Trait-based approach in ecology

Aurele Toussaint (aurele.toussaint@cnrs.fr)

Autumn 2024

Contents

Functional traits

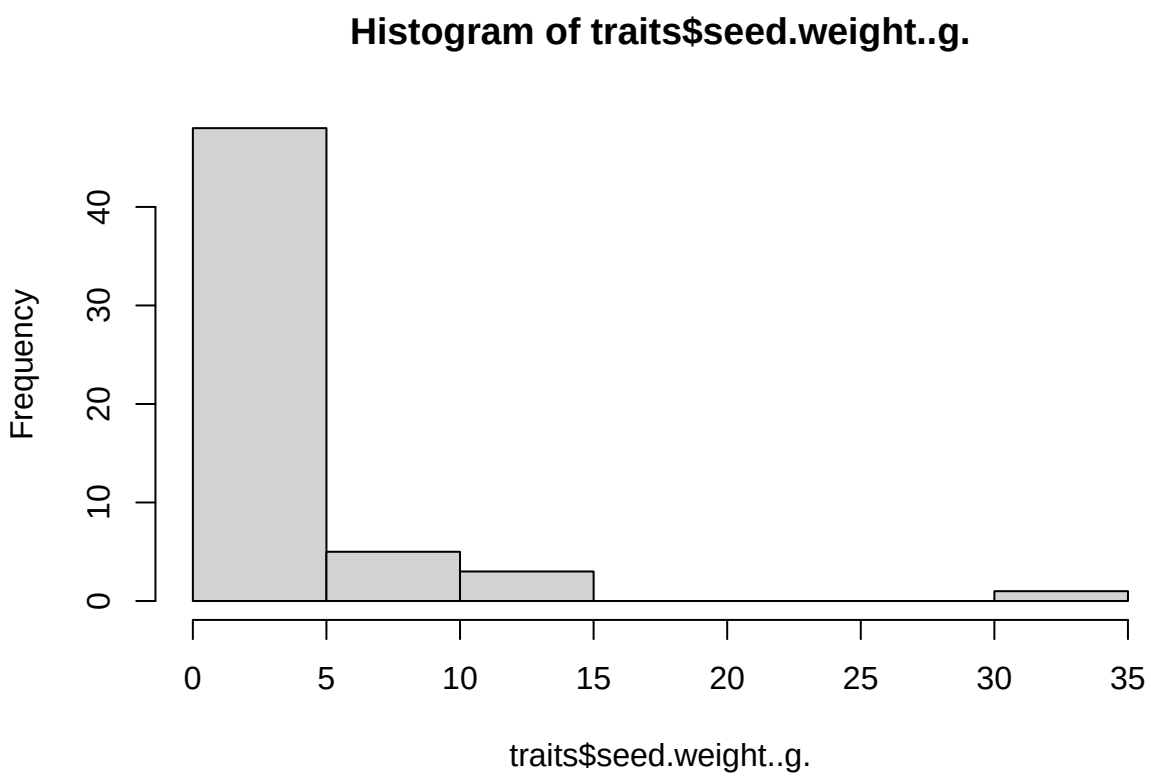
Besides the number and abudance of taxa, a very important aspect of ecological communities includes the ecological differences between the organisms composing them. This variation can be expressed in terms of functional traits, which are measurable features of organisms that influence how they respond to and affect the environment and how they interact with other organisms. If we have two communities with five taxa each, but the taxa present in the first one have very different traits while those present inthe second are very similar, we have higher functional diversity in the first community.

```
load("community.rda")
traits <- read.table("vas.plant.traits.txt")
# seed weight mg, clonal spread classes, leaf size classes. NB! Many NA values!
head(traits)
##   seed.weight..g. clonal leaf
## 1            6.40      1   NA
## 2            2.36      1    5
## 3            4.00      2    4
## 4            3.16      2   NA
## 5              NA      1   NA
## 6            2.56      3    4
```

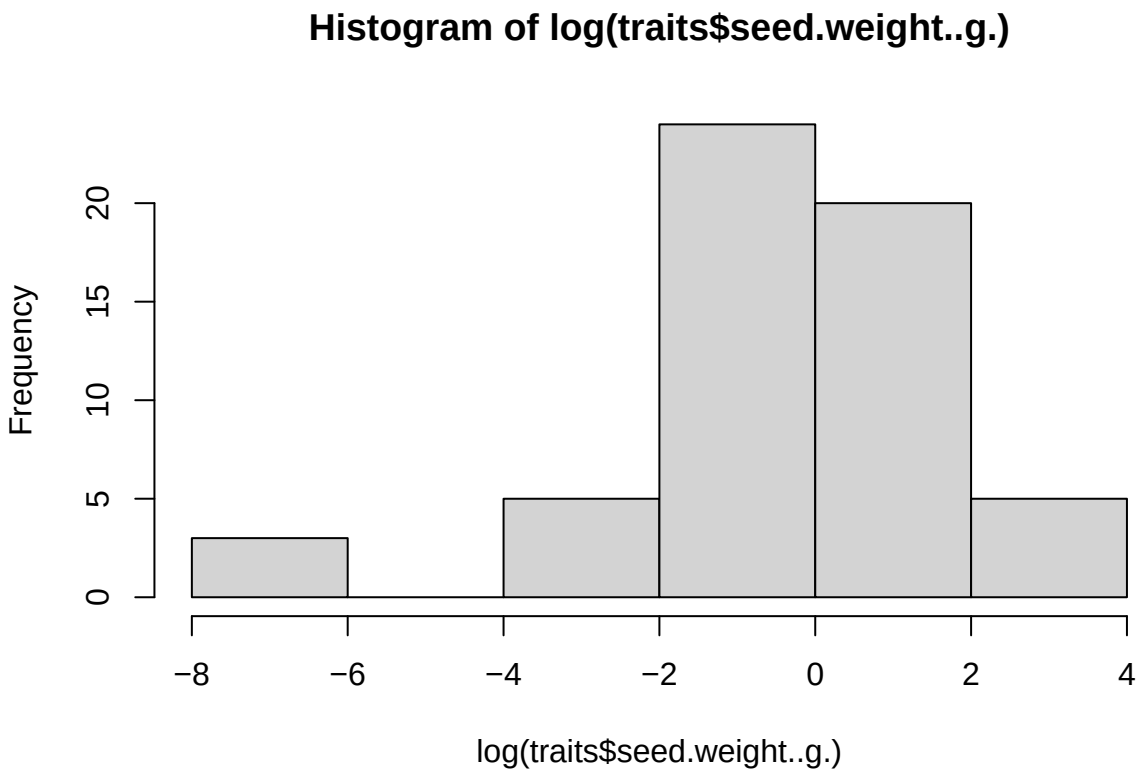
Weighted mean

The average trait value of a sample, weighted according to species' abundances. It shows which trait value is the most common. If our traits have a very skewed distribution (which is generally the case with seed mass), it is recommendable to work with the logarithm, to avoid species with very high values being extremely influential (the mean is sensitive to outliers):

```r
hist(traits$seed.weight..g.) # very skewed
```




```r
hist(log(traits$seed.weight..g.)) # looks better
```

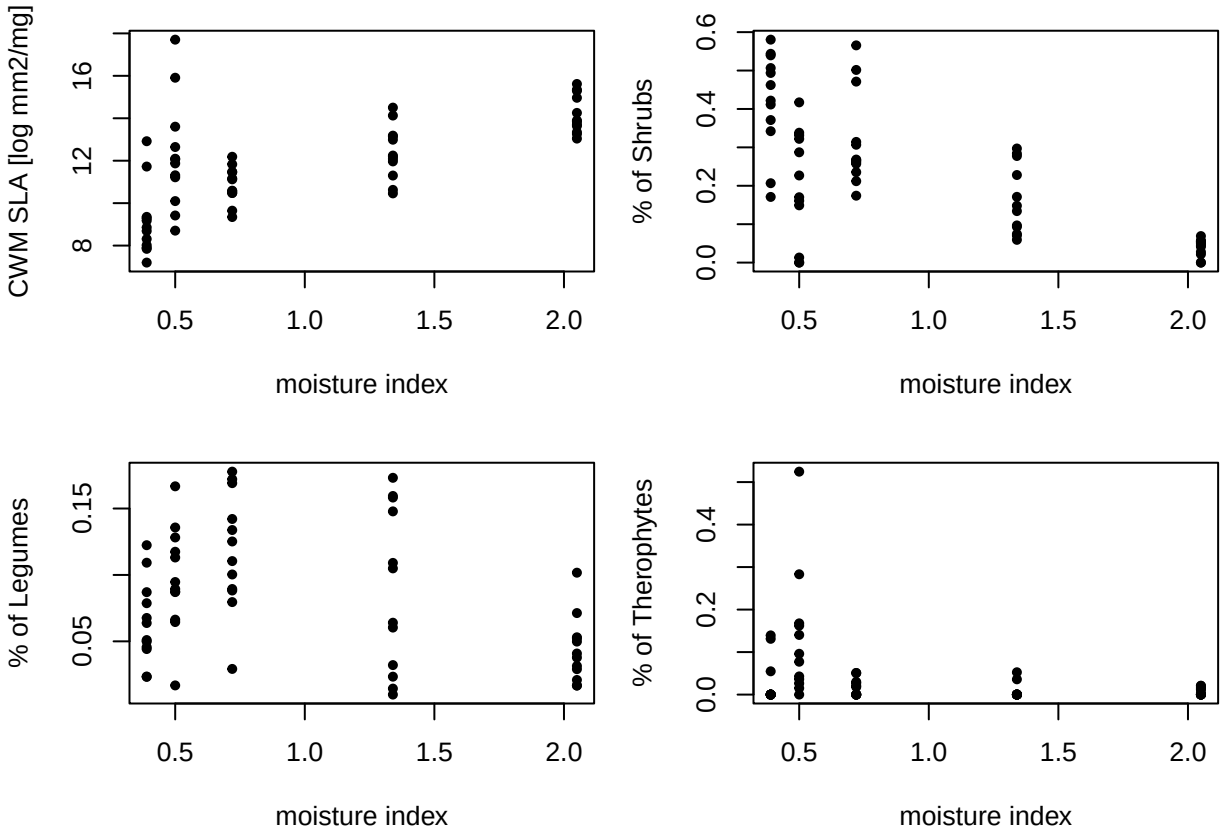



```
traits$seed.weight..g.  <- log(traits$seed.weight..g.)

mean.seed <- numeric() # empty numeric object
for (i in 1:nrow(vas.plants)) {
  mean.seed[i] <- weighted.mean(traits$seed.weight..g.,
                                w = vas.plants[i, ],
                                na.rm = T)
}

mean.seed #In the log-scale
##  [1]  0.7587248981 -1.0523588258  0.0755782788 -0.3534966954 -0.4341183939
##  [6] -1.8067751943  1.3512422403  0.2903592473 -2.3447292993  0.2643987001
## [11]  0.4283597436 -0.0004662142  0.6720201664 -0.0751237178  0.7660636021
## [16]  0.2171485442  0.7208543721  0.3357505635 -0.1310540885  0.0600293211
## [21] -0.7430225683  0.0250576883 -0.5571641625 -0.9151129339 -0.0155489843
## [26] -0.0655949418  0.0077888053 -0.5051300122 -0.1111478318  0.6928334449
exp(mean.seed) #In the original scale
##  [1] 2.13555144 0.34911328 1.07850765 0.70222831 0.64783555 0.16418274
##  [7] 3.86222036 1.33690768 0.09587315 1.30264746 1.53473809 0.99953389
## [13] 1.95818920 0.92762872 2.15128127 1.24252866 2.05618921 1.39899002
## [19] 0.87717033 1.06186768 0.47567398 1.02537427 0.57283122 0.40047140
## [25] 0.98457128 0.93651013 1.00781922 0.60342712 0.89480646 1.99937263

mean.clonal <- numeric() # empty numeric object
for (i in 1:nrow(vas.plants)) {
  mean.clonal[i] <- weighted.mean(traits[, 2], vas.plants[i, ], na.rm = T)
```

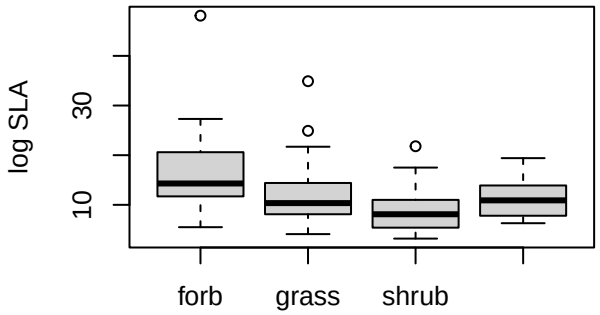
```
}
mean.clonal
##  [1] 1.742857 2.076923 1.581395 2.163636 2.705882 2.000000 1.812500 1.136364
##  [9] 2.800000 1.679245 1.571429 1.685714 1.800000 1.761905 2.000000 1.722222
## [17] 1.685714 1.695652 2.200000 1.543860 2.764706 1.693878 2.250000 2.500000
## [25] 1.833333 1.900000 2.250000 2.473684 2.333333 1.866667

mean.leaf <- numeric()
for (i in 1:nrow(vas.plants)) {
  mean.leaf[i] <- weighted.mean(traits[, 3], vas.plants[i, ], na.rm = T)
}
mean.leaf
##  [1] 3.517241 2.923077 3.303030 3.545455 3.080000 3.523810 3.722222 3.705882
##  [9] 3.000000 3.333333 3.250000 3.333333 3.300000 3.108108 3.368421 3.562500
## [17] 3.470588 3.375000 3.000000 3.794118 3.000000 3.577778 3.000000 3.000000
## [25] 3.382353 3.222222 3.250000 3.166667 3.125000 3.400000
```

Correlation with environment.

```
spxp <- read.table("speciesXplotsNE.txt", row.names = 1, head = T)
spxt <- read.table("speciesXtraitsNE.txt", row.names = 1, head = T)
library(FD)
resCWM = functcomp(spxt, log(t(spxp) + 1), CWM.type = "all")
envxp <- read.table("environXplotsNE.txt", row.names = 1, head = T)
par(mfrow = c(2, 2))
par(mar = c(4, 4, 2, 1))
plot(envxp$moisture.index, resCWM$SLA, xlab = "moisture index",
ylab = "CWM SLA [log mm2/mg]", pch = 20)
plot(envxp$moisture.index, resCWM$GrowhtForm_shrub, xlab = "moisture index",
ylab = "% of Shrubs", pch = 20)
plot(envxp$moisture.index, resCWM$LEG_1, xlab = "moisture index",
ylab = "% of Legumes", pch = 20)
plot(envxp$moisture.index, resCWM$LF_Th, xlab = "moisture index",
ylab = "% of Therophytes", pch = 20)
```



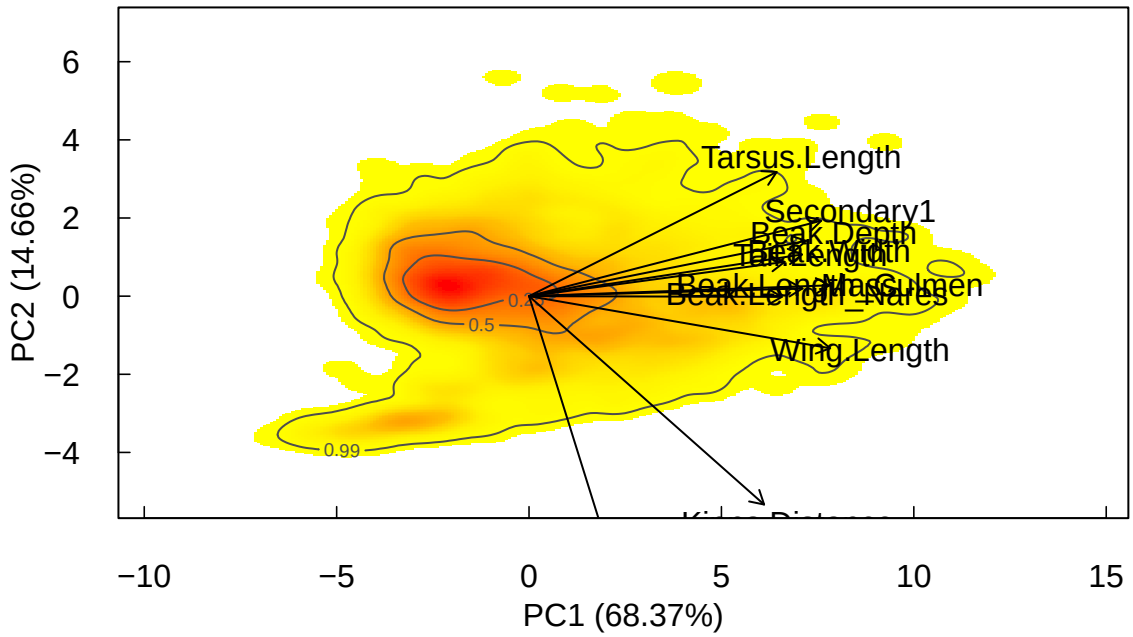

```r
boxplot(spxt$SLA ~ spxt$GrowhtForm, ylab = "log SLA", xlab = "")
summary(lm(resCWM$SLA ~ moisture.index * grazing, data = envxp))
##
## Call:
## lm(formula = resCWM$SLA ~ moisture.index * grazing, data = envxp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1564 -1.0146  0.0383  0.7592  5.8350
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.5733     0.6290  12.040  < 2e-16 ***
## moisture.index             3.5435     0.5347   6.627 1.43e-08 ***
## grazing                    1.8867     0.4872   3.872 0.000284 ***
## moisture.index:grazing    -1.2485     0.4142  -3.014 0.003867 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.623 on 56 degrees of freedom
## Multiple R-squared:  0.5227, Adjusted R-squared:  0.4971
## F-statistic: 20.44 on 3 and 56 DF,  p-value: 4.509e-09
```



Functional space of taxa

Besides mean values, we can also look at how large is the functional space covered by a taxa within a site. Size of the functional space describes functional diversity. NB! Related to species richness!

```
library(funspace)
#1 TRAITS
avonet  =  readxl::read_excel("AVONET.xlsx", sheet  =  "AVONET3_BirdTree", col_types  =  c("text", "tex
avonet  =  avonet[-which(avonet$Family3 == "Apterygidae"), ]
avonet  =  as.data.frame(avonet)
avonet$Species3  =  gsub(" ", "_", avonet$Species3)
traitsp  =  avonet$Species3
toremSP  =  c('Atlantisia_rogersi', 'Casuarius_bennetti', 'Casuarius_casuarius', 'Casuarius_unappendicul

avonet  =  avonet[order(avonet$Species3), ]
avonet  =  avonet[!(avonet$Species3%in%toremSP), ]

imputed.traits  =  scale(log10(avonet[, -c(1:3)]))

# Run PCA
pca.trait  =  princomp(imputed.traits, cor  =  TRUE)

# Building the functional trait space (using the first two PCs)
trait_space_global  =  funspace(x  =  pca.trait, PCs  =  c(1, 2), n_divisions  =  300)
plot(x  =  trait_space_global, type  =  "global", quant.plot  =  TRUE, arrows  =  TRUE, arrows.length  =
```



Community assembly tests with traits

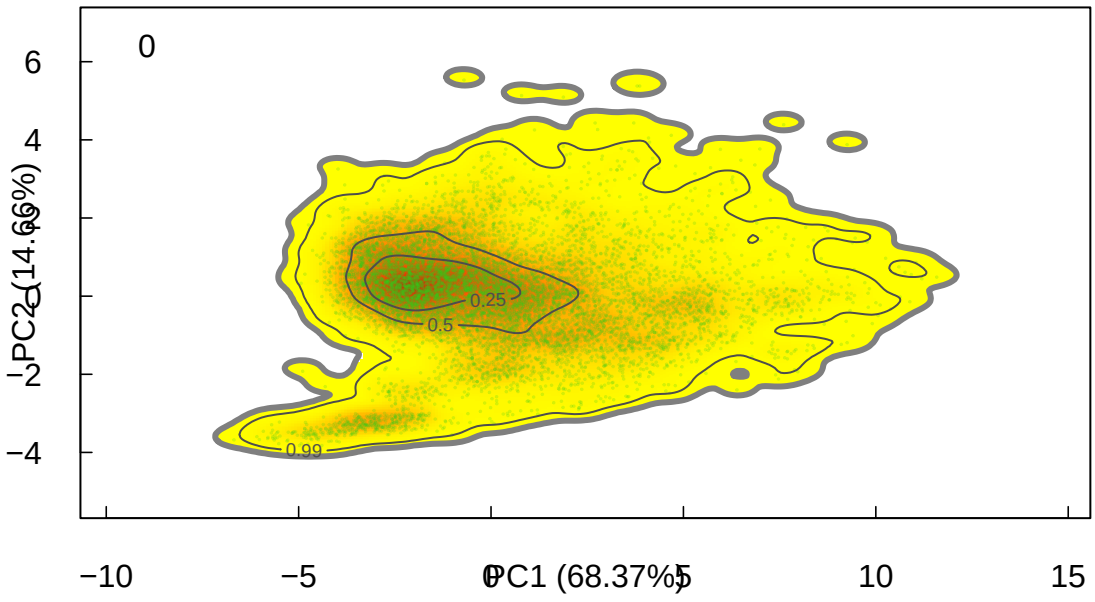
With traits we can make community assembly tests. These tests check if functional trait values are randomly assembled in a community, or if taxa are more similar (or dissimilar) than expected by chance. If there are more similar species, then often it has been explained by habitat filtering (similar taxa fits to similar habitats). On the other hand, when co-existing species are more dissimilar than expected, it has been explained by competition – similar species compete most strongly and might exclude each other.

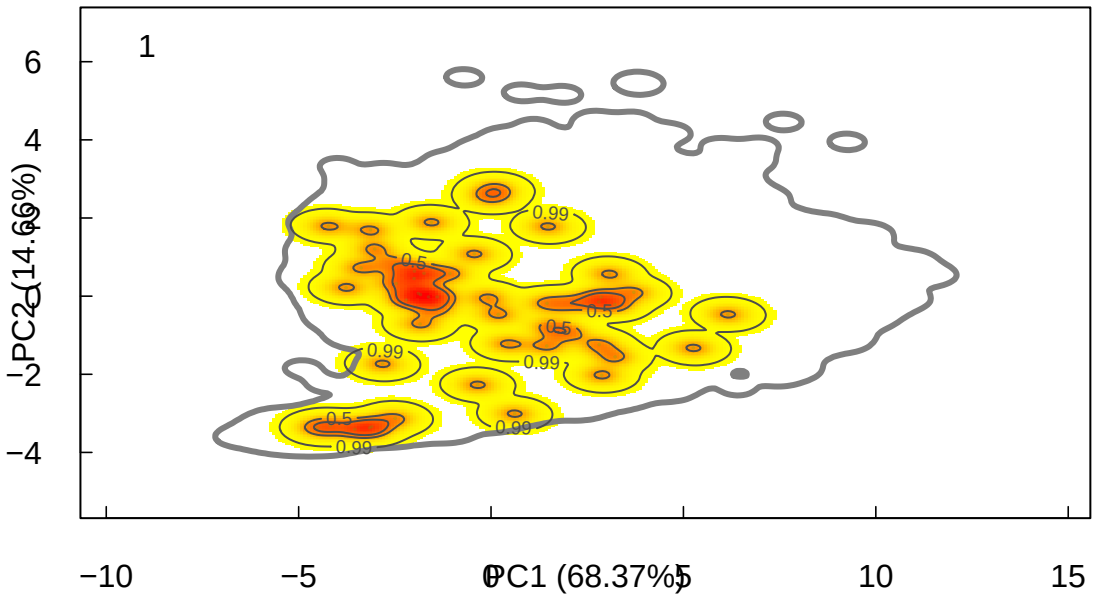
```r
# Install necessary package for convex hull calculation
if (!require("geometry")) install.packages("geometry", repos = "http://cran.us.r-project.org")
library(geometry)
library(FD)
library(funspace)

# Step 1: Define a function to calculate FRic using the convex hull
calculate_fric <- function(traits,comm) {
  traits = traits[names(which(apply(comm,2,sum)>0)),]
  comm = comm[,names(which(apply(comm,2,sum)>0))]
  # Check if there are enough points to form a convex hull
  if (nrow(traits) > ncol(traits)) {
    fric <- dbFD(traits,comm,calc.FRic = TRUE)$FRic
  } else {
    fric <- 0  # If not enough points, FRic is 0
  }
  return(fric)
```

```
}

# Step 2: Create a null model function to randomize species' traits and recalculate FRic
null_model_fric <- function(traits, comm,  num_simulations = 999) {
  null_fric_values <- numeric(num_simulations)

    fd.rand = list()
    for (rand in 1:num_simulations){
      sample.comm = comm
      sample.comm[] = 0
      for(j in 1:nrow(comm)){
        sample.comm[j, names(sample(comm[j,],sum(comm[j,])))] = 1
      }
      null_fric_values[rand]  = calculate_fric(traits,sample.comm)
    }

  return(null_fric_values)
}

# Step 3: Compare observed FRic with the null model
compare_fric_with_null <- function(traits,comm, num_simulations = 999) {
  # Calculate observed FRic
  observed_fric <- calculate_fric(traits,comm)

  # Generate null FRic values using the null model
  null_fric_values <- null_model_fric(traits,comm, num_simulations)

  # Calculate ses and p-value: Proportion of null FRic values greater than or equal to observed FRic
  ses <- (observed_fric - mean(null_fric_values)) / sd(null_fric_values)
  p_value <- mean(null_fric_values >= observed_fric)

  # Return the results
  return(list(observed_fric = observed_fric, null_fric_values = null_fric_values, ses = ses, p_value = p
}

# Step 4: Example usage
traits <- pca.trait$scores[,c(1,2)]
rownames(traits)  = avonet$Species3

comm <- matrix(0,nr = 1, nc = nrow(traits),dimnames = list(c("Comm.X"),rownames(traits)))
comm["Comm.X",sample(colnames(comm),50)] = 1

# Compare observed FRic with null model
result <- compare_fric_with_null(traits, comm, num_simulations = 99)
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
```

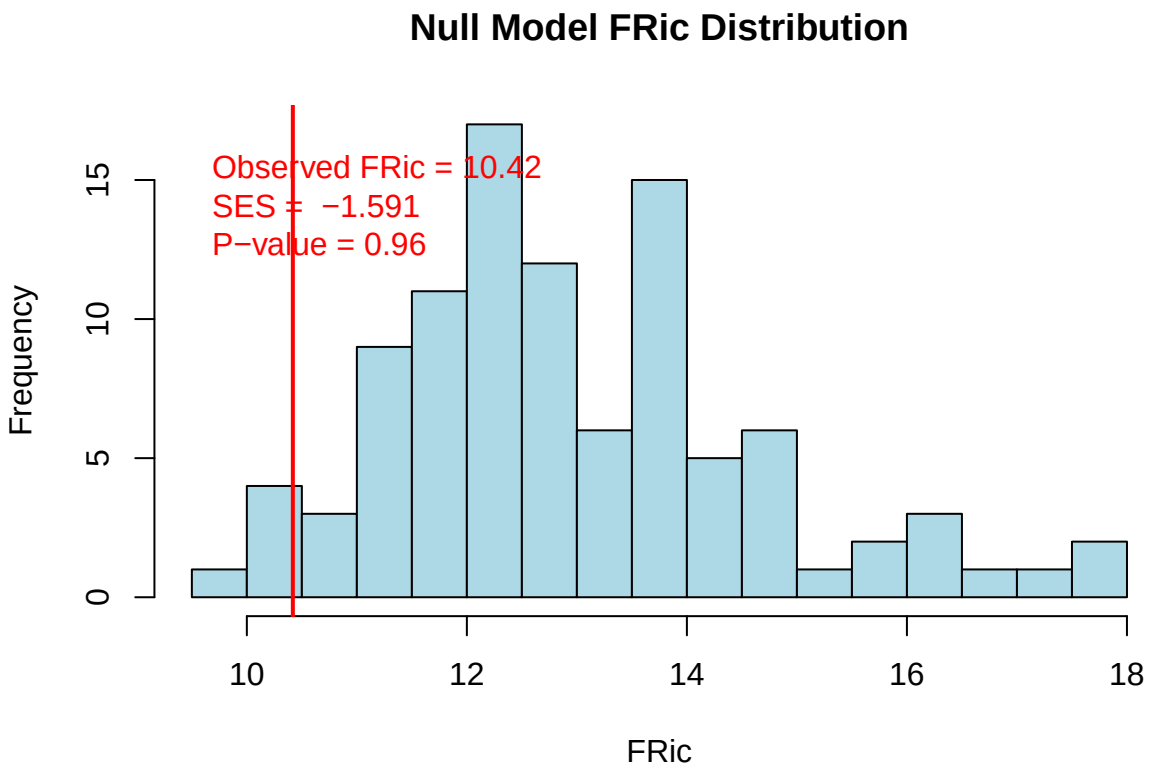
```
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
```

```
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.
## FRic: No dimensionality reduction was required. The 2 PCoA axes were kept as 'traits'.

# Print the results
trait_space_comm  =  funspace(x  =  pca.trait, PCs  =  c(1, 2),
                              group.vec  =  comm[1,,drop = F],
                              n_divisions  =  300)

plot(x = trait_space_comm, type = "groups",
     quant.plot = TRUE, globalContour = T,
     pnt = T,
     pnt.cex = 0.1,
     pnt.col = rgb(0.2, 0.8, 0.1, alpha = 0.2), axis.title.line = 1)
```


0
0.25
0.5
0.99
6
4
2
0
−4
−10
−5
0
5
10
15
PC2 (14.66%)
PC1 (68.37%)




```
# Plot the distribution of null FRic values
  hist(result$null_fric_values, main = "Null Model FRic Distribution", xlab = "FRic", breaks = 20, col =
  abline(v = result$observed_fric, col = "red", lwd = 2)
  legend("topleft", legend = paste("Observed FRic =", round(result$observed_fric, 2), "\nSES = ",round(
```




```
print(paste("Observed FRic:", result$observed_fric))
## [1] "Observed FRic: 10.4180056981255"
print(paste("P-value from null model:", result$p_value))
## [1] "P-value from null model: 0.95959595959596"
```