Package 'TileManager'

February 2, 2022

Type Package

Title Tile Manager

Version 0.4.1

Date 2022-02-01

Author Andrew Plowright

Maintainer Andrew Plowright <andrew.plowright@alumni.ubc.ca>

Description Tools for creating and detecting tiling schemes for geospatial datasets.

Depends R (>= 3.5.0)

License GPL (>= 3)

LazyData TRUE

Imports APfun, raster, rgeos, sp, utils, methods, graphics, stats, XML

RoxygenNote 7.1.2

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2022-02-02 14:40:05 UTC

R topics documented:

CHMdemo *Canopy height model demo*

Description

A small section of a canopy height model of a forest in British Columbia, Canada.

Usage

```
CHMdemo
```

Format

A RasterLayer

Cell values are equal to canopy height above ground (in meters)

---

length,tileScheme-method
Length

Description

Get the number of a cells contained in a 'tileScheme' object

Usage

```
## S4 method for signature 'tileScheme'
length(x)
```

Arguments

x	'tileScheme' object

plot *Plot*

Description

Plot a 'tileScheme' object

Usage

```
plot(x, y, ...)

## S4 method for signature 'tileScheme,ANY'
plot(x, labels = TRUE, add = FALSE, ...)
```

Arguments

x	a 'tileScheme' object
y	argument not used for 'tileScheme' objects
...	arguments t be passed to methods
labels	logical. Add row and column labels
add	logical. Plot 'tileScheme' on top of existing plot

show,tileScheme-method *Show*

Description

Print information about a 'tileScheme' object

Usage

```
## S4 method for signature 'tileScheme'
show(object)
```

Arguments

object	a 'tileScheme' object

tileDetector *Tile Detector*

Description

Function for detecting existing tiling scheme from a list of raster files or RasterLayers. This includes detecting buffers, row and column positions, and re-ordering based on tile location.

Usage

```
tileDetector(inRasters, roundCoords = 4)
```

Arguments

inRasters	a list of RasterLayers
roundCoords	numeric. Round input coordinates to this digit

tileLoad *Load Tile Scheme*

Description

Load a Tile Scheme to a SHP file. The file needs to be originally saved using `tileSave`, since some metadata (saved to an XML file) is required.

Usage

```
tileLoad(filepath)
```

Arguments

filepath	file path

tileSave *Save Tile Scheme*

Description

Save a Tile Scheme to a SHP file.

Usage

```
tileSave(ts, filepath, overwrite = FALSE)
```

Arguments

ts	tileScheme
filepath	file path
overwrite	logical. Overwrite existing file

tileScheme *Tile Scheme*

Description

This function aims to provide an all-in-one tool for creating tiling schemes, which includes options for overlapping buffers and methods for describing tile sizes in various ways (i.e. using either distance units or cell numbers).

Usage

```
tileScheme(
  input,
  tiledim,
  cells = FALSE,
  buffer = 0,
  bufferspill = FALSE,
  round = NA,
  roundDir = "out",
  crs = NULL,
  origin = NULL,
  removeEmpty = FALSE
)
```

Arguments

input	filename (character), Extent, Raster or a vector of four numbers
tiledim	numeric. Defines the 'x' and 'y' dimensions of each tile. By default, dimensions are in map units. If 'cells' is set to TRUE, then dimensions are in number of cells
cells	logical. If set to TRUE, tiledim and buffer dimensions will be in number of cells instead of map units
buffer	numeric. If set to >0, overlapping buffers will be created around each tile
bufferspill	logical. Default is FALSE, in which case the tiling grid will be pushed inwards so that the buffers of the outer tiles are within the extent of input. If set to TRUE, the buffers will extend outside of the extent of input
round	numeric. Round the extent of the input Extent to the number of digits specified here.
roundDir	character. The direction of the rounding, either in for inwards or out for outwards.
crs	character. PROJ4 string defining output coordinate reference system (CRS). If set to NULL, the function will attempt to get a CRS from input (only works if it is a raster). Set to NA to force the output to have no CRS.
origin	numeric. Optional vector of two numbers corresponding to a pair of coordinates to which the tiling scheme will be aligned. Cannot be used in conjunction with cells. The coordinates do not need to be within the extent of input
removeEmpty	logical. Default is FALSE. If set to TRUE, tiles containing only NA cell values will be removed from the tiling scheme. Can only be used when input is a Raster object.

Value

a 'tileScheme' object

Non-overlapping buffers

When processing a tiled dataset, using buffered tiles can help remove the edge effects along the individual tile borders. However, overlapping buffers generally need to be removed when recombining a series of tiles back into a single raster. Although this can be accomplished by using the unbuffered tile extent, this will also remove the buffered areas along the edge of the tile set. Once these unbuffered tiles are reassembled, the resulting raster will then be smaller than the original dataset before it was tiled.

This may not be a desirable result. The polygons located in the nbuffs slot will produce a set of polygons that correspond to the tile extents that conserve buffers only where they do not overlap onto neighboring tiles (i.e.: along the edge of the tile set). These polygons are useful for cropping out overlapping areas from buffered tiles in order to reassemble the tiles into a single raster.

Examples

```
## Not run:
ts1 <- tileScheme(CHMdemo, tiledim = c(50,50))
```

```
ts2 <- tileScheme(CHMdemo, tiledim = c(100,120), cells = TRUE)

ts3 <- tileScheme(CHMdemo, tiledim = 40, buffer = 5, origin = c(0.5, 0.5))

## End(Not run)
```

`tileScheme-class` *Tile Scheme class #'*

Description

Tile Scheme class #'

`[,tileScheme,character,ANY-method` *Subset*

Description

Subset tiles using the single bracket operator. Subset geometry (tiles, buffs or nbuffs) using the double brackets

Usage

```
## S4 method for signature 'tileScheme,character,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'tileScheme,numeric,numeric'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'tileScheme,numeric,missing'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'tileScheme,missing,numeric'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'tileScheme,character,missing'
x[[i, j, ..., exact = TRUE]]
```

Arguments

`x`	a 'tileScheme' object
`i, j, ...`	indices specifying elements to extract
`drop, exact`	arguments not used for 'tileScheme'

`$,tileScheme-method` *Get and set data*

Description

Get and set data

Usage

```
## S4 method for signature 'tileScheme'
x$name

## S4 replacement method for signature 'tileScheme'
x$name <- value
```

Arguments

`x`	a 'tileScheme' object
`name`	name of data column
`value`	vector of new data values

Index