Package 'RFplus'

March 1, 2025

Type Package

Title Machine Learning for Merging Satellite and Ground Precipitation Data

Version 1.4-0

Date 2025-03-15

Encoding UTF-8

Maintainer Jonnathan Augusto Landi Bermeo <jonnathan.landi@outlook.com>

Description A machine learning algorithm that merges satellite and ground precipitation data using Random Forest for spatial prediction, residual modeling for bias correction, and quantile mapping for adjustment, ensuring accurate estimates across temporal scales and regions.

License GPL (>=3)

Depends R (>= 4.4.0)

Imports terra, randomForest, data.table, pbapply, qmap, hydroGOF

URL https://github.com/Jonnathan-Landi/RFplus

BugReports https://github.com/Jonnathan-Landi/RFplus/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2025-02-04 10:10:00 UTC

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0), covr

Config/testthat/edition 3

VignetteBuilder knitr

Author Jonnathan Augusto Landi Bermeo [aut, cre, cph] (<https://orcid.org/0009-0003-3162-6647>), Alex Avilés [aut] (<https://orcid.org/0000-0001-9278-5738>), Darío Zhiña [aut] (<https://orcid.org/0000-0001-9556-4025>), Marco Mogro [aut] (<https://orcid.org/0009-0007-1802-9417>)

Contents

BD_Insitu *Precipitation Station Measurement Dataset*

Description

This dataset contains daily measurements from several precipitation stations. The first column represents the measurement date, and the following columns correspond to the measurements from each station on that date. The station columns are labeled with unique identifiers for each station, and the number of stations may vary depending on the dataset configuration.

Usage

```
data("BD_Insitu")
```

Format

A 'data.table' object with station measurements. The dataset includes the following columns:

`Date` The measurement date (type `Date`).

`Station_ID_1, Station_ID_2, ...` Measurements from the stations (numeric values). Each column after `Date` represents the measurements of a precipitation station for the corresponding date. The columns are labeled with unique identifiers (e.g., `Station_ID_1`, `Station_ID_2`, etc.) for each station, and the number of stations (columns) may vary.

Details

The data represents daily measurements taken from several precipitation stations. The first column contains the measurement dates, and the following columns represent the measurements of each station on those dates. The number of stations may vary depending on the dataset, and each station is uniquely identified by its column name (e.g., `Station_ID_1`, `Station_ID_2`, etc.).

Source

The data was generated for use in the bias correction model for satellite products, RFplus.

Examples

```
data(BD_Insitu)
## You can use str(BD_Insitu) to get a description of the structure
## or view some of the first rows using head(BD_Insitu)
```

Cords_Insitu *Precipitation Station Coordinates Dataset*

Description

This dataset contains the coordinates (in UTM format) of several precipitation stations. Each station is uniquely identified by the Cod column, which corresponds to the station identifiers used in the BD_Insitu dataset. The coordinates of each station are provided in two columns: X for the Easting (longitude) and Y for the Northing (latitude).

Usage

```
data("Cords_Insitu")
```

Format

A 'data.table' object with station coordinates. The dataset includes the following columns:

Cod The unique identifier for each station. This should correspond to the station columns in the BD_Insitu dataset.

X The Easting (X-coordinate) of the station in UTM format (numeric).

Y The Northing (Y-coordinate) of the station in UTM format (numeric).

Details

The data represents the geographic coordinates of precipitation stations used in the analysis. The first column, Cod, contains the unique identifiers of the stations, which should match the column names in the BD_Insitu dataset. The subsequent columns, X and Y, contain the UTM coordinates for each station, representing the station's location on the Earth's surface.

Source

The data was generated for use in the bias correction model for satellite products, RFplus.

Examples

```
data(Cords_Insitu)
## You can use str(Cords_Insitu) to get a description of the structure
## or view some of the first rows using head(Cords_Insitu)
```

RFplus *Machine learning algorithm for fusing ground and satellite precipitation data.*

Description

MS-GOP is a machine learning algorithm for merging satellite-based and ground precipitation data. It combines Random Forest for spatial prediction, residual modeling for bias correction, and quantile mapping for final adjustment, ensuring accurate precipitation estimates across different temporal scales

Usage

```
RFplus(BD_Insitu, Cords_Insitu, Covariates, ...)

## Default S3 method:
RFplus(
  BD_Insitu,
  Cords_Insitu,
  Covariates,
  n_round = NULL,
  wet.day = FALSE,
  ntree = 2000,
  seed = 123,
  training = 1,
  Rain_threshold = list(no_rain = c(0, 1)),
  method = c("RQUANT", "QUANT", "none"),
  ratio = 15,
  save_model = FALSE,
  name_save = NULL,
  ...
)

## S3 method for class 'data.table'
RFplus(
  BD_Insitu,
  Cords_Insitu,
  Covariates,
  n_round = NULL,
  wet.day = FALSE,
  ntree = 2000,
  seed = 123,
  training = 1,
  Rain_threshold = list(no_rain = c(0, 1)),
  method = c("RQUANT", "QUANT", "none"),
  ratio = 15,
  save_model = FALSE,
  name_save = NULL,
  ...
)
```

Arguments

BD_Insitu	'data.table' containing the ground truth measurements (dependent variable) used to train the RFplus model. Each column represents a ground station, and station identifiers are stored as column names. The class of 'BD_Insitu' must be 'data.table'. Each row represents a time step with measurements of the corresponding station.
Cords_Insitu	'data.table' containing metadata for the ground stations. Must include the following columns: - 'Cod': Unique identifier for each ground station. - 'X': Latitude of the station in UTM format. - 'Y': Longitude of the station in UTM format.
Covariates	A list of covariates used as independent variables in the RFplus model. Each covariate should be a 'SpatRaster' object (from the 'terra' package) and can represent satellite-derived weather variables or a Digital Elevation Model (DEM). All covariates should have the same number of layers (bands), except for the DEM, which must have only one layer.
...	Additional arguments to pass to the underlying methods (e.g., for model tuning or future extensions).
n_round	Numeric indicating the number of decimal places to round the corrected values. If 'n_round' is set to 'NULL', no rounding is applied.
wet.day	Numeric value indicating the threshold for wet day correction. Values below this threshold will be set to zero. - 'wet.day = FALSE': No correction is applied (default). - For wet day correction, provide a numeric threshold (e.g., 'wet.day = 0.1').
ntree	Numeric indicating the maximum number trees to grow in the Random Forest algorithm. The default value is set to 2000. This should not be set to too small a number, to ensure that every input row gets predicted at least a few times. If this value is too low, the prediction may be biased.
seed	Integer for setting the random seed to ensure reproducibility of results (default: 123).
training	Numerical value between 0 and 1 indicating the proportion of data used for model training. The remaining data are used for validation. Note that if you enter, for example, 0.8 it means that 80 If you do not want to perform validation, set training = 1. (Default training = 1).
Rain_threshold	A list of numeric vectors that define the precipitation thresholds for classifying rainfall events into different categories based on intensity. Each element of the list should represent a category, with the category name as the list element's name and a numeric vector specifying the lower and upper bounds for that category. **Note:** See the "Notes" section for additional details on how to define categories, use this parameter for validation, and example configurations.
method	A character string specifying the quantile mapping method used for distribution adjustment. Options are: - "RQUANT": Robust quantile mapping to adjust satellite data distribution to observed data. - "QUANT": Standard quantile mapping. - "none": No distribution adjustment is applied. Only Random Forest-based bias correction and residual correction are performed.
ratio	integer Maximum search radius (in kilometers) for the quantile mapping setting using the nearest station. (default = 15 km)

save_model	Logical value indicating whether the corrected raster layers should be saved to disk. The default is 'FALSE'. If set to 'TRUE', make sure to set the working directory beforehand using 'setwd(path)' to specify where the files should be saved.
name_save	Character string. Base name for output file (default: NULL). The output file will be saved as "Model_RFplus.nc". If you set a different name, make sure you do not set the ".nc" format, as the code will internally assign it.

Details

The 'RFplus' method implements a three-step approach:

- **Base Prediction**: Random Forest model is trained using satellite data and covariates.

- **Residual Correction**: A second Random Forest model is used to correct the residuals from the base prediction.

- **Distribution Adjustment**: Quantile mapping (QUANT or RQUANT) is applied to adjust the distribution of satellite data to match the observed data distribution.

The final result combines all three steps, correcting the biases while preserving the outliers, and improving the accuracy of satellite-derived data such as precipitation and temperature.

Value

A list containing two elements:

Ensamble: A 'SpatRaster' object containing the bias-corrected layers for each time step. The number of layers corresponds to the number of dates for which the correction is applied. This represents the corrected satellite data adjusted for bias.

Validation: A list containing the statistical results obtained from the validation process. This list includes:

- gof: A data table with goodness-of-fit metrics such as Kling-Gupta Efficiency (KGE), Nash-Sutcliffe Efficiency (NSE), Percent Bias (PBIAS), Root Mean Square Error (RMSE), and Pearson Correlation Coefficient (CC). These metrics assess the overall performance of the bias correction process.

- categorical_metrics: A data frame containing categorical evaluation metrics such as Probability of Detection (POD), Success Ratio (SR), False Alarm Rate (FAR), Critical Success Index (CSI), and Hit Bias (HB). These metrics evaluate the classification performance of rainfall event predictions based on user-defined precipitation thresholds.

Notes

The 'Rain_threshold' parameter is used to classify precipitation events based on intensity into different categories. For example:

```
Rain_threshold = list( no_rain = c(0, 1), light_rain = c(1, 5), moderate_rain = c(5, 20),
heavy_rain = c(20, 40), violent_rain = c(40, 100) )
```

Precipitation values will be classified into these categories based on their intensity. Users can define as many categories as necessary, or just two (e.g., "rain" vs. "no rain").

This parameter is required only when 'training' is not equal to 1, as it is needed to calculate performance metrics such as the Probability of Detection (POD), False Alarm Rate (FAR), and Critical Success Index (CSI).

Author(s)

Jonnathan Augusto landi Bermeo, jonnathan.landi@outlook.com

Examples

```
# Load the libraries
library(terra)
library(data.table)

# Load the data
 data("BD_Insitu", package = "RFplus")
 data("Cords_Insitu", package = "RFplus")

# Load the covariates
Covariates <- list(
 MSWEP = terra::rast(system.file("extdata/MSWEP.nc", package = "RFplus")),
 CHIRPS = terra::rast(system.file("extdata/CHIRPS.nc", package = "RFplus")),
 DEM = terra::rast(system.file("extdata/DEM.nc", package = "RFplus"))
 )

 # Apply the RFplus bias correction model
model = RFplus(BD_Insitu, Cords_Insitu, Covariates, n_round = 1, wet.day = 0.1,
        ntree = 2000, seed = 123, training = 1, Rain_threshold = 0.1, method = "RQUANT",
        ratio = 5, save_model = FALSE, name_save = NULL)
# Visualize the results
# Precipitation results within the study area
modelo_rainfall = model$Ensamble
# Validation statistic results
# goodness-of-fit metrics
metrics_gof = model$Validation$gof

# categorical metrics
metrics_cat = model$Validation$categorical_metrics
# Note: In the above example we used 80% of the data for training and 20% for # model validation.
```

Index