

Code "ecospat" package

Valeria Di Cola, Olivier Broennimann & Antoine Guisan

July 22, 2014

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.

Contents

1 LOADING DATA EXAMPLE	3
1.1 Test data for the ecospat library	4
1.2 Test data for Niche Overlap Analysis	4
1.3 Test tree for the phylogenetic diversity analysis	5
2 Pre-Modelling Analysis.....	7
2.1 Spatial Auto-correlation.....	7
2.2 Predictor Variable Selection.....	7
2.3 MESS - Multivariate Environmental Similarity Surfaces	8
2.4 Phylogenetic Diversity Measures.....	11
2.5 Biotic Interactions	13
2.6 Niche Quantification with Ordination techniques.....	15
2.6.1 Species Occurrences Desaggregation	15
2.6.2 Sample Environmental Variables.....	17
2.6.3 Dynamic Occurrence Densities Grid.....	17
3 ECOLOGICAL NICHE MODELLING	28
3.1 Data Preparation for Modelling	28
3.1.1 Calibration and Evaluation Dataset	28
3.1.2 Sample Pseudo Absences	30
3.1.3 Make Data Frame	30
3.2 Model Calibration	34
3.2.1 Modelling Techniques - Cross-validation	34
3.2.2 Variables Importance	37
3.2.3 Model Averaging and Ensemble Forecasting	37
3.3 Model Evaluation.....	38
3.3.1 Presence-only Evaluation Indices- Boyce Index	38
3.3.2 Accuracy of Community Prediction	40
3.4 Spatial Predictions and Projections.....	41
3.4.1 Model Binarization.....	41
3.4.2 Implementing species dispersal into projections of species distribution models.....	45
4 COMPLEMENTARY ANALYSIS	46
4.1 Biotic Interactions	46
4.1.1 Co-occurrence pattern.....	46
4.1.2 Co-occurrence analysis Environmentally Constrained Null Models.....	47

1 LOADING DATA EXAMPLE

```
> library(ecospat)
```

```
> ls("package:ecospat")
```

```
[1] "ecospat.binary.model"      "ecospat.boyce"
[3] "ecospat.calculate.pd"      "ecospat.caleval"
[5] "ecospat.co_occurrences"    "ecospat.cons_Cscore"
[7] "ecospat.cv.example"        "ecospat.cv.gam"
[9] "ecospat.cv.gbm"            "ecospat.cv.glm"
[11] "ecospat.cv.me"             "ecospat.cv.rf"
[13] "ecospat.env"               "ecospat.Epred"
[15] "ecospat.fun.arrows"        "ecospat.grid.clim.dyn"
[17] "ecospat.makeDataFrame"     "ecospat.mantel.correlogram"
[19] "ecospat.maxentvarimport"    "ecospat.mess"
[21] "ecospat.migclim"           "ecospat.niche.dyn.index"
[23] "ecospat.niche.equivalency.test" "ecospat.niche.overlap"
[25] "ecospat.niche.similarity.test" "ecospat.npred"
[27] "ecospat.occ.desaggregation" "ecospat.permut.glm"
[29] "ecospat.plot.contrib"      "ecospat.plot.mess"
[31] "ecospat.plot.niche"        "ecospat.plot.niche.dyn"
[33] "ecospat.plot.overlap.test" "ecospat.rand.pseudoabsences"
[35] "ecospat.sample.envvar"     "ecospat.SSDMeval"
[37] "ecospat.testData"          "ecospat.testNiche"
[39] "ecospat.varpart"
```

```
> data(package="ecospat").
```

Data sets in package 'ecospat':

ecospat.testData	Test data for the ecospat library
ecospat.testNiche	Test data for the niche overlap analysis

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

To cite package 'ecospat' in publications use:

Olivier Broenniman, Blaise Petitpierre, Christophe Randin, Robin Engler, Frank Breiner, Manuela D'Amen, Loic Pellissier, Julien Pottier, Dorothea Pio, Ruben Garcia Mateo, Valeria Di Cola, Wim Hordijk, Anne Dubuis, Daniel Scherrer, Nicolas Salamin and Antoine Guisan (2014). ecospat: Spatial ecology miscellaneous methods. R package version 1.0.

A BibTeX entry for LaTeX users is

```
@Manual{,
  title = {ecospat: Spatial ecology miscellaneous methods},
  author = {Olivier Broenniman and Blaise Petitpierre and Christophe Randin and Robin Engler and Frank Breiner and Manuela D'Amen and Loic Pellissier and Julien Pottier and Dorothea Pio and Ruben Garcia Mateo and Valeria Di Cola and Wim Hordijk and Anne Dubuis and Daniel Scherrer and Nicolas Salamin and Antoine Guisan},
  year = {2014},
  note = {R package version 1.0},
}
```

1.1 Test data for the ecospat library

```
> dim(ecospat.testData)
```

```
[1] 300 82
```

```
> names(ecospat.testData)
```

[1]	"numplots"	"long"
[3]	"lat"	"ddeg"
[5]	"mind"	"srad"
[7]	"slp"	"topo"
[9]	"Agrostis_capillaris"	"Leontodon_hispidus_sl"
[11]	"Dactylis_glomerata"	"Trifolium_repens_sstr"
[13]	"Geranium_sylvaticum"	"Ranunculus_acris_sl"
[15]	"Prunella_vulgaris"	"Veronica_chamaedrys"
[17]	"Taraxacum_officinale_aggr"	"Plantago_lanceolata"
[19]	"Potentilla_erecta"	"Carex_sempervirens"
[21]	"Soldanella_alpina"	"Cynosurus_cristatus"
[23]	"Campanula_scheuchzeri"	"Festuca_pratensis_sl"
[25]	"Polygonum_viviparum"	"Ranunculus_montanus_aggr"
[27]	"Rumex_acetosa"	"Carex_flacca"
[29]	"Potentilla_aurea"	"Homogyne_alpina"
[31]	"Briaza_media"	"Lathyrus_pratensis"
[33]	"Poa_alpina"	"Sesleria_caerulea"
[35]	"Trollius_europaeus"	"Anthyllis_vulneraria_sl"
[37]	"Deschampsia_cespitosa"	"Trisetum_flavescens"
[39]	"Galium_anisophyllum"	"Euphrasia_minima"
[41]	"Astrantia_majior"	"Nardus_stricta"
[43]	"Ligusticum_mutellina"	"Hieracium_lactucella"
[45]	"Chaerophyllum_hirsutum_aggr"	"Helianthemum_nummularium_sl"
[47]	"Scabiosa_lucida"	"Cerastium_fontanum_sl"
[49]	"Carex_pallescens"	"Plantago_alpina"
[51]	"Crepis_aurea"	"Leontodon_helveticus"
[53]	"Bromus_erectus_sstr"	"Polygonum_bistorta"
[55]	"Saxifraga_oppositifolia"	"Daucus_carota"
[57]	"Parnassia_palustris"	"Pritzelago_alpina_sstr"
[59]	"glm_Agrostis_capillaris"	"glm_Leontodon_hispidus_sl"
[61]	"glm_Dactylis_glomerata"	"glm_Trifolium_repens_sstr"
[63]	"glm_Geranium_sylvaticum"	"glm_Ranunculus_acris_sl"
[65]	"glm_Prunella_vulgaris"	"glm_Veronica_chamaedrys"
[67]	"glm_Taraxacum_officinale_aggr"	"glm_Plantago_lanceolata"
[69]	"glm_Potentilla_erecta"	"glm_Carex_sempervirens"
[71]	"glm_Soldanella_alpina"	"glm_Cynosurus_cristatus"
[73]	"glm_Campanula_scheuchzeri"	"glm_Festuca_pratensis_sl"
[75]	"glm_Bromus_erectus_sstr"	"glm_Saxifraga_oppositifolia"
[77]	"glm_Daucus_carota"	"glm_Pritzelago_alpina_sstr"
[79]	"gbm_Bromus_erectus_sstr"	"gbm_Saxifraga_oppositifolia"
[81]	"gbm_Daucus_carota"	"gbm_Pritzelago_alpina_sstr"

1.2 Test data for Niche Overlap Analysis

```
> dim(ecospat.testNiche)
```

```
[1] 99 4
```

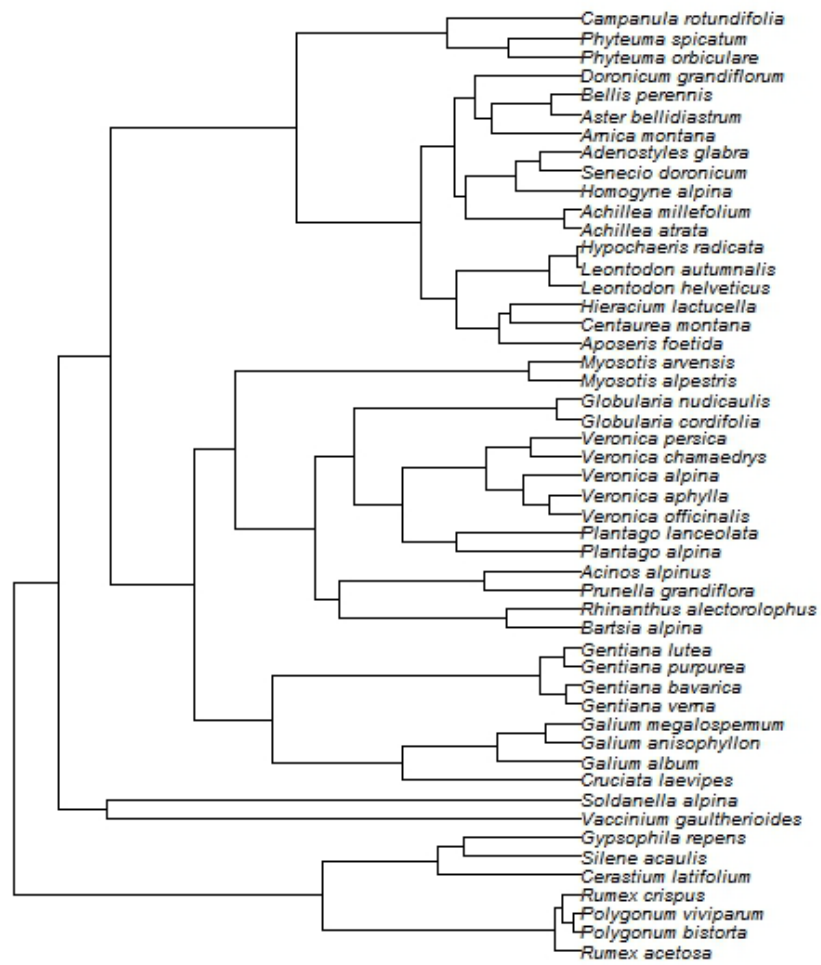
```
> names(ecospat.testNiche)
```

```
[1] "species" "long" "lat" "Spp"
```

1.3 Test tree for the phylogenetic diversity analysis

```
> fpath<-system.file("extdata", "ecospat.testTree.tre", package="ecospat")
> fpath
[1] "~/Documents/R/win-library/3.0/ecospat/extdata/ecospat.testTree.tre"
> tree<-read.tree(fpath)
> is.ultrametric(tree)
[1] TRUE
> is.rooted(tree)
[1] TRUE
> 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" "Crucifera_laevipes"
[11] "Galium_album" "Galium_aniaphyllum"
[13] "Galium_megalospermum" "Gentiana_verna"
[15] "Gentiana_bavaria" "Gentiana_purpurea"
[17] "Gentiana_lutea" "Bartsia_alpina"
[19] "Rhinanthus_allectorolophus" "Prunella_grandiflora"
[21] "Acinosa_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_radiata"
[39] "Achillea_atrata" "Achillea_millefolium"
[41] "Homogyne_alpina" "Senecio_doronicum"
[43] "Adenostyles_glabra" "Arnica_montana"
[45] "Aster_bellidiastrum" "Bellis_perennis"
[47] "Doronicum_grandiflorum" "Phyteuma_orbiculare"
[49] "Phyteuma_spicatum" "Campanula_rotundifolia"
```

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

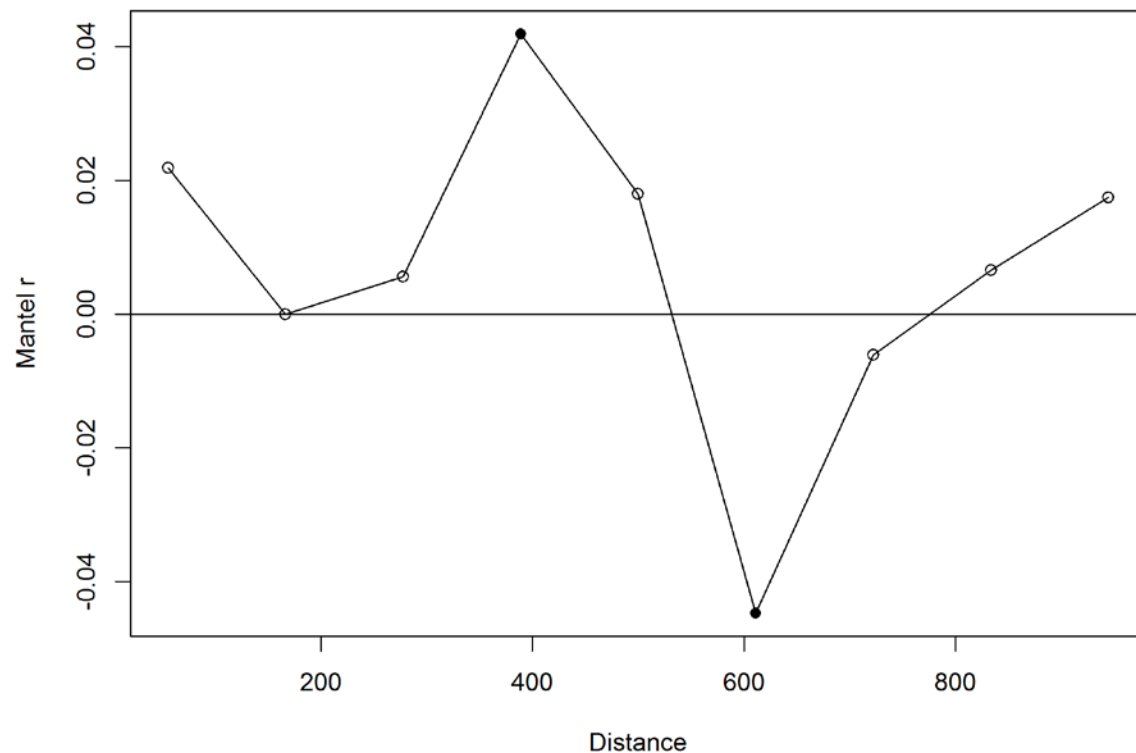


2 Pre-Modelling Analysis

2.1 Spatial Auto-correlation

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)
```



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

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
```

Number of Predictors with Spearman Correlation `ecospat.npred`

```
> x <- cor(colvar, method="spearman")  
> ecospat.npred(x, th=0.75)  
[1] 4
```

2.3 MESS - Multivariate Environmental Similarity Surfaces

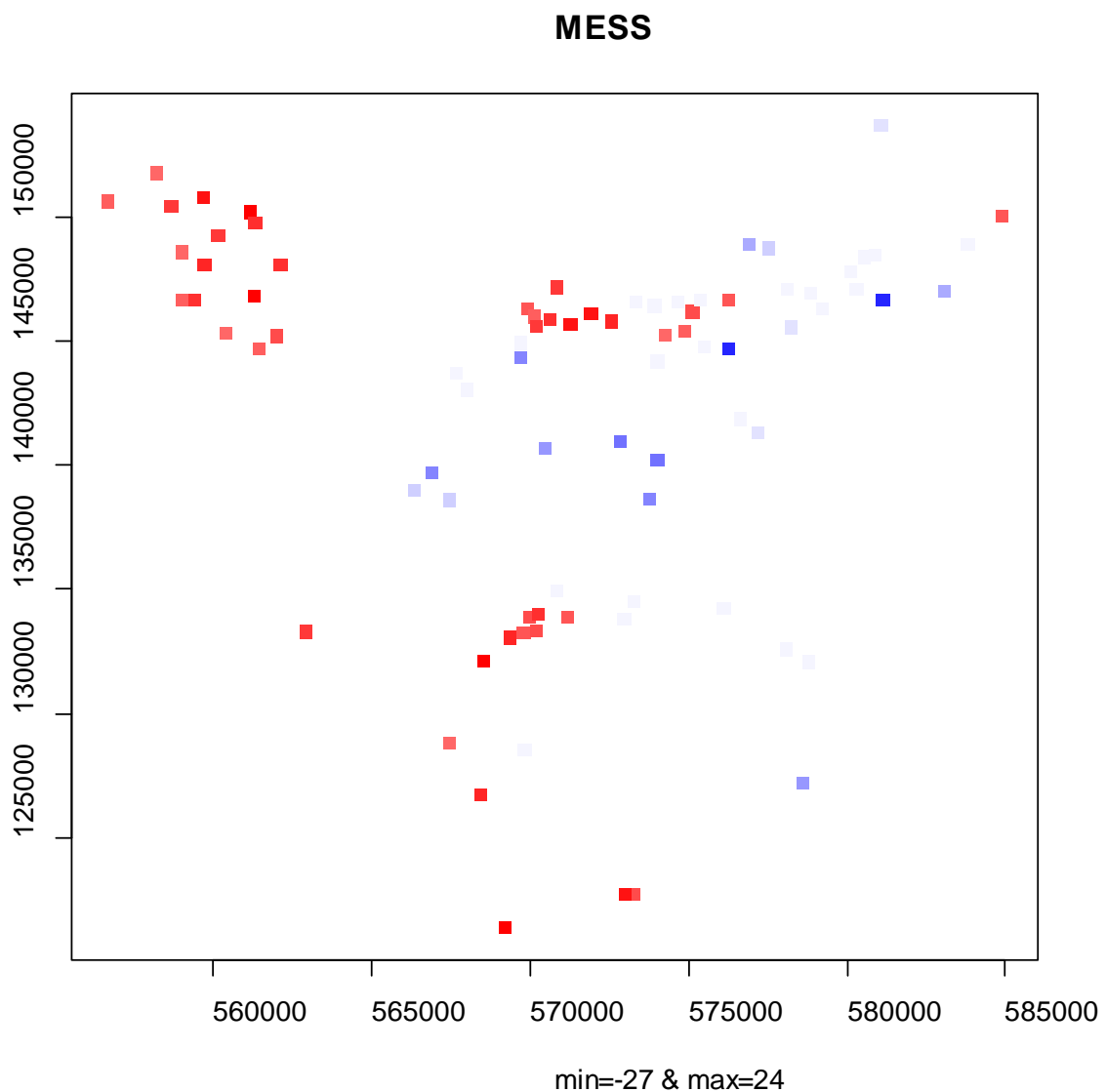
```
> x <- ecospat.testData[c(2, 3, 4:8)]  
> proj <- x[1:90,] #A projection dataset.  
> cal <- x[91:300,] #A calibration dataset
```

Create a MESS object with `ecospat.mess`

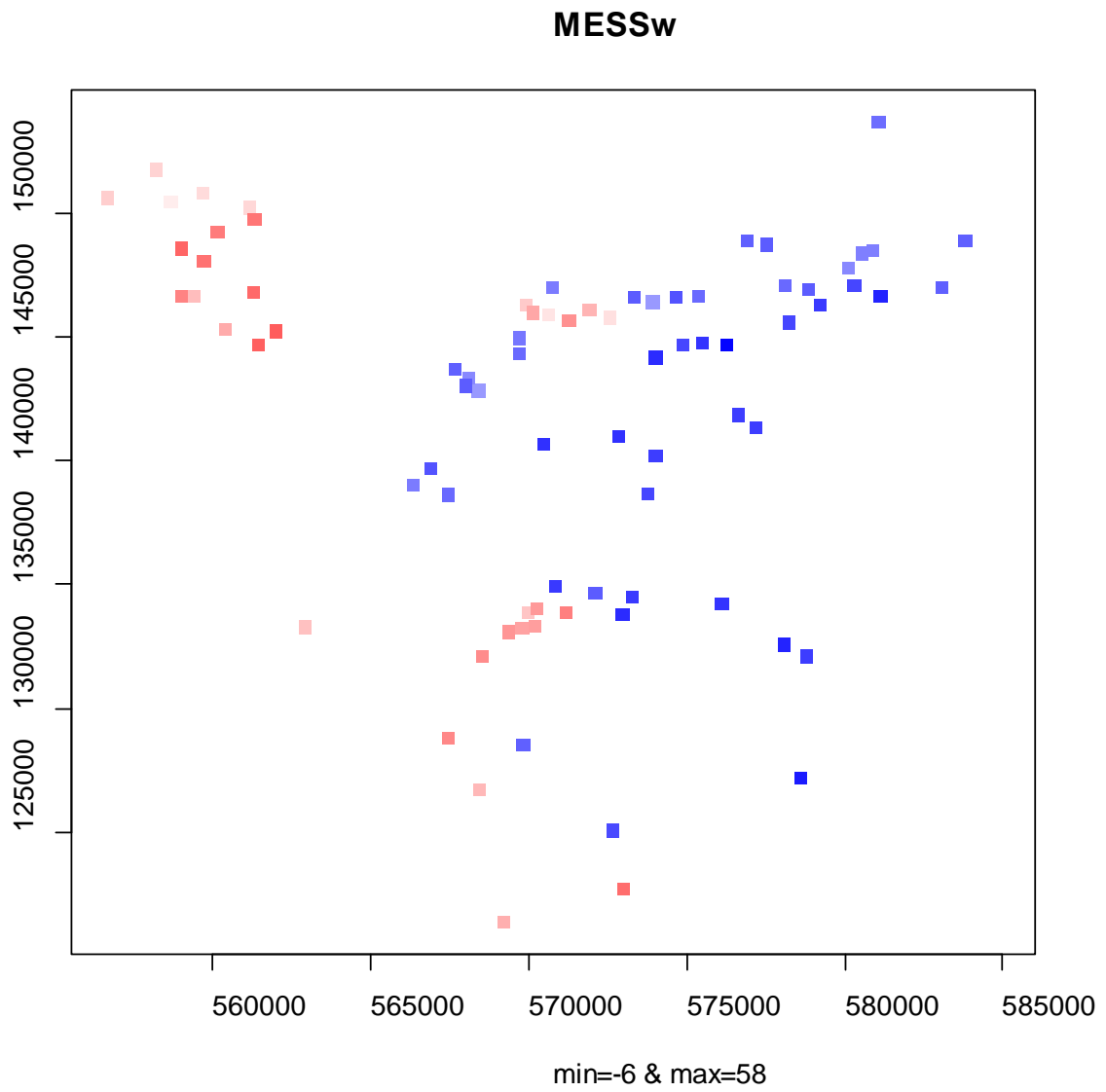
```
> mess.object <- ecospat.mess (proj, cal, w="default")
```

Plot MESS with `ecospat.plot.mess`

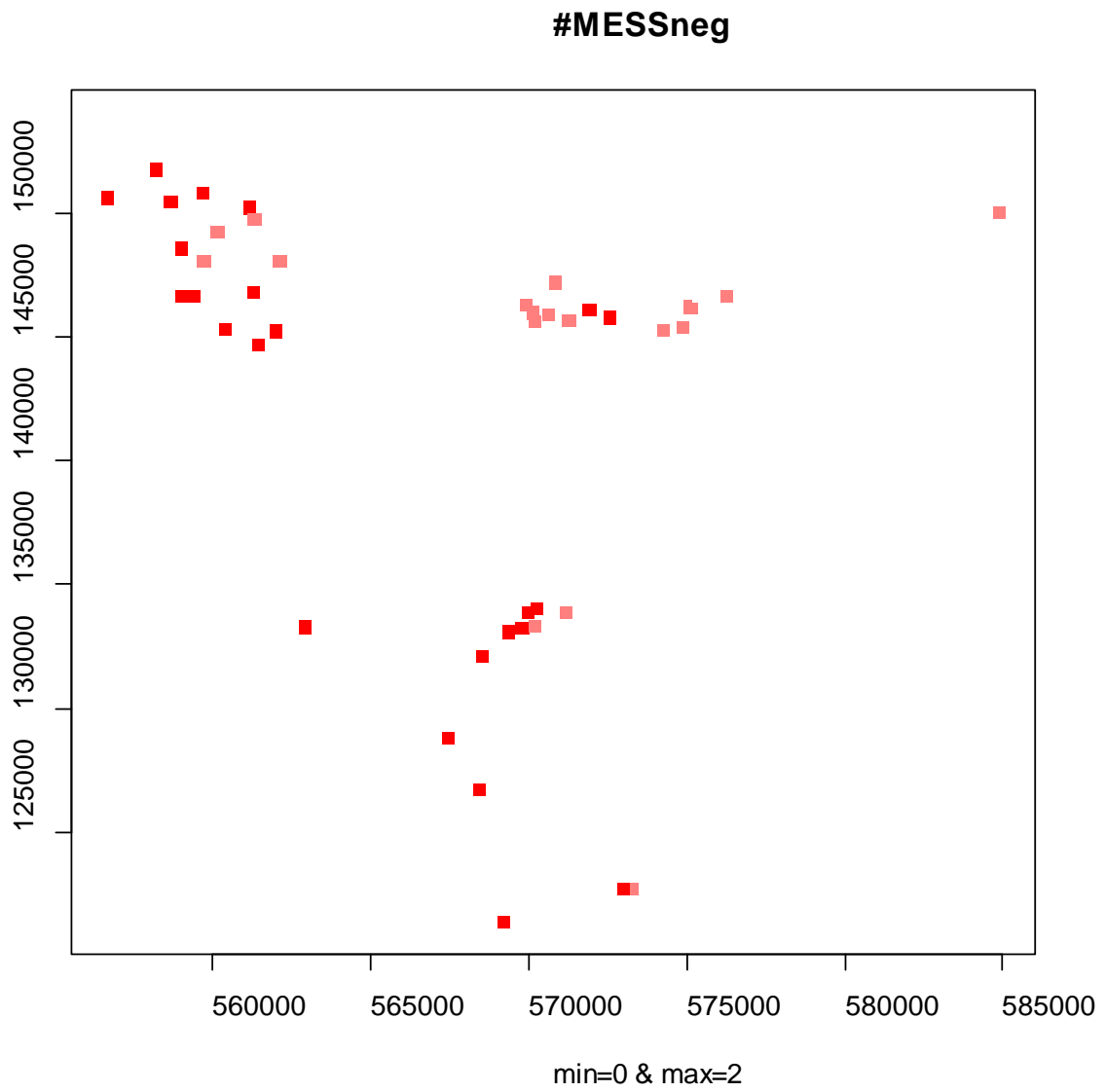
```
> ecospat.plot.mess (xy=proj[c(1:2)], mess.object, cex=1, pch=15)
```



Pixels in red indicate sites where at least one environmental predictor has values outside of the range of that predictor in the calibration dataset.



Same as previous plot but with weighted by the number of predictors



The plot shows at each site how many predictors have values outside of their calibration range.

2.4 Phylogenetic Diversity Measures

Calculate Phylogenetic Diversity Measures `ecospat.calculate.pd`

```
> data <- ecospat.testData[9: 58]
```

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

Progress (. = 100 pixels calculated):

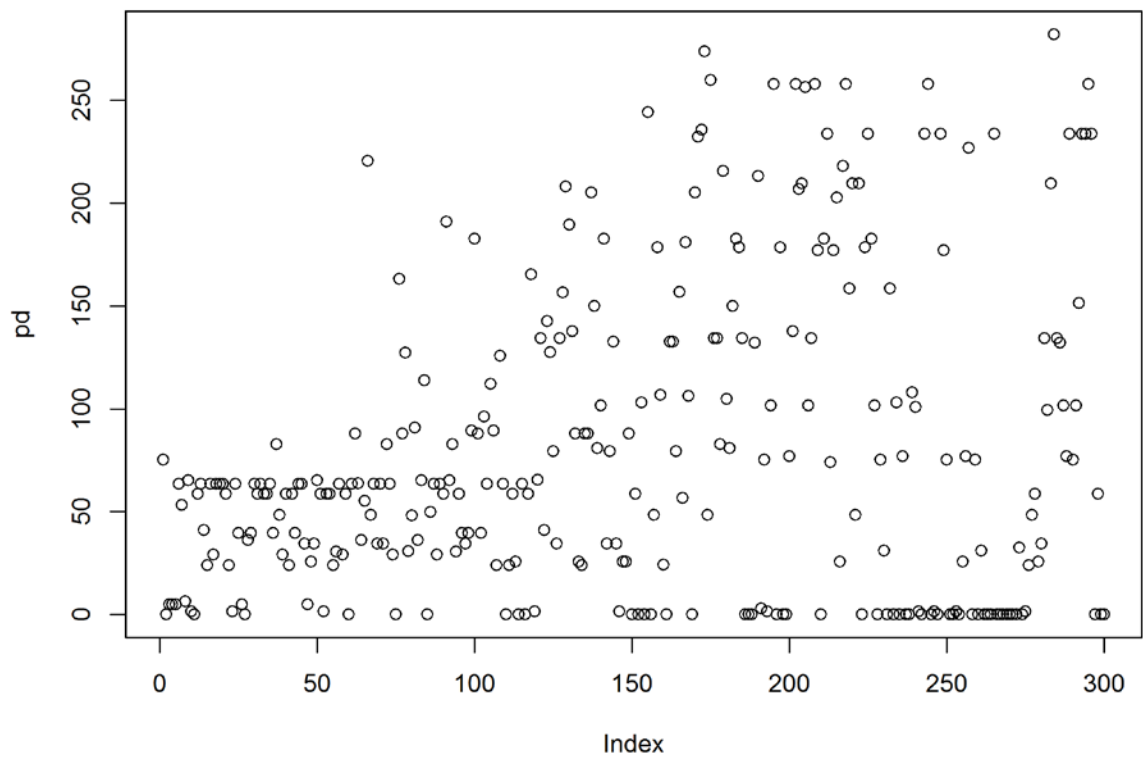
```
... [300]
```

All 300 pixels done.

```
> pd
```

[1]	75.311274	0.000000	4.996470	4.996470	4.996470	63.636036
[7]	53.302515	6.522304	65.161869	1.525834	0.000000	58.639566
[13]	63.636036	41.068925	24.092944	63.636036	29.089414	63.636036
[19]	63.636036	63.636036	58.639566	24.092944	1.525834	63.636036
[25]	39.543091	4.996470	0.000000	36.072455	39.543091	63.636036
[31]	58.639566	63.636036	58.639566	58.639566	63.636036	39.543091
[37]	82.852666	48.306045	29.089414	58.639566	24.092944	58.639566
[43]	39.543091	63.636036	63.636036	34.546621	4.996470	25.618778
[49]	34.546621	65.161869	58.639566	1.525834	58.639566	58.639566
[55]	24.092944	30.615248	63.636036	29.089414	58.639566	0.000000
[61]	63.636036	87.849136	63.756192	36.072455	55.133956	220.560121
[67]	48.306045	63.636036	34.546621	63.636036	34.546621	82.852666
[73]	63.636036	29.089414	0.000000	163.160410	87.849136	127.288751
[79]	30.735404	48.185889	90.900803	36.072455	65.161869	113.893678
[85]	0.000000	49.831879	63.636036	29.089414	63.636036	58.639566
[91]	191.044944	65.161869	82.852666	30.615248	58.639566	39.543091
[97]	34.546621	39.543091	89.374970	182.542864	87.849136	39.543091
[103]	96.323037	63.636036	112.062237	89.374970	24.092944	125.762918
[109]	63.636036	0.000000	24.092944	58.639566	25.618778	0.000000
[115]	63.636036	0.000000	58.639566	165.306009	1.525834	65.587633
[121]	134.236818	41.068925	142.738898	127.408908	79.347056	34.546621
[127]	134.236818	156.498322	208.022259	189.519110	137.681015	87.849136
[133]	25.738934	24.092944	87.849136	87.849136	205.109974	149.976019
[139]	80.872890	101.669974	182.542864	34.546621	79.347056	132.710985
[145]	34.546621	1.525834	25.618778	25.618778	87.849136	0.000000
[151]	58.639566	0.000000	103.195807	0.000000	244.094714	0.000000
[157]	48.306045	178.507081	106.666444	24.213101	0.000000	132.710985
[163]	132.710985	79.347056	156.803929	56.659789	181.017030	106.352285
[169]	0.000000	205.230131	232.235360	235.620813	273.593338	48.306045
[175]	259.833914	134.236818	134.236818	82.732510	215.563651	104.721641
[181]	80.872890	149.855862	182.542864	178.507081	134.236818	0.000000
[187]	0.000000	0.000000	132.091220	213.053702	3.051667	75.311274
[193]	1.525834	101.669974	257.854137	0.000000	178.507081	0.000000
[199]	0.000000	76.837107	137.707455	257.854137	206.755964	209.548092
[205]	256.328304	101.669974	134.236818	257.854137	176.981248	0.000000
[211]	182.542864	233.761193	74.044979	176.981248	202.600025	25.738934
[217]	218.050172	257.854137	158.449919	209.548092	48.306045	209.548092
[223]	0.000000	178.507081	233.641037	182.542864	101.669974	0.000000
[229]	75.311274	31.041011	0.000000	158.449919	0.000000	103.195807
[235]	0.000000	76.837107	0.000000	0.000000	107.878119	100.930052
[241]	1.525834	0.000000	233.761193	257.854137	0.000000	1.525834
[247]	0.000000	233.761193	176.981248	75.311274	0.000000	0.000000
[253]	1.525834	0.000000	25.738934	76.837107	226.813126	0.000000
[259]	75.311274	0.000000	31.041011	0.000000	0.000000	0.000000
[265]	233.761193	0.000000	0.000000	0.000000	0.000000	0.000000
[271]	0.000000	0.000000	32.566845	0.000000	1.525834	24.092944
[277]	48.306045	58.639566	25.618778	34.546621	134.236818	99.404218
[283]	209.548092	282.067238	134.236818	132.091220	101.669974	76.837107
[289]	233.641037	75.311274	101.669974	151.501852	233.761193	233.641037
[295]	257.854137	233.761193	0.000000	58.639566	0.000000	0.000000

```
> plot(pd)
```



2.5 Biotic Interactions

Species Co-occurrences pattern with a Presence-absence matrix

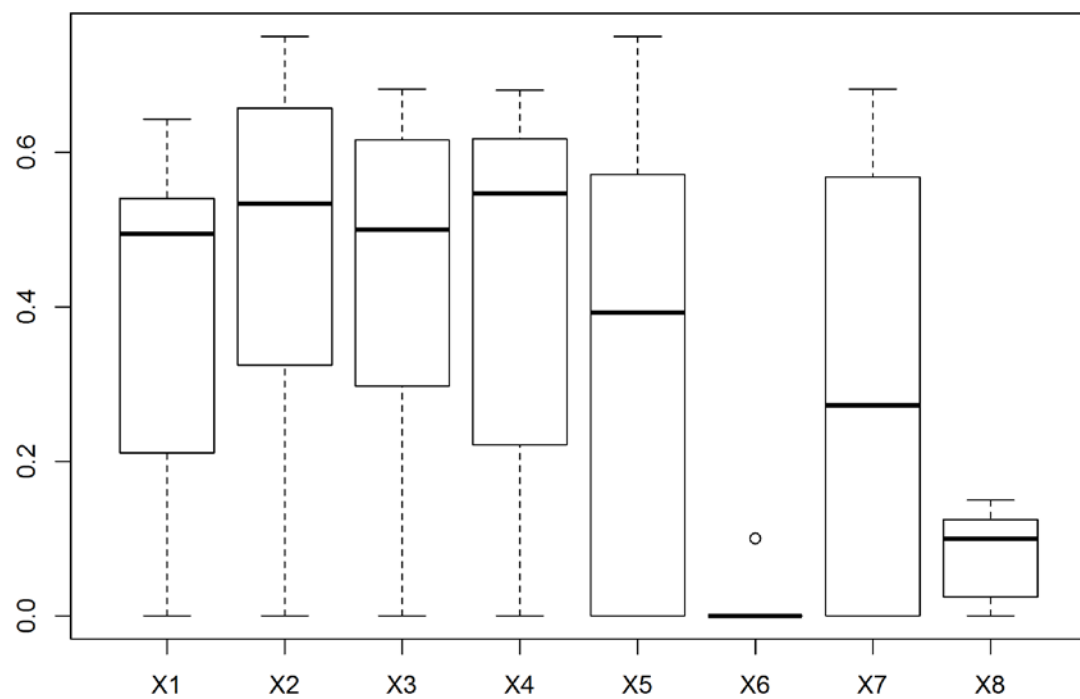
```
> data <- ecospat.testData[c(9: 16, 54: 57)]
> ecospat.co_occurrences (data)
```

	Gerani um_syl vati cum	Ranuncul us_acri s_sl
Gerani um_syl vati cum	1. 0000000	0. 5339806
Ranuncul us_acri s_sl	0. 5339806	1. 0000000
Prunell a_vul gari s	0. 4950495	0. 6336634
Veroni ca_chamaedrys	0. 5463918	0. 6804124
Polygonum_bi storta	0. 6428571	0. 7500000
Saxi fraga_opposi ti fol ia	0. 0000000	0. 0000000
Daucus_carota	0. 2727273	0. 5000000
Parnassi a_pal ustri s	0. 1500000	0. 1500000
	Prunell a_vul gari s	Veroni ca_chamaedrys
Gerani um_syl vati cum	0. 4950495	0. 5463918
Ranuncul us_acri s_sl	0. 6336634	0. 6804124
Prunell a_vul gari s	1. 0000000	0. 5979381
Veroni ca_chamaedrys	0. 5979381	1. 0000000
Polygonum_bi storta	0. 5000000	0. 3928571
Saxi fraga_opposi ti fol ia	0. 0000000	0. 0000000
Daucus_carota	0. 6818182	0. 6363636
Parnassi a_pal ustri s	0. 1000000	0. 0500000
	Polygonum_bi storta	
Gerani um_syl vati cum	0. 6428571	
Ranuncul us_acri s_sl	0. 7500000	
Prunell a_vul gari s	0. 5000000	
Veroni ca_chamaedrys	0. 3928571	
Polygonum_bi storta	1. 0000000	
Saxi fraga_opposi ti fol ia	0. 0000000	
Daucus_carota	0. 0000000	
Parnassi a_pal ustri s	0. 0000000	
	Saxi fraga_opposi ti fol ia	Daucus_carota
Gerani um_syl vati cum	0. 0	0. 2727273
Ranuncul us_acri s_sl	0. 0	0. 5000000
Prunell a_vul gari s	0. 0	0. 6818182
Veroni ca_chamaedrys	0. 0	0. 6363636
Polygonum_bi storta	0. 0	0. 0000000
Saxi fraga_opposi ti fol ia	1. 0	0. 0000000
Daucus_carota	0. 0	1. 0000000
Parnassi a_pal ustri s	0. 1	0. 0000000
	Parnassi a_pal ustri s	
Gerani um_syl vati cum	0. 15	
Ranuncul us_acri s_sl	0. 15	
Prunell a_vul gari s	0. 10	
Veroni ca_chamaedrys	0. 05	
Polygonum_bi storta	0. 00	
Saxi fraga_opposi ti fol ia	0. 10	
Daucus_carota	0. 00	
Parnassi a_pal ustri s	1. 00	

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.

$$\text{Ind}_{\omega o} = \frac{N_{(S1 \cap S2)}}{\text{Min}(N_{S1}, N_{S2})},$$

where $N_{(S1 \cap S2)}$ is the number of times species S1 and S2 co-occur, while $\text{Min}(N_{S1}, N_{S2})$ is the number of times species S1 and S2 co-occur, while is the occurrence frequency of the rarest of the two species.

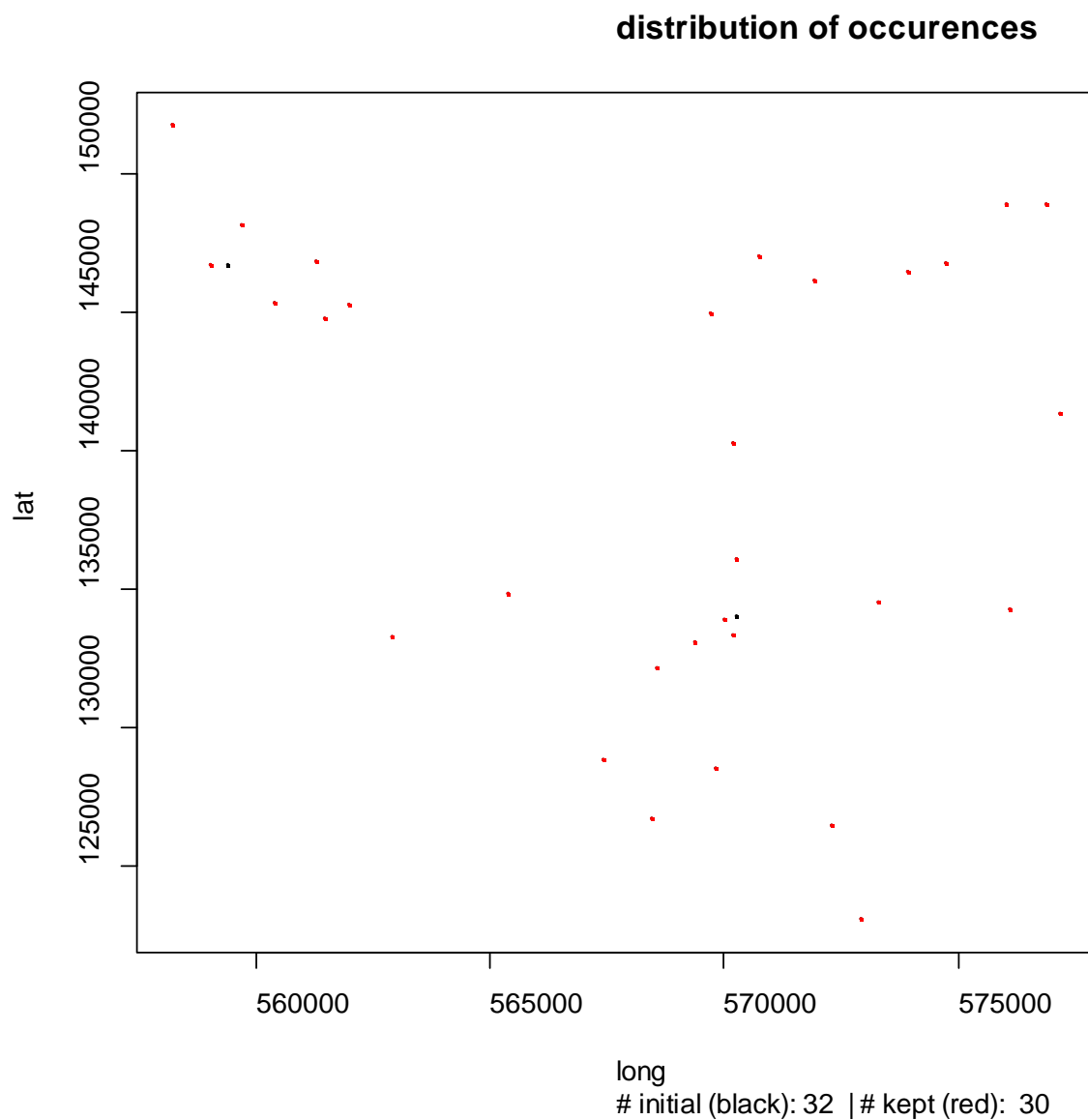


2.6 Niche Quantification with Ordination techniques

2.6.1 Species Occurrences Desaggregation

Occurrences Desaggregation of *Bromus erectus sstr* with
`ecospat.occ.desaggregation`

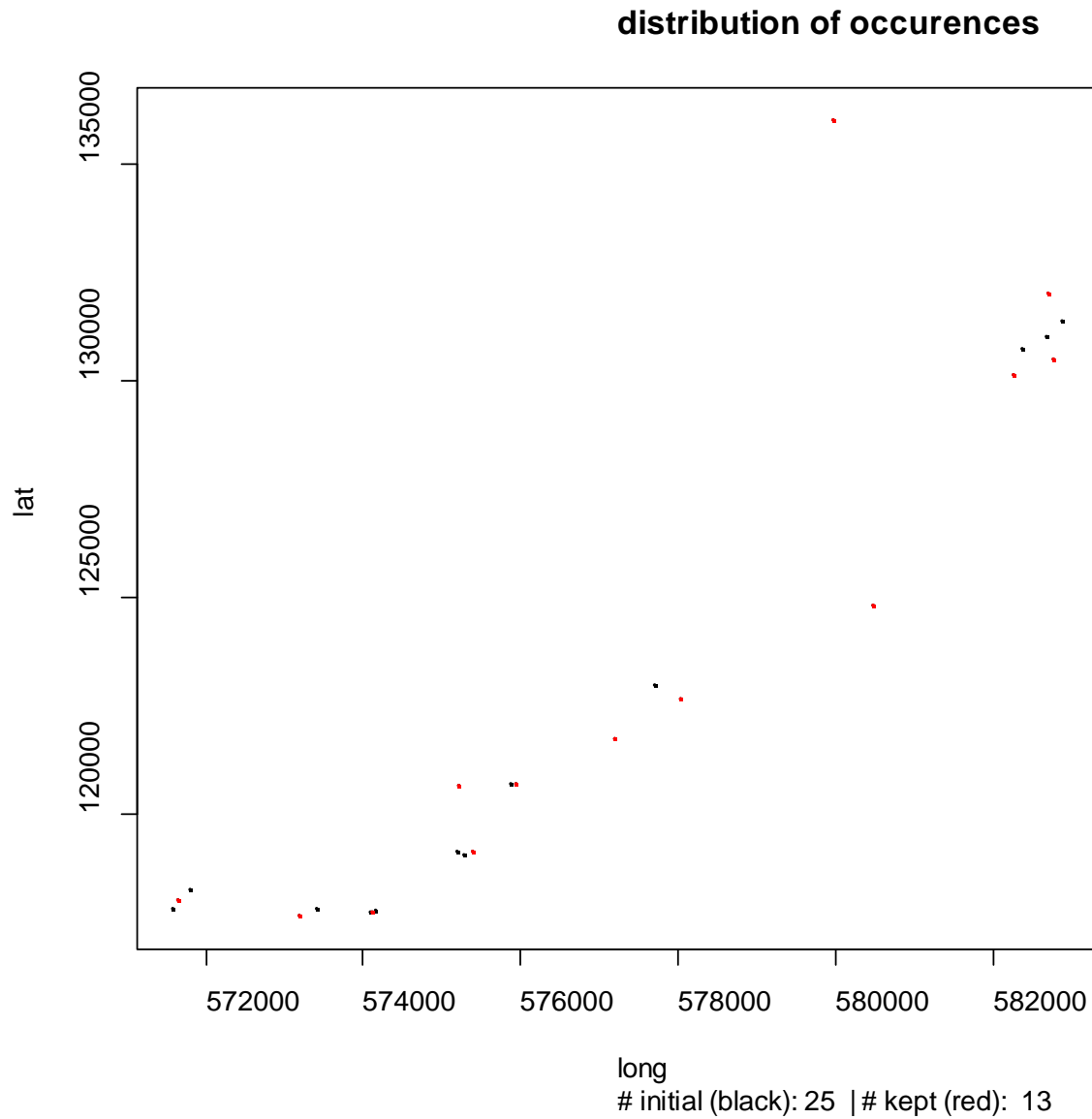
```
> spp<-ecospat.testNiche  
> sp1<- spp[1:32, 1:3]  
> occ.sp1<-ecospat.occ.desaggregation(dfvar=sp1, col xy=2:3, col var=NULL,  
min.dist=500, plot=TRUE)
```



Pixels in red are more distant from each other than 500 meters and are kept in the final dataset.

Occurrences Desaggregation of *Saxifraga oppositifolia* with
`ecospat.occ.desaggregation`

```
> sp2<- spp[33:57, 1:3]  
> occ.sp2<- ecospat.occ.desaggregation(dfvar=sp2, col xy=2:3, col var=NULL,  
mi n. di st=500, pl ot=TRUE)
```



Pixels in red are more distant from each other than 500 meters and are kept in the final dataset.

2.6.2 Sample Environmental Variables

Add environmental values to the *Bromus erectus sstr* species dataframe with `ecospat.sample.envar`

```
> clim<- ecospat.testData[2:8]
> occ_sp1<-na.exclude(ecospat.sample.envar(dfsp=occ.sp1, colspxy=1:2,
colspkept=1:2, dfvar=clim, colvarxy=1:2, colvar="all", resolution=25))
```

Add environmental values to the *Saxifraga oppositifolia* species dataframe with `ecospat.sample.envar`

```
> occ_sp2<-na.exclude(ecospat.sample.envar(dfsp=occ.sp2, colspxy=1:2,
colspkept=1:2, dfvar=clim, colvarxy=1:2, colvar="all", resolution=25))
```

2.6.3 Dynamic Occurrence Densities Grid

```
> spp<- ecospat.testNiche
> clim<- ecospat.testData[2:8]
>
>
> occ.sp_test<-na.exclude(ecospat.sample.envar(dfsp=spp, colspxy=2:3, colspkept=1:3, dfvar=clim, colvarxy=1:2, colvar="all", resolution=25))
>
> occ.sp<-cbind(occ.sp_test, spp[,4]) #add species names
> # list of species
> sp.list<-levels(occ.sp[,1])
> sp.nbocc<-c()
> for (i in 1:length(sp.list)){sp.nbocc<-c(sp.nbocc,length(which(occ.sp[,1]
== sp.list[i])))}
> #calculate the nb of occurrences per species
> #remove species with less than 5 occurrences
> sp.list<-sp.list[sp.nbocc>4]
> nb.sp<-length(sp.list) #nb of species
> ls()
[1] "clim"          "ecospat.testData"  "ecospat.testNiche" "i"          "nb.sp"
[6] "occ.sp"        "occ.sp_test"      "occ.sp1"          "occ.sp2"    "occ_sp1"
[11] "occ_sp2"       "sp.list"          "sp.nbocc"         "sp1"        "sp2"
[16] "spp"

> # selection of variables to include in the analyses
> # try with all and then try only WorldClim Variables
> Xvar<-c(3:7)
> nvar<-length(Xvar)
>
> #number of iteration for the tests of equivalency and similarity
> iterations<-100
> #resolution of the gridding of the climate space
> R=100
```

```

> #####
> ##### PCA- ENVIRONMENT #####
> #####

> data<-rbind(occ.sp[,Xvar+1],clim[,Xvar])
> #dataset for the analysis, includes all the sites of the study area + the
occurences for all the species
> head(data)
  ddeg mind   srad slp topo
1 2416 -137 228574   9   4
2 2221 -612 212396  17 -137
3 2281 -995 252419  20   36
4 2271  128 261695  20   65
5 2160 -714 260045  23    6
6 2212 -632 210162   4 -121
> dim(data)
[1] 399 5

> w<-c(rep(0,nrow(occ.sp)),rep(1,nrow(clim)))
> #vector of weight, 0 for the occurences, 1 for the sites of the study ar
ea
> pca.cal<-dudi.pca(data, row.w = w, center = T, scale = T, scanf = F, n
f = 2)
> # the pca is calibrated on all the sites of the study area
> # occurences are not used for the calibration, but their scores are calc
ulated
> ##### selection of species #####
> sp.list
[1] "sp1" "sp2" "sp3" "sp4"
> sp.combn<-combn(1:2,2)
> for(i in 1:ncol(sp.combn)) {
+   row.sp1<-which(occ.sp[,1] == sp.list[sp.combn[1,i]]) # rows in data co
rresponding to sp1
+   row.sp2<-which(occ.sp[,1] == sp.list[sp.combn[2,i]]) # rows in data co
rresponding to sp2
+   name.sp1<-sp.list[sp.combn[1,i]]
+   name.sp2<-sp.list[sp.combn[2,i]]
+   # predict the scores on the axes
+   scores.clim<-pca.cal$li[(nrow(occ.sp)+1):nrow(data),] #scores for gl
obal climate
+   scores.sp1<-pca.cal$li[row.sp1,] #scores for sp1
+   scores.sp2<-pca.cal$li[row.sp2,] #scores for sp2
+ }

```

Dynamic Occurrence Densities Grid `ecospat.grid.clim.dyn`

```

> z1.dyn<-ecospat.grid.clim.dyn (scores.clim, scores.clim, scores.sp1, R=1
00)
> z2.dyn<-ecospat.grid.clim.dyn (scores.clim, scores.clim, scores.sp2, R=1
00)

```

Niche Equivalency Test `ecospat.niche.equivalency.test`

```

> a.dyn<-ecospat.niche.equivalency.test(z1=z1.dyn , z2=z2.dyn, rep=100)

```

Niche Similarity Test `ecospat.niche.similarity.test`

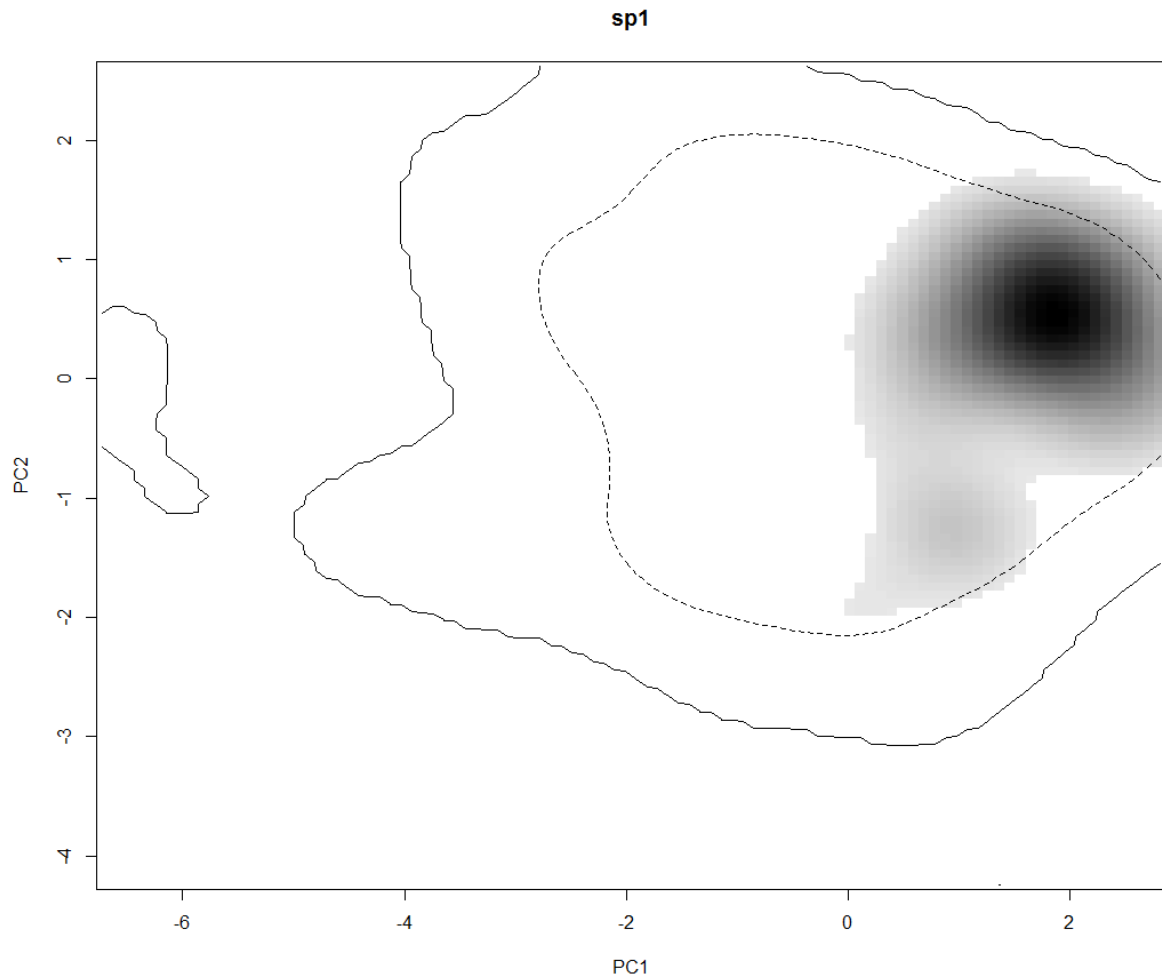
```

> b.dyn<-ecospat.niche.similarity.test(z1=z1.dyn , z2=z2.dyn, rep=100)
> b2.dyn<-ecospat.niche.similarity.test(z1=z2.dyn , z2=z1.dyn, rep=100)

```

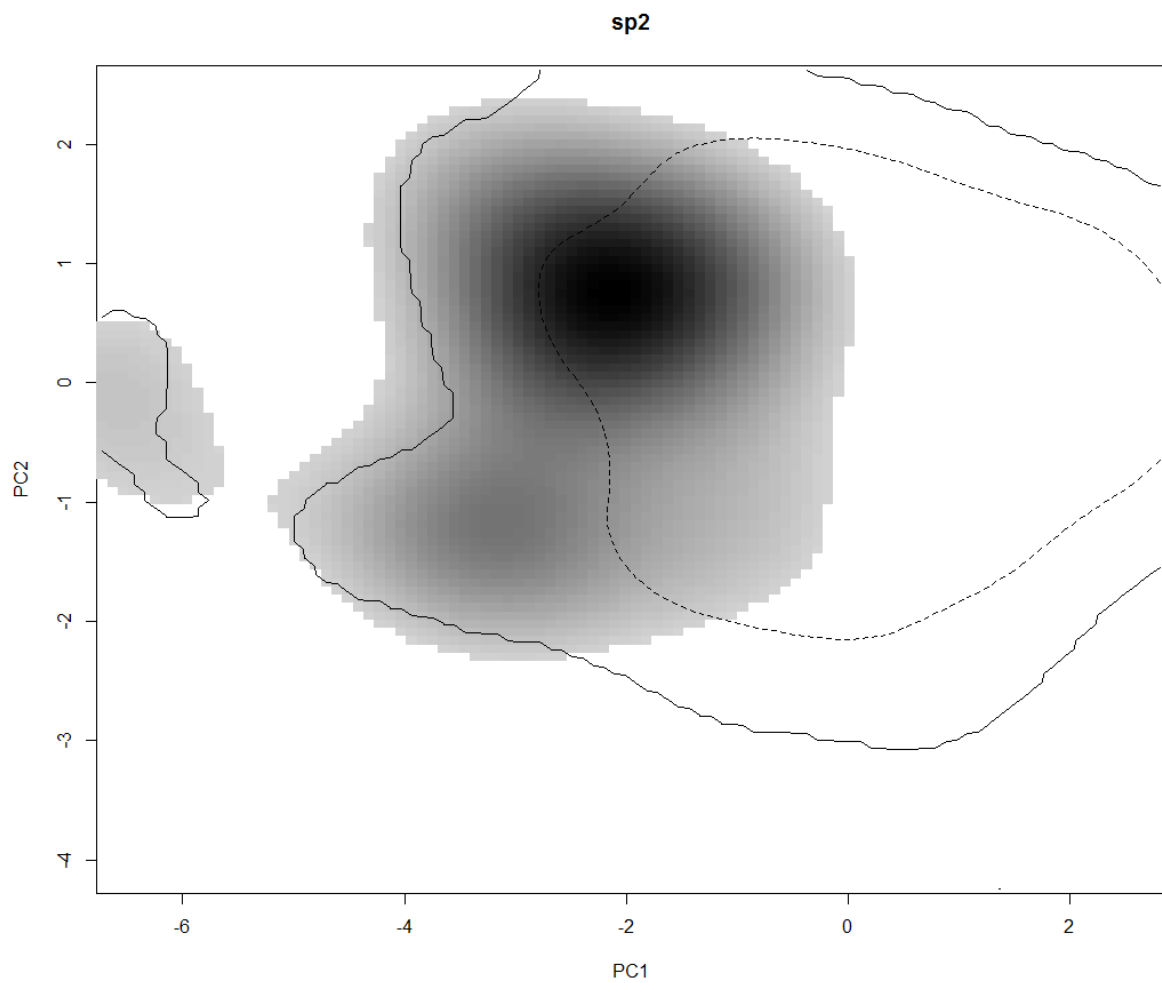
Plot Species Density `ecospat.plot.niche` for one species

```
> z1<- ecospat.grid.clim.dyn(scores.clim,scores.clim,scores.sp1,R)
> ecospat.plot.niche (z1, title="sp1", name.axis1="PC1", name.axis2="PC2",
cor=F)
```



The plot shows the occupancy of species 1 (grey gradient) and environmental availability in the study area (solid line shows 100% of available climates; dashed line shows 50% most frequent available climates)

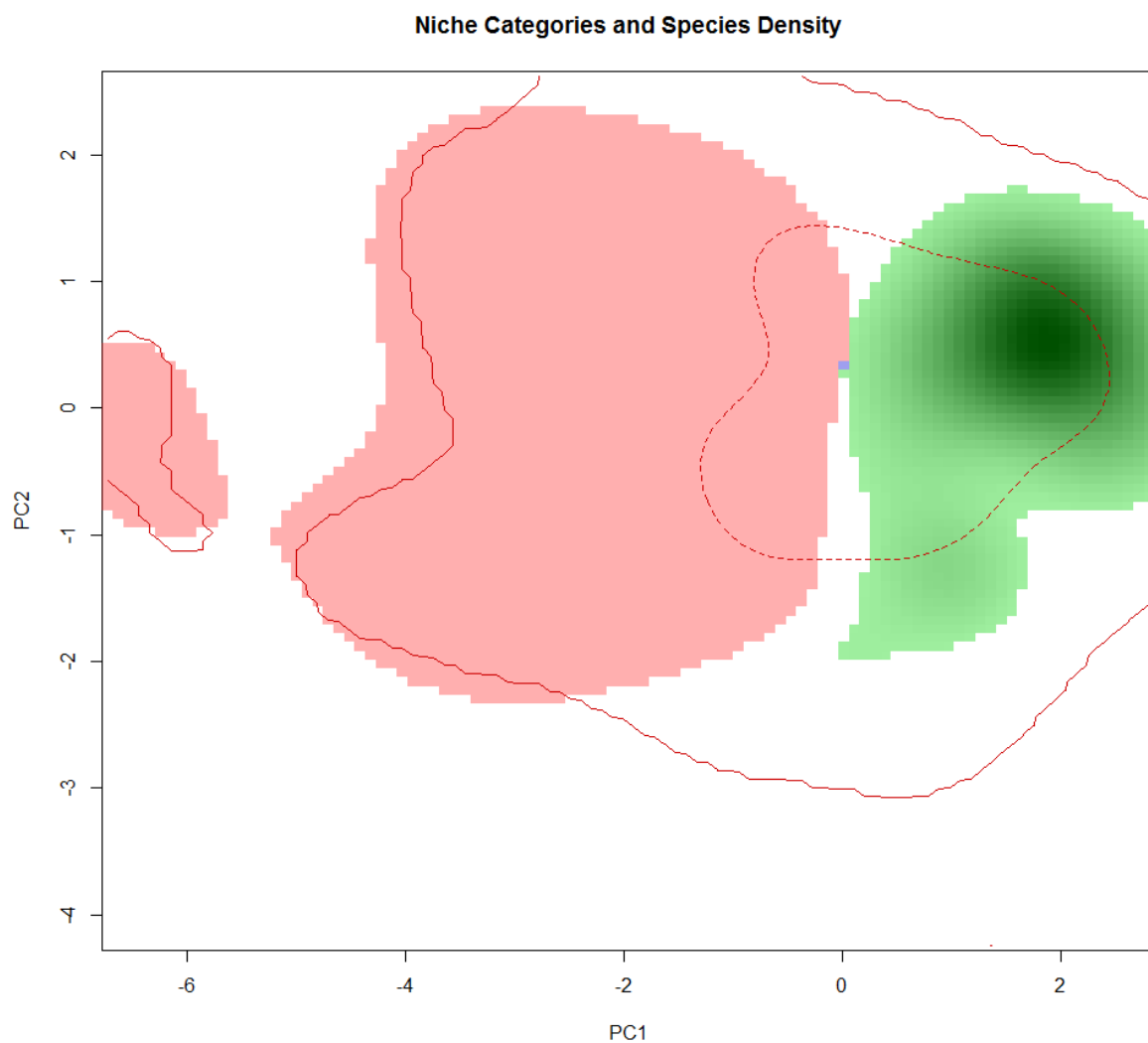
```
> z2<- ecospat.grid.clim.dyn(scores.clim,scores.clim,scores.sp2,R)
> ecospat.plot.niche (z2, title="sp2", name.axis1="PC1", name.axis2="PC2",
cor=F)
```



The plot shows the occupancy of species 2 (grey gradient) and environmental availability in the study area (solid line shows 100% of available climates; dashed line shows 50% most frequent available climates)

Niche Categories and Species Density `ecospat.plot.niche.dyn`

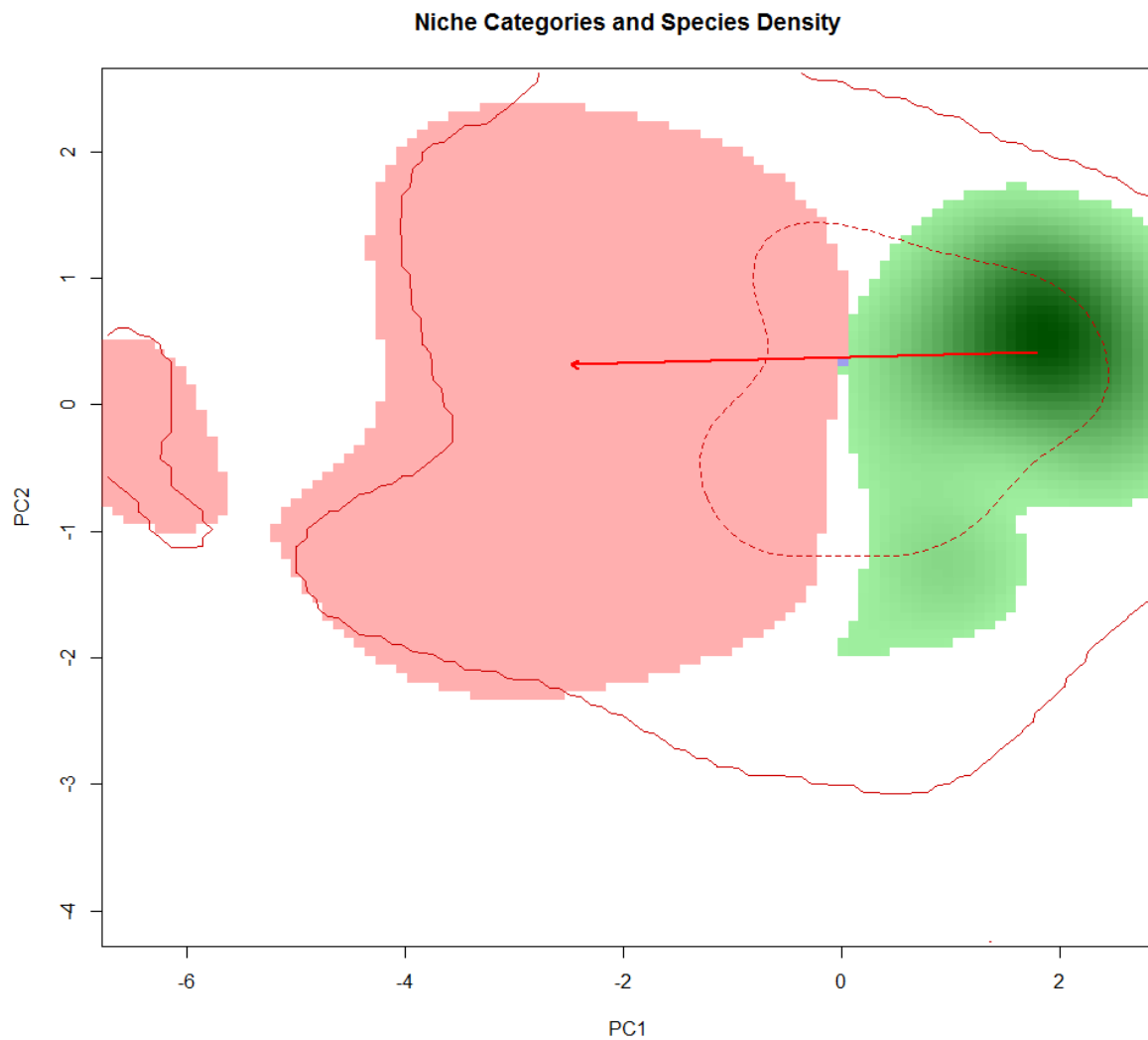
```
> ecospat.plot.niche.dyn(z1.dyn, z2.dyn, title="Niche Categories and Species Density", quant=0.75)
```



Green pixels indicate the unfilled niche (native only), blue pixels the stable niche (common between native and invasive) and red pixels the expansion of the niche (invasive only).

```
## Draw Centroid Arrows  
## ecospat.fun.arrows
```

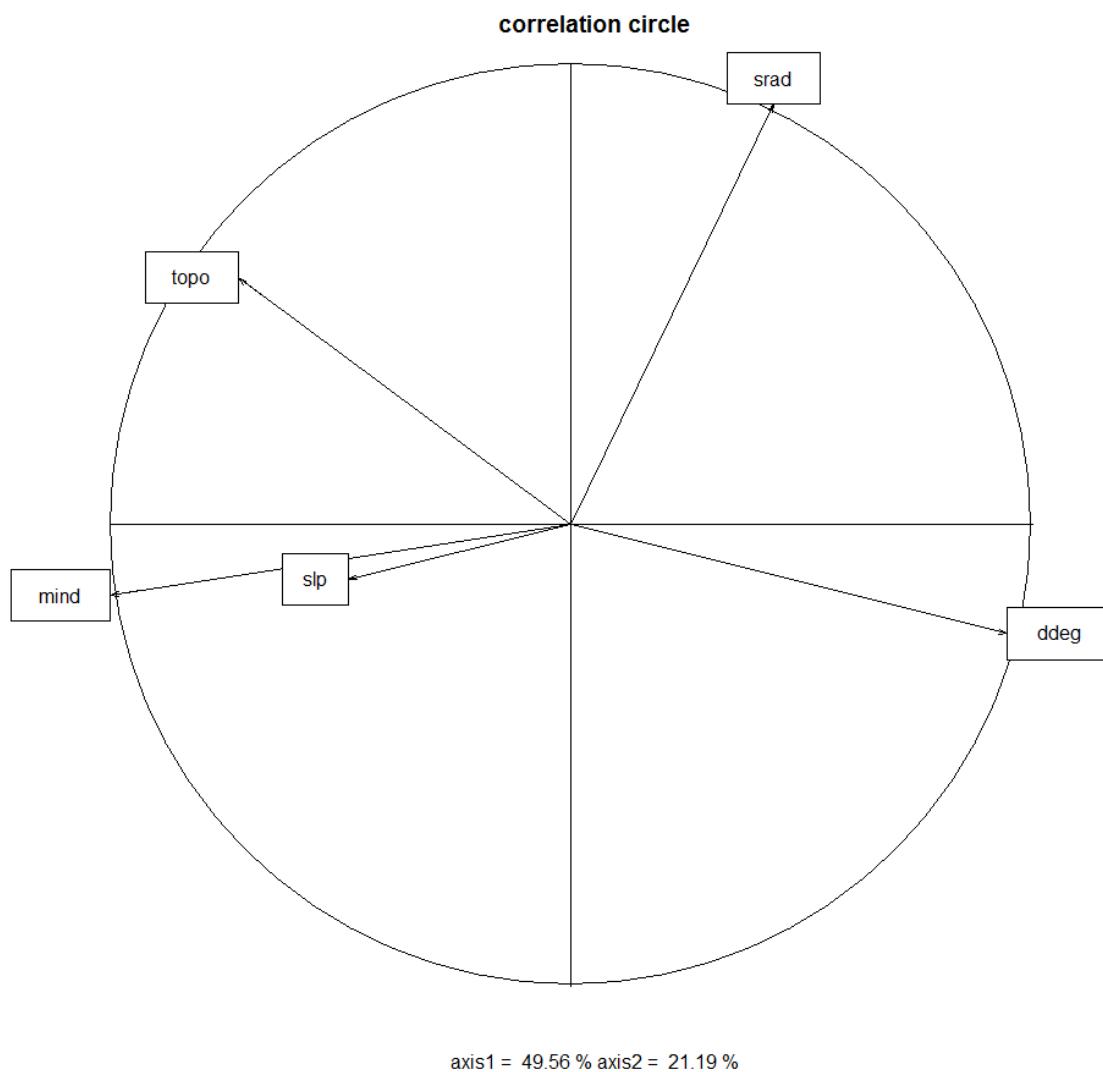
```
> ecospat.fun.arrows (scores.sp1, scores.sp2, scores.clim, scores.clim)
```



The arrow indicates the change in the centroid of the niche.

Plot Variables Contribution `ecospat.plot.contrib`

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



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

```
> ecospat.niche.overlap (z1=z1.dyn, z2=z2.dyn, cor=TRUE)
```

\$D

```
[1] 2.409473e-05
```

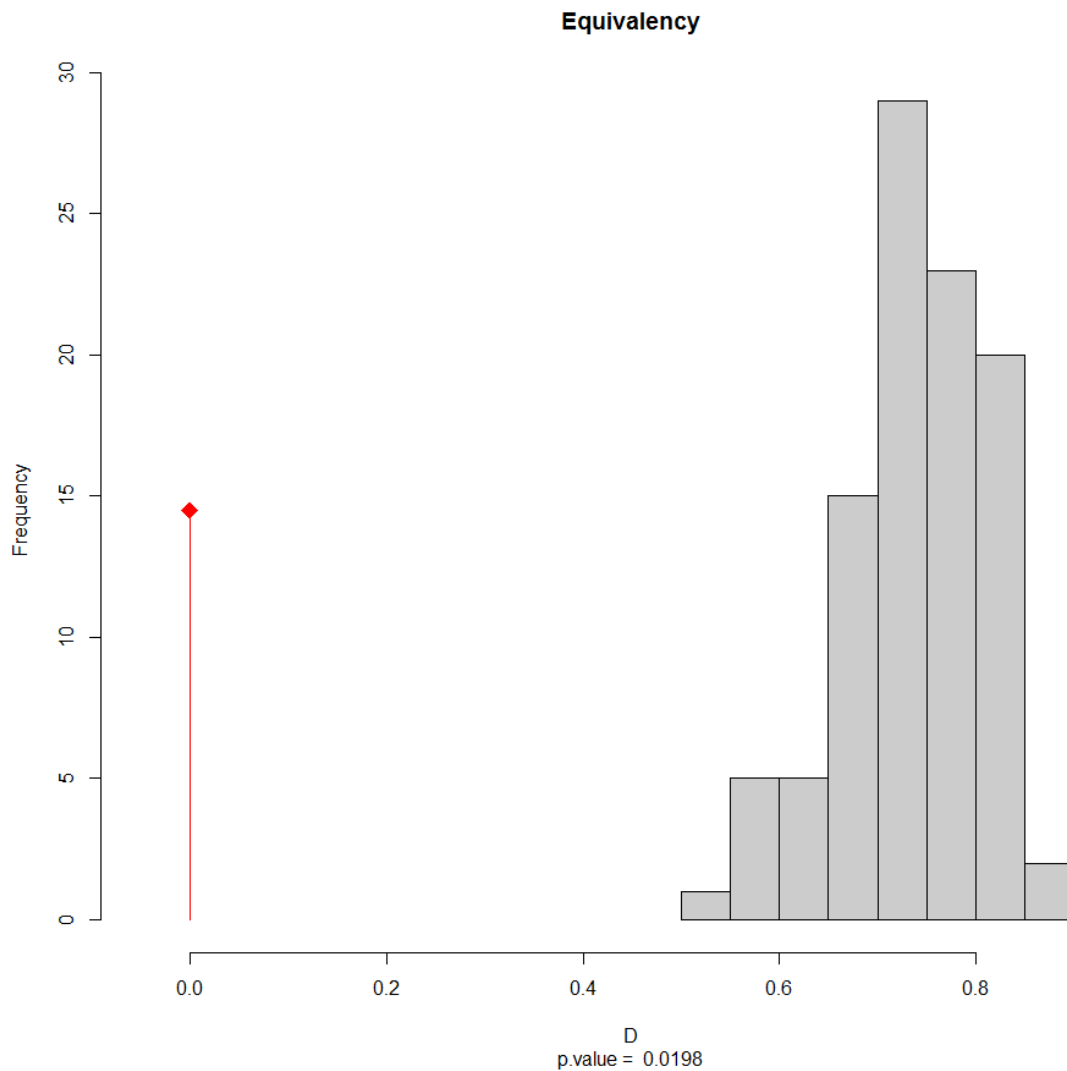
\$I

```
[1] 0.2929149
```

Plot Overlap Test `ecospat.plot.overlap.test`

Equivalency

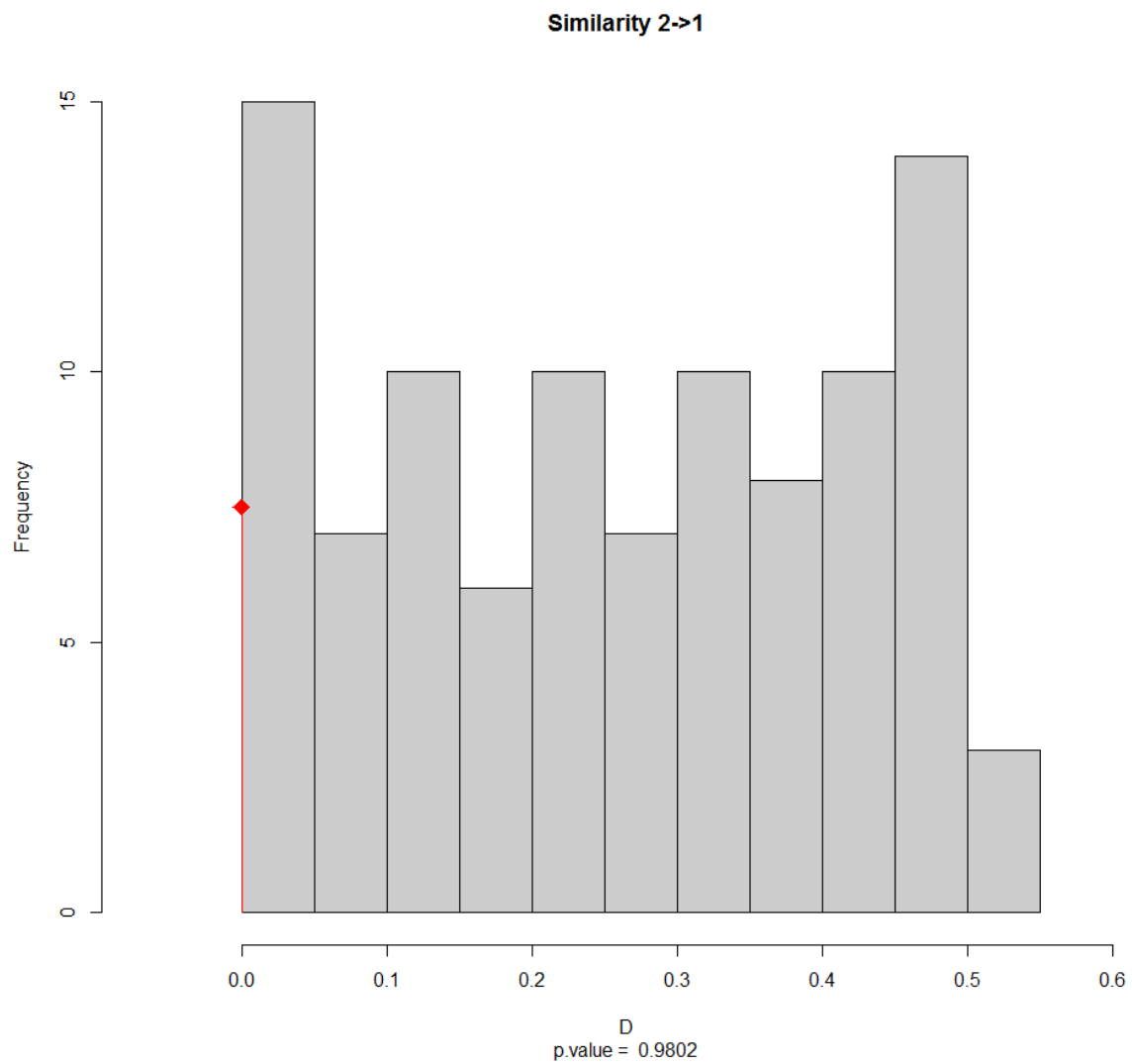
```
> ecospat.plot.overlap.test(a.dyn, "D", "Equivalency")
```



The plot indicates that the observed overlap is lower than 95% of simulated overlaps. The assumption of niche equivalency is thus rejected.

Similarity 2->1

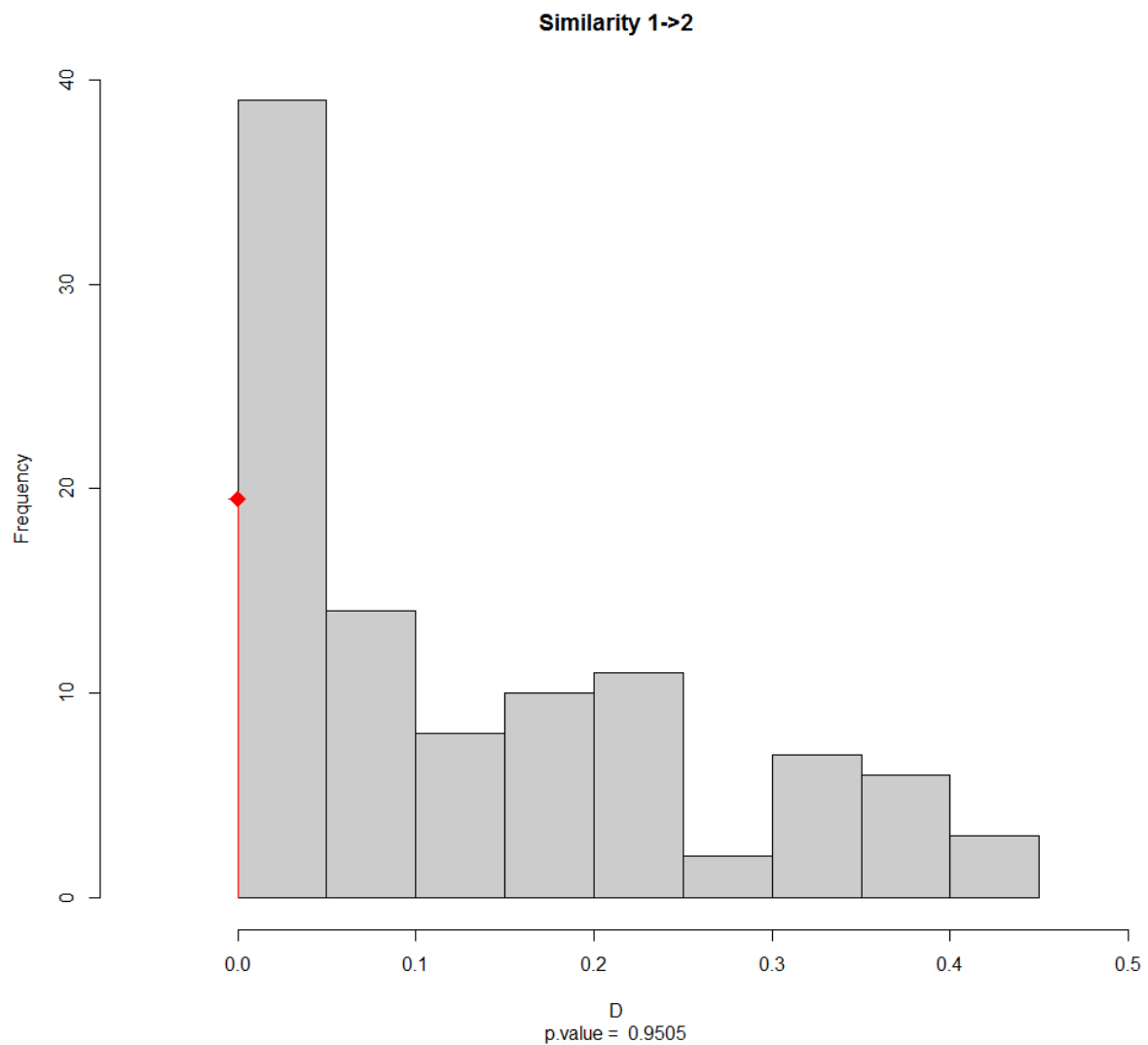
```
> ecospat.plot.overlap.test(b.dyn, "D", "Similarity 2->1")
```



The plot indicates that the observed overlap is within 95% of simulated overlaps. The assumption of niche similarity thus cannot be rejected.

Similarity 1->2

```
> ecospat.plot.overlap.test(b2.dyn, "D", "Similarity 1->2")
```



The plot indicates that the observed overlap is within 95% of simulated overlaps. The assumption of niche similarity thus cannot be rejected.

Niche Expansion, Stability, and Unfilling `ecospat.niche.dyn.index`

```
R=10 #resolution of grid = 10
```

```
> test<-ecospat.niche.dyn.index (z1.dyn, z2.dyn, intersection=NA)  
> test
```

\$dyn

	[, 1]	[, 2]	[, 3]	[, 4]	[, 5]	[, 6]	[, 7]	[, 8]	[, 9]	[, 10]
[1,]	0	0	0	0	-1	-1	-1	0	0	0
[2,]	0	0	0	-1	-1	0	0	0	0	0
[3,]	0	0	-1	-1	-1	-1	-1	-1	-1	0
[4,]	0	0	-1	-1	-1	-1	-1	-1	-1	0
[5,]	0	0	-1	-1	-1	-1	-1	-1	-1	0
[6,]	0	0	0	-1	-1	-1	-1	-1	-1	0
[7,]	0	0	1	2	-1	2	2	-1	0	0
[8,]	0	0	0	1	1	1	1	1	0	0
[9,]	0	0	0	0	1	1	1	1	0	0
[10,]	0	0	0	0	1	1	1	1	0	0

\$dynamic.index.w

expansion	stability	unfilling
0.96058755	0.03941245	0.95792798

3 ECOLOGICAL NICHE MODELLING

3.1 Data Preparation for Modelling

3.1.1 Calibration and Evaluation Dataset

Calibration and Evaluation Dataset `ecospat.caleval`

```
> caleval <- ecospat.caleval (data = ecospat.testData[9], xy = data[2:3],  
                             row.num = 1:nrow(data), nrep = 2, ratio = 0.7,  
                             disaggregate = 0.2, pseudoabs = 100, npres = 10,  
                             replace = F)
```

```
> caleval
```

Seval	yeval	yeval
1	NA	105
2	NA	122
3	NA	NA
4	4	214
5	100	134
6	293	200
7	155	239
8	36	290
9	30	222
10	140	199
11	250	184
12	224	260
13	272	171
14	246	150
15	115	279
16	154	234
17	245	33
18	249	16
19	79	241
20	292	228
21	139	229
22	85	286
23	223	278
24	186	255
25	110	14
26	269	220
27	205	152
28	37	24
29	5	296
30	51	198
31	31	18
32	182	140
33	34	230

```
$cal
```

ycal	ycal
1	NA
2	102
3	63
4	NA
5	NA
6	91
7	146
8	113
9	185
10	232
11	238
12	244
13	44
14	178

15	188	5
16	192	147
17	168	205
18	94	53
19	11	169
20	203	238
21	296	219
22	289	157
23	254	95
24	300	188
25	253	248
26	243	4
27	106	20
28	152	123
29	14	30
30	57	115
31	180	253
32	67	121
33	49	22
34	221	288
35	24	133
36	220	271
37	27	193
38	256	267
39	181	182
40	262	240
41	251	177
42	242	116
43	294	154
44	235	56
45	237	264
46	114	252
47	261	281
48	240	204
49	198	235
50	255	8
51	145	294
52	276	178
53	15	225
54	283	273
55	268	249
56	236	2
57	116	49
58	43	272
59	45	23
60	201	299
61	247	263
62	295	233
63	206	231
64	17	269
65	258	291
66	157	120
67	156	274
68	210	266
69	217	259
70	3	270
71	275	236
72	120	221
73	18	196
74	230	223
75	75	212
76	84	265
77	166	297

3.1.2 Sample Pseudo Absences

Sample Pseudo Absences with `ecospat.rand.pseudoabsences`

```
> glob<-ecospat.testData[2:8]
> presence<-ecospat.testData[c(2:3,9)]
> presence<-presence[presence[,3]==1,1:2]

> ecospat.rand.pseudoabsences (nbabsences=10, glob=glob, colxyglob=1:2,
colvar = "all", presence= presence, colxypresence=1:2, mindist=20)
      long      lat      ddeg      mind      srad      slp      topo
113  576027.9  148852.9  1375    634    243876    35    -205
156  573894.0  118674.0  1394   1643    118865    27    -180
180  582675.7  153675.5  1021   1131    204805    8     -145
134  571125.0  136374.6  1523    161    244663    15     21
245  577524.7  122699.5   952   1655    213170    23   -270
193  580599.0  142450.0   999   1340    277033    34     83
232  576053.4  136301.8   778   2054    168960    16    412
293  572325.0  119773.0   845   1960    150244    22    390
94   572950.0  134925.0  1749   -83    246320    12     17
150  579424.7  152150.3  1195   568    271496    29     29
```

3.1.3 Make Data Frame

Create a biomod2-compatible dataframe with `ecospat.makeDataFrame` for *Solanum acaule* and *Bradypus variegatus*

```
> files <- list.files(path=paste(system.file(package="dismo"),
+ '/ex', sep=''), pattern='grd', full.names=TRUE )
> predictors <- raster::stack(files[c(9,1:8)]) #file 9 has more NA values
than the other files, this is why we choose it as the first layer (see ?
randomPoints)
> solanum <- ecospat.makeDataFrame("Solanum acaule", n=5000, expl.var=predictors,
use.gbif=T)
```

Solanum acaule* : 1079 occurrences found
1-1000-1079

```
#####
```

Solanum acaule

```
#####
```

Occurrence data of following species where added from GBIF:

Solanum acaule

solanum acaule acaule BITTER

Solanum acaule Bitter

Solanum acaule f. acaule

Solanum acaule subsp. punae

```
#####
```

Dataframe created with 132 Presence Points and 5000 Pseudo-Absence Points.

```
#####
```

Warning: There are 8 predictor variable pairs with a correlation coefficients of $|r| > 0.7$. Be aware of collinearity!

```
#####
      biome bio1 bio12 bio16 bio17 bio5 bio6 bio7
bio1    0.41
bio12   0.64 0.53
bio16   0.62 0.62 0.93
bio17   0.43 0.13 0.73 0.45
bio5    0.10 0.71 0.14 0.21 0.08
bio6    0.46 0.93 0.63 0.71 0.23 0.42
bio7    0.45 0.61 0.61 0.66 0.29 0.10 0.86
bio8    0.30 0.86 0.36 0.43 0.06 0.76 0.69 0.33
> head(sol anum)
x      y      cell.id PA      Sol anum. acaul e      biome      bio1      bio12
1 -97.25 16.75 8612    1      0      3      189      1230
2 -66.75 -29.25 25785  1      0      7      189      319
3 -69.75 -44.25 31359  1      0      8      79      142
4 -75.75 -14.75 20373  1      0      13     192      2
5 -76.25 35.25 1772    1      0      5      167     1372
6 -56.75 -10.75 18923  1      0      1      251     2097
bio16  bio17  bio5  bio6  bio7  bio8
1    706    30    295    73    222    196
2    182    12    341    22    319    248
3     54    18    218   -33    250    44
4      2     0    286   100    186    228
5    451   282    311    12    299    251
6    986    36    345   152   192    249
```

```
> file <- paste(system.file(package="dismo"), "/ex/bradypus.csv", sep="")
> bradypus <- read.table(file, header=TRUE, sep=',')[,c(2,3,1)]
> head(bradypus)
```

```
      lon      lat      species
1 -65.4000 -10.3833 Bradypus variegatus
2 -65.3833 -10.3833 Bradypus variegatus
3 -65.1333 -16.8000 Bradypus variegatus
4 -63.6667 -17.4500 Bradypus variegatus
5 -63.8500 -17.4000 Bradypus variegatus
6 -64.4167 -16.0000 Bradypus variegatus
```

```
> sol.aca <- sol anum[sol anum$Sol anum. acaul e==1, c('x','y')]
> sol.aca$Spec <- "Sol anum. acaul e"
> col.names(bradypus) <- col.names(sol.aca)
> spec.list <- rbind(bradypus, sol.aca)
> df <- ecospat.makeDataFrame(spec.list, expl.var=predictors, n=5000, use.
gbif=F)
```

```
#####
Bradypus. variegatus
#####
Dataframe created with 94 Presence Points and 5000 Pseudo-Absence Points.
#####

#####
Sol anum. acaul e
#####
Dataframe created with 132 Presence Points and 5000 Pseudo-Absence Points.
#####

#####
Warning: There are 8 predictor variable pairs with a correlation coefficients of |r| > 0.7. Be aware of collinearity!
```

```
#####
      biome bio1 bio12 bio16 bio17 bio5 bio6 bio7
bio1    0.39
bio12   0.63 0.51
bio16   0.62 0.61 0.93
bio17   0.42 0.13 0.75 0.49
bio5    0.09 0.71 0.12 0.19 0.09
bio6    0.46 0.93 0.62 0.70 0.23 0.42
bio7    0.45 0.60 0.61 0.66 0.31 0.12 0.85
bio8    0.28 0.86 0.34 0.43 0.05 0.76 0.70 0.33
> head(df)
      x      y cell.id PA      Bradypus.variegatus
1 -61.25 -23.75 23750 1      0
2 -94.25 33.75 2294 1      0
3 -80.25 34.75 1950 1      0
4 -55.75 -10.25 18739 1      0
5 -87.25 37.75 820 1      0
6 -97.25 16.75 8612 1      0
Sol anum. acaule biome bio1 bio12 bio16 bio17 bio5 bio6
1      0      2      231      631      292      35      350      102
2      0      5      169      1270      386      274      338      -12
3      0      4      160      1179      358      244      321      -9
4      0      1      255      2183      1034      47      357      156
5      0      4      135      1157      350      244      315      -52
6      0      3      189      1230      706      30      295      73
      bio7 bio8
1      248      267
2      350      169
3      330      252
4      201      251
5      367      133
6      222      196

> df.1 <- ecospat.makeDataFrame(spec.list, expl.var=predictors, n=5000 ,
use.gbi f=F, mi ndist=1)

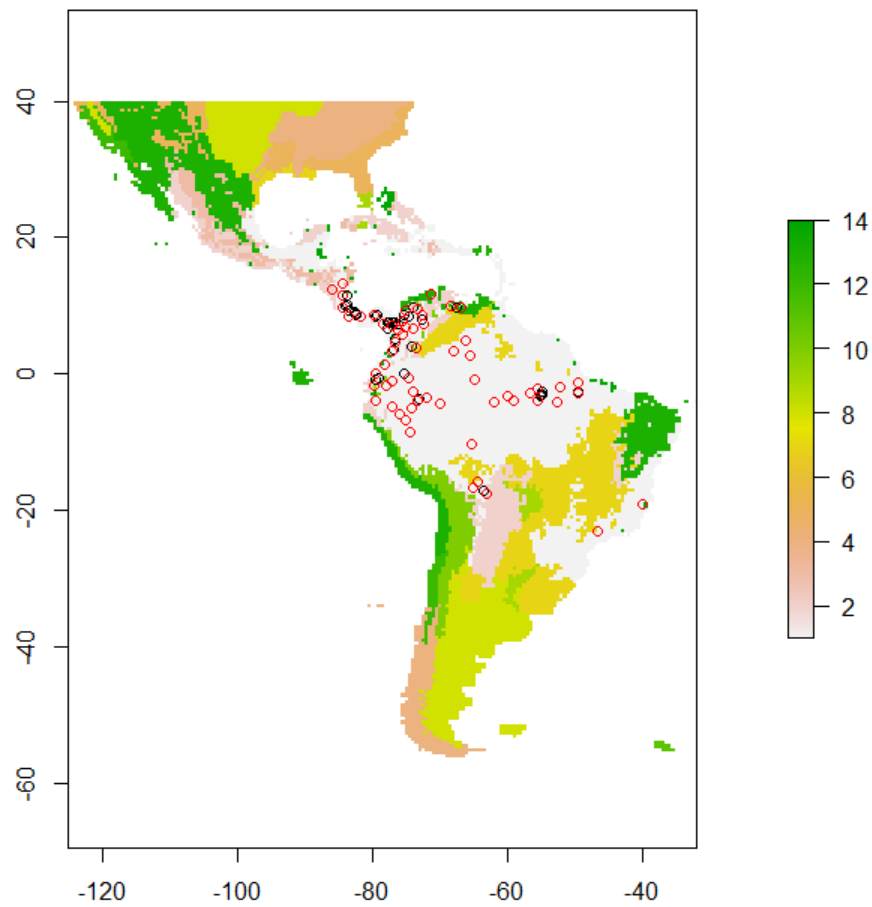
#####
Bradypus.variegatus
#####
Dataframe created with 60 Presence Points and 5000 Pseudo-Absence Points.
#####
Warning: Number of presence points is less than 10 x number of predictors.
Be aware of overparametrization. You only have 60 Presences but 9 predictors.

#####
Sol anum. acaule
#####
Dataframe created with 45 Presence Points and 5000 Pseudo-Absence Points.
#####
Warning: Number of presence points is less than 10 x number of predictors.
. Be aware of overparametrization. You only have 45 Presences but 9 predictors.

#####
Warning: There are 7 predictor variable pairs with a correlation coefficients of |r| > 0.7. Be aware of collinearity!
#####
      biome bio1 bio12 bio16 bio17 bio5 bio6 bio7
bio1    0.40
bio12   0.64 0.54
bio16   0.62 0.63 0.93
bio17   0.44 0.14 0.73 0.47
bio5    0.08 0.69 0.14 0.21 0.09
bio6    0.45 0.93 0.63 0.72 0.23 0.40
bio7    0.45 0.63 0.61 0.66 0.30 0.11 0.87
bio8    0.29 0.85 0.37 0.44 0.06 0.74 0.68 0.34
```



```
> plot(predictors[[1]])
> points(df[df$Bradypus.variegatus==1, c('x', 'y')])
> points(df.1[df.1$Bradypus.variegatus==1, c('x', 'y')], col="red")
```



3.2 Model Calibration

3.2.1 Modelling Techniques - Cross-validation

Cross Validation Example Function `ecospat.cv.example`

```
> ecospat.cv.example ()
Computations 1 for species Agrostis_capillaris is starting now...
.....
GLM
> calibration
GAM
> calibration
GBM
> calibration
Distribution not specified, assuming bernoulli ...
ME
> calibration
RF
> calibration
.....
Computations 2 for species Leontodon_hispidus_sl is starting now...
.....
GLM
> calibration
GAM
> calibration
GBM
> calibration
Distribution not specified, assuming bernoulli ...
ME
> calibration
RF
> calibration
.....
.....
K has been finally set to 10
K has been finally set to 10
K has been finally set to 10
K has been finally set to 10
K has been finally set to 10
```

K-fold and leave-one-out cross validation for GLM with `ecospat.cv.glm`

```
> glm<-ecospat.cv.glm (glm.obj = get ("glm.Agrostis_capillaris", envir=eco
spat.env), K=10, cv.lim=10, jack.knife=F)
K has been finally set to 10

> head(glm)
  id obs predictions
1  1  1    0.5218106
2  2  0    0.2620242
3  3  0    0.1763081
4  4  0    0.2712630
5  5  0    0.2261130
6  6  1    0.3348829
```

Permutation function to get p-values on GLM coefficients and deviance with `ecospat.permut.glm`

```
> ecospat.permut.glm (get ("glm.Agrostis_capillaris", envir=ecospat.env),
1000)
Permutation 100 is starting now...
Permutation 200 is starting now...
Permutation 300 is starting now...
Permutation 400 is starting now...
Permutation 500 is starting now...
Permutation 600 is starting now...
Permutation 700 is starting now...
Permutation 800 is starting now...
Permutation 900 is starting now...
Permutation 1000 is starting now...
...
Computing pvalues by permutations...
...
      pval
Intercept 0.00000
ddeg      0.00000
ddeg^2    0.00000
mi nd     0.00000
mi nd^2   0.00000
slp       0.00000
slp^2     0.00000
topo      0.00000
topo^2    0.00000
D2        0.00094
adj . D2  0.00000
```

Permutation test on the GLM parameters. The function permutes randomly the observations (by default 999 times) and re-calculate some parameters (intercept, coefficients of the predictors, pseudo-R²=deviance D2) for each of the random iterations. Then, the distribution of the ‘random’ parameters is compared to the true parameters. From this, it is possible to derive a p-value, which shows how far a true value for a given parameters deviates from the random distribution. All the parameters are robust and significantly different from the random distribution for ‘glm.Agrostis_capillaris’ (at a 0.05 confidence interval).

K-fold and leave-one-out cross validation for GAM with `ecospat.cv.gam`

```
> gam<-ecospat.cv.gam (gam.obj= get ("gam.Agrostis_capillaris", envir=ecos
pat.env), K=10, cv.lim=10, jack.knife=F)
K has been finally set to 10
> head(gam)
  id obs predictions
1  1  1    0.5157950
2  2  0    0.3548040
3  3  0    0.2422279
4  4  0    0.3803751
5  5  0    0.2419198
6  6  1    0.4553161
```

K-fold and leave-one-out cross validation for GBM with `ecospat.cv.gbm`

```
> gbm<-ecospat.cv.gbm (gbm.obj= get ("gbm.Agrostis_capillaris", envir=ecospat.env), ecospat.testData, K=10, cv.lim=10, jack.knife=F)
K has been finally set to 10
```

```
> head(gbm)
  id obs predictions
1  1  1    0.5489325
2  2  0    0.2905992
3  3  0    0.2614747
4  4  0    0.3128251
5  5  0    0.2652168
6  6  1    0.4580952
```

K-fold and leave-one-out cross validation for Maxent with `ecospat.cv.me`

```
> df.out.cv.me.agroca <- ecospat.cv.me(ecospat.testData, names(ecospat.testData)[9], names(ecospat.testData)[4:8], K = 10, cv.lim = 10, jack.knife = F)
K has been finally set to 10
```

```
> head(df.out.cv.me.agroca)
  id obs predictions
1  1  1    0.421
2  2  0    0.335
3  3  0    0.289
4  4  0    0.354
5  5  0    0.344
6  6  1    0.455
```

K-fold and leave-one-out cross validation for Random Forest with `ecospat.cv.rf`

```
> rf <- ecospat.cv.rf(get("rf.Agrostis_capillaris", envir = ecospat.env), ecospat.testData[, c(9, 4:8)], K = 10, cv.lim = 10, jack.knife = F)
K has been finally set to 10
```

```
> head(rf)
  id obs predictions
1  1  1    0.389
2  2  0    0.081
3  3  0    0.139
4  4  0    0.185
5  5  0    0.116
6  6  1    0.484
```

Variance partitioning for binomial GLM or GAM based on the deviance of two groups or predicting variables with `ecospat.varpart`

```
> ecospat.varpart (model.1= get ("glm.Agrostis_capillaris", envir=ecospat.env), model.2= get ("glm.Leontodon_hispidus_sl", envir=ecospat.env), model.12= get ("glm.Leontodon_hispidus_sl", envir=ecospat.env))
```

	Deviance
D2. model. 1	0.000
D2. model. 2	-0.275
D2. Joined. Contrib. 1. 2.	0.420
D2. Unexplained	0.855

3.2.2 Variables Importance

Calculate the importance of variables for Maxent in the same way Biomod does, with the function `ecospat.maxentvarimport`

```
> model <- get ("me.Leontodon_hispidus_sl", envir=ecospat.env)
> dfvar <- ecospat.testData[4:8]
> nperm <- 5
> ecospat.maxentvarimport (model, dfvar, nperm)
[1] "ddeg"
[1] "mi nd"
[1] "srad"
[1] "slp"
[1] "topo"
```

3.2.3 Model Averaging and Ensemble Forecasting

Calculate the mean (or weighted mean) of several predictions with `ecospat.Epred`

```
> x <- ecospat.testData[c(78,82)] #A dataframe object with SDM predictions
> mean<-ecospat.Epred (x, w=rep(1,ncol(x)), th=0.5)
> head(mean)
      glm_Pri tzel ago_al pi na_sstr  gbm_Pri tzel ago_al pi na_sstr  E
[1, ]                0                0  0
[2, ]                0                0  0
[3, ]                0                0  0
[4, ]                0                0  0
[5, ]                0                0  0
[6, ]                0                0  0

> tail(mean)
      glm_Pri tzel ago_al pi na_sstr  gbm_Pri tzel ago_al pi na_sstr  E
[295, ]                0                0  0
[296, ]                0                0  0
[297, ]                0                0  0
[298, ]                0                0  0
[299, ]               1000             1000 1000
[300, ]                0                0  0
```

3.3 Model Evaluation

3.3.1 Presence-only Evaluation Indices- Boyce Index

Calculate the Boyce index with `ecospat.boyce`

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

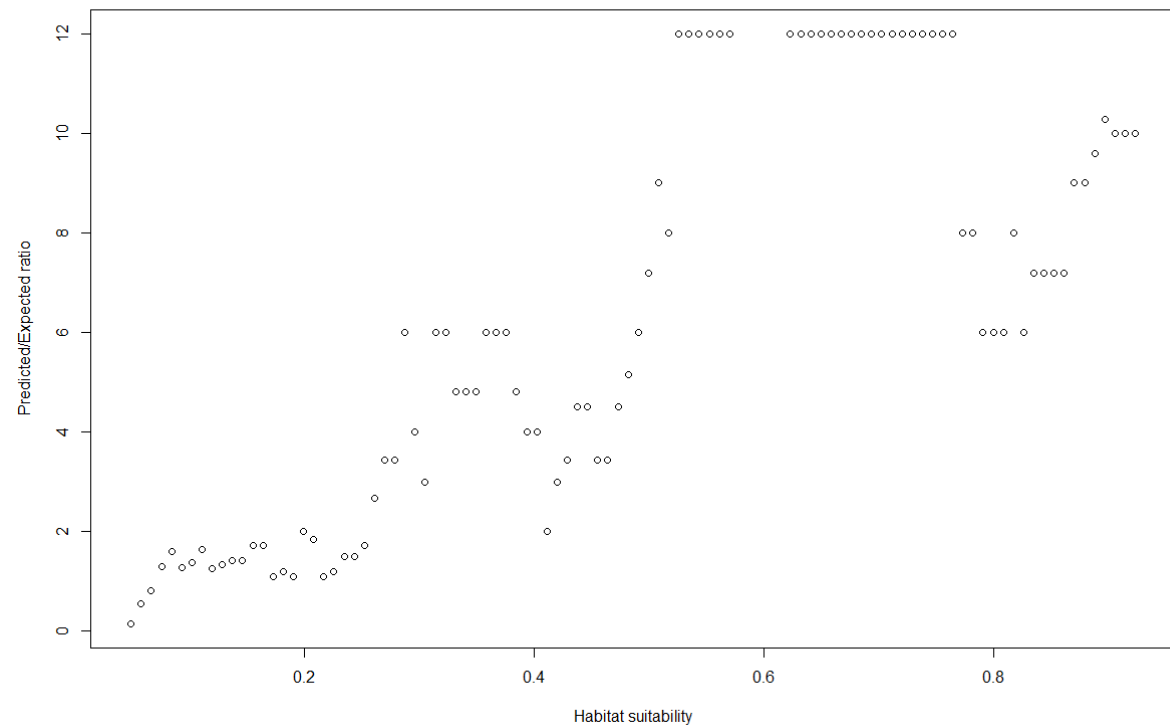
```
> ecospat.boyce (fit = ecospat.testData$glm_Saxifraga_oppositifolia , obs, nclass=0, window.w="default", res=100, PPlot=T)
```

```
$F.ratio
 [1] 0.1457490 0.5454545 0.8181818 1.2972973 1.6000000
 [6] 1.2857143 1.3846154 1.6363636 1.2631579 1.3333333
[11] 1.4117647 1.4117647 1.7142857 1.7142857 1.0909091
[16] 1.2000000 1.0909091 2.0000000 1.8461538 1.0909091
[21] 1.2000000 1.5000000 1.5000000 1.7142857 2.6666667
[26] 3.4285714 3.4285714 6.0000000 4.0000000 3.0000000
[31] 6.0000000 6.0000000 4.8000000 4.8000000 4.8000000
[36] 6.0000000 6.0000000 6.0000000 4.8000000 4.0000000
[41] 4.0000000 2.0000000 3.0000000 3.4285714 4.5000000
[46] 4.5000000 3.4285714 3.4285714 4.5000000 5.1428571
[51] 6.0000000 7.2000000 9.0000000 8.0000000 12.0000000
[56] 12.0000000 12.0000000 12.0000000 12.0000000 12.0000000
[61]      NaN      NaN      NaN      NaN      NaN
[66] 12.0000000 12.0000000 12.0000000 12.0000000 12.0000000
[71] 12.0000000 12.0000000 12.0000000 12.0000000 12.0000000
[76] 12.0000000 12.0000000 12.0000000 12.0000000 12.0000000
[81] 12.0000000 12.0000000 8.0000000 8.0000000 6.0000000
[86] 6.0000000 6.0000000 8.0000000 6.0000000 7.2000000
[91] 7.2000000 7.2000000 7.2000000 9.0000000 9.0000000
[96] 9.6000000 10.2857143 10.0000000 10.0000000 10.0000000
[101]      NaN
```

```
$Pearson.cor
 [1] 0.819
```

```
$HS
 [1] 0.0490350 0.0578613 0.0666876 0.0755139 0.0843402
 [6] 0.0931665 0.1019928 0.1108191 0.1196454 0.1284717
[11] 0.1372980 0.1461243 0.1549506 0.1637769 0.1726032
[16] 0.1814295 0.1902558 0.1990821 0.2079084 0.2167347
[21] 0.2255610 0.2343873 0.2432136 0.2520399 0.2608662
[26] 0.2696925 0.2785188 0.2873451 0.2961714 0.3049977
[31] 0.3138240 0.3226503 0.3314766 0.3403029 0.3491292
[36] 0.3579555 0.3667818 0.3756081 0.3844344 0.3932607
[41] 0.4020870 0.4109133 0.4197396 0.4285659 0.4373922
[46] 0.4462185 0.4550448 0.4638711 0.4726974 0.4815237
[51] 0.4903500 0.4991763 0.5080026 0.5168289 0.5256552
[56] 0.5344815 0.5433078 0.5521341 0.5609604 0.5697867
[61] 0.5786130 0.5874393 0.5962656 0.6050919 0.6139182
[66] 0.6227445 0.6315708 0.6403971 0.6492234 0.6580497
[71] 0.6668760 0.6757023 0.6845286 0.6933549 0.7021812
[76] 0.7110075 0.7198338 0.7286601 0.7374864 0.7463127
[81] 0.7551390 0.7639653 0.7727916 0.7816179 0.7904442
[86] 0.7992705 0.8080968 0.8169231 0.8257494 0.8345757
[91] 0.8434020 0.8522283 0.8610546 0.8698809 0.8787072
[96] 0.8875335 0.8963598 0.9051861 0.9140124 0.9228387
[101] 1.9316650
```

SID																	
[1]	1	2	3	4	5	6	7	8	9	10	11	12	13	14			
[15]	15	16	17	18	19	20	21	22	23	24	25	26	27	28			
[29]	29	30	31	32	33	34	35	36	37	38	39	40	41	42			
[43]	43	44	45	46	47	48	49	50	51	52	53	54	55	56			
[57]	57	58	59	60	61	62	63	64	65	66	67	68	69	70			
[71]	71	72	73	74	75	76	77	78	79	80	81	82	83	84			
[85]	85	86	87	88	89	90	91	92	93	94	95	96	97	98			
[99]	99	100	101														



Here the boyce index is 0.819. If the rank of predicted expected ratio would be completely ordered along habitat suitability axis then boyce index would be 1.

3.3.2 Accuracy of Community Prediction

Indices of accuracy of community predictions `ecospat.SSDMeval`

```
> eval<-ecospat.testData[c(9: 24, 53, 55, 56, 58) ]
> pred<-ecospat.testData[c(59: 78) ]
> ecospat.SSDMeval (eval, pred, proba=T, ntir=2)
```

	deviati on. ri ch. pred. 1	deviati on. ri ch. pred. 2	overpredi cti on. 1			
1	-2	-3	0.2352941			
2	-5	-8	0.3125000			
3	-6	-4	0.4666667			
4	-4	-3	0.3333333			
5	-9	-9	0.5000000			
6	0	-3	0.0000000			
	overpredi cti on. 2	underpredi cti on. 1	underpredi cti on. 2	predi cti on. success.		
1						
1	0.2352941	0.6666667	0.3333333	0.7		
0						
2	0.5000000	0.0000000	0.0000000	0.7		
5						
3	0.3333333	0.2000000	0.2000000	0.6		
0						
4	0.2666667	0.2000000	0.2000000	0.7		
0						
5	0.5000000	0.0000000	0.0000000	0.5		
5						
6	0.3000000	0.0000000	0.0000000	1.0		
0						
	predi cti on. success. 2	sensi ti vi ty. 1	sensi ti vi ty. 2	speci fi ci ty. 1		
1	0.75	0.2000000	0.3333333	0.8666667		
2	0.60	0.4444444	0.3333333	1.0000000		
3	0.70	0.3636364	0.4444444	0.8888889		
4	0.75	0.4444444	0.5000000	0.9090909		
5	0.55	0.1818182	0.1818182	1.0000000		
6	0.85	1.0000000	0.7692308	1.0000000		
	speci fi ci ty. 2	kappa. 1	kappa. 2	TSS. 1	TSS. 2	simi la ri ty. 1
1	0.9285714	0.07692308	0.3055556	0.06666667	0.2619048	0.2500000
2	1.0000000	0.46808511	0.2857143	0.44444444	0.3333333	0.6153846
3	0.9090909	0.23809524	0.3684211	0.25252525	0.3535354	0.5000000
4	0.9166667	0.36842105	0.4444444	0.35353535	0.4166667	0.5714286
5	1.0000000	0.16666667	0.1666667	0.18181818	0.1818182	0.3076923
6	1.0000000	1.0000000	0.7000000	1.00000000	0.7692308	1.0000000
	simi la ri ty. 2	Jaccard. 1	Jaccard. 2			
1	0.4444444	0.1428571	0.2857143			
2	0.5000000	0.4444444	0.3333333			
3	0.5714286	0.3333333	0.4000000			
4	0.6153846	0.4000000	0.4444444			
5	0.3076923	0.1818182	0.1818182			
6	0.8695652	1.0000000	0.7692308			

3.4 Spatial Predictions and Projections

3.4.1 Model Binarization

Generate a binary model from an original biomod2 model
ecospat.binary.model

```
> library(biomod2)

#Run biomod2 to produce a model prediction
> DataSpecies <- read.csv(system.file("external/species/mammals_table.csv",
, package="biomod2"))
>
> myRespName <- 'GuloGulo'
> # the presence/absences data for our species
> myResp <- as.numeric(DataSpecies[, myRespName])
> # the XY coordinates of species data
> myRespXY <- DataSpecies[, c("X_WGS84", "Y_WGS84")]
> # load the environmental raster layers (could be .img, ArcGIS
> # rasters or any supported format by the raster package)
> # Environmental variables extracted from Worldclim (bio_3, bio_4,
> # bio_7, bio_11 & bio_12)
> myExpl = stack( system.file( "external/bioclim/current/bio3.grd",
+                             package="biomod2"),
+                 system.file( "external/bioclim/current/bio4.grd",
+                             package="biomod2"),
+                 system.file( "external/bioclim/current/bio7.grd",
+                             package="biomod2"),
+                 system.file( "external/bioclim/current/bio11.grd",
+                             package="biomod2"),
+                 system.file( "external/bioclim/current/bio12.grd",
+                             package="biomod2"))
>
> myBiomodData <- BIOMOD_FormatingData(resp.var = myResp,
+                                     expl.var = myExpl,
+                                     resp.xy = myRespXY,
+                                     resp.name = myRespName)

----- GuloGulo Data Formating -----
> No pseudo absences selection !
! No data has been set aside for modeling evaluation
----- Done -----

> myBiomodData
----- 'BIOMOD.formated.data' -----

sp. name = GuloGulo

661 presences, 1827 true absences and 0 undifined points in dataset

5 explanatory variables

      bio3      bio4      bio7      bio11
Min.   : 10.19  Min.   : 72    Min.   : 54.5  Min.   : -447.75
1st Qu.: 21.22  1st Qu.: 2641   1st Qu.: 186.0  1st Qu.: -184.32
Median : 35.00  Median : 6682   Median : 306.2  Median : 24.23
Mean   : 40.29  Mean   : 7358   Mean   : 310.9  Mean   : -2.64
3rd Qu.: 56.35  3rd Qu.: 11752  3rd Qu.: 424.6  3rd Qu.: 196.30
Max.   : 92.00  Max.   : 22314  Max.   : 718.0  Max.   : 283.00

      bio12
Min.   : 0.028
1st Qu.: 276.493
Median : 562.931
Mean   : 853.516
3rd Qu.: 1200.592
Max.   : 5431.002
-----
```

```

> myBi omodOpti on <- BI OMOD_ModelingOptions()
>
>
> myBi omodModel Out <- BI OMOD_Modeling(
+ myBi omodData,
+ models = c(' GLM' ),
+ models. options = myBi omodOpti on,
+ NbRunEval=1,
+ DataSpl it=80,
+ Prevalence=0. 5,
+ VarImport=3,
+ models. eval. meth = c(' TSS', ' ROC' ),
+ SaveObj = TRUE,
+ rescal. all. models = TRUE,
+ do. full. models = FALSE,
+ modeling. id = paste(myRespName, "FirstModeling", sep=""))

```

Loading required library...

Checking Models arguments...

Creating suitable Workdir...

> Automatic weights creation to rise a 0.5 prevalence

----- Gul oGul o Modeling Summary -----

5 environmental variables (bio3 bio4 bio7 bio11 bio12)
Number of evaluation repetitions : 1
Models selected : GLM

Total number of model runs : 1

----- Run : Gul oGul o_All Data

----- Gul oGul o_All Data_RUN1

Model=Surface Range Envelop
Evaluating Model stuff...
Evaluating Predictor Contributions...

----- Done -----

```

>
>
> myBi omodModel Out

```

----- BI OMOD. models. out -----

Modeling id : Gul oGul oFirstModeling

Species modeled : Gul oGul o

Considered variables : bio3 bio4 bio7 bio11 bio12

Computed Models : Gul oGul o_All Data_RUN1_GLM

Failed Models : none

```

> myBi omodModel Eval <- get_eval uati ons(myBi omodModel Out)
>
>
> myBi omodEM <- BI OMOD_EnsembleModeling(
+   modeling. output = myBi omodModel Out,
+   chosen. models = 'all',
+   em. by='all',
+   eval. metric = c('TSS'),
+   eval. metric. quality. threshold = c(0.7),
+   prob. mean = T,
+   prob. cv = T,
+   prob. ci = T,
+   prob. ci. alpha = 0.05,
+   prob. median = T,
+   commi ttee. averaging = T,
+   prob. mean. weight = T,
+   prob. mean. weight. decay = 'proportional' )

----- Build Ensemble Models -----

! all models available will be included in ensemble.modeling
> Evaluation & Weighting methods summary :
    TSS over 0.7

> Total Consensus ensemble modeling
! Models projections for whole zonation required...
  > Projecting Gul oGul o_AllData_RUN1_GLM ...
> Mean of probabilities...
    Evaluating Model stuff...
> Coef of variation of probabilities...
    Evaluating Model stuff...
> Confidence Interval...
    Evaluating Model stuff...
    Evaluating Model stuff...
> Median of ptobabilities...
    Evaluating Model stuff...
> Comi ttee averaging...
    Evaluating Model stuff...
> Prababilities wegthing mean...
    Evaluating Model stuff...
----- Done -----
> myBi omodEM

----- ' BI OMOD. EnsembleModeling. out' -----

sp. name : Gul oGul o

expl. var. names : bi o3 bi o4 bi o7 bi o11 bi o12

models computed:
Gul oGul o_Total Consensus_TSS_EMmean, Gul oGul o_Total Consensus_TSS_EMcv, Gul o
Gul o_Total Consensus_TSS_EMci Inf, Gul oGul o_Total Consensus_TSS_EMci Sup, Gul o
Gul o_Total Consensus_TSS_EMmedian, Gul oGul o_Total Consensus_TSS_EMca, Gul oGu
lo_Total Consensus_TSS_EMwmean

-----
>
> myBi omodProj <- BI OMOD_Projection(
+   modeling. output = myBi omodModel Out,
+   new. env = myExpl,
+   proj. name = 'current',
+   selected. models = 'all',
+   binary. meth = 'TSS',
+   compress = 'xz',
+   clamping. mask = F,
+   output. format = '.grd')

```

```

===== Do Models Projections =====
=
> Building clamping mask
> Projecting GuloGulo_AllData_RUN1_GLM ...
> Building TSS binaries
===== Done =====

> myBiomodProj

===== 'BIOMOD.projection.out' =====

Projection directory : GuloGulo/current

sp.name : GuloGulo

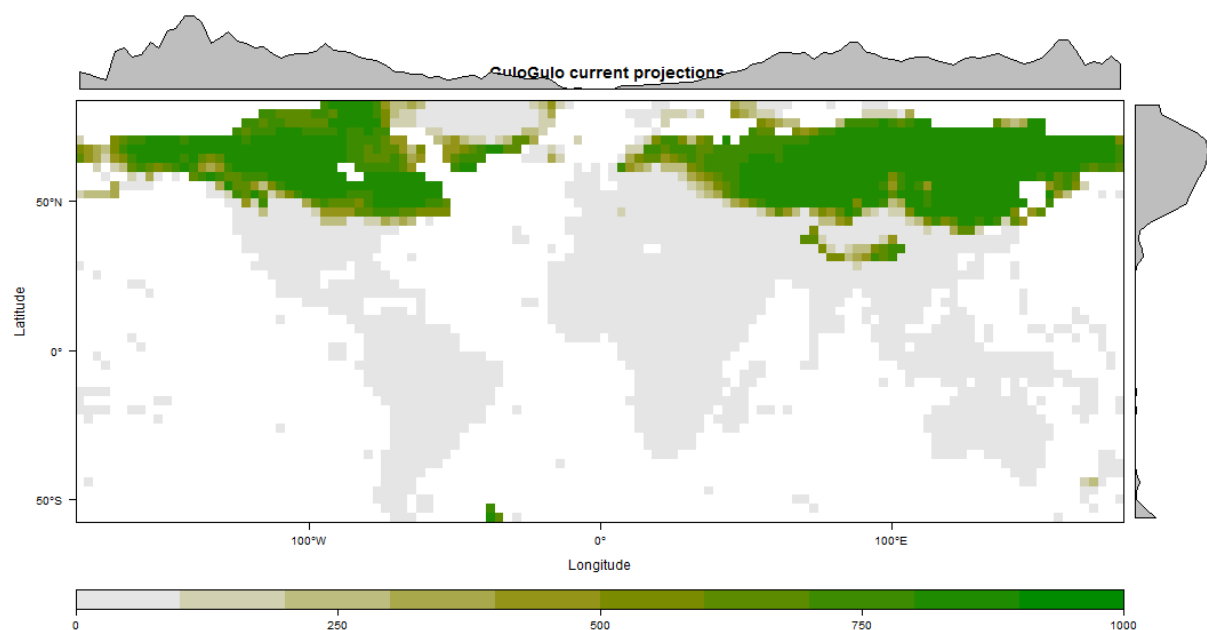
expl.var.names : bio3 bio4 bio7 bio11 bio12

modeling.id : GuloGuloFirstModeling (
GuloGulo/GuloGulo.GuloGuloFirstModeling.models.out )

models.projected : GuloGulo_AllData_RUN1_GLM

=====
=
> plot(myBiomodProj, str.grep = 'GLM')

```

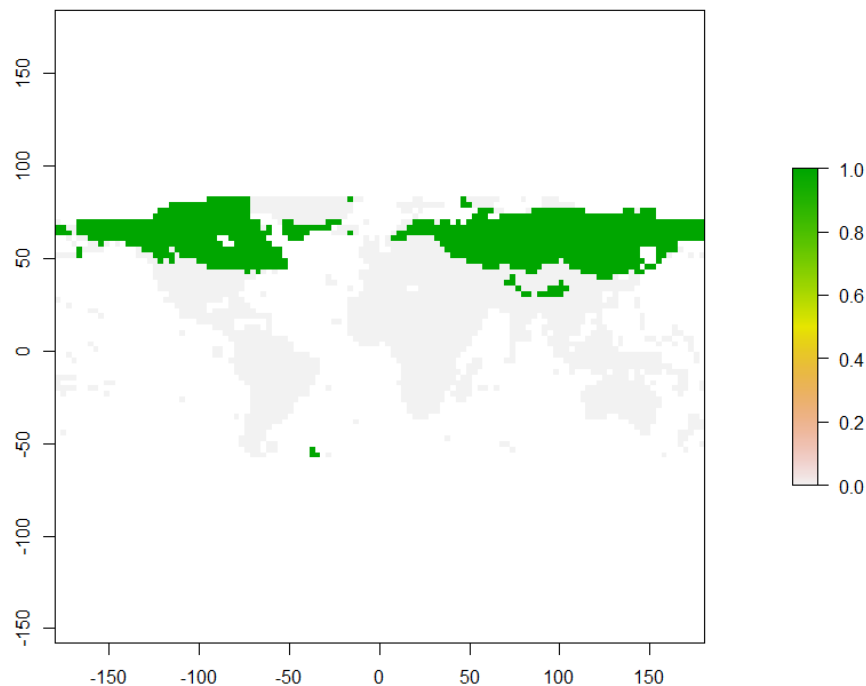


```

> Pred <- get_predictions(myBiomodProj)
> Sp.occ.xy <- DataSpecies[DataSpecies[, 5]==1, 2:3]
> Percentage <- 7

> binary.model <- ecospat.binary.model (Pred, Sp.occ.xy, Percentage)
> plot(binary.model)

```



3.4.2 Implementing species dispersal into projections of species distribution models

Implementation of species-specific dispersal constraints into projections of species distribution models under environmental change and/or landscape fragmentation scenarios with `ecospat.migclim`

```

> ecospat.migclim()
load the MigClim package
initializing ...
... done

```

4 COMPLEMENTARY ANALYSIS

4.1 Biotic Interactions

4.1.1 Co-occurrence pattern

Calculate an index of species co-occurrences where species were predicted as present with `ecospat.co_occurrences`

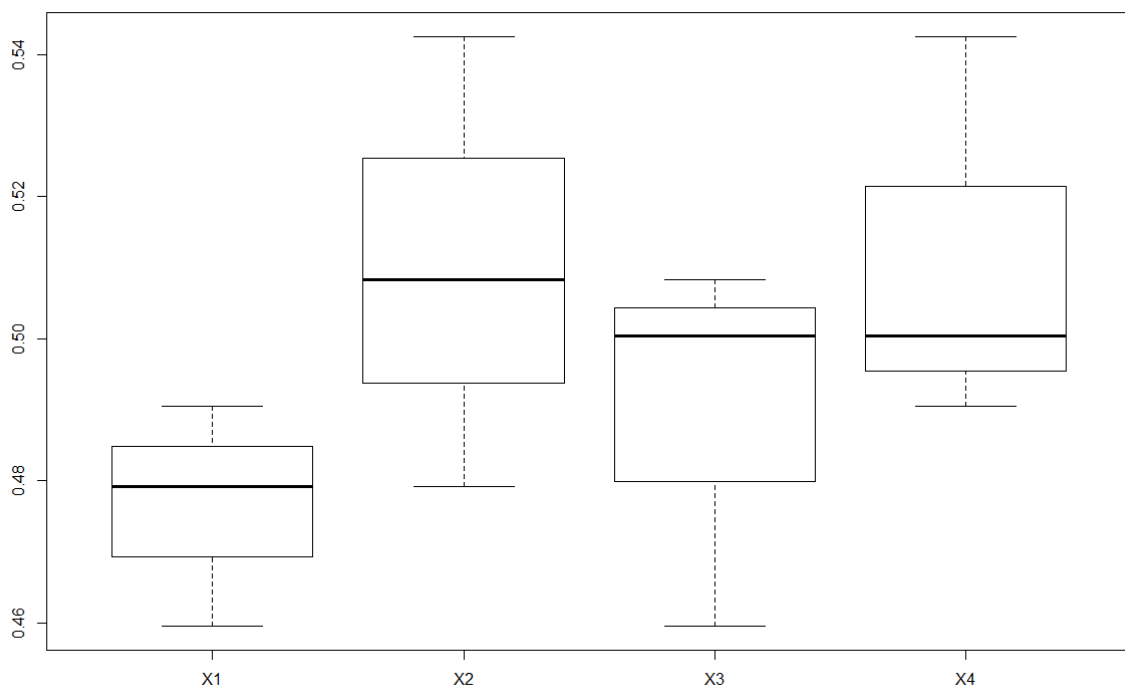
```
> test<-ecospat.testData[c(59:66)]
> ecospat.co_occurrences(test)
```

	glm_Geranium_sylvaticum	glm_Ranunculus_acris_sl
glm_Geranium_sylvaticum	0.4523111	0.4791825
glm_Ranunculus_acris_sl	0.4791825	0.5533304
glm_Prunella_vulgaris	0.4595555	0.5083275
glm_Veronica_chamaedrys	0.4905250	0.5425753
	glm_Prunella_vulgaris	glm_Veronica_chamaedrys
glm_Geranium_sylvaticum	0.4595555	0.4905250
glm_Ranunculus_acris_sl	0.5083275	0.5425753
glm_Prunella_vulgaris	0.4708349	0.5004026
glm_Veronica_chamaedrys	0.5004026	0.5503356

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.

$$\text{Ind}_{\infty} = \frac{N_{(S1 \cap S2)}}{\text{Min}(N_{S1}, N_{S2})},$$

where $N_{(S1 \cap S2)}$ is the number of times species S1 and S2 co-occur, while $\text{Min}(N_{S1}, N_{S2})$ is the number of times species S1 and S2 co-occur, while is the occurrence frequency of the rarest of the two species.



4.1.2 Co-occurrence analysis Environmentally Constrained Null Models

Co-occurrence Analysis & Environmentally Constrained Null Models

`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.

```
> presence<- ecospat.testData[ c(9: 24) ]
> pred<- ecospat.testData[65: 82]
> nbpermut<- 10000
> outpath<-getwd()

> ecospat.cons_Cscore(presence, pred, nbpermut, outpath)
Computing observed co-occurrence matrix
.....
.....
Computing permutations
.....
.....
Permutations finished Fri Jun 20 16:32:31 2014
.....
.....
Exporting dataset
.....
.....
$ObsCscoreTot
[1] 3308.25

$SimCscoreTot
[1] 4876.848

$PVal.less
[1] 9.999e-05

$PVal.greater
[1] 1

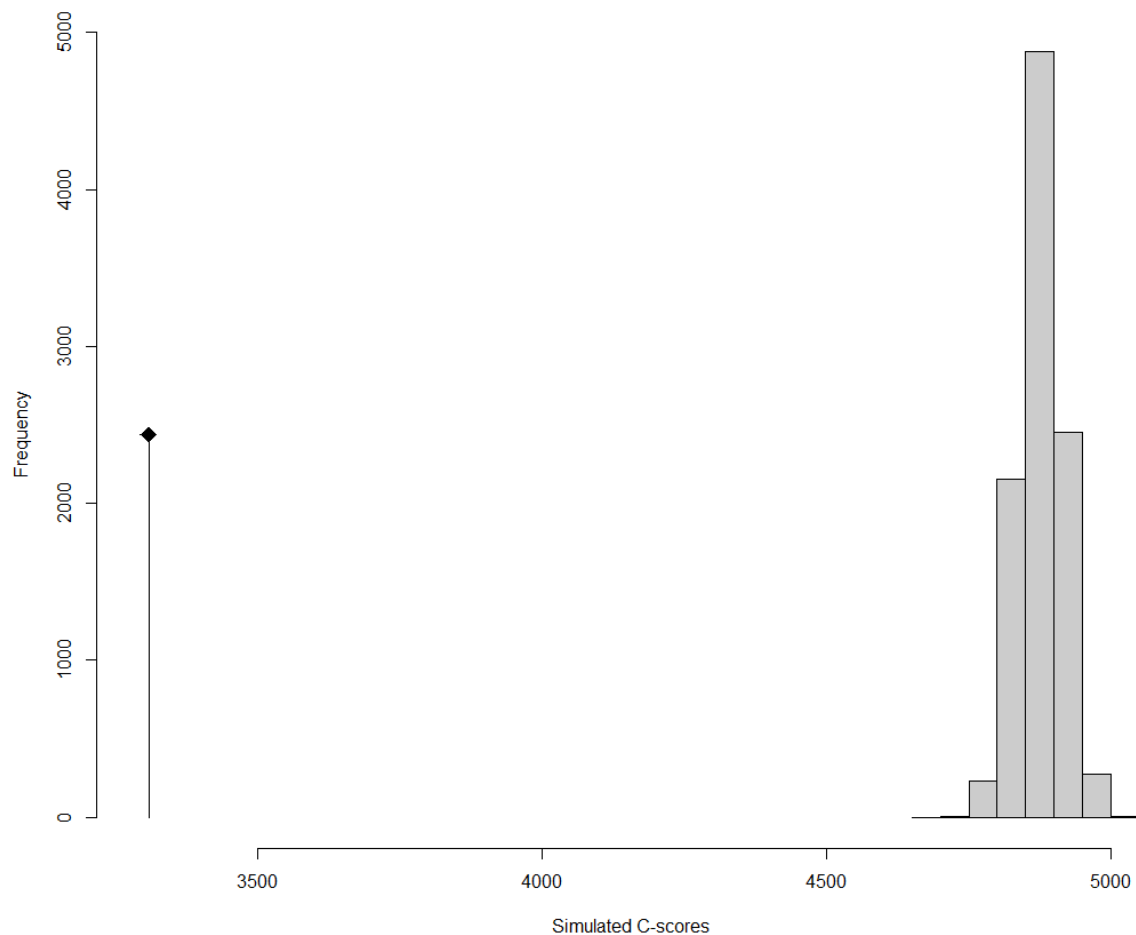
$SES.Tot
[1] -40.77995
```

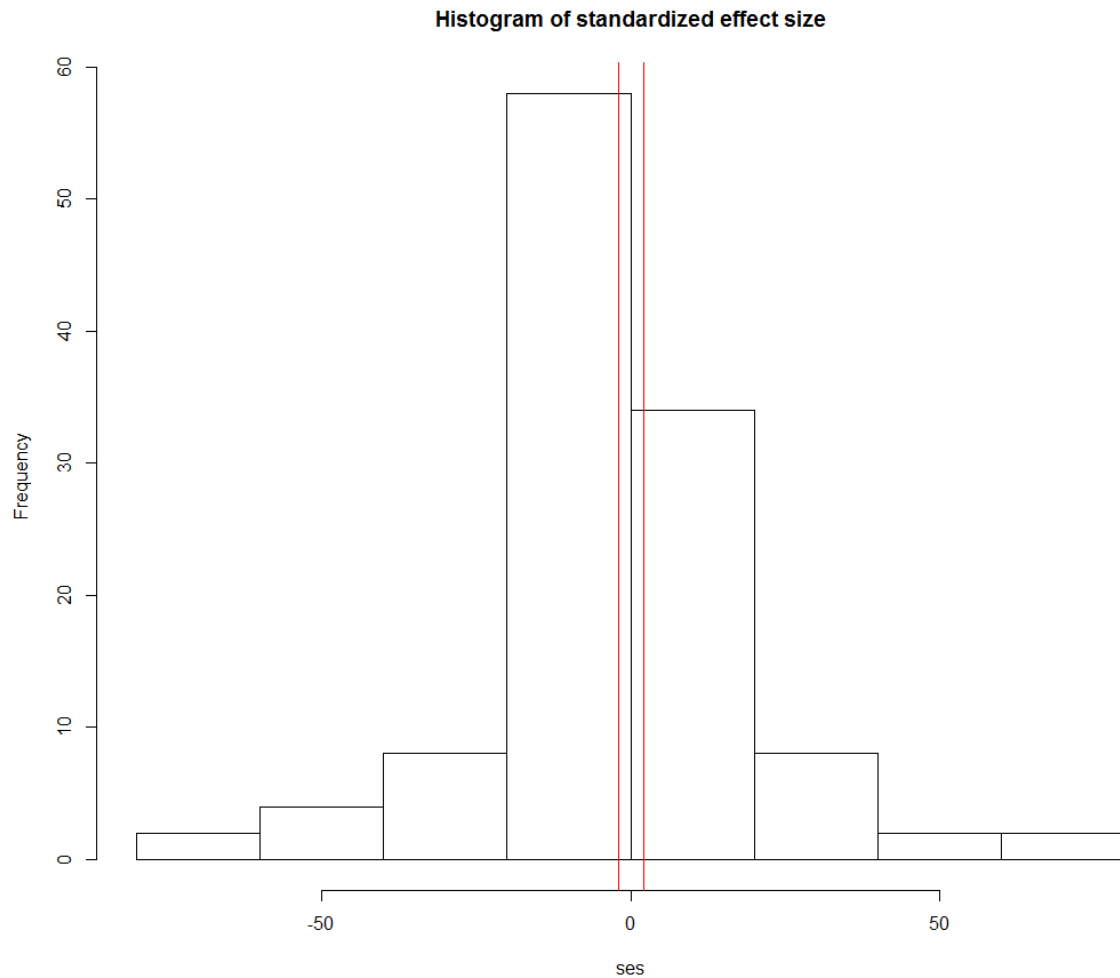
The function returns

- 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 was 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).





```
> Cscores<- read.table("Signifi c_const_Cscores.txt", head=T)
> head(Cscores)
```

Col	Row	Sps1	Sps2	C. score0bs
1	1	2 Agrostis_capillaris	Leontodon_hispidus_sl	2970
2	1	6 Agrostis_capillaris	Ranunculus_acris_sl	2080
3	1	7 Agrostis_capillaris	Prunella_vulgaris	1176
4	1	9 Agrostis_capillaris	Taraxacum_officinale_aggr	3200
5	1	10 Agrostis_capillaris	Plantago_lanceolata	2409
6	1	12 Agrostis_capillaris	Carex sempervirens	5936
C. scoreExp p. less p. greater ses				
1	2173.144	0.98500150	0.02739726	2.215496
2	8917.397	0.00009999	1.00000000	-9.982066
3	7231.134	0.00009999	1.00000000	-9.497185
4	8838.164	0.00009999	1.00000000	-8.505857
5	1728.877	0.98000200	0.03859614	2.059305
6	9973.942	0.00009999	1.00000000	-7.753379