Package ‘rfishdraw’

October 14, 2022

Type Package

Title Automatically Generated Fish Drawings via JavaScript

Version 0.1.0

Date 2021-09-06

Maintainer Liuyong Ding <ly_ding@126.com>

Description Automatic generation of fish drawings based on JavaScript library <https://github.com/LingDong-/fishdraw>, including JavaScript code for dynamic generation of fish drawings.

Depends R (>= 3.5.0)

Imports jsonlite, ggplot2

Suggests rmarkdown, knitr, prettydoc, patchwork, devtools

VignetteBuilder knitr

License Artistic-2.0

URL https://github.com/Otoliths/rfishdraw

BugReports https://github.com/Otoliths/rfishdraw/issues

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author Liuyong Ding [aut, cre] (<https://orcid.org/0000-0002-5490-182X>),
Songhao Ji [ctb]

Repository CRAN

Date/Publication 2021-09-08 09:30:01 UTC

R topics documented:

rfishdraw-package *Automatically Generated Fish Drawings via JavaScript*

Description

This package is designed for programmatically generated fish drawings via ggplot2 based on JavaScript library <https://github.com/LingDong-/fishdraw>.

Details

Package:	rfishdraw
Type:	Package
Title:	Automatically Generated Fish Drawings via JavaScript
Version:	0.1.0
Date:	2021-09-06
Authors@R:	c(person(given = "Liuyong", family = "Ding", email = "ly_ding@126.com", role = c("aut", "cre"), comme
Maintainer:	Liuyong Ding <ly_ding@126.com>
Description:	Automatic generation of fish drawings based on JavaScript library <https://github.com/LingDong-/fishdraw
Depends:	R (>= 3.5.0)
Imports:	jsonlite, ggplot2
Suggests:	rmarkdown, knitr, prettydoc, patchwork, devtools
VignetteBuilder:	knitr
License:	Artistic-2.0
URL:	https://github.com/Otoliths/rfishdraw
BugReports:	https://github.com/Otoliths/rfishdraw/issues
Encoding:	UTF-8
RoxygenNote:	7.1.1
Author:	Liuyong Ding [aut, cre] (<https://orcid.org/0000-0002-5490-182X>), Songhao Ji [ctb]

Index of help topics:

```
custom_params           Create and implement a custom JS library.
fish_draw               Automatically generated fish drawings via
                        ggplot2
get_polylines           Get outputs polylines (supported format svg,
                        json, csv, etc.)
rfishdraw-package       Automatically Generated Fish Drawings via
                        JavaScript
```

Author(s)

Maintainer: Liuyong Ding <ly_ding@126.com>

custom_params *Create and implement a custom JS library.*

Description

Programmatically generated fish drawings via ggplot2 based on JavaScript library https://github.com/LingDong-/fishdraw.

Usage

```
custom_params(
  path = "./fishdraw.js",
  save = getwd(),
  body_curve_type = 0,
  body_curve_amount = 0.85,
  body_length = 420,
  body_height = 90,
  scale_type = 1,
  scale_scale = 1,
  pattern_type = 3,
  pattern_scale = 1,
  dorsal_texture_type = 1,
  dorsal_type = 0,
  dorsal_length = 100,
  dorsal_start = 8,
  dorsal_end = 27,
  wing_texture_type = 0,
  wing_type = 0,
  wing_start = 6,
  wing_end = 6,
  wing_y = 0.7,
  wing_length = 130,
  wing_width = 10,
  pelvic_start = 9,
  pelvic_end = 14,
  pelvic_length = 85,
  pelvic_type = 0,
  pelvic_texture_type = 0,
  anal_start = 19,
  anal_end = 29,
  anal_length = 50,
  anal_type = 0,
  anal_texture_type = 0,
  tail_type = 0,
  tail_length = 75,
  finlet_type = 0,
  neck_type = 0,
  nose_height = 0,
  mouth_size = 8,
  head_length = 30,
  head_texture_amount = 60,
  has_moustache = 1,
  moustache_length = 10,
  has_beard = 0,
  has_teeth = 1,
  teeth_length = 8,
  teeth_space = 3.5,
  beard_length = 30,
  eye_type = 0,
  eye_size = 10,
  jaw_size = 1,
  jaw_open = 1
)
```

Arguments

`path` Path for JavaScript fishdraw.js

`save` Save path for custom JavaScript fishdraw.js(e.g.getwd())

`body_curve_type` `numeric` body_curve_type options:0 or 1.

`body_curve_amount` `numeric` body_curve_amount options:0.5,0.85 or 0.98.

`body_length` `numeric` body_length options:200,350 or 420.

`body_height` `numeric` body_height options:45,90 or 150.

`scale_type` `numeric` scale_type options:0,1,2 or 3.

`scale_scale` `numeric` scale_scale options:0.8,1 or 1.5.

`pattern_type` `numeric` pattern_type options:0,1,2,3 or 4.

`pattern_scale` `numeric` pattern_scale options:0.5,1 or 2.

`dorsal_texture_type` `numeric` dorsal_texture_type options:0 or 1.

`dorsal_type` `numeric` dorsal_type options:0 or 1.

`dorsal_length` `numeric` dorsal_length options:30,90 or 180.

`dorsal_start` `numeric` dorsal_start options:7,8,15;11,12,16.

`dorsal_end` `numeric` dorsal_end options:20,27,28;19,21,24.

`wing_texture_type` `numeric` wing_texture_type options:0 or 1.

`wing_type` `numeric` wing_type options:0 or 1.

`wing_start` `numeric` wing_start options:5,6 or 8.

`wing_end` `numeric` wing_end options:5,6 or 8.

`wing_y` `numeric` wing_y options:0.45,0.7 or 0.85.

`wing_length`	`numeric` wing_length options:40,130,200;40,150,350.
`wing_width`	`numeric` wing_width options:7,10,20;20,30,50.
`pelvic_start`	`numeric` pelvic_start options:7,9,11;7,9,12.
`pelvic_end`	`numeric` pelvic_end options:13,14,15
`pelvic_length`	`numeric` pelvic_length options:0 or 1.
`pelvic_type`	`numeric` pelvic_type options:0 or 1.
`pelvic_texture_type`	`numeric` pelvic_texture_type options:0 or 1.
`anal_start`	`numeric` anal_start options:16,19 or 23.
`anal_end`	`numeric` anal_end options:25,29 or 31.
`anal_length`	`numeric` anal_length options:20,50 or 80.
`anal_type`	`numeric` anal_type options:0 or 1.
`anal_texture_type`	`numeric` anal_texture_type options:0 or 1.
`tail_type`	`numeric` tail_type options:0,1,2,3,4 or 5.
`tail_length`	`numeric` tail_length options:50,75 or 180.
`finlet_type`	`numeric` finlet_type options:0,1,2 or 3.
`neck_type`	`numeric` neck_type options:0 or 1.
`nose_height`	`numeric` nose_height options:-50,0 or 35.
`mouth_size`	`numeric` mouth_size options:6,8 or 11.
`head_length`	`numeric` head_length options:20,30 or 50.
`head_texture_amount`	`numeric` head_texture_amount options:30,60 or 160.
`has_moustache`	`numeric` has_moustache options:0,0,0,1.
`moustache_length`	`numeric` moustache_length options:10,20,40.
`has_beard`	`numeric` has_beard options:0,0,0,0,0,1.
`has_teeth`	`numeric` has_teeth options:0,1,1.
`teeth_length`	`numeric` teeth_length options:5,8 or 15.
`teeth_space`	`numeric` teeth_space options:3,3.5 or 6.
`beard_length`	`numeric` beard_length options:20,30 or 50.
`eye_type`	`numeric` eye_type options:0 or 1.
`eye_size`	`numeric` eye_size options:8,10 or 28.
`jaw_size`	`numeric` jaw_size options:0.7,1 or 1.4.
`jaw_open`	`numeric` jaw_open options:0 or 1.

Value

Custom JS library.

Note

Note that some fish species might not be representable with this system, and passing "bad" params might produce weird results or crash the program.

Author(s)

Liuyong Ding <ly_ding@126.com>

Examples

```
## Not run:
# create and implement a custom JS library
custom_params(path = "./fishdraw.js",save = getwd())


## End(Not run)
```

fish_draw *Automatically generated fish drawings via ggplot2*

Description

Automatically generated fish drawings via ggplot2

Usage

```
fish_draw(data = NULL, x = x, y = y, group = group, ...)
```

Arguments

data	Path for *.json via get_polylines.
x	see aes for details
y	see aes for details
group	see aes for details
...	see geom_path for details

Value

ggplot object.

Author(s)

Liuyong Ding <ly_ding@126.com>

Examples

```
## Not run:
# fish drawings via plot
file <- system.file("extdata",package = "rfishdraw")
dat <- readRDS(paste0(file,"/","output_json.rds"))
for (i in seq(length(dat))) {
dat[[i]] <- cbind(dat[[i]],i)
}
plot(NA, xlim = c(0,520),ylim = c(-320,0),axes = F,xlab = " ",ylab = " ")
for(i in seq(length(dat))) {
  lines(x = dat[[i]][,1],y = -dat[[i]][,2], lwd=2, col = "blue")
}

# fish drawings via ggplot2
fish_draw()

## End(Not run)
```

get_polylines *Get outputs polylines (supported format svg, json, csv, etc.)*

Description

Get outputs polylines (supported format svg, json, csv, etc.)

Usage

```
get_polylines(
  path = "./fishdraw.js",
  name = NULL,
  format = "svg",
  output = "output.svg",
  draw_type = "random"
)
```

Arguments

`path` Path for fishdraw.js or customs.js via `custom_params`

`name` The name string is used as the name of the fish (printed in the drawing). If unspecified, a random pseudo-Latin name will be auto generated.

`format` Format options: svg (regular svg), smil (animated svg), csv (each polyline on a comma-separated line) and json.

`output` Outputs polylines (supported format svg, json, csv, etc.)

`draw_type` Draw_type options: random(by design fishdraw.js program is for randomly generated fishes),custom(by create and implement a custom customs.js for your favorite fish).

Value

Get outputs fish drawings polylines (in format .svg, .json and .csv).

Author(s)

Liuyong Ding <ly_ding@126.com>

Examples

```
## Not run:
# Get outputs polylines in svg
get_polylines(path = "inst/fishdraw.js",
              format = "svg",
              output = "output.svg",
              draw_type = "random")

# Get outputs polylines in json
get_polylines(path = "inst/fishdraw.js",
              format = "json",
              output = "output.json",
              draw_type = "random")

# Get outputs polylines in smil
get_polylines(path = "inst/fishdraw.js",
              format = "smil",
              output = "output.svg",
              draw_type = "random")

# browse animated svg
browseURL("inst/animated.svg")

## End(Not run)
```

Index