Package ‘glcm’

August 29, 2016

Version 1.6.1
Date 2016-03-08
Title Calculate Textures from Grey-Level Co-Occurrence Matrices (GLCMs)
Maintainer Alex Zvoleff <azvoleff@conservation.org>
Depends R (>= 2.10.0)
Imports Rcpp (>= 0.11.0)
Suggests raster, testthat (>= 0.8.0)
LinkingTo Rcpp, RcppArmadillo
Description Enables calculation of image textures derived from grey-level co-occurrence matrices (GLCMs). Supports processing images that cannot fit in memory.
License GPL (>= 3)
URL http://www.azvoleff.com/glcm
BugReports https://github.com/azvoleff/glcm/issues
LazyData true
Encoding UTF-8
RoxygenNote 5.0.1
NeedsCompilation yes
Author Alex Zvoleff [aut, cre]
Repository CRAN
Date/Publication 2016-03-09 01:06:32

`R` topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>glcm-package</td>
<td>2</td>
</tr>
<tr>
<td>calc_texture</td>
<td>2</td>
</tr>
<tr>
<td>expected_textures_3x3_1x1</td>
<td>3</td>
</tr>
<tr>
<td>expected_textures_5x3_n1xn2</td>
<td>3</td>
</tr>
</tbody>
</table>
calc_texture

Description

Enables calculation of image textures derived from grey-level co-occurrence matrices (GLCMs) in R. The texture calculation is coded in C++ to optimize computation time.

Author(s)

Alex Zvoleff, <azvoleff@conservation.org>

calc_texture

Description

This function is called by the `glcm` function. It is not intended to be used directly.

Usage

calc_texture(rast, n_grey, window_dims, shift, statistics, na_opt, na_val)

Arguments

- `rast`: a matrix containing the pixels to be used in the texture calculation
- `n_grey`: number of grey levels to use in texture calculation
- `window_dims`: 2 element list with row and column dimensions of the texture window
- `shift`: a matrix where each row gives an (x, y) shift to use when computing co-occurrence matrices. Textures will be calculated for each shift, and the average over all shifts will be returned.
- `statistics`: a list of strings naming the texture statistics to calculate
- `na_opt`: one of "ignore", "center", or "any"
- `na_val`: what value to use to fill missing values on edges or where necessary due to chosen `na_opt` value

Value

A list of length equal to the length of the `statistics` input parameter, containing the selected textures measures.
GLCM textures calculated in EXELIS ENVI (for testing purposes)

Description
This is the output from running a "co-occurrence measures" calculation to calculate GLCM textures in EXELIS ENVI from the test_raster included in the glcm package. The following settings were used: window size 3x3; co-occurrence shift 1 row (y in ENVI), 1 column (x in ENVI); greyscale textures to compute: mean, variance, homogeneity, contrast, dissimilarity, entropy, second moment, correlation.

See Also
- expected_textures_5x7_2x3
- expected_textures_5x3_1x1

GLCM textures calculated in EXELIS ENVI (for testing purposes)

Description
This is the output from running a "co-occurrence measures" calculation to calculate GLCM textures in EXELIS ENVI from the test_raster included in the glcm package. The following settings were used: window size 5x3; co-occurrence shift -1 row (y in ENVI), -2 columns (x in ENVI); greyscale quantization levels 32; textures to compute: mean, variance, homogeneity, contrast, dissimilarity, entropy, second moment, correlation.

See Also
- expected_textures_3x3_1x1
- expected_textures_5x7_2x3

GLCM textures calculated in EXELIS ENVI (for testing purposes)

Description
This is the output from running a "co-occurrence measures" calculation to calculate GLCM textures in EXELIS ENVI from the test_raster included in the glcm package. The following settings were used: window size 5x7; co-occurrence shift 2 rows (y in ENVI), 3 columns (x in ENVI); greyscale textures to compute: mean, variance, homogeneity, contrast, dissimilarity, entropy, second moment, correlation.
glcm

Image texture measures from grey-level co-occurrence matrices (GLCM)

Description

This function supports calculating texture statistics derived from grey-level co-occurrence matrices (GLCMs). The default textures are calculated using a 45 degree shift. See Details for other options.

Usage

```r
glcm(x, n_grey = 32, window = c(3, 3), shift = c(1, 1), statistics =
c("mean", "variance", "homogeneity", "contrast", "dissimilarity", "entropy",
"second_moment", "correlation"), min_x=NULL, max_x=NULL, na_opt="any",
na_val=NA, scale_factor=1, asinteger=FALSE)
```

Arguments

- `x`: a RasterLayer or matrix
- `n_grey`: number of grey levels to use in texture calculation
- `window`: the window size to consider for texture calculation as a two element integer vector (number of rows, number of columns)
- `shift`: a list or matrix specifying the shift to use. See Details.
- `statistics`: A list of GLCM texture measures to calculate (see Details).
- `min_x`: minimum value of input RasterLayer (optional, glcm will calculate if not supplied). Useful when running glcm over blocks of a raster.
- `max_x`: maximum value of input RasterLayer (optional, glcm will calculate if not supplied). Useful when running glcm over blocks of a raster.
- `na_opt`: How to handle NA values in x. Can be set to "ignore", "any" or "center". If set to "any", all textures statistics for a given pixel will be set to NA if there are any NA values in the window around that pixel. If set to "center" this will only occur if the center value is an NA. If set to "ignore", NA values in window will be ignored.
- `na_val`: the value to use to fill NA values on edges of x where textures cannot be calculated due to the window falling outside of the image, and as necessary depending on the chosen na_opt.
- `scale_factor`: factor by which to multiply results. Useful if rounding results to integers (see asinteger argument).
- `asinteger`: whether to round results to nearest integer. Can be used to save space by saving results as, for example, an 'INT2S' raster.

See Also

eXpected_textures_3x3_1x1 expected_textures_5x3_1xn2
Details

The statistics parameter should be a list, and can include any (one or more) of the following: 'mean', 'mean_ENVI', 'variance', 'variance_ENVI', 'homogeneity', 'contrast', 'dissimilarity', 'entropy', 'second_moment', and/or 'correlation'. By default all of the statistics except for "mean_ENVI" and "variance_ENVI" will be returned.

shift can be one of:

1. a two element integer vector giving the shift (Q in Gonzalez and Woods, 2008), as (number of rows, number of columns).
2. a list of integer vectors of length 2 specifying multiple (row, col) shifts over which to calculate the GLCM textures. For example: shift=list(c(1,1), c(-1,-1))
3. a matrix with two columns specifying, in rows, multiple (row, col) shifts over which to calculate the GLCM textures. For example: shift=matrix(c(1,1,-1,-1), byrow=TRUE, ncol=2)

If multiple shifts are supplied, glcm will calculate each texture statistic using all the specified shifts, and return the mean value of the texture for each pixel. To calculate GLCM textures over "all directions" (in the terminology of commonly used remote sensing software), use: shift=list(c(0,1), c(1,1), c(1,0), c(1,-1)). This will calculate the average GLCM texture using shifts of 0 degrees, 45 degrees, 90 degrees, and 135 degrees.

Value

A RasterLayer or RasterStack with the requested GLCM texture measures.

References


Examples

```r
## Not run:
require(raster)
# Calculate GLCM textures using default 90 degree shift
textures_shift90 <- glcm(raster(LSTSR_1986, layer=1))
plot(textures_shift90)

# Calculate GLCM textures over all directions
textures_all_dir <- glcm(raster(LSTSR_1986, layer=1),
shift=list(c(0,1), c(1,1), c(1,0), c(1,-1)))
plot(textures_all_dir)
## End(Not run)
```
L5TSR_1986  
*Landsat 5 Surface Reflectance Image from February 6, 1986 (path 15, row 53)*

**Description**

Portion of Landsat 5 Surface Reflectance image from the Landsat Climate Data Record archive. This subset of the image includes only bands 1–4.

---

test_raster  
*_Randomly generated 100x100 test image*_

**Description**

Used in testing the output from the GLCM texture statistics C++ code.

**Examples**

```r
# The image was generated with the following code:
require(raster)
set.seed(0)
test_matrix <- matrix(runif(100)*32, nrow=10)
test_raster <- raster(test_matrix, crs=’+init=EPSG:4326’)
test_raster <- cut(test_raster, seq(0, 32))
```
Index

*Topic package
  glcm-package, 2

calc_texture, 2

expected_textures_3x3_1x1, 3, 3, 4
expected_textures_5x3_n1xn2, 3, 3, 4
expected_textures_5x7_2x3, 3, 3

glcm, 2, 4
glcm-package, 2

L5TSR_1986, 6
test_raster, 6