Package ‘Mapinguari’

September 30, 2019

**Type** Package

**Title** Process-Based Biogeographical Analysis

**Version** 1.0.0

**Description** Facilitates the incorporation of biological processes in biogeographical analyses. It offers conveniences in fitting, comparing and extrapolating models of biological processes such as physiology and phenology. These spatial extrapolations can be informative by themselves, but also complement traditional correlative species distribution models, by mixing environmental and process-based predictors.

**Depends** R (>= 3.5)

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**URL** http://github.com/gabrielhoc/Mapinguari

**BugReports** http://github.com/gabrielhoc/Mapinguari/issues

**Suggests** EcoHydRology, geosphere, mgcv

**Imports** dplyr, magrittr, parallel, raster, rgdal, rlang, stringr, testthat

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Author** Gabriel Caetano [aut, cre], Juan Santos [aut], Barry Sinervo [aut]

**Maintainer** Gabriel Caetano <gabrielhoc@gmail.com>

**Repository** CRAN

**Date/Publication** 2019-09-30 16:40:02 UTC
\textbf{R topics documented:}

- **AETFUN** .............................................................. 2
- clean_points ......................................................... 3
- daylengthFUN ....................................................... 4
- df_alt ............................................................... 4
- df_prec ............................................................. 5
- df_tmax ............................................................ 5
- df_tmin ............................................................. 6
- get\_predict ........................................................ 6
- get\_rasters ........................................................ 7
- Mapinguari .......................................................... 8
- multi\_extract ...................................................... 9
- PC2FmFUN .......................................................... 10
- PETFUN ........................................................... 10
- rhFUN ............................................................... 11
- sin\_h ............................................................... 12
- sradFUN ............................................................ 12
- transform\_rasters ................................................ 13
- TtorquatusBreeding ............................................... 14
- TtorquatusDistribution ......................................... 15
- TtorquatusGradient ................................................ 15
- TtorquatusOperative ............................................. 16
- TtorquatusPerformance ......................................... 16

\textbf{Index} ............................................................ 18

\begin{tabular}{ll}
\hline
AETFUN & \emph{Generates Actual EvapoTranspiration rasters} \\
\hline
\end{tabular}

\textbf{Description}

\texttt{AETFUN} Applies Duncan Golicher’s Bucket model to Potential EvapoTranspiration and precipitation rasters in order to get Actual Evapotranspiration estimates for an area (Golicher, 2012).

\textbf{Usage}

\begin{verbatim}
AETFUN(PET, prec, separator = "_")
\end{verbatim}

\textbf{Arguments}

- **PET** .......................................................... RasterStack with 12 layers. Total month Potential EvapoTranspiration rasters.
- **prec** ......................................................... RasterStack with 12 layers. Total month precipitation rasters.
- **separator** .................................................. character. Character that separates variable names, and scenarios.

\textbf{Value}

Returns a RasterLayer with estimates of Actual EvapoTranspiration in millimeters.
clean_points

References


Examples

```r
PET <- PETFUN(df_tmax, df_tmin, df_alt)
AETFUN(PET, df_prec)
```

<table>
<thead>
<tr>
<th>clean_points</th>
<th>Clean occurrence records</th>
</tr>
</thead>
</table>

Description

clean_points Eliminates species occurrence records that are too close to each other or at undesired locations.

Usage

clean_points(coord, merge_dist, coord_col = c("Lon", "Lat"),
filter_layer = NULL, na.rm = FALSE)

Arguments

coord data.frame. Data frame containing longitudes (Lon) and latitudes (Lat) of occurrence records of a species.
merge_dist numeric. Maximum distance between points to be merged, in meters.
coord_col vector of strings or integers. If x has more than two columns, indicate the name or position of longitude and latitude columns
filter_layer RasterLayer. Binary raster with 1 representing the regions where records should be kept and 0 the regions where they should be eliminated.
na.rm logical. if TRUE, remove lines with NA in any coordinate.

Value

Data frame with remaining longitudes and latitudes.

Examples

```r
TtorquatusDistribution_clean <-
clean_points(coord = TtorquatusDistribution,
merge_dist = 20000,
filter_layer = !is.na(df_alt))
```
daylengthFUN

Gets average month day lengths for an area

Description

daylengthFUN Generates surfaces with information on day length for each month across an area.

Usage

daylengthFUN(reference_raster)

Arguments

reference_raster

RasterStack or RasterLayer. Any raster representative of the area you want day lengths to. The day length raster will have the same projection, resolution and extension.

Value

Returns a RasterStack with 12 layers, one for each month, containing information on the duration of the day at each pixel.

Examples

daylengthFUN(df_alt)

df_alt

Altitude for Distrito Federal, Brazil

Description

CGIAR-SRTM Altitude data (Jarvis et al, 2010) resampled to 2.5 minute resolution for the Distrito Federal region in central Brazil from 1960 to 1990.

Usage

df_alt

Format

A raster layer with 13 rows and 24 columns

References

df_prec

**Precipitation for Distrito Federal, Brazil**

**Description**
Average monthly precipitation data from WorldClim 1.4 (Hijmans et al, 2005, licensed under CC BY-SA 4.0) in 2.5 minute resolution for the Distrito Federal region in central Brazil from 1960 to 1990.

**Usage**
df_prec

**Format**
A raster stack with 12 layers, 13 rows and 24 columns

**References**
Hijmans et al (2005) <doi.org/10.1002/joc.1276>

---

df_tmax

**Maximum temperature for Distrito Federal, Brazil**

**Description**
Average monthly maximum temperature data from WorldClim 1.4 (Hijmans et al, 2005, licensed under CC BY-SA 4.0) in 2.5 minute resolution for the Distrito Federal region in central Brazil from 1960 to 1990.

**Usage**
df_tmax

**Format**
A raster stack with 12 layers, 13 rows and 24 columns

**References**
Hijmans et al (2005) <doi.org/10.1002/joc.1276>
**df_tmin**  
*Minimum temperature for Distrito Federal, Brazil*

**Description**

Average monthly minimum temperature data from WorldClim 1.4 (Hijmans et al, 2005, licensed under CC BY-SA 4.0) in 2.5 minute resolution for the Distrito Federal region in central Brazil from 1960 to 1990.

**Usage**

```r
df_tmin
```

**Format**

A raster stack with 12 layers, 13 rows and 24 columns

**References**

Hijmans et al (2005) <doi.org/10.1002/joc.1276>

---

**get_predict**  
*Creates vectorized predict functions from models.*

**Description**

get_predict Takes inputed models and create vectorized functions able to get the model predictions for any value inputed. Also outputs a table comparing models. nls gam glm lm randomForest gbm gls bam

**Usage**

```r
get_predict(models, separator = " ", ...)
```

**Arguments**

- `models`  
  list. List with models to create the prediction function. The model objects must have methods for function ‘predict’.
- `separator`  
  character. Character that separates variable names, years and scenarios.
- `...`  
  additional arguments to be passed to predict function (specific for the method of the models supplied).

**Value**

Returns a list of vectorized functions that get predictions for the models inputted. The functions generated do not perform lazy evaluation, the user must be explicit
get_rasters

Examples

library(mgcv)

perf_no_size <-
gamm(performance ~ s(temp, bs = "cs"),
     random = list(id = ~ 1),
     data = TtorquatusPerformance)

perf_size <-
gamm(performance ~ s(temp, bs = "cs") + size,
     random = list(id = ~ 1),
     data = TtorquatusPerformance)

perf_functions <- get_predict(list(perf_s = perf_size,
                                  perf_ns = perf_no_size),
                              type = "response")

perf_nsFUN <- perf_functions$perf_ns
perf_sFUN <- perf_functions$perf_s

perf_nsFUN(temp = 30)
perf_sFUN(temp = 30, size = 70)
perf_nsFUN(temp = 30:35)
perf_sFUN(temp = 30:35, size = 70:75)

get_rasters

Retrieve and organize spatial rasters.

Description

get_rasters Loads rasters from directory and returns them in an organized list of specified scenarios.

Usage

get_rasters(var = NULL, scenario = NULL, raster_path = NULL,
            ext = c(-180, 180, -60, 90), coord_col = c("Lon", "Lat"),
            margin = 0, separator = ".")

Arguments

var character. Names of variables to be loaded.
scenario character. Names of scenarios for the variables.
raster_path character. Path to folder with raster files. See writeFormats for supported formats
Mapinguari provides solutions for incorporating biological processes in biogeographical analysis.

**Mapinguari functions**

- `clean_points`
- `get_predict`
- `get_rasters`
- `multi_extract`
- `transform_rasters`
- `daylengthFUN`
- `PETFUN`
- `AETFUN`
- `sradFUN`
- `sin_h`
- `PC2FMFUN`
- `rhFUN`
multi_extract

*Gets values from multiple rasters.*

**Description**

`multi_extract` Extract values of multiple spatial rasters for a set of geographical coordinates.

**Usage**

```r
multi_extract(raster_path, coord, folders = NULL, files = NULL, layers = NULL, ncores = 1)
```

**Arguments**

- `raster_path`: character. Path to the folder with raster folders.
- `coord`: data.frame or matrix. Longitude and Latitude from where to extract raster values.
- `folders`: character. folders from which to get rasters for extraction. If NULL, all folders are selected.
- `files`: numeric. Index for raster files to be extracted from each folder. If NULL, all files are selected.
- `layers`: numeric. Index for layers to be extracted from each raster file. If NULL, all layers are selected.
- `ncores`: integer. Number of cores to use in parallel processing.

**Value**

Data frame with extracted values from multiple rasters

**Examples**

```r
# replace rasterpath with the directory on your computer containing worldclim data

temp_pres <-
  multi_extract(raster_path = system.file("extdata/wc-10m", package="Mapinguari"),
                coord = TtorquatusDistribution[-1],
                folders = c("tmax_present", "tmin_present"))
```
PC2FMFUN

Guyette’s Physical Chemistry Fire Frequency Model (PC2FM)

Description

PC2FMFUN Applies Guyette’s Fire frequency model (Guyette et al, 2012).

Usage

PC2FMFUN(prec, temp, alt)

Arguments

prec numeric. Precipitation raster.
temp numeric. Temperature raster.
alt numeric. Altitude raster.

Value

numeric. Fire frequency based on physical chemical factors

References


Examples

PC2FMFUN(50, 25, 1000)

PETFUN

Spatialize PET_fromTemp function from package EcoHydrology

Description

PETFUN Gets Potential EvapoTranspiration (PET) rasters from maximum temperature, minimum temperature and altitude rasters by applying function PET_fromTemp from package EcoHydrology.

Usage

PETFUN(tmax, tmin, alt, separator = "_")
**rhFUN**

**Arguments**

- `tmax` Raster* object. Maximum temperature raster.
- `tmin` Raster* object. Minimum temperature raster.
- `alt` Raster* object. Altitude raster.
- `separator` character. Character that separates variable names, and scenarios.

**Value**

Returns a RasterLayer with estimates of Potential EvapoTranspiration in milimeters.

**Examples**

```r
PET <- PETFUN(df_tmax, df_tmin, df_alt)
```

---

**rhFUN**

*Relative humidity from temperature and water vapor pressure*

**Description**

`rhFUN` Calculates relative humidity from air temperature in Celsius and water vapor pressure in milibars.

**Usage**

`rhFUN(temp, vapor)`

**Arguments**

- `temp` numeric. temperature in celsius
- `vapor` numeric. water vapor pressure in milibars

**Value**

a vector of relative humidity values, in decimal.

**Examples**

```r
rhFUN(25, 20)
rhFUN(25:40, 20:35)
```
**Description**

*sin_h* Simulates daily variation in temperature and counts amount of time above a temperature threshold, as seen in Sinervo et al (2010).

**Usage**

\[
\text{sin}_h(t_{\text{max}}, t_{\text{min}}, \text{thrs}, \text{res})
\]

**Arguments**

- `t_{\text{max}}`: Raster* object. Maximum temperature raster.
- `t_{\text{min}}`: Raster* object. Minimum temperature raster.
- `\text{thrs}`: numeric. Temperature threshold in same unit as rasters.
- `\text{res}`: numeric. time resolution in parts of hour.

**Value**

numeric. Amount of time in hours above temperature threshold in simulated daily temperature variation.

**References**


**Examples**

\[
\text{sin}_h(28, 10, 23, 3)
\]

---

**Description**

*sradFUN* Gets estimates of solar radiation for an area, based on maximum and minimum temperatures and altitude rasters by applying function *Solar* from package *EcoHydrology*

**Usage**

\[
s\text{radFUN}(\text{alt, } t_{\text{max}}, t_{\text{min}}, \text{separator = "{\_\_}"})
\]
Arguments

- alt: Raster* object. Altitude raster.
- tmax: Raster* object. Maximum temperature raster.
- tmin: Raster* object. Minimum temperature raster.
- separator: character. Character that separates variable names, and scenarios.

Value

Returns a RasterLayer with estimates of Solar in kiloJoules by square meter by day.

Examples

```r
srad <- sradFUN(df_alt, df_tmax, df_tmin)
```

transform_rasters

Transform raster values using custom calls.

Description

transform_rasters Applies custom expressions to transform the values of spatial rasters in a stack, taking into account temporal repetition of those rasters.

Usage

transform_rasters(raster_stack, separator = " ", ncores = 1, ...)

Arguments

- raster_stack: RasterStack. Stack with environmental layers.
- separator: character. Character that separates variable names, years and scenarios.
- ncores: integer. Number of cores to use in parallel processing.
- ...: New rasters created.

Value

Returns a RasterStack with layers for the predictions required.

Examples

# You can apply any function to subsets of rasters in the stack,
# by selecting the layers with double brackets.

transform_rasters(raster_stack = df_tmax,
         total_1sem = sum(tmax[1:6]),
         mean_1sem = mean(tmax[1:6]),
         sd_1sem = sd(tmax[1:6]))
Breeding status of *Tropidurus torquatus* lizards at 15 locations in Brazil during each month of the year.

**Description**

A dataset containing information on if *Tropidurus torquatus* is breeding or not at specific locations and times.

**Usage**

TtorquatusBreeding

**Format**

A data frame with 15 rows and 14 variables:

- **Lon**: Longitude of occurrence records in decimal degrees
- **Lat**: Latitude of occurrence records in decimal degrees
- **January**: Binary breeding status at each location for the month of January, 1 means breeding, 0 means not breeding
- **February**: Binary breeding status at each location for the month of February, 1 means breeding, 0 means not breeding
- **March**: Binary breeding status at each location for the month of March, 1 means breeding, 0 means not breeding
- **April**: Binary breeding status at each location for the month of April, 1 means breeding, 0 means not breeding
- **May**: Binary breeding status at each location for the month of May, 1 means breeding, 0 means not breeding
- **June**: Binary breeding status at each location for the month of June, 1 means breeding, 0 means not breeding
- **July**: Binary breeding status at each location for the month of July, 1 means breeding, 0 means not breeding
- **August**: Binary breeding status at each location for the month of August, 1 means breeding, 0 means not breeding
- **September**: Binary breeding status at each location for the month of September, 1 means breeding, 0 means not breeding
- **October**: Binary breeding status at each location for the month of October, 1 means breeding, 0 means not breeding
- **November**: Binary breeding status at each location for the month of November, 1 means breeding, 0 means not breeding
- **December**: Binary breeding status at each location for the month of December, 1 means breeding, 0 means not breeding
**TtorquatusDistribution**

359 occurrence records of *Tropidurus torquatus* in Brazil

### Description

A dataset containing *Tropidurus torquatus* distribution records

### Usage

TtorquatusDistribution

### Format

A data frame with 359 rows and 3 variables:

- **species**: species name
- **Lon**: Longitude of occurrence point
- **Lat**: Latitude of occurrence point ...

**TtorquatusGradient**

* *Tropidurus torquatus* body temperatures at temperature gradient experiments.*

### Description

A dataset containing 3443 body temperature records of 52 *Tropidurus torquatus* from 6 localities at temperature gradients.

### Usage

TtorquatusGradient

### Format

A data frame with 3443 rows and 3 variables:

- **id**: individual identity of the lizard perform
- **temp**: lizard body temperature at the moment of the trial, in Celsius
- **site**: place where lizard was collected ...
**TtorquatusOperative**  
Operative temperatures of multiple microhabitats at 6 localities in Brazil from 2014 to 2016.

**Description**
A dataset containing operative temperatures of multiple microhabitats at 6 localities in Brazil from 2014 to 2016.

**Usage**
TtorquatusOperative

**Format**
A data frame with 915684 rows and 13 variables:
- site: place where temperatures were collected
- description: description of site
- Lon: Longitude of sampling point
- Lat: Latitude of sampling point
- temp: temperature at microhabitat, in degrees Celsius
- microhabitat: microhabitat sampled
- year: year of sampling
- month: month of sampling
- day: day of sampling
- hour: hour of sampling
- minute: minute of sampling
- t_air_max: maximum air temperature of the day at nearest weather station, in degrees Celsius...

---

**TtorquatusPerformance**  
Running speed achieved by 72 *Tropidurus torquatus* lizards in 274 trials under different temperatures.

**Description**
A dataset containing 274 running speed trials of *Tropidurus torquatus* lizards under different temperatures, the temperatures of the runs, individual identities for each lizard, body size of each individual and the site where they were captured.

**Usage**
TtorquatusPerformance
Format

A data frame with 274 rows and 6 variables:

- **species** species name
- **id** individual identity of the lizard perform
- **temp** lizard body temperature at the moment of the trial, in Celsius
- **performance** maximum running speed at trial, in meters per second
- **size** lizard body size, in centimeters
- **site** place where lizard was collected ...
Index

*Topic datasets
  df_alt, 4
  df_prec, 5
  df_tmax, 5
  df_tmin, 6
  TtorquatusBreeding, 14
  TtorquatusDistribution, 15
  TtorquatusGradient, 15
  TtorquatusOperative, 16
  TtorquatusPerformance, 16

AETFUN, 2, 8

clean_points, 3, 8

daylengthFUN, 4, 8
df_alt, 4
df_prec, 5
df_tmax, 5
df_tmin, 6

generate, 6, 8

generate_rasters, 7, 8

Mapinguari, 8
Mapinguari-package (Mapinguari), 8
multi_extract, 8, 9

PC2FMFUN, 8, 10
PETFUN, 8, 10

rhFUN, 8, 11

sin_h, 8, 12

sradFUN, 8, 12

transform_rasters, 8, 13
TtorquatusBreeding, 14
TtorquatusDistribution, 15
TtorquatusGradient, 15
TtorquatusOperative, 16
TtorquatusPerformance, 16

writeFormats, 7