Package ‘ggspavis’

April 8, 2024

Version 1.8.1
Title Visualization functions for spatially resolved transcriptomics data
Description Visualization functions for spatially resolved transcriptomics datasets stored in SpatialExperiment format. Includes functions to create several types of plots for data from from spot-based (e.g. 10x Genomics Visium) and molecule-based (e.g. seqFISH) technological platforms.

URL https://github.com/lmweber/ggspavis
BugReports https://github.com/lmweber/ggspavis/issues
License MIT + file LICENSE
Encoding UTF-8
biocViews SingleCell, Transcriptomics, Spatial
Depends ggplot2
Imports SpatialExperiment, SingleCellExperiment, SummarizedExperiment, ggscale, grid, methods, stats
VignetteBuilder knitr
Suggests BiocStyle, rmarkdown, knitr, STexampleData, BumpyMatrix, scater, scran, uwot, testthat (>= 3.0.0)
RoxygenNote 7.2.1
Config/testthat/edition 3

git_url https://git.bioconductor.org/packages/ggspavis
git_branch RELEASE_3_18
git_last_commit c036da7
git_last_commit_date 2024-03-17
Repository Bioconductor 3.18
Date/Publication 2024-04-08
Author Lukas M. Weber [aut, cre] (<https://orcid.org/0000-0002-3282-1730>), Helena L. Crowell [aut] (<https://orcid.org/0000-0002-4801-1767>)
Maintainer Lukas M. Weber <lmwebereedu@gmail.com>
**Description**

Plotting functions for spatially resolved transcriptomics data.

**Usage**

```r
plotDimRed(
    spe,
    type = c("UMAP", "PCA"),
    x_axis = NULL,
    y_axis = NULL,
    annotate = NULL,
    palette = "libd_layer_colors",
    size = 0.3
)
```

**Arguments**

- `spe` (SpatialExperiment) Input data, assumed to be a SpatialExperiment object.
- `type` (character) Type of reduced dimension plot. Options are "UMAP" or "PCA". Default = "UMAP".
- `x_axis` (character) Name of column in reducedDim containing x-coordinates. Default = "UMAP1" or "PC1", depending on plot type.
- `y_axis` (character) Name of column in reducedDim containing y-coordinates. Default = "UMAP2" or "PC2", depending on plot type.
- `annotate` (character) Name of column in colData containing values to annotate spots with colors, e.g. cluster labels (discrete values) or total UMI counts (continuous values).
- `palette` (character) Color palette for annotation. Options for discrete labels are "libd_layer_colors", "Okabe-Ito", or a vector of color names or hex values. For continuous values, provide a vector of length 2 for the low and high range, e.g. c("gray90", "navy"). Default = "libd_layer_colors".
- `size` (numeric) Point size for geom_point(). Default = 0.3.
**plotDimRed**

**Details**

Function to plot spot-based spatially resolved transcriptomics data stored in a SpatialExperiment object.

This function generates a plot in reduced dimension coordinates (PCA or UMAP), along with annotation such as cluster labels or total UMI counts.

**Value**

Returns a ggplot object. Additional plot elements can be added as ggplot elements (e.g. title, labels, formatting, etc).

**Examples**

```r
library(STexampleData)
spe <- Visium_humanDLPFC()

# use small subset of data for this example
# for longer examples see our online book OSTA
spe <- spe[, colData(spe)$in_tissue == 1]
set.seed(100)

n <- 200
spe <- spe[, sample(seq_len(ncol(spe)), n)]

# calculate log-transformed normalized counts
library(scran)
set.seed(100)
qclus <- quickCluster(spe)
spe <- computeSumFactors(spe, cluster = qclus)
spe <- logNormCounts(spe)

# identify top highly variable genes (HVGs)
is_mito <- grepl("MT-|mt-", rowData(spe)$gene_name)
spe <- spe[!is_mito, ]
dec <- modelGeneVar(spe)
top_hvgs <- getTopHVGs(dec, prop = 0.1)

# run dimensionality reduction
library(scater)
set.seed(100)
spe <- runPCA(spe, subset_row = top_hvgs)
set.seed(100)
spe <- runUMAP(spe, dimred = "PCA")
colnames(reducedDim(spe, "UMAP")) <- paste0("UMAP", 1:2)

# generate plot
plotDimRed(spe, type = "UMAP", annotate = "ground_truth")
```
plotMolecules

Description

Plotting functions for spatially resolved transcriptomics data.

Usage

plotMolecules(
  spe,
  molecule = NULL,
  x Coord = NULL,
  y Coord = NULL,
  sample_id = "sample_id",
  palette = c("gray90", "navy"),
  size = 0.3
)

Arguments

  spe     (SpatialExperiment) Input data, assumed to be a SpatialExperiment object.
  molecule (character) Name of mRNA molecule to plot (assumed to match one of the row
             names of rowData).
  x_coord  (character) Name of column in spatialCoords containing x-coordinates of the
             cell centroids. Default = NULL, which selects the first column.
  y_coord  (character) Name of column in spatialCoords containing y-coordinates of the
             cell centroids. Default = NULL, which selects the second column.
  sample_id (character) Name of column in colData containing sample IDs. For datasets
                 with multiple samples, this is used to plot multiple panels (one per sample)
                 using faceting.
  palette  (character) Color palette, provided as a vector of length 2 for the low and high
             range. Default = c("gray90", "navy").
  size     (numeric) Point size for geom_point(). Default = 0.3.

Details

Function to plot molecule-based spatially resolved transcriptomics data stored in a SpatialExperiment
object.

This function generates a plot in spatial coordinates (e.g. x-y coordinates on a tissue slide), for a
selected molecule.

Value

Returns a ggplot object. Additional plot elements can be added as ggplot elements (e.g. title, labels,
formatting, etc).
`plotQC`  

**Examples**

```r
library(STexampleData)
spe <- seqFISH_mouseEmbryo()
plotMolecules(spe, molecule = "Sox2")
```

**Description**

Quality control (QC) plots for spatially resolved transcriptomics data.

**Usage**

```r
plotQC(
spe,  
type = c("bar", "scatter", "spots"),  
x_coord = NULL,  
y_coord = NULL,  
in_tissue = "in_tissue",  
metric_x = "cell_count",  
metric_y = "sum",  
discard = "discard",  
highlight_zeros = TRUE,  
threshold_x = NULL,  
threshold_y = NULL,  
trend = TRUE,  
marginal = TRUE,  
y_reverse = TRUE
)
```

**Arguments**

- **spe** *(SpatialExperiment)* Input data, assumed to be a SpatialExperiment object.
- **type** *(character)* Type of QC plot. Options are "bar", "scatter", and "spots". See details in description.
- **x_coord** *(character)* Name of column in spatialCoords containing x-coordinates. Default = NULL, which selects the first column. Used for spot-based plots.
- **y_coord** *(character)* Name of column in spatialCoords containing y-coordinates. Default = NULL, which selects the second column. Used for spot-based plots.
- **in_tissue** *(character)* Name of column in colData identifying spots over tissue, e.g. "in_tissue" for 10x Genomics Visium data. If this argument is provided, only spots over tissue will be shown. Alternatively, set to NULL to display all spots. Default = "in_tissue".
Function to generate plots for quality control (QC) purposes for spatially resolved transcriptomics data.

The following types of QC plots are available:
- Barplot (type = "bar") for a single QC metric, e.g. number of cells per spot. For number of cells per spot, the barplot highlights spots with zero cells. - Scatterplot (type = "scatter") comparing two QC metrics, e.g. number of detected features vs. number of cells per spot, with optional vertical and horizontal lines highlighting QC filtering thresholds. - Spots (type = "spots") i.e. spots in spatial (x-y) coordinates, highlighting discarded spots that do not meet filtering thresholds.

Value
Returns a ggplot object. Additional plot elements can be added as ggplot elements (e.g. title, labels, formatting, etc).

Examples
```
library(STexampleData)
spe <- Visium_humanDLPFC()
plotQC(spe, type = "bar", metric_x = "cell_count")
colData(spe)$sum <- colSums(counts(spe))
plotQC(spe, type = "scatter", metric_x = "cell_count", metric_y = "sum")
```
Description
Plotting functions for spatially resolved transcriptomics data.

Usage
plotSpots(
spe,
x_coord = NULL,
y_coord = NULL,
sample_id = "sample_id",
in_tissue = "in_tissue",
annotate = NULL,
palette = "libd_layer_colors",
y_reverse = TRUE,
size = 0.3
)

Arguments

spe (SpatialExperiment) Input data, assumed to be a SpatialExperiment object.
x_coord (character) Name of column in spatialCoords containing x-coordinates. Default = NULL, which selects the first column.
y_coord (character) Name of column in spatialCoords containing y-coordinates. Default = NULL, which selects the second column.
sample_id (character) Name of column in colData containing sample IDs. For datasets with multiple samples, this is used to plot multiple panels (one per sample) using facetting.
in_tissue (character) Name of column in colData identifying spots over tissue, e.g. "in_tissue" for 10x Genomics Visium data. If this argument is provided, only spots over tissue will be shown. Alternatively, set to NULL to display all spots. Default = "in_tissue".
annotate (character) Name of column in colData containing values to annotate spots with colors, e.g. cluster labels (discrete values) or total UMI counts (continuous values).
palette (character) Color palette for annotation. Options for discrete labels (e.g. cluster labels) are "libd_layer_colors", "Okabe-Ito", or a vector of color names or hex values. For continuous values (e.g. total UMI counts), provide a vector of length 2 for the low and high range, e.g. c("gray90", "navy"). Default = "libd_layer_colors".
y_reverse (logical) Whether to reverse y coordinates, which is often required for 10x Genomics Visium data. Default = TRUE.
size (numeric) Point size for geom_point(). Default = 0.3.
**Details**

Function to plot spot-based spatially resolved transcriptomics data stored in a SpatialExperiment object.

This function generates a plot in spatial coordinates (e.g. x-y coordinates on a tissue slide), along with annotation such as cluster labels or total UMI counts.

**Value**

Returns a ggplot object. Additional plot elements can be added as ggplot elements (e.g. title, labels, formatting, etc).

**Examples**

```r
library(STexampleData)
spe <- Visium_humanDLPFC()
plotSpots(spe, annotate = "ground_truth")
```

---

**plotVisium**

**Description**

Plots for spatially resolved transcriptomics data from the 10x Genomics Visium platform

**Usage**

```r
plotVisium(
  spe,
  spots = TRUE,
  fill = NULL,
  highlight = NULL,
  facets = "sample_id",
  image = TRUE,
  assay = "logcounts",
  trans = "identity",
  x_coord = NULL,
  y_coord = NULL,
  y_reverse = TRUE,
  sample_ids = NULL,
  image_ids = NULL,
  palette = NULL
)
```
plotVisium

Arguments

spe (SpatialExperiment) Input data object.
spots (logical) Whether to display spots (spatial barcodes) as points. Default = TRUE.
fill (character) Column in colData to use to fill points by color. If fill contains a numeric column (e.g. total UMI counts), a continuous color scale will be used. If fill contains a factor (e.g. cluster labels), a discrete color scale will be used. Default = NULL.
highlight (character) Column in colData to use to highlight points by outlining them. For example, in_tissue will highlight spots overlapping with tissue. Default = NULL.
facets (character) Column in colData to use to facet plots, i.e. show multiple panels of plots. Default = "sample_id". Set to NULL to disable.
image (logical) Whether to show histology image as background. Default = TRUE.
assay (character) Name of assay data to use when fill is in rownames(spe). Should be one of assayNames(spe).
trans Transformation to apply for continuous scales. Ignored unless fill is numeric, e.g. feature expression. (See ggplot2{continuous_scale} for valid options.)
x_coord (character) Column in spatialCoords containing x-coordinates. Default = NULL, which selects the first column.
y_coord (character) Column in spatialCoords containing y-coordinates. Default = NULL, which selects the second column.
y_reverse (logical) Whether to reverse y coordinates, which is often required for Visium data, depending on the orientation of the raw data. Default = TRUE.
sample_ids (character) Samples to show, if multiple samples are available. Default = NULL (show all samples).
image_ids (character) Images to show, if multiple images are available. Default = NULL (show all images).
palette (character) Color palette for points. Options for discrete labels are "libd_layer_colors", "Okabe-Ito", or a custom vector of hex color codes. Options for continuous values are "viridis", a single color name (e.g. "red", "navy", etc), or a vector of length two containing color names for each end of the scale. Default = "libd_layer_colors" for discrete data, and "viridis" for continuous data.

Details

Function to generate plots for spatially resolved transcriptomics datasets from the 10x Genomics Visium spatially platform.

This function generates a plot for spot-based spatially resolved transcriptomics data from the 10x Genomics Visium platform, with several options available to adjust the plot type and style.

Value

Returns a ggplot object. Additional plot elements can be added as ggplot elements (e.g. title, customized formatting, etc).
Author(s)
Helena L. Crowell with modifications by Lukas M. Weber

Examples

library(STexampleData)

spe <- Visium_mouseCoronal()

# color by x coordinate, highlight in-tissue spots
plotVisium(spe, fill = "pxl_col_in_fullres", highlight = "in_tissue")

# subset in-tissue spots
sub <- spe[, as.logical(colData(spe)$in_tissue)]

# color by feature counts, don’t include image
rownames(sub) <- make.names(rowData(sub)$gene_name)
plotVisium(sub, fill = "Gad2", assay = "counts")
Index

ggplot2, 9
plotDimRed, 2
plotMolecules, 4
plotQC, 5
plotSpots, 7
plotVisium, 8