Package ‘rphylopic’

October 14, 2022

Title Get 'Silhouettes' of 'Organisms' from 'Phylopic'

Description Work with 'Phylopic' web service (<`http://phylopic.org/api/`>) to get 'silhouette' images of 'organisms', search names, and more. Includes functions for adding 'silhouettes' to both base plots and ggplot2 plots.

Version 0.3.0

License MIT + file LICENSE

URL `https://github.com/sckott/rphylopic`

BugReports `https://github.com/sckott/rphylopic/issues`

Encoding UTF-8

Language en-US

VignetteBuilder knitr, R.rsp

Imports ggplot2, crul (>= 0.5.2), jsonlite, grid, gridBase, graphics, png

Suggests testthat, vcr (>= 0.2.0), knitr, rmarkdown, R.rsp

RoxygenNote 7.1.0

NeedsCompilation no

Author Scott Chamberlain [aut, cre] (<`https://orcid.org/0000-0003-1444-9135`>), David Miller [ctb]

Maintainer Scott Chamberlain <`myrmecocystus@gmail.com`>

Repository CRAN

Date/Publication 2020-06-04 10:00:10 UTC

R topics documented:

`add_phylopic` *Input an image and create a ggplot2 layer to add to an existing plot*

Description

Input an image and create a ggplot2 layer to add to an existing plot

Usage

```
add_phylopic(img, alpha = 0.2, x = NULL, y = NULL, ysize = NULL, color = NULL)
```

Arguments

`img`	A png object, e.g, from using `image_data()`
`alpha`	A value between 0 and 1, specifying the opacity of the silhouette.
`x`	x value of the silhouette center. Ignored if y and ysize are not specified.
`y`	y value of the silhouette center. Ignored if x and ysize are not specified.
`ysize`	Height of the silhouette. The width is determined by the aspect ratio of the original image. Ignored if x and y are not specified.
`color`	Color to plot the silhouette in.

Details

Use parameters `x`, `y`, and `ysize` to place the silhouette at a specified position on the plot. If all three of these parameters are unspecified, then the silhouette will be plotted to the full height and width of the plot.

Examples

```
## Not run:
# Put a silhouette behind a plot
library(ggplot2)
img <- image_data("27356f15-3cf8-47e8-ab41-71c6260b2724", size = "512")[[1]]
qplot(x=Sepal.Length, y=Sepal.Width, data=iris, geom="point") +
  add_phylopic(img)

# Put a silhouette anywhere
```

```
library(ggplot2)
posx <- runif(50, 0, 10)
posy <- runif(50, 0, 10)
sizey <- runif(50, 0.4, 2)
cols <- sample(c("black", "darkorange", "grey42", "white"), 50,
  replace = TRUE)

cat <- image_data("23cd6aa4-9587-4a2e-8e26-de42885004c9", size = 128)[[1]]
(p <- ggplot(data.frame(cat.x = posx, cat.y = posy), aes(cat.x, cat.y)) +
 geom_point(color = rgb(0,0,0,0)))
for (i in 1:50) {
  p <- p + add_phylopic(cat, 1, posx[i], posy[i], sizey[i], cols[i])
}
p + ggtitle("R Cat Herd!!")

## End(Not run)
```

`add_phylopic_base` *Input an image and add to an existing plot made with base graphics*

Description

Input an image and add to an existing plot made with base graphics

Usage

```
add_phylopic_base(
  img,
  x = NULL,
  y = NULL,
  ysize = NULL,
  alpha = 0.2,
  color = NULL
)
```

Arguments

`img`	A png object, e.g, from using `image_data()`
`x`	x value of the silhouette center. Ignored if y and ysize are not specified.
`y`	y value of the silhouette center. Ignored if x and ysize are not specified.
`ysize`	Height of the silhouette. The width is determined by the aspect ratio of the original image. Ignored if x and y are not specified.
`alpha`	A value between 0 and 1, specifying the opacity of the silhouette.
`color`	Color to plot the silhouette in.

Details

Use parameters `x`, `y`, and `ysize` to place the silhouette at a specified position on the plot. If all three of these parameters are unspecified, then the silhouette will be plotted to the full height and width of the plot.

Examples

```
## Not run:
# get a silhouette
cat <- image_data("23cd6aa4-9587-4a2e-8e26-de42885004c9", size = 128)[[1]]

# single image
plot(1, 1, type="n", main="A cat")
add_phylopic_base(cat, 1, 1, 0.2)

# lots of images
posx <- runif(50, 0, 1)
posy <- runif(50, 0, 1)
size <- runif(50, 0.01, 0.2)
plot(posx, posy, type="n", main="A cat herd")
for (i in 1:50) {
  add_phylopic_base(cat, posx[i], posy[i], size[i])
}

# Example using a cat background
# setup plot area
plot(posx, posy, type="n", main="A cat herd, on top of a cat",
     xlim=c(0,1), ylim=c(0,1))
# get a higher-resolution cat
cat_hires <- image_data("23cd6aa4-9587-4a2e-8e26-de42885004c9", size = 512)[[1]]
# plot background cat
add_phylopic_base(cat_hires, 0.5, 0.5, 1, alpha=0.2)
# overlay smaller cats
for (i in 1:50) {
  add_phylopic_base(cat, posx[i], posy[i], size[i], alpha=.8)
}

## End(Not run)
```

`get_names` *Get names for uuids.*

Description

Get names for uuids.

Usage

```
get_names(
  uuid,
  supertaxa = NULL,
  subtaxa = NULL,
  options = NULL,
  stripauthority = TRUE,
  ...
)
```

Arguments

`uuid`	UUID to get names for
`supertaxa`	If immediate, returns data for immediate supertaxa ("parents"). If all, returns data for all supertaxa ("ancestors"). Otherwise, does not include supertaxa.
`subtaxa`	If immediate, returns data for immediate subtaxa ("children"). Otherwise, does not include subtaxa.
`options`	See details for the options for options, get it, ha.
`stripauthority`	If TRUE (default) the authority is stripped off of the scientific name.
`...`	curl options passed on to crul::HttpClient

Details

Here are the options for the options argument:

- citationStart: (optional) Integer Indicates where in the string the citation starts. May be null.
- html: (optional) StringHTML version of the name.
- namebankID: (optional) StringuBio Namebank identifier. May be null.
- root: (optional) Boolean If true, this name has no hyperonyms (names of supertaxa). (Should only be true for Panbiota/Vitae.)
- string: (optional) String The text of the name, including the citation, if any.
- type: (optional) String Either "scientific or "vernacular.
- uid: (always) String Universally unique identifier.
- uri: (optional) String The unique URI associated with the name.
- votes: (optional) Integer The number of votes this name has received. (Currently unused.)

Examples

```
## Not run:
get_names(uuid = "f3254fbd-284f-46c1-ae0f-685549a6a373", options = "string")
get_names(uuid = "f3254fbd-284f-46c1-ae0f-685549a6a373", supertaxa="immediate",
   options=c("string namebankID"))
get_names(uuid = "f3254fbd-284f-46c1-ae0f-685549a6a373", supertaxa="all",
   options="string")

## End(Not run)
```

`image` *Perform actions with images.*

Description

Perform actions with images.

Usage

```
image_get(uuid, options = NULL, ...)

image_list(start = 1, length = 10, options = NULL, ...)

image_timerange(
  timestamp = "modified",
  from = NULL,
  to = NULL,
  options = NULL,
  ...
)

image_count(...)

image_data(input, size, ...)
```

Arguments

`uuid`	One or more name UUIDs.
`options`	(character) One or more of citationStart, html, namebankID, root, string, type, uid, uri, and/or votes
`...`	curl options passed on to crul::HttpClient
`start`	The index to start with. Using 0 starts with the most recently-submitted image.
`length`	Number of images to list.
`timestamp`	Either `modified` (to go by the last time the image file was modified) or `submitted` (to go by the time the image was first submitted).
`from`	timestamp string, in "YYYY-MM-DD-HH-MM-SS" format, telling the earliest time to retrieve images for. All numbers past the year are optional. For example, these are acceptable: "2011-10-29-20-30", "2011-10-29-20", "2011-10-29", "2011-10", and "2011". Omitted numbers indicate the lowest possible value for that number, for example, "2011" indicates "2011-01-01-00-00-00" (2011 January 1, midnight). Numbers in the string do not need to be padded. For example, this is acceptable: "2011-1-1-0-0-0". The image list will include any images dated at the indicated time.

`to`	A date-time string, in "YYYY-MM-DD-HH-MM-SS" format, telling the earliest time to retrieve images for. See the from parameter for more details on the format. The image list will include any images dated up to, but not including, the indicated time.
`input`	Either a vector of uuids or the output from the function `search_images`
`size`	Height of the image, one of 64, 128, 256, 512, 1024, "thumb", or "icon"

Details

I'm not adding methods for modifying images, including add, edit, updated, delete, and transfer, because I can't imagine doing those things from R. Am I wrong?

Note: uid is always returned

Examples

```
## Not run:
# Get info on an image
uuid <- "9fae30cd-fb59-4a81-a39c-e1826a35f612"
image_get(uuid = uuid)
image_get(uuid = uuid, options=c('credit','pngFiles','taxa','canonicalName','string','uri','type'))
image_get(uuid = uuid, options=c('credit','licenseURL','pngFiles','submitted','submitter',
   'svgFile','taxa','canonicalName','string','uri','type','citationStart'))

# Count images in Phylopic database
image_count()
image_count(verbose = TRUE)

# Lists images in chronological order, from most to least recently modified
image_list(start=1, length=10)
image_list(start=1, length=10, options="taxa")

# Lists images within a given time range, from most to least recent
image_timerange(from="2013-05-11")
image_timerange(from="2013-05-11", to="2013-05-12")
image_timerange(from="2013-05-11", to="2013-05-12", options='credit')

# Get data for an image
## input uuids
toget <- c("c089caae-43ef-4e4e-bf26-973dd4cb65c5", "41b127f6-0824-4594-a941-5ff571f32378",
   "9c6af553-390c-4bdd-baeb-6992cbc540b1")
image_data(toget, size = "64")
image_data(toget, size = "thumb")

## input the output from search_images
x <- search_text(text = "Homo sapiens", options = "names")
output <- search_images(x[1:10], options=c("pngFiles", "credit", "canonicalName"))
image_data(output, size = "64")

## Put a silhouette behind a plot
library('ggplot2')
img <- image_data("27356f15-3cf8-47e8-ab41-71c6260b2724", size = "512")[[1]]
```

```
qplot(x=Sepal.Length, y=Sepal.Width, data=iris, geom="point") + add_phylopic(img)

## Use as points in a ggplot plot
library('ggplot2')
uuid <- "c089caae-43ef-4e4e-bf26-973dd4cb65c5"
img <- image_data(uuid, size = "64")[[1]]
(p <- ggplot(mtcars, aes(drat, wt)) + geom_blank())
for(i in 1:nrow(mtcars)) p <- p + add_phylopic(img, 1, mtcars$drat[i], mtcars$wt[i], ysize = 0.3)
p

## End(Not run)
```

`name` *Perform actions with names.*

Description

Perform actions with names.

Usage

```
name_get(uuid, options = NULL, ...)

name_images(
  uuid,
  subtaxa = NULL,
  supertaxa = NULL,
  other = FALSE,
  options = NULL,
  ...
)

name_minsuptaxa(uuid, options = NULL, ...)

name_search(text, options = NULL, as = "table", ...)

name_taxonomy(
  uuid,
  subtaxa = NULL,
  supertaxa = NULL,
  useUBio = FALSE,
  options = NULL,
  as = "table",
  ...
)

name_taxonomy_many(uuid, options = NULL, as = "table", ...)

name_taxonomy_sources(uuid, options = NULL, as = "list", ...)
```

Arguments

`uuid`	One or more name UUIDs.
`options`	(character) One or more of citationStart, html, namebankID, root, string, type, uid, uri, and/or votes
`...`	curl options passed on to crul::HttpClient
`subtaxa`	If immediate, returns data for immediate subtaxa ("children"). Otherwise, does not include subtaxa.
`supertaxa`	If immediate, returns data for immediate supertaxa ("parents"). If all, returns data for all supertaxa ("ancestors"). Otherwise, does not include supertaxa.
`other`	If set to `TRUE`, includes related taxa in the search.
`text`	(character) The text string to search on.
`as`	(character) What to return. One of table (default, a data.frame), list, or json.
`useUBio`	(logical) If TRUE, and there is pending data from uBio that needs to be cached, a list of commands will be passed back instead of the normal result.

Details

I'm not adding methods for modifying names, including add, edit, or toggle, because I can't imagine doing those things from R. Am I wrong?

Options for the `options` parameter:

- citationStart: (optional) Integer Indicates where in the string the citation starts. May be null.
- html: (optional) StringHTML version of the name.
- namebankID: (optional) StringuBio Namebank identifier. May be null.
- root: (optional) Boolean If true, this name has no hyperonyms (names of supertaxa). (Should only be true for Panbiota/Vitae.)
- string: (optional) String The text of the name, including the citation, if any.
- type: (optional) String Either "scientific or "vernacular.
- uid: (always) String Universally unique identifier.
- uri: (optional) String The unique URI associated with the name.
- votes: (optional) Integer The number of votes this name has received. (Currently unused.)

Examples

```
## Not run:
# parse as different outputs
name_taxonomy(uuid = "f3254fbd-284f-46c1-ae0f-685549a6a373",
   options = "string", as="table")
name_taxonomy(uuid = "f3254fbd-284f-46c1-ae0f-685549a6a373",
   options = "string", as="list")
name_taxonomy(uuid = "f3254fbd-284f-46c1-ae0f-685549a6a373",
   options = "string", as="json")

# Get info on a name
```

```
id <- "1ee65cf3-53db-4a52-9960-a9f7093d845d"
name_get(uuid = id)
name_get(uuid = id, options=c('citationStart','html'))
name_get(uuid = id, options=c('namebankID','root','votes'))

# Searches for images for a taxonomic name.
name_images(uuid = "1ee65cf3-53db-4a52-9960-a9f7093d845d")
name_images(uuid = "1ee65cf3-53db-4a52-9960-a9f7093d845d",
   options='credit')

# Finds the minimal common supertaxa for a list of names.
name_minsuptaxa(uuid=c("1ee65cf3-53db-4a52-9960-a9f7093d845d",
   "08141cfc-ef1f-4d0e-a061-b1347f5297a0"))

# Finds the taxa whose names match a piece of text.
name_search(text = "Homo sapiens")
name_search(text = "Homo sapiens", options = "names")
name_search(text = "Homo sapiens", options = "type")
name_search(text = "Homo sapiens", options = "namebankID")
name_search(text = "Homo sapiens", options = "root")
name_search(text = "Homo sapiens", options = "uri")
name_search(text = "Homo sapiens", options = c("string","type","uri"))

# Collects taxonomic data for a name.
name_taxonomy(uuid = "f3254fbd-284f-46c1-ae0f-685549a6a373",
   options = "string")
name_taxonomy(uuid = "f3254fbd-284f-46c1-ae0f-685549a6a373",
   supertaxa="immediate", options=c("string","namebankID"))
name_taxonomy(uuid = "f3254fbd-284f-46c1-ae0f-685549a6a373", supertaxa="all",
   options="string")
name_taxonomy(uuid = "f3254fbd-284f-46c1-ae0f-685549a6a373", supertaxa="all",
   options=c("string","uri"))

# Collects taxonomic data for multiple names.
name_taxonomy_many(uuid = c("f3254fbd-284f-46c1-ae0f-685549a6a373",
 "1ee65cf3-53db-4a52-9960-a9f7093d845d"))

# Collects data about the sources for a name's taxonomy.
name_taxonomy_sources(uuid = "f3254fbd-284f-46c1-ae0f-685549a6a373")
name_taxonomy_sources(uuid = "f3254fbd-284f-46c1-ae0f-685549a6a373",
  as="json")
name_taxonomy_sources(uuid = "1ee65cf3-53db-4a52-9960-a9f7093d845d")

## End(Not run)
```

`nameset` *Perform actions with name sets*

Description

Perform actions with name sets

Usage

```
nameset_get(uuid, options = NULL, ...)

nameset_taxonomy(uuid, options = NULL, ...)
```

Arguments

`uuid`	the UUID of a set of taxonomic names
`options`	(character) One or more of citationStart, html, namebankID, root, string, type, uid, uri, and/or votes
`...`	curl options passed on to crul::HttpClient

Details

`nameset_get()` retrieves information on a set of taxonomic names. `nameset_taxonomy()` collects taxonomic data for a set of taxonomic names.

Value

a named list

`options` parameter

Same as those for `name_*()` functions

Examples

```
## Not run:
# Retrieves information on a set of taxonomic names.
id <- "8d9a9ea3-95cc-414d-1000-4b683ce04be2"
nameset_get(uuid = id)
nameset_get(uuid = id, options=c('names','string'))

# Collects taxonomic data for a name.
nameset_taxonomy(uuid = "8d9a9ea3-95cc-414d-1000-4b683ce04be2",
  options = "string")
nameset_taxonomy(uuid = "8d9a9ea3-95cc-414d-1000-4b683ce04be2",
  supertaxa="immediate", options=c("string","namebankID"))

## End(Not run)
```

`recolor_phylopic` *Recolor a phylopic image*

Description

Internal function to recolour and change alpha levels of a phylopic image.

Usage

```
recolor_phylopic(img, alpha = 0.2, color = NULL)
```

Arguments

`img`	A png object, e.g, from using `image_data()`
`alpha`	A value between 0 and 1, specifying the opacity of the silhouette.
`color`	Color to plot the silhouette in.

`rphylopic` *rphylopic*

Description

Get Silhouettes of Organisms from Phylopic

Author(s)

Scott Chamberlain <`myrmecocystus@gmail.com`>

David Miller <`dave@ninepointeightone.net`>

`save_png` *Save an image to disk as a `.png` file*

Description

Save an image to disk as a `.png` file

Usage

```
save_png(img, target = tempfile(fileext = ".png"), ...)
```

Arguments

`img`	A png object, e.g, from using `image_data()`
`target`	the file or other connection to write to, passed on to png::writePNG, see its docs for details. By default we use `tempfile(fileext = ".png")`, a temporary `.png` file which is cleaned up (deleted) at the end of the R session
`...`	additional parameters passed on to png::writePNG

Value

path to the `.png` file on disk

Examples

```
## Not run:
# get a silhouette
cat <- image_data("23cd6aa4-9587-4a2e-8e26-de42885004c9", size = 128)[[1]]

# save image
out <- save_png(cat)
identical(png::readPNG(out), cat)

save_png(cat, dpi = 1000)

## End(Not run)
```

`search_images` *Search for images for a taxon (via its uuid)*

Description

Search for images for a taxon (via its uuid)

Usage

```
search_images(
  uuid,
  subtaxa = NULL,
  supertaxa = NULL,
  other = NULL,
  options = NULL,
  cleanoutput = TRUE,
  ...
)
```

Arguments

`uuid`	(character) one or more UUIDs of taxonomic names
`subtaxa`	If set to `TRUE`, includes subtaxa in the search.
`supertaxa`	If not set to `FALSE`, includes supertaxa in the search.
`other`	If set to `TRUE`, includes related taxa in the search.
`options`	Space-separated list of options for the result value.
`cleanoutput`	If `TRUE`, remove elements with no data.
`...`	curl options passed on to crul::HttpClient

Examples

```
## Not run:
search_images('1ee65cf3-53db-4a52-9960-a9f7093d845d',
  subtaxa = "true", options = c("pngFiles", "credit", "licenseURL",
  "svgFile", "canonicalName", "html"))

# all of them
search_images(c("1ee65cf3-53db-4a52-9960-a9f7093d845d",
   "08141cfc-ef1f-4d0e-a061-b1347f5297a0"),
   options=c("pngFiles", "credit", "canonicalName"))

## End(Not run)
```

`search_text` *Text search for uuids*

Description

Text search for uuids

Usage

```
search_text(text, options = "string", simplify = TRUE, ...)
```

Arguments

`text`	(character) Search string, see examples
`options`	(character) See here for options
`simplify`	(logical) Simplify result
`...`	curl options passed on to crul::HttpClient

Details

These aren’t necessarily ones with images though. See examples

Value

A list. You always get back the UUID, and any other fields requested.

Examples

```
## Not run:
search_text(text = "Homo sapiens")
search_text(text = "Homo sapiens", options = "names")
search_text(text = "Homo sapiens", options = "type")
search_text(text = "Homo sapiens", options = "namebankID")
search_text(text = "Homo sapiens", options = "root")
search_text(text = "Homo sapiens", options = "uri")
search_text(text = "Homo sapiens", options = c("string","type","uri"))
search_text(text = "Homo sapiens", options = c("string","type","uri"),
  simplify=FALSE)

# pass in curl options
search_text(text = "Homo sapiens", options = "names", verbose = TRUE)

## End(Not run)
```

`theme_phylo_blank2` *Fixed phylogeny blank theme for ggphylo*

Description

Fixed phylogeny blank theme for ggphylo

Usage

```
theme_phylo_blank2()
```

`ubio` *Perform actions with uBio data.*

Description

Perform actions with uBio data.

Usage

```
ubio_get(namebankID, options = NULL, ...)
```

Arguments

`namebankID`	The identifier for a name in uBio Namebank.
`options`	(character) One or more of citationStart, html, namebankID, root, string, type, uid, uri, and/or votes
`...`	curl options passed on to crul::HttpClient

Details

There is only one function for working with uBio data right now, `ubio_get()`

Options for the `options` parameter: Same as those for `name_*()` functions.

Examples

```
## Not run:
# Retrieves information on a set of taxonomic names.
ubio_get(109086)
ubio_get(109086, options=c('names','string'))

## End(Not run)
```

Index