Package: CySA (via r-universe)

November 25, 2026

Title Interactive Cluster Selector for Cytometry Data

Version 1.0.0

Description CySA provides an interactive Shiny-based tool for analyzing flow and mass cytometry data. It supports self-organizing map (SOM) based clustering, dimension reduction (t-SNE, UMAP, PCA), and statistical comparison of cell populations across sample groups. Users can select cell clusters interactively via 2D scatter plots, dendrograms, or SOM node grids, and explore marker expression patterns across the selected populations.

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, tidyselect, matrixStats, RColorBrewer, CATALYST, FlowSOM, dplyr, tidyr, tibble, data.table, stringr, purrr, reshape2, rlang, raster, ggplot2, plotly, viridis, dendextend, ComplexHeatmap, cowplot, Rtsne, umap, DT, collapsibleTree, withr

Suggests KernSmooth, knitr, jsonlite, rmarkdown, shinytest2, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

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

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:56:45 UTC

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

RemoteRef RELEASE_3_24

RemoteSha 2a0563f52b03a9939e88dbe5c6aa95bfec5c5a2d

Contents

clusterSelector 2
CySA_default_cluster_cols 4
CySA_example_sce 5
plotCytoScatter 5
plotSOMScatter 6
prepClusterSelector 8
prepClusterSelectorData 9

Index **11**

`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,
  fsom = NULL,
  env = environment(),
```

```
  verbose = FALSE
)
```

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. If NULL, the SOM grid dimensions are inferred from somRasterData.
fsom	Optional FlowSOM object. If provided, an interactive FlowSOM star plot is shown.
env	Environment used to store mutable state (legacy argument).
verbose	Logical indicating whether to show detailed runtime messages.

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)
```

plotCytoScatter *Plot Scatter for a SingleCellExperiment*

Description

Generates a 2-D scatter / density / colored point plot for a `SingleCellExperiment`.

Usage

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

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

Arguments

x	A SingleCellExperiment.
chs	Character vector of one or two markers/channels to plot. More than two channels will be melted and faceted.
gate	Currently unused.
color_by	Optional column name used to color points.
facet_by	Optional column name used for faceting.
bins	Number of bins for the 2-D histogram (default 100).
assay	Assay name to use (default "exprs").
label	Axis label style: "target", "channel" or "both".
zeros	Logical: if TRUE, remove rows where both plotted dimensions are zero.
k_pal	Color palette used when color_by is a clustering.
rownms	Optional character vector used instead of rownames(x) for matching chs.

Value

A ggplot object.

Examples

```
sce <- CySA_example_sce()
plotCytoScatter(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"))
```

prepClusterSelector *Prepare Cluster Selector Outputs*

Description

Wrapper that takes a post-SOM `SingleCellExperiment`, builds all supporting data structures required by `clusterSelector`, and saves a `sce.shiny.RData` bundle compatible with the original `prepCytoApp` output.

Usage

```
prepClusterSelector(
  currentRoot,
  sce = NULL,
  useColTree = FALSE,
  somCodesName = "SOM_codes",
  clusterColName = "cluster_id",
  total_cells_to_sample = 1e+05,
  dList = NULL,
  assay = "exprs",
  verbose = TRUE,
  force = FALSE,
  outputList = NULL,
  seed = 123,
  subsampledSCEFile = NULL
)
```

Arguments

`currentRoot`	Character string specifying the project root directory. Output files are written to `currentRoot/_data/`.
`sce`	A `SingleCellExperiment` with SOM clustering. If `NULL`, the function reads `currentRoot/_data/sce.RDS`.
`useColTree`	Logical indicating whether to create a collapsible tree.
`somCodesName`	Name of the SOM codes metadata slot.
`clusterColName`	Column name containing cluster IDs.
`total_cells_to_sample`	Number of cells to subsample for the Shiny app.
`dList`	Optional list of marker pairs for 2D plots.
`assay`	Name of the assay to use.
`verbose`	Logical indicating whether to print progress messages.
`force`	Logical indicating whether to force recomputation.
`outputList`	Optional output list to include in the saved bundle.
`seed`	Random seed for subsampling.

`subsampledSCEFile`
: Optional path to a previously subsampled SCE object (for example `sce.subsampled.RDS`). If provided and the file exists, it is reused instead of subsampling the full SCE. This avoids the memory cost of subsampling large datasets.

Details

The function performs the following steps:

1. Subsample the SCE using `prepClusterSelectorData()`, or reuse a previously subsampled SCE file if `subsampledSCEFile` is provided.
2. Update `cluster_id` to a factor with all SOM clusters as levels.
3. Build the cluster-by-patient table.
4. Recompute missing SOM code means.
5. Build SOM summary stats and cluster-level stats.
6. Build SOM raster data.
7. Build hierarchical clustering dendrogram and table.
8. Save `sce.shiny.RData`.

Value

The path `currentRoot`.

`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)

sce <- CySA_example_sce()
# prepClusterSelector(sce, ...)
```

Index

clusterSelector, 2, *8*, *9*
CySA_default_cluster_cols, 4
CySA_example_sce, 5

plotCytoScatter, 5
plotSOMScatter, 6
prepClusterSelector, 8
prepClusterSelectorData, 9

shinyApp, *3*
SingleCellExperiment, *3*, *5*, *7*, *10*