Package: spatialLagGLM (via r-universe)

November 25, 2026

Title Spatial Lag Regression for Spatial Transcriptomics

Version 0.99.4

Description Tests each gene in spatial transcriptomics data for association between a spot's expression and the expression of its neighbours. For every gene, a quasi-Poisson generalized linear model is fitted with a library-size offset, user-defined design terms from the colData, and the spatial lag of the gene's own normalised counts as an additional covariate. The lag is a row-normalised, inverse-distance weighted mean over spots within a given radius, computed separately for each sample. The coefficient, p-value and adjusted p-value of the lag are stored in the rowData of the input SpatialExperiment. Weights are computed in C++ with RcppArmadillo, and genes are fitted in parallel with BiocParallel. The p-values are intended for exploratory use. Plotting functions and a Shiny explorer are included.

License MIT + file LICENSE

Encoding UTF-8

biocViews GeneExpression, Normalization, Regression, RNASeq, Software, Spatial, Transcriptomics

URL https://github.com/sahalpaladan/spatialLagGLM

BugReports https://github.com/sahalpaladan/spatialLagGLM/issues

Depends R (>= 4.5.0)

Imports BiocParallel (>= 1.36.0), Matrix, methods, Rcpp (>= 1.0.11), S4Vectors, SpatialExperiment (>= 1.12.0), stats, SummarizedExperiment (>= 1.32.0), ggplot2 (>= 3.4.0), utils

LinkingTo Rcpp, RcppArmadillo (>= 0.12.0)

Suggests BiocStyle (>= 2.30.0), knitr (>= 1.45), rmarkdown (>= 2.25), shiny (>= 1.8.0), STexampleData, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

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

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

Date/Publication 2026-11-25 04:29:10 UTC

RemoteUrl https://github.com/BiocStaging/spatialLagGLM

RemoteRef HEAD

RemoteSha cfe83d26a1b69127b14cdd4b92c2efe66903e94f

Contents

makeMockSpatial 2
plotLagVolcano 3
plotSpatialLag 3
runSpatialLagExplorer 4
spatialLagGLM 5

Index **7**

`makeMockSpatial` *Generate Mock Spatial Transcriptomics Data*

Description

Simulates a 30 x 30 grid of spots and 50 negative binomial genes. Spots within 7 units of the grid centre are labelled `"TumorCore"`, the rest `"Margin"`, and `Gene1` is up-regulated in the core.

Usage

```
makeMockSpatial()
```

Value

A `SpatialExperiment` object with synthetic counts and coordinates.

Examples

```
se <- makeMockSpatial()
```

plotLagVolcano *Volcano Plot of Spatial Lag Results*

Description

Volcano Plot of Spatial Lag Results

Usage

```
plotLagVolcano(se, alpha = 0.05, effect_cutoff = 0.5)
```

Arguments

`se`	A `SpatialExperiment` analysed with `spatialLagGLM`.
`alpha`	A single number in (0, 1]: adjusted p-value cutoff.
`effect_cutoff`	A single non-negative number: effect size cutoff.

Value

A ggplot2 object.

Examples

```
se <- makeMockSpatial()
analyzed <- spatialLagGLM(se, design = ~condition, radius = 2.5)
plotLagVolcano(analyzed)
```

plotSpatialLag *Plot a Gene on the Tissue Coordinates*

Description

Plots either the raw counts of a gene or its spatial lag (the weighted mean of its neighbours' normalised counts, i.e. the covariate used by `spatialLagGLM`). With several samples, each is drawn in its own panel.

Usage

```
plotSpatialLag(
  se,
  gene_id,
  metric = c("expression", "spatial_lag"),
  assay.type = "counts"
)
```

Arguments

se	A SpatialExperiment analysed with spatialLagGLM (needed only for "spatial_lag").
gene_id	A single gene name.
metric	"expression" for the raw counts, or "spatial_lag" for the spatial lag.
assay.type	A string naming the assay with raw counts.

Value

A ggplot2 object.

Examples

```
se <- makeMockSpatial()
analyzed <- spatialLagGLM(se, design = ~condition, radius = 2.5)
plotSpatialLag(analyzed, gene_id = "Gene1", metric = "expression")
plotSpatialLag(analyzed, gene_id = "Gene1", metric = "spatial_lag")
```

---

runSpatialLagExplorer *Launch Interactive Spatial Lag Explorer*

Description

Launch Interactive Spatial Lag Explorer

Usage

```
runSpatialLagExplorer(se)
```

Arguments

se	A SpatialExperiment analysed with spatialLagGLM.

Value

Called for its side effect of starting a Shiny app; returns nothing useful.

Examples

```
if (interactive()) {
    se <- makeMockSpatial()
    analyzed <- spatialLagGLM(se, design = ~condition, radius = 2.5)
    runSpatialLagExplorer(analyzed)
}
```

`spatialLagGLM` *Quasi-Poisson Regression with a Spatially Lagged Covariate*

Description

For every gene, fits a quasi-Poisson regression with a library-size offset, the terms of `design`, and the spatial lag of the gene's own size-factor-normalised counts as an extra covariate. The coefficient of the spatial lag, its p-value and its Benjamini-Hochberg adjusted p-value are stored in `rowData(se)`.

Usage

```
spatialLagGLM(
  se,
  design,
  radius = 2.5,
  assay.type = "counts",
  size.factors = NULL,
  BPPARAM = BiocParallel::bpparam()
)
```

Arguments

`se`	A `SpatialExperiment` object.
`design`	A formula using variables in `colData(se)`, e.g. `~ condition`.
`radius`	A positive number: the distance within which spots are neighbours, in the units of `spatialCoords(se)`. For several samples, all samples should share the same units.
`assay.type`	A string naming the assay with raw counts.
`size.factors`	`NULL` (default) to compute size factors from the library sizes of `assay.type`, or a positive numeric vector with one value per spot. Supply this when `se` has been subsetted to a few genes, so that normalisation still reflects all genes.
`BPPARAM`	A `BiocParallelParam` object.

Details

For spot i and a given gene, the model is

$$\log E[y_i] = \log(s_i) + X_i\beta + \gamma\, l_i,$$

where s_i is the size factor (library size divided by its geometric mean), X_i the row of the design matrix, and $l_i = \sum_j w_{ij} y_j / s_j$ the spatial lag. The weights w_{ij} are inverse distances between spots within `radius`, row-normalised, and are computed separately for each `sample_id`.

The null hypothesis tested is $\gamma = 0$: given the offset and the design terms, a spot's expression is not associated with the weighted mean of its neighbours' normalised expression.

Because the lag is built from the same gene's counts, this is not a spatial autoregressive model, and the quasi-Poisson p-values have not been shown to be calibrated for this test. Treat them as exploratory and, where it matters, check them against a permutation of the spot coordinates.

Genes whose fit fails or does not converge get NA results, and a warning reports how many.

The results are stored in `rowData(se)` as `spatial_effect_size` ($\hat{\gamma}$), `p_value` and `adjusted_p_value`. The radius, assay and size factors used are stored in `metadata(se)$spatialLagGLM`.

Value

`se` with `spatial_effect_size`, `p_value` and `adjusted_p_value` added to `rowData`.

Examples

```
se <- makeMockSpatial()
analyzed <- spatialLagGLM(se, design = ~condition, radius = 2.5)
head(SummarizedExperiment::rowData(analyzed))
```

Index

BiocParallelParam, *5*

makeMockSpatial, 2

plotLagVolcano, 3
plotSpatialLag, 3

runSpatialLagExplorer, 4

SpatialExperiment, *2*, *5*
spatialLagGLM, *3*, *4*, 5