-c(16,18,21,29,31)]
```

```
envfit16s <- envfit(phy16s_ord_NMDS , meta3, na.rm = TRUE, permu= 10000)
envfit16s
```

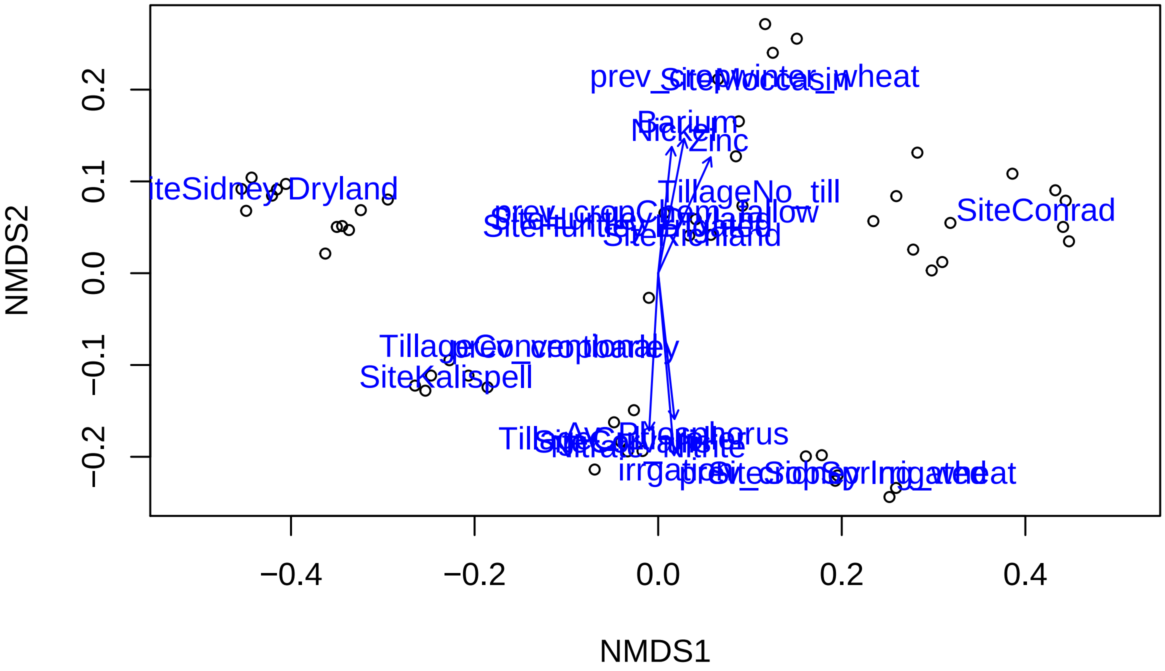
```
##
## ***VECTORS
##
##                     NMDS1    NMDS2     r2    Pr(>r)
## season_precip    -0.29503  0.95549 0.0412   0.34227
## irrgation         0.08747 -0.99617 0.5795 9.999e-05 ***
## total_precip_irr -0.03427 -0.99941 0.0989   0.07309 .
## grain_yield      -0.02370 -0.99972 0.1658   0.01210 *
## Organic_Matter   -0.36906 -0.92940 0.0463   0.30387
## Moisture_Content  0.37515 -0.92696 0.1496   0.01510 *
## Nitrate_Nitrite  -0.05717 -0.99836 0.4311 9.999e-05 ***
## Ammonia          -0.18950  0.98188 0.0869   0.10329
## Av_Phosphorus     0.11141 -0.99378 0.3735 9.999e-05 ***
## Av_Potassium      0.41077 -0.91174 0.2500   0.00140 **
## pH                0.94794 -0.31846 0.0116   0.74163
## Barium            0.18972  0.98184 0.3251   0.00030 ***
## Calcium           0.08816  0.99611 0.0695   0.16048
## Cobalt           -0.04769  0.99886 0.1502   0.01260 *
## Copper            0.45597  0.89000 0.0829   0.10809
## Iron              0.17116  0.98524 0.1417   0.01670 *
## Magnesium        -0.12438 -0.99224 0.0698   0.16078
## Manganese        -0.75752  0.65281 0.0235   0.54255
## Nickel            0.10687  0.99427 0.2801   0.00020 ***
## Phosphorus        0.17352 -0.98483 0.1356   0.02390 *
## Sulfur            0.22889  0.97345 0.0534   0.24398
## Zinc              0.41230  0.91105 0.2815 9.999e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 10000
##
## ***FACTORS:
##
## Centroids:
##                           NMDS1   NMDS2
## SiteConrad               0.4116  0.0696
## SiteCorvallis           -0.0391 -0.1828
## SiteHuntley Dryland     -0.0248  0.0566
## SiteHuntley Irrigated   -0.0337  0.0487
## SiteKalispell           -0.2310 -0.1154
## SiteMoccasin             0.1050  0.2119
## SiteRichland             0.0368  0.0423
## SiteSidney Dryland      -0.4312  0.0895
## SiteSidney Irrigated     0.2064 -0.2203
## Pea_varietyAC Earlystar -0.0754 -0.0086
## Pea_varietyCDC Saffron   0.1209  0.0000
## Pea_varietyCDC Saffron  -0.1299 -0.0575
## Pea_varietyDelta         0.0628 -0.0029
## Pea_varietyDS Admiral    0.0703  0.0032
## Pea_varietyMajoret      -0.0587  0.0082
## Pea_varietyNavarro      -0.0641  0.0129
## PlotDryland              0.0195  0.0940
## PlotIrrigated           -0.0244 -0.1175
## TillageConventional     -0.1519 -0.0821
## TillageCulti-roller     -0.0391 -0.1828
## TillageNo_till           0.0990  0.0858
## prev_cropbarley         -0.1013 -0.0832
## prev_cropChem_fallow    -0.0019  0.0645
## prev_cropSpring_wheat    0.2064 -0.2203
## prev_cropwinter_wheat    0.1050  0.2119
##
## Goodness of fit:
##                 r2    Pr(>r)
## Site        0.7556 9.999e-05 ***
## Pea_variety 0.0692  0.813919
## Plot        0.1248  0.005799 **
## Tillage     0.2532 9.999e-05 ***
## prev_crop   0.2592  0.000200 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 10000
```

The envfit function allows us to see the correlation of our environmental vectors to the bray-curtis species dissimilarity matrix in NMDS space. This is a loose correlation to real linear correlation but it can tell us how the NMDS orientation is being driven.

write to table

Try a quick plot with base r and vegan for the vectors

```
plot(phy16s_ord_NMDS, display = "sites")
plot(envfit16s, p.max = 0.001 )
```



Model Selection

Bioenv

Bioenv is an iterative procedure in that links enviromental variables to community strucutre by seeking the best subset of enviromental variables that explains community strucutre

---

Info

BIOENV

"The function calculates a community dissimilarity matrix using vegdist. Then it selects all possible subsets of environmental variables, scales the variables, and calculates Euclidean distances for this subset using dist. The function finds the correlation between community dissimilarities and environmental distances, and for each size of subsets, saves the best result. There are 2^p-1 subsets of p variables, and an exhaustive search may take a very, very, very long time (parameter upto offers a partial relief)."

Mantel test

"Mantel statistic is simply a correlation between entries of two dissimilarity matrices (some use cross products, but these are linearly related). However, the significance cannot be directly assessed, because there are N(N-1)/2 entries for just N observations. Mantel developed asymptotic test, but here we use permutations of N rows and columns of dissimilarity matrix."

```
OTU_16s_trim<- as(otu_table(physeq_16s_ord ), "matrix")
```

```
#Transpose the data to have sample names on rows
abund_table16s<-t(OTU_16s_trim)

nrow(abund_table16s)
```

```
## [1] 54
```

```
setdiff(rownames(meta3), rownames(abund_table16s))
```

```
## character(0)
```

Our meta data and sample data match with 0 diffrence in rownames

will use parallel processing to speed up calculations

```
#First detect amount of cores avalible
detectCores()
```

```
## [1] 12
```

Hashed for PDF using ten cores takes around a hour

```r
#get bray-curtis distance
abund_dist16s<-vegdist(abund_table16s, method = "bray")
#make bioenv model against whole meta table with ".", and use gower metric to measure distance in order
#n16s.bioenv <- bioenv(abund_dist16s ~ ., meta3, index = "bray", method = "pearson",
  #                    metric = "gower", upto = 7, parallel = 10)

#summary(n16s.bioenv)
```

```r
#n16s.bioenvdist<-bioenvdist(n16s.bioenv, which = "best")
#mantel(n16s.bioenvdist, abund_dist16s)
```

```r
#n16s.bioenv$whichbest
```

Try BioENV without site, this will favor farm managment variables since they are nested within.

Remove site from meta table

```r
meta4<-meta3[,-c(1)]
```

Rerun BioEnv

```r
#n16s.bioenv.site.na <- bioenv(abund_dist16s ~ ., meta4, index = "bray",
                      #        method = "pearson", metric = "gower", upto = 7, parallel = 10)

#summary(n16s.bioenv.site.na)
```

```r
#n16s.bioenvdist.site.na<-bioenvdist(n16s.bioenv.site.na, which = "best")
#mantel(n16s.bioenvdist.site.na, abund_dist16s)
```

```r
#n16s.bioenv.chem <- bioenv(abund_dist16s ~ Organic_Matter + Moisture_Content +
 #                              Nitrate_Nitrite + Ammonia + Av_Phosphorus + Av_Potassium +
  #                              pH + Barium + Calcium + Cobalt + Copper + Iron + Magnesium +
   #                             Manganese + Nickel + Phosphorus + Sulfur + Zinc, meta3,
    #                          index = "bray", method = "pearson", metric = "gower",
     #                         upto = 7, parallel = 10)

#summary(n16s.bioenv.chem)
```

```r
#n16s.bioenvdist.chem<-bioenvdist(n16s.bioenv.chem, which = "best")
#mantel(n16s.bioenvdist.chem, abund_dist16s)
```

```r
#n16s.bioenv.chem$whichbest
```

CCA/ordistep model selection

CCA model selection uses a procedure to take the a constrained distance ordination with the complete model and compare it with a unconstrainted model. The model starts with no variables then adds variables that make the best model. These models can then be plotted against the same ordination space as vectors.

Ordistep use Akaike information criterion (AIC) which is a estimator of relative quality of the models. AIC is relative to models you are comparing when you compare two models the one with the lower AIC is favored.

```
m1_16 <- cca(abund_table16s ~ ., meta3)
m0_16 <- cca(abund_table16s ~ 1, meta3)
m1_16
```

```
## Call: cca(formula = abund_table16s ~ Site + Pea_variety + Plot +
## season_precip + irrgation + total_precip_irr + Tillage + prev_crop
## + grain_yield + Organic_Matter + Moisture_Content +
## Nitrate_Nitrite + Ammonia + Av_Phosphorus + Av_Potassium + pH +
## Barium + Calcium + Cobalt + Copper + Iron + Magnesium + Manganese
## + Nickel + Phosphorus + Sulfur + Zinc, data = meta3)
##
##               Inertia Proportion Rank
## Total          1.1148     1.0000
## Constrained    0.9560     0.8575   33
## Unconstrained  0.1588     0.1425   20
## Inertia is scaled Chi-square
## Some constraints were aliased because they were collinear (redundant)
##
## Eigenvalues for constrained axes:
##    CCA1    CCA2    CCA3    CCA4    CCA5    CCA6    CCA7    CCA8    CCA9
## 0.25431 0.14197 0.11851 0.08932 0.06617 0.05394 0.04354 0.03038 0.02516
##   CCA10   CCA11   CCA12   CCA13   CCA14   CCA15   CCA16   CCA17   CCA18
## 0.01951 0.01088 0.00777 0.00691 0.00632 0.00595 0.00561 0.00540 0.00513
##   CCA19   CCA20   CCA21   CCA22   CCA23   CCA24   CCA25   CCA26   CCA27
## 0.00510 0.00480 0.00448 0.00432 0.00425 0.00412 0.00394 0.00390 0.00381
##   CCA28   CCA29   CCA30   CCA31   CCA32   CCA33
## 0.00370 0.00350 0.00343 0.00339 0.00327 0.00318
##
## Eigenvalues for unconstrained axes:
##     CA1     CA2     CA3     CA4     CA5     CA6     CA7     CA8
## 0.03364 0.02261 0.01465 0.01075 0.00771 0.00704 0.00614 0.00568
## (Showing 8 of 20 unconstrained eigenvalues)
```

```
m0_16
```

```
## Call: cca(formula = abund_table16s ~ 1, data = meta3)
##
##               Inertia Rank
## Total           1.115
## Unconstrained   1.115   53
## Inertia is scaled Chi-square
```

```
##
## Eigenvalues for unconstrained axes:
##     CA1     CA2     CA3     CA4     CA5     CA6     CA7     CA8
## 0.27434 0.14289 0.11948 0.09087 0.06900 0.05843 0.04769 0.03622
## (Showing 8 of 53 unconstrained eigenvalues)
```

Ordistep

```
model_16s <-ordistep(m0_16, scope=formula(m1_16))
```

```
model_16s$anova
```

	Df	AIC	F	Pr(>F)
+ Site	8	715.5896	9.7553	0.005

Site is again nesting the data in the model selection

CCA without site

```
m1_16_site_na <- cca(abund_table16s ~ ., meta4)
m0_16_site_na <- cca(abund_table16s ~ 1, meta4)
m1_16_site_na
```

```
## Call: cca(formula = abund_table16s ~ Pea_variety + Plot +
## season_precip + irrgation + total_precip_irr + Tillage + prev_crop
## + grain_yield + Organic_Matter + Moisture_Content +
## Nitrate_Nitrite + Ammonia + Av_Phosphorus + Av_Potassium + pH +
## Barium + Calcium + Cobalt + Copper + Iron + Magnesium + Manganese
## + Nickel + Phosphorus + Sulfur + Zinc, data = meta4)
##
##               Inertia Proportion Rank
## Total          1.1148     1.0000
## Constrained    0.9419     0.8449   32
## Unconstrained  0.1729     0.1551   21
## Inertia is scaled Chi-square
## Some constraints were aliased because they were collinear (redundant)
##
## Eigenvalues for constrained axes:
##    CCA1    CCA2    CCA3    CCA4    CCA5    CCA6    CCA7    CCA8    CCA9
## 0.24650 0.14197 0.11840 0.08910 0.06616 0.05385 0.04354 0.02939 0.02516
##   CCA10   CCA11   CCA12   CCA13   CCA14   CCA15   CCA16   CCA17   CCA18
## 0.01945 0.01087 0.00764 0.00688 0.00632 0.00567 0.00558 0.00529 0.00512
##   CCA19   CCA20   CCA21   CCA22   CCA23   CCA24   CCA25   CCA26   CCA27
## 0.00496 0.00467 0.00448 0.00427 0.00418 0.00403 0.00391 0.00388 0.00371
##   CCA28   CCA29   CCA30   CCA31   CCA32
## 0.00352 0.00346 0.00341 0.00331 0.00322
##
## Eigenvalues for unconstrained axes:
##     CA1     CA2     CA3     CA4     CA5     CA6     CA7     CA8
## 0.04099 0.02271 0.01465 0.01076 0.00813 0.00717 0.00684 0.00608
## (Showing 8 of 21 unconstrained eigenvalues)
```

```
m0_16_site_na
```

```
## Call: cca(formula = abund_table16s ~ 1, data = meta4)
##
##               Inertia Rank
## Total           1.115
## Unconstrained   1.115   53
## Inertia is scaled Chi-square
##
## Eigenvalues for unconstrained axes:
##     CA1     CA2     CA3     CA4     CA5     CA6     CA7     CA8
## 0.27434 0.14289 0.11948 0.09087 0.06900 0.05843 0.04769 0.03622
## (Showing 8 of 53 unconstrained eigenvalues)
```

Ordistep

```
model_16s_site_na <-ordistep(m0_16_site_na, scope=formula(m1_16_site_na))
```

```
model_16s_site_na$anova
```

	Df	AIC	F	Pr(>F)
+ prev_crop	3	742.2419	6.449479	0.005
+ Tillage	2	732.3506	7.040700	0.005
+ Manganese	1	726.2969	7.559469	0.005
+ Organic_Matter	1	720.2780	7.364108	0.005
+ grain_yield	1	718.5951	3.176197	0.005
+ season_precip	1	717.3366	2.736860	0.005

Will use CCA modeling with only chemistry data

```
colnames(meta3)
```

```
##  [1] "Site"             "Pea_variety"      "Plot"
##  [4] "season_precip"    "irrgation"        "total_precip_irr"
##  [7] "Tillage"          "prev_crop"        "grain_yield"
## [10] "Organic_Matter"   "Moisture_Content" "Nitrate_Nitrite"
## [13] "Ammonia"          "Av_Phosphorus"    "Av_Potassium"
## [16] "pH"               "Barium"           "Calcium"
## [19] "Cobalt"           "Copper"           "Iron"
## [22] "Magnesium"        "Manganese"        "Nickel"
## [25] "Phosphorus"       "Sulfur"           "Zinc"
```

Organic_Matter + Moisture_Content + Nitrate_Nitrite + Ammonia + Av_Phosphorus + Av_Potassium + pH + Barium + Calcium + Cobalt + Copper + Iron + Magnesium + Manganese + Nickel + Phosphorus + Sulfur + Zinc

```
m1_16_cca_chem <- cca(abund_table16s ~ Organic_Matter + Moisture_Content +
                        Nitrate_Nitrite + Ammonia + Av_Phosphorus + Av_Potassium +
                        pH + Barium + Calcium + Cobalt + Copper + Iron + Magnesium +
                        Manganese + Nickel + Phosphorus + Sulfur + Zinc , meta3)
m0_16_cca_chem <- cca(abund_table16s ~ 1, meta3)
m1_16_cca_chem
```

```
## Call: cca(formula = abund_table16s ~ Organic_Matter +
## Moisture_Content + Nitrate_Nitrite + Ammonia + Av_Phosphorus +
## Av_Potassium + pH + Barium + Calcium + Cobalt + Copper + Iron +
## Magnesium + Manganese + Nickel + Phosphorus + Sulfur + Zinc, data
## = meta3)
##
##               Inertia Proportion Rank
## Total           1.1148     1.0000
## Constrained     0.7502     0.6730   18
## Unconstrained  0.3646     0.3270   35
## Inertia is scaled Chi-square
##
## Eigenvalues for constrained axes:
##    CCA1    CCA2    CCA3    CCA4    CCA5    CCA6    CCA7    CCA8    CCA9
## 0.19836 0.13542 0.10981 0.08695 0.06061 0.04668 0.03967 0.01684 0.01073
##   CCA10   CCA11   CCA12   CCA13   CCA14   CCA15   CCA16   CCA17   CCA18
## 0.00876 0.00582 0.00534 0.00468 0.00463 0.00450 0.00405 0.00380 0.00357
##
## Eigenvalues for unconstrained axes:
##     CA1     CA2     CA3     CA4     CA5     CA6     CA7     CA8
## 0.10577 0.04047 0.02449 0.01786 0.01555 0.01398 0.01082 0.00917
## (Showing 8 of 35 unconstrained eigenvalues)
```

```
m0_16_cca_chem
```

```
## Call: cca(formula = abund_table16s ~ 1, data = meta3)
##
##               Inertia Rank
## Total            1.115
## Unconstrained    1.115   53
## Inertia is scaled Chi-square
##
## Eigenvalues for unconstrained axes:
##     CA1     CA2     CA3     CA4     CA5     CA6     CA7     CA8
## 0.27434 0.14289 0.11948 0.09087 0.06900 0.05843 0.04769 0.03622
## (Showing 8 of 53 unconstrained eigenvalues)
```

```
model_16s_cca_chem <-ordistep(m0_16_cca_chem, scope=formula(m1_16_cca_chem))
```

```
model_16s_cca_chem$anova
```

	Df	AIC	F	Pr(>F)
+ Av_Phosphorus	1	750.9760	4.971297	0.005
+ Zinc	1	747.5992	5.339484	0.005

	Df	AIC	F	Pr(>F)
+ Magnesium	1	743.1612	6.330986	0.005
+ Phosphorus	1	738.0787	6.867315	0.005
+ Manganese	1	734.0363	5.682971	0.005
+ Av_Potassium	1	729.3925	6.153349	0.005
+ Sulfur	1	725.8136	5.006619	0.005
+ Barium	1	723.4083	3.824976	0.005

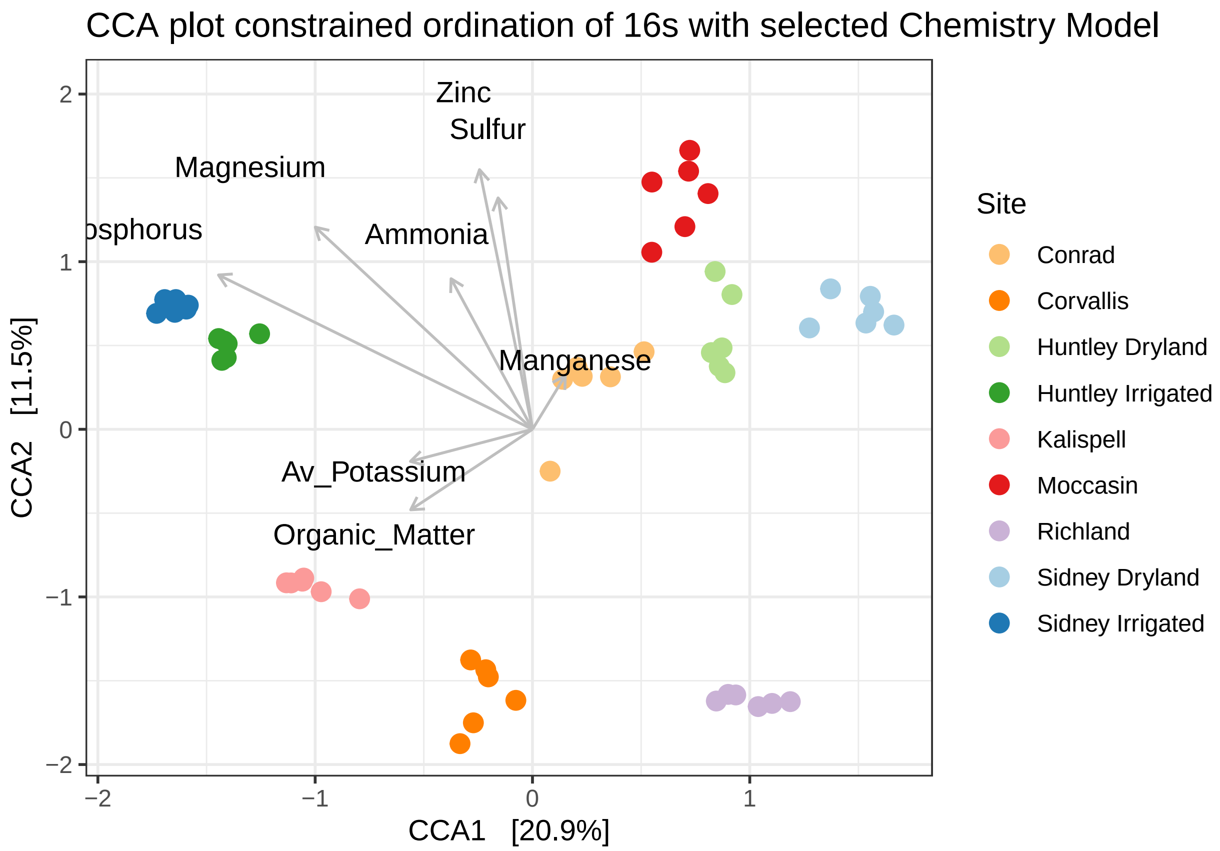
Constrained Ordination plot in ggplot

http://deneflab.github.io/MicrobeMiseq/demos/mothur_2_phyloseq.html#constrained_ordinations

CCA with selected model from Chemical CCA model

```
# CCA ordinate
cca_ord_16s <- ordinate(
    physeq = physeq_16s_ord,
    method = "CCA",
    distance = abund_distnifH,
    formula = ~  Zinc + Magnesium + Phosphorus + Sulfur + Av_Potassium +
      Organic_Matter + Manganese + Ammonia  )

# CCA plot
cca_plot_16s <- plot_ordination(
  physeq = physeq_16s_ord,
  ordination = cca_ord_16s,
    color = "Site",
    axes = c(1,2)) +
    geom_point(aes(colour = Site), size = 3) +
    scale_color_manual(values =  farm_col_paired)

# Now add the environmental variables as arrows
arrowmat_16s_cca <- vegan::scores(cca_ord_16s, display = "bp")

#Get appropiate scalling multipler
mul<-vegan::ordiArrowMul(arrowmat_16s_cca)

#Multiply biplot by scaling multiplier
arrowmat_16s_cca_scale<-arrowmat_16s_cca*2

# Add labels, make a data.frame
arrowdf_16s_cca <- data.frame(labels = rownames(arrowmat_16s_cca_scale), arrowmat_16s_cca_scale)

# Define the arrow aesthetic mapping
arrow_map <- aes(xend = CCA1,
    yend = CCA2,
    x = 0,
    y = 0,
    shape = NULL,
    color = NULL,
    label = labels)
```

```
label_map <- aes(x = 1.3* CCA1,
    y = 1.3 * CCA2,
    shape = NULL,
    color = NULL,
    label = labels)

arrowhead = arrow(length = unit(0.02, "npc"))

# Make a new graphic
cca_plot_16s +
  geom_segment(
    mapping = arrow_map,
    size = .5,
    data = arrowdf_16s_cca,
    color = "gray",
    arrow = arrowhead
  ) +
  geom_text(
    mapping = label_map,
    size = 4,
    data = arrowdf_16s_cca,
    show.legend = FALSE
  )+
  ggtitle("CCA plot constrained ordination of 16s with selected Chemistry Model")+
  theme_bw()
```



RDA is a linear cca and offers a R2 option in OrdiR2step

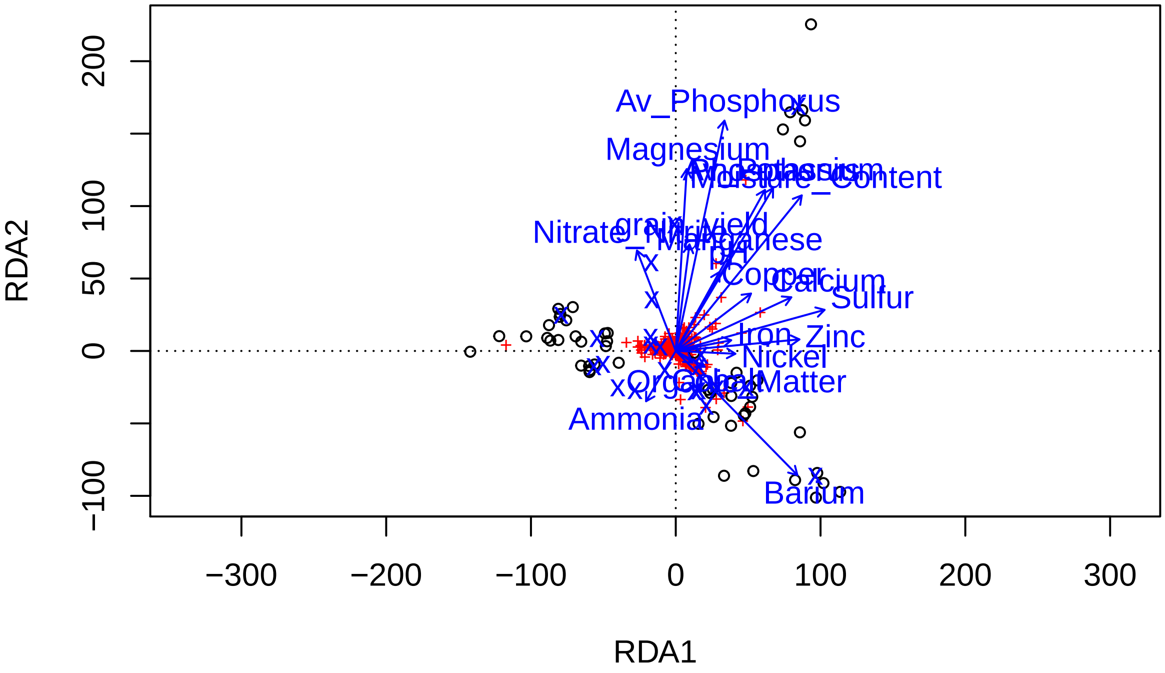
```
m1_16_rda <- rda(abund_table16s ~ ., meta3)
m0_16_rda <- rda(abund_table16s ~ 1, meta3)
m1_16_rda
```

```
## Call: rda(formula = abund_table16s ~ Site + Pea_variety + Plot +
## season_precip + irrgation + total_precip_irr + Tillage + prev_crop
## + grain_yield + Organic_Matter + Moisture_Content +
## Nitrate_Nitrite + Ammonia + Av_Phosphorus + Av_Potassium + pH +
## Barium + Calcium + Cobalt + Copper + Iron + Magnesium + Manganese
## + Nickel + Phosphorus + Sulfur + Zinc, data = meta3)
##
##                 Inertia Proportion Rank
## Total         1.222e+09  1.000e+00
## Constrained   1.061e+09  8.688e-01   33
## Unconstrained 1.603e+08  1.312e-01   20
## Inertia is variance
## Some constraints were aliased because they were collinear (redundant)
##
## Eigenvalues for constrained axes:
##      RDA1      RDA2      RDA3      RDA4      RDA5      RDA6      RDA7
## 310742139 203402945 141163796 102688310  64539375  55815780  44958974
##      RDA8      RDA9     RDA10     RDA11     RDA12     RDA13     RDA14
##  33866810  27564134  17442608   9180065   6205653   5056231   3410341
##     RDA15     RDA16     RDA17     RDA18     RDA19     RDA20     RDA21
##   3380260   3173777   3013165   2623449   2268625   2177296   2052801
##     RDA22     RDA23     RDA24     RDA25     RDA26     RDA27     RDA28
##   1854580   1816607   1640937   1474199   1456017   1398362   1322690
##     RDA29     RDA30     RDA31     RDA32     RDA33
##   1218431   1170995   1113599   1068931   1058113
##
## Eigenvalues for unconstrained axes:
##      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
## 44676851 36203002 21512201 12410457  8113571  6984376  4859210  3504941
## (Showing 8 of 20 unconstrained eigenvalues)
```

```
m0_16_rda
```

```
## Call: rda(formula = abund_table16s ~ 1, data = meta3)
##
##                 Inertia Rank
## Total         1.222e+09
## Unconstrained 1.222e+09   53
## Inertia is variance
##
## Eigenvalues for unconstrained axes:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7
## 333527848 207774138 148081640 115338042  77931145  70371775  54144488
##       PC8
##  39551204
## (Showing 8 of 53 unconstrained eigenvalues)
```

```r
plot(m1_16_rda)
```



```r
model_rda_16s <-ordiR2step(m0_16_rda, scope=formula(m1_16_rda))
```

```r
model_rda_16s$anova
```

	R2.adj	Df	AIC	F	Pr(>F)
+ Site	0.6091540	8	1087.291	11.32541	0.002
	0.6522382	NA	NA	NA	NA

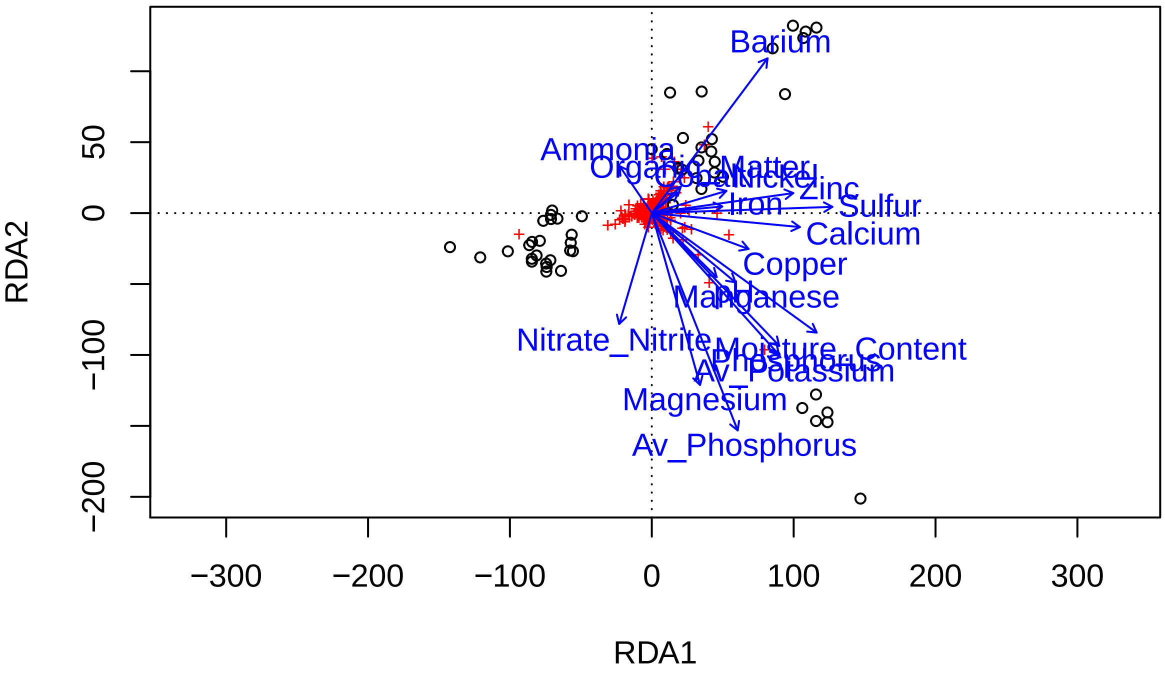
```r
m1_16_rda_chem <- rda(abund_table16s ~ Organic_Matter + Moisture_Content + Nitrate_Nitrite + Ammonia +
m0_16_rda_chem <- rda(abund_table16s ~ 1, meta3)
m1_16_rda_chem
```

```
## Call: rda(formula = abund_table16s ~ Organic_Matter +
## Moisture_Content + Nitrate_Nitrite + Ammonia + Av_Phosphorus +
## Av_Potassium + pH + Barium + Calcium + Cobalt + Copper + Iron +
## Magnesium + Manganese + Nickel + Phosphorus + Sulfur + Zinc, data
## = meta3)
##
##                Inertia Proportion Rank
## Total         1.222e+09  1.000e+00
## Constrained   8.397e+08  6.874e-01   18
## Unconstrained 3.819e+08  3.126e-01   35
## Inertia is variance
##
## Eigenvalues for constrained axes:
##      RDA1      RDA2      RDA3      RDA4      RDA5      RDA6      RDA7
## 247535365 192597084 125062770  94495381  53823699  41434944  34837340
##      RDA8      RDA9     RDA10     RDA11     RDA12     RDA13     RDA14
##  14789649   9836444   7157427   4266682   2937564   2474562   2133052
##     RDA15     RDA16     RDA17     RDA18
##   1879776   1768141   1423320   1276105
##
## Eigenvalues for unconstrained axes:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7
## 127912063  77108629  43267750  17762259  17435253  13308526  12154176
##       PC8
##  10101813
## (Showing 8 of 35 unconstrained eigenvalues)
```

```
m0_16_rda_chem
```

```
## Call: rda(formula = abund_table16s ~ 1, data = meta3)
##
##                  Inertia Rank
## Total          1.222e+09
## Unconstrained 1.222e+09   53
## Inertia is variance
##
## Eigenvalues for unconstrained axes:
##       PC1       PC2       PC3       PC4       PC5       PC6       PC7
## 333527848 207774138 148081640 115338042  77931145  70371775  54144488
##       PC8
##  39551204
## (Showing 8 of 53 unconstrained eigenvalues)
```

```
plot(m1_16_rda_chem)
```



```r
model_rda_chem_16s <-ordiR2step(m0_16_rda_chem, scope=formula(m1_16_rda_chem))
```

Again the above analysis does not consider the site effect but I think we are ok with that because all of the chemcial vriables are "idependent" (sorta). I am not personally as confident in publishing this as I am with the envfit or bioenv since those are in non-metric space and have less correlation to actuall species distance.

```r
model_rda_chem_16s$anova
```

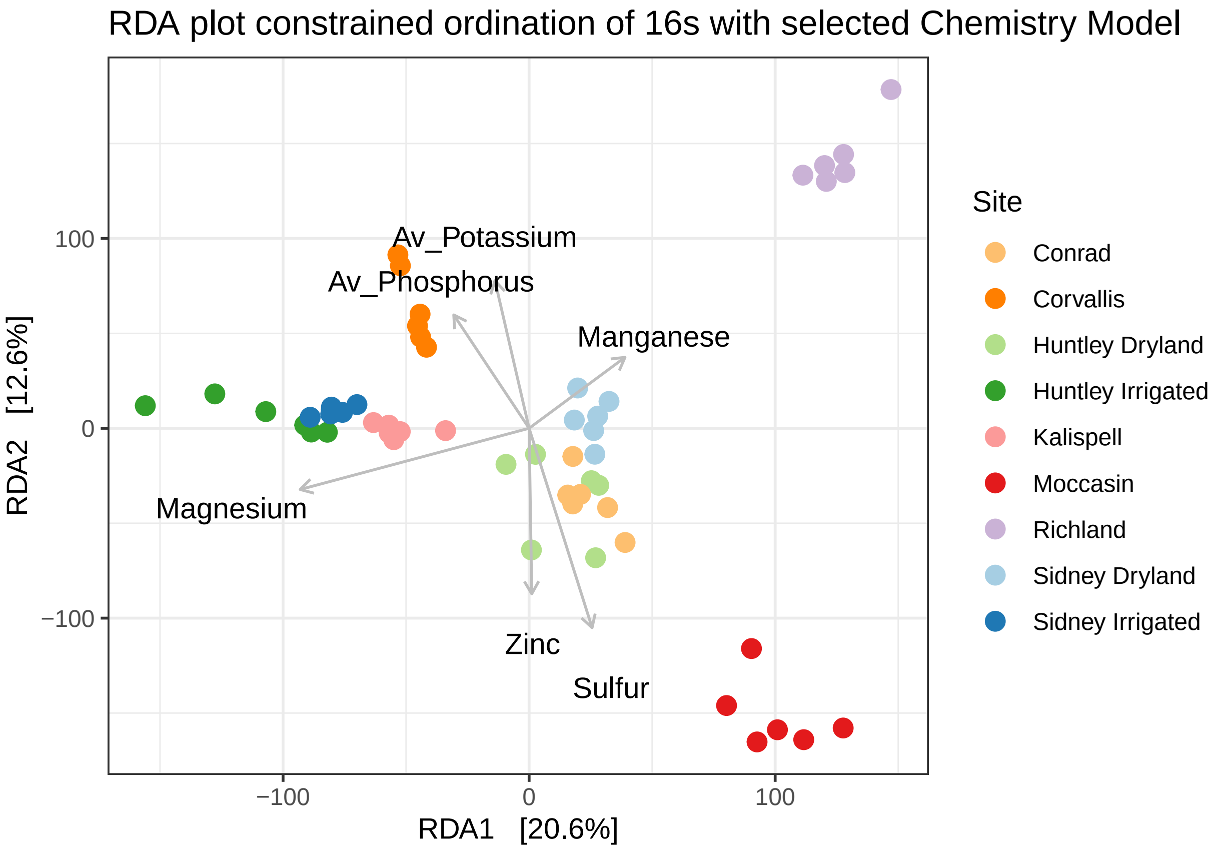
	R2.adj	Df	AIC	F	Pr(>F)
+ Av_Phosphorus	0.1144431	1	1125.266	7.849344	0.002
+ Sulfur	0.2236236	1	1119.112	8.312672	0.002
+ Av_Potassium	0.2912158	1	1115.124	5.863541	0.002
+ Magnesium	0.3678303	1	1109.856	7.059653	0.002
+ Zinc	0.4316879	1	1104.992	6.505816	0.002
+ Manganese	0.4827818	1	1100.768	5.741728	0.002
+ Organic_Matter	0.5189662	1	1097.690	4.535441	0.002
	0.5266051	NA	NA	NA	NA

RDA with selected model from Chemical RDA model

```r
# RDA ordinate
rda_ord_16s <- ordinate(
    physeq = physeq_16s_ord,
    method = "RDA",
    distance = abund_dist16s,
    formula = ~   Av_Phosphorus + Sulfur + Magnesium + Zinc + Av_Potassium + Manganese )

# RDA plot
rda_plot_16s <- plot_ordination(
  physeq = physeq_16s_ord,
  ordination = rda_ord_16s,
    color = "Site",
    axes = c(1,2)) +
    geom_point(aes(colour = Site), size = 3) +
    scale_color_manual(values =  farm_col_paired)

# Now add the environmental variables as arrows
arrowmat_16s_rda <- vegan::scores(rda_ord_16s, display = "bp")

#Get appropiate scalling multipler
mul<-vegan::ordiArrowMul(arrowmat_16s_rda)

#Multiply biplot by scaling multiplier
arrowmat_16s_rda_scale<-arrowmat_16s_rda*150

# Add labels, make a data.frame
arrowdf_16s_rda <- data.frame(labels = rownames(arrowmat_16s_rda_scale), arrowmat_16s_rda_scale)
```

```
# Define the arrow aesthetic mapping
arrow_map <- aes(xend = RDA1,
    yend = RDA2,
    x = 0,
    y = 0,
    shape = NULL,
    color = NULL,
    label = labels)

label_map <- aes(x = 1.3* RDA1,
    y = 1.3 * RDA2,
    shape = NULL,
    color = NULL,
    label = labels)

arrowhead = arrow(length = unit(0.02, "npc"))

# Make a new graphic
rda_plot_16s +
  geom_segment(
    mapping = arrow_map,
    size = .5,
    data = arrowdf_16s_rda,
    color = "gray",
    arrow = arrowhead
  ) +
  geom_text(
    mapping = label_map,
    size = 4,
    data = arrowdf_16s_rda,
    show.legend = FALSE
  )+
  ggtitle("RDA plot constrained ordination of 16s with selected Chemistry Model")+
  theme_bw()
```

```
## Warning: Ignoring unknown aesthetics: label
```



####Removed Organic matter since the ordination was so low.

```
## Warning: Ignoring unknown aesthetics: label
```

```
## pdf
##   2
```

CAP model building

canonical analysis of principal coordinates (CAP) is similar to RDA but allows for non-euclidian dissimilarity like Bray-Curtis which we have been using.

https://esajournals.onlinelibrary.wiley.com/doi/epdf/10.1890/0012-9658%282003%29084%5B0511%3ACAOPCA%5D2.0.CO%3B2

Guide for the CAP Ordination

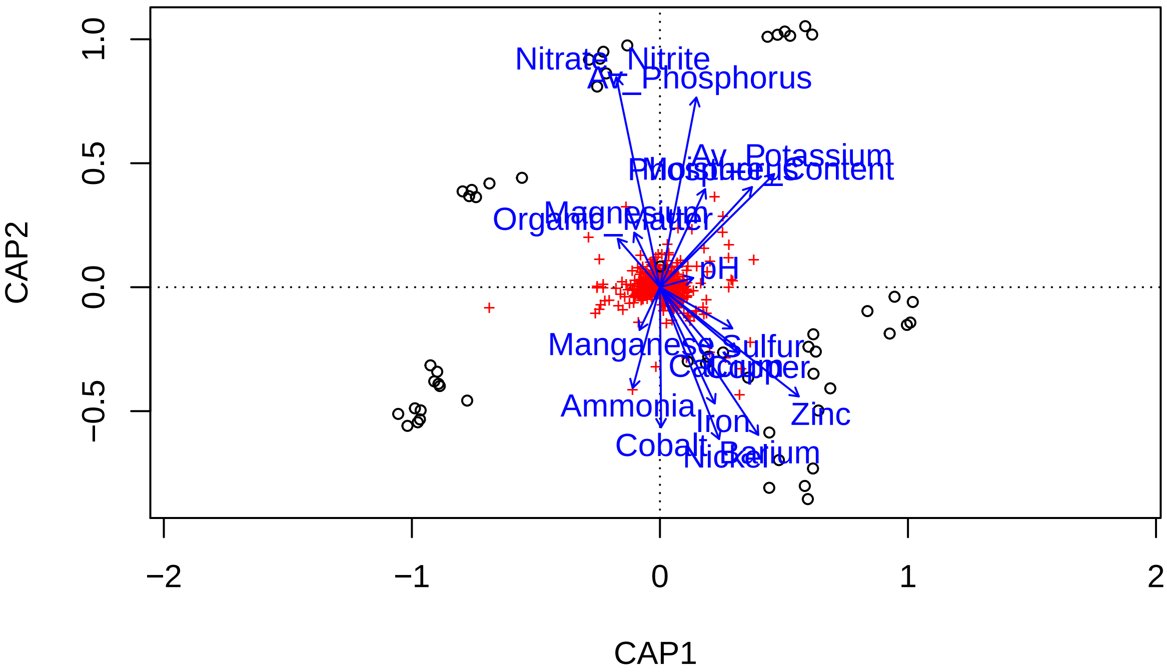
https://quantpalaeo.wordpress.com/2014/04/14/variance-inflation-factors-and-ordination-model-selection/

If the VIF of a predictor is high, it indicates that that predictor is highly correlated with other predictors, it contains little or no unique information, and there is redundancy in the set of predictors.

STEPS

```
1)Generate a constrained ordination with all available predictors.
2)Calculate the VIF of each variable.
3)If any variable has a VIF over a threshold (typically 10), drop the variable with the highest VIF
4)Repeat until all remaining variables have a VIF below the threshold.
```

```
m1_16s_cap_chem<- capscale(abund_table16s ~ Organic_Matter + Moisture_Content + Nitrate_Nitrite + Ammoni
m0_16s_cap_chem<- capscale(abund_table16s ~ 1, meta3)
plot(m1_16s_cap_chem)
```



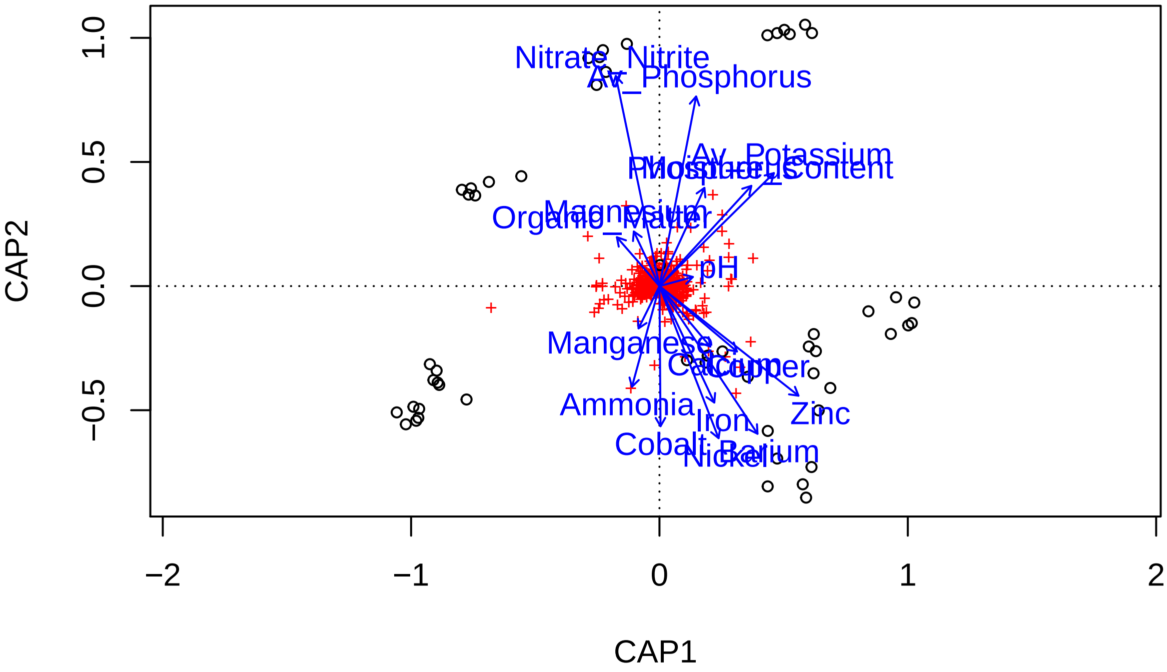
```
vif.cca(m1_16s_cap_chem)
```

```
##   Organic_Matter Moisture_Content  Nitrate_Nitrite          Ammonia
##        15.457094         8.111272         4.360780         4.369015
##    Av_Phosphorus     Av_Potassium               pH           Barium
##        11.821766        13.816708        19.007192        34.110255
##          Calcium           Cobalt           Copper             Iron
##        69.959837        20.726173        37.675483        55.160314
##        Magnesium        Manganese           Nickel       Phosphorus
##        38.155344        25.416138        39.209635        37.072036
##           Sulfur             Zinc
##       126.793579        83.408013
```

Lots of high VIF scores, this might take a while,

Droping Sulfur from the model

```
m1_16s_cap_chem_1<- capscale(abund_table16s ~ Organic_Matter + Moisture_Content + Nitrate_Nitrite + Ammo
m0_16s_cap_chem<- capscale(abund_table16s ~ 1, meta3)
plot(m1_16s_cap_chem_1)
```

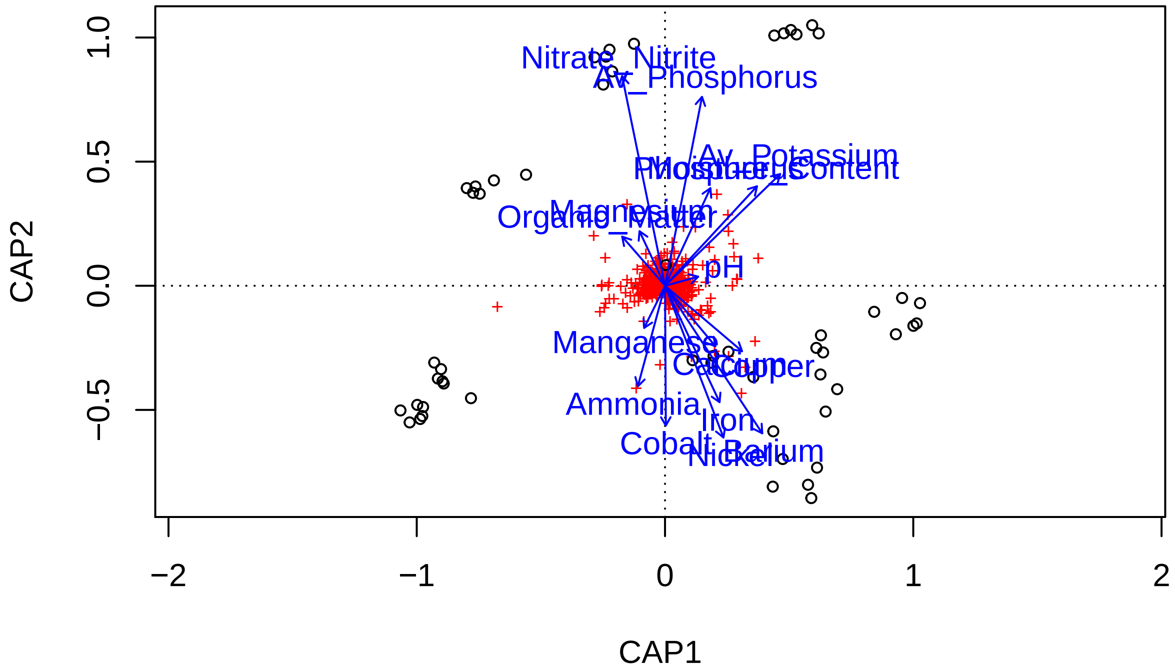


```
vif.cca(m1_16s_cap_chem_1)
```

```
##   Organic_Matter Moisture_Content  Nitrate_Nitrite          Ammonia
##        14.498699         6.988162         3.713414         4.309993
##    Av_Phosphorus     Av_Potassium               pH           Barium
##        11.350534        12.916393        16.155227        31.533526
##          Calcium           Cobalt           Copper             Iron
##         9.761153        15.971211        35.302426        53.661315
##        Magnesium        Manganese           Nickel       Phosphorus
##        31.551213        17.840002        36.191208        28.682249
##             Zinc
##        80.556370
```

Dropping Zinc from the model

```
m1_16s_cap_chem_2<- capscale(abund_table16s ~ Organic_Matter + Moisture_Content + Nitrate_Nitrite + Ammo
m0_16s_cap_chem<- capscale(abund_table16s ~ 1, meta3)
plot(m1_16s_cap_chem_2)
```

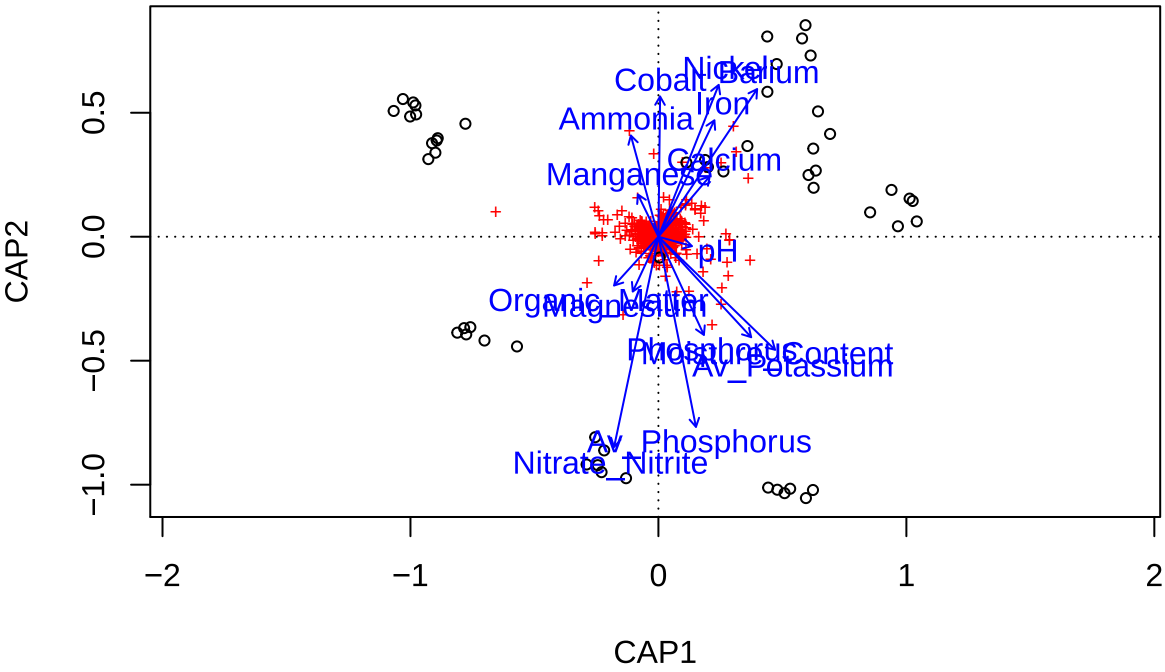


```
vif.cca(m1_16s_cap_chem_2)
```

```
##   Organic_Matter Moisture_Content   Nitrate_Nitrite           Ammonia
##        12.646738         5.847957          3.672388          4.309848
##    Av_Phosphorus     Av_Potassium                pH            Barium
##        11.225914        10.884874         15.913142         18.081149
##          Calcium           Cobalt            Copper              Iron
##         9.388663        14.920297         31.854122         41.065111
##        Magnesium        Manganese            Nickel        Phosphorus
##        31.536509        16.206354         29.523059         28.553042
```

Dropping Copper

```
m1_16s_cap_chem_3<- capscale(abund_table16s ~ Organic_Matter + Moisture_Content + Nitrate_Nitrite + Amm
m0_16s_cap_chem<- capscale(abund_table16s ~ 1, meta3)
plot(m1_16s_cap_chem_3)
```

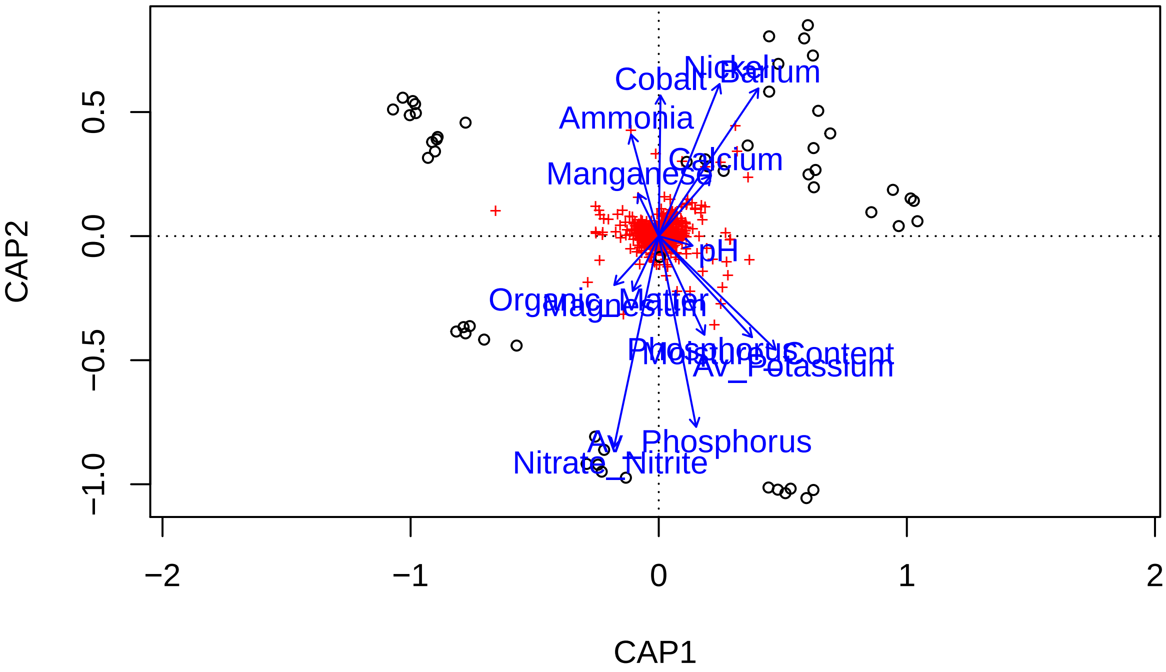


```
vif.cca(m1_16s_cap_chem_3)
```

```
##    Organic_Matter Moisture_Content   Nitrate_Nitrite          Ammonia
##         12.291375         5.597957          3.497928         4.299572
##     Av_Phosphorus     Av_Potassium                pH           Barium
##          9.988488        10.879178         15.756696        11.018789
##           Calcium           Cobalt              Iron        Magnesium
##          8.965312        14.025547         39.394900        23.081257
##         Manganese           Nickel        Phosphorus
##         11.259872        28.212639         28.271886
```

Dropping Iron

```
m1_16s_cap_chem_4<- capscale(abund_table16s ~ Organic_Matter + Moisture_Content + Nitrate_Nitrite + Amm
m0_16s_cap_chem<- capscale(abund_table16s ~ 1, meta3)
plot(m1_16s_cap_chem_4)
```

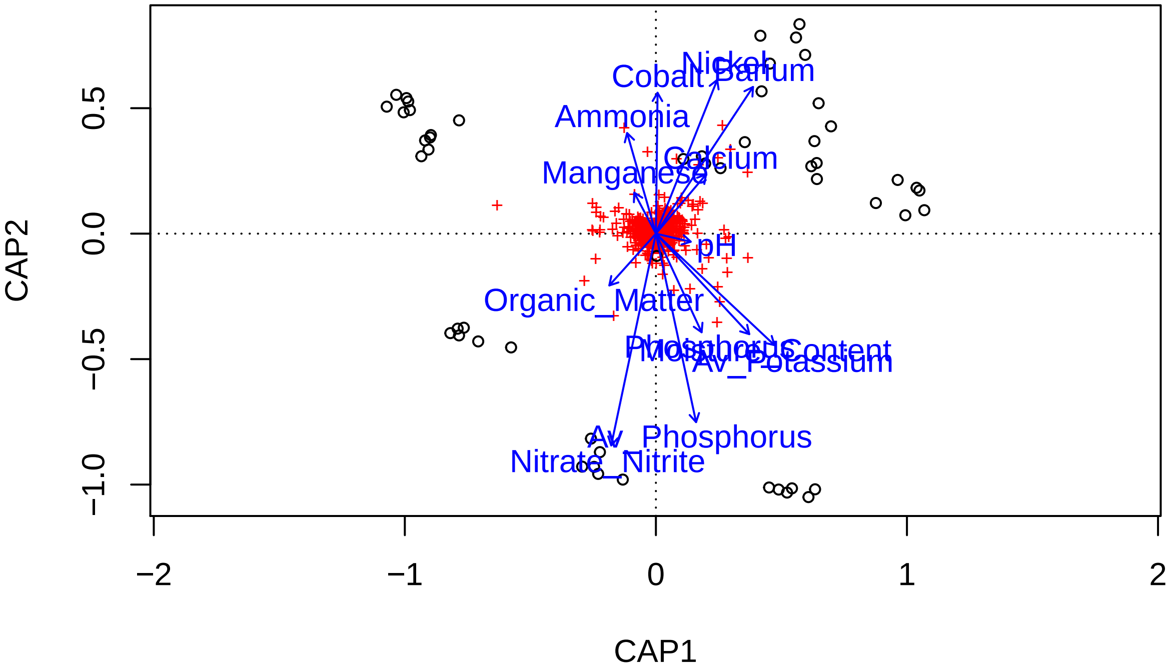


```
vif.cca(m1_16s_cap_chem_4)
```

```
##    Organic_Matter Moisture_Content   Nitrate_Nitrite          Ammonia
##          9.382412         5.403654          3.279764         4.290460
##     Av_Phosphorus     Av_Potassium                pH           Barium
##          9.817094        10.719638         14.623835         9.620284
##           Calcium           Cobalt         Magnesium        Manganese
##          6.023507        11.675665         22.563802        10.794157
##            Nickel       Phosphorus
##         21.922046        16.432692
```

Removing Magnesium Getting close!

```
m1_16s_cap_chem_5<- capscale(abund_table16s ~ Organic_Matter + Moisture_Content + Nitrate_Nitrite + Ammo
m0_16s_cap_chem<- capscale(abund_table16s ~ 1, meta3)
plot(m1_16s_cap_chem_5)
```

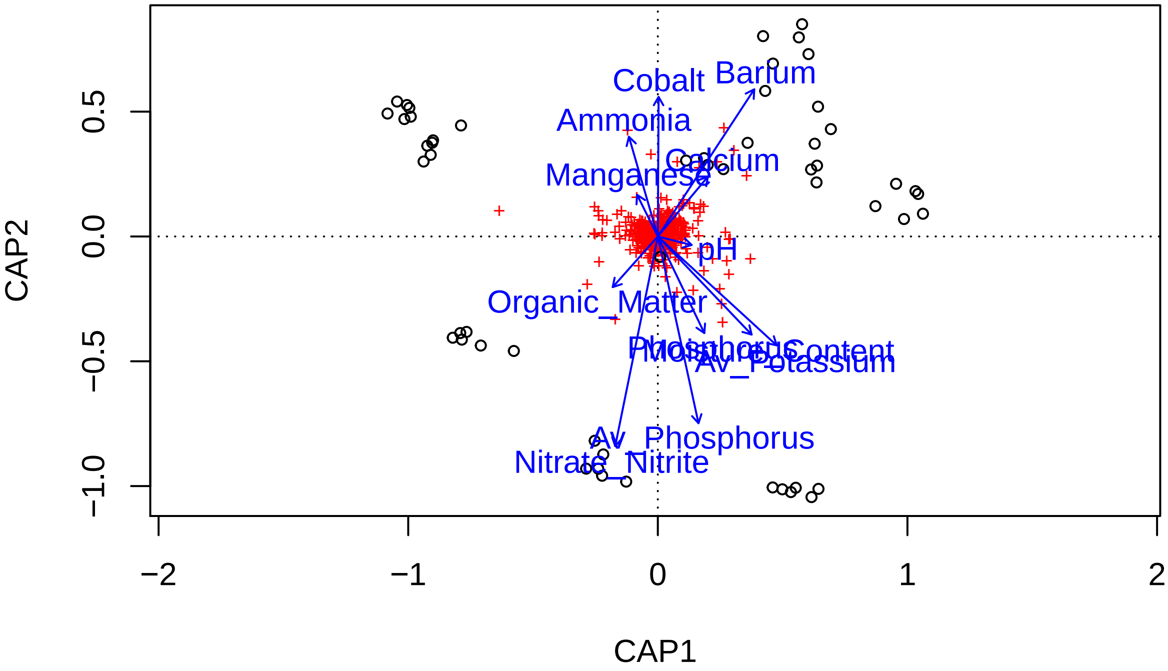


```r
vif.cca(m1_16s_cap_chem_5)
```

```
##   Organic_Matter Moisture_Content  Nitrate_Nitrite          Ammonia
##         9.280442         5.335735         3.115965         4.278840
##    Av_Phosphorus     Av_Potassium               pH           Barium
##         9.280456         9.657791        12.581736         5.989495
##          Calcium           Cobalt        Manganese           Nickel
##         4.196095        11.661999         7.354337        18.320162
##       Phosphorus
##        10.931904
```

Dropping Nickel

```r
m1_16s_cap_chem_6<- capscale(abund_table16s ~ Organic_Matter + Moisture_Content + Nitrate_Nitrite + Amm
m0_16s_cap_chem<- capscale(abund_table16s ~ 1, meta3)
plot(m1_16s_cap_chem_6)
```

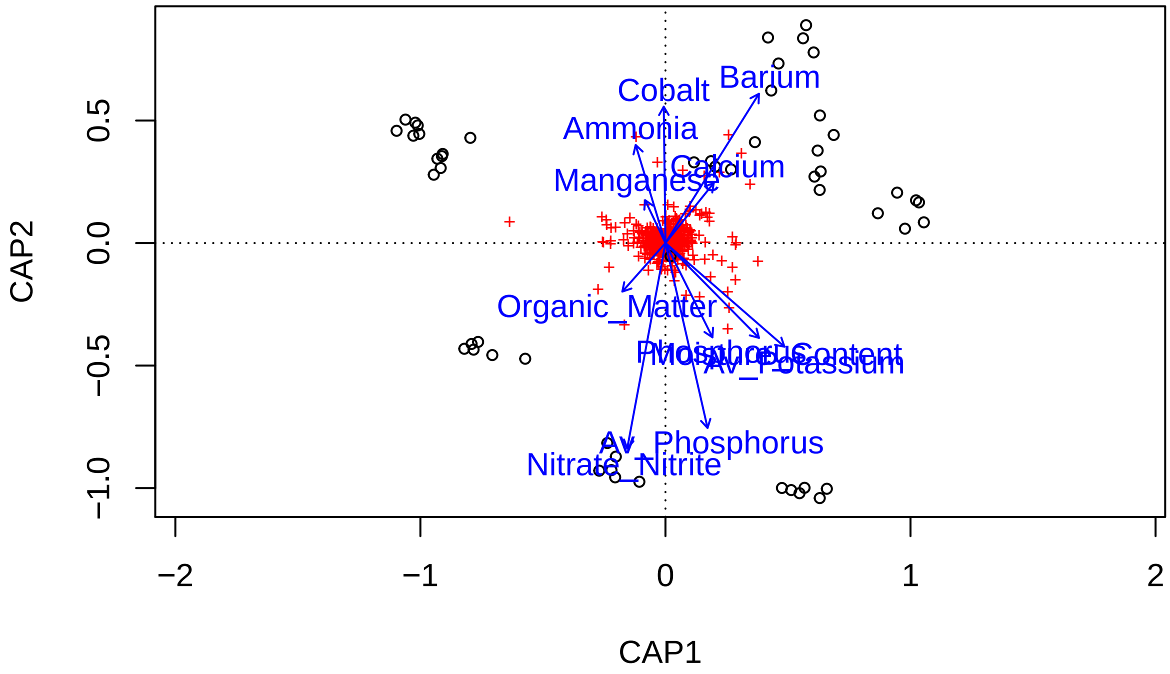


```r
vif.cca(m1_16s_cap_chem_6)
```

```
##    Organic_Matter Moisture_Content  Nitrate_Nitrite          Ammonia
##          5.450452         5.201185         3.037290         4.048144
##     Av_Phosphorus     Av_Potassium               pH           Barium
##          6.938227         8.023786        11.023209         3.787367
##           Calcium           Cobalt        Manganese       Phosphorus
##          3.190323         7.915189         7.350927        10.302951
```

Droping pH

```r
m1_16s_cap_chem_7<- capscale(abund_table16s ~ Organic_Matter + Moisture_Content + Nitrate_Nitrite + Ammo
m0_16s_cap_chem<- capscale(abund_table16s ~ 1, meta3)
plot(m1_16s_cap_chem_7)
```



```
vif.cca(m1_16s_cap_chem_7)
```

```
##   Organic_Matter Moisture_Content   Nitrate_Nitrite          Ammonia
##         5.450062         4.277670         2.403275         4.041205
##     Av_Phosphorus     Av_Potassium           Barium          Calcium
##         5.760670         6.150896         3.501933         2.850287
##           Cobalt        Manganese       Phosphorus
##         6.357215         4.269795         8.053834
```

Everything under 10! We will put these variables into the model building because it will not have co lineraity

```
model_cap_chem_16s <-ordiR2step(m0_16s_cap_chem, scope=formula(m1_16s_cap_chem_7))
```

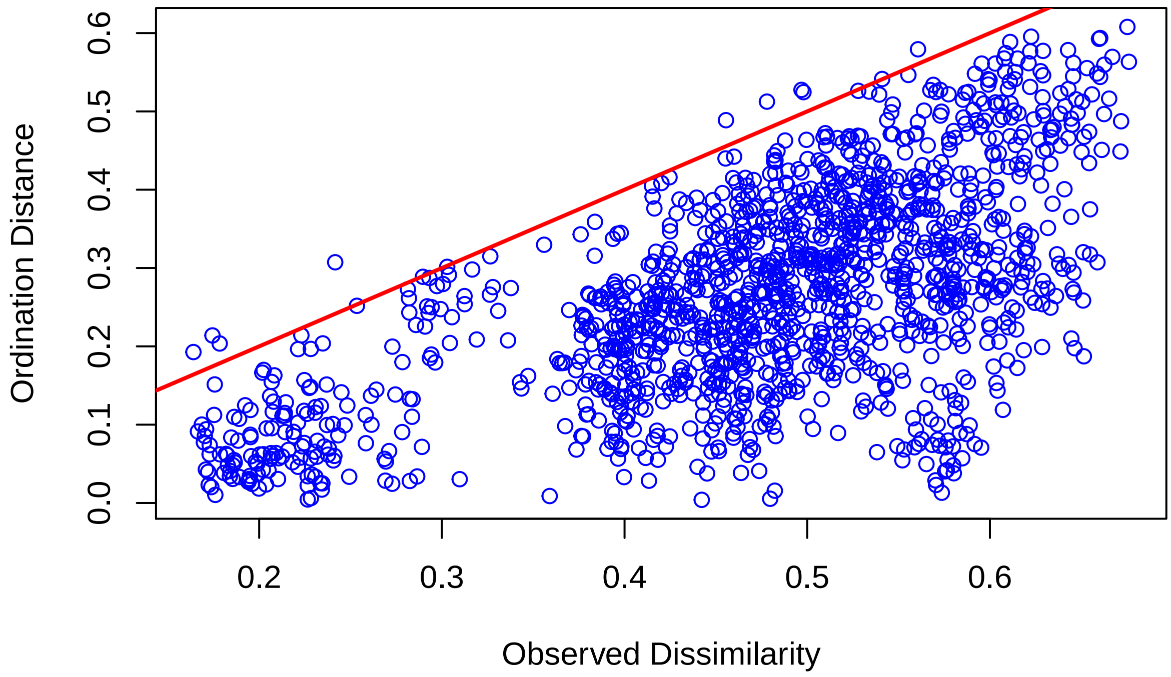
```
aov_model_cap_chem_16s<-model_cap_chem_16s$anova
aov_model_cap_chem_16s
```

	R2.adj	Df	AIC	F	Pr(>F)
+ Av_Phosphorus	0.1144431	1	1125.266	7.849344	0.002
+ Barium	0.2016805	1	1120.617	6.682370	0.002
+ Calcium	0.2622274	1	1117.288	5.185422	0.002
+ Av_Potassium	0.3254256	1	1113.361	5.684299	0.002
+ Manganese	0.4179832	1	1106.279	8.792428	0.002
+ Organic_Matter	0.4640796	1	1102.686	5.128654	0.002
+ Nitrate_Nitrite	0.4902114	1	1100.825	3.409218	0.012
+ Phosphorus	0.5150213	1	1098.944	3.353210	0.006
	0.5262981	NA	NA	NA	NA

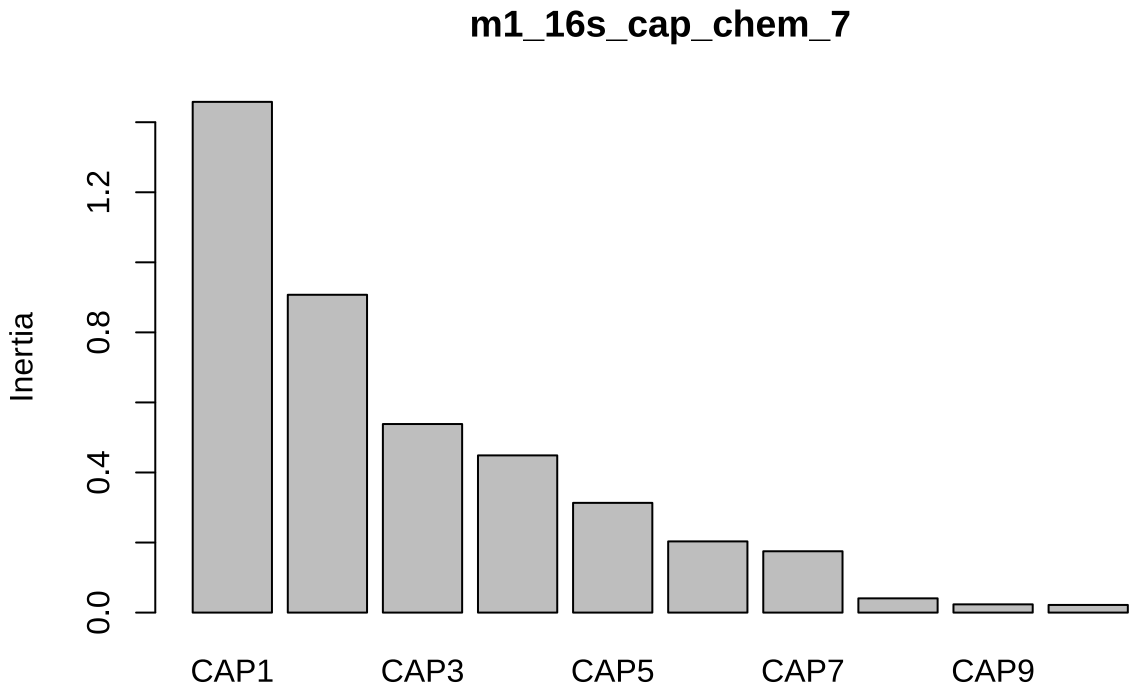
```r
capture.output(aov_model_cap_chem_16s,file="aov_model_cap_chem_16s.txt")
```

Our CAP Model explains 51% of the variance in our Bray-Curtis distances

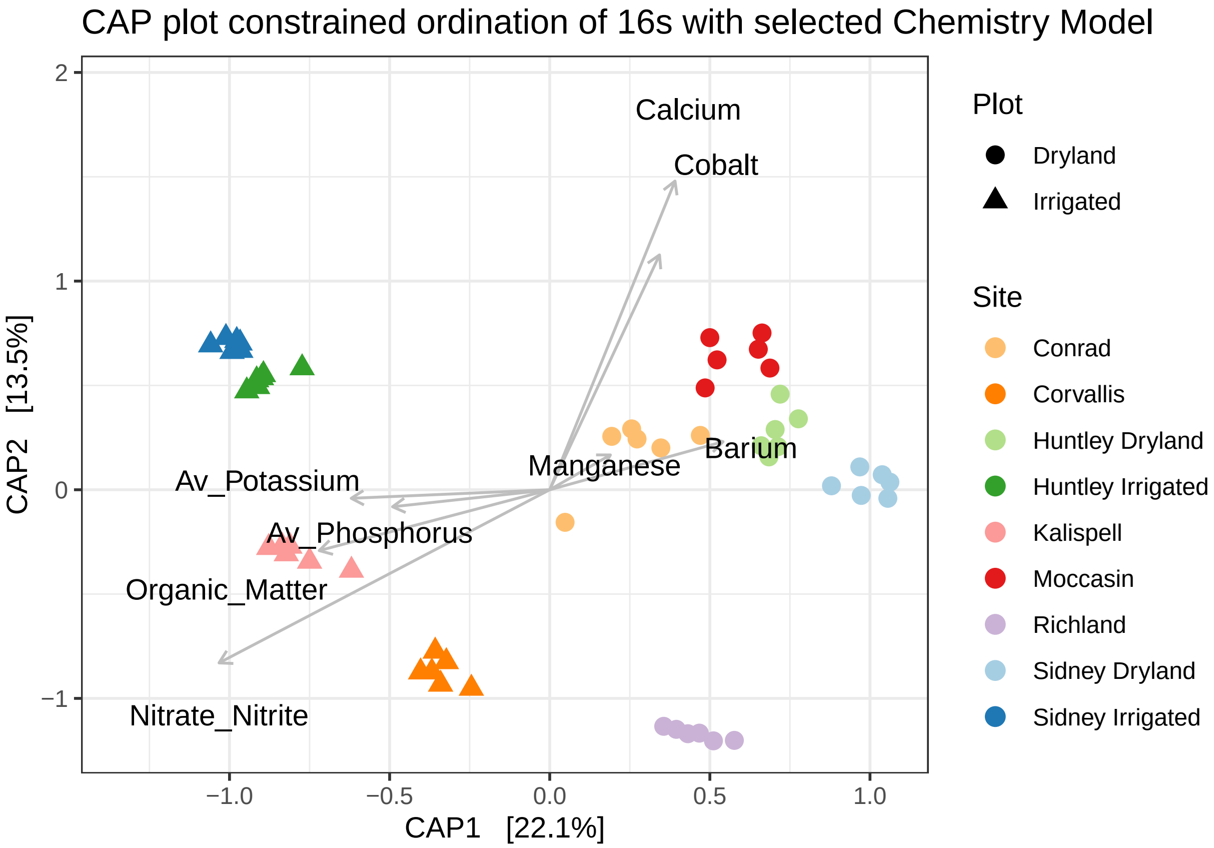
```r
stressplot(m1_16s_cap_chem_7)
```



```r
screeplot(m1_16s_cap_chem_7)
```



```
# CAP ordinate
cap_ord_16s <- ordinate(
    physeq = physeq_16s_ord,
    method = "CAP",
    distance = abund_dist16s,
    formula = ~   Av_Phosphorus + Barium + Calcium + Av_Potassium +
      Manganese + Organic_Matter + Cobalt + Nitrate_Nitrite  )

# CCA plot
cap_plot_16s <- plot_ordination(
  physeq = physeq_16s_ord,
  ordination = cap_ord_16s,
    color = "Site",
    shape = "Plot",
    axes = c(1,2)) +
    geom_point(aes(colour = Site), size = 3) +
    scale_color_manual(values =  farm_col_paired)

# Now add the environmental variables as arrows
arrowmat_16s_cap <- vegan::scores(cap_ord_16s, display = "bp")

#Get appropiate scalling multipler
mul<-vegan::ordiArrowMul(arrowmat_16s_cap)

#Multiply biplot by scaling multiplier
arrowmat_16s_cap_scale<-arrowmat_16s_cap*1.9
# Add labels, make a data.frame
arrowdf_16s_cap <- data.frame(labels = rownames(arrowmat_16s_cap_scale), arrowmat_16s_cap_scale)

# Define the arrow aesthetic mapping
arrow_map <- aes(xend = CAP1,
    yend = CAP2,
    x = 0,
    y = 0,
    shape = NULL,
    color = NULL,
    label = labels)

label_map <- aes(x = 1.3* CAP1,
    y = 1.3 * CAP2,
    shape = NULL,
    color = NULL,
    label = labels)

arrowhead = arrow(length = unit(0.02, "npc"))

# Make a new graphic
cap_plot_16s +
  geom_segment(
    mapping = arrow_map,
    size = .5,
    data = arrowdf_16s_cap,
    color = "gray",
    arrow = arrowhead
  ) +
  geom_text_repel(
    mapping = label_map,
    size = 4,
    data = arrowdf_16s_cap,
    show.legend = FALSE
  )+
  ggtitle("CAP plot constrained ordination of 16s with selected Chemistry Model")+
  theme_bw()
```



Publish to tiff

```
## Warning: Ignoring unknown aesthetics: label

## pdf
##   2
```

Fitting species to cap plot

```
m1_16s_cap_chem_species<- capscale(abund_table16s ~ Av_Phosphorus + Barium + Calcium + Av_Potassium +
      Manganese + Organic_Matter + Cobalt + Nitrate_Nitrite, data = meta3, distance = "bray")
dims=c(1,2)
```

```r
site=scores(m1_16s_cap_chem_species,display="wa",choices=dims)
cor.min=0.9 #below this threshold, arrows will be not plotted
#because orrelation is considered too much week
cor_sp=as.data.frame(scores(m1_16s_cap_chem_species, dis="sp", scaling=1,choices=dims))
cor_sp$cor=with(cor_sp,sqrt(CAP1^2+CAP2^2))
cor_sp$sup=FALSE;cor_sp$sup[cor_sp$cor>=cor.min]<-TRUE
cor_sp$labels=row.names(cor_sp)
cor_sp=cor_sp[cor_sp$sup==TRUE,]
cor_sp_s1=cor_sp
```

```r
tax_16s_cap<- read.delim("~/Alex Alleman/Statewide Microbiome Analysis/Statewide analysis/16s_OTU_ids_2
head(tax_16s)[,1:8]
```

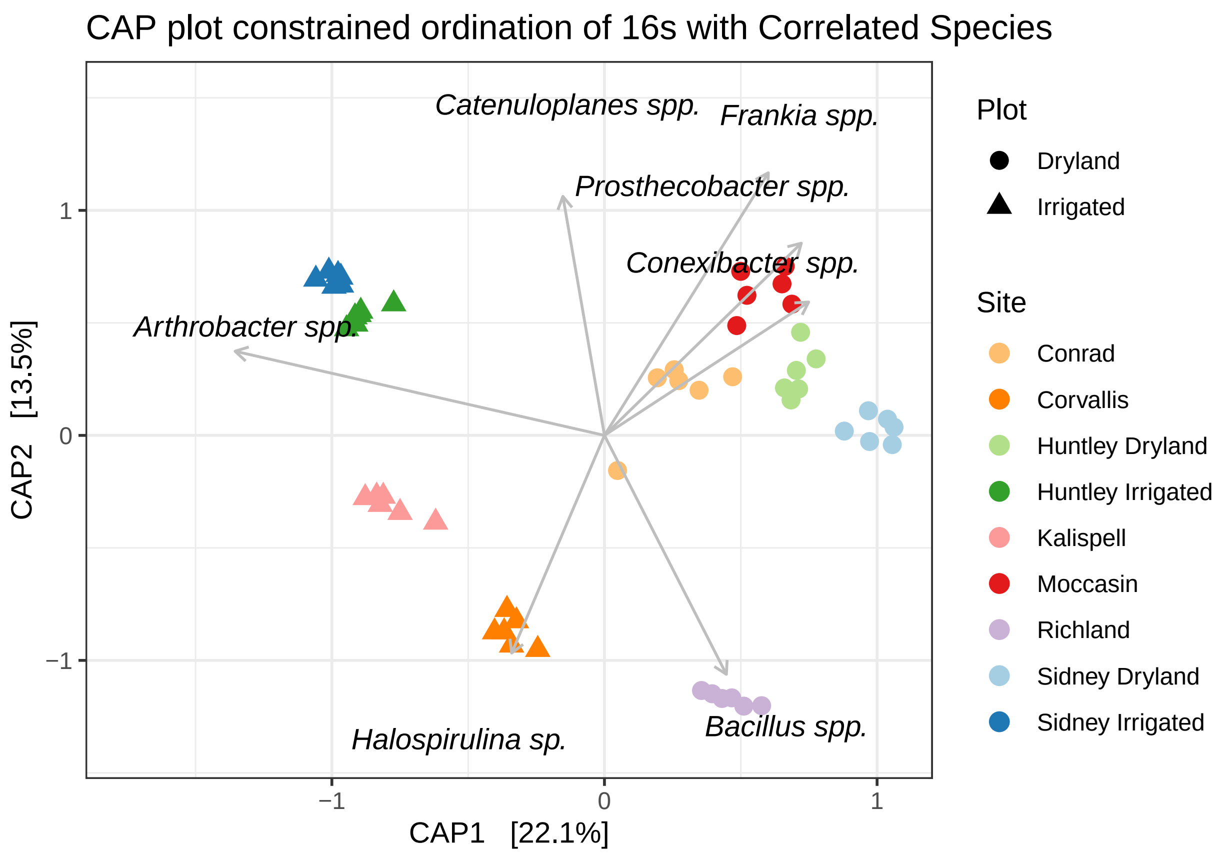
	kingdom	phylum	class	order	family	genu
OTU2	k___bacteria	p___actinobacteria	c___actinobacteria	o___gaiellales	f___gaiellaceae	g___g
OTU7	k___bacteria	p___acidobacteria	c___acidobacteriia	o___acidobacteriales	f___acidobacteriaceae	g___a
OTU8	k___bacteria	p___actinobacteria	c___actinobacteria	o___actinomycetales	f___geodermatophilaceae	g___
OTU10	k___bacteria	p___actinobacteria	c___actinobacteria	o___actinomycetales	f___geodermatophilaceae	g___g
OTU11	k___bacteria	p___actinobacteria	c___actinobacteria	o___actinomycetales	f___mycobacteriaceae	g___
OTU12	k___bacteria	p___actinobacteria	c___actinobacteria	o___actinomycetales	f___microbacteriaceae	g___a

```r
#use perl scirpt to remove teh g_
tax_16s_cap$species<- sub(".*_", "", tax_16s_cap$species)
#use the function captilize to captilize first letter
tax_16s_cap$species<- capitalize(tax_16s_cap$species)

species_16s<-merge(cor_sp_s1, tax_16s_cap, by="row.names")
```

Replot with species

```r
# CAP ordinate
cap_ord_16s <- ordinate(
    physeq = physeq_16s_ord,
    method = "CAP",
    distance = abund_dist16s,
    formula = ~   Av_Phosphorus + Barium + Calcium + Av_Potassium +
      Manganese + Organic_Matter + Cobalt + Nitrate_Nitrite  )

# CCA plot
cap_plot_16s <- plot_ordination(
  physeq = physeq_16s_ord,
  ordination = cap_ord_16s,
    color = "Site",
    shape = "Plot",
    axes = c(1,2)) +
    geom_point(aes(colour = Site), size = 3) +
    scale_color_manual(values =  farm_col_paired)



# Define the arrow aesthetic mapping
```

```r
arrow_map <- aes(xend = CAP1,
    yend = CAP2,
    x = 0,
    y = 0,
    shape = NULL,
    color = NULL,
    label = species)

label_map <- aes(x = 1.3* CAP1,
    y = 1.3 * CAP2,
    shape = NULL,
    color = NULL,
    label = species)

arrowhead = arrow(length = unit(0.02, "npc"))

# Make a new graphic
cap_plot_16s +
  geom_segment(
    mapping = arrow_map,
    size = .5,
    data = species_16s,
    color = "gray",
    arrow = arrowhead
  ) +
  geom_text_repel(
    mapping = label_map,
    size = 4,
    data = species_16s,
    show.legend = FALSE,
    fontface="italic"
  )+
  ggtitle("CAP plot constrained ordination of 16s with Correlated Species")+
  theme_bw()
```



Publish to tiff

```
## Warning: Ignoring unknown aesthetics: label
```

```
## pdf
##   2
```