

Package ‘SPUTNIK’

April 16, 2024

Type Package

Title Spatially Automatic Denoising for Imaging Mass Spectrometry Toolkit

Version 1.4.2

Author Paolo Inglese [aut, cre],
Goncalo Correia [aut, ctb]

Maintainer Paolo Inglese <p.inglese@outlook.com>

Description Set of tools for peak filtering of mass spectrometry imaging data based on spatial distribution of signal. Given a region-of-interest, representing the spatial region where the informative signal is expected to be localized, a series of filters determine which peak signals are characterized by an implausible spatial distribution. The filters reduce the dataset dimension and increase its information vs noise ratio, improving the quality of the unsupervised analysis results, reducing data dimension and simplifying the chemical interpretation.
The methods are described in Inglese P. et al (2019) <[doi:10.1093/bioinformatics/bty622](https://doi.org/10.1093/bioinformatics/bty622)>.

Depends R (>= 3.4.0)

License GPL (>= 3)

Encoding UTF-8

Imports e1071, edgeR, spatstat.explore, spatstat.geom, viridis,
ggplot2, reshape, imager, methods, infotheo, parallel, irlba,
doSNOW, foreach

Suggests testthat

RoxygenNote 7.3.1

URL <https://github.com/paoloinglese/SPUTNIK>

BugReports <https://github.com/paoloinglese/SPUTNIK/issues>

NeedsCompilation no

Repository CRAN

Date/Publication 2024-04-16 14:30:05 UTC

R topics documented:

applyPeaksFilter,msi.dataset-method	3
binKmeans,msi.dataset-method	4
binKmeans2,msi.dataset-method	5
binOtsu,ms.image-method	6
binSupervised,msi.dataset-method	7
bladderMALDIRompp2010	8
closeImage,ms.image-method	8
countPixelsFilter	9
createPeaksFilter	11
CSRPeaksFilter	12
getIntensityMat,msi.dataset-method	14
getMZ,msi.dataset-method	15
getShapeMSI,msi.dataset-method	16
gini.index	17
globalPeaksFilter	18
invertImage,ms.image-method	20
ms.image-class	21
msi.dataset-class	21
msiDataset	22
msImage	23
NMI	24
normIntensity,msi.dataset-method	24
numDetectedMSI,msi.dataset-method	26
ovarianDESIDoria2016	27
PCAImage,msi.dataset-method	27
plot,ms.image,missing-method	28
refImageBinaryKmeans	29
refImageBinaryKmeansMulti	29
refImageBinaryOtsu	30
refImageBinarySVM	31
refImageContinuous	32
removeSmallObjects,ms.image-method	34
scatter.ratio	35
smoothImage,ms.image-method	36
spatial.chaos	37
splitPeaksFilter	38
SSIM	40
totalIonCountMSI,msi.dataset-method	41
varTransform,msi.dataset-method	41

`applyPeaksFilter,msi.dataset-method`*Apply the results of a peaks filter.*

Description

`applyPeaksFilter` select the peaks returned by a peak filter. Custom filters can be created passing a named array of selected peak indices to `createPeaksFilter`. Names correspond to the m/z values of the selected peaks and must coincide with those of the MS dataset.

Usage

```
## S4 method for signature 'msi.dataset'  
applyPeaksFilter(object, peakFilter)
```

Arguments

`object` [msi.dataset-class](#) object.
`peakFilter` peaks filter results.

Value

[msi.dataset-class](#) object with only selected peaks.

Examples

```
## Load package  
library("SPUTNIK")  
  
## Mass spectrometry intensity matrix  
X <- matrix(rnorm(16000), 400, 40)  
X[X < 0] <- 0  
  
## Print original dimensions  
print(dim(X))  
  
## m/z vector  
mzVector <- seq(600, 900, by = (900 - 600) / 39)  
  
## Read the image size  
imSize <- c(20, 20)  
  
## Construct the msi.dataset object  
msiX <- msiDataset(X, mzVector, imSize[1], imSize[2])  
  
## Calculate the p-values using the Clark Evans test, then apply Benjamini-  
## Hochberg correction.  
csr <- CSRPeaksFilter(  
  msiData = msiX, method = "ClarkEvans",
```

```

    calculateCovariate = FALSE, adjMethod = "BH"
  )

  ## Print selected peaks
  print(csr$q.value)

  ## Create a new filter selecting corrected p-values < 0.001
  selIdx <- which(csr$q.value < 0.001)
  csrFilter <- createPeaksFilter(selIdx)

```

binKmeans,msi.dataset-method

Return a binary mask generated applying k-means clustering on first 10 principal components of peaks intensities.

Description

Return a binary mask generated applying k-means clustering on first 10 principal components of peaks intensities.

Usage

```

## S4 method for signature 'msi.dataset'
binKmeans(object, ref = "detected", invert = FALSE, npcs = 10)

```

Arguments

object	msi.dataset-class object
ref	string (default = "detected"). Sample reference image used to align the clusters.
invert	boolean (default = FALSE). If FALSE, the clusters are inversely aligned to the sample reference image.
npcs	int (default = 10). Number of principal components to calculate.

Value

[ms.image-class](#) object representing the binary mask image.

Examples

```

## Load package
library("SPUTNIK")

## Create the msi.dataset-class object
sz <- c(5, 4)
x <- matrix(rnorm(sz[1] * sz[2] * 20), sz[1] * sz[2], 20)
x[x < 0] <- 0
mz <- sort(sample(100, ncol(x)))
msiX <- msiDataset(x, mz, sz[1], sz[2])

```

```
## Generate binary mask by applying k-means on the entire dataset
roiImg <- refImageBinaryKmeans(msiX, npcs = 3)

## Plot the mask
# plot(roiImg)
```

```
binKmeans2,msi.dataset-method
```

Return a binary mask generated applying k-means clustering on peaks intensities. A finer segmentation is obtained by using a larger number of clusters than 2. The off-sample clusters are merged looking at the most frequent labels in the image corners. The lookup areas are defined by the kernel size.

Description

Return a binary mask generated applying k-means clustering on peaks intensities. A finer segmentation is obtained by using a larger number of clusters than 2. The off-sample clusters are merged looking at the most frequent labels in the image corners. The lookup areas are defined by the kernel size.

Usage

```
## S4 method for signature 'msi.dataset'
binKmeans2(
  object,
  mzQuery = numeric(),
  useFullMZ = TRUE,
  mzTolerance = Inf,
  numClusters = 4,
  kernelSize = c(3, 3, 3, 3),
  numCores = 1,
  verbose = TRUE
)
```

Arguments

object	msi.dataset-class object
mzQuery	numeric. Values of m/z used to calculate the reference image. 2 values are interpreted as interval, multiple or single values are searched in the m/z vector. It should be left unset when using useFullMZRef = TRUE.
useFullMZ	logical (default = 'TRUE'). Whether all the peaks should be used to calculate the reference image.
mzTolerance	numeric (default = Inf). Tolerance in PPM to match the mzQueryRef. values in the m/z vector. It overrides useFullMZ.

numClusters	numeric (default = 4). Number of k-means clusters.
kernelSize	4D array (default = c(3, 3, 3, 3)). Array of sizes in pixels of the corner kernels used to identify the off-sample clusters. The elements represent the size of the top-left, top-right, bottom-right and bottom-left corners. A negative value can be used to skip the corresponding corner.
numCores	(default = 1). Multi-core parallel computation of k-means. Each core corresponds to a repetition of k-means. If numCores = 1, a serial k-means with 5 repetitions is performed.
verbose	logical (default = 'TRUE'). Show additional output.

Value

[ms.image-class](#) object representing the binary mask image.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

binOtsu,ms.image-method

Binarize MS image using Otsu's thresholding.

Description

Binarize MS image using Otsu's thresholding.

Usage

```
## S4 method for signature 'ms.image'
binOtsu(object)
```

Arguments

object [ms.image-class](#) object. See [msImage](#).

Value

[ms.image-class](#) object with binary intensities.

Examples

```
## Load package
library("SPUTNIK")

## Create ms.image-class object
msIm <- msImage(values = matrix(rnorm(200), 40, 50), name = "test", scale = TRUE)

## Generate binary image
binIm <- refImageBinaryOtsu(msIm)
```

binSupervised,msi.dataset-method

Return a binary mask generated applying a supervised classifier on peaks intensities using manually selected regions corresponding to off-sample and sample-related areas.

Description

Return a binary mask generated applying a supervised classifier on peaks intensities using manually selected regions corresponding to off-sample and sample-related areas.

Usage

```
## S4 method for signature 'msi.dataset'  
binSupervised(  
  object,  
  refImage,  
  mzQuery = numeric(),  
  mzTolerance = Inf,  
  useFullMZ = TRUE,  
  method = "svm",  
  verbose = TRUE  
)
```

Arguments

object	msi.dataset-class object
refImage	ms.image-class object. Image used as reference to manually select the ROI pixels.
mzQuery	numeric. Values of m/z used to calculate the reference image. 2 values are interpreted as interval, multiple or single values are searched in the m/z vector. It overrides useFullMZ.
mzTolerance	numeric (default = Inf). Tolerance in PPM to match the mzQueryRef. values in the m/z vector. It overrides useFullMZ.
useFullMZ	logical (default = 'TRUE'). Whether all the peaks should be used to perform the supervised segmentation.
method	string (default = 'svm'). Supervised classifier used to segment the ROI.
verbose	boolean (default = 'TRUE'). Additional output.

Value

[ms.image-class](#) object representing the binary mask image.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

bladderMALDIRompp2010 *Load the example MALDI-MSI data.*

Description

Loads a single mouse urinary bladder MALDI mass spectrometry imaging dataset acquired in positive ionization mode using Thermo qExactive Orbitrap. The dataset is available at "https://raw.github.com/paoloinglese/SPUT". The dataset is loaded in the R environment under the variable name `malDiData`.

Usage

```
bladderMALDIRompp2010(verbose = TRUE)
```

Arguments

`verbose` Logical (default = TRUE). Show additional output text.

Value

`desiData` MS intensity matrix. Rows represent pixels, columns represent matched peaks.

References

Rompp, A., Guenther, S., Schober, Y., Schulz, O., Takats, Z., Kummer, W., & Spengler, B. (2010). Histology by mass spectrometry: label-free tissue characterization obtained from high-accuracy bioanalytical imaging. *Angewandte chemie international edition*, 49(22), 3834-3838.

closeImage,ms.image-method

Apply morphological closing to binary image.

Description

Apply morphological closing to binary image.

Usage

```
## S4 method for signature 'ms.image'
closeImage(object, kern.size = 5)
```

Arguments

`object` [ms.image-class](#) object. See [msImage](#).
`kern.size` numeric. Kernel size.

Value

[ms.image-class](#) object after closing.

Examples

```
## Load package
library("SPUTNIK")

## Create ms.image-class object
msIm <- msImage(values = matrix(rnorm(200), 40, 50), name = "test", scale = TRUE)

## Generate binary image
msImBin <- refImageBinaryOtsu(msIm)

## Apply the morphological closing
msImClosed <- closeImage(msImBin, kern.size = 3)
```

countPixelsFilter *Filter based on the minimum number of connected pixels in the ROI.*

Description

countPixelsFilter selects peaks which signals are localized in regions consisting of a minimum number of connected pixels in the ROI.

Usage

```
countPixelsFilter(
  msiData,
  roiImage,
  minNumPixels = 9,
  smoothPeakImage = FALSE,
  smoothSigma = 2,
  closePeakImage = FALSE,
  closeKernSize = 5,
  aggressive = 0,
  verbose = TRUE
)
```

Arguments

msiData [msi.dataset-class](#) object. See [msiDataset](#).

roiImage [ms.image-class](#) object representing the ROI mask. See [msImage](#).

minNumPixels integer (default = 9). Smallest number of connected pixels used to select a peak.

smoothPeakImage logical (default = FALSE). Whether the peak images should be smoothed before determining the connected components.

smoothSigma	numeric (default = 2). Standard deviation of the smoothing Gaussian kernel.
closePeakImage	logical (default = FALSE). Whether morphological closing should be applied to the binary peak images.
closeKernSize	numeric (default = 5). Kernel size for the morphological closing operation. Kernel shape is fixed to diamond.
aggressive	integer (default = 0). Defines the level of aggressiveness of the filter. See 'Details' section.
verbose	logical (default = TRUE). Additional output text.

Details

Count filter tries to determine and remove peaks which signal is scattered in a region unrelated with the expected ROI. A minimum number of connected pixels in the ROI is used to trigger the filter. This value should be carefully set equal to the geometrical size of the smallest expected informative sub-region. Each peak image is binarized using Otsu's thresholding and the connected components are extracted. The filter selects those peaks that show, within the ROI, at least one connected component of size larger or equal to `minNumPixels`. The level of aggressiveness, associated with increasing values of the parameter `aggressive`, determines whether the size of the connected components within the ROI should be compared with that of the connected components localized outside the ROI. If `aggressive = 0`, no comparison is performed. If `aggressive = 1`, the filter checks whether the max size of the connected components localized outside the ROI is smaller or equal to the maximum size of the connected components inside the ROI. If `aggressive = 2`, a stricter filter checks whether the maximum size of the connected components localized outside the ROI is smaller than `minNumPixels`. Different aggressiveness levels can produce completely different results, depending on the nature of the analyzed dataset.

Value

peak.filter object. See [applyPeaksFilter-msi.dataset-method](#).

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

See Also

`applyPeaksFilter`

Examples

```
## Load package
library("SPUTNIK")

## Mass spectrometry intensity matrix
X <- matrix(rnorm(16000), 400, 40)
X[X < 0] <- 0

## Print original dimensions
print(dim(X))
```

```
## m/z vector
mzVector <- seq(600, 900, by = (900 - 600) / 39)

## Read the image size
imSize <- c(20, 20)

## Construct the ms.dataset object
msiX <- msiDataset(X, mzVector, imSize[1], imSize[2])

## Extract the ROI using k-means

refImg <- refImageContinuous(msiX, method = "sum")
roiImg <- refImageBinaryOtsu(refImg)

## Perform count pixels filtering
count.sel <- countPixelsFilter(
  msiData = msiX, roiImage = roiImg,
  minNumPixels = 4, aggressive = 1
)

## Apply the filter
msiX <- applyPeaksFilter(msiX, count.sel)

## Print new dimensions
print(dim(getIntensityMat(msiX)))
```

createPeaksFilter *Generate a peak filter object.*

Description

[createPeaksFilter](#) returns a peak.filter object.

Usage

```
createPeaksFilter(peaksIndices)
```

Arguments

peaksIndices a named array representing the selected peaks. Names correspond to the m/z values.

Details

Function to create a custom peak that can be subsequently applied using the function [applyPeaksFilter-msi.dataset-meth](#). Argument of the function is the index vector of the selected peaks named with their m/z values. The m/z values are used to check whether the indices correspond to the common m/z values in the [msi.dataset-class](#) object.

Value

peak.filter object.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

See Also

[applyPeaksFilter-msi.dataset-method](#)

Examples

```
library("SPUTNIK")
mz <- seq(100, 195, 5)
mzIdx <- sample(100, 20)
names(mzIdx) <- mz
peaksFilter <- createPeaksFilter(mzIdx)
```

CSRPeaksFilter

Performs the peak selection based on complete spatial randomness test.

Description

CSRPeaksFilter returns the significance for the null hypothesis that the spatial distribution of the peak intensities follow a random pattern. A significant p-value (q-values can be returned after applying multiple testing correction) allows to reject the hypothesis that the spatial distribution of a peak signal is random. The tests are performed using the functions available in the statspat R package.

Usage

```
CSRPeaksFilter(  
  msiData,  
  method = "ClarkEvans",  
  covariateImage = NULL,  
  adjMethod = "bonferroni",  
  returnQvalues = TRUE,  
  plotCovariate = FALSE,  
  cores = 1,  
  verbose = TRUE,  
  ...  
)
```

Arguments

msiData	msi.dataset-class object. See msiDataset .
method	string (default = "ClarkEvans"). CSR statistical test applied to the peaks signal. Accepted values are: <ul style="list-style-type: none"> • "ClarkEvans": performs a test based on the Clark and Evans aggregation R index. This test evaluates the compares of the nearest-neighbors distances to the case of purely random pattern. • "KS": performs a test of goodness-of-fit between the signal pixels associated point process pattern and a spatial covariate using the Kolmogorov-Smirnov test. The covariate is defined by the reference image.
covariateImage	ms.image-class object. An image used as covariate (required for Kolmogorov-Smirnov test).
adjMethod	string (default = "bonferroni"). Multiple testing correction method. Possible values coincide with those of the <code>stats::p.adjust</code> function.
returnQvalues	logical (default = TRUE). Whether the computed q-values should be returned together with the p-values.
plotCovariate	logical (default = FALSE). Whether the covariate image should be visualized. Read only when <code>method = "KS"</code> .
cores	integer (default = 1). Number of CPU cores. Parallel computation if greater than 1.
verbose	logical (default = TRUE). Additional output texts are generated.
...	additional parameters compatible with the <code>statspat.core</code> functions. See cdf.test for "KS" and clarkevans.test for "ClarkEvans"

Value

List of the p-values and adjusted p-values for the CSR test.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

References

- Baddeley, A., & Turner, R. (2005). Spatstat: an R package for analyzing spatial point patterns. *Journal of statistical software*, 12(6), 1-42.
- Clark, P.J. and Evans, F.C. (1954) Distance to nearest neighbour as a measure of spatial relationships in populations. *Ecology* 35, 445–453.
- Berman, M. (1986) Testing for spatial association between a point process and another stochastic process. *Applied Statistics* 35, 54–62.

Examples

```
## Load package
library("SPUTNIK")

## Mass spectrometry intensity matrix
X <- matrix(rnorm(16000), 400, 40)
X[X < 0] <- 0

## Print original dimensions
print(dim(X))

## m/z vector
mzVector <- seq(600, 900, by = (900 - 600) / 39)

## Read the image size
imSize <- c(20, 20)

## Construct the msi.dataset object
msiX <- msiDataset(X, mzVector, imSize[1], imSize[2])

## Calculate the p-values using the Clark Evans test, then apply Benjamini-
## Hochberg correction.
csr <- CSRPeaksFilter(
  msiData = msiX, method = "ClarkEvans",
  calculateCovariate = FALSE, adjMethod = "BH"
)

## Print selected peaks
print(csr$q.value)

## Create a new filter selecting corrected p-values < 0.001
selIdx <- which(csr$q.value < 0.001)
csrFilter <- createPeaksFilter(selIdx)
```

getIntensityMat,msi.dataset-method

Return the peaks intensity matrix.

Description

Return the peaks intensity matrix.

Usage

```
## S4 method for signature 'msi.dataset'
getIntensityMat(object)
```

Arguments

object [msi.dataset-class](#) object.

Value

peaks intensity matrix. Rows represent pixels, and columns represent peaks.

Examples

```
## Load package
library("SPUTNIK")

## Create the msi.dataset-class object
sz <- c(5, 4)
x <- matrix(rnorm(sz[1] * sz[2] * 20), sz[1] * sz[2], 20)
x[x < 0] <- 0
mz <- sort(sample(100, ncol(x)))
msiX <- msiDataset(x, mz, sz[1], sz[2])

## Get m/z vector
mz <- getMZ(msiX)

## Get intensity matrix
X <- getIntensityMat(msiX)

## Get image size
sz <- getShapeMSI(msiX)
```

getMZ,msi.dataset-method

Return the m/z vector.

Description

Return the m/z vector.

Usage

```
## S4 method for signature 'msi.dataset'
getMZ(object)
```

Arguments

object [msi.dataset-class](#) object.

Value

vector containing the m/z values.

Examples

```
## Load package
library("SPUTNIK")

## Create the msi.dataset-class object
sz <- c(5, 4)
x <- matrix(rnorm(sz[1] * sz[2] * 20), sz[1] * sz[2], 20)
x[x < 0] <- 0
mz <- sort(sample(100, ncol(x)))
msiX <- msiDataset(x, mz, sz[1], sz[2])

## Get m/z vector
mz <- getMZ(msiX)

## Get intensity matrix
X <- getIntensityMat(msiX)

## Get image size
sz <- getShapeMSI(msiX)
```

getShapeMSI,msi.dataset-method

Returns the geometrical shape of MSI dataset

Description

Returns the geometrical shape of MSI dataset

Usage

```
## S4 method for signature 'msi.dataset'
getShapeMSI(object)
```

Arguments

object [msi.dataset-class](#) object.

Value

number of rows and number of columns of the MS image.

Examples

```
## Load package
library("SPUTNIK")

## Create the msi.dataset-class object
sz <- c(5, 4)
x <- matrix(rnorm(sz[1] * sz[2] * 20), sz[1] * sz[2], 20)
```



```
x[x < 0] <- 0
mz <- sort(sample(100, ncol(x)))
msiX <- msiDataset(x, mz, sz[1], sz[2])

## Get m/z vector
mz <- getMZ(msiX)

## Get intensity matrix
X <- getIntensityMat(msiX)

## Get image size
sz <- getShapeMSI(msiX)
```

gini.index

Gini index.

Description

`gini.index` returns the Gini index of the ion intensity vector as a measure of its sparseness. The intensity vector is first quantized in N levels (default = 256). A value close to 1 represents a high level of sparseness, a value close to 0 represents a low level of sparseness.

Usage

```
gini.index(x, levels = 256)
```

Arguments

`x` numeric. Peak intensity array.
`levels` numeric (default = 256). Number of levels for the peak intensity quantization.

Value

A value between 0 and 1. High levels of signal sparsity are associated with values close to 1, whereas low levels of signal sparsity are associated with values close to 0.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

References

Hurley, N., & Rickard, S. (2009). Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10), 4723-4741.

See Also

[scatter.ratio](#) [spatial.chaos](#)

Examples

```
## Load package
library("SPUTNIK")

## Image
im <- matrix(rnorm(100), 10, 10)
im[im < 0] <- 0

## Spatial chaos
sc <- spatial.chaos(im, levels = 30, morph = TRUE)
stopifnot(sc <= 1)

## Gini index
gi <- gini.index(im, levels = 16)
stopifnot(gi >= -1 && gi <= 1)

## Scatter ratio
sr <- scatter.ratio(im)
stopifnot(sr <= 1)
```

globalPeaksFilter

Reference similarity based peak selection.

Description

globalPeaksFilter returns a list of peaks selected by their similarity with a reference image.

Usage

```
globalPeaksFilter(
  msiData,
  referenceImage,
  method = "pearson",
  threshold = NULL,
  cores = 1,
  verbose = TRUE
)
```

Arguments

msiData [msi.dataset-class](#) object. See [msiDataset](#).

referenceImage [ms.image-class](#) object. Reference image used to calculate the similarity values.

method method used to calculate the similarity between the peak intensities and the reference image. Accepted values are:

- **pearson**: Pearson's correlation
- **spearman**: Spearman's correlation
- **ssim**: structural similarity index measure

- nmi: normalized mutual information.

threshold	numeric (default = 0, default = 0.001 (SSIM)). The threshold applied to the similarity values between the peaks images and the reference image. The default value of 0 guarantees that only the ions with a positive similarity with the reference image (typically representing the spatial distribution of the signal source) are retrieved. For consistency, the NMI are scaled in [-1, 1] to match the same range of correlations.
cores	integer (default = 1). Number of cores for parallel computing.
verbose	logical (default = TRUE). Additional output text.

Details

A filter based on the similarity between the peak signals and a reference signal. The reference signal, passed as an `ms.image-class` object. Both continuous and binary references can be passed. The filter then calculates the similarity between the peaks signal and the reference image and select those with a similarity larger than `threshold`. Multiple measures are available, correlation, structural similarity index measure (SSIM), and normalized mutual information (NMI). Since correlation can assume values in [-1, 1], also NMI are scaled in [-1, 1].

Value

peak.filter object. See [applyPeaksFilter](#).

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

References

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4), 600-612.

Meyer, P. E. (2009). Infotheo: information-theoretic measures. R package. Version, 1(0).

See Also

[countPixelsFilter](#) [applyPeaksFilter-msi.dataset-method](#)

Examples

```
## Load package
library("SPUTNIK")

## Mass spectrometry intensity matrix
X <- matrix(rnorm(16000), 400, 40)
X[X < 0] <- 0

## Print original dimensions
print(dim(X))
```

```
## m/z vector
mzVector <- seq(600, 900, by = (900 - 600) / 39)

## Read the image size
imSize <- c(20, 20)

## Construct the ms.dataset object
msiX <- msiDataset(X, mzVector, imSize[1], imSize[2])

## Generate the reference image and the ROI mask
refImg <- refImageContinuous(msiX, method = "sum")

## Perform global peaks filter
glob.peaks <- globalPeaksFilter(
  msiData = msiX, referenceImage = refImg,
  method = "pearson", threshold = 0
)

## Apply the filter
msiX <- applyPeaksFilter(msiX, glob.peaks)

## Print the new dimensions
print(dim(getIntensityMat(msiX)))
```

`invertImage,ms.image-method`

Invert the colors of an MS image.

Description

Invert the colors of an MS image.

Usage

```
## S4 method for signature 'ms.image'
invertImage(object)
```

Arguments

`object` [ms.image-class](#) object. See [msImage](#).

Value

[ms.image-class](#) object after inverting colors.

Examples

```
## Load package
library("SPUTNIK")

## Create ms.image-class object
msIm <- msImage(values = matrix(rnorm(200), 40, 50), name = "test", scale = TRUE)

## Invert the colors
msImInverted <- invertImage(msIm)
```

ms.image-class [ms.image-class definition.](#)

Description

[ms.image-class](#) definition.

Slots

values numeric 2-D matrix representing the pixel intensity values.

name string. Image name used for plotting.

scaled logical. Whether the pixels intensities have been scaled in [0, 1] or not.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

msi.dataset-class [msi.dataset-class](#) S4 class definition containing the information about the mass spectrometry imaging dataset.

Description

[msi.dataset-class](#) S4 class definition containing the information about the mass spectrometry imaging dataset.

Slots

matrix the peaks intensity matrix. Rows represent pixels, and columns represent peaks.

mz vector of matched m/z values.

nrow geometrical shape (number of rows) of image.

ncol geometrical shape (number of columns) of image.

norm normalization method.

normoffset numeric offset used for the normalization.

vartr variance stabilizing transformation.
vartroffset numeric offset used for the variance stabilizing transformation.
numdetected msImage of number of detected peaks.
totalioncount msImage of total-ion-count per pixel.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

msiDataset *Constructor for [msi.dataset-class](#) objects.*

Description

msiDataset returns a [msi.dataset-class](#) object. It contains information about the matched peaks intensities, the geometrical dimensions of the mass spectral image, and the common m/z values.

Usage

```
msiDataset(values, mz, rsize, csize, verbose = TRUE)
```

Arguments

values	numeric matrix containing the peaks intensities. Rows represent pixels and columns represent peaks.
mz	array of m/z peaks values.
rsize	geometric shape (number of rows) of image.
csize	geometric shape (number of columns) of image.
verbose	boolean (default = TRUE). Additional output.

Details

Function used to construct the main object [msi.dataset-class](#). This object contains all the information about peaks intensities (intensity matrix), the geometrical shape of the image (rows, columns), and the vector of the common m/z values, generated during the peak matching process.

Value

[msi.dataset-class](#) object.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

Examples

```
## Load package
library("SPUTNIK")

## Create the msi.dataset-class object
sz <- c(5, 4)
numIons <- 20
x <- matrix(rnorm(prod(sz) * numIons), prod(sz), numIons)
mz <- sort(sample(100, numIons))
msiX <- msiDataset(x, mz, sz[1], sz[2])
```

msImage *Constructor for [ms.image-class](#) objects.*

Description

Constructor for [ms.image-class](#) objects.

Usage

```
msImage(values, name = character(), scale = TRUE)
```

Arguments

values	numeric matrix representing the pixels intensities. Rows and columns represent the geometrical shape of the image.
name	image name.
scale	logical (default = TRUE). Whether the intensities should be scaled in [0, 1].

Value

[ms.image-class](#) object.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

Examples

```
## Load package
library("SPUTNIK")

## MS image
imShape <- c(40, 50)
matIm <- matrix(rnorm(200), imShape[1], imShape[2])
im <- msImage(values = matIm, name = "random", scale = TRUE)
```

NMI *Normalized mutual information (NMI).*

Description

NMI returns the normalized mutual information between two `ms.image` objects. The normalized mutual information is calculated as the mutual information divided by square-root of the product of the entropies. This function makes use of the functions available in `infotheo` R package.

Usage

```
NMI(x, y, numBins = 256)
```

Arguments

`x` numeric array. Image 1 color intensity array.
`y` numeric array. Image 2 (binary mask).
`numBins` numeric. Number of bins for discretizing the image colors.

Value

NMI value between 0 and 1.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

References

Meyer, P. E. (2009). `Infotheo`: information-theoretic measures. R package. Version, 1(0).

normIntensity,msi.dataset-method
Normalize the peaks intensities.

Description

Normalize the peaks intensities.

Usage

```
## S4 method for signature 'msi.dataset'  
normIntensity(object, method = "median", peaksInd = NULL, offsetZero = 0)
```


Arguments

object	msi.dataset-class object.
method	string (default = "median"). The normalization method to be used. Valid values are: "median", "PQN", "TIC", TMM, or "upperQuartile". See 'Details' section.
peaksInd	numeric array (default = NULL). Array of peak indices used to calculate the scaling factors (TIC, median). If NULL, all the peaks are used.
offsetZero	numeric (default = 0). This value is added to all the peak intensities to take into accounts of the zeros.

Details

The valid values for method are:

- "median": median of spectrum intensities is scaled to one.
- "PQN":
 1. apply "TIC" normalization
 2. calculate the median reference spectrum (after removing the zeros)
 3. calculate the quotients of peaks intensities and reference
 4. calculate the median of quotients for each peak (after removing the zeros)
 5. divide all the peak intensities by the median of quotients
- "TIC": total ion current normalization assign the sum of the peaks intensities to one.
- "TMM": trimmed mean of M-values (TMM with zero pairing). Called TMMwzp in edgeR.
- "upperQuartile": spectra are scaled by their 3rd quartile.

Value

object [msi.dataset-class](#) object, with normalized peaks intensities.

When using TIC scaling, if zeros are present in the matrix, a positive offset must be added to all the peak intensities through the parameter `offsetZero`. This is necessary for applying the CLR transformation. TIC scaling transforms the spectra into compositional data; in this case the CLR transformation must be applied through the `varTransform` function.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

References

F. Dieterle, A. Ross, G. Schlotterbeck, and Hans Senn. 2006. Probabilistic quotient normalization as robust method to account for dilution of complex biological mixtures. Application in 1H NMR metabonomics. *Analytical Chemistry* 78(13): 4281-4290.

Robinson MD, Oshlack A (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* 11, R25.

See Also[msi.dataset-class](#)**Examples**

```
## Load package
library("SPUTNIK")

## Create the msi.dataset-class object
sz <- c(40, 40)
x <- matrix(rnorm(sz[1] * sz[2] * 20) * 1000, sz[1] * sz[2], 20)
x[x < 0] <- 0 # MS data is positive
mz <- sort(sample(100, ncol(x)))
msiX <- msiDataset(x, mz, sz[1], sz[2])

## Normalize and log-transform
msiX <- normIntensity(msiX, "median")
msiX <- varTransform(msiX, "log")

## Create the msi.dataset-class object
sz <- c(40, 40)
x <- matrix(rnorm(sz[1] * sz[2] * 20) * 1000, sz[1] * sz[2], 20)
x[x < 0] <- 0 # MS data is positive
mz <- sort(sample(100, ncol(x)))
msiX <- msiDataset(x, mz, sz[1], sz[2])

## Normalize using PQN
msiX <- normIntensity(msiX, "PQN")
```

numDetectedMSI,msi.dataset-method

Generates an msImage representing the number of detected peaks per pixel. This image can be used to qualitatively evaluate the spatial heterogeneity of the sample.

Description

Generates an msImage representing the number of detected peaks per pixel. This image can be used to qualitatively evaluate the spatial heterogeneity of the sample.

Usage

```
## S4 method for signature 'msi.dataset'
numDetectedMSI(object)
```

Arguments

object [msi.dataset-class](#) object.

Value

`ms.image-class` object representing the detected ions per pixel.

ovarianDESIDoria2016 *Load the example DESI-MSI data.*

Description

Loads a single human ovarian cancer DESI mass spectrometry imaging dataset acquired in negative ionization mode using Waters XEVO-GS2 qToF. The dataset is available at "<https://raw.githubusercontent.com/paoloinglese/SPUTNIKex>". The dataset is loaded in the R environment under the variable name `maldiData`.

Usage

```
ovarianDESIDoria2016(verbose = TRUE)
```

Arguments

`verbose` Logical (default = TRUE). Show additional output text.

Value

`maldiData` MS intensity matrix. Rows represent pixels, columns represent matched peaks.

References

Doria, M. L., McKenzie, J. S., Mroz, A., Phelps, D. L., Speller, A., Rosini, F., ... & Ghaem-Maghani, S. (2016). Epithelial ovarian carcinoma diagnosis by desorption electrospray ionization mass spectrometry imaging. *Scientific reports*, 6, 39219.

`PCAImage,msi.dataset-method`

Generates an RGB msImage representing the first 3 principal components. This image can be used to qualitatively evaluate the spatial heterogeneity of the sample.

Description

Generates an RGB `msImage` representing the first 3 principal components. This image can be used to qualitatively evaluate the spatial heterogeneity of the sample.

Usage

```
## S4 method for signature 'msi.dataset'  
PCAImage(object, alignToSample = TRUE, seed = NULL)
```

Arguments

object	msi.dataset-class object.
alignToSample	boolean (default = TRUE). If TRUE, the principal component scores are aligned to the pixel mean intensity.
seed	set the random seed (default = NULL).

Value

RGB raster representing the first 3 principal components (see [msImage](#)).

plot,ms.image,missing-method

Visualize an MS image. plot extends the generic function to [ms.image-class](#) objects.

Description

Visualize an MS image. plot extends the generic function to [ms.image-class](#) objects.

Usage

```
## S4 method for signature 'ms.image,missing'
plot(x, palette = "inferno")
```

Arguments

x	ms.image-class object. See msImage .
palette	string. Color palette. See viridis .

Value

a ggplot2 plot.

Examples

```
## Load package

library("SPUTNIK")

## Create ms.image-class object
msIm <- msImage(values = matrix(rnorm(200), 40, 50), name = "test", scale = TRUE)

## Plot the image
plot(msIm)
```

refImageBinaryKmeans *Calculate the binary reference image using k-means clustering. K-Means is run on the first 'npcs' principal components to speed up the calculations.*

Description

Calculate the binary reference image using k-means clustering. K-Means is run on the first 'npcs' principal components to speed up the calculations.

Usage

```
refImageBinaryKmeans(
  dataset,
  npcs = 10,
  alignTo = "detected",
  invertAligned = FALSE
)
```

Arguments

dataset	msi.dataset-class object. See msiDataset .
npcs	int (default = 10). Number of principal components to calculate.
alignTo	string (default = "detected"). Sample reference image to align the estimate binary image. It is expected to correlate with the sample location.
invertAligned	boolean (default = FALSE). If TRUE, the binary image is inverted after being aligned to the sample reference (alignTo).

Value

[ms.image-class](#) object with binary intensities.

refImageBinaryKmeansMulti *Calculate the binary reference image using k-means clustering with multi-cluster merging. K-means is run on the first 'npcs' principal components to speed up the calculations.*

Description

Calculate the binary reference image using k-means clustering with multi-cluster merging. K-means is run on the first 'npcs' principal components to speed up the calculations.

Usage

```
refImageBinaryKmeansMulti(
  dataset,
  npcs = 10,
  mzQuery = numeric(),
  mzTolerance = Inf,
  useFullMZ = TRUE,
  numClusters = 4,
  kernelSize = 5,
  cores = 1,
  verbose = TRUE
)
```

Arguments

dataset	msi.dataset-class object. See msiDataset .
npcs	int (default = 10). Number of principal components to calculate.
mzQuery	numeric. Values of m/z used to calculate the reference image. 2 values are interpreted as interval, multiple or single values are searched in the m/z vector. It overrides the argument useFullMZ.
mzTolerance	numeric (default = Inf). Tolerance in PPM to match the mzQueryRef values in the m/z vector.
useFullMZ	logical (default = TRUE). Whether all the peaks should be used to calculate the reference image.
numClusters	numeric (default = 4). Number of clusters.
kernelSize	4-D numeric array or numeric (default = 5). Each element of the 4-D array represents the size of the corners square kernels used to determine the off-tissue clusters. The element order is clockwise: top-left, top-right, bottom-left, bottom-right. If negative, the corresponding corner is skipped. If only a single value is passed, the same kernel size is used for the 4 corners.
cores	numeric (default = 1). Number of CPU cores for parallel k-means.
verbose	boolean (default = TRUE). Additional output.

Value

[ms.image-class](#) object with binary intensities.

refImageBinaryOtsu *Calculate the binary reference image using Otsu's thresholding.*

Description

Calculate the binary reference image using Otsu's thresholding.

Usage

```
refImageBinaryOtsu(image)
```

Arguments

image [ms.image-class](#) object. See [msImage](#).

Value

[ms.image-class](#) object with binary intensities. #'

refImageBinarySVM	<i>Calculate the binary reference image using linear SVM trained on manually selected pixels.</i>
-------------------	---

Description

Calculate the binary reference image using linear SVM trained on manually selected pixels.

Usage

```
refImageBinarySVM(
  dataset,
  mzQueryRef = numeric(),
  mzTolerance = Inf,
  useFullMZ = TRUE
)
```

Arguments

dataset [msi.dataset-class](#) object. See [msiDataset](#).

mzQueryRef numeric. Values of m/z used to calculate the reference image. 2 values are interpreted as interval, multiple or single values are searched in the m/z vector. It overrides the argument useFullMZ.

mzTolerance numeric (default = Inf). Tolerance in PPM to match the mzQueryRef values in the m/z vector.

useFullMZ logical (default = TRUE). Whether all the peaks should be used to calculate the reference image.

Value

[ms.image-class](#) object with binary intensities.

refImageContinuous	refImageContinuous returns the reference image, calculated using the method. This image represents the basic measure for the filters in SPUTNIK.
--------------------	--

Description

refImageContinuous returns the reference image, calculated using the method. This image represents the basic measure for the filters in SPUTNIK.

Usage

```
refImageContinuous(
  msiData,
  method = "sum",
  mzQueryRef = numeric(),
  mzTolerance = Inf,
  useFullMZRef = TRUE,
  doSmooth = FALSE,
  smoothSigma = 2,
  alignTo = "detected",
  invertAligned = FALSE,
  verbose = TRUE
)
```

Arguments

msiData	msiDataset object..
method	string (default = "sum"). Method used to calculate the reference image. Valid values are: <ul style="list-style-type: none"> • "sum": peak intensities sum • "mean": average peak intensities (without zeros) • "median": median peak intensities (without zeros) • "pca": first principal component scores.
mzQueryRef	numeric. Mass-to-charge ratios used to calculate the reference image. Two values are interpreted as interval, multiple or single values are searched in the m/z vector. It overrides the useFullMZRef argument.
mzTolerance	numeric (default = Inf). Search window in parts-per-million units for the mzQueryRef.
useFullMZRef	logical (default = TRUE). Whether all peaks should be used to calculate the reference image. Ignored if mzQueryRef is provided.
doSmooth	logical (default = FALSE). If TRUE, the reference image is smoothed using a Gaussian kernel.
smoothSigma	numeric (default = 2). Standard deviation of the smoothing Gaussian kernel.
alignTo	string (default = "detected"). The reference image is aligned to the image representing:

- "detected": number of detected peaks
- "tic": total-ion-count image

The reference image will have a positive Pearson's correlation with the selected image.

`invertAligned` logical (default = FALSE). If TRUE, the reference image has negative correlation with the selected image in `alignTo`.

`verbose` logical (default = TRUE). Additional output text.

Details

Function to extract the continuous reference image from a `msi.dataset-class` object. The continuous reference image represents the spatial location of the sample. By default, it is aligned with either the image representing the number of detected peaks, or the total-ion-count in all pixels. It is expected to be higher in the region occupied by the sample (positive correlation with the mask representing the real sample pixels). In some cases, the alignment images can have higher values in the pixels outside of the sample. In these circumstances, the argument `invertAligned` should be set to TRUE.

Value

A continuous valued reference image (see `msImage`).

See Also

`msiDataset`, `msImage`

Examples

```
## Load package
library("SPUTNIK")

## Mass spectrometry intensity matrix
X <- matrix(rnorm(200), 20, 40)
X[X < 0] <- 0

## Print original dimensions
print(dim(X))

## m/z vector
mzVector <- seq(600, 900, by = (900 - 600) / 39)

## Read the image size
imSize <- c(5, 4)

## Construct the ms.dataset object
msiX <- msiDataset(X, mzVector, imSize[1], imSize[2])

## Calculate the reference and ROI images from the ms.dataset-class object msiX.
## The reference is calculated as the first principal component scores scaled
## in [0, 1]; the binary ROI is calculated applying k-means on the entire dataset.
```

```
## Use only m/z values in the range of [700, 900]. The interval extremal values
## are matched within a tolerance of 50 ppm.

refImg <- refImageContinuous(msiX, method = "sum")
roiImg <- refImageBinaryOtsu(refImg)

## Plot the reference and region of interest ROI
## plot(ref.roi$Reference)
## plot(ref.roi$ROI)
```

```
removeSmallObjects,ms.image-method
```

Remove binary ROI objects smaller than user-defined number of pixels

Description

Remove binary ROI objects smaller than user-defined number of pixels

Usage

```
## S4 method for signature 'ms.image'
removeSmallObjects(object, threshold = 5, border = 3)
```

Arguments

object	ms.image-class object. See msImage .
threshold	numeric. Smallest number of connected pixels.
border	numeric (default = 3). Size of the empty border to add before detecting the connected objects. The border is removed at the end of the process. If 'border = 0', no border is added.

Value

[ms.image-class](#) object after filtering.

Examples

```
library(SPUNIK)

fakeBinImage <- matrix(0, 100, 100)
fakeBinImage[sample(prod(dim(fakeBinImage)), 2000)] <- 1

fakeBinMsImage <- msImage(values = fakeBinImage, name = "ROI", scale = FALSE)

# Remove the objects with a number of connected pixels smaller than 5
fakeBinMsImage <- removeSmallObjects(fakeBinMsImage, threshold = 5)
```

scatter.ratio	<i>Pixel scatteredness ratio.</i>
---------------	-----------------------------------

Description

scatter.ratio returns a measure of image scatteredness represented by the ratio between the number of connected components and the total number of non-zero pixels. The number of connected components is calculated from the binarized image using Otsu's method.

Usage

```
scatter.ratio(im)
```

Arguments

im 2-D numeric matrix representing the image pixel intensities.

Value

calculated scatter ratio.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

References

Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 9(1), 62-66.

See Also

[gini.index](#) [spatial.chaos](#)

Examples

```
## Load package
library("SPUTNIK")

## Image
im <- matrix(rnorm(100), 10, 10)
im[im < 0] <- 0

## Spatial chaos
sc <- spatial.chaos(im, levels = 30, morph = TRUE)
stopifnot(sc <= 1)

## Gini index
gi <- gini.index(im, levels = 16)
```

```
stopifnot(gi >= -1 && gi <= 1)

## Scatter ratio
sr <- scatter.ratio(im)
stopifnot(sr <= 1)
```

smoothImage,ms.image-method

Apply Gaussian smoothing to an MS image.

Description

Apply Gaussian smoothing to an MS image.

Usage

```
## S4 method for signature 'ms.image'
smoothImage(object, sigma = 2)
```

Arguments

object [ms.image-class](#) object. See [msImage](#).

sigma numeric (default = 2). Standard deviation of the smoothing Gaussian kernel.

Value

[ms.image-class](#) smoothed [msImage](#).

Examples

```
## Load package
library("SPUTNIK")

## Create ms.image-class object
msIm <- msImage(values = matrix(rnorm(200), 40, 50), name = "test", scale = TRUE)

## Smooth the image colors
msImSmoothed <- smoothImage(msIm, sigma = 5)
```

spatial.chaos	<i>Spatial chaos measure.</i>
---------------	-------------------------------

Description

spatial.chaos returns the 'spatial chaos' randomness measure for imaging data.

Usage

```
spatial.chaos(im, levels = 30, morph = TRUE)
```

Arguments

im	2-D numeric matrix representing the image pixel intensities.
levels	numeric (default = 30). Number of histogram bins.
morph	logical (default = TRUE). Whether morphological operations should be applied to the binary image.

Value

A value between 0 and 1. A value close to 1 represents a high level of spatial scatteredness, a value close to 0 represents a less level of spatial scatteredness. Maximum possible value is $1 - 1 / (\# \text{ histogram bins})$

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

References

Palmer, A., Phapale, P., Chernyavsky, I., Lavigne, R., Fay, D., Tarasov, A., ... & Becker, M. (2017). FDR-controlled metabolite annotation for high-resolution imaging mass spectrometry. *Nature methods*, 14(1), 57.

See Also

[gini.index scatter.ratio](#)

Examples

```
## Load package
library("SPUTNIK")

## Image
im <- matrix(rnorm(100), 10, 10)
im[im < 0] <- 0

## Spatial chaos
```

```

sc <- spatial.chaos(im, levels = 30, morph = TRUE)
stopifnot(sc <= 1)

## Gini index
gi <- gini.index(im, levels = 16)
stopifnot(gi >= -1 && gi <= 1)

## Scatter ratio
sr <- scatter.ratio(im)
stopifnot(sr <= 1)

```

splitPeaksFilter *Test for the presence of split peaks.*

Description

[splitPeaksFilter](#) returns a list of estimated split peak indices. Each element of the list contains an array of the original peak indices that can be merged. The name of the list element is the new m/z value associated with the merged peaks.

Usage

```

splitPeaksFilter(
  msiData,
  mzTolerance = 5,
  sharedPixelsRatio = 0,
  sparseness = "scatter.ratio",
  threshold = 0.5,
  returnDetails = TRUE,
  verbose = TRUE
)

```

Arguments

msiData	msi.dataset-class object. See msiDataset .
mzTolerance	numeric (default = 5). Maximum distance in PPM between the m/z values of two peaks to consider them for merging. See 'Details' section.
sharedPixelsRatio	numeric (default = 0). Maximum fraction of common pixels where the signal of two peaks is different from zero to consider them for merging. See 'Details' section.
sparseness	string (default = "scatter.ratio"). Method used to estimate the 'scatteredness' of the peak image. See 'Details' section.
threshold	numeric (default = 0.5). Threshold for scatteredness measure to consider peaks for merging. At least one of the merging peaks should have a measure associated with presence of structure.
returnDetails	logical (default = TRUE). Add details on merged peaks in the results.
verbose	logical (default = TRUE). Additional output text.

Details

[splitPeaksFilter](#) determines whether close peaks represent the same signal. This estimation is based on multiple conditions:

1. peaks m/z values should be closer than `mzTolerance` PPM
2. at least one of the peak images should be structured, accordingly to the sparseness measure. The threshold determines whether the pixel images are structured or not. The possible measures are:
 - `"scatter.ratio"`: ratio between the number of non-zero pixels and the image size after binarization using Otsu's thresholding. A value close to 0 is associated with a more structured image, whereas a value close to 1 is associated with a less structured image. A suggested parameter of `threshold = 0.5` represents the maximum value for this measure for a structured image. Minimum possible value is $1 / (\# \text{ non-zero pixels })$.
 - `"spatial.chaos"`: similar to the scatter ratio taking into account of the color histogram. A value close to 1 represents a structured image, whereas a value close to 0 represents a more scattered image. A suggested parameter of `threshold = 0.8` represents the minimum value for this measure for a structured image. Maximum possible value is $1 - 1 / (\# \text{ histogram bins })$. Here, we use the default number of bins equal to 30.
 - `"gini.index"`: Gini index measures the image sparsity. A value close to 1 is associated with a sparse image whereas a value close to 0 is associated with a more uniform image. A suggested value of `threshold = 0.9` represents the maximum value of this measure for a structured image.
3. the merged peaks image should be more structured than the single peak images, accordingly to the selected sparseness.

Value

`peak.filter` object. See [applyPeaksFilter](#).

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

References

Palmer, A., Phapale, P., Chernyavsky, I., Lavigne, R., Fay, D., Tarasov, A., ... & Becker, M. (2017). FDR-controlled metabolite annotation for high-resolution imaging mass spectrometry. *Nature methods*, 14(1), 57.

Hurley, N., & Rickard, S. (2009). Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10), 4723-4741.

Examples

```
## Load package
library("SPUTNIK")

## Mass spectrometry intensity matrix
X <- matrix(rnorm(200), 20, 40)
```

```
X[X < 0] <- 0

## Print original dimensions
print(dim(X))

## m/z vector
mzVector <- seq(600, 601, by = (601 - 600) / 39)

## Read the image size
imSize <- c(5, 4)

## Construct the ms.dataset object
msiX <- msiDataset(X, mzVector, imSize[1], imSize[2])

## Determine split peaks
sp.filter <- splitPeaksFilter(
  msiData = msiX, mzTolerance = 50,
  sharedPixelsRatio = 0,
  sparseness = "spatial.chaos", threshold = 0.5
)
```

SSIM

Structural similarity index (SSIM).

Description

ssim returns the value of SSIM between two vectors representing the color intensities of two images.

Usage

```
SSIM(x, y, numBreaks = 256)
```

Arguments

x	numeric array. Image 1 color intensity array.
y	numeric array. Image 2 color intensity array.
numBreaks	numeric. Number of bins for the color histogram.

Details

SSIM is an image quality measure, given a reference considered as noise-less image. It can be also used as a perceived similarity measure between images. The images are converted by default in 8bit.

Value

value of SSIM defined between 0 and 1.

Author(s)

Paolo Inglese <p.inglese14@imperial.ac.uk>

References

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4), 600-612.

totalIonCountMSI,msi.dataset-method

Generates an msImage representing pixels total-ion-counts. This image can be used to qualitatively evaluate the spatial heterogeneity of the sample.

Description

Generates an msImage representing pixels total-ion-counts. This image can be used to qualitatively evaluate the spatial heterogeneity of the sample.

Usage

```
## S4 method for signature 'msi.dataset'  
totalIonCountMSI(object)
```

Arguments

object [msi.dataset-class](#) object.

Value

[ms.image-class](#) object representing the total ion counts.

varTransform,msi.dataset-method

Variance stabilizing transformation.

Description

varTransform transforms the MS intensities in order to reduce heteroscedasticity.

Usage

```
## S4 method for signature 'msi.dataset'  
varTransform(object, method = "log", offsetZero = 1)
```

Arguments

object	msi.dataset-class object. See msiDataset .
method	string (default = log). Transformation method. Valid values are: <ul style="list-style-type: none">• "log", "log2", "log10": log-transformation defined as $\log(x + \text{offsetZero})$.• "sqrt": square-root transformation.• "clr": centered log-transformation. To be used when TIC scaling normalization is applied.
offsetZero	numeric (default = 1). This value is added to all the peak intensities to take into accounts of the zeros. It must be positive.

Value

[msi.dataset-class](#) object with transformed peaks intensities.

Examples

```
## Load package
library("SPUTNIK")

## Create the msi.dataset-class object
sz <- c(40, 40)
x <- matrix(rnorm(sz[1] * sz[2] * 20) * 1000, sz[1] * sz[2], 20)
x[x < 0] <- 0 # MS data is positive
mz <- sort(sample(100, ncol(x)))
msiX <- msiDataset(x, mz, sz[1], sz[2])

## Normalize and log-transform
msiX <- normIntensity(msiX, "median")
msiX <- varTransform(msiX, "log")

## Create the msi.dataset-class object
sz <- c(40, 40)
x <- matrix(rnorm(sz[1] * sz[2] * 20) * 1000, sz[1] * sz[2], 20)
x[x < 0] <- 0 # MS data is positive
mz <- sort(sample(100, ncol(x)))
msiX <- msiDataset(x, mz, sz[1], sz[2])

## Normalize using PQN
msiX <- normIntensity(msiX, "PQN")
```

Index

applyPeaksFilter, [19](#), [39](#)
applyPeaksFilter
 (applyPeaksFilter,msi.dataset-method),getShapeMSI
 [3](#)
applyPeaksFilter,msi.dataset-method, [3](#)
applyPeaksFilter-*msi.dataset-method*,
 [10](#), [12](#)
applyPeaksFilter-*msi.dataset-method*
 (applyPeaksFilter,msi.dataset-method),
 [3](#)

binKmeans
 (binKmeans,msi.dataset-method),
 [4](#)
binKmeans,msi.dataset-method, [4](#)
binKmeans2
 (binKmeans2,msi.dataset-method),
 [5](#)
binKmeans2,msi.dataset-method, [5](#)
binOtsu (binOtsu,ms.image-method), [6](#)
binOtsu,ms.image-method, [6](#)
binSupervised
 (binSupervised,msi.dataset-method),
 [7](#)
binSupervised,msi.dataset-method, [7](#)
bladderMALDIRompp2010, [8](#)

cdf.test, [13](#)
clarkevans.test, [13](#)
closeImage
 (closeImage,ms.image-method), [8](#)
closeImage,ms.image-method, [8](#)
countPixelsFilter, [9](#), [19](#)
createPeaksFilter, [3](#), [11](#), [11](#)
CSRPeaksFilter, [12](#)

getIntensityMat
 (getIntensityMat,msi.dataset-method),
 [14](#)
getIntensityMat,msi.dataset-method, [14](#)

getMZ (getMZ,msi.dataset-method), [15](#)
getMZ,msi.dataset-method, [15](#)
 (getShapeMSI,msi.dataset-method),
 [16](#)
getShapeMSI,msi.dataset-method, [16](#)
gini.index, [17](#), [35](#), [37](#)
globalPeaksFilter, [18](#)

invertImage
 (invertImage,ms.image-method),
 [20](#)
invertImage,ms.image-method, [20](#)

ms.image-class, [4](#), [6–9](#), [13](#), [18](#), [20](#), [21](#), [21](#),
 [23](#), [27–31](#), [34](#), [36](#), [41](#)
msi.dataset-class, [3–5](#), [7](#), [9](#), [13–16](#), [18](#), [21](#),
 [21](#), [22](#), [25](#), [26](#), [28–31](#), [38](#), [41](#), [42](#)
msiDataset, [9](#), [13](#), [18](#), [22](#), [29–32](#), [38](#), [42](#)
msImage, [6](#), [8](#), [9](#), [20](#), [23](#), [28](#), [31](#), [33](#), [34](#), [36](#)

NMI, [24](#)
normIntensity
 (normIntensity,msi.dataset-method),
 [24](#)
normIntensity,msi.dataset-method, [24](#)
numDetectedMSI
 (numDetectedMSI,msi.dataset-method),
 [26](#)
numDetectedMSI,msi.dataset-method, [26](#)

ovarianDESIDoria2016, [27](#)

PCAIImage (PCAIImage,msi.dataset-method),
 [27](#)
PCAIImage,msi.dataset-method, [27](#)
plot (plot,ms.image,missing-method), [28](#)
plot,ms.image,missing-method, [28](#)

refImageBinaryKmeans, [29](#)
refImageBinaryKmeansMulti, [29](#)

refImageBinaryOtsu, [30](#)
refImageBinarySVM, [31](#)
refImageContinuous, [32](#)
refImageOtsu (refImageBinaryKmeans), [29](#)
removeSmallObjects
 (removeSmallObjects,ms.image-method),
 [34](#)
removeSmallObjects,ms.image-method, [34](#)

scatter.ratio, [17](#), [35](#), [37](#)
smoothImage
 (smoothImage,ms.image-method),
 [36](#)
smoothImage,ms.image-method, [36](#)
spatial.chaos, [17](#), [35](#), [37](#)
splitPeaksFilter, [38](#), [38](#), [39](#)
SSIM, [40](#)

totalIonCountMSI
 (totalIonCountMSI,msi.dataset-method),
 [41](#)
totalIonCountMSI,msi.dataset-method,
 [41](#)

varTransform
 (varTransform,msi.dataset-method),
 [41](#)
varTransform,msi.dataset-method, [41](#)
viridis, [28](#)