

# Package ‘SPUTNIK’

October 25, 2018

**Type** Package

**Title** SPatially aUTomatic deNoising for Ims toolKit

**Version** 1.1.1

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

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

**Description** A set of tools for the peak filtering of mass spectrometry imaging data (MSI or IMS) based on spatial distribution of signal. Given a region-of-interest (ROI), 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 dimensionality and increase its information vs noise ratio, improving the quality of the unsupervised analysis results, reducing data dimensionality and simplifying the chemical interpretation.

**Depends** R (>= 3.4.0)

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** no

**Imports** e1071, spatstat, viridis, SDMTools, ggplot2, reshape, imager,  
methods, infotheo, parallel

**RoxygenNote** 6.1.0

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

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

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-10-25 21:50:28 UTC

**R topics documented:**

applyPeaksFilter,msi.dataset-method . . . . .	2
binKmeans,msi.dataset-method . . . . .	4
binKmeans2,msi.dataset-method . . . . .	4
binOtsu,ms.image-method . . . . .	6
binSupervised,msi.dataset-method . . . . .	6
bladderMALDIRompp2010 . . . . .	7
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 . . . . .	20
msi.dataset-class . . . . .	21
msiDataset . . . . .	21
msImage . . . . .	22
NMI . . . . .	23
normIntensity,msi.dataset-method . . . . .	24
ovarianDESIDoria2016 . . . . .	25
plot,ms.image,missing-method . . . . .	26
refAndROLimages . . . . .	26
removeSmallObjects,ms.image-method . . . . .	29
scatter.ratio . . . . .	30
smoothImage,ms.image-method . . . . .	31
spatial.chaos . . . . .	31
splitPeaksFilter . . . . .	33
SSIM . . . . .	35
varTransform,msi.dataset-method . . . . .	36
<b>Index</b> . . . . .	<b>37</b>

---

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(200), 20, 40)  
  
## 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 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 peaks intensities.*

---

### Description

Return a binary mask generated applying k-means clustering on peaks intensities.

### Usage

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

### Arguments

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

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

## Binarize the reference image
refBin <- binKmeans(msiX)

## Plot the binarized reference
plot(refBin)
```

---

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 = numeric(), numClusters = 4,  
  kernelSize = c(3, 3, 3, 3), numCores = 1, verbose = TRUE)
```

## Arguments

object	<a href="#">msi.dataset-class</a> 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. Tolerance in PPM to match the mzQueryRef values in the m/z vector. Only valid when useFullMZ = FALSE.
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 clusters.
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 <- binOtsu(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(), useFullMZ = T, mzTolerance = numeric(),  
  method = "svm")
```

**Arguments**

object	<a href="#">msi.dataset-class</a> object
refImage	<a href="#">ms.image-class</a> 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 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. Tolerance in PPM to match the mzQueryRef values in the m/z vector. Only valid when useFullMZ = FALSE.
method	string (default = 'svm'). Supervised classifier used to segment the ROI.

**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.githubusercontent.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 <- binOtsu(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	<a href="#">msi.dataset-class</a> object. See <a href="#">msiDataset</a> .
roiImage	<a href="#">ms.image-class</a> object representing the ROI mask. See <a href="#">msImage</a> .
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(200), 20, 40)

## 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])

## Extract the ROI using k-means
ref.roi <- refAndROIimages(msiData = msiX, refMethod = "sum",
                          roiMethod = "otsu", useFullMZRef = TRUE)

## Perform count pixels filtering
count.sel <- countPixelsFilter(msiData = msiX, roiImage = ref.roi$ROI,
                              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-method](#). 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	<i>Performs the peak selection based on complete spatial randomness test.</i>
----------------	---

---

## 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,
  covMethod = "sum", mzQueryCov = numeric(), mzTolerance = numeric(),
  useFullMZCov = TRUE, smoothCov = FALSE, smoothCovSigma = 2,
  invertCov = FALSE, adjMethod = "bonferroni", returnQvalues = TRUE,
  plotCovariate = FALSE, verbose = TRUE, ...)
```

## Arguments

msiData	<a href="#">msi.dataset-class</a> object. See <a href="#">msiDataset</a> .
method	string (default = "ClarkEvans"). CSR statistical test applied to the peaks signal. Accepted values are: <ul style="list-style-type: none"> <li>"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.</li> <li>"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.</li> </ul>
covariateImage	<a href="#">ms.image-class</a> object. An image used as covariate (necessary for Kolmogorov-Smirnov test). If NULL, the covariate image is calculated using the method defined by 'covMethod'.
covMethod	string (default = "sum"). Method used to calculate the reference image. Read only when method = "KS". Possible values are described in 'refAndROIimages'.
mzQueryCov	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 useFullMZCov = TRUE. Read only when method = "KS".
mzTolerance	numeric. Tolerance in PPM to match the mzQueryCov values in the m/z vector. It should be left unset when using useFullMZCov = TRUE. Read only when method = "KS".
useFullMZCov	logical (default = TRUE). Whether all the peaks should be used to calculate the covariate image. Read only when method = "KS".

smoothCov	logical (default = FALSE). Whether the covariate image should be smoothed using a Gaussian kernel. Read only when method = "KS".
smoothCovSigma	numeric (default = 2). Standard deviation of the smoothing Gaussian kernel. Read only when method = "KS".
invertCov	logical (default = FALSE). Whether the covariate image colors should be inverted.]
adjMethod	string (default = "bonferroni"). Multiple testing correction method. Possible values coincide with those of the stats::p.adjust 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 method = "KS".
verbose	logical (default = TRUE). Additional output texts are generated.
...	additional parameters compatible with the statspat functions. See <a href="#">cdf.test</a> for "KS" and <a href="#">clarkevans.test</a> for "ClarkEvans"

**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(200), 20, 40)

## 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 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)
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)
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 MS image*

---

## Description

Returns the geometrical shape of MS image

## 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)  
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)

## 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, verbose = TRUE)

```

## Arguments

msiData	<a href="#">msi.dataset-class</a> object. See <a href="#">msiDataset</a> .
referenceImage	<a href="#">ms.image-class</a> 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: <ul style="list-style-type: none"> <li>• pearson: Pearson's correlation</li> <li>• spearman: Spearman's correlation</li> <li>• ssim: structural similarity index measure</li> <li>• nmi: normalized mutual information.</li> </ul>
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.
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, can be calculated using the [refAndROIimages](#) function. 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 linkapplyPeaksFilter.

**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(200), 20, 40)

## 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])

## Generate the reference image and the ROI mask
ref.roi <- refAndROIimages(msiX, refMethod = "sum", roiMethod = "otsu")

## Perform global peaks filter
glob.peaks <- globalPeaksFilter(msiData = msiX, referenceImage = ref.roi$Reference,
                              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.

### 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)
```

### Arguments

<code>values</code>	numeric matrix containing the peaks intensities. Rows represent pixels and columns represent peaks.
<code>mz</code>	array of m/z peaks values.
<code>rsize</code>	geometric shape (number of rows) of image.
<code>csize</code>	geometric shape (number of columns) of image.

### 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	<i>Normalized mutual information (NMI).</i>
-----	---

---

**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**

<code>x</code>	numeric array. Image 1 color intensity array.
<code>y</code>	numeric array. Image 2 (binary mask).
<code>numBins</code>	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")
```

## Arguments

object            [msi.dataset-class](#) object.  
method           String (default = "median"). the normalization method to be used. Valid values are: "TIC", "median", or "PQN". See 'Details' section.

## Details

The valid values for method are:

- "TIC": total ion current normalization assign the sum of the peaks intensities to one.
- "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

## Value

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

## 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.



**See Also**[msi.dataset-class](#)**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 # 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(5, 4)
x <- matrix(rnorm(sz[1] * sz[2]), 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")
```

---

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/SPUTNIK>". 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.

---

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](#).

## 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)
```

---

refAndROIimages

*Compute the reference image and the ROI mask.*

---

## Description

refAndROIimages returns the reference image, calculated using the refMethod, and the ROI binary mask, calculated using roiMethod. These images represent the basic measures for the filters in SPUTNIK.

**Usage**

```
refAndROIimages(msiData, refMethod = "sum", roiMethod = "otsu",
  mzQueryRef = numeric(), mzTolerance = numeric(),
  useFullMZRef = TRUE, smoothRef = FALSE, smoothSigma = 2,
  invertRef = FALSE, numClusters = 4, sizeKernel = 5, numCores = 1,
  verbose = TRUE)
```

**Arguments**

msiData	<a href="#">msiDataset</a> object..
refMethod	string (default = "sum"). Method used to calculate the reference image. Valid values are: <ul style="list-style-type: none"> <li>• "sum": peak intensities sum</li> <li>• "mean": average peak intensities (without zeros)</li> <li>• "median": median peak intensities (without zeros)</li> <li>• "pca": first principal component scores.</li> </ul>
roiMethod	string (default = "otsu"). Method used to extract the ROI binary mask. Valid values are: <ul style="list-style-type: none"> <li>• "otsu": the reference image is binarized using Otsu's thresholding</li> <li>• "kmeans": msiData is partitioned in 2 clusters using k-means</li> <li>• "kmeans2": k-means is applied with a user-defined number of clusters (see Details)</li> <li>• "supervised": supervised segmentation based on user-defined areas corresponding to off-sample and sample regions.</li> </ul>
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 should be left unset when using useFullMZRef = TRUE.
mzTolerance	numeric. Tolerance in PPM to match the mzQueryRef values in the m/z vector. Only valid when useFullMZ = FALSE.
useFullMZRef	logical (default = TRUE). Whether all the peaks should be used to calculate the reference image.
smoothRef	logical (default = FALSE). Whether the reference image should be smoothed before binarizing. Only valid for roiMethod = "otsu".
smoothSigma	numeric (default = 2). Standard deviation of Gaussian kernel.
invertRef	logical (default = FALSE). Whether the reference image colors should be inverted. This can be necessary when the signal is more intense outside the ROI.
numClusters	numeric (default = 4). Only for 'kmeans2' method. Number of clusters.
sizeKernel	4-D numeric array or numeric (default = 5). Only for 'kmeans2'. 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.
numCores	numeric (default = 1). Only for 'kmeans2' method. Number of CPU cores for parallel k-means. It must be smaller than the number of available cores.
verbose	logical (default = TRUE). Additional output text.

## Details

Function to extract the reference image from a `msi.dataset-class` object. Two reference images are returned, a continuous-valued and a binary-valued. Multiple methods can be used to extract both the continuous and the binary reference images, which afterwards can be used as argument for the `globalPeaksFilter` filter. When 'kmeans2' is applied, the ROI is obtained by merging the sample-related clusters. The user can set a larger number of cluster than 2 (like in 'kmeans'), in such a way a finer segmentation of the sample-related area can be generated. Currently, the off-sample clusters are identified by looking at the most frequent (statistical mode) labels in the corners of the image.

## Author(s)

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

## See Also

`msiDataset`, `binOtsu`, `binKmeans`

## Examples

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

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

## 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.

ref.roi <- refAndROIimages(msiData = msiX, refMethod = "sum",
                          roiMethod = "otsu", useFullMZRef = TRUE)

## 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

<code>object</code>	<a href="#">ms.image-class</a> object. See <a href="#">msImage</a> .
<code>threshold</code>	numeric. Smallest number of connected pixels.
<code>border</code>	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.

### 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)

## 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

*Spatial chaos measure.*

---

### 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)

## 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	<a href="#">msi.dataset-class</a> object. See <a href="#">msiDataset</a> .
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.

### 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)

## 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	<i>Structural similarity index (SSIM).</i>
------	--

---

**Description**

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

**Usage**

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

**Arguments**

<code>x</code>	numeric array. Image 1 color intensity array.
<code>y</code>	numeric array. Image 2 color intensity array.
<code>numBreaks</code>	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.

---

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")
```

## Arguments

object            [msi.dataset-class](#) object. See [msiDataset](#).  
method            string (default = log). Transformation method. Valid values are:

- "log": log-transformation defined as  $\log(x + 1)$
- "sqrt": square-root transformation.

## Value

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

## 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 # 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(5, 4)  
x <- matrix(rnorm(sz[1] * sz[2]), 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  
    (applyPeaksFilter,msi.dataset-method),  
    2  
applyPeaksFilter,msi.dataset-method, 2  
applyPeaksFilter-msi.dataset-method,  
    10, 11  
applyPeaksFilter-msi.dataset-method  
    (applyPeaksFilter,msi.dataset-method),  
    2  
binKmeans  
    (binKmeans,msi.dataset-method),  
    4  
binKmeans,msi.dataset-method, 4  
binKmeans2  
    (binKmeans2,msi.dataset-method),  
    4  
binKmeans2,msi.dataset-method, 4  
binOtsu (binOtsu,ms.image-method), 6  
binOtsu,ms.image-method, 6  
binSupervised  
    (binSupervised,msi.dataset-method),  
    6  
binSupervised,msi.dataset-method, 6  
bladderMALDIRompp2010, 7  
cdf.test, 13  
clarkevans.test, 13  
closeImage  
    (closeImage,ms.image-method), 8  
closeImage,ms.image-method, 8  
countPixelsFilter, 9, 19  
createPeaksFilter, 2, 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  
    (getShapeMSI,msi.dataset-method),  
    16  
getShapeMSI,msi.dataset-method, 16  
gini.index, 17, 30, 32  
globalPeaksFilter, 18, 28  
invertImage  
    (invertImage,ms.image-method),  
    20  
invertImage,ms.image-method, 20  
ms.image-class, 4–9, 12, 18, 20, 20, 22, 26,  
    29, 31  
msi.dataset-class, 3–5, 7, 9, 12, 14–16, 18,  
    21, 21, 22, 24, 25, 33, 36  
msiDataset, 9, 12, 18, 21, 27, 33, 36  
msImage, 6, 8, 9, 20, 22, 26, 29, 31  
NMI, 23  
normIntensity  
    (normIntensity,msi.dataset-method),  
    24  
normIntensity,msi.dataset-method, 24  
ovarianDESIDoria2016, 25  
plot (plot,ms.image,missing-method), 26  
plot,ms.image,missing-method, 26  
refAndROIimages, 18, 26  
removeSmallObjects  
    (removeSmallObjects,ms.image-method),  
    29  
removeSmallObjects,ms.image-method, 29  
scatter.ratio, 17, 30, 32  
smoothImage  
    (smoothImage,ms.image-method),  
    31

`smoothImage`, `ms.image-method`, [31](#)

`spatial.chaos`, [17](#), [30](#), [31](#)

`splitPeaksFilter`, [33](#), [33](#)

SSIM, [35](#)

`varTransform`

(`varTransform`, `msi.dataset-method`),

[36](#)

`varTransform`, `msi.dataset-method`, [36](#)

`viridis`, [26](#)