Package ‘landscapeR’

October 13, 2022

Title  Categorical Landscape Simulation Facility
Version 1.2
Date 2017-07-05
Copyright Natural Environment Research Council (NERC) and Centre for Ecology and Hydrology (CEH)
URL https://github.com/dariomasante/landscapeR
BugReports https://github.com/dariomasante/landscapeR/issues
Imports raster, Rcpp
Suggests knitr, rmarkdown
Description Simulates categorical maps on actual geographical realms, starting from either empty landscapes or landscapes provided by the user (e.g. land use maps). Allows to tweak or create landscapes while retaining a high degree of control on its features, without the hassle of specifying each location attribute. In this it differs from other tools which generate null or neutral landscapes in a theoretical space. The basic algorithm currently implemented uses a simple agent style/cellular automata growth model, with no rules (apart from areas of exclusion) and von Neumann neighbourhood (four cells, aka Rook case). Outputs are raster dataset exportable to any common GIS format.

License GPL (>= 3)
LazyLoad yes
LazyData TRUE
NeedsCompilation yes
RoxygenNote 6.0.1
VignetteBuilder knitr
LinkingTo Rcpp

Author Dario Masante [aut, cre]
Maintainer Dario Masante <dario.masante@gmail.com>
Repository CRAN
Date/Publication 2017-07-05 14:54:40 UTC
R topics documented:

landscapeR-package .................................................. 2
expandClass .......................................................... 3
makeClass ........................................................... 4
makeLine .............................................................. 5
makePatch ............................................................. 6
rmSingle ............................................................... 7

Index 9

landscapeR-package  landscapeR: A landscape simulator for R.

Description

This package is aimed at simulating categorical landscapes on actual geographical realms, starting from either empty landscapes, or landscapes provided by the user (e.g. land use maps). landscapeR allows to tweak or create landscapes while retaining a high degree of control on its features, without the hassle of specifying each location attribute. In this it differs from other tools which generate null or neutral landscape in a theoretical space. The basic algorithm currently implemented uses a simple agent style/cellular automata growth model, with no rules (apart from areas of exclusion). Outputs are raster dataset exportable to any common GIS format.

Details

Check out the vignette illustrating the use of landscapeR.
Also: https://github.com/dariomasante/landscapeR

landscapeR functions

- makePatch creates a single patch in the landscape.
- makeClass creates a group of patches belonging to the same class.
- expandClass expands an existing class of patches.
- makeLine creates a linear patch.
- rmSingle removes single tones from patches and background.

Author(s)

Dario Masante
**expandClass**

Expand an existing class of patches.

**Description**

Expand an existing class of patches.

**Usage**

```r
expandClass(context, class, size, bgr = 0, pts = NULL)
```

**Arguments**

- **context**: Raster object or matrix, an empty landscape raster or a mask indicating where the patch cannot be generated (see bgr below).
- **class**: The raster value of class (or patch) to expand.
- **size**: integer. Size of expansion, as number of raster cells.
- **bgr**: integer. The background available where expansion is allowed (i.e. shrinking classes).
- **pts**: integer or matrix. The seed point location around which the patches are built (random points are given by default). It can be an integer, as indexes of the cells in the raster, or a two columns matrix indicating x and y coordinates.

**Value**

A RasterLayer object. If `rast=FALSE` returns a list of vectors, each containing the `context` cells assigned to each patch.

**Examples**

```r
code
```

```r
code
```

```r
code
```

```r
code
```

## This function can be used to mimic shapes, by providing a skeleton:
```r
code
```
makeClass

Create a class of patches.

Description

Create a class of patches.

Usage

makeClass(context, npatch, size, pts = NULL, bgr = 0, edge = FALSE, rast = TRUE, val = 1)

Arguments

context Raster object, a raster of an empty landscape or a mask, indicating where the patch cannot be generated (see bgr argument).
npatch number of patches per class
size integer. The size of patches, as number of raster cells. A single integer can be provided, in which case all patches will have that size.
pts integer or matrix. The seed point location around which the patches are built (random points are given by default). It can be an integer, as indexes of the cells in the raster, or a two columns matrix indicating x and y coordinates.
bgr integer. A single value of background cells, where a patch can be generated (default is zero). Cells/classes which cannot be changed must have a different value.
edge logical. Should the vector of edge cells of the patch be returned?
rast logical. If TRUE returns a Raster object, otherwise a vector of cell numbers where the patch occurs
val integer. The value to be assigned to patch cells, when rast=TRUE

Details

The patches created can be contiguous, therefore resembling a single patch with size equal to the sum of contiguous cells. The patches are created starting from the seed points (if provided) and iteratively sampling randomly neighbouring cells at the edge of the patch, according to von Neumann neighbourhood (four cells, aka Rook case). There is a tolerance of +/- 3 cells from the patch size declared in size argument.

Value

A RasterLayer object, or a vector of cell numbers if rast=FALSE.
Examples

library(raster)

m = matrix(0, 33, 33)
r = raster(m, xmn=0, xmx=10, ymn=0, ymx=10)
num = 5
size = 15
rr = makeClass(r, num, size)
plot(rr)

### Create a class of three patches of given size at three corners of the spatial context
size = c(10, 50, 200)
pts = c(1, 33, 1089)
rr = makeClass(r, 3, size, pts)
plot(rr)

makeLine

Create a linear patch (beta version).

Description

Create a linear patch, setting direction and convolution. The higher the convolution degree, the weaker the linear shape (and direction).

Usage

makeLine(context, size, direction = NULL, convol = 0.5, spt = NULL,
bgr = 0, edge = FALSE, rast = FALSE, val = 1)

Arguments

context Raster object or matrix, an empty landscape raster or a mask indicating where the patch cannot be generated (see bgr below).

size integer. Size of the patch to be generated, as number of raster cells.

direction numeric. The direction towards which the patch will point and grow, expressed in degrees between 0 and 360.

convol numeric. Level of convolution to be assigned to the patch (default is convol=0.5). A value between 0 and 1, where 0 is no convolution at all (basically a straight line) and 1 is maximum convolution (i.e. tends to form a circular patch).

spt integer or matrix. The seed point location around which the patch is generated (a random point is given by default). It can be an integer, as index of the cell in the raster, or a two columns matrix indicating x and y coordinates (an integer vector of length 2 is accepted too).

bgr integer. A single value of background cells, where a patch can be generated (default is zero). Cells/classes which cannot be changed must have a different value.
edge logical. Should the vector of edge cells of the patch be returned?
rast logical. If TRUE returns a Raster object, otherwise a vector of cell numbers where the patch occurs
val integer. The value to be assigned to patch cells, when rast=TRUE

Details
For any values of convol > 0.8, no big differences are observed noted. Also direction is progressively lost as convolution increases.

Value
A vector of raster cell numbers, or a RasterLayer object if rast=TRUE. If edge=TRUE a list of two vectors is returned: one for the inner raster cells and the second for cells at the edge of the patch.

Examples
library(raster)
r <- matrix(0,33,33)
r <- raster(r, xmn=0, xmx=10, ymn=0, ymx=10)
plot(makeLine(r, size=50, spt = 545, direction=45, convol=0.05, rast=TRUE))

makePatch Create a single patch

Description
Function will create a single patch. NOTE: makeClass should be used preferably when creating a single patch, as better error and exception handling is provided.

Usage
makePatch(context, size, spt = NULL, bgr = 0, edge = FALSE, rast = FALSE, val = 1)

Arguments
context Raster object or matrix, an empty landscape raster or a mask indicating where the patch cannot be generated (see bgr below).
size integer. Size of the patch to be generated, as number of raster cells.
spt integer or matrix. The seed point location around which the patch is generated (a random point is given by default). It can be an integer, as index of the cell in the raster, or a two columns matrix indicating x and y coordinates (an integer vector of length 2 is accepted too).
**rmSingle**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bgr</code></td>
<td>integer. A single value of background cells, where a patch can be generated (default is zero). Cells/classes which cannot be changed must have a different value.</td>
</tr>
<tr>
<td><code>edge</code></td>
<td>logical. Should the vector of edge cells of the patch be returned?</td>
</tr>
<tr>
<td><code>rast</code></td>
<td>logical. If TRUE returns a Raster object, otherwise a vector of cell numbers where the patch occurs</td>
</tr>
<tr>
<td><code>val</code></td>
<td>integer. The value to be assigned to patch cells, when <code>rast=TRUE</code></td>
</tr>
</tbody>
</table>

**Details**

The patch is created starting from the seed point and iteratively sampling randomly neighbouring cells at the edge of the patch, according to von Neumann neighbourhood (four cells, aka Rook case). There is a tolerance of +/- 3 cells from the patch size declared in `size` argument. Argument `bgr` accepts a single value only, unlike `makeClass` that accepts multiple and should therefore preferred.

**Value**

A vector of raster cell numbers, or a RasterLayer object if `rast=TRUE`. If `edge=TRUE` a list of two vectors is returned: one for the inner raster cells and the second for cells at the edge of the patch.

**Examples**

```r
library(raster)
mtx = matrix(0, 33, 33)
r = raster(mtx, xmn=0, xmx=10, ymn=0, ymx=10)
patchSize = 500
rr = makePatch(r, patchSize, rast=TRUE)
plot(rr)

## Create a patch with value 3, starting from the centre cell
mtx = matrix(0, 33, 33)
r = raster(mtx, xmn=0, xmx=10, ymn=0, ymx=10)
newVal = 3
centre = 545
cells = makePatch(r, patchSize, centre)
r[cells] = newVal
plot(r)

## Now create a new patch with value 10 and size 100 inside the existing patch
rr = makePatch(r, 100, bgr=newVal, rast=TRUE, val=10)
plot(rr)
```

---

**rmSingle** *Remove single tones from patches*

**Description**

Patch creation algorithm can occasionally leave single cells scattered within patches. This function reduces the “salt-pepper” effect, identifying or correcting those cells.
Usage

`rmSingle(rst, rm = TRUE)`

Arguments

- `rst` input raster landscape.
- `rm` logical, if TRUE returns the raster without single tones cells, if FALSE a vector of numbers identifying the index of single tones cells.

Value

A raster without single tones cells. If `rm=FALSE`, it returns a vector of numbers identifying the index of single tones cells. The value assigned to single tone cells is picked from one of the four neighbouring cells, selected at random.

Examples

```r
library(raster)
m = matrix(0, 33, 33)
r = raster(m, xmn=0, xmx=10, ymn=0, ymx=10)
patchSize = 500

## Make a patch and introduce a single tone cell
r = makePatch(r, patchSize, spt=578, rast=TRUE)
r[578] = 0
plot(r)

## Now remove it
plot( rmSingle(r) )

## Single tones can be identified but not removed:
rmSingle(r, rm = FALSE)
```
Index

expandClass, 2, 3

landscapeR (landscapeR-package), 2
landscapeR-package, 2

makeClass, 2, 4
makeLine, 2, 5
makePatch, 2, 6

rmSingle, 2, 7