

Package ‘focusmapr’

June 15, 2014

Type Package

Title computation of Focus Maps from a set of input raster or vector layers

Version 1.0

Date 2014-06-11

Author Massimiliano Pittore - GFZ-Potsdam

Maintainer <pittore@gfz-potsdam.de>

Description the package implements the concept of Focus Map, a raster description of the combined spatial importance of a set of input layers. The Focus Map can be interpreted as a spatial distribution of data sampling probability given the input layers and the weights of their combination. The Focus Map is generated by a “normalization” (or “mapping”) and a “pooling” operations. The package has been implemented in the framework of SENSUM FP7-SPACE Project.

License GPL

Depends rgdal (>= 0.8.16), raster (>= 2.2.31), spatstat (>= 1.34.0)

R topics documented:

focusmapr-package	2
FocusMap	2
LoadRasterLayers	3
MosaicRasters	4
NormalizeLayer	5
ResampleLayers	6
SamplingPoints	6
Vec2Raster	7
Index	8

focusmapr-package	<i>Computation of Focus Maps</i>
-------------------	----------------------------------

Description

The package implements the concept of Focus Maps, a raster description of the spatial data sampling probability (or "importance") as a result of a combination of several raster or vector input layers. Every input layer is "mapped" (usually normalized) and "pooled"

Details

Package:	focusmapr
Type:	Package
Version:	1.0
Date:	2014-06-11
License:	GPL

Author(s)

Massimiliano Pittore

Examples

```
# loading two raster layers from disk
r1 <- system.file("extdata/p1dens.tif", package="focusmapr")
r2 <- system.file("extdata/p2dens.tif", package="focusmapr")
layers <- LoadRasterLayers(c(r1,r2),resamp=TRUE)
# normalize the layers
norm_layers <- lapply(layers,NormalizeLayer)
# create the focus map with a multiplicative pooling and inequal weights
focmap <- FocusMap(norm_layers,pooling="loglinear",weights=c(.7,.3))
```

FocusMap	<i>create a new raster object representing a Focus Map from a list of input layers</i>
----------	--

Description

the function applies a pooling method to the list of input rasters, returning the result in form of a raster with same extent and resolution. The pooling methods are specified by the argument `pooling`

Usage

```
FocusMap(layers, pooling = "loglinear", weights)
```

Arguments

layers	A list of raster objects with same resolution
pooling	The desired pooling method. Possible values are \itemlinearadditive pooling \itemloglinearmultiplicative pooling
weights	a vector of (real numbers) weights. The lenght of the vector is equal to the number of input layers. Weights sum to one for the linear pooling, and are not constrained in case of loglinear pooling.

Value

a raster with same extent of the intersection of input layers representing the estimated focus map

Author(s)

Massimiliano Pittore

Examples

```
# loading two raster layers from disk
r1 <- system.file("extdata/p1dens.tif", package="focusmapr")
r2 <- system.file("extdata/p2dens.tif", package="focusmapr")
layers <- LoadRasterLayers(c(r1,r2),resamp=TRUE)
# normalize the layers
norm_layers <- lapply(layers,NormalizeLayer)

# create the focus map with a multiplicative pooling and inequal weights
focmap1 <- FocusMap(norm_layers,pooling="loglinear",weights=c(.7,.3))

# create another focus map with a additive pooling and equal weights
focmap2 <- FocusMap(norm_layers,pooling="linear",weights=c(.5,.5))
```

LoadRasterLayers	<i>bulk load a set of raster layers from file</i>
------------------	---

Description

Given a list of filenames (including path) the function loads the different raster layers, returning a list or object of type raster (from package raster). If needed, the function performs flippng, resample and reprojection of the individual layer to obtain an harmonized list.

Usage

```
LoadRasterLayers(layers_paths, repro = FALSE, resamp = FALSE, flip = FALSE)
```

Arguments

<code>layers_paths</code>	list of filenames (including complete path) for the layers to be loaded. Supported file types are the 'native' raster package format and those that can be read via <code>rgdal</code> http://cran.at.r-project.org/web/packages/raster/raster.pdf#Rfn.readGDALrgdal .
<code>repro</code>	logical, if YES reprojects the layer into a default WGS84 reference system
<code>resamp</code>	logical, if YES resamples all the raster layers following the resolution of the first layer. ResampleLayers .
<code>flip</code>	logical, if YES a flipping of the input layer around the 'y' axis is performed. The flip options is handy to fix the occasional flipping of rasters created with QGIS.

Value

list of raster objects ([raster](#)).

Author(s)

Massimiliano Pittore

Examples

```
# reads two raster files from a folder

# N.B.: For your own files, omit the system.file and package="focusmapr" bits
# these are just to get the path to files installed with the package

r1 <- system.file("extdata/p1dens.tif", package="focusmapr")
r2 <- system.file("extdata/p2dens.tif", package="focusmapr")
layers<-LoadRasterLayers(c(r1,r2),resamp=TRUE)
```

MosaicRasters

Create a tile mosaic of input rasters

Description

Builds a single mosaic from a set of raster tiles. It uses the `do.Mosaic` function of the package `raster` ([raster](#))

Usage

```
MosaicRasters(paths)
```

Arguments

paths a list of input rasters

Value

a single mosaic raster

Author(s)

Massimiliano Pittore

NormalizeLayer	<i>Performs a mapping of input layer through normalization</i>
----------------	--

Description

Normalizes a single input raster. Normalization either simply re-scales the raster values to fit the interval [0,1], or uses a rejection bounds specified by the argument `rej`. If rejection bounds are used, normalization is applied only within the percentiles defined by the rejection bounds. Rejection bounds are useful to avoid bias from outliers.

Usage

```
NormalizeLayer(rast, norm_type = "linear", rej = NULL)
```

Arguments

`rast` Input raster layer

`norm_type` Type of normalization. Currently only `norm_type="linear"` is implemented.

`rej` Rejection bounds. Specify two percentiles outside which normalization is not applied. if `rej=c(0,1)` is applied, this is equivalent to simple normalization.

Value

Normalized raster layer

Author(s)

Massimiliano Pittore

Examples

```
# loading two raster layers from disk
r1 <- system.file("extdata/p1dens.tif", package="focusmapr")
r2 <- system.file("extdata/p2dens.tif", package="focusmapr")
layers <- LoadRasterLayers(c(r1,r2),resamp=TRUE)
# normalize the layers with a 0.01 rejection bound
norm_layers<-lapply(layers,FUN=function(x) NormalizeLayer(x,rej=c(0.01,0.99)))
```

ResampleLayers	<i>Resample the input layers (i.e. changes resolution)</i>
----------------	--

Description

the function resamples all layers in the list except the reference, which is specified by the `ref_index` argument. The procedure changes the resolution of the processed layers.

Usage

```
ResampleLayers(rasters, ref_index = NULL)
```

Arguments

<code>rasters</code>	a list of raster objects
<code>ref_index</code>	index of the raster layer to be used as reference for resampling. By default=1 (first element of the list)

Value

a list of layers with updated resolution

Author(s)

Massimiliano Pittore

Examples

```
r1 <- system.file("extdata/p1dens.tif", package="focusmapr")
r2 <- system.file("extdata/p2dens.tif", package="focusmapr")
layers <- LoadRasterLayers(c(r1,r2),resamp=TRUE)
# use the resolution of the second layer for resampling the first
resamp_layers<-ResampleLayers(layers,ref_index=2)
```

SamplingPoints	<i>Generates a set of sampling points according to a density distribution</i>
----------------	---

Description

A set of spatial points is generated using the input raster as spatial density distribution. The function created an inhomogeneous Poisson Point Process to generate the points. A scaling coefficient is used to control the final amount of sampling points.

Usage

```
SamplingPoints(rast, coef)
```

Arguments

rast	input raster defining the spatial density of probability of sampling. A FocusMap focus map is typically used
coef	scaling coefficient to tune the number of generated points

Value

A set of sampling points.

Author(s)

Massimiliano Pittore

Vec2Raster

Convert a vector (shapefile) layer into a raster layer

Usage

```
Vec2Raster(vec, ras_attr, res)
```

Arguments

vec	spatial object of type <code>SpatialPolygonsDataFrame</code> with a set of numeric attributes (at least one)
ras_attr	attribute of the vector layer used to generate the raster. The attribute must exist, and be a number
res	resolution of the output raster. This argument must be provided. The resolution must be specified in the same unit of measure related to the input CRS.

Details

The drivers available will depend on the installation of GDAL/OGR, and can vary; the `ogrDrivers()` function shows which are available, and which may be written (but all are assumed to be readable). Note that stray files in data source directories (such as *.dbf) may lead to suprious errors that accompanying *.shp are missing.

Value

a raster with same extent as input vector, and resolution specified by the user.

Author(s)

Massimiliano Pittore

Index

*Topic **\textasciitildenormalization**

NormalizeLayer, [5](#)

*Topic **focus map**

FocusMap, [2](#)

*Topic **mosaic**

MosaicRasters, [4](#)

*Topic **package**

focusmapr-package, [2](#)

*Topic **raster**

LoadRasterLayers, [3](#)

*Topic **resampling**

ResampleLayers, [6](#)

*Topic **resolution**

ResampleLayers, [6](#)

*Topic **sampling**

SamplingPoints, [6](#)

*Topic **vector**

Vec2Raster, [7](#)

FocusMap, [2](#), [7](#)

focusmapr (focusmapr-package), [2](#)

focusmapr-package, [2](#)

<http://cran.at.r-project.org/web/packages/raster/raster.pdf#Rfn.readGDAL>,
[4](#)

LoadRasterLayers, [3](#)

MosaicRasters, [4](#)

NormalizeLayer, [5](#)

raster, [4](#)

ResampleLayers, [4](#), [6](#)

SamplingPoints, [6](#)

Vec2Raster, [7](#)