Package 'DASC'

May 7, 2017

Type Package

Title Detecting hidden batch factors through data adaptive adjustment for biological effects

Version 0.99.11

Author Haidong Yi, Ayush T. Raman

Maintainer Haidong Yi <982869874@qq.com>

Description The package is used for identifying batches in high-dimensional dataset.

Depends R (>= 3.4.0),

Imports NMF (>= 0.20.6), cvxclustr (>= 1.1.1), Biobase

License MIT + file LICENSE

Encoding UTF-8

LazyData true

biocViews BatchEffect, RNASeq, GeneExpression, Normalization, Preprocessing, QualityControl, StatisticalMethod, GeneTarget, Clustering

RoxygenNote 6.0.1

Suggests BiocStyle, knitr, rmarkdown, DESeq2, pcaExplorer, testthat, ggplot2

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`adj2vector` *Transform the adjacency matrix to a vector*

Description

Transform the adjacency matrix to a vector

Usage

```
adj2vector(Adjacency, n)
```

Arguments

`Adjacency`	the adjacency matrix of factor
`n`	number of samples

Value

`w` the vector of the adjacency matrix

Author(s)

Haidong Yi, Ayush T. Raman

Examples

```
W <- matrix(c(0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0),nrow=4)
w <- adj2vector(W,4)
```

`DASC` *Detecting hidden batch factors through data adaptive shrinkage and clustering (DASC)*

Description

Detecting hidden batch factors through data adaptive shrinkage and clustering (DASC)

Batch factor detection via DASC (Data-adaptive Shrinkage and Clustering-DASC)

Usage

```
DASC(edata, pdata, factor, method = "ama", type = 3, lambda, rank, nrun,
  spanning = FALSE, annotation)
```

Arguments

edata	the normalized target matrix, a data.frame The row is gene, the column is sample
pdata	Phenotypic data summarizes information about the samples
factor	A factor vector which controls the convex clustering
method	Algorithm to use: 'admm' or 'ama'
type	An integer indicating the norm used: 1 = 1-norm 2 = 2-norm 3 = 2-norm^2
lambda	A double number A regularization parameter in the convex optimization
rank	integer sequence
nrun	the iteration numbers of Semi-NMF
spanning	parameter is assigned as false
annotation	An annotation of the dataset

Details

The DASC function is the main function of our algorithm DASC (Data-adaptive Shrinkage and Clustering-DASC) package. The DASC includes two main steps

- Data-adaptive shrinkage using convex clustering shrinkage (Implemented by convex optimization.);
- Extract batch factors using matrix factorization.

Value

outputs the result of semi-NMF. It classifies each sample to its batch factor.

Author(s)

Haidong Yi, Ayush T. Raman

Haidong Yi, Ayush T. Raman

See Also

cvxclust_path_ama and cvxclust_path_admm for the detailed algorithm

Examples

```
library(NMF)
library(cvxclustr)
library(Biobase)
dat <- data.frame(matrix(rnbinom(n=200, mu=100, size=1/0.5), ncol=4))
pdat <- data.frame(sample = colnames(dat), type = c(rep('A',2), rep('B',2)))
rownames(pdat) <- colnames(dat)
res <- DASC(edata=dat, pdata=pdat, factor=pdat$type, method='ama', type=3,
lambda = 1, rank = 2, nrun = 50, spanning = FALSE,
annotation='simulated dataset')
```

get_father *Representing node in this subtype*

Description

Representing node in this subtype

Usage

```
get_father(v, X)
```

Arguments

v	the index of the node
X	the saved vector with the information of the parent of every node

Value

r the parent index of the node

Author(s)

Haidong Yi, Ayush T. Raman

Examples

```
nodes <- c(2,3,4,4)
get_father(2, nodes)
```

Ini_SemiNMF *Initialization of the semi-NMF*

Description

Initialization of the semi-NMF

Usage

```
Ini_SemiNMF(model, target)
```

Arguments

model	Object of class: NMFfit
target	gene expression matrix

Value

model The initial objective of class: NMFfit

Author(s)

Haidong Yi, Ayush T. Raman

`Loss_Fro` *Get the error of Semi-NMF using frobenius norm*

Description

Get the error of Semi-NMF using frobenius norm

Usage

```
Loss_Fro(model, target)
```

Arguments

`model`	Object of class: NMFfit
`target`	gene expression matrix

Details

This is a customerized function defined in terms of `nmf`. For more information, please go through the NMF vignette https://cran.r-project.org/web/packages/NMF/vignettes/NMF-vignette.pdf

Value

The result of semi-NMF for the current iteration

Author(s)

Haidong Yi, Ayush T. Raman

`merge` *Combine two trees into one*

Description

Combine two trees into one

Usage

```
merge(x, y, X)
```

Arguments

`x`	the index of the node
`y`	the index of the node
`X`	the saved vector with the information of the parent of every node

Details

During the traversal of the graph matrix, merge function joins two disjoint sets into a single subset. It is a union step of Disjoint-set algorithm by Bernard A. Galler and Michael J. Fischer. For further details, please refer to: https://en.wikipedia.org/wiki/Disjoint-set_data_structure

Value

X A updated X vector with updates on father of every node

Author(s)

Haidong Yi, Ayush T. Raman

Semi_NMF *Main function of semi-NMF*

Description

Main function of semi-NMF

Usage

```
Semi_NMF(target, model, iternum = 100)
```

Arguments

target	gene expression matrix
model	Object of class: NMFfit
iternum	Number of iterations

Value

model the result from the semi-NMF algorithm

Author(s)

Haidong Yi, Ayush T. Raman

Sptree *Get Spanning tree from adjacency matrix*

Description

Get Spanning tree from adjacency matrix

Usage

```
Sptree(ADJ)
```

Arguments

ADJ	the adjacency matrix of the factor

Value

ADJ the spaning tree of the adjacency matrix

Author(s)

Haidong Yi, Ayush T. Raman

Examples

```
W <- matrix(c(0,1,1,1,1,0,1,1,1,1,0,1,1,1,1,0), nrow=4)
Sptree(W)
```

stanfordData *Stanford RNA-Seq dataset*

Description

Stanford RNA-Seq dataset

Format

A list with two components.

Stanford gene expression dataset (Chen et. al. PNAS, 2015; Gilad et. al. F1000 Research, 2015). It is a filtered raw counts dataset which was published by Gilad et al. F1000 Research. 30 expression & mitochondrial genes were removed (Gilad et al.F1000 Research)

rawCounts is a DataFrame object where each column represents a sample and each row represents a gene.

metadata is a metadata information about each samples

Source

https://f1000research.com/articles/4-121/v1

References

1. Gilad Y and Mizrahi-Man O. A reanalysis of mouse ENCODE comparative gene expression data. F1000Research (2015)

2. in S, Lin Y, Nery JR, et al.: Comparison of the transcriptional landscapes between human and mouse tissues. Proc Natl Acad Sci USA (2014)

`trans_ADJ` *Outputs Adjacency matrix from the factor vector*

Description

Outputs Adjacency matrix from the factor vector

Usage

```
trans_ADJ(col_data)
```

Arguments

`col_data`	A factor vector

Value

`adjacency` Adjacency matrix of `col_data`

Author(s)

Haidong Yi, Ayush T. Raman

Examples

```
batch.factor <- c(rep('human',13),rep('mouse',13))
batch.factor <- as.factor(batch.factor)
adj <- trans_Laplace(batch.factor)
```

`trans_Laplace` *Get Laplace matrix from factor vector*

Description

Get Laplace matrix from factor vector

Usage

```
trans_Laplace(col_data)
```

Arguments

`col_data`	A factor vector

Value

adjacency The Laplace matrix of col_data

Author(s)

Haidong Yi, Ayush T. Raman

Examples

```
batch.factor <- c(rep('human',13),rep('mouse',13))
batch.factor <- as.factor(batch.factor)
adj <- trans_Laplace(batch.factor)
```

---

update_G *Update G in Semi-NMF*

Description

Update G in Semi-NMF

Usage

```
update_G(X, mf, mg)
```

Arguments

X	Data expression matrix need to be factorized
mf	The basis matrix
mg	The co-efficient matrix

Details

By definition, G is a graph adjacency matrix. The update_G updates G after every iteration.

Value

G The basis matrix

Author(s)

Haidong Yi, Ayush T. Raman

Examples

```
X <- matrix(1:12,nrow=4)
mf <- matrix(1:8,nrow=4)
mg <- matrix(1:6,ncol=2)
mg <- update_G(X,mf,mg)
```

Index