Package: pdindicatoR (via r-universe)

February 17, 2025

Title Calculate and visualize a phylogenetic diversity indicators based on species occurence data cubes

Version 0.0.2

Description A package to calculate and produce a map of phylogenetic diversity scores for each grid cell in the input species occurences data cube, and calculate the percentage of high PD cells within currently protected areas. You provide a phylogenetic tree, the datacube for the corresponding species is downloaded from GBIF (or user-uploaded), and the package does the rest.

License MIT + file LICENSE

URL https://github.com/b-cubed-eu/pdindicatoR, https://b-cubed-eu.github.io/pdindicatoR/

Imports ape, dplyr, gdalUtilities, ggplot2, rlang, rnaturalearth, rnaturalearthdata, rotl, sf, shiny, stringr

Suggests purrr, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev make libicu-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev zlib1g-dev

Repository https://b-cubed-eu.r-universe.dev

RemoteUrl https://github.com/b-cubed-eu/pdindicatoR

RemoteRef HEAD

RemoteSha 8e86f2c1215944abcaab7a73d89be74dcda9e677

Contents

`aggregate_cube` *Aggregate datacube over grid cell to create new dataframe with species list per grid*

Description

This function aggregates a provided datacube over grid cell id, so that a new datacube is outputted with 3 variables that contain the lists of species that are observed for each grid cell (as `speciesKeys`, `ott_id`'s and names).

Usage

```
aggregate_cube(mcube, timegroup = NULL)
```

Arguments

`mcube`	An occurrence datacube with appended `ott_id`'s, as produced by the `append_ott_id()` function
`timegroup`	An integer, representing the number of years by which you want to group your occurrence data

Value

A dataframe with for each grid cell

Examples

```
ex_data <- retrieve_example_data()
mcube <- append_ott_id(ex_data$tree, ex_data$cube, ex_data$matched_nona)
mcube <- dplyr::filter(mcube, !is.na(ott_id))
aggr_cube <- aggregate_cube(mcube)
```

`append_ott_id` *Append ott id's to cube*

Description

This function uses the table produced by the `taxonmatch()` function to create a linking table, and then append the `ott_id`'s as a new field to the occurrence cube.

Usage

```
append_ott_id(tree, cube, matched)
```

Arguments

`tree`	An object of class `'phylo'`, a phylogenetic tree in Newick format that was parsed by `ape::read_tree()`
`cube`	A dataframe with for selected taxa, the number of occurrences per taxa and per grid cell
`matched`	A dataframe, returned by running the function `taxonmatch()` on a phylogenetic tree, which contains the tip labels of the tree and their corresponding `gbif_id`'s

Value

A dataframe which consist of all the data in the original datacube, appended with column `ott_id`

Examples

```
ex_data <- retrieve_example_data()
mcube <- append_ott_id(ex_data$tree, ex_data$cube, ex_data$matched_nona)
mcube <- dplyr::filter(mcube, !is.na(ott_id))
aggr_cube <- aggregate_cube(mcube)
```

`calculate_faithpd` *Calculation of Faith's PD*

Description

This function calculates Faith's PD, based on a provided list of species and a phylogenetic tree.

Usage

```
calculate_faithpd(tree, species, MRCA)
```

Arguments

`tree`	An object of class `'phylo'`, a phylogenetic tree in Newick format that was parsed by `ape::read_tree()`
`species`	A character vector where each element is a species, and more specifically, matches a tip label of the phylogenetic tree exactly
`MRCA`	Node id of the taxon that represents the most recent common ancestor of the set of species under study

Value

A string that combines "Phylogenetic diversity:" and the calculated value

Examples

```
ex_data <- retrieve_example_data()
# determine the most recent common ancestor of all species under study
# (not necessarily all species in the tree!)
MRCA <- ape::getMRCA(ex_data$tree, ex_data$tree$tip.label)
species <- c("Fagus lucida", "Castanopsis fabri", "Quercus_robur")
calculate_faithpd(ex_data$tree, species, MRCA)
```

`check_completeness` *Check if provided phylogenetic tree is complete and covers all species in occurrence cube*

Description

This function calculates which number of species in the provided occurrence cube, is not a tip label of the provided phylogenetic tree.

Usage

```
check_completeness(mcube)
```

Arguments

`mcube`	A dataframe which is returned by the function append_ott_id(), and contains the occurrence datacube with `ott_id` variable appended. format that was parsed by `ape::read_tree()`

Value

a list - first element is the total number of species in the occurrence cube, second element is the number of species lacking in the phylogenetic tree.

Examples

```
ex_data <- retrieve_example_data()
mcube<- append_ott_id(ex_data$tree, ex_data$cube, ex_data$matched_nona)
check_completeness(mcube)
```

---

convert_multipolygons *Convert multisurface object to multipolygon object*

Description

Convert multisurface object to multipolygon object

Usage

```
convert_multipolygons(object)
```

Arguments

`object`	An object of class multisurface

Value

An object of class multipolygon

Examples

```
library(dplyr)
ex_data <- retrieve_example_data()
mcube <- append_ott_id(ex_data$tree, ex_data$cube, ex_data$matched_nona)
mcube <- dplyr::filter(mcube, !is.na(ott_id))
PD_cube <- get_pd_cube(mcube, ex_data$tree)
PD_cube_geo <- right_join(ex_data$grid, PD_cube,
                          by = join_by(CELLCODE == eeacellcode))
cutoff <- 150
PD_cube_geo$PD_high <- as.factor(ifelse((PD_cube_geo$PD > cutoff), 1, 0))
cube_highPD <- PD_cube_geo[PD_cube_geo$PD_high == 1,
    c("OBJECTID", "CELLCODE", "PD", "geometry", "PD_high")]
cube_mp <- convert_multipolygons(cube_highPD)
```

`generate_map_and_indicator` *Mapping PD and calculating indicator*

Description

This function creates, for a geographic area defined by the user, a map with the calculated PD metric for each grid cell and the location of protected nature areas.

Usage

```
generate_map_and_indicator(
  PD_cube,
  grid,
  taxon = NULL,
  bbox_custom = NULL,
  cutoff = NULL
)
```

Arguments

`PD_cube`	An sf dataframe containing the calculated PD metrics (column name 'PD') for each grid cell with occurrences of a selected higher taxon, and the geometries of those grid cells.
`grid`	An sf object with variable detailing grid cell codes and a geometry column
`taxon`	A selected higher taxon, for which the occurrence cube was generated. Used to generate the map's title only.
`bbox_custom`	Optional, numeric vector with custom bounding box coordinates as c(xmin, xmax, ymin, ymax)
`cutoff`	A variable of type numeric which determines the cut-off point between low PD and high PD

Value

a list `PDindicator`, which contains one or more maps in it's first element, and possibly one or more indicator values in it's second element

Examples

```
library(dplyr)
ex_data <- retrieve_example_data()
mcube <- append_ott_id(ex_data$tree, ex_data$cube, ex_data$matched_nona)
mcube <- dplyr::filter(mcube, !is.na(ott_id))
PD_cube <- get_pd_cube(mcube, ex_data$tree)
PDindicator <- generate_map_and_indicator(
  PD_cube,
  ex_data$grid,
  taxon="Fagales",
  cutoff=150)
map <- PDindicator[[1]]
indicator <- PDindicator[[2]]
```

get_pd_cube — *Find MRCA for data cube and call function to calculate PD metrics*

Description

This function determines the MRCA of all species in the datacube and calls the function(s) to calculate PD metrics

Usage

```
get_pd_cube(mcube, tree, timegroup = NULL, metric = "faith")
```

Arguments

mcube	An occurrence data cube with matched names appended, product of function taxonmatch()
tree	A phylogenetic tree with branch lengths
timegroup	Optional, an integer which represents the number of years over which occurrences need to be aggregated and the PD value calculated
metric	Name of the PD metric to be calculated

Value

Calculated PD value

Examples

```
library(dplyr)
ex_data <- retrieve_example_data()
mcube <- append_ott_id(ex_data$tree, ex_data$cube, ex_data$matched_nona)
mcube <- dplyr::filter(mcube, !is.na(ott_id))
PD_cube <- get_pd_cube(mcube, ex_data$tree, metric="faith")
```

make_shiny_maps *Visualizing PD maps for time periods in tabs*

Description

This function creates produces an r-shiny app that can showcase multiple PD maps (for separate time periods) in tabs

Usage

```
make_shiny_maps(PDindicator, plots)
```

Arguments

`PDindicator`	List containing PD plots and indicators, produced by function generate_map_and_indicator.R
`plots`	A list of PD maps produced by the function `generate_map_and_indicator()`, named by their time-period.

Value

An r-shiny app with PD maps in tabs

Examples

```
library(dplyr)
ex_data <- retrieve_example_data()
mcube <- append_ott_id(ex_data$tree, ex_data$cube, ex_data$matched_nona)
mcube <- dplyr::filter(mcube, !is.na(ott_id))
PD_cube <- get_pd_cube(mcube, ex_data$tree)
PDindicator<- generate_map_and_indicator(
  PD_cube,
  ex_data$grid,
  taxon="Fagales")
plots <- PDindicator[[1]]
indicators <- PDindicator[[2]]
## Not run: make_shiny_maps(PDindicator, plots)
```

retrieve_example_data *Retrieve example data*

Description

This function specifies the paths to the example data and reads the example data files in and processes them so they are ready to be used in the workflow.

Usage

```
retrieve_example_data(data = "all")
```

Arguments

`data`	a list with the names of the datasets to be retrieved. Can be one or multiple of the following: "all", "tree", "cube", "grid", "pa"

Value

Objects tree (a phylogenetic tree of the order Fagales), cube (an occurrence datacube, see query specifications: https://www.gbif.org/occurrence/download/0004018-241107131044228), grid (EEA 1km grid for study area) and pa (Natura2000 protected area polygons for study area

Examples

```
ex_data <- retrieve_example_data()
print(ex_data$tree)
print(ex_data$cube)
print(ex_data$grid)
print(ex_data$pa)
print(ex_data$matched_nona)
```

`taxonmatch` *Taxon matching*

Description

This function matches the tip labels of a phylogenetic tree (Taxon names or OTT id's) with corresponding GBIF id's.

Usage

```
taxonmatch(tree)
```

Arguments

`tree`	An object of class `'phylo'`, a phylogenetic tree in Newick format that was parsed by `ape::read_tree()`

Value

A dataframe with columns `ott_id` and `gbif_id`

Examples

```
## Not run: ex_data <- retrieve_example_data()
# This can take a while!
mtable <- taxonmatch(ex_data$tree)
## End(Not run)
```

Index