

Package: aelab (via r-universe)

September 8, 2024

Type Package

Title Calculation of Greenhouse Gas Flux

Version 0.4.0

Maintainer Zhao-Jun Yong <nuannuan0425@gmail.com>

Description Facilitate the analysis of data related to ecology, specifically the establishment of carbon budget. Currently, the package allows the below analysis. (i) the calculation of greenhouse gas flux based on data obtained from trace gas analyzer using the method described in Lin et al. (2024). (ii) the calculation of Dissolved Oxygen (DO) metabolism based on data obtained from dissolved oxygen data logger using the method described in Staehr et al. (2010). Lin et al. (2024) <[doi:10.1016/j.scitotenv.2024.170768](https://doi.org/10.1016/j.scitotenv.2024.170768)>. Staehr et al. (2010) <[doi:10.4319/lom.2010.8.0628](https://doi.org/10.4319/lom.2010.8.0628)>.

Imports tibble, lubridate, stats, dplyr, openxlsx, readxl, ggplot2, readr, tidy

License GPL (>= 3)

Encoding UTF-8

RoxxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

VignetteBuilder knitr

LazyData true

Repository <https://zhao-jun-yong.r-universe.dev>

RemoteUrl <https://github.com/zhao-jun-yong/aelab>

RemoteRef HEAD

RemoteSha d5edacd0ed0d348a4c8895a5777213c93e8e4af3

Contents

calculate_do	2
calculate_regression	3
convert_time	4
hobo	4
n2o	5
plot_hobo	6
process_hobo	6
process_info	7
process_weather	7
tidy_licor	8
Index	9

calculate_do	<i>calculate_do</i>
--------------	---------------------

Description

Calculate the Net Ecosystem Production, Gross Primary Production and Ecosystem respiration based on the change in dissolved oxygen concentration.

Usage

```
calculate_do(df)
```

Arguments

df	Merged dataframe produced by process_hobo(), process_weather() and process_info() functions.
----	--

Value

A dataframe.

Examples

```
data(hobo)
calculate_do(hobo)
```

`calculate_regression` *calculate_regression*

Description

Calculate the slope of greenhouse gas (GHG) concentration change over time using simple linear regression.

Usage

```
calculate_regression(  
  data,  
  ghg,  
  reference_time,  
  duration_minutes = 7,  
  num_rows = 300  
)
```

Arguments

<code>data</code>	Data from the LI-COR Trace Gas Analyzer that has been processed and time-converted.
<code>ghg</code>	Column name of the file containing data on GHG concentration (e.g., "CH4", "N2O").
<code>reference_time</code>	The date and time at which the measurement started.
<code>duration_minutes</code>	The duration of the measurement, default to 7.
<code>num_rows</code>	The number of rows used to perform the regression, default to 300.

Value

A tibble containing the time range (POSIXct format) of the slope and R2 (both numeric) from the simple linear regression.

Examples

```
data(n2o)  
calculate_regression(n2o, "N2O", as.POSIXct("2023-05-04 09:16:15", tz = "UTC"))
```

convert_time	<i>convert_time</i>
--------------	---------------------

Description

Convert the time of the LI-COR Trace Gas Analyzer to match the time in real life.

Usage

```
convert_time(data, day = 0, hr = 0, min = 0, sec = 0)
```

Arguments

data	Data from the LI-COR Trace Gas Analyzer that had been processed by tidy_licor().
day	Day(s) to add or subtract.
hr	Hour(s) to add or subtract.
min	Minute(s) to add or subtract.
sec	Second(s) to add or subtract.

Value

The input data with a new column in POSIXct format converted based on the input value.

Examples

```
data(n2o)
converted_n2o <- convert_time(n2o, min = -10, sec = 5)
```

hobo	<i>Processed data from Onset HOBO Dissolved Oxygen Data Logger. A dataset containing 336 dissolved oxygen concentrations changed over time.</i>
------	---

Description

Processed data from Onset HOBO Dissolved Oxygen Data Logger. A dataset containing 336 dissolved oxygen concentrations changed over time.

Format

A data.frame with 336 rows and 13 variables:

- date_time: Date and time in POSIXct format.
- pressure_hpa: Atmospheric pressure (hpa).
- wind_ms: Wind speed (m/s).
- do: Dissolved oxygen concentrations (mg/L)
- temp: Water temperature (Celsius)
- depth: Water depth (m).
- salinity: Salinity (ppt).
- start_date_time: Start date and time of the deployment.
- end_date_time: End date and time of the deployment.
- sunrise: Sunrise time during that day.
- sunset: Sunset time during that day.
- no_hobo: Name for the data logger .
- site: Name for the site.

Source

own data.

n2o

Processed data from N2O LI-COR Trace Gas Analyzer. A dataset containing 567 N2O concentrations changed over time.

Description

Processed data from N2O LI-COR Trace Gas Analyzer. A dataset containing 567 N2O concentrations changed over time.

Format

A data.frame with 567 rows and 4 variables:

- DATE: Date in character format.
- TIME: Time in character format.
- N2O: Concentrations of nitrous oxide (N2O), in ppb.
- date_time: Date and time in POSIXct format.

Source

own data.

`plot_hobo`*plot_hobo*

Description

Plot the dissolved oxygen concentration over time series grouped by different data loggers to observe the variations.

Usage

```
plot_hobo(df)
```

Arguments

`df` Dataframe produced by `process_hobo()` function.

Value

A plot generated by `ggplot2`.

Examples

```
data(hobo)
plot_hobo(hobo)
```

`process_hobo`*process_hobo*

Description

Tidy the data retrieved from HOB0 U26 Dissolved Oxygen Data Logger.

Usage

```
process_hobo(file_path, no_hobo)
```

Arguments

`file_path` Directory of file.
`no_hobo` The code for the data logger.

Value

A dataframe.

Examples

```
hobo_data_path <- system.file("extdata", "ex_hobo.csv", package = "aelab")
df <- process_hobo(hobo_data_path, "code_for_logger")
```

process_info	<i>process_info</i>
--------------	---------------------

Description

Import and process the necessary information, including the sunrise and sunset times of the day, the date and time range of the deployment, and the code for the data logger.

Usage

```
process_info(file_path)
```

Arguments

file_path Directory of file.

Value

A dataframe.

Examples

```
info_data_path <- system.file("extdata", "info.xlsx", package = "aelab")
df <- process_info(info_data_path)
```

process_weather	<i>convert_time</i>
-----------------	---------------------

Description

Tidy the daily weather data downloaded from weather station in Taiwan.

Usage

```
process_weather(file_path, date, zone)
```

Arguments

file_path Directory of file.
date Date of the daily weather data in yyyy-mm-dd format.
zone Code for the region of the weather station.

Value

A dataframe.

Examples

```
weather_data_path <- system.file("extdata", "ex_weather.csv", package = "aelab")
df <- process_weather(weather_data_path, "2024-01-01", "site_A")
```

tidy_licor	<i>tidy_licor</i>
------------	-------------------

Description

Tidy the data downloaded from GHG Analyzer.

Usage

```
tidy_licor(file_path, gas, analyzer = "licor")
```

Arguments

file_path	Directory of file.
gas	Choose between CO2/CH4 or N2O LI-COR Trace Gas Analyzer, which is "ch4" and "n2o", respectively.
analyzer	The brand of the analyzer which the data was downloaded from.

Value

Return the loaded XLSX file after tidying for further analysis.

Examples

```
ghg_data_path <- system.file("extdata", "ch4.xlsx", package = "aelab")
tidy_licor(ghg_data_path, "ch4")
```


Index

calculate_do, [2](#)
calculate_regression, [3](#)
convert_time, [4](#)

hobo, [4](#)

n2o, [5](#)

plot_hobo, [6](#)
process_hobo, [6](#)
process_info, [7](#)
process_weather, [7](#)

tidy_licor, [8](#)