Package ‘fluxfixer’

February 2, 2026

Type Package

Title Advanced Framework for Sap Flow Data Post-Process

Version 1.0.0

Maintainer Yoshiaki Hata <yoshiakihata0806@gmail.com>

Description Provides a flexible framework for post-processing thermal dissipation sap flow data using statistical methods and machine learning. This framework includes anomaly correction, outlier removal, gap-filling, trend removal, signal damping correction, and sap flux density calculation. The functions in this package can also apply to other time series with various artifacts.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr (>= 1.0.7), gsignal (>= 0.3-4), lubridate (>= 1.7.9), magrittr (>= 2.0.3), ranger (>= 0.13.1), rlang (>= 1.1.3), stats (>= 4.0.3), tidyr (>= 1.1.4), tidyselect (>= 1.2.0), xts (>= 0.12.1), zoo (>= 1.8-8)

RoxygenNote 7.3.3

Depends R (>= 4.0)

URL https://github.com/yhata86/fluxfixer,
https://yhata86.github.io/fluxfixer/

BugReports https://github.com/yhata86/fluxfixer/issues

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Collate 'utils.R' 'calc_dtmax.R' 'calc_fd.R' 'construct_randomforest.R' 'datasets.R' 'fluxfixer-package.R' 'remove_statistical_outlier.R' 'run_fluxfixer.R' 'utils-pipe.R'

VignetteBuilder knitr

NeedsCompilation no

Author Yoshiaki Hata [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0001-7703-0863>)

Repository CRAN

Date/Publication 2026-02-02 10:30:02 UTC

Contents

calc_dtmax *Calculate dTmax by various methods*

Description

‘calc_dtmax()‘ calculates the time series of dTmax (the maximum temperature difference between sap flow probes under zero-flow conditions) using multiple methods.

Usage

```
calc_dtmax(
  vctr_time,
  vctr_dt,
  vctr_radi = NULL,
  vctr_ta = NULL,
  vctr_vpd = NULL,
  method = c("sp"),
  thres_hour_sp = 5,
  thres_radi = 100,
  thres_ta = 1,
  thres_vpd = 1,
  thres_cv = 0.005,
  thres_hour_pd = 8,
  min_n_wndw_dtmax = 3,
  wndw_size_dtmax = 11,
  output_daily = FALSE
)
```

Arguments

`vctr_time` A timestamp vector of class POSIXct or POSIXt. This vector indicates the timings of the end of each measurement in local time. Any interval (typically 15 to 60 min) is allowed, but the timestamps must be equally spaced and arranged chronologically.

`vctr_dt` A vector of dT (the temperature difference between sap flow probes, in degrees Celsius) time series. The length of the vector must match that of the timestamp vector. Missing values must be gap-filled previously.

`vctr_radi` A vector of global solar radiation or a similar radiative variable time series. Default is ‘NULL‘, but this vector must be provided when ‘method‘ includes ‘pd‘, ‘mw‘, ‘dr‘, or ‘ed‘. The length of the vector must match that of the timestamp vector. Missing values must be gap-filled previously. The unit of the time series must match that of ‘thres_radi‘.

`vctr_ta` A vector of air temperature (degrees Celsius) time series. Default is ‘NULL‘, but this vector must be provided when ‘method‘ includes ‘ed‘. The length of the vector must match that of the timestamp vector. Missing values must be gap-filled previously. The unit of the time series must match that of ‘thres_ta‘.

`vctr_vpd` A vector of vapor pressure deficit (VPD, in hPa) time series. Default is ‘NULL‘, but this vector must be provided when ‘method‘ includes ‘ed‘. The length of the vector must match that of the timestamp vector. Missing values must be gap-filled previously. The unit of the time series must match that of ‘thres_vpd‘.

`method` A vector of characters indicating the dTmax estimation methods. "sp", "pd", "mw", "dr", and "ed" represent the successive predawn, daily predawn, moving window, double regression, and environmental dependent method, respectively. Default is ‘c("sp")‘.

`thres_hour_sp` An integer from 0 to 23. The threshold hour of the day which defines the start of predawn in local time (default is 5).

`thres_radi` A threshold value of the radiation to define daytime. Default is 100 (W m-2). The data points with radiation values above the threshold are considered daytime values. The unit of the threshold must match that of the input radiation time series.

`thres_ta` A threshold value of the air temperature to define predawn. Default is 1.0 (degrees Celsius). The dTmax, estimated by the PD method, with air temperature values below the threshold, is selected as a candidate for the final dTmax. The unit of the threshold must match that of the input air temperature time series.

`thres_vpd` A threshold value of the VPD to define predawn. Default is 1.0 (hPa). The dTmax, estimated by the PD method, with VPD values below the threshold, is selected as a candidate for the final dTmax. The unit of the threshold must match that of the input VPD time series.

thres_cv A threshold value of the coefficient of variation (CV) to define predawn. Default is 0.005. The dTmax, estimated by the PD method, with CV values below the threshold, is selected as a candidate for the final dTmax.

thres_hour_pd An integer from 0 to 23. The threshold hour of the day which defines the end of predawn in local time (default is 8).

min_n_wndw_dtmax

A positive integer indicating the minimum number of data points for calculating statistics using a moving window (default is 3). If the number of data points is less than this threshold, the statistics are not calculated in the window.

wndw_size_dtmax

A positive integer indicating the window size (days) for determining moving window maximum values of dTmax. Default is 11 (days).

output_daily A boolean. If ‘TRUE‘, returns dTmax time series in daily steps; else, returns dTmax in the original time steps. Default is ‘FALSE‘.

Details

This function provides multiple dTmax time series estimated by different methods below:

* The successive predawn (SP) method defines the dTmax for a day as the maximum dT (the temperature difference between sap flow probes) within a 24-hour period that begins at 5:00 a.m. (default; just before daybreak in temperate zones). In other words, the day starts at predawn, not midnight, and the maximum value for that period is assumed to be dTmax. This method has the advantage of being able to calculate dTmax quickly while minimizing the effect of nocturnal transpiration on dTmax estimation.

* The daily predawn (PD) method defines the dTmax for a day as the maximum dT between midnight and the morning (8:00 a.m. in local time) when the global solar radiation is below the threshold value (100 W m-2). See more details in Peters et al. (2018; New Phytologist).

* The moving window (MW) method selects the maximum value of dTmax, estimated by the daily predawn method, using a moving window with an eleven-day length. The selected dTmax is considered to be the final dTmax. See more details in Peters et al. (2018; New Phytologist).

* The double regression (DR) method first calculates the moving window mean value of dTmax, estimated by the daily predawn method, with an eleven-day length. The dTmax that is lower than the mean is omitted, and then the moving window mean is recalculated as the final dTmax. See more details in Peters et al. (2018; New Phytologist).

* The environmental dependent (ED) method filters the dTmax, estimated by the daily predawn method, using the environmental conditions when plants let their sap flow nearly zero. A stable dT, with a low coefficient of variation, and low air temperature or vapor pressure deficit over a two-hour period, characterizes these zero-flow conditions. See more details in Oishi et al. (2016; SoftwareX) and Peters et al. (2018; New Phytologist). After the filtering, the daily dTmax is interpolated if necessary.

Value

A data frame with columns below:

* The first column, ‘time‘, gives the timestamp of the measurements. If ‘output_daily‘ is ‘FALSE‘ (default), this column is the same as the input timestamp, ‘vctr_time‘. If ‘output_daily‘ is ‘TRUE‘, the timestamp in daily steps is returned.

* The second column, ‘dt‘, gives the input dT time series. If ‘output_daily‘ is ‘TRUE‘, dT is returned in daily steps. If ‘output_daily‘ is ‘FALSE‘ (default), this column is not output.

* The other columns, which have the prefix "dtmax_", provide the dTmax calculated by the methods specified in ‘method‘. The last two letters of the column name represent the name of the dTmax estimation method. "sp", "pd", "mw", "dr", and "ed" represent the successive predawn, daily predawn, moving window, double regression, and environmental dependent method, respectively. If ‘output_daily‘ is ‘FALSE‘ (default), this column has the same time step as the input timestamp. If ‘output_daily‘ is ‘TRUE‘, the dTmax is returned in daily steps."

Examples

```
## Load data
data(dt_gf)
time <- dt_gf$time[1:480]
dt <- dt_gf$dt[1:480]
radi <- dt_gf$sw_in[1:480]
ta <- dt_gf$ta[1:480]
vpd <- dt_gf$vpd[1:480]

## Calculate dTmax from gap-filled dT time series
result <-
 calc_dtmax(vctr_time = time, vctr_dt = dt, vctr_radi = radi, vctr_ta = ta,
            vctr_vpd = vpd, method = c("sp", "pd", "mw", "dr", "ed"),
            thres_vpd = 6.0)
```

calc_fd *Calculate sap flux density time series*

Description

‘calc_fd()‘ calculates Fd (sap flux density) time series by a power-type function, including heartwood correction.

Usage

```
calc_fd(
  vctr_dt,
  vctr_dtmax,
  alpha = 1.19 * 10^(-4),
  beta = 1.231,
  do_heartwood_correction = FALSE,
  ratio_conductive = NULL
)
```

Arguments

`vctr_dt` A vector of dT (the temperature difference between sap flow probes, in degrees Celsius) time series. The length of the vector must match that of the timestamp vector. Missing values must be gap-filled previously.

`vctr_dtmax` A vector of dTmax (the maximum temperature difference between sap flow probes under zero-flow conditions, in degrees Celsius) time series. The length of the vector must match that of ‘vctr_dt‘. Missing values must be gap-filled previously.

`alpha` A positive value representing a multiplier in the equation to calculate Fd. Default is 1.19 * 10^(-4) (m3 m-2 s-1).

`beta` A positive value representing a power in the equation to calculate Fd. Default is 1.231.

`do_heartwood_correction`
A boolean. If ‘TRUE‘, the heartwood correction is applied to correct dT before calculating Fd; else, the correction is not applied. Default is ‘FALSE‘.

`ratio_conductive`
A number between 0 and 1, indicating the ratio of the probe length to sapwood width. This parameter must be provided if ‘do_heartwood_correction‘ is ‘TRUE‘. Default is ‘NULL‘.

Details

Fd is estimated using a power-type function introduced by Granier (1985; Annales des Sciences Forestieres, 1987; Tree Physiology). First, a dimensionless index K is obtained from dT and dTmax. Second, K is raised to the power ‘beta‘ and then multiplied by ‘alpha‘, obtaining Fd.

If the sapwood width is shorter than the probe insertion length, dT can be overestimated, resulting in an underestimation of Fd. Therefore, heartwood correction is required to correct dT. Optionally, before calculating Fd, dT can be replaced with the corrected dT by specifying the ratio of the probe length to sapwood width. This correction assumes that the dT measured by the part of the probe that is inserted into the heartwood is always dTmax. See more details in Clearwater et al. (1999; Tree Physiology).

Value

A vector of Fd (m3 m-2 s-1). The length of the vector matches that of the input dT and dTmax vectors.

Author(s)

Yoshiaki Hata

Examples

```
## Load data
data(dt_gf)
data(dtmax)
dt <- dt_gf$dt
dtmax <- dtmax$dtmax_sp
```

```
## Calculate sap flux density
result <- calc_fd(vctr_dt = dt, vctr_dtmax = dtmax)
```

`calc_ref_stats` *Define reference values of average and standard deviation*

Description

‘calc_ref_stats()‘ determines reference values of average and standard deviation for the entire period by calculating the median of these statistical values for the first several days in each sub-period.

Usage

```
calc_ref_stats(
  vctr_time,
  vctr_target,
  vctr_time_prd_tail = NULL,
  wndw_size_ref = 48 * 15,
  label_err = -9999
)
```

Arguments

`vctr_time` A timestamp vector of class POSIXct or POSIXt.

`vctr_target` A vector of a targeted time series to be checked. The length of the time series must be the same as that of ‘vctr_time‘.

`vctr_time_prd_tail` A timestamp vector of class POSIXct or POSIXt, indicating the end timings of each sub-period. Note that users must not include the final timestamp for the entire time series. For instance, if users want to split the entire measurement period into three sub-periods, they only need to specify the end time stamps of the first two sub-periods. Default is ‘NULL‘.

`wndw_size_ref` A positive integer indicating the number of data points included in calculating the average and standard deviation for their reference value determination. The default is 48 * 15, meaning that the first 15 days of each sub-period are used in the calculation when the time interval of the input timestamp is 30 minutes.

`label_err` A numeric value representing a missing value in the input vector(s). Default is -9999.

Value

A vector of two components. The first one is the reference average, and the second one is the reference standard deviation. The unit of these values is the same as that of the input time series.

Author(s)

Yoshiaki Hata

See Also

retrieve_ts

Examples

```
## Load data
data(dt_noisy)
time <- dt_noisy$time[11931:12891]
target <- dt_noisy$dt[11931:12891]
time_prd_tail <- time[148]

## Calculate reference values of average and standard deviation
result <-
  calc_ref_stats(vctr_time = time, vctr_target = target,
                 vctr_time_prd_tail = time_prd_tail)
```

calc_sw_in_toa *Calculate global solar radiation time series at TOA*

Description

'calc_sw_in_toa()' obtains incident global solar radiation time series at TOA (top of atmosphere) at a specific location by calculating solar elevation angle estimated from the equations of Campbell and Norman (1998).

Usage

```
calc_sw_in_toa(
  vctr_time,
  lat,
  lon,
  std_meridian,
  solar_const = 1365,
  sbeta_min = 0.001
)
```

Arguments

vctr_time	A timestamp vector of class POSIXct or POSIXt. The timestamps must be equally spaced and arranged chronologically.
lat	A numeric value (degrees) between -90 and 90, indicating the latitude of the specific location.

lon	A numeric value (degrees) between -180 and 180, indicating the longitude of the specific location.
std_meridian	A numeric value (degrees) between -180 and 180, indicating the standard meridian of the specific location.
solar_const	A positive value (W m-2) indicating the solar constant. Default is 1365 (W m-2).
sbeta_min	A threshold value of the solar elevation angle (degrees). If the calculated solar elevation angle is less than this threshold, the corresponding global solar radiation becomes zero.

Value

A vector of the global solar radiation at TOA (W m-2). The length of the vector matches that of the input timestamp vector.

Author(s)

Yoshiaki Hata

Examples

```
## Make a timestamp vector
timezone <- "Etc/GMT-8"
time <- seq(as.POSIXct("2026/01/01", tz = timezone),
 as.POSIXct("2026/01/02", tz = timezone), by = "30 min")

## Obtain global solar radiation at Lambir Hills National Park in Malaysia
result <-
 calc_sw_in_toa(vctr_time = time, lat = 4.201007, lon = 114.039079,
                std_meridian = 120)
```

check_absolute_limits *Remove outliers by absolute limits*

Description

‘check_absolute_limits()‘ removes out-of-range values by setting lower and upper limits.

Usage

```
check_absolute_limits(
  vctr_target,
  thres_al_min = 3,
  thres_al_max = 50,
  label_err = -9999
)
```

Arguments

`vctr_target` A vector of a targeted time series to be checked.

`thres_al_min` A threshold value for the input time series to define the lower limit. Default is 3.0. The data points with values below the threshold are considered outliers and removed. The unit of the threshold must match that of the input time series.

`thres_al_max` A threshold value for the input time series to define the upper limit. Default is 50.0. The data points with values above the threshold are considered outliers and removed. The unit of the threshold must match that of the input time series.

`label_err` A numeric value representing a missing value in the input vector(s). Default is -9999.

Value

A vector of cleaned time series. The length of the time series is the same as the input time series. The data points with values below ‘thres_al_min‘ or above ‘thres_al_max‘ are replaced with the error label specified in ‘label_err‘.

Author(s)

Yoshiaki Hata

Examples

```
## Load data
data(dt_noisy)
target <- dt_noisy$dt

## Remove out-of-range values
result <- check_absolute_limits(vctr_target = target)
```

dtmax *dTmax time series estimated by multiple methods*

Description

Dataset consisting of the time series of the dTmax (the maximum temperature difference between thermal dissipation sap flow probes under zero-flow conditions) calculated by the successive predawn (SP), daily predawn (PD), moving window (MW), double regression (DR), and environmental dependent (ED) methods.

Usage

```
data(dtmax)
```

Format

A data frame with 17520 rows and 6 variables:

time Timestamp of the measurement end timing

dtmax_sp dTmax estimated by the SP method (degrees Celsius)

dtmax_pd dTmax estimated by the PD method (degrees Celsius)

dtmax_mw dTmax estimated by the MW method (degrees Celsius)

dtmax_dr dTmax estimated by the DR method (degrees Celsius)

dtmax_ed dTmax estimated by the ED method (degrees Celsius)

`dt_gf` *Quality-controlled and gap-filled dT time series*

Description

Dataset consisting of the time series of the quality-controlled and gap-filled dT (the temperature difference between thermal dissipation sap flow probes), meteorological factors, and soil water content at Lambir Hills National Park, Malaysia.

Usage

```
data(dt_gf)
```

Format

A data frame with 17520 rows and 8 variables:

time Timestamp of the measurement end timing

dt Quality-controlled and gap-filled dT (degrees Celsius)

p Precipitation (mm)

sw_in Global solar radiation (W m-2)

ta Air temperature (degrees Celsius)

vpd Vapor pressure deficit (hPa)

ws Horizontal wind speed (m s-1)

swc Soil water content (m3 m-3)

Source

Meteorological factors and soil water content data were provided by Dr. Tomonori Kume [ORCiD: 0000-0001-6569-139X]

`dt_noisy` *Raw dT time series with various artifacts*

Description

Dataset consisting of the time series of a raw dT (the temperature difference between thermal dissipation sap flow probes), meteorological factors, and soil water content at Lambir Hills National Park, Malaysia. Missing values are represented as -9999.

Usage

```
data(dt_noisy)
```

Format

A data frame with 17520 rows and 8 variables:

time Timestamp of the measurement end timing

dt Raw dT (degrees Celsius)

p Precipitation (mm)

sw_in Global solar radiation (W m-2)

ta Air temperature (degrees Celsius)

vpd Vapor pressure deficit (hPa)

ws Horizontal wind speed (m s-1)

swc Soil water content (m3 m-3)

Source

Meteorological factors and soil water content data were provided by Dr. Tomonori Kume [ORCiD: 0000-0001-6569-139X]

`fill_gaps` *Fill missing values with a random forest model*

Description

‘fill_gaps()‘ replaces all missing values in a target time series with values estimated by a random forest model whose hyperparameters are calibrated using a grid search approach and out-of-bag evaluation.

Usage

```
fill_gaps(
  df,
  colname_label,
  vctr_colname_feature = NULL,
  vctr_min_nodesize = c(5),
  vctr_m_try = NULL,
  vctr_subsample_gf = c(1),
  frac_train = 0.75,
  n_tree = 500,
  ran_seed = 12345,
  label_err = -9999
)
```

Arguments

`df` A data frame including label (explained variable) and feature (explanatory variables) time series for model input. It is acceptable to include missing values in each column.

`colname_label` A character representing the name of the column for the label time series.

`vctr_colname_feature` A vector of characters indicating the name of the feature time series columns used in constructing a random forest model. If ‘NULL‘ (default), all columns excluding the label column specified as ‘colname_label‘ in the input data frame are used as feature columns.

`vctr_min_nodesize` A vector of positive integers indicating candidates of a hyperparameter for the random forest model, defining the minimal node size (the minimum number of data points included in each leaf node). Default is ‘c(5)‘.

`vctr_m_try` A vector of positive integers indicating candidates of a hyperparameter for the random forest model, defining the number of features to be used in splitting each node. If ‘NULL‘ (default), integers between two and the number of all feature variables are tested.

`vctr_subsample_gf` A vector of numerical values between 0 and 1, indicating candidates of a hyperparameter for the random forest model, defining the fraction of input training data points to be sampled in constructing the random forest. Default is ‘c(1)‘.

`frac_train` A numerical value between 0 and 1, defining the fraction of data points to be categorized as training data for the random forest model construction. The other data points are classified as test data. Default is 0.75.

`n_tree` An integer representing the number of trees in the random forest. Default is 500.

`ran_seed` An integer representing the random seed for the random forest model construction. Default is 12345.

`label_err` A numeric value representing a missing value in the input vector(s). Default is -9999.

Details

A random forest model is constructed for the targeted time series to fill missing values. The time series is assumed to be stationary, so detrending is needed before inputting if it has a trend. Users can input any feature from the dataset, and out-of-bag evaluation is used to determine the hyperparameters. This evaluation is applied to a training dataset separated from the entire input data. To reduce the computational cost, the only hyperparameter used by default for grid search is the number of candidate features. After determining the optimal hyperparameters, they are used to construct the optimal random forest model. Predicted time series are equal to the average of 500 (default) tree outputs at each time point. If the input targeted value is missing, the predicted value is used for the imputation.

Value

A list with two elements. The first element ‘mse‘ is the mean squared error between predicted and original values in the test data set. The second element ‘stats‘ is a data frame with columns below:

* The first column, ‘gapfilled‘, gives the gap-filled time series, where missing values are replaced with the predicted values from the random forest model.

* The second column, ‘avg_predicted‘, gives the ensemble mean time series calculated from estimated values at each time point for each tree in the constructed random forest.

* The third column, ‘sd_predicted‘, gives the ensemble mean time series calculated from estimated values at each time point for each tree in the constructed random forest.

Author(s)

Yoshiaki Hata

Examples

```
## Load data
data(dt_noisy)
df_raw <- dt_noisy[c(13105:14112), ]

## Remove error values for making data gaps
df_raw$dt <- ifelse(df_raw$dt > 9.5, df_raw$dt, -9999)

## Fill data gaps
result <-
  fill_gaps(df = df_raw, colname_label = "dt",
            vctr_colname_feature = c("sw_in", "vpd", "swc", "ta"))$stats
```

`filter_highfreq_noise` *Filter high frequency noise by Gaussian filter*

Description

‘filter_highfreq_noise()‘ filters a time series with a specific period by convolving it with a Gaussian window, removing high-frequency noise.

Usage

```
filter_highfreq_noise(
  vctr_time,
  vctr_target,
  vctr_time_noise,
  wndw_size_noise = 13,
  inv_sigma_noise = 0.01,
  label_err = -9999
)
```

Arguments

`vctr_time` A timestamp vector of class POSIXct or POSIXt. The timestamps must be equally spaced and arranged chronologically.

`vctr_target` A vector of a targeted time series to be checked. The length of the time series must be the same as that of ‘vctr_time‘.

`vctr_time_noise`
A timestamp vector of class POSIXct or POSIXt, indicating when high-frequency noise exists in the targeted time series.

`wndw_size_noise`
A positive integer indicating the number of data points included in a moving Gaussian window for the high-frequency noise filtering. The default is 13, meaning that the window size is 6.5 hours if the time interval of the input timestamp is 30 minutes.

`inv_sigma_noise`
A positive value defining a Gaussian window width for the high-frequency noise filtering. The width of the Gaussian window is inversely proportional to this parameter. Default is 0.01.

`label_err` A numeric value representing a missing value in the input vector(s). Default is -9999.

Value

A vector of the noise-filtered time series. The length of the time series is the same as the input time series.

Author(s)

Yoshiaki Hata

Examples

```
## Create data
time <- seq(as.POSIXct("2026/01/01"), length.out = 360, by = "1 day")
x <- seq(1, 360)
target <- sin(x / 180 * pi) + stats::rnorm(length(x), sd = 0.2)
time_noise <-
  seq(as.POSIXct("2026/01/01"), as.POSIXct("2026/09/01"), by = "1 day")
```

```
## Filter noise
result <-
  filter_highfreq_noise(vctr_time = time, vctr_target = target,
                        vctr_time_noise = time_noise)
```

modify_short_drift *Modify short-term drifts*

Description

‘modify_short_drift()‘ corrects short-term drifts by adjusting the 5th and 95th percentiles of the drifted time series to those of the reference time series.

Usage

```
modify_short_drift(
  vctr_time,
  vctr_target,
  vctr_time_drft_head,
  vctr_time_drft_tail,
  n_day_ref = 3,
  label_err = -9999
)
```

Arguments

`vctr_time` A timestamp vector of class POSIXct or POSIXt. The timestamps must be equally spaced and arranged chronologically.

`vctr_target` A vector of a targeted time series to be checked. The length of the time series must be the same as that of ‘vctr_time‘.

`vctr_time_drft_head` A timestamp vector of class POSIXct or POSIXt, indicating when each drift starts.

`vctr_time_drft_tail` A timestamp vector of class POSIXct or POSIXt, indicating when each drift ends. The length of the time series must be the same as that of ‘vctr_time_drft_head‘.

`n_day_ref` A positive integer representing the number of days to be referenced before and after the anomaly period. Default is 3 (days).

`label_err` A numeric value representing a missing value in the input vector(s). Default is -9999.

Details

The short-term drift correction is to correct sudden changes in the average in the time series over a short period (hours to days) specified by ‘vctr_time_drft_head‘ and ‘vctr_time_drft_tail‘. Multiple short-term drifts can be corrected at once using this function.This procedure uses a reference period, which is defined to consist of the three days (default) before and after the occurrence of the anomaly. Then, the anomalous time series is standardized so that the 5th and 95th percentile values of the anomalous and reference (non-anomalous) time series match over this period. These percentile values are used instead of the maximum and minimum values to ensure robustness against possible outliers in the original or reference time series.

Value

A vector of the drift-corrected time series. The length of the time series is the same as the input time series.

Author(s)

Yoshiaki Hata

Examples

```
## Load data
data(dt_noisy)
time <- dt_noisy$time[11931:12891]
target <- dt_noisy$dt[11931:12891]
time_drft_head <- time[1]
time_drft_tail <- time[148]

## Correct a short-term drift
result <-
  modify_short_drift(vctr_time = time, vctr_target = target,
                     vctr_time_drft_head = time_drft_head,
                     vctr_time_drft_tail = time_drft_tail)
```

remove_manually *Remove error values manually*

Description

‘remove_manually()‘ removes unreasonable values manually by indicating specific timestamps.

Usage

```
remove_manually(vctr_time, vctr_target, vctr_time_err, label_err = -9999)
```

Arguments

vctr_time	A timestamp vector of class POSIXct or POSIXt.
vctr_target	A vector of a targeted time series to be checked. The length of the time series must be the same as that of ‘vctr_time‘.
vctr_time_err	A timestamp vector of class POSIXct or POSIXt, indicating specific error timings.
label_err	A numeric value representing a missing value in the input vector(s). Default is -9999.

Value

A vector of cleaned time series. The length of the time series is the same as the input time series. The data points at the indicated time points by ‘vctr_time_err‘ are replaced with the error label specified in ‘label_err‘.

Author(s)

Yoshiaki Hata

Examples

```
## Load data
data(dt_noisy)
time <- dt_noisy$time[12097:14400]
target <- dt_noisy$dt[12097:14400]
time_err <- seq(as.POSIXct("2013/06/27 18:00", tz = "Etc/GMT-8"),
                as.POSIXct("2013/06/27 22:30", tz = "Etc/GMT-8"),
                by = "30 min")

## Remove error values
result <-
 remove_manually(vctr_time = time, vctr_target = target,
                 vctr_time_err = time_err)
```

remove_rf_outlier *Remove outliers detected by a random forest model*

Description

‘remove_rf_outlier()‘ detects and removes outliers by a random forest model whose hyperparameters are calibrated using a grid search approach and out-of-bag evaluation.

Usage

```
remove_rf_outlier(
  df,
  colname_label,
  vctr_colname_feature = NULL,
  vctr_min_nodesize = c(5),
  vctr_m_try = NULL,
  vctr_subsample_outlier = c(0.1),
  frac_train = 0.75,
  n_tree = 500,
  ran_seed = 12345,
  coef_iqr = 1.5,
  label_err = -9999
)
```

Arguments

`df` A data frame including label (explained variable) and feature (explanatory variables) time series for model input. It is acceptable to include missing values in each column.

`colname_label` A character representing the name of the column for the label time series.

`vctr_colname_feature`
A vector of characters indicating the name of the feature time series columns used in constructing a random forest model. If ‘NULL‘ (default), all columns excluding the label column specified as ‘colname_label‘ in the input data frame are used as feature columns.

`vctr_min_nodesize`
A vector of positive integers indicating candidates of a hyperparameter for the random forest model, defining the minimal node size (the minimum number of data points included in each leaf node). Default is ‘c(5)‘.

`vctr_m_try` A vector of positive integers indicating candidates of a hyperparameter for the random forest model, defining the number of features to be used in splitting each node. If ‘NULL‘ (default), integers between two and the number of all feature variables are tested.

`vctr_subsample_outlier`
A vector of numerical values between 0 and 1, indicating candidates of a hyperparameter for the random forest model, defining the fraction of input training data points to be sampled in constructing the random forest. Default is ‘c(0.1)‘.

`frac_train` A numerical value between 0 and 1, defining the fraction of data points to be categorized as training data for the random forest model construction. The other data points are classified as test data. Default is 0.75.

`n_tree` An integer representing the number of trees in the random forest. Default is 500.

`ran_seed` An integer representing the random seed for the random forest model construction. Default is 12345.

`coef_iqr` A positive value defining a multiplier of the interquartile range (IQR). If the value to be checked is less than Q1 (first quartile) - ‘coef_iqr‘ * IQR or more than Q3 (third quartile) + ‘coef_iqr‘ * IQR, the value is detected as a random forest outlier. Default is 1.5.

`label_err` A numeric value representing a missing value in the input vector(s). Default is -9999.

Details

A random forest model is constructed for the targeted time series to remove outliers. The time series is assumed to be stationary, so detrending is needed before inputting if it has a trend. Users can input any feature from the dataset, and out-of-bag evaluation is used to determine the hyperparameters. This evaluation is applied to a training dataset separated from the entire input data. To reduce the computational cost, the only hyperparameter used by default for grid search is the number of candidate features. To reduce the risk of learning noise, the training data sampling ratio is set to 0.1 by default. After determining the optimal hyperparameters, they are used to construct the optimal random forest model. Output values are obtained from 500 (default) trees, and the first quartile (Q1), third quartile (Q3), and interquartile range (IQR) of the output values at each time point are calculated. If the targeted value is less than Q1 minus 1.5IQR or more than Q3 plus 1.5IQR (default), the data point is identified as an outlier and removed.

Value

A list with two elements. The first element ‘mse‘ is the mean squared error between predicted and original values in the test data set. The second element ‘stats‘ is a data frame with columns below:

* The first column, ‘cleaned‘, gives the cleaned time series after replacing the detected outliers with the value specified by ‘label_err‘.

* The second column, ‘flag_out‘, gives a flag variable time series indicating the status of the cleaned time series (0: the input data point is not originally missing and not detected as an outlier; 1: the input data point is not originally missing but detected as an outlier; 2: the input data point is originally missing).

* The third column, ‘med‘, gives the ensemble median time series calculated from estimated values at each time point for each tree in the constructed random forest.

* The fourth column, ‘q1‘, gives the ensemble Q1 (first quartile) time series calculated from estimated values at each time point for each tree in the constructed random forest.

* The fifth column, ‘q3‘, gives the ensemble Q3 (third quartile) time series calculated from estimated values at each time point for each tree in the constructed random forest.

Author(s)

Yoshiaki Hata

Examples

```
## Load data
data(dt_noisy)
df_raw <- dt_noisy[c(12097:14400), ]

## Remove outliers
result <-
```

```
remove_rf_outlier(df = df_raw, colname_label = "dt",
                  vctr_colname_feature = c("sw_in", "vpd", "swc", "p"),
                  coef_iqr = 3.0)$stats
```

remove_zscore_outlier *Remove outliers by Z-score time series*

Description

'remove_zscore_outlier()' detects and removes outlier values by converting an original time series into a Z-score time series using a moving window.

Usage

```
remove_zscore_outlier(
  vctr_time,
  vctr_target,
  vctr_time_prd_tail = NULL,
  wndw_size_z = 48 * 15,
  min_n_wndw_z = 5,
  thres_z = 5,
  n_calc_max = 10,
  modify_z = FALSE,
  vctr_time_zmod = NULL,
  wndw_size_conv = 48 * 15,
  inv_sigma_conv = 0.01,
  thres_ratio = 0.5,
  label_err = -9999
)
```

Arguments

`vctr_time` A timestamp vector of class POSIXct or POSIXt.

`vctr_target` A vector of a targeted time series to be checked. The length of the time series must be the same as that of 'vctr_time'.

`vctr_time_prd_tail`
A timestamp vector of class POSIXct or POSIXt, indicating the end timings of each sub-period. Note that users must not include the final timestamp for the entire time series. For instance, if users want to split the entire measurement period into three sub-periods, they only need to specify the end time stamps of the first two sub-periods. Default is 'NULL'.

`wndw_size_z` A positive integer indicating the number of data points included in a moving window for the Z-score outlier removal. The default is 48 * 15, meaning that the window size is 15 days if the time interval of the input timestamp is 30 minutes.

`min_n_wndw_z` A positive integer indicating the minimum number of data points for calculating statistics using a moving window (default is 5) for the Z-score outlier removal. If the number of data points is less than this threshold, the statistics are not calculated in the window.

`thres_z` A positive threshold value for the Z-score time series to define outliers. Default is 5.0. The data points with Z-scores (absolute values) above the threshold are considered outliers and removed.

`n_calc_max` A positive integer indicating the maximum number of Z-score outlier detection iterations. Default is 10.

`modify_z` A boolean. If ‘TRUE‘, conduct Z-score short attenuation correction; else, the correction is not applied. Default is ‘FALSE‘.

`vctr_time_zmod` Only valid if ‘modify_z‘ is ‘TRUE‘. A timestamp vector of class POSIXct or POSIXt, indicating the timings when the short-term signal attenuation correction is applied. Default is ‘NULL‘.

`wndw_size_conv` Only valid if ‘modify_z‘ is ‘TRUE‘. A positive integer indicating the number of data points included in a moving window for the short-term signal attenuation detection. The default is 48 * 15, meaning that the window size is 15 days if the time interval of the input timestamp is 30 minutes.

`inv_sigma_conv` Only valid if ‘modify_z‘ is ‘TRUE‘. A positive value defining a Gaussian window width for the short-term signal attenuation detection. The width of the Gaussian window is inversely proportional to this parameter. Default is 0.01.

`thres_ratio` Only valid if ‘modify_z‘ is ‘TRUE‘. A positive threshold value of the ratio for determining whether the signal attenuation correction is applied to each detected attenuation period. The ratio represents the average of the standard deviation at the detected attenuation peak relative to that at the beginning and end of the attenuation period. If the ratio is below this threshold value, the correction is applied. Default is 0.5.

`label_err` A numeric value representing a missing value in the input vector(s). Default is -9999.

Details

The input time series is standardized using a moving window, and the data values are converted to Z-scores. In this step, the width of the moving window is set to 15 days by default, centered on the target time point, and standardization is performed individually for each time point in the time series. The threshold of the Z-score absolute value (default: 5 as specified by ’thres_z’) is set, and data points outside that range are removed as outliers. After the outliers have been removed, the Z-score is returned to the original value using the original mean and standard deviation time series, and standardization is performed again using a moving window to remove additional outliers. These procedures are repeated until either no more outliers are removed or the maximum number of iterations (default 10) is reached.

Users can define sub-periods across the entire time series using ‘vctr_time_prd_tail‘, and the Z-score conversion is performed in each sub-period separately. This separated conversion is useful when the input time series suddenly changes its nature, such as after a sensor replacement.

In some cases, for sap flow measurements, the input dT (the temperature difference between sap flow probes) time series may yield a signal that is attenuated for only a short period, for example, when rainfall continues for days, causing the moving window mean (or standard deviation) to increase (or decrease). In such cases, standardization will cause the Z-score time series immediately before and after the rainfall to be unnaturally distorted, hindering the construction of the random forest model. If ‘modify_z‘ is ‘TRUE‘, after the outlier removal, this function modifies the Z-score time series for periods when the moving window average has an upward peak, and the moving window standard deviation has a downward peak simultaneously. First, the average and standard deviation time series are interpolated if they contain missing values. Second, they are smoothed by convolution with a user-specified Gaussian window, defined by the parameters ‘wndw_size_conv‘ and ‘inv_sigma_conv‘. Third, the first-order and second-order differences of both smoothed time series are calculated, which determine the upward peak positions of the average and the downward peak positions of the standard deviation. Fourth, possible signal attenuation periods are determined based on these peak positions. The start and end of the periods are defined by the timings when the first-order differenced standard deviation time series changes its sign before and after each peak. Fifth, the final attenuation periods are selected if the average of the ratio of the standard deviation at the detected attenuation peak to that at the beginning and end of the attenuation period is below the threshold value specified by ‘thres_ratio‘. Optionally, users can specify the periods to be modified by ‘vctr_time_zmod‘. Sixth, the average and standard deviation time series during the attenuation periods are deleted and linearly interpolated. Finally, the modified Z-score time series is calculated using these average and standard deviation time series.

Value

A data frame with columns below:

* The first column, ‘time‘, gives the same timestamp as ‘vctr_time‘.

* The second column, ‘cleaned‘, gives the cleaned time series after replacing the detected outliers with the value specified by ‘label_err‘.

* The third column, ‘z_cleaned‘, gives the Z-score of the input time series after removing the detected outliers.

* The fourth column, ‘avg_cleaned‘, gives the moving window average of the input time series after removing the detected outliers.

* The fifth column, ‘sd_cleaned‘, gives the moving window standard deviation of the input time series after removing the detected outliers.

* The sixth column, ‘flag_out‘ gives a flag variable time series indicating the status of the cleaned time series (0: the input data point is not originally missing and not detected as an outlier; 1: the input data point is not originally missing but detected as an outlier; 2: the input data point is originally missing).

Author(s)

Yoshiaki Hata

Examples

```
## Load data
data(dt_noisy)
time <- dt_noisy$time[12097:14400]
target <- dt_noisy$dt[12097:14400]
```

```
## Remove outliers
result <- remove_zscore_outlier(vctr_time = time, vctr_target = target)
```

retrieve_ts *Retrieve time series in its original units*

Description

‘retrieve_ts()‘ converts a standardized Z-score time series into a time series in its original units using specific average and standard deviation time series.

Usage

```
retrieve_ts(
  vctr_target_z,
  vctr_target_avg = NA,
  vctr_target_sd = NA,
  detrend = FALSE,
  correct_damping = FALSE,
  avg_ref = NULL,
  sd_ref = NULL,
  label_err = -9999
)
```

Arguments

`vctr_target_z`	A vector of Z-score time series to be converted. Missing values must be gap-filled previously.
`vctr_target_avg`	A vector of average time series. Missing values are acceptable but automatically gap-filled by interpolation during the retrieving process. The length of the vector must match that of ‘vctr_target_z‘. The unit of the time series must match that of time series to be output. Default is ‘NA‘.
`vctr_target_sd`	A vector of standard deviation time series. Missing values are acceptable but automatically gap-filled by interpolation during the retrieving process. The length of the vector must match that of ‘vctr_target_z‘. The unit of the time series must match that of time series to be output. Default is ‘NA‘.
`detrend`	A boolean. If ‘TRUE‘, detrending is applied and the reference average specified by ‘avg_ref‘ is used to convert Z-score time series into the time series with the reference average in its original units; else, the detrending is not applied, and the average time series specified by ‘vctr_target_avg‘ is used in the conversion. Default is ‘FALSE‘.
`correct_damping`	A boolean. If ‘TRUE‘, the signal damping correction is applied and the reference standard deviation specified by ‘sd_ref‘ is used to convert Z-score time

series into the time series in its original units with the reference standard deviation; else, the correction is not applied, and the standard deviation time series specified by ‘vctr_target_sd‘ is used in the conversion. Default is ‘FALSE‘.

`avg_ref` Only valid if ‘detrend‘ is ‘TRUE‘. A numeric value representing the reference average. A vector of reference average time series is also acceptable, but the length of the vector must match that of ‘vctr_target_z‘, and the unit of the time series must match that of time series to be output. Default is ‘NULL‘.

`sd_ref` Only valid if ‘correct_damping‘ is ‘TRUE‘. A positive numeric value representing the reference standard deviation. A vector of reference standard deviation time series is also acceptable, but the length of the vector must match that of ‘vctr_target_z‘, and the unit of the time series must match that of time series to be output. Default is ‘NULL‘.

`label_err` A numeric value representing a missing value in the input vector(s). Default is -9999.

Details

Retrieving a time series with its original units is conducted by multiplying a Z-score by the standard deviation, followed by adding the average. If the average and standard deviation time series are the same as those in converting the original time series into the Z-score time series, the original values with the original average and standard deviation are retrieved. If reference values of the average and/or standard deviation are used, the output time series are detrended and/or applied to signal damping correction.

Value

A vector of the retrieved time series. The length of the vector is the same as ‘vctr_target_z‘.

Author(s)

Yoshiaki Hata

See Also

calc_ref_stats

Examples

```
## Create data
target <- seq(1, 10)
target_avg <- rep(mean(target), 10)
target_sd <- rep(stats::sd(target), 10)
target_z <- (target - target_avg) / target_sd

## Retrieve time series in its original units
result <-
  retrieve_ts(vctr_target_z = target_z, vctr_target_avg = target_avg,
              vctr_target_sd = target_sd)
```

run_fluxfixer *Run all quality control processes automatically*

Description

‘run_fluxfixer()‘ provides a sophisticated protocol for post-processing raw time series, which can be applied not only to thermal dissipation sap flow data but also to other noisy time series, using classic statistical and machine-learning methods. In the sap flow data processing, users can select multiple methods to estimate the dTmax (the maximum temperature difference between sap flow probes under zero-flow conditions) and Fd (sap flux density) time series.

Usage

```
run_fluxfixer(
  df,
  colname_time,
  colname_target,
  vctr_time_err = NULL,
  label_err = -9999,
  thres_al_min = 3,
  thres_al_max = 50,
  vctr_time_drft_head = NULL,
  vctr_time_drft_tail = NULL,
  n_day_ref = 3,
  vctr_time_noise = NULL,
  wndw_size_noise = 13,
  inv_sigma_noise = 0.01,
  vctr_time_prd_tail = NULL,
  wndw_size_z = 48 * 15,
  min_n_wndw_z = 5,
  thres_z = 5,
  n_calc_max = 10,
  modify_z = FALSE,
  vctr_time_zmod = NULL,
  wndw_size_conv = 48 * 15,
  inv_sigma_conv = 0.01,
  thres_ratio = 0.5,
  vctr_colname_feature = NULL,
  vctr_min_nodesize = c(5),
  vctr_m_try = NULL,
  vctr_subsample_outlier = c(0.1),
  vctr_subsample_gf = c(1),
  frac_train = 0.75,
  n_tree = 500,
  ran_seed = 12345,
  coef_iqr = 1.5,
  wndw_size_ref = 48 * 15,
  detrend = FALSE,
  correct_damping = FALSE,
  skip_sapflow_calc = FALSE,
  colname_radi = NULL,
  colname_ta = NULL,
  colname_vpd = NULL,
  method = c(”sp”),
  thres_hour_sp = 5,
  thres_radi = 100,
  thres_ta = 1,
  thres_vpd = 1,
  thres_cv = 0.005,
  thres_hour_pd = 8,
  min_n_wndw_dtmax = 3,
  wndw_size_dtmax = 11,
  alpha = 1.19 * 10^(-4),
  beta = 1.231,
  do_heartwood_correction = FALSE,
  ratio_conductive = NULL
)
```

Arguments

`df` A data frame including evenly spaced time stamps in local time and the corresponding raw time series to be post-processed. For thermal dissipation sap flow data, the targeted time series must be a dT (the temperature difference between sap flow probes) time series. To conduct outlier removal and gap-filling using a random forest model, any time series of meteorological, soil environment, and ecophysiological factors can be included. If users conduct the zero-flow condition estimation, the gap-filled time series of global solar radiation or a similar radiative variable (for the PD, MW, DR, and ED methods) and air temperature and vapor pressure deficit (for the ED method) must be included.

`colname_time` A character representing the name of the column in the input data frame for the timestamp time series. This column indicates the timings of the end of each measurement in local time. Any interval (typically 15 to 60 min) is allowed, but the timestamps must be equally spaced and arranged chronologically.

`colname_target` A character representing the name of the column in the input data frame for a targeted time series to be post-processed. For thermal dissipation sap flow data, the targeted time series must be a dT (the temperature difference between sap flow probes) time series.

`vctr_time_err` A timestamp vector of class POSIXct or POSIXt, indicating specific error timings.

`label_err` A numeric value representing a missing value in the input vector(s). Default is -9999.

`thres_al_min` A threshold value for the input time series to define the lower limit. Default is 3.0. The data points with values below the threshold are considered outliers and removed. The unit of the threshold must match that of the input time series.

`thres_al_max` A threshold value for the input time series to define the upper limit. Default is 50.0. The data points with values above the threshold are considered outliers and removed. The unit of the threshold must match that of the input time series.

`vctr_time_drft_head`
A timestamp vector of class POSIXct or POSIXt, indicating when each drift starts.

`vctr_time_drft_tail`
A timestamp vector of class POSIXct or POSIXt, indicating when each drift ends. The length of the time series must be the same as that of 'vctr_time_drft_head'.

`n_day_ref` A positive integer representing the number of days to be referenced before and after the anomaly period. Default is 3 (days).

`vctr_time_noise`
A timestamp vector of class POSIXct or POSIXt, indicating when high-frequency noise exists in the targeted time series.

`wndw_size_noise`
A positive integer indicating the number of data points included in a moving Gaussian window for the high-frequency noise filtering. The default is 13, meaning that the window size is 6.5 hours if the time interval of the input timestamp is 30 minutes.

`inv_sigma_noise`
A positive value defining a Gaussian window width for the high-frequency noise filtering. The width of the Gaussian window is inversely proportional to this parameter. Default is 0.01.

`vctr_time_prd_tail`
A timestamp vector of class POSIXct or POSIXt, indicating the end timings of each sub-period. Note that users must not include the final timestamp for the entire time series. For instance, if users want to split the entire measurement period into three sub-periods, they only need to specify the end time stamps of the first two sub-periods. Default is 'NULL'.

`wndw_size_z` A positive integer indicating the number of data points included in a moving window for the Z-score outlier removal. The default is 48 * 15, meaning that the window size is 15 days if the time interval of the input timestamp is 30 minutes.

`min_n_wndw_z` A positive integer indicating the minimum number of data points for calculating statistics using a moving window (default is 5) for the Z-score outlier removal. If the number of data points is less than this threshold, the statistics are not calculated in the window.

`thres_z` A positive threshold value for the Z-score time series to define outliers. Default is 5.0. The data points with Z-scores (absolute values) above the threshold are considered outliers and removed.

`n_calc_max` A positive integer indicating the maximum number of Z-score outlier detection iterations. Default is 10.

`modify_z` A boolean. If 'TRUE', conduct Z-score short attenuation correction; else, the correction is not applied. Default is 'FALSE'.

`vctr_time_zmod` Only valid if 'modify_z' is 'TRUE'. A timestamp vector of class POSIXct or POSIXt, indicating the timings when the short-term signal attenuation correction is applied. Default is 'NULL'.

`wndw_size_conv` Only valid if ‘modify_z‘ is ‘TRUE‘. A positive integer indicating the number of data points included in a moving window for the short-term signal attenuation detection. The default is 48 * 15, meaning that the window size is 15 days if the time interval of the input timestamp is 30 minutes.

`inv_sigma_conv` Only valid if ‘modify_z‘ is ‘TRUE‘. A positive value defining a Gaussian window width for the short-term signal attenuation detection. The width of the Gaussian window is inversely proportional to this parameter. Default is 0.01.

`thres_ratio` Only valid if ‘modify_z‘ is ‘TRUE‘. A positive threshold value of the ratio for determining whether the signal attenuation correction is applied to each detected attenuation period. The ratio represents the average of the standard deviation at the detected attenuation peak relative to that at the beginning and end of the attenuation period. If the ratio is below this threshold value, the correction is applied. Default is 0.5.

`vctr_colname_feature`
A vector of characters indicating the name of the feature time series columns used in constructing a random forest model. If ‘NULL‘ (default), all columns excluding the label column specified as ‘colname_label‘ in the input data frame are used as feature columns.

`vctr_min_nodesize`
A vector of positive integers indicating candidates of a hyperparameter for the random forest model, defining the minimal node size (the minimum number of data points included in each leaf node). Default is ‘c(5)‘.

`vctr_m_try` A vector of positive integers indicating candidates of a hyperparameter for the random forest model, defining the number of features to be used in splitting each node. If ‘NULL‘ (default), integers between two and the number of all feature variables are tested.

`vctr_subsample_outlier`
A vector of numerical values between 0 and 1, indicating candidates of a hyperparameter for the random forest model, defining the fraction of input training data points to be sampled in constructing the random forest. Default is ‘c(0.1)‘.

`vctr_subsample_gf`
A vector of numerical values between 0 and 1, indicating candidates of a hyperparameter for the random forest model, defining the fraction of input training data points to be sampled in constructing the random forest. Default is ‘c(1)‘.

`frac_train` A numerical value between 0 and 1, defining the fraction of data points to be categorized as training data for the random forest model construction. The other data points are classified as test data. Default is 0.75.

`n_tree` An integer representing the number of trees in the random forest. Default is 500.

`ran_seed` An integer representing the random seed for the random forest model construction. Default is 12345.

`coef_iqr` A positive value defining a multiplier of the interquartile range (IQR). If the value to be checked is less than Q1 (first quartile) - ‘coef_iqr‘ * IQR or more than Q3 (third quartile) + ‘coef_iqr‘ * IQR, the value is detected as a random forest outlier. Default is 1.5.

`wndw_size_ref` A positive integer indicating the number of data points included in calculating the average and standard deviation for their reference value determination. The default is 48 * 15, meaning that the first 15 days of each sub-period are used in the calculation when the time interval of the input timestamp is 30 minutes.

`detrend` A boolean. If ‘TRUE‘, detrending is applied and the reference average is used to convert the Z-score time series to the time series in its original units; else, detrending is not applied, and the moving window average time series is used for the conversion. Default is ‘FALSE‘.

`correct_damping`
A boolean. If ‘TRUE‘, the signal damping correction is applied, and the reference standard deviation is used to convert the Z-score time series into the time series in its original units; else, the correction is not applied, and the moving window standard deviation time series is used in the conversion. Default is ‘FALSE‘.

`skip_sapflow_calc`
A boolean. If ‘TRUE‘, zero-flow condition estimation and sap flux density calculation are skipped in the whole process. This setting is useful when users want to post-process a time series unrelated to sap flow measurements. Default is ‘FALSE‘.

`colname_radi` A character representing the name of the column in the input data frame for global solar radiation or a similar radiative variable time series. Default is ‘NULL‘, but this name must be provided when ‘method‘ includes ‘pd‘, ‘mw‘, ‘dr‘, or ‘ed‘. Missing values in the column must be gap-filled previously. The unit of the time series must match that of ‘thres_radi‘.

`colname_ta` A character representing the name of the column in the input data frame for air temperature (degrees Celsius) time series. Default is ‘NULL‘, but this name must be provided when ‘method‘ includes ‘ed‘. Missing values must be gap-filled previously. The unit of the time series must match that of ‘thres_ta‘.

`colname_vpd` A character representing the name of the column in the input data frame for vapor pressure deficit (VPD, in hPa) time series. Default is ‘NULL‘, but this name must be provided when ‘method‘ includes ‘ed‘. Missing values must be gap-filled previously. The unit of the time series must match that of ‘thres_vpd‘.

`method` A vector of characters indicating the dTmax estimation methods. "sp", "pd", "mw", "dr", and "ed" represent the successive predawn, daily predawn, moving window, double regression, and environmental dependent method, respectively. Default is ‘c("sp")‘.

`thres_hour_sp` An integer from 0 to 23. The threshold hour of the day which defines the start of predawn in local time (default is 5).

`thres_radi` A threshold value of the radiation to define daytime. Default is 100 (W m-2). The data points with radiation values above the threshold are considered daytime values. The unit of the threshold must match that of the input radiation time series.

`thres_ta` A threshold value of the air temperature to define predawn. Default is 1.0 (degrees Celsius). The dTmax, estimated by the PD method, with air temperature values below the threshold, is selected as a candidate for the final dTmax. The unit of the threshold must match that of the input air temperature time series.

`thres_vpd` A threshold value of the VPD to define predawn. Default is 1.0 (hPa). The dTmax, estimated by the PD method, with VPD values below the threshold, is selected as a candidate for the final dTmax. The unit of the threshold must match that of the input VPD time series.

`thres_cv` A threshold value of the coefficient of variation (CV) to define predawn. Default is 0.005. The dTmax, estimated by the PD method, with CV values below the threshold, is selected as a candidate for the final dTmax.

`thres_hour_pd` An integer from 0 to 23. The threshold hour of the day which defines the end of predawn in local time (default is 8).

`min_n_wndw_dtmax`

A positive integer indicating the minimum number of data points for calculating statistics using a moving window (default is 3). If the number of data points is less than this threshold, the statistics are not calculated in the window.

`wndw_size_dtmax`

A positive integer indicating the window size (days) for determining moving window maximum values of dTmax. Default is 11 (days).

`alpha` A positive value representing a multiplier in the equation to calculate Fd. Default is 1.19 * 10^(-4) (m3 m-2 s-1).

`beta` A positive value representing a power in the equation to calculate Fd. Default is 1.231.

`do_heartwood_correction`

A boolean. If ‘TRUE‘, the heartwood correction is applied to correct dT before calculating Fd; else, the correction is not applied. Default is ‘FALSE‘.

`ratio_conductive`

A number between 0 and 1, indicating the ratio of the probe length to sapwood width. This parameter must be provided if ‘do_heartwood_correction‘ is ‘TRUE‘. Default is ‘NULL‘.

Details

This function executes a series of quality-control processes on the input time series. The protocol includes the absolute limit test, short-term drift correction, high-frequency noise filtering, outlier removal by Z-score and a random forest model, gap-filling using the data-driven model, detrending, signal attenuation correction, as well as zero-flow condition estimation, heartwood correction, and sap flux density calculation for thermal dissipation sap flow data. See more details in the vignettes: ‘browseVignettes("fluxfixer")‘, or in each step-by-step function help page.

Here are some tips helpful for users:

* If users want to do only quality control unrelated to sap flow calculation, set ‘skip_sapflow_calc‘ as ‘TRUE‘. Estimation of the zero-flow condition and calculation of sap flux density are skipped in this setting.

* If the input time series has sudden shifts of average and/or standard deviation due to various reasons, including sensor replacement, specify the end timings of these events in ‘vctr_time_prd_tail‘. The Z-score transformation gets applied to each sub-period defined by these timestamps, calculating a more reasonable Z-score time series.

* When processing sap flow data, it is highly recommended that users include a time series of vapor pressure deficit into the input data frame, as it typically controls stomatal opening. If the sap flow measurement is conducted in forests with high seasonality, such as deciduous forests, it is also recommended to include a time series for the amount of forest leaf (leaf area index).

Value

* The first column, ‘time‘, gives the same timestamp as the input timestamp specified by ‘colname_time‘.

* The second column, ‘raw‘, gives the same targeted time series specified by ‘colname_target‘.

* The third column, ‘processed‘, gives the post-processed targeted time series.

* The fourth column, ‘qc‘, gives a quality-control (QC) flag time series indicating the history of modifications to each data point. The flag is originally represented as 10-bit binary numbers, but is converted to decimal before being output. Each bit is set to 1 if the corresponding process is applied to the data point. From right to left, the number represents the process of initial missing value detection, manual error value removal, outlier removal by absolute limit, short-term drift correction, high-frequency noise filtering, Z-score outlier removal, random forest outlier removal, gap-filling, detrending, and signal damping correction.

If ‘skip_sapflow_calc‘ is ‘FALSE‘, the columns below are also output.

* The columns which have the prefix "dtmax_" provide the dTmax calculated by the methods specified in ‘method‘. The last two letters of the column name represent the name of the dTmax estimation method. "sp", "pd", "mw", "dr", and "ed" represent the successive predawn, daily predawn, moving window, double regression, and environmental dependent method, respectively.

* The columns which have the prefix "fd_" provide the Fd calculated using the dTmax estimated by the methods specified in ‘method‘. The last two letters of the column name represent the name of the dTmax estimation method.

Author(s)

Yoshiaki Hata

See Also

‘remove_manually‘, ‘check_absolute_limits‘, ‘modify_short_drift‘, ‘filter_highfreq_noise‘, ‘remove_zscore_outlier‘, ‘remove_rf_outlier‘, ‘calc_ref_stats‘, ‘fill_gaps‘, ‘retrieve_ts‘, ‘calc_dtmax‘, ‘calc_fd‘

Examples

```
## Load data
data("dt_noisy")
df_input <- dt_noisy[c(13105:13920), ]

## Run all processes automatically
result <-
  run_fluxfixer(df = df_input, colname_time = "time", colname_target = "dt",
                vctr_colname_feature = c("sw_in", "vpd", "swc"),
                skip_sapflow_calc = TRUE)
```

Index