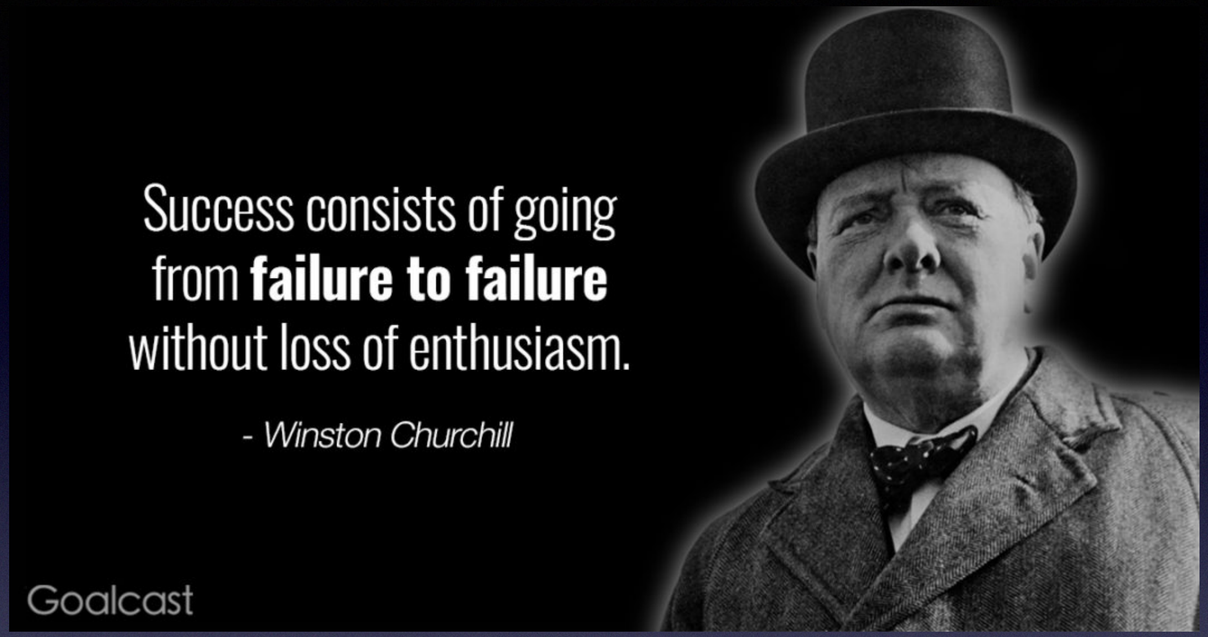
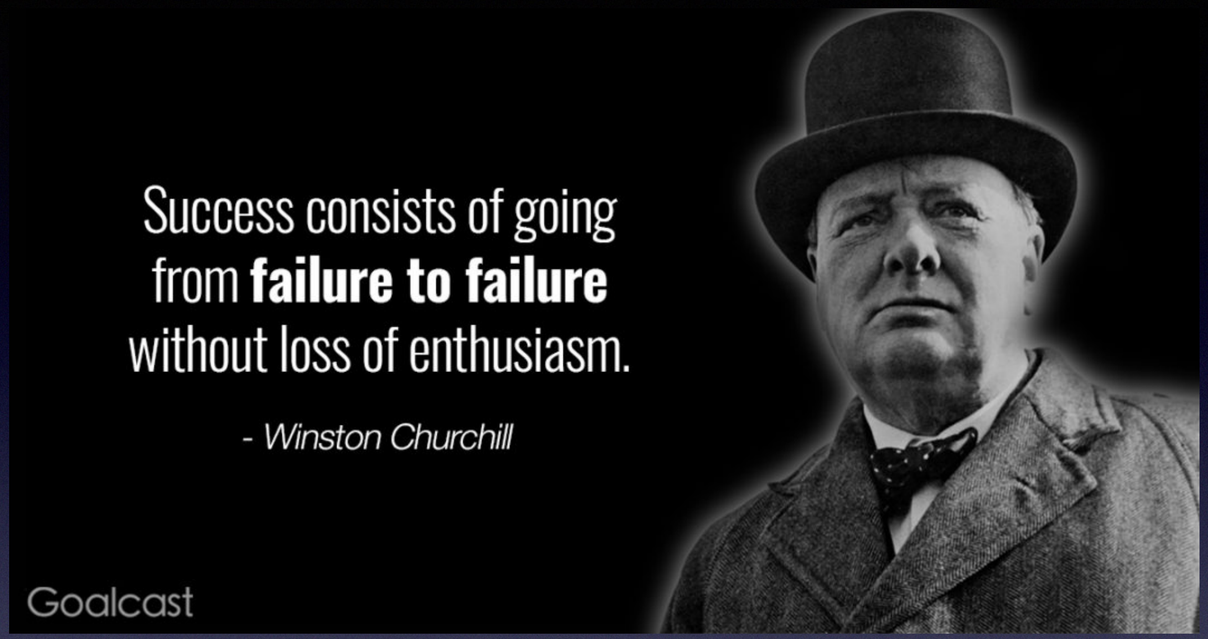
Community ecology and multivariate analyses

Ramiro Logares (ICM-CSIC, Barcelona)


Success consists of going
from **failure to failure**
without loss of enthusiasm.
- Winston Churchill
Goalcast



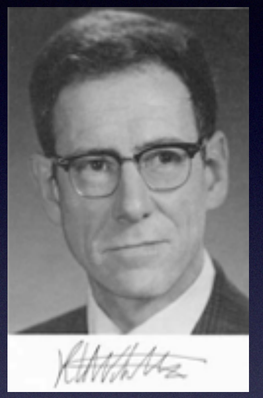
Success in bioinformatics consists of going from error to error without loss of enthusiasm.

Metabarcoding projects

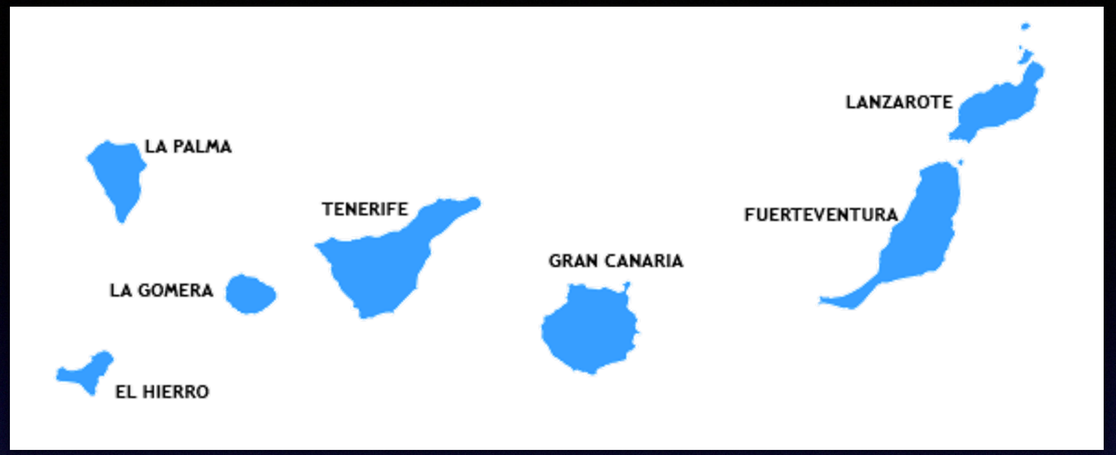
- 1. Sampling and wet-lab
- 2. Sequence processing
- 3. Ecological analyses : what questions do we want to answer?
    - Diversity analyses

Diversity

- Alpha
    - Richness: number of species in a location/sample
    - Evenness: relative species abundance in a location/sample
- Beta
    - Species turnover across locations/timepoints/samples
- Gamma
    - Species in all analysed locations/samples



Robert Whittaker

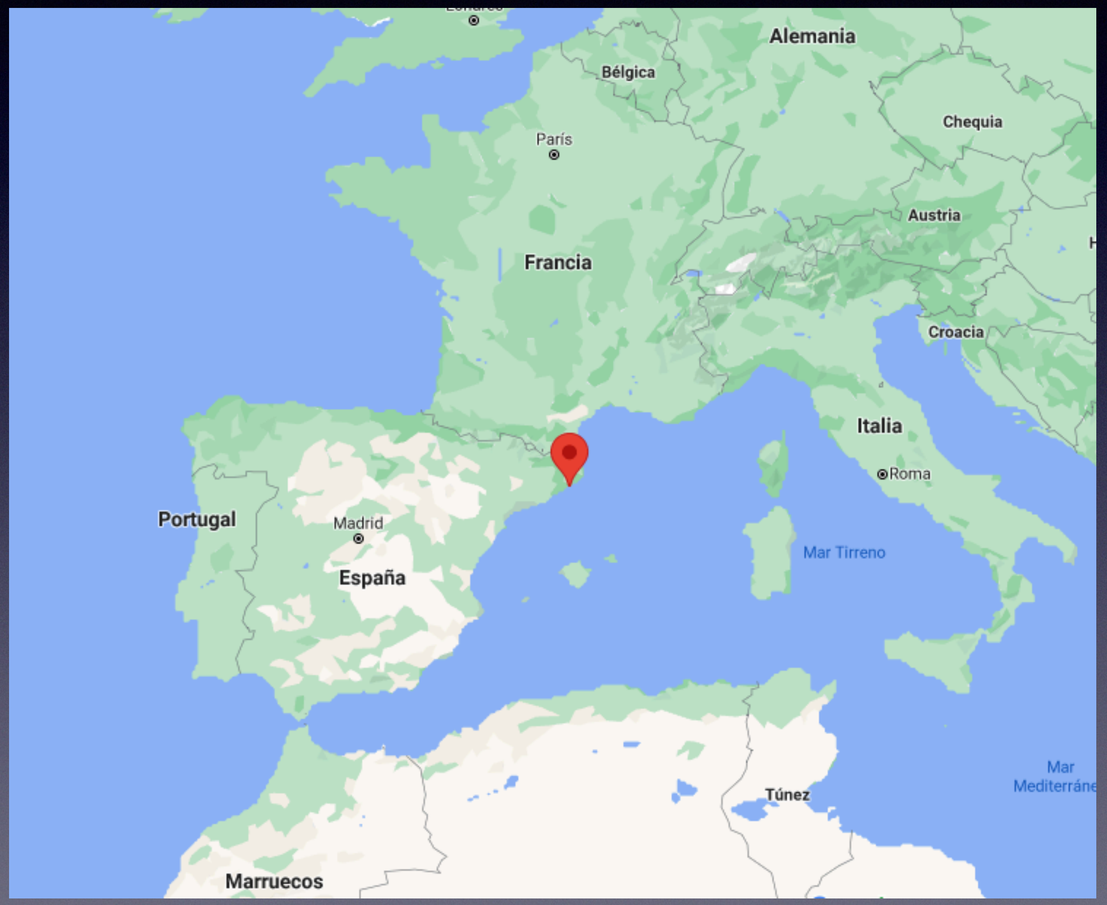


- Alpha diversity: number of species in each island
- Beta diversity: species change between islands
- Gamma diversity: species in all islands

Toy dataset

- 8 samples of the marine microbiome
  - Blanes Bay Microbial Observatory
- Community 18S rRNA gene
- 8 samples
  - January, April, July & October of 2004 and 2005

Blanes Bay Microbial Observatory



```r
#####################
## Community ecology
#####################

# Install packages (in case you didn't before)

install.packages("vegan")     # Community ecology functions
library(vegan)

# Read dada2 otuput

otu.tab<-read_tsv("https://raw.githubusercontent.com/krabberod/BIO9905MERG1_V21/main/Dada2_Pipeline/dada2_results/OTU_table.tsv")

head(otu.tab)
names(otu.tab)
dim(otu.tab) # 2107   26

#Let's reorder the table
otu.tab<-otu.tab[,c(17,19:26,1:16,18)]

#We assign to rownames the OTU names

otu.tab <- column_to_rownames(otu.tab, var = "OTUNumber") # %>% as_tibble()

rownames(otu.tab)
dim(otu.tab) # 2107   25

otu.tab.simple<-otu.tab[,1:8] # We'll need this table for community ecology analyses

#We transpose the table, as this is how Vegan likes it

otu.tab.simple<-t(otu.tab.simple)
otu.tab.simple[1:5,1:5]

#         OTU_00001 OTU_00002 OTU_00004 OTU_00005 OTU_00006
# BL040126      4996     12348     11426         0      3958
# BL040419       739       684        97     16605      4702
# BL040719         0         0       166         0       806
# BL041019        78        74         0       184       286
# BL050120     30697     12885      5417         0      3739

```

Alpha diversity

Number of species in specific samples/location

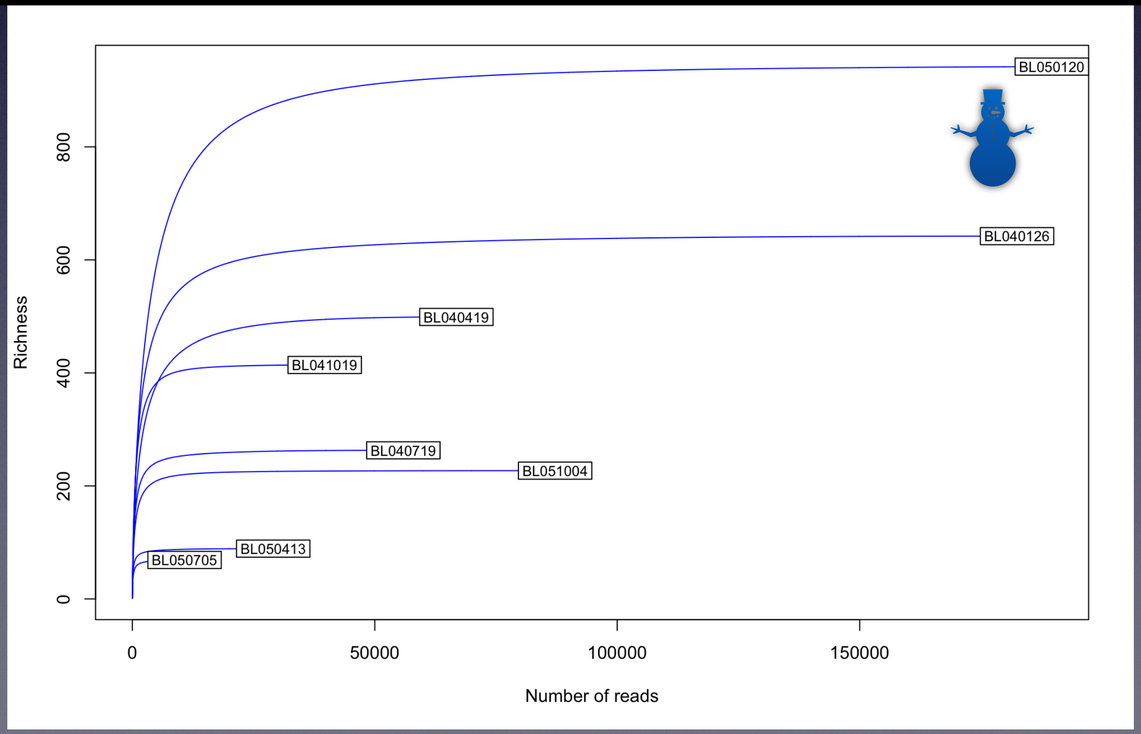
Richness estimates

```
 1 richness<-estimateR(otu.tab.simple)
 2
 3 #            BL040126   BL040419   BL040719   BL041019   BL050120  BL050413  BL050705   BL051004
 4 # S.obs    642.000000 499.000000 263.000000 414.000000 942.000000 89.000000 69.000000 227.000000
 5 # S.chao1  642.000000 499.000000 263.000000 414.000000 943.250000 89.000000 69.000000 227.000000
 6 # se.chao1   0.000000   0.000000   0.000000   0.000000   1.621617  0.000000  0.000000   0.000000
 7 # S.ACE    642.000000 499.000000 263.000000 414.000000 943.399653 89.000000 69.000000 227.000000
 8 # se.ACE     7.091415   9.573887   4.198497   6.011262  10.391615  2.539574  2.797514   2.604638
 9
10 # Above we have the estimators Chao and ACE as well as the species number.
11
```

Are we recovering all diversity?

Rarefaction

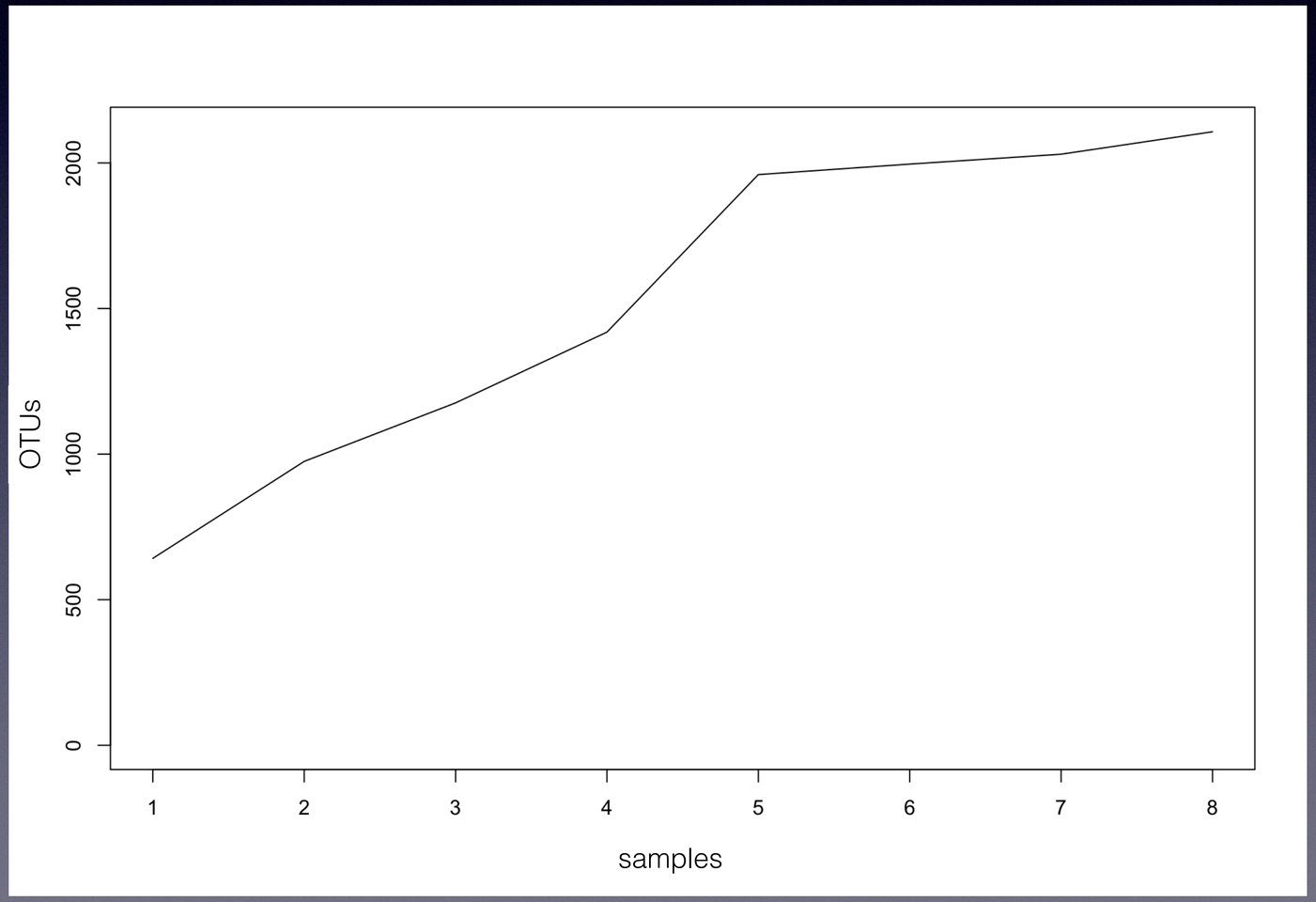
```
# Rarefaction

#Let's calculate the number of reads per sample

rowSums(otu.tab.simple)

# BL040126 BL040419 BL040719 BL041019 BL050120 BL050413 BL050705 BL051004
#   182462    66827    55896    39672   189636    29053    10771    87192


rarecurve (otu.tab.simple, step=100, xlab= "Number of reads", ylab="Richness", col="blue")

```



What are these results telling us?

Accumulation curves

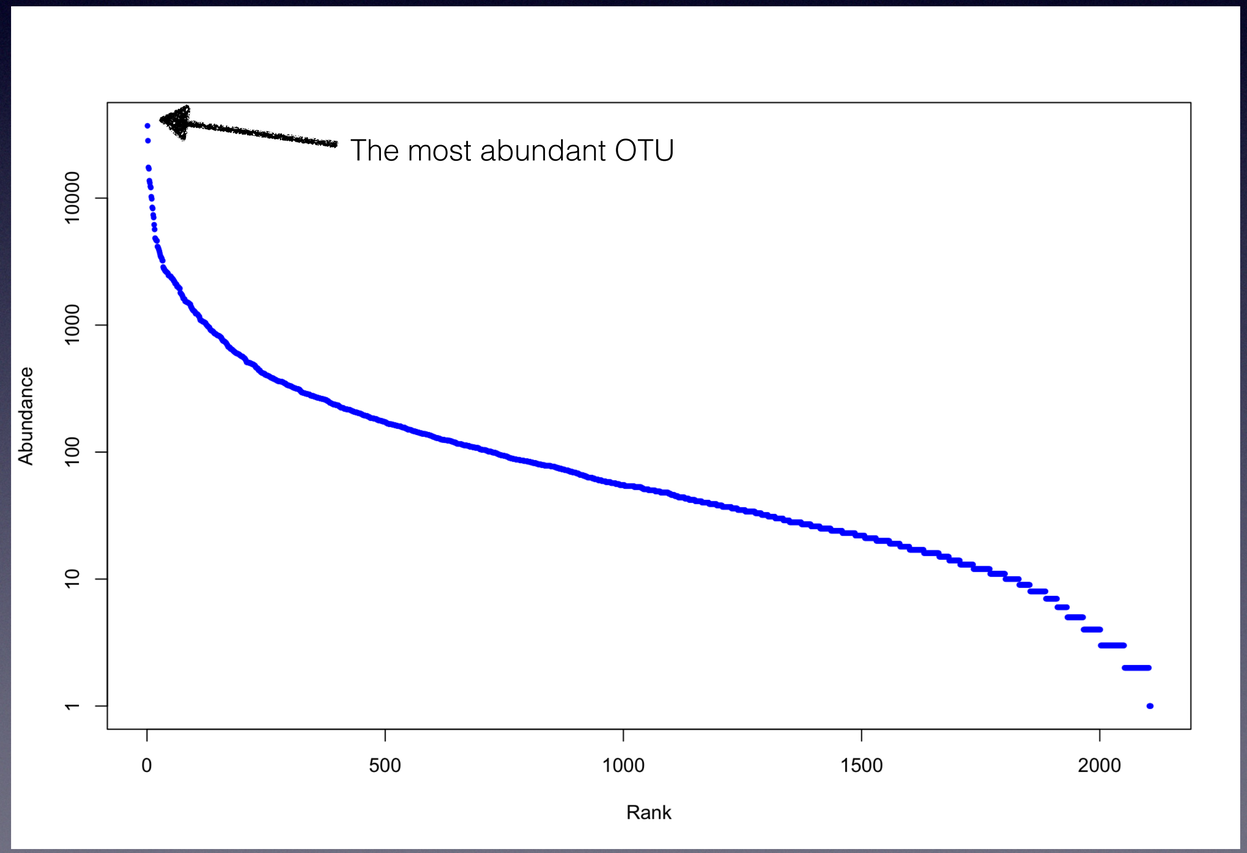
```
1 #Accumulation curves
2
3 accum.curve<-specaccum(otu.tab.simple, method="collector")
4 plot(accum.curve)
5
```



In this example, we want to know the increase of richness with the addition of new samples.

Evenness

```
1 #Evenness
2
3 plot(colSums(otu.tab.simple),log="y",xlab="Rank", ylab="Abundance", pch=19, cex=0.5, col="blue")
4
5
```



Few species highly abundant, while most species have a low abundance

Characteristic of microbiota
Why?

The Rare Bacterial Biosphere

Annual Review of Marine Science
Vol. 4:449-466 (Volume publication date January 2012)
First published online as a Review in Advance on September 19, 2011
https://doi.org/10.1146/annurev-marine-120710-100948

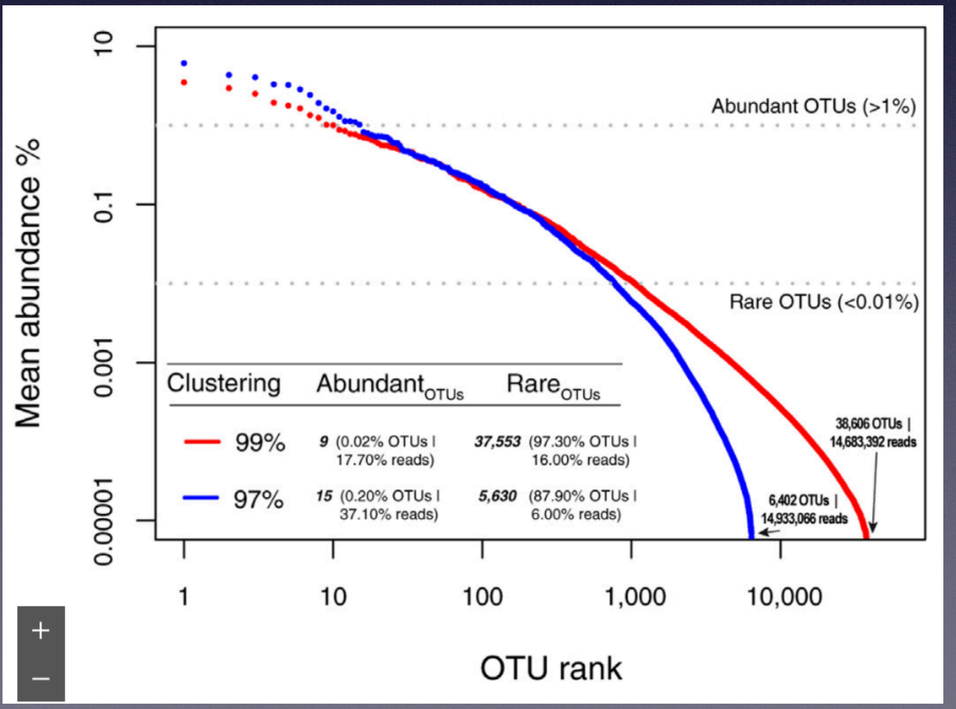
Carlos Pedrós-Alió
Institut de Ciències del Mar, CSIC, 08003 Barcelona, Spain; email: cpedros@icm.csic.es

ELSEVIER
Research in Microbiology
Volume 166, Issue 10, December 2015, Pages 831-841

Rarity in aquatic microbes: placing protists on the map

Ramiro Logares, Jean-François Mangot, Ramon Massana

Show more



Scaling laws predict global microbial diversity

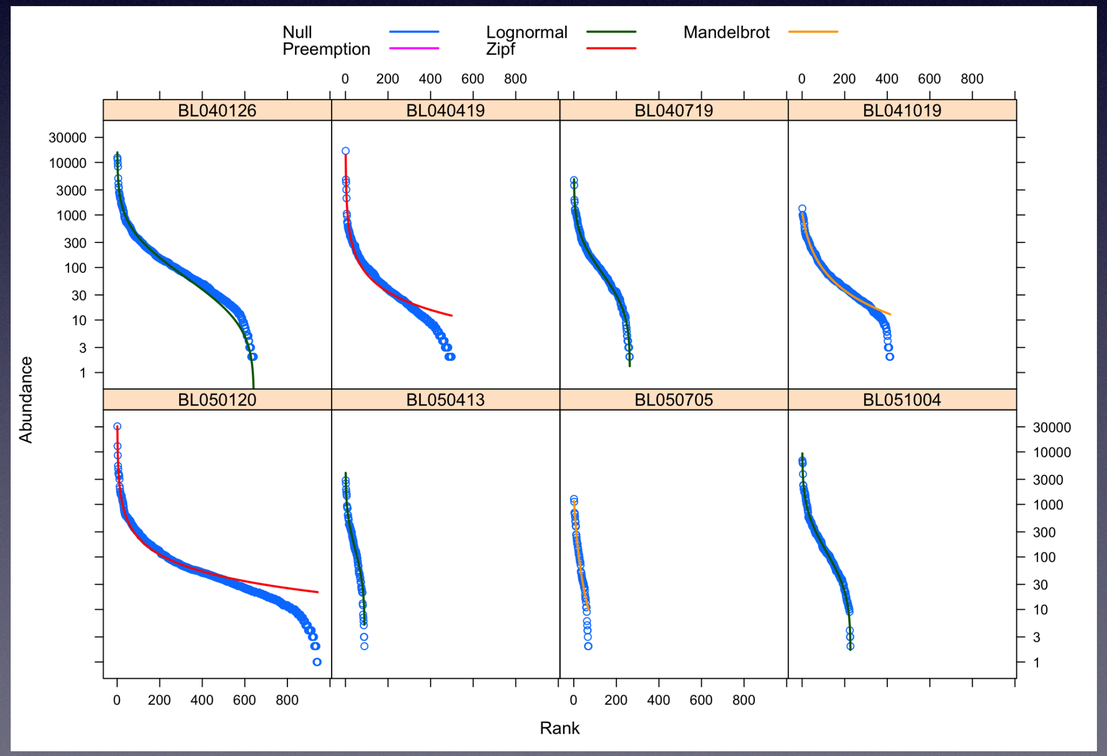
Kenneth J. Locey[a,1] and Jay T. Lennon[a,1]

[a]Department of Biology, Indiana University, Bloomington, IN 47405

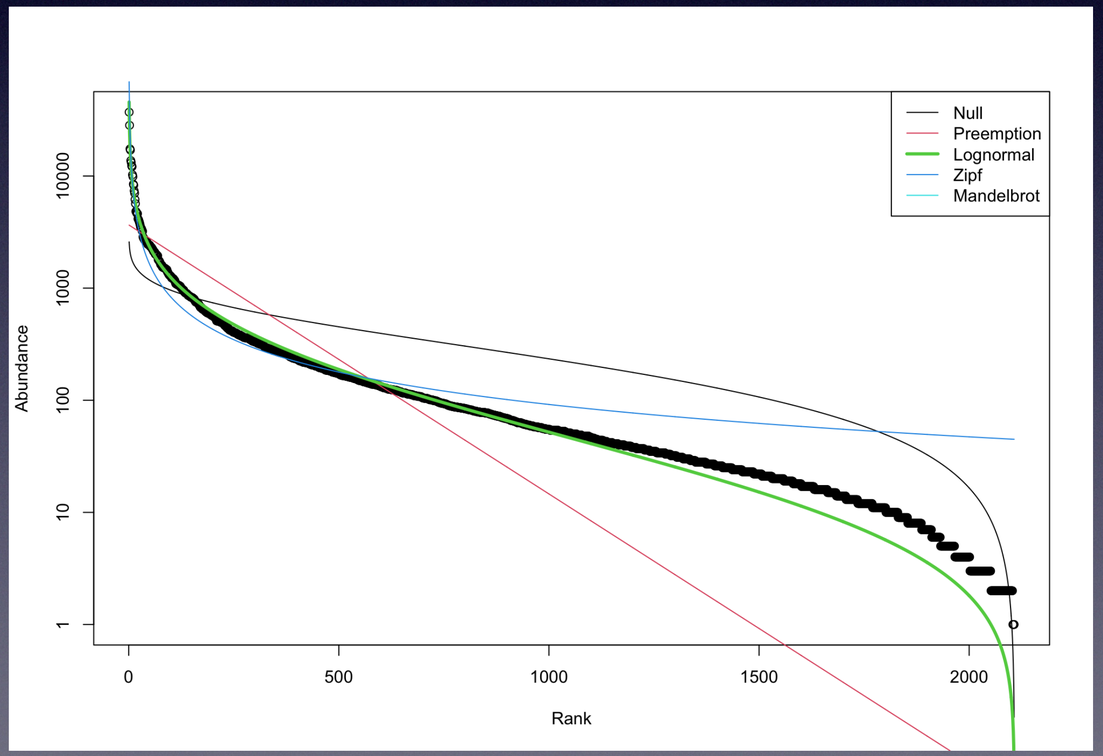
1 trillion (10^{12}) microbial species on Earth

Fitting rank-abundance distributions

```
1 #Fitting rank-abundance distribution models to the data
2
3 mod<-radfit(otu.tab.simple)
4 plot(mod)
5
6 mod.all<-radfit(colSums(otu.tab.simple))
7 plot(mod.all)
8
```



Best model is indicated

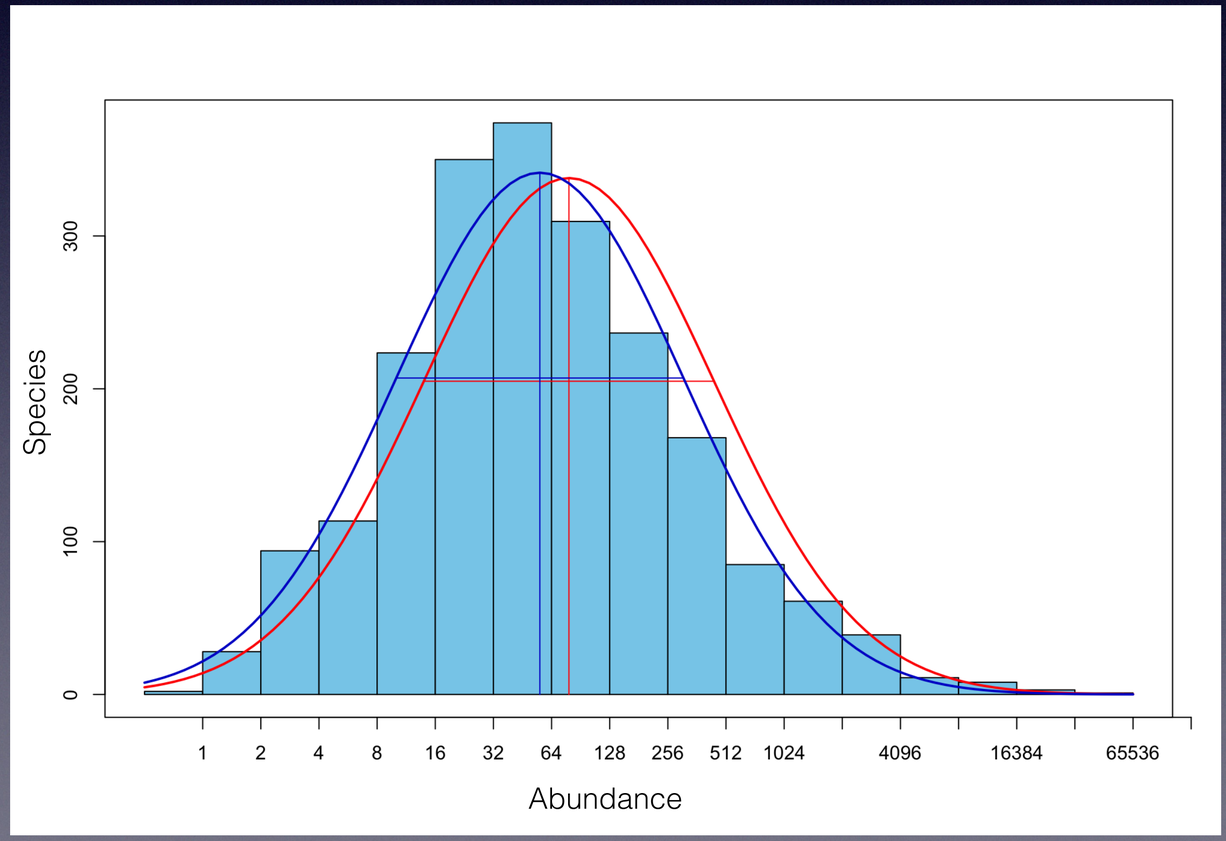


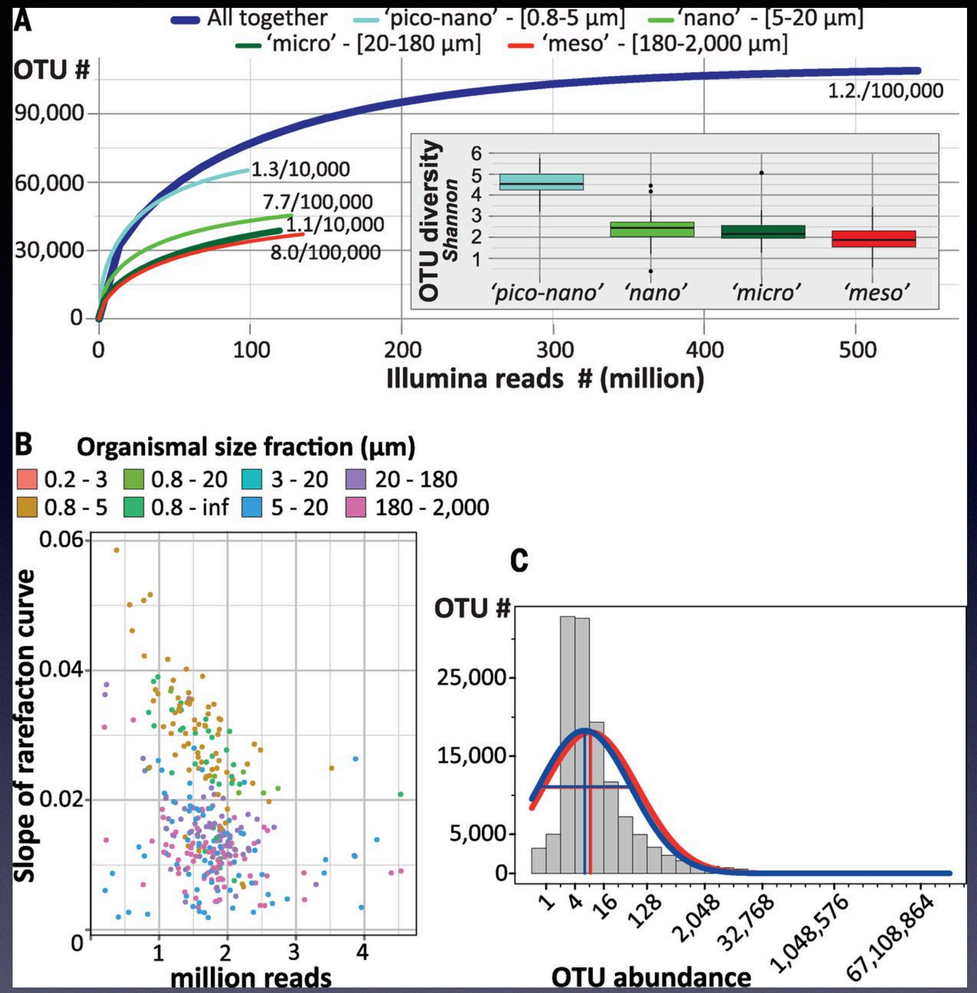
Best model is the thicker curve

Fitting to the Preston model

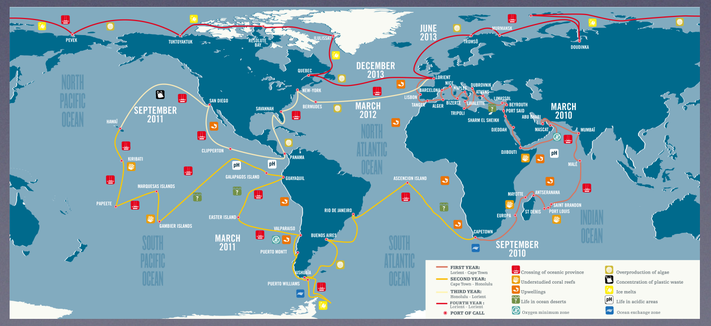
- Frank W. Preston (1948) proposed that species abundances (when binned logarithmically) follow a normal distribution.
- This leads to a lognormal abundance distribution.
- Before using ASVs, these plots were rarely observed for microbial data

```
#Fitting data to the Preston model

preston<-prestonfit(colSums(otu.tab.simple))
preston.dist<-prestondistr(colSums(otu.tab.simple))
plot(preston)
lines(preston.dist, line.col="blue3")

## Extrapolated richness
veiledspec(preston)
# Extrapolated     Observed       Veiled
# 2113.475329  2107.000000     6.475329

veiledspec(preston.dist)
# Extrapolated     Observed       Veiled
# 2113.236021  2107.000000     6.236021
```





18S V9 (Swarm)



RESEARCH ARTICLE

Eukaryotic plankton diversity in the sunlit ocean

Colomban de Vargas[1,2,*,†], Stéphane Audic[1,2,†], Nicolas Henry[1,2,†], Johan Decelle[1,2,†], Frédéric Mahé[3,1,2,†], Ramiro Logares…

+ See all authors and affiliations

Science 22 May 2015:
Vol. 348, Issue 6237, 1261605
DOI: 10.1126/science.1261605

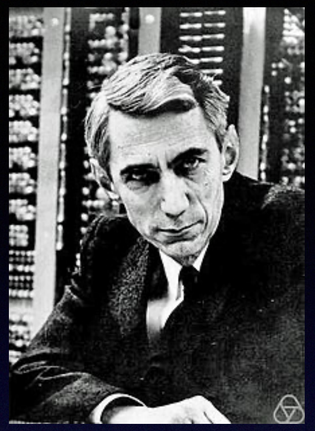
Shannon H index

-Considers richness and evenness

-Originally proposed by Claude Shannon in 1948 to quantify the entropy in strings of text.

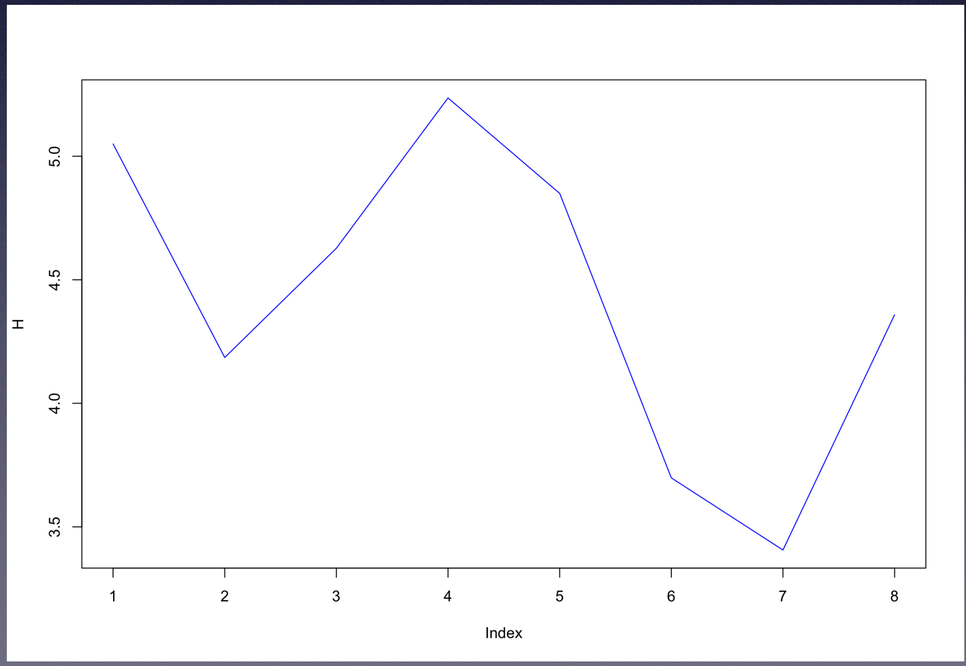
$$H' = -\sum_{i=1}^{R} p_i \ln p_i$$

p_i is the relative abundance of the ith species



Claude Shannon

```r
#Shannon H index (considers richness and evenness)

H<-diversity(otu.tab.simple, index="shannon")

# BL040126 BL040419 BL040719 BL041019 BL050120 BL050413 BL050705 BL051004
# 5.049747 4.185494 4.627698 5.236017 4.849669 3.698185 3.406164 4.358232

plot(H, type="l", col="blue")
```



Pielou's index of evenness

E.C. Pielou

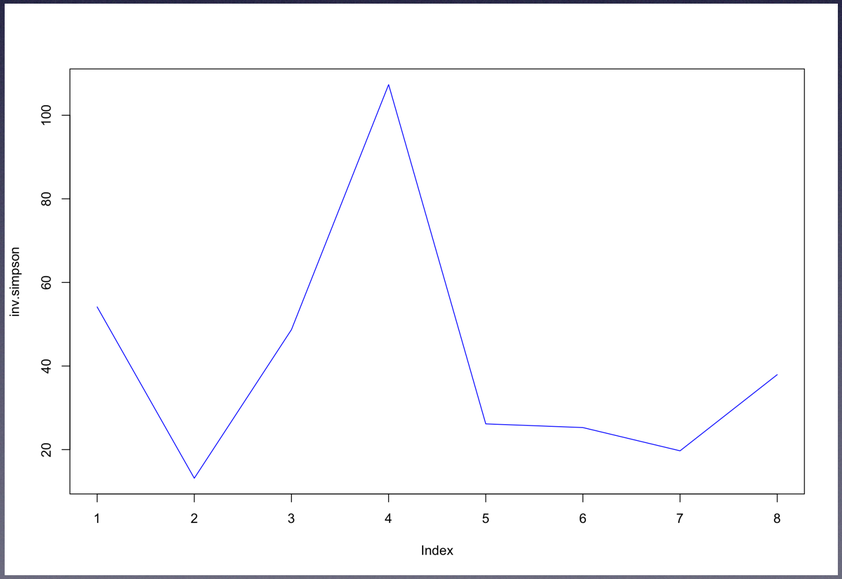
$$J' = \frac{H'}{H'_{\max}}$$

Inverse Simpson's *D* index

Edward Simpson

$$\frac{1}{\lambda} = \frac{1}{\sum_{i=1}^{R} p_i^2} = {}^2D$$

```
#Pielou's index of evenness (range 0-1, 1 = maximum evenness)
# J=H/Hmax
# J=Shannon (H) / log(S=species richness)

J=H/log(rowSums(otu.tab.simple>0))

#  BL040126  BL040419  BL040719  BL041019  BL050120  BL050413  BL050705  BL051004
# 0.7811398 0.6737098 0.8305043 0.8689236 0.7081871 0.8238995 0.8044587 0.8033681

# Inverse Simpson's D index (richness+evenness. Larger values, larger diversity)

inv.simpson<-diversity(otu.tab.simple, "invsimpson")
plot(inv.simpson, type="l", col="blue")

# BL040126  BL040419  BL040719  BL041019  BL050120  BL050413  BL050705  BL051004
# 54.13768  13.15796  48.69382 107.30411  26.16040  25.27907  19.71550  37.93128

```



Beta diversity

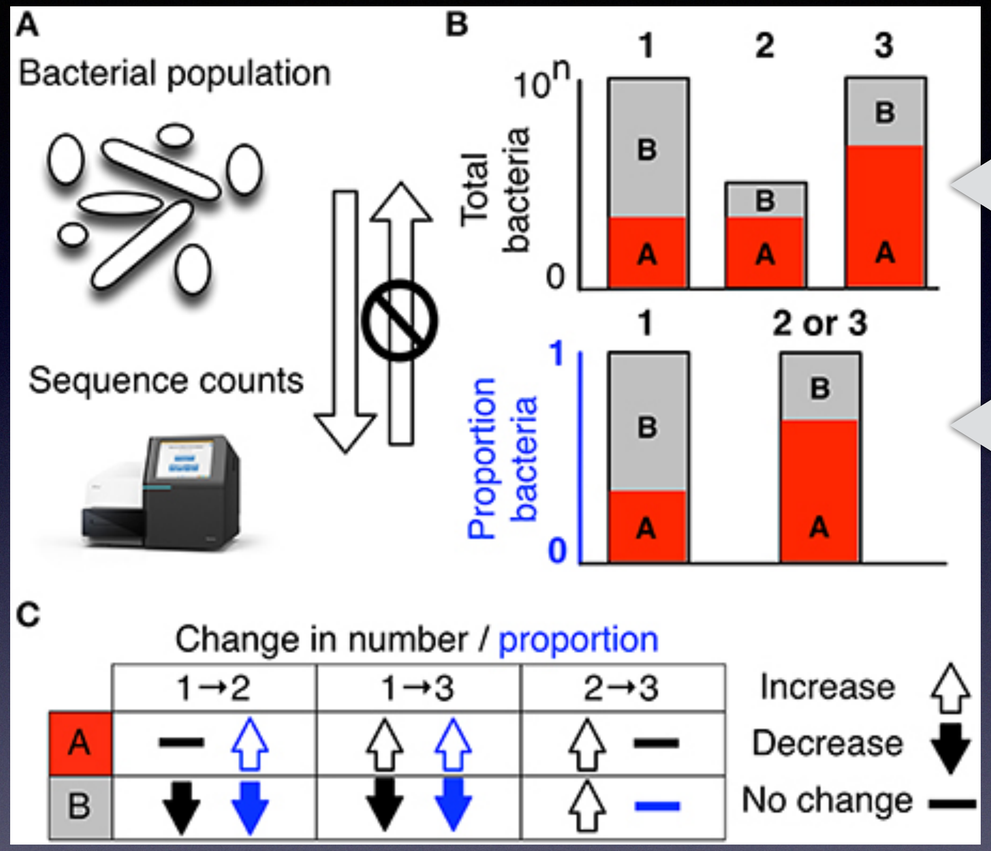
Species turnover between samples/location

- Beta diversity analyses will investigate how communities change over different samples (timepoints, locations, etc.)
- Analyses can be biased if different samples have different sequencing efforts
- HTS are compositional

HTS data are compositional

- HTS datasets are compositional, due to the total limits imposed by sequencers. Then, the increase of one OTU means the decrease of another OTU in the HTS dataset.
- Total read count in a HTS run is a fixed-size, random sample of the relative abundance of the molecules in the underlying ecosystem
- The count can not be related to the absolute number of molecules in the input sample
- This is implicitly acknowledged when microbiome datasets are converted to relative abundance values, or normalized counts, or are rarefied
- Data described as proportions or probabilities, or with a constant or irrelevant sum, are referred to as compositional data
- Compositional data is all about the relationships between the parts (can’t inform on absolute abundances of molecules)
- The abundance of one OTU is only interpretable relative to another

Real vs. perceived change



Absolute abundance before sequencing

Relative abundance after sequencing

After samples are sequenced we lose the absolute count information and only have relative abundances, proportions, or “normalised counts”

Example (panel B & C in figure):

-The absolute number of species A is the same in sample 1 and 2

-The proportional number of species A increases from 1 to 2


frontiers in Microbiology

MINI REVIEW
published: 15 November 2017
doi: 10.3389/fmicb.2017.02224

Microbiome Datasets Are Compositional: And This Is Not Optional

Gregory B. Gloor[1], Jean M. Macklaim[1], Vera Pawlowsky-Glahn[2] and Juan J. Egozcue[3]*

[1] Department of Biochemistry, University of Western Ontario, London, ON, Canada, [2] Departments of Computer Science, Applied Mathematics, and Statistics, Universitat de Girona, Girona, Spain, [3] Department of Applied Mathematics, Universitat Politècnica de Catalunya, Barcelona, Spain

- Different sequencing depths may bias the calculation of distances for multivariate analyses
    - One way to mitigate this is to subsample or “rarefy” samples to the same sequencing depth
        - But, it has been criticised due to loss of information and precision
    - Anyways, let’s try rarefying the samples to the same sequencing depth

```r
#We rarefy all samples to the same sequencing depth, to reduce biases
min(rowSums(otu.tab.simple)) # We calculate the sample with the minimum amount of reads
# [1] 10771

otu.tab.simple.ss<-rrarefy(otu.tab.simple, 10771) #Samples are rarefied to 10771 reads per sample

rowSums(otu.tab.simple.ss) # We check the number of reads per sample
# BL040126 BL040419 BL040719 BL041019 BL050120 BL050413 BL050705 BL051004
#  10771    10771    10771    10771    10771    10771    10771    10771

#Check the dimensions of the tables
dim(otu.tab.simple)
# [1]    8 2107
dim(otu.tab.simple.ss)
# [1]    8 2107

#Tables have the same size, but, after removing reads, several OTUs are left with cero abundances
length(which(colSums(otu.tab.simple)==0))
# [1] 0 #No OTU has an abundance sum that is 0, as expected

length(which(colSums(otu.tab.simple.ss)==0))
#  [1] 273 # A total of 273 OTUs were found in the rarefied table with cero abundance. Let's corroborate

which(colSums(otu.tab.simple.ss)==0) # Show the OTUs and the position in the table that have 0 abundance
# A small subsample of them
# OTU_00814 OTU_01076 OTU_01077 OTU_01232 OTU_01242
#    772      1020      1021      1166      1176

otu.tab.simple[,772] # This gives the abundance of the OTU_00814 across the different samples in the table that is NOT subsampled
# BL040126 BL040419 BL040719 BL041019 BL050120 BL050413 BL050705 BL051004
#    0        0        0        0       88        0        0        0

otu.tab.simple.ss[,772] # # This gives the abundance of the OTU_00814 across the different samples in the table that IS subsampled
# BL040126 BL040419 BL040719 BL041019 BL050120 BL050413 BL050705 BL051004
# 0        0        0        0        0        0        0        0

otu.tab.simple.ss.nocero<-otu.tab.simple.ss[,-(which(colSums(otu.tab.simple.ss)==0))] # Removes OTUs with cero abundance
length(which(colSums(otu.tab.simple.ss.nocero)==0)) # Check that no cero abundance OTUs are left
# [1] 0 # correct
# Let's check dimensions
dim(otu.tab.simple.ss)
# [1]    8 2107
dim(otu.tab.simple.ss.nocero)
# [1]    8 1834
# 2107-1834 = 273 , This is the number of OTUs that we expected to be removed.

```

Compositional analyses

- Ratio transformation of the data: captures relationships between features (e.g. OTUs)
- Taking the logarithm of the ratios (log-ratios) makes data symmetric and linearly related
- We will use the *centered log-ratio (clr)* transformation
- Since we use logarithms, zeros need to be replaced before clr transformation
  - There are different methods, we will use a Bayesian multiplicative replacement generating pseudo-counts
- In a composition, all components (OTUs) are mutually dependent, and can not be understood in isolation
- Analyses of individual components (OTUs) is done with respect to a reference
- The clr transformation uses the geometric mean as a reference

clr transformation

Given a vector of *D* “counted” OTUs in a sample x $\mathbf{x} = [x_1, x_2, \ldots x_D]$

The clr transformation for the sample x is calculated as

$$\mathbf{x}_{clr} = [log\,(x_1/G\,(\mathbf{x}))\,, log\,(x_2/G\,(\mathbf{x}))\,\ldots\,log\,(x_D/G\,(\mathbf{x}))]\,,$$
$$G\,(\mathbf{x}) = \sqrt[D]{x_1 \cdot x_2 \cdot \ldots \cdot x_D} \qquad (1)$$

With *G*(x) being the geometric mean of x

-clr will indicate how OTUs behave relative to the per-sample average

-clr values can be used as inputs for multivariate analyses

-clr values are scale invariant: same ratios are expected independently of the number of reads per sample

```
### Compositional data analyses
# We install packages to work with compositional data
install.packages("compositions")
install.packages("zCompositions")
library(compositions)
library(zCompositions)

otu.tab.simple.gbm<-cmultRepl(t(otu.tab.simple), output = "p-counts")  # replace zeros (problems with log calculations) with pseudo-counts
# No. corrected values:  12246
otu.tab.simple.gbm[1:5,1:5] # We have a look to the replaced values
#              BL040126 BL040419    BL040719    BL041019     BL050120
# OTU_00001  4996.000000      739   0.9810100  78.0000000 30697.000000
# OTU_00002 12348.000000      684   0.9744656  74.0000000 12885.000000
# OTU_00004 11426.000000       97 166.0000000   0.6851427  5417.000000
# OTU_00005     3.229364    16605   0.9892938 184.0000000     3.356335
# OTU_00006  3958.000000     4702 806.0000000 286.0000000  3739.000000

# centered log-ratio (clr) transformation
otu.tab.simple.gbm.clr<-clr(otu.tab.simple.gbm) # We apply a centered log-ratio (clr) transformation
otu.tab.simple.gbm.clr[1:5,1:5] #Values now look different than counts.
# clr values indicate how OTUs behave relative to the per-sample average
#           BL040126   BL040419   BL040719    BL041019  BL050120
# OTU_00001  3.016847 1.10575200 -5.5187186 -1.14283710  4.832374
# OTU_00002  5.034361 2.14106914 -4.4127548 -0.08282368  5.076930
# OTU_00004  4.818212 0.04927624  0.5865531 -4.90356291  4.071863
# OTU_00005 -2.082582 6.46259197 -3.2656311  1.96006859 -2.044017
# OTU_00006  3.237162 3.40941121  1.6457517  0.60965979  3.180241

```

What do the clr values tell us about OTU 00001 in sample 1 vs sample 3?

- The rarefaction approach has been widely used
- Now, clr transformations are becoming more popular
- In the following slides we will try to compare how the two approaches behave in standard multivariate analyses

Distance metrics

- Statistical distance: distance between variables
- *Distance metrics in ecology: allow measuring the dissimilarity between communities composed by several species (OTUs)*
- Several distance metrics available in R
- Often used: Bray Curtis, Euclidean, Jaccard, Sorensen, Simpson

```
1 Distance metrics available in Vegan
2 "manhattan", "euclidean", "canberra", "clark", "bray", "kulczynski", "jaccard", "gower",
3 "altGower", "morisita", "horn", "mountford", "raup", "binomial", "chao", "cao", "mahalanobis",
4 "chisq" or "chord".
5
```

- It is important to think what is the most appropriate distance metric for the data being analysed
- Different distance metrics can have different ranges
- Bray-Curtis: influenced by abundant taxa (but normally used)
  - Ranges between 0-1 (1 most dissimilar)
- Euclidean and Sorensen: influenced by large differences in species abundances, data sparsity (lots of zeros) and many observations
  - Euclidean: no upper limit of values

More info: https://www.davidzeleny.net/anadat-r/doku.php/en:similarity

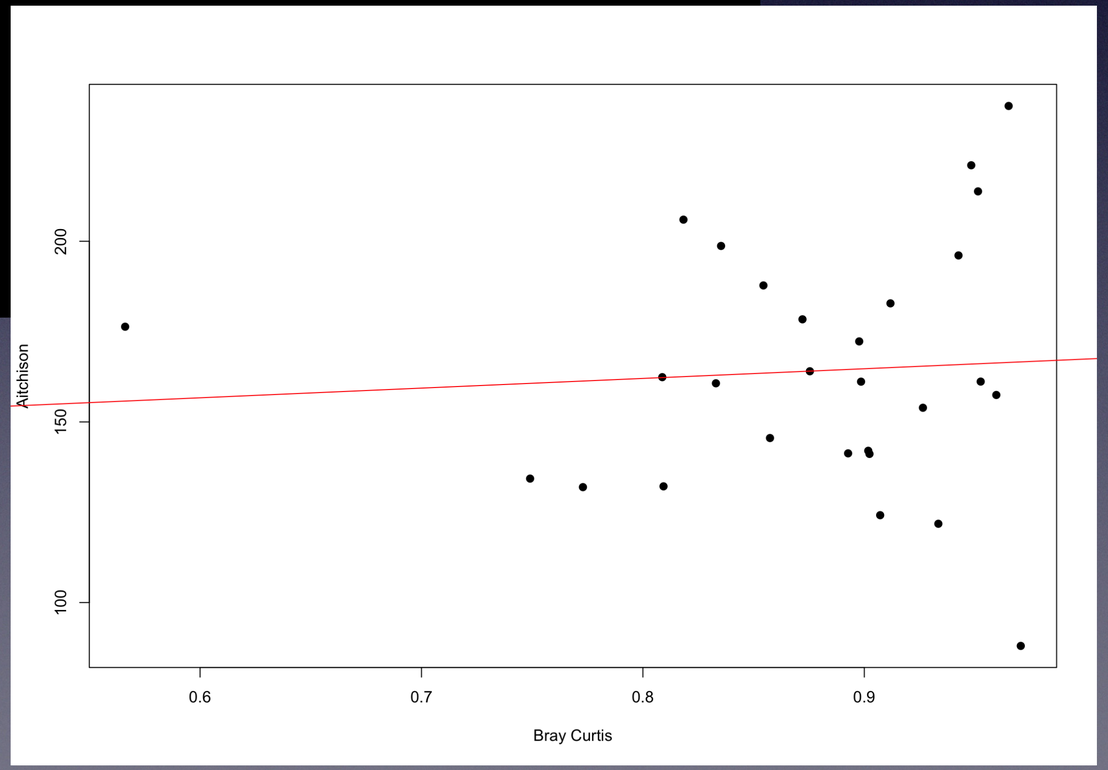
Bray Curtis & Aitchison* distances for the rarefied and clr datasets

```
# Distance metrics
# We calculate the Bray Curtis dissimilarities for the rarefied dataset
otu.tab.simple.ss.nozero.bray<-vegdist(otu.tab.simple.ss.nozero, method="bray")
as.matrix(otu.tab.simple.ss.nozero.bray)[1:5,1:5]
#           BL040126  BL040419  BL040719  BL041019  BL050120
# BL040126 0.0000000 0.8087457 0.9264692 0.8720639 0.5661498
# BL040419 0.8087457 0.0000000 0.9017733 0.8754062 0.8352985
# BL040719 0.9264692 0.9017733 0.0000000 0.7490484 0.9118002
# BL041019 0.8720639 0.8754062 0.7490484 0.0000000 0.8183084
# BL050120 0.5661498 0.8352985 0.9118002 0.8183084 0.0000000


# We calculate the Euclidean distance based on the clr data (also known as Aitchison distance)
otu.tab.simple.gbm.clr.euclidean<-dist(t(otu.tab.simple.gbm.clr), method = "euclidean")
as.matrix(otu.tab.simple.gbm.clr.euclidean)[1:5,1:5]
#          BL040126 BL040419 BL040719 BL041019 BL050120
# BL040126   0.0000 162.3852 153.9187 178.3993 176.3504
# BL040419 162.3852   0.0000 142.0086 164.0183 198.7059
# BL040719 153.9187 142.0086   0.0000 134.3082 182.8134
# BL041019 178.3993 164.0183 134.3082   0.0000 205.9812
# BL050120 176.3504 198.7059 182.8134 205.9812   0.0000
```

*Aitchison distance = euclidean distance of clr-transformed values

Comparing both distance matrices (Bray Curtis vs. Aitchison)

```r
#Let's compare the distance matrices
identical(rownames(as.matrix(otu.tab.simple.ss.nozero.bray)),rownames(as.matrix(otu.tab.simple.gbm.clr.euclidean))) # Check matrices have same
          order
# [1] TRUE

plot(otu.tab.simple.ss.nozero.bray, otu.tab.simple.gbm.clr.euclidean, pch=19) # We generate a simple x-y plot
lm<-lm(otu.tab.simple.gbm.clr.euclidean~otu.tab.simple.ss.nozero.bray) # Fitting a linear model (regression)
abline(lm, col="red")

mantel(otu.tab.simple.ss.nozero.bray, otu.tab.simple.gbm.clr.euclidean) # The correlation between distance matrices is tested with a Mantel test
# Mantel statistic based on Pearson's product-moment correlation

# Call:
# mantel(xdis = otu.tab.simple.ss.nozero.bray, ydis = otu.tab.simple.gbm.clr.euclidean)

# Mantel statistic r: 0.06774
#       Significance: 0.346  # Correlations between distances matrices is not significant

# Upper quantiles of permutations (null model):
#  90%   95% 97.5%   99%
# 0.280 0.355 0.419 0.462
# Permutation: free
# Number of permutations: 999
```



Ordination

- Is a collective term for multivariate techniques which summarise a multidimensional dataset in such a way that when it is projected onto a two dimensional space any intrinsic pattern the data may possess becomes apparent upon visual inspection (Pielou, 1984)

- In ecological terms: ordination serves to summarise community data (such as species abundance data) by producing a low-dimensional ordination space in which *similar species and samples are plotted close together, and dissimilar species and samples are placed far apart*.

- Ordination is used in ecology to investigate relationships between species composition patterns and environmental variability. Many times, these techniques are used to address the question: what environmental variables shape communities?

- The relative importance of environmental gradients in shaping communities can be estimated

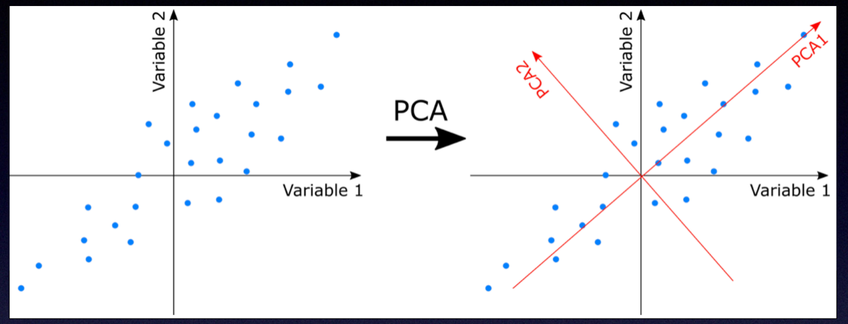
Ordination approaches

Main unconstrained techniques

- Principal Component Analysis (PCA)
- Principal Coordinate Analysis (PCoA)
- Non-metric Multidimensional Scaling (NMDS)

PCA

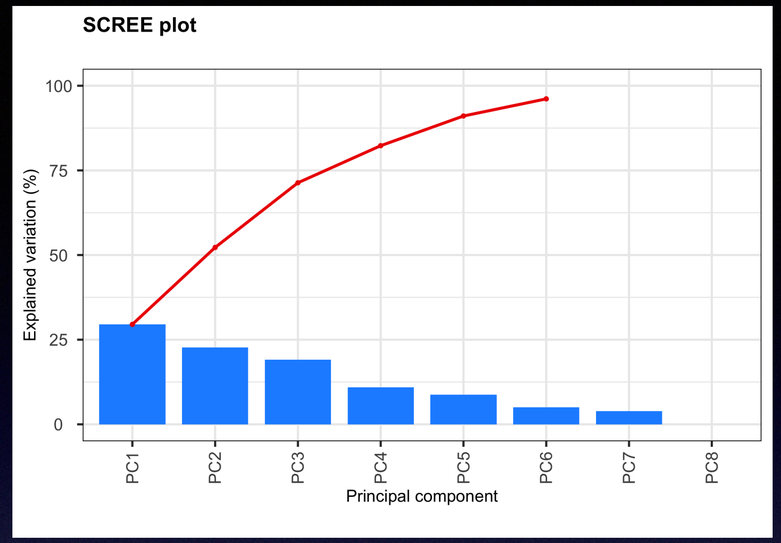
Rotates the original axes in order to maximise the 2D variability. The first principal component (PC) will be placed in the direction of the maximum variability and subsequent PCs will be generated in the same manner



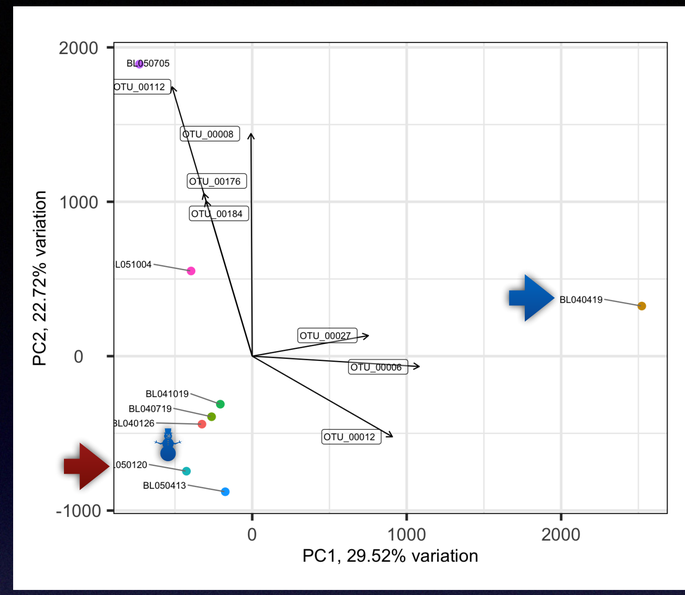

```r
#Ordination and clustering

#PCA

# We install PCAtools
if (!requireNamespace('BiocManager', quietly = TRUE))
  install.packages('BiocManager')

BiocManager::install('PCAtools')

library(PCAtools)

#PCA rarefied table
otu.tab.simple.ss.nozero.pca<-pca(t(otu.tab.simple.ss.nozero), scale=FALSE) # Runs de PCA
biplot(otu.tab.simple.ss.nozero.pca, showLoadings = T, lab=rownames(otu.tab.simple.ss.nozero)) # Plots de PCA
screeplot(otu.tab.simple.ss.nozero.pca, axisLabSize = 18, titleLabSize = 22) # We plot the percentage of variance explained by each axis

#We install mixOmics
install.packages("mixOmics") # We change the package mixOmics, as PCAtools had some issues with clr tables
library(mixOmics)
#PCA clr table (calculated with the vegan "rda" function, as pca from PCAtools gives errors
otu.tab.simple.gbm.clr.pca<-pca(otu.tab.simple.gbm.clr, scale=FALSE, ncomp=6) # NB: the pca used here is from "mixOmics" while the pca above is
                              from "PCAtools"
plotVar(otu.tab.simple.gbm.clr.pca)
plot(otu.tab.simple.gbm.clr.pca)
```

PCA

Rarefied dataset

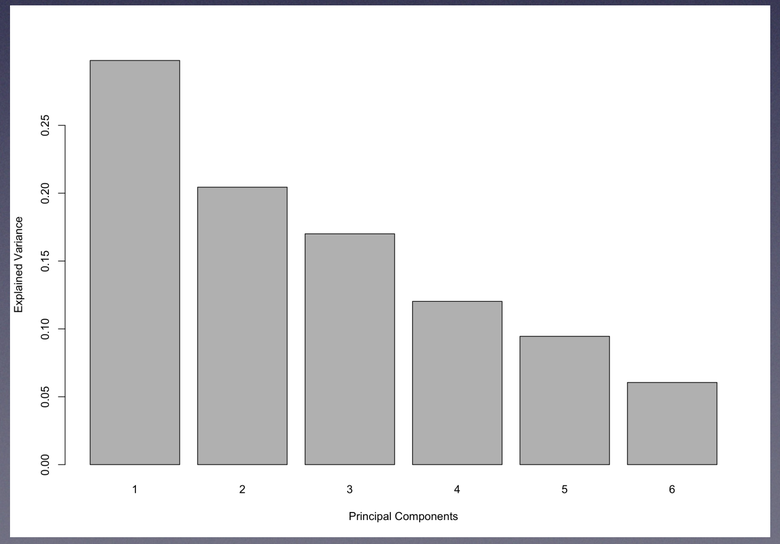


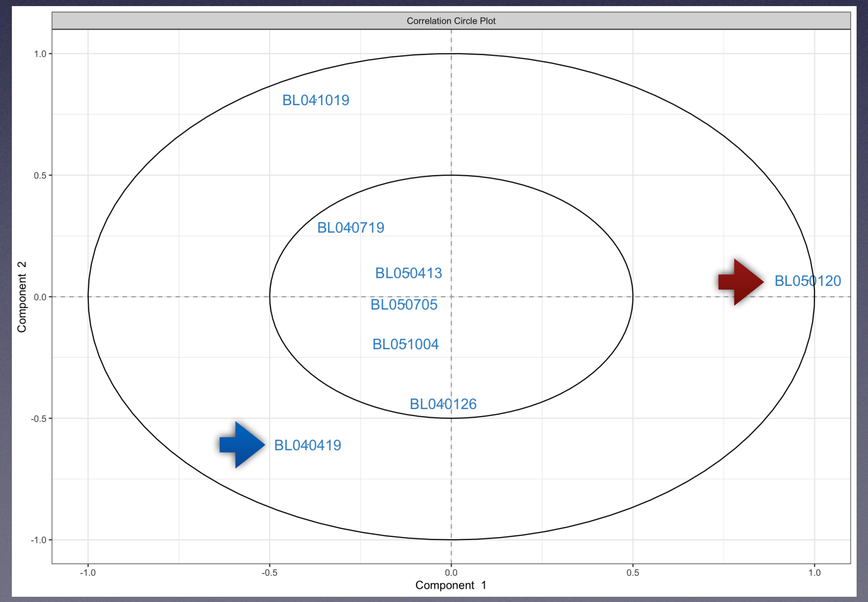
Percentage of variance explain by each PC



Samples and OTUs are plotted. The arrows indicate the weight of each OTU in the different directions

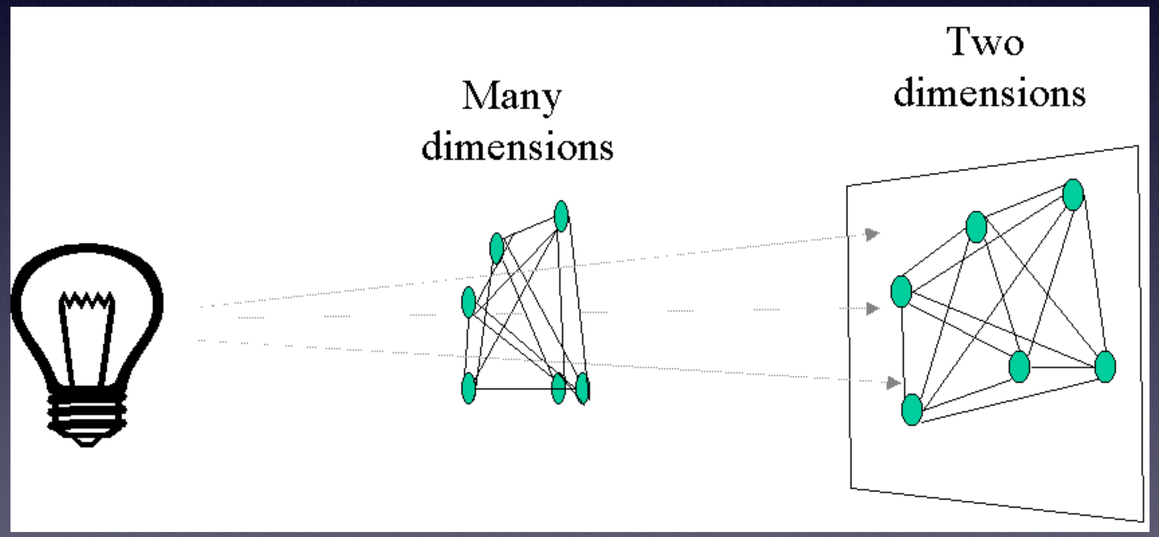
clr dataset





Principal Coordinate Analysis (PCoA)

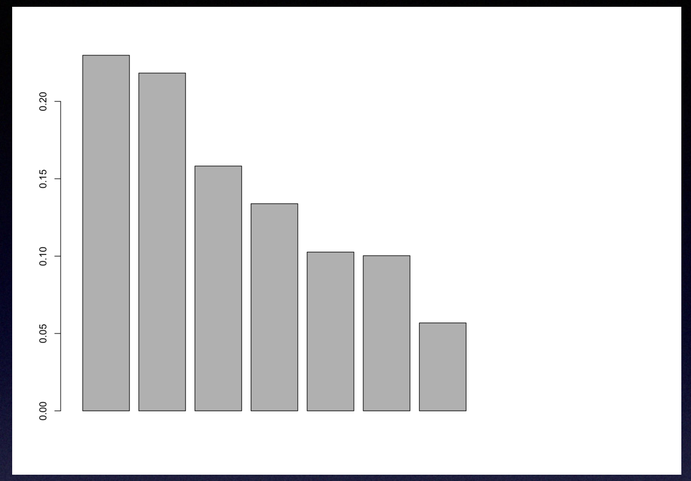
- Attempts to represent distances between samples in a low dimensional Euclidean space
- It maximises the linear correlation between the distances in the distance matrix, and the distances in a space of low dimension (typically, 2 or 3 axes are selected)
- The PCoA algorithm is analogous to rotating the multidimensional object such that the distances (lines) in the shadow are maximally correlated with the distances (connections) in the object

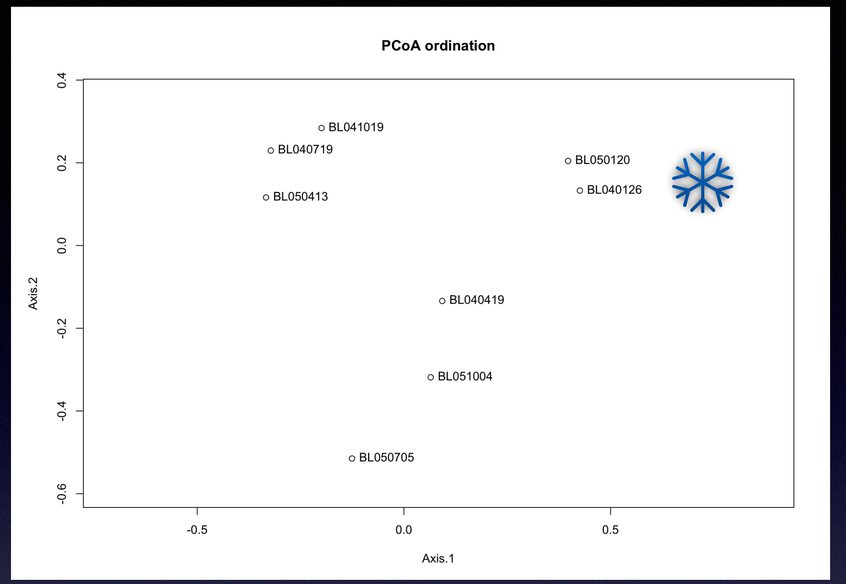


- PCoA requires a distance matrix (typically, Bray Curtis for abundance data or Jaccard for presence-absence)
- When using an Euclidean distance matrix, PCoA is equivalent to PCA

Principal Coordinate Analysis (PCoA)

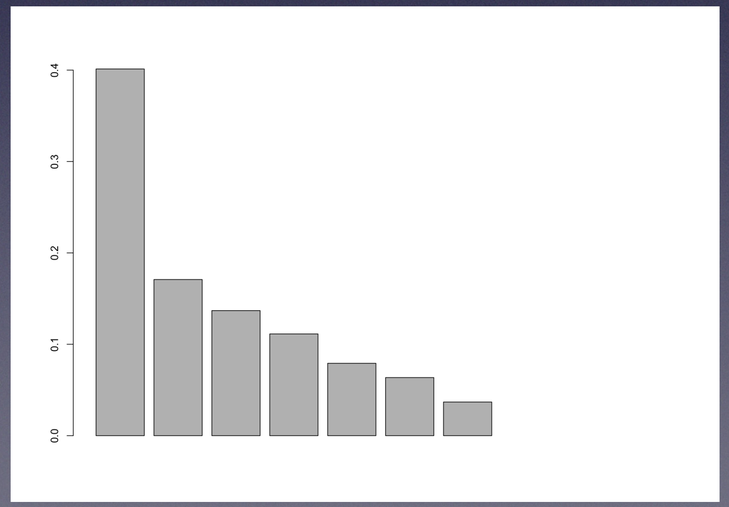
Rarefied dataset + Bray Curtis

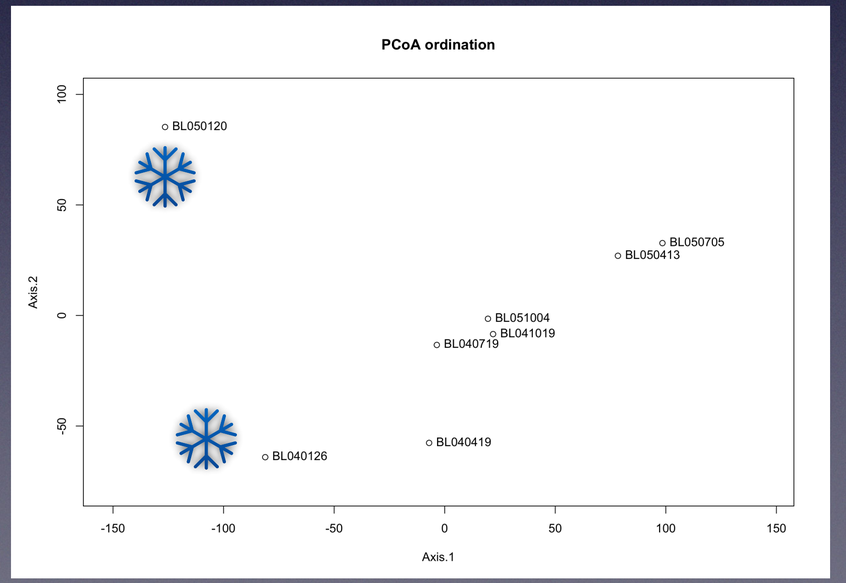




Percentage of variance explained by each axis

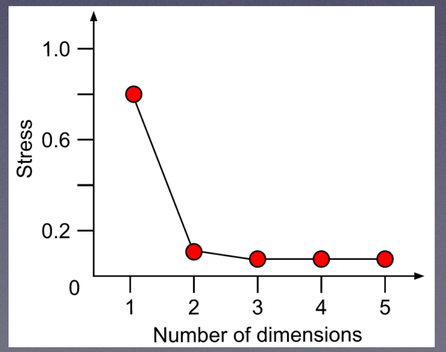
clr dataset + Aitchison





Non-metric Multidimensional Scaling (NMDS)

- NMDS is more robust than PCoA (e.g. is not affected by the arch effect)
- NMDS attempts to represent the pairwise dissimilarity between objects in a low-dimensional space
- Any distance metric can be used to build the distance matrix
- NMDS is a rank approach, meaning that distances are replaced by ranks
- The stress value indicates how well the ordination summarises the observed distances among the samples
- NMDS differs from PCA and PCoA in that:
  - There is not a unique ordination result (thus, algorithms run NMDS multiple times)
  - The axes of the ordination are not ordered according to the variance they explain (but metaMDS() in Vegan rotates final results to make Axis 1 correspond to the greatest variance among samples)
  - The number of dimensions of the low-dimensional space must be specified before running NMDS
    - Plotting stress (goodness of fit) vs. dimensionality can be used to assess the choice of dimensions. Stress values should be <0.2. We choose the minimum number of dimensions

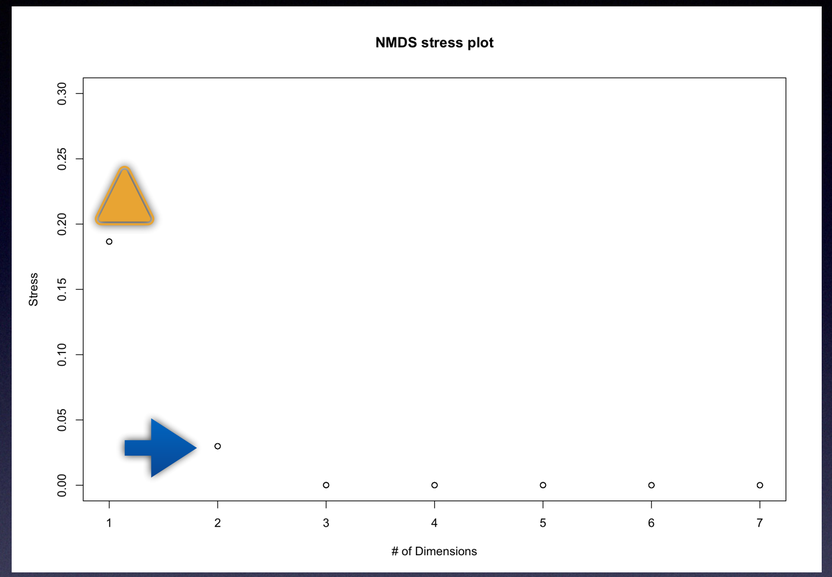


Calculating NMDS

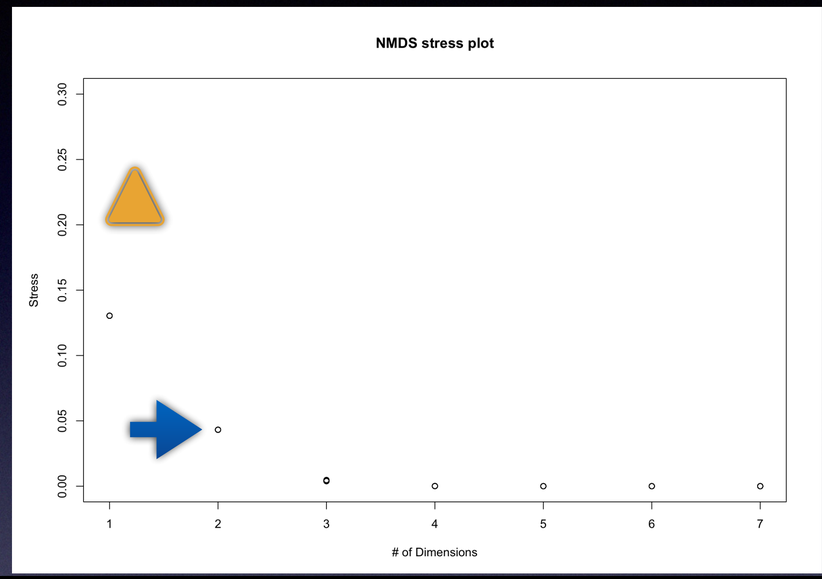
- Step1: run NMDS with e.g. 1 to 10 dimensions
- Step2: Check stress vs. dimension plot
- Step3: Choose optimal number of dimensions (typically k=2)
- Step5: Check for convergent solution and final stress

NMDS stress vs. dimensions plot

Rarefied table



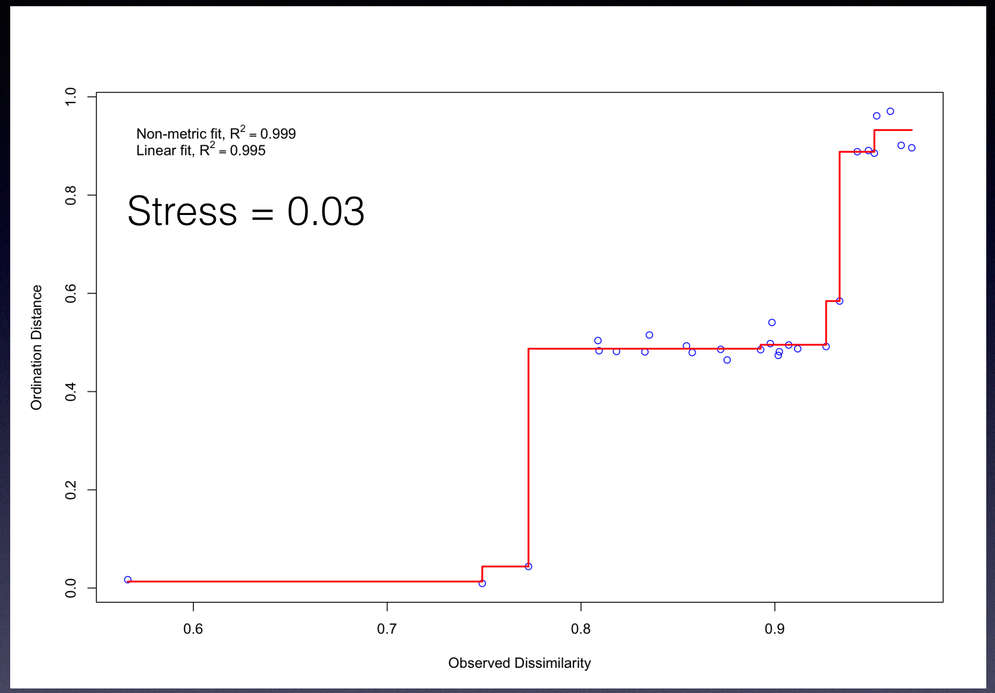
clr table



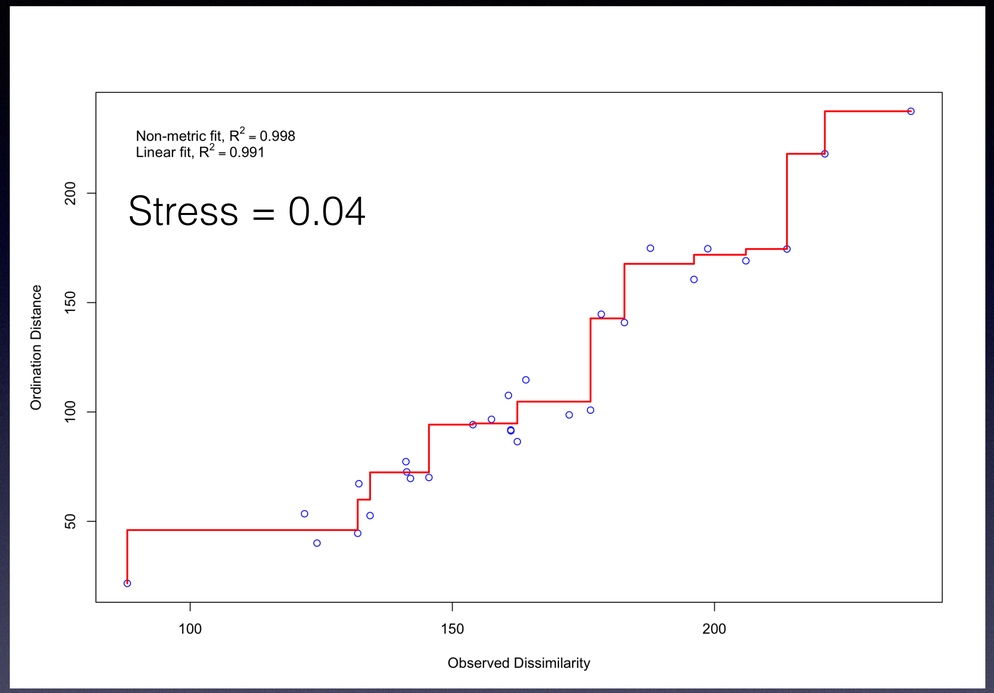

```r
#NMDS

# We will define the function NMDS.scree() that automatically performs a NMDS for 1-7 dimensions
# and plots the number of dimensions vs. stress

set.seed(666) # We include this value to make results reproducible
NMDS.scree <- function(x) { # x is the name of the distance matrix
  plot(rep(1, 7), replicate(7, metaMDS(x, autotransform = F, k = 1)$stress), xlim = c(1, 7),ylim = c(0, 0.30), xlab = "# of Dimensions", ylab =
       "Stress", main = "NMDS stress plot")
  for (i in 1:7) {
    points(rep(i + 1,7),replicate(7, metaMDS(x, autotransform = F, k = i + 1)$stress))
  }
}

# Using the function to determine the optimal number of dimensions
# Using the rarefied table
NMDS.scree(otu.tab.simple.ss.nozero.bray)
# Using the clr table
NMDS.scree(otu.tab.simple.gbm.clr.euclidean)
```

NMDS stressplot for k(dimensions)=2

Rarefied table



clr table

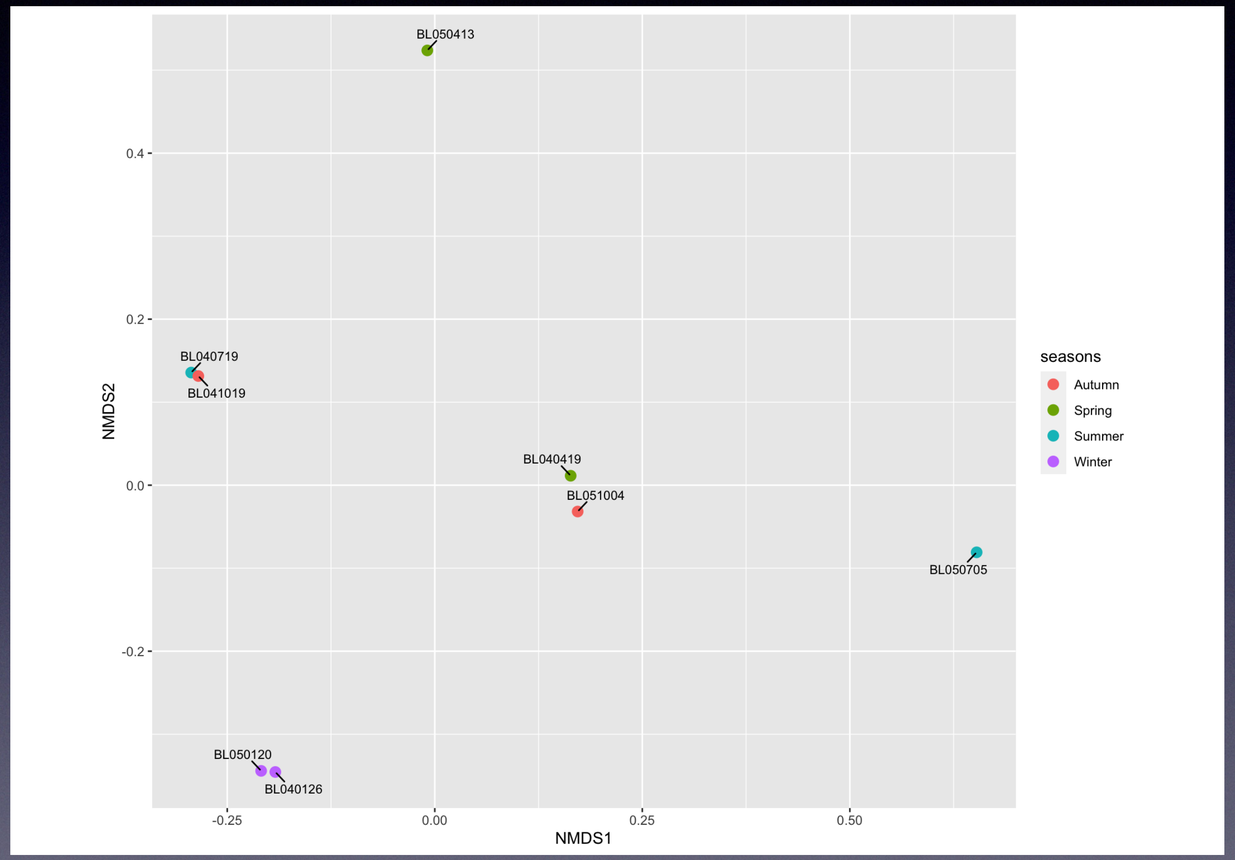


There is a good non-metric fit between observed dissimilarities in the distance matrix and the distances in ordination space

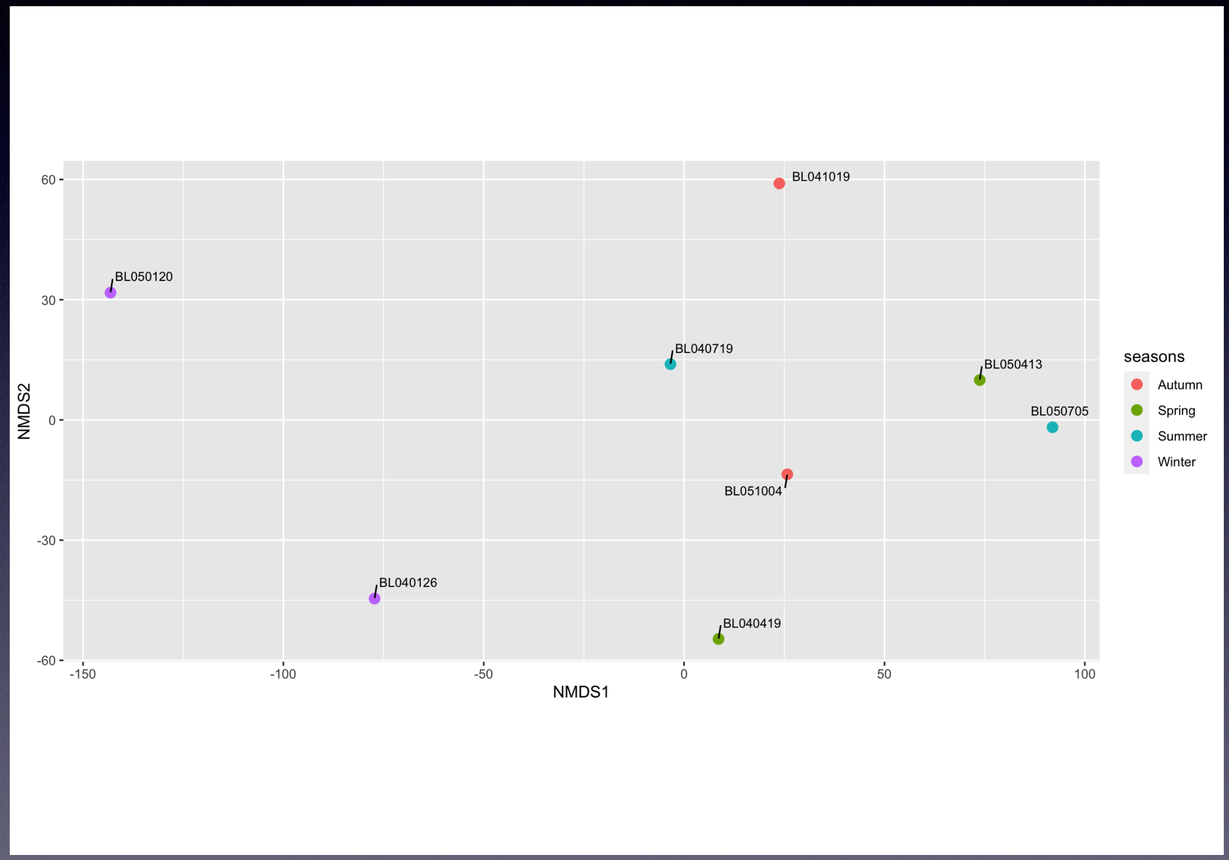
```
# We calculate NMDS for k(dimensions)=2
# Rarefied table (we use the dataframe to have access to sample and OTU names)
otu.tab.simple.ss.nozero.bray.nmds<-metaMDS(otu.tab.simple.ss.nozero, k=2, trymax=100, trace=F, autotransform = F, distance="bray")
stressplot(otu.tab.simple.ss.nozero.bray.nmds) # Make stressplot

# clr table (we use the dataframe to have access to sample and OTU names)
otu.tab.simple.gbm.clr.euclidean.nmds<-metaMDS(t(as.data.frame(otu.tab.simple.gbm.clr)), k=2, trymax=100, trace=F, autotransform = F,
                                               distance="euclidean")
stressplot(otu.tab.simple.gbm.clr.euclidean.nmds) # Make stressplot
```

NMDS plots

Rarefied table



clr table



What ordination axis corresponds to the largest gradient in our dataset (i.e. the gradient explaining most of the variance)?

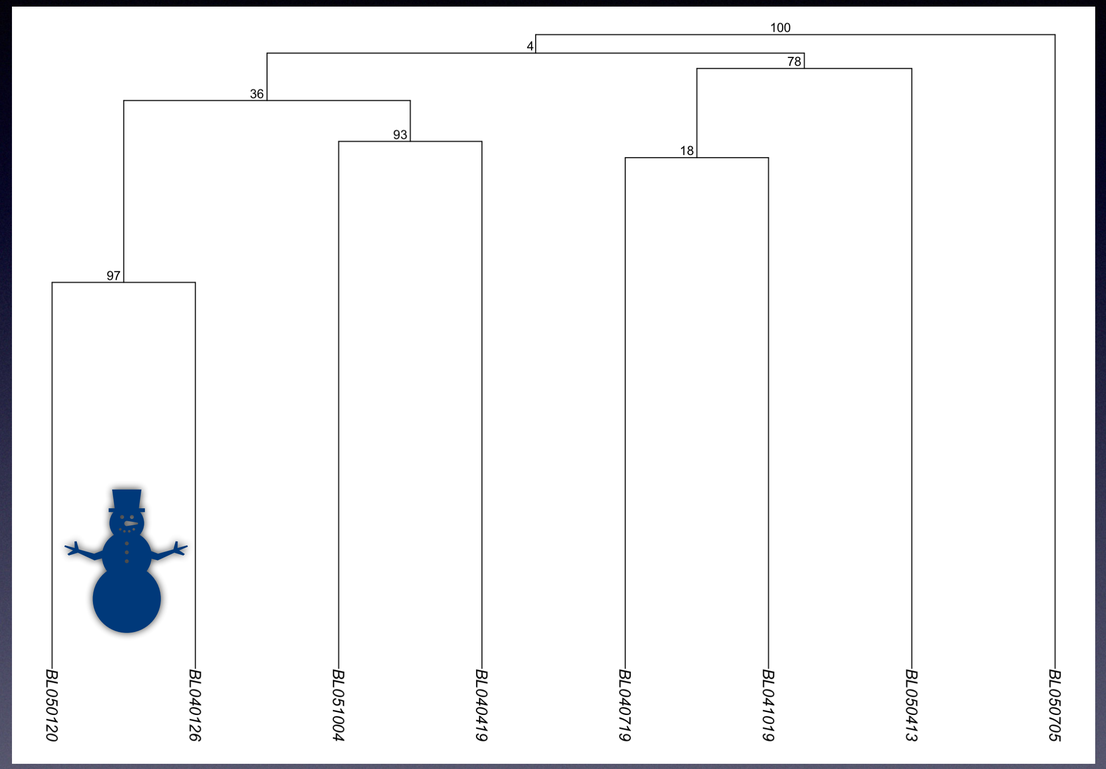
```r
# Simple plotting
# Rarefied table
plot(otu.tab.simple.ss.nozero.bray.nmds, display="sites", type="n")
points(otu.tab.simple.ss.nozero.bray.nmds, display = "sites", col = "red", pch=19)
text(otu.tab.simple.ss.nozero.bray.nmds, display ="sites")

# clr table
plot(otu.tab.simple.gbm.clr.euclidean.nmds, display="sites", type="n")
points(otu.tab.simple.gbm.clr.euclidean.nmds, display = "sites", col = "red", pch=19)
text(otu.tab.simple.gbm.clr.euclidean.nmds, display ="sites")

# Let's make nicer plots
# We define seasons for samples
seasons<-c("Winter","Spring","Summer","Autumn","Winter","Spring","Summer","Autumn")
months<-c("January","April","July","October","January","April","July","October")

library(ggplot2) # Generates nice plots
library(ggrepel) # Adds in to ggplot

# Rarefied table
# We generate a table of nmds scores and other features
otu.tab.simple.ss.nozero.bray.nmds.scores<-as.data.frame(scores(otu.tab.simple.ss.nozero.bray.nmds))
otu.tab.simple.ss.nozero.bray.nmds.scores$seasons<-seasons
otu.tab.simple.ss.nozero.bray.nmds.scores$months<-months
otu.tab.simple.ss.nozero.bray.nmds.scores$samples<-rownames(otu.tab.simple.ss.nozero.bray.nmds.scores)

#                NMDS1       NMDS2 seasons  months  samples
# BL040126 -0.192087931 -0.34552707  Winter January BL040126
# BL040419  0.163687487  0.01138097  Spring   April BL040419
# BL040719 -0.293448084  0.13565597  Summer    July BL040719
# BL041019 -0.284857321  0.13150682  Autumn October BL041019
# BL050120 -0.209189049 -0.34417159  Winter January BL050120
# BL050413 -0.009003643  0.52375809  Spring   April BL050413
# BL050705  0.652757387 -0.08086158  Summer    July BL050705
# BL051004  0.172141153 -0.03174161  Autumn October BL051004


# Create the plot
p <- ggplot(otu.tab.simple.ss.nozero.bray.nmds.scores) +
  geom_point(mapping = aes(x = NMDS1, y = NMDS2, colour = seasons), size=3) +
  coord_fixed()+## need aspect ratio of 1!
  geom_text_repel(box.padding = 0.5, aes(x = NMDS1, y = NMDS2, label = samples),
            size = 3)
p

# clr table
# We generate a table of nmds scores and other features
otu.tab.simple.gbm.clr.euclidean.nmds.scores<-as.data.frame(scores(otu.tab.simple.gbm.clr.euclidean.nmds))
tu.tab.simple.gbm.clr.euclidean.nmds.scores$seasons<-seasons
otu.tab.simple.gbm.clr.euclidean.nmds.scores$months<-months
otu.tab.simple.gbm.clr.euclidean.nmds.scores$samples<-rownames(otu.tab.simple.gbm.clr.euclidean.nmds.scores)

#                NMDS1      NMDS2 seasons  months  samples
# BL040126  -77.250023 -44.586424  Winter January BL040126
# BL040419    8.589223 -54.666069  Spring   April BL040419
# BL040719   -3.408569  13.911075  Summer    July BL040719
# BL041019   23.754238  59.015021  Autumn October BL041019
# BL050120 -143.136183  31.763211  Winter January BL050120
# BL050413   73.779344   9.957496  Spring   April BL050413
# BL050705   91.926317  -1.829225  Summer    July BL050705
# BL051004   25.745653 -13.565085  Autumn October BL051004

p <- ggplot(otu.tab.simple.gbm.clr.euclidean.nmds.scores) +
  geom_point(mapping = aes(x = NMDS1, y = NMDS2, colour = seasons), size=3) +
  coord_fixed()+## need aspect ratio of 1!
  geom_text_repel(box.padding = 0.5, aes(x = NMDS1, y = NMDS2, label = samples),
                  size = 3)
p
```

Clustering

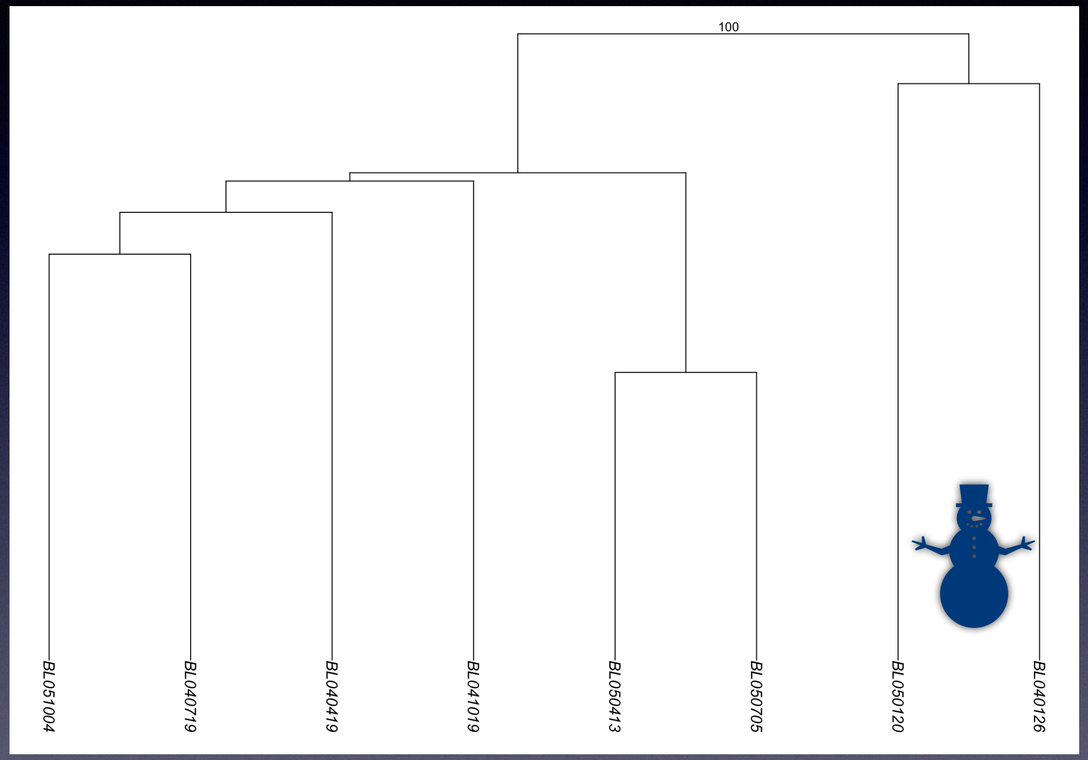
- Allows determining the similarity between samples
- Organises the samples in groups
- Hierarchical clustering: groups are organised in ranks according to their similarity
- UPGMA (unweighted pair group method with arithmetic mean): agglomerative (bottom up) hierarchical clustering
- All based in a dissimilarity matrix

```r
#Clustering of samples

# Allows determining the similarity between samples as well as the organization of samples in groups.
# Hierarchical clustering: samples will be organized in ranks according to their similarity and all samples will be included in a large group
# Unweighted Pair-Group Method Using Arithmetic Averages (UPGMA): This linkage method will link samples by considering their distance
#   to a subgroup arithmetic average. This is a method widely used in ecology

install.packages("recluster")
library("recluster")

#UPGMA

# Rarefied dataset
# We generate 100 trees by resampling and then, we use the consensus
otu.tab.simple.ss.nozero.bray.upgma<-recluster.cons(otu.tab.simple.ss.nozero.bray, tr=100, p=0.5, method="average")
plot(otu.tab.simple.ss.nozero.bray.upgma$cons) # plot consensus tree
# We'll calculate bootstrap support values (0: bad - 100: perfect)
# This allows us to know how well supported is the branching pattern
otu.tab.simple.ss.nozero.bray.upgma.boot<-recluster.boot(otu.tab.simple.ss.nozero.bray.upgma$cons, otu.tab.simple.ss.nozero,
                                                         tr=100, p=0.5, method="average", boot=1000, level=1)
recluster.plot(otu.tab.simple.ss.nozero.bray.upgma$cons, otu.tab.simple.ss.nozero.bray.upgma.boot) # We add bootstrap values to the branching
               pattern

#clr transformed dataset
# We generate 100 trees by resampling and then, we build the consensus
otu.tab.simple.gbm.clr.euclidean.upgma<-recluster.cons(otu.tab.simple.gbm.clr.euclidean, tr=100, p=0.5, method="average")
plot(otu.tab.simple.gbm.clr.euclidean.upgma$cons) # plot consensus tree
# We'll calculate bootstrap support values (0: bad - 100: perfect)
otu.tab.simple.gbm.clr.euclidean.upgma.boot<-recluster.boot(otu.tab.simple.gbm.clr.euclidean.upgma$cons, t(otu.tab.simple.gbm.clr),
                                                         tr=100, p=0.5, method="average", boot=100, level=1)
recluster.plot(otu.tab.simple.gbm.clr.euclidean.upgma$cons, otu.tab.simple.gbm.clr.euclidean.upgma.boot) # We add bootstrap values to the branching
               pattern

```

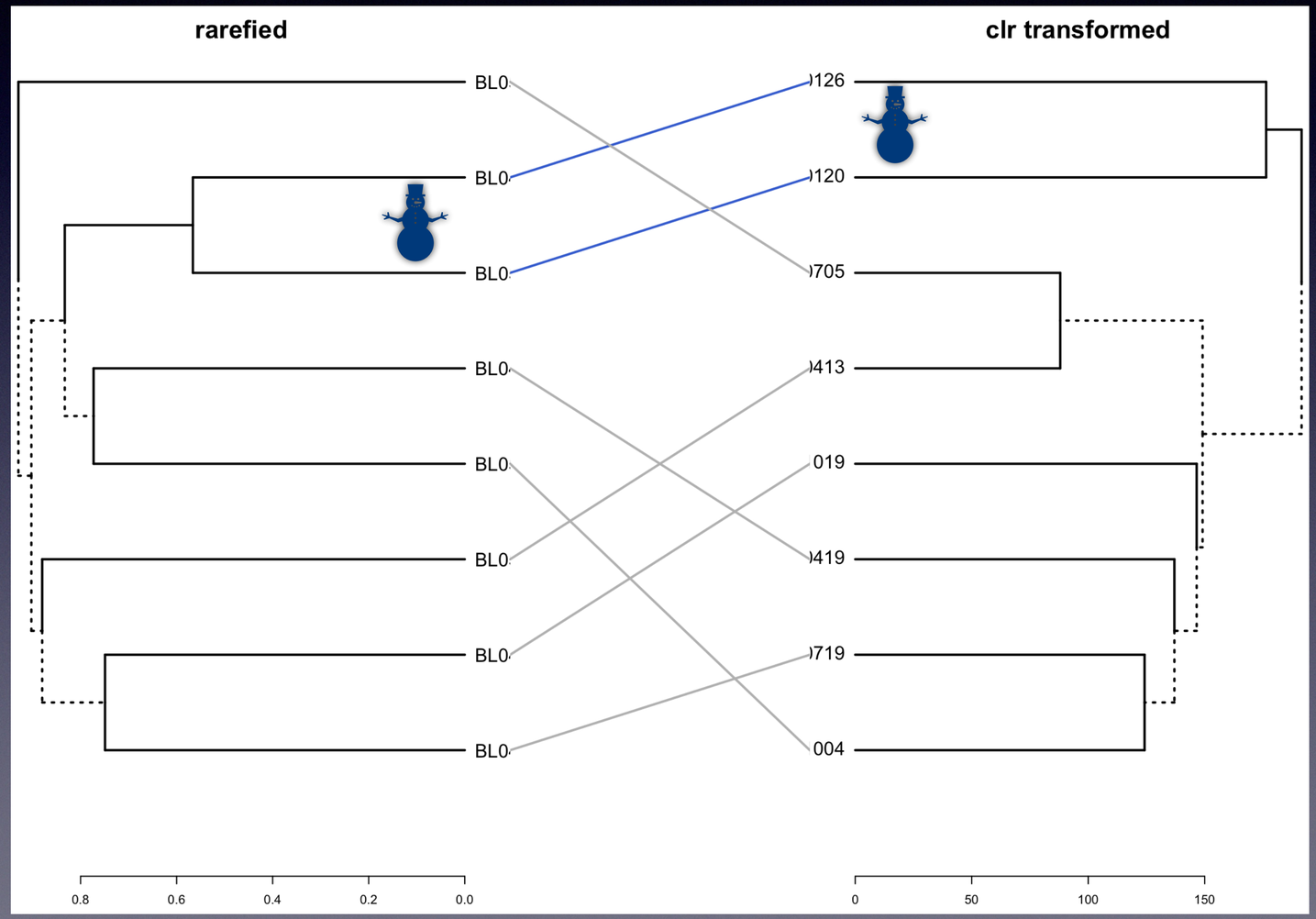
Rarefied



clr transformed



```r
#Let's compare both dendrograms using tanglegrams

install.packages("dendextend")
library(dendextend)

dendlist(as.dendrogram(otu.tab.simple.ss.nozero.bray.upgma$cons), as.dendrogram(otu.tab.simple.gbm.clr.euclidean.upgma$cons)) %>%
  untangle(method = "step1side") %>% # Find the best alignment layout
  tanglegram(cex_main=0.7, cex_sub=1, lwd=2.0, main_left="rarefied", main_right="clr transformed",cex_main_left=2, lab.cex=1.5, edge.lwd=2)
```

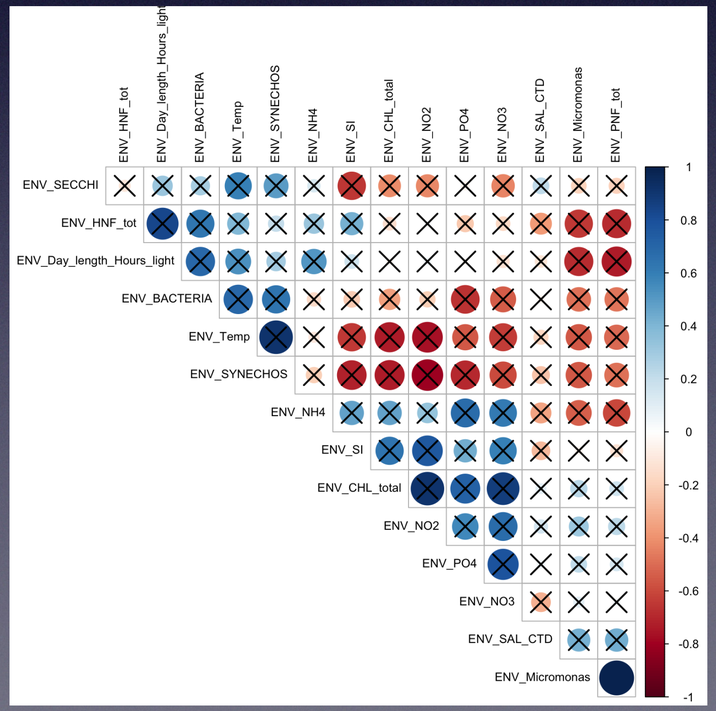


Incorporating environmental data

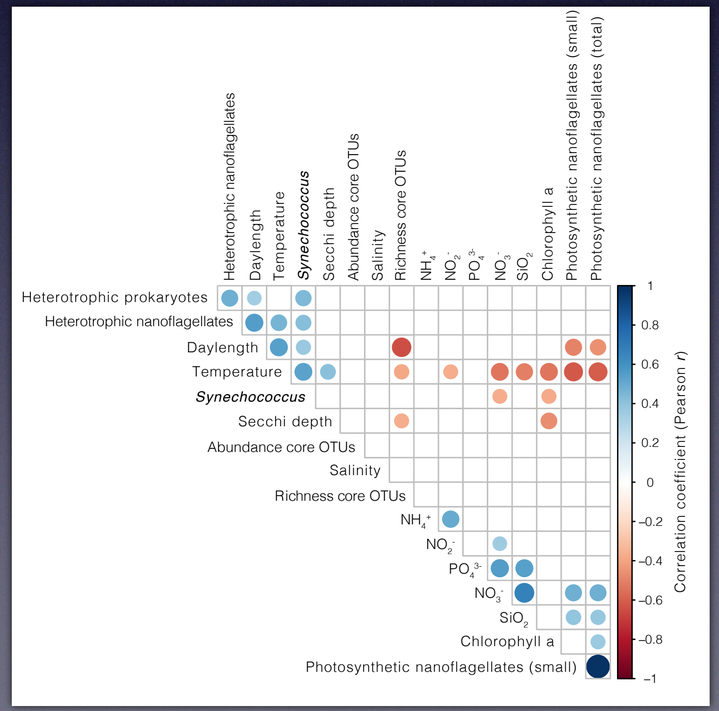
- We aim at investigating whether environmental variability could explain community variance
- Environmental variables are standardised to have comparable ranges of variation
- For each datapoint:

```r

#Analyses using environmental variation
# We aim at investigating the environmental variation that may explain community variance.
# Read environmental table
bbmo.metadata.course<-read_tsv("https://raw.githubusercontent.com/krabberod/BIO9905MERG1_V21/main/community.ecology/bbmo.metadata.course.tsv",
                               col_names = T)
bbmo.metadata.course<-as.data.frame(bbmo.metadata.course)
rownames(bbmo.metadata.course)<-bbmo.metadata.course[,1]
bbmo.metadata.course<-bbmo.metadata.course[,-1]

#                           BL040126 BL040419 BL040719 BL041019 BL050120 BL050413 BL050705 BL051004
# ENV_Temp                        14     12.6       24     19.2       13       13       24     21.5
# ENV_SECCHI                      14        6       24       12       19       18       22       17
# ENV_SAL_CTD                   37.9     35.9     36.9     37.5       37     37.7    37.35     35.1
# ENV_CHL_total                  1.1      1.4      0.4      0.3      0.5        2      0.1      0.6
# ENV_PO4                        0.2      0.2      0.1      0.1      0.2      0.3      0.2      0.2
# ENV_NH4                        0.3      1.5        1      0.5      1.1      2.1      1.4      1.5
# ENV_NO2                        0.3      0.4      0.2      0.1      0.2      0.4      0.1      0.1
# ENV_NO3                        1.5      2.5      0.1      0.4      1.1      3.3      0.2      2.4
# ENV_SI                         1.8      6.1      1.4      1.4      2.6      3.4      1.8      1.6
# ENV_BACTERIA                854356  1046779  1654834  1083724   582655   788163  1127596   885144
# ENV_SYNECHOS                  5927     1411    38741  30915.5     8253     4169    24823    33866
# ENV_Micromonas                9258     1424      203      730     4414     1543      505      573
# ENV_PNF_tot                  11451     2266     1228     2811     5853     2506     1699     2052
# ENV_HNF_tot                    329     1793     1357      822      420      669     1528      837
# ENV_Day_length_Hours_light     9.8    13.51    14.81    10.94     9.61     13.2    15.12    11.67
# Month                       01_jan   04_apr   07_jul   10_oct   01_jan   04_apr   07_jul   10_oct
# Season                         win      spr      sum      aut      win      spr      sum      aut
# Season_corr                    win      spr      sum      aut      win      spr      sum      aut
# Year                          2004     2004     2004     2004     2005     2005     2005     2005

#Double check samples are correct
identical(colnames(otu.tab.simple.gbm.clr),colnames(bbmo.metadata.course))
# [1] TRUE #Both tables have the same names
#We transform variables 1:15 using z-scores to have comparable ranges of variation
bbmo.metadata.course.15vars<-bbmo.metadata.course[1:15,] #We select continuous variables
bbmo.metadata.course.15vars[]<- lapply(bbmo.metadata.course.15vars, as.character) #We transform the datatype to characters
bbmo.metadata.course.15vars[]<- lapply(bbmo.metadata.course.15vars, as.numeric) #We transform to numeric
#lapply : applies a function to a list object
bbmo.metadata.course.15vars.zscores<-scale(t(bbmo.metadata.course.15vars), center = T, scale = T)
bbmo.metadata.course.15vars.zscores[,1:5]

#            ENV_Temp  ENV_SECCHI ENV_SAL_CTD ENV_CHL_total    ENV_PO4
# BL040126 -0.7223777 -0.43425521  1.02225526     0.4644927  0.1950474
# BL040419 -0.9985084 -1.82387188 -1.06132234     0.9289853  0.1950474
# BL040719  1.2499845  1.30276563 -0.01953354    -0.6193235 -1.3653316
# BL041019  0.3032507 -0.78165938  0.60553974    -0.7741544 -1.3653316
# BL050120 -0.9196139  0.43425521  0.08464534    -0.4644927  0.1950474
# BL050413 -0.9196139  0.26055313  0.81389750     1.8579706  1.7554264
# BL050705  1.2499845  0.95536146  0.44927142    -1.0838162  0.1950474
# BL051004  0.7568940  0.08685104 -1.89475338    -0.3096618  0.1950474
```

```r
#Let's check the correlation in environmental variables
install.packages("corrplot") # makes nice correlation plots
install.packages("RcmdrMisc") # diverse tools
library("corrplot")
library("RcmdrMisc")

#We calculate correlations and p-values
env.corr.signif.adjust<-rcorr.adjust(as.matrix(bbmo.metadata.course.15vars.zscores)) # The p-values are corrected for multiple inference using
                                Holm's method (see p.adjust).

#Holm corrected values for multiple comparisons
env.corr.signif.r<-env.corr.signif.adjust$R$r
env.corr.signif.p<-env.corr.signif.adjust$P
# We edit the objetc to replace any "<" by "0" using the function "gsub"
env.corr.signif.p<-gsub("<","0", env.corr.signif.p)
# We modify the object to be numeric datatype. #NB: the transformation is done so the matrix of p values can be read as numeric!
env.corr.signif.p <- apply(env.corr.signif.p, 2 ,as.numeric)
# We plot the correlation plot
corrplot(env.corr.signif.r , type="upper", order="hclust", p.mat = env.corr.signif.p, sig.level = 0.05,
         insig = "pch", hclust.method = c("average"), tl.cex= 0.8, tl.col="black", diag=F)
```



Using 8 samples (our dataset)



Using 120 samples (full dataset)

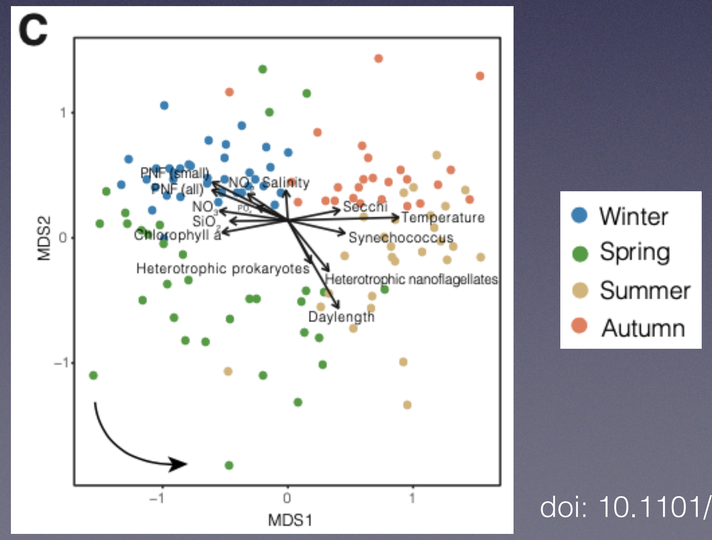
doi: 10.1101/2021.03.18.435965

Unconstrained vs. Constrained ordination

- In unconstrained ordination we first find the major compositional variation, and then relate this variation to observed environmental variation (envfit e.g.)
- In constrained ordination we do not want to display all or even most of the compositional variation, but only the variation that can be explained by the used environmental variables, or constraints
- The constrained ordination is non-symmetric: we have independent variables or constraints (environmental data) and we have dependent variables or the community

Unconstrained ordination: Fitting vectors

- We correlate environmental variables with ordination axes
- The arrow points to the direction of most rapid change in the the environmental variable. Often this is called the direction of the gradient
- The length of the arrow is proportional to the correlation between ordination and environmental variable. Often this is called the strength of the gradient



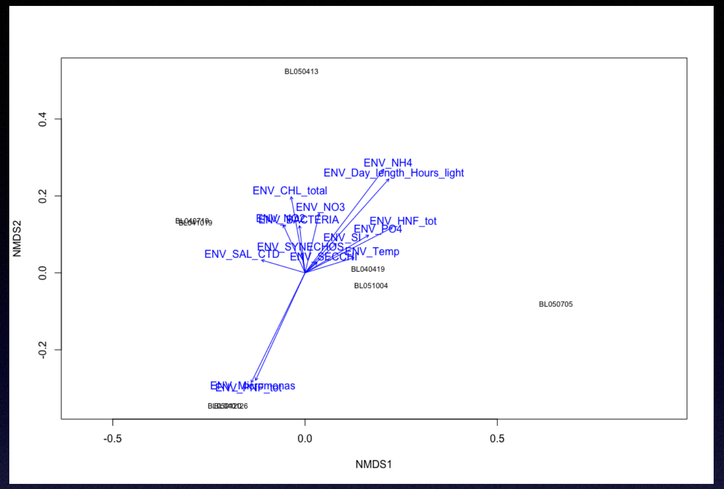
doi: 10.1101/2021.03.18.435965

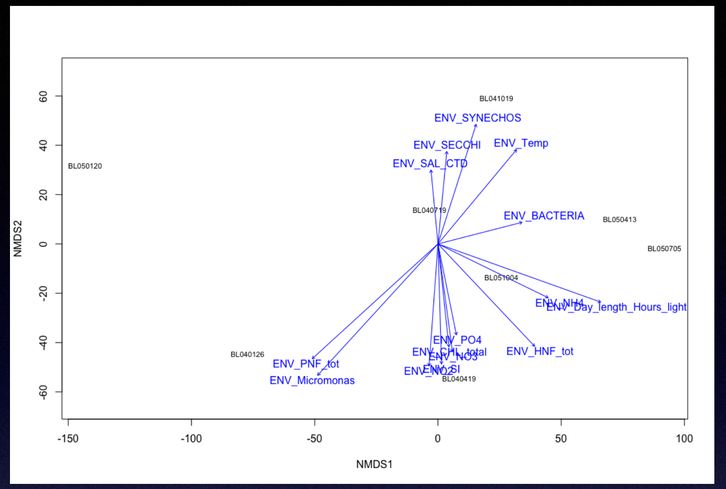
Rarefied

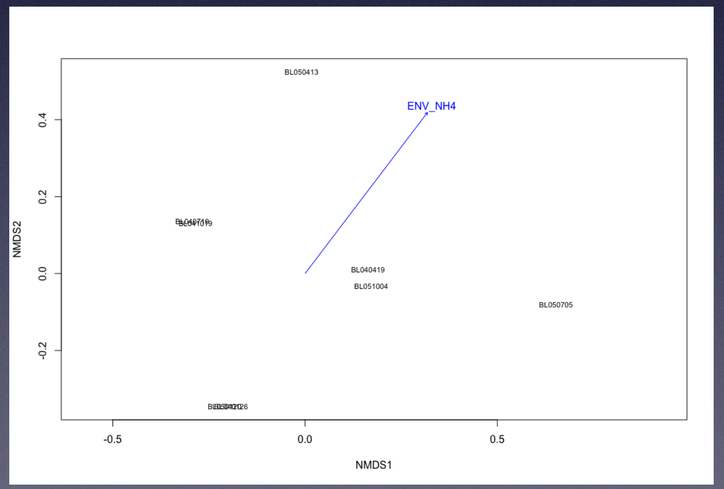
clr transformed

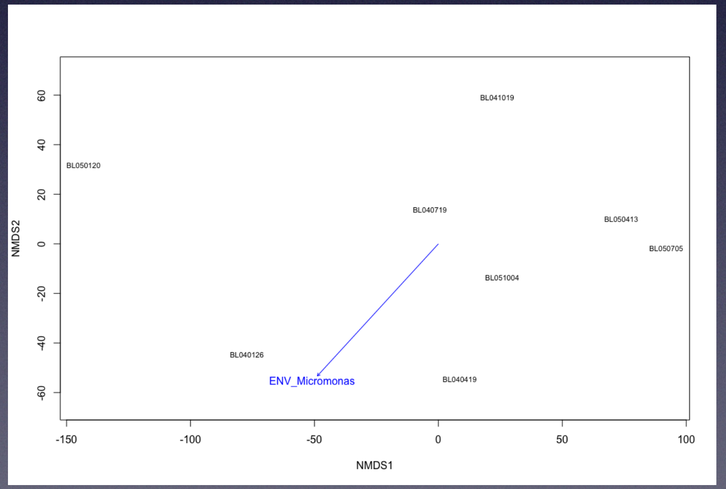
All

p<0.1









Constrained ordination

1) Redundancy Analysis (RDA) can be considered as a constrained version of PCA

-Distance based RDA: allows calculating RDA with a chosen distance matrix

2) Constrained Correspondence Analysis (CCA) is the constrained version of Correspondence Analyses

Selecting environmental variables that explain most community variance

- *Forward selection:* begins with an empty model and adds variables one by one. In each step forward, it adds one variable that gives the single best improvement to the model

- *Backwards elimination:* starts with a model that includes all variables and eliminates variables with low explanatory power one by one

- Ordistep (Vegan): Performs step-wise selection of environmental variables based on two criteria:
  - If their inclusion into the model leads to a significant increase of the explained variance
  - If the AIC (Akaike Information Criterion) of the new model is lower than the AIC of the more simple model
    - AIC: estimates the quality of models relative to other models (model selection). It is an estimator of prediction error

dbRDA

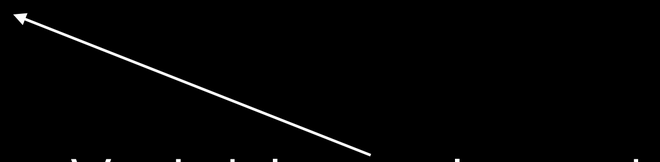
Rarefied table

```
#Constrained Ordination
# Selection of the most important variables for distance-based redundancy analyses

#Rarefaction table
mod0.rarefaction<-capscale(otu.tab.simple.ss.nozero.bray~1, as.data.frame(bbmo.metadata.course.15vars.zscores)) # model containing only species
                           matrix and intercept
mod1.rarefaction<-capscale(otu.tab.simple.ss.nozero.bray~ ., as.data.frame(bbmo.metadata.course.15vars.zscores)) # # model including all variables
                           from env matrix (the dot after tilde (~) means ALL!)
ordistep(mod0.rarefaction, scope = formula(mod1.rarefaction), perm.max = 1000, direction="forward")

# Start: otu.tab.simple.ss.nozero.bray ~ 1
#                              Df    AIC      F Pr(>F)
# + ENV_PNF_tot                 1 9.2535 1.3702  0.050 *
# + ENV_Day_length_Hours_light  1 9.1702 1.4474  0.055 .
# + ENV_Micromonas              1 9.2311 1.3909  0.055 .
# + ENV_BACTERIA                1 9.4129 1.2248  0.110
# + ENV_Temp                    1 9.3168 1.3121  0.170
# + ENV_PO4                     1 9.4892 1.1562  0.195
# + ENV_HNF_tot                 1 9.4548 1.1870  0.240
# + ENV_SYNECHOS                1 9.4613 1.1812  0.255
# + ENV_NH4                     1 9.5305 1.1193  0.285
# + ENV_NO3                     1 9.5767 1.0784  0.350
# + ENV_NO2                     1 9.6558 1.0087  0.380
# + ENV_CHL_total               1 9.5684 1.0857  0.385
# + ENV_SAL_CTD                 1 9.6782 0.9891  0.485
# + ENV_SI                      1 9.7718 0.9078  0.590
# + ENV_SECCHI                  1 9.8076 0.8770  0.675
# ---
#   Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# Step: otu.tab.simple.ss.nozero.bray ~ ENV_PNF_tot
#                              Df     AIC      F Pr(>F)
# + ENV_SAL_CTD                 1  9.5007 1.2248  0.225
# + ENV_PO4                     1  9.4897 1.2333  0.235
# + ENV_CHL_total               1  9.5584 1.1800  0.285
# + ENV_NO3                     1  9.6004 1.1477  0.285
# + ENV_NH4                     1  9.6031 1.1456  0.330
# + ENV_NO2                     1  9.7120 1.0625  0.370
# + ENV_Temp                    1  9.7212 1.0555  0.380
# + ENV_SYNECHOS                1  9.7690 1.0195  0.465
# + ENV_SI                      1  9.7931 1.0013  0.600
# + ENV_BACTERIA                1  9.8945 0.9258  0.620
# + ENV_HNF_tot                 1  9.9316 0.8983  0.645
# + ENV_SECCHI                  1  9.9584 0.8786  0.675
# + ENV_Day_length_Hours_light  1 10.0502 0.8115  0.720
# + ENV_Micromonas              1 10.0363 0.8216  0.745

# Call: capscale(formula = otu.tab.simple.ss.nozero.bray ~ ENV_PNF_tot, data = as.data.frame(bbmo.metadata.course.15vars.zscores))
# NB: the variables in this model are the ones that were selected.

# Inertia Proportion Rank
# Total          2.7072     1.0000
# Constrained    0.5033     0.1859    1
# Unconstrained  2.2039     0.8141    6
# Inertia is squared Bray distance

# Eigenvalues for constrained axes:
#   CAP1
# 0.5033

# Eigenvalues for unconstrained axes:
#  MDS1   MDS2   MDS3   MDS4   MDS5   MDS6
# 0.5958 0.4353 0.3912 0.2791 0.2778 0.2246
```

Variables selected

clr table

```
#clr table
mod0.clr<-capscale(otu.tab.simple.gbm.clr.euclidean~1, as.data.frame(bbmo.metadata.course.15vars.zscores)) # model containing only species matrix
                  and intercept
mod1.clr<-capscale(otu.tab.simple.gbm.clr.euclidean~ ., as.data.frame(bbmo.metadata.course.15vars.zscores)) # # model including all variables from
                  env matrix (the dot after tilde (~) means ALL!)
ordistep(mod0.clr, scope = formula(mod1.clr), perm.max = 1000, direction="forward")

# Start: otu.tab.simple.gbm.clr.euclidean ~ 1
#                            Df    AIC      F Pr(>F)
# + ENV_Day_length_Hours_light  1 76.928 2.0796  0.025 *
# + ENV_Micromonas              1 77.021 1.9868  0.060 .
# + ENV_PNF_tot                 1 77.049 1.9591  0.065 .
# + ENV_NH4                     1 77.529 1.4954  0.110
# + ENV_HNF_tot                 1 77.774 1.2687  0.190
# + ENV_Temp                    1 78.088 0.9896  0.400
# + ENV_SECCHI                  1 78.249 0.8497  0.500
# + ENV_BACTERIA                1 78.185 0.9050  0.535
# + ENV_SI                      1 78.382 0.7368  0.675
# + ENV_SYNECHOS                1 78.334 0.7778  0.685
# + ENV_PO4                     1 78.364 0.7525  0.695
# + ENV_SAL_CTD                 1 78.523 0.6196  0.860
# + ENV_NO2                     1 78.604 0.5528  0.940
# + ENV_NO3                     1 78.680 0.4909  0.980
# + ENV_CHL_total               1 78.756 0.4298  0.995
# ---
#  Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# Step: otu.tab.simple.gbm.clr.euclidean ~ ENV_Day_length_Hours_light

#                  Df    AIC      F Pr(>F)
# + ENV_BACTERIA    1 77.310 1.1210  0.365
# + ENV_Micromonas  1 77.406 1.0484  0.390
# + ENV_SECCHI      1 77.418 1.0391  0.470
# + ENV_PNF_tot     1 77.477 0.9945  0.475
# + ENV_HNF_tot     1 77.568 0.9267  0.565
# + ENV_PO4         1 77.596 0.9060  0.630
# + ENV_NH4         1 77.634 0.8781  0.640
# + ENV_SI          1 77.620 0.8880  0.650
# + ENV_NO3         1 77.918 0.6733  0.735
# + ENV_SAL_CTD     1 77.857 0.7165  0.760
# + ENV_SYNECHOS    1 77.855 0.7183  0.795
# + ENV_NO2         1 77.943 0.6556  0.825
# + ENV_Temp        1 78.013 0.6063  0.870
# + ENV_CHL_total   1 78.151 0.5100  0.920

# Call: capscale(formula = otu.tab.simple.gbm.clr.euclidean ~ ENV_Day_length_Hours_light, data = as.data.frame(bbmo.metadata.course.15vars.zscores))

# Inertia Proportion Rank
# Total         1.400e+04  1.000e+00
# Constrained   3.605e+03  2.574e-01    1
# Unconstrained 1.040e+04  7.426e-01    6
# Inertia is mean squared Euclidean distance

# Eigenvalues for constrained axes:
#  CAP1
#  3605

# Eigenvalues for unconstrained axes:
#  MDS1 MDS2 MDS3 MDS4 MDS5 MDS6
#  3253 2380 1780 1498  931  558
```



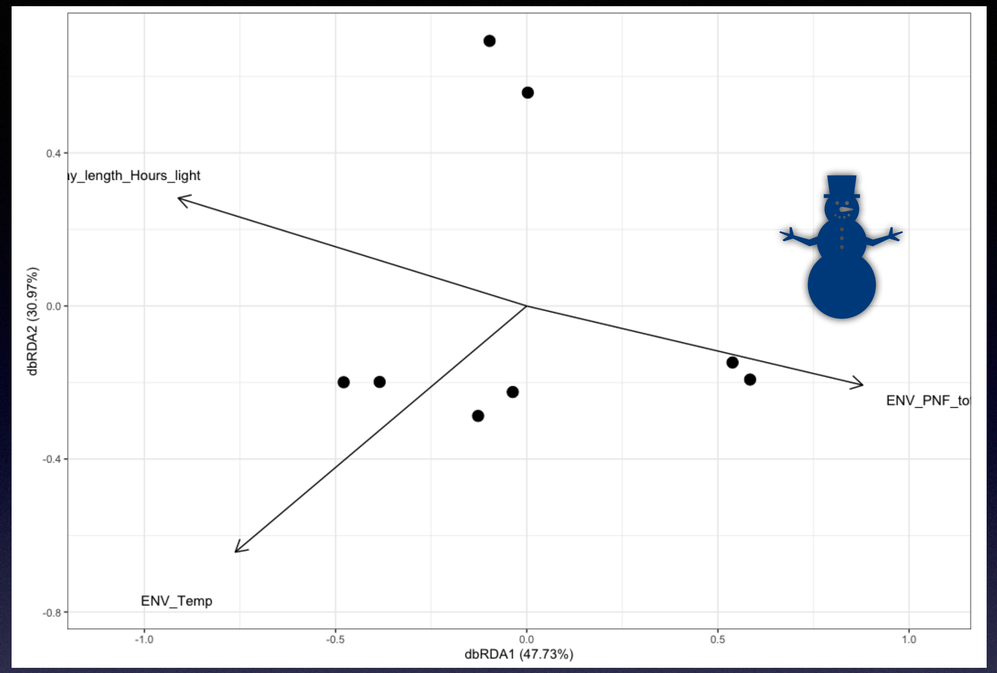
Variables selected

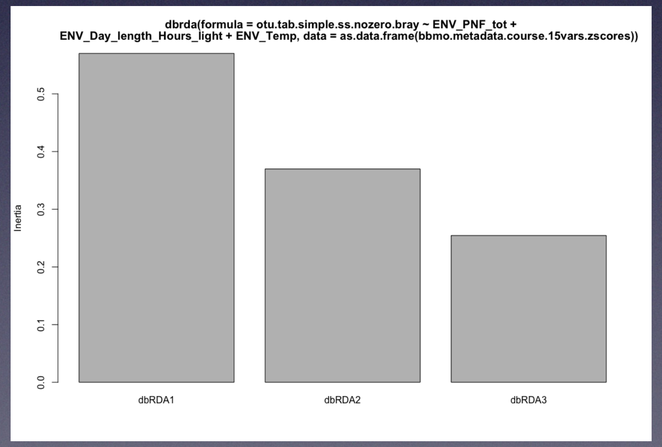
We use a few more variables that are known to be important drivers of community variance

1. Day-length
2. Temperature

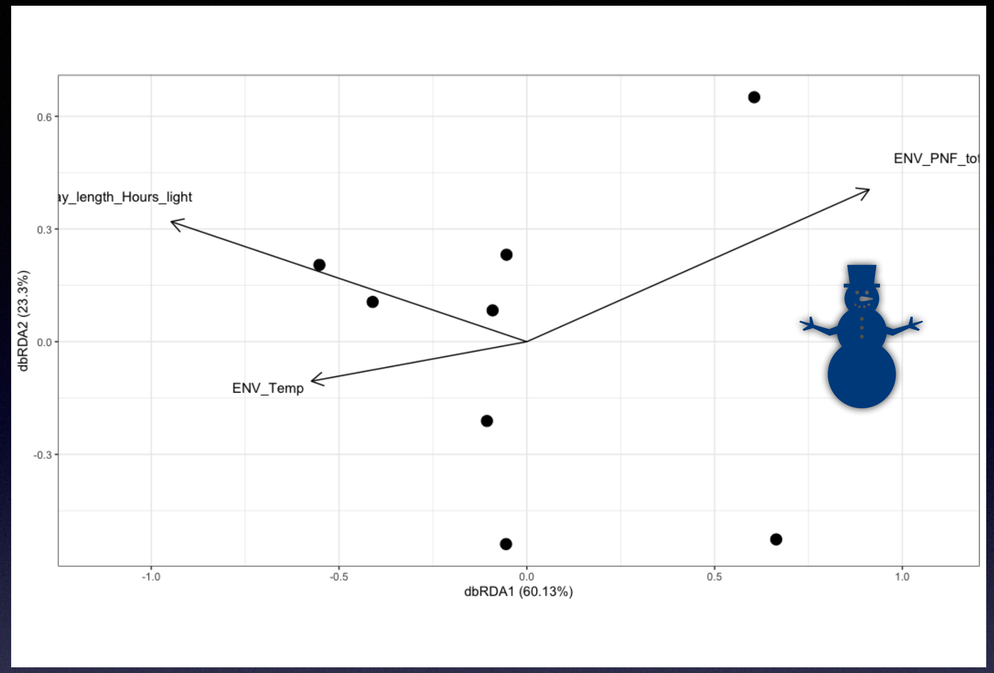
```
#Generate the ordination
# We will use two more variables that we know they are important in for this dataset

#We install ggord for nicer plots
library(devtools)
install_github('fawda123/ggord')
library(ggord)
library(ggplot2)

#rarefied table
ggord(dbrda(formula = otu.tab.simple.ss.nozero.bray ~ ENV_PNF_tot+ENV_Day_length_Hours_light+ENV_Temp, data = as.data.frame(bbmo.metadata.course.
      15vars.zscores)))
screeplot(dbrda(formula = otu.tab.simple.ss.nozero.bray ~ ENV_PNF_tot+ENV_Day_length_Hours_light+ENV_Temp, data = as.data.frame(bbmo.metadata.
          course.15vars.zscores)))
dbrda(formula = otu.tab.simple.ss.nozero.bray ~ ENV_PNF_tot+ENV_Day_length_Hours_light+ENV_Temp, data = as.data.frame(bbmo.metadata.course.15vars.
      zscores))

# Call: dbrda(formula = otu.tab.simple.ss.nozero.bray ~ ENV_PNF_tot + ENV_Day_length_Hours_light + ENV_Temp, data =
#               as.data.frame(bbmo.metadata.course.15vars.zscores))
#                Inertia Proportion Rank
# Total          2.7072     1.0000
# Constrained    1.1945     0.4412    3   # Community variation constrained by the used variables
# Unconstrained  1.5127     0.5588    4
# Inertia is squared Bray distance  # Inertia = variance in species abundances

# Eigenvalues for constrained axes:
#   dbRDA1 dbRDA2 dbRDA3
#   0.5701 0.3699 0.2545

# Eigenvalues for unconstrained axes:
#     MDS1   MDS2   MDS3   MDS4
#  0.5818 0.3960 0.2997 0.2353

#clr table
ggord(dbrda(formula = otu.tab.simple.gbm.clr.euclidean ~ ENV_PNF_tot+ENV_Day_length_Hours_light+ENV_Temp, data = as.data.frame(bbmo.metadata.course.
      15vars.zscores)))
screeplot(dbrda(formula = otu.tab.simple.gbm.clr.euclidean ~ ENV_PNF_tot+ENV_Day_length_Hours_light+ENV_Temp, data = as.data.frame(bbmo.metadata.
          course.15vars.zscores)))
dbrda(formula = otu.tab.simple.gbm.clr.euclidean ~ ENV_PNF_tot+ENV_Day_length_Hours_light+ENV_Temp, data = as.data.frame(bbmo.metadata.course.
      15vars.zscores))

# Call: dbrda(formula = otu.tab.simple.gbm.clr.euclidean ~ ENV_PNF_tot + ENV_Day_length_Hours_light + ENV_Temp, data =
#             as.data.frame(bbmo.metadata.course.15vars.zscores))

#                 Inertia Proportion Rank
# Total         14004.823      1.000
# Constrained    6399.625      0.457    3   # Community variation constrained by the used variables
# Unconstrained  7605.199      0.543    4
# Inertia is mean squared Euclidean distance

# Eigenvalues for constrained axes:
#  dbRDA1 dbRDA2 dbRDA3
#   3848   1491   1061

# Eigenvalues for unconstrained axes:
#     MDS1   MDS2   MDS3   MDS4
#   3033.4 2224.1 1504.9  842.9
```

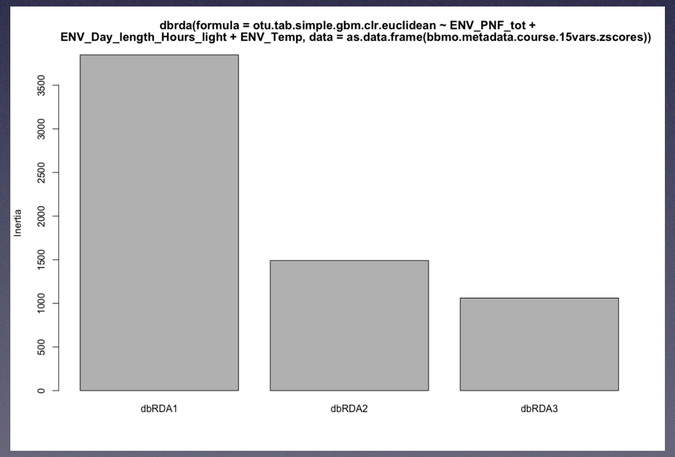
Rarefied





clr transformed





CCA

Rarefied table

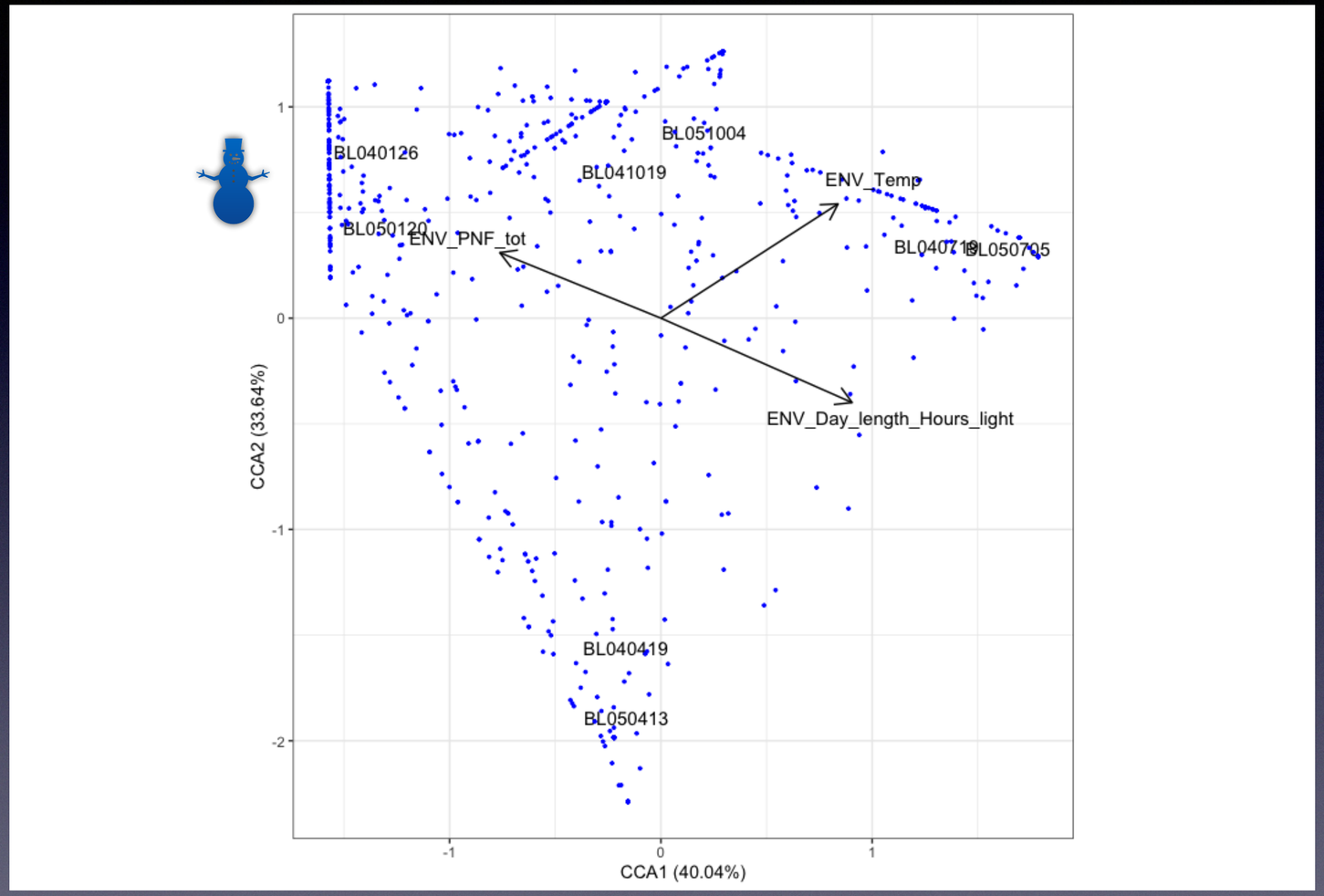
We use a few more variables that are known to be important drivers of community variance

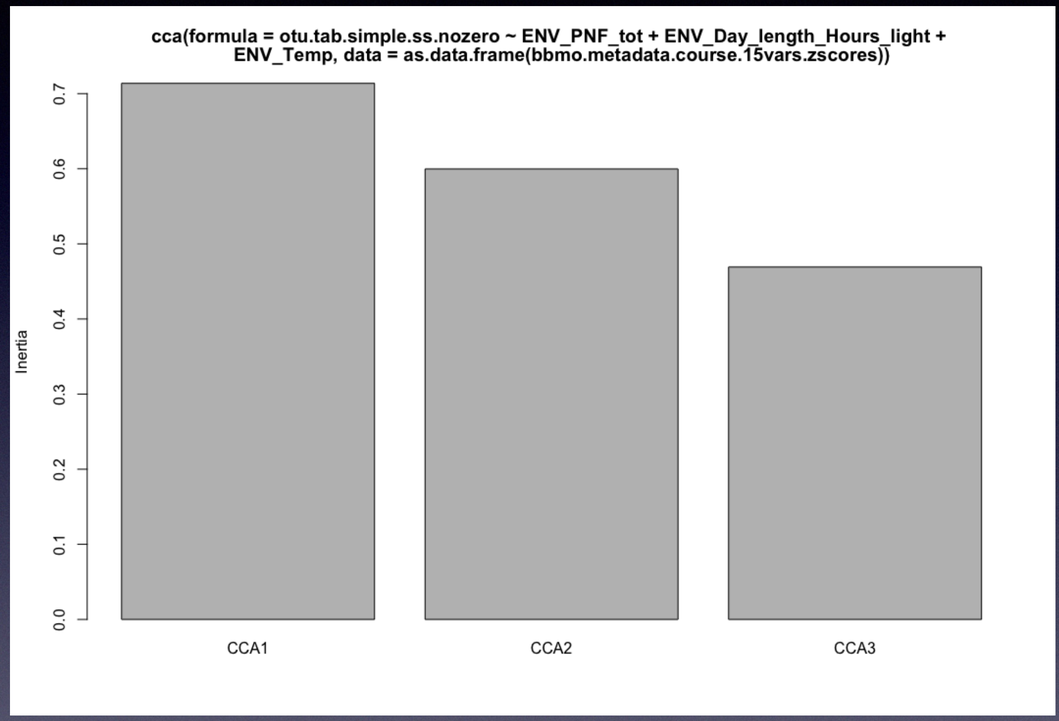
1. Day-length
2. Temperature

```
#Constrained Correspondence Analyses (CCA)
# Selection of the most important variables
# rarefaction table
mod0.cca.rarefaction<-cca(otu.tab.simple.ss.nozero~1, as.data.frame(bbmo.metadata.course.15vars.zscores)) # model containing only species matrix
                          and intercept
mod1.cca.rarefaction<-cca(otu.tab.simple.ss.nozero~ ., as.data.frame(bbmo.metadata.course.15vars.zscores)) # # model including all variables from
                          env matrix (the dot after tilde (~) means ALL!)
ordistep(mod0.cca.rarefaction, scope = formula(mod1.cca.rarefaction), perm.max = 1000, direction="forward")

# Call: cca(formula = otu.tab.simple.ss.nozero ~ ENV_Day_length_Hours_light, data = as.data.frame(bbmo.metadata.course.15vars.zscores)) # Best
            model

#                Inertia Proportion Rank
# Total          4.1915     1.0000
# Constrained    0.6904     0.1647    1   # Community variation constrained by the used variables
# Unconstrained  3.5012     0.8353    6
# Inertia is scaled Chi-square

# Eigenvalues for constrained axes:
#   CCA1
# 0.6904

# Eigenvalues for unconstrained axes:
# CA1    CA2    CA3    CA4    CA5    CA6
# 0.7987 0.6988 0.6026 0.5395 0.5271 0.3345

ggord(cca(formula = otu.tab.simple.ss.nozero ~ ENV_PNF_tot+ENV_Day_length_Hours_light+ENV_Temp, data = as.data.frame(bbmo.metadata.course.15vars.
      zscores)), obslab=T, addsize=0.6)
screeplot(cca(formula = otu.tab.simple.ss.nozero ~ ENV_PNF_tot+ENV_Day_length_Hours_light+ENV_Temp, data = as.data.frame(bbmo.metadata.course.
          15vars.zscores)))
cca(formula = otu.tab.simple.ss.nozero ~ ENV_PNF_tot+ENV_Day_length_Hours_light+ENV_Temp, data = as.data.frame(bbmo.metadata.course.15vars.zscores))

# Call: cca(formula = otu.tab.simple.ss.nozero ~ ENV_PNF_tot + ENV_Day_length_Hours_light + ENV_Temp, data =
#            as.data.frame(bbmo.metadata.course.15vars.zscores))

#                Inertia Proportion Rank
# Total          4.1915     1.0000
# Constrained    1.7827     0.4253    3   # Community variation constrained by the used variables
# Unconstrained  2.4089     0.5747    4
# Inertia is scaled Chi-square

# Eigenvalues for constrained axes:
#   CCA1   CCA2   CCA3
# 0.7137 0.5997 0.4692

# Eigenvalues for unconstrained axes:
#  CA1    CA2    CA3    CA4
# 0.7667 0.6562 0.5585 0.4275
```

CCA

Rarefied





NB: Not calculated for the clr table due to issues with the CCA running on clr

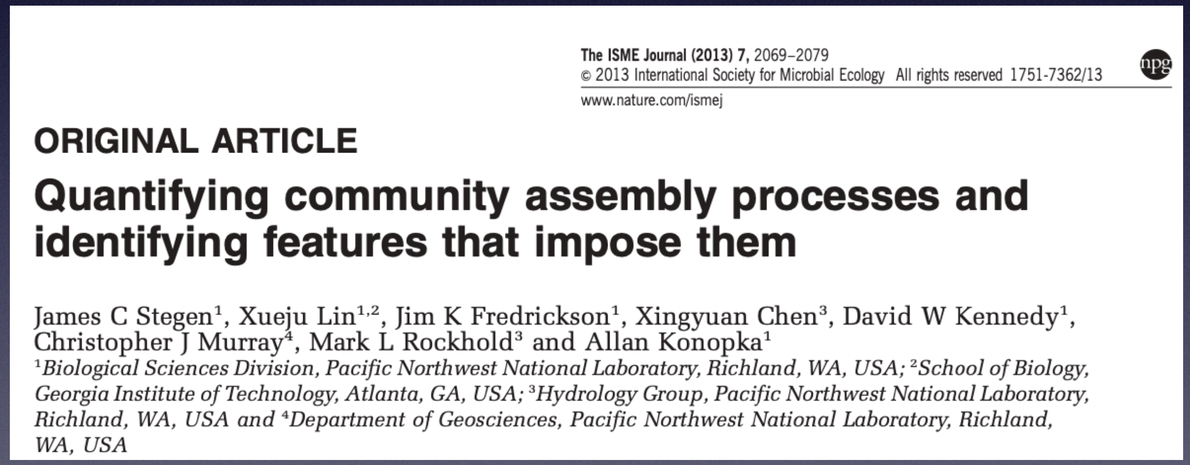
PERMANOVA
(Permutation multivariate analysis of variance)

- Used to partition the variation between multiple factors
- Here, we will use it to quantify the amount of community variation explained by different environmental factors
- The order of the factors in the analysis is important
    - We should have preliminary information on the importance of the different environmental variables
    - The most important variables should be tested first

```r
#PERMANOVA

#rarefied table
permanova.rarefaction<-adonis(otu.tab.simple.ss.nozero~ENV_PNF_tot+ENV_Day_length_Hours_light+ENV_Temp, data = as.data.frame(bbmo.metadata.course.
                              15vars.zscores),
                              method="bray", permutations=9999)
# Call:
# adonis(formula = otu.tab.simple.ss.nozero ~ ENV_PNF_tot + ENV_Day_length_Hours_light + ENV_Temp, data = as.data.frame(bbmo.metadata.course.15vars.
         zscores), permutations = 9999, method = "bray")

# Permutation: free
# Number of permutations: 9999
# Terms added sequentially (first to last)

#                           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
# ENV_PNF_tot                1   0.50330 0.50330 1.33088 0.18591 0.0844 . # PNF is close to significance, explaining ca. 18% of the variance
# ENV_Day_length_Hours_light 1   0.30774 0.30774 0.81378 0.11368 0.7359
# ENV_Temp                   1   0.38345 0.38345 1.01397 0.14164 0.4271
# Residuals                  4   1.51267 0.37817         0.55877
# Total                      7   2.70716                 1.00000
# ---
# Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

#
#clr-transformed table
permanova.clr<-adonis(t(otu.tab.simple.gbm.clr)~ENV_PNF_tot+ENV_Day_length_Hours_light+ENV_Temp, data = as.data.frame(bbmo.metadata.course.15vars.
                      zscores),
                      method="euclidean", permutations=9999)
# Call:
#  adonis(formula = t(otu.tab.simple.gbm.clr) ~ ENV_PNF_tot + ENV_Day_length_Hours_light +      ENV_Temp, data = as.data.frame(bbmo.metadata.course.
          15vars.zscores),      permutations = 9999, method = "euclidean")

# Permutation: free
# Number of permutations: 9999
# Terms added sequentially (first to last)

#                           Df SumsOfSqs MeanSqs F.Model     R2 Pr(>F)
# ENV_PNF_tot                1     24130 24130.3 1.81307 0.24614 0.1085
# ENV_Day_length_Hours_light 1     13180 13180.0 0.99030 0.13444 0.3917
# ENV_Temp                   1      7487  7487.1 0.56255 0.07637 0.9368
# Residuals                  4     53236 13309.1         0.54304
# Total                      7     98034                 1.00000
```

Other things you could explore

- The relative importance of the main processes structuring microbiotas
  - Selection
  - Dispersal
  - Drift



The ISME Journal (2013) 7, 2069–2079

www.nature.com/ismej

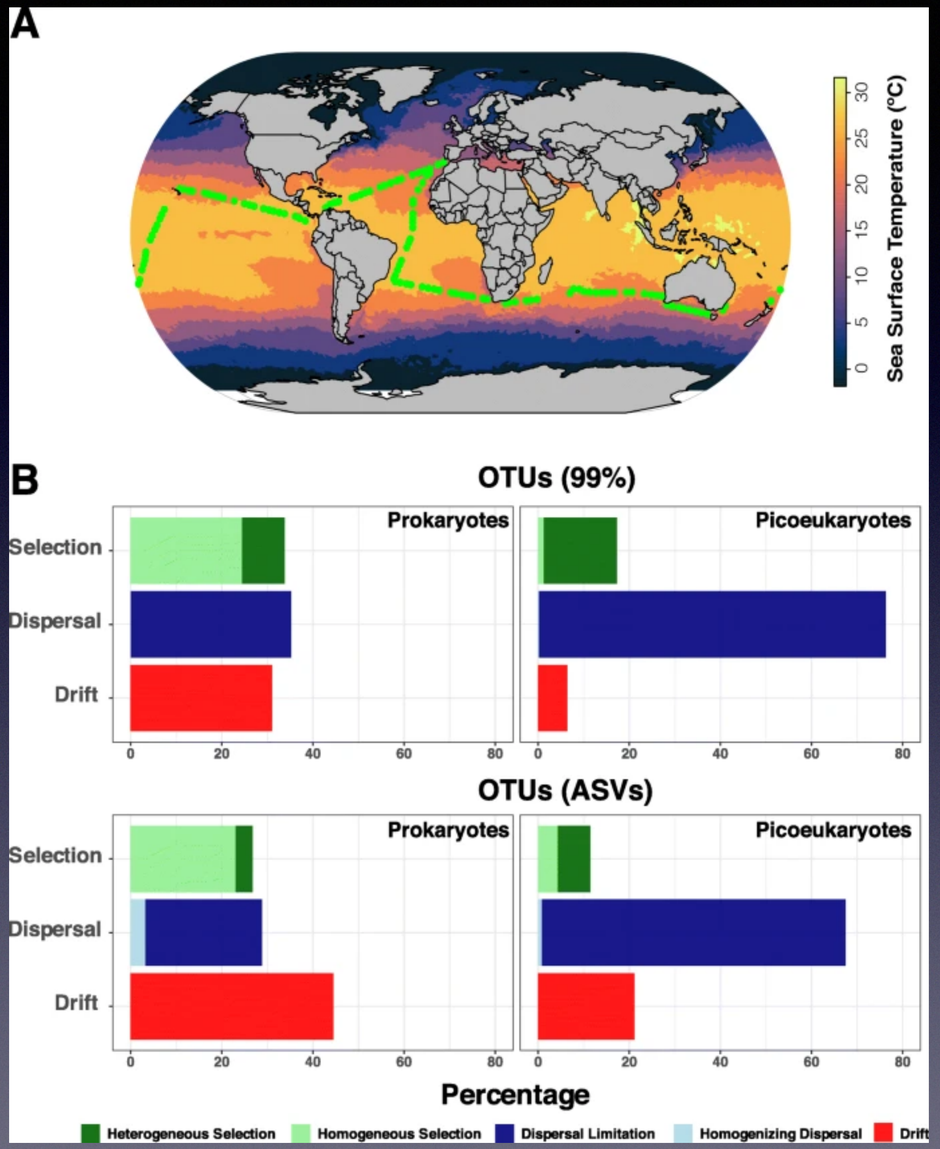
ORIGINAL ARTICLE

Quantifying community assembly processes and identifying features that impose them

James C Stegen[1], Xueju Lin[1,2], Jim K Fredrickson[1], Xingyuan Chen[3], David W Kennedy[1], Christopher J Murray[4], Mark L Rockhold[3] and Allan Konopka[1]

[1]*Biological Sciences Division, Pacific Northwest National Laboratory, Richland, WA, USA;* [2]*School of Biology, Georgia Institute of Technology, Atlanta, GA, USA;* [3]*Hydrology Group, Pacific Northwest National Laboratory, Richland, WA, USA and* [4]*Department of Geosciences, Pacific Northwest National Laboratory, Richland, WA, USA*

R code
https://github.com/stegen/Stegen_etal_ISME_2013



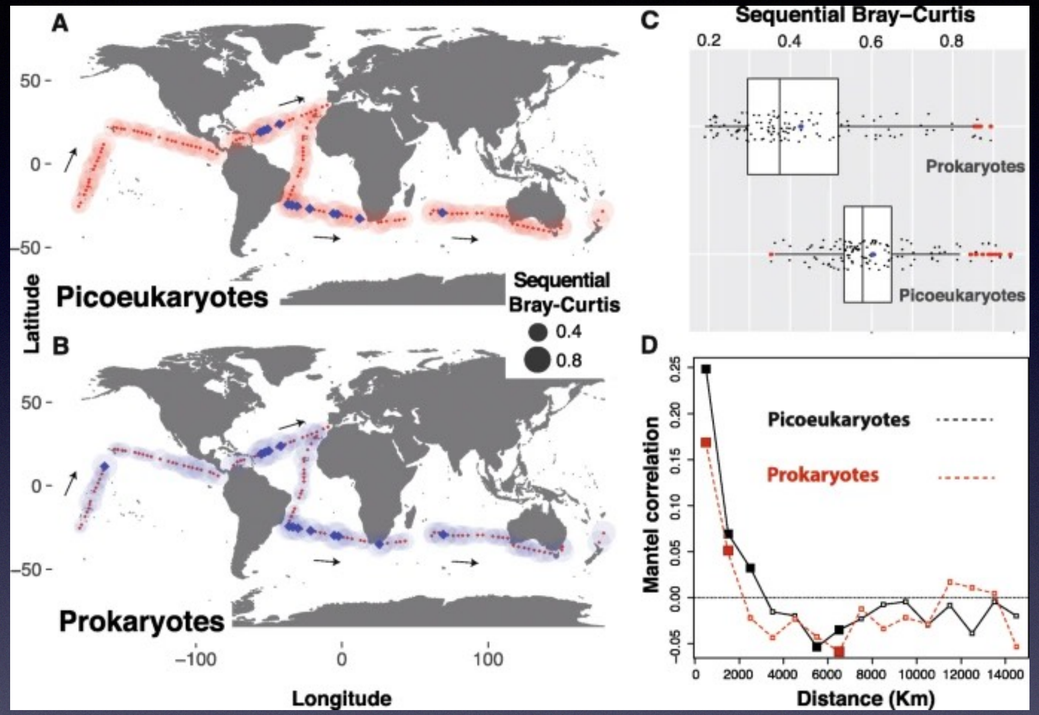
Logares *et al. Microbiome* (2020) 8:55
https://doi.org/10.1186/s40168-020-00827-8

Microbiome

RESEARCH Open Access

Disentangling the mechanisms shaping the surface ocean microbiota

Ramiro Logares[1,2*], Ina M. Deutschmann[1], Pedro C. Junger[3], Caterina R. Giner[1,4], Anders K. Krabberød[2], Thomas S. B. Schmidt[5], Laura Rubinat-Ripoll[6], Mireia Mestre[1,7,8], Guillem Salazar[1,9], Clara Ruiz-González[1], Marta Sebastián[1,10], Colomban de Vargas[6], Silvia G. Acinas[1], Carlos M. Duarte[11], Josep M. Gasol[1,12] and Ramon Massana[1]



Logares *et al. Microbiome* (2020) 8:55
https://doi.org/10.1186/s40168-020-00827-8

Microbiome

RESEARCH Open Access

Disentangling the mechanisms shaping the surface ocean microbiota

Ramiro Logares[1,2*], Ina M. Deutschmann[1], Pedro C. Junger[3], Caterina R. Giner[1,4], Anders K. Krabberød[2], Thomas S. B. Schmidt[5], Laura Rubinat-Ripoll[6], Mireia Mestre[1,7,8], Guillem Salazar[1,9], Clara Ruiz-González[1], Marta Sebastián[1,10], Colomban de Vargas[6], Silvia G. Acinas[1], Carlos M. Duarte[11], Josep M. Gasol[1,12] and Ramon Massana[1]

Tutorial: follow all the steps in the presentation

Script: https://github.com/krabberod/BIO9905MERG1_V21/blob/main/community.ecology/Comm.Ecology.R.BIO9905MERG1_V21.R

```r
#Files
# All content in https://github.com/krabberod/BIO9905MERG1_V21/tree/main/community.ecology

# OTU table raw
otu.tab<-read_tsv("https://raw.githubusercontent.com/krabberod/BIO9905MERG1_V21/main/Dada2_Pipeline/dada2_results/OTU_table.tsv")
dim(otu.tab) # 2107   26
#Let's reorder the table
otu.tab<-otu.tab[,c(17,19:26,1:16,18)]
#We assign to rownames the OTU names
otu.tab <- column_to_rownames(otu.tab, var = "OTUNumber") # %>% as_tibble()
dim(otu.tab) # 2107   25 <- Dimensions of the table
otu.tab.simple<-otu.tab[,1:8] # We'll need this table for community ecology analyses
#We transpose the table, as this is how Vegan likes it
otu.tab.simple<-t(otu.tab.simple)
otu.tab.simple # => ready to use <=

# OTU table rarefied
otu.tab.simple.ss.nozero<-read_tsv("https://raw.githubusercontent.com/krabberod/BIO9905MERG1_V21/main/community.ecology/otu.tab.simple.ss.nozero.
                                   tsv", col_names = T)
otu.tab.simple.ss.nozero<-as.data.frame(otu.tab.simple.ss.nozero) # transform to dataframe
rownames(otu.tab.simple.ss.nozero)<-otu.tab.simple.ss.nozero[,1] # fix row names
otu.tab.simple.ss.nozero<-otu.tab.simple.ss.nozero[,-1] # fix row names
otu.tab.simple.ss.nozero # => ready to use <=

# Bray Curtis distance matrix
otu.tab.simple.ss.nozero.bray<-read_tsv("https://raw.githubusercontent.com/krabberod/BIO9905MERG1_V21/main/community.ecology/otu.tab.simple.ss.
                                        nozero.mat.tsv", col_names = T)
otu.tab.simple.ss.nozero.bray<-as.dist(otu.tab.simple.ss.nozero.bray) # Transform to a distance object
otu.tab.simple.ss.nozero.bray # <- ready to use

# OTU table centered log-ratio transformed
otu.tab.simple.gbm.clr<-read_tsv("https://raw.githubusercontent.com/krabberod/BIO9905MERG1_V21/main/community.ecology/otu.tab.simple.gbm.clr.tsv",
                                 col_names = T)
otu.tab.simple.gbm.clr<-as.data.frame(otu.tab.simple.gbm.clr) # transform to dataframe
rownames(otu.tab.simple.gbm.clr)<-otu.tab.simple.gbm.clr[,1]# fix row names
otu.tab.simple.gbm.clr<-otu.tab.simple.gbm.clr[,-1] # fix row names
otu.tab.simple.gbm.clr # => ready to use <=

# Aitchison distance matrix
otu.tab.simple.gbm.clr.euclidean<-read_tsv("https://raw.githubusercontent.com/krabberod/BIO9905MERG1_V21/main/community.ecology/otu.tab.simple.gbm.
                                           clr.mat.tsv", col_names = T)
otu.tab.simple.gbm.clr.euclidean<-as.dist(otu.tab.simple.gbm.clr.euclidean) # Transform to a distance object
otu.tab.simple.gbm.clr.euclidean # => ready to use <=

# Environmental table
bbmo.metadata.course<-read_tsv("https://raw.githubusercontent.com/krabberod/BIO9905MERG1_V21/main/community.ecology/bbmo.metadata.course.tsv",
                               col_names = T)
bbmo.metadata.course<-as.data.frame(bbmo.metadata.course)
rownames(bbmo.metadata.course)<-bbmo.metadata.course[,1]
bbmo.metadata.course<-bbmo.metadata.course[,-1]
bbmo.metadata.course # <- Ready to use, table to z-score transformed
#We transform variables 1:15 using z-scores to have comparable ranges of variation
bbmo.metadata.course.15vars<-bbmo.metadata.course[1:15,] #We select continuous variables
bbmo.metadata.course.15vars[]<- lapply(bbmo.metadata.course.15vars, as.character) #We transform the datatype to characters
bbmo.metadata.course.15vars[]<- lapply(bbmo.metadata.course.15vars, as.numeric) #We transform to numeric
bbmo.metadata.course.15vars.zscores<-scale(t(bbmo.metadata.course.15vars), center = T, scale = T) #zscore transform
bbmo.metadata.course.15vars.zscores # => ready to use <=
```

THE END