Package 'ASIP'

May 11, 2018

Type Package

Date 2018-05-11

Title Automated Satellite Image Processing

Version 0.4.9

Author M J Riyas [aut, cre],
T H Syed [aut]

Maintainer M J Riyas <riyasmjgeo@gmail.com>

Description

Efficiently perform complex satellite image processes automatically with minimum inputs.
Functions are providing more control on the user to specify how the function needs to be executed by offering more customization options and facilitate more functionalities.
The functions are designed to identify the type of input satellite images and perform accordingly.
Also, some functions are giving options to perform multiple satellite data (even from different types) in single run.
Package currently supports satellite images from most widely used Landsat 4,5,7 and 8 and Sentinel-2 MSI data.
The primary applications of this package are given below.
1. Conversion of optical bands to top of atmosphere reflectance.
2. Conversion of thermal bands to corresponding temperature images.
3. Derive application oriented products directly from source satellite image bands.
4. Compute user defined equation and produce corresponding image product.
5. Other basic tools for satellite image processing.
REFERENCES.
i. Chander and Markham (2003) <doi:10.1109/TGRS.2003.818464>.
ii. Roy et.al, (2014) <doi:10.1016/j.rse.2014.02.001>.
iii. Abrams (2000) <doi:10.1080/014311600210326>.

License GPL-3

Encoding UTF-8

Depends R (>= 3.4.0)

Imports raster (>= 2.6-7), utils, rgdal (>= 1.2-16), stringr (>= 1.2.0)

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no
Repository CRAN
Date/Publication 2018-05-11 14:56:27 UTC

R topics documented:

`arvi` *Atmospherically Resistant Vegetation Index*

Description

Atmospherically Resistant Vegetation Index (ARVI) is a vegetation based index which minimizes the effects of atmospheric scattering in comparison to NDVI.

Usage

```
arvi(directory = getwd(), crop = "n", ext2crop = "none", gamma = 1)
```

Arguments

`directory`	Path to Satellite image folder. Assign as string (inside double quotes). Either assing inside function or set up satellite image folder as the current working directory before running the function. To define current working directory, either use shortcut key Ctrl+Shift+H or use `setwd` funtion.
`crop`	Defines the method of cropping outputs to custom extent. "n" <- No cropping required (Default). "u" <- Satellite image will be plotted in the plot window and user can choose the extent by clicking on the top left maximum followed by bottom right maximum. "y" <- Crop to the maximum and minimum extent of the shapefile. "f" <- Crop to exact shapefile boundary.

ext2crop	Path to the shapefile (*.shp) which will be used for cropping. Shapefile should have SAME CORDINATE SYSTEM as the satellite image. Either provide the full path of .shp file or provide the name of the shapefile variable which is already opened.
gamma	It is an aerosol dependant factor. For more details please refer Kaufman and Tanre (1992). By default the value is 1.

Value

Computed ARVI product

Note

1. ARVI = (r_nir - rb)/(r_nir + rb), where

rb = r_red - gamma (r_blue - r_red)' and "r_" denotes Top Of Atmoshpere (TOA) reflection, 'gamma' value is 1 by default as recommended if information about the aerosol type is not available. Please refer Kaufman and Tanre (1992) for more details.

Other important notes are mentioned in `custom.eqn`.

References

Kaufman, Y. J. and D. Tanre (1992) Atmospherically resistant vegetation index (ARVI) for EOS-MODIS, IEEE Transactions on Geoscience and Remote Sensing, 30 (2). doi:10.1109/36.134076.

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
shapefil <- paste0 (path, "/test.shp")
op <- arvi (directory = path, crop = "y", ext2crop = shapefil)
```

---

crop.bands *Cropping of satellite image bands*

Description

Crop desired satellite image bands either using a shapefile or draw custom extent from a plot image while running the function.

Usage

```
crop.bands(directory = getwd(), crop = "n", ext2crop = "none",
  op_directory = directory, b1 = 1, b2 = 1, b3 = 1, b4 = 1, b5 = 1,
  b6 = 1, b7 = 1)
```

Arguments

ext2crop, crop, directory	Same as mentioned in arvi.
op_directory	Specify the output directory (within double quotes). By default the input satellite image directory will be selected as the output directory.
b1	By default Band1 will be cropped. To cancel cropping of this band assign value 0.
b2	By default Band2 will be cropped. To cancel cropping of this band assign value 0.
b3	By default Band3 will be cropped. To cancel cropping of this band assign value 0.
b4	By default Band4 will be cropped. To cancel cropping of this band assign value 0.
b5	By default Band5 will be cropped. To cancel cropping of this band assign value 0.
b6	By default Band6 will be cropped. To cancel cropping of this band assign value 0.
b7	By default Band7 will be cropped. To cancel cropping of this band assign value 0.

Value

Each bands selected will cropped and produce corresponding <bandname>_crop.tif format in the input directory.

Note

1. FILENAMES OF ANY BAND FILES (*.TIF files) SHOULDN'T CHANGED.
2. Windows users should be careful while assigning directory. Use "/" to seperate folders not "\".

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
shapefil <- paste0 (path, "/test.shp")
# Assign 0 values to band names which are not required
crop.bands (path, crop = "f", ext2crop = shapefil, b3=0, b4=0, b5=0, b6 = 0, b7 = 0)
```

custom.eqn *Make your own custom satellite image product*

Description

If any product or index is not available in this package, you don't need to do it manually. This function intakes a custom formula & produced new product according to the formula. This function converts DN bands to corresponding TOA reflectance prior to the computation of user defined formula.

Usage

```
custom.eqn(directory = getwd(), cus.formula = "none", crop = "n",
  ext2crop = "none")
```

Arguments

`cus.formula`	Assign custom formula to be computed AS TEXT input (inside double quotes). To assign bands, ONLY USE BELOW DEFINED WORDS to indicate different bands in the formula. nir for NIR (Near Infra-red) Top Of Atmosphere (TOA) reflectance band. red for Red TOA reflectance band. green for Green TOA reflectance band. blue for Blue TOA reflectance band. swir1 for SWIR-1 (Short Wave Infra-red -1) swir2 for SWIR-2 (Short Wave Infra-red -2) aero for Aerosol/coastal band (Only on Landsat OLI images)
`ext2crop, crop, directory`	Same as mentioned in `arvi`.

Value

Computed custom equation based product.

Note

1. FILENAMES OF ANY BAND FILES (*.TIF files) SHOULDN'T CHANGED.

2. Windows users should be careful while assigning directory. Use "/" to seperate folders not "\".

3. Earth-sun distance is calculated according to Epema (1992) if the value is not mentioned in the meta data (*MTL.txt) file.

4. Currently recommended ESUN values provided by USGS is used.

References

Epema G F (1992) Atmospheric condition and its influence on reflectance of bare soil surfaces in southern Tunisia. International Journal of Remote Sensing, 13(5), pp:853-868. doi:10.1080/01431169208904159.

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
# Input equation should be as text (inside double quotes)
eqn <- "(2 * red) + (nir/blue)"
shapefil <- paste0 (path, "/test.shp")
op <- custom.eqn (directory = path, cus.formula = eqn, crop = "y", ext2crop = shapefil)
```

`dn2toa` *DN to TOA conversion of optical bands*

Description

Digital number (DN) bands to Top of Atmosphere (TOA) conversion.

Usage

```
dn2toa(directory = getwd(), crop = "n", ext2crop = "none",
  op_directory = directory, b1 = 1, b2 = 1, b3 = 1, b4 = 1, b5 = 1,
  b6 = 1, b7 = 1)
```

Arguments

`ext2crop, crop, directory`	Same as mentioned in `arvi`.
`op_directory`	Specify the output directory (within double quotes). By default the input satellite image directory will be selected as the output directory.
`b1`	By default Band1 will be processed to TOA reflectance. To cancel production of this band assign value 0.
`b2`	By default Band2 will be processed to TOA reflectance. To cancel production of this band assign value 0.
`b3`	By default Band3 will be processed to TOA reflectance. To cancel production of this band assign value 0.
`b4`	By default Band4 will be processed to TOA reflectance. To cancel production of this band assign value 0.
`b5`	By default Band5 will be processed to TOA reflectance. To cancel production of this band assign value 0.
`b6`	By default Band6 will be processed to TOA reflectance. To cancel production of this band assign value 0.
`b7`	By default Band7 will be processed to TOA reflectance. To cancel production of this band assign value 0.

Value

Each bands selected will produce corresponding image in *.tif format in the input directory.

Note

1. This function followed by `multi.indices` is recommended only if user is intended to produce multiple indices like ndvi & gemi other than running seperate function for each product to save processing time and resources.

Other important notes are mentioned in `custom.eqn`.

References

USGS (2016) Landsat 8 (L8) data users handbook, version 2.

Landsat 7 science data users handbook, NASA. Available at "https://landsat.gsfc.nasa.gov/wp-content/uploads/2016/08/Lands

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
shapefil <- paste0 (path, "/test.shp")
# Assign 0 values to band names which are not required
dn2toa (path, crop = "f", ext2crop = shapefil, b3=0, b4=0, b5=0, b6 = 0, b7 = 0)
```

---

gemi *Global Environmental Monitoring Index*

Description

Global Environmental Monitoring Index (GEMI) is a vegetation based index designed to reduce the effects of atmosperic effects.

Usage

```
gemi(directory = getwd(), crop = "n", ext2crop = "none")
```

Arguments

`ext2crop, crop, directory`
: Same as mentioned in `arvi`.

Value

Computed GEMI product

Note

1. GEMI = n (1 - 0.25n) - ((r_red - 0.125)/(1 - r_red))

where, n= (2 (r_nir^2 - r_red^2) + 1.5 r_nir + 0.5 r_red)/ (r_nir + r_red + 0.5) and "r_" denotes TOA reflectance band.

Other important notes are mentioned in `custom.eqn`.

References

Pinty, B. and M. M. Verstraete (1992) GEMI: a non-linear index to monitor global vegetation from satellites, Vegetatio, 101 (1), 15-20.

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
shapefil <- paste0 (path, "/test.shp")
op <- gemi (directory = path, crop = "y", ext2crop = shapefil)
```

`gvmi` *Global Vegetation Moisture Index*

Description

Global Vegetation Moisture Index (GVMI) is suitable for retrieving vegetation water content when the LAI is equal to or greater than 2. For sparsely vegetated area, where LAI is less than 2, further research is required to understand the role of soil effects on reflectance measured in the all-optical spectrum (Ceccato et, al. 2002).

Usage

```
gvmi(directory = getwd(), crop = "n", ext2crop = "none")
```

Arguments

`ext2crop, crop, directory`
: Same as mentioned in `arvi`.

Value

Computed GVMI product

Note

1. GVMI = ((r_nir + 0.1)- (r_swir2+0.2))/ ((r_nir + 0.1)- (r_swir2+0.2))

where, "r_" denotes TOA reflectance band. Other important notes are mentioned in `custom.eqn`.

References

Ceccato P, Gobron N, Flasse S, Pinty B and Tarantola S (2002) Designing a spectral index to estimate vegetation water content from remote sensing data: Part 1: Theoretical approach. Remote Sensing of Environment, 82(2-3), pp:188-197. doi:10.1016/S0034-4257(02)00037-8.

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
shapefil <- paste0 (path, "/test.shp")
op <- gvmi (directory = path, crop = "y", ext2crop = shapefil)
```

`msavi` *Modified soil adjusted vegetation index*

Description

Modified Soil Adjusted Vegetation Index (MSAVI) is a vegetation index. Advantage of this index is that, it increases the dynamic range of the vegetation signal while further minimizing the soil background influences, resulting in greater vegetation sensitivity as defined by a 'vegetation signal' to 'soil noise' ratio.

Usage

```
msavi(directory = getwd(), crop = "n", ext2crop = "none")
```

Arguments

`ext2crop, crop, directory`
: Same as mentioned in `arvi`.

Value

Computed MSAVI product

Note

1. MSAVI=((2r_nir + 1) - ((2r_nir + 1)^2 - 8(r_nir - r_red))^0.5)/2

where, "r_" denotes TOA reflectance band.

Other important notes are mentioned in `custom.eqn`.

References

Qi J, Chehbouni A, Huete A R, Kerr Y, Sorooshian S (1994) A modified soil adjusted vegetation index. Remote Sensing of Environment, 48 (2), pp: 119-126. doi:10.1016/0034-4257(94)90134-1.

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
shapefil <- paste0 (path, "/test.shp")
op <- msavi (directory = path, crop = "y", ext2crop = shapefil)
```

`multi.indices` *Produce multiple indices*

Description

This function is designed to produce multiple indices directly. The source satellite image bands will be converted Top of Atmosphere (TOA) reflectance prior to do the indices production.

Usage

```
multi.indices(directory = getwd(), crop = "n", ext2crop = "none",
  op_directory = directory, arvi = 0, gamma = 1, gemi = 0, gvmi = 0,
  msavi = 0, ndbi = 0, ndvi = 1, ndwi = 0, pavi = 0, all = 0)
```

Arguments

`ext2crop, crop, directory`	Same as mentioned in `arvi`.
`op_directory`	Specify the output directory <within double quotes>. By default the input satellite image directory will be selected as the output directory.
`arvi`	This product won't be produced by default. To produce this product, assign value 1. This product is same as that of the output from `arvi`. Only difference is, this function intakes already produced TOA bands instead of creating new TOA bands from source satellite image bands.
`gamma`	Defined in `arvi`.
`gemi`	This product won't be produced by default. To produce this product, assign value 1. This product is same as that of the output from `gemi`. Only difference is, this function intakes already produced TOA bands instead of creating new TOA bands from source satellite image bands.
`gvmi`	This product won't be produced by default. To produce this product, assign value 1. This product is same as that of the output from `gvmi`. Only difference is, this function intakes already produced TOA bands instead of creating new TOA bands from source satellite image bands.
`msavi`	This product won't be produced by default. To produce this product, assign value 1. This product is same as that of the output from `msavi`. Only difference is, this function intakes already produced TOA bands instead of creating new TOA bands from source satellite image bands.

ndbi	This product won't be produced by default. To produce this product, assign value 1. This product is same as that of the output from ndbi. Only difference is, this function intakes already produced TOA bands instead of creating new TOA bands from source satellite image bands.
ndvi	This product will be produced by default. To disable this product, assign value 0. This product is same as that of the output from ndvi. Only difference is, this function intakes already produced TOA bands instead of creating new TOA bands from source satellite image bands.
ndwi	This product won't be produced by default. To produce this product, assign value 1. This product is same as that of the output from ndwi. Only difference is, this function intakes already produced TOA bands instead of creating new TOA bands from source satellite image bands.
pavi	This product won't be produced by default. To produce this product, assign value 1. This product is same as that of the output from pavi. Only difference is, this function intakes already produced TOA bands instead of creating new TOA bands from source satellite image bands.
all	If you wish to produce all the indices, assign value 1.

Value

Each bands selected will produce corresponding image in *.tif format in the input directory.

Note

1. This function followed by multi.indices is recommended only if user is intended to produce multiple indices like ndvi & gemi other than running seperate function for each product to save processing time and resources.

Other important notes are mentioned in custom.eqn.

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
shapefil <- paste0 (path, "/test.shp")
multi.indices (path, crop = "f", ext2crop = shapefil, msavi =1, ndvi = 0)
```

---

ndbi *Normalized Difference Built-up Index*

Description

Normalized Difference Built-up Index (NDBI) is used for mapping built-up areas

Usage

```
ndbi(directory = getwd(), crop = "n", ext2crop = "none")
```

Arguments

`ext2crop, crop, directory`
: Same as mentioned in `arvi`.

Value

Computed NDBI product

Note

1. NDBI= (r_swir1 - r_nir)/ (r_swir1 + r_nir)

where, "r_" denotes TOA reflectance band.

Other important notes are mentioned in `custom.eqn`.

References

Zha Y, Gao J and Ni S (2003) Use of normalized difference built-up index in automatically mapping urban areas from TM imagery. International Journal of Remote Sensing, 24 (3), pp:583-594. doi:10.1080/01431160304987.

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
shapefil <- paste0 (path, "/test.shp")
op <- ndbi (directory = path, crop = "y", ext2crop = shapefil)
```

---

ndvi *Normalized Difference Vegetation Index*

Description

Normalized Difference Vegetation Index (NDVI). It is the most widely used satellite image derived index emphasizing on vegetation mapping. This function can also be used to obtain the ground emissivity as well.

Usage

```
ndvi(directory = getwd(), crop = "n", ext2crop = "none",
  op_directory = directory, emissivity = "n")
```

Arguments

ext2crop, crop, directory	Same as mentioned in arvi.
op_directory	Specify the output directory (within double quotes). By default the input satellite image directory will be selected as the output directory.
emissivity	Assign value "y" to get the emissivity of the surface as a seperate raster file.

Value

File named ndvi_'date of satellite image acqisition'.tif in the input folder

Note

1. NDVI= (r_nir - r_red)/(r_nir + r_red)

where, "r_" denotes TOA reflectance band.

2. Emissivity is calculated according to Van De Griend and Owe (1993).

3. Emissivity values will be absent on pixels with negative NDVI values. This can affect the land surface temperature results (not at-sensor brightness temperature) as well. So, please review the study region and the requirements before using this function.

4. Other important notes are mentioned in custom.eqn.

References

1. Huetet A R and Jackson R D (1987) Suitability of spectral indices for evaluating vegetation characteristics on arid rangelands, Remote sensing of environment, 23(2), pp:213-232. doi: 10.1016/0034-4257(87)90038-1.

2. Van De Griend AA, Owe M (1993) On the relationship between thermal emissivity and the normalized difference vegetation index for natural surfaces. Int J Remote Sens 14:1119–1131. doi: 10.1080/01431169308904400

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
shapefil <- paste0 (path, "/test.shp")
ndvi (directory = path, crop = "y", ext2crop = shapefil)
```

ndwi *Normalized Difference Water Index*

Description

Normalized Difference Water Index (NDWI) is used to monitor changes related to water content in water bodies proposed by McFeeters (1996).

Usage

```
ndwi(directory = getwd(), crop = "n", ext2crop = "none")
```

Arguments

`ext2crop, crop, directory`
: Same as mentioned in `arvi`.

Value

Computed NDWI product

Note

1. NDWI = (r_green - r_nir) / (r_nir + r_green)

where, "r_" denotes TOA reflectance band.

2. There is another NDWI to monitor changes in water content of leaves proposed by Gao (1996). User should understand the requirements and run accordingly.

Other important notes are mentioned in `custom.eqn`.

References

McFeeters, S.K. (1996) The Use of the Normalized Difference Water Index (NDWI) in the Delineation of Open Water Features. International Journal of Remote Sensing, 17, 1425-1432. doi:10.1080/01431169608948714.

Gao Bo-cai (1996) NDWI-A normalized difference water index for remote sensing of vegetation liquid water from space. Remote Sensing of Environment, 58 (3), 257-266. doi:10.1016/S0034-4257(96)00067-3.

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
shapefil <- paste0 (path, "/test.shp")
op <- ndwi (directory = path, crop = "y", ext2crop = shapefil)
```

pavi *Purified Adjusted Vegetation Index*

Description

Purified Adjusted Vegetation Index (PAVI) is a vegetation based index.

Usage

```
pavi(directory = getwd(), crop = "n", ext2crop = "none")
```

Arguments

`ext2crop, crop, directory`
: Same as mentioned in `arvi`.

Value

Computed PAVI product

Note

1. PAVI= (r_nir^2 - r_red^2)/(r_nir^2 + r_red^2)

where, "r_" denotes TOA reflectance band.

Other important notes are mentioned in `custom.eqn`.

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
shapefil <- paste0 (path, "/test.shp")
op <- pavi (directory = path, crop = "y", ext2crop = shapefil)
```

sen2_msi *Make your own custom Sentinel-2 MSI satellite image products*

Description

This function is dedicated to Sentinel-2 MSI satellite image processing. Provide your custom equation to produce the desired results (Tested only Sentinel 2 L1C products).

Usage

```
sen2_msi(directory = getwd(), cus.formula = "none", crop = "n",
  ext2crop = "none")
```

Arguments

cus.formula	Assign custom formula to be computed AS TEXT input (inside double quotes). To assign bands, ONLY USE BAND NUMBERS (b1, b2,....,b12) to indicate different bands in the formula.
ext2crop, crop, directory	Same as mentioned in arvi.

Value

Computed Sentinel 2 custom product

Note

1. FILENAMES OF ANY BAND FILES (*.jp2 files) SHOULDN'T CHANGED.
2. Bands with same resolution can only be computed.
2. Windows users should be careful while assigning directory. Use "/" to seperate folders not "\".

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
##path <- system.file ("TM_sample", package = "ASIP")
# Input equation should be as text (inside double quotes)
eqn <- "((2 * b4)+ (b3+pi+b8))/(b3+b4+b8)"
##shapefil <- paste0 (path, "/test.shp")
##op <- custom.eqn (directory = path, cus.formula = eqn, crop = "y", ext2crop = shapefil)
```

---

thermal *TIR bands to at satellite brightness temperature conversion*

Description

Identifies Thermal Infra-Red (TIR) bands and converts them to at satellite brightness temperature images.

Usage

```
thermal(directory = getwd(), crop = "n", ext2crop = "none",
  op_directory = directory, unit = "Deg Kel")
```

Arguments

ext2crop, crop, directory	Same as mentioned in arvi.
op_directory	Specify the output directory (within double quotes). By default the input satellite image directory will be selected as the output directory.
unit	By default the temperature image will be produced in Degree Kelvin. To produce the thermal image in Degree celcius, assign vale "c". To produce the thermal image in Degree celcius, assign vale "c".

Value

At Satellite Brightness Temperature images in .tif format in input directory.

Note

1. FILENAMES OF ANY BAND FILES (*.TIF files) SHOULDN'T CHANGED.
2. Windows users should be careful while assigning directory. Use "/" to seperate folders.
3. Emissivity value used is 1.

Examples

```
library (raster)
library (rgdal)
# Finding the path of the sample satellite image directory.
# User may define paths directly like "/home/ur_folder" or "C:/ur_folder"
path <- system.file ("TM_sample", package = "ASIP")
shapefil <- paste0 (path, "/test.shp")
thermal (directory = path, crop = "y", ext2crop = shapefil, unit = "c")
```

Index