Package 'clustSIGNAL'

April 21, 2025

Type Package

Title ClustSIGNAL: a spatial clustering method

Version 1.0.0

Description clustSIGNAL: clustering of Spatially Informed Gene expression with Neighbourhood Adapted Learning. A tool for adaptively smoothing and clustering gene expression data. clustSIGNAL uses entropy to measure heterogeneity of cell neighbourhoods and performs a weighted, adaptive smoothing, where homogeneous neighbourhoods are smoothed more and heterogeneous neighbourhoods are smoothed less. This not only overcomes data sparsity but also incorporates spatial context into the gene expression data. The resulting smoothed gene expression data is used for clustering and could be used for other downstream analyses.

License GPL-2

Encoding UTF-8

URL https://sydneybiox.github.io/clustSIGNAL/

BugReports https://github.com/sydneybiox/clustSIGNAL/issues

biocViews Clustering, Software, GeneExpression, Spatial, Transcriptomics, SingleCell

RoxygenNote 7.3.2

Depends R (>= 4.4.0), SpatialExperiment

Imports BiocParallel, BiocNeighbors, bluster (>= 1.16.0), scater, harmony, SingleCellExperiment, SummarizedExperiment, methods, Matrix, reshape2

Suggests knitr, BiocStyle, testthat (>= 3.0.0), aricode, ggplot2, patchwork, dplyr, scattermore

VignetteBuilder knitr

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

git_branch RELEASE_3_21

git_last_commit e239a7a

git_last_commit_date 2025-04-15

Repository Bioconductor 3.21

Date/Publication 2025-04-21

Author Pratibha Panwar [cre, aut, ctb] (ORCID:
<https://orcid.org/0000-0002-7437-7084>),
Boyi Guo [aut],
Haowen Zhao [aut],
Stephanie Hicks [aut],
Shila Ghazanfar [aut, ctb] (ORCID:
<https://orcid.org/0000-0001-7861-6997>)

Maintainer Pratibha Panwar <pratibhapanwar.4@gmail.com>

Contents

`.calculateProp` *Cell neighbourhood composition*

Description

A function to calculate the cell neighbourhood composition of initial cluster labels.

Usage

```
.calculateProp(arr)
```

Arguments

`arr`	a vector of initial subcluster labels of each cell in the neighbourhood.

Value

a table of initial subcluster proportions in a neighbourhood.

.cellName *Neighbour cell naming*

Description

A function to fetch cell IDs.

Usage

```
.cellName(cell, Clust)
```

Arguments

cell	a vector of neighbourhood cell indices. The cell indices indicate the row number of cells in sample metadata.
Clust	a data frame of initial cluster labels of each cell in the sample.

Value

a data frame of cell IDs of neighbourhood cells.

.cellNameSort *Neighbour cell sorting*

Description

A function to perform neighbourhood cell sorting. Neighbourhood cells that belong to the same initial cluster as the index cell are moved closer to the index cell.

Usage

```
.cellNameSort(cell, Clust)
```

Arguments

cell	a vector of neighbourhood cell indices. The cell indices indicate the row number of cells in sample metadata.
Clust	a data frame of initial cluster labels of each cell in the sample.

Value

a data frame of cell IDs of sorted neighbourhood cells.

.clustNum *Neighbour cell initial subcluster label*

Description

A function to fetch initial subcluster label of cell.

Usage

```
.clustNum(cell, subClust)
```

Arguments

cell	a vector of neighbourhood cell indices. The cell indices indicate the row number of cells in sample metadata.
subClust	a data frame of initial subcluster labels of each cell in the sample.

Value

a data frame of initial subcluster labels of neighbourhood cells.

.exp_kernel *Exponential distribution weights*

Description

A function to generate weights from an exponential distribution.

Usage

```
.exp_kernel(ed, NN, rate)
```

Arguments

ed	a numeric vector of entropy values of all cell neighbourhoods.
NN	an integer for the number of neighbourhood cells including the index cell.
rate	a numeric value for rate of exponential distribution.

Value

a matrix where the columns contain weights associated with the entropy values.

.gauss_kernel *Gaussian distribution weights*

Description

A function to generate weights from a Gaussian distribution.

Usage

```
.gauss_kernel(ed, NN, sd)
```

Arguments

ed	a numeric vector of entropy values of all cell neighbourhoods.
NN	an integer for the number of neighbourhood cells including the index cell.
sd	a numeric value for standard deviation of Gaussian distribution.

Value

a matrix where the columns contain weights associated with the entropy values.

.generateBPParam *Generating BPParam object*

Description

A utility function to generate BPPARAM object.

Usage

```
.generateBPParam(cores = 1)
```

Arguments

cores	Desired number of cores for BPPARAM object.

Value

A BPPPARAM object.

adaptiveSmoothing *Adaptive smoothing*

Description

A function to perform a weighted, adaptive smoothing of the gene expression of each cell based on the heterogeneity of its neighbourhood. Heterogeneous neighbourhoods are smoothed less with higher weights given to cells belonging to same initial cluster as the index cell. Homogeneous neighbourhoods are smoothed more with similar weights given to most cells.

Usage

```
adaptiveSmoothing(spe, nnCells, NN = 30, kernel = "G", spread = 0.3)
```

Arguments

spe	SpatialExperiment object containing neighbourhood entropy values of each cell.
nnCells	a character matrix of NN nearest neighbours - rows are index cells and columns are their nearest neighbours ranging from closest to farthest neighbour. For sort = TRUE, the neighbours belonging to the same initial cluster as the index cell are moved closer to it.
NN	an integer for the number of neighbouring cells the function should consider. The value must be greater than or equal to 1. Default value is 30.
kernel	a character for type of distribution to be used. The two valid values are "G" or "E" for Gaussian and exponential distributions, respectively. Default value is "G".
spread	a numeric value for distribution spread, represented by standard deviation for Gaussian distribution and rate for exponential distribution. Default value is 0.3 for Gaussian distribution. The recommended value is 5 for exponential distribution.

Value

SpatialExperiment object including smoothed gene expression as an additional assay.

Examples

```
data(ClustSignal_example)

# requires matrix containing NN nearest neighbour cell labels (nnCells),
# generated using the neighbourDetect() function
spe <- clustSIGNAL::adaptiveSmoothing(spe, nnCells)
spe
```

clustSIGNAL *ClustSIGNAL*

Description

A clustering method for spatially-resolved cell-type classification of spatial transcriptomics data. The tool generates and uses an adaptively smoothed, spatially informed gene expression data for clustering.

Usage

```
clustSIGNAL(
  spe,
  samples,
  dimRed = "None",
  batch = FALSE,
  batch_by = "None",
  NN = 30,
  kernel = "G",
  spread = 0.3,
  sort = TRUE,
  threads = 1,
  outputs = "c",
 clustParams = list(clust_c = 0, subclust_c = 0, iter.max = 30, k = 10, cluster.fun =
    "louvain")
)
```

Arguments

spe	a SpatialExperiment object containing spatial coordinates in 'spatialCoords' matrix and normalised gene expression in 'logcounts' assay.
samples	a character indicating name of colData(spe) column containing sample names.
dimRed	a character indicating the name of the reduced dimensions to use from the SpatialExperiment object (i.e., from reducedDimNames(spe)). Default value is 'None'.
batch	a logical parameter for whether to perform batch correction. Default value is FALSE.
batch_by	a character indicating name of colData(spe) column containing the batch names. Default value is 'None'.
NN	an integer for the number of neighbouring cells the function should consider. The value must be greater than or equal to 1. Default value is 30.
kernel	a character for type of distribution to be used. The two valid values are "G" or "E" for Gaussian and exponential distributions, respectively. Default value is "G".

`spread` a numeric value for distribution spread, represented by standard deviation for Gaussian distribution and rate for exponential distribution. Default value is 0.3 for Gaussian distribution. The recommended value is 5 for exponential distribution.

`sort` a logical parameter for whether to sort the neighbourhood by initial clusters. Default value is TRUE.

`threads` a numeric value for the number of CPU cores to be used for the analysis. Default value set to 1.

`outputs` a character for the type of output to return to the user. "c" for data frame of cell IDs and their respective ClustSIGNAL cluster labels, "n" for ClustSIGNAL cluster dataframe plus neighbourhood matrix, "s" for ClustSIGNAL cluster dataframe plus final SpatialExperiment object, or "a" for all 3 outputs.

`clustParams` a list of parameters for TwoStepParam clustering methods: clust_c is the number of centers to use for clustering with KmeansParam. By default set to 0, in which case the method uses either 5000 centers or 1/5th of the total cells in the data as the number of centers, whichever is lower. subclust_c is the number of centers to use for sub-clustering the initial clusters with KmeansParam. The default value is 0, in which case the method uses either 1 center or half of the total cells in the initial cluster as the number of centers, whichever is higher. iter.max is the maximum number of iterations to perform during clustering and sub-clustering with KmeansParam. Default value is 30. k is a numeric value indicating the k-value used for clustering and sub-clustering with NNGraphParam. Default value is 10. cluster.fun is a character indicating the graph clustering method used with NNGraphParam. By default, the Louvain method is used.

Value

a list of outputs depending on the type of outputs specified in the main function call.

1. clusters: a data frame of cell names and their ClustSIGNAL cluster classification.

2. neighbours: a character matrix containing cells IDs of each cell's NN neighbours.

3. spe_final: a SpatialExperiment object containing the original spe object data plus initial cluster and subcluster labels, entropy values, smoothed gene expression, and ClustSIGNAL cluster labels.

Examples

```
data(ClustSignal_example)

names(colData(spe))
# identify the column name with sample labels
samples = "sample_id"
res_list <- clustSIGNAL(spe, samples, outputs = "c")
```

ClustSignal_example *Example data with SpatialExperiment object*

Description

This example data was generated from the mouse embryo spatial transcriptomics dataset of 3 mouse embryos, with 351 genes and a total of 57536 cells. For running examples, we subset the data by selecting 1000 random cells from embryo 2, excluding any cells annotated as 'low quality'. After subsetting, we have expression for 351 genes from 1000 cells in embryo 2.

Usage

```
data(ClustSignal_example)
```

Format

`spe` a spatialExperiment object containing gene expression matrix with normalised counts, where rows indicate genes and columns indicate cells. Also, contains a cell metadata including cell IDs, sample IDs, cell type annotations, and x-y coordinates of cells. `nnCells` a matrix where each row corresponds to a cell in spe object, and the columns correspond to the nearest neighbors. `regXclust` a list where each element corresponds to a cell in spe object, and contains the cluster composition proportions.

Source

Integration of spatial and single-cell transcriptomic data elucidates mouse organogenesis, *Nature Biotechnology*, 2021. Webpage: https://www.nature.com/articles/s41587-021-01006-2

entropyMeasure *Heterogeneity measure*

Description

A function to measure the heterogeneity of a cell's neighbourhood in terms of entropy. Generally, homogeneous neighbourhoods have low entropy and heterogeneous neighbourhoods have high entropy.

Usage

```
entropyMeasure(spe, regXclust, threads = 1)
```

Arguments

`spe`	SpatialExperiment object with initial cluster and subcluster labels.
`regXclust`	a list of vectors of each cell's neighbourhood composition indicated by the proportion of initial subclusters it contains.
`threads`	a numeric value for the number of CPU cores to be used for the analysis. Default value set to 1.

Value

SpatialExperiment object with entropy values associated with each cell.

Examples

```
data(ClustSignal_example)

# requires list containing cluster proportions of each region (regXclust),
# generated using the neighbourDetect() function
spe <- clustSIGNAL::entropyMeasure(spe, regXclust)
spe$entropy |> head()
```

---

mEmbryo2 *Mouse Embryo Data*

Description

This dataset contains spatial transcriptomics data from 3 mouse embryos, with 351 genes and a total of 57536 cells. For vignettes, we subset the data by randomly selecting 5000 cells from embryo 2, excluding cells that were annotated as 'low quality'.

Usage

```
data(mEmbryo2)
```

Format

`me_expr` a gene expression matrix with normalised counts, where rows indicate genes and columns indicate cells. `me_data` a data frame of cell metadata including cell IDs, sample IDs, cell type annotations, and x-y coordinates of cells.

Source

Integration of spatial and single-cell transcriptomic data elucidates mouse organogenesis, *Nature Biotechnology*, 2021. Webpage: https://www.nature.com/articles/s41587-021-01006-2

mHypothal *Mouse Hypothalamus Data*

Description

This dataset contains spatial transcriptomics data from 181 mouse hypothalamus samples, 155 genes and a total of 1,027,080 cells. For running the vignettes, we subset the data by selecting total 6000 cells from only 3 samples - Animal 1 Bregma -0.09 (2080 cells) and Animal 7 Bregmas 0.16 (1936 cells) and -0.09 (1984 cells), excluding cells that were annotated as 'ambiguous', and removed 20 genes that were assessed using a different technology.

Usage

```
data(mHypothal)
```

Format

`mh_expr` a gene expression matrix with normalised counts, where rows indicate genes and columns indicate cells. `mh_data` a data frame of cell metadata including cell IDs, sample IDs, cell type annotations, and x-y coordinates of cells.

Source

Molecular, Spatial and Functional Single-Cell Profiling of the Hypothalamic Preoptic Region, *Science*, 2018. Webpage: https://www.science.org/doi/10.1126/science.aau5324

neighbourDetect *Cell neighbourhood detection*

Description

A function to identify the neighbourhood of each cell. If sort = TRUE, the neighbourhoods are also sorted such that cells belonging to the same 'initial cluster' as the index cell are arranged closer to it.

Usage

```
neighbourDetect(spe, samples, NN = 30, sort = TRUE, threads = 1)
```

Arguments

spe	SpatialExperiment object with initial cluster and subcluster labels.
samples	a character indicating name of colData(spe) column containing sample names.
NN	an integer for the number of neighbouring cells the function should consider. The value must be greater than or equal to 1. Default value is 30.
sort	a logical parameter for whether to sort the neighbourhood by initial clusters. Default value is TRUE.
threads	a numeric value for the number of CPU cores to be used for the analysis. Default value set to 1.

Value

a list containing two items:

1. nnCells, a character matrix of NN nearest neighbours - rows are index cells and columns are their nearest neighbours ranging from closest to farthest neighbour. For sort = TRUE, the neighbours belonging to the same initial cluster as the index cell are moved closer to it.

2. regXclust, a list of vectors of each cell's neighbourhood composition indicated by the proportion of initial subclusters it contains.

Examples

```
data(ClustSignal_example)

out_list <- clustSIGNAL::neighbourDetect(spe, samples = "sample_id")
out_list |> names()
```

---

p1_clustering *Initial non-spatial clustering*

Description

A function to perform initial non-spatial clustering and sub-clustering of normalised gene expression to generate 'initial clusters' and 'initial subclusters'.

Usage

```
p1_clustering(
  spe,
  dimRed = "None",
  batch = FALSE,
  batch_by = "None",
  threads = 1,
 clustParams = list(clust_c = 0, subclust_c = 0, iter.max = 30, k = 10, cluster.fun =
    "louvain")
)
```

Arguments

spe	a SpatialExperiment object containing spatial coordinates in 'spatialCoords' matrix and normalised gene expression in 'logcounts' assay.
dimRed	a character indicating the name of the reduced dimensions to use from the SpatialExperiment object (i.e., from reducedDimNames(spe)). Default value is 'None'.
batch	a logical parameter for whether to perform batch correction. Default value is FALSE.
batch_by	a character indicating name of colData(spe) column containing the batch names. Default value is 'None'.
threads	a numeric value for the number of CPU cores to be used for the analysis. Default value set to 1.
clustParams	a list of parameters for TwoStepParam clustering methods: clust_c is the number of centers to use for clustering with KmeansParam. By default set to 0, in which case the method uses either 5000 centers or 1/5th of the total cells in the data as the number of centers, whichever is lower. subclust_c is the number of centers to use for sub-clustering the initial clusters with KmeansParam. The default value is 0, in which case the method uses either 1 center or half of the total cells in the initial cluster as the number of centers, whichever is higher. iter.max is the maximum number of iterations to perform during clustering and sub-clustering with KmeansParam. Default value is 30. k is a numeric value indicating the k-value used for clustering and sub-clustering with NNGraphParam. Default value is 10. cluster.fun is a character indicating the graph clustering method used with NNGraphParam. By default, the Louvain method is used.

Value

SpatialExperiment object with initial cluster and subcluster labels of each cell.

Examples

```
data(ClustSignal_example)

spe <- clustSIGNAL::p1_clustering(spe, dimRed = "PCA")
spe$nsCluster |> head()
spe$initCluster |> head()
```

---

p2_clustering *Final non-spatial clustering*

Description

A function to perform clustering on adaptively smoothed gene expression data to generate ClustSIGNAL clusters.

Usage

```
p2_clustering(
  spe,
  batch = FALSE,
  batch_by = "None",
  threads = 1,
 clustParams = list(clust_c = 0, subclust_c = 0, iter.max = 30, k = 10, cluster.fun =
    "louvain")
)
```

Arguments

spe	SpatialExperiment object containing the adaptively smoothed gene expression.
batch	a logical parameter for whether to perform batch correction. Default value is FALSE.
batch_by	a character indicating name of colData(spe) column containing the batch names. Default value is 'None'.
threads	a numeric value for the number of CPU cores to be used for the analysis. Default value set to 1.
clustParams	a list of parameters for TwoStepParam clustering methods: clust_c is the number of centers to use for clustering with KmeansParam. By default set to 0, in which case the method uses either 5000 centers or 1/5th of the total cells in the data as the number of centers, whichever is lower. subclust_c is the number of centers to use for sub-clustering the initial clusters with KmeansParam. This parameter is not used in the final clustering step. iter.max is the maximum number of iterations to perform during clustering and sub-clustering with KmeansParam. Default value is 30. k is a numeric value indicating the k-value used for clustering with NNGraphParam. Default value is 10. cluster.fun is a character indicating the graph clustering method used with NNGraphParam. By default, the Louvain method is used.

Value

SpatialExperiment object containing clusters generated from smoothed data.

Examples

```
data(ClustSignal_example)

# For non-spatial clustering of normalised counts
spe <- clustSIGNAL::p2_clustering(spe)
spe$ClustSIGNAL |> head()
```

Index