Package 'evapoRe'

April 10, 2025

Title Evapotranspiration R Recipes

Version 1.0.1

Description
An R-based application for exploratory data analysis of global EvapoTranspiration (ET) datasets. 'evapoRe' enables users to download, validate, visualize, and analyze multi-source ET data across various spatio-temporal scales.
Also, the package offers calculation methods for estimating potential ET (PET), including temperature-based, combined type, and radiation-based approaches described in : Oudin et al., (2005) <doi:10.1016/j.jhydrol.2004.08.026>.
'evapoRe' supports hydrological modeling, climate studies, agricultural research, and other data-driven fields by facilitating access to ET data and offering powerful analysis capabilities.
Users can seamlessly integrate the package into their research applications and explore diverse ET data at different resolutions.

Depends R (>= 4.0.0)

Imports methods, parallel, utils, data.table, doParallel, foreach, lubridate, raster,twc

License GPL-3

Encoding UTF-8

LazyData true

URL https://github.com/AkbarR1184/evapoRe

BugReports https://github.com/AkbarR1184/evapoRe/issues

SystemRequirements PROJ (>= 6, https://proj.org/download.html), GDAL (>= 3, https://gdal.org/download.html), NetCDF (>= 4, https://www.unidata.ucar.edu/software/netcdf/).

RoxygenNote 7.3.2

Suggests rmarkdown, ggpubr, knitr, spelling, kableExtra, tibble, testthat (>= 3.0.0)

Language en-US

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Akbar Rahmati Ziveh [aut, cre]
(<https://orcid.org/0000-0003-4854-451X>),
Mijael Rodrigo Vargas Godoy [aut]
(<https://orcid.org/0000-0002-1828-9266>),
Vishal Thakur [ctb] (<https://orcid.org/0000-0003-2864-9517>),
Yannis Markonis [aut, ths] (<https://orcid.org/0000-0003-0144-8969>)

Maintainer Akbar Rahmati Ziveh <rahmati_ziveh@fzp.czu.cz>

Repository CRAN

Date/Publication 2025-04-10 20:10:02 UTC

Contents

`calc_rn` *Calculate Net Radiation (Rn)*

Description

Computes net radiation (Rn) based on solar radiation, temperature, and elevation.

Usage

```
calc_rn(tmax, tmin, rs, elevation, albedo = 0.23, x = NULL)

## S4 method for signature 'missing,missing,missing,missing,missing'
calc_rn(x)

## S4 method for signature 'Raster,Raster,Raster,Raster,ANY'
calc_rn(tmax, tmin, rs, elevation, albedo = 0.23, x = NULL)
```

```
## S4 method for signature 'character,character,character,character,ANY'
calc_rn(tmax, tmin, rs, elevation, albedo = 0.23, x = NULL)
```

Arguments

tmax	Raster* object or file path for maximum temperature (°C)
tmin	Raster* object or file path for minimum temperature (°C)
rs	Raster* object or file path for solar radiation (MJ m-2 day-1)
elevation	Raster* object or file path for elevation (m)
albedo	Numeric, Raster*, or file path for albedo (optional, default = 0.23)
x	A ‘data.table‘ with columns: "lon", "lat", "date", "rs", "tmax", "tmin", "elevation", and optionally "albedo"

Details

For raster inputs, provide individual raster objects or file paths for ‘tmax‘, ‘tmin‘, ‘rs‘, ‘elevation‘, and optionally ‘albedo‘. For ‘data.table‘ input, provide a single ‘data.table‘ with columns: "lon", "lat", "date", "rs", "tmax", "tmin", "elevation", and optionally "albedo".

Value

RasterBrick or data.table of net radiation values (MJ m-2 day-1)

Examples

```
# Example using Raster* input
if (requireNamespace("raster", quietly = TRUE)) {
  tmax_path <- file.path(tempdir(), "tmax.nc")
  tmin_path <- file.path(tempdir(), "tmin.nc")
  rs_path   <- file.path(tempdir(), "rs.nc")
  elev_path <- file.path(tempdir(), "elevation.nc")

  if (file.exists(tmax_path) && file.exists(tmin_path) &&
      file.exists(rs_path) && file.exists(elev_path)) {
    tmax <- raster::brick(tmax_path)
    tmin <- raster::brick(tmin_path)
    rs   <- raster::brick(rs_path)
    elev <- raster::brick(elev_path)

    rn <- calc_rn(tmax = tmax, tmin = tmin, rs = rs, elevation = elev)
  }
}

# Example using data.table input
if (requireNamespace("data.table", quietly = TRUE)) {
  dt <- data.table::data.table(
    lon = c(10.0, 10.5),
    lat = c(45.0, 45.5),
    date = as.Date(c("2001-06-01", "2001-06-01")),
    tmax = c(28.3, 27.6),
```

```
    tmin = c(14.1, 13.5),
    rs = c(22.5, 21.9),
    elevation = c(400, 420)
  )
  rn_dt <- calc_rn(x = dt)
}
```

---

calc_u2 *Calculate Wind Speed at 2 Meters (u2)*

Description

Computes wind speed at 2 meters (u2) based on wind speed measured at a different height (‘z_u‘).

Usage

```
calc_u2(x, z_u = 10)

## S4 method for signature 'Raster'
calc_u2(x, z_u = 10)

## S4 method for signature 'character'
calc_u2(x, z_u = 10)

## S4 method for signature 'data.table'
calc_u2(x, z_u = 10)
```

Arguments

x	A ‘Raster*‘ object, a file path (‘character‘) to a raster file, or a ‘data.table‘ containing wind speed data.
z_u	Measurement height (m) of the provided wind speed. Default is 10.

Details

For raster inputs, provide a ‘Raster*‘ object or file path (‘character‘) for wind speed at height ‘z_u‘. For ‘data.table‘ input, provide a single ‘data.table‘ with columns: ‘"lon"‘, ‘"lat"‘, ‘"date"‘, and ‘"value"‘, where ‘"value"‘ contains wind speeds at height ‘z_u‘.

If ‘z_u‘ is 2, the function returns the input unchanged.

Value

Wind speed adjusted to 2 meters, returned as:

- A ‘RasterBrick‘, if input is a raster object or file path.
- A ‘data.table‘ with updated ‘"value"‘ column, if input is a ‘data.table‘.

Examples

```
# Raster input
if (requireNamespace("raster", quietly = TRUE)) {
  wind_path <- file.path(tempdir(), "wind_speed.nc")

  if (file.exists(wind_path)) {
    dummie_u <- raster::brick(wind_path)
    u2_raster <- calc_u2(dummie_u, z_u = 10)
  }
}

# File path input
wind_path <- file.path(tempdir(), "wind_speed.nc")
if (file.exists(wind_path)) {
  u2_from_file <- calc_u2(wind_path, z_u = 10)
}

# data.table input
if (requireNamespace("data.table", quietly = TRUE)) {
  dt <- data.table::data.table(
    lon = c(10.0, 10.5),
    lat = c(45.0, 45.5),
    date = as.Date(c("2001-06-01", "2001-06-02")),
    value = c(3.5, 4.1)
  )
  u2_dt <- calc_u2(dt, z_u = 10)
}
```

`detect_exeve` *Detect Extreme Evaporation Events (ExEvE)*

Description

The function `detect_exeve` identifies extreme evaporation events based on standardized evaporation and extreme thresholds.

Usage

```
detect_exeve(x, EXTREMES_THRES = 0.95, LOW_THRES = 0.8)
```

Arguments

`x`	A `data.table` containing columns: `lon`, `lat`, `date`, and `value`, representing daily evaporation values.
`EXTREMES_THRES`	Numeric. Quantile threshold used to define extreme evaporation events. Default is 0.95.
`LOW_THRES`	Numeric. Lower quantile threshold. Default is 0.80.

Value

A `data.table` with original columns and added columns:

- `std_value` — standardized evaporation
- `pentad_std_q80`, `pentad_std_q95` — pentad thresholds
- `value_above_low_thres`, `extreme`, `evap_event` — flags
- `above_low_thres_id`, `extreme_id`, `event_id` — event group IDs

Examples

```
# Example using an RDS file (only run if file exists)
evap_path <- file.path(tempdir(), "czechia_evap_gleam.rds")
if (file.exists(evap_path)) {
  evap <- readRDS(evap_path)
  events <- detect_exeve(evap)
}
```

---

download_data *Download various evapotranspiration data products*

Description

The function `download_data` downloads the selected data product.

Usage

```
download_data(
  data_name = "all",
  path = "",
  domain = "raw",
  time_res = "monthly",
  variable = "e"
)
```

Arguments

`data_name` a character string with the name(s) of the desired data set. Suitable options are:

- "all" for all of the below listed data sets (default),
- "bess" for BESS,
- "camele" for CAMELE,
- "era5" for ERA5,
- "era5-land" for ERA5-Land,
- "fldas" for FLDAS,
- "gldas-clsm-v2-0" for GLDAS CLSM version 2.0,

- "gldas-clsm-v2-1" for GLDAS CLSM version 2.1,
- "gldas-noah-v2-0" for GLDAS NOAH version 2.0,
- "gldas-noah-v2-1" for GLDAS NOAH version 2.1,
- "gldas-vic-v2-0" for GLDAS VIC version 2.0,
- "gldas-vic-v2-1" for GLDAS VIC version 2.1,
- "gleam-v3-7a" for GLEAM version 3.7a,
- "gleam-v4-1a" for GLEAM version 4.1a,
- "jra-55" for JRA-55,
- "merra-2" for MERRA-2,
- "terraclimate" for TerraClimate,
- "etmonitor" for ETMonitor,
- "etsynthesis" for SynthesizedET,
- "sith" for Simple Terrestrial Hydrosphere model, version 2.

`path` a character string with the path where the database will be downloaded.

`domain` a character string with the desired domain data set. Suitable options are:

- "raw" for default available spatial coverage,
- "global" for data sets with global (land and ocean) coverage,
- "land" for data sets with land only coverage,
- "ocean", for data sets with ocean only coverage.

`time_res` a character string with the desired time resolution. Suitable options are:

- "daily",
- "monthly",
- "yearly".

`variable` a character string specifying the variable to download (default = "e").

Value

No return value, called to download the required data sets.

Examples

```
download_data("gldas-vic-v2-1", tempdir())
```

---

download_t_data *Temperature Data Downloader*

Description

Downloading Temperature data from different datasets

Usage

```
download_t_data(
  data_name,
  path = "",
  domain = "raw",
  time_res = "monthly",
  variable = "all"
)
```

Arguments

data_name	a character string indicating the dataset to download. Suitable options are: • "terraclimate" for TerraClimate dataset, • "cru" for CRU dataset, • "mswx" for MSWX dataset.
path	a character string with the path where the data will be downloaded.
domain	a character string with the desired domain data set. Suitable options are: • "raw" for default available spatial coverage, • "global" for data sets with global (land and ocean) coverage, • "land" for data sets with land only coverage, • "ocean" for data sets with ocean only coverage.
time_res	a character string with the desired time resolution. Suitable options are: • "monthly", • "yearly".
variable	a character string indicating the variable to download. Suitable options are: For TerraClimate dataset: • "t2m" for average temperature, • "tmin" for minimum temperature, • "tmax" for maximum temperature. Use "all" to download all available variables for the dataset.

Value

No return value, called to download the required data sets.

Examples

```
download_t_data("cru", tempdir())
```

gldas_clsm_esp_ts *Monthly Evapotranspiration data*

Description

A subset of GLDAS CLSM monthly Evapotranspiration data in mm over Spain. More detail about raw data can be found here.

Usage

```
gldas_clsm_esp_ts
```

Format

A data.table with 120 obs. of 2 variables:

date IDate format %Y-%m-%d

value monthly average values

Source

National Aeronautics and Space Administration (NASA)

gldas_clsm_global_ts *Monthly Evapotranspiration data*

Description

Global GLDAS monthly Evapotranspiration data in mm. More details of the raw data can be found here.

Usage

```
gldas_clsm_global_ts
```

Format

A data.table with 120 obs. of 2 variables:

date IDate format %Y-%m-%d

value monthly average values

Source

National Aeronautics and Space Administration (NASA)

gldas_clsm_subset_ts *Monthly Evapotranspiration data*

Description

A subset of GLDAS monthly Evapotranspiration data in mm over -10-40E, 30-45N. More details of the raw data can be found here.

Usage

```
gldas_clsm_subset_ts
```

Format

A data.table with 120 obs. of 2 variables:

date IDate format %Y-%m-%d

value monthly average values

Source

National Aeronautics and Space Administration (NASA)

pet *Potential Evapotranspiration*

Description

The function pet estimates PET using various methods.

Usage

```
pet(method = "oudin", ...)
```

Arguments

method	Character string indicating the PET estimation method. Available options include:

- "abtew" — Abtew (1996)
- "baier_robertson" — Baier and Robertson (1965)
- "blaney_criddle" — Blaney and Criddle (1950)
- "hamon" — Hamon (1961)
- "hargreaves_samani" — Hargreaves and Samani (1985)
- "jensen_haise" — Jensen and Haise (1963)
- "mcguinness_bordne" — McGuinness and Bordne (1972)

- "oudin" — Oudin (2005). **Default**
- "penman_monteith_f56" — FAO Penman-Monteith (FAO-56)
- "priestly_taylor" — Priestly and Taylor (1972)
- "thornthwaite" — Thornthwaite (1948)
- "turc" — Turc (1961)

`...` Inputs passed to the selected method. These can be:

- A single Raster* object, file path, or data.table — for methods requiring one variable (e.g., oudin);
- Named arguments (e.g., tavg = ..., tmax = ...) — for multi-variable methods;
- A data.table with all required variables — passed as x = your_data.

Details

For single-input methods (e.g., Oudin), you can pass the input directly. For multi-input methods (e.g., Penman-Monteith), use named arguments or a data.table. Use pet_method_requirements() to check required variables.

Value

A 'Raster*' object in mm/day (if raster-based inputs) or a 'data.table' with columns lon, lat, date, and value (PET in mm/day).

Examples

```
# Oudin method with NetCDF path
tavg_path <- file.path(tempdir(), "tavg.nc")
if (file.exists(tavg_path)) {
  pet_od <- pet(method = "oudin", x = tavg_path)
  pet_od <- muldpm(pet_od)
}

# Oudin method with raster
if (requireNamespace("raster", quietly = TRUE)) {
  if (file.exists(tavg_path)) {
    tavg <- raster::brick(tavg_path)
    pet_od <- pet("oudin", tavg)
    pet_od <- muldpm(pet_od)
  }
}

# Oudin method with data.table
if (requireNamespace("data.table", quietly = TRUE)) {
  dt <- data.table::data.table(
    lon = c(10.0, 10.5),
    lat = c(45.0, 45.5),
    date = as.Date(c("2001-06-01", "2001-06-02")),
    tavg = c(18.5, 19.2)
  )
  pet_od <- pet(method = "oudin", x = dt)
```

```
  pet_od <- muldpm(pet_od)
}

# Hargreaves-Samani method with multiple raster inputs
tmax_path <- file.path(tempdir(), "tmax.nc")
tmin_path <- file.path(tempdir(), "tmin.nc")
if (requireNamespace("raster", quietly = TRUE)) {
  if (file.exists(tavg_path) && file.exists(tmax_path) && file.exists(tmin_path)) {
    tavg <- raster::brick(tavg_path)
    tmax <- raster::brick(tmax_path)
    tmin <- raster::brick(tmin_path)
    pet_hs <- pet(method = "hargreaves_samani", tavg = tavg, tmin = tmin, tmax = tmax)
    pet_hs <- muldpm(pet_hs)
  }
}
```

`pet_method_requirements` *Required Input Variables for PET Methods*

Description

This function returns the required input variables for a given PET method, or all methods if none is specified.

Usage

```
pet_method_requirements(x = NULL)
```

Arguments

x	Optional. A character string naming a PET method. If NULL (default), the function returns a named list of all available methods and their required input variables.

Value

A character vector of required input variables (if x is provided), or a named list of all PET methods and their required inputs (if x is NULL).

Examples

```
# PET method requirements
pet_method_requirements()
```

pet_oudin_esp_ts *Monthly Potential Evapotranspiration data*

Description

A subset of calculated monthly Potential Evapotranspiration data in mm over Spain. More details of the used method can be found 'https://www.sciencedirect.com/science/article/pii/S0022169404004056'.

Usage

```
pet_oudin_esp_ts
```

Format

A data.table with 120 obs. of 2 variables:

date IDate format %Y-%m-%d

value monthly average values

Source

Data was calculated using the Oudin method based on raw temperature data. More details of the raw data can be found 'https://journals.ametsoc.org/view/journals/bams/103/3/BAMS-D-21-0145.1.xml'.

pet_oudin_global_ts *Monthly Potential Evapotranspiration data*

Description

Monthly Potential Evapotranspiration data in mm calculated by Oudin method. More details of the used method can be found 'https://www.sciencedirect.com/science/article/pii/S0022169404004056'.

Usage

```
pet_oudin_global_ts
```

Format

A data.table with 120 obs. of 2 variables:

date IDate format %Y-%m-%d

value monthly average values

Source

Data was calculated using the Oudin method based on raw temperature data. More details of the raw data can be found 'https://journals.ametsoc.org/view/journals/bams/103/3/BAMS-D-21-0145.1.xml'.

pet_oudin_subset_ts *Monthly Potential Evapotranspiration data*

Description

A subset of Monthly Potential Evapotranspiration data in mm calculated by Oudin method over -10-40E, 30-45N. More details of the used method can be found 'https://www.sciencedirect.com/science/article/pii/S

Usage

```
pet_oudin_subset_ts
```

Format

A data.table with 120 obs. of 2 variables:

date IDate format %Y-%m-%d

value monthly average values

Source

Data was calculated using the Oudin method based on raw temperature data. More details of the raw data can be found 'https://journals.ametsoc.org/view/journals/bams/103/3/BAMS-D-21-0145.1.xml'.

Index