Description: Cell clustering is one of the most important and commonly performed tasks in single-cell RNA sequencing (scRNA-seq) data analysis. An important step in cell clustering is to select a subset of genes (referred to as “features”), whose expression patterns will then be used for downstream clustering. A good set of features should include the ones that distinguish different cell types, and the quality of such set could have significant impact on the clustering accuracy.

FEAST is an R library for selecting most representative features before performing the core of scRNA-seq clustering. It can be used as a plug-in for the established clustering algorithms such as SC3, TSCAN, SHARP, SIMLR, and Seurat. The core of FEAST algorithm includes three steps:
1. consensus clustering;
2. gene-level significance inference;
3. validation of an optimized feature set.

License: GPL-2
Encoding: UTF-8
LazyData: true
Depends: R (>= 4.1), mclust, BiocParallel, SummarizedExperiment
biocViews: Sequencing, SingleCell, Clustering, FeatureExtraction
BugReports: https://github.com/suke18/FEAST/issues
Imports: SingleCellExperiment, methods, stats, utils, irlba, TSCAN, SC3, matrixStats
Suggests: rmarkdown, Seurat, ggpubr, knitr, testthat (>= 3.0.0), BiocStyle
VignetteBuilder: knitr
RoxygenNote: 7.1.1
NeedsCompilation: yes
**align_CellType**

**Author** Kenong Su [aut, cre], Hao Wu [aut]

**Maintainer** Kenong Su <kenong.su@emory.edu>

**git_url** https://git.bioconductor.org/packages/FEAST

**git_branch** RELEASE_3_18

**git_last_commit** fe87a45

**git_last_commit_date** 2023-10-24

**Repository** Bioconductor 3.18

**Date/Publication** 2024-02-20

---

**R topics documented:**

- align_CellType .................................................. 2
- cal_F2 ......................................................... 3
- cal_Fisher2 .................................................... 4
- cal_metrics .................................................... 4
- cal_MSE .......................................................... 5
- Consensus ....................................................... 5
- eval_Cluster .................................................... 6
- FEAST ............................................................. 7
- FEAST_fast ....................................................... 8
- Norm_Y ........................................................... 8
- process_Y ....................................................... 9
- Purity ............................................................ 10
- SC3_Clust ....................................................... 10
- Select_Model_short_SC3 ........................................ 11
- Select_Model_short_TSCAN ..................................... 12
- setUp_BPPARAM ................................................ 13
- trueclass ....................................................... 13
- TSCAN_Clust .................................................... 14
- vector2matrix .................................................. 15
- Visual_Rslt .................................................... 15
- Y ................................................................. 16

---

**Index** 17

---

align_CellType Align the cell types from the prediction with the truth.

**Description**

Align the cell types from the prediction with the truth.
Usage

align_CellType(tt0)

Arguments

 tt0 a N*N table.

Value

the matched (re-ordered) table

Examples

vec1 = rep(1:4, each=100)
vec2 = sample(vec1)
tb = table(vec1, vec2)
#tb_arg = align_CellType(tb)

cal_F2

Calculate the gene-level F score and corresponding significance level.

Description

Calculate the gene-level F score and corresponding significance level.

Usage

cal_F2(Y, classes)

Arguments

Y A gene expression matrix

classes The initial cluster labels NA values are allowed. This can directly from the Consensus function.

Value

The score vector

Examples

data(Yan)
cal_F2(Y, classes = trueclass)
cal_Fisher2

Calculate the gene-level fisher score.

Description

Calculate the gene-level fisher score.

Usage

cal_Fisher2(Y, classes)

Arguments

Y  
A gene expression matrix

classes  
The initial cluster labels NA values are allowed. This can directly from the Consensus function.

Value

The score vector This is from the paper https://arxiv.org/pdf/1202.3725.pdf Vector based calculation

cal_metrics

Calculate 3 metrics and these methods are exported in C codes. flag = 1 — Rand index, flag = 2 — Fowlkes and Mallows’s index, flag = 3 — Jaccard index

Description

Calculate 3 metrics and these methods are exported in C codes. flag = 1 — Rand index, flag = 2 — Fowlkes and Mallows’s index, flag = 3 — Jaccard index

Usage

cal_metrics(cl1, cl2, randMethod = c("Rand", "FM", "Jaccard"))

Arguments

cl1  
a vector

cl2  
a vector

randMethod  
a string chosen from "Rand", "FM", or "Jaccard"

Value

a numeric vector including three values
**cal_MSE**

| cal_MSE | Standard way to preprocess the count matrix. It is the QC step for the genes. |

### Description

Standard way to preprocess the count matrix. It is the QC step for the genes.

### Usage

```r
cal_MSE(Ynorm, cluster, return_mses = FALSE)
```

### Arguments

- `Ynorm`: A normalized gene expression matrix. If not, we will normalize it for you.
- `cluster`: The clustering outcomes. Specifically, they are cluster labels.
- `return_mses`: True or False indicating whether returning the MSE.

### Value

The MSE of the clustering centers with the predicted Y.

### Examples

```r
data(Yan)
Ynorm = Norm_Y(Y)
cluster = trueclass
MSE_res = cal_MSE(Ynorm, cluster)
```

---

**Consensus**

### Consensus Clustering

### Description

Consensus Clustering

### Usage

```r
Consensus(Y, num_pcs = 10, top_pctg = 0.33, k = 2, thred = 0.9, nProc = 1)
```
**Arguments**

- `Y` A expression matrix. It is recommended to use the raw count matrix. Users can input normalized matrix directly.
- `num_pcs` The number of top pcs that will be investigated on through consensus clustering.
- `top_pctg` Top percentage of features for dimension reduction
- `k` The number of input clusters (best guess).
- `thred` For the final GMM clustering, the probability of a cell belonging to a certain cluster.
- `nProc` number of cores for BiocParallel enviroment.

**Value**

the clustering labels and the featured genes.

**Examples**

```r
data(Yan)
set.seed(123)
rixs = sample(nrow(Y), 500)
cixs = sample(ncol(Y), 40)
Y = Y[rixs, cixs]
con = Consensus(Y, k=5)
```

```
**eval_Cluster** Calculate the a series of the evaluation statistics.

**Description**

Calculate the a series of the evaluation statistics.

**Usage**

`eval_Cluster(vec1, vec2)`

**Arguments**

- `vec1` a vector.
- `vec2` a vector. x and y are with the same length.

**Value**

a vector of evaluation metrics

**Examples**

```r
vec2 = vec1 = rep(1:4, each = 100)
vec2[1:10] = 4
acc = eval_Cluster(vec1, vec2)
```
Description

FEAST main function

Usage

FEAST(
  Y, 
  k = 2, 
  num_pcs = 10, 
  dim_reduce = c("irlba", "svd", "pca"), 
  split = FALSE, 
  batch_size = 1000, 
  nProc = 1
)

Arguments

Y  A expression matrix. Raw count matrix or normalized matrix.

k  The number of input clusters (best guess).

num_pcs  The number of top pcs that will be investigated through the consensus clustering.

dim_reduce  dimension reduction methods chosen from pca, svd, or irlba.

split  boolean. If T, using subsampling to calculate the gene-level significance.

batch_size  when split is true, need to claim the batch size for splitting the cells.

nProc  number of cores for BiocParallel enviroment.

Value

the rankings of the gene-significance.

Examples

data(Yan)
k = length(unique(trueclass))
set.seed(123)
rixs = sample(nrow(Y), 500)
cixs = sample(ncol(Y), 40)
Y = Y[rixs, cixs]
ixs = FEAST(Y, k=k)
FEAST_fast  

**Description**

FEAST main function (fast version)

**Usage**

```r
FEAST_fast(Y, k = 2, num_pcs = 10, split = FALSE, batch_size = 1000, nProc = 1)
```

**Arguments**

- `Y`: A expression matrix. Raw count matrix or normalized matrix.
- `k`: The number of input clusters (best guess).
- `num_pcs`: The number of top pcs that will be investigated through the consensus clustering.
- `split`: boolean. If T, using subsampling to calculate the gene-level significance.
- `batch_size`: when split is true, need to claim the batch size for splitting the cells.
- `nProc`: number of cores for BiocParallel enviroment.

**Value**

the rankings of the gene-significance.

**Examples**

```r
data(Yan)
k = length(unique(trueclass))
res = FEAST_fast(Y, k=k)
```

---

**Norm_Y**  

**Description**

Normalize the count expression matrix by the size factor and take the log transformation.

**Usage**

```r
Norm_Y(Y)
```

**Arguments**

- `Y`: a count expression matrix
process_Y

Value

a normalized matrix

Examples

data(Yan)
Ynorm = Norm_Y(Y)

process_Y

Standard way to preprocess the count matrix. It is the QC step for the genes.

Description

Standard way to preprocess the count matrix. It is the QC step for the genes.

Usage

process_Y(Y, thre = 2)

Arguments

Y A gene expression data (Raw count matrix)
thre The threshold of minimum number of cells expressing a certain gene (default =2)

Value

A processed gene expression matrix. It is not log transformed

Examples

data(Yan)
YY = process_Y(Y, thre=2)
**Purity**

*Calculate the purity between two vectors.*

**Description**

Calculate the purity between two vectors.

**Usage**

\[ \text{Purity}(x, y) \]

**Arguments**

- **x**: a vector.
- **y**: a vector. \( x \) and \( y \) are with the same length.

**Value**

the purity score

---

**SC3_Clust**

*SC3 Clustering*

**Description**

SC3 Clustering

**Usage**

\[ \text{SC3}_\text{Clust}(Y, k = \text{NULL}, \text{input_markers} = \text{NULL}) \]

**Arguments**

- **Y**: A expression matrix. It is recommended to use the raw count matrix.
- **k**: The number of clusters. If it is not provided, \( k \) is estimated by the default method in SC3.
- **input_markers**: A character vector including the featured genes. If they are not presented, SC3 will take care of this.

**Value**

the clustering labels and the featured genes.
Using clustering results based on feature selection to perform model selection.

**Description**

Using clustering results based on feature selection to perform model selection.

**Usage**

```r
Select_Model_short_SC3(Y, cluster, tops = c(500, 1000, 2000))
```

**Arguments**

- `Y`: A gene expression matrix
- `cluster`: The initial cluster labels. NA values are allowed. This can directly from the `Consensus` function.
- `tops`: A numeric vector containing a list of numbers corresponding to top genes; e.g., `tops = c(500, 1000, 2000)`.

**Value**

`mse` and the SC3 clustering result.

**Examples**

```r
data(Yan)
k = length(unique(trueclass))
Y = process_Y(Y, thre = 2)  # preprocess the data
set.seed(123)
rixs = sample(nrow(Y), 500)
cixs = sample(ncol(Y), 40)
Y = Y[rixs, cixs]
con_res = Consensus(Y, k=k)
  # not run
# mod_res = Select_Model_short_SC3(Y, cluster = con_res$cluster, top = c(100, 200))
```
Select_Model_short_TSCAN

Using clustering results (from TSCAN) based on feature selection to perform model selection.

Description

Using clustering results (from TSCAN) based on feature selection to perform model selection.

Usage

```r
Select_Model_short_TSCAN(
  Y,
  cluster,
  minexpr_percent = 0.5,
  cvcutoff = 1,
  tops = c(500, 1000, 2000)
)
```

Arguments

- **Y**
  A gene expression matrix
- **cluster**
  The initial cluster labels. NA values are allowed. This can directly from the Consensus function.
- **minexpr_percent**
  The threshold used for processing data in TSCAN. Using it by default.
- **cvcutoff**
  The threshold used for processing data in TSCAN. Using it by default.
- **tops**
  A numeric vector containing a list of numbers corresponding to top genes; e.g., `tops = c(500, 1000, 2000)`.

Value

`mse` and the TSCAN clustering result.

Examples

```r
data(Yan)
k = length(unique(trueclass))
Y = process_Y(Y, thre = 2) # preprocess the data
set.seed(123)
rixs = sample(nrow(Y), 500)
cixs = sample(ncol(Y), 40)
Y = Y[rixs, cixs]
con_res = Consensus(Y, k=k)
# not run
# mod_res = Select_Model_short_TSCAN(Y, cluster = con_res$cluster, top = c(100, 200))
```
**setUp_BPPARAM**

**Description**

This function sets up the environment for parallel computing.

**Usage**

```r
setUp_BPPARAM(nProc = 0, BPPARAM = NULL)
```

**Arguments**

- `nProc`: number of processors
- `BPPARAM`: bpparameter from bpparam

**Value**

BAPPARAM settings

**Examples**

```r
setUp_BPPARAM(nProc=1)
```

---

**trueclass**

*An example single cell dataset for the cell label information (Yan)*

**Description**

The true cell type labels for Yan dataset. It includes 8 different cell types.

**Usage**

```r
data("Yan")
```

**Format**

A character vector contains the cell type label

**Source**


**References**

Examples

data("Yan")
table(trueclass)

TSCAN_Clust  TSCAN Clustering

Description

TSCAN Clustering

Usage

TSCAN_Clust(Y, k, minexpr_percent = 0.5, cvcutoff = 1, input_markers = NULL)

Arguments

Y  A expression matrix. It is recommended to use the raw count matrix.
k  The number of clusters. If it is not provided, k is estimated by the default method in SC3.
minexpr_percent  minimum expression threshold (default = 0.5).
cvcutoff  the cv cutoff to filter the genes (default = 1).
input_markers  A character vector including the featured genes. If they are not presented, SC3 will take care of this.

Value

the clustering labels and the featured genes.

Examples

data(Yan)
k = length(unique(trueclass))
# TSCAN_res = TSCAN_Clust(Y, k=k)
vector2matrix

function for convert a vector to a binary matrix

Description

function for convert a vector to a binary matrix

Usage

vector2matrix(vec)

Arguments

vec a vector.

Value

a n by n binary matrix indicating the adjacency.

Visual_Rslt

Using clustering results based on feature selection to perform model selection.

Description

Using clustering results based on feature selection to perform model selection.

Usage

Visual_Rslt(model_cv_res, trueclass)

Arguments

model_cv_res model selection result from Select_Model_short_SC3.
trueclass The real class labels

Value

a list of mse dataframe, clustering accuracy dataframe, and ggplot object.
Examples

data(Yan)
k = length(unique(trueclass))
Y = process_Y(Y, thre = 2) # preprocess the data
set.seed(123)
rixs = sample(nrow(Y), 500)
cixs = sample(ncol(Y), 40)
Y = Y[rixs,]
con_res = Consensus(Y, k=k)
# Not run
# mod_res = Select_Model_short_SC3(Y, cluster = con_res$cluster, top = c(100, 200))
library(ggpubr)
# Visual_Rslt(model_cv_res = mod_res, trueclass = trueclass)

Y  
\textit{An example single cell count expression matrix (Yan)}

Description

Y is a count expression matrix which belongs to "matrix" class. The data includes 124 cells about human preimplantation embryos and embryonic stem cells. It contains 19304 genes after removing genes with extreme high dropout rate.

Usage

data("Yan")

Format

An object of "matrix" class contains the count expressions

Source


References


Examples

data("Yan")
Y[1:10, 1:4]
Index

* datasets
  trueclass, 13
  Y, 16
* internal
  setUp_BPPARAM, 13

align_CellType, 2

cal_F2, 3
cal_Fisher2, 4
cal_metrics, 4
cal_MSE, 5
Consensus, 5

eval_Cluster, 6

FEAST, 7
FEAST_fast, 8

Norm_Y, 8

process_Y, 9
Purity, 10

SC3_Clust, 10
Select_Model_short_SC3, 11
Select_Model_short_TSCAN, 12
setUp_BPPARAM, 13

trueclass, 13
TSCAN_Clust, 14

vector2matrix, 15
Visual_Rslt, 15

Y, 16