

Package ‘crestr’

November 27, 2020

Title What the Package Does (One Line, Title Case)

Version 0.0.0.9000

Description What the package does (one paragraph).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports DBI,
RPostgreSQL,
rio,
graphics,
plot3D,
viridis,
grDevices,
methods,
stringr,
raster,
plyr,
sp,
rgdal,
RPostgres

Depends R (>= 2.10)

Suggests testthat,
knitr,
rmarkdown,
clipr

VignetteBuilder knitr

R topics documented:

accClimateVariables	2
accContinentNames	3
accRealmNames	3
calib_clim_space	4
cite_GBIF	4

close_db_connection	5
connect_online	5
convert2percentages	6
convert2presenceAbsence	7
copy_crest	7
crest	8
crest.calibrate	10
crest.get_modern_data	11
crest.reconstruct	13
crest_ex	14
crest_ex_pse	15
crest_ex_selection	15
crestObj	16
dbRequest	18
eqearth_get_ext	18
fit_pdfsp	19
fit_xrange	20
getClimateSpace	21
getDistribTaxa	22
getTaxonID	23
getTaxonomy	24
isColourStr	25
loo	25
M1	26
meanPositiveValues	26
normalise	27
plot.crestObj	27
plot_climateSpace	28
plot_diagram	29
plot_loo	30
plot_map_eqearth	32
plot_taxaCharacteristics	33
Index	35

accClimateVariables	<i>Describes all the variables available in the database.</i>
---------------------	---

Description

Provides the index and the short and full names of all the variables available in the database.

Usage

```
accClimateVariables(v = NA)
```

Arguments

v	The name of a variable to quickly access its description and ID (default NA returns all possible values).
---	---

Value

A data frame descriptive of the climate variables available in the database (if v=NA) or the description of variable v.

Examples

```
accClimateVariables()  
accClimateVariables(v='bio12')
```

accContinentNames	<i>Return the list of the continents and associated countries.</i>
-------------------	--

Description

Return the list of the continents and associated countries.

Usage

```
accContinentNames(dbname = "gbif4crest_02")
```

Arguments

dbname The name of the database. Default is gbif4crest_02.

Value

A list where each element is a vector of corresponding country names.

Examples

```
accContinentNames()
```

accRealmNames	<i>Return the list of the realms and associated biomes and ecoregions.</i>
---------------	--

Description

Return the list of the realms and associated biomes and ecoregions.

Usage

```
accRealmNames(ecoregion = TRUE, dbname = "gbif4crest_02")
```

Arguments

ecoregion A boolean to choose whether to get the ecoregions names.
dbname The name of the database. Default is gbif4crest_02.

Value

A list with elements that correspond to the biomes (and possibly ecoregions) of each realm.

Examples

```
accRealmNames()
```

calib_clim_space	<i>Calibrate the distribution of the modern climate space.</i>
------------------	--

Description

Calibrate the distribution of the modern climate space.

Usage

```
calib_clim_space(climate, bin_width)
```

Arguments

climate	A vector of climatic values where the species is present.
bin_width	The width of the bins used to correct for unbalanced climate state. Use values that split the studied climate gradient in 15-25 classes (e.g. 2°C for temperature variables). Default is 1.

Value

A ccs object that will be used by [fit_pdfsp](#).

Examples

```
calib_clim_space(sample(0:300 / 10, 4000, replace = TRUE), 2)
```

cite_GBIF	<i>Returns the citations associated to the GBIF data used to fit the pdfs.</i>
-----------	--

Description

Returns the citations associated to the GBIF data used to fit the pdfs.

Usage

```
cite_GBIF(x, dbname = "gbif4crest_02", verbose = TRUE)
```

Arguments

x	A crestObj produced by one of the crest functions.
dbname	The name of the database. Default is gbif4crest_02.
verbose	A boolean to print non-essential comments on the terminal (default TRUE).

close_db_connection	<i>Disconnect the database connection.</i>
---------------------	--

Description

Disconnect the database connection.

Usage

```
close_db_connection(db)
```

Arguments

db	An active database connection
----	-------------------------------

Examples

```
db <- connect_online()
close_db_connection(db)
## Not run:
db <- connect_online()
close_db_connection(db)

## End(Not run)
```

connect_online	<i>Connect to the gbif4crest database</i>
----------------	---

Description

Connect to the gbif4crest_02 database by accessing the server on Amazon.

Usage

```
connect_online(
  dbname = "gbif4crest_02",
  port = 5432,
  host = "gbif4crest.cvqgy2mnjwgtg.eu-west-3.rds.amazonaws.com",
  user = "guestuser",
  password = "pwd12345"
)
```

Arguments

dbname	The name of the database. Default is gbif4crest_02.
port	The port to connect to the server. Default is 5432.
host	The host of the database server. Default is gbif4crest.cvqgy2mnjwgtg.eu-west-3.rds.amazonaws.com
user	The user name to use to connect. Default is guestuser.
password	The password associated with the user name. Default is pwd12345

Value

An active connection to a database

Examples

```
## Not run:  
db <- connect_online()  
  
## End(Not run)
```

convert2percentages	<i>Convert abundance data into percentage data.</i>
---------------------	---

Description

Convert abundance data into percentage data.

Usage

```
convert2percentages(df, col2convert = 2:ncol(df))
```

Arguments

df	The dataframe containing the data to convert.
col2convert	A vector of the columns to convert. Default is all the columns but the first, which contains an age, a depth or a sampleID.

Value

A vector of unique taxonIDs.

Examples

```
df <- data.frame(matrix(1:25, ncol = 5))  
colnames(df) <- paste(rep("col", 5), 1:5, sep = "")  
convert2percentages(df)  
convert2percentages(df, col2convert = 3:5)
```

 convert2presenceAbsence

Convert data into presence/absence data.

Description

Convert data into presence/absence data.

Usage

```
convert2presenceAbsence(df, threshold = 2, col2convert = 2:ncol(df))
```

Arguments

df	The dataframe containing the data to convert.
threshold	The threshold that defines presence (presence if \geq threshold)
col2convert	A vector of the columns to convert. Default is all the columns but the first, which contains an age, a depth or a sampleID.

Value

A vector of unique taxonIDs.

Examples

```
df <- data.frame(matrix(1:25, ncol = 5))
colnames(df) <- paste(rep("col", 5), 1:5, sep = "")
convert2presenceAbsence(df, threshold = 15)
convert2presenceAbsence(df, col2convert = 3:5)
```

 copy_crest

Copy crest data to the clipboard.

Description

Copy crest data to the clipboard for an easy extraction of the data from the R environment.

Usage

```
copy_crest(
  x,
  climate = x$parameters$climate,
  optima = TRUE,
  mean = FALSE,
  uncertainties = FALSE
)
```

Arguments

x	A crestObj produced by the crest.reconstruct() or crest() functions.
climate	A vector of the climate variables to extract. See accClimateVariables for the list of accepted values.
optima	A boolean value to indicate if the optima should be copied to the clipboard.
mean	A boolean value to indicate if the means should be copied to the clipboard.
uncertainties	A boolean value to indicate if the uncertainties should be copied to the clipboard.

Examples

```
## Not run:
if(requireNamespace('clipr', quietly=TRUE)) {
  recons <- crest(
    df = crest_ex, pse = crest_ex_pse, taxaType = 0,
    climate = c("bio1", "bio12"), bin_width = c(2, 20),
    shape = c("normal", "lognormal"),
    selectedTaxa = crest_ex_selection, dbname = "crest_example",
    leave_one_out = TRUE
  )
  copy_crest(recons, uncertainties=TRUE)
  ## You can now paste the values in a spreadsheet.
}

## End(Not run)
```

crest

Connect to the gbif4crest database

Description

Connect to the gbif4crest_02 database by accessing the server on Amazon.

Usage

```
crest(
  df,
  pse,
  taxaType,
  climate,
  xmn = -180,
  xmx = 180,
  ymn = -90,
  ymx = 90,
  continents = NA,
  countries = NA,
  realms = NA,
  biomes = NA,
  ecoregions = NA,
  minGridCells = 20,
  selectedTaxa = NA,
```



```

bin_width = rep(1, length(x$parameters$climate)),
shape = rep("normal", length(x$parameters$climate)),
npoints = 500,
geoWeighting = TRUE,
climateSpaceWeighting = TRUE,
presenceThreshold = 0,
taxWeight = "normalisation",
uncertainties = c(0.5, 0.95),
leave_one_out = FALSE,
verbose = TRUE,
dbname = "gbif4crest_02"
)

```

Arguments

df	A data frame containing the data to reconstruct (counts, percentages or presence/absence data).
pse	A pollen-Species equivalency table. See get_pse for details.
taxaType	A numerical index (between 1 and 6) to define the type of palaeoproxy used: 1 for plants, 2 for beetles, 3 for foraminifers, 4 for diatoms, 5 for chironomids and 6 for rodents. The example dataset uses taxaType=0. Default is 1.
climate	A vector of the climate variables to extract. See accClimateVariables for the list of accepted values.
xmn	The coordinates defining the study area.
xmx	The coordinates defining the study area.
ymn	The coordinates defining the study area.
ymx	The coordinates defining the study area.
continents	A vector of the continent names defining the study area.
countries	A vector of the country names defining the study area.
realms	A vector of the studied botanical realms defining the study area.
biomes	A vector of the studied botanical biomes defining the study area.
ecoregions	A vector of the studied botanical ecoregions defining the study area.
minGridCells	The minimum number of unique presence data necessary to estimate a species' climate response. Default is 20.
selectedTaxa	A data frame assigns which taxa should be used for each variable (1 if the taxon should be used, 0 otherwise). The colnames should be the climate variables' names and the rownames the taxa names. Default is 1 for all taxa and all variables.
bin_width	The width of the bins used to correct for unbalanced climate state. Use values that split the studied climate gradient in 15-25 classes (e.g. 2°C for temperature variables). Default is 1.
shape	The imposed shape of the species pdfs. We recommend using 'normal' for temperature variables and 'lognormal' for the variables that can only take positive values, such as precipitation or aridity. Default is 'normal' for all.
npoints	The number of points to be used to fit the pdfs. Default 200.
geoWeighting	A boolean to indicate if the species should be weighting by the squareroot of their extension when estimating a genus/family level taxon-climate relationships.

climateSpaceWeighting	A boolean to indicate if the species pdfs should be corrected for the modern distribution of the climate space (default TRUE).
presenceThreshold	All values above that threshold will be used in the reconstruction (e.g. if set at 1, all percentages below 1 will be set to 0 and the associated presences discarded). Default is 0.
taxWeight	One value among the following: 'originalData', 'presence/absence', 'percentages' or 'normalisation' (default).
uncertainties	A (vector of) threshold value(s) indicating the error bars that should be calculated (default both 50 and 95% ranges).
leave_one_out	A boolean to indicate whether the leave one out (loo) reconstructions should be computed (default FALSE).
verbose	A boolean to print non-essential comments on the terminal (default TRUE).
dbname	The name of the database. Default is gbif4crest_02.

Value

The parameters to be used by crest()

Examples

```
data(crest_ex)
data(crest_ex_pse)
data(crest_ex_selection)
recons <- crest(
  df = crest_ex, pse = crest_ex_pse, taxaType = 0,
  climate = c("bio1", "bio12"), bin_width = c(2, 20),
  shape = c("normal", "lognormal"),
  selectedTaxa = crest_ex_selection, dbname = "crest_example",
  leave_one_out = TRUE
)
plot(recons)
plot_loo(recons)
```

crest.calibrate	<i>Fit the species and proxy pdfs</i>
-----------------	---------------------------------------

Description

This function fits the climate response of the selected taxa to the selected climate variables.

Usage

```
crest.calibrate(
  x,
  bin_width = x$parameters$bin_width,
  shape = x$parameters$shape,
  npoints = x$parameters$npoints,
  geoWeighting = x$parameters$geoWeighting,
```

```

    climateSpaceWeighting = x$parameters$climateSpaceWeighting,
    verbose = TRUE
  )

```

Arguments

x	A crestObj produced by the <code>crest.get_modern_data</code> function.
bin_width	The width of the bins used to correct for unbalanced climate state. Use values that split the studied climate gradient in 15-25 classes (e.g. 2°C for temperature variables). Default is 1.
shape	The imposed shape of the species pdfs. We recommend using 'normal' for temperature variables and 'lognormal' for the variables that can only take positive values, such as precipitation or aridity. Default is 'normal' for all.
npoints	The number of points to be used to fit the pdfs. Default 200.
geoWeighting	A boolean to indicate if the species should be weighting by the squareroot of their extension when estimating a genus/family level taxon-climate relationships.
climateSpaceWeighting	A boolean to indicate if the species pdfs should be corrected for the modern distribution of the climate space (default TRUE).
verbose	A boolean to print non-essential comments on the terminal (default TRUE).

Value

A `crest()` object containing the spatial distributions and the climate space.

Examples

```

## Not run:
data(crest_ex)
data(crest_ex_pse)
data(crest_ex_selection)
x <- crest.get_modern_data(
  pse = crest_ex_pse, taxaType = 0,
  climate = c("bio1", "bio12"),
  selectedTaxa = crest_ex_selection, dbname = "crest_example"
)
x <- crest.calibrate(x,
  geoWeighting = TRUE, climateSpaceWeighting = TRUE,
  bin_width = c(2, 20), shape = c("normal", "lognormal")
)

## End(Not run)

```

`crest.get_modern_data` *Extract distributions from the database*

Description

This function will extract the distributions of all the species composing each taxon and return them as a list.

Usage

```
crest.get_modern_data(
  pse,
  taxaType,
  climate,
  taxa.name = unique(pse[, "ProxyName"]),
  xmn = NA,
  xmx = NA,
  ymn = NA,
  ymx = NA,
  continents = NA,
  countries = NA,
  realms = NA,
  biomes = NA,
  ecoregions = NA,
  minGridCells = 20,
  selectedTaxa = NA,
  dbname = "gbif4crest_02",
  verbose = TRUE
)
```

Arguments

pse	A pollen-Species equivalency table. See get_pse for details.
taxaType	A numerical index (between 1 and 6) to define the type of palaeoproxy used: 1 for plants, 2 for beetles, 3 for foraminifers, 4 for diatoms, 5 for chironomids and 6 for rodents. The example dataset uses taxaType=0. Default is 1.
climate	A vector of the climate variables to extract. See accClimateVariables for the list of accepted values.
taxa.name	A vector that contains the names of the taxa to study.
xmn	The coordinates defining the study area.
xmx	The coordinates defining the study area.
ymn	The coordinates defining the study area.
ymx	The coordinates defining the study area.
continents	A vector of the continent names defining the study area.
countries	A vector of the country names defining the study area.
realms	A vector of the studied botanical realms defining the study area.
biomes	A vector of the studied botanical biomes defining the study area.
ecoregions	A vector of the studied botanical ecoregions defining the study area.
minGridCells	The minimum number of unique presence data necessary to estimate a species' climate response. Default is 20.
selectedTaxa	A data frame assigns which taxa should be used for each variable (1 if the taxon should be used, 0 otherwise). The colnames should be the climate variables' names and the rownames the taxa names. Default is 1 for all taxa and all variables.
dbname	The name of the database. Default is gbif4crest_02.
verbose	A boolean to print non-essential comments on the terminal (default TRUE).

Value

A crest() object containing the spatial distributions

Examples

```
## Not run:
data(crest_ex_pse)
data(crest_ex_selection)
x <- crest.get_modern_data(
  pse = crest_ex_pse, taxaType = 0,
  climate = c("bio1", "bio12"),
  selectedTaxa = crest_ex_selection, dbname = "crest_example"
)
x
lapply(x$modelling$distributions, head)

## End(Not run)
```

crest.reconstruct	<i>Fit the species and proxy pdfs</i>
-------------------	---------------------------------------

Description

This function fits the climate response of the selected taxa to the selected climate variables.

Usage

```
crest.reconstruct(
  x,
  df,
  presenceThreshold = 0,
  taxWeight = "normalisation",
  uncertainties = c(0.5, 0.95),
  skip_for_loo = FALSE,
  verbose = TRUE
)
```

Arguments

x	A crestObj produced by the crest.fit_pdfs function.
df	A data frame containing the data to reconstruct (counts, percentages or presence/absence data).
presenceThreshold	All values above that threshold will be used in the reconstruction (e.g. if set at 1, all percentages below 1 will be set to 0 and the associated presences discarded). Default is 0.
taxWeight	One value among the following: 'originalData', 'presence/absence', 'percentages' or 'normalisation' (default).
uncertainties	A (vector of) threshold value(s) indicating the error bars that should be calculated (default both 50 and 95% ranges).

skip_for_loo	A boolean that tells the loo() function to skip parts and fasten the process. Not for users.
verbose	A boolean to print non-essential comments on the terminal (default TRUE).

Value

A crest() object containing the reconstructions and all the associated data.

Examples

```
data(crest_ex)
data(crest_ex_pse)
data(crest_ex_selection)
x <- crest.get_modern_data(
  pse = crest_ex_pse, taxaType = 0,
  climate = c("bio1", "bio12"),
  selectedTaxa = crest_ex_selection, dbname = "crest_example"
)
x <- crest.calibrate(x,
  geoWeighting = TRUE, climateSpaceWeighting = TRUE,
  bin_width = c(2, 20), shape = c("normal", "lognormal")
)
x <- crest.reconstruct(x, crest_ex)
plot(x)
```

crest_ex

Example dataset to run the CREST method for the first time.

Description

A dataset containing 20 randomly generated pollen samples for 7 pollen taxa.

Usage

```
crest_ex
```

Format

A data frame with 20 rows (samples) and 8 columns (1 column for the age and one for each of the 7 taxa):

Age: Age of each sample

Taxon1: Percentage of Taxon1 in each sample.

Taxon2: Percentage of Taxon2 in each sample.

Taxon3: Percentage of Taxon3 in each sample.

Taxon4: Percentage of Taxon4 in each sample.

Taxon5: Percentage of Taxon5 in each sample.

Taxon6: Percentage of Taxon6 in each sample.

Taxon7: Percentage of Taxon7 in each sample.

`crest_ex_pse`*Example dataset to Extract data from the example database.*

Description

A database indicating the taxonomy of the example proxies.

Usage`crest_ex_pse`**Format**

A data frame with 7 rows (taxa) and 5 columns (taxonomy description):

Level: An integer indicating the taxonomic resolution (1 family, 2 genus, 3 species, 4 or higher ignore taxon)

Family: The family corresponding to the ProxyName

Genus: The genus corresponding to the ProxyName

Species: The species corresponding to the ProxyName

ProxyName: The names of the observed proxies, as reported in the main data file

`crest_ex_selection`*Example dataset to associate taxa with climate variables.*

Description

A data frame indicating the taxa that should be used to reconstruct each climate variable (1s in the matrix) and those who should be excluded (0s).

Usage`crest_ex_selection`**Format**

A data frame with 7 rows (taxa) and 2 columns (climate variables):

bio1: The first variable to reconstruct (mean annual temperature)

bio12: The second variable to reconstruct (annual precipitation)

crestObj	Create a crest() object.
----------	--------------------------

Description

Creates a crest() object with all default parameters.

Usage

```
crestObj(
  taxa.name,
  taxaType,
  climate,
  pse = NA,
  continents = NA,
  countries = NA,
  realms = NA,
  biomes = NA,
  ecoregions = NA,
  xmn = -180,
  xmx = 180,
  ymn = -90,
  ymx = 90,
  df = NA,
  x = NA,
  x.name = "",
  minGridCells = 20,
  bin_width = rep(1, length(climate)),
  shape = rep("normal", length(climate)),
  npoints = 200,
  geoWeighting = TRUE,
  climateSpaceWeighting = TRUE,
  selectedTaxa = NA,
  presenceThreshold = 0,
  taxWeight = "normalisation",
  uncertainties = c(0.5, 0.95)
)
```

Arguments

taxa.name	A vector that contains the names of the taxa to study.
taxaType	A numerical index (between 1 and 6) to define the type of palaeoproxy used: 1 for plants, 2 for beetles, 3 for foraminifers, 4 for diatoms, 5 for chironomids and 6 for rodents. The example dataset uses taxaType=0. Default is 1.
climate	A vector of the climate variables to extract. See accClimateVariables for the list of accepted values.
pse	A pollen-Species equivalency table. See get_pse for details.
continents	A vector of the continent names defining the study area.
countries	A vector of the country names defining the study area.

realms	A vector of the studied botanical realms defining the study area.
biomes	A vector of the studied botanical biomes defining the study area.
ecoregions	A vector of the studied botanical ecoregions defining the study area.
xmn, xmx, ymn, ymx	The coordinates defining the study area.
df	A data frame containing the data to reconstruct (counts, percentages or presence/absence data).
x	The name, age or depth of the rows of df (the samples).
x.name	A string describing the x axis (e.g. 'Sample Name', 'Age', 'Depth').
minGridCells	The minimum number of unique presence data necessary to estimate a species' climate response. Default is 20.
bin_width	The width of the bins used to correct for unbalanced climate state. Use values that split the studied climate gradient in 15-25 classes (e.g. 2°C for temperature variables). Default is 1.
shape	The imposed shape of the species pdfs. We recommend using 'normal' for temperature variables and 'lognormal' for the variables that can only take positive values, such as precipitation or aridity. Default is 'normal' for all.
npoints	The number of points to be used to fit the pdfs. Default 200.
geoWeighting	A boolean to indicate if the species should be weighting by the squareroot of their extension when estimating a genus/family level taxon-climate relationships.
climateSpaceWeighting	A boolean to indicate if the species pdfs should be corrected for the modern distribution of the climate space (default TRUE).
selectedTaxa	A data frame assigns which taxa should be used for each variable (1 if the taxon should be used, 0 otherwise). The colnames should be the climate variables' names and the rownames the taxa names. Default is 1 for all taxa and all variables.
presenceThreshold	All values above that threshold will be used in the reconstruction (e.g. if set at 1, all percentages below 1 will be set to 0 and the associated presences discarded). Default is 0.
taxWeight	One value among the following: 'originalData', 'presence/absence', 'percentages' or 'normalisation' (default).
uncertainties	A (vector of) threshold value(s) indicating the error bars that should be calculated (default both 50 and 95% ranges).

Value

A CREST object that is used to store data and information for reconstructing climate

dbRequest	<i>Connect to the gbif4crest database</i>
-----------	---

Description

Connect to the gbif4crest_02 database by accessing the server on Amazon.

Usage

```
dbRequest(request, dbname = "gbif4crest_02")
```

Arguments

request	A SQL request to be executed.
dbname	The name of the database. Default is gbif4crest_02.

Value

The result of the request.

Examples

```
# Extracting the number of taxa recorded in the database
dbRequest("SELECT count(*) FROM taxa")

# Extracting all the taxa that have at least one occurrence in South Africa.
## Not run:
southAfricaTaxa <- dbRequest(paste0(
  "SELECT DISTINCT taxa.* ",
  "FROM taxa, distrib_qdgc, geo_qdgc ",
  "WHERE taxa.taxonid=distrib_qdgc.taxonid ",
  "AND distrib_qdgc.latitude=geo_qdgc.latitude ",
  "AND distrib_qdgc.longitude=geo_qdgc.longitude ",
  "AND geo_qdgc.countryname='South Africa' ")
))
head(southAfricaTaxa)

## End(Not run)
```

eqearth_get_ext	<i>Calculates the extent of the plot in the equal earth projection.</i>
-----------------	---

Description

Calculates the extent of the plot in the equal earth projection.

Usage

```
eqearth_get_ext(ext, npoints = 15)
```

Arguments

ext	A set of coordinates.
npoints	The number of points used to draw the polygon along each dimension.

Value

The set of coordinates ext projected in equal earth.

Examples

```
eqearth_get_ext(c(-15, 50, 30, 70))
```

fit_pdfsp	<i>Fit the species pdfs.</i>
-----------	------------------------------

Description

Fit the species pdfs.

Usage

```
fit_pdfsp(climate, ccs, bin_width, shape, xrange, use_ccs = TRUE)
```

Arguments

climate	A vector of climatic values where the species is present.
ccs	A ccs object returned by calib_clim_space .
bin_width	The width of the bins used to correct for unbalanced climate state. Use values that split the studied climate gradient in 15-25 classes (e.g. 2°C for temperature variables). Default is 1.
shape	The imposed shape of the species pdfs. We recommend using 'normal' for temperature variables and 'lognormal' for the variables that can only take positive values, such as precipitation or aridity. Default is 'normal' for all.
xrange	The climate gradient upon which the pdf will be defined.
use_ccs	Boolean to indicate if the pdfsp should be corrected by the distribution of the modern climate space

Value

The pdf of the species.

Examples

```
# Creating one randomised species
climate_species <- round(stats::rnorm(50, 15, 2), 1)

# Creating one randomised climate space
climate_space <- base::sample(0:300 / 10, 4000, replace = TRUE)

ccs <- calib_clim_space(climate_space, 2)
xrange <- fit_xrange(ccs, "normal", 2)
pdfsp <- fit_pdfsp(climate_species, ccs, 2, "normal", xrange)
plot(xrange, pdfsp, type = "l")

# Testing that the area under the curve is equal to 1.
sum(pdfsp * (xrange[2] - xrange[1])) == 1
```

fit_xrange	<i>Define the climate gradient to fit the pdfs.</i>
------------	---

Description

Define the climate gradient to fit the pdfs.

Usage

```
fit_xrange(ccs, shape, bin_width, npoints = 500)
```

Arguments

ccs	A ccs object returned by calib_clim_space .
shape	The imposed shape of the species pdfs. We recommend using 'normal' for temperature variables and 'lognormal' for the variables that can only take positive values, such as precipitation or aridity. Default is 'normal' for all.
bin_width	The width of the bins used to correct for unbalanced climate state. Use values that split the studied climate gradient in 15-25 classes (e.g. 2°C for temperature variables). Default is 1.
npoints	The number of points to be used to fit the pdfs. Default 200.

Value

A regularly spaced climate gradient with npoints points.

Examples

```
# Creating one randomised climate space
climate_space <- sample(0:300 / 10, 4000, replace = TRUE)
ccs <- calib_clim_space(climate_space, 2)
xrange <- fit_xrange(ccs, "normal", 2)
head(xrange)
```

getClimateSpace	<i>Extract the distribution of the studied climate gradient(s) across the study area.</i>
-----------------	---

Description

Extract the distribution of the studied climate gradient(s) across the study area.

Usage

```
getClimateSpace(
  climate,
  xmn = -180,
  xmx = 180,
  ymn = -90,
  ymx = 90,
  continents = NA,
  countries = NA,
  realms = NA,
  biomes = NA,
  ecoregions = NA,
  dbname = "gbif4crest_02"
)
```

Arguments

climate	A vectof of the climate variables to extract.
xmn, xmx, ymn, ymx	The coordinates defining the study area.
continents	A vector of the continent names defining the study area.
countries	A vector of the country names defining the study area.
realms	A vector of the studied botanical realms defining the study area.
biomes	A vector of the studied botanical biomes defining the study area.
ecoregions	A vector of the studied botanical ecoregions defining the study area.
dbname	The name of the database. Default is gbif4crest_02.

Value

A matrix of occurrence records with the associated climate.

See Also

[accClimateVariables](#) for a list of accepted climate variable names, [accContinentNames](#) for a list of accepted continent and country names, [accRealmNames](#) for a list of accepted realm, biome and ecoregion names.

Examples

```
climate <- getClimateSpace("bio1", -90, 90, -90, 90,
  continents = "Europe",
  countries = c("Germany", "Netherlands", "Sweden"),
  realms = "Palearctic"
)
head(climate)
plot(climate[, -3], asp = 1)
```

getDistribTaxa

Extract taxonID(s) corresponding to the taxonomic description

Description

Extract all possible TaxonIDs corresponding to the provided taxonomical description, which can be at the family, the genus or the species levels.

Usage

```
getDistribTaxa(
  taxIDs,
  climate,
  xmn = -180,
  xmx = 180,
  ymn = -90,
  ymx = 90,
  continents = NA,
  countries = NA,
  realms = NA,
  biomes = NA,
  ecoregions = NA,
  dbname = "gbif4crest_02"
)
```

Arguments

taxIDs	A vector of accepted Taxa IDs (as returned by getTaxonID).
climate	A vectof of the climate variables to extract.
xmn, xmx, ymn, ymx	The coordinates defining the study area.
continents	A vector of the continent names defining the study area.
countries	A vector of the country names defining the study area.
realms	A vector of the studied botanical realms defining the study area.
biomes	A vector of the studied botanical biomes defining the study area.
ecoregions	A vector of the studied botanical ecoregions defining the study area.
dbname	The name of the database. Default is gbif4crest_02.

Value

A matrix of occurrence records with the associated climate.

See Also

[getTaxonID](#) for taxIDs, [accClimateVariables](#) for a list of accepted climate variable names, [accContinentNames](#) for a list of accepted continent and country names, [accRealmNames](#) for a list of accepted realm, biome and ecoregion names.

Examples

```
taxIDs <- getTaxonID("Zamiaceae", "Ceratozamia")
distrib <- getDistribTaxa(taxIDs, "bio1", -90, 90, -90, 90,
  continents = "Europe",
  countries = c("Germany", "Netherlands", "Sweden"),
  realms = "Palearctic"
)
distrib
```

getTaxonID

*Extract taxonID(s) corresponding to the taxonomic description***Description**

Extract all possible TaxonIDs corresponding to the provided taxonomical description, which can be at the family, the genus or the species levels.

Usage

```
getTaxonID(
  family,
  genus = "",
  species = "",
  taxaType = 1,
  dbname = "gbif4crest_02"
)
```

Arguments

family	The name of the family.
genus	The name of the genus.
species	The name of the species.
taxaType	A numerical index (between 1 and 6) to define the type of palaeoproxy used: 1 for plants, 2 for beetles, 3 for foraminifers, 4 for diatoms, 5 for chironomids and 6 for rodents. The example dataset uses taxaType=0. Default is 1.
dbname	The name of the database. Default is gbif4crest_02.

Value

A vector of unique taxonIDs.

Examples

```
getTaxonID("Zamiaceae")
getTaxonID("Zamiaceae", "Ceratozamia")
getTaxonID("Zamiaceae", "Ceratozamia", taxaType = 2)
```

getTaxonomy

Extract taxonID(s) corresponding to the taxonomic description

Description

Extract all possible TaxonIDs corresponding to the provided taxonomical description, which can be at the family, the genus or the species levels.

Usage

```
getTaxonomy(
  family = "",
  genus = "",
  species = "",
  taxaType = 1,
  depth.out = 8,
  dbname = "gbif4crest_02"
)
```

Arguments

family	The name of the family.
genus	The name of the genus.
species	The name of the species.
taxaType	A numerical index (between 1 and 5) to define the type of palaeoproxy used: 1 for plants, 2 for beetles, 3 for foraminifers, 4 for diatoms, 5 for chironomids and 6 for rodents.
depth.out	The taxonomic resolution of the output table. 1 for Kingdom, 2 for phylum, 3 for class_name, 4 for order_name, 5 for family, 6 for genus, 7 for species and 8 to also include the taxonID.
dbname	The name of the database. Default is gbif4crest_02.

Value

A vector of unique taxonIDs.

Examples

```
getTaxonomy("Zamiaceae")
getTaxonomy(genus="Ceratozamia", depth.out=8)
getTaxonomy("Zamiaceae", "Ceratozamia", taxaType = 2)
```

isColourStr	<i>Test if R can interpret a string as a colour</i>
-------------	---

Description

Test if R can interpret a string as a colour

Usage

```
isColourStr(col)
```

Arguments

col	The string to be tested.
-----	--------------------------

Value

A boolean value, TRUE if col is a valid colour, FALSE otherwise

Examples

```
isColourStr('black')
isColourStr('blakc')
```

loo	<i>Connect to the gbif4crest database</i>
-----	---

Description

Connect to the gbif4crest_02 database by accessing the server on Amazon.

Usage

```
loo(x, verbose = TRUE)
```

Arguments

x	a crestObj produced by the crest.reconstruct() or crest() functions.
verbose	A boolean to print non-essential comments on the terminal (default TRUE).

Value

A crest() object containing the reconstructions and all the associated data.

Examples

```
data(crest_ex)
data(crest_ex_pse)
data(crest_ex_selection)
recons <- crest(
  df = crest_ex, pse = crest_ex_pse, taxaType = 0,
  climate = c("bio1", "bio12"), bin_width = c(2, 20),
  shape = c("normal", "lognormal"),
  selectedTaxa = crest_ex_selection, dbname = "crest_example"
)
recons <- loo(recons)
recons$reconstructions$bio12$loo
plot_loo(recons)
```

M1	<i>A shapefile of the world's country borders.</i>
----	--

Description

A shapefile of the world's country borders.

Usage

M1

Format

An object of class SpatialPolygonsDataFrame with 252 rows and 1 columns.

meanPositiveValues	<i>Calculate the mean of all stricly positive values.</i>
--------------------	---

Description

Calculate the mean of all stricly positive values.

Usage

meanPositiveValues(x)

Arguments

x A vector of values.

Value

The average of all the positive values. Returns NaN is no positive values are found.

Examples

```
meanPositiveValues(-10:10)
```

normalise	<i>Convert data into presence/absence data.</i>
-----------	---

Description

Convert data into presence/absence data.

Usage

```
normalise(df, threshold = 2, col2convert = 2:ncol(df))
```

Arguments

df	The dataframe containing the data to convert.
threshold	The threshold that defines presence (presence if \geq threshold)
col2convert	A vector of the columns to convert. Default is all the columns but the first, which contains an age, a depth or a sampleID.

Value

A vector of unique taxonIDs.

Examples

```
df <- data.frame(matrix(1:25, ncol = 5))
colnames(df) <- paste(rep("col", 5), 1:5, sep = "")
convert2presenceAbsence(df, threshold = 15)
convert2presenceAbsence(df, col2convert = 3:5)
```

plot.crestObj	<i>Plot the reconstructions.</i>
---------------	----------------------------------

Description

Plot the reconstructions and their uncertainties

Usage

```
## S3 method for class 'crestObj'
plot(
  x,
  climate = x$parameters$climate,
  uncertainties = x$parameters$uncertainties,
  optima = TRUE,
  xlim = NA,
  ylim = NA,
  save = FALSE,
  loc = getwd(),
  ...
)
```

Arguments

x	A crestObj produced by the crest.reconstruct() or crest() functions.
climate	The climate variables to plot (default is all the reconstructed variables from x)
uncertainties	A (vector of) threshold value(s) indicating the error bars that should be calculated (default are the values stored in x).
optima	A boolean to indicate whether to plot the optimum (TRUE) or the mean (FALSE) estimates.
xlim	the x limits (x1, x2) of the plot. Note that x1 > x2 is allowed and leads to a 'reversed axis'. The default value, NULL, indicates that the range of the finite values to be plotted should be used.
ylim	the y limits of the plot.
save	A boolean to indicate if the diagram should be saved as a pdf file. Default is FALSE.
loc	An absolute or relative path that indicates the folder where the diagram(s) should be saved. Also used to specify the name of the file. Default: the file is saved in the working directory with a file created for each variable as variable.pdf.
...	other graphical parameters (see par and section 'Details' below).

plot_climateSpace	<i>Plot the studied climate space.</i>
-------------------	--

Description

Plot the studied climate space.

Usage

```
plot_climateSpace(
  x,
  save = FALSE,
  loc = "Climate_space.pdf",
  width = 7.48,
  height = min(9, 3 * length(x$parameters$climate)),
  y0 = 0.3
)
```

Arguments

x	A crestObj generated by either the crest.calibrate(), crest.reconstruct() or crest() functions.
save	A boolean to indicate if the diagram should be saved as a pdf file. Default is FALSE.
loc	An absolute or relative path that indicates where the diagram should be saved. Also used to specify the name of the file. Default: the file is saved in the working directory under the name Climate_space.pdf.
width	The width of the output file in inches (default 7.48in ~ 19cm).
height	The height of the output file in inches (default 3in ~ 7.6cm per variables).
y0	The space to allocate to each title (default 0.3in ~ 0.76 cm)

Examples

```
## Not run:
data(crest_ex_pse)
data(crest_ex_selection)
x <- crest.get_modern_data(
  pse = crest_ex_pse, taxaType = 0,
  climate = c("bio1", "bio12"),
  selectedTaxa = crest_ex_selection, dbname = "crest_example"
)
x <- crest.calibrate(x,
  geoWeighting = TRUE, climateSpaceWeighting = TRUE,
  bin_width = c(2, 20), shape = c("normal", "lognormal")
)
plot_climateSpace(x)

## End(Not run)
```

plot_diagram

Plot stratigraphic data as polygons or barplots.

Description

This function plots stratigraphic data either as polygons or bars.

Usage

```
plot_diagram(
  x,
  bars = FALSE,
  col = "black",
  amplif = 5,
  save = FALSE,
  loc = "Diagram.pdf",
  width = 3.54,
  height = 9,
  yax_incr = 5,
  bar_width = 1,
  xlim = NA,
  tickAtSample = TRUE,
  col_pos = "black",
  col_neg = "grey80",
  title = NA
)
```

Arguments

- | | |
|------|--|
| x | A data frame of the data to plot (first column with age or depth) and the taxa in the following columns. x can also be a crestObj. |
| bars | A boolean that indicates if the data should be plotted as polygons (default: bars=FALSE) or vertical bars (bars=TRUE). |

col	Colours to be used for the polygons. If the number of colours does not match the number of taxa, colors will be recycled.
amplif	A factor the show exaggeration on the diagram. Only for polygon plot. Default 5.
save	A boolean to indicate if the diagram should be saved as a pdf file. Default is FALSE.
loc	An absolute or relative path that indicates where the diagram should be saved. Also used to specify the name of the file. Default: the file is saved in the working directory under the name Diagram.pdf.
width	The width of the output file in inches (default 3.54in ~ 9cm).
height	The height of the output file in inches (default 9in ~ 23cm).
yax_incr	Graphical parameters describing the increment size on the y-axis (default 5).
bar_width	Width of the bars of the barplot (default 1).
xlim	The range covered by the x-axis. Canbe adjusted to get round numbers on the x-ais. If smaller than the range overed by the data, the data will be truncated (default: range of the data).
tickAtSample	Boolean that indicates whether a tick mark should be added on the x-axis at the location of each sample (default TRUE).
col_pos	Graphical parameter for the barplot. Colour of all the positive values (default black).
col_neg	Graphical parameter for the barplot. Colour of all the negative values (default light grey).
title	Name to be added on top of the plot (default NA).

Examples

```
data(crest_ex)
plot_diagram(crest_ex, bars=TRUE, col='black', bar_width=0.8)
plot_diagram(crest_ex, col=1:7, tickAtSample=FALSE)
## Not run:
plot_diagram(crest_ex, save=TRUE, loc='testDiagram.pdf',
             bars=TRUE, col_pos='cornflowerblue', col_neg='darkgreen',
             bar_width=0.8, xlim=c(3,15))

## End(Not run)
```

plot_loo

Plot stratigraphic data as polygons or barplots.

Description

This function plots stratigraphic data either as polygons or bars.

Usage

```
plot_loo(
  x,
  optima = TRUE,
  save = FALSE,
  loc = "Diagram_loo",
  width = 3.54,
  height = 9,
  yax_incr = NA,
  bar_width = 1,
  xlim = NA,
  tickAtSample = FALSE,
  col_pos = "black",
  col_neg = "grey80",
  title = NA
)
```

Arguments

x	A data frame of the data to plot (first column with age or depth) and the taxa in the following columns. x can also be a crestObj.
optima	A boolean to indicate whether to plot the optimum (TRUE) or the mean (FALSE) estimates.
save	A boolean to indicate if the diagram should be saved as a pdf file. Default is FALSE.
loc	An absolute or relative path that indicates where the diagram should be saved. Also used to specify the name of the file. Default: the file is saved in the working directory under the name Diagram.pdf.
width	The width of the output file in inches (default 3.54in ~ 9cm).
height	The height of the output file in inches (default 9in ~ 23cm).
yax_incr	Graphical parameters describing the increment size on the y-axis (default 5).
bar_width	Width of the bars of the barplot (default 1).
xlim	The range covered by the x-axis. Can be adjusted to get round numbers on the x-axis. If smaller than the range covered by the data, the data will be truncated (default: range of the data).
tickAtSample	Boolean that indicates whether a tick mark should be added on the x-axis at the location of each sample (default TRUE).
col_pos	Graphical parameter for the barplot. Colour of all the positive values (default black).
col_neg	Graphical parameter for the barplot. Colour of all the negative values (default white).
title	Name to be added on top of the plot (default NA).

Examples

```
## Not run:
data(crest_ex)
data(crest_ex_pse)
data(crest_ex_selection)
```

```

recons <- crest(
  df = crest_ex, pse = crest_ex_pse, taxaType = 0,
  climate = c("bio1", "bio12"), bin_width = c(2, 20),
  shape = c("normal", "lognormal"),
  selectedTaxa = crest_ex_selection, dbname = "crest_example"
)
recons <- loo(recons)
plot_loo(recons, yax_incr=c(0.5, 50), bar_width=0.8,
         col_pos=c('blue', 'cornflowerblue'), col_neg=c('red', 'goldenrod3'))

## End(Not run)

```

plot_map_eqearth

Plots raster data in equal earth projection.

Description

Plots raster data in equal earth projection.

Usage

```

plot_map_eqearth(
  dat,
  ext = raster::extent(dat),
  zlim = range(raster::values(dat), na.rm = TRUE),
  col = viridis::viridis(20),
  brks.pos,
  brks.lab = brks.pos,
  npoints = 15,
  nlines = 9,
  title = "",
  colour_scale = TRUE,
  top_layer = NA
)

```

Arguments

dat	The raster data to plot.
ext	The extent to use to plot the data. (default is extent of dat)
zlim	The range of the values to plot. (default is estimated from dat)
col	The color gradient to use. (default is viridis)
brks.pos	The position where to draw tick marks on the legend
brks.lab	The labels to add where the tickmarks are draw (default is tickmarks position)
npoints	The number of points used to draw the polygons and lines along each dimension. (default is 15 for a smooth result)
nlines	The number of coordinate lines to add in the background (default is 9)
title	A description title (default is empty)
colour_scale	A boolean to add the colour scale to the plot (default TRUE).
top_layer	A raster to overlay on top of the map (e.g. a distribution).

plot_taxaCharacteristics

Plot the studied climate space.

Description

Plot the studied climate space.

Usage

```
plot_taxaCharacteristics(
  x,
  taxanames = x$inputs$taxa.name,
  save = FALSE,
  loc = "taxaCharacteristics.pdf",
  width = 7.48,
  w0 = 0.3,
  height = min(9, 3 * length(x$parameters$climate)),
  h0 = 0.3
)
```

Arguments

x	A crestObj generated by either the crest.calibrate(), crest.reconstrut() or crest() functions.
taxanames	A list of taxa to use for the plot (default is all the recorded taxa).
save	A boolean to indicate if the diagram should be saved as a pdf file. Default is FALSE.
loc	An absolute or relative path that indicates where the diagram should be saved. Also used to specify the name of the file. Default: the file is saved in the working directory under the name Climate_space.pdf.
width	The width of the output file in inches (default 7.48in ~ 19cm).
w0	The width of the left column with the names.
height	The height of the output file in inches (default 3in ~ 7.6cm per variables).
h0	The vertical space used for the x-axes.

Examples

```
## Not run:
data(crest_ex_pse)
data(crest_ex_selection)
x <- crest.get_modern_data(
  pse = crest_ex_pse, taxaType = 0,
  climate = c("bio1", "bio12"),
  selectedTaxa = crest_ex_selection, dbname = "crest_example"
)
x <- crest.calibrate(x,
  geoWeighting = TRUE, climateSpaceWeighting = TRUE,
  bin_width = c(2, 20), shape = c("normal", "lognormal")
)
```

```
plot_taxaCharacteristics(x)  
## End(Not run)
```

Index

* datasets

- crest_ex, [14](#)
- crest_ex_pse, [15](#)
- crest_ex_selection, [15](#)
- M1, [26](#)

accClimateVariables, [2](#), [8](#), [9](#), [12](#), [16](#), [21](#), [23](#)

accContinentNames, [3](#), [21](#), [23](#)

accRealmNames, [3](#), [21](#), [23](#)

calib_clim_space, [4](#), [19](#), [20](#)

cite_GBIF, [4](#)

close_db_connection, [5](#)

connect_online, [5](#)

convert2percentages, [6](#)

convert2presenceAbsence, [7](#)

copy_crest, [7](#)

crest, [8](#)

crest.calibrate, [10](#)

crest.get_modern_data, [11](#), [11](#)

crest.reconstruct, [13](#)

crest_ex, [14](#)

crest_ex_pse, [15](#)

crest_ex_selection, [15](#)

crestObj, [16](#)

dbRequest, [18](#)

eqearth_get_ext, [18](#)

finite, [28](#)

fit_pdfsp, [4](#), [19](#)

fit_xrange, [20](#)

get_pse, [9](#), [12](#), [16](#)

getClimateSpace, [21](#)

getDistribTaxa, [22](#)

getTaxonID, [22](#), [23](#), [23](#)

getTaxonomy, [24](#)

graphical parameters, [28](#)

isColourStr, [25](#)

loo, [25](#)

M1, [26](#)

meanPositiveValues, [26](#)

normalise, [27](#)

par, [28](#)

plot.crestObj, [27](#)

plot_climateSpace, [28](#)

plot_diagram, [29](#)

plot_loo, [30](#)

plot_map_eqearth, [32](#)

plot_taxaCharacteristics, [33](#)