The Journal of Open Source Software

# PatchBatch: A Python Tool for Batch Analysis of Electrophysiology Data

**R. Charles Kissell** [1]¶

**1** Northeastern University, United States ¶ Corresponding author

**DOI:** 10.xxxxxx/draft

**Software**

- Review
- Repository
- Archive

**Editor:**

**Submitted:** 24 January 2026
**Published:** unpublished

## Summary

Electrophysiology methods such as patch clamp and two-electrode voltage clamp (TEVC) have contributed greatly to our understanding of physiological determinants of disease, such as the role of cardiac sodium channel dysfunction in common arrhythmias that can cause sudden cardiac events. While traditional tools for data acquisition remain valuable to electrophysiology researchers, data analysis throughput is limited by workflows that rely on outdated computational frameworks that can be drastically improved through purpose-built Python automation scripts. PatchBatch was built to standardize and streamline these workflows, greatly reducing time expenditure needed for data analysis. This frees up resources for more important tasks, facilitates greater data output, and makes feasible more exhaustive analysis strategies for electrophysiology researchers specializing in voltage-clamped experiments.

## Statement of Need

PatchBatch is a Python GUI program that provides a standardized data analysis pathway from data acquisition to data interpretation for patch clamp electrophysiology data. While programs such as PatchView (Huang, 2024) and SanPy (Cudmore, 2024) have been developed to streamline neuroscience and current-clamp data analysis, other electrophysiology domains are still lacking such innovations. As a result, many scientists studying current-voltage relationships, concentration-response curves, and ion channel characterizations via patch clamp or TEVC are frequently using tedious, outdated data processing schemas. Typical analysis of electrophysiology data through existing platforms is often repetitive, yet algorithmic, opening the opportunity for automation. It requires manually extracting the desired data by configuring analysis parameters and axis selection one file at a time (which are often the same for each file). In addition to having a high time cost, this process also invites user error resulting from tedious actions that must be repeated for each file. For one scientist, a day's patch clamp data can take an hour or more to process into a meaningful form. PatchBatch automates these repetitive data processing steps by enabling batch analysis of multiple data files at once with the same parameters, cutting analysis time to minutes rather than hours. It can also extract individual data sweeps from one or several files at a time, expediting sweep analyses that similarly are very tedious to perform manually. For example, one can extract the +100mV sweep of a full dataset from I-V relationship experiments in order to compare current waveforms measured at +100mV from multiple conditions, enabling rapid quantification of waveform changes associated with different conditions. Batch sweep export facilitates quick and easy plotting of multiple conditions for comparison.

PatchBatch was built primarily for .wcp files output by WinWCP (Dempster, 2024), and it seeks to replicate many of its data analysis functionalities, leveraging Python to automate a subset of analysis workflows performed by many electrophysiologists. It is also compatible with .abf formatted data files via pyABF (Harden, 2022). This program is intended to be usable by anyone who needs to analyze patch clamp or TEVC data, while not requiring any coding/programming knowledge. Analysis results can be copied or exported to a CSV file for downstream analysis in a dedicated data visualization tool such as Excel or Graphpad Prism. PatchBatch is intended to be an intermediate tool that serves as a quick pipeline from raw to processed data, ready for drop-in integration into a separate application for publication-ready figures.

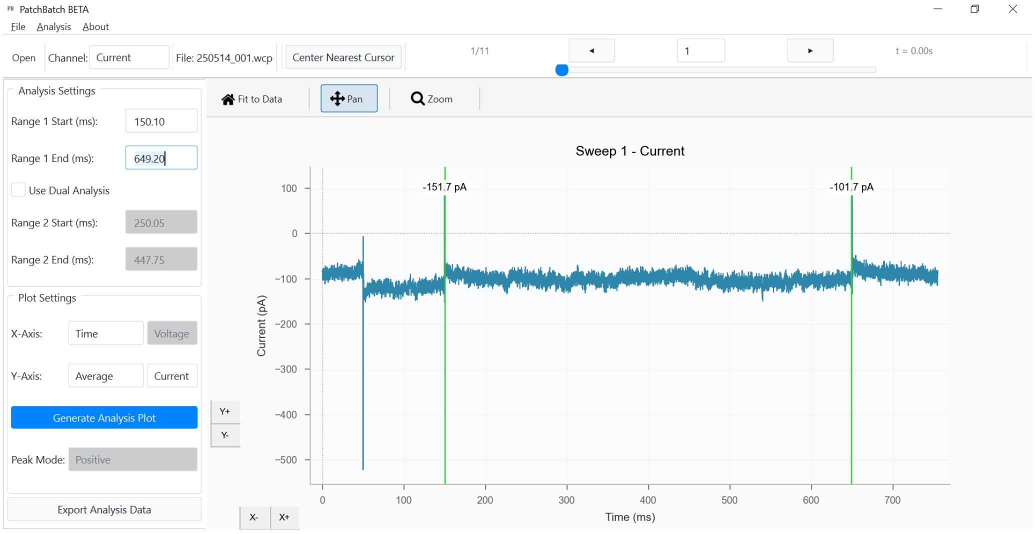
The aforementioned WinWCP, as well as Clampfit(Molecular Devices, 2024), remain valuable tools capable of extended data analysis methods, many of which (such as action potential detection) are not currently replicated in this program. PatchBatch provides a streamlined analysis pathway for dual-channel (Current channel and Voltage channel) data containing repeating Voltage/Current “sweeps”, from which scientists extract average or peak signals from a specified temporal region in each sweep. WinWCP and Clampfit lack such a streamlined analysis pathway, necessitating long, manual analysis processes that this program automates. This program also adds further data processing capabilities in the form of Current Density analysis, which simplifies the process of converting raw current values to current densities for whole-cell recordings. By facilitating the rapid processing of raw data, PatchBatch builds on previous software to provide a convenient platform for electrophysiologists to standardize and streamline their data analysis protocols. This not only saves time, but enhances reproducibility of analyses and supports more exhaustive data processing workflows.

## AI Usage Disclosure

This software was developed with the aid of large language model (LLM) tools. ChatGPT-4o and Gemini 2.5 were used during early development to produce a basic working prototype. Claude Sonnet 4.0–4.5 and Opus 4.0–4.5 (Anthropic) were used in later development to generate code and refactor the prototype into a user-friendly, fully functional software. All user-facing design decisions were made by the author based on domain expertise in electrophysiology. LLMs aided in the design of the data processing architecture, fine-tuning of the desired analysis modes and GUI features, and troubleshooting coding errors. All code outputs were reviewed and refined by the author for data operations integrity and overall functionality. All analysis scripts were extensively validated by the author by comparing analysis outputs against known reference values (Figure 3).

LLMs were used to develop pytest scripts to validate all data workflows. Pytests were validated for accurate functionality by the author. Likewise, LLMs were used to script Continuous Integration to run all pytests, and later cross-platform builds, upon each push to GitHub. All build scripts (build_windows.py, build_macos.py, and pyproject.toml) were developed with LLM assistance. Boilerplate documentation (BUILD-INSTRUCTIONS.md, CITATION.cff, CODE_OF_CONDUCT.md, LICENSE.md) were generated with LLM assistance with light refinement by the author. The design principles of ARCHITECTURE.md were written by the author, while the service layer overview was guided by LLM assistance to comprehensively describe the architecture that performs the core analysis workflow. All text of the present document and the vast majority of README.md was written by ther author and formatted with LLM assistance.

# Figures



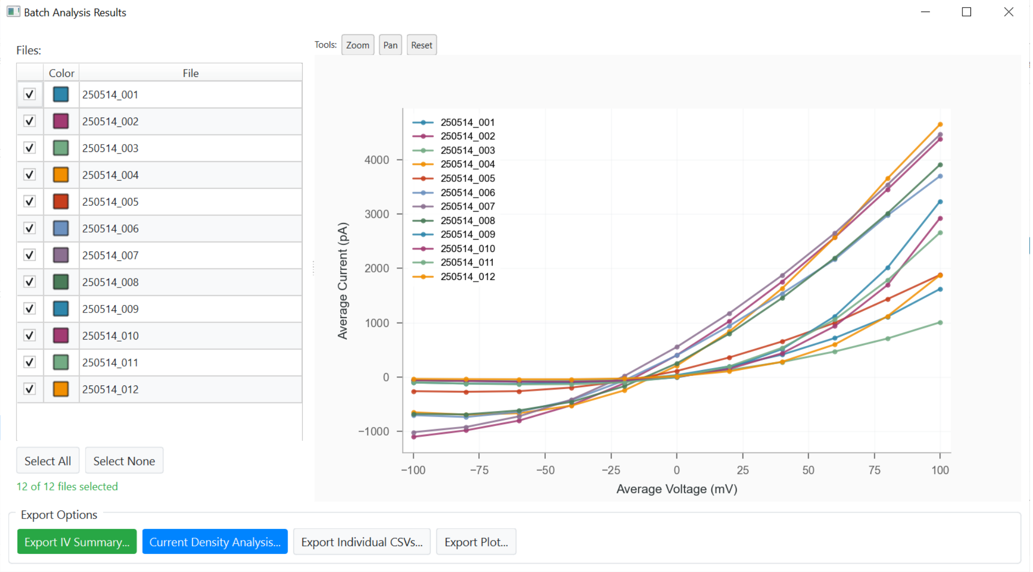
Channel: Current
File: 250514_001.wcp
Center Nearest Cursor
1/11
1
t = 0.00s

Analysis Settings
Range 1 Start (ms): 150.10
Range 1 End (ms): 649.20
□ Use Dual Analysis
Range 2 Start (ms): 250.05
Range 2 End (ms): 447.75

Plot Settings
X-Axis: Time Voltage
Y-Axis: Average Current
Generate Analysis Plot
Peak Mode: Positive
Export Analysis Data

**Figure 1:** Entry point of PatchBatch.

Initial window where data sweeps are displayed and analysis ranges defined. Analysis parameters are entered on left-side control panel. The main plot features draggable cursors whose positions are coupled to the input fields in the control panel, enabling fine-tuning of analysis parameters. Users may select one or two time ranges per analysis as needed.



Files:

| | Color | File |
|---|---|---|
| ☑ | 🟦 | 250514_001 |
| ☑ | 🟪 | 250514_002 |
| ☑ | 🟩 | 250514_003 |
| ☑ | 🟧 | 250514_004 |
| ☑ | 🟥 | 250514_005 |
| ☑ | 🟦 | 250514_006 |
| ☑ | 🟪 | 250514_007 |
| ☑ | 🟩 | 250514_008 |
| ☑ | 🟦 | 250514_009 |
| ☑ | 🟪 | 250514_010 |
| ☑ | 🟩 | 250514_011 |
| ☑ | 🟧 | 250514_012 |

Select All
Select None
12 of 12 files selected

Export Options
Export IV Summary...
Current Density Analysis...
Export Individual CSVs...
Export Plot...

**Figure 2:** Batch Analysis window.

Batch analysis enables rapid output of full dataset, each analyzed by identical parameters. Batch-analyzed data may be exported as a single summary CSV file or several individual CSV files. Data may also be copied to clipboard directly from the program.

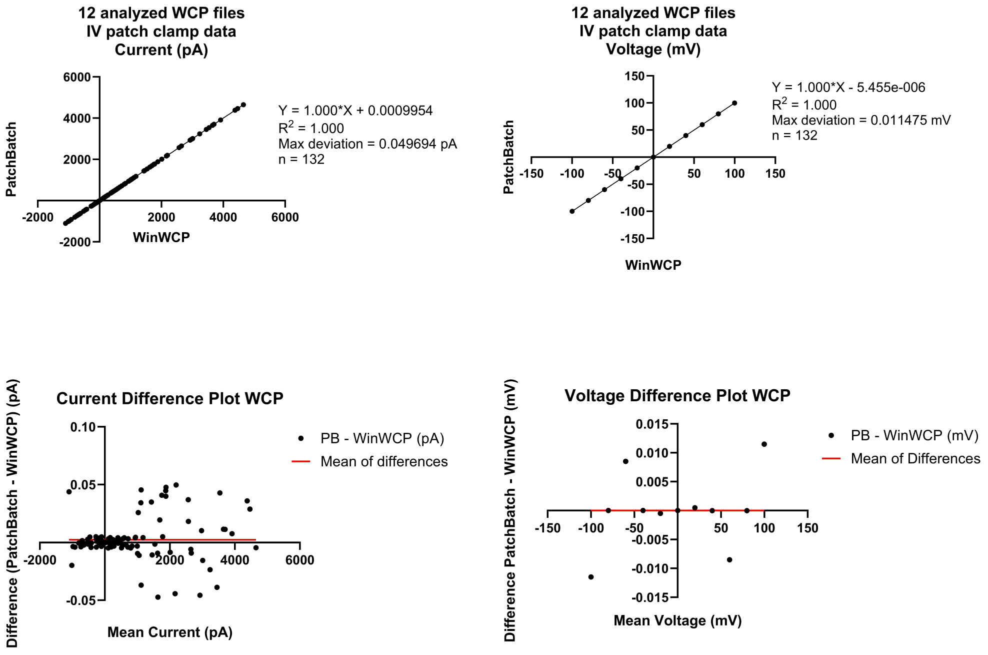


**Figure 3:** Validation of data operations.

Analysis outputs match those of WinWCP. A set of 12 files was analyzed in PatchBatch and in WinWCP, each using identical analysis parameters. Largest discrepancy was 0.05 pA and 0.01 mV for Average Current and Average Voltage measurements respectively.

# References

Cudmore, R. H. (2024). SanPy: Software for the analysis of electrophysiology data. In *GitHub repository*. GitHub. https://github.com/cudmore/SanPy

Dempster, J. (2024). WinWCP: Whole cell program for patch clamp electrophysiology. In *GitHub repository*. GitHub. https://github.com/johndempster/WinWCPXE

Harden, S. W. (2022). *pyABF* (Version 2.3.5). https://pypi.org/project/pyabf

Huang, Z. (2024). PatchView: A python-based GUI for analysis of whole-cell patch clamp recording data. In *GitHub repository*. GitHub. https://github.com/ZeitgeberH/patchview

Molecular Devices. (2024). *Clampfit 11 software: pCLAMP software suite*. Molecular Devices. https://www.moleculardevices.com/products/axon-patch-clamp-system/acquisition-and-analysis-software/pclamp-software-suite