Package ‘scCAN’
July 19, 2021

Type  Package
Title  Single-Cell Clustering using Autoencoder and Network Fusion
Version  1.0.1
Maintainer  Bang Tran <bang.t.s@nevada.unr.edu>
Description  A single-cell Clustering method using 'Autoencoder' and Network fusion ('sc-CAN') for segregating the cells from the high-dimensional 'scRNA-Seq' data. The software automatically determines the optimal number of clusters and then partitions the cells in a way such that the results are robust to noise and dropouts. 'sc-CAN' is fast and it supports Windows, Linux, and Mac OS.
License  LGPL
Encoding  UTF-8
LazyData  true
LazyDataCompression  xz
Depends  R (>= 3.4), scDHA, FNN
Imports  purrr, stats, markdown
RoxygenNote  7.1.1
Suggests  knitr
VignetteBuilder  knitr
NeedsCompilation  no
Author  Bang Tran [aut, cre],
       Duc Tran [aut],
       Hung Nguyen [aut],
       Tin Nguyen [fnd]
Repository  CRAN
Date/Publication  2021-07-19 08:00:05 UTC

R topics documented:

adjustedRandIndex ..................................................  2
scCAN ...............................................................  2
SCE ...............................................................  4
### adjustedRandIndex

**Description**

The function to calculate adjusted Rand index value with the inputs of true clusters and predicted clusters.

**Usage**

```r
adjustedRandIndex(x, y)
```

**Arguments**

- `x`: A vector that contain predicted cluster assignment.
- `y`: A vector that contain true cluster assignment.

---

### scCAN

**Description**

This is the main function to perform sc-RNA seq data clustering. **scCAN** is a fully unsupervised scRNA-seq clustering framework that uses deep neural network and network fusion-based clustering algorithm. First, **scCAN** applies a non-negative autoencoder to filter scRNA-seq data. Second, the filtered data is passed to stacked Bayesian autoencoder to get multiple low-dimensional representations of input data. Subsequently, **scCAN** converts these compressed data into networks and unify those networks to a single graph. Then, **scCAN** uses a spectral clustering algorithm to obtain final clusters assignment.

**Usage**

```r
scCAN(
  data, 
  sparse = FALSE, 
  n.neighbors = 30, 
  alpha = 0.5, 
  n.iters = 10, 
  ncores = 10, 
  r.seed = 1
)
```
Arguments

data  Gene expression matrix, with rows represent samples and columns represent genes.
sparse  Boolean variable indicating whether data is a sparse matrix. The input must be a non negative sparse matrix.
n.neighbors  Number of neighboring cells that are used to calculate the edge’s weight. The number of neighbors are set \( n_{neighbors} = 30 \) by default.
alpha  A hyper-parameter that is used to calculate the network kernel. The value is set to \( \alpha = 0.5 \) by default.
n.iters  A hyper-parameter to set the number of network fusion iterations. It is set to \( n_{iters} = 10 \) by default.
ncores  Number of processor cores to use.
r.seed  A parameter to set a seed for reproducibility. This value is set to \( r seed = 1 \) by default.

Value

List with the following keys:

- cluster - A numeric vector containing cluster assignment for each sample.
- k - The optimal number of cluster.

References


Examples

```R
# Load the package and the example data (SCE dataset)
library(scCAN)
# Load example data
data("SCE")

# Get data matrix and label
data <- t(SCE$data); label <- as.character(SCE$cell_type)

# Generate clustering result, the input matrix has rows as samples and columns as genes
result <- scCAN(data, r.seed = 1)

# Get the clustering result
cluster <- result$cluster

# Calculate adjusted Rand Index
ari <- round(scCAN::adjustedRandIndex(cluster,label), 2)
print(paste0("ARI = ", ari))
```
Description

SCE dataset includes scRNA-seq data and cell type information.

Usage

SCE

Format

An object of class list of length 2.
Index

* datasets
  SCE, 4

adjustedRandIndex, 2

scCAN, 2
SCE, 4