

Package ‘rangeMapper’

February 15, 2012

Version 0.0-6

Date 2011-Apr-07

Title A platform for the study of macroecology of life history traits

Author Mihai Valcu <valcu@orn.mpg.de>, James Dale <J.Dale@massey.ac.nz>

Maintainer Mihai Valcu <valcu@orn.mpg.de>

Depends R (>= 2.10.0), methods, utils, RSQLite,RSQLite.extfuns, sp,rgdal, maptools, lattice, RColorBrewer, tcltk, tcltk2, pixmap,classInt

Suggests raster

SystemRequirements Bwidget, Tktable

Description A package to manipulate species range (extent-of-occurrence) maps, mainly tools for easy generation of biodiversity (species richness) or life-history traits maps.

License GPL (>= 2)

URL <http://www.r-project.org>, <http://rangemapper.r-forge.r-project.org/>

Repository CRAN

Date/Publication 2011-04-08 07:11:59

R topics documented:

bio.save	2
bioSaveDataFrame-class	3
bioSaveFile-class	4
brewer.pal.get	5
canvas.save	5
global.bbox	6
gridSize	7
gridSize.save	8

plot-methods	9
processRanges	10
rangeFiles-class	11
rangeMap-class	12
rangeMap.export	13
rangeMap.fetch	13
rangeMap.save	14
rangeMap.start	15
rangeMapBbox	16
rangeMapFetch	16
rangeMapper	17
rangeMapProcess	21
rangeMapRemove	22
rangeMapSave	23
rangeMapStart	24
rm.rangeMapper	24
RMQuery	25
selectShpFiles	26
SpatialPixelsRangeMap	27
tkColorPalette	28
tkdbBrowse	29
wrens	30
Index	32

bio.save

Import 'BIO' tables

Description

Import tables (e.g. life history data) to a rangeMapper project.

Usage

```
bio.save(con, loc, tableName, ...)
metadata2bio(con, ...)
bio.merge(con, tableName, ...)
```

Arguments

con	An sqlite connection pointing to a valid rangeMapper project.
loc	file location or data.frame name
tableName	if missing, the name of the file or data.frame is used
...	Other arguments to pass to methods of bioSaveFile-class class, e.g. the ID, the column corresponding to the names of the range files

Value

A 'BIO' table is created in the corresponding rangeMapper project.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

References

Valcu, M., Dale, J. and Kempnaers, B. (in prep) rangeMapper: A platform for the study of macroecology of life history traits.

See Also

[rangeMapper](#) for a complete example. [bioSaveDataFrame-class](#) [bioSaveFile-class](#) [wrens](#)

bioSaveDataFrame-class

Class "bioSaveDataFrame"

Description

The methods of this class allows importing [data.frames](#) to a rangeMapper project.

Objects from the Class

Objects can be created by calls of the form `new("bioSaveDataFrame", ...)`.

Slots

loc: Object of class "data.frame"

Extends

Class "[rangeMapSave](#)", directly. Class "[rangeMap](#)", by class "[rangeMapSave](#)", distance 2.

Methods

bioSave signature(object = "bioSaveDataFrame"): ...

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[bio.save](#)

bioSaveFile-class *Class "bioSaveFile"*

Description

The methods of this class allows importing external tables to a rangeMapper project.

Objects from the Class

Objects can be created by calls of the form `new("bioSaveFile", ...)`.

Objects from the Class

Objects can be created by calls of the form `new("bioSaveDataFrame", ...)`.

Slots

loc: Object of class "data.frame"

sep: Object of class "character"

Extends

Class "[rangeMapSave](#)", directly. Class "[rangeMap](#)", by class "rangeMapSave", distance 2.

Methods

bioSave signature(object = "bioSaveDataFrame")

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[bio.save](#)

brewer.pal.get	<i>brewer.pal color palettes.</i>
----------------	-----------------------------------

Description

A simple wrapper around [brewer.pal.info](#)

Usage

```
brewer.pal.get(palette)
```

Arguments

palette	The name of a color palette. When NULL all the qualitative color palettes are returned.
---------	---

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[tkColorPalette](#) [brewer.pal](#)

canvas.save	<i>Project's canvas</i>
-------------	-------------------------

Description

The canvas is a regular grid of a given resolution. Each range map is overlaid onto the canvas and the results saved to project.

Usage

```
canvas.save(con)
canvas.fetch(con)
```

Arguments

con	An sqlite connection pointing to a valid rangeMapper project.
-----	---

Value

canvas.fetch Returns a [SpatialPixelsDataFrame](#) object.

Note

The method `canvasSave()` fails if `grid.size` was not set. The method `canvasSave()` fails if the canvas was already constructed for the given project.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

References

Valcu, M., Dale, J. and Kempnaers, B. (in prep) rangeMapper: A platform for the study of macroecology of life history traits.

See Also

[rangeMapper](#) for a complete example.
[gridSize.save](#)

global.bbox

Global bounding box

Description

Computes, sets or retrieves the global spatial bounding box.

Usage

```
global.bbox.save(con, ...)  
global.bbox.fetch(con)
```

Arguments

<code>con</code>	An <code>SQLiteConnection</code> object pointing to a rangeMapper project
<code>...</code>	Other arguments to pass to the corresponding methods

Details

[global.bbox.save](#) saves the global bounding box and the proj4 string to the rangeMapper's 'meta-data' table. [global.bbox.fetch](#) retrieves the global bounding box from a rangeMapper project as as a `SpatialPolygonsDataFrame`.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

References

Valcu, M., Dale, J. and Kempnaers, B. (in prep) rangeMapper: A platform for the study of macroecology of life history traits.

See Also

[rangeMapper](#) for a complete example. [proj4string bbox](#)

gridSize	<i>Class "gridSize"</i>
----------	-------------------------

Description

Objects of class gridSize are used to set the pixel size and thus the canvas size of a rangeMapper project. If gridSize is not given a sensible default is set.

Objects from the Class

Objects can be created by calls of the form `new("gridSize", ...)`.

Slots

gridSize: Object of class "numeric"

Extends

Class "[rangeMap](#)", directly.

Methods

gridSizeSave signature(object = "gridSize"): ...

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[global.bbox](#)

gridSize.save	<i>Save or retrieve the grid size from an rangeMapper project.</i>
---------------	--

Description

Save or retrieve the grid size from the active sqlite database.

Usage

```
gridSize.save(con, ...)  
gridSize.fetch(con)
```

Arguments

con	A connection pointing to a valid rangeMapper project.
...	Other arguments to pass to methods of class gridSize-class
gridSize	A numeric vector of one unit length.

Value

gridSize.fetch	Returns a numeric vector of one unit length containing the grid size previously saved by gridSize.save
----------------	--

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

References

Valcu, M., Dale, J. and Kempnaers, B. (in prep) rangeMapper: A platform for the study of macroecology of life history traits.

See Also

[rangeMapper](#) for a complete example. [global.bbox](#)

plot-methods

Plot a SpatialPixelsRangeMap

Description

This is a wrapper around [splot](#)

Methods

```
signature(x = "SpatialPixelsRangeMap", y = "missing") x=SpatialPixelsRangeMap  
plot(x, colorpalette = brewer.pal.get('Spectral')[11:1], ncols = 20, scales  
= FALSE, style = "equal", ...)
```

x	a SpatialPixelsRangeMap object
colorpalette	A color palette. See also tkColorPalette which allows for interactive choice of color palettes
ncols	Number of color classes required, default to 20; argument to be passed to classIntervals
scales	If 'FALSE', default, axes scale are not drawn.
style	class interval style; see classIntervals for more details
...	Any argument that can be passed to image.plot and to splot

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[rangeMapper](#) for a complete example.

processRanges	<i>Process ranges</i>
---------------	-----------------------

Description

Each polygon range is overlaid on the [canvas](#) and the results are saved to the active project file.

Usage

```
processRanges(dir,con,metadata = TRUE, ...)
```

Arguments

con	An sqlite connection pointing to a valid rangeMapper project.
dir	ranges file directory
metadata	A logical value. If 'TRUE' ranges metadata: range area, median range centroid and range extents (min, max) are saved to 'metadata_ranges' table
...	One or several user defined functions applied on each range. The output of these functions will be appended to metadata_ranges table

Details

The overlay is performed using [overlay](#). If the overlay returns no results (i.e. the polygon is smaller than a grid cell) then the centroid of the range will snap to the nearest point and only one grid cell will be returned for that range.

Warning

If thousands of range map polygons are processed, their geometries are complex and/or the canvas resolution is relatively high this step can be time consuming. However, once this step has been performed and the results saved to the database, the other operations are very fast.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

References

Valcu, M., Dale, J. and Kempnaers, B. (in prep) rangeMapper: A platform for the study of macroecology of life history traits.

See Also

[rangeMapper](#) for a complete example.

rangeFiles-class *Class "rangeFiles"*

Description

A class used to handle polygon shp files.

Objects from the Class

Objects can be created by calls of the form `new("rangeFiles", ...)`.

Slots

dir: Object of class "character"; directory containing the range files; one for each taxon (e.g. species)

ogr: Object of class "logical" if TRUE a data.frame with two columns is returned (see [selectShpFiles](#) for details)

polygons.only: Object of class "logical" If TRUE only polygon file paths are returned, else the location to all shape files regardless their type is returned.

Methods

rangeFiles signature(object = "rangeFiles"): initiate a new rangeFiles object

Author(s)

Mihai Valcu, <valcu@orn.mpg.de>

See Also

[selectShpFiles](#)

rangeMap-class	Class "rangeMap"
----------------	------------------

Description

A class to formally define a rangeMapper project.

Objects from the Class

Objects can be created by calls of the form `new("rangeMap", ...)`.

Slots

CON: Object of class "SQLiteConnection" pointing to a rangeMapper project

VERSION: Object of class "character" corresponding to the rangeMapper version

ID: Object of class "character"; the ID column corresponding to the names of the imported range files.

BIOID: Object of class "character" the ID column corresponding to the names of imported 'BIO' tables.

PROJ4STRING: Object of class "character" the proj4 string

GRIDSIZE: Object of class "character" grid size (in map units)

BBOX: Object of class "character" bounding box

METADATA_RANGES: Object of class "character" The name of a predefined table optionally constructed at the range import stage

CANVAS: Object of class "character" The name of the table hosting the project's regular grid (the canvas)

RANGES: Object of class "character" The name of the table hosting the ranges

BIO: Object of class "character" The prefix for 'BIO' tables

MAP: Object of class "character" The prefix for 'MAP' tables

Methods

canvasFetch signature(object = "rangeMap"):

canvasSave signature(object = "rangeMap"):

gridSizeFetch signature(object = "rangeMap"):

rangeMapBboxFetch signature(object = "rangeMap"):

Author(s)

Mihai Valcu, <valcu@orn.mpg.de>

See Also

[rangeMapStart-class](#), [gridSize-class](#), [rangeMapProcess-class](#), [rangeMapSave-class](#), [rangeMapFetch-class](#), [rangeMapRemove-class](#), [MapImport-class](#)

rangeMap.export *Export 'MAP' tables*

Description

Export 'MAP' tables as single-band geotiff files

Usage

```
rangeMap.export(con, dirName, ...)
```

Arguments

con	An sqlite connection pointing to a valid rangeMapper project.
dirName	The directory name where the 'MAP's will be exported. If missing, the 'MAP's will be exported in project's directory
...	Further arguments to pass to writeGDAL

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

References

Valcu, M., Dale, J. and Kempnaers, B. (in prep) rangeMapper: A platform for the study of macroecology of life history traits.

rangeMap.fetch *Fetch 'MAP' tables*

Description

Fetch 'MAP's from a rangeMapper project.

Usage

```
rangeMap.fetch(dbcon, maps)
```

Arguments

dbcon	An sqlite connection pointing to a valid rangeMapper project.
maps	A character vector containing 'MAP' names.

Value

A [SpatialPixelsRangeMap-class](#) is returned.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

References

Valcu, M., Dale, J. and Kempnaers, B. (in prep) rangeMapper: A platform for the study of macroecology of life history traits.

See Also

[rangeMapper](#) for a complete example. [SpatialPixelsRangeMap-class](#)

rangeMap.save	<i>Save a rangeMap</i>
---------------	------------------------

Description

Compute a chosen statistic (e.g. richness (count), mean, median or any user-defined statistic) at each grid cell.

Usage

```
rangeMap.save(CON, tableName, FUN, biotab, biotrait, subset, path , overwrite, ...)
```

Arguments

CON	An sqlite connection pointing to a valid rangeMapper project.
tableName	Name of the table (quoted) to be added to the sqlite database. The prefix 'MAP' will be appended to tableName prior to saving.
FUN	the function to be applied to each pixel. If FUN is missing then species richness (species count) is computed.
biotab	character string identifying the 'BIO' table to use.
biotrait	character string identifying the ID of the 'BIO' table. see bio.save
subset	A named list . See details
path	Path to the raster file(quoted) to be imported to the existing project. raster package is required at this step.
overwrite	If TRUE then the table is removed
...	When R is an R function, ... denotes any extra arguments to be passed to it.

Details

The subset argument accepts a named list. Names refers to 'BIO', 'MAP' and 'metadata_ranges' table names while the strings in the list are character strings containing the SQL WHERE clause. The subset can point to only one table type (e.g. `list(MAP_species_richness = "species_richness > 500")`) or can point to several table types (e.g. `list(BIO_lifeHistory = "clutch_size > 4", MAP_meanAltitude = "meanAltitude < 1000", metadata_ranges = "Area < 1000")`) Any valid SQL expression can be used. See http://www.sqlite.org/lang_expr.html

Value

Returns TRUE when the MAP was created successfully.

Note

SQL aggregate functions are bound to be faster than R functions. For simple aggregate functions like mean, median, sd, count it is advisable to use SQL functions rather than R functions.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[rangeMapper](#) for a complete example.

rangeMap.start	<i>Initiate/open a new rangeMapper project</i>
----------------	--

Description

Initiate/open a new rangeMapper project using an object of class [rangeMapStart-class](#)

Usage

```
rangeMap.start(...)  
rangeMap.open(path, verbose)
```

Arguments

path	Character vector; a path to a valid rangeMapper project
verbose	Character vector; if TRUE the project's summary is printed
...	Arguments to be passed to rangeMapStart-class

Value

An sqlite connection is returned.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[rangeMapper](#) for a complete example.
[rangeMapStart-class](#)

rangeMapBbox	<i>Methods for Function rangeMapBbox in Package 'rangeMapper'</i>
--------------	---

Description

Methods for function rangeMapBbox in Package 'rangeMapper'

Methods

signature(object = "rangeFiles")

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[global.bbox](#)

rangeMapFetch	<i>Class "rangeMapFetch"</i>
---------------	------------------------------

Description

The methods of this class allows fetching maps from a rangeMapper project.

Objects from the Class

Objects can be created by calls of the form new("rangeMapFetch", ...).

Slots

tableName: Object of class "character" MAP name (without the prefix) to fetch.

Extends

Class "[rangeMap](#)", directly.

Methods

rangeMapFetch signature(object = "rangeMapFetch"): ...

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[rangeMap.fetch SpatialPixelsRangeMap-class](#)

rangeMapper	<i>rangeMapper: A platform for the study of macroecology of life history traits.</i>
-------------	--

Description

rangeMapper is a front end platform for the study of macroecology of life history traits at both inter-specific and assemblage levels.

Details

The package uses species range (extent-of-occurrence) vector polygons and life history traits datasets to build up maps (e.g. species richness, mean body mass, ...).

rangeMapper comes with an user-friendly platform-independent tcltk graphical user interface.

Getting Started

For a quick start open the graphical user interface (**gui**) by typing `rangeMapper()`. Mouse over the buttons to see further notes (tool-tips) regarding each step.

A tutorial can be accessed from Help/‘Get started’ under the **gui** or by browsing the ‘doc’ package directory.

See also the example below in the *examples* section on how to use **rangeMapper** from the command line.

The rangeMapper pipeline

- Initiate a new project (an empty sqlite database) using a pre-defined template.
- Extract/define the global bounding box of all selected range maps.
- Generate a regular grid (the canvas) using the previously constructed global bounding box.
- Perform polygon-grid overlay of all selected range maps.
- Optionally compute pre-defined or user-defined range structure indexes.
- Optionally import non-spatial data to be mapped at each grid cell (data are saved as a ‘BIO’ table to database).
- Optionally import georeferenced raster files.
- Compute a chosen statistical model at each grid cell optionally using complex subsets.
- Save, plot, export the MAP-s.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>, <http://orn.mpg.de/mitarbeiter/valcu.html>

James Dale <J.Dale@massey.ac.nz>, <http://quelea.net>

References

Valcu, M., Dale, J. and Kempnaers, B. (in prep) rangeMapper: A platform for the study of macroecology of life history traits.

Examples

```
#####

require(rangeMapper)

# Start or open a project
wd = tempdir()

# open the GUI, messages will be re-directed here
if(interactive() ) rangeMapper()

dbcon = rangeMap.start(file = "wrens.sqlite",
  overwrite = TRUE, dir = tempdir() )

# To connect latter to the project created with rangeMap.start use
dbcon = rangeMap.open(path =
  paste(tempdir(), "wrens.sqlite" , sep = .Platform$file.sep) )

# Project summary
summary(new("rangeMap", CON = dbcon) )

# Breeding range vector files location
f = system.file(package = "rangeMapper",
  "extdata", "wrens", "vector")

# There are several alternatives to save the global bounding box,
# a few are exemplified below
# The default creates a global unprojected bounding box
global.bbox.save(con = dbcon)

# Creating a global bbox using the breeding range vector files

# manually remove all data from the bbox and proj4string table
RMQuery(dbcon, 'DELETE from bbox')
RMQuery(dbcon, 'DELETE from proj4string')

global.bbox.save(con = dbcon, bbox = f)

plot( global.bbox.fetch(dbcon) , axes = TRUE)

# Creating a global bbox using the breeding range vector files
# and re-projecting
# manually remove all data from the bbox table
RMQuery(dbcon, 'DELETE from bbox')
RMQuery(dbcon, 'DELETE from proj4string')

global.bbox.save(con = dbcon,bbox = f,
  p4s =
  CRS(" +proj=cea +lon_0=0 +lat_ts=30 +x_0=0 +y_0=0 +ellps=WGS84 +units=m +no_defs"))

# upload grid size using the default
```

```

gridSize.save(dbcon)

# upload grid size to 100000 sqm
RMQuery(dbcon, 'DELETE from gridSize')
gridSize.save(dbcon, gridSize = 100000)

# save canvas
canvas.save(dbcon)
summary(canvas.fetch(dbcon) )

# Process ranges & BIO tables

# first define a new range structure function

APratio = function(x) {
# returns 1 for a circle polygon
# area
A = sapply(slot(x, "polygons"),
function(x) slot(x, "area") )

# perimeter
a = lapply(slot(x, "polygons"),
function(x) slot(x, "Polygons") )
b = do.call(c, a)
a1 = lapply(b, function(x)
Line(coordinates(x)))
L = sapply(a1, LineLength)

# A-P ratio
2*(sqrt(pi*sum(A)))/sum(L)

}

# the APratio function will also be computed together with the pre-defined functions
processRanges(dir = f, con = dbcon, metadata = TRUE, APratio)

# Manage BIO tables
csvfile = system.file(package="rangeMapper", "data", "wrens.csv")
bio.save(con = dbcon, loc = csvfile , ID = "sci_name")

rm.rangeMapper(con = dbcon, tablePrefix = "BIO")

data(wrens)
bio.save(con = dbcon, loc = wrens, ID = "sci_name")

# metadata_ranges table can be merged with the wrens table
metadata2bio(dbcon)
bio.merge(dbcon, tableNam = "wrensRanges",
"wrens", "metadata_ranges")

```

```

# Save MAPs
rangeMap.save(dbcon) #species richness is the default

#Using sqlite aggregate functions
.sqlAggregate()
rangeMap.save(dbcon, FUN = "median" , biotab = "wrens",
biotrait = "body_size", tableName = "body_size")

rangeMap.save(dbcon, FUN = "median" ,
biotab = "metadata_ranges", biotrait = "V1", tableName = "APratio")

#Using an R function which does not require a formula argument
rangeMap.save(dbcon, FUN = mean , biotab = "wrens",
biotrait = "clutch_size", tableName = "clutch_size",
na.rm = TRUE, overwrite = TRUE)

#Using an user defined R function which has a formula argument
lmSlope = function (formula, data) {
  fm = try(lm(formula, data = data), silent = TRUE)
  if (inherits(fm, "try-error"))
  res = NA else
  res = coef(fm)[2]
  as.numeric(res)
}

# Map of log(body_mass) ~ clutch_size only for species richness > 10
rangeMap.save(dbcon, FUN = lmSlope, biotab = "wrens",
biotrait = "body_mass",
tableName = "slope_body_mass_clutchSize",
formula = log(body_mass) ~ clutch_size , overwrite = TRUE,
list(MAP_species_richness = "species_richness > 5")
)

# rangeMap.save can be also used to import raster maps the current project
r = system.file(package = "rangeMapper",
"extdata", "etopo1", "etopo1_Americas.tif")

rangeMap.save(dbcon, path = r, tableName = 'meanAltitude',
FUN = mean , overwrite = TRUE)

# rangeMap.save takes a subset argument on both
# metadata_ranges, MAP and BIO tables

# clutch size of the highland species
rangeMap.save(dbcon, biotab = "wrens", biotrait = "clutch_size",
tableName = "highlandClutch", overwrite = TRUE,
subset = list(MAP_meanAltitude = "meanAltitude > 2000") )

# clutch size of restricted range lowland species
rangeMap.save(dbcon, biotab = "wrens", biotrait = "clutch_size",
tableName = "endemicsLowlandClutch", overwrite = TRUE,
subset = list(MAP_meanAltitude = "meanAltitude < 1000",

```

```

metadata_ranges = "Area < 100000000000" )

# Fetch and plot Maps

plot(rangeMap.fetch(dbcon, 'species_richness') ,n = 10, style = "fisher")

plot(rangeMap.fetch(dbcon, "APratio"), style = "equal" )
plot(rangeMap.fetch(dbcon, 'meanAltitude'), style = "quantile")

clutch = rangeMap.fetch(dbcon,
c('clutch_size', 'slope_body_mass_clutchSize', 'body_size') )
plot(clutch, style = "fisher")

plot(rangeMap.fetch(dbcon, 'highlandClutch'), n = 4)

image(rangeMap.fetch(dbcon, 'meanAltitude') )
image(rangeMap.fetch(dbcon, 'endemicsLowlandClutch') ,
add = TRUE, col = "blue")

setwd(wd)

```

rangeMapProcess	<i>Class "rangeMapProcess"</i>
-----------------	--------------------------------

Description

The methods of this class allows importing vector (shp) range files to a rangeMapper project.

Objects from the Class

Objects can be created by calls of the form `new("rangeMapProcess", ...)`.

Slots

metadata: Object of class "logical" Should the range structure parameters be extracted?

parallel: Object of class "character" Not yet implemented.

Extends

Class "[rangeFiles](#)", directly. Class "[rangeMap](#)", directly.

Methods

rangeMapProcess signature(object = "rangeMapProcess"): ...

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[rangeMapper](#) for a complete example. [rangeMapProcess](#)

rangeMapRemove	<i>Class "rangeMapRemove"</i>
----------------	-------------------------------

Description

The method defined for this class allows removing rangeMapper objects.

Objects from the Class

Objects can be created by calls of the form `new("rangeMapRemove", ...)`.

Slots

`tableName`: Object of class "character"

`tablePrefix`: Object of class "character" 'MAP' or 'BIO'

Extends

Class "[rangeMap](#)", directly.

Methods

rangeMapRemove signature(object = "rangeMapRemove"): ...

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[rm.rangeMapper](#)

rangeMapSave	<i>Class "rangeMapSave"</i>
--------------	-----------------------------

Description

This class' methods are used to save 'MAP' tables to an existing rangeMapper project.

Objects from the Class

Objects can be created by calls of the form `new("rangeMapSave", ...)`.

Slots

`biotab`: Object of class "character"

`biotrait`: Object of class "character"

`tableName`: Object of class "character"

`subset`: Object of class "list"

`path`: Object of class "character" Path to the file to be imported

Extends

Class "[rangeMap](#)", directly. Class "[rangeMapSave](#)", directly. Class "[rangeMap](#)", by class "rangeMapSave", distance 2.

Methods

rangeMapSave signature(object = "rangeMapSave", FUN = "character", formula = "missing")

rangeMapSave signature(object = "rangeMapSave", FUN = "function", formula = "formula")

rangeMapSave signature(object = "rangeMapSave", FUN = "function", formula = "missing")

rangeMapSave signature(object = "rangeMapSave", FUN = "missing", formula = "missing")

rangeMapSave signature(object = "MapImport"): ...

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[rangeMap.save](#)

rangeMapStart	<i>Class "rangeMapStart"</i>
---------------	------------------------------

Description

A class used to initiate a rangeMapper project.

Objects from the Class

Objects can be created by calls of the form `new("rangeMapStart", ...)`.

Slots

dir: Object of class "character"; the directory hosting the rangeMapper project

file: Object of class "character" Name of the rangeMapper project; an *.sqlite file

skeleton: Object of class "list" containing the SQL calls to initiate a rangeMapper project

overwrite: Object of class "logical"; if TRUE all the project's table are removed

Methods

rangeMapStart signature(object = "rangeMapStart"): starts a new rangeMapper project

Author(s)

Mihai Valcu, <valcu@orn.mpg.de>

See Also

[rangeMapper](#) for a complete example.

rm.rangeMapper	<i>Remove tables from a give project</i>
----------------	--

Description

Remove tables given prefix attribute or by name

Usage

```
rm.rangeMapper(con, ...)
```

Arguments

con A valid sqlite connection.

... arguments passed to methods of "[rangeMapRemove](#)", specifically 'tablePrefix' or 'tableName'

Note

The default 'rm.rangeMapper(con)' will remove all 'MAP' and 'BIO' tables.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[rangeMapper](#) for a complete example.

RMQuery

Query a rangeMapper project

Description

For the moment a simple wrapper to sqliteQuickSQL.

Usage

```
RMQuery(con, statement)
```

Arguments

con	An sqlite connection
statement	An SQL string

Value

Returns either a data.frame if the statement is a select-like or NULL otherwise.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[sqliteQuickSQL](#).

selectShpFiles	<i>Select (recursively) shape files</i>
----------------	---

Description

Returns the file path to all '.shp' files in a directory.

Usage

```
selectShpFiles(dir, ...)
```

Arguments

dir	character string specifying the directory containing .shp files.
...	Other arguments to pass to methods of class rangeFiles-class

Value

Either a [data.frame](#) or a character vector is returned.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

References

Valcu, M., Dale, J. and Kempnaers, B. (in prep) rangeMapper: A platform for the study of macroecology of life history traits.

See Also

[rangeMapper](#) for a complete example.
[rangeFiles-class](#)

Examples

```
f= system.file(package="rangeMapper", "extdata", "wrens", "vector")
res = selectShpFiles(f, ogr = TRUE, polygons.only = TRUE)
head(res)
```

SpatialPixelsRangeMap *Class "SpatialPixelsRangeMap"*

Description

A class extending "[SpatialPixelsDataFrame](#)" used for 'MAP's retrieval from a rangeMapper project

Objects from the Class

Objects can be created by calls of the form `new("SpatialPixelsRangeMap", ...)`.

Slots

mapvar: Object of class "character"

Extends

Class "[SpatialPixelsDataFrame](#)", directly.

Methods

plot signature(x = "SpatialPixelsRangeMap", y = "missing"): ...

Note

For the moment mapvar is the only slot which makes this class distinct from the "[SpatialPixelsDataFrame](#)" class.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

[rangeMapFetch-class](#)

tkColorPalette	<i>Interactively choose a color palette.</i>
----------------	--

Description

Interactively choose a color palette. Palettes are created on the fly as `tkbuttons`.

Usage

```
tkColorPalette(pal, name, palette.size = 45, envir = .GlobalEnv)
tkMakeColorPalette(n)
```

Arguments

pal	pal a named list as e.g. returned by <code>brewer.pal.get</code> .
name	A quoted name to be assigned to <code>envir</code> .
palette.size	palette.size is the size of the tiles (<code>tkbutton</code> -s) in pixels.
envir	the <code>environment</code> to use.
n	length of the user-defined palette returned by <code>{tkMakeColorPalette</code>

Details

`tkMakeColorPalette` is called by `tkColorPalette`

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

See Also

`colorRampPalette` `brewer.pal.get`

Examples

```
## Not run:
## These cannot be run by examples() but should be OK when pasted
## into an interactive R session
pals = c(brewer.pal.get(), list(heatcol = heat.colors(9), terraincol = terrain.colors(9), topocol = topo.colors(9))

tkColorPalette(pal = pals, name = "pal1")
tkColorPalette(pal = pals, name = "pal2", palette.size = 17)
pal1
pal2

## End(Not run)
```

tkdbBrowse	<i>Browse an sqlite database.</i>
------------	-----------------------------------

Description

Browse an sqlite database using a tcltk interface.

Usage

```
tkdbBrowse(con, prefix = NULL, tables.name.only = FALSE, info)
```

Arguments

con	A valid sqlite connection.
prefix	Shows only tables with the given prefix
tables.name.only	If 'FALSE', the default, shows only table names.
info	A character string containing a short text to be shown left to the OK button.

Author(s)

Mihai Valcu <valcu@orn.mpg.de>

Examples

```
## Not run:
## These cannot be run by examples() but should be OK when pasted
## into an interactive R session

# Create an empty rangeMapper project
dbcon = rangeMap.start(file = "wrens.sqlite", overwrite = TRUE, dir = tempdir() )

# Select one table and then press OK
tkdbBrowse(dbcon, info = " select a field in one table and then push OK!")

tkdbBrowse(dbcon, tables.name.only = TRUE)

## End(Not run)
```

wrens

*Life history data of the New World Wrens***Description**

Life history data (body size, body mass and clutch size) of 84 wren (**Troglodytidae**) species

Usage

```
data(wrens)
```

Format

A data frame with 84 observations on the following 7 variables.

ID_HBW Handbook of the birds of the world ID

sci_name scientific name; a factor with 84 levels

com_name English name; a factor with 84 levels

genus Genus name

body_size body size (cm)

body_mass body mass (grams)

clutch_size mean or modal clutch size

source bibliographic source of each trait (see references)

Details

Taxonomic nomenclature follows (Kroodsma & Brewer, 2005) with the exception of *Donacoblis atricapilla* which has been excluded due to its uncertain taxonomic position.

References

- Auer, S.K., Logue, D.M., Bassar, R.D. & Gammon, D.E. (2007) Nesting biology of the Black-bellied Wren (*Thryothorus fasciatoventris*) in central Panama. *Wilson Journal of Ornithology*, 119, 71-76.
- Dunning, J.B. (2008) CRC handbook of avian body masses, 2nd edn. CRC Press, Boca Raton.
- Freeman, B.G. & Greeney, H.F. (2008) First description of the nest, eggs and cooperative breeding behavior in sharpe's wren (*Cinnycerthia olivascens*). *Ornitologia Colombiana*, 7, 88-92.
- Hron, K., Templ, M. & Filzmoser, P. (2010) Imputation of missing values for compositional data using classical and robust methods. *Computational Statistics & Data Analysis*, 54, 3095-3107 (function `impKNNa` using default arguments).
- Kroodsma, D.E. & Brewer, D. (2005) Family Troglodytidae (Wrens). Lynx Edicions, Barcelona, Spain.
- Londono, G.A. (2009) Eggs, Nests, and Incubation Behavior of the Moustached Wren (*Thryothorus genibarbis*) in Manu National Park, Peru. *Wilson Journal of Ornithology*, 121, 623-627.
- Ridgely, R.S., T. F. Allnutt, T. Brooks, D. K. McNicol, D. W. Mehlman, B. E. & Young, a.J.R.Z.

(2007) Digital Distribution Maps of the Birds of the Western Hemisphere, version 3.0. NatureServe, Arlington, Virginia, USA.

Vargas-Soriano, J., Ortiz, J.S. & Segura, G.E. (2010) Breeding Phenology and Nesting Success of the Yucatan Wren in the Yucatan Peninsula, Mexico. *Wilson Journal of Ornithology*, 122, 439-446.

See Also

[rangeMapper](#) for a complete example.

Examples

```
data(wrens)
plot(body_size ~ body_mass, wrens)
plot(clutch_size ~ log(body_mass), wrens)
```

Index

- *Topic **SQL**
 - RMQuery, 25
- *Topic **classes**
 - bioSaveDataFrame-class, 3
 - bioSaveFile-class, 4
 - gridSize, 7
 - rangeFiles-class, 11
 - rangeMap-class, 12
 - rangeMapFetch, 16
 - rangeMapProcess, 21
 - rangeMapRemove, 22
 - rangeMapSave, 23
 - rangeMapStart, 24
 - SpatialPixelsRangeMap, 27
- *Topic **color**
 - brewer.pal.get, 5
 - tkColorPalette, 28
- *Topic **database**
 - tkdbBrowse, 29
- *Topic **datasets**
 - wrens, 30
- *Topic **export**
 - rangeMap.export, 13
- *Topic **fetch**
 - rangeMap.fetch, 13
- *Topic **gui**
 - rangeMapper, 17
- *Topic **import**
 - bio.save, 2
- *Topic **macroecology**
 - rangeMapper, 17
- *Topic **methods**
 - plot-methods, 9
 - rangeMapBbox, 16
- *Topic **misc**
 - selectShpFiles, 26
- *Topic **package**
 - rangeMapper, 17
- *Topic **spatial**
 - canvas.save, 5
 - global.bbox, 6
 - gridSize.save, 8
 - processRanges, 10
 - rangeMap.save, 14
 - rangeMap.start, 15
 - rangeMapper, 17
- *Topic **sqlite**
 - canvas.save, 5
 - gridSize.save, 8
- *Topic **tktk**
 - rangeMapper, 17
 - tkColorPalette, 28
 - tkdbBrowse, 29
- bbox, 7
- bio.merge (bio.save), 2
- bio.save, 2, 3, 4, 14
- bioSave (bioSaveDataFrame-class), 3
- bioSave, bioSaveDataFrame-method (bioSaveDataFrame-class), 3
- bioSave, bioSaveFile-method (bioSaveFile-class), 4
- bioSave-methods (bioSaveDataFrame-class), 3
- bioSaveDataFrame-class, 3
- bioSaveDataFrame-class, 3
- bioSaveDataFrame-method (bioSaveDataFrame-class), 3
- bioSaveFile-class, 2, 3
- bioSaveFile-class, 4
- brewer.pal, 5
- brewer.pal.get, 5, 28
- brewer.pal.info, 5
- canvas, 10
- canvas.fetch (canvas.save), 5
- canvas.save, 5
- canvasFetch (gridSize), 7

- canvasFetch, canvasFetch-method (gridSize), 7
- canvasFetch, rangeMap-method (gridSize), 7
- canvasFetch-methods (gridSize), 7
- canvasSave (gridSize), 7
- canvasSave, canvasSave-method (gridSize), 7
- canvasSave, rangeMap-method (gridSize), 7
- canvasSave-methods (gridSize), 7
- classIntervals, 10
- colorRampPalette, 28
- data.frame, 3, 26
- environment, 28
- global.bbox, 6, 7, 8, 16
- global.bbox.fetch, 6
- global.bbox.save, 6
- gridSize, 7
- gridSize, gridSize-method (gridSize), 7
- gridSize-class, 8, 12
- gridSize-class (gridSize), 7
- gridSize-methods (gridSize), 7
- gridSize.fetch (gridSize.save), 8
- gridSize.save, 6, 8
- gridSizeFetch (gridSize), 7
- gridSizeFetch, gridSizeFetch-method (gridSize), 7
- gridSizeFetch, rangeMap-method (gridSize), 7
- gridSizeFetch-methods (gridSize), 7
- gridSizeSave (gridSize), 7
- gridSizeSave, gridSize-method (gridSize), 7
- gridSizeSave, gridSizeSave-method (gridSize), 7
- gridSizeSave-methods (gridSize), 7
- image.plot, 10
- list, 14
- MapImport (rangeMapSave), 23
- MapImport, MapImport-method (rangeMapSave), 23
- MapImport-class, 12
- MapImport-class (rangeMapSave), 23
- MapImport-methods (rangeMapSave), 23
- metadata2bio (bio.save), 2
- overlay, 10
- plot, SpatialPixelsRangeMap, missing-method (plot-methods), 9
- plot-methods, 9
- print.summary.rangeMap (rangeMap-class), 12
- processRanges, 10
- proj4string, 7
- rangeFiles, 21
- rangeFiles (rangeFiles-class), 11
- rangeFiles, rangeFiles-method (rangeFiles-class), 11
- rangeFiles-class, 26
- rangeFiles-class, 11
- rangeFiles-methods (rangeFiles-class), 11
- rangeMap, 3, 4, 7, 16, 21–23
- rangeMap-class, 12
- rangeMap.export, 13
- rangeMap.fetch, 13, 16
- rangeMap.fetch (rangeMap.fetch), 13
- rangeMap.open (rangeMap.start), 15
- rangeMap.save, 14, 23
- rangeMap.start, 15
- rangeMapBbox, 16
- rangeMapBbox, rangeFiles-method (rangeMapBbox), 16
- rangeMapBbox-method (rangeMapBbox), 16
- rangeMapBbox-methods (rangeMapBbox), 16
- rangeMapBboxFetch (rangeMapBbox), 16
- rangeMapBboxFetch, rangeMap-method (rangeMapBbox), 16
- rangeMapBboxFetch, rangeMapBboxFetch-method (rangeMapBbox), 16
- rangeMapBboxFetch-methods (rangeMapBbox), 16
- rangeMapBboxSave (rangeMapBbox), 16
- rangeMapBboxSave, rangeMap, character, CRS-method (rangeMapBbox), 16
- rangeMapBboxSave, rangeMap, character, missing-method (rangeMapBbox), 16
- rangeMapBboxSave, rangeMap, missing, CRS-method (rangeMapBbox), 16
- rangeMapBboxSave, rangeMap, missing, missing-method (rangeMapBbox), 16

- rangeMapBboxSave, rangeMap, Spatial, missing-method (rangeMapBbox), 16
- rangeMapBboxSave, rangeMapBboxSave-method (rangeMapBbox), 16
- rangeMapBboxSave-methods (rangeMapBbox), 16
- rangeMapExport (rangeMap.export), 13
- rangeMapExport, rangeMap, character-method (rangeMap.export), 13
- rangeMapExport, rangeMapExport-methods (rangeMap.export), 13
- rangeMapExport-methods (rangeMap.export), 13
- rangeMapFetch, 16
- rangeMapFetch, rangeMapFetch-method (rangeMapFetch), 16
- rangeMapFetch-class, 12, 27
- rangeMapFetch-class (rangeMapFetch), 16
- rangeMapFetch-methods (rangeMapFetch), 16
- rangeMapImport (rangeMapSave), 23
- rangeMapImport, MapImport, function-method (rangeMapSave), 23
- rangeMapImport, rangeMapImport-method (rangeMapSave), 23
- rangeMapImport-methods (rangeMapSave), 23
- rangeMapper, 3, 6–8, 10, 11, 14, 15, 17, 22, 24–26, 31
- rangeMapProcess, 21, 22
- rangeMapProcess, rangeMapProcess-method (rangeMapProcess), 21
- rangeMapProcess-class, 12
- rangeMapProcess-class (rangeMapProcess), 21
- rangeMapProcess-methods (rangeMapProcess), 21
- rangeMapRemove, 22, 24
- rangeMapRemove, rangeMapRemove-method (rangeMapRemove), 22
- rangeMapRemove-class, 12
- rangeMapRemove-class (rangeMapRemove), 22
- rangeMapRemove-methods (rangeMapRemove), 22
- rangeMapSave, 3, 4, 23, 23
- rangeMapSave, rangeMapSave, character, missing-method (rangeMapSave), 23
- rangeMapSave, rangeMapSave, function, formula-method (rangeMapSave), 23
- rangeMapSave, rangeMapSave, function, missing-method (rangeMapSave), 23
- rangeMapSave, rangeMapSave, missing, missing-method (rangeMapSave), 23
- rangeMapSave, rangeMapSave-method (rangeMapSave), 23
- rangeMapSave-class, 12
- rangeMapSave-class (rangeMapSave), 23
- rangeMapSave-methods (rangeMapSave), 23
- rangeMapStart, 24
- rangeMapStart, rangeMapStart-method (rangeMapStart), 24
- rangeMapStart-class, 12, 15
- rangeMapStart-class (rangeMapStart), 24
- rangeMapStart-methods (rangeMapStart), 24
- rm.rangeMapper, 22, 24
- RMQuery, 25
- selectShpFiles, 11, 26
- SpatialPixelsDataFrame, 5, 27
- SpatialPixelsRangeMap, 27
- SpatialPixelsRangeMap, SpatialPixelsRangeMap, missing-method (SpatialPixelsRangeMap), 27
- SpatialPixelsRangeMap, SpatialPixelsRangeMap-method (SpatialPixelsRangeMap), 27
- SpatialPixelsRangeMap-class, 13, 14, 16
- SpatialPixelsRangeMap-class (SpatialPixelsRangeMap), 27
- SpatialPolygonsDataFrame, 6
- spplot, 9, 10
- sqliteQuickSQL, 25
- summary, rangeMap-method (rangeMap-class), 12
- summary.rangeMap (rangeMap-class), 12
- tkbutton, 28
- tkColorPalette, 5, 10, 28
- tkdbBrowse, 29
- tkMakeColorPalette (tkColorPalette), 28
- wrens, 3, 30
- writeGDAL, 13