Package: satres (via r-universe)

October 5, 2026

Title Grouping Satellite Bands by Spectral and Spatial Resolution

Version 1.1.1.9000

Description Given raster files directly downloaded from various websites, it generates a raster structure where it merges them if they are tiles of the same scene and classifies them according to their spectral and spatial resolution for easy access by name.

License MIT + file LICENSE

URL https://josesamos.github.io/satres/, https://github.com/josesamos/satres

BugReports https://github.com/josesamos/satres/issues

Depends R (>= 2.10)

Imports sf, snakecase, terra, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libicu-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://josesamos.r-universe.dev

Date/Publication 2024-10-16 04:18:13 UTC

RemoteUrl https://github.com/josesamos/satres

RemoteRef HEAD

RemoteSha 26c1aad08c8d9ff1a2505079f671e53c04889d48

Contents

as_SpatRaster . . . 2
clip_bands . . . 3
get_band_names . . . 4
get_spatial_resolution . . . 5
get_spectral_band_names . . . 5
merge_tiles . . . 6
sat_band . . . 7
sat_rest . . . 8
sat_rest_msk . . . 8
sat_untarzip . . . 9
satres . . . 10
save_by_resolution . . . 11
select_bands . . . 12

Index **13**

`as_SpatRaster` *As* `terra` `SpatRaster` *class*

Description

Returns the multi-band raster of the indicated spatial resolution as an object of class `SpatRaster` from package `terra`

Usage

```
as_SpatRaster(sr, res)

## S3 method for class 'satres'
as_SpatRaster(sr, res = NULL)
```

Arguments

`sr`	A `satres` object.
`res`	A string, spatial resolution.

Value

A vector of strings.

See Also

`sat_untarzip`

Other satellite exportation: `save_by_resolution()`

Examples

```
esa <- system.file("extdata", "esa", package = "satres")
sr <- satres(dir = esa)

r <- sr |>
     as_SpatRaster("r1000m")
```

`clip_bands` *Clip all the bands based on a polygon*

Description

Clips all bands of each spatial resolution according to the given polygon.

Usage

```
clip_bands(sr, polygon)

## S3 method for class 'satres'
clip_bands(sr, polygon)
```

Arguments

`sr`	A `satres` object.
`polygon`	A `sf` polygon layer.

Details

It performs the operation independently of the CRS of the polygon and preserves the CRS of the bands.

Value

A `satres` object.

See Also

`satres`

Other satellite transformation: `merge_tiles()`, `select_bands()`

Examples

```
file <- system.file("extdata", "lanjaron.gpkg", package = "satres")
lanjaron <- sf::st_read(file, layer = "lanjaron_bbox", quiet = TRUE)

esa <- system.file("extdata", "esa", package = "satres")
sr <- satres(dir = esa) |>
     clip_bands(polygon = lanjaron)
```

`get_band_names` *Get band names*

Description

Returns all names of the multi-band raster that make up the object.

Usage

```
get_band_names(sr, res)

## S3 method for class 'satres'
get_band_names(sr, res = NULL)
```

Arguments

`sr`	A `satres` object.
`res`	A string, spatial resolution.

Details

We can indicate the name of a certain spatial resolution to obtain only its names.

Value

A vector of strings.

See Also

`sat_untarzip`

Other satellite definition: `get_spatial_resolution()`, `get_spectral_band_names()`, `satres()`

Examples

```
esa <- system.file("extdata", "esa", package = "satres")
sr <- satres(dir = esa, only_spectral_bands = FALSE)
r <- sr |>
     get_band_names()
```

get_spatial_resolution *Get spatial resolutions*

Description

Returns the spatial resolutions of the multi-band raster that make up the object.

Usage

```
get_spatial_resolution(sr)

## S3 method for class 'satres'
get_spatial_resolution(sr)
```

Arguments

sr	A satres object.

Value

A vector of strings.

See Also

sat_untarzip

Other satellite definition: get_band_names(), get_spectral_band_names(), satres()

Examples

```
esa <- system.file("extdata", "esa", package = "satres")
sr <- satres(dir = esa)

r <- sr |>
     get_spatial_resolution()
```

get_spectral_band_names *Get band names*

Description

Returns the band names of the multi-band raster that make up the object.

Usage

```
get_spectral_band_names(sr, res)

## S3 method for class 'satres'
get_spectral_band_names(sr, res = NULL)
```

Arguments

sr	A satres object.
res	A string, spatial resolution.

Details

We can indicate the name of a certain spatial resolution to obtain only its band names.

Value

A vector of strings.

See Also

sat_untarzip

Other satellite definition: get_band_names(), get_spatial_resolution(), satres()

Examples

```
esa <- system.file("extdata", "esa", package = "satres")
sr <- satres(dir = esa, only_spectral_bands = FALSE)
r <- sr |>
     get_spectral_band_names()
```

---

merge_tiles *Merge objects that are tiles*

Description

Merge objects whose bands are tiles of a mosaic.

Usage

```
merge_tiles(sr, ...)

## S3 method for class 'satres'
merge_tiles(sr, ...)
```

Arguments

sr	A satres object.
...	satres objects.

Details

The objects must have the same CRS, spatial resolution and bands.

Value

A satres object.

See Also

satres

Other satellite transformation: clip_bands(), select_bands()

Examples

```
esa_f <- system.file("extdata", "esa/f", package = "satres")
esa_g <- system.file("extdata", "esa/g", package = "satres")
sr2 <- satres(dir = esa_f)
sr <- satres(dir = esa_g) |>
  merge_tiles(sr2)
```

---

sat_band *Final part of the name and extension of the satellite band files*

Description

The name of each element is the band identifier.

Usage

```
sat_band
```

Format

A vector.

See Also

Other satellite data: sat_rest_msk, sat_rest

sat_rest — *Final part of the name and extension of satellite rasters that are not bands*

Description

The name of each element is the raster identifier.

Usage

```
sat_rest
```

Format

A vector.

See Also

Other satellite data: sat_band, sat_rest_msk

sat_rest_msk — *Mask of name of satellite rasters that are not bands*

Description

Raster name patterns to treat and not consider bands.

Usage

```
sat_rest_msk
```

Format

A vector.

See Also

Other satellite data: sat_band, sat_rest

sat_untarzip *Unzip compressed files in tar or zip format*

Description

Given a vector of compressed file names or the name of a folder containing compressed files, unzip the files to the given output folder. If no output folder is indicated, it is considered the same folder where the input files are.

Usage

```
sat_untarzip(
  file,
  out_dir = NULL,
  include_filename = NULL,
  only_show_files = FALSE
)
```

Arguments

file	A string or string vector.
out_dir	A string or string vector, output folder.
include_filename	A boolean, include file name as a folder in the output.
only_show_files	A boolean, only show the files that would be unzipped, and the destination folders, not unzip them.

Details

We can indicate whether to include the file name (without the extension) as a folder in the output folder.

Value

A vector of strings, name of the processed files.

See Also

satres

Examples

```
f <- system.file("extdata", package = "satres")
r <- sat_untarzip(f, only_show_files = TRUE)

f1 <- system.file("extdata", "satres.zip", package = "satres")
```

```
f2 <- system.file("extdata", "satres.tar", package = "satres")
r <- sat_untarzip(c(f1, f2), only_show_files = TRUE)
```

---

satres satres *S3 class*

Description

Creates a satres object from a set of raster files.

Usage

```
satres(dir, out_dir = NULL, only_spectral_bands = TRUE)
```

Arguments

dir	A string or string vector, folder names.
out_dir	A string, output folder.
only_spectral_bands	A boolean, include only spectral bands.

Details

Given a folder name or a vector of folder names, containing satellite band raster files, creates an object containing all rasters grouped according to their spatial resolution.

If there are several rasters of the same area (tiles), it previously merges them to form a single raster of the total area.

A working folder where the virtual rasters are created can be indicated as a parameter. Additionally, we indicate whether we wish to process only the spectral band files (B1 to B12) or all available files.

Value

A satres object.

See Also

sat_untarzip

Other satellite definition: get_band_names(), get_spatial_resolution(), get_spectral_band_names()

Examples

```
esa <- system.file("extdata", "esa", package = "satres")

sr <- satres(dir = esa)

sr <- satres(dir = esa,
             out_dir = tempdir(),
             only_spectral_bands = FALSE)
```

`save_by_resolution` *Save multi-band rasters according to their spatial resolution*

Description

Saves multi-band raster files of the object according to its spatial resolution. The file names correspond to the resolution of each one.

Usage

```
save_by_resolution(sr, out_dir, only_show_files)

## S3 method for class 'satres'
save_by_resolution(sr, out_dir = NULL, only_show_files = FALSE)
```

Arguments

`sr`	A `satres` object.
`out_dir`	A string, output folder.
`only_show_files`	A boolean, only show the files that would be created, not create them.

Details

They are stored in the folder that is indicated or, if none is indicated, in the folder that was used to create the object.

Value

A vector of strings, name of the saved files.

See Also

`sat_untarzip`

Other satellite exportation: `as_SpatRaster()`

Examples

```
esa <- system.file("extdata", "esa", package = "satres")
sr <- satres(dir = esa)
f <- sr |>
     save_by_resolution(only_show_files = TRUE)
```

`select_bands` *Select bands by spatial resolution and name*

Description

Select the bands of an object based on spatial resolution and band name.

Usage

```
select_bands(sr, res, bands)

## S3 method for class 'satres'
select_bands(sr, res = NULL, bands = NULL)
```

Arguments

`sr`	A `satres` object.
`res`	A string, spatial resolution.
`bands`	A string, band name.

Value

A `satres` object.

See Also

`satres`

Other satellite transformation: `clip_bands()`, `merge_tiles()`

Examples

```
esa <- system.file("extdata", "esa", package = "satres")
sr <- satres(dir = esa) |>
     select_bands(res = c("r2000m", "r6000m"), bands = c("B02", "B03", "B04"))
```

Index

∗ **datasets**
    sat_band, 7
    sat_rest, 8
    sat_rest_msk, 8
∗ **satellite data**
    sat_band, 7
    sat_rest, 8
    sat_rest_msk, 8
∗ **satellite definition**
    get_band_names, 4
    get_spatial_resolution, 5
    get_spectral_band_names, 5
    satres, 10
∗ **satellite exportation**
    as_SpatRaster, 2
    save_by_resolution, 11
∗ **satellite previous functions**
    sat_untarzip, 9
∗ **satellite transformation**
    clip_bands, 3
    merge_tiles, 6
    select_bands, 12

as_SpatRaster, 2, *11*

clip_bands, 3, *7*, *12*

get_band_names, 4, *5, 6*, *10*
get_spatial_resolution, *4*, 5, *6*, *10*
get_spectral_band_names, *4*, 5, *5*, *10*

merge_tiles, *3*, 6, *12*

sat_band, 7, *8*
sat_rest, *7*, 8, *8*
sat_rest_msk, *7*, 8, *8*
sat_untarzip, *2*, *4–6*, 9, *10, 11*
satres, *3–7*, *9*, 10, *12*
save_by_resolution, *2*, 11
select_bands, *3*, *7*, 12