Package ‘RFlux’

January 4, 2021

Type Package

Title Eddy Covariance Flux Data Processing

Version 1.1.0

Date 2020-12-29

Author Domenico Vitale [aut, cre],
Dario Papale [com, ctb]

Maintainer Domenico Vitale <domvit@unitus.it>

Description An R graphical user interface for processing eddy covariance raw data and release high quality fluxes of the main GHGs exchanged by ecosystems and agricultural fields. Fluxes are estimated through a call to the open source EddyPro software (registered trademark, LI-COR, Biosciences, 2020). 'RFlux' provides tools for the metadata management as well as for the implementation of the robust data cleaning procedure described by Vitale et al (2020) <doi:10.5194/bg-17-1367-2020>.

License GPL(>=2)

Imports stats, utils, bit64, data.table, imputeTS, robustbase, zoo, xts, stringr, stlplus, egcm, alphaOutlier, timeDate, car, robfilter, future.apply

Depends R (>= 3.5.0)

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

RFlux-package *Eddy Covariance Flux Data Processing*

Description

An R graphical user interface for processing eddy covariance raw data and release high quality fluxes of the main GHGs exchanged by ecosystems and agricultural fields. Fluxes are estimated through a call to the open source EddyPro software (registered trademark, LI-COR, Biosciences, 2020). 'RFlux' provides tools for the metadata management as well as for the implementation of the robust data cleaning procedure described by Vitale et al (2020) <doi:10.5194/bg-17-1367-2020>.

Details

RFlux package ingests eddy covariance rawdata sampled by either open- or closed-path system and implement the processing pipeline adopted by the ICOS-ETC (Integrated Carbon Observation System European Research Infrastructure - Ecosystem Thematic Center, `http://www.icos-etc.eu/icos/`). All metadata information have to be reported in the *filename*_ecmd.csv table. Such information are then processed by the `get_md` function.

The processing of rawdata aims at

i estimating fluxes and other micrometeorolgical parameters.

ii performing data quality control.

Flux estimation involves the following options/methods:

- Planar fit for tilt correction (Wilczak et al, 2001).
- Maximum cross-covariance method for time lag determination (see Rebmann et al, 2012).
- Block average (see Moncrieff et al, 2004).
- WPL correction, only for open path systems (Webb et al, 1980).
- In-situ spectral corrections (Fratini et al, 2012).

The open source EddyPro software (registered trademark, LI-COR Biosciences, 2019) is used to this aim employing also the estimation of micrometeorological parameters useful in subsequent analyses. It is required the EddyPro software is installed on your system (for download see `www.licor.com/EddyPro`).

Quality control involves the data cleaning procedure described in Vitale et al (2019). Its implementation involves a three-step procedure

Step 1: Estimation of the test statistics via the `qcStat` function.

Step 2: Generating the workset via the `ecworkset` function.

Step 3: Application of data cleaning procedure (including despiking) via the `cleanFlux` function.

Acknowledgements. Rflux has been developed in the context of the ICOS Ecosystem Thematic Centre. DV thanks the ENVRIPLUS H2020 European project (Grant Agreement 654182) for the support. DP thanks the ENVRIFAIR H2020 European project (Grant Agreement 824068) for the support.

Author(s)

Domenico Vitale [aut, cre], Dario Papale [com, ctb]

Maintainer: Domenico Vitale <domvit@unitus.it>

References

Fratini, G., Mauder, M. (2014). Towards a consistent eddy-covariance processing: an intercomparison of EddyPro and TK3. Atmospheric Measurement Techniques, 7(7), 2273-2281, doi: https://doi.org/10.5194/amt-7-2273-2014.

Fratini, G., Ibrom, A., Arriga, N., Burba, G., Papale, D. (2012). Relative humidity effects on water vapour fluxes measured with closed-path eddy-covariance systems with short sampling lines. Agricultural and forest meteorology, 165, pp 53-63, doi: https://doi.org/10.1016/j.agrformet.2012.05.018.

LI-COR Biosciences: EddyPro 7.0.4: Help and User's Guide, LI-COR Biosciences, Lincoln, Nebraska USA, www.licor.com/EddyPro, 2019.

Moncrieff, J., Clement, R., Finnigan, J., Meyers, T. (2004). Averaging, detrending, and filtering of eddy covariance time series. In Handbook of micrometeorology, pp. 7-31, Springer, Dordrecht, doi: https://doi.org/10.1007/1-4020-2265-4_2.

Rebmann, C., Kolle, O., Heinesch, B., Queck, R., Ibrom, A., Aubinet, M. (2012). Data acquisition and flux calculations. In Eddy covariance, pp. 59-83, Springer, Dordrecht.

Vitale, D. Fratini, G. Bilancia, M. Nicolini, G. Sabbatini, S. Papale, D. (2019). A robust data cleaning procedure for eddy covariance flux measurements, Biogeosciences, 17, 1367–1391, doi: https://doi.org/10.5194/bg-17-1367-2020, 2020.

Webb, E.K., Pearman, G.I., Leuning, R. (1980). Correction of flux measurements for density effects due to heat and water vapour transfer. Quarterly Journal of the Royal Meteorological Society, 106(447), pp 85-100, doi: https://doi.org/10.1002/qj.49710644707.

Wilczak, J.M., Oncley, S.P., Stage, S.A. (2001). Sonic anemometer tilt correction algorithms. Boundary-Layer Meteorology, 99(1), pp 127-150, doi: https://doi.org/10.1023/A:1018966204465.

`cleanFlux` *Cleaning eddy covariane flux measurements*

Description

This is the main function of the RFlux library. It performs the data cleaning procedure described by Vitale et al (2019).

Usage

```
cleanFlux(path_workset, path_ecmd, path_output=NULL, FileName=NULL,
          plotQC=FALSE, storage=FALSE)
```

Arguments

`path_workset`	path where the workset file generated from function `ecworkset` is stored.
`path_ecmd`	path where the eddy covariance metadata file (*CC-xxx*_ecmd.csv) is stored.
`path_output`	path where the output file will be stored. Default is NULL.

FileName — file name for the output file. Deafult is NULL.

plotQC — Logical. Should the details of data cleaning procedure be saved. If TRUE two .jpeg files (Details and Synthesis) for each flux variables will be stored in the path specified by path_output. Default is FALSE.

storage — Logical. Should the Net Ecosystem Exchange flux take into account the CO2 storage term. Default is FALSE.

Details

Comparing statistics estimated by qcStat with two threshold values, each test returns one of 3 possible statements:

SevEr: if the test provides strong evidence about the presence of a specific source of systematic error.

ModEr: if the test provides only weak evidence about the presence of a specific source of systematic error.

NoEr: if the test does not provide evidence about the presence of a specific source of systematic error.

The data cleaning is based on a two-step procedure. In the first stage, fluxes that inherited at least one SevEr statement are rejected, while fluxes that inherited no SevEr statements and any number of ModEr statements are retained. In the second stage, flux data that inherited no SevEr statement are subject to an outlier detection procedure and only flux data that are both detected as outlier and inherited at least a ModEr statement are conclusively rejected. This implies that data points that inherited any number of ModEr statements but were not detected to be outliers, as well as outliers which showed no evidence of systematic errors, are retained in the dataset and can be used for any analysis or modeling purposes.

Value

Returns a dataframe containing:

TIMESTAMP_START — ISO timestamp start of averaging period (format: *yyyymmddHHMM*).

TIMESTAMP_END — ISO timestamp end of averaging period (format: *yyyymmddHHMM*).

H_UNCLEANED — Sensible heat turbulent flux (no storage correction, uncleaned).

H — Sensible heat turbulent flux (no storage correction, cleaned).

H_DATA_FLAG — Flag for H (0: observed flux for which any quality control (QC) tests provided negligible evidences of error; 1: outlying flux rejected because at least one of the QC tests provided a moderate evidence of error; 2: flux removed because at least one of the QC test provided a severe evidence of error).

LE_UNCLEANED — Latent heat turbulent flux (no storage correction, uncleaned).

LE — Latent heat turbulent flux (no storage correction, cleaned).

LE_DATA_FLAG — Flag for LE (0: observed flux for which any quality control (QC) tests provided negligible evidences of error; 1: outlying flux rejected because at least one of the QC tests provided a moderate evidence of error; 2: flux removed because at least one of the QC test provided a severe evidence of error).

FC — Carbon Dioxide (CO2) turbulent flux (no storage correction).

SC — Carbon Dioxide (CO2) storage flux.

NEE_UNCLEANED — Net Ecosystem Exchange (uncleaned).

NEE
: Net Ecosystem Exchange (cleaned).

NEE_DATA_FLAG
: Flag for NEE (i.e., 0: observed flux for which any quality control (QC) tests provided negligible evidences of error; 1: outlying flux rejected because at least one of the QC tests provided a moderate evidence of error; 2: flux removed because at least one of the QC test provided a severe evidence of error).

H_OUTLYING_FLAG
: Flag for H denoting outliers (0: no outlying flux; 1: outlying flux).

LE_OUTLYING_FLAG
: Flag for LE denoting outliers (0: no outlying flux; 1: outlying flux).

NEE_OUTLYING_FLAG
: Flag for NEE denoting outliers (0: no outlying flux; 1: outlying flux).

H_FMR_STAT
: Fraction of Missing Records in raw, high-frequency, data used for H flux estimation.

H_FMR_FLAG
: Flag for the FMR test for H (0: negligible evidences of error, IF FMR<5; 1: moderate evidences of error, 1 IF 5≤FMR≤15; 2: severe evidences of error, 2 IF FMR>15).

H_LGD_STAT
: Longest Gap Duration in raw, high-frequency, data used for H flux estimation.

H_LGD_FLAG
: Flag for the LGD test for H (0: negligible evidences of error, IF LGD<90; 1: moderate evidences of error, IF 90≤LGD≤180; 2: severe evidences of error, IF LGD>180).

LE_FMR_STAT
: Fraction of Missing Records in raw, high-frequency, data used for LE flux estimation.

LE_FMR_FLAG
: Flag for the FMR test for LE (0: negligible evidences of error, IF FMR<5; 1, moderate evidences of error, IF 5≤FMR≤15; 2, severe evidences of error, IF FMR>15).

LE_LGD_STAT
: Longest Gap Duration in raw, high-frequency, data used for LE flux estimation.

LE_LGD_FLAG
: Flag for the LGD test for LE (0: negligible evidences of error, IF LGD<90; 1: moderate evidences of error, IF 90≤LGD≤180; 2: severe evidences of error, IF LGD>180).

FC_FMR_STAT
: Fraction of Missing Records in raw, high-frequency, data used for FC flux estimation.

FC_FMR_FLAG
: Flag for the FMR test for FC (0: negligible evidences of error, IF FMR<5; 1: moderate evidences of error, IF 5≤FMR≤15; 2: severe evidences of error, IF FMR>15).

FC_LGD_STAT
: Longest Gap Duration in raw, high-frequency, data used for FC flux estimation.

FC_LGD_FLAG
: Flag for the LGD test for FC (0: negligible evidences of error, IF LGD<90; 1: moderate evidences of error, IF 90≤LGD≤180; 2: severe evidences of error, IF LGD>180).

SA_DIAG_FLAG
: Flag for Sonic Anemometer (SA) instrumental diagnostics (0: negligible evidences of error; 2: severe evidences of error).

GA_DIAG_FLAG
: Flag for gas analyzer (GA) instrumental diagnostics (0: negligible evidences of error; 2: severe evidences of error).

WD
: Wind direction.

WSECT_FLAG
: Footprint quality flag indicating periods when wind was blowing from directions known to significantly affect the turbulent flow (0: negligible evidences of error; 2: severe evidences of error).

`H_LSR_STAT` Statistic of the Low Signal Resolution test for H.

`H_LSR_FLAG` Flag for the LSR test for H (0: negligible evidences of error, IF LSR_STAT>0.995; 1: moderate evidences of error, IF 0.99≤LSR_STAT≤0.995; 2: severe evidences of error, IF LSR_STAT<0.99).

`LE_LSR_STAT` Statistic of the Low Signal Resolution test for LE.

`LE_LSR_FLAG` Flag for the LSR test for LE (0: negligible evidences of error, IF LSR_STAT>0.995; 1: moderate evidences of error, IF 0.99≤LSR_STAT≤0.995; 2: severe evidences of error, IF LSR_STAT<0.99).

`FC_LSR_STAT` Statistic of the Low Signal Resolution test for FC.

`FC_LSR_FLAG` Flag for the LSR test for FC (0: negligible evidences of error, IF LSR_STAT>0.995; 1: moderate evidences of error, IF 0.99 ≤LSR_STAT≤0.995; 2: severe evidences of error, IF LSR_STAT<0.99).

`W_HF5_STAT` Statistic of the homogeneity test applied on vertical wind velocity fluctuations (percentage of data exceeding $\mu + / - 5\sigma$).

`W_HF5_FLAG` Flag for the homogeneity test applied on vertical wind velocity fluctuations (0: negligible evidences of error, IF HF5_STAT<2; 1: moderate evidences of error, IF 2≤HF5_STAT≤4; 2: severe evidences of error, IF HF5_STAT>4).

`W_HF10_STAT` Statistic of the homogeneity test applied on vertical wind velocity fluctuations (percentage of data exceeding $\mu + / - 10\sigma$).

`W_HF10_FLAG` Flag for the homogeneity test applied on vertical wind velocity fluctuations (0: negligible evidences of error, IF HF10_STAT<0.5; 1: moderate evidences of error, IF 0.5≤HF10_STAT≤1; 2: severe evidences of error, IF HF10_STAT>1).

`W_HD5_STAT` Statistic of the homogeneity test applied on differenced vertical wind velocity (percentage of data exceeding $\mu + / - 5\sigma$).

`W_HD5_FLAG` Flag for the homogeneity test applied on differenced vertical wind velocity (0: negligible evidences of error, IF HD5_STAT<2; 1: moderate evidences of error, IF 2≤HD5_STAT≤4; 2: severe evidences of error, IF HD5_STAT>4).

`W_HD10_STAT` Statistic of the homogeneity test applied on differenced vertical wind velocity (percentage of data exceeding $\mu + / - 10\sigma$).

`W_HD10_FLAG` Flag for the homogeneity test applied on differenced vertical wind velocity (0: negligible evidences of error, IF HD10_STAT<0.5; 1: moderate evidences of error, IF 0.5≤HD10_STAT≤1; 2: severe evidences of error, IF HD10_STAT>1).

`T_SONIC_HF5_STAT`
Statistic of the homogeneity test applied on sonic temperature fluctuations (percentage of data exceeding $\mu + / - 5\sigma$).

`T_SONIC_HF5_FLAG`
Flag for the homogeneity test applied on sonic temperature fluctuations (0: negligible evidences of error, IF HF5_STAT<2; 1: moderate evidences of error, IF 2≤HF5_STAT≤4; 2: severe evidences of error, IF HF5_STAT>4).

`T_SONIC_HF10_STAT`
Statistic of the homogeneity test applied on sonic temperature fluctuations (percentage of data exceeding $\mu + / - 10\sigma$).

`T_SONIC_HF10_FLAG`
Flag for the homogeneity test applied on sonic temperature fluctuations (0: negligible evidences of error, IF HF10_STAT<0.5; 1: moderate evidences of error, IF 0.5≤HF10_STAT≤1; 2: severe evidences of error, IF HF10_STAT>1).

`T_SONIC_HD5_STAT`
Statistic of the homogeneity test applied on differenced sonic temperature (percentage of data exceeding $\mu + / - 5\sigma$).

T_SONIC_HD5_FLAG
: Flag for the homogeneity test applied on differenced sonic temperature (0: negligible evidences of error, IF HD5_STAT<2; 1: moderate evidences of error, IF 2≤HD5_STAT≤4; 2: severe evidences of error, IF HD5_STAT>4).

T_SONIC_HD10_STAT
: Statistic of the homogeneity test applied on differenced sonic temperature (percentage of data exceeding $\mu + / - 10\sigma$).

T_SONIC_HD10_FLAG
: Flag for the homogeneity test applied on differenced sonic temperature (0: negligible evidences of error, IF HD10_STAT<0.5; 1: moderate evidences of error, IF 0.5≤HD10_STAT≤1; 2: severe evidences of error, IF HD10_STAT>1).

H2O_HF5_STAT
: Statistic of the homogeneity test applied on water vapor fluctuations (percentage of data exceeding $\mu + / - 5\sigma$).

H2O_HF5_FLAG
: Flag for the homogeneity test applied on water vapor fluctuations (0: negligible evidences of error, IF HF5_STAT<2; 1: moderate evidences of error, IF 2≤HF5_STAT≤4; 2: severe evidences of error, IF HF5_STAT>4).

H2O_HF10_STAT
: Statistic of the homogeneity test applied on water vapor fluctuations (percentage of data exceeding $\mu + / - 10\sigma$).

H2O_HF10_FLAG
: Flag for the homogeneity test applied on water vapor fluctuations (0: negligible evidences of error, IF HF10_STAT<0.5; 1: moderate evidences of error, IF 0.5≤HF10_STAT≤1; 2: severe evidences of error, IF HF10_STAT>1).

H2O_HD5_STAT
: Statistic of the homogeneity test applied on differenced water vapor (percentage of data exceeding $\mu + / - 5\sigma$).

H2O_HD5_FLAG
: Flag for the homogeneity test applied on differenced water vapor (0: negligible evidences of error, IF HD5_STAT<2; 1: moderate evidences of error, IF 2≤HD5_STAT≤4; 2: severe evidences of error, IF HD5_STAT>4).

H2O_HD10_STAT
: Statistic of the homogeneity test applied on differenced water vapor (percentage of data exceeding $\mu + / - 10\sigma$).

H2O_HD10_FLAG
: Flag for the homogeneity test applied on differenced water vapor (0: negligible evidences of error, IF HD10_STAT<0.5; 1: moderate evidences of error, IF 0.5≤HD10_STAT≤1; 2: severe evidences of error, IF HD10_STAT>1).

CO2_HF5_STAT
: Statistic of the homogeneity test applied on carbon dioxide fluctuations (percentage of data exceeding $\mu + / - 5\sigma$).

CO2_HF5_FLAG
: Flag for the homogeneity test applied on carbon dioxide fluctuations (0: negligible evidences of error, IF HF5_STAT<2; 1: moderate evidences of error, IF 2≤HF5_STAT≤4; 2: severe evidences of error, IF HF5_STAT>4).

CO2_HF10_STAT
: Statistic of the homogeneity test applied on carbon dioxide fluctuations (percentage of data exceeding $\mu + / - 10\sigma$).

CO2_HF10_FLAG
: Flag for the homogeneity test applied on carbon dioxide fluctuations (0: negligible evidences of error, IF HF10_STAT<0.5; 1: moderate evidences of error, IF 0.5≤HF10_STAT≤1; 2: severe evidences of error, IF HF10_STAT>1).

CO2_HD5_STAT
: Statistic of the homogeneity test applied on differenced carbon dioxide (percentage of data exceeding $\mu + / - 5\sigma$).

CO2_HD5_FLAG
: Flag for the homogeneity test applied on differenced carbon dioxide (0: negligible evidences of error, IF HD5_STAT<2; 1: moderate evidences of error, IF 2≤HD5_STAT≤4; 2: severe evidences of error, IF HD5_STAT>4).

CO2_HD10_STAT
: Statistic of the homogeneity test applied on differenced carbon dioxide (percentage of data exceeding $\mu + / - 10\sigma$).

`CO2_HD10_FLAG` Flag for the homogeneity test applied on differenced carbon dioxide (0: negligible evidences of error, IF HD10_STAT<0.5; 1: moderate evidences of error, IF 0.5≤HD10_STAT≤1; 2: severe evidences of error, IF HD10_STAT>1).

`W_KID_STAT` Kurtosis Index of Differenced vertical wind velocity.

`W_KID_FLAG` Flag for the W_KID_STAT (0: negligible evidences of error, IF KID_STAT<30; 1: moderate evidences of error, IF 30≤KID_STAT≤50; 2: severe evidences of error, IF KID_STAT>50).

`T_SONIC_KID_STAT`
Kurtosis Index of Differenced sonic temperature.

`T_SONIC_KID_FLAG`
Flag for the T_SONIC_KID_STAT (0: negligible evidences of error, IF KID_STAT<30; 1: moderate evidences of error, IF 30≤KID_STAT≤50; 2: severe evidences of error, IF KID_STAT>50).

`H2O_KID_STAT` Kurtosis Index of Differenced water vapor.

`H2O_KID_FLAG` Flag for the H2O_KID_STAT (0: negligible evidences of error, IF KID_STAT<30; 1: moderate evidences of error, IF 30≤KID_STAT≤50; 2: severe evidences of error, IF KID_STAT>50).

`CO2_KID_STAT` Kurtosis Index of Differenced carbon dioxide.

`CO2_KID_FLAG` Flag for the CO2_KID_STAT (0: negligible evidences of error, IF KID_STAT<30; 1: moderate evidences of error, IF 30≤KID_STAT≤50; 2: severe evidences of error, IF KID_STAT>50).

`ITC_STAT` Statistic of the Integral Turbulence Characteristics test (Foken and Wichura, 1996).

`ITC_FLAG` Flag for the ITC test (0: negligible evidences of error, IF ITC_STAT<30; 1: moderate evidences of error, IF 30≤ITC_STAT≤50; 2: severe evidences of error, IF ITC_STAT>50).

`H_SCF_STAT` Spectral correction factor for H.

`LE_SCF_STAT` Spectral correction factor for LE.

`FC_SCF_STAT` Spectral correction factor for FC.

`H_M98_STAT` Statistic of the nonstationarity ratio test by Mahrt (1998) for H.

`H_M98_FLAG` Flag of the H_M98_STAT (0: negligible evidences of error, IF M_98_STAT<2; 1: moderate evidences of error, IF 2≤M98_STAT≤3; 2: severe evidences of error, IF M98_STAT>3).

`LE_M98_STAT` Statistic of the nonstationarity ratio test by Mahrt (1998) for LE.

`LE_M98_FLAG` Flag of the LE_M98_STAT (0: negligible evidences of error, IF M_98_STAT<2; 1: moderate evidences of error, IF 2≤M98_STAT≤3; 2: severe evidences of error, IF M98_STAT>3).

`FC_M98_STAT` Statistic of the nonstationarity ratio test by Mahrt (1998) for FC.

`FC_M98_FLAG` Flag of the FC_M98_STAT (0: negligible evidences of error, IF M_98_STAT<2; 1: moderate evidences of error, IF 2≤M98_STAT≤3; 2: severe evidences of error, IF M98_STAT>3).

`CO2` Carbon Dioxide (CO2) in mole fraction of wet air.

`CO2_SIGMA` Standard deviation of carbon dioxide in mole fraction of wet air.

`H2O` Water (H2O) vapor mole fraction.

`H2O_SIGMA` Standard deviation of water vapor mole fraction.

`T_SONIC`	Sonic temperature.
`T_SONIC_SIGMA`	Standard deviation of sonic temperature.
`WS`	Wind speed.
`USTAR`	Friction velocity.
`W_SIGMA`	Standard deviation of vertical velocity fluctuations.
`U_SIGMA`	Standard deviation of lateral velocity fluctuations (towards main-wind direction after coordinates rotation).
`V_SIGMA`	Standard deviation of lateral velocity fluctuations (cross main-wind direction after coordinates rotation).
`ZL`	Monin-Obukhov stability parameter.
`MO_LENGTH`	Monin-Obukhov length.
`AT`	Air temperature.
`AP`	Air pressure.
`RHO`	Air density.
`CP`	Air heat capacity.
`CANOPY_HEIGHT`	Canopy height.
`SA_HEIGHT`	Sonic anemometer height.
`SA_NORTH_OFFSET`	Sonic anemometer north offset.
`INVALID_WIND_SECTOR_c1`	Center of the first invalid wind sector, if any.
`INVALID_WIND_SECTOR_w1`	Width of the first invalid wind sector, if any.
`INVALID_WIND_SECTOR_c2`	Center of the second invalid wind sector, if any.
`INVALID_WIND_SECTOR_w2`	Width of the second invalid wind sector, if any.
`INVALID_WIND_SECTOR_c3`	Center of the third invalid wind sector, if any.
`INVALID_WIND_SECTOR_w3`	Width of the third invalid wind sector, if any.

Author(s)

Domenico Vitale

References

Vitale, D. Fratini, G. Bilancia, M. Nicolini, G. Sabbatini, S. Papale, D. A robust data cleaning procedure for eddy covariance flux measurements, Biogeosciences, 17, 1367–1391, doi: https://doi.org/10.5194/bg-17-1367-2020, 2020.

Examples

```
PATH_WORKSET <- system.file("extdata", "ecworkset_example.csv", package = "RFlux")
PATH_ECMD <- system.file("extdata", "DE-HoH_ecmd.csv", package = "RFlux")

cleanset <- cleanFlux(path_workset=PATH_WORKSET,
                      path_ecmd=PATH_ECMD,
                      path_output=NULL,
                      FileName=NULL,
                      plotQC=FALSE,
                      storage=TRUE)
str(cleanset)
```

`closed_path_rawdata` *Raw, high-frequency, data for closed path systems*

Description

An example of raw, high-frequency, eddy covariance data for a closed path systems.

Usage

```
data("closed_path_rawdata")
```

Format

A data frame with 36000 observations of

- `U` Horizontal wind component along the Gill HS-50 sonic anemometer x axis (m/s).
- `V` Horizontal wind component along the Gill HS-50 sonic anemometer y axis (m/s).
- `W` Vertical wind component along the Gill HS-50 sonic anemometer z axis (m/s).
- `T_SONIC` Temperature measured by the Gill HS-50 sonic anemometer (kelvin).
- `CO2` Carbon dioxide atmospheric concentrations (mixing ratio) measured by the LI-7200 gas analyzer (ppm).
- `H2O` Water vapor atmospheric concentrations (mixing ratio) measured by the LI-7200 gas analyzer (ppt).
- `SA_DIAG` Diagnostic flag output by the Gill HS-50 sonic anemometer (dimensionless).
- `GA_DIAG` Diagnostic flag output by the LI-7200 gas analyzer (dimensionless).
- `T_CELL` Average cell temperature of the LI-7200 gas analyzer (celsius).
- `T_CELL_IN` Temperature at the inlet of the LI-7200 gas analyzer cell (celsius).
- `T_CELL_OUT` Temperature at the outlet of the LI-7200 gas analyzer cell (celsius).
- `PRESS_CELL` Cell pressure of the LI-7200 gas analyzer (Kpa).

Source

Data are from the Integrated Carbon Observation System (ICOS) European Research Infrastructure and accessible through the ICOS Carbon Portal http://www.icos-cp.eu.

Examples

```
data(closed_path_rawdata)
str(closed_path_rawdata)
```

`despiking` *Despiking algorithms*

Description

A suite of despiking algorithms for micro-meteorological variables

Usage

```
despiking(x, mfreq, variant, wsignal, wscale, wby = 1, zth = 5, alpha = 0.01)
```

Arguments

`x`	a vector of the observed time-series values.
`mfreq`	the main frequency of the observed time series (24 or 48 for hourly and half-hourly time series, respectively; 10 or 20 for raw EC data acquired at 10Hz or 20Hz, respectively).
`variant`	"v1" mainly designed for (half-)hourly EC fluxes (missing values are allowed); "v2" mainly designed for (half-)hourly meteo time series (a low percentage of missing values is allowed); "v3" mainly designed for high-frequency EC raw data.
`wsignal`	the window width used to estimate the underlying signal (only for met and raw variants).
`wscale`	the window width used to estimate the local scale paramater (only for met and raw variants).
`wby`	calculate the scale parameter at every by-th point rather than every point (default=1). Large values of wby reduce the computational time, but can introduce bias in the scale parameter estimation.
`zth`	the threshold value of the z-sigma rule ((only for met and raw variants).
`alpha`	the significance level used in the outlier detection rule (only for flx variant).

Value

A list containing the following components:

`despiked_ts`	A vector of the despiked time-series.
`spike_loc`	Integer. Location of the detected spikes.

Author(s)

Domenico Vitale

References

Fried, R. Schettlinger, K. Matthias Borowski, M. robfilter: Robust Time Series Filters. R package version 4.1.1. https://CRAN.R-project.org/package=robfilter, 2018.

Hafen, R. stlplus: Enhanced Seasonal Decomposition of Time Series by Loess. R package version 0.5.1. https://CRAN.R-project.org/package=stlplus, 2016.

Vitale, D. Fratini, G. Bilancia, M. Nicolini, G. Sabbatini, S. Papale, D. A robust data cleaning procedure for eddy covariance flux measurements, Biogeosciences, 17, 1367–1391, doi: https://doi.org/10.5194/bg-17-1367-2020, 2020.

Vitale, D. A perfomance evaluation of despiking algorithms for eddy covariance data. Nature Scientific Reports (in review).

`ecmd_table` *Eddy covariance metadata table*

Description

A collection of metadata useful for the proper settings of eddy covariance data processing options.

Usage

```
data("ecmd_table")
```

Format

A data frame with the following variables:

`DATE_OF_VARIATION_DB` Optional. ISO timestamp of the variation in central database. Required format: *yyyymmddHHMM*.

`DATE_OF_VARIATION_EF` ISO timestamp of effective date of variation. Required format: *yyyymmdd* or *yyyymmddHHMM*.

`SITEID` Character string indicative of the EC Site's ID (*CC-Xxx*).

`LATITUDE` Latitude in decimal degree

`LONGITUDE` Longitude in decimal degree

`ALTITUDE` Altitude in meters

`CANOPY_HEIGHT` Canopy height in meters

`SA_MANUFACTURER` Sonic anemometer manufacturer (see EddyPro manual.)

`SA_MODEL` Sonic anemometer model (see EddyPro manual.)

`SA_SW_VERSION` Sonic anemometer software version.

`SA_WIND_DATA_FORMAT` Format of the wind components (see EddyPro manual.)

`SA_NORTH_ALIGNEMENT` Sonic anemometer north alignment (see EddyPro manual.)

`SA_HEIGHT` Sonic anemometer height (see EddyPro manual.)

`SA_NORTH_OFFSET` Sonic anemometer north offset (see EddyPro manual.)

`SA_NORTH_MAGDEC` Sonic anemometer magnetic declination (see EddyPro manual.)

`SA_INVALID_WIND_SECTOR_c1` Center of the first invalid wind sector.

`SA_INVALID_WIND_SECTOR_w1` Width of the first invalid wind sector.

SA_INVALID_WIND_SECTOR_c2 Center of the second invalid wind sector.

SA_INVALID_WIND_SECTOR_w2 Width of the second invalid wind sector.

SA_INVALID_WIND_SECTOR_c3 Center of the third invalid wind sector.

SA_INVALID_WIND_SECTOR_w3 Width of the third invalid wind sector.

GA_PATH Specify the eddy covariance path system: closed or open.

GA_MANUFACTURER Gas analyzer manufacturer (see EddyPro manual).

GA_MODEL Gas analyzer model (see EddyPro manual).

GA_SW_VERSION Gas analyzer software version (see EddyPro manual).

GA_NORTHWARD_SEPARATION The distance between the center of the sample volume (or the inlet of the intake tube) of the current gas analyzer and the sonic anemometer, as measured horizontally along the north-south axis (see EddyPro manual).)

GA_EASTWARD_SEPARATION The distance between the center of the sample volume (or the inlet of the intake tube) of the current gas analyzer and the sonic anemometer, as measured horizontallu along the east-west axis (see EddyPro manual).)

GA_VERTICAL_SEPARATION The distance between the center of the sample volume (or the inlet of the intake tube) of the current gas analyzer and the sonic anemometer, as measured vertically (see EddyPro manual).

GA_TUBE_DIAMETER The inside diameter of the intake tube in centimeters. Mandatory for closed path system.

GA_FLOWRATE The flow rate in the intake tube. Mandatory for closed path system.

GA_TUBE_LENGTH The length of the intake tube in centimeters.

FILE_DURATION File lenght duration in minutes

ACQUISITION_FREQUENCY Number of sample records per second.

FILE_FORMAT ASCII (fixed)

FILE_EXTENSION csv (fixed)

LN 99 (fixed)

FN 1 (fixed)

EXTERNAL_TIMESTAMP END (fixed)

EOL crlf (fixed)

SEPARATOR comma (fixed)

MISSING_DATA_STRING -9999 (fixed)

NROW_HEADER 1 (fixed)

UVW_UNITS Units of wind speed components (see EddyPro manual).

T_SONIC_UNITS Units of sonic temperature (see EddyPro manual).

T_CELL_UNITS Units of closed path gas analyzer cell temperature (see EddyPro manual).

P_CELL_UNITS Units of closed path gas analyzer cell pressure (see EddyPro manual).

CO2_measure_type Carbon dioxide concentration measurement type (e.g. mixing_ratio, see EddyPro manual).

CO2_UNITS Units of carbon dioxide concentration (e.g. ppm, see EddyPro manual).

H2O_measure_type Water vapor concentration measurement type (e.g. mixing_ratio, see EddyPro manual).

H2O_UNITS Units of water vapor concentration (e.g. ppt, see EddyPro manual).

SA_DIAG Units of the sonic anemometer diagnostics (dimensionless, fixed)

GA_DIAG Units of the gas analyzer diagnostics (dimensionless (fixed).

Source

Metadata elaborated by ICOS-ETC team (Integrated Carbon Observation System (ICOS-ETC) European Research Infrastructure - Ecosystem Thematic Center, http://www.icos-etc.eu/icos/).

References

LI-COR Biosciences: EddyPro 7.0.4: Help and User's Guide, LI-COR Biosciences, Lincoln, Nebraska USA, www.licor.com/EddyPro, 2019.

Examples

```
data(ecmd_table)
str(ecmd_table)
```

---

ecworkset — *Merge time series and returns the workset dataframe*

Description

Merge time series with common indexes (times) and returns the *workset* data frame to be used as input for the data cleaning procedure via `cleanFlux` function.

Usage

```
ecworkset(path_EPout, path_EPqc, path_EPmd, path_QCstat, path_output=NULL, FileName=NULL)
```

Arguments

`path_EPout`	path where the fulloutput file generated by EddyPro software is stored.
`path_EPqc`	path where the qcdetails file generated by EddyPro software is stored.
`path_EPmd`	path where the metadata file generated by EddyPro software is stored.
`path_QCstat`	path where the QC statistics file generated by `qcStat` is stored.
`path_output`	path where the results will be stored. Default is NULL.
`FileName`	a character string naming a file for writing. Default is NULL.

Details

Returns a dataframe ordered by common indexes (times) containing a set of variables selected from the fulloutput, metadata, qcdetails files generated by EddyPro via `eddypro_run` and the statistics of the quality control routines (Vitale et al, 2019) generated by `qcStat`.

Value

A dataframe object to be used as input of the `cleanFlux`. For the meaning of variables and units see the EddyPro manual and the `qcStat` function description.

Author(s)

Domenico Vitale, Dario Papale

References

LI-COR Biosciences: EddyPro 7.0.4: Help and User's Guide, LI-COR Biosciences, Lincoln, Nebraska USA, www.licor.com/EddyPro, 2019.

Vitale, D. Fratini, G. Bilancia, M. Nicolini, G. Sabbatini, S. Papale, D. A robust data cleaning procedure for eddy covariance flux measurements, Biogeosciences, 17, 1367–1391, doi: https://doi.org/10.5194/bg-17-1367-2020, 2020.

Examples

```
PATH_EPOUT <- system.file("extdata", "eddypro_DE-HoH_full_output_example.csv", package = "RFlux")
PATH_EPQC <- system.file("extdata", "eddypro_DE-HoH_qc_details_example.csv", package = "RFlux")
PATH_EPMD <- system.file("extdata", "eddypro_DE-HoH_metadata_example.csv", package = "RFlux")
PATH_QCSTAT <- system.file("extdata", "qcStat_example.csv", package = "RFlux")

WorkSet <- ecworkset(path_EPout=PATH_EPOUT,
                     path_EPqc=PATH_EPQC,
                     path_EPmd=PATH_EPMD,
                     path_QCstat=PATH_QCSTAT,
                     path_output=NULL,FileName=NULL)
str(WorkSet)
```

---

eddypro_run *Fluxes estimation*

Description

Estimates flux values and other micrometeorological parameters through a call to LI-COR EddyPro software.

Usage

```
eddypro_run(siteID, path_eddypro_bin, path_eddypro_projfiles, showLOG = TRUE)
```

Arguments

siteID	Character string indicative of the site's ID (i.e. CC-Xxx)
path_eddypro_bin	path where eddypro_rp and eddypro_fcc executables (i.e. the bin folder) are stored.
path_eddypro_projfiles	path where the EddyPro project files (i.e. the *filename*.eddypro file) is stored.
showLOG	logical. If TRUE (default), the EddyPro Output Console Page is shown on R console.

Value

A set of .csv files (e.g. fulloutput, qcdetails and metadata) generated by EddyPro software will be stored in the path specified by get_md.

Warning

It is required LI-COR EddyPro software is currently installed on your system (see www.licor.com/EddyPro).

Author(s)

Domenico Vitale, Dario Papale

References

LI-COR Biosciences: EddyPro 7.0.4: Help and User's Guide, LI-COR Biosciences, Lincoln, Nebraska USA, www.licor.com/EddyPro, 2019.

Examples

```
## Not run:
workdir <- getwd()
siteID <- "DE-HoH"
PATH_ECMD <- system.file("extdata", "DE-HoH_ecmd.csv", package = "RFlux")
PATH_RAWDATA <- system.file("extdata", package = "RFlux")
PATH_OUTPUT <- paste0(workdir, "/eddypro/processing")
PATH_SA_FILE <- system.file("extdata", "spectral_assessment.txt", package = "RFlux")
PATH_PF_FILE <- system.file("extdata", "planar_fit.txt", package = "RFlux")

MD <- get_md(path_ecmd=PATH_ECMD,
             path_rawdata=PATH_RAWDATA,
             path_output=PATH_OUTPUT,
             online=TRUE,
             path_sa_file=PATH_SA_FILE,
             path_pf_file=PATH_PF_FILE,
             tlag_meth=2,
             despike_meth="VM97",
             detrend_meth="BA",
             tilt_correction_meth="PF")

PATH_EDDYPRO_BIN <- "/Applications/eddypro.app/Contents/MacOS/bin" ## put your path!
PATH_EDDYPRO_PROJ <- PATH_OUTPUT
eddypro_run(siteID="DE-HoH",
            path_eddypro_bin=PATH_EDDYPRO_BIN,
            path_eddypro_projfiles=PATH_EDDYPRO_PROJ,
            showLOG=TRUE)

## End(Not run)
```

---

get_md *Metadata file management*

Description

Returns the input files required by EddyPro software (LI-COR registered trademark; Fratini and Mauder, 2014).

Usage

```
get_md(path_ecmd = NULL,
       path_rawdata = NULL,
       path_output = NULL,
       online,
       path_sa_file = NULL,
       path_pf_file = NULL,
       tlag_meth,
       despike_meth=c("None", "VM97", "M13"),
       detrend_meth=c("BA", "LD"),
       tilt_correction_meth=c("DR", "PF"))
```

Arguments

`path_ecmd`	path where the CC-Xxx_ecmd.csv file containing metadata information is stored.
`path_rawdata`	path where eddy covariance rawdata files are stored.
`path_output`	path where the results of `eddypro_run` will be stored (it will be created, if any).
`online`	logical. Indicating whether the parameters of the planar fit method and the spectral correction factors are taken from results of previous processing (online=TRUE) or estimated by using the current set of EC rawdata (online=FALSE).
`path_sa_file`	path where the spectral assessment file generated by EddyPro software is stored.
`path_pf_file`	path where the planar fit file generated by EddyPro software is stored.
`tlag_meth`	time lag detection method: 0 None; 1 Constant time lag, 2 Maximum covariance with default; 3 Maximum covariance; 4 Automatic optimization.
`despike_meth`	despiking method: "None", not apply; "VM97", algorithm by Vickers and Mahrt (1997); "M13", algorithm by Mauder et al (2013).
`detrend_meth`	trend removal method: "BA" Block Average; "LD" Linear Detrending (Rannik and Vesala, 2001).
`tilt_correction_meth`	axis rotation method for tilt correction: "DR" Double Rotation; "PF" Planar Fit (Wilczak et al, 2001).

Details

This function builds the input files required by EddyPro software to process eddy covariance rawdata files.

Value

Returns the following files

`CC-Xxx.eddypro`

`CC-Xxx.metadata`

`CC-Xxx_dynamic_metadata.txt`

where CC-Xxx denoting the site's ID.

Warning

If online=FALSE, the number of rawdata needs to be large enough to allow robust estimates of the planar fit parameters (Wilczak et al 2001) and of the spectral correction factors (Fratini et al, 2012). In case of few rawdata, the double rotation method for sonic anemometer tilt correction and the analytical method by Moncrieff et al (1997) for spectral correction will be performed.

Note

Rawdata must be provided as .csv file with the following name: *CC-Xxx????yyyymmddHHMM????.csv*, where *CC-Xxx* is the site's ID, *yyyymmddHHMM* is the ISO timestamp, and ?s denoting free characters. For example: DE-HoH_EC_201901010030_v01.csv.

Author(s)

Domenico Vitale, Dario Papale

References

Fratini, G., Mauder, M. (2014). Towards a consistent eddy-covariance processing: an intercomparison of EddyPro and TK3. Atmospheric Measurement Techniques, 7(7), 2273-2281, doi: https://doi.org/10.5194/amt-7-2273-2014.

Fratini, G., Ibrom, A., Arriga, N., Burba, G., Papale, D. (2012). Relative humidity effects on water vapour fluxes measured with closed-path eddy-covariance systems with short sampling lines. Agricultural and forest meteorology, 165, pp 53-63, doi: https://doi.org/10.1016/j.agrformet.2012.05.018.

LI-COR Biosciences: EddyPro 7.0.4: Help and User's Guide, LI-COR Biosciences, Lincoln, Nebraska USA, www.licor.com/EddyPro, 2019.

Mauder, M., Cuntz, M., Drue, C., Graf, A., Rebmann, C., Schmid, H. P., ..., Steinbrecher, R. (2013). A strategy for quality and uncertainty assessment of long-term eddy-covariance measurements. Agricultural and Forest Meteorology, 169, 122-135, doi: https://doi.org/10.1016/j.agrformet.2012.09.006.

Moncrieff, J. B., Massheder, J. M., De Bruin, H., Elbers, J., Friborg, T., Heusinkveld, B., ..., Verhoef, A. (1997). A system to measure surface fluxes of momentum, sensible heat, water vapour and carbon dioxide. Journal of Hydrology, 188, 589-611, doi: https://doi.org/10.1016/S0022-1694(96)03194-0.

Rannik, U., Vesala, T. (1999). Autoregressive filtering versus linear detrending in estimation of fluxes by the eddy covariance method. Boundary-Layer Meteorology, 91(2), 259-280, doi: https://doi.org/10.1023/A:1001840416858.

Vickers, D., Mahrt, L. (1997). Quality control and flux sampling problems for tower and aircraft data. Journal of atmospheric and oceanic technology, 14(3), 512-526, doi: https://doi.org/10.1175/1520-0426(1997)014<0512:QCAFSP>2.0.CO;2.

Wilczak, J.M., Oncley, S.P., Stage, S.A. (2001). Sonic anemometer tilt correction algorithms. Boundary-Layer Meteorology, 99(1), 127-150, doi: https://doi.org/10.1023/A:1018966204465.

See Also

closed_path_rawdata

Examples

```
## Not run:
workdir <- getwd()
siteID <- "DE-HoH"
PATH_ECMD <- system.file("extdata", "DE-HoH_ecmd.csv", package = "RFlux")
PATH_RAWDATA <- system.file("extdata", package = "RFlux")
PATH_OUTPUT <- paste0(workdir, "/eddypro/processing")
PATH_SA_FILE <- system.file("extdata", "spectral_assessment.txt", package = "RFlux")
PATH_PF_FILE <- system.file("extdata", "planar_fit.txt", package = "RFlux")

MD <- get_md(path_ecmd=PATH_ECMD,
             path_rawdata=PATH_RAWDATA,
             path_output=PATH_OUTPUT,
             online=TRUE,
             path_sa_file=PATH_SA_FILE,
             path_pf_file=PATH_PF_FILE,
             tlag_meth=2,
             despike_meth="VM97",
             detrend_meth="BA",
             tilt_correction_meth="PF")

## End(Not run)
```

`inst_prob_test` *Instrumental problem detection*

Description

A set of tests aims at detecting instrumental malfunctions affecting eddy covariance systems.

Usage

```
inst_prob_test(x)
```

Arguments

x	raw high frequency eddy covariance time series.

Value

Skew	Skewness.
Kurt	Kurtosis index on original data.
KID0	Kurtosis index on difference data.
KID1	Kurtosis index on difference data excluding low resolution problems.
HFx	Homogeneity test statistic based on fluctuation.
HDx	Homogeneity test statistic based on difference data.

Author(s)

Domenico Vitale

References

Vitale, D. Fratini, G. Bilancia, M. Nicolini, G. Sabbatini, S. Papale, D. A robust data cleaning procedure for eddy covariance flux measurements, Biogeosciences, 17, 1367–1391, doi: https://doi.org/10.5194/bg-17-1367-2020, 2020.

Examples

```
data(closed_path_rawdata)
inst_prob_test(closed_path_rawdata$W)
```

---

mahrt   *Stationary test for eddy covariance fluxes*

Description

Performs the non-stationary ratio test described by Mahrt (1998)

Usage

```
mahrt(x)
```

Arguments

x	two-column dataframe containing raw high-frequency time series of vertical wind component (i.e. W) and scalar atmospheric variable (e.g. CO2).

Value

Returns the non-stationary ratio test statistic.

Author(s)

Domenico Vitale

References

Mahrt L (1998) Flux sampling errors for aircraft and towers, J. Atmos. Ocean. Tech., 15, 416-429, https://doi.org/10.1175/1520-0426(1998)015<0416:fsefaa>2.0.co;2.

Examples

```
PATH_RAWDATA <- system.file("extdata", "DE-HoH_EC_201907301200_v01.csv", package = "RFlux")
data(closed_path_rawdata)
mahrt(data.frame("W"=closed_path_rawdata$W,"CO2"=closed_path_rawdata$CO2))
```

parallel_despiking — *Despiking eddy covariance raw data*

Description

A parallel implementation of the despiking algorithm for high-frequency eddy covariance raw data

Usage

```
parallel_despiking(x, mfreq, wsignal = mfreq*30+1, wscale=mfreq*30+1, wby = 1, zth = 5)
```

Arguments

`x`	a vector of the observed time-series values.
`mfreq`	the main frequency of the observed time series (24 or 48 for hourly and half-hourly time series, respectively; 10 or 20 for raw EC data acquired at 10Hz or 20Hz, respectively).
`wsignal`	the window width used to estimate the underlying signal (only for met and raw variants).
`wscale`	the window width used to estimate the local scale paramater (only for met and raw variants).
`wby`	calculate the scale parameter at every by-th point rather than every point (default=1). Large values of wby reduce the computational time, but can introduce bias in the scale parameter estimation.
`zth`	the threshold value of the z-sigma rule ((only for met and raw variants).

Value

A list containing the following components:

`despiked_ts`	A vector of the despiked time-series.
`spike_loc`	Integer. Location of the detected spikes.

Author(s)

Domenico Vitale

References

Fried, R. Schettlinger, K. Matthias Borowski, M. robfilter: Robust Time Series Filters. R package version 4.1.1. https://CRAN.R-project.org/package=robfilter, 2018.

Vitale, D. A perfomance evaluation of despiking algorithms for eddy covariance data. Nature Scientific Reports (in review).

qcStat *Quality control tests for eddy covariance fluxes*

Description

Returns the test statistics of the quality control routines described by Vitale et al (2019).

Usage

```
qcStat(path_rawdata, ext_tstamp=c("START", "END"), path_output=NULL, FileName=NULL)
```

Arguments

path_rawdata	path where raw high-frequency eddy covariance data are stored.
ext_tstamp	Character string specifying whether the timestamp in the file name is indicative of the beginning ("START") or the end ("END") of time series involved.
path_output	path where the results will be stored. Default is NULL.
FileName	a character string naming a file for writing. Default is NULL.

Value

A data frame containing:

TSTAMP	ISO timestamp start of averaging period (format: *yyyymmddHHMM*).
SADiag	Diagnostic for the sonic anemometer.
FMR_X	Fraction of Missing Records in raw, high-frequency, data used for X flux variable estimation.
LGD_X	Longest Gap Duration in raw, high-frequency, data used for X flux variable estimation.
Skew_X	Skewness of X variable.
Kurt_X	Kurtosis of X variable.
KID0_X	Kurtosis Index of Differenced X variable.
KID1_X	Kurtosis Index of Differenced X variable excluding the effect of possible low resolution problems.
HF5_X	Statistic of the homogeneity test applied on X variable fluctuations (percentage of data exceeding $+/-5\sigma$).
HF10_X	Statistic of the homogeneity test applied on X variable fluctuations (percentage of data exceeding $+/-10\sigma$).
HD5_X	Statistic of the homogeneity test applied on differenced X variable (percentage of data exceeding $+/-5\sigma$).
HD10_X	Statistic of the homogeneity test applied on differenced X variable (percentage of data exceeding $+/-10\sigma$).
COV_XY	Covariance between X and Y variables.
N0_X	Number of repeated consecutive values affecting X flux variable.
LSR_X	Statistic of the Low Signal Resolution test for X flux variable.
M98_X	Statistic of the non-stationary ratio test by Mahrt (1988) for X flux variable.

Note

The output of qcStat constiutes one of input files required by the ecworkset function.

Author(s)

Domenico Vitale

References

Mahrt L (1998) Flux sampling errors for aircraft and towers, J. Atmos. Ocean. Tech., 15, 416-429, https://doi.org/10.1175/1520-0426(1998)015<0416:fsefaa>2.0.co;2.

Vitale, D. Fratini, G. Bilancia, M. Nicolini, G. Sabbatini, S. Papale, D. A robust data cleaning procedure for eddy covariance flux measurements, Biogeosciences, 17, 1367–1391, doi: https://doi.org/10.5194/bg-17-1367-2020, 2020.

Examples

```
PATH_RAWDATA <- system.file("extdata", "DE-HoH_EC_201907301200_v01.csv", package = "RFlux")
QC_STAT <- qcStat(path_rawdata=PATH_RAWDATA, ext_tstamp="END", path_output=NULL, FileName=NULL)
QC_STAT
```

---

tlag_detection *Time-lag detection*

Description

Optimal time-lag detection by means of the pre-whitening procedure.

Usage

```
tlag_detection(x, y, mfreq, x.model = ar.res,
               AIC = FALSE, LAG.MAX = mfreq*10,
               show.plot = FALSE)
```

Arguments

x	first component series (e.g. CO2 concentrations).
y	second component series (e.g. vertical wind speed).
mfreq	acquisition frequency.
x.model	an ARIMA model; if provided, it is used to prewhiten both series. Otherwise, an AR model is fitted to the x-series and used to pre-whiten both series.
AIC	Logical. If an AR model is used for prewhitening, the AR order is chosen by minimizing the Akaike Information Criteria (AIC=TRUE), otherwise the AR model is chosen as 10*log10(N) where N is the number of observations. Default is FALSE.
LAG.MAX	maximum lag at which to calculate the ccf. Default is 10 seconds.
show.plot	Logical. If TRUE, the CCFs will be displayed (default FALSE).

Value

A list containing the following components:

ccf	Output from the ccf function on the prewhitened data.
ar	The AR model fit to the x-series, or x.model if it is provided.
opt_tlag	Optimal time lag detected (timestep).
tlag_pw	Time lag of the main peak of the cross-correlation function after pre-whitening.
tlag_lmax	Local maxima of the cross-covariance function closest to tlag_pw.
tlag_lmin	Local minima of the cross-covariance function closest to tlag_pw.
corr_est	Estimated correlation at .
cv1pct	Critical (abs) value of the ccf at 0.01 level.
cv5pct	Critical (abs) value of the ccf at 0.05 level.

Author(s)

Domenico Vitale

References

Vitale et al. (in prep) Optimal Time Lag Detection for Eddy Covariance Data Acquisition Systems

See Also

See also prewhiten function in the TSA R package.

Examples

```
## Not run:
library(RFlux)
library(forecast)
library(xts)
data("closed_path_rawdata")
rawdata <- closed_path_rawdata
N <- nrow(rawdata)
timestamp_orig <- strptime("0000.00", format="%H%M.%OS", tz="GMT") + seq(1, N, 1);
data.xts <- xts(rawdata, order.by=timestamp_orig);

## Detecting time lag betweem H2O and vertical wind speed (W) by using an AR model (default)
tlag_h2o_out <- tlag_detection(x=rawdata$H2O, y=rawdata$W, mfreq=20,
                               show.plot=TRUE)
tlag_h2o_out$opt_tlag
tlag_h2o_out$corr_est

## Detecting time lag between CO2 and Sonic Temperature by using an ARIMA model
mod <- auto.arima(rawdata$CO2, stationary=FALSE)
tlag_co2_out <- tlag_detection(x=rawdata$CO2, y=rawdata$T_SONIC, mfreq=20,
                               show.plot=TRUE, x.model=mod)
tlag_co2_out$opt_tlag
tlag_co2_out$corr_est

## Time series alignment
tlag_co2 <- tlag_co2_out$opt_tlag
tlag_h2o <- tlag_h2o_out$opt_tlag
```

```
timestamp_co2 <- strptime("0000.00", format="%H%M.%OS", tz="GMT") +
                 seq(-tlag_co2+1, N - tlag_co2, 1);
timestamp_h2o <- strptime("0000.00", format="%H%M.%OS", tz="GMT") +
                 seq(-tlag_h2o+1, N - tlag_h2o, 1);

co2s <- xts(rawdata$CO2, order.by=timestamp_co2);
h2os <- xts(rawdata$H2O, order.by=timestamp_h2o);
rawdata_aligned <- na.omit(data.frame(merge(data.xts[,c(1:4)],
                  co2s, h2os, data.xts[,c(7:12)], tzone="GMT")));
colnames(rawdata_aligned) <- c("U", "V", "W", "T_SONIC", "CO2", "H2O",
                               "SA_DIAG", "GA_DIAG",
                               "T_CELL", "T_CELL_IN", "T_CELL_OUT", "PRESS_CELL")

## End(Not run)
```

Index