The Journal of Open Source Software

Fluxtools: interactive Shiny tool for QA/QC and code generation of AmeriFlux eddy covariance data

Kesondra Key [1]

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

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Published: unpublished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; R Core Team (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.

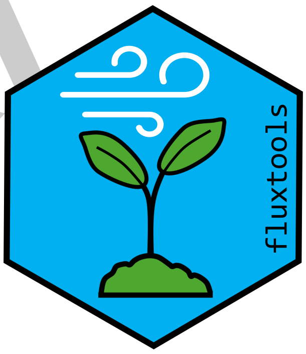

Figure 1: *Fluxtools* hex logo

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**: Select data points via box, lasso, or by standard-deviation (σ) cutoffs. *See Fig 2* for interface and data selection example

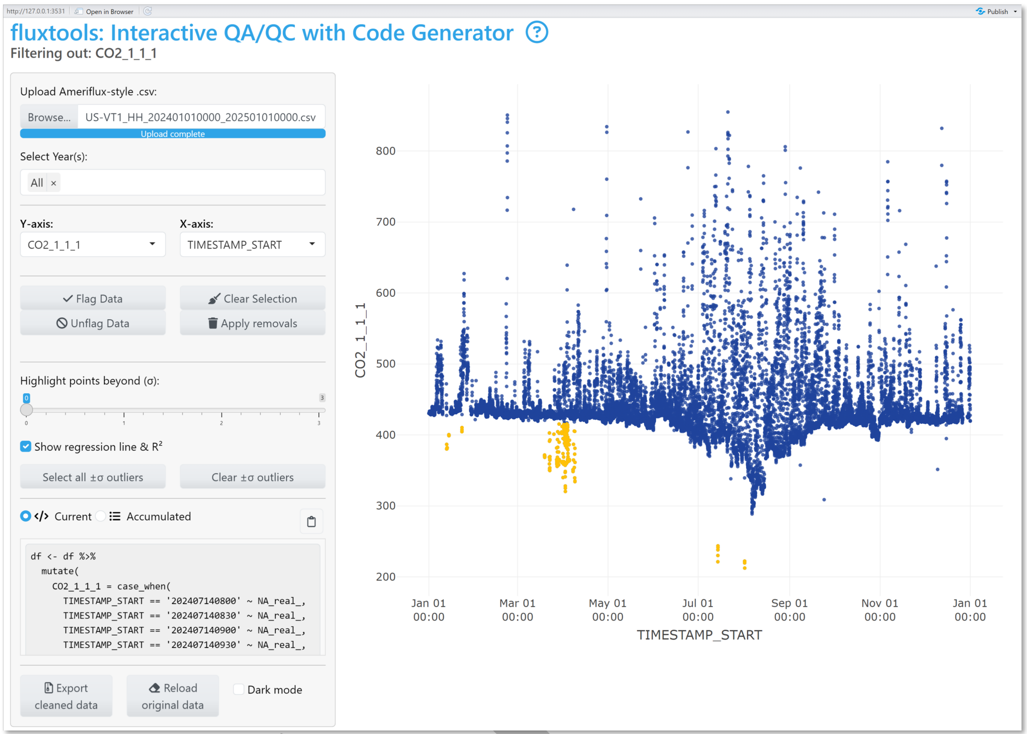

Figure 2: Example of the *Fluxtools* interface and data selection

- **On-the-fly R code generation**: The *Preview* pane shows selected timestamps and values; ready-to-copy R code using *dplyr's case_when(… ~ NA)* snippets generate in the *current* code box automatically; *add current selection* adds code to the *accumulated* code box for easy and continuous data selection
- **Before/after R^2 diagnostics**: When numeric variables are compared against each other, a linear regression generates a R^2 value. Automatically computes post-removal R^2 value where selected data points are dropped to see step comparisons. *See Fig 3* for an example of the *Fluxtools* interfacing using the $\pm\sigma$ outliers selection tool. The top (red) R^2 is for all data points and the bottom R^2 (orange) is when selected points are dropped from the linear regression model

The Journal of Open Source Software

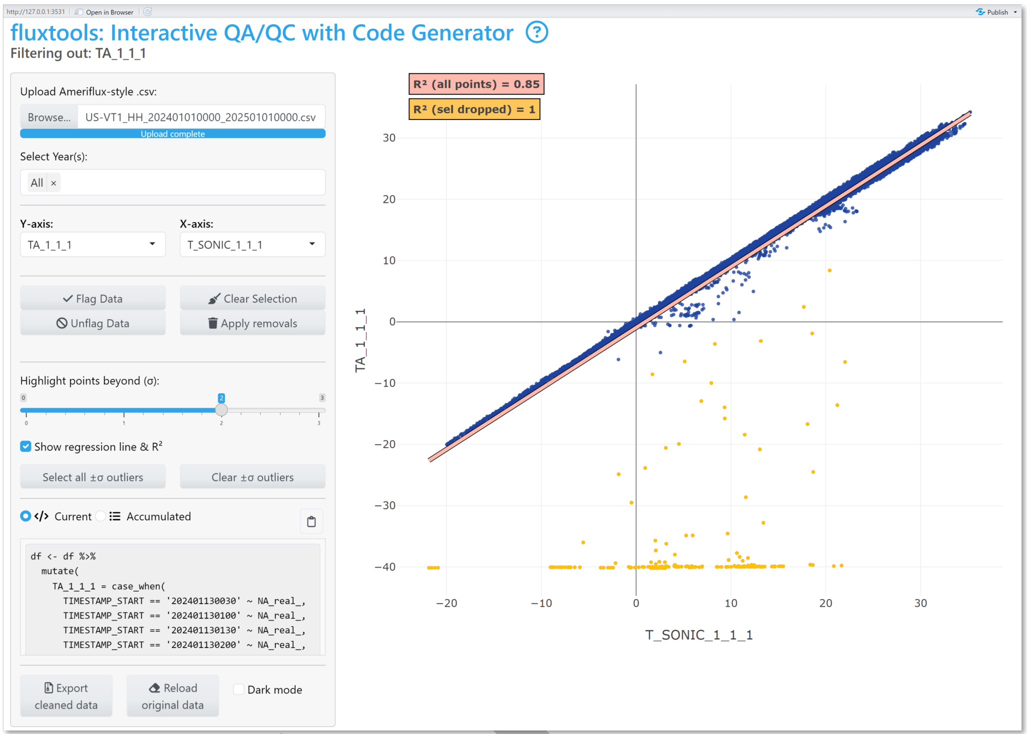

Figure 3: Example of R² diagnostics using the $\pm\sigma$ outliers cutoffs and selection

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

Statement of need

High-frequency (10 Hz; data recorded 10 times per second) eddy covariance measurements generate large datasets that must be carefully aggregated into half-hourly fluxes, using careful quality assurance and quality control (Burba, 2021). At this high frequency, intermittent periods of sensor drift or failure are common, making manual data cleaning an integral part of the workflow. Tools like *EddyPro* (LI-COR Biosciences, 2021) converts raw 10Hz data into half-hourly fluxes, while 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 that require a human eye.

In practice, data managers resort to custom scripts, extensive 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, prone to errors, challenging to reproduce, and lack transparency. *Fluxtools* addresses this challenge 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 the exact *case_when(… ~ NA) dplyr* code snippets (or apply removes automatically in the app) and export a .zip file containing a cleaned .csv file plus a comprehensive R script documenting each data removal step that captures every user-made QA/QC decision.

Fluxtools streamlines and clarifies the QA/QC workflow by combining interactivity with code-based reproducibility. It promotes transparent documentation of decisions, reduces manual effort, and accelerates the preparation of flux data for repository uploads such as individual site submissions to AmeriFlux. Ultimately, *Fluxtools* lowers the barriers to robust and reproducible QA/QC workflows, enabling researchers to devote less time to manual anomaly detection and more time to scientific analysis.

Code Example

Fluxtools can be installed from Github.

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


Acknowledgments

Fluxtools was developed in my role as Co-Principal Investigator for the AmeriFlux US-VT1 (Key & Novick, 2025a) and US-VT2 (Key & 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 acknowlege 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*. https://ameriflux.lbl.gov/

Burba, G. (2021). *Eddy covariance method - for scientific, regulatory, and commercial applications* (p. 702). LI-COR Biosciences. ISBN: 978-0-578-97714-0

Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., & Borges, B. (2024). *Shiny: Web application framework for r*. https://doi.org/10.32614/CRAN.package.shiny

Isaac, P. (2021). *PyFluxPro*. https://github.com/OzFlux/PyFluxPro

Key, K. (2025). *Fluxtools: Interactive QA/QC for AmeriFlux data* (Version 0.2.0). Zenodo. https://doi.org/10.5281/zenodo.15626645

Key, K., & Novick, K. (2025a). *AmeriFlux BASE US-VT1 vermillion tributary paired cropland – site 1 (corn/soy; no cover crops)* (1–5 (BASE, CC-BY-4.0)) [Data set]. AmeriFlux Network, Lawrence Berkeley National Laboratory. https://doi.org/10.17190/AMF/2567994

Key, K., & Novick, K. (2025b). *AmeriFlux BASE US-VT2 vermillion tributary paired cropland – site 2 (corn/soy; cover crops)* (1–5 (BASE, CC-BY-4.0)) [Data set]. AmeriFlux Network, Lawrence Berkeley National Laboratory. https://doi.org/10.17190/AMF/2567995

LI-COR Biosciences. (2021). *EddyCovariance Processing Software (EddyPro)* (Version 7.0.9). https://www.licor.com/env/products/eddy-covariance/eddypro

OpenAI. (2025). *Introducing OpenAI o3 and o4-mini*. https://openai.com/index/introducing-o3-and-o4-mini/

R Core Team. (2025). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M., Despouy, P., Brüggemann, S., & Inc, P. T. (2024). *Plotly: Create interactive web graphics via 'plotly.js'*. https://doi.org/10.32614/CRAN.package.plotly

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *Dplyr: A grammar of data manipulation*. https://doi.org/10.32614/CRAN.package.dplyr

Wutzler, T., Reichstein, M., Moffat, A. M., & Migliavacca, M. (2024). *REddyProc: Post processing of (half-)hourly eddy-covariance measurements*. https://doi.org/10.32614/CRAN.package.REddyProc