Package ‘nichevol’

July 22, 2025

Type Package

Title Tools for Ecological Niche Evolution Assessment Considering Uncertainty

Version 0.1.20

Author Marlon E. Cobos [aut, cre],
Hannah L. Owens [aut],
A. Townsend Peterson [aut]

Maintainer Marlon E. Cobos `<manubio13@gmail.com>`

Date 2023-03-09

Description A collection of tools that allow users to perform critical steps in the process of assessing ecological niche evolution over phylogenies, with uncertainty incorporated explicitly in reconstructions. The method proposed here for ancestral reconstruction of ecological niches characterizes species' niches using a bin-based approach that incorporates uncertainty in estimations. Compared to other existing methods, the approaches presented here reduce risk of overestimation of amounts and rates of ecological niche evolution. The main analyses include: initial exploration of environmental data in occurrence records and accessible areas, preparation of data for phylogenetic analyses, executing comparative phylogenetic analyses of ecological niches, and plotting for interpretations. Details on the theoretical background and methods used can be found in: Owens et al. (2020) <doi:10.1002/ece3.6359>, Peterson et al. (1999) <doi:10.1126/science.285.5431.1265>, Soberón and Peterson (2005) <doi:10.17161/bi.v2i0.4>, Peterson (2011) <doi:10.1111/j.1365-2699.2010.02456.x>, Barve et al. (2011) <doi:10.1111/ecog.02671>, Machado-Stredel et al. (2021) <doi:10.21425/F5FBG48814>, Owens et al. (2013) <doi:10.1016/j.ecolmodel.2013.04.011>, Saupe et al. (2018) <doi:10.1093/sysbio/syx084>, and Cobos et al. (2021) <doi:10.1111/jav.02868>.

URL `https://github.com/marlonecobos/nichevol`

BugReports `https://github.com/marlonecobos/nichevol/issues`

Imports castor (>= 1.4), geiger (>= 2.0), graphics (>= 3.6), grDevices (>= 3.6), stringr (>= 1.4), stats (>= 3.6), terra (>= 1.6), utils (>= 3.6)

Suggests knitr(>= 1.25), phytools (>= 0.6)

Depends ape (>= 5.3), R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

NeedsCompilation no

Repository CRAN

Date/Publication 2023-03-10 23:20:02 UTC

Contents

`bin_env` *Helper function to prepare bin tables*

Description

Helper function to prepare bin tables

Usage

```
bin_env(overall_range, M_range, sp_range, bin_size)
```

Arguments

`overall_range`	(numeric) minimum and maximum values of all species and Ms to be analyzed.
`M_range`	matrix of ranges of environmental values in M for all species. Columns must be minimum and maximum, and rows correspond to species.
`sp_range`	matrix of ranges of environmental values in occurrences for all species. Columns must be minimum and maximum, and rows correspond to species.
`bin_size`	(numeric) size of bins. Range of environmental values to be considered when creating each character in bin tables. See details.

Details

The argument `bin_size` helps to create characters that represent not only one value of an environmental variable, but a range of environmental conditions. For instance, if a variable of precipitation in mm is used, a value of 10 for `bin_size` indicates that each character will represent a class that correspond to 10 continuous values of precipitation (e.g., from 100 to 110 mm).

Value

A character matrix (table of characters) containing bins for a given variable and for all species considered. See more details in `bin_tables`.

Examples

```
# example
o_range <- c(1, 25)
m_range <- rbind(c(5, 15), c(10, 23), c(4, 20))
s_range <- rbind(c(7, 15), c(12, 21), c(3, 18))

# bin preparation
bins <- bin_env(overall_range = o_range, M_range = m_range,
                sp_range = s_range, bin_size = 1)
```

`bin_ml_rec` *Maximum likelihood reconstruction of ancestral character states*

Description

Maximum likelihood reconstruction of ancestral character states

Usage

```
bin_ml_rec(tree_data, ...)
```

Arguments

`tree_data`	a list of two elements (phy and data) resulting from using the function `treedata`.
`...`	other arguments from `ace`. Arguments `x`, `phy`, `type`, and `method` are fixed.

Details

Reconstructions are done using the function `ace` from the ape package. The argument method is set as "ML" and the type of variable is "discrete".

Value

A table with columns representing bins, rows representing first tip states and then reconstructed nodes.

Examples

```
# a simple tree
data("tree5", package = "nichevol")

# a matrix of niche charactes (1 = present, 0 = absent, ? = unknown)
dataTable <- cbind("241" = rep("1", length(tree5$tip.label)),
                   "242" = rep("1", length(tree5$tip.label)),
                   "243" = c("1", "1", "0", "0", "0"),
                   "244" = c("1", "1", "0", "0", "0"),
                   "245" = c("1", "?", "0", "0", "0"))
rownames(dataTable) <- tree5$tip.label

# list with two objects (tree and character table)
treeWdata <- geiger::treedata(tree5, dataTable)

# Maximum likelihood reconstruction
ml_rec <- bin_ml_rec(treeWdata)
```

`bin_par_rec` *Maximum parsimony reconstruction of ancestral character states*

Description

Maximum parsimony reconstruction of ancestral character states

Usage

```
bin_par_rec(tree_data, ...)
```

Arguments

`tree_data`	a list of two elements (phy and data) resulting from using the function `treedata`.
`...`	other arguments from `asr_max_parsimony`. Arguments `tree` and `tip_states` are fixed.

Details

Reconstructions are done using the `asr_max_parsimony` function from the `castor` package.

Value

A table with columns representing bins, rows representing first tip states and then reconstructed nodes.

Examples

```
# a simple tree
data("tree5", package = "nichevol")

# a matrix of niche charactes (1 = present, 0 = absent, ? = unknown)
dataTable <- cbind("241" = rep("1", length(tree5$tip.label)),
                   "242" = rep("1", length(tree5$tip.label)),
                   "243" = c("1", "1", "0", "0", "0"),
                   "244" = c("1", "1", "0", "0", "0"),
                   "245" = c("1", "?", "0", "0", "0"))
rownames(dataTable) <- tree5$tip.label

# list with two objects (tree and character table)
treeWdata <- geiger::treedata(tree5, dataTable)

# Maximum parsimony reconstruction
par_rec <- bin_par_rec(treeWdata)
```

`bin_table` *Bin table of environmental conditions in M and for occurrences*

Description

bin_table helps in creating a bin table of environmental conditions in accessible areas (M) and for species occurrence records (i.e., table of characters).

Usage

```
bin_table(Ms, occurrences, species, longitude, latitude, variable,
          percentage_out = 5, n_bins = 20, bin_size, verbose = TRUE)
```

Arguments

`Ms`	a list of SpatVector objects representing the accessible area (M) for all species to be analyzed. The order of species represented by each object here must coincide with the one in `occurrences`. See details.
`occurrences`	a list of data.frames of occurrence records for all species. The order of species represented by each data.frame must coincide with the one in `Ms`. See details.
`species`	(character) name of the column in occurrence data.frames that contains the name of the species.
`longitude`	(character) name of the column in occurrence files containing values of longitude.
`latitude`	(character) name of the column in occurrence files containing values of latitude.
`variable`	a single SpatRaster layer representing an environmental variable of interest. See details.
`percentage_out`	(numeric) percentage of extreme environmental data in M to be excluded in bin creation for further analyses. See details. Default = 5.
`n_bins`	(numeric) number of bins to be created from the range of environmental values considered when creating each character in bin tables. Default = 20. See details.
`bin_size`	(numeric) argument deprecated, use n_bins instead.
`verbose`	(logical) whether messages should be printed. Default = TRUE.

Details

Coordinates in `occurrences`, SpatVector objects in `Ms`, and SpatRaster in `variable` must coincide in the geographic projection in which they are represented. WGS84 with no planar projection is recommended.

Accessible area (M) is understood as the geographic area that has been accessible for a species for relevant periods of time. Defining M is usually a hard task, but also a very important one, because it allows identifying uncertainties about the ability of a species to maintain populations in certain environmental conditions. For further details on this topic, see Barve et al. (2011) `doi:10.1016/j.ecolmodel.2011.02.011` and Machado-Stredel et al. (2021) `doi:10.21425/F5FBG48814`.

The percentage to be defined in `percentage_out` excludes a percentage of extreme environmental values to prevent from considering extremely rare environmental values in the accessible area for the species (M). Being too rare, these values may have never been explored by the species; therefore, including them in the process of preparation of the table of characters (bin table) is risky.

The argument `n_bins` helps to define how many characters (bins) will be considered for the range of values in each variable. This is, a value of 20 determines that a range of temperature (5-25) will be split approximately every 1 degree. The argument `bin_size` has been deprecated.

Value

A list containing a table of characters to represent ecological niches of the species of interest.

Potential values for characters are:

- "1" = the species is present in those environmental conditions.
- "0" = the species is not present in those environmental conditions. This is, those environmental conditions inside the accessible area (M) are more extreme than the ones used for the species.
- "?" = there is no certainty about the species presence in those environmental conditions. This happens in environmental combinations more extreme than the ones found in the accessible area (M), when environmental conditions in species records are as extreme as the most extreme ones in M.

Examples

```
# example data
## list of species records
data("occ_list", package = "nichevol")

## list of species accessible areas
m_files <- list.files(system.file("extdata", package = "nichevol"),
                      pattern = "m\\d.gpkg", full.names = TRUE)

m_list <- lapply(m_files, terra::vect)

## raster variable
temp <- terra::rast(system.file("extdata", "temp.tif", package = "nichevol"))


# preparing bins
char_table <- bin_table(Ms = m_list, occurrences = occ_list, species = "species",
                        longitude = "x", latitude = "y", variable = temp,
                        percentage_out = 5, n_bins = 20)
```

`bin_tables` *Bin tables of environmental conditions in M and for occurrences from objects*

Description

bin_tables helps in creating bin tables of environmental conditions in accessible areas (M) and species occurrence records (i.e., table of characters). This is done using results from previous analyses, and can be applied to various species and multiple variables.

Usage

```
bin_tables(ranges, percentage_out = 5, n_bins = 20, bin_size, save = FALSE,
           output_directory, overwrite = FALSE, verbose = TRUE)
```

Arguments

`ranges`	list of ranges of environmental values in M and in species occurrences derived from using the function `histograms_env`.
`percentage_out`	(numeric) percentage of extreme environmental data in M to be excluded in bin creation for further analyses. See details. Default = 5.
`n_bins`	(numeric) number of bins to be created from the range of environmental values considered when creating each character in bin tables. Default = 20. See details.
`bin_size`	(numeric) argument deprecated, use n_bins instead.
`save`	(logical) whether or not to save the results in working directory. Default = FALSE.
`output_directory`	(character) name of the folder in which results will be written.
`overwrite`	(logical) whether or not to overwrite existing results in `output_directory`. Default = FALSE.
`verbose`	(logical) whether messages should be printed. Default = TRUE.

Details

The percentage to be defined in `percentage_out` must correspond with one of the confidence limits defined in `histograms_env` (argument `CL_lines`). For instance, if `CL_lines` = 95, then `percentage_out` can only be either 5 (keeping data inside the 95 CL) or 0 (to avoid exclusion of extreme values in M).

Excluding a certain percentage of extreme environmental values prevents the algorithm from considering extremely rare environmental values in the accessible area for the species (M). Being too rare, these values may have never been explored by the species; therefore, including them in the process of preparation of the table of characters (bin table) is risky.

The argument `n_bins` helps to define how many characters (bins) will be considered for the range of values in each variable. This is, a value of 20 determines that a range of temperature (5-25) will be split approximately every 1 degree. The argument `bin_size` has been deprecated.

Value

A list named as in `ranges` containing the table(s) of characters. A folder named as in `output_directory` containing all resulting csv files with the tables of characters will be created if `save` is set as TRUE.

Potential values for characters are:

- "1" = the species is present in those environmental conditions.
- "0" = the species is not present in those environmental conditions. This is, those environmental conditions inside the accessible area (M) are more extreme than the ones used for the species.
- "?" = there is no certainty about the species presence in those environmental conditions. This happens if environmental combinations are more extreme than the ones found in the accessible area (M), when environmental conditions in species records are as extreme as the most extreme ones in M.

Examples

```
# simple list of ranges
ranges <- list(temp = data.frame(Species = c("sp1", "sp2", "sp3"),
                                 Species_lower = c(120, 56, 59.75),
                                 Species_upper = c(265, 333, 333),
                                 M_lower = c(93, 39, 56),
                                 M_upper = c(302, 333, 333),
                                 M_95_lowerCL = c(158, 91, 143),
                                 M_95_upperCL = c(292, 290, 326)),
               prec = data.frame(Species = c("sp1", "sp2", "sp3"),
                                 Species_lower = c(597, 3, 3),
                                 Species_upper = c(3492, 2673, 6171),
                                 M_lower = c(228, 3, 3),
                                 M_upper = c(6369, 7290, 6606),
                                 M_95_lowerCL = c(228, 3, 3),
                                 M_95_upperCL = c(3114, 2376, 2568)))

# bin preparation
bins <- bin_tables(ranges, percentage_out = 5, n_bins = 20)

# see arguments save and output_directory to write results in local directory
```

`bin_tables0` *Bin tables of environmental conditions in M and for occurrences from data*

Description

bin_tables0 helps in creating bin tables of environmental conditions in accessible areas (M) and species occurrence records (i.e., table of characters). This is done using data read directly from a local directory, and can be applied to various species and multiple variables.

Usage

```
bin_tables0(M_folder, M_format, occ_folder, longitude,
            latitude, var_folder, var_format, round = FALSE,
            round_names, multiplication_factor = 1,
            percentage_out = 5, n_bins = 20, bin_size, save = FALSE,
            output_directory, overwrite = FALSE, verbose = TRUE)
```

Arguments

`M_folder`	(character) name of the folder containing files representing the accessible area (M) for all species to be analyzed. See details.
`M_format`	format of files representing the accessible area (M) for the species. Names of M files must match the ones for occurrence files in `occ_folder`. Format options are: "shp", "gpkg", or any of the options supported by `rast` (e.g., "tif" or "asc").
`occ_folder`	(character) name of the folder containing csv files of occurrence data for all species. Names of csv files must match the ones of M files in `M_folder`.
`longitude`	(character) name of the column in occurrence files containing values of longitude.
`latitude`	(character) name of the column in occurrence files containing values of latitude.
`var_folder`	(character) name of the folder containing layers to represent environmental variables.
`var_format`	format of layers to represent environmental variables. Format options are all the ones supported by `rast` (e.g., "tif" or "asc").
`round`	(logical) whether or not to round the values of one or more variables after multiplying them times the value in `multiplication_factor`. Default = FALSE. See details.
`round_names`	(character) names of the variables to be rounded. Default = NULL. If `round` = TRUE, names must be defined.
`multiplication_factor`	(numeric) value to be used to multiply the variables defined in `round_names`. Default = 1.
`percentage_out`	(numeric) percentage of extreme environmental data in M to be excluded in bin creation for further analyses. See details. Default = 5.
`n_bins`	(numeric) number of bins to be created from the range of environmental values considered when creating each character in bin tables. Default = 20. See details.
`bin_size`	(numeric) argument deprecated, use n_bins instead.
`save`	(logical) whether or not to save the results in working directory. Default = FALSE.
`output_directory`	(character) name of the folder in which results will be written.
`overwrite`	(logical) whether or not to overwrite existing results in `output_directory`. Default = FALSE.
`verbose`	(logical) whether messages should be printed. Default = TRUE.

Details

Coordinates in csv files in `occ_folder`, SpatVector-like files in `M_folder`, and raster layers in `var_folder` must coincide in the geographic projection in which they are represented. WGS84 with no planar projection is recommended.

Accessible area (M) is understood as the geographic area that has been accessible for a species for relevant periods of time. Defining M is usually a hard task, but also a very important one, because it allows identifying uncertainties about the ability of a species to maintain populations in certain environmental conditions. For further details on this topic, see Barve et al. (2011) `doi:10.1016/j.ecolmodel.2011.02.011` and Machado-Stredel et al. (2021) `doi:10.21425/F5FBG48814`.

Rounding variables may be useful when multiple variables are considered and the values of some or all of them are too small (e.g., when using principal components). To round specific variables arguments `round`, `round_names`, and `multiplication_factor`, must be used accordingly.

The percentage to be defined in `percentage_out` excludes a percentage of extreme environmental values to prevent from considering extremely rare environmental values in the accessible area for the species (M). Being too rare, these values may have never been explored by the species; therefore, including them in the process of preparation of the table of characters (bin table) is risky.

The argument `n_bins` helps to define how many characters (bins) will be considered for the range of values in each variable. This is, a value of 20 determines that a range of temperature (5-25) will be split approximately every 1 degree. The argument `bin_size` has been deprecated.

Value

A list named as the variables present in `var_folder`, containing all tables of characters. A folder named as in `output_directory` containing all resultant csv files with the tables of characters will be created if `save` is set as TRUE.

Potential values for characters are:

- "1" = the species is present in those environmental conditions.
- "0" = the species is not present in those environmental conditions. This is, those environmental conditions inside the accessible area (M) are more extreme than the ones used for the species.
- "?" = there is no certainty about the species presence in those environmental conditions. This happens in environmental combinations more extreme than the ones found in the accessible area (M), when environmental conditions in species records are as extreme as the most extreme ones in M.

Examples

```
# preparing data and directories for example
## directories
tempdir <- file.path(tempdir(), "nevol_test")
dir.create(tempdir)

cvariables <- paste0(tempdir, "/variables")
dir.create(cvariables)

records <- paste0(tempdir, "/records")
dir.create(records)

m_areas <- paste0(tempdir, "/M_areas")
dir.create(m_areas)

## data
data("occ_list", package = "nichevol")

temp <- system.file("extdata", "temp.tif", package = "nichevol")
```

```
m_files <- list.files(system.file("extdata", package = "nichevol"),
                      pattern = "m\\d.gpkg", full.names = TRUE)

## writing data in temporal directories
spnames <- sapply(occ_list, function (x) as.character(x[1, 1]))
ocnames <-  paste0(records, "/", spnames, ".csv")

occs <- lapply(1:length(spnames), function (x) {
  write.csv(occ_list[[x]], ocnames[x], row.names = FALSE)
})

to_replace <- paste0(system.file("extdata", package = "nichevol"), "/")

otemp <- gsub(to_replace, "", temp)
file.copy(from = temp, to = paste0(cvariables, "/", otemp))

file.copy(from = m_files, to = paste0(m_areas, "/", spnames, ".gpkg"))

# preparing tables
tabs <- bin_tables0(M_folder = m_areas, M_format = "gpkg", occ_folder = records,
                    longitude = "x", latitude = "y", var_folder = cvariables,
                    var_format = "tif")
```

`character_table` *Example of character table for six species*

Description

A character table representing species ecological niches derived from previous preparation processes. Each row represents a species and each column a binary character in which one or more values of the environmental variable are categorized as used "1", non used "0", or uncertain "?".

Usage

```
character_table
```

Format

A character matrix with 6 rows and 28 columns.

Examples

```
data("character_table", package = "nichevol")

head(character_table)
```

`histograms_env` *Histograms of environmental conditions in M and for occurrences*

Description

histograms_env creates PDF files with histogram plots of environmental conditions in M, lines for the confidence limits of values in M, and the location of values in occurrence records. This is done using data read directly from a local directory, and can be applied to various species and multiple variables.

Usage

```
histograms_env(M_folder, M_format, occ_folder, longitude, latitude,
               var_folder, var_format, CL_lines = c(95, 99), col = NULL,
               round = FALSE, round_names = NULL, multiplication_factor = 1,
               save_ranges = FALSE, output_directory, overwrite = FALSE,
               verbose = TRUE)
```

Arguments

`M_folder`	(character) name of the folder containing files representing the accessible area (M) for all species to be analyzed. See details.
`M_format`	format of files representing the accessible area (M) for the species. Names of M files must match the ones for occurrence files in `occ_folder`. Format options are: "shp", "gpkg", or any of the options supported by `rast` (e.g., "tif" or "asc").
`occ_folder`	(character) name of the folder containing csv files of occurrence data for all species. Names of csv files must match the ones of M files in `M_folder`.
`longitude`	(character) name of the column in occurrence files containing values of longitude.
`latitude`	(character) name of the column in occurrence files containing values of latitude.
`var_folder`	(character) name of the folder containing layers to represent environmental variables.
`var_format`	format of layers to represent environmental variables. Format options are all the ones supported by `rast` (e.g., "tif" or "asc").
`CL_lines`	(numeric) confidence limits of environmental values in M to be plotted as lines in the histograms. See details. Default = c(95, 99).
`col`	colors for lines representing confidence limits. If NULL, colors are selected from a gray palette. Default = NULL.
`round`	(logical) whether or not to round values of one or more variables after multiplying them times the value in `multiplication_factor`. Default = FALSE. See details.
`round_names`	(character) names of the variables to be rounded. Default = NULL. If `round` = TRUE, names must be defined.

`multiplication_factor`	(numeric) value to be used to multiply the variables defined in `round_names`. Default = 1.
`save_ranges`	(logical) whether or not to save the values identified as ranges considering the whole set of values and confidence limits defined in `CL_lines`. Default = FALSE.
`output_directory`	(character) name of the folder in which results will be written.
`overwrite`	(logical) whether or not to overwrite existing results in `output_directory`. Default = FALSE.
`verbose`	(logical) whether messages should be printed. Default = TRUE.

Details

Coordinates in csv files in `occ_folder`, SpatVector-like files in `M_folder`, and raster layers in `var_folder` must coincide in the geographic projection in which they are represented. WGS84 with no planar projection is recommended.

Accessible area (M) is understood as the geographic area that has been accessible for a species for relevant periods of time. Defining M is usually a hard task, but also a very important one, because it allows identifying uncertainties about the ability of a species to maintain populations under certain environmental conditions. For further details on this topic, see Barve et al. (2011) `doi:10.1016/j.ecolmodel.2011.02.011` and Machado-Stredel et al. (2021) `doi:10.21425/F5FBG48814`.

Rounding variables may be useful when multiple variables are considered and the values of some or all of them are too small (e.g., when using principal components). To round specific variables arguments `round`, `round_names`, and `multiplication_factor`, must be used accordingly.

Value

A list of data.frames containing intervals of environmental values in species occurrences and accessible areas (M), as well as values corresponding to the confidence limits defined in `CL_lines`. A folder named as in `output_directory` containing all resulting PDF files (one per variable) with histograms for all species. Files (csv) of ranges found during the analyses will be also written in `output_directory` if `save_ranges` is set as TRUE.

Examples

```
# preparing data and directories for examples
## directories
tempdir <- file.path(tempdir(), "nevol_test")
dir.create(tempdir)

cvariables <- paste0(tempdir, "/variables")
dir.create(cvariables)

records <- paste0(tempdir, "/records")
dir.create(records)

m_areas <- paste0(tempdir, "/M_areas")
dir.create(m_areas)
```

```
histdir <- paste0(tempdir, "/Hists")

## data
data("occ_list", package = "nichevol")

temp <- system.file("extdata", "temp.tif", package = "nichevol")

m_files <- list.files(system.file("extdata", package = "nichevol"),
                      pattern = "m\\d.gpkg", full.names = TRUE)

## writing data in temporal directories
spnames <- sapply(occ_list, function (x) as.character(x[1, 1]))
ocnames <-  paste0(records, "/", spnames, ".csv")

occs <- lapply(1:length(spnames), function (x) {
  write.csv(occ_list[[x]], ocnames[x], row.names = FALSE)
})

to_replace <- paste0(system.file("extdata", package = "nichevol"), "/")

otemp <- gsub(to_replace, "", temp)
file.copy(from = temp, to = paste0(cvariables, "/", otemp))

file.copy(from = m_files, to = paste0(m_areas, "/", spnames, ".gpkg"))

# running analysis to produce plots
hists <- histograms_env(M_folder = m_areas, M_format = "gpkg",
                        occ_folder = records, longitude = "x",
                        latitude = "y", var_folder = cvariables,
                        var_format = "tif", output_directory = histdir)
```

`hist_evalues` *Histograms of environmental conditions in M and for occurrences (one species)*

Description

hist_evalues helps in creating histograms to explore environmental conditions in M, lines for the confidence limits of values in M, and the location of values in occurrence records, for one species at the time.

Usage

```
hist_evalues(M, occurrences, species, longitude, latitude, variable,
             CL_lines = c(95, 99), col = NULL)
```

Arguments

`M`	a SpatVector object representing the accessible area (M) for one species. See details.

`occurrences`	a data.frame of occurrence records for one species. See details.
`species`	(character) name of the column in `occurrences` that contains the name of the species.
`longitude`	(character) name of the column in `occurrences` containing values of longitude.
`latitude`	(character) name of the column in `occurrences` containing values of latitude.
`variable`	a single SpatRaster layer representing an environmental variable of interest. See details.
`CL_lines`	(numeric) confidence limits of environmental values in M to be plotted as lines in the histograms. See details. Default = c(95, 99).
`col`	colors for lines representing confidence limits. If NULL, colors are selected from a gray palette. Default = NULL.

Details

Coordinates in `occurrences`, SpatVector object in `M`, and SpatRaster in `variable` must coincide in the geographic projection in which they are represented. WGS84 with no planar projection is recommended.

The accessible area (M) is understood as the geographic area that has been accessible to a species over relevant periods of time. Defining M is usually a hard task, but also a very important one because it allows identifying uncertainties about the ability of a species to maintain populations under certain environmental conditions. For further details on this topic, see Barve et al. (2011) `doi:10.1016/j.ecolmodel.2011.02.011` and Machado-Stredel et al. (2021) `doi:10.21425/F5FBG48814`.

Examples

```
# example data
## list of species records
data("occ_list", package = "nichevol")

## list of species accessible areas
m_files <- list.files(system.file("extdata", package = "nichevol"),
                      pattern = "m\\d.gpkg", full.names = TRUE)

m_list <- lapply(m_files, terra::vect)

## raster variable
temp <- terra::rast(system.file("extdata", "temp.tif", package = "nichevol"))

# running stats
hist_evalues(M = m_list[[1]], occurrences = occ_list[[1]], species = "species",
             longitude = "x", latitude = "y", variable = temp,
             CL_lines = c(95, 99), col = c("blue", "red"))
```

`m` *Example of accessible areas for a species*

Description

A SpatVector object representing the accessible area for a species.

Format

A SpatVector object.

Value

No return value, used with function `vect` to bring an example of an accessible area for a species.

Examples

```
m1 <- terra::vect(system.file("extdata", "m1.gpkg", package = "nichevol"))

terra::plot(m1)
```

`map_nichevol` *Maps of niche reconstructions and changes detected*

Description

map_nichevol produces a SpatRaster layer representing geographic areas corresponding to environmental bins of niche or events of niche evolution detected in reconstructions.

Usage

```
map_nichevol(whole_rec_table, variable, return = "niche", from, to = NULL,
             id_unknown = TRUE, verbose = TRUE)
```

Arguments

`whole_rec_table`	matrix of environmental bins for all tips and nodes derived from functions `bin_par_rec` or `bin_ml_rec`.
`variable`	a SpatRaster layer corresponding to the variable for which the reconstruction was performed (represented in `whole_rec_table`).
`return`	(character) type of result to return. Options are: "niche", "evolution", or "nichevol" (a combination of both). Default = "niche". If "niche", values correspond to that defined in `from`. See Value.

`from`	(character) if `return` = "niche" tip or node for which layer will be prepared, otherwise, initial node from which niche comparison will be performed. See example.
`to`	(character) valid if `return` = "evolution" or "nichevol". Tip or node to compare against `from` to detected changes. Default = NULL. See example.
`id_unknown`	(logical) whether to identify areas of unknown or uncertain change. Default = TRUE. See details.
`verbose`	(logical) whether messages should be printed. Default = TRUE.

Details

Mapping is done following Cobos et al. (2021) `doi:10.1111/jav.02868`. This allows to represent geographic areas with environments where niche expanded, retracted, or stayed stable (evolution). Niche is represented as presence, absence, or unknown.

Defining `id_unknown` = TRUE allows to map areas where niche or niche change are uncertain. `id_unknown` = FALSE returns NA in areas with these characteristics, hence they will not be visible when plotting the resulting map.

Value

A SpatRaster object classified according to values of niche in `whole_rec_table`, and/or according to niche changes detected in comparisons between an ancestor and a tip, or another more recent ancestor.

Options of values resulting from classifications are as follow:

If `return` = "niche":

ID	category
0	Absent
10	Unknown
100	Present

If `return` = "evolution":

ID	category
0	Stable
1	Expansion low
3	Expansion high
2	Retraction high
4	Retraction low
10	Unknown

If `return` = "nichevol":

ID	category

0	Stable
1	Expansion low
3	Expansion high
10	Unknown
100	Present
102	Retraction high
104	Retraction low

Examples

```
# a tree
data("tree", package = "nichevol")

# raster variable
temp <- terra::rast(system.file("extdata", "temp.tif", package = "nichevol"))

# results from reconstruction
data("par_rec_table", package = "nichevol")

# rename tree tips
tree$tip.label <- rownames(par_rec_table)[1:6]

# check in plot
plot.phylo(tree, label.offset = 0.02)
nodelabels()
nichevol_labels(tree, par_rec_table)

# mapping nichevol
nevol_map <- map_nichevol(whole_rec_table = par_rec_table, variable = temp,
                          return = "nichevol", from = "9", to = "RD 6933")

terra::plot(nevol_map)
```

`nichevol` *nichevol: Assessment of Species' Ecological Niche Evolution Considering Uncertainty in Reconstructions*

Description

nichevol is a collection of tools that allow users to perform critical steps in the process of assessing ecological niche evolution over phylogenies, with uncertainty incorporated explicitly in reconstructions. The method proposed here for ancestral reconstruction of ecological niches characterizes species' niches using a bin-based approach that incorporates uncertainty in estimations. Compared to other existing methods, the approaches presented here reduce risk of overestimation of amounts and rates of ecological niche evolution. The main analyses include: initial exploration of environmental data in occurrence records and accessible areas, preparation of data for phylogenetic analyses, executing comparative phylogenetic analyses of ecological niches, and plotting for interpretations.

Main functions in nichevol

`bin_ml_rec`, `bin_par_rec`, `bin_table`, `bin_tables`, `bin_tables0`, `hist_evalues`, `histograms_env`, `map_nichevol`, `niche_bars`, `nichevol_bars`, `niche_labels`, `nichevol_labels`, `niche_legend`, `nichevol_legend`, `set_uncertainty`, `smooth_rec`, `stats_eval`, `stats_evalues`

Other functions (important helpers)

`bin_env`, `pdf_histograms`, `rename_tips`, `score_tip`, `score_tree`, `sig_sq`

`nichevol_bars` *PNG bar figures for representing niche evolution*

Description

nichevol_bars produces bar plots that represent how species' niches (considering one environmental variable at a time) have evolved. Bars are exported as png figures to an output directory for posterior use.

Usage

```
nichevol_bars(tree, whole_rec_table, ancestor_line = FALSE,
              present = "1", absent = "0", unknown = "?",
              present_col = "#252525", unknown_col = "#d9d9d9",
              no_change_col = "#b2df8a", retraction_col = "#984ea3",
              expansion_col = "#4daf4a", width = 50, height = 5,
              res = 300, output_directory, overwrite = FALSE)
```

Arguments

`tree`	an object of class "phylo".
`whole_rec_table`	matrix of reconstructed bins for nodes and species derived from a process of maximum parsimony reconstruction.
`ancestor_line`	controls whether ancestor line is plotted. Default = FALSE.
`present`	(character) code indicating environmental bins in which the species is present. Default = "1".
`absent`	(character) code indicating environmental bins in which the species is absent. Default = "0".
`unknown`	(character) code indicating environmental bins in which the species presence is unknown (uncertain). Default = "?".
`present_col`	color for line representing environments where the species is present. Default = "#252525".
`unknown_col`	color for line representing environments where the species presence is unknown (uncertain). Default = "#d9d9d9".
`no_change_col`	color for area of the bar representing environments where no change has been detected. Default = "#b2df8a".

`retraction_col`	color for area of the bar representing environments where niche retraction has been detected. Default = "#984ea3".
`expansion_col`	color for area of the bar representing environments where niche expansion has been detected. Default = "#4daf4a".
`width`	(numeric) width of the device in mm to be passed to the `png` function. Default = 50.
`height`	(numeric) height of the device in mm to be passed to the `png` function. Default = 5.
`res`	(numeric) nominal resolution in ppi to be passed to the `png` function. Default = 300.
`output_directory`	(character) name of the folder in which results will be written. The directory will be created as part of the process.
`overwrite`	(logical) whether or not to overwrite existing results in `output_directory`. Default = FALSE.

Details

Evolution of ecological niches is represented in one environmental dimension with horizontal bars indicating if the niche of the descendant has expanded, retracted, or has not changed compared to its ancestor. Lower values of environmental variables are represented in the left part of the bar, higher values at the right.

Changes in niches (evolution) are defined as follows:

- if (ancestor == present & descendant == absent) change <- "retraction"
- if (ancestor == present & descendant == present) change <- "no_change"
- if (ancestor == present & descendant == unknown) change <- "no_change"
- if (ancestor == absent & descendant == present) change <- "expansion"
- if (ancestor == absent & descendant == absent) change <- "no_change"
- if (ancestor == absent & descendant == unknown) change <- "no_change"
- if (ancestor == unknown & descendant == absent) change <- "no_change"
- if (ancestor == unknown & descendant == present) change <- "no_change"
- if (ancestor == unknown & descendant == unknown) change <- "no_change"

If `ancestor_line` is TRUE, the ancestor line will be plotted on the bar representing niche evolution. The line will represent where, in the range of environmental conditions, the ancestor was present, and where its presence is uncertain (unknown).

Value

A folder named as in `output_directory` containing all bar figures produced, as well as a legend to describe what is plotted.

Examples

```
# a simple tree
data("tree5", package = "nichevol")

# a matrix of niche charactes (1 = present, 0 = absent, ? = unknown)
dataTable <- cbind("241" = rep("1", length(tree5$tip.label)),
                   "242" = rep("1", length(tree5$tip.label)),
                   "243" = c("1", "1", "0", "0", "0"),
                   "244" = c("1", "1", "0", "0", "0"),
                   "245" = c("1", "?", "0", "0", "0"))
rownames(dataTable) <- tree5$tip.label

# list with two objects (tree and character table)
treeWdata <- geiger::treedata(tree5, dataTable)

# Maximum parsimony reconstruction
rec_tab <- smooth_rec(bin_par_rec(treeWdata))

# the running (before running, define a working directory)
nichevol_bars(tree5, rec_tab, output_directory = file.path(tempdir(), "evolbars"))
```

`nichevol_labels` *Labels to represent changes of niche characteristics between ancestors and descendants*

Description

nichevol_labels helps in adding bar-type labels that represent how species' niches changed from ancestors to descendants.

Usage

```
nichevol_labels(tree, whole_rec_table, ancestor_line = FALSE,
  present = "1", absent = "0", unknown = "?",
  present_col = "#252525", unknown_col = "#d9d9d9",
  no_change_col = "#b2df8a", retraction_col = "#984ea3",
  expansion_col = "#4daf4a", width = 1, height = 1)
```

Arguments

`tree`	an object of class "phylo".
`whole_rec_table`	matrix of reconstructed bins for nodes and species derived from a process of maximum parsimony or maximum likelihood reconstruction. See functions `bin_par_rec` or `bin_ml_rec`.
`ancestor_line`	controls whether ancestor line is plotted. Default = FALSE.
`present`	(character) code indicating environmental bins in which the species is present. Default = "1".

`absent`	(character) code indicating environmental bins in which the species is absent. Default = "0".
`unknown`	(character) code indicating environmental bins in which the species presence is unknown (uncertain). Default = "?".
`present_col`	color for line representing environments where the species is present. Default = "#252525".
`unknown_col`	color for line representing environments where the species presence is unknown (uncertain). Default = "#d9d9d9".
`no_change_col`	color for area of the bar representing environments where no change has been detected. Default = "#b2df8a".
`retraction_col`	color for area of the bar representing environments where niche retraction has been detected. Default = "#984ea3".
`expansion_col`	color for area of the bar representing environments where niche expansion has been detected. Default = "#4daf4a".
`width`	value defining the width of bars representing changes in niches; default = 1.
`height`	value defining the height of bars representing changes in niches; default = 1.

Details

For the moment, only plots of type "phylogram" with "rightwards" or "leftwards" directions, created with the function `plot.phylo` from the package ape are supported.

Evolution of ecological niches is represented in one environmental dimension, with vertical bars indicating if the niche of the descendant has expanded, retracted, or has not changed compared to its ancestor's niche. Lower values of environmental variables are represented in the lower part of the bar, and the opposite part of the bar represents higher values.

Changes in niches (evolution) are defined as follows:

- if (ancestor == present & descendant == absent) change <- "retraction"
- if (ancestor == present & descendant == present) change <- "no_change"
- if (ancestor == present & descendant == unknown) change <- "no_change"
- if (ancestor == absent & descendant == present) change <- "expansion"
- if (ancestor == absent & descendant == absent) change <- "no_change"
- if (ancestor == absent & descendant == unknown) change <- "no_change"
- if (ancestor == unknown & descendant == absent) change <- "no_change"
- if (ancestor == unknown & descendant == present) change <- "no_change"
- if (ancestor == unknown & descendant == unknown) change <- "no_change"

If `ancestor_line` is TRUE, the ancestor line will be plotted on the bar representing niche evolution. The line will represent where, in the range of environmental conditions, the ancestor was present, and where its presence is uncertain (unknown).

Examples

```
# a simple tree
data("tree5", package = "nichevol")

# a matrix of niche charactes (1 = present, 0 = absent, ? = unknown)
dataTable <- cbind("241" = rep("1", length(tree5$tip.label)),
                   "242" = rep("1", length(tree5$tip.label)),
                   "243" = c("1", "1", "0", "0", "0"),
                   "244" = c("1", "1", "0", "0", "0"),
                   "245" = c("1", "?", "0", "0", "0"))
rownames(dataTable) <- tree5$tip.label

# list with two objects (tree and character table)
treeWdata <- geiger::treedata(tree5, dataTable)

# Maximum parsimony reconstruction
rec_tab <- smooth_rec(bin_par_rec(treeWdata))

# plotting and adding labels
ape::plot.phylo(tree5, label.offset = 0.04)
nichevol_labels(tree5, rec_tab, height = 0.6)
```

`nichevol_legend` *Legends for niche evolution labels in phylogenetic trees*

Description

Legends for niche evolution labels in phylogenetic trees

Usage

```
nichevol_legend(position, ancestor_line = FALSE,
  ancestor_legend = c("Uncertain", "Present"),
  evol_legend = c("No change", "Retraction", "Expansion"),
  ancestor_col = c("#d9d9d9", "#252525"),
  evol_col = c("#b2df8a", "#984ea3", "#4daf4a"),
  pch = 22, pt.cex = 2.2, lty = 1, lwd = 1, cex = 1, bty = "n", ...)
```

Arguments

`position`	(character or numeric) position of legend. If character, part of the plot (e.g., "topleft"), see `legend`. If numeric, vector of two values indicating x and y position (e.g., c(0.1, 6)).
`ancestor_line`	whether or not ancestor line was plotted. Default = FALSE.
`ancestor_legend`	(character) vector of length = two indicating the text to identify environments with uncertain presence and true presence of the species. Default = c("Uncertain", "Present").

`evol_legend`	(character) vector of length = three indicating the text to identify environments where niches have not changed, have retracted or expanded. Default = c("No change", "Retraction", "Expansion").
`ancestor_col`	vector of two colors to represent what is indicated in `ancestor_legend`. Default = c("#d9d9d9", "#252525").
`evol_col`	vector of three colors to represent what is indicated in `evol_legend`. Default = c("#b2df8a", "#984ea3", "#4daf4a").
`pch`	point type as in `points`. Default = 22.
`pt.cex`	size of symbol (points). Default = 2.2.
`lty`	line type see `par`. Default = 1.
`lwd`	line width see `par`. Default = 1.
`cex`	size of all elements in legend see `par`. Default = 1.
`bty`	legend border type. Default = "n".
`...`	Other arguments from function `legend` other than the ones described above.

Examples

```
# a simple tree
data("tree5", package = "nichevol")

# a matrix of niche charactes (1 = present, 0 = absent, ? = unknown)
dataTable <- cbind("241" = rep("1", length(tree5$tip.label)),
                   "242" = rep("1", length(tree5$tip.label)),
                   "243" = c("1", "1", "0", "0", "0"),
                   "244" = c("1", "1", "0", "0", "0"),
                   "245" = c("1", "?", "0", "0", "0"))
rownames(dataTable) <- tree5$tip.label

# list with two objects (tree and character table)
treeWdata <- geiger::treedata(tree5, dataTable)

# Maximum parsimony reconstruction
rec_tab <- smooth_rec(bin_par_rec(treeWdata))

# plotting and adding labels and legend
ape::plot.phylo(tree5, label.offset = 0.04)
nichevol_labels(tree5, rec_tab, height = 0.6)
nichevol_legend(position = "bottomleft", cex = 0.7)
```

`niche_bars` *PNG bar figures to represent ecological niches of distinct taxa*

Description

niche_bars produces bar plots that represent species ecological niches in one environmental variable. Bars are exported as png figures to an output directory for posterior use.

Usage

```
niche_bars(tree, whole_rec_table, present = "1", unknown = "?",
           present_col = "#e41a1c", unknown_col = "#969696",
           absent_col = "#377eb8", width = 50, height = 5, res = 300,
           output_directory, overwrite = FALSE)
```

Arguments

`tree`	an object of class "phylo".
`whole_rec_table`	matrix of environmental bins for all tips and nodes derived from functions `bin_par_rec` or `bin_ml_rec`.
`present`	(character) code indicating environmental bins in which the species is present. Default = "1".
`unknown`	(character) code indicating environmental bins in which the species presence is unknown (uncertain). Default = "?".
`present_col`	color for area of the bar representing environments where the species is present. Default = "#e41a1c".
`unknown_col`	color for area of the bar representing environments where the species presence is unknown (uncertain). Default = "#969696".
`absent_col`	color for area of the bar representing environments where no change has been detected. Default = "#377eb8".
`width`	(numeric) width of the device in mm to be passed to the `png` function. Default = 50.
`height`	(numeric) height of the device in mm to be passed to the `png` function. Default = 5.
`res`	(numeric) nominal resolution in ppi to be passed to the `png` function. Default = 300.
`output_directory`	(character) name of the folder in which results will be written. The directory will be created as part of the process.
`overwrite`	(logical) whether or not to overwrite existing results in `output_directory`. Default = FALSE.

Details

Ecological niches are represented in one environmental dimension with vertical bars that indicate if the species is present, absent, or if its presence is uncertain in the range of environmental conditions. Lower values of environmental variables are represented in the left part of the bar, and the opposite part of the bar represents higher values.

Value

A folder named as in `output_directory` containing all bar figures produced, as well as a legend to describe what is plotted.

Examples

```
# a simple tree
data("tree5", package = "nichevol")

# a matrix of niche charactes (1 = present, 0 = absent, ? = unknown)
dataTable <- cbind("241" = rep("1", length(tree5$tip.label)),
                   "242" = rep("1", length(tree5$tip.label)),
                   "243" = c("1", "1", "0", "0", "0"),
                   "244" = c("1", "1", "0", "0", "0"),
                   "245" = c("1", "?", "0", "0", "0"))
rownames(dataTable) <- tree5$tip.label

# list with two objects (tree and character table)
treeWdata <- geiger::treedata(tree5, dataTable)

# Maximum parsimony reconstruction
rec_tab <- smooth_rec(bin_par_rec(treeWdata))

# the running (before running, define a working directory)
niche_bars(tree5, rec_tab, output_directory = file.path(tempdir(), "nichebars"))
```

`niche_labels` *Labels to represent niches of tips and ancestors*

Description

niche_labels helps in adding bar-type labels that represent species ecological niches in one environmental variable.

Usage

```
niche_labels(tree, whole_rec_table, label_type = "tip_node",
             tip_offset = 0.015, present = "1", unknown = "?",
             present_col = "#e41a1c", unknown_col = "#969696",
             absent_col = "#377eb8", width = 1, height = 1)
```

Arguments

`tree`	an object of class "phylo".
`whole_rec_table`	matrix of environmental bins for all tips and nodes derived from functions `bin_par_rec` or `bin_ml_rec`.
`label_type`	(character) type of label; options are: "tip", "node", and "tip_node". Default = "tip_node".
`tip_offset`	(numeric) space between tips and the labels. Default = 0.015.
`present`	(character) code indicating environmental bins in which the species is present. Default = "1".

`unknown`	(character) code indicating environmental bins in which the species presence is unknown (uncertain). Default = "?".
`present_col`	color for area of the bar representing environments where the species is present. Default = "#e41a1c".
`unknown_col`	color for area of the bar representing environments where the species presence is unknown (uncertain). Default = "#969696".
`absent_col`	color for area of the bar representing environments where no change has been detected. Default = "#377eb8".
`width`	value defining the width of niche bars; default = 1.
`height`	value defining the height of niche bars; default = 1.

Details

For the moment, only plots of type "phylogram" with "rightwards" or "leftwards" directions, created with the function `plot.phylo` from the package ape are supported.

Ecological niches are represented in one environmental dimension with vertical bars that indicate if the species is present, absent, or if its presence is uncertain in the range of environmental conditions. Lower values of environmental variables are represented in the lower part of the bar, and the opposite part of the bar represents higher values.

Examples

```
# a simple tree
data("tree5", package = "nichevol")

# a matrix of niche charactes (1 = present, 0 = absent, ? = unknown)
dataTable <- cbind("241" = rep("1", length(tree5$tip.label)),
                   "242" = rep("1", length(tree5$tip.label)),
                   "243" = c("1", "1", "0", "0", "0"),
                   "244" = c("1", "1", "0", "0", "0"),
                   "245" = c("1", "?", "0", "0", "0"))
rownames(dataTable) <- tree5$tip.label

# list with two objects (tree and character table)
treeWdata <- geiger::treedata(tree5, dataTable)

# Maximum parsimony reconstruction
rec_tab <- smooth_rec(bin_par_rec(treeWdata))

# plotting and adding labels
ape::plot.phylo(tree5, label.offset = 0.04)
niche_labels(tree5, rec_tab, height = 0.6)
```

`niche_legend` *Legends for niche labels in phylogenetic trees*

Description

Legends for niche labels in phylogenetic trees

Usage

```
niche_legend(position, legend = c("Uncertain", "Present", "Not present"),
  pch = 22, pt.bg = c("#969696", "#e41a1c", "#377eb8"),
  col = "transparent", pt.cex = 2.2, bty = "n", ...)
```

Arguments

`position`	(character or numeric) position of legend. If character, part of the plot (e.g., "topleft"), see `legend`. If numeric, vector of two values indicating x and y position (e.g., c(0.1, 6)).
`legend`	(character) vector of length = three indicating the text to identify environments with uncertain presence, presence, and absence of the species. Default = c("Uncertain", "Present", "Not present").
`pch`	point type as in `points`. Default = 22.
`pt.bg`	colors to represent what is in `legend`. Default = c("#969696", "#e41a1c", "#377eb8").
`col`	border of symbol (points). Default = "transparent".
`pt.cex`	size of symbol (points). Default = 2.2.
`bty`	legend border type. Default = "n".
`...`	Other arguments from function `legend` other than the ones described above.

Examples

```
# a simple tree
data("tree5", package = "nichevol")

# a matrix of niche charactes (1 = present, 0 = absent, ? = unknown)
dataTable <- cbind("241" = rep("1", length(tree5$tip.label)),
                   "242" = rep("1", length(tree5$tip.label)),
                   "243" = c("1", "1", "0", "0", "0"),
                   "244" = c("1", "1", "0", "0", "0"),
                   "245" = c("1", "?", "0", "0", "0"))
rownames(dataTable) <- tree5$tip.label

# list with two objects (tree and character table)
treeWdata <- geiger::treedata(tree5, dataTable)

# Maximum parsimony reconstruction
rec_tab <- smooth_rec(bin_par_rec(treeWdata))
```

```
# plotting and adding labels and legend
ape::plot.phylo(tree5, label.offset = 0.04)
niche_labels(tree5, rec_tab, height = 0.6)
niche_legend(position = "topleft", cex = 0.7)
```

`occ_list` *Example of occurrence records for six species*

Description

A list of 6 data.frames containing name and geographic coordinates for 6 species.

Usage

```
occ_list
```

Format

A list of 6 data.frames:

species species name, a code in this example

x longitude, longitude value

y latitude, latitude value

Examples

```
data("occ_list", package = "nichevol")

str(occ_list)
```

`par_rec_table` *Example of table with results from parsimony reconstructions*

Description

A character table representing species ecological niches derived from previous preparation processes and reconstructed niches for ancestors. Each row represents a species or a node and each column a binary character in which one or more values of the environmental variable are categorized as used "1", non used "0", or uncertain "?".

Usage

```
par_rec_table
```

Format

A character matrix with 11 rows and 20 columns.

Examples

```
data("par_rec_table", package = "nichevol")

head(par_rec_table)
```

`pdf_histograms` *Helper function to create PDF files with histograms*

Description

Helper function to create PDF files with histograms

Usage

```
pdf_histograms(env_data, occ_data, y_values, sp_names, variable_name,
               CL_lines, limits, col, output_directory)
```

Arguments

`env_data`	list of environmental values in M for all species.
`occ_data`	list of environmental values in occurrences for all species.
`y_values`	list of values for the y axis to be used to represent where occurrences are distributed across the environmental values in M.
`sp_names`	(character) names of the species for which the process will be performed.
`variable_name`	(character) name of the variable to be plotted.
`CL_lines`	(numeric) confidence limits to be plotted in the histograms.
`limits`	numeric matrix containing the actual values for the confidence limits of M.
`col`	color for lines representing the confidence limits of M.
`output_directory`	(character) name of the folder in which results will be written.

Value

A PDF file written in the output directory containing all resulting figures.

Examples

```
# example data
e_data <- list(rnorm(1000, 15, 7), rnorm(800, 20, 6), rnorm(1000, 12, 3))
o_data <- list(sample(seq(5, 29, 0.1), 45), sample(seq(10, 33, 0.1), 40),
               sample(seq(1, 16, 0.1), 50))
for (i in 1:3) {
  names(e_data[[i]]) <- e_data[[i]]
  names(o_data[[i]]) <- o_data[[i]]
}
y_val <- list(rep(3, length(o_data)), rep(4, length(o_data)),
              rep(2, length(o_data)))
s_names <- c("sp1", "sp2", "sp3")
lims <- rbind(c(3.5, 26.47), c(10.83, 29.66), c(6.92, 16.91))

tmpd <- file.path(tempdir(), "Hist_to_check") # temporal directory
dir.create(tmpd)

# the running (before running, create output_directory in current directory)
bins <- pdf_histograms(env_data = e_data, occ_data = o_data, y_values = y_val,
                       sp_names = s_names, variable_name = "Temperature",
                       CL_lines = 95, limits = lims, col = "green",
                       output_directory = tmpd)
```

`read_bin_table` *Read tables of binary niche characters from directory*

Description

Read one or multiple tables binary niche characters from directory.

Usage

```
read_bin_table(file)

read_bin_tables(directory)
```

Arguments

`file`	(character) name of CSV file containing a table of binary niche characters.
`directory`	(character) name of directory where tables of binary niche characters were written as CSV files.

Value

A matrix if `read_bin_table` is used.

A list of matrices if `read_bin_tables` is used.

`rename_tips` *Helper function to rename tips of trees for simulations*

Description

Helper function to rename tips of trees for simulations

Usage

```
rename_tips(tree, names)
```

Arguments

`tree`	an object of class "phylo".
`names`	(character) vector of new names. Length must be equal to number of tips. They will be assigned in the order given.

Value

Tree of class "phylo" with specified names

Examples

```
# a simple tree
data("tree5", package = "nichevol")

# renaming tips
renamedTree <- rename_tips(tree5, c("a", "b", "c", "d", "e"))
```

`score_tip` *Helper function to calculate the median bin score for a given species*

Description

Helper function to calculate the median bin score for a given species

Usage

```
score_tip(character_table, species_name, include_unknown = FALSE)
```

Arguments

`character_table`	data.frame containing bin scores for all species. NOTE: row names must be species' names.
`species_name`	(character) name of the species to be analyzed.
`include_unknown`	(logical) whether or not unknown bin status should be included.

Value

Median bin value for a given species (for inferring sigma squared or other comparative phylogenetic analyses requiring a single continuous variable).

Examples

```
# Simulate data for single number bin labels
dataTable <- cbind("241" = rep("1", 5),
                   "242" = rep("1", 5),
                   "243" = c("1", "1", "0", "0", "0"),
                   "244" = c("1", "1", "0", "0", "0"),
                   "245" = c("1", "?", "0", "0", "0"))
 rownames(dataTable) <- c("GadusMorhua", "GadusMacrocephalus",
                          "GadusChalcogrammus", "ArctogadusGlacials",
                          "BoreogadusSaida")
# Simulate data for bin labels as strings
dataTableStringLabel <- cbind("241 to 244" = rep("1", 5),
                              "244 to 246" = c("1", "1", "0", "0", "0"),
                              "246 to 248" = c("1", "?", "0", "0", "0"))
rownames(dataTableStringLabel) <- c("GadusMorhua", "GadusMacrocephalus",
                                    "GadusChalcogrammus", "ArctogadusGlacials",
                                    "BoreogadusSaida")
# Use function
score_tip(character_table = dataTable, species_name = "GadusMorhua",
          include_unknown = TRUE)
score_tip(character_table = dataTableStringLabel, species_name = "GadusMorhua",
          include_unknown = FALSE)
```

`score_tree` *Helper function to assign bin scores to every tip in a given tree*

Description

Helper function to assign bin scores to every tip in a given tree

Usage

```
score_tree(tree_data, include_unknown = FALSE)
```

Arguments

`tree_data`	a list of two elements (phy and data) resulting from using the function `treedata`.
`include_unknown`	(logical) whether or not there are unknown tips.

Value

a list of two elements (phy and data). Data is the median bin scored as present or present + unknown.

Examples

```
# Simulate data table
dataTable <- cbind("241" = rep("1", 5),
                   "242" = rep("1", 5),
                   "243" = c("1", "1", "0", "0", "0"),
                   "244" = c("1", "1", "0", "0", "0"),
                   "245" = c("1", "?", "0", "0", "0"))
rownames(dataTable) <- c("GadusMorhua", "GadusMacrocephalus",
                         "GadusChalcogrammus", "ArctogadusGlacials",
                         "BoreogadusSaida")

# a simple tree
data("tree5", package = "nichevol")
tree5$tip.label <- c("GadusMorhua", "GadusMacrocephalus",
                     "GadusChalcogrammus", "ArctogadusGlacials",
                     "BoreogadusSaida")

# Unite data
treeWithData <- geiger::treedata(tree5, dataTable)

# Get a new tree with tips scored from median bin scores
score_tree(treeWithData, include_unknown = TRUE)
```

`set_uncertainty` *Set values of uncertainty towards one or both ends of the variable*

Description

set_uncertainty allows to define uncertainty ("?") values around values denoting presence ("1") towards one or both ends of the variable in a table of binary characters.

Usage

```
set_uncertainty(character_table, species, end)
```

Arguments

`character_table`	a matrix of characters to represent ecological niches of the species of interest. A matrix containing values "1" = presence, "0" = absence, and "?" = uncertain. See `bin_table`.
`species`	(character) name of the species in the table for which values of uncertainty will be set.
`end`	(character) end towards which uncertainty values ("?") will be set. Options are: "high", "low", or "both".

Details

Values of characters around those denoting presence ("1") are manually transformed to uncertain ("?") to help producing more conservative reconstructions of ancestral ecological niches. This increases uncertainty in reconstructions and further niche comparisons, which reduces the events of niche change that can be detected. This may be especially useful when dealing with species with one or just a few known records.

Value

A modified matrix of characters to represent ecological niches of the species of interest.

Potential values for characters are:

- "1" = the species is present in those environmental conditions.
- "0" = the species is not present in those environmental conditions. This is, those environmental conditions inside the accessible area (M) are more extreme than the ones used for the species.
- "?" = there is no certainty about the species presence in those environmental conditions.

Examples

```
# a character table
data("character_table", package = "nichevol")

character_table[, 20:28]

# set values of uncertainty towards the lower end of the variable for species t3
char_tableu <- set_uncertainty(character_table, species = "t2", end = "low")

char_tableu[, 20:28]
```

`sig_sq` *Helper function to get sigma squared values for a given dataset*

Description

Sigma squared values for a single niche summary statistic are calculated using `fitContinuous`.

Usage

```
sig_sq(tree_data, model = "BM")
```

Arguments

`tree_data`	a list of two elements (phy and data) resulted from using the function `treedata`. NOTE: data must be a single vector (i.e., a single column).
`model`	model to fit to comparative data; see `fitContinuous`. Default = "BM".

Value

the sigma squared value (evolutionary rate) for the data, given the tree.

Examples

```
# a simple tree
data("tree5", package = "nichevol")

# simple data
data <- rnorm(n = length(tree5$tip.label))
names(data) <- tree5$tip.label
# tree with data
treeWdata <- geiger::treedata(tree5, data)

# Estimating sigma squared for the dataset
sig_sq(treeWdata)
```

`smooth_rec` *Smooth character table values resulted from ancestral character state reconstructions*

Description

Smooth character table values resulted from ancestral character state reconstructions

Usage

```
smooth_rec(whole_rec_table)
```

Arguments

`whole_rec_table`	matrix containing all reconstructed characters for all tips and nodes. It results from using the functions `bin_par_rec` or `bin_ml_rec`.

Value

The matrix of reconstructed characters with smoothed values.

Examples

```
# a simple tree
data("tree5", package = "nichevol")

# simple matrix of data
dataTable <- cbind("241" = rep("1", length(tree5$tip.label)),
                   "242" = rep("1", length(tree5$tip.label)),
                   "243" = c("1", "1", "0", "0", "0"),
                   "244" = c("1", "1", "0", "0", "0"),
```

```
                    "245" = c("1", "?", "0", "0", "0"))
rownames(dataTable) <- tree5$tip.label
treeWdata <- geiger::treedata(tree5, dataTable)

# ancestral reconstruction
parsimonyReconstruction <- bin_par_rec(treeWdata)

# smoothing reconstructions
smooth_rec(parsimonyReconstruction)
```

`stats_eval` *Statistics of environmental conditions in M and for occurrences (one variable)*

Description

stats_eval helps in creating tables of descriptive statistics of environmental conditions in accessible areas (M) and occurrence records for one environmental variable at a time.

Usage

```
stats_eval(stats = c("median", "range"), Ms, occurrences, species,
           longitude, latitude, variable, percentage_out = 0, verbose = TRUE)
```

Arguments

`stats`	(character) name or vector of names of functions to be applied to get basic statistics of environmental values.
`Ms`	a list of SpatVector objects representing the accessible area (M) for each species to be analyzed. The order of species represented by each object here must coincide with the one in `occurrences`. See details.
`occurrences`	a list of data.frames of occurrence records for all species. The order of species represented by each data.frame must coincide with the one in `Ms`. See details.
`species`	(character) name of the column in occurrence data.frames that contains the name of the species.
`longitude`	(character) name of the column in occurrence files containing values of longitude.
`latitude`	(character) name of the column in occurrence files containing values of latitude.
`variable`	a single SpatRaster layer of an environmental variable of interest. See details.
`percentage_out`	(numeric) percentage of extreme environmental data in M to be excluded in bin creation for further analyses. See details. Default = 0.
`verbose`	(logical) whether messages should be printed. Default = TRUE.

Details

Coordinates in `occurrences`, SpatVector objects in `Ms`, and SpatRaster in `variable` must coincide in the geographic projection in which they are represented. WGS84 with no planar projection is recommended.

Accessible area (M) is understood as the geographic area that has been accessible for a species for relevant periods of time. Defining M is usually a hard task, but also a very important one, because it allows identifying uncertainties about the ability of a species to maintain populations in certain environmental conditions. For further details on this topic, see Barve et al. (2011) `doi:10.1016/j.ecolmodel.2011.02.011` and Machado-Stredel et al. (2021) `doi:10.21425/F5FBG48814`.

The percentage to be defined in `percentage_out` excludes a percentage of extreme environmental values to prevent from considering extremely rare environmental values in the accessible area for the species (M). Being too rare, these values may have never been explored by the species; therefore, including them in the process of preparation of the table of characters (bin table) is risky.

Value

A list containing tables with statistics of the values in `variable`, for the species M and occurrences.

Examples

```
# example data
## list of species records
data("occ_list", package = "nichevol")

## list of species accessible areas
m_files <- list.files(system.file("extdata", package = "nichevol"),
                      pattern = "m\\d.gpkg", full.names = TRUE)

m_list <- lapply(m_files, terra::vect)

## raster variable
temp <- terra::rast(system.file("extdata", "temp.tif", package = "nichevol"))

# running stats
stat <- stats_eval(stats = c("mean", "sd", "median", "range", "quantile"),
                   Ms = m_list, occurrences = occ_list, species = "species",
                   longitude = "x", latitude = "y", variable = temp,
                   percentage_out = 0)
```

`stats_evalues` *Statistics of environmental conditions in M and for occurrences (multiple variables)*

Description

stats_evalues helps in creating csv files with statistics of environmental conditions in accessible areas (M) and species occurrence records. This is done using data read directly from a local directory, and can be applied to various species and multiple variables.

Usage

```
stats_evalues(stats = c("median", "range"), M_folder, M_format, occ_folder,
              longitude, latitude, var_folder, var_format, round = FALSE,
              round_names, multiplication_factor = 1, percentage_out = 0,
              save = FALSE, output_directory, overwrite = FALSE,
              verbose = TRUE)
```

Arguments

`stats`	(character) name or vector of names of functions to be applied to get basic statistics of environmental values.
`M_folder`	(character) name of the folder containing files representing the accessible area (M) for each species to be analyzed. See details.
`M_format`	format of files representing the accessible area (M) for the species. Names of M files must match the ones for occurrence files in `occ_folder`. Format options are: "shp", "gpkg", or any of the options supported by `rast` (e.g., "tif" or "asc").
`occ_folder`	(character) name of the folder containing csv files of occurrence data for all species. Names of csv files must match the ones of M files in `M_folder`.
`longitude`	(character) name of the column in occurrence files containing values of longitude.
`latitude`	(character) name of the column in occurrence files containing values of latitude.
`var_folder`	(character) name of the folder containing layers to represent environmental variables.
`var_format`	format of layers to represent environmental variables. Format options are all the ones supported by `rast` (e.g., "tif" or "asc").
`round`	(logical) whether or not to round the values of one or more variables after multiplying them times the value in `multiplication_factor`. Default = FALSE. See details.
`round_names`	(character) names of the variables to be rounded. Default = NULL. If `round` = TRUE, names must be defined.
`multiplication_factor`	(numeric) value to be used to multiply the variables defined in `round_names`. Default = 1.
`percentage_out`	(numeric) percentage of extreme environmental data in M to be excluded in bin creation for further analyses. See details. Default = 0.
`save`	(logical) whether or not to save the results in working directory. Default = FALSE.
`output_directory`	(character) name of the folder in which results will be written.
`overwrite`	(logical) whether or not to overwrite existing results in `output_directory`. Default = FALSE.
`verbose`	(logical) whether messages should be printed. Default = TRUE.

Details

Coordinates in csv files in `occ_folder`, SpatVector-like files in `M_folder`, and raster layers in `var_folder` must coincide in the geographic projection in which they are represented. WGS84 with no planar projection is recommended.

Accessible area (M) is understood as the geographic area that has been accessible for a species for relevant periods of time. Defining M is usually a hard task, but also a very important one, because it allows identifying uncertainties about the ability of a species to maintain populations in certain environmental conditions. For further details on this topic, see Barve et al. (2011) `doi:10.1016/j.ecolmodel.2011.02.011` and Machado-Stredel et al. (2021) `doi:10.21425/F5FBG48814`.

Rounding variables may be useful when multiple variables are considered and the values of some or all of them are too small (e.g., when using principal components). To round specific variables arguments `round`, `round_names`, and `multiplication_factor`, must be used accordingly.

The percentage to be defined in `percentage_out` excludes a percentage of extreme environmental values to prevent the algorithm from considering extremely rare environmental values in the accessible area for the species (M). Being too rare, these values may have never been explored by the species; therefore, including them in the process of preparation of the table of characters (bin table) is risky.

Value

A list named as the variables present in `var_folder`, containing all tables with statistics of environmental values in M and in species records. A folder named as in `output_directory` containing all resultant csv files with the tables of statistics will be created if `save` is set as TRUE.

Examples

```
# preparing data and directories for examples
## directories
tempdir <- file.path(tempdir(), "nevol_test")
dir.create(tempdir)

cvariables <- paste0(tempdir, "/variables")
dir.create(cvariables)

records <- paste0(tempdir, "/records")
dir.create(records)

m_areas <- paste0(tempdir, "/M_areas")
dir.create(m_areas)

## data
data("occ_list", package = "nichevol")

temp <- system.file("extdata", "temp.tif", package = "nichevol")

m_files <- list.files(system.file("extdata", package = "nichevol"),
                      pattern = "m\\d.gpkg", full.names = TRUE)

## writing data in temporal directories
```

```
spnames <- sapply(occ_list, function (x) as.character(x[1, 1]))
ocnames <-  paste0(records, "/", spnames, ".csv")

occs <- lapply(1:length(spnames), function (x) {
  write.csv(occ_list[[x]], ocnames[x], row.names = FALSE)
})

to_replace <- paste0(system.file("extdata", package = "nichevol"), "/")

otemp <- gsub(to_replace, "", temp)
file.copy(from = temp, to = paste0(cvariables, "/", otemp))

file.copy(from = m_files, to = paste0(m_areas, "/", spnames, ".gpkg"))
stats <- stats_evalues(stats = c("median", "range"), M_folder = m_areas,
                       M_format = "gpkg", occ_folder = records,
                       longitude = "x", latitude = "y",
                       var_folder = cvariables, var_format = "tif",
                       percentage_out = 5)
```

`temp` *Example of an environmental variable used in analysis*

Description

A SpatRaster object representing the variable temperature.

Format

A SpatRaster object.

Value

No return value, used with function `rast` to bring an example of an environmental variable used in analysis.

Examples

```
temp <- terra::rast(system.file("extdata", "temp.tif", package = "nichevol"))

terra::plot(temp)
```

`tree` *Example of a phylogenetic tree for six species*

Description

A phylogenetic tree with 6 species and their relationships.

Usage

```
tree
```

Format

An object of class phylo for 6 species.

Examples

```
data("tree", package = "nichevol")

str(tree)
```

`tree5` *Example of a phylogenetic tree for five species*

Description

A phylogenetic tree with 5 species and their relationships.

Usage

```
tree5
```

Format

An object of class phylo for 5 species.

Examples

```
data("tree5", package = "nichevol")

str(tree5)
```

`tree_data` *Example of a list containing a tree and a table of characters for six species*

Description

A list of 2 elements (phy and data) resulting from using the function `treedata`.

Usage

```
tree_data
```

Format

A list of 2 elements:

phy object of class phylo for 6 species

data matrix of 6 rows and 28 columns

Examples

```
data("tree_data", package = "nichevol")

str(tree_data)
```

Index