Package 'radiomics'

March 30, 2018

Type Package

Title 'Radiomic' Image Processing Toolbox

Version 0.1.3

Date 2018-03-15

Author Joel Carlson

Maintainer Joel Carlson <jnkcarlson@gmail.com>

Description Functions to extract first and second order statistics from images.

License MIT + file LICENSE

Depends R (>= 2.10)

Imports spatstat, reshape2, methods, Rcpp

Suggests testthat, knitr, rmarkdown, viridis, devtools, roxygen2

LazyData true

VignetteBuilder rmarkdown, knitr

Collate 'GLSZM.R' 'MGLSZM.R' 'GLRLM.R' 'GLCM.R' 'CalculateFeatures.R' 'FirstOrder.R' 'GLCMFeatures.R' 'GLRLMFeatures.R' 'GLSZMFeatures.R' 'ImageQuantize.R' 'RcppExports.R' 'data.R' 'image.R' 'radiomics.R' 'startupMessage.R'

LinkingTo Rcpp

RoxygenNote 6.0.1.9000

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-03-30 12:48:24 UTC

R topics documented:

`bars` *Vertical Bars*

Description

A Matrix of vertical bars, each column with a different value. Column 1 contains the value 1, up to column 20 with value 20.

Usage

```
bars
```

Format

A matrix, 20 rows by 20 columns

`calc_features` *Calculate texture and first order statistics.*

Description

`calc_features` Calculates features of given texture matrix. If a simple matrix is given, will calculate first order features. If desired, user may input the features they wish to calculate for a given matrix type by passing them as a vector of strings to the `features` argument.

Usage

```
calc_features(object, features = c())

## S4 method for signature 'matrix'
calc_features(object, features = c())

## S4 method for signature 'glcm'
calc_features(object, features = c())

## S4 method for signature 'glrlm'
calc_features(object, features = c())

## S4 method for signature 'glszm'
calc_features(object, features = c())

## S4 method for signature 'mglszm'
calc_features(object, features = c())
```

Arguments

`object`	An object of class "matrix", "glcm", "glrlm", "glszm", or "mglszm"
`features`	A vector containing the features the user wishes to calculate for a given matrix type.

Details

Lists of features available for each matrix type can be accessed through `?first_order_features` `?glcm_features`, `?glrlm_features`, `?glszm_features`.

Matrices of class `mglszm` accept features belonging to the glszm.

Value

A data frame with a single observation. The columns of the dataframe correspond to the calculated features.

Methods (by class)

- `matrix`: Calculate first order features of a numeric matrix
- `glcm`: Calculate texture features of a glcm matrix
- `glrlm`: Calculate texture features of a glrlm matrix
- `glszm`: Calculate texture features of a glszm matrix
- `mglszm`: Calculate texture features of an mglszm matrix

References

http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0102107

See Also

glcm glrlm glszm mglszm

Examples

```
## Not run:
calc_features(glcm(hallbey))
calc_features(glrlm(psf, n_grey=10))
calc_features(glcm(hallbey), features=c("glcm_mean", "glcm_variance", "pickles"))

## End(Not run)
```

---

discretizeImage *Image Discretization.*

Description

`discretizeImage` Scales the grey values of an image into a specified number of values.

Usage

```
discretizeImage(data, n_grey = 32, verbose = TRUE)
```

Arguments

`data`	A numeric 2D matrix.
`n_grey`	an integer value, the number of grey levels the image should be quantized into.
`verbose`	Logical, a message is given when the user supplies more grey values than exist in the image. Setting this value to FALSE will suppress this message.

Details

This function is called in `glcm`, `glrlm`, `glszm`, and `mglszm`.

If n_grey is greater than the number of unique grey levels in the matrix then no action is taken.

Value

A matrix of the same dimensions as the input matrix. The entries of the matrix will be set to begin at 1, and go up to the specified value. There is no guarantee that each gray level between 1 and n_grey will have pixels of that value (for example, although n_grey = 32 may be specified, certain images may contain fewer than 32 grey levels).

Examples

```
image(psf)
image(discretizeImage(psf, n_grey=5, verbose=F))

image(tumor)
image(discretizeImage(tumor, n_grey=8, verbose=F))
```

---

discretizeImage2 *Image Discretization.*

Description

#' discretizeImage2 Scales the grey values of an image into a specified number of values.

Usage

```
discretizeImage2(image, n_grey = 32)
```

Arguments

image	A numeric image matrix.
n_grey	The grey levels the output image will have

Details

Not currently used. Different methods of discretizing the image will be explored in the future.

---

first_order_features *First order features*

Description

First order features

Usage

```
calc_energy(data)

calc_entropy(data, base = 2, nbins = length(unique(c(data))))

calc_kurtosis(data)

calc_meanDeviation(data)

calc_skewness(data)

calc_uniformity(data, nbins = length(unique(c(data))))

calc_mean(data)

calc_median(data)

calc_max(data)

calc_min(data)

calc_variance(data)

calc_RMS(data)

calc_sd(data)
```

Arguments

data	Numeric 2D matrix data.
base	The base for which the logarithm is calculate
nbins	The number of bins the histogram is discretized into

Functions

- calc_energy: Energy (ASM)
- calc_entropy: Entropy
- calc_kurtosis: Kurtosis
- calc_meanDeviation: Mean Deviation
- calc_skewness: Skewness
- calc_uniformity: Uniformity
- calc_mean: Mean
- calc_median: Median
- calc_max: Maximum Value
- calc_min: Minimum Value

- calc_variance: Variance
- calc_RMS: Root Mean Squared
- calc_sd: Standard Deviation

References

http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0102107#s5

---

glcm *Gray level co-occurrence matrix.*

Description

glcm returns a gray level co-occurrence matrix for a given matrix.

Usage

```
glcm(data, angle = 0, d = 1, n_grey = 32, normalize = TRUE, ...)
```

Arguments

data	A numeric 2D matrix.
angle	One of "0", "45", "90" or "135", the pixel to which the current pixel is compared.
d	an integer value, the distance between the current pixel, and the pixel to which it is compared.
n_grey	an integer value, the number of grey levels the image should be quantized into. If greater than the number of unique values in the image, no action will be taken.
normalize	Logical value, if TRUE (default) the matrix will be normalized such that the sum of it's components is 1.
...	Can be given verbose=FALSE to suppress output from the n_grey conversion.

Details

Can be visualized using image(glcm(data)). For visualization info see ?image.radiomics

Value

a matrix of dimension n_grey by n_grey, the GLCM. The column and row names represent grey values in the image.

References

http://www.fp.ucalgary.ca/mhallbey/tutorial.htm

Examples

```
## Not run:
hallbey
glcm(hallbey)
glcm(hallbey, angle="90") #vertical GLCM

## End(Not run)
```

`glcm0` *Create a 0 degree GLCM*

Description

Used internally by glcm()

Usage

```
glcm0(x, n_grey, d)
```

Arguments

`x`	A Numeric matrix, integer values only
`n_grey`	Number of grey levels
`d`	distance from reference pixel to neighbour pixel

`glcm135` *Create a 135 degree GLCM*

Description

Used internally by glcm()

Usage

```
glcm135(x, n_grey, d)
```

Arguments

`x`	A Numeric matrix, integer values only
`n_grey`	Number of grey levels
`d`	distance from reference pixel to neighbour pixel

glcm45 *Create a 45 degree GLCM*

Description

Used internally by glcm()

Usage

```
glcm45(x, n_grey, d)
```

Arguments

x	A Numeric matrix, integer values only
n_grey	Number of grey levels
d	distance from reference pixel to neighbour pixel

glcm90 *Create a 90 degree GLCM*

Description

Used internally by glcm()

Usage

```
glcm90(x, n_grey, d)
```

Arguments

x	A Numeric matrix, integer values only
n_grey	Number of grey levels
d	distance from reference pixel to neighbour pixel

glcm_features *GLCM Features*

Description

GLCM Features

Usage

```
glcm_mean(glcm)

glcm_variance(glcm)

glcm_autoCorrelation(glcm)

glcm_cProminence(glcm)

glcm_cShade(glcm)

glcm_cTendency(glcm)

glcm_contrast(glcm)

glcm_correlation(glcm)

glcm_differenceEntropy(glcm, base = 2)

glcm_dissimilarity(glcm)

glcm_energy(glcm)

glcm_entropy(glcm, base = 2)

glcm_homogeneity1(glcm)

glcm_homogeneity2(glcm)

glcm_IDMN(glcm)

glcm_IDN(glcm)

glcm_inverseVariance(glcm)

glcm_maxProb(glcm)

glcm_sumAverage(glcm)
```

```
glcm_sumEntropy(glcm, base = 2)

glcm_sumVariance(glcm)
```

Arguments

glcm	A matrix of class "glcm" produced by glcm.
base	Base of the logarithm in differenceEntropy.

Functions

- glcm_mean: Mean
- glcm_variance: Variance
- glcm_autoCorrelation: Autocorrelation
- glcm_cProminence: Cluster Prominence
- glcm_cShade: Cluster Shade
- glcm_cTendency: Cluster Tendency
- glcm_contrast: Contrast
- glcm_correlation: Correlation
- glcm_differenceEntropy: Difference Entropy
- glcm_dissimilarity: Dissimilarity
- glcm_energy: Energy
- glcm_entropy: Entropy
- glcm_homogeneity1: Homogeneity
- glcm_homogeneity2: Homogeneity 2
- glcm_IDMN: Inverse Difference Moment (Normalized)
- glcm_IDN: Inverse Difference (Normalized)
- glcm_inverseVariance: Inverse Variance
- glcm_maxProb: Maximum Probability
- glcm_sumAverage: Sum Average
- glcm_sumEntropy: Sum Entropy
- glcm_sumVariance: Sum Variance

References

http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0102107#s5

`glrlm` *Gray level run length matrix.*

Description

`glrlm` returns a gray level run length matrix for a given matrix.

Usage

```
glrlm(data, angle = 0, n_grey = 32, max_run_length = min(dim(data)),
  truncate = TRUE, ...)
```

Arguments

`data`	A numeric 2D matrix.
`angle`	One of 0, 45, 90 or 135, the direction the run is calculated.
`n_grey`	an integer value, the number of grey levels the image should be quantized into.
`max_run_length`	An integer value, the default is the maximum possible run length. Setting it to a smaller value truncates the output. Desirable in cases where the matrix is extremely sparse, for example when there are few long runs.
`truncate`	Logical Remove run lengths which have no entries
`...`	Can be given verbose=FALSE to suppress output from the n_grey conversion.

Details

Can be visualized using `image(glrlm(data))`. For visualization info see `?image.radiomics`

Value

a matrix of class "glrlm" of dimension n_grey by run length. The column names represent the length of the run, and row names represent grey values in the image.

References

http://www.sciencedirect.com/science/article/pii/S0146664X75800086

Examples

```
## Not run:
hallbey
glrlm(hallbey)
glrlm(hallbey, angle="90")

## End(Not run)
```

glrlm_features *GLRLM Features*

Description

GLRLM Features

Usage

```
glrlm_GLN(glrlm)

glrlm_HGLRE(glrlm)

glrlm_LRE(glrlm)

glrlm_LRHGLE(glrlm)

glrlm_LRLGLE(glrlm)

glrlm_LGLRE(glrlm)

glrlm_RLN(glrlm)

glrlm_RP(glrlm)

glrlm_SRE(glrlm)

glrlm_SRHGLE(glrlm)

glrlm_SRLGLE(glrlm)
```

Arguments

glrlm	A matrix of class "glrlm" produced by glrlm.

Functions

- glrlm_GLN: Grey level non-uniformity
- glrlm_HGLRE: High Gray level run emphasis
- glrlm_LRE: Long Run Emphasis
- glrlm_LRHGLE: Long run high gray level emphasis
- glrlm_LRLGLE: Long Run Low Gray Level Emphasis
- glrlm_LGLRE: Low gray level run emphasis
- glrlm_RLN: Run length non-uniformity
- glrlm_RP: Run Percentage

- glrlm_SRE: Short run emphasis
- glrlm_SRHGLE: rt run high gray level emphasis
- glrlm_SRLGLE: Short run low grey emphasis

References

http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0102107#s5

---

glszm *Gray level size zone matrix.*

Description

glszm returns a gray level size zone matrix for a given matrix.

Usage

```
glszm(data, n_grey = 32, truncate = TRUE, ...)
```

Arguments

data	A numeric 2D matrix.
n_grey	an integer value, the number of grey levels the image should be quantized into.
truncate	Logical. Remove values for sizes that have no entries
...	Can be given verbose=FALSE to suppress output from the n_grey conversion.

Details

Can be visualized using image(glszm(data)). For visualization info see ?image.radiomics

Value

a matrix of dimension n_grey by region size, the GLSZM. The column names represent the region size, row names represent grey level, and the entries represent the count of how many times a given size of given grey level occur.

References

http://thibault.biz/Research/ThibaultMatrices/GLSZM/GLSZM.html

Examples

```
## Not run:
image(psf)
glszm(psf)

image(discretizeImage(psf, n_grey=5, verbose=F))
glszm(psf, n_grey=5, verbose=F)

## End(Not run)
```

`glszm_features` *GLSZM Features*

Description

GLSZM Features

Usage

```
glszm_SAE(glszm)

glszm_LAE(glszm)

glszm_IV(glszm)

glszm_SZV(glszm)

glszm_ZP(glszm)

glszm_LIE(glszm)

glszm_HIE(glszm)

glszm_LISAE(glszm)

glszm_HISAE(glszm)

glszm_LILAE(glszm)

glszm_HILAE(glszm)
```

Arguments

`glszm`	A matrix of class "glszm" produced by `glszm`.

Functions

- `glszm_SAE`: Small Area Emphasis
- `glszm_LAE`: Large Area Emphasis
- `glszm_IV`: Intensity Variability
- `glszm_SZV`: Size Zone Variability
- `glszm_ZP`: Zone percentage
- `glszm_LIE`: Low intensity emphasis
- `glszm_HIE`: High intensity emphasis
- `glszm_LISAE`: Low intensity small area emphasis
- `glszm_HISAE`: High intensity small area emphasis
- `glszm_LILAE`: Low intensity large area emphasis
- `glszm_HILAE`: High intensity Large area emphasis

References

http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0102107#s5

`hallbey` *Hall Bey*

Description

A Matrix used as examples in the hallbey explanation of glcms.

Usage

```
hallbey
```

Format

A matrix, 4 rows by 4 columns

image.radiomics *Texture Matrix Visualization*

Description

Texture Matrix Visualization

GLCM image

GLRLM image

GLSZM image

MGLSZM image

Usage

```
## S4 method for signature 'glcm'
image(x, xlab = "Grey Level", ylab = "Grey Level",
  col = colscale(length(unique(c(x@.Data)))))

## S4 method for signature 'glrlm'
image(x, xlab = "Grey Level", ylab = "Run Length",
  col = colscale(length(unique(c(x@.Data)))))

## S4 method for signature 'glszm'
image(x, xlab = "Grey Level", ylab = "Zone Size",
  col = colscale(length(unique(c(x@.Data)))))

## S4 method for signature 'mglszm'
image(x, xlab = "Grey Level", ylab = "Zone Size",
  col = colscale(length(unique(c(x@.Data)))))
```

Arguments

x	Matrix of class "glcm", "glrlm", "glszm" or "mglszm"
xlab	The label for the x-axis
ylab	The label for the y-axis
col	Use viridis scale if available

Examples

```
## Not run:
image(psf)
image(glszm(psf))

## End(Not run)
```

`mglszm` *Multiple gray level size zone matrix.*

Description

`mglszm` returns a matrix of class "mglszm", the multiple gray level size zone matrix for a given matrix.

Usage

```
mglszm(data, truncate = TRUE, ...)
```

Arguments

`data`	A 2D image matrix.
`truncate`	Logical, removes any sizes or gray levels that have no entries.
`...`	Can be given verbose=FALSE to suppress output from the n_grey conversion.

Details

The function creates a GLSZM using grey levels: 2, 4, 8, 16, 32, 64, 128, and 256. The values of these GLSZM's are then weighted and combined using a gaussian distribution with mean of 0 and sd of 1.

Can be visualized using `image(mglszm(data))`. For visualization info see `?image.radiomics`

Value

a matrix of dimension n_grey by region size, the MGLSZM. The column names represent the region size, row names represent grey level, and the entries represent the count of how many times a given size of given grey level occur.

References

http://thibault.biz/Research/ThibaultMatrices/MGLSZM/MGLSZM.html

Examples

```
## Not run:
image(psf)
mglszm(psf)

image(discretizeImage(psf, n_grey=5, verbose=F))
mglszm(psf, n_grey=5, verbose=F)

## End(Not run)
```

noise *Noise*

Description

A Matrix uniformly distributed (on 1 to 100) noise.

Usage

```
noise
```

Format

A matrix, 50 rows by 50 columns

psf *Point Spread Function*

Description

A Matrix of a point spread function. Values are smallest in the middle, and increase in a radial fashion.

Usage

```
psf
```

Format

A matrix, 50 rows by 50 columns

radiomics *radiomics: A texture analysis toolbox for image classification*

Description

radiomics provides a several new classes of matrices: GLCM (grey level co-occurrence matrix), GLRLM (grey level run-length matrix), GLSZM (grey level size-zone matrix), and the MGLSZM (multiple GLSZM).

Details

To learn more about radiomics and texture matrices, start with the vignettes: `browseVignettes(package = "radiomics")`

tumor *Brain Tumor Slice*

Description

A Matrix of a single image slice of a tumor taken from an MRI image. This slice was extracted from one of the sample data sets of 3DSlicer.

Usage

```
tumor
```

Format

A matrix, 47 rows by 46 columns

Index