

# Spatial vs. temporal storage: The dual role of microbial seed banks in driving geographical patterns of microbial diversity

August 4, 2015

## Constrained Ordination

Constrained ordination explores the relationships between two matrices: an **explanatory matrix** and a **response matrix**. Canonical correspondence analysis (CCA) and redundancy analysis (RDA) are two types of constrained ordination. Constrained ordination works by first conducting multivariate multiple linear regression followed either by correspondence analysis (CA) with CCA or Principal Components Analysis (PCA) with RDA, while using the matrix of fitted values to obtain a constrained ordination. A permutation test can then be used to test for overall significance.

We will start by creating an explanatory matrix that contains water chemistry data. We will then use the `cca()` function from the `vegan` package. Note, we have to specify that we want the `cca` function in the `vegan` package because there are `cca` functions in both `vegan` and `ade4`! We will then use permutation tests to evaluate the significance of our model. Finally, we will test the influence of each environmental variable on the constrained axes.

```
# Get Dominant members
cutoffs <- c(15, 30, 60, 125, 250)#, 500, 1000, 2000, 4000, 8000, 16000, 22098)
cutoff <- 30

all.sub.comm <- all.comm[, 1:cutoff]
active.sub.comm <- active.comm[, 1:cutoff]

# Define Environmental Matrix
env.chem <- env.dat

# Conduct CCA
active.cca <- vegan::cca(active.sub.comm ~ env.chem)

# Permutation Tests
anova(active.cca, by = "axis")
cca.fit <- envfit(active.cca, env.chem, perm = 999)
cca.fit

# Calculate Explained Variation
cca.explainvar1 <- round(active.cca$CCA$eig[1] /
                        sum(c(active.cca$CCA$eig, active.cca$CA$eig)), 3) * 100
cca.explainvar2 <- round(active.cca$CCA$eig[2] /
                        sum(c(active.cca$CCA$eig, active.cca$CA$eig)), 3) * 100

# Define Plot Parameters
par(mar = c(5, 5, 4, 4) + 0.1)

# Initiate Plot
plot(scores(active.cca, display = "wa"), xlim = c(-3.5, 2), ylim = c(-3.2, 3.2),
```

```

xlab = paste("CCA 1 (", cca.explainvar1, "%)", sep = ""),
ylab = paste("CCA 2 (", cca.explainvar2, "%)", sep = ""),
pch = 16, cex = 2.0, type = "n", cex.lab = 1.5, cex.axis = 1.2, axes = FALSE)

# Add Axes
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)

# Add Points & Labels
points(scores(active.cca, display = "wa"),
       pch = 19, cex = 3, bg = "gray", col = "gray")
text(scores(active.cca, display = "wa"),
     labels = row.names(scores(active.cca, display = "wa"))))

# Add Environmental Vectors
vectors <- scores(active.cca, display = "bp")
row.names(vectors) <- c("diameter", "depth", "volume", "ORP", "temp", "SpC", "DO", "TDS", "salinity", "pH")

arrows(0, 0, vectors[,1] * 2, vectors[, 2] * 2,
       lwd = 2, lty = 1, length = 0.2, col = "red")
text(vectors[,1] * 2, vectors[, 2] * 2, pos = 3,
     labels = row.names(vectors))
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])))

```

