Package: CySA (via r-universe)

June 21, 2026

Title Interactive Cluster Selector for Cytometry Data

Version 0.99.7

Author Bernd Jagla [aut, cre]
(<https://orcid.org/0000-0002-7696-0484>)

Maintainer Bernd Jagla <bernd.jagla@pasteur.fr>

Description Provides an interactive Shiny application for selecting and visualizing clusters from flow cytometry data stored in SingleCellExperiment objects.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

biocViews SingleCell, FlowCytometry, Clustering, Visualization, ShinyApps

URL https://github.com/baj12/CySA

BugReports https://github.com/baj12/CySA/issues

Depends R (>= 4.5.0)

Imports shiny, shinydashboard, shinydashboardPlus, shinyjs, shinyjqui, htmltools, SingleCellExperiment, S4Vectors, SummarizedExperiment, Matrix, RColorBrewer, CATALYST, dplyr, tidyr, tibble, data.table, stringr, purrr, reshape2, rlang, raster, ggplot2, plotly, viridis, dendextend, ComplexHeatmap, cowplot, Rtsne, umap, DT, collapsibleTree, withr

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev libgdal-dev gdal-bin libgeos-dev libglpk-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libxml2-dev libssl-dev perl libproj-dev python3 libsqlite3-dev zlib1g-dev

Repository https://biocstaging.r-universe.dev

Date/Publication 2026-06-21 09:06:56 UTC

RemoteUrl https://github.com/BiocStaging/CySA

RemoteRef HEAD

RemoteSha 4e0520d0254e615a5d8adc3ae8562f0f2e6b4b81

Contents

`clusterSelector` *Cluster Selector Shiny Application*

Description

Creates an interactive Shiny dashboard for selecting and visualizing clusters from a SingleCellExperiment object.

Usage

```
clusterSelector(
  sce,
  sce_subsampled,
  outputList = list(),
  colTree = NULL,
  dList,
  dend,
  dendTable,
  clusterPatientTable,
  somCodesName = "SOM_codes",
  nPlots = 6,
  somRasterData,
  somRasterObj,
  env = environment()
)
```

Arguments

sce	A SingleCellExperiment containing the full dataset.
sce_subsampled	A subsampled SingleCellExperiment for performance-sensitive plots.
outputList	A named list of cluster groupings.
colTree	Optional collapsible tree object.
dList	List of marker pairs for 2D plots.
dend	Dendrogram object.
dendTable	Data frame for dendrogram navigation.
clusterPatientTable	Table of sample by cluster counts.
somCodesName	Name of the SOM codes metadata slot.
nPlots	Number of 2D SOM plots to display.
somRasterData	Data frame for SOM raster visualization.
somRasterObj	Raster object for SOM visualization.
env	Environment used to store mutable state (legacy argument).

Value

A shinyApp object. Launch the app with `shiny::runApp(app)`.

Examples

```
sce <- CySA_example_sce()
prepped <- prepClusterSelectorData(sce, total_cells_to_sample = 100)
som_codes <- S4Vectors::metadata(sce)$SOM_codes
dend <- stats::as.dendrogram(stats::hclust(stats::dist(som_codes)))
dendTable <- data.frame(
  id = seq_len(nrow(som_codes)),
  label = rownames(som_codes),
  stringsAsFactors = FALSE
)
clusterPatientTable <- table(
  sample_id = sce$sample_id,
  cluster_id = sce$cluster_id
)
markers <- S4Vectors::metadata(sce)$map$colsUsed
somRasterData <- data.frame(
  x = rep(seq_len(5), length.out = nrow(som_codes)),
  y = rep(seq_len(2), each = nrow(som_codes) / 2),
  id = seq_len(nrow(som_codes))
)
for (m in markers) {
  somRasterData[[m]] <- seq_len(nrow(som_codes)) / nrow(som_codes)
}
arr <- array(
  data = seq_len(10 * 10 * length(markers)),
  dim = c(10, 10, length(markers))
```

```
)
somRasterObj <- raster::brick(arr)
names(somRasterObj) <- markers

app <- clusterSelector(
  sce = prepped$sce,
  sce_subsampled = prepped$sce_subsampled,
  dList = prepped$dList,
  dend = dend,
  dendTable = dendTable,
  clusterPatientTable = clusterPatientTable,
  somRasterData = somRasterData,
  somRasterObj = somRasterObj
)
if (interactive()) {
  shiny::runApp(app)
}
```

`CySA_default_cluster_cols` *Default cluster color palette*

Description

Returns the default 20-color palette used by CySA for cluster visualizations.

Usage

```
CySA_default_cluster_cols()
```

Value

A character vector of hex colors.

Examples

```
CySA_default_cluster_cols()
```

CySA_example_sce *Minimal example SingleCellExperiment for CySA*

Description

Creates a small, deterministic `SingleCellExperiment` object that contains the metadata and column data expected by `CySA` functions.

Usage

```
CySA_example_sce(n_cells = 1000, n_nodes = 50, n_markers = 12)
```

Arguments

`n_cells`	Number of cells (columns).
`n_nodes`	Number of SOM nodes.
`n_markers`	Number of markers (rows).

Value

A `SingleCellExperiment` with `SOM_codes`, `SOM_stats`, `sample_id`, and `cluster_id`.

Examples

```
sce <- CySA_example_sce()
head(S4Vectors::metadata(sce)$SOM_codes)
```

plotScatterBJ *Plot Scatter for Bioconductor Experiment*

Description

This function generates a scatter plot for a Bioconductor Experiment object.

Usage

```
plotScatterBJ(
  x,
  chs,
  gate = NULL,
  color_by = NULL,
  facet_by = NULL,
  bins = 100,
  assay = "exprs",
```

```
  label = c("target", "channel", "both"),
  zeros = FALSE,
  k_pal = NULL,
  rowNames = NULL
)
```

Arguments

x	A Bioconductor Experiment object.
chs	A character vector specifying the variables to be plotted on the x and y axes.
gate	Optional gate object.
color_by	A character specifying the variable used for color coding points (optional).
facet_by	A character specifying the variable used for faceting the plot (optional).
bins	Number of bins for the 2D histogram (default is 100).
assay	The assay to use for plotting (default is "exprs").
label	Label for points on the plot ("target", "channel", or "both").
zeros	Logical, whether to exclude rows with all zero values (default is FALSE).
k_pal	Color palette for plotting.
rowNames	Optional character vector to specify row names for the plot.

Value

A ggplot2 scatter plot.

Examples

```
sce <- CySA_example_sce()
plotScatterBJ(sce, chs = c("marker1", "marker2"))
```

---

plotSOMScatter *Scatter plot*

Description

Bivariate scatter plots including visualization of (group-specific) gates, their boundaries and percentage of selected cells.

Usage

```
plotSOMScatter(
  x,
  chs,
  metaSlot = "SOM_codes",
  pointSize = "n",
  color_by = "n",
  bins = 100,
  assay = "exprs",
  statsSlot = "SOM_stats",
  label = c("target", "channel", "both"),
  zeros = FALSE,
  k_pal = NULL,
  xRN = NULL,
  xCN = NULL
)
```

Arguments

x	a SingleCellExperiment.
chs	character vector specifying which channels to plot.
metaSlot	name of the metadata slot containing SOM codes.
pointSize	column in SOM stats used to size points.
color_by	column to color points by.
bins	number of bins when coloring by density.
assay	name of the assay to use.
statsSlot	name of the metadata slot containing SOM stats.
label	how axis labels should be constructed.
zeros	logical specifying whether to include 0 values.
k_pal	optional cluster color palette.
xRN	optional row names.
xCN	optional channel names.

Value

a ggplot object.

Examples

```
sce <- CySA_example_sce()
plotSOMScatter(sce, chs = c("marker1", "marker2"))
```

prepClusterSelectorData *Prepare Data for the Cluster Selector Shiny App*

Description

Subsamples a SingleCellExperiment object and builds the inputs required by clusterSelector.

Usage

```
prepClusterSelectorData(
  sce,
  somFile = NULL,
  dList = NULL,
  total_cells_to_sample = 1e+05,
  somCodesName = "SOM_codes",
  assay = "exprs",
  seed = 123
)
```

Arguments

sce	A SingleCellExperiment containing the full dataset.
somFile	Path to the SOM object. Used to cache/restore the subsampled version.
dList	Optional list of marker pairs for 2D plots. If NULL, defaults to the first 12 row names of sce.
total_cells_to_sample	Number of cells to subsample in total.
somCodesName	Name of the SOM codes metadata slot.
assay	Name of the assay to use.
seed	Random seed for reproducibility.

Value

A list with sce, sce_subsampled, and dList.

Examples

```
sce <- CySA_example_sce(n_cells = 200, n_nodes = 10)
prepped <- prepClusterSelectorData(sce, total_cells_to_sample = 100)
names(prepped)
```

Index