Vignette ecospat package

Contents

==

Valeria Di Cola, Olivier Broennimann, Blaise Petitpierre, Manuela D'Amen, Frank Breiner & Antoine Guisan ##### 08 November, 2017

Miscellaneous methods and utilities for spatial ecology analysis, written by current and former members and collaborators of the ecospat group of Antoine Guisan, Department of Ecology and Evolution (DEE) & Institute of Earth Surface Dynamics (IDYST), University of Lausanne, Switzerland.

ecospat offers the possibility to perform Pre-modelling Analysis, such as Spatial autocorrelation analysis, MESS (Multivariate Environmental Similarity Surfaces) analyses, Phylogenetic diversity Measures, Biotic Interactions. It also provides functions to complement *biomod2* in preparing the data, calibrating and evaluating (e.g. boyce index) and projecting the models. Complementary analysis based on model predictions (e.g. co-occurrences analyses) are also provided.

In addition, the *ecospat* package includes Niche Quantification and Overlap functions that were used in Broennimann et al. 2012 and Petitpierre et al. 2012 to quantify climatic niche shifts between the native and invaded ranges of invasive species.

1 Load data

```r
library(ecospat)
citation("ecospat")
```

```
## To cite package 'ecospat' in publications use:
##
##   Broennimann O, Di Cola V, Guisan A (2026). _ecospat: Spatial Ecology
##   Miscellaneous Methods_. R package version 4.1.3,
##   <https://www.unil.ch/ecospat/home/menuguid/ecospat-resources.html>.
##
## A BibTeX entry for LaTeX users is
```

```
##
##   @Manual{,
##     title = {ecospat: Spatial Ecology Miscellaneous Methods},
##     author = {Olivier Broennimann and Valeria {Di Cola} and Antoine Guisan},
##     year = {2026},
##     note = {R package version 4.1.3},
##     url = {https://www.unil.ch/ecospat/home/menuguid/ecospat-resources.html},
##   }
```

1.0.1 Test data for the ecospat library

ecospat.testData()

```r
data(ecospat.testData)
names(ecospat.testData)
```

```
##  [1] "numplots"                         "long"
##  [3] "lat"                              "ddeg"
##  [5] "mind"                             "srad"
##  [7] "slp"                              "topo"
##  [9] "Achillea_atrata"                  "Achillea_millefolium"
## [11] "Acinos_alpinus"                   "Adenostyles_glabra"
## [13] "Aposeris_foetida"                 "Arnica_montana"
## [15] "Aster_bellidiastrum"              "Bartsia_alpina"
## [17] "Bellis_perennis"                  "Campanula_rotundifolia"
## [19] "Centaurea_montana"                "Cerastium_latifolium"
## [21] "Cruciata_laevipes"                "Doronicum_grandiflorum"
## [23] "Galium_album"                     "Galium_anisophyllon"
## [25] "Galium_megalospermum"             "Gentiana_bavarica"
## [27] "Gentiana_lutea"                   "Gentiana_purpurea"
## [29] "Gentiana_verna"                   "Globularia_cordifolia"
## [31] "Globularia_nudicaulis"            "Gypsophila_repens"
## [33] "Hieracium_lactucella"             "Homogyne_alpina"
## [35] "Hypochaeris_radicata"             "Leontodon_autumnalis"
## [37] "Leontodon_helveticus"             "Myosotis_alpestris"
## [39] "Myosotis_arvensis"                "Phyteuma_orbiculare"
## [41] "Phyteuma_spicatum"                "Plantago_alpina"
## [43] "Plantago_lanceolata"              "Polygonum_bistorta"
## [45] "Polygonum_viviparum"              "Prunella_grandiflora"
## [47] "Rhinanthus_alectorolophus"        "Rumex_acetosa"
## [49] "Rumex_crispus"                    "Vaccinium_gaultherioides"
## [51] "Veronica_alpina"                  "Veronica_aphylla"
## [53] "Agrostis_capillaris"              "Bromus_erectus_sstr"
## [55] "Campanula_scheuchzeri"            "Carex_sempervirens"
## [57] "Cynosurus_cristatus"              "Dactylis_glomerata"
## [59] "Daucus_carota"                    "Festuca_pratensis_sl"
## [61] "Geranium_sylvaticum"              "Leontodon_hispidus_sl"
## [63] "Potentilla_erecta"                "Pritzelago_alpina_sstr"
## [65] "Prunella_vulgaris"                "Ranunculus_acris_sl"
## [67] "Saxifraga_oppositifolia"          "Soldanella_alpina"
## [69] "Taraxacum_officinale_aggr"        "Trifolium_repens_sstr"
## [71] "Veronica_chamaedrys"              "Parnassia_palustris"
## [73] "glm_Agrostis_capillaris"          "glm_Leontodon_hispidus_sl"
## [75] "glm_Dactylis_glomerata"           "glm_Trifolium_repens_sstr"
## [77] "glm_Geranium_sylvaticum"          "glm_Ranunculus_acris_sl"
## [79] "glm_Prunella_vulgaris"            "glm_Veronica_chamaedrys"
## [81] "glm_Taraxacum_officinale_aggr" "glm_Plantago_lanceolata"
## [83] "glm_Potentilla_erecta"            "glm_Carex_sempervirens"
## [85] "glm_Soldanella_alpina"            "glm_Cynosurus_cristatus"
```

```
## [87] "glm_Campanula_scheuchzeri"     "glm_Festuca_pratensis_sl"
## [89] "glm_Bromus_erectus_sstr"       "glm_Saxifraga_oppositifolia"
## [91] "glm_Daucus_carota"             "glm_Pritzelago_alpina_sstr"
## [93] "gbm_Bromus_erectus_sstr"       "gbm_Saxifraga_oppositifolia"
## [95] "gbm_Daucus_carota"             "gbm_Pritzelago_alpina_sstr"
```

1.0.2 Test data for the Niche Overlap Analysis

ecospat.testNiche.inv()

```r
data(ecospat.testNiche.inv)
names(ecospat.testNiche.inv)
```

```
##  [1] "x"           "y"           "aetpet"      "gdd"         "p"
##  [6] "pet"         "stdp"        "tmax"        "tmin"        "tmp"
## [11] "species_occ" "predictions"
```

ecospat.testNiche.nat()

```r
data(ecospat.testNiche.nat)
names(ecospat.testNiche.nat)
```

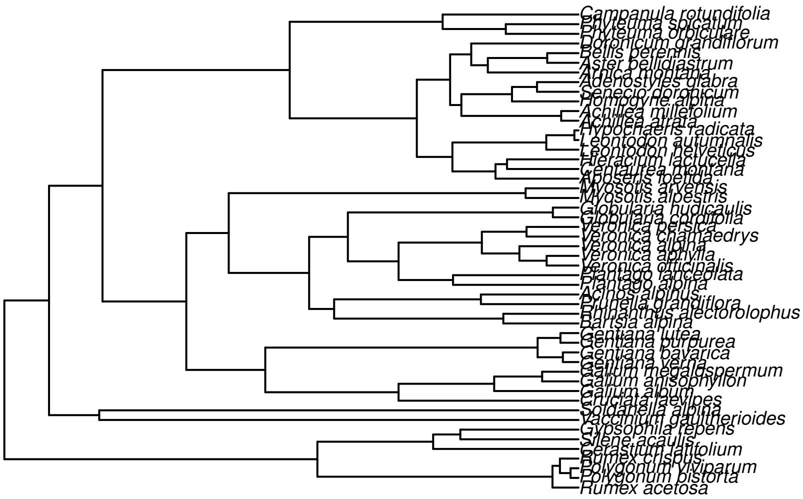
```
##  [1] "x"           "y"           "aetpet"      "gdd"         "p"
##  [6] "pet"         "stdp"        "tmax"        "tmin"        "tmp"
## [11] "species_occ" "predictions"
```

1.0.3 Test tree for Phylogenetic Diversity Analysis

ecospat.testTree()

```r
if(requireNamespace("ape")){
  fpath <- system.file("extdata", "ecospat.testTree.tre", package="ecospat")
  tree<-ape::read.tree(fpath)
  tree$tip.label
  plot(tree, cex=0.6)
}
```

```
## Loading required namespace: ape
```



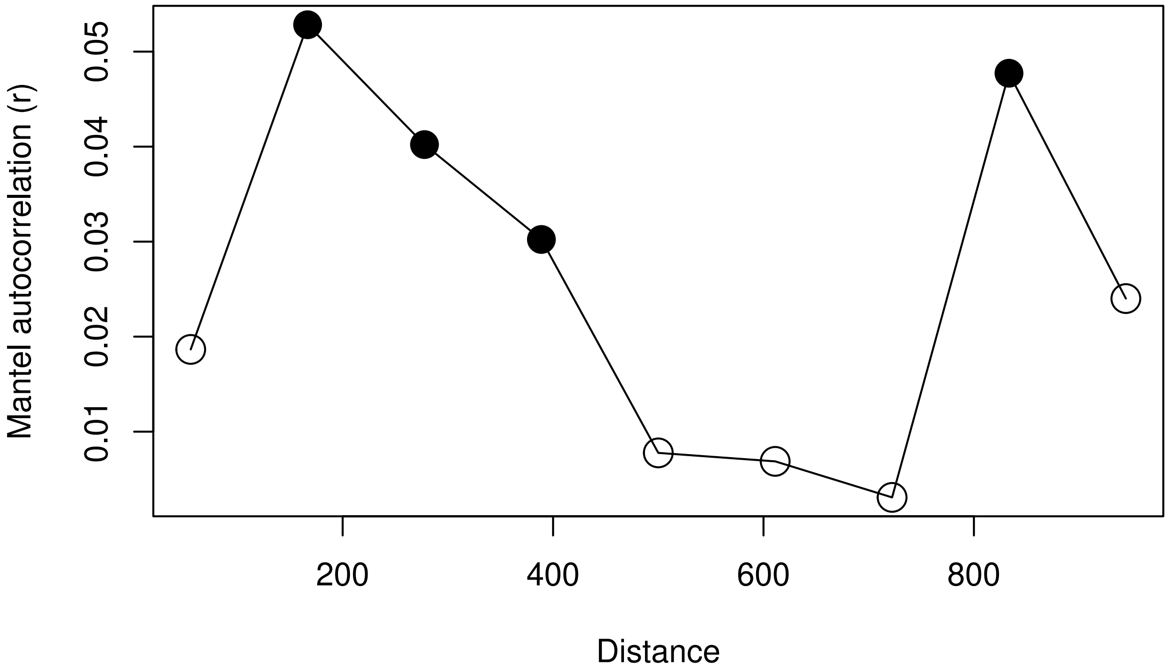
2 Pre-Modelling Analysis

2.1 Spatial Auto-correlation

2.1.1 Mantel Correlogram with *ecospat.mantel.correlogram()*

```
ecospat.mantel.correlogram(dfvar=ecospat.testData[c(2:16)],colxy=1:2, n=100,
                           colvar=3:7, max=1000, nclass=10, nperm=100)
```

```
## Registered S3 method overwritten by 'ecodist':
##   method   from
##   dim.dist proxy
```



The graph indicates that spatial autocorrelation (SA) is minimal at a distance of 180 meters. Note however that SA is not significantly different than zero for several distances (open circles).

2.2 Predictor Variable Selection

2.2.1 Number of Predictors with Pearson Correlation *ecospat.npred()*

```
colvar <- ecospat.testData[c(4:8)]
x <- cor(colvar, method="pearson")
ecospat.npred (x, th=0.75)
```

```
## [1] 4
```

2.2.2 Number of Predictors with Spearman Correlation *ecospat.npred()*

```
x <- cor(colvar, method="spearman")
ecospat.npred (x, th=0.75)
```

```
## [1] 4
```

2.3 Climate Analogy Tools

2.3.1 Climate Analogy with *ecospat.climan()*

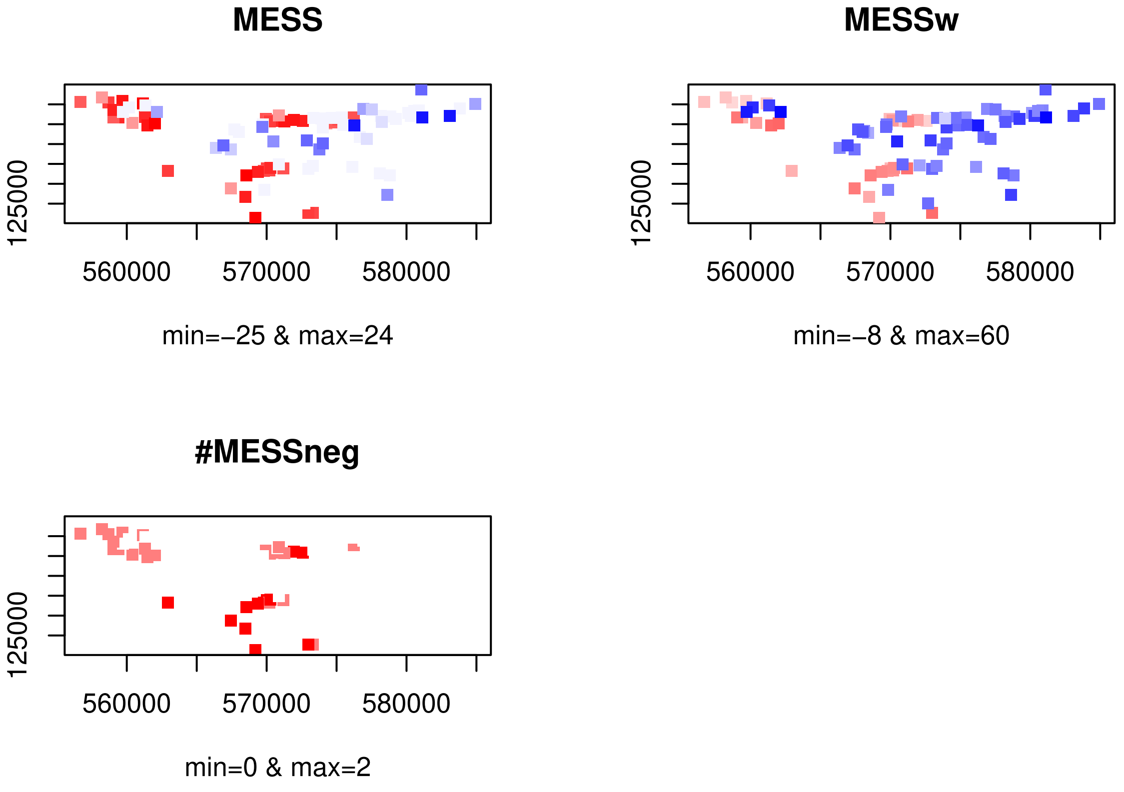
```r
x <- ecospat.testData[c(4:8)]
p<- x[1:90,] #A projection dataset.
ref<- x[91:300,] # A reference dataset
```

```r
ecospat.climan(ref,p)
```

```
##  [1]  0.185415746 -0.028290993 -0.032909931 -0.009237875 -0.034642032
##  [6] -0.209006928 -0.084295612 -0.103622863  0.355220600 -0.136258661
## [11] -0.087182448 -0.209006928 -0.143187067 -0.124711316 -0.114844720
## [16] -0.230596451  0.276046242  0.249093277 -0.125288684 -0.101226337
## [21] -0.113883908 -0.204653076 -0.001154734 -0.132217090 -0.100461894
## [26]  0.464738681 -0.416578541 -0.044457275 -0.018475751 -0.122225532
## [31] -0.137611720 -0.050808314  0.254605027 -0.062012319  0.238294633
## [36] -0.159141330 -0.147806005  0.277670365 -0.071593533 -0.019053118
## [41]  0.390781314  0.175132571  0.401892929  0.843703731  0.286155800
## [46]  0.321142114  0.668511130  0.252253209  0.440050672  0.177247206
## [51]  0.831525456  0.303710525  0.197182304  0.219273698  0.196637663
## [56]  0.195300816  0.142395786  0.176988160 -0.051991905  0.265163111
## [61] -0.020785219 -0.017898383  0.553965995  0.409635110  0.323633285
## [66]  0.468693064  0.124983005 -0.032909931  0.165642783  0.147046687
## [71]  0.202895471  0.341992334  0.225508458  0.133254065  0.485295264
## [76] -0.047344111 -0.012282931  0.165429659  0.134199992  0.216655251
## [81]  0.139419127  0.121254775  0.098782992  0.591393741  0.110866239
## [86]  0.146010655  0.095562156  0.093353356  0.081712342  0.160531262
```

2.3.2 Extrapolation detection, creating a MESS object with *ecospat.mess()*

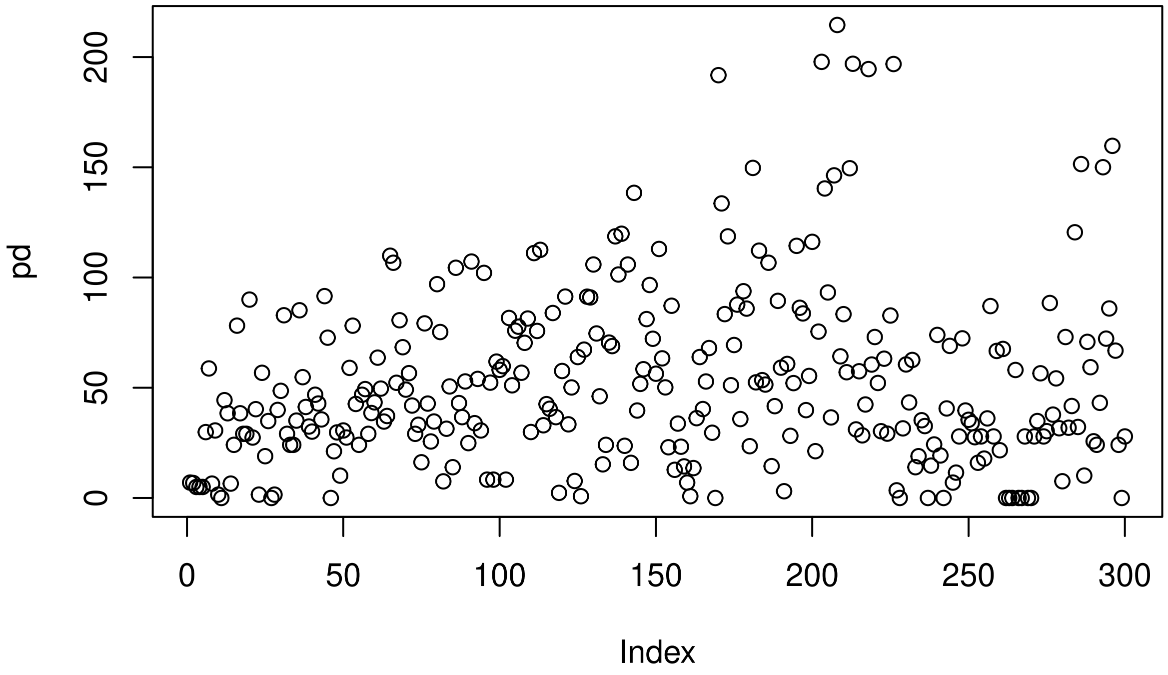
```r
x <- ecospat.testData[c(2,3,4:8)]
proj<- x[1:90,] #A projection dataset.
cal<- x[91:300,] #A calibration dataset
mess.object<-ecospat.mess (proj, cal, w="default")
ecospat.plot.mess (mess.object, cex=1, pch=15)
```



In the MESS plot pixels in red indicate sites where at least one environmental predictor has values outside of the range of that predictor in the calibration dataset. In the MESSw plot, same as previous plot but with weighted by the number of predictors.Finally, the MESSneg plot shows at each site how many predictors have values outside of their calibration range.

2.4 Phylogenetic Diversity Measures

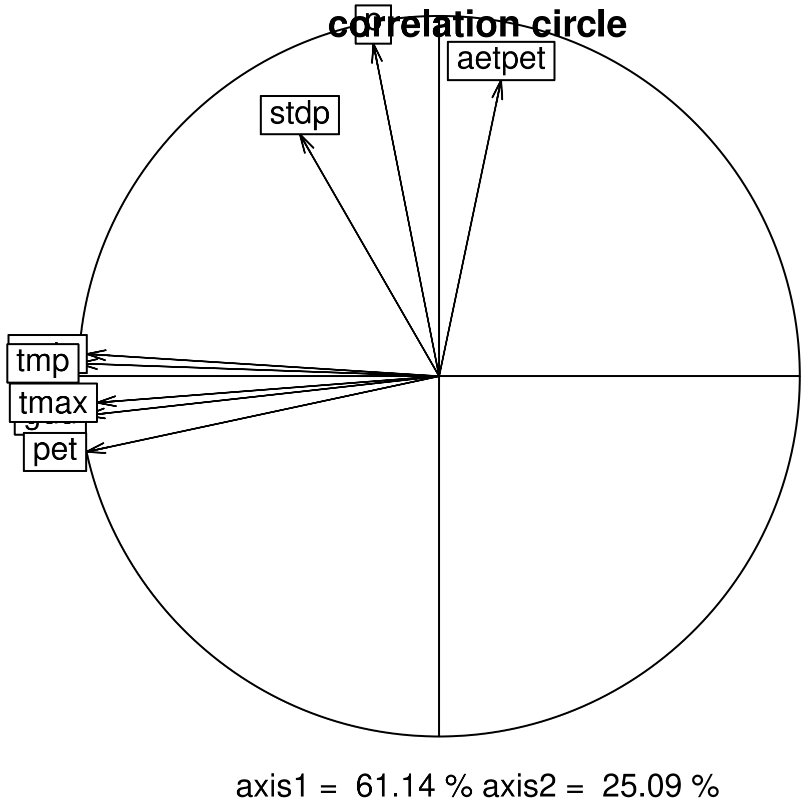
```
if(requireNamespace("ape")){
  fpath <- system.file("extdata", "ecospat.testTree.tre", package="ecospat")
  tree <- ape::read.tree(fpath)
  data <- ecospat.testData[9:52]
  pd<- ecospat.calculate.pd(tree, data, method = "spanning", type = "species", root = TRUE, average
  plot(pd)
}
```



2.5 Niche Quantification and Comparison with Ordination techniques

First we load the test data for the niche dynamics analysis in invaded and native range. A PCA is calibrated on all the sites of the study area, including both native and invaded ranges (same as PCAenv in Broenniman et al. 2012). Finally, we plot the variables Contributions

```
library(ade4)
inv <- ecospat.testNiche.inv
nat <- ecospat.testNiche.nat
pca.env <- ade4::dudi.pca(rbind(nat,inv)[,3:10],scannf=F,nf=2)
ecospat.plot.contrib(contrib=pca.env$co, eigen=pca.env$eig)
```



The correlation circle indicate the contribution of original predictors to the PCA axes.

Now we can predict the scores on the axes

```
# PCA scores for the whole study area
scores.globclim <- pca.env$li

# PCA scores for the species native distribution
scores.sp.nat <- ade4::suprow(pca.env,nat[which(nat[,11]==1),3:10])$li

# PCA scores for the species invasive distribution
scores.sp.inv <- ade4::suprow(pca.env,inv[which(inv[,11]==1),3:10])$li

# PCA scores for the whole native study area
scores.clim.nat <- ade4::suprow(pca.env,nat[,3:10])$li

# PCA scores for the whole invaded study area
scores.clim.inv <- ade4::suprow(pca.env,inv[,3:10])$li
```

2.5.1 Calculate the Occurrence Densities Grid with *ecospat.grid.clim.dyn()*

For a species in the native range (North America)

```
# gridding the native niche
grid.clim.nat <- ecospat.grid.clim.dyn(glob=scores.globclim,
                                       glob1=scores.clim.nat,
                                       sp=scores.sp.nat, R=100,
                                       th.sp=0)
```

```
## Registered S3 methods overwritten by 'adehabitatMA':
##   method                       from
##   print.SpatialPixelsDataFrame sp
##   print.SpatialPixels          sp
```

For a species in the invaded range (Australia)

```r
# gridding the invasive niche
grid.clim.inv <- ecospat.grid.clim.dyn(glob=scores.globclim,
                                       glob1=scores.clim.inv,
                                       sp=scores.sp.inv, R=100,
                                       th.sp=0)
```

2.5.2 Calculate Niche Overlap with *ecospat.niche.overlap()*

```r
# Compute Schoener's D, index of niche overlap
D.overlap <- ecospat.niche.overlap (grid.clim.nat, grid.clim.inv, cor = TRUE)$D
D.overlap
```

```
## [1] 0.224586
```

The niche overlap between the native and the invaded range is 22%.

2.5.3 Perform the Niche Equivalency Test with *ecospat.niche.equivalency.test()* according to Warren et al. (2008)

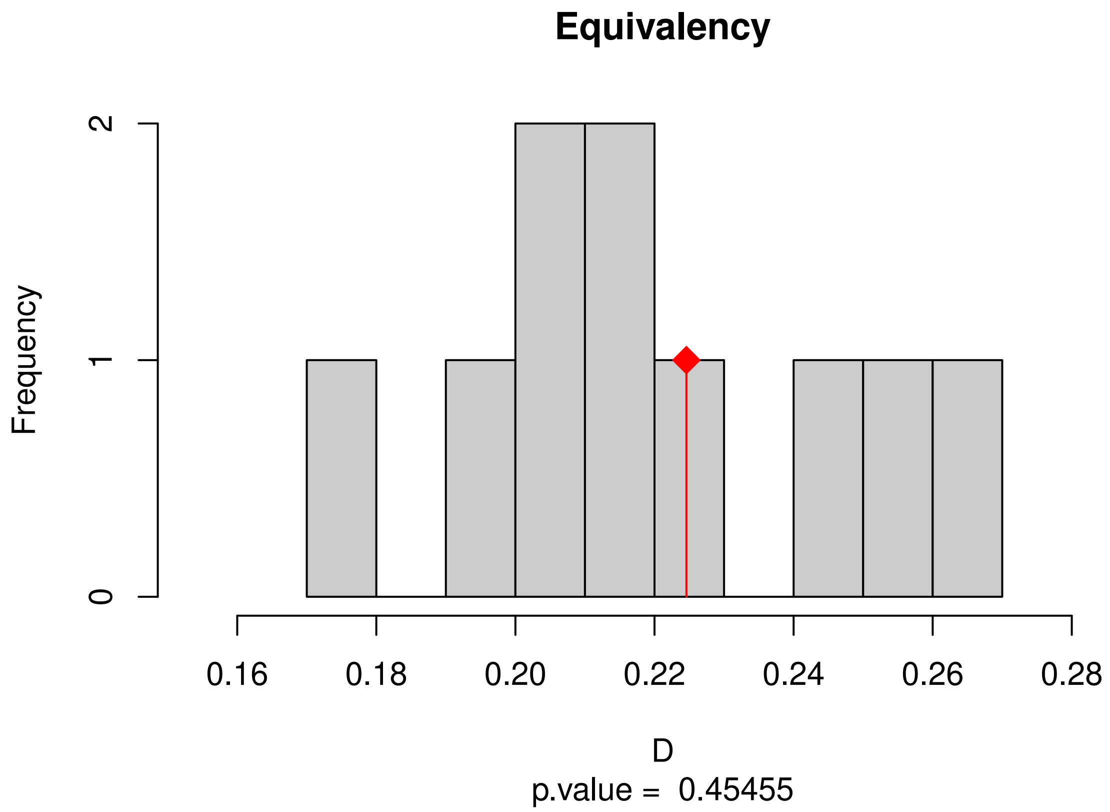
It is reccomended to use at least 1000 replications for the equivalency test. As an example we used rep = 10, to reduce the computational time.

```r
eq.test <- ecospat.niche.equivalency.test(grid.clim.nat, grid.clim.inv,rep=10,
                                          intersection = 0.1,
                                          overlap.alternative =  "higher",
                                          expansion.alternative = "lower",
                                          stability.alternative = "higher",
                                          unfilling.alternative = "lower")
```

Niche equivalency test H1: the observed overlap between the native and invaded niche is higher than if the two niches are randomized, the niche expansion of the invaded niche is lower, the niche stability is higher and the niche unfilling is lower than if the two niches are randomized.

Plot Equivalency test

```r
ecospat.plot.overlap.test(eq.test, "D", "Equivalency")
```



2.5.4 Perform the Niche Similarity Test with *ecospat.niche.similarity.test()*

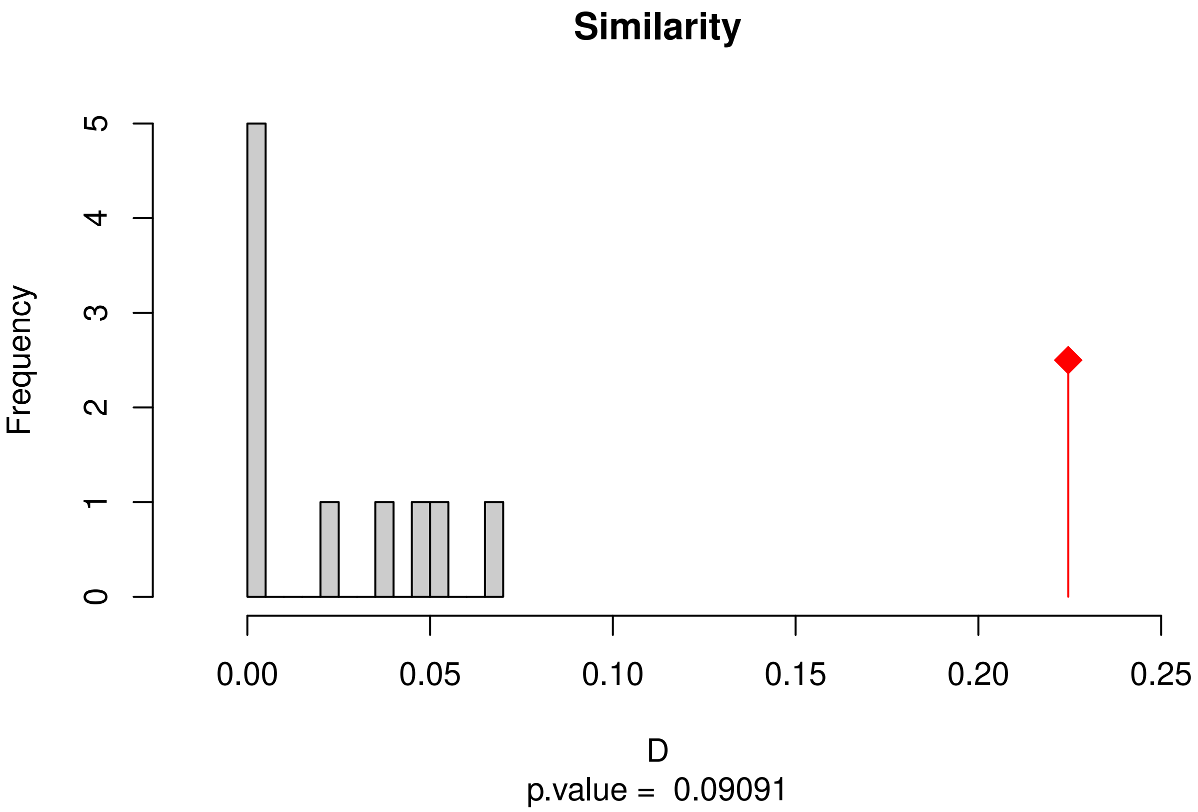
Shifts randomly on niche (here the invasive niche) in the study area It is recomended to use at least 1000 replications for the similarity test. As an example we used rep = 10, to reduce the computational time.

```
sim.test <- ecospat.niche.similarity.test(grid.clim.nat, grid.clim.inv,rep=10,
                                          overlap.alternative =  "higher",
                                          expansion.alternative = "lower",
                                          stability.alternative = "higher",
                                          unfilling.alternative = "lower",
                                          intersection = 0.1,
                                          rand.type=1)
```

Niche similarity test H1: the observed overlap between the native and invaded is higher than randomly shifted invasive niches in the invaded study area, the niche expansion of the invaded niche is lower, the niche stability is higher and the niche unfilling is lower than if the two niches are randomized.

Plot Similarity test

```
ecospat.plot.overlap.test(sim.test, "D", "Similarity")
```



2.5.5 Delimiting niche categories and quantifying niche dynamics in analogue climates with *ecospat.niche.dyn.index()*

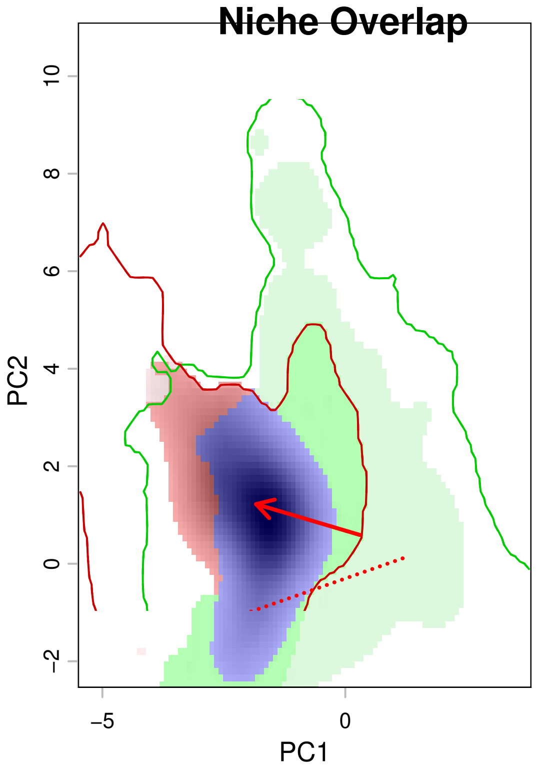
```
niche.dyn <- ecospat.niche.dyn.index (grid.clim.nat, grid.clim.inv)
```

2.5.5.1 Visualizing niche categories, niche dynamics and climate analogy between ranges with *ecospat.plot.niche.dyn()*

Plot niche overlap

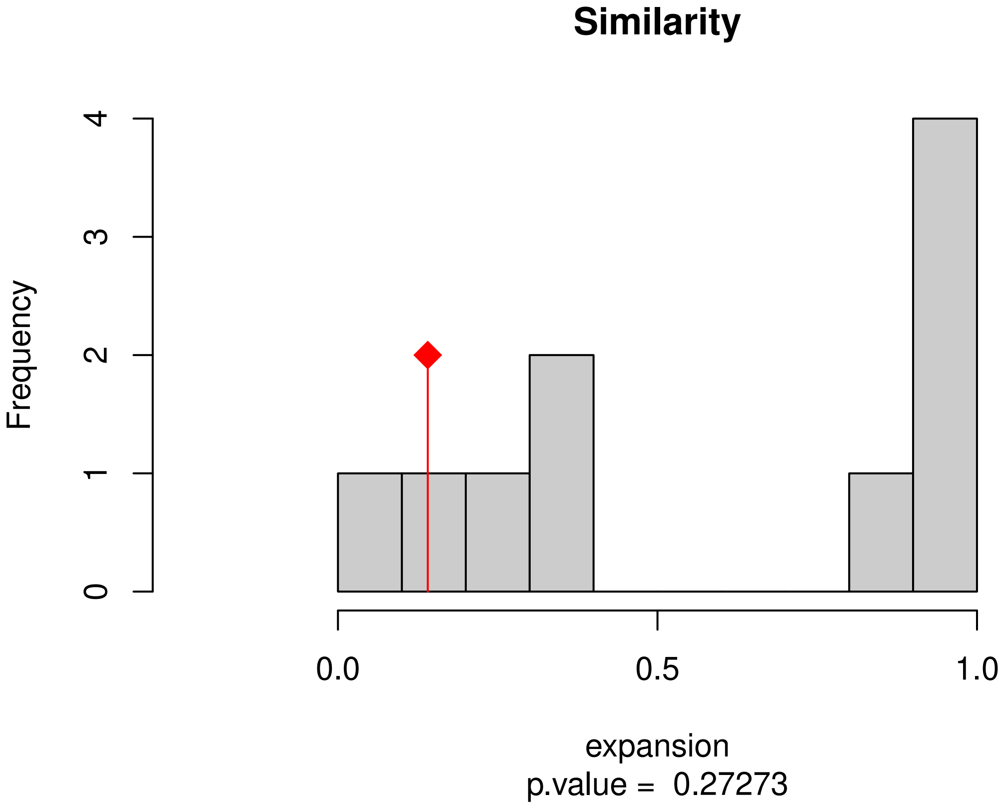
```
ecospat.plot.niche.dyn(grid.clim.nat, grid.clim.inv, quant=0.25, interest=2,
                       title= "Niche Overlap", name.axis1="PC1",
                       name.axis2="PC2")

ecospat.shift.centroids(scores.sp.nat, scores.sp.inv, scores.clim.nat, scores.clim.inv)
```

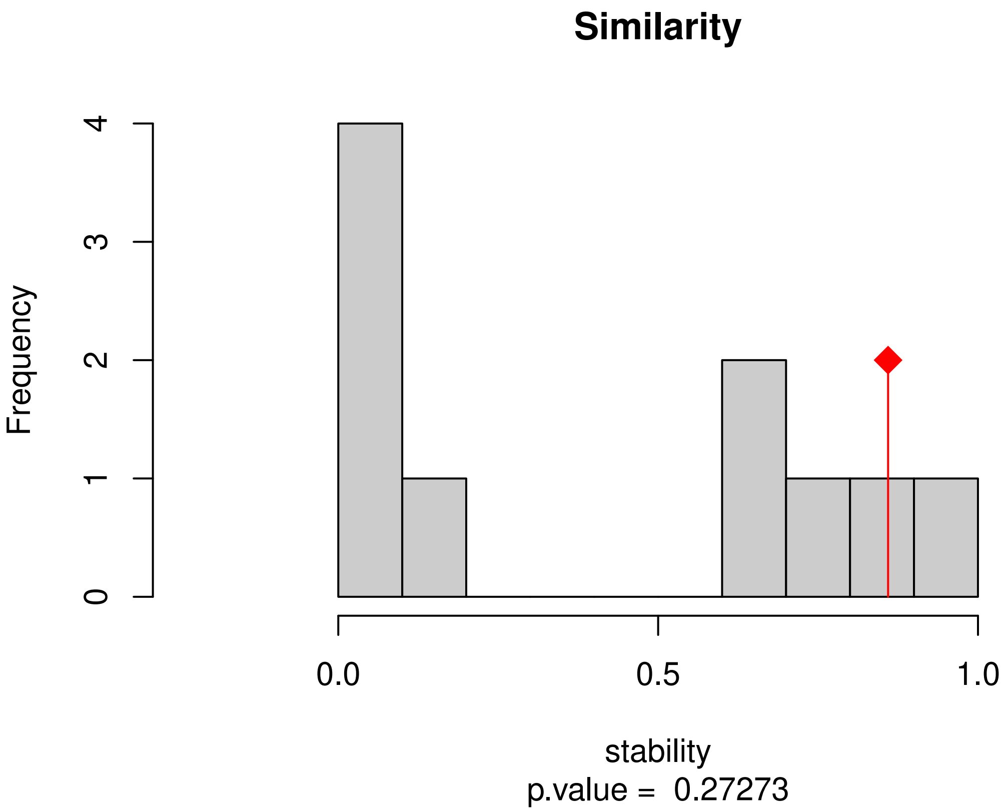


Plot Similarity test for niche expansion, stability and unfilling

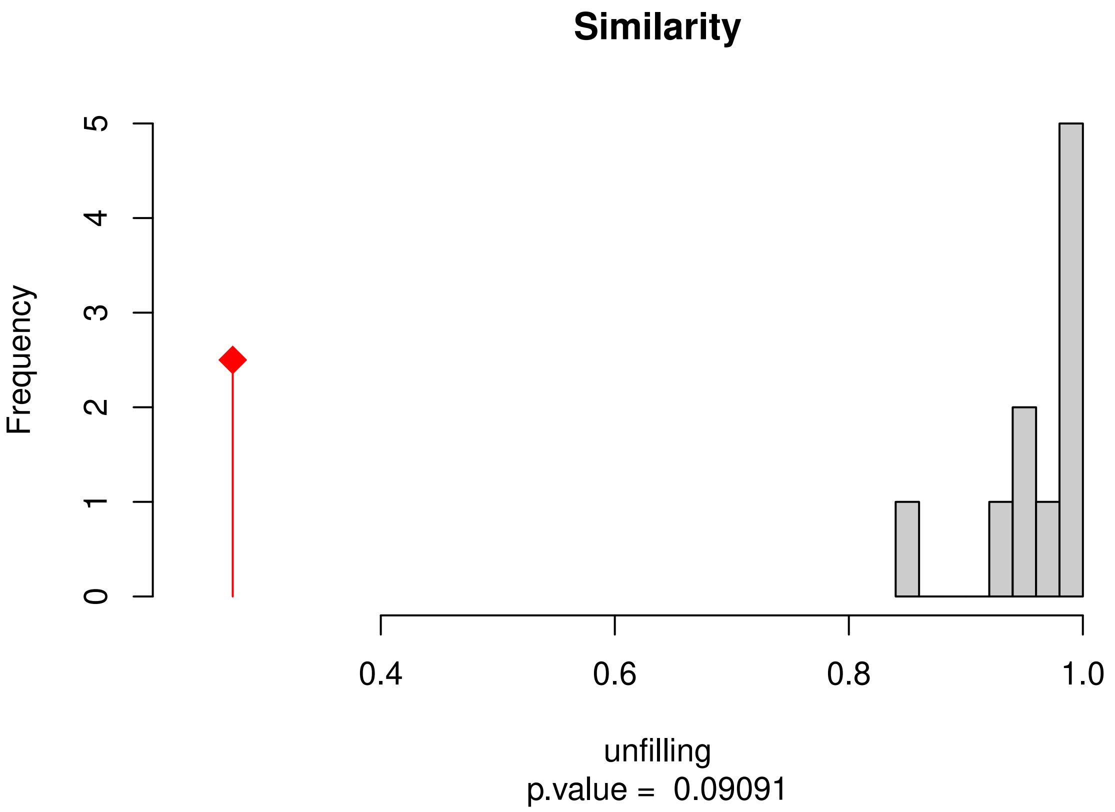
```
ecospat.plot.overlap.test(sim.test, "expansion", "Similarity")
```



```
ecospat.plot.overlap.test(sim.test, "stability", "Similarity")
```



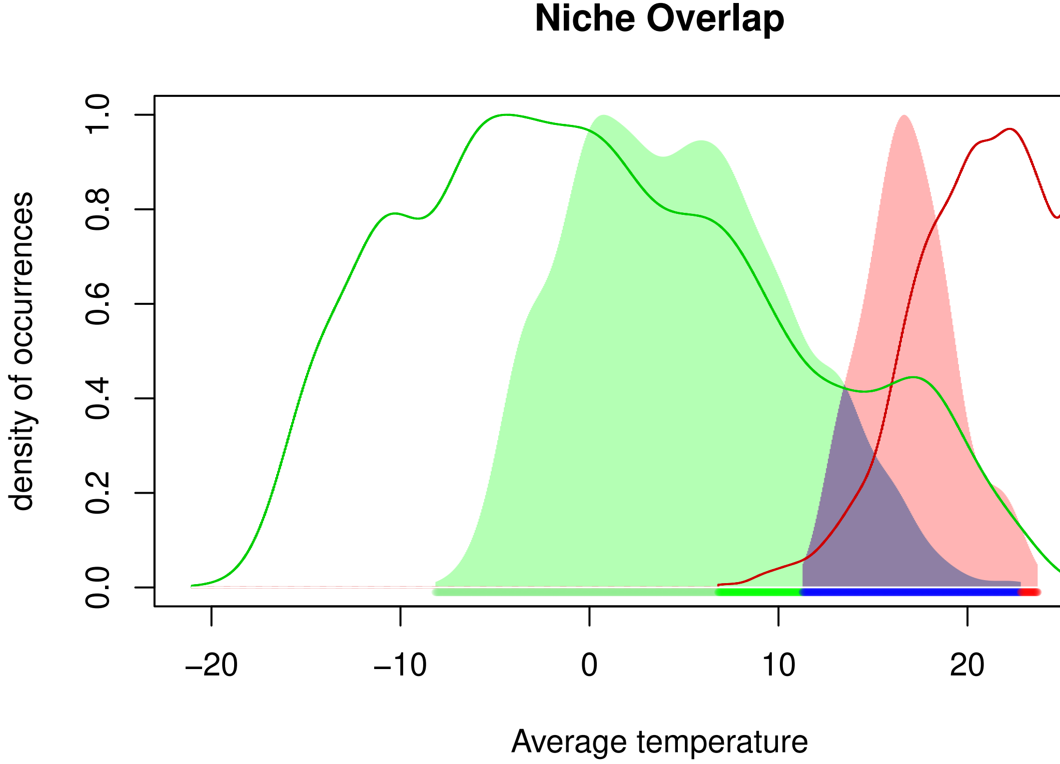
```
ecospat.plot.overlap.test(sim.test, "unfilling", "Similarity")
```



```
# gridding the native niche
grid.clim.t.nat <- ecospat.grid.clim.dyn(glob=as.data.frame(rbind(nat,inv)[,10]),
                                          glob1=as.data.frame(nat[,10]),
```

```
                                    sp=as.data.frame(nat[which(nat[,11]==1),10]),
                                    R=1000, th.sp=0)
# gridding the invaded niche
grid.clim.t.inv <- ecospat.grid.clim.dyn(glob=as.data.frame(rbind(nat,inv)[,10]),
                                         glob1=as.data.frame(inv[,10]),
                                         sp=as.data.frame(inv[which(inv[,11]==1),10]),
                                         R=1000, th.sp=0)
t.dyn<-ecospat.niche.dyn.index (grid.clim.t.nat, grid.clim.t.inv)
ecospat.plot.niche.dyn(grid.clim.t.nat, grid.clim.t.inv, quant=0,
                       interest=2, title= "Niche Overlap",
                       name.axis1="Average temperature")
```

2.5.5.2 Plot the niche dynamics along one gradient (here temperature) with *ecospat.plot.niche.dyn()*



2.6 Biotic Interactions

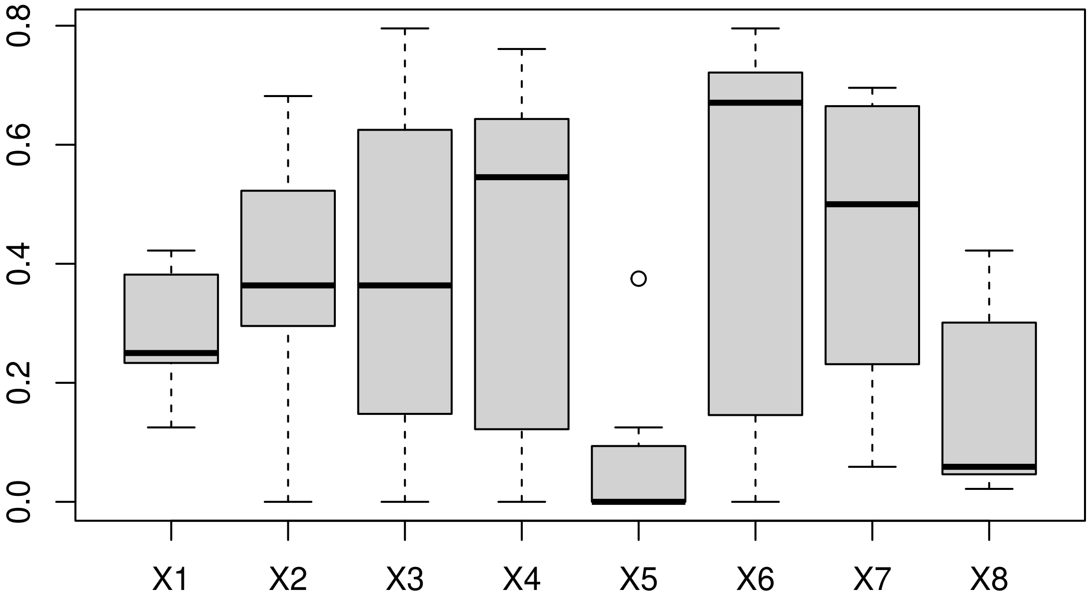
2.6.1 Species Co-occurrences Analysis with a Presence-absence matrix using the function *ecospat.co_occurrences()*

```
data <- ecospat.testData[c(9:16,54:57)]
```

For each pair of species (sp1, sp2), the number (N) of plots where both species were present is divided by the number of plots where the rarest of the two species is present. This index ranges from 0 (no co-occurrence) to 1 (always in co-occurrence) as given in eq. 1.

where N(S1 intersects S2) is the number of times species S1 and S2 co-occur, while Min(NS1, NS2) is the number of times species S1 and S2 co-occur, while is the occurrence frequency of the rarest of the two species.

```
ecospat.co_occurrences (data)
```



```
##                       Aposeris_foetida Arnica_montana Aster_bellidiastrum
## Aposeris_foetida             1.0000000      0.3636364          0.25000000
## Arnica_montana               0.3636364      1.0000000          0.36363636
## Aster_bellidiastrum          0.2500000      0.3636364          1.00000000
## Bartsia_alpina               0.2222222      0.5454545          0.59090909
## Bromus_erectus_sstr          0.1250000      0.0000000          0.00000000
## Campanula_scheuchzeri        0.2444444      0.6818182          0.79545455
## Carex_sempervirens           0.4000000      0.5000000          0.65909091
## Cynosurus_cristatus          0.4222222      0.2272727          0.04545455
##                       Bartsia_alpina Bromus_erectus_sstr Campanula_scheuchzeri
## Aposeris_foetida          0.22222222              0.1250            0.24444444
## Arnica_montana            0.54545455              0.0000            0.68181818
## Aster_bellidiastrum       0.59090909              0.0000            0.79545455
## Bartsia_alpina            1.00000000              0.0000            0.76086957
## Bromus_erectus_sstr       0.00000000              1.0000            0.00000000
## Campanula_scheuchzeri     0.76086957              0.0000            1.00000000
## Carex_sempervirens        0.69565217              0.0625            0.67058824
## Cynosurus_cristatus       0.02173913              0.3750            0.04705882
##                       Carex_sempervirens Cynosurus_cristatus
## Aposeris_foetida              0.40000000          0.42222222
## Arnica_montana                0.50000000          0.22727273
## Aster_bellidiastrum           0.65909091          0.04545455
## Bartsia_alpina                0.69565217          0.02173913
## Bromus_erectus_sstr           0.06250000          0.37500000
## Campanula_scheuchzeri         0.67058824          0.04705882
## Carex_sempervirens            1.00000000          0.05882353
## Cynosurus_cristatus           0.05882353          1.00000000
```

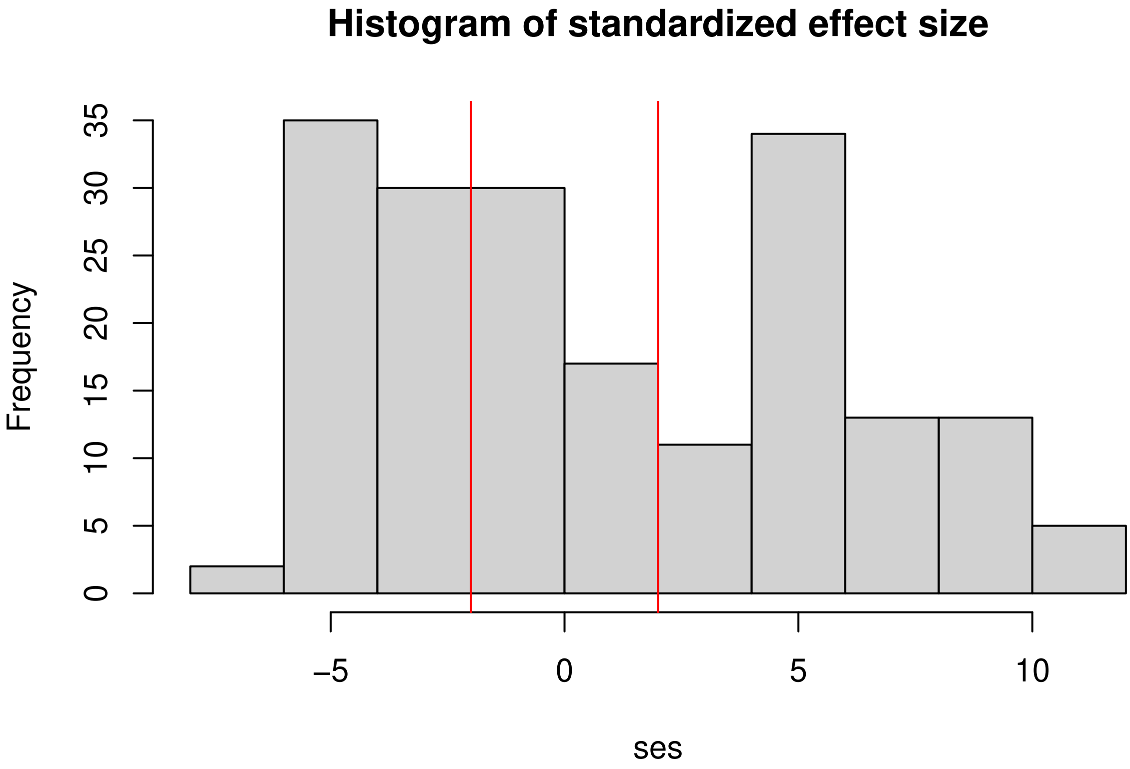
2.6.2 Pairwise co-occurrence Analysis with calculation of the C-score index using the function *ecospat.Cscore()*

This function allows to apply a pairwise null model analysis to a presence-absence community matrix to determine which species associations are significant across the study area. The strength of associations is quantified by the C-score index and a 'fixed-equiprobable' null model algorithm is applied.

It is recomended to use at least 10000 permutatiobns for the test. As an example we used nperm = 100, to reduce the computational time.

```
data<- ecospat.testData[c(53,62,58,70,61,66,65,71,69,43,63,56,68,57,55,60,54,67,59,64)]
nperm <- 100
```

```
outpath <- getwd()
ecospat.Cscore(data, nperm, outpath)
```



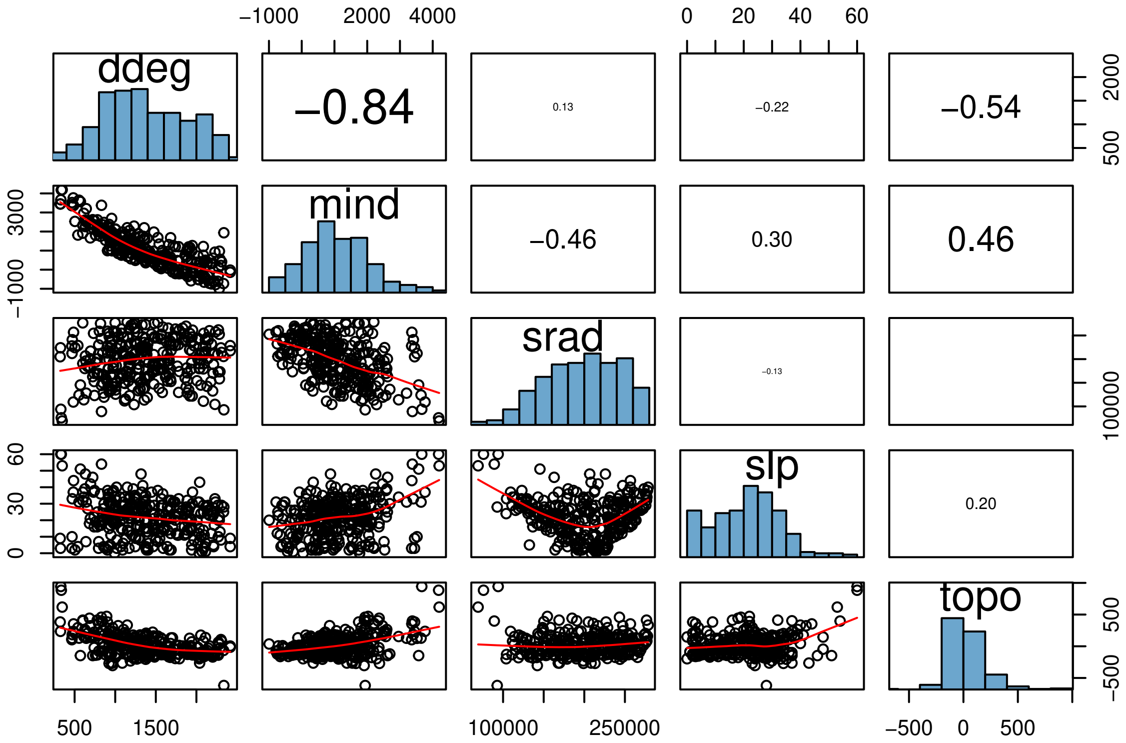
```
## $ObsCscoreTot
## [1] 2675.468
##
## $SimCscoreTot
## [1] 2465.931
##
## $PVal.less
## [1] 1
##
## $PVal.greater
## [1] 0.00990099
##
## $SES.Tot
## [1] 54.55579
```

The function returns the C-score index for the observed community (ObsCscoreTot), p.value (PValTot) and standardized effect size (SES.Tot). It saves also a table in the working directory where the same metrics are calculated for each species pair (only the table with species pairs with significant p-values is saved in this version)

2.7 Data Preparation

2.7.1 Correlation Plot of Variables with *ecospat.cor.plot()*

```
data <- ecospat.testData[,4:8]
ecospat.cor.plot(data)
```



A scatter plot of matrices, with bivariate scatter plots below the diagonal, histograms on the diagonal, and the Pearson correlation above the diagonal. Useful for descriptive statistics of small data sets (better with less than 10 variables).

2.7.2 Calibration And Evaluation Dataset

```r
data <- ecospat.testData
caleval <- ecospat.caleval (data = ecospat.testData[53], xy = data[2:3],
                            row.num = 1:nrow(data), nrep = 2, ratio = 0.7,
                            disaggregate = 0.2, pseudoabs = 100, npres = 10,
                            replace = FALSE)
head(caleval)
```

```
## $eval
##     yeval yeval
## 1      92    NA
## 2     160    NA
## 3     119    NA
## 4     152   254
## 5     238   239
## 6     220   260
## 7     235   211
## 8     242    57
## 9     121    84
## 10    296    34
## 11    113   184
## 12    214   300
## 13    204   133
## 14    116   253
## 15    189   198
## 16     85   120
## 17    283   272
## 18     67   292
## 19    155   243
## 20    229   212
```

```
## 21   203   225
## 22   225   248
## 23   295   275
## 24   293   156
## 25   273   189
## 26   206   231
## 27   250   281
## 28   217   258
## 29   248   145
## 30   134   262
## 31    30   278
## 32   246   266
## 33   114   249
##
## $cal
##    ycal ycal
## 1     99   NA
## 2     NA   NA
## 3     90   NA
## 4     NA   NA
## 5     NA   NA
## 6     54   NA
## 7      9  103
## 8    224   53
## 9     56  200
## 10   251  188
## 11   266   79
## 12    55   85
## 13   166   31
## 14   228   17
## 15   244  139
## 16   180  274
## 17   291   95
## 18    37   15
## 19    14  240
## 20   145  100
## 21   232  154
## 22     3  233
## 23   147   18
## 24    27  106
## 25   276    8
## 26   212  178
## 27    23   36
## 28     2  265
## 29   271   51
## 30   110    3
## 31   278   67
## 32    24   21
## 33   258  181
## 34   261  171
## 35   297   43
## 36   123  115
## 37    33  222
## 38   279  270
## 39   223  293
## 40     4   33
## 41   264  289
## 42   290   49
```

```
## 43  275  204
## 44  289  273
## 45  186  221
## 46  294  259
## 47  249   44
## 48  267  245
## 49  219  290
## 50  230  252
## 51   11  168
## 52   45   16
## 53  262  255
## 54  263  203
## 55  247  269
## 56  185  236
## 57  193  182
## 58   20  256
## 59  268  205
## 60  231  169
## 61   94  246
## 62   16  201
## 63  237   45
## 64  243    5
## 65  281  291
## 66  177   24
## 67  240  234
## 68  192  196
## 69  140  224
## 70  210  230
## 71  288  206
## 72  286  199
## 73  150  299
## 74   75   71
## 75   53   22
## 76  157  283
## 77  156  241
```

We obtained an evaluation and calibration dataset with a desired ratio of disaggregation.

3 Core Niche Modelling

3.1 Model Evaluation

3.1.1 Presence-only Evaluation Indices- Boyce Index

The argument fit is a vector containing the predicted suitability values

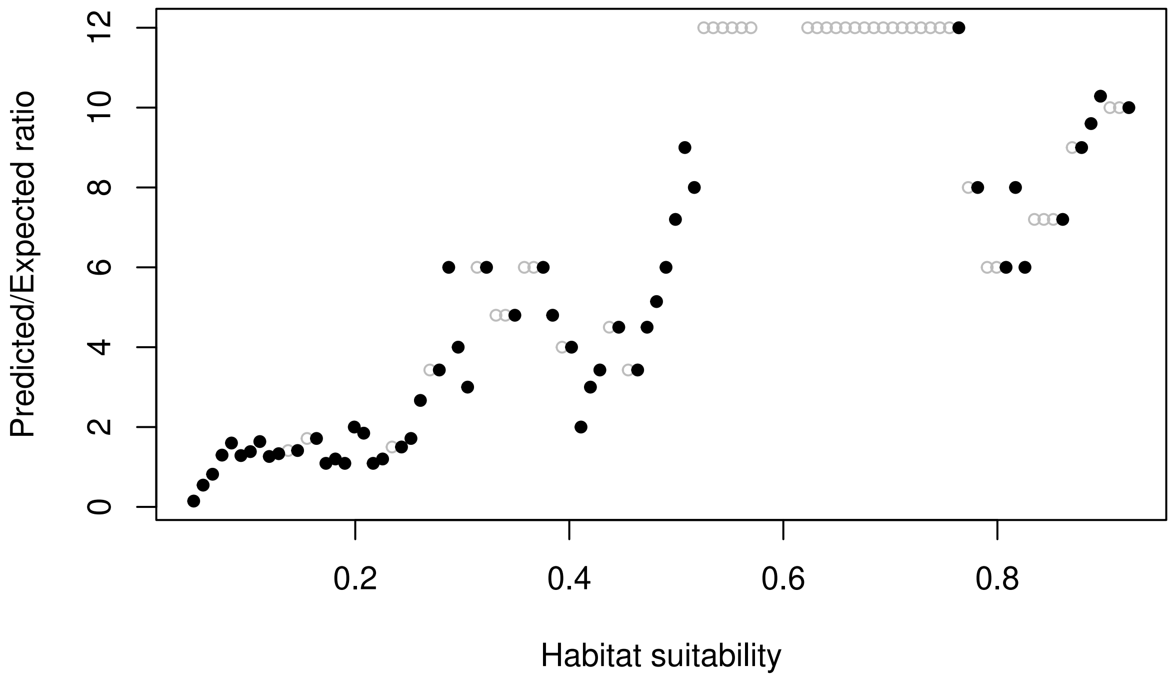
```
fit <- ecospat.testData$glm_Saxifraga_oppositifolia
```

The argument obs is a vector containing the predicted suitability values of the validation points (presence records)

```
obs<-ecospat.testData$glm_Saxifraga_oppositifolia[which(ecospat.testData$Saxifraga_oppositifolia==1)
```

Calculate and plot Boyce Index with *ecospat.boyce*

```
ecospat.boyce (fit, obs, nclass = 0, window.w = "default", res = 100,
               PEplot = TRUE)$cor
```



```
## [1] 0.91
```

Here the boyce index is 0.91. If the rank of predicted expected ratio would be completely ordered along habitat suitability axis then boyce index would be 1.

3.1.2 Accuracy of Community Prediction

Indices of accuracy of community predictions *ecospat.CommunityEval()*

```
eval<-ecospat.testData[c(53,62,58,70,61,66,65,71,69,43,63,56,68,57,55,60,54,67,59,64)]
pred<-ecospat.testData[c(73:92)]
```

```
CommunityEval<-ecospat.CommunityEval (eval, pred, proba = TRUE, ntir=5,verbose = T)
```

```
## trial 1 on 5
## trial 2 on 5
## trial 3 on 5
## trial 4 on 5
## trial 5 on 5
```

3.2 Spatial Predictions and Projections

3.2.1 ESM Ensemble of Small Models

```
library(biomod2)
```

```
## biomod2 4.3-4-3 loaded.
##  /!\ Welcome to augmented biomod2 with abundance modeling available! (*o*)
##  Take a look at the HOME and NEWS section on the website to see all the features!
```

```
## Loading required package: nnet
```

```
## Loading required package: rpart
```

```
## Loading required package: cito
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'cito'
```

```
## Loading required package: mda
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'mda'
```

```
## Loading required package: gam
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'gam'
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.9-3. For overview type 'help("mgcv-package")'.
```

```
##
## Attaching package: 'mgcv'
```

```
## The following object is masked from 'package:nnet':
##
##     multinom
```

```
## Loading required package: gbm
```

```
## Loaded gbm 2.2.2
```

```
## This version of gbm is no longer under development. Consider transitioning to gbm3, https://githu
```

```
## Loading required package: earth
```

```
## Loading required package: Formula
```

```
## Loading required package: plotmo
```

```
## Loading required package: plotrix
```

```
## Loading required package: maxnet
```

```
## Loading required package: randomForest
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
## Loading required package: xgboost
```

```r
# species
# occurrences
xy <- inv[,1:2]
head(xy)
```

```
##        x      y
## 1 142.25 -10.25
## 2 142.25 -10.75
## 3 131.25 -11.25
## 4 132.25 -11.25
## 5 142.25 -11.25
## 6 142.75 -11.25
```

```r
sp_occ <- inv[11]

# env
current <- inv[3:7]
head(current)
```

```
##      aetpet    gdd      p      pet     stdp
## 1 0.3180346 7965.1 1595.7 1950.320 137.8134
## 2 0.2807616 7888.9 1693.7 1991.475 156.3950
## 3 0.2638533 8165.3 1595.0 2179.968 127.0621
## 4 0.2790938 8195.6 1346.0 1919.897 114.7686
```

```
## 5 0.3030646 7858.1 1711.1 1795.255 158.3286
## 6 0.3217786 7888.5 1711.1 1788.220 151.8030
```

```
## BIOMOD
t1 <- Sys.time()
sp<-1
```

```
### Formating the data with the BIOMOD_FormatingData() function form the package biomod2

myBiomodData <- biomod2::BIOMOD_FormatingData( resp.var = as.numeric(sp_occ[,sp]),
                                       expl.var = current,
                                       resp.xy = xy,
                                       resp.name = colnames(sp_occ)[sp])
```

```
##
## -=-=-=-=-=-=-=-=-=-=-=-=-=-= species_occ Data Formating -=-=-=-=-=-=-=-=-=-=-=-=-=-=
##
##       ! Response variable name was converted into species.occ
## -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-= Done -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

```
### Calibration of simple bivariate models

# remove insivible(capture.output)) to print output in the console
# this is just to keep the vignette short
invisible(capture.output(my.ESM <- ecospat.ESM.Modeling( data=myBiomodData,
                                 models=c('GLM'),
                                 NbRunEval=2,
                                 DataSplit=70,
                                 weighting.score=c("AUC"),
                                 parallel=F)
        )
)
```

```
## Warning in .BIOMOD_Modeling.check.args(bm.format = bm.format, modeling.id =
## modeling.id, : The metric 'ROC' will be switch to 'AUCroc'.

## Warning: executing %dopar% sequentially: no parallel backend registered

## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.

## Warning in .BIOMOD_Modeling.check.args(bm.format = bm.format, modeling.id =
## modeling.id, : The metric 'ROC' will be switch to 'AUCroc'.

## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
```

```
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
```

```
## Warning in .BIOMOD_Modeling.check.args(bm.format = bm.format, modeling.id =
## modeling.id, : The metric 'ROC' will be switch to 'AUCroc'.
```

```
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
```

```
## Warning in .BIOMOD_Modeling.check.args(bm.format = bm.format, modeling.id =
## modeling.id, : The metric 'ROC' will be switch to 'AUCroc'.
```

```
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
```

```
## Warning in .BIOMOD_Modeling.check.args(bm.format = bm.format, modeling.id =
## modeling.id, : The metric 'ROC' will be switch to 'AUCroc'.
```

```
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
```

```
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
```

```
## Warning in .BIOMOD_Modeling.check.args(bm.format = bm.format, modeling.id =
## modeling.id, : The metric 'ROC' will be switch to 'AUCroc'.
```

```
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
```

```
## Warning in .BIOMOD_Modeling.check.args(bm.format = bm.format, modeling.id =
## modeling.id, : The metric 'ROC' will be switch to 'AUCroc'.
```

```
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
```

```
## Warning in .BIOMOD_Modeling.check.args(bm.format = bm.format, modeling.id =
## modeling.id, : The metric 'ROC' will be switch to 'AUCroc'.
```

```
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
```

```
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
```

```
## Warning in .BIOMOD_Modeling.check.args(bm.format = bm.format, modeling.id =
## modeling.id, : The metric 'ROC' will be switch to 'AUCroc'.
```

```
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
```

```
## Warning in .BIOMOD_Modeling.check.args(bm.format = bm.format, modeling.id =
## modeling.id, : The metric 'ROC' will be switch to 'AUCroc'.
```

```
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
## Warning in coords.roc(roc1, "best", ret = c("threshold", "sens", "spec"), :
## 'transpose=TRUE' is deprecated. Only 'transpose=FALSE' will be allowed in a
## future version.
```

```r
### Evaluation and average of simple bivariate models to ESMs
my.ESM_EF <- ecospat.ESM.EnsembleModeling(my.ESM,weighting.score=c("SomersD"),threshold=0)
```

```r
### Projection of simple bivariate models into new space
my.ESM_proj_current <- ecospat.ESM.Projection(ESM.modeling.output=my.ESM,
                                              new.env=current)
```

```r
### Projection of calibrated ESMs into new space
my.ESM_EFproj_current <- ecospat.ESM.EnsembleProjection(ESM.prediction.output=my.ESM_proj_current,
                                                        ESM.EnsembleModeling.output=my.ESM_EF)
```

3.3 Spatial prediction of communities

Input data for the first argument (proba) as data frame of rough probabilities from SDMs for all species in columns in the considered sites in rows.

```r
proba <- ecospat.testData[,73:92]
```

Input data for the second argument (sr) as data frame with richness value in the first column and sites.

```
sr <- as.data.frame(rowSums(proba))
```

3.4 SESAM framework with *ecospat.SESAM.prr()*

```
prr<-ecospat.SESAM.prr(proba, sr)
head(prr)[,1:4]
```

```
##   glm_Agrostis_capillaris glm_Leontodon_hispidus_sl glm_Dactylis_glomerata
## 1                       0                         1                      1
## 2                       1                         0                      1
## 3                       1                         0                      1
## 4                       1                         0                      1
## 5                       1                         0                      1
## 6                       1                         0                      1
##   glm_Trifolium_repens_sstr
## 1                         0
## 2                         1
## 3                         1
## 4                         1
## 5                         1
## 6                         1
```

4 Post-Modelling

4.1 Spatial Predictions of species assamblages

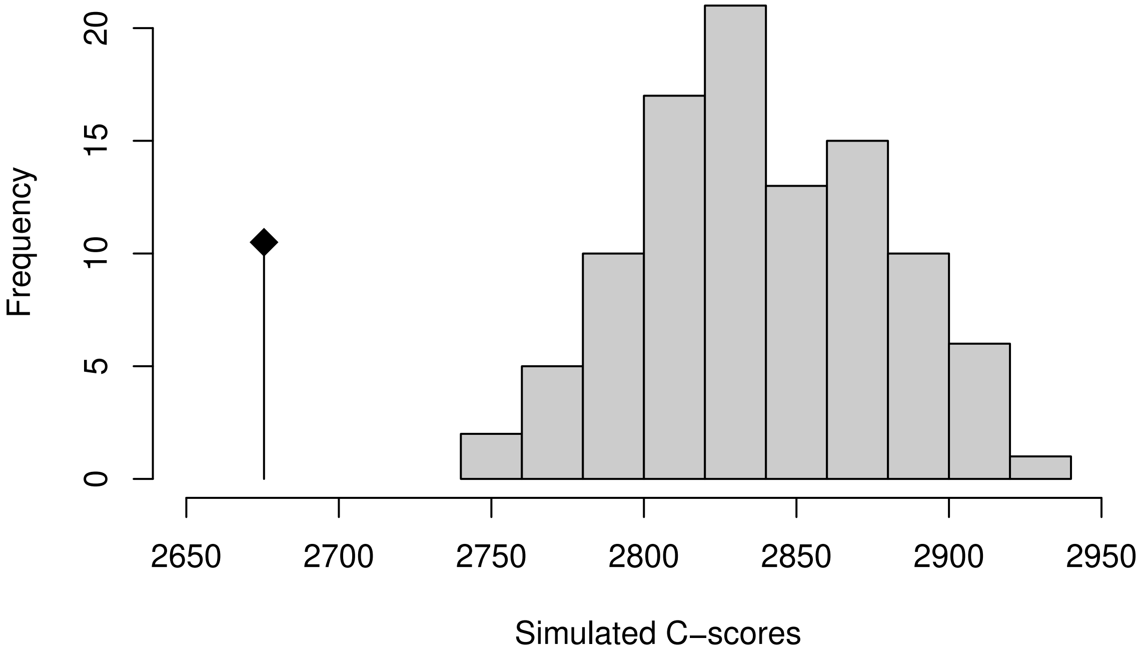
4.1.1 Co-occurrence analysis & Environmentally Constrained Null Models

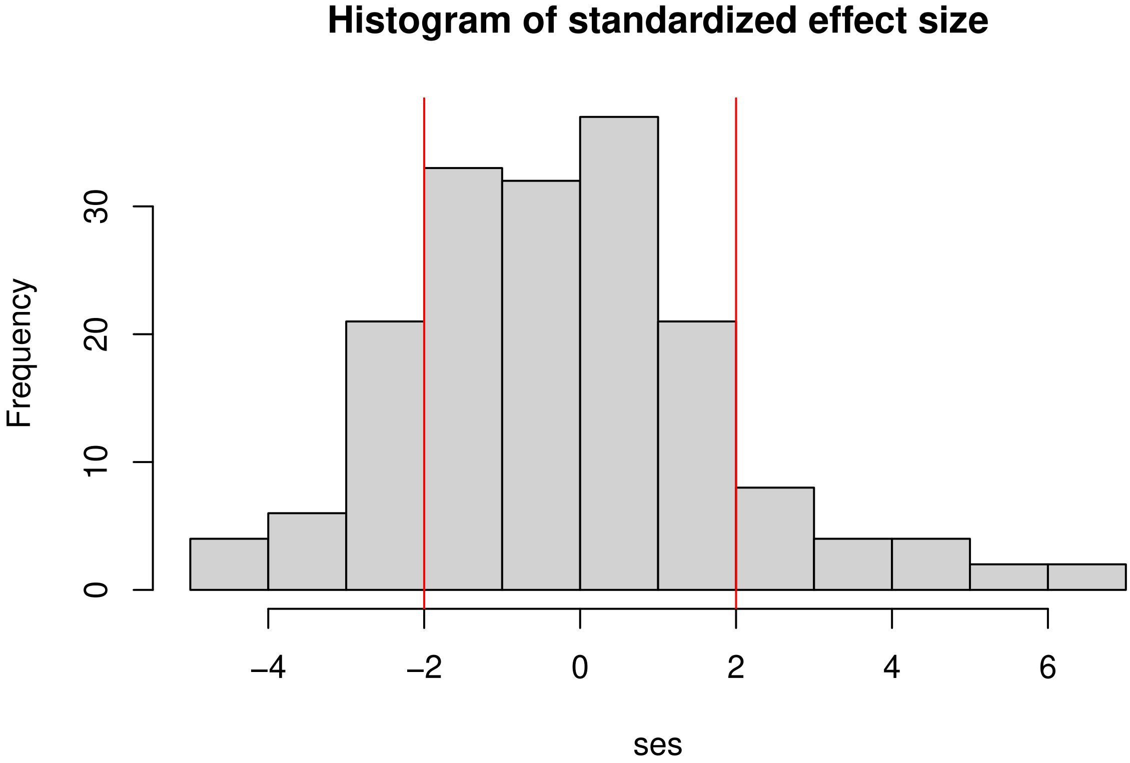
Input data as a matrix of plots (rows) x species (columns). Input matrices should have column names (species names) and row names (sampling plots).

```
presence<-ecospat.testData[c(53,62,58,70,61,66,65,71,69,43,63,56,68,57,55,60,54,67,59,64)]
pred<-ecospat.testData[c(73:92)]
```

Define the number of permutations. It is recommended to use at least 10000 permutations for the test. As an example we used nperm = 100, to reduce the computational time. Then Define the outpath. Then we can run Run the function *ecospat.cons_Cscore*. The function tests for non-random patterns of species co-occurrence in a presence-absence matrix. It calculates the C-score index for the whole community and for each species pair. An environmental constraint is applied during the generation of the null communities.

```
nbpermut <- 100
outpath <- getwd()
ecospat.cons_Cscore(presence, pred, nbpermut, outpath)
```





```
## $ObsCscoreTot
## [1] 2675.468
##
## $SimCscoreTot
## [1] 2839.598
##
## $PVal.less
## [1] 0.00990099
##
## $PVal.greater
## [1] 1
##
## $SES.Tot
```

```
## [1] -4.064741
```

The function returns - the C-score index for the observed community (ObsCscoreTot), - the mean of C-score for the simulated communities (SimCscoreTot), - the p.values (PVal.less and PVal.greater) to evaluate the significance of the difference between the former two indices. - the standardized effect size for the whole community (SES.Tot). A SES that is greater than 2 or less than -2 is statistically significant with a tail probability of less than 0.05 (Gotelli & McCabe 2002 - Ecology). If a community is structured by competition, we would expect the C-score to be large relative to a randomly assembled community (positive SES). In this case the observed C-score is significantly lower than expected by chance, this meaning that the community is dominate by positive interactions (aggregated pattern).

A table is saved in the path specified where the same metrics are calculated for each species pair (only the table with species pairs with significant p.values is saved).