Package 'ecospat'

November 7, 2022

Version 3.4

Date 2022-11-04

Title Spatial Ecology Miscellaneous Methods

Author Olivier Broennimann [cre, aut, ctb],
Valeria Di Cola [aut, ctb],
Blaise Petitpierre [ctb],
Frank Breiner [ctb],
Daniel Scherrer [ctb],
Manuela D`Amen [ctb],
Christophe Randin [ctb],
Robin Engler [ctb],
Wim Hordijk [ctb],
Heidi Mod [ctb],
Julien Pottier [ctb],
Mirko Di Febbraro [ctb],
Loic Pellissier [ctb],
Dorothea Pio [ctb],
Ruben Garcia Mateo [ctb],
Anne Dubuis [ctb],
Luigi Maiorano [ctb],
Achilleas Psomas [ctb],
Charlotte Ndiribe [ctb],
Nicolas Salamin [ctb],
Niklaus Zimmermann [ctb],
Flavien Collart [ctb],
Antoine Guisan [aut]

Maintainer Olivier Broennimann <olivier.broennimann@unil.ch>

VignetteBuilder knitr

Depends R (>= 4.0.0)

Imports ade4 (>= 1.6-2), gbm (>= 2.1.1), ape (>= 3.2), sp (>= 1.0-15), foreach (>= 1.4.3),adehabitatHR (>= 0.4.11), adehabitatMA (>= 0.3.8), biomod2 (>= 4.1), dismo (>= 0.9-3), ecodist (>= 1.2.9), maptools (>= 0.8-39), randomForest (>= 4.6-7), raster (>= 2.5-8), gtools (>= 3.4.1), PresenceAbsence (>= 1.1.9), methods

(>= 3.1.1), parallel, classInt (>= 0.1-23), vegan (>= 2.4-1), poibin (>= 1.3), matrixStats (>= 0.53.1), ks (>= 1.12.0), nabor (>= 0.5.0), Hmisc (>= 4.4-2)

Suggests rgdal (>= 1.2-15), rJava (>= 0.9-6), XML (>= 3.98-1.1), knitr (>= 1.14), alphahull (>= 2.1), rgeos (>= 0.3-28), igraph (>= 1.2.1), markdown (>= 1.1), dplyr (>= 0.3.0.1), rmarkdown (>= 2.6), qpdf (>= 1.1), snowfall (>= 1.61), maps (>= 3.0.0), rms

LazyData true

URL http://www.unil.ch/ecospat/home/menuguid/ecospat-resources/tools.html

Description Collection of R functions and data sets for the support of spatial ecology analyses with a focus on pre, core and post modelling analyses of species distribution, niche quantification and community assembly. Written by current and former members and collaborators of the ecospat group of Antoine Guisan, Department of Ecology and Evolution (DEE) and Institute of Earth Surface Dynamics (IDYST), University of Lausanne, Switzerland. Read Di Cola et al. (2016) <doi:10.1111/ecog.02671> for details.

License GPL

BugReports https://github.com/ecospat/ecospat

RoxygenNote 7.2.1

NeedsCompilation no

Repository CRAN

Date/Publication 2022-11-07 17:40:02 UTC

R topics documented:

ecospat-package 4
ecospat.adj.D2.glm 5
ecospat.binary.model 6
ecospat.boyce 8
ecospat.calculate.pd 9
ecospat.caleval 11
ecospat.CCV.communityEvaluation.bin 12
ecospat.CCV.communityEvaluation.prob 14
ecospat.CCV.createDataSplitTable 17
ecospat.CCV.modeling 18
ecospat.climan 22
ecospat.cohen.kappa 23
ecospat.CommunityEval 24
ecospat.cons_Cscore 25
ecospat.cor.plot 27
ecospat.co_occurrences 27
ecospat.Cscore 28
ecospat.cv.gbm 30
ecospat.cv.glm 31
ecospat.cv.me 33

ecospat.cv.rf . . . 34
ecospat.Epred . . . 35
ecospat.ESM.EnsembleEvaluation . . . 36
ecospat.ESM.EnsembleModeling . . . 38
ecospat.ESM.EnsembleProjection . . . 40
ecospat.ESM.Modeling . . . 41
ecospat.ESM.Projection . . . 44
ecospat.ESM.responsePlot . . . 46
ecospat.ESM.threshold . . . 47
ecospat.ESM.VarContrib . . . 48
ecospat.grid.clim.dyn . . . 49
ecospat.makeDataFrame . . . 52
ecospat.mantel.correlogram . . . 54
ecospat.margin . . . 56
ecospat.max.kappa . . . 58
ecospat.max.tss . . . 59
ecospat.maxentvarimport . . . 60
ecospat.mdr . . . 61
ecospat.mess . . . 63
ecospat.meva.table . . . 64
ecospat.mpa . . . 65
ecospat.niche.dyn.index . . . 66
ecospat.niche.dynIndexProjGeo . . . 67
ecospat.niche.equivalency.test . . . 68
ecospat.niche.overlap . . . 70
ecospat.niche.similarity.test . . . 71
ecospat.niche.zProjGeo . . . 73
ecospat.nicheNBmean . . . 75
ecospat.nichePOSNB . . . 76
ecospat.npred . . . 77
ecospat.occ.desaggregation . . . 78
ecospat.occupied.patch . . . 79
ecospat.permut.glm . . . 81
ecospat.plot.contrib . . . 82
ecospat.plot.kappa . . . 83
ecospat.plot.mess . . . 84
ecospat.plot.niche . . . 85
ecospat.plot.niche.dyn . . . 86
ecospat.plot.overlap.test . . . 87
ecospat.plot.tss . . . 88
ecospat.poolingEvaluation . . . 89
ecospat.rand.pseudoabsences . . . 91
ecospat.rangesize . . . 92
ecospat.rcls.grd . . . 97
ecospat.recstrat_prop . . . 98
ecospat.recstrat_regl . . . 99
ecospat.sample.envar . . . 100
ecospat.SESAM.prr . . . 101

ecospat.shift.centroids 102
ecospat.testData 103
ecospat.testEnvRaster 107
ecospat.testMdr 108
ecospat.testNiche 109
ecospat.testNiche.inv 110
ecospat.testNiche.nat 111
ecospat.testNichePOSNB 112
ecospat.testTree 113
ecospat.varpart 114

Index **116**

`ecospat-package` *Spatial Ecology Miscellaneous Methods*

Description

Collection of methods, utilities and data sets for the support of spatial ecology analyses with a focus on pre-, core and post- modelling analyses of species distribution, niche quantification and community assembly. The `ecospat` package was 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.

Pre-modelling:

- Spatial autocorrelation:`ecospat.mantel.correlogram`
- Variable selection: `ecospat.npred`
- Climate Analalogy: `ecospat.climan`, `ecospat.mess`, `ecospat.plot.mess`
- Phylogenetic diversity measures: `ecospat.calculate.pd`
- Biotic Interactions: `ecospat.cons_Cscore`, `ecospat.Cscore`, `ecospat.co_occurrences`
- Minimum Dispersal routes: `ecospat.mdr`
- Niche Quantification: `ecospat.grid.clim.dyn`, `ecospat.niche.equivalency.test`, `ecospat.niche.similarit` `ecospat.plot.niche`, `ecospat.plot.niche.dyn`, `ecospat.plot.contrib`, `ecospat.niche.overlap`, `ecospat.plot.overlap.test`, `ecospat.niche.dyn.index`, `ecospat.shift.centroids`, `ecospat.niche.dynIndexProjGeo`, `ecospat.niche.zProjGeo`, `ecospat.margin`
- Data Preparation: `ecospat.caleval`, `ecospat.cor.plot`, `ecospat.makeDataFrame`, `ecospat.occ.desaggregati` `ecospat.rand.pseudoabsences`, `ecospat.rcls.grd`, `ecospat.recstrat_prop`, `ecospat.recstrat_regl`, `ecospat.sample.envar`

Core Niche Modelling:

- Model evaluation: `ecospat.cv.glm`, `ecospat.permut.glm`, `ecospat.cv.gbm`, `ecospat.cv.me`, `ecospat.cv.rf`, `ecospat.boyce`, `ecospat.CommunityEval`, `ecospat.cohen.kappa`, `ecospat.max.kappa`, `ecospat.max.tss`, `ecospat.meva.table`, `ecospat.plot.kappa`, `ecospat.plot.tss`, `ecospat.adj.D2.glm`, `ecospat.CCV.createDataSplitTable`, `ecospat.CCV.modeling`, `ecospat.CCV.communityEvaluation.bin`, `ecospat.CCV.communityEvaluation.prob`

- Spatial predictions and projections: ecospat.ESM.Modeling, ecospat.ESM.EnsembleModeling, ecospat.ESM.Projection, ecospat.ESM.EnsembleProjection, ecospat.SESAM.prr, ecospat.binary.model, ecospat.Epred, ecospat.mpa
- Variable Importance: ecospat.maxentvarimport, ecospat.ESM.VarContrib

Post Modelling:

- Variance Partition: ecospat.varpart
- Spatial predictions of species assemblages: ecospat.cons_Cscore
- Range size quantification: ecospat.rangesize, ecospat.occupied.patch

---

`ecospat.adj.D2.glm` *Calculate An Adjusted D2*

Description

This function is used for calculating an adjusted D2 from a calibrated GLM object

Usage

```
ecospat.adj.D2.glm(glm.obj)
```

Arguments

`glm.obj`	Any calibrated GLM object with a binomial error distribution

Details

This function takes a calibrated GLM object with a binomial error distribution and returns an evaluation of the model fit. The measure of the fit of the models is expressed as the percentage of explained deviance adjusted by the number of degrees of freedom used (similar to the adjusted-R2 in the case of Least-Square regression; see Weisberg 1980) and is called the adjusted-D2 (see guisan and Zimmermann 2000 for details on its calculation).

Value

Returns an adjusted D square value (proportion of deviance accounted for by the model).

Author(s)

Christophe Randin <christophe.randin@unibas.ch> and Antoine Guisan <antoine.guisan@unil.ch>

References

Weisberg, S. 1980. Applied linear regression. Wiley.

Guisan, A., S.B. Weiss and A.D. Weiss. 1999. GLM versus CCA spatial modeling of plant species distribution. *Plant Ecology*, **143**, 107-122.

Guisan, A. and N.E. Zimmermann. 2000. Predictive habitat distribution models in ecology. *Ecol. Model.*, **135**, 147-186.

Examples

```
data(ecospat.testData)
glm.obj<-glm(Achillea_millefolium~ddeg+mind+srad+slp+topo,
family = binomial, data=ecospat.testData)

ecospat.adj.D2.glm(glm.obj)
```

`ecospat.binary.model` *Generate Binary Models*

Description

Generate a binary map from a continuous model prediction.

Usage

```
ecospat.binary.model (Pred, Threshold)
```

Arguments

`Pred`	RasterLayer predicted suitabilities from a SDM prediction.
`Threshold`	A threshold to convert continous maps into binary maps (e.g. the output of the function `ecospat.mpa()` or use the `optimal.thresholds` from Presence-Absence R package.

Details

This function generates a binary model prediction (presence/absence) from an original model applying a threshold. The threshold could be arbitrary, or be based on the maximum acceptable error of false negatives (i.e. percentage of the presence predicted as absences, omission error).

Value

The binary model prediction (presence/absence).

Author(s)

Ruben G. Mateo <rubeng.mateo@gmail.com> with contributions of Frank Breiner <frank.breiner@wsl.ch>

References

Fielding, A.H. and J.F. Bell. 1997. A review of methods for the assessment of prediction errors in conservation presence/absence models. *Environmental Conservation*, **24**: 38-49.

Engler, R., A Guisan and L. Rechsteiner. 2004. An improved approach for predicting the distribution of rare and endangered species from occurrence and pseudo-absence data. *Journal of Applied Ecology*, **41**, 263-274.

Liu, C., Berry, P. M., Dawson, T. P. and R. G. Pearson. 2005. Selecting thresholds of occurrence in the prediction of species distributions. *Ecography*, **28**, 385-393.

Jimenez-Valverde, A. and J.M.Lobo. 2007. Threshold criteria for conversion of probability of species presence to either-or presence-absence. *Acta oecologica*, **31**, 361-369.

Liu, C., White, M. and G. Newell. 2013. Selecting thresholds for the prediction of species occurrence with presence-only data. *J. Biogeogr.*, **40**, 778-789.

Freeman, E.A. and G.G. Moisen. 2008. A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa. *Ecological Modelling*, **217**, 48-58.

See Also

ecospat.mpa, optimal.thresholds

Examples

```
library(raster)
library(dismo)

### make a maxent model

# path to maxent.jar file
path<- paste0(system.file(package="dismo"), "/java/maxent.jar")

if (file.exists(path) & require(rJava)) {

  # get predictor variables
  fnames <- list.files(path=paste(system.file(package="dismo"), '/ex', sep=''),
                   pattern='grd', full.names=TRUE )
  predictors <- stack(fnames)
  #plot(predictors)

  # file with presence points
  occurence <- paste(system.file(package="dismo"), '/ex/bradypus.csv', sep='')
  occ <- read.table(occurence, header=TRUE, sep=',')[,-1]
  colnames(occ) <- c("x","y")
  occ <- ecospat.occ.desaggregation(occ,min.dist=1)

  # fit a domain model, biome is a categorical variable
  me <- maxent(predictors, occ, factors='biome')

  # predict to entire dataset
```

```
  pred <- predict(me, predictors)

  plot(pred)
  points(occ)
}

### convert suitability to binary map

mpa.cutoff <- ecospat.mpa(pred,occ,perc = 0.9) # 90 percent of the presences encompassed
pred.bin.mpa <- ecospat.binary.model(pred,mpa.cutoff)

plot(pred.bin.mpa)
points(occ)
```

`ecospat.boyce` *Calculate Boyce Index*

Description

Calculate the Boyce index as in Hirzel et al. (2006). The Boyce index is used to assess model performance.

Usage

```
ecospat.boyce (fit, obs, nclass=0, window.w="default", res=100, PEplot = TRUE,
rm.duplicate = TRUE, method = 'spearman')
```

Arguments

`fit`	A vector or Raster-Layer containing the predicted suitability values
`obs`	A vector containing the predicted suitability values or xy-coordinates (if "fit" is a Raster-Layer) of the validation points (presence records)
`nclass`	The number of classes or vector with class thresholds. If `nclass=0`, the Boyce index is calculated with a moving window (see next parameters)
`window.w`	The width of the moving window (by default 1/10 of the suitability range)
`res`	The resolution of the moving window (by default 100 focals)
`PEplot`	If `TRUE`, plot the predicted to expected ratio along the suitability class
`rm.duplicate`	If `TRUE`, only one of the successive duplicated P/E values are used to compute the correlation. See Details
`method`	Method used to compute the correlation. The default is `'spearman'` but `'pearson'` and `'kendall'` can also be used. See Details
	rm.duplicate = TRUE, method = 'spearman'

Details

The Boyce index only requires presences and measures how much model predictions differ from random distribution of the observed presences across the prediction gradients (Boyce et al. 2002). It is thus an appropriate metric in the case of presence-only models. It is continuous and varies between -1 and +1. Positive values indicate a model which present predictions are consistent with the distribution of presences in the evaluation dataset, values close to zero mean that the model is not different from a random model, negative values indicate counter predictions, i.e., predicting poor quality areas where presences are more frequent (Hirzel et al. 2006).

Removing the successive duplicated P/E values (`rm.duplicate = TRUE`) focuses more on the discriminative aspect of the predictions, lowering the assessment of the evaluation of the model resolution (sensu Hirzel et al. 2006 p. 150). However, it seems that in the initial version, dupplicated values were not removed.

In the initial publication on the continuous Boyce index, the correlation was set to `method = 'spearman'`. However, using `method = 'kendall'` or `method = 'pearson'` might be more informative about the accuracy of the predictions.

Value

The function returns a list that contains a vector F.ratio (the predicted-to-expected ratio for each class-interval) and a numeric Spearman.cor (the Boyce index value)

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com> and Frank Breiner <frank.breiner@unil.ch>

References

Boyce, M.S., P.R. Vernier, S.E. Nielsen and F.K.A. Schmiegelow. 2002. Evaluating resource selection functions. *Ecol. Model.*, **157**, 281-300.

Hirzel, A.H., G. Le Lay, V. Helfer, C. Randin and A. Guisan. 2006. Evaluating the ability of habitat suitability models to predict species presences. *Ecol. Model.*, **199**, 142-152.

Examples

```
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, PEplot = TRUE)
```

---

ecospat.calculate.pd *Calculate Phylogenetic Diversity Measures*

Description

Calculate all phylogenetic diversity measures listed in Schweiger et al., 2008 (see full reference below).

Usage

```
ecospat.calculate.pd (tree, data, method="spanning", type="clade", root=FALSE,
average=FALSE, verbose=FALSE)
```

Arguments

tree	The phylogenetic tree
data	A presence or absence (binary) matrix for each species (columns) in each location or grid cell (rows)
method	The method to use. Options are "pairwise", "topology", and "spanning". Default is "spanning".
type	Phylogenetic measure from those listed in Schweiger et al 2008. Options are "Q", "P", "W", "clade", "species", "J", "F", "AvTD","TTD", "Dd". Default is "clade".
root	Phylogenetic diversity can either be rooted or unrooted. Details in Schweiger et al 2008. Default is FALSE.
average	Phylogenetic diversity can either be averaged or not averaged. Details in Schweiger et al 2008. Default is FALSE.
verbose	Boolean indicating whether to print progress output during calculation. Default is FALSE.

Details

Given a phylogenetic tree and a presence/absence matrix this script calculates phylogenetic diversity of a group of species across a given set of grid cells or locations. The library "ape" is required to read the tree in R. Command is "read.tree" or "read.nexus". Options of type: "P" is a normalized mearure of "Q". "clade" is "PDnode" when root= FALSE, and is "PDroot" ehn root =TRUE. "species" is "AvPD".

Value

This function returns a list of phylogenetic diversity values for each of the grid cells in the presence/absence matrix

Author(s)

Nicolas Salamin <nicolas.salamin@unil.ch> and Dorothea Pio <Dorothea.Pio@fauna-flora.org>

References

Schweiger, O., S. Klotz, W. Durka and I. Kuhn. 2008. A comparative test of phylogenetic diversity indices. *Oecologia*, **157**, 485-495.

Pio, D.V., O. Broennimann, T.G. Barraclough, G. Reeves, A.G. Rebelo, W. Thuiller, A. Guisan and N. Salamin. 2011. Spatial predictions of phylogenetic diversity in conservation decision making. *Conservation Biology*, **25**, 1229-1239.

Pio, D.V., R. Engler, H.P. Linder, A. Monadjem, F.P.D. Cotterill, P.J. Taylor, M.C. Schoeman, B.W. Price, M.H. Villet, G. Eick, N. Salamin and A. Guisan. 2014. Climate change effects on animal and plant phylogenetic diversity in southern Africa. *Global Change Biology*, **20**, 1538-1549.

Examples

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

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

plot(pd)
```

---

ecospat.caleval *Calibration And Evaluation Dataset*

Description

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

Usage

```
ecospat.caleval (data, xy, row.num=1:nrow(data), nrep=1, ratio=0.7,
disaggregate=0, pseudoabs=0, npres=0, replace=FALSE)
```

Arguments

data	A vector with presence-absence (0-1) data for one species.
xy	The x and y coordinates of the projection dataset.
row.num	Row original number
nrep	Number of repetitions
ratio	Ratio of disaggregation
disaggregate	Minimum distance of disaggregation (has to be in the same scale as xy)
pseudoabs	Number of pseudoabsences
npres	To select a smaller number of presences from the dataset to be subsetted. The maximum number is the total number of presences
replace	F to replace de pseudoabsences

Details

This functions generates two list, one with the calibration or training dataset and other list with the evaluation or testing dataset disaggregated with a minimum distance.

Value

list("eval"=eval,"cal"=cal))

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com>

Examples

```
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)
caleval
```

---

ecospat.CCV.communityEvaluation.bin

Calculates a range of community evaluation metrics based on different thresholding techniques.

Description

The function uses the output of ecospat.CCV.modeling to calculate a range of community evaluation metrics based on a selection of thresholding techniques both for the calibration data and independent evaluation data.

Usage

```
ecospat.CCV.communityEvaluation.bin(ccv.modeling.data,
                                    thresholds= c('MAX.KAPPA', 'MAX.ROC','PS_SDM'),
                                    community.metrics=c('SR.deviation','Sorensen'),
                                        parallel=FALSE,
                                        cpus=4,
                                        fix.threshold=0.5,
                                        MCE=5,
                                        MEM=NULL)
```

Arguments

ccv.modeling.data	a 'ccv.modeling.data' object returned by ecospat.CCV.modeling
thresholds	a selection of thresholds ('FIXED', 'MAX.KAPPA', 'MAX.ACCURACY', 'MAX.TSS', 'SENS_SPEC', 'MAX.ROC', 'OBS.PREVALENCE', 'AVG.PROBABILITY', 'MCE', 'PS_SDM, MEM') to be calculated and applied for the model evaluation.
community.metrics	a selection of community evaluation metrics ('SR.deviation', 'community.AUC', 'community.overprediction' ,'community.underprediction',' community.accuracy', 'community.sensitivity', 'community.specificity', 'community.kappa', 'community.tss', 'Sorensen', 'Jaccard', 'Simpson') to be calculated for each seleted thresholding technique.
parallel	should parallel computing be allowed (TRUE/FALSE)

cpus	number of cpus to use in parallel computing
fix.threshold	fixed threshold to be used. Only gets used if thresholding technique FIXED is selected.
MCE	maximum omission error (%) allowed for the thresholding. Only gets used if thresholding technique MCE is selected.
MEM	a vetor with the species richness prediction of a MEM for each site. Only needed if MEM is selected.

Details

The function uses the probability output of the ecospat.CCV.modeling function and creates binary maps based on the selected thresholding methods. These binary maps are then used to calculate the selected community evaluation metrics both for the calibration and evaluation data of each modeling run.

Value

DataSplitTable	a matrix with TRUE/FALSE for each model run (TRUE=Calibration point, FALSE=Evaluation point)
CommunityEvaluationMetrics.CalibrationSites	a 4-dimensional array containing the community evaluation metrics for the calibartion sites of each run (NA means that the site was used for evaluation)
CommunityEvaluationMetrics.EvaluationSites	a 4-dimensional array containing the community evaluation metrics for the evaluation sites of each run (NA means that the site was used for calibaration)
PA.allSites	a 4-dimensional array of the binary prediction for all sites and runs under the different thresholding appraoches.

Author(s)

Daniel Scherrer <daniel.j.a.scherrer@gmail.com>

References

Scherrer, D., D'Amen, M., Mateo, M.R.G., Fernandes, R.F. & Guisan , A. (2018) How to best threshold and validate stacked species assemblages? Community optimisation might hold the answer. Methods in Ecology and Evolution, in review

See Also

ecospat.CCV.createDataSplitTable; ecospat.CCV.communityEvaluation.prob

Examples

```
#Loading species occurence data and remove empty communities
data(ecospat.testData)
testData <- ecospat.testData[,c(24,34,43,45,48,53,55:58,60:63,65:66,68:71)]
sp.data <- testData[which(rowSums(testData)>0), sort(colnames(testData))]
```

```
#Loading environmental data
env.data <- ecospat.testData[which(rowSums(testData)>0),4:8]

#Coordinates for all sites
xy <- ecospat.testData[which(rowSums(testData)>0),2:3]

#Running all the models for all species
myCCV.Models <- ecospat.CCV.modeling(sp.data = sp.data,
                                     env.data = env.data,
                                     xy = xy,
                                     NbRunEval = 5,
                                     minNbPredictors = 10,
                                     VarImport = 3)

#Thresholding all the predictions and calculating the community evaluation metrics
myCCV.communityEvaluation.bin <- ecospat.CCV.communityEvaluation.bin(
      ccv.modeling.data = myCCV.Models,
      thresholds = c('MAX.KAPPA', 'MAX.ROC','PS_SDM'),
      community.metrics= c('SR.deviation','Sorensen'),
      parallel = FALSE,
      cpus = 4)
```

`ecospat.CCV.communityEvaluation.prob` *Evaluates community predictions directly on the probabilities (i.e., threshold independent)*

Description

This function generates a number of community evaluation metrics directly based on the probability returned by the individual models. Instead of thresholding the predictions (`ecospat.CCV.communityEvaluation.bin` this function directly uses the probability and compares its outcome to null models or average expectations.)

Usage

```
ecospat.CCV.communityEvaluation.prob(ccv.modeling.data,
    community.metrics=c('SR.deviation','community.AUC','Max.Sorensen',
                        'Max.Jaccard','probabilistic.Sorensen',
                        'probabilistic.Jaccard'),
    parallel = FALSE,
    cpus = 4)
```

Arguments

ccv.modeling.data	a 'ccv.modeling.data' object returned by ecospat.CCV.modeling
community.metrics	a selection of community metrics to calculate ('SR.deviation','community.AUC','Max.Sorensen','probabilistic.Sorensen','probabilistic.Jaccard'))
parallel	should parallel computing be allowed (TRUE/FALSE)
cpus	number of cpus to use in parallel computing

Value

DataSplitTable	a matrix with TRUE/FALSE for each model run (TRUE=Calibration point, FALSE=Evaluation point)
CommunityEvaluationMetrics.CalibrationSites	a 3-dimensional array containing the community evaluation metrics for the calibartion sites of each run (NA means that the site was used for evaluation)
CommunityEvaluationMetrics.EvaluationSites	a 3-dimensional array containing the community evaluation metrics for the evaluation sites of each run (NA means that the site was used for calibaration)

Note

If the community evaluation metric 'SR.deviation' is selected the returned tables will have the following columns:

- SR.obs = observed species richness,
- SR.mean = the predicted species richness (based on the probabilities assuming poission binomial distribution),
- SR.dev = the deviation of observed and predicted species richness,
- SR.sd = the standard deviation of the predicted species richness (based on the probabilities assuming poission binomial distribution),
- SR.prob = the probability that the observed species richness falls within the predicted species richness (based on the probabilities assuming poission binomial distribution),
- SR.imp.05 = improvement of species richness prediction over null-model 0.5,
- SR.imp.average.SR = improvement of species richness prediction over null-model average.SR and
- SR.imp.prevalence = improvement of species richness prediction over null-model prevalence.

If the community evalation metric community.AUC is selected the returned tables will have the following colums:

- Community.AUC = The AUC of ROC of a given site (in this case the ROC plot is community sensitiviy [percentage species predicted corretly present] vs 1 - community specificity [percentage of species predicted correctly absent])

If the community evaluation metrics (`'Max.Sorensen'`, `'Max.Jaccard'`) is selected the returned tables will have the follwing colums:

- `Max.Jaccard` = The maximum possible Jaccard similarity based on an optimal site specific threshold.
- `Max.Sorensen` = The maximum possible Sorensen similarity based on an optimal site specific threshold.

If the community evaluation metrics (`'probabilistic.Sorensen'`, `'probabilistic.Jaccard'`) is selected the returned tables will have the follwing colums:

- `probabilistic.Jaccard` = The probabilistic Jaccard similarity index based on Scherrer et al. 2019, Methods in Ecology and Evolution
- `probabilistic.Sorensen` = The probabilistic Sorensen similarity index based on Scherrer et al. 2019, Methods in Ecology and Evolution
- composition.imp.05 = improvement of species compostion prediction over the null-model 0.5.
- composition.imp.average.SR = improvement of the species composition prediction over the null-model average.SR.
- composition.imp.prevalence = improvement of the species composition prediction over the null-model prevalence.

For detailed descriptions of the null models see Scherrer et al. 2019

Author(s)

Daniel Scherrer <daniel.j.a.scherrer@gmail.com>

See Also

`ecospat.CCV.createDataSplitTable`; `ecospat.CCV.communityEvaluation.bin`;

Examples

```
#Loading species occurence data and remove empty communities
testData <- ecospat.testData[,c(24,34,43,45,48,53,55:58,60:63,65:66,68:71)]
sp.data <- testData[which(rowSums(testData)>0), sort(colnames(testData))]

#Loading environmental data
env.data <- ecospat.testData[which(rowSums(testData)>0),4:8]

#Coordinates for all sites
xy <- ecospat.testData[which(rowSums(testData)>0),2:3]

#Running all the models for all species
myCCV.Models <- ecospat.CCV.modeling(sp.data = sp.data,
                                     env.data = env.data,
                                     xy = xy,
                                     NbRunEval = 2,
                                     minNbPredictors = 10,
```

```
                                    VarImport = 2)

  #Calculating the probabilistic community metrics
  metrics = c('SR.deviation','community.AUC','probabilistic.Sorensen','Max.Sorensen')
  myCCV.Eval.prob <- ecospat.CCV.communityEvaluation.prob(
          ccv.modeling.data = myCCV.Models,
          community.metrics = metrics)
```

`ecospat.CCV.createDataSplitTable` *Creates a DataSplitTable for usage in ecospat.ccv.modeling.*

Description

Creates a DataSplitTable with calibration and evaluation data either for cross-validation or repeated split sampling at the community level (i.e., across all species).

Usage

```
ecospat.CCV.createDataSplitTable(NbRunEval,
                                 DataSplit,
                                 validation.method,
                                 NbSites,
                                 sp.data=NULL,
                                 minNbPresences=NULL,
                                 minNbAbsences=NULL,
                                 maxNbTry=1000)
```

Arguments

`NbRunEval`	number of cross-validation or split sample runs
`DataSplit`	proportion (%) of sites used for model calibration
`validation.method`	the type of `DataSplitTable` that should be created. Must be either `'cross-validation'` or `'split-sample'`
`NbSites`	number of total sites available. Is ignored if sp.data is provided.
`sp.data`	a data.frame where the rows are sites and the columns are species (values 1,0)
`minNbPresences`	the desired minimum number of Presences required in each run
`minNbAbsences`	the desired minimum number of Absences required in each run
`maxNbTry`	number of random tries allowed to create a fitting DataSplitTable

Details

If a `sp.data` data.frame with species presences and absences is provided the function tries to create a `DataSplitTable` which ensures that the maximum possible number of species can be modelled (according to the specified minimum presences and absences.)

Value

`DataSplitTable` a matrix with `TRUE/FALSE` for each model run (`TRUE`=Calibration point, `FALSE`=Evaluation point)

Author(s)

Daniel Scherrer <daniel.j.a.scherrer@gmail.com>

See Also

`ecospat.CCV.modeling`

Examples

```
#Creating a DataSplitTable for 200 sites, 25 runs with an
#80/20 calibration/evaluation cross-validation

DataSplitTable <- ecospat.CCV.createDataSplitTable(NbSites = 200,
                                                   NbRunEval=25,
                                                   DataSplit=80,
                                                   validation.method='cross-validation')

#Loading species occurence data and remove empty communities
testData <- ecospat.testData[,c(24,34,43,45,48,53,55:58,60:63,65:66,68:71)]
sp.data <- testData[which(rowSums(testData)>0), sort(colnames(testData))]

#Creating a DataSplitTable based on species data directly
DataSplitTable <- ecospat.CCV.createDataSplitTable(NbRunEval = 20,
                                                   DataSplit = 70,
                                                 validation.method = "cross-validation",
                                                   NbSites = NULL,
                                                   sp.data = sp.data,
                                                   minNbPresence = 15,
                                                   minNbAbsences = 15,
                                                   maxNbTry = 250)
```

`ecospat.CCV.modeling` *Runs indivudual species distribuion models with SDMs or ESMs*

Description

Creates probabilistic prediction for all species based on SDMs or ESMs and returns their evaluation metrics and variable importances.

Usage

```
ecospat.CCV.modeling(sp.data,
                     env.data,
                     xy,
                     DataSplitTable=NULL,
                     DataSplit = 70,
                     NbRunEval = 25,
                     minNbPredictors =5,
                     validation.method = "cross-validation",
                     models.sdm = c("GLM","RF"),
                     models.esm = "CTA",
                     modeling.options.sdm = NULL,
                     modeling.options.esm = NULL,
                     ensemble.metric = "AUC",
                     ESM = "YES",
                     parallel = FALSE,
                     cpus = 4,
                     VarImport = 10,
                     modeling.id)
```

Arguments

sp.data	a data.frame where the rows are sites and the columns are species (values 1,0)
env.data	either a data.frame where rows are sites and colums are environmental variables or a raster stack of the envrionmental variables
xy	two column data.frame with X and Y coordinates of the sites (most be same coordinate system as env.data)
DataSplitTable	a table providing TRUE/FALSE to indicate what points are used for calibration and evaluation. As returned by ecospat.CCV.createDataSplitTable
DataSplit	percentage of dataset observations retained for the model training (only needed if no DataSplitTable provided)
NbRunEval	number of cross-validatio/split sample runs (only needed if no DataSplitTable provided)
minNbPredictors	minimum number of occurences [min(presences/Absences] per predicotors needed to calibrate the models
validation.method	either "cross-validation" or "split-sample" used to validate the communtiy predictions (only needed if no DataSplitTable provided)
models.sdm	modeling techniques used for the normal SDMs. Vector of models names choosen among 'GLM', 'GBM', 'GAM', 'CTA', 'ANN', 'SRE', 'FDA', 'MARS', 'RF', 'MAXENT.Phillips' and 'MAXENT.Tsuruoka'
models.esm	modeling techniques used for the ESMs. Vector of models names choosen among 'GLM', 'GBM', 'GAM', 'CTA', 'ANN', 'SRE', 'FDA', 'MARS', 'RF', 'MAXENT.Phillips' and 'MAXENT.Tsuruoka'

`modeling.options.sdm`
: modeling options for the normal SDMs. "BIOMOD.models.options"" object returned by BIOMOD_ModelingOptions

`modeling.options.esm`
: modeling options for the ESMs. "BIOMOD.models.options" object returned by BIOMOD_ModelingOptions

`ensemble.metric`
: evaluation score used to weight single models to build ensembles: 'AUC', 'Kappa' or 'TSS'

`ESM`
: either 'YES' (ESMs allowed), 'NO' (ESMs not allowed) or 'ALL' (ESMs used in any case)

`parallel`
: should parallel computing be allowed (TRUE/FALSE)

`cpus`
: number of cpus to use in parallel computing

`VarImport`
: number of permutation runs to evaluate variable importance

`modeling.id`
: character, the ID (=name) of modeling procedure. A random number by default

Details

The basic idea of the community cross-validation (CCV) is to use the same data (sites) for the model calibration/evaluation of all species. This ensures that there is "independent" cross-validation/split-sample data available not only at the individual species level but also at the community level. This is key to allow an unbiased estimation of the ability to predict species assemblages (Scherrer et al. 2018). The output of the ecospat.CCV.modeling function can then be used to evaluate the species assemblage predictions with the ecospat.CCV.communityEvaluation.bin or ecospat.CCV.communityEvaluation.prob functions.

Value

`modelling.id`
: character, the ID (=name) of modeling procedure

`output.files`
: vector with the names of the files written to the hard drive

`speciesData.calibration`
: a 3-dimensional array of presence/absence data of all species for the calibration plots used for each run

`speciesData.evaluation`
: a 3-dimensional array of presence/absence data of all species for the evaluation plots used for each run

`speciesData.full`
: a data.frame of presence/absence data of all species (same as sp.data input)

`DataSplitTable`
: a matrix with TRUE/FALSE for each model run (TRUE=Calibration point, FALSE=Evaluation point)

`singleSpecies.ensembleEvaluationScore`
: a 3-dimensional array of single species evaluation metrics ('Max.KAPPA', 'Max.TSS', 'AUC of ROC')

`singleSpecies.ensembleVariableImportance`
: a 3-dimensional array of single species variable importance for all predictors

`singleSpecies.calibrationSites.ensemblePredictions`
: a 3-dimensional array of the predictions for each species and run at the calibration sites

`singleSpecies.evaluationSites.ensemblePredictions`
: a 3-dimensional array of the predictions for each species and run at the evaluation sites

`allSites.averagePredictions.cali`
: a matrix with the average predicted probabilities for each site across all the runs the sites were used for model calibration

`allSites.averagePredictions.eval`
: a matrix with the average predicted probabilities for each site across all the runs the sites were used as independent evaluation sites

Author(s)

Daniel Scherrer <daniel.j.a.scherrer@gmail.com>

References

Scherrer, D., D'Amen, M., Mateo, M.R.G., Fernandes, R.F. & Guisan , A. (2018) How to best threshold and validate stacked species assemblages? Community optimisation might hold the answer. Methods in Ecology and Evolution, in review

See Also

ecospat.CCV.createDataSplitTable; ecospat.CCV.communityEvaluation.bin; ecospat.CCV.communityEvaluatio

Examples

```
#Loading species occurence data and remove empty communities
testData <- ecospat.testData[,c(24,34,43,45,48,53,55:58,60:63,65:66,68:71)]
sp.data <- testData[which(rowSums(testData)>0), sort(colnames(testData))]

#Loading environmental data
env.data <- ecospat.testData[which(rowSums(testData)>0),4:8]

#Coordinates for all sites
xy <- ecospat.testData[which(rowSums(testData)>0),2:3]

#Running all the models for all species
myCCV.Models <- ecospat.CCV.modeling(sp.data = sp.data,
                                     env.data = env.data,
                                     xy = xy,
                                     NbRunEval = 5,
                                     minNbPredictors = 10,
                                     VarImport = 3)
```

`ecospat.climan` *A climate analogy setection tool for the modeling of species distributions*

Description

Assess climate analogy between a projection extent (p) and a reference extent (ref, used in general as the background to calibrate SDMs)

Usage

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

Arguments

`ref`	A dataframe with the value of the variables (i.e columns) for each point of the reference exent.
`p`	A dataframe with the value of the variables (i.e columns) for each point of the projection exent.

Value

Returns a vector. Values below 0 are novel conditions at the univariate level (similar to the MESS), values between 0 and 1 are analog and values above 1 are novel covariate condtions. For more information see Mesgeran et al. (2014)

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com>

References

Mesgaran, M.B., R.D. Cousens and B.L. Webber. 2014. Here be dragons: a tool for quantifying novelty due to covariate range and correlation change when projecting species distribution models. *Diversity & Distributions*, **20**, 1147-1159.

Examples

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

`ecospat.cohen.kappa` *Cohen's Kappa*

Description

Calculates Cohen's kappa and variance estimates, within a 95 percent confidence interval.

Usage

```
ecospat.cohen.kappa(xtab)
```

Arguments

`xtab`	A symmetric agreement table.

Details

The argument xtab is a contingency table. xtab <- table(Pred >= th, Sp.occ)

Value

A list with elements 'kap', 'vark', 'totn' and 'ci' is returned. 'kap' is the cohen's kappa, 'vark' is the variance estimate within a 95 percent confidence interval, 'totn' is the number of plots and 'ci' is the confidence interval.

Author(s)

Christophe Randin <christophe.randin@wsl.ch> with contributions of Niklaus. E. Zimmermann <niklaus.zimmermann@wsl.ch> and Valeria Di Cola <valeria.dicola@unil.ch>

References

Bishop, Y.M.M., S.E. Fienberg and P.W. Holland. 1975. Discrete multivariate analysis: Theory and Practice. Cambridge, MA: MIT Press. pp. 395-397.

Pearce, J. and S. Ferrier. 2000. Evaluating the predictive performance of habitat models developed using logistic regression. *Ecol. Model.*, **133**, 225-245.

See Also

`ecospat.meva.table`, `ecospat.max.tss`, `ecospat.plot.tss`, `ecospat.plot.kappa`, `ecospat.max.kappa`

Examples

```
Pred <- ecospat.testData$glm_Agrostis_capillaris
Sp.occ <- ecospat.testData$Agrostis_capillaris
th <- 0.39 # threshold
xtab <- table(Pred >= th, Sp.occ)

ecospat.cohen.kappa(xtab)
```

ecospat.CommunityEval *Community Evaluation*

Description

Calculate several indices of accuracy of community predictions.

Usage

```
ecospat.CommunityEval (eval, pred, proba, ntir, verbose = FALSE)
```

Arguments

eval	A matrix of observed presence-absence (ideally independent from the dataset used to fit species distribution models) of the species with n rows for the sites and s columns for the species.
pred	A matrix of predictions for the s species in the n sites. Should have the same dimension as eval.
proba	Logical variable indicating whether the prediction matrix contains presences-absences (FALSE) or probabilities (TRUE).
ntir	Number of trials of presence-absence predictions if pred is a probability matrix.
verbose	Boolean indicating whether to print progress output during calculation. Default is FALSE.

Details

This function calculates several indices of accuracy of community predictions based on stacked predictions of species ditribution models. In case proba is set to FALSE the function returns one value per index and per site. In case proba is set to TRUE the function generates presences-absences based on the predicted probabilities and returns one value per index, per site and per trial.

Value

A list of evaluation metrics calculated for each site (+ each trial if proba is set to TRUE):

deviance.rich.pred: the deviation of the predicted species richness to the observed

overprediction: the proportion of species predicted as present but not observed among the species predicted as present

underprediction: the proportion of species predicted as absent but observed among the species observed as present

prediction.success: the proportion of species correctly predicted as present or absent

sensitivity: the proportion of species correctly predicted as present among the species observed as present

specificity : the proportion of species correctly predicted as absent among the species observed as absent

kappa: the proportion of specific agreement

TSS: sensitivity+specificity-1

similarity: the similarity of community composition between the observation and the prediction. The calculation is based on the Sorenses index.

Jaccard: this index is a widely used metric of community similarity.

Author(s)

Julien Pottier <julien.pottier@clermont.inra.fr>

with contribution of Daniel Scherrer <daniel.scherrer@unil.ch>, Anne Dubuis <anne.dubuis@gmail.com> and Manuela D'Amen <manuela.damen@unil.ch>

References

Pottier, J., A. Dubuis, L. Pellissier, L. Maiorano, L. Rossier, C.F. Randin, P. Vittoz and A. Guisan. 2013. The accuracy of plant assemblage prediction from species distribution models varies along environmental gradients. *Global Ecology and Biogeography*, **22**, 52-63.

Examples

```
data(ecospat.testData)
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)]

ecospat.CommunityEval (eval, pred, proba=TRUE, ntir=10)
```

---

ecospat.cons_Cscore *Constrained Co-Occurrence Analysis.*

Description

Co-occurrence Analysis & Environmentally Constrained Null Models. 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.

Usage

```
ecospat.cons_Cscore(presence,pred,nperm,outpath,verbose = FALSE)
```

Arguments

presence	A presence-absence dataframe for each species (columns) in each location or grid cell (rows) Column names (species names) and row names (sampling plots).
pred	A dataframe object with SDM predictions. Column names (species names SDM) and row names (sampling plots).
nperm	The number of permutation in the null model.
outpath	Path to specify where to save the results.
verbose	Boolean indicating whether to print progress output during calculation. Default is FALSE.

Details

An environmentally constrained approach to null models will provide a more robust evaluation of species associations by facilitating the distinction between mutually exclusive processes that may shape species distributions and community assembly. The format required for input databases: a plots (rows) x species (columns) matrix. Input matrices should have column names (species names) and row names (sampling plots). NOTE: 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)

Value

Returns the C-score index for the observed community (ObsCscoreTot), the mean of C-score for the simulated communities (SimCscoreTot), p.value (PValTot) and standardized effect size (SES.Tot). It also saves a table in the specified path 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).

Author(s)

Anne Dubuis <anne.dubuis@gmail.com> and Manuela D'Amen <manuela.damen@unil.ch>

References

Gotelli, N.J. and D.J. McCabe. 2002. Species co-occurrence: a meta-analysis of JM Diamond's assembly rules model. *Ecology*, **83**, 2091-2096.

Peres-Neto, P.R., J.D. Olden and D.A. Jackson. 2001. Environmentally constrained null models: site suitability as occupancy criterion. *Oikos*, **93**, 110-120.

Examples

```
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)]
nperm <- 10000
outpath <- getwd()
cons_Cscore<-ecospat.cons_Cscore(presence, pred, nperm, outpath)
```

ecospat.cor.plot *Correlation Plot*

Description

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).

Usage

```
ecospat.cor.plot(data)
```

Arguments

data	A dataframe object with environmental variables.

Details

Adapted from the pairs help page. Uses panel.cor, and panel.hist, all taken from the help pages for pairs. It is a simplifies version of `pairs.panels()` function of the package `psych`.

Value

A scatter plot matrix is drawn in the graphic window. The lower off diagonal draws scatter plots, the diagonal histograms, the upper off diagonal reports the Pearson correlation.

Author(s)

Adjusted by L. Mathys, 2006, modified by N.E. Zimmermann

Examples

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

ecospat.co_occurrences *Species Co-Occurrences*

Description

Calculate an index of species co-occurrences.

Usage

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

Arguments

data	A presence-absence matrix for each species (columns) in each location or grid cell (rows) or a matrix with predicted suitability values.

Details

Computes an index of co-occurrences ranging from 0 (never co-occurring) to 1 (always co-occuring).

Value

The species co-occurrence matrix and box-plot of the co-occurrence indices

Author(s)

Loic Pellissier <loic.pellissier@unifr.ch>

References

Pellissier, L., K.A. Brathen, J. Pottier, C.F. Randin, P. Vittoz, A. Dubuis, N.G. Yoccoz, T. Alm, N.E. Zimmermann and A. Guisan. 2010. Species distribution models reveal apparent competitive and facilitative effects of a dominant species on the distribution of tundra plants. *Ecography*, **33**, 1004-1014.

Guisan, A. and N. Zimmermann. 2000. Predictive habitat distribution models in ecology. *Ecological Modelling*, **135**:147-186

Examples

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

---

ecospat.Cscore *Pairwise co-occurrence Analysis with calculation of the C-score index.*

Description

The function tests for nonrandom 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. Null communities have column sum fixed.

Usage

```
ecospat.Cscore (data, nperm, outpath, verbose = FALSE)
```

Arguments

data	A presence-absence dataframe for each species (columns) in each location or grid cell (rows). Column names (species names) and row names (sampling plots).
nperm	The number of permutation in the null model.
outpath	Path to specify where to save the results.
verbose	Boolean indicating whether to print progress output during calculation. Default is FALSE.

Details

This function allows to apply a pairwise null model analysis (Gotelli and Ulrich 2010) 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 (Stone and Roberts 1990) and a 'fixed-equiprobable' null model algorithm is applied. The format required for input databases: a plots (rows) x species (columns) matrix. Input matrices should have column names (species names) and row names (sampling plots). NOTE: 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).

Value

The function returns the C-score index for the observed community (ObsCscoreTot), p.value (PVal-Tot) 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)

Author(s)

Christophe Randin < christophe.randin@wsl.ch> and Manuela D'Amen <manuela.damen@msn.com>

References

Gotelli, N.J. and D.J. McCabe. 2002. Species co-occurrence: a meta-analysis of JM Diamond's assembly rules model. *Ecology*, **83**, 2091-2096.

Gotelli, N.J. and W. Ulrich. 2010. The empirical Bayes approach as a tool to identify non-random species associations. *Oecologia*, **162**, 463-477

Stone, L. and A. Roberts, A. 1990. The checkerboard score and species distributions. *Oecologia*, **85**, 74-79

See Also

ecospat.co_occurrences and ecospat.cons_Cscore

Examples

```
## Not run:
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 <- 10000
outpath <- getwd()
Cscore<-ecospat.Cscore(data, nperm, outpath)


## End(Not run)
```

`ecospat.cv.gbm` *GBM Cross Validation*

Description

K-fold and leave-one-out cross validation for GBM.

Usage

```
ecospat.cv.gbm (gbm.obj, data.cv, K=10, cv.lim=10, jack.knife=FALSE, verbose = FALSE)
```

Arguments

`gbm.obj`	A calibrated GBM object with a binomial error distribution. Attention: users have to tune model input parameters according to their study!
`data.cv`	A dataframe object containing the calibration data set with the same names for response and predictor variables.
`K`	Number of folds. 10 is recommended; 5 for small data sets.
`cv.lim`	Minimum number of presences required to perform the K-fold cross-validation.
`jack.knife`	If TRUE, then the leave-one-out / jacknife cross-validation is performed instead of the 10-fold cross-validation.
`verbose`	Boolean indicating whether to print progress output during calculation. Default is FALSE.

Details

This function takes a calibrated GBM object with a binomial error distribution and returns predictions from a stratified 10-fold cross-validation or a leave-one-out / jack-knived cross-validation. Stratified means that the original prevalence of the presences and absences in the full dataset is conserved in each fold.

Value

Returns a dataframe with the observations (obs) and the corresponding predictions by cross-validation or jacknife.

Author(s)

Christophe Randin <christophe.randin@unibas.ch> and Antoine Guisan <antoine.guisan@unil.ch>

References

Randin, C.F., T. Dirnbock, S. Dullinger, N.E. Zimmermann, M. Zappa and A. Guisan. 2006. Are niche-based species distribution models transferable in space? *Journal of Biogeography*, **33**, 1689-1703.

Pearman, P.B., C.F. Randin, O. Broennimann, P. Vittoz, W.O. van der Knaap, R. Engler, G. Le Lay, N.E. Zimmermann and A. Guisan. 2008. Prediction of plant species distributions across six millennia. *Ecology Letters*, **11**, 357-369.

Examples

```
library(gbm)
data('ecospat.testData')

# data for Soldanella alpina
data.Solalp<- ecospat.testData[c("Soldanella_alpina","ddeg","mind","srad","slp","topo")]

# gbm model for Soldanella alpina
gbm.Solalp <- gbm(Soldanella_alpina ~ ., data = data.Solalp,
                  distribution = "bernoulli", cv.folds = 10, n.cores=2)

# cross-validated predictions
gbm.pred <- ecospat.cv.gbm (gbm.obj= gbm.Solalp,data.Solalp,
                            K=10, cv.lim=10, jack.knife=FALSE)
```

---

ecospat.cv.glm *GLM Cross Validation*

Description

K-fold and leave-one-out cross validation for GLM.

Usage

```
ecospat.cv.glm (glm.obj, K=10, cv.lim=10, jack.knife = FALSE, verbose = FALSE)
```

Arguments

glm.obj	Any calibrated GLM object with a binomial error distribution.
K	Number of folds. 10 is recommended; 5 for small data sets.
cv.lim	Minimum number of presences required to perform the K-fold cross-validation.
jack.knife	If TRUE, then the leave-one-out / jacknife cross-validation is performed instead of the 10-fold cross-validation.
verbose	Boolean indicating whether to print progress output during calculation. Default is FALSE.

Details

This function takes a calibrated GLM object with a binomial error distribution and returns predictions from a stratified 10-fold cross-validation or a leave-one-out / jack-knived cross-validation. Stratified means that the original prevalence of the presences and absences in the full dataset is conserved in each fold.

Value

Returns a dataframe with the observations (obs) and the corresponding predictions by cross-validation or jacknife.

Author(s)

Christophe Randin <christophe.randin@unibas.ch> and Antoine Guisan <antoine.guisan@unil.ch>

References

Randin, C.F., T. Dirnbock, S. Dullinger, N.E. Zimmermann, M. Zappa and A. Guisan. 2006. Are niche-based species distribution models transferable in space? *Journal of Biogeography*, **33**, 1689-1703.

Pearman, P.B., C.F. Randin, O. Broennimann, P. Vittoz, W.O. van der Knaap, R. Engler, G. Le Lay, N.E. Zimmermann and A. Guisan. 2008. Prediction of plant species distributions across six millennia. *Ecology Letters*, **11**, 357-369.

Examples

```
if(require("rms",quietly=TRUE)){
  data('ecospat.testData')

  # data for Soldanella alpina
 data.Solalp<- ecospat.testData[c("Soldanella_alpina","ddeg","mind","srad","slp","topo")]

  # gbm model for Soldanella alpina

 glm.Solalp <- glm(Soldanella_alpina ~ pol(ddeg,2) + pol(mind,2) + pol(srad,2) + pol(slp,2)
      + pol(topo,2), data = data.Solalp, family = binomial)

  # cross-validated predictions
  glm.pred <- ecospat.cv.glm (glm.obj=glm.Solalp , K=10, cv.lim=10, jack.knife=FALSE)
}
```

`ecospat.cv.me` *Maxent Cross Validation*

Description

K-fold and leave-one-out cross validation for Maxent.

Usage

```
ecospat.cv.me(data.cv.me, name.sp, names.pred, K=10, cv.lim=10,
              jack.knife=FALSE, verbose=FALSE)
```

Arguments

`data.cv.me`	A dataframe object containing the calibration data set of a Maxent object to validate with the same names for response and predictor variables.
`name.sp`	Name of the species / response variable.
`names.pred`	Names of the predicting variables.
`K`	Number of folds. 10 is recommended; 5 for small data sets.
`cv.lim`	Minimum number of presences required to perform the K-fold cross-validation.
`jack.knife`	If TRUE, then the leave-one-out / jacknife cross-validation is performed instead of the 10-fold cross-validation.
`verbose`	Boolean indicating whether to print progress output during calculation. Default is FALSE.

Details

This function takes a calibrated Maxent object with a binomial error distribution and returns predictions from a stratified 10-fold cross-validation or a leave-one-out / jack-knived cross-validation. Stratified means that the original prevalence of the presences and absences in the full dataset is conserved in each fold.

Value

Returns a dataframe with the observations (obs) and the corresponding predictions by cross-validation or jacknife.

Author(s)

Christophe Randin <christophe.randin@unibas.ch> and Antoine Guisan <antoine.guisan@unil.ch>

References

Randin, C.F., T. Dirnbock, S. Dullinger, N.E. Zimmermann, M. Zappa and A. Guisan. 2006. Are niche-based species distribution models transferable in space? *Journal of Biogeography*, **33**, 1689-1703.

Pearman, P.B., C.F. Randin, O. Broennimann, P. Vittoz, W.O. van der Knaap, R. Engler, G. Le Lay, N.E. Zimmermann and A. Guisan. 2008. Prediction of plant species distributions across six millennia. *Ecology Letters*, **11**, 357-369.

Examples

```
data('ecospat.testData')

# data for Soldanella alpina
data.Solalp<- ecospat.testData[c("Soldanella_alpina","ddeg","mind","srad","slp","topo")]

# maxent modelling and cross-validated predictions

# path to maxent.jar file
path<- paste0(system.file(package="dismo"), "/java/maxent.jar")

if (file.exists(path) & require(rJava)) {
  me.pred <- ecospat.cv.me(data.Solalp, names(data.Solalp)[1],
             names(data.Solalp)[-1], K = 10, cv.lim = 10, jack.knife = FALSE)
  }
```

---

ecospat.cv.rf *RandomForest Cross Validation*

Description

K-fold and leave-one-out cross validation for randomForest.

Usage

```
ecospat.cv.rf (rf.obj, data.cv, K=10, cv.lim=10, jack.knife=FALSE, verbose = FALSE)
```

Arguments

rf.obj	Any calibrated randomForest object with a binomial error distribution.
data.cv	A dataframe object containing the calibration data set with the same names for response and predictor variables.
K	Number of folds. 10 is recommended; 5 for small data sets.
cv.lim	Minimum number of presences required to perform the K-fold cross-validation.
jack.knife	If TRUE, then the leave-one-out / jacknife cross-validation is performed instead of the 10-fold cross-validation.

verbose — Boolean indicating whether to print progress output during calculation. Default is FALSE.

Details

This function takes a calibrated randomForest object with a binomial error distribution and returns predictions from a stratified 10-fold cross-validation or a leave-one-out / jack-knived cross-validation. Stratified means that the original prevalence of the presences and absences in the full dataset is conserved in each fold.

Value

Returns a dataframe with the observations (obs) and the corresponding predictions by cross-validation or jacknife.

Author(s)

Christophe Randin <christophe.randin@wsl.ch> and Antoine Guisan <antoine.guisan@unil.ch>

References

Randin, C.F., T. Dirnbock, S. Dullinger, N.E. Zimmermann, M. Zappa and A. Guisan. 2006. Are niche-based species distribution models transferable in space? *Journal of Biogeography*, **33**, 1689-1703.

Pearman, P.B., C.F. Randin, O. Broennimann, P. Vittoz, W.O. van der Knaap, R. Engler, G. Le Lay, N.E. Zimmermann and A. Guisan. 2008. Prediction of plant species distributions across six millennia. *Ecology Letters*, **11**, 357-369.

Examples

```
data('ecospat.testData')

# data for Soldanella alpina
data.Solalp<- ecospat.testData[c("Soldanella_alpina","ddeg","mind","srad","slp","topo")]

library(randomForest)
rf.Solalp <- randomForest(x = data.Solalp[,-1], y = as.factor(data.Solalp[,1]))
rf.pred <- ecospat.cv.rf(rf.Solalp, data.Solalp, K = 10, cv.lim = 10,
                         jack.knife = FALSE, verbose = FALSE)
```

---

ecospat.Epred *Prediction Mean*

Description

Calculate the mean (or weighted mean) of several predictions.

Usage

```
ecospat.Epred (x, w=rep(1,ncol(x)), th=0)
```

Arguments

x	A dataframe object with SDM predictions.
w	Weight of the model, e.g. AUC. The default is 1.
th	Threshold used to binarize.

Details

The Weighted Average consensus method utilizes pre-evaluation of the predictive performance of the single-models. In this approach, half (i.e. four) of the eight single-models with highest accuracy are selected first, and then a WA is calculated based on the pre-evaluated AUC of the single-models

Value

A weighted mean binary transformation of the models.

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com>

References

Boyce, M.S., P.R. Vernier, S.E. Nielsen and F.K.A. Schmiegelow. 2002. Evaluating resource selection functions. *Ecol. Model.*, **157**, 281-300.

Marmion, M., M. Parviainen, M. Luoto, R.K. Heikkinen andW. Thuiller. 2009. Evaluation of consensus methods in predictive species distribution modelling. *Diversity and Distributions*, **15**, 59-69.

Examples

```
x <- ecospat.testData[c(92,96)]
mean <- ecospat.Epred (x, w=rep(1,ncol(x)), th=0.5)
```

---

ecospat.ESM.EnsembleEvaluation

Ensemble of Small Models: Evaluation via the Pooling procedure

---

Description

This function evaluates the Ensemble of Small Models by pooling the different runs of the cross validation as in Collart et al. 2021

Usage

```
ecospat.ESM.EnsembleEvaluation(ESM.modeling.output,
                               ESM.EnsembleModeling.output,
                     metrics = c("SomersD","AUC","MaxTSS","MaxKappa","Boyce"),
                               EachSmallModels = FALSE)
```

Arguments

`ESM.modeling.output`
: a `list` returned by `ecospat.ESM.Modeling`

`ESM.EnsembleModeling.output`
: a `list` returned by `ecospat.ESM.EnsembleModeling`

`metrics`
: a vector of evaluation metrics chosen among "SomersD", "AUC", "MaxTSS", "MaxKappa", "Boyce"

`EachSmallModels`
: should the individual bivariate models be evaluated by the pooling procedure?

Details

Because a minimum sample size is needed to evaluate models (see Jimenez-Valverde, 2020), this function uses the approach from Collart et al.(2021), which consists to pool the suitability values of the hold-out data (evaluation dataset) across replicates. As the same data point (presence or absence or background point) is presumably sampled in several replicates, the suitability values for each data point is consequently averaged across replicates where they were sampled. This procedure generates a series of independent suitability values with a size approximately equal (as some data points may not have been sampled by chance in any of the *n* replicates) to that of the number of data point.

Value

a list containing:

`ESM.evaluations`
: a matrix with the evaluation scores for the ESMs based on the different modelling algorithms and based on the consensus across the modelling algorithms (called here "ensemble")

`ESM.fit`
: a matrix of predicted values resulting from the pooling procedure and used to compute the evaluation scores. The column *resp* is where the species occurs or not

`ESM.evaluations.bivariate.models`
: a list containing a matrix of evaluation scores for each bivariate models (generated only if EachSmallModels = T)

`ESM.fit.bivariate.models`
: a list containing a matrix of of predicted values resulting from the pooling procedure for each bivariate models (generated only if EachSmallModels = T)

Author(s)

Flavien Collart <flavien.collart@unil.ch>

with contributions of Olivier Broennimann <olivier.broennimann@unil.ch>

References

Collart, F., Hedenas, L., Broennimann, O., Guisan, A. and Vanderpoorten, A. 2021. Intraspecific differentiation: Implications for niche and distribution modelling. *Journal of Biogeography*. **48**, 415-426. doi:10.1111/jbi.14009

Jimenez-Valverde, A. 2020. Sample size for the evaluation of presence-absence models. *Ecological Indicators*. **114**, 106289. doi:10.1016/j.ecolind.2020.106289

See Also

ecospat.ESM.EnsembleModeling

---

ecospat.ESM.EnsembleModeling

Ensemble of Small Models: Evaluates and Averages Simple Bivariate Models To ESMs

---

Description

This function evaluates and averages simple bivariate models by weighted means to Ensemble Small Models as in Lomba et al. 2010 and Breiner et al. 2015.

Usage

```
ecospat.ESM.EnsembleModeling( ESM.modeling.output,
                              weighting.score,
                              threshold=NULL,
                              models)
```

Arguments

ESM.modeling.output	a list returned by ecospat.ESM.Modeling
weighting.score	an evaluation score used to weight single models to build ensembles:"AUC","TSS", "Boyce","Kappa","SomersD" #the evaluation methods used to evaluate ensemble models (see "BIOMOD_Modeling" **models.eval.meth** section for more detailed informations)
threshold	threshold value of an evaluation score to select the bivariate model(s) included for building the ESMs

`models` vector of models names choosen among 'GLM', 'GBM', 'GAM', 'CTA', 'ANN', 'SRE', 'FDA', 'MARS', 'RF','MAXENT.Phillips', "MAXENT.Tsuruoka" (same as in `biomod2`)

#a character vector (either 'all' or a sub-selection of model names) that defines the models kept for building the ensemble models (might be useful for removing some non-preferred models)

Details

The basic idea of ensemble of small models (ESMs) is to model a species distribution based on small, simple models, for example all possible bivariate models (i.e. models that contain only two predictors at a time out of a larger set of predictors), and then combine all possible bivariate models into an ensemble (Lomba et al. 2010; Breiner et al. 2015).

The ESM set of functions could be used to build ESMs using simple bivariate models which are averaged using weights based on model performances (e.g. AUC) according to Breiner et al. (2015). They provide full functionality of the approach described in Breiner et al. (2015).

Value

species: species name ESM.fit: data.frame of the predicted values for the data used to build the models. ESM.evaluations: data.frame with evaluations scores for the ESMs weights: weighting scores used to weight the bivariate models to build the single ESM weights.EF: weighting scores used to weight the single ESM to build the ensemble of ESMs from different modelling techniques (only available if >1 modelling techniques were selected). failed: bivariate models which failed because they could not be calibrated.

A "`BIOMOD.ensemble.models.out`" object. This object will be later given to `ecospat.ESM.EnsembleProjection` if you want to make some projections of this ensemble-models.

Author(s)

Frank Breiner <frank.breiner@wsl.ch>

with contributions of Olivier Broennimann <olivier.broennimann@unil.ch>

References

Lomba, A., L. Pellissier, C.F. Randin, J. Vicente, F. Moreira, J. Honrado and A. Guisan. 2010. Overcoming the rare species modelling paradox: A novel hierarchical framework applied to an Iberian endemic plant. *Biological Conservation*, **143**,2647-2657.

Breiner F.T., A. Guisan, A. Bergamini and M.P. Nobis. 2015. Overcoming limitations of modelling rare species by using ensembles of small models. *Methods in Ecology and Evolution*, **6**,1210-1218.

Breiner F.T., Nobis M.P., Bergamini A., Guisan A. 2018. Optimizing ensembles of small models for predicting the distribution of species with few occurrences. *Methods in Ecology and Evolution*. doi:10.1111/2041210X.12957

See Also

`ecospat.ESM.Modeling`

`ecospat.ESM.EnsembleProjection` *Ensemble of Small Models: Projects Calibrated ESMs Into New Space Or Time.*

Description

This function projects calibrated ESMs into new space or time.

Usage

```
ecospat.ESM.EnsembleProjection( ESM.prediction.output,
                                ESM.EnsembleModeling.output,
                                chosen.models = 'all')
```

Arguments

`ESM.prediction.output`	a list object returned by `ecospat.ESM.Projection`
`ESM.EnsembleModeling.output`	a list object returned by `ecospat.ESM.EnsembleModeling`
`chosen.models`	a character vector (either 'all' or a sub-selection of model names, e.g. c(GLM, GBM)) to remove models from the ensemble. Default is 'all'.

Details

The basic idea of ensemble of small models (ESMs) is to model a species distribution based on small, simple models, for example all possible bivariate models (i.e. models that contain only two predictors at a time out of a larger set of predictors), and then combine all possible bivariate models into an ensemble (Lomba et al. 2010; Breiner et al. 2015).

The ESM set of functions could be used to build ESMs using simple bivariate models which are averaged using weights based on model performances (e.g. AUC) according to Breiner et al. (2015). They provide full functionality of the approach described in Breiner et al. (2015). For projections only the full models (100

For further details please refer to `BIOMOD_EnsembleForecasting`.

Value

Returns the projections of ESMs for the selected single models and their ensemble (data frame or raster stack). ESM.projections 'projection files' are saved on the hard drive projection folder. This files are either an `array` or a `RasterStack` depending the original projections data type. Load these created files to plot and work with them.

Author(s)

Frank Breiner <frank.breiner@wsl.ch>

References

Lomba, A., L. Pellissier, C.F. Randin, J. Vicente, F. Moreira, J. Honrado and A. Guisan. 2010. Overcoming the rare species modelling paradox: A novel hierarchical framework applied to an Iberian endemic plant. *Biological Conservation*, **143**,2647-2657.

Breiner F.T., A. Guisan, A. Bergamini and M.P. Nobis. 2015. Overcoming limitations of modelling rare species by using ensembles of small models. *Methods in Ecology and Evolution*, **6**,1210-1218.

Breiner F.T., Nobis M.P., Bergamini A., Guisan A. 2018. Optimizing ensembles of small models for predicting the distribution of species with few occurrences. *Methods in Ecology and Evolution*. doi:10.1111/2041210X.12957

See Also

ecospat.ESM.EnsembleModeling

`ecospat.ESM.Modeling` *Ensemble of Small Models: Calibration of Simple Bivariate Models*

Description

This function calibrates simple bivariate models as in Lomba et al. 2010 and Breiner et al. 2015.

Usage

```
ecospat.ESM.Modeling( data,
                      NbRunEval,
                      DataSplit,
                      DataSplitTable,
                      Prevalence,
                      weighting.score,
                      models,
                      tune,
                      modeling.id,
                      models.options,
                      which.biva,
                      parallel,
                      cleanup,
                      Yweights)
```

Arguments

`data`	`BIOMOD.formated.data` object returned by `BIOMOD_FormatingData`
`NbRunEval`	number of dataset splitting replicates for the model evaluation (same as in `biomod2`)
`DataSplit`	percentage of dataset observations retained for the model training (same as in `biomod2`)

DataSplitTable	a matrix, data.frame or a 3D array filled with TRUE/FALSE to specify which part of data must be used for models calibration (TRUE) and for models validation (FALSE). Each column corresponds to a 'RUN'. If filled, arguments NbRunEval and DataSplit will be ignored.
Prevalence	either NULL or a 0-1 numeric used to build 'weighted response weights'. In contrast to Biomod the default is 0.5 (weighting presences equally to the absences). If NULL each observation (presence or absence) has the same weight (independent of the number of presences and absences).
weighting.score	evaluation score used to weight single models to build ensembles: 'AUC', 'SomersD' (2xAUC-1), 'Kappa', 'TSS' or 'Boyce'
models	vector of models names choosen among 'GLM', 'GBM', 'GAM', 'CTA', 'ANN', 'SRE', 'FDA', 'MARS', 'RF','MAXENT.Phillips', 'MAXENT.Phillips.2' (same as in biomod2)
tune	logical. if true model tuning will be used to estimate optimal parameters for the models (Default: False).
modeling.id	character, the ID (=name) of modeling procedure. A random number by default.
models.options	BIOMOD.models.options object returned by BIOMOD_ModelingOptions (same as in biomod2)
Yweights	response points weights. This argument will only affect models that allow case weights.
which.biva	integer. which bivariate combinations should be used for modeling? Default: all
parallel	logical. If TRUE, the parallel computing is enabled (highly recommended)
cleanup	numeric. Calls removeTmpFiles() to delete all files from rasterOptions()$tmpdir which are older than the given time (in hours). This might be necessary to prevent running over quota. No cleanup is used by default.

Details

The basic idea of ensemble of small models (ESMs) is to model a species distribution based on small, simple models, for example all possible bivariate models (i.e. models that contain only two predictors at a time out of a larger set of predictors), and then combine all possible bivariate models into an ensemble (Lomba et al. 2010; Breiner et al. 2015).

The ESM set of functions could be used to build ESMs using simple bivariate models which are averaged using weights based on model performances (e.g. AUC) according to Breiner et al. (2015). They provide full functionality of the approach described in Breiner et al. (2015).

The argument which.biva allows to split model runs, e.g. if which.biva is 1:3, only the three first bivariate variable combinations will be modeled. This allows to run different biva splits on different computers. However, it is better not to use this option if all models are run on a single computer. Default: running all biva models. NOTE: Make sure to give each of your biva runs a unique modeling.id. Please avoid space characters in your working directory path if you are using MAXENT.Phillips because this can cause an error.

Value

A BIOMOD.models.out object (same as in biomod2) See "BIOMOD.models.out" for details.

Author(s)

Frank Breiner <frank.breiner@wsl.ch> and Mirko Di Febbraro <mirkodifebbraro@gmail.com> with contributions of Olivier Broennimann <olivier.broennimann@unil.ch>

References

Lomba, A., L. Pellissier, C.F. Randin, J. Vicente, F. Moreira, J. Honrado and A. Guisan. 2010. Overcoming the rare species modelling paradox: A novel hierarchical framework applied to an Iberian endemic plant. *Biological Conservation*, **143**,2647-2657.

Breiner F.T., A. Guisan, A. Bergamini and M.P. Nobis. 2015. Overcoming limitations of modelling rare species by using ensembles of small models. *Methods in Ecology and Evolution*, **6**,1210-1218.

Breiner F.T., Nobis M.P., Bergamini A., Guisan A. 2018. Optimizing ensembles of small models for predicting the distribution of species with few occurrences. *Methods in Ecology and Evolution*. doi:10.1111/2041210X.12957

See Also

ecospat.ESM.Projection, ecospat.ESM.EnsembleModeling, ecospat.ESM.EnsembleProjection, ecospat.ESM.EnsembleEvaluation, ecospat.ESM.threshold,ecospat.ESM.VarContrib, ecospat.ESM.responsePlot

BIOMOD_FormatingData, BIOMOD_ModelingOptions, BIOMOD_Modeling,BIOMOD_Projection

Examples

```
library(biomod2)

# Loading test data
data(ecospat.testNiche.inv)
inv <- ecospat.testNiche.inv

# species occurrences
xy <- inv[,1:2]
sp_occ <- inv[11]

# env data
current <- inv[3:8]

### Formating the data with the BIOMOD_FormatingData() function from the package biomod2
sp <- 1
myBiomodData <- BIOMOD_FormatingData( resp.var = as.numeric(sp_occ[,sp]),
                                      expl.var = current,
                                      resp.xy = xy,
                                      resp.name = colnames(sp_occ)[sp])

### Calibration of simple bivariate models
my.ESM <- ecospat.ESM.Modeling( data=myBiomodData,
                                models=c('GLM'),
                                NbRunEval=2,
                                DataSplit=70,
```

```
                                weighting.score=c("AUC"),
                                parallel=FALSE)

### Ensemble models
my.ESM_EF <- ecospat.ESM.EnsembleModeling(my.ESM,weighting.score=c("SomersD"),threshold=0)

### thresholds to produce binary maps
my.ESM_thresholds <- ecospat.ESM.threshold(my.ESM_EF)

### Evaluation of bivariate and ensemble models based on standard cross-validation
my.ESM_EF$ESM.evaluations
my.ESM_thresholds

### Evaluation of the ensemble models based on the pooling procedure
my.ESM_evaluations <- ecospat.ESM.EnsembleEvaluation(ESM.modeling.output= my.ESM,
                                                     ESM.EnsembleModeling.output = my.ESM_EF,
                                                          metrics= c("AUC","MaxTSS"),
                                                          EachSmallModels = FALSE)
my.ESM_evaluations$ESM.evaluations

### Projection of simple bivariate models into new space
my.ESM_proj_current<-ecospat.ESM.Projection(ESM.modeling.output=my.ESM,
                                            new.env=current)
### 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)
### Binary Projection based on max TSS of calibrated ESMs into new space
my.ESM_EFproj_current_binary <- (my.ESM_EFproj_current > (my.ESM_thresholds$TSS.th*1000))*1

## get the variable contributions of ESMs
ecospat.ESM.VarContrib(my.ESM,my.ESM_EF)

## get the response plots of ESMs
my.ESM_responsePlot<-ecospat.ESM.responsePlot(my.ESM_EF,my.ESM,fixed.var.metric = 'mean')
```

`ecospat.ESM.Projection` *Ensemble of Small Models: Projects Simple Bivariate Models Into New Space Or Time*

Description

This function projects simple bivariate models on new.env

Usage

```
ecospat.ESM.Projection(ESM.modeling.output,
                       new.env,
                       name.env,
```

```
                                parallel,
                                cleanup)
```

Arguments

ESM.modeling.output	list object returned by ecospat.ESM.Modeling
new.env	A set of explanatory variables onto which models will be projected. It could be a data.frame, a matrix, or a rasterStack object. Make sure the column names (data.frame or matrix) or layer Names (rasterStack) perfectly match with the names of variables used to build the models in the previous steps.
name.env	A name for the new.env object. If not specified (default) the name of the new.env object will be used. It is necessary to specify a unique name when projecting various new.env objects in a loop.
parallel	Logical. If TRUE, the parallel computing is enabled
cleanup	Numeric. Calls removeTmpFiles() to delete all files from rasterOptions()$tmpdir which are older than the given time (in hours). This might be necessary to prevent running over quota. No cleanup is used by default

Details

The basic idea of ensemble of small models (ESMs) is to model a species distribution based on small, simple models, for example all possible bivariate models (i.e. models that contain only two predictors at a time out of a larger set of predictors), and then combine all possible bivariate models into an ensemble (Lomba et al. 2010; Breiner et al. 2015).

The ESM set of functions could be used to build ESMs using simple bivariate models which are averaged using weights based on model performances (e.g. AUC) accoring to Breiner et al (2015). They provide full functionality of the approach described in Breiner et al. (2015).

The name of new.env must be a regular expression (see ?regex)

Value

Returns the projections for all selected models (same as in biomod2) See "BIOMOD.projection.out" for details.

Author(s)

Frank Breiner <frank.breiner@wsl.ch>

with contributions of Olivier Broennimann <olivier.broennimann@unil.ch>

References

Lomba, A., L. Pellissier, C.F. Randin, J. Vicente, F. Moreira, J. Honrado and A. Guisan. 2010. Overcoming the rare species modelling paradox: A novel hierarchical framework applied to an Iberian endemic plant. *Biological Conservation*, **143**,2647-2657.

Breiner F.T., A. Guisan, A. Bergamini and M.P. Nobis. 2015. Overcoming limitations of modelling rare species by using ensembles of small models. *Methods in Ecology and Evolution*, **6**,1210-1218.

Breiner F.T., Nobis M.P., Bergamini A., Guisan A. 2018. Optimizing ensembles of small models for predicting the distribution of species with few occurrences. *Methods in Ecology and Evolution*. doi:10.1111/2041210X.12957

See Also

ecospat.ESM.Modeling

`ecospat.ESM.responsePlot`

Produce response plots for ESMs

Description

This function creates response plots (evaluation strips) for Ensmebles of Small Models (ESMs).

Usage

```
ecospat.ESM.responsePlot( ESM.EnsembleModeling.output,
                          ESM.modeling.output,
                          fixed.var.metric = 'median')
```

Arguments

`ESM.modeling.output`
: a list object returned by ecospat.ESM.Modeling

`ESM.EnsembleModeling.output`
: a list object returned by ecospat.ESM.EnsembleModeling

`fixed.var.metric`
: either 'median' (default), 'mean', 'min' or 'max' specifying the statistic used to fix as constant the remaining variables when the predicted response is estimated for one of the variables. (same as in response.plot2

Details

This function plots the response curves of a model for each variable, while keeping the remianing variables constant. This is an adaptation of the Evaluation Strip method proposed by Elith et al.(2005)

Value

A plot of the response curves is produced (red line Ensemble, grey lines single algorithms) and a list with the output is provided.

Author(s)

Frank Breiner <frank.breiner@unil.ch>

References

Elith, J., Ferrier, S., Huettmann, FALSE. & Leathwick, J. R. 2005 The evaluation strip: A new and robust method for plotting predicted responses from species distribution models. Ecological Modelling 186, 280-289.

See Also

ecospat.ESM.Modeling

`ecospat.ESM.threshold` *Thresholds for Ensemble of Small Models*

Description

This function evaluates the full model which is used for projections and provides thresholds to produce binary maps.

Usage

```
ecospat.ESM.threshold( ESM.EnsembleModeling.output,
                       PEplot = FALSE)
```

Arguments

`ESM.EnsembleModeling.output`	a list object returned by `ecospat.ESM.EnsembleModeling`
`PEplot`	logical. Should the predicted to expected ratio along the suitability class from the boyce index be plotted. Default FALSE (see `ecospat.boyce`

Details

This function provides evaluation scores of the full model (no split sampling) and thresholds which can be used to convert suitability maps into binary maps. Various thresholds are provided: TSS (where sensitivity and specificity are maximised), MPA 1.0 (where all presences are prdicted positive), MPA 0.95 (where 95% of all presences are predicted positive), MPA 0.90 (where 90% of all presences are predicted positive), Boyce.th.min (the lowest suitability value where the predicted/expected ratio is >1) and Boyce.th.max (the highest suitability value where the predicted/expected ratio is =1).

Value

A data.frame with evluation scores and thresholds.

Author(s)

Frank Breiner <frank.breiner@unil.ch>

References

Hirzel, Alexandre H., et al. Evaluating the ability of habitat suitability models to predict species presences. *Ecological modelling*, **199.2** (2006): 142-152.

Engler, Robin, Antoine Guisan, and Luca Rechsteiner. An improved approach for predicting the distribution of rare and endangered species from occurrence and pseudo-absence data. *Journal of applied ecology*, **41.2** (2004): 263-274.

Fielding, Alan H., and John F. Bell. A review of methods for the assessment of prediction errors in conservation presence/absence models." *Environmental conservation*, **24.1** (1997): 38-49.

See Also

ecospat.ESM.Modeling

`ecospat.ESM.VarContrib`

Variable contribution in ESM

Description

calculates the variable contribution of each variable and method in an ESM model

Usage

```
ecospat.ESM.VarContrib(ESM.modeling.output,
                       ESM_EF.output)
```

Arguments

`ESM.modeling.output`	`BIOMOD.formated.data` object returned by `ecospat.ESM.Modeling`
`ESM_EF.output`	`BIOMOD.formated.data` object returned by `ecospat.ESM.EnsembleModeling`

Details

Calculates the ratio between sum of weights of bivariate models where a focal variable was used and sum of weights of bivariate models where the focal variable was not used. The ratio is corrected for the number of models with or without the focal variable. This ratio gives an indication on the proportional contribution of the variable in the final ensemble model. A value of higher than 1 indicate that the focal variable has a higher contribution than average. In the case of multiple methods (e.g., GLM, GAM...), the contributions are counted per method. For ensemble model, the contributions are then weighted means (based on the weighting score as chosen in ecospat.ESM.EnsembleModeling()) of single methods

Value

Returns a dataframe with contribution values (i.e. proportional contribution) by variable and model

Author(s)

Olivier Broennimann <Olivier.Broennimann@unil.ch> with contributions of Heidi Mod <Heidi.mod1@gmail.com> and Daniel Scherrer <daniel.j.a.scherrer@gmail.com>

See Also

ecospat.ESM.Modeling

---

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

Description

Create a grid with occurrence densities along one or two gridded environmental gradients.

Usage

```
ecospat.grid.clim.dyn (glob, glob1, sp, R, th.sp, th.env, geomask,
  kernel.method, extend.extent)
```

Arguments

glob	A two-column dataframe (or a vector) of the environmental values (in column) for background pixels of the whole study area (in row).
glob1	A two-column dataframe (or a vector) of the environmental values (in column) for the background pixels of the species (in row).
sp	A two-column dataframe (or a vector) of the environmental values (in column) for the occurrences of the species (in row).
R	The resolution of the grid.
th.sp	The quantile used to delimit a threshold to exclude low species density values.
th.env	The quantile used to delimit a threshold to exclude low environmental density values of the study area.
geomask	A geographical mask to delimit the background extent if the analysis takes place in the geographical space.It can be a SpatialPolygon or a raster object. Note that the CRS should be the same as the one used for the points.
kernel.method	Method used to estimate the the kernel density. Currently, there are two methods: by default, it is the methode from 'adehabitat'. Method from the library 'ks' is also available.
extend.extent	Vector with extention values of the window size (see details).

Details

Using the scores of an ordination (or SDM prediction), create a grid z of RxR pixels (or a vector of R pixels when using scores of dimension 1 or SDM predictions) with occurrence densities. Only scores of one, or two dimensions can be used. `th.sp` is the quantile of the distribution of species density at occurrence sites. For example, if `th.sp` is set to 0.05, the species niche is drawn by including 95 percent of the species occurrences, removing the more marginal populations. Similarly, `th.env` is the quantile of the distribution of the environmental density at all sites of the study area. If `th.env` is set to 0.05, the delineation of the study area in the environmental space includes 95 percent of the study area, removing the more marginal sites of the study area. By default, these thresholds are set to 0 but can be modified, depending on the importance of some marginal sites in the delineation of the species niche and/or the study area in the environmnental space. It is recommended to check if the shape of the delineated niche and study area corresponds to the shape of the plot of the PCA scores (or any other ordination techniques used to set the environmental space). Visualisation of the gridded environmental space can be done through the functions `ecospat.plot.niche` or `ecospat.plot.niche.dyn` If you encounter a problem during your analyses, please first read the FAQ section of "Niche overlap" in http://www.unil.ch/ecospat/home/menuguid/ecospat-resources/tools.html The argument `geomask` can be a SpatialPolygon or a raster object. Note that the CRS should be the same as the one used for the points.

The parameter `extend.extent` allows modifying the extent of the grid. By default, the window covers from the minimum to the maximum value of the environmental values present in the study area. The vector `extend.extent` indicates how much you want to shift the x-minimal, x-maximal, y-minimal and y-maximal values respectively.

Value

A grid z of RxR pixels (or a vector of R pixels) with z.uncor being the density of occurrence of the species, and z.cor the occupancy of the environment by the species (density of occurrences divided by the desinty of environment in the study area.

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch> and Blaise Petitpierre <bpetitpierre@gmail.com>

References

Broennimann, O., M.C. Fitzpatrick, P.B. Pearman, B. Petitpierre, L. Pellissier, N.G. Yoccoz, W. Thuiller, M.J. Fortin, C. Randin, N.E. Zimmermann, C.H. Graham and A. Guisan. 2012. Measuring ecological niche overlap from occurrence and spatial environmental data. *Global Ecology and Biogeography*, **21**:481-497.

Petitpierre, B., C. Kueffer, O. Broennimann, C. Randin, C. Daehler and A. Guisan. 2012. Climatic niche shifts are rare among terrestrial plant invaders. *Science*, **335**:1344-1348.

See Also

`ecospat.plot.niche.dyn`

Examples

```
library(ade4)
library(raster)

data(ecospat.testNiche)
data(ecospat.testData)

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 occurences per species

sp.list <- sp.list[sp.nbocc > 4] # remove species with less than 5 occurences
nb.sp <- length(sp.list) # nb of species
ls()
# 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 interation 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])
w <- c(rep(0, nrow(occ.sp)), rep(1, nrow(clim)))
pca.cal <- dudi.pca(data, row.w = w, center = TRUE, scale = TRUE, scannf = FALSE, nf = 2)

####### selection of species ######
sp.list
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 corresponding to sp1
 row.sp2 <- which(occ.sp[, 1] == sp.list[sp.combn[2, i]]) # rows in data corresponding 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$li[(nrow(occ.sp) + 1):nrow(data), ] # scores for global climate
  scores.sp1 <- pca.cal$li[row.sp1, ] # scores for sp1
  scores.sp2 <- pca.cal$li[row.sp2, ] # scores for sp2
}
# calculation of occurence density and test of niche equivalency and similarity
# with the default kernel method
z1 <- ecospat.grid.clim.dyn(scores.clim, scores.clim, scores.sp1, R = 100)
z2 <- ecospat.grid.clim.dyn(scores.clim, scores.clim, scores.sp2, R = 100)

# calculation of occurence density and test of niche equivalency and similarity
# with the ks kernel method
z1.kde <- ecospat.grid.clim.dyn(scores.clim, scores.clim, scores.sp1, R = 100, kernel.method = "ks")
z2.kde <- ecospat.grid.clim.dyn(scores.clim, scores.clim, scores.sp2, R = 100, kernel.method = "ks")

par(mfrow = c(2, 2))
plot(z1$z.uncor, main = "Sp1 with default kernel")
plot(z1.kde$z.uncor, main = "Sp1 with KS kernel")
plot(z2$z.uncor, main = "Sp2 with default kernel")
plot(z2.kde$z.uncor, main = "Sp2 with KS kernel")

z1.ext <- ecospat.grid.clim.dyn(scores.clim, scores.clim, scores.sp1, R = 100,
  extend.extent = c(0,1,0,0))

par(mfrow = c(1, 2))
plot(z1$z.uncor, main = "Sp1 with default extent")
plot(z1.ext$z.uncor, main = "Sp1 with extended extent")
```

`ecospat.makeDataFrame` *Make Data Frame*

Description

Create a biomod2-compatible dataframe. The function also enables to remove duplicate presences within a pixel and to set a minimum distance between presence points to avoid autocorrelation. Data from GBIF can be added.

Usage

```
ecospat.makeDataFrame (spec.list, expl.var, use.gbif=FALSE, precision=NULL,
year=NULL, remdups=TRUE, mindist=NULL, n=1000, type='random', PApoint=NULL,
ext=expl.var, tryf=5)
```

Arguments

`spec.list`	Data.frame or Character. The species occurrence information must be a data.frame in the form: \'x-coordinates\' , \'y-coordinates\' and \'species name\' (in the same projection/coordinate system as expl.var!).

expl.var	a RasterStack object of the environmental layers.
use.gbif	Logical. If TRUE presence data from GBIF will be added. It is also possible to use GBIF data only. Default: FALSE. See ?gbif dismo for more information. Settings: geo=TRUE, removeZeros=TRUE, all sub-taxa will be used. \'species name\' in spec.list must be in the form: \'genus species\', \'genus_species\' or \'genus.species\'. If there is no species information available on GBIF an error is returned. Try to change species name (maybe there is a synonym) or switch use.gbif off.
precision	Numeric. Use precision if use.gbif = TRUE to set a minimum precision of the presences which should be added. For precision = 1000 e.g. only presences with precision of at least 1000 meter will be added from GBIF. When precision = NULL all presences from GBIF will be used, also presences where precision information is NA.
year	Numeric. Latest year of the collected gbif occurrences. If year=1960 only occurrences which were collected since 1960 are used.
remdups	Logical. If TRUE (Default) duplicated presences within a raster pixel will be removed. You will get only one presence per pixel.
mindist	Numeric. You can set a minimum distance between presence points to avoid autocorrelation. nndist spatstat is used to calculate the nearest neighbour (nn) for each point. From the pair of the minimum nn, the point is removed of which the second neighbour is closer. Unit is the same as expl.var.
n	number of Pseudo-Absences. Default 1000.
type	sampling dessign for selecting Pseudo-Absences. If \'random\' (default) background points are selected with the function randomPoints dismo. When selecting another sampling type (\'regular\', \'stratified\', \'nonaligned\', \'hexagonal\', \'clustered\' or \'Fibonacci\') spsample sp will be used. This can immensely increase computation time and RAM usage if ext is a raster, especially for big raster layers because it must be converted into a \'SpatialPolygonsDataFrame\' first.
PApoint	data.frame or SpatialPoints. You can use your own set of Pseudo-Absences instead of generating new PAs. Two columns with \'x\' and \'y\' in the same projection/coordinate system as expl.var!
ext	a Spatial Object or Raster object. Extent from which PAs should be selected from (Default is expl.var).
tryf	numeric > 1. Number of trials to create the requested Pseudo-Absences after removing NA points (if type='random'). See ?randomPoints dismo

Details

If you use a raster stack as explanatory variable and you want to model many species in a loop with Biomod, formating data will last very long as presences and PA's have to be extracted over and over again from the raster stack. To save computation time, it is better to convert the presences and PAs to a data.frame first.

Value

A data.frame object which can be used for modeling with the Biomod package.

Author(s)

Frank Breiner <frank.breiner@unil.ch>

Examples

```
library(dismo)

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)

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

random.spec <- cbind(as.data.frame(randomPoints(predictors,50,extf=1)),species="randomSpec")
colnames(random.spec)[1:2] <- c("lon","lat")

spec.list <- rbind(bradypus, random.spec)

df <- ecospat.makeDataFrame(spec.list, expl.var=predictors, n=5000)
head(df)

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

---

ecospat.mantel.correlogram

Mantel Correlogram

Description

Investigate spatial autocorrelation of environmental covariables within a set of occurrences as a function of distance.

Usage

```
ecospat.mantel.correlogram (dfvar, colxy, n, colvar, max, nclass, nperm)
```

Arguments

dfvar	A dataframe object with the environmental variables.
colxy	The range of columns for x and y in df.
n	The number of random occurrences used for the test.
colvar	The range of columns for variables in df.
max	The maximum distance to be computed in the correlogram.
nclass	The number of classes of distances to be computed in the correlogram.
nperm	The number of permutations in the randomization process.

Details

Requires ecodist library. Note that computation time increase tremendously when using more than 500 occurrences (n>500)

Value

Draws a plot with distance vs. the mantel r value. Black circles indicate that the values are significative different from zero. White circles indicate non significant autocorrelation. The selected distance is at the first white circle where values are non significative different from cero.

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch>

References

Legendre, P. and M.J. Fortin. 1989. Spatial pattern and ecological analysis. *Vegetatio*, **80**, 107-138.

See Also

mgram

Examples

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

`ecospat.margin` *Delineation of the distribution's margin and its uncertainty*

Description

Estimate the margin of the distribution in the bi-dimensional environmental space based on bootstrapped kernel density estimation the percentage of the distribution included within the margin.

Usage

```
ecospat.margin (z, th.quant, kern.method, bootstrap, bootstrap.rep, bootstrap.ncore)
```

Arguments

`z`	object (list) resulting from the function `ecospat.grid.clim.dyn`
`th.quant`	The quantile (between 0 and 1) used to delimit a threshold to exclude low species density values (see details)
`kern.method`	Method used to draw the kernel density estimation. it can be "adehabitat" (default) or "ks"
`bootstrap`	Boolean. If `TRUE`, a confidence interval based on bootstrap is estimated for the margin of the distribution
`bootstrap.rep`	Number of resamplings if the boostrap is selected.
`bootstrap.ncore`	Number of cores to use for parallelization if the boostrap estimation is selected.

Details

`th.quant` is the quantile of the distribution of species density at occurrence sites. For example, if `th.quant` is set to 0.05, the margin of the distribution is drawn by including 95 percent of the species occurrences, removing the more marginal populations. If `th.quant` is set to 0, the margin of the distribution is the minimal envelope including 100 percent of the species occurrences. The bootstrap estimation of the margin allows representing the uncertainty around this margin. By default it returns a contour covering a 95 percents confidence interval (CI), but you can easily choose a custom CI.

Value

a list with $niche.envelope = a raster of the niche envelope including the distribution, $niche.contour = a SpatiaLine object of the margin, $niche.density = a raster of the niche density within the niche envelope.

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com>

See Also

ecospat.plot.niche.dyn

Examples

```
library(ade4)
library(raster)

data(ecospat.testData)

sp1 <- ecospat.testData[ecospat.testData$Bromus_erectus == 1, 1:8] # species occurences
clim <- ecospat.testData[4:8] # environment of the study area

################################### PCA-ENVIRONMENT ##################################
pca.cal <- dudi.pca(clim, center = TRUE, scale = TRUE, scannf = FALSE, nf = 2)

####### projection of the species distribution into the environmental space ######
# predict the scores on the axes
scores.clim <- pca.cal$li # scores for global climate
scores.sp1 <- suprow(pca.cal, sp1[, 4:8])$li # scores for sp1

z1 <- ecospat.grid.clim.dyn(scores.clim, scores.clim, scores.sp1, R = 100,
  extend.extent = c(0, 1, 0, 0))

####### estimate the margin ######
z1.margin <- ecospat.margin(z1, th.quant = 0, bootstrap = FALSE) # including all occurrences
z1.95margin <- ecospat.margin(z1, th.quant = 0.05,
  bootstrap = FALSE) # including 95 percent of the occurrences
z1.bootstrap.margin <- ecospat.margin(z1,th.quant = 0,
  bootstrap = TRUE, bootstrap.rep = 100) # bootstrap estimation of the niche limit

####### plot the margin and its uncertainty ######
plot(z1.margin$niche.density) # plot of the niche density
points(z1$sp) # with species occurences
plot(z1.margin$niche.contour, add = TRUE) # limit of the margin if you include all the distribution
plot(z1.95margin$niche.contour, col = 2, add = TRUE) # limit of the margin if you exclude the
                                          # 5 percents of the most marginal distribution
plot(z1.bootstrap.margin$niche.contour, col = 2, add = TRUE, lty = "dotted")
      # limit of the niche based on the 95 percent CI of the bootstrap

par(mfrow=c(1,2))
plot(z1.bootstrap.margin$niche.envelope, main = "Margin uncertainty",
 legend.args=list(text="CI", cex = 0.8)) # shows the uncertainty of the niche margin in raster mode
points(z1$sp) # with species occurences
# shows the uncertainty of the niche limit in vector mode
raster::contour(z1.bootstrap.margin$niche.envelope, col = gray(10:1 / 10),
  main = "Margin uncertainty")
confInt80 <- raster::rasterToPolygons(z1.bootstrap.margin$niche.envelope >= 50, dissolve = TRUE,
  fun = function(x) {x == 1}
  ) # select a customized confidence interval (here for example 80 percent)
```

```
plot(as(confInt80, "SpatialLines"), add = TRUE, lty = "dotted", col = 2)
```

`ecospat.max.kappa` *Maximum Kappa*

Description

Calculates values for Cohen's Kappa along different thresholds, considering this time 0.01 increments (i.e. 99 thresholds).

Usage

```
ecospat.max.kappa(Pred, Sp.occ)
```

Arguments

`Pred`	A vector of predicted probabilities
`Sp.occ`	A vector of binary observations of the species occurrence

Value

Return values for Cohen's Kappa for 99 thresholds at 0.01 increments.

Author(s)

Antoine Guisan <antoine.guisan@unil.ch> with contributions of Luigi Maiorano <luigi.maiorano@gmail.com> and Valeria Di Cola <valeria.dicola@unil.ch>.

References

Liu, C., P.M. Berry, T.P. Dawson, and R.G. Pearson. 2005. Selecting thresholds of occurrence in the prediction of species distributions. *Ecography*, **28**, 385-393.

See Also

`ecospat.meva.table`, `ecospat.max.tss`, `ecospat.plot.tss`, `ecospat.cohen.kappa`, `ecospat.plot.kappa`

Examples

```
Pred <- ecospat.testData$glm_Agrostis_capillaris
Sp.occ <- ecospat.testData$Agrostis_capillaris
kappa100 <- ecospat.max.kappa(Pred, Sp.occ)
```

ecospat.max.tss *Maximum TSS*

Description

Calculates values for True skill statistic (TSS) along different thresholds, considering this time 0.01 increments (i.e. 99 thresholds).

Usage

```
ecospat.max.tss(Pred, Sp.occ)
```

Arguments

Pred	A vector of predicted probabilities
Sp.occ	A vector of binary observations of the species occurrence

Value

Return values for TSS for 99 thresholds at 0.01 increments.

Author(s)

Luigi Maiorano <luigi.maiorano@gmail.com> with contributions of Antoine Guisan <antoine.guisan@unil.ch>

References

Liu, C., P.M. Berry, T.P. Dawson, and R.G. Pearson. 2005. Selecting thresholds of occurrence in the prediction of species distributions. *Ecography*, **28**, 385-393.

See Also

ecospat.meva.table, ecospat.max.kappa, ecospat.plot.tss, ecospat.cohen.kappa, ecospat.plot.kappa

Examples

```
data(ecospat.testData)
Pred <- ecospat.testData$glm_Agrostis_capillaris
Sp.occ <- ecospat.testData$Agrostis_capillaris
TSS100 <- ecospat.max.tss(Pred, Sp.occ)
```

ecospat.maxentvarimport

Maxent Variable Importance

Description

Calculate the importance of variables for Maxent in the same way Biomod does, by randomly permuting each predictor variable independently, and computing the associated reduction in predictive performance.

Usage

```
ecospat.maxentvarimport (model, dfvar, nperm)
```

Arguments

model	The name of the maxent model.
dfvar	A dataframe object with the environmental variables.
nperm	The number of permutations in the randomization process. The default is 5.

Details

The calculation is made as biomod2 "variables_importance" function. It's more or less base on the same principle than randomForest variables importance algorithm. The principle is to shuffle a single variable of the given data. Make model prediction with this 'shuffled' data.set. Then we compute a simple correlation (Pearson's by default) between references predictions and the 'shuffled' one. The return score is 1-cor(pred_ref,pred_shuffled). The highest the value, the more influence the variable has on the model. A value of this 0 assumes no influence of that variable on the model. Note that this technique does not account for interactions between the variables.

Value

a `list` which contains a `data.frame` containing variables importance scores for each permutation run.

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com>

Examples

```
library(dismo)
data('ecospat.testData')

# data for Soldanella alpina
data.Solalp<- ecospat.testData[c("Soldanella_alpina","ddeg","mind","srad","slp","topo")]
```

```
# path to maxent.jar file
path<- paste0(system.file(package="dismo"), "/java/maxent.jar")

if (file.exists(path) & require(rJava)) {
  me <- maxent(data.Solalp[,-1],data.Solalp[,1])
  ecospat.maxentvarimport (model=me, dfvar=data.Solalp[,-1], nperm=5)
  }
```

---

ecospat.mdr *Minimum Dispersal Routes)*

Description

ecospat.mdr is a function that implement a minimum cost arborescence approach to analyse the invasion routes of a species from dates occurrence data.

Usage

```
ecospat.mdr (data, xcol, ycol, datecol, mode, rep, mean.date.error, fixed.sources.rows)
```

Arguments

data	dataframe with occurence data. Each row correspond to an occurrence.
xcol	The column in data containing x coordinates.
ycol	The column in data containing y coordinates.
datecol	The column in data containing dates.
mode	"observed", "min" or "random". "observed" calculate routes using real dates. "min" reorder dates so the the total length of the routes are minimal. "random" reatribute dates randomly.
rep	number of iteration of the analyse. if > 1, boostrap support for each route is provided.
mean.date.error	mean number of years to substract to observed dates. It is the mean of the truncated negative exponential distribution from which the time to be substracted is randomly sampled.
fixed.sources.rows	the rows in data (as a vector) corresponding to source occurrence(s) that initiated the invasion(s). No incoming routes are not calculated for sources.

Details

The function draws an incoming route to each occurence from the closest occurrence already occupied (with a previous date) and allows to substract a random number of time (years) to the observed dates from a truncated negative exponential distribution. It is possible to run several iterations and to get boostrap support for each route. itexp and rtexp functions are small internal functions to set a truncated negative exponential distribution.

Value

A list is returned by the function with in positon [[1]], a datafame with each row corresponding to a route (with new/old coordinates, new/old dates, length of the route, timespan, dispersal rate), in position [[2]] the total route length, in position [[3]] the median dispersal rate and in position [[4]] the number of outgoing nodes (index of clustering of the network)

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch>

References

Hordijk, W. and O. Broennimann. 2012. Dispersal routes reconstruction and the minimum cost arborescence problem. *Journal of theoretical biology*, **308**, 115-122.

Broennimann, O., P. Mraz, B. Petitpierre, A. Guisan, and H. Muller-Scharer. 2014. Contrasting spatio-temporal climatic niche dynamics during the eastern and western invasions of spotted knapweed in North America.*Journal of biogeography*, **41**, 1126-1136.

Examples

```
if(require("maps",quietly=TRUE)){

data(ecospat.testMdr)
data<- ecospat.testMdr
intros<-order(data$date)[1:2] # rows corresponding to first introductions

# plot observed situation

plot(data[,2:1],pch=15,cex=0.5)
points(data[intros,2:1],pch=19,col="red")
text(data[,2]+0.5,data[,1]+0.5,data[,3],cex=0.5)
map(add=TRUE)

# calculate minimum cost arborescence (MCA) of dispersal routes

obs<-ecospat.mdr(data=data,xcol=2,ycol=1,datecol=3,mode="min",rep=100,
                  mean.date.error=1,fixed.sources.rows=intros)

# plot MCA results
# arrows' thickness indicate support for the routes

mca<-obs[[1]]
plot(mca[,3:4],type="n",xlab="longitude",ylab="latitude")
arrows(mca[,1],mca[,2],mca[,3],mca[,4],length = 0.05,lwd=mca$bootstrap.value*2)
map(add=TRUE)

# plot intros

points(data[intros,2:1],pch=19,col="red")
text(data[intros,2]+0.5,data[intros,1]+0.5,data[intros,3],cex=1,col="red")
```

```
# dispersal routes statistics

obs[[2]] # total routes length in DD
obs[[3]] # median dispersal rate in DD/yr
obs[[4]] # number of outcoming nodes
}
```

---

ecospat.mess *MESS*

Description

Calculate the MESS (i.e. extrapolation) as in Maxent.

Usage

```
ecospat.mess (proj, cal, w="default")
```

Arguments

proj	A dataframe object with x, y and environmental variables, used as projection dataset.
cal	A dataframe object with x, y and environmental variables, used as calibration dataset.
w	The weight for each predictor (e.g. variables importance in SDM).

Details

Shows the variable that drives the multivariate environmental similarity surface (MESS) value in each grid cell.

Value

MESS	The mess as calculated in Maxent, i.e. the minimal extrapolation values.
MESSw	The sum of negative MESS values corrected by the total number of predictors. If there are no negative values, MESSw is the mean MESS.
MESSneg	The number of predictors on which there is extrapolation.

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com>. Modified by Daniel Scherrer <daniel.j.a.scherrer@gmail.com>

References

Elith, J., M. Kearney and S. Phillips. 2010. The art of modelling range-shifting species. *Methods in ecology and evolution*, **1**, 330-342.

See Also

ecospat.plot.mess

Examples

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

#Plot MESS
ecospat.plot.mess (mess.object, cex=1, pch=15)
```

---

ecospat.meva.table *Model Evaluation For A Given Threshold Value*

Description

Calculates values of a series of different evaluations metrics for a model and for a given threshold value

Usage

```
ecospat.meva.table(Pred, Sp.occ, th)
```

Arguments

Pred	A vector of predicted probabilities
Sp.occ	A vector of binary observations of the species occurrence
th	Threshold used to cut the probability to binary values

Value

A contingency table of observations and predicted probabilities of presence values, and a list of evaluation metrics for the selected threshold.

Author(s)

Antoine Guisan <antoine.guisan@unil.ch> with contributions of Luigi Maiorano <luigi.maiorano@gmail.com>

References

Pearce, J. and S. Ferrier. 2000. Evaluating the predictive performance of habitat models developed using logistic regression. *Ecol. Model.*, **133**, 225-245.

See Also

ecospat.max.kappa, ecospat.max.tss, ecospat.plot.tss, ecospat.cohen.kappa, ecospat.plot.kappa

Examples

```
Pred <- ecospat.testData$glm_Agrostis_capillaris
Sp.occ <- ecospat.testData$Agrostis_capillaris

meva <- ecospat.meva.table (Pred, Sp.occ, 0.39)
```

---

ecospat.mpa *Minimal Predicted Area*

Description

Calculate the minimal predicted area.

Usage

```
ecospat.mpa (Pred, Sp.occ.xy, perc)
```

Arguments

Pred	Numeric or RasterLayer predicted suitabilities from a SDM prediction.
Sp.occ.xy	xy-coordinates of the species (if Pred is a RasterLayer).
perc	Percentage of Sp.occ.xy that should be encompassed by the binary map.

Details

The minimal predicted area (MPA) is the minimal surface obtained by considering all pixels with predictions above a defined probability threshold (e.g. 0.7) that still encompasses 90 percent of the species‘ occurrences (Engler et al. 2004).

Value

Returns the minimal predicted area.

Author(s)

Frank Breiner <frank.breiner@wsl.ch>

References

Engler, R., A. Guisan and L. Rechsteiner. 2004. An improved approach for predicting the distribution of rare and endangered species from occurrence and pseudo-absence data. *Journal of Applied Ecology*, **41**, 263-274.

Examples

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

ecospat.mpa(obs)
ecospat.mpa(obs,perc=1) ## 100 percent of the presences encompassed
```

`ecospat.niche.dyn.index`

Niche Expansion, Stability, and Unfilling

Description

Calculate niche expansion, stability and unfilling.

Usage

```
ecospat.niche.dyn.index (z1, z2, intersection=NA)
```

Arguments

`z1`	A gridclim object for the native distribution.
`z2`	A gridclim object for the invaded range.
`intersection`	The quantile of the environmental density used to remove marginal climates. If `intersection=NA`, the analysis is performed on the whole environmental extent (native and invaded). If `intersection=0`, the analysis is performed at the intersection between native and invaded range. If `intersection=0.05`, the analysis is performed at the intersection of the 5th quantile of both native and invaded environmental densities.

Details

If you encounter a problem during your analyses, please first read the FAQ section of "Niche overlap" in http://www.unil.ch/ecospat/home/menuguid/ecospat-resources/tools.html

Value

A list of dynamic indices: dynamic.index.w [expansion.index.w, stability.index.w, restriction.index.w]

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com>

See Also

`ecospat.grid.clim.dyn`

ecospat.niche.dynIndexProjGeo

Projection of niche dynamic indices to the Geography

Description

Creates a raster in geography with each pixel containing a niche dynamic index (stability, expansion, or unfilling) extracted from 2 niches generated with ecospat.grid.clim.dyn.

Usage

```
ecospat.niche.dynIndexProjGeo(z1,z2,env,index)
```

Arguments

z1	Species 1 occurrence density grid created by ecospat.grid.clim.dyn.
z2	Species 2 occurrence density grid created by ecospat.grid.clim.dyn.
env	A RasterStack or RasterBrick of environmental variables corresponding to the background (glob in ecospat.grid.clim.dyn).
index	"stability", "unfilling" or "expansion"

Details

extracts the niche dynamic index of objects created by ecospat.niche.dyn.index for each point of the background (glob) using extract (package raster). The values are binded to the geographic coordinates of env and a raster is then recreated using rasterFromXYZ

Value

raster of class RasterLayer

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch>

References

Broennimann, O., M.C. Fitzpatrick, P.B. Pearman, B. Petitpierre, L. Pellissier, N.G. Yoccoz, W. Thuiller, M.J. Fortin, C. Randin, N.E. Zimmermann, C.H. Graham and A. Guisan. 2012. Measuring ecological niche overlap from occurrence and spatial environmental data. *Global Ecology and Biogeography*, **21**:481-497.

Petitpierre, B., C. Kueffer, O. Broennimann, C. Randin, C. Daehler and A. Guisan. 2012. Climatic niche shifts are rare among terrestrial plant invaders. *Science*, **335**:1344-1348.

See Also

ecospat.plot.niche.dyn,ecospat.niche.dyn.index, ecospat.niche.zProjGeo

Examples

```
library(ade4)
library(raster)

data(ecospat.testNiche)
spp <- ecospat.testNiche
xy.sp1<-subset(spp,species=="sp1")[2:3] #Bromus_erectus
xy.sp2<-subset(spp,species=="sp3")[2:3] #Daucus_carota

load(system.file("extdata", "ecospat.testEnvRaster.Rdata", package="ecospat"))

env.sp1<-extract(env,xy.sp1)
env.sp2<-extract(env,xy.sp2)
env.bkg<-na.exclude(values(env))

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

pca.cal <- dudi.pca(env.bkg, center = TRUE, scale = TRUE, scannf = FALSE, nf = 2)

# predict the scores on the axes
scores.bkg <- pca.cal$li  #scores for background climate
scores.sp1 <- suprow(pca.cal,env.sp1)$lisup #scores for sp1
scores.sp2 <- suprow(pca.cal,env.sp2)$lisup #scores for sp2

# calculation of occurence density (niche z)

z1 <- ecospat.grid.clim.dyn(scores.bkg, scores.bkg, scores.sp1,R=100)
z2 <- ecospat.grid.clim.dyn(scores.bkg, scores.bkg, scores.sp2,R=100)

plot(z1$z.uncor)
points(scores.sp1)

plot(z2$z.uncor)
points(scores.sp2)

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

################################### stability S in space ##################################

geozS<-ecospat.niche.dynIndexProjGeo(z1,z2,env,index="stability")

plot(geozS,main="Stability")
points(xy.sp1,col="red")
points(xy.sp2,col="blue")
```

---

ecospat.niche.equivalency.test

Niche Equivalency Test

Description

Run a niche equivalency test (see Warren et al 2008) based on two species occurrence density grids.

Usage

```
ecospat.niche.equivalency.test (z1, z2, rep, intersection = NA,
overlap.alternative = "higher", expansion.alternative = "lower",
stability.alternative = "higher", unfilling.alternative = "lower", ncores = 1)
```

Arguments

z1	Species 1 occurrence density grid created by ecospat.grid.clim.
z2	Species 2 occurrence density grid created by ecospat.grid.clim.
rep	The number of replications to perform.
intersection	The quantile of the environmental density used to remove marginal climates. See Details.
overlap.alternative	To indicate the alternative hypothesis of the test. See Details.
expansion.alternative	To indicate the alternative hypothesis of the test. See Details.
stability.alternative	To indicate the alternative hypothesis of the test. See Details.
unfilling.alternative	To indicate the alternative hypothesis of the test. See Details.
ncores	The number of cores used for parallelisation.

Details

Compares the observed niche overlap, expansion, stability and unfilling between z1 and z2 to simulated values between random niches z1.sim and z2.sim, which are built from random reallocations of occurences of z1 and z2.

intersection allows setting if the niche dynamic indices (expansion, stability and unfilling) are measured across the full extent pooling the two study areas or not. If intersection = NA, the analysis is performed on the whole environmental extent (native and invaded). If instersection = 0, the analysis is performed at the intersection between native and invaded range. If instersection = 0.05, the analysis is performed at the intersection of the 5th quantile of both native and invaded environmental densities. Etc...

overlap.alternative specifies if you want to test for niche conservatism (overlap.alternative = "higher", i.e. the niche overlap is more equivalent/similar than random) or for niche divergence (overlap.alternative = "lower", i.e. the niche overlap is less equivalent/similar than random). You can also specifiy if you want to test if you have more, less or different observed niche dynamics than random niches(with expansion.alternative, stability.alternative and unfilling.alternative). If you want to test for niche conservatism, we recommande to set these niche dynamic hypotheses to "lower", "higher" and "lower" respectively for expansion, stability and unfilling.

If you encounter a problem during your analyses, please first read the FAQ section of "Niche overlap" in http://www.unil.ch/ecospat/home/menuguid/ecospat-resources/tools.html

The arguments ncores allows choosing the number of cores used to parallelize the computation. The default value is 1. On multicore computers, the optimal would be ncores = detectCores() - 1.

Value

a list with $obs = observed overlap and dynamic indices, $sim = simulated overlap and dynamic indices, $p.D = p-value of the test on D, $p.I = p-value of the test on I, $p.expansion = p-value for the test on expansion, $p.stability = p-value for the test on stability, $p.unfilling = p-value for the test on unfilling.

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch> with contributions of Blaise Petitpierre <bpetitpierre@gmail.com>

References

Broennimann, O., M.C. Fitzpatrick, P.B. Pearman,B. Petitpierre, L. Pellissier, N.G. Yoccoz, W. Thuiller, M.J. Fortin, C. Randin, N.E. Zimmermann, C.H. Graham and A. Guisan. 2012. Measuring ecological niche overlap from occurrence and spatial environmental data. *Global Ecology and Biogeography*, **21**, 481-497.

Warren, D.L., R.E. Glor and M. Turelli. 2008. Environmental niche equivalency versus conservatism: quantitative approaches to niche evolution. *Evolution*, **62**, 2868-2883.

See Also

ecospat.grid.clim.dyn, ecospat.niche.similarity.test

---

ecospat.niche.overlap *Calculate Niche Overlap*

Description

Calculate the overlap metrics D and I based on two species occurrence density grids z1 and z2 created by ecospat.grid.clim.

Usage

```
ecospat.niche.overlap (z1, z2, cor)
```

Arguments

z1	Species 1 occurrence density grid created by ecospat.grid.clim.
z2	Species 2 occurrence density grid created by ecospat.grid.clim.
cor	Correct the occurrence densities of each species by the prevalence of the environments in their range (TRUE = yes, FALSE = no).

Details

if cor=FALSE, the z$uncor objects created by `ecospat.grid.clim` are used to calculate the overlap. if cor=TRUE, the z$cor objects are used.

If you encounter a problem during your analyses, please first read the FAQ section of "Niche overlap" in http://www.unil.ch/ecospat/home/menuguid/ecospat-resources/tools.html

Value

Overlap values D and I. D is Schoener's overlap metric (Schoener 1970). I is a modified Hellinger metric(Warren et al. 2008)

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch>

References

Broennimann, O., M.C. Fitzpatrick, P.B. Pearman, B. Petitpierre, L. Pellissier, N.G. Yoccoz, W. Thuiller, M.J. Fortin, C. Randin, N.E. Zimmermann, C.H. Graham and A. Guisan. 2012. Measuring ecological niche overlap from occurrence and spatial environmental data. *Global Ecology and Biogeography*, **21**, 481-497.

Schoener, T.W. 1968. Anolis lizards of Bimini: resource partitioning in a complex fauna. *Ecology*, **49**, 704-726.

Warren, D.L., R.E. Glor and M. Turelli. 2008. Environmental niche equivalency versus conservatism: quantitative approaches to niche evolution. *Evolution*, **62**, 2868-2883.

See Also

`ecospat.grid.clim.dyn`

---

ecospat.niche.similarity.test

Niche Similarity Test

Description

Run a niche similarity test (see Warren et al 2008) based on two species occurrence density grids.

Usage

```
ecospat.niche.similarity.test (z1, z2, rep, intersection = NA,
  rand.type = 1, ncores= 1, overlap.alternative = "higher",
  expansion.alternative = "lower", stability.alternative = "higher",
  unfilling.alternative = "lower")
```

Arguments

z1	Species 1 occurrence density grid created by ecospat.grid.clim.
z2	Species 2 occurrence density grid created by ecospat.grid.clim.
rep	Number of replications to perform.
intersection	Quantile of the environmental density used to remove marginal climates. See Details.
rand.type	Type of randomization on the density grids (1 or 2).
ncores	Number of cores used for parallelisation.
overlap.alternative	Alternative hypothesis of the test. See Details.
expansion.alternative	Alternative hypothesis of the expansion test. See Details.
stability.alternative	Alternative hypothesis of the stability test. See Details.
unfilling.alternative	Alternative hypothesis of the unfilling test. See Details.

Details

Compares the observed niche overlap between z1 and z2 to overlaps between z1 and random niches (z2.sim) as available in the range of z2 (z2$Z). z2.sim has the same pattern as z2 but the center is randomly translatated in the availabe z2$Z space and weighted by z2$Z densities.

intersection allows setting if the niche dynamic indices (expansion, stability and unfilling) are measured across the full extent pooling the two study areas or not. If intersection = NA, the analysis is performed on the whole environmental extent (native and invaded). If instersection = 0, the analysis is performed at the intersection between native and invaded range. If instersection = 0.05, the analysis is performed at the intersection of the 5th quantile of both native and invaded environmental densities. Etc...

If rand.type = 1, both z1 and z2 are randomly shifted, if rand.type = 2, only z2 is randomly shifted.

overlap.alternative specifies if you want to test for niche conservatism (overlap.alternative = "higher", i.e. the niche overlap is more equivalent/similar than random) or for niche divergence (overlap.alternative = "lower", i.e. the niche overlap is less equivalent/similar than random). You can also specifiy if you want to test if you have more, less or different observed niche dynamics than random niches(with expansion.alternative, stabilty.alternative and unfilling.alternative). If you want to test for niche conservatism, we recommande to set these niche dynamic hypotheses to "lower", "higher" and "lower" respectively for expansion, stability and unfilling.

If you encounter a problem during your analyses, please first read the FAQ section of "Niche overlap" in http://www.unil.ch/ecospat/home/menuguid/ecospat-resources/tools.html

The arguments ncores allows choosing the number of cores used to parallelize the computation. The default value is 1. On multicore computers, the optimal would be ncores = detectCores() - 1.

Value

a list with $obs = observed overlap and dynamic indices, $sim = simulated overlap and dynamic indices, $p.D = p-value of the test on D, $p.I = p-value of the test on I, $p.expansion = p-value for the test on expansion, $p.stability = p-value for the test on stability, $p.unfilling = p-value for the test on unfilling.

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch> with contributions of Blaise Petitpierre <bpetitpierre@gmail.com>

References

Broennimann, O., M.C. Fitzpatrick, P.B. Pearman, B. Petitpierre, L. Pellissier, N.G. Yoccoz, W. Thuiller, M.J. Fortin, C. Randin, N.E. Zimmermann, C.H. Graham and A. Guisan. 2012. Measuring ecological niche overlap from occurrence and spatial environmental data. *Global Ecology and Biogeography*, **21**, 481-497.

Warren, D.L., R.E. Glor and M. Turelli. 2008. Environmental niche equivalency versus conservatism: quantitative approaches to niche evolution. *Evolution*, **62**, 2868-2883.

See Also

ecospat.grid.clim.dyn, ecospat.niche.equivalency.test

---

ecospat.niche.zProjGeo

Projection of Occurrence Densities to the Geography

Description

Creates a raster in geography with each pixel containing the occurrence densities extracted from a z object generated with ecospat.grid.clim.dyn.

Usage

```
ecospat.niche.zProjGeo(z1,env,cor)
```

Arguments

z1	Species 1 occurrence density grid created by ecospat.grid.clim.dyn.
env	A RasterStack or RasterBrick of environmental variables corresponding to the background (glob in ecospat.grid.clim.dyn).
cor	FALSE by default. If TRUE corrects the occurrence densities of each species by the prevalence of the environments in their range

Details

extracts the occurrence density of z objects created by `ecospat.grid.clim.dyn` for each point of the background (`glob`) using `extract` (package raster). The values are binded to the geographic coordinates of `env` and a raster is then recreated using `rasterFromXYZ`

Value

raster of class RasterLayer

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch>

References

Broennimann, O., M.C. Fitzpatrick, P.B. Pearman, B. Petitpierre, L. Pellissier, N.G. Yoccoz, W. Thuiller, M.J. Fortin, C. Randin, N.E. Zimmermann, C.H. Graham and A. Guisan. 2012. Measuring ecological niche overlap from occurrence and spatial environmental data. *Global Ecology and Biogeography*, **21**:481-497.

Petitpierre, B., C. Kueffer, O. Broennimann, C. Randin, C. Daehler and A. Guisan. 2012. Climatic niche shifts are rare among terrestrial plant invaders. *Science*, **335**:1344-1348.

See Also

`ecospat.plot.niche.dyn`, `ecospat.niche.dynIndexProjGeo`

Examples

```
library(ade4)
library(raster)

spp <- ecospat.testNiche
xy.sp1<-subset(spp,species=="sp1")[2:3] #Bromus_erectus

load(system.file("extdata", "ecospat.testEnvRaster.Rdata", package="ecospat"))
#?ecospat.testEnvRaster

env.sp1<-extract(env,xy.sp1)
env.bkg<-na.exclude(values(env))

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

pca.cal <- dudi.pca(env.bkg, center = TRUE, scale = TRUE, scannf = FALSE, nf = 2)

# predict the scores on the axes
scores.bkg <- pca.cal$li  #scores for background climate
scores.sp1 <- suprow(pca.cal,env.sp1)$lisup #scores for sp1

# calculation of occurence density (niche z)
```

```
z1 <- ecospat.grid.clim.dyn(scores.bkg, scores.bkg, scores.sp1,R=100)

plot(z1$z.uncor)
points(scores.sp1)

################################### occurrence density in space ###################################

# sp1
geoz1<-ecospat.niche.zProjGeo(z1,env)
plot(geoz1,main="z1")
points(xy.sp1)
```

`ecospat.nicheNBmean` *Niche position and niche breadth)*

Description

ecospat.nicheNBmean calculates the weighted mean niche breadth accross several axes from a nichePOSNB object.

Usage

```
ecospat.nicheNBmean(POSNB,w)
```

Arguments

`POSNB`	an object created by the function ecospat.nichePOSNB
`w`	a vector with a weight for each environmental axes.

Details

The function calculates the weighted mean niche breadth. The weights w are rescaled so that their sum equals 1.

Value

The function returns a 1 column dataframe with the weighted mean niche breadth for each taxa.

Author(s)

Lucie Malard <lucie.malard@unil.ch> and Olivier Broennimann <olivier.broennimann@unil.ch>

References

L.A. Malard, H.K. Mod, N. Guex, O. Broennimann, E. Yashiro, E. Lara, E.D.A. Mitchell, H. Niculita-Hirzel & A. Guisan. The ecological niche of soil bacterial, archaeal, fungal and protist communities along environmental gradients in the Alps. 2021. Accepted in Soil Biology and Biochemistry.

Examples

```
data(ecospat.testNichePOSNB)
df<-ecospat.testNichePOSNB

# 1 axes
POSNB<-ecospat.nichePOSNB(df,colvar=c(2),colfreq = 6:17)

# 2 axes
POSNB<-ecospat.nichePOSNB(df,colvar=c(2:3),colfreq = 6:17)
ecospat.nicheNBmean(POSNB,w=c(2,1))

# 4 axes
POSNB<-ecospat.nichePOSNB(df,colvar=c(2:5),colfreq = 6:17)
ecospat.nicheNBmean(POSNB,w=c(1,0.8,0.2,0.1))
```

---

ecospat.nichePOSNB *Niche position and niche breadth)*

Description

ecospat.nichePOSNB calculates the niche breadth and niche position of taxa along environmental gradients from abundance data.

Usage

```
ecospat.nichePOSNB (df,colvar,colfreq)
```

Arguments

df	dataframe with (relative) abundance data. Each row correspond to an abundance.
colvar	The column(s) in df corresponding to environmental axe(s).
colfreq	The columns in df corresponding to taxa frequencies.

Details

The function calculates niche position and niche breadth of taxa along one or multiple environmental axes. Niche position is calculated as the mean of the variable, weighted by the relative abundance of the species. Niche breadth is calculated as the standard deviation of each variable, weighted by the relative abundance of the species at each sampling site.

Value

The function returns a matrix containing the average niche position and niche breadth of each taxa along each environmental axi.

Author(s)

Lucie Malard <lucie.malard@unil.ch> and Olivier Broennimann <olivier.broennimann@unil.ch>

References

L.A. Malard, H.K. Mod, N. Guex, O. Broennimann, E. Yashiro, E. Lara, E.D.A. Mitchell, H. Niculita-Hirzel & A. Guisan. The ecological niche of soil bacterial, archaeal, fungal and protist communities along environmental gradients in the Alps. 2021. Accepted in Soil Biology and Biochemistry.

Examples

```
data(ecospat.testNichePOSNB)
df<-ecospat.testNichePOSNB

# 1 axes
POSNB<-ecospat.nichePOSNB(df,colvar=c(2),colfreq = 6:17)

# 2 axes
POSNB<-ecospat.nichePOSNB(df,colvar=c(2:3),colfreq = 6:17)
ecospat.nicheNBmean(POSNB,w=c(2,1))

# 4 axes
POSNB<-ecospat.nichePOSNB(df,colvar=c(2:5),colfreq = 6:17)
ecospat.nicheNBmean(POSNB,w=c(1,0.8,0.2,0.1))
```

---

ecospat.npred *Number Of Predictors*

Description

Calculate the maximum number of predictors to include in the model with a desired correlation between predictors.

Usage

```
ecospat.npred (x, th)
```

Arguments

x	Correlation matrix of the predictors.
th	Desired threshold of correlation between predictors.

Value

Returns the number of predictors to use.

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com>

Examples

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

---

ecospat.occ.desaggregation

Species Occurrences Desaggregation

Description

Remove species occurrences in a dataframe which are closer to each other than a specified distance threshold.

Usage

```
ecospat.occ.desaggregation (xy, min.dist, by)
```

Arguments

xy	A dataframe with xy-coordinates (x-column must be named 'x' and y-column 'y')
min.dist	The minimun distance between points in the sub-dataframe.
by	Grouping element in the dataframe (e.g. species, NULL)

Details

This function will desaggregate the original number of occurrences, according to a specified distance.

Value

A subset of the initial dataframe. The number of points is printed as "initial", "kept" and "out".

Author(s)

Frank Breiner <frank.breiner@unil.ch>

with contributions of Olivier Broennimann <olivier.broennimann@unil.ch>

Examples

```
spp <- ecospat.testNiche
colnames(spp)[2:3] <- c('x','y')
sp1 <- spp[1:32,2:3]

occ.sp1 <- ecospat.occ.desaggregation(xy=sp1, min.dist=500, by=NULL)
occ.all.sp <- ecospat.occ.desaggregation(xy=spp, min.dist=500, by='Spp')
```

---

ecospat.occupied.patch

Extract occupied patches of a species in geographic space.)

Description

This function determines the occupied patch of a species using standard IUCN criteria (AOO, EOO) or predictive binary maps from Species Distribution Models.

Usage

```
ecospat.occupied.patch (bin.map, Sp.occ.xy, buffer = 0)
```

Arguments

bin.map	Binary map (single layer or raster stack) from a Species Distribution Model.
Sp.occ.xy	xy-coordinates of the species presence.
buffer	numeric. Calculate occupied patch models from the binary map using a buffer (predicted area occupied by the species or within a buffer around the species, for details see ?extract).

Details

Predictive maps derived from SDMs inform about the potential distribution (or habitat suitability) of a species. Often it is useful to get information about the area of the potential distribution which is occupied by a species, e.g. for Red List assessments.

Value

a RasterLayer with value 1.

Author(s)

Frank Breiner <frank.breiner@wsl.ch>

References

IUCN Standards and Petitions Subcommittee. 2016. Guidelines for Using the IUCN Red List Categories and Criteria. Version 12. Prepared by the Standards and Petitions Subcommittee. Downloadable from http://www.iucnredlist.org/documents/RedListGuidelines.pdf

See Also

ecospat.rangesize, ecospat.mpa, ecospat.binary.model

Examples

```
library(raster)
library(dismo)

### make a maxent model

# path to maxent.jar file
path<- paste0(system.file(package="dismo"), "/java/maxent.jar")

if (file.exists(path) & require(rJava) & require(igraph)) {

  # get predictor variables
  fnames <- list.files(path=paste(system.file(package="dismo"), '/ex', sep=''),
                    pattern='grd', full.names=TRUE )
  predictors <- stack(fnames)
  #plot(predictors)

  # file with presence points
  occurence <- paste(system.file(package="dismo"), '/ex/bradypus.csv', sep='')
  occ <- read.table(occurence, header=TRUE, sep=',')[,-1]
  colnames(occ) <- c("x","y")
  occ <- ecospat.occ.desaggregation(occ,min.dist=1)

  # fit a domain model, biome is a categorical variable
  me <- maxent(predictors, occ, factors='biome')

  # predict to entire dataset
  pred <- predict(me, predictors)

  plot(pred)
  points(occ)
}

### to convert suitability to binary map

mpa.cutoff <- ecospat.mpa(pred,occ)

pred.bin.mpa <- ecospat.binary.model(pred,mpa.cutoff)
names(pred.bin.mpa) <- "me.mpa"
pred.bin.arbitrary <- ecospat.binary.model(pred,0.5)
```

```
names(pred.bin.arbitrary) <- "me.arbitrary"

### calculate occupied patch

mpa.ocp  <- ecospat.occupied.patch(pred.bin.mpa,occ)
arbitrary.ocp  <- ecospat.occupied.patch(pred.bin.arbitrary,occ)

par(mfrow=c(1,2))
plot(mpa.ocp) ## occupied patches: green area
points(occ,col="red",cex=0.5,pch=19)
plot(arbitrary.ocp)
points(occ,col="red",cex=0.5,pch=19)

## with buffer:
mpa.ocp  <- ecospat.occupied.patch(pred.bin.mpa,occ, buffer=500000)
arbitrary.ocp  <- ecospat.occupied.patch(pred.bin.arbitrary,occ, buffer=500000)

plot(mpa.ocp) ## occupied patches: green area
points(occ,col="red",cex=0.5,pch=19)
plot(arbitrary.ocp)
points(occ,col="red",cex=0.5,pch=19)
```

`ecospat.permut.glm` *GLM Permutation Function*

Description

A permutation function to get p-values on GLM coefficients and deviance.

Usage

```
ecospat.permut.glm (glm.obj, nperm, verbose = FALSE)
```

Arguments

`glm.obj`	Any calibrated GLM or GAM object with a binomial error distribution.
`nperm`	The number of permutations in the randomization process.
`verbose`	Boolean indicating whether to print progress output during calculation. Default is FALSE.

Details

Rows of the response variable are permuted and a new GLM is calibrated as well as deviance, adjusted deviance and coefficients are calculated. These random parameters are compared to the true parameters in order to derive p-values.

Value

Return p-values that are how the true parameters of the original model deviate from the disribution of the random parameters. A p-value of zero means that the true parameter is completely outside the random distribution.

Author(s)

Christophe Randin <christophe.randin@unibas.ch>, Antoine Guisan <antoine.guisan@unil.ch> and Trevor Hastie

References

Hastie, T., R. Tibshirani and J. Friedman. 2001. *Elements of Statistical Learning; Data Mining, Inference, and Prediction*, Springer-Verlag, New York.

Legendre, P. and L. Legendre. 1998. *Numerical Ecology*, 2nd English edition. Elsevier Science BV, Amsterdam.

Examples

```
if(require("rms",quietly=TRUE)){
  data('ecospat.testData')

  # data for Soldanella alpina
 data.Solalp<- ecospat.testData[c("Soldanella_alpina","ddeg","mind","srad","slp","topo")]

  # gbm model for Soldanella alpina

 glm.Solalp <- glm(Soldanella_alpina ~ pol(ddeg,2) + pol(mind,2) + pol(srad,2) + pol(slp,2)
      + pol(topo,2), data = data.Solalp, family = binomial)

  # p-values
  ecospat.permut.glm (glm.Solalp, 1000)
}
```

---

ecospat.plot.contrib *Plot Variables Contribution*

Description

Plot the contribution of the initial variables to the analysis (i.e. correlation circle). Typically these are the eigen vectors and eigen values in ordinations.

Usage

```
ecospat.plot.contrib (contrib, eigen)
```

Arguments

contrib	A dataframe of the contribution of each original variable on each axis of the analysis, i.e. the eigen vectors.
eigen	A vector of the importance of the axes in the ordination, i.e. a vector of eigen values.

Details

Requires ade4 library. If using princomp , use $loadings and $sdev of the princomp object. if using dudi.pca, use $li and $eig of the dudi.pca object.

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch>

References

Broennimann, O., M.C. Fitzpatrick, P.B. Pearman, B. Petitpierre, L. Pellissier, N.G. Yoccoz, W. Thuiller, M.J. Fortin, C. Randin, N.E. Zimmermann, C.H. Graham and A. Guisan. 2012. Measuring ecological niche overlap from occurrence and spatial environmental data. *Global Ecology and Biogeography*, **21**, 481-497.

See Also

ecospat.plot.niche.dyn,ecospat.plot.overlap.test, ecospat.niche.similarity.test,princomp

---

ecospat.plot.kappa *Plot Kappa*

Description

Plots the values for Cohen's Kappa along different thresholds.

Usage

```
ecospat.plot.kappa(Pred, Sp.occ)
```

Arguments

Pred	A vector of predicted probabilities
Sp.occ	A vector of binary observations of the species occurrence

Value

A plot of the Cohen's Kappa values along different thresholds.

Author(s)

Luigi Maiorano <luigi.maiorano@gmail.com> with contributions of Valeria Di Cola <valeria.dicola@unil.ch>.

References

Liu, C., P.M. Berry, T.P. Dawson, and R.G. Pearson. 2005. Selecting thresholds of occurrence in the prediction of species distributions. *Ecography*, **28**, 385-393.

Landis, J.R. and G.G. Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, **33**,159-174.

See Also

ecospat.meva.table, ecospat.max.tss, ecospat.plot.tss, ecospat.cohen.kappa, ecospat.max.kappa

Examples

```
Pred <- ecospat.testData$glm_Agrostis_capillaris
Sp.occ <- ecospat.testData$Agrostis_capillaris
ecospat.plot.kappa(Pred, Sp.occ)
```

---

ecospat.plot.mess *Plot MESS*

Description

Plot the MESS extrapolation index onto the geographical space.

Usage

```
ecospat.plot.mess (mess.object, cex=1, pch=15)
```

Arguments

mess.object	A dataframe as returned by the ecospat.mess function.
cex	Specify the size of the symbol.
pch	Specify the point symbols.

Value

Returns a plot of the the MESS extrapolation index onto the geographical space.

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com>

References

Elith, J., M. Kearney and S. Phillips. 2010. The art of modelling range-shifting species. *Methods in ecology and evolution*, **1**, 330-342.

See Also

ecospat.mess

Examples

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

#Plot MESS
ecospat.plot.mess (mess.object, cex=1, pch=15)
```

---

ecospat.plot.niche *Plot Niche*

Description

Plot a niche z created by ecospat.grid.clim.dyn.

Usage

```
ecospat.plot.niche (z, title, name.axis1, name.axis2, cor=FALSE)
```

Arguments

z	A gridclim object for the species distribution created by ecospat.grid.clim.dyn.
title	A title for the plot.
name.axis1	A label for the first axis.
name.axis2	A label for the second axis.
cor	Correct the occurrence densities of the species by the prevalence of the environments in its range (TRUE = yes, FALSE = no).

Details

if z is bivariate, a bivariate plot of the niche of the species. if z is univariate, a histogram of the niche of the species

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch>

References

Broennimann, O., M.C. Fitzpatrick, P.B. Pearman, B. Petitpierre, L. Pellissier, N.G. Yoccoz, W. Thuiller, M.J. Fortin, C. Randin, N.E. Zimmermann, C.H. Graham and A. Guisan. 2012. Measuring ecological niche overlap from occurrence and spatial environmental data. *Global Ecology and Biogeography*, **21**, 481-497.

See Also

ecospat.grid.clim.dyn

---

ecospat.plot.niche.dyn

Niche Categories and Species Density

Description

Plot niche categories and species density created by ecospat.grid.clim.dyn.

Usage

```
ecospat.plot.niche.dyn(z1, z2, quant = 0, title = "", name.axis1 = "Axis 1",
                    name.axis2 = "Axis 2", interest = 1, col.unf =
                    "green", col.exp = "red", col.stab = "blue", colZ1 =
                    "green3", colZ2 = "red3", transparency = 70)
```

Arguments

z1	A gridclim object for the native distribution.
z2	A gridclim object for the invaded range.
quant	The quantile of the environmental density used to delimit marginal climates.
title	The title of the plot.
name.axis1	A label for the first axis.
name.axis2	A label for the second axis
interest	Choose which density to plot: if interest=1, plot native density, if interest=2, plot invasive density.
col.unf	The color used to depict unfilling area.
col.exp	The color used to depict expansion area.
col.stab	The color used to depict stability area.
colZ1	The color used to delimit the native extent.
colZ2	The color used to delimit the invaded extent.
transparency	A value between 0 and 100 to set the transparency level of the niche categories

Details

Using the default colors, the plot will show the niche stability in blue, niche expansion in red, and niche unfilling in green. The solid contour line indicates the extent of environmental conditions that exists in the native and invaded ranges; the dotted contour line indicates the quantile indicated by the `quant` argument. The densities of occurrences are displayed using gray shading. This shading shows occurences in the native or invaded range only, as determined by the value of the `interest` argument.

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com>

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

Description

Plot a histogram of observed and randomly simulated overlaps, with p-values of equivalency or similarity tests.

Usage

```
ecospat.plot.overlap.test (x, type, title)
```

Arguments

`x`	Object created by `ecospat.niche.similarity.test` or `ecospat.niche.equivalency.test`.
`type`	Select the tested index. Must be "D", "I", "expansion", "stability", "unfilling".
`title`	The title for the plot.

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch>

References

Broennimann, O., M.C. Fitzpatrick, P.B. Pearman, B. Petitpierre, L. Pellissier, N.G. Yoccoz, W. Thuiller, M.J. Fortin, C. Randin, N.E. Zimmermann, C.H. Graham and A. Guisan. 2012. Measuring ecological niche overlap from occurrence and spatial environmental data. *Global Ecology and Biogeography*, **21**, 481-497.

See Also

`ecospat.niche.similarity.test`, `ecospat.niche.equivalency.test`

`ecospat.plot.tss` *Plot True skill statistic (TSS)*

Description

Plots the values for True skill statistic (TSS) along different thresholds.

Usage

```
ecospat.plot.tss(Pred, Sp.occ)
```

Arguments

`Pred`	A vector of predicted probabilities
`Sp.occ`	A vector of binary observations of the species occurrence

Value

A plot of the TSS values along different thresholds.

Author(s)

Luigi Maiorano <luigi.maiorano@gmail.com>

References

Liu, C., P.M. Berry, T.P. Dawson, and R.G. Pearson. 2005. Selecting thresholds of occurrence in the prediction of species distributions. *Ecography*, **28**, 385-393.

Liu, C., M. White and G. Newell. 2013. Selecting thresholds for the prediction of species occurrence with presence-only data. *Journal of Biogeography*, *40*, 778-789.

See Also

`ecospat.meva.table`, `ecospat.max.tss`, `ecospat.plot.kappa`, `ecospat.cohen.kappa`, `ecospat.max.kappa`

Examples

```
Pred <- ecospat.testData$glm_Agrostis_capillaris
Sp.occ <- ecospat.testData$Agrostis_capillaris
ecospat.plot.tss(Pred, Sp.occ)
```

ecospat.poolingEvaluation *Evaluation of species distribution models using the pooling procedure*

Description

This function evaluates species distribution models using 100% of the dataset by pooling the different runs of the cross validation as in Collart et al. 2021

Usage

```
ecospat.poolingEvaluation(fit,
                          calib,
                          resp,
                          AlgoName = NULL,
                    metrics = c("SomersD","AUC","MaxTSS","MaxKappa","Boyce"),
                          ensembleEvaluation=FALSE,
                          w=NULL,
                          metricToEnsemble = "MaxTSS")
```

Arguments

fit	a list containing *n* data.frame or matrix, where *n* corresponds to the number of algorithm you want to evaluate. The data.frames (matrices) need to contain the model predictions (ranging between 0 and 1) resulting from the different runs of cross-validation. These data.frame need to have the same number of rows as in the full dataset (100% of the occurrences and 100% of the absences or background points) and a number of column equal to the number of cross-validation runs
calib	a logical matrix with a number of rows equal to the full dataset and a number of column corresponding to the number of cross-validation runs. The value TRUE is to mention the elements that where used to calibrate the models whereas FALSE corresponds to the one that will be used for the evaluation (*NB* the points used to calibrate the models during a cross-validation run should be the same across algoritms)
resp	a numeric vector where 1 corresponds to a species response and 0 to an absence (or background point) with a length corresponding to number of rows in calib
AlgoName	a character vector for giving a name to each elements of fit. If NULL, the position in the list will be used instead.
metrics	a vector of evaluation metrics chosen among "SomersD", "AUC", "MaxTSS", "MaxKappa", "Boyce"
ensembleEvaluation	logical. If TRUE, the ensemble model will be evaluated applying a weighted mean across algorithms.

`w` a numeric vector of the weights used to realize the ensemble model. The length should match the number of algorithms.

`metricToEnsemble` character. Metric to use to ensemble the models with a weighted mean when w is not given. The metric should be one in metrics

Details

Because a minimum sample size is needed to evaluate models (see Jiménez-Valverde, 2020), this function uses the approach from Collart et al.(2021), which consists to pool the suitability values of the hold-out data (evaluation dataset) across replicates. As the same data point (presence or absence or background point) is presumably sampled in several replicates, the suitability values for each data point is consequently averaged across replicates where they were sampled. This procedure generates a series of independent suitability values with a size approximately equal (as some data points may not have been sampled by chance in any of the *n* replicates) to that of the number of data point.

Value

a list containing:

`evaluations` a matrix with the evaluation scores based on the different modelling algorithms and based on the consensus across the modelling algorithms (called here "ensemble")

`fit` a matrix of predicted values resulting from the pooling procedure and used to compute the evaluation scores. The column *resp* is where the species occurs or not

Author(s)

Flavien Collart <flavien.collart@unil.ch>

with contributions of Olivier Broennimann <olivier.broennimann@unil.ch>

References

Collart, F., Hedenäs, L., Broennimann, O., Guisan, A. and Vanderpoorten, A. 2021. Intraspecific differentiation: Implications for niche and distribution modelling. *Journal of Biogeography*. **48**, 415-426. doi:10.1111/jbi.14009

Jiménez-Valverde, A. 2020. Sample size for the evaluation of presence-absence models. *Ecological Indicators*. **114**, 106289. doi:10.1016/j.ecolind.2020.106289

See Also

`ecospat.ESM.EnsembleEvaluation`

Examples

```
 set.seed(42)
 resp <- c(rep(1,15),rep(0,85)) #15 presences and 85 absences
 #Generating a fake fit object whith two algorithms and 3 cross-vaidation
 fit <- list(matrix(0,nc=3,nr=100),
             matrix(0,nc=3,nr=100))
fit[[1]][1:15,] = sample(seq(0,1, by=0.01),15*3,prob=c(rep(1,51),rep(10,50)),replace=TRUE)
fit[[2]][1:15,] = sample(seq(0,1, by=0.01),15*3,prob=c(rep(1,51),rep(10,50)),replace=TRUE)
fit[[1]][16:100,] = sample(seq(0,1, by=0.01),85*3,prob=c(rep(10,51),rep(1,50)),replace=TRUE)
fit[[2]][16:100,] = sample(seq(0,1, by=0.01),85*3,prob=c(rep(10,51),rep(1,50)),replace=TRUE)

 # Generating a calib object where 80% of the dataset is used to calibrate the model
 # and 20% to evaluate it
 calib <- matrix(TRUE,nc=3,nr=100)
 calib[c(sample(1:15,3),sample(16:100,17)),1]=FALSE
 calib[c(sample(1:15,3),sample(16:100,17)),2]=FALSE
 calib[c(sample(1:15,3),sample(16:100,17)),3]=FALSE

 # Evaluation via the pooling procedure
eval <- ecospat.poolingEvaluation(fit=fit,calib=calib,resp=resp,metrics=c("AUC","MaxTSS"))
 eval$evaluations

 # Evaluation including the ensemble model based on a weighted mean using MaxTSS
evalEns <- ecospat.poolingEvaluation(fit=fit,calib=calib,resp=resp,ensembleEvaluation=TRUE,
                                     metrics=c("AUC","MaxTSS"))
 evalEns$evaluations

 # Evaluation including the ensemble model based on a mean by giving the same weight for
 # each algorithm
evalEns <- ecospat.poolingEvaluation(fit=fit,calib=calib,resp=resp,ensembleEvaluation=TRUE,
                                     metrics=c("AUC","MaxTSS"),w=c(1,1))
 evalEns$evaluations
```

---

ecospat.rand.pseudoabsences

Sample Pseudo-Absences

Description

Randomly sample pseudoabsences from an environmental dataframe covering the study area.

Usage

```
ecospat.rand.pseudoabsences (nbabsences, glob, colxyglob, colvar="all",
presence, colxypresence, mindist)
```

Arguments

nbabsences	The number of pseudoabsences desired.
glob	A two-column dataframe (or a vector) of the environmental values (in column) for background pixels of the whole study area (in row).
colxyglob	The range of columns for x and y in glob.
colvar	The range of columns for the environmental variables in glob. colvar="all" keeps all the variables in glob in the final dataframe. colvar=NULL keeps only x and y.
presence	A presence-absence dataframe for each species (columns) in each location or grid cell (rows).
colxypresence	The range of columns for x and y in presence.
mindist	The minimum distance from presences within wich pseudoabsences should not be drawn (buffer distance around presences).

Value

A dataframe of random absences.

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch>

Examples

```
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)
```

---

ecospat.rangesize *Quantification of the range size of a species using habitat suitability maps and IUCN criteria)*

Description

This function quantifies the range size of a species using standard IUCN criteria (Area of Occupancy AOO, Extent of Occurence EOO) or binary maps derived from Species Distribution Models.

Usage

```
ecospat.rangesize (bin.map, ocp, buffer, eoo.around.model, eoo.around.modelocp,
xy, EOO, Model.within.eoo, AOO, resol, AOO.circles, d, lonlat, return.obj,
save.obj, save.rangesize, directory)

ecospat.rangesize (bin.map = NULL,
                   ocp = TRUE,
                   buffer = 0,
                   eoo.around.model = TRUE,
                   eoo.around.modelocp = FALSE,
                   xy = NULL,
                   EOO = TRUE,
                   Model.within.eoo = TRUE,
                   AOO = TRUE,
                   resol = c(2000, 2000),
                   AOO.circles = FALSE,
                   d = sqrt((2000 * 2)/pi),
                   lonlat = FALSE,
                   return.obj = TRUE,
                   save.obj = FALSE,
                   save.rangesize = FALSE,
                   directory = getwd())
```

Arguments

bin.map	Binary map (single layer or raster stack) from a Species Distribution Model.
ocp	logical. Calculate occupied patch models from the binary map (predicted area occupied by the species)
buffer	numeric. Calculate occupied patch models from the binary map using a buffer (predicted area occupied by the species or within a buffer around the species, for details see ?extract).
eoo.around.model	logical. The EOO around all positive predicted values from the binary map.
eoo.around.modelocp	logical. EOO around all positive predicted values of occupied patches.
xy	xy-coordinates of the species presence
EOO	logical. Extent of Occurrence. Convex Polygon around occurrences.
Model.within.eoo	logical. Area predicted as suitable by the model within EOO.
AOO	logical. Area of Occupancy ddervied by the occurrences.
resol	Resolution of the grid frame at which AOO should be calculated.
AOO.circles	logical. AOO calculated by circles around the occurrences instead of using a grid frame.

d	Radius of the AOO.circles around the occurrences.
lonlat	Are these longitude/latidue coordinates? (Default = FALSE).
return.obj	logical. should the objects created to estimate range size be returned (rasterfiles and spatial polygons). Default: TRUE
save.obj	logical. should objects be saved on hard drive?
save.rangesize	logical. should range size estimations be saved on hard drive .
directory	directory in which objects should be saved (Default = getwd())

Details

The range size of a species is important for many conservation purposes, e.g. to assess the status of threat for IUCN Red Lists. This function quantifies the range size using different IUCN measures, i.e. the Area Of Occupancy (AOO), the Extent Of Occurrence (EOO) and from binary maps derived from Species Distribution Models (SDMs). Different ways to extract range size from SDMs are available, e.g. using occupied patches, the suitable habitat within EOO or a convex hull around the suitable habitat.

Value

A list with the values of range size quantification and the stored objects used for quantification (of class RasterLayers, ahull, ConvexHull).

Author(s)

Frank Breiner <frank.breiner@wsl.ch>

References

IUCN. 2012. IUCN Red List Categories and Criteria: Version 3.1. Second edition. Gland, Switzerland and Cambridge, UK: IUCN. iv + 32pp.

IUCN Standards and Petitions Subcommittee. 2016. Guidelines for Using the IUCN Red List Categories and Criteria. Version 12. Prepared by the Standards and Petitions Subcommittee. Downloadable from http://www.iucnredlist.org/documents/RedListGuidelines.pdf

Pateiro-Lopez, B., and A. Rodriguez-Casal. 2010. Generalizing the Convex Hull of a Sample: The R Package alphahull. *Journal of Statistical software*, **34**, 1-28.

See Also

ecospat.occupied.patch, ecospat.mpa, ecospat.binary.model

Examples

```
library(raster)
library(dismo)

### make a maxent model
```

```
# path to maxent.jar file
path<- paste0(system.file(package="dismo"), "/java/maxent.jar")

if (file.exists(path) & require(rJava)) {

# get predictor variables
fnames <- list.files(path=paste(system.file(package="dismo"), '/ex', sep=''),
                     pattern='grd', full.names=TRUE )
predictors <- stack(fnames)
#plot(predictors)

# file with presence points
occurence <- paste(system.file(package="dismo"), '/ex/bradypus.csv', sep='')
occ <- read.table(occurence, header=TRUE, sep=',')[,-1]
colnames(occ) <- c("x","y")
occ <- ecospat.occ.desaggregation(occ,min.dist=1)

# fit a domain model, biome is a categorical variable
me <- maxent(predictors, occ, factors='biome')

# predict to entire dataset
pred <- predict(me, predictors)

plot(pred)
points(occ)

### make binary maps

# use MPA to convert suitability to binary map
mpa.cutoff <- ecospat.mpa(pred,occ)

# use Boyce index to convert suitability to binary map
boyce <- ecospat.boyce(pred,  occ)
### use the boyce index to find a threshold
pred.bin.arbitrary <- ecospat.binary.model(pred,0.5)

pred.bin.mpa <- ecospat.binary.model(pred,mpa.cutoff)
names(pred.bin.mpa) <- "me.mpa"
pred.bin.arbitrary <- ecospat.binary.model(pred,0.5)
names(pred.bin.arbitrary) <- "me.arbitrary"

### rangesize calculations

if(require(rgeos,alphahull,quietly=TRUE)){

  rangesize <- ecospat.rangesize(stack(pred.bin.mpa,pred.bin.arbitrary),
                                 xy=occ,
                                 resol=c(1,1),
                                 eoo.around.modelocp =TRUE,
                                 AOO.circles = TRUE,
                                 d=200000,
                                 lonlat =TRUE)
  rangesize$RangeSize
```

```
names(rangesize$RangeObjects)

par(mfrow=c(1,3))

plot(ecospat.binary.model(pred,0),legend=FALSE, main="IUCN criteria")

### IUCN criteria & derivates

# plot AOO
plot(rangesize$RangeObjects$AOO,add=TRUE, col="red",legend=FALSE)

# plot EOO
plot(rangesize$RangeObjects$EOO@polygons,add=TRUE, border="red", lwd=2)

# plot circles around occurrences
plot(rangesize$RangeObjects$AOO.circle@polygons,add=TRUE,border="blue")

for(i in 1:2){
  ## plot the occupied patches of the model
  plot(rangesize$RangeObjects$models.ocp[[i]],col=c("grey","blue","darkgreen"),
  main=names(rangesize$RangeObjects$models.ocp[[i]]),legend=FALSE)
  points(occ,col="red",cex=0.5,pch=19)
  ## plot EOO around model
 plot(rangesize$RangeObjects$eoo.around.model[[i]]@polygons,add=TRUE,border="blue",lwd=2)
  ## plot EOO around occupied patches
 plot(rangesize$RangeObjects$eoo.around.mo.ocp[[i]]@polygons,add=TRUE,border="darkgreen",
  lwd=2)
  ## plot the modeled area within EOO
  #plot(rangesize$RangeObjects$model.within.eoo[[i]],col=c("grey","blue","darkgreen"))
  #points(occ,col="red",cex=0.5,pch=19)
  #plot(rangesize$RangeObjects$EOO@polygons,add=TRUE, border="red", lwd=2)
}
par(mfrow=c(1,1))

### Alpha-hulls are not included in the function yet because of Licence limitations.
### However, alpha-hulls can easily be included manually (see also the help file of
### the alpha hull package):

alpha = 2 # alpha value of 2 recommended by IUCN

del<-alphahull::delvor(occ)
dv<-del$mesh
mn <- mean(sqrt(abs(del$mesh[,3]-del$mesh[,5])^2+abs(del$mesh[,4]-del$mesh[,6])^2))*alpha
alpha.hull<-alphahull::ahull(del,alpha=mn)

#Size of alpha-hulls
#areaahull(alpha.hull) #works but uses a deprecated function in alphahull 2.1

#plot alphahulls
plot(rangesize$RangeObjects$models.ocp[[i]],col=c("grey","blue","darkgreen"),
main=names(rangesize$RangeObjects$models.ocp[[i]]),legend=FALSE)
plot(alpha.hull,add=TRUE,lwd=1)
```

```
}
}
```

---

ecospat.rcls.grd *Reclassifying grids function*

Description

Function for reclassifying grid files to get a combined statification from more than one grid

Usage

```
ecospat.rcls.grd(in_grid,no.classes)
```

Arguments

in_grid	The grid to be reclassified.
no.classes	The number of desired new classes.

Details

This function reclassifies the input grid into a number of new classes that the user defines. The boundaries of each class are decided automatically by splitting the range of values of the input grid into the user defined number of classes.

Value

Returns a reclassified Raster object

Author(s)

Achilleas Psomas <achilleas.psomas@wsl.ch> and Niklaus E. Zimmermann <niklaus.zimmermann@wsl.ch>

Examples

```
library(raster)
library(classInt)

bio3<- raster(system.file("external/bioclim/current/bio3.grd",package="biomod2"))
bio12<- raster(system.file("external/bioclim/current/bio12.grd",package="biomod2"))

B3.rcl<-ecospat.rcls.grd(bio3,9)
B12.rcl<-ecospat.rcls.grd(bio12,9)
B3B12.comb <- B12.rcl+B3.rcl*10

# Plotting a histogram of the classes
hist(B3B12.comb,breaks=100,col=heat.colors(88))
```

```
# Plotting the new RasterLayer (9x9 classes)
plot(B3B12.comb,col=rev(rainbow(88)),main="Stratified map")
```

---

ecospat.recstrat_prop *Random Ecologically Stratified Sampling of propotional numbers*

Description

This function randomly collects a user-defined total number of samples from the stratification layer.

Usage

```
ecospat.recstrat_prop(in_grid, sample_no)
```

Arguments

in_grid	The stratification grid to be sampled.
sample_no	The total number of pixels to be sampled.

Details

The number of samples per class are determined proportional to the abundance of each class. The number of classes in the stratification layer are determined automatically from the integer input map. If the proportion of samples for a certain class is below one then no samples are collected for this class.

Value

Returns a dataframe with the selected sampling locations their coordinates and the strata they belong in.

Author(s)

Achilleas Psomas <achilleas.psomas@wsl.ch> and Niklaus E. Zimmermann <niklaus.zimmermann@wsl.ch>

See Also

ecospat.recstrat_regl ecospat.rcls.grd

Examples

```
library(raster)
library(classInt)

bio3<- raster(system.file("external/bioclim/current/bio3.grd",package="biomod2"))
bio12<- raster(system.file("external/bioclim/current/bio12.grd",package="biomod2"))

B3.rcl<-ecospat.rcls.grd(bio3,9)
```

```
B12.rcl<-ecospat.rcls.grd(bio12,9)
B3B12.comb <- B12.rcl+B3.rcl*10

B3B12.prop_samples <- ecospat.recstrat_prop(B3B12.comb,100)

plot(B3B12.comb)
points(B3B12.prop_samples$x,B3B12.prop_samples$y,pch=16,cex=0.6,col=B3B12.prop_samples$class)
```

`ecospat.recstrat_regl` *Random Ecologically Stratified Sampling of equal numbers*

Description

This function randomly takes an equal number of samples per class in the stratification layer.

Usage

```
ecospat.recstrat_regl(in_grid, sample_no)
```

Arguments

`in_grid`	The stratification grid to be sampled.
`sample_no`	The total number of pixels to be sampled.

Details

The number of classes in the stratification layer is determined automatically from the integer input map. If the number of pixels in a class is higher than the number of samples, then a random selection without re-substitution is performed, otherwise all pixels of that class are selected.

Value

Returns a dataframe with the selected sampling locations their coordinates and the strata they belong in.

Author(s)

Achilleas Psomas <achilleas.psomas@wsl.ch> and Niklaus E. Zimmermann <niklaus.zimmermann@wsl.ch>

See Also

`ecospat.recstrat_prop` `ecospat.rcls.grd`

Examples

```
library(raster)
library(classInt)

bio3<- raster(system.file("external/bioclim/current/bio3.grd",package="biomod2"))
bio12<- raster(system.file("external/bioclim/current/bio12.grd",package="biomod2"))

B3.rcl<-ecospat.rcls.grd(bio3,9)
B12.rcl<-ecospat.rcls.grd(bio12,9)
B3B12.comb <- B12.rcl+B3.rcl*10

B3B12.regl_samples <- ecospat.recstrat_prop(B3B12.comb,100)

plot(B3B12.comb)
points(B3B12.regl_samples$x,B3B12.regl_samples$y,pch=16,cex=0.6,col=B3B12.regl_samples$class)
```

`ecospat.sample.envar` *Sample Environmental Variables*

Description

Add environmental values to a species dataframe.

Usage

```
ecospat.sample.envar (dfsp, colspxy, colspkept = "xy", dfvar,
colvarxy, colvar = "all", resolution)
```

Arguments

`dfsp`	A species dataframe with x (long), y (lat) and optional other variables.
`colspxy`	The range of columns for x (long) and y (lat) in dfsp.
`colspkept`	The columns of dfsp that should be kept in the final dataframe (default: xy).
`dfvar`	A dataframe object with x, y and environmental variables.
`colvarxy`	The range of columns for x and y in dfvar.
`colvar`	The range of enviromental variable columns in dfvar (default: all except xy).
`resolution`	The distance between x,y of species and environmental datafreme beyond which values shouldn't be added.

Details

The xy (lat/long) coordinates of the species occurrences are compared to those of the environment dataframe and the value of the closest pixel is added to the species dataframe. When the closest environment pixel is more distant than the given resolution, NA is added instead of the value. This function is similar to sample() in ArcGIS.

Value

A Dataframe with the same rows as dfsp, with environmental values from dfvar in column.

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch>

Examples

```
data("ecospat.testNiche")
spp <- ecospat.testNiche
sp1 <- spp[1:32,2:3]
names(sp1)<-c("x","y")
occ.sp1 <- ecospat.occ.desaggregation(xy=sp1,min.dist=500)
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))
```

---

ecospat.SESAM.prr *SESAM Probability Ranking Rule*

Description

Implement the SESAM framework to predict community composition using a 'probability ranking' rule.

Usage

```
ecospat.SESAM.prr(proba, sr, verbose = FALSE)
```

Arguments

proba	A data frame object of SDMs probabilities (or other sources) for all species. Column names (species names SDM) and row name (sampling sites) (need to have defined row names).
sr	A data frame object with species richness value in the first column. Sites should be arranged in the same order as in the 'prob' argument.
verbose	Boolean indicating whether to print progress output during calculation. Default is FALSE.

Details

The SESAM framework implemented in ecospat is based on 1) probabilities of individual species presence for each site - these can be obtained for example by fitting SDMs. This step represents the application of an environmental filter to the community assembly, 2) richness predictions for each site - the richness prediction can be derived in different ways, for instance by summing probabilities from individual species presence for each site or by fitting direct richness models. This step represents the application of a macroecological constraint to the number of species that can coexist in the considered unit, 3) a biotic rule to decide which species potentially present in the site are retained in the final prediction to match the richness value predicted. The biotic rule applied here is called 'probability ranking' rule: the community composition in each site is determined by ranking the species in decreasing order of their predicted probability of presence from SDMs up to a richness prediction.

Value

Returns a '.txt' file saved in the working directory that contains the community prediction by the SESAM framework, i.e. binary predictions for all species (columns) for each site (rows).

Author(s)

Manuela D'Amen <manuela.damen@unil.ch> and Anne Dubuis <anne.dubuis@gmail.com>

References

D'Amen, M., A. Dubuis, R.F. Fernandes, J. Pottier, L. Pellissier and A. Guisan. 2015. Using species richness and functional traits predictions to constrain assemblage predictions from stacked species distribution models. *J. Biogeogr.*, **42**, 1255-1266.

Guisan, A. and C. Rahbek. 2011. SESAM - a new framework integrating macroecological and species distribution models for predicting spatio-temporal patterns of species assemblages. *J. Biogeogr.*, **38**, 1433-1444.

Examples

```
proba <- ecospat.testData[,73:92]
sr <- as.data.frame(rowSums(proba))
ppr<-ecospat.SESAM.prr(proba, sr)
head(ppr)
```

---

ecospat.shift.centroids

Draw Centroid Arrows

Description

Draw arrows linking the centroid of the native and exotic (non-native) distribution (continuous line) and between native and invaded extent (dashed line).

Usage

```
ecospat.shift.centroids(sp1, sp2, clim1, clim2,col)
```

Arguments

sp1	The scores of the species native distribution along the the two first axes of the PCA.
sp2	The scores of the species invasive distribution along the the two first axes of the PCA.
clim1	The scores of the entire native extent along the the two first axes of the PCA.
clim2	The scores of the entire invaded extent along the the two first axes of the PCA.
col	Colour of the arrow.

Details

Allows to visualize the shift of the niche centroids of the species and the centroids of the climatic conditions in the study area. To compare invasive species niche, the arrow links the centroid of the native and inasive distribution (continuous line) and between native and invaded extent (dashed line).

Value

Arrow on the overlap test plot

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com>

---

ecospat.testData *Test Data For The Ecospat package*

Description

Data frame that contains vegetation plots data: presence records of 50 species, a set of environmental variables (topo-climatic) and SDM predictions for some species in the Western Swiss Alps (Canton de Vaud, Switzerland).

Usage

```
data("ecospat.testData")
```

Format

A data frame with 300 observations on the following 96 variables.

`numplots` Number of the vegetation plot.

`long` Longitude, in Swiss plane coordinate system of the vegetation plot.

`lat` Latitude, in Swiss plane coordinate system of the vegetation plot.

`ddeg` Growing degree days (with a 0 degrees Celsius threshold).

`mind` Moisture index over the growing season (average values for June to August in mm day-1).

`srad` The annual sum of radiation (in kJ m-2 year-1).

`slp` Slope (in degrees) calculated from the DEM25.

`topo` Topographic position (an integrated and unitless measure of topographic exposure.

`Achillea_atrata`

`Achillea_millefolium`

`Acinos_alpinus`

`Adenostyles_glabra`

`Aposeris_foetida`

`Arnica_montana`

`Aster_bellidiastrum`

`Bartsia_alpina`

`Bellis_perennis`

`Campanula_rotundifolia`

`Centaurea_montana`

`Cerastium_latifolium`

`Cruciata_laevipes`

`Doronicum_grandiflorum`

`Galium_album`

`Galium_anisophyllon`

`Galium_megalospermum`

`Gentiana_bavarica`

`Gentiana_lutea`

`Gentiana_purpurea`

`Gentiana_verna`

`Globularia_cordifolia`

`Globularia_nudicaulis`

`Gypsophila_repens`

`Hieracium_lactucella`

`Homogyne_alpina`

`Hypochaeris_radicata`

Leontodon_autumnalis
Leontodon_helveticus
Myosotis_alpestris
Myosotis_arvensis
Phyteuma_orbiculare
Phyteuma_spicatum
Plantago_alpina
Plantago_lanceolata
Polygonum_bistorta
Polygonum_viviparum
Prunella_grandiflora
Rhinanthus_alectorolophus
Rumex_acetosa
Rumex_crispus
Vaccinium_gaultherioides
Veronica_alpina
Veronica_aphylla
Agrostis_capillaris
Bromus_erectus_sstr
Campanula_scheuchzeri
Carex_sempervirens
Cynosurus_cristatus
Dactylis_glomerata
Daucus_carota
Festuca_pratensis_sl
Geranium_sylvaticum
Leontodon_hispidus_sl
Potentilla_erecta
Pritzelago_alpina_sstr
Prunella_vulgaris
Ranunculus_acris_sl
Saxifraga_oppositifolia
Soldanella_alpina
Taraxacum_officinale_aggr
Trifolium_repens_sstr
Veronica_chamaedrys
Parnassia_palustris

`glm_Agrostis_capillaris` GLM model for the species Agrostis_capillaris.

`glm_Leontodon_hispidus_sl` GLM model for the species Leontodon_hispidus_sl.

`glm_Dactylis_glomerata` GLM model for the species Dactylis_glomerata.

`glm_Trifolium_repens_sstr` GLM model for the species Trifolium_repens_sstr.

`glm_Geranium_sylvaticum` GLM model for the species Geranium_sylvaticum.

`glm_Ranunculus_acris_sl` GLM model for the species Ranunculus_acris_sl.

`glm_Prunella_vulgaris` GLM model for the species Prunella_vulgaris.

`glm_Veronica_chamaedrys` GLM model for the species Veronica_chamaedrys.

`glm_Taraxacum_officinale_aggr` GLM model for the species Taraxacum_officinale_aggr.

`glm_Plantago_lanceolata` GLM model for the species Plantago_lanceolata.

`glm_Potentilla_erecta` GLM model for the species Potentilla_erecta.

`glm_Carex_sempervirens` GLM model for the species Carex_sempervirens.

`glm_Soldanella_alpina` GLM model for the species Soldanella_alpina.

`glm_Cynosurus_cristatus` GLM model for the species Cynosurus_cristatus.

`glm_Campanula_scheuchzeri` GLM model for the species Campanula_scheuchzeri.

`glm_Festuca_pratensis_sl` GLM model for the species Festuca_pratensis_sl.

`gbm_Bromus_erectus_sstr` GBM model for the species Bromus_erectus_sstr.

`glm_Saxifraga_oppositifolia` GLM model for the species Saxifraga_oppositifolia.

`glm_Daucus_carota` GLM model for the species Daucus_carota.

`glm_Pritzelago_alpina_sstr` GLM model for the species Pritzelago_alpina_sstr.

`glm_Bromus_erectus_sstr` GLM model for the species Bromus_erectus_sstr.

`gbm_Saxifraga_oppositifolia` GBM model for the species Saxifraga_oppositifolia.

`gbm_Daucus_carota` GBM model for the species Daucus_carota.

`gbm_Pritzelago_alpina_sstr` GBM model for the species Pritzelago_alpina_sstr.

Details

The study area is the Western Swiss Alps of Canton de Vaud, Switzerland.

Five topo-climatic explanatory variables to calibrate the SDMs: growing degree days (with a 0 degrees Celsius threshold); moisture index over the growing season (average values for June to August in mm day-1); slope (in degrees); topographic position (an integrated and unitless measure of topographic exposure; Zimmermann et al., 2007); and the annual sum of radiation (in kJ m-2 year-1). The spatial resolution of the predictor is 25 m x 25 m so that the models could capture most of the small-scale variations of the climatic factors in the mountainous areas.

Two modelling techniques were used to produce the SDMs: generalized linear models (GLM; McCullagh & Nelder, 1989; R library 'glm') and generalized boosted models (GBM; Friedman, 2001; R library 'gbm'). The SDMs correpond to 20 species: Agrostis_capillaris, Leontodon_hispidus_sl, Dactylis_glomerata, Trifolium_repens_sstr, Geranium_sylvaticum, Ranunculus_acris_sl, Prunella_vulgaris, Veronica_chamaedrys, Taraxacum_officinale_aggr, Plantago_lanceolata, Potentilla_erecta, Carex_sempervirens, Soldanella_alpina, Cynosurus_cristatus, Campanula_scheuchzeri, Festuca_pratensis_sl, Daucus_carota, Pritzelago_alpina_sstr, Bromus_erectus_sstr and Saxifraga_oppositifolia.

Author(s)

Antoine Guisan <antoine.guisan@unil.ch>, Anne Dubuis <anne.dubuis@gmail.com> and Valeria Di Cola <valeria.dicola@unil.ch>

References

Guisan, A. 1997. Distribution de taxons vegetaux dans un environnement alpin: Application de modelisations statistiques dans un systeme d'information geographique. PhD Thesis, University of Geneva, Switzerland.

Guisan, A., J.P. Theurillat. and F. Kienast. 1998. Predicting the potential distribution of plant species in an alpine environment. *Journal of Vegetation Science*, **9**, 65-74.

Guisan, A. and J.P. Theurillat. 2000. Assessing alpine plant vulnerability to climate change: A modeling perspective. *Integrated Assessment*, **1**, 307-320.

Guisan, A. and J.P. Theurillat. 2000. Equilibrium modeling of alpine plant distribution and climate change : How far can we go? *Phytocoenologia*, **30**(3-4), 353-384.

Dubuis A., J. Pottier, V. Rion, L. Pellissier, J.P. Theurillat and A. Guisan. 2011. Predicting spatial patterns of plant species richness: A comparison of direct macroecological and species stacking approaches. *Diversity and Distributions*, **17**, 1122-1131.

Zimmermann, N.E., T.C. Edwards, G.G Moisen, T.S. Frescino and J.A. Blackard. 2007. Remote sensing-based predictors improve distribution models of rare, early successional and broadleaf tree species in Utah. *Journal of Applied Ecology* **44**, 1057-1067.

Examples

```
data(ecospat.testData)
str(ecospat.testData)
dim(ecospat.testData)
names(ecospat.testData)
```

---

ecospat.testEnvRaster *Test Environmental Rasters for The Ecospat package*

Description

A stack of 5 topoclimatic rasters at 250m resolution for the Western Swiss Alps. It includes "ddeg0" (growing degree-days above 0C), "mind68" (moisture index for month June to August), "srad68" (solar radiation for month June to August), "slope25" (average of slopes at 25m resolution) and "topos25" (average of topographic positions at 25m resolution)

Format

ecospat.testEnvRaster is a RasterBrick encapsulated in a .Rdata that contains the following rasters:

[1] "ddeg0" [2] "mind68" [3] "srad68" [4] "slope25" [5] "topos25"

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch>

References

Zimmermann, N.E., F. Kienast. 2009. Predictive mapping of alpine grasslands in Switzerland: Species versus community approach. *Journal of Vegetation Science*, **10**, 469-482.

Examples

```
## Not run:
fpath <- system.file("extdata", "ecospat.testEnvRaster.RData", package="ecospat")
load(fpath)
plot(env)

## End(Not run)
```

---

ecospat.testMdr *Test Data For The ecospat.mdr function*

Description

Data frame that contains presence records the species Centaurea stoebe along years in North America.

Usage

```
data("ecospat.testMdr")
```

Format

A data frame with 102 observations of Centaurea stoebe.

`latitude` Latitude, in WGS coordinate system.

`longitude` Longitude, in WGS coordinate system.

`date` Year of the presence record.

Details

Simplified dataset to exemplify the use of the ecospat.mdr function to calculate minimum dispersal routes.

Author(s)

Olivier Broennimann <olivier.broennimann@unil.ch>

References

Broennimann, O., P. Mraz, B. Petitpierre, A. Guisan, and H. Muller-Scharer. 2014. Contrasting spatio-temporal climatic niche dynamics during the eastern and western invasions of spotted knapweed in North America.*Journal of biogeography*, **41**, 1126-1136.

Hordijk, W. and O. Broennimann. 2012. Dispersal routes reconstruction and the minimum cost arborescence problem. *Journal of theoretical biology*, **308**, 115-122.

Examples

```
data(ecospat.testMdr)
str(ecospat.testMdr)
dim(ecospat.testMdr)
```

---

ecospat.testNiche *Test Data For The Niche Overlap Analysis*

Description

Data frame that contains occurrence sites for each species, long, lat and the name of the species at each site.

Usage

```
data(ecospat.testNiche)
```

Format

ecospat.testNiche is a data frame with the following columns:

`species` sp1, sp2, sp3 and sp4.

`long` Longitude, in Swiss plane coordinate system of the vegetation plot.

`lat` Latitude, in Swiss plane coordinate system of the vegetation plot.

`Spp` Scientific name of the species used in the exmaple: Bromus_erectus_sstr, Saxifraga_oppositifolia, Daucus_carota and Pritzelago_alpina_sstr.

Details

List of occurence sites for the species.

Author(s)

Antoine Guisan <antoine.guisan@unil.ch>, Anne Dubuis <anne.dubuis@gmail.com> and Valeria Di Cola <valeria.dicola@unil.ch>

See Also

ecospat.testData

Examples

```
data(ecospat.testNiche)
dim(ecospat.testNiche)
names(ecospat.testNiche)
```

---

ecospat.testNiche.inv *Test Data For The Niche Dynamics Analysis In The Invaded Range Of A Hypothetical Species*

Description

Data frame that contains geographical coordinates, environmental variables, occurrence sites for the studied species and the prediction of its distribution in the invaded range. These predictions are provided by SDM calibrated on the native range.

Usage

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

Format

ecospat.testNiche.inv is a data frame with the following columns:

`x` Longitude, in WGS84 coordinate system of the species occurrence.

`y` Latitude, in WGS84 coordinate system of the species occurrence.

`aetpet` Ratio of actual to potential evapotranspiration.

`gdd` Growing degree-days above 5 degrees C.

`p` Annual amount of precipitations.

`pet` Potential evapotranspiration.

`stdp` Annual variation of precipitations.

`tmax` Maximum temperature of the warmest month.

`tmin` Minimum temperature of the coldest month.

`tmp` Annual mean temperature.

`species_occ` Presence records of the species occurrence.

`predictions` Species Distribution Model predictions of the studied species.

Details

The study area is Australia, which is the invaded range of the hypothetical species.

Eight topo-climatic explanatory variables to quantify niche differences: ratio of the actual potential evapotranspiration; growing degree days; precipitation; potential evapotranspiration; annual variation of precipitations; maximum temperature of the warmest month; minimum temperature of the coldest month; and annual mean temperature.

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com> and Valeria Di Cola <valeria.dicola@unil.ch>

References

Petitpierre, B., C. Kueffer, O. Broennimann, C. Randin, C. Daehler and A. Guisan. 2012. Climatic niche shifts are rare among terrestrial plant invaders. *Science*, **335**, 1344-1348.

See Also

ecospat.testNiche.nat

Examples

```
data(ecospat.testNiche.inv)
str(ecospat.testNiche.inv)
dim(ecospat.testNiche.inv)
names(ecospat.testNiche.inv)
```

---

ecospat.testNiche.nat *Test Data For The Niche Dynamics Analysis In The Native Range Of A Hypothetical Species*

Description

Data frame that contains geographical coordinates, environmental variables, occurrence sites for the studied species and the prediction of its distribution in the native range. These predictions are provided by SDM calibrated on the native range.

Usage

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

Format

ecospat.testNiche.nat is a data frame with the following columns:

`x` Longitude, in WGS84 coordinate system of the species occurrence.

`y` Latitude, in WGS84 coordinate system of the species occurrence.

`aetpet` Ratio of actual to potential evapotranspiration.

`gdd` Growing degree-days above 5 degrees C.

`p` Annual amount of precipitations.

`pet` Potential evapotranspiration.

`stdp` Annual variation of precipitations.

`tmax` Maximum temperature of the warmest month.

`tmin` Minimum temperature of the coldest month.

`tmp` Annual mean temperature.

`species_occ` Presence records of the species occurrence.

`predictions` Species Distribution Model predictions of the studied species.

Details

The study area is North America, which is the native range of the hypothetical species.

Eight topo-climatic explanatory variables to quantify niche differences: ratio of the actual potential evapotranspiration; growing degree days; precipitation; potential evapotranspiration; annual variation of precipitations; maximum temperature of the warmest month; minimum temperature of the coldest month; and annual mean temperature.

Author(s)

Blaise Petitpierre <bpetitpierre@gmail.com> and Valeria Di Cola <valeria.dicola@unil.ch>

References

Petitpierre, B., C. Kueffer, O. Broennimann, C. Randin, C. Daehler and A. Guisan. 2012. Climatic niche shifts are rare among terrestrial plant invaders. *Science*, **335**, 1344-1348.

See Also

`ecospat.testNiche.inv`

Examples

```
data(ecospat.testNiche.nat)
str(ecospat.testNiche.nat)
dim(ecospat.testNiche.nat)
names(ecospat.testNiche.nat)
```

---

ecospat.testNichePOSNB

Test AVS Dataset For The Ecospat package

Description

The dataset is contains frequencies of 15 bacterial consortium AVS for 16s site in the Western Swiss Alps along with 4 PCA scores representing environmental axes.

Format

ecospat.testTree is a 16 rows (sites) x 18 collumns (14 AVS + 4 PCA axes) dataframe

Author(s)

Lucie Malard <lucie.malard@unil.ch> and Olivier Broennimann <olivier.broennimann@unil.ch>

References

L.A. Malard, H.K. Mod, N. Guex, O. Broennimann, E. Yashiro, E. Lara, E.D.A. Mitchell, H. Niculita-Hirzel & A. Guisan. The ecological niche of soil bacterial, archaeal, fungal and protist communities along environmental gradients in the Alps. 2021. Accepted in Soil Biology and Biochemistry.

Examples

```
data(ecospat.testNichePOSNB)
df<-ecospat.testNichePOSNB
ecospat.nichePOSNB(df,colvar=c(2),colfreq = 6:17) # 1 axes
ecospat.nichePOSNB(df,colvar=c(2:3),colfreq = 6:17) # 2 axes
ecospat.nichePOSNB(df,colvar=c(2:5),colfreq = 6:17) # 4 axes #
```

---

ecospat.testTree *Test Tree For The Ecospat package*

Description

The tree object is a phylogenetic tree of class 'phylo' (see read.tree) that contains data of 50 angiosperm species from the Western Swiss Alps.

Format

ecospat.testTree is a tree contains the following species:

[1] "Rumex_acetosa" [2] "Polygonum_bistorta" [3] "Polygonum_viviparum" [4] "Rumex_crispus" [5] "Cerastium_latifolium" [6] "Silene_acaulis" [7] "Gypsophila_repens" [8] "Vaccinium_gaultherioides" [9] "Soldanella_alpina" [10] "Cruciata_laevipes" [11] "Galium_album" [12] "Galium_anisophyllon" [13] "Galium_megalospermum" [14] "Gentiana_verna" [15] "Gentiana_bavarica" [16] "Gentiana_purpurea" [17] "Gentiana_lutea" [18] "Bartsia_alpina" [19] "Rhinanthus_alectorolophus" [20] "Prunella_grandiflora" [21] "Acinos_alpinus" [22] "Plantago_alpina" [23] "Plantago_lanceolata" [24] "Veronica_officinalis" [25] "Veronica_aphylla" [26] "Veronica_alpina" [27] "Veronica_chamaedrys" [28] "Veronica_persica" [29] "Globularia_cordifolia" [30] "Globularia_nudicaulis" [31] "Myosotis_alpestris" [32] "Myosotis_arvensis" [33] "Aposeris_foetida" [34] "Centaurea_montana" [35] "Hieracium_lactucella" [36] "Leontodon_helveticus" [37] "Leontodon_autumnalis" [38] "Hypochaeris_radicata" [39] "Achillea_atrata" [40] "Achillea_millefolium" [41] "Homogyne_alpina" [42] "Senecio_doronicum" [43] "Adenostyles_glabra" [44] "Arnica_montana" [45] "Aster_bellidiastrum" [46] "Bellis_perennis" [47] "Doronicum_grandiflorum" [48] "Phyteuma_orbiculare" [49] "Phyteuma_spicatum" [50] "Campanula_rotundifolia"

Author(s)

Charlotte Ndiribe <charlotte.ndiribe@unil.ch>, Nicolas Salamin <nicolas.salamin@unil.ch> and Antoine Guisan <antoine.guisan@unil.ch>

References

Ndiribe, C., L. Pellissier, S. Antonelli, A. Dubuis, J. Pottier, P. Vittoz, A. Guisan and N. Salamin. 2013. Phylogenetic plant community structure along elevation is lineage specific. *Ecology and Evolution*, **3**, 4925-4939.

Examples

```
fpath <- system.file("extdata", "ecospat.testTree.tre", package="ecospat")
library(ape)
tree <- read.tree(fpath)
plot(tree)
```

---

ecospat.varpart *Variation Partitioning For GLM Or GAM*

Description

Perform variance partitioning for binomial GLM or GAM based on the deviance of two groups or predicting variables.

Usage

```
ecospat.varpart (model.1, model.2, model.12)
```

Arguments

`model.1`	GLM / GAM calibrated on the first group of variables.
`model.2`	GLM / GAM calibrated on the second group of variables.
`model.12`	GLM / GAM calibrated on all variables from the two groups.

Details

The deviance is calculated with the adjusted geometric mean squared improvement rescaled for a maximum of 1.

Value

Return the four fractions of deviance as in Randin et al. 2009: partial deviance of model 1 and 2, joined deviance and unexplained deviance.

Author(s)

Christophe Randin <christophe.randin@unibas.ch>, Helene Jaccard and Nigel Gilles Yoccoz

References

Randin, C.F., H. Jaccard, P. Vittoz, N.G. Yoccoz and A. Guisan. 2009. Land use improves spatial predictions of mountain plant abundance but not presence-absence. *Journal of Vegetation Science*, **20**, 996-1008.

Examples

```
if(require("rms",quietly=TRUE)){
  data('ecospat.testData')

  # data for Soldanella alpina and Achillea millefolium
 data.Solalp<- ecospat.testData[c("Soldanella_alpina","ddeg","mind","srad","slp","topo")]

  # glm models for Soldanella alpina

  glm.Solalp1 <- glm("Soldanella_alpina ~ pol(ddeg,2) + pol(mind,2) + pol(srad,2)",
                 data = data.Solalp, family = binomial)
  glm.Solalp2 <- glm("Soldanella_alpina ~ pol(slp,2) + pol(topo,2)",
                 data = data.Solalp, family = binomial)

  ecospat.varpart (model.1= glm.Solalp1, model.2= glm.Solalp2, model.12= glm.Solalp2)
}
```

Index

∗ **file**
ecospat.max.tss, 59
ecospat.meva.table, 64
ecospat.occupied.patch, 79
ecospat.plot.kappa, 83
ecospat.plot.tss, 88
ecospat.rangesize, 92

BIOMOD.ensemble.models.out, *39*
BIOMOD.models.out, *42*
BIOMOD.projection.out, *45*
BIOMOD_EnsembleForecasting, *40*
BIOMOD_FormatingData, *41*, *43*
BIOMOD_Modeling, *38*, *43*
BIOMOD_ModelingOptions, *20*, *43*
BIOMOD_Projection, *43*

dudi.pca, *83*

ecospat *(*ecospat-package*)*, 4
ecospat-package, 4
ecospat.adj.D2.glm, *4*, 5
ecospat.binary.model, *5*, 6, *80*, *94*
ecospat.boyce, *4*, 8, *47*
ecospat.calculate.pd, *4*, 9
ecospat.caleval, *4*, 11
ecospat.CCV.communityEvaluation.bin, *4*, 12, *14*, *16*, *20, 21*
ecospat.CCV.communityEvaluation.prob, *4*, *13*, 14, *20, 21*
ecospat.CCV.createDataSplitTable, *4*, *13*, *16*, 17, *19*, *21*
ecospat.CCV.modeling, *4*, *12, 13*, *15*, *18*, 18
ecospat.climan, *4*, 22
ecospat.co_occurrences, *4*, 27, *29*
ecospat.cohen.kappa, *4*, 23, *58, 59*, *65*, *84*, *88*
ecospat.CommunityEval, *4*, 24
ecospat.cons_Cscore, *4, 5*, 25, *29*
ecospat.cor.plot, *4*, 27
ecospat.Cscore, *4*, 28
ecospat.cv.gbm, *4*, 30
ecospat.cv.glm, *4*, 31
ecospat.cv.me, *4*, 33
ecospat.cv.rf, *4*, 34
ecospat.Epred, *5*, 35
ecospat.ESM.EnsembleEvaluation, 36, *43*, *90*
ecospat.ESM.EnsembleModeling, *5*, *37, 38*, 38, *40, 41*, *43*, *46–48*
ecospat.ESM.EnsembleProjection, *5*, *39*, 40, *43*
ecospat.ESM.Modeling, *5*, *37–39*, 41, *45–49*
ecospat.ESM.Projection, *5*, *40*, *43*, 44
ecospat.ESM.responsePlot, *43*, 46
ecospat.ESM.threshold, *43*, 47
ecospat.ESM.VarContrib, *5*, *43*, 48
ecospat.grid.clim.dyn, *4*, 49, *66*, *70, 71*, *73*, *86*
ecospat.makeDataFrame, *4*, 52
ecospat.mantel.correlogram, *4*, 54
ecospat.margin, *4*, 56
ecospat.max.kappa, *4*, *23*, 58, *59*, *65*, *84*, *88*
ecospat.max.tss, *4*, *23*, *58*, 59, *65*, *84*, *88*
ecospat.maxentvarimport, *5*, 60
ecospat.mdr, *4*, 61
ecospat.mess, *4*, 63, *85*
ecospat.meva.table, *4*, *23*, *58, 59*, 64, *84*, *88*
ecospat.mpa, *5*, *7*, 65, *80*, *94*
ecospat.niche.dyn.index, *4*, 66, *67*
ecospat.niche.dynIndexProjGeo, *4*, 67, *74*
ecospat.niche.equivalency.test, *4*, 68, *73*, *87*
ecospat.niche.overlap, *4*, 70
ecospat.niche.similarity.test, *4*, *70*, 71, *83*, *87*
ecospat.niche.zProjGeo, *4*, *67*, 73
ecospat.nicheNBmean, 75
ecospat.nichePOSNB, 76

ecospat.npred, *4*, 77
ecospat.occ.desaggregation, *4*, 78
ecospat.occupied.patch, *5*, 79, *94*
ecospat.permut.glm, *4*, 81
ecospat.plot.contrib, *4*, 82
ecospat.plot.kappa, *4*, *23*, *58, 59*, *65*, 83, *88*
ecospat.plot.mess, *4*, *64*, 84
ecospat.plot.niche, *4*, *50*, 85
ecospat.plot.niche.dyn, *4*, *50*, *57*, *67*, *74*, *83*, 86
ecospat.plot.overlap.test, *4*, *83*, 87
ecospat.plot.tss, *4*, *23*, *58, 59*, *65*, *84*, 88
ecospat.poolingEvaluation, 89
ecospat.rand.pseudoabsences, *4*, 91
ecospat.rangesize, *5*, *80*, 92
ecospat.rcls.grd, *4*, 97, *98, 99*
ecospat.recstrat_prop, *4*, 98, *99*
ecospat.recstrat_regl, *4*, *98*, 99
ecospat.sample.envar, *4*, 100
ecospat.SESAM.prr, *5*, 101
ecospat.shift.centroids, *4*, 102
ecospat.testData, 103, *109*
ecospat.testEnvRaster, 107
ecospat.testMdr, 108
ecospat.testNiche, 109
ecospat.testNiche.inv, 110, *112*
ecospat.testNiche.nat, *111*, 111
ecospat.testNichePOSNB, 112
ecospat.testTree, 113
ecospat.varpart, *5*, 114

mgram, *55*

optimal.thresholds, *7*

princomp, *83*

response.plot2, *46*