

Package ‘bdvis’

December 8, 2023

Type Package

Title Biodiversity Data Visualizations

Version 0.2.37

Date 2023-12-07

Maintainer Vijay Barve <vijay.barve@gmail.com>

Description Provides a set of functions to create basic visualizations to quickly preview different aspects of biodiversity information such as inventory completeness, extent of coverage (taxonomic, temporal and geographic), gaps and biases. Barve & Otegui (2016) <[DOI:10.1093/bioinformatics/btw333](https://doi.org/10.1093/bioinformatics/btw333)>.

License GPL-3

Encoding UTF-8

Imports maps, plotrix, sqldf, plyr, taxize, treemap, ggplot2, lattice, chron, sf

Suggests rinat

RoxygenNote 7.2.3

BugReports <https://github.com/vijaybarve/bdvis/issues>

NeedsCompilation no

Author Vijay Barve [aut, cre] (<<https://orcid.org/0000-0002-4852-2567>>),
Javier Otegui [aut] (<<https://orcid.org/0000-0002-8997-3559>>)

Repository CRAN

Date/Publication 2023-12-08 00:50:02 UTC

R topics documented:

bdcalendarheat	2
bdcomplete	3
bdsummary	4
bdvis	5
bdwebmap	6
chronohorogram	6

distrigraph	7
format_bdvis	8
getcellid	10
gettaxo	11
mapgrid	12
taxotree	13
tempolar	14

Index	17
--------------	-----------

bdcalendarheat	<i>Calendar heat map of biodiversity data</i>
----------------	-----------------------------------------------

Description

Produces a heat map https://en.wikipedia.org/wiki/Heat_map representing the distribution of records in time.

Usage

```
bdcalendarheat(indf = NA, title = NA)
```

Arguments

indf	input data frame containing biodiversity data set
title	title custom title for the plot

Details

The calendar heat map is a matrix-like plot where each cell represents a unique date, and the color the cell is painted with shows the amount of records that have that particular date. Rows are week-days and columns are week numbers, each year having its own "panel".

Value

No return value, called for plotting the heatmap plot

See Also

Other Temporal visualizations: [chronohorogram\(\)](#), [tempolar\(\)](#)

Examples

```
## Not run:
bdcalendarheat(inat)

## End(Not run)
```

bdcomplete	<i>Computes completeness values of the dataset</i>
------------	----------------------------------------------------

Description

Computes completeness values for each cell. Currently returns Chao2 index of species richness.

Usage

```
bdcomplete(indf, recs = 50, gridscale = 1)
```

Arguments

indf	input data frame containing biodiversity data set
recs	minimum number of records per grid cell required to make the calculations. Default is 50. If there are too few records, the function throws an error.
gridscale	plot the map grids at specific degree scale. Default is 1.

Details

After dividing the extent of the dataset in cells (via the [getcellid](#) function), the function calculates the Chao2 estimator of species richness. Given the nature of the calculations, a minimum number of records must be present on each cell to properly compute the index. If there are too few records in the cells, the function is unable to finish, and it throws an error.

This function produces a plot of number of species versus completeness index to give an idea of output. The data frame returned can be used to visualize the completeness of the data using [mapgrid](#) function with ptype as "complete".

Value

data.frame with the columns

- "Cell_id" - id of the cell
 - "nrec" - Number of records in the cell
 - "Sobs" - Number of Observed species
 - "Sest" - Estimated number of species
 - "c" - Completeness ratio the cell
- Plots a graph of Number of species vs completeness

See Also

[getcellid](#)

Examples

```
## Not run:  
bdcomplete(inat)  
  
## End(Not run)
```

bdsummary

Provides summary of biodiversity data

Description

Calculates some general indicators of the volume, spatial, temporal and taxonomic aspects of the provided data set.

Usage

```
bdsummary(indf)
```

Arguments

indf input data frame containing biodiversity data set

Details

The function returns information on the volume of the data set (number of records), temporal coverage (minimum and maximum dates), taxonomic coverage (brief breakdown of the records by taxonomic levels) and spatial coverage (coordinates of the edges of the bounding box containing all records and division of covered area in degree cells) of the records.

To update spatial grid data to dataset, please use [format_bdvis](#) or [getcellid](#) function before using bdsummary.

Value

No return value, just displays the summary in console

See Also

Other Data preparation functions: [format_bdvis\(\)](#), [getcellid\(\)](#), [gettaxo\(\)](#)

Examples

```
## Not run:  
if (requireNamespace("rinat", quietly=TRUE)) {  
  inat <- get_inat_obs_project("reptileindia")  
  inat <- format_bdvis(inat, source="rinat")  
  bdsummary(inat)  
}  
  
## End(Not run)
```

Description

Biodiversity data visualizations using R would be helpful to understand completeness of biodiversity inventory, extent of geographical, taxonomic and temporal coverage, gaps and biases in data.

Data preparation

- [bdsummary](#)
- [format_bdvis](#)
- [getcellid](#)
- [gettaxo](#)

Spatial visualizations

- [bdwebmap](#)
- [mapgrid](#)

Temporal Visualizations

- [bdcalendarheat](#)
- [chronohorogram](#)
- [tempolar](#)

Taxonomic Visualizations

- [taxotree](#)

Miscellaneous functions

- [bdcomplete](#)
- [distrigraph](#)

Citation

- Barve, V., & Otegui, J. (2016). *bdvis: Biodiversity data visualizations* (R package V 0.2). Retrieved from <https://cran.r-project.org/web/packages/bdvis/index.html>

bdwebmap	<i>(Deprecated) Interactive web page based map of records</i>
----------	---------------------------------------------------------------

Description

(Deprecated) Interactive web page based map of records

Usage

```
bdwebmap()
```

Value

No return value. NULL

chronohorogram	<i>Draws a chronohorogram of records</i>
----------------	------------------------------------------

Description

Draws a detailed temporal representation (also known as chronohorogram) of the dates in the provided data set. For more information on the chronohorogram, please see the References section.

Usage

```
chronohorogram(  
  indf = NA,  
  title = "Chronohorogram",  
  startyear = 1980,  
  endyear = NA,  
  colors = c("red", "blue"),  
  ptsize = 1  
)
```

Arguments

indf	input data frame containing biodiversity data set
title	title of the plot. Default is "Chronohorogram"
startyear	starting year for the plot. Default is 1980
endyear	end year for the plot. Default is current year
colors	Pair of colors to build color gradient, in the form of a character vector. Default is blue (less) - red (more) gradient <code>c("red", "blue")</code>
ptsized	point size adjustment factor. Default is 1

Value

No return value, called for plotting the graph

References

Arino, A. H., & Otegui, J. (2008). Sampling biodiversity sampling. In Proceedings of TDWG (pp. 77-78). Retrieved from <http://www.tdwg.org/fileadmin/2008conference/documents/Proceedings2008.pdf#page=77>

See Also

Other Temporal visualizations: [bdcalendarheat\(\)](#), [tempolar\(\)](#)

Examples

```
## Not run:
chronohorogram(inat)

## End(Not run)
```

distrigraph

Distribution graphs

Description

Build plots displaying distribution of biodiversity records among user-defined features.

Usage

```
distrigraph(indf, ptype = NA, cumulative = FALSE, ...)
```

Arguments

<code>indf</code>	input data frame containing biodiversity data set
<code>ptype</code>	Feature to represent. Accepted values are "species", "cell", "efforts" and "effort-species" (year)
<code>cumulative</code>	with <code>ptype</code> as efforts, plot a cumulative records graph
<code>...</code>	any additional parameters for the plot function.

Details

The main use of this function is to create record histograms according to different features of the data set. For example, one might want to see the evolution of records by year, or by species. This function enables easy access to such plots.

Value

No return value, called for plotting the graphs

Examples

```
## Not run:
  distrigraph(inat,ptype="cell",col="tomato")
  distrigraph(inat,ptype="species",ylab="Species")
  distrigraph(inat,ptype="efforts",col="red")
  distrigraph(inat,ptype="efforts",col="red",type="s")

## End(Not run)
```

 format_bdvis

Prepare data frame for flagging functions

Description

format_bdvis renames certain fields in the data frame to make sure the other package functions knows how to use them. This step is highly recommended for the proper working of the functions.

Usage

```
format_bdvis(
  indf,
  source = NULL,
  config = NULL,
  quiet = FALSE,
  gettaxo = FALSE,
  ...
)
```

Arguments

indf	Required. The data.frame on which to operate.
source	Optional. Indicates the package that was used to retrieve the data. Currently accepted values are "rvertnet", "rgbif", "bdsns" or "rinat". Either source, config or individual parameters must be present (see details).
config	Optional. Configuration object indicating mapping of field names from the data.frame to the DarwinCore standard. Useful when importing data multiple times from a source not available via the source argument. Either source, config or individual parameters must be present (see details).
quiet	Optional. Don't show any logging message at all. Defaults to FALSE.
gettaxo	optional. Call function gettaxo to build higher level taxanony. Defaults to FALSE.
...	Optional. If none of the previous is present, the four key arguments (Latitude, Longitude, Date_collected, Scientific_name) can be put here. See examples.

Details

When invoked, there are three ways of indicating the function how to transform the data.frame: using the source parameter, providing a config object with field mapping, or passing individual values to the mapping function. This is the order in which the function will parse arguments; source overrides config, which overrides other mapping arguments.

source refers to the package that was used to retrieve the data. Currently, three values are supported for this argument: "rgbif", "rvertnet", "besns" and "rinat", but many more are on their way. A caution with "besns" data is the scientific name has to be in the field "searchText".

config asks for a configuration object holding the mapping of the field names. This option is basically a shortcut for those users with custom-formatted data.frames who will use the same mapping many times, to avoid having to type them each time. In practice, this object is a named list with the following four fields: Latitude, Longitude, Date_collected and Scientific_name. Each element must have a string indicating the name of the column in the data.frame holding the values #' for that element. If the data.frame doesn't have one or more of these fields, #' put NA in that element; otherwise, the function will throw an error. See the examples section.

If none of the two is provided, the function expects the user to provide the mapping by passing the individual column names associated with the right term. See the examples section.

Value

The provided data frame, with field names changed to suite the functioning of further visualization functions.

See Also

Other Data preparation functions: [bdsummary\(\)](#), [getcellid\(\)](#), [gettaxo\(\)](#)

Examples

```
## Not run:
# Using the rgbif package and the source argument
if (requireNamespace("rinat", quietly=TRUE)) {
  d <- get_inat_obs_project("reptileindia")
  d <- format_bdvis(d, source="rinat")

  # Using a configuration object, matches 'rinat' schema
  conf <- list(Latitude <- "latitude",
              Longitude <- "longitude",
              Date_collected <- "Observed.on",
              Scientific_name <- "Scientific.name")
  d <- format_bdvis(d, config=conf)

  # Passing individual parameters, all optional
  d <- format_bdvis(d,
                  Latitude <- "lat",
                  Longitude <- "lng",
                  Date_collected <- "ObservedOn",
                  Scientific_name <- "sciname")
}
```

```
## End(Not run)
```

getcellid	<i>Assign GBIF style degree cell ids and generate custom grid cell ids</i>
-----------	----------------------------------------------------------------------------

Description

Calculate and assign a GBIF-style degree cell id and centi-degree (0.1 degrees, dividing a 1 degree cell into 100 centi-degree cells) cell id to each record. This function also creates a custom grid scale if parameter `gridscale` is supplied. This is a necessary previous step for some functions like [mapgrid](#)

Usage

```
getcellid(indf, gridscale = 0)
```

Arguments

<code>indf</code>	input data frame containing biodiversity data set
<code>gridscale</code>	generate custom grid scale column for mapping. Default is 0.

Value

data frame with two columns for `cell_id` added

See Also

Other Data preparation functions: [bdsummary\(\)](#), [format_bdvis\(\)](#), [gettaxo\(\)](#)

Examples

```
## Not run:  
getcellid(inat)  
  
## End(Not run)
```

`gettaxo`*Get higher taxonomy data*

Description

This function is slated to deprecate in next version. Please use function `taxotools::list_higher_taxo` instead.

Usage

```
gettaxo(indf, genus = FALSE, verbose = FALSE, progress = TRUE)
```

Arguments

<code>indf</code>	input data frame containing biodiversity data set
<code>genus</code>	If TRUE, use only genus level data to get taxonomy
<code>verbose</code>	If TRUE, displays each name string for which the higher taxonomy is sought
<code>progress</code>	If TRUE prints progress bar and messages on the consol.

Details

Retrieve higher taxonomy information (like Family and Order) for each record from the "Encyclopedia of Life" web API.

This function makes use of certain functions in the `taxize` package. It scans and retrieves the taxonomic hierarchy for each scientific name (or just genus name) in the data set. When new data are retrieved, they are stored in a local sqlite database, `taxo.db`, for faster further access.

Value

`indf` with added / updated columns

- "Kingdom" - Kingdom of the Scientific name
- "Phylum" - Phylum of the Scientific name
- "Order_" - Order of the Scientific name
- "Family" - Family of the Scientific name
- "Genus" - Genus of the Scientific name

and also saves a local copy of taxonomy downloaded for future use in `taxo.db` sqlite file

See Also

Other Data preparation functions: `bdsummary()`, `format_bdvis()`, `getcellid()`

Examples

```
## Not run:
inat <- gettaxo(inat)

## End(Not run)
```

mapgrid

Maps the data points on the map in grid format

Description

Customizable grid-based spatial representation of the coordinates of the records in the data set.

Usage

```
mapgrid(
  indf = NULL,
  comp = NULL,
  ptype = "records",
  title = "",
  bbox = NA,
  legscale = 0,
  collow = "blue",
  colhigh = "red",
  mapdatabase = NULL,
  region = NULL,
  shp = NA,
  gridscale = 1,
  customize = NULL
)
```

Arguments

indf	input data frame containing biodiversity data set
comp	Completeness matrix generate by function bdcomplete
ptype	Type of map on the grid. Accepted values are "presence" for presence/absence maps, "records" for record-density map, "species" for species-density map and "complete" for completeness map
title	title for the map. There is no default title
bbox	bounding box for the map in format c(xmin,xmax,ymin,ymax)
legscale	Set legend scale to a higher value than the max value in the data
collow	Color for lower range in the color ramp of the grid
colhigh	Color for higher range in the color ramp of the grid
mapdatabase	Parameter is deprecated

region	Parameter is deprecated. Please use shape files.
shp	path to shapefile to load as basemap (default NA)
gridscale	plot the map grids at scale specified. Scale needs to be specified in decimal degrees. Default is 1 degree which is approximately 100km.
customize	additional customization string to customize the map output using ggplot2 parameters

Details

This function builds a grid map colored according to the density of records in each cell. Grids are 1-degree cells, built with the `getcellid` function. Currently, four types of maps can be rendered. Presence maps show only if the cell is populated or not, without paying attention to how many records or species are present. Record-density maps apply a color gradient according to the number of records in the cell, regardless of the number of species they represent. Species-density maps apply a color gradient according to the number of different species in the cell, regardless of how many records there are for each one of those. Completeness maps apply a color gradient according to the completeness index, from 0 (incomplete) to 1 (complete).

See parameter descriptions for ways of customizing the map.

Value

No return value, called for plotting the graph

Examples

```
## Not run:
mapgrid(inat,ptype="records", region="India")

## End(Not run)
```

taxotree

Treemap based on taxonomic hierarchy of records

Description

Draws a treemap (<https://en.wikipedia.org/wiki/Treemapping>) based on the taxonomic information of the records.

Usage

```
taxotree(
  indf,
  n = 30,
  title = NA,
  legend = NA,
  sum1 = "Family",
  sum2 = "Genus"
)
```

Arguments

indf	input data frame containing biodiversity data set
n	maximum number of rectangles to be plotted in the treemap. Default is 30
title	title for the tree. Default is "Records per <sum1>"
legend	legend title. Default is "Number of <sum2>"
sum1	Taxonomic level whose density will be represented with different cell sizes
sum2	Taxonomic level whose density will be represented with a color gradient

Details

This function builds a treemap of the taxonomic information present in the data set. It represents this information at two levels (with the arguments sum1 and sum2). The first level (sum1) will be represented with cell sizes and is a reflection of the number of records in that group. If, for example, "Family" is selected as value for sum1, the size of the cells in the treemap will be directly proportional to the number of records for that taxonomic family. The second level (sum2) will be represented by color and is a reflection of the number of sub-groups in a particular cell. If, for example, "Genus" is selected as value for sum2, the color of the cell will depend on the number of different genera for that particular cell.

Value

No return value, called for plotting the graph

References

Otegui, J., Arino, A. H., Encinas, M. A., & Pando, F. (2013). Assessing the Primary Data Hosted by the Spanish Node of the Global Biodiversity Information Facility (GBIF). PLoS ONE, 8(1), e55144. doi:10.1371/journal.pone.0055144

Examples

```
## Not run:
taxotree(inat)

## End(Not run)
```

tempolar

Polar plot of temporal data

Description

Representation in polar axis of the distribution of dates in the provided data set.

Usage

```
tempolar(  
  indf = NA,  
  timescale = NA,  
  title = NA,  
  color = NA,  
  plottype = NA,  
  avg = FALSE  
)
```

Arguments

<code>indf</code>	input data frame containing biodiversity data set
<code>timescale</code>	Temporal scale of the graph, or how are dates aggregated. Accepted values are: d (daily, each feature in the plot represents a day), w (weekly, each feature in the plot represents a week) and m (monthly, each feature in the plot represents a month). Default is d (daily).
<code>title</code>	Title for the graph. Default is "Temporal coverage".
<code>color</code>	color of the graph plot. Default is "red".
<code>plottype</code>	Type of feature. Accepted values are: r (lines), p (polygon) and s (symbols). Default is p (polygon).
<code>avg</code>	If TRUE plots a graph of the average records rather than total numbers. Default is FALSE.

Details

This function returns a plot representing the temporal distribution of records in the data set. This is done by representing dates in a radial axis, with the distance from the center being the amount of records for that particular date. This function allows several arguments indicating different representation types. See the arguments section for an enumeration of them.

Value

No return value, called for plotting the graph

References

Otegui, J., Arino, A. H., Encinas, M. A., & Pando, F. (2013). Assessing the Primary Data Hosted by the Spanish Node of the Global Biodiversity Information Facility (GBIF). *PLoS ONE*, 8(1), e55144. doi:10.1371/journal.pone.0055144

See Also

Other Temporal visualizations: [bdcalendarheat\(\)](#), [chronohorogram\(\)](#)

Examples

```
## Not run:  
tempolar(inat)  
  
## End(Not run)
```


Index

* Data preparation functions

bdsummary, 4
format_bdvis, 8
getcellid, 10
gettaxo, 11

* Spatial visualizations

mapgrid, 12

* Taxonomic visualizations

taxotree, 13

* Temporal visualizations

bdcalendarheat, 2
chronohorogram, 6
tempolar, 14

bdcalendarheat, 2, 5, 7, 15

bdcomplete, 3, 5, 12

bdsummary, 4, 5, 9–11

bdvis, 5

bdwebmap, 5, 6

chronohorogram, 2, 5, 6, 15

distrigraph, 5, 7

format_bdvis, 4, 5, 8, 10, 11

getcellid, 3–5, 9, 10, 11, 13

gettaxo, 4, 5, 8–10, 11

mapgrid, 3, 5, 10, 12

plot, 7

taxize, 11

taxotree, 5, 13

tempolar, 2, 5, 7, 14