

Package: RImageJROI (via r-universe)

October 13, 2026

Title Read and Write 'ImageJ' Region of Interest (ROI) Files

Description Provides functions to read and write 'ImageJ'
([<https://imagej.net>](https://imagej.net)) Region of Interest (ROI) files, to plot
the ROIs and to convert them to 'spatstat'
([<https://spatstat.org/>](https://spatstat.org)) spatial patterns.

Version 0.1.3

URL <https://github.com/davidcsterratt/RImageJROI>

BugReports <https://github.com/davidcsterratt/RImageJROI/issues>

Date 2024-08-17

Depends R ($\geq$ 3.0.2)

Imports spatstat.geom

Suggests png, testthat

License GPL-3

RoxygenNote 7.2.3

Encoding UTF-8

Repository <https://davidcsterratt.r-universe.dev>

Date/Publication 2024-08-17 12:15:24 UTC

RemoteUrl <https://github.com/davidcsterratt/rimagejroi>

RemoteRef HEAD

RemoteSha 9580f007b5c702adacaa8993b7dea0f9a23ac6c1

Contents

ij2spatstat	2
plot.ijroi	3
plot.ijzip	5
print.ijroi	6
read.ijroi	7
read.ijzip	8
RImageJROI	9
write.ijroi	9
write.ijzip	10

Index

11

ij2spatstat	<i>Convert 'ijroi' and 'ijzip' objects to spatstat spatial patterns</i>
-------------	---

Description

Converts [ijroi](#) and [ijzip](#) objects to a list of [spatstat.geom](#) spatial patterns.

Usage

```
ij2spatstat(
  X,
  window = NULL,
  pattern.type = NULL,
  unitname = NULL,
  scale = 1,
  return.type = FALSE,
  convert.only = NULL
)
```

Arguments

<code>X</code>	ijroi or ijzip object to be converted.
<code>window</code>	the window for returned spatial patterns. Can be an owin object defining a common window for all returned patterns, a character string 'range' leading to a common window based range of all returned patterns, or NULL (default) leading to separate windows for each pattern.
<code>pattern.type</code>	a character string specifying the desired pattern type to be returned (ppp , psp or owin). Works only if <code>X</code> is an ijroi object. Ignored otherwise. Defaults to an appropriate pattern type depending on the ROI type (see 'Details').
<code>unitname</code>	Name of the unit of length for the resulting window(s) (see owin).
<code>scale</code>	A numeric value defining the scale of photograph in pixels / <code>unitname</code> . Defaults to 1.
<code>return.type</code>	should the type of ROI object(s) be returned in addition to spatstat.geom spatial patterns? Defaults to FALSE.
<code>convert.only</code>	a character vector specifying the <code>strType</code> of ROI objects to be converted (see plot.ijroi for possible pattern types). Pattern types not mentioned will not be converted. Works only if <code>X</code> is an ijzip object. Ignored otherwise.

Details

The function converts [ijroi](#) and [ijzip](#) objects to [spatstat.geom](#) spatial patterns for further calculations with the objects. By default, areal types ("rect", "oval", "ELLIPSE", "polygon") are converted to [owin](#) objects. Line types ("line" (including "ARROW"), "freeline", "polyline", "angle", "free-hand" (excluding "ELLIPSE")) are converted to [psp](#) objects and "point" types to [ppp](#) objects.

Value

Returns a list of [spatstat.geom](#) patterns of appropriate type (see 'Details'). If `return.type = TRUE` returns a list with two levels specifying the `spatstat.geom` pattern and the ROI type.

Author(s)

Mikko Vihtakari

See Also

[read.ijroi](#) [read.ijzip](#)

Examples

```
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "ijzip.zip")
x <- read.ijzip(file)
ij2spatstat(x)
```

plot.ijroi

Plot ijroi object

Description

Plots ImageJ ROI objects using the '[base](#)' [graphics](#) package.

Usage

```
## S3 method for class 'ijroi'
plot(x, add = FALSE, xlab = "", ylab = "", main = "", asp = 1, ...)
```

Arguments

<code>x</code>	The <code>ijroi</code> object.
<code>add</code>	Whether to add to an existing plot.
<code>xlab</code>	a title for the x axis: title .
<code>ylab</code>	a title for the y axis: title .
<code>main</code>	an overall title for the plot: title .
<code>asp</code>	numeric defining the aspect ratio y/x: see plot.window . Defaults to 1.
<code>...</code>	Additional parameters.

Details

ImageJ ROI objects created with following tools are plotted using following graphics commands:

- Rectangle tool ("rect") [rect](#). Plotted based on coordinates.
- Oval selections ("oval") [polygon](#). Plotted based on equation.
- Freehand selections ("freehand") [lines](#). Plotted based on coordinates.
- Elliptical selections ("freehand", "ELLIPSE") [lines](#). Plotted based on equation.
- Point Tool and Multi-Point Tool ("point") [points](#). Plotted based on coordinates.
- Straight Line ("line") [lines](#). Plotted based on coordinates.
- Arrow tool ("line", "ARROW") [arrows](#). Plotted based on coordinates. Stroke width passed to [lwd](#) argument.
- Segmented Line ("polyline") [lines](#). Plotted based on coordinates.
- Freehand Line ("freeline") [lines](#). Plotted based on coordinates.

All graphics allow the additional parameters from appropriate functions. Aspect ratio (asp) is 1 by default leading to correct representation of ImageJ objects. If correct representation is not important, set `asp = NA` to use the R base-graphics default setting.

Author(s)

David Sterratt, Mikko Vihtakari

See Also

[read.ijroi](#), [read.ijzip](#), [plot.ijzip](#)

Examples

```
# type 0 'polygon' ROIs are plotted using lines()
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "polygon.roi")
x <- read.ijroi(file)
plot(x, col = "red")

# type 1 'rect' ROIs are plotted using rect()
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "rect.roi")
x <- read.ijroi(file)
plot(x, border = "red")

# type 2 'oval' ROIs are plotted using polygon()
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "oval.roi")
x <- read.ijroi(file)
plot(x, border = "red")

# type 3 'line' ROIs (among others listed in 'details') are plotted using lines()
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "line.roi")
x <- read.ijroi(file)
plot(x, col = "red")

# type 3 arrows are a subtype of 'line'. Plotted using arrows(). The stroke width is
```

```

# carried over. To change width, use lwd argument
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "arrow.roi")
x <- read.ijroi(file)
plot(x, col = "red")

# type 4 'freeline' ROIs are plotted using lines()
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "freehand_line.roi")
x <- read.ijroi(file)
plot(x, col = "red")

# type 5 'polyline' ROIs are plotted using lines()
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "segmented_line.roi")
x <- read.ijroi(file)
plot(x, col = "red")

# type 7 'freehand' selection ROIs are plotted using lines()
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "freehand_selection.roi")
x <- read.ijroi(file)
plot(x, col = "red")

# type 7 Objects created using 'Elliptical selections' tool are also saved as
# 'freehand', but with subtype 'ELLIPSE'. The coordinates for this type are flawed
# and plotting is done using equation for an ellipse
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "elliptical.roi")
x <- read.ijroi(file)
plot(x, border = "red")
lines(x$coords[,1], x$coords[,2]) ## plotted based on coordinates.

# type 10 'point' ROIs are plotted using points()
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "multi_point.roi")
x <- read.ijroi(file)
plot(x, col = "red")

# If following is shown as a (round) circle, asp = 1
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "circle.roi")
x <- read.ijroi(file)
plot(x, border = "red")

# text is stored as type 'rect' with subtype 'TEXT'. Currently
# only the outlining rectangle is returned
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "text.roi")
x <- read.ijroi(file)
plot(x, border = "red")

```

plot.ijzip

Plot ijzip object

Description

Plots .zip files containing ImageJ ROI objects using the ['base' graphics](#) package.

Usage

```
## S3 method for class 'ijzip'
plot(x, add = FALSE, xlab = "", ylab = "", main = "", asp = 1, ...)
```

Arguments

x	The ijzip object.
add	Whether to add to an existing plot.
xlab	a title for the x axis: see title .
ylab	a title for the y axis: see title .
main	an overall title for the plot: see title .
asp	numeric defining the aspect ratio y/x: see plot.window . Defaults to 1.
...	Arguments to be passed to methods, such as graphical parameters (see par).

Details

The function loops [plot.ijroi](#) plotting function over all elements in x. See `plot.ijroi` for further details.

Author(s)

Mikko Vihtakari, David Sterratt

See Also

[read.ijzip](#), [plot.ijroi](#)

Examples

```
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "ijzip.zip")
x <- read.ijzip(file)
plot(x)
```

print.ijroi

Print ijroi objects

Description

Print ijroi objects

Usage

```
## S3 method for class 'ijroi'
print(x, all = FALSE, ...)
```

Arguments

`x` ijroi object to be printed.

`all` logical indicating whether to print all information from ijroi object as opposed to a subset of relevant information. Defaults to FALSE.

`...` further arguments passed to [print](#).

Author(s)

Mikko Vihtakari, David Sterratt

See Also

[read.ijroi](#)

read.ijroi	<i>Read an ImageJ ROI file</i>
------------	--------------------------------

Description

Read an **ImageJ** ROI file. This returns a structure containing the ImageJ data.

Usage

```
read.ijroi(file, verbose = FALSE)
```

Arguments

`file` Name of ImageJ ROI file to read

`verbose` Whether to report information

Value

A structure of class `ijroi` containing the ROI information

Author(s)

David Sterratt

See Also

[plot.ijroi](#) for plotting single ROI objects.

[read.ijzip](#) for reading several ROI objects from .zip files.

Examples

```
library(png)
path <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi")
im <- as.raster(readPNG(file.path(path, "imagej-logo.png")))
plot(NA, NA, xlim=c(0, ncol(im)), ylim=c(nrow(im), 0), asp=1)
rasterImage(im, 0, nrow(im), ncol(im), 0, interpolate=FALSE)
r <- read.ijroi(file.path(path, "rect.roi"))
plot(r, TRUE)
r <- read.ijroi(file.path(path, "polygon.roi"))
plot(r, TRUE)
r <- read.ijroi(file.path(path, "oval.roi"))
plot(r, TRUE)
```

read.ijzip	<i>Read ImageJ zip file containing several ROI files</i>
------------	--

Description

A wrapper function, which reads a zip file containing ImageJ ROI files using [read.ijroi](#) function.

Usage

```
read.ijzip(file, names = TRUE, list.files = FALSE, verbose = FALSE)
```

Arguments

file	zip file containing a collection of ImageJ ROI files
names	Logical, indicating whether the ROI file names should be used as names for the elements in the list (see Return). If FALSE a sequence of names specifying the type of ROI is automatically generated.
list.files	logical, indicating whether a data.frame of ROI files in file should be returned instead of a list of results. Defaults to FALSE. If TRUE equals to <code>unzip(file, list = TRUE)</code> .
verbose	Whether to report information (see read.ijroi).

Value

An object of class `ijzip` containing a list of the coordinates and types of ImageJ ROIs. Each element is named after option specified in `names`.

Author(s)

Mikko Vihtakari

See Also

[read.ijroi](#), [plot.ijzip](#).

Examples

```
file <- file.path(system.file(package = "RImageJROI"), "extdata", "ijroi", "ijzip.zip")
x <- read.ijzip(file)
plot(x)
```

RImageJROI

*Read and write ImageJ Region of Interest (ROI) files***Description**

Provides functions to read and write ImageJ (<https://imagej.net/>) Region of Interest (ROI) files, to plot the ROIs and to convert them as spatstat (<https://spatstat.org/>) spatial patterns.

Details

ImageJ ROI files can be read into R using the `read.ijroi` and `read.ijzip` functions, resulting in `ijroi` and `ijzip` objects.

The objects can be plotted using generic `plot` command and converted to `spatstat.geom` spatial patterns by using `ij2spatstat` function.

The `ijroi` and `ijzip` objects can be written to file using the `write.ijroi` and `write.ijzip` functions.

write.ijroi

*Write an ImageJ ROI file.***Description**

Write an ImageJ ROI file.

Usage

```
write.ijroi(file, roi, verbose = TRUE)
```

Arguments

file	Name of ImageJ ROI file to write
roi	A structure of class <code>ijroi</code> containing the ROI information
verbose	Whether to report information

Author(s)

Le Gao

See Also

`read.ijroi` for reading an ROI file

`write.ijzip`*Write ImageJ zip file containing several ROI files*

Description

Write or add to a zip archive containing ImageJ ROI files using the [write.ijroi](#) function.

Usage

```
write.ijzip(file, roi, verbose = TRUE)
```

Arguments

<code>file</code>	zip archive to write that will contain a collection of ImageJ ROI files
<code>roi</code>	A list of ROIs
<code>verbose</code>	Whether to report information

Author(s)

Le Gao, David Sterratt

See Also

[write.ijroi](#)

Index

'base' graphics, [3](#), [5](#)

arrows, [4](#)

ij2spatstat, [2](#), [9](#)

ijroi, [2](#)

ijzip, [2](#)

lines, [4](#)

lwd, [4](#)

owin, [2](#)

par, [6](#)

plot, [9](#)

plot.ijroi, [2](#), [3](#), [6](#), [7](#)

plot.ijzip, [4](#), [5](#), [8](#)

plot.window, [3](#), [6](#)

points, [4](#)

polygon, [4](#)

ppp, [2](#)

print, [7](#)

print.ijroi, [6](#)

psp, [2](#)

range, [2](#)

read.ijroi, [3](#), [4](#), [7](#), [7–9](#)

read.ijzip, [3](#), [4](#), [6](#), [7](#), [8](#), [9](#)

rect, [4](#)

RImageJROI, [9](#)

spatstat.geom, [2](#), [3](#), [9](#)

title, [3](#), [6](#)

window, [2](#)

write.ijroi, [9](#), [9](#), [10](#)

write.ijzip, [9](#), [10](#)