

# Elements Read GUI: a Versatile Tool to Display and Analyse Electrophysiological Experimental Data

Eugenio Ricci<sup>1</sup>, Filippo Cona<sup>2</sup>, Stefano Severi<sup>1</sup>

<sup>1</sup> Alma Mater Studiorum - University of Bologna, Cesena, Italy

<sup>2</sup> Elements srl, Cesena, Italy

## Abstract

*In this work we developed a tool to load, display, analyse and export data coming from cellular electrophysiology experiments. This tool was realized for and thanks to Elements srl, which will distribute it as an open-source software.*

*This will allow researchers to use this GUI to perform simple analyses on their data, but also to modify it (or to add code to it) in order to implement the functionalities they desire. The possibility of customization to the needs of every laboratory was one of the main goals of our work. This is the reason why the tool was developed in MATLAB (version 2019b) using only the Home license, without the requirement of additional toolboxes.*

*The analyses that can be performed include: I/V and G/V graphs, histograms and power spectral densities (Welch's method). Furthermore, fittings (linear, exponential, Gaussian, Boltzmann's curve) can be performed on both raw data and data coming from the analyses.*

*Finally, it is possible to export the original data, the data of the analyses and the fitting parameters in a .mat file, thus allowing further and more complex analyses in MATLAB.*

## 1. Introduction

The current software to analyse electrophysiological data is mainly represented by Axon's Clampfit. This is a proprietary software with no possibility of customization and most of all with the necessity of Axon hardware, which is expensive, to generate an Axon binary file (Abf) to be read by the software. Lately, Elements srl has developed and distributed miniaturized, affordable and reliable patch-clamp amplifiers that come along with a dedicated software to analyse the acquired electrophysiological data: Elements Data Analyzer (EDA). The fact that the latter does not have direct possibilities of customization is the principal rationale of the development of the platform described in this paper.

## 2. Needs and proposed solution

Many laboratories use for example different acquisition protocols (e.g.: shorter sweeps when the cells are stimulated with absolute higher voltages, in order to not excessively stress the cells, Figure 1). This fact lead us to decide to leave the freedom to adapt the software to the data and protocols used and to the analyses desired. Another issue related to electrophysiological data acquisition is that the researchers often extract the values (e.g. maximum current in response to a voltage step) by hand, with consequent intra- and inter-operator errors. To avoid this, our effort was that of automatize this operation as much as possible. The risk in this case is that the automatic extraction of the values may not work on data obtained with particular experimental protocols, but this should not be a problem since, as already stated, the tool can be adapted to any particular need.

The Elements Read GUI is designed to be similar to the EDA software, in order to facilitate its use from owners of Elements amplifiers. Differently from EDA however, this tool was developed in a common programming language such as MATLAB so that new users could easily access its functionalities. Furthermore, and this is the main feature of this platform, Elements Read GUI is designed to be an open-source software, that anyone with basic programming skills can modify and contribute to develop.

## 3. Implementation

The tool was developed in MATLAB (2019b, Mathworks) using App Designer, a MATLAB tool dedicated to the creation of GUIs. Only the Home license was used to develop the GUI, so its use does not need any additional toolbox.

In order to load ABF files with this interface, the MATLAB script `abfload.m` (available open-source [1]) was used. The ABF2 file format is indeed a proprietary format, where the position of data and information such as the experimental protocol, the number of sweeps, etc are stored in unknown locations of the file [2]. This script takes there-

fore charge of all the operations (e.g. finding the pointer to the location of the desired data) needed to retrieve raw data and the other information about them from the ABF2 file.

The tool was designed to assign as much space as possible to the current and voltage graphs, in order to better visualize the raw data (Figure 1). The remaining space was assigned to the cursor handling fields and to fields (such as Resting Potential or Cell Capacitance) that can be useful during the analyses, since they are used to normalize raw data. The Experimental Condition field is used to name the file to be exported.

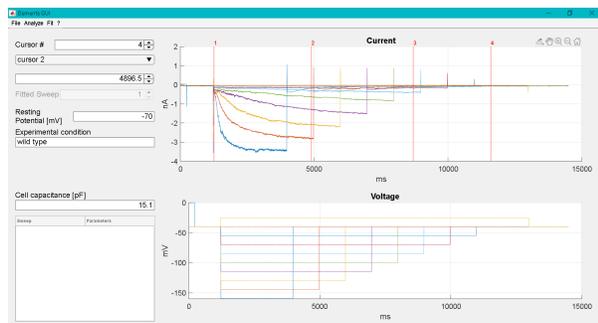


Figure 1. Appearance of the Elements Read GUI.

Finally, the drop down menu on the top left allows to access all the operations (loading, analysing, exporting, etc) that is possible to execute with this tool, operations which are explained in more detail in the next section.

#### 4. Functionalities

The Elements Read GUI owns several functionalities, namely:

- Loading .abf, .edh (Elements file format) or .mat files;
- Visualization of basic data information (file name, sampling frequency, number of samples, number of sweeps);
- Cursors handling to extract portions of signal to be analyzed;
- Data Analysis: I/V and G/V graphs, histograms, power spectral densities (Welch's method [3]);
- Fittings (linear, exponential, Gaussian, Boltzmann's curve) of both raw data and data coming from analyses;
- Exporting of original data, analyses and fittings in a .mat file.

The GUI allows one to load files through the drop down menu; this operation brings to the visualization of current data (top graph) and voltage data (bottom graph). Every sweep is plotted in a different colour. It is possible to select up to 10 cursors, in order to define the portion of signal to analyse or fit. The cursors can be selected and located with the related fields in the top left, or by dragging them with the mouse.

About the analyses, it is possible to perform I/V graphs, G/V graphs (activation graphs), histograms, power spectral densities or to plot fitting parameters one against the other (e.g. time constant of the current curve, obtained through an exponential fitting, against voltage). The capacitance of the cell can be used to normalize current data, in order to obtain a current density measure. Indeed, bigger cells tend to have more channels, thus they express higher currents. By normalizing with respect to this parameter, it is therefore possible to obtain a measure of the current flowing through a single channel. Another option, available for I/V and G/V graphs, is represented by the possibility to choose where to calculate the graphs depending on the experimental protocol (activation, deactivation): in maximum, minimum and mean (chosen by default) point of the sweep. In the G/V graphs, it is also possible to use the tail currents, in order to avoid the dependence from the driving force.

In Figure 2, an example of the I/V relationship obtained from data shown in Figure 1 is depicted.

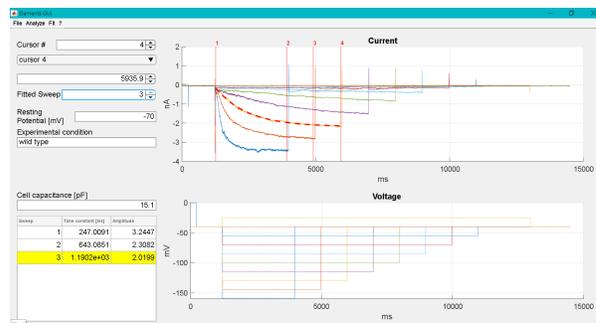


Figure 2. Example of analysis and fitting performed through the tool: exponential fitting of an I/V graph, with fitting parameters highlighted on the left (window in the foreground).

Note how the analysis is shown in a new window, which first of all presents the possibility to execute fittings on the analysis itself (in this case an exponential fitting was performed), but also to change the axes mapping from linear to quadratic, logarithmic or square root. This can be useful if the data have a particular trend (e.g.: an exponential curve will appear as a straight line if y axis mapping is set to logarithmic). Fittings can be performed on raw current data as well, in both cases the fitting parameters are shown in a table on the left, with the values of the selected curve highlighted (Figure 3). Finally, the GUI allows one to export raw data, analyses data and fitting parameters in a .mat file, so that one can perform more complex analyses directly in MATLAB.

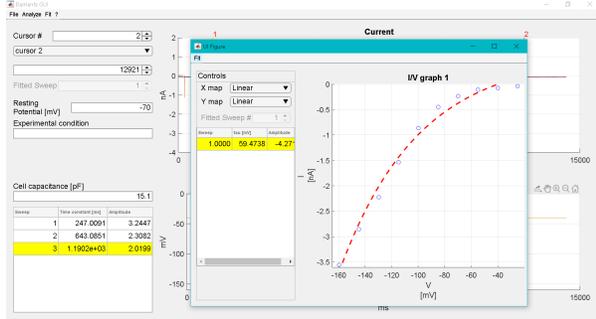


Figure 3. Fitting curves and parameters of raw data, 3<sup>rd</sup> sweep selected.

## 5. Design & development

Figure 4 shows the development environment: App Designer. On the top left it is possible to create new callback functions inside which to insert the code to be executed when an event triggers the callback. In this case it was chosen not to directly write the code, but to call a script (or a function, for better performance) containing it, in order to access the code more neatly and thus simplify its modification.

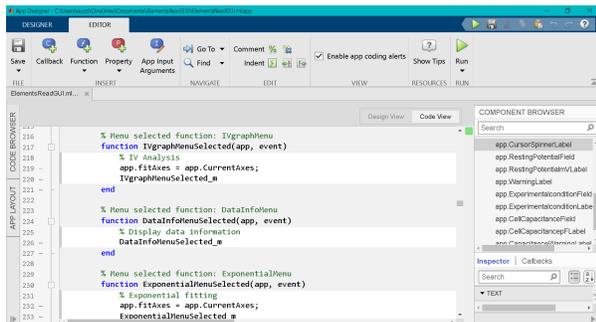


Figure 4. Appearance of MATLAB App Designer: the callback functions contain the calls to scripts with the code to be executed.

The easier way to modify a functionality is therefore changing the dedicated script/function, whereas in order to add a new feature, one can create a new script/function and call it inside a new callback.

About multi-step procedures, new objects can be created with their own functionalities. For example, the platform already provides a new window for the analyses (e.g. I/V graph) on which to execute some fitting (Figure 2). This is obtained by calling a new object created in App Designer (AnalysisFigure) - with its own callbacks (that can execute the fitting) - inside the callback of the Elements Read GUI object that executes the I/V graph.

However, another possibility is that of exporting the data (raw, analysis and fitting ones) in a .mat file in order to load

them directly in MATLAB and thus exploit all of its built-in functions or other scripts made by the user to process the data.

## 6. Conclusions

We developed an open-source tool - Elements Read GUI - to load, visualize, analyse and export electrophysiological experimental data. This tool gives the users the possibility to execute simple analyses and fittings on the data, as well as the option to export the same data in order to further analyze them in MATLAB.

The main feature of the platform is that, being open-source, it is fully editable: the user is free to modify the existing code or to add new features basing on his needs or preferences.

The complete customization, in terms of analyses executed, disposition of the controls, etc is the feature that makes this tool so versatile, in that it can be adapted to many different experimental protocols using a common and easy programming language such as MATLAB.

To conclude, the tool is downloadable under the name "EDAmat" on the Elements srl website [4], where soon a link to the github repository will be added to give the possibility to contribute to this open-source project.

## References

- [1] Collman F. abfload.m. <https://github.com/fcollman/abfload>.
- [2] Harden SW. Unofficial guide to the abf file format. <https://sharden.com/pyabf/abf-file-format.md.html>.
- [3] Welch P. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics* 1967;15(2):70–73.
- [4] Elements. Miniaturized, easy to use, affordable instruments for electrophysiology and nanopore applications. <https://elements-ic.com/>.

Address for correspondence:

Filippo Cona  
 R&D SW engineer, Elements srl,  
 Viale G. Marconi 438, 47521 Cesena (FC),  
 Italy  
 fcona@elements-ic.com