

Package ‘MODISTools’

September 8, 2014

Type Package

Title MODIS Subsetting Tools

Version 0.94.4

Date 2014-09-08

Author Sean Tuck, Helen Phillips

Imports RCurl, XML

Maintainer Sean Tuck <sean.tuck@plants.ox.ac.uk>

Description Provides several functions for downloading, storing and processing subsets of MODIS Land Processes data as a batch process.

License GPL-3

LazyLoad yes

LazyData yes

URL <https://github.com/seantuck12/MODISTools/>

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-08 22:59:24

R topics documented:

BatchDownload	2
ConvertExample	2
ConvertToDD	3
EndCoordinates	4
EndCoordinatesExample	5
ExtractTile	6
FindID	7

FindIDExample	8
GetBands	9
GetDates	10
GetProducts	11
GetSubset	12
LandCover	14
MODISGrid	15
MODISSubsets	17
MODISSummaries	19
MODISTimeSeries	22
MODISTools	23
MODISTransects	25
QualityCheck	27
QualityCheckExample	28
SubsetExample	29
TransectExample	30
UpdateSubsets	31

Index 33

BatchDownload *Internal function.*

Description

An internal function, used by MODISSubsets, that is not intended for use independently. Use MODISSubsets for download of subsets of MODIS data.

ConvertExample *Example dataset for ConvertToDD*

Description

A dataset consisting of a set of arbitrary locations in degrees minutes seconds (DMS) format that ConvertToDD can convert to decimal degrees format.

Usage

```
data(ConvertExample)
```

Format

A data frame with 4 observations on the following 2 variables.

lat A dataframe column containing latitude degrees minutes seconds (DDdMMmSS.SSSsN. ConvertToDD can take other formats and layouts in general.

long A dataframe column containing longitude degrees minutes seconds (DDdMMmSS.SSSsW. ConvertToDD can take other formats and layouts in general.

Details

The various compatible layouts for each format are described at ?ConvertToDD (see also). If hemispheres are denoted using letters (N/S; E/W) then the same transformation (+ive -ive coordinates) will be carried out on all points. Therefore, in this case, if points within a dataset are in different hemispheres, they will be all be placed into one.

Source

Locations were arbitrarily drawn from the grounds of Silwood Park Campus, Imperial College London.

See Also

[ConvertToDD](#)

ConvertToDD	<i>Degrees Minutes Seconds/Decimal Minutes - Decimal Degrees Converter</i>
-------------	--

Description

Takes a set of coordinates in either degrees minutes seconds or degrees decimal minutes, and converts them to decimal degrees format (##.#####; -##.#####). The function is flexible to take the input coordinates in several different formats.

Usage

```
ConvertToDD(XY, FileSep=NULL, LatColName, LongColName)
```

Arguments

XY	Input dataset; either the name of an object already in the workspace, or a file to be read in by specifying its file path as a character string.
FileSep	If XY is a character string that corresponds to a file path, choose the delimiter character for that file (e.g. "," for comma separated).
LatColName	Character string; the name of the column in XY containing latitude data.
LongColName	Character string; the name of the column in XY containing longitude data.

Details

There is flexibility in the layout of each degrees input format. For example, N-S/E-W can be described by positive-negative coordinates or by using respective letters at the end of each coordinate. For coordinates in degrees minutes seconds to be converted requires three numbers - degrees, minutes, and seconds - each separated by a single non-numeric character (e.g. 51d24'51.106"N 0d38'56.018"W). For coordinates in degrees minutes to be converted requires just two numbers - degrees and minutes (e.g. 51d24.106°N 0d38.018°W). Coordinates should not contain spaces in between the numbers. The degrees format does not have to be consistent throughout the data set.

Value

Output is a data frame of same dimensions as input, with each value converted to decimal degrees, named "DD.lat" and "DD.long".

Author(s)

Sean Tuck

See Also

[MODISSubsets](#)

Examples

```
data(ConvertExample)
ConvertToDD(XY=ConvertExample, LatColName="lat", LongColName="long")
```

EndCoordinates	<i>Calculate end location of a transect</i>
----------------	---

Description

Function that uses a set of locations in decimal degrees to calculate new locations, at a specified distance and angle away, that can form the start and end coordinates for transects.

Usage

```
EndCoordinates(LoadDat, FileSep = NULL, Distance = 1000, Angle = 90,
  AngleUnits = 'radians'|'degrees', Dir = ".", FileName = "Transect Coordinates")
```

Arguments

LoadDat	Input data; the name of an object or a character string that specifies a file path for an external file to be read in, containing a set of locations to be used for calculating new end coordinates.
FileSep	If a file path is entered to LoadDat, specify the file delimiter character.
Distance	Numeric; distance, in metres, that the end coordinates should be from the focal coordinates.
Angle	Numeric; angle, in either degrees or radians, that the end coordinates should be from the focal coordinates.
AngleUnits	Character String; the unit the angle is given in, either 'radians' or 'degrees'.
Dir	Character string; an optional argument to specify the subdirectory where the output file should be saved. Default Dir='.' writes file to the working directory.
FileName	Character string; an optional argument to specify the name given to the output file. Default is "Transect Coordinates".

Details

Input dataset, LoadDat, should contain separate columns for latitudes and longitudes, named 'start.lat' and 'start.long' respectively. Coordinates should be in decimal degrees.

Value

A new csv file containing the new end coordinate data appended to the original data.

Author(s)

Helen Phillips

See Also

[MODISTransects ConvertToDD](#)

Examples

```
data(EndCoordinatesExample)

EndCoordinates(LoadDat=EndCoordinatesExample,
  Distance = 2000, Angle = 90, AngleUnits = 'degrees')
```

EndCoordinatesExample *Example dataset for EndCoordinates()*

Description

A dataset consisting of a set of arbitrary locations with a corresponding set of arbitrary start and end time-series dates and IDs. This file can be used in the example for the MODIS EndCoordinates function.

Usage

```
data(EndCoordinatesExample)
```

Format

A data frame with 4 observation on the following 5 variables.

start.lat A numeric vector; decimal degrees latitudes in WGS-1984 coordinate system.

start.long A numeric vector; decimal degrees longitudes in WGS-1984 coordinate system.

start.date A numeric vector; listing the date to begin the time-series for each corresponding locations. Used during MODISTransects.

end.date A numeric vector; listing the date to end the time-series for each corresponding locations. Used during MODISTransects.

ID A numeric vector; a unique ID code for each unique location.

Source

Locations were arbitrarily drawn from the grounds of Silwood Park Campus, Imperial College London.

See Also

[MODISSubsets](#)

ExtractTile

Extract subset tiles from bigger tiles of MODIS data.

Description

Input a dataset of MODIS data, comprised of one or many tiles of pixels - a column for each pixel in a tile and a row for each tile - and extract a nested subset from within these tiles.

Usage

```
ExtractTile(Data, Rows, Cols, Grid=FALSE)
```

Arguments

Data	Numeric vector, matrix, or data frame; The input data, containing the large tile(s) that smaller tiles will be extracted from. If a matrix or data frame, each row should represent a distinct tile, whilst each column in a row will be a different pixel within the that tile.
Rows	Numeric - two integers; Rows[1] should be the number of rows in the large tile(s) of pixels. Rows[2] is the number of rows you would like either side of the tile(s) central pixel in the small tile(s) output.
Cols	Numeric - two integers; Cols[1] should be the number of columns in the large tile(s) of pixels. Cols[2] is the number of columns you would like either side of the tile(s) central pixel in the small tile(s) output.
Grid	Logical; if Grid=FALSE, the smaller tile(s) will be in a matrix, like the input. If Grid=TRUE, the output will be an array, with tile(s) presented explicitly. See value.

Details

Data should only include MODIS data and not any other metadata for the input tiles (see example). Rows[2] and Cols[2] should equate to a nested subset of Rows[1] and Cols[1]. Rows[1] and Cols[1] specify the dimensions of the tiles, laid out in rows in matrix, and therefore should equate to the dimensions of a matrix that is filled by ncols(Data) data points.

Value

The output may have two possible structures, optionally chosen with the Grid argument. If Grid=FALSE, the output will be the same structure as the input - a nxm matrix, where n is the number of tiles and m is the number of pixels in each tile - with the same number of rows but only the number of pixels in the smaller tile(s) for each row. If Grid=TRUE, the output will be an array, with the tiles laid out with pixels in spatial order - a matrix of the subset of pixels, for each row in Data.

Author(s)

Sean Tuck

Examples

```
## Not run: # dontrun() used because running the example requires internet access.
data(SubsetExample)
MODISSubsets(LoadDat = SubsetExample, Products = "MOD13Q1",
             Bands = c("250m_16_days_EVI", "250m_16_days_pixel_reliability"), Size = c(1,1),
             StartDate = FALSE, TimeSeriesLength = 1)
MODISSummaries(LoadDat = SubsetExample, Product = "MOD13Q1", Bands = "250m_16_days_EVI",
              ValidRange = c(-2000,10000), NoDataFill = -3000, ScaleFactor = 0.0001,
              StartDate = FALSE, QualityScreen = TRUE, QualityThreshold = 0,
              QualityBand = "250m_16_days_pixel_reliability")

if(sum(grepl("MODIS_Data", list.files())) != 1){
  print("Could not identify 'MODIS_Data' csv output file from MODISSummaries")
} else {
  TileExample <- read.csv(list.files(pattern = "MODIS_Data"))
  TileExample <- TileExample[1,which(grepl("pixel", names(TileExample)))]

  dim(TileExample)
  dim(ExtractTile(Data = TileExample, Rows = c(9,2), Cols = c(9,2), Grid = FALSE))
  ExtractTile(Data = TileExample, Rows = c(9,2), Cols = c(9,2), Grid = FALSE)

  matrix(TileExample, nrow = 9, ncol = 9, byrow = TRUE)
  ExtractTile(Data = TileExample, Rows = c(9,2), Cols = c(9,2), Grid = TRUE)
}

## End(Not run)
```

FindID

Time-series matching to IDs

Description

A function that takes a unique time-series definition and looks inside a large dataset to find all the IDs that correspond to records with matching time-series definitions. A useful function for maintaining transparency in the data collection and processing chain.

Usage

```
FindID(ID, Data)
```

Arguments

ID	A data frame with one or a set of rows containing a time-series definition, in terms of lat, long, and dates information.
Data	A larger data frame consisting of at least the same information as ID, which is to be looked in to find matches.

Details

Rows in ID do not need to be the same order as in Data, but the variables in dat to be looked in must have the same names as those in ID. Date information can include one date, or optionally a start and end date. Any information in ID must be included in Data.

Value

A set of character strings, listing all the IDs that correspond to the given unique time-series.

Author(s)

Sean Tuck

See Also

[MODISSummaries](#)

Examples

```
data(SubsetExample, FindIDExample)  
FindID(ID=SubsetExample, Data=FindIDExample)
```

FindIDExample

Example dataset for FindIDSubsets()

Description

A dataset consisting of a set of arbitrary locations with a corresponding set of arbitrary start and end time-series dates and IDs. This file can be used in the example for the function that takes a set of time-series and finds all IDs for records in a larger data frame that match the time-series definition.

Usage

```
data(FindIDExample)
```

Format

A data frame with 4 observations on the following 5 variables.

lat A numeric vector; decimal degrees latitudes in WGS-1984 coordinate system.

long A numeric vector; decimal degrees longitudes in WGS-1984 coordinate system.

start.date A numeric vector; listing the date to begin the time-series for each corresponding location.

end.date A numeric vector; listing the date to end the time-series for each corresponding location.

ID A numeric vector; a unique ID code for each unique time-series (either unique in time or in space).

Details

Rows in ID do not need to be the same order as in Data, but the variables in Data to be looked in must have the same names as those in ID. Date information can include one date, or optionally a start and end date. Any information in ID must be included in Data.

Source

Locations were arbitrarily drawn from the grounds of Silwood Park Campus, Imperial College London.

See Also

[MODISubsets](#)

GetBands

Get MODIS Product Data Band Names

Description

Internal function that uses the MODIS SOAP Web Service to retrieve a list of all of the codes to identify the data bands within a specified MODIS product.

Usage

```
GetBands(Product)
```

Arguments

Product Character string; the unique code for a given MODIS product. Will only accept one product code. See references for the product list.

Value

A character vector; containing the codes for each data band.

Author(s)

Sean Tuck

Referenceshttp://daac.ornl.gov/MODIS/MODIS-menu/modis_webservice.htmlhttps://lpdaac.usgs.gov/products/modis_products_tablehttp://daac.ornl.gov/cgi-bin/MODIS/GLBVIZ_1_Glb/modis_subset_order_global_col5.pl**See Also**[MODISSubsets](#)**Examples**

```
## Not run: # dontrun() used because running the example requires internet access.
GetBands("MOD13Q1")
## End(Not run)
```

 GetDates

Get list of available dates for requested time-series

Description

Internal function that uses the MODIS SOAP Web Service to retrieve a list of all dates when the requested data, specified by MODIS product and a location, are available for download.

Usage

```
GetDates(Lat, Long, Product)
```

Arguments

Lat	Numeric; a decimal degrees latitude in WGS-1984 coordinate system.
Long	Numeric; a decimal degrees longitude in WGS-1984 coordinate system.
Product	Character string; a product code to request subsets from. The MODIS product table shows all available products and their respective product titles (see references).

Value

A character vector; each element representing a time-step (in MODIS date format) where data is available.

Author(s)

Sean Tuck

References

http://daac.ornl.gov/MODIS/MODIS-menu/modis_webservice.html

http://daac.ornl.gov/cgi-bin/MODIS/GLBVIZ_1_Glb/modis_subset_order_global_col5.pl

https://lpdaac.usgs.gov/products/modis_products_table

See Also

[MODISSubsets](#)

Examples

```
## Not run: # don't run() used because running the example requires internet access.
GetDates(Lat=51.41363, Long=-0.64875, Product="MOD13Q1")
## End(Not run)
```

GetProducts

Get MODIS Product List

Description

Internal function that uses the MODIS SOAP Web Service to retrieve a list of all of the codes for available MODIS products.

Usage

```
GetProducts()
```

Arguments

No input.

Value

A character vector; containing the codes for each product.

Author(s)

Sean Tuck

References

http://daac.ornl.gov/MODIS/MODIS-menu/modis_webservice.html
http://daac.ornl.gov/cgi-bin/MODIS/GLBVIZ_1_Glb/modis_subset_order_global_col5.pl
https://lpdaac.usgs.gov/products/modis_products_table

See Also

[MODISSubsets](#)

Examples

```
## Not run: # dontrun() used because running the example requires internet access.
GetProducts()
## End(Not run)
```

GetSubset	<i>Subset download using MODIS web service method.</i>
-----------	--

Description

Internal function that uses the MODIS SOAP Web Service (see references) `getsubset` method to download a requested time-series subset of a MODIS product for a given area surrounding many locations. This function is used by the `MODISSubsets` function, which is the main subset download tool for the user.

Usage

```
GetSubset(Lat, Long, Product, Band, StartDate, EndDate, KmAboveBelow, KmLeftRight)
```

Arguments

Lat	Numeric; a decimal degrees latitude in WGS-1984 coordinate system.
Long	Numeric; a decimal degrees longitude in WGS-1984 coordinate system.
Product	A character string; a product code to request the subset from. The MODIS product table shows all available products and their respective product titles (see references).
Band	A character string; a code or vector of codes to retrieve the desired data band(s) within the specified product to be requested. To get a list of the available bands in a product of interest use <code>GetBands()</code> .
StartDate	Character; in MODIS date format, listing the date to begin the time-series for each corresponding location.
EndDate	Character; in MODIS date format, listing the date to end the time-series for each corresponding location.

KmAboveBelow	An integer; the distance, in kilometres, from the centre of the tile of pixels to the top and bottom of the tile. So, if KmAboveBelow=1, the total distance from top to bottom of the tile will be 2km.
KmLeftRight	An integer; the distance, in kilometres, from the centre of the tile of pixels to the left and right sides of the tile. So, if KmLeftRight=1, the total distance from side to side of the tile will be 2km.

Value

A data frame containing: xll=The x coordinate, in the MODIS coordinate system, of the lower left corner of the pixel in which the location data falls. yll=The y coordinate, in the MODIS coordinate system, of the lower left corner of the pixel in which the location data falls. pixelsize=The actual calculated width of the pixels of interest, in metres. nrow=Corresponds to KmAboveBelow argument. ncol=Corresponds to KmLeftRight argument. band=Corresponds to band argument. scale=The number by which raw downloaded data values can be scaled by. Specific to each band. lat=Corresponds to the Lat argument. long=Corresponds to the Long argument. subset=The downloaded data, as a vector of character strings, with length number of time-steps requested, that include data attached to metadata.

This data frame is then passed to MODISSubsets.

Author(s)

Sean Tuck

References

http://daac.ornl.gov/MODIS/MODIS-menu/modis_webservice.html

https://lpdaac.usgs.gov/products/modis_products_table

http://daac.ornl.gov/cgi-bin/MODIS/GLBVIZ_1_Glb/modis_subset_order_global_col5.pl

See Also

[MODISSubsets](#)

Examples

```
## Not run:
# dontrun() used because running the example requires internet access,
# and takes over a minute to run.
data(SubsetExample)
GetSubset(Lat=SubsetExample[,1], Long=SubsetExample[,2], Product="MOD13Q1",
          Band="250m_16_days_EVI", StartDate="A2001001", EndDate="A2001025",
          KmAboveBelow=0, KmLeftRight=0)
## End(Not run)
```

LandCover

Summaries on tiles of land cover data.

Description

The MODIS product MCD12Q1 contains data bands for land cover types. This function summarises that data, reporting: the most common land cover type, the richness of land cover types, and landscape diversity and heterogeneity using Simpson's D and Simpson's measure of evenness.

Usage

```
LandCover(Dir=".", Band)
```

Arguments

Dir	Character string; an optional argument to specify the subdirectory where downloaded ascii files to be processed are located and the output is written: default Dir="." extracts files from the working directory.
Band	Character string; the shortname code that identifies what band type of land cover data from the MCD12Q1 product is being processed.

Details

No data fill values are removed from the tiles before summaries are reported. Additional to the no data fill values, the land cover bands include an 'Unclassified' land cover class. This would indicate poor quality, but is not removed.

Value

One csv file is written as an output to the function. It contains all of the summaries for each tile input, from all ascii files read in. Each row corresponds to a different tile of pixels - from a different time-step in the time-series, or from a different time-series (ascii file) altogether. Columns are: latitude, longitude, date, land cover band type, most common land cover class, landscape richness, Simpson's D, Simpson's evenness, proportion of data that are no data fill values.

Author(s)

Sean Tuck

References

https://lpdaac.usgs.gov/products/modis_products_table/mcd12q1

See Also

[MODISubsets](#) [MODISSummaries](#)

Examples

```
## Not run: # dontrun() used because running the example requires internet access.
data(SubsetExample)
# Extend the years to download for
SubsetExample$start.date <- 2001
SubsetExample$end.date <- 2009
MODISSubsets(LoadDat=SubsetExample, Product="MCD12Q1", Bands=c("Land_Cover_Type_1"),
             Size=c(3,3), StartDate=TRUE)

LandCover(Band="Land_Cover_Type_1")

## End(Not run)
```

 MODISGrid

Create GIS ASCII files from downloaded MODIS data files

Description

Looks for ASCII files of downloaded MODIS data in a directory and creates new ASCII raster grid files that allow data downloaded using MODISTools to connect to a GIS environment. The new raster grid files can be converted into raster images and loaded into a GIS environment where further data processing or spatial analysis can take place.

Usage

```
MODISGrid(Dir = ".", DirName = "MODIS_GRID", SubDir = TRUE, NoDataValues)
```

Arguments

Dir	Character; file path to the directory where downloaded ASCII files to be processed are located and the output is written: default Dir = "." extracts files from the working directory.
DirName	Character; the name of the directory (a subdirectory found in directory specified as Dir) where the new ASCII raster grids will be written. If the directory does not already exist, it will be written.
SubDir	Logical; if TRUE, all ASCII raster grids created from one MODIS data file will be collected under a subdirectory, within DirName, that has the same name as the MODIS data file from which it was created. If FALSE all ASCII raster grids will simply be written to DirName, with no nested file structure.
NoDataValues	Named numeric vector; the no data values for all the data bands contained in the files to be processed. All data bands found in the MODIS data files should be listed in this named vector. These values are necessary to identify pixels with missing data. The correct no data values for the data bands of interest can be found at https://lpdaac.usgs.gov/products/modis_products_table .

Details

A raster grid file is created for each grid found in a MODIS data file (i.e. a new file is written for each date that occurs in a time series). In other words, where the MODIS data file may follow a tile of pixels over time, the raster grids split up each tile to create a new file for that tile at each new date. Therefore, multiple raster grids may be created from one MODIS data file. These files can be stored in one directory or nested in subdirectories for each MODIS data file, using the optional argument `SubDir`. By default, the subdirectories are created to retain the spatial structure.

The data are in the MODIS projection system. When converted these ASCII raster grids to raster images, the PRJ file for the MODIS projection system will be required to project the data in the correct datum. For this PRJ file, access <http://daac.ornl.gov/MODIS/modis.prj>. To combine a raster grid with a projection, the two files must be stored in the same directory with the same file name. Every raster grid needs a respective PRJ file. Once combined with this PRJ file and converted into a raster image, the user can easily convert the data into a datum of more general usage by working in a GIS environment.

Value

Multiple ASCII raster grid files, containing a tile of MODIS pixels at a given date, either all written to one directory or separated into subdirectories for each MODIS data file found. The file names for each raster grid created are composed of the following: "GRID_", the file name of the relevant MODIS data file, the data band that the grid pertains to, and finally the MODIS acquisition date for the grid (in Julian date format).

Author(s)

Sean Tuck

See Also

[MODISSubsets](#) [MODISSummaries](#) [ExtractTile](#)

Examples

```
## Not run:
# dontrun() used because running the example requires internet access.

dat <- EndCoordinatesExample
names(dat) <- c("lat", "long", "start.date", "end.date", "ID")

MODISSubsets(dat, Products = "MOD13Q1", Size = c(1,1), StartDate = TRUE,
             Bands = c("250m_16_days_EVI", "250m_16_days_NDVI", "250m_16_days_pixel_reliability"))
MODISGrid(NoDataValues = c("250m_16_days_EVI" = -3000, "250m_16_days_NDVI" = -3000,
                          "250m_16_days_pixel_reliability" = -1))

## End(Not run)
```

Description

Uses the MODIS SOAP Web Service (see references) as a batch process to remotely extract and locally download time-series subsets of a MODIS product for a given area surrounding many locations.

Usage

```
MODISSubsets(LoadDat, FileSep = NULL, Products, Bands, Size,
             SaveDir = ".", StartDate = FALSE, TimeSeriesLength = 0, Transect = FALSE)
```

Arguments

LoadDat	Input dataset: either the name of an object already in the workspace, or a file to be read in by specifying its file path as a character string, that has location data, dates (end date, and optionally start date) and study ID for each location. For required data organisation, see Details.
FileSep	If LoadDat is a character string that corresponds to a file path, choose the delimiter character for that file (e.g. "," for comma separated).
Products	Character string; a product code to request subsets from. The MODIS product table shows all available products and their respective product titles (see references). Data from multiple Products can be downloaded.
Bands	A character vector; each string a code to retrieve the desired data bands within the specified product to be requested. To get a list of the available bands in a product of interest, use GetBands(). Multiple data band across multiple products can be specified and downloaded at the same time.
Size	Numeric vector of two non-negative integers defining the dimensions of tile requested at each location. The first element identifies the distance from the centre to the bottom/top (in both directions), and the second element to the left/right (in both directions) in km. For example, Size = c(0,0) identifies the centre pixel only, whereas Size = c(1,1) identifies a tile of 2kmsq.
SaveDir	Character string; an optional argument to specify a file path to the subdirectory where downloaded ASCII files should be saved: default, SaveDir = ".", saves the files to the working directory. The directory where the downloaded data files will be located is printed at the beginning of the function call. These downloaded files can then be re-loaded into the R environment in a manageable way by using the function MODISTimeSeries.
StartDate	Logical; indicate whether the input dataset contains information on the time-series start date. If StartDate = TRUE, start dates will be taken from the input data and will expect the data frame to have a column named start.date. Default is StartDate = FALSE, whereby the input data is assumed to have only time-series end date and the length of time-series recorded prior to that date is determined by another optional argument, TimeSeriesLength.

TimeSeriesLength	Non-negative integer; if StartDate = FALSE specified the length of time-series. Default TimeSeriesLength = 0, giving a time-series length of 1 year, taken from the beginning to the end of the year.
Transect	Logical; argument to be used when MODISSubsets() is called in MODISTransects() to merge all subset points for each transect into one file. Should not be optionally used - only useful for MODISTransects().

Details

The data should be organised accordingly: "lat" and "long" columns showing WGS-1984 decimal degrees latitudes and longitudes respectively; "end.date" for study end date (and optionally "start.date"), that can be in years or in POSIXlt date format; "ID" containing a unique ID for each unique time-series (in time or space). Column names should be exactly as shown.

Value

An ASCII file written for each unique time-series (unique location, or unique time-series at a duplicated location), containing all data requested, for each band, for every pixel in the specified tile size. A subset download CSV file will also be written, that contains all the unique time-series information, and a status report for each time-series telling the user if the respective time-series was downloaded successfully or identifying any problems with that particular download.

Author(s)

Sean Tuck

References

http://daac.ornl.gov/MODIS/MODIS-menu/modis_webservice.html
https://lpdaac.usgs.gov/products/modis_products_table
http://daac.ornl.gov/cgi-bin/MODIS/GLBVIZ_1_Glb/modis_subset_order_global_col5.pl

See Also

[MODISSummaries](#) [MODISTransects](#) [MODISGrid](#) [MODISTimeSeries](#)

Examples

```
## Not run:
# dontrun() used because running the example requires internet access.

data(SubsetExample)

MODISSubsets(LoadDat = SubsetExample, Products = "MOD13Q1", Bands = c("250m_16_days_EVI",
  "250m_16_days_pixel_reliability"), Size = c(0,0), StartDate = TRUE)

MODISSummaries(LoadDat = SubsetExample, Product = "MOD13Q1", Bands = "250m_16_days_EVI",
  ValidRange = c(-2000,10000), NoDataFill = -3000, ScaleFactor = 0.0001,
```

```
StartDate = TRUE, QualityScreen = TRUE, QualityThreshold = 0,
QualityBand = "250m_16_days_pixel_reliability")
```

```
## End(Not run)
```

 MODISSummaries

MODIS subset processing & organisation tool

Description

A function to run time-series analysis and compute summary statistics for a downloaded MODIS subset, writing a summary file and another file with processed MODIS data tagged onto the original file inputted to MODISSubsets. This function allows the user to easily explore the characteristics of the downloaded data, and then process them into a form that is ready for use in modelling.

Usage

```
MODISSummaries(LoadDat, FileSep = NULL, Dir = ".", Product, Bands, ValidRange, NoDataFill,
ScaleFactor, StartDate = FALSE, QualityScreen = FALSE, QualityBand = NULL,
QualityThreshold = NULL, Mean = TRUE, SD = TRUE, Min = TRUE, Max = TRUE, Yield = FALSE,
Interpolate = FALSE, InterpolateN = NULL, DiagnosticPlot = FALSE)
```

Arguments

LoadDat	Input dataset: either the name of an object already in the workspace, or a file to be read in by specifying its file path as a character string, that has location data, dates (end date, and optionally start date) and study ID for each location. If IDs are found in LoadDat that provide a primary key for unique time series then these IDs will be used. Otherwise a set of unique IDs will be generated and used to identify, and file name, each time series.
FileSep	If LoadDat is a character string that corresponds to a file path, choose the delimiter character for that file (e.g. "," for comma separated).
Dir	Character string; an optional argument to specify a file path to the subdirectory where downloaded ASCII files to be processed are located and the output is written: default Dir = "." extracts files from the working directory.
Product	Character; The product shortname code, that the data band input belongs to. The MODIS product table shows all available products and their respective product shortname codes (see references).
Bands	Character; the code that identifies from which band types are the data to be processed. Multiple bands can be specified as a character vector, including the quality control data bands, providing they all come from the same product. With the exception of BRDF Reflectance data products (MCD43A4) that have quality information stored as a separate product (MCD43A2).
ValidRange	Numeric vector of two elements; states the lower (ValidRange[1]) and upper (ValidRange[2]) bounds within which the data to be processed should be found.

NoDataFill	Numeric; the missing data fill value that is used for Bands.
ScaleFactor	Numeric; The specified scaling for the given band type, which the data is to be multiplied by. If a scale factor does not exist for the data band, ScaleFactor should be set to 1.
StartDate	Logical; indicate whether the input dataset contains information on the time-series start date. If StartDate = TRUE, start dates will be taken from the input data and will expect the data frame to have a column named start.date. Default is StartDate = FALSE, whereby the input data is assumed to have only time-series end date. This should be the same as that used in the relevant call to MODISSubsets.
QualityScreen	Logical; optional argument for screening the band data for unreliable pixels. If QualityScreen = TRUE, band data must be downloaded from MODISSubsets with the quality control data corresponding to the same product included. Therefore, both band data and reliability data will be in the same ASCII files for each time-series downloaded. Quality screening is completed by the QualityCheck function, and the arguments for this function need to be included in a MODISSummaries call, if QualityScreen = TRUE. The default is QualityScreen = FALSE, meaning the function will omit data equal to NoDataFill, but will not omit poor quality data.
QualityBand	Character; if QualityScreen = TRUE, the shortname code for the quality data band that you are using to screen Band for poor quality data.
QualityThreshold	Numeric integer; if QualityScreen = TRUE, set the threshold between acceptable and unacceptable quality. Any pixels of lower quality than the class set by QualityThreshold will be removed, and those equal to or of higher quality will be kept. QualityThreshold should be a number within the range of possible QualityScores for the given Product QA data.
Mean, SD, Min, Max, Yield	Logical; optional arguments that allow selecting which summaries will be included in the summary file that gets written - see value. Selecting Yield requires Interpolate to also be set as TRUE.
Interpolate	Logical; determines whether, after poor quality data is removed, to linearly interpolate between high quality data before calculating the summary statistics. Must be TRUE if Yield = TRUE. The interpolation function used is stats::approx. See ?stats::approx for more details.
InterpolateN	Numeric; if Interpolate = TRUE, optionally set the number interpolated data points to be requested from the time-series interpolation. The default is set to a daily interpolation of the data.
DiagnosticPlot	Logical; if TRUE will produce an additional folder in the specified directory to which plots of the time series data for each site will be saved. Will add the interpolation line, mean, min and max values if specified in the function call.

Details

If QualityScreen = TRUE, subsets to be processed should include a pixel reliability layer, so the data can be screened for poor quality data, removing them and using linear interpolation to refill data between high quality values.

Value

Two CSV files: One file (MODIS_Summary...) contains summary statistics and computed values for each data. The information this file contains is partly defined by the optional arguments settings: Mean is arithmetic mean; SD is standard deviation; Min and Max are minimum and maximum band values; Yield is the average annual yield (designed for vegetation indices, may not be sensible for all band types); NoFill and PoorQuality show the percentage of values in each time-series that were NoDataFill and omitted by QualityCheck (if QualityScreen = TRUE) respectively. All summary statistics, except yield, are included by default.

The second file (MODIS_Data...) that has the information from the original file inputted (which should have been used in MODISSubsets too) with computed means of the MODIS data tagged on, coupling the input with the output in one form ready for use, such as modelling. In the second file, each nth column of MODIS data, if more than one, will be for each pixel within the whole tile of n pixels collected for the time-series on that row.

Author(s)

Sean Tuck

References

http://daac.ornl.gov/MODIS/MODIS-menu/modis_webservice.html

https://lpdaac.usgs.gov/products/modis_products_table

http://daac.ornl.gov/cgi-bin/MODIS/GLBVIZ_1_Glb/modis_subset_order_global_col5.pl

See Also

[MODISSubsets QualityCheck](#)

Examples

```
## Not run:
# dontrun() used because running the example requires internet access,
# and takes over a minute to run.

data(SubsetExample)

MODISSubsets(LoadDat = SubsetExample, Products = "MOD13Q1",
             Bands = c("250m_16_days_EVI", "250m_16_days_NDVI", "250m_16_days_pixel_reliability"),
             Size = c(0,0), StartDate = TRUE)

# Without quality checking
MODISSummaries(LoadDat = SubsetExample, Product = "MOD13Q1", Bands = "250m_16_days_EVI",
              ValidRange = c(-2000,10000), NoDataFill = -3000, ScaleFactor = 0.0001,
              StartDate = TRUE)

# With quality checking
MODISSummaries(LoadDat = SubsetExample, Product = "MOD13Q1", Bands = "250m_16_days_EVI",
              ValidRange = c(-2000,10000), NoDataFill = -3000, ScaleFactor = 0.0001,
```

```

StartDate = TRUE, QualityScreen = TRUE, QualityThreshold = 0,
QualityBand = "250m_16_days_pixel_reliability")

# For both EVI and NDVI
MODISSummaries(LoadDat = SubsetExample, Product = "MOD13Q1",
Bands = c("250m_16_days_EVI", "250m_16_days_NDVI"),
ValidRange = c(-2000,10000), NoDataFill = -3000, ScaleFactor = 0.0001,
StartDate = TRUE, QualityScreen = TRUE, QualityThreshold = 0,
QualityBand = "250m_16_days_pixel_reliability")

## End(Not run)

```

MODISTimeSeries *Create one matrix of data from many downloaded timeseries.*

Description

Take many downloaded timeseries of a MODIS data band and simplify them into one list or matrix of data, with a row for each date and a column for each unique timeseries (i.e., each ASCII file).

Usage

```
MODISTimeSeries(Dir, Band, Simplify = FALSE)
```

Arguments

Dir	Character; the directory in which to search for downloaded ASCII files of MODIS data.
Band	Character; the MODIS data band shortname for the downloaded data.
Simplify	Logical; specifies whether the data output should be presented as a list (by default) or simplified to one matrix. See details for more information.

Details

Dir can contain files other than the ASCII files containing data for Band, including other MODIS ASCII files. But, only the data for Band will be included in the output dataset. The names of the downloaded ASCII files must be their original names, containing their correct data product code. The timeseries length of different subsets can be different (the output dataset will contain NAs in the final rows of any timeseries that are shorter than the maximum).

If Simplify = FALSE, each element in the list represents a unique time series tile as a matrix, with columns for each pixel in that tile and rows for each date in the time series. If Simplify = TRUE, the matrices for each time series tile will be appended together (columnwise) to form one large matrix with the same number of rows (for each date) but many columns (for all pixels of interest). If, however, the number of dates varies in different timeseries, the number of rows in each tile matrix will not be equal and these data cannot be simplified into a matrix. In this case, the data will always be returned as a list.

Value

A list with elements for each time series tile (i.e., each MODIS data file), and a matrix with columns for pixels and rows for dates in each element. Or, one large matrix with rows for each date and columns for each unique pixel across all MODIS data of interest (found in Dir).

Author(s)

Sean Tuck

See Also

[MODISSubsets](#) [MODISSummaries](#)

Examples

```
## Not run:
# dontrun() used because running the example requires internet access,
# and takes over a minute to run.
time.series <- data.frame(lat = c(51.41363, 51.41421),
                          long = c(-0.64875, -0.641607),
                          start.date = c(2002, 2002),
                          end.date = c(2004, 2004),
                          ID = c(1, 2))

MODISSubsets(LoadDat = time.series, Product = "MOD13Q1",
             Bands = c("250m_16_days_EVI"), Size = c(0,0), StartDate = TRUE)

ts <- MODISTimeSeries(Dir = ".", Band = "250m_16_days_EVI", Simplify = FALSE)
class(ts)
dim(ts[[1]])
dim(ts[[2]])

ts <- MODISTimeSeries(Dir = ".", Band = "250m_16_days_EVI", Simplify = TRUE)
class(ts)
dim(ts)

time.series$end.date[2] <- 2005
MODISSubsets(LoadDat = time.series, Product = "MOD13Q1",
             Bands = c("250m_16_days_EVI"), Size = c(1,1), StartDate = TRUE)

ts <- MODISTimeSeries(Dir = ".", Band = "250m_16_days_EVI", Simplify = TRUE)
class(ts)

## End(Not run)
```

Description

Provides an automated batch method for retrieving subsets of MODIS Land Processes data through the MODIS Web Service and processing them to a format ready for user friendly application in R such as statistical modelling. Datasets currently available - from terra, aqua, and combined platforms - through the web service: surface reflectance; land cover/land cover change; land surface temperature and emissivity; vegetation indices; leaf area index and FPAR; evapotranspiration; net photosynthesis and primary productivity.

Details

Imports RCurl and XML. The main purpose of this package is to employ the MODIS SOAP Web Service to remotely interrogate the MODIS archive and locally download the requested datasets. The most important function is MODISSubsets, which allows the user to request subsets from a given MODIS product for multiple time-series. Each time-series is defined by a coordinate location (WGS-1984), a specified surrounding extent of pixels, and a start and end date. The relevant MODIS product data is then extracted for these location/time combinations. Automating this as a batch process greatly reduces time, effort, and human error. Alternatively, MODISTransects expands upon MODISSubsets by extracting MODIS data along a transect, and its surrounding neighbourhood. Downloaded subsets are saved in ascii files, which are then accessed by MODISSummaries for computing summary statistics. It also organises downloaded data back with the original input data into a csv file that can be easily used for modelling; this provides efficient storage of data and a transparent process from data collection, to processing, to a form that is ready for final use.

Author(s)

Sean Tuck

References

http://daac.ornl.gov/MODIS/MODIS-menu/modis_webservice.html

http://daac.ornl.gov/cgi-bin/MODIS/GLBVIZ_1_Glb/modis_subset_order_global_col5.pl

https://lpdaac.usgs.gov/products/modis_products_table

<http://www.omegahat.org/RCurl>

<http://www.omegahat.org/RXML>

See Also

[MODISSubsets](#)

 MODIStransects

MODIS Transect Subset Tool

Description

A function that downloads batches of MODIS data transects.

Usage

```
MODIStransects(LoadData, FileSep = NULL, Product, Bands, Size,
  SaveDir = ".", StartDate = FALSE, TimeSeriesLength = 0)
```

Arguments

LoadData	Input data: either the name of an object already in the workspace, or a file to be read in by specifying its file path as a character string. For data input requirements, see details. All arguments to be passed to MODISSubsets(); see ?MODISSubsets for more information.
FileSep	If LoadData is a character string that corresponds to a file path, choose the delimiter character for that file (e.g. "," for comma separated).
Product	Character string; code denoting which MODIS product to be requested.
Bands	Character vector; which bands of data within specified product are of interest.
Size	Numeric vector of two non-negative integers defining the dimensions of tile requested at each location. The first element identifies the distance from the centre to the bottom/top (in both directions), and the second element to the left/right (in both directions) in km. For example, Size = c(0,0) identifies the centre pixel only, whereas Size = c(1,1) identifies a tile of 2kmsq.
SaveDir	Character string; an optional argument to specify the subdirectory where downloaded ascii files should be saved: default SaveDir = "." saves the files to the working directory.
StartDate	Logical; indicate whether the input dataset contains information on the time-series start date. If StartDate = TRUE, start dates will be taken from the input data and will expect the data frame to have a column named start.date. Default is StartDate = FALSE, whereby the input data is assumed to have only time-series end date and the length of time-series recorded prior to that date is determined by another optional argument, TimeSeriesLength.
TimeSeriesLength	Non-negative integer; if StartDate = FALSE specified the length of time-series. Default TimeSeriesLength = 0, giving a time-series length of 1 year, taken from the beginning to the end of the year.

Details

File input requirements are a transect ID, a start lat and long, and an end lat and long for each transect, and an end date (start date optional) for the requested time-series for each transect. These fields must be labelled with headers: "transect"; "start.lat"; "start.long"; "end.lat"; "end.long"; "start.date", and; "end.date" respectively.

All data types requested must be of the same resolution (pixel size). See ?MODISSubsets for more details.

The transects lengths requested must be longer than the requested pixel size. Therefore, each transect must contain more than 1 pixel.

Value

ASCII file written for each location, containing the specified time-series for each band of data requested. File name states the transect ID that the location is within, and a unique ID for that specific location.

Author(s)

Sean Tuck

References

http://daac.ornl.gov/MODIS/MODIS-menu/modis_webservice.html

http://daac.ornl.gov/cgi-bin/MODIS/GLBVIZ_1_Glb/modis_subset_order_global_col5.pl

https://lpdaac.usgs.gov/products/modis_products_table

See Also

[MODISSubsets](#)

Examples

```
## Not run:
# dontrun() used because running the example requires internet access,
# and takes over a minute to run.
data(TransectExample)

MODISTransects(LoadData = TransectExample, Product = "MOD13Q1",
  Bands = c("250m_16_days_EVI", "250m_16_days_pixel_reliability"),
  Size = c(0,0), StartDate = TRUE)

## End(Not run)
```

QualityCheck

Remove unreliable data and no fill values from MODIS data

Description

Designed as an internal function for MODISSummaries, but can also be used independently. MODIS data of a given product band are screened for poor quality pixels, using the reliability data available for that product. The missing data values are also filtered out. The output is the same MODIS data, with screened poor quality and missing data converted to NAs.

Usage

```
QualityCheck(Data, Product, Band, NoDataFill, QualityBand, QualityScores,
             QualityThreshold)
```

Arguments

Data	Numeric vector or matrix; the input MODIS data, for QualityCheck to screen for poor quality and missing data.
Product	Character; the product code for Data (e.g. "MOD15A2" if you want to screen the quality of a downloaded data band from the leaf area index product).
Band	Character; the shortname code for the data band that you are quality screening.
NoDataFill	Numeric; the missing data fill value for the data band (Data) that you are screening.
QualityBand	Character; the shortname code for the quality data band that you are using to screen Band for poor quality data.
QualityScores	Numeric integer vector or matrix; the quality control control that matches your Data input. QualityScores should be the same length/dimensions as Data.
QualityThreshold	Numeric integer; set the threshold between acceptable and unacceptable quality. Any pixels of lower quality than the class set by QualityThreshold will be removed, and those equal to or of higher quality will be kept. QualityThreshold should be a number within the range of possible QualityScores for the given Product QA data.

Details

Although there are several data bands within each MODIS product, only one band type of data should be passed to QualityCheck at a given time, as different bands within a product may have different NoDataFill values, and may even have different data bands that contain their relevant quality information.

Value

Data, as it was, but with poor quality data and no data fill values replaced with NAs.

Author(s)

Sean Tuck

References

http://daac.ornl.gov/MODIS/MODIS-menu/modis_webservice.html

https://lpdaac.usgs.gov/products/modis_products_table

http://daac.ornl.gov/cgi-bin/MODIS/GLBVIZ_1_Glb/modis_subset_order_global_col5.pl

See Also

[MODISSummaries](#)

Examples

```
## Not run:
# dontrun() used because running the example requires internet access.
data(QualityCheckExample)
EVIdata <- QualityCheckExample[1:5, ]
QAdata <- QualityCheckExample[6:10, ]

QualityCheck(Data=EVIdata, Product="MOD13Q1", Band="250m_16_days_EVI",
             NoDataFill=-3000, QualityBand="250m_16_days_pixel_reliability",
             QualityScores=QAdata, QualityThreshold=0)

## End(Not run)
```

QualityCheckExample *Example dataset for QualityCheck*

Description

A dataset containing the MODIS band data, and it's quality control data for the same pixels. The first half rows are band data (EVI data from the MOD13Q1 vegetation indices product), and the second half are pixel reliability data. There are 5 rows for each data type. Each row contains 5 pixels (columns) at that time-step.

Usage

```
data(QualityCheckExample)
```

Format

A data frame with 10 observations on the following 5 variables.

pixel.1 Pixel 1.

pixel.2 Pixel 2.

pixel.3 Pixel 3.

pixel.4 Pixel 4.

pixel.5 Pixel 5.

Details

The data input to `QualityCheck` can be a vector or a matrix. It will return an object with the same dimensions. `QualityCheck` was designed for internal use within `MODISSummaries`, but can be used separately. When called via `MODISSummaries`, the data input and its quality information is expected to be contained in the same ascii files, downloaded using `MODISSubsets`.

Source

Arbitrary values within the valid range of EVI data and its pixel reliability information.

SubsetExample

Example dataset for MODISSubsets()

Description

A dataset consisting of a set of arbitrary locations with a corresponding set of arbitrary start and end time-series dates and IDs. This file can be used in the example for the MODIS subset function.

Usage

```
data(SubsetExample)
```

Format

A data frame with 1 observation on the following 5 variables.

lat A numeric vector; decimal degrees latitudes in WGS-1984 coordinate system.

long A numeric vector; decimal degrees longitudes in WGS-1984 coordinate system.

start.date A numeric vector; listing the date to begin the time-series for each corresponding location.

end.date A numeric vector; listing the date to end the time-series for each corresponding location.

ID A numeric vector; a unique ID code for each unique time-series (either unique in time or in space).

Details

Each time-series that has a unique combination of location/time-series dates should ideally have a unique ID. If IDs are not unique, then an arbitrary set of IDs will be assigned and the optional summary file should be written to maintain transparency in the data collection process.

Source

Locations were arbitrarily drawn from the grounds of Silwood Park Campus, Imperial College London.

See Also

[MODISsubsets](#)

TransectExample

Example dataset for MODISTransects()

Description

A dataset consisting of a set of arbitrary transect start/end locations with a corresponding set of arbitrary start and end time-series dates and transect IDs. This file can be used in the example for the MODIS subset function.

Usage

```
data(TransectExample)
```

Format

A data frame with 1 observation on the following 7 variables.

transect A numeric vector; a unique ID code for each transect.

start.lat A numeric vector; decimal degrees latitudes in WGS-1984 coordinate system for the transect start point.

start.long A numeric vector; decimal degrees longitudes in WGS-1984 coordinate system for the transect start point.

end.lat A numeric vector; decimal degrees latitudes in WGS-1984 coordinate system for the transect end point.

end.long A numeric vector; decimal degrees longitudes in WGS-1984 coordinate system for the transect end point.

start.date A numeric vector; listing the year to begin the time-series for each corresponding location.

end.date A numeric vector; listing the year to end the time-series for each corresponding location.

Source

Locations were arbitrarily drawn from the grounds of Silwood Park Campus, Imperial College London.

See Also

[MODISTransects](#)

UpdateSubsets

Updating Subset List

Description

A function that identifies, from a given dataset, which subsets have already been downloaded and returns a dataframe of those yet to be downloaded. A useful function if the list of locations for subsets is continually increasing or if download was halted previously.

Usage

```
UpdateSubsets(LoadDat, StartDate=FALSE, Dir=".")
```

Arguments

LoadDat	Input dataset: either the name of an object already in the workspace, or a file to be read in by specifying its file path as a character string, that has location data, end date (end.date) and study ID for each location.
StartDate	Logical: Specifying whether StartDate was specified in the original MODIS-Subset download.
Dir	String: Directory where previously downloaded subsets are saved. Default is current working directory.

Details

The input dataset should be organised accordingly: "lat" and "long" columns showing WGS-1984 decimal degrees latitudes and longitudes respectively; "end.date" for study end date.

Value

A dataframe containing any data for which a subset has not been downloaded. This can then be used with MODISSubsets.

Author(s)

Helen Phillips and Sean Tuck

See Also

[MODISSubsets](#)

Examples

```
## Not run:
# dontrun() used because running the example requires internet access.

data(SubsetExample, ConvertExample)

modis.subset <- ConvertToDD(XY = ConvertExample, LatColName = "lat", LongColName = "long")
modis.subset <- data.frame(lat = c(SubsetExample$lat, modis.subset[,1]),
                          long = c(SubsetExample$long, modis.subset[,2]),
                          start.date = rep(SubsetExample$start.date, 9),
                          end.date = rep(SubsetExample$end.date, 9))

MODISSubsets(LoadDat = SubsetExample, Product = "MOD13Q1", Bands = c("250m_16_days_EVI",
"250m_16_days_pixel_reliability"), Size = c(0,0), StartDate = TRUE)

Updated.modis.subset <- UpdateSubsets(LoadDat = modis.subset, StartDate = TRUE)

MODISSubsets(LoadDat = Updated.modis.subset, Product = "MOD13Q1", Bands = c("250m_16_days_EVI",
"250m_16_days_pixel_reliability"), Size = c(0,0), StartDate = TRUE)

## End(Not run)
```

Index

*Topic **datasets**

- ConvertExample, [2](#)
- EndCoordinatesExample, [5](#)
- FindIDExample, [8](#)
- QualityCheckExample, [28](#)
- SubsetExample, [29](#)
- TransectExample, [30](#)

BatchDownload, [2](#)

ConvertExample, [2](#)
ConvertToDD, [3](#), [3](#), [5](#)

EndCoordinates, [4](#)
EndCoordinatesExample, [5](#)
ExtractTile, [6](#), [16](#)

FindID, [7](#)
FindIDExample, [8](#)

GetBands, [9](#)
GetDates, [10](#)
GetProducts, [11](#)
GetSubset, [12](#)

LandCover, [14](#)

MODISGrid, [15](#), [18](#)
MODISSubsets, [4](#), [6](#), [9–14](#), [16](#), [17](#), [21](#), [23](#), [24](#),
[26](#), [30](#), [31](#)
MODISSummaries, [8](#), [14](#), [16](#), [18](#), [19](#), [23](#), [28](#)
MODISTimeSeries, [18](#), [22](#)
MODISTools, [23](#)
MODISTools–package (MODISTools), [23](#)
MODISTransects, [5](#), [18](#), [25](#), [31](#)

QualityCheck, [21](#), [27](#)
QualityCheckExample, [28](#)

SubsetExample, [29](#)

TransectExample, [30](#)

UpdateSubsets, [31](#)