Package: SpotSweeper (via r-universe)

October 16, 2026

Title Spatially-aware quality control for spatial transcriptomics

Version 1.9.3

Date 2026-10-15

Description Spatially-aware quality control (QC) software for both spot-level and artifact-level QC in spot-based spatial transcriptomics, such as 10x Visium. These methods calculate local (nearest-neighbors) mean and variance of standard QC metrics (library size, unique genes, and mitochondrial percentage) to identify outlier spots and large technical artifacts.

License MIT + file LICENSE

URL https://github.com/MicTott/SpotSweeper

BugReports https://support.bioconductor.org/tag/SpotSweeper

biocViews Software, Spatial, Transcriptomics, QualityControl, GeneExpression,

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.4.0)

Imports SpatialExperiment, SummarizedExperiment, S4Vectors, BiocNeighbors, SingleCellExperiment, stats, escheR, MASS, ggplot2, grDevices, BiocParallel

Suggests knitr, BiocStyle, rmarkdown, scuttle, STexampleData, ggpubr, testthat (>= 3.0.0), SeuratObject, Seurat

Config/testthat/edition 3

VignetteBuilder knitr

LazyData False

News NEWS.md

Config/pak/sysreqs libmagick++-dev gsfonts libicu-dev libssl-dev zlib1g-dev

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

Date/Publication 2026-10-16 02:22:19 UTC

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

RemoteRef HEAD

RemoteSha 7e6ebf68c205cf0c102d8ed6b363752d5288d179

Contents

biased_spots . . . 2
DLPFC_artifact . . . 3
findArtifacts . . . 4
flagVisiumOutliers . . . 5
getMetadata . . . 6
getSpatialCoords . . . 6
localOutliers . . . 7
localVariance . . . 9
plotQCmetrics . . . 10
plotQCpdf . . . 12
setMetadata . . . 13
subsetSpatialObject . . . 14
validateMetadataColumns . . . 15

Index **16**

`biased_spots` *Biased Spots Data*

Description

The `biased_spots` dataset is a `data.frame` containing information about specific spatial spots identified as technical outliers in spatial transcriptomics experiments. Each entry represents a biased spot characterized by its spatial coordinates (row and column) and a unique barcode. This dataset is utilized by the `flagVisiumOutliers` function to flag and exclude these outlier spots from downstream analyses, thereby enhancing data quality and reliability.

Usage

```
data(biased_spots)
```

Format

A `data.frame` with the following columns:

row Integer. The row position of a biased spot within the spatial grid.

col Integer. The column position of a biased spot within the spatial grid.

barcode Character. A unique identifier corresponding to the spatial transcriptomics barcode of the biased spot.

Source

The `biased_spots.rds` file was generated in the analysis of local outliers. See https://github.com/boyiguo1/Manuscript-SpotSweeper/blob/main/code/03_Visium/figure_3.R for more details.

Examples

```
data(biased_spots)
```

`DLPFC_artifact` *human DLPFC dataset with a technical artifact (hangnail).*

Description

The `DLPFC_artifact` dataset is a `SpatialExperiment` object containing a single-sample subset of the human dorsolateral prefrontal cortex (DLPFC) dataset from Hukki-Myers et al. 2023. This particular sample ('Br2743_ant') is included to demonstrate the identification and removal of technical artifacts within spatial transcriptomics data. The dataset serves as an example for artifact detection using the 'SpotSweeper' workflow.

Usage

```
data(DLPFC_artifact)
```

Format

An SpatialExperiment object.

Source

spatialLIBD

References

Hukki-Myers et al. (2023) bioRxiv (bioRxiv)

Examples

```
data(DLPFC_artifact)
```

`findArtifacts` *Identify and annotate artifacts in spatial transcriptomics data*

Description

This function identifies and annotates potential artifacts in spatial transcriptomics data. Artifacts are detected based on local mito variance, and the results are added to the original SpatialExperiment (sce) object.

Usage

```
findArtifacts(
  spe,
  mito_percent = "expr_chrM_ratio",
  mito_sum = "expr_chrM",
  samples = "sample_id",
  n_order = 5,
  shape = "hexagonal",
  log = TRUE,
  name = "artifact",
  var_output = TRUE
)
```

Arguments

`spe`	A SingleCellExperiment object.
`mito_percent`	The column name representing the mitochondrial percent. Default is 'expr_chrM_ratio'.
`mito_sum`	The column name representing sum mitochondrial expression. Default is 'expr_chrM'.
`samples`	The column name representing sample IDs. Default is 'sample_id'.
`n_order`	The number of orders for local mito variance calculation. Default is 5.
`shape`	The shape of the neighborhood for local variance calculation. Can be either 'hexagonal' or 'square'. Default is 'hexagonal'.
`log`	Logical, indicating whether to log1p transform mito_percent. Default is TRUE.
`name`	Prefix for the local variance column names. Default is 'artifact'.
`var_output`	Logical, indicating whether to include local variances in the output. Default is TRUE.

Value

Returns the modified SingleCellExperiment object with artifact annotations.

See Also

`localVariance`

Examples

```
library(SpotSweeper)
library(SpatialExperiment)
library(escheR)

data(DLPFC_artifact)
spe <- DLPFC_artifact

# find artifacts
spe <- findArtifacts(spe,
    mito_percent = "expr_chrM_ratio",
    mito_sum = "expr_chrM",
    n_order = 2, # 5 recommended, using 2 for time
    shape = "hexagonal",
    name = "artifact"
)
```

`flagVisiumOutliers` *Flag systematic Visium outlier positions*

Description

`flagVisiumOutliers()` marks six Visium array positions that have shown systematically low library sizes across multiple datasets. It accepts either a `SpatialExperiment` or `Seurat` object and adds a logical `systematic_outliers` column to its metadata.

Usage

```
flagVisiumOutliers(spe, image_id = NULL)
```

Arguments

`spe`	A `SpatialExperiment` or `Seurat` object containing Visium array coordinates. `SpatialExperiment` inputs must contain `array_row` and `array_col` metadata columns. Seurat inputs may contain the same columns or expose them through a Visium image.
`image_id`	For Seurat inputs, the image containing Visium array coordinates. The first image is used when `NULL`.

Value

`spe` with a logical `systematic_outliers` metadata column.

Examples

```
spe <- STexampleData::Visium_humanDLPFC()
spe <- flagVisiumOutliers(spe)
spe <- spe[, !spe$systematic_outliers]
```

`getMetadata` *Get column metadata from a spatial object*

Description

Get column metadata from a spatial object

Usage

```
getMetadata(x)
```

Arguments

`x`	A `SpatialExperiment` or `Seurat` object.

Value

A `data.frame` with one row per spot, in object order.

Examples

```
spe <- STexampleData::Visium_humanDLPFC()
metadata <- getMetadata(spe)
head(metadata)
```

`getSpatialCoords` *Get spatial coordinates from a spatial object*

Description

Get spatial coordinates from a spatial object

Usage

```
getSpatialCoords(x, image_id = NULL)
```

Arguments

`x`	A `SpatialExperiment` or `Seurat` object.
`image_id`	For a Seurat object, an optional image name. If omitted, coordinates from all images are combined and returned in object order.

Value

A numeric matrix with spots as rows and spatial dimensions as columns.

Examples

```
spe <- STexampleData::Visium_humanDLPFC()
coordinates <- getSpatialCoords(spe)
head(coordinates)
```

`localOutliers` *Detect local outliers in spatial transcriptomics data*

Description

`localOutliers()` compares each spot's quality-control metric with the same metric in its nearest neighbors. It supports `SpatialExperiment` and `Seurat` objects and stores the resulting robust z-scores and outlier calls in the object's column metadata.

Usage

```
localOutliers(
  spe,
  metric = "detected",
  direction = "lower",
  n_neighbors = 36,
  samples = "sample_id",
  log = TRUE,
  cutoff = 3,
  workers = 1,
  coords = NULL
)
```

Arguments

`spe`	A `SpatialExperiment` or `Seurat` object.
`metric`	A single column name in the object's column metadata containing a numeric QC metric.
`direction`	Direction of outlier detection: `"higher"`, `"lower"`, or `"both"`.
`n_neighbors`	Number of nearest neighbors used as the local reference.
`samples`	A single metadata column name containing sample identifiers.
`log`	Whether to apply `log1p()` to `metric` before scoring.
`cutoff`	Non-negative absolute robust z-score cutoff.
`workers`	Number of workers for nearest-neighbor search. Uses serial execution for one worker, socket workers on Windows, and multicore workers on macOS/Linux. Per-spot scoring remains serial.
`coords`	Optional numeric coordinate matrix with spots as rows. When `NULL`, spatial coordinates are obtained from `spe`.

Details

The robust z-score for spot i is `qnorm(0.75) * (x_i - median(x_neighbors)) / MAD_unscaled(x_neighbors)`. The focal spot is not included when estimating the neighborhood median or median absolute deviation. A score of zero is returned when the neighborhood MAD is zero or non-finite.

Value

The input object with `<metric>_z` and `<metric>_outliers` metadata columns. When `log = TRUE`, `<metric>_log` is also added.

Examples

```
library(SpotSweeper)
library(SpatialExperiment)

spe <- STexampleData::Visium_humanDLPFC()
rownames(spe) <- rowData(spe)$gene_name
spe <- spe[, spe$in_tissue == 1]
spe <- spe[, !is.na(spe$ground_truth)]

is.mito <- grepl("^MT-", rownames(spe))
spe <- scuttle::addPerCellQCMetrics(
    spe,
    subsets = list(Mito = is.mito)
)

spe <- localOutliers(
    spe,
    metric = "sum",
    direction = "lower",
    log = TRUE
)

# Seurat objects use the same interface. Custom coordinates can also be
# supplied for either object type.
if (requireNamespace("SeuratObject", quietly = TRUE)) {
    counts <- matrix(
        seq_len(24),
        nrow = 4,
        dimnames = list(paste0("gene", 1:4), paste0("spot", 1:6))
    )
    seurat <- SeuratObject::CreateSeuratObject(counts)
    seurat$sample_id <- "sample"
    seurat$detected <- c(100, 10, 11, 9, 12, 10)
    example_coords <- cbind(x = seq_len(6), y = 0)
    seurat <- localOutliers(
        seurat,
        n_neighbors = 4,
        coords = example_coords
    )
}
```

`localVariance` *localVariance Function*

Description

This function calculates the local variance based on kNN.

Usage

```
localVariance(
  spe,
  n_neighbors = 36,
  metric = c("expr_chrM_ratio"),
  samples = "sample_id",
  log = FALSE,
  name = NULL,
  workers = 1
)
```

Arguments

`spe`	SpatialExperiment object with the following columns in colData: sample_id, sum_umi, sum_gene
`n_neighbors`	Number of nearest neighbors to use for variance calculation
`metric`	Metric to use for variance calculation
`samples`	Column in colData to use for sample ID
`log`	Whether to log1p transform the metric
`name`	Name of the new column to add to colData
`workers`	Number of workers for nearest-neighbor search. Uses serial execution for one worker, socket workers on Windows, and multicore workers on macOS/Linux. Subsequent variance calculations remain serial.

Value

SpatialExperiment object with metric variance added to colData

Examples

```
# for more details see extended example in vignettes
library(SpotSweeper)
library(SpatialExperiment)
library(escheR)

# load example data
spe <- STexampleData::Visium_humanDLPFC()
```

```
# change from gene id to gene names
rownames(spe) <- rowData(spe)$gene_name

# show column data before SpotSweepR
colnames(colData(spe))

# drop out-of-tissue spots
spe <- spe[, spe$in_tissue == 1]
spe <- spe[, !is.na(spe$ground_truth)]

# Identifying the mitochondrial transcripts in our SpatialExperiment.
is.mito <- rownames(spe)[grepl("^MT-", rownames(spe))]

# Calculating QC metric for each spot using scuttle
spe <- scuttle::addPerCellQCMetrics(spe, subsets = list(Mito = is.mito))
colnames(colData(spe))

spe <- localVariance(spe,
    metric = "subsets_Mito_percent",
    n_neighbors = 36,
    name = "local_mito_variance_k36",
    workers = 1
    )
```

`plotQCmetrics` *Plot QC metrics for a Single Sample in a SpatialExperiment object*

Description

This function generates a plot for a specified sample within a SpatialExperiment object, highlighting outliers based on a specified metric. The plot visualizes the metric of interest and indicates outliers with a distinct color.

Usage

```
plotQCmetrics(
  spe,
  sample_id = "sample_id",
  sample = unique(spe$sample_id)[1],
  metric = "detected",
  outliers = NULL,
  point_size = 2,
  colors = c("white", "black"),
  stroke = 1
)
```

Arguments

spe	A SpatialExperiment object containing the data to be plotted.
sample_id	A character string specifying the column name in colData(spe) that contains unique sample identifiers. Default is "sample_id".
sample	A character string or numeric value specifying the sample to be plotted. By default, it plots the first unique sample found in spe$sample_id.
metric	A character string specifying the metric to be visualized in the plot. This metric should be a column name in colData(spe).
outliers	A character string specifying the column name in colData(spe) that indicates whether a data point is considered an outlier. Default is NULL.
point_size	A numeric value specifying the size of the points in the plot. Default is 2.
colors	A character vector specifying the colors to be used for the gradient scale. If length is 2, the gradient will be a single color gradient.
stroke	A numeric value specifying the border thickness for outlier points. Default is 1.

Value

The function returns a plot object created by make_escheR and modified with additional layers for visualizing the specified metric and outliers. The plot is not explicitly printed by the function and should be printed by the caller.

Examples

```
library(SpotSweeper)
library(SpatialExperiment)
library(escheR)

# load example data
spe <- STexampleData::Visium_humanDLPFC()

# change from gene id to gene names
rownames(spe) <- rowData(spe)$gene_name

# drop out-of-tissue spots
spe <- spe[, spe$in_tissue == 1]
spe <- spe[, !is.na(spe$ground_truth)]

# Identifying the mitochondrial transcripts in our SpatialExperiment.
is.mito <- rownames(spe)[grepl("^MT-", rownames(spe))]

# Calculating QC metrics for each spot using scuttle
spe <- scuttle::addPerCellQCMetrics(spe, subsets = list(Mito = is.mito))
colnames(colData(spe))

# Identifying local outliers using SpotSweeper
spe <- localOutliers(spe,
                     metric = "sum",
                     direction = "lower",
```

```
                    log = TRUE
)

plotQCmetrics(spe, metric="sum", outliers="sum_outliers")
```

`plotQCpdf` *Plot Outlier Metrics to PDF*

Description

This function generates a PDF file containing plots for each sample in the SpatialExperiment object, highlighting outliers based on specified metrics. Each plot visualizes outlier metrics for a single sample, allowing for easy comparison and analysis across samples.

Usage

```
plotQCpdf(
  spe,
  sample_id = "sample_id",
  metric = "detected",
  outliers = "local_outliers",
  colors = c("white", "black"),
  stroke = 1,
  point_size = 2,
  width = 5,
  height = 5,
  fname
)
```

Arguments

`spe`	A SpatialExperiment object containing the data to be plotted.
`sample_id`	A character string specifying the column name in `colData(spe)` that contains unique sample identifiers. Default is 'sample_id'.
`metric`	A character string specifying the metric to be visualized in the plot. This metric should be a column name in `colData(spe)`.
`outliers`	A character string specifying the column name in `colData(spe)` that indicates whether a data point is considered an outlier. Default is local_outliers'.
`colors`	A character vector specifying the colors to be used for the gradient scale. If length is 2, the gradient will be a single color gradient
`stroke`	A numeric value specifying the border thickness for outlier points. Default is 1.
`point_size`	A numeric value specifying the size of the points in the plot. Default is 2.
`width`	A numeric value indicating the width of the plot. Default is 5.
`height`	A numeric value indicating the height of the plot. Default is 5.
`fname`	A character string specifying the path and name of the output PDF file.

Value

ggplot object if specified. Generates a plot otherwise.

Examples

```
library(SpotSweeper)
library(SpatialExperiment)
library(escheR)

# load example data
spe <- STexampleData::Visium_humanDLPFC()

tempFilePath <- file.path(tempdir(), "examplePlot.pdf")

# change from gene id to gene names
rownames(spe) <- rowData(spe)$gene_name

# drop out-of-tissue spots
spe <- spe[, spe$in_tissue == 1]
spe <- spe[, !is.na(spe$ground_truth)]

# Identifying the mitochondrial transcripts in our SpatialExperiment.
is.mito <- rownames(spe)[grepl("^MT-", rownames(spe))]

# Calculating QC metrics for each spot using scuttle
spe <- scuttle::addPerCellQCMetrics(spe, subsets = list(Mito = is.mito))
colnames(colData(spe))

# Identifying local outliers using SpotSweeper
spe <- localOutliers(spe,
                     metric = "sum",
                     direction = "lower",
                     log = TRUE
)

plotQCpdf(spe,
          metric="sum",
          outliers="sum_outliers",
          fname=tempFilePath)
```

`setMetadata` *Replace column metadata in a spatial object*

Description

Replace column metadata in a spatial object

Usage

```
setMetadata(x, metadata)
```

Arguments

x	A SpatialExperiment or Seurat object.
metadata	A data-frame-like object with one row per spot. Named rows are reordered to match the object before assignment.

Value

x with replaced column metadata.

Examples

```
spe <- STexampleData::Visium_humanDLPFC()
metadata <- getMetadata(spe)
metadata$example_column <- seq_len(nrow(metadata))
spe <- setMetadata(spe, metadata)
```

---

subsetSpatialObject *Subset a spatial object by spots*

Description

Subset a spatial object by spots

Usage

```
subsetSpatialObject(x, indices, column_name = NULL)
```

Arguments

x	A SpatialExperiment or Seurat object.
indices	Logical, integer, or character spot indices. If column_name is supplied, these are values to retain from that column.
column_name	Optional metadata column used for value-based subsetting.

Value

A subset of x containing the selected spots.

Examples

```
spe <- STexampleData::Visium_humanDLPFC()
spe_subset <- subsetSpatialObject(spe, seq_len(100))
```

validateMetadataColumns *Validate columns in spatial-object metadata*

Description

Validate columns in spatial-object metadata

Usage

```
validateMetadataColumns(x, required_columns)
```

Arguments

x	A SpatialExperiment or Seurat object.
required_columns	Character vector of required metadata column names.

Value

TRUE, invisibly. An error is raised if columns are missing.

Examples

```
spe <- STexampleData::Visium_humanDLPFC()
validateMetadataColumns(spe, c("sample_id", "in_tissue"))
```

Index

∗ **datasets**
biased_spots, 2
DLPFC_artifact, 3

biased_spots, 2

DLPFC_artifact, 3

findArtifacts, 4
flagVisiumOutliers, 5

getMetadata, 6
getSpatialCoords, 6

localOutliers, 7
localVariance, *4*, 9

plotQCmetrics, 10
plotQCpdf, 12

setMetadata, 13
subsetSpatialObject, 14

validateMetadataColumns, 15