Package: licoread (via r-universe)

May 10, 2026

Title Reads Raw Files from Li-COR Gas Analyzers

Version 0.3.0

Date 2025-08-05

Description Reads raw files from Li-COR gas analyzers and produces a dataframe that can directly be used with 'fluxible' (Gaudard et al 2025 <doi:10.1111/2041-210X.70161>).

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports dplyr, jsonlite, lubridate, purrr, readr, rlang, stringr, tibble, tidyr, yaml

Depends R (>= 4.1)

LazyData true

URL https://jogaudard.github.io/licoread/

BugReports https://github.com/jogaudard/licoread/issues

VignetteBuilder knitr

Config/pak/sysreqs libicu-dev libx11-dev

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

Date/Publication 2026-02-09 16:05:02 UTC

RemoteUrl https://github.com/jogaudard/licoread

RemoteRef HEAD

RemoteSha c1d1cd5a563e32f85b49e9f2783bd128066391af

Contents

`data_82z` *to read the raw data*

Description

to read the raw data

Usage

```
data_82z(filepath, data_file, data_name, gases, filename)
```

Arguments

`filepath`	name and path to the 82z archive
`data_file`	name of the file with raw data
`data_name`	vector of colnames
`gases`	list of gases
`filename`	name of the 82z archive

Value

a long df with the actual data contained in the data file

data_name_82z *create colnames for data tibble*

Description

create colnames for data tibble

Usage

```
data_name_82z(filepath, data_file)
```

Arguments

filepath	name and path to the 82z archive
data_file	name of the file with raw data

Value

a character vector or the colnames of the data file

fluxible_81x *makes df from 81x files compatible with fluxible*

Description

makes df from 81x files compatible with fluxible

Usage

```
fluxible_81x(df, focus_gas, id_cols, datetime_col)
```

Arguments

df	input dataframe from licoread
focus_gas	gas to select
id_cols	columns to identify unique fluxes
datetime_col	column containing datetime information

Value

a df with the focus gas column renamed as "f_conc" and f_fluxid in chronological order of datetime

fluxible_82z *makes df from 82z files compatible with fluxible*

Description

makes df from 82z files compatible with fluxible

Usage

```
fluxible_82z(df, focus_gas)
```

Arguments

df	input dataframe from licoread
focus_gas	gas to select

Value

an unnested df with only the selected gas

import7500 *imports data from li7500 setup*

Description

imports data from the li7500 setup described in Halbritter et al (2024)

Usage

```
import7500(path, version = "till2023", plotinfo = FALSE)
```

Arguments

path	location of the files. Note that the function is recursive and will import all the files at the location
version	"till2023" (default) refers to the setup before the li7500 update. For the update version, use "post2023".
plotinfo	character vector of names to use when separating the plot info from the file name. Names in the file name have to be separated with an underscore, ex: [location]_[date]_[time of day]_[trial].txt. FALSE (default) means no information will be fetched from the filename.

Value

a dataframe with all data present in the files to import. New datetime and fluxid columns contain respectively the datetime of measurements and their original filename. If comments are present, a column will contain the comments. Measurement informations from the file names are stored in new columns according to `plotinfo`.

References

Halbritter, A.H., Vandvik, V., Cotner, S.H. et al. Plant trait and vegetation data along a 1314 m elevation gradient with fire history in Puna grasslands, Perú. Sci Data 11, 225 (2024). https://doi.org/10.1038/s41597-024-02980-3

Examples

```
path_pftc7 <- system.file("extdata/pftc7", package = "licoread")
import7500(path_pftc7, version = "post2023")

path_pftc5 <- system.file("extdata/pftc5", package = "licoread")
import7500(path_pftc5,
plotinfo = c("site", "treatment", "date", "plot", "trial"))

path_co2fluxtent <- system.file("extdata/co2fluxtent",
package = "licoread")
import7500(path_co2fluxtent,
plotinfo = c("date", "location", "time_of_day", "trial"))
```

`import7500_new_oneobs` *imports one file from li7500 (old)*

Description

imports one file from li7500 (old)

Usage

```
import7500_new_oneobs(filepath)
```

Arguments

`filepath`	path to the file

Value

a df with the content of the file

import7500_old_oneobs *imports one file from li7500 (old)*

Description

imports one file from li7500 (old)

Usage

```
import7500_old_oneobs(filepath)
```

Arguments

filepath	path to the file

Value

a df with the content of the file

licoread *reads Li-COR files in a given location*

Description

reads Li-COR files in a given location

Usage

```
licoread(
  location,
  file_type = "auto",
  file_type_list = c("82z", "81x", "auto"),
  data_file = "data.csv",
  meta_file = "metadata.json",
  regex_file = "(\\w*-)*\\w*(?=([.]82z$))",
  sample = FALSE
)
```

Arguments

location	location of the files
file_type	type of file (82z or 81x). If "auto" (default), the function will try to detect it by itself.
file_type_list	list of file types
data_file	name of the file with raw data

meta_file	name of the file with meta data
regex_file	regex expression matching the name of the 82z file. Here in case the user has a different than the default and for easier updates.
sample	sample = n randomly selects n files to be imported. This allows for testing the setup before importing a potentially large list of files which will take time and be difficult to handle.

Value

a tibble (nested or not depending on raw data) containing all the data from the raw files present at the location provided

Examples

```
path_82z <- system.file("extdata/82z", package = "licoread")
licoread(path_82z)
```

---

licoread_auto *finds out the file type for licoread*

Description

finds out the file type for licoread

Usage

```
licoread_auto(file_list)
```

Arguments

file_list	list of files found in the location

Value

a single character string indicating the file type

`licoread_to_fluxible` *selects the focus gas and makes a df to use in fluxible*

Description

selects the focus gas and checks the columns needed for the fluxible workflow

Usage

```
licoread_to_fluxible(
  df,
  focus_gas,
  datetime_col,
  id_cols = c("File Name", "Obs#"),
  file_type = "auto",
  file_type_list = c("82z", "81x", "auto")
)
```

Arguments

`df`	input dataframe from licoread
`focus_gas`	gas to select
`datetime_col`	column containing datetime information if date and time are in two different columns, provide a character vector of the form c("date", "time")
`id_cols`	columns to identify unique fluxes
`file_type`	type of file (82z or 81x). If "auto" (default), the function will try to detect it by itself.
`file_type_list`	list of file types

Value

an unnested df with only the selected gas

Examples

```
path_82z <- system.file("extdata/82z", package = "licoread")
gas_df_82z <- licoread(path_82z)
licoread_to_fluxible(gas_df_82z, "LI-7810_CH4_DRY",
datetime_col = c("LI-8250_DATE", "LI-8250_TIME"))
```

list_gases *lists gases present in the dataset*

Description

prints a character vector of the gases present in the dataset. This is to be used to obtain the exact names of the gases before using licoread_to_fluxible.

Usage

```
list_gases(df, file_type = "auto", file_type_list = c("82z", "81x", "auto"))
```

Arguments

df	the dataframe as imported with licoread
file_type	type of file (currently works only for 82z). If "auto" (default), the function will try to detect it by itself.
file_type_list	list of file types

Value

a character vector of the gases present in the dataset

Examples

```
path_82z <- system.file("extdata/82z", package = "licoread")
gas_df_82z <- licoread(path_82z)
list_gases(gas_df_82z)
```

metadata_82z *read meta data file inside 82z archive*

Description

read meta data file inside 82z archive

Usage

```
metadata_82z(filepath, meta_file)
```

Arguments

filepath	name and path to the 82z archive
meta_file	name of the file with meta data

Value

a tibble with the metadata from one observation

names_df *to get a vector of names of a df, matching a regex*

Description

to get a vector of names of a df, matching a regex

Usage

```
names_df(df, regname = "name\\d")
```

Arguments

df	the df to get the names from
regname	the regex expression to match

Value

a df with the names of the meta df

oneobs_81x *reading a single measurement from 81x file*

Description

reads a single measurement from a licor .81x file

Usage

```
oneobs_81x(start, end, all_obs, file)
```

Arguments

start	line number at which the measurement starts
end	line number at which the measurement ends
all_obs	list of all the lines from the full file
file	filepath to the 81x files

Value

a df with 1 row with the meta data of the measurement and raw data nested

oneobs_82z *to read one measurement from the 82z archive*

Description

to read one measurement from the 82z archive

Usage

```
oneobs_82z(filepath, data_file, meta_file, regex_file)
```

Arguments

filepath	path to the 82z archive
data_file	name of the file with raw data
meta_file	name of the file with meta data
regex_file	regex expression matching the name of the 82z file. Here in case the user has a different than the default and for easier updates.

Value

a tibble with all the data and metadata from one observation (one file)

read_81x_onefile *reads 81x licor file*

Description

reads a .81x file with several measurements

Usage

```
read_81x_onefile(file)
```

Arguments

file	filepath the the 81x file to read

Value

a nested tibble with the meta data from each measurements as row and the raw data nested

`units_82z` *to create a nested tibble with the units of data*

Description

to create a nested tibble with the units of data

Usage

```
units_82z(filepath, data_file, data_name, filename)
```

Arguments

`filepath`	name and path to the 82z archive
`data_file`	name of the file with raw data
`data_name`	vector of colnames
`filename`	name of the 82z archive

Value

a tibble with the units of the variables contained in the raw data

Index