Package: malariaAtlas (via r-universe)

November 16, 2024

Title An R Interface to Open-Access Malaria Data, Hosted by the 'Malaria Atlas Project'

Version 1.6.1.9999

Description A suite of tools to allow you to download all publicly available parasite rate survey points, mosquito occurrence points and raster surfaces from the 'Malaria Atlas Project' <https://malariaatlas.org/> servers as well as utility functions for plotting the downloaded data.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports xml2, gridExtra, httr, dplyr, tidyr, methods, stats, utils, rlang, sf, lifecycle, terra, tidyterra, ows4R, future.apply, lubridate, jsonlite, stringr, ggnewscale

Depends ggplot2

RoxygenNote 7.2.3

Suggests testthat, knitr, rmarkdown, palettetown, magrittr, tibble, rdhs (>= 0.8.0)

URL https://github.com/malaria-atlas-project/malariaAtlas

BugReports https://github.com/malaria-atlas-project/malariaAtlas/issues

VignetteBuilder knitr

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libicu-dev libsecret-1-dev libsodium-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev

Repository https://epiverse-connect.r-universe.dev

RemoteUrl https://github.com/malaria-atlas-project/malariaAtlas

RemoteRef HEAD

RemoteSha 76223bc0c22ad59c2e33cdf04ef61d7aea96297a

Contents

`as.MAPraster` *Convert Raster objects into MAPraster objects*

Description

`as.MAPraster` converts a RasterLayer or RasterStack object into a 'MAPraster' object (data.frame) for easy plotting with ggplot.

Usage

```
as.MAPraster(raster_object)
```

Arguments

`raster_object`	RasterLayer or Rasterstack object to convert into a MAPraster.

Value

`as.MAPraster` returns a MAPraster object (data.frame) containing the below columns.

1. `x` - x coordinates of raster pixels
2. `y` - y coordinates of raster pixels
3. `z` - value of raster pixels
4. `raster_name` - name of raster for which values are stored in z

See Also

`getRaster`:

to download rasters directly from MAP.

`as.MAPraster`:

to convert RasterLayer/RasterStack objects into a 'MAPraster' object (data.frame) for easy plotting with ggplot.

`autoplot.MAPraster`:

to quickly visualise MAPraster objects created using `as.MAPraster`.

Examples

```
# Download PfPR2-10 Raster for Madagascar in 2015 and visualise this on a map.
## Not run:
MDG_shp <- getShp(ISO = "MDG", admin_level = "admin0")
MDG_PfPR2_10 <- getRaster(surface = "Plasmodium falciparum PR2-10", shp = MDG_shp, year = 2015)
MDG_PfPR2_10 <- as.MAPraster(MDG_PfPR2_10)
autoplot(MDG_PfPR2_10)
## End(Not run)

#Download global raster of G6PD deficiency from Howes et al 2012 and visualise this on a map.
```

```
## Not run:
G6PDd_global <- getRaster(surface = "G6PD Deficiency Allele Frequency")
G6PDd_global <- as.MAPraster(G6PDd_global)
autoplot(G6PDd_global)
## End(Not run)
```

---

as.MAPshp *Convert SpatialPolygon objects into MAPshp objects*

Description

as.MAPshp converts a SpatialPolygon or SpatialPolygonsDataframe object downloaded using getShp into a 'MAPshp object (data.frame) for easy plotting with ggplot.

Usage

```
as.MAPshp(object)
```

Arguments

object	SpatialPolygon or SpatialPolygonsDataframe object to convert into a 'MAPshp'.

Value

as.MAPshp returns a MAPshp object (data.frame) containing the below columns.

1. country_id ISO-3 code of given administrative unit (or the ISO code of parent unit for administrative-level 1 units).
2. gaul_code GAUL code of given administrative unit.
3. admn_level administrative level of the given administrative unit - either 0 (national) or 1 (first-level division)
4. parent_id GAUL code of parent administrative unit of a given polygon (for admin0 polygons, PARENT_ID = 0).
5. country_level composite country_id_admn_level field.

See Also

autoplot.MAPshp

to download rasters directly from MAP.

Examples

```
#Download shapefiles for Madagascar and visualise these on a map.

## Not run:
MDG_shp <- getShp(ISO = "MDG", admin_level = "admin0")
MDG_shp <- as.MAPshp(MDG_shp)
autoplot(MDG_shp)

## End(Not run)
```

`as.pr.points` *Convert data.frames to pr.points objects.*

Description

Will create empty columns for any missing columns expected in a pr.points data.frame. This function is particularly useful for use with packages like dplyr that strip objects of their classes.

Usage

```
as.pr.points(x)
```

Arguments

x	A data.frame

Examples

```
#Download PfPR data for Nigeria and Cameroon and map the locations of these points using autoplot
## Not run:
library(dplyr)
NGA_CMR_PR <- getPR(country = c("Nigeria", "Cameroon"), species = "Pf")

# Filter the data frame then readd pr.points class so that autoplot can be used.
NGA_CMR_PR %>%
  filter(year_start > 2010) %>%
  as.pr.points %>%
  autoplot


## End(Not run)
```

as.vectorpoints *Convert data.frames to vector.points objects.*

Description

Will create empty columns for any missing columns expected in a vector.points data.frame. This function is particularly useful for use with packages like dplyr that strip objects of their classes.

Usage

```
as.vectorpoints(x)
```

Arguments

x	A data.frame

Examples

```
## Not run:
library(dplyr)
Brazil_vec <- getVecOcc(country = "Brazil")

# Filter data.frame then readd vector points class so autoplot can be used.
Brazil_vec %>%
  filter(sample_method1 == 'larval collection') %>%
  as.vectorpoints %>%
  autoplot

## End(Not run)
```

autoplot.MAPraster *Quickly visualise Rasters downloaded from MAP*

Description

`autoplot.MAPraster` creates a map of all rasters in a MAPraster object and displays these in a grid.

Usage

```
## S3 method for class 'MAPraster'
autoplot(
  object,
  ...,
  shp_df = NULL,
  legend_title = "",
```

```
  plot_titles = TRUE,
  fill_scale_transform = "identity",
  fill_colour_palette = "RdYlBu",
  printed = TRUE
)
```

Arguments

object	MAPraster object to be visualised.
...	Other arguments passed to specific methods
shp_df	Shapefile(s) (data.frame) to plot with downloaded raster.
legend_title	String used as title for all colour scale legends.
plot_titles	Plot name of raster object as header for each individual raster plot?
fill_scale_transform	String givning a transformation for the fill aesthetic. See the trans argument in continuous_scale for possible values.
fill_colour_palette	String referring to a colorbrewer palette to be used for raster colour scale.
printed	Logical vector indicating whether to print maps of supplied rasters.

Value

autoplot.MAPraster returns a list of plots (gg objects) for each supplied raster.

See Also

getRaster:

to download rasters directly from MAP.

as.MAPraster:

to convert RasterLayer/RasterStack objects into a 'MAPraster' object (data.frame) for easy plotting with ggplot.

autoplot.MAPraster:

to quickly visualise MAPraster objects created using as.MAPraster.

Examples

```
## Not run:
# Download PfPR2-10 Raster (Bhatt et al 2015) and raw survey points
#   for Madagascar in 2013 and visualise these together on a map.

# Download madagascar shapefile to use for raster download.
MDG_shp <- getShp(ISO = "MDG", admin_level = "admin0")

# Download PfPR2-10 Raster for 2013 & plot this
MDG_PfPR2_10 <- getRaster(surface = "Plasmodium falciparum PR2-10",
                          shp = MDG_shp, year = 2013)
p <- autoplot(MDG_PfPR2_10, shp_df = MDG_shp)
```

```
# Download raw PfPR survey points & plot these over the top of the raster
pr <- getPR(country = c("Madagascar"), species = "Pf")

# Download global raster of G6PD deficiency (Howes et al 2012) and visualise this on a map.
G6PDd_global <- getRaster(surface = "G6PD Deficiency Allele Frequency")
autoplot(G6PDd_global)

## End(Not run)
```

`autoplot.MAPshp` *Create a basic plot to visualise downloaded shapefiles*

Description

`autoplot.MAPshp` creates a map of shapefiles downloaded using getShp.

Usage

```
## S3 method for class 'MAPshp'
autoplot(object, ..., map_title = NULL, facet = FALSE, printed = TRUE)
```

Arguments

`object`	A MAPshp object downloaded using `getShp` with format = "df" specified.
`...`	Other arguments passed to specific methods
`map_title`	Custom title used for the plot.
`facet`	If TRUE, splits map into a separate facet for each administrative level.
`printed`	Should the plot print to graphics device.

Value

`autoplot.MAPshp` returns a map (gg object) of the supplied MAPShp dataframe.

Examples

```
## Not run:
MDG_shp <- getShp(ISO = "MDG", admin_level = "admin0")
autoplot(as.MAPshp(MDG_shp))

## End(Not run)
```

`autoplot.pr.points` *Create a basic plot showing locations of downloaded PR points*

Description

`autoplot.pr.points` creates a map of PR points downloaded from MAP.

Usage

```
## S3 method for class 'pr.points'
autoplot(
  object,
  ...,
  shp_df = NULL,
  admin_level = "admin0",
  map_title = "PR Survey Locations",
  fill_legend_title = "Raw PR",
  fill_scale_transform = "identity",
  facet = NULL,
  hide_confidential = FALSE,
  printed = TRUE
)
```

Arguments

`object`	a pr.points object downloaded using `getPR`
`...`	Other arguments passed to specific methods
`shp_df`	Shapefile(s) (data.frame) to plot with downloaded points. (If not specified automatically uses getShp() for all countries included in pr.points object).
`admin_level`	the administrative level used for plotting administrative boundaries; either `"admin0"`; `"admin1"` OR `"both"`
`map_title`	custom title used for the plot
`fill_legend_title`	Add a title to the legend.
`fill_scale_transform`	String givning a transformation for the fill aesthetic. See the trans argument in `continuous_scale` for possible values.
`facet`	if TRUE, splits map into a separate facet for each malaria species; by default FALSE if only one species is present in object, TRUE if both P. falciparum and P. vivax data are present in object.
`hide_confidential`	if TRUE, removes confidential points from the map
`printed`	Should the plot be printed to the graphics device.

Value

`autoplot.pr.points` returns a plots (gg object) for the supplied pr.points dataframe.

Examples

```
## Not run:
PfPR_surveys_NGA <- getPR(country = c("Nigeria"), species = "Pf")
autoplot(PfPR_surveys_NGA)

# Download PfPR2-10 Raster (Bhatt et al. 2015) and raw survey points for Madagascar in
#   2013 and visualise these together on a map.

# Download madagascar shapefile to use for raster download.
MDG_shp <- getShp(ISO = "MDG", admin_level = "admin0")

# Download PfPR2-10 Raster for 2013 & plot this
MDG_PfPR2_10 <- getRaster(surface = "Plasmodium falciparum PR2-10", shp = MDG_shp, year = 2013)
p <- autoplot(MDG_PfPR2_10)

# Download raw PfPR survey points & plot these over the top of the raster
pr <- getPR(country = c("Madagascar"), species = "Pf")
# p[[1]] +
# geom_point(data = pr[pr$year_start==2013,],
#            aes(longitude, latitude, fill = positive / examined,
#                size = examined), shape = 21) +
#   scale_size_continuous(name = "Survey Size") +
#   scale_fill_distiller(name = "PfPR", palette = "RdYlBu") +
#   ggtitle("Raw PfPR Survey points\n + Modelled PfPR 2-10 in Madagascar in 2013")

## End(Not run)
```

`autoplot.sf` *Create a basic plot to visualise downloaded shapefiles*

Description

`autoplot.sf` creates a map of shapefiles downloaded using getShp.

Usage

```
## S3 method for class 'sf'
autoplot(object, ..., map_title = NULL, facet = FALSE, printed = TRUE)
```

Arguments

`object`	A sf object downloaded using `getShp`.
`...`	Other arguments passed to specific methods

map_title	Custom title used for the plot.
facet	If TRUE, splits map into a separate facet for each administrative level.
printed	Should the plot print to graphics device.

Value

autoplot.sf returns a map of the supplied sf object

Examples

```
## Not run:
MDG_shp <- getShp(ISO = "MDG", admin_level = "admin0")
autoplot(MDG_shp)

## End(Not run)
```

---

autoplot.SpatRaster *Quickly visualise Rasters downloaded from MAP*

Description

autoplot.SpatRaster creates a map of all rasters in a SpatRaster object and displays these in a grid.

Usage

```
## S3 method for class 'SpatRaster'
autoplot(
  object,
  ...,
  shp_df = NULL,
  legend_title = "",
  plot_titles = TRUE,
  fill_scale_transform = "identity",
  fill_colour_palette = "RdYlBu",
  printed = TRUE
)
```

Arguments

object	SpatRaster object to be visualised.
...	Other arguments passed to specific methods
shp_df	Shapefile(s) (data.frame) to plot with downloaded raster.
legend_title	String used as title for all colour scale legends.
plot_titles	Plot name of raster object as header for each individual raster plot?

`fill_scale_transform`
: String givning a transformation for the fill aesthetic. See the trans argument in `continuous_scale` for possible values.

`fill_colour_palette`
: String referring to a colorbrewer palette to be used for raster colour scale.

`printed`
: Logical vector indicating whether to print maps of supplied rasters.

Value

`autoplot.SpatRaster` returns a list of plots (gg objects) for each supplied raster.

See Also

`getRaster`:

to download rasters directly from MAP.

Examples

```
## Not run:
# Download PfPR2-10 Raster (Bhatt et al 2015) and raw survey points
#   for Madagascar in 2013 and visualise these together on a map.

# Download madagascar shapefile to use for raster download.
MDG_shp <- getShp(ISO = "MDG", admin_level = "admin0")

# Download PfPR2-10 Raster for 2013 & plot this
MDG_PfPR2_10 <- getRaster(surface = "Plasmodium falciparum PR2-10",
                          shp = MDG_shp, year = 2013)
p <- autoplot(MDG_PfPR2_10, shp_df = MDG_shp)

# Download raw PfPR survey points & plot these over the top of the raster
pr <- getPR(country = c("Madagascar"), species = "Pf")
# p[[1]] + geom_point(data = pr[pr$year_start==2013,],
#            aes(longitude, latitude, fill = positive / examined,
#                size = examined), shape = 21) +
#   scale_size_continuous(name = "Survey Size") +
#    scale_fill_distiller(name = "PfPR", palette = "RdYlBu") +
#    ggtitle("Raw PfPR Survey points\n +
#          Modelled PfPR 2-10 in Madagascar in 2013")


# Download global raster of G6PD deficiency (Howes et al 2012) and visualise this on a map.
G6PDd_global <- getRaster(surface = "G6PD Deficiency Allele Frequency")
autoplot(G6PDd_global)

## End(Not run)
```

`autoplot.SpatRasterCollection` *Quickly visualise Rasters downloaded from MAP*

Description

`autoplot.SpatRasterCollection` creates a map of all rasters in a autoplot.SpatRasterCollection object and displays these in a grid.

Usage

```
## S3 method for class 'SpatRasterCollection'
autoplot(
  object,
  ...,
  shp_df = NULL,
  legend_title = "",
  plot_titles = TRUE,
  fill_scale_transform = "identity",
  fill_colour_palette = "RdYlBu",
  printed = TRUE
)
```

Arguments

`object`	SpatRasterCollection object to be visualised.
`...`	Other arguments passed to specific methods
`shp_df`	Shapefile(s) (data.frame) to plot with downloaded raster.
`legend_title`	String used as title for all colour scale legends.
`plot_titles`	Plot name of raster object as header for each individual raster plot?
`fill_scale_transform`	String givning a transformation for the fill aesthetic. See the trans argument in `continuous_scale` for possible values.
`fill_colour_palette`	String referring to a colorbrewer palette to be used for raster colour scale.
`printed`	Logical vector indicating whether to print maps of supplied rasters.

Value

`autoplot.SpatRasterCollection` returns a list of plots (gg objects) for each supplied raster.

gg object

`autoplot.vector.points` *Create a basic plot showing locations of downloaded Vector points*

Description

`autoplot.vector.points` creates a map of Vector points downloaded from MAP.

Usage

```
## S3 method for class 'vector.points'
autoplot(
  object,
  ...,
  shp_df = NULL,
  admin_level = "admin0",
  map_title = "Vector Survey Locations",
  fill_legend_title = "Raw Vetor Occurrences",
  fill_scale_transform = "identity",
  facet = NULL,
  printed = TRUE
)
```

Arguments

`object`	a vector.points object downloaded using `getVecOcc`
`...`	Other arguments passed to specific methods
`shp_df`	Shapefile(s) (data.frame) to plot with downloaded points. (If not specified automatically uses getShp() for all countries included in vector.points object).
`admin_level`	the administrative level used for plotting administrative boundaries; either `"admin0"`; `"admin1"` OR `"both"`
`map_title`	custom title used for the plot
`fill_legend_title`	Add a title to the legend.
`fill_scale_transform`	String givning a transformation for the fill aesthetic. See the trans argument in `continuous_scale` for possible values.
`facet`	if TRUE, splits map into a separate facet for each malaria species; by default FALSE.
`printed`	Should the plot be printed to the graphics device.

Value

`autoplot.vector.points` returns a plots (gg object) for the supplied vector.points dataframe.

Examples

```
## Not run:
Vector_surveys_NGA_NG <- getVecOcc(country = c("Nigeria", "Niger"))
autoplot(Vector_surveys_NGA_NG)

# Download the predicted distribution of An. dirus species complex Raster and
#  vector points for Myanmar and visualise these together on a map.

# Download Myanmar shapefile to use for raster download.
MMR_shp <- getShp(ISO = "MMR", admin_level = "admin0")

# Download An. dirus predicted distribution Raster & plot this
MMR_An_dirus <- getRaster(surface = "Anopheles dirus species complex", shp = MMR_shp)
p <- autoplot(MMR_An_dirus, shp_df = MMR_shp, printed = FALSE)

# Download raw occurrence points & plot these over the top of the raster
species <- getVecOcc(country = "Myanmar", species = "Anopheles dirus")
# p[[1]] +
# geom_point(data = species,
#  aes(longitude,
#   latitude,
#   colour = species))+
#   scale_colour_manual(values = "black", name = "Vector suvery locations")+
# scale_fill_distiller(name = "Predicted distribution of An. dirus complex",
#  palette = "PuBuGn",
#   direction = 1)+
#   ggtitle("Vector Survey points\n + The predicted distribution of An. dirus complex")

## End(Not run)
```

`autoplot_MAPraster` *Quickly visualise Rasters downloaded from MAP*

Description

`autoplot_MAPraster` is a wrapper for `autoplot.MAPraster` that calls `as.MAPraster` to allow automatic map creation for RasterLayer/RasterStack objects downloaded from MAP.

Usage

```
autoplot_MAPraster(object, ...)
```

Arguments

`object`	RasterLayer/RasterStack object to be visualised.
`...`	other optional arguments to autoplot.MAPraster (e.g. shp_df, legend_title, page_title...)

Value

autoplot_MAPraster returns a list of plots (gg objects) for each supplied raster.

See Also

getRaster:

to download rasters directly from MAP.

as.MAPraster:

to convert RasterLayer/RasterStack objects into a 'MAPraster' object (data.frame) for easy plotting with ggplot.

autoplot.MAPraster:

to quickly visualise MAPraster objects created using as.MAPraster.

Examples

```
## Not run:
#Download PfPR2-10 Raster (Bhatt et al 2015) and raw survey points for Madagascar in
#  2013 and visualise these together on a map.

# Download madagascar shapefile to use for raster download.
MDG_shp <- getShp(ISO = "MDG", admin_level = "admin0")

# Download PfPR2-10 Raster for 2013 & plot this
MDG_PfPR2_10 <- getRaster(surface = "Plasmodium falciparum PR2-10", shp = MDG_shp, year = 2013)
# p <- autoplot_MAPraster(MDG_PfPR2_10)

# Download raw PfPR survey points & plot these over the top of the raster
pr <- getPR(country = c("Madagascar"), species = "Pf")
# p[[1]] +
# geom_point(data = pr[pr$year_start==2013,],
#        aes(longitude, latitude, fill = positive / examined, size = examined), shape = 21) +
#   scale_size_continuous(name = "Survey Size") +
#   scale_fill_distiller(name = "PfPR", palette = "RdYlBu") +
#   ggtitle("Raw PfPR Survey points\n + Modelled PfPR 2-10 in Madagascar in 2013")
## End(Not run)


# Download global raster of G6PD deficiency (Howes et al 2012) and visualise this on a map.
## Not run:
G6PDd_global <- getRaster(surface = "G6PD Deficiency Allele Frequency")
#autoplot_MAPraster(G6PDd_global)

## End(Not run)
```

`convertPrevalence` *convert prevalences from one age range to another*

Description

convert prevalences from one age range to another

Usage

```
convertPrevalence(
  prevalence,
  age_min_in,
  age_max_in,
  age_min_out = rep(2, length(prevalence)),
  age_max_out = rep(9, length(prevalence)),
  parameters = "Pf_Smith2007",
  sample_weights = NULL
)
```

Arguments

`prevalence`	Vector of prevalence values
`age_min_in`	Vector of minimum ages sampled
`age_max_in`	Vector maximum ages sampled.
`age_min_out`	Length 1 numeric or vector of same length as prevalence given the required age range upper bound
`age_max_out`	Length 1 numeric or vector of same length as prevalence given the required age range lower bound
`parameters`	Specifies the set of parameters to use in the model. This can either be "Pf_Smith2007" to use the parameters for *Plasmodium falciparum* defined in that paper, "Pv_Gething2012" for the *P. vivax* parameters used in that paper, or a user-specified vector givng the values of parameters ‘b‘, ‘s‘, ‘c‘ and ‘alpha‘, in that order. If specified,
`sample_weights`	Must be a vector of length 85 giving the sample weights for each age category (the proportion of the population in that age category) . If ‘NULL‘, The sample weights used in Smith et al. 2007 are used.

References

Smith, D. L. et al. Standardizing estimates of the Plasmodium falciparum parasite rate. Malaria Journal 6, 131 (2007).

Gething, Peter W., et al. "A long neglected world malaria map: Plasmodium vivax endemicity in 2010." PLoS neglected tropical diseases 6.9 (2012): e1814.

Code written by Nick Golding and Dave Smith

Examples

```
# Convert from prevalence 2 to 5 to prevalence 2 to 10
pr2_10 <- convertPrevalence(0.1, 2, 5, 2, 10)

# Convert many surveys all to 2 to 10.
pr <- runif(20, 0.1, 0.15)
min_in <- sample(1:5, 20, replace = TRUE)
max_in <- rep(8, 20)
min_out <- rep(2, 20)
max_out <- rep(10, 20)
pr_standardised <- convertPrevalence(pr, min_in, max_in,
                                     min_out, max_out)

plot(pr_standardised, pr)
```

`download_rst` *Download rasters from the MAP geoserver to a specifed location. If file already exists it will read it instead.*

Description

Download rasters from the MAP geoserver to a specifed location. If file already exists it will read it instead.

Usage

```
download_rst(dataset_id, extent, year, file_name, file_path)
```

Arguments

`dataset_id`	ID for dataset on MAP geoserver
`extent`	desired raster extent
`year`	desired year to download
`file_name`	file name (excluding extension) to save raster to
`file_path`	path to save raster to

Value

SpatRaster

`extractRaster` *Extract pixel values from MAP rasters using point coordinates.*

Description

`extractRaster` extracts pixel values from MAP rasters at user-specified point locations (without downloading the entire raster).

Usage

```
extractRaster(
  df,
  csv_path = NULL,
  surface = NULL,
  year = NULL,
  dataset_id = NULL
)
```

Arguments

`df`	data.frame containing coordinates of input point locations, must contain columns named 'latitude'/'lat'/'x' AND 'longitude'/'long'/'y')
`csv_path`	(optional) user-specified path to which extractRaster coordinates and results are stored.
`surface`	deprecated argument. Please remove it from your code.
`year`	for time-varying rasters: if downloading a single surface for one or more years, `year` should be a vector specifying the desired year(s). if downloading more than one surface, use a list the same length as `surface`, providing the desired year-range for each time-varying surface in `surface` or `NA` for static rasters.
`dataset_id`	A character string specifying the dataset ID(s) of one or more rasters. These dataset ids can be found in the data.frame returned by listRaster, in the dataset_id column e.g. c('Malaria__202206_Global_Pf_Mortality_Count', 'Malaria__202206_Global_Pf_Parasite_

Value

`extractRaster` returns the input dataframe (`df`), with the following columns appended, providing values for each raster, location and year.

1. `layerName` dataset id corresponding to extracted raster values for a given row, check `listRaster` for raster metadata.
2. `year` the year for which raster values were extracted (time-varying rasters only; static rasters do not have this column).
3. `value` the raster value for the pixel in which a given point location falls.

See Also

`autoplot` method for quick mapping of PR point locations (`autoplot.pr.points`).

Examples

```
#Download PfPR data for Nigeria and Cameroon and map the locations of these points using autoplot
## Not run:
# Get some data and remove rows with NAs in location or examined or positive columns.
NGA_CMR_PR <- getPR(country = c("Nigeria", "Cameroon"), species = "Pf")
complete <- complete.cases(NGA_CMR_PR[, c(4, 5, 16, 17)])
NGA_CMR_PR <- NGA_CMR_PR[complete, ]

# Extract PfPR data at those locations.
data <- extractRaster(df = NGA_CMR_PR[, c('latitude', 'longitude')],
                      dataset_id = 'Malaria__202206_Global_Pf_Parasite_Rate',
                      year=2020)

# Some rasters are stored with NA encoded as -9999
data$value[data$value == -9999] <- NA

# We can quickly plot a summary
plot((NGA_CMR_PR$positive / NGA_CMR_PR$examined) ~ data$value)

## End(Not run)
```

---

fillDHSCoordinates *Add DHS locations to malaria data*

Description

We cannot directly share DHS data. We can share coordinates, but not the data values. This function attempts to fill the data gaps directly from the DHS server using the package rdhs. To use the function you will need to setup an account on the DHS website and request any datasets you wish to use (including requesting the GPS data). Confirmation can take a few days. Once this has been verified, you should be able to use this function.

Usage

```
fillDHSCoordinates(
  data,
  email = NULL,
  project = NULL,
  cache_path = NULL,
  config_path = NULL,
  global = TRUE,
  verbose_download = FALSE,
  verbose_setup = TRUE,
  data_frame = NULL,
  timeout = 30,
  password_prompt = FALSE,
  prompt = TRUE
)
```

Arguments

data	Data to add DHS coordinates to
email	Character for email used to login to the DHS website.
project	Character for the name of the DHS project from which datasets should be downloaded.
cache_path	Character for directory path where datasets and API calls will be cached. If left bank, a suitable directory will be created within your user cache directory for your operating system (permission granting).
config_path	Character for where the config file should be saved. For a global configuration, 'config_path' must be '~/.rdhs.json'. For a local configuration, 'config_path' must be 'rdhs.json'. If left bank, the config file will be stored within your user cache directory for your operating system (permission granting).
global	Logical for the config_path to be interpreted as a global config path or a local one. Default = TRUE.
verbose_download	Logical for dataset download progress bars to be shown. Default = FALSE.
verbose_setup	Logical for rdhs setup and messages to be printed. Default = TRUE.
data_frame	Function with which to convert API calls into. If left blank `data_frame` objects are returned. Must be passed as a character. Examples could be: `data.table::as.data.table` `tibble::as.tibble`
timeout	Numeric for how long in seconds to wait for the DHS API to respond. Default = 30.
password_prompt	Logical whether user is asked to type their password, even if they have previously set it. Default = FALSE. Set to TRUE if you have mistyped your password when using `set_rdhs_config`.
prompt	Logical for whether the user should be prompted for permission to write to files. This should not need be

Details

This function requires the package `rdhs` which is currently only suggested by the package (not a dependency). So you will need to install it.

Note that the `project` has to be the exact name in your DHS project.

Author(s)

OJ Watson

Examples

```
## Not run:
pf <- malariaAtlas::getPR("all",species = "pf")
pf <- fillDHSCoordinates(pf,
email = "pretend_email@emaildomain.com",
project = "pretend project name")
```

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

---

getPR *Download PR points from the MAP database*

Description

getPR downloads all publicly available PR points for a specified country (or countries) and returns this as a dataframe.

Usage

```
getPR(
  country = NULL,
  ISO = NULL,
  continent = NULL,
  species = NULL,
  extent = NULL,
  start_date = NULL,
  end_date = NULL,
  version = NULL
)
```

Arguments

country	string containing name of desired country, e.g. c("Country1", "Country2", ...) OR = "ALL". (Use one of country OR ISO OR continent, not combined)
ISO	string containing ISO3 code for desired country, e.g. c("XXX", "YYY", ...) OR = "ALL". (Use one of country OR ISO OR continent, not combined)
continent	string containing continent (one of "Africa", "Americas", "Asia", "Oceania") for desired data, e.g. c("continent1", "continent2", ...). (Use one of country OR ISO OR continent, not combined)
species	string specifying the Plasmodium species for which to find PR points, options include: "Pf" OR "Pv" OR "BOTH"
extent	an object specifying spatial extent within which PR data is desired, as returned by sf::st_bbox() - the first column has the minimum, the second the maximum values; rows 1 & 2 represent the x & y dimensions respectively (matrix(c("xmin", "ymin","xmax", "ymax"), nrow = 2, ncol = 2, dimnames = list(c("x", "y"), c("min", "max"))))
start_date	string object representing the lower date to filter the PR data by (inclusive) e.g. '2020-01-01'
end_date	string object representing the upper date to filter the PR data by (exclusive) e.g. '2020-01-01'
version	(optional) The PR dataset version to return. If not provided, will just use the most recent version of PR data. (To see available version options, use listPR-PointVersions)

Details

country and ISO refer to countries and a lower-level administrative regions such as Mayotte and French Guiana. While we cannot direectly distribute DHS coordinates, we can distribute the number of examined and positive. If the coordinates are needed they can be downloaded from www.measuredhs.com, via the rdhs package or using malariaAtlas:fillDHSCoordinates().

Value

getPR returns a dataframe containing the below columns, in which each row represents a distinct data point/ study site.

1. dhs_id The dhs survey id if appropriate.
2. site_id Unique site identifier
3. site_name Name of site.

See Also

autoplot method for quick mapping of PR point locations (autoplot.pr.points).

Examples

```
#Download PfPR data for Nigeria and Cameroon and map the locations of these points using autoplot
## Not run:
NGA_CMR_PR <- getPR(country = c("Nigeria", "Cameroon"), species = "Pf")
autoplot(NGA_CMR_PR)

#Download PfPR data for Madagascar and map the locations of these points using autoplot
Madagascar_pr <- getPR(ISO = "MDG", species = "Pv")
autoplot(Madagascar_pr)

getPR(country = "ALL", species = "BOTH")

## End(Not run)
```

---

getRaster *Download Rasters produced by the Malaria Atlas Project*

Description

getRaster downloads publicly available MAP rasters for a specific surface & year, clipped to a provided bounding box or shapefile.

Usage

```
getRaster(
  dataset_id = NULL,
  surface = NULL,
  shp = NULL,
  extent = NULL,
  file_path = NULL,
  year = NULL,
  vector_year = NULL
)
```

Arguments

dataset_id	A character string specifying the dataset ID(s) of one or more rasters. These dataset ids can be found in the data.frame returned by listRaster, in the dataset_id column e.g. c('Malaria__202206_Global_Pf_Mortality_Count', 'Malaria__202206_Global_Pf_Parasite_
surface	deprecated argument. Please remove it from your code.
shp	SpatialPolygon(s) object of a shapefile to use when clipping downloaded rasters. (use either shp OR extent; if neither is specified global raster is returned).
extent	2x2 matrix specifying the spatial extent within which raster data is desired, as returned by sf::st_bbox() - the first column has the minimum, the second the maximum values; rows 1 & 2 represent the x & y dimensions respectively (matrix(c("xmin", "ymin","xmax", "ymax"), nrow = 2, ncol = 2, dimnames = list(c("x", "y"), c("min", "max")))) (use either shp OR extent; if neither is specified global raster is returned).
file_path	string specifying the directory to which raster files will be downloaded, if you want to download them. If none given, rasters will not be saved to files.
year	default = rep(NA, length(dataset_id)) (use NA for static rasters); for time-varying rasters: if downloading a single surface for one or more years, year should be a string specifying the desired year(s). if downloading more than one surface, use a list the same length as dataset_id, providing the desired year-range for each time-varying surface in dataset_id or NA for static rasters.
vector_year	deprecated argument. Please remove it from your code.

Value

getRaster returns a SpatRaster for the specified extent. Or a SpatRasterCollection if the two rasters are incompatible in terms of projection/extent/resolution

See Also

autoplot_MAPraster

to quickly visualise rasters downloded using getRaster.

as.MAPraster

to convert RasterLayer/RasterStack objects into a 'MAPraster' object (data.frame) for easy plotting with ggplot.

autoplot.MAPraster

to quickly visualise MAPraster objects created using as.MAPraster.

Examples

```
# Download PfPR2-10 Raster for Madagascar and visualise this immediately.
## Not run:
MDG_shp <- getShp(ISO = "MDG", admin_level = "admin0")
MDG_PfPR2_10 <- getRaster(dataset_id = "Malaria__202206_Global_Pf_Parasite_Rate", shp = MDG_shp)
autoplot(MDG_PfPR2_10)

## End(Not run)
```

---

getShp *Download MAPadmin2013 Administrative Boundary Shapefiles from the MAP geoserver*

Description

getShp downloads a shapefile for a specified country (or countries) and returns this as either a spatialPolygon or data.frame object.

Usage

```
getShp(
  country = NULL,
  ISO = NULL,
  extent = NULL,
  admin_level = c("admin0"),
  format = NULL,
  long = NULL,
  lat = NULL,
  version = NULL
)
```

Arguments

country	string containing name of desired country, e.g. c("Country1", "Country2", ...) OR = "ALL" (use either ISO OR country)
ISO	string containing ISO3 code for desired country, e.g. c("XXX", "YYY", ...) OR = "ALL" (use either ISO OR country)
extent	2x2 matrix specifying the spatial extent within which polygons are desired, as returned by sp::bbox() - the first column has the minimum, the second the maximum values; rows 1 & 2 represent the x & y dimensions respectively (matrix(c("xmin", "ymin","xmax", "ymax"), nrow = 2, ncol = 2, dimnames = list(c("x", "y"), c("min", "max")))).

admin_level	string specifying the administrative level for which shapefile are desired (only "admin0","admin1","admin2","admin3", or "all" accepted). N.B. Not all administrative levels are available for all countries. Use listShp to check which shapefiles are available. If an administrative level is requested that is not available, the closest available administrative level shapefiles will be returned.
format	deprecated argument. Please remove it from your code.
long	longitude of a point location falling within the desired shapefile.
lat	latitude of a point location falling within the desired shapefile.
version	The admin unit dataset version to return. Is NULL by default, and if left NULL will just use the most recent version of admin unit data.

Value

getShp returns a sf object for requested administrative unit polygons. The following attribute fields are included:

1. iso ISO-3 code of given administrative unit (or the ISO code of parent unit for administrative-level 1 units).
2. admn_level administrative level of the given administrative unit - either 0 (national), 1 (first-level division), 2 (second-level division), 3 (third-level division).
3. name_0 name of admin0 parent of a given administrative unit (or just shapefile name for admin0 units)
4. id_0 id code of admin0 parent of the current shapefile (or just shapefile id for admin0 units)
5. type_0 if applicable, type of administrative unit or admin0 parent
6. name_1 name of admin1 parent of a given administrative unit (or just shapefile name for admin1 units); NA for admin0 units
7. id_1 id code of admin1 parent of the current shapefile (or just shapefile id for admin1 units); NA for admin0 units
8. type_1 if applicable, type of administrative unit or admin1 parent
9. name_2 name of admin2 parent of a given administrative unit (or just shapefile name for admin2 units); NA for admin0, admin1 units
10. id_2 id code of admin2 parent of the current shapefile (or just shapefile id for admin2 units); NA for admin0, admin1 units
11. type_2 if applicable, type of administrative unit or admin2 parent
12. name_3 name of admin3 parent of a given administrative unit (or just shapefile name for admin3 units); NA for admin0, admin1, admin2 units
13. id_3 id code of admin3 parent of the current shapefile (or just shapefile id for admin3 units); NA for admin0, admin1, admin2 units
14. type_3 if applicable, type of administrative unit
15. source source of administrative boundaries
16. geometry geometry of administrative boundaries
17. country_level composite iso_admn_level field.

See Also

autoplot method for quick mapping of PR point locations (autoplot.pr.points).

Examples

```
#Download PfPR data & associated shapefiles for Nigeria and Cameroon
## Not run:
NGA_CMR_PR <- getPR(country = c("Nigeria", "Cameroon"), species = "Pf")
NGA_CMR_shp <- getShp(country = c("Nigeria", "Cameroon"))

#Download PfPR data & associated shapefiles for Chad
Chad_PR <- getPR(ISO = "TCD", species = "both")
Chad_shp <- getShp(ISO = "TCD")

#' #Download PfPR data & associated shapefiles defined by extent for Madagascar
MDG_PR <- getPR(country = "Madagascar", species = "Pv")

## End(Not run)
```

---

getSpBbox *Return sp style bbox*

Description

Return sp style bbox

Usage

```
getSpBbox(sfBboxOrShp)
```

Arguments

sfBboxOrShp	sf shapefile or result of sf::st_bbox(sf_shp)

Value

bbox in sp style. A 2x2 matrix - the first column has the minimum, the second the maximum values; rows 1 & 2 represent the x & y dimensions respectively (matrix(c("xmin", "ymin","xmax", "ymax"), nrow = 2, ncol = 2, dimnames = list(c("x", "y"), c("min", "max"))))

getVecOcc *Download Vector Occurrence points from the MAP database* getVecOcc *downloads all publicly available vector occurrence points for a specified country (or countries) and returns this as a dataframe.* country *and* ISO *refer to countries and a lower-level administrative regions such as French Guiana.*

Description

Download Vector Occurrence points from the MAP database getVecOcc downloads all publicly available vector occurrence points for a specified country (or countries) and returns this as a dataframe. country and ISO refer to countries and a lower-level administrative regions such as French Guiana.

Usage

```
getVecOcc(
  country = NULL,
  ISO = NULL,
  continent = NULL,
  species = "all",
  extent = NULL,
  start_date = NULL,
  end_date = NULL,
  version = NULL
)
```

Arguments

country	string containing name of desired country, e.g. c("Country1", "Country2", ...) OR = "ALL". (Use one of country OR ISO OR continent, not combined)
ISO	string containing ISO3 code for desired country, e.g. c("XXX", "YYY", ...) OR = "ALL". (Use one of country OR ISO OR continent, not combined)
continent	string containing continent (one of "Africa", "Americas", "Asia", "Oceania") for desired data, e.g. c("continent1", "continent2", ...). (Use one of country OR ISO OR continent, not combined)
species	string specifying the Anopheles species for which to find vector occurrence points, options include: "Anopheles...." OR "ALL"
extent	an object specifying spatial extent within which PR data is desired, as returned by sf::st_bbox(). - the first column has the minimum, the second the maximum values; rows 1 & 2 represent the x & y dimensions respectively (matrix(c("xmin", "ymin","xmax", "ymax"), nrow = 2, ncol = 2, dimnames = list(c("x", "y"), c("min", "max"))))
start_date	string object representing the lower date to filter the vector occurrence data by (inclusive)
end_date	string object representing the upper date to filter the vector occurrence data by (exclusive)

version (optional) The vector points dataset version to use. If not provided, will just use the most recent version of vector points data. (To see available version options, use listVecOccPointVersions)

Value

getVecOcc returns a dataframe containing the below columns, in which each row represents a distinct data point/ study site.

1. COLUMNNAME description of contents
2. COLUMNNAME description of contents
3. COLUMNNAME description of contents

See Also

autoplot method for quick mapping of Vector occurrence point locations (autoplot.vector.points).

Examples

```
# Download vector occurrence data for Brazil and map the locations using autoplot.vector.points
## Not run:
Brazil_vec <- getVecOcc(country = "Brazil")
autoplot(Brazil_vec)

# Download vector data for Madagascar and map the locations using autoplot
Madagascar_vec <- getVecOcc(ISO = "MDG", species = "All")
autoplot(Madagascar_vec)

# Subset by extent.
extent_vec <- getVecOcc(extent = matrix(c(100,13,110,18), nrow = 2), species = 'all')

## End(Not run)
```

---

isAvailable *Available data to download from the MAP geoserver.*

Description

isAvaiable is a wrapper for isAvailable_pr and isAvailable_vec, listing data (PR survey point location data and vector occurrence locations available to download from the MAP geoserver.

Usage

```
isAvailable(
  sourcedata = NULL,
  full_results = FALSE,
  country = NULL,
  ISO = NULL,
  continent = NULL,
  ...
)
```

Arguments

sourcedata	One of 'pr points' or 'vector points'
full_results	Should the list be printed to the console?
country	string containing name of desired country, e.g. c("Country1", "Country2", ...) OR = "ALL" (use one of country OR ISO OR continent, not combined)
ISO	string containing ISO3 code for desired country, e.g. c("XXX", "YYY", ...) OR = "ALL" (use one of country OR ISO OR continent, not combined)
continent	string containing continent for desired data, e.g. c("continent1", "continent2", ...) (use one of country OR ISO OR continent, not combined)
...	passed on to isAvailable_vec and isAvailable_pr

Value

isAvailable returns a data.frame detailing the administrative units for which shapefiles are stored on the MAP geoserver.

See Also

link{isAvailable_pr} isAvailable_vec

Examples

```
## Not run:
available_pr_locations <- isAvailable_pr(ISO = 'IDN')
available_vector_locations <- isAvailable_vec(ISO = 'IDN')

## End(Not run)
```

isAvailable_pr *Check whether PR points are available for a given location*

Description

isAvailable_pr checks whether the MAP database contains PR points for the specified country/location.

Usage

```
isAvailable_pr(
  sourcedata = NULL,
  country = NULL,
  ISO = NULL,
  continent = NULL,
  full_results = FALSE,
  version = NULL
)
```

Arguments

sourcedata	deprecated argument. Please remove it from your code.
country	string containing name of desired country, e.g. c("Country1", "Country2", ...) (use one of country OR ISO OR continent, not combined)
ISO	string containing ISO3 code for desired country, e.g. c("XXX", "YYY", ...) (use one of country OR ISO OR continent, not combined)
continent	string containing name of continent for desired data, e.g. c("Continent1", "Continent2", ...)(use one of country OR ISO OR continent, not combined)
full_results	By default this is FALSE meaning the function only gives a message outlining whether specified country is available, if full_results == TRUE, the function returns a named list outlining data availability.
version	(optional) The PR dataset version to use. If not provided, will just use the most recent version of PR data. (To see available version options, use listPRPointVersions)

Value

isAvailable_pr returns a named list of input locations with information regarding data availability.

if full_results == TRUE, a named list is returned with the following elements:

1. location - specified input locations
2. is_available- 1 or 0; indicating whether data is available for this location
3. possible_match- agrep-matched country names indicating potential mispellings of countries where is_available == 0; NA if data is available for this location.

Examples

```
## Not run:
isAvailable_pr(country = "Suriname")
x <- isAvailable_pr(ISO = "NGA", full_results = TRUE)
x <- isAvailable_pr(continent = "Oceania", full_results = TRUE)

## End(Not run)
```

`isAvailable_vec` *Check whether Vector Occurrence points are available for a given location*

Description

`isAvailable_vec` checks whether the MAP database contains Vector Occurrence points for the specified country/location.

Usage

```
isAvailable_vec(
  sourcedata = NULL,
  country = NULL,
  ISO = NULL,
  continent = NULL,
  full_results = FALSE,
  version = NULL
)
```

Arguments

`sourcedata`	deprecated argument. Please remove it from your code.
`country`	string containing name of desired country, e.g. `c("Country1", "Country2", ...)` (use one of `country` OR `ISO` OR `continent`, not combined)
`ISO`	string containing ISO3 code for desired country, e.g. `c("XXX", "YYY", ...)` (use one of `country` OR `ISO` OR `continent`, not combined)
`continent`	string containing name of continent for desired data, e.g. `c("Continent1", "Continent2", ...)`(use one of `country` OR `ISO` OR `continent`, not combined)
`full_results`	By default this is FALSE meaning the function only gives a message outlining whether specified country is available, if `full_results == TRUE`, the function returns a named list outlining data availability.
`version`	(optional) The vector points dataset version to use. If not provided, will just use the most recent version of vector points data. (To see available version options, use listVecOccPointVersions)

Value

isAvailable_Vec returns a named list of input locations with information regarding data availability.

if full_results == TRUE, a named list is returned with the following elements:

1. location - specified input locations
2. is_available- 1 or 0; indicating whether data is available for this location
3. possible_match- agrep-matched country names indicating potential mispellings of countries where is_available == 0; NA if data is available for this location.

Examples

```
## Not run:
isAvailable_vec(country = "Suriname")
x <- isAvailable_vec(ISO = "NGA", full_results = TRUE)
x <- isAvailable_vec(continent = "Oceania", full_results = TRUE)

## End(Not run)
```

---

isMaskedRaster *Returns true if second band of raster is a mask*

Description

Returns true if second band of raster is a mask

Usage

```
isMaskedRaster(raster)
```

Arguments

raster	SpatRaster object containing a single layer

---

listData *Deprecated function. Please instead use listPRPointCountries for pr points, listVecOccPointCountries for vector points, listRaster for raster and listShp for shape.*

Description

listData deprecated function Please remove it from your code.

Usage

```
listData(datatype, printed = TRUE, ...)
```

Arguments

datatype	"pr points", "vector points" "raster", or "shape"
printed	whether to pretty print the output in console
...	passed on to listPRPointCountries, listVecOccPointCountries, listShp

---

listPoints *Deprecated function. Please instead use listPRPointCountries for pr points, and listVecOccPointCountries for vector points*

Description

listPoints deprecated function Please remove it from your code.

Usage

```
listPoints(printed = TRUE, sourcedata, version = NULL)
```

Arguments

printed	whether to pretty print the output in console
sourcedata	"pr points" or "vector points"
version	(optional) The PR dataset version to use If not provided, will just use the most recent version of PR data. (To see available version options, use listPRPointVersions)

---

listPRPointCountries *List countries where there is pr point data available*

Description

listPRPointCountries

Usage

```
listPRPointCountries(printed = TRUE, version = NULL)
```

Arguments

printed	Should the list be printed to the console?
version	(optional) The PR dataset version to use If not provided, will just use the most recent version of PR data. (To see available version options, use listPRPointVersions)

Value

listPRPointCountries returns a data.frame detailing the countries for which PR points are publicly available.

listPRPointVersions — *List all dataset versions from the Web Feature Services provided by the Malaria Atlas Project within the Parasite Rate workspace.*

Description

`listPRPointVersions` lists available versions of parasite rate point data from the Web Feature Services provided by the Malaria Atlas Project.

Usage

```
listPRPointVersions(printed = TRUE)
```

Arguments

`printed`	Should the list be printed to the console?

Value

A data.frame with column 'version' The version can then be provided to other functions to fetch the data within that dataset. e.g. in getPR

Examples

```
## Not run:
prDatasets <- listPRPointVersions()

## End(Not run)
```

listRaster — *List all MAP Rasters available to download.*

Description

`listRaster` lists all rasters available to download from the Malaria Atlas Project database.

Usage

```
listRaster(printed = TRUE)
```

Arguments

`printed`	Should the list be printed to the console?

Value

listRaster returns a data.frame detailing the following information for each raster available to download from the Malaria Atlas Project database.

1. dataset_id the unique dataset ID of the raster, which can the be used in functions such as getRaster and extractRaster
2. raster_code unique identifier for each raster
3. title abbreviated title for each raster, used as surface argument in getRaster()
4. title_extended extended title for each raster, detailing raster content
5. abstract full description of each raster, outlining raster creation methods, raster content and more.
6. citation citation of peer-reviewed article in which each raster has been published
7. pub_year year in which raster was published, used as pub_year argument in getRaster() to updated raster versions from their predecessor(s).
8. min_raster_year earliest year for which each raster is available
9. max_raster_year latest year for which each raster is available

Examples

```
## Not run:
available_rasters <- listRaster()

## End(Not run)
```

---

listShp *List administrative units for which shapefiles are stored on the MAP geoserver.*

Description

listShp lists all administrative units for which shapefiles are stored on the MAP geoserver.

Usage

```
listShp(printed = TRUE, admin_level = c("admin0", "admin1"), version = NULL)
```

Arguments

printed	Should the list be printed to the console?
admin_level	Specifies which administrative unit level for which to return available polygon shapefiles. A string vector including one or more of"admin0", "admin1", "admin2" OR "admin3". Default: c("admin0", "admin1")
version	The admin unit dataset version to return. Is NULL by default, and if left NULL will just use the most recent version of admin unit data.

Value

listShp returns a data.frame detailing the administrative units for which shapefiles are stored on the MAP geoserver.

Examples

```
## Not run:
available_admin_units <- listShp()
available_admin_units <- listShp(admin_level = c('admin2','admin3'), version = '202206')

## End(Not run)
```

---

listShpVersions *List all versions of admin unit shapes from the Web Feature Services provided by the Malaria Atlas Project within the Admin Units workspace.*

Description

listShpVersions lists available versions of Admin Unit shapefiles from the Web Feature Services provided by the Malaria Atlas Project.

Usage

```
listShpVersions(printed = TRUE)
```

Arguments

printed	Should the list be printed to the console?

Value

A data.frame with column 'version'. The version can then be provided to other functions to fetch the data within that dataset. e.g. in getShp

Examples

```
## Not run:
vecOccDatasets <- listShpVersions()

## End(Not run)
```

listSpecies *list all species which have occurrence data within the MAP database.*

Description

`listSpecies` lists all species occurrence data available to download from the Malaria Atlas Project database.

Usage

```
listSpecies(printed = TRUE, version = NULL)
```

Arguments

`printed`	should the list be printed to the database.
`version`	(optional) The vector dataset version to use If not provided, will just use the most recent version of vector dataset data. (To see available version options, use listVecOccPointVersions)

Value

`listSpecies` returns a data.frame detailing the following information for each species available to download from the Malaria Atlas Project database.

1. `species` string detailing species

Examples

```
## Not run:
available_species <- listSpecies()

## End(Not run)
```

listVecOccPointCountries *List countries where there is vector occurrence point data available*

Description

`listVecOccPointCountries`

Usage

```
listVecOccPointCountries(printed = TRUE, version = NULL)
```

Arguments

printed	Should the list be printed to the console?
version	(optional) The vector occurrence dataset version to use If not provided, will just use the most recent version of vector occurrence data. (To see available version options, use listVecOccPointVersions)

Value

listVecOccPointCountries returns a data.frame detailing the countries for which vector occurrence points are publicly available.

---

listVecOccPointVersions

List all dataset versions from the Web Feature Services provided by the Malaria Atlas Project within the Vector Occurrence workspace.

Description

listVecOccPointVersions lists available versions of all the feature datasets in the Vector Occurrence workspace from the Web Feature Services provided by the Malaria Atlas Project.

Usage

```
listVecOccPointVersions(printed = TRUE)
```

Arguments

printed	Should the list be printed to the console?

Value

A data.frame with column 'version'. The version can then be provided to other functions to fetch the data within that dataset. e.g. in getVecOcc

Examples

```
## Not run:
vecOccDatasets <- listVecOccPointVersions()

## End(Not run)
```

`makeSpatRasterAutoplot` *Create a single (sub) plot for a SpatRaster*

Description

Create a single (sub) plot for a SpatRaster

Usage

```
makeSpatRasterAutoplot(
  spatraster,
  rastername,
  shp_df,
  legend_title,
  fill_scale_transform,
  fill_colour_palette,
  plot_titles
)
```

Arguments

`spatraster`	SpatRaster object containing a single layer
`rastername`	raster name, to include in title
`shp_df`	sf shapefile
`legend_title`	title for legend
`fill_scale_transform`	scale
`fill_colour_palette`	palette
`plot_titles`	bool, whether to include title

Value

ggplot object

malariaAtlas *An R interface to open-access malaria data, hosted by the Malaria Atlas Project.*

Description

malariaAtlas provides a suite of tools to allow you to download all publicly available PR points for a specified country (or ALL countries) as a dataframe within R.

malariaAtlas functions

1. listAll - lists all countries for which there are publicly visible PR datapoints in the MAP database.
2. is_available - checks whether the MAP database contains PR points for the specified country/countries.
3. getPR - downloads all publicly available PR points for a specified country (or countries) and returns this as a dataframe.

Index