



Fluxible: an R package to process ecosystem gas fluxes

Joseph Gaudard (joseph.gaudard@pm.me), Jonas Trepel, Hilary Rose Dawson, Brian Enquist, Aud H Halbritter, Michael Mustri, Pekka Niittynen, Paul Efren Santos-Andrade, Joachim P Topper, Vigdis Vandvik, and Richard J Telford



 R-CMD-check.yaml passing

CRAN 1.1.0

lifecycle stable

```
install.packages\("fluxible"\)
```



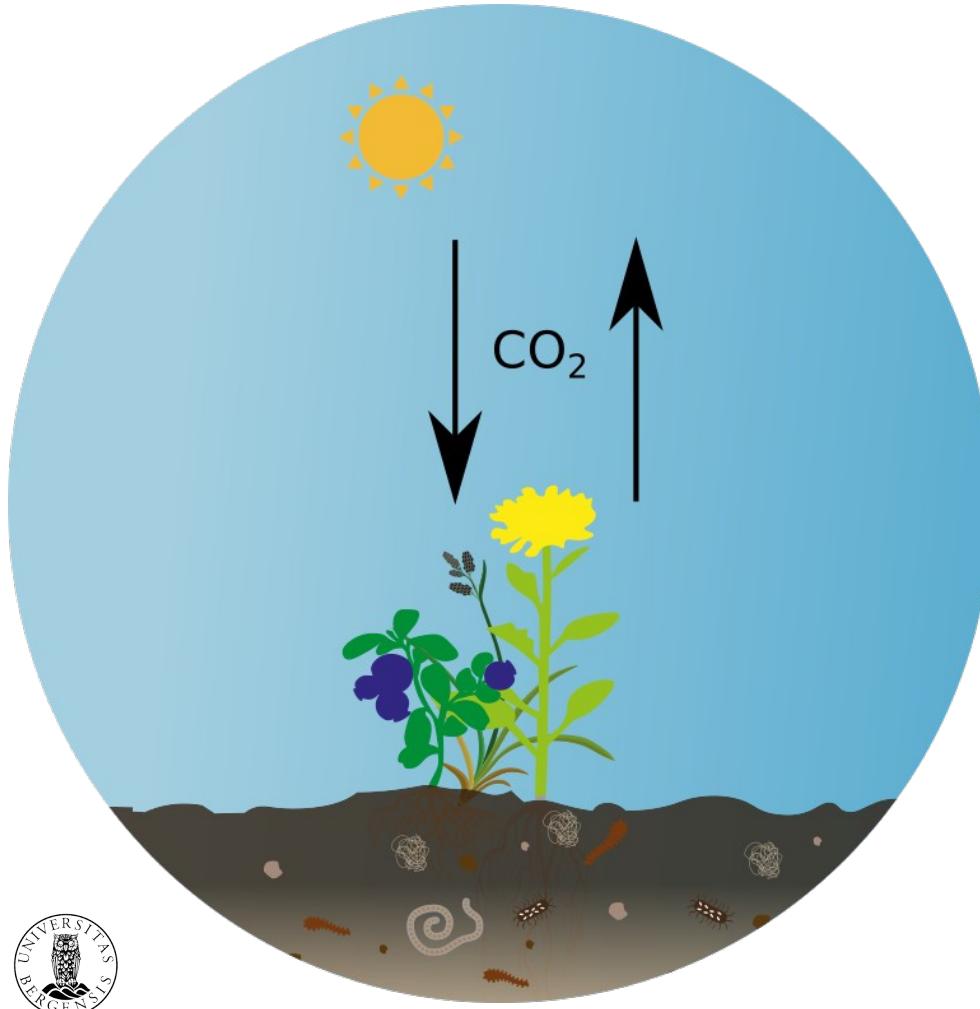
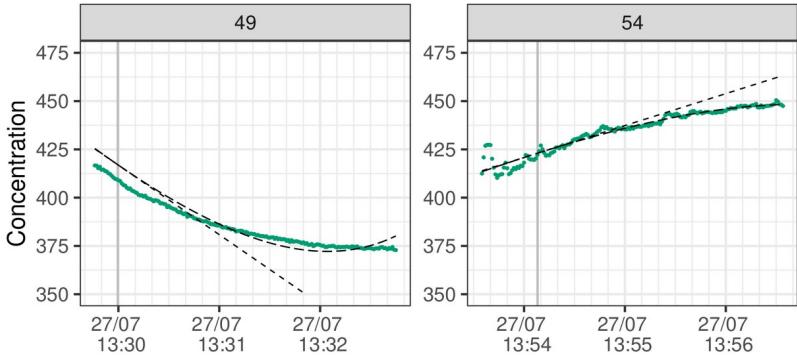




Photo: Aud H Halvorsen



$$\text{flux} = \text{slope} \times \frac{P \times V}{R \times T \times A}$$

Data collection



Fit a model to the data
Extract the slope



Calculate fluxes

Fluxible R package



Fluxible R package

- Reproducible, flexible, and automated workflow



Fluxible R package

- Reproducible, flexible, and automated workflow
- Homogenous data processing
 - Different models produce different bias (Johannesson, 2024; Forbrich, 2010)
 - Length of the observation window strongly influence final flux value (Johannesson, 2024)



Fluxible R package

- Reproducible, flexible, and automated workflow
- Homogenous data processing
 - Different models produce different bias (Johannesson, 2024; Forbrich, 2010)
 - Length of the observation window strongly influence final flux value (Johannesson, 2024)
- Toolbox approach
 - Choice of models
 - User set their own quality thresholds

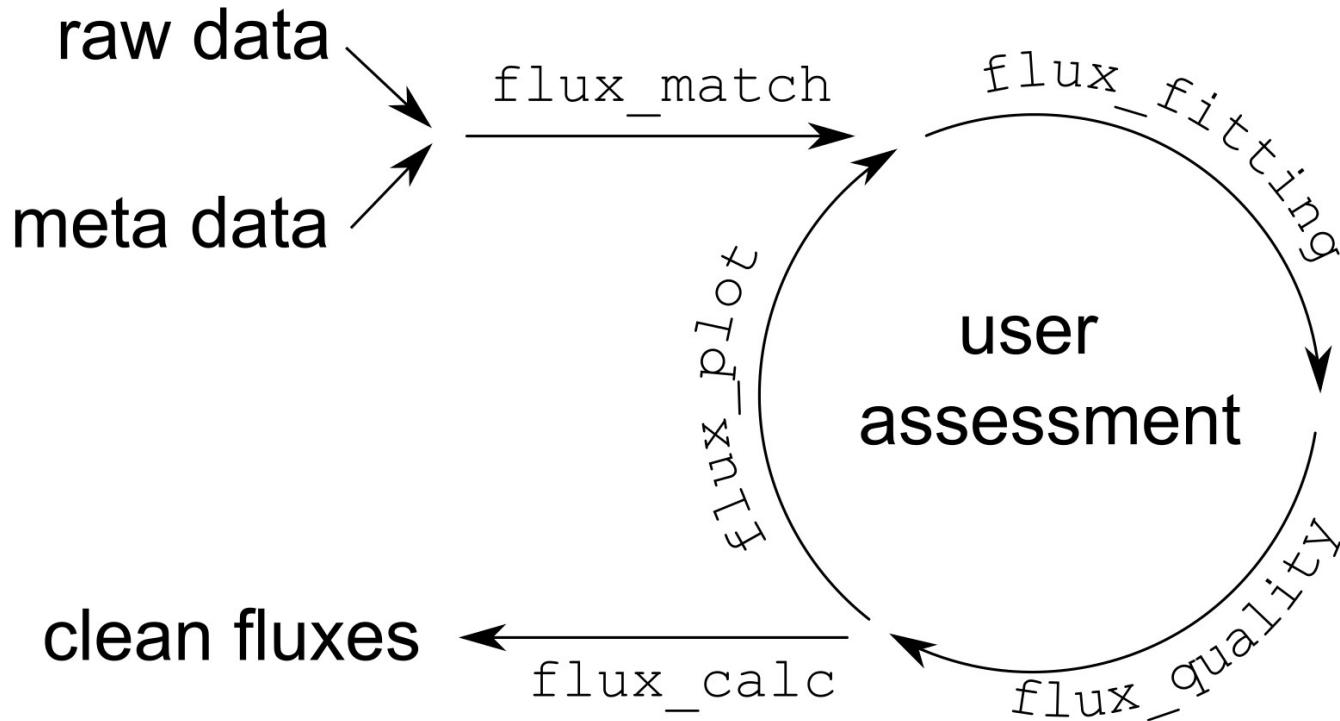


Fluxible R package

- Reproducible, flexible, and automated workflow
- Homogenous data processing
 - Different models produce different bias (Johannesson, 2024; Forbrich, 2010)
 - Length of the observation window strongly influence final flux value (Johannesson, 2024)
- Toolbox approach
 - Choice of models
 - User set their own quality thresholds
- Step by step workflow
 - Cleaning other variables simultaneously
 - Allows for targeted investigations of outliers



Fluxible workflow



Fitting a model to the data

```
flux_fitting(conc_df, conc_col,  
            datetime, fit_type = "exp_tz",  
            t_zero = 10, start_cut = 10)
```

fit_type	model
linear	$C(t) = a * t + b$
quadratic	$C(t) = a * t^2 + b * t + c$
exp_zhao18 (Zhao <i>et al.</i> , 2018)	$C(t) = C_m + a * (t - t_z) + (C_z - C_m) * \exp(-b * (t - t_z))$
exp_tz	$C(t) = C_m + a * t + (C_z - C_m) * \exp(-b * t)$
exp_hm (Hutchinson and Mosier, 1981)	$C(t) = C_m + (C_z - C_m) * \exp(-b * t)$

Quality assessment

- Columns added:
 - quality flags
 - corrected slopes
- Provides a summary of the dataset quality
- User defined diagnostic thresholds
- Kappamax method (Hüppi et al., 2018)
 - Uses linear slopes when $|b| > |\frac{f_slope_lm}{instr_error}|$
 - Can be applied to any exponential model

```
flux_quality(slopes_df, conc_col,  
             rsquared_threshold = 0.7,  
             rmse_threshold = 25,  
             force_discard = c(45, 56))
```

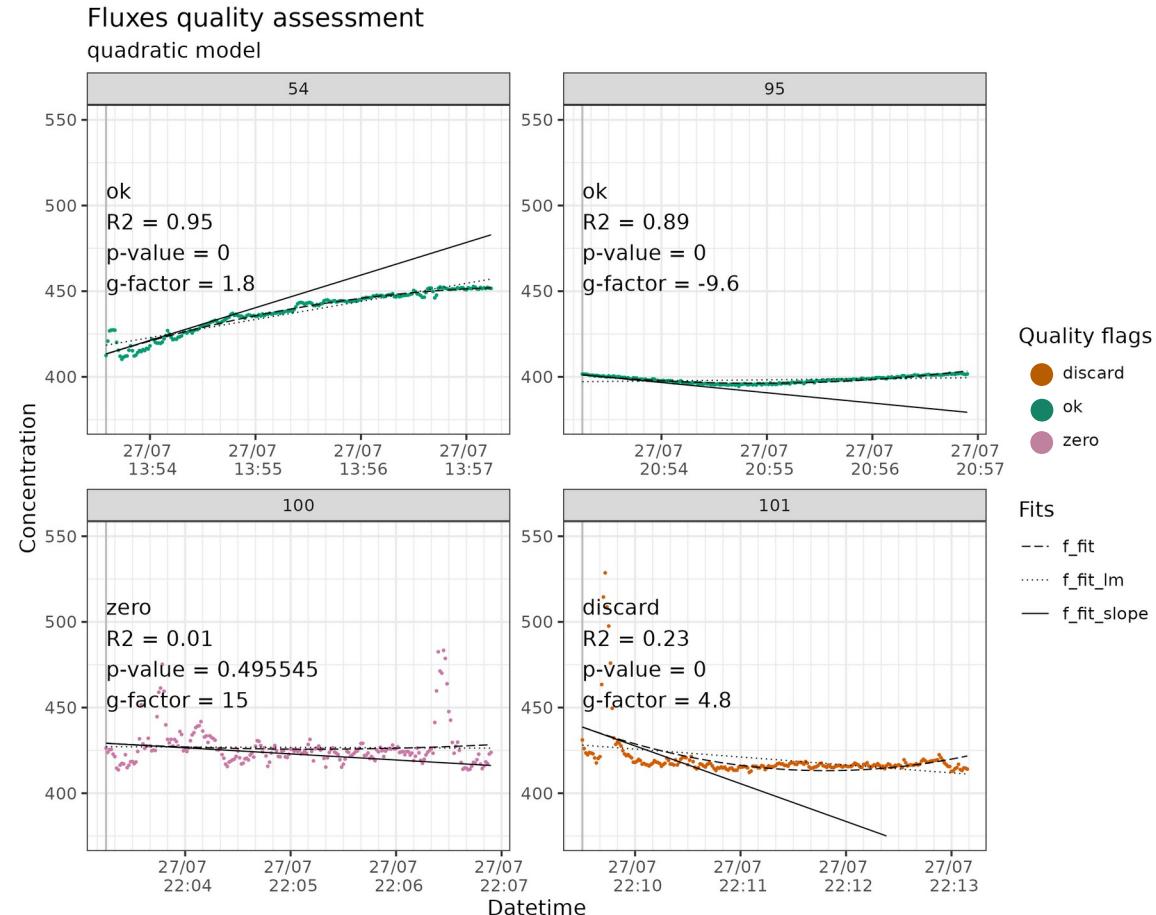
```
Total number of measurements: 138  
  
ok      120      87 %  
zero      14      10 %  
discard      4      3 %  
force_discard      0      0 %  
start_error      0      0 %  
no_data      0      0 %  
force_ok      0      0 %  
force_zero      0      0 %  
force_lm      0      0 %
```



Visual assessment

```
flux_plot(slopes_df,
          conc_col, datetime,
          output = "pdfpages")
```

- From flux_quality
 - quality flags
 - diagnostics
- From flux_fitting
 - fit
 - slope
 - linear fit



Flux calculation

$$\text{flux} = \text{slope} \times \frac{P \times V}{R \times T \times A}$$

P: pressure (atm)

V: volume (L)

R: gas constant (0.082057

L*atm*K⁻¹*mol⁻¹)

T: temperature (K)

A: plot area (m²)

```
flux_calc(slopes_df, slope_col, f_datetime,
          temp_air_col, chamber_volume, atm_pressure,
          plot_area = 0.25, conc_unit = "ppm",
          flux_unit = "mmol", cols_ave = "PAR",
          cols_keep = c("site", "plotID"))
```

- Area, pressure, volume can be constants or variables
- Processes other variables present in the dataset (sum, mean or median)
- Unit conversion
 - temperature
 - concentration
- Output in mmol m⁻² h⁻¹ or µmol m⁻² h⁻¹

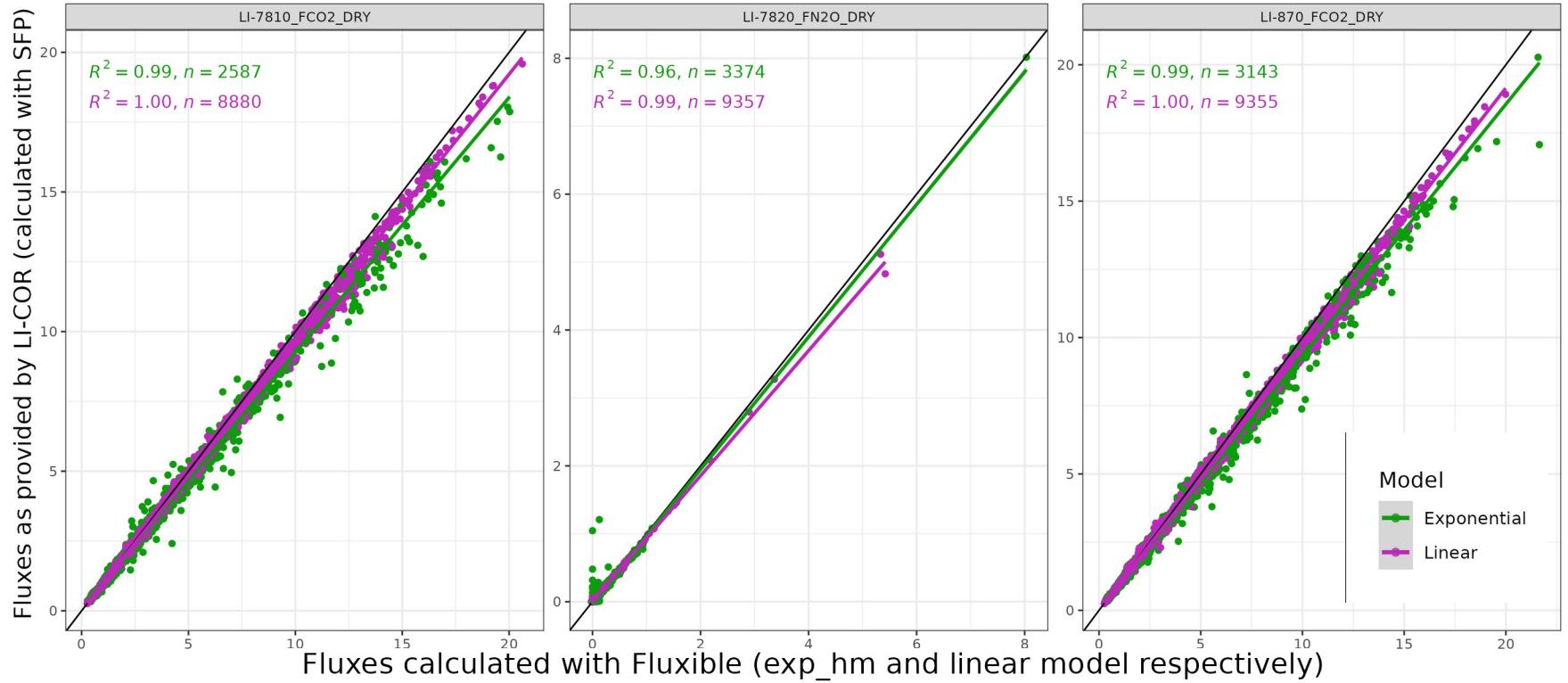


Supporting infrastructure

- Licoread R package (LI-COR collaboration; on CRAN)
 - Imports LI-COR raw files (81x and 82z)
 - Produces fluxible-friendly dataframes from LI-COR raw files
- Segmentation tool (in dev)
 - Defines the observation window based on an environmental variable
- Common calculations around ecosystem gas fluxes
 - `flux_gpp`: gross primary production
 - coming: evapotranspiration, carbon usage effectiveness, water use efficiency, light response curves



Comparison with SoilFluxPro output



Thank you for your attention



```
install.packages\("fluxible"\)
```

plant-functional-trait-
course.github.io/fluxible

✉️ joseph.gaudard@pm.me
🦋 [@jogaudard.bsky.social](https://bsky.social/@jogaudard)
>ID 0000-0002-6989-7624



Australian
National
University



AARHUS
UNIVERSITY



Presentation supported by the "Climate and Energy
Transition Travel Support" from the University of Bergen

