nifH_irr_dry_analysis

Alexander Alleman

March 18, 2019

```
## [1] "2019-10-31-10-25-00"
```

```
## [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 <- brewer.pal(10, "Paired")
```

Load OTU, Taxa, and Meta data

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

```
OTU_nifH <- read.delim("~/Alex Alleman/Statewide Microbiome Analysis/Statewide analysis/nifH_OTUallspri
    row.names = 1)
head(OTU_nifH)[, 1:10]
```

	JZ017	JZ018	JZ019	JZ020	JZ021	JZ022	JZ023	JZ024	JZ025	JZ026
OTU1	21	32923	48199	22	117	103	147	4054	5597	246
OTU2	35281	5	15	1075	101	1335	26249	376	1705	256
OTU3	11	14	13	49	20	48	9	8	16	9
OTU4	80	4	14	2576	144	136	150	129	29299	77807
OTU5	6	2	7	146	749	526	432	1115	13256	0
OTU6	19317	11274	799	748	1532	348	32	21	28	2517

These taxa were also created by Mr. DNA through a blast program and the NCBI database

```
tax_nifH <- read.delim("~/Alex Alleman/Statewide Microbiome Analysis/Statewide analysis/nifH_OTU_ids_20
    row.names = 1)
head(tax_nifH)[, 1:8]
```

	kingdom	phylum	class	order	family	genus
OTU1	k___bacteria	p___firmicutes	c___bacilli	o___bacillales	f___paenibacillaceae	g___pae
OTU2	k___bacteria	p___firmicutes	c___bacilli	o___bacillales	f___paenibacillaceae	g___pae
OTU3	k___bacteria	p___proteobacteria	c___alphaproteobacteria	o___rhizobiales	f___rhizobiaceae	g___rhiz
OTU4	k___bacteria	p___firmicutes	c___bacilli	o___bacillales	f___paenibacillaceae	g___pae
OTU5	k___bacteria	p___proteobacteria	c___alphaproteobacteria	o___rhizobiales	f___bradyrhizobiaceae	g___bra
OTU6	k___bacteria	p___proteobacteria	c___alphaproteobacteria	o___rhizobiales	f___bradyrhizobiaceae	g___rho

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

```
meta_irr <- read.delim("~/Alex Alleman/Statewide Microbiome Analysis/Statewide analysis/all_metadata_su
    row.names = 1, na.strings = "NA", sep = ",", colClasses = c(rep("factor", 7), rep("numeric", 3),
        rep("factor", 4), "numeric", rep("factor", 3), rep("numeric", 28)))
head(meta_irr)[, 1:5]
```

	Site	ARC	Season	Sample_dates	Pea_variety
JZ040	Huntley Dryland	SARC	Summer	8/5/2016	Delta
JZ046	Huntley Dryland	SARC	Summer	8/5/2016	Navarro
JZ042	Huntley Dryland	SARC	Summer	8/5/2016	DS Admiral
JZ044	Huntley Dryland	SARC	Summer	8/5/2016	Majoret
JZ038	Huntley Dryland	SARC	Summer	8/5/2016	CDC Saffron
JZ036	Huntley Dryland	SARC	Summer	8/5/2016	AC Earlystar

Convert to matrix

```
OTU_nifH_m <- as.matrix(OTU_nifH)
tax_nifH_m <- as.matrix(tax_nifH)
meta_m <- as.matrix(meta_irr)
```

```
class(OTU_nifH_m)
```

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

```
class(tax_nifH_m)
```

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

```
class(meta_m)
```

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

```
OTUnifH = otu_table(OTU_nifH_m, taxa_are_rows = TRUE)
TAXnifH = tax_table(tax_nifH_m)

physeq_nifH = phyloseq(OTUnifH, TAXnifH)
```

Get physeq info

```
physeq_nifH
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 8821 taxa and 101 samples ]
## tax_table()   Taxonomy Table:    [ 8821 taxa by 8 taxonomic ranks ]
```

Add meta data to both phyoseq

```
meta_phy_irr <- sample_data(meta_irr)
sample_names(meta_phy_irr)
```

```
##  [1] "JZ040" "JZ046" "JZ042" "JZ044" "JZ038" "JZ036" "JZ041" "JZ047"
##  [9] "JZ043" "JZ045" "JZ039" "JZ037" "JZ084" "JZ085" "JZ086" "JZ087"
## [17] "JZ088" "JZ089" "JZ090" "JZ091" "JZ092" "JZ093" "JZ094" "JZ095"
```

```
physeq_nifH_irr <- merge_phyloseq(physeq_nifH, meta_phy_irr)
physeq_nifH_irr
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 8821 taxa and 24 samples ]
## sample_data() Sample Data:       [ 24 samples by 45 sample variables ]
## tax_table()   Taxonomy Table:    [ 8821 taxa by 8 taxonomic ranks ]
```

Rarefiy data

```
physeq_nifH_irr <- rarefy_even_depth(physeq_nifH_irr)
```

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

```
## ...
```

```
## 4705OTUs were removed because they are no longer
## present in any sample after random subsampling
```

```
## ...
```

```
physeq_nifH_irr
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 4116 taxa and 24 samples ]
## sample_data() Sample Data:       [ 24 samples by 45 sample variables ]
## tax_table()   Taxonomy Table:    [ 4116 taxa by 8 taxonomic ranks ]
```

Trimming

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

```
physeq_nifH_irr_trim = filter_taxa(physeq_nifH_irr, function(x) sum(x > 3) > (0.2 * length(x)), TRUE)
physeq_nifH_irr_trim
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 489 taxa and 24 samples ]
## sample_data() Sample Data:       [ 24 samples by 45 sample variables ]
## tax_table()   Taxonomy Table:    [ 489 taxa by 8 taxonomic ranks ]
```

We have removed the majorty of the low abundance data with a remaining 3596 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_nifH_irr_ord = transform_sample_counts(physeq_nifH_irr_trim, function(x) x/sum(x))
physeq_nifH_irr_ord_phylum <- tax_glom(physeq_nifH_irr_ord, "phylum")
data_nifH_irr_phylum <- psmelt(physeq_nifH_irr_ord_phylum)
data_nifH_irr_phylum$phylum <- as.character(data_nifH_irr_phylum$phylum)
data_nifH_irr_phylum$phylum[data_nifH_irr_phylum$Abundance < 0.05] <- "<5% abdund"
count <- length(unique(data_nifH_irr_phylum$phylum))
count
```

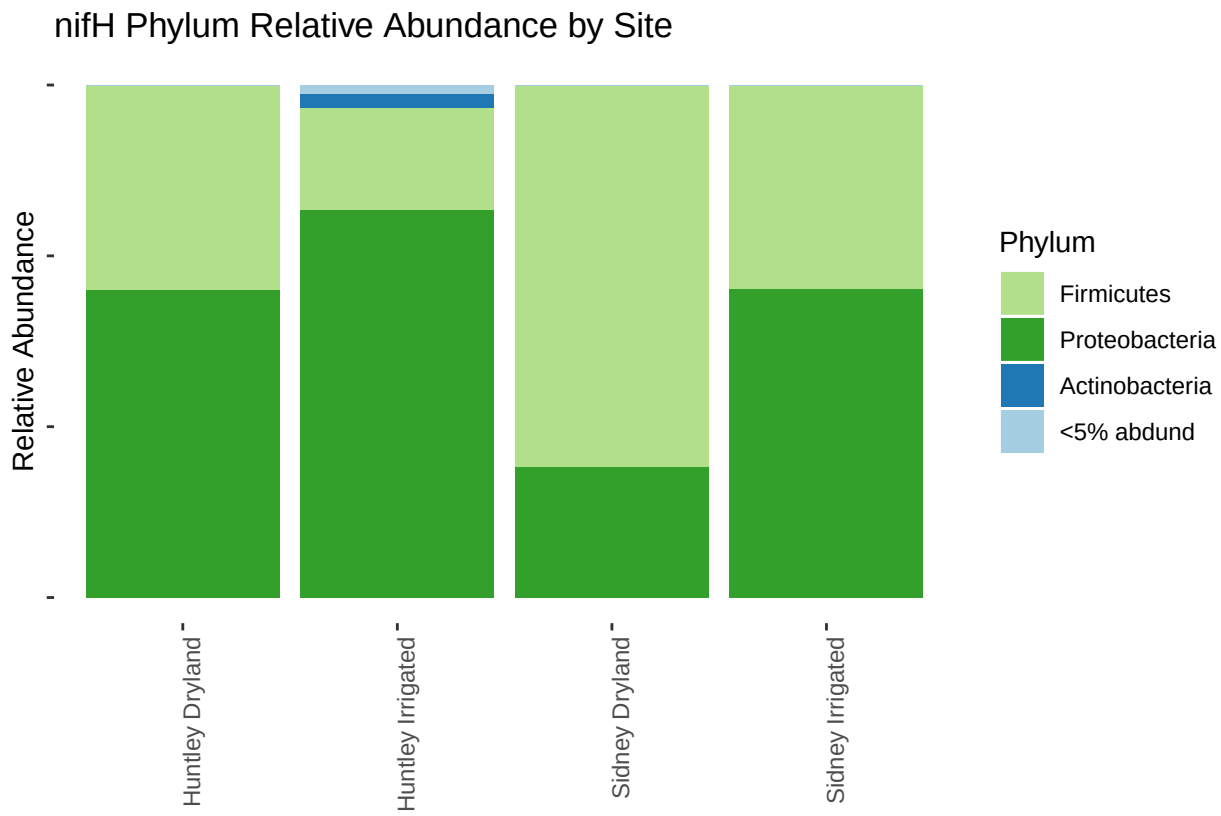
```
## [1] 4
```

```
unique(data_nifH_irr_phylum$phylum)
```

```
## [1] "p__firmicutes"     "p__proteobacteria" "p__actinobacteria"
## [4] "<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_nifH_irr_phylum, aes(x = Site, y = Abundance, fill = phylum)) + geom_bar(aes(fill = p
    stat = "identity", position = "stack", show.legend = TRUE) + scale_fill_manual(name = "Phylum", valu
    "#1f78b4", "#b2df8a", "#33a02c"), breaks = c("p__firmicutes", "p__proteobacteria", "p__actinobacteri
    "<5% abdund"), labels = c("Firmicutes", "Proteobacteria", "Actinobacteria", "<5% abdund"), guide = g
    ggtitle("nifH Phylum Relative Abundance by Site") + ylab("Relative Abundance") + xlab(" ") + scale_x
    "Huntley Irrigated", "Sidney Dryland", "Sidney Irrigated")) + theme(axis.text.x = element_text(angle
    hjust = 1), axis.text.y = element_blank(), panel.background = element_blank())
```



Publish figure as a tiff

```
tiff("nifH_barplot_sid_hunt.tiff", width = 6, height = 4, units = "in", res = 600)
ggplot(data = data_nifH_irr_phylum, aes(x = Site, y = Abundance, fill = phylum)) + geom_bar(aes(fill = p
    stat = "identity", position = "stack", show.legend = TRUE) + scale_fill_manual(name = "Phylum", valu
    "#1f78b4", "#b2df8a", "#33a02c"), breaks = c("p__firmicutes", "p__proteobacteria", "p__actinobacteri
    "<5% abdund"), labels = c("Firmicutes", "Proteobacteria", "Actinobacteria", "<5% abdund"), guide = g
    ggtitle("nifH Phylum Relative Abundance by Site") + ylab("Relative Abundance") + xlab(" ") + scale_x
    "Huntley Irrigated", "Sidney Dryland", "Sidney Irrigated")) + theme(axis.text.x = element_text(angle
    hjust = 1), axis.text.y = element_blank(), panel.background = element_blank())
dev.off()
```

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

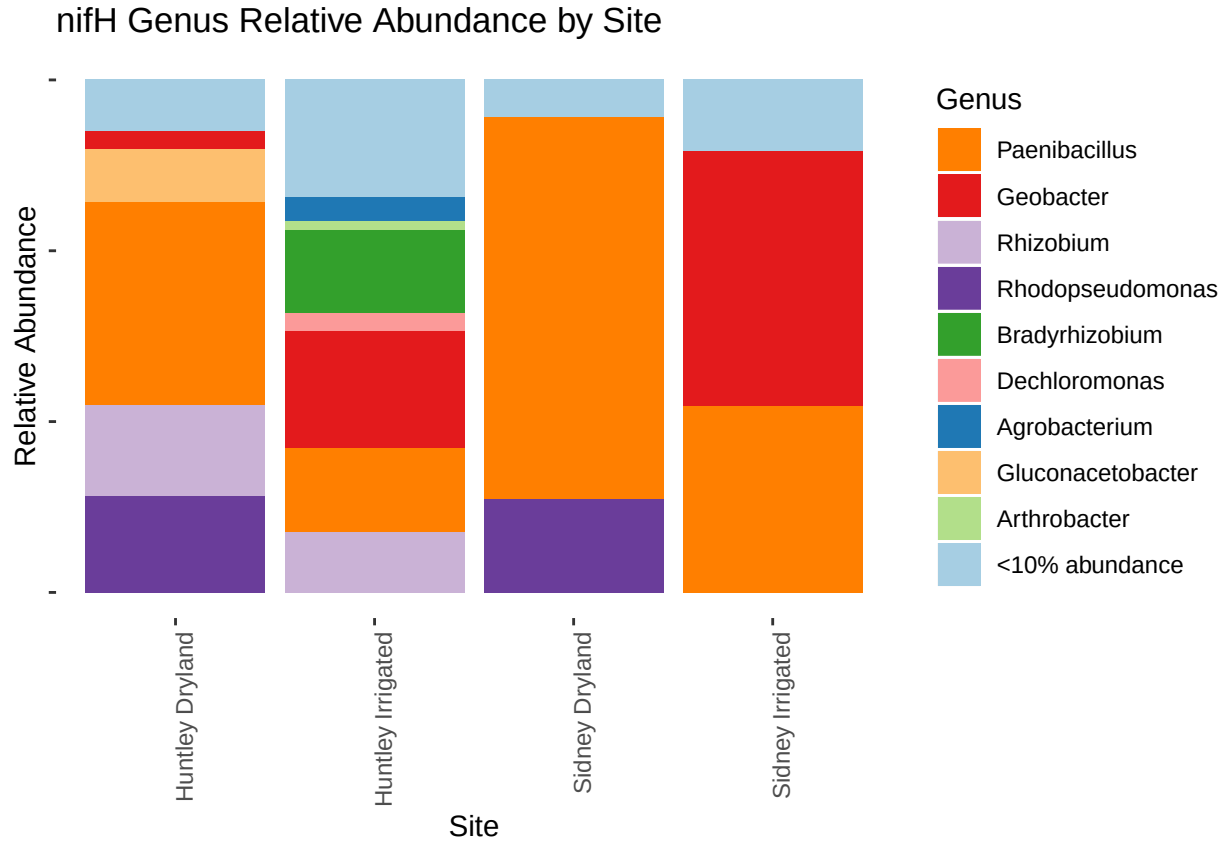
nifH genus barplot

```
physeq_nifH_ord_irr_genus <- tax_glom(physeq_nifH_irr_ord, "genus")
data_nifH__irr_genus <- psmelt(physeq_nifH_ord_irr_genus)
data_nifH__irr_genus$genus <- as.character(data_nifH__irr_genus$genus)
data_nifH__irr_genus$genus[data_nifH__irr_genus$Abundance < 0.1] <- "<10% abdund"
unique(data_nifH__irr_genus$genus)
```

```
##  [1] "g__paenibacillus"     "g__geobacter"         "g__rhizobium"
```

```
##  [4] "g__rhodopseudomonas"  "g__bradyrhizobium"    "g__dechloromonas"
##  [7] "g__agrobacterium"     "g__gluconacetobacter" "g__arthrobacter"
## [10] "<10% abdund"
```

```
ggplot(data = data_nifH__irr_genus, aes(x = Site, y = Abundance, fill = genus)) + geom_bar(aes(fill = g
    stat = "identity", position = "stack", show.legend = TRUE) + scale_fill_manual(name = "Genus", valu
    "#1f78b4", "#b2df8a", "#33a02c", "#fb9a99", "#e31a1c", "#fdbf6f", "#ff7f00", "#cab2d6", "#6a3d9a",
    "#ffff99", "#b15928"), breaks = c("g__paenibacillus", "g__geobacter", "g__rhizobium", "g__rhodopseu
    "g__bradyrhizobium", "g__dechloromonas", "g__agrobacterium", "g__gluconacetobacter", "g__arthrobact
    "<10% abdund"), labels = c("Paenibacillus", "Geobacter", "Rhizobium", "Rhodopseudomonas", "Bradyrhi
    "Dechloromonas", "Agrobacterium", "Gluconacetobacter", "Arthrobacter", "<10% abundance"), guide = gu
    ggtitle("nifH Genus Relative Abundance by Site") + ylab("Relative Abundance") + scale_x_discrete(la
    "Huntley Irrigated", "Sidney Dryland", "Sidney Irrigated")) + theme(axis.text.x = element_text(angl
    hjust = 1), axis.text.y = element_blank(), panel.background = element_blank())
```



Publish to a tiff image

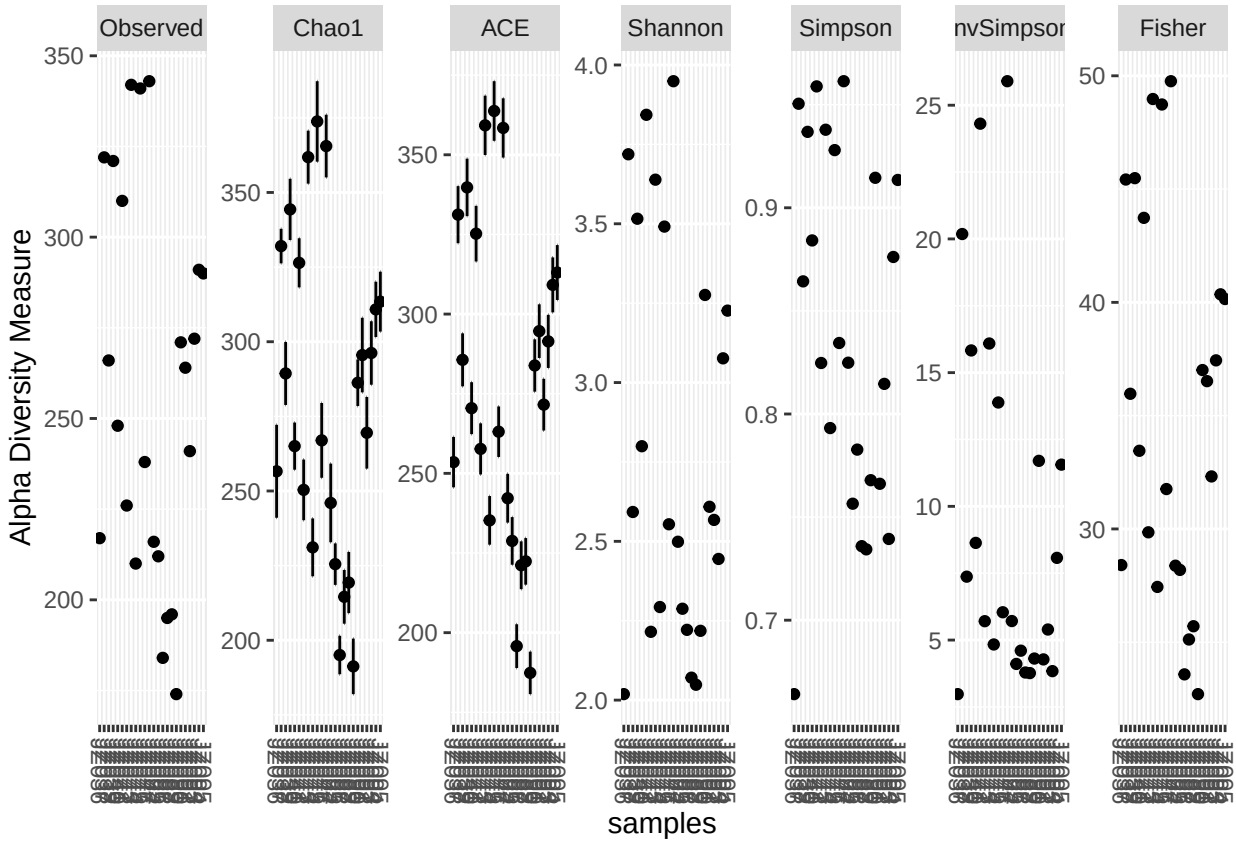
```
## pdf
##   2
```

Alpha diversity metrics

Use phyloseq internal packages to calculate the alpha diversity

```
plot_richness(physeq_nifH_irr_trim)
```

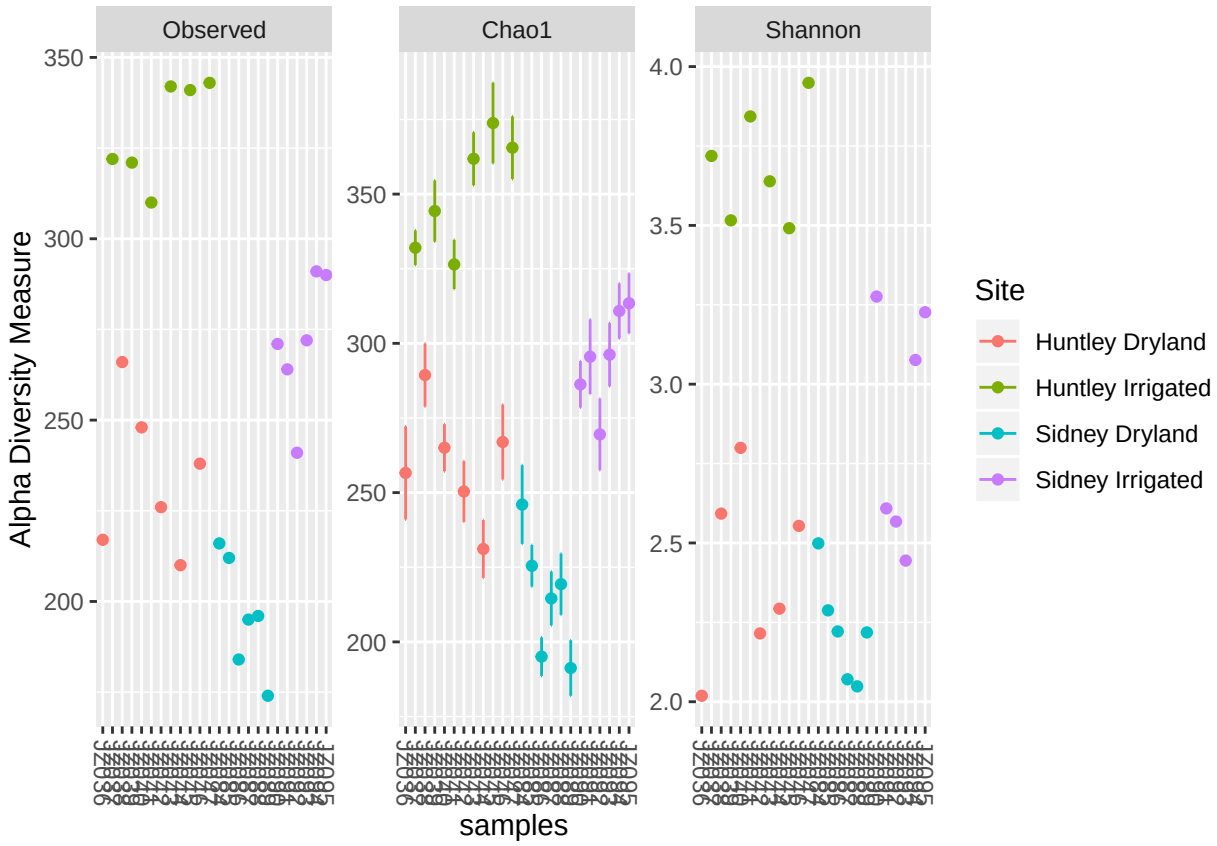
```
## Warning: Removed 120 rows containing missing values (geom_errorbar).
```



Simplify to just observed and Chao1 and Shannon

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

```
## Warning: Removed 48 rows containing missing values (geom_errorbar).
```



Make a table of the alpha and write table to folder

	Observed	Chao1	se.chao1	ACE	se.ACE	Shannon	Simpson	InvSimpson	Fisher
JZ036	217	256.6000	15.568279	253.5259	7.914123	2.018819	0.6641435	2.977463	28.40523
JZ037	322	332.0435	5.784462	331.2057	8.956926	3.718885	0.9504545	20.183460	45.42288
JZ038	266	289.3750	10.555891	285.6303	8.317239	2.592357	0.8643164	7.370089	35.96617
JZ039	321	344.3333	10.245919	339.7647	9.021636	3.515862	0.9368300	15.830297	45.48287
JZ040	248	265.0323	7.896504	270.4646	8.153909	2.799469	0.8841695	8.633305	33.44890
JZ041	310	326.4348	8.227439	325.2285	8.726253	3.843113	0.9588639	24.309542	43.73061

Just make a shannon table for further analysis

```
statewide_nifH_irr_shannon <- estimate_richness(physeq_nifH_irr_trim, split = TRUE, measures = "Shannon"

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

head(statewide_nifH_irr_shannon)
```

	Shannon
JZ036	2.018819
JZ037	3.718885
JZ038	2.592357
JZ039	3.515862

	Shannon
JZ040	2.799469
JZ041	3.843113

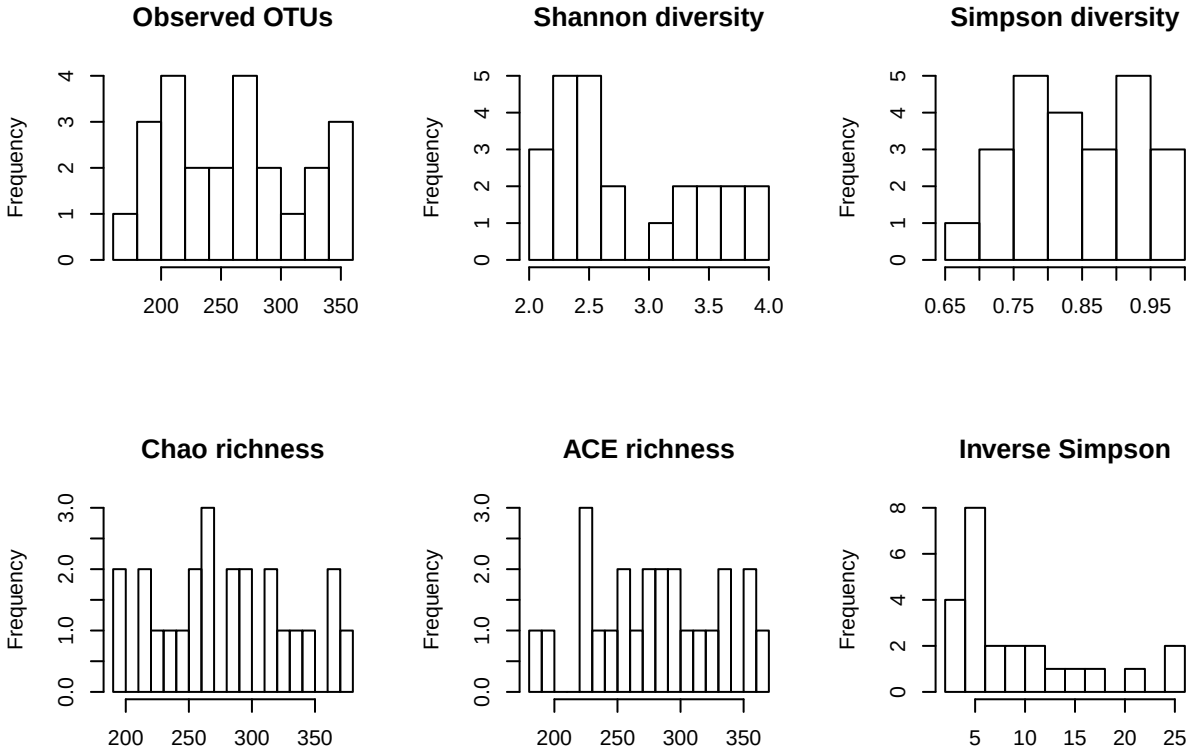
Plot Shannon diversity boxplot using ggpubr

We see there is a signficant diffrence between sites using the Kruskal-Wallis test. From the above graph we see the plots look like normal distrubution 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 2x3 plot environment so that we can see all 6 metrics at once.
par(mfrow = c(2, 3))

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

```r
shapiro.test(rich_irr_nifH$Observed)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rich_irr_nifH$Observed
## W = 0.94578, p-value = 0.2191
```

```r
shapiro.test(rich_irr_nifH$Shannon)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rich_irr_nifH$Shannon
## W = 0.90031, p-value = 0.02185
```

```r
shapiro.test(rich_irr_nifH$InvSimpson)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rich_irr_nifH$InvSimpson
## W = 0.81257, p-value = 0.0004762
```

```r
shapiro.test(rich_irr_nifH$Chao1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rich_irr_nifH$Chao1
## W = 0.96998, p-value = 0.6665
```

```r
shapiro.test(rich_irr_nifH$ACE)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rich_irr_nifH$ACE
## W = 0.97093, p-value = 0.6901
```

```r
shapiro.test(rich_irr_nifH$InvSimpson)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rich_irr_nifH$InvSimpson
## W = 0.81257, p-value = 0.0004762
```

All noraml from shapiro test we can use anova for variance diffrence

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

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

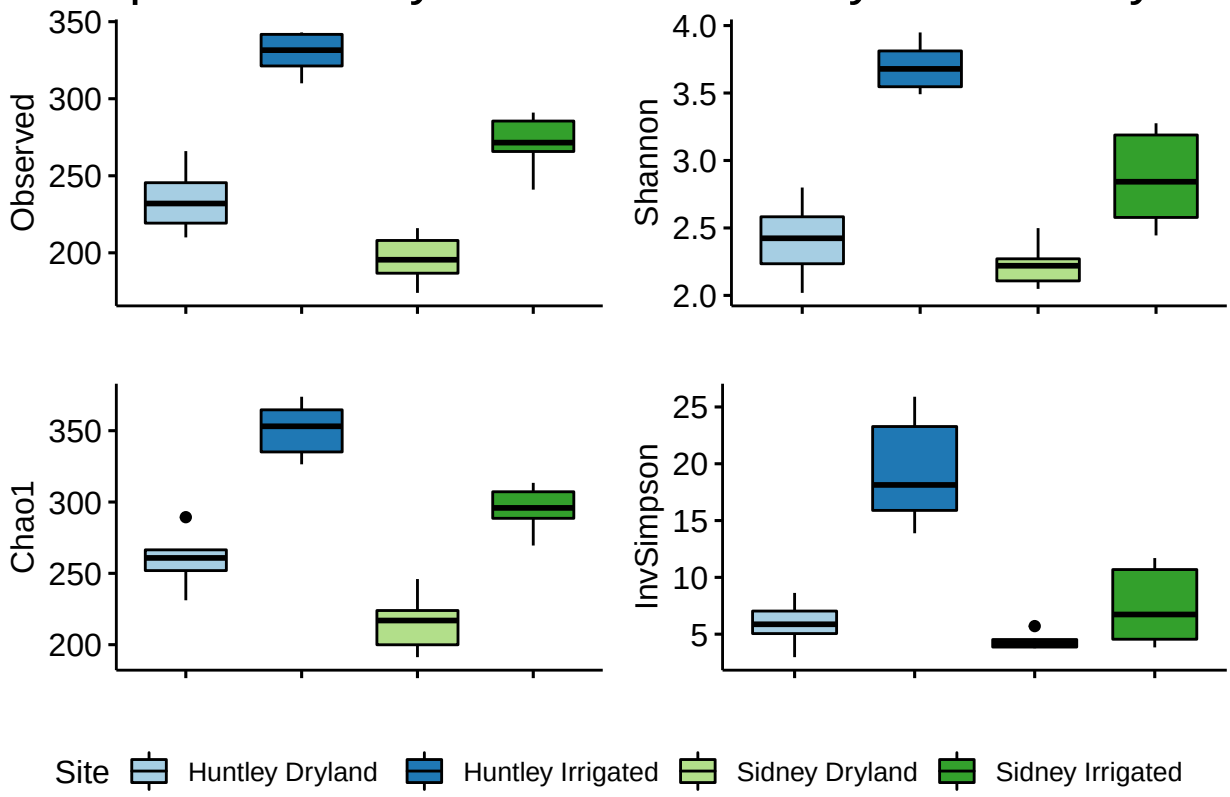
	Site	ARC	Season	Sample_dates	Pea_variety	Plot	season_precip	irrgation	to
JZ036	Huntley Dryland	SARC	Summer	8/5/2016	AC Earlystar	Dryland	8.79	0.0	
JZ037	Huntley Irrigated	SARC	Summer	8/5/2016	AC Earlystar	Irrigated	8.79	2.5	
JZ038	Huntley Dryland	SARC	Summer	8/5/2016	CDC Saffron	Dryland	8.79	0.0	
JZ039	Huntley Irrigated	SARC	Summer	8/5/2016	CDC Saffron	Irrigated	8.79	2.5	
JZ040	Huntley Dryland	SARC	Summer	8/5/2016	Delta	Dryland	8.79	0.0	
JZ041	Huntley Irrigated	SARC	Summer	8/5/2016	Delta	Irrigated	8.79	2.5	

```
mean(meta_irr_nifH$Observed)
```

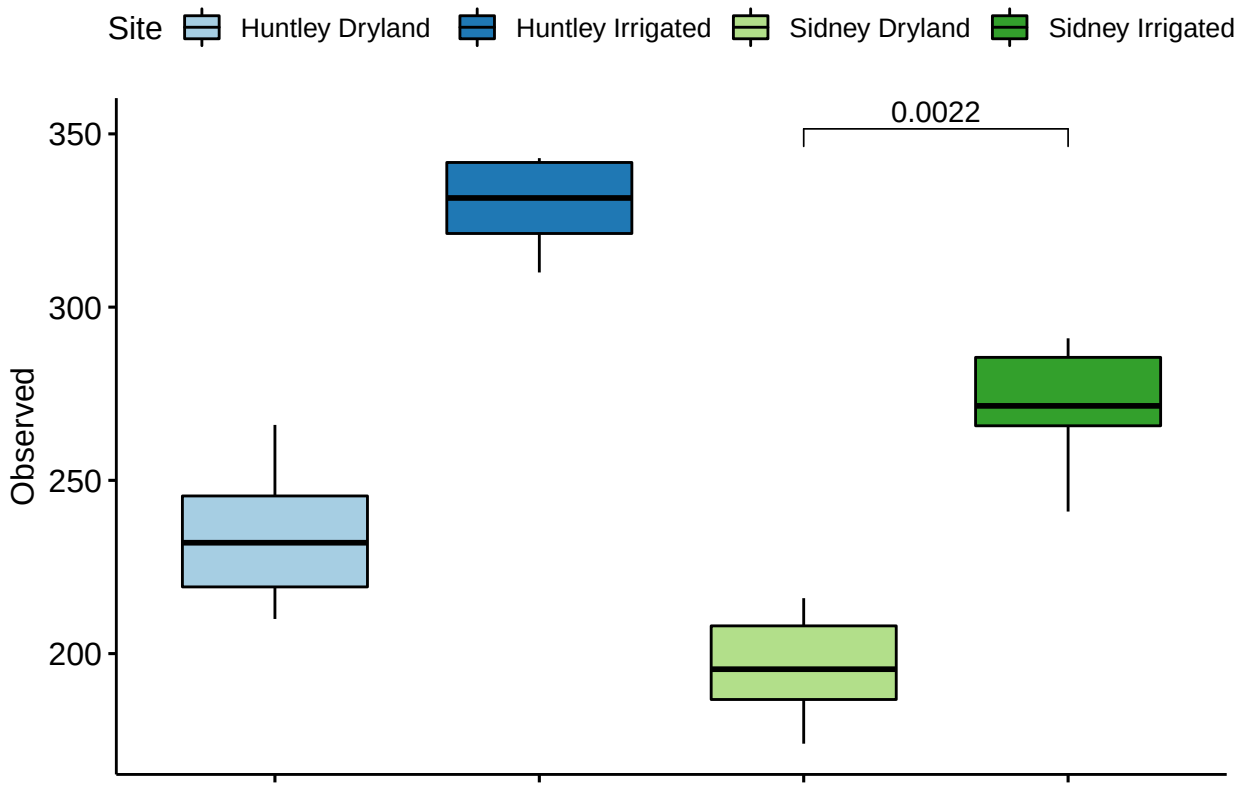
```
## [1] 257.9167
```

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

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

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

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

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

alpha_nifH_fig <- ggarrange(nifH_irr_Observ, nifH_irr_Shannon, nifH_irr_Chao, nifH_irr_InvSim, ncol = 2
    nrow = 2, common.legend = TRUE, legend = "bottom")

annotate_figure(alpha_nifH_fig, top = text_grob("Alpha Diversity of nifH for Huntley and Sidney", size =
```

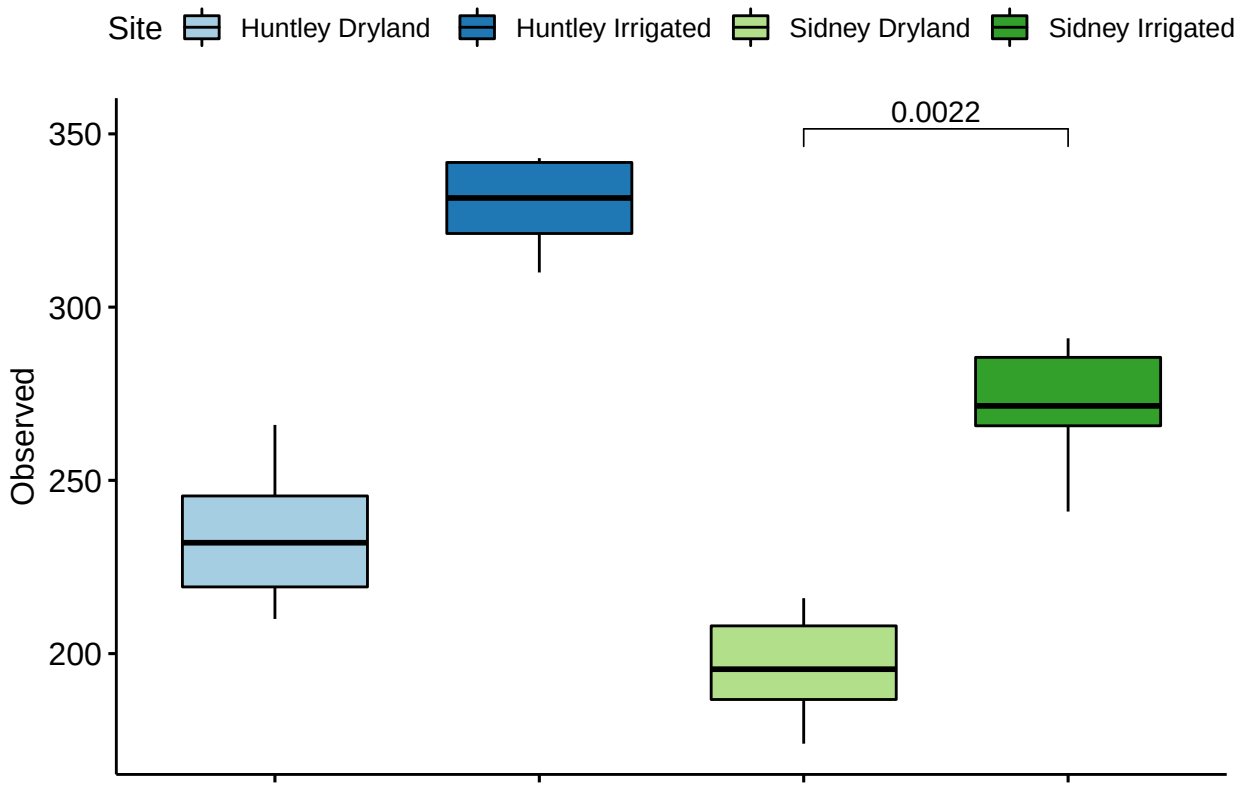



```
nifH_irr_Observ <- ggboxplot(meta_irr_nifH, x = "Site", y = "Observed", rug = TRUE, fill = "Site", xlab
    palette = farm_col_paired) + rremove("x.text") + stat_compare_means(comparisons = list(c("Sidney Dry
    "Sidney Irrigated")))
nifH_irr_Observ
```



meta_irr

```
nifH_irr_precip <- ggboxplot(meta_irr_nifH, x = "Site", y = "season_precip", rug = TRUE, fill = "Site",
    xlab = " ", palette = farm_col_paired) + rremove("x.text") + stat_compare_means(comparisons = list(c
    "Sidney Irrigated")))
nifH_irr_Observ
```




```
nifH_irr_Observ_stats <- ggplot_build(nifH_irr_Observ)
nifH_irr_Observ_stats$data
```

```
## [[1]]
##      fill ymin  lower middle  upper ymax outliers notchupper notchlower x
## 1 #A6CEE3  210 219.25  232.0 245.50  266            248.9321   215.0679 1
## 2 #1F78B4  310 321.25  331.5 341.75  343            344.7232   318.2768 2
## 3 #B2DF8A  174 186.75  195.5 208.00  216            209.2069   181.7931 3
## 4 #33A02C  241 265.75  271.5 285.50  291            284.2394   258.7606 4
##   PANEL group ymin_final ymax_final xmin xmax weight colour size alpha
## 1     1     1         210         266 0.65 1.35      1  black  0.5    NA
## 2     1     2         310         343 1.65 2.35      1  black  0.5    NA
## 3     1     3         174         216 2.65 3.35      1  black  0.5    NA
## 4     1     4         241         291 3.65 4.35      1  black  0.5    NA
##   shape linetype
## 1    19    solid
## 2    19    solid
## 3    19    solid
## 4    19    solid
##
## [[2]]
##   x xend      y   yend annotation                              group PANEL
## 1 3    3 346.38 351.45     0.0022 Sidney Dryland-Sidney Irrigated-1     1
## 2 3    4 351.45 351.45     0.0022 Sidney Dryland-Sidney Irrigated-1     1
## 3 4    4 351.45 346.38     0.0022 Sidney Dryland-Sidney Irrigated-1     1
##   shape colour textsize angle hjust vjust alpha family fontface lineheight
## 1    19  black     3.88     0   0.5     0    NA                1        1.2
```

```
## 2    19  black     3.88     0   0.5     0    NA               1        1.2
## 3    19  black     3.88     0   0.5     0    NA               1        1.2
##   linetype size
## 1        1  0.3
## 2        1  0.3
## 3        1  0.3
```

```
nifH_irr_shannon_stats <- ggplot_build(nifH_irr_Shannon)
nifH_irr_shannon_stats$data
```

```
## [[1]]
##      fill     ymin    lower   middle    upper     ymax outliers notchupper
## 1 #A6CEE3 2.018819 2.234629 2.423432 2.582699 2.799469            2.647948
## 2 #1F78B4 3.491114 3.546611 3.678872 3.812056 3.948972            3.850092
## 3 #B2DF8A 2.048428 2.107541 2.219769 2.271130 2.498719            2.325289
## 4 #33A02C 2.444517 2.577766 2.842540 3.188935 3.275835            3.236764
##   notchlower x PANEL group ymin_final ymax_final xmin xmax weight colour
## 1   2.198916 1     1     1   2.018819   2.799469 0.65 1.35      1  black
## 2   3.507651 2     1     2   3.491114   3.948972 1.65 2.35      1  black
## 3   2.114248 3     1     3   2.048428   2.498719 2.65 3.35      1  black
## 4   2.448316 4     1     4   2.444517   3.275835 3.65 4.35      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
```

Save to .tiff

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

Explination 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 dominace probabilty of any two indviduals drawn at random beloging to the same species

Use ANOVA on alpha diversity metrics for main variables

Shannon first

```
aov_shannon_site_nifH <- aov(Shannon ~ Site, meta_irr_nifH)
summary(aov_shannon_site_nifH)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Site         3  7.701  2.5671   37.12 2.3e-08 ***
## Residuals   20  1.383  0.0692
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Correct for multiple comparisons

```
shannon_nifH_site <- TukeyHSD(aov_shannon_site_nifH, "Site", ordered = TRUE)
shannon_nifH_site
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##     factor levels have been ordered
##
## Fit: aov(formula = Shannon ~ Site, data = meta_irr_nifH)
##
## $Site
##                                       diff         lwr       upr
## Huntley Dryland -Sidney Dryland    0.1879280 -0.23701502 0.6128710
## Sidney Irrigated-Sidney Dryland    0.6423787  0.21743564 1.0673217
## Huntley Irrigated-Sidney Dryland   1.4686230  1.04368003 1.8935661
## Sidney Irrigated-Huntley Dryland   0.4544507  0.02950764 0.8793937
## Huntley Irrigated-Huntley Dryland  1.2806950  0.85575203 1.7056381
## Huntley Irrigated-Sidney Irrigated 0.8262444  0.40130138 1.2511874
##                                       p adj
## Huntley Dryland -Sidney Dryland    0.6110632
## Sidney Irrigated-Sidney Dryland    0.0021361
## Huntley Irrigated-Sidney Dryland   0.0000000
## Sidney Irrigated-Huntley Dryland   0.0334007
## Huntley Irrigated-Huntley Dryland  0.0000003
## Huntley Irrigated-Sidney Irrigated 0.0001371
```

irrigation

```
aov_shannon_irr_nifH <- aov(Shannon ~ Plot, meta_irr_nifH)
summary(aov_shannon_irr_nifH)
```

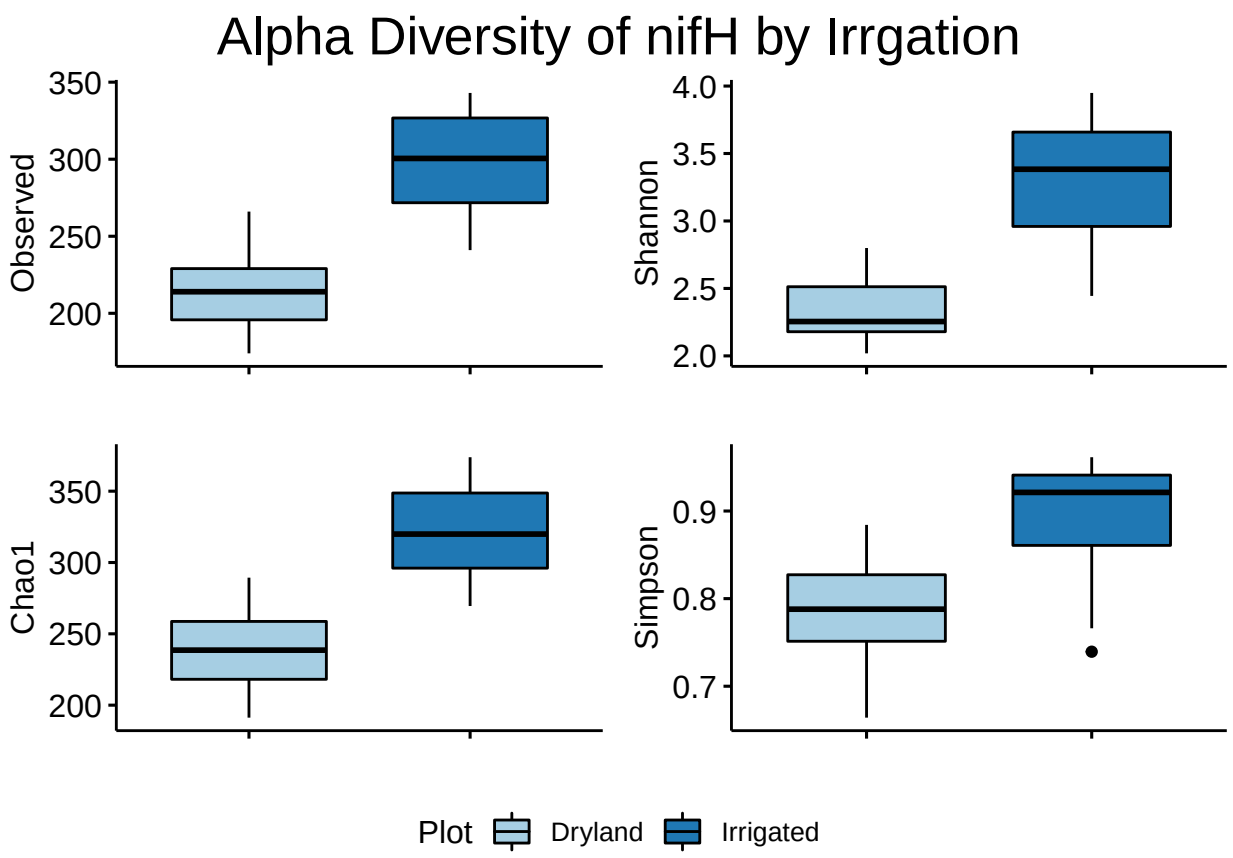
```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Plot         1  5.547   5.547    34.5 6.54e-06 ***
## Residuals   22  3.537   0.161
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

irrgation is not sigficant driver of alpha diversity

Plot Irrigation

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

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

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

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

alpha_nifH_irr_fig <- ggarrange(nifH_Observ_irr, nifH_Shannon_irr, nifH_Chao_irr, nifH_InvSim_irr, ncol
    nrow = 2, common.legend = TRUE, legend = "bottom")

annotate_figure(alpha_nifH_irr_fig, top = text_grob("Alpha Diversity of nifH by Irrgation", size = 20))
```




```
tiff("aDiv_irr_plot_nifH.tiff", width = 8, height = 4.5, units = "in", res = 600)


alpha_nifH_irr_fig <- ggarrange(nifH_Observ_irr, nifH_Shannon_irr, nifH_Chao_irr, nifH_InvSim_irr, ncol
    nrow = 2, common.legend = TRUE, legend = "bottom")

annotate_figure(alpha_nifH_irr_fig, top = text_grob("Alpha Diversity of nifH by Irrigation", size = 20))

dev.off()
```

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

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

```r
aov_observed_nifH_irr <- aov(Observed ~ Plot, meta_irr_nifH)
summary(aov_observed_nifH_irr)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Plot         1  43862   43862   46.66 7.31e-07 ***
## Residuals   22  20678     940
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
aov_shannon_nifH_irr <- aov(Shannon ~ Plot, meta_irr_nifH)
summary(aov_shannon_nifH_irr)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Plot         1  5.547   5.547    34.5 6.54e-06 ***
## Residuals   22  3.537   0.161
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

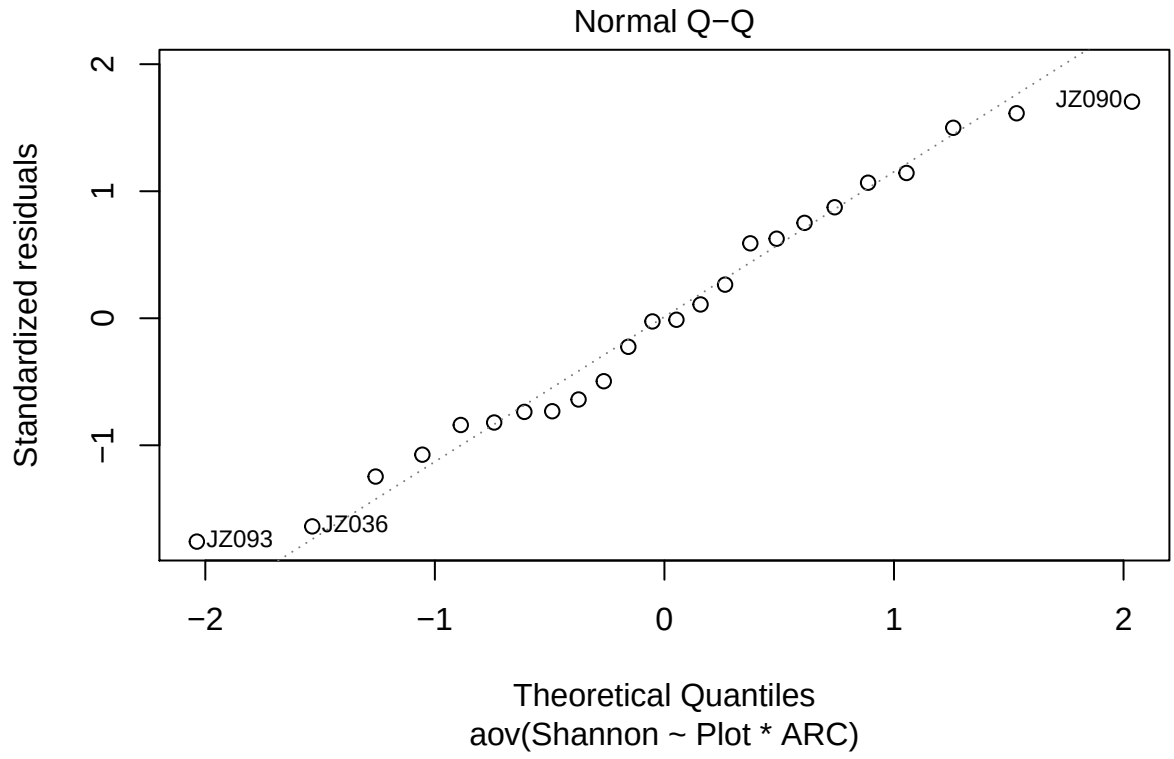
```r
aov_shannon_irr_nifH_site <- aov(Shannon ~ Plot + ARC, meta_irr_nifH)
summary(aov_shannon_irr_nifH_site)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Plot         1  5.547   5.547   58.42  1.7e-07 ***
## ARC          1  1.543   1.543   16.25 0.000604 ***
## Residuals   21  1.994   0.095
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

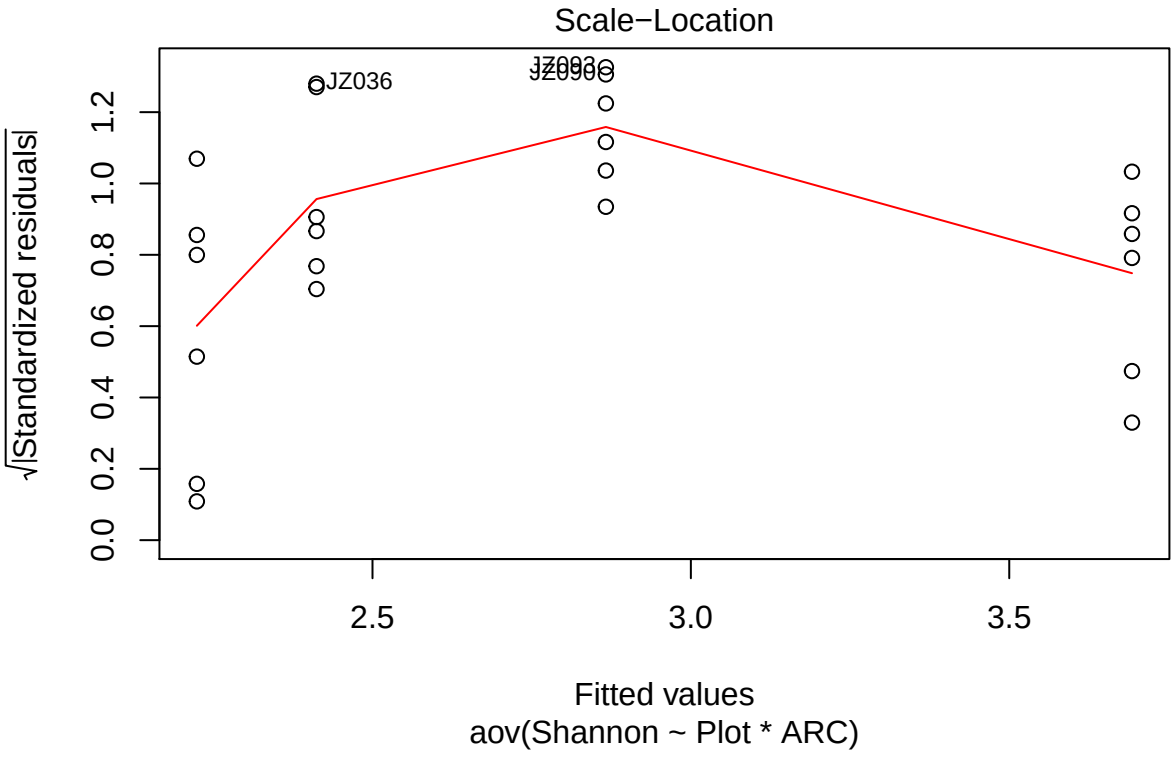
```r
aov_shannon_irr_nifH_site <- aov(Shannon ~ Plot * ARC, meta_irr_nifH)
summary(aov_shannon_irr_nifH_site)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Plot         1  5.547   5.547  80.221 1.95e-08 ***
## ARC          1  1.543   1.543  22.311  0.00013 ***
## Plot:ARC     1  0.611   0.611   8.838  0.00752 **
## Residuals   20  1.383   0.069
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

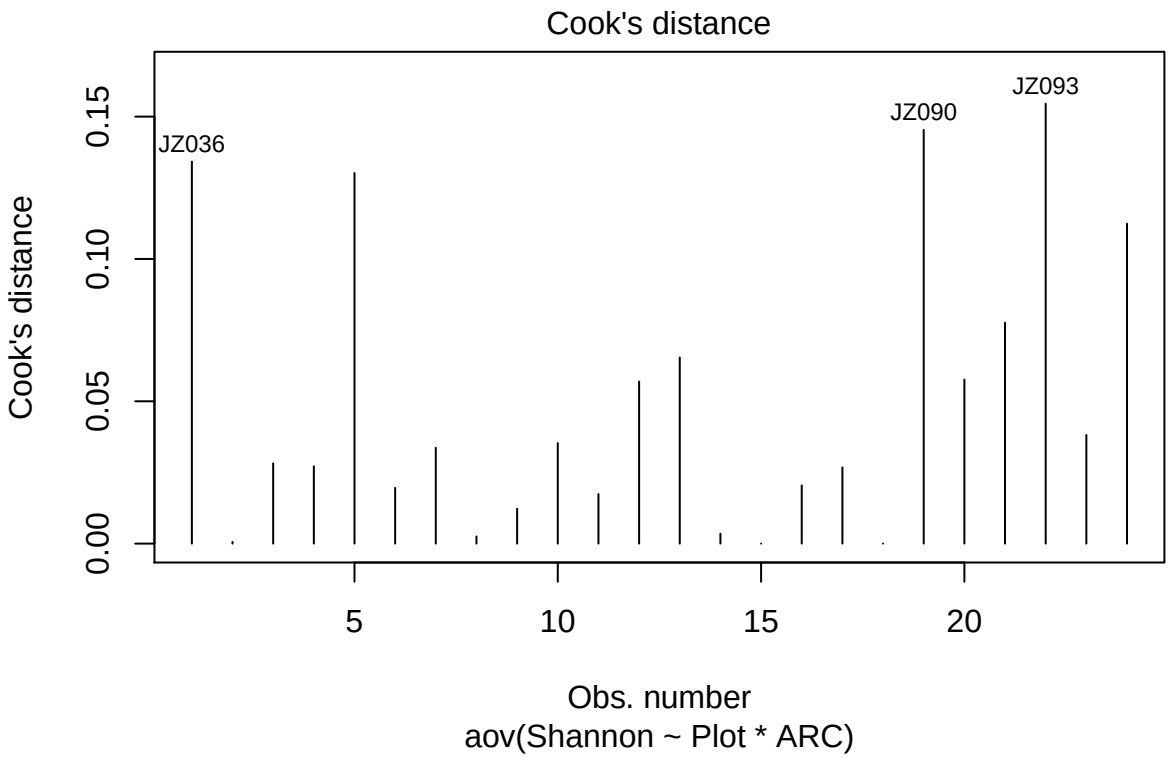
```r
plot(aov_shannon_irr_nifH_site, 2)
```




```
plot(aov_shannon_irr_nifH_site, 3)
```




```
plot(aov_shannon_irr_nifH_site, 4)
```




```
aov_shannon_irr_nifH_site <- aov(Shannon ~ Plot/ARC, meta_irr_nifH)
summary(aov_shannon_irr_nifH_site)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Plot         1  5.547   5.547   80.22 1.95e-08 ***
## Plot:ARC     2  2.154   1.077   15.57 8.35e-05 ***
## Residuals   20  1.383   0.069
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
aov_Chao1_nifH_irr <- aov(Chao1 ~ Plot, meta_irr_nifH)
summary(aov_Chao1_nifH_irr)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Plot         1  43745   43745   43.21 1.3e-06 ***
## Residuals   22  22271    1012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
aov_Simpson_nifH_irr <- aov(Simpson ~ Plot, meta_irr_nifH)
summary(aov_Simpson_nifH_irr)
```

```
##             Df  Sum Sq Mean Sq F value  Pr(>F)
## Plot         1 0.06309 0.06309      13 0.00157 **
## Residuals   22 0.10675 0.00485
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
aov_Simpson_nifH_irr_site <- aov(Simpson ~ Plot/Site, meta_irr_nifH)
summary(aov_Simpson_nifH_irr_site)
```

```
##             Df  Sum Sq Mean Sq F value   Pr(>F)
## Plot         1 0.06309 0.06309   19.15 0.000292 ***
## Plot:Site    2 0.04085 0.02043    6.20 0.008033 **
## Residuals   20 0.06590 0.00329
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

https://joey711.github.io/phyloseq/plot_ordination-examples.html

```
wh0 = genefilter_sample(physeq_nifH_irr_trim, filterfun_sample(function(x) x > 5), A = 0.1 * nsamples(pl
physeq_nifH_irr_ord = prune_taxa(wh0, physeq_nifH_irr_trim)
physeq_nifH_irr_ord
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 466 taxa and 24 samples ]
## sample_data() Sample Data:       [ 24 samples by 45 sample variables ]
## tax_table()   Taxonomy Table:    [ 466 taxa by 8 taxonomic ranks ]
```

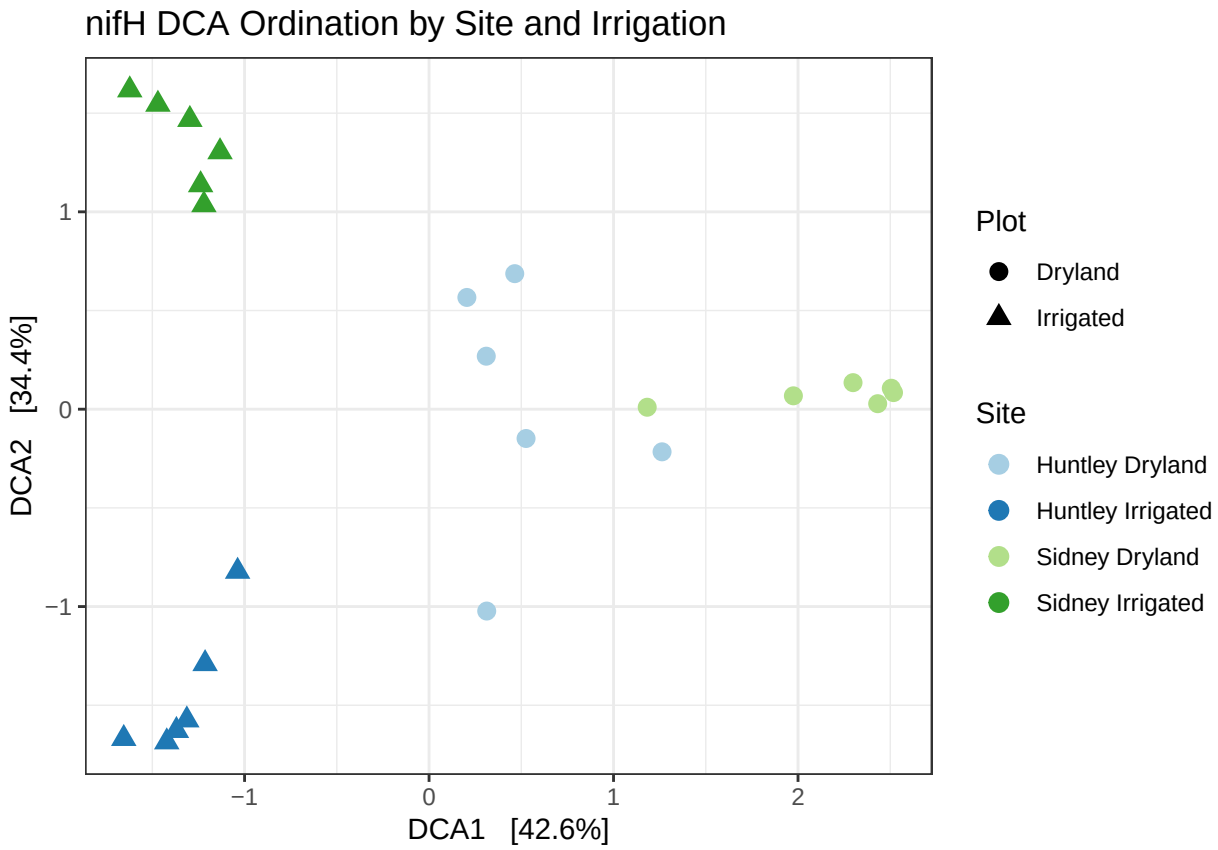
Transform to even sampling depth

```
physeq_nifH_irr_ord = transform_sample_counts(physeq_nifH_irr_ord, function(x) 1e+06 * x/sum(x))
physeq_nifH_irr_ord
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 466 taxa and 24 samples ]
## sample_data() Sample Data:       [ 24 samples by 45 sample variables ]
## tax_table()   Taxonomy Table:    [ 466 taxa by 8 taxonomic ranks ]
```

DCA Ordination

```
physeq_nifH_irr_DCA <- ordinate(physeq_nifH_irr_ord, "DCA", "bray")
plot_ordination(physeq_nifH_irr_ord, physeq_nifH_irr_DCA, color = "Site", shape = "Plot") + geom_point(s
    scale_color_manual(values = farm_col_paired) + ggtitle("nifH DCA Ordination by Site and Irrigation")
    theme_bw()
```

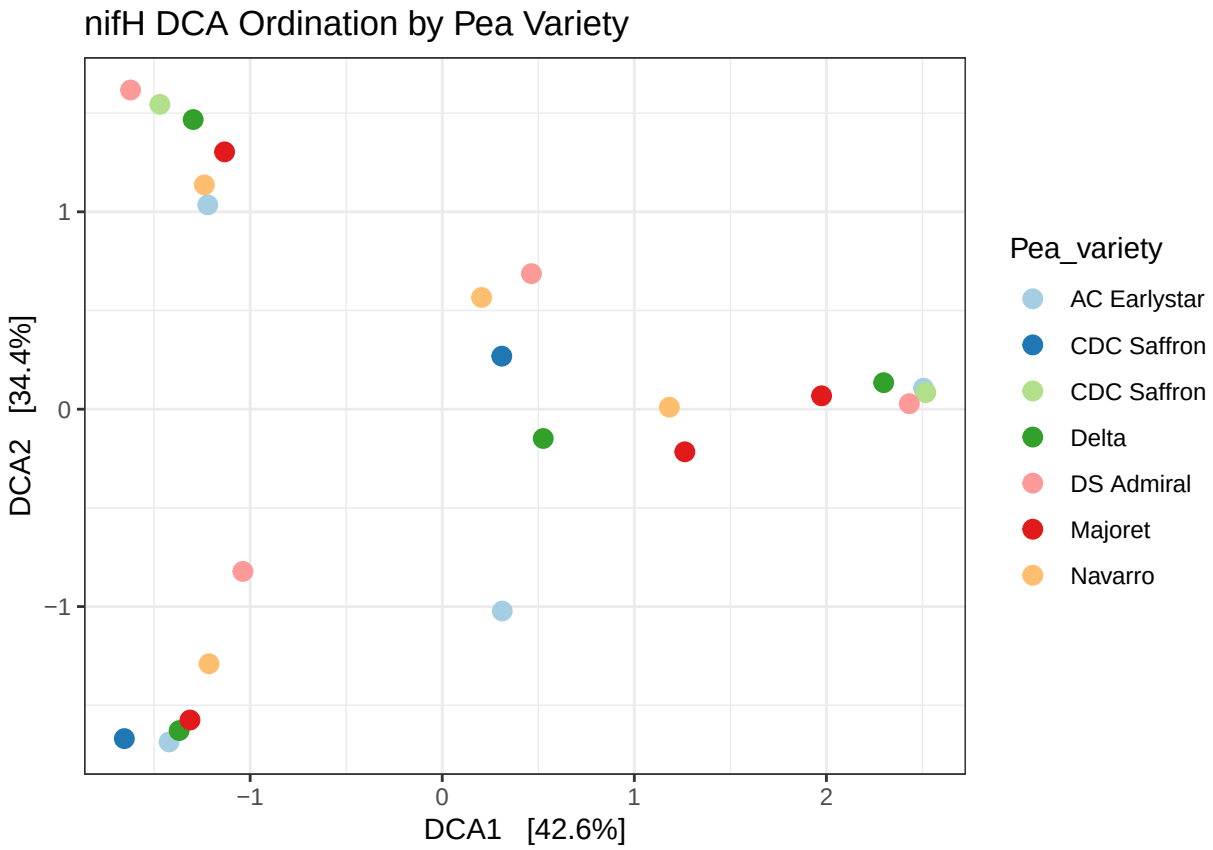



```
tiff("nifHDCA_IRR.tiff", width = 6, height = 4, units = "in", res = 600)
plot_ordination(physeq_nifH_irr_ord, physeq_nifH_irr_DCA, color = "Site", shape = "Plot") + geom_point(s
    scale_color_manual(values = farm_col_paired) + ggtitle("nifH DCA Ordination by Site and Irrigation")
    theme_bw()

dev.off()
```

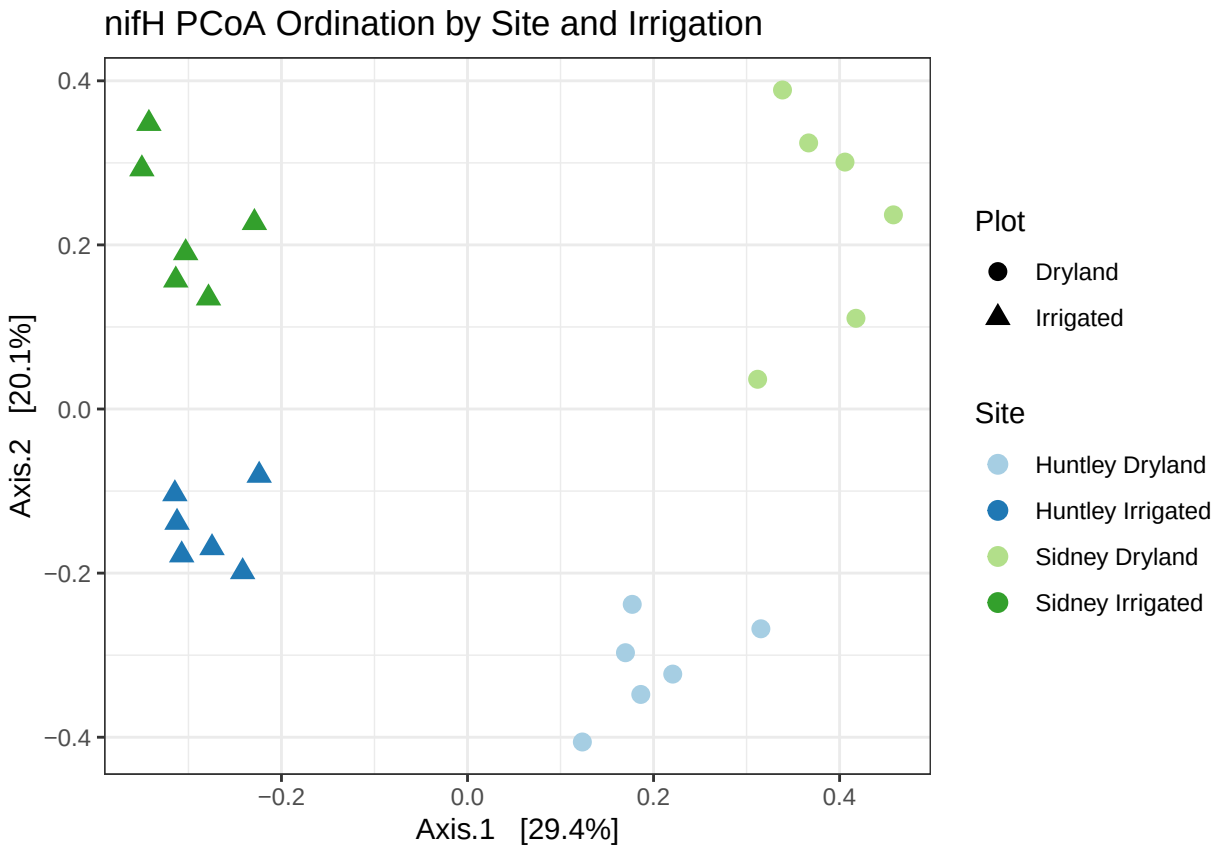
```
## pdf
##   2
```

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

```
plot_ordination(physeq_nifH_irr_ord, physeq_nifH_irr_DCA, color = "Pea_variety") + geom_point(size = 3)
    scale_color_manual(values = farm_col_paired) + ggtitle("nifH DCA Ordination by Pea Variety") + theme
```



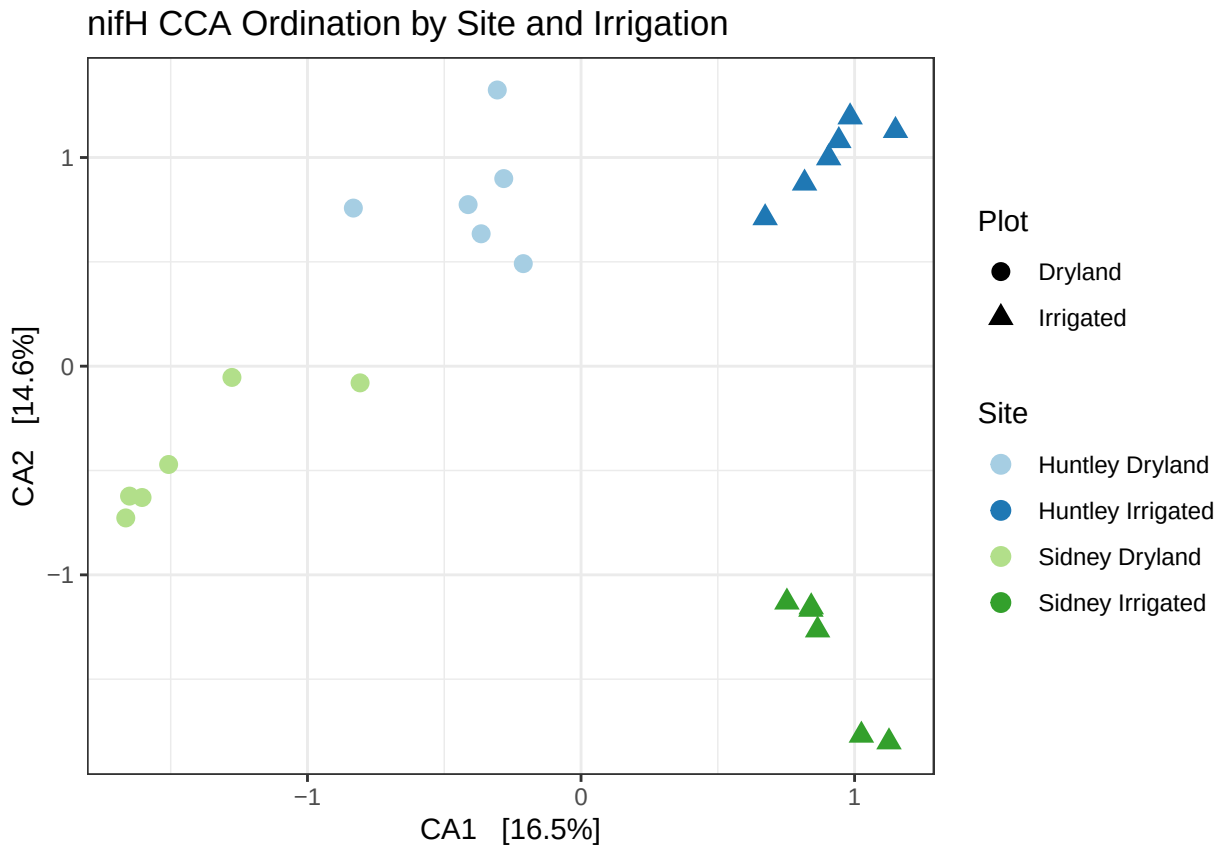

```
phynifH_ord_irr_PCoA <- ordinate(physeq_nifH_irr_ord, "PCoA", "bray")
plot_ordination(physeq_nifH_irr_ord, phynifH_ord_irr_PCoA, color = "Site", shape = "Plot") + geom_point
    scale_color_manual(values = farm_col_paired) + ggtitle("nifH PCoA Ordination by Site and Irrigation
    theme_bw()
```



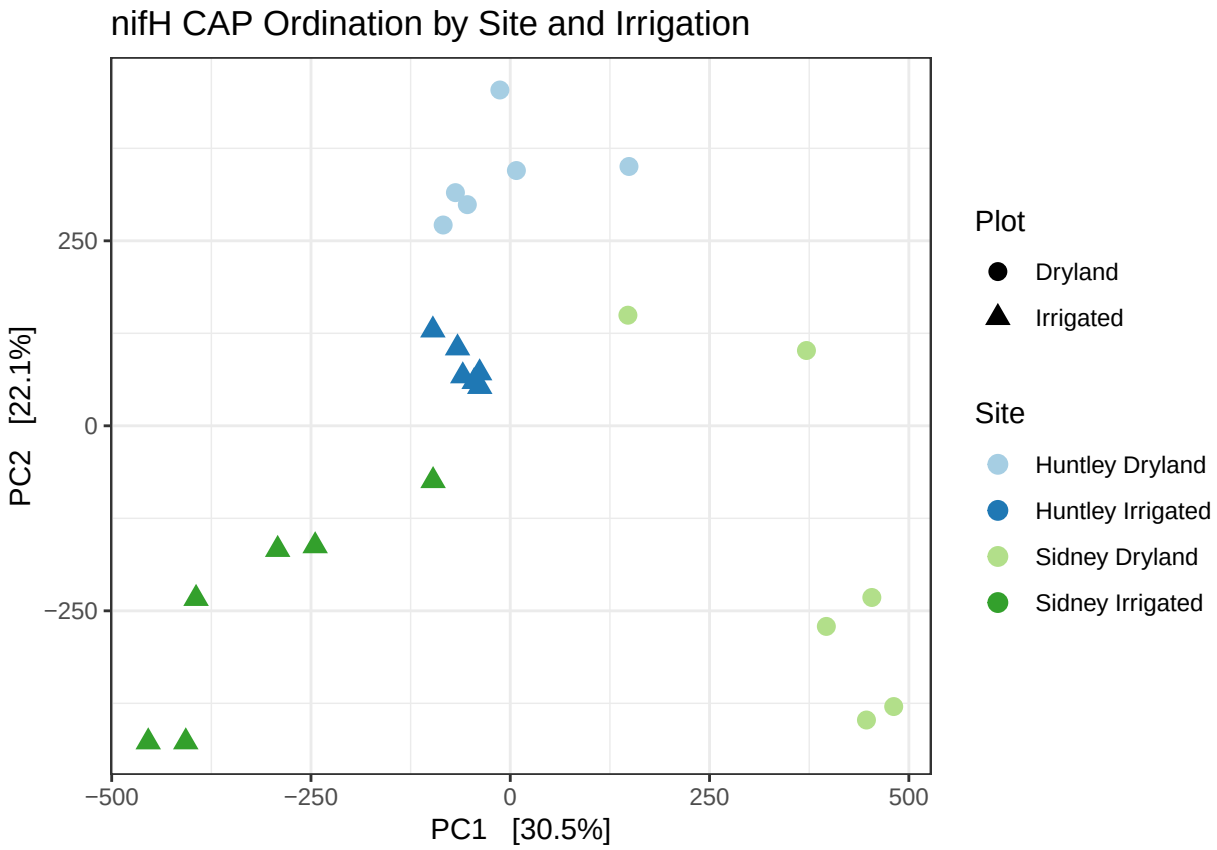

```
tiff("nifHPCoA_IRR.tiff", width = 6, height = 4, units = "in", res = 600)
plot_ordination(physeq_nifH_irr_ord, phynifH_ord_irr_PCoA, color = "Site", shape = "Plot") + geom_point
    scale_color_manual(values = farm_col_paired) + ggtitle("nifH PCoA Ordination by Site and Irrigation
    theme_bw()

dev.off()
```

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

```
phynifH_ord_CCA <- ordinate(physeq_nifH_irr_ord, "CCA", "bray")
plot_ordination(physeq_nifH_irr_ord, phynifH_ord_CCA, color = "Site", shape = "Plot") + geom_point(size
    scale_color_manual(values = farm_col_paired) + ggtitle("nifH CCA Ordination by Site and Irrigation"
    theme_bw()
```




```
phynifH_ord_CAP <- ordinate(physeq_nifH_irr_ord, "RDA", "bray")
plot_ordination(physeq_nifH_irr_ord, phynifH_ord_CAP, color = "Site", shape = "Plot") + geom_point(size
    scale_color_manual(values = farm_col_paired) + ggtitle("nifH CAP Ordination by Site and Irrigation"
    theme_bw()
```



####CDA 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

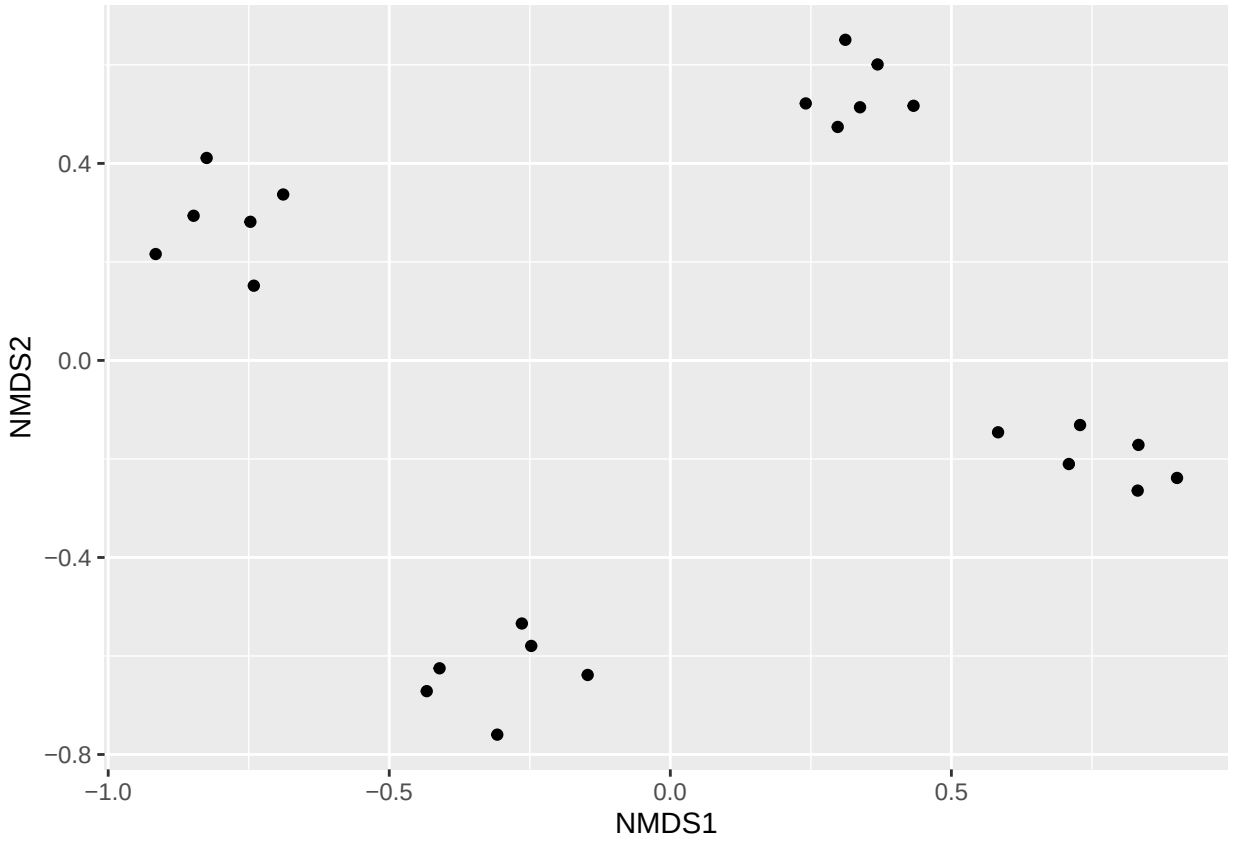
NMDS Ordination

```
phynifH_ord_NMDS <- ordinate(physeq_nifH_irr_ord, "NMDS", "bray")
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.0703047
## Run 1 stress 0.0703047
## ... Procrustes: rmse 7.220217e-06  max resid 2.084526e-05
## ... Similar to previous best
## Run 2 stress 0.0703047
## ... New best solution
## ... Procrustes: rmse 3.232562e-06  max resid 6.298465e-06
## ... Similar to previous best
## Run 3 stress 0.2014757
## Run 4 stress 0.0703047
## ... Procrustes: rmse 2.663184e-06  max resid 7.381894e-06
## ... Similar to previous best
## Run 5 stress 0.0703047
## ... Procrustes: rmse 7.305813e-07  max resid 1.442144e-06
## ... Similar to previous best
## Run 6 stress 0.1351306
## Run 7 stress 0.0703047
## ... Procrustes: rmse 6.346739e-06  max resid 2.352965e-05
## ... Similar to previous best
## Run 8 stress 0.0703047
## ... Procrustes: rmse 3.651635e-06  max resid 1.461817e-05
## ... Similar to previous best
## Run 9 stress 0.0703047
## ... Procrustes: rmse 5.86677e-06  max resid 2.301643e-05
## ... Similar to previous best
## Run 10 stress 0.0703047
## ... Procrustes: rmse 1.243745e-06  max resid 3.946907e-06
## ... Similar to previous best
## Run 11 stress 0.0703047
## ... New best solution
## ... Procrustes: rmse 1.55868e-06  max resid 6.071146e-06
## ... Similar to previous best
## Run 12 stress 0.0703047
## ... Procrustes: rmse 4.345144e-06  max resid 1.704601e-05
## ... Similar to previous best
## Run 13 stress 0.0703047
## ... Procrustes: rmse 4.936888e-06  max resid 1.568148e-05
## ... Similar to previous best
## Run 14 stress 0.0703047
## ... Procrustes: rmse 1.860098e-06  max resid 4.279083e-06
## ... Similar to previous best
## Run 15 stress 0.0703047
## ... Procrustes: rmse 4.867059e-06  max resid 1.930747e-05
## ... Similar to previous best
## Run 16 stress 0.0703047
## ... Procrustes: rmse 4.787794e-06  max resid 1.886497e-05
## ... Similar to previous best
```

```
## Run 17 stress 0.0703047
## ... Procrustes: rmse 2.818355e-06  max resid 1.085974e-05
## ... Similar to previous best
## Run 18 stress 0.0703047
## ... Procrustes: rmse 3.30776e-06  max resid 1.275659e-05
## ... Similar to previous best
## Run 19 stress 0.0703047
## ... Procrustes: rmse 1.827264e-06  max resid 7.084549e-06
## ... Similar to previous best
## Run 20 stress 0.0703047
## ... Procrustes: rmse 1.133056e-06  max resid 3.368451e-06
## ... Similar to previous best
## *** Solution reached
```

```
plot_ordination(physeq_nifH_irr_ord, phynifH_ord_NMDS)
```



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

```
phynifH_ord_NMDS
```

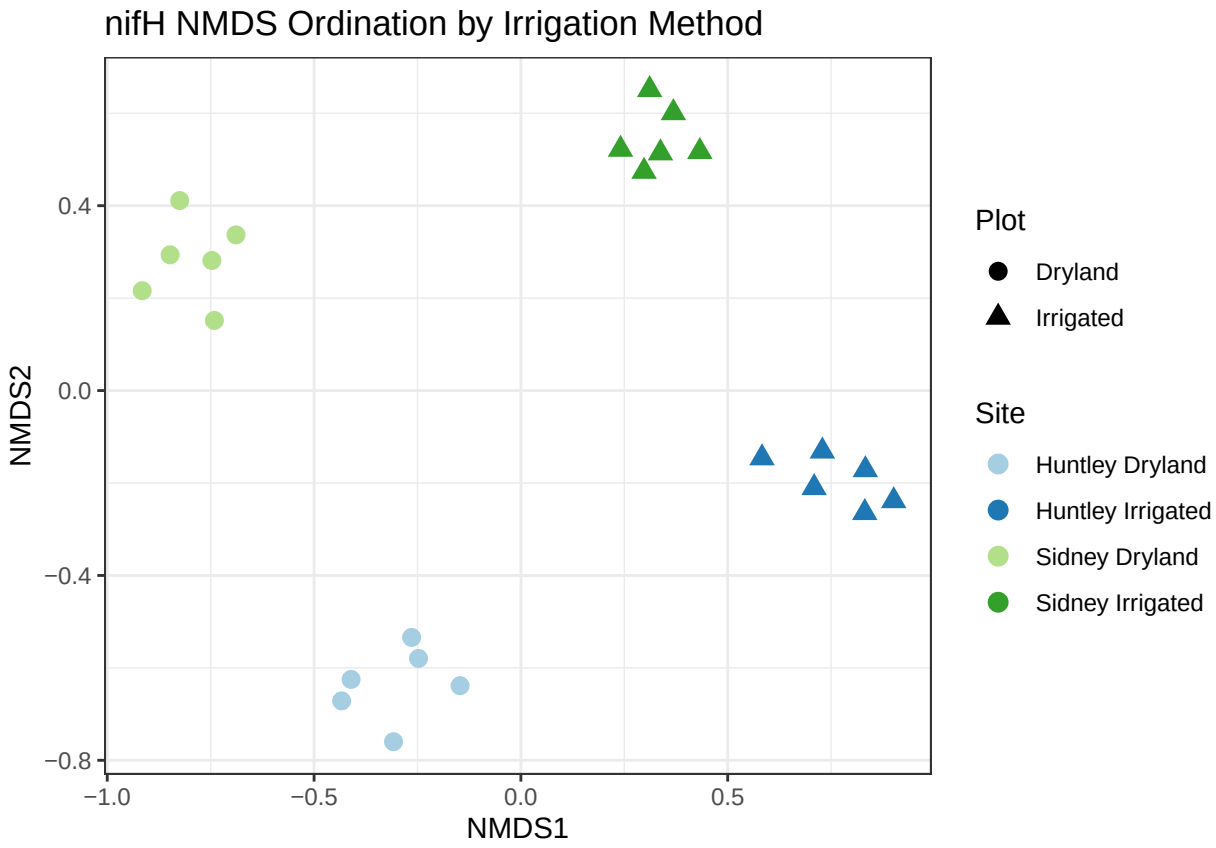
```
##
## Call:
## metaMDS(comm = veganifyOTU(physeq), distance = distance)
##
```

```
## global Multidimensional Scaling using monoMDS
##
## Data:     wisconsin(sqrt(veganifyOTU(physeq)))
## Distance: bray
##
## Dimensions: 2
## Stress:     0.0703047
## 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)))'
```

Stress is really low NMDS ordinate too much.

Plot NMDS with Site and Irrigation

```
plot_ordination(physeq_nifH_irr_ord, phynifH_ord_NMDS, shape = "Plot", color = "Site") + geom_point(size
    scale_color_manual(values = farm_col_paired) + ggtitle("nifH NMDS Ordination by Irrigation Method")
    theme_bw()
```




```
tiff("nifH_NMDS_IRR.tiff", width = 6, height = 4, units = "in", res = 600)
plot_ordination(physeq_nifH_irr_ord, phynifH_ord_NMDS, shape = "Plot", color = "Site") + geom_point(size
    scale_color_manual(values = farm_col_paired) + ggtitle("nifH NMDS Ordination by Irrigation Method")
    theme_bw()
dev.off()
```

```
## 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

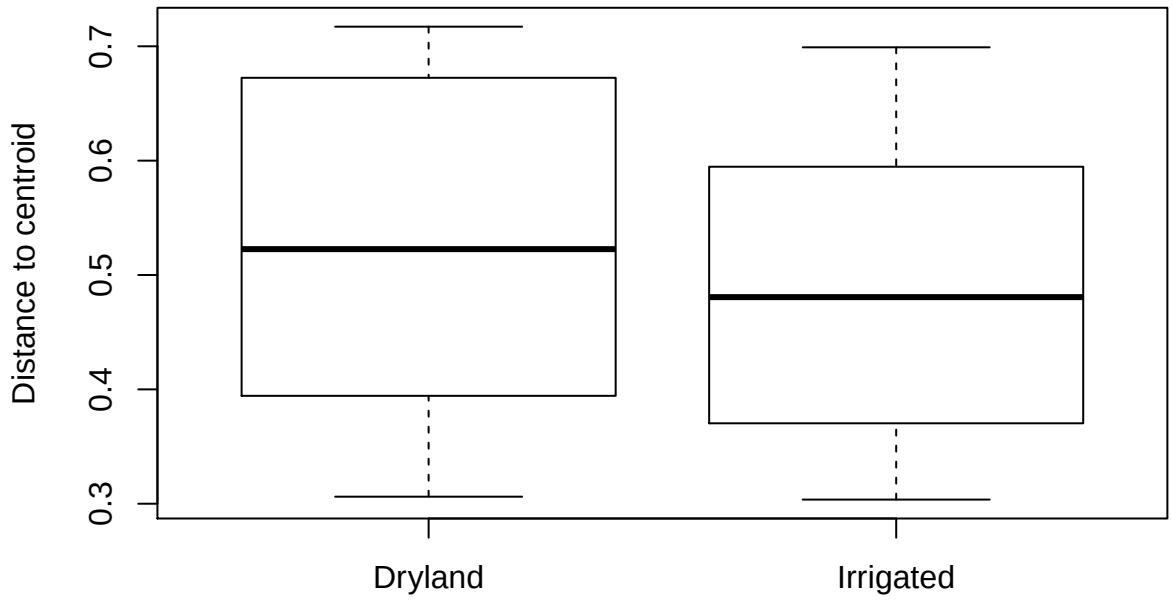
https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1461-0248.2006.00926.x

Irrigation beta dispersion

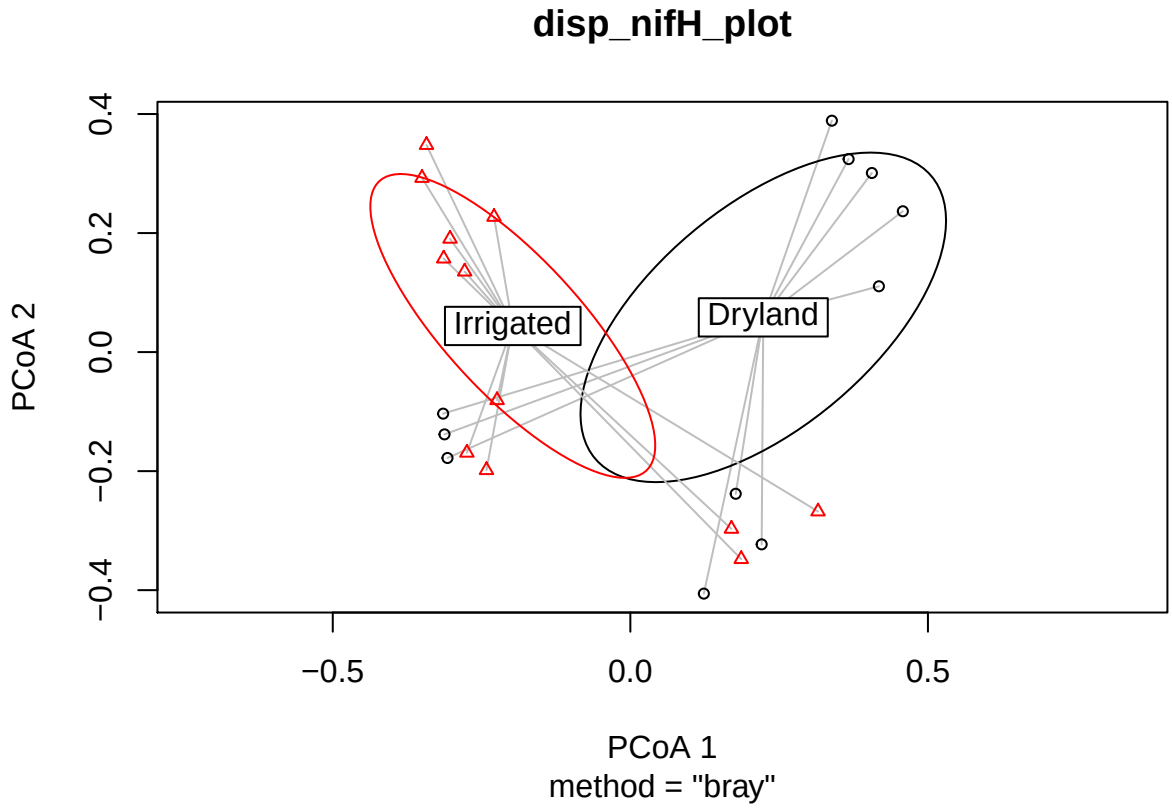
```
disp_nifH_plot <- betadisper(distance(physeq_nifH_irr_ord, method = "bray"), meta_irr$Plot)
permutest(disp_nifH_plot, pairwise = TRUE, permutations = 1000)
```

```
##
## 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 0.00942 0.0094196 0.4658   1000 0.4875
## Residuals 22 0.44493 0.0202239
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##           Dryland Irrigated
## Dryland              0.4865
## Irrigated 0.50206
```

```
boxplot(disp_nifH_plot)
```




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

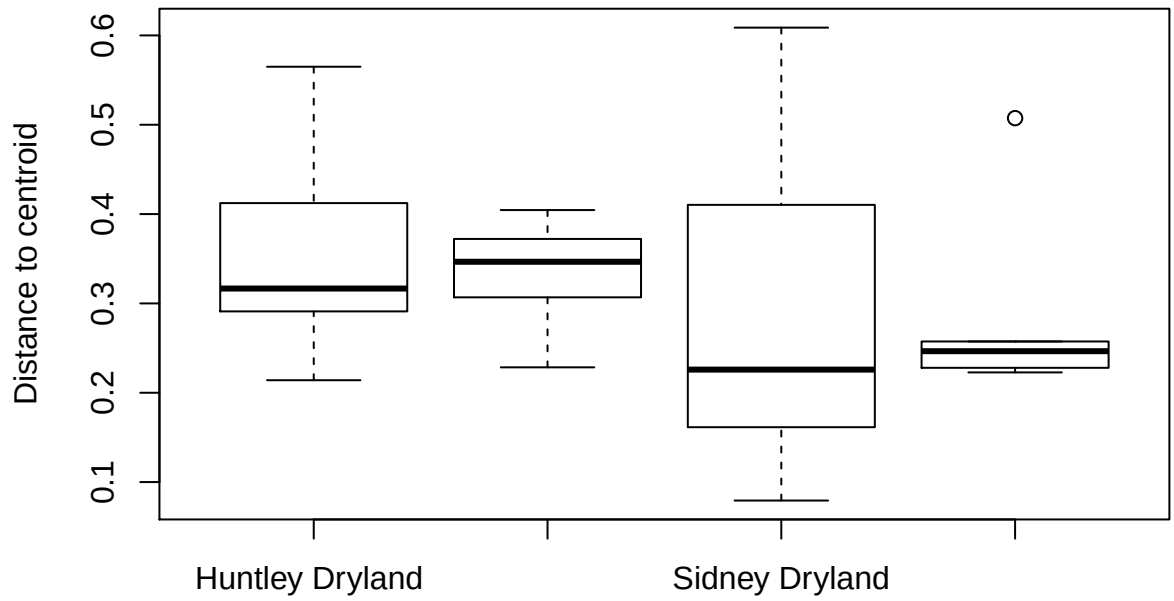



```
disp_nifH_irr_site <- betadisper(distance(physeq_nifH_irr_ord, method = "bray"), meta_irr_nifH$Site)
permutest(disp_nifH_irr_site, pairwise = TRUE, permutations = 1000)
```

```
##
## 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     3 0.02147 0.007157 0.4207   1000 0.7632
## Residuals 20 0.34025 0.017012
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##                   Huntley Dryland  Huntley Irrigated Sidney Dryland
## Huntley Dryland                                0.74725        0.48951
## Huntley Irrigated          0.74958                            0.59441
## Sidney Dryland             0.48672           0.56653
## Sidney Irrigated           0.33714           0.35944        0.99552
##                   Sidney Irrigated
## Huntley Dryland             0.3347
## Huntley Irrigated           0.3676
## Sidney Dryland              0.9940
## Sidney Irrigated
```

```
boxplot(disp_nifH_irr_site)
```

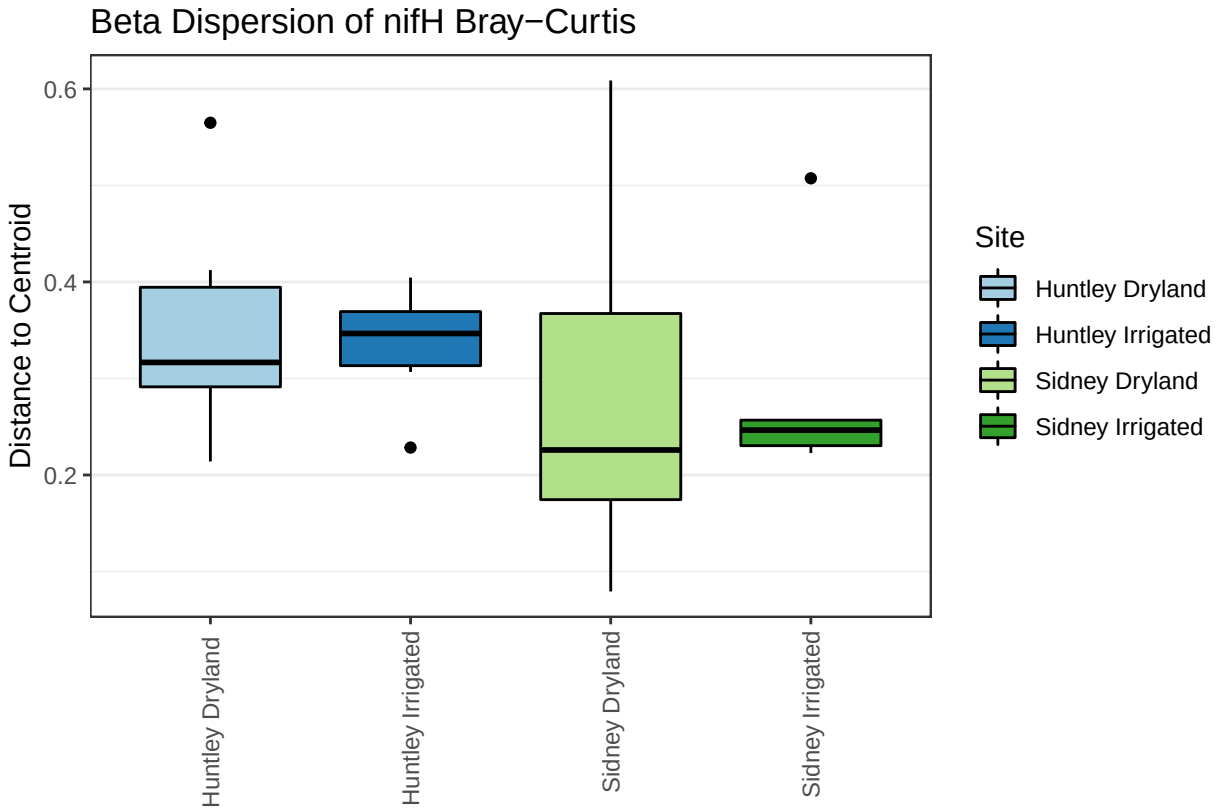



```
TukeyHSD(disp_nifH_irr_site)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##                                           diff        lwr       upr
## Huntley Irrigated-Huntley Dryland  -0.0184096995 -0.2291819 0.1923625
```

```
## Sidney Dryland-Huntley Dryland     -0.0673301192 -0.2781023 0.1434421
## Sidney Irrigated-Huntley Dryland   -0.0678516310 -0.2786238 0.1429206
## Sidney Dryland-Huntley Irrigated   -0.0489204198 -0.2596926 0.1618518
## Sidney Irrigated-Huntley Irrigated -0.0494419316 -0.2602141 0.1613303
## Sidney Irrigated-Sidney Dryland    -0.0005215118 -0.2112937 0.2102507
##                                        p adj
## Huntley Irrigated-Huntley Dryland  0.9946960
## Sidney Dryland-Huntley Dryland     0.8079557
## Sidney Irrigated-Huntley Dryland   0.8044007
## Sidney Dryland-Huntley Irrigated   0.9144374
## Sidney Irrigated-Huntley Irrigated 0.9119961
## Sidney Irrigated-Sidney Dryland    0.9999999
```

```
dispersion_nifH_site <- data.frame(Distance_to_centroid = disp_nifH_irr_site$distances, Site = disp_nifH
ggboxplot(dispersion_nifH_site, x = "Site", y = "Distance_to_centroid", rug = TRUE, fill = "Site", ylab
    xlab = " ", title = "Beta Dispersion of nifH Bray-Curtis", palette = farm_col_paired) + theme_bw() +
    theme(panel.grid.major.x = element_blank()) + rotate_x_text()
```




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

PERMANOVA (adoins)

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

First try univariate fram managment

```
adonis(distance(physeq_nifH_irr_ord, method = "bray") ~ Plot, data = meta_irr, permutations = 1000)
```

```
##
## Call:
## adonis(formula = distance(physeq_nifH_irr_ord, method = "bray") ~      Plot, data = meta_irr, permuta
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2   Pr(>F)
## Plot       1    0.9800 0.97998  3.3401 0.13181 0.004995 **
## Residuals 22    6.4547 0.29339         0.86819
## Total     23    7.4347                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(distance(physeq_nifH_irr_ord, method = "bray") ~ Plot * Site, data = meta_irr, permutations = 10
```

```
##
## Call:
## adonis(formula = distance(physeq_nifH_irr_ord, method = "bray") ~      Plot * Site, data = meta_irr,
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2   Pr(>F)
## Plot       1    0.9800 0.97998  4.9167 0.13181 0.000999 ***
## Site       2    2.4683 1.23417  6.1920 0.33200 0.000999 ***
## Residuals 20    3.9863 0.19932         0.53618
## Total     23    7.4347                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model Selection

ENVFIT

Envfit does not like single variable values so we remove them

```
meta3 <- meta_irr_nifH[, -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)]
```

```
envfit_nifH_irr <- envfit(physeq_nifH_irr_DCA, meta3, na.rm = TRUE, permu = 10000)
envfit_nifH_irr
```

```
##
## ***VECTORS
##
##                      DCA1     DCA2     r2    Pr(>r)
## season_precip     0.24968  0.96833 0.5206 0.0006999 ***
## irrgation        -0.77276  0.63470 0.8213 9.999e-05 ***
## total_precip_irr -0.22301  0.97482 0.5339 0.0005000 ***
## grain_yield       0.02960  0.99956 0.3877 0.0066993 **
## Organic_Matter   -0.39993  0.91655 0.6253 0.0002000 ***
## Moisture_Content -0.74014  0.67245 0.6469 9.999e-05 ***
## Nitrate_Nitrite  -0.50581  0.86264 0.5406 0.0005999 ***
## Ammonia          -0.32434 -0.94594 0.6918 9.999e-05 ***
## Av_Phosphorus    -0.10562  0.99441 0.7333 9.999e-05 ***
## Av_Potassium     -0.70782  0.70639 0.8917 9.999e-05 ***
## pH               -0.64549  0.76377 0.5679 0.0002000 ***
## Barium           -0.46723 -0.88414 0.1984 0.0644936 .
## Calcium           0.18360  0.98300 0.5258 0.0007999 ***
## Cobalt           -0.24026 -0.97071 0.2495 0.0252975 *
## Copper           -0.99910 -0.04247 0.3729 0.0046995 **
## Iron             -0.63279 -0.77432 0.3135 0.0077992 **
## Magnesium        -0.57740  0.81646 0.5687 0.0003000 ***
## Manganese        -0.38914  0.92118 0.1994 0.0868913 .
## Nickel            0.78143 -0.62399 0.2522 0.0389961 *
## Phosphorus       -0.83129  0.55584 0.7918 9.999e-05 ***
## Sulfur           -0.13038  0.99146 0.4687 0.0018998 **
## Zinc             -0.95269 -0.30393 0.4614 0.0008999 ***
## Chao1            -0.86613 -0.49981 0.8117 9.999e-05 ***
## se.chao1         -0.67715  0.73584 0.0215 0.7945205
```

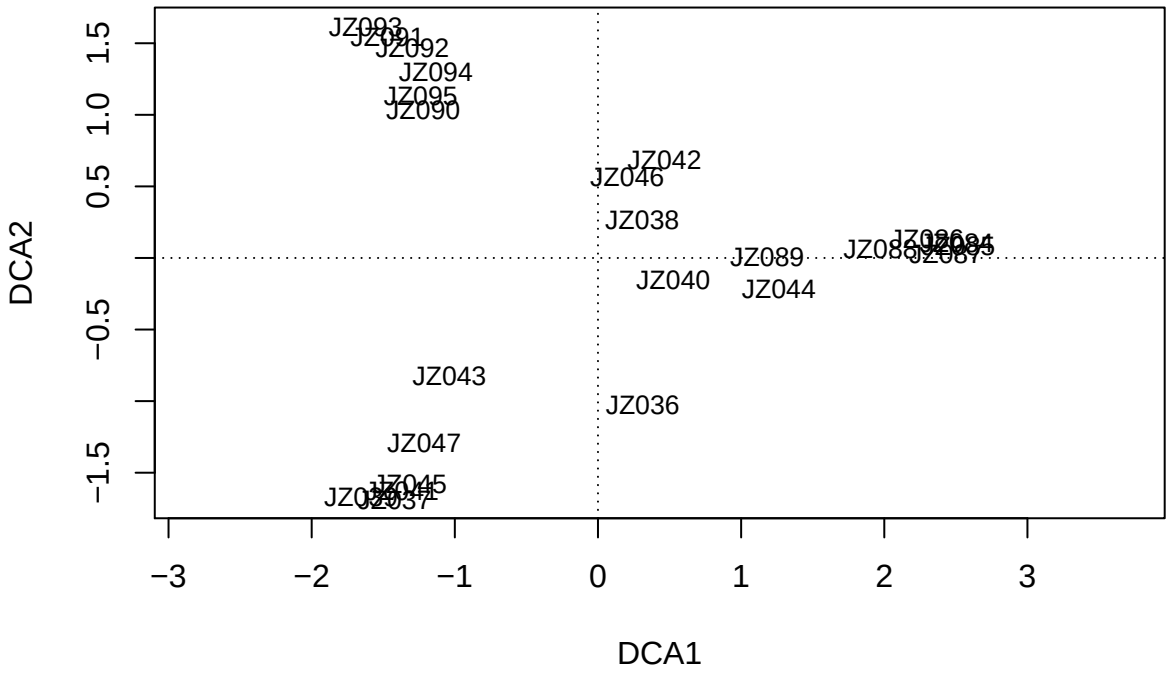
```
## ACE              -0.87011 -0.49286 0.8055 9.999e-05 ***
## se.ACE           -0.87368 -0.48649 0.7672 9.999e-05 ***
## Shannon          -0.73987 -0.67275 0.7299 9.999e-05 ***
## Simpson          -0.72367 -0.69015 0.5002 0.0010999 **
## InvSimpson       -0.53579 -0.84435 0.7122 9.999e-05 ***
## Fisher           -0.84023 -0.54223 0.8071 9.999e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 10000
##
## ***FACTORS:
##
## Centroids:
##                             DCA1    DCA2
## SiteHuntley Dryland       0.5135  0.0224
## SiteHuntley Irrigated    -1.3352 -1.4450
## SiteSidney Dryland        2.1517  0.0718
## SiteSidney Irrigated     -1.3301  1.3508
## Pea_varietyAC Earlystar   0.0440 -0.3918
## Pea_varietyCDC Saffron   -0.6724 -0.7002
## Pea_varietyCDC Saffron    0.5238  0.8143
## Pea_varietyDelta          0.0395 -0.0437
## Pea_varietyDS Admiral     0.0591  0.3775
## Pea_varietyMajoret        0.1979 -0.1048
## Pea_varietyNavarro       -0.2662  0.1058
## PlotDryland               1.3326  0.0471
## PlotIrrigated            -1.3326 -0.0471
## TillageConventional       0.4108  0.7113
## TillageNo_till           -0.4108 -0.7113
## prev_cropbarley          -1.3352 -1.4450
## prev_cropChem_fallow      1.3326  0.0471
## prev_cropSpring_wheat    -1.3301  1.3508
##
## Goodness of fit:
##                  r2    Pr(>r)
## Site         0.9336 9.999e-05 ***
## Pea_variety  0.0693  0.997400
## Plot         0.5370 9.999e-05 ***
## Tillage      0.2038  0.009699 **
## prev_crop    0.8322 9.999e-05 ***
## ---
## 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 envriomental vectors to the bray-curtis species dissmilarity matrix in NMDS space. This is a loose corelation to real linear corelation but it can tell us how the NMDS orinetaion is being driven.

write to table

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

```
plot(physeq_nifH_irr_DCA, display = "sites")
```



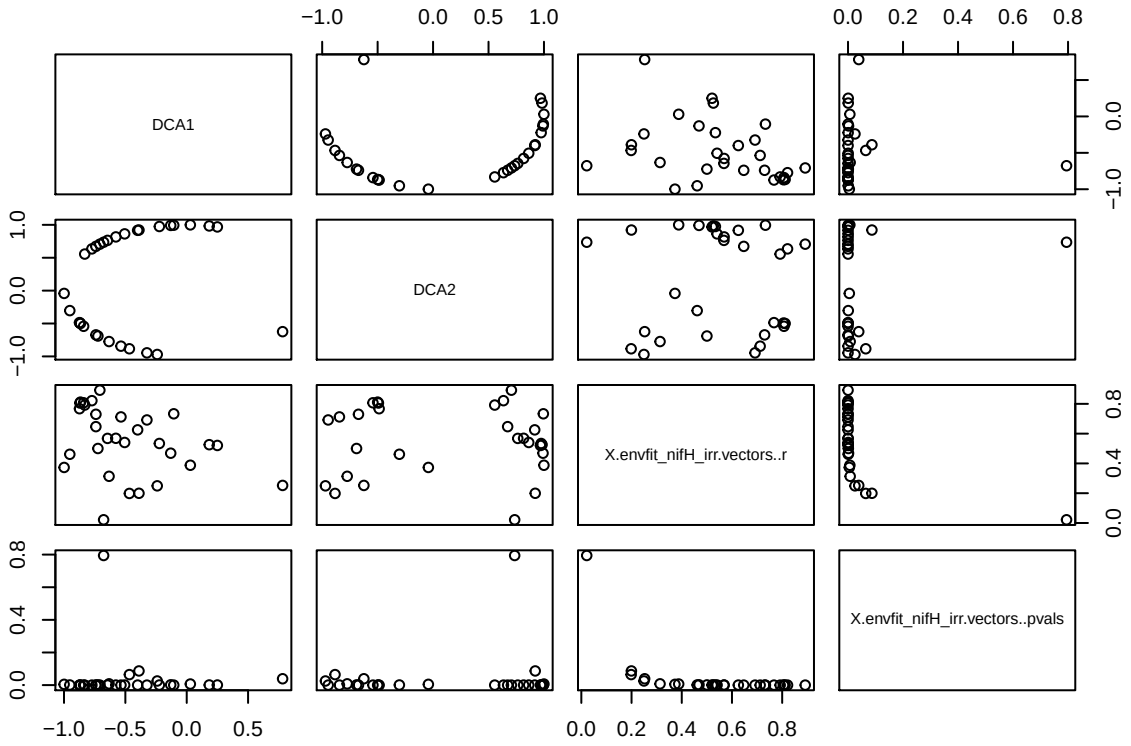

```
plot(envfit_nifH_irr, p.max = 0.001)
```

```
## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "p.max" is not
## a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "p.max" is not a
## graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "p.max" is not a
## graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "p.max" is not
## a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "p.max" is not
## a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "p.max" is not a
## graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "p.max" is not
## a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "p.max" is not a
## graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "p.max" is not a
## graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter
```

```
## Warning in title(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "p.max" is not a
## graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "p.max" is not a
## graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "p.max" is not a
## graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "p.max" is not
## a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "p.max" is not a
## graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "p.max" is not
## a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "p.max" is not
## a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "p.max" is not a
## graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "p.max" is not a
## graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "p.max" is not
## a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "p.max" is not a
## graphical parameter

## Warning in plot.window(...): "p.max" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "p.max" is not a graphical parameter

## Warning in title(...): "p.max" is not a graphical parameter
```



```
OTU_nifH_trim <- as(otu_table(physeq_nifH_irr_ord), "matrix")
```

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

nrow(abund_table_irr_nifH)
```

```
## [1] 24
```

```
setdiff(rownames(meta3), rownames(abund_table_irr_nifH))
```

```
## 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
```

```r
# get bray-curtis distance
abund_dist_nifH_irr <- vegdist(abund_table_irr_nifH, method = "bray")
```

Remove site from meta table

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

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 smae 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.

```r
m1_irr_nifH <- cca(abund_table_irr_nifH ~ ., meta3)
m0_irr_nifH <- cca(abund_table_irr_nifH ~ 1, meta3)
m1_irr_nifH
```

```
## Call: cca(formula = abund_table_irr_nifH ~ 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 + Chao1 + se.chao1 + ACE +
## se.ACE + Shannon + Simpson + InvSimpson + Fisher, data = meta3)
##
##               Inertia Proportion Rank
## Total             4.4        1.0
## Constrained       4.4        1.0   23
## Unconstrained     0.0        0.0    0
## 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  CCA10
## 0.7260 0.6410 0.5985 0.3250 0.2689 0.2351 0.2117 0.1848 0.1793 0.1665
##  CCA11  CCA12  CCA13  CCA14  CCA15  CCA16  CCA17  CCA18  CCA19  CCA20
## 0.1443 0.1149 0.0997 0.0835 0.0752 0.0706 0.0618 0.0548 0.0406 0.0390
##  CCA21  CCA22  CCA23
## 0.0353 0.0259 0.0181
```

```r
m0_irr_nifH
```

```
## Call: cca(formula = abund_table_irr_nifH ~ 1, data = meta3)
##
##               Inertia Rank
## Total             4.4
## Unconstrained     4.4   23
## Inertia is scaled Chi-square
##
## Eigenvalues for unconstrained axes:
##    CA1    CA2    CA3    CA4    CA5    CA6    CA7    CA8
## 0.7260 0.6410 0.5985 0.3250 0.2689 0.2351 0.2117 0.1848
## (Showing 8 of 23 unconstrained eigenvalues)
```

Ordistep

```
model_irr_nifH <- ordistep(m0_irr_nifH, scope = formula(m1_irr_nifH))
```

```
model_irr_nifH$anova
```

	Df	AIC	F	Pr(>F)
+ Site	3	361.9826	4.864143	0.005
+ Copper	1	361.4551	2.110175	0.010
+ Moisture_Content	1	361.4269	1.587238	0.025

Site is again nesting the data in the model selection

CCA without site

```
m1_table_irr_nifH_site_na <- cca(abund_table_irr_nifH ~ ., meta4)
m0_table_irr_nifH_site_na <- cca(abund_table_irr_nifH ~ 1, meta4)
m1_table_irr_nifH_site_na
```

```
## Call: cca(formula = abund_table_irr_nifH ~ 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 + Chao1 + se.chao1 + ACE +
## se.ACE + Shannon + Simpson + InvSimpson + Fisher, data = meta4)
##
##               Inertia Proportion Rank
## Total             4.4        1.0
## Constrained       4.4        1.0   23
## Unconstrained     0.0        0.0    0
## 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  CCA10
## 0.7260 0.6410 0.5985 0.3250 0.2689 0.2351 0.2117 0.1848 0.1793 0.1665
```

```
##  CCA11  CCA12  CCA13  CCA14  CCA15  CCA16  CCA17  CCA18  CCA19  CCA20
## 0.1443 0.1149 0.0997 0.0835 0.0752 0.0706 0.0618 0.0548 0.0406 0.0390
##  CCA21  CCA22  CCA23
## 0.0353 0.0259 0.0181
```

```
m0_table_irr_nifH_site_na
```

```
## Call: cca(formula = abund_table_irr_nifH ~ 1, data = meta4)
##
##               Inertia Rank
## Total             4.4
## Unconstrained     4.4   23
## Inertia is scaled Chi-square
##
## Eigenvalues for unconstrained axes:
##    CA1    CA2    CA3    CA4    CA5    CA6    CA7    CA8
## 0.7260 0.6410 0.5985 0.3250 0.2689 0.2351 0.2117 0.1848
## (Showing 8 of 23 unconstrained eigenvalues)
```

Ordistep

```
model_table_irr_nifH_site_na <- ordistep(m0_table_irr_nifH_site_na, scope = formula(m1_table_irr_nifH_s
```

```
model_table_irr_nifH_site_na$anova
```

	Df	AIC	F	Pr(>F)
+ prev_crop	2	364.7878	4.365772	0.005
+ total_precip_irr	1	361.9826	4.433344	0.005
+ Phosphorus	1	361.5411	2.034587	0.005
+ Moisture_Content	1	361.4867	1.608709	0.030

Constrained Ordination plot in ggplot

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

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

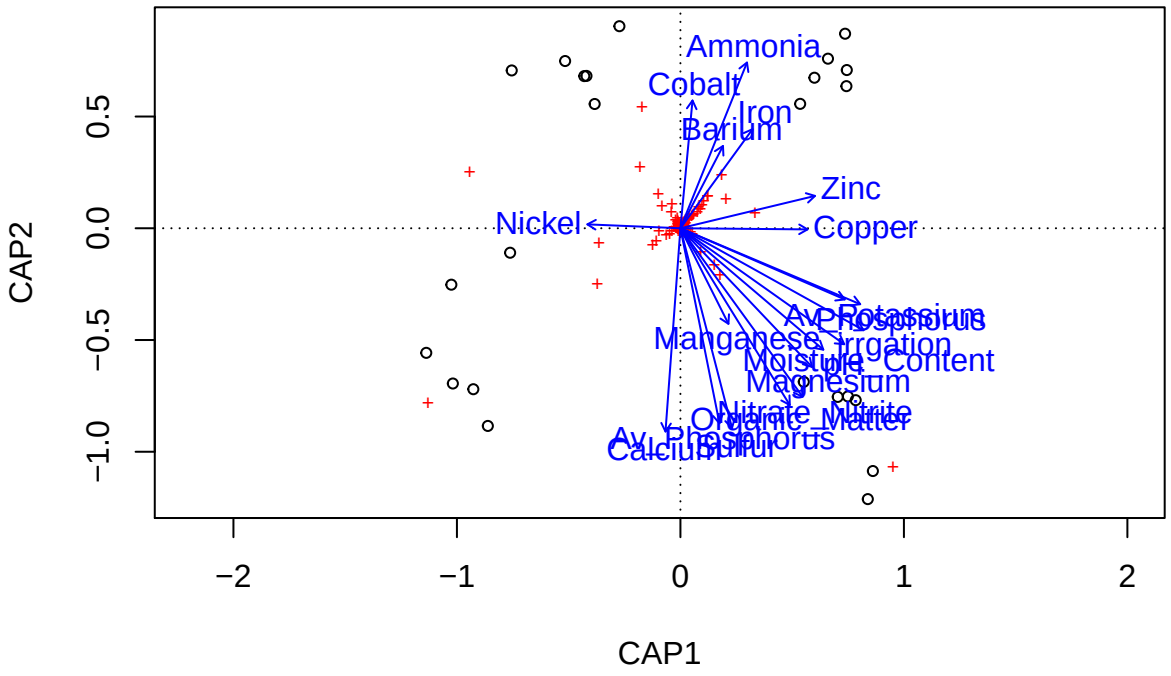
https://quantpalaeo.wordpress.com/2014/04/14/variance-inflation-factors-and-ordination-model-selection/

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_nifH_irr_cap_chem <- capscale(abund_table_irr_nifH ~ irrgation + 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, distance = "bray")
m0_16s_cap_chem <- capscale(abund_table_irr_nifH ~ 1, meta3)
plot(m1_nifH_irr_cap_chem)
```



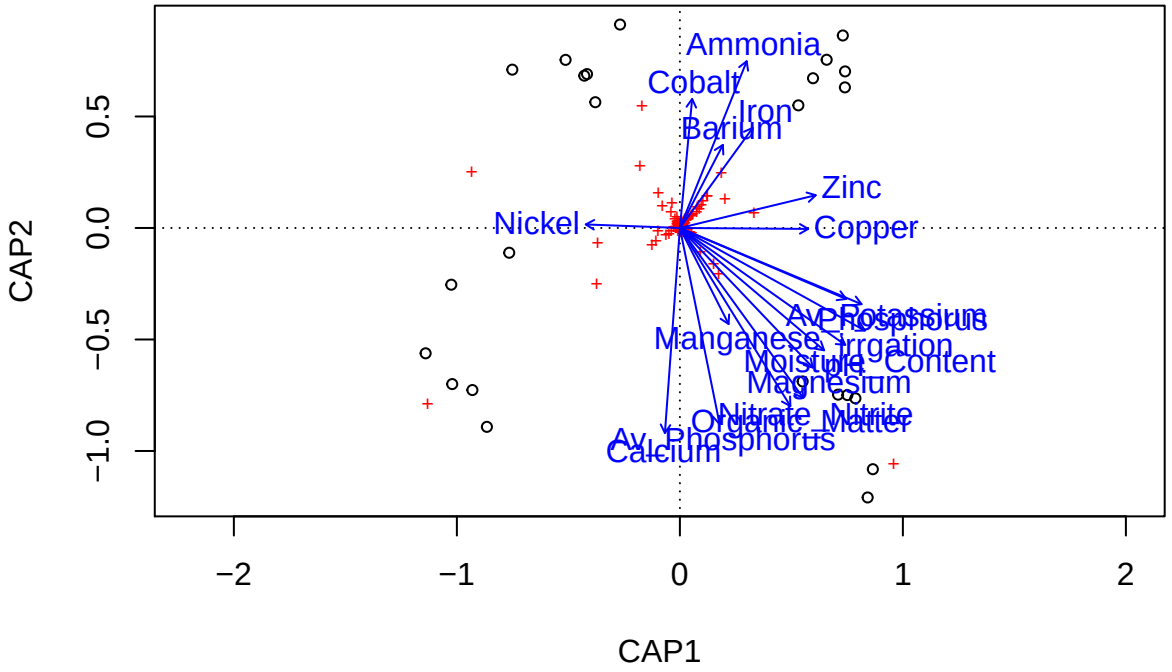

```
vif.cca(m1_nifH_irr_cap_chem)
```

```
##        irrgation   Organic_Matter Moisture_Content  Nitrate_Nitrite
##        146.63511        264.59858         43.02707         28.33216
##          Ammonia    Av_Phosphorus     Av_Potassium               pH
##         11.03965         86.96226         66.51282         20.28679
##           Barium          Calcium           Cobalt           Copper
##         71.86241        189.13062         76.20825         80.85081
##             Iron        Magnesium        Manganese           Nickel
##        157.67041        105.45005         58.16505         15.85561
##       Phosphorus           Sulfur             Zinc
##        215.83777        479.07896        173.10384
```

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

Droping Sulfur from the model

```
m1_nifH_irr_cap_chem_1 <- capscale(abund_table_irr_nifH ~ irrgation + Organic_Matter + Moisture_Content
    Nitrate_Nitrite + Ammonia + Av_Phosphorus + Av_Potassium + pH + Barium + Calcium + Cobalt + Copper +
    Iron + Magnesium + Manganese + Nickel + Phosphorus + Zinc, data = meta3, distance = "bray")
m1_nifH_irr_cap_chem <- capscale(abund_table_irr_nifH ~ 1, meta3)
plot(m1_nifH_irr_cap_chem_1)
```

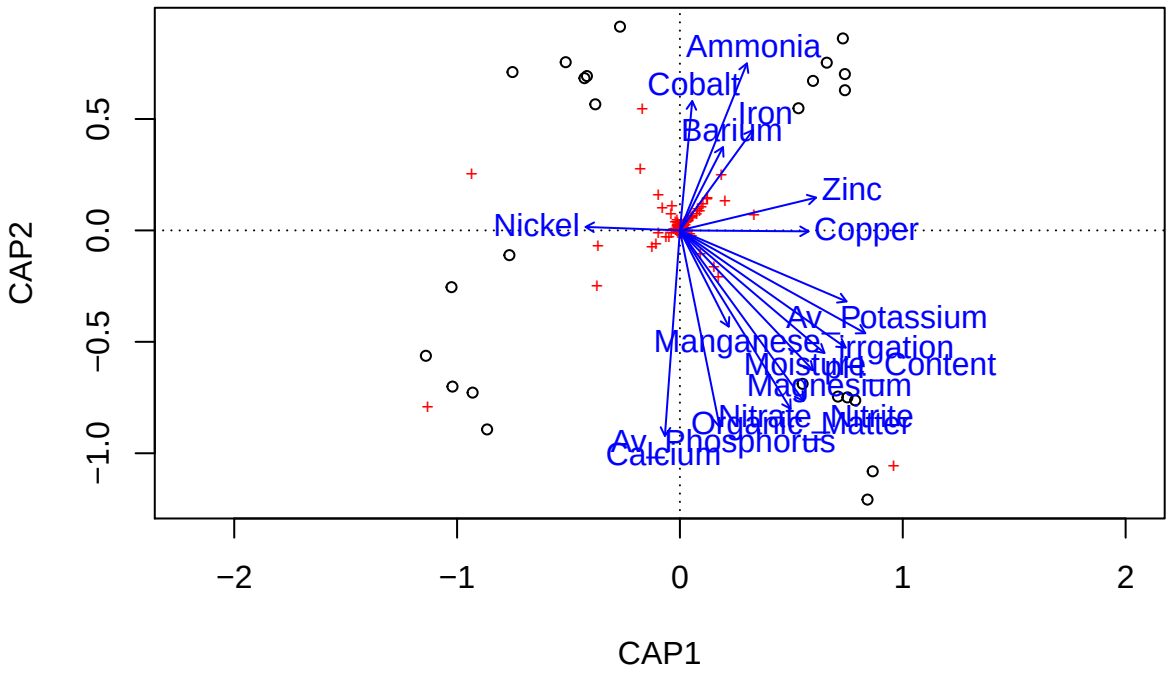



```
vif.cca(m1_nifH_irr_cap_chem_1)
```

```
##        irrgation   Organic_Matter Moisture_Content  Nitrate_Nitrite
##        146.50106        175.89795         39.10162         22.78882
##          Ammonia    Av_Phosphorus     Av_Potassium               pH
##         10.93461         84.77130         47.29707         17.62232
##           Barium          Calcium           Cobalt           Copper
##         67.52866         50.44216         71.74590         78.05182
##             Iron        Magnesium        Manganese           Nickel
##        153.62189        105.44201         49.59095         15.67621
##       Phosphorus             Zinc
##        193.53177        170.39086
```

Dropping Phosporus from the model

```
m1_nifH_irr_cap_chem_2 <- capscale(abund_table_irr_nifH ~ irrgation + Organic_Matter + Moisture_Content
    Nitrate_Nitrite + Ammonia + Av_Phosphorus + Av_Potassium + pH + Barium + Calcium + Cobalt + Copper +
    Iron + Magnesium + Manganese + Nickel + Zinc, data = meta3, distance = "bray")
m1_nifH_irr_cap_chem <- capscale(abund_table_irr_nifH ~ 1, meta3)
plot(m1_nifH_irr_cap_chem_2)
```

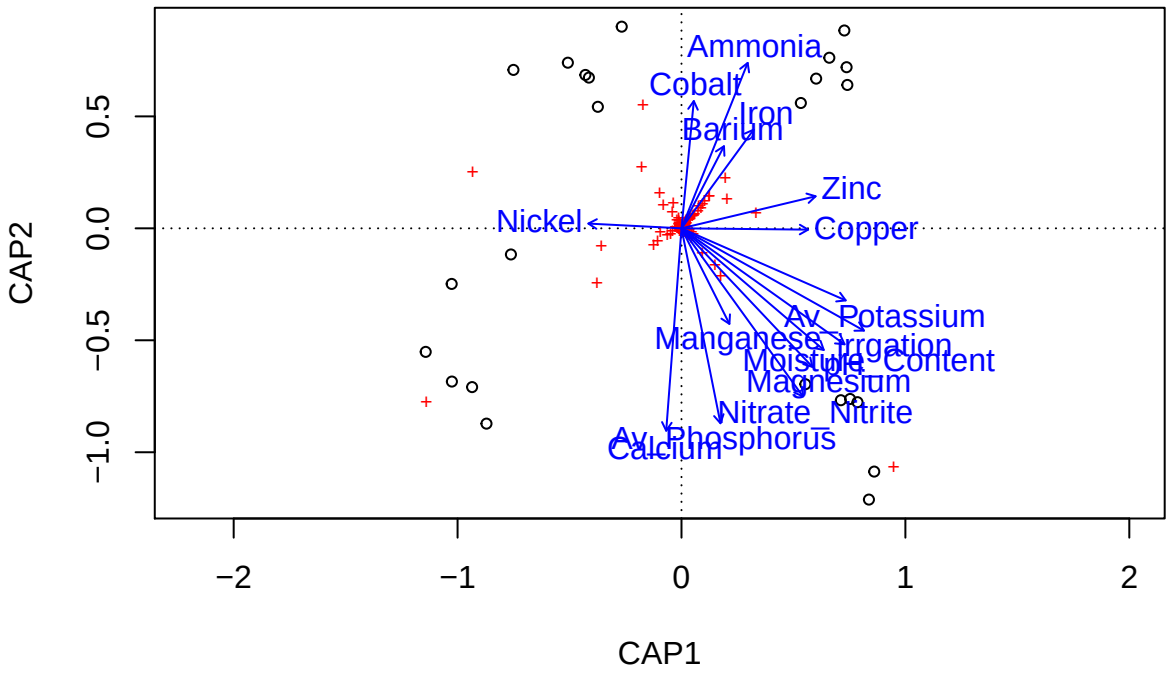



```
vif.cca(m1_nifH_irr_cap_chem_2)
```

```
##        irrgation   Organic_Matter Moisture_Content  Nitrate_Nitrite
##       134.614685       172.652131        39.070713        22.780698
##          Ammonia    Av_Phosphorus     Av_Potassium               pH
##         8.427309        57.602123        43.265329        17.516511
##           Barium          Calcium           Cobalt           Copper
##        57.036374        49.760956        71.726460        60.601967
##             Iron        Magnesium        Manganese           Nickel
##       133.383936        90.034114        37.826872        15.289941
##             Zinc
##       165.303403
```

Dropping Organic Matter

```
m1_nifH_irr_cap_chem_3 <- capscale(abund_table_irr_nifH ~ irrgation + Moisture_Content + Nitrate_Nitrite
    Ammonia + Av_Phosphorus + Av_Potassium + pH + Barium + Calcium + Cobalt + Copper + Iron + Magnesium
    Manganese + Nickel + Zinc, data = meta3, distance = "bray")
m1_nifH_irr_cap_chem <- capscale(abund_table_irr_nifH ~ 1, meta3)
plot(m1_nifH_irr_cap_chem_3)
```

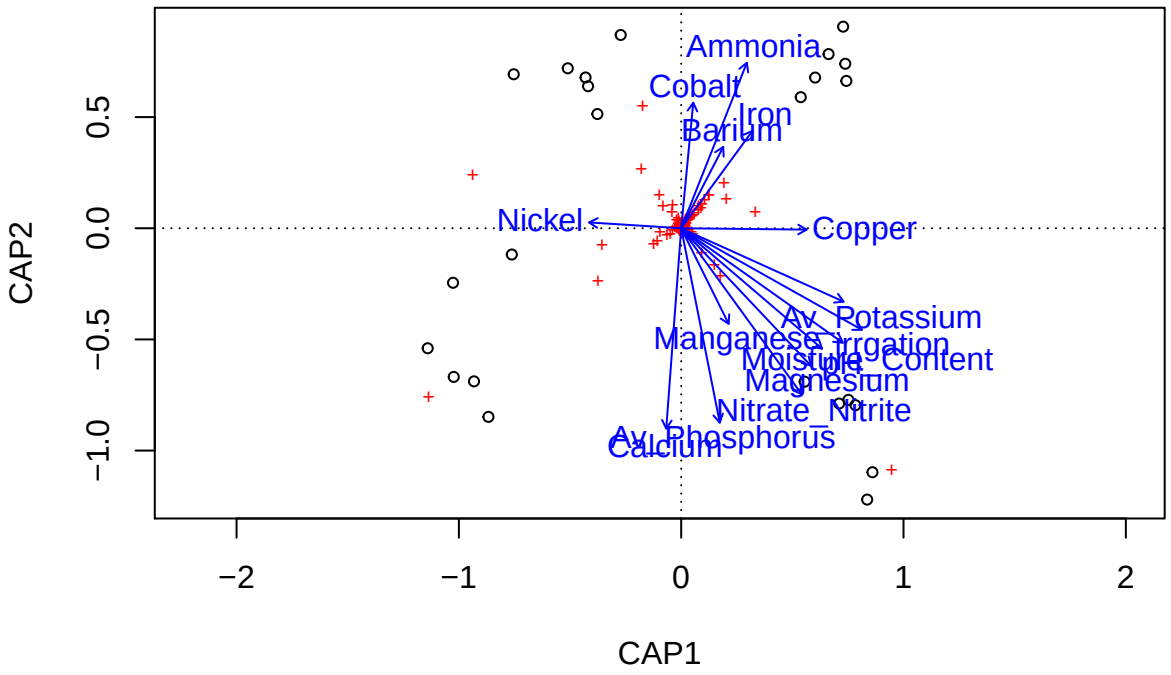



```
vif.cca(m1_nifH_irr_cap_chem_3)
```

```
##        irrgation Moisture_Content  Nitrate_Nitrite          Ammonia
##        133.41172         30.57461         22.59306          6.89853
##    Av_Phosphorus     Av_Potassium               pH          Barium
##         31.22720         42.04839         17.14479         56.52610
##          Calcium           Cobalt           Copper             Iron
##         44.53383         55.18950         60.45348        114.52317
##        Magnesium        Manganese           Nickel             Zinc
##         65.43465         37.45890         10.13252        158.88600
```

Dropping Zinc

```
m1_nifH_irr_cap_chem_4 <- capscale(abund_table_irr_nifH ~ irrgation + Moisture_Content + Nitrate_Nitrite
    Ammonia + Av_Phosphorus + Av_Potassium + pH + Barium + Calcium + Cobalt + Copper + Iron + Magnesium
    Manganese + Nickel, data = meta3, distance = "bray")
m1_nifH_irr_cap_chem <- capscale(abund_table_irr_nifH ~ 1, meta3)
plot(m1_nifH_irr_cap_chem_4)
```

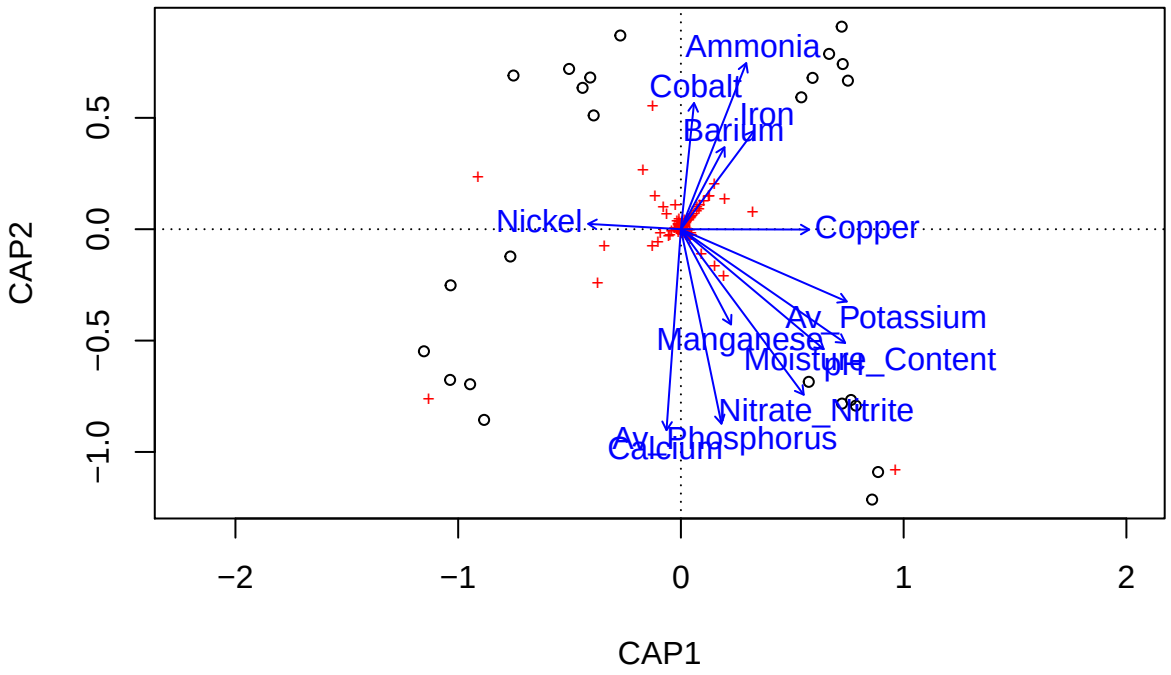



```
vif.cca(m1_nifH_irr_cap_chem_4)
```

```
##        irrgation Moisture_Content  Nitrate_Nitrite          Ammonia
##       131.870601        26.362849        19.937767         6.894727
##    Av_Phosphorus     Av_Potassium               pH          Barium
##       31.173959        41.651455        16.875334        56.295044
##          Calcium           Cobalt           Copper             Iron
##       40.047047        39.923453        50.877617        48.951973
##        Magnesium        Manganese           Nickel
##       64.529436        24.042910        10.132444
```

Removing irragtion

```
m1_nifH_irr_cap_chem_5 <- capscale(abund_table_irr_nifH ~ Moisture_Content + Nitrate_Nitrite + Ammonia +
    Av_Phosphorus + Av_Potassium + pH + Barium + Calcium + Cobalt + Copper + Iron + Manganese + Nickel,
    data = meta3, distance = "bray")
m1_nifH_irr_cap_chem <- capscale(abund_table_irr_nifH ~ 1, meta3)
plot(m1_nifH_irr_cap_chem_5)
```

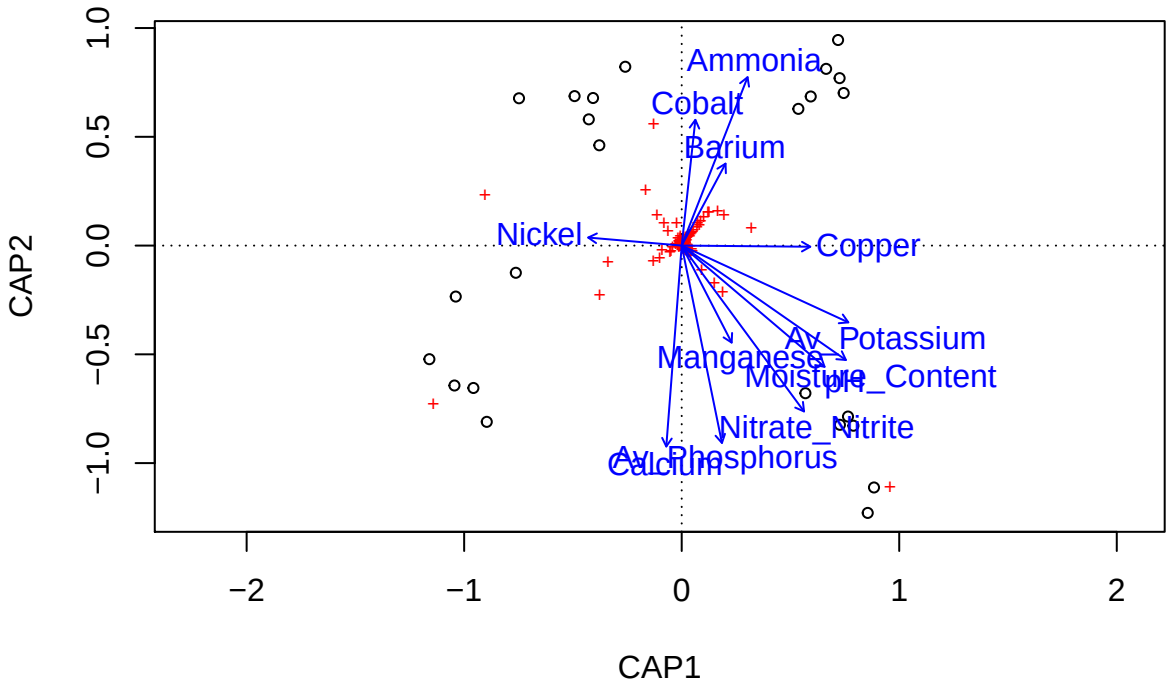



```
vif.cca(m1_nifH_irr_cap_chem_5)
```

```
## Moisture_Content  Nitrate_Nitrite          Ammonia    Av_Phosphorus
##        19.323415        18.330891         6.107546        26.225579
##     Av_Potassium               pH          Barium          Calcium
##        17.986948         7.831355        44.202690        16.901864
##           Cobalt           Copper             Iron        Manganese
##        31.681897        29.757930        47.965398        20.641916
##           Nickel
##         8.835635
```

Dropping iron

```
m1_nifH_irr_cap_chem_6 <- capscale(abund_table_irr_nifH ~ Moisture_Content + Nitrate_Nitrite + Ammonia +
    Av_Phosphorus + Av_Potassium + pH + Barium + Calcium + Cobalt + Copper + Manganese + Nickel, data =
    distance = "bray")
m1_nifH_irr_cap_chem <- capscale(abund_table_irr_nifH ~ 1, meta3)
plot(m1_nifH_irr_cap_chem_6)
```

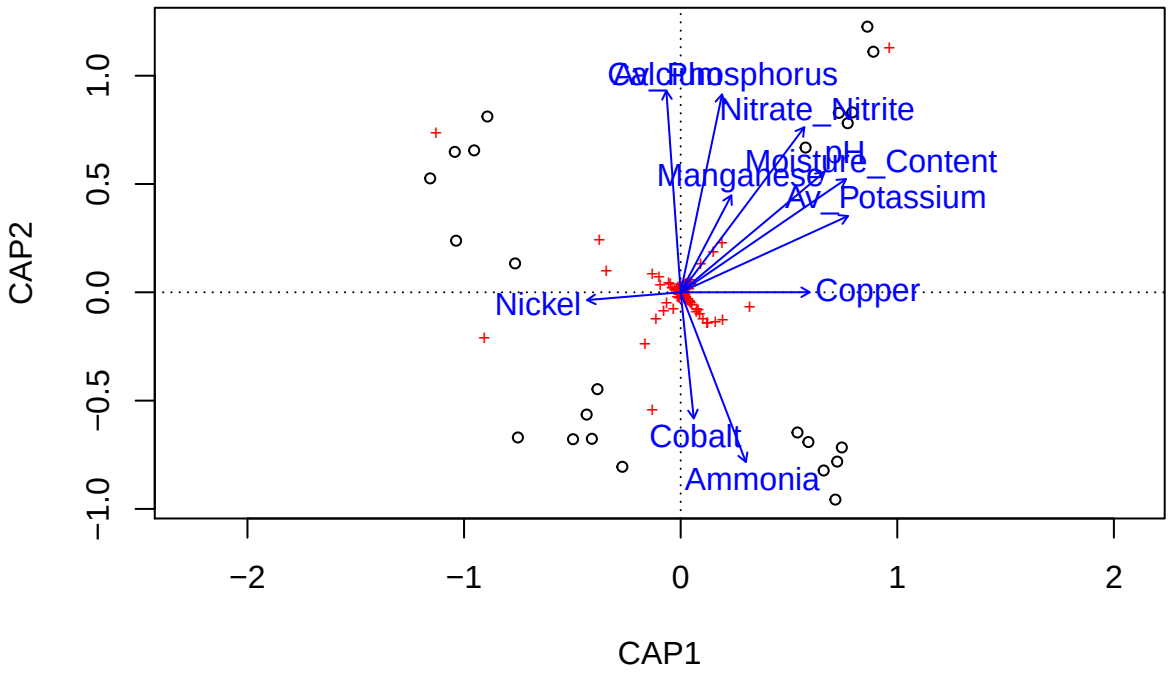



```
vif.cca(m1_nifH_irr_cap_chem_6)
```

```
## Moisture_Content  Nitrate_Nitrite          Ammonia     Av_Phosphorus
##        17.318792        18.330471         5.629288        21.601310
##     Av_Potassium               pH          Barium          Calcium
##        13.959671         7.161925        32.994173        13.788783
##           Cobalt           Copper        Manganese           Nickel
##        30.842734        29.395092        12.013136         8.441058
```

Droping BArium

```
m1_nifH_irr_cap_chem_7 <- capscale(abund_table_irr_nifH ~ Moisture_Content + Nitrate_Nitrite + Ammonia +
    Av_Phosphorus + Av_Potassium + pH + Calcium + Cobalt + Copper + Manganese + Nickel, data = meta3,
    distance = "bray")
m1_nifH_irr_cap_chem <- capscale(abund_table_irr_nifH ~ 1, meta3)
plot(m1_nifH_irr_cap_chem_7)
```

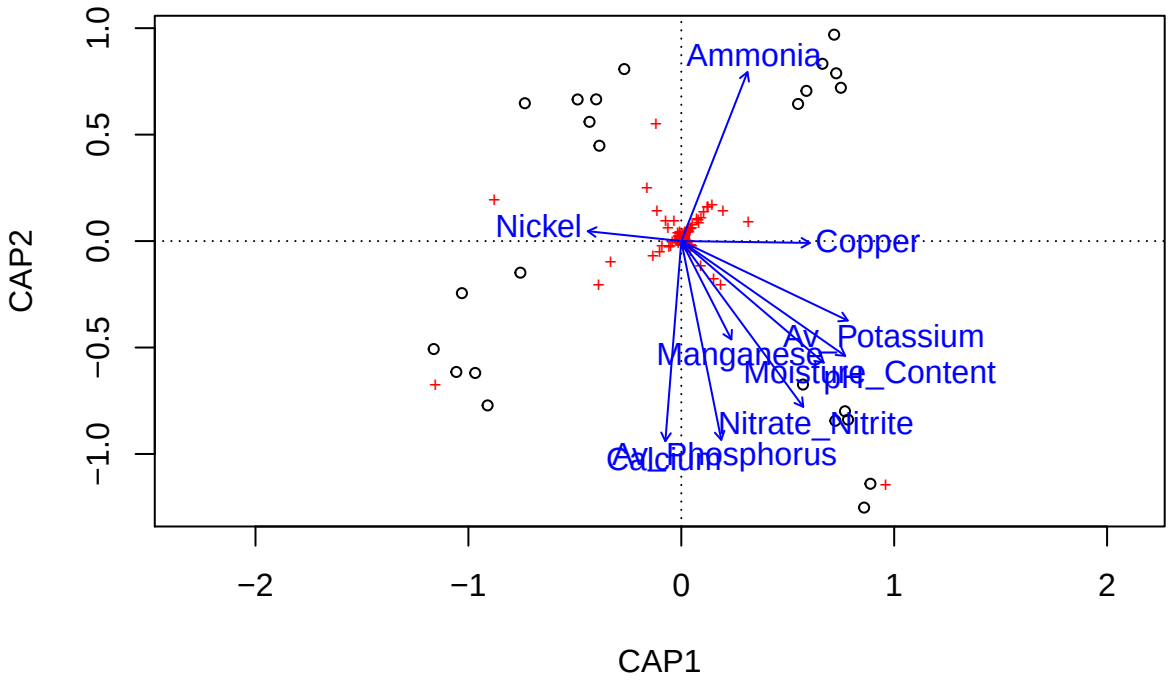



```
vif.cca(m1_nifH_irr_cap_chem_7)
```

```
## Moisture_Content  Nitrate_Nitrite          Ammonia    Av_Phosphorus
##        15.688324        16.382037         4.960616        19.025242
##     Av_Potassium               pH          Calcium           Cobalt
##        13.959622         6.775731        13.713988        26.493378
##           Copper        Manganese           Nickel
##        16.514124        11.952580         8.404323
```

Droping Cobalt

```
m1_nifH_irr_cap_chem_8 <- capscale(abund_table_irr_nifH ~ Moisture_Content + Nitrate_Nitrite + Ammonia +
    Av_Phosphorus + Av_Potassium + pH + Calcium + Copper + Manganese + Nickel, data = meta3, distance =
m1_nifH_irr_cap_chem <- capscale(abund_table_irr_nifH ~ 1, meta3)
plot(m1_nifH_irr_cap_chem_8)
```

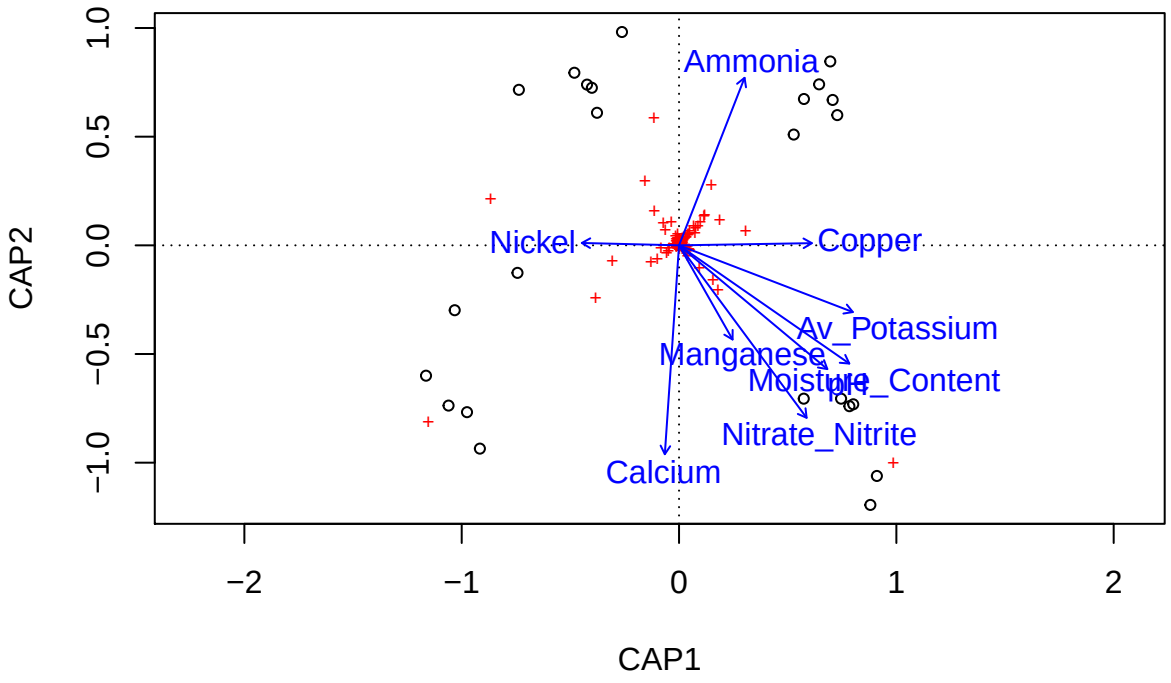



```
vif.cca(m1_nifH_irr_cap_chem_8)
```

```
## Moisture_Content  Nitrate_Nitrite          Ammonia    Av_Phosphorus
##        15.459077        14.224287         4.960535        18.595385
##     Av_Potassium               pH          Calcium           Copper
##        11.859534         6.745006         9.744613        13.420297
##        Manganese           Nickel
##        10.412205         5.120134
```

Droping Av_Phosphorus

```
m1_nifH_irr_cap_chem_9 <- capscale(abund_table_irr_nifH ~ Moisture_Content + Nitrate_Nitrite + Ammonia +
    Av_Potassium + pH + Calcium + Copper + Manganese + Nickel, data = meta3, distance = "bray")
m1_nifH_irr_cap_chem <- capscale(abund_table_irr_nifH ~ 1, meta3)
plot(m1_nifH_irr_cap_chem_9)
```

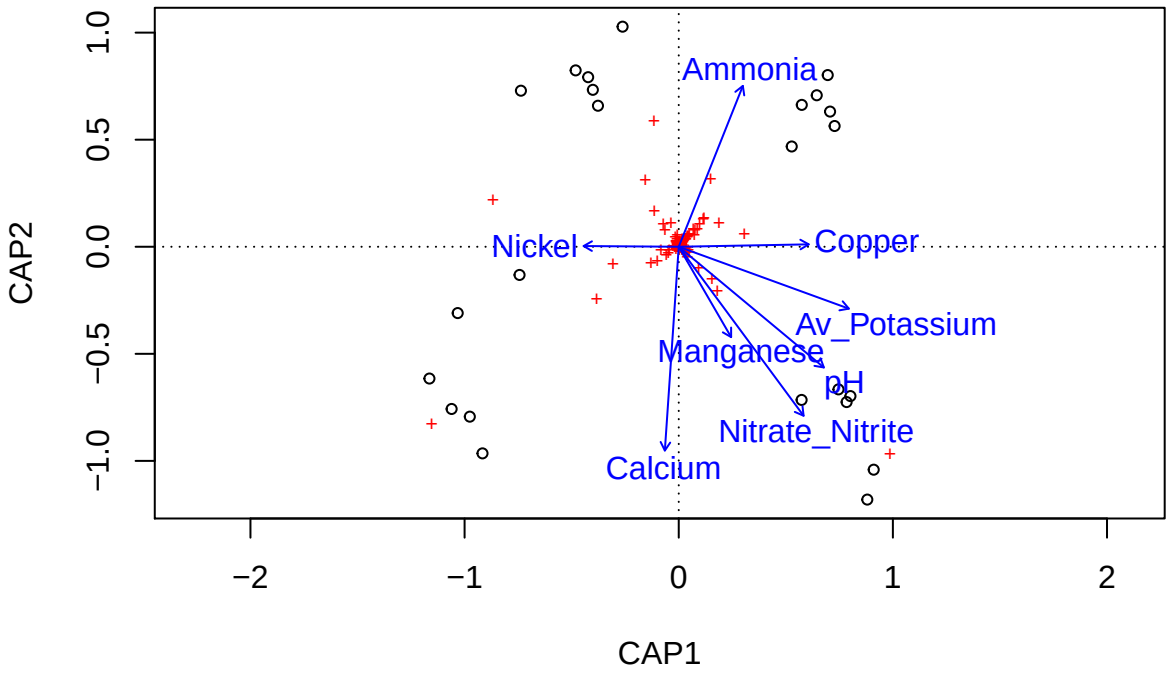



```
vif.cca(m1_nifH_irr_cap_chem_9)
```

```
## Moisture_Content  Nitrate_Nitrite          Ammonia     Av_Potassium
##        14.912289        10.108808         4.493088         8.850649
##                pH          Calcium          Copper        Manganese
##         6.663273         8.426718        10.974760         8.584170
##           Nickel
##         5.116304
```

Droping moisture

```
m1_nifH_irr_cap_chem_10 <- capscale(abund_table_irr_nifH ~ Nitrate_Nitrite + Ammonia + Av_Potassium +
    pH + Calcium + Copper + Manganese + Nickel, data = meta3, distance = "bray")
m0_nifH_irr_cap_chem <- capscale(abund_table_irr_nifH ~ 1, meta3)
plot(m1_nifH_irr_cap_chem_10)
```




```
vif.cca(m1_nifH_irr_cap_chem_10)
```

```
## Nitrate_Nitrite         Ammonia    Av_Potassium              pH
##        9.087620        4.389155        7.856473        5.567799
##         Calcium          Copper       Manganese          Nickel
##        6.908722        8.764451        7.262445        4.729897
```

```
model_cap_chem_nifH <- ordiR2step(m0_nifH_irr_cap_chem, scope = formula(m1_nifH_irr_cap_chem_10))
```

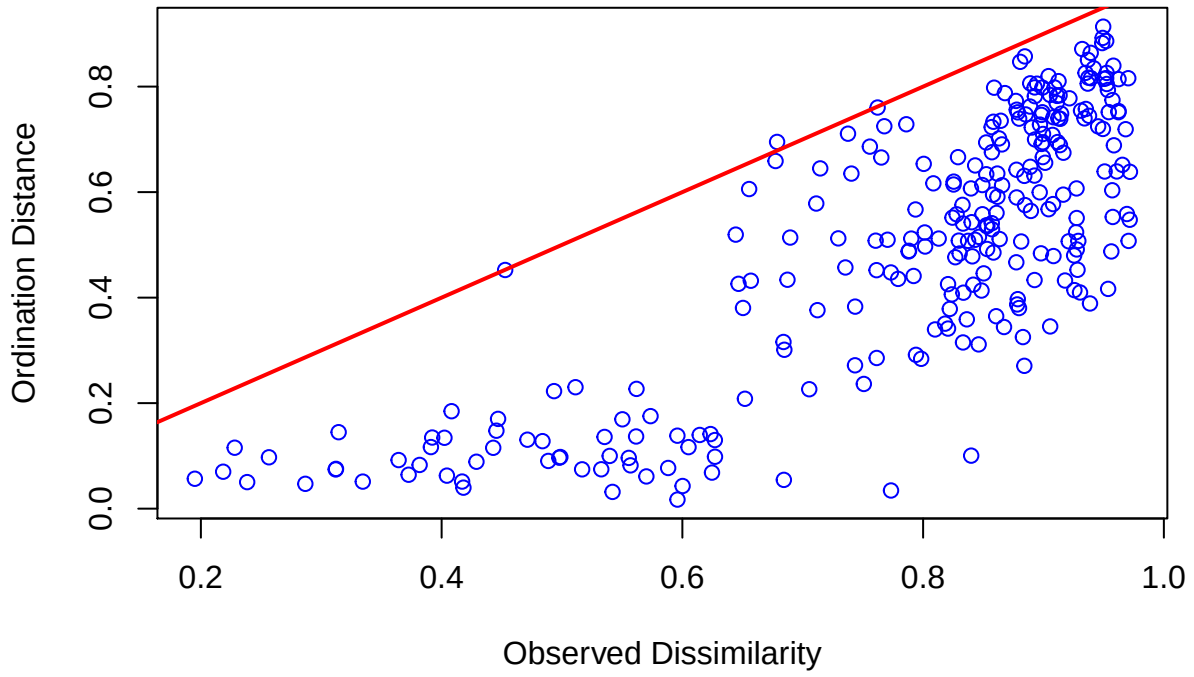
```
aov_model_cap_chem_nifH <- model_cap_chem_nifH$anova
aov_model_cap_chem_nifH
```

	R2.adj	Df	AIC	F	Pr(>F)
+ Av_Potassium	0.2234476	1	609.0666	7.618090	0.002
+ Nitrate_Nitrite	0.3919542	1	604.0794	7.096818	0.002
+ Copper	0.4712110	1	601.5565	4.147559	0.002
+ Calcium	0.5387800	1	599.0443	3.930009	0.002
	0.5497256	NA	NA	NA	NA

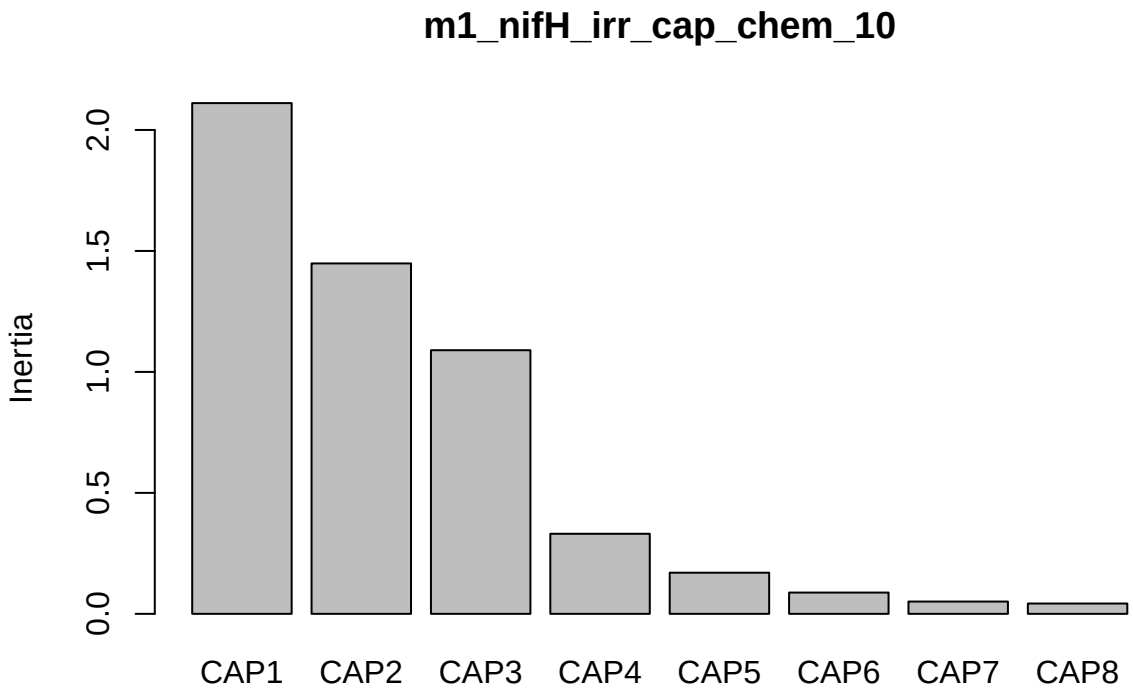
```
capture.output(aov_model_cap_chem_nifH, file = "aov_model_cap_chem_nifH_irr.txt")
```

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

```
stressplot(m1_nifH_irr_cap_chem_10)
```

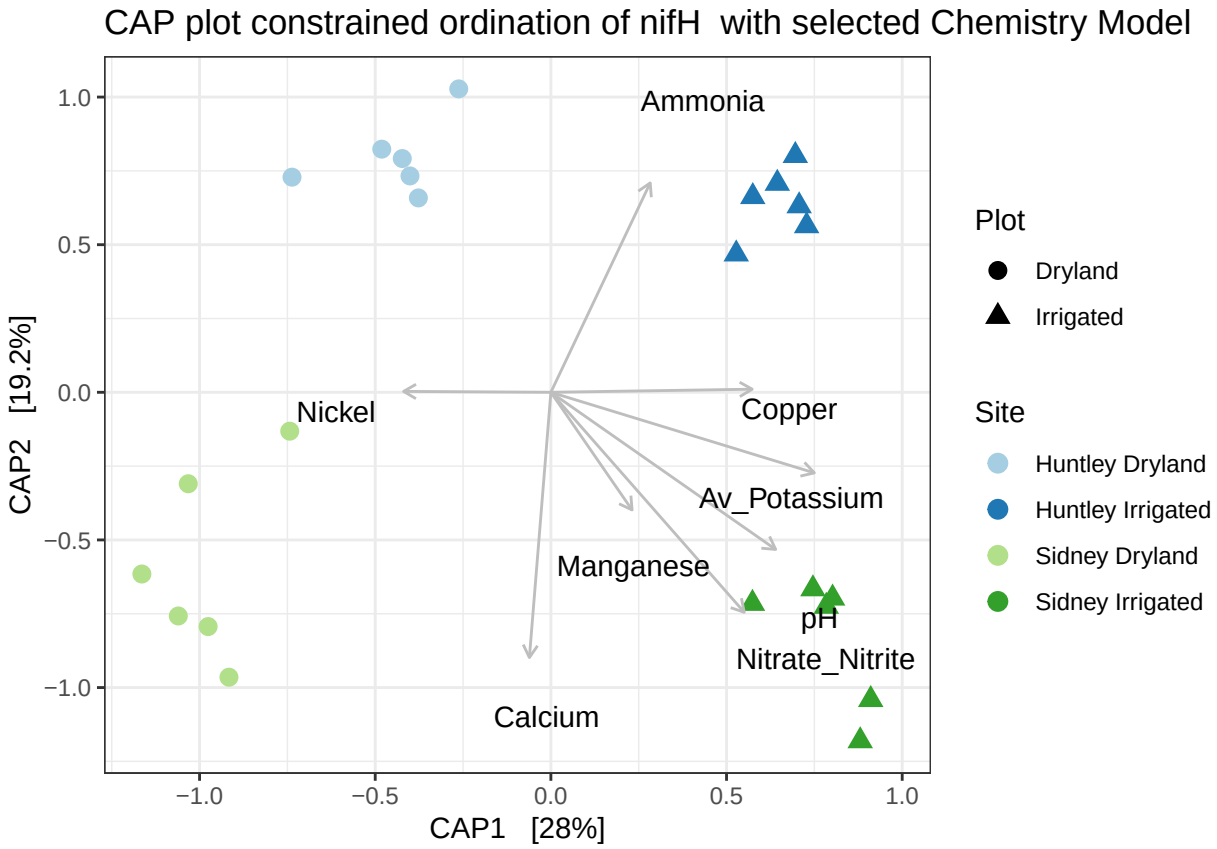



```
screeplot(m1_nifH_irr_cap_chem_10)
```




```
# CAP ordinate
cap_ord_nifH_irr <- ordinate(physeq = physeq_nifH_irr_ord, method = "CAP", distance = abund_dist_nifH_i
    formula = ~Nitrate_Nitrite + Ammonia + Av_Potassium + pH + Calcium + Copper + Manganese + Nickel)

# CCA plot
cap_plot_nifH_irr <- plot_ordination(physeq = physeq_nifH_irr_ord, ordination = cap_ord_nifH_irr, color
    shape = "Plot", axes = c(1, 2)) + geom_point(aes(colour = Site), size = 3) + scale_color_manual(valu
```

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

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

# Multiply biplot by scaling multiplier
arrowmat_nifH_irr_cap_scale <- arrowmat_nifH_irr_cap * mul
# Add labels, make a data.frame
arrowdf_nifH_irr_cap <- data.frame(labels = rownames(arrowmat_nifH_irr_cap_scale), arrowmat_nifH_irr_ca

# 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_nifH_irr + geom_segment(mapping = arrow_map, size = 0.5, data = arrowdf_nifH_irr_cap, color =
    arrow = arrowhead) + geom_text_repel(mapping = label_map, size = 4, data = arrowdf_nifH_irr_cap,
    show.legend = FALSE) + ggtitle("CAP plot constrained ordination of nifH  with selected Chemistry Mo
    theme_bw()
```



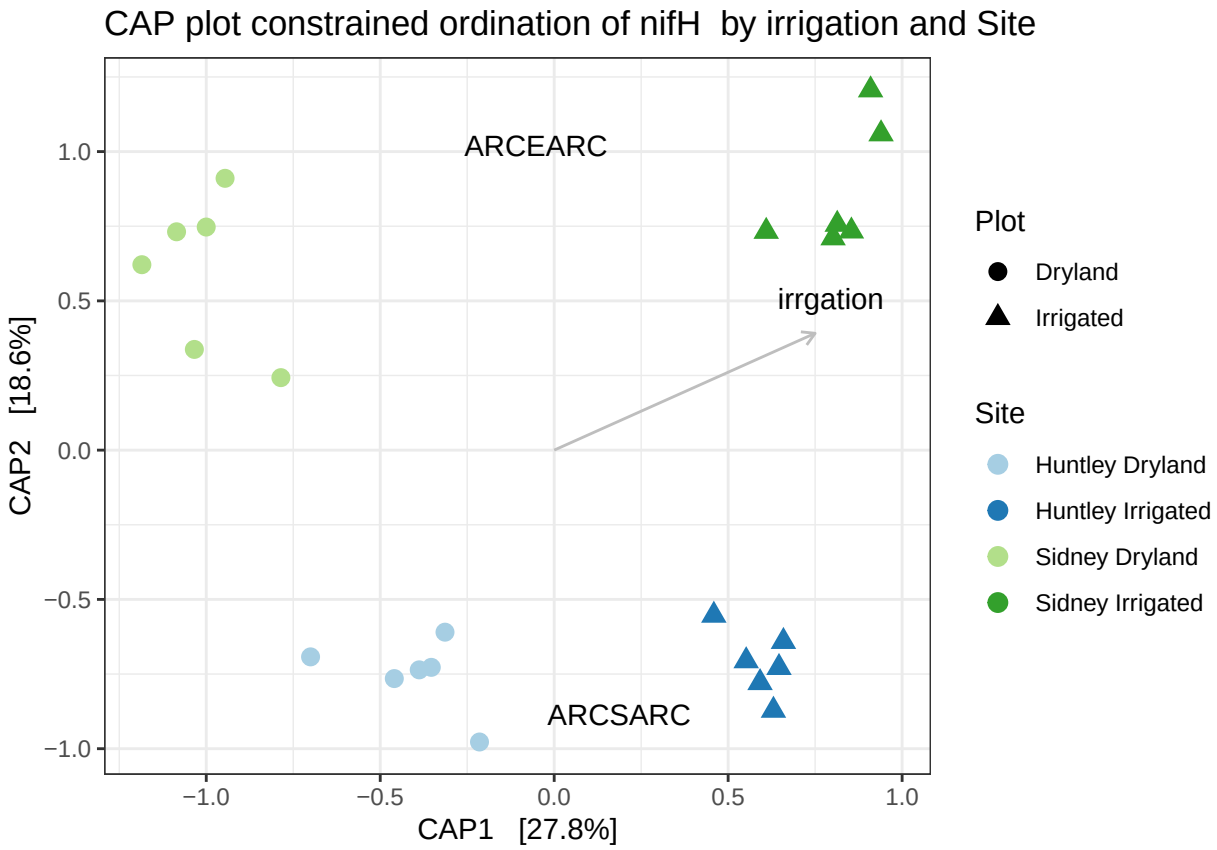

```
## Warning: Ignoring unknown aesthetics: label
```

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

Irrgation vs Site

```
# CAP ordinate
cap_ord_nifH_irr_site <- ordinate(physeq = physeq_nifH_irr_ord, method = "CAP", distance = abund_dist_ni
    formula = ~irrgation + ARC)

# CCA plot
cap_plot_nifH_irr_site <- plot_ordination(physeq = physeq_nifH_irr_ord, ordination = cap_ord_nifH_irr_si
    color = "Site", shape = "Plot", axes = c(1, 2)) + geom_point(aes(colour = Site), size = 3) + scale_

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


elipse_nifH_irr_site_cap <- vegan::scores(cap_ord_nifH_irr_site, display = "cn")

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

# Multiply biplot by scaling multiplier
arrowmat_nifH_irr_site_cap_scale <- arrowmat_nifH_irr_site_cap * mul
# Add labels, make a data.frame
arrowdf_nifH_irr_site_cap <- data.frame(labels = rownames(arrowmat_nifH_irr_site_cap_scale), arrowmat_ni

# remove ARC variable
arrowdf_nifH_irr_site_cap <- arrowdf_nifH_irr_site_cap[-2, ]

centdf_nifH_irr_site_cap <- data.frame(labels = rownames(elipse_nifH_irr_site_cap), elipse_nifH_irr_site


# 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_nifH_irr_site + geom_segment(mapping = arrow_map, size = 0.5, data = arrowdf_nifH_irr_site_cap
    color = "gray", arrow = arrowhead) + geom_text_repel(mapping = label_map, size = 4, data = arrowdf_
    show.legend = FALSE) + geom_text_repel(mapping = label_map, size = 4, data = centdf_nifH_irr_site_ca
    show.legend = FALSE) +
ggtitle("CAP plot constrained ordination of nifH  by irrigation and Site") + theme_bw()
```

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




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

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

Fitting species to cap plot

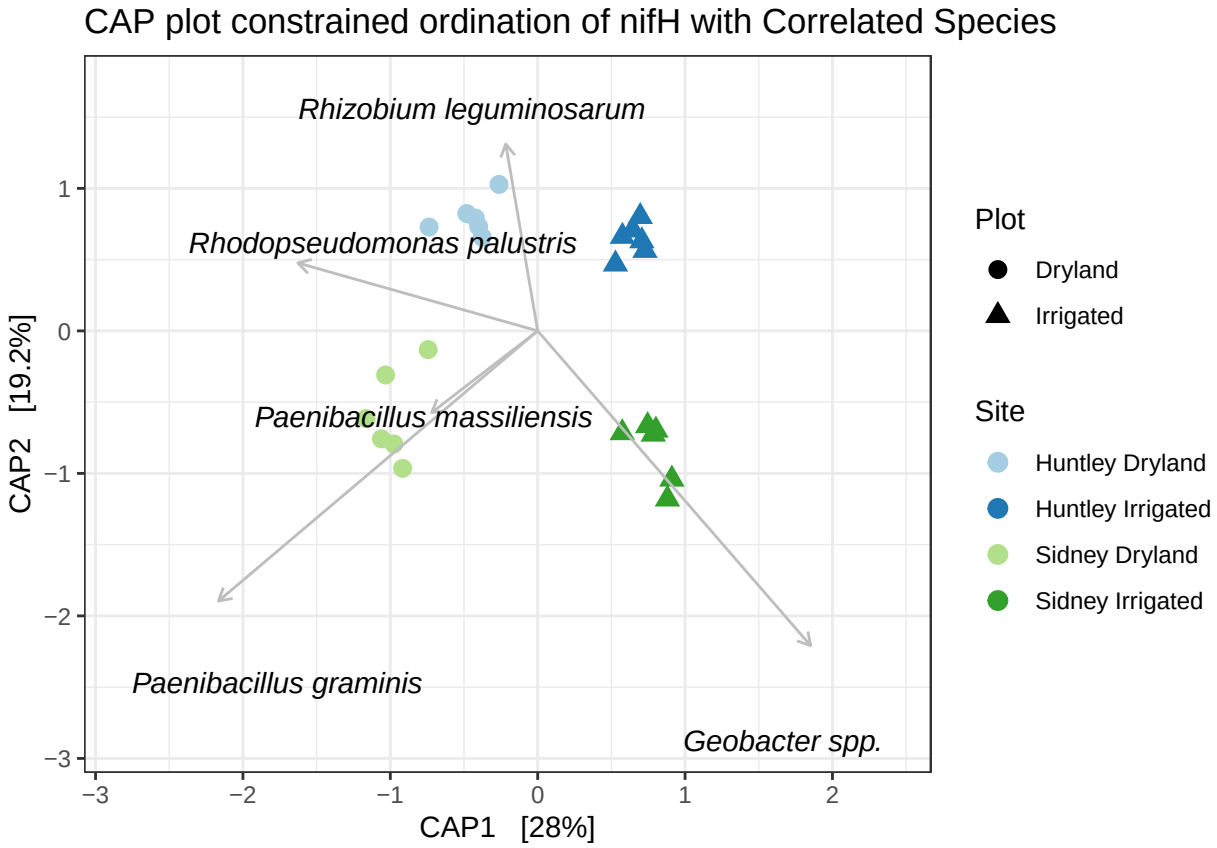
```
m1_nifH_irr_cap_chem_species <- capscale(abund_table_irr_nifH ~ Nitrate_Nitrite + Ammonia + Av_Potassiu
    pH + Calcium + Copper + Manganese + Nickel, data = meta3, distance = "bray")
dims = c(1, 2)
site = scores(m1_nifH_irr_cap_chem_species, display = "wa", choices = dims)
cor.min = 0.9  #below this threshold, arrows will be not plotted
# because orrelation is considered too much week
cor_sp = as.data.frame(scores(m1_nifH_irr_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
```

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

```
tax_nifH_cap$species <- capitalize(tax_nifH_cap$species)

species_nifH_irr_cap <- merge(cor_sp_s1, tax_nifH_cap, by = "row.names")
```

Replot with species

```
# CAP ordinate
cap_ord_nifH_irr <- ordinate(physeq = physeq_nifH_irr_ord, method = "CAP", distance = abund_dist_nifH_i
    formula = ~Nitrate_Nitrite + Ammonia + Av_Potassium + pH + Calcium + Copper + Manganese + Nickel)

# CCA plot
cap_plot_nifH_irr <- plot_ordination(physeq = physeq_nifH_irr_ord, ordination = cap_ord_nifH_irr, color
    shape = "Plot", axes = c(1, 2)) + geom_point(aes(colour = Site), size = 3) + scale_color_manual(valu



# Define the arrow aesthetic mapping
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_nifH_irr + geom_segment(mapping = arrow_map, size = 0.5, data = species_nifH_irr_cap, color =
    arrow = arrowhead) + geom_text_repel(mapping = label_map, size = 4, data = species_nifH_irr_cap,
    show.legend = FALSE, fontface = "italic") + ggtitle("CAP plot constrained ordination of nifH with C
    theme_bw()
```




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

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