The Journal of Open Source Software

Fluxtools: An R Package with an Interactive Shiny App for Reproducible QA/QC of Eddy Covariance Data Aligned with AmeriFlux Standards

Kesondra Key [1]

1 Paul H. O'Neill School of Public and Environmental Affairs, Indiana University, Bloomington, Indiana, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Eddy covariance data processing requires extensive quality control (QA/QC) to identify and remove implausible or erroneous half-hourly flux data before submission to public data repositories such as AmeriFlux (AmeriFlux Management Project 2025). *Fluxtools* (Key 2025) is an R (4.5.0; (2025)) *Shiny* (Chang et al. 2024) application built with *Plotly* (Sievert et al. 2024) and *dplyr* (Wickham et al. 2023) packages designed to streamline this workflow by providing interactive visualization, year-based filtering, and on-the-fly R code generation for specified data removal. Users can visually flag anomalous data points (i.e., periods of sensor failure, physically implausible data), accumulate multiple cleaning steps, inspect R^2 values before and after data cleaning via base R's *lm()* function, and export a zipped folder containing a cleaned .csv file and a full R script that records every decision. *Fluxtools* significantly accelerates the QA/QC workflow, ensuring transparent, reproducible, and shareable data cleaning suitable for final dataset preparation and repository submission.

Fluxtools hex logo

Key features:

- **Interactive Plotly Scatterplots**: Plot any numeric or time variable; hover mouse over data points to see timestamps and values; export plots as .png directly from the app
- **Flexible point selection**: Box-select, lasso, or apply standard-deviation (σ) cutoffs to mark selected points. *Fluxtools* automatically generates removal ready-to-copy R snippets (*dplyr::case_when(... ~ NA)*) in the *Current* code pane. *See Fig 2* for interface and data selection example
- **On-the-fly R code generation**: After point selection, clicking *Flag Data* automatically highlights chosen points in yellow and appends the corresponding removal code into the *Accumulated* code panel for easy and continuous data selection

Example of the Fluxtools interface and data selection

- **Before/after R^2 diagnostics**: For any numeric variable comparison, *Fluxtools* fits a linear regression model and reports its R^2 value. Selecting points re-computes R^2 as if those points were removed, allowing for easy comparison. *Fig 3* shows this process in *Fluxtools* using the *$\pm\sigma$ outliers selection tool:* The top (red) R^2 uses all data, while the bottom R^2 (orange) omits selected points from the linear regression

Example of R^2 diagnostics using the $\pm\sigma$ outliers cutoffs and selection

The Journal of Open Source Software

- **Export cleaned .csv file and R script**: *Apply removals* in-app (converting data points into *NA*s for selected timestamps) then *Export cleaned data* to download a cleaned .csv file and a comprehensive R script documenting each data removal step

Statement of need

High-frequency (10Hz; data recorded 10x per second) eddy covariance measurements generate large datasets that must be aggregated into half-hourly fluxes, using careful QA/QC (Burba 2021). At this high temporal resolution, intermittent periods of sensor drift or failure are common, making manual data cleaning an integral part of the workflow.

Tools like *Loggernet's CardConvert* feature (Campbell Scientific, Inc. 2025) convert raw 10Hz eddy covariance data into half-hourly intervals, preparing them for flux estimation. Software like *EddyPro* (LI-COR Biosciences 2021) then computes turbulent fluxes of CO , H O, and energy using these half-hourly inputs. Post-processing *R* packages like *REddyProc* (Wutzler et al. 2024), and Python tools like PyFluxPro (Isaac 2021), automate u*-threshold filtering, gap-filling, and flux partitioning. These tools excel at bulk data processing but offer no interactive means to inspect or carefully remove outliers that require a human eye.

In practice, data managers resort to custom scripts, manual visualization, and fragmented documentation to detect and remove erroneous data points caused by sensor drift, malfunction, or calibration issues. These procedures are labor-intensive, error-prone, difficult to reproduce, and lack transparency. *Fluxtools* addresses these challenges by pairing an interactive scatterplot-based interface with on-the-fly R code generation. Users can visually flag implausible half-hourly data points, automatically generate or directly apply the exact *case_when(... ~ NA) dplyr* code snippets and export a .zip file containing a cleaned .csv file plus a comprehensive R script documenting each QA/QC decision.

By combining interactivity with code-based reproducibility, *Fluxtools* streamlines QA/QC workflows by enhancing transparency, reducing manual effort, and accelerating the preparation of flux data for repository submissions (e.g., Ameriflux). Ultimately, *Fluxtools* lowers the barriers to robust and reproducible QA/QC, allowing data managers and scientists to spend less time on manual anomaly detection and more on scientific analysis and interpretation.

Code Example

Fluxtools can be installed from Github.

Installation

```
# Option 1.) Install from CRAN (Fluxtools R Package submission pending)
#install.packages("fluxtools")

# Option 2.) Install most recent version from GitHub
library(remotes)
remotes::install_git("https://github.com/kesondrakey/fluxtools.git")
```

Call the Fluxtools App

```
library(fluxtools)

#Set your site's UTC offset (e.g., -5 for Eastern Standard Time)
fluxtools::run_flux_qaqc(-5)
```

The Journal of Open Source Software

Acknowledgments

Fluxtools is an independent project developed in my role as Co-Principal Investigator for the AmeriFlux US-VT1 (Key and Novick 2025a) and US-VT2 (Key and Novick 2025b) sites to streamline our QA/QC pipeline. I thank AmeriFlux for maintaining clear, up-to-date data-formatting specifications (see AmeriFlux Data Variables) and I acknowledge the broader *R* and *Shiny* communities for their efforts in advancing interactive data-visualization tools.

I am grateful to Dr. Kim Novick (Indiana University), Housen Chu (AmeriFlux), and Benju Baniya (Texas A&M University) for their invaluable expertise in assisting with the AmeriFlux data processing workflow. Without their support, this data pipeline would not have been possible. I also thank Dr. Mallory Barnes, Dr. Daniel Beverly, and Dr. Xian Wang (all at Indiana University) for their support during development and testing. Special thanks goes to George Burba (LI-COR Biosciences) for his encouragement and for insightful discussions that have helped shape the future of *Fluxtools*.

This work was supported by NSF's Division of Environmental Biology (DEB) Dynamics of Socio-Environmental Systems Program (Award 22-06086) and NASA's Carbon Monitoring System (Award 80NSSC23K1254).

Partial development of the Fluxtools application logic, debugging assistance, and initial code drafting were supported by OpenAI's *GPT-4-mini-high* large language model (OpenAI 2025). All final code and revisions were authored and approved by the human author.

References

AmeriFlux Management Project. 2025. "AmeriFlux Data Portal." 2025. https://ameriflux.lbl.gov/.

Burba, George. 2021. *Eddy Covariance Method - for Scientific, Regulatory, and Commercial Applications*. Lincoln, Nebraska: LI-COR Biosciences. https://www.licor.com/products/eddy-covariance/ec-book.

Campbell Scientific, Inc. 2025. "LoggerNet: Datalogger Support Software." https://www.campbellsci.com/loggernet.

Chang, Winston, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert, and Barbara Borges. 2024. *Shiny: Web Application Framework for r*. https://doi.org/10.32614/CRAN.package.shiny.

Isaac, Peter. 2021. "PyFluxPro." https://github.com/OzFlux/PyFluxPro.

Key, Kesondra. 2025. "Fluxtools: Interactive QA/QC for Eddy Covariance Data." Zenodo. https://doi.org/10.5281/zenodo.15626645.

Key, Kesondra, and Kim Novick. 2025a. "AmeriFlux BASE US-VT1 Vermillion Tributary Paired Cropland – Site 1 (Corn/Soy; No Cover Crops)." AmeriFlux Network, Lawrence Berkeley National Laboratory. https://doi.org/10.17190/AMF/2567994.

———. 2025b. "AmeriFlux BASE US-VT2 Vermillion Tributary Paired Cropland – Site 2 (Corn/Soy; Cover Crops)." AmeriFlux Network, Lawrence Berkeley National Laboratory. https://doi.org/10.17190/AMF/2567995.

LI-COR Biosciences. 2021. "EddyCovariance Processing Software (EddyPro)." https://www.licor.com/env/products/eddy-covariance/eddypro.

OpenAI. 2025. "Introducing OpenAI O3 and O4-Mini." 2025. https://openai.com/index/introducing-o3-and-o4-mini/.

R Core Team. 2025. *R: A Language and Environment for Statistical Computing*. Vienna,

Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Sievert, Carson, Chris Parmer, Toby Hocking, Scott Chamberlain, Karthik Ram, Marianne Corvellec, Pedro Despouy, Salim Brüggemann, and Plotly Technologies Inc. 2024. "Plotly: Create Interactive Web Graphics via 'Plotly.js'." https://doi.org/10.32614/CRAN.package.plotly.

Wickham, Hadley, Romain François, Lionel Henry, Kirill Müller, and Davis Vaughan. 2023. *Dplyr: A Grammar of Data Manipulation*. https://doi.org/10.32614/CRAN.package.dplyr.

Wutzler, Thomas, Markus Reichstein, Antje Maria Moffat, and Mirco Migliavacca. 2024. *REddyProc: Post Processing of (Half-)hourly Eddy-Covariance Measurements*. https://doi.org/10.32614/CRAN.package.REddyProc.