Package 'CENFA'

August 16, 2021

Type Package

Title Climate and Ecological Niche Factor Analysis

Version 1.1.1

Date 2021-07-27

Author D. Scott Rinnan

Maintainer D. Scott Rinnan <scott.rinnan@yale.edu>

Description Tools for climate- and ecological-niche factor analysis of spatial data, including methods for visualization of spatial variability of species sensitivity, exposure, and vulnerability to climate change. Processing of large files and parallel methods are supported. Climate-niche factor analysis is described in Rinnan and Lawler (2019) <doi:10.1111/ecog.03937>.

Depends R (>= 3.5.0), raster (>= 2.6.7)

Imports doSNOW (>= 1.0.16), foreach (>= 1.4.4), graphics, grDevices, magrittr, methods, parallel (>= 3.3.3), pbapply (>= 1.3.3), Rcpp, snow (>= 0.4.2), sp (>= 1.2.7)

LinkingTo Rcpp

License GPL (>= 3)

LazyData TRUE

RoxygenNote 7.1.1

Suggests knitr, maps, rmarkdown

VignetteBuilder knitr

Collate 'CENFA.R' 'GLcenfa-class.R' 'GLcenfa.R' 'GLdeparture-class.R' 'GLdeparture.R' 'RcppExports.R' 'brStick.R' 'calc.R' 'climdat.fut.R' 'climdat.hist.R' 'cnfa-class.R' 'cnfa.R' 'covij.R' 'departure-class.R' 'departure.R' 'enfa-class.R' 'enfa.R' 'exposure_map.R' 'overlay.R' 'parCov.R' 'parScale.R' 'vulnerability-class.R' 'predict.R' 'scatter.R' 'sensitivity_map.R' 'slots.R' 'stretchPlot.R' 'tree-data.R' 'vulnerability.R' 'vulnerability_map.R'

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-08-16 07:50:09 UTC

R topics documented:

`CENFA-package` *Tools for climate- and ecological-niche factor analysis*

Description

`CENFA` provides tools for performing ecological-niche factor analysis (ENFA) and climate-niche factor analysis (CNFA).

Details

This package was created with three goals in mind:

- To update the ENFA method for use with large datasets and modern data formats.

- To expand the application of ENFA in the context of climate change in order to quantify different aspects of species vulnerability to climate change, and to facilitate quantitative comparisons of vulnerability between species.

- To correct a minor error in the ENFA method itself, that has persisted in the literature since Hirzel et al. first introduced ENFA in 2002.

`CENFA` takes advantage of the `raster` and `sp` packages, allowing the user to conduct analyses directly with raster, shapefile, and point data, and to handle large datasets efficiently via partial data loading and parallelization.

In addition, `CENFA` also contains a few functions that speed up some basic ‘raster‘ functions considerably by parallelizing on a layer-by-layer basis rather than a cell-by-cell basis.

Author(s)

D. Scott Rinnan

References

Basille, Mathieu, et al. Assessing habitat selection using multivariate statistics: Some refinements of the ecological-niche factor analysis. Ecological Modelling 211.1 (2008): 233-240.

Hirzel, Alexandre H., et al. Ecological-niche factor analysis: how to compute habitat-suitability maps without absence data?. Ecology 83.7 (2002): 2027-2036.

See Also

`sp`, `raster-package`

`brStick` *Broken-stick method for detection of significant factors*

Description

This function provides a simple way to determine the number of significant factors in a factor analysis. This is done by comparing the eigenvalues of each factor with those expected from a broken-stick distribution.

Usage

```
brStick(eigs)
```

Arguments

`eigs`	numeric. Vector of eigenvalues

Value

Returns the number of significant factors.

References

Jackson, Donald A. "Stopping rules in principal components analysis: a comparison of heuristical and statistical approaches." Ecology 74.8 (1993): 2204-2214.

Examples

```
mod1 <- enfa(x = climdat.hist, s.dat = ABPR, field = "CODE")
brStick(s.factor(mod1))
```

climdat.fut *Future climate data*

Description

Future climate dataset of 10 bioclimate variables (https://www.worldclim.org). Based on the MIROC5 GCM projections under the RCP8.5 scenario for 2050.

Usage

```
climdat.fut
```

Format

A RasterBrick with 10 layers:

MDR mean diurnal range (mean of monthly max temp - min temp)

ISO isothermality (MDR/TAR * 100)

TS temperature seasonality (sd monthly temp * 100)

HMmax max temp of warmest month

CMmin min temp of coldest month

PWM precip of wettest month

PDM precip of driest month

PS precip seasonality (sd/mean monthly precip)

PWQ precip of wettest quarter

PDQ precip of driest quarter

Source

https://www.worldclim.org

See Also

climdat.hist, tree-data

climdat.hist *Historical climate data*

Description

Historical climate dataset of 10 bioclimate variables (https://www.worldclim.org).

Usage

```
climdat.hist
```

Format

A RasterBrick with 10 layers:

MDR mean diurnal range (mean of monthly max temp - min temp)

ISO isothermality (MDR/TAR * 100)

TS temperature seasonality (sd monthly temp * 100)

HMmax max temp of warmest month

CMmin min temp of coldest month

PWM precip of wettest month

PDM precip of driest month

PS precip seasonality (sd/mean monthly precip)

PWQ precip of wettest quarter

PDQ precip of driest quarter

Source

https://www.worldclim.org

See Also

climdat.fut, tree-data

cnfa *Climate-niche factor analysis*

Description

Performs climate-niche factor analysis using climate raster data and species presence data.

Usage

```
cnfa(x, s.dat, ...)

## S4 method for signature 'GLcenfa,Raster'
cnfa(
  x,
  s.dat,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE,
  ...
)

## S4 method for signature 'GLcenfa,Spatial'
cnfa(
  x,
  s.dat,
  field,
  fun = "last",
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE,
  ...
)

## S4 method for signature 'Raster,Raster'
cnfa(
  x,
  s.dat,
  scale = TRUE,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
```

```
  n = 1,
  cl = NULL,
  keep.open = keep.open,
  ...
)

## S4 method for signature 'Raster,Spatial'
cnfa(
  x,
  s.dat,
  field,
  fun = "last",
  scale = TRUE,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE,
  ...
)
```

Arguments

x	Raster* object, typically a brick or stack with p climate raster layers, or a GLcenfa object
s.dat	RasterLayer, SpatialPolygons*, or SpatialPoints* object indicating species presence or abundance
...	Additional arguments for writeRaster
filename	character. Optional filename to save the Raster* output to file. If this is not provided, a temporary file will be created for large x
progress	logical. If TRUE, messages and progress bar will be printed
parallel	logical. If TRUE then multiple cores are utilized for the calculation of the covariance matrices
n	numeric. Number of CPU cores to utilize for parallel processing
cl	optional cluster object
keep.open	logical. If TRUE and parallel = TRUE, the cluster object will not be closed after the function has finished
field	field of s.dat that specifies presence or abundance. This is equivalent to the field argument in rasterize
fun	function or character. Determines what values to assign to cells with multiple spatial features, similar to the fun argument in rasterize. Options are 'first', 'last' (default), and 'count' (see Details)
scale	logical. If TRUE then the values of x will get centered and scaled. Depending on the resolution of the climate data and the extent of the study area, this can be quite time consuming. If running this function for multiple species, it is recommended that the data be scaled beforehand using the GLcenfa function

Details

The `cnfa` function is not to be confused with the `enfa` function. `enfa` performs ENFA as described by Hirzel et al. (2002) and Basille et al. (2008), and is offered as an alternative to the `enfa` function in the `adehabitatHS` package. There are several key differences between ENFA and CNFA.

Whereas ENFA returns a **specialization factor** that describes the specialization in each **ENFA factor**, CNFA returns a **sensitivity factor** `sf` that describes the sensitivity in each **environmental variable**. This makes the sensitivity factor more directly comparable to the marginality factor `mf`, because their dimensions are identical. Sensitivity is calculated by a weighted sum of the amount of specialization found in each CNFA factor, *including* the marginality factor. As such, the sensitivity factor offers a more complete measure of specialization than ENFA's specialization factor, which does not calculate the amount of specialization found in the marginality factor. As such, CNFA's overall sensitivity (found in the slot `sensitivity`) offers a more complete measure of niche specialization than ENFA's overall specialization (found in the slot `specialization`).

The default `fun = 'last'` gives equal weight to each occupied cell. If multiple species observations occur in the same cell, the cell will only be counted once. `fun = 'count'` will weight the cells by the number of observations.

If there is too much correlation between the layers of `x`, the global covariance matrix will be singular, and the overall marginality and overall sensitivity will not be meaningful. In this case, a warning is issued, and `marginality` and `sensitivity` are both returned as `NA`.

Value

Returns an S4 object of class `cnfa` with the following components:

call Original function call

mf Marginality factor. Vector of length p that describes the location of the species Hutchinsonian niche relative to the global niche

marginality Magnitude of the marginality factor

sf Sensitivity factor. Vector of length p that describes the amount of sensitivity for each climate variable

sensitivity Square root of the mean of the sensitivity factor

eig Named vector of eigenvalues of specialization for each CNFA factor

co A p x p matrix describing the amount of marginality and specialization in each CNFA factor.

cov p x p species covariance matrix

g.cov p x p global covariance matrix

ras RasterBrick of transformed climate values, with p layers

weights Raster layer of weights used for CNFA calculation

References

Rinnan, D. Scott and Lawler, Joshua. Climate-niche factor analysis: a spatial approach to quantifying species vulnerability to climate change. Ecography (2019): <doi:10.1111/ecog.03937>.

Basille, Mathieu, et al. Assessing habitat selection using multivariate statistics: Some refinements of the ecological-niche factor analysis. Ecological Modelling 211.1 (2008): 233-240.

Hirzel, Alexandre H., et al. Ecological-niche factor analysis: how to compute habitat-suitability maps without absence data?. Ecology 83.7 (2002): 2027-2036.

See Also

GLcenfa, enfa

Examples

```
mod1 <- cnfa(x = climdat.hist, s.dat = ABPR, field = "CODE")

# using GLcenfa as an initial step
# for multi-species comparison

glc <- GLcenfa(x = climdat.hist)
mod2 <- cnfa(x = glc, s.dat = ABPR, field = "CODE")

# same results either way
all.equal(m.factor(mod1), m.factor(mod2))
all.equal(s.factor(mod1), s.factor(mod2))
```

---

cnfa-class *cnfa-class*

Description

An object of class `cnfa` is created by performing climate-niche factor analysis on species presence data using the `cnfa` function.

Slots

- `call` Original function call
- `mf` numeric. Named vector representing the marginality factor, describing the location of the species niche relative to the global niche
- `marginality` numeric. Magnitude of the marginality factor `mf`, scaled by the global covariance matrix
- `sf` numeric. Named vector representing the sensitivity factor
- `sensitivity` numeric. The magnitude of the sensitivity factor `sf`, scaled by the global covariance matrix
- `eig` numeric. Named vector representing the eigenvalues of specialization, reflecting the amount of variance on each factor
- `co` p x p matrix of standardized variable loadings
- `cov` p x p species covariance matrix
- `g.cov` p x p global covariance matrix
- `ras` RasterBrick of transformed climate values, with p layers
- `weights` Raster layer of weights used for CNFA calculation

departure *Climatic departure*

Description

This function quantifies the amount of change between historical and future climate conditions inside a species' habitat.

Usage

```
departure(x, y, s.dat, ...)

## S4 method for signature 'GLdeparture,missing,cnfa'
departure(x, s.dat, filename = "", ...)

## S4 method for signature 'GLdeparture,missing,Spatial'
departure(x, s.dat, field, fun = "last", filename = "", ...)

## S4 method for signature 'Raster,Raster,cnfa'
departure(
  x,
  y,
  s.dat,
  center = TRUE,
  scale = TRUE,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  ...
)

## S4 method for signature 'Raster,Raster,Spatial'
departure(
  x,
  y,
  s.dat,
  center = TRUE,
  scale = TRUE,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  ...
)
```

Arguments

`x`	Raster* object, typically a brick or stack of historical climate raster layers or a brick of absolute differences (see Details)
`y`	Raster* object, future climate values with the same layers as `x`
`s.dat`	SpatialPolygons*, sf, or cnfa object detailing species presence
`...`	Additional arguments for `clusterR`
`filename`	character. Optional filename to save the Raster* output to file. If this is not provided, a temporary file will be created for large `x`
`field`	field of `s.dat` that specifies presence. This is equivalent to the `field` argument of `raster::rasterize`. Options are 'first', 'last' (default), and 'count'
`fun`	function or character. Determines what values to assign to cells with multiple spatial features, similar to the `fun` argument in `rasterize`
`center`	logical. If `TRUE` then the values of `x` and `y` will be centered on the means of the historical climate data
`scale`	logical. If `TRUE` then the values of `x` and `y` will be scaled by the sds of the historical climate data
`progress`	logical. If `TRUE`, messages and progress bar will be printed
`parallel`	logical. If `TRUE` then multiple cores are utilized
`n`	numeric. Optional number of CPU cores to utilize for parallel processing

Details

For comparisons of multiple species in the same study area, it is much more efficient to first construct a Raster* object of absolute differences between the historical and future values, so that the differences do not need to be recalculated for each species. This can be achieved with by passing `x` and `y` to the `difRaster` function, and then passing the results to the `departure` function.

When only one Raster* object is supplied, it is assumed that `x` is a Raster* object containing the absolute differences of a historical and future dataset.

Value

Returns an S4 object of class `departure` with the following slots:

call Original function call

df Departure factor. Vector of length p that describes the amount of departure between future and historical conditions for each climate variable

departure Magnitude of the departure factor

g.cov p x p historical global covariance matrix

ras RasterBrick of climate departures, with p layers

weights Raster layer of weights used for departure calculation

References

Rinnan, D. Scott and Lawler, Joshua. Climate-niche factor analysis: a spatial approach to quantifying species vulnerability to climate change. Ecography (2019): <doi:10.1111/ecog.03937>.

Examples

```
dep1 <- departure(x = climdat.hist, y = climdat.fut, s.dat = ABPR, field = "CODE")

# using difRaster as an initial step
# for multi-species comparison

gld <- GLdeparture(x = climdat.hist, y = climdat.fut)
dep2 <- departure(x = gld, s.dat = ABPR, field = "CODE")

# same results either way
all.equal(dep1@df, dep2@df)
```

---

departure-class *departure-class*

Description

An object of class `departure` is created by the `departure` departure function, which quantifies the amount of change between historical and future climate conditions inside a species' habitat.

Slots

`call` Original function call

`df` Departure factor

`departure` Magnitude of the departure factor

`g.cov` historical global covariance matrix

`ras` Raster* object of transformed climate values

`weights` Raster layer of weights used for departure calculation

---

enfa *Ecological-niche factor analysis*

Description

Performs ecological-niche factor analysis using environmental raster data and species presence data.

Usage

```
enfa(x, s.dat, ...)

## S4 method for signature 'GLcenfa,Raster'
enfa(
  x,
  s.dat,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE,
  ...
)

## S4 method for signature 'GLcenfa,Spatial'
enfa(
  x,
  s.dat,
  field,
  fun = "last",
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE,
  ...
)

## S4 method for signature 'Raster,Raster'
enfa(
  x,
  s.dat,
  scale = TRUE,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE,
  ...
)

## S4 method for signature 'Raster,Spatial'
enfa(
  x,
```

```
  s.dat,
  field,
  fun = "last",
  scale = TRUE,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE,
  ...
)
```

Arguments

x	Raster* object, typically a brick or stack of ecological raster layers, or a GLcenfa object
s.dat	RasterLayer, SpatialPolygons*, or SpatialPoints* object indicating species presence or abundance
...	Additional arguments for writeRaster
filename	character. Optional filename to save the Raster* output to file. If this is not provided, a temporary file will be created for large x
progress	logical. If TRUE, messages and progress bar will be printed
parallel	logical. If TRUE then multiple cores are utilized for the calculation of the covariance matrices
n	numeric. Number of CPU cores to utilize for parallel processing
cl	optional cluster object
keep.open	logical. If TRUE and parallel = TRUE, the cluster object will not be closed after the function has finished
field	field of s.dat that specifies presence or abundance. This is equivalent to the field argument in the raster package
fun	function or character. Determines what values to assign to cells with multiple spatial features, similar to the fun argument in rasterize. Options are 'first', 'last' (default), and 'count' (see Details)
scale	logical. If TRUE then the values of the Raster* object will be centered and scaled. Depending on the resolution of the climate data and the extent of the study area, this can be quite time consuming. If running this function for multiple species, it is recommended that the climate data be scaled beforehand using the GLcenfa function

Details

The cnfa function is not to be confused with the enfa function. enfa performs ENFA as described by Hirzel et al. (2002) and Basille et al. (2008), and is offered as an alternative to the enfa function in the adehabitatHS package. CENFA::enfa will give different results than adehabitatHS::enfa for versions of adehabitatHS 0.3.13 or earlier, however, for two primary reasons.

First, `CENFA::enfa` corrects a minor mistake in the calculation of the species covariance matrix. This correction influences the values of the coefficients of specialization in each ecological variable, which will lead to a different interpretation of the degree of specialization. Second, we define the overall marginality M as the norm of the marginality factor `mf`, rather than the square of the norm of `mf`.

The default `fun = 'last'` gives equal weight to each occupied cell. If multiple species observations occur in the same cell, the cell will only be counted once. `fun = 'count'` will weight the cells by the number of observations.

If there is too much correlation between the layers of `x`, the global covariance matrix will be singular, and the overall marginality will not be meaningful. In this case, a warning is issued and `marginality` is returned as `NA`.

Value

Returns an S4 object of class `enfa` with the following components:

call Original function call

mf Marginality factor. Vector that describes the location of the species Hutchinsonian niche relative to the global niche

marginality Magnitude of the marginality factor

sf Specialization factor. Vector of eigenvalues of specialization

specialization Square root of the mean of the specialization factor

sf.prop Vector representing the proportion of specialization found in each ENFA factor

co A matrix describing the amount of marginality and specialization on each ENFA factor

ras RasterBrick of transformed climate values, with p layers

weights Raster layer of weights used for ENFA calculation

References

Basille, Mathieu, et al. Assessing habitat selection using multivariate statistics: Some refinements of the ecological-niche factor analysis. Ecological Modelling 211.1 (2008): 233-240.

Hirzel, Alexandre H., et al. Ecological-niche factor analysis: how to compute habitat-suitability maps without absence data?. Ecology 83.7 (2002): 2027-2036.

See Also

`GLcenfa`, `cnfa`

Examples

```
mod1 <- enfa(x = climdat.hist, s.dat = ABPR, field = "CODE")

# using GLcenfa as an initial step
# for multi-species comparison

glc <- GLcenfa(x = climdat.hist)
mod2 <- enfa(x = glc, s.dat = ABPR, field = "CODE")
```

```
all.equal(m.factor(mod1), m.factor(mod2))
```

enfa-class *enfa-class*

Description

An object of class `enfa` is created from performing ecological-niche factor analysis on species presence data using the `enfa` function.

Slots

- `call` Original function call
- `mf` numeric. Named vector representing the marginality factor, describing the location of the species niche relative to the global niche
- `marginality` numeric. Magnitude of the marginality factor `mf`, scaled by the global covariance matrix
- `sf` numeric. Named vector representing the specialization factor, equivalent to the eigenvalues of specialization
- `specialization` numeric. The square root of the sum of eigenvalues, divided by the length of `sf`
- `sf.prop` numeric. Named vector representing the proportion of specialization found on each factor
- `co` p x p matrix of standardized variable loadings
- `cov` p x p species covariance matrix
- `ras` RasterBrick of transformed climate values, with p layers
- `weights` Raster layer of weights used for ENFA calculation

exposure_map *Create an exposure map*

Description

Creates a map of exposure to climate change in a species' habitat from a `departure` object.

Usage

```
exposure_map(dep, parallel = FALSE, n, filename = "", ...)
```

Arguments

`dep`	Object of class `departure`
`parallel`	logical. If `TRUE` then multiple cores are utilized
`n`	numeric. Number of cores to use for calculation (optional)
`filename`	character. Output filename (optional)
`...`	Additional arguments for file writing as for `writeRaster`

Details

The values of the exposure raster are calculated by projecting onto the departure factor **d**, given by the formula

$\epsilon = \mathbf{Fd}$.

Value

A RasterLayer of exposure values

See Also

departure, sensitivity_map, vulnerability_map

Examples

```
dep <- departure(x = climdat.hist, y = climdat.fut, s.dat = ABPR)
exp.map <- exposure_map(dep)
```

GLcenfa *Climate-niche factor analysis for reference study area*

Description

This function is used to facilitate comparisons between species in the same study area. It speeds up the computation of multiple CNFAs or ENFAs by calculating the global covariance matrix as a first step, which can then be fed into the cnfa or enfa functions as their first argument. This saves the user from having to calculate the global covariance matrix for each species, which can take quite a bit of time.

Usage

```
GLcenfa(
  x,
  center = TRUE,
  scale = TRUE,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE,
  ...
)

## S4 method for signature 'Raster'
GLcenfa(
```

```
  x,
  center = TRUE,
  scale = TRUE,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE,
  ...
)
```

Arguments

x	Raster* object, typically a brick or stack of p environmental raster layers
center	logical or numeric. If TRUE, centering is done by subtracting the layer means (omitting NAs), and if FALSE, no centering is done. If center is a numeric vector with length equal to the nlayers(x), then each layer of x has the corresponding value from center subtracted from it
scale	logical or numeric. If TRUE, scaling is done by dividing the (centered) layers of x by their standard deviations if center is TRUE, and the root mean square otherwise. If scale is FALSE, no scaling is done. If scale is a numeric vector with length equal to nlayers(x), each layer of x is divided by the corresponding value. Scaling is done after centering
filename	character. Optional filename to save the RasterBrick output to file. If this is not provided, a temporary file will be created for large x
progress	logical. If TRUE, messages and progress bar will be printed
parallel	logical. If TRUE then multiple cores are utilized
n	numeric. Number of CPU cores to utilize for parallel processing
cl	optional cluster object
keep.open	logical. If TRUE and parallel = TRUE, the cluster object will not be closed after the function has finished
...	Additional arguments for writeRaster

Details

If there is too much correlation between the layers of x, the covariance matrix will be singular, which will lead to later problems in computing the overall marginalities, sensitivities, or specializations of species. In this case, a warning will be issued, suggesting the removal of correlated variables or a transformation of the data.

Value

Returns an S4 object of class GLcenfa with the following components:

global_ras Raster* x of p layers, possibly centered and scaled

cov Global p x p covariance matrix

See Also

cnfa, enfa

Examples

```
glc <- GLcenfa(x = climdat.hist)
```

GLcenfa-class *GLcenfa-class*

Description

An S4 object of class GLcenfa is created using the GLcenfa function on a Raster* object. It is best used for making comparisons between species in the same study area. It speeds up the computation of multiple CNFAs or ENFAs by calculating the global covariance matrix as a first step, which can then be fed into the cnfa or enfa functions as their first argument. This saves the user from having to calculate the global covariance matrix for each species, which can take quite a bit of time.

Slots

global_ras Raster* object x with p layers

cov matrix. Global p x p covariance matrix

GLdeparture *Climatic departure of reference study area*

Description

This function is used to facilitate comparisons between species in the same study area. It speeds up the computation of multiple departures by calculating the global covariance matrix as a first step, which can then be fed into the departure function as a first argument. This saves the user from having to calculate the global covariance matrix for each species, which can take quite a bit of time.

Usage

```
GLdeparture(
  x,
  y,
  center = TRUE,
  scale = TRUE,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
```

```
  cl = NULL,
  keep.open = FALSE,
  ...
)

## S4 method for signature 'Raster,Raster'
GLdeparture(
  x,
  y,
  center = TRUE,
  scale = TRUE,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE,
  ...
)

## S4 method for signature 'Raster,missing'
GLdeparture(
  x,
  y,
  center = TRUE,
  scale = TRUE,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE,
  ...
)
```

Arguments

`x`	Raster* object of p historical climate layers
`y`	Raster* object of p future climate layers, with the same names as `x`
`center`	logical or numeric. If `TRUE`, centering is done by subtracting the layer means (omitting NAs), and if `FALSE`, no centering is done. If `center` is a numeric vector with length equal to the `nlayers(x)`, then each layer of `x` has the corresponding value from center subtracted from it
`scale`	logical or numeric. If `TRUE`, scaling is done by dividing the (centered) layers of `x` by their standard deviations if center is `TRUE`, and the root mean square otherwise. If scale is `FALSE`, no scaling is done. If scale is a numeric vector with length equal to `nlayers(x)`, each layer of `x` is divided by the corresponding value. Scaling is done after centering

`filename`	character. Optional filename to save the RasterBrick output to file. If this is not provided, a temporary file will be created for large `x`
`progress`	logical. If `TRUE`, messages and progress bar will be printed
`parallel`	logical. If `TRUE` then multiple cores are utilized
`n`	numeric. Number of CPU cores to utilize for parallel processing
`cl`	optional cluster object
`keep.open`	logical. If `TRUE` and `parallel = TRUE`, the cluster object will not be closed after the function has finished
`...`	Additional arguments for `writeRaster`

Details

If there is too much correlation between the layers of `x`, the covariance matrix will be singular, which will lead to later problems in computing the overall departures of species. In this case, a warning will be issued, suggesting the removal of correlated variables or a transformation of the data.

Value

Returns an S4 object of class `GLcenfa` with the following components:

global_difras Raster* `x` of p layers, possibly centered and scaled

cov Global p x p covariance matrix

See Also

`departure`

Examples

```
gld <- GLdeparture(x = climdat.hist, y = climdat.fut)
```

`GLdeparture-class` *GLdeparture-class*

Description

An object of class `GLdeparture` is created by the `GLdeparture` function. It is best used for making comparisons between species in the same study area. It speeds up the computation of multiple departures by calculating the global covariance matrix as a first step, which can then be fed into the `departure` function as a first argument. This saves the user from having to calculate the global covariance matrix for each species, which can take quite a bit of time.

Slots

global_difras Raster* object of absolute differences between historical x and future y climate values

cov matrix. Global covariance matrix

---

parCov *Efficient calculation of covariance matrices for Raster* objects*

Description

parCov efficiently calculates the covariance of Raster* objects, taking advantage of parallel processing and pulling data into memory only as necessary. For large datasets with lots of variables, calculating the covariance matrix rapidly becomes unwieldy, as the number of calculations required grows quadratically with the number of variables.

Usage

```
parCov(x, y, ...)

## S4 method for signature 'Raster,missing'
parCov(
  x,
  w = NULL,
  sample = TRUE,
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE
)

## S4 method for signature 'Raster,Raster'
parCov(
  x,
  y,
  w = NULL,
  sample = TRUE,
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE
)
```

Arguments

x	Raster* object, typically a brick or stack
y	NULL (default) or a Raster* object with the same extent and resolution as x
...	additional arguments, including any of the following:
w	optional Raster* object of weights for a weighted covariance matrix
sample	logical. If TRUE, the sample covariance is calculated with a denominator of $n-1$
progress	logical. If TRUE, messages and progress bar will be printed
parallel	logical. If TRUE then multiple cores are utilized
n	numeric. Number of CPU cores to utilize for parallel processing
cl	optional cluster object
keep.open	logical. If TRUE and parallel = TRUE, the cluster object will not be closed after the function has finished

Details

This function is designed to work similarly to the cov and the layerStats functions, with two major differences. First, parCov allows you to calculate the covariance between two different Raster* objects, whereas layerStats does not. Second, parCov can (optionally) compute each element of the covariance matrix in parallel, offering a dramatic improvement in computation time for large Raster* objects.

The raster layer of weights w should contain raw weights as values, and should *not* be normalized so that sum(w) = 1. This is necessary for computing the sample covariance, whose formula contains sum(w) -1 in its denominator.

Value

Returns a matrix with the same row and column names as the layers of x. If y is supplied, then the covariances between the layers of x and the layers of codey are computed.

See Also

cov, layerStats

Examples

```
mat1 <- parCov(climdat.hist)

# correlation matrix
Z <- parScale(climdat.hist)
mat2 <- parCov(Z)

# covariance between two Raster* objects
mat3 <- parCov(x = climdat.hist, y = climdat.fut)
```

parScale *Efficient scaling of Raster* objects*

Description

parScale expands the raster::scale function to allow for faster parallel processing, scaling each layer of x in parallel.

Usage

```
parScale(x, ...)

## S4 method for signature 'Raster'
parScale(
  x,
  center = TRUE,
  scale = TRUE,
  filename = "",
  progress = FALSE,
  parallel = FALSE,
  n = 1,
  cl = NULL,
  keep.open = FALSE,
  ...
)
```

Arguments

x	Raster* object
...	Additional arguments for writeRaster
center	logical or numeric. If TRUE, centering is done by subtracting the layer means (omitting NAs), and if FALSE, no centering is done. If center is a numeric vector with length equal to the nlayers(x), then each layer of x has the corresponding value from center subtracted from it
scale	logical or numeric. If TRUE, scaling is done by dividing the (centered) layers of x by their standard deviations if center is TRUE, and the root mean square otherwise. If scale is FALSE, no scaling is done. If scale is a numeric vector with length equal to nlayers(x), each layer of x is divided by the corresponding value. Scaling is done after centering
filename	character. Optional filename to save the Raster* output to file. If this is not provided, a temporary file will be created for large x
progress	logical. If TRUE, messages and progress bar will be printed
parallel	logical. If TRUE then multiple cores are utilized
n	numeric. Number of CPU cores to utilize for parallel processing
cl	optional cluster object

`keep.open`	logical. If `TRUE` and `parallel` = `TRUE`, the cluster object will not be closed after the function has finished

Value

Raster* object

See Also

`scale`, `scale`

Examples

```
ch.scale <- parScale(x = climdat.hist)
```

`predict` *Predict methods*

Description

Make a RasterLayer with predictions from a fitted model object.

Usage

```
## S4 method for signature 'cnfa'
predict(object, newdata, filename = "", parallel = FALSE, n = 1, ...)

## S4 method for signature 'enfa'
predict(object, newdata, filename = "", parallel = FALSE, n = 1, ...)

## S4 method for signature 'departure'
predict(object, filename = "", parallel = FALSE, n = 1, ...)

## S4 method for signature 'vulnerability'
predict(object, newdata, filename = "", parallel = FALSE, n = 1, ...)
```

Arguments

`object`	model object
`newdata`	optional new data
`filename`	character. Optional filename to save the RasterBrick output to file. If this is not provided, a temporary file will be created for large `x`
`parallel`	logical. If `TRUE` then multiple cores are utilized
`n`	numeric. Number of CPU cores to utilize for parallel processing
`...`	Additional arguments for `writeRaster`

Value

Returns a RasterLayer of sensitivity, departure, or vulnerability predictions, depending on the class of object.

---

scatter *Biplots of* `cnfa` *and* `enfa` *objects.*

Description

Biplots of `cnfa` and `enfa` objects.

Usage

```
scatter(x, y, xax = 1, yax = 2, p = 0.99, n = 5, ...)

## S4 method for signature 'cnfa,GLcenfa'
scatter(x, y, xax = 1, yax = 2, p = 0.99, n = 5, ...)

## S4 method for signature 'enfa,GLcenfa'
scatter(x, y, xax = 1, yax = 2, p = 0.99, n = 5, ...)
```

Arguments

`x`	an object of class `cnfa` or `enfa` describing the occupied habitat
`y`	an object of class `GLcenfa` describing the global reference habitat
`xax`	the column number of the x-axis
`yax`	the column number of the y-axis
`p`	the proportion of observations to include in the calculations of the minimum convex polygons
`n`	the number of projected variables to label
`...`	additional `plot` arguments

Value

Returns a biplot of the `cnfa` or `enfa` object.

See Also

`biplot`

Examples

```
mod <- cnfa(x = climdat.hist, s.dat = ABPR, field = "CODE")
glc <- GLcenfa(x = climdat.hist)
scatter(x = mod, y = glc)
```

sensitivity_map — *Create a sensitivity map*

Description

Creates a sensitivity map of species habitat from a cnfa object.

Usage

```
sensitivity_map(cnfa, parallel = FALSE, n = 1, filename = "", ...)
```

Arguments

cnfa	Object of class cnfa
parallel	logical. If TRUE then multiple cores are utilized
n	numeric. Number of cores to use for calculation
filename	character. Output filename (optional)
...	Additional arguments for file writing as for writeRaster

Details

The values of the sensitivity raster are calculated by centering the habitat's climate data around the marginality factor **m** and projecting onto the sensitivity factor **s**, given by the formula

$\sigma = |\mathbf{S} - \mathbf{m}|\mathbf{s}$.

Value

A RasterLayer of sensitivity values

See Also

cnfa, exposure_map, vulnerability_map

Examples

```
mod1 <- cnfa(x = climdat.hist, s.dat = ABPR, field = "CODE")
sens.map <- sensitivity_map(mod1)
```

slot-access *Accessing CENFA slots*

Description

Functions for extracting data from slots of objects of classes `cnfa` and `enfa`.

Usage

```
m.factor(x)

s.factor(x)

marginality(x)

specialization(x)

sensitivity(x)

cov.cnfa(x)

cov.enfa(x)

cov.GLcenfa(x)

## S4 method for signature 'enfa'
raster(x)

## S4 method for signature 'cnfa'
raster(x)

## S4 method for signature 'GLcenfa'
raster(x)

## S4 method for signature 'GLdeparture'
raster(x)

## S4 method for signature 'GLcenfa'
names(x)

## S4 method for signature 'GLdeparture'
names(x)

## S4 method for signature 'cnfa'
names(x)

## S4 method for signature 'enfa'
```

```
names(x)

## S4 method for signature 'departure'
names(x)
```

Arguments

x	cnfa or enfa object

Value

Object stored in slot.

Examples

```
mod1 <- cnfa(x = climdat.hist, s.dat = ABPR, field = "CODE")
m.factor(mod1)
```

---

stretchPlot — *Contrast adjustments for RasterLayer plots*

Description

A plotting function that provides methods for improving the contrast between values.

Usage

```
stretchPlot(x, type = "linear", n, ...)

## S4 method for signature 'RasterLayer'
stretchPlot(x, type = "linear", n, ...)
```

Arguments

x	a RasterLayer
type	character. Possible values are "linear", "hist.equal", and "sd"
n	number of standard deviations to include if type = "sd"
...	Additional arguments for raster::plot

Details

If type = "hist.equal", a histogram equalization procedure will be applied to the values of x. If type = "sd", the values of x will be scaled between values that fall between n standard deviations of the mean.

Value

Returns a RasterLayer plot.

Examples

```
mod <- enfa(x = climdat.hist, s.dat = ABPR, field = "CODE")
sm <- sensitivity_map(mod)
stretchPlot(sm)
stretchPlot(sm, type = "hist.equal")
stretchPlot(sm, type = "sd", n = 2)
```

---

tree-data *Tree distribution data*

Description

Some example datasets of historical tree distributions, from "Atlas of United States Trees" by Elbert L. Little, Jr.

Format

SpatialPolygonDataFrame

Details

ABPR Historical range map for the noble fir (*Abies procera*)

QUGA Historical range map for the Oregon white oak (*Quercus garryana*)

SESE Historical range map for the coast redwood (*Sequoia sempervirens*)

Source

https://www.usgs.gov/

See Also

climdat.hist, climdat.fut

vulnerability *Climatic vulnerability*

Description

Calculates the climatic vulnerability of a species using a `cnfa` and `departure` object.

Usage

```
vulnerability(
  cnfa,
  dep,
  method = "geometric",
  w,
  parallel = FALSE,
  n = 1,
  filename = "",
  ...
)

## S4 method for signature 'cnfa,departure'
vulnerability(
  cnfa,
  dep,
  method = "geometric",
  w,
  parallel = FALSE,
  n = 1,
  filename = "",
  ...
)
```

Arguments

`cnfa`	Object of class `cnfa`
`dep`	Object of class `departure`
`method`	character. What type of mean should be used to combine sensitivity and exposure. Choices are "arithmetic" and "geometric"
`w`	numeric. Optional vector of length two specifying the relative weights of sensitivity and exposure. See Details
`parallel`	logical. If `TRUE` then multiple cores are utilized
`n`	numeric. Number of cores to use for calculation
`filename`	character. Output filename (optional)
`...`	Additional arguments for file writing as for `writeRaster`

Details

The values of the vulnerability raster are calculated by combining the sensitivity σ and the exposure ϵ. If `method = "arithmetic"`, they will be combined as

$\nu = (w_1\sigma + w_2\epsilon)/(\sum_i w_i)$.

If `method = "geometric"`, they will be combined as

$\nu = \sqrt{(\sigma * \epsilon)}$.

Value

Returns an S4 object of class `vulnerability` with the following slots:

call Original function call

vf Vulnerability factor. Vector of length p that describes the amount of vulnerability in each climate variable

vulnerability Magnitude of the vulnerability factor

ras RasterLayer of climate vulnerability

weights Raster layer of weights used for departure calculation

References

Rinnan, D. Scott and Lawler, Joshua. Climate-niche factor analysis: a spatial approach to quantifying species vulnerability to climate change. Ecography (2019): <doi:10.1111/ecog.03937>.

See Also

`departure`

Examples

```
mod1 <- cnfa(x = climdat.hist, s.dat = ABPR, field = "CODE")
dep <- departure(x = climdat.hist, y = climdat.fut, s.dat = ABPR)
vuln <- vulnerability(cnfa = mod1, dep = dep)
```

`vulnerability-class` *vulnerability-class*

Description

An object of class `vulnerability` is created from a `cnfa` object and a `dep` object.

Slots

- `call` Original function call
- `vf` vulnerability factor
- `vulnerability` Magnitude of the vulnerability factor
- `ras` RasterLayer of vulnerability values
- `weights` Raster layer of weights used for departure calculation

`vulnerability_map` *Create a vulnerability map*

Description

Extracts a vulnerability map of species habitat from a `vulnerability` object.

Usage

```
vulnerability_map(vuln)
```

Arguments

`vuln`	Object of class `vulnerability`

Details

Note: this is only a convenience function. The `vulnerability` function creates a vulnerability map, and `vulnerability_map` simply extracts it. This is included for consistency with the `sensitivity_map` and `departure_map` functions.

Value

RasterLayer of vulnerability values

See Also

`vulnerability`, `sensitivity_map`, `exposure_map`

Examples

```
mod1 <- cnfa(x = climdat.hist, s.dat = ABPR, field = "CODE")
dep <- departure(x = climdat.hist, y = climdat.fut, s.dat = ABPR)
vuln <- vulnerability(cnfa = mod1, dep = dep)
vuln.map <- vulnerability_map(vuln)
```

Index