

## Appendix 1. R scripts and examples

The R function `distorefpool` calculates, in a set of  $N$  plots, the functional dissimilarity between a given plot and a theoretical plot obtained by pooling all other plots (like in Eq. 4 and 5 of the main text). When the plots are distributed between defined groups, the R function `rtestsampledis` performs the ANOVA-like approach that evaluates how different the average dissimilarity between two plots of a group (see argument `fac`) is from one group to another. The function `posthoc` completes function `rtestsampledis` by performing pair-wise tests: an ANOVA-like test is performed to evaluates how different the average dissimilarity between two plots of a group is from the average dissimilarity between two plots of another group (all possible combinations of two groups are considered). This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License <http://www.gnu.org/licenses/>.

It will be integrated in a new version of the `adiv` package of R: <https://cran.r-project.org/web/packages/adiv/index.html>. The functions were checked and applied with R-4.0.5.

**Disclaimer:** users of this code are cautioned that, while due care has been taken and it is believed accurate, it has not been rigorously tested and its use and results are solely the responsibilities of the user.

**Dependencies:** `ade4` (Thioulouse et al. 2018)

**Function Syntax:** Functions are available in Appendix 2 (.txt file)

```
Usage: distorefpool(comm, dis, fundis, ...)
rtestsampledis(comm, dis, fac, fundis, option = 1:2, nrep = 999,
...)
posthoc(Xtest, p.adjust.method = "none", output = c("light",
"full"))
```

### Arguments:

`comm`: a matrix or a data frame of  $N$  plots  $\times$   $S$  species containing the relative or absolute abundance of all species. Columns are species and plots are rows.

`dis`: a matrix or an object of class `dist` providing the functional dissimilarities between species (dissimilarities are nonnegative, symmetric, and the dissimilarity between a species and itself is zero). ***Species here must be in the same order as in the columns of comm.***

`fundis`: a character string giving the name of a function to calculate plot to plot dissimilarity. The first argument of this function must be a matrix or a data frame of plots  $\times$  species containing the relative or absolute abundance of all species. The second argument of this function must be an object of class `dist` providing the functional dissimilarities between species (dissimilarities are nonnegative, symmetric, and the dissimilarity between a species and itself is zero).

`...`: further arguments that can be passed to function `fundis`.

`fac`: a factor with the length equal to the number of rows in `comm`. The factor indicates which group each plot belongs to.

`option`: either 1 or 2. If 1, then the functional distinctiveness of a plot is measured as its average functional dissimilarity to another plot (like in Eq. 2 of the main text). If 2, then the functional distinctiveness of a plot is measured as its functional dissimilarity to a theoretical plot obtained by pooling all other plots (like in Eq. 5 of the main text).

`nrep`: the number of permutations to be done in the ANOVA-like approach that evaluates how different the average dissimilarity between two plots of a group (see argument `fac`) is from one group to another.

`Xtest`: an object of class "rtestsampledis" obtained by function `rtestsampledis`.

`p.adjust.method`: a character string. See function `p.adjust` for details and choice of adjustment methods.

`output`: a logical. If `TRUE`, all simulated values are saved and returned.

### Value:

The function `distorefpool` returns a vector with, for each studied plot, the average functional dissimilarity between the plot and the other plots (Eq. 2 of the main text).

The function `rtestsampledis` returns an object of class "randtest" (from package `ade4`, Thioulouse et al. 2018).

The function `posthoc` returns an object of class "krandtest" (also from package `ade4`).

### Example:

Install and load packages `lpSolve`, `adiv`, `ade4` and `ggplot2`.

```
install.packages("lpSolve") # Berkelaar et al. (2020)
install.packages("adiv") # Pavoine (2020a,b)
install.packages("ade4") # Thioulouse et al. (2018)
install.packages("ggplot2") # Wickham (2016)
```

```
library(lpSolve)
library(adiv)
library(ade4)
library(ggplot2)
```

Load functions `distorefpool`, `rtestsampledis` and `posthoc` available in Appendix 2 (a .txt file) in R. For that, use the following script to source the Appendix:

```
source(file.choose())
```

Load also function `dislptransport` available in Ricotta et al. (2021, Appendix S3). For that, copy and paste the function directly in your R console or copy and paste each of them in a .txt file and use the script `source(file.choose())` to source it in R

Load the dataset `RutorGlacier` available in the `adiv` package of R.

```
data(RutorGlacier)
```

`RutorGlacier` is a list. `RutorGlacier$Abund` is an object of class `data.frame` that contains the abundance of plant species in plots (plots in rows; species in columns). `RutorGlacier$Traits2` is an object of class `data.frame` that contains the values of traits for all observed plant species.

As in Ricotta et al. (2021), the functional distances between species are calculated using the Euclidean distance applied to species' traits, each standardized to zero mean and unit standard deviation :

```
fdis <- dist(scale(RutorGlacier$Traits2[1:6]))
```

Then the resulting functional distances among species were scaled to the unit range by dividing each distance by the maximum value in the distance matrix :

```
fdis <- fdis/max(fdis)
```

Pairwise functional distances between plots can be obtained by using the following command:

```
Dhk <- dislptransport(RutorGlacier$Abund, fdis) # Eq. 1 of the main text
```

The associated overall beta diversity over all plots simply is

```
BetaN <- mean(Dhk) # Eq. 2 of the main text
BetaN
[1] 0.2116677
```

The average dissimilarity of each individual plot to all other plots can be obtained as follows:

```
Dkbar <- sapply(1:attributes(Dhk)$Size, function(i) mean(as.matrix(Dhk)[i,
-i])) # Eq. 3 of the main text
names(Dkbar) <- attributes(Dhk)$Labels
Dkbar
```

	X1	X2	X3	X4	X5	X6	X7	X8
0.2833631								
	X9							
0.3444822								
	X17							
0.2490277								
	X25							
0.1787682								
	X33							
0.1655225								
	X41							
0.1706312								
	X49							
0.1852885								
	X57							
0.2303822								

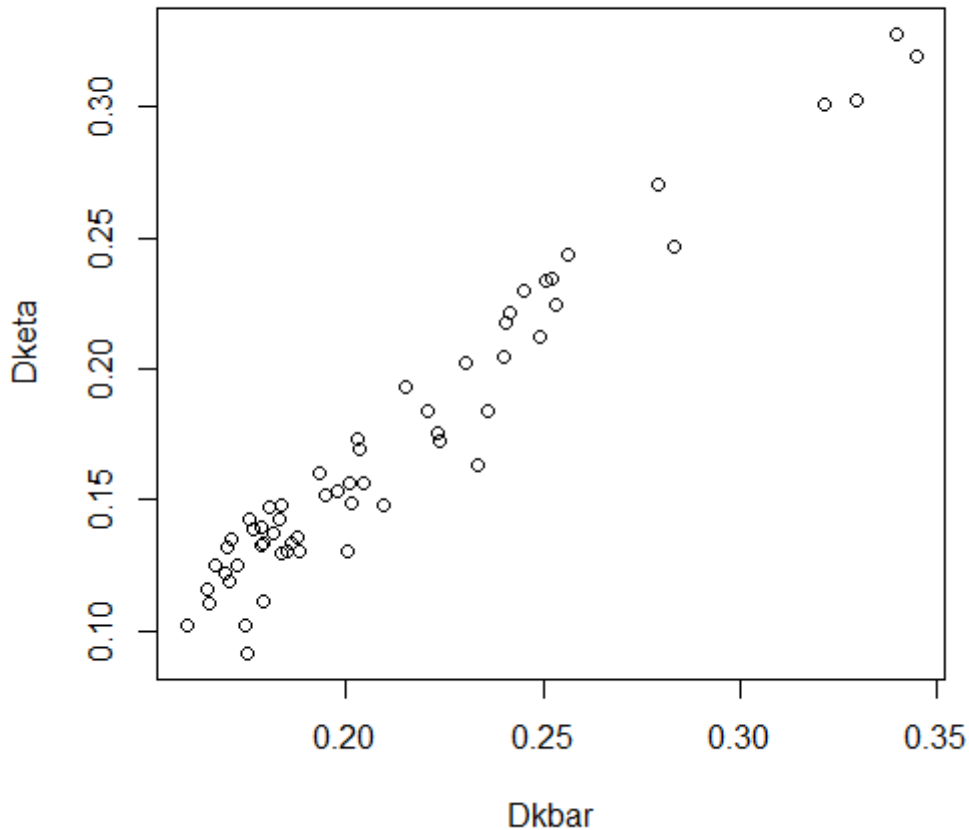
The average dissimilarity of each individual plot to the theoretical plot obtained by pooling all other plots can be obtained as follows:

```
Dketa <- distorefpool(RutorGlacier$Abund, fdis, "dislptransport")
Dketa
```

	X1	X2	X3	X4	X5	X6	X7
0.24676468							
	X8						
0.14835707							
	X15						
0.30088606							
	X22						
0.12467190							
	X29						
0.10180148							
	X36						
0.14724159							
	X43						
0.13484530							
	X50						
0.22158769							
	X57						
0.20210019							

Link between Dketa and Dkbar:

```
plot(Dkbar, Dketa)
```



The overall beta diversity over all plots associated with Dketa simply is

```
BetaEta <- mean(Dketa) # Eq. 5 of the main text
BetaEta
[1] 0.1706723
```

RutorGlacier\$Fac is a vector that indicates which level of the successional gradient a given plot belongs to ("early" = early-successional stage, "mid" = mid-successional stage and "late" = late-successional stage).

```
FAC <- factor(RutorGlacier$Fac, levels = c("early", "mid", "late"))
```

The same calculation of the functional distinctiveness of a plot can thus also be done within each successional stage:

```
Listcomm <- split(1:nrow(RutorGlacier$Abund), FAC)

funx <- function(x) {
  funi <- function(i) mean((as.matrix(Dhk)[x, x])[i, -i])
  RES <- sapply(1:length(x), funi)
  names(RES) <- rownames(RutorGlacier$Abund)[x]
```

```

return(RES)
}

LDkbar <- lapply(Listcomm, funx)
LDkbar
$early
  X1      X2      X3      X4      X5      X6      X7      X8
0.2793514 0.2063564 0.2146637 0.2145762 0.1902042 0.2452806 0.1980554 0.1963205
  X9      X10     X11     X12     X13     X14     X15     X16
0.2663816 0.1839177 0.2005795 0.1834980 0.1820301 0.2263759 0.2358982 0.2760978
  X17
0.1926447

$mid
  X18     X19     X20     X21     X22     X23     X24     X25
0.1421271 0.1504958 0.1306955 0.1609434 0.1237694 0.1233812 0.1472974 0.1387185
  X26     X27     X28     X29     X30     X31     X32     X33
0.1150523 0.1174132 0.1419706 0.1069922 0.1350532 0.1246678 0.1200482 0.1132186
  X34     X35     X36     X37     X38     X39     X40     X41
0.1368429 0.1490411 0.1279568 0.1240154 0.1311320 0.1355208 0.1443436 0.1225057
  X42     X43     X44     X45     X46     X47     X48     X49
0.1563800 0.1197213 0.1292627 0.1328761 0.1536369 0.2102809 0.1542701 0.1452713

$late
  X50     X51     X52     X53     X54     X55     X56     X57
0.1191709 0.1334624 0.1510494 0.1325934 0.1473606 0.1686762 0.1262438 0.1119260
  X58     X59
0.1395953 0.1234304

```

```

DkbarWithinGroup <- unlist(LDkbar, use.names = FALSE)
names(DkbarWithinGroup) <- unlist(lapply(LDkbar, names))
DkbarWithinGroup <- DkbarWithinGroup[rownames(RutorGlacier$Abund)]
DkbarWithinGroup
  X1      X2      X3      X4      X5      X6      X7      X8
0.2793514 0.2063564 0.2146637 0.2145762 0.1902042 0.2452806 0.1980554 0.1963205
  X9      X10     X11     X12     X13     X14     X15     X16
0.2663816 0.1839177 0.2005795 0.1834980 0.1820301 0.2263759 0.2358982 0.2760978
  X17     X18     X19     X20     X21     X22     X23     X24
0.1926447 0.1421271 0.1504958 0.1306955 0.1609434 0.1237694 0.1233812 0.1472974
  X25     X26     X27     X28     X29     X30     X31     X32
0.1387185 0.1150523 0.1174132 0.1419706 0.1069922 0.1350532 0.1246678 0.1200482
  X33     X34     X35     X36     X37     X38     X39     X40
0.1132186 0.1368429 0.1490411 0.1279568 0.1240154 0.1311320 0.1355208 0.1443436
  X41     X42     X43     X44     X45     X46     X47     X48
0.1225057 0.1563800 0.1197213 0.1292627 0.1328761 0.1536369 0.2102809 0.1542701
  X49     X50     X51     X52     X53     X54     X55     X56
0.1452713 0.1191709 0.1334624 0.1510494 0.1325934 0.1473606 0.1686762 0.1262438
  X57     X58     X59
0.1119260 0.1395953 0.1234304

```

```

Listab <- split(RutorGlacier$Abund, FAC)
LDketa <- lapply(Listab, function(x) distorefpool(x, fdis,
"dislptransport"))
LDketa
$early
  X1      X2      X3      X4      X5      X6      X7
0.26392207 0.15402068 0.16524452 0.18909005 0.10639756 0.20661496 0.14685590
  X8      X9      X10     X11     X12     X13     X14
0.14954525 0.23729735 0.11137968 0.15482078 0.11607055 0.09729446 0.20217095
  X15     X16     X17
0.20733192 0.26076225 0.13230378

$mid
  X18     X19     X20     X21     X22     X23     X24
0.11767510 0.12624949 0.11017017 0.13823905 0.09834954 0.09531792 0.12972948
  X25     X26     X27     X28     X29     X30     X31
0.11711503 0.08477043 0.08440223 0.12207607 0.06362562 0.10044876 0.07859484

```

```

      X32      X33      X34      X35      X36      X37      X38
0.08973293 0.07011797 0.10849854 0.11154361 0.10510035 0.09342271 0.10286584
      X39      X40      X41      X42      X43      X44      X45
0.08521716 0.12458756 0.08391134 0.13600940 0.09119526 0.09812298 0.10750062
      X46      X47      X48      X49
0.13347024 0.19162813 0.13794545 0.11570316

```

```
$late
```

```

      X50      X51      X52      X53      X54      X55      X56
0.08910403 0.10888189 0.13266509 0.10950880 0.10701116 0.16268995 0.08632015
      X57      X58      X59
0.06387000 0.12289391 0.09773765

```

```

DketaWithinGroup <- unlist(LDketa, use.names = FALSE)
names(DketaWithinGroup) <- unlist(lapply(LDketa, names))
DketaWithinGroup <- DketaWithinGroup[rownames(RutorGlacier$Abund)]
DketaWithinGroup

```

```

      X1      X2      X3      X4      X5      X6      X7
0.26392207 0.15402068 0.16524452 0.18909005 0.10639756 0.20661496 0.14685590
      X8      X9      X10     X11     X12     X13     X14
0.14954525 0.23729735 0.11137968 0.15482078 0.11607055 0.09729446 0.20217095
      X15     X16     X17     X18     X19     X20     X21
0.20733192 0.26076225 0.13230378 0.11767510 0.12624949 0.11017017 0.13823905
      X22     X23     X24     X25     X26     X27     X28
0.09834954 0.09531792 0.12972948 0.11711503 0.08477043 0.08440223 0.12207607
      X29     X30     X31     X32     X33     X34     X35
0.06362562 0.10044876 0.07859484 0.08973293 0.07011797 0.10849854 0.11154361
      X36     X37     X38     X39     X40     X41     X42
0.10510035 0.09342271 0.10286584 0.08521716 0.12458756 0.08391134 0.13600940
      X43     X44     X45     X46     X47     X48     X49
0.09119526 0.09812298 0.10750062 0.13347024 0.19162813 0.13794545 0.11570316
      X50     X51     X52     X53     X54     X55     X56
0.08910403 0.10888189 0.13266509 0.10950880 0.10701116 0.16268995 0.08632015
      X57     X58     X59
0.06387000 0.12289391 0.09773765

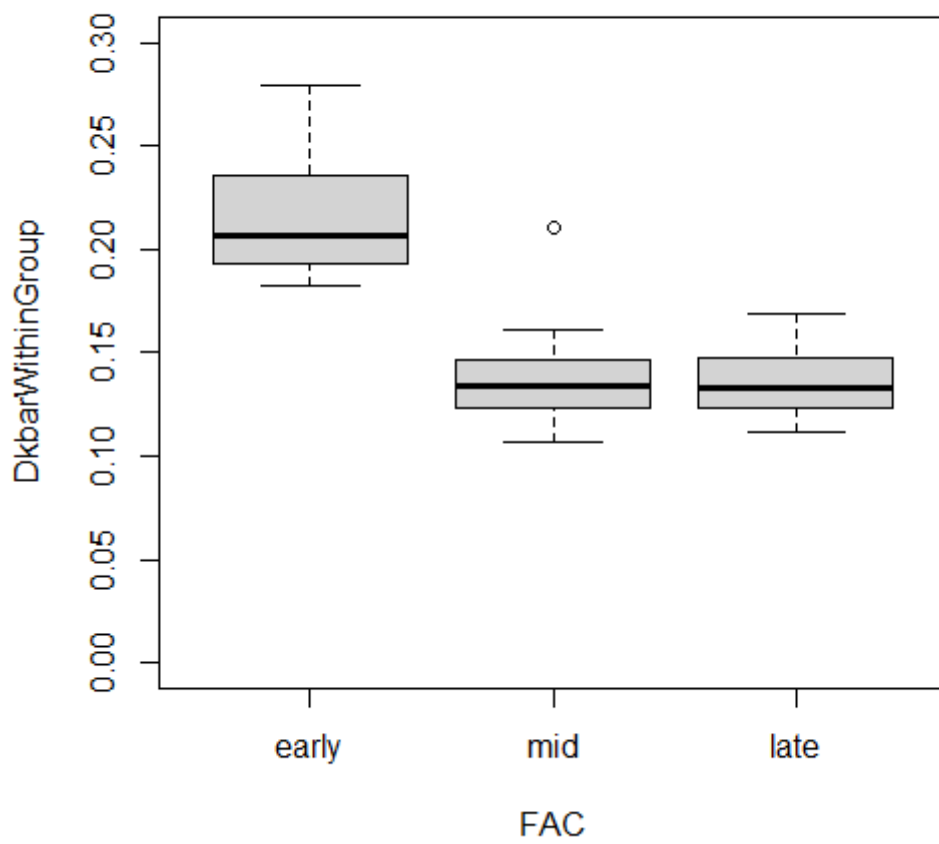
```

```

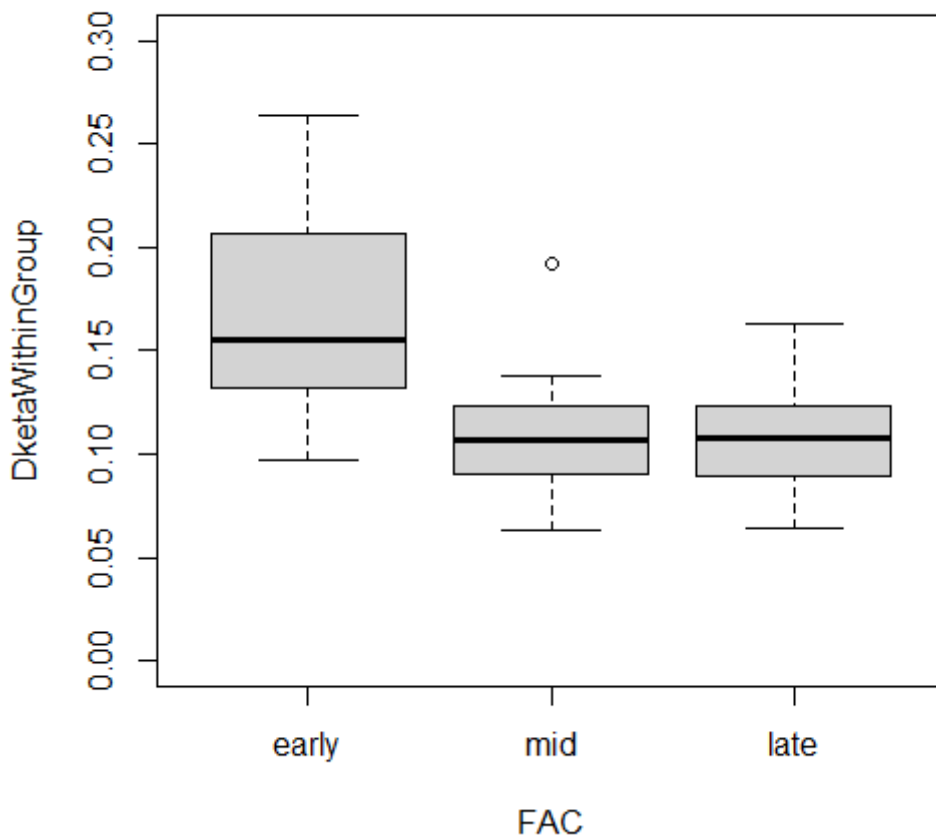
# Box plots of the functional distinctiveness of plots (as defined by  $D_k$ 
and  $D_{k\eta}$ , respectively) within each successional stage:

```

```
boxplot(DkbarWithinGroup~FAC, ylim=c(0,0.3))
```



```
boxplot(DketaWithinGroup ~FAC, ylim=c(0,0.3))
```



ANOVA-like approach to evaluate how different the average dissimilarity between two plots of a group:

```
Rkbar <- rtestsamplendis(RutorGlacier$Abund, fdis, FAC, "dislptransport",
option = 1, nrep = 9999) # Here with index  $D_{\bar{k}}$ 
```

```
Rkbar
```

```
Monte-Carlo test
```

```
Call: rtestsamplendis(comm = RutorGlacier$Abund, dis = fdis, fac = FAC,
fundis = "dislptransport", option = 1, nrep = 9999)
```

```
Observation: 71.56196 # F statistic
```

```
Based on 9999 replicates
```

```
Simulated p-value: 1e-04 # P-value
```

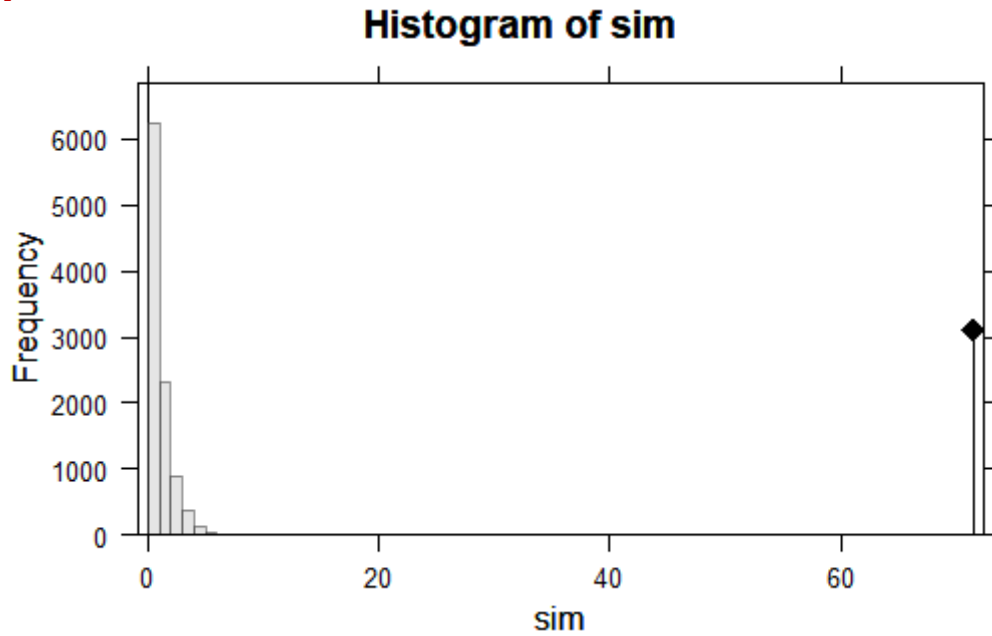
```
Alternative hypothesis: greater
```

Std.Obs	Expectation	Variance
65.043802	1.039703	1.175548

```
# Below the histogram of simulated values with an arrow indicating the
observed value of explained sum of squares divided by residual sum of
squares.
```



```
plot(Rkbar)
```



```
Rketa <- rtestsampledis(RutorGlacier$Abund, fdis, FAC, "dislptransport",  
option = 2, nrep = 9999) # Here with index  $D_{kn}$ 
```

```
Rketa
```

```
Monte-Carlo test
```

```
Call: rtestsampledis(comm = RutorGlacier$Abund, dis = fdis, fac = FAC,  
fundis = "dislptransport", option = 2, nrep = 9999)
```

```
Observation: 18.91216 # F statistic
```

```
Based on 9999 replicates
```

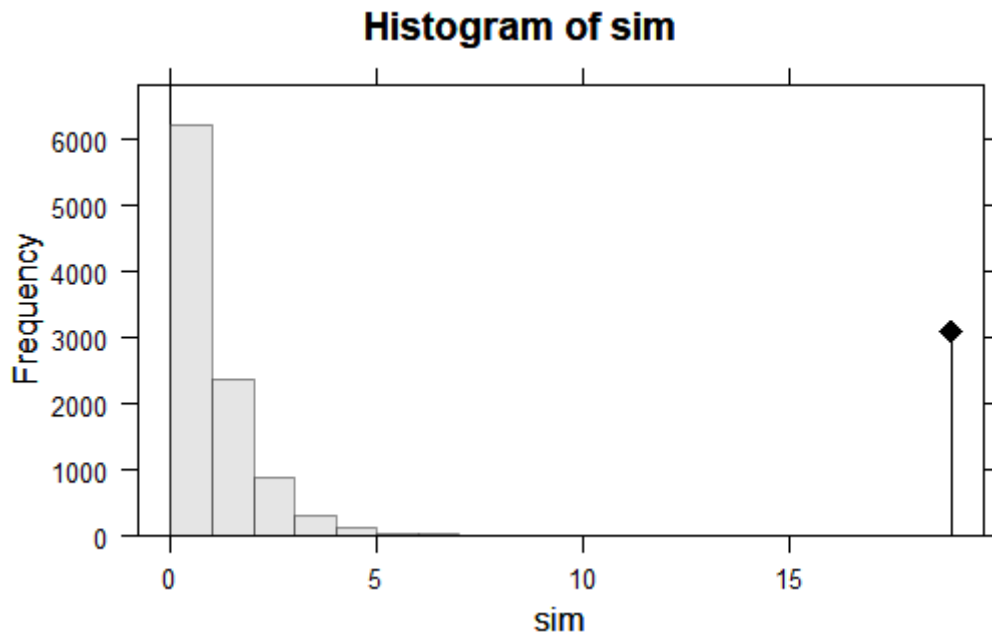
```
Simulated p-value: 1e-04 # P-value
```

```
Alternative hypothesis: greater
```

Std.Obs	Expectation	Variance
16.744722	1.033480	1.140026

```
# Below the histogram of simulated values with an arrow indicating the  
observed value of explained sum of squares divided by residual sum of  
squares.
```

```
plot(Rketa)
```



Below are the pair-wise tests: ANOVA-like tests performed to evaluate how different the average dissimilarity between two plots of a group is from the average dissimilarity between two plots of another group (all possible combinations of two groups are considered)

```
PHkbar <- posthoc(Rkbar, p.adjust.method = "holm")
PHkbar # Here with index  $D_k$ 
class: krandtest lightkrandtest
Monte-Carlo tests
Call: posthoc(Xtest = Rkbar, p.adjust.method = "holm")
```

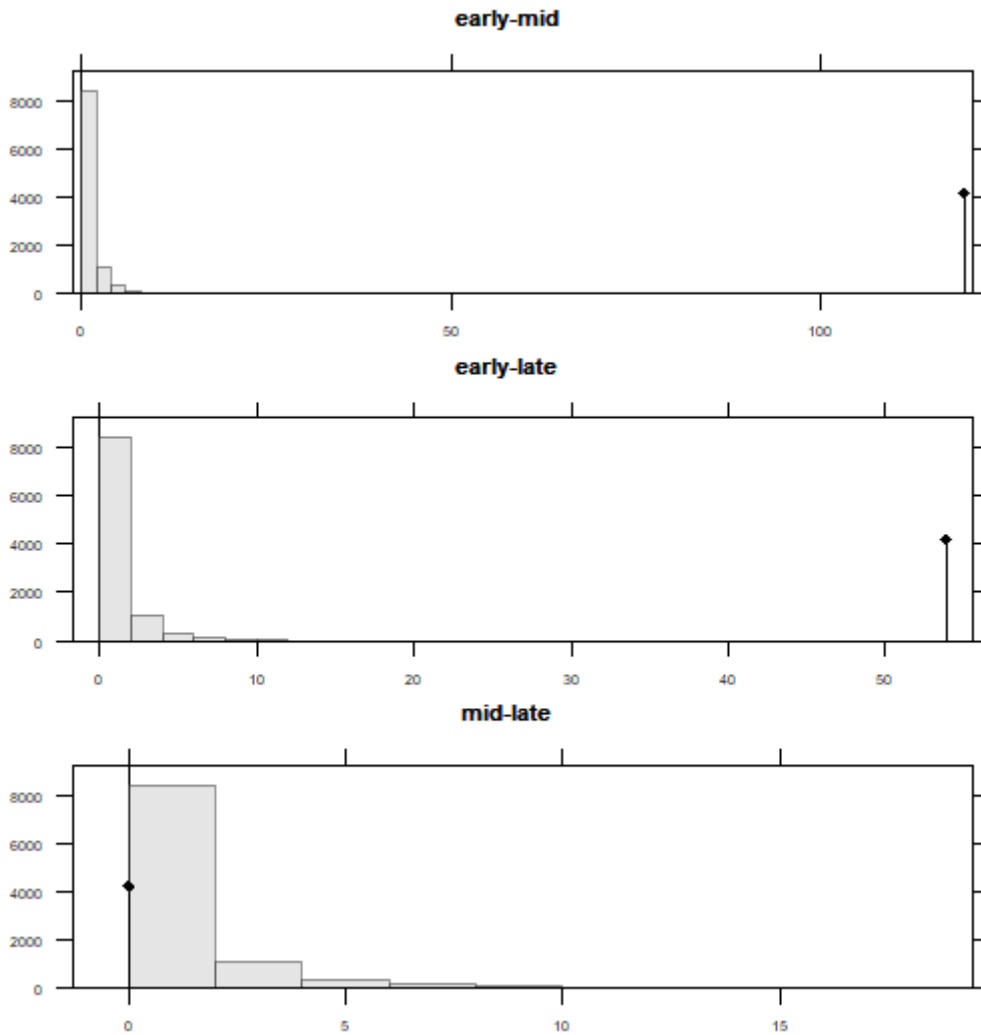
Number of tests: 3

```
Adjustment method for multiple comparisons: holm
Permutation number: 9999
  Test      Obs      Std.Obs  Alter Pvalue Pvalue.adj
1 early-mid 119.34000057 78.1224147 greater 0.0001 0.0003
2 early-late 53.87378756 31.4392769 greater 0.0001 0.0003
3 mid-late 0.02383877 -0.6623418 greater 0.8805 0.8805
```

# Column Test indicates which successional stages are compared; Obs = F statistic; Pvalue.adj = adjusted P-value

# Below, for each pair of compared succession stages, the histogram of simulated values with an arrow indicating the observed F value.

```
plot(PHkbar)
```



```
PHketa <- posthoc(Rketa, p.adjust.method = "holm")
PHketa # Here with index  $D_{k\eta}$ 
class: krantest lightkrantest
Monte-Carlo tests
Call: posthoc(Xtest = Rketa, p.adjust.method = "holm")
```

Number of tests: 3

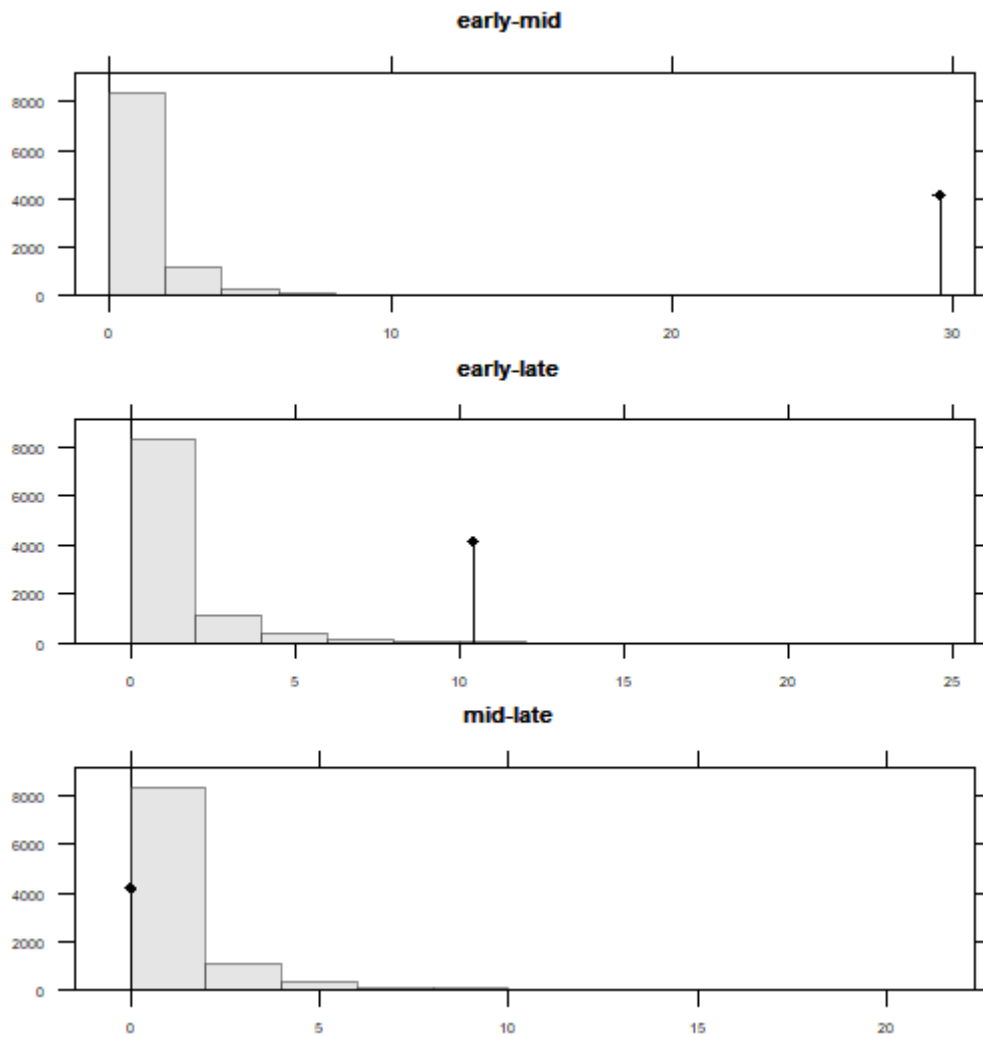
Adjustment method for multiple comparisons: holm  
 Permutation number: 9999

	Test	Obs	Std.Obs	Alter	Pvalue	Pvalue.adj
1	early-mid	29.54092333	19.531254	greater	0.0001	0.0003
2	early-late	10.45608502	5.507140	greater	0.0036	0.0072
3	mid-late	0.02462257	-0.666635	greater	0.8825	0.8825

# Column Test indicates which successional stages are compared; Obs = F statistic; Pvalue.adj = adjusted P-value

# Below, for each pair of compared succession stages, the histogram of simulated values with an arrow indicating the observed F value.

```
plot(PHketa)
```



## References

- Berkelaar, M. et al. (2020) lpSolve: Interface to 'Lp\_solve' v. 5.5 to Solve Linear/Integer Programs. R package version 5.6.15. <https://CRAN.R-project.org/package=lpSolve>
- Pavoine, S. (2020a) adiv: Analysis of Diversity. R package version 2.0. <https://CRAN.R-project.org/package=adiv>
- Pavoine, S. (2020b) adiv: an R package to analyse biodiversity in ecology. *Methods in Ecology and Evolution* 11: 1106-1112. <https://doi.org/10.1111/2041-210X.13430>
- Ricotta, C., Kosman, E., Laroche, F., Pavoine, S. (2021) Beta redundancy for functional ecology. *Methods in Ecology and Evolution*. <https://doi.org/10.1111/2041-210X.13587>
- Thioulouse, J., Dray, S., Dufour, A.-B., Siberchicot, A., Jombart, J., Pavoine, S. (2018) *Multivariate Analysis of Ecological Data with ade4*. Springer, New York.
- Wickham, H. (2016) *ggplot2: Elegant Graphics for Data Analysis*. Springer, New York.