

Package: cytoviewer (via r-universe)

October 1, 2026

Version 1.13.3

Title An interactive multi-channel image viewer for R

Description This R package supports interactive visualization of multi-channel images and segmentation masks generated by imaging mass cytometry and other highly multiplexed imaging techniques using shiny. The cytoviewer interface is divided into image-level (Composite and Channels) and cell-level visualization (Masks). It allows users to overlay individual images with segmentation masks, integrates well with SingleCellExperiment and SpatialExperiment objects for metadata visualization and supports image downloads.

License GPL-3

Imports shiny, shinydashboard, utils, colourpicker, shinycssloaders, svgPanZoom, htmlwidgets, viridis, archive, grDevices, RColorBrewer, svglite, EBImage, methods, cytomapper, ggplot2, ggraph, tidygraph, rlang, SingleCellExperiment, SpatialExperiment, S4Vectors, SummarizedExperiment

Suggests BiocStyle, knitr, rmarkdown, markdown, testthat

biocViews ImmunoOncology, Software, SingleCell, OneChannel, TwoChannel, MultiChannel, Spatial, DataImport

VignetteBuilder knitr

URL <https://github.com/BodenmillerGroup/cytoviewer>

BugReports <https://github.com/BodenmillerGroup/cytoviewer/issues>

RoxygenNote 7.3.3

Encoding UTF-8

Config/pak/sysreqs cmake libfftw3-dev libfontconfig1-dev libfreetype6-dev libfribidi-dev libgdal-dev gdal-bin libgeos-dev libglpk-dev make libharfbuzz-dev libmagick++-dev gsfonts libarchive-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev zlib1g-dev

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

Date/Publication 2026-08-31 13:54:22 UTC

RemoteUrl <https://github.com/bioc/cytoviewer>

RemoteRef HEAD

RemoteSha 99da834f095ec1be0a33ec29b83fbc6381ecd6ac

Contents

cytoviewer	2
Index	5

cytoviewer	<i>cytoviewer - Shiny application to interactively browse multi-channel images</i>
------------	--

Description

This shiny R application allows users to interactively visualize multi-channel images, segmentation masks and spatial cell locations generated by imaging mass cytometry and other highly multiplexed imaging techniques. The cytoviewer interface is divided into image-level (Composite and Channels), cell-level (Masks) and point-level (Graph) visualization. It allows users to overlay individual images with segmentation masks, integrates well with `SingleCellExperiment` and `SpatialExperiment` objects for metadata visualization and supports image downloads.

Usage

```
cytoviewer(
  image = NULL,
  mask = NULL,
  object = NULL,
  cell_id = NULL,
  img_id = NULL,
  coords = NULL
)
```

Arguments

image	(optional) a <code>CytoImageList</code> object containing single or multi-channel <code>Image</code> objects.
mask	(optional) a <code>CytoImageList</code> containing single-channel <code>Image</code> objects.
object	(optional) a <code>SingleCellExperiment</code> or <code>SpatialExperiment</code> object.
cell_id	character specifying the <code>colData(object)</code> entry, in which the integer cell IDs are stored. These IDs should match the integer pixel values in the segmentation mask object (<code>mask</code>).
img_id	character specifying the <code>colData(object)</code> and <code>mcols(mask)</code> and/or <code>mcols(image)</code> entry, in which the image IDs are stored.
coords	character vector of length 2 specifying the names of the <code>colData</code> (for a <code>SingleCellExperiment</code> object) or the <code>spatialCoords</code> entries indicating the cells' x and y locations.

Value

A Shiny app object for interactive multi-channel image visualization and exploration

The input objects

The functionality of `cytoviewer` depends on which input objects are user-provided. Below we describe the four use cases in respect to input objects and functionality.

1. Usage of cytoviewer with images, masks and object

The full functionality of `cytoviewer` can be leveraged when image, mask and object are provided. This allows image-level visualization (Composite and Channels), cell-level visualization, point-level visualization, overlaying images with segmentation masks as well as metadata visualization.

2. Usage of cytoviewer with images only

If only image is specified, image-level visualization (Composite and Channels) is possible.

3. Usage of cytoviewer with images and masks

Image-level visualization (Composite and Channels), overlaying of images with masks and cell-level visualization is feasible when image and mask are provided.

4. Usage of cytoviewer with masks and object

If mask and object are specified, cell-level visualization, point-level visualization as well as meta-data visualization is possible.

Author(s)

Lasse Meyer (<lasse.meyer@dqbm.uzh.ch>)

See Also

[plotPixels](#) for the function underlying image-level visualization

[plotCells](#) for the function underlying cell-level visualization

[plotSpatial](#) for the function underlying point-level visualization

[cytomapperShiny](#) for a shiny application that visualizes gated cells on images

Examples

```
# Load example datasets from cytomapper
library(cytomapper, quietly = TRUE)
data("pancreasImages")
data("pancreasMasks")
data("pancreasSCE")

# 1. Use cytoviewer with images, masks and object
app <- cytoviewer(image = pancreasImages,
                  mask = pancreasMasks,
                  object = pancreasSCE,
                  img_id = "ImageNb",
                  cell_id = "CellNb",
                  coords = c("Pos_X", "Pos_Y"))
```

```
if (interactive()) {
  shiny::runApp(app, launch.browser = TRUE)
}

## Other input variations (see "The input objects" section):

# 2. Use cytoviewer with images
app_1 <- cytoviewer(image = pancreasImages)
if (interactive()) {
  shiny::runApp(app_1, launch.browser = TRUE)
}

# 3. Use cytoviewer with images and masks
app_2 <- cytoviewer(image = pancreasImages,
                    mask = pancreasMasks,
                    img_id = "ImageNb")
if (interactive()) {
  shiny::runApp(app_2, launch.browser = TRUE)
}

# 4. Use cytoviewer with masks and object
app_3 <- cytoviewer(mask = pancreasMasks,
                    object = pancreasSCE,
                    img_id = "ImageNb",
                    cell_id = "CellNb",
                    coords = c("Pos_X", "Pos_Y"))
if (interactive()) {
  shiny::runApp(app_3, launch.browser = TRUE)
}
```

Index

cytomapperShiny, [3](#)

cytoviewer, [2](#)

plotCells, [3](#)

plotPixels, [3](#)

plotSpatial, [3](#)