Package 'MutExMatSorting'

August 2, 2021

Type Package

Title Sort Rows and Columns of a Binary Matrix in a Way that the Patterns of Non-Null Entries Have a Minimal Overlap Across Rows

Version 0.1.0

Author Alessandro Vinceti

Maintainer Alessandro Vinceti <alessandro.vinceti@fht.org>

Description Heuristic algorithm that takes in input a sparse binary matrix and sorts its rows and columns in a way that the patterns of non-null entries have a minimal overlap across rows. This highlights possible mutual exclusive trends among these patterns.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends pheatmap

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`MExMaS.HeuristicMutExSorting`

Minimal overlap sorting

Description

This function implements an heuristic algrorithm that takes in input a sparse binary matrix and sorts its rows and column in a way that the patterns of non null entries have a minimal overalp across rows.

Usage

```
MExMaS.HeuristicMutExSorting(mutPatterns, display = TRUE)
```

Arguments

mutPatterns	numeric binary matrix of the values to be sorted.
display	boolean, whether to display the original and sorted matrices. Default is true.

Examples

```
library(pheatmap)

# Generating a random binary matrix with row and column names
r <- 100
c <- 100
dens <- 0.10
mutPatterns <- matrix(0, r, c,dimnames = list(paste('row',1:r,sep=''),paste('col',1:c,sep='')))
mutPatterns[sample(r*c,round(r*c*dens))] <- 1

# Executing mutual exclusivity sorting
sortedMat <- MExMaS.HeuristicMutExSorting(mutPatterns)
```

MExMaS.Memo *OncoPrint sorting*

Description

This function implements the sorting algorithm derived from Memo. It takes in input a sparse binary matrix and sorts its rows according to the numbers of non-null entries, whereas columns are sorted through a weighted scoring based on rows ordering.

Usage

```
MExMaS.Memo(mutPatterns, display = TRUE)
```

Arguments

mutPatterns	numeric binary matrix of the values to be sorted.
display	boolean, whether to display the original and sorted matrices. Default is true.

Examples

```
library(pheatmap)

# Generating a random binary matrix with row and column names
r <- 100
c <- 100
dens <- 0.10
mutPatterns <- matrix(0, r, c,dimnames = list(paste('row',1:r,sep=''),paste('col',1:c,sep='')))
mutPatterns[sample(r*c,round(r*c*dens))] <- 1
```

```
# Executing mutual exclusivity sorting
sortedMat <- MExMaS.Memo(mutPatterns)
```

Index