fluxtools: interactive Shiny tool for QA/QC and code generation of Ameriflux eddy-covariance data

2025-06-17

Summary

Eddy covariance data processing (Burba 2021) requires extensive quality control QA/QC to identify and remove implausible or erroneous half-hourly flux data before submission to public data repositories (i.e., Ameriflux) . Fluxtools (Key 2025) is an R (4.5.0) Shiny application (required packages: *Shiny* (Chang et al. 2024), *Plotly* (Sievert et al. 2024), *dplyr* (Wickham et al. 2023)) specifically designed to streamline this QA/QC process by providing interactive visualization, year-based filtering, and on-the-fly R code generation. This allows data managers the ability to visually pinpoint "bad" data (i.e., sensors went down, physically implausible data) and immediately extract reproducible removal snippets. With a few clicks, you can accumulate multiple removal steps, inspect before/after R2 values via base R's *lm()*, and export both a cleaned .csv file and a full R script for data removal code. *Fluxtools* significantly accelerates the QA/QC workflow, ensuring transparent, reproducible, and shareable data cleaning suitable for final dataset preparation and repository submission (i.e., Ameriflux sites like US-VT1 (AmeriFlux 2025a) and US-VT2 (AmeriFlux 2025b)).

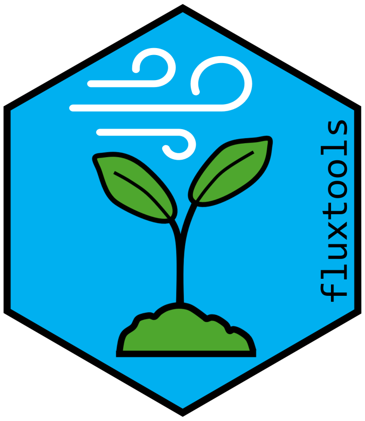


Figure 1: *Fluxtools* hex logo

Key features:

Interactive Plotly Scatterplots: Plot any numeric or time variable; hover mouse over individual datapoints to see timestamps and point values; export plots as .png directly from the app

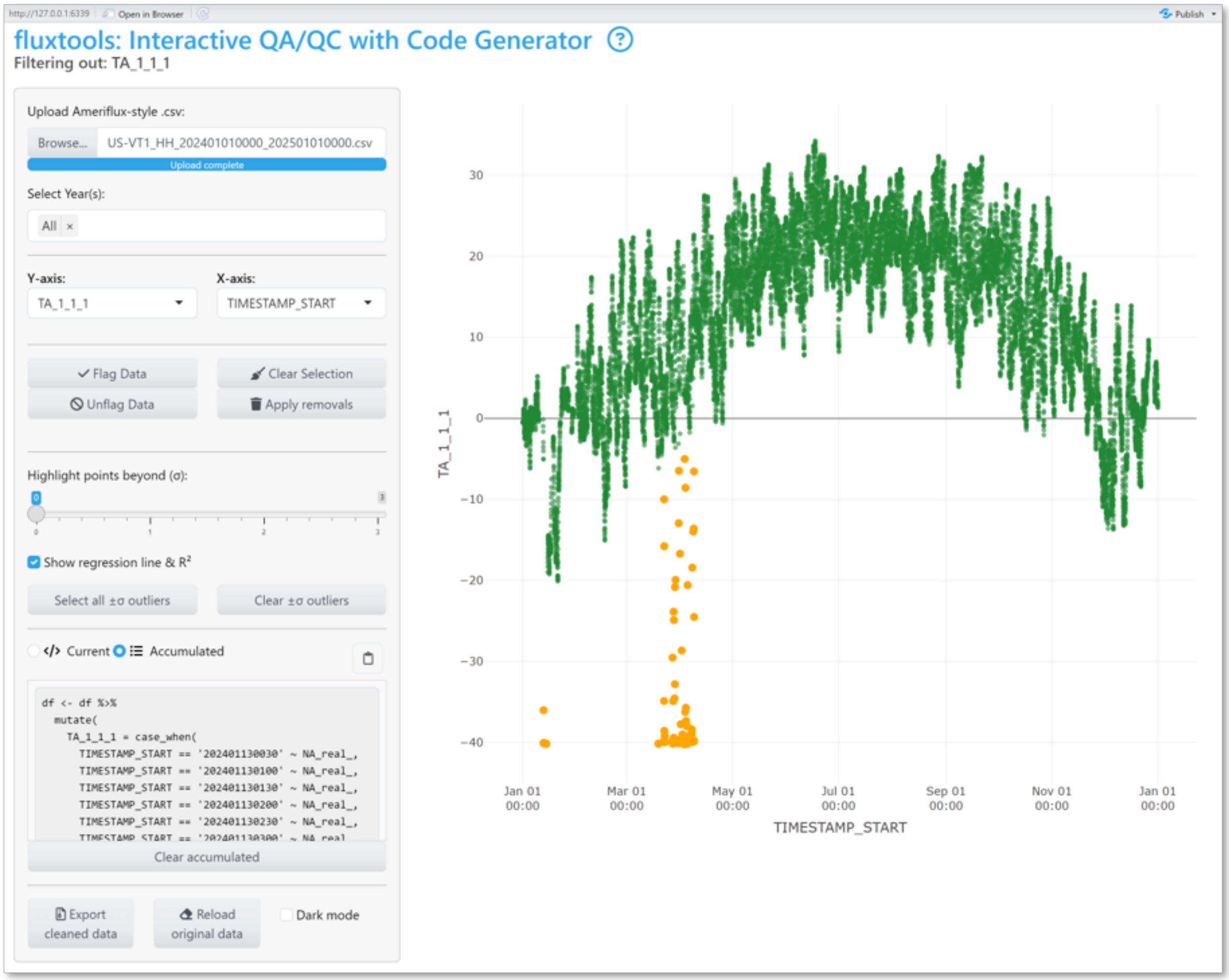
Flexible point selection: Select suspicious data points using a box, lasso, or by standard-deviation ( ) cutoffs. *See Fig 2* for interface and data selection example

On-the-fly R code generation: *Preview* pane shows selected timestamps and values; ready-to-copy R code using *dplyr's case_when(... ~ NA_real)* 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 R² diagnostics: When variables are compared against each other, a linear regression generates a R² value. Automatically computes post-removal R² value where selected data points are dropped to see step comparisons. *See Fig 3* for an example of the *Fluxtools* interfacing using the ± outliers selection tool. The top (red) R² is for all data points and the bottom R² (orange) is when selected points are dropped from the linear regression model

Export a cleaned CSV: Download a cleaned CSV, where *apply removals* turns data points into "NA", with a comprehensive R script documenting each data removal step

By reducing manual scripting and visual inspections, *Fluxtools* promotes reproducible, transparent, and efficient QA/QC workflows, greatly simplifying data preparation for Ameriflux and similar data repositories.



fluxtools: Interactive QA/QC with Code Generator

Filtering out: TA_1_1_1

Upload Ameriflux-style .csv: US-VT1_HH_202401010000_202501010000.csv

Select Year(s): All

Y-axis: TA_1_1_1

X-axis: TIMESTAMP_START

Highlight points beyond (σ): 0

☑ Show regression line & R²

○ </> Current ● ☰ Accumulated

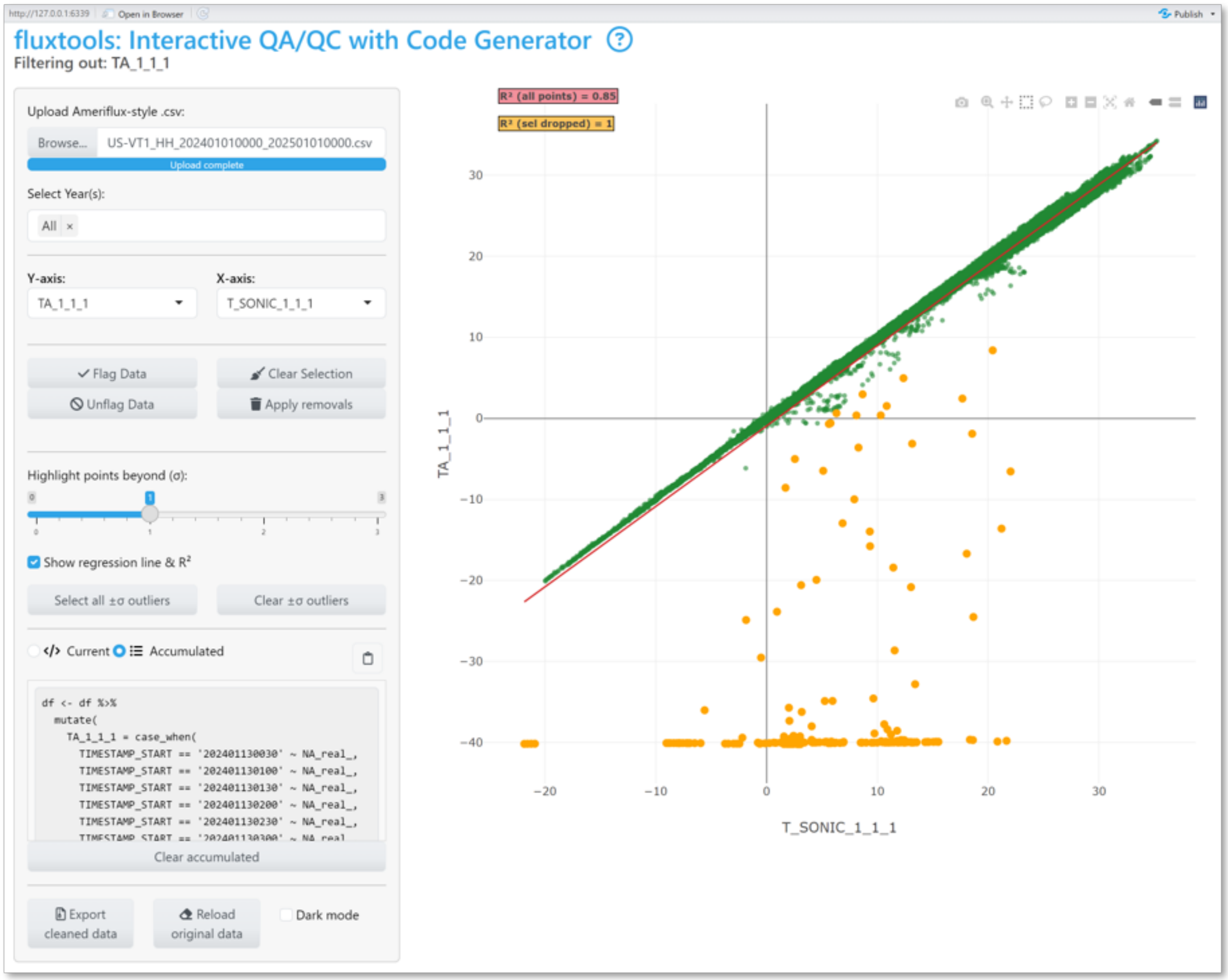
```
df <- df %>%
  mutate(
    TA_1_1_1 = case_when(
      TIMESTAMP_START == '202401130030' ~ NA_real_,
      TIMESTAMP_START == '202401130100' ~ NA_real_,
      TIMESTAMP_START == '202401130130' ~ NA_real_,
      TIMESTAMP_START == '202401130200' ~ NA_real_,
      TIMESTAMP_START == '202401130230' ~ NA_real_,
      TIMESTAMP_START == '202401130300' ~ NA_real
```

□ Dark mode

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

Statement of need

High-frequency (10 Hz; data recorded 10 times per second) eddy covariance measurements produce large datasets that must be aggregated carefully into half-hourly fluxes, requiring meticulous quality assurance and quality control. Traditional QA/QC workflows typically involve multiple custom scripts, extensive manual visualization, fragmented documentation, and significant effort to detect and remove erroneous data



fluxtools: Interactive QA/QC with Code Generator

Filtering out: TA_1_1_1

Upload Ameriflux-style .csv: US-VT1_HH_202401010000_202501010000.csv

Upload complete

Select Year(s): All

Y-axis: TA_1_1_1

X-axis: T_SONIC_1_1_1

Highlight points beyond (σ): 1

☑ Show regression line & R²

○ Current ● Accumulated

```
df <- df %>%
  mutate(
    TA_1_1_1 = case_when(
      TIMESTAMP_START == '202401130030' ~ NA_real_,
      TIMESTAMP_START == '202401130100' ~ NA_real_,
      TIMESTAMP_START == '202401130130' ~ NA_real_,
      TIMESTAMP_START == '202401130200' ~ NA_real_,
      TIMESTAMP_START == '202401130230' ~ NA_real_,
      TIMESTAMP_START == '202401130300' ~ NA_real
```

□ Dark mode

R² (all points) = 0.85

R² (sel dropped) = 1

Figure 3: Example of R² diagnostics using the ± outliers cutoffs and selection

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 these challenges by providing an interactive scatterplot-based interface paired with automatic, reproducible R code generation. Users can visually flag implausible half-hourly data points, automatically generate the exact *case_when(… ~ NA_real_) dplyr* snippets needed to set those points to NA, and either copy and paste the code for their removal into their own R script or remove them immediately in the app and export a .zip file containing a cleaned CSV plus a comprehensive R script documenting each step. This workflow guarantees transparency and reproducibility.

By combining interactivity with code-based reproducibility, *Fluxtools* significantly streamlines and clarifies the QA/QC workflow. It promotes transparent documentation of decisions, reduces manual effort, and accelerates the preparation of flux data for repository uploads such as Ameriflux (i.e., sites like US-VT1 and US-VT2). 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.

How to

```r
library(fluxtools)

fluxtools::run_flux_qaqc(-5) # To run the app, replace ‘-5’ with your site’s UTC offset (e.g., -5 is fo
```

Acknowledgments

This application was developed in my role as Co-Principal Investigator for the AmeriFlux US-VT1 and US-VT2 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 ChatGPT: GPT-4-mini-high (OpenAI 2025). All final code and revisions were authored and approved by the human author.

References

AmeriFlux. 2025a. “Site Info for US-VT1.” https://ameriflux.lbl.gov/sites/siteinfo/US-VT1.

———. 2025b. “Site Info for US-VT2.” https://ameriflux.lbl.gov/sites/siteinfo/US-VT2.

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.

Chang, Winston, Joe Cheng, J. J. Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, et al. 2024. “Shiny: Web Application Framework for R.” https://cran.r-project.org/web/packages/shiny/index.html.

Key, Kesondra. 2025. “fluxtools: Interactive QA/QC for AmeriFlux Data.” *GitHub Repository*. https://github.com/kesondrakey/fluxtools.

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

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://cran.r-project.org/web/packages/plotly/index.html.

Wickham, Hadley, Romain François, Lionel Henry, Kirill Müller, Davis Vaughan, Posit Software, and PBC. 2023. “Dplyr: A Grammar of Data Manipulation.” https://cran.r-project.org/web/packages/dplyr/index.html.