Package ‘EnDecon’

November 23, 2022

Type Package

Title EnDecon: cell type deconvolution of spatially resolved transcriptomics data via ensemble learning

Version 0.2.0

Author Jian-Juan Tu, Hui-Sheng Li

Maintainer Jian-Juan Tu, Hui-Sheng Li<lihs@mails.ccnu.edu.cn>

Description EnDecon is an ensemble learning method to estimate cell type abundances within spots for spatially resolved transcriptomics data by borrowing strengths from existing cell type deconvolution methods. EnDecon utilizes an optimization strategy for the combination of the base deconvolution results from twelve individual methods (designed for both bulk RNA-seq and SRT data) to produce a consensus deconvolution result. The current implementation of EnDecon integrates twelve state-of-the-art methods: CARD, cell2location, DeconRNASeq, DWLS, MuSiC (MuSiC weighted and MuSiC all gene), RCTD, SCDC, SpatialDWLS, SPOTlight,Stereoscope, and SVR.

Depends R (>= 3.5.0), pcaMethods

Imports methods,
SCDC,
spacexr,
MuSiC,
DeconRNASeq,
DWLS,
Seurat,
SPOTlight,
Giotto,
spatstat.geom,
CARD,
NMF,
utils,
stats,
graphics,
parallel,
doParallel,
foreach,
reticulate,
Biobase,

data.table,
Matrix,
abind,
STdeconvolve

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.2.1

License GPL(>= 2)

Encoding UTF-8

LazyData true

biocViews

Remotes renozao/xbioc,
meichendong/SCDC,
dmcable/spacexr,
xuranw/MuSiC,
RubD/Giotto,
YingMa0107/CARD

R topics documented:

`breast.sc.cell.label` *cell type labels of scRNA-seq data*

Description

cell type labels of scRNA-seq data

Usage

```
data(breast.sc.cell.label)
```

Format

a vector

Examples

```
data(breast.sc.cell.label)
```

breast.sc.ref *raw count matrix of reference scRNA-seq dataset*

Description

We obtain the filtered human breast cancers scRNA-seq dataset from [Zenodo data repository] (<https://doi.org/10.5281/zenodo.4739739>) The dataset contains the expression levels of 11920 genes and 3024 cells.

Usage

```
data(breast.sc.ref)
```

Format

a large matrix

Examples

```
data(breast.sc.ref)
```

breast.spot.annotation *The annotation of spots of breast cancer data*

Description

109 spots in connective tissue, 7 spots in immune infiltrate, 139 spots in invasive cancer, 51 spots in undermined. The breast.spot.annotation just for the down-streaming analysis.

Usage

```
data(breast.spot.annotation)
```

Format

a vector

Examples

```
data(breast.spot.annotation)
```

breast.st *raw count matrix of spatial transcriptomics data*

Description

We obtain the filtered human breast cancers scRNA-seq dataset from [Zenodo data repository] (<https://doi.org/10.5281/zenodo.4739739>). The dataset contains the expression levels of 11920 genes and 306 spots.

Usage

```
data(breast.st)
```

Format

a large matrix

Examples

```
data(breast.st)
```

breast.st.loc *coordinate of spots for the spatially resolved transcriptomics data*

Description

coordinate of spots for the breast.st. The center of the grids are served as the coordinates of the corresponding generated spots.

Usage

```
data(breast.st.loc)
```

Format

a list

Examples

```
data(breast.st.loc)
```

data_process *This function focuses on cleaning scRNA-seq and stRNA-seq datasets.*

Description

This function focuses on cleaning scRNA-seq and stRNA-seq datasets.

Usage

```
data_process(
  sc_exp,
  sc_label,
  spot_exp,
  spot_loc,
  gene_det_in_min_cells_per = 0.01,
  expression_threshold = 1,
  nUMI = 100,
  verbose = FALSE,
  plot = FALSE
)
```

Arguments

`sc_exp` scRNA-seq matrix, genes * cells. The format should be raw-counts. The matrix need include gene names and cell names.

`sc_label` cell type information. The cells are need be divided into multiple category.

`spot_exp` stRNA-seq matrix, genes * spots. The format should be raw counts. The matrix need include gene names and spot names.

`spot_loc` coordinate matrix, spots * coordinates. The matrix need include spot names and coordinate name (x, y).

`gene_det_in_min_cells_per`
a floor variable. minimum percent of genes that need to be detected in a cell.

`expression_threshold`
a floor variable. Threshold to consider a gene expressed.

`nUMI` a floor variable. minimum of read count that need to be detected in a cell or spot.

`verbose` a logical variable that defines whether to print the processing flow of data process.

`plot` a logical variable that defines whether to plot the selected genes and selected cell expression.

Value

a list includes processed scRNA-seq matrix, cell type, stRNA-seq matrix.

Examples

```
data("breast.sc.ref")
data("breast.sc.cell.label")
data("breast.st")
data("breast.st.loc")
database <- data_process(breast.sc.ref, breast.sc.cell.label, breast.st, breast.st.loc)
```

EnDecon_individual_methods

Running each base deconvolution method individually to obtain the base cell type deconvolution results on spatially resolved transcriptomics data.

Description

This function is implemented to perform individual deconvolution methods. The current implementation of EnDecon integrates twelve state-of-the-art methods: CARD, cell2location, DeconRNASeq, DWLS, MuSiC (MuSiC weighted and MuSiC all gene), RCTD, SCDC, SpatialDWLS, SPOTlight,Stereoscope, and SVR. These packages will be automatically installed along with EnDecon.

Usage

```
EnDecon_individual_methods(
  sc_exp,
  sc_label,
  spot_exp,
  spot_loc,
  gene_det_in_min_cells_per = 0.01,
  expression_threshold = 1,
  nUMI = 100,
  verbose = FALSE,
  plot = FALSE,
  python_env = NULL,
  use_gpu = FALSE,
  saving_results = FALSE,
  SCDC = TRUE,
  RCTD = TRUE,
  MuSiC = TRUE,
  DeconRNASeq = TRUE,
  DestVI = TRUE,
  DWLS = TRUE,
  SPOTlight = TRUE,
  SpatialDWLS = TRUE,
  Stereoscope = TRUE,
  cell2location = TRUE,
  CARD = TRUE,
  STdeconvolve = TRUE,
  SCDC.iter.max = 1000,
  RCTD.CELL_MIN_INSTANCE = 10,
  MuSiC.iter.max = 1000,
  MuSiC.nu = 1e-04,
  MuSiC.eps = 0.01,
  DeconRNASeq.perc = 0.05,
  DWLS.parallel = TRUE,
  DWLS.is_select_DEGs = TRUE,
  SPOTlight.cl_n = 100,
  SPOTlight.hvg = 3000,
  SPOTlight.min_cont = 0.001,
  SpatialDWLS.findmarker_method = "gini",
  SpatialDWLS.ncp_spa = 100,
  SpatialDWLS.dimensions_to_use = 10,
  SpatialDWLS.k = 10,
  SpatialDWLS.resolution = 0.4,
  SpatialDWLS.n_iterations = 1000,
  SpatialDWLS.n_cell = 50,
  SpatialDWLS.is_select_DEGs = TRUE,
  Stereoscope.sc_training_plot = FALSE,
  Stereoscope.sc_training_save_trained_model = FALSE,
  Stereoscope.sc_max_epochs = 10000,
  Stereoscope.sc_lr = 0.01,
  Stereoscope.select_HVG = TRUE,
  Stereoscope.HVG_num = 5000,
  Stereoscope.st_training_plot = FALSE,
  Stereoscope.st_training_save_trained_model = FALSE,
  Stereoscope.st_max_epochs = 10000,
  Stereoscope.st_lr = 0.01,
  cell2location.sc_max_epoches = 1000,
  cell2location.sc_lr = 0.002,
  cell2location.st_N_cells_per_location = 30,
  cell2location.st_detection_alpha = 200,
  cell2location.st_max_epoches = 10000,
  CARD.minCountGene = 100,
  CARD.minCountSpot = 5,
  STdeconvolve.min.lib.size = 100,
  STdeconvolve.min.reads = 1,
  STdeconvolve.nTopOD = 1000,
  STdeconvolve.betaScale = 1000,
  DestVI.n_top_genes = 2000,
  DestVI.max_iter_sc = 400,
  DestVI.max_iter_st = 3000
)
```

Arguments

`sc_exp` scRNA-seq matrix, genes * cells. The format should be raw-counts. The matrix need include gene names and cell names.

`sc_label` cell type information. The cell need be divided into multiple categories.

`spot_exp` stRNA-seq matrix, genes * spots. The format should be raw-counts. The matrix need include gene names and spot names.

`spot_loc` coordinate matrix, spots * coordinates. The matrix need include spot names and coordinate name (x, y).

`gene_det_in_min_cells_per` a floor variable. minimum percent # of genes that need to be detected in a cell.

`expression_threshold` a floor variable. Threshold to consider a gene expressed.

`nUMI` a floor variable. minimum # of read count that need to be detected in a cell or spot.

`verbose` a logical variable that defines whether to print the processing flow of data process.

`plot` a logical variable that defines whether to plot the selected genes and selected cell expression.

`python_env` the path of python environment. We recommend user construct python environment by the .yml provided by ours.

`use_gpu` a logical variable whether to use GPU to train Stereoscope and cell2location.

`saving_results` a logical variable whether to save the results of individual deconvolution methods.

`SCDC` a logical variable whether to apply SCDC.

`RCTD` a logical variable whether to apply RCTD.

`MuSiC` a logical variable whether to apply MuSiC all gene and MuSiC weighted.

`DeconRNASeq` a logical variable whether to apply DeconRNASeq.

`DestVI` a logical variable whether to apply DestVI.

`DWLS` a logical variable whether to apply DWLS and SVR.

`SPOTlight` a logical variable whether to apply SPOTlight.

`SpatialDWLS` a logical variable whether to apply SpatialDWLS.

`Stereoscope` a logical variable whether to apply Stereoscope.

`cell2location` a logical variable whether to apply cell2location.

`CARD` a logical variable whether to apply CARD.

`STdeconvolve` a logical variable whether to apply STdeconvolve.

`SCDC.iter.max` a integer variable represents the maximum number of iteration in WNNLS of SCDC.

`RCTD.CELL_MIN_INSTANCE` a integer value represent the min cells in one cell type for reference scRAN-seq.

`MuSiC.iter.max` a integer variable represents maximum iteration number of MuSiC training.

`MuSiC.nu` a floor variable represents regulation parameter in MuSiC model.

`MuSiC.eps` a floor variable represents threshold of convergence of training model.

`DeconRNASeq.perc`
a floor variable represents the values for filter cells.

`DWLS.parallel` a logical variable indicating whether to apply DWL with multiple CPU. Default setting is TRUE.

`DWLS.is_select_DEGs`
a logical variable indicating whether to select genes for each cell type of scRNA-seq dataset. Default setting is TRUE.

`SPOTlight.cl_n` integer variable indicating how many cells to keep from each cluster. If a cluster has n < cl_n then all cells will be selected, if it has more then cl_n will be sampled randomly. Default value is 100.

`SPOTlight.hvg` integer variable that represents number of highly variable genes to use on top of the marker genes. Default values is 3000.

`SPOTlight.min_cont`
floor variable indicates the minimum contribution we expect from a cell in that spot. Default values is 0.001.

`SpatialDWLS.findmarker_method`
a string vector indicating method to use to detect differentially expressed genes.

`SpatialDWLS.ncp_spa`
a integer value indicating number of principal components to calculate. Default setting is 100.

`SpatialDWLS.dimensions_to_use`
a integer value indicating number of dimensions to use as input for constructing KNN network. Default setting is 10.

`SpatialDWLS.k` a integer value indicating number of k neighbors to use for constructing KNN network. Default setting is 10.

`SpatialDWLS.resolution`
resolution in doLeidenCluster function in Giotto package. Default setting is 0.4.

`SpatialDWLS.n_iterations`
number of interations to run the Leiden algorithm. If the number of iterations is negative, the Leiden algorithm is run until an iteration in which there was no improvement.

`SpatialDWLS.n_cell`
number of cells per spot. Default setting is 50.

`SpatialDWLS.is_select_DEGs`
a logical value whether to select genes before applying for the SpatialDWLS.

`Stereoscope.sc_training_plot`
a logical variable whether to plot the training loss indicating whether to increase the number of maximum epoch for training for scRNA-seq dataset. Default setting is FALSE.

`Stereoscope.sc_training_save_trained_model`
a logical variable whether to save the trained model for scRNA-seq dataset. Default setting is FALSE.

`Stereoscope.sc_max_epochs`
an integer variable indicating the maximum epoches for training scRNA-seq. Default setting is 400.

`Stereoscope.sc_lr`
an integer variable indicating the learning rate for training scRNA-seq. Default setting is 0.01.

`Stereoscope.select_HVG`
a logical variable whether to select highly variable genes for the scRNA-seq data. Default setting is TRUE.

`Stereoscope.HVG_num`
number of selected highly variable genes if Stereoscope.select_HVG = TRUE. Default setting is 5000.

`Stereoscope.st_training_plot`
a logical variable whether to plot the training loss indicating whether to increase the number of maximum epoch for training for stRNA-seq dataset. Default setting is FALSE.

`Stereoscope.st_training_save_trained_model`
a logical variable whether to plot the training loss indicating whether to increase the number of maximum epoch for training for stRNA-seq dataset. Default setting is FALSE.

`Stereoscope.st_max_epochs`
an integer variable indicating the maximum epoches for training sTRNA-seq. Default setting is 400.

`Stereoscope.st_lr`
an integer variable indicating the learning rate for training sTRNA-seq. Default setting is 0.01.

`cell2location.sc_max_epoches`
an integer variable indicating the maximum epoches for training scRNA-seq.

`cell2location.sc_lr`
an integer variable indicating the learning rate for training scRNA-seq.

`cell2location.st_N_cells_per_location`
a integer variable indicating the number of cells in each spot.

`cell2location.st_detection_alpha`
a floor variable indicating the super-parameter of regularization.

`cell2location.st_max_epoches`
an integer variable indicating the maximum epoches for training stRNA-seq.

`CARD.minCountGene`
an integer variable indicating the minimum counts for each gene for the construct CARD object. Default setting is 100.

`CARD.minCountSpot`
an integer variable indicating the minimum counts for each spatial location. Default setting is 5.

`STdeconvolve.min.lib.size`
Minimum number of genes detected in a cell. Cells with fewer genes will be removed for the stRNA-seq dataset. Default setting is 1.

`STdeconvolve.min.reads`
: Minimum number of reads per gene. Genes with fewer reads will be removed. Default setting is 1.

`STdeconvolve.nTopOD`
: Number of top over-dispersed genes to use for the stRNA-seq data. Default setting is 1000.

`STdeconvolve.betaScale`
: Factor to scale the predicted cell-type gene expression profiles. Default setting is 1000.

`DestVI.n_top_genes`
: Number of selected HVGs by Seurat.V3. Default setting is 2000.

`DestVI.max_iter_sc`
: Maximum number of epoches for the training of scRNA-seq data. Default setting is 400.

`DestVI.max_iter_st`
: Maximum number of epoches for the training of stRNA-seq data. Default setting is 3000.

Value

a list contains all the results inferred by individual deconvolution methods and the times of running individual methods. The elements of list is a matrix, spots * cell-type and a time vector.

---

solve_ensemble *Ensemble the results of individual deconvolution results. This function uses the weighted median methods to integrate the results obtained by individual deconvolution methods.*

Description

Ensemble the results of individual deconvolution results. This function uses the weighted median methods to integrate the results obtained by individual deconvolution methods.

Usage

```
solve_ensemble(
  Results.Deconv,
  lambda = NULL,
  prob.quantile = 0.5,
  niter = 100,
  epsilon = 1e-05
)
```

Arguments

`Results.Deconv` a list contains all the results of individual deconvolution methods. The elements of list is a matrix, spots * cell-type.

`lambda` hyper-parameter constrain the weight of individual methods for ensemble. If the parameter is set to NULL, then, we will adopt the value in our algorithm.

`prob.quantile` numeric of probabilities with values in [0,1]. Default setting is 0.5.

`niter` a positive integer represents the maximum number of updating algorithm. Default setting is 100.

`epsilon` a parameter represents the stop criterion.

Value

a list contains a matrix of the ensemble deconvolution result and a vector of the weight assigned to individual methods.

Examples

```
data("breast.sc.ref")
data("breast.sc.cell.label")
data("breast.st")
data("breast.st.loc")
##### path on ubuntu platform on our computer
python_env <- "~/.conda/envs/EnDecon_GPU/bin/python"
Results.dec.mouse <- EnDecon_individual_methods(sc_exp = breast.sc.ref,
sc_label = breast.sc.cell.label, spot_exp = breast.st,
spot_loc = breast.st.loc, python_env = python_env,
use_gpu = TRUE,gene_det_in_min_cells_per = 0.01,
RCTD.CELL_MIN_INSTANCE = 5, saving_results = FALSE)
ensemble.results <- solve_ensemble(Results.dec.mouse[[1]])
```

Index