

Package ‘changeRangeR’

January 9, 2024

Type Package

Title Change Metrics for Species Geographic Ranges

Version 1.1.0

Description Facilitates workflows to reproducibly transform estimates of species’ distributions into metrics relevant for conservation. For example, combining predictions from species distribution models with other maps of environmental data to characterize the proportion of a species’ range that is under protection, calculating metrics used under the International Union for Conservation of Nature (IUCN) Criteria A and B guidelines (Area of Occupancy and Extent of Occurrence), and calculating more general metrics such as taxonomic and phylogenetic diversity, as well as endemism. Also facilitates temporal comparisons among biodiversity metrics to inform efforts towards complementarity and consideration of future scenarios in conservation decisions. ‘changeRangeR’ also provides tools to determine the effects of modeling decisions through sensitivity tests.

License GPL-3

VignetteBuilder knitr

Encoding UTF-8

Depends R (>= 4.0)

Imports phylobase, raster, dplyr, rangeModelMetadata, sf, sp

Suggests ape, tidyverse, rmarkdown, testthat, tinytex, knitr, picante, wallace, R.utils, tools, dismo

RoxygenNote 7.2.3

NeedsCompilation no

Author Peter J. Galante [aut],
Samuel Chang [aut],
Andrea Paz [aut],
Jamie M. Kass [aut],
Matthew Aiello-Lammens [aut],
Beth Gerstner [aut],
Bethany A. Johnson [aut],

Cory Merow [aut],
 Elkin Noguera-Urbano [aut],
 Gonzalo E. Pinilla-Buitrago [aut],
 Dan Rosauer [aut],
 Mary E. Blair [aut, cre]

Maintainer Mary E. Blair <mblair1@amnh.org>

Repository CRAN

Date/Publication 2024-01-09 09:30:22 UTC

R topics documented:

| | |
|------------------|----|
| AOOarea | 2 |
| buildRMM | 3 |
| calc_PE | 5 |
| complementarity | 7 |
| envChange | 7 |
| futureOverlap | 9 |
| mcp | 10 |
| mcpSDM | 10 |
| ratioOverlap | 12 |
| Species Endemism | 13 |

Index 15

| | |
|---------|----------------------|
| AOOarea | <i>Calculate AOO</i> |
|---------|----------------------|

Description

Calculate area of occupancy measured in 2km resolution using a binary SDM

Usage

```
AOOarea(r, locs = NULL)
```

Arguments

| | |
|------|--|
| r | Raster layer of a binary SDM. Must be either unprojected in the WGS84 datum, or projected in an equal area projection (see IUCN guidelines) measured in meters. Unprojected rasters will result in an estimate while those projected in an equal area projection will be more accurate. Raster values must be either NA indicating absence, or 1, indicating presence. |
| locs | (optional) data.frame of occurrence records: Longitude and latitude. If provided, AOO of cells containing occurrence points is returned. If NULL, AOO of SDM is returned. The projection should match that of r. |

Value

a list of three objects. The first object is a character showing the AOO of cells with occurrence records. The second is a raster object of the resampled AOO. The third object (optional) represents the AOO raster showing pixels in which the localities occur, resampled to 2kmx2km (only if locs argument is supplied).

Examples

```
# create raster
r1 <- raster::raster(nrows=5, ncols=5, xmn=-5, xmx=5, ymn=-5, ymx=5)
raster::values(r1)<- runif(n = (5*5))
r1[r1 < 0.5] <- NA
r1[!is.na(r1)] <- 1
# calculate aooArea
Aooarea(r = r1)
```

 buildRMM

Create metadata objects from 'changeRangeR'

Description

Creates and populates a 'rangeModelMetadata' object from the output of 'changeRangeR'. See Merow et al. (2019) <doi:10.1111/geb.12993> for more details on the nature of the metadata and the 'rangeModelMetadata' package. To improve reproducibility of the study, this metadata object can be used as supplemental information for a manuscript, shared with collaborators, etc.

Usage

```
buildRMM(
  rmm = NULL,
  binaryRange = NULL,
  locs = NULL,
  Aooarea = NULL,
  PE = NULL,
  PhyloTree = NULL,
  complementarity = NULL,
  complementarity.of = NULL,
  complementarity.mask = NULL,
  envChange = NULL,
  envChange.rStack = NULL,
  envChange.binaryRange = NULL,
  envChange.threshold = NULL,
  envChange.bound = NULL,
  envChange.correlation = NULL,
  futureOverlap = NULL,
  futureOverlap.binRasters = NULL,
  futureOverlap.futures = NULL,
```

```

mcp = NULL,
mcpSDM = NULL,
ratioOverlap = NULL,
ratioOverlap.shape = NULL,
ratioOverlap.field = NULL,
SE = NULL,
SE.ranges = NULL
)

```

Arguments

| | |
|-----------------------|---|
| rmm | A 'rangemodelmetadata' object. |
| binaryRange | Raster layer of a binary SDM. Must be either unprojected in the WGS84 datum, or projected in a UTM projection measured in meters. |
| locs | 'data.frame' of occurrence records: Longitude and latitude. |
| AOOarea | The output of the function 'AOOarea'. |
| PE | The output of the function 'calc_PE'. |
| PhyloTree | class phylo object of phylogenetic tree. The names on tree's tip labels need to match the column names on the matrix. |
| complementarity | The out put of the function 'complementarity'. |
| complementarity.of | 'raster' of categorical values (e.g., a species richness map). |
| complementarity.mask | 'raster' object representing areas of interest (e.g., protected areas). |
| envChange | The output of the function 'envChange' |
| envChange.rStack | 'rasterStack' of environmental variable to measure within binary SDM through time. |
| envChange.binaryRange | 'raster' object or shapefile of binary range (SDM, AOO, EOO) with same projection as rStack. |
| envChange.threshold | integer (or integers if bound = "both") of where rStack layers should be thresholded. |
| envChange.bound | character string characterizing the way the threshold should happen. "upper" removes values above the threshold (e.g., maximum human footprint). "lower" removes values below the threshold (e.g., minimum forest cover). "neither" does not threshold at any point. "both" thresholds at both threshold values (if provided; e.g., minimum and maximum temperature). |
| envChange.correlation | boolean. If FALSE, environmental variable will be converted to a binary map and used as a mask. If TRUE, environmental variable is only thresholded by bounds, but left continuous. Then, Pearson's correlation coefficient with SDM will be computed for overlapping areas. |

| | |
|--------------------------|--|
| futureOverlap | The output of the function ‘futureOverlap’. |
| futureOverlap.binRasters | ‘list’ of rasters of binary SDMs.. |
| futureOverlap.futures | ‘list’ of ‘SpatialPolygons*’ objects with same CRS as futureOverlap.binRasters. |
| mcp | The output of the function ‘mcp’ |
| mcpSDM | The output of the function ‘mcpSDM’ |
| ratioOverlap | The output of the function ‘ratioOverlap’ |
| ratioOverlap.shape | Either 1) a ‘shapefile’ of land cover features or 2) a continuous ‘raster’. Must be in same projection as r parameter. If shp is a raster, then the number of cells within each quantile are calculated. |
| ratioOverlap.field | A character string representing the shapefile field attribute containing the features to compare (i.e., the column name). |
| SE | The output of the function ‘SE’ |
| SE.ranges | a ‘rasterStack’ of binary species presences |

Value

Populated rmm object

Author(s)

pgalante@amnh.org

References

Merow, C., Maitner, B. S., Owens, H. L., Kass, J. M., Enquist, B. J., Jetz, W., & Guralnick, R. (2019). Species’ range model metadata standards: RMMS. *Global Ecology and Biogeography*, **28**: 1912-1924. doi:[10.1111/geb.12993](https://doi.org/10.1111/geb.12993)

Examples

```
buildRMM(binaryRange = NULL)
```

calc_PE

Calculate phylogenetic endemism

Description

Calculates phylogenetic endemism from a tree and a sites by tips (species) matrix.

Usage

```
calc_PE(phylo.tree, sites_x_tips, presence = NULL)
```

Arguments

| | |
|--------------|--|
| phylo.tree | class phylo object of phylogenetic tree. The names on tree's tip labels need to match the column names on the matrix |
| sites_x_tips | class data.frame object. Rows should be locations and columns should be species. The sites should represent equal areas (presumably grid cells). There is no need to include unoccupied grid cells. One easy way to get this, is to simply round the coordinates to the appropriate grid resolution, and then group occurrences at the same rounded location together using aggregate or dplyr |
| presence | character string of either: "presence", "abundance", or "probability". presence specifies what the values in the matrix cells mean, and how to calculate PE abundance is an amount (could be number of individuals, proportion of cell occupied etc). With abundance, PE is equivalent to Caddotte & Davies BED) probability is a value from 0 to 1, for example from an SDM. Probability then propagates to internal branches at the probability that any of the descendent branches are present. This method is described in a paper of mine in, which is being pending minor revisions. |

Value

dataframe showing the PE at each site

Examples

```
## Convert raster stack to points
r1 <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(r1)<- runif(n = (108*108))
r1[r1 < 0.5] <- NA
r1[r1 > 0.5] <- 1
r2 <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(r2)<- runif(n = (108*108))
r2[r2 < 0.5] <- NA
r2[r2 > 0.5] <- 1
r3 <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(r3)<- runif(n = (108*108))
r3[r3 < 0.5] <- NA
r3[r3 > 0.5] <- 1
rStack <- raster::stack(r1, r2, r3)
Allxy <- raster::rasterToPoints(rStack)
# Drop first 2 columns (lat/long)
sites <- Allxy[,2:ncol(Allxy)]
sites[is.na(sites)] <- 0
library(ape)
tree <- rtree(n = 3)
tree$tip.label <- names(rStack)
calc_PE(phylo.tree = tree, sites_x_tips = sites, presence = "presence")
```

complementarity *Compare raster values within and outside of a mask*

Description

Compare raster values within and outside of a mask

Usage

```
complementarity(ras1, ras1mask)
```

Arguments

ras1 'raster' object of categorical values (e.g., a species richness map)
 ras1mask 'raster' object representing areas of interest (e.g., protected areas)

Value

A list of two objects. The first is the proportion of the range within the mask, and the second is the proportion of unique values masked.

Examples

```
# create raster
ras1 <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(ras1) <- runif(n = (108*108))
ras1[ras1 < 0.5] <- NA
ras1[!is.na(ras1)] <- 1
# create ras1mask
ras1mask <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(ras1mask) <- runif(n = (108*108))
ras1mask[ras1mask < 0.15] <- NA
ras1mask[!is.na(ras1mask)] <- 1
# complementarity
complementarity(ras1, ras1mask)
```

envChange *Calculate change in suitable SDM area through time*

Description

Calculate SDM area after masking for environmental variables through time

Usage

```
envChange(rStack, binaryRange, threshold, bound, correlation = FALSE)
```

Arguments

| | |
|-------------|---|
| rStack | rasterStack of environmental variable to measure within binary SDM through time |
| binaryRange | raster object or shapefile of binary range (SDM, AOO, EOO) with same projection as rStack |
| threshold | integer (or integers if bound = "both") of where rStack layers should be thresholded |
| bound | character string characterizing the way the threshold should happen. "upper" removes values above the threshold (e.g., maximum human footprint). "lower" removes values below the threshold (e.g., minimum forest cover). "neither" does not threshold at any point. "both" thresholds at both threshold values (if provided; e.g., minimum and maximum temperature). |
| correlation | boolean. If FALSE, environmental variable will be converted to a binary map and used as a mask. If TRUE, environmental variable is only thresholded by bounds, but left continuous. Then, Pearson's correlation coefficient with SDM will be computed for overlapping areas. |

Value

A list two objects. Area is a data.frame showing the total area of each masked raster. masks is a rasterStack of each masked raster.

Author(s)

pgalante@amnh.org

Examples

```
# create rStack
r1 <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(r1) <- runif(n = (108*108))
r2 <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(r2) <- runif(n = (108*108))
rStack <- raster::stack(r1,r2)
# create binaryRange
binaryRange <- raster::crop(r1, raster::extent(c(-50, 50, 0, 90)))
binaryRange <- raster::extend(binaryRange, r1)
binaryRange[!is.na(binaryRange)] <- 1
# set threshold
threshold <- 0.5
# set bound
bound <- "upper"
# Run function
envChange(rStack = rStack, binaryRange = binaryRange, threshold = threshold, bound = bound)
```

| | |
|---------------|--|
| futureOverlap | <i>Calculate the ratio of future overlap of SDMs with shapefile categories</i> |
|---------------|--|

Description

Calculate future overlap of SDMs with shapefile categories

Usage

```
futureOverlap(r, futures, field, category, r.names, futures.names)
```

Arguments

| | |
|---------------|--|
| r | list of rasters of binary SDMs |
| futures | List of SpatialPolygons* objects with same CRS as r |
| field | The shapefile field attribute containing the features to compare (i.e., the column name). |
| category | a list of the names of shapefile features to compare. If all features are to be used, input "All". |
| r.names | list of character values of the names representing each raster in r |
| futures.names | list of character values of the names representing each SpatialPolygons* object in futures. |

Value

a matrix showing the overlap between raster names and features.

Examples

```
#create rasters
r1 <- raster::raster(nrows=108, ncols=108, xmin=-50, xmax=50)
raster::values(r1) <- runif(n = (108*108))
r2 <- r1 >= 0.5
r2[r2 < 1] <- NA
r3 <- r1 >= 0.75
r3[r3 < 1] <- NA
# Create r
r <- list(r2, r3)
# create r.names
r.names <- c('scenario 1', 'scenario 2')
# create futures
coords <- dismo::randomPoints(r1, 3)
future <- sp::Polygon(coords)
future <- sp::SpatialPolygons(list(sp::Polygons(list(future), ID = "a")))
futures <- list(future, future)
futures.names <- list("fut1", "fut2")
# set field and category
```

```

field = "a"
category = "All"
# run function
futureOverlap(r, futures, field, category, r.names, futures.names)

```

mcp

Minimum Convex Hull Polygon

Description

Generates a minimum convex polygon (MCP; i.e., convex hull) that is delineated from occurrence locality coordinates. This function is a wrapper for ‘chull()’ that makes a SpatialPolygons object.

Usage

```
mcp(xy, crs = NULL)
```

Arguments

| | |
|-----|--|
| xy | Matrix or Data frame of occurrence coordinates |
| crs | Character of coordinate reference system for minimum convex hull |

Value

a SpatialPolygons object of the minimum convex hull around occurrences.

Examples

```

# generate occurrences
ras1 <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(ras1) <- runif(n = (108*108))
occs <- dismo::randomPoints(ras1, 4)
# create mcp
mcp(occs)

```

mcpSDM

SDM-based Minimum Convex Hull Polygon

Description

Implements the technique to estimate IUCN's extent of occurrence (EOO) geographic range estimate of species threat level by delineating a minimum convex polygon (i.e., convex hull) around a thresholded SDM prediction, first described by Syfert et al. (2014) <doi:10.1016/j.biocon.2014.06.012>. For each increment of 0.01 between a user-specified threshold and the maximum SDM prediction value, the prediction is thresholded to this value to make a binary raster. This raster is then converted to points, which are used to delineate a trial MCP. Each trial MCP is spatially intersected with the original MCP (based on the occurrence coordinates) and the original occurrence points. The Jaccard similarity index is calculated to determine geographic similarity between the trial and observed MCP. The trial MCP is also spatially intersected with the original occurrence points to determine how many were omitted. The "best" MCP is the one that has the highest JSI and also omits the least original occurrence points.

Usage

```
mcpSDM(p, xy, ch.orig, thr)
```

Arguments

| | |
|---------|--|
| p | Raster* object of a continuous species distribution model prediction to base hull calculation on |
| xy | Matrix or Data frame of occurrence coordinates |
| ch.orig | SpatialPolygons object of original minimum convex hull based on occurrence locality coordinates |
| thr | Numeric threshold used to convert the continuous SDM prediction to a binary range map; this is then used to delineate the hull |

Value

a list of 5 objects.

Note

Thresholds for SDM predictions greater than the minimum suitability across all occurrence localities will result in some occurrences being excluded from the EOO, which does not match the definition from IUCN.

References

Syfert, M. M., Joppa, L., Smith, M. J., Coomes, D. A., Bachman, S. P., & Brummitt, N. A. (2014). Using species distribution models to inform IUCN Red List assessments. *Biological Conservation*, 177, 174–184. <https://doi.org/10.1016/j.biocon.2014.06.012>

Examples

```
# create continuous raster
p <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(p) <- runif(n = (108*108))
```

```

raster::crs(p) <- "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"
# create occurrences
xy <- dismo::randomPoints(p, 4)
# create original convex hull
ch.orig <- mcp(xy, crs = "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs")
# set threshold
thr <- 0.5
# mcpSDM
mcpSDM(p, xy, ch.orig, thr)

```

| | |
|--------------|---|
| ratioOverlap | <i>Calculate the proportion of a range area that is either 1: contained by geographic categories, or 2: correlated with a continuous environmental layer.</i> |
|--------------|---|

Description

Calculate the proportion of the species' range (e.g., a thresholded SDM) that is contained by land-cover categories taken from a shapefile. Example shapefile categories include protected areas, threatened areas. ratioOverlap returns a list of the masked raster layer and the percent of the total range that lies within the shapefile polygons specified. If shp is a raster representing an environmental layer, the proportion of presence within quantiles of the environmental layer and the correlation between the layer and the range are returned.

Usage

```

ratioOverlap(
  r,
  shp = NULL,
  rasMask = NULL,
  field = NULL,
  category = NULL,
  subfield = FALSE,
  quant = "quartile"
)

```

Arguments

| | |
|---------|--|
| r | either categorical raster or shapefile object representing a binary range. |
| shp | either 1) a shapefile of land cover features or 2) a continuous raster. Must be in same projection as r parameter. If shp is a raster, then the number of cells within each quantile are calculated. |
| rasMask | (optional) a raster layer to calculate the Pearson correlation with the object r. Only if r or shp is a raster layer. |
| field | The shapefile field attribute containing the features to compare (i.e., the column name). |

| | |
|----------|---|
| category | a list of the names of shapefile features to compare. If all features are to be used, input "All". |
| subfield | boolean. If TRUE, the overlap ratio of all unique categories of shp will be calculated. |
| quant | Either the character string "quartile" for the ratio of each quartile, or a concatenation of values to use instead. |

Value

a list of three objects. The first object is a raster object showing the masked range. The second is a character showing the percentage of range within the category of interest. The third shows the correlation with rasMask if it is supplied.

Examples

```
# create binary raster
r <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(r) <- runif(n = (108*108))
r[r < 0.5] <- NA
r[r > 0.5] <- 1
# create shp
shp <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(shp) <- runif(n = (108*108))
# ratioOverlap
ratioOverlap(r = r, shp = shp)
```

| | |
|------------------|-----------------------------------|
| Species Endemism | <i>Calculate species endemism</i> |
|------------------|-----------------------------------|

Description

Calculate species endemism as the number of species in a place divided by the total number of places in which those species are found.

Usage

```
SpeciesEndemism(rStack)
```

Arguments

rStack a rasterStack of binary species presences

Value

Raster object showing species endemism.

Author(s)

pgalante@amnh.org

Examples

```
# create binary raster
r1 <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(r1)<- runif(n = (108*108))
r1[r1 < 0.5] <- NA
r1[r1 > 0.5] <- 1
r2 <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(r2)<- runif(n = (108*108))
r2[r2 < 0.5] <- NA
r2[r2 > 0.5] <- 1
r3 <- raster::raster(nrows=108, ncols=108, xmn=-50, xmx=50)
raster::values(r3)<- runif(n = (108*108))
r3[r3 < 0.5] <- NA
r3[r3 > 0.5] <- 1
rStack <- raster::stack(r1, r2, r3)
# calculate SE
SpeciesEndemism(rStack)
```

Index

A00area, [2](#)

buildRMM, [3](#)

calc_PE, [5](#)

complementarity, [7](#)

envChange, [7](#)

futureOverlap, [9](#)

mcp, [10](#)

mcpSDM, [10](#)

ratioOverlap, [12](#)

Species Endemism, [13](#)

SpeciesEndemism (Species Endemism), [13](#)