Package: SpatialData.plot (via r-universe)

November 25, 2026

Title SpatialData visualization

Depends R (>= 4.5), spatialdataR

Version 1.0.0

Description Visualization suit for 'spatialdataR'. Current functionality includes handling of multiscale 'images', visualizing 'labels', 'points', and 'shapes'. For the latter, POINT, POLYGON, and MULTIPOLYGON geometries are supported. For all but 'images', there is support to pass information from annotation 'tables' as aesthetics (e.g., 'colData' or 'assay' data).

Imports BiocGenerics, dplyr, farver, ggnewscale, ggplot2 (>= 4.0.0), MatrixGenerics, methods, rlang, sf, S4Vectors

Suggests BiocStyle, DelayedArray, knitr, magick, patchwork, SingleCellExperiment, testthat, vdiffr, ZarrArray

biocViews Spatial, SingleCell, Visualization, Infrastructure, ImmunoOncology, GeneExpression, Transcriptomics

License Artistic-2.0

Encoding UTF-8

VignetteBuilder knitr

URL https://github.com/HelenaLC/SpatialData.plot

BugReports https://github.com/HelenaLC/SpatialData.plot/issues

Config/roxygen2/version 8.1.1

Config/pak/sysreqs libabsl-dev libblosc-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libpng-dev libxml2-dev libzstd-dev libssl-dev libproj-dev python3 libsqlite3-dev libudunits2-dev zlib1g-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:57:16 UTC

RemoteUrl https://github.com/bioc/SpatialData.plot

RemoteRef RELEASE_3_24

RemoteSha da40070be6326f6a9ea09d08477f3960ce1b6ab1

Contents

blobs 2
plotImage 2
plotLabel 4
plotPoint 5
plotShape 6
plotSpatialData 8
scalebar 8

Index **10**

`blobs` *'SpatialData' .zarr toy datasets*

Description

data were retrieved on Nov. 11th, 2024, from here.

Value

a `SpatialData` .zarr store of toy example data with all types of elements represented: (multiscale) RGB `image`, (multiscale) `label`, `point`, circle and (multi)polygon `shape`, `table` annotation. In addition, all types of coordinate transformations are represented: identity, scale, translation, affine, and sequence.

Examples

```
x <- file.path("extdata", "blobs.zarr")
x <- system.file(x, package="spatialdataR")
(x <- readSpatialData(x))
```

`plotImage` *Add image layer to SpatialData plot*

Description

Add image layer to SpatialData plot

Usage

```
plotImage(
  i = 1,
  j = NULL,
  k = NULL,
  ch = NULL,
  c = NULL,
  cl = NULL,
  t = NULL,
  z = NULL,
  x = NULL
)
```

Arguments

i	Index or name of image to plot.
j	Index or name of coordinate system to use. If NULL, will be inherited from plotSpatialData().
k	Index of the scale to render; by default (NULL), will auto-select scale in order to minimize memory-usage and blurring for a target size of 800 x 800px; use Inf to plot the lowest resolution available.
ch	Image channel(s) to be used for plotting (defaults to the first channel(s) available); use channels() to see which channels are available for a given SpatialDataImage
c	Character vector; colors to use for each channel.
cl	List of length-2 numeric vectors (non-negative, increasing); specifies channel-wise contrast limits - defaults to [0, 1] for all ([0, 1/3] when image(x, i) is an RGB image). For convenience, any NULL = [0, 1], and n = [0, n].
t, z	Integer scalar to indicate a specific time- or z-slice; if left unspecified (default NULL), will perform a max-projection.
x	SpatialData object. If NULL, will be inherited from plotSpatialData().

Value

An object of type sd_image, which can be added to an existing ggplot.

Examples

```
x <- file.path("extdata", "blobs.zarr")
x <- system.file(x, package="spatialdataR")
x <- readSpatialData(x, tables=FALSE)

ms <- lapply(seq(3), \(.)
  plotSpatialData(x) +
  plotImage(2, k=.))
patchwork::wrap_plots(ms)

# custom colors
```

```
cmy <- c("cyan", "magenta", "yellow")
plotSpatialData(x) + plotImage(c=cmy)

# contrast limits
cl <- rep(list(c(0, 1/3)), 3)
plotSpatialData(x, ct="global") +
  plotImage(k=1, c=cmy, cl=cl) +
  plotShape(i="blobs_circles", fill="pink") +
  plotPoint(i="blobs_points", colour="instance_id")
```

`plotLabel` *Add label layer to SpatialData plot*

Description

Add label layer to SpatialData plot

Usage

```
plotLabel(
  i = 1,
  j = NULL,
  k = NULL,
  c = NULL,
  a = 0.5,
  pal = NULL,
  nan = NA,
  assay = 1,
  t = NULL,
  z = NULL,
  x = NULL
)
```

Arguments

`i`	Index or name of label to plot.
`j`	Index or name of coordinate system to use. If `NULL`, will be inherited from `plotSpatialData()`.
`k`	Index of the scale to render; by default (NULL), will auto-select scale in order to minimize memory-usage and blurring for a target size of 800 x 800px; use Inf to plot the lowest resolution available.
`c`	determines label colors; the default (NULL), gives a binary image of whether or not a pixel is non-zero; alternatively, a character string specifying a `colData` column or row name in an annotation `table`.
`a`	scalar numeric in [0, 1]; alpha value passed to `geom_tile`.

pal	character vector; color for discrete/continuous values (interpolated automatically when insufficient values are provided). When left unspecified, color will be sampled at random.
nan	character string; color for missing values (hidden by default).
assay	character string; in case of c denoting a row name, specifies which assay data to use (see getTable).
t, z	Integer scalar to indicate a specific time- or z-slice; if left unspecified (default NULL), will perform a max-projection.
x	SpatialData object. If NULL, will be inherited from plotSpatialData().

Value

An object of type sd_label, which can be added to an existing ggplot.

Examples

```
x <- file.path("extdata", "blobs.zarr")
x <- system.file(x, package="spatialdataR")
x <- readSpatialData(x)

i <- "blobs_labels"
p <- plotSpatialData(x)

# simple binary image
p + plotLabel(i)

# mock up some extra data
t <- getTable(x, i)
t$id <- sample(letters, ncol(t))
table(x) <- t

# coloring by 'colData'
t <- getTable(x, i)
t$id <- sample(letters, ncol(t))
table(x) <- t
plotSpatialData(x) + plotLabel(i=i, c="id")

# coloring by 'assay' data
plotSpatialData(x) + plotLabel(i=i,
  c="channel_1_sum",
  pal=c("lavender", "blue"))
```

---

plotPoint *Add point layer to SpatialData plot*

Description

Add point layer to SpatialData plot

Usage

```
plotPoint(i = 1, j = NULL, x = NULL, n = NULL, key = NULL, ...)
```

Arguments

i	Index or name of point to plot.
j	Index or name of coordinate system to use. If NULL, will be inherited from plotSpatialData().
x	SpatialData object. If NULL, the object will be inherited from plotSpatialData().
n	Positive integer scalar; if not NULL, will randomly sample n points for plotting. Use set.seed() before plotting to get a reproducible subset.
key	Character vector; if not NULL, only plot points where feature_key(point(x, i)) is a value in key.
...	Optional aesthetic arguments (for example, fill or color) passed to geom_sf.

Value

An object of type sd_point, which can be added to an existing ggplot.

Examples

```
x <- file.path("extdata", "blobs.zarr")
x <- system.file(x, package="spatialdataR")
x <- readSpatialData(x)

i <- "blobs_points"
p <- plotSpatialData(x)
p + plotPoint(i=i)                    # simple
p + plotPoint(i=i, colour="genes")     # discrete
p + plotPoint(i, colour="instance_id") # continuous
```

---

plotShape *Add shape layer to SpatialData plot*

Description

Add shape layer to SpatialData plot

Usage

```
plotShape(
  i = 1,
  j = NULL,
  assay = 1,
  x = NULL,
```

```
  n = NULL,
  as = c("border", "solid", "centroid"),
  ...
)
```

Arguments

i	Index or name of shape to plot.
j	Index or name of coordinate system to use. If NULL, will be inherited from plotSpatialData().
assay	Index or name of an assay; in case of fill or colour denoting a row name, specifies which assay data to use (see getTable). Ignored when x is a SpatialDataPoint.
x	SpatialData object. If NULL, the object will be inherited from plotSpatialData().
n	Positive integer scalar; if not NULL, will randomly sample n points for plotting. Use set.seed() before plotting to get a reproducible subset.
as	Character scalar defining how to plot shapes. Must be one of "border" (just plot the outline, the default), "solid" (plot filled shapes) or "centroid" (just plot a point at the coordinates of the centroid of the shape).
...	Optional aesthetic arguments (for example, fill or color) passed to geom_sf.

Value

An object of type sd_shape, which can be added to an existing ggplot.

Examples

```
x <- file.path("extdata", "blobs.zarr")
x <- system.file(x, package="spatialdataR")
x <- readSpatialData(x)

# shapes
p <- plotSpatialData(x)
a <- p + plotShape(i="blobs_polygons")
b <- p + plotShape(i="blobs_multipolygons")
c <- p + plotShape(i="blobs_circles")
patchwork::wrap_plots(a, b, c)

# layered
p +
  plotShape("blobs_circles", fill="pink") +
  plotShape("blobs_polygons", colour="red")
patchwork::wrap_plots(a, b)
```

plotSpatialData *Plot a SpatialData object*

Description

Initialize an empty ggplot for a SpatialData object. This function is typically combined with one or more calls to add specific plot layers.

Usage

```
plotSpatialData(x = NULL, ct = NULL)
```

Arguments

x	A SpatialData object.
ct	The name of a coordinate transformation to use for the plot.

Value

A ggplot object.

Examples

```
x <- file.path("extdata", "blobs.zarr")
x <- system.file(x, package="spatialdataR")
x <- readSpatialData(x, tables=FALSE)
ms <- lapply(seq(3), \(.) plotSpatialData(x) + plotImage(i=2, k=.))
patchwork::wrap_plots(ms)
```

scalebar *Add scalebar to plot*

Description

`scalebar` will get the axis unit and scale information from the closest preceding layer where this information is present. Therefore, the placement of `scalebar` in the sequence of layers is important.

Usage

```
scalebar(len = NULL, col = "red", lwd = 1, xrel = 0.05, yrel = 0.05)
```

Arguments

len	scalar numeric giving the length of the scalebar in physical coordinate space; the unit will be extracted from the data's Zarr specifications (see axes(x)).
col	string indicating the color to use for the scalebar.
lwd	scalar numeric indicating the linewidth to use for the scalebar.
xrel, yrel	scalar numeric indicating relative position of the scalebar.

Value

length-two list of ggplot2::annotate() layers corresponding to scalebar line (geom="segment") and unit label (geom="text")

Examples

```
zs <- file.path("extdata", "blobs.zarr")
zs <- system.file(zs, package="spatialdataR")
sd <- readSpatialData(zs, tables=FALSE)

# mock unit (data misses specification!)
md <- meta(image(sd, 2))
md$multiscales[[1]]$coordinateTransformations[[1]]$output$axes[[3]]$unit <- "micron"
sd$images[[2]]@meta <- md

plotSpatialData(sd) +
  plotImage(i=2) +
  scalebar(len=10)
```

Index

blobs, 2

getTable, *5*, *7*

plotImage, 2
plotLabel, 4
plotPoint, 5
plotShape, 6
plotSpatialData, 8

scalebar, 8
SpatialData, *3*, *5–7*