

Simulates assemblages and calculate FD metrics.  
Supplementary material in: Species and functional diversity accumulate differently in terrestrial mammals

Brunno F. Oliveira<sup>1,2</sup>; Antonin Machac<sup>3,4,5</sup>; Gabriel C. Costa<sup>1</sup>; Thomas M. Brooks<sup>6</sup>; Ana D. Davidson<sup>7</sup>; Carlo Rondinini<sup>8</sup>; Catherine H. Graham<sup>2</sup>;

*Global Ecology and Biogeography, 2016 - Submitted / 1 Departamento de Ecologia, Universidade Federal do Rio Grande do Norte, Natal, Brasil. / 2 Department of Ecology and Evolution, Stony Brook University, Stony Brook, New York 11794, USA. / 3 Center for Macroecology, Evolution, and Climate, Natural History Museum of Denmark, Universitetsparken 15, DK 2100 Copenhagen, Denmark. / 4 Department of Ecology, Charles University, Vinična 7, 128 44 Praha 2, Czech Republic. / 5 Center for Theoretical Study, Charles University and Academy of Sciences of the Czech Republic, Jilská 1, 110 00 Praha 1, Czech Republic. / 6 International Union for Conservation of Nature, 1196 Gland, Switzerland / 7NatureServe, Arlington, VA, USA. / 8 Global Mammal Assessment Program, Department of Biology and Biotechnologies, Sapienza Universita di Roma, Rome, Italy.* / **Corresponding author:** ([brunno.oliveira@me.com](mailto:brunno.oliveira@me.com))

## Contents

<b>1. Packages versions:</b>	<b>2</b>
<b>2. Load functions:</b>	<b>2</b>
<b>3. Simulate assemblages and calculate FD metrics:</b>	<b>3</b>
3.1 <i>Experiment 1:</i> Simulate trait values following a standardized normal distribution . . . . .	3
3.2 <i>Experiment 2:</i> Simulate trait values following a normal distribution . . . . .	4
3.3 <i>Experiment 3:</i> Simulate trait values following a log normal distribution . . . . .	5
3.4 <i>Experiment 4:</i> Simulate trait values following a exponential distribution . . . . .	6

---

This document follows the principles of reproducible research (Peng, 2011). This document was generated in **R studio** with **knitr** package.

---

## 1. Packages versions:

We used R version 3.2.2 (2015-08-14) and the following packages:

```
library(FD)
```

## 2. Load functions:

```
# Modified version of simul_dbFD function from FD package
# Available at: https://github.com/oliveirab/simulation_traits/blob/master/simul_dbFB_mod.R
# The modified version of the simul.dbFD function from FD package allows the choice of
# parameters to generate trait values, which is not possible with the original simul.dbFD
# function.
# To simulate traits using simul.dbFD.mod insert parameters in the argument tr.par, separated
# by comma, as follow:
# If tr.method = 'norm' ... tr.par = c(mean, sd)
# If tr.method = 'lnorm' ... tr.par = c(mean, sd)
# If tr.method = 'unif' ... tr.par = c(min, max)
# If tr.method = 'exp' ... tr.par = rate
source("simul_dbFB_mod.R")

# Pairs correlation
# P-value and r coefficient
panel.cor <- function(x, y, digits=2, cex.cor)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y, method = "spearman"))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  test <- cor.test(x,y,method = "spearman")
  Signif <- ifelse(round(test$p.value,3)<0.001,"p<0.001",paste("p=",round(test$p.value,3)))
  text(0.5, 0.25, paste("r=",txt))
  text(.5, .75, Signif)
}
# Apply smooth regression line
panel.smooth<-function (x, y, col = "black", bg = NA, pch = 18,
                        cex = 0.8, col.smooth = "red", span = 2/3, iter = 3, ...)
{
  points(x, y, pch = pch, col = col, bg = bg, cex = cex)
  ok <- is.finite(x) & is.finite(y)
  if (any(ok))
    lines(stats:::lowess(x[ok], y[ok], f = span, iter = iter),
          col = col.smooth, ...)
}
# Add histogram to the diagonal
panel.hist <- function(x, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
```

```

y <- h$counts; y <- y/max(y)
rect(breaks[-nB], 0, breaks[-1], y, col="white", ...)
}

```

### 3. Simulate assemblages and calculate FD metrics:

#### 3.1 *Experiment 1*: Simulate trait values following a standardized normal distribution

```

set.seed(403) # Fix the seed to be able to reproduce the experiment

res1 <- simul.dbFD(s = c(5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100),
                     t = 5, r = 1000, p = 500, tr.method ="norm", w.abun = F)

results1 <- data.frame(res1$results)
colnames(results1)[1] <- "Richness"

# Get standard deviation and range of simulates traits in assemblages:

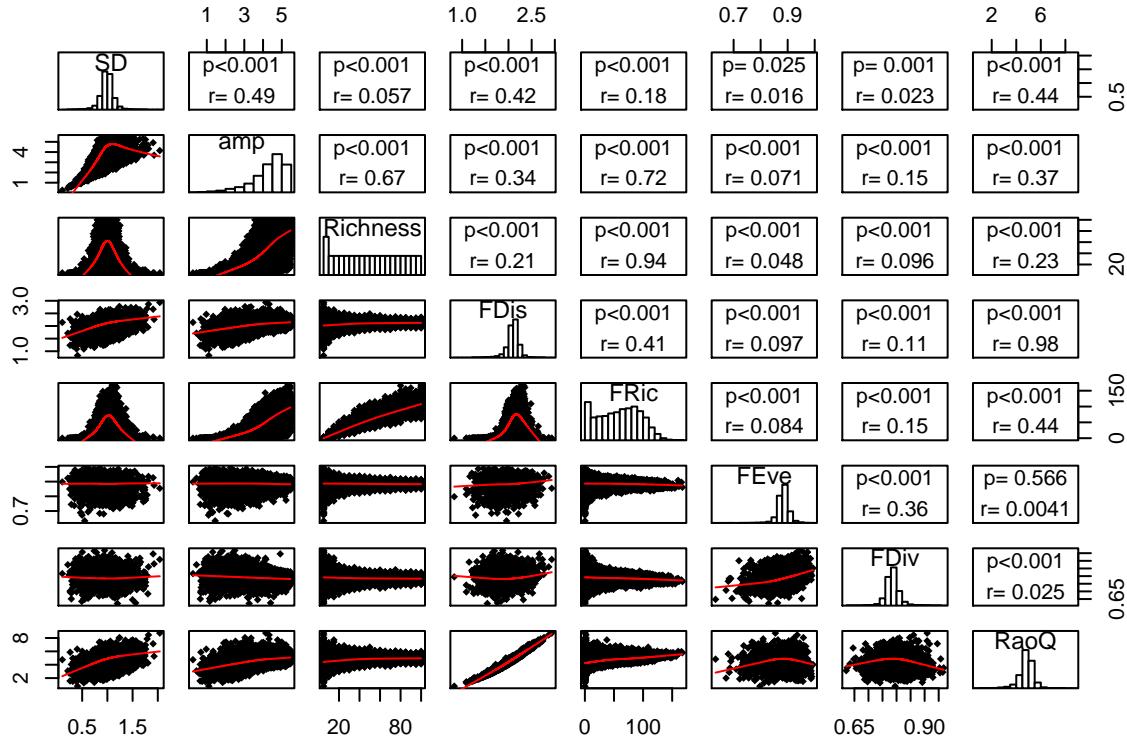
res1$abun[res1$abun > 0] <- 1

SD <- NA
amp <- NA
for(i in 1:nrow(res1$abun)){
  tmp <- res1$traits[which(res1$abun[i,]==1)]
  SD[i] <- sd(tmp)
  amp[i] <- max(tmp)-min(tmp)
}

experiment1 <- cbind(SD, amp, results1)

pairs(experiment1,
      lower.panel=panel.smooth, upper.panel=panel.cor,diag.panel=panel.hist)

```



## 2.2 Experiment 2: Simulate trait values following a normal distribution

In the following experiments we simulated trait values using parameters extracted from the distribution of body mass values observed across mammals. We used a modified version of the simul.dbFD function from FD package, which allows the choice of parameters to generate trait values. simul.dbFD.mod can be found at [github.com/oliveirab/simulation\\_traits](https://github.com/oliveirab/simulation_traits)

```
set.seed(403) # Fix the seed to be able to reproduce the experiment

res2 <- simul.dbFD.mod(s = c(5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100),
                        t = 5, r = 1000, p = 500, tr.method ="norm", tr.par = c(5.05, 2.47),
                        w.abun = F)

results2 <- data.frame(res2$results)
colnames(results2)[1] <- "Richness"

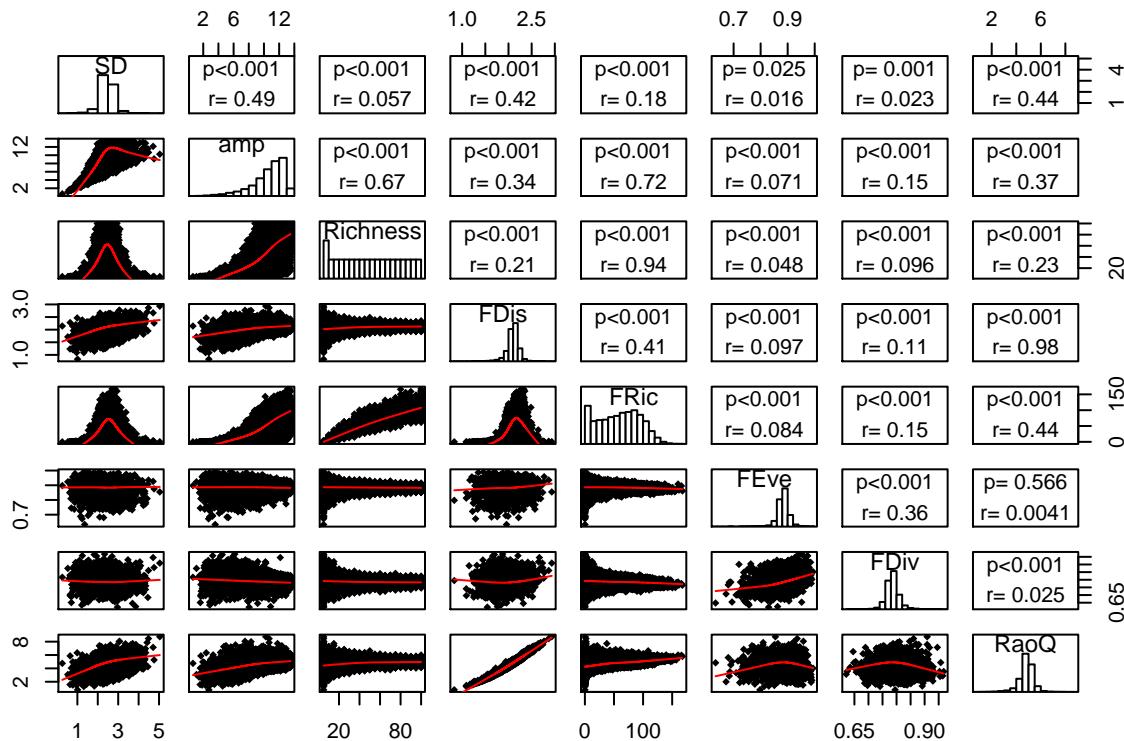
# Get standard deviation and range of simulates traits in assemblages:

res2$abun[res2$abun > 0] <- 1

SD <- NA
amp <- NA
for(i in 1:nrow(res2$abun)){
  tmp <- res2$traits[which(res2$abun[i,]==1)]
  SD[i] <- sd(tmp)
  amp[i] <- max(tmp)-min(tmp)
}

experiment2 <- cbind(SD, amp, results2)
```

```
pairs(experiment2,
      lower.panel=panel.smooth, upper.panel=panel.cor, diag.panel=panel.hist)
```



### 2.3 Experiment 3: Simulate trait values following a log normal distribution

```
set.seed(403) # Fix the seed to be able to reproduce the experiment

res3 <- simul.dbFD.mod(s = c(5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100),
                        t = 5, r = 1000, p = 500, tr.method ="lnorm", tr.par = c(5.05, 2.47),
                        w.abun = F)

results3 <- data.frame(res3$results)
colnames(results3)[1] <- "Richness"

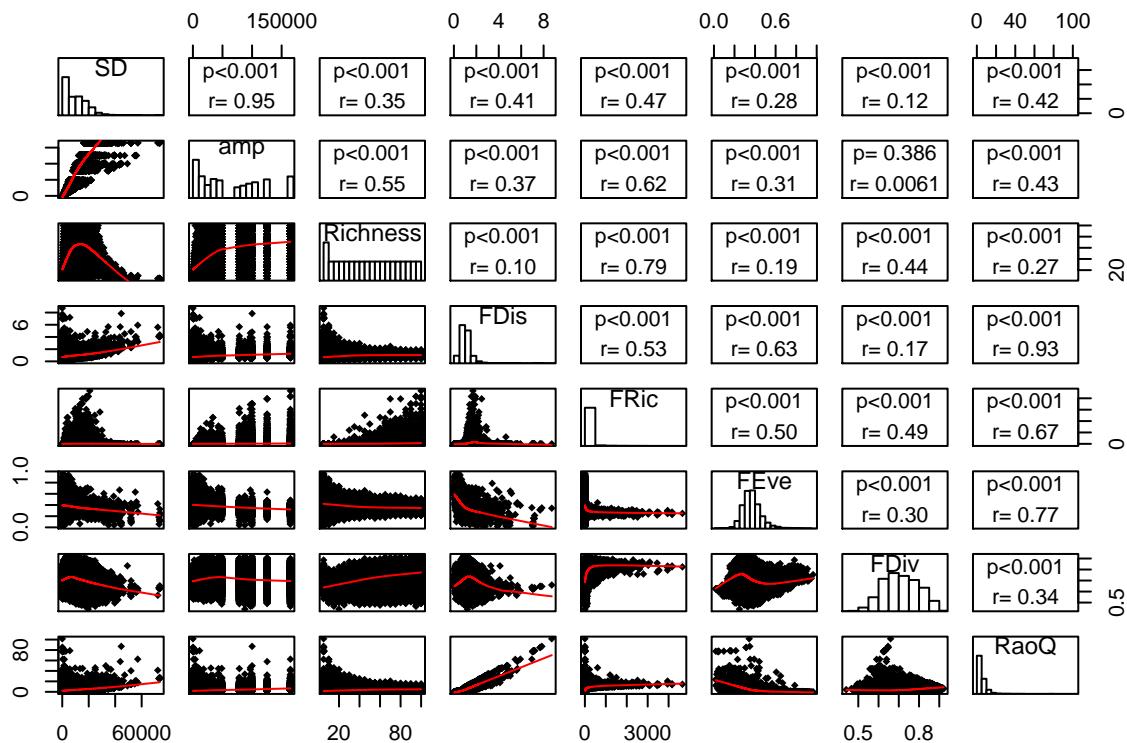
# Get standard deviation and range of simulates traits in assemblages:

res3$abun[res3$abun > 0] <- 1

SD <- NA
amp <- NA
for(i in 1:nrow(res3$abun)){
  tmp <- res3$traits[which(res3$abun[i,]==1)]
  SD[i] <- sd(tmp)
  amp[i] <- max(tmp)-min(tmp)
}

experiment3 <- cbind(SD, amp, results3)
```

```
pairs(experiment3,
      lower.panel=panel.smooth, upper.panel=panel.cor, diag.panel=panel.hist)
```



## 2.4 Experiment 4: Simulate trait values following a exponential distribution

```
set.seed(403) # Fix the seed to be able to reproduce the experiment

res4 <- simul.dbFD.mod(s = c(5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100),
                        t = 5, r = 1000, p = 500, tr.method ="exp", tr.par = 2,
                        w.abun = F)

results4 <- data.frame(res4$results)
colnames(results4)[1] <- "Richness"

# Get standard deviation and range of simulates traits in assemblages:

res4$abun[res4$abun > 0] <- 1

SD <- NA
amp <- NA
for(i in 1:nrow(res4$abun)){
  tmp <- res4$traits[which(res4$abun[i,]==1)]
  SD[i] <- sd(tmp)
  amp[i] <- max(tmp)-min(tmp)
}

experiment4 <- cbind(SD, amp, results4)
```

```

pairs(experiment4,
      lower.panel=panel.smooth, upper.panel=panel.cor, diag.panel=panel.hist)

```

