
R topics documented:

- DNMF
- ndNMF
- NMFpval
- rnk
Description

Discriminant Non-Negative Matrix Factorization, DNMF, is to extend the Non-negative Matrix Factorization algorithm in order to extract features that enforce not only the spatial locality, but also the separability between classes in a discriminant manner.

Usage

\[
\text{DNMF}(\text{data, trainlabel, } r = 2, \ \text{gamma} = 0.1, \ \text{delta} = 1e-04, \\
\text{maxIter} = 1000, \ \text{tol} = 1e-07, \ \text{log} = \text{TRUE}, \ \text{plotit} = \text{FALSE}, \\
\text{checkH} = \text{TRUE}, \ ...) \]

Arguments

- **data**: a matrix, like expression profilings of some samples. the columns are samples and the rows are gene’s expression.
- **trainlabel**: a numeric vector of sample type of all the samples, this vector should ONLY contain 1 and 2 so far and length of it should equal the column (sample) size of data.
- **r**: the dimension of expected reduction dimension, with the default value 2.
- **gamma**: the tradeoff value for the within scatter matrix, with the default value 0.1.
- **delta**: the tradeoff value for the between scatter matrix, with the default value 1e-4.
- **maxIter**: the maximum iteration of update rules, with the default value 1000.
- **tol**: the toleration of coverange, with the default value 1e-7.
- **log**: log2 data. Default is TRUE.
- **plotit**: whether plot H (V=WH). Default: FALSE.
- **checkH**: whether or not check H. Default: TRUE. This parameter aims to check whether or not the H satisfy the discriminant metagenes. Usually, this should be TRUE.
- **...**: to gplots::heatmap.2

Details

The main algorithm is based on Zafeiriou, S., et al. (2006) Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification, IEEE transactions on neural networks, 17, 683-695, with some CORRECTIONs.

Author(s)

Zhilong Jia and Xiang Zhang
Examples

```r
dat <- rbind(matrix(c(rep(3, 16), rep(8, 24)), ncol=5),
matrix(c(rep(5, 16), rep(5, 24)), ncol=5),
matrix(c(rep(18, 16), rep(7, 24)), ncol=5) +
matrix(runif(120,-1,1), ncol=5)
trainlabel <- c(1,1,2,2,2)

DNMF_result <- DNMF(dat, trainlabel, r=2)
```

## Not run:
# Gene ranking. dat is the raw read count matrix with sample in column.

```r
#normalising dat
Sizefactors <- DESeq::estimateSizeFactorsForMatrix(dat)
dat = sweep(dat, 2, Sizefactors, '/')

res <- DNMF(dat, trainlabel, r=2)
rnk <- res$rnk

#The end of gene ranking examples
#Other examples
DNMF_result <- DNMF(dat, trainlabel, r=2, gamma=0.1, delta=0.0001, plotit=TRUE)
## End(Not run)
```

ndNMF

*a new discriminant Non-Negative Matrix Factorization (dNMF)*

Description

The ndNMF algorithm with the additional Fisher criterion on the cost function of conventional
NMF was designed to increase class-related discriminating power.

Usage

```r
ndNMF(dat, trainlabel, r = 2, lambada = 0.1, maxIter = 1000,
tol = 1e-07, log = TRUE, plotit = FALSE, verbose = FALSE, ...)
```

Arguments

dat a matrix with gene in row and sample in column

trainlabel the label of sample, like c(1,1,2,2,2)

r the dimension of expected reduction dimension, with the default value 2

lambada a relative weighting factor for the discriminant. Default 0.1

maxIter the maximum iteration of update rules, with the default value 1000
tol the toleration of coverange, with the default value 1e-7
log log2 data. Default is TRUE.
plotit whether plot H (V=WH). Default: FALSE.
verbose TRUE
... to gplots::heatmap.2

Details

This algorithm is based on articles.


Author(s)

Zhilong Jia and Xiang Zhang

Examples

dat <- rbind(matrix(c(rep(3, 16), rep(8, 24)), ncol=5),
matrix(c(rep(5, 16), rep(5, 24)), ncol=5),
matrix(c(rep(18, 16), rep(7, 24)), ncol=5)) +
matrix(runif(120, -1, 1), ncol=5)
trainlabel <- c(1,1,2,2,2)
res <- ndNMF(dat, trainlabel, r=2, lambada = 0.1)
res$H
res$rnk

nmfpval $ P value for discriminant Non-Negative Matrix Factorization

Description

Estimate the significance of differentially expressed genes in parallel.

Usage

```
nmfpval(nmf_res, np = 100, ncores = parallel::detectCores(), fdr = FALSE, top = 1000, verbose = FALSE)```
Arguments

- **nmf_res**: result from DNMF or dNMF
- **np**: number of permutations
- **ncores**: cores used. Default is all the available cores
- **fdr**: false discovery rate. Default is FALSE
- **top**: only include top ranked genes. Default is 1000
- **verbose**: verbose

Details


Value

A matrix with columns rnk, p (and fdr)

Author(s)

Zhilong Jia

Examples

```r
dat <- rbind(matrix(c(rep(3, 16), rep(8, 24)), ncol=5),
             matrix(c(rep(5, 16), rep(5, 24)), ncol=5),
             matrix(c(rep(18, 16), rep(7, 24)), ncol=5) +
             matrix(runif(120,-1,1), ncol=5)
trainlabel <- c(1,1,2,2,2)

nmf_res <- ndnmf(dat, trainlabel, r=2, lambada = 0.1)
pmat <- nmfpval(nmf_res, np=10, ncores=2, top=4)
```

Usage

```r
rnk(object, fn = "./tmp.rnk", type = "o2m")
```
Arguments

- **object**: a returned object of function DNMF
- **fn**: the output filename. Default is "/tmp.rnk"
- **type**: type o2m (Default) or o2o. to compare with multi sample labels. o2m means one Vs others, while o2o means one Vs another one.

Examples

```r
## Not run:
rnk(dnmf_result, fn="tmp.rnk")

## End(Not run)
```
Index

DNMF, 2

ndNMF, 3

NMFpval, 4

rnk, 5