Package 'DNMF'

June 9, 2015

Version 1.3

Date 2015-06-09

Title Discriminant Non-Negative Matrix Factorization

Description Discriminant Non-Negative Matrix Factorization aims 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. It refers to three article, Zafeiriou, Stefanos, et al. ``Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification." Neural Networks, IEEE Transactions on 17.3 (2006): 683-695. Kim, Bo-Kyeong, and Soo-Young Lee. ``Spectral Feature Extraction Using dNMF for Emotion Recognition in Vowel Sounds." Neural Information Processing. Springer Berlin Heidelberg, 2013. and Lee, Soo-Young, Hyun-Ah Song, and Shun-ichi Amari. ``A new discriminant NMF algorithm and its application to the extraction of subtle emotional differences in speech." Cognitive neurodynamics 6.6 (2012): 525-535.

Depends foreach

Imports Matrix, gplots, parallel, doParallel

License GPL (>= 2)

LazyData true

URL https://github.com/zhilongjia/DNMF

BugReports https://github.com/zhilongjia/DNMF/issues

NeedsCompilation no

Author Zhilong Jia [aut, cre],
Xiang Zhang [aut]

Maintainer Zhilong Jia <zhilongjia@gmail.com>

Repository CRAN

Date/Publication 2015-06-09 21:29:09

R topics documented:

DNMF *Discriminant Non-Negative Matrix Factorization.*

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

```
DNMF(data, trainlabel, r = 2, gamma = 0.1, delta = 1e-04,
  maxIter = 1000, tol = 1e-07, log = TRUE, plotit = FALSE,
  checkH = 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 safisfy 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

```
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 maatrix with sample in column.

#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 exmaples

#Other exmaples
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

```
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.

1. Kim, Bo-Kyeong, and Soo-Young Lee. "Spectral Feature Extraction Using dNMF for Emotion Recognition in Vowel Sounds." Neural Information Processing. Springer Berlin Heidelberg, 2013.
2. Lee, Soo-Young, Hyun-Ah Song, and Shun-ichi Amari. "A new discriminant NMF algorithm and its application to the extraction of subtle emotional differences in speech." Cognitive neurodynamics 6.6 (2012): 525-535.

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 availiable cores
fdr	false discovery rate. Default is FALSE
top	only include top ranked genes. Default is 1000
verbose	verbose

Details

P value is caculated based on aatricle, Wang, Hong-Qiang, Chun-Hou Zheng, and Xing-Ming Zhao. "jNMFMA: a joint non-negative matrix factorization meta-analysis of transcriptomics data." Bioinformatics (2014): btu679.

Value

a matrix with columns rnk, p (and fdr)

Author(s)

Zhilong Jia

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)

nmf_res <- ndNMF(dat, trainlabel, r=2, lambada = 0.1)
pMat <- NMFpval(nmf_res, np=10, ncores=2, top=4)
```

---

rnk *write rnk to a file from matrix W.*

Description

write a rnk file from matrix W in a returned object of function DNMF. The rnk format is referred RNK

Usage

```
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

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

## End(Not run)
```

Index