Package 'aelab'

January 9, 2025

Type Package

Title Data Processing for Aquatic Ecology

Version 1.0.1

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

Description Facilitate the analysis of data related to aquatic 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).
Yong et al. (2024) <doi:10.5194/bg-21-5247-2024>.
Staehr et al. (2010) <doi:10.4319/lom.2010.8.0628>.

Imports tibble, lubridate, stats, dplyr, openxlsx, readxl, ggplot2, readr, tidyr, stringr

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

VignetteBuilder knitr

LazyData true

NeedsCompilation no

Author Zhao-Jun Yong [cre, aut]

Repository CRAN

Date/Publication 2025-01-09 04:10:02 UTC

Contents

`calculate_do` *calculate_do*

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_ghg_flux *calculate_ghg_flux*

Description

Calculate the greenhouse gas (GHG) flux based on input parameters from a data frame.

Usage

```
calculate_ghg_flux(
  data,
  slope = "slope",
  area = "area",
  volume = "volume",
  temp = "temp"
)
```

Arguments

data	A data frame containing relevant data with columns for slope, area, volume, and temperature.
slope	Name of the column in 'data' that contains the slope values of the GHG concentration change (in ppm/s).
area	Name of the column in 'data' that contains the values of the area of the chamber (in square meter).
volume	Name of the column in 'data' that contains values of the volume of the chamber (in litre).
temp	Name of the column in 'data' that contains values of the temperature of the gas (in Celsius).

Value

A list containing the calculated flux and its unit.

Examples

```
data <- data.frame(
  slope = c(1.2, 1.5, 1.1),
  area = c(100, 150, 120),
  volume = c(10, 15, 12),
  temp = c(25, 30, 22)
)
results <- calculate_ghg_flux(data)
print(results)
```

`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

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

combine_hobo *combine_hobo*

Description

Tidy multiple data retrieved from HOBO U26 Dissolved Oxygen Data Logger.

Usage

```
combine_hobo(file_path, file_prefix = "no.")
```

Arguments

file_path	Directory of the folder containing the files.
file_prefix	The prefix before the code for the data logger, defaults to "no."

Value

A dataframe.

Examples

```
hobo_data_path <- system.file("extdata", package = "aelab")
df <- combine_hobo(hobo_data_path, file_prefix = "ex_ho")
```

combine_weather *combine_weather*

Description

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

Usage

```
combine_weather(file_path, start_date, end_date, zone)
```

Arguments

file_path	Directory of folder containing the files (including the character in the file name that precedes the date).
start_date	Date of the daily weather data in yyyy-mm-dd format.
end_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", package = "aelab")
modified_data_path <- paste0(weather_data_path, "/ex_")
df <- combine_weather(modified_data_path,
start_date = "2024-01-01",
end_date = "2024-01-02", "site_A")
```

---

convert_ghg_unit *convert_ghg_unit*

Description

Convert the greenhouse gas (GHG) flux to micromoles per square meter per hour.

Usage

```
convert_ghg_unit(ghg_value, ghg, mass = "µmol", area = "m2", time = "h")
```

Arguments

ghg_value	The value of the flux.
ghg	The molecular formula of greenhouse gases (co2: carbon dioxide; ch4: methane; n2o: nitrous oxide).
mass	The mass component of the input GHG flux, default to micromoles.
area	The area component of the input GHG flux, default to square meter.
time	The time component of the input GHG flux, default to hour.

Value

A numeric value.

Examples

```
convert_ghg_unit(1, ghg = "co2")
```

convert_time *convert_time*

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

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 HOBO 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` *process_info*

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` *convert_time*

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_ghg_analyzer` *tidy_ghg_analyzer*

Description

Tidy the data downloaded from GHG Analyzer.

Usage

```
tidy_ghg_analyzer(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_ghg_analyzer(ghg_data_path, "ch4")
```

Index