

Package: ijAnalysis (via r-universe)

October 17, 2026

Title Miscellaneous tools for image analysis tasks
Version 0.0.4
Description A miscellaneous toolbox for various image analysis tasks.
Initially contains tools for downstream analysis of my ImageJ
plugin Puncta Tracker and hence the ij name. Part of the y3628
analysis suite.
License GPL (>= 3)
Encoding UTF-8
Roxygen list(markdown = TRUE)
RoxygenNote 7.3.2
Imports y3628 (>= 0.0.3), dplyr, purrr, reticulate, rlang, vctrs,
stringr, jsonlite, tibble, readr, utils, ggplot2, readxl, damr,
ggetho, behavr, data.table, stats
Depends R (>= 4.1)
Suggests knitr, rmarkdown
VignetteBuilder knitr
Config/pak/sysreqs libicu-dev libpng-dev python3 libx11-dev
Repository <https://yeyuan98.r-universe.dev>
Date/Publication 2025-06-24 17:14:26 UTC
RemoteUrl <https://github.com/yeyuan98/ijAnalysis>
RemoteRef HEAD
RemoteSha d8a47c4d5dbc2d1f21bec84c0e63815b7780e897

Contents

.parse_meta	2
.selectNCT.index	3
actogram_loadFromMetadata	3
actogram_plot	4
behavior_loadMetadata	5

ctcf_plot	5
ctcf_read_csv	6
ij_nnd.compute	7
ij_RoiPointsSaver	7
jtk.init	8
jtkdist	8
jtkget	9
jtkstat	10
jtkx	10
nlsGroup	11
nnd.order	11
periodogram_loadClocklab	12
plot.boxPoint	12
plot.linePoint	13
plot.themes	14
plot_ggsave	15
rfish_count	15
rfish_read_dots	16
rfish_read_samples	17
spotInRoi_addMeta	17
spotInRoi_as.data.frame.df.nnd	18
spotInRoi_joinMeta	19
spotInRoi_nnd.sort	19
spotInRoi_read_csv	20
spotInRoi_selectNCT	20
subset_behavr	21

Index

22

.parse_meta	<i>Parse spot array JSON data to metaRcd.</i>
-------------	---

Description

Parse spot array JSON data to metaRcd.

Usage

.parse_meta(*ser.coord*, *ser.dist*)

Arguments

- ser.coord* Serialized coord.
- ser.dist* Serialized dist.

Value

A y3628::metaRcd object.

Examples

```
# Does not apply. Internal use only.
```

<code>.selectNCT.index</code>	<i>Get index for spotInRoi nuclear/full-cell data subsetting.</i>
-------------------------------	---

Description

Get index for spotInRoi nuclear/full-cell data subsetting.

Usage

```
.selectNCT.index(dfNucCyt, what)
```

Arguments

<code>dfNucCyt</code>	Parsed spotInRoi nuclear/full-cell data.
<code>what</code>	What to subset. must be nuclear, cytoplasm, or full.

Value

Indices for subsetting.

Examples

```
# Does not apply. Internal use only.
```

<code>actogram_loadFromMetadata</code>	<i>Load behavior activity data from metadata</i>
--	--

Description

Load behavior activity data from metadata

Usage

```
actogram_loadFromMetadata(path, data_dir = "./", ...)
```

Arguments

<code>path</code>	Path to metadata, must be commonly used delimited or excel.
<code>data_dir</code>	Path to the behavior data dir.
<code>...</code>	These dots are for future extensions and must be empty.

Value

behavr : : behavr behavior data table.

Examples

TODO

actogram_plot	<i>Plotting actogram</i>
---------------	--------------------------

Description

Plotting actogram

Usage

```
actogram_plot(  
  behavr,  
  file = NULL,  
  file.options = list(device = tools::file_ext(file), height = 9.14, width = 10.6)  
)
```

Arguments

- behavr behavr : : behavr behavior data table.
- file Optional path to save the plot. Backend is ggplot2::ggsave().
- file.options List to specify options for saving the plot.

Value

ggplot plot object.

Examples

TODO

behavior_loadMetadata	<i>Load behavior metadata</i>
-----------------------	-------------------------------

Description

Load behavior metadata

Usage

```
behavior_loadMetadata(path)
```

Arguments

path	Path to metadata file. See body for column spec.
------	--

Value

Metadata tibble.

Examples

```
# Not exported.
```

ctcf_plot	<i>Standardized plotting of CTCF data</i>
-----------	---

Description

This is de facto the versatile X-Y plot generator of this package. Supports multiple plot templates.

Usage

```
ctcf_plot(data, x, y, fill = NA, template = "boxPoint")
```

Arguments

data	tibble of CTCF data
x	x-axis variable for plotting
y	y-axis variable for plotting
fill	fill variable for plotting
template	which template to use, see details.

Details

All possible templates are not exported functions `plot.` in this package. Supported ones are:

- "linePoint" - uses `plot.linePoint()` - point plot with summary line
- "boxPoint" - uses `plot.boxPoint()` - point plot with box

Value

ggplot object

Examples

```
# Will warn about the `am` continuous color variable
ctcf_plot(mtcars, cyl, mpg, am, template = "linePoint")
# Will create a boxPoint plot. Note that for boxPoint X must be discrete.
ctcf_plot(
  mtcars |> dplyr::mutate(gear = as.character(gear), am = as.character(am)),
  am, mpg, gear)
```

ctcf_read_csv

Read CTCF results table

Description

Read CTCF results table

Usage

```
ctcf_read_csv(path, ...)
```

Arguments

<code>path</code>	Path to the CTCF analysis csv
<code>...</code>	Forwarded to <code>readr::read_csv</code>

Value

tibble of CTCF table

Examples

```
#TODO
```

ij_nnd.compute	<i>Compute NND metrics for a pair of spot sets.</i>
----------------	---

Description

Compute NND metrics for a pair of spot sets.

Usage

```
ij_nnd.compute(foreach, to, z.proximal)
```

Arguments

foreach	For each spot in foreach,
to	compute its distance to the nearest neighbor in to spot set.
z.proximal	Only spots in close enough z-proximity is computed.

Value

y3628::metaRcrd object storing the NND results. For spots that have no Z-proximal spots in the to spot set, NAs are returned.

Examples

```
# TODO
```

ij_RoiPointsSaver	<i>Initialize an ImageJ ROI point set saver function</i>
-------------------	--

Description

Initialize an ImageJ ROI point set saver function

Usage

```
ij_RoiPointsSaver(use_python)
```

Arguments

use_python	Path to the Python binary to use.
------------	-----------------------------------

Value

A ROI saver function.

Examples

```
# TODO
```

jtk.init	<i>JTK Cycle V3.1 Adaptation</i> jtk.init
----------	---

Description

Initialize the JTK environment for all periods

Usage

```
jtk.init(periods, interval = 1)
```

Arguments

periods	Periods to test on (in units of spacing interval).
interval	Spacing of time points (in physical unit like hours). For example, <code>jtk.init(9:15, 2)</code> means that data time point spacing is 2-hour and possible periods would be 18-hour to 30-hour (2-hour spacing).

Value

None; Run only for side effects on JTK_env.

Examples

```
# Refer to the JTK Cycle guide.
```

jtkdist	<i>JTK Cycle V3.1 Adaptation</i> jtkdist
---------	--

Description

JTK Cycle V3.1 Adaptation jtkdist

Usage

```
jtkdist(timepoints, reps = 1, normal = FALSE, alt = FALSE)
```

Arguments

timepoints	Number of time points in the data.
reps	Number of replicates for each time point.
normal	Not used. Refer to JTK Cycle paper.
alt	Not used. Refer to JTK Cycle paper.

Value

Not used. Refer to JTK Cycle paper.

Examples

```
# Refer to the JTK Cycle guide.
```

`jtkget`*JTK Cycle V3.1 Adaptation*

Description

This is getter function to fetch computation results of `jtkx`.

Usage

```
jtkget(get.AMP.CI = FALSE)
```

Arguments

<code>get.AMP.CI</code>	Bool, whether to get amplitude confidence interval, which will be length=2 numeric vector.
-------------------------	--

Details

In this adaptation, `jtkx` results are stored internally in a package environment `JTK_env`. This function is a getter of the environment.

Value

Named numeric vector of current `jtkx` results stored in `JTK_env`. If `get.AMP.CI==TRUE`, only return the amplitude confidence interval.

Examples

```
# Refer to the JTK Cycle guide.
```

jtkstat	<i>JTK Cycle V3.1 Adaptation</i> jtkstat
---------	--

Description

jtkstat: calculate the p-values for all (period,phase) combos. v3.1 modified to analyze data with missing values.

Usage

```
jtkstat(z)
```

Arguments

z	Numeric vector of data at each time point.
---	--

Value

None; Run only for side effects on JTK_env.

Examples

```
# Internal use by `jtkx` only.
```

jtkx	<i>JTK Cycle V3.1 Adaptation</i> jtkx
------	---------------------------------------

Description

jtkx: integration of jtkstat and jtkdist for repeated use.

Usage

```
jtkx(z, ampci = FALSE, conf = 0.8)
```

Arguments

z	Numeric vector of data at each time point.
ampci	Boolean, whether to compute amplitude CI and p-value.
conf	Confidence level of amplitude. Only useful if ampci == TRUE.

Value

None; Run only for side effects on JTK_env.

Examples

```
# Refer to the JTK Cycle guide.
```

nlsGroup	nls() <i>fitting of multiple groups</i>
----------	---

Description

nls() fitting of multiple groups

Usage

```
nlsGroup(.data, var.group, formula, ...)
```

Arguments

.data	data.frame with variables to fit with additional group variables.
var.group	group variables. Follow y3628::grouper().
formula	Formula to run nls()
...	Additional parameters forwarded to nls(). Most notably, you need to provide start values unless you are using self-starting models. See stats::nls().

Value

Data frame of fitting parameters, with group variables.

Examples

```
# Puromycin, self-starting SSmicmen kinetics
nlsGroup(Puromycin, state, rate~SSmicmen(conc,Vm,K))
# Custom formula. Must provide start values.
michaelis_menten <- rate ~ (Vm)*conc/(K+conc)
nlsGroup(Puromycin, state, michaelis_menten, start=list(Vm=100,K=0.01))
```

nnd.order	<i>Order permutation for a NND result record</i>
-----------	--

Description

Order permutation for a NND result record

Usage

```
nnd.order(nnd.results, ...)
```

Arguments

nnd.results	y3628::metaRcrd of NND result.
...	Passed to base::order

Value

Order permutation using the nnd field.

Examples

Does not apply. Internal use only.

periodogram_loadClocklab	<i>Load Clocklab periodogram batch export table</i>
--------------------------	---

Description

Load Clocklab periodogram batch export table

Usage

periodogram_loadClocklab(data, meta, ...)

Arguments

- data Path to the Clocklab periodogram csv export file.
- meta Path to the metadata file.
- ... These dots are for future extensions and must be empty.

Value

Periodogram table with metadata columns

Examples

TODO

plot.boxPoint	<i>Boxplot with data points</i>
---------------	---------------------------------

Description

Boxplot with data points

Usage

```
## S3 method for class 'boxPoint'  
plot(data, x, y, fill, theme = "classic")
```

Arguments

data	Data
x	x, must be already quoted
y	y, must be already quoted
fill	fill, must be already quoted
theme	what theme to apply, see plot.themes().

Value

ggplot object

Examples

```
# Internal use
```

plot.linePoint	<i>Point plot with lines connecting 'mean' values.</i>
----------------	--

Description

If color not specified or equal to x, whole data will be one color.

Usage

```
## S3 method for class 'linePoint'
plot(
  data,
  x,
  y,
  color = NA,
  theme = "classic",
  mean_func = mean,
  sd_func = function(x) stats::sd(x)/sqrt(length(x))
)
```

Arguments

data	Data
x	x, must be already quoted
y	y, must be already quoted
color	color, must be already quoted
theme	what theme to apply, see plot.themes()
mean_func	function to compute 'mean' (sample representative value)
sd_func	function to compute 'sd' (sample spread)

Value

ggplot object

Examples

```
# Internal use
```

plot.themes	<i>Standardized themes for plotting</i>
-------------	---

Description

Themes used by different plots in this package for standardized behavior.

Usage

```
## S3 method for class 'themes'  
plot(type = "classic")
```

Arguments

type standardized theme name, see details.

Details

Implemented themes:

- classic: Based on `ggplot2::theme_classic()`.

Value

GG theme object.

Examples

```
# Internal use
```

plot_ggsave	<i>Simple wrapper of ggsave for saving plots.</i>
-------------	---

Description

Simple wrapper of ggsave for saving plots.

Usage

```
plot_ggsave(  
  plots = list(),  
  widths,  
  heights,  
  units = "in",  
  dir = "plots_eps",  
  ...  
)
```

Arguments

plots	Named list of plots to save.
widths	Width of the plots. Either length one or same as plots.
heights	Height of the plots. Either length one or same as plots.
units	Unit of width and height
dir	Directory under which to save the plot
...	These dots are for future extensions and must be empty.

Examples

```
# TODO
```

rfish_count	<i>Count number of dots per cell</i>
-------------	--------------------------------------

Description

Count number of dots per cell

Usage

```
rfish_count(samples, dots, by = c("sample", "image"), ...)
```

Arguments

<code>samples</code>	tibble of samples table, <code>rfish_read_samples()</code>
<code>dots</code>	tibble of dots table, <code>rfish_read_dots()</code>
<code>by</code>	group by which variables before counting
<code>...</code>	<tidy-select> extra columns (from samples table) to retain

Value

Count table (number of dots per cell).

Examples

```
# TODO
```

<code>rfish_read_dots</code>	<i>Read dots table from snakemake pipeline output</i>
------------------------------	---

Description

Read dots table from snakemake pipeline output

Usage

```
rfish_read_dots(path, ...)
```

Arguments

<code>path</code>	Path to the dots csv
<code>...</code>	Forwarded to <code>readr::read_csv</code>

Value

tibble of dots table

Examples

```
#TODO
```

rfish_read_samples	<i>Read samples table from snakemake pipeline output</i>
--------------------	--

Description

Read samples table from snakemake pipeline output

Usage

```
rfish_read_samples(path, ...)
```

Arguments

path	Path to the samples excel.
...	Forwarded to <code>readxl::read_excel</code>

Value

tibble of samples table

Examples

```
#TODO
```

spotInRoi_addMeta	<i>Add metadata column to spotInRoi data</i>
-------------------	--

Description

Add metadata column to spotInRoi data

Usage

```
spotInRoi_addMeta(df, idMap, meta, ...)
```

Arguments

df	spotInRoi table, see details.
idMap	identifier map table, see details.
meta	metadata table.
...	<code><tidy-select></code> columns to select from the metadata table.

Details

spotInRoi data here takes a generalized meaning - a data frame with two id columns \$tp, \$meas, plus any number of additional columns.

The identifier map table must contain two id columns \$tp, \$meas, plus additional columns that are identifiers in the metadata table.

Value

spotInRoi data with the specified metadata columns.

Examples

```
# TODO
```

```
spotInRoi_as.data.frame.df.nnd
```

Expand NND results as data frame entries

Description

Expand NND results as data frame entries

Usage

```
spotInRoi_as.data.frame.df.nnd(df.nnd, nnd.column = "result")
```

Arguments

<code>df.nnd</code>	Data frame with a NND result column
<code>nnd.column</code>	Character name of the column that stores NND results.

Value

Expanded data frame with each NND value as one row

Examples

```
# TODO
```

spotInRoi_joinMeta	<i>Join spotInRoi data with metadata columns</i>
--------------------	--

Description

Join spotInRoi data with metadata columns

Usage

```
spotInRoi_joinMeta(df, df.meta, by, ...)
```

Arguments

df	spotInRoi data frame.
df.meta	metadata data frame.
by	columns to perform join
...	<tidy-select> columns to select from the metadata data frame.

Value

spotInRoi data frame joined with metadata.

Examples

```
# Not exported
```

spotInRoi_nnd.sort	<i>Sort spotInRoi data based on NND results</i>
--------------------	---

Description

Sort spotInRoi data based on NND results

Usage

```
spotInRoi_nnd.sort(df.nnd, nnd.column = "result", ...)
```

Arguments

df.nnd	spotInRoi data with NND results.
nnd.column	Character name of the column that stores NND results.
...	Passed to base::order for order permutation.

Value

Sorted spotInRoi results.

Examples

```
# TODO
```

spotInRoi_read_csv	<i>Read data for spotInRoi analysis of the Puncta Tracker ImageJ plugin.</i>
--------------------	--

Description

Read data for spotInRoi analysis of the Puncta Tracker ImageJ plugin.

Usage

```
spotInRoi_read_csv(file, parse.spotOut = FALSE, ...)
```

Arguments

file	File passed to readr::read_csv.
parse.spotOut	Whether to parse spotOut array data.
...	Other parameters passed to readr::read_csv.

Value

tibble of the spotInRoi analysis result.

Examples

```
# TODO
```

spotInRoi_selectNCT	<i>Predefined subsetting of spotInRoi nuclear/full-cell data.</i>
---------------------	---

Description

Predefined subsetting of spotInRoi nuclear/full-cell data.

Usage

```
spotInRoi_selectNCT(dfNucCyt, what)
```

Arguments

dfNucCyt Parsed spotInRoi nuclear/full-cell data.
 what What to subset. must be nuclear, cytoplasm, or full.

Value

Subsetted spotInRoi nuclear/full-cell data.

Subset defaults

Predefined subsetting rules are as follows:

nuclear - for each cell select the ROI with the smallest equi.D

full - for each cell select the ROI with the largest equi.D

cytoplasm - currently not implemented and will error.

Examples

```
# TODO
```

subset_behavr	<i>Simple subset of behavr : : behavr behavior tables</i>
---------------	---

Description

Simple subset of behavr : : behavr behavior tables

Usage

```
subset_behavr(behavr, expr)
```

Arguments

behavr Behavior table to subset
 expr An expression that evaluates to logical. See details.

Details

Right now the subsetting creates an intermediate logical vector, which can be very large.

Value

Subsetted behavior table.

Examples

```
# Not exported
```

Index

[.parse_meta](#), [2](#)
[.selectNCT.index](#), [3](#)

[actogram_loadFromMetadata](#), [3](#)
[actogram_plot](#), [4](#)

[behavior_loadMetadata](#), [5](#)

[ctcf_plot](#), [5](#)
[ctcf_read_csv](#), [6](#)

[ij_nnd.compute](#), [7](#)
[ij_RoiPointsSaver](#), [7](#)

[jtk.init](#), [8](#)
[jtkdist](#), [8](#)
[jtkget](#), [9](#)
[jtkstat](#), [10](#)
[jtkx](#), [10](#)

[nlsGroup](#), [11](#)
[nnd.order](#), [11](#)

[periodogram_loadClocklab](#), [12](#)
[plot.boxPoint](#), [12](#)
[plot.linePoint](#), [13](#)
[plot.themes](#), [14](#)
[plot_ggsave](#), [15](#)

[rfish_count](#), [15](#)
[rfish_read_dots](#), [16](#)
[rfish_read_samples](#), [17](#)

[spotInRoi_addMeta](#), [17](#)
[spotInRoi_as.data.frame.df.nnd](#), [18](#)
[spotInRoi_joinMeta](#), [19](#)
[spotInRoi_nnd.sort](#), [19](#)
[spotInRoi_read_csv](#), [20](#)
[spotInRoi_selectNCT](#), [20](#)
[subset_behavr](#), [21](#)