Package 'bossMaps'

December 30, 2016

Type Package

Title Convert Binary Species Range Maps into Continuous Surfaces Based on Distance to Range Edge

Version 0.1.0

Date 2016-12-21

Description Contains functions to convert binary (presence-absence) expert species range maps (like those found in a field guide) into continuous surfaces based on distance to range edge. These maps can then be used in species distribution models such as Maximum Entropy (Phillips 2008 <DOI:10.1111/j.0906-7590.2008.5203.x>) using additional information (such as point occurrence data) to refine the expert map.

Depends R (>= 2.10)

License MIT + file LICENSE

Imports raster, sp, rgeos, rgdal, ggplot2, scales, foreach, doParallel, methods, grDevices, stats, tidyr, utils

LazyData true

Suggests testthat, directlabels, profr, rasterVis, magrittr, dplyr, knitr, rgrass7, rmarkdown

RoxygenNote 5.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Adam M. Wilson [aut, cre], Cory Merow [aut]

Maintainer Adam M. Wilson <adamw@buffalo.edu>

Repository CRAN

Date/Publication 2016-12-30 01:07:50

R topics documented:

`asinh_trans` *Create an axis transform using the Inverse hyperbolic sine transformation*

Description

Create an axis transform using the Inverse hyperbolic sine transformation to allow log-like axis using data with negative values

Usage

```
asinh_trans()
```

Value

Adds a new transformation for use with ggplot

References

http://wresch.github.io/2013/03/08/asinh-scales-in-ggplot2.html

http://robjhyndman.com/hyndsight/transformations/

http://stackoverflow.com/questions/14504869/histogram-with-negative-logarithmic-scale-in-r

Examples

```
library(ggplot2)
ggplot(data.frame(x=seq(-1000,1000,len=200)), aes(x=x,y=x))+
geom_line(size=1)+
scale_x_continuous(trans = 'asinh',breaks=c(-1000,-100,10,-1,0,1,10,100,1000))
```

Beamys_hindei_points *Point distributional data for the Beamys hindei*

Description

Point ocurrance data from GBIF and eBird extracted from the Map Of Life (mol.org).

Usage

```
Beamys_hindei_points
```

Format

A SpatialPointsDataFrame object with 171 point locations and the following variables: `cartodb_id`, `type`, `provider`, `seasonality`, `presence`, `uncertainty`, and `date` collected.

Author(s)

Adam M. Wilson

References

MOL.org

Beamys_hindei_range *Expert range data for Beamys_hindei*

Description

Expert range data extracted from the Map Of Life (mol.org).

Usage

```
Beamys_hindei_range
```

Format

A SpatialPolygonDataFrame object with 2 versions of the expert range map

Author(s)

Adam M. Wilson

References

MOL.org

---

checkRates *Evaluate whether potential curve parameters are feasible given the range and domain geometry.*

Description

Note that parallelization is supported to expedite evaluation; see examples

Usage

```
checkRates(rdist, dists = NULL, prob = seq(0.1, 1, len = 10),
  rate = exp(seq(log(0.01), log(10), len = 10)), skew = c(0.2), shift = 0,
  verbose = T, plot = T)
```

Arguments

rdist	raster* object of distances to the range edge (output from rangeDist
dists	frequency table of unique distance values (output from running freq on the rdist object). If NULL, it will be created within the function but can also be supplied here to speed up processing multiple logistic parameters
prob	vector of "probability inside" values to consider
rate	vector of rate values to consider
skew	vector of skew values to consider
shift	vector of shift values to consider
verbose	logical indicating whether to print verbose messages
plot	logical indicating whether to draw a plot of desired probability vs. decay rate.

Value

dataframe of fitted parameters for each parameter combination provided with prob, rate, skew, and shift.

Examples

```
data(Beamys_hindei_range)
# Generate domain
 domain=raster::raster(xmn=-180, xmx=180, ymn=-90, ymx=90,
 crs="+proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +no_defs",
 resolution=.1, vals=NULL)
 # registerDoParallel(2) #optionally use to speed up calculation
 # Calculate distance to range
 rdist=rangeDist(range=Beamys_hindei_range,domain=domain,
                 domainkm=1000,mask=FALSE,fact=2,verbose=FALSE)
 dpar=checkRates(rdist,plot=FALSE)
 head(dpar)
```

`compress` *Compress maxent output by converting from ASCII file to compressed geotif*

Description

Maxent writes out ASCII files that are uncompressed. This function transforms the data, converts it to a compressed geotif, and deletes the original. The default transform `rtrans` is to take the $round(log(x) * 100)$. Note that this conversion may be lossy depending on the transform applied.

Usage

```
compress(file, fun = rtrans)

rtrans(x)

unrtrans(x)
```

Arguments

`file`	`character` path to file to be compressed
`fun`	Function to use to compress the data. Default is `rtrans()`, which is `round(log(x)*100`.
`x`	value passed to `rtrans` or `untrans` for conversion.
`...`	additional functions to be passed to `writeRaster()`

Value

Returns the converted file as a raster() object

Functions

- `rtrans`: transform data to faciliate storing as an integer as $round(log(x) * 100)$.
- `unrtrans`: uncompress data transformed with `rtrans`

`cumulative` *Make maxent's cumulative output from raw output and (optionally) apply a threshold*

Description

The function first converts the raw maxent output to cumulative (by sorting and calculating the cumulative sum). Then the (optional) threshold is applied to return a binary map of the estimated range. If `threshold` is `NULL`, the cumulative output is returned.

Usage

```
cumulative(x, threshold = NULL)
```

Arguments

`x`	raster* object of continuous (raw) maxent output
`threshold`	`scalar` value to threshold the continuous map into binary predictions

`fbbi` *Factor Bias Back In (FBBI)*

Description

Calculate minxent layer using maxent output (from run with bias layer) and prior (i.e., the bias layer)

Usage

```
fbbi(prior, maxent, ...)
```

Arguments

`prior`	raster* object of spatial priors
`maxent`	raster* object of maxent output
`...`	additional functions to be passed to `writeRaster`

hcols *Aesthetically pleasing color ramp for plotting habitat suitability*

Description

Aesthetically pleasing color ramp for plotting habitat suitability

Usage

```
hcols(x, bias = 1)
```

Arguments

x	number of colors to return
bias	a positive number. Higher values give more widely spaced colors at the high end (as in colorRampPalette).

Value

Returns a character vector of colors (see rgb) interpolating the given sequence (similar to heat.colors or terrain.colors.

Examples

```
library(ggplot2)
data=cbind.data.frame(expand.grid(x=1:10,y=1:10),z=rnorm(100))
ggplot(data,aes(x=x,y=y,fill=z))+geom_raster()+
scale_fill_gradientn(colours=hcols(20,bias=2))
```

Leucadendron_lanigerum_range *Expert range data for Leucadendron lanigerum*

Description

Expert range data for Leucadendron lanigerum, a South African Shrub.

Usage

```
Leucadendron_lanigerum_range
```

Format

A Raster layer object with 1s for the expert range, 0s for the rest of the domain, and NAs outside the domain.

Author(s)

Adam M. Wilson

References

MOL.org

`logistic` *Apply a custom logisitic transform to a vector*

Description

Performs a generalized logistic transform (aka Richard's curve) as follows:

$$Y(t) = upper - \frac{upper - lower}{(1 + e^{-rate(x-shift)})^{1/skew}}$$

Usage

```
logistic(x, lower = 0, upper = 1, rate = 0.04, skew = 0.2, shift = 0,
  parms = NULL)
```

Arguments

`x`	`vector` or `raster*` object to apply logistic transform.
`lower`	The lower asymptote
`upper`	The upper asymptote
`rate`	The rate of decay. A rate of 0 indicates no decay, while large rates (>10) become like step-functions.
`skew`	Affects near which asymptote maximum rate (`skew` must be >0). If skew=1, the curve becomes a standard symetrical logistic centered at `m`. As skew approaches zero, it becomes more asymetrical.
`shift`	The time of maximum decay. Slides the curve left-right.
`parms`	an optional list with each of the following named parameters (e.g. `list(lower=0.01,upper=1,rate=0.0` Use of `parms` rather than naming each parameter separately is for convenience with the `optim` function.

References

Richards, F. J. (1959). 'A Flexible Growth Function for Empirical Use'. Journal of Experimental Botany 10 (2): 290-300.

Examples

```
curve(logistic(x,rate=0.05,skew=0.5),-150,300)
```

normalize *Normalize a raster*

Description

Divide raster by the sum of all cells.

Usage

```
normalize(x, ...)
```

Arguments

x	raster* object
...	additional functions to be passed to writeRaster

optifix *optifix. Optimise with some fixed parameters*

Description

Like optim, but with option to fix some parameters.

Usage

```
optifix(parms, fixed, fn, gr = NULL, ..., method = c("Nelder-Mead", "BFGS",
  "CG", "L-BFGS-B", "SANN"), lower = -Inf, upper = Inf, control = list(),
  hessian = FALSE)
```

Arguments

parms	Parameters to potentially optimize in fn
fixed	A vector of TRUE/FALSE values indicating which parameters in parms to hold constant (not optimize). If TRUE, the corresponding parameter in fn() is fixed. Otherwise it's variable and optimised over.
fn	Function to optimize (same as in optim)
gr	Gradient function (same as in optim)
...	Further arguments to be passed to fn and gr.
method	Method to use for optimization (same as in optim)
lower	Lower limits (same as in optim)
upper	Upper limits (same as in optim)
control	Control list (same as in optim)
hessian	Return Hessian object (same as in optim)

Details

Originally written by Barry Rowlingson"

Value

Similar to optim but adds a vector of all the parameters and a vector copy of the 'fixed' argument.

Author(s)

Written by Barry Rowlingson October 2011

References

http://www.maths.lancs.ac.uk/~rowlings/R/Optifix/optifix.R

---

pinside *Calculate the probability inside an expert range given a logistic decay*

Description

Uses the logistic function to estimate the probability inside the range

Usage

```
pinside(dists, parms)
```

Arguments

dists	frequency table of unique distance values (output from running freq on the rdist object).
parms	A named vector of parameter values from logistic that describe the desired curve. These include all parameters for logistic (rate, skew, shift). Optionally, you can add a parameter buffer which adds a buffer around the range as 'inside.'

Value

Scalar of the estimated probability inside the range given the specified logistic decay (with optional buffer).

rangeDist *Use Calcuate distance to range edge*

Description

Calculates distance to range boundary and returns a raster object. Range is projected to World Azimuthal Equidistant (centered on range centroid) before buffering. Starting a cluster with `beginCluster()` can improve processing time if multiple cores are available.

Usage

```
rangeDist(range, points, domain, domainkm = 1000, fact = 2, mask = FALSE,
  verbose = TRUE, ...)
```

Arguments

`range`	SpatialPolygons* object with species range
`points`	Optional SpatialPoints* object with species occurrences to be included in defining the domain, but not in the distance calculations.
`domain`	Empty raster extent or path to geotif representing the potential modelling domain with resolution, projection, etc.
`domainkm`	Define the distance (in km) from expert range to include in the modeling domain. The range polygon will be buffered by this value and used to define a rectangular region around the range. See `mask` to additionally mask pixels with distances farther than a specified value.
`fact`	`numeric` aggregation factor used to speed up distance calculation (larger numbers faster but less accurate).
`mask`	`logical` indicating whether to mask the domain to the range buffered by `domainkm`. If `FALSE` (the default), the returned object will have complete data across a rectangular domain. If `TRUE`, the rectangular domain will be masked by the buffered range.
`verbose`	`logical` indicating whether to print status messages.
`...`	additional functions to be passed to `writeRaster`

rangeDist_grass *Use GRASS to calcuate distance to range edge*

Description

Calculates distance to range boundary and returns a raster object. Alternatively, one could use `raster::distance()` but it is very slow with large rasters. This function uses GRASS (and requires grass to be available), but is much faster.

Usage

```
rangeDist_grass(spdata, domain, domainkm = 1000, mask = T, verbose = T,
  clean = F, gisBase = NULL, file)
```

Arguments

spdata	list object with range and points
domain	Empty raster extent or path to geotif representing the potential modelling domain with resolution, projection, etc.
domainkm	Distance threshold (in km) from expert range to set species domain.
mask	logical indicating whether to mask the distance to values <= domainkm.
verbose	logical indicating whether to print status messages.
clean	logical indicating whether to delete all temporary GRASS files (the default) or keep them. By default these files are in the current tempdir() and will be deleted when quitting R.
gisBase	Path to GRASS Binaries as required by initGRASS
file	filename for output file exported from GRASS.

---

rangeOffset — *Generate a spatial map of an expert map (with decay) to be used as an offset*

Description

Uses the logistic function to transform the rdist object to a normalized expert range with a desired proportion of the total probability inside the range. You can set fitdist to include a buffer around the range as 'inside' for the purposes of the prior expectation of what Typically the user supplies prob, rate, and skew as fixparms while upper and lower are estimated.

Usage

```
rangeOffset(rdist, parms, dists = NULL, returnRaster = TRUE,
  doNormalize = TRUE, verbose = TRUE, doWriteRaster = FALSE,
  doWriteMetadata = TRUE, ...)
```

Arguments

rdist	raster* object of distances to the range edge (output from rangeDist
parms	A named vector of parameter values from logistic that describe the desired curve. These include all parameters for logistic (rate, skew, shift) and two additional parameters named prob (indicating the desired probability inside the range, e.g. 0.95) and buffer (a desired buffer, in meters) outside the expert range that should be considered 'inside' the range. See description for details.

dists	frequency table of unique distance values (output from running freq on the rdist object). If NULL, it will be created within the function but can also be supplied here to speed up processing multiple logistic parameters
returnRaster	logiical indicating whether to calculate the full spatial prior and return the raster. If FALSE, the result will simply be the fitted parameters.
doNormalize	logical indicating whether to normalize the raster before returning it. If FALSE, the result will range from lower to upper.
verbose	logical indicating whether to print verbose messages
doWriteRaster	logical indicating whether to write the output to disk
doWriteMetadata	logical indicating whether to write summary metadata to disk as a .csv file
...	additional functions to be passed to writeRaster

Examples

```
## Not run:
  library(raster)
  library(ggplot2)

  data("Tinamus_solitarius_points")
  data("Tinamus_solitarius_range")

  ## Define global modeling grid
  domain = raster(
    xmn = -180,
    xmx = 180,
    ymn = -90,
    ymx = 90,
    crs = "+proj=longlat +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +no_defs",
    resolution = 10 / 360,
    vals = NULL
  )

  ## turn on raster progress bar
  rasterOptions(progress = "")

  ## calculate distance-to-range: this is slow but only has to
  ## be done once per species.  Can speed it up by increasing
  ## 'fact' (at the expense of reduced accuracy).
  range = Tinamus_solitarius_range
  points=Tinamus_solitarius_points

  rdist = rangeDist(range=range,
                    domain=domain,
                    domainkm = 100,
                    mask = FALSE,
                    fact = 10)

  ## Mask out undesired areas (ocean, etc.)  Typically you would
  ## do this using your environmental data, but here we'll just
```

```
  ## use a coastline polygon from the maps package
#    land = map(
#    interior = F,
#    fill = T,
#    xlim = bbox(rdist)[1, ],
#    ylim = bbox(rdist)[2, ]
#  )
#  land = map2SpatialPolygons(land, IDs = land$names)
#  rdist = mask(rdist, land)

  ## calculate frequency table of distances
  dists = freq(rdist)

  ### plot to visualize potential decay parameters
  vars = expand.grid(
    rate = c(0, 0.03, 0.05, 0.1, 10),
    skew = c(0.2,
             0.4),
    shift = 0,
    stringsAsFactors = FALSE
  )
  x = seq(-150, 300, len = 1000)

  ## Calculate all the curves
  erd = do.call(rbind, lapply(1:nrow(vars), function(i) {
    y = logistic(x, parms = unlist(c(
      lower = 0, upper = 1, vars[i, ]
    )))
    return(cbind.data.frame(
      group = i,
      c(vars[i, ]),
      x = x,
      y = y
    ))
  }))

  ## plot it
  ggplot(erd,
         aes(
           x = x,
           y = y,
           linetype = as.factor(skew),
           colour = as.factor(rate),
           group = group
         )) +
    geom_vline(aes(xintercept=0),
               colour = "red") + geom_line() +
    xlab("Prior value (not normalized)") +
    xlab("Distance to range edge (km)")


  ## calculate the expert range prior
  expert = rangeOffset(
```

```
    rdist,
    dists = dists,
    parms = c(
      prob = 0.9,
      rate = 0.05,
      skew = 0.4,
      shift = 0
    ),
    normalize = TRUE,
    verbose = TRUE
  )

  ## View the metadata
  metadata(expert)$parms

  ## plot it
  plot(expert)

## End(Not run)
```

`rmRaster` *Really remove raster files*

Description

The raster package has the sometimes problematic behavior of leaving temporary files after the R object has been removed. This function first checks if a temporary file exists on disk and removes it.

Usage

```
rmRaster(x, verbose = FALSE)
```

Arguments

`x`	a `raster*` object to be removed
`verbose`	logical indicating whether to print verbose messages

Author(s)

Adam M. Wilson

Keith Ma

References

https://gist.github.com/adammwilson/10180852

`Tinamus_solitarius_env` *Expert range environmental data for the Solitary Tinamou (Tinamus solitarius)*

Description

Environmental data (from WorldClim) for the Solitary Tinamou's range (mol.org). Bio1 is Mean Annual Temperature and Bio12 is Mean Annual Precipitation.

Usage

```
Tinamus_solitarius_env
```

Format

A rasterStack object

Author(s)

Adam M. Wilson

`Tinamus_solitarius_points` *Point distributional data for the Solitary Tinamou (Tinamus solitarius)*

Description

Point ocurrance data from GBIF and eBird extracted from the Map Of Life (mol.org).

Usage

```
Tinamus_solitarius_points
```

Format

A SpatialPointsDataFrame object with 171 point locations and the following variables: `cartodb_id`, `type`, `provider`, `seasonality`, `presence`, `uncertainty`, and `date` collected.

Author(s)

Adam M. Wilson

References

MOL.org

Tinamus_solitarius_range *Expert range data for the Solitary Tinamou (Tinamus solitarius)*

Description

Expert range data extracted from the Map Of Life (mol.org).

Usage

```
Tinamus_solitarius_range
```

Format

A SpatialPolygonDataFrame object with 2 versions of the expert range map

Author(s)

Adam M. Wilson

References

MOL.org

tree *Graphically display folder tree in terminal*

Description

Uses sed to display formatted folder structure. Needs sed to be installed on system (may not be available on windows)

Usage

```
tree(x = NULL)
```

Arguments

x	File path to display

Value

Prints the folder structure

References

http://www.cnet.com/news/terminal-fun-options-for-printing-folder-and-subfolder-contents/

Index