Basics in Functional Diversity

Master FBE · CRBE · Université de Toulouse

Covers functional traits and databases, functional space construction (PCA/PCoA), visualisation with funspace, CWM, FD indices, TPD framework, functional beta diversity, RLQ analysis, functional vulnerability, and null models.

1. Functional traits and databases

Type	Examples	Database
Morphological	Body length, fin area, beak depth, SLA	FishMORPH, AVONET, TRY
Physiological	Metabolic rate, thermal tolerance	COMPADRE, COMADRE
Life-history	Age at maturity, clutch size, longevity	COMPADRE, AmphiBIO
Diet/behaviour	Trophic level, habitat use, activity	EltonTraits, IUCN

Accessing databases in R:

```
# FishMORPH (Brosse et al. 2021, GEB)
# Download from figshare; load as CSV
fish_traits <- read.csv("FishMORPH_traits.csv")

# AVONET birds (Tobias et al. 2022)
library(readxl)
avonet <- read_excel("AVONET.xlsx",sheet="AVONET1_BirdLife")

# TRY plant traits (online request)
# Access via API: rtry package
library(rtry)
rtry_import(input=read.csv("TRY_data.txt",sep="\t"))

# EltonTraits (Wilman et al. 2014)
elton <- read.table("BirdFuncDat.txt",header=TRUE,sep="\t")
```

2. Checking and cleaning trait data

```
# Check completeness
complete_pct <- colMeans(!is.na(sp_traits))*100
data.frame(trait=names(complete_pct),
           pct_complete=round(complete_pct,1))

# Visualise missing data
library(naniar)
vis_miss(sp_traits)              # heatmap of NAs
gg_miss_var(sp_traits)           # bar chart of NAs per variable
miss_var_summary(sp_traits)      # table

# Outlier detection
for(col in names(sp_traits[,sapply(sp_traits,is.numeric)])) {
```

```
  z <- scale(sp_traits[[col]])
  outliers <- which(abs(z) > 3)
  if(length(outliers)>0)
    cat(col,":",length(outliers),"outliers\n")
}
```

3. Building the functional space

Method	Input	Advantages	Limitation
PCA	Raw traits (numeric)	Interpretable loadings	Only quantitative traits
PCoA	Gower distance	Any trait type	Negative eigenvalues possible
FAMD	Mixed data	Handles all types natively	Less used in ecology
NMDS	Distance	Non-parametric	No % variance explained

```
library(FD); library(ade4)

# Species-level traits (iris means)
library(dplyr)
sp_traits <- iris|>group_by(Species)|>
  summarise(across(is.numeric,mean))|>column_to_rownames("Species")

# PCA (all quantitative)
pca <- prcomp(sp_traits,scale.=TRUE)
biplot(pca)                         # quick biplot
summary(pca)$importance[2,]        # variance per axis

# PCoA with Gower (mixed traits)
d_gower <- gowdis(sp_traits)        # Gower distance
pcoa    <- cmdscale(d_gower,k=4,eig=TRUE)
eig_pct <- 100*pcoa$eig/sum(abs(pcoa$eig))
cat(sprintf("PC1=%.1f%%  PC2=%.1f%%\n",eig_pct[1],eig_pct[2]))

# How many axes? Rule: cumulative variance > 70%
cumsum(eig_pct)[cumsum(eig_pct)>70][1]
```

4. Visualising functional spaces: funspace

funspace (Carmona et al.) produces KDE-based functional space visualisations that incorporate intraspecific variability when individual-level data are available.

```
library(funspace)

# Individual-level data (iris: 150 plants)
pca <- prcomp(iris[,1:4],scale.=TRUE)

# All species together
funspace(x=pca,axes=c(1,2))

# Per-species density plots
funspace(x=pca,axes=c(1,2),
         group.vec = iris$Species,
         pnt       = TRUE,
         fill.col  = c("#2980b9","#27ae60","#e67e22"))

# Pairwise overlap
fun_overlap(x=pca,axes=c(1,2),group.vec=iris$Species)

# Functional uniqueness per individual
```

```
uniq <- funspaceUniqueness(x=pca,axes=c(1,2))
iris$uniq <- uniq
ggplot(iris,aes(PC1<-pca$x[,1],PC2<-pca$x[,2],
                colour=uniq,shape=Species)) +
  geom_point(size=2.5)+
  scale_colour_gradient(low="#27ae60",high="#c0392b")+theme_bw()
```

5. Community-weighted mean and trait–environment links

CWM links species traits to ecosystem functioning (mass ratio hypothesis). The **fourth-corner** and **RLQ** analyses formally test trait–environment relationships.

```
library(FD)

# Community matrix
comm <- matrix(c(10,2,0, 3,5,2, 0,1,8),nrow=3,byrow=TRUE,
               dimnames=list(c("C1","C2","C3"),rownames(sp_traits)))

# CWM
cwm <- functcomp(sp_traits,comm)
print(round(cwm,2))

# Fourth-corner: trait × environment associations
# Requires: site × species, species × traits, site × environment
library(ade4)
env_df  <- data.frame(temp=c(15,20,25),precip=c(800,600,400))
trait_df <- sp_traits
fc <- fourthcorner(env_df,comm,trait_df,
                   modeltype=6,npermut=999)
plot(fc,alpha=0.05)

# RLQ analysis (ordination of fourth-corner)
rlq <- rlq(dudi.pca(env_df,scannf=FALSE),
           dudi.coa(comm,scannf=FALSE),
           dudi.pca(trait_df,scannf=FALSE),
           scannf=FALSE)
plot(rlq)
```

6. Functional diversity indices

Index	Measures	Range	Ecological meaning
FRic	Volume occupied	0–1	Width of strategies present
FEve	Regularity in space	0–1	Low = clumped strategies
FDiv	Spread of abundance	0–1	High = dominants at periphery
FDis	Mean dist. to centroid	≥0	Weighted by abundance
Rao Q	Expected dissimilarity	≥0	Related to FDis
FSpe	Functional specialisation	0–1	High = peripheral species
FOri	Functional originality	0–1	High = isolated species

```
library(FD)
fd <- dbFD(sp_traits,comm,m=2)

results <- data.frame(
  FRic=fd$FRic,FEve=fd$FEve,FDiv=fd$FDiv,
  FDis=fd$FDis,RaoQ=fd$RaoQ
)
print(round(results,3))
```

```
# Uniqueness, specialisation, originality
library(funrar)
rel_comm <- make_relative(comm)
funrar(sp_traits,rel_comm,
       metric=c("Ui","Si","Ri"))   # uniqueness, specialisation, restrictedness
```

7. Functional beta diversity

Functional beta diversity = variation in functional composition between communities. Analogous to taxonomic beta diversity but in trait space.

```
library(betapart)
library(FD)

# Build functional hypervolumes per community (using FRic)
# Method: Villéger et al. (2011)

# Functional distances between communities
fun_dist <- function(comm, traits, m=2) {
  n <- nrow(comm)
  d <- matrix(0,n,n,dimnames=list(rownames(comm),rownames(comm)))
  for (i in 1:(n-1)) {
    for (j in (i+1):n) {
      # Keep species present in either community
      both <- union(
        colnames(comm)[comm[i,]>0],
        colnames(comm)[comm[j,]>0]
      )
      c2 <- comm[c(i,j), both, drop=FALSE]
      tr <- traits[both,,drop=FALSE]
      fd_all  <- dbFD(tr,c2,m=m,print.pco=FALSE)$FRic
      fd_both <- sum(fd_all)
      fd_int  <- dbFD(tr,matrix(1,1,length(both),
                     dimnames=list("all",both)),
                     m=m,print.pco=FALSE)$FRic
      d[j,i] <- d[i,j] <- 1 - fd_int/fd_both
    }
  }
  as.dist(d)
}
# Note: computationally intensive; use for small datasets
```

8. Trait Probability Densities (TPD framework)

Carmona et al. (2016) Ecology. Represents diversity as probability density functions over trait space.

Each species = a distribution over trait space (not a point). When individual data are available, TPDs capture intraspecific variability. Communities are obtained by weighting species TPDs by abundance.

```
library(TPD)

# Species TPDs from individual data
TPDs_sp <- TPDs(
  species = iris$Species,
  traits  = iris[,1:2],
  alpha   = 0.95
)

# Community TPDs
TPDc_obj <- TPDc(TPDs=TPDs_sp, sampUnit=comm)

# Functional richness, evenness, divergence
fd_tpd <- REND(TPDc_obj)
print(fd_tpd)

# Overlap
overlapF(TPDs_sp, type="communities", TPDc=TPDc_obj)

# Change between scenarios
TPDc2 <- TPDc(TPDs=TPDs_sp,
              sampUnit=comm*0.5+0.1)  # simulate change
from <- REND(TPDc_obj)
to   <- REND(TPDc2)
data.frame(delta_FRic=to$communities$FRichness-from$communities$FRichness)
```

9. Functional vulnerability and extinction risk

Species at risk of extinction may represent unique functional strategies. Functional vulnerability = FD potentially lost under extinction scenarios.

```
library(FD)

# Simulate: remove threatened species
threatened <- sample(rownames(sp_traits),1)  # random for demo
current_fd   <- dbFD(sp_traits,comm,m=2,print.pco=FALSE)$FRic

# Remove threatened species
sp_ext  <- sp_traits[rownames(sp_traits)!=threatened,,drop=FALSE]
comm_ext <- comm[,colnames(comm)!=threatened]
# Fill to same size
future_fd <- tryCatch(
  dbFD(sp_ext,comm_ext,m=2,print.pco=FALSE)$FRic,
  error = function(e) rep(NA,nrow(comm))
)

fun_loss <- (current_fd - future_fd) / current_fd * 100
cat(sprintf("FRic loss after removing %s: %.1f%%\n",
            threatened, mean(fun_loss,na.rm=TRUE)))
```

10. Null models and SES

SES removes the confounding effect of species richness on FD. Use **SES > +1.96** (overdispersion) or **SES < –1.96** (clustering).

```
# Permutation null model
set.seed(42)
null_dist <- replicate(499, {
  tr_null <- sp_traits[sample(nrow(sp_traits)),,drop=FALSE]
  rownames(tr_null) <- rownames(sp_traits)
  tryCatch(
    dbFD(tr_null,comm,m=2,print.pco=FALSE)$FRic,
    error=function(e) rep(NA,nrow(comm))
  )
})

FRic_obs <- fd$FRic
SES  <- (FRic_obs - rowMeans(null_dist,na.rm=TRUE)) /
         apply(null_dist,1,sd,na.rm=TRUE)
pval <- rowMeans(sweep(null_dist,1,FRic_obs,">="),na.rm=TRUE)

data.frame(
  community = rownames(comm),
  FRic      = round(FRic_obs,3),
  SES       = round(SES,2),
  pval      = round(pval,3),
  pattern   = ifelse(SES>1.96,"overdispersed",
              ifelse(SES< -1.96,"clustered","random"))
)
```

References: Villeger et al. (2008) Ecology · Carmona et al. (2016) Ecology · Toussaint et al. (2021) Nat. Comms · Brosse et al. (2021) GEB · Dray et al. (2014) Ecology (RLQ/fourth-corner)