Diversity Project Abundances

Joshua

4/1/2021

```
rm(list=ls())
package.list <- c('vegan', 'data.table', 'reshape2', 'ggplot2', 'ape')
for (package in package.list){
  if (!require(package, character.only = TRUE, quietly = TRUE)) {
    install.packages(package)
    library(package,character.only = TRUE)
  }
}
```

```
## Warning: package 'vegan' was built under R version 4.0.4
```

```
## Warning: package 'permute' was built under R version 4.0.4
```

```
## This is vegan 2.5-7
```

```
## Warning: package 'data.table' was built under R version 4.0.5
```

```
## Warning: package 'reshape2' was built under R version 4.0.5
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
## Warning: package 'ape' was built under R version 4.0.5
```

Importing site-species data into R

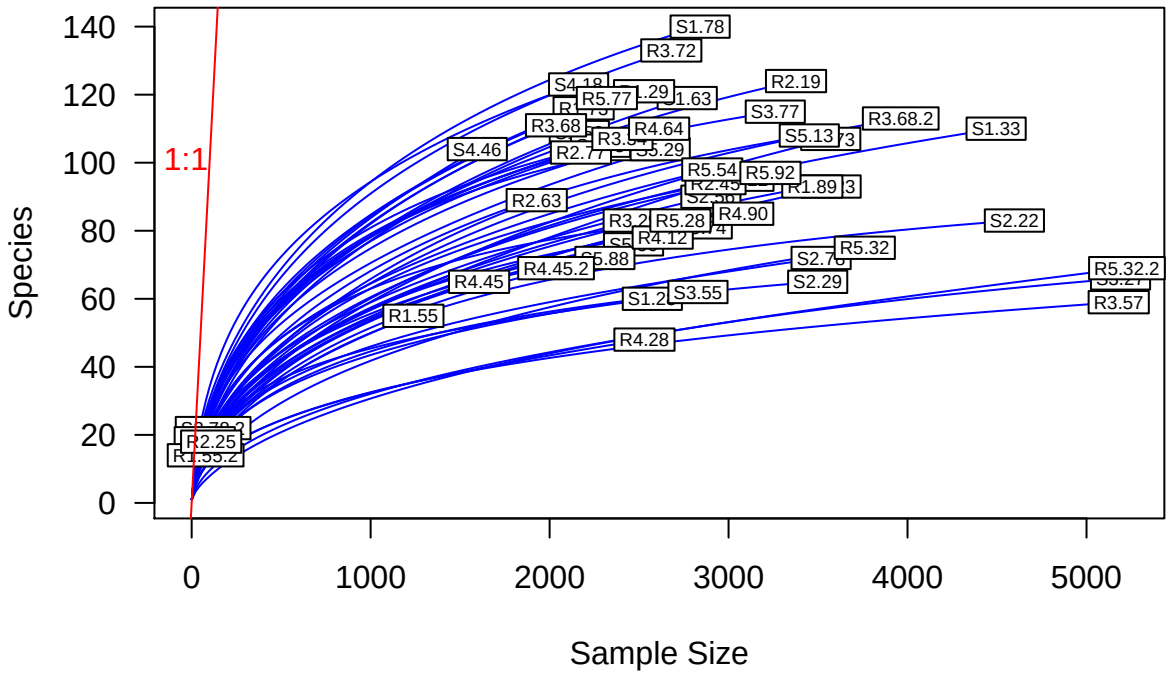
```
site_species <- read.csv("alpine_ridge_data/OTU_table.csv", header = TRUE)
site_species.t <- t(site_species)
```

#Rarefaction

```r
#Visualization of rarecurves for all samples
richness <- rowSums((site_species.t > 0) * 1)
print(richness)
```

```
##   S1.28   S1.33   S1.56   S1.63   S1.78   S2.22   S2.29   S2.56   S2.73   S2.78
##      60     110     109     119     140      83      65      90     105      72
## S2.78.2   S3.23   S3.27   S3.55   S3.73   S3.77   S4.18   S4.22   S4.46   S4.68
##      22      93      66      62     107     115     123      95     104     104
##   S4.73   S5.13   S5.29   S5.56   S5.74   S5.88   R1.14   R1.29   R1.55 R1.55.2
##     105     108     104      76      81      72      19     121      55      14
##   R1.73   R1.89   R2.19   R2.25   R2.45   R2.63   R2.77   R3.29   R3.34   R3.57
##     116      93     124      18      94      89     103      83     107      59
##   R3.68 R3.68.2   R3.72   R4.12   R4.28   R4.45 R4.45.2   R4.64   R4.90   R5.28
##     111     113     133      78      48      65      69     110      85      83
##   R5.32 R5.32.2   R5.54   R5.77   R5.92
##      75      69      98     119      97
```

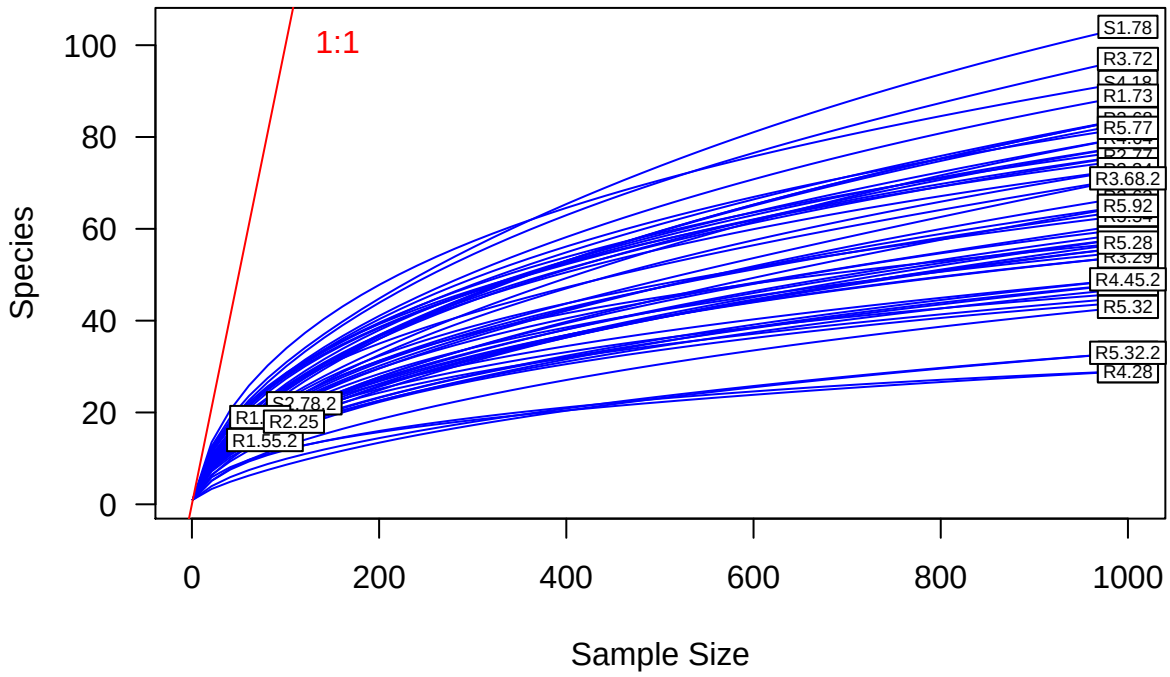
```r
minimum.r <- min(rowSums(site_species.t))
rarefy <- rarefy(x = site_species.t, sample = minimum.r, se = TRUE)

rarecurve(x = site_species.t, step = 20, col = "blue", cex = .6, las = 1)
abline(0, 1, col = 'red')
text(200, 100, "1:1", pos = 2, col = 'red')
```



```
#Rarefaction of samples
site_species.r <- rrarefy(site_species.t, 1000)
```

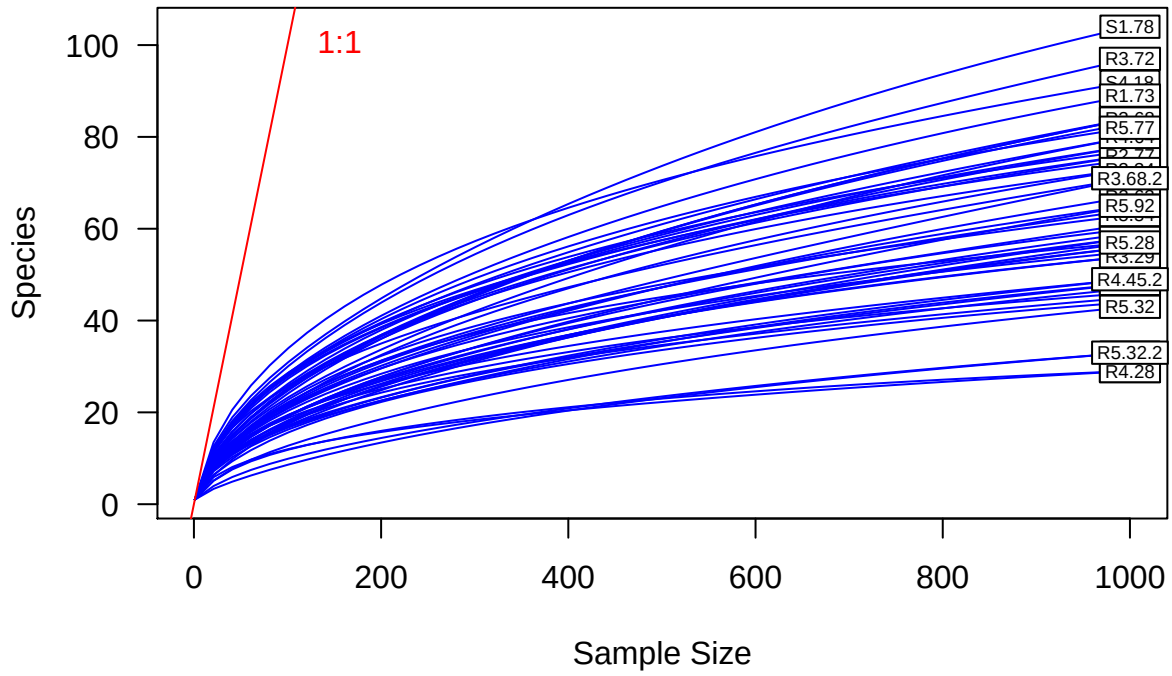
```
## Warning in rrarefy(site_species.t, 1000): some row sums < 'sample' and are not
## rarefied
```

```
richness <- rowSums((site_species.r > 0) * 1)
minimum.r <- min(rowSums(site_species.r))
rarefy <- rarefy(x = site_species.r, sample = minimum.r, se = TRUE)

rarecurve(x = site_species.r, step = 20, col = "blue", cex = .6, las = 1)
abline(0, 1, col = 'red')
text(200, 100, "1:1", pos = 2, col = 'red')
```




```
#Remove samples containing less than 1000 reads (R1.14, R1.55.2, R2.25, S2.78.2)

df.site_species.r <- as.data.frame(site_species.r)
rarefied_site_species <- data.frame()

for (i in 1:nrow(df.site_species.r)){
  if (rowSums(df.site_species.r[i,]) >= 1000){
    rarefied_site_species <- rbind(rarefied_site_species, df.site_species.r[i,])
  }
}
```

```
#Visualizing
richness <- rowSums((rarefied_site_species > 0) * 1)
minimum.r <- min(rowSums(rarefied_site_species))
rarefy <- rarefy(x = rarefied_site_species, sample = minimum.r, se = TRUE)

rarecurve(x = rarefied_site_species, step = 20, col = "blue", cex = .6, las = 1)
abline(0, 1, col = 'red')
text(200, 100, "1:1", pos = 2, col = 'red')
```



#Removing samples to match environmental data downstream

```
rarefied_site_species <- rarefied_site_species[-c(38,43,48),]
```

Importing Environmental Data

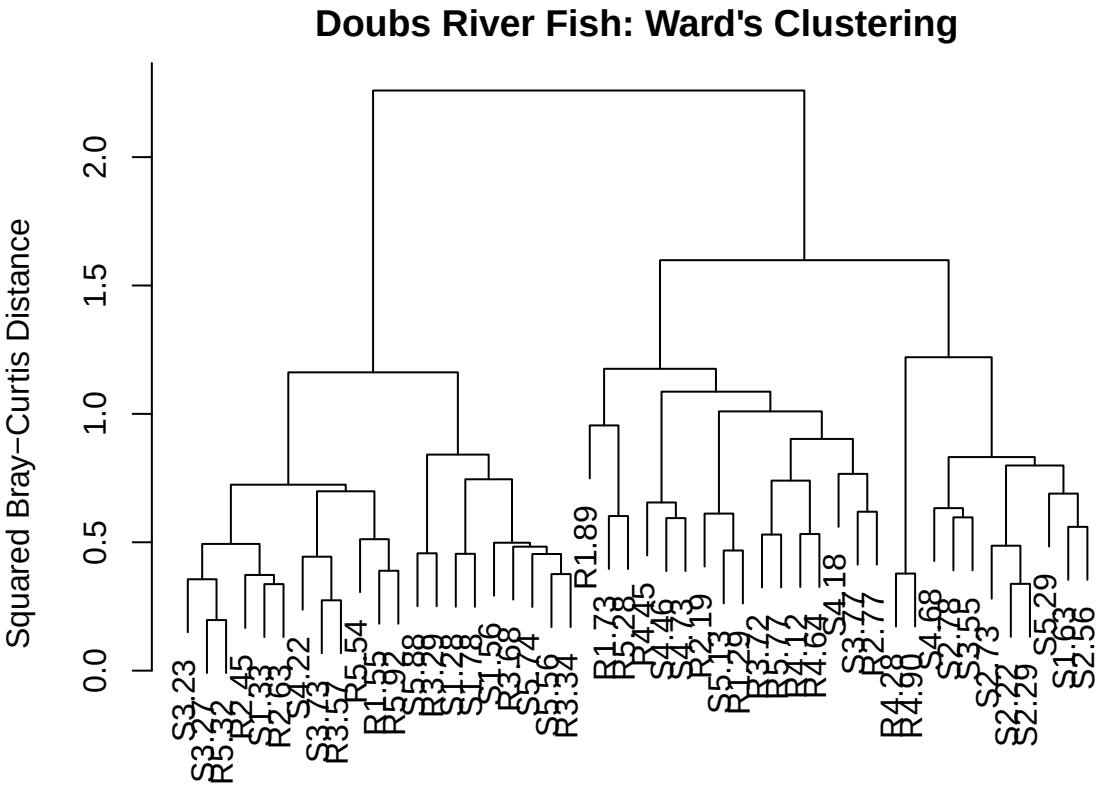
```
env <- read.csv("alpine_ridge_data/variables.txt", header = TRUE, sep = "\t")
env <- env[-c(26,31),]
```

Calculating Bray-Curtis Beta-Diversity

```
fungalBC <- vegdist(rarefied_site_species, method = "bray")
```

Cluster Analysis of Fungal Communities

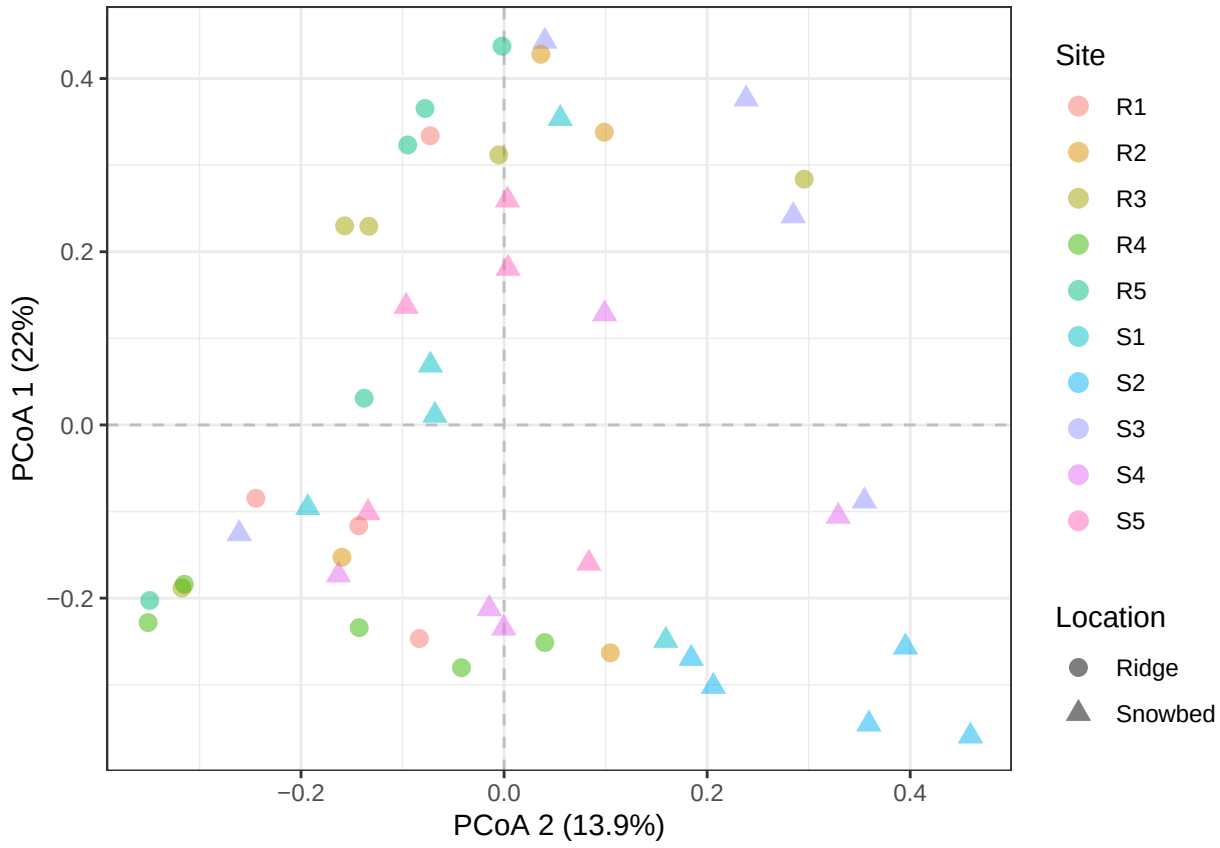
```
#Performing Cluster Analysis
fungal.ward <- hclust(fungalBC, method = "ward.D2")

#Plotting Cluster
par(mar = c(1,5,2,2) + .1)
plot(fungal.ward, main = "Doubs River Fish: Ward's Clustering",
     ylab = "Squared Bray-Curtis Distance")
```



PCoA of Fungal Communities

```
fungal.pcoa <- cmdscale(fungalBC, eig = TRUE, k = 3)
```

```r
explainvar1 <- round(fungal.pcoa$eig[1]/sum(fungal.pcoa$eig), 3) * 100
explainvar2 <- round(fungal.pcoa$eig[2]/sum(fungal.pcoa$eig), 3) * 100
explainvar3 <- round(fungal.pcoa$eig[3]/sum(fungal.pcoa$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

test <- rarefied_site_species

test$Site <- c(rep("S1", 5), rep("S2", 5), rep("S3", 5), rep("S4", 5), rep("S5", 5), rep("R1", 4), rep(

test$Location <- c(rep("Snowbed", 25), rep("Ridge", 23))

## Plotting
# make a fake pcoa
dist      <- vegdist(test[,1:807],  method = "bray")
fake_pcoa <- pcoa(dist)
#summary(fake_pcoa)

# get the out of pcoa
# site scores = samples
sitescores   <- fake_pcoa$vectors #we need only the first two axis
newdata <- as.data.frame(cbind(sitescores[,1:2], Location = test$Location, Site = test$Site))

#one way
bray <- ggplot(newdata, aes(y = as.numeric(Axis.1), x = as.numeric(Axis.2), shape = Location, color = S
bray <- bray + geom_hline(yintercept = 0, color = "grey", linetype = "dashed")
bray <- bray + geom_vline(xintercept = 0, color = "grey", linetype = "dashed")
bray <- bray + geom_point(size = 3, alpha = .5)
bray <- bray + theme_bw()
bray <- bray + labs(y = paste("PCoA 1 (", explainvar1, "%)", sep = ""), x = paste("PCoA 2 (", explainva
bray
```



How much variance is explained by site location (Bray-Curtis)

```
site <- c(rep("S1", 5), rep("S2", 5), rep("S3", 5), rep("S4", 5), rep("S5", 5), rep("R1", 4), rep("R2",

adonis(rarefied_site_species ~ env$V + site, permutations = 999)
```

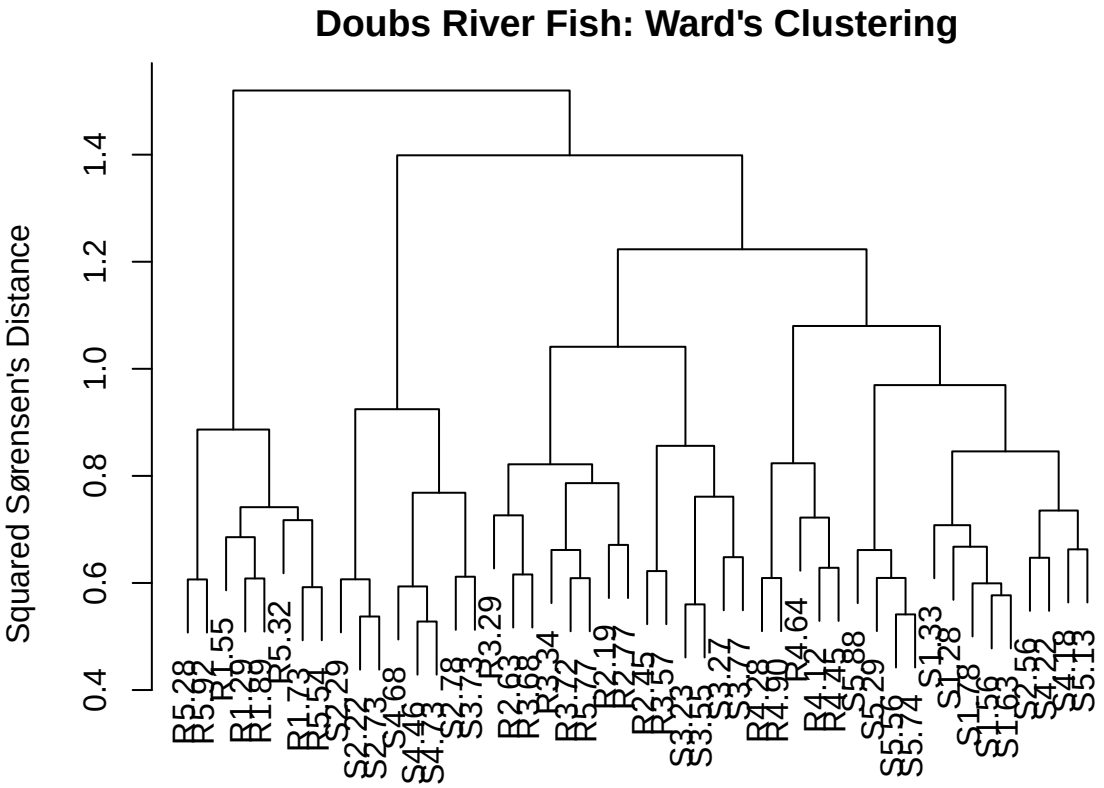
```
##
## Call:
## adonis(formula = rarefied_site_species ~ env$V + site, permutations = 999)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## env$V      1    0.9254 0.92539  3.9321 0.06642  0.001 ***
## site       8    4.0643 0.50804  2.1587 0.29171  0.001 ***
## Residuals 38    8.9430 0.23534         0.64187
## Total     47   13.9327                1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Calculating Sorensen Beta-Diversity

```
fungalS <- vegdist(rarefied_site_species, method = "bray", binary = "TRUE")
```

Cluster Analysis of Fungal Communities

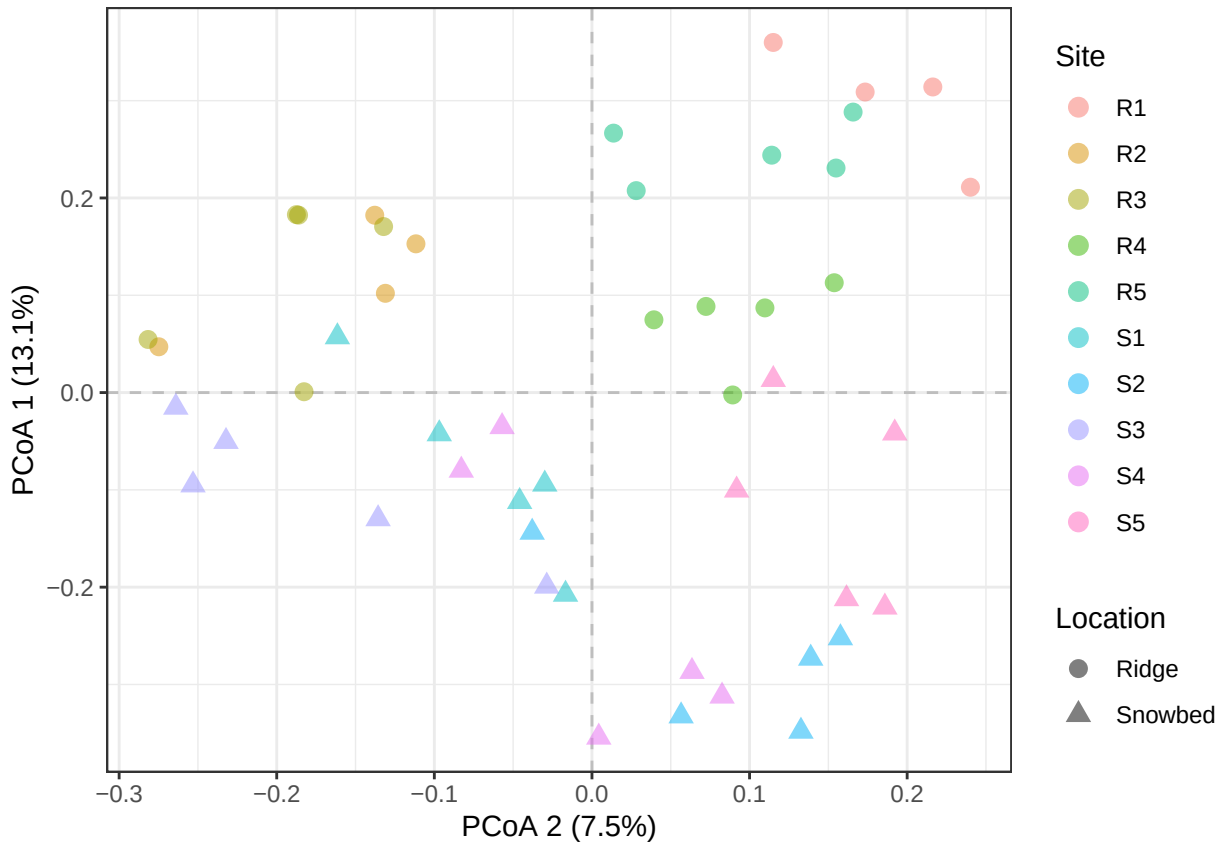
```
#Performing Cluster Analysis
fungal.wardS <- hclust(fungalS, method = "ward.D2")

#Plotting Cluster
par(mar = c(1,5,2,2) + .1)
plot(fungal.wardS, main = "Doubs River Fish: Ward's Clustering",
     ylab = "Squared Sørensen's Distance")
```



PCoA of Fungal Communities

```
fungal.S.pcoa <- cmdscale(fungalS, eig = TRUE, k = 3)
```

```r
explainvar1 <- round(fungal.S.pcoa$eig[1]/sum(fungal.pcoa$eig), 3) * 100
explainvar2 <- round(fungal.S.pcoa$eig[2]/sum(fungal.pcoa$eig), 3) * 100
explainvar3 <- round(fungal.S.pcoa$eig[3]/sum(fungal.pcoa$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

## Plotting
# make a fake pcoa
dist      <- vegdist(test[,1:807],  method = "bray", binary = TRUE)
fake_pcoa <- pcoa(dist)
#summary(fake_pcoa)

# get the out of pcoa
# site scores = samples
sitescores   <- fake_pcoa$vectors #we need only the first two axis
newdata <- as.data.frame(cbind(sitescores[,1:2], Location = test$Location, Site = test$Site))

#one way
soren <- ggplot(newdata, aes(y = as.numeric(Axis.1), x = as.numeric(Axis.2), shape = Location, color = S
soren <- soren + geom_hline(yintercept = 0, color = "grey", linetype = "dashed")
soren <- soren + geom_vline(xintercept = 0, color = "grey", linetype = "dashed")
soren <- soren + geom_point(size = 3, alpha = .5)
soren <- soren + theme_bw()
soren <- soren + labs(y = paste("PCoA 1 (", explainvar1, "%)", sep = ""), x = paste("PCoA 2 (", explainv
soren
```



How much variance is explained by site location (Sorensen)

```
site <- c(rep("S1", 5), rep("S2", 5), rep("S3", 5), rep("S4", 5), rep("S5", 5), rep("R1", 4), rep("R2",

adonis(rarefied_site_species ~ env$V + site, binary = TRUE, permutations = 999)
```

```
##
## Call:
## adonis(formula = rarefied_site_species ~ env$V + site, permutations = 999,      binary = TRUE)
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## env$V      1    1.3766 1.37664  5.9565 0.09770  0.001 ***
## site       8    3.9322 0.49152  2.1267 0.27905  0.001 ***
## Residuals 38    8.7824 0.23111         0.62325
## Total     47   14.0912                1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

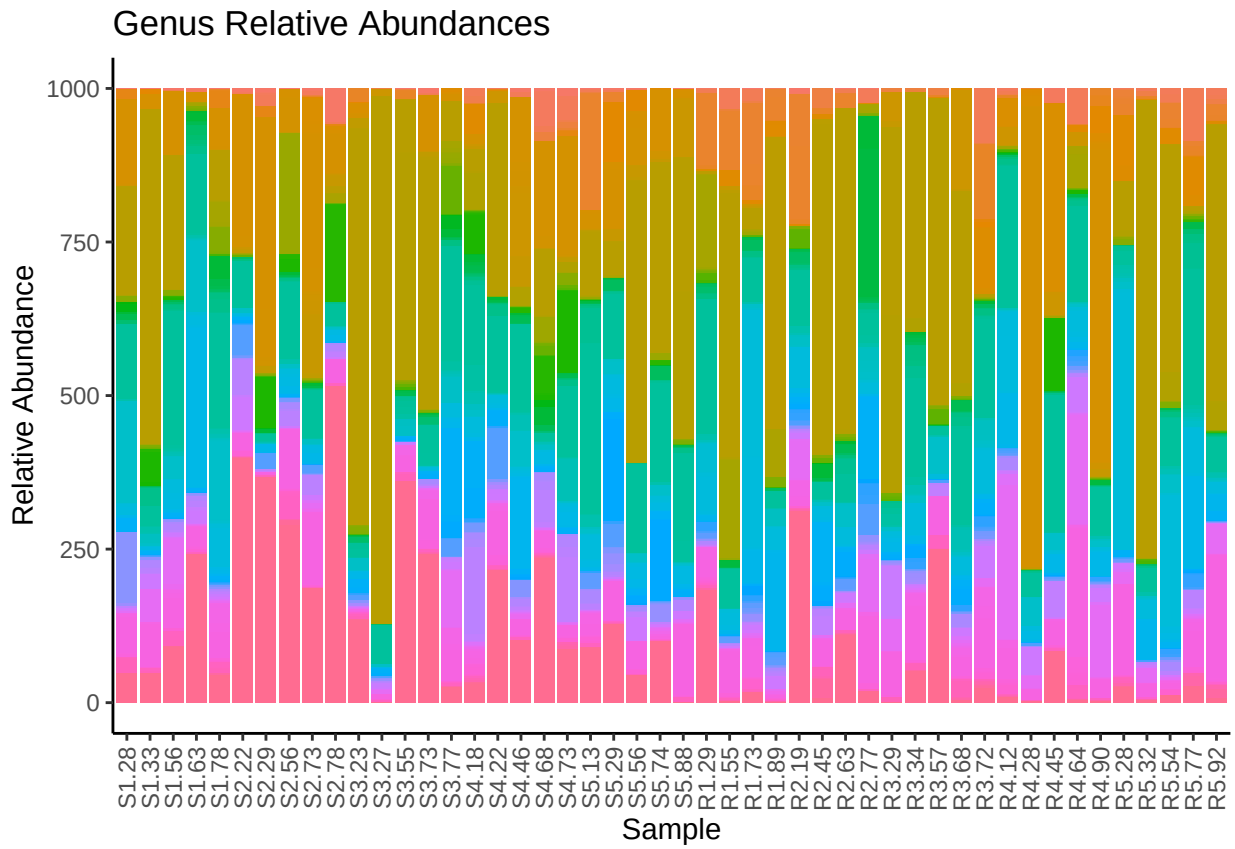
Relative Abundance Visualization

```
genus <- read.csv("alpine_ridge_data/genus_table.csv", header = TRUE)
#convert rownames into column

genus.1 <- as.data.frame(t(rarefied_site_species))
setDT(genus.1, keep.rownames = TRUE)[]
```

```
##        rn S1.28 S1.33 S1.56 S1.63 S1.78 S2.22 S2.29 S2.56 S2.73 S2.78 S3.23
##   1:   V1     0     0     0     1     0     0     0     0     0     0     0
##   2:   V2     0     0     0     0     0     0     0     0     0     0     0
##   3:   V3     0     0     0     0     0     0     0     0     0     0     0
##   4:   V4     0     0     0     0     0     0     0     4     0     0     0
##   5:   V5     0     0     0     0     0     0     0     0     0     0     0
##  ---
## 803: V803     0     0     0     0     2     0     1     2     0     0     1
## 804: V804     0     0     0     0     0     0     0     0     1     0     0
## 805: V805     0     0     0     0     0     0     0     0     0     0     0
## 806: V806     0     0     0     0     0     0     0     0     0     0     0
## 807: V807     0     0     0     0     0     0     0     0     0     0     0
##      S3.27 S3.55 S3.73 S3.77 S4.18 S4.22 S4.46 S4.68 S4.73 S5.13 S5.29 S5.56
##   1:     0     0     0     0     0     0     0     0     1     0     0     0
##   2:     0     0     0     0     6     0    32     0     0     0    17    17
##   3:     0     0     0     0     0     0     0     0     0     0     0     0
##   4:     0     0     0     0     0     0     0     0     0     0     0     0
##   5:     0     0     0     0     0     0     0     0     0     0     0     0
##  ---
```

```
## 803:     0     0     0     1     9     0     0     4     0     0     0     0
## 804:     0     1     0     0     0     0     0     0     0     0     0     0
## 805:     0     0     0     0     0     0     0     0     0     0     0     0
## 806:     0     0     0     0     0     0     0     0     0     0     0     0
## 807:     0     0     0     0     0     0     0     0     0     0     0     0
##      S5.74 S5.88 R1.29 R1.55 R1.73 R1.89 R2.19 R2.45 R2.63 R2.77 R3.29 R3.34
##   1:     0     0     0     0     0     0     0     0     0     0     0     0
##   2:     5     1     0     0     0     0     0     0     0     0     0     0
##   3:     0     0     0     0     0     0     0     0     0     0     0     0
##   4:     0     0     0     0     0     0     0     0     0     0     0     0
##   5:     0     0     0     0     0     0     0     0     1     1     0     0
##  ---
## 803:     0     0     0     0     0     0     0     4     0     0     0     1
## 804:     2     0     5     1     2     0     0     0     0     0     0     0
## 805:     0     0     0     0     0     0     0     1     0     0     0     0
## 806:     0     0     0     0     0     0     0     0     0     0     0     0
## 807:     0     0     0     0     0     0     0     0     0     0     0     0
##      R3.57 R3.68 R3.72 R4.12 R4.28 R4.45 R4.64 R4.90 R5.28 R5.32 R5.54 R5.77
##   1:     0     0     0     0     0     0     0     0     0     0     0     0
##   2:     0     0     0     0     0     0     0     0     0     0     0     0
##   3:     0     0     0     1     0     0     0     0     0     0     0     0
##   4:     0     0     0     0     0     0     0     0     0     0     0     0
##   5:     0     0     0     0     0     0     0     0     0     0     0     0
##  ---
## 803:     0     0     1     0     0     0     2     0     0     3     0     0
## 804:     0     0     0     0     2     0     0     2    12     0    11    26
## 805:     0     0     0     0     0     0     0     0     0     0     0     0
## 806:     0     0     0     0     0     0     0     0     0     0     0     0
## 807:     0     0     0     0     0     0     0     0     0     0     0     0
##      R5.92
##   1:     0
##   2:     0
##   3:     0
##   4:     0
##   5:     0
##  ---
## 803:     0
## 804:     3
## 805:     0
## 806:     0
## 807:     0
```

```
# Converting to Long Format
genus_long <- melt(genus.1, id.vars = "rn", variable.name = "Sample")

# Creating Graph of data
genus_graph <- ggplot(data = genus_long, mapping = aes(x = Sample,  y = value, fill = rn))
genus_graph <- genus_graph + geom_bar(stat="identity")
genus_graph <- genus_graph + labs(y = "Relative Abundance", x = "Sample", title = "Genus Relative Abunda
genus_graph <- genus_graph + theme(legend.position = "None")
genus_graph <- genus_graph + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
genus_graph
```



Importing environmental data and testing for significance

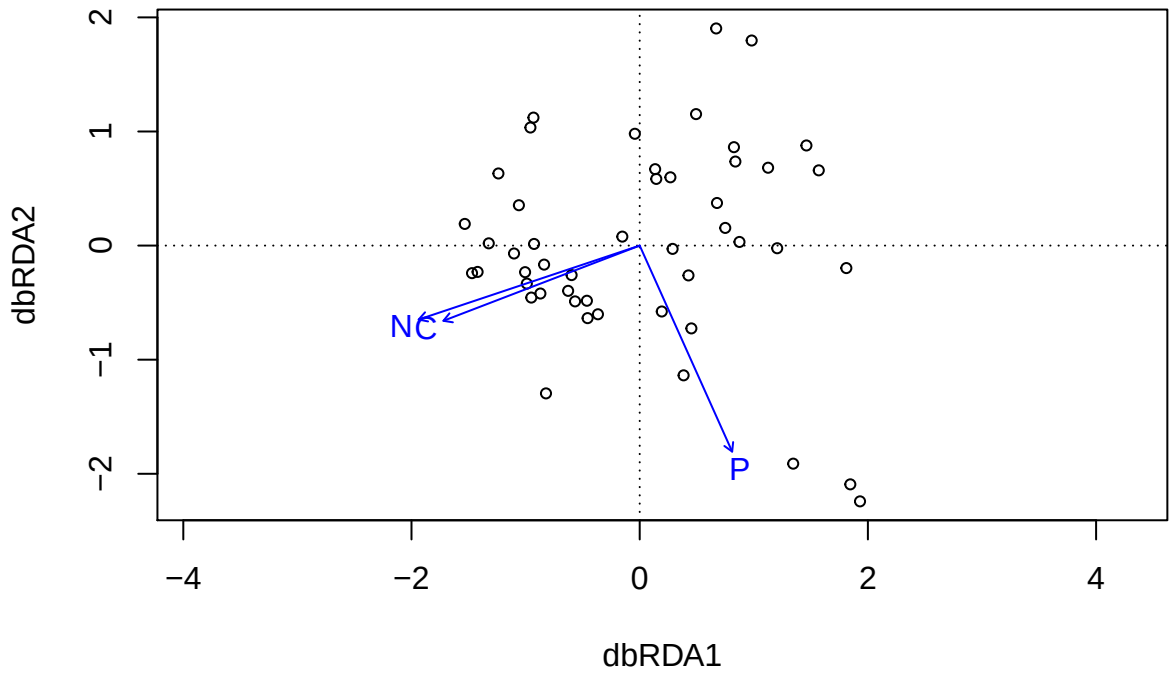
```
env <- read.csv("alpine_ridge_data/variables.txt", header = TRUE, sep = "\t")
env <- env[-c(26,31),]

site <- c(rep("S1", 5), rep("S2", 5), rep("S3", 5), rep("S4", 5), rep("S5", 5), rep("R1", 4), rep("R2",
adonis(rarefied_site_species ~ env$V + env$P + env$N + env$C + site, method = "bray", permutations = 99
```

```
##
## Call:
## adonis(formula = rarefied_site_species ~ env$V + env$P + env$N +      env$C + site, permutations = 9
##
## Permutation: free
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## env$V      1    0.9254 0.92539  3.8670 0.06642  0.001 ***
## env$P      1    0.2891 0.28906  1.2079 0.02075  0.232
## env$N      1    0.3016 0.30161  1.2604 0.02165  0.185
## env$C      1    0.3127 0.31269  1.3067 0.02244  0.213
## site       8    3.7284 0.46605  1.9475 0.26760  0.001 ***
```

```
## Residuals 35    8.3756 0.23930         0.60114
## Total     47   13.9327                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Constructing Constrained Ordination

```
env.chem <- as.matrix(env[,c(2:4)])


S.dbrda <- dbrda(fungalS ~ ., as.data.frame(env.chem))
ordiplot(S.dbrda)
```




```
S.dbrda0 <- dbrda(fungalS ~ 1, as.data.frame(env.chem))
S.dbrda1 <- dbrda(fungalS ~ ., as.data.frame(env.chem))

S.dbrda <- ordiR2step(S.dbrda0, S.dbrda1, perm.max = 999)
```

```
## Step: R2.adj= 0
## Call: fungalS ~ 1
##
##                 R2.adjusted
```

```
## <All variables> 0.038892534
## + N             0.028980574
## + C             0.022257290
## + P             0.005337757
## <none>          0.000000000
##
##     Df    AIC      F Pr(>F)
## + N  1 127.53 2.4027  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.02898057
## Call: fungalS ~ N
##
##                 R2.adjusted
## <All variables>  0.03889253
## + C              0.03457388
## + P              0.03109050
## <none>           0.02898057
##
##     Df   AIC      F Pr(>F)
## + C  1 128.2 1.2665   0.11
```

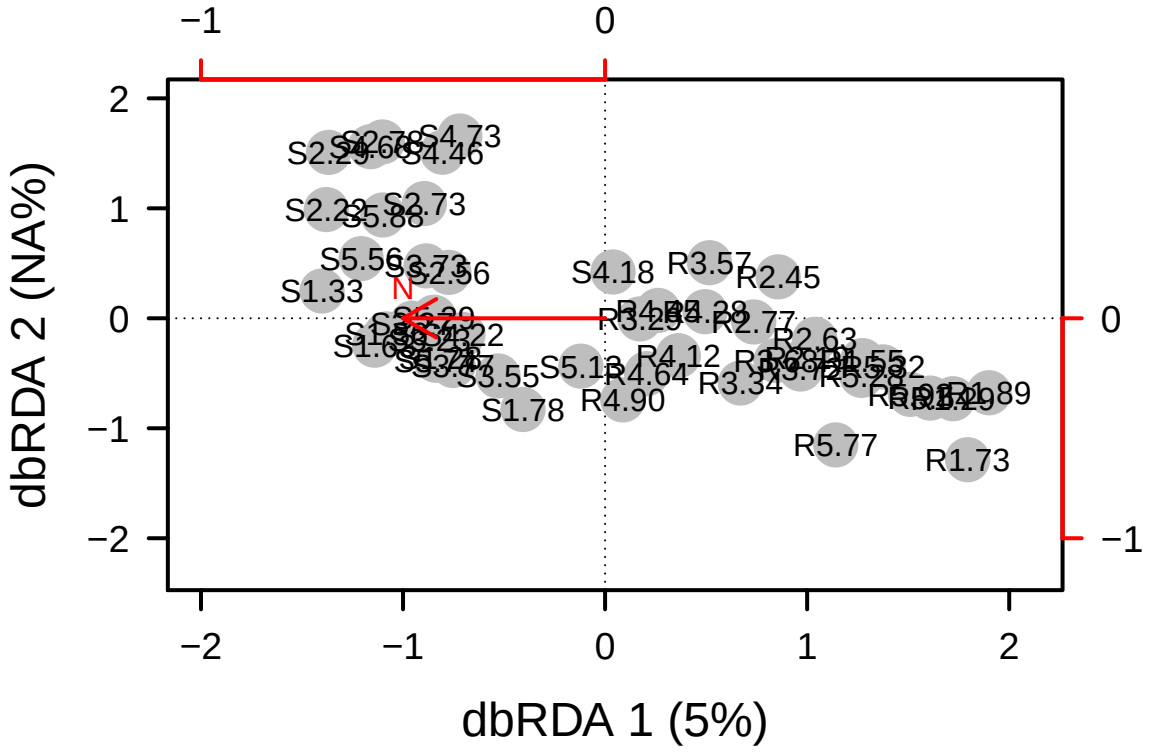
```
permutest(S.dbrda, permutations = 999)
```

```
##
## Permutation test for dbrda under reduced model
##
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = fungalS ~ N, data = as.data.frame(env.chem))
## Permutation test for all constrained eigenvalues
##          Df Inertia      F Pr(>F)
## Model     1  0.6995 2.4027  0.001 ***
## Residual 46 13.3917
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

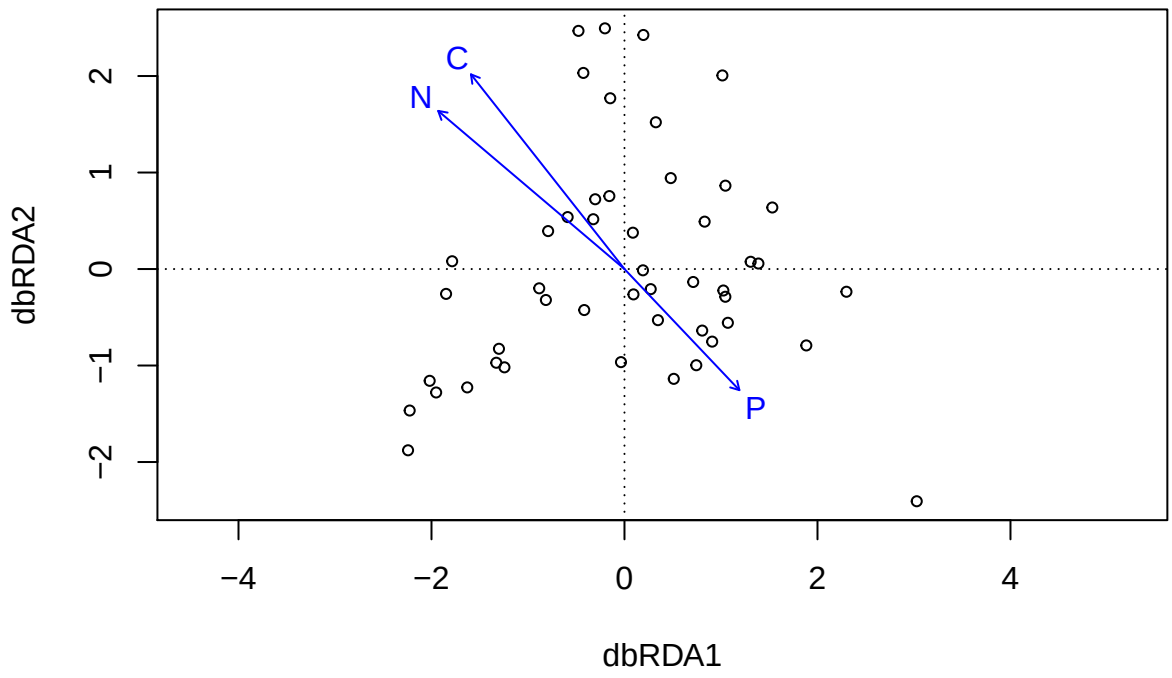
```
envfit(S.dbrda, env.chem, permutations = 999)
```

```
##
## ***VECTORS
##
##     dbRDA1     MDS1     r2 Pr(>r)
## P  0.88312  0.46914 0.0691  0.207
## N -0.75538 -0.65528 0.8570  0.001 ***
## C -0.72724 -0.68638 0.8253  0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

```r
#Calculating explained variation on axes
S.explainvar1 <- round(S.dbrda$CCA$eig[1]/
                           sum(c(S.dbrda$CCA$eig, S.dbrda$CA$eig)),
                       3 ) * 100
S.explainvar2 <- round(S.dbrda$CCA$eig[2]/
                           sum(c(S.dbrda$CCA$eig, S.dbrda$CA$eig)),
                       3 ) * 100

#Plotting constrained ordination results
par(mar = c(5,5,4,4) + .1)

plot(scores(S.dbrda, display = "wa"), xlim = c(-2, 2.1), ylim = c(-2.3, 2.0),
     xlab = paste("dbRDA 1 (", S.explainvar1, "%)", sep = ""),
     ylab = paste("dbRDA 2 (", S.explainvar2, "%)", sep = ""),
     pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE
     )

axis(side = 1, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = T, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
box(lwd = 2)

points(scores(S.dbrda, display = "wa"),
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(scores(S.dbrda, display = "wa"),
     labels = row.names(scores(S.dbrda, display = "wa")))

#Plotting vectors for influence of environmental factors
vectors <- scores(S.dbrda, display = "bp")
arrows(0, 0, vectors[,1], vectors[,2],
       lwd = 2, lty = 1, length = .2, col = "red")
text(vectors[,1], vectors[,2], pos = 3,
     labels = row.names(vectors), col = "red")
axis(side = 3, lwd.ticks = 2, cex.axis = 1.2, las = 1, col = "red", lwd = 2.2,
     at = pretty(range(vectors[,1])) * 2, labels = pretty(range(vectors[,1])))
axis(side = 4, lwd.ticks = 2, cex.axis = 1.2, las = 1, col = "red", lwd = 2.2,
     at = pretty(range(vectors[,2])) * 2, labels = pretty(range(vectors[,2])))
```




```
env.chem <- as.matrix(env[,c(2:4)])

fungalS <- fungalBC

S.dbrda <- dbrda(fungalS ~ ., as.data.frame(env.chem))
ordiplot(S.dbrda)
```




```
S.dbrda0 <- dbrda(fungalS ~ 1, as.data.frame(env.chem))
S.dbrda1 <- dbrda(fungalS ~ ., as.data.frame(env.chem))

S.dbrda <- ordiR2step(S.dbrda0, S.dbrda1, perm.max = 999)
```

```
## Step: R2.adj= 0
## Call: fungalS ~ 1
##
##                 R2.adjusted
## <All variables> 0.012992343
## + N             0.006675066
## + C             0.004817216
## + P             0.001444302
## <none>          0.000000000
##
##     Df    AIC      F Pr(>F)
## + N  1 128.08 1.3158   0.17
```

Sorensen distance is influenced by Nitrogen concentration in the soil but BC is not influenced by Nitrogren, Carbon, or Phosphorus

Effect of plant abundance and diversity on fungal community structure

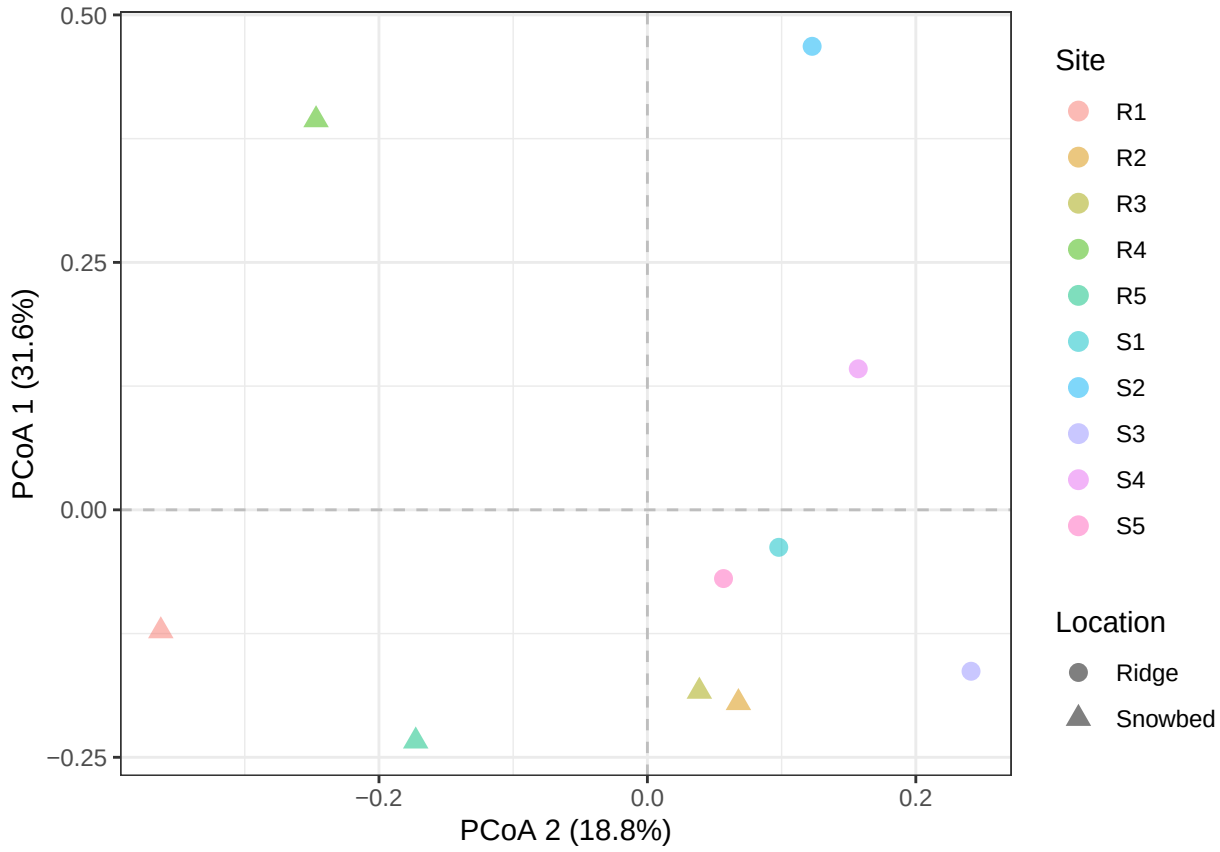
```
# Making new dataframe with summary of all statistics
plot.info <- as.data.frame(matrix(nrow = 10, ncol=1))
plot.info <- plot.info[,-1]
plot.info$Location <- c(rep("Ridge", 5), rep("Snowbed", 5))
rownames(plot.info) <- c("S1","S2","S3","S4","S5","R1","R2","R3","R4","R5")
plot.info$site <- c("S1","S2","S3","S4","S5","R1","R2","R3","R4","R5")

# Adding average soil nutrients
plot.info$P <- c(mean(env[1:5, 2]), mean(env[6:10, 2]), mean(env[11:15, 2]), mean(env[16:20, 2]), mean(
plot.info$N <- c(mean(env[1:5, 3]), mean(env[6:10, 3]), mean(env[11:15, 3]), mean(env[16:20, 3]), mean(
plot.info$C <- c(mean(env[1:5, 4]), mean(env[6:10, 4]), mean(env[11:15, 4]), mean(env[16:20, 4]), mean(

# Adding alpha diversity data
rarefied.sites <- as.data.frame((matrix(nrow = 10, ncol = 807)))
for (i in 1:(ncol(rarefied_site_species))){
  rarefied.sites[,i] <- c(sum(rarefied_site_species[1:5, i]), sum(rarefied_site_species[6:10, i]), sum(
}
rownames(rarefied.sites) <- c("S1","S2","S3","S4","S5","R1","R2","R3","R4","R5")

# Species richness
plot.info$richness <- rowSums((rarefied.sites > 0) * 1)


plot.info <- as.data.frame(plot.info)
```

PCoA of Grouped Fungal Communities

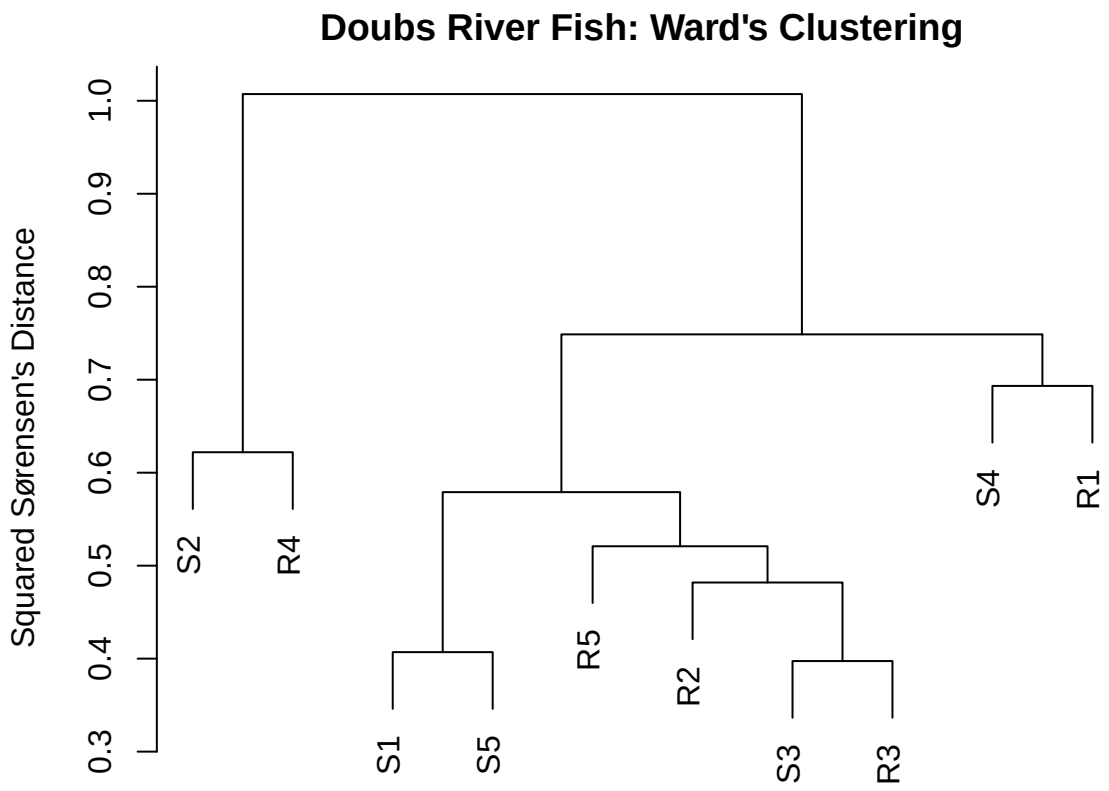
#Site-grouped Bray-Curtis

```
fungalBC <- vegdist(rarefied.sites, method = "bray")

fungal.pcoa <- cmdscale(fungalBC, eig = TRUE, k = 3)

explainvar1 <- round(fungal.pcoa$eig[1]/sum(fungal.pcoa$eig), 3) * 100
explainvar2 <- round(fungal.pcoa$eig[2]/sum(fungal.pcoa$eig), 3) * 100
explainvar3 <- round(fungal.pcoa$eig[3]/sum(fungal.pcoa$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

test <- rarefied.sites

test$Site <- c("S1","S2","S3","S4","S5","R1","R2","R3","R4","R5")

test$Location <- c(rep("Ridge", 5), rep("Snowbed", 5))

## Plotting
# make a fake pcoa
dist      <- vegdist(test[,1:807],  method = "bray")
```

```
fake_pcoa <- pcoa(dist)
#summary(fake_pcoa)

# get the out of pcoa
# site scores = samples
sitescores   <- fake_pcoa$vectors #we need only the first two axis
newdata <- as.data.frame(cbind(sitescores[,1:2], Location = test$Location, Site = test$Site))

#one way
bray <- ggplot(newdata, aes(y = as.numeric(Axis.1), x = as.numeric(Axis.2), shape = Location, color = Si
bray <- bray + geom_hline(yintercept = 0, color = "grey", linetype = "dashed")
bray <- bray + geom_vline(xintercept = 0, color = "grey", linetype = "dashed")
bray <- bray + geom_point(size = 3, alpha = .5)
bray <- bray + theme_bw()
bray <- bray + labs(y = paste("PCoA 1 (", explainvar1, "%)", sep = ""), x = paste("PCoA 2 (", explainva
bray
```



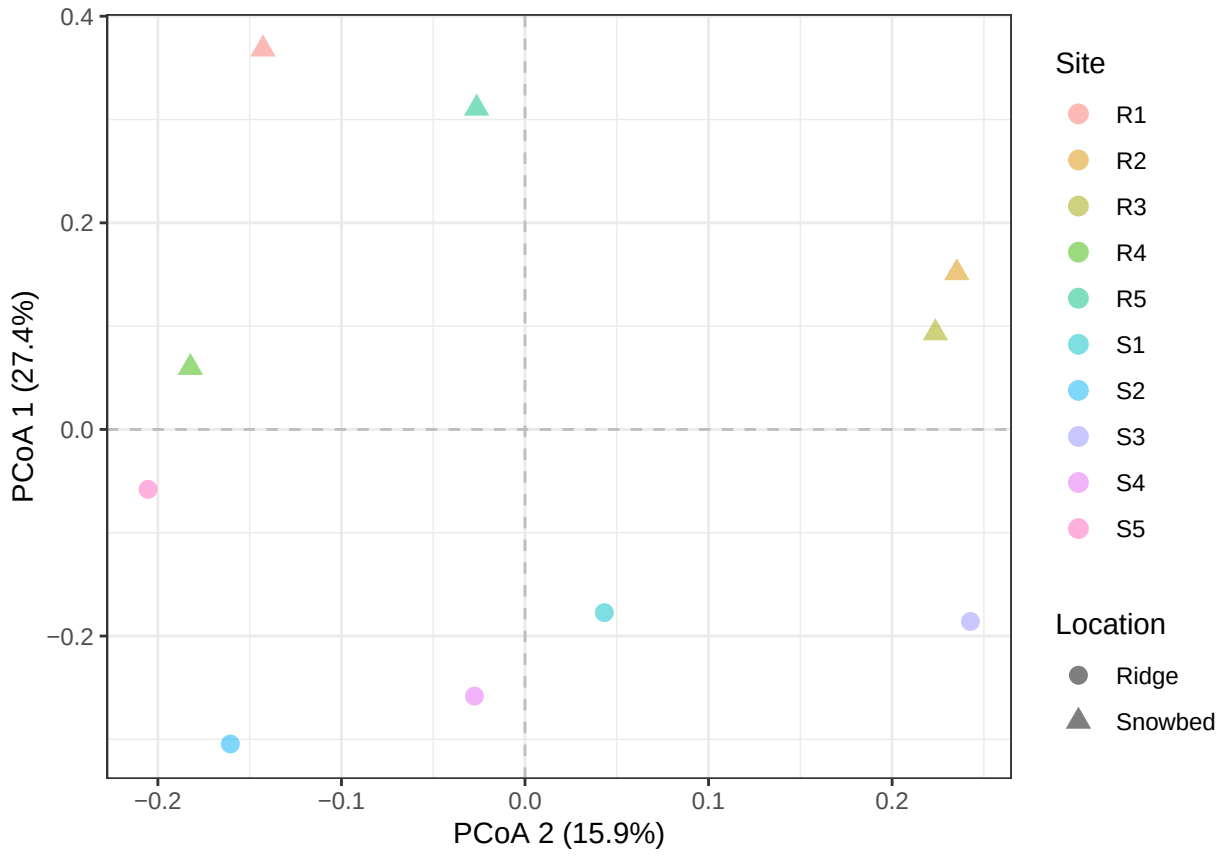

```
#Performing Cluster Analysis
fungal.ward <- hclust(fungalBC, method = "ward.D2")

#Plotting Cluster
par(mar = c(1,5,2,2) + .1)
plot(fungal.ward, main = "Doubs River Fish: Ward's Clustering",
     ylab = "Squared Sørensen's Distance")
```



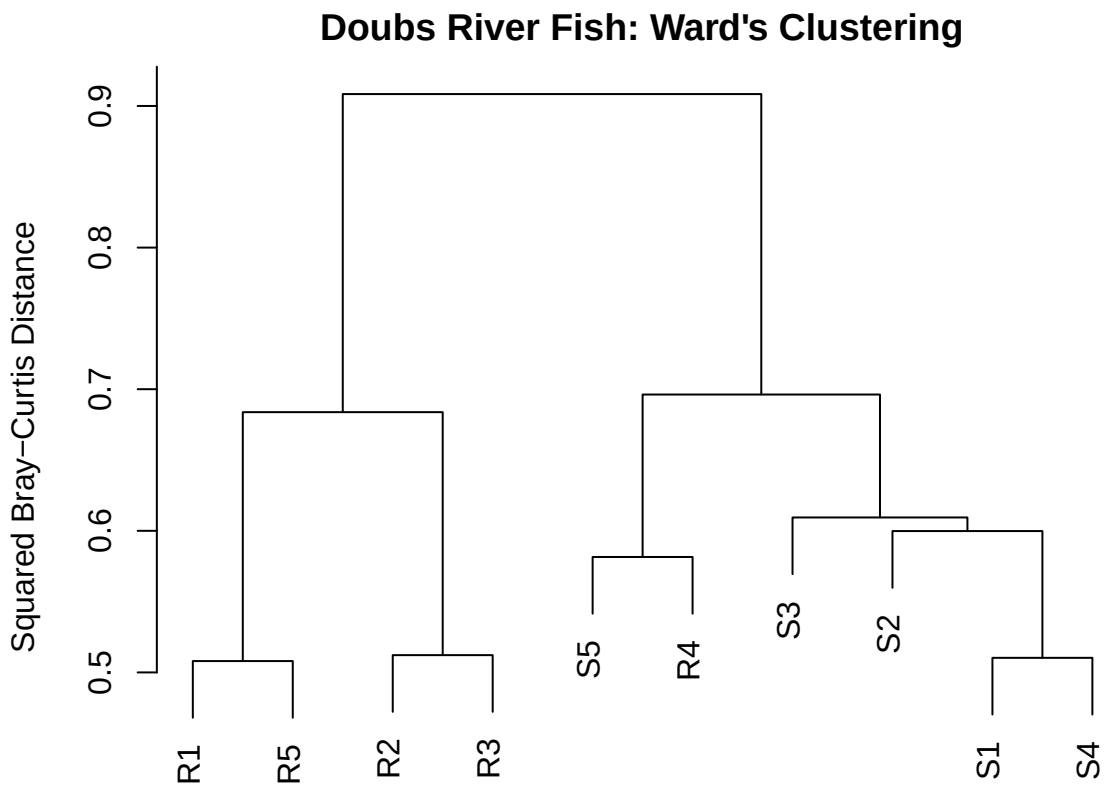
Site-grouped Sorensen

```
fungalBC <- vegdist(rarefied.sites, method = "bray", binary = TRUE)

fungal.pcoa <- cmdscale(fungalBC, eig = TRUE, k = 3)

explainvar1 <- round(fungal.pcoa$eig[1]/sum(fungal.pcoa$eig), 3) * 100
explainvar2 <- round(fungal.pcoa$eig[2]/sum(fungal.pcoa$eig), 3) * 100
explainvar3 <- round(fungal.pcoa$eig[3]/sum(fungal.pcoa$eig), 3) * 100
sum.eig <- sum(explainvar1, explainvar2, explainvar3)

test <- rarefied.sites

test$Site <- c("S1","S2","S3","S4","S5","R1","R2","R3","R4","R5")

test$Location <- c(rep("Ridge", 5), rep("Snowbed", 5))

## Plotting
# make a fake pcoa
dist      <- vegdist(test[,1:807],  method = "bray", binary = TRUE)
fake_pcoa <- pcoa(dist)
#summary(fake_pcoa)
```

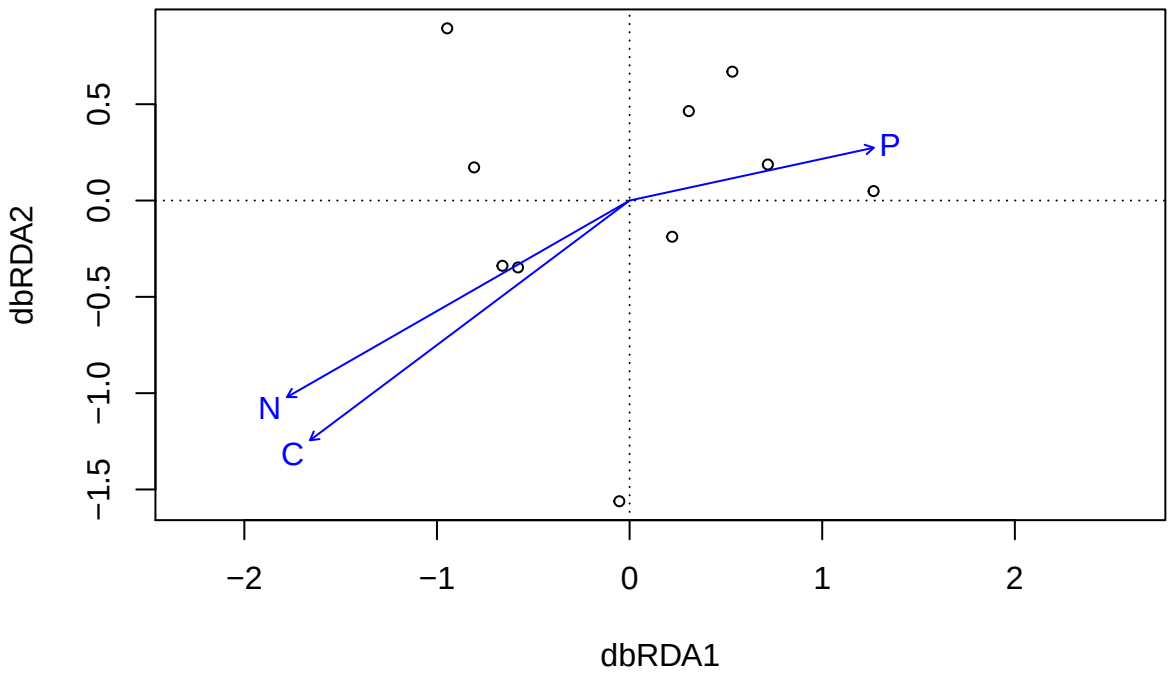
```
# get the out of pcoa
# site scores = samples
sitescores   <- fake_pcoa$vectors #we need only the first two axis
newdata <- as.data.frame(cbind(sitescores[,1:2], Location = test$Location, Site = test$Site))

#one way
bray <- ggplot(newdata, aes(y = as.numeric(Axis.1), x = as.numeric(Axis.2), shape = Location, color = S
bray <- bray + geom_hline(yintercept = 0, color = "grey", linetype = "dashed")
bray <- bray + geom_vline(xintercept = 0, color = "grey", linetype = "dashed")
bray <- bray + geom_point(size = 3, alpha = .5)
bray <- bray + theme_bw()
bray <- bray + labs(y = paste("PCoA 1 (", explainvar1, "%)", sep = ""), x = paste("PCoA 2 (", explainva
bray
```




```
#Performing Cluster Analysis
fungal.ward <- hclust(fungalBC, method = "ward.D2")

#Plotting Cluster
par(mar = c(1,5,2,2) + .1)
plot(fungal.ward, main = "Doubs River Fish: Ward's Clustering",
     ylab = "Squared Bray-Curtis Distance")
```




```
env.chem <- as.matrix(plot.info[,3:5])

fungalS <- fungalBC

S.dbrda <- dbrda(fungalS ~ ., as.data.frame(env.chem))
ordiplot(S.dbrda)
```




```
S.dbrda0 <- dbrda(fungalS ~ 1, as.data.frame(env.chem))
S.dbrda1 <- dbrda(fungalS ~ ., as.data.frame(env.chem))

S.dbrda <- ordiR2step(S.dbrda0, S.dbrda1, perm.max = 999)
```

```
## Step: R2.adj= 0
## Call: fungalS ~ 1
##
##                  R2.adjusted
## + N              0.066582703
## <All variables> 0.064946459
## + C              0.056735634
## + P              0.007422007
## <none>           0.000000000
```

```
plant <- read.csv("alpine_ridge_data/veg.csv")
plant <- t(plant)
colnames(plant) <- plant[1,]
plant <- plant[-1,]

plot.info$num.rock <- plant[,49]
plant <- plant[,-c(48,49)]

plant.num <- as.data.frame(matrix(nrow = 10, ncol = 47))
colnames(plant.num) <- colnames(plant)
```

```r
rownames(plant.num) <- rownames(plant)

for (i in 1:ncol(plant)){
  plant.num[,i] <- c(as.numeric(plant[,i]))
}


plot.info$plant.richness <- rowSums((plant.num > 0) * 1)

library(lmerTest)
```

```
## Warning: package 'lmerTest' was built under R version 4.0.5
```

```
## Loading required package: lme4
```

```
## Warning: package 'lme4' was built under R version 4.0.5
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
##
##     lmer
```

```
## The following object is masked from 'package:stats':
##
##     step
```

```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.0.5
```

```
## Loading required package: carData
```

```
## Registered S3 methods overwritten by 'car':
##   method                          from
##   influence.merMod                lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod         lme4
##   dfbetas.influence.merMod        lme4
```

```r
#richness.lm <- lm(data = plot.info, richness ~ C + P + N + num.rock + plant.richness)
# mixed model
#richness.lm <- lmer(data = plot.info, richness ~ C + P + N+ (1|num.rock))
#anv <- Anova(richness.lm, type = "III")
#print(anv)
# with plant richness mixed model
#richness.lm.1 <- lmer(data = plot.info, richness ~ C * P * N * plant.richness + (1|num.rock))
```

```
#anv.1 <- Anova(richness.lm, type = "III")
#print(anv.1)
# fixed model
#richness.lm.fixed <- lm(data = plot.info, richness ~ C + P + N)
#anv.fixed <- Anova(richness.lm.fixed, type = "III")
#print(anv.fixed)
# with plant richness model
#richness.lm.fixed <- lm(data = plot.info, richness ~ C + P + N + C:P + C:N + P:N + plant.richness + pl
#summary(richness.lm.fixed)
#anv.fixed <- Anova(richness.lm.fixed, type = "III")
#print(anv.fixed)
# fixed model
#richness.lm.fixed <- lm(data = plot.info, richness ~ C + P + N + Location)
#anv.fixed <- Anova(richness.lm.fixed, type = "III")
#print(anv.fixed)
```

```
average <- as.data.frame(matrix(nrow = 1, ncol = 807))

for (i in 1:ncol(rarefied_site_species)){
  average[,i] <- mean(rarefied_site_species[,i])
}

colnames(average) <- c(genus[,1])
average <- t(average)
```