

# Package ‘SpaNorm’

December 31, 2024

**Title** Spatially-aware normalisation for spatial transcriptomics data

**Version** 1.0.0

**Description** This package implements the spatially aware library size normalisation algorithm, SpaNorm. SpaNorm normalises out library size effects while retaining biology through the modelling of smooth functions for each effect. Normalisation is performed in a gene- and cell-/spot- specific manner, yielding library size adjusted data.

**License** GPL (>= 3)

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**biocViews** Software, GeneExpression, Transcriptomics, Spatial,  
CellBiology

**Depends** R (>= 4.4)

**Imports** edgeR, ggplot2, Matrix, matrixStats, methods, rlang, scran,  
SeuratObject, SingleCellExperiment, SpatialExperiment, stats,  
SummarizedExperiment, S4Vectors, utils

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown, prettydoc, pkgdown,  
covr, BiocStyle, Seurat, patchwork, ggforce, ggnewscale

**URL** <https://bhuvad.github.io/SpaNorm>

**BugReports** <https://github.com/bhuvad/SpaNorm/issues>

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**LazyData** false

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

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** e04495f

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.20

**Date/Publication** 2024-12-30

**Author** Dharmesh D. Bhuva [aut, cre] (<<https://orcid.org/0000-0002-6398-9157>>),  
Agus Salim [aut] (<<https://orcid.org/0000-0003-3999-7701>>),  
Ahmed Mohamed [aut] (<<https://orcid.org/0000-0001-6507-5300>>)

**Maintainer** Dharmesh D. Bhuva <dharmesh.bhuva@adelaide.edu.au>

## Contents

fastSizeFactors . . . . .	2
filterGenes . . . . .	3
HumanDLPFC . . . . .	3
plotSpatial . . . . .	4
SpaNorm . . . . .	5
SpaNormFit . . . . .	7
<b>Index</b>	<b>9</b>

---

fastSizeFactors	<i>Filter genes based on expression</i>
-----------------	---

---

## Description

This function computes the size factors using a fast but inaccurate approach. Size factors are computed using the direct estimate of library sizes (sum of all counts). Though fast, this approach does not cater for compositional biases in the data and therefore is less accurate than scran-based estimates.

## Usage

```
fastSizeFactors(spe)

## S4 method for signature 'SpatialExperiment'
fastSizeFactors(spe)
```

## Arguments

spe a SpatialExperiment, Seurat, or SpatialFeatureExperiment object containing count data.

## Value

a SpatialExperiment, Seurat, or SpatialFeatureExperiment, containing size factors in the 'sizeFactor' column of the column annotation.

## Examples

```
data(HumanDLPFC)
HumanDLPFC <- fastSizeFactors(HumanDLPFC)
head(HumanDLPFC$sizeFactor)
```

---

filterGenes	<i>Filter genes based on expression</i>
-------------	---

---

**Description**

This function removes genes that are very lowly expressed.

**Usage**

```
filterGenes(spe, prop = 0.1)

## S4 method for signature 'SpatialExperiment'
filterGenes(spe, prop = 0.1)

## S4 method for signature 'Seurat'
filterGenes(spe, prop = 0.1)
```

**Arguments**

spe	a SpatialExperiment, Seurat, or SpatialFeatureExperiment object containing count data.
prop	a numeric, indicating the proportion of loci/cells where the gene should be expressed (default is 0.1, i.e., genes should be expressed in at least 10% of the loci/cells).

**Value**

a logical vector encoding which genes should be kept for further analysis.

**Examples**

```
data(HumanDLPFC)
keep <- filterGenes(HumanDLPFC)
table(keep)
```

---

HumanDLPFC	<i>Human dorsolateral prefrontal cortex (DLPFC) visium sample</i>
------------	---

---

**Description**

Human DLPFC sample profiled using the 10x Visium platform. Lowly expressed genes were filtered out using the `filterGenes` function to retain genes expressed in at least 10% of samples. This version was obtained from the SpatialLIBD R/Bioconductor package.

**Usage**

```
data(HumanDLPFC)
```

**Format**

A SpatialExperiment containing region annotations in the colData column 'AnnotatedCluster'.

**Source**

SpatialLIBD R/Bioconductor package.

**References**

Maynard KR, Collado-Torres L, Weber LM, Uytingco C, Barry BK, Williams SR, Catallini JL, Tran MN, Besich Z, Tippani M, Chew J. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nature neuroscience*. 2021 Mar;24(3):425-36.

---

plotSpatial

*Plot spatial transcriptomic annotations per spot*

---

**Description**

Plot spatial transcriptomic annotations per spot

**Usage**

```
plotSpatial(
  spe,
  what = c("annotation", "expression", "reduceddim"),
  ...,
  assay = SummarizedExperiment::assayNames(spe),
  dimred = SingleCellExperiment::reducedDimNames(spe),
  img = FALSE,
  crop = FALSE,
  imgAlpha = 1,
  r1 = 1,
  circles = FALSE
)
```

**Arguments**

spe	a SpatialExperiment object.
what	a character, specifying what aspect should be plot, "annotation", "expression", or reduced dimension ("reduceddim").
...	additional aesthetic mappings or fixed parameters (e.g., shape = ".").
assay	a character or numeric, specifying the assay to plot (default is the first assay).
dimred	a character or numeric, specifying the reduced dimension to plot (default is the first reduced dimension).
img	a logical, indicating whether the tissue image (if present) should be plot (default = FALSE).
crop	a logical, indicating whether the image should be cropped to the spatial coordinates (default = FALSE).
imgAlpha	a numeric, specifying the alpha value for the image (default = 1).

`r1` a numeric, specifying the relative size of the text (default = 1).  
`circles` a logical, indicating whether the spots should be plotted as circles (default = FALSE). This can be slower for large datasets.

**Value**

a ggplot2 object

**Examples**

```
library(SpatialExperiment)
library(ggplot2)

# load data
data(HumanDLPFC)

# plot spatial region annotations
p1 <- plotSpatial(HumanDLPFC, colour = AnnotatedCluster)
p1

# change colour scale
p1 + scale_colour_brewer(palette = "Paired")

# plot spatial expression
plotSpatial(HumanDLPFC, what = "expression", colour = ENSG00000075624) +
  scale_colour_viridis_c(option = "F")

# plot logcounts
logcounts(HumanDLPFC) <- log2(counts(HumanDLPFC) + 1)
plotSpatial(HumanDLPFC, what = "expression", colour = ENSG00000075624, assay = "logcounts") +
  scale_colour_viridis_c(option = "F")

# change point shape
plotSpatial(HumanDLPFC, what = "expression", colour = ENSG00000075624, assay = "logcounts", shape = 18) +
  scale_colour_viridis_c(option = "F")
```

---

 SpaNorm

*Spatially-dependent normalisation for spatial transcriptomics data*


---

**Description**

Performs normalisation of spatial transcriptomics data using spatially-dependent spot- and gene-specific size factors.

**Usage**

```
SpaNorm(
  spe,
  sample.p = 0.25,
  gene.model = c("nb"),
  adj.method = c("auto", "logpac", "pearson", "medbio", "meanbio"),
  scale.factor = 1,
```

```

df.tps = 6,
lambda.a = 1e-04,
batch = NULL,
tol = 1e-04,
step.factor = 0.5,
maxit.nb = 50,
maxit.psi = 25,
maxn.psi = 500,
overwrite = FALSE,
verbose = TRUE,
...
)

## S4 method for signature 'SpatialExperiment'
SpaNorm(
  spe,
  sample.p = 0.25,
  gene.model = c("nb"),
  adj.method = c("auto", "logpac", "pearson", "medbio", "meanbio"),
  scale.factor = 1,
  df.tps = 6,
  lambda.a = 1e-04,
  batch = NULL,
  tol = 1e-04,
  step.factor = 0.5,
  maxit.nb = 50,
  maxit.psi = 25,
  maxn.psi = 500,
  overwrite = FALSE,
  verbose = TRUE,
  ...
)

```

### Arguments

<code>spe</code>	a <code>SpatialExperiment</code> or <code>Seurat</code> object, with the count data stored in 'counts' or 'data' assays respectively.
<code>sample.p</code>	a numeric, specifying the (maximum) proportion of cells/spots to sample for model fitting (default is 0.25).
<code>gene.model</code>	a character, specifying the model to use for gene/protein abundances (default 'nb'). This should be 'nb' for count based datasets.
<code>adj.method</code>	a character, specifying the method to use to adjust the data (default 'auto', see details)
<code>scale.factor</code>	a numeric, specifying the sample-specific scaling factor to scale the adjusted count.
<code>df.tps</code>	a numeric, specifying the degrees of freedom for the thin-plate spline (default is 6).
<code>lambda.a</code>	a numeric, specifying the smoothing parameter for regularizing regression coefficients (default is 0.0001). Actual <code>lambda.a</code> used is <code>lambda.a * ncol(spe)</code> .
<code>batch</code>	a vector or numeric matrix, specifying the batch design to regress out (default <code>NULL</code> , representing no batch effects). See details for more information on how

	to define this variable.
<code>tol</code>	a numeric, specifying the tolerance for convergence (default is 1e-4).
<code>step.factor</code>	a numeric, specifying the multiplicative factor to decrease IRLS step by when log-likelihood diverges (default is 0.5).
<code>maxit.nb</code>	a numeric, specifying the maximum number of IRLS iteration for estimating NB mean parameters for a given dispersion parameter (default is 50).
<code>maxit.psi</code>	a numeric, specifying the maximum number of IRLS iterations to estimate the dispersion parameter (default is 25).
<code>maxn.psi</code>	a numeric, specifying the maximum number of cells/spots to sample for dispersion estimation (default is 500).
<code>overwrite</code>	a logical, specifying whether to force recomputation and overwrite an existing fit (default FALSE). Note that if <code>df.tps</code> , <code>batch</code> , <code>lambda.a</code> , or <code>gene.model</code> are changed, the model is recomputed and overwritten.
<code>verbose</code>	a logical, specifying whether to show update messages (default TRUE).
<code>...</code>	other parameters fitting parameters.

### Details

SpaNorm works by first fitting a spatial regression model for library size to the data. Normalised data can then be computed using various adjustment approaches. When a negative binomial gene-model is used, the data can be adjusted using the following approaches: 'logpac', 'pearson', 'med-bio', and 'meanbio'.

Batch effects can be specified using the `batch` parameter. If this parameter is a vector, a design matrix will be created within the function using `model.matrix`. If a custom design is provided in the form of a numeric matrix, this should ideally be created using `model.matrix`. The batch matrix should be created with an intercept term. The SpaNorm function will automatically detect the intercept term and remove the relevant column. Alternatively, users can subset the model matrix to remove this column manually. Please note that the model formula should include the intercept term and that the intercept column should be subset out after.

### Value

a `SpatialExperiment` or `Seurat` object with the adjusted data stored in 'logcounts' or 'data', respectively.

### Examples

```
data(HumanDLPFC)

SpaNorm(HumanDLPFC, sample.p = 0.05, df.tps = 2, tol = 1e-2)
```

---

SpaNormFit

*An S4 class to store a SpaNorm model fit*

---

### Description

An S4 class to store a SpaNorm model fit

**Usage**

```
## S4 method for signature 'SpaNormFit'
x$name
```

**Arguments**

`x` an object of class `SpaNormFit`.  
`name` a character, specifying the name of the slot to retrieve.

**Value**

Return value varies depending on method.

**Slots**

`ngenes` a numeric, specifying the number of genes in the dataset.  
`ncells` a numeric, specifying the number of cells/spots in the dataset.  
`gene.model` a character, specifying the gene-specific model to used (see `getGeneModels()`).  
`df.tps` an integer, specifying the degrees of freedom to used for the thin plate spline.  
`sample.p` a numeric, specifying the proportion of samples used to approximated the model.  
`lambda.a` a numeric, specifying the shrinkage parameter used.  
`batch` a vector or matrix, specifying the batch design used (if any).  
`W` a matrix, specifying the covariate matrix of the linear model.  
`alpha` a matrix, specifying the coefficients of the linear model.  
`gmean` a numeric, specifying the mean estimate for each gene in the linear model.  
`psi` a numeric, specifying the over-dispersion parameter for each gene if a negative binomial model was used (or a vector of NAs if another gene model is used).  
`wtype` a factor, specifying the covariate types of columns in the covariate matrix, `W`. These could be "biology", "ls", or "batch".  
`loglik` a numeric, specifying the log-likelihood of the model at each external iteration.

**Examples**

```
example(SpaNorm)
```



# Index

## \* datasets

HumanDLPFC, [3](#)

\$, SpaNormFit-method (SpaNormFit), [7](#)

fastSizeFactors, [2](#)

fastSizeFactors, SpatialExperiment-method  
(fastSizeFactors), [2](#)

filterGenes, [3](#)

filterGenes, Seurat-method  
(filterGenes), [3](#)

filterGenes, SpatialExperiment-method  
(filterGenes), [3](#)

HumanDLPFC, [3](#)

plotSpatial, [4](#)

SpaNorm, [5](#)

SpaNorm, SpatialExperiment-method  
(SpaNorm), [5](#)

SpaNormFit, [7](#)

SpaNormFit-class (SpaNormFit), [7](#)