Package 'CySA'

November 12, 2026

Title Interactive Cluster Selector for Cytometry Data

Version 0.99.28

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

git_url https://git.bioconductor.org/packages/CySA

git_branch devel

git_last_commit fab459b

git_last_commit_date 2026-08-02

Repository Bioconductor 3.24

Date/Publication 2026-11-12

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

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

Contents

CySA-package . . . 3
.applySelectMode . . . 3
.buildBaseRasterGgplot . . . 4
.buildClusterSelectorServer . . . 4
.buildClusterSelectorSidebar . . . 5
.buildDendTable . . . 6
.buildDimRedReactives . . . 6
.buildFirstBodyRow . . . 7
.buildFlowSOMPieBox . . . 7
.buildFlowSOMPiePlot . . . 8
.buildFlowSOMStarsBox . . . 8
.buildSelectionObserver . . . 9
.buildSixStaticSOMBox . . . 9
.buildSOM2DPlotsBox . . . 10
.buildSOMClusterStats . . . 10
.buildSOMCodes . . . 11
.buildSOMDataObservers . . . 11
.buildSOMRasterBox . . . 12
.buildSOMRasterData . . . 13
.buildSOMRasterSelectPlot . . . 13
.buildSOMStats . . . 14
.buildStatsBox . . . 14
.buildUpSetBox . . . 15
.buildViolinBox . . . 15
.buildZoomObservers . . . 16
.computeRelativeCounts . . . 16
.computeScatterLimits . . . 17
.defaultDList . . . 17
.initializeOutputList . . . 18
.isDendrogramLeaf . . . 18
.nDendrogramLeaves . . . 19
.rebuildOutputList . . . 19
.updateOutputListInputs . . . 20
.validateClusterSelectorInputs . . . 20
.writeClusterSelectorPdf . . . 21
buildProjectionDf . . . 22
clusterSelector . . . 23
CySA_default_cluster_cols . . . 25
CySA_example_sce . . . 25
INTERNAL_buildClusterSelectorApp . . . 26
INTERNAL_registerClusterSelectorObservers . . . 27
INTERNAL_registerClusterSelectorOutputs . . . 28
plotCytoScatter . . . 31
plotSOMScatter . . . 32
prepClusterSelector . . . 33

prepClusterSelectorData 35

Index **36**

`CySA-package` *CySA: Interactive Cluster Selector for Cytometry Data*

Description

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

Author(s)

Maintainer: Bernd Jagla <bernd.jagla@pasteur.fr> (ORCID)

Authors:

- Bernd Jagla <bernd.jagla@pasteur.fr> (ORCID)

See Also

Useful links:

- https://github.com/baj12/CySA
- Report bugs at https://github.com/baj12/CySA/issues

`.applySelectMode` *Combine an existing selection with a newly picked set of IDs*

Description

Shared "selectMode" semantics used by every plotly click/select observer in this app. `"remove others"` narrows the current selection down to only the overlap with the new pick (it does NOT replace the selection with the new pick alone).

Usage

```
.applySelectMode(rs, picked, mode)
```

Arguments

`rs`	Integer vector of the currently selected IDs.
`picked`	Integer vector of newly picked IDs.
`mode`	One of `"remove others"`, `"add"`, `"remove"`, or anything else (falls through to `picked`). `NULL` is treated as `"view"` (falls through to `picked`) rather than erroring in `switch()`.

Value

Integer vector: the combined selection.

.buildBaseRasterGgplot

Build a Base SOM Raster Heatmap

Description

Creates a faceted ggplot2 raster heatmap from `somRasterData`. The resulting object is reused as the background for selected-node overlays.

Usage

```
.buildBaseRasterGgplot(somRasterData, colsUsed)
```

Arguments

`somRasterData`	Data frame with `x`, `y`, and marker columns.
`colsUsed`	Marker columns to include.

Value

A `ggplot` object or `NULL` if `somRasterData` is `NULL`.

.buildClusterSelectorServer

Build clusterSelector Server

Description

Creates the Shiny server function for the clusterSelector app.

Usage

```
.buildClusterSelectorServer(
  sce,
  sce_subsampled,
  outputList,
  colTree,
  dList,
  dend,
  dendTable,
  clusterPatientTable,
  somCodesName,
  nPlots,
  somRasterData,
  somRasterObj,
  fsom,
  env,
  verbose
)
```

Arguments

sce	Full SingleCellExperiment.
sce_subsampled	Subsampled SingleCellExperiment.
outputList	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.
fsom	Optional FlowSOM object.
env	Environment used to store mutable state (legacy argument).
verbose	Logical indicating whether to show detailed runtime messages.

Value

A shiny server function.

.buildClusterSelectorSidebar

Build clusterSelector Sidebar UI

Description

Constructs the left-hand control panel: selection mode, sample filter, scatter-plot auto-zoom, SOM 2D plot aesthetics, and group management.

Usage

```
.buildClusterSelectorSidebar(sce, outputList, nPlots)
```

Arguments

sce	Full SingleCellExperiment for sample-id choices.
outputList	Named list of cluster groupings.
nPlots	Kept for API compatibility; no longer controls dim-UI layout.

Details

Axis controls have moved to the SOM 2D plots body box to avoid the selectize-dropdown overflow that occurs inside the narrow sidebar.

Value

A shinydashboard::dashboardSidebar UI object.

.buildDendTable *Build Dendrogram Table*

Description

Creates a table describing the hierarchical clustering dendrogram of the SOM codes, optionally building a collapsible tree network widget.

Usage

```
.buildDendTable(sce, somCodesName = "SOM_codes", useColTree = FALSE)
```

Arguments

sce	A SingleCellExperiment with SOM codes in metadata(sce)[[somCodesName]].
useColTree	Logical indicating whether to build the collapsible tree.

Value

A list with dend, dendTable, and colTree.

.buildDimRedReactives *Build Dimension Reduction Reactives*

Description

Creates the t-SNE, UMAP, and PCA reactives used by the clusterSelector app.

Usage

```
.buildDimRedReactives(input, metaD, sce, somCodesName)
```

Arguments

input	Shiny input object.
metaD	metadata(sce) list.
sce	Full SingleCellExperiment.
somCodesName	Name of the SOM codes metadata slot.

Value

A named list of three reactives: tsne, umap, pca.

.buildFirstBodyRow *Build the First Dashboard Body Row*

Description

Top row of the dashboard: collapsible sample tree (optional), the main 2D scatter plot, and the interactive dendrogram. All three outputs support node/point selection.

Usage

```
.buildFirstBodyRow(colTree, nPlots)
```

Arguments

colTree	Optional collapsible tree object.
nPlots	Number of 2D SOM plots (kept for signature consistency).

Value

A `shiny::fluidRow` object.

.buildFlowSOMPieBox *Build FlowSOM Marker Pies UI*

Description

Displays each selected SOM node as a pie chart coloured by mean marker expression. Users control the grid layout and pie size.

Usage

```
.buildFlowSOMPieBox()
```

Value

A `shinydashboardPlus::box` UI object.

.buildFlowSOMPiePlot *Build a FlowSOM Marker Pie Plot*

Description

Build a FlowSOM Marker Pie Plot

Usage

```
.buildFlowSOMPiePlot(somCodes, rs, colsUsed, ncol = NULL, maxPies = 5L)
```

Arguments

somCodes	SOM codes matrix.
rs	Selected SOM node ids.
colsUsed	Marker names to include.
ncol	Number of columns passed to facet_wrap. NULL lets ggplot2 choose automatically.
maxPies	Maximum number of pies to plot. When more nodes are selected, only the first maxPies are shown. Default is 5.

Value

A ggplot object or NULL.

.buildFlowSOMStarsBox *Build FlowSOM Stars UI*

Description

Interactive FlowSOM star plot. Each node shows marker expression as a star, background colour shows the metaclustering. Click or lasso nodes to select clusters.

Usage

```
.buildFlowSOMStarsBox(fsom)
```

Arguments

fsom	Optional FlowSOM object. If NULL, returns NULL.

Value

A shinydashboardPlus::box UI object or NULL.

.buildSelectionObserver *Build Selection Observer*

Description

Registers a `plotly_selected` observer for a single source. When a selection event fires, it updates `rsUsed` according to the current selection mode.

Usage

```
.buildSelectionObserver(
  sourceId,
  input,
  rsUsed,
  rsUsed_d,
  inputSelect,
  verbose
)
```

Arguments

`sourceId`	Plotly source id.
`input`	Shiny input object.
`rsUsed`	Reactive value holding the currently selected nodes.
`rsUsed_d`	Debounced reactive read as the selection baseline before applying `inputSelect`.
`inputSelect`	Function that maps event data to new selected ids.
`verbose`	Logical indicating whether to print debug messages.

.buildSixStaticSOMBox *Build Six Static SOM Views Box*

Description

Grid of static ggplot SOM pair views. Users can add/remove marker pairs, change the grid layout, and adjust per-plot height.

Usage

```
.buildSixStaticSOMBox(markers, dList)
```

Arguments

`markers`	Character vector of all available markers (`rownames(sce)`).
`dList`	Initial list of marker pairs.

Value

A `shinydashboardPlus::box` UI object.

.buildSOM2DPlotsBox *Build SOM 2D Plot Box*

Description

Main SOM visualisation panel: interactive axis controls, the primary 2D SOM plot, dimension-reduction controls, and t-SNE/UMAP/PCA projection plots. All plot outputs are selection-enabled.

Usage

```
.buildSOM2DPlotsBox(nPlots, colsUsed, outputList, markers, dList)
```

Arguments

nPlots	Kept for API compatibility.
colsUsed	Character vector of SOM column names for dim-red selector.
outputList	Named list of cluster groupings.
markers	Character vector of all available markers (rownames(sce)).
dList	List of marker pairs used to populate the preset-pair picker.

Value

A shinydashboardPlus::box UI object.

.buildSOMClusterStats *Build SOM Cluster Stats*

Description

Computes per-cluster summary statistics for each marker relative to the SOM code, returning a long-format data frame suitable for downstream plotting.

Usage

```
.buildSOMClusterStats(
  sce,
  somCodesName = "SOM_codes",
  assay = "exprs",
  verbose = TRUE
)
```

Arguments

sce	A SingleCellExperiment with cluster_id in colData and SOM codes in metadata(sce)[[somCodesName]].
somCodesName	Name of the SOM codes metadata slot.
assay	Name of the assay to use.
verbose	Logical indicating whether to print progress messages.

Value

A long `data.frame` with columns `cluster`, `stat`, `variable`, and `value`.

.buildSOMCodes *Build SOM Codes From Cell-Level Expression Data*

Description

Computes per-cluster mean expression for all markers in `sce`. Markers already present in the existing SOM codes metadata slot are left unchanged; missing markers are calculated from the cell-level assay.

Usage

```
.buildSOMCodes(
  sce,
  somCodesName = "SOM_codes",
  markerList = NULL,
  assay = "exprs"
)
```

Arguments

`sce`	A `SingleCellExperiment` with `cluster_id` in `colData` and SOM codes stored in `metadata(sce)[[somCodesName]]`.
`somCodesName`	Name of the SOM codes metadata slot.
`markerList`	Optional character vector of markers to include. If `NULL`, all markers except `"label"` and `"TIME"` are used.
`assay`	Name of the assay to use.

Value

The updated `SingleCellExperiment` with refreshed SOM codes.

.buildSOMDataObservers *Build SOM Data Selection Observers*

Description

Loops over the 2D SOM plot outputs and registers selection observers.

Usage

```
.buildSOMDataObservers(
  nPlots,
  input,
  output,
  rsUsed,
  rsUsed_d,
  activePlot,
  inputSelect,
  verbose
)
```

Arguments

nPlots	Number of 2D SOM plots.
input	Shiny input object.
output	Shiny output object.
rsUsed	Reactive value holding the currently selected nodes.
activePlot	Reactive value tracking the active plot index.
inputSelect	Function that maps event data to new selected ids.
verbose	Logical indicating whether to print debug messages.

.buildSOMRasterBox *Build SOM Raster UI*

Description

Raster heatmap of SOM node marker expression plus a small interactive plot for selecting nodes from the SOM grid.

Usage

```
.buildSOMRasterBox()
```

Value

A shinydashboardPlus::box UI object.

.buildSOMRasterData *Build SOM Raster Data*

Description

Builds a data frame representing the SOM grid, including the SOM codes and per-node cell counts.

Usage

```
.buildSOMRasterData(sce, somCodesName = "SOM_codes")
```

Arguments

sce	A SingleCellExperiment with SOM codes in metadata(sce)[[somCodesName]] and optional SOM_stats.

Value

A data.frame.

.buildSOMRasterSelectPlot *Build a SOM Raster Selection Plot*

Description

Creates a plotly-friendly ggplot showing the full SOM grid with selected nodes highlighted.

Usage

```
.buildSOMRasterSelectPlot(somRasterData, rs)
```

Arguments

somRasterData	Data frame containing x and y grid columns.
rs	Selected SOM node ids.

Value

A ggplot object.

.buildSOMStats *Build SOM Summary Statistics*

Description

Computes per-cluster summary statistics (median, mean, third quartile, max, cell count) based on distances between SOM codes and their assigned cells.

Usage

```
.buildSOMStats(sce, somCodesName = "SOM_codes", assay = "exprs")
```

Arguments

sce	A SingleCellExperiment with cluster_id in colData and SOM codes in metadata(sce)[[somCodesName]].
somCodesName	Name of the SOM codes metadata slot.
assay	Name of the assay to use.

Value

A data.frame of per-cluster statistics.

.buildStatsBox *Build Stats UI*

Description

Statistics panel: compare cell counts, compute percentages, run sample-group t-tests, and display the results as tables and bar plots.

Usage

```
.buildStatsBox(metaD, somCodesName, outputList)
```

Arguments

metaD	metadata(sce) list used to build numeric/factor choices.
somCodesName	Name of the SOM codes metadata slot.
outputList	Named list of cluster groupings.

Value

A shinydashboardPlus::box UI object.

.buildUpSetBox *Build UpSet Plot UI*

Description

UpSet-style overlap visualisation of saved cluster groups. Users choose which groups to compare.

Usage

```
.buildUpSetBox(outputList)
```

Arguments

outputList	Named list of cluster groupings.

Value

A shinydashboardPlus::box UI object.

.buildViolinBox *Build Violin Plots UI*

Description

Marker expression violin plots for the saved cluster groups. Users choose which markers are shown.

Usage

```
.buildViolinBox(somCodes, colsUsed)
```

Arguments

somCodes	SOM codes matrix; column names become marker choices.
colsUsed	Default marker selection.

Value

A shinydashboardPlus::box UI object.

.buildZoomObservers *Build Zoom Observers for SOM Data Plots*

Description

Registers one `plotly_relayout` observer per 2D SOM plot.

Usage

```
.buildZoomObservers(nPlots, input, zoomFunc, verbose)
```

Arguments

`nPlots`	Number of 2D SOM plots.
`input`	Shiny input object.
`zoomFunc`	Function that applies zoom state to `dimSelection`.
`verbose`	Logical indicating whether to print debug messages.

.computeRelativeCounts *Compute Relative Cell Counts*

Description

Computes the percentage of selected cells relative to a reference column, another cluster group, or the total population.

Usage

```
.computeRelativeCounts(
  clusterPatientTable,
  rs,
  relativeToCol,
  expInfo,
  outputList
)
```

Arguments

`clusterPatientTable`	Table of sample by cluster counts.
`rs`	Selected cluster ids.
`relativeToCol`	Reference column or group name.
`expInfo`	Sample-level metadata data.frame.
`outputList`	Named list of cluster groupings.

Value

Numeric vector of percentages.

.computeScatterLimits *Compute Percentile Axis Limits for Scatter Plots*

Description

Compute Percentile Axis Limits for Scatter Plots

Usage

```
.computeScatterLimits(x, y, pctl)
```

Arguments

x	Numeric vector of x values.
y	Numeric vector of y values.
pctl	Quantile between 0.5 and 1.

Value

A named list with `xlim` and `ylim`.

.defaultDList *Default Marker Pairs for 2D Plots*

Description

Builds a default list of marker pairs from a `SingleCellExperiment` object.

Usage

```
.defaultDList(sce)
```

Arguments

sce	A `SingleCellExperiment`.

Value

A list of character pairs.

.initializeOutputList *Initialize Output List for the Cluster Selector*

Description

Ensures that the `outputList` always contains a `"Rest"` group covering all cluster levels not already assigned to a named group.

Usage

```
.initializeOutputList(outputList, clusterLevels)
```

Arguments

`outputList`	Named list of cluster groupings.
`clusterLevels`	Character or integer vector of all cluster levels.

Value

Updated named list with `"Rest"` populated and empty groups removed.

.isDendrogramLeaf *Test if an Object is a Dendrogram Leaf*

Description

Test if an Object is a Dendrogram Leaf

Usage

```
.isDendrogramLeaf(x)
```

Arguments

`x`	A dendrogram object.

Value

Logical.

.nDendrogramLeaves *Count Leaves in a Dendrogram*

Description

Count Leaves in a Dendrogram

Usage

```
.nDendrogramLeaves(x)
```

Arguments

x	A dendrogram object.

Value

Integer.

.rebuildOutputList *Update outputList After Cluster Assignment Change*

Description

Rebuilds `outputList` after a group is removed or renamed, ensuring the `"Rest"` group is always correct.

Usage

```
.rebuildOutputList(outputList, sceLevels, removed = NULL)
```

Arguments

outputList	Current named list of cluster groupings.
sceLevels	Character or integer vector of all cluster levels.
removed	Optional vector of cluster ids that should be excluded.

Value

Updated named list.

.updateOutputListInputs *Synchronize Select Input Choices with outputList*

Description

Updates the UI select inputs that depend on the names of `outputList`.

Usage

```
.updateOutputListInputs(session, input, outputList, metaD)
```

Arguments

`session`	Shiny session object.
`input`	Shiny input object.
`outputList`	Named list of cluster groupings.
`metaD`	`metadata(sce)` list.

.validateClusterSelectorInputs *Validate clusterSelector Inputs*

Description

Checks that all required inputs are present and correctly typed before the Shiny application is constructed.

Usage

```
.validateClusterSelectorInputs(
  sce,
  sce_subsampled,
  outputList,
  dList,
  dend,
  dendTable,
  clusterPatientTable,
  somRasterData,
  somCodesName,
  nPlots,
  fsom
)
```

Arguments

sce	Full SingleCellExperiment.
sce_subsampled	Subsampled SingleCellExperiment.
outputList	Named list of cluster groupings.
dList	List of marker pairs.
dend	Dendrogram object.
dendTable	Data frame for dendrogram navigation.
clusterPatientTable	Table of sample by cluster counts.
somRasterData	Data frame for SOM raster visualization.
somCodesName	Name of the SOM codes metadata slot.
nPlots	Number of 2D SOM plots.
fsom	Optional FlowSOM object.

Value

The validated metadata(sce) object invisibly.

.writeClusterSelectorPdf

Write All ClusterSelector Plots to a PDF

Description

Builds the full "Download Plots" PDF export: the dendrogram, count/percent bar charts, every preset SOM marker-pair panel (not just the currently active one), the SOM raster overlay, t-SNE/UMAP/PCA, the main scatter plot, FlowSOM pie/star plots when available, both violin plots, and the UpSet plot. Each entry in plotRegistry is wrapped in its own tryCatch() so one broken/unavailable plot is skipped with a warning rather than aborting the whole export (which previously surfaced to users as a corrupt/HTML "download").

Usage

```
.writeClusterSelectorPdf(
  file,
  rs,
  sce,
  sceRN,
  sceCN,
  metaD,
  somCodesName,
  dendPlotObj,
  dimPairs,
  getBasePlotFn,
  dlColorVar,
  dlSizeVar,
  pctl,
  somRasterXy,
```

```
  baseRasterGgplot,
  plotRegistry = list()
)
```

Arguments

file	Output PDF path (as required by downloadHandler's content argument).
rs	Integer vector of currently selected SOM node ids.
sce	Full SingleCellExperiment.
sceRN, sceCN	rownames(sce), colnames(sce).
metaD	metadata(sce) list.
somCodesName	Name of the SOM codes metadata slot.
dendPlotObj	A dendrogram object (already computed; this function only plots it).
dimPairs	List of marker-pair vectors, one per SOM panel shown on screen (i.e. dListRV()) – ALL pairs, not just the active plot.
getBasePlotFn	Function (x, y, colorVar, sizeVar) -> ggplot used to build each SOM panel's base plot, matching whatever the live static-grid renderer uses.
somRasterXy	Data frame of selected-node raster coordinates (or NULL).
baseRasterGgplot	Pre-built raster background ggplot (or NULL).
plotRegistry	Named list of zero-argument functions, each returning a plot object (ggplot, base plot, or a plotly/htmlwidget whose print() method renders sensibly) or NULL to skip. Errors thrown by any entry are caught and reported as a warning, not propagated.

---

buildProjectionDf *Build a Projection Data Frame for drawProjection*

Description

Constructs the per-SOM-node data frame that drawProjection expects: the two requested channel columns come first, followed by all SOM_stats columns. Reads coordinates directly from metadata(sce)[[somCodesName]] instead of parsing ggplot_build() output, which is both faster and independent of which aesthetics the base plot happens to expose.

Usage

```
buildProjectionDf(pp1, plotIdx, dimSelection, sce, somCodesName = "SOM_codes")
```

Arguments

pp1	Base ggplot object (kept in signature for call-site compatibility; no longer used internally).
plotIdx	Index into dimSelection.
dimSelection	List of per-plot dimension specs.
sce	Full SingleCellExperiment.
somCodesName	Name of the SOM codes metadata slot (default "SOM_codes").

Value

A data frame or NULL on failure.

---

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)
```

INTERNAL_buildClusterSelectorApp
Build clusterSelector UI and Server

Description

Build clusterSelector UI and Server

Usage

```
.buildClusterSelectorApp(
  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 named list with elements `ui` and `server`.

`INTERNAL_registerClusterSelectorObservers`
Register clusterSelector Observers

Description

Attaches all `shiny::observe*` calls used by the clusterSelector app. Output renderers live in `.registerClusterSelectorOutputs()`.

Usage

```
.registerClusterSelectorObservers(
  input,
  output,
  session,
  rv,
  rsUsed,
  rsUsed_d,
  activePlot,
  dListRV,
  dimSelection,
  triggerRedraw,
  selectedUpdate2,
  selectedUpdate,
  updatedoutputList,
  sample2PlotDb,
  dfPlot,
  groupsInput,
  inputClusterNumber,
  zoomFunc,
  sce,
  sce_subsampled,
  metaD,
  dend,
  dendTable,
  clusterPatientTable,
  somCodesName,
  colsUsed,
  nPlots,
  verbose,
  fsom = NULL,
  env
)
```

Arguments

`input`, `output`, `session`
: Shiny objects.

`rv`	Reactive values object holding `outputList`.
`rsUsed`, `rsUsed_d`	Reactive values for the current node selection.
`activePlot`	Reactive value tracking the active SOM plot.
`dListRV`	Reactive value holding the list of marker pairs.
`dimSelection`	Reactive value holding per-plot axis limits.
`triggerRedraw`	Reactive value used to force plot redraws.
`selectedUpdate2`, `selectedUpdate`	Reactive counters for invalidation.
`updatedoutputList`	Function that updates UI choices from outputList.
`sample2PlotDb`	Debounced reactive of samples to plot.
`dfPlot`	Reactive data frame for the scatter plot.
`groupsInput`	Debounced reactive of sample-group selections.
`inputClusterNumber`	Debounced reactive of typed cluster ids.
`zoomFunc`	Function handling plotly zoom relayout events.
`sce`	Full `SingleCellExperiment`.
`sce_subsampled`	Subsampled `SingleCellExperiment`.
`metaD`	`metadata(sce)` list.
`dend`	Dendrogram object.
`dendTable`	Data frame for dendrogram navigation.
`clusterPatientTable`	Table of sample by cluster counts.
`somCodesName`	Name of the SOM codes metadata slot.
`colsUsed`	Character vector of SOM columns used.
`nPlots`	Number of 2D SOM plots.
`verbose`	Logical indicating whether to print debug messages.

`INTERNAL_registerClusterSelectorOutputs`
Register clusterSelector Outputs

Description

Attaches all `output$*` renderers used by the clusterSelector app. Observer logic lives in `.registerClusterSelectorOb`

Usage

```
.registerClusterSelectorOutputs(
  input,
  output,
  session,
  rv,
  rsUsed,
  rsUsed_d,
  activePlot,
  dListRV,
  dimSelection,
  triggerRedraw,
  selectedUpdate2,
  selectedUpdate,
  updatedoutputList,
  colTree,
  somRasterData,
  somRasterObj,
  baseRasterGgplot,
  somRasterPlot,
  somBasePlot,
  somProjectionDf,
  tsnePlot,
  umapPlot,
  pcaPlot,
  scatterPlot,
  dendPlot,
  dendPlotlyData,
  countBarPlot,
  PercentBarPlot,
  vlnPlot,
  VlnPlot2,
  upSetPlot,
  flowSOMPiePlot,
  groupsInput,
  channelLimits,
  getBasePlot,
  fsom,
  sce,
  sce_subsampled,
  sce_subsampledRN,
  sceRN,
  sceCN,
  metaD,
  dend,
  dendTable,
  clusterPatientTable,
  somCodesName,
  colsUsed,
  rnSCE,
  nPlots,
  df,
```

```
  dfPlot,
  env,
  verbose
)
```

Arguments

`input`, `output`, `session`	Shiny objects.
`rv`	Reactive values object holding `outputList`.
`rsUsed`, `rsUsed_d`	Reactive values for the current node selection.
`activePlot`	Reactive value tracking the active SOM plot.
`dListRV`	Reactive value holding the list of marker pairs.
`dimSelection`	Reactive value holding per-plot axis limits.
`triggerRedraw`	Reactive value used to force plot redraws.
`selectedUpdate2`, `selectedUpdate`	Reactive counters for invalidation.
`updatedoutputList`	Function that updates UI choices from outputList.
`colTree`	Optional collapsible tree object.
`somRasterData`	Data frame for SOM raster visualization.
`somRasterObj`	Raster object for SOM visualization.
`baseRasterGgplot`	Pre-built ggplot raster background.
`somRasterPlot`	Reactive selected-node coordinates.
`somBasePlot`	Reactive base SOM scatter plot.
`somProjectionDf`	Reactive projection data frame for showGroups mode.
`tsnePlot`, `umapPlot`, `pcaPlot`	Reactive dimension-reduction plots.
`scatterPlot`	Reactive 2-D scatter plot.
`dendPlot`	Reactive dendrogram object.
`dendPlotlyData`	Reactive dendrogram plotting data.
`countBarPlot`, `PercentBarPlot`	Reactive stats bar plots.
`vlnPlot`, `VlnPlot2`	Reactive violin plots.
`upSetPlot`	Reactive UpSet plot.
`flowSOMPiePlot`	Reactive FlowSOM marker pie plot.
`groupsInput`	Debounced reactive of sample-group selections.
`channelLimits`	Named list of per-channel axis limits.
`getBasePlot`	Function that returns a cached base SOM ggplot.
`fsom`	Optional `FlowSOM` object.
`sce`	Full `SingleCellExperiment`.

sce_subsampled	Subsampled SingleCellExperiment.
sce_subsampledRN, sceRN, sceCN	Row/column name helpers.
metaD	metadata(sce) list.
dend	Dendrogram object.
dendTable	Data frame for dendrogram navigation.
clusterPatientTable	Table of sample by cluster counts.
somCodesName	Name of the SOM codes metadata slot.
colsUsed	Character vector of SOM columns used.
rnSCE	rownames(sce).
nPlots	Number of 2D SOM plots.
df	Static subsampled data frame.
dfPlot	Reactive subsampled data frame.
env	Environment used to store mutable state.
verbose	Logical indicating whether to print debug messages.

---

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

∗ **internal**
.applySelectMode, 3
.buildBaseRasterGgplot, 4
.buildClusterSelectorServer, 4
.buildClusterSelectorSidebar, 5
.buildDendTable, 6
.buildDimRedReactives, 6
.buildFirstBodyRow, 7
.buildFlowSOMPieBox, 7
.buildFlowSOMPiePlot, 8
.buildFlowSOMStarsBox, 8
.buildSOM2DPlotsBox, 10
.buildSOMClusterStats, 10
.buildSOMCodes, 11
.buildSOMDataObservers, 11
.buildSOMRasterBox, 12
.buildSOMRasterData, 13
.buildSOMRasterSelectPlot, 13
.buildSOMStats, 14
.buildSelectionObserver, 9
.buildSixStaticSOMBox, 9
.buildStatsBox, 14
.buildUpSetBox, 15
.buildViolinBox, 15
.buildZoomObservers, 16
.computeRelativeCounts, 16
.computeScatterLimits, 17
.defaultDList, 17
.initializeOutputList, 18
.isDendrogramLeaf, 18
.nDendrogramLeaves, 19
.rebuildOutputList, 19
.updateOutputListInputs, 20
.validateClusterSelectorInputs, 20
.writeClusterSelectorPdf, 21
buildProjectionDf, 22
CySA-package, 3
INTERNAL_buildClusterSelectorApp, 26
INTERNAL_registerClusterSelectorObservers, 27
INTERNAL_registerClusterSelectorOutputs, 28

.applySelectMode, 3
.buildBaseRasterGgplot, 4
.buildClusterSelectorApp (INTERNAL_buildClusterSelectorApp), 26
.buildClusterSelectorServer, 4
.buildClusterSelectorSidebar, 5
.buildDendTable, 6
.buildDimRedReactives, 6
.buildFirstBodyRow, 7
.buildFlowSOMPieBox, 7
.buildFlowSOMPiePlot, 8
.buildFlowSOMStarsBox, 8
.buildSOM2DPlotsBox, 10
.buildSOMClusterStats, 10
.buildSOMCodes, 11
.buildSOMDataObservers, 11
.buildSOMRasterBox, 12
.buildSOMRasterData, 13
.buildSOMRasterSelectPlot, 13
.buildSOMStats, 14
.buildSelectionObserver, 9
.buildSixStaticSOMBox, 9
.buildStatsBox, 14
.buildUpSetBox, 15
.buildViolinBox, 15
.buildZoomObservers, 16
.computeRelativeCounts, 16
.computeScatterLimits, 17
.defaultDList, 17
.initializeOutputList, 18
.isDendrogramLeaf, 18
.nDendrogramLeaves, 19
.rebuildOutputList, 19
.registerClusterSelectorObservers (INTERNAL_registerClusterSelectorObservers), 27
.registerClusterSelectorOutputs (INTERNAL_registerClusterSelectorOutputs), 28
.updateOutputListInputs, 20
.validateClusterSelectorInputs, 20
.writeClusterSelectorPdf, 21

buildProjectionDf, 22

clusterSelector, 23, *33*, *35*
CySA *(*CySA-package*)*, 3
CySA-package, 3
CySA_default_cluster_cols, 25
CySA_example_sce, 25

INTERNAL_buildClusterSelectorApp, 26
INTERNAL_registerClusterSelectorObservers, 27
INTERNAL_registerClusterSelectorOutputs, 28

plotCytoScatter, 31
plotSOMScatter, 32
prepClusterSelector, 33
prepClusterSelectorData, 35

shinyApp, *24*
SingleCellExperiment, *23*, *25, 26*, *33*, *35*