

# Package ‘ggspavis’

July 9, 2025

**Version** 1.14.3

**Title** Visualization functions for spatial transcriptomics data

**Description** Visualization functions for spatial transcriptomics data. Includes functions to generate several types of plots, including spot plots, feature (molecule) plots, reduced dimension plots, spot-level quality control (QC) plots, and feature-level QC plots, for datasets from the 10x Genomics Visium and other technological platforms. Datasets are assumed to be in either SpatialExperiment or SingleCellExperiment format.

**URL** <https://github.com/lmweber/ggspavis>

**BugReports** <https://github.com/lmweber/ggspavis/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**biocViews** Spatial, SingleCell, Transcriptomics, GeneExpression, QualityControl, DimensionReduction

**Depends** ggplot2

**Imports** SpatialExperiment, SingleCellExperiment, SummarizedExperiment, ggside, grid, ggrepel, RColorBrewer, scales, grDevices, methods, stats

**VignetteBuilder** knitr

**Suggests** BiocStyle, rmarkdown, knitr, STexampleData, BumpyMatrix, scater, scran, uwot, testthat, patchwork

**RoxygenNote** 7.3.2

**git\_url** <https://git.bioconductor.org/packages/ggspavis>

**git\_branch** RELEASE\_3\_21

**git\_last\_commit** 2aef401

**git\_last\_commit\_date** 2025-06-04

**Repository** Bioconductor 3.21

**Date/Publication** 2025-07-09

**Author** Lukas M. Weber [aut, cre] (ORCID: <https://orcid.org/0000-0002-3282-1730>),  
 Helena L. Crowell [aut] (ORCID: <https://orcid.org/0000-0002-4801-1767>),  
 Yixing E. Dong [aut] (ORCID: <https://orcid.org/0009-0003-5115-5686>)  
**Maintainer** Lukas M. Weber <lmweberedu@gmail.com>

## Contents

|                         |           |
|-------------------------|-----------|
| plotDimRed . . . . .    | 2         |
| plotFeatureQC . . . . . | 4         |
| plotMolecules . . . . . | 6         |
| plotSpotQC . . . . .    | 7         |
| plotSpots . . . . .     | 9         |
| plotVisium . . . . .    | 12        |
| <b>Index</b>            | <b>15</b> |

---

|            |                   |
|------------|-------------------|
| plotDimRed | <i>plotDimRed</i> |
|------------|-------------------|

---

## Description

Plotting functions for spatial transcriptomics data.

## Usage

```
plotDimRed(
  spe,
  plot_type = c("UMAP", "PCA"),
  annotate = NULL,
  feature_names = NULL,
  assay_name = "counts",
  update_dimnames = TRUE,
  pal = NULL,
  point_size = 0.3,
  legend_point_size = 3,
  text_by = NULL,
  text_by_size = 5,
  text_by_color = "black"
)
```

## Arguments

|     |   |
|-----|---|
| spe | Input data, assumed to be a SpatialExperiment or SingleCellExperiment object. |
|-----|---|

|                   |  |
|-------------------|--|
| plot_type         | Type of reduced dimension plot. Possible options are "UMAP", "PCA", or any other set of reduced dimensions stored in the input object. Default = "UMAP".   |
| annotate          | Variable to show as annotations. This may be discrete or continuous. For a discrete variable (e.g. cluster labels), this should be the name of a column in colData containing a character vector or factor. For a continuous variable (e.g. a gene name), this should be an entry in feature_names. Default = NULL.                            |
| feature_names     | Name of column in rowData containing names of continuous features to plot (e.g. gene names). For example, set to feature_names = "gene_name" if gene names are stored in a column named "gene_name". This argument is used if annotate is a continuous variable. Default = NULL, in which case the row names of the input object will be used. |
| assay_name        | Name of assay in input object containing values to plot for a continuous variable. Default = "counts".   |
| update_dimnames   | Whether to update column names of reducedDims to default values for plotting. Default = TRUE.  |
| pal               | Color palette for annotations. Options for discrete values are "libd_layer_colors", "Okabe-Ito", or any 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").   |
| point_size        | Point size. Default = 0.3.   |
| legend_point_size | Legend point size for discrete annotations. Default = 3.   |
| text_by           | Column name of annotation labels to display over each cluster of points. This will usually be the same as annotate. Alternatively, another column may be used (e.g. with more readable classes or shorter strings). Only used for discrete annotate. Default = NULL.   |
| text_by_size      | Text size for annotation labels over each cluster. Default = 5.  |
| text_by_color     | Color name or hex code for annotation labels. Default = "black".   |

## Details

Function to create reduced dimension plot (e.g. PCA or UMAP) with additional optional annotations such as cluster labels, expression of a gene, or quality control metrics.

## Value

Returns a ggplot object, which may be further modified using ggplot functions.

## Author(s)

Lukas M. Weber and Yixing E. Dong

## Examples

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

```

# select spots over tissue
spe <- spe[, colData(spe)$in_tissue == 1]

# use small subset of data for this example
n <- 200
set.seed(123)
spe <- spe[, sample(seq_len(ncol(spe)), n)]

# calculate logcounts
library(scraper)
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(123)
spe <- runPCA(spe, subset_row = top_hvgs)
set.seed(123)
spe <- runUMAP(spe, dimred = "PCA")
colnames(reducedDim(spe, "UMAP")) <- paste0("UMAP", 1:2)

# generate plot
plotDimRed(spe, plot_type = "UMAP", annotate = "ground_truth")

```

---

plotFeatureQC

*plotFeatureQC*


---

## Description

Plotting functions for spatial transcriptomics data.

## Usage

```

plotFeatureQC(
  spe,
  plot_type = c("histogram", "violin"),
  x_metric = NULL,
  annotate = NULL,
  n_bins = 100,
  point_size = 0.1,
  scale_log1p = TRUE
)

```

**Arguments**

|             |  |
|-------------|--|
| spe         | Input data, assumed to be a SpatialExperiment or SingleCellExperiment object.  |
| plot_type   | Type of QC plot. Options are "histogram" and "violin". See Details for additional details.   |
| x_metric    | Name of column in rowData containing feature-level QC metric to plot on x-axis. Required for histograms and violin plots.  |
| annotate    | Name of column in rowData identifying selected features that do not meet QC filtering thresholds, which will be highlighted on a histogram or violin plot. Default = NULL. Optional argument used for histograms and violin plots. |
| n_bins      | Number of bins for histograms. Default = 100. Optional argument used for histograms.   |
| point_size  | Point size. Default = 0.1. Optional argument for violin plots.   |
| scale_log1p | Whether to log1p-scale axes. Default = TRUE.   |

**Details**

Function to create quality control (QC) plots for spatial transcriptomics data.

The following types of QC plots are available for feature-level QC (see [plotSpotQC](#) for spot-level or cell-level QC):

- Histogram (plot\_type = "histogram") for a single QC metric, e.g. total UMI counts across all spots per feature. The histogram can optionally highlight selected features, e.g. low abundance features.
- Violin (plot\_type = "violin") for a single QC metric, e.g. total UMI counts across all spots per feature. The violin plot can optionally highlight selected features, e.g. low abundance features.

**Value**

Returns a ggplot object, which may be further modified using ggplot functions.

**Author(s)**

Yixing E. Dong and Lukas M. Weber

**Examples**

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

rowData(spe)$feature_sum <- rowSums(counts(spe))
rowData(spe)$low_abundance <- rowSums(counts(spe) > 0) < 20

plotFeatureQC(spe, plot_type = "histogram",
               x_metric = "feature_sum", annotate = "low_abundance")
plotFeatureQC(spe, plot_type = "violin",
               x_metric = "feature_sum", annotate = "low_abundance")
```

---

plotMolecules

*plotMolecules*


---

## Description

Plotting functions for spatial transcriptomics data.

## Usage

```
plotMolecules(
  spe,
  molecule = NULL,
  x_coord = NULL,
  y_coord = NULL,
  sample_id = "sample_id",
  pal = c("gray90", "navy"),
  point_size = 0.3
)
```

## Arguments

|            |  |
|------------|--|
| spe        | (SpatialExperiment) Input data, assumed to be a SpatialExperiment object.  |
| molecule   | Name of mRNA molecule to plot (assumed to match one of the row names of rowData).  |
| x_coord    | Name of column in spatialCoords containing x coordinates. Default = NULL, which selects the first column of spatialCoords.   |
| y_coord    | Name of column in spatialCoords containing y coordinates. Default = NULL, which selects the second column of spatialCoords.  |
| sample_id  | Name of column in colData containing sample IDs. This argument is only required for datasets containing multiple samples (tissue sections). If provided, samples will be shown in multiple panels using facetting. Default = NULL. |
| pal        | Color palette, provided as a vector of length 2 for the low and high range. Default = c("gray90", "navy").   |
| point_size | Point size. Default = 0.3.   |

## Details

Function to create spot plot for molecule-based datasets, showing spatial locations in x-y coordinates with optional annotations such as expression of a gene.

## Value

Returns a ggplot object, which may be further modified using ggplot functions.

**Author(s)**

Lukas M. Weber

**Examples**

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

plotSpotQC

*plotSpotQC***Description**

Plotting functions for spatial transcriptomics data.

**Usage**

```
plotSpotQC(
  spe,
  plot_type = c("histogram", "scatter", "spot", "violin"),
  x_coord = NULL,
  y_coord = NULL,
  x_metric = NULL,
  y_metric = NULL,
  x_threshold = NULL,
  y_threshold = NULL,
  trend = TRUE,
  marginal = TRUE,
  annotate = NULL,
  in_tissue = NULL,
  legend_point_size = 3,
  n_bins = 100,
  point_size = 0.3,
  y_reverse = TRUE
)

plotQC(...)

plotObsQC(...)
```

**Arguments**

**spe** Input data, assumed to be a `SpatialExperiment` or `SingleCellExperiment` object.

|                   |  |
|-------------------|--|
| plot_type         | Type of QC plot. Options are "histogram", "scatter", "spot", and "violin". See Details for additional details.   |
| x_coord           | Name of column in spatialCoords (for a SpatialExperiment input object) or colData (for a SingleCellExperiment input object) containing x coordinates. Default = NULL (for a SpatialExperiment, the first column of spatialCoords will be selected in this case). Used for spot plots.  |
| y_coord           | Name of column in spatialCoords (for a SpatialExperiment input object) or colData (for a SingleCellExperiment input object) containing y coordinates. Default = NULL (for a SpatialExperiment, the second column of spatialCoords will be selected in this case). Used for spot plots. |
| x_metric          | Name of column in colData containing QC metric to plot on x-axis. Required for histograms, scatter plots, and violin plots.  |
| y_metric          | Name of column in colData containing QC metric to plot on y-axis. Required for histograms, scatter plots, and violin plots.  |
| x_threshold       | QC filtering threshold on x-axis metric to highlight with vertical line. Default = NULL. Optional argument used for scatter plots.   |
| y_threshold       | QC filtering threshold on y-axis metric to highlight with horizontal line. Default = NULL. Optional argument used for scatter plots.   |
| trend             | Whether to show smoothed trend line (loess). Default = TRUE. Optional argument used for scatter plots.   |
| marginal          | Whether to show marginal histograms. Default = TRUE. Optional argument used for scatter plots.   |
| annotate          | Name of column in colData identifying selected spots that do not meet QC filtering thresholds, which will be highlighted on a histogram, spot plot, or violin plot. Default = NULL. Optional argument used for histograms, spot plots, and violin plots.                               |
| in_tissue         | Name of column in colData identifying spots over tissue (e.g. "in_tissue" for 10x Genomics Visium datasets). If this argument is provided, only spots over tissue will be shown. Default = NULL. Optional argument used for spot plots.  |
| legend_point_size | Legend point size. Default = 3. Optional argument used for spot plots.   |
| n_bins            | Number of bins for histograms. Default = 100. Optional argument used for histograms.   |
| point_size        | Point size. Default = 0.3. Optional argument for scatter plots, spot plots, and violin plots. Suggested values: 0.5 for scatter plots, 0.3 for spot plots, 0.1 for violin plots.   |
| y_reverse         | Whether to reverse y coordinates. This is usually required for 10x Genomics Visium datasets when using the default coordinate values. Default = TRUE. Set to FALSE if not needed, e.g. for other platforms. Optional argument used for spot plots.                                     |
| ...               | Not used.  |



## Details

Function to create quality control (QC) plots for spatial transcriptomics data.

The following types of QC plots are available for spot-level or cell-level QC (see [plotFeatureQC](#) for feature-level QC):

- Histogram (`plot_type = "histogram"`) for a single QC metric, e.g. number of UMI counts per spot. For number of counts per spot, the histogram can optionally highlight selected spots, e.g. spots with low library size.
- Scatter plot (`plot_type = "scatter"`) comparing two QC metrics, e.g. number of detected features vs. number of cells per spot, with optional horizontal and vertical lines highlighting QC filtering thresholds.
- Spot plot (`plot_type = "spot"`) showing spots in spatial x-y coordinates, e.g. highlighting selected spots that do not meet filtering thresholds.
- Violin plot (`plot_type = "violin"`) for a single QC metric, e.g. number of UMI counts per spot. For number of counts per spot, the violin plot can optionally highlight selected spots, e.g. spots with low library size.

## Value

Returns a ggplot object, which may be further modified using ggplot functions.

## Author(s)

Lukas M. Weber and Yixing E. Dong

## Examples

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

colData(spe)$sum <- colSums(counts(spe))
colData(spe)$low_libsize <- colData(spe)$sum < 400

plotSpotQC(spe, plot_type = "histogram", x_metric = "sum", annotate = "low_libsize")
plotSpotQC(spe, plot_type = "scatter", x_metric = "sum", y_metric = "cell_count")
plotSpotQC(spe, plot_type = "spot", annotate = "low_libsize", in_tissue = "in_tissue")
plotSpotQC(spe, plot_type = "violin", x_metric = "sum", annotate = "low_libsize")
```

---

plotSpots

*plotSpots*

---

## Description

Plotting functions for spatial transcriptomics data.

**Usage**

```

plotSpots(
  spe,
  x_coord = NULL,
  y_coord = NULL,
  sample_id = NULL,
  in_tissue = "in_tissue",
  annotate = NULL,
  feature_names = NULL,
  assay_name = "counts",
  pal = NULL,
  point_size = 0.3,
  legend_position = "right",
  legend_point_size = 3,
  show_axes = FALSE,
  y_reverse = TRUE,
  text_by = NULL,
  text_by_size = 5,
  text_by_color = "black"
)

plotCoords(...)

```

**Arguments**

|                            |   |
|----------------------------|---|
| <code>spe</code>           | Input data, assumed to be a <code>SpatialExperiment</code> or <code>SingleCellExperiment</code> object.   |
| <code>x_coord</code>       | Name of column in <code>spatialCoords</code> (for a <code>SpatialExperiment</code> input object) or <code>colData</code> (for a <code>SingleCellExperiment</code> input object) containing x coordinates. Default = <code>NULL</code> (for a <code>SpatialExperiment</code> , the first column of <code>spatialCoords</code> will be selected in this case).  |
| <code>y_coord</code>       | Name of column in <code>spatialCoords</code> (for a <code>SpatialExperiment</code> input object) or <code>colData</code> (for a <code>SingleCellExperiment</code> input object) containing y coordinates. Default = <code>NULL</code> (for a <code>SpatialExperiment</code> , the second column of <code>spatialCoords</code> will be selected in this case). |
| <code>sample_id</code>     | Name of column in <code>colData</code> containing sample IDs. This argument is only required for datasets containing multiple samples (tissue sections). If provided, samples will be shown in multiple panels using facetting. Default = <code>NULL</code> .   |
| <code>in_tissue</code>     | Name of column in <code>colData</code> identifying spots over tissue (e.g. "in_tissue" for 10x Genomics Visium datasets). If this argument is provided, only spots over tissue will be shown. Default = "in_tissue". Set to <code>NULL</code> to display all spots.   |
| <code>annotate</code>      | Variable to show as annotations. This may be discrete or continuous. For a discrete variable (e.g. cluster labels), this should be the name of a column in <code>colData</code> containing a character vector or factor. For a continuous variable (e.g. a gene name), this should be an entry in <code>feature_names</code> . Default = <code>NULL</code> .  |
| <code>feature_names</code> | Name of column in <code>rowData</code> containing names of continuous features to plot (e.g. gene names). For example, set to <code>feature_names = "gene_name"</code> if gene  |

|                                |  |
|--------------------------------|--|
|                                | names are stored in a column named "gene_name". This argument is used if <code>annotate</code> is a continuous variable. Default = NULL, in which case the row names of the input object will be used.   |
| <code>assay_name</code>        | Name of assay in input object containing values to plot for a continuous variable. Default = "counts".   |
| <code>pal</code>               | Color palette for annotations. Options for discrete values are "libd_layer_colors", "Okabe-Ito", or any vector of color names or hex values. For continuous values, provide a vector of length 2 for the low and high range, e.g. <code>c("gray90", "navy")</code> .                             |
| <code>point_size</code>        | Point size. Default = 0.3.   |
| <code>legend_position</code>   | Legend position for discrete annotations. Options are "left", "right", "top", "bottom", and "none". Default = "right".   |
| <code>legend_point_size</code> | Legend point size for discrete annotations. Default = 3.   |
| <code>show_axes</code>         | Whether to show axis titles, text, and ticks. Default = FALSE.   |
| <code>y_reverse</code>         | Whether to reverse y coordinates. This is usually required for 10x Genomics Visium datasets when using the default coordinate values. Default = TRUE. Set to FALSE if not needed, e.g. for other platforms.  |
| <code>text_by</code>           | Column name of annotation labels to display over each cluster of points. This will usually be the same as <code>annotate</code> . Alternatively, another column may be used (e.g. with more readable classes or shorter strings). Only used for discrete <code>annotate</code> . Default = NULL. |
| <code>text_by_size</code>      | Text size for annotation labels over each cluster. Default = 5.  |
| <code>text_by_color</code>     | Color name or hex code for annotation labels. Default = "black".   |
| <code>...</code>               | Not used.  |

## Details

Function to create spot plot showing spatial locations in x-y coordinates with optional annotations such as cluster labels, expression of a gene, or quality control metrics.

## Value

Returns a ggplot object, which may be further modified using ggplot functions.

## Author(s)

Lukas M. Weber and Yixing E. Dong

## Examples

```
library(STexampleData)

# discrete annotations
spe <- Visium_humanDLPFC()
plotSpots(spe, annotate = "ground_truth")
```

```
# continuous annotations
spe <- Visium_mouseCoronal()
plotSpots(spe, annotate = "Gapdh", feature_names = "gene_name")
```

---

plotVisium

*plotVisium*


---

## Description

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

## Usage

```
plotVisium(
  spe,
  spots = TRUE,
  annotate = NULL,
  highlight = NULL,
  facets = "sample_id",
  image = TRUE,
  zoom = FALSE,
  show_axes = FALSE,
  assay = "counts",
  trans = "identity",
  point_size = 1,
  legend_position = "right",
  x_coord = NULL,
  y_coord = NULL,
  y_reverse = TRUE,
  sample_ids = NULL,
  image_ids = NULL,
  pal = NULL
)
```

## Arguments

|           |   |
|-----------|---|
| spe       | (SpatialExperiment) Input data object.  |
| spots     | (logical) Whether to display spots (spatial barcodes) as points. Default = TRUE.  |
| annotate  | (character) Column in colData to use to fill points by color. If annotate contains a numeric column (e.g. total UMI counts), a continuous color scale will be used. If annotate 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.   |
| zoom            | (logical) Whether to zoom to area of tissue containing spots. Default = FALSE  |
| show_axes       | (logical) Whether to show axes and coordinates. Default = FALSE  |
| assay           | (character) Name of assay data to use when annotate is in rownames(spe). Should be one of assayNames(spe).   |
| trans           | Transformation to apply for continuous scales. Ignored unless annotate is numeric, e.g. feature expression. (See <a href="#">ggplot2{continuous_scale}</a> for valid options.)   |
| point_size      | (numeric) Point size. Default = 1.   |
| legend_position | Legend position for annotations. Options are "left", "right", "top", "bottom", and "none". Default = "right".  |
| 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).  |
| pal             | (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 and Yixing E. Dong

**Examples**

```
library(STexampleData)

spe <- Visium_mouseCoronal()

# color by x coordinate, highlight in-tissue spots
plotVisium(spe, annotate = "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, annotate = "Gad2", assay = "counts")
```

# Index

ggplot2, [13](#)

plotCoords (plotSpots), [9](#)

plotDimRed, [2](#)

plotFeatureQC, [4](#), [9](#)

plotMolecules, [6](#)

plotObsQC (plotSpotQC), [7](#)

plotQC (plotSpotQC), [7](#)

plotSpotQC, [5](#), [7](#)

plotSpots, [9](#)

plotVisium, [12](#)