

# Vignette ecospat package

## Contents

<b>1</b>	<b>Load data</b>	<b>1</b>
<b>2</b>	<b>Pre-Modelling Analysis</b>	<b>5</b>
2.1	Spatial Auto-correlation . . . . .	5
2.2	Predictor Variable Selection . . . . .	6
2.3	Climate Analogy Tools . . . . .	6
2.4	Phylogenetic Diversity Measures . . . . .	8
2.5	Niche Quantification and Comparison with Ordination techniques . . . . .	10
2.6	Biotic Interactions . . . . .	16
2.7	Data Preparation . . . . .	19
<b>3</b>	<b>Core Niche Modelling</b>	<b>22</b>
3.1	Model Evaluation . . . . .	22
3.2	Spatial Predictions and Projections . . . . .	23
3.3	Spatial prediction of communities . . . . .	27
3.4	SESAM framework with <i>ecospat.SESAM.prr()</i> . . . . .	28
<b>4</b>	<b>Post-Modelling</b>	<b>28</b>
4.1	Spatial Predictions of species assamblages . . . . .	28

=====  
**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

```
library(ecospat)
```

```
## Loading required package: ade4

## Loading required package: ape

## Loading required package: gbm

## Loaded gbm 2.1.5

## Loading required package: sp

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

```
citation("ecospat")
```

```
##
## To cite package 'ecospat' in publications use:
##
##   Olivier Broennimann, Valeria Di Cola and Antoine Guisan (2020).
##   ecospat: Spatial Ecology Miscellaneous Methods. R package version
##   3.1.
##   http://www.unil.ch/ecospat/home/menuguid/ecospat-resources/tools.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 = {2020},
##     note = {R package version 3.1},
##     url = {http://www.unil.ch/ecospat/home/menuguid/ecospat-resources/tools.html},
##   }
```

### 1.0.1 Test data for the ecospat library

```
ecospat.testData()
```

```
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_laevipipes"        "Doronicum_grandiflorum"
## [23] "Galium_album"               "Galium_anisophyllum"
## [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()*

```

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()*

```

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()*

```
fpath <- system.file("extdata", "ecospat.testTree.tre", package="ecospat")
fpath
```

```
## [1] "C:/Users/obroenni/AppData/Local/Temp/RtmpyqAPOH/Rinst19b846562869/ecospat/extdata/ecospat.tre"
```

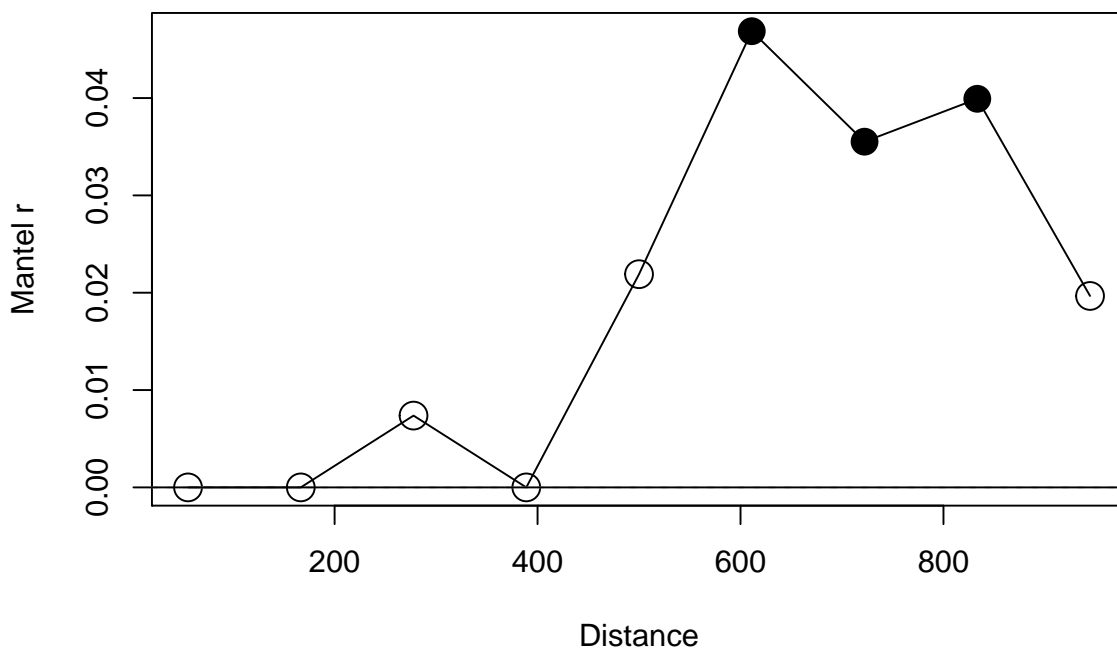
```
tree<-read.tree(fpath)
tree$tip.label
```

```
## [1] "Rumex_acetosa"          "Polygonum_bistorta"
## [3] "Polygonum_viviparum"   "Rumex_crispus"
## [5] "Cerastium_latifolium"  "Silene_aucaulis"
## [7] "Gypsophila_repens"     "Vaccinium_gaultherioides"
## [9] "Soldanella_alpina"     "Cruciata_laevipes"
## [11] "Galium_album"          "Galium_anisophyllum"
## [13] "Galium_megalospermum"  "Gentiana_verna"
## [15] "Gentiana_bavarica"     "Gentiana_purpurea"
## [17] "Gentiana_lutea"        "Bartsia_alpina"
## [19] "Rhinanthus_alectorolophus" "Prunella_grandiflora"
## [21] "Acinos_alpinus"        "Plantago_alpina"
## [23] "Plantago_lanceolata"   "Veronica_officinalis"
## [25] "Veronica_aphylla"      "Veronica_alpina"
## [27] "Veronica_chamaedrys"   "Veronica_persica"
## [29] "Globularia_cordifolia" "Globularia_nudicaulis"
## [31] "Myosotis_alpestris"    "Myosotis_arvensis"
## [33] "Aposeris_foetida"      "Centaurea_montana"
## [35] "Hieracium_lactucella"  "Leontodon_helveticus"
## [37] "Leontodon_autumnalis"  "Hypochaeris_radicata"
## [39] "Achillea_atrata"       "Achillea_millefolium"
## [41] "Homogyne_alpina"       "Senecio_doronicum"
## [43] "Adenostyles_glabra"    "Arnica_montana"
## [45] "Aster_bellidiasstrum"  "Bellis_perennis"
## [47] "Doronicum_grandiflorum" "Phyteuma_orbiculare"
## [49] "Phyteuma_spicatum"     "Campanula_rotundifolia"
```

Plot tree

```
plot(tree, cex=0.6)
```





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()*

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

```
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
```

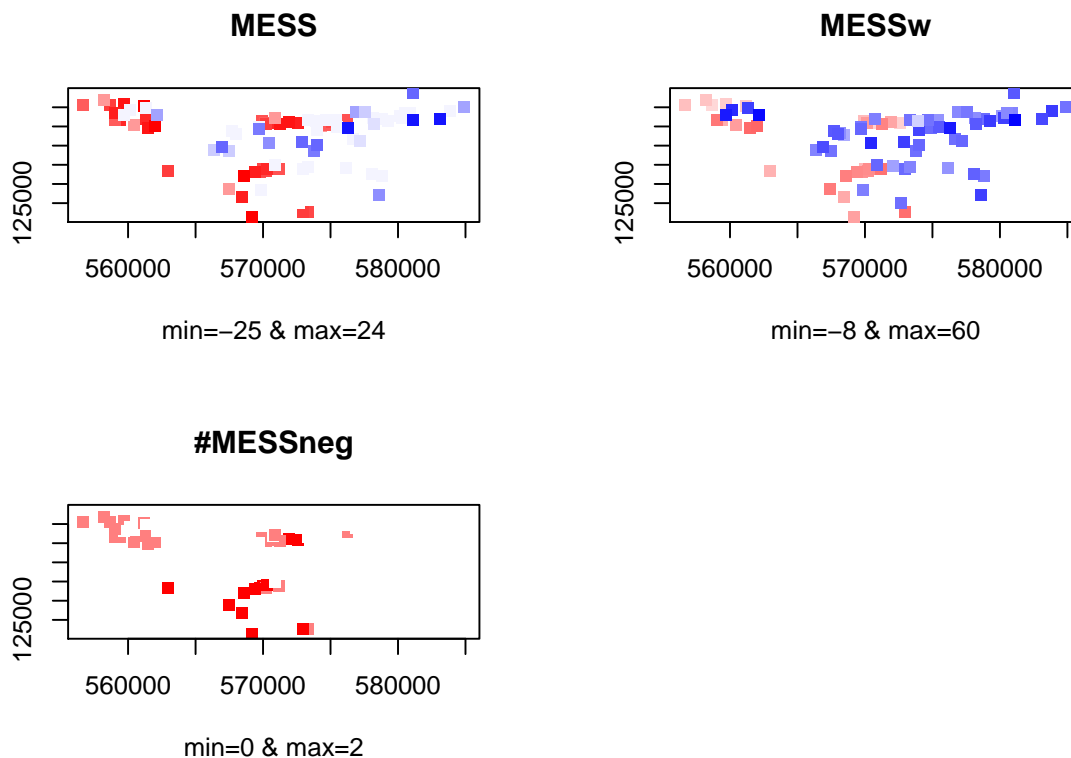
### 2.3.2 Extrapolation detection, creating a MESS object with *ecospat.mess()*

```
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")
```

### 2.3.3 Plot MESS with *ecospat.plot.mess()*

```
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

```
fpath <- system.file("extdata", "ecospat.testTree.tre", package="ecospat")
tree <- read.tree(fpath)
data <- ecospat.testData[9:52]
```

### 2.4.1 Calculate Phylogenetic Diversity Measures *ecospat.calculate.pd*

```
pd<- ecospat.calculate.pd(tree, data, method = "spanning", type = "species", root = TRUE, average =
```

```
## Progress (. = 100 pixels calculated):
## ... [300]
## All 300 pixels done.
```

```
pd
```

```
## [1] 6.9782188 6.7981743 4.9964700 4.9964700 4.9964700 29.8820547
## [7] 58.7451752 6.5223035 30.6152478 1.5258335 0.0000000 44.3661803
## [13] 38.4155607 6.5223035 24.0929443 78.1607950 38.4155607 29.0894143
## [19] 29.0894143 89.9839758 27.4135569 40.2827035 1.5258335 56.7686202
## [25] 18.9535475 34.8871800 0.0000000 1.5258335 39.9291325 48.5997861
```



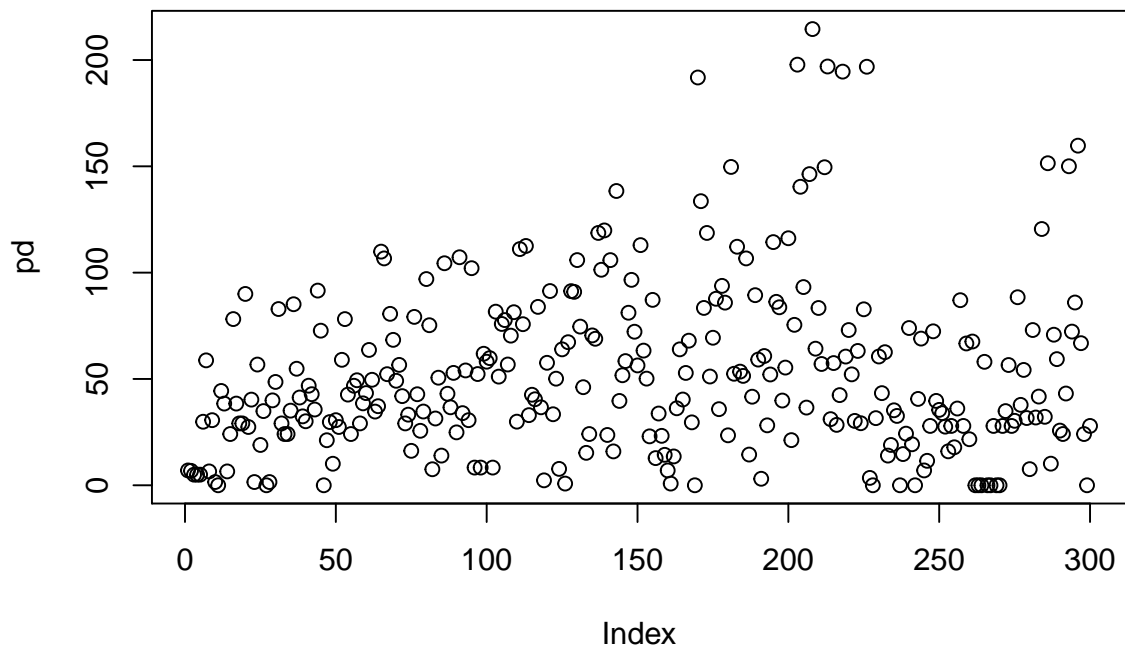
```

## [31] 82.8763723 29.0894143 24.0929443 24.0929443 35.0949481 85.1406422
## [37] 54.7974724 41.2817284 32.4100269 30.0984781 46.8247511 42.8358475
## [43] 35.6223697 91.5539224 72.7022527 0.0000000 21.1862293 29.7320308
## [49] 10.1187868 30.6152478 27.4135569 59.0015345 78.1536692 42.6423378
## [55] 24.0929443 46.8050070 49.3924266 29.0894143 38.5290848 43.3611373
## [61] 63.6397674 49.6097169 34.6522309 37.1871282 109.8813371 106.6971561
## [67] 52.2512132 80.6221671 68.3867818 49.1362998 56.6138690 41.9283257
## [73] 29.0894143 33.2026673 16.1897593 79.1938213 42.8115427 25.6187778
## [79] 34.6805724 96.9902366 75.2672695 7.5313673 31.4078882 50.5865673
## [85] 13.9570775 104.4121025 43.0464918 36.6693230 52.8590823 24.8855847
## [91] 107.2302322 33.9358604 54.0048319 30.6152478 102.0983385 8.3170826
## [97] 52.3071062 8.3170826 61.8562896 58.1179346 59.7939424 8.3170826
## [103] 81.6495398 51.1054635 75.8701970 77.6947419 56.7929250 70.3693202
## [109] 81.3965205 29.9118877 111.0790432 75.7518798 112.5482496 32.9763735
## [115] 42.5644761 40.4507005 83.8955419 36.6693230 2.3184739 57.5978451
## [121] 91.3453370 33.3983912 50.1351419 7.7084002 63.9227817 0.7926404
## [127] 67.2813325 91.2965996 90.9578739 105.9024741 74.6128871 46.1321553
## [133] 15.2479619 24.0929443 70.4802708 68.8949899 118.6657550 101.3545260
## [139] 119.8539056 23.6602184 105.8968281 15.9336325 138.4059855 39.6674173
## [145] 51.7391372 58.4119283 81.1388699 96.6048825 72.2156025 56.3601992
## [151] 112.9489963 63.3258805 50.1594468 23.0021994 87.1886965 12.7714946
## [157] 33.7421666 23.2537702 14.3226164 6.9752071 0.7926404 13.5641350
## [163] 36.2007616 63.9227817 40.3310946 52.8264129 67.9956878 29.5843437
## [169] 0.0000000 191.7818606 133.6077875 83.3977825 118.6711630 51.1512871
## [175] 69.3838811 87.7066616 35.8005270 93.7797077 85.8984840 23.4933413
## [181] 149.7094684 52.4451847 112.1873673 53.4479612 51.4341108 106.6959500
## [187] 14.4361405 41.6547546 89.4018733 59.1068292 3.0516670 60.7852739
## [193] 28.1850877 52.1002690 114.3651475 86.2640717 83.7092232 39.8499777
## [199] 55.3514065 116.1795597 21.2346203 75.4593878 197.8157358 140.3806968
## [205] 93.2192350 36.5337815 146.3370747 214.5450205 64.2439145 83.3740177
## [211] 57.0440643 149.5697614 196.9415036 31.0984631 57.4769230 28.4014469
## [217] 42.3978747 194.5384819 60.5204195 73.0060715 52.1628582 30.2801165
## [223] 63.1752097 29.1789484 82.7662787 196.8309769 3.4666557 0.0000000
## [229] 31.5688084 60.5650008 43.3334929 62.5952411 13.9570775 18.9495667
## [235] 35.2646601 32.6155790 0.0000000 14.6693623 24.2745827 73.9480832
## [241] 19.2825866 0.0000000 40.6115985 68.9862341 6.9782188 11.5030881
## [247] 27.9105497 72.4020225 39.6781995 35.4596364 33.9160835 27.5735165
## [253] 15.9619740 27.9105497 17.8628493 36.0936777 87.0440848 27.9105497
## [259] 66.6907987 21.6475811 67.5969904 0.0000000 0.0000000 0.0000000
## [265] 58.0542370 0.0000000 0.0000000 27.9105497 0.0000000 0.0000000
## [271] 27.9105497 34.8887684 56.5556633 27.9105497 30.3097595 88.4296666
## [277] 37.8150727 54.2397810 31.6243116 7.5799087 73.0136833 31.8638035
## [283] 41.7172212 120.5228857 32.2001243 151.4545228 10.1544492 70.8133537
## [289] 59.3255687 25.7211220 24.1115267 43.1500941 150.0299191 72.2758570
## [295] 85.9498096 159.7242106 66.8328159 24.0929443 0.0000000 27.9105497

```

```
plot(pd)
```

### 2.4.1.1 Plot the results (correlation of phylogenetic diversity with species richness)



## 2.5 Niche Quantification and Comparison with Ordination techniques

Loading test data for the niche dynamics analysis in the invaded range

```
inv <- ecospat.testNiche.inv
```

Loading test data for the niche dynamics analysis in the native range

```
nat <- ecospat.testNiche.nat
```

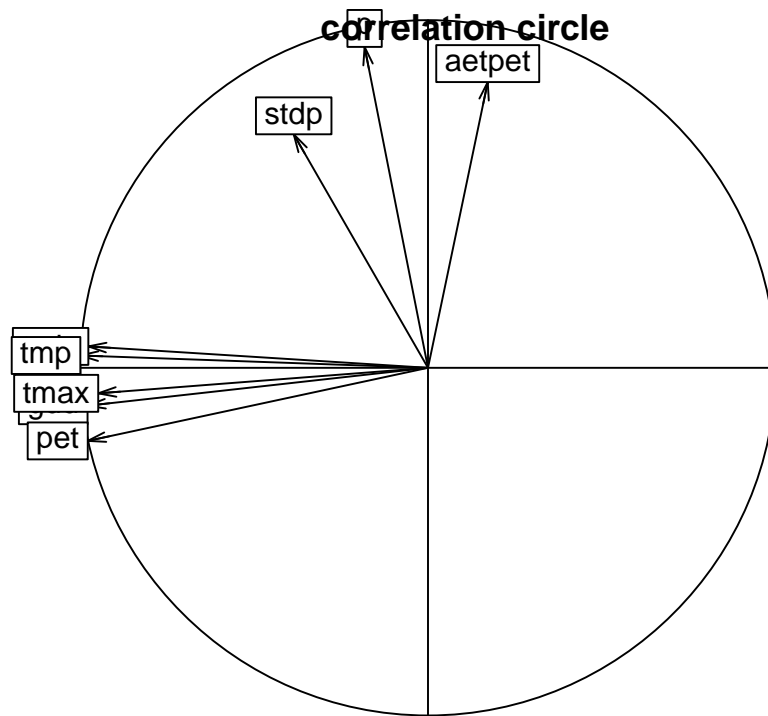
### 2.5.1 PCA-ENVIRONMENT

The 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)

```
pca.env <- dudi.pca(rbind(nat,inv)[,3:10],scannf=F,nf=2)
```

Plot Variables Contribution with *ecospat.plot.contrib()*

```
ecospat.plot.contrib(contrib=pca.env$co, eigen=pca.env$eig)
```



axis1 = 61.14 % axis2 = 25.09 %

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

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 <- suprow(pca.env,nat[which(nat[,11]==1),3:10])$li

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

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

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

## 2.5.2 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)
```

For a species in the invaded range (Australia)

```
# 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)
```

### 2.5.3 Calculate Niche Overlap with *ecospat.niche.overlap()*

```
# 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%.

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

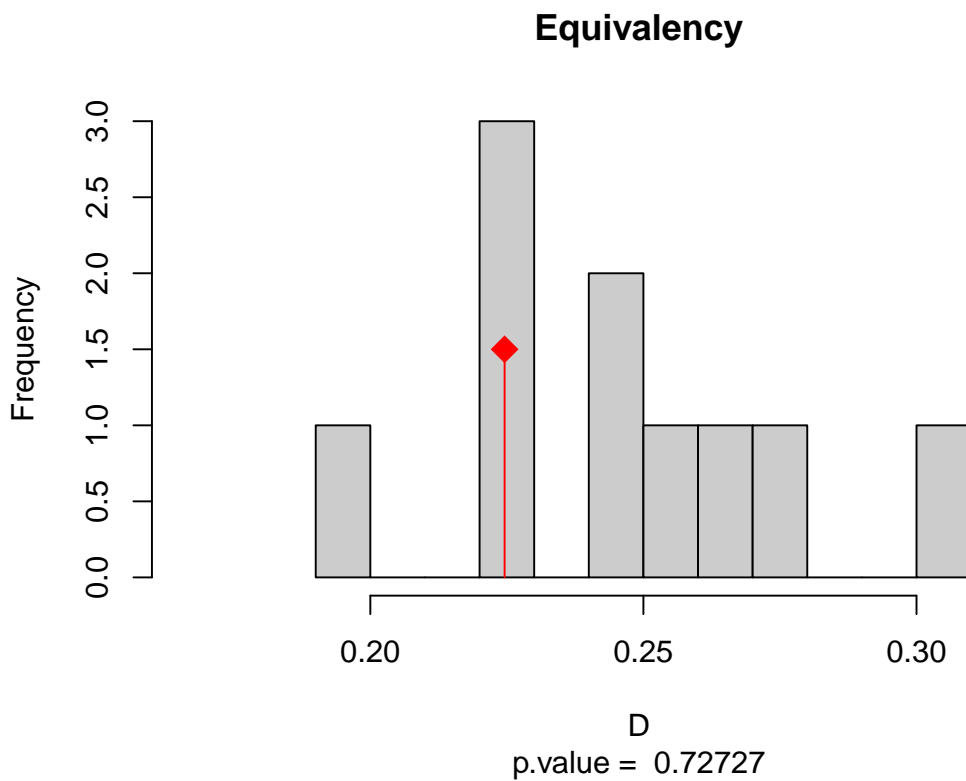
It is recommended to use at least 1000 replications for the equivalency test. As an example we used `rep = 10`, to reduce the computational time.

```
eq.test <- ecospat.niche.equivalency.test(grid.clim.nat, grid.clim.inv,
                                         rep=10, alternative = "greater")
```

Niche equivalency test H1: the observed overlap between the native and invaded niche is higher than if the two niches are randomized

Plot Equivalency test

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



#### 2.5.5 Perform the Niche Similarity Test with `ecospat.niche.similarity.test()`

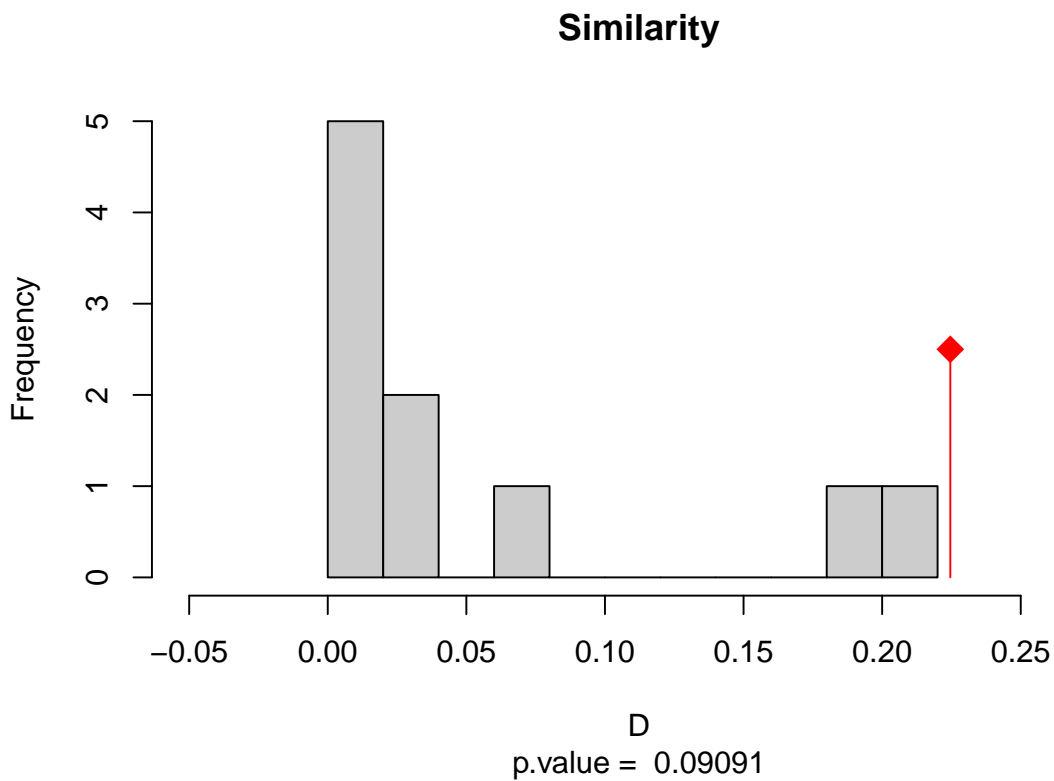
Shifts randomly on niche (here the invasive niche) in the study area. It is recommended to use at least 1000 replications for the similarity test. As an example we used `rep = 10`, to reduce the computational time.

```
sim.test <- ecospat.niche.similarity.test(grid.clim.nat, grid.clim.inv,
                                         rep=10, alternative = "greater",
                                         rand.type=2)
```

Niche similarity test H1: the observed overlap between the native and invaded is higher than randomly shifted invasive niches in the invaded study area

Plot Similarity test

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



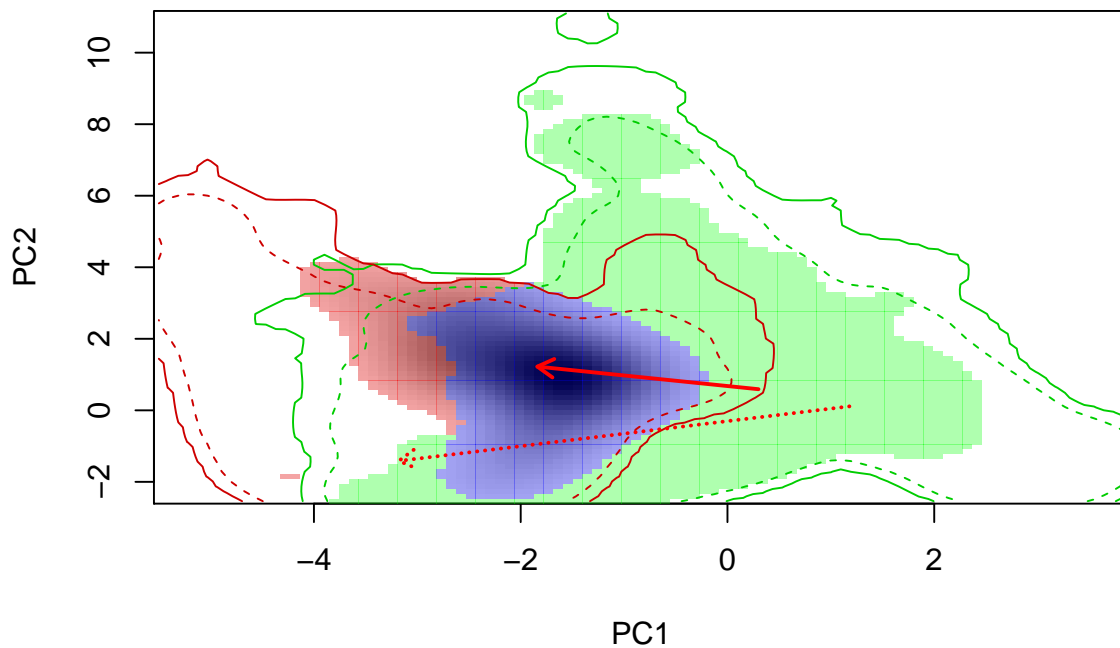
#### 2.5.6 Delimiting niche categories and quantifying niche dynamics in analogue climates with *ecospat.niche.dyn.index()*

```
niche.dyn <- ecospat.niche.dyn.index (grid.clim.nat, grid.clim.inv, intersection = 0.1)
```

##### 2.5.6.1 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)
```

## Niche Overlap



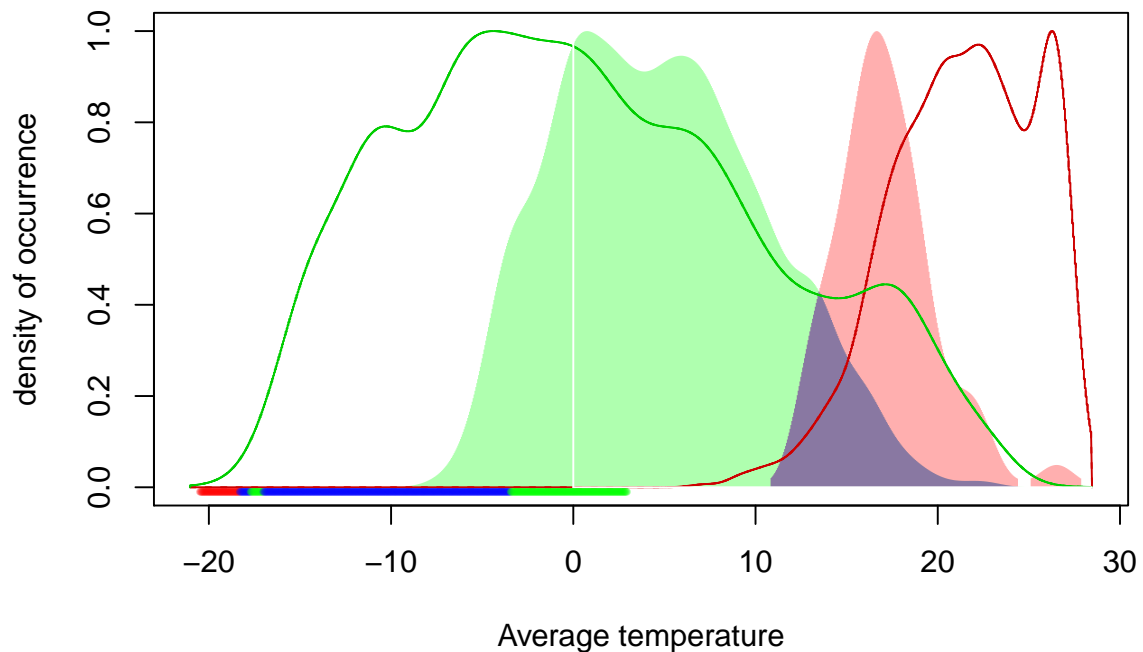
```
# gridding the native niche
grid.clim.t.nat <- ecospat.grid.clim.dyn(glob=as.data.frame(rbind(nat,inv)[,10]),
                                       glob1=as.data.frame(nat[,10]),
                                       sp=as.data.frame(nat[which(nat[,11]==1),10]),
                                       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,
                               intersection=0.1)

ecospat.plot.niche.dyn(grid.clim.t.nat, grid.clim.t.inv, quant=0,
                      interest=2, title= "Niche Overlap",
                      name.axis1="Average temperature")
```

### 2.5.6.2 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()`

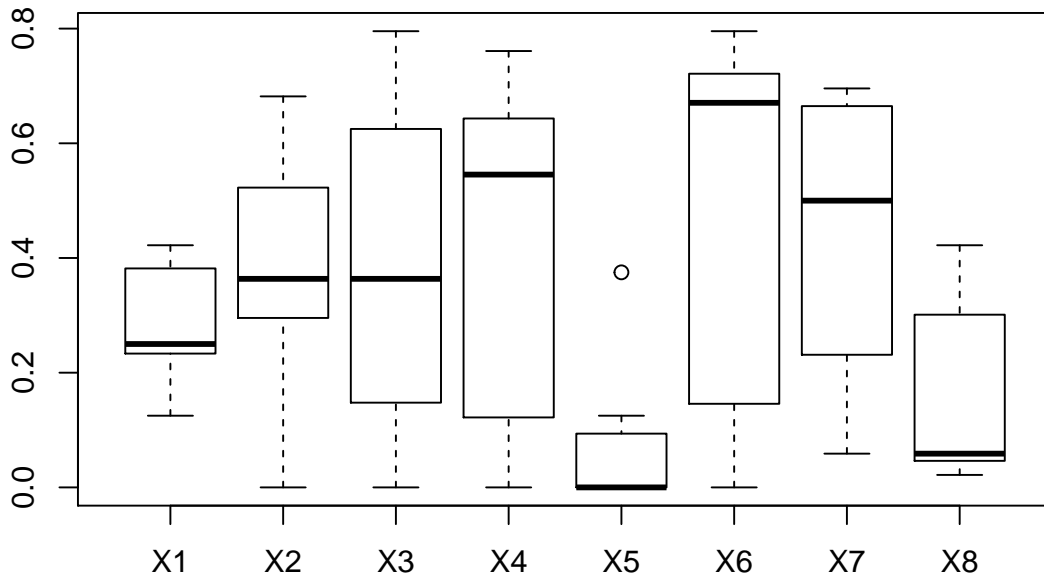
```
data <- ecospat.testData[c(9:16,54:57)]
```

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  $N(S1 \text{ intersects } S2)$  is the number of times species S1 and S2 co-occur, while  $\text{Min}(NS1, NS2)$  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      0.2500000
## Arnica_montana             0.3636364      1.0000000      0.36363636
## Aster_bellidiastrum        0.2500000      0.3636364      1.0000000
## Bartsia_alpina             0.2222222      0.5454545      0.59090909
## Bromus_erectus_sstr        0.1250000      0.0000000      0.0000000
## 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          0.2222222          0.1250          0.24444444
## Arnica_montana            0.5454545          0.0000          0.68181818
## Aster_bellidiastrum       0.59090909         0.0000          0.79545455
## Bartsia_alpina            1.0000000         0.0000          0.76086957
## Bromus_erectus_sstr       0.0000000         1.0000          0.0000000
## Campanula_scheuchzeri     0.76086957         0.0000          1.0000000
## Carex sempervirens        0.69565217         0.0625          0.67058824
## Cynosurus_cristatus       0.02173913         0.3750          0.04705882
##           Carex sempervirens  Cynosurus_cristatus
## Aposeris_foetida            0.4000000          0.4222222
## Arnica_montana              0.5000000          0.22727273
## Aster_bellidiastrum         0.65909091          0.04545455
## Bartsia_alpina              0.69565217          0.02173913
## Bromus_erectus_sstr         0.06250000          0.3750000
## Campanula_scheuchzeri       0.67058824          0.04705882
## Carex sempervirens          1.0000000          0.05882353
## Cynosurus_cristatus         0.05882353          1.0000000

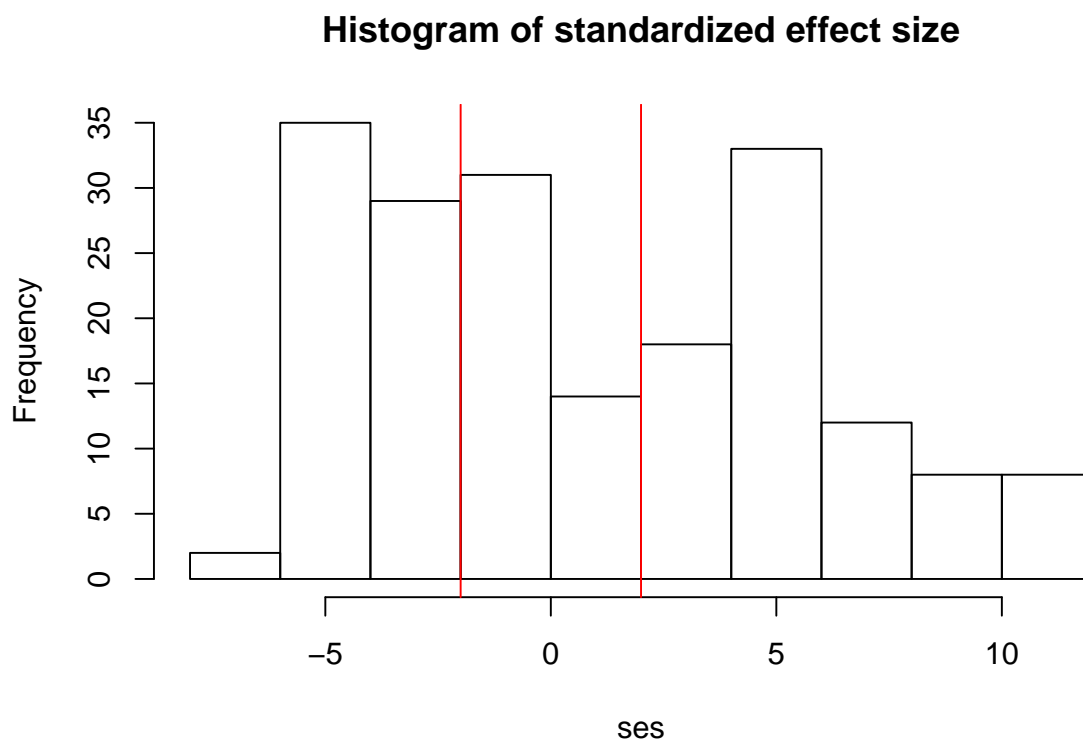
```

## 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 recommended to use at least 10000 permutations 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] 2466.121
##
## $PVal.less
## [1] 1
##
## $PVal.greater
## [1] 0.00990099
##
## $SES.Tot
## [1] 52.07788
```

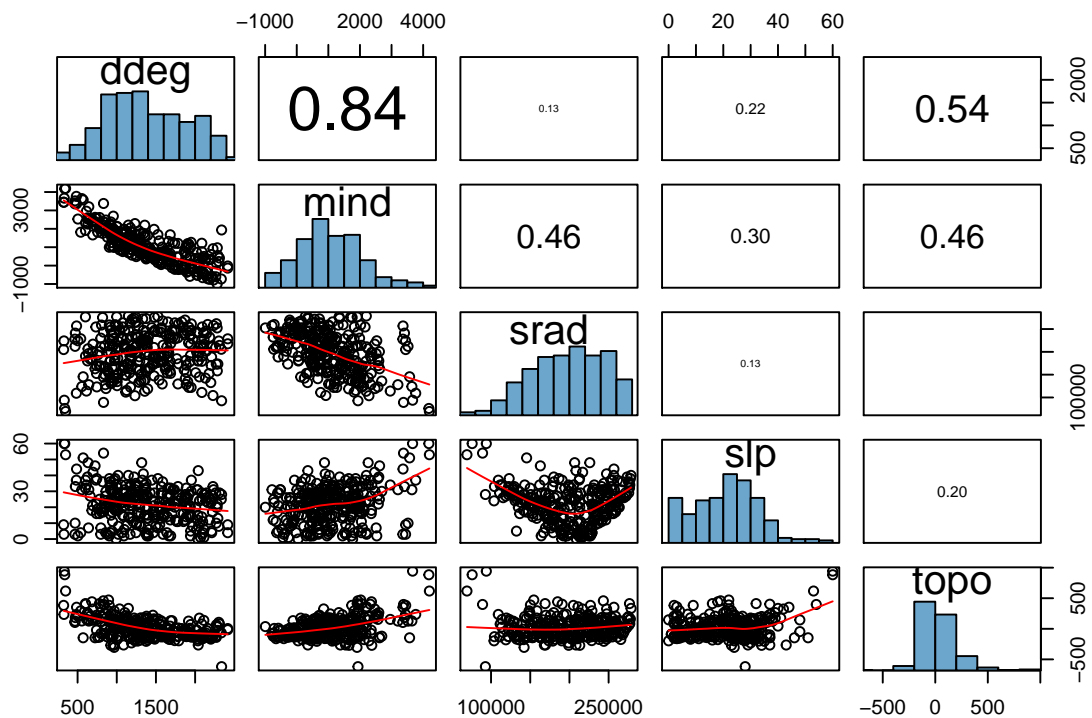
The function returns the C-score index for the observed community (`ObsCscoreTot`), p.value (`PValTot`) and standardized effect size (`SES.Tot`). It saves also a table in the working directory where the same

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

```
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   119  111
## 2    26   86
## 3   175  118
```

```

## 4 279 51
## 5 169 3
## 6 155 17
## 7 243 211
## 8 245 240
## 9 214 249
## 10 281 223
## 11 116 264
## 12 37 210
## 13 246 296
## 14 171 266
## 15 270 265
## 16 271 156
## 17 236 67
## 18 45 290
## 19 23 220
## 20 79 4
## 21 252 222
## 22 219 34
## 23 33 294
## 24 274 231
## 25 168 288
## 26 241 206
## 27 238 152
## 28 11 275
## 29 123 55
## 30 27 185
## 31 140 15
## 32 75 188
## 33 178 186
##
## $cal
## ycal ycal
## 1 126 NA
## 2 158 NA
## 3 NA NA
## 4 68 NA
## 5 42 NA
## 6 NA 161
## 7 NA NA
## 8 181 292
## 9 232 114
## 10 71 36
## 11 278 204
## 12 56 5
## 13 185 181
## 14 49 115
## 15 121 22
## 16 268 250
## 17 235 291
## 18 293 168
## 19 133 205
## 20 275 279
## 21 234 84
## 22 150 254
## 23 134 166
## 24 201 139
## 25 189 193

```

```
## 26 15 30
## 27 95 100
## 28 283 228
## 29 276 242
## 30 100 157
## 31 186 225
## 32 198 56
## 33 85 251
## 34 154 106
## 35 263 256
## 36 180 116
## 37 203 23
## 38 212 24
## 39 300 145
## 40 43 286
## 41 258 244
## 42 230 184
## 43 196 261
## 44 188 253
## 45 289 182
## 46 114 20
## 47 224 248
## 48 233 203
## 49 272 200
## 50 22 199
## 51 260 274
## 52 256 260
## 53 217 246
## 54 31 8
## 55 115 295
## 56 84 255
## 57 21 180
## 58 57 237
## 59 244 11
## 60 2 299
## 61 44 247
## 62 267 278
## 63 147 178
## 64 120 150
## 65 229 192
## 66 192 297
## 67 291 221
## 68 239 14
## 69 113 2
## 70 273 110
## 71 18 269
## 72 255 293
## 73 30 94
## 74 55 16
## 75 24 53
## 76 177 259
## 77 262 85
```

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

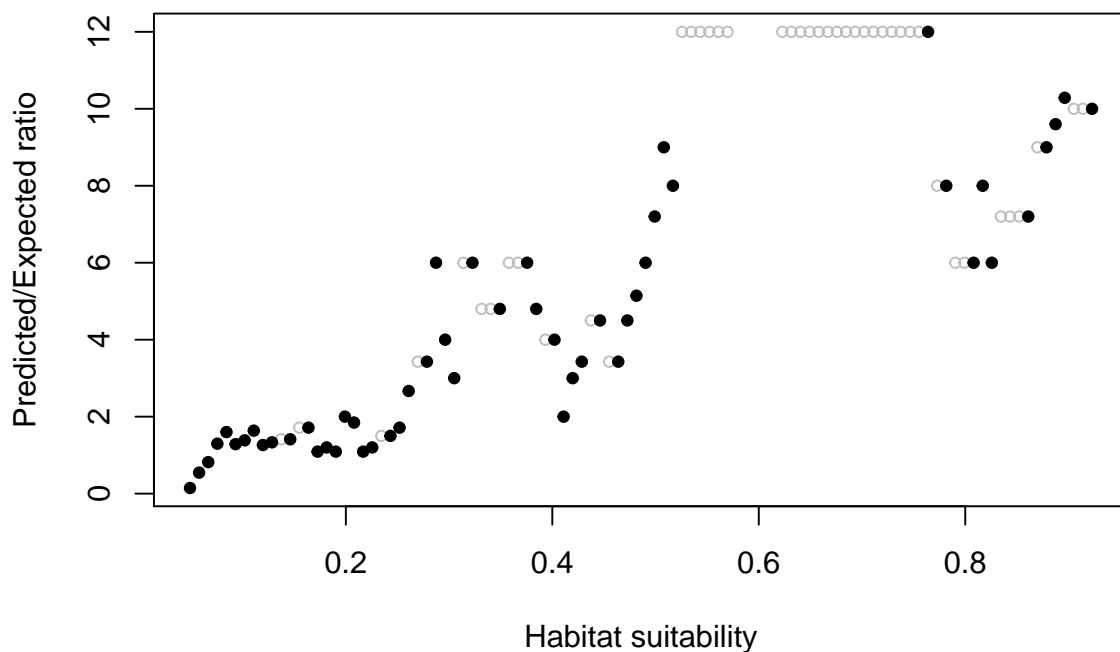
```
fit <- ecospat.testData$glm_Saxifraga_oppositifolia
```

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)$Spearman.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 3.4.6 loaded.
##
## Type browseVignettes(package='biomod2') to access directly biomod2 vignettes.
```

```
# 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
```

```
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 <- 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  
## > No pseudo absences selection !  
## ! No data has been set aside for modeling evaluation  
## ----- Done -----
```

```
myBiomodOption <- Print_Default_ModelingOptions()
```

```
##  
## Defaut modeling options. copy, change what you want paste it as arg to BIOMOD_ModelingOptions  
##  
## ----- 'BIOMOD.Model.Options' -----  
##  
##  
## GLM = list( type = 'quadratic',  
              interaction.level = 0,  
              myFormula = NULL,  
              test = 'AIC',  
              family = binomial(link = 'logit'),  
              mustart = 0.5,  
              control = glm.control(epsilon = 1e-08, maxit = 50, trace = FALSE  
## ) ),  
##  
##  
## GBM = list( distribution = 'bernoulli',  
              n.trees = 2500,  
              interaction.depth = 7,  
              n.minobsinnode = 5,  
              shrinkage = 0.001,  
              bag.fraction = 0.5,  
              train.fraction = 1,  
              cv.folds = 3,  
              keep.data = FALSE,  
              verbose = FALSE,  
              perf.method = 'cv',  
              n.cores = 1),  
##  
## GAM = list( algo = 'GAM_mgcv',  
              type = 's_smoother',  
              k = -1,  
              interaction.level = 0,  
              myFormula = NULL,  
              family = binomial(link = 'logit'),  
              method = 'GCV.Cp',  
              optimizer = c('outer','newton'),  
              select = FALSE,  
              knots = NULL,
```



```

##           paraPen = NULL,
##           control = list(nthreads = 1, irls.reg = 0, epsilon = 1e-07
## , maxit = 200, trace = FALSE, mgcv.tol = 1e-07, mgcv.half = 15
## , rank.tol = 1.49011611938477e-08
## , nlm = list(ndigit=7, gradtol=1e-06, stepmax=2, steptol=1e-04, iterlim=200, check.analyticals=0)
## , optim = list(factr=1e+07)
## , newton = list(conv.tol=1e-06, maxNstep=5, maxSstep=2, maxHalf=30, use.svd=0)
## , outerPIsteps = 0, idLinksBases = TRUE, scalePenalty = TRUE, efs.lspmax = 15
## , efs.tol = 0.1, keepData = FALSE, scale.est = fletcher, edge.correct = FALSE
## ) ),
##
##
## CTA = list( method = 'class',
##           parms = 'default',
##           cost = NULL,
##           control = list(xval = 5, minbucket = 5, minsplit = 5, cp = 0.001
## , maxdepth = 25) ),
##
##
## ANN = list( NbcV = 5,
##           size = NULL,
##           decay = NULL,
##           rang = 0.1,
##           maxit = 200),
##
## SRE = list( quant = 0.025),
##
## FDA = list( method = 'mars',
##           add_args = NULL),
##
## MARS = list( type = 'simple',
##           interaction.level = 0,
##           myFormula = NULL,
##           nk = NULL,
##           penalty = 2,
##           thresh = 0.001,
##           nprune = NULL,
##           pmethod = 'backward'),
##
## RF = list( do.classif = TRUE,
##           ntree = 500,
##           mtry = 'default',
##           nodesize = 5,
##           maxnodes = NULL),
##
## MAXENT.Phillips = list( path_to_maxent.jar = 'C:/Users/obroenni/AppData/Local/Temp/RtmpyqAPOH/Rbu
##           memory_allocated = 512,
##           background_data_dir = 'default',
##           maximumbackground = 'default',
##           maximumiterations = 200,
##           visible = FALSE,
##           linear = TRUE,
##           quadratic = TRUE,
##           product = TRUE,
##           threshold = TRUE,
##           hinge = TRUE,
##           lq2lqptthreshold = 80,
##           l2lqthreshold = 10,

```

```

##             hingethreshold = 15,
##             beta_threshold = -1,
##             beta_categorical = -1,
##             beta_lqp = -1,
##             beta_hinge = -1,
##             betamultiplier = 1,
##             defaultprevalence = 0.5),
##
## MAXENT.Phillips.2 =
##   list(
##     myFormula = NULL,
##     regmult = 1,
##     regfun = <function>
##   )
## )
## -----

```

```

myBiomodOption@GLM$test = 'none'
myBiomodOption@GBM$interaction.depth = 2

```

```

### Calibration of simple bivariate models

# remove invisible(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'),
  models.options=myBiomodOption,
  NbRunEval=1,
  DataSplit=70,
  weighting.score=c("AUC"),
  parallel=F)
)
)

```

```

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

```

```

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

```

```

##
## ----- Do Models Projections -----
##
##   ! 'do.stack' arg is always set as TRUE for data.frame/matrix dataset
## > Projecting ESM.BIOMOD.1_AllData_RUN2_GLM ...
## ----- Done -----
##
## ----- Do Models Projections -----
##
##   ! 'do.stack' arg is always set as TRUE for data.frame/matrix dataset
## > Projecting ESM.BIOMOD.2_AllData_RUN2_GLM ...
## ----- Done -----
##
## ----- Do Models Projections -----
##
##   ! 'do.stack' arg is always set as TRUE for data.frame/matrix dataset

```

```

## > Projecting ESM.BIOMOD.3_AllData_RUN2_GLM ...
## ===== Done =====
##
## ----- Do Models Projections -----
##
## ! 'do.stack' arg is always set as TRUE for data.frame/matrix dataset
## > Projecting ESM.BIOMOD.4_AllData_RUN2_GLM ...
## ===== Done =====
##
## ----- Do Models Projections -----
##
## ! 'do.stack' arg is always set as TRUE for data.frame/matrix dataset
## > Projecting ESM.BIOMOD.5_AllData_RUN2_GLM ...
## ===== Done =====
##
## ----- Do Models Projections -----
##
## ! 'do.stack' arg is always set as TRUE for data.frame/matrix dataset
## > Projecting ESM.BIOMOD.6_AllData_RUN2_GLM ...
## ===== Done =====
##
## ----- Do Models Projections -----
##
## ! 'do.stack' arg is always set as TRUE for data.frame/matrix dataset
## > Projecting ESM.BIOMOD.7_AllData_RUN2_GLM ...
## ===== Done =====
##
## ----- Do Models Projections -----
##
## ! 'do.stack' arg is always set as TRUE for data.frame/matrix dataset
## > Projecting ESM.BIOMOD.8_AllData_RUN2_GLM ...
## ===== Done =====
##
## ----- Do Models Projections -----
##
## ! 'do.stack' arg is always set as TRUE for data.frame/matrix dataset
## > Projecting ESM.BIOMOD.9_AllData_RUN2_GLM ...
## ===== Done =====
##
## ----- Do Models Projections -----
##
## ! 'do.stack' arg is always set as TRUE for data.frame/matrix dataset
## > Projecting ESM.BIOMOD.10_AllData_RUN2_GLM ...
## ===== Done =====

```

```

### 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.

```
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 assemblages

#### 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.

```
nbpermut <- 100
```

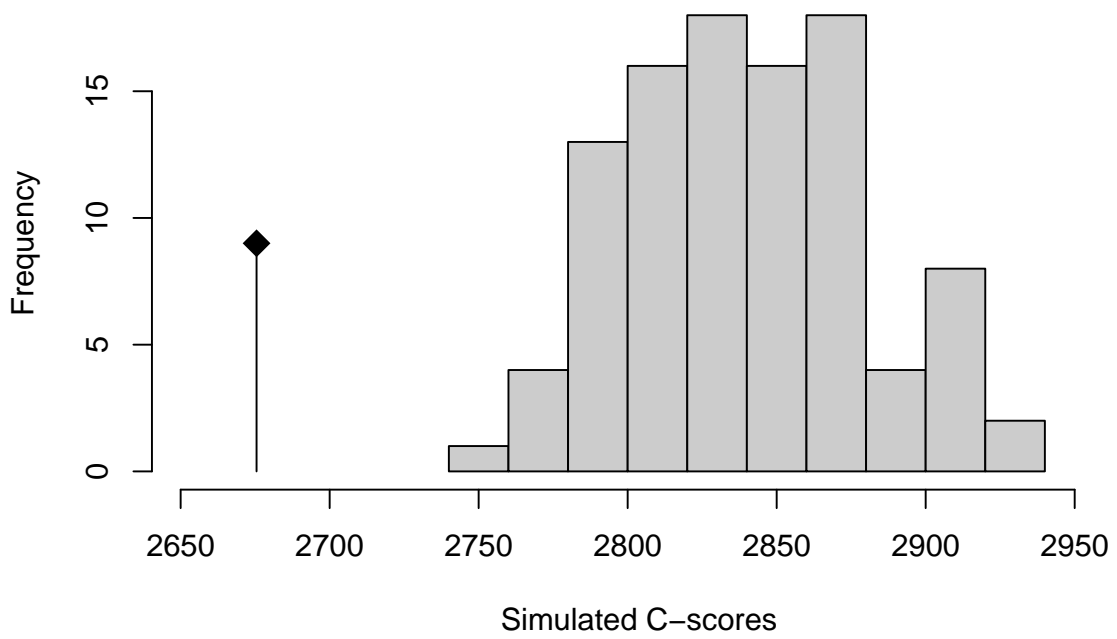
Define the outpath

```
outpath <- getwd()
```

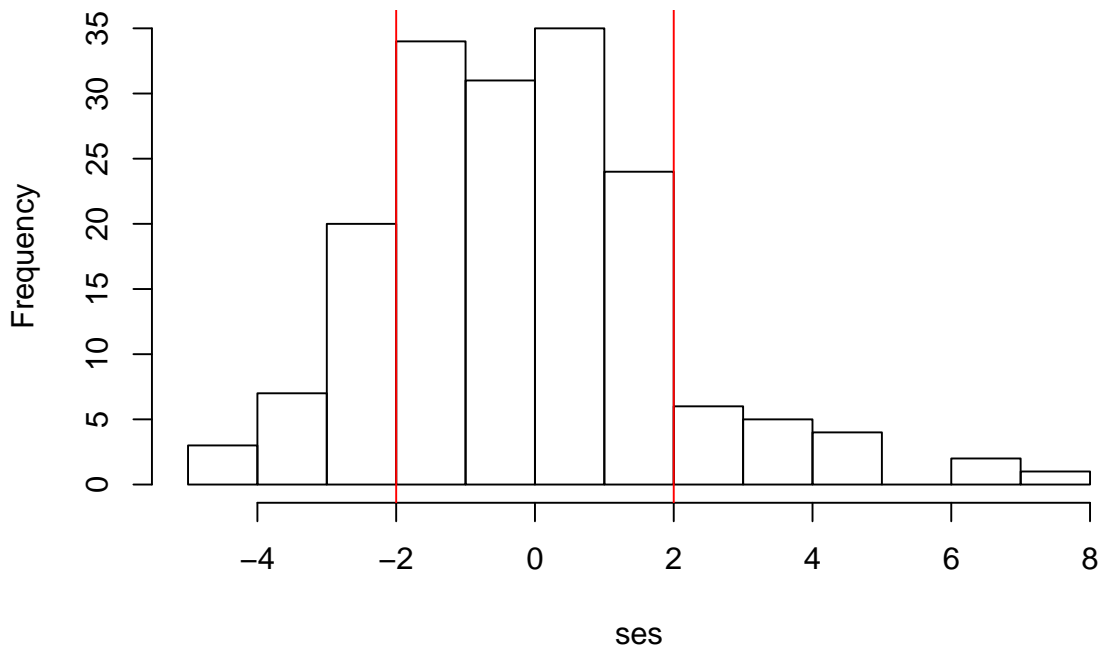
Run the function *ecospat.cons\_Cscore*

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.

```
ecospat.cons_Cscore(presence, pred, nbpermut, outpath)
```



**Histogram of standardized effect size**



```
## $ObsCscoreTot
## [1] 2675.468
##
## $SimCscoreTot
## [1] 2838.714
```

```
##  
## $PVal.less  
## [1] 0.00990099  
##  
## $PVal.greater  
## [1] 1  
##  
## $SES.Tot  
## [1] -4.090832
```

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 dominated 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).