Package 'spatialDE'

November 12, 2026

Title R wrapper for SpatialDE

Version 1.18.0

Description SpatialDE is a method to find spatially variable genes (SVG) from spatial transcriptomics data. This package provides wrappers to use the Python SpatialDE library in R, using reticulate and basilisk.

License MIT + file LICENSE

Encoding UTF-8

LazyData FALSE

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

URL https://github.com/sales-lab/spatialDE, https://bioconductor.org/packages/spatialDE/

BugReports https://github.com/sales-lab/spatialDE/issues

Depends R (>= 4.3)

Imports reticulate, basilisk (>= 1.9.10), checkmate, stats, SpatialExperiment, methods, SummarizedExperiment, Matrix, ggplot2, ggrepel, scales, gridExtra

Suggests knitr, BiocStyle, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

biocViews Software, Transcriptomics

StagedInstall no

Config/testthat/edition 3

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

git_branch RELEASE_3_23

git_last_commit 6830500

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Davide Corso [aut] (ORCID: <https://orcid.org/0000-0001-8845-3693>),
Milan Malfait [aut] (ORCID: <https://orcid.org/0000-0001-9144-3701>),
Lambda Moses [aut] (ORCID: <https://orcid.org/0000-0002-7092-9427>),
Gabriele Sales [cre]

Maintainer Gabriele Sales <gabriele.sales@unipd.it>

Contents

.importPyModule 2
FSV_sig 3
MOB_sample_info 4
mockSVG 5
modelSearch 5
model_search 7
multiGenePlots 8
regress_out 10
Rep11_MOB_0 11
run 11
spatialDE 12
spatialPatterns 14
spatial_patterns 16
stabilize 17

Index **19**

`.importPyModule` *Import SpatialDE*

Description

This function loads the SpatialDE Python module and optionally monkey-patches it to remove tqdm calls.

Usage

```
.importPyModule(proc, patch_tqdm)
```

Arguments

`proc`	A process object generated by `basilisk::basiliskStart()`
`patch_tqdm`	If `TRUE` patch calls to tqdm.

Value

An R wrapper for the SpatialDE Python module.

FSV_sig *Plot Fraction Spatial Variance vs Q-value*

Description

Plot Fraction Spatial Variance vs Q-value

Usage

```
FSV_sig(
  results,
  ms_results = NULL,
  certain_only = FALSE,
  log_x = FALSE,
  do_label = TRUE,
  covariate_names = NULL
)
```

Arguments

results	results from SpatialDE.
ms_results	model selection results, should be a data frame with columns g for gene names and model for the model selected.
certain_only	only plot results with narrow 95% confidence interval.
log_x	Whether to display x axis in log scale.
do_label	display gene names for statistically significant genes, default TRUE.
covariate_names	names of covariates as a reference, default to NULL.

Value

A ggplot2 object.

Author(s)

Davide Corso, Milan Malfait, Lambda Moses

References

Svensson, V., Teichmann, S. & Stegle, O. SpatialDE: identification of spatially variable genes. Nat Methods 15, 343–346 (2018). https://doi.org/10.1038/nmeth.4636

SpatialDE 1.1.3: the version of the Python package used under the hood.

Examples

```
## Mock up a SpatialExperiment object wit 400 cells and 3 genes
set.seed(42)
spe <- mockSVG(size = 20, tot_genes = 3, de_genes = 1, return_SPE = TRUE)

## Run spatialDE with S4 integration
results <- spatialDE(spe)

## Run model search
msearch <- modelSearch(spe, de_results = results, qval_thresh = NULL,
  verbose = FALSE)

plot <- FSV_sig(results, msearch)
```

MOB_sample_info *Mouse Olfactory Bulb sample metadata*

Description

Coordinates and total counts for the samples from the Mouse Olfactory Bulb data generated by Stahl et al. (2016). This data was originally downloaded from https://github.com/Teichlab/SpatialDE/blob/master/Analysis/MouseOB/MOB_sample_info.csv.

Usage

```
data(MOB_sample_info)
```

Format

A `data.frame` with 262 rows and 3 variables as columns: the `x` and `y` coordinates and `total_counts` corresponding to each spot.

References

Ståhl, P. L. et al. (2016) 'Visualization and analysis of gene expression in tissue sections by spatial transcriptomics', *Science*, 353(6294), p. 78. doi: 10.1126/science.aaf2403.

mockSVG *Generate count matrix for spatially variable genes.*

Description

Generate count matrix for spatially variable genes.

Usage

```
mockSVG(size, tot_genes, de_genes, return_SPE = FALSE)
```

Arguments

size	An integer scalar. Cells will be spatially arranged on a size x size grid. Default: 10, corresponding to 100 cells.
tot_genes	An integer scalar. Total number of genes. Default: 1000.
de_genes	An integer scaler. The number of spatially variable genes. Default: 100.
return_SPE	A logical, whether to return result as a SpatialExperiment. Default: FALSE.

Value

If return_SPE = TRUE, returns a SpatialExperiment object.

If not, a list containing:

- coordinates: data.frame with x and y columns;
- counts: matrix with generated gene counts.

Examples

```
spe <- mockSVG(size = 20, tot_genes = 3, de_genes = 1, return_SPE = TRUE)
spe
```

modelSearch *Classify Spatially Variable Genes to interpretable fitting classes*

Description

Compare model fits with different models, using the **SpatialDE** Python package.

Usage

```
modelSearch(x, de_results, ...)

## S4 method for signature 'matrix'
modelSearch(x, de_results, coordinates, qval_thresh = 0.05, verbose = FALSE)

## S4 method for signature 'SpatialExperiment'
modelSearch(
  x,
  de_results,
  assay_type = "counts",
  qval_thresh = 0.05,
  verbose = FALSE
)
```

Arguments

x	A numeric `matrix` of counts where genes are rows and cells are columns. Alternatively, a SpatialExperiment object.
de_results	`data.frame` resulting from `run()` or `spatialDE()`.
...	For the generic, arguments to pass to specific methods.
coordinates	A `data.frame` with sample coordinates. Each row is a sample, the columns with coordinates should be named 'x' and 'y'. For the *SpatialExperiment* method, coordinates are taken from `spatialCoords(x)`.
qval_thresh	`numeric` scalar, specifying the q-value significance threshold to filter `de_results`. Only rows in `de_results` with `qval` < `qval_thresh` will be kept. To disable, set `qval_thresh = NULL`.
verbose	A `logical` controlling the display of a progress bar from the Python package.
assay_type	A `character` string specifying the assay from `x` to use as input. Defaults to `"counts"`.

Value

`data.frame` of model_search results.

Author(s)

Davide Corso, Milan Malfait, Lambda Moses

References

Svensson, V., Teichmann, S. & Stegle, O. SpatialDE: identification of spatially variable genes. Nat Methods 15, 343–346 (2018). https://doi.org/10.1038/nmeth.4636

SpatialDE 1.1.3: the version of the Python package used under the hood.

See Also

The individual steps performed by this function: `stabilize()`, `regress_out()` and `model_search()`.

Examples

```
## Mock up a SpatialExperiment object wit 400 cells and 3 genes
set.seed(42)
spe <- mockSVG(size = 20, tot_genes = 3, de_genes = 1, return_SPE = TRUE)

## Run spatialDE with S4 integration
de_results <- spatialDE(spe)

## Run model search
model_search <- modelSearch(spe, de_results = de_results,
    qval_thresh = NULL, verbose = FALSE
)
```

`model_search` *Compare model fits with different models*

Description

Classify DE genes to interpretable fitting classes.

Usage

```
model_search(x, coordinates, de_results, qval_thresh = 0.05, verbose = FALSE)
```

Arguments

`x`	`matrix`-like object of normalized counts. E.g. resulting from `regress_out()`.
`coordinates`	`data.frame` with sample coordinates. Each row is a sample, the columns with coordinates must be named 'x' and 'y'.
`de_results`	`data.frame` resulting from `run()`.
`qval_thresh`	`numeric` scalar, specifying the q-value significance threshold to filter `de_results`. Only rows in `de_results` with `qval` < `qval_thresh` will be kept. To disable, set `qval_thresh = NULL`.
`verbose`	`logical` controlling the display of the progress bar.

Value

`data.frame` of model_search results.

References

Svensson, V., Teichmann, S. & Stegle, O. SpatialDE: identification of spatially variable genes. Nat Methods 15, 343–346 (2018). https://doi.org/10.1038/nmeth.4636

Examples

```
## Mock up a SpatialExperiment object wit 400 cells and 3 genes
set.seed(42)
mock <- mockSVG(size = 20, tot_genes = 3, de_genes = 1)

stabilized <- stabilize(mock$counts)
sample_info <- mock$coordinates
sample_info$total_counts <- colSums(mock$counts)
regressed <- regress_out(counts = stabilized, sample_info = sample_info)

## Run SpatialDE
de_results <- run(regressed, coordinates = mock$coordinates)

## Run model search
ms_results <- model_search(
    x = regressed,
    coordinates = mock$coordinates,
    de_results = de_results,
    qval_thresh = NULL
)
```

`multiGenePlots` *Plot Spatial Patterns of Multiple Genes*

Description

Plot Spatial Patterns of Multiple Genes

Usage

```
multiGenePlots(x, ...)

## S4 method for signature 'matrix'
multiGenePlots(
  x,
  coordinates,
  genes_plot,
  viridis_option = "D",
  ncol = 2,
  point_size = 1,
  dark_theme = TRUE
)

## S4 method for signature 'SpatialExperiment'
multiGenePlots(
  x,
  assay_type = "counts",
```

```
  genes_plot,
  viridis_option = "D",
  ncol = 2,
  point_size = 1,
  dark_theme = TRUE
)
```

Arguments

x	A numeric `matrix` of stabilized counts (e.g. resulting from `stabilize()`) where genes are rows and cells are columns. Alternatively, a SpatialExperiment object.
...	For the generic, arguments to pass to specific methods.
coordinates	A `data.frame` with sample coordinates. Each row is a sample, the columns with coordinates should be named 'x' and 'y'. For the *SpatialExperiment* method, coordinates are taken from `spatialCoords(x)`.
genes_plot	character vector specifying which genes are to be plotted.
viridis_option	This function uses the `viridis` palette to color cells for gene expression. Four options are available: "magma" (or "A"), "inferno" (or "B"), "plasma" (or "C"), "viridis" (or "D", the default option) and "cividis" (or "E").
ncol	Number of columns to arrange the plots.
point_size	Point size of each plot.
dark_theme	Whether dark background should be used; this is helpful to highlight cells with high expression when using the `viridis` palette.
assay_type	A `character` string specifying the assay from `x` to use as input. Defaults to `"counts"`.

Value

This function draws a plot for each specified genes

Author(s)

Davide Corso, Milan Malfait, Lambda Moses

References

Svensson, V., Teichmann, S. & Stegle, O. SpatialDE: identification of spatially variable genes. Nat Methods 15, 343–346 (2018). https://doi.org/10.1038/nmeth.4636

SpatialDE 1.1.3: the version of the Python package used under the hood.

See Also

The individual steps performed by this function: `stabilize()`, `spatialDE()`.

For further analysis of the DE results: `model_search()` and `spatial_patterns()`.

Examples

```
## Mock up a SpatialExperiment object wit 400 cells and 3 genes
set.seed(42)
spe <- mockSVG(size = 20, tot_genes = 3, de_genes = 1, return_SPE = TRUE)

## Run spatialDE
results <- spatialDE(spe)

ordered_spe_results <- results[order(results$qval), ]
head(ordered_spe_results)

plots <- multiGenePlots(spe,
    assay_type = "counts",
    ordered_spe_results$g,
    point_size = 4,
    viridis_option = "D"
)
```

`regress_out` *Regress out library size effect*

Description

Regresses out the effect of library size. This function is a wrapper for `regress_out` from the NaiveDE Python package.

Usage

```
regress_out(counts, sample_info)
```

Arguments

`counts`	`matrix` of variance stabilized counts, e.g. resulting from `stabilize()`.
`sample_info`	`data.frame` with samples as rows and at least a column with `total_counts`.

Value

`matrix` of normalized counts.

Examples

```
## Mock up a SpatialExperiment object wit 400 cells and 3 genes
set.seed(42)
mock <- mockSVG(20, 3, 1)

stabilized <- stabilize(mock$counts)
sample_info <- mock$coordinates
sample_info$total_counts <- colSums(mock$counts)
```

```
regressed <- regress_out(counts = stabilized, sample_info = sample_info)
```

`Rep11_MOB_0` *Mouse Olfactory Bulb spatial gene expression data*

Description

Replicate 11 from the spatially dependent gene expression data from the mouse olfactory bulb generated by Stahl et al. (2016). This data was originally downloaded from https://github.com/Teichlab/SpatialDE/blob/master/Analysis/MouseOB/data/Rep11_MOB_0.csv.

Usage

```
data(Rep11_MOB_0)
```

Format

A matrix with 16218 genes as rows and 262 spots as columns.

References

Ståhl, P. L. et al. (2016) 'Visualization and analysis of gene expression in tissue sections by spatial transcriptomics', *Science*, 353(6294), p. 78. doi: 10.1126/science.aaf2403.

`run` *Perform SpatialDE test*

Description

Wraps the `run` function from the SpatialDE Python package.

Usage

```
run(x, coordinates, verbose = FALSE)
```

Arguments

`x`	`matrix`-like object of normalized counts. E.g. resulting from `regress_out()`.
`coordinates`	`data.frame` with sample coordinates. Each row is a sample, the columns with coordinates must be named 'x' and 'y'.
`verbose`	`logical` controlling the display of the progress bar.

Value

A `data.frame` with DE results where each row is a gene and columns contain relevant statistics.

The most important columns are:

- `g`: the name of the gene
- `pval`: the p-value for spatial differential expression
- `qval`: the q-value, indicating significance after correcting for multiple testing
- `l`: A parameter indicating the distance scale a gene changes expression over

References

Svensson, V., Teichmann, S. & Stegle, O. SpatialDE: identification of spatially variable genes. Nat Methods 15, 343–346 (2018). https://doi.org/10.1038/nmeth.4636

Examples

```
## Mock up a SpatialExperiment object wit 400 cells and 3 genes
set.seed(42)
mock <- mockSVG(size = 20, tot_genes = 3, de_genes = 1)

stabilized <- stabilize(mock$counts)
sample_info <- mock$coordinates
sample_info$total_counts <- colSums(mock$counts)
regressed <- regress_out(counts = stabilized, sample_info = sample_info)

## Run SpatialDE
de_results <- run(regressed, coordinates = mock$coordinates)
```

`spatialDE` *Find spatially variable genes with* **SpatialDE**

Description

Identify genes that significantly depend on spatial coordinates with the **SpatialDE** Python package.

Usage

```
spatialDE(x, ...)

## S4 method for signature 'matrix'
spatialDE(x, coordinates, verbose = FALSE)

## S4 method for signature 'SpatialExperiment'
spatialDE(x, assay_type = "counts", verbose = FALSE)
```

Arguments

x	A numeric matrix of counts where genes are rows and cells are columns. Alternatively, a SpatialExperiment object.
...	For the generic, arguments to pass to specific methods.
coordinates	A data.frame with sample coordinates. Each row is a sample, the columns with coordinates should be named 'x' and 'y'. For the *SpatialExperiment* method, coordinates are taken from spatialCoords(x).
verbose	A logical controlling the display of a progress bar from the Python package.
assay_type	A character string specifying the assay from x to use as input. Defaults to "counts".

Value

A data.frame with DE results where each row is a gene and columns contain relevant statistics.

The most important columns are:

- g: the name of the gene
- pval: the p-value for spatial differential expression
- qval: the q-value, indicating significance after correcting for multiple testing
- l: A parameter indicating the distance scale a gene changes expression over

Author(s)

Davide Corso, Milan Malfait, Lambda Moses

References

Svensson, V., Teichmann, S. & Stegle, O. SpatialDE: identification of spatially variable genes. Nat Methods 15, 343–346 (2018). https://doi.org/10.1038/nmeth.4636

SpatialDE 1.1.3: the version of the Python package used under the hood.

See Also

The individual steps performed by this function: stabilize(), regress_out() and run().

For further analysis of the DE results: model_search() and spatial_patterns().

Examples

```
## Mock up a SpatialExperiment object wit 400 cells and 3 genes
set.seed(42)
spe <- mockSVG(size = 20, tot_genes = 3, de_genes = 1, return_SPE = TRUE)

## Run spatialDE
de_results <- spatialDE(spe)

head(de_results)
```

spatialPatterns *Automatic expression histology in* **SpatialDE**

Description

Group spatially variable genes into spatial patterns using Automatic Expression Histology, using the **SpatialDE** Python package.

Usage

```
spatialPatterns(x, de_results, ...)

## S4 method for signature 'matrix'
spatialPatterns(
  x,
  de_results,
  coordinates,
  qval_thresh = 0.05,
  n_patterns,
  length,
  verbose = FALSE
)

## S4 method for signature 'SpatialExperiment'
spatialPatterns(
  x,
  de_results,
  qval_thresh = 0.05,
  n_patterns,
  length,
  assay_type = "counts",
  verbose = FALSE
)
```

Arguments

x	A numeric `matrix` of counts where genes are rows and cells are columns. Alternatively, a SpatialExperiment object.
de_results	`data.frame` resulting from `run()` or `spatialDE()`.
...	For the generic, arguments to pass to specific methods.
coordinates	A `data.frame` with sample coordinates. Each row is a sample, the columns with coordinates should be named 'x' and 'y'. For the *SpatialExperiment* method, coordinates are taken from `spatialCoords(x)`.
qval_thresh	`numeric` scalar, specifying the q-value significance threshold to filter `de_results`. Only rows in `de_results` with `qval` < `qval_thresh` will be kept. To disable, set `qval_thresh = NULL`.

n_patterns	integer The number of spatial patterns
length	numeric The characteristic length scale of the clusters
verbose	A logical controlling the display of a progress bar from the Python package.
assay_type	A character string specifying the assay from x to use as input. Defaults to "counts".

Value

A list of two data.frames (pattern_results, patterns):

- pattern_results: data.frame with pattern membership information for each gene.
- patterns the posterior mean underlying expression from genes in given spatial patterns.

Author(s)

Davide Corso, Milan Malfait, Lambda Moses

References

Svensson, V., Teichmann, S. & Stegle, O. SpatialDE: identification of spatially variable genes. Nat Methods 15, 343–346 (2018). https://doi.org/10.1038/nmeth.4636

SpatialDE 1.1.3: the version of the Python package used under the hood.

See Also

The individual steps performed by this function: stabilize(), regress_out() and spatial_patterns().

Examples

```
## Mock up a SpatialExperiment object wit 100 cells and 3 genes
set.seed(42)
spe <- mockSVG(size = 10, tot_genes = 3, de_genes = 1, return_SPE = TRUE)

## Run spatialDE
de_results <- spatialDE(spe)

spatial_patterns <- spatialPatterns(spe, de_results = de_results,
    qval_thresh = NULL, n_patterns = 4L, length = 1.5,
    verbose = FALSE
)

head(spatial_patterns$pattern_results)
head(spatial_patterns$patterns)
```

`spatial_patterns` *Group spatially variable genes into spatial patterns using automatic expression histology (AEH)*

Description

Group spatially variable genes into spatial patterns using automatic expression histology (AEH)

Usage

```
spatial_patterns(
  x,
  coordinates,
  de_results,
  qval_thresh = 0.05,
  n_patterns,
  length,
  verbose = FALSE
)
```

Arguments

`x`	`matrix`-like object of normalized counts. E.g. resulting from `regress_out()`.
`coordinates`	`data.frame` with sample coordinates. Each row is a sample, the columns with coordinates must be named 'x' and 'y'.
`de_results`	`data.frame` resulting from `run()`.
`qval_thresh`	`numeric` scalar, specifying the q-value significance threshold to filter `de_results`. Only rows in `de_results` with `qval` < `qval_thresh` will be kept. To disable, set `qval_thresh = NULL`.
`n_patterns`	`integer` The number of spatial patterns
`length`	`numeric` The characteristic length scale of the clusters
`verbose`	`logical` controlling the display of the progress bar.

Value

`list` of two dataframe (pattern_results, patterns): `pattern_results` dataframe with pattern membership information for each gene. `patterns` the posterior mean underlying expression fro genes in given spatial patterns.

References

Svensson, V., Teichmann, S. & Stegle, O. SpatialDE: identification of spatially variable genes. Nat Methods 15, 343–346 (2018). https://doi.org/10.1038/nmeth.4636

Examples

```
## Mock up a SpatialExperiment object wit 400 cells and 3 genes
set.seed(42)
mock <- mockSVG(size = 20, tot_genes = 3, de_genes = 1)

stabilized <- stabilize(mock$counts)
sample_info <- mock$coordinates
sample_info$total_counts <- colSums(mock$counts)
regressed <- regress_out(counts = stabilized, sample_info = sample_info)

## Run SpatialDE
de_results <- run(x = regressed, coordinates = mock$coordinates)

## Run Spatial_patterns
sp <- spatial_patterns(
    x = regressed,
    coordinates = mock$coordinates,
    de_results = de_results,
    qval_thresh = NULL,
    n_patterns = 5, length = 1.5
)

sp$pattern_results
sp$patterns
```

`stabilize` *Stabilize variance of counts*

Description

Stabilize variance of negative binomial data using Anscombe's approximation. This function is a wrapper for `stabilize` from the NaiveDE Python package.

Usage

```
stabilize(counts)
```

Arguments

`counts`	`matrix` with expression values for samples in columns and genes in rows.

Value

`matrix` of variance stabilized counts.

Examples

```
## Mock up a SpatialExperiment object wit 400 cells and 3 genes
set.seed(42)
mock <- mockSVG(20, 3, 1)

stabilized <- stabilize(mock$counts)
```

Index

∗ **datasets**
MOB_sample_info, 4
Rep11_MOB_0, 11
∗ **internal**
.importPyModule, 2
.importPyModule, 2

basilisk::basiliskStart(), *2*

FSV_sig, 3

MOB_sample_info, 4
mockSVG, 5
model_search, 7
model_search(), *7*, *9*, *13*
modelSearch, 5
modelSearch,matrix-method
*(*modelSearch*)*, 5
modelSearch,SpatialExperiment-method
*(*modelSearch*)*, 5
multiGenePlots, 8
multiGenePlots,matrix-method
*(*multiGenePlots*)*, 8
multiGenePlots,SpatialExperiment-method
*(*multiGenePlots*)*, 8

regress_out, 10
regress_out(), *7*, *11*, *13*, *15*, *16*
Rep11_MOB_0, 11
run, 11
run(), *6*, *7*, *13*, *14*, *16*

spatial_patterns, 16
spatial_patterns(), *9*, *13*, *15*
spatialDE, 12
spatialDE(), *6*, *9*, *14*
spatialDE,matrix-method *(*spatialDE*)*, 12
spatialDE,SpatialExperiment-method
*(*spatialDE*)*, 12
SpatialExperiment, *5*, *6*, *9*, *13*, *14*
spatialPatterns, 14
spatialPatterns,matrix-method
*(*spatialPatterns*)*, 14
spatialPatterns,SpatialExperiment-method
*(*spatialPatterns*)*, 14
stabilize, 17
stabilize(), *7*, *9*, *10*, *13*, *15*