

Package: rOPTRAM (via r-universe)

October 30, 2024

Title Derive Soil Moisture Using the OPTRAM Algorithm

Version 0.3.0.000

Description The Optical TRapezoid Model (OPTRAM) derives soil moisture based on the linear relation between a vegetation index and Land Surface Temperature (LST). The Short Wave Infra-red (SWIR) band is used as a proxy for LST. See: Sadeghi, M. et al., 2017. <<https://doi.org/10.1016/j.rse.2017.05.041>> .

License GPL (>= 3)

URL <https://github.com/ropensci/rOPTRAM.git>

BugReports <https://github.com/ropensci/rOPTRAM/issues>

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports dplyr, ggplot2, MASS, sf, terra, utils

Suggests geojsonio, lwgeom, knitr, rmarkdown, stats, testthat (>= 3.0.0), xml2, openeo, CDSE, jsonlite, qpdf

Config/testthat/edition 3

VignetteBuilder knitr

Spelling en-US

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

RemoteUrl <https://github.com/ropensci/rOPTRAM>

RemoteRef main

RemoteSha aa147a8be6a8809b49bc0151f4617c0a9f1261f9

Contents

acquire_openeo	2
acquire_scihub	4

calculate_str	6
calculate_vi	6
crop_landsat_list	7
exponential_coefficients	8
exponential_soil_moisture	9
linear_coefficients	10
linear_soil_moisture	11
optram	12
optram_acquire_s2	13
optram_calculate_soil_moisture	15
optram_calculate_str	17
optram_landsat	17
optram_ndvi_str	19
optram_options	20
optram_prepare_other_vi_str	21
optram_safe	22
optram_wetdry_coefficients	23
plot_vi_str_cloud	24
polynomial_coefficients	25
polynomial_soil_moisture	26
retrieve_cdse_credentials	27
store_cdse_credentials	27

Index	29
--------------	-----------

acquire_openeo	<i>Utility Function to Acquire Sentinel-2 Imagery using openEO</i>
----------------	--

Description

This non-exported function uses the `openeo` package to send a request to Copernicus DataSpace, and prepare the products. Called by `optram_acquire_s2`

Usage

```
acquire_openeo(
  aoi,
  from_date,
  to_date,
  output_dir = tempdir(),
  scale_factor = 10000
)
```

Arguments

aoi	sf object, POLYGON or MULTIPOLYGON of area of interest
from_date	string, represents start of date range, formatted as "YYYY-MM-DD"
to_date	string, end of date range, formatted as "YYYY-MM-DD"
output_dir	string, path to save downloaded, and processed imagery
scale_factor	integer, scaling factor for EO data source default 10000 , to scale Sentinel-2 15 bit DN to range (0, 1)

Value

list of BOA files

Note

This function utilizes the openeo package. Instructions for the login process: First of all, to authenticate your account on the backend of the Copernicus Data Space Ecosystem, it is necessary for you to complete the registration process. Follow these instructions for registration: <https://documentation.dataspace.copernicus.eu/Registration.html> After you have registered and installed the openeo package, you can run the acquire_openeo function. During the process of connecting to the server and logging in, you need to follow these steps:

1. When the message "Press to proceed:" appears in the console, press enter.
2. When prompted with 'Copy CTGB-UGFU and paste when requested by the browser' in the console, it may appear but can be ignored, as it is related to an older version of the openeo package.
3. Calling this method opens your system web browser, with which you can authenticate yourself on the back-end authentication system.
4. After that, the website will give you instructions to go back to the R client, where your connection has logged your account in. This means that every call that comes after that via the connection variable is executed by your user account.
5. You will be redirected to <https://identity.dataspace.copernicus.eu/>. Ensure you have an account and are logged in. You will be required to grant access - press "yes".

Examples

```
## Not run:
from_date <- "2018-12-01"
to_date <- "2019-04-30"
aoi <- sf::st_read(system.file("extdata",
                             "lachish.gpkg", package = 'roptram'))
Important! Assign the output to an object for later retrieval
acq <- acquire_openeo(aoi, from_date, to_date,
                     veg_index = "SAVI",
                     SWIR_band = 11)

## End(Not run)
```

`acquire_scihub`*Utility Function to Acquire Sentinel-2 Imagery Using CDSE Package*

Description

This function uses the CDSE package to send a request to Copernicus Dataspace, and prepare the products. Called by `optram_acquire_s2`

Usage

```
acquire_scihub(  
  aoi,  
  from_date,  
  to_date,  
  output_dir = tempdir(),  
  save_creds = TRUE,  
  clientid = NULL,  
  secret = NULL  
)
```

Arguments

<code>aoi</code>	sf object, POLYGON or MULTIPOLYGON of area of interest
<code>from_date</code>	string, represents start of date range, formatted as "YYYY-MM-DD"
<code>to_date</code>	string, end of date range, formatted as "YYYY-MM-DD"
<code>output_dir</code>	string, path to save downloaded, and processed imagery
<code>save_creds</code>	logical, whether to save CDSE credentials. Default TRUE.
<code>clientid</code>	string, user's OAuth client id. Required if <code>save_creds</code> is TRUE.
<code>secret</code>	string, user's OAuth secret. Required if <code>save_creds</code> is TRUE.

Value

list of STR files

Note

#* This function utilizes the CDSE package. Make sure to install the CDSE and jsonlite packages. Create OAuth account and token: Creating an Account:

1. Navigate to the <https://dataspace.copernicus.eu/>.
2. Click the "Register" button to access the account creation page.
3. If already registered, enter your username and password, and click "Login."
4. Once logged in, go to the User dashboard and click "User Settings" to access the Settings page.

Creating OAuth Client:

1. On the Settings page, click the green "Create New" button located on the right.
2. Enter a suitable "Client Name" and click the green "Create Client" button.
3. A Client secret is generated.

The user must save her secret and clientid somewhere. These credentials will be saved automatically to a standard filesystem location if the user calls `store_cdse_credentials()` with the `clientid` and `secret` parameters. If the user chooses not to save credentials to the standard filesystem location, then she will need to add both `clientid` and `secret` to each `acquire_scihub()` function call.

Using Credentials with `acquire_scihub`:

- If the credentials were stored using `store_cdse_credentials()`, the credentials are retrieved automatically.
- Otherwise, you can utilize the generated `clientid` and `secret` from <https://dataspace.copernicus.eu/> within the `acquire_scihub()` function.
- If you want to store your credentials on your computer, ensure that when running `acquire_scihub()`, the `save_creds` parameter is set to 'TRUE'.
- During the first run of `acquire_scihub()`, manually input your `clientid` and `secret` in the function signature. Subsequent runs will use the stored credentials.

Subject Area Constraint: The downloadable images are restricted to a maximum size of 2500 pixels on each side. This limitation is established due to the final resolution set to 10 meters using JavaScript. Consequently, the subject area available for download is limited to 25 kilometers in both directions. Please be aware of this restriction when selecting your desired area for download.

Area of Interest (AOI) Specification: When defining your Area of Interest (AOI), please ensure that it is represented as a polygonal layer with only one feature. This feature can either be a single POLYGON or a MULTIPOLYGON, which may consist of non-contiguous areas, but only one feature is permissible.

Examples

```
## Not run:
from_date <- "2018-12-01"
to_date <- "2019-04-30"
aoi <- sf::st_read(system.file("extdata",
                             "lachish.gpkg", package = 'roptram'))
Make sure the AOI is a MULTIPOLYGON
aoi <- sf::st_as_sf(sf::st_union(aoi))
optram_options("veg_index", "NDVI", show_opts = FALSE)
optram_options("SWIR_band", 11)
acquire_scihub(aoi, from_date, to_date,
               clientid = '...enter clientid here...',
               secret = '...enter secret here...')

## End(Not run)
```

calculate_str *Calculate STR from SWIR Bottom of Atmosphere Band*

Description

Use this function to prepare STR from SAFE imagery when you have already downloaded Sentinel 2 image files in advance

Usage

```
calculate_str(img_stk, scale_factor = 10000)
```

Arguments

img_stk	terra SpatRaster, multiband stack of images, already clipped to aoi
scale_factor	integer, scaling factor for EO data source default 10000, to scale Sentinel-2 15 bit DN to range (0, 1)

Value

STR, SpatRaster of STR band

Note

For Landsat images, scale_factor should be 1, since Landsat metadata contains gain and offset for scaling image bands.

Examples

```
img_stk <- terra::rast(system.file("extdata", "BOA",  
                                  "BOA_2022-12-11_T36RXV.tif", package = "ropTRAM"))  
STR_dir = tempdir()  
str <- calculate_str(img_stk)
```

calculate_vi *Calculate Vegetation Index from Bottom of Atmosphere Image Bands*

Description

Use this function to prepare vegetation index from SAFE imagery when you have already downloaded Sentinel 2 image files in advance

Usage

```
calculate_vi(
  img_stk,
  redband = 4,
  greenband = 3,
  blueband = 2,
  nirband = 5,
  scale_factor = 2^15
)
```

Arguments

img_stk	terra SpatRaster, multiband stack of images, already clipped to aoi
redband	integer, number of red band
greenband	integer, number of green band
blueband	integer, number of blue band
nirband	integer, number of NIR band
scale_factor	numeric, scales Sentinel-2 15 bit values down to range (0,255) as expected for vegetation indices default 32768 (2^15)

Value

vi_rast, SpatRaster of vegetation index

Note

The scale_factor parameter reduces numeric range "digital number", (DN) of Sentinel-2 images from 15 bit integer to 8 bit. Vegetation indices such as SAVI expect values in the range (0, 255). This scale_factor produces values in that range. When using Landsat 8 images (DN already in range of (0, 255)) then set scale_factor to 255.

Examples

```
img_stk <- terra::rast(system.file("extdata", "BOA",
  "BOA_2023-01-20_T36RXV.tif", package = "ropTRAM"))
vi <- calculate_vi(img_stk)
```

crop_landsat_list *Crop List of Landsat Bands to AOI*

Description

Utility function to prepare BOA bands cropped to Area of Interest.

Usage

```
crop_landsat_list(landsat_dir)
```

Arguments

landsat_dir string, directory containing the original downloaded Landsat imagery folders.

Value

cropped_list, list, paths to derived BOA stacks, cropped to study area

Examples

```
## Not run:  
cropped_landsat_list <- crop_landsat_list(landsat_dir)  
  
## End(Not run)
```

exponential_coefficients

Utility Function to Prepare Exponential Fitted Edges of Trapezoid

Description

Called by [optram_wetdry_coefficients](#) to prepare exponential curve along trapezoid edges. Calculates the intercept and slope of both wet and dry edges and updates the edges data.frame with these exp fitted values. Not exported.

Usage

```
exponential_coefficients(df, output_dir)
```

Arguments

df data.frame, values of VI and STR along edges of trapezoid
output_dir string, path to save coefficients CSV file

Value

df, data.frame, the trapezoid line edge points with fitted wet/dry values added

Note

Three CSV files are saved:

- the regressions coefficients,
- the trapezoid edge points
- RMSE of the fitted curve

Examples

```
## Not run:
df <- read.csv(system.file("extdata", "trapezoid_edges.csv",
                          package = "rOPTRAM"))
output_dir <- tempdir()
coeffs <- rOPTRAM::exponential_coefficients(df, output_dir)
coeffs

## End(Not run)
```

exponential_soil_moisture

Calculate Soil Moisture Grid, Exponential Trapezoid

Description

Prepare soil moisture grid from STR and VI images for a single date, based on exponential function to derive trapezoid edges, using the derived slope and intercept coefficients

Usage

```
exponential_soil_moisture(coeffs, VI, STR)
```

Arguments

coeffs	list, 4 trapezoid coefficients
VI	terra rast, the vegetation index raster
STR	terra rast, the STR raster

Value

rast, soil moisture grid

Note

This function is used after preparing the OPTRAM model coefficients with: [optram_wetdry_coefficients](#). Typically a new image date, (that was not used for preparing the model), will be referenced in the `img_date` parameter. The resulting soil moisture raster is saved to `output_dir`. This function implements an exponential trapezoid, following: Ambrosone, Mariapaola, et al. 2020. "Retrieving Soil Moisture in Rainfed and Irrigated Fields Using Sentinel-2 Observations and a Modified OPTRAM Approach." International Journal of Applied Earth Observation and Geoinformation 89 (July): 102113. [doi:10.1016/j.jag.2020.102113](https://doi.org/10.1016/j.jag.2020.102113)

Examples

```
img_date <- "2023-03-11"
VI_dir <- system.file("extdata", "NDVI", package = "rOPTRAM")
STR_dir <- system.file("extdata", "STR", package = "rOPTRAM")
SM <- optram_calculate_soil_moisture(img_date,
                                   VI_dir, STR_dir,
                                   data_dir = tempdir())
```

linear_coefficients *Utility Function to Prepare Linear Regression Edges of Trapezoid*

Description

Called by [optram_wetdry_coefficients](#) to prepare linear regression line along trapezoid edges
Calculates the intercept and slope of both wet and dry edges Not exported

Usage

```
linear_coefficients(df, output_dir)
```

Arguments

df data.frame, values of VI and STR along edges of trapezoid
output_dir string, path to save coefficients CSV file

Value

df, data.frame, the trapezoid line edge points with fitted wet/dry values added

Note

Three CSV files are saved:

- the regressions coefficients,
- the trapezoid edge points
- RMSE of the fitted curve

Examples

```
## Not run:
edges_file <- system.file("extdata/trapezoid_edges.csv",
                          package = "rOPTRAM")
df <- read.csv(edges_file)
output_dir <- tempdir()
coeffs <- linear_coefficients(df, output_dir)
coeffs

## End(Not run)
```

linear_soil_moisture *Calculate Soil Moisture Grid, Linear Trapezoid*

Description

Prepare soil moisture grid from STR and VI images for a single date, using linear trapezoid method, and using the derived slope and intercept coefficients

Usage

```
linear_soil_moisture(coeffs, VI, STR)
```

Arguments

coeffs	list, 4 trapezoid coefficients
VI	SpatRaster, the vegetation index raster
STR	SpatRaster, the STR raster

Value

rast, soil moisture grid

Note

This function is used after preparing the OPTRAM model coefficients with: [optram_wetdry_coefficients](#). Typically a new image date, (that was not used for preparing the model), will be referenced in the `img_date` parameter. The resulting soil moisture raster is saved to `output_dir`.

Examples

```
img_date <- "2023-03-11"  
VI_dir <- system.file("extdata", "NDVI", package = "rOPTRAM")  
STR_dir <- system.file("extdata", "STR", package = "rOPTRAM")  
SM <- optram_calculate_soil_moisture(img_date,  
  VI_dir, STR_dir,  
  data_dir = tempdir())
```

optram	<i>Prepare Sentinel Imagery for Soil Moisture Prediction Using OP-TRAM.</i>
--------	---

Description

The main wrapper function to download, and preprocess Sentinel 2 imagery then prepare the OP-TRAM trapezoid to derive slope and intercept for wet and dry trapezoid lines. See: Sadeghi, M., Babaeian, E., Tuller, M., Jones, S.B., 2017. The optical trapezoid model: A novel approach to remote sensing of soil moisture applied to Sentinel-2 and Landsat-8 observations. Remote Sensing of Environment 198, 52–68, doi:10.1016/j.rse.2017.05.041

Usage

```
optram(
  aoi,
  from_date,
  to_date,
  S2_output_dir = tempdir(),
  data_output_dir = tempdir()
)
```

Arguments

aoi	sf object, a POLYGON or MULTIPOLYGON, boundary of area of interest
from_date	string, the start of the date range, Formatted as "YYYY-MM-DD"
to_date	the end of the date range.
S2_output_dir	string, directory to save downloaded S2 and the derived products, defaults to tempdir()
data_output_dir	string, path to save coeffs_file and STR-VI data.frame, default is tempdir()

Value

rmse_df, data.frame, RMSE values of fitted trapezoid lines the coefficients are also saved to a csv file in data_output_dir.

Note

- Sentinel downloaded products are saved to S2_output_dir. Data files (Trapezoid coefficients and STR-VI data) to data_output_dir
- Products can be downloaded covering two "period" options: either "full" - all available images from from_date to to_date or "seasonal" - available images for all years but only for the months between day/month of from_date to the day/month of to_date. Use [optram_options](#) to set this option.

- Three trapezoid fitting methods are implemented: "linear", "exponential" and "polynomial". See the `optram_options` function for details
- Two SWIR wavelength bands are available in Sentinel-2: 1610 nanometer (nm) and 2190 nm. The option `SWIR_bands` can be set in `optram_options` to choose which band is used in this model.
- Several vegetation indices are implemented: "NDVI", "SAVI", etc. The `optram_options` function also sets this option.

Examples

```
## Not run:
from_date <- "2018-12-01"
to_date <- "2020-04-30"
aoi <- sf::st_read(system.file("extdata",
                              "lachish.gpkg", package = "roptram"))
rmse <- optram(aoi,
               from_date, to_date)

## End(Not run)
```

optram_acquire_s2 *Acquire Sentinel 2 Images at a Given Location and Date Range*

Description

Use the CDSE package to acquire, preprocess and crop Sentinel 2 satellite imagery.

Usage

```
optram_acquire_s2(
  aoi,
  from_date,
  to_date,
  output_dir = tempdir(),
  scale_factor = 10000,
  save_creds = TRUE,
  clientid = NULL,
  secret = NULL
)
```

Arguments

<code>aoi</code>	sf POLYGON or MULTIPOLYGON, of boundary of area of interest
<code>from_date</code>	string, represents start of date range, formatted as "YYYY-MM-DD"
<code>to_date</code>	string, end of date range, formatted as "YYYY-MM-DD"
<code>output_dir</code>	string, path to save processed imagery.

scale_factor	numeric, scale factor for reflectance values. Default 10000.
save_creds	logical, whether to save CDSE credentials. Default TRUE.
clientid	string, user's OAuth client id. Required if save_creds is TRUE.
secret	string, user's OAuth secret. Required if save_creds is TRUE.

Value

output_paths, strings, list of downloaded (STR) files

Note

This wrapper function calls one of multiple download functions, each accessing a different cloud-based resource. The cloud based resource can be one of "scihub" or "openeo":

"scihub",... If "scihub" then: This function utilizes the CDSE package. Make sure to install the CDSE and jsonlite packages. Create OAuth account and token: Creating an Account:

1. Navigate to the [Copernicus portal](#).
2. Click the "Register" button to access the account creation page.
3. If already registered, enter your username and password, and click "Login."
4. Once logged in, go to the User dashboard and click "User Settings" to access the Settings page.

Creating OAuth Client:

1. On the Settings page, click the green "Create New" button located on the right.
2. Enter a suitable "Client Name" and click the green "Create Client" button.
3. A Client secret is generated.

the user must save her secret and clientid somewhere. these credentials will be saved automatically to a standard filesystem location if the user calls check_scihub() with the argument save_creds set to TRUE (recommended). if the user chooses not to save credentials to the standard filesystem location, then she will need to add both clientid and secret to each acquire_scihub() function call.

Using Credentials with acquire_scihub():

- Now, you can utilize the generated clientid and secret in the acquire_scihub() function.
- If you want to store your credentials on your computer, ensure that when running acquire_scihub, the save_creds parameter is set to TRUE.
- During the first run of acquire_scihub(), manually input your clientid and secret in the function's signature.
- Subsequent runs will use the stored credentials.

Using Credentials with acquire_scihub():

- Now, you can utilize the generated clientid and secret in the acquire_scihub() function.
- If you want to store your credentials on your computer, ensure that when running acquire_scihub, the save_creds parameter is set to TRUE.

- During the first run of `acquire_scihub()`, manually input your `clientid` and `secret` in the function's signature.
- Subsequent runs will use the stored credentials.

Subject Area Constraint: The downloadable images are restricted to a maximum size of 2500 pixels on each side. This limitation is established due to the final resolution set to 10 meters using JavaScript. Consequently, the subject area available for download is limited to 25 kilometers in both directions. Please be aware of this restriction when selecting your desired area for download.

Area of Interest (AOI) Specification: When defining your Area of Interest (AOI), please ensure that it is represented as a POLYGON or MULTIPOLYGON layer.

"openeo",... If "openeo" then: This function utilizes the `openeo` package. Instructions for the login process: First of all, to authenticate your account on the backend of the Copernicus Data Space Ecosystem, it is necessary for you to complete the registration process. Follow these instructions for registration: <https://documentation.dataspace.copernicus.eu/Registration.html> After you have registered and installed the `openeo` package, you can run the `acquire_openeo()` function. During the process of connecting to the server and logging in, you need to follow these steps:

1. When the message "Press to proceed:" appears in the console, press enter.
2. Calling this method opens your system web browser, with which you can authenticate yourself on the back-end authentication system. After that, the website will give you instructions to go back to the R client, where your connection has logged your account in. This means that every call that comes after that via the connection variable is executed by your user account.
3. You will be redirected to <https://identity.dataspace.copernicus.eu/>. Ensure you have an account and are logged in. You will be required to grant access - press "yes".

Two SWIR bands are available in Sentinel-2: 1610 nanometer (nm) and 2190 nm. Setting the option `SWIR_bands` with `optram_options` allows to choose which band is used in this model.

Examples

```
## Not run:
from_date <- "2018-12-01"
to_date <- "2019-04-30"
aoi <- sf::st_read(system.file("extdata",
                             "lachish.gpkg", package = 'roptram'))
aoi <- sf::st_as_sf(sf::st_union(aoi))
s2_file_list <- optram_acquire_s2(aoi,
                                 from_date, to_date)

## End(Not run)
```

```
optram_calculate_soil_moisture
      Calculate Soil Moisture Grid
```

Description

Prepare soil moisture grid from STR and VI images for a single date using the derived slope and intercept coefficients

Usage

```
optram_calculate_soil_moisture(
  img_date,
  VI_dir,
  STR_dir,
  data_dir,
  output_dir = tempdir()
)
```

Arguments

img_date	string, image date of single Sentinel 2 acquisition formatted as "YYYY-MM-DD"
VI_dir	string, full path to directory holding the VI rasters
STR_dir	string, full path to directory holding the STR rasters
data_dir	string, the directory where coefficients file was saved (the output_dir parameter in optram_wetdry_coefficients function)
output_dir	string, full path to output directory for saving soil moisture raster

Value

SpatRaster, raster of soil moisture, file(s) saved in output_dir

Note

This function is used after preparing the OPTRAM model coefficients with: [optram_wetdry_coefficients](#). Typically a new image date, (possibly an image that was not used for preparing the model), will be referenced in the img_date parameter. The resulting soil moisture raster is saved to output_dir.

Three trapezoid models are offered through the trapezoid_method option: either "linear", "exponential", or "polynomial". (set using [optram_options](#)) The data_dir parameter is a directory name. The coefficients CSV file that matches trapezoid_method should be in that directory

For further details see: Ambrosone, Mariapaola, et al. 2020. "Retrieving Soil Moisture in Rainfed and Irrigated Fields Using Sentinel-2 Observations and a Modified OPTRAM Approach." *International Journal of Applied Earth Observation and Geoinformation* 89 (July): 102113. [doi:10.1016/j.jag.2020.102113](https://doi.org/10.1016/j.jag.2020.102113).

The data_dir directory must contain the coefficients CSV file i.e. for "linear" method the file was saved as 'coefficients_lin.csv' for "exponential" it was saved as 'coefficients_exp.csv' for "polynomial" it was saved as 'coefficients_poly.csv'

Examples

```
img_date <- "2023-03-11"
VI_dir <- system.file("extdata", "NDVI", package = "rOPTRAM")
STR_dir <- system.file("extdata", "STR", package = "rOPTRAM")
data_dir <- system.file("extdata")
SM <- optram_calculate_soil_moisture(img_date,
  VI_dir, STR_dir, data_dir)
```

optram_calculate_str *Create SWIR Transformed Reflectance*

Description

Create SWIR Transformed Reflectance

Usage

```
optram_calculate_str(BOA_dir, STR_dir = NULL)
```

Arguments

BOA_dir string, the path to the Bottom of Atmosphere bands
STR_dir string, output directory for STR rasters, Default is NULL, in which case, the STR_dir will be created alongside BOA_dir

Value

list of string, the path to transformed raster

Note

This function follows: Sadeghi, M., Babaeian, E., Tuller, M., Jones, S.B., 2017. The optical trapezoid model: A novel approach to remote sensing of soil moisture applied to Sentinel-2 and Landsat-8 observations. *Remote Sensing of Environment* 198, 52–68. [doi:10.1016/j.rse.2017.05.041](https://doi.org/10.1016/j.rse.2017.05.041)

SWIR Transformed Reflectance is calculated as $STR = (1 - SWIR)^2 / 2 * SWIR$ SWIR is band 12 (2190 nm) or 11 (1610 nm)

Examples

```
BOA_dir <- system.file("extdata", "BOA", package = "roptram")  
STR_dir = tempdir()  
STR <- optram_calculate_str(BOA_dir, STR_dir)
```

optram_landsat *Handle Satellite Imagery in Original Landsat Format*

Description

Use this function to prepare vegetation index and SWIR Transformed Reflectance (STR) rasters when you have already downloaded Landsat image files in advance. This function assumes that atmospheric correction has been applied.

Usage

```
optram_landsat(
  landsat_dir,
  aoi,
  LC_output_dir = tempdir(),
  data_output_dir = tempdir()
)
```

Arguments

`landsat_dir` string, full path to containing folder of downloaded (unzipped) Landsat data in original landsat format, after atmospheric correction (L2A)

`aoi` sf object, POLYGON or MULTIPOLYGON of AOI boundary of area of interest

`LC_output_dir` string, directory to save the derived products, defaults to tempdir()

`data_output_dir` string, path to save coeffs_file and STR-VI data.frame, default is tempdir()

Value

rmse_df, data.frame, RMSE values of fitted trapezoid edges

Note

Unlike the `optram_acquire_s2` function, there is no implementation for automatic download of Landsat images. This function requires a directory, set in the `landsat_dir` parameter, which contains the set of Landsat tiles downloaded manually by the user, in advance. This directory should contain folders of Landsat images, where each folder consists of the individual Landsat bands as Geotiff files, as well as the metadata files as downloaded from, i.e. the USGS EarthExplorer <https://earthexplorer.usgs.gov/> website.

Examples

```
## Not run:
aoi <- sf::st_read(system.file("extdata",
                              "lachish.gpkg", package = "roptram"))
# landsat_dir is a directory containing the original downloaded Landsat images.
landsat_dir <- "...enter full path here..."
optram_landsat(landsat_dir, aoi,
              veg_index = 'SAVI',
              LC_output_dir = tempdir(), data_output_dir = tempdir())

## End(Not run)
```

optram_ndvi_str	<i>Prepare Dataframe with Pairs of NDVI and STR Values</i>
-----------------	--

Description

Collect all pixel values of both vegetation index, and Swir Transformed Reflectance, for a time series of images. Prepare data.frame of all pairs of values (as input for the [optram_wetdry_coefficients](#) function)

Usage

```
optram_ndvi_str(STR_list, VI_list, output_dir = tempdir(), aoi = NULL)
```

Arguments

STR_list	list of paths to STR raster files
VI_list	list of paths to NDVI raster files
output_dir	string, path to save data.frames (in RDS format)
aoi	sf POLYGON or MULTIPOLYGON, must have a numeric column named "ID" for coloring trapezoid points by features Default NULL, (no coloring)

Value

full_df, data.frame with 7 columns: X,Y,Date,Month,NDVI,STR,Density and optionally a 7th column with feature ID values from the AOI polygon. The columns Month, Density, Feature_ID can be used in [plot_vi_str_cloud\(\)](#) function to color the points in the scatter plot in various ways.

Note

Use the option `max_tbl_size` (see [optram_options](#)) to limit size of the NDVI-STR data.frame. With a large area of interest, and long time frame, the number of data points can overrun the computation resources. This parameter sets a total size of data.frame from the `max_tbl_size` parameter, together with the number of image time slots in the time range.

In some cases (i.e. water surfaces) NDVI can have values below zero. These pixels can be removed from the trapezoid by setting `rm.low.vi` option to TRUE.

The vegetation index column is named "NDVI" even though it can represent other vegetation indices, such as SAVI, or MSAVI.

Examples

```
VI_list <- list.files(system.file("extdata", "NDVI", package = "rOPTRAM"),
  pattern = ".tif$", full.names = TRUE)
STR_list <- list.files(system.file("extdata", "STR", package = "rOPTRAM"),
  pattern = ".tif$", full.names = TRUE)
full_df <- optram_ndvi_str(STR_list, VI_list)
# Show structure of output data.frame
str(full_df)
```

optram_options *Display or Set Package Options*

Description

rOPTRAM uses several package options. This function displays the current defined options, (showing the default options when the package is first loaded), and allows users to set new values for each option.

Usage

```
optram_options(opt_name = NULL, opt_value = NULL, show_opts = TRUE)
```

Arguments

opt_name string, one of the package options. Default NULL
 opt_value string, numeric, or boolean, depending on opt_name.
 show_opts boolean, default TRUE, prints a list of all optram options Default NULL

Note

When no new option name or value is specified, a list of currently defined options is printed.

rOPTRAM defines the following options at startup

opt_name	default	other possible values
veg_index	"NDVI"	"SAVI", "MSAVI"
remote	"scihub"	"openeo"
period	"full"	"seasonal"
max_cloud	12	between 0 and 100
vi_step	0.005	usually between 0.01 and 0.001
trapezoid_method	"linear"	"polynomial" or "exponential"
SWIR_band	11	12
max_tbl_size	1e+6	depends on computer resources
rm.low.vi	FALSE	TRUE
rm.hi.str	FALSE	TRUE
plot_colors	"no"	"no" = uniform green color for all points "features" = points colored by aoi features "density" = points colored by point density "contours" = plots density contour lines "months" = points colored by month of image date
feature_col	"ID"	string, name of <i>numeric</i> column that contains feature ID's for coloring plot
edge_points	TRUE	FALSE, whether to add the trapezoid edge points to the plot
only_vi_str	FALSE	TRUE (avoids downloading all Sentinel bands)

tileid	NA	a string of 5 characters, download only the requested tileid
scm_mask	TRUE	FALSE, whether to mask out: cloud shadows, clouds, water, snow, using the Copernicus SCM product.
overwrite	FALSE	Set to TRUE to re-download previously acquired images

Examples

```
opts <- options()
optram_options() # prints out list of current options
optram_options("SWIR_band", 12)
optram_options("veg_index", "SAVI")
optram_options("trapezoid_list", "exp") # fails
options(opts)
```

```
optram_prepare_other_vi_str
```

Handle Third Party Imagery With Red, NIR and SWIR Bands

Description

This function prepares vegetation index and STR rasters from other (non-Sentinel) image files downloaded in advance.

Usage

```
optram_prepare_other_vi_str(
  img_dir,
  aoi,
  viname = "NDVI",
  output_dir = tempdir()
)
```

Arguments

img_dir	string, full path directory of downloaded images (assumes Geotiff)
aoi	string, full path to polygon spatial file of area of interest
viname	string, which VI to prepare, 'NDVI', 'SAVI', etc.
output_dir	string, where to save Geotiff, default is tempdir()

Value

output_files, list, full paths to saved Geotiff files (not exported yet)

optram_safe

*Handle Sentinel Imagery in Original Copernicus SAFE Format***Description**

Use this function to prepares vegetation index and SWIR Transformed Reflectance (STR) rasters when you have already downloaded Sentinel 2 image files in advance. Unzip the downloaded Sentinel 2 files and do not change the folder structure. This function assumes that atmospheric correction has been applied. i.e. by downloading Level L2A product or using the SNAP L2A_Process,

Usage

```
optram_safe(
  safe_dir,
  aoi,
  S2_output_dir = tempdir(),
  overwrite = TRUE,
  data_output_dir = tempdir()
)
```

Arguments

safe_dir	string, full path to containing folder of downloaded (unzipped) Sentinel 2 data in original SAFE format, after atmospheric correction (L2A)
aoi	sf object, a POLYGON or MULTIPOLYGON of the AOI boundary
S2_output_dir	string, directory to save the derived products, defaults to tempdir()
overwrite	boolean, overwrite derived products that already were created, defaults to TRUE
data_output_dir	string, path to save coeffs_file and STR-VI data.frame, default is tempdir()

Value

rmse_df, data.frame, RMSE values of fitted trapezoid lines

Note

Use the max_tbl_size parameter to limit the total number of rows in the VI-STR data.frame. When the area of interest is large, or the time range of datasets is long, the total size of the data.frame can grow beyond the capacity of computation resources. This parameter limits the size of the table by sampling a number of data points from each time slot. The sample size is determined based on max_tbl_size and the total number of time slots in the full time range.

Two SWIR bands are available in Sentinel-2: 1610 nanometer (nm) and 2190 nm. Set which band to use with optram_options.

Examples

```
## Not run:
aoi <- sf::st_read(system.file("extdata",
                              "lachish.gpkg", package = "roptram"))
safe_dir <- "...enter directory containing downloaded SAFE folders..."
rmse <- optram_safe(safe_dir, aoi_file)

## End(Not run)
```

optram_wetdry_coefficients

Derive Coefficients of Slope and Intercept

Description

Derive slope and intercept coefficients for both wet and dry trapezoid lines. Write coefficients to a CSV file (as input to [optram_calculate_soil_moisture](#) function)

Usage

```
optram_wetdry_coefficients(full_df, output_dir = tempdir())
```

Arguments

full_df	data.frame of STR and NDVI values
output_dir	string, directory to save coefficients CSV file

Value

rmse_df, data.frame, RMSE values of fitted trapezoid edges

Note

The vegetation index column is named "VI" though it can represent several vegetation indices, such as SAVI, or MSAVI.

The trapezoid method option (see [optram_options](#)) allows to choose one of three models for creating the edge coefficients of the trapezoid.

- "linear" prepares a simple OLS regression line along the wet and dry edges of the trapezoid. Four coefficients are returned: intercept and slope for both edges.
- "exponential" creates an exponential curve fitted to the intercept and slope, following: Ambrosone, Mariapaola, Alessandro Matese, et al. 2020. "Retrieving Soil Moisture in Rainfed and Irrigated Fields Using Sentinel-2 Observations and a Modified OPTRAM Approach." International Journal of Applied Earth Observation and Geoinformation [doi:10.1016/j.jag.2020.102113](https://doi.org/10.1016/j.jag.2020.102113).

- "polynomial" fits a second order polynomial curve to the wet and dry edges of the trapezoid, following: Ma, Chunfeng, Kasper Johansen, and Matthew F. McCabe. 2022. "Combining Sentinel-2 Data with an Optical-Trapezoid Approach to Infer within-Field Soil Moisture Variability and Monitor Agricultural Production Stages." *Agricultural Water Management* 274 (December): 107942. doi:10.1016/j.agwat.2022.107942 This curve fitting function returns six coefficients: alpha, beta_1, and beta_2 for both wet and dry edges

Examples

```
full_df <- readRDS(system.file("extdata", "VI_STR_data.rds",
  package = "rOPTRAM"))
rmse_df <- optram_wetdry_coefficients(full_df, tempdir())
print(rmse_df)
optram_options("trapezoid_method", "polynomial")
rmse_df <- optram_wetdry_coefficients(full_df, tempdir())
print(rmse_df)
```

plot_vi_str_cloud *Create Scatter Plot of STR-NDVI Data Points,*

Description

Plot STR-NDVI scatterplot to show dry and wet trapezoid lines over scatterplot of multi-temporal STR and NDVI pixel values

Usage

```
plot_vi_str_cloud(full_df, edges_df, edge_points = FALSE)
```

Arguments

full_df	data.frame of NDVI and STR pixel values
edges_df	data.frame, points along the wet/dry edges for trapezoid
edge_points	boolean, whether to add to the plot the linear regression points that were used to derive coefficients. default FALSE

Value

ggplot object

Note

Points in scatter plot can be colored in various ways, depending on the plot_colors option, as set in optram_options(). If "plot_colors = none" all points are colored green. If "plot_colors = density" points are colored by point density on the plot. If "plot_colors = contour" points are colored green and density contour lines are overlaid. If "plot_colors = feature" points are colored by some feature ID in the original AOI polygon. If "plot_colors = month" points are colored by the month of image acquisition.

Examples

```
aoi_name <- "Soil Moisture AOI"
optram_options("trapezoid_method", "polynomial")
full_df <- readRDS(system.file("extdata", "VI_STR_data.rds",
                             package = "rOPTRAM"))
edges_df <- read.csv(system.file("extdata", "trapezoid_edges_lin.csv",
                               package = "rOPTRAM"))
pl <- plot_vi_str_cloud(full_df, edges_df)
pl + ggplot2::ggtitle(paste("Trapezoid plot for:", aoi_name))
pl
```

polynomial_coefficients

Utility Function to Prepare Polynomial Fitted Edges of Trapezoid

Description

Called by [optram_wetdry_coefficients](#) to prepare second order polynomial curve along trapezoid edges. Calculates six coefficients: intercept (alpha) of both wet and dry edges and first and second order coefficients (beta), as in $STR = \alpha + \beta_1 * VI + \beta_2 * VI^2$ and updates the edges data.frame with these polynomial values fitted values.

Usage

```
polynomial_coefficients(df, output_dir)
```

Arguments

df	data.frame, values of VI and STR along edges of trapezoid
output_dir	string, path to save coefficients CSV file

Value

df, data.frame, the trapezoid line edge points with fitted values added for both wet and dry edges

Note

Three CSV files are saved:

- the regressions coefficients,
- the trapezoid edge points
- RMSE of the fitted curve

Examples

```
## Not run:
df <- read.csv(system.file("extdata", "trapezoid_edges.csv",
                           package = "rOPTRAM"))
output_dir <- tempdir()
coeffs <- rOPTRAM::polynomial_coefficients(df, output_dir)
coeffs

## End(Not run)
```

polynomial_soil_moisture

Calculate Soil Moisture, Polynomial Fitted Curve

Description

Prepare soil moisture grid from STR and VI images for a single date, based on polynomial function fitted to trapezoid edges.

Usage

```
polynomial_soil_moisture(coeffs, VI, STR)
```

Arguments

coeffs	list, 6 trapezoid coefficients
VI	terra rast, the vegetation index raster
STR	terra rast, the STR raster

Value

rast, soil moisture grid

Note

This function is used after preparing the OPTRAM model coefficients with: [optram_wetdry_coefficients](#). Typically a new image date, (that was not used for preparing the model), will be referenced in the `img_date` parameter. The resulting soil moisture raster is saved to `output_dir`. This function implements a polynomial fitted curve, following: Ma, Chunfeng, Kasper Johansen, and Matthew F. McCabe. 2022. "Combining Sentinel-2 Data with an Optical-Trapezoid Approach to Infer within-Field Soil Moisture Variability and Monitor Agricultural Production Stages." *Agricultural Water Management* 274 (December): 107942. doi:10.1016/j.agwat.2022.107942.

Examples

```
img_date <- "2023-03-11"
VI_dir <- system.file("extdata", "NDVI", package = "rOPTRAM")
STR_dir <- system.file("extdata", "STR", package = "rOPTRAM")
optram_options("trapezoid_method", "polynomial")
SM <- optram_calculate_soil_moisture(img_date,
                                     VI_dir, STR_dir,
                                     data_dir = tempdir())
```

retrieve_cdse_credentials

Retrieve CDSE Client Credentials from File

Description

Retrieve CDSE clientid and secret from file The file location is system specific. It would have been setup in advance using the [store_cdse_credentials](#) function

Usage

```
retrieve_cdse_credentials()
```

Value

A data frame containing the retrieved CDSE clientid and secret, or NULL if credentials are not available.

store_cdse_credentials

Store CDSE Client Credentials

Description

Store CDSE clientid and secret into a file The file location is system specific. Users who chose to save CDSE credentials can use this function (and then the [retrieve_cdse_credentials](#) afterwards) The clientid and secret are obtained from:

Usage

```
store_cdse_credentials(clientid = NULL, secret = NULL)
```

Arguments

clientid	string, user's OAuth client id
secret	string, user's OAuth secret

Note

Both `clientid` and `secret` can alternatively be supplied as environment variables: `OAUTH_CLIENTID` and `OAUTH_SECRET`. If these env variables are available (and no values are entered as function arguments) they will be used to store credentials.

Examples

```
## Not run:  
store_cdse_credentials(clientid="<...enter your client id...>",  
                      secret = "<...enter your secret...>")  
  
## End(Not run)
```

Index

acquire_openeo, [2](#)
acquire_scihub, [4](#)

calculate_str, [6](#)
calculate_vi, [6](#)
crop_landsat_list, [7](#)

exponential_coefficients, [8](#)
exponential_soil_moisture, [9](#)

linear_coefficients, [10](#)
linear_soil_moisture, [11](#)

optram, [12](#)
optram_acquire_s2, [2](#), [4](#), [13](#), [18](#)
optram_calculate_soil_moisture, [15](#), [23](#)
optram_calculate_str, [17](#)
optram_landsat, [17](#)
optram_ndvi_str, [19](#)
optram_options, [12](#), [13](#), [15](#), [16](#), [19](#), [20](#), [23](#)
optram_prepare_other_vi_str, [21](#)
optram_safe, [22](#)
optram_wetdry_coefficients, [8–11](#), [16](#), [19](#),
[23](#), [25](#), [26](#)

plot_vi_str_cloud, [24](#)
polynomial_coefficients, [25](#)
polynomial_soil_moisture, [26](#)

retrieve_cdse_credentials, [27](#), [27](#)

store_cdse_credentials, [27](#), [27](#)