Package 'hNMF'

June 7, 2017

Title Hierarchical Non-Negative Matrix Factorization

Version 0.3

Author Nicolas Sauwen

Maintainer Nicolas Sauwen <nicolas.sauwen@openanalytics.eu>

Description Hierarchical non-negative matrix factorization
for tumor segmentation based on multi-parametric MRI data.
Several NMF algorithms are available.

Depends R (>= 3.3.2)

License GPL-3

Encoding UTF-8

LazyData true

Imports NMF, oro.nifti, tcltk, nnls, R.matlab, spatialfil,
rasterImage, stats, graphics, grDevices

RoxygenNote 6.0.0

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2017-06-07 17:20:48 UTC

R topics documented:

`divergenceCheck` *This function performs a divergence check, by comparing the current NMF sources with the initial ones. 3 divergence criteria are implemented.*

Description

This function performs a divergence check, by comparing the current NMF sources with the initial ones. 3 divergence criteria are implemented.

Usage

```
divergenceCheck(W, W0)
```

Arguments

`W`	Current NMF source matrix
`W0`	Initial NMF source matrix

Value

Boolean value, indicating whether or not one of the divergence criteria has been reached

Author(s)

Nicolas Sauwen

`HALSacc` *Accelerated hierarchical alternating least squares NMF. For a reference to the method, see N. Gillis, Nonnegative matrix factorization: complexity, algorithms and applications [Section 4.2, Algo. 6], PhD thesis, Université catholique de Louvain, February 2011.*

Description

Accelerated hierarchical alternating least squares NMF. For a reference to the method, see N. Gillis, Nonnegative matrix factorization: complexity, algorithms and applications [Section 4.2, Algo. 6], PhD thesis, Université catholique de Louvain, February 2011.

Usage

```
HALSacc(X, nmfMod, alpha = 1, maxiter = 1000, checkDivergence = FALSE)
```

Arguments

X	Input data matrix, each column represents one data point and the rows correspond to the different features
nmfMod	Valid NMF model, containing initialized factor matrices (in accordance with the NMF package definition)
alpha	Nonnegative parameter of the accelerated method
maxiter	Maximum number of iterations
checkDivergence	currently not in use, to be implemented

Value

Resulting NMF model (in accordance with the NMF package definition)

Author(s)

nsauwen

HALSupdt *Updating step for accelerated HALS NMF*

Description

Updating step for accelerated HALS NMF

Usage

```
HALSupdt(M, V, UtU, UtM, alpha, delta)
```

Arguments

M	Input data matrix
V	Factor matrix to be updated
UtU	Product of the other transposed factor matrix with itself
UtM	Product of the other transposed factor matrix with the input matrix
alpha	Nonnegative parameter of the accelerated method
delta	Convergence parameter

Value

Updated factor matrix V

Author(s)

Nicolas Sauwen

hNMF *Hierarchical non-negative matrix factorization.*

Description

Hierarchical non-negative matrix factorization.

Usage

```
hNMF(nmfInput, nmfMethod = "HALSacc")
```

Arguments

nmfInput	List with NMF input attributes
nmfMethod	String referring to the NMF algorithm to be used.

Value

Resulting NMF model (in accordance with NMF package definition)

Author(s)

Nicolas Sauwen

imoverlay *Overlay a mask or a color scaled image on top of a background image*

Description

Overlay a mask or a color scaled image on top of a background image

Usage

```
imoverlay(image, overlay, selectVect = NULL, color = c(0, 1, 0))
```

Arguments

image	A matrix, background image
overlay	A matrix, serving as the overlay mask or figure
selectVect	A matrix (binary values), specifying which matrix elements are to be overlaid
color	3-element vector, defining the RGB color to be used in case the overlay is a mask

Author(s)

Nicolas Sauwen

`importMatlabMriData` *Interactive method to select Matlab (.mat) input dataset to perform (h)NMF analyses*

Description

Interactive method to select Matlab (.mat) input dataset to perform (h)NMF analyses

Usage

```
importMatlabMriData()
```

Value

List of input data attributes

Author(s)

Nicolas Sauwen

`initializeSPA` *The successive projection algorithm, a useful method for initializing the NMF source matrix*

Description

The successive projection algorithm, a useful method for initializing the NMF source matrix

Usage

```
initializeSPA(data, nSources)
```

Arguments

`data`	Input data matrix. The columns correspond to the data points, each row represents one feature
`nSources`	Number of sources to be obtained

Value

Matrix with initialized sources as its columns

Author(s)

Nicolas Sauwen

`nlsSubProb` *Algorithm for solving convex non-negative least squares subproblem using projected gradients*

Description

Algorithm for solving convex non-negative least squares subproblem using projected gradients

Usage

```
nlsSubProb(X, W, Hinit, tol, maxIter)
```

Arguments

`X`	Input data matrix
`W`	NMF basis matrix
`Hinit`	Initial NMF coef matrix
`tol`	Tolerance for a relative stopping condition
`maxIter`	Maximum number of iterations

Value

List containing updated H matrix, its gradient and number of iterations

Author(s)

Nicolas Sauwen

`PGNMF` *NMF by alternating non-negative least squares using projected gradients. For a reference to the method, see C.-J. Lin, "Projected Gradient Methods for Non-negative Matrix Factorization", Neural computation 19.10 (2007): 2756-2779.*

Description

NMF by alternating non-negative least squares using projected gradients. For a reference to the method, see C.-J. Lin, "Projected Gradient Methods for Non-negative Matrix Factorization", Neural computation 19.10 (2007): 2756-2779.

Usage

```
PGNMF(X, nmfMod, tol = 1e-05, maxIter = 500, timeLimit = 300,
  checkDivergence = TRUE)
```

Arguments

X	Input data matrix, each column represents one data point and the rows correspond to the different features
nmfMod	Valid NMF model, containing initialized factor matrices (in accordance with the NMF package definition)
tol	Tolerance for a relative stopping condition
maxIter	Maximum number of iterations
timeLimit	Limit of time duration NMF analysis
checkDivergence	Boolean indicating whether divergence checking should be performed Default is TRUE, but it should be set to FALSE when using random initialization

Value

Resulting NMF model (in accordance with the NMF package definition)

Author(s)

nsauwen

---

validateNMFResult *Validation tool for (h)NMF based segmentation.*

Description

This function converts NMF abundance maps to actual segmentation results, and validates the result with respect to manual segmentation. The manual segmentation results are provided in nifti (.nii) format. Segmentation overlap is quantified by means of the Dice-score. Input to the method is the NMF input data and a valid nmfModel object obtained from the NMF package. sliceRange must be a 2-element vector specifying the analyzed slice range.

Usage

```
validateNMFResult(nmfInput, nmfModel, sliceRange, indTumor = 0,
  indNecrosis = 0, indEdema = 0, tumorNiftiFile = NULL,
  necrosisNiftiFile = NULL, edemaNiftiFile = NULL)
```

Arguments

nmfInput	List with NMF input attributes
nmfModel	NMF model (in accordance with NMF package definition)
sliceRange	2-element vector containing first and last image slice number that were analyzed
indTumor, indNecrosis, indEdema	Indices of the NMF sources representing tumor, necrosis and edema, respectively

`tumorNiftiFile, necrosisNiftiFile, edemaNiftiFile`
: Character strings containing the path to the manual segmentation files (in nifti format) for tumor, necrosis and edema

Value

Dice scores quantifying the segmentation quality are printed in the R console

Author(s)

Nicolas Sauwen

`viewNMFResult` *Visualization of (h)NMF results*

Description

(h)NMF abundance maps are plotted overlaid on top of a background (input) image

Usage

```
viewNMFResult(nmfInput, nmfModel)
```

Arguments

`nmfInput`	List with NMF input attributes
`nmfModel`	NMF model (in accordance with NMF package definition)

Author(s)

Nicolas Sauwen

Index