

Package ‘sdmApp’

January 22, 2021

Title A User-Friendly Application for Species Distribution Modeling

Version 0.0.1

Author Aboubacar HEMA [aut, cre],
Babacar NDAO [ctb],
Louise LEROUX [aut],
Abdoul Aziz DIOUF [ctb]

Maintainer Aboubacar HEMA <aboubacarhema94@gmail.com>

Description A 'shiny'
application that allows non-expert 'R' users to easily model
species distribution. It offers a reproducible work flow for
species distribution modeling into a single and user friendly environment.
'sdmApp' takes 'raster' data (in format supported by the 'raster package')
and species occurrence data (several format supported) as input argument.
It provides an interactive graphical user interface (GUI).

License GPL-3

URL <https://github.com/Abson-dev/sdmApp>

BugReports <https://github.com/Abson-dev/sdmApp/issues>

Depends R (>= 3.5.0)

Imports raster (>= 2.6.7), sp (>= 1.2.0), biomod2 (>= 3.4.6), blockCV
(>= 2.1.1), CENFA (>= 1.1.0), dismo (>= 1.0.12), DT, kernlab
(>= 0.9-29), randomForest (>= 4.6.10), readxl (>= 1.3.1),
rhandsontable (>= 0.3.7), sf, shiny (>= 0.12.2), shinyBS (>=
0.61), shinyFiles (>= 0.7.0), SSDM (>= 0.2.8), ggcorrplot (>=
0.1.3), ggplot2 (>= 3.1.1), ggpubr (>= 0.4.0), haven (>=
2.3.1), tidyverse (>= 1.3.0), data.table, rgeos (>= 0.3-8),
rJava (>= 0.9-13)

Suggests covr, grDevices, knitr, rmarkdown, stats, testthat, utils,
rgdal (>= 1.5-8), automap (>= 1.0-14), graphics, future.apply

VignetteBuilder knitr, rmarkdown

Encoding UTF-8

Language en-US

LazyData true**RoxygenNote** 7.1.1**SystemRequirements** Java (>= 8)**NeedsCompilation** no**Repository** CRAN**Date/Publication** 2021-01-22 10:40:03 UTC

R topics documented:

sdmApp	2
sdmApp_fold_Explorer	3
sdmApp_PA	4
sdmApp_RasterPlot	5
sdmApp_TimesRasters	5
Index	7

sdmApp	<i>starts the graphical user interface developed with shiny.</i>
--------	--

Description

starts the graphical user interface developed with shiny.

Usage

```
sdmApp(
  maxRequestSize = 50,
  debug = FALSE,
  theme = "IHSN",
  ...,
  shiny.server = FALSE
)
```

Arguments

maxRequestSize	(numeric) number defining the maximum allowed file size (in megabytes) for uploaded files, defaults to 50MB
debug	logical if TRUE, set shiny-debugging options
theme	select style sheet for the interface.
...	arguments (e.g host) that are passed through <code>runApp</code> when starting the shiny application
shiny.server	Setting this parameter to TRUE will return the app in the form of an object rather than invoking it. This is useful for deploying <code>sdmApp</code> via shiny-server.

Value

starts the interactive graphical user interface.

Examples

```
if(interactive()){  
  #load the package  
  library(sdmApp)  
  sdmApp()  
}
```

sdmApp_fold_Explorer *Explore the generated folds and visualize the placement of folds and distribution of species data over folds.*

Description

Explore the generated folds and visualize the placement of folds and distribution of species data over folds.

Usage

```
sdmApp_fold_Explorer(blocks, rasterLayer, speciesData, num)
```

Arguments

blocks	A SpatialBlock object.
rasterLayer	A raster object as background map for visualization.
speciesData	A simple features (sf) or SpatialPoints object containing species data (response variable).
num	A number of fold to assign as data test set.

Value

A map showing folds and the species data, that can be used to explore folds.

See Also

[foldExplorer](#)

Examples

```
# load blockCV package data
library(blockCV)
awt <- raster::brick(system.file("extdata", "awt.grd", package = "blockCV"))
#import presence-absence species data
PA <- read.csv(system.file("extdata", "PA.csv", package = "blockCV"))
#make a sf object from data.frame
pa_data <- sf::st_as_sf(PA, coords = c("x", "y"), crs = raster::crs(awt))
#spatial blocking by specified range and random assignment
sb <- spatialBlock(speciesData = pa_data, species = "Species",
  rasterLayer = awt, theRange = 70000, k = 5,
  selection = "random", iteration = 100)
sdmApp_fold_Explorer(sb, awt, pa_data, 1)
```

sdmApp_PA

Plot presence/absence map

Description

Plot presence/absence map

Usage

```
sdmApp_PA(x)
```

Arguments

x Raster object

Value

a ggplot object

Examples

```
r <- raster::raster(system.file("extdata", "AETI.tif", package = "sdmApp"))
r <- r > 4000
sdmApp_PA(r)
```

sdmApp_RasterPlot *Plot a raster*

Description

Plot a raster

Usage

```
sdmApp_RasterPlot(x)
```

Arguments

x Raster object

Value

a ggplot object

Examples

```
r <- raster::raster(system.file("extdata", "AETI.tif", package = "sdmApp"))
sdmApp_RasterPlot(r)
```

sdmApp_TimesRasters *Multiply the probability of occurrence map with the presence/absence map to get a presence map only.*

Description

Multiply the probability of occurrence map with the presence/absence map to get a presence map only.

Usage

```
sdmApp_TimesRasters(x, y)
```

Arguments

x Probability of occurrence map, a Raster object
y Presence/Absence map, a Raster object

Value

Probability of occurrence map with only presence

Examples

```
r <- raster::raster(system.file("extdata", "AETI.tif", package = "sdmApp"))
r2 <- r > raster::cellStats(r, stat='mean', na.rm=TRUE)
r <- r/raster::maxValue(r)
names(r) <- "propability of occurence"
z<-sdmApp_TimesRasters(r,r2)
sdmApp_RasterPlot(z)
```

Index

[foldExplorer](#), 3

[runApp](#), 2

[sdmApp](#), 2

[sdmApp_fold_Explorer](#), 3

[sdmApp_PA](#), 4

[sdmApp_RasterPlot](#), 5

[sdmApp_TimesRasters](#), 5