

# Package ‘proccalibrad’

October 14, 2022

**Type** Package

**Title** Extraction of Bands from MODIS Calibrated Radiances MOD02 NRT

**Version** 0.14

**Author** Rishabh Gupta <rishabh.uk@gmail.com>, Nicholas J. Matzke

**Maintainer** Rishabh Gupta <rishabh.uk@gmail.com>

**Description** Package for processing downloaded MODIS Calibrated radiances Product HDF files. Specifically, MOD02 calibrated radiance product files, and the associated MOD03 geolocation files (for MODIS-TERRA). The package will be most effective if the user installs MRTSwath (MODIS Reprojection Tool for swath products; <[https://lpdaac.usgs.gov/tools/modis\\_reprojection\\_tool\\_swath](https://lpdaac.usgs.gov/tools/modis_reprojection_tool_swath)>, and adds the directory with the MRTSwath executable to the default R PATH by editing ~/.Rprofile.

**SystemRequirements** MRTSwath

**Imports** utils

**License** GPL (>= 2)

**LazyData** TRUE

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-07-31 08:54:00

## R topics documented:

adf . . . . .	2
check_for_matching_geolocation_files_mod02nrt . . . . .	2
extract_fn_from_path . . . . .	4
run_swath2grid_mod02nrt . . . . .	4
slasheslash . . . . .	6
write_MRTSwath_param_file_mod02nrt . . . . .	7

<b>Index</b>	<b>10</b>
--------------	-----------

adf *Convert to data.frame, without factors*

---

### Description

Shortcut for: `as.data.frame(x, row.names=NULL, stringsAsFactors=FALSE)`

### Usage

```
adf(x)
```

### Arguments

x                   matrix or other object transformable to data.frame

### Details

This function is useful for dealing with errors due to automatic conversion of some columns to factors. Another solution may be to prepend `options(stringsAsFactors = FALSE)` at the start of one's script, to turn off all default stringsAsFactors silliness.

### Value

data.frame

### Examples

```
x = matrix(c(1,2,3,4,5,6), nrow=3, ncol=2)
adf(x)
```

---

check\_for\_matching\_geolocation\_files\_mod02nrt

*Checks that every MODIS calibrated radiance project HDF has a matching MOD03 file*

---

### Description

Each MOD02 calibrated radiance product file requires a corresponding MOD03 geolocation file to be successfully processed with the MRTSwath tool.

### Usage

```
check_for_matching_geolocation_files_mod02nrt(moddir = getwd(),
  modtxt = "MOD02", geolotxt = "MOD03", return_geoloc = FALSE,
  return_product = FALSE)
```

**Arguments**

moddir	the string describing the directory containing the MOD02 and MOD03 files; both must be in the same directory. Default: getwd(), which gives the present working directory.
modtxt	the text string indicating which HDF files are the MODIS calibrated radiance product (or hypothetically, other product). Default: MOD02 (MODIS calibrated radiance product)
geolotxt	the text string indicating which HDF files are the MODIS geolocation files (or hypothetically, another set of files). Default: MOD03
return_geoloc	if TRUE, return the list of unmatched geolocation files (e.g. MOD03 )
return_product	if TRUE, return the list of unmatched product files (e.g. MOD02)

**Details**

MRTSwath is the MRT (MODIS Reprojection Tool) for the MODIS

E.g. this calibrated radiance file:

MOD021KM.A2016209.0515.005.NRT.hdf

...goes with this corresponding geolocation file:

MOD03.A2016209.0515.005.NRT.hdf

...which is a large file (~30 MB) containing detailed information on the position, tilt, etc. of the MODIS satellite. MRTSwath tool needs one of each, however.

**Value**

data.frame of matching files; or a list of non-matching files, if return\_geoloc or return\_product are TRUE.

**Author(s)**

Rishabh Gupta <rishabh.uk@gmail.com>

**Examples**

```
# Check your working directory
moddir = getwd()

# Here are some example MODIS files in mod02nrt/extdata/
# Code excluded from CRAN check because it depends on modiscdata
## Not run:
library(devtools)
library(modiscdata)
moddir = system.file("extdata/2002raw/", package="modiscdata")

# You need to have some e.g. MOD files in it (from the MODIS-TERRA platform)
list.files(path=moddir, pattern="MOD")
list.files(path=moddir, pattern="MOD")
```

```
# Check for matches (for MODIS-TERRA platform)
check_for_matching_geolocation_files_mod02nrt(moddir=moddir, modtxt="MOD02", geolotxt="MOD03",
  return_geoloc=FALSE, return_product=FALSE)

## End(Not run)
```

---

extract\_fn\_from\_path *Get the filename from a path*

---

### Description

The filename is split on slashes, and the last item is taken; this should be just the filename.

### Usage

```
extract_fn_from_path(fn_with_path)
```

### Arguments

fn\_with\_path The filename, with partial or full path

### Value

fn The extracted filename

### Examples

```
fn_with_path = "/Library/Frameworks/R.framework/Versions/2.15/Resources/library/
MOD021KM.A2016209.0515.005.NRT.hdf"
extract_fn_from_path(fn_with_path)
```

---

run\_swath2grid\_mod02nrt  
*Run MRTSwath swath2grid tool*

---

### Description

MRTSwath is the "MODIS Reprojection Tool for swath products". See: [https://lpdaac.usgs.gov/tools/modis\\_reprojection\\_tool\\_swath](https://lpdaac.usgs.gov/tools/modis_reprojection_tool_swath)).

### Usage

```
run_swath2grid_mod02nrt(mrtpath = "swath2grid", prmfnc = "tmpMRTparams.prm",
  tifmdir, modfn, geoloc_fn, ul_lon, ul_lat, lr_lon, lr_lat)
```

**Arguments**

mrtpath	This is the path to the MRTSwath executable swath2grid. If your ~/.Rprofile file has the location of swath2grid in the PATH, then you can just use mrtpath="swath2grid". Otherwise, the user must provide the full path to swath2grid.
prmfnc	The name of the parameter/control file which will be the input to MRTSwath's swath2grid function.
tifsdire	The directory to save the output TIF files in
modfn	The filename of the MODIS data
geoloc_fn	The filename of the corresponding geolocation file (annoyingly, this is a much larger file than the data file!)
ul_lon	Upper left (ul) longitude (x-coordinate) for subsetting
ul_lat	Upper left (ul) latitude (y-coordinate) for subsetting
lr_lon	Lower right (lr) longitude (x-coordinate) for subsetting
lr_lat	Lower right (lr) latitude (y-coordinate) for subsetting

**Details**

If you want this function to use MRTSwath tool successfully, you should add the directory with the MRTSwath executable to the default R PATH by editing ~/.Rprofile.

**Value**

cmdstr The string giving the system command that ran swath2grid

**See Also**

[write\\_MRTSwath\\_param\\_file\\_mod02nrt](#)

[http://landweb.nascom.nasa.gov/cgi-bin/QA\\_WWW/newPage.cgi?fileName=hdf\\_filename@cite NASA2001](http://landweb.nascom.nasa.gov/cgi-bin/QA_WWW/newPage.cgi?fileName=hdf_filename@cite NASA2001)

**Examples**

```
#####
# Run MRTSwath tool "swath2grid"
#####

# Source MODIS files (both data and geolocation)
# Code excluded from CRAN check because it depends on modiscdata
## Not run:
library(devtools)
library(modiscdata)
moddir = system.file("extdata/2002raw/", package="modiscdata")

# Get the matching data/geolocation file pairs
fns_df = check_for_matching_geolocation_files(moddir, modtxt="MOD02", geoltxt="MOD03")
fns_df
```

```

# Resulting TIF files go in this directory
tifmdir = getwd()

# Box to subset
ul_lat = 13
ul_lon = -87
lr_lat = 8
lr_lon = -82

for (i in 1:nrow(fns_df))
{

prmfnc = write_MRTPSwath_param_file_mod02nrt(prmfnc="tmpMRTPparams.prm", tifmdir=tifmdir,
  modfn=fns_df$mod02_fns[i], geoloc_fn=fns_df$mod03_fns[i], ul_lon=ul_lon, ul_lat=ul_lat,
  lr_lon=lr_lon, lr_lat=lr_lat)
print(scan(file=prmfnc, what="character", sep="\n"))

run_swath2grid_mod02nrt(mrtpath="swath2grid", prmfnc="tmpMRTPparams.prm", tifmdir=tifmdir,
  modfn=fns_df$mod03_fns[i], geoloc_fn=fns_df$mod03_fns[i], ul_lon=ul_lon, ul_lat=ul_lat,
  lr_lon=lr_lon, lr_lat=lr_lat)

}

list.files(tifmdir, pattern=".tif", full.names=TRUE)

## End(Not run)

```

---

slasheslash

*Remove double slash (slash a slash)*


---

### Description

Shortcut for: `gsub(pattern="//", replacement="/", x=tmpstr)`

### Usage

```
slasheslash(tmpstr)
```

### Arguments

`tmpstr` a path that you want to remove double slashes from

### Details

This function is useful for removing double slashes that can appear in full pathnames due to inconsistencies in trailing slashes in working directories etc.

**Value**

outstr a string of the fixed path

**Examples**

```
tmpstr = "/Library/Frameworks/R.framework/Versions/2.15/Resources/library/
MOD03.A2016209.0515.005.NRT.hdf"
```

```
outstr = slasheslash(tmpstr)
outstr
```

---

```
write_MRTSwath_param_file_mod02nrt
```

*Write a parameter control file for MRTSwath*

---

**Description**

MRTSwath is the "MODIS Reprojection Tool for swath products". See: [https://lpdaac.usgs.gov/tools/modis\\_reprojection\\_tool\\_swath](https://lpdaac.usgs.gov/tools/modis_reprojection_tool_swath)).

**Usage**

```
write_MRTSwath_param_file_mod02nrt(prmfn = "tmpMRTparams.prm", tifmdir, modfn,
  geoloc_fn, ul_lon, ul_lat, lr_lon, lr_lat)
```

**Arguments**

prmfn	The name of the parameter/control file which will be the input to MRTSwath's swath2grid function.
tifmdir	The directory to save the output TIF files in
modfn	The filename of the MODIS data
geoloc_fn	The filename of the corresponding geolocation file (annoyingly, this is a much larger file than the data file!)
ul_lon	Upper left (ul) longitude (x-coordinate) for subsetting
ul_lat	Upper left (ul) latitude (y-coordinate) for subsetting
lr_lon	Lower right (lr) longitude (x-coordinate) for subsetting
lr_lat	Lower right (lr) latitude (y-coordinate) for subsetting

## Details

If you want this function to use MRTSwath tool successfully, you should add the directory with the MRTSwath executable to the default R PATH by editing `~/.Rprofile`.

This function hard-codes these options into the parameter file:

- \* all the bands are extracted
- \* the output file is a GeoTIFF
- \* the output projection is Geographic (plain unprojected Latitude/Longitude)
- \* the resampling is Nearest Neighbor (NN), which of course is the only one which makes sense when the pixels encode bytes that encode bits that encode discrete classification results, 0/1 error flags, etc.

MRTSwath can do many other projections and output formats; users can modify this function to run those options.

## Value

`prmf` The name of the temporary parameter file

## Author(s)

Rishabh Gupta <[rishabh.uk@gmail.com](mailto:rishabh.uk@gmail.com)>

## See Also

[run\\_swath2grid\\_mod02nrt](#)

[http://landweb.nascom.nasa.gov/cgi-bin/QA\\_WWW/newPage.cgi?fileName=hdf\\_filename@cite NASA2001](http://landweb.nascom.nasa.gov/cgi-bin/QA_WWW/newPage.cgi?fileName=hdf_filename@cite NASA2001)

## Examples

```
# Source MODIS files (both data and geolocation)
# Code excluded from CRAN check because it depends on modiscdata
## Not run:
library(devtools)
library(modiscdata)
moddir = system.file("extdata/2002raw/", package="modiscdata")

# Get the matching data/geolocation file pairs
fns_df = check_for_matching_geolocation_files_mod02nrt(moddir, modtxt="MOD02", geolotxt="MOD03")
fns_df

# Resulting TIF files go in this directory
tifsdir = getwd()

# Box to subset
ul_lat = 13
ul_lon = -87
```



```
lr_lat = 8
lr_lon = -82

for (i in 1:nrow(fns_df))
{

prmfnc = write_MRTSwath_param_file_mod02nrt(prmfnc="tmpMRTparams.prm", tifsdirc=tifsdirc,
modfnc=fns_dfc$mod02_fnc[i], geoloc_fnc=fns_dfc$mod03_fnc[i], ul_lon=ul_lon, ul_lat=ul_lat,
lr_lon=lr_lon, lr_lat=lr_lat)
print(scan(file=prmfnc, what="character", sep="\n"))

}

## End(Not run)
```

# Index

adf, [2](#)

check\_for\_matching\_geolocation\_files\_mod02nrt, [2](#)

extract\_fn\_from\_path, [4](#)

run\_swath2grid\_mod02nrt, [4](#), [8](#)

slasheslash, [6](#)

write\_MRTSwath\_param\_file\_mod02nrt, [5](#), [7](#)