Package 'REddyProc'

May 17, 2018

Type Package

Version 1.1.5

Title Post Processing of (Half-)Hourly Eddy-Covariance Measurements

Description Standard and extensible Eddy-Covariance data post-processing includes
uStar-filtering, gap-filling, and flux-partitioning.
The Eddy-Covariance (EC) micrometeorological technique quantifies continuous
exchange fluxes of gases, energy, and momentum between an ecosystem and the atmosphere.
It is important for understanding ecosystem dynamics and upscaling exchange fluxes.
(Aubinet et al. (2012) <doi:10.1007/978-94-007-2351-1>).
This package inputs pre-processed (half-)hourly data and supports further processing.
First, a quality-check and filtering is performed based on the relationship between
measured flux and friction
velocity (uStar) to discard biased data
(Papale et al. (2006) <doi:10.5194/bg-3-571-2006>).
Second, gaps in the data are filled based on information from environmental conditions
(Reichstein et al. (2005) <doi:10.1111/j.1365-2486.2005.001002.x>).
Third, the net flux of carbon dioxide is partitioned
into its gross fluxes in and out of the ecosystem by night-time
based and day-time based approaches
(Lasslop et al. (2010) <doi:10.1111/j.1365-2486.2009.02041.x>).

URL https://www.bgc-jena.mpg.de/bgi/index.php/Services/REddyProcWeb,
https://github.com/bgctw/REddyProc

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

VignetteBuilder knitr

LinkingTo Rcpp

Depends R (>= 3.0.0), methods

Imports dplyr, purrr, rlang, mlegp, tibble

Suggests testthat, minpack.lm, segmented, knitr, rmarkdown

Collate 'CheckVal.R' 'DataFunctions.R' 'aEddy.R' 'EddyGapfilling.R' 'EddyPartitioning.R' 'EddyPlotting.R' 'EddyUStarFilterChangePointDetection.R' 'EddyUStarFilterDP.R' 'Example.R' 'FileHandling.R' 'FileHandlingFormats.R' 'GeoFunctions.R' 'LRC_base.R' 'LRC_logisticSigmoid.R' 'LRC_nonrectangular.R' 'LRC_rectangular.R' 'PartitioningLasslop10.R' 'PartitioningLasslop10Nighttime.R' 'RcppExports.R' 'imports.R' 'logitnorm.R' 'variableNames.R' 'zzzDebugCode.R'

NeedsCompilation yes

Author Department for Biogeochemical Integration at MPI-BGC, Jena, Germany [cph],
Thomas Wutzler [aut, cre],
Markus Reichstein [aut],
Antje Maria Moffat [aut, trl],
Olaf Menzer [ctb],
Mirco Migliavacca [aut],
Kerstin Sickel [ctb, trl],
Ladislav Šigut [ctb]

Maintainer Thomas Wutzler <twutz@bgc-jena.mpg.de>

Repository CRAN

Date/Publication 2018-05-17 17:37:50 UTC

R topics documented:

`REddyProc-package` *Post Processing of (Half-)Hourly Eddy-Covariance Measurements*

Description

Standard and extensible Eddy-Covariance data post-processing includes uStar-filtering, gap-filling, and flux-partitioning.

The Eddy-Covariance (EC) micrometeorological technique quantifies continuous exchange fluxes of gases, energy, and momentum between an ecosystem and the atmosphere. It is important for understanding ecosystem dynamics and upscaling exchange fluxes. (Aubinet et al. (2012) <doi:10.1007/978-94-007-2351-1>).

This package inputs pre-processed (half-)hourly data and supports further processing. First, a quality-check and filtering is performed based on the relationship between measured flux and friction velocity (uStar) to discard biased data (Papale et al. (2006) <doi:10.5194/bg-3-571-2006>).

Second, gaps in the data are filled based on information from environmental conditions (Reichstein et al. (2005) <doi:10.1111/j.1365-2486.2005.001002.x>).

Third, the net flux of carbon dioxide is partitioned into its gross fluxes in and out of the ecosystem by night-time based and day-time based approaches (Lasslop et al. (2010) <doi:10.1111/j.1365-2486.2009.02041.x>).

A general description and an online tool based on this package can be found here: https://www.bgc-jena.mpg.de/bgi/index.php/Services/REddyProcWeb.

Details

A **detailed example** of the processing can be found in the useCase vignette.

A first overview of the REddyProc functions:

These functions help with the preparation of your data for the analysis:

- Loading text files into dataframes: `fLoadTXTIntoDataframe`
- Preparing a proper time stamp: `fConvertTimeToPosix`
- Calculating latent variables, e.g. VPD: `fCalcVPDfromRHandTair`

Then the data can be processed with the `sEddyProc-class` R5 reference class:

- Initializing the R5 reference class: `sEddyProc_initialize`
- Estimating the turbulence criterion, Ustar threshold, for omitting data from periods of low turbulence: Functions `sEddyProc_sEstUstarThreshold` and `sEddyProc_sEstUstarThresholdDistribution` .
- Gap filling: `sEddyProc_sMDSGapFill` and `sEddyProc_sMDSGapFillAfterUstar`.
- Flux partitioning based on Night-Time: `sEddyProc_sMRFluxPartition`
- Flux partitioning based on Day-Time: `sEddyProc_sGLFluxPartition`

Before or after processing, the data can be plotted:

- Fingerprint: `sEddyProc_sPlotFingerprint`
- Half-hourly fluxes and their daily means: `sEddyProc_sPlotHHFluxes`
- Daily sums (and their uncertainties): `sEddyProc_sPlotDailySums`
- Diurnal cycle: `sEddyProc_sPlotDiurnalCycle`

A **complete list** of REddyProc functions be viewed by clicking on the **Index** link at the bottom of this help page.

Also have a look at the package vignettes.

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany

References

Reichstein M, Falge E, Baldocchi D et al. (2005) On the separation of net ecosystem exchange into assimilation and ecosystem respiration: review and improved algorithm. Global Change Biology, 11, 1424-1439.

BerkeleyJulianDateToPOSIXct

BerkeleyJulianDateToPOSIXct

Description

convert JulianDate format used in Berkeley release to POSIXct

Usage

```
BerkeleyJulianDateToPOSIXct(julianDate, tz = "GMT",
    ...)
```

Arguments

julianDate	numeric vector representing times (see details for format)
tz	time zone used to represent the dates
...	further arguments to strptime, such as tz

Details

In the Berkeley-Release of the fluxnet data, the time is stored as an number with base10-digits representing YYYYMMddhhmm

Author(s)

TW, Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

POSIXctToBerkeleyJulianDate

DEGebExample — *Eddy covariance data from Gebesee crop site, Germany*

Description

The data frame 'DEGebExample' contains half-hourly eddy covariance measurements from Gebesee of the years 2004 to 2006.

Usage

```
data(DEGebExample)
```

Format

For each column, the attributes 'varnames' for the variable names and 'units' for the variable units are provided.

Time stamp DateTime: POSIXct-time of the end of the half-hour period, Use as.POSIXlt(DateTime)$year to get hour, day of year, ...

Flux measurements NEE

Meteo measurements Rg, Tair, rH, VPD, Ustar

For processing of the example data see `vignette("DEGebExample")`.

Details

DISCLAIMER: This example dataset should only be used for test purposes of the REddyProc R package. For other uses, the data is openly available through the European Fluxes Database (http://www.europe-fluxdata.eu/home/site-details?id=3) and upon registration the current version can be downloaded there.

Source

The data was downloaded from `http://www.europe-fluxdata.eu` at date 2016-01-25.

Example_DETha98 — *Eddy covariance data from Tharandt, Germany*

Description

The data frame 'EddyData.F' contains half-hourly eddy covariance measurements from Tharandt of the year 1998.

Usage

```
data(Example_DETha98)
```

Format

For each column, the attributes 'varnames' for the variable names and 'units' for the variable units are provided.

Time stamp Year - Year provided with century 1998.
DoY - Day of year provided as 1 to 365 (or 1 to 366 in leap years).
Hour - Hour provided as decimal 0.0 to 23.5.

Flux measurements NEE, LE, H

Meteo measurements Rg, Tair, Tsoil, rH, VPD, Ustar

For processing of the example data see useCase vignette.

Source

The data originates from the CARBODATA CD.

`fCalcAVPfromVMFandPress` *fCalcAVPfromVMFandPress*

Description

Calculate AVP from VMF and Press

Usage

```
fCalcAVPfromVMFandPress(VMF.V.n, Press.V.n)
```

Arguments

`VMF.V.n`	Vapor mole fraction (VMF, mol / mol)
`Press.V.n`	Atmospheric pressure (Press, hPa)

Value

Data vector of actual vapor pressure (AVP, hPa (mbar))

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`fCalcETfromLE` *fCalcETfromLE*

Description

Calculate ET from LE and Tair

Usage

```
fCalcETfromLE(LE.V.n, Tair.V.n)
```

Arguments

`LE.V.n`	Data vector of latent heat (LE, W m-2)
`Tair.V.n`	Data vector of air temperature (Tair, degC)

Value

Data vector of evapotranspiration (ET, mmol H20 m-2 s-1)

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

fCalcExtRadiation *fCalcExtRadiation*

Description

Calculate the extraterrestrial solar radiation with the eccentricity correction

Usage

```
fCalcExtRadiation(DoY.V.n)
```

Arguments

`DoY.V.n`	Data vector with day of year (DoY)

Value

Data vector of extraterrestrial radiation (ExtRad, W_m-2)

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

fCalcPotRadiation *fCalcPotRadiation*

Description

Calculate the potential radiation

Usage

```
fCalcPotRadiation(DoY.V.n, Hour.V.n, Lat_deg.n,
    Long_deg.n, TimeZone_h.n, useSolartime.b = TRUE)
```

Arguments

`DoY.V.n`	Data vector with day of year (DoY), same length as Hour or length 1
`Hour.V.n`	Data vector with time as decimal hour of local time zone
`Lat_deg.n`	Latitude in (decimal) degrees
`Long_deg.n`	Longitude in (decimal) degrees
`TimeZone_h.n`	Time zone (in hours)
`useSolartime.b`	by default corrects hour (given in local winter time) for latitude to solar time « (where noon is exactly at 12:00). Set this to FALSE to directly use local winter time

Value

Data vector of potential radiation (PotRad, W_m-2)

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

Examples

```
hour <- seq(8, 16, by = 0.1)
potRadSolar <- fCalcPotRadiation(160, hour, 39.94, -5.77, TimeZone =+ 1)
potRadLocal <- fCalcPotRadiation(160, hour, 39.94, -5.77, TimeZone =+ 1
, useSolartime.b = FALSE)
plot(potRadSolar ~ hour, type = 'l')
abline(v = 13, lty = "dotted")
lines(potRadLocal ~  hour, col = "blue")
abline(v = 12, col = "blue", lty = "dotted")
legend("bottomright", legend = c("solar time", "local winter time")
, col = c("black", "blue"), inset = 0.05, lty = 1)
```

`fCalcRHfromAVPandTair` *fCalcRHfromAVPandTair*

Description

Calculate relative humidity from actual vapour pressure and air temperature

Usage

```
fCalcRHfromAVPandTair(AVP.V.n, Tair.V.n)
```

Arguments

`AVP.V.n`	Data vector of actual vapour pressure (AVP, hPa (mbar))
`Tair.V.n`	Data vector of air temperature (Tair, degC)

Value

Data vector of relative humidity (rH, %)

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`fCalcSunPosition` *fCalcSunPosition*

Description

Calculate the position of the sun

Usage

```
fCalcSunPosition(DoY.V.n, Hour.V.n, Lat_deg.n,
    Long_deg.n, TimeZone_h.n, useSolartime.b = TRUE)
```

Arguments

`DoY.V.n`	Data vector with day of year (DoY)
`Hour.V.n`	Data vector with time as decimal hour
`Lat_deg.n`	Latitude in (decimal) degrees
`Long_deg.n`	Longitude in (decimal) degrees
`TimeZone_h.n`	Time zone (in hours)
`useSolartime.b`	by default corrects hour (given in local winter time) for latitude to solar time « where noon is exactly at 12:00. Set this to FALSE to compare to code that uses local winter time

Details

This code assumes that Hour is given in local winter time zone, and corrects it by longitude to solar time (where noon is exactly at 12:00). Note: This is different form reference PVWave-code, that does not account for solar time and uses winter time zone. Set argument `useSolartime.b` to FALSE to use the local winter time instead.

Value

Data list with the following items:

`SolTime`	Solar time (SolTime, hours)
`SolDecl`	Solar declination (SolDecl, rad)
`SolElev`	Solar elevation with 0 at horizon (SolElev, rad)
`SolAzim`	Solar azimuth with 0 at North (SolAzim, rad)

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`fCalcSVPfromTair` *fCalcSVPfromTair*

Description

Calculate SVP (of water) from Tair

Usage

```
fCalcSVPfromTair(Tair.V.n)
```

Arguments

`Tair.V.n`	Data vector of air temperature (Tair, degC)

Value

Data vector of saturation vapor pressure (SVP, hPa (mbar))

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

fCalcVPDfromRHandTair *fCalcVPDfromRHandTair*

Description

Calculate VPD from rH and Tair

Usage

```
fCalcVPDfromRHandTair(RH.V.n, Tair.V.n)
```

Arguments

RH.V.n	Data vector of relative humidity (rH, %)
Tair.V.n	Data vector of air temperature (Tair, degC)

Value

Data vector of vapour pressure deficit (VPD, hPa (mbar))

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

fCheckHHTimeSeries *fCheckHHTimeSeries*

Description

Check half-hourly time series data

Usage

```
fCheckHHTimeSeries(Time.V.p, DTS.n, CallFunction.s = "")
```

Arguments

Time.V.p	Time vector in POSIX format
DTS.n	Number of daily time steps (24 or 48)
CallFunction.s	Name of function called from

Details

The number of steps per day can be 24 (hourly) or 48 (half-hourly).

The time stamp needs to be provided in POSIX time format,

equidistant half-hours,

and stamped on the half hour.

The sEddyProc procedures require at least three months of data.

Full days of data are preferred: the total amount of data rows should be a multiple of the daily time step, and

in accordance with FLUXNET standards, the dataset is spanning from the end of the first (half-)hour (0:30 or 1:00, respectively) and to midnight (0:00).

Value

Function stops on errors.

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`fConvertCtoK` *fConvertCtoK*

Description

Convert degree Celsius to degree Kelvin

Usage

```
fConvertCtoK(Celsius.V.n)
```

Arguments

`Celsius.V.n`	Data vector in Celsius (degC)

Value

Data vector in temperature Kelvin (Temp_K, degK)

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`fConvertGlobalToVisible` *fConvertGlobalToVisible*

Description

Partition global (solar) radiation into only visible (the rest is UV and infrared)

Usage

```
fConvertGlobalToVisible(Global.V.n)
```

Arguments

`Global.V.n`	Data vector of global radiation (W m-2)

Value

Data vector of visible part of solar radiation (VisRad, W m-2)

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`fConvertKtoC` *fConvertKtoC*

Description

Convert degree Kelvin to degree Celsius

Usage

```
fConvertKtoC(Kelvin.V.n)
```

Arguments

`Kelvin.V.n`	Data vector in Kelvin (degK)

Value

Data vector in temperature Celsius (Temp_C, degC)

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`fConvertTimeToPosix` *fConvertTimeToPosix*

Description

Convert different time formats to POSIX

Usage

```
fConvertTimeToPosix(Data.F, TFormat.s, Year.s = "none",
    Month.s = "none", Day.s = "none", Hour.s = "none",
    Min.s = "none", TName.s = "DateTime",
    tz = "GMT")
```

Arguments

`Data.F`	Data frame with time columns to be converted
`TFormat.s`	Abbreviation for implemented time formats
`Year.s`	Column name of year
`Month.s`	Column name of month
`Day.s`	Column name of day
`Hour.s`	Column name of hour
`Min.s`	Column name of min
`TName.s`	Column name of new column
`tz`	timezone used to store the data. Advised to keep GMT to avoid daytime shifting issues

Details

The different time formats are converted to POSIX (GMT) and a 'TimeDate' column is prefixed to the data frame

Implemented time formats:

Value

Data frame with prefixed POSIX time column.

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

Examples

```
# See unit test in test_fConvertTimeToPosix for example
```

`fConvertVisibleWm2toPhotons` *fConvertVisibleWm2toPhotons*

Description

Convert units of visible radiation from irradiance to photons flux

Usage

```
fConvertVisibleWm2toPhotons(Wm2.V.n)
```

Arguments

Wm2.V.n	Data vector in units of irradiance (W m-2)

Value

Data vector in units of photons flux (PPFD, umol photons m-2 s-1)

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

---

fLloydTaylor — *Temperature dependence of soil respiration*

Description

Temperature dependence of soil respiration after Equation 11 in Lloyd & Taylor (1994)

Usage

```
fLloydTaylor(R_ref.n, E_0.n, Tsoil.n, T_ref.n = 273.15 +
    10, T_0.n = 227.13)
```

Arguments

R_ref.n	Respiration rate at reference temperature
E_0.n	Temperature sensitivity ("activation energy") in Kelvin (degK)
Tsoil.n	Soil temperature in Kelvin (degK)
T_ref.n	Reference temperature of 10 degC in Kelvin (degK)
T_0.n	Regression temperature as fitted by LloydTaylor (1994) in Kelvin (degK)

Value

Data vector of soil respiration rate (R, umol CO2 m-2 s-1)

Author(s)

AMM reference« Lloyd J, Taylor JA (1994) On the temperature dependence of soil respiration. Functional Ecology, 8, 315-323. Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

Examples

```
  T <- c(-10:30)
resp <- fLloydTaylor(10, 330, T + 273.15)
plot(resp ~ T)
```

---

fLoadEuroFlux16 *fLoadEuroFlux16*

Description

reads a sequence of annual files in the format of europe-fluxdata 2016

Usage

```
fLoadEuroFlux16(siteName, dirName = "", additionalColumnNames = character(0))
```

Arguments

- `siteName` scalar string: the name of the site, i.e. start of the filename before _<year>_
- `dirName` scalar string: the directory where the files reside
- `additionalColumnNames` character vector: column names to read in addition to c("Month", "Day", "Hour", "NEE_st", "qf_NEE_st", "ustar", "Ta", 'Rg')

Details

The filenames should correspond to the pattern <sitename>_<YYYY>_. * .txt And hold columns c("Month", "Day", "Hour", "NEE_st", "qf_NEE_st", "ustar", "Ta", 'Rg'). By default only those columns are read and reported only c("DateTime", "NEE", "Ustar", "Tair", "Rg", "qf_NEE_st" (Note the renaming). NEE is set to NA for all values with "qf_NEE_st != 0. Values of -9999.0 are replaced by NA

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`fLoadTXTIntoDataframe` *Load text file with one header and one unit row into data frame*

Description

If gaps with the flag -9999.0 exist, these are set to NA.

Usage

```
fLoadTXTIntoDataframe(FileName.s, Dir.s = "")
```

Arguments

`FileName.s`	File name as a character string
`Dir.s`	Directory as a character string

Details

Function fLoadFluxNCIntoDataframe, which loads data from NetCDF-Files, has been moved to add-on package REddyProcNCDF. In addition, `fLoadEuroFlux16` loads data from several annual files in format corresponding to europe-fluxdata 2016.

For using only part of the records, use `fFilterAttr` to keep units attributes.

Value

Data frame with data from text file.

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

Examples

```
examplePath <- getExamplePath('Example_DETha98.txt', TRUE)
EddyData.F <- fLoadTXTIntoDataframe(examplePath)
```

---

fWriteDataframeToFile *fWriteDataframeToFile*

Description

Write data frame to ASCII tab-separated text file

Usage

```
fWriteDataframeToFile(Data.F, FileName.s,
    Dir.s = "", Digits.n = 5)
```

Arguments

Data.F	Data frame
FileName.s	File base name as a string
Dir.s	Directory as a string
Digits.n	(integer) number of digits, i.e. precision, for numeric values

Details

Missing values are flagged as -9999.0

Value

Output of data frame written to file of specified type.

Author(s)

AMM, KS Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

Examples

```
(Dir.s <- tempdir())   # directory where output is written to
fWriteDataframeToFile(Example_DETha98, 'OutputTest.txt', Dir.s = Dir.s)
```

getAmerifluxToBGC05VariableNameMapping *getAmerifluxToBGC05VariableNameMapping*

Description

map Ameriflux variable names to REddyProc defaults to names

Usage

```
getAmerifluxToBGC05VariableNameMapping(map = character(),
    mapDefault = c(YEAR = "Year", DOY = "DoY",
        NEE = "NEE", LE = "LE", H = "H",
        SW_IN = "Rg", TA = "Tair", TS = "Tsoil",
        RH = "rH", VPD = "VPD", USTAR = "Ustar",
        NEE_PI = "NEE_orig", H_PI = "H_orig",
        LE_PI = "LE_orig", NEE_F = "NEE_f",
        H_F = "H_f", LE_F = "LE_f", NEE_QC = "NEE_fqc",
        H_QC = "H_fqc", LE_QC = "LE_fqc"))
```

Arguments

map	named character vector: additional mapping, that extends or overwrites defaults in mapDefault
mapDefault	named character vector: default mapping

Details

Get a mapping of variable names of Ameriflux (Berkley 2016 Fluxnet release) to of REddyProc defaults to names

Author(s)

TW, Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

renameVariablesInDataframe

getBGC05ToAmerifluxVariableNameMapping
getBGC05ToAmerifluxVariableNameMapping

Description

map REddyProc names the Berkeley 2016 release of the Fluxnet data

Usage

```
getBGC05ToAmerifluxVariableNameMapping(map = character(),
    mapDefault = c(Year = "YEAR", DoY = "DOY",
        Rg = "SW_IN", Tair = "TA", Tsoil = "TS",
        rH = "RH", VPD = "VPD", Ustar = "USTAR",
        NEE_orig = "NEE_PI", H_orig = "H_PI",
        LE_orig = "LE_PI", NEE_f = "NEE_F",
        H_f = "H_F", LE_f = "LE_F", NEE_fqc = "NEE_QC",
        H_fqc = "H_QC", LE_fqc = "LE_QC"))
```

Arguments

map	named character vector: additional mapping, that extends or overwrites defaults in mapDefault
mapDefault	named character vector: default mapping

Details

Get a mapping of variable names of REddyProc defaults to names of the Berkeley 2016 release of the Fluxnet data

Author(s)

TW, Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

renameVariablesInDataframe

Examples

```
# adding mapping of foo, and overwriting mapping of DoY
getBGC05ToAmerifluxVariableNameMapping(c(foo = "FOO", DoY = "doy"))
```

getExamplePath *getExamplePath*

Description

checks if example filename is existing and if not tries to download it.

Usage

```
getExamplePath(filename = "Example_DETha98.txt",
    isTryDownload = FALSE, exampleDir = getREddyProcExampleDir(),
    remoteDir = "")
```

Arguments

filename	the name of the example file
isTryDownload	scalar logical whether to try downloading the file to package or tmp directory. Because of CRAN checks, need to explicitly set to TRUE
exampleDir	directory where examples are looked up and downloaded to
remoteDir	the URL do download from

Details

Example input text data files are not distributed with the package, because it exceeds allowed package size. Rather, the example files will be downloaded when required from github by this function.

The remoteDir (github) must be reachable, and the writing directory must be writeable.

Value

the full path name to the example data or if not available an zero-length character. Allows to check for `if (length(getExamplePath()) ) ...`

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

getFilledExampleDETha98Data
getFilledExampleDETha98Data

Description

Get or create the gapfilled version of the Example_DETha98 example data

Usage

```
getFilledExampleDETha98Data(exampleDir = getREddyProcExampleDir())
```

Arguments

exampleDir	the directory where the cached filled example data is stored

Value

example data.frame Example_DETha98 processed by gapfilling.

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

getREddyProcExampleDir
getREddyProcExampleDir

Description

get the example directory inside temporary directory

Usage

```
getREddyProcExampleDir(isPreferParentDir = identical(Sys.getenv("NOT_CRAN"),
    "true"), subDir = "REddyProcExamples")
```

Arguments

isPreferParentDir
: logical scalar, whether to prefer temp parent directory instead of the R-session temp-Directory. See details.

subDir
: the name of the subdirectory inside the tmp directory, where examples are stored

Details

If `isPreferParentDir = FALSE` (the default), the examples will be downloaded again for each new R-session in a session specific directory as given by `tempdir`. This corresponds to CRAN policy. IF TRUE, the parent of `tempdir` will be used, so that downloads of examples are preserved across R-sessions. This is the default if environment variable "NOT_CRAN" is defined, when running from testthat::`check`.

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

`getExamplePath`

---

getTZone *getTZone*

Description

extracts the timezone attribute from POSIXct with default on missing

Usage

```
getTZone(x, default = "GMT")
```

Arguments

x
: POSIXct vector

default
: time zone returned, if x has not timezone associated or attribute is the zero string

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

Examples

```
getTZone(as.POSIXct("2010-07-01 16:00:00", tz = "etc/GMT-1") )
getTZone(as.POSIXct("2010-07-01 16:00:00") )
# printed with local time zone, but actually has no tz attribute
getTZone(Sys.time())
```

`globalDummyVars` *globalDummyVars*

Description

Dummy global variables with the same name as fields in R5 classes have been defined.

Reason: Class methods have been defined as plain functions, so that they can be better documented. However, the assignment operator <<- has no meaning in it and therefore R CMD check complains. As a workaround they have been defined as global variable. Do not use them.

Author(s)

(Department for Biogeochemical Integration at MPI-BGC, Jena, Germany)

`LightResponseCurveFitter` *LightResponseCurveFitter*

Description

Constructs an instance of class `LightResponseCurveFitter-class`

Usage

```
LightResponseCurveFitter(...)
```

Arguments

`...`

Author(s)

(Department for Biogeochemical Integration at MPI-BGC, Jena, Germany)

LightResponseCurveFitter-class
Class "LightResponseCurveFitter"

Description

Base class for fitting parameters to light response curves (LRC)

Concrete classes for the following LRC functions are available:

- common rectangular hyperbolic light-response: RectangularLRCFitter-class
- nonrectangular hyperbolic light-response: NonrectangularLRCFitter-class
- logistic sigmoid light-response: LogisticSigmoidLRCFitter-class

They mostly differ in their prediction of GPP by method LightResponseCurveFitter_predictGPP.

Extends

All reference classes extend and inherit methods from "envRefClass".

Methods

LightResponseCurveFitter_computeLRCGradient(theta, Rg, VPD, Temp, VPD0, fixVPD, TRef):

LightResponseCurveFitter_predictGPP(Rg, ...):

LightResponseCurveFitter_predictLRC(theta, Rg, VPD, Temp, VPD0, fixVPD, TRef):

LightResponseCurveFitter_computeCost(thetaOpt, theta, iOpt, flux, sdFlux, parameterPrior, sdParamet

LightResponseCurveFitter_optimLRC(theta, iOpt, sdParameterPrior, ..., ctrl, isUsingHessian):

LightResponseCurveFitter_isParameterInBounds(theta, sdTheta, RRefNight, ctrl):

LightResponseCurveFitter_optimLRCOnAdjustedPrior(theta, iOpt, dsDay, parameterPrior, ctrl, ...):

LightResponseCurveFitter_getOptimizedParameterPositions(isUsingFixedVPD, isUsingFixedAlpha):

LightResponseCurveFitter_optimLRCBounds(theta0, parameterPrior, ..., lastGoodParameters, ctrl):

LightResponseCurveFitter_getParameterInitials(thetaPrior):

LightResponseCurveFitter_getPriorScale(thetaPrior, medianRelFluxUncertainty, nRec, ctrl):

LightResponseCurveFitter_getPriorLocation(NEEDay, RRefNight, E0):

LightResponseCurveFitter_fitLRC(dsDay, E0, sdE0, RRefNight, controlGLPart, lastGoodParameters):

LightResponseCurveFitter_getParameterNames():

Author(s)

TW

`LightResponseCurveFitter_computeCost`
LightResponseCurveFitter computeCost

Description

Computing residual sum of squares for predictions vs. data of NEE

Usage

```
LightResponseCurveFitter_computeCost(thetaOpt,
    theta, iOpt, flux, sdFlux, parameterPrior,
    sdParameterPrior, ...)
```

Arguments

`thetaOpt`	parameter vector with components of theta0 that are optimized
`theta`	parameter vector with positions as in argument of `LightResponseCurveFitter_getParameterNames`
`iOpt`	position in theta that are optimized
`flux`	numeric: NEP (-NEE) or GPP time series [umolCO2 / m2 / s], should not contain NA
`sdFlux`	numeric: standard deviation of Flux [umolCO2 / m2 / s], should not contain NA
`parameterPrior`	numeric vector along theta: prior estimate of parameter (range of values)
`sdParameterPrior`	standard deviation of parameterPrior
`...`	other arguments to `LightResponseCurveFitter_predictLRC`, such as VPD0, fixVPD

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

LightResponseCurveFitter_computeLRCGradient
LightResponseCurveFitter computeLRCGradient

Description

Gradient of LightResponseCurveFitter_predictLRC

Usage

```
LightResponseCurveFitter_computeLRCGradient(theta,
    Rg, VPD, Temp, VPD0 = 10, fixVPD = (k ==
        0), TRef = 15)
```

Arguments

theta	theta [numeric] -> parameter vector (theta[1] = k (k), theta[2] = beta (beta), theta[3] = alpha, theta[4] = RRef (rb), theta[4] = E0)
Rg	ppfd [numeric] -> photosynthetic flux density [umol / m2 / s] or Global Radiation
VPD	VPD [numeric] -> Vapor Pressure Deficit [hPa]
Temp	Temp [degC] -> Temperature [degC]
VPD0	VPDQ0 [hPa] -> Parameters VPD0 fixed to 10 hPa according to Lasslop et al 2010
fixVPD	boolean scalar or vector of nrow(theta): fixVPD if TRUE the VPD effect is not considered and VPD is not part of the computation
TRef	numeric scalar of Temperature (degree Celsius) for reference respiration RRef

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`LightResponseCurveFitter_fitLRC`
LightResponseCurveFitter fitLRC

Description

Optimize rectangular hyperbolic light response curve in one window

Usage

```
LightResponseCurveFitter_fitLRC(dsDay, E0,
    sdE0, RRefNight, controlGLPart = partGLControl(),
    lastGoodParameters = rep(NA_real_, 7L))
```

Arguments

`dsDay`	data.frame with columns NEE, Rg, Temp_C, VPD, and no NAs in NEE
`E0`	temperature sensitivity of respiration
`sdE0`	standard deviation of E_0.n
`RRefNight`	basal respiration estimated from night time data
`controlGLPart`	further default parameters (see `partGLControl`)
`lastGoodParameters`	numeric vector returned by last reasonable fit

Details

Optimization is performed for three initial parameter sets that differ by `beta0 (* 1.3, * 0.8)`. From those three, the optimization result is selected that yielded the lowest misfit. Starting values are: `k = 0`, `beta = interpercentileRange(0.03, 0.97)` of respiration, `alpha = 0.1`, `R_ref` from nightTime estimate. `E0` is fixed to the night-time estimate, but varies for estimating parameter uncertainty.

If `controlGLPart.l$nBootUncertainty == 0L` then the covariance matrix of the parameters is estimated by the Hessian of the LRC curve at optimum. Then, the additional uncertainty and covariance with uncertainty E0 is neglected.

If `controlGLPart.l$nBootUncertainty > 0L` then the covariance matrix of the parameters is estimated by a bootstrap of the data. In each draw, E0 is drawn from N ~ (E_0, sdE_0).

If there are no estimates for more than 20% of the bootstrapped samples The an NA-result with convergence code 1001L is returned.

Value

a list, If none of the optimizations from different starting conditions converged, the parameters are NA.

`thetaOpt`	numeric vector of optimized parameters including the fixed ones and E0

iOpt	index of parameters that have been optimized, here including E0, which has been optimized prior to this function.
thetaInitialGuess	the initial guess from data
covParms	numeric matrix of the covariance matrix of parameters, including E0
convergence	integer code specifying convergence problems: \ 0: good convergence \ , 1-1000: see optim \ , 1001: too few bootstraps converged \ , 1002: fitted parameters were outside reasonable bounds \ , 1003: too few valid records in window \ , 1004: near zero covariance in bootstrap indicating bad fit \ , 1005: covariance from curvature of fit yielded negative variances indicating bad fit \ , 1006: prediction of highest PAR in window was far from saturation indicating insufficient data to constrain LRC \ , 1010: no temperature-respiration relationship found \ , 1011: too few valid records in window (from different location: partGLFitLRCOneWindow) \

Author(s)

TW, MM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

partGLFitLRCWindows

LightResponseCurveFitter_optimLRCBounds

LightResponseCurveFitter_getOptimizedParameterPositions
LightResponseCurveFitter getOptimizedParameterPositions

Description

get the positions of the parameters to optimize for given Fixed

Usage

```
LightResponseCurveFitter_getOptimizedParameterPositions(isUsingFixedVPD,
    isUsingFixedAlpha)
```

Arguments

isUsingFixedVPD	boolean scalar: if TRUE, VPD effect set to zero and is not optimized
isUsingFixedAlpha	boolean scalar: if TRUE, initial slope is fixed and is not optimized

Details

If subclasses extend the parameter vector, they need to override this method.

Value

integer vector of positions in parameter vector

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`LightResponseCurveFitter_getParameterInitials` *LightResponseCurveFitter getParameterInitials*

Description

return the prior distribution of parameters

Usage

```
LightResponseCurveFitter_getParameterInitials(thetaPrior)
```

Arguments

`thetaPrior`	numeric vector prior estimate of parameters

Value

a numeric matrix (3, nPar) of initial values for fitting parameters

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

LightResponseCurveFitter_getParameterNames

LightResponseCurveFitter getParameterNames

Description

return the parameter names used by this Light Response Curve Funciton

Usage

```
LightResponseCurveFitter_getParameterNames()
```

Value

string vector of parameter names. Positions are important.

k	VPD effect
beta	saturation of GPP at high radiation
alpha	initial slope
RRef	basal respiration (units of provided NEE, usually mumol CO2 m-^-2 s^-2)
E0	temperature sensitivity estimated from night-time data (K)

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

LightResponseCurveFitter_getPriorLocation

LightResponseCurveFitter getPriorLocation

Description

return the prior distribution of parameters

Usage

```
LightResponseCurveFitter_getPriorLocation(NEEDay,
    RRefNight, E0)
```

Arguments

`NEEDay`	numeric vector of daytime NEE
`RRefNight`	numeric scalar of basal respiration estimated from night-time data
`E0`	numeric scalar of night-time estimate of temperature sensitivity

Value

a numeric vector with prior estimates of the parameters

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`LightResponseCurveFitter_getPriorScale`

LightResponseCurveFitter getPriorScale

Description

return the prior distribution of parameters

Usage

```
LightResponseCurveFitter_getPriorScale(thetaPrior,
    medianRelFluxUncertainty, nRec, ctrl)
```

Arguments

`thetaPrior`	numeric vector of location of priors
`medianRelFluxUncertainty`	numeric scalar: median across the relative uncertainty of the flux values, i.e. sdNEE / NEE
`nRec`	integer scalar: number of finite observations
`ctrl`	list of further controls, with entry `isLasslopPriorsApplied`

Details

The beta parameter is quite well defined. Hence use a prior with a standard deviation. The specific results are sometimes a bit sensitive to the uncertainty of the beta prior. This uncertainty is set corresponding to 20 times the median relative flux uncertainty. The prior is weighted n times the observations in the cost. Hence, overall it is using a weight of 1 / 20 of the weight of all observations.

However, its not well defined if PAR does not reach saturation. Need to check before applying this prior

Value

a numeric vector with prior estimates of the parameters

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`LightResponseCurveFitter_isParameterInBounds` *LightResponseCurveFitter isParameterInBounds*

Description

Check if estimated parameter vector is within reasonable bounds

Usage

```
LightResponseCurveFitter_isParameterInBounds(theta,
    sdTheta, RRefNight, ctrl)
```

Arguments

`theta`	estimate of parameter
`sdTheta`	estimate of uncertainty of the parameter
`RRefNight`	numeric scalar: night-time based estimate of basal respiration
`ctrl`	list of further controls

Details

check the Beta bounds that depend on uncertainty: outside if (beta > 100 and sdBeta >= beta)

Value

FALSE if parameters are outside reasonable bounds, TRUE otherwise

Author(s)

TW, MM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

LightResponseCurveFitter_optimLRC
LightResponseCurveFitter optimLRC

Description

call the optimization function

Usage

```
LightResponseCurveFitter_optimLRC(theta,
    iOpt, sdParameterPrior, ..., ctrl, isUsingHessian)
```

Arguments

theta	numeric vector: starting parameters
iOpt	integer vector: positions of parameters to optimize
sdParameterPrior	numeric vector: prior uncertainty
...	further arguments to the cost function
ctrl	list of further controls
isUsingHessian	scalar boolean: set to TRUE to compute Hessian at optimum

Value

list of result of optim amended with list

theta	numeric vector: optimized parameter vector including the fixed components
iOpt	integer vector: position of parameters that have been optimized

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

LightResponseCurveFitter_optimLRCBounds
LightResponseCurveFitter optimLRCBounds

Description

Optimize parameters with refitting with some fixed parameters if outside bounds

Usage

```
LightResponseCurveFitter_optimLRCBounds(theta0,
    parameterPrior, ..., lastGoodParameters,
    ctrl)
```

Arguments

`theta0`	initial parameter estimate
`parameterPrior`	prior estimate of model parameters
`...`	further parameters to `.optimLRC`, such as `dsDay`
`lastGoodParameters`	parameters vector of last successful fit
`ctrl`	list of further controls, such as `isNeglectVPDEffect = TRUE`

Details

If parameters alpha or k are outside bounds (Table A1 in Lasslop 2010), refit with some parameters fixed to values from fit of previous window.

No parameters are reported if alpha<0 or RRef < 0 or beta0 < 0 or beta0 > 250

Not parameters are reported if the data did not contain records that are near light saturation. This is checked by comparing the prediction at highest PAR with the beta parameter

Value

list result of optimization as of `LightResponseCurveFitter_optimLRCOnAdjustedPrior` with entries

`theta`	numeric parameter vector that includes the fixed components
`iOpt`	integer vector of indices of the vector that have been optimized
`convergence`	scalar integer indicating bad conditions on fitting (see `LightResponseCurveFitter_fitLRC`)

Author(s)

TW, MM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

`LightResponseCurveFitter_fitLRC`

`LightResponseCurveFitter_optimLRCOnAdjustedPrior`

LightResponseCurveFitter optimLRCOnAdjustedPrior

Description

Lower bound flux uncertainty and adjust prior uncertainty before calling optimLRC

Usage

```
LightResponseCurveFitter_optimLRCOnAdjustedPrior(theta,
    iOpt, dsDay, parameterPrior, ctrl, ...)
```

Arguments

`theta`	numeric vector of starting values
`iOpt`	integer vector: positions of subset of parameters that are optimized
`dsDay`	dataframe of NEE, sdNEE and predictors Rg, VPD and Temp
`parameterPrior`	numeric vector of prior parameter estimates (corresponding to theta) # TODO rename to thetaPrior
`ctrl`	list of further controls
`...`	further arguments to `LightResponseCurveFitter_optimLRC` (passed to `LightResponseCurveFitter_c`

Details

Only those records are used for optimization where both NEE and sdNEE are finite. In larger settings, already filtered at

Optimization of LRC parameters takes into account the uncertainty of the flux values. In order to avoid very strong leverage, values with a very low uncertainty (< a lower quantile) are assigned the lower quantile is assigned. This procedure downweighs records with a high uncertainty, but does not apply a large leverage for records with a very low uncertainty. Avoid this correction by setting `ctrl$isBoundLowerNEEUncertainty = FALSE`

The uncertainty of the prior, that maybe derived from fluxes) is allowed to adapt to the uncertainty of the fluxes. This is done in `link{LightResponseCurveFitter_getPriorScale}`

Value

result of LightResponseCurveFitter_optimLRC with items theta, iOpt and convergence

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`LightResponseCurveFitter_predictGPP`

LightResponseCurveFitter predictGPP

Description

Light Response function for GPP

Usage

```
LightResponseCurveFitter_predictGPP(Rg, ...)
```

Arguments

Rg	ppfd [numeric] -> photosynthetic flux density [mumol / m2 / s] or Global Radiation
...	further parameters to the LRC

Details

This method must be be implemented by a specific subclass. Currently there are several alternatives:

- Rectangular: RectangularLRCFitter_predictGPP
- Nonrectangular: NonrectangularLRCFitter_predictGPP
- Rectangular: LogisticSigmoidLRCFitter_predictGPP

Value

numeric vector of length(Rg) of GPP

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

`partitionNEEGL`

`LightResponseCurveFitter_predictLRC`

LightResponseCurveFitter predictLRC

Description

Light Response Function

Usage

```
LightResponseCurveFitter_predictLRC(theta,
    Rg, VPD, Temp, VPD0 = 10, fixVPD = (k ==
        0), TRef = 15)
```

Arguments

`theta`	numeric vector of parameters
`Rg`	ppfd [numeric] -> photosynthetic flux density [umol / m2 / s] or Global Radiation
`VPD`	VPD [numeric] -> Vapor Pressure Deficit [hPa]
`Temp`	Temp [degC] -> Temperature [degC]
`VPD0`	VPD0 [hPa] -> Parameters VPD0 fixed to 10 hPa according to Lasslop et al 2010
`fixVPD`	boolean scalar or vector of nrow theta: fixVPD if TRUE the VPD effect is not considered and VPD is not part of the computation
`TRef`	numeric scalar of Temperature (degree Celsius) for reference respiration RRef

Details

Predict ecosystem fluxes (Reco, GPP, NEP = GPP-Reco) for given parameters and environmental conditions.

The VPD effect is included according to Lasslop et al., 2010.

If theta is a matrix, a different row of parameters is used for different entries of other inputs

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`LogisticSigmoidLRCFitter`

LogisticSigmoidLRCFitter

Description

Constructs an instance of class `LogisticSigmoidLRCFitter-class`

Usage

```
LogisticSigmoidLRCFitter(...)
```

Arguments

`...`

Author(s)

(Department for Biogeochemical Integration at MPI-BGC, Jena, Germany)

`LogisticSigmoidLRCFitter-class`

Class `"LogisticSigmoidLRCFitter"`

Description

Logistic sigmoid light-response curve fitting.

Extends

Class "`LightResponseCurveFitter`", directly.

All reference classes extend and inherit methods from "`envRefClass`".

Methods

`computeGPPGradient(Rg, Amax, alpha):` ~~

`predictGPP(Rg, Amax, alpha):` ~~

The following methods are inherited (from the corresponding class): predictGPP ("LightResponseCurveFitter"), getParameterNames ("LightResponseCurveFitter"), fitLRC ("LightResponseCurveFitter"), getPriorLocation ("LightResponseCurveFitter"), getPriorScale ("LightResponseCurveFitter"), getParameterInitials ("LightResponseCurveFitter"), optimLRCBounds ("LightResponseCurveFitter"), getOptimizedParameterPositions ("LightResponseCurveFitter"), optimLRCOnAdjustedPrior ("LightResponseCurveFitter"), isParameterInBounds ("LightResponseCurveFitter"), optimLRC ("LightResponseCurveFitter"), computeCost ("LightResponseCurveFitter"), predictLRC ("LightResponseCurveFitter"), computeLRCGradient ("LightResponseCurveFitter")

LogisticSigmoidLRCFitter_predictGPP

LogisticSigmoidLRCFitter predictGPP

Description

Logistic Sigmoid Light Response function for GPP

Usage

```
LogisticSigmoidLRCFitter_predictGPP(Rg, Amax,
    alpha)
```

Arguments

`Rg`	ppfd [numeric] -> photosynthetic flux density [mumol / m2 / s] or Global Radiation
`Amax`	vector of length(Rg): saturation (beta parameter) adjusted for effect of VPD for each line of Rg
`alpha`	numeric scalar or vector of length(Rg): alpha parameter: slope at Rg = 0

Details

```
GPP <- Amax * tanh(alpha * Rg / Amax)
```

Value

numeric vector of length(Rg) of GPP

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

`LightResponseCurveFitter_predictGPP`

`NonrectangularLRCFitter` *NonrectangularLRCFitter*

Description

Constructs an instance of class `NonrectangularLRCFitter-class`

Usage

```
NonrectangularLRCFitter(...)
```

Arguments

`...`

Author(s)

(Department for Biogeochemical Integration at MPI-BGC, Jena, Germany)

`NonrectangularLRCFitter-class` *Class* `"NonrectangularLRCFitter"`

Description

Nonrectangular hyperbolic light-response curve fitting.

Extends

Class `"LightResponseCurveFitter"`, directly.

All reference classes extend and inherit methods from `"envRefClass"`.

Methods

`computeGPPGradient(Rg, Amax, alpha, logitconv): ~~`

`getParameterNames(): ~~`

`getPriorLocation(NEEDay, RRefNight, E0): ~~`

`getPriorScale(thetaPrior, medianRelFluxUncertainty, nRec, ctrl): ~~`

`getOptimizedParameterPositions(isUsingFixedVPD, isUsingFixedAlpha): ~~`

`predictLRC(theta, Rg, VPD, Temp, VPD0, fixVPD, TRef): ~~`

`predictGPP(Rg, Amax, alpha, conv): ~~`

`computeLRCGradient(theta, Rg, VPD, Temp, VPD0, fixVPD, TRef): ~~`

The following methods are inherited (from the corresponding class): computeLRCGradient ("LightResponseCurveFitter"), predictGPP ("LightResponseCurveFitter"), predictLRC ("LightResponseCurveFitter"), getOptimizedParameterPositions ("LightResponseCurveFitter"), getPriorScale ("LightResponseCurveFitter"), getPriorLocation ("LightResponseCurveFitter"), getParameterNames ("LightResponseCurveFitter"), fitLRC ("LightResponseCurveFitter"), getParameterInitials ("LightResponseCurveFitter"), optimLRCBounds ("LightResponseCurveFitter"), optimLRCOnAdjustedPrior ("LightResponseCurveFitter"), isParameterInBounds ("LightResponseCurveFitter"), optimLRC ("LightResponseCurveFitter"), computeCost ("LightResponseCurveFitter")

`NonrectangularLRCFitter_getParameterNames`

NonrectangularLRCFitter getParameterNames

Description

return the parameter names used by this Light Response Curve Function

Usage

```
NonrectangularLRCFitter_getParameterNames()
```

Value

string vector of parameter names. Positions are important. Adds sixth parameter, `logitconv` to the parameters of `LightResponseCurveFitter_getParameterNames`

`logitconf` logit-transformed convexity parameter. The value at original scale is obtained by `conv = 1 / (1 + exp(-logitconv))`

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

NonrectangularLRCFitter_predictGPP

NonrectangularLRCFitter_predictGPP
NonrectangularLRCFitter predictGPP

Description

Nonrectangular hyperbolic Light Response function for GPP

Usage

```
NonrectangularLRCFitter_predictGPP(Rg, Amax,
    alpha, conv)
```

Arguments

Rg	ppfd [numeric] -> photosynthetic flux density [mumol / m2 / s] or Global Radiation
Amax	numeric scalar or vector of length(Rg): beta parameter adjusted for VPD effect
alpha	numeric scalar or vector of length(Rg): alpha parameter: initial slope
conv	numeric scalar or vector of length(Rg): convexity parameter (see details)

Details

This function generalizes the RectangularLRCFitter_predictGPP by adding the convexity parameter conv. For conv -> 0 (logitconv -> -Inf): approaches the rectangular hyperbolic. For conv -> 1 (logitconv -> + Inf): approaches a step function. Expected values of conv are about 0.7-0.9 (Moffat 2012).

Value

numeric vector of length(Rg) of GPP

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

LightResponseCurveFitter_predictGPP

`partGLControl` *partGLControl*

Description

Default list of parameters for Lasslop 2010 daytime flux partitioning For highest compatibility to the pvWave code of G.Lasslop (used by first BGC-online tool) see function `partGLControlLasslopCompatible`.

Usage

```
partGLControl(LRCFitConvergenceTolerance = 0.001,
    nLRCFitConvergenceTolerance = 0.001,
    nBootUncertainty = 30L, minNRecInDayWindow = 10L,
    isAssociateParmsToMeanOfValids = TRUE,
    isLasslopPriorsApplied = TRUE, isUsingLasslopQualityConstraints = FALSE,
    isSdPredComputed = TRUE, isFilterMeteoQualityFlag = FALSE,
    isBoundLowerNEEUncertainty = TRUE, fixedTRefAtNightTime = NA,
    isExtendTRefWindow = TRUE, smoothTempSensEstimateAcrossTime = TRUE,
    isNeglectPotRadForNight = FALSE, NRHRfunction = FALSE,
    isNeglectVPDEffect = FALSE, isRefitMissingVPDWithNeglectVPDEffect = TRUE,
    fixedTempSens = data.frame(E0 = NA_real_,
        sdE0 = NA_real_, RRef = NA_real_),
    replaceMissingSdNEEParms = c(perc = 0.2,
        minSd = 0.7), neglectNEEUncertaintyOnMissing = FALSE,
    minPropSaturation = NA)
```

Arguments

`LRCFitConvergenceTolerance`
: convergence criterion for rectangular light response curve fit. If relative improvement of reducing residual sum of squares between predictions and observations is less than this criterion, assume convergence. Decrease to get more precise parameter estimates, Increase for speedup.

`nLRCFitConvergenceTolerance`
: convergence criterion for nonrectangular light response curve fit. Here its a factor of machine tolerance.

`nBootUncertainty`
: number of bootstrap samples for estimating uncertainty. Set to zero to derive uncertainty from curvature of a single fit

`minNRecInDayWindow`
: Minimum number of data points for regression

`isAssociateParmsToMeanOfValids`
: set to FALSE to associate parameters to the first record of the window for interpolation instead of mean across valid records inside a window

`isLasslopPriorsApplied`
: set to TRUE to apply strong fixed priors on LRC fitting. Returned parameter estimates claimed valid for some case where not enough data was available

isUsingLasslopQualityConstraints
: set to TRUE to avoid quality constraints additional to Lasslop 2010

isSdPredComputed
: set to FALSE to avoid computing standard errors of Reco and GPP for small performance increase

isFilterMeteoQualityFlag
: set to TRUE to use only records where quality flag of meteo drivers (radiation, temperature, VPD) is zero, i.e. non-gapfilled for parameter estimation. For prediction, the gap-filled value is used always, to produce predictions also for gaps.

isBoundLowerNEEUncertainty
: set to FALSE to avoid adjustment of very low uncertainties before day-Time fitting that avoids the high leverage those records with unreasonable low uncertainty.

fixedTRefAtNightTime
: if a finite value (degree Centigrade) is given, it is used instead of median data temperature as reference temperature in estimation of temperature sensitivity from night data

isExtendTRefWindow
: set to FALSE to avoid successively extending the night-time window in order to estimate a temperature sensitivity where previous estimates failed

smoothTempSensEstimateAcrossTime
: set to FALSE to use independent estimates of temperature sensitivity on each windows instead of a vector of E0 that is smoothed over time

isNeglectPotRadForNight
: set to TRUE to not use potential radiation in determining night-time data.

NRHRfunction
: deprecated: Flag if TRUE use the NRHRF for partitioning; Now use `lrcFitter = NonrectangularLRCF`

isNeglectVPDEffect
: set to TRUE to avoid using VPD in the computations. This may help when VPD is rarely measured.

isRefitMissingVPDWithNeglectVPDEffect
: set to FALSE to avoid repeating estimation with `isNeglectVPDEffect = TRUE` trying to predict when VPD is missing

fixedTempSens

replaceMissingSdNEEParms
: parameters for replacing missing standard deviation of NEE. see `replaceMissingSdByPercentage`. Default sets missing uncertainty to 20% of NEE but at least 0.7 flux-units (usually mumol CO2 / m2 / s). Specify c(NA, NA) to avoid replacing missings in standard deviation of NEE and to omit those records from LRC fit.

neglectNEEUncertaintyOnMissing
: If set to TRUE: if there are records with missing uncertainty of NEE inside one window, set all uncertainties to 1. This overrules option replaceMissingSdNEEParms.

minPropSaturation
: quality criterion for sufficient data in window. If GPP prediction of highest PAR of window is less than minPropSaturation * (GPP at light-saturation, i.e. beta) this indicates that PAR is not sufficiently high to constrain the shape of the LRC

Value

list with entries of given arguments.

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

`partitionNEEGL`

Examples

```
partGLControl(nBootUncertainty = 40L)
```

---

partGLControlLasslopCompatible

partGLControlLasslopCompatible

Description

Daytime flux partitioning parms compatible with with the pvWave

Usage

```
partGLControlLasslopCompatible(nBootUncertainty = 0L,
    minNRecInDayWindow = 10L, isAssociateParmsToMeanOfValids = FALSE,
    isLasslopPriorsApplied = TRUE, isUsingLasslopQualityConstraints = TRUE,
    isBoundLowerNEEUncertainty = FALSE, fixedTRefAtNightTime = 15,
    isExtendTRefWindow = FALSE, smoothTempSensEstimateAcrossTime = FALSE,
    isRefitMissingVPDWithNeglectVPDEffect = FALSE,
    minPropSaturation = NA, isNeglectVPDEffect = FALSE,
    replaceMissingSdNEEParms = c(NA, NA),
    neglectNEEUncertaintyOnMissing = TRUE,
    ...)
```

Arguments

`nBootUncertainty`
: 0: Derive uncertainty from curvature of a single fit, neglecting the uncertainty of previously estimated temperature sensitivity, E0

`minNRecInDayWindow`
: Minimum number of 10 valid records for regression in a single window

`isAssociateParmsToMeanOfValids`
: associate parameters to the first record of the window for interpolation instead of mean across valid records inside a window

`isLasslopPriorsApplied`
: Apply fixed Lasslop priors in LRC fitting.

`isUsingLasslopQualityConstraints`
: avoid quality constraints additional to the ones in Lasslop 2010

`isBoundLowerNEEUncertainty`
: FALSE: avoid adjustment of very low uncertainties before day-Time fitting that avoids the high leverage those records with unreasonable low uncertainty.

`fixedTRefAtNightTime`
: use fixed (degree Centigrade) temperature sensitivity instead of median data temperature as reference temperature in estimation of temperature sensitivity from night data

`isExtendTRefWindow`
: avoid successively extending the night-time window in order to estimate a temperature sensitivity where previous estimates failed

`smoothTempSensEstimateAcrossTime`
: FALSE: use independent estimates of temperature sensitivity on each windows instead of a vector of E0 that is smoothed over time

`isRefitMissingVPDWithNeglectVPDEffect`
: FALSE: avoid repeating estimation with `isNeglectVPDEffect = TRUE`

`minPropSaturation`
: NA: avoid quality constraint of sufficient saturation in data This option is overruled, i.e. not considered, if option isUsingLasslopQualityConstraints = TRUE.

`isNeglectVPDEffect`
: FALSE: do not neglect VPD effect

`replaceMissingSdNEEParms`
: do not replace missing NEE, but see option

`neglectNEEUncertaintyOnMissing`
: if there are records with missing uncertainty of NEE inside one window, set all sdNEE to 1. This overrules option replaceMissingSdNEEParms.

`...`
: further arguments to `partGLControl`

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

`partGLControl`

Examples

```
partGLControlLasslopCompatible()
```

---

partGLExtractStandardData
partGLExtractStandardData

Description

Relevant columns from original input with defined names

Usage

```
partGLExtractStandardData(ds, NEEVar.s = paste0("NEE",
    SuffixDash.s, "_f"), QFNEEVar.s = paste0("NEE",
    SuffixDash.s, "_fqc"), QFNEEValue.n = 0,
    NEESdVar.s = paste0("NEE", SuffixDash.s,
        "_fsd"), TempVar.s = paste0("Tair_f"),
    QFTempVar.s = paste0("Tair_fqc"), QFTempValue.n = 0,
    VPDVar.s = paste0("VPD_f"), QFVPDVar.s = paste0("VPD_fqc"),
    QFVPDValue.n = 0, RadVar.s = "Rg_f",
    QFRadVar.s = paste0("Rg_fqc"), QFRadValue.n = 0,
    PotRadVar.s = "PotRad_NEW", Suffix.s = "",
    controlGLPart = partGLControl())
```

Arguments

`ds`	dataset with all the specified input columns and full days in equidistant times
`NEEVar.s`	Variable of NEE
`QFNEEVar.s`	Quality flag of variable
`QFNEEValue.n`	Value of quality flag for _good_ (original) data
`NEESdVar.s`	Variable of standard deviation of net ecosystem fluxes
`TempVar.s`	Filled air or soil temperature variable (degC)
`QFTempVar.s`	Quality flag of filled temperature variable
`QFTempValue.n`	Value of temperature quality flag for _good_ (original) data
`VPDVar.s`	Filled Vapor Pressure Deficit, VPD (hPa)
`QFVPDVar.s`	Quality flag of filled VPD variable
`QFVPDValue.n`	Value of VPD quality flag for _good_ (original) data
`RadVar.s`	Filled radiation variable
`QFRadVar.s`	Quality flag of filled radiation variable
`QFRadValue.n`	Value of radiation quality flag for _good_ (original) data
`PotRadVar.s`	Variable name of potential rad. (W / m2)
`Suffix.s`	string inserted into column names before identifier for NEE column defaults (see `sEddyProc_sMDSGapFillAfterUstar`).
`controlGLPart`	further default parameters, see `partGLControl`

Details

The LRC fit usually weights NEE records by its uncertainty. In order to also use records with missing `NEESdVar.s`, uncertainty of the missing values is by default set to a conservatively high value, parameterized by `controlGLPart$replaceMissingSdNEEParms)`. Controlled by argument `replaceMissingSdNEEParms` in `partGLControl`, but overruled by argument `neglectNEEUncertaintyOnMissing`.

Value

a data.frame with columns

`sDateTime`	first column of `ds`, usually the time stamp not used, but usually first column is a dateTime is kept for aiding debug
`NEE`	NEE filtered for quality flay
`sdNEE`	standard deviation of NEE with missing values replaced
`Temp`	Temperature, quality filtered if isTRUE(controlGLPart$isFilterMeteoQualityFlag)
`VPD`	Water pressure deficit, quality filtered if isTRUE(controlGLPart$isFilterMeteoQualityFlag)
`Rg`	Incoming radiation
`isDay`	Flag that is true for daytime records
`isNight`	Flag that is true for nighttime records

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`partitionNEEGL` *partitionNEEGL*

Description

Partitioning NEE fluxes into GP and Reco after daytime method.

Usage

```
partitionNEEGL(ds, NEEVar.s = paste0("NEE",
    SuffixDash.s, "_f"), TempVar.s = "Tair_f",
    VPDVar.s = "VPD_f", RadVar.s = "Rg_f",
    Suffix.s = "", ..., controlGLPart = partGLControl(),
    isVerbose = TRUE, nRecInDay = 48L, lrcFitter = RectangularLRCFitter())
```

Arguments

`ds`	dataset with all the specified input columns and full days in equidistant times
`NEEVar.s`	Variable of NEE
`TempVar.s`	Filled air or soil temperature variable (degC)
`VPDVar.s`	Filled Vapor Pressure Deficit - VPD - (hPa)
`RadVar.s`	Filled radiation variable
`Suffix.s`	string inserted into column names before identifier for NEE column defaults (see `sEddyProc_sMDSGapFillAfterUstar`).
`...`	further arguments to `partGLExtractStandardData`, such as `PotRadVar.s`
`controlGLPart`	further default parameters, see `partGLControl`
`isVerbose`	set to FALSE to suppress output messages
`nRecInDay`	number of records within one day (for half-hourly data its 48)
`lrcFitter`	R5 class instance responsible for fitting the light response curve. Current possibilities are `RectangularLRCFitter()`, `NonrectangularLRCFitter()`, and `LogisticSigmoidLRCFitter()`.

Details

Daytime-based partitioning of measured net ecosystem fluxes into gross primary production (GPP) and ecosystem respiration (Reco)

The fit to the light-response-curve is done by default using the Rectangular hyperbolic function, as in Lasslop et al. (2010) Alternative fittings can be used by providing the corresponding subclass of `LightResponseCurveFitter-class` to `lrcFitter` argument. (see `LightResponseCurveFitter_predictGPP`)

While the extrapolation uses filled data, the parameter optimization uses only measured data, i.e. with specified quality flag. With the common case where VPD is missing for fitting the LRC, by default (with `controlGLPart$isRefitMissingVPDWithNeglectVPDEffect = TRUE`) is to redo the estimation of LRC parameters with neglecting the VPD-effect. Next, in the predictions (rows) with missing VPD are then replaced with predictions based on LRC-fits that neglected the VPD effect.

Value

`Reco_DT_<suffix>`	predicted ecosystem respiration: mumol CO2/m2/s
`GPP_DT_<suffix>`	predicted gross primary production mumol CO2/m2/s
`GPP2000`	predicted gross primary production mumol CO2 / m2 / s for VPD = 0 at Rg = 2000
`<LRC>`	Further light response curve (LRC) parameters and their standard deviation depend on the used LRC (e.g. for the non-rectangular LRCC see `NonrectangularLRCFitter_getParamete`). They are estimated for windows and are reported with the first record of the window
`FP_VARnight`	NEE filtered for nighttime records (others NA)
`FP_VARday`	NEE filtered for daytime records (others NA)

NEW_FP_Temp	temperature after filtering for quality flag degree Celsius
NEW_FP_VPD	vapour pressure deficit after filtering for quality flag, hPa
FP_RRef_Night	basal respiration estimated from nighttime (W / m2)
FP_qc	quality flag: 0: good parameter fit, 1: some parameters out of range, required refit, 2: next parameter estimate is more than two weeks away
FP_dRecPar	records until or after closest record that has a parameter estimate associated
FP_errorcode	information why LRC-fit was not successful or was rejected, see result of LightResponseCurveFitter_f
FP_GPP2000	predicted GPP at VPD = 0 and PAR = 2000: a surrogate for maximum photosynthetic capacity
FP_OPT_VPD	list object of fitting results including iOpt and covParms
FP_OPT_NoVPD	same as FP_OPT_VPD holding optimization results with fit neglecting the VPD effect

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

References

Lasslop G, Reichstein M, Papale D, et al. (2010) Separation of net ecosystem exchange into assimilation and respiration using a light response curve approach: critical issues and global evaluation. Global Change Biology, Volume 16, Issue 1, Pages 187-208

See Also

partGLFitNightTimeTRespSens

partGLFitLRCWindows

partGLInterpolateFluxes

POSIXctToBerkeleyJulianDate

POSIXctToBerkeleyJulianDate

Description

convert POSIXct to JulianDate format used in Berkeley release

Usage

```
POSIXctToBerkeleyJulianDate(sDateTime)
```

Arguments

sDateTime	POSIXct vector

Details

In the Berkeley-Release of the fluxnet data, the time is stored as an number with base10-digits representing YYYYMMddhhmm

Author(s)

TW, Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

BerkeleyJulianDateToPOSIXct

RectangularLRCFitter *RectangularLRCFitter*

Description

Constructs an instance of class RectangularLRCFitter-class

Usage

```
RectangularLRCFitter(...)
```

Arguments

...	

Author(s)

(Department for Biogeochemical Integration at MPI-BGC, Jena, Germany)

RectangularLRCFitter-class
Class "RectangularLRCFitter"

Description

Common rectangular hyperbolic light-response curve fitting.

Extends

Class "LightResponseCurveFitter", directly.

All reference classes extend and inherit methods from "envRefClass".

Methods

`computeGPPGradient(Rg, Amax, alpha):` ~~

`predictGPP(Rg, Amax, alpha):` ~~

The following methods are inherited (from the corresponding class): predictGPP ("LightResponseCurveFitter"), getParameterNames ("LightResponseCurveFitter"), fitLRC ("LightResponseCurveFitter"), getPriorLocation ("LightResponseCurveFitter"), getPriorScale ("LightResponseCurveFitter"), getParameterInitials ("LightResponseCurveFitter"), optimLRCBounds ("LightResponseCurveFitter"), getOptimizedParameterPositions ("LightResponseCurveFitter"), optimLRCOnAdjustedPrior ("LightResponseCurveFitter"), isParameterInBounds ("LightResponseCurveFitter"), optimLRC ("LightResponseCurveFitter"), computeCost ("LightResponseCurveFitter"), predictLRC ("LightResponseCurveFitter"), computeLRCGradient ("LightResponseCurveFitter")

Author(s)

TW

RectangularLRCFitterCVersion
RectangularLRCFitterCVersion

Description

Constructs an instance of class `RectangularLRCFitterCVersion-class`

Usage

```
RectangularLRCFitterCVersion(...)
```

Arguments

`...`

Author(s)

(Department for Biogeochemical Integration at MPI-BGC, Jena, Germany)

RectangularLRCFitterCVersion-class *Class* "RectangularLRCFitterCVersion"

Description

Common rectangular hyperbolic light-response curve fitting, implemented with faster C-based cost function.

Extends

Class "RectangularLRCFitter", directly. Class "LightResponseCurveFitter", by class "RectangularLRCFitter", distance 2.

All reference classes extend and inherit methods from "envRefClass".

Methods

`computeCost(thetaOpt, theta, iOpt, flux, sdFlux, parameterPrior, sdParameterPrior, ..., VPD0, fixVP`
~~

The following methods are inherited (from the corresponding class): computeCost ("LightResponseCurveFitter"), computeLRCGradient ("LightResponseCurveFitter"), predictGPP ("RectangularLRCFitter"), predictLRC ("LightResponseCurveFitter"), optimLRC ("LightResponseCurveFitter"), isParameterInBounds ("LightResponseCurveFitter"), optimLRCOnAdjustedPrior ("LightResponseCurveFitter"), getOptimizedParameterPositions ("LightResponseCurveFitter"), optimLRCBounds ("LightResponseCurveFitter"), getParameterInitials ("LightResponseCurveFitter"), getPriorScale ("LightResponseCurveFitter"), getPriorLocation ("LightResponseCurveFitter"), fitLRC ("LightResponseCurveFitter"), getParameterNames ("LightResponseCurveFitter"), predictGPP ("LightResponseCurveFitter"), computeGPPGradient ("RectangularLRCFitter")

RectangularLRCFitter_predictGPP *RectangularLRCFitter predictGPP*

Description

Rectangular hyperbolic Light Response function for GPP

Usage

```
RectangularLRCFitter_predictGPP(Rg, Amax,
    alpha)
```

Arguments

`Rg`	ppfd [numeric] -> photosynthetic flux density [mumol / m2 / s] or Global Radiation
`Amax`	vector of length(Rg): saturation (beta parameter) adjusted for effect of VPD for each line of Rg
`alpha`	numeric scalar or vector of length(Rg): alpha parameter: slope at Rg = 0

Value

numeric vector of length(Rg) of GPP

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

`LightResponseCurveFitter_predictGPP`

`renameVariablesInDataframe`

renameVariablesInDataframe

Description

Rename the column names of a data.frame according to a given mapping

Usage

```
renameVariablesInDataframe(data.F, mapping = getBGC05ToAmerifluxVariableNameMapping())
```

Arguments

`data.F`	data.frame whose columns should be renamed
`mapping`	named character vector: specifying a renaming (name -> value) of the variables, see e.g. `getAmerifluxToBGC05VariableNameMapping`

Author(s)

TW, Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`RHLightResponseCostC` *RHLightResponseCostC*

Description

Computing residual sum of squares for predictions vs. data of NEE implemented in C

Usage

```
RHLightResponseCostC(theta, flux, sdFlux,
    parameterPrior, sdParameterPrior, Rg,
    VPD, Temp, VPD0, fixVPD)
```

Arguments

`theta`

`flux`

`sdFlux`

`parameterPrior`

`sdParameterPrior`

`Rg`

`VPD`

`Temp`

`VPD0`

`fixVPD`

Author(s)

(Department for Biogeochemical Integration at MPI-BGC, Jena, Germany)

`sEddyProc` *sEddyProc*

Description

create an instance of class `sEddyProc-class`

Usage

```
sEddyProc(...)
```

Arguments

`...`

Author(s)

(Department for Biogeochemical Integration at MPI-BGC, Jena, Germany)

`sEddyProc-class` *Class* `"sEddyProc"`

Description

R5 reference class for processing of site-level half-hourly eddy data

Extends

All reference classes extend and inherit methods from `"envRefClass"`.

Fields

private, not to be accessed directly:

`sID`: Object of class `character` with Site ID

`sDATA`: Object of class `data.frame` with (fixed) site data

`sINFO`: Object of class `list` with site information

`sLOCATION`: Object of class `list` with site location information

`sTEMP`: Object of class `data.frame` of (temporary) result data

`sUSTAR`: Object of class `list` with results form uStar Threshold estimation

Methods

Setup, import and export

`sEddyProc_initialize(ID.s, Data.F, ColNames.V.s, ColPOSIXTime.s, DTS.n, ColNamesNonNumeric.V.s, Lat`

`sEddyProc_sSetLocationInfo(Lat_deg.n, Long_deg.n, TimeZone_h.n)`
`sEddyProc_sExportResults(isListColumnsExported)`
`sEddyProc_sExportData()`
`sEddyProc_sGetData()`

uStar threshold estimation

`sEddyProc_sEstUstarThresholdDistribution(ctrlUstarEst.l, ctrlUstarSub.l, UstarColName, NEEColName,`

`sEddyProc_sEstUstarThreshold(UstarColName, NEEColName, TempColName, RgColName, ...)`

`sEddyProc_sPlotNEEVersusUStarForSeason(season.s, Format.s, Dir.s, UstarColName, NEEColName, TempCol`

Gapfilling

`sEddyProc_sCalcPotRadiation(useSolartime.b)`
`sEddyProc_sMDSGapFill(Var.s, QFVar.s, QFValue.n, V1.s, T1.n, V2.s, T2.n, V3.s, T3.n, FillAll.b, Ver`

`sEddyProc_sMDSGapFillAfterUStarDistr(..., UstarThres.df, UstarSuffix.V.s)`
`sEddyProc_sMDSGapFillAfterUstar(FluxVar.s, UstarVar.s, UstarThres.df, UstarSuffix.s, FlagEntryAfter`

`sEddyProc_sFillMDC(WinDays.i, Verbose.b)`
`sEddyProc_sFillLUT(WinDays.i, V1.s, T1.n, V2.s, T2.n, V3.s, T3.n, V4.s, T4.n, V5.s, T5.n, Verbose.b`

`sEddyProc_sFillInit(Var.s, QFVar.s, QFValue.n, FillAll.b)`

Flux partitioning

`sEddyProc_sMRFluxPartition(FluxVar.s, QFFluxVar.s, QFFluxValue.n, TempVar.s, QFTempVar.s, QFTempVal`

`sEddyProc_sGLFluxPartition(..., debug.l, isWarnReplaceColumns)`

Plotting

`sEddyProc_sPlotDailySums(Var.s, VarUnc.s, Format.s, Dir.s, unit.s, ...)`
`sEddyProc_sPlotDailySumsY(Var.s, VarUnc.s, Year.i, timeFactor.n, massFactor.n, unit.s)`

`sEddyProc_sPlotHHFluxes(Var.s, QFVar.s, QFValue.n, Format.s, Dir.s)`
`sEddyProc_sPlotHHFluxesY(Var.s, QFVar.s, QFValue.n, Year.i)`
`sEddyProc_sPlotDiurnalCycle(Var.s, QFVar.s, QFValue.n, Format.s, Dir.s)`
`sEddyProc_sPlotFingerprint(Var.s, QFVar.s, QFValue.n, Format.s, Dir.s, ...)`
`sEddyProc_sPlotFingerprintY(Var.s, QFVar.s, QFValue.n, Year.i, Legend.b, Col.V, valueLimits)`

Note

for examples see useCase vignette

Author(s)

AM, TW

`sEddyProc_initialize` *sEddyProc_initialize - Initialization of sEddyProc*

Description

This function is called when writing `sEddyProc$new`. It creates the fields of the sEddyProc R5 reference class for processing of half-hourly eddy data

Usage

```
sEddyProc_initialize(ID.s, Data.F, ColNames.V.s,
    ColPOSIXTime.s = "DateTime", DTS.n = 48,
    ColNamesNonNumeric.V.s = character(0),
    Lat_deg.n = NA_real_, Long_deg.n = NA_real_,
    TimeZone_h.n = NA_integer_, ...)
```

Arguments

`ID.s`	String with site ID
`Data.F`	Data frame with at least three month of half-hourly site-level eddy data
`ColNames.V.s`	Vector with selected column names, the less columns the faster the processing
`ColPOSIXTime.s`	Column name with POSIX time stamp
`DTS.n`	Daily time steps
`ColNamesNonNumeric.V.s`	Names of columns that should not be checked for numeric type, e.g. season column
`Lat_deg.n`	Latitude in (decimal) degrees (-90 to + 90)
`Long_deg.n`	Longitude in (decimal) degrees (-180 to + 180)
`TimeZone_h.n`	Time zone (in hours) shift to UTC, e.g. 1 for Berlin
`...`	('...' required for initialization of class fields)

Details

The time stamp must be provided in POSIX format, see also fConvertTimeToPosix. For required properties of the time series, see fCheckHHTimeSeries.

Internally the half-hour time stamp is shifted to the middle of the measurement period (minus 15 minutes or 30 minutes).

All other columns may only contain numeric data. Please use NA as a gap flag for missing data or low quality data not to be used in the processing. The columns are also checked for plausibility with warnings if outside range.

sID is a string for the site ID.

sDATA is a data frame with site data.

sTEMP is a temporal data frame with the processing results.

sINFO is a list containing the time series information.

DIMS Number of data rows

DTS Number of daily time steps (24 or 48)

Y.START Starting year

Y.END Ending year

Y.NUMS Number of years

Y.NAME Name for years

sLOCATION is a list of information on site location and timezone (see sEddyProc_sSetLocationInfo).

sTEMP is a data frame used only temporally.

Value

Initialized fields of sEddyProc.

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`sEddyProc_sCalcPotRadiation`
sEddyProc sCalcPotRadiation

Description

compute potential radiation from position and time

Usage

```
sEddyProc_sCalcPotRadiation(useSolartime.b = TRUE)
```

Arguments

`useSolartime.b`	by default corrects hour (given in local winter time) for latitude to solar time « (where noon is exactly at 12:00). Set this to FALSE to directly use local winter time

Value

column PotRad_NEW in sTEMP

Author(s)

Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`sEddyProc_sEstUstarThreshold`
sEddyProc$sEstUstarThreshold - Estimating ustar threshold

Description

Calling `usEstUstarThreshold` for class data and storing results

Usage

```
sEddyProc_sEstUstarThreshold(UstarColName = "Ustar",
    NEEColName = "NEE", TempColName = "Tair",
    RgColName = "Rg", ...)
```

Arguments

UstarColName	column name for UStar
NEEColName	column name for NEE
TempColName	column name for air temperature
RgColName	column name for solar radiation for omitting night time data
...	further arguments to usEstUstarThreshold

Value

result of usEstUstarThreshold. In addition the result is stored in class variable sUSTAR and the bins as additional columns to sDATA

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc_sEstUstarThresholdDistribution

sEddyProc sEstUstarThresholdDistribution

Description

Estimating the distribution of u * threshold by bootstrapping over data

Usage

```
sEddyProc_sEstUstarThresholdDistribution(ctrlUstarEst.l = usControlUstarEst(),
    ctrlUstarSub.l = usControlUstarSubsetting(),
    UstarColName = "Ustar", NEEColName = "NEE",
    TempColName = "Tair", RgColName = "Rg",
    ..., seasonFactor.v = usCreateSeasonFactorMonth(sDATA$sDateTime),
    seasonFactorsYear = usGetYearOfSeason(seasonFactor.v,
        ds$sDateTime), nSample = 100L, probs = c(0.05,
        0.5, 0.95), verbose.b = TRUE)
```

Arguments

`ctrlUstarEst.l` control parameters for estimating uStar on a single binned series, see `usControlUstarEst`

`ctrlUstarSub.l` control parameters for subsetting time series (number of temperature and Ustar classes ...), see `usControlUstarSubsetting`

`UstarColName` column name for UStar

`NEEColName` column name for NEE

`TempColName` column name for air temperature

`RgColName` column name for solar radiation for omitting night time data

`...` further arguments to `sEddyProc_sEstUstarThreshold`

`seasonFactor.v` factor of seasons to split (data is resampled only within the seasons)

`seasonFactorsYear` named integer vector: for each seasonFactor level, get the year that this season belongs to

`nSample` the number of repetitions in the bootstrap

`probs` the quantiles of the bootstrap sample to return. Default is the 5%, median and 95% of the bootstrap

`verbose.b` set to FALSE to omit printing progress

Details

The choice of the criterion for sufficiently turbulent conditions (u * > chosen threshold) introduces large uncertainties in calculations based on gap-filled Eddy data. Hence, it is good practice to compare derived quantities based on gap-filled data using a range of u * threshold estimates.

This method explores the probability density of the threshold by repeating its estimation on a bootstrapped sample. By default it returns the 90% confidence interval (argument `probs`). For larger intervals the sample number need to be increased (argument `probs`).

Quality Assurance If more than `ctrlUstarEst.l$minValidBootProp` (default 40%) did not report a threshold, no quantiles (i.e. NA) are reported.

Value

A data.frame with columns `aggregationMode`, `year`, and `UStar` estimate based on the non-resampled data. The other columns correspond to the quantiles of Ustar estimate for given probabilities (argument `probs`) based on the distribution of estimates using resampled the data.

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

sEddyProc_sEstUstarThreshold, sEddyProc_sMDSGapFillAfterUStarDistr

---

sEddyProc_sExportData *sEddyProc$sExportData - Export internal sDATA data frame*

Description

Export class internal sDATA data frame

Usage

```
sEddyProc_sExportData()
```

Value

Return data frame sDATA with time stamp shifted back to original.

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc_sExportResults *sEddyProc$sExportData - Export internal sTEMP data frame with result columns*

Description

Export class internal sTEMP data frame with result columns

Usage

```
sEddyProc_sExportResults(isListColumnsExported = FALSE)
```

Arguments

`isListColumnsExported`
: if TRUE export list columns in addition to numeric columns, such as the covariance matrices of the the day-time-partitioning LRC fits

Value

Return data frame sTEMP with results.

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc_sFillInit *sEddyProc$sFillInit - Initialize gap filling*

Description

Initializes data frame sTEMP for newly generated gap filled data and qualifiers.

Usage

```
sEddyProc_sFillInit(Var.s, QFVar.s = "none",
    QFValue.n = NA_real_, FillAll.b = TRUE)
```

Arguments

`Var.s`	Variable to be filled
`QFVar.s`	Quality flag of variable to be filled
`QFValue.n`	Value of quality flag for _good_ (original) data, other data is set to missing
`FillAll.b`	Fill all values to estimate uncertainties

Details

Description of newly generated variables with gap filled data and qualifiers:

VAR_*orig* - Original values used for gap filling
VAR_*f* - Original values and gaps filled with mean of selected datapoints (condition depending on gap filling method)
VAR_*fqc* - Quality flag assigned depending on gap filling method and window length (0 = original data, 1 = most reliable, 2 = medium, 3 = least reliable)
VAR_*fall* - All values considered as gaps (for uncertainty estimates)
VAR_*fall_qc* - Quality flag assigned depending on gap filling method and window length (1 = most reliable, 2 = medium, 3 = least reliable)
VAR_*fnum* - Number of datapoints used for gap-filling
VAR_*fsd* - Standard deviation of datapoints used for gap filling (uncertainty)
VAR_*fmeth* - Method used for gap filling (1 = similar meteo condition (sFillLUT with Rg, VPD,

Tair), 2 = similar meteo (sFillLUT with Rg only), 3 = mean diurnal course (sFillMDC))
VAR_*fwin* - Full window length used for gap filling

Long gaps (larger than 60 days) are not filled.

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc_sFillLUT — *sEddyProc$sFillLUT - Gap filling with Look-Up Table (LUT)*

Description

Look-Up Table (LUT) algorithm of up to five conditions within prescribed window size

Usage

```
sEddyProc_sFillLUT(WinDays.i, V1.s = "none",
    T1.n = NA_real_, V2.s = "none", T2.n = NA_real_,
    V3.s = "none", T3.n = NA_real_, V4.s = "none",
    T4.n = NA_real_, V5.s = "none", T5.n = NA_real_,
    Verbose.b = TRUE)
```

Arguments

`WinDays.i`	Window size for filling in days
`V1.s`	Condition variable 1
`T1.n`	Tolerance interval 1
`V2.s`	Condition variable 2
`T2.n`	Tolerance interval 2
`V3.s`	Condition variable 3
`T3.n`	Tolerance interval 3
`V4.s`	Condition variable 4
`T4.n`	Tolerance interval 4
`V5.s`	Condition variable 5
`T5.n`	Tolerance interval 5
`Verbose.b`	Print status information to screen

Details

Quality flags
- 1: at least one variable and nDay <= 14
- 2: three variables and nDay in [14,56) or one variable and nDay in [14,28)
- 3: three variables and nDay > 56 or one variable and nDay > 28

Value

LUT filling results in sTEMP data frame.

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc_sFillMDC — *sEddyProc$sFillMDC - Gap filling with Mean Diurnal Course (MDC)*

Description

Mean Diurnal Course (MDC) algorithm based on average values within +/- one hour of adjacent days

Usage

```
sEddyProc_sFillMDC(WinDays.i, Verbose.b = TRUE)
```

Arguments

`WinDays.i`	Window size for filling in days
`Verbose.b`	Print status information to screen

Details

Quality flag
- 1: nDay <= 1
- 2: nDay [2,5)
- 3: nDay > 5

Value

MDC filling results in sTEMP data frame.

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc_sGetData *sEddyProc$sGetData - Get internal sDATA data frame*

Description

Get class internal sDATA data frame

Usage

```
sEddyProc_sGetData()
```

Value

Return data frame sDATA.

Author(s)

AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc_sGLFluxPartition *sGLFluxPartition: Flux partitioning after Lasslop et al. (2010)*

Description

Daytime-based partitioning of measured net ecosystem fluxes into gross primary production (GPP) and ecosystem respiration (Reco)

Usage

```
sEddyProc_sGLFluxPartition(..., debug.l = list(useLocaltime.b = FALSE),
    isWarnReplaceColumns = TRUE)
```

Arguments

`...`	arguments to `partitionNEEGL` additional to the dataset `ds` such as `Suffix.s`
`debug.l`	List with debugging control. **useLocaltime.b** if TRUE use local time zone instead of geo-solar time to compute potential radiation
`isWarnReplaceColumns`	set to FALSE to avoid the warning on replacing output columns

Value

Flux partitioning results are in sTEMP data frame of the class.

Author(s)

MM, TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

References

Lasslop G, Reichstein M, Papale D, et al. (2010) Separation of net ecosystem exchange into assimilation and respiration using a light response curve approach: critical issues and global evaluation. Global Change Biology, Volume 16, Issue 1, Pages 187-208

`sEddyProc_sMDSGapFill` *sEddyProc$sMDSGapFill - MDS gap filling algorithm*

Description

MDS gap filling algorithm adapted after the PV-Wave code and paper by Markus Reichstein.

Usage

```
sEddyProc_sMDSGapFill(Var.s, QFVar.s = "none",
    QFValue.n = NA_real_, V1.s = "Rg", T1.n = 50,
    V2.s = "VPD", T2.n = 5, V3.s = "Tair",
    T3.n = 2.5, FillAll.b = TRUE, Verbose.b = TRUE,
    Suffix.s = "")
```

Arguments

Var.s	Variable to be filled
QFVar.s	Quality flag of variable to be filled
QFValue.n	Value of quality flag for _good_ (original) data, other data is set to missing
V1.s	Condition variable 1 (default: Global radiation 'Rg' in W m-2)
T1.n	Tolerance interval 1 (default: 50 W m-2)
V2.s	Condition variable 2 (default: Vapour pressure deficit 'VPD' in hPa)
T2.n	Tolerance interval 2 (default: 5 hPa)
V3.s	Condition variable 3 (default: Air temperature 'Tair' in degC)
T3.n	Tolerance interval 3 (default: 2.5 degC)
FillAll.b	Fill all values to estimate uncertainties
Verbose.b	Print status information to screen
Suffix.s	String suffix needed for different processing setups on the same dataset (for explanations see below)

Details

Initialize temporal data frame sTEMP for newly generated gap filled data and qualifiers, see `sEddyProc_sFillInit` for explanations on suffixes.

MDS gap filling algorithm calls the subroutines Look Up Table `sEddyProc_sFillLUT` and Mean Diurnal Course `sEddyProc_sFillMDC` with different window sizes as described in the reference.

To run dataset only with MDC algorithm `sEddyProc_sFillMDC`, set condition variable V1.s to 'none'.

Different processing setups on the same dataset Attention: When processing the same site data set with different setups for the gap filling or flux partitioning (e.g. due to different ustar filters), a string suffix is needed! This suffix is added to the result column names to distinguish the results of the different setups.

Value

Gap filling results in sTEMP data frame (with renamed columns).

Author(s)

AMM, TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

References

Reichstein, M. et al. (2005) On the separation of net ecosystem exchange into assimilation and ecosystem respiration: review and improved algorithm. Global Change Biology, 11, 1424-1439.

---

sEddyProc_sMDSGapFillAfterUstar
*sEddyProc$sMDSGapFillAfterUstar - MDS gap filling algorithm after u * filtering*

Description

Calling sEddyProc_sMDSGapFill after filtering for (provided) friction velocity u *

Usage

```
sEddyProc_sMDSGapFillAfterUstar(FluxVar.s,
    UstarVar.s = "Ustar", UstarThres.df = usGetAnnualSeasonUStarMap(sUSTAR$uStarTh),
     UstarSuffix.s = "WithUstar", FlagEntryAfterLowTurbulence.b = FALSE,
     isFilterDayTime = FALSE, swThr = 10,
     RgColName = "Rg", ...)
```

Arguments

`FluxVar.s`	Flux variable to gap fill after ustar filtering
`UstarVar.s`	Column name of friction velocity u * (ms-1), default 'Ustar'
`UstarThres.df`	data.frame with first column, season names, and second column estimates of uStar Threshold. « Alternatively, a single value to be used as threshold for all records
`UstarSuffix.s`	Different suffixes required for different u * scenarios
`FlagEntryAfterLowTurbulence.b`	Set to TRUE for flagging the first entry after low turbulance as bad condition (by value of 2).
`isFilterDayTime`	Set to TRUE to also filter day-time values, default only filters night-time data
`swThr`	threshold of solar radiation below which data is marked as night time respiration.
`RgColName`	Column name of incoming short wave radiation
`...`	Other arguments passed to sEddyProc_sMDSGapFill

Details

The u * threshold(s) are provided with argument `UstarThres.df` for filtering the conditions of low turbulence. After filtering, the data is gap filled using the MDS algorithm sEddyProc_sMDSGapFill.

With `isFlagEntryAfterLowTurbulence set to TRUE`, to be more conservative, in addition to the data acquired when uStar is below the threshold, the first half hour measured with good turbulence conditions after a period with low turbulence is also removed (Papaple et al. 2006).

Value

Vector with quality flag from filtering (here 0: good data, 1: low turbulence, 2: first half hour after low turbulence, 3: no threshold available, 4: missing uStar value) Gap filling results are in sTEMP data frame (with renamed columns) that can be retrieved by `sEddyProc_sExportResults`.

Author(s)

AMM, TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

- `sEddyProc_sMDSGapFillAfterUStarDistr` for automated gapfilling for several u * threshold estimates.
- `sEddyProc_sEstUstarThreshold` for estimating the u * threshold from the data.

---

sEddyProc_sMDSGapFillAfterUStarDistr

GapFilling for several filters of estimated friction velocity Ustar thresholds.

Description

sEddyProc$sMDSGapFillUStarDistr - calling `sEddyProc_sMDSGapFillAfterUstar` for several filters of friction velocity Ustar

Usage

```
sEddyProc_sMDSGapFillAfterUStarDistr(...,
    UstarThres.df, UstarSuffix.V.s = colnames(UstarThres.df)[-1])
```

Arguments

`...`	other arguments to `sEddyProc_sMDSGapFillAfterUstar` and `sEddyProc_sMDSGapFill`
`UstarThres.df`	data.frame with first column, season names, and remaining columns different estimates of uStar Threshold. If the data.frame has only one row, then each uStar threshold estimate is applied to the entire dataset. Entries in first column must match levels in argument `seasonFactor.v`
`UstarSuffix.V.s`	String vector to distinguish result columns for different ustar values. Its length must correspond to column numbers in `UstarThres.m.n`.

Details

The eddy covariance method does not work with low turbulence conditions. Hence the periods with low turbulence indicated by a low friction velocity u * needs to be filtered out and gapfilled (see sEddyProc_sMDSGapFill). The threshold value of a sufficient u * causes one of the largest uncertainty components within the gap-filled data. Hence, it is good practice to compare derived quantities based on gap-filled data using different u * threshold values.

For example a user could provide the the following columns in argument `UstarThres.df` and corresponding suffixes in argument `UstarSuffix.V.s`

- season: identifier for which season this row is used.
- Ustar: estimate on original series
- U05: 5% of bootstrap
- U50: median of bootstrap
- U95: 95% of bootstrap)

Then the difference between output columns NEE_U05_f and NEE_U95_f corresponds to the uncertainty introduced by the uncertain estimate of the u * threshold.

Value

Matrix (columns correspond to u * Scenarios) with quality flag from filtering ustar (0 - good data, 1 - filtered data)

Gap filling results in sTEMP data frame (with renamed columns), that can be retrieved by sEddyProc_sExportResults. Each of the columns is calculated for several u * r-estimates and distinguished by a suffix after the variable. E.g. with an first entry "U05" in `UstarSuffix.V.s` corresponding to the first column in `UstarThres.m.n`, the corresponding filled NEE can be found in output column "NEE_U05_f".

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

../doc/useCase.html

sEddyProc_sMRFluxPartition

sEddyProc$sMRFluxPartition - Flux partitioning after Reichstein et al. (2005)

Description

Nighttime-based partitioning of measured net ecosystem fluxes into gross primary production (GPP) and ecosystem respiration (Reco)

Usage

```
sEddyProc_sMRFluxPartition(FluxVar.s = "NEE_f",
    QFFluxVar.s = "NEE_fqc", QFFluxValue.n = 0,
    TempVar.s = "Tair_f", QFTempVar.s = "Tair_fqc",
    QFTempValue.n = 0, RadVar.s = "Rg", T_ref.n = 273.15 +
        15, Suffix.s = "", debug.l = list(useLocaltime.b = FALSE),
    parsE0Regression = list())
```

Arguments

FluxVar.s	Variable name of column with original and filled net ecosystem fluxes (NEE)
QFFluxVar.s	Quality flag of NEE variable
QFFluxValue.n	Value of quality flag for _good_ (original) data
TempVar.s	Filled air- or soil temperature variable (degC)
QFTempVar.s	Quality flag of filled temperature variable
QFTempValue.n	Value of temperature quality flag for _good_ (original) data
RadVar.s	Unfilled (original) radiation variable
T_ref.n	Reference temperature in Kelvin (degK) used in fLloydTaylor for regressing Flux and Temperature
Suffix.s	String suffix needed for different processing setups on the same dataset (for explanations see below)
debug.l	List with debugging control (passed also to sEddyProc_sRegrE0fromShortTerm). **useLocaltime.b** see details on solar vs local time
parsE0Regression	list with further parameters passed down to sEddyProc_sRegrE0fromShortTerm and fRegrE0fromShortTerm, such as TempRange.n

Details

Description of newly generated variables with partitioning results:
- PotRad - Potential radiation

- FP_NEEnight - Good (original) NEE nighttime fluxes used for flux partitioning
- FP_Temp - Good (original) temperature measurements used for flux partitioning
- E_0 - Estimated temperature sensitivity
- R_ref - Estimated reference respiration
- Reco - Estimated ecosystem respiration
- GPP_f - Estimated gross primary production

Background This partitioning is based on the regression of nighttime respiration with temperature using the Lloyd-Taylor-Function `fLloydTaylor`. First the temperature sensitivity E_0 is estimated from short term data, see `sEddyProc_sRegrE0fromShortTerm`. Next the reference temperature R_ref is estimated for successive periods throughout the whole dataset (see `sEddyProc_sRegrRref`). These estimates are then used to calculate the respiration during daytime and nighttime and with this GPP. Attention: Gap filling of the net ecosystem fluxes (NEE) and temperature measurements (Tair or Tsoil) is required prior to the partitioning!

Selection of daytime data based on solar time The respiration-temperature regression is very sensitive to the selection of night- and daytime data. Nighttime is selected by a combined threshold of current solar radiation and potential radiation. The current implementation calculates potential radiation based on exact solar time, based on latitude and longitude. (see `fCalcPotRadiation`) Therefore it might differ from implementations that use local winter clock time instead.

Different processing setups on the same dataset Attention: When processing the same site data set with different setups for the gap filling or flux partitioning (e.g. due to different ustar filters), a string suffix is needed! This suffix is added to the result column names to distinguish the results of the different setups. If a Suffix.s is provided and if the defaults for FluxVar.s and QFFluxVar.s are used, the Suffix.s will be added to their variable names (e.g. 'NEE_f' will be renamed to 'NEE_WithUstar_f' and 'NEE_fqc' to 'NEE_WithUstar_fqc' for the Suffix.s = 'WithUstar'). Currently, this works only with defaults of FluxVar.s = 'NEE_f' and QFFluxVar.s = 'NEE_fqc'.

Value

Flux partitioning results (see variables in details) in sTEMP data frame (with renamed columns). On success, return value is NULL. On failure an integer scalar error code is returned: -111 if regression of E_0 failed due to insufficient relationship in the data.

Author(s)

AMM, TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de>

[aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

References

Reichstein M, Falge E, Baldocchi D et al. (2005) On the separation of net ecosystem exchange into assimilation and ecosystem respiration: review and improved algorithm. Global Change Biology, 11, 1424-1439.

---

sEddyProc_sPlotDailySums

sEddyProc$sPlotDailySums - Image with daily sums of each year

---

Description

Generates image in specified format ('pdf' or 'png') with daily sums, see also `sEddyProc_sPlotDailySumsY`.

Usage

```
sEddyProc_sPlotDailySums(Var.s, VarUnc.s = "none",
    Format.s = "pdf", Dir.s = "plots", unit.s = "gC/m2/day",
    ...)
```

Arguments

`Var.s`	(Filled) variable to plot
`VarUnc.s`	Uncertainty estimates for variable
`Format.s`	Graphics file format ('pdf' or 'png')
`Dir.s`	Directory for plotting
`unit.s`	unit of the daily sums
`...`	further arguments to `sEddyProc_sPlotDailySumsY`, such as `timeFactor.n` and `massFactor.n`.

Author(s)

KS, AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

sEddyProc_sPlotDailySumsY
sEddyProc$sPlotDailySumsY - Plot daily sum of specified year

Description

The daily sums for a single year are plotted to the current device, scaled to all data. The daily sums are only calculated for days with complete data. This function first computes the average flux for each day. If the original unit is not "per day", then it need to be converted to "per day" by argument `timeFactor.n`. Furthermore, a change of the mass unit is provided by argument `massFactor.n`. The default parameters assume original units of mumol CO2 / m2 / second and convert to gC / m2 / day. The conversion factors allow plotting variables with different units

Usage

```
sEddyProc_sPlotDailySumsY(Var.s, VarUnc.s = "none",
    Year.i, timeFactor.n = 3600 * 24, massFactor.n = (44.0096/1e+06) *
        (12.011/44.0096), unit.s = "gC/m2/day")
```

Arguments

`Var.s`	(Filled) variable to plot
`VarUnc.s`	Uncertainty estimates for variable
`Year.i`	Year to plot
`timeFactor.n`	time conversion factor with default per second to per day
`massFactor.n`	mass conversion factor with default from mumol CO2 to g C
`unit.s`	resulting unit

Author(s)

AMM, KS Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

sEddyProc_sPlotDiurnalCycle
sEddyProc$sPlotDiurnalCycle - Image with diurnal cycles of each month

Description

Generates image in specified format ('pdf' or 'png') with diurnal cycles.

Usage

```
sEddyProc_sPlotDiurnalCycle(Var.s, QFVar.s = "none",
    QFValue.n = NA_real_, Format.s = "pdf",
    Dir.s = "plots")
```

Arguments

Var.s	Variable to plot
QFVar.s	Quality flag of variable to be filled
QFValue.n	Value of quality flag for data to plot
Format.s	Graphics file format ('pdf' or 'png')
Dir.s	Directory for plotting

Author(s)

KS, AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

sEddyProc_sPlotFingerprint
sEddyProc$sPlotFingerprint - Image with fingerprints of each year

Description

Generates image in specified format Format.s (e.g. 'pdf' or 'png') with fingerprint, see also sEddyProc_sPlotFingerprintY.

Usage

```
sEddyProc_sPlotFingerprint(Var.s, QFVar.s = "none",
    QFValue.n = NA_real_, Format.s = "pdf",
    Dir.s = "plots", ...)
```

Arguments

Var.s	Variable to plot
QFVar.s	Quality flag of variable to be filled
QFValue.n	Value of quality flag for data to plot
Format.s	Graphics file format (e.g. 'pdf', 'png')
Dir.s	Directory for plotting
...	further arguments to sEddyProc_sPlotFingerprintY

Author(s)

KS, AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc_sPlotFingerprintY

sEddyProc$sPlotFingerprintY - Plot fingerprint of specified year

---

Description

The fingerprint for a single year is plotted to the current device, scaled to all data.

Usage

```
sEddyProc_sPlotFingerprintY(Var.s, QFVar.s = "none",
    QFValue.n = NA_real_, Year.i, Legend.b = F,
    Col.V = colorRampPalette(c("#00007F",
        "blue", "#007FFF", "cyan", "#7FFF7F",
        "yellow", "#FF7F00", "red", "#7F0000"))(50),
    valueLimits = range(Plot.V.n, na.rm = TRUE))
```

Arguments

Var.s	Variable to plot
QFVar.s	Quality flag of variable to be filled
QFValue.n	Value of quality flag for data to plot
Year.i	Year to plot
Legend.b	Plot only legend
Col.V	Color palette for fingerprint plot (can be also defined by user), i.e. color scale argument to image
valueLimits	values outside this range will be set to the range borders to avoid distorting colour scale e.g. valueLimits = quantile(EddyProc.C$sDATA$NEE, prob = c(0.05, 0.95), na.rm = TRUE)

Author(s)

AMM, KS, TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`sEddyProc_sPlotHHFluxes`

sEddyProc sPlotHHFluxes

Description

Produce image-plot with half-hourly fluxes for each year

Usage

```
sEddyProc_sPlotHHFluxes(Var.s, QFVar.s = "none",
    QFValue.n = NA_real_, Format.s = "pdf",
    Dir.s = "plots")
```

Arguments

`Var.s`	(Filled) variable to plot
`QFVar.s`	Quality flag of variable to be filled
`QFValue.n`	Value of quality flag for data to plot
`Format.s`	Graphics file format ('pdf' or 'png')
`Dir.s`	Directory for plotting

Details

Generates image in specified format ('pdf' or 'png') with half-hourly fluxes and their daily means, see also `sEddyProc_sPlotHHFluxesY`.

Author(s)

KS, AMM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`sEddyProc_sPlotHHFluxesY`
sEddyProc$sPlotHHFluxesY - Plot half-hourly fluxes of specified year

---

Description

The half-hourly fluxes for a single year are plotted to the current device, scaled to all data.

Usage

```
sEddyProc_sPlotHHFluxesY(Var.s, QFVar.s = "none",
    QFValue.n = NA_real_, Year.i)
```

Arguments

`Var.s`	Variable to plot
`QFVar.s`	Quality flag of variable to be filled
`QFValue.n`	Value of quality flag for data to plot
`Year.i`	Year to plot

Author(s)

AMM, KS Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`sEddyProc_sPlotNEEVersusUStarForSeason`
sEddyProc$sPlotNEEVersusUStarForSeason - Image with NEE versus UStar for each Temperature class of given season

---

Description

Generates image in specified format ('pdf' or 'png')

Usage

```
sEddyProc_sPlotNEEVersusUStarForSeason(season = levels(sDATA$season)[1],
    format = "pdf", dir = "plots", UstarColName = "Ustar",
    NEEColName = "NEE", TempColName = "Tair",
    WInch = 16 * 0.394, HInchSingle = 6 *
        0.394, ...)
```

Arguments

`season`	string of season, i.e. time period, to plot
`format`	string of Graphics file format ('pdf' or 'png')
`dir`	string of Directory for plotting
`UstarColName`	column name for UStar
`NEEColName`	column name for NEE
`TempColName`	column name for air temperature
`WInch`	width of the plot in inches, defaults to 16cm
`HInchSingle`	height of a subplot in inches, defaults to 6cm
`...`	other arguments to `.plotNEEVersusUStarTempClass`, such as xlab and ylab axis label strings

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`sEddyProc_sSetLocationInfo`

sEddyProc sSetLocationInfo

Description

set Location and time Zone information to sLOCATION

Usage

```
sEddyProc_sSetLocationInfo(Lat_deg.n, Long_deg.n,
    TimeZone_h.n)
```

Arguments

`Lat_deg.n`	Latitude in (decimal) degrees (-90 to + 90)
`Long_deg.n`	Longitude in (decimal) degrees (-180 to + 180)
`TimeZone_h.n`	Time zone (in hours) shift to UTC, e.g. + 1 for Berlin

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`usControlUstarEst` *usControlUstarEst*

Description

Default list of parameters for determining UStar of a single binned series

Usage

```
usControlUstarEst(ustPlateauFwd = 10, ustPlateauBack = 6,
    plateauCrit = 0.95, corrCheck = 0.5,
    firstUStarMeanCheck = 0.2, isOmitNoThresholdBins = TRUE,
    isUsingCPTSeveralT = FALSE, isUsingCPT = FALSE,
    minValidUStarTempClassesProp = 0.2, minValidBootProp = 0.4,
    minNuStarPlateau = 3L)
```

Arguments

`ustPlateauFwd` number of subsequent uStar bin values to compare to in fwd mode

`ustPlateauBack` number of subsequent uStar bin values to compare to in back mode

`plateauCrit` significant differences between a uStar value and the mean of a "plateau"

`corrCheck` threshold value for correlation between Tair and u * data

`firstUStarMeanCheck`
: if first uStar bin average of a class is already larger than this value, the temperature class is skipped.

`isOmitNoThresholdBins`
: if TRUE, bins where no threshold was found are ignored. Set to FALSE to report highest uStar bin for these cases

`isUsingCPTSeveralT`
: set to TRUE to use changePointDetection without binning uStar but with additionally changed aggregation scheme for several temperature classifications

`isUsingCPT` set to TRUE to use changePointDetection without binning uStar before in usual aggregation method (good for comparing methods, but not recommended, overruled by isUsingCPTSeveralT = TRUE)

`minValidUStarTempClassesProp`
: seasons, in which only less than this proportion of temperature classes a threshold was detected, are excluded from aggregation

minValidBootProp
: minimum proportion of bootstrap samples for which a threshold was detected. Below this proportion NA quantiles are reported.

minNuStarPlateau
: minimum number of records in plateau, threshold must be larger than mean of this many bins

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

usEstUstarThresholdSingleFw2Binned, usControlUstarSubsetting

Examples

```
usControlUstarEst()
```

---

usControlUstarSubsetting *usControlUstarSubsetting*

Description

Default list of parameters for subsetting the data for uStarThreshold estimation

Usage

```
usControlUstarSubsetting(taClasses = 7, UstarClasses = 20,
    swThr = 10, minRecordsWithinTemp = 100,
    minRecordsWithinSeason = 160, minRecordsWithinYear = 3000,
    isUsingOneBigSeasonOnFewRecords = TRUE)
```

Arguments

taClasses
: set number of air temperature classes

UstarClasses
: set number of Ustar classes

swThr
: nighttime data threshold for solar radiation [Wm-2]

minRecordsWithinTemp
: integer scalar: the minimum number of Records within one Temperature-class

`minRecordsWithinSeason`
: integer scalar: the minimum number of Records within one season

`minRecordsWithinYear`
: integer scalar: the minimum number of Records within one year

`isUsingOneBigSeasonOnFewRecords`
: boolean scalar: set to FALSE to avoid aggregating all seasons on too few records

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

`usEstUstarThresholdSingleFw2Binned`, `usControlUstarSubsetting`

Examples

```
usControlUstarSubsetting()
```

`usCreateSeasonFactorMonth` *usCreateSeasonFactorMonth*

Description

Compute year-spanning Seasonfactor by starting month

Usage

```
usCreateSeasonFactorMonth(dates, month = as.POSIXlt(dates)$mon +
    1L, year = as.POSIXlt(dates)$year + 1900L,
    startMonth = c(3, 6, 9, 12))
```

Arguments

`dates`	POSIXct vector of length of the data set to be filled, specifying the center-time of each record
`month`	integer (1-12) vector of length of the data set to be filled, specifying the month for each record
`year`	integer vector of length of the data set to be filled, specifying the year
`startMonth`	integer vector specifying the starting month for each season, counting from one. Default is (Dez, Jan, Feb)(Mar, April, May)(June, July, August), (Sept, Oct, Nov)

Details

Compute factors to denote the season for uStar-Filtering by specifying starting months, with continuous seasons spanning year boundaries If Jan is not a starting month, then the first months of each year will be part of the last period in the year. E.g. with the default the fourth period of the first year consists of Jan, Feb, Dec.

REddyProc internally works with a timestamp 15 minutes after the start of each half hour. When providing the `dates` argument, user may shift the start time by `dates = myDataset$DateTime + 15 * 60`

Value

Integer vector length(dates), with each unique value representing one season

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

`usCreateSeasonFactorMonthWithinYear`, `usCreateSeasonFactorYday`, `usCreateSeasonFactorYdayYear`

`usCreateSeasonFactorMonthWithinYear`

usCreateSeasonFactorMonthWithinYear

Description

Compute year-bounded Seasonfactor by starting month

Usage

```
usCreateSeasonFactorMonthWithinYear(dates,
    month = as.POSIXlt(dates)$mon + 1, year = as.POSIXlt(dates)$year +
        1900, startMonth = c(3, 6, 9, 12))
```

Arguments

`dates`	POSIXct vector of length of the data set to be filled, specifying the center-time of each record
`month`	integer (1-12) vector of length of the data set to be filled, specifying the month for each record
`year`	integer vector of length of the data set to be filled, specifying the year

`startMonth` integer vector specifying the starting month for each season, counting from one. Default is (Dez, Jan, Feb)(Mar, April, May)(June, July, August), (Sept, Oct, Nov)

Details

Calculate factors to denote the season for uStar-Filtering by specifying starting months, with seasons not spanning year boundaries If Jan is not a starting month, then the first months of each year will be part of the last period in the year. E.g. with the default the fourth period of the first year consists of Jan, Feb, Dec.

Value

Integer vector length(dates), with each unique value representing one season

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

`usCreateSeasonFactorMonth`

---

usCreateSeasonFactorYday

usCreateSeasonFactorYday

Description

Compute year-spanning Seasonfactor by starting year-day

Usage

```
usCreateSeasonFactorYday(dates, yday = as.POSIXlt(dates)$yday +
    1L, year = as.POSIXlt(dates)$year + 1900L,
    startYday = c(335, 60, 152, 244))
```

Arguments

`dates`	POSIXct vector of length of the data set to be filled, specifying the center-time of each record
`yday`	integer (1-366) vector of length of the data set to be filled, specifying the day of the year (1..366) for each record
`year`	integer vector of length of the data set to be filled, specifying the year
`startYday`	integer vector (1-366) specifying the starting yearDay for each season in increasing order

Details

With default parameterization, dates are assumed to denote begin or center of the eddy time period.
If working with dates that denote the end of the period, use `yday = as.POSIXlt(fGetBeginOfEddyPeriod(dates))$yday`

Value

Integer vector of length `nrow(ds)`, each unique class representing one season

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

`usCreateSeasonFactorMonth`

---

usCreateSeasonFactorYdayYear

usCreateSeasonFactorYdayYear

Description

Compute year-spanning Seasonfactor by starting year and yearday

Usage

```
usCreateSeasonFactorYdayYear(dates, yday = as.POSIXlt(dates)$yday +
    1L, year = as.POSIXlt(dates)$year + 1900L,
    starts)
```

Arguments

dates	POSIXct vector of length of the data set to be filled, specifying the center-time of each record
yday	integer (1-366) vector of length of the data set to be filled, specifying the day of the year (1..366) for each record
year	integer vector of length of the data set to be filled, specifying the year
starts	data.frame with first column specifying the starting yday (integer 1-366) and second column the year (integer e.g. 1998) for each season in increasing order

Details

With default parameterization, dates are assumed to denote begin or center of the eddy time period. If working with dates that denote the end of the period, use `yday = as.POSIXlt(fGetBeginOfEddyPeriod(dates))$yday`

Value

Integer vector of length `nrow(ds)`, each unique class representing one season

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

See Also

`usCreateSeasonFactorMonth`

---

usEstUstarThreshold *usEstUstarThreshold - Estimating ustar threshold*

Description

Estimate the Ustar threshold by aggregating the estimates for seasonal and temperature subsets.

Usage

```
usEstUstarThreshold(ds, seasonFactor.v = usCreateSeasonFactorMonth(ds$sDateTime),
    seasonFactorsYear = usGetYearOfSeason(seasonFactor.v,
        ds$sDateTime), ctrlUstarEst.l = usControlUstarEst(),
    ctrlUstarSub.l = usControlUstarSubsetting(),
    fEstimateUStarBinned = usEstUstarThresholdSingleFw2Binned,
    isCleaned = FALSE)
```

Arguments

`ds` data.frame with columns "sDateTime", "Ustar", "NEE", "Tair", and "Rg"

`seasonFactor.v` factor for subsetting times (see details)

`seasonFactorsYear` named integer vector: for each seasonFactor level, get the year (aggregation period) that this season belongs to

`ctrlUstarEst.l` control parameters for estimating uStar on a single binned series, see usControlUstarEst

`ctrlUstarSub.l` control parameters for subsetting time series (number of temperature and Ustar classes ...), see usControlUstarSubsetting

`fEstimateUStarBinned` function to estimate UStar on a single binned series, see usEstUstarThresholdSingleFw2Binned

`isCleaned` set to TRUE, if the data was cleaned already, to avoid expensive call to usGetValidUstarIndices.

Details

The threshold for sufficiently turbulent conditions u * (Ustar) is estimated for different subsets of the time series. From the estimates for each season (each value in seasonFactor.v) the maximum of all seasons of one year is reported as estimate for this year. Within each season the time series is split by temperature classes. Among these Ustar estimates, the median is reported as season value.

In order to split the seasons, the uses must provide a vector with argument seasonFactor.v. All positions with the same factor, belong to the same season. It is conveniently generated by one of the following functions:

- usCreateSeasonFactorMonth (default DJF-MAM-JJA-SON with December from previous to January of the year)
- usCreateSeasonFactorMonthWithinYear (default DJF-MAM-JJA-SON with December from the same year)
- usCreateSeasonFactorYday for a refined specification of season starts.
- usCreateSeasonFactorYdayYear for specifying different seasons season between years.

The estimation of Ustar on a single binned series can be selected argument fEstimateUStarBinned.

- usEstUstarThresholdSingleFw1Binned
- usEstUstarThresholdSingleFw2Binned (default)

This function is called by

- sEddyProc_sEstUstarThreshold which stores the result in the class variables (sUSTAR and sDATA).
- sEddyProc_sEstUstarThresholdDistribution which additionally estimates median and confidence intervals for each year by bootstrapping the original data within seasons.

For inspecting the NEE~uStar relationship plotting is provided by sEddyProc_sPlotNEEVersusUStarForSeason

change point detection (CPT) method With specifying `ctrlUstarEst.l = usControlUstarEst(isUsingCPTSeveralT` change point detection is applied instead of the moving point test (e.g. with Fw2Binned).

The sometimes sensitive binning of uStar values within a temperature class is avoided. Further, possible spurious thresholds are avoid by testing that the model with a threshold fits the data better than a model without a threshold using a likelihood ratio test. In addition, with CPT seasons are excluded where a threshold was detected in only less than ctrlUstarEst.l$minValidUStarTempClassesProp (default 20%) of the temperature classes.

Note, that this method often gives higher estimates of the u * threshold.

One-big-season fallback If there are too few records within one year, of when no season yielded a finite u * Threshold estimate, then the yearly u * Th is estimated by pooling the data from seasons within one seasonYear. The user can suppress using pooled data on few records by providing option `ctrlUstarSub.l$isUsingOneBigSeasonOnFewRecords = FALSE` (see `usControlUstarSubsetting`)

Value

A list with entries data.frame with columns "aggregationMode", "seasonYear", "season", "uStar" with rows for "single": the entire aggregate (median across years) , "seasonYear": each year (maximum across seasons or estimate on pooled data) , "season": each season (median across temperature classes)

`seasonYear` data.frame listing results for year with columns "seasonYear" , "uStarMaxSeason" the maximum across seasonal estimates within the year , "uStarPooled" the estimate based on data pooled across the year (only calculated on few valid records or on uStarMaxSeason was nonfinite) , "nRec" number of valid records (only if the pooled estimate was calculated) , "uStarAggr" chosen estimate, corresponding to uStarPooled if this was calculated, or uStarMaxSeason or uStarTh across years if the former was non-finite

`season` data.frame listing results for each season , "nRec" the number of valid records , "uStarSeasonEst" the estimate for based on data within the season (median across temperature classes) , "uStarAggr" chose estimate, corresponding to uStarSeasonEst, or the yearly seasonYear$uStarAggr, if the former was non-finite

`tempInSeason` numeric matrix (nTemp x nAggSeason): estimates for each temperature subset for each season

`bins` columns `season`, `tempBin` and `uStarBin` for each record of input `ds` reporting classes of similar environmental conditions that the record belongs to.

Author(s)

TW, OM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

References

Ustar filtering following the idea in Papale, D. et al. (2006) Towards a standardized processing of net ecosystem exchange measured with eddy covariance technique: algorithms and uncertainty estimation. Biogeosciences 3(4): 571-583.

---

usEstUstarThresholdSingleFw1Binned

usEstUstarThresholdSingleFw1Binned

Description

estimate the Ustar threshold for single subset, using FW1 algorithm

Usage

```
usEstUstarThresholdSingleFw1Binned(Ust_bins.f,
    ctrlUstarEst.l = usControlUstarEst())
```

Arguments

`Ust_bins.f`	data.frame with columns NEE_avg and Ust_avg, of Ustar bins
`ctrlUstarEst.l`	parameter list, see `usControlUstarEst` for defaults and description

Details

Relying on binned NEE and Ustar

Author(s)

TW, OM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

References

inspired by Papale 2006

`usEstUstarThresholdSingleFw2Binned`
usEstUstarThresholdSingleFw2Binned

Description

estimate the Ustar threshold for single subset, using FW2 algorithm

Usage

```
usEstUstarThresholdSingleFw2Binned(Ust_bins.f,
    ctrlUstarEst.l = usControlUstarEst())
```

Arguments

`Ust_bins.f`	data.frame with column s NEE_avg and Ust_avg, of Ustar bins
`ctrlUstarEst.l`	parameter list, see `usControlUstarEst` for defaults and description

Details

Demand that threshold is higher than `ctrlUstarEst.l$minNuStarPlateau` records. If fewer records

Author(s)

TW, OM Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`usGetAnnualSeasonUStarMap`
usGetAnnualSeasonUStarMap

Description

extract mapping season -> uStar columns from Distribution result

Usage

```
usGetAnnualSeasonUStarMap(uStarTh)
```

Arguments

`uStarTh` result of `sEddyProc_sEstUstarThresholdDistribution` or `sEddyProc_sEstUstarThreshold`$uStarT

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`usGetSeasonalSeasonUStarMap`

usGetSeasonalSeasonUStarMap

Description

extract mapping season -> uStar columns from Distribution result

Usage

```
usGetSeasonalSeasonUStarMap(uStarTh)
```

Arguments

`uStarTh` result of `sEddyProc_sEstUstarThresholdDistribution` or `sEddyProc_sEstUstarThreshold`$uStarT

Details

from result of `sEddyProc_sEstUstarThresholdDistribution`

Value

a data frame with first column the season, and other columns different uStar threshold estimates

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

`usGetYearOfSeason` *usGetYearOfSeason*

Description

determine the year of the record of middle of seasons

Usage

```
usGetYearOfSeason(seasonFactor.v, sDateTime.v)
```

Arguments

`seasonFactor.v`	factor vector of length data: for each record which season it belongs to
`sDateTime.v`	POSIX.t vector of length data: for each record: center of half-hour period (corresponding to sDATA$sDateTime)

Value

named integer vector, with names corresponding to seasons

Author(s)

TW Department for Biogeochemical Integration at MPI-BGC, Jena, Germany <REddyProc-help@bgc-jena.mpg.de> [cph], Thomas Wutzler <twutz@bgc-jena.mpg.de> [aut, cre], Markus Reichstein <mreichstein@bgc-jena.mpg.de> [aut], Antje Maria Moffat <antje.moffat@bgc.mpg.de> [aut, trl], Olaf Menzer <omenzer@bgc-jena.mpg.de> [ctb], Mirco Migliavacca <mmiglia@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav Šigut <sigut.l@czechglobe.cz> [ctb]

Index