Group.Bac_Diversity

Jocelyn Huang & Aishwarya Vaidya; Z620: Quantitative Biodiversity, Indiana University

07 March, 2025

Overview

Here we can write the basic introduction to the data set we are using, the citation to the data source, citation to the published article, and what the variables are in the data sets. We can also specify the research question we are curious about in this section.

Set up/Data clean-ups:

In the following chunk, we set up the document and load all required packages:

```r
rm(list = ls())
getwd()
```

```
## [1] "/cloud/project/QB2025_Huang/Group-project"
```

```r
setwd("/cloud/project/QB2025_Huang/Group-project")

library(vegan)
```

```
## Loading required package: permute

## Loading required package: lattice
```

```r
library(ade4)
library(viridis)
```

```
## Loading required package: viridisLite
```

```r
library(gplots)
```

```
##
## Attaching package: 'gplots'

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

```r
library(indicspecies)
library(ggplot2)
```

We begin by first loading all required data:

```r
# Load long data set to build site by species matrix:
load("./raw_data/longdataBac_objects2_datadryad.rda")
Bacteria <- longdataBac_datadryad #rename
rm(longdataBac_datadryad)

## Make SbyS matrix based on Plot ID:
```

```
### Note that each Plot ID has a corresponding habitat type
### and each Quarant ID (Plot ID without letter) has a landscape type
SbyS <- with(Bacteria, tapply(Counts, list(PlotID, Sender), sum, default = 0))

## Store SbyS matrix into a .txt file in Cleaned_data
####write.table(SbyS, file = "bac_SbyS.txt", sep = "\t", row.names = TRUE, col.names = NA, quote = FALS
### If loading SbyS again:
####SbyS<- read.table("/cloud/project/QB2025_Huang/Group-project/Cleaned_data/bac_SbyS.txt", header = T


# Load wide data set to build environmental matrix:
load("./raw_data/Bac_wide_plot_final2_datadryad.rda")
env <- Bac_wide_plot #rename
rm(Bac_wide_plot)
## Drop unnecessary variables in env matrix:
rownames(env) <- env$PlotID
env <- env[, -c(1, 2, 3, 9)]
## Drop NA in env:
env <- na.omit(env)

## Match SbyS with it:
SbyS <- SbyS[rownames(env),]

## Similarly, store the env matrix
###write.table(env, file = "bac_env.txt", sep = "\t", row.names = TRUE, col.names = NA, quote = FALSE)

# Use Bacteria data frame to find spatial data (xy):
xy <- aggregate(cbind(POINT_X, POINT_Y) ~ PlotID, data = Bacteria, FUN = mean)
## Match xy to env:
xy <- xy[rownames(env),]
## Store xy matrix
###write.table(xy, file = "bac_xy.txt", sep = "\t", row.names = TRUE, col.names = NA, quote = FALSE)
```

Richness and env:

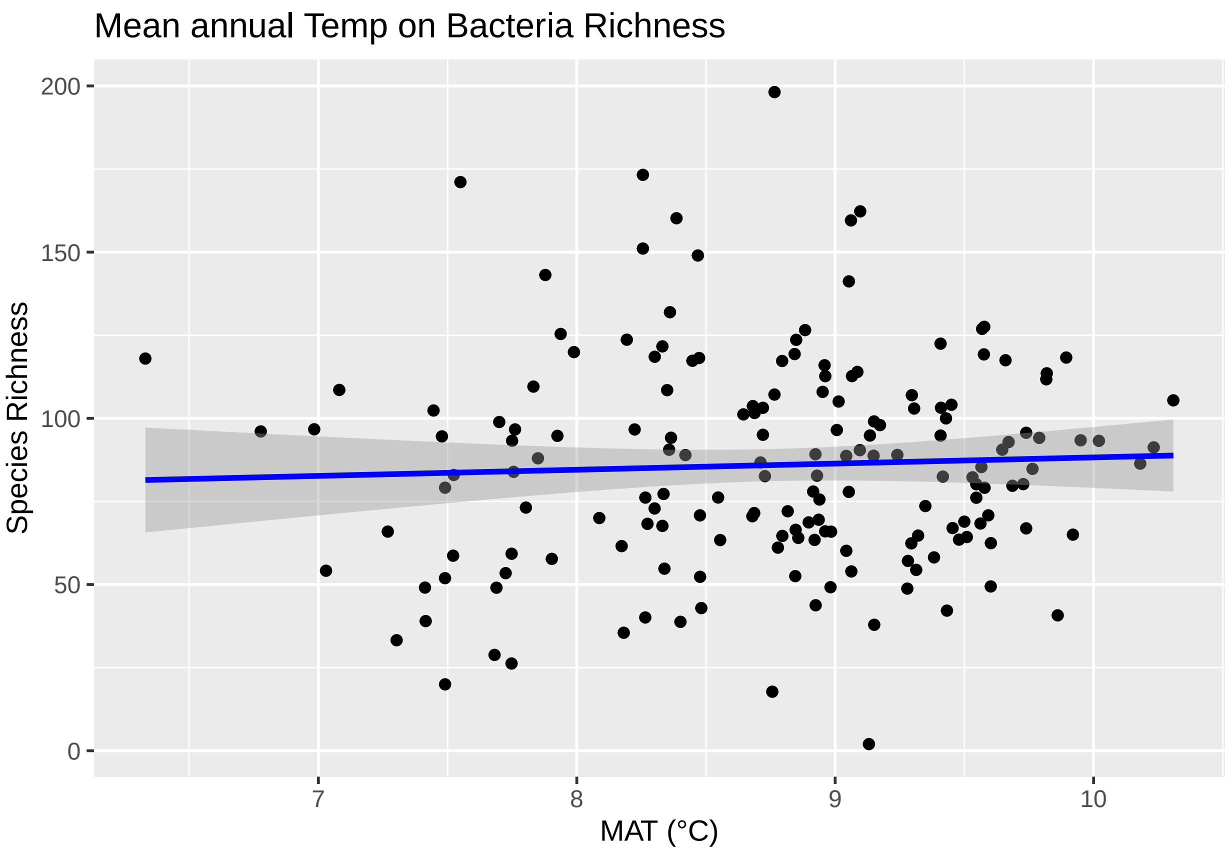
Calculate the species observed richness for each sites after rarefaction.

```
min.N <- min(rowSums(SbyS))
rarefy <- rarefy(x = SbyS, sample = min.N, se = TRUE)
bac.richness <- t(rarefy)
#bac.richness$S is the species richness

env.analyze <- cbind(bac.richness, env)

mod <- lm(S ~ MAP + MAT + Habitat + Landscape + average_Temp_DL, data = env.analyze)
summary(mod)
```

```
##
## Call:
## lm(formula = S ~ MAP + MAT + Habitat + Landscape + average_Temp_DL,
##     data = env.analyze)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -80.503 -18.154  -3.302  15.053  89.184
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -1.89746   39.37633  -0.048 0.961626
## MAP                    0.06634    0.01196   5.546 1.16e-07 ***
## MAT                   15.20233    3.89774   3.900 0.000141 ***
## Habitatgrassland      14.81109    5.98164   2.476 0.014311 *
## Habitatarable         13.88445    6.33949   2.190 0.029943 *
## Habitatsettlement     14.69627    7.03014   2.090 0.038137 *
## Landscapeagriculture -0.24874    5.80236  -0.043 0.965859
## Landscapeurban        -4.98850    5.61993  -0.888 0.376048
## average_Temp_DL       -6.62152    2.25773  -2.933 0.003846 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.19 on 162 degrees of freedom
## Multiple R-squared:  0.2047, Adjusted R-squared:  0.1654
## F-statistic: 5.212 on 8 and 162 DF,  p-value: 8.293e-06
```

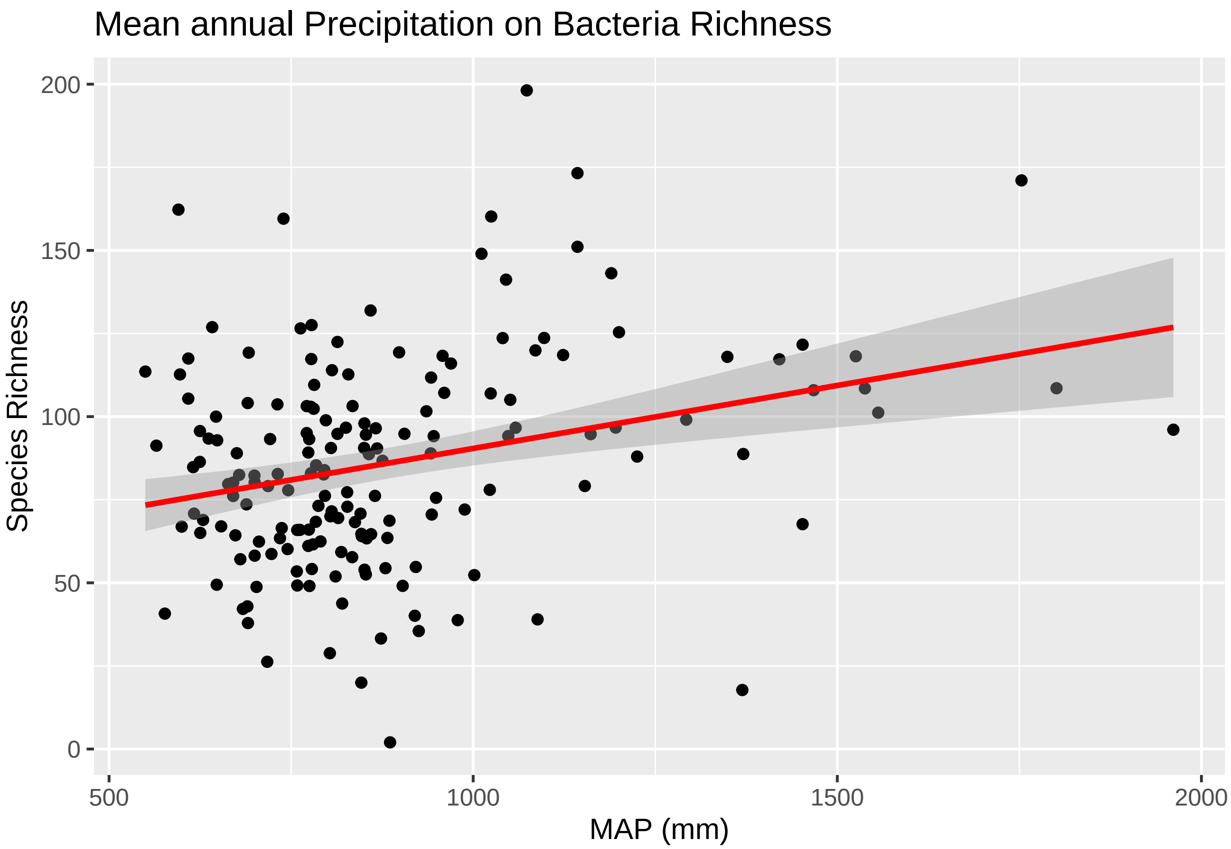
```
ggplot(env.analyze, aes(x = MAT, y = S)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Mean annual Temp on Bacteria Richness", x = "MAT (°C)", y = "Species Richness")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



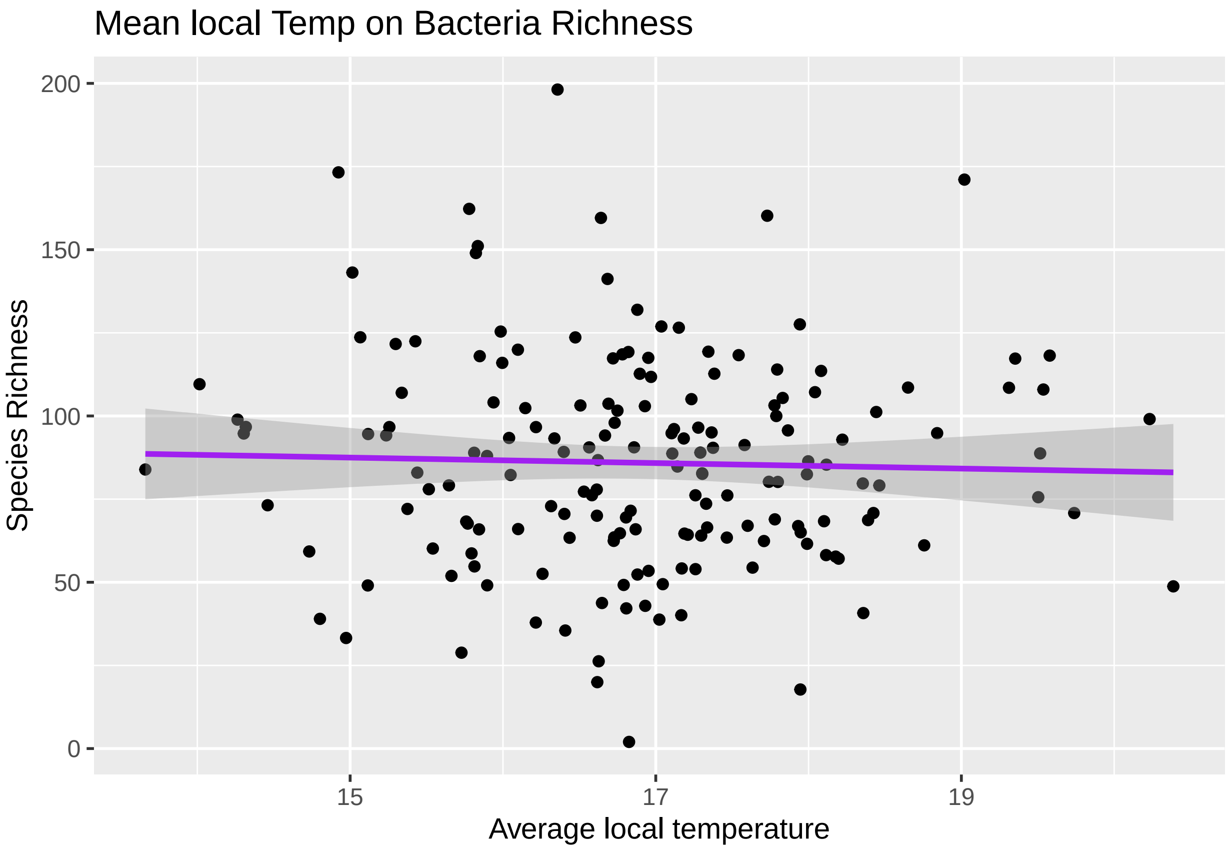
```r
ggplot(env.analyze, aes(x = MAP, y = S)) +
  geom_point() +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Mean annual Precipitation on Bacteria Richness", x = "MAP (mm)", y = "Species Richness"
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



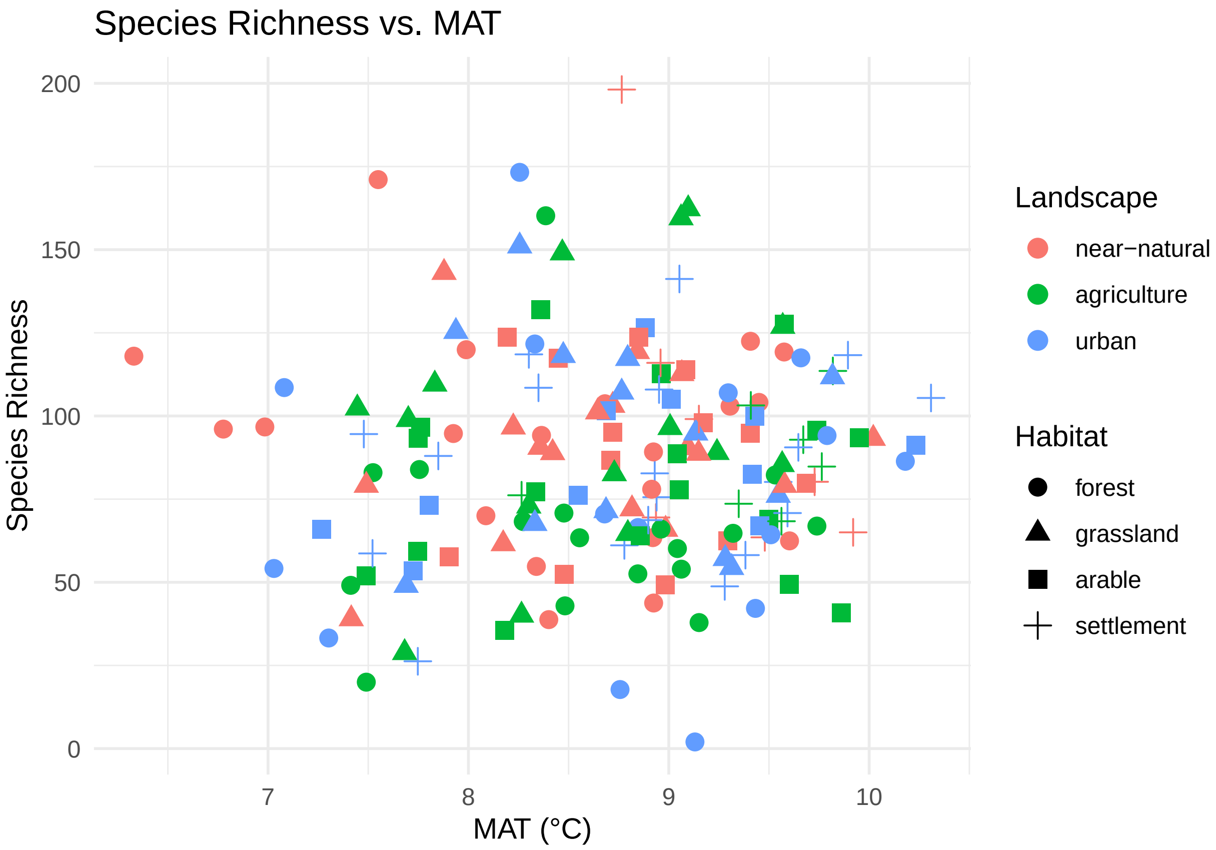
```r
ggplot(env.analyze, aes(x = average_Temp_DL, y = S)) +
  geom_point() +
  geom_smooth(method = "lm", col = "purple") +
  labs(title = "Mean local Temp on Bacteria Richness", x = "Average local temperature", y = "Species Ri
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

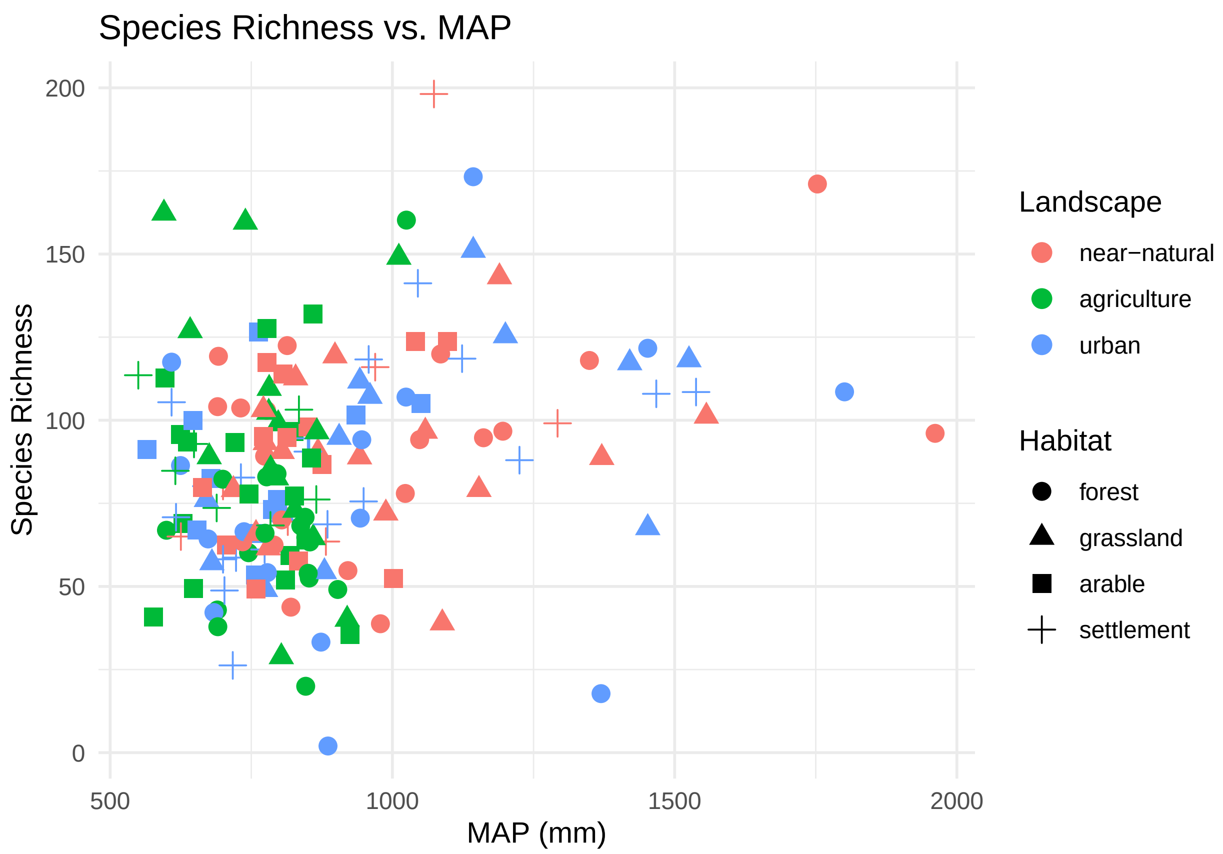


Richness for habitat and landscape:

```r
ggplot(env.analyze, aes(x = MAT, y = S, color = Landscape, shape = Habitat)) +
  geom_point(size = 3) +
  labs(title = "Species Richness vs. MAT", x = "MAT (°C)", y = "Species Richness") +
  theme_minimal()
```



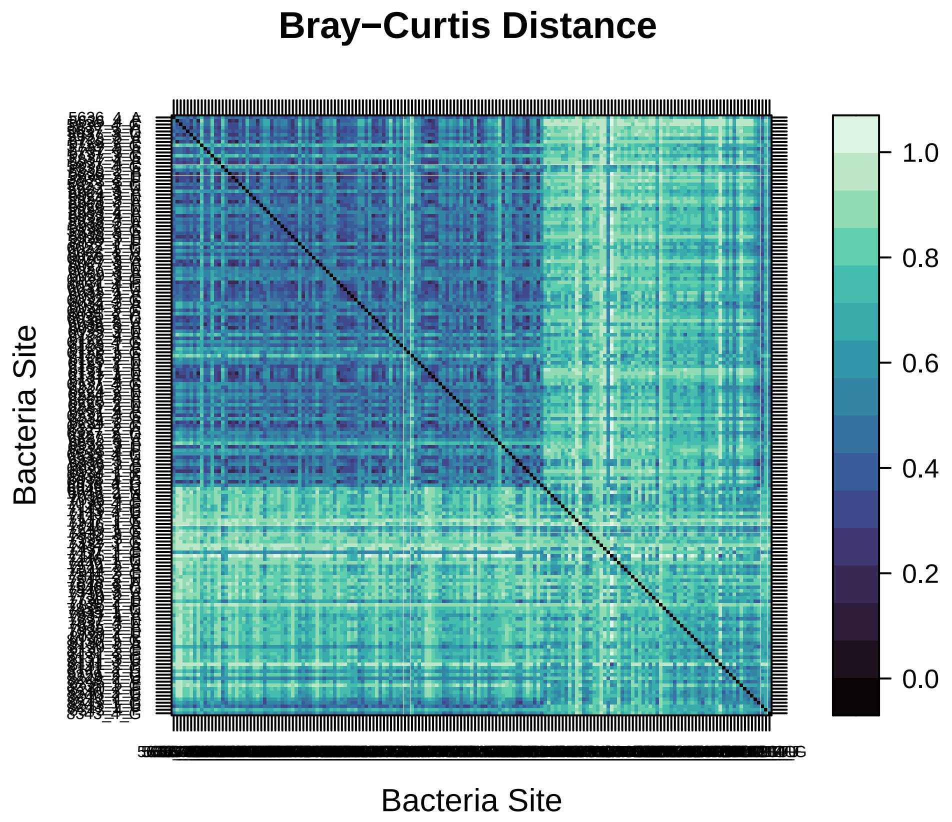
```r
ggplot(env.analyze, aes(x = MAP, y = S, color = Landscape, shape = Habitat)) +
  geom_point(size = 3) +
  labs(title = "Species Richness vs. MAP", x = "MAP (mm)", y = "Species Richness") +
  theme_minimal()
```



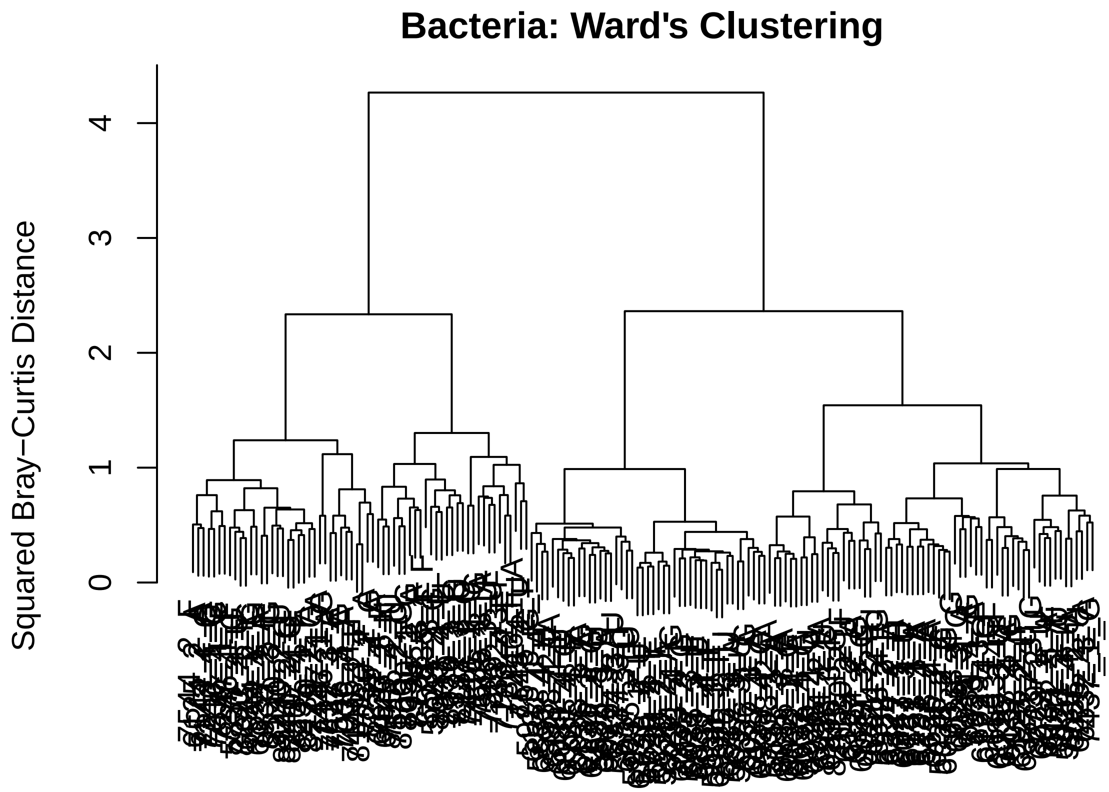
Next, we find the remsemblence matrix based on the cleaned up Site-by-Species matrix. Here, we try using Bary-Curtis Dissimilarity since we are working on abundance data.

Visualization

```
bac.bc <- vegdist(SbyS, method = "bray", upper = TRUE, diag = TRUE)

# Heatmap:
order <- rev(attr(bac.bc, "Labels"))
levelplot(as.matrix(bac.bc)[, order], aspect = "iso", col.regions = mako,
          xlab = "Bacteria Site", ylab = "Bacteria Site", scales = list(cex = 0.5),
          main = "Bray-Curtis Distance")
```

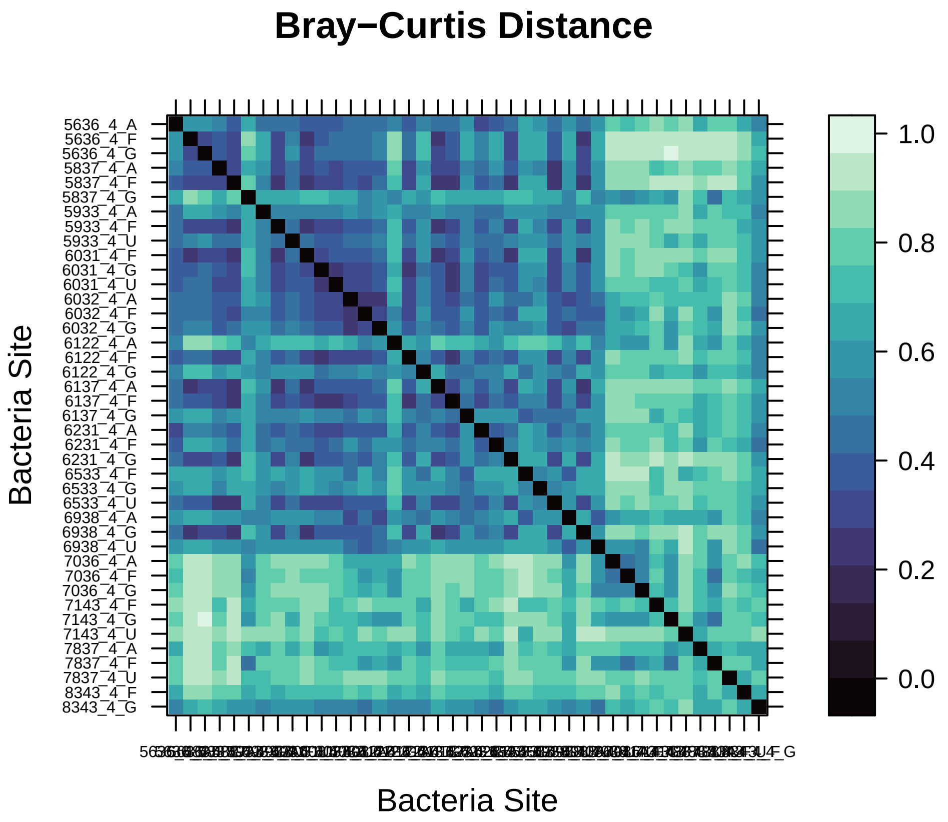


```
# Cluster:
bac.ward <- hclust(bac.bc, method = "ward.D2")
par(mar = c(1, 5, 2, 2) + 0.1)
plot(bac.ward, main = "Bacteria: Ward's Clustering",
     ylab = "Squared Bray-Curtis Distance")
```

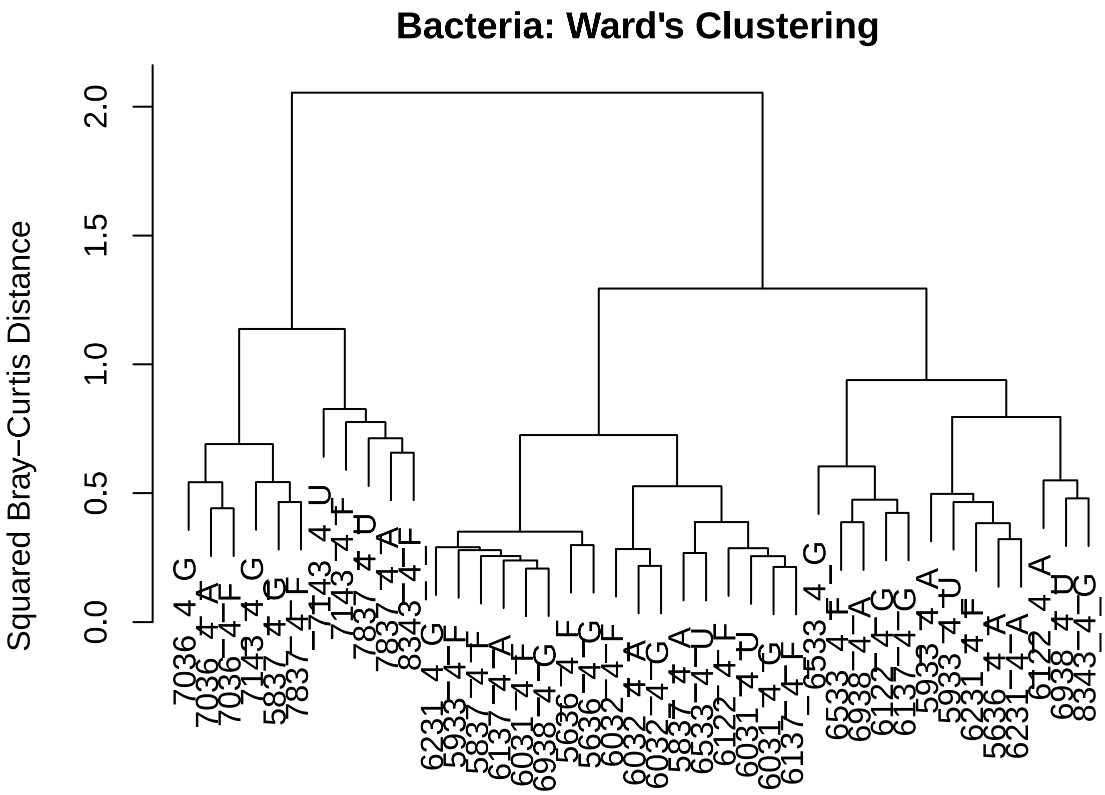


For our complete dataset, the visualization seems to be too blury to be informative. Therefore, to get a better visual, we decided to subset the dataset:

```
# Drop all but "_4" sites to have a better idea:
bac.reduced <- SbyS[!grepl("_3|_2|_1", rownames(SbyS)), ]
env.reduced <- env[!grepl("_3|_2|_1", rownames(env)), ]

# Reduced resemblance matrix:
bac.rd.bc <- vegdist(bac.reduced, method = "bray", upper = TRUE, diag = TRUE)

# Heat Map:
order.rd <- rev(attr(bac.rd.bc, "Labels"))
levelplot(as.matrix(bac.rd.bc)[, order.rd], aspect = "iso",
          col.regions = mako,
          xlab = "Bacteria Site", ylab = "Bacteria Site",
          scales = list(cex = 0.5),
          main = "Bray-Curtis Distance")
```



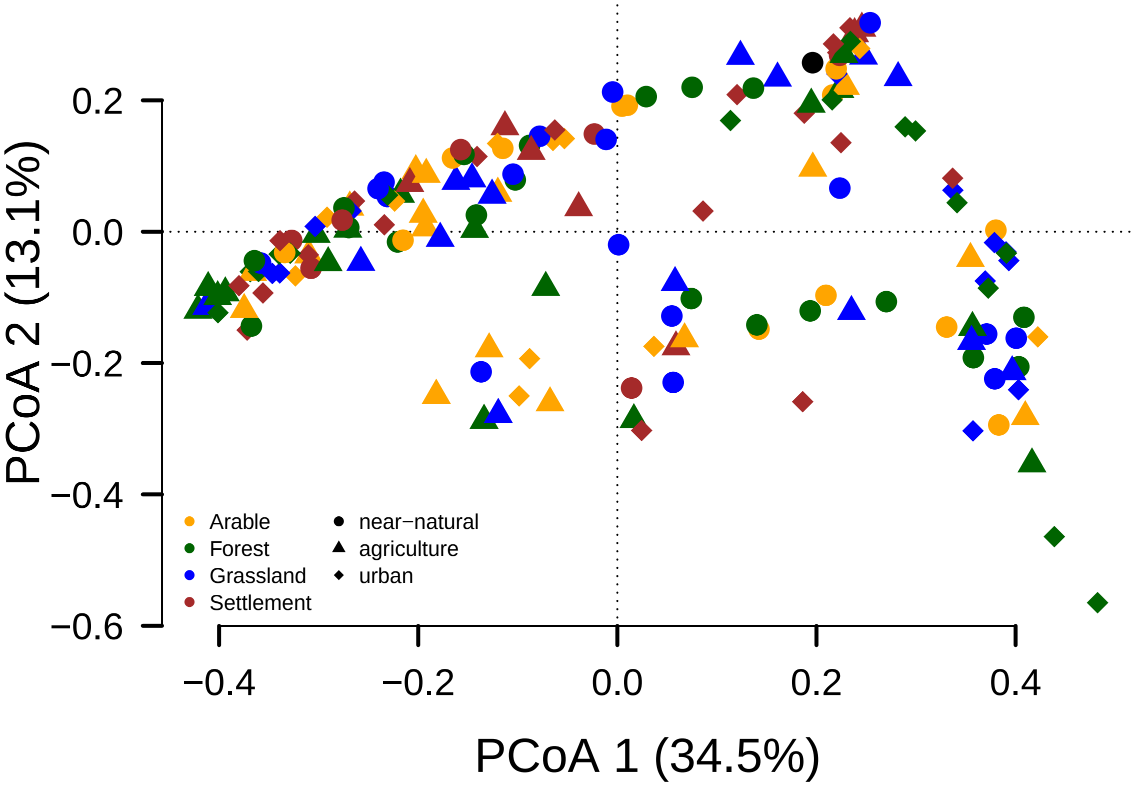
```
# Cluster:
bac.ward.rd <- hclust(bac.rd.bc, method = "ward.D2")
par(mar = c(1, 5, 2, 2) + 0.1)
plot(bac.ward.rd, main = "Bacteria: Ward's Clustering",
     ylab = "Squared Bray-Curtis Distance")
```



Next, we performed a PCoA Analysis on the original dataset:

```
# PCoA:
##using original SbyS/resemblance matrix
bac.pcoa <- cmdscale(bac.bc, eig = TRUE, k = 3)

##Variation explained by the first three axes:
bac.explainvar1 <- round(bac.pcoa$eig[1]/sum(bac.pcoa$eig), 3)*100
bac.explainvar2 <- round(bac.pcoa$eig[2]/sum(bac.pcoa$eig), 3)*100
bac.explainvar3 <- round(bac.pcoa$eig[3]/sum(bac.pcoa$eig), 3)*100
sum.eig <- sum(bac.explainvar1, bac.explainvar2, bac.explainvar3)

##Begin graphing the PCoA:
###Define each point with different color representing different habitat
habitat_colors <- c("_A" = "orange", "_F" = "darkgreen", "_G" = "blue", "_U" = "brown")
point_name <- gsub("_(A|U|G|F)$", "", rownames(bac.pcoa$points)) #display name
point_colors <- sapply(row.names(bac.pcoa$points), function(name) {
  match <- grep("_A|_F|_G|_U", name, value = TRUE)
  if (length(match) > 0) {
    return(habitat_colors[substr(match, nchar(match) - 1, nchar(match))])
  }
  else {
    return("black")  # Default color for other rows
  }
})

###Define each point symbol with landscape type in env subset
ldsp <- env[rownames(bac.pcoa$points), "Landscape"]
pch_val <- c(16,17,18)
```

```
pch_map <- pch_val[as.numeric(factor(ldsp))]

## Plot the PCoA:
par(mar = c(5, 5, 1, 2) + 0.1)
plot(bac.pcoa$points[, 1], bac.pcoa$points[, 2],
     xlim = range(bac.pcoa$points[, 1]),
     ylim = range(bac.pcoa$points[, 2]),
     xlab = paste("PCoA 1 (", 34.5, "%)", sep = ""),
     ylab = paste("PCoA 2 (", 13.1, "%)", sep = ""),
     pch = pch_map, cex = 1.5, col = point_colors,
     type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE)
axis(side = 1, labels = TRUE, lwd.ticks = 2, cex.axis = 1.2, las = 1)
axis(side = 2, labels = TRUE, lwd.ticks = 2, cex.axis = 1.2, las = 1)
abline(h = 0, v = 0, lty = 3)
points(bac.pcoa$points[, 1], bac.pcoa$points[, 2],
       pch = pch_map, cex = 1.5, bg = "gray", col = point_colors)
names(habitat_colors) <- c("Arable", "Forest", "Grassland", "Settlement")
legend(x = -0.45, y = -0.4, legend = names(habitat_colors),
       col = habitat_colors,
       pch = 16,
       cex = 0.7, # Smaller text
       pt.cex = 0.7,
       bty = "n")
legend(x = -0.3, y = -0.4, legend = levels(factor(ldsp)),
       pch = pch_val,
       col = "black",
       cex = 0.7,
       pt.cex = 0.7,
       bty = "n")
```



In the above graph, color represets habitat tyepe, and shapre represents landscape type.

Hypothesis testing:

PERMANOVA:

We then performed hypothesis testing on that.

```
# PERMANOVA:
land_type <- env$Landscape
habitat <- env$Habitat
adonis2(SbyS ~ land_type, method = "bray", permutation = 999)
```

```
## Permutation test for adonis under reduced model
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = SbyS ~ land_type, permutations = 999, method = "bray")
##           Df SumOfSqs      R2      F Pr(>F)
## Model      2    0.512 0.01293 1.1002   0.29
## Residual 168   39.113 0.98707
## Total    170   39.626 1.00000
```

```
adonis2(SbyS ~ habitat, method = "bray", permutation = 999)
```

```
## Permutation test for adonis under reduced model
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = SbyS ~ habitat, permutations = 999, method = "bray")
##           Df SumOfSqs      R2      F Pr(>F)
## Model      3    0.847 0.02137 1.2157  0.185
## Residual 167   38.779 0.97863
## Total    170   39.626 1.00000
```

Neither of them are significant. (p all larger than 0.2).

###Mantel test:

```
# Mantel test:
env.ds <- vegdist(scale(env[3:5]), method = "euclid", na.rm = T)
mantel(bac.bc, env.ds)
```

```
##
## Mantel statistic based on Pearson's product-moment correlation
##
## Call:
## mantel(xdis = bac.bc, ydis = env.ds)
##
## Mantel statistic r: 0.04693
##       Significance: 0.152
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0596 0.0750 0.0933 0.1144
## Permutation: free
## Number of permutations: 999
```

Correlation is only 0.05, which is very weak.

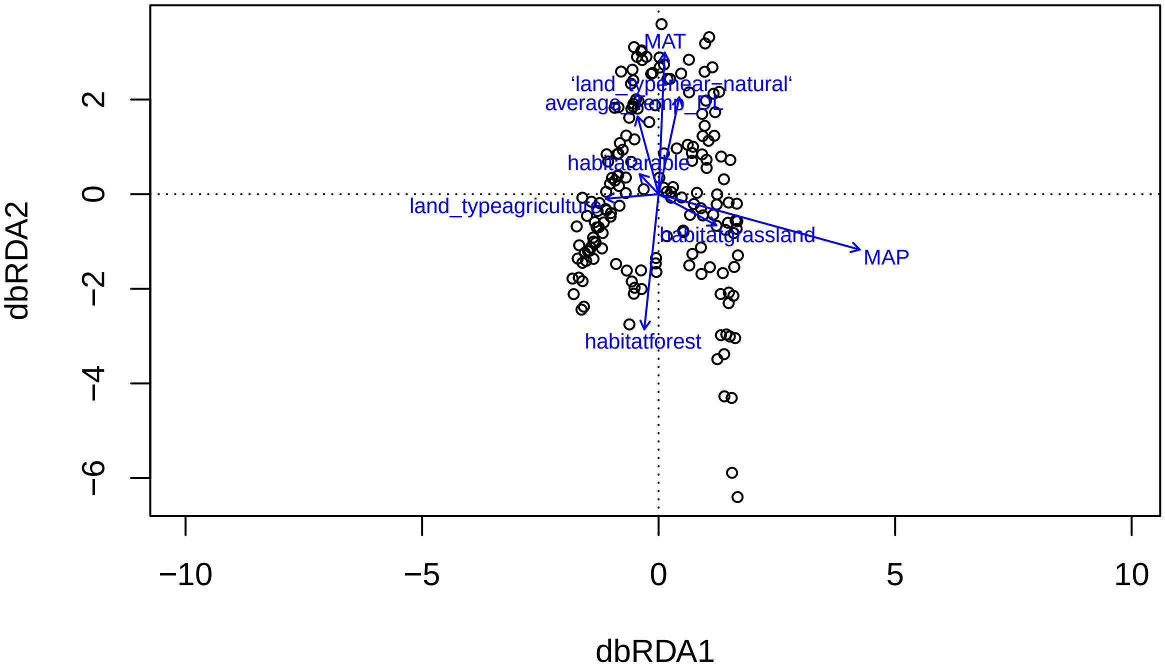
Constriant ordination:

Let's take a look at the dbRDA:

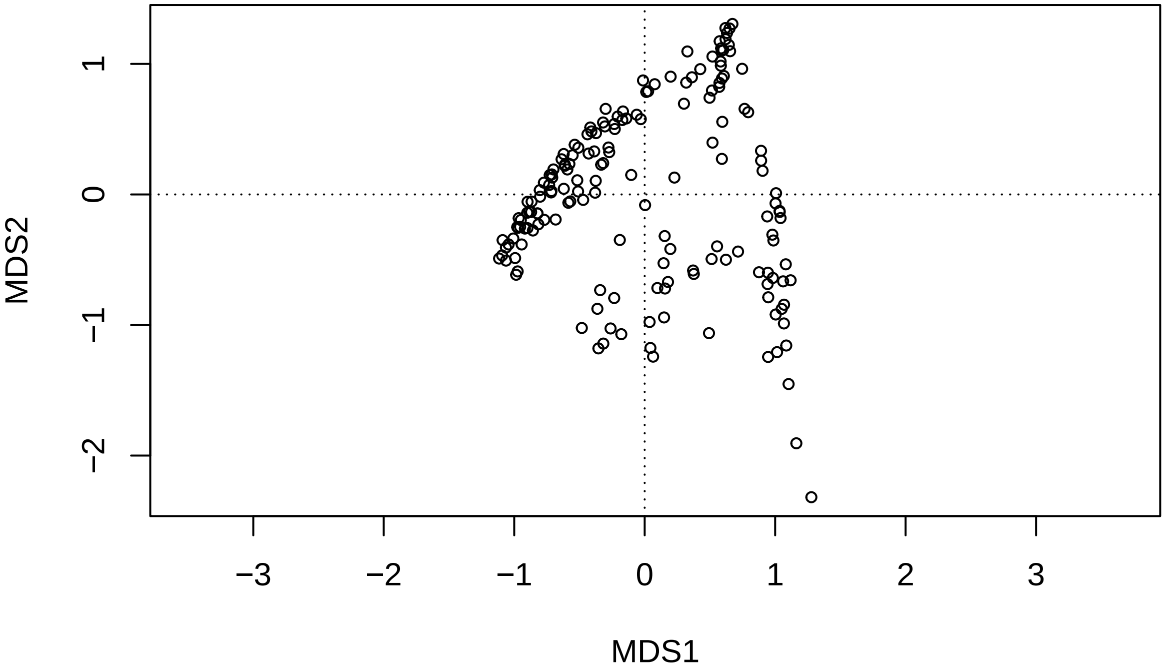
```
# Constraint Ordination:
## Take all variables (not just the continous ones)
## To make dbrda able to handle categorical variable, convert the cat variable to dummy variables:
landscape_dummy <- model.matrix(~ land_type - 1, data = env)
habitat_dummy <- model.matrix(~ habitat - 1, data = env)

## Now that landscape and habitat are converted into binary (0 and 1), cbind them back into env:
env_final <- cbind(env[, 3:5], landscape_dummy, habitat_dummy)

## Perform dbRDA:
bac.dbrda <- dbrda(bac.bc ~ ., as.data.frame(env_final)) # using abundance based distance
```

```
##
## Some constraints or conditions were aliased because they were redundant. This
## can happen if terms are linearly dependent (collinear): 'land_typeurban',
## 'habitatsettlement'
```

```
ordiplot(bac.dbrda)
```



```
## Model selection:
bac.dbrda.mod0 <- dbrda(bac.bc ~ 1, as.data.frame(env_final)) # Model only the intercept
ordiplot (bac.dbrda.mod0) # This is PCoA
```



```
bac.dbrda.mod1 <- dbrda(bac.bc ~ ., as.data.frame (env_final)) # Full model
```

```
##
## Some constraints or conditions were aliased because they were redundant. This
## can happen if terms are linearly dependent (collinear): 'land_typeurban',
## 'habitatsettlement'
```

```r
bac.dbrda <- ordiR2step(bac.dbrda.mod0, bac.dbrda.mod1, perm.max = 200) #select lowest AIC
```

```
##
## Some constraints or conditions were aliased because they were redundant. This
## can happen if terms are linearly dependent (collinear): 'land_typeurban',
## 'habitatsettlement'

## Step: R2.adj= 0
## Call: bac.bc ~ 1
##
##                              R2.adjusted
## <All variables>            0.1125123347
## + MAP                      0.0659054311
## + habitatgrassland         0.0033008729
## + habitatsettlement        0.0027127408
## + land_typeagriculture     0.0021476072
## + habitatforest            0.0004251180
## <none>                     0.0000000000
## + land_typeurban          -0.0000995165
## + MAT                     -0.0001172578
## + `land_typenear-natural` -0.0003120578
## + average_Temp_DL         -0.0013213171
## + habitatarable           -0.0015387323
##
##       Df    AIC      F Pr(>F)
## + MAP  1 619.52 12.994  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.06590543
## Call: bac.bc ~ MAP
##
##                            R2.adjusted
## <All variables>             0.11251233
## + MAT                       0.08749355
## + habitatforest             0.06886891
## + habitatsettlement         0.06841110
## + habitatgrassland          0.06604278
## + `land_typenear-natural`   0.06598177
## <none>                      0.06590543
## + average_Temp_DL           0.06582890
## + habitatarable             0.06508705
## + land_typeurban            0.06442872
## + land_typeagriculture      0.06356090
##
##       Df    AIC      F Pr(>F)
## + MAT  1 616.51 4.9982  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.08749355
## Call: bac.bc ~ MAP + MAT
##
##                            R2.adjusted
## <All variables>            0.11251233
## + average_Temp_DL          0.10466208
## + habitatsettlement        0.09221234
## + habitatforest            0.08898646
## + `land_typenear-natural`  0.08795143
## + habitatgrassland         0.08793302
## + habitatarable            0.08780195
## <none>                     0.08749355
## + land_typeagriculture     0.08687125
## + land_typeurban           0.08651667
##
##                   Df    AIC      F Pr(>F)
## + average_Temp_DL  1 614.24 4.2215  0.004 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.1046621
## Call: bac.bc ~ MAP + MAT + average_Temp_DL
##
##                           R2.adjusted
## <All variables>             0.1125123
## + habitatforest             0.1099402
## + habitatsettlement         0.1070189
## + `land_typenear-natural`   0.1068611
## + habitatarable             0.1059685
## + habitatgrassland          0.1046695
## <none>                      0.1046621
## + land_typeurban            0.1041510
## + land_typeagriculture      0.1039309
##
##                 Df   AIC      F Pr(>F)
## + habitatforest  1 614.2 1.9903  0.022 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Step: R2.adj= 0.1099402
## Call: bac.bc ~ MAP + MAT + average_Temp_DL + habitatforest
##
##                           R2.adjusted
## + habitatsettlement         0.1134846
## <All variables>             0.1125123
## + `land_typenear-natural`   0.1120930
## <none>                      0.1099402
## + land_typeurban            0.1094994
## + habitatarable             0.1093976
## + land_typeagriculture      0.1093416
## + habitatgrassland          0.1081475
```

```
bac.dbrda$call$formula #model fomular
```

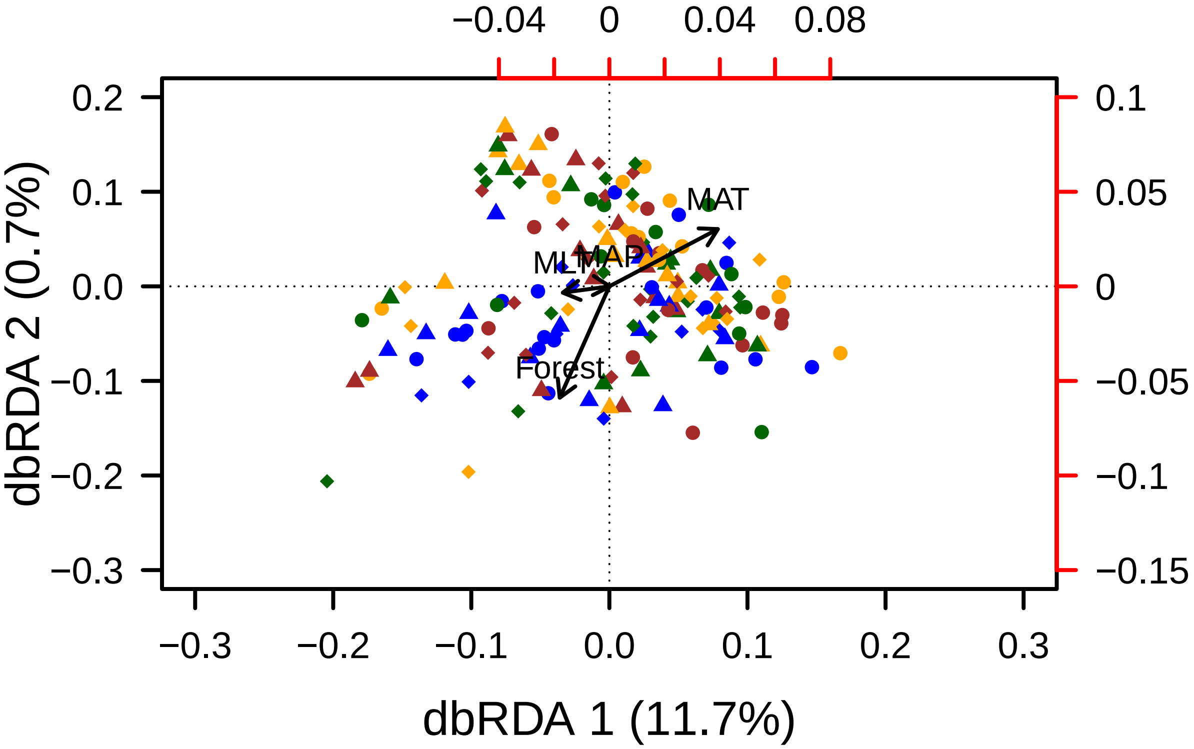
```
## bac.bc ~ MAP + MAT + average_Temp_DL + habitatforest
```

```
permutest(bac.dbrda, permutations = 999) # model significance
```

```
##
```

```
## Permutation test for dbrda under reduced model
##
## Permutation: free
## Number of permutations: 999
##
## Model: dbrda(formula = bac.bc ~ MAP + MAT + average_Temp_DL +
## habitatforest, data = as.data.frame(env_final))
## Permutation test for all constrained eigenvalues
##           Df Inertia      F Pr(>F)
## Model      4   5.186 6.2496  0.001 ***
## Residual 166  34.439
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
explainvar1 <-  round(bac.dbrda$CCA$eig[1] /
                           sum(c(bac.dbrda$CCA$eig, bac.dbrda$CA$eig)), 3) * 100
explainvar2 <- round(bac.dbrda$CCA$eig[2] /
                          sum(c(bac.dbrda$CCA$eig, bac.dbrda$CA$eig)), 3) * 100

# Plot the ordination plot:
par(mar = c(5,5,4,4) + 0.1)
plot(bac.dbrda$CA$u,
     xlim = c(-0.3, 0.3),
     ylim = c(-0.3, 0.2),
     xlab = paste("dbRDA 1 (", explainvar1, "%)",sep = ""),
     ylab = paste("dbRDA 2 (", 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(bac.dbrda$CA$u, pch = pch_map, cex = 1, bg = "gray",
       col = habitat_colors)
bc.vectors <- coef(bac.dbrda)
rownames(bc.vectors) <- c("MAP", "MAT", "MLT", "Forest")
arrows(0, 0, bc.vectors[, 1], bc.vectors[, 2],
       lwd = 2, lty = 1, length = 0.1, col = "black")
text(bc.vectors[, 1], bc.vectors[, 2], pos = 3,
     labels = row.names(bc.vectors), col = "black")
axis(side = 3, lwd.ticks = 2, cex.axis = 1.2, las = 1, col = "red",
     lwd = 2.2, at = pretty(range(bc.vectors[, 1])) * 2,
     labels = pretty(range(bc.vectors[, 1])))
axis(side = 4, lwd.ticks = 2, cex.axis = 1.2, las = 1, col = "red",
     lwd = 2.2, at = pretty(range(bc.vectors[, 2])) * 2,
     labels = pretty(range(bc.vectors[, 2])))
```


−0.04
0
0.04
0.08
0.2
0.1
0.0
−0.1
−0.2
−0.3
0.1
0.05
0
−0.05
−0.1
−0.15
MAT
MLT
MAP
Forest
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
dbRDA 1 (11.7%)
dbRDA 2 (0.7%)