Package 'scDHA'

April 3, 2024

Type Package

Title Single-Cell Decomposition using Hierarchical Autoencoder

Version 1.2.2

Maintainer Ha Nguyen <hvn0006@auburn.edu>

Description Provides a fast and accurate pipeline for single-cell analyses.
The 'scDHA' software package can perform clustering, dimension reduction and visualization, classification, and time-trajectory inference on single-cell data (Tran et.al. (2021) <DOI:10.1038/s41467-021-21312-2>).

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.4)

Imports matrixStats, foreach, doParallel, igraph, Matrix, uwot, cluster, Rcpp, RcppParallel, RcppAnnoy, methods, torch (>= 0.3.0), RhpcBLASctl, coro

LinkingTo Rcpp, RcppArmadillo, RcppParallel, RcppAnnoy

RoxygenNote 7.2.3

Suggests testthat, knitr, mclust

NeedsCompilation yes

VignetteBuilder knitr

URL https://github.com/duct317/scDHA

BugReports https://github.com/duct317/scDHA/issues

Author Ha Nguyen [cre],
Duc Tran [aut],
Tin Nguyen [fnd]

Repository CRAN

Date/Publication 2024-04-02 22:42:03 UTC

R topics documented:

`Goolam` *Goolam*

Description

Goolam dataset in list format, include scRNA-seq data and cell type information.

Usage

```
Goolam
```

Format

An object of class `list` of length 2.

`Goolam_result` *Goolam_result*

Description

Result of processing Goolam dataset using 'scDHA' function.

Usage

```
Goolam_result
```

Format

An object of class `list` of length 4.

scDHA *scDHA*

Description

The main function to perform dimension deduction and clustering.

Usage

```
scDHA(
  data = data,
  k = NULL,
  method = "scDHA",
  sparse = FALSE,
  n = 5000,
  ncores = 10L,
  gen_fil = TRUE,
  do.clus = TRUE,
  sample.prob = NULL,
  seed = NULL
)
```

Arguments

data	Gene expression matrix, with rows represent samples and columns represent genes.
k	Number of clusters, leave as default for auto detection. Has no effect when do.clus = False.
method	Method used for clustering. It can be "scDHA" or "louvain". The default setting is "scDHA".
sparse	Boolen variable indicating whether data is a sparse matrix. The input must be a non negative sparse matrix.
n	Number of genes to keep after feature selection step.
ncores	Number of processor cores to use.
gen_fil	Boolean variable indicating whether to perform scDHA gene filtering before performing dimension deduction and clustering.
do.clus	Boolean variable indicating whether to perform scDHA clustering. If do.clus = False, only dimension deduction is performed.
sample.prob	Probability used for classification application only. Leave this parameter as default, no user input is required.
seed	Seed for reproducibility.

Value

List with the following keys:

- cluster - A numeric vector containing cluster assignment for each sample. If `do.clus = False`, this values is always `NULL`.
- latent - A matrix representing compressed data from the input data, with rows represent samples and columns represent latent variables.

Examples

```
library(scDHA)
#Load example data (Goolam dataset)
data('Goolam'); data <- t(Goolam$data); label <- as.character(Goolam$label)
#Log transform the data
data <- log2(data + 1)
if(torch::torch_is_installed()) #scDHA need libtorch installed
{
  #Generate clustering result, the input matrix has rows as samples and columns as genes
  result <- scDHA(data, ncores = 2, seed = 1)
  #The clustering result can be found here
  cluster <- result$cluster
}
```

---

scDHA.class *scDHA classification*

Description

Perform classification of new data based on available data.

Usage

```
scDHA.class(
  train = train,
  train.label = train.label,
  test = test,
  ncores = 10L,
  seed = NULL
)
```

Arguments

`train`	Expression matrix of available data, with rows represent samples and columns represent genes.
`train.label`	A vector containing label for each sample in training data.

test	Expression matrix new data for classification, with rows represent samples and columns represent genes.
ncores	Number of processor cores to use.
seed	Seed for reproducibility.

Value

A vector contain classified labels for new data.

Examples

```
library(scDHA)
#Load example data (Goolam dataset)
data('Goolam'); data <- t(Goolam$data); label <- as.character(Goolam$label)
#Log transform the data
data <- log2(data + 1)
#Split data into training and testing sets
set.seed(1)
idx <- sample.int(nrow(data), size = round(nrow(data)*0.75))
train.x <- data[idx, ]; train.y <- label[idx]
test.x <- data[-idx, ]; test.y <- label[-idx]
if(torch::torch_is_installed()) #scDHA need libtorch installed
{
  #Predict the labels of cells in testing set
  prediction <- scDHA.class(train = train.x, train.label = train.y, test = test.x,
                            ncores = 2, seed = 1)
  #Calculate accuracy of the predictions
  acc <- round(sum(test.y == prediction)/length(test.y), 2)
  print(paste0("Accuracy = ", acc))
}
```

---

scDHA.pt *scDHA pseudo time inference*

Description

Inferring pseudo-time data.

Usage

```
scDHA.pt(sc = sc, start.point = 1, ncores = 10L, seed = NULL)
```

Arguments

sc	Embedding object, produced by scDHA function.
start.point	Starting point of the trajectory.
ncores	Number of processor cores to use.
seed	Seed for reproducibility.

Value

List with the following keys:

- pt - Pseudo-time values for each sample.

Examples

```
library(scDHA)
#Load example data (Goolam dataset)
data('Goolam'); data <- t(Goolam$data); label <- as.character(Goolam$label)
#Log transform the data
data <- log2(data + 1)
if(torch::torch_is_installed()) #scDHA need libtorch installed
{
  #Generate clustering result, the input matrix has rows as samples and columns as genes
  result <- scDHA(data, ncores = 2, seed = 1)
  #Cell stage order in Goolam dataset
  cell.stages <- c("2cell", "4cell", "8cell", "16cell", "blast")
  #Generate pseudo-time for each cell, the input is the output from scDHA function
  result <- scDHA.pt(result, start.point = 1, ncores = 2, seed = 1)
  #Calculate R-squared value
 r2 <- round(cor(result$pt, as.numeric(factor(label, levels = cell.stages)))^2, digits = 2)
}
```

`scDHA.vis` *scDHA visulization*

Description

Generating 2D embeded data for visulation.

Usage

```
scDHA.vis(sc = sc, method = "UMAP", ncores = 10L, seed = NULL)
```

Arguments

`sc`	Embedding object produced by the `scDHA` function.
`method`	Visualization method to use. It can be "UMAP" or "scDHA". The default setting is "UMAP".
`ncores`	Number of processor cores to use.
`seed`	Seed for reproducibility.

Value

a list with the following keys:

- pred - A matrix representing the 2D projection of single-cell data, where rows represent samples and columns represent latent components.

Examples

```
library(scDHA)
#Load example data (Goolam dataset)
data('Goolam'); data <- t(Goolam$data); label <- as.character(Goolam$label)
#Log transform the data
data <- log2(data + 1)
if(torch::torch_is_installed()) #scDHA need libtorch installed
{
  #Generate clustering result, the input matrix has rows as samples and columns as genes
  result <- scDHA(data, ncores = 2, seed = 1)
  #Generate 2D representation, the input is the output from scDHA function
  result <- scDHA.vis(result, ncores = 2, seed = 1)
  #Plot the representation of the dataset, different colors represent different cell types
  plot(result$pred, col=factor(label), xlab = "scDHA1", ylab = "scDHA2")
}
```

Index