

Package: cytoviewer (via r-universe)

May 29, 2026

Version 1.12.0

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, viridis, archive, grDevices, RColorBrewer, svglite, EBImage, methods, cytomapper, SingleCellExperiment, 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 7.2.3

Encoding UTF-8

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

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

Date/Publication 2026-04-28 13:00:22 UTC

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

RemoteRef RELEASE_3_23

RemoteSha 53d83b6cca674cbc4445481434e0c4e03206d0f2

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 and segmentation masks generated by imaging mass cytometry and other highly multiplexed imaging techniques. 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.

Usage

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

Arguments

<code>image</code>	(optional) a <code>CytoImageList</code> object containing single or multi-channel <code>Image</code> objects.
<code>mask</code>	(optional) a <code>CytoImageList</code> containing single-channel <code>Image</code> objects.
<code>object</code>	(optional) a <code>SingleCellExperiment</code> or <code>SpatialExperiment</code> object.
<code>cell_id</code>	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>).
<code>img_id</code>	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.

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, 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 as well as metadata 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

[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")

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")
if (interactive()) {
  shiny::runApp(app_3, launch.browser = TRUE)
}
```

Index

cytomapperShiny, 3
cytoviewer, 2

plotCells, 3
plotPixels, 3