

RaCoCOS: Range-constrained co-occurrence simulation

Peter Solymos and Zoltan Feher

2017-12-21

Introduction

This is supplementary information for the manuscript *Range-constrained co-occurrence simulation reveals little niche partitioning among rock-dwelling Montenegrina land snails (Gastropoda: Clausiliidae)* by Z. Feher, K. Jaksch, M. Szekeres, E. Haring, S. Bamberger, B. Pall-Gergely, and P. Solymos (*Journal of Biogeography*, in press). The document is a tutorial and reproducible example of the *Range-constrained co-occurrence simulation (RaCoCOS)* methodology outlined in the manuscript.

Required packages

The following packages are required to run the code:

```
if (!require(mefa))
  install.packages("mefa")
library(mefa)
if (!require(pbapply))
  install.packages("pbapply")
library(pbapply)
if (!require(vegan))
  install.packages("vegan")
library(vegan)
if (!require(RColorBrewer))
  install.packages("RColorBrewer")
library(RColorBrewer)
if (!require(rworldmap))
  install.packages("rworldmap")
library(rworldmap)
if (!require(sp))
  install.packages("sp")
library(sp)
```

Distribution data

Use the `setwd` to set working directory where the `Supplementary_data_1.csv` file can be found.

```
d <- read.csv("Supplementary_data_1.csv")
## sample identifiers and coordinates
x <- d[,c("ID", "code", "habitat_type", "Latitude", "Longitude")]
rownames(x) <- d$ID
summary(x)
```

##	ID	code	habitat_type	Latitude
## Min.	: 3 0	: 172	Min. :1.000	Min. :36.45
## 1st Qu.:	759	1972/001:	1 1st Qu.:1.000	1st Qu.:40.69
## Median	:1428	1972/002:	1 Median :1.000	Median :41.89
## Mean	:1394	1972/004:	1 Mean :1.146	Mean :41.77

```
## 3rd Qu.:2004    1972/005:    1    3rd Qu.:1.000    3rd Qu.:42.91
## Max.      :2637    1972/010:    1    Max.      :3.000    Max.      :46.56
##              (Other) :1472
##      Longitude
## Min.      :13.48
## 1st Qu.:19.17
## Median :20.16
## Mean     :20.14
## 3rd Qu.:20.98
## Max.     :27.62
##
## detections
y <- d[,!(colnames(d) %in% colnames(x))]
rownames(y) <- d$ID
y <- as.matrix(y)
y[y > 0] <- 1
str(y)

## num [1:1649, 1:47] 1 1 1 1 0 1 0 0 1 0 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:1649] "3" "16" "19" "20" ...
## ..$ : chr [1:47] "prosobranch" "Carychiidae" "Ferussaciidae" "Azeciidae" ...
```

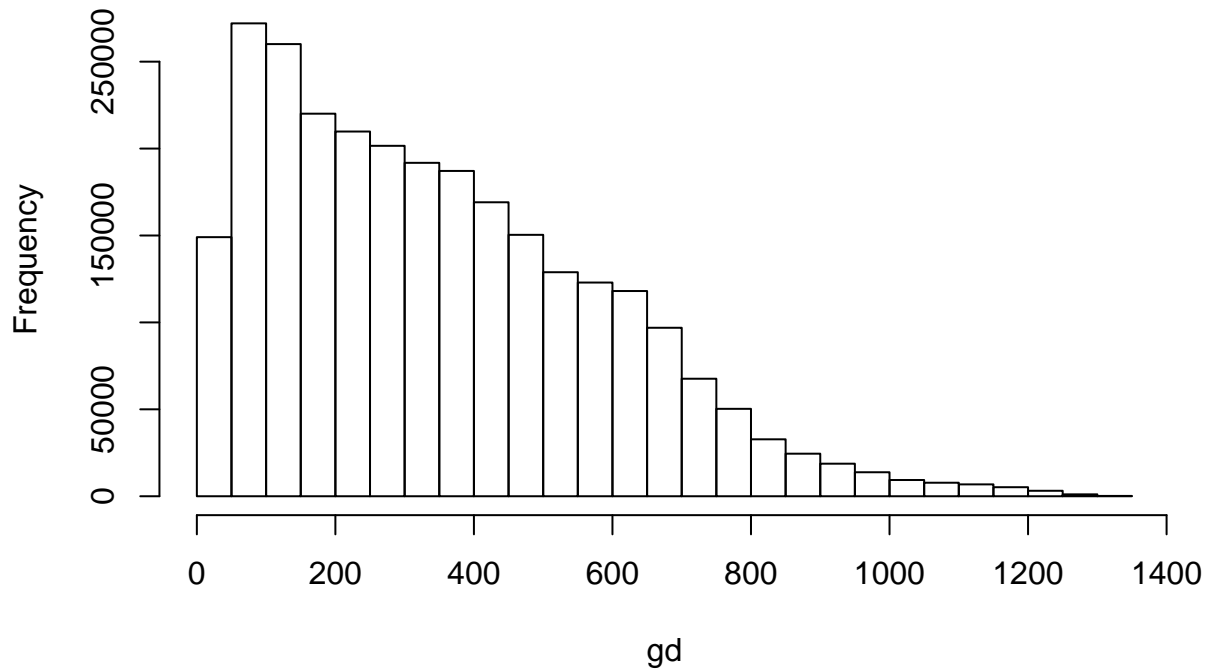
Definition of taxon ranges

Calculate great circle distance from latitude/longitude in km's:

```
xy <- as.matrix(x[,c("Longitude", "Latitude")])
gd <- spDists(x=xy, longlat=TRUE)
str(gd)

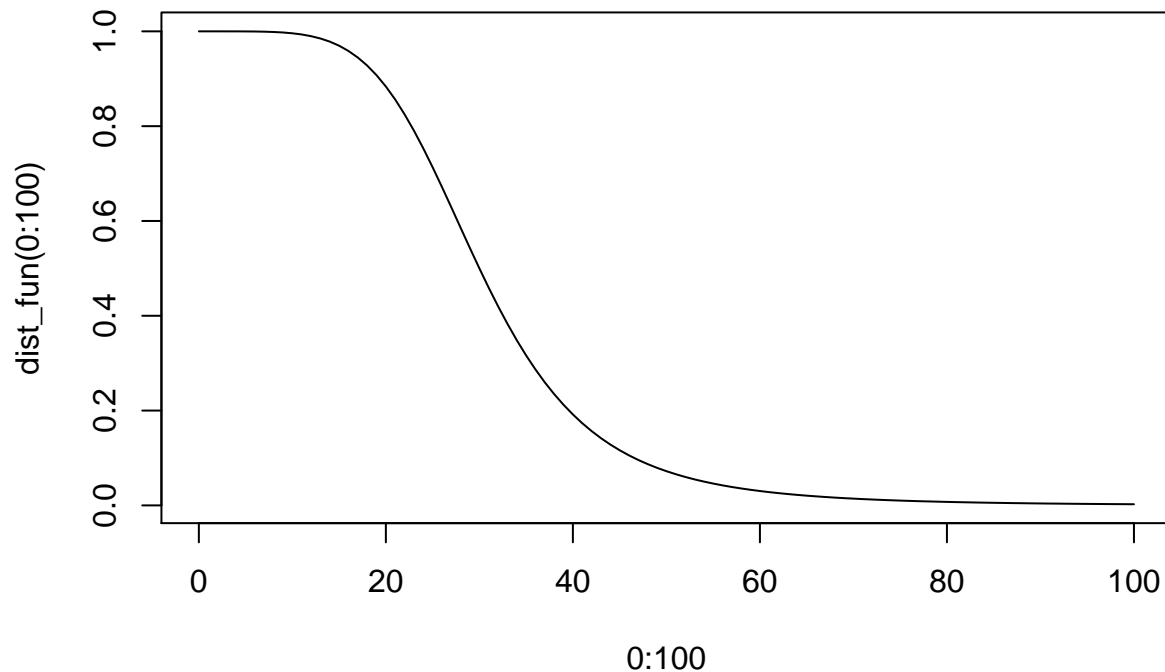
## num [1:1649, 1:1649] 0 184.52 142.53 20.09 1.13 ...
hist(gd)
```

Histogram of gd



The function `dist_fun` calculates the distance weights based on the distances (`d`) and the `k` and `d0` arguments:

```
dist_fun <- function(d, k=5, d0=30) {  
  1 / (1 + exp(k * (log(d) - log(d0))))  
}  
plot(0:100, dist_fun(0:100), type="l")
```



```
w <- dist_fun(gd, k=5, d0=30)
str(w)
```

```
## num [1:1649, 1:1649] 1 0.000114 0.000413 0.881231 1 ...
```

Next we calculate the OP matrix. The `fun_ij` function calculates the OP for site *i* and taxon *j*, the *y* argument is the site x taxa table (the function can be used for simulated matrices as well). The `fun_j` function calculates the OP for species *j* at all sites. The `OP_fun` calculates all the OP values based on an input matrix *y* given the *w* weight matrix based on the distances and the distance decay function:

```
fun_ij <- function(i, j, y) {
  sum(w[which(y[,j] > 0), i]) / sum(w[, i])
}
fun_j <- function(j, y) {
  sapply(1:nrow(y), fun_ij, j=j, y=y)
}
OP_fun <- function(y) {
  pbsapply(1:ncol(y), fun_j, y=y)
}
OP <- OP_fun(y)
dimnames(OP) <- dimnames(y)
str(OP)
```

```
## num [1:1649, 1:47] 0.674 0.889 0.765 0.676 0.678 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:1649] "3" "16" "19" "20" ...
## ..$ : chr [1:47] "prosobranch" "Carychiidae" "Ferussaciidae" "Azeciidae" ...
```

The following function puts the detections or distance weighted probabilities from the OP matrix onto the map:

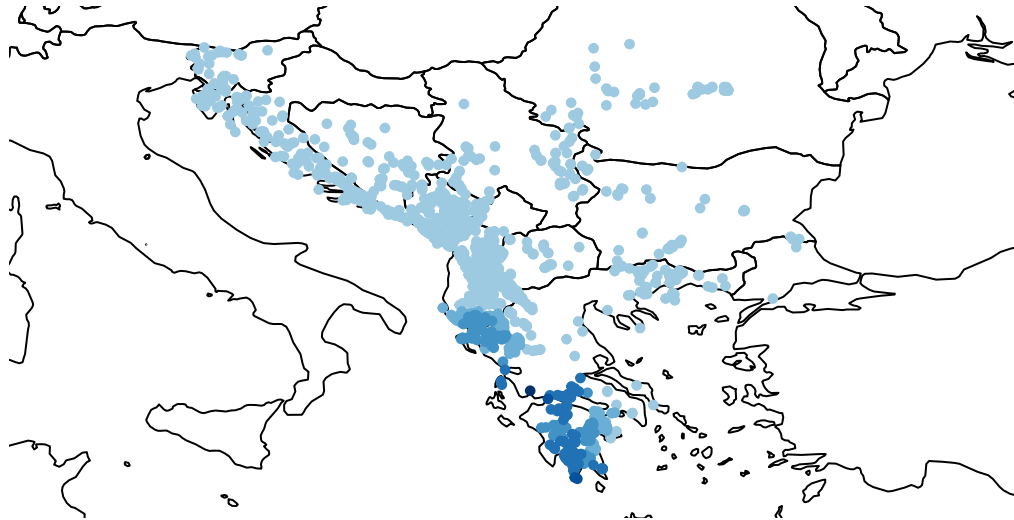
```
plot_distr <-
function(j, type = c("01", "p"), ...)
{
  type <- match.arg(type)
  plot(getMap(resolution = "low"),
        xlim = c(14, 28), ylim = c(36, 47), asp = 1, ...)
  Col <- brewer.pal(9, "Blues")[4:9]
  if (type == "p") {
    z <- cut(OP[,j], breaks=length(Col))
    points(xy[order(z),], pch=19, col = Col[z[order(z)]], cex = .6)
  } else {
    points(xy[y[,j]==0,], pch=".", col=Col[1], cex=1)
    points(xy[y[,j]==1,], pch=19, col=rev(Col)[1], cex=0.6)
  }
  invisible(NULL)
}
plot_distr(32, "01", main="detections")
```

detections



```
plot_distr(32, "p", main="distance weighted probabilities")
```

distance weighted probabilities



Uncorrected, soft, and hard correction:

```
rnd_fun <- function(y, OP, type="uncorrected") {
  type <- match.arg(type, c("uncorrected", "hard", "soft"))
  yout <- y
  yout[] <- 0
  if (type=="uncorrected") {
    n <- nrow(y)
    for (i in 1:ncol(yout)) {
      yout[sample.int(n, sum(y[,i] > 0), replace=FALSE,
        prob=OP[,i]),i] <- 1
    }
  } else {
    wfun_hard <- function(y) {
      rsum <- rowSums(y)
      sample(rsum)
    }
    wfun_soft <- function(y) {
      rsum <- rowSums(y)+1
      sample(rsum)
    }
    if (type == "hard")
      w <- wfun_hard(y)
    if (type == "soft")
      w <- wfun_soft(y)
    wOP <- w*OP
    yout[sample.int(prod(dim(y)), sum(y > 0),
```

```

        prob=wOP)] <- 1
    }
    yout
}
str(rnd_fun(y, OP, "uncorrected"))

## num [1:1649, 1:47] 0 1 1 0 1 0 0 0 1 1 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:1649] "3" "16" "19" "20" ...
## ..$ : chr [1:47] "prosobranch" "Carychiidae" "Ferussaciidae" "Azeciidae" ...
str(rnd_fun(y, OP, "hard"))

## num [1:1649, 1:47] 0 1 1 0 0 0 0 0 0 0 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:1649] "3" "16" "19" "20" ...
## ..$ : chr [1:47] "prosobranch" "Carychiidae" "Ferussaciidae" "Azeciidae" ...
str(rnd_fun(y, OP, "soft"))

## num [1:1649, 1:47] 1 1 1 1 1 0 1 0 0 0 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:1649] "3" "16" "19" "20" ...
## ..$ : chr [1:47] "prosobranch" "Carychiidae" "Ferussaciidae" "Azeciidae" ...

```

Co-occurrence simulations

Now we can calculate co-occurrences. The `coocc_fun` calculates joint occurrences for species pairs (lower triangle of taxa x taxa matrix).

```

coocc_fun <- function(y) {
  co <- t(y) %*% y
  as.dist(co)
}

```

We set number of simulations, and set correction type (change these settings as desired):

```

B <- 99
TYPE <- "uncorrected"

```

We use the `summloden` infrastructure in the **vegan** package:

```

f <- function(x, n, ...)
  array(replicate(n, rnd_fun(x, OP, type=TYPE)), c(dim(x), n))
(cs <- commsim("dist_prob", fun=f, binary=TRUE,
  isSeq=FALSE, mode="integer"))

```

```

## An object of class "commsim"
## 'dist_prob' method (binary, non-sequential, integer mode)

```

```

(nm <- nullmodel(y, cs))

```

```

## An object of class "nullmodel"
## 'dist_prob' method (binary, non-sequential)
## 1649 x 47 matrix

```

```

os <- oecosimu(nm, coocc_fun, nsimul=B)
str(os)

```

```
## List of 2
## $ statistic:Class 'dist' atomic [1:1081] 4 17 33 9 293 6 1 155 163 6 ...
## ..- attr(*, "Labels")= chr [1:47] "prosobranch" "Carychiidae" "Ferussaciidae" "Azeciidae" ...
## ..- attr(*, "Size")= int 47
## ..- attr(*, "call")= language as.dist.default(m = co)
## ..- attr(*, "Diag")= logi FALSE
## ..- attr(*, "Upper")= logi FALSE
## $ oecosimu :List of 8
## ..$ z :Class 'dist' atomic [1:1081] 0.798 2.463 3.356 1.472 4.782 ...
## ..- attr(*, "Labels")= chr [1:47] "prosobranch" "Carychiidae" "Ferussaciidae" "Azeciidae" ...
## ..- attr(*, "Size")= int 47
## ..- attr(*, "call")= language as.dist.default(m = co)
## ..- attr(*, "Diag")= logi FALSE
## ..- attr(*, "Upper")= logi FALSE
## ..$ means : num [1:1081] 3 11.55 22.52 6.46 247.83 ...
## ..$ pval : num [1:1081] 0.69 0.01 0.01 0.23 0.01 0.11 1 0.01 0.01 0.23 ...
## ..$ simulated : num [1:1081, 1:99] 2 9 21 5 245 2 1 118 106 4 ...
## ..$ method : chr "dist_prob"
## ..$ statistic :Class 'dist' atomic [1:1081] 4 17 33 9 293 6 1 155 163 6 ...
## ..- attr(*, "Labels")= chr [1:47] "prosobranch" "Carychiidae" "Ferussaciidae" "Azeciidae" ...
## ..- attr(*, "Size")= int 47
## ..- attr(*, "call")= language as.dist.default(m = co)
## ..- attr(*, "Diag")= logi FALSE
## ..- attr(*, "Upper")= logi FALSE
## ..$ alternative: chr "two.sided"
## ..$ isSeq : logi FALSE
## - attr(*, "call")= language oecosimu(comm = nm, nestfun = coocc_fun, nsimul = B)
## - attr(*, "class")= chr "oecosimu"
```

Now we list species co-occurrences:

```
str(coocc <- coocc_fun(y))

## Class 'dist' atomic [1:1081] 4 17 33 9 293 6 1 155 163 6 ...
## ..- attr(*, "Labels")= chr [1:47] "prosobranch" "Carychiidae" "Ferussaciidae" "Azeciidae" ...
## ..- attr(*, "Size")= int 47
## ..- attr(*, "call")= language as.dist.default(m = co)
## ..- attr(*, "Diag")= logi FALSE
## ..- attr(*, "Upper")= logi FALSE

stack_dist <-
  function (x, dim.names = FALSE, ...)
  {
    id <- as.matrix(x)
    id[lower.tri(id)] <- 1
    id[upper.tri(id)] <- 0
    diag(id) <- 0
    rm <- row(id)
    cm <- col(id)
    rm <- array(rm)[array(id) == 1]
    cm <- array(cm)[array(id) == 1]
    d <- as.vector(x)
    attr(d, "call") <- attr(x, "call")
    attr(d, "method") <- attr(x, "method")
    out <- data.frame(row=rm, col=cm, dist=d)
    if (dim.names) {
```



```

    out$row <- as.factor(out$row)
    out$col <- as.factor(out$col)
    levels(out$row) <- rownames(id)[-1]
    levels(out$col) <- colnames(id)[-ncol(id)]
  }
  out
}
sc <- stack_dist(coocc, dim.names = TRUE)
sc <- sc[order(sc$dist, decreasing=TRUE),]
head(sc)

```

```

##           row           col dist
## 18 Chondrinidae prosobran  430
## 12   Helicidae prosobran  419
## 492 Chondrinidae   Helicidae 352
## 14  Hygromiidae prosobran  326
## 488 Hygromiidae   Helicidae 319
## 5    Enidae prosobran  293

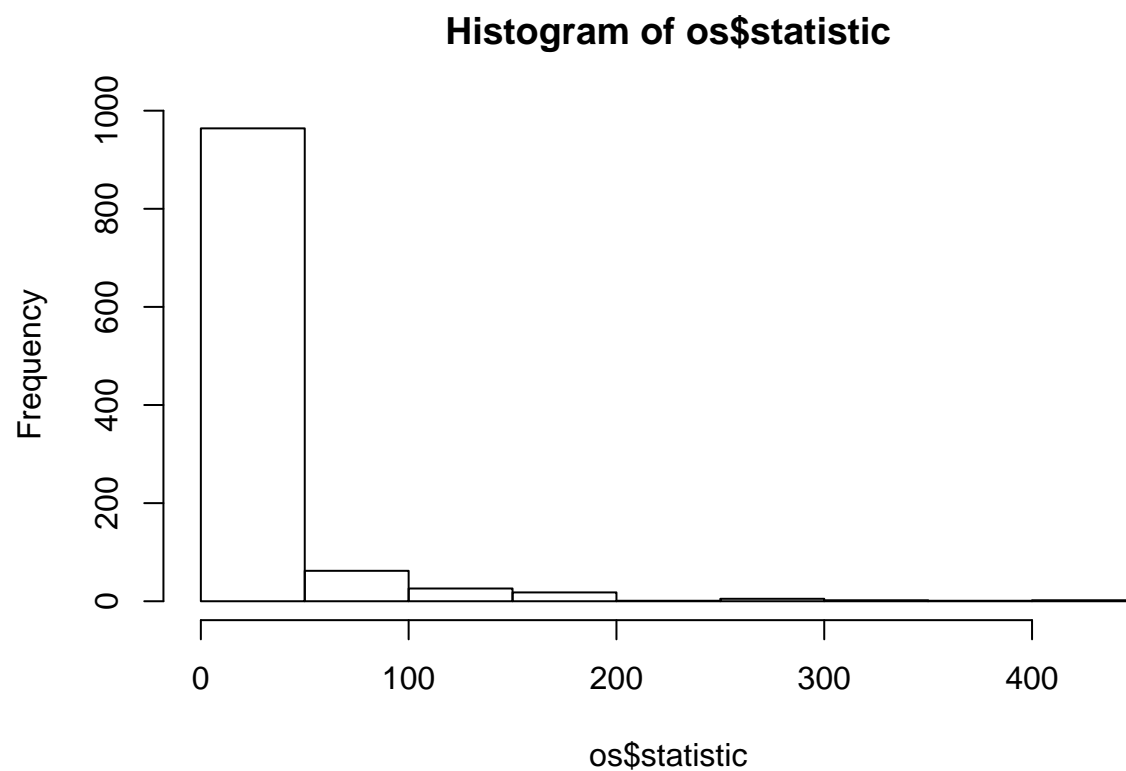
```

The `d2tab` function makes a sorted table of the distances. Let's inspect the `nullmodel` object and plot the results:

```

d2tab <- function(d) {
  if (!inherits(d, "dist"))
    d <- vec2dist(d, attr(coocc, "Size"))
  tab <- stack(d, dim.names=TRUE)
  tab[order(tab$dist, decreasing=TRUE),]
}
## observed co-occurrences
hist(os$statistic)

```

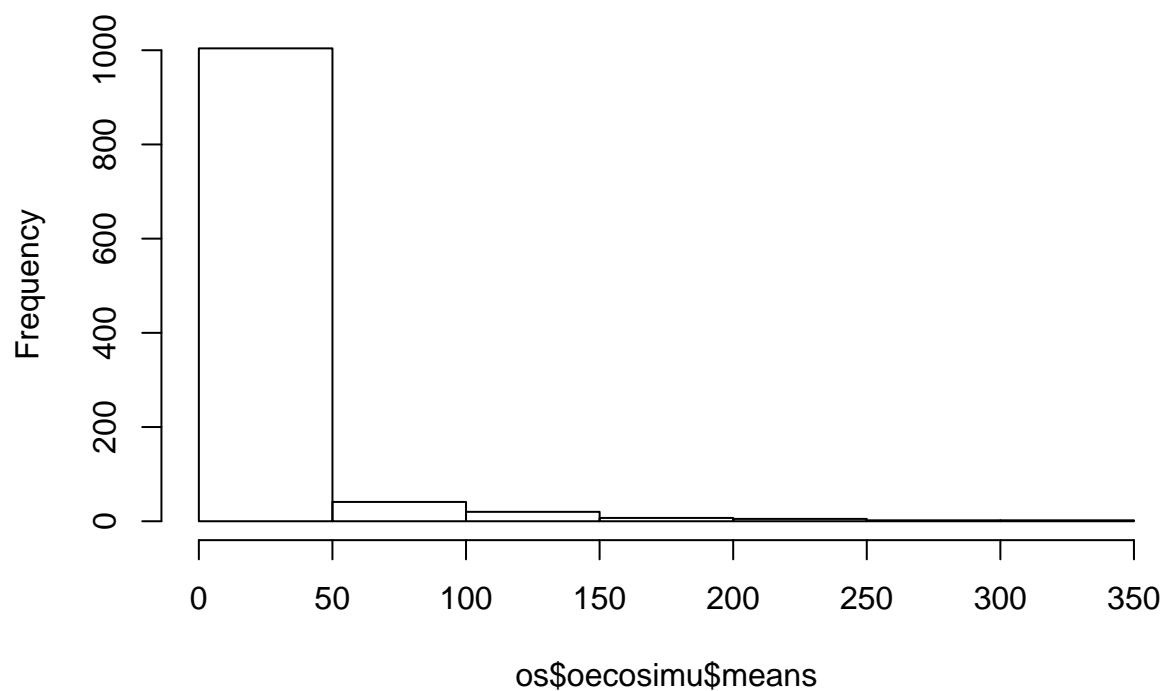


```
head(d2tab(os$statistic))
```

```
##           row      col dist
## 18 Chondrinidae prosobranch 430
## 12  Helicidae prosobranch 419
## 492 Chondrinidae Helicidae 352
## 14 Hygromiidae prosobranch 326
## 488 Hygromiidae Helicidae 319
## 5   Enidae prosobranch 293
```

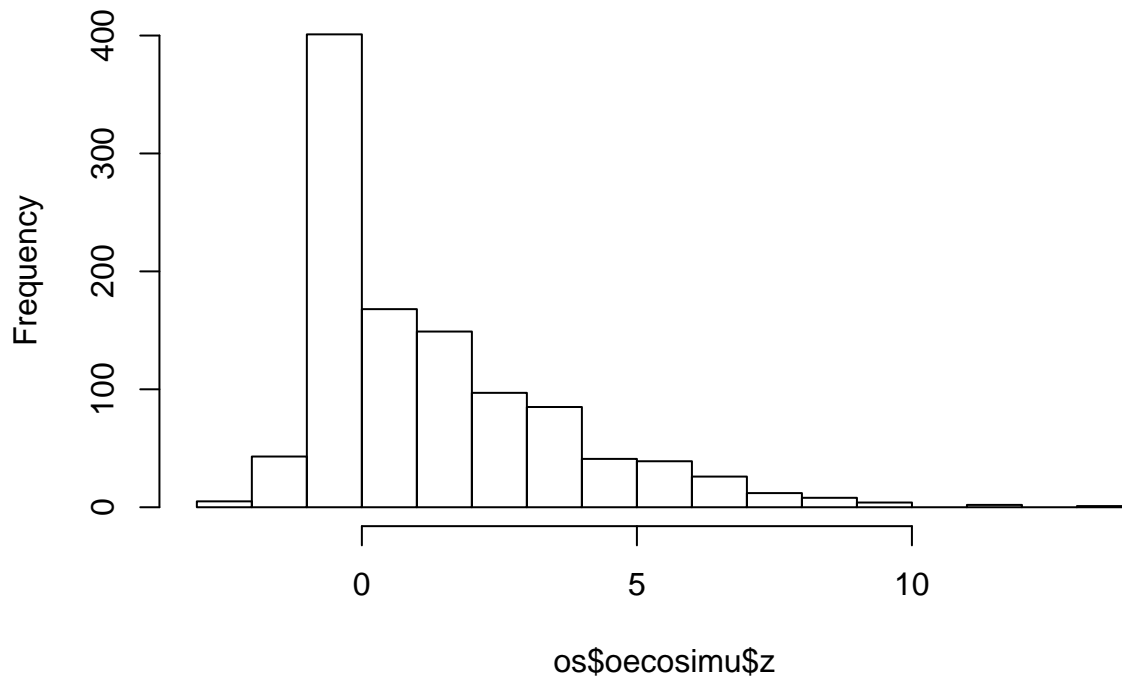
```
## simulated co-occurrences, average
hist(os$oecosimu$means)
```

Histogram of os\$oecosimu\$means



```
## standardized effect size  
hist(os$oecosimu$z)
```

Histogram of os\$oecosimu\$z

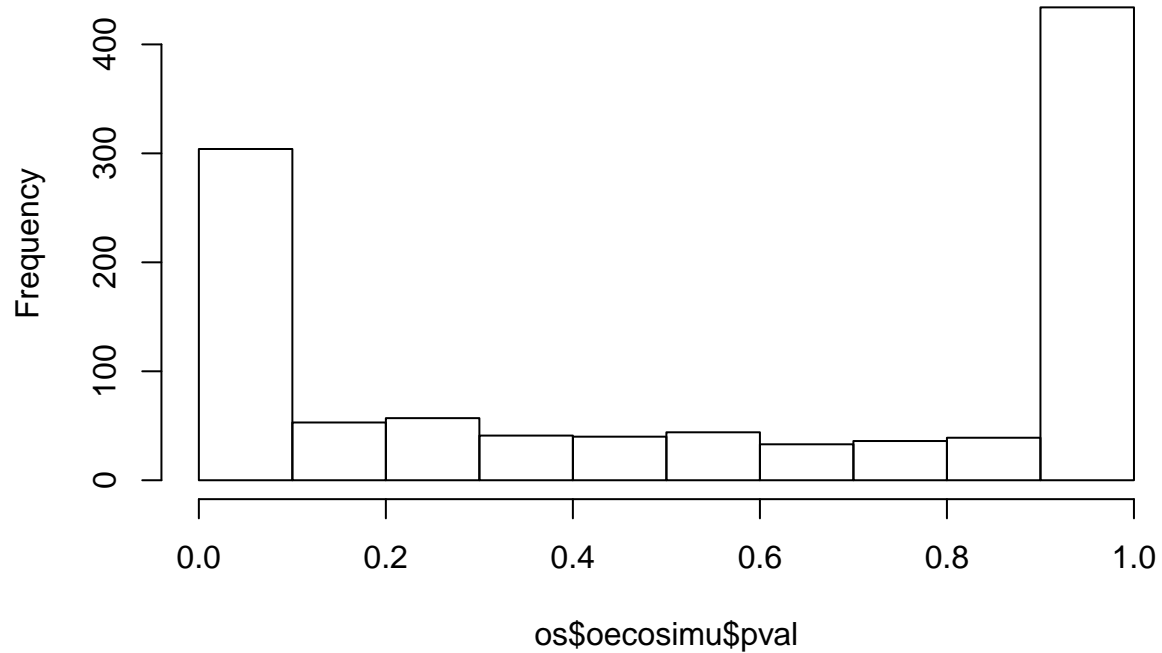


```
head(d2tab(os$oecosimu$z))
```

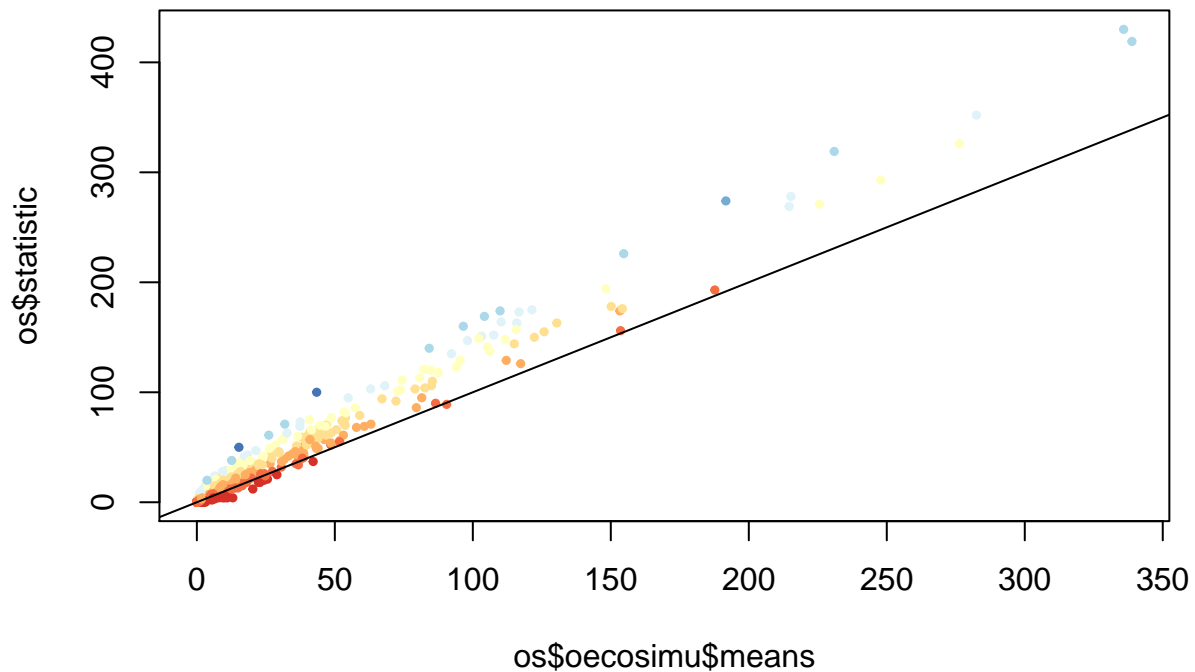
```
##           row           col      dist
## 341 Pristilomatidae Oxychilidae 13.063188
## 394  Vertiginidae Pristilomatidae 11.461709
## 229  Hygromiidae      Enidae 11.129885
## 344  Helicidae      Oxychilidae 9.478166
## 346  Hygromiidae      Oxychilidae 9.260751
## 566  Oleacinidae      Hygromiidae 9.194819
```

```
## two-sided p-values
hist(os$oecosimu$pval)
```

Histogram of os\$oecosimu\$pval



```
col <- brewer.pal(9, "RdYlBu")
plot(os$oecosimu$means, os$statistic,
     col=col[cut(os$oecosimu$z, 9)],
     pch=19, cex=0.5)
abline(0,1,col=1)
```



Session info

```
sessionInfo()
```

```
## R version 3.3.3 (2017-03-06)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 15063)
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] rworldmap_1.3-6   sp_1.2-4      RColorBrewer_1.1-2
## [4] vegan_2.4-3       lattice_0.20-35 permute_0.9-4
## [7] pbapply_1.3-4     mefa_3.2-7
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.13      knitr_1.15.1     cluster_2.0.6    magrittr_1.5
```

```
## [5] maptools_0.9-2  maps_3.2.0      MASS_7.3-45     stringr_1.2.0
## [9] fields_9.0      tools_3.3.3     dotCall64_0.9-04 parallel_3.3.3
## [13] grid_3.3.3      spam_2.1-1      nlme_3.1-131    mgcv_1.8-17
## [17] htmltools_0.3.5 yaml_2.1.14     rprojroot_1.2   digest_0.6.12
## [21] Matrix_1.2-8     evaluate_0.10.1 rmarkdown_1.6   stringi_1.1.5
## [25] backports_1.0.5  foreign_0.8-67
```