

# Package ‘REddyProc’

December 1, 2021

**Type** Package

**Version** 1.3.1

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

**Description** Standard and extensible Eddy-Covariance data post-processing

(Wutzler et al. (2018) <[doi:10.5194/bg-15-5015-2018](https://doi.org/10.5194/bg-15-5015-2018)>)

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](https://doi.org/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](https://doi.org/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](https://doi.org/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](https://doi.org/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** 7.1.2

**VignetteBuilder** knitr

**LinkingTo** Rcpp

**Depends** R (>= 3.0.0), methods

**Imports** Rcpp, dplyr, purrr, rlang, readr, mlegp, tibble, magrittr, solartime, bigleaf (>= 0.7)

**Suggests** testthat, minpack.lm, segmented, knitr, rmarkdown, lognorm, ggplot2, tidyr, markdown

**Collate** 'CheckVal.R' 'DataFunctions.R' 'aEddy.R' 'EddyGapfilling.R' 'EddyPartitioning.R' 'EddyPlotting.R' 'EddyUStarFilterChangePointDetection.R' 'EddyUStarFilterDPR.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' 'estimate\_vpd\_from\_dew.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 <U+0160>igut [ctb]

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

**Repository** CRAN

**Date/Publication** 2021-12-01 14:10:02 UTC

## R topics documented:

REddyProc-package	5
BerkeleyJulianDateToPOSIXct	7
DEGebExample	8
estimate_vpd_from_dew	8
Example_DETha98	9
extract_FN15	10
fCalcAVPfromVMFandPress	10
fCalcETfromLE	11
fCalcExtRadiation	12
fCalcPotRadiation	13
fCalcRHfromAVPandTair	14
fCalcSVPfromTair	15
fCalcVPDfromRHAndTair	15
fCheckHHTimeSeries	16
fConvertCtoK	17
fConvertGlobalToVisible	18
fConvertKtoC	18

fConvertTimeToPosix . . . . .	19
fConvertVisibleWm2toPhotons . . . . .	21
filterLongRuns . . . . .	21
filterLongRunsInVector . . . . .	22
fLloydTaylor . . . . .	23
fLoadEuroFlux16 . . . . .	24
fLoadFluxnet15 . . . . .	25
fLoadTXTIntoDataframe . . . . .	26
fWriteDataframeToFile . . . . .	27
getAmerifluxToBGC05VariableNameMapping . . . . .	28
getBGC05ToAmerifluxVariableNameMapping . . . . .	29
getExamplePath . . . . .	30
getFilledExampleDETha98Data . . . . .	31
getREddyProcExampleDir . . . . .	32
getTZone . . . . .	33
get_timestep_hours . . . . .	33
globalDummyVars . . . . .	34
LightResponseCurveFitter . . . . .	34
LightResponseCurveFitter-class . . . . .	35
LightResponseCurveFitter_computeCost . . . . .	36
LightResponseCurveFitter_computeLRCGradient . . . . .	37
LightResponseCurveFitter_fitLRC . . . . .	38
LightResponseCurveFitter_getOptimizedParameterPositions . . . . .	39
LightResponseCurveFitter_getParameterInitials . . . . .	40
LightResponseCurveFitter_getParameterNames . . . . .	41
LightResponseCurveFitter_getPriorLocation . . . . .	41
LightResponseCurveFitter_getPriorScale . . . . .	42
LightResponseCurveFitter_isParameterInBounds . . . . .	43
LightResponseCurveFitter_optimLRC . . . . .	44
LightResponseCurveFitter_optimLRCBounds . . . . .	45
LightResponseCurveFitter_optimLRConAdjustedPrior . . . . .	46
LightResponseCurveFitter_predictGPP . . . . .	47
LightResponseCurveFitter_predictLRC . . . . .	48
LogisticSigmoidLRCFitter . . . . .	49
LogisticSigmoidLRCFitter-class . . . . .	49
LogisticSigmoidLRCFitter_predictGPP . . . . .	50
NonrectangularLRCFitter . . . . .	51
NonrectangularLRCFitter-class . . . . .	51
NonrectangularLRCFitter_getParameterNames . . . . .	52
NonrectangularLRCFitter_predictGPP . . . . .	53
partGLControl . . . . .	54
partGLControlLasslopCompatible . . . . .	56
partGLExtractStandardData . . . . .	58
partitionNEEGL . . . . .	60
POSIXctToBerkeleyJulianDate . . . . .	63
read_from_fluxnet15 . . . . .	63
RectangularLRCFitter . . . . .	64
RectangularLRCFitter-class . . . . .	65

RectangularLRCFitterCVersion . . . . .	65
RectangularLRCFitterCVersion-class . . . . .	66
RectangularLRCFitter_predictGPP . . . . .	66
renameVariablesInDataframe . . . . .	67
RHLightResponseCostC . . . . .	68
sEddyProc . . . . .	69
sEddyProc-class . . . . .	69
sEddyProc_initialize . . . . .	71
sEddyProc_sApplyUStarScen . . . . .	73
sEddyProc_sCalcPotRadiation . . . . .	74
sEddyProc_sEstimateUstarScenarios . . . . .	74
sEddyProc_sEstUstarThold . . . . .	76
sEddyProc_sEstUstarThreshold . . . . .	77
sEddyProc_sEstUstarThresholdDistribution . . . . .	78
sEddyProc_sExportData . . . . .	78
sEddyProc_sExportResults . . . . .	79
sEddyProc_sFillInit . . . . .	80
sEddyProc_sFillLUT . . . . .	81
sEddyProc_sFillMDC . . . . .	82
sEddyProc_sFillVPDFromDew . . . . .	83
sEddyProc_sGetData . . . . .	83
sEddyProc_sGetEstimatedUstarThresholdDistribution . . . . .	84
sEddyProc_sGetUstarScenarios . . . . .	84
sEddyProc_sGetUstarSuffixes . . . . .	85
sEddyProc_sGLFluxPartition . . . . .	86
sEddyProc_sGLFluxPartitionUStarScens . . . . .	87
sEddyProc_sMDSGapFill . . . . .	88
sEddyProc_sMDSGapFillAfterUstar . . . . .	90
sEddyProc_sMDSGapFillAfterUstarDistr . . . . .	91
sEddyProc_sMDSGapFillUStarScens . . . . .	92
sEddyProc_sMRFluxPartition . . . . .	93
sEddyProc_sMRFluxPartitionUStarScens . . . . .	95
sEddyProc_sPlotDailySums . . . . .	96
sEddyProc_sPlotDailySumsY . . . . .	97
sEddyProc_sPlotDiurnalCycle . . . . .	98
sEddyProc_sPlotFingerprint . . . . .	99
sEddyProc_sPlotFingerprintY . . . . .	100
sEddyProc_sPlotHHFluxes . . . . .	101
sEddyProc_sPlotHHFluxesY . . . . .	102
sEddyProc_sPlotNEEVersusUStarForSeason . . . . .	103
sEddyProc_sSetLocationInfo . . . . .	104
sEddyProc_sSetUstarScenarios . . . . .	105
sEddyProc_sSetUstarSeasons . . . . .	106
sEddyProc_sTKFluxPartition . . . . .	106
sEddyProc_sTKFluxPartitionUStarScens . . . . .	107
sEddyProc_update_ustarthreshold_columns . . . . .	108
sEddyProc_useAnnualUStarThresholds . . . . .	108
sEddyProc_useSeaonsalUStarThresholds . . . . .	109

usControlUstarEst . . . . .	109
usControlUstarSubsetting . . . . .	111
usCreateSeasonFactorMonth . . . . .	112
usCreateSeasonFactorMonthWithinYear . . . . .	113
usCreateSeasonFactorYday . . . . .	114
usCreateSeasonFactorYdayYear . . . . .	115
usEstUstarThreshold . . . . .	116
usEstUstarThresholdSingleFw1Binned . . . . .	118
usEstUstarThresholdSingleFw2Binned . . . . .	119
usGetAnnualSeasonUStarMap . . . . .	120
usGetSeasonalSeasonUStarMap . . . . .	120
usGetYearOfSeason . . . . .	121

<b>Index</b>	<b>122</b>
--------------	------------

---

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

---

## Description

Standard and extensible Eddy-Covariance data post-processing including uStar-filtering, gap-filling, and flux-partitioning (Wutzler et al. (2018) <doi:10.5194/bg-15-5015-2018>).

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](#)

Processing across different scenarios of  $u^*$  threshold estimate is supported by

- Estimating the turbulence criterion, Ustar threshold, for omitting data from periods of low turbulence: [sEddyProc\\_sEstimateUstarScenarios](#) and associated
  - query the thresholds to be used [sEddyProc\\_sGetUstarScenarios](#)
  - set the thresholds to be used [sEddyProc\\_sSetUstarScenarios](#)
  - query the estimated thresholds all different aggregation levels [sEddyProc\\_sGetEstimatedUstarThresholdDistribution](#)
- Gap-Filling: [sEddyProc\\_sMDSGapFillUstarScens](#)
- Flux partitioning based on Night-Time (Reichstein 2005): [sEddyProc\\_sMRFluxPartitionUstarScens](#)
- Flux partitioning based on Day-Time (Lasslop 2010): [sEddyProc\\_sGLFluxPartitionUstarScens](#)
- Flux partitioning based on modified Day-Time (Keenan 2019): [sEddyProc\\_sTKFluxPartitionUstarScens](#)

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 = "UTC",  
  ...)
```

## Arguments

julianDate	numeric vector representing times (see details for format)
tz	time zone used to represent the dates
...	further arguments to <a href="#">strptime</a>

## 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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

## See Also

[POSIXctToBerkeleyJulianDate](#) [fConvertTimeToPosix](#)

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 REdyProc 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.

---

estimate\_vpd\_from\_dew *Estimate VPD from assuming dewpoint at daily minimum temperature*

---

**Description**

VPD is required for daytime NEE flux partitioning. Hence, it is necessary to estimate VPD also for long gaps in data. With two assumptions, VPD can be estimated from temperature 1). The change of water mass in air is negligible during the day. VPD is the difference of actual vapour pressure to saturation vapour pressure. 2.) At morning minimum temperature, vapour pressure is at minimum in many cases at saturation. Hence

$$VPD = Esat(Tair) - E \approx Esat(Tair) - Esat_{daymin} \approx Esat(Tair) - Esat(Tair_{min})$$

**Usage**

```
estimate_vpd_from_dew(df, pNonMissing = 0.1)
```

**Arguments**

**df** data.frame with columns DateTime, VPD, Tair, and Tair\_f  
**pNonMissing** numeric scalar of the necessary fraction of finite VPD and Tair. If fraction is lower then a warning is thrown.

**Details**

Since sometimes Esat\_daymin is lower than Esat(Tair\_min) the estimated VPDfromDew is underestimated. This function applies a linear model of the existing VPD and estimated VPD to correct for this bias:  $VPD \sim 0 + VPD_{fromDew} * Tair\_f * hourOfDay * TminOftheDay * TRangeDay$

**Value**

numeric vector of length(nrow(data)) of estimated VPD

---

Example_DETha98	<i>Eddy covariance data from Tharandt, Germany</i>
-----------------	--

---

**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.

---

extract_FN15	<i>extract processing results with columns corresponding to Fluxnet15 release</i>
--------------	---

---

**Description**

extract processing results with columns corresponding to Fluxnet15 release

**Usage**

```
extract_FN15(
  EProc = .self,
  is_export_nonfilled = TRUE,
  keep_other_cols = FALSE
)
```

**Arguments**

EProc	sEddyProc class with uncertainty also in meteo variables and both nighttime and daytime partitioning columns present
is_export_nonfilled	set to FALSE to not export columns before gapfilling
keep_other_cols	set to TRUE to report also other columns

**Value**

data.frame with columns names of Fluxnet15. Timestamps are in ISO string format [POSIXctToBerkeleyJulianDate](#)

---

fCalcAVPfromVMFandPress	<i>fCalcAVPfromVMFandPress</i>
-------------------------	--------------------------------

---

**Description**

Calculate AVP from VMF and Press

**Usage**

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

**Arguments**

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

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

*fCalcETfromLE*


---

*fCalcETfromLE*


---

**Description**

Calculate ET from LE and Tair

**Usage**

*fCalcETfromLE*(LE = LE.V.n, Tair = Tair.V.n,  
LE.V.n, Tair.V.n)

**Arguments**

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

**Value**

Data vector of evapotranspiration (ET, mmol H2O 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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

fCalcExtRadiation      *fCalcExtRadiation*

---

**Description**

Calculate the extraterrestrial solar radiation with the eccentricity correction

**Usage**

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

**Arguments**

DoY

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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

fCalcPotRadiation      *fCalcPotRadiation*

---

### Description

Calculate the potential radiation

### Usage

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

### Arguments

DoY	Data vector with day of year (DoY), same length as Hour or length 1
Hour	Data vector with time as decimal hour of local time zone
LatDeg	Latitude in (decimal) degrees
LongDeg	Longitude in (decimal) degrees
TimeZone	Time zone (in hours)
useSolartime	
DoY.V.n	deprecated
Hour.V.n	deprecated
Lat_deg.n	deprecated
Long_deg.n	deprecated
TimeZone_h.n	deprecated
useSolartime.b	deprecated

### Value

Data vector of potential radiation (PotRad, W<sub>m-2</sub>)

### 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 <U+0160>igut <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 = 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 = AVP.V.n, Tair = Tair.V.n,
AVP.V.n, Tair.V.n)

```

**Arguments**

AVP	Data vector of actual vapour pressure (AVP, hPa (mbar))
Tair	Data vector of air temperature (Tair, degC)
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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

fCalcSVPfromTair      *fCalcSVPfromTair*

---

**Description**

Calculate SVP (of water) from Tair

**Usage**

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

**Arguments**

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

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

fCalcVPDfromRHAndTair      *fCalcVPDfromRHAndTair*

---

**Description**

Calculate VPD from rH and Tair

**Usage**

```
fCalcVPDfromRHAndTair(rH = RH.V.n, Tair = Tair.V.n,
  RH.V.n, Tair.V.n)
```

**Arguments**

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

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

fCheckHHTimeSeries     *fCheckHHTimeSeries*

---

**Description**

Check half-hourly time series data

**Usage**

```
fCheckHHTimeSeries(Time = Time.V.p, DTS = DTS.n,
  CallFunction = if (!missing(CallFunction.s)) CallFunction.s else "",
  Time.V.p, DTS.n, CallFunction.s)
```

**Arguments**

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

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

fConvertCtoK

*fConvertCtoK*


---

**Description**

Convert degree Celsius to degree Kelvin

**Usage**

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

**Arguments**

Celsius	Data vector in Celsius (degC)
Celsius.V.n	deprecated way of specifying Celsius

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

fConvertGlobalToVisible  
*fConvertGlobalToVisible*

---

**Description**

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

**Usage**

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

**Arguments**

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

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

fConvertKtoC                      *fConvertKtoC*

---

**Description**

Convert degree Kelvin to degree Celsius

**Usage**

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

**Arguments**

Kelvin	Data vector in Kelvin (degK)
Kelvin.V.n	deprecated, use Kelvin instead

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

fConvertTimeToPosix    *fConvertTimeToPosix*

---

**Description**

Convert different time formats to POSIX

**Usage**

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

**Arguments**

Data.F	Data frame with time columns to be converted
TFormat	Abbreviation for implemented time formats, see details
Year	Column name of year
Month	Column name of month
Day	Column name of day
Hour	Column name of hour
Min	Column name of min
TName	Column name of new column
TFormat.s	deprecated
Year.s	deprecated
Month.s	deprecated

Day.s	deprecated
Hour.s	deprecated
Min.s	deprecated
TName.s	deprecated
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:

**YDH** year, day of year, hour in decimal (e.g. 1998, 1, 10.5). The day (of year) format is (1-365 or 1-366 in leap years). The hour format is decimal time (0.0-23.5).

**YMDH** year, month, day of month, hour in decimal (e.g. 1998, 1, 1, 10.5) The month format is (1-12) The day (of month) format is (1-31).

**YMDHM** year, month, day of month, integer hour, minute (e.g. 1998, 1, 1, 10, 30) The hour format is (0-23) The minute format is (0-59)

### 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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

### See Also

[BerkeleyJulianDateToPOSIXct](#)

### 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 = Wm2.V.n,
                             Wm2.V.n)
```

**Arguments**

Wm2	Data vector in units of irradiance (W m <sup>-2</sup> )
Wm2.V.n	deprecated

**Value**

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

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

```
filterLongRuns      filterLongRuns
```

---

**Description**

replace runs, i.e sequences of numerically equal values, by NA

**Usage**

```
filterLongRuns(data, colNames, ...)
```

**Arguments**

data	data.frame with columns to filter
colNames	string vector of names indicating which columns to filter
...	further arguments to <a href="#">filterLongRunsInVector</a> such as minNRunLength.

**Details**

Longer runs, i.e. sequences of numerically identical values, in a series of measurements hint to problems during a noisy measurement, e.g. by sensor malfunction due to freezing. This function, replaces such values in such runs to indicate missing values.

**Value**

data.frame ans with long runs in specified columns replaced by NA

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

filterLongRunsInVector

*filterLongRunsInVector*

---

**Description**

replace runs of numerically equal values by NA

**Usage**

```
filterLongRunsInVector(x, minNRunLength = 8,
  replacement = NA, na.rm = TRUE)
```

**Arguments**

x	vector in which to replace long runs
minNRunLength	minimum length of a run to replace. Defaults to 4 hours in half-hourly spaced data.
replacement	value replacing the original values in long run
na.rm	set to FALSE if NA values interrupt runs

**Value**

vector x with long runs replaced by NA

**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 <U+0160>igut <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(RRef = R_ref.n, E0 = E_0.n,
             TSoil = Tsoil.n, TRef = if (missing(T_ref.n)) 273.15 +
             10 else T_ref.n, T0 = if (missing(T_0.n)) 227.13 else T_0.n,
             R_ref.n, E_0.n, Tsoil.n, T_ref.n, T_0.n)
```

**Arguments**

RRef	Respiration rate at reference temperature
E0	Temperature sensitivity ("activation energy") in Kelvin (degK)
TSoil	Soil temperature in Kelvin (degK)
TRef	
T0	
R_ref.n	deprecated way to specify RRef
E_0.n	deprecated way to specify E0
Tsoil.n	deprecated way to specify Tsoil
T_ref.n	deprecated way to specify TRef
T_0.n	deprecated way to specify T0

**Value**

Data vector of soil respiration rate (R,  $\mu\text{mol CO}_2 \text{ m}^{-2} \text{ s}^{-1}$ )

**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 <U+0160>igut <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 <code>&lt;year&gt;</code>
dirName	scalar string: the directory where the files reside
additionalColumnNames	character vector: column names to read in addition to <code>c("Month", "Day", "Hour", "NEE_st", "qf_NEE_st", "ustar", "Ta", 'Rg')</code>

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

fLoadFluxnet15

*Read a file in the format of Fluxnet 2015 release*


---

**Description**

Assigns default units to the columns and keeps variable name attributes as in original file.

**Usage**

```
fLoadFluxnet15(
  file_path,
  additional_columns = character(0),
  colname_NEE = "NEE",
  ...
)
```

**Arguments**

file_path	scalar string: the path to the csv file
additional_columns	character vector of columns to read in addition of standard columns of <a href="#">read_from_fluxnet15</a> . Can be a character vector or a object return by <a href="#">cols</a>
colname_NEE	name (scalar string) of column that reports NEE observations
...	further arguments to <a href="#">read_csv</a>

**Examples**

```
ds_fn15 <- Example_DETha98 %>%
  fConvertTimeToPosix('YDH', Year = 'Year', Day = 'DoY', Hour = 'Hour') %>%
  dplyr::mutate(
    TIMESTAMP_END = POSIXctToBerkeleyJulianDate(.data$DateTime),
    season = factor(199801)
  ) %>%
  dplyr::rename(SW_IN = .data$Rg, TA = .data$Tair, USTAR = .data$Ustar) %>%
  dplyr::select(dplyr::one_of(c(
    "TIMESTAMP_END", "NEE", "SW_IN", "TA", "VPD", "USTAR", "season")))
head(ds_fn15)
fname <- tempfile()
readr::write_csv(ds_fn15, fname)
```

```

# standard columns are renamed to REddyProc defaults
ds_eproc <- fLoadFluxnet15(fname)
head(ds_eproc)
EProc <- sEddyProc$new("DE-Tha", ds_eproc)
head(EProc$sExportData())

# Additional columns can be specified, e.g. factor column season
ds_eproc <- fLoadFluxnet15(fname,
  additional_columns = readr::cols(season = readr::col_factor()))
head(ds_eproc)
EProc <- sEddyProc$new("DE-Tha", ds_eproc,
  c("NEE", "Rg", "Tair", "VPD", "Ustar", "season"),
  ColNamesNonNumeric = "season"
)
head(EProc$sExportData())

```

---

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 = FileName.s,
  Dir = if (!missing(Dir.s)) Dir.s else "",
  FileName.s, Dir.s = "")

```

### Arguments

FileName	File name as a character string
Dir	Directory as a character string
FileName.s	deprecated
Dir.s	deprecated way of specifying Dir

### 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)**

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 <U+0160>igut <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 = FileName.s,
  Dir = if (!missing(Dir.s)) Dir.s else "",
  Digits = if (!missing(Digits.n)) Digits.n else 5,
  FileName.s, Dir.s, Digits.n)
```

**Arguments**

Data.F	Data frame
FileName	File base name as a string
Dir	Directory as a string
Digits	
FileName.s	deprecated
Dir.s	deprecated
Digits.n	deprecated

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

**Examples**

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

---

```
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 <U+0160>igut <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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

**See Also**

[renameVariablesInDataframe](#)

**Examples**

```
# adding mapping of foo, and overwriting mapping of DoY
getBGC05ToAmerifluxVariableNameMapping(c(foo = "F00", DoY = "doy"))
```

---

<code>getExamplePath</code>	<i>getExamplePath</i>
-----------------------------	-----------------------

---

**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**

<code>filename</code>	the name of the example file
<code>isTryDownload</code>	scalar logical whether to try downloading the file to package or tmp directory. Because of CRAN checks, need to explicitly set to TRUE
<code>exampleDir</code>	directory where examples are looked up and downloaded to
<code>remoteDir</code>	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 <U+0160>igut <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 <U+0160>igut <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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

## See Also

[getExamplePath](#)

---

getTZone	<i>getTZone</i>
----------	-----------------

---

**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 <U+0160>igut <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())
```

---

get_timestep_hours	<i>Get the timestep in fractional hours</i>
--------------------	---

---

**Description**

Get the timestep in fractional hours

**Usage**

```
get_timestep_hours(x)
```

**Arguments**

x	Vector of POSIX timestamps of at least length 2.
---	--

**Value**

Numeric scalar of the time difference of the first two entries in fraction hours.

---

<code>globalDummyVars</code>	<code>globalDummyVars</code>
------------------------------	------------------------------

---

**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)

---

<code>LightResponseCurveFitter</code>	<code>LightResponseCurveFitter</code>
---------------------------------------	---------------------------------------

---

**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, sdParameterP

[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 <a href="#">LightResponseCurveFitter_getParameterNames</a>
iOpt	position in theta that are optimized
flux	numeric: NEE (-NEE) or GPP time series [ $\mu\text{molCO}_2 / \text{m}^2 / \text{s}$ ], should not contain NA
sdFlux	numeric: standard deviation of Flux [ $\mu\text{molCO}_2 / \text{m}^2 / \text{s}$ ], should not contain NA
parameterPrior	numeric vector along theta: prior estimate of parameter (range of values)
sdParameterPrior	standard deviation of parameterPrior
...	other arguments to <a href="#">LightResponseCurveFitter_predictLRC</a> , 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 <U+0160>igut <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 [ $\mu\text{mol} / \text{m}^2 / \text{s}$ ] 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

## 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 <U+0160>igut <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 <a href="#">partGLControl</a> )
lastGoodParameters	numeric vector returned by last reasonable fit

### Details

Optimization is performed for three initial parameter sets that differ by  $\beta_0$  ( $* 1.3, * 0.8$ ). From those three, the optimization result is selected that yielded the lowest misfit. Starting values are:  $k = 0$ ,  $\beta = \text{interpercentileRange}(0.03, 0.97)$  of respiration,  $\alpha = 0.1$ ,  $R_{\text{ref}}$  from night-time estimate.  $E_0$  is fixed to the night-time estimate, but varies for estimating parameter uncertainty.

If `controlGLPart$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  $E_0$  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,  $E_0$  is drawn from  $N \sim (E_0, \text{sd}E_0)$ .

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 $E_0$
----------	---

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 <a href="#">optim</a> \, 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 <U+0160>igut <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 <U+0160>igut <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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

LightResponseCurveFitter\_getParameterNames  
*LightResponseCurveFitter\_getParameterNames*

---

**Description**

return the parameter names used by this Light Response Curve Function

**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 $\mu\text{mol CO}_2 \text{ m}^{-2} \text{ s}^{-2}$ )
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 <U+0160>igut <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 <U+0160>igut <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 <U+0160>igut <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 <U+0160>igut <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 <U+0160>igut <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, ..., dsDay, lastGoodParameters,
    ctrl)
```

### Arguments

theta0	initial parameter estimate
parameterPrior	prior estimate of model parameters
...	further parameters to .optimLRC,
dsDay	argument to .optimLRC, here checked for occurrence of high VPD
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  $R_{ref} < 0$  or  $\beta_0 < 0$  or  $\beta_0 > 250$

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 <a href="#">LightResponseCurveFitter_fitLRC</a> )

**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 <U+0160>igut <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 <a href="#">LightResponseCurveFitter_optimLRC</a> (passed to <a href="#">LightResponseCurveFitter_</a>

**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 downweights 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 <U+0160>igut <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 [ $\mu\text{mol} / \text{m}^2 / \text{s}$ ] or Global Radiation
...	further parameters to the LRC

**Details**

This method must 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 <U+0160>igut <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 <U+0160>igut <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 [ $\mu\text{mol} / \text{m}^2 / \text{s}$ ] 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 $R_g = 0$

**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 <U+0160>igut <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](#)

logitconv      logit-transformed convexity parameter. The value at original scale is obtained by  $\text{conv} = 1 / (1 + \exp(-\text{logitconv}))$

**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 <U+0160>igut <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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

**See Also**[LightResponseCurveFitter\\_predictGPP](#)

---

partGLControl	<i>partGLControl</i>
---------------	----------------------

---

### 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,
  isAssociateParmsToMeanOfValid = 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, useNighttimeBasalRespiration = FALSE)
```

### 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

**isAssociateParmsToMeanOfValid**  
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 = NonrectangularLRCFitter()`
- `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` data.frame of one row or `nRow = nWindow` corresponding to return value of `partGLFitNightTimeTRespSens` While column `RRef` is used only as a prior and initial value for the daytime-fitting and can be NA, `E0` is used as given temperature sensitivity and varied according to `sde0` in the bootstrap.
- `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  $\mu\text{mol CO}_2 / \text{m}^2 / \text{s}$ ). 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  $\text{minPropSaturation} * (\text{GPP at light-saturation, i.e. } \beta)$  this indicates that PAR is not sufficiently high to constrain the shape of the LRC

**useNighttimeBasalRespiration**

set to TRUE to estimate nighttime respiration based on basal respiration estimated on nighttime data instead of basal respiration estimated from daytime data. This implements the modified daytime method from Keenan 2019 (doi:10.1038/s41559-019-0809-2)

**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 <U+0160>igut <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, isAssociateParmsToMeanOfValid = 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

**isAssociateParmsToMeanOfValid**  
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 <U+0160>igut <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 = paste0("NEE",
  suffixDash, "_f"), QFNEEVar = if (!missing(QFNEEVar.s)) QFNEEVar.s else paste0("NEE",
  suffixDash, "_fqc"), QFNEEValue = if (!missing(QFNEEValue.n)) QFNEEValue.n else 0,
  NEESdVar = if (!missing(NEESdVar.s)) NEESdVar.s else paste0("NEE",
  suffixDash, "_fsd"), TempVar = paste0("Tair_f"),
  QFTempVar = if (!missing(QFTempVar.s)) QFTempVar.s else paste0("Tair_fqc"),
  QFTempValue = if (!missing(QFTempValue.n)) QFTempValue.n else 0,
  VPDVar = if (!missing(VPDVar.s)) VPDVar.s else paste0("VPD_f"),
  QFVPDVar = if (!missing(QFVPDVar.s)) QFVPDVar.s else paste0("VPD_fqc"),
  QFVPDValue = if (!missing(QFVPDValue.n)) QFVPDValue.n else 0,
  RadVar = if (!missing(RadVar.s)) RadVar.s else "Rg_f",
  QFRadVar = if (!missing(QFRadVar.s)) QFRadVar.s else paste0("Rg_fqc"),
  QFRadValue = if (!missing(QFRadValue.n)) QFRadValue.n else 0,
  PotRadVar = if (!missing(PotRadVar.s)) PotRadVar.s else "PotRad_NEW",
  suffix = if (!missing(Suffix.s)) Suffix.s else "",
  NEEVar.s, QFNEEVar.s, QFNEEValue.n, NEESdVar.s,
  TempVar.s, QFTempVar.s, QFTempValue.n,
  VPDVar.s, QFVPDVar.s, QFVPDValue.n, RadVar.s,
  QFRadVar.s, QFRadValue.n, PotRadVar.s,
  Suffix.s, controlGLPart = partGLControl())
```

**Arguments**

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

**Details**

The LRC fit usually weights NEE records by its uncertainty. In order to also use records with missing NEESdVar, 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\$sisFilterMeteoQualityFlag)
VPD	Water pressure deficit, quality filtered if isTRUE(controlGLPart\$sisFilterMeteoQualityFlag)
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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

partitionNEEGL

*partitionNEEGL*

---

**Description**

Partition NEE fluxes into GP and Reco using the daytime method.

**Usage**

```
partitionNEEGL(ds, NEEVar = if (!missing(NEEVar.s)) NEEVar.s else paste0("NEE",
  suffixDash, "_f"), TempVar = if (!missing(TempVar.s)) TempVar.s else "Tair_f",
  VPDVar = if (!missing(VPDVar.s)) VPDVar.s else "VPD_f",
  RadVar = if (!missing(RadVar.s)) RadVar.s else "Rg_f",
  suffix = if (!missing(Suffix.s)) Suffix.s else "",
  NEEVar.s, TempVar.s, VPDVar.s, RadVar.s,
  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	Variable of NEE
TempVar	Filled air or soil temperature variable (degC)
VPDVar	Filled Vapor Pressure Deficit - VPD - (hPa)
RadVar	Filled radiation variable
suffix	string inserted into column names before identifier for NEE column defaults (see <a href="#">sEddyProc_sMDSGapFillAfterUstar</a> ).
NEEVar.s	deprecated
TempVar.s	deprecated
VPDVar.s	deprecated
RadVar.s	deprecated
Suffix.s	deprecated identifier for NEE column defaults (see <a href="#">sEddyProc_sMDSGapFillAfterUstar</a> ).
...	further arguments to <a href="#">partGLExtractStandardData</a> , such as PotRadVar
controlGLPart	further default parameters, see <a href="#">partGLControl</a>
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 <code>RectangularLRCFitter()</code> , <code>NonrectangularLRCFitter()</code> , and <code>LogisticSigmoidLRCFitter()</code> .

**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 may use only measured data, i.e. with specified quality flag. Even with using filled VPD, there may be large gaps that have not been filled. 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: $\mu\text{mol CO}_2/\text{m}^2/\text{s}$
GPP_DT_<suffix>	predicted gross primary production $\mu\text{mol CO}_2/\text{m}^2/\text{s}$

<LRC>	Further light response curve (LRC) parameters and their standard deviation depend on the used LRC (e.g. for the non-rectangular LRC see <a href="#">NonrectangularLRCFitter_getParameter</a> ). 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 <a href="#">LightResponseCurveFitter_f</a>
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 <U+0160>igut <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, tz = getTZone(sDateTime))
```

**Arguments**

sDateTime      POSIXct vector  
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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

**See Also**

[BerkeleyJulianDateToPOSIXct](#),

---

read\_from\_fluxnet15      *extract REddyProc input columns from data.frame in Fluxnet15 format*

---

**Description**

Column format as described at <https://fluxnet.org/data/fluxnet2015-dataset/fullset-data-product/>

**Usage**

```
read_from_fluxnet15(ds, colname_NEE = "NEE")
```

**Arguments**

ds                    data.frame with columns TIMESTAMP\_END (Time YYYYMMDDHHMM),  
NEE, LE, H, USTAR, TA, TS, VPD, SW\_IN and optionally USTAR\_QC

colname\_NEE        name (scalar string) of column that reports NEE observations

**Details**

If input has numeric column USTAR\_QC then USTAR of records with USTAR\_QC > 2 are set to NA.

**Value**

data.frame with additional columns 'DateTime', 'NEE', 'Ustar' and 'Rg', 'Tair', 'Tsoil' if columns 'SW\_IN', 'TA', or 'TS' are present respectively

---

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, fixVPD):  
 ~~

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 [ $\mu\text{mol} / \text{m}^2 / \text{s}$ ] 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 $R_g = 0$

**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 <U+0160>igut <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. <a href="#">getAmerifluxToBGC05VariableNameMapping</a>

**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 <U+0160>igut <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\_deg.n, Long\_deg.n, TimeZone\_h.n)

`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, TempColName, RgColName, ...)

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

`sEddyProc_sPlotNEEVersusUstarForSeason`(season.s, Format.s, Dir.s, UstarColName, NEEColName, TempColName, RgColName, ...)

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, Verbose.b)

`sEddyProc_sMDSGapFillAfterUstarDistr`(..., UstarThres.df, UstarSuffix.V.s)

`sEddyProc_sMDSGapFillAfterUstar`(FluxVar.s, UstarVar.s, UstarThres.df, UstarSuffix.s, FlagEntryAfterLoss.b)

`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, QFTempValue.n, ...)

`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_sPlotHHFFluxes`(Var.s, QFVar.s, QFValue.n, Format.s, Dir.s)

`sEddyProc_sPlotHHFFluxesY`(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*

---

**Description**

Initializing sEddyProc class during sEddyProc\$new.

**Usage**

```
sEddyProc_initialize(ID = ID.s, Data = Data.F,
  ColNames = c("NEE", "Rg", "Tair", "VPD",
    "Ustar"), ColPOSIXTime = "DateTime",
  DTS = if (!missing(DTS.n)) DTS.n else 48,
  ColNamesNonNumeric = character(0), LatDeg = NA_real_,
  LongDeg = if (!missing(Long_deg.n)) Long_deg.n else NA_real_,
  TimeZoneHour = if (!missing(TimeZone_h.n)) TimeZone_h.n else NA_integer_,
  ID.s, Data.F, ColNames.V.s, ColPOSIXTime.s,
  DTS.n, ColNamesNonNumeric.V.s, Lat_deg.n,
  Long_deg.n, TimeZone_h.n, ...)
```

**Arguments**

ID	String with site ID
Data	Data frame with at least three month of (half-)hourly site-level eddy data
ColNames	Vector with selected column names, the fewer columns the faster the processing. The default specifies column names assumed in further processing.
ColPOSIXTime	Column name with POSIX time stamp
DTS	Daily time steps
ColNamesNonNumeric	Names of columns that should not be checked for numeric type, e.g. season column
LatDeg	Latitude in (decimal) degrees (-90 to + 90)
LongDeg	Longitude in (decimal) degrees (-180 to + 180)
TimeZoneHour	Time zone: hours shift to UTC, e.g. 1 for Berlin
ID.s	deprecated
Data.F	deprecated
ColNames.V.s	deprecated

```

ColPOSIXTime.s deprecated
DTS.n           deprecated
ColNamesNonNumeric.V.s
                deprecated
Lat_deg.n      deprecated
Long_deg.n     deprecated
TimeZone_h.n   deprecated
...           ('...' 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.

There are several fields initialized within the class.

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

sUSTAR\_SCEN a data.frame with first column the season, and other columns different uStar threshold estimates, as returned by [usGetAnnualSeasonUStarMap](#)

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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sApplyUStarScen

*sEddyProc sApplyUStarScen*

---

**Description**

apply a function with changing the suffix argument

**Usage**

```
sEddyProc_sApplyUStarScen(FUN, ..., uStarScenKeep = character(0),
  warnOnOtherErrors = FALSE, uStarSuffixes = .self$sGetUstarSuffixes())
```

**Arguments**

FUN	function to be applied
...	further arguments to FUN
uStarScenKeep	Scalar string specifying the scenario for which to keep parameters. If not specified defaults to the first entry in uStarSuffixes.
warnOnOtherErrors	Set to only display a warning on errors in uStarScenarios other than uStarScenKeep instead of stopping.
uStarSuffixes	

**Details**

When repeating computations, some of the output variables maybe replaced. Argument uStarKeep allows to select the scenario which is computed last, and hence to which output columns refer to.

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

```
sEddyProc_sCalcPotRadiation
      sEddyProc sCalcPotRadiation
```

---

**Description**

compute potential radiation from position and time

**Usage**

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

**Arguments**

```
useSolartime
useSolartime.b by default corrects hour (given in 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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

```
sEddyProc_sEstimateUstarScenarios
      sEddyProc sEstimateUstarScenarios
```

---

**Description**

Estimate the distribution of  $u^*$  threshold by bootstrapping over data

**Usage**

```
sEddyProc_sEstimateUstarScenarios(ctrlUstarEst = usControlUstarEst(),
                                   ctrlUstarSub = usControlUstarSubsetting(),
                                   UstarColName = "Ustar", NEEColName = "NEE",
                                   TempColName = "Tair", RgColName = "Rg",
                                   ..., seasonFactor = usCreateSeasonFactorMonth(sDATA$sDateTime),
                                   nSample = 200L, probs = c(0.05, 0.5,
                                                           0.95), isVerbose = TRUE, suppressWarningsAfterFirst = TRUE)
```

**Arguments**

ctrlUstarEst	control parameters for estimating uStar on a single binned series, see <a href="#">usControlUstarEst</a>
ctrlUstarSub	control parameters for subsetting time series (number of temperature and Ustar classes ...), see <a href="#">usControlUstarSubsetting</a>
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 <a href="#">sEddyProc_sEstUstarThreshold</a>
seasonFactor	
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
isVerbose	set to FALSE to omit printing progress
suppressWarningsAfterFirst	set to FALSE to show also warnings for all bootstrap estimates instead of only the first bootstrap sample

**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$minValidBootProp` (default 40%) did not report a threshold, no quantiles (i.e. NA) are reported.

**Value**

updated class. Request results by [sEddyProc\\_sGetEstimatedUstarThresholdDistribution](#)

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

**See Also**

[sEddyProc\\_sEstUstarThold](#), [sEddyProc\\_sGetEstimatedUstarThresholdDistribution](#), [sEddyProc\\_sSetUstarScena](#), [sEddyProc\\_sMDSGapFillUstarScens](#)

---

sEddyProc\_sEstUstarThold

*sEddyProc\$sEstUstarThreshold - Estimating ustar threshold*


---

## Description

Calling [usEstUstarThreshold](#) for class data and storing results

## Usage

```
sEddyProc_sEstUstarThold(UstarColName = "Ustar",
  NEEColName = "NEE", TempColName = "Tair",
  RgColName = "Rg", ..., seasonFactor = usCreateSeasonFactorMonth(sDATA$sDateTime))
```

## 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 <a href="#">usEstUstarThreshold</a>
seasonFactor	

## Value

result component uStarTh of [usEstUstarThreshold](#). In addition the result is stored in class variable sUSTAR\_DETAILS.

## 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 <U+0160>igut <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", ..., isWarnDeprecated = TRUE)
```

## 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 <a href="#">usEstUstarThreshold</a>
isWarnDeprecated	set to FALSE to avoid deprecated warning.

## Value

result of [usEstUstarThreshold](#). In addition the result is stored in class variable sUSTAR\_DETAILS and the bins as additional columns to sTemp

## 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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sEstUstarThresholdDistribution  
*sEddyProc sEstUstarThresholdDistribution*

---

### Description

Estimate the distribution of  $u^*$  threshold by bootstrapping over data

### Usage

```
sEddyProc_sEstUstarThresholdDistribution(...)
```

### Arguments

... further parameters to [sEddyProc\\_sEstimateUstarScenarios](#)

### Details

This method returns the results directly, without modifying the class. It is there for portability reasons. Recommended is using method [sEddyProc\\_sEstimateUstarScenarios](#) to update the class and then getting the results from the class by [sEddyProc\\_sGetEstimatedUstarThresholdDistribution](#).

### Value

result of [sEddyProc\\_sGetEstimatedUstarThresholdDistribution](#)

### 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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sExportData *sEddyProc sExportData*

---

### 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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sExportResults

*sEddyProc sExportResults*

---

**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 <U+0160>igut <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 <code>_good_</code> (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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sFillLUT      *sEddyProc sFillLUT*

---

## 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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sFillMDC      *sEddyProc sFillMDC*

---

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

`sEddyProc_sFillVPDFromDew`*Estimate VPD from daily minimum temperature*

---

**Description**

of the data in the class function using [estimate\\_vpd\\_from\\_dew](#).

**Usage**

```
sEddyProc_sFillVPDFromDew(...)
```

**Arguments**

... further arguments to [estimate\\_vpd\\_from\\_dew](#)

**Value**

side effect of updated column VPDfromDew in class

---

`sEddyProc_sGetData`*sEddyProc sGetData*

---

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sGetEstimatedUstarThresholdDistribution  
*sEddyProc sGetEstimatedUstarThresholdDistribution*

---

**Description**

return the results of [sEddyProc\\_sEstimateUstarScenarios](#)

**Usage**

sEddyProc\_sGetEstimatedUstarThresholdDistribution()

**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)**

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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

**See Also**

[sEddyProc\\_sSetUstarScenarios](#)

---

sEddyProc\_sGetUstarScenarios  
*sEddyProc sGetUstarScenarios*

---

**Description**

get the current uStar processing scenarios

**Usage**

sEddyProc\_sGetUstarScenarios()

**Details**

the associated suffixes can be retrieved by `colnames(myClass$sGetUstarScenarios())[-1]`

**Value**

a data.frame with first column listing each season and other column a scenario of uStar thresholds.

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

**See Also**

[sEddyProc\\_sSetUstarScenarios](#)

---

sEddyProc\_sGetUstarSuffixes

*sEddyProc sGetUstarSuffixes*

---

**Description**

get the current uStar suffixes

**Usage**

```
sEddyProc_sGetUstarSuffixes()
```

**Value**

a character vector of suffixes. If no uStar thresholds have been estimated, returns character(0)

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

**See Also**

[sEddyProc\\_sGetUstarScenarios](#)

---

sEddyProc\_sGLFluxPartition  
*sEddyProc sGLFluxPartition*

---

### Description

Daytime-based Flux partitioning after Lasslop et al. (2010)

### Usage

```
sEddyProc_sGLFluxPartition(..., debug = list(useLocaltime = FALSE),
  debug.1, isWarnReplaceColumns = TRUE)
```

### Arguments

...	arguments to <code>partitionNEEGL</code> in addition to the dataset such as <code>suffix</code>
debug	List with debugging control.
	<b>useLocaltime</b> if TRUE use local time zone instead of geo-solar time to compute potential radiation
debug.1	deprecated, renamed to debug
isWarnReplaceColumns	set to FALSE to avoid the warning on replacing output columns

### Details

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

### 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 <U+0160>igut <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_sGLFluxPartitionUStarScens
      sEddyProc sGLFluxPartitionUStarScens
```

---

## Description

Flux partitioning after Lasslop et al. (2010)

## Usage

```
sEddyProc_sGLFluxPartitionUStarScens(...,
  uStarScenKeep = suffixes[1], isWarnReplaceColumns = FALSE,
  warnOnOtherErrors = FALSE, controlGLPart = partGLControl())
```

## Arguments

... arguments to [sEddyProc\\_sGLFluxPartition](#)

uStarScenKeep Scalar string specifying the scenario for which to keep parameters (see [sEddyProc\\_sApplyUStarScen](#)). Defaults to the first scenario, which is usually the uStar without bootstrap: "uStar".

isWarnReplaceColumns overriding default to avoid the warning on replacing output columns, because this is intended when processing several uStar scenarios.

warnOnOtherErrors Set to TRUE to only display a warning on errors in uStarScenarios other than uStarScenKeep instead of stopping.

controlGLPart further default parameters

## Details

Daytime-based partitioning of measured net ecosystem fluxes into gross primary production (GPP) and ecosystem respiration (Reco) for all u\* threshold scenarios.

For the uStarScenKeep, a full set of output columns is returned. For the other scenarios, the bootstrap of GPP uncertainty is omitted and columns "FP\_<x>" are overridden.

## 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 <mmigliavacca@bgc-jena.mpg.de> [aut], Kerstin Sickel <ksickel@bgc-jena.mpg.de> [ctb, trl], Ladislav <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sMDSGapFill *sEddyProc sMDSGapFill*

---

## Description

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

## Usage

```
sEddyProc_sMDSGapFill(Var = Var.s, QFVar = if (!missing(QFVar.s)) QFVar.s else "none",
  QFValue = if (!missing(QFValue.n)) QFValue.n else NA_real_,
  V1 = if (!missing(V1.s)) V1.s else "Rg",
  T1 = if (!missing(T1.n)) T1.n else 50,
  V2 = if (!missing(V2.s)) V2.s else "VPD",
  T2 = if (!missing(T2.n)) T2.n else 5,
  V3 = if (!missing(V3.s)) V3.s else "Tair",
  T3 = if (!missing(T3.n)) T3.n else 2.5,
  FillAll = if (!missing(FillAll.b)) FillAll.b else TRUE,
  isVerbose = if (!missing(Verbose.b)) Verbose.b else TRUE,
  suffix = if (!missing(Suffix.s)) Suffix.s else "",
  minNWarnRunLength = if (Var == "NEE") 4 *
    .self$sINFO$DTS/24 else NA_integer_,
  Var.s, QFVar.s, QFValue.n, V1.s, T1.n,
  V2.s, T2.n, V3.s, T3.n, FillAll.b, Verbose.b,
  Suffix.s)
```

## Arguments

Var	Variable to be filled
QFVar	
QFValue	
V1	Condition variable 1 (default: Global radiation 'Rg' in W m-2)
T1	Tolerance interval 1 (default: 50 W m-2)
V2	Condition variable 2 (default: Vapour pressure deficit 'VPD' in hPa)
T2	Tolerance interval 2 (default: 5 hPa)
V3	Condition variable 3 (default: Air temperature 'Tair' in degC)
T3	Tolerance interval 3 (default: 2.5 degC)
FillAll	Fill all values to estimate uncertainties
isVerbose	Print status information to screen
suffix	String suffix needed for different processing setups on the same dataset (for explanations see below)

minNWarnRunLength	scalar integer: warn if number of subsequent numerically equal values exceeds this number. Set to Inf or NA for no warnings. defaults for "NEE" to records across 4 hours and no warning for others.
Var.s	deprecated
QFVar.s	deprecated
QFValue.n	deprecated
V1.s	deprecated
T1.n	deprecated
V2.s	deprecated
T2.n	deprecated
V3.s	deprecated
T3.n	deprecated
FillAll.b	deprecated
Verbose.b	deprecated
Suffix.s	deprecated

### Details

Initialize temporal data frame sTEMP for newly generated gap filled data and qualifiers, see [sEddyProc\\_sFillInit](#) for explanations on suffixes.

Runs of numerically equal numbers hint to problems of the data and cause unreasonable estimates of uncertainty. This routine warns the user.

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 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 <U+0160>igut <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*

---

## Description

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

## Usage

```
sEddyProc_sMDSGapFillAfterUstar(fluxVar,
  uStarVar = "Ustar", uStarTh = .self$sGetUstarScenarios()[,
    c("season", uStarSuffix), drop = FALSE],
  uStarSuffix = "uStar", isFlagEntryAfterLowTurbulence = FALSE,
  isFilterDayTime = FALSE, swThr = 10,
  RgColName = "Rg", ...)
```

## Arguments

fluxVar	Flux variable to gap fill after ustar filtering
uStarVar	Column name of friction velocity u * (ms-1), default 'Ustar'
uStarTh	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. If only one value is given, it is used for all records.
uStarSuffix	Different suffixes required are for different u * scenarios
isFlagEntryAfterLowTurbulence	Set to TRUE for flagging the first entry after low turbulence 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 <a href="#">sEddyProc_sMDSGapFill</a>

## Details

Calling [sEddyProc\\_sMDSGapFill](#) after filtering for (provided) friction velocity u\*

The u\* threshold(s) are provided with argument uStarTh 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 (Papale 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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

**See Also**

- [sEddyProc\\_sEstimateUstarScenarios](#) and `link{sEddyProc_sEstUstarThold}` for estimating the  $u^*$  threshold from the data.
- [sEddyProc\\_sMDSGapFillUStarScens](#) for automated gapfilling for several scenarios of  $u^*$  threshold estimates.

---

sEddyProc\_sMDSGapFillAfterUStarDistr

*sEddyProc sMDSGapFillAfterUStarDistr*

---

**Description**

gapfilling for several filters of estimated friction velocity Ustar thresholds.

**Usage**

```
sEddyProc_sMDSGapFillAfterUStarDistr(...,
  uStarTh, uStarSuffixes = colnames(uStarTh)[-1])
```

**Arguments**

...	other arguments to <a href="#">sEddyProc_sMDSGapFillAfterUstar</a> and <a href="#">sEddyProc_sMDSGapFill</a> such as <code>fluxVar</code>
uStarTh	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 <code>seasonFactor</code>
uStarSuffixes	String vector to distinguish result columns for different ustar values. Its length must correspond to column numbers in <code>UstarThres.m.n</code> .

**Details**

This method is superseded by [sEddyProc\\_sMDSGapFillUStarScens](#) and only there for backward portability.

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sMDSGapFillUStarScens

*sEddyProc sMDSGapFillUStarScens*

---

**Description**

gapfilling for several filters of estimated friction velocity Ustar thresholds.

**Usage**

sEddyProc\_sMDSGapFillUStarScens(...)

**Arguments**

... other arguments to [sEddyProc\\_sMDSGapFillAfterUstar](#) and [sEddyProc\\_sMDSGapFill](#) such as fluxVar

**Details**

sEddyProc\$sMDSGapFillUStarDistr: calling [sEddyProc\\_sMDSGapFillAfterUstar](#) for several filters of friction velocity Ustar.

The scenarios need to be set before by [sEddyProc\\_sSetUstarScenarios](#) or accepting the defaults annual estimates of `link{sEddyProc_sEstimateUstarScenarios}`.

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 outputs is calculated for several  $u^*$  r-estimates and distinguished by a suffix after the variable. E.g. with an an entry "U05" in uStarSuffixes in [sEddyProc\\_sSetUstarScenarios](#) 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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

**See Also**

[useCase vignette](#)

---

sEddyProc\_sMRFluxPartition

*sEddyProc sMRFluxPartition*

---

**Description**

Nighttime-based partitioning of net ecosystem fluxes into gross fluxes GPP and REco

**Usage**

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

**Arguments**

FluxVar	Variable name of column with original and filled net ecosystem fluxes (NEE)
QFFluxVar	Quality flag of NEE variable
QFFluxValue	Value of quality flag for <code>_good_</code> (original) data
TempVar	Filled air- or soil temperature variable (degC)
QFTempVar	Quality flag of filled temperature variable
QFTempValue	Value of temperature quality flag for <code>_good_</code> (original) data

RadVar	Unfilled (original) radiation variable
TRef	Reference temperature in Kelvin (degK) used in fLloydTaylor for regressing Flux and Temperature
suffix	String suffix needed for different processing setups on the same dataset (for explanations see below)
FluxVar.s	deprecated
QFFluxVar.s	deprecated
QFFluxValue.n	deprecated
TempVar.s	deprecated
QFTempVar.s	deprecated
QFTempValue.n	deprecated
RadVar.s	deprecated
T_ref.n	deprecated
Suffix.s	deprecated
debug.l	deprecated
debug	List with debugging control (passed also to sEddyProc_sRegrE0fromShortTerm for providing fixedE0 = myE0).
	<b>useLocaltime</b> see details on solar vs local time
parse0Regression	list with further parameters passed down to sEddyProc_sRegrE0fromShortTerm and fRegrE0fromShortTerm, such as TempRange

## 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 is provided and if the defaults for FluxVar and QFFluxVar are used, the suffix will be added to their variable names (e.g. 'NEE\_f' will be renamed to 'NEE\_uStar\_f' and 'NEE\_fqc' to 'NEE\_uStar\_fqc' for the suffix = 'uStar'). Currently, this works only with defaults of FluxVar = 'NEE\_f' and QFFluxVar = '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)

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 <U+0160>igut <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\_sMRFluxPartitionUStarScens

*sEddyProc sMRFluxPartitionUStarScens*

---

### Description

Flux partitioning after Reichstein et al. (2005)

**Usage**

```
sEddyProc_sMRFluxPartitionUStarScens(...,
  uStarScenKeep = character(0))
```

**Arguments**

... arguments to [sEddyProc\\_sMRFluxPartition](#)

uStarScenKeep Scalar string specifying the scenario for which to keep parameters (see [sEddyProc\\_sApplyUStarScen](#)). Defaults to the first scenario.

**Details**

Nighttime-based partitioning of measured net ecosystem fluxes into gross primary production (GPP) and ecosystem respiration (Reco) for all u\* threshold scenarios.

**Value**

NULL, it adds output columns in the class

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

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

**Arguments**

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

**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 <U+0160>igut <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`. Furthermore, a change of the mass unit is provided by argument `massFactor`. The default parameters assume original units of  $\mu\text{mol CO}_2 / \text{m}^2 / \text{second}$  and convert to  $\text{gC} / \text{m}^2 / \text{day}$ . The conversion factors allow plotting variables with different units

**Usage**

```
sEddyProc_sPlotDailySumsY(Var = Var.s, VarUnc = "none",
  Year = Year.i, timeFactor = if (!missing(timeFactor.n)) timeFactor.n else 3600 *
    24, massFactor = if (!missing(massFactor.n)) massFactor.n else (44.0096/1e+06) *
    (12.011/44.0096), unit = if (!missing(unit.s)) unit.s else "gC/m2/day",
  data = cbind(sDATA, sTEMP), dts = sINFO$DTS,
  Var.s, VarUnc.s, Year.i, timeFactor.n,
  massFactor.n, unit.s)
```

**Arguments**

Var	(Filled) variable to plot
VarUnc	Uncertainty estimates for variable
Year	Year to plot
timeFactor	time conversion factor with default per second to per day
massFactor	mass conversion factor with default from mumol CO2 to g C
unit	unit of the daily sums
data	data.frame with variables to plot
dts	numeric integer
Var.s	
VarUnc.s	
Year.i	
timeFactor.n	
massFactor.n	
unit.s	

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sPlotDiurnalCycle  
*sEddyProc sPlotDiurnalCycle*

---

**Description**

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

**Usage**

```
sEddyProc_sPlotDiurnalCycle(Var = Var.s,
  QFVar = if (!missing(QFVar.s)) QFVar.s else "none",
  QFValue = if (!missing(QFValue.n)) QFValue.n else NA_real_,
  Format = if (!missing(Format.s)) Format.s else "pdf",
  Dir = if (!missing(Dir.s)) Dir.s else "plots",
  data = cbind(sDATA, sTEMP), dts = sINFO$DTS,
  Var.s, QFVar.s, QFValue.n, Format.s,
  Dir.s)
```

**Arguments**

Var	Variable to plot
QFVar	Quality flag of variable to be filled
QFValue	Value of quality flag for data to plot
Format	Graphics file format (e.g. 'pdf', 'png')
Dir	Directory for plotting
data	data.frame with variables to plot
dts	numeric integer
Var.s	deprecated
QFVar.s	deprecated
QFValue.n	deprecated
Format.s	deprecated
Dir.s	deprecated

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sPlotFingerprint  
*sEddyProc sPlotFingerprint*

---

**Description**

Generates fingerprint in file

**Usage**

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

**Arguments**

Var	Variable to plot
QFVar	Quality flag of variable to be filled
QFValue	Value of quality flag for data to plot
Format	
Dir	Directory for plotting
...	further arguments to <a href="#">sEddyProc_sPlotFingerprintY</a>
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

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sPlotFingerprintY  
*sEddyProc sPlotFingerprintY*

---

**Description**

Plot fingerprint for a single year scaled to all data.

**Usage**

```
sEddyProc_sPlotFingerprintY(Var = Var.s,
  QFVar = "none", QFValue = if (!missing(QFValue.n)) QFValue.n else NA_real_,
  Year = Year.i, onlyLegend = if (!missing(Legend.b)) Legend.b else F,
  colors = if (!missing(Col.V)) Col.V else colorRampPalette(c("#00007F",
    "blue", "#007FFF", "cyan", "#7FFF7F",
    "yellow", "#FF7F00", "red", "#7F0000"))(50),
  valueLimits = range(Plot.V.n, na.rm = TRUE),
  data = cbind(sDATA, sTEMP), dts = sINFO$dts,
  Var.s, QFVar.s, QFValue.n, Year.i, Legend.b,
  Col.V)
```

**Arguments**

Var	Variable to plot
QFVar	
QFValue	
Year	Year to plot
onlyLegend	Plot only legend
colors	Color palette for fingerprint plot (can be also defined by user), i.e. color scale argument to <a href="#">image</a>
valueLimits	values outside this range will be set to the range borders to avoid distorting colour scale e.g. valueLimits = quantile(EddyProc.C\$DATA\$NEE, prob = c(0.05, 0.95), na.rm = TRUE)
data	data.frame with variables to plot
dts	numeric integer of hours in day
Var.s	deprecated
QFVar.s	deprecated
QFValue.n	deprecated
Year.i	deprecated
Legend.b	deprecated
Col.V	deprecated

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sPlotHHFluxes

*sEddyProc sPlotHHFluxes*

---

**Description**

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

**Usage**

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

**Arguments**

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

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sPlotHHFluxesY

*sEddyProc sPlotHHFluxesY*

---

**Description**

Plot half-hourly fluxes for a single year scaled to all data.

**Usage**

```
sEddyProc_sPlotHHFluxesY(Var = Var.s, QFVar = if (!missing(QFVar.s)) QFVar.s else "none",
  QFValue = if (!missing(QFValue.n)) QFValue.n else NA_real_,
  Year = Year.i, data = cbind(sDATA, sTEMP),
  dts = sINFO$DTS, Var.s, QFVar.s, QFValue.n,
  Year.i)
```

**Arguments**

Var	Variable to plot
QFVar	
QFValue	
Year	Year to plot
data	data.frame with variables to plot
dts	numeric integer
Var.s	deprecated
QFVar.s	deprecated
QFValue.n	deprecated
Year.i	deprecated

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sPlotNEEVersusUStarForSeason

*sEddyProc sPlotNEEVersusUStarForSeason*

---

**Description**

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

**Usage**

```
sEddyProc_sPlotNEEVersusUStarForSeason(season = levels(data$season)[1],
  format = "pdf", dir = "plots", UstarColName = "Ustar",
  NEEColName = "NEE", TempColName = "Tair",
  WInch = 16 * 0.394, HInchSingle = 6 *
  0.394, ..., data = cbind(sDATA, sTEMP,
  sUSTAR_DETAILS$bins[, c("uStarBin",
  "tempBin")]))
```

**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
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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sSetLocationInfo

*sEddyProc sSetLocationInfo*

---

**Description**

set Location and time Zone information to sLOCATION

**Usage**

```
sEddyProc_sSetLocationInfo(LatDeg = if (!missing(Lat_deg.n)) Lat_deg.n else NA_real_,
  LongDeg = if (!missing(Long_deg.n)) Long_deg.n else NA_real_,
  TimeZoneHour = if (!missing(TimeZone_h.n)) TimeZone_h.n else NA_integer_,
  Lat_deg.n, Long_deg.n, TimeZone_h.n)
```

**Arguments**

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

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sSetUstarScenarios

*sEddyProc sSetUstarScenarios*

---

**Description**

set uStar processing scenarios

**Usage**

```
sEddyProc_sSetUstarScenarios(uStarTh, uStarSuffixes = colnames(uStarTh)[-1])
```

**Arguments**

uStarTh	data.frame as returned by <a href="#">usGetAnnualSeasonUStarMap</a> or <a href="#">usGetSeasonalSeasonUStarMap</a> : First column, season names, and remaining columns different estimates of uStar Threshold. If uStarTh 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 of <a href="#">sEddyProc_sEstUstarThresholdDistribution</a>
uStarSuffixes	the suffixes appended to result column names by default the column names of uStarTh unless its first season column

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

**See Also**

[sEddyProc\\_sGetUstarScenarios](#)

---

sEddyProc\_sSetUStarSeasons  
*sEddyProc sSetUStarSeasons*

---

**Description**

Defining seasons for the uStar threshold estimation

**Usage**

```
sEddyProc_sSetUStarSeasons(seasonFactor = usCreateSeasonFactorMonth(sDATA$sDateTime))
```

**Arguments**

seasonFactor    factor for subsetting times with different uStar threshold (see details)

**Value**

class with updated seasonFactor

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sTKFluxPartition  
*sEddyProc sTKFluxPartition*

---

**Description**

Modified daytime-based Flux partitioning after Keenan et al. (2019)

**Usage**

```
sEddyProc_sTKFluxPartition(..., controlGLPart = partGLControl())
```

**Arguments**

...                    arguments to [sEddyProc\\_sGLFluxPartition](#) in addition to the dataset  
controlGLPart    further default parameters, such as suffix

**Value**

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

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

sEddyProc\_sTKFluxPartitionUStarScens

*sEddyProc sTKFluxPartitionUStarScens*

---

**Description**

Flux partitioning after Lasslop 2015

**Usage**

```
sEddyProc_sTKFluxPartitionUStarScens(...,
  uStarScenKeep = character(0))
```

**Arguments**

...	arguments to <a href="#">sEddyProc_sTKFluxPartition</a>
uStarScenKeep	Scalar string specifying the scenario for which to keep parameters (see <a href="#">sEddyProc_sApplyUStarScen</a> ). Defaults to the first scenario.

**Details**

Daytime-based partitioning of measured net ecosystem fluxes into gross primary production (GPP) and ecosystem respiration (Reco) for all  $u^*$  threshold scenarios.

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

sEddyProc\_update\_ustarthreshold\_columns

*Add columns reporting the uStar threshold for each scenario to sDATA*

---

**Description**

Add columns reporting the uStar threshold for each scenario to sDATA

**Usage**

```
sEddyProc_update_ustarthreshold_columns()
```

**Value**

side effect in .self\$sDATA new columns Ustar\_Thresh\_<ustarsuffix>

**See Also**

[sEddyProc\\_sGetUstarScenarios](#)

---

sEddyProc\_useAnnualUStarThresholds

*sEddyProc useAnnualUStarThresholds*

---

**Description**

use seasonal estimates of uStar thresholds

**Usage**

```
sEddyProc_useAnnualUStarThresholds()
```

**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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

**See Also**

[sEddyProc\\_sSetUstarScenarios](#), [sEddyProc\\_useSeasonalUStarThresholds](#)

---

sEddyProc\_useSeasonalUStarThresholds

*sEddyProc useSeasonalUStarThresholds*


---

### Description

use seasonal estimates of uStar thresholds

### Usage

```
sEddyProc_useSeasonalUStarThresholds()
```

### 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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

### See Also

[sEddyProc\\_sSetUstarScenarios](#), [sEddyProc\\_useAnnualUStarThresholds](#)

---

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 change point detection without binning uStar but with additionally changed aggregation scheme for several temperature classifications
isUsingCPT	set to TRUE to use change point detection 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 <U+0160>igut <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 <U+0160>igut <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 <U+0160>igut <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(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 <U+0160>igut <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 <U+0160>igut <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 <U+0160>igut <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 = usCreateSeasonFactorMonth(ds$sDateTime),
  yearOfSeasonFactor = usGetYearOfSeason(seasonFactor,
    ds$sDateTime), ctrlUstarEst = usControlUstarEst(),
  ctrlUstarSub = usControlUstarSubsetting(),
  fEstimateUstarBinned = usEstUstarThresholdSingleFw2Binned,
  isCleaned = FALSE, isInBootstrap = FALSE)
```

### Arguments

ds	data.frame with columns "sDateTime", "Ustar", "NEE", "Tair", and "Rg"
seasonFactor	factor for subsetting times (see details)
yearOfSeasonFactor	named integer vector: for each seasonFactor level, get the year (aggregation period) that this season belongs to
ctrlUstarEst	control parameters for estimating uStar on a single binned series, see <a href="#">usControlUstarEst</a>
ctrlUstarSub	control parameters for subsetting time series (number of temperature and Ustar classes ...), see <a href="#">usControlUstarSubsetting</a>
fEstimateUstarBinned	function to estimate UStar on a single binned series, see <a href="#">usEstUstarThresholdSingleFw2Binned</a>
isCleaned	set to TRUE, if the data was cleaned already, to avoid expensive call to usGetValidUstarIndices.
isInBootstrap	set to TRUE if this is called from <a href="#">sEddyProc_sEstimateUstarScenarios</a> to avoid further bootstraps in change-point detection

### 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) 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. 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\\_sEstUstarThold](#) 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 = usControlUstarEst(isUsingCPTSeveralT = TRUE)` 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$minValidUstarTempClassesProp` (default 20%) of the temperature classes.

Note, that this method often gives higher estimates of the  $u * \text{threshold}$ .

**One-big-season fallback** If there are too few records within one year, or when no season yielded a finite  $u * \text{Threshold}$  estimate, then the yearly  $u * \text{Th}$  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$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	<code>data.frame</code> 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 <code>uStarMaxSeason</code> was nonfinite), "nRec" number of valid records (only if the pooled estimate was calculated), "uStarAggr" chosen estimate, corresponding to <code>uStarPooled</code> if this was calculated, or <code>uStarMaxSeason</code> or <code>uStarTh</code> across years if the former was non-finite
season	<code>data.frame</code> 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 <code>uStarSeasonEst</code> , or the yearly <code>seasonYear\$uStarAggr</code> , 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 <U+0160>igut <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 = usControlUstarEst())
```

**Arguments**

Ust_bins.f	data.frame with columns NEE_avg and Ust_avg, of Ustar bins
ctrlUstarEst	parameter list, see <a href="#">usControlUstarEst</a> 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 <U+0160>igut <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 = usControlUstarEst())
```

**Arguments**

Ust\_bins.f      data.frame with column s NEE\_avg and Ust\_avg, of Ustar bins  
ctrlUstarEst    parameter list, see [usControlUstarEst](#) for defaults and description

**Details**

Demand that threshold is higher than ctrlUstarEst\$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 <U+0160>igut <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](#)

**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](mailto:REddyProc-help@bgc-jena.mpg.de)> [cph], Thomas Wutzler <[twutz@bgc-jena.mpg.de](mailto:twutz@bgc-jena.mpg.de)> [aut, cre], Markus Reichstein <[mreichstein@bgc-jena.mpg.de](mailto:mreichstein@bgc-jena.mpg.de)> [aut], Antje Maria Moffat <[antje.moffat@bgc.mpg.de](mailto:antje.moffat@bgc.mpg.de)> [aut, trl], Olaf Menzer <[omenzer@bgc-jena.mpg.de](mailto:omenzer@bgc-jena.mpg.de)> [ctb], Mirco Migliavacca <[mmiglia@bgc-jena.mpg.de](mailto:mmiglia@bgc-jena.mpg.de)> [aut], Kerstin Sickel <[ksickel@bgc-jena.mpg.de](mailto:ksickel@bgc-jena.mpg.de)> [ctb, trl], Ladislav <U+0160>igut <[sigut.l@czechglobe.cz](mailto: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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

---

usGetYearOfSeason	<i>usGetYearOfSeason</i>
-------------------	--------------------------

---

**Description**

determine the year of the record of middle of seasons

**Usage**

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

**Arguments**

seasonFactor	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 <U+0160>igut <sigut.l@czechglobe.cz> [ctb]

# Index

- \* **classes**
  - LightResponseCurveFitter-class, [35](#)
  - LogisticSigmoidLRCFitter-class, [49](#)
  - NonrectangularLRCFitter-class, [51](#)
  - RectangularLRCFitter-class, [65](#)
  - RectangularLRCFitterCVersion-class, [66](#)
  - sEddyProc-class, [69](#)
- \* **dataset**
  - DEGebExample, [8](#)
  - Example\_DETha98, [9](#)
- \* **package**
  - REddyProc-package, [5](#)
- \_REddyProc\_RHLightResponseCostC (RHLightResponseCostC), [68](#)
- BerkeleyJulianDateToPOSIXct, [7](#), [20](#), [63](#)
- check, [32](#)
- cols, [25](#)
- DEGebExample, [8](#)
- envRefClass, [35](#), [49](#), [51](#), [65](#), [66](#), [69](#)
- estimate\_vpd\_from\_dew, [8](#), [83](#)
- Example\_DETha98, [9](#)
- extract\_FN15, [10](#)
- fCalcAVPfromVMFandPress, [10](#)
- fCalcETfromLE, [11](#)
- fCalcExtRadiation, [12](#)
- fCalcPotRadiation, [13](#), [95](#)
- fCalcRHfromAVPandTair, [14](#)
- fCalcSVPfromTair, [15](#)
- fCalcVPDfromRHandTair, [6](#), [15](#)
- fCheckHHTimeSeries, [16](#), [72](#)
- fConvertCtoK, [17](#)
- fConvertGlobalToVisible, [18](#)
- fConvertKtoC, [18](#)
- fConvertTimeToPosix, [5](#), [7](#), [19](#), [72](#)
- fConvertVisibleWm2toPhotons, [21](#)
- filterLongRuns, [21](#)
- filterLongRunsInVector, [21](#), [22](#)
- fLloydTaylor, [23](#), [94](#)
- fLoadEuroFlux16, [24](#), [26](#)
- fLoadFluxnet15, [25](#)
- fLoadTXTIntoDataframe, [5](#), [26](#)
- fWriteDataframeToFile, [27](#)
- get\_timestep\_hours, [33](#)
- getAmerifluxToBGC05VariableNameMapping, [28](#), [67](#)
- getBGC05ToAmerifluxVariableNameMapping, [29](#)
- getExamplePath, [30](#), [32](#)
- getFilledExampleDETha98Data, [31](#)
- getREddyProcExampleDir, [32](#)
- getTZone, [33](#)
- globalDummyVars, [34](#)
- image, [101](#)
- LightResponseCurveFitter, [34](#), [49](#), [51](#), [65](#), [66](#)
- LightResponseCurveFitter-class, [35](#)
- LightResponseCurveFitter\_computeCost, [35](#), [36](#), [46](#)
- LightResponseCurveFitter\_computeLRCGradient, [35](#), [37](#)
- LightResponseCurveFitter\_fitLRC, [35](#), [38](#), [45](#), [46](#), [62](#)
- LightResponseCurveFitter\_getOptimizedParameterPositions, [35](#), [39](#)
- LightResponseCurveFitter\_getParameterInitials, [35](#), [40](#)
- LightResponseCurveFitter\_getParameterNames, [35](#), [36](#), [41](#), [52](#)
- LightResponseCurveFitter\_getPriorLocation, [35](#), [41](#)
- LightResponseCurveFitter\_getPriorScale, [35](#), [42](#)

- LightResponseCurveFitter\_isParameterInBounds, [sEddyProc\\_sApplyUStarScen](#), [73](#), [87](#), [96](#), [35](#), [43](#)
- LightResponseCurveFitter\_optimLRC, [35](#), [44](#), [46](#), [47](#)
- LightResponseCurveFitter\_optimLRCBounds, [35](#), [39](#), [45](#)
- LightResponseCurveFitter\_optimLRCOnAdjustedPrior, [35](#), [45](#), [46](#)
- LightResponseCurveFitter\_predictGPP, [35](#), [47](#), [51](#), [53](#), [61](#), [67](#)
- LightResponseCurveFitter\_predictLRC, [35–37](#), [48](#)
- LogisticSigmoidLRCFitter, [49](#)
- LogisticSigmoidLRCFitter-class, [49](#)
- LogisticSigmoidLRCFitter\_predictGPP, [47](#), [50](#)
  
- NonrectangularLRCFitter, [51](#)
- NonrectangularLRCFitter-class, [51](#)
- NonrectangularLRCFitter\_getParameterNames, [52](#), [62](#)
- NonrectangularLRCFitter\_predictGPP, [47](#), [53](#), [53](#)
  
- optim, [39](#), [44](#)
  
- partGLControl, [38](#), [54](#), [57–59](#), [61](#)
- partGLControlLasslopCompatible, [54](#), [56](#)
- partGLExtractStandardData, [58](#), [61](#)
- partitionNEEGL, [48](#), [56](#), [60](#), [86](#)
- POSIXctToBerkeleyJulianDate, [7](#), [10](#), [63](#)
  
- read\_csv, [25](#)
- read\_from\_fluxnet15, [25](#), [63](#)
- RectangularLRCFitter, [64](#), [66](#)
- RectangularLRCFitter-class, [65](#)
- RectangularLRCFitter\_predictGPP, [47](#), [53](#), [66](#)
- RectangularLRCFitterCVersion, [65](#)
- RectangularLRCFitterCVersion-class, [66](#)
- REddyProc (REddyProc-package), [5](#)
- REddyProc-package, [5](#)
- renameVariablesInDataframe, [29](#), [30](#), [67](#)
- RHLightResponseCostC, [68](#)
  
- sDATA (globalDummyVars), [34](#)
- sEddyProc, [69](#)
- sEddyProc-class, [69](#)
- sEddyProc\_initialize, [6](#), [70](#), [71](#)
- sEddyProc\_sApplyUStarScen, [73](#), [87](#), [96](#), [107](#)
- sEddyProc\_sCalcPotRadiation, [70](#), [74](#)
- sEddyProc\_sEstimateUstarScenarios, [6](#), [74](#), [78](#), [84](#), [91](#), [116](#)
- sEddyProc\_sEstUstarThold, [70](#), [75](#), [76](#), [117](#)
- sEddyProc\_sEstUstarThreshold, [6](#), [75](#), [77](#), [120](#)
- sEddyProc\_sEstUstarThresholdDistribution, [6](#), [70](#), [78](#), [105](#), [117](#), [120](#), [121](#)
- sEddyProc\_sExportData, [70](#), [78](#)
- sEddyProc\_sExportResults, [70](#), [79](#), [91](#), [92](#)
- sEddyProc\_sFillInit, [70](#), [80](#), [89](#)
- sEddyProc\_sFillLUT, [70](#), [81](#), [89](#)
- sEddyProc\_sFillMDC, [70](#), [82](#), [89](#)
- sEddyProc\_sFillVPDFfromDew, [83](#)
- sEddyProc\_sGetData, [70](#), [83](#)
- sEddyProc\_sGetEstimatedUstarThresholdDistribution, [6](#), [75](#), [78](#), [84](#)
- sEddyProc\_sGetUstarScenarios, [6](#), [84](#), [85](#), [105](#), [108](#)
- sEddyProc\_sGetUstarSuffixes, [85](#)
- sEddyProc\_sGLFluxPartition, [6](#), [70](#), [86](#), [87](#), [106](#)
- sEddyProc\_sGLFluxPartitionUStarScens, [6](#), [87](#)
- sEddyProc\_sMDSGapFill, [6](#), [70](#), [88](#), [90–92](#)
- sEddyProc\_sMDSGapFillAfterUstar, [6](#), [59](#), [61](#), [70](#), [90](#), [91](#), [92](#)
- sEddyProc\_sMDSGapFillAfterUstarDistr, [70](#), [91](#)
- sEddyProc\_sMDSGapFillUStarScens, [6](#), [75](#), [91](#), [92](#), [92](#)
- sEddyProc\_sMRFluxPartition, [6](#), [70](#), [93](#), [96](#)
- sEddyProc\_sMRFluxPartitionUStarScens, [6](#), [95](#)
- sEddyProc\_sPlotDailySums, [6](#), [70](#), [96](#)
- sEddyProc\_sPlotDailySumsY, [70](#), [96](#), [97](#), [97](#)
- sEddyProc\_sPlotDiurnalCycle, [6](#), [70](#), [98](#)
- sEddyProc\_sPlotFingerprint, [6](#), [70](#), [99](#)
- sEddyProc\_sPlotFingerprintY, [70](#), [100](#), [100](#)
- sEddyProc\_sPlotHHFluxes, [6](#), [70](#), [101](#)
- sEddyProc\_sPlotHHFluxesY, [70](#), [102](#), [102](#)
- sEddyProc\_sPlotNEEversusUStarForSeason, [70](#), [103](#), [117](#)
- sEddyProc\_sSetLocationInfo, [70](#), [72](#), [104](#)
- sEddyProc\_sSetUstarScenarios, [6](#), [75](#), [84](#)

[85, 92, 105, 108, 109](#)  
 sEddyProc\_sSetUStarSeasons, [106](#)  
 sEddyProc\_sTKFluxPartition, [106, 107](#)  
 sEddyProc\_sTKFluxPartitionUStarScens,  
[6, 107](#)  
 sEddyProc\_update\_ustarthreshold\_columns,  
[108](#)  
 sEddyProc\_useAnnualUStarThresholds,  
[108, 109](#)  
 sEddyProc\_useSeasonalUStarThresholds,  
[108, 109](#)  
 sID (globalDummyVars), [34](#)  
 sINFO (globalDummyVars), [34](#)  
 sLOCATION (globalDummyVars), [34](#)  
 sPlotFingerprint  
     (sEddyProc\_sPlotFingerprint),  
[99](#)  
 sTEMP (globalDummyVars), [34](#)  
 strptime, [7](#)  
 sUSTAR (globalDummyVars), [34](#)  
  
 tempdir, [32](#)  
  
 usControlUstarEst, [75, 109, 116, 118, 119](#)  
 usControlUstarSubsetting, [75, 110, 111,](#)  
[111, 116, 117](#)  
 usCreateSeasonFactorMonth, [112, 114–116](#)  
 usCreateSeasonFactorMonthWithinYear,  
[113, 113, 117](#)  
 usCreateSeasonFactorYday, [113, 114, 117](#)  
 usCreateSeasonFactorYdayYear, [113, 115,](#)  
[117](#)  
 usEstUstarThreshold, [76, 77, 116](#)  
 usEstUstarThresholdSingleFw1Binned,  
[117, 118](#)  
 usEstUstarThresholdSingleFw2Binned,  
[110, 111, 116, 117, 119](#)  
 usGetAnnualSeasonUStarMap, [72, 105, 120](#)  
 usGetSeasonalSeasonUStarMap, [105, 120](#)  
 usGetYearOfSeason, [121](#)