Package: WtRegDO (via r-universe)

September 5, 2024

Type Package

Title Implement Weighted Regression on Dissolved Oxygen Time Series

Version 1.0.2

Date 2024-05-31

Author Marcus W. Beck [aut, cre]

Maintainer Marcus W. Beck <mbafs2012@gmail.com>

Description A sample dataset and functions to implement weighted regression on dissolved oxygen time series are included as a simple example to reduce the effects of tidal advection. Functions are also available to estimate net ecosystem metabolism using the open-water method.

LazyData TRUE

BugReports https://github.com/fawda123/WtRegDO/issues

License CC0

Imports data.table, dplyr, foreach, ggplot2, lubridate, plyr, oce, reshape2, suncalc, tibble, tidyr

Suggests doParallel

Depends R (>= 3.5.0)

RoxygenNote 7.2.3

Repository https://fawda123.r-universe.dev

RemoteUrl https://github.com/fawda123/WtRegDO

RemoteRef HEAD

RemoteSha 4ea340e1e1e5fc8c93df83e9a7e981b08d1939a8

Contents

`aggregate.metab` *Aggregete metabolism data*

Description

Aggregate metabolism data for a metab object

Usage

```
## S3 method for class 'metab'
aggregate(x, by = "weeks", na.action = "na.pass", alpha = 0.05, ...)
```

Arguments

`x`	input data object as returned by `ecometab`
`by`	character string indicating aggregation period or numeric value indicating moving window width for daily averages
`na.action`	function for treating missing data, default `na.pass`
`alpha`	level for estimating confidence intervals in aggregated data
`...`	arguments passed to or from other methods

climmeans *Calculate climate means for relevant weather variables*

Description

Calculate climate means for relevant weather variables

Usage

```
climmeans(dat_in, gasex = c("Thiebault", "Wanninkhof"))
```

Arguments

dat_in	Input data frame as a similar format required for ecometab
gasex	chr indicating if gas exchange is estimated using equations in Thiebault et al. 2008 or Wanninkhof 2014 (see f_calcKL or f_calcWanninkhof)

Details

Function is used internally within ecometab. Missing values for weather variables are replaced by the monthly/hourly average calculated from the available data in dat_in. If gasex = 'Thiebault', this applies to air temperature, wind speed, and barometric pressure. If gasex = 'Wanninkhof', this applies to wind speed and barometric pressure, where the latter is not needed for the Wanninkhof method, but is required for dissolved oxygen at saturation (oxySol) with gas exchange.

Value

The same data frame as in dat_in, except missing values for the relevant weather variables are replaced with estimated means.

Examples

```
climmeans(SAPDC)
```

ecometab *Ecosystem metabolism*

Description

Estimate ecosystem metabolism using the Odum open-water method. Estimates of daily integrated gross production, total respiration, and net ecosystem metabolism are returned. A plotting method is also provided.

Usage

```
ecometab(dat_in, ...)

## Default S3 method:
ecometab(
  dat_in,
  tz,
  DO_var = ”DO_mgl”,
  depth_val = ”Tide”,
  metab_units = ”mmol”,
  bott_stat = FALSE,
  depth_vec = NULL,
  replacemet = TRUE,
  instant = FALSE,
  gasex = c(”Thiebault”, ”Wanninkhof”),
  gasave = c(”instant”, ”daily”, ”all”),
  ...
)

## S3 method for class 'metab'
plot(
  x,
  by = ”months”,
  metab_units = ”mmol”,
  alpha = 0.05,
  width = 10,
  pretty = TRUE,
  ...
)
```

Arguments

`dat_in` Input data frame which must include time series of dissolved oxygen (mg L-1), see `SAPDC` for data structure

`...` arguments passed to or from other methods

`tz` chr string for timezone, e.g., ’America/Chicago’, must match the time zone in `dat_in$DateTimeStamp`

`DO_var` chr string indicating the name of the column with the dissolved oxygen variable for estimating metabolism

`depth_val` chr indicating the name of a column in the input data for estimating depth for volumetric integration. This column is typically the tidal height vector. Use `depth_val` = `NULL` if supplying an alternative depth vector to `depth_vec`.

`metab_units` chr indicating desired units of output for oxygen, either as mmol or grams

`bott_stat` logical if air-sea gas exchange is removed from the estimate

`depth_vec` numeric value for manual entry of station depth (m). Use a single value if the integration depth is constant or a vector of depth values equal in length to the time series. Leave `NULL` if estimated from `depth_val` column.

`replacemet` logical indicating if missing values for appropriate weather variables are replaced by monthly/hourly means with `climmeans`

`instant` logical indicating if the instantaneous data (e.g., 30 minutes observations) used to estimate the daily metabolic rates are returned, see details

`gasex` chr indicating if gas exchange is estimated using equations in Thiebault et al. 2008 or Wanninkhof 2014 (see `f_calcKL` or `f_calcWanninkhof`)

`gasave` chr indicating one of `"instant"` (default), `"daily"`, or `"all"` indicating if gas exchange estimates are based on instantaneous estimates, averaged within a day prior to estimating metabolism, or averaged across the entire period record. All options require an instantaneous record as a starting point.

`x` input object to plot

`by` chr string describing aggregation period, passed to `aggregate`. See details for accepted values.

`alpha` numeric indicating alpha level for confidence intervals in aggregated data. Use `NULL` to remove from the plot.

`width` numeric indicating width of top and bottom segments on error bars

`pretty` logical indicating use of predefined plot aesthetics

Details

Input data include both water quality and weather time series, which are typically collected with independent instrument systems. This requires merging of the time series datasets. These include time series of dissolved oxygen, salinity, air and water temperature, barometric pressure, and wind speed (see `SAPDC` for an example of the data structure for `ecometab`).

The open-water method is a common approach to quantify net ecosystem metabolism using a mass balance equation that describes the change in dissolved oxygen over time from the balance between photosynthetic and respiration processes, corrected using an empirically constrained air-sea gas diffusion model (see Ro and Hunt 2006, Thebault et al. 2008). The diffusion-corrected DO flux estimates are averaged separately over each day and night of the time series. The nighttime average DO flux is used to estimate respiration rates, while the daytime DO flux is used to estimate net primary production. To generate daily integrated rates, respiration rates are assumed constant such that hourly night time DO flux rates are multiplied by 24. Similarly, the daytime DO flux rates are multiplied by the number of daylight hours, which varies with location and time of year, to yield net daytime primary production. Respiration rates are subtracted from daily net production estimates to yield gross production rates. The metabolic day is considered the 24 hour period between sunrises on two adjacent calendar days.

Areal rates for gross production and total respiration are based on volumetric rates normalized to the depth of the water column at the sampling location, which is assumed to be well-mixed, such that the DO sensor is reflecting the integrated processes in the entire water column (including the benthos). Water column depth is calculated as the mean value of the depth variable across the time series. Depth values are floored at one meter for very shallow stations and 0.5 meters is also added to reflect the practice of placing sensors slightly off of the bottom. Additionally, the air-sea gas exchange model is calibrated with wind data either collected at, or adjusted to, wind speed at 10 m above the surface. The metadata should be consulted for exact height. The value can be changed manually using a `height` argument, which is passed to `f_calcKL`.

A minimum of three records are required for both day and night periods to calculate daily metabolism estimates. Occasional missing values for air temperature, barometric pressure, and wind speed are replaced with the climatological means (hourly means by month) for the period of record using adjacent data within the same month as the missing data.

All DO calculations within the function are done using molar units (e.g., mmol O2 m-3).

The specific approach for estimating metabolism with the open-water method is described in Caffrey et al. 2013 and references cited therein.

The plotting method plots daily metabolism estimates using different aggregation periods. Accepted aggregation periods are `'years'`, `'quarters'`, `'months'`, `'weeks'`, and `'days'` (if no aggregation is preferred). The default function for aggregating is the `mean` for the periods specified by the `by` argument. Setting `pretty = FALSE` will return the plot with minimal modifications to the `ggplot` object.

Value

A `metab` object with daily integrated metabolism estimates including gross production (Pg, mmol O2 m-2 d-1), total respiration (Rt), and net ecosystem metabolism (NEM). Attributes of the object include the raw data (`rawdat`), a character string indicating name of the tidal column if supplied in the raw data (`depth_val`), and a character string indicating name of the dissolved oxygen column in the raw data that was used to estimate metabolism (`DO_var`).

The plot method returns a `ggplot` object which can be further modified.

If `instant = TRUE` the instantaneous data (e.g., 30 minutes observations) used to estimate the daily metabolic rates are returned at the midpoint time steps from the raw time series. The instantaneous data will also return metabolism estimates as flux per day, including the DO flux (dDO, mmol d-1), air-sea exchange rate (D, mmol m-2 d-1), the volumetric reaeration coefficient (Ka, hr-1), the gas transfer coefficient (KL, m d-1), gross production (Pg, mmol O2 m-2 d-1), respiration (Rt, mmol O2 m-2 d-1), net ecosystem metabolism (mmol O2 m-2 d-1), volumetric gross production (Pg_vol, mmol O2 m-3 d-1), volumetric respiration (Rt_vol, mmol O2 m-3 d-1), and volumetric net ecosystem metabolism (mmol O2 m-3 d-1). If `metab_units = "grams"`, the same variables are returned as grams of O2. The daily and nightly DO and gas exchange fluxes are also returned in units per hour (`DOF_d`, `D_d`, `DOF_n`, `D_n`). Note that `NA` values are returned for gross production and NEM during "sunset" hours as production is assumed to not occur during the night.

References

Caffrey JM, Murrell MC, Amacker KS, Harper J, Phipps S, Woodrey M. 2013. Seasonal and interannual patterns in primary production, respiration and net ecosystem metabolism in 3 estuaries in the northeast Gulf of Mexico. Estuaries and Coasts. 37(1):222-241.

Odum HT. 1956. Primary production in flowing waters. Limnology and Oceanography. 1(2):102-117.

Ro KS, Hunt PG. 2006. A new unified equation for wind-driven surficial oxygen transfer into stationary water bodies. Transactions of the American Society of Agricultural and Biological Engineers. 49(5):1615-1622.

Thebault J, Schraga TS, Cloern JE, Dunlavey EG. 2008. Primary production and carrying capacity of former salt ponds after reconnection to San Francisco Bay. Wetlands. 28(3):841-851.

See Also

f_calcKL for estimating the oxygen mass transfer coefficient used with the air-sea gas exchange model and met_day_fun for identifying the metabolic day for each observation in the time series

Examples

```
## Not run:
data(SAPDC)

# metadata for the location
tz <- 'America/Jamaica'
lat <- 31.39
long <- -81.28

# estimate ecosystem metabolism using observed DO time series
metab <- ecometab(SAPDC, DO_var = 'DO_obs', tz = tz,
 lat = lat, long = long)

## plot
plot(metab)

## change alpha, aggregation period, widths
plot(metab, by = 'quarters', alpha = 0.1, widths = 0)

## plot daily raw, no aesthetics
plot(metab, by = 'days', pretty = FALSE)

## End(Not run)
```

evalcor *Evaluate tide and sun correlation*

Description

Evaluate the correlation between tide change and sun angle to determine potential effectiveness of weighted regression

Usage

```
evalcor(
  dat_in,
  tz,
  lat,
  long,
  depth_val = "Tide",
  daywin = 6,
  method = "pearson",
  plot = TRUE,
  lims = c(-0.5, 0.5),
  progress = FALSE,
  harm = TRUE,
  chk_tide = FALSE,
 constituents = c("M2", "S2", "N2", "K2", "K1", "O1", "P1", "Q1", "MF", "MM", "SSA",
    "M4", "M6", "S4", "MS4")
)
```

Arguments

`dat_in` Input `data.frame`

`tz` chr string for timezone, e.g., 'America/Chicago', must match the time zone in `dat_in$DateTimeStamp`

`lat` numeric for latitude

`long` numeric for longitude (negative west of prime meridian)

`depth_val` chr indicating name of the tidal height column in `dat_in`

`daywin` numeric for half-window width used in moving window correlation

`method` chr string for corrrelation method, passed to `cor`

`plot` logical to return a plot

`lims` two element numeric vector indicating y-axis limits on plot

`progress` logical if progress is saved to a text file named 'log.txt' in the working directory

`harm` logical indicating if the tidal height vector indicated in `depth_val` is modelled using harmonic regression, see details

`chk_tide` logical indicating if harmonic regression output is returned for diagnostics

`constituents` chr string of harmonic constituents to predict if `harm = TRUE`

Details

This function can be used before weighted regression to identify locations in the time series when tidal and solar changes are not correlated. In general, the `wtreg` will be most effective when correlations between the two are zero, whereas `wtreg` will remove both the biological and physical components of the dissolved oxygen time series when the sun and tide are correlated. The correlation between tide change and sun angle is estimated using a moving window for the time series, where the half-window width is defined by `daywin` (i.e., the default value is a moving window with six days on each side for 12 days total). Tide changes are estimated as angular rates for the tidal height vector and sun angles are estimated from the time of day and geographic location.

The `foreach` function is used to execute the moving window correlation in parallel and will be run automatically if a backend is created.

Setting `harm = TRUE` will predict the tidal time series with harmonic regression using the `tidem` function. This is useful if there are missing observations in the observed tidal vector. The correlation time series in the plot will also be smoother. Use the `chk_tide` function before applying this option to verify the harmonic regression model is adequate for the observed time series. The predicted tidal constituents in the default argument should account for a majority of the variation in the observed data. These include M2: principal lunar semidiurnal, S2: principal solar semidiurnal, N2: larger lunar elliptic semidiurnal, K2: lunisolar semidiurnal, K1: lunisolar declinational diurnal, O1: principal lunar diurnal, P1: principal solar diurnal, Q1: larger lunar elliptic diurnal, MF: lunar fortnightly, MM: lunar monthly, SSA: solar semi annual, M4: first overtide of M2, M6: second overtide of M2, S4: first overtide of S2, and MS4: compound tide of M2 and S2.

Figure 9 in Beck et al. 2015 was created using this function.

Value

A ggplot object if plot = TRUE, otherwise a numeric vector of the correlations for each row in the input dataset. A two-element list will be returned for the tidal model and predicted resutls of chk_tide = TRUE

References

Beck MW, Hagy III JD, Murrell MC. 2015. Improving estimates of ecosystem metabolism by reducing effects of tidal advection on dissolved oxygen time series. Limnology and Oceanography Methods. DOI: 10.1002/lom3.10062

See Also

wtreg

Examples

```
## Not run:

data(SAPDC)

# metadata
tz <- 'America/Jamaica'
lat <- 31.39
long <- -81.28

# setup parallel backend
library(doParallel)
registerDoParallel(cores = 7)

evalcor(SAPDC, tz, lat, long, progress = TRUE)

# check fit of tidal predictions
# in this case, predictions = observed because sample data are already predicted
tochk <- evalcor(SAPDC, tz, lat, long, progress = TRUE, chk_tide = TRUE)
tochk <- tochk$tide_pred
plot(tide_obs ~ tide_pred, tochk)


## End(Not run)
```

f_calcKL *Calculate oxygen mass transfer coefficient*

Description

Calculate oxygen mass transfer coefficient using equations in Thiebault et al. 2008. Output is used to estimate the volumetric reaeration coefficient for ecosystem metabolism.

Usage

```
f_calcKL(Temp, Sal, ATemp, WSpd, BP, Height = 10)
```

Arguments

`Temp`	numeric for water temperature (C)
`Sal`	numeric for salinity (ppt)
`ATemp`	numeric for air temperature (C)
`WSpd`	numeric for wind speed (m/s)
`BP`	numeric for barometric pressure (mb)
`Height`	numeric for height of anemometer (meters)

Details

This function is used within the `ecometab` function and should not be used explicitly.

References

Ro KS, Hunt PG. 2006. A new unified equation for wind-driven surficial oxygen transfer into stationary water bodies. Transactions of the American Society of Agricultural and Biological Engineers. 49(5):1615-1622.

Thebault J, Schraga TS, Cloern JE, Dunlavey EG. 2008. Primary production and carrying capacity of former salt ponds after reconnection to San Francisco Bay. Wetlands. 28(3):841-851.

See Also

`ecometab`

f_calcWanninkhof *Calculate gas transfer velocity for Wanninkhof equation*

Description

Calculate gas transfer velocity for Wanninkhof equation

Usage

```
f_calcWanninkhof(Temp, Sal, WSpd)
```

Arguments

`Temp`	numeric for water temperature (C)
`Sal`	numeric for salinity (ppt)
`WSpd`	numeric for wind speed (m/s)

Details

Output is Kw vector that is alternative to calculating KL using Thiebault et al. 2008 (`f_calcKL`). Interpreted as oxygen mass transfer coefficient.

Value

numeric vector of Kw in m/d

References

Wanninkhof, R. 2014. Relationship between wind speed and gas exchange over the ocean revisited. Limnology and Oceanograpy Methods. 12(6):351-362. 10.4319/lom.2014.12.351

Examples

```
data(SAPDC)
f_calcWanninkhof(SAPDC$Temp, SAPDC$Sal, SAPDC$WSpd)
```

metab_dtd *Ecosystem metabolism for SAPDC from detided DO*

Description

Ecosystem metabolism results for SAPDC from detided dissolved oxygen time series. The dataset was created by running `wtreg` on the sample dataset for Sapelo Island Dean Creek station, 2012 data. Each row represents a daily estimate or average for each metabolic day defined as the period between sunrises for two calendar days. See documentation for `ecometab` for a description of the object attributes.

Usage

```
metab_dtd
```

Format

A metab object with 367 rows and 4 variables:

Date Date, metabolic day

Pg numeric, gross production, mmol m-2 d-1

Rt numeric, total respiration, mmol m-2 d-1

NEM numeric, net ecosytem metabolism, mmol m-2 d-1

metab_obs *Ecosystem metabolism for SAPDC from observed DO*

Description

Ecosystem metabolism results for SAPDC from observed dissolved oxygen time series. The dataset was created by running `wtreg` on the sample dataset for Sapelo Island Dean Creek station, 2012 data. Each row represents a daily estimate or average for each metabolic day defined as the period between sunrises for two calendar days. See documentation for `ecometab` for a description of the object attributes.

Usage

```
metab_obs
```

Format

A metab object with 367 rows and 4 variables:

Date Date, metabolic day

Pg numeric, gross production, mmol m-2 d-1

Rt numeric, total respiration, mmol m-2 d-1

NEM numeric, net ecosytem metabolism, mmol m-2 d-1

meteval *Evaluate metabolism results*

Description

Evaluate metabolism results before and after weighted regression

Usage

```
meteval(metab_in, ...)

## S3 method for class 'metab'
meteval(metab_in, all = TRUE, ...)
```

Arguments

`metab_in`	input `metab` object as returned from `ecometab`
`...`	additional arguments passed to other methods
`all`	logical indicating if all evaluation summaries are returned or just mean, sd, and percent anomalies

Details

This function provides summary statistics of metabolism results to evaluate the effectiveness of weighted regression. These estimates are mean production, standard deviation of production, percent of production estimates that were anomalous, mean respiration, standard deviation of respiration, percent of respiration estimates that were anomalous, correlation of dissolved oxygen with tidal height changes, correlation of production with tidal height changes, and the correlation of respiration with tidal height changes. The correlation estimates are based on an average of the correlations by each month in the time series from the raw data for dissolved oxygen and the daily results for the metabolic estimates. Dissolved oxygen is correlated directly with tidal height at each time step. The metabolic estimates are correlated with the tidal height ranges during the day for production and during the night for respiration. Tidal height ranges are estimated from the raw data during each diurnal period for each metabolic day.

In general, useful results for weighted regression are those that remove the correlation of dissolved oxygen, production, and respiration with tidal changes. Similarly, the mean estimates of metabolism should not change if a long time series is evaluated, whereas the standard deviation and percent anomalous estimates should decrease.

Tables 2 and 3 in Beck et al. 2015 were created using these methods.

Value

A two-element list of summary statistics for the complete period of record (`cmp`) and by month (`mos`). The complete record summary has columns named `meanPg`, `sdPg`, `anomPg`, `meanRt`, `sdRt`, `anomRt`. The monthly summary has `DOcor`, `Pgcor`, `Rtcor` for the correlations of each with the tidal cycle for the given month and `anomPg` and `anomRt` for the anomalous tallies of the metabolism estimates in each month. See the details above for a meaning of each.

References

Beck MW, Hagy III JD, Murrell MC. 2015. Improving estimates of ecosystem metabolism by reducing effects of tidal advection on dissolved oxygen time series. Limnology and Oceanography Methods. DOI: 10.1002/lom3.10062

See Also

ecometab

Examples

```
## Not run:

# load library and sample data
# metab_obs and metab_dtd
 library(WtRegDO)
 data(metab_obs)
 data(metab_dtd)

 meteval(metab_obs)
 meteval(metab_dtd)

## End(Not run)
```

met_day_fun *Identify metabolic days in a swmpr time series*

Description

Identify metabolic days in a time series based on sunrise and sunset times for a location and date. The metabolic day is considered the 24 hour period between sunrises for two adjacent calendar days.

Usage

```
met_day_fun(dat_in, tz, lat, long)
```

Arguments

dat_in	data.frame
tz	chr string for timezone, e.g., 'America/Chicago', must match the time zone in dat_in$DateTimeStamp
lat	numeric for latitude
long	numeric for longitude (negative west of prime meridian)

Details

This function is only used within ecometab and should not be called explicitly.

See Also

ecometab

---

objfun *An objective function to minimize for finding optimal window widths*

Description

An objective function to minimize for finding optimal window widths

Usage

```
objfun(
  metab_obs,
  metab_dtd,
  vls = c("meanPg", "sdPg", "anomPg", "meanRt", "sdRt", "anomRt")
)
```

Arguments

`metab_obs`	A metab object estimated the observed dissolved oxygen time series
`metab_dtd`	A metab object estimated the detided dissolved oxygen time series
`vls`	chr vector of summary evaluation object to optimize, see details

Details

This function is an attempt to quantify a relative measure of comparison to evaluate metabolism estimates from observed and detided dissolved oxygen time series. It is the sole function that is optimized when identifying window widths that produce "best" detided metabolism estimates. The summary is based on an assumption that a detided estimate provides an improved measure of metabolism following several rules of thumb. Specifically, improved estimates are assumed to have lower anomalies (less negative production and positive respiration values), lower standard deviation, and similar mean values for gross production and respiration between the observed and detided estimates.

The quantification of improved fit is based on a sum of percent differences for the six paired measures for percent anomalous production, percent anomalous respiration, mean production, mean respiration, standard deviation of production, and standard deviation of respiration for the estimates from the observed and detided metabolism. The comparisons of the means are taken as the inverse (1 / mean) such that optimization should attempt to keep the values as similar as possible. The final sum is multiplied by negative one such that the value is to be optimized by minimization, i.e., a lower value indicates improved detiding across all measures.

The function can also quantify a comparison based on different measures supplied by the user. By default, all six measures are used. However, selecting specific measures, such as only optimizing by reducing anomalous values, may be preferred. Changing the argument for `vls` changes which comparisons are used for the summary value.

Value

A single numeric value indicating the estimate from the objective function

See Also

wtobjfun, winopt

Examples

```
# estimate a summary value for all six measures
objfun(metab_obs, metab_dtd)

# estimate a summary value for only anomalies
objfun(metab_obs, metab_dtd, vls = c('anomPg', 'anomRt'))
```

oxySchmidt *Calculate Schmidt number for oxygen*

Description

Calculate Schmidt number for oxygen

Usage

```
oxySchmidt(Temp, Sal)
```

Arguments

Temp	numeric for water temperature (C)
Sal	numeric for salinity (ppt)

Value

Cald Sc at given salinity for oxygen, unitless

Examples

```
data(SAPDC)
oxySchmidt(SAPDC$Temp, SAPDC$Sal)
```

oxySol *Dissolved oxygen at saturation*

Description

Finds dissolved oxygen concentration in equilibrium with water-saturated air. Function and documentation herein are from archived wq package: https://cran.r-project.org/package=wq

Usage

```
oxySol(t, S, P = NULL)
```

Arguments

t	tem temperature, degrees C
S	salinity, on the Practical Salinity Scale
P	pressure, atm

Details

Calculations are based on the approach of Benson and Krause (1984), using Green and Carritt's (1967) equation for dependence of water vapor partial pressure on t and S. Equations are valid for temperature in the range 0-40 C and salinity in the range 0-40.

Value

Dissolved oxygen concentration in mg/L at 100% saturation. If P = NULL, saturation values at 1 atm are calculated.

References

Benson, B.B. and Krause, D. (1984) The concentration and isotopic fractionation of oxygen dissolved in fresh-water and seawater in equilibrium with the atmosphere. *Limnology and Oceanography* **29,** 620-632.

Green, E.J. and Carritt, D.E. (1967) New tables for oxygen saturation of seawater. *Journal of Marine Research* **25,** 140-147.

Examples

```
# Convert DO into % saturation for 1-m depth at Station 32.
# Use convention of expressing saturation at 1 atm.
data(sfbay)

sfb1 <- subset(sfbay, depth == 1 & stn == 32)
dox.pct <- with(sfb1, 100 * dox/oxySol(temp, sal))
summary(dox.pct)
```

SAPDC *Sample dataset for weighted regression*

Description

Sample dataset for weighted regression, Sapelo Island Dean Creek station, 2012 data. Only DateTimeSTamp, DO_obs, and Tide are needed for weighted regression, whereas all remaining columns are needed for estimating ecosystem metabolism. Metadata about the location should also be available including the timezone and lat/long coordinates.

Usage

```
SAPDC
```

Format

A data frame with 17568 rows and 9 variables:

DateTimeStamp POSIXct, timestamp of water quality observation

Temp numeric, water temperature, celsius

Sal numeric, salinity, ppt

DO_obs numeric, dissolved oxygen, mg L-1

ATemp numeric, air temperature, celsius

BP numeric, barometric pressure, mb

WSpd numeric, wind speed, m s-1

Tide numeric, tide height, m, estimated from pressure data using harmonic regression

sfbay *San Francisco Bay water quality data*

Description

Selected observations and variables from U.S. Geological Survey water quality stations in south San Francisco Bay. Data include CTD and nutrient measurements. Data and documentation herein are from archived wq package: https://cran.r-project.org/package=wq

Usage

```
sfbay
```

Format

`sfbay` is a data frame with 23207 observations (rows) of 12 variables (columns):

`[, 1]`	`date`	date
`[, 2]`	`time`	time
`[, 3]`	`stn`	station code
`[, 4]`	`depth`	measurement depth
`[, 5]`	`chl`	chlorophyll *a*
`[, 6]`	`dox.pct`	dissolved oxygen
`[, 7]`	`spm`	suspended particulate matter
`[, 8]`	`ext`	extinction coefficient
`[, 9]`	`sal`	salinity
`[, 10]`	`temp`	water temperature
`[, 11]`	`nox`	nitrate + nitrite
`[, 12]`	`nhx`	ammonium

Details

The original downloaded dataset was modified by taking a subset of six well-sampled stations and the period 1985–2004. Variable names were also simplified.

Source

Downloaded from `http://sfbay.wr.usgs.gov/access/wqdata` on 2009-11-17.

`smoother` *Smooth a plot of metabolism data*

Description

Smooth a plot of metabolism data using a moving window average

Usage

```
smoother(x, ...)

## Default S3 method:
smoother(x, window = 5, sides = 2, ...)
```

Arguments

`x` input object

`...` additional arguments passed to `filter`

`window` numeric vector defining size of the smoothing window, passed to `filter`

`sides` numeric vector defining method of averaging, passed to `filter`

Details

This function uses a moving window average to smooth metabolism data for plotting. It has nothing to do with weighted regression (`wtreg`) and is meant only for plotting aesthetics. The function is a simple wrapper to `filter`. The window argument specifies the number of observations included in the moving average. The sides argument specifies how the average is calculated for each observation (see the documentation for `filter`). A value of 1 will filter observations within the window that are previous to the current observation, whereas a value of 2 will filter all observations within the window centered at zero lag from the current observation.

Value

Returns a `data.frame` of the smoothed metabolism data.

See Also

`filter`

Examples

```
## Not run:
data(SAPDC)

# metadata for the location
tz <- 'America/Jamaica'
lat <- 31.39
long <- -81.28

# estimate ecosystem metabolism using observed DO time series
metab <- ecometab(SAPDC, DO_var = 'DO_obs', tz = tz,
 lat = lat, long = long)

# smooth metabolism data with 20 day moving window average
tosmooth <- metab[, c('Pg', 'Rt', 'NEM')]
smoother(tosmooth, window = 20)

## End(Not run)
```

`winopt` *Find the optimal half-window width combination*

Description

Find the optimal half-window width combination to use for weighted regression.

Usage

```
winopt(
  dat_in,
  tz,
  lat,
  long,
  wins,
  vls = c("meanPg", "sdPg", "anomPg", "meanRt", "sdRt", "anomRt"),
  parallel = F,
  progress = T,
  control = list(factr = 1e+07, parscale = c(50, 100, 50)),
  lower = c(0.1, 0.1, 0.1),
  upper = c(12, 12, 1)
)
```

Arguments

`dat_in` input data frame

`tz` chr string specifying timezone of location, e.g., ’America/Jamaica’ for EST, no daylight savings, must match the time zone in `dat_in$DateTimeStamp`

`lat` numeric for latitude of location

`long` numeric for longitude of location (negative west of prime meridian)

`wins` list of half-window widths to use in the order specified by `wtfun` (i.e., days, hours, tide height).

`vls` chr vector of summary evaluation object to optimize, see details for `objfun`

`parallel` logical if regression is run in parallel to reduce processing time, requires a parallel backend outside of the function

`progress` logical if progress saved to a txt file names ’log.txt’ in the working directory,

`control` A list of control parameters passed to `optim` (see details in `optim` help file). The value passed to `factr` controls the convergence behavior of the `"L-BFGS-B"` method. Values larger than the default will generally speed up the optimization with a potential loss of precision. `parscale` describes the scaling values of the parameters.

`lower` vector of minimum half-window widths to evaluate

`upper` vector of maximum half-window widths to evaluate

Details

This is a super sketchy function based on many assumptions, see details in `objfun`

Value

Printed text to the console showing progress. Output from `optim` will also be returned if convergence is achieved.

See Also

objfun, wtobjfun

Examples

```
## Not run:

library(foreach)
library(doParallel)

data(SAPDC)

tz <- 'America/Jamaica'
lat <- 31.39
long <- -81.28

ncores <- detectCores()
cl <- makeCluster(ncores)
registerDoParallel(cl)

winopt(SAPDC, tz = tz, lat = lat, long = long, wins = list(6, 6, 0.5), parallel = T)

stopCluster(cl)

## End(Not run)
```

wtfun *Get weights used during weighted regression*

Description

Get weights used during weighted regression for a single observation in the dissolved oxygen time series

Usage

```
wtfun(
  ref_in,
  dat_in,
  wt_vars = c("dec_time", "hour", "Tide"),
  wins = list(4, 12, NULL),
  all = FALSE,
  slice = TRUE,
  subs_only = FALSE
)
```

Arguments

`ref_in` one row of the data frame of `dat_in` that is teh center of the window

`dat_in` data frame for estimating weights

`wt_vars` chr string indicating names of weighting variables

`wins` numeric vecotr for windows for the three wt variables, values represent halves. A `NULL` value for Tide specifies the half-window width is set automatically to one half the tidal range.

`all` logical to return all weights, rather than the product of all three

`slice` logical for subsetting `dat_in` for faster wt selection

`subs_only` logical for returning only wt vectors that are non-zero

Details

The default behavior is to subset the data frame for faster wt selection by limiting the input the maximum window size. Subsetted weights are recombined to equal a vector of length equal to the original data.

See Also

`wtreg`

`wtobjfun` *An objective function to minimize plus weighted regression for finding optimal window widths*

Description

An objective function to minimize plus weighted regression for finding optimal window widths

Usage

```
wtobjfun(
  wins,
  dat_in,
  tz,
  lat,
  long,
  metab_obs,
  strt = NULL,
  vls = c("meanPg", "sdPg", "anomPg", "meanRt", "sdRt", "anomRt"),
  parallel = F,
  progress = T
)
```

Arguments

wins	list of half-window widths to use in the order specified by wtfun (i.e., days, hours, tide height).
dat_in	input data frame
tz	chr string specifying timezone of location, e.g., 'America/Jamaica' for EST, no daylight savings, must match the time zone in dat_in$DateTimeStamp
lat	numeric for latitude of location
long	numeric for longitude of location (negative west of prime meridian)
metab_obs	A metab object returned by ecometab based on the observed DO time series in dat_in, used as comparison for the objective function
strt	a POSIXct object returned by Sys.time
vls	chr vector of summary evaluation object to optimize, see details for objfun
parallel	logical if regression is run in parallel to reduce processing time, requires a parallel backend outside of the function
progress	logical if progress saved to a txt file names 'log.txt' in the working directory

Value

A single numeric value to minimize, as output from objfun

See Also

objfun, winopt

Examples

```
## Not run:

library(foreach)
library(doParallel)

data(SAPDC)

tz <- 'America/Jamaica'
lat <- 31.39
long <- -81.28

metobs <- ecometab(SAPDC, DO_var = 'DO_obs', tz = tz, lat = lat, long = long)

ncores <- detectCores()
cl <- makeCluster(ncores)
registerDoParallel(cl)

wtobjfun(SAPDC, tz = tz, lat = lat, long = long, metab_obs = metobs, strt = Sys.time(),
   wins = list(6, 6, 0.5), parallel = T)

stopCluster(cl)

## End(Not run)
```

wtreg *Weighted regression for dissolved oxygen time series*

Description

Use weighted regression to reduce effects of tidal advection on dissolved oxygen time series

Usage

```
wtreg(
  dat_in,
  DO_obs = "DO_obs",
  depth_val = "Tide",
  wins = list(4, 12, NULL),
  tz,
  lat,
  long,
  progress = FALSE,
  parallel = FALSE,
  sine = F,
  ...
)
```

Arguments

`dat_in` input data frame

`DO_obs` name of dissolved oxygen column

`depth_val` name of tidal height column

`wins` list of half-window widths to use in the order specified by `wtfun` (i.e., days, hours, tide height).

`tz` chr string specifying timezone of location, e.g., 'America/Jamaica' for EST, no daylight savings, must match the time zone in `dat_in$DateTimeStamp`

`lat` numeric for latitude of location

`long` numeric for longitude of location (negative west of prime meridian)

`progress` logical if progress saved to a txt file names 'log.txt' in the working directory

`parallel` logical if regression is run in parallel to reduce processing time, requires a parallel backend outside of the function

`sine` logical if a sinusoidal curve is used in the regression

`...` additional arguments passed to `met_day_fun`, particularly timezone, lat, and long information.

Details

See the supplied dataset for required input data. The wtreg function only requires date/time, dissolved oxygen, and tidal height columns.

Timezone specifications can be found here: https://en.wikipedia.org/wiki/List_of_tz_database_time_zones

Value

The original data frame with additional columns describing the metabolic day, decimal time, the slope estimate for DO relative to tidal height for each window (Beta2), predicted DO from weighted regression (DO_prd) and detided (normalized) DO from weighted regression (DO_nrm).

Examples

```
## Not run:
data(SAPDC)

tz <- 'America/Jamaica'
lat <- 31.39
long <- -81.28

res <- wtreg(SAPDC, tz = tz, lat = lat, long = long)


## End(Not run)
```

---

wtreg_res *Results from weighted regression with the SAPDC dataset*

Description

Results after detiding the SAPDC sample dataset with wtreg. The dataset is identical to SAPDC with the addition of eight columns that were used during regression.

Usage

```
wtreg_res
```

Format

A data.frame with 17568 rows and 16 columns.

DateTimeStamp POSIXct, timestamp of water quality observation

Temp numeric, water temperature, celsius

Sal numeric, salinity, ppt

DO_obs numeric, dissolved oxygen, mg L-1

ATemp numeric, air temperature, celsius

BP numeric, barometric pressure, mb

WSpd numeric, wind speed, m s-1

Tide numeric, tide height, m, estimated from pressure data using harmonic regression

metab_date Date, the metabolic day defined as the period from sunrise to sunrise on two adjacent calender days

solar_period Factor, identifier that categorizes each time step as occuring during daylight (`sunrise`) or nighttime (`sunset`) based on sunrise and sunset times in the `value` column

solar_time POSIXct, sequential sunrise and sunset times for the time series

day_hrs numeric, total time of sunlight in the metabolic day

dec_time numeric, decimal time in days

hour numeric, hour of the day

DO_prd numeric, predicted dissolved oxygen from `wtreg`

DO_nrm numeric, detided (or normalized) dissolved oxygen from `wtreg`

Index