

Package: luna (via r-universe)

September 1, 2024

Type Package

Title Tools for Satellite Remote Sensing (Earth Observation) Data Processing

Version 0.3-6

Date 2023-12-14

Encoding UTF-8

Language en-US

Depends R (>= 4.0.0), terra (>= 1.6.41)

LinkingTo Rcpp, RcppArmadillo

Imports methods, Rcpp (>= 1.0-10)

Suggests httr, rvest, xml2, jsonlite, meteor, signal

Maintainer Robert J. Hijmans <r.hijmans@gmail.com>

Description Tools for acquiring and (pre-) processing satellite remote sensing data. Including for downloading data from NASA such as LANDSAT and MODIS.

License GPL (>=3)

BugReports <https://github.com/rspatial/luna/issues/>

Repository <https://rspatial.r-universe.dev>

RemoteUrl <https://github.com/rspatial/luna>

RemoteRef HEAD

RemoteSha 2b56014071b1c30b5afa6cd3e642baa30dc41f79

Contents

luna-package	2
filterVI	2
getAVHRR	3
getCGLS	4
getEE	5
getERS	5

getLandsat	6
getModis	8
getNASA	8
getProducts	10
mesma	10
modisDate	12
modisExtent	12
modis_mask	13
panSharpen	14
productInfo	15

Index	16
--------------	-----------

luna-package	<i>Tools for Satellite Remote Sensing (Earth Observation) Data Processing</i>
--------------	---

Description

Tools for acquiring and (pre-) processing satellite remote sensing data. Including for downloading data from NASA such as LANDSAT and MODIS.

filterVI	<i>Gapfill and filter time series satellite data</i>
----------	--

Description

Estimate missing values using smooth gap-filled time series data with a two-stage Savitzky-Golay smoothing filter.

Usage

```
filterVI(x, ...)
```

```
fillVI(x, method = "natural")
```

Arguments

x	numeric or SpatRaster representing time series signal to be filtered
method	type of spline to be used. Possible values are "fmm", "natural", "periodic", "monoH.FC" and "hyman"
...	additional arguments: method: as above p1: filter order for first pass n1: filter length for first pass (must be odd) p2: filter order for inner passes n2: filter length for inner passes (must be odd)

Details

Performs smoothing of time series data using a two pass Savitzky-Golay smoothing filter. Second pass is an iterative method to minimize the difference between actual and fitted values that are significantly lower than the actual values. Missing values designated as NA are automatically interpolated with spline. For more details on the arguments see [splinefun](#) and [sgolayfilt](#).

Value

Filtered signal of length(x)

Examples

```
# apply on single pixel
#vi <- ....
#s <- filterVI(vi)

# for time series raster
#fevi <- app(evi, filterVI)
```

getAVHRR

Find and download AVHRR NDVI Climate Data Record (CDR)

Description

Find and download AVHRR NDVI CDR global layers for a time period. The function returned is list of filenames that may have been downloaded.

Usage

```
getAVHRR(start_date, end_date, path = "", overwrite=FALSE, update=FALSE, ...)
```

Arguments

start_date	character. Start date for the data requested formatted yyyy-m-d
end_date	Character. end date for the data requested formatted yyyy-m-d
overwrite	logical. If TRUE, existing files are overwritten
path	character. Path name indicating where to store the data
update	logical. If TRUE, existing records of AVHRR NDVI CDR will be updated
...	Additional arguments that can be passed to download.file)

Details

Downloads compressed global layers within the dates specified. For further processing, uncompressed ".nc4" files can be opened with the terra package.

Value

Character vector of file names pointing to the downloaded files

Examples

```
start <- "2019-05-17"
end <- "2019-05-23"
path <- tempdir()
# ff <- getAVHRR(start, end, path)
```

getCGLS

Download CGLS products

Description

Download data provided by the Copernicus Global Land Service (<https://land.copernicus.eu/global/>)
You need to register at <https://land.copernicus.eu/global/> and create a username and password.

See <https://land.copernicus.eu/global/products/> for a product overview and product details and see <https://land.copernicus.vgt.vito.be/manifest/> for an overview for data availability.

Usage

```
getCGLS(product, start_date, end_date, resolution="1km",
         version=1, path, username, password, ...)
```

Arguments

product	character. For example: "fapar", "fcover", "lai", "ndvi", "ssm", "swi", "lst"
resolution	character. one of "1km", "300m" or "100m"
version	positive integer. product version
start_date	character or Date. Start date for the data requested formatted yyyy-m-d
end_date	Character or Date. End date for the data requested formatted yyyy-m-d
path	character. Path name indicating where to store the data
username	character. EOSDIS user name (see Details)
password	character. EOSDIS password (see Details)
...	Additional arguments passed to download.file)

Value

SpatRaster

See Also

[getNASA](#)

getEE *download from Earth Explorer*

Description

Find and download Earth Explorer data

Usage

```
getEE(product, ids, path, username, password, ...)
```

Arguments

product	character. Supported products can be found using getProducts
ids	entity IDs
path	character. Path name indicating where to store the data
username	character. EOSDIS user name (see Details)
password	character. EOSDIS password (see Details)
...	Additional arguments. None implemented

Details

This function is an R interface to the Earth Explores system. Access is free, but it requires a "Earth-data" account. If you do not have an account, you can sign up here: <https://urs.earthdata.nasa.gov/users/new>

Value

Character vector of file names pointing to the downloaded files

See Also

[getLandsat](#)

getERS *Download data from ERS*

Description

Find and download ERS data

Usage

```
getERS(scenes, path, username, password, overwrite=FALSE, ...)
```

Arguments

scenes	character
path	character. Path name indicating where to store the data
username	character. User name
password	character. Password
overwrite	logical. If TRUE, existing files are overwritten
...	additional arguments. None implemented

Value

Character vector of file names pointing to the downloaded files

See Also

[getLandsat](#)

getLandsat	<i>Find and download Landsat products</i>
------------	---

Description

Find and download Landsat data for a specific product, area, and time period. The function returned is list of filenames that may have been downloaded.

Usage

```
getLandsat(product="Landsat_8_OLI_TIRS_C1", start_date, end_date, aoi,
           download=FALSE, path, username, password, version="1",
           limit=100000, server="AWS", overwrite=FALSE, ...)
```

Arguments

product	character. Supported products can be found using getProducts , currently limited to Landsat_8_OLI_TIRS_C1
start_date	character. Start date for the data requested formatted yyyy-m-d
end_date	character. end date for the data requested formatted yyyy-m-d
aoi	numeric vector of four elements (minimum longitude, maximum longitude, minimum latitude, maximum latitude) encompassing the area of interest. Or a SpatialExtent or Extent object, or any object from which an Extent can be extracted (see examples)
download	logical. If TRUE data will be downloaded unless it is present in the download directory
path	character. Path name indicating where to store the data
username	character. EROS user name (see Details)

password	character. EROS password (see Details)
version	character
limit	integer > 0
server	character. Either AWS (Landsat 8 only) or ERS. See Details
overwrite	logical. If TRUE, existing files are overwritten
...	Additional arguments. These can be product specific. See Details)

Details

The AWS (Amazon Web Services) server provides Landsat 8 data only. It does not require credentials (username and password).

The EROS service from the USGS also provides other Landsat products. To use for downloading data, you need to provide "ERS" credentials. If you do not have these, you can sign up for a free account here: <https://ers.cr.usgs.gov/register>

If no data is available between `start_date` and `end_date`, files for the closest dates are returned.

Value

character vector of file names pointing to the downloaded files (if `download=TRUE`)

See Also

[getProducts](#), [getNASA](#)

Examples

```
product <- "Landsat_8_OLI_TIRS_C1"
sdate <- "2019-05-01"
edate <- "2019-05-30"
area <- c(33, 34, -5, -4)
dir <- tempdir()
f <- getLandsat(product, start_date=sdate, end_date=edate,
               aoi=area, download=FALSE, path=dir)

# An example of ARD Tiles from ERS server
product <- "Landsat4-8_ARD_US_C1"
area <- c(-122.43, -121.50, 38.32, 38.91)
f1 <- getLandsat(product, start_date=sdate, end_date=edate,
                 aoi=area, download=FALSE, path=dir, server="EROS")
```

getModis	<i>Obsolete</i>
----------	-----------------

Description

Obsolete. See [getNASA](#)

Usage

```
getModis(...)
```

Arguments

... Arguments passed to [getNASA](#)

getNASA	<i>Find and download NASA products</i>
---------	--

Description

Find and download NASA data for a specific product, area, and time period. The function returned is list of filenames that may have been downloaded.

It is important to first consult [getProducts](#) to discover what products and version numbers are available, and which server to use.

Usage

```
getNASA(product, start_date, end_date, aoi, version="006", download=FALSE, path,
  username, password, server="LPDAAC_ECS", limit=100000, overwrite=FALSE, ...)
```

Arguments

product	character. Supported products can be found using getProducts
start_date	character. Start date for the data requested formatted yyyy-m-d
end_date	Character. end date for the data requested formatted yyyy-m-d
aoi	numeric vector of four elements (minimum longitude, maximum longitude, minimum latitude, maximum latitude) encompassing the area of interest. It can also be a <code>SpatExtent</code> , or any object from which an extent can be extracted
version	character. product version
download	logical. If TRUE data will be downloaded unless it is present in the download directory
path	character. Path name indicating where to store the data
username	character. EOSDIS user name (see Details)

password	character. EOSDIS password (see Details)
server	character. The server to download the data from
limit	positive integer
overwrite	logical. If TRUE, existing files are overwritten
...	Additional arguments. These can be product specific. See Details)

Details

This function is an R interface to the EOSDIS system. Access is free, but it requires a "Earthdata" account. If you do not have an account, you can sign up here: <https://urs.earthdata.nasa.gov/users/new>

If no data is available between `start_date` and `end_date`, files for the closest dates are returned.

Value

Character vector of file names pointing to the downloaded files (if `download=TRUE`)

See Also

[getProducts](#), [getLandsat](#)

Examples

```
product <- "MOD09A1"
# information about the product
getProducts(product)
sdate <- "2019-05-17"
edate <- "2019-05-23"
area <- c(33, 34, -5, -4)
path <- tempdir()
# list available tiles
f <- getNASA(product = product,
             start_date = sdate, end_date = edate,
             aoi = area, download = FALSE, path=path)
f

# get the date from the filenames
modisDate(f)

# to download the tiles, set download = TRUE
# and provide your credentials

usr <- "your user name"
pwd <- "your password"
#f <- getNASA(product = product, username=usr, password=pwd,
#             start_date = sdate, end_date = edate,
#             aoi = area, download = TRUE, path=path)
```

getProducts	<i>Find NASA product names</i>
-------------	--------------------------------

Description

Find available products in NASA's Common Metadata Repository (CMR) that can be downloaded. See [getNASA](#) for further use.

Usage

```
getProducts(product, ...)
```

Arguments

product	character
...	additional arguments. None implemented

Value

character

See Also

[getNASA](#)

Examples

```
# list of all available MODIS products
getProducts("^MOD|^MYD|^MCD")
```

mesma	<i>Multiple Endmember Spectral Mixture Analysis (Spectral Unmixing)</i>
-------	---

Description

mesma performs a multiple endmember spectral mixture analysis on a multiband raster image.

For unmixing, a non-negative least squares (NNLS) regression which is using a sequential coordinate-wise algorithm (SCA) based on Franc et al. (2005).

Usage

```
## S4 method for signature 'SpatRaster'
mesma(x, em, iterate=400, tolerance=0.0000001, ...)
```

Arguments

x	SpatRaster. Usually representing a hyperspectral remotely sensed image
em	matrix or data.frame with spectral endmembers. Rows represent a single end-member of a class, columns represent the spectral bands (i.e. columns correspond to number of bands in img). The number of rows needs to be > 1
iterate	integer. Set maximum iteration per pixel. Processing time could increase the more iterations are made possible
tolerance	numeric. Tolerance limit representing a nearly zero minimal number
...	further arguments passed to writeRaster

Value

SpatRaster with one layer per end-member, with each value representing the estimated presence probability of the end-member per pixel (0 to 1), and an RMSE layer.

Note

Depending on `iterate` and `tolerance` settings, the sum of estimated presence probabilities per pixel varies around 1.

Author(s)

Jakob Schwalb-Willmann

References

Franc, V., Hlaváč, V., & Navara, M. (2005). Sequential coordinate-wise algorithm for the non-negative least squares problem. In: International Conference on Computer Analysis of Images and Patterns (pp. 407-414). Berlin, Heidelberg.

Examples

```
lsat <- rast(system.file("ex/lsat.tif", package="luna"))

# endmember spectra: water and land
em_names <- c("water", "land")
pts <- data.frame(class=em_names, cell = c(47916,5294))
em <- lsat[pts$cell]
rownames(em) <- em_names

# unmix the image for water and land
probs <- mesma(lsat, em)

# take a look
hist(probs$water)
plot(probs$water, col = c("white", "blue"))
hist(probs$land)
plot(probs$land, col = c("white", "brown"))
```

modisDate	<i>Extract the date from a MODIS filename</i>
-----------	---

Description

Extract the date that is encoded in a MODIS filename

Usage

```
modisDate(filename)
```

Arguments

filename	character. MODIS product filenames
----------	------------------------------------

Value

data.frame

See Also

[getNASA](#)

Examples

```
modisDate("MOD09A1.A2019137.h21v09.006.2019150085736.hdf")
```

modisExtent	<i>Get the extent of a MODIS tile</i>
-------------	---------------------------------------

Description

Get the extent of a MODIS tile from the h (horizontal; column number) and v (vertical; row number) identifier in the filename

Usage

```
modisExtent(f=NULL, h, v)
```

Arguments

f	character. filename that includes a h12v09 pattern
h	integer between 0 and 35
v	integer between 0 and 17

Value

SpatExtent

See Also[modisDate](#)**Examples**

```
f <- "MOD09A1.A2019137.h21v09.006.2019150085736.hdf"
modisExtent(f)

modisExtent(h=21, v=9)
```

 modis_mask

MODIS mask

Description

Create a "mask" for a MODIS image based your selection of bad quality pixels, using the Quality Assessment (QA) band.

Usage

```
modis_mask(x, bits, qmat, ...)
```

Arguments

x	SpatRaster
bits	single integer value: 16 or 32
qmat	three-column matrix of data.frame. The first and second columns are positive integers (or character representation thereof) indicating the start and end bits (starting counting at one, not at zero!). The third column has the patterns to be rejected, stored as comma-separated characters. See example
...	additional arguments, none implemented

Value

SpatRaster

Examples

```
## Not run:
f <- "MOD09A1.A2011009.h21v08.006.2015216112834.hdf"
mod <- rast(f)
shortnames <- substr(names(mod), 86, 120)
names(mod) <- shortnames

qc <- mod[[12]]

from <- c(1,3,11,12) #,16)
to <- c(2,6,11,14) #,17)
reject <- c("10,11", "1100,1101,1110,1111", "1", "000,110,111") # "11")

qmat <- cbind(from, to, reject)

m <- modis_mask(qc, 16, qmat)
plot(m)

## End(Not run)
```

panSharpen

Panchromatic sharpening

Description

Panchromatic band sharpening of lower resolution satellite images.

Usage

```
## S4 method for signature 'SpatRaster,SpatRaster'
panSharpen(x, p, method="Brovey", weights=NULL, filename="", overwrite=FALSE, ...)
```

Arguments

x	SpatRaster with layers (bands) to be sharpened
p	SpatRaster with a single panchromatic layer at a higher spatial resolution than x
method	character. Either "Brovey" or "HSI"
weights	optional vector of weights for the "Brovey" method. The weights should have a length equal to nlyr(x) and they should normally add up to 1
filename	character. Output filename
overwrite	logical. If TRUE, existing files are overwritten
...	additional arguments for writing files as in writeRaster

Value

SpatRaster

productInfo	<i>NASA product information</i>
-------------	---------------------------------

Description

Get complete product information in the corresponding product website. The information is displayed in the default browser. See [getProducts](#) for available options of products.

Usage

```
productInfo(product, ...)
```

Arguments

product	character
...	additional arguments (non implemented)

See Also

[getProducts](#)

Examples

```
## Not run:  
# MODIS in LP DAAC with version information  
productInfo(product = "MCD18A1")  
# Without any version or server information  
productInfo(product = "LANDSAT_SURFACE_REFLECTANCE_L8_OLI_TIRS")  
  
## End(Not run)
```

Index

* **spatial**

modis_mask, [13](#)

fillVI (filterVI), [2](#)

filterVI, [2](#)

getAVHRR, [3](#)

getCGLS, [4](#)

getEE, [5](#)

getERS, [5](#)

getLandsat, [5](#), [6](#), [6](#), [9](#)

getModis, [8](#)

getNASA, [4](#), [7](#), [8](#), [8](#), [10](#), [12](#)

getProducts, [5-9](#), [10](#), [15](#)

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

luna-package, [2](#)

mesma, [10](#)

mesma, SpatRaster-method (mesma), [10](#)

modis_mask, [13](#)

modisDate, [12](#), [13](#)

modisExtent, [12](#)

panSharpen, [14](#)

panSharpen, SpatRaster, SpatRaster-method
(panSharpen), [14](#)

productInfo, [15](#)

sgolayfilt, [3](#)

splinefun, [3](#)

writeRaster, [11](#), [14](#)