Package 'rasterDT'

March 4, 2020

Type Package

Title Fast Raster Summary and Manipulation

Version 0.3.1

Date 2020-02-26

Author Joshua O'Brien

Maintainer Joshua O'Brien <joshmobrien@gmail.com>

Description Fast alternatives to several relatively slow 'raster' package functions. For large rasters, the functions run from 5 to approximately 100 times faster than the 'raster' package functions they replace. The 'fasterize' package, on which one function in this package depends, includes an implementation of the scan line algorithm attributed to Wylie et al. (1967) <doi:10.1145/1465611.1465619>.

License GPL (>= 2)

URL https://github.com/JoshOBrien/rasterDT/

BugReports https://github.com/JoshOBrien/rasterDT/issues/

Depends methods, raster, data.table

Imports fasterize, sf

Suggests rasterVis, rgdal

LazyData true

Encoding UTF-8

RoxygenNote 7.0.2

NeedsCompilation no

Repository CRAN

Date/Publication 2020-03-04 16:30:02 UTC

R topics documented:

`rasterDT-package` *Fast Raster Summary and Manipulation*

Description

Fast alternatives to several relatively slow 'raster' package functions. For large rasters, the functions run from 5 to approximately 100 times faster than the 'raster' package functions they replace. The 'fasterize' package, on which one function in this package depends, includes an implementation of the scan line algorithm attributed to Wylie et al. (1967) <doi:10.1145/1465611.1465619>.

Details

The DESCRIPTION file:

Package:	rasterDT
Type:	Package
Title:	Fast Raster Summary and Manipulation
Version:	0.3.1
Date:	2020-02-26
Author:	Joshua O'Brien
Maintainer:	Joshua O'Brien <joshmobrien@gmail.com>
Description:	Fast alternatives to several relatively slow 'raster' package functions. For large rasters, the functions run from
License:	GPL (>= 2)
URL:	https://github.com/JoshOBrien/rasterDT/
BugReports:	https://github.com/JoshOBrien/rasterDT/issues/
Depends:	methods, raster, data.table
Imports:	fasterize, sf
Suggests:	rasterVis, rgdal
LazyData:	true
Encoding:	UTF-8
RoxygenNote:	7.0.2

Index of help topics:

`cat_to_val` `Convert a Categorical Raster to a Value Raster`

crosstabDT	Speedy Raster Cross-tabulation
fasterizeDT	Polygon Rasterization Using Numeric, Factor, or Character Fields
freqDT	Speedy Raster Value Frequency Tabulation
rasterDT-package	Fast Raster Summary and Manipulation
subsDT	Speedy Raster Value Substitution
zonalDT	Speedy Zonal Statistics

Fast alternatives to several relatively slow raster package functions. For large rasters, the functions run from 5 to approximately 100 times faster than the raster package functions they replace.

Author(s)

Joshua O'Brien

Maintainer: Joshua O'Brien <joshmobrien@gmail.com>

`cat_to_val` *Convert a Categorical Raster to a Value Raster*

Description

Use a categorical raster's RAT to convert it to a continuous raster

Usage

```
cat_to_val(r, which = 2)
```

Arguments

`r`	A categorical raster with a RAT (returned by `levels(r)[[1]]`), whose first column contain an entry for every factor level present in the raster. At least one of the subsequent columns should contain numeric values to which each level should be converted.
`which`	An integer or character string giving the index or name of the column in `r`'s RAT with the numerical values to which each value in `r` should be mapped. Default value is `2`.

Value

A continuous raster with each category level in `r` replaced by its corresponding value.

Author(s)

Joshua O'Brien

Examples

```
r_cat <- raster(matrix(c(2, 2, 2, 1), ncol = 2))
levels(r_cat) <- data.frame(ID = c(1, 2),
                            VAL1 = c(0.1, 200),
                            VAL2 = c(33, 44))

## Second column of RAT is used by default
r_con1 <- cat_to_val(r_cat)
as.matrix(r_con1)

## Use 'which=' argument for conversion to another RAT column
r_con2 <- cat_to_val(r_cat, which = "VAL2")
as.matrix(r_con2)
```

`crosstabDT` *Speedy Raster Cross-tabulation*

Description

A fast `data.table`-based alternative to `raster::crosstab()`.

Usage

```
crosstabDT(x, y, digits = 0, long = FALSE, useNA = FALSE)
```

Arguments

`x`	A `Raster*` object
`y`	If `x` has just one layer, a `RasterLayer` object. Otherwise, if `x` is a multi-layered `RasterStack` or `RasterBrick`, this argument (if any) is unused.
`digits`	Integer. The number of digits for rounding the values before cross-tabulation. Default is `0`.
`long`	Logical. If `TRUE`, the results are returned in a 'long' format `data.table` instead of as a table. Default is `FALSE`.
`useNA`	Logical. Should the returned table or `data.table` include counts of `NA` values? Default is `FALSE`.

Value

Either a table or a `data.table` recording the frequency of each combination of raster values.

Author(s)

Joshua O'Brien

Examples

```
r <- raster(nc = 5, nr = 5)
r[] <- runif(ncell(r)) * 2
s <- setValues(r, runif(ncell(r)) * 3)
crosstabDT(r, s)

rs <- r/s
r[1:5] <- NA
s[20:25] <- NA
x <- stack(r, s, rs)
crosstabDT(x, useNA = TRUE, long = TRUE)
```

`fasterizeDT` *Polygon Rasterization Using Numeric, Factor, or Character Fields*

Description

A front end for `fasterize::fasterize()`, fixing several of its infelicities.

Usage

```
fasterizeDT(
  x,
  raster,
  field = NULL,
  fun = "last",
  background = NA_real_,
  by = NULL
)
```

Arguments

`x`	Either an `sf::sf()` object with a geometry column of `POLYGON` and/or `MULTIPOLYGON` objects or a `sp::SpatialPolygonsDataFrame` object.
`raster`	A `RasterLayer` object to be used as a template for the raster output.
`field`	Character. The name of a column in `x`, providing a value for each of the polygons rasterized. If `NULL` (the default), all polygons will be given a value of `1`.
`fun`	Character. The name of a function by which to combine overlapping polygons. Currently takes `"sum"`, `"first"`, `"last"`, `"min"`, `"max"`, `"count"`, or `"any"`. For more details, see `?fasterize::fasterize`.
`background`	Value to put in the cells that are not covered by any of the features of `x`. Default is `NA`.
`by`	Character string giving the name of a column in `x` by which to aggregate layers. If set, `fasterizeDT` will return a `RasterBrick` with as many layers as unique values of the `by` column.

Details

Unlike other functions in this package, `fasterizeDT()` does not use `data.table` to speed up its computations. Instead, it is a wrapper for `fasterize::fasterize()`, intended to address several of that function's limitations.

Most importantly, `fasterizeDT()` takes care to properly handle rasterization operations in which either the template `RasterLayer` or the selected polygon feature field is a factor. Specifically, it always returns a raster whose type (numeric or factor) and levels (if a factor) match that of the spatial polygon attribute indicated by its `field` argument. Second, when `field` specifies an attribute of class `"character"`, `fasterizeDT()` automatically converts it to a factor and returns a factor raster. In this, it is unlike both `fasterize::fasterize()` and `raster::rasterize()`. Finally, unlike `fasterize::fasterize()`, `fasterizeDT()` accepts as inputs either `sf::sf()` objects or `sp::SpatialPolygonsDataFrame` objects.

Value

A raster of the same size, extent, resolution and projection as the supplied raster template. Unlike `fasterize::fasterize()`, `fasterizeDT` returns a raster of the same type as the data in the column of `x` selected by the `field` argument.

Author(s)

Joshua O'Brien

Examples

```
## Load example polygons and prepare a template raster
SPDF <- rgdal::readOGR(system.file("external", package = "raster"), "lux")
llratio <- 1/cos(pi * mean(coordinates(SPDF)[, 2])/180)
rr <- raster(extent(SPDF),
             resolution = c(llratio * 0.01, 0.01),
             crs = proj4string(SPDF))

## An integer-valued field produces a numeric raster
rInt <- fasterizeDT(SPDF, rr, field = "ID_2")
plot(rInt, col = colorRampPalette(blues9)(12))

## A character-valued field returns a factor raster
rFac <- fasterizeDT(SPDF, rr, field = "NAME_2")
if (require(rasterVis)) {
    levelplot(rFac)
}
```

---

freqDT *Speedy Raster Value Frequency Tabulation*

Description

A fast `data.table`-based alternative to `raster::freq()`.

Usage

```
freqDT(x, ...)

## S4 method for signature 'RasterLayer'
freqDT(x, digits = 0, value = NULL, useNA = c("ifany", "no", "always"), ...)

## S4 method for signature 'RasterStackBrick'
freqDT(
  x,
  digits = 0,
  value = NULL,
  useNA = c("ifany", "no", "always"),
  merge = FALSE,
  ...
)
```

Arguments

x	A RasterLayer, RasterStack, or RasterBrick object field class.
...	Additional arguments as for raster::writeRaster(), on which this function relies.
digits	Integer for rounding the cell values. Argument is passed to round
value	A single numeric, logical, or NA value. If supplied, freqDT() will only count the number of cells with that value.
useNA	Character (one of "no", "ifany", or "always"). What to do with NA values? See table for details.
merge	Logical. If TRUE the list will be merged into a single data.table.

Author(s)

Joshua O'Brien

Examples

```
r <- raster(nrow = 18, ncol = 36)
r[] <- runif(ncell(r))
r[1:5] <- NA
r <- r * r * r * 5
freqDT(r)

freqDT(r, value = 2)

s <- stack(r, r*2, r*3)
freqDT(s, merge = TRUE)
```

subsDT *Speedy Raster Value Substitution*

Description

A fast `data.table`-based alternative to `raster::subs()`.

Usage

```
subsDT(x, dict, by = 1, which = 2, subsWithNA = TRUE, filename = "", ...)
```

Arguments

`x`	Categorical `RasterLayer` with integer values giving field class.
`dict`	A `data.frame` or `data.table` with one (or possibly more) columns corresponding to the values of cells in `x` and one (or possibly more) columns giving the value to which each value in `x` should be mapped.
`by`	Vector of one or possibly more integers or character strings giving the indices or names of the column in `dict` containing the categorical values in `x`.
`which`	Vector of one or possibly more integers or character strings giving the indices or names of the column in `dict` with the numerical values to which each value in `by` should be mapped.
`subsWithNA`	Logical. If `TRUE` values that are not matched become NA. If `FALSE`, they retain their original value (which could also be `NA`). This latter option is handy when you want to replace only one or a few values. It cannot be used when `x` has multiple layers
`filename`	Character string giving (optional) file name to which the resultant raster should be written.
`...`	Additional arguments as for `raster::writeRaster()`, on which this function relies.

Value

A `RasterLayer` object.

Author(s)

Joshua O'Brien

Examples

```
r <- raster(ncol = 10, nrow = 10)
r[] <- round(runif(ncell(r)) * 10)
df <- data.frame(id = 2:8, v = c(10, 10, 11, 11, 12:14))
x <- subsDT(r, df)
x2 <- subsDT(r, df, subsWithNA = FALSE)
```

```
df$v2 <- df$v * 10
x3 <- subsDT(r, df, which = 2:3)

s <- stack(r, r*3)
names(s) <- c("first", "second")
x4 <- subsDT(s, df)
x5 <- subsDT(s, df, which = 2:3)
```

`zonalDT` *Speedy Zonal Statistics*

Description

A fast `data.table`-based alternative to `raster::zonal()`.

Usage

```
zonalDT(x, z, fun = sum, na.rm = TRUE)
```

Arguments

`x`	A `Raster*` to the totality of whose values `fun` should be applied within each zone.
`z`	A categorical `RasterLayer` with codes representing zones.
`fun`	A name or character string giving the function to be applied to summarize the values by zone. It needs to return a single (or at least a length-one vector). If `x` might contain any `NA` values, it should be equipped to handle them. For large rasters, this function needs to be one, like `sum()` whose value is the same even if carried out in a two-stage application (i.e. first to data subsets and then to the results of those subset applications).
`na.rm`	Logical. If `TRUE`, `NA` values in `x` are ignored.

Value

A `data.table` with a summary value for each zone.

Author(s)

Joshua O'Brien

Examples

```
r <- raster(ncols = 10, nrows = 10)
r[] <- runif(ncell(r)) * 1:ncell(r)
z <- r
z[] <- rep(1:5, each = 20)
## for big files, use a character value rather than a function
zonalDT(r, z, "sum")

## for smaller files you can also provide a function
zonalDT(r, z, mean)
zonalDT(r, z, min)

## multiple layers
zonalDT(stack(r, r*10), z, "sum")
```

Index