

# Package ‘GLCMTextures’

November 10, 2023

**Title** GLCM Textures of Raster Layers

**Version** 0.3.9

**Description** Calculates grey level co-occurrence matrix (GLCM) based texture measures (Hall-Beyer (2017) <[https://prism.ucalgary.ca/bitstream/handle/1880/51900/texture%20tutorial%20v%203\\_0%20180206.pdf](https://prism.ucalgary.ca/bitstream/handle/1880/51900/texture%20tutorial%20v%203_0%20180206.pdf)>; Haralick et al. (1973) <[doi:10.1109/TSMC.1973.4309314](https://doi.org/10.1109/TSMC.1973.4309314)>) of raster layers using a sliding rectangular window. It also includes functions to quantize a raster into grey levels as well as tabulate a glcm and calculate glcm texture metrics for a matrix.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**BugReports** <https://github.com/ailich/GLCMTextures/issues>

**Depends** terra

**SystemRequirements** C++17

**LinkingTo** Rcpp, RcppArmadillo

**Imports** Rcpp, raster

**Suggests** knitr, rmarkdown

**URL** <https://ailich.github.io/GLCMTextures/>,  
<https://github.com/ailich/GLCMTextures>

**VignetteBuilder** knitr

**NeedsCompilation** yes

**Author** Alexander Ilich [aut, cre] (<<https://orcid.org/0000-0003-1758-8499>>)

**Maintainer** Alexander Ilich <[ailich@usf.edu](mailto:ailich@usf.edu)>

**Repository** CRAN

**Date/Publication** 2023-11-10 21:23:24 UTC

## R topics documented:

glcm_metrics	2
glcm_textures	3
make_glcm	5
quantize_raster	6

<b>Index</b>	<b>8</b>
--------------	----------

---

glcm_metrics	<i>Calculates the GLCM Texture Metrics from a GLCM</i>
--------------	--

---

### Description

Calculates the GLCM Texture Metrics from a GLCM

### Usage

```
glcm_metrics(
  GLCM,
  metrics = c("glcm_contrast", "glcm_dissimilarity", "glcm_homogeneity", "glcm_ASM",
             "glcm_entropy", "glcm_mean", "glcm_variance", "glcm_correlation")
)
```

### Arguments

GLCM	A numeric matrix representing a Normalized GLCM.
metrics	A vector of texture metrics to return. Valid entries include "glcm_contrast", "glcm_dissimilarity", "glcm_homogeneity", "glcm_ASM", "glcm_entropy", "glcm_mean", "glcm_variance", "glcm_correlation".

### Value

GLCM based texture measures as a numeric vector.

### References

Hall-Beyer, M., 2017. GLCM Texture: A Tutorial v. 3.0. University of Calgary, Alberta, Canada.  
 Haralick, R.M., Shanmugam, K., Dinstein, I., 1973. Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics 610–621. <https://doi.org/10.1109/TSMC.1973.4309314>

### Examples

```
test_matrix<- matrix(data=c(2,0,1,3,0,0,0,3,2), nrow = 3, ncol=3)
horizontal_glcm<- make_glcm(test_matrix, n_levels = 4,
  shift = c(1,0), normalize = TRUE)
metrics<-glcm_metrics(horizontal_glcm, metrics= c("glcm_contrast",
  "glcm_dissimilarity", "glcm_homogeneity", "glcm_ASM",
  "glcm_entropy", "glcm_mean", "glcm_variance", "glcm_correlation"))
```

glcm\_textures

*Calculates GLCM texture metrics of a Raster Layer***Description**

Calculates GLCM texture metrics of a RasterLayer over a sliding rectangular window

**Usage**

```
glcm_textures(
  r,
  w = c(3, 3),
  n_levels,
  shift = list(c(1, 0), c(1, 1), c(0, 1), c(-1, 1)),
  metrics = c("glcm_contrast", "glcm_dissimilarity", "glcm_homogeneity", "glcm_ASM",
    "glcm_entropy", "glcm_mean", "glcm_variance", "glcm_correlation"),
  quantization,
  min_val = NULL,
  max_val = NULL,
  na.rm = FALSE,
  include_scale = FALSE,
  filename = NULL,
  overwrite = FALSE,
  wopt = list()
)
```

**Arguments**

<code>r</code>	A single layer SpatRaster or RasterLayer. If already quantized set quantization to "none". The valid range of values for a quantized raster is from 0 to n_levels-1 (e.g. a raster with 32 grey levels would have a valid range of 0-31).
<code>w</code>	A vector of length 2 specifying the dimensions of the rectangular window to use where the first number is the number of rows and the second number is the number of columns. Window size must be an odd number.
<code>n_levels</code>	Number of grey levels used in the quantization (Typically set to 16 or 32).
<code>shift</code>	A vector of length 2, or a list of vectors each of length 2 specifying the relationship between neighboring pixel to the reference pixel. The first number represents the shift in the x direction and the second number represents the shift in the y direction, where up and right are positive. For example c(1,0) is the pixel directly to the right. The GLCM is made symmetrical by counting each pair twice, once "forwards" and once "backwards" by interchanging reference and neighbor pixels. Therefore a shift directly to the right c(1,0) is equivalent to a shift directly to the left c(-1,0). To average over "all directions" you can use shift=list(c(1,0), c(1,1), c(0,1), c(-1,1)), which is the default.

<code>metrics</code>	A vector of glcm texture metrics to return. Valid entries include "glcm_contrast", "glcm_dissimilarity", "glcm_homogeneity", "glcm_ASM", "glcm_entropy", "glcm_mean", "glcm_variance", "glcm_correlation".
<code>quantization</code>	quantization method (either "equal range", "equal prob", or "none"). "equal range" quantization will create bins that cover a range of equal size. "equal prob" performs equal probability quantization and will use quantiles to create bins with approximately equal number of samples. "none" means the layer has already been quantized.
<code>min_val</code>	minimum value for equal range quantization (if not supplied, the minimum value of the raster is used)
<code>max_val</code>	maximum value for equal range quantization (if not supplied, the maximum value of the raster is used)
<code>na.rm</code>	a logical value indicating whether NA values should be stripped before the computation proceeds (default=FALSE)
<code>include_scale</code>	Logical indicating whether to append window size to the layer names (default = FALSE).
<code>filename</code>	character Output filename. Can be a single filename, or as many filenames as there are layers to write a file for each layer
<code>overwrite</code>	logical. If TRUE, filename is overwritten (default is FALSE).
<code>wopt</code>	list with named options for writing files as in <code>writeRaster</code>

**Value**

a `SpatRaster` or `Raster*` Object

**References**

Hall-Beyer, M., 2017. GLCM Texture: A Tutorial v. 3.0. University of Calgary, Alberta, Canada.

Haralick, R.M., Shanmugam, K., Dinstein, I., 1973. Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics 610–621. <https://doi.org/10.1109/TSMC.1973.4309314>

**Examples**

```
r<- rast(volcano, extent= ext(2667400, 2667400 + ncol(volcano)*10,
6478700, 6478700 + nrow(volcano)*10), crs = "EPSG:27200")
txt <- glcm_textures(r, w = c(3,5), n_levels = 16,
quantization = "equal prob", shift = list(c(1, 0), c(1, 1),
c(0, 1), c(-1, 1)))
plot(txt)
```

---

make_glcm	<i>Creates a symmetrical normalized GLCM for a given matrix and shift</i>
-----------	---

---

### Description

Creates a symmetrical normalized GLCM for a given matrix and shift

### Usage

```
make_glcm(x, n_levels, shift, na.rm = FALSE, normalize = TRUE)
```

### Arguments

x	a matrix of integers representing quantized values. The valid range of values is from 0 to n_levels-1 (e.g. a matrix with 32 grey levels would have a valid range of 0-31).
n_levels	Number of grey levels used in the quantization
shift	A vector of length 2 specifying the relationship between neighboring pixel to the reference pixel. The first number represents the shift in the x direction and the second number represents the shift in the y direction, where up and right are positive. For example c(1,0) is the pixel directly to the right. The GLCM is made symmetrical by counting each pair twice, once "forwards" and once "backwards" by interchanging reference and neighbor pixels. Therefore a shift directly to the right c(1,0) is equivalent to a shift directly to the left c(-1,0)
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds (default=FALSE)
normalize	a logical specifying whether to normalize the counts to probabilities by dividing by the sum of the GLCM (TRUE, the default) or to express the GLCM as counts (FALSE)

### Value

A symmetric GLCM as a matrix

### References

Hall-Beyer, M., 2017. GLCM Texture: A Tutorial v. 3.0. University of Calgary, Alberta, Canada.  
 Haralick, R.M., Shanmugam, K., Dinstein, I., 1973. Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics 610–621. <https://doi.org/10.1109/TSMC.1973.4309314>

### Examples

```
test_matrix<- matrix(data=c(2,0,1,3,0,0,0,3,2), nrow = 3, ncol=3)
# Tabulate a GLCM of counts
horizontal_glcm_counts<- make_glcm(test_matrix, n_levels = 4, shift = c(1,0), normalize = FALSE)
# Calculate a normalized GLCM of probabilities
horizontal_glcm_norm<- make_glcm(test_matrix, n_levels = 4, shift = c(1,0), normalize = TRUE)
```

---

quantize\_raster                      *Quantizes raster to a set number of discrete levels*

---

### Description

Quantizes raster to a set number of discrete levels starting at 0. There are 2 methods of quantization are available: "uniform" and "equal prob"

### Usage

```
quantize_raster(
  r,
  n_levels,
  method,
  min_val = NULL,
  max_val = NULL,
  filename = NULL,
  overwrite = FALSE,
  wopty = list()
)
```

### Arguments

r	A single layer SpatRaster or RasterLayer.
n_levels	number of levels to quantize to
method	quantization method (either "equal range" or "equal prob"). "equal range" quantization will create bins that cover a range of equal size. "equal prob" performs equal probability quantization and will use quantiles to create bins with approximately equal number of samples.
min_val	minimum value for equal range quantization (if not supplied, the minimum value of the raster is used)
max_val	maximum value for equal range quantization (if not supplied, the maximum value of the raster is used)
filename	character Output filename.
overwrite	logical. If TRUE, filename is overwritten (default is FALSE).
wopty	list with named options for writing files as in writeRaster

### Details

Equal probability quantization is the method recommended in Haralick et al., 1973. However, equal range may be more desirable if making comparisons across several different rasters where you need the gray levels to correspond in a consistent way to the original data, as you can supply the global max/min or the theoretical max/min values that could occur. When equal probability quantization is used, quantiles are generated using type 8 as recommended by Hyndman and Fan (1996). This method provides estimates that are approximately median-unbiased regardless of the distribution of x.

**Value**

a single layer `SpatRaster` or `RasterLayer` with integer values ranging from 0 to `n_levels-1`

**References**

Haralick, R.M., Shanmugam, K., Dinstein, I., 1973. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics* 610–621. <https://doi.org/10.1109/TSMC.1973.4309314>

Hyndman, R.J., Fan, Y., 1996. Sample Quantiles in Statistical Packages. *The American Statistician* 50, 361–365. <https://doi.org/10.1080/00031305.1996.10473566>

**Examples**

```
r<- rast(volcano, extent= ext(2667400, 2667400 + ncol(volcano)*10,  
6478700, 6478700 + nrow(volcano)*10),  
crs = "EPSG:27200")  
rq1 <- quantize_raster(r = r, n_levels = 16, method = "equal prob")  
rq2 <- quantize_raster(r = r, n_levels = 16, method = "equal range")
```

# Index

[glcm\\_metrics](#), 2  
[glcm\\_textures](#), 3

[make\\_glc](#), 5

[quantize\\_raster](#), 6