Package ‘SpotSweeper’

May 30, 2024

Title  Spatiaely-aware quality control for spatial transcriptomics
Version  1.0.0
Date  2024-4-10
Description  Spatially-aware quality control (QC) software for both spot-level and artifact-level QC in spot-based spatial transcriptomics, such as 10x Visiun. These methods calculate local (nearest-neighbors) mean and variance of standard QC metrics (library size, unique genes, and mitochondrial percentage) to identify outliers spot and large technical artifacts. Scales linearly with the number of spots and is designed to be used with 'SpatialExperiment' objects.
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  7.3.1
Depends  R (>= 4.3.0)
Imports  SpatialExperiment, SummarizedExperiment, BiocNeighbors, SingleCellExperiment, stats, escheR, MASS, ggplot2, spatialEco, grDevices
Suggests  knitr, BiocStyle, rmarkdown, scuttle, STexampleData, ggpubr, testthat (>= 3.0.0)
Config/testthat/edition  3
VignetteBuilder  knitr
LazyData  False
git_url  https://git.bioconductor.org/packages/SpotSweeper
git_branch  RELEASE_3_19
git_last_commit  935948f
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

```r
findArtifacts(
  spe,
  mito_percent = "expr_chrM_ratio",
  mito_sum = "expr_chrM",
  samples = "sample_id",
  n_rings = 5,
  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_rings`: The number of rings for local mito variance calculation. Default is 5.
- `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

```r
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_rings = 2, # 5 recommended, using 1 for time
  name = "artifact"
)
```

---

**localOutliers**  
*localOutliers Function*

**Description**

This function detects local outliers in spatial transcriptomics data based on standard quality control metrics, such as library size, unique genes, and mitochondrial ratio. Local outliers are defined as spots with low/high quality metrics compared to their surrounding neighbors, based on a modified z-score statistic.

**Usage**

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

**Arguments**

- **spe**: SpatialExperiment object
- **metric**: colData QC metric to use for outlier detection
- **direction**: Direction of outlier detection (higher, lower, or both)
- **n_neighbors**: Number of nearest neighbors to use for outlier detection
- **samples**: Column name in colData to use for sample IDs
**localVariance**

**Function**

Description

This function calculates the local variance based on kNN.

**Usage**

```r
localVariance(
  spe,
  n_neighbors = 36,
  log = TRUE,
  cutoff = 3
)
```

**Value**

SpatialExperiment object with updated colData containing outputs.

**Examples**

```r
library(SpotSweeper)
library(SpatialExperiment)

# 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))

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

    metric = c("expr_chrM_ratio"),
    samples = "sample_id",
    log = FALSE,
    name = NULL
  )

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

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)[grep("^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,
```r
name = "local_mito_variance_k36"
)
plotQC(spe, metric="local_mito_variance_k36")
```

---

**plotQC**

*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**

```r
plotQC(
  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

```r
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
)

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

plotQCppdf

### 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(escherR)

# 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)
Index

* datasets
  DLPFC_artifact, 2

DLPFC_artifact, 2
findArtifacts, 3
localOutliers, 4
localVariance, 3, 5
plotQC, 7
plotQCpdf, 8