Package ‘scDHA’

October 14, 2022

Type Package
Title Single-Cell Decomposition using Hierarchical Autoencoder
Version 1.2.0
Maintainer Duc Tran <duct@nevada.unr.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, clusterCrit, Rcpp, RcppParallel, RcppAnnoy, methods, torch (>= 0.3.0), RhpcBLASctl, coro
LinkingTo Rcpp, RcppArmadillo, RcppParallel, RcppAnnoy
RoxygenNote 7.1.0
Suggests testthat, knitr, mclust
NeedsCompilation yes
VignetteBuilder knitr
URL https://github.com/duct317/scDHA
BugReports https://github.com/duct317/scDHA/issues
Author Duc Tran [aut, cre], Tin Nguyen [fnd]
Repository CRAN
Date/Publication 2022-08-18 20:40:02 UTC
R topics documented:

- Goolam
- Goolam_result
- scDHA
- scDHA.class
- scDHA.pt
- scDHA.vis

Index

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Goolam</td>
<td>Goolam</td>
</tr>
</tbody>
</table>

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.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Goolam_result</td>
<td>Goolam_result</td>
</tr>
</tbody>
</table>

Description

Result of processing Goolam dataset using 'scDHA' function.

Usage

Goolam_result

Format

An object of class list of length 4.
**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**: Boolean 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 value is always NULL.
- **latent** - A matrix representing compressed data from the input data, with rows represent samples and columns represent latent variables.

Examples

```r
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
}
```

---

**Description**

Perform classification of new data based on available data.

**Usage**

```r
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.
Expressions matrix new data for classification, with rows represent samples and columns represent genes.

Value

A vector contain classified labels for new data.

Examples

```r
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

```r
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.
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 visualization**

**Description**
Generating 2D embeded data for visulation.

**Usage**

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

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sc</td>
<td>Embedding object produced by the scDHA function.</td>
</tr>
<tr>
<td>method</td>
<td>Visualization method to use. It can be &quot;UMAP&quot; or &quot;scDHA&quot;. The default setting is &quot;UMAP&quot;.</td>
</tr>
<tr>
<td>ncores</td>
<td>Number of processor cores to use.</td>
</tr>
<tr>
<td>seed</td>
<td>Seed for reproducibility.</td>
</tr>
</tbody>
</table>
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

* datasets
  Goolam, 2
  Goolam_result, 2

Goolam, 2
Goolam_result, 2

scDHA, 3
scDHA.class, 4
scDHA.pt, 5
scDHA.vis, 6