Easy Electrophysiology

User Manual

Contents

1. Welcome
2. Acknowledgements
3. Graphical Interface Overview
4. Loading Files
5. Analysis Overview
6. Curve Fitting
7. Event Detection and Analysis
    a. Background
    b. Events – Thresholding
    c. Events - Template
        - Generate Template
        - Refine Template
    d. Analysis Options
    e. Events Overlay
    f. Average Event
    g. Cumulative Probability & Histograms
8. Action Potential Counting
9. Input Resistance
10. Action Potential Kinetics
11. Data Tools
    - Upsample
    - Downsample
    - Filter
    - Detrend
    - Remove Baseline
    - Average Records
    - Record Arithmetic
    - Stimulus Artefact Removal
    - Reshape Records
    - Normalize Time
    - Cut Trace Length
    - Reset to Raw Data
12. Analysis
    a. Kolmogorov-Smirnov Tests
    b. Summary Statistics
13. Burst Analysis
14. Analyzing in Batch Mode
15. Saving Data and Analysis Results
16. Saving and Resetting Option Defaults
17. Graph Options
18. Plot Exporter
19. Legacy Options
20. Keyboard Shortcuts
21. License

Appendix I – Mathematical Formulas

1. Welcome

Welcome to Easy Electrophysiology. Our aim is to provide a platform for quick and easy patch-clamp analysis – with extensive opportunity for checking and manual adjustment.

This documentation provides detail on available options, analysis routines, underlying algorithms and formulas. All file-loading and analysis code is open source [1] and we are happy to answer questions on implementation and code.

We hope you will find our software a useful tool in your patch-clamp electrophysiology analysis. Please don't hesitate to contact us at support@easyelectrophysiology.com with any questions, suggestions or feedback.

Happy Analyzing!

[1] https://github.com/easy-electrophysiology/easy-electrophysiology-analysis

2. Acknowledgements

For those using Easy Electrophysiology in published research, our RRID is RRID:SCR_021190.

Neo

Easy Electrophysiology uses Neo (Garcia et al., 2014), a data-loading package that handles many electrophysiological filetypes.

Garcia, S., Guarino, D., Jaillet, F., Jennings, T., Pröpper, R., Rautenberg, P. L., Rodgers, C. C., Sobolev, A., Wachtler, T., Yger, P., & Davison, A. P. (2014). Neo: An object model for handling electrophysiology data in multiple formats. Frontiers in Neuroinformatics, 8:10.

We also thank the contributing authors of the relevant Neo submodules including Forrest Collman, Harald Hentschke (Axon files) and Jeffrey Gill (AxoGraph files).

Events – Template Matching

The template matching event detection algorithm with a biexponential template and correlation coefficient threshold was first described by:

Jonas, P., Major, G., & Sakmann, B. (1993). Quantal components of unitary EPSCs at the mossy fibre synapse on CA3 pyramidal cells of rat hippocampus. Journal of Physiology, 472, 615-663.

The Detection Criterion thresholding method was introduced by:

Clements, J. D., & Bekkers, J. M. (1997). Detection of spontaneous synaptic events with an optimally selected template. Biophysical Journal, 73(1), 220-229.

The deconvolution method for template event detection was introduced by:

Pernia-Andrade, A., Goswami, S. P., Stickler, Y., Frobe, U. Schlogl, A., & Jonas, P. (2012). A Deconvolution-Based Method with High Sensitivity and Temporal Resolution for Detection of Spontaneous Synaptic Currents In Vitro and In Vivo. Biophysical Journal. 103(7), 1429-1439.

Action Potential Threshold Detection

Easy Electrophysiology provides a number of options for threshold estimation in *Action Potential Kinetics* analysis, including Method I and II from Sekerli et al. (2004):

Sekerli, M., Del Negro, C. A., Lee, R. H., & Butera, R. J. (2004). Estimating action potential thresholds from neuronal time-series: New metrics and evaluation of methodologies. IEEE Transactions on Biomedical Engineering, 51(9), 1665–1672.

and maximum curvature threshold detection:

Rossokhin, A. V., & Saakian, I. Z. (1992). Dependence of the nerve pulse generation threshold on the ability of forming an input signal based on the Hodgkin-Huxley model. Biofizika, 37(6), 1071–1075.

Spike Frequency Accommodation – Local Variance

The Local Variance measure for spike frequency accommodation from Shinomoto et al., (2003):

Shinomoto S, Shima K, Tanji J. (2003). Differences in spiking patterns among cortical neurons. Neural Computation, 15, 2823–2842.

LogISI

The logISI method was introduced, and extensively detailed, by Pasquale et al., 2010:

PyQtGraph

We would also like to thank Luke Campagnola and the PyQtGraph team for their work on the excellent PyQtGraph package (http://www.pyqtgraph.org) that provides high performance, flexible graphing.

3. Graphical Interface Overview

The ‘Plot’ and ‘Table’ views are the two main ways to visualize data and results in Easy Electrophysiology, accessed through the ‘Plot’ and ‘Table’ tabs (Fig. 1). The Plot tab is used to view voltage (Vm) and current (Im) traces, setup and run analysis and visualize the results. The ‘Table’ tab displays results in table format with options for saving to Excel or CSV.

To choose an analysis to run, click ‘File’ > ‘Open Analysis Window’. When an analysis type is selected, the ‘Analysis Panel’ on the left-hand side of the Plot tab will show the relevant options for running that analysis.

In the Table plot, options for saving analysis results can be found using the drop-down menu in the ‘Save Table’ box. Results can be saved as .xlsx (includes border formatting) or as .csv (no formatting). For some analyses (*Events*, *Action Potential Counting, Input Resistance*), checkboxes in the left-hand panel can be used to select which results to display.

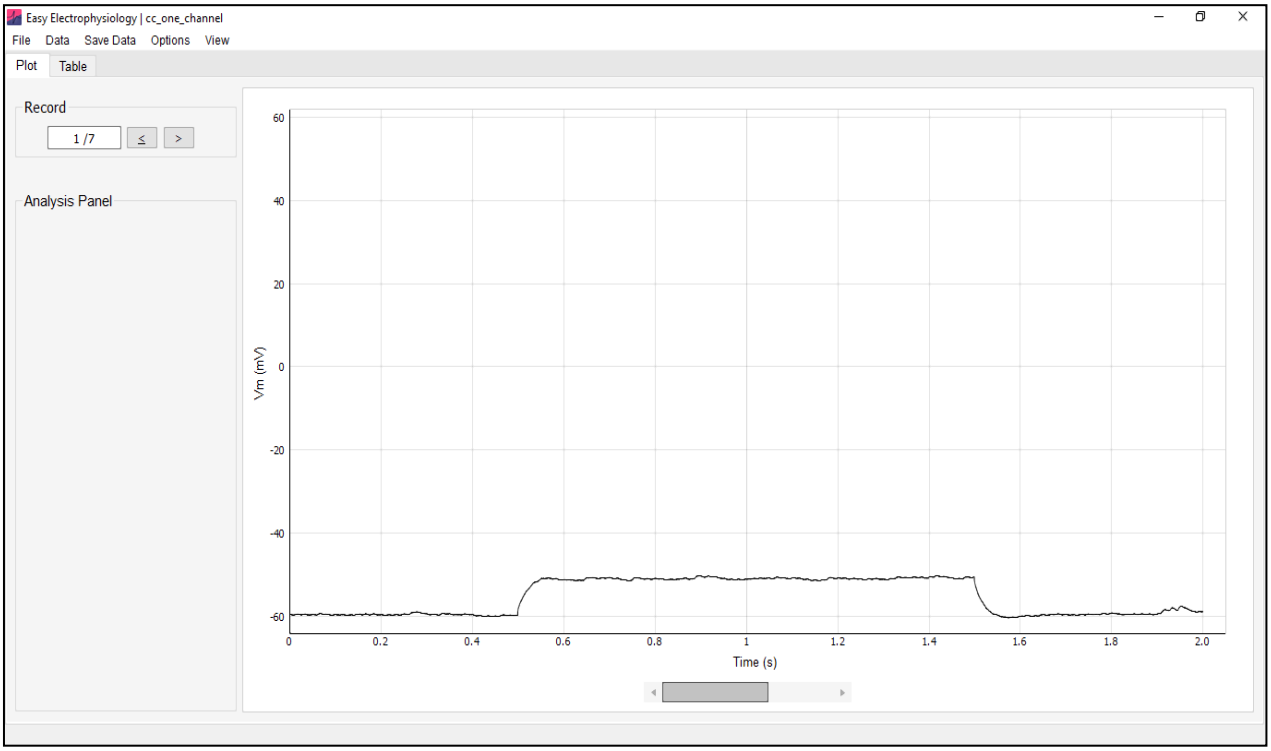

Figure 1. Graphical interface overview. File data and analysis results are displayed in the ‘Plot’ tab graph and tabular analysis results are displayed in the ‘Table’ tab. The analysis panel (*left*, *bottom*) contain options for analysis when an analysis type is selected. The buttons in the ‘Record’ box (*left, top*) and scrollbar under the graph can be used to move through records.

Zooming and Panning the Graph

There are two zoom / pan modes available when using the Plot tab graph to explore your data. By default, ‘One Button’ mode is selected and you can zoom by clicking anywhere on the plot and dragging the mouse to draw a box to zoom in to. To un-zoom, you can right click on the graph and click 'Undo Zoom'

In the second mode, both mouse buttons can be used to navigate the graph. To access the second mode right-click anywhere on the graph and select 'Mouse Mode' > 'Two button (pan left, zoom right)'. To zoom, hold down the right mouse button and drag the mouse to dynamically zoom in and out. To pan, click and hold down the left mouse button and drag to move the plot. You can still use 'Undo Zoom' in this mode.

4. Loading Files

Axon (ABF-1, ABF-2), Axograph (.axgx, .axgd), HEKA (.dat), WinWCP / WinEDR (.wcp / .edr) and Spike2 (.smr) filetypes can be loaded into Easy Electrophysiology. Please contact us at if you have other filetypes you would like to see supported.

Files can be drag-and-dropped directly onto the main window or loaded by clicking ‘File’ > ‘Open File’. Note at the sampling frequency must be identical for all records in the file. Multiple files can also be loaded at once, and they will be concatenated together. A window for selecting the order and method by which the files are concatenated will appear.

For some .ABF files, all records may be erroneously concatenated to a single record (typically if exported from software that is not pClamp). If this occurs, please see the ‘Reshape Records’ section of this manual.

The following default file-loading options can be set on the File’ > ‘File Loading Options menu (Fig. 2).

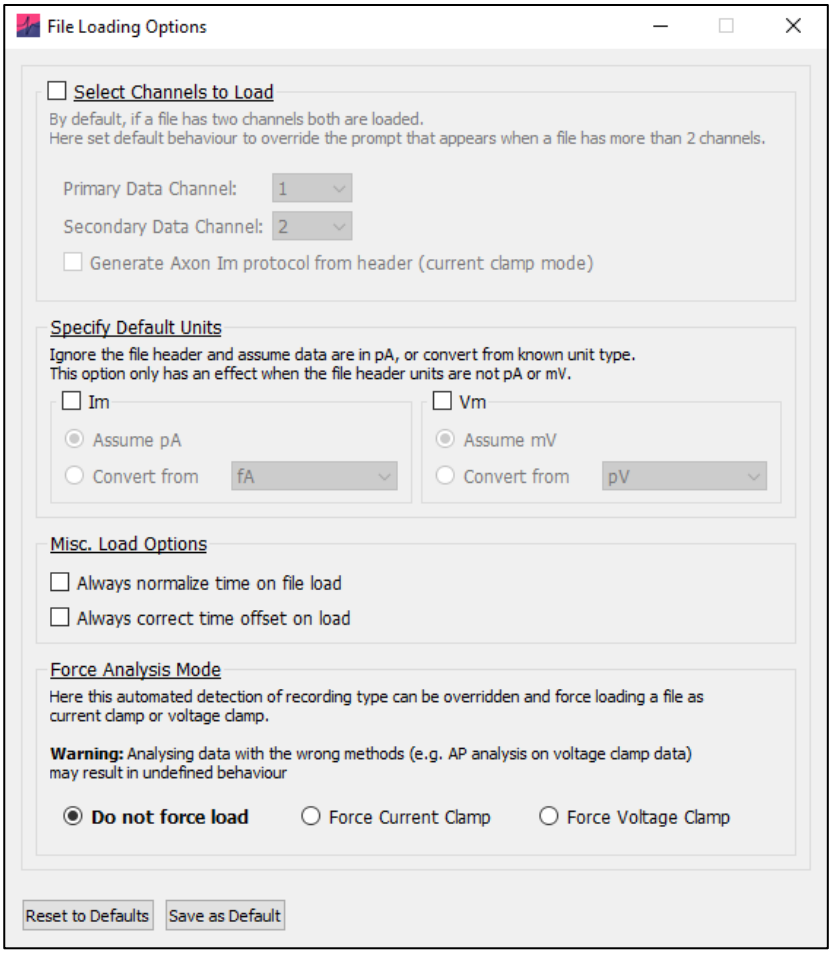

Figure 2. File Loading Options. Set default behaviour when loading files.

Select Channels to Load

Data from up to two channels can be loaded at once. The ‘primary data channel’ is the primary channel for all analysis (e.g. *Curve Fitting* is conducted on the primary data channel) and displayed in the upper graph. By default, the first channel in the file is selected as the primary data channel. The ‘secondary data channel’ is typically used for current / voltage injection protocols. In *Action Potential Counting* and *Input Resistance* analysis, Im values are calculated from the secondary data channel.

If the file contains more than two channels, default channels to load can be selected from the drop-down menu. To only load a single channel, set ‘Secondary data channel’ to ‘None’.

‘Generate Axon Im protocol from header’ will generate the secondary data channel from Axon files in which the stimulus protocol is stored in the header, rather than in raw data. Currently this option is only available in current clamp mode.

Specify Default Units

Analysis routines expect units in seconds (s), picoamperes (pA) and millivolts (mV). If time units are detected as milliseconds (ms) in the file header, they will be automatically converted to seconds. If units are not in mV or pA, a window will warn that the file contains unexpected units. To load the file, click ‘specify default values’ and select the units to convert from the drop-down menu (e.g. if the file units are in A, select convert from A).

In some cases, the units specified in the file header do not match the true values. In this case, the dialog will also show a warning and an estimate of the true unit scaling (Fig. 3). Again select ‘specify default values’ and select the appropriate file units to convert from, or select ‘Assume pA / mV’ (e.g. if the file header is in units A and V but the data is scaled to pA or mV).

These options only have an effect when the file header is not in pA or mV.

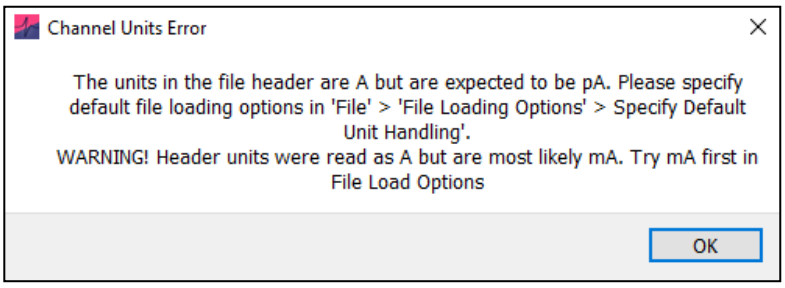

Figure 3. Channel Units Error. The warning shown when the units in the file header are not pA / mV.

Always Normalize Time on File Load

Select this option to ignore when time is recorded as cumulatively increasing across records. See the ‘Normalize Time’ section of this manual for more details.

Always Correct Time Offset on Load

On some recording setups, a small offset is added resulting in a non-zero starting timepoint (e.g. Record 1: 0.02 – 1.52 s). If detected, a window will warn the user. To ignore this window and always correct time offset, select *‘Always correct time offset on load’* option. This is recommended as small errors may be introduced when user-input times are required if the user is not aware of the offset. For example, a stimulus onset may be input as 0.1 s but due to the offset occur on the trace at 0.12 s.

Force Analysis Mode (Voltage- vs. Current-clamp)

By default, the ‘recording type’ (current-clamp or voltage-clamp) is automatically detected based on the primary and secondary channel units in the file header. This determines which analyses are available to run (e.g. *Action Potential* analysis is only possible for current clamp recordings). Here automatic detection can be overridden and the selected analysis options will be available no matter the loaded filetype. Note that this may result in undefined behavior if incorrect analyses are conducted (e.g. *Action Potential Counting* on voltage-clamp data).

5. Analysis Overview

The ‘Analysis’ window is used to switch between analysis types and can be accessed through ‘File’ > ‘Open Analysis Window’ (Fig. 4).

The analyses available to are determined by the file recording type. *Curve Fitting* is supported for both voltage-clamp and current-clamp recordings, while *Event* detection and analysis is supported for single-record voltage-clamp recordings. *Action Potential Counting, Action Potential Kinetics* and *Input Resistance* are available for current-clamp recordings.

For maximum flexibility, each analysis type is independent and can be run concurrently. This means it is possible to switch from one analysis type to another mid-analysis without overwriting results. The relevant plots and results tables will be updated on analysis change.

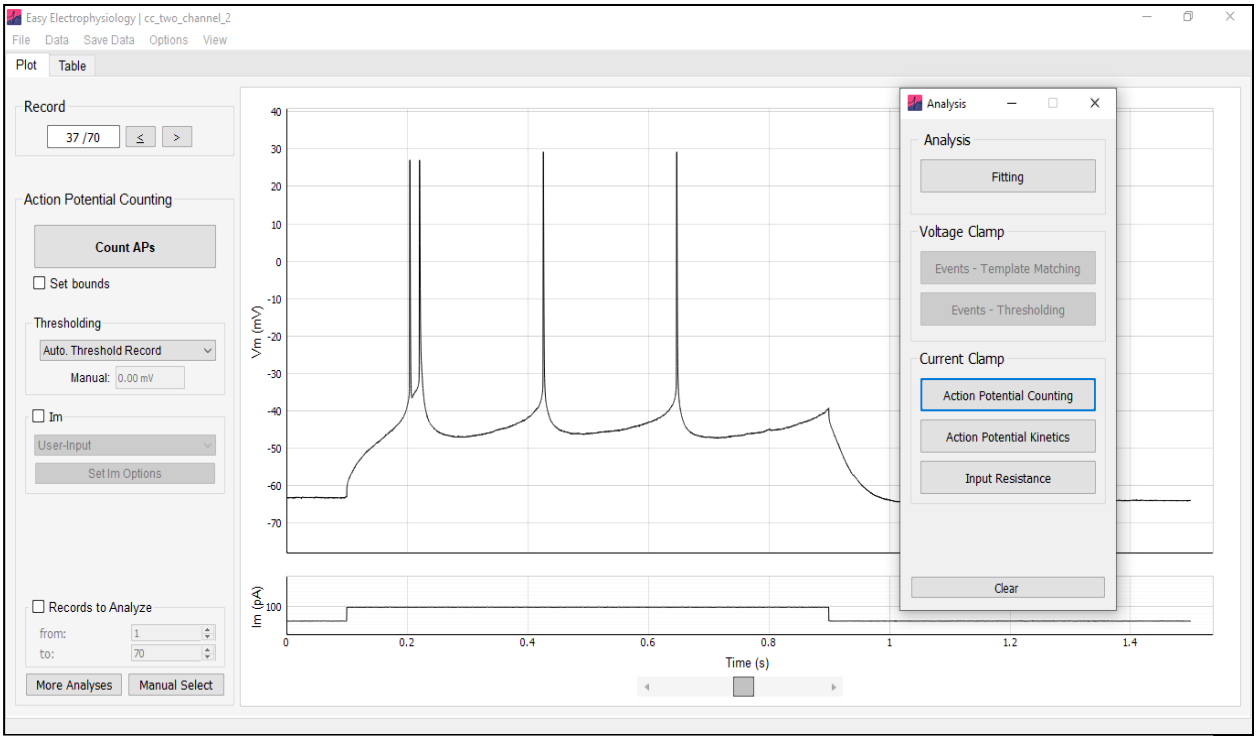

Figure 4. Switching Analysis. The analysis type can be switched through the ‘Analysis’ options window (‘File’ > ‘Analysis Options Window’). Switching to an analysis type will bring up the relevant options on the analysis panel (*left*).

6. Curve Fitting

Curve Fitting is used for calculating summary statistics or fitting functions to the data (see *Functions to Fit* and *Appendix I* for details of available analyses). The best-fit curve is plot on the graph (Fig. 5) and coefficients displayed in the analysis results table. Up to six different fits can be applied simultaneously – this is useful for recording protocols requiring measurement of multiple evoked events (e.g. paired-pulse recordings with a seal-checking voltage step). All curve fitting analysis is conducted on the primary input channel.

Curve fitting analysis fits a curve for every record in the file by default. To select specific records to analyze set the 'Records to Analyze' box at the bottom of the *Curve Fitting* analysis panel.

Curve Fitting Regions

Curve Fitting analysis is conducted within 'curve fitting regions'. These consist of two regions, the 'baseline' region and the 'measure' region. The baseline is calculated as the average of all datapoints within the baseline region. The measured region defines the section of data to fit.

Each of the six available regions has a designated color (red, blue, green, orange, pink and teal; Fig 6). Switching between regions is achieved using the left / right buttons on the *Curve Fitting* analysis panel, and *Curve Fitting Options* window. Only the currently selected region can be moved.

Switching to a curve fitting region which is not yet shown will initialize the region. To remove a region and delete all associated analysis, use the 'Delete selected region / analysis' box on the analysis panel.

Curve Fitting Options

Options for conducting curve fitting analysis are accessed by clicking the 'Curve Fitting Options' button on the analysis panel. The following view options for the curve fitting regions are available:

- *Show region* – show both the baseline and measure region.
- *Hide baseline* – show the measure region only.
- *Hide region* – hide both the baseline and measure region.
- *Show selected region only* – Hide all regions except the currently selected region.

- *Link Baseline* – By default the baseline and measure regions are linked i.e. moving the baseline region will move the measure region and vice-versa. This option switches between linking / unlinking the baseline and measure regions so they can be moved separately.

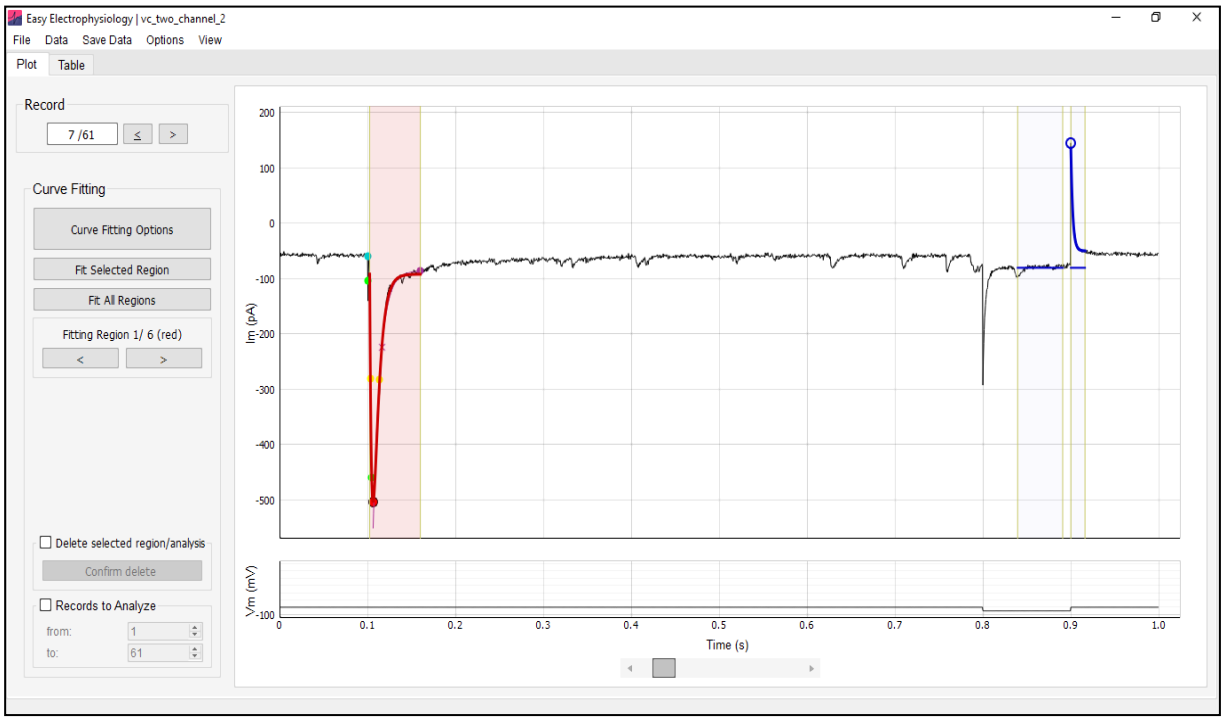

Figure 5. Curve Fitting Analysis. Curve fitting boundaries are used to define the 'experimental' and 'baseline' regions. The experimental regions indicate the data over which to fit the curve. In red, a *biexponential (event)* is fit to the data (and event kinetics calculated) between the experimental region (baseline region not shown). The baseline region defines the data over which to average to calculate the baseline (show with a horizontal line). In blue, a monoexponential function is fit in the experimental region with the baseline shown.

Functions to Fit

Summary statistics or functions to fit to the data within the measure region can be selected, along with starting and boundary coefficients, on the *Curve Fitting Options* window. Following analysis, the baseline, peak, amplitude (peak minus baseline) and coefficients of the best-fit curve are shown in the 'Table' tab table.

The mathematical formulas for all functions can be found in *Appendix I* of this documentation, and their implementation at our GitHub. Please see below for a full list of parameters and functions:

- *Minimum / maximum:* Finds the minimum or maximum (depending on selection) within the measured region.
- *Mean / Median:* Finds the mean / median (depending on selection) of the measured region. The 'amplitude' shown in the results table represents the difference between this average and the baseline.
- *Area Under Curve (AUC)*: Calculates the area under the curve of the measured region, with respect to the baseline.
- *Max Slope:* Finds the maximum slope within the measured region, calculated as a regression over n samples. Options are provided for selecting slope direction, the number of samples to calculate the regression over, and applying smoothing prior to calculation.
- *Monoexponential, biexponential (decay) and triexponential:* Fit an exponential curve with one, two or three terms to the data within the measure region.
- *Biexponential (event).* Fit a biexponential curve in the form used for event-detection (with rise and decay coefficients). This will also calculate event kinetics (rise-time, half-width, decay). Event analysis options can be accessed with the 'Event Kinetics Options' button on the *biexponential (event)* panel. These include:
    - *Average peak (ms)* – the time period to average the event peak (see the *Average peak (ms)* section in *Events* for detail).
    - *Baseline search settings* – 'Auto.' Will automatically detect the foot of the event and use this as baseline. Alternatively, 'Use Baseline' can be selected which will calculate the baseline as the first sample the data that crosses the baseline in the direction opposite to the peak (e.g. if the peak is negative it will look for the first positive crossing of the baseline).
    - *Baseline search period (ms)* – period to search for the baseline.
    - *Average baseline (ms)* – period over which to average the data preceding the baseline).

Note that these configurations are separate from those in *Events* analysis (*Events – Template Matching*, *Events – Thresholding*). This means that changing these settings will not affect other event analyses. This is distinct from the events settings accessed through 'Options' > 'Events Analysis' – 'Misc. Options' which are shared between all Events analysis.

Curve Fitting Algorithm

Least-squares fitting of exponential functions uses the Trust Region Reflective algorithm as implemented in the SciPy scientific analysis package (*optimize.least_squares*).

The starting coefficients of exponential fits are estimated from the data (see *Appendix I*). The only default bounds on the coefficients is that exponential time constants ('tau') must be positive. For biexponential event fitting the default starting coefficients are 0.5 ms and 5 ms (Jonas et al, 1993). These default coefficients are always used for *biexponential (event)* Curve Fitting analysis and are distinct from those the *Events Analysis – Misc. Options* menu (which are for event template generation only).

For all functions, default starting coefficients and bounds can overridden in the *Curve Fitting Options* window.

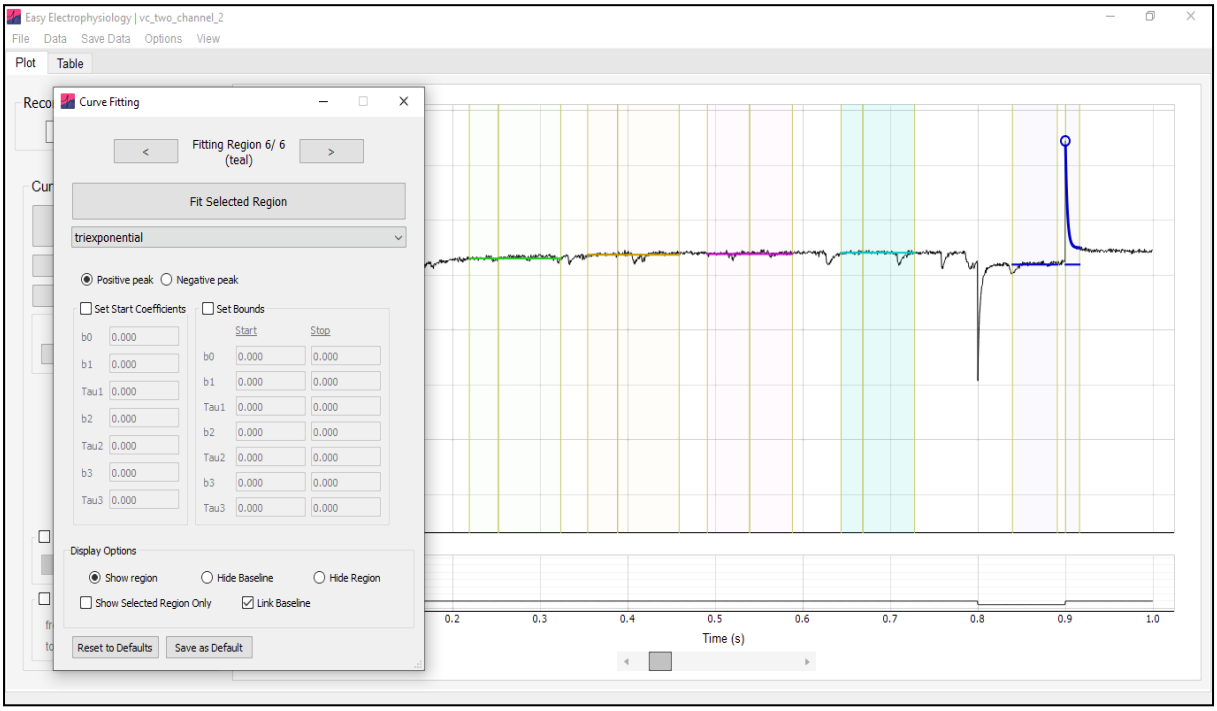

Figure 6. Curve Fitting Analysis Options. The *Curve Fitting Options* window contains options for setting starting coefficients and boundaries. Additional options for displaying regions (e.g. show region, hide region) are also available. Fitting up to six curves simultaneously (in red, blue, green, orange, purple, teal curve fitting regions) is supported.

7. Event Detection and Analysis

Overview

Easy Electrophysiology provides two options for events analysis, *Events – Template Matching* and *Events –Thresholding*. For simplicity, all relevant options, graphs and results tables are shared between *Events* detection methods (e.g. running an events analysis with *Template Matching* will overwrite a previous analysis with T*hresholding*).

Events analysis is currently only supported for single-record voltage clamp data. To reshape data from multi-record to single-record, please see the 'Reshape Records' section of this manual.

Event Detection Methods – Background

Events – Template Matching

Template matching relies on the generation of a 'template' event in the form of a biexponential function (Jonas et al., 1993 see also Guzman et al., 2014). The template is generated by fitting the biexponential function to an exemplar event from the recorded data (the key parameters to fit are the 'rise' and 'decay' coefficients; *Appendix I*). This ensures the biexponential rise, decay, and fitting window are tailored to the shape of the post-synaptic events to be detected. This initial event template can be further refined by fitting to the average of all detected events (see *Refine Template* for details).

The main aim of all event detection algorithms is to calculate the measure of the similarity between the data and the template at every sample in the data. Easy Electrophysiology implements the correlation method (Jonas et al., 1993), the detection criterion method (Clements and Bekkers, 1997) and the deconvolution method (Javier Pernía-Andrade et al., 2012). In the correlation approach the template is correlated to the data at each timepoint using a sliding window. A correlation cutoff value is then specified (default r = 0.40) and a contiguous area above the correlation cutoff is considered an event region. In the Detection Criteria approach a 'detection criterion' (default = 4) rather than correlation is calculated. This scales the event amplitude to the noise and is closely related to the signal-to-noise ratio of the event. Both correlation and detection criterion are implemented using sliding window implementation of Clements and Bekkers, (1997) for performance. The correlation and detection criterion generally perform well but have poor temporal resolution and typically fail to detect high-frequency, closely spaced events. In contrast, the deconvolution method can

reliably detect closely spaced events. In the deconvolution method, deconvolution of template and data is performed in the Fourier domain and bandpass filtered with a Gaussian window. The default cutoff value (specified as standard deviation of the amplitude distribution) is 3.5, with default cutoff for the bandpass filter 0.1 – 200 Hz. These often need to be adjusted to the dataset (see *Refine Template).*

These approaches all yield a 'similarity measure' indicating the similarity between the template and data and each sample point. This is then thresholded with the respective cutoff value, and any contiguous event that exceeds this threshold is considered a detected event. Interactive plots of the biexponential event fit and similarity measure across the entire dataset to help assess the fit (see *Refine Template* for details).

Events – Thresholding

Events thresholding is a simpler method of event detection in which any datapoint that crosses a set threshold is considered an event. This method is less precise than the template matching method (e.g. a noise spike over threshold would be considered an event) but is less computationally demanding.

Events – Template Matching

Generate Template

The first step in template matching is to generate the template event (Fig. 7). Click '1. Generate Template' on the *Events – Template Matching* analysis panel for template generation. A biexponential event template is shown on the graph with sliders and input boxes for manual adjustment of template coefficients. It is possible to increase the coefficients outside the slider range by typing the values directly into the input boxes. The coefficients that define the template shape are:

- *b0* and *b1*– the offset (mean) and slope parameter that determines the direction (positive or negative) and event amplitude, respectively. These coefficients are left free during event detection and so are not important for template generation.
- *Rise* – the event rise-time (defaults 0.5 ms from Jonas et al., 1993)
- *Decay* – the event decay-time (default of 5 ms from Jonas et al., 1993)
- *Width* – The width (in ms) of the event template.

The best way to generate a template event is to fit the biexponential function to a recorded event from the dataset. An event from the data can be selected by clicking 'Select Data' and drawing around an event to fit. When selecting an event it is helpful to change the graph zoom settings to *two button* by right-clicking clicking on the plot, 'Mouse Mode' > 'Two Button (Pan Left, Zoom Right)'. This allows the left button to zoom while selecting data with the right button.

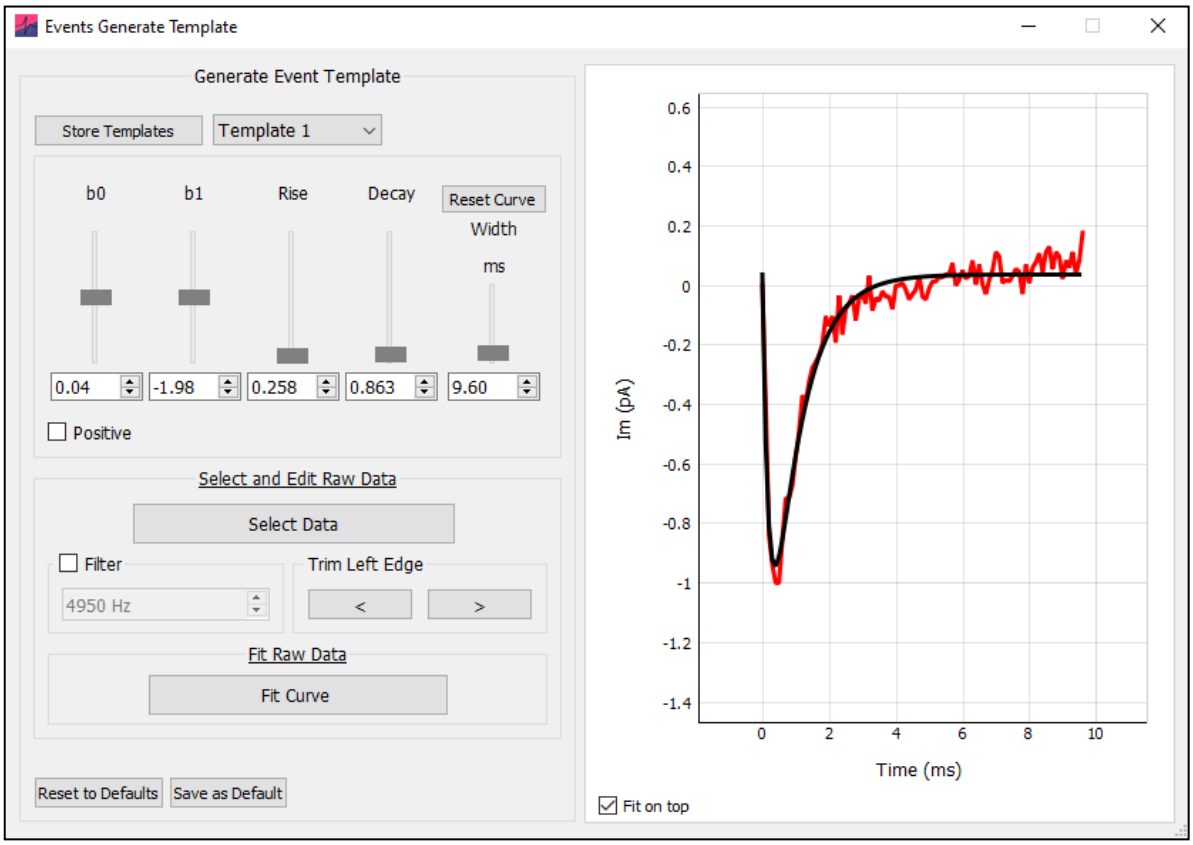

Figure 7. Generate template. The *Generate Template* Window. Selected data is shown in red, and the current template fit to the data in black. Coefficients for the fit are displayed in the input boxes.

Fitting the Event

Once an event from the data is selected it will appear on the graph red). The start of the event to fit can be shifted with the buttons in the 'Trim Left Edge' box – the start point is critical for optimal fitting. The event from the data can also be filtered prior to fitting (8th order Bessel).

Clicking 'Fit Curve' will fit the biexponential function to the event. This will change the coefficients in the top panel. These coefficients are now saved until Easy Electrophysiology

is closed. To save these for the next session, click the ‘Save button (see *Saving and Resetting Option Defaults* for details).

Storing Templates for Later Use

Templates may be stored for later use and loaded by clicking the ‘Store Template’ button in the top left of the Generate Template menu. This will bring up another window allowing the current template to be saved with an identifying name. To save the current template, first enter the name in the ‘Save Template Name’ text box and click ‘Save Template’. To load a template as the current template, click on the template name (the row will become highlighted) and press ‘Load Template’. Note this will overwrite the coefficients in the current template. To delete a stored template, click on the template name and press ‘Delete Selected Template’.

Generation of Multiple Templates for Simultaneous Analysis

Use of up to three templates can be used simultaneously for event detection (See *Template to Use* for how to run analysis with multiple templates). The three templates can be accessed using the drop-down menu (‘Template 1 – 3’) in the Generate Template menu. The displayed template is referred to as the ‘currently selected template’ and will be used in analysis when this option is selected in *Analyze Events*. When fitting a template to the data or loaded from ‘Stored Templates’, the coefficients will be stored in the currently selected template (e.g. ‘Template 2’). The *Average Event* will always be fit using the currently selected template.

Refine Template

The *Refine Template* window is used for a) assessing the fit of an event template (Fig. 8) and b) generating a refined template by fitting to the average of all events detected with the initial template.

Clicking ‘Fit All Events’ detects events using the currently selected template. Detected events are displayed on the graph with a filled yellow circle at the peak and their baseline in blue. The average of all detected event is plot on the graph in red. Options to align by the ‘Rise half-width (half-way up the event rise), ‘Peak’ and ‘Baseline’ are provided in the ‘Alignment Method’ drop-down box. The process of fitting to the average event is identical to fitting in *Generate Template*.

The dropdown menu at the top of the window can be used to select the event detection method. If ‘Correlation’ or ‘Detection Cutoff’ is chosen, the desired cutoff value can be entered in the box next to the drop-down menu. Alternatively, if ‘Deconvolution’ is selected, the cutoff can be adjusted by clicking ‘Options’ (detailed below).

Assessing Event Detection

Clicking the ‘Show (detection method name) button will show the detection measure overlaid on the plot. The cutoff value is shown as a horizontal line. This option is very useful for assisting in choosing a cutoff value, as its position in relation to the detected peaks can be observed. The detection measure units are scaled to fit to the plot for visualization purposes and cannot be interpreted directly.

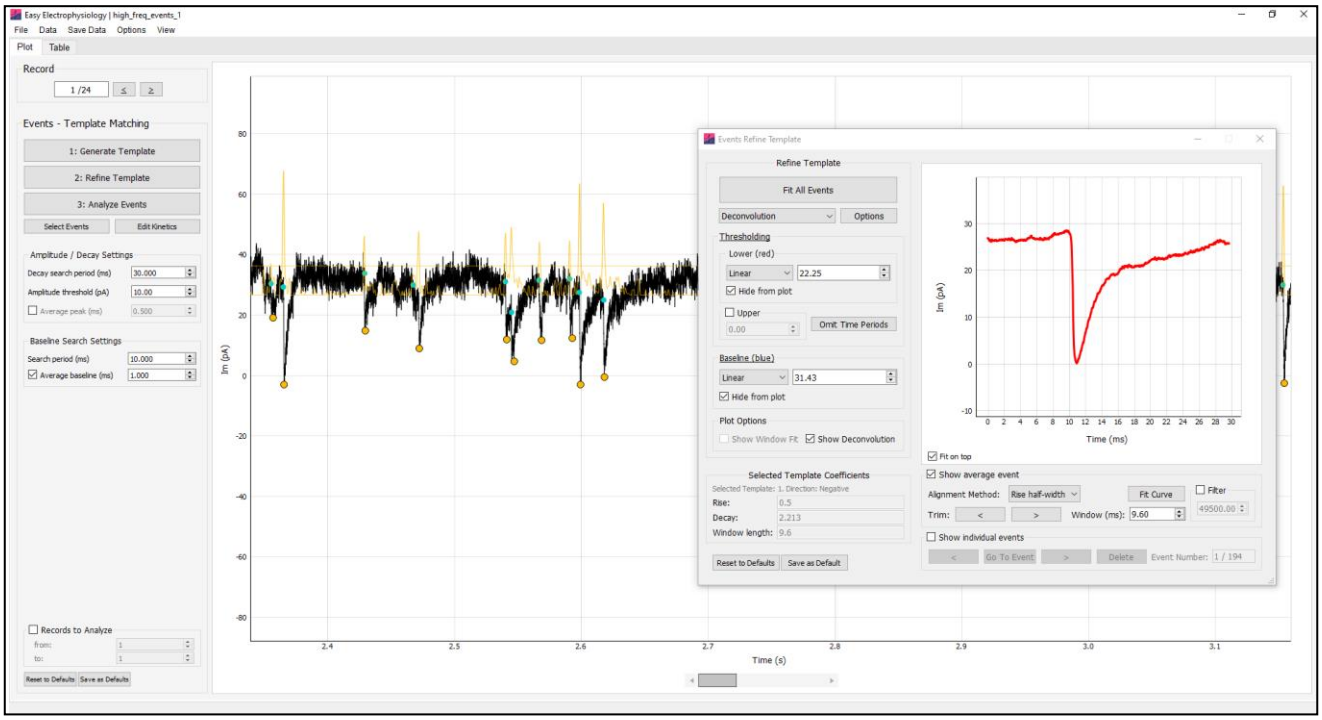

Figure 8. Refine Template. Refine template analysis. The detection measure (in this case the filtered deconvolution) is displayed on the graph in orange. The current cutoff value is displayed as a horizontal orange line. The average detected event is also displayed in red. This average event can be fit by clicking the ‘Fit Curve’ button (in this case, it would first be necessary to align the first sample to fit to the left edge of the plot with the ‘Trim’ buttons).

For correlation and detection criterion methods, the ‘Show Window Fit’ button will display the best-fit biexponential overlaid on the data of the main plot. When zoomed in, scrolling will move the biexponential along the data, allowing inspection of the fit at every timepoint. For this, it is convenient to be in to *two button* by right-clicking clicking on the plot, ‘Mouse Mode’ > ‘Two Button (Pan Left, Zoom Right)’.

Deconvolution Options

In 'Deconvolution Options' (Fig. 9), a cutoff value can be specified in the *'Standard Dev' box.* This will select the standard deviation of a distribution created by the amplitude of the cutoff measure at each sample. For example, choosing 3.5 will select events that have a detection measure only > 3.5 standard deviations larger than the mean (default value = 3.5).

Additionally, the cutoff measure undergoes bandpass filtering which can affect its shape. Default values are 1 for the *Low Cutoff (Hz)* and 200 for *High Cutoff (Hz).* Changing the high cutoff will influence the height and sharpness of the detection measure peaks and is useful to adjust when selecting optimal values for your data. The best way to choose values is to show the deconvolution on the plot (click *'Show Deconvolution'*) and visually inspect how it changes when the filter settings are adjusted and analysis re-run.

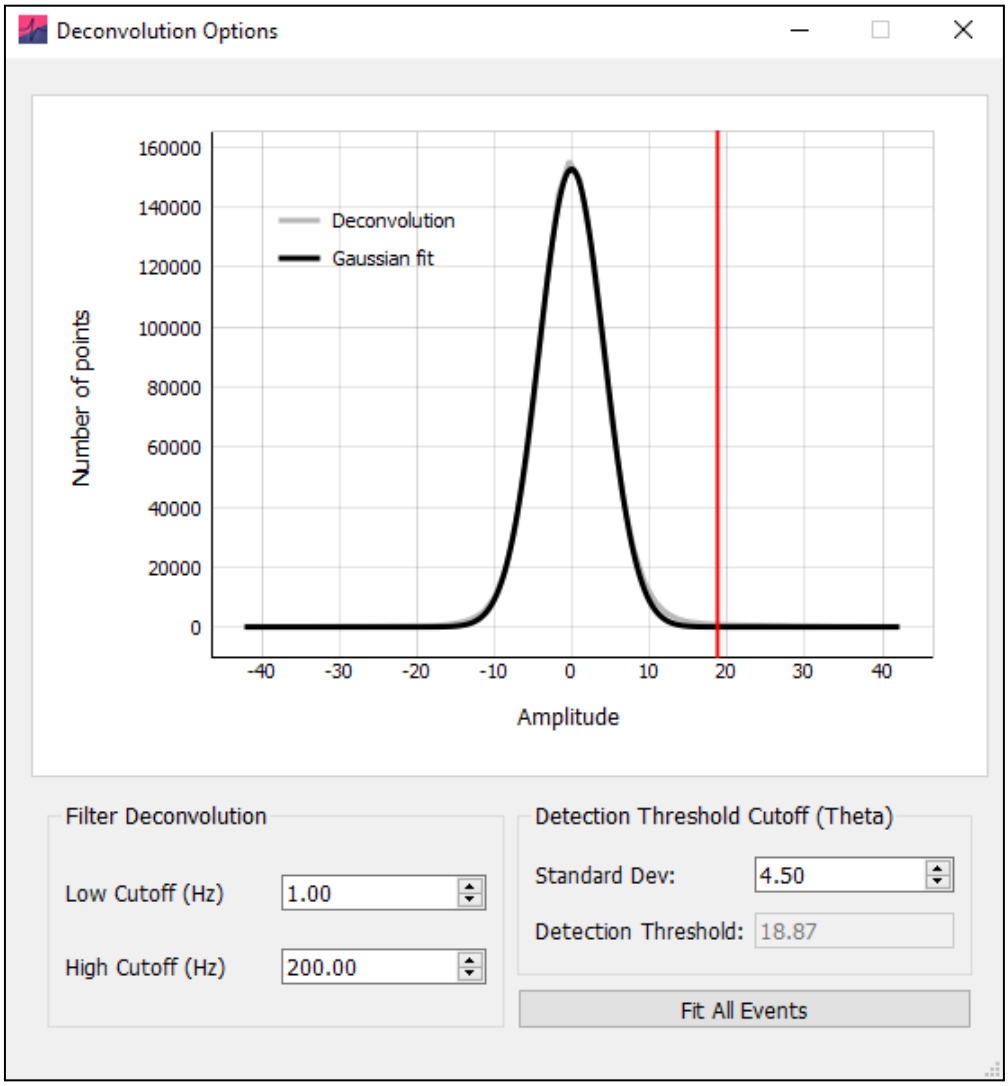

Figure 9. Deconvolution Options. A plot displays the amplitude histogram for all points in the deconvolution, fit with a Gaussian function. The cutoff value is defined in terms of the standard deviation of this distribution. Low and high cutoff for bandpass filtering of the deconvolution can also be specified here.

Events Analysis Options

Due to the large number of possible configurations for running event analysis, the options are split into three categories:

<u>Event Detection Window Options</u>: These options appear in the analysis windows when running *Events Template Refine, Events Template Analyze* (Fig. 10) and *Events Threshold Analyze* analysis. These options are related to the detection of events.

<u>Event Fitting Analysis Panel Options</u>: These options appear on the *Events – Template Matching* and *Events – Thresholding* analysis panels. These options are related to analysis of event parameters (e.g. amplitude threshold, peak smoothing).

<u>Events Options</u>: These options cover general event analysis settings for *Events – Template Matching*, *Events – Thresholding*, and all *Curve Fitting* biexponential event analysis. These options are found at 'Options' > 'Events Analysis – Misc. Options'.

The settings of each of these option categories can be saved / reset separately (see *Saving and Resetting Default Options* for details).

Event Detection Window Options

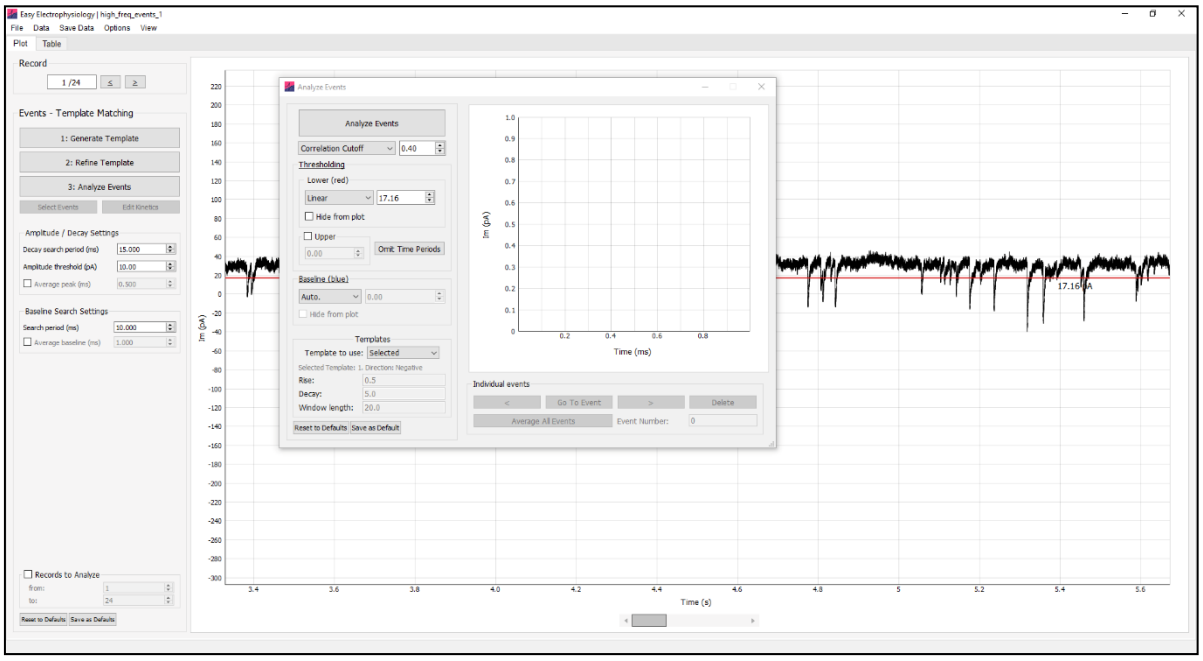

Figure 10. Event Detection Options. The Analyze Events window contains many options for customizing event detection. On the main plot the *lower threshold* line is displayed in *linear* mode. Any putative event that does not pass this threshold is excluded from analysis.

Lower Threshold

This is the minimum value that an event must pass to be considered within threshold, specified in pA. It is dependent on the direction of events (e.g. for a threshold of 5 pA, positive events must contain a datapoint > 5 pA, negative events must contain a datapoint < 5 pA. The *lower threshold* is displayed on the graph in red. Using the dropdown menu, it can be specified as:

- *Linear* – A straight horizontal line across the trace. This can be moved up and down with the mouse, or the desired threshold value typed into the input box.
- *Curve* – A polynomial is fit to the data and used as the threshold. This can be moved up or down by typing the desired offset into the input box. The order of polynomial fit can be changed by clicking 'Options' > 'Events Analysis – Misc. Options' > 'Dynamic Curve polynomial order' (under 'other options'). Note fitting of higher-order polynomials slow performance.
- *Drawn* – Setting the drop-down menu to 'Draw' then clicking the 'Draw Threshold' button will enter threshold drawing mode. The cursor will change to a pointed hand; clicking the graph will begin drawing a baseline. The first click will draw a horizontal line from the left edge of the record to the cursor. Subsequent clicks will draw a straight line from the previous point to the cursor. To finish drawing the threshold, click the right edge of the graph past the final plot datapoint. This mode is particularly useful when there are large, sudden changes in the data's baseline.

  Note that if a new file is loaded which is longer than the currently drawn threshold, the threshold will be extended as a straight line to the new end timepoint.
- *RMS* – The root mean squared error (RMS) of the data (multiplied by some scalar *n)* can also be used as a minimum threshold. The minimum threshold is the baseline minus the *n* x RMS (for negative events, baseline plus *n* x RMS for positive events) and is shown on the plot in red. The baseline that the RMS is added to is determined by the choice of event baseline (if 'Auto.' is selected, the mean of the data will be used, separately for each record.).

  When selecting 'RMS' from the drop-down menu, clicking 'settings' will open a window allowing choice of settings for calculating the RMS. The multiple *n* to scale the RMS to (e.g. 2x RMS) can be typed into the input box. If 'All Data (Per Record)' is selected, the RMS will be calculated separately for each record, using the mean of the record as calculation baseline. Alternatively, selecting 'Within Selected Region' will allow calculation of the RMS within a movable region on the graph. Selecting the option 'Baseline used: mean' will use the mean of the data within this region as the calculation baseline for the RMS. Otherwise, if 'baseline used: current baseline' then

the currently selected baseline will be used as the calculation baseline during RMS calculation. In ‘baseline used: current baseline’ mode, for *Linear, Curve and Drawn* baselines, the RMS is calculated as the root mean square error between the baseline and data. For *Auto*, the calculation baseline is the mean of the entire record (not the data within the bounds).

Baseline Options (Analyze windows only)

The event baseline can detect automatically (‘Auto.’) or used as the first sample that the data crosses a pre-defined baseline value. The baseline datapoint acts as both the baseline Im value and the event foot (i.e. the timepoint at which the event starts).

For automatic detection, the baseline is first estimated per-event (see Legacy options). Next, the method of Jonas et al., (1993) (the intersection between the baseline and a straight line drawn through the 20-80 rise time) is used to shift the x-axis position of the baseline to the point of the event foot. Together, these methods result in robust simultaneous estimation of the event baseline and foot.

This baseline value can also be specified in a similar way as *lower threshold* (i.e. *linear*, *curve* or *drawn*). This will search the *baseline search region* for the first datapoint that has crossed this baseline threshold and set it as the baseline. If no datapoint crosses the baseline threshold within the search region, the closest value to the threshold is used.

Upper Threshold

The *upper threshold* is defined as the maximum value an event can pass before it is excluded from analysis (e.g. for a 100 pA threshold, any positive event > 100 pA or negative event < 100 pA will be excluded). This option can be specified by typing the absolute value in the input box. This is particularly useful to excluding large noise spikes e.g. stimulus artefacts.

Omit Time Periods

Omitting time periods from analysis can be useful for event recordings with interleaved stimulation. The ‘Omit Time Period’ button opens a table for inputting time periods to omit from analysis. Add the start time of the period to exclude in the first column and the end time in the second column. Multiple time periods to exclude can be input, one on each row.

Template to Use (Events – Template only)

Event detection may be run with up to three different templates simultaneously. The *Templates to Use* specifies the templates to use for the analysis (‘Selected’, ‘Template 1 and

2' or 'Template 1, 2 and 3'). See *Generation of Multiple Templates for Simultaneous Analysis* for details on generating the three templates. The *Average Event* and manually selected events are always fit using the currently selected template.

If set to 'Selected', the currently selected template (in *Generate* Template) is used. The coefficients of the currently selected template can be seen in directly underneath the drop-down menu. Otherwise, templates 1 and 2, or 1, 2 and 3 are used depending on the selected option.

Analysis with multiple templates is run in the same way as a normal analysis. On the plot, the events detected by different templates can be distinguished by the peak color (Template 1: red, Template 2: pale blue, Template 3: pale orange). In the results table, the column 'Table' holds the number of the template that was used to detect the event. To set the behavior in the case that the same event is detected by two different templates, see *Multi-template event assignment.*

Event Kinetics Overview

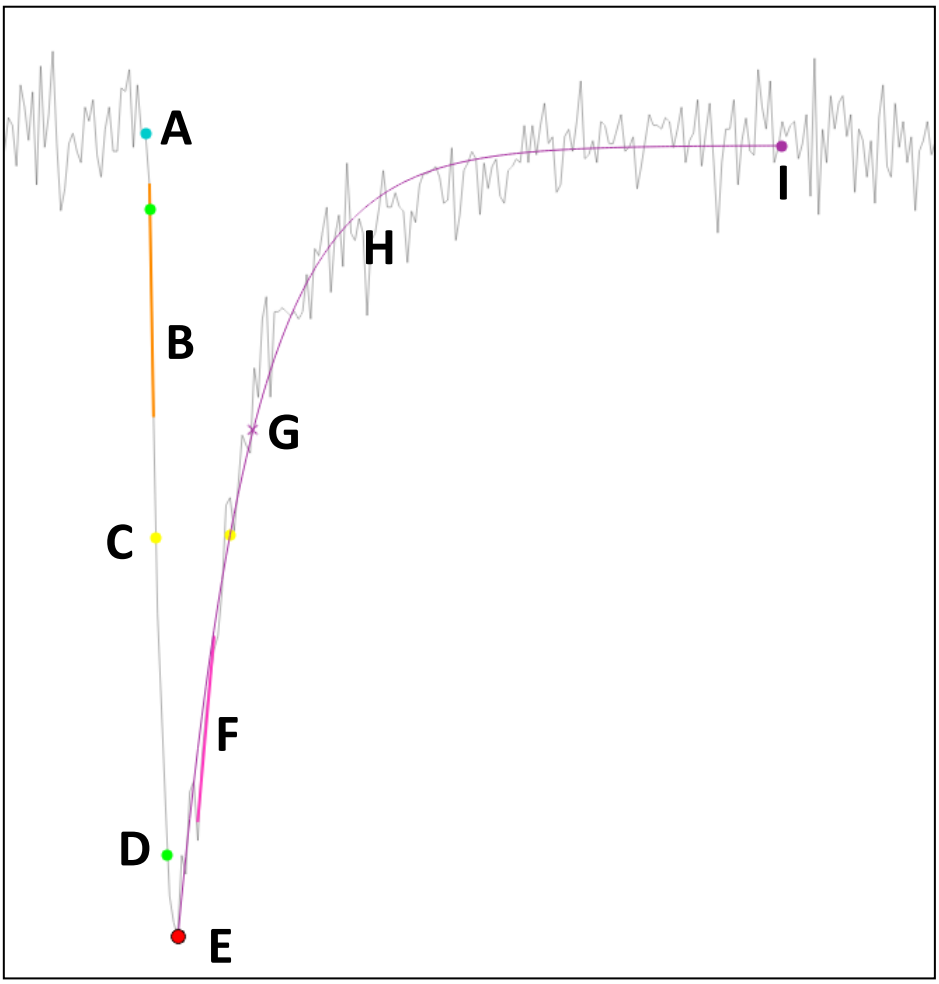

Figure 11. Event Kinetics. Analyzed event with 200 kHz interpolation for rise-time, decay-time and half-width.

A *Baseline* – The event baseline (filled blue circle) acts as both the baseline value and timepoint considered the start of the event. The baseline calculation method is set in *Baseline Options (Analyze windows only)* and modified with *Baseline – Search period (ms)* and *Average Baseline (ms).*

B & F – *Max Rise / Decay Slope* – The maximum slope on the rise or decay, calculated as a linear regression over n points (specified in *Calculate Max Rise / Decay Slope*).

C *Half-width* – The half-width is calculated as the full-width at half-maximum (FWHM). This is the time between two half-amplitude samples (on the rise and decay; displayed on the plot with filled yellow circles). The nearest samples to the 'true' half-amplitude are used to calculate the FWHM; as such *Interpolate to 200 kHz* can greatly improve accuracy. If a monoexponential curve is fit to the decay, or biexponential to the event, this will be used to find the decay, and rise and decay respectively for half-width calculation.

D *Rise-time* – The rise time is measured as the time between two timepoints (filled green circles), specified as the percentage of the event amplitude (peak minus baseline). These 'percentage cut-offs' can be changed using the *Rise-time Cutoffs* setting (10-90 rise-time is default, *shown*).

E *Peak* – The peak of the event can be smoothed using *Average Peak (ms).*

G *Decay %* – This is the time between the peak and the first sample at which the event has decayed to specified percentage of the event amplitude (default 37%). The sample closest to the true *Decay %* is used (displayed on the graph with a purple cross. If a monoexponential or biexponential curve is fit to the data, this will be used to find the decay time and half-width. If a curve is not fit, the nearest data-sample is used (smoothing the decay can improve accuracy; see *Set Decay Tau Range*).

The percentage is set with the *Decay %* setting. The period to search for the *Decay %* can be set using *Decay Search Period (ms).*

H *Decay Fit* – A monoexponential fit to the decay is displayed with a purple line. The tau of this fit is reported in the Events results table (if this fit is good, this result should closely match the *Decay %* when set to 37%). The period to search for the *Decay %* can be set using *Decay search period (ms)* and bounds on the fit exponential in *Set Decay Tau Range.*

I *Decay endpoint* – If a monoexponential decay is fit, the automatically-detected end-of-event is used as the fit endpoint (shown with a filled purple circle). See *Decay Search Period (ms)* for details.

Event Fitting Analysis Panel Options

Peak Direction (Events Threshold only)

Direction of the events to detect.

Local Maximum Period (ms)

The minimum period between local maxima (i.e. event peaks) in milliseconds. For example, if set to 1000 ms, detected events will be at least 1 second apart. Detected events that are closer than the local minimum period (i.e. 1 s in our example) will be excluded, with smallest events excluded first.

Decay Search Period (ms)

When fitting a monoexponential function to the event decay, the end of the event is estimated as the closest datapoint to the baseline (using a weighting and smoothing algorithm used to account for noise and complex waveforms). This end-of-event point is indicated with a closed purple circle on the graph. The monoexponential function is fit between the event peak and the event-of-event datapoint. Here, the *decay search period* indicates the period to search for the event-of-event datapoint.

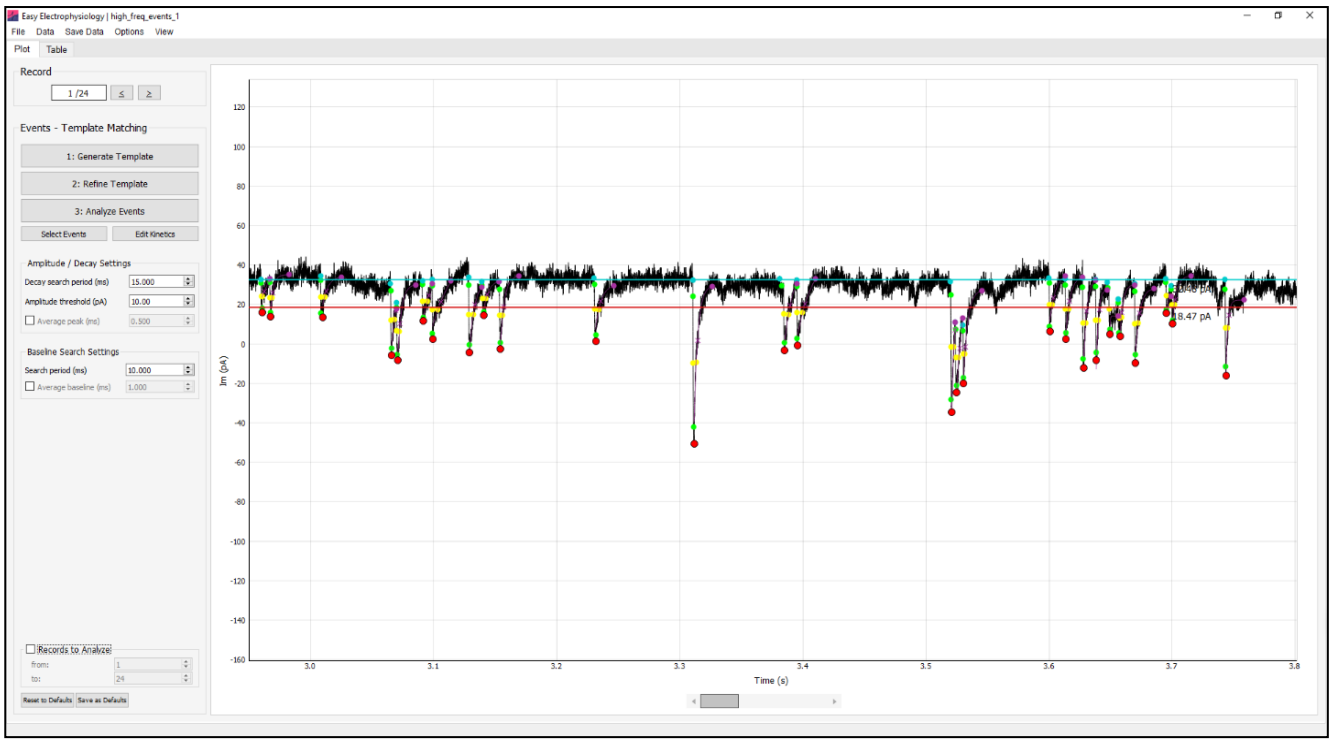

Figure 12. Event Detection Results. Detected events are displayed on the graph (peaks indicated by red filled circle). Double clicking on the event peak will discard the event. Also show is the *lower threshold* (*linear*; red) and *baseline (linear*, blue). The position of each events baseline and decay endpoint can also be adjusted in 'Edit Kinetics' mode. Additional options for event kinetics analysis are displayed on the analysis panel (*left*).

If a monoexponential curve is not fit to the decay, the *decay search period* defines the search region, starting at the event peak, for the *Decay* % value (as specified in 'Events Analysis – Misc. Options' > 'Decay %).

Amplitude Threshold (pA)

Sets the minimum amplitude (event peak minus baseline; absolute value) for detected events. Any events with an amplitude smaller than this threshold are excluded.

Average Peak (ms)

Period to smooth around the event peak. The intuitive method of peak averaging is to simply average around the peak detected from unsmoothed data. However, this method often detects the peak at off-center noise spikes (i.e. not centered on the natural peak of the event).

In Easy Electrophysiology, the event is first detected without smoothing. Then, the event is smoothed with a window of size specified with this *average peak* option. Finally, the new peak is determined from the smoothed data in a small region around the previously detected peak (±3 * *average peak*). This method simultaneously smooths the peak Im value as an average of surrounding points and adjusts the position to reduce noise bias.

AUC min. (pA ms)

Area Under Curve (AUC) threshold can be set to exclude putative events with small areas. Note this is the absolute value (as for negative events area is defined as negative area) (e.g. if set to 5, negative events with area -10 pA ms will be included, but -3 pA ms excluded).

Baseline – Search Period (ms)

The time period before the event peak to search for the baseline. When baseline detection method is set to 'Auto.' (in the *Analyze Events* window), this specifies the search period for automated baseline detection. For all other baseline options (*linear, curve, drawn*) this is the search period for the first datapoint that crosses the baseline (see *Baseline Options (Analyze windows only)*.

Average Baseline (ms)

The period over which to average the event baseline. First, the event baseline is detected with the selected method (*Auto., linear, curve, drawn*). Then, the baseline value is set to the average of a window before the baseline. *Average baseline* sets the size of this averaging window.

Events Options Window

Additional event detection and analysis options can be found by clicking ‘Options’ > ‘Events''. These options are shared between *Events – Template Matching, Events – Thresholding*, *Average Event* and *Curve Fitting* (biexponential event) analysis.

Decay Endpoint search method

This option selects the method used to calculate the endpoint of the event (i.e. when the event is considered to have finished). This is the period across which the decay % parameter will be searched, and act at the endpoint for monoexponentially and biexponential fits. A third, older event endpoint detection method is available in *Legacy Options.*

Entire Search Region, the event endpoint will be at the exact time after the peak specified in the *Decay search period (ms)* (e.g. if the decay search period (ms) is 30 ms, and the peak is detected at 100 ms, the decay endpoint will be at timepoint 130 ms). However, if another event occurs before this decay endpoint, the decay endpoint will be moved to one sample prior to the next event baseline.

First Baseline Cross, the first datapoint that crosses the baseline is used as the event endpoint. The data is smoothed slightly (3 samples) to avoid large noise spikes that may occur during the decay period. If no sample in the search period crosses the baseline, the datapoint closest to the baseline is used. Similar to ‘Entire Search Region’, if the next events baseline is before the detected decay endpoint, the decay endpoint will be moved

Interpolate to 200 kHz

Interpolate the event to 200 kHz prior to decay-time, rise-time and half-width analysis (linear interpolation). First, the event peak, baseline and event end-point is calculated. Then the event (i.e. the period between the baseline and event end-point) is interpolated before decay, rise and half-width analysis.

Rise-time Cutoffs

Specifies the cutoffs for rise-time calculation (by default, the 10-90 rise-time is used).

Decay %

This option specifies the value used in the *Decay %* decay detection method. The default value is 37%.

Calculate Max Rise / Decay Slope

The maximum rise and decay for each event will be calculated if this option is on. It is not calculated by default as it may increase analysis time, particularly if the slope is calculated over many samples.

The maximum slope is calculated as a regression over n points. The number of samples to calculate the regression over can be specified separately for the rise and decay. If the number of samples to calculate the regression is larger than the number of samples in the rise / decay, this parameter will not be calculated for the event.

In some cases, noise spikes can bias the calculation of the maximum slope. As such, an option to smooth the rise / decay before calculation of maximum slope is provided.

Finally, it is recommended to select 'Always use baseline crossing as max slope search endpoint'. This ensures that the algorithm will stop searching for the maximum slope close to the point when the decay returns to baseline. Otherwise if this is unchecked and the *Decay endpoint search method* is set to 'Entire Search Region', the entire *Decay search period (ms)* will be searched for regression, leading to poorer accuracy and increased analysis time.

Event Curve Fitting

Event curve fitting provides options for fitting a biexponential function to the entire event ('Biexponential'), monoexponential function to the decay period ('Monoexponential to Decay') or not fitting ('Don't Fit Event') (Fig. 13).

If a biexponential is fit, half-width and decay % will be calculated on the fit. If a monoexponential is fit, the decay part of the half-width and decay % will be calculated from the fit. If 'Don't Fit Event' is selected it is recommended to slightly smooth the decay period for calculation of decay % and half-width to reduce the influence of noise spikes that may occur after the true decay mid-point (Fig 13C & D).

Options are provided to ensure only optimal fits are included in the results. Two measures to assess the fit are provided as a basis to exclude or attempt to improve the fit – the R^2 or coefficient of the fit function.

If excluding based on R^2 or bounds, any event with a fit lower than the R^2 value or outside of the coefficient bounds will be excluded from the results. Alternatively, it is possible to try and improve the fit by adjusting the first sample of the data the function is fit to. Both

biexponential and monoexponential fitting are extremely sensitive to the position of the event foot and peak, respectively. For monoexponential fits, new-start-points both to the right and left of the peak will be attempted. For biexponential fits, samples only to the right of the baseline will be attempted, as choosing samples further from the natural foot of the event typically worsens biexponential fitting. If improvement is not sufficient such that the fits are within threshold, the event will be excluded.

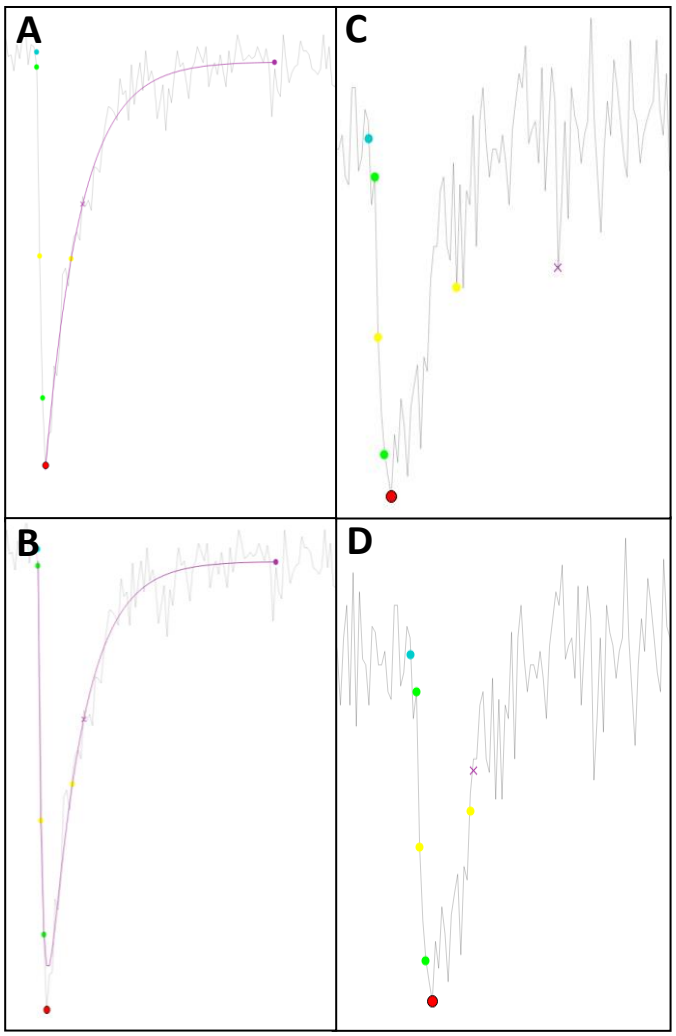

Figure 13. Event Fitting. **A)** Monoexponential fit to event decay. **B)** Biexponential fit to event. **C)** No event fit. The decay % and half-width parameters are erroneously fit samples in noise spikes close in Im value to the true parameters**. D)** slight smoothing of the decay period reduces the influence of noise spikes and ensures optimal parameter calculation.

For biexponential fits, the starting estimate for the coefficients will be taken from the template used to detect the event in *Events Template* analysis. Otherwise, for *Events Thresholding* and *Curve Fitting* analysis the default coefficients specified in *Template Matching – Initial Coefficients* will be used.

These options can also be useful for excluding erroneously detected events. The most typical cause of poor fits is that the putative event is in fact a noise spike.

Other Options

- *Don't threshold manually selected events* – by default, when manually selecting events the *Amplitude Threshold (pA), Omit Start Times, Threshold – Lower* and *Threshold* – Upper are applied. If his option is off, these exclusion criteria will not be applied for manually selected events.
- *Dynamic curve polynomial order* – specifies the order of the polynomial used in *Lower Threshold* and *Baseline* curves. Fitting high-order polynomials will increase computation time.
- *Multi-template event assignment* – If set to 'To best fitting template', the template with the highest value of the detection measure (see *Events – Template Matching)* at the start of event estimated from the template is used. If set to 'To first template used 1 > 2 > 3', the first template that detects the event (in the order 1, 2, 3) will be used.
- *Template Matching – Initial Coefficients* specifies the default coefficients for the *Events – Template Matching* template generation, *Curve Fitting* (biexponential event), and biexponential fitting to events in *Events - Thresholding*.

Manual Selection / Rejection of Events and Kinetics Editing

Easy Electrophysiology offers a high degree of flexibility when it comes to manual adjustment of event detection and analysis. Erroneously detected events may be discarded, or undetected events manually selected. Furthermore, the position of event baseline / foot and endpoint can be easily adjusted.

Discard and Select Events

To discard detected events, click twice on the filled circle at the peak of an event. The first click will prime the event for deletion (the circle will turn blue) and the second click will delete the event from analysis.

To manually select an event, click the 'Manually Select Events' option on the Analysis Panel (for both *Events – Template Matching* and *Events – Thresholding*). The cursor will change to a crosshair and click & dragging on the graph will draw a circle. The peak (minimum for negative events, maximum for positive events) in this region will be taken as the new event peak. All current settings (e.g., smoothing, search periods, though see *Don't threshold manually selected events*) apply to manually selected events. However, the fitting method must be the same for every event in an analysis, as such the fit of a manually selected event will always match the analysis used

Edit Kinetics

The position of the event baseline and decay endpoint can be manually adjusted. Clicking 'Edit Kinetics' will enter Edit Kinetics mode, indicated by the cursor changing to a pointing hand. To edit a kinetic in this mode, first click on the baseline (light blue circle) or event endpoint (purple circle). Next, the cursor will change to a crosshair. Click on the desired position for the kinetic, and the kinetic will be moved and the other event parameters re-analyzed.

Grouping Detected Events

Event grouping makes it possible to select subsets of detected events and restrict results display to these subsets (or, excluding a subset of events). To group selected events following event detection, click the peak of an event holding the group number you would like to assign to the event on the keyboard (Fig.14). The group number can range from 1 to 5.

To selectively view grouped events on the main 'Table' tab table, click 'Analysis' > 'Table Options'. In the *Show Events by Group* section, the group to selectively display can be chosen from the drop-down menu. To display all events excluding the selected group, click the 'Exclude group' button.

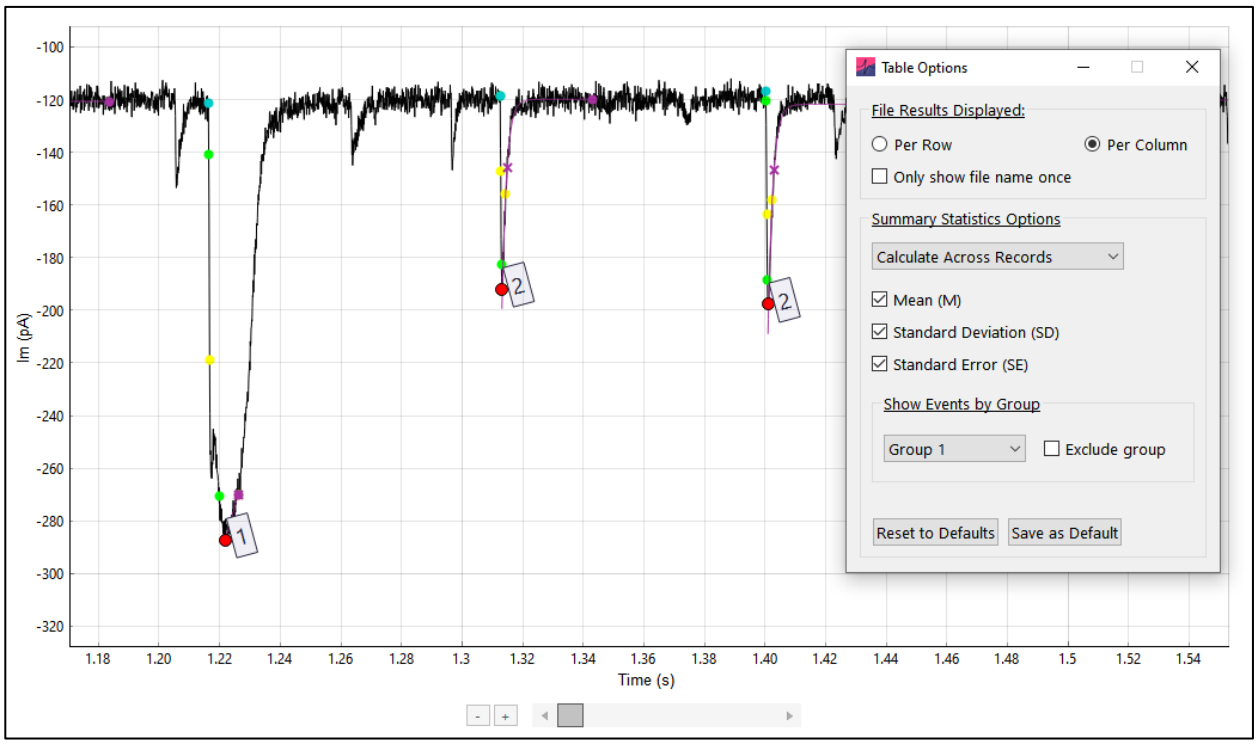

Figure 14. Grouping Detected Events. Detected events can be grouped by clicking the peak of the event while holding down the group number to assign on the keyboard. In this instance, the large event has been assigned 'Group 1' and the other events 'Group 2'. The groups can be displayed on the main plot by switching the 'Options' > 'Table Options'

Results

Events Overlay

The *Events Overlay* window displays all detected events on the same plot, providing a convenient way to find and remove outliers.

Events Overlay can be opened (after event detection has been performed) by clicking the 'Overlay' button on the 'Events Analyze' window. All detected events are shown overlaid on the plot in black (Fig. 15). The display of the overlaid events can be adjusted with the *Alignment method* and *Window size (ms)* settings. *Alignment method* determines the point of the event used for alignment (e.g. the event peak). *Window size (ms)* controls the size of the window used to display the events (the displayed period prior to the peak is fixed, this setting controls the window from the peak to the right edge of the data).

The Selected Event and Average Event

In the *Events Overlay* window, the currently selected individual event is shown in blue. When the window first opens, the individual event mode is set to *Selected*. Clicking the left or right arrow buttons in *Selected* mode will scroll through all events in numerical order. Changing the selected event through the *Events Overlay* window also scrolls through the individual events shown on the *Events Analysis* window, and vice-versa.

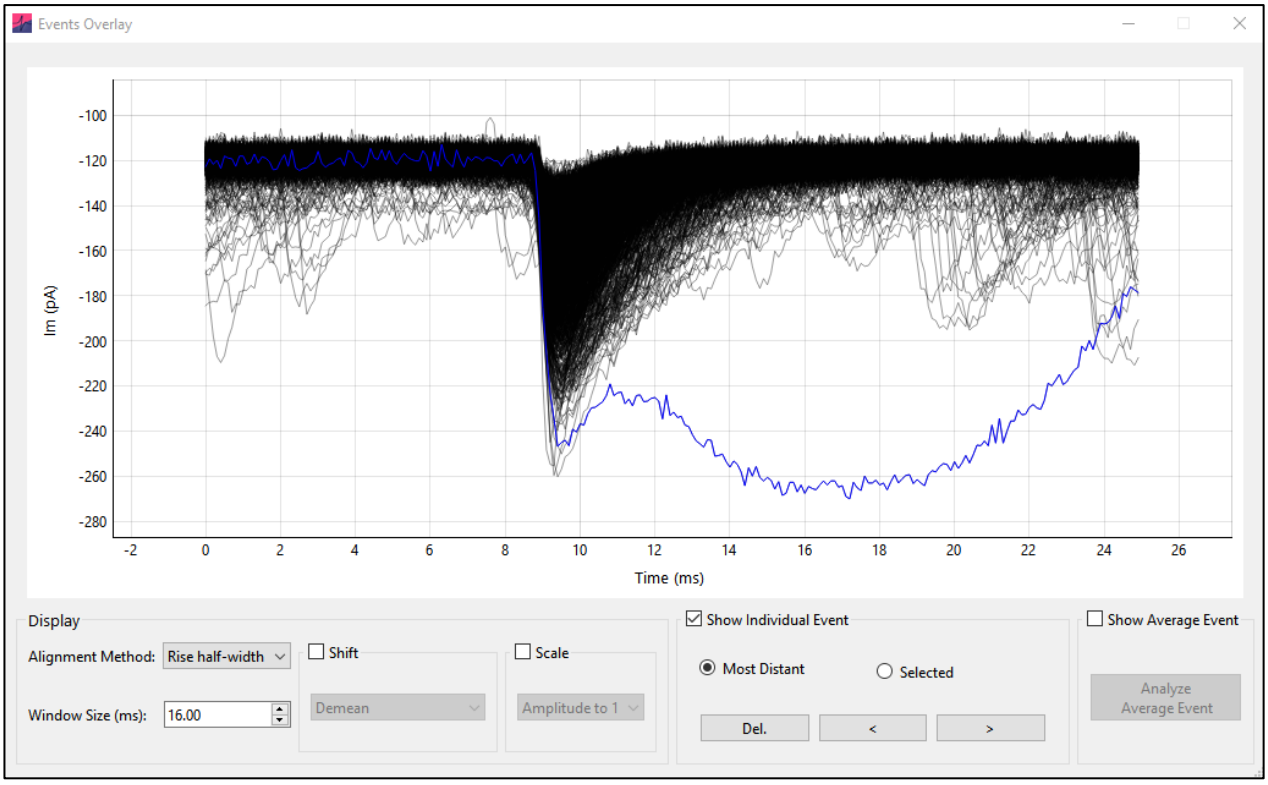

Figure 15. Events Overlay window. This window shows all detected events overlaid. In blue, the currently selected event is visible, in *Most Distant* mode. The selected event has the largest distance of all detected events (distance defined as the L2 norm from the average of all events).

Alternatively, individual events can be scrolled through in *Most Distant* mode. The selected event will become the most distant event; here, distance is measured as the L2 norm from the average of all events. Clicking the left or right buttons will scroll through selected events based on their distance (in the order most to least distant)—this can be a very useful way to detect and remove outlier events.

Further, individual events can be selected by clicking on them directly on the plot.

To remove an event from the analysis the *Del.* Button can be clicked on the *Events Overlay* window or the on the 'Delete' keyboard button pressed. The currently selected event will be removed from all analyses.

The average of all detected events can be displayed instead of the currently selected event by clicking 'Show Average Event'. Then, clicking 'Analyze Average Event' will open the analyze event dialog window, allowing analysis of the average event kinetics. This provides the possibility to shift and scale individual events prior to analysis of the average event.

Shift and Scale

The *shift* and *scale* options provide the options to shift (add a constant) and scale (multiply by a constant) the individual events prior to presentation. When shift and scale operations are applied to data, scale operations are applied prior to shift operations.

Shift

<u>Demean:</u> The mean of the event is subtracted (so the mean of the event becomes zero).

<u>Aligned param to zero:</u> The parameter used to align the events (e.g. rise half-width) is subtracted from the event (so that the aligned parameter value becomes zero).

<u>First sample to zero:</u> The first datapoint value of the event is subtracted (so the first sample of the event is zero).

Scale

<u>Amplitude to 1:</u> The event is multiplied by a constant such that the amplitude (from most negative to most positive sample) becomes 1.

<u>Std. Dev. To 1:</u> The event is multiplied by a constant such that the standard deviation of the event becomes 1.

Note that the shift and scale operations are identical to those in the *Average Event* window (below).

Average Event

Analysis

Following events analysis an average event may be generated and analyzed (Fig. 16). To an average event, click the 'Average All Events' button in the *Analyze Events* popup window (for both *Template Matching* and *Thresholding* analysis).

Generate an Average Event

To generate an average event, a point on each event is chosen to align ('peak', 'rise half-width' or 'baseline'). The data around this point is windowed and displayed in the *Average Events* window. If a window around an event extends further than the limits of the data, the event is excluded from analysis. This means that events to the extreme edge of the record are not included (e.g. if an event peak is at 10 ms, but the window size is 20 ms, the event will not be included in the average).

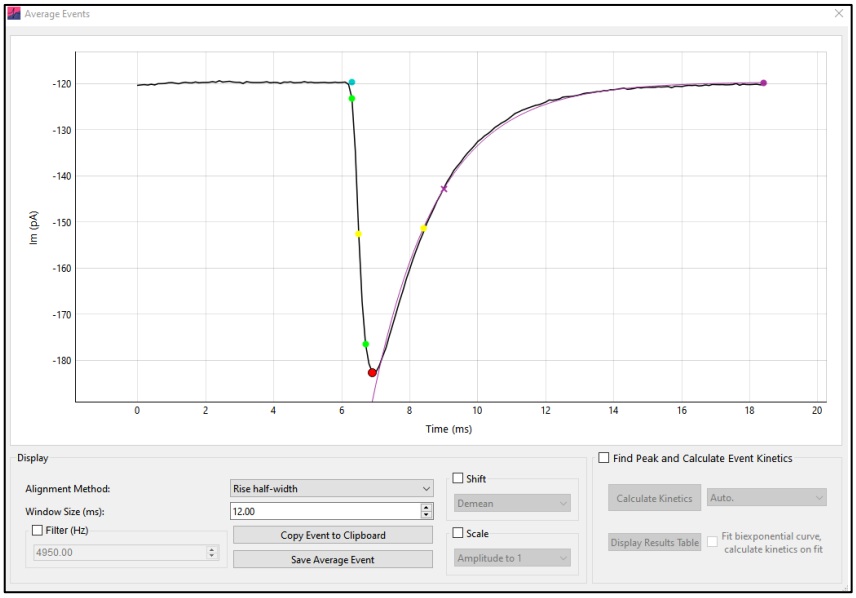

Figure 16. Average Event Analysis. Following events analysis, an average event can be generated and kinetics calculated. The average event Im and time data can be copied to clipboard or saved to file, and analysis results displayed in a popup table.

Click *Copy Event to Clipboard* and *Save Average Event* to copy or save (to .xlsx or .csv) the displayed data, respectively.

The averaged event can be filtered prior to analysis with the *Filter (Hz)* box (8th order Bessel). The averaged event can also be normalized (amplitude scaled to 1 and mean subtracted).

Analyze the Average Event

Click the 'Find Peak and Calculate Event Kinetics' button to start *Average Event* analysis. This will bring up a region on the plot that defines the analysis space.

The dropdown box to the right of the ‘Calculate Kinetics’ button can be used to set the baseline detection method. Set ‘Auto.’ for automatic baseline detection, or ‘Use Baseline’ to manually set the baseline. If use baseline is selected, a baseline region will appear and can be moved to set the baseline for the data (This is analogous to the *linear* baseline during event detection).

To fit a biexponential function to the average event and calculate the kinetics on the function, select ‘Fit biexponential curve, calculate kinetics on fit’. The region displayed on the graph can be used to define the fit start and end points.

The results of average event analysis can be viewed by clicking ‘Display Results Table’ and copied to the clipboard with the mouse / keyboard.

Saving Analysis Results

Events analysis (all detected events, plots and tabular results) can be saved for later use. To save the current analysis, click ‘Save Data’ and ‘Save Events Analysis’. The events analysis is stored as a .json file.

To re-load the events analysis, click ‘Save Data’ > ‘Load Events Analysis’ and select the previously saved JSON file. Note that the file name, number of records and number of samples of the data file are saved with the analysis results to ensure the events analysis match the recording file. As such, the recording filename should not be changed in between saving and loading events results.

Cumulative Probability & Histograms

Frequency data (as Histograms or Cumulative Probabilities) for all analyzed parameters can be calculated by selecting the check boxes on the Events ‘Table’ tab.

To specify how the frequency data is calculated click ‘Options’ > ‘Events’. In the ‘Frequency Data’ section it is possible to select ‘Cumulative Probability’ or ‘Histogram’ format. Clicking ‘More Options’ will provide options for binning. These options will come into effect from the next analysis run (that is, re-run the analysis for the new options to take effect).

In the ‘More Options’ window, the binning method can be selected from the top drop-down menu. If ‘Auto.’, the maximum of the Sturges or Freedman Diaconis Estimation methods are used, as implemented in the NumPy method *histogram_bin_edges().* In some instances,

depending on the scaling and units of the data, this will erroneously return a very large number of bins – in this case the square root method is used instead.

Alternatively, the number of bins can be set manually ('Custom Bin number'), the size of the bins in the units of the relevant parameter can be set ('Custom Bin Size') or the number of bins can be a function of the total number of events ('total number of events divided by'). Finally, the bin edges can be chosen as 'Bin Centre', 'Left Edge' or 'Right Edge' from the 'X axis display' drop-down menu (e.g. if the bins are 0-1, 1-2, 2-3, the X axis values will be Bin Centre: 0.5, 1.5, 2.5, Left Edge: 0, 1, 2, Right Edge: 1, 2, 3).

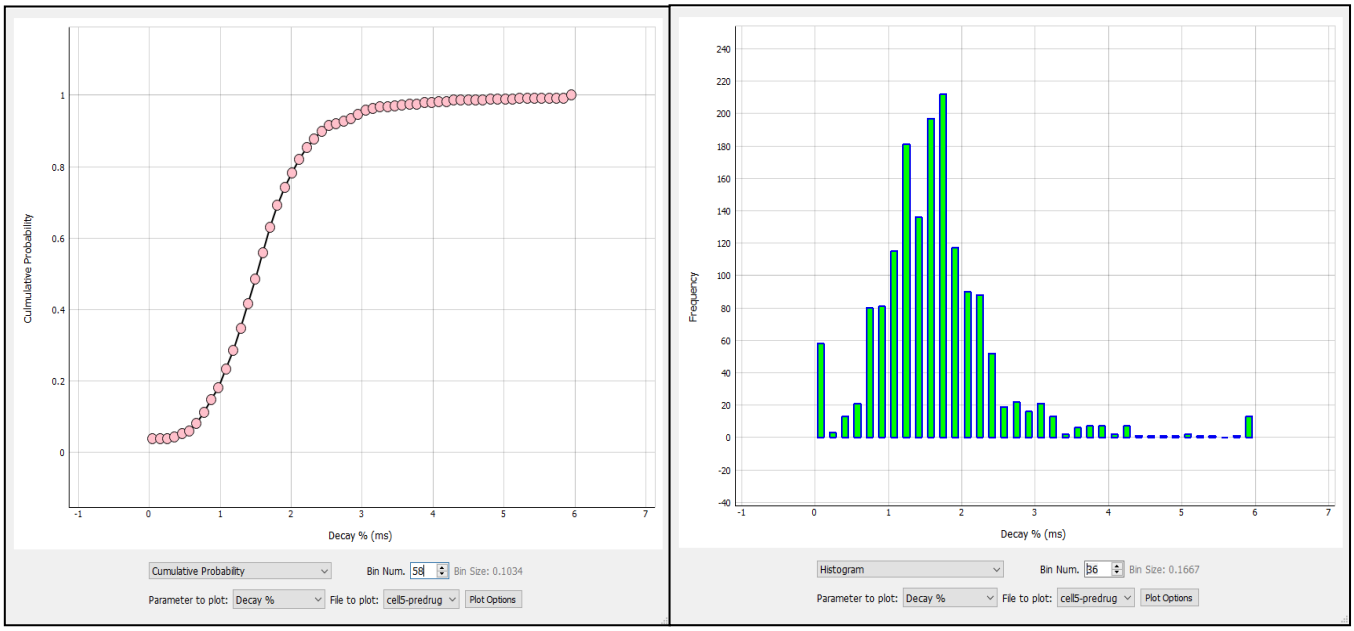

Figure 17. Events Frequency Plots. Cumulative probability (*left*) and histogram (*right*) of decay % frequency data. The plots are highly customizable, with options to change line and fill color, line thickness and marker size, x axis limits and grid lines on / off.

Note that when calculating inter-event interval for multi-record files, the ISI between the last event of a record and first event of the following record will be ignored.

By default, frequency data is sorted on the table by value (e.g. smallest to largest amplitude). To sort by event number, right click the table header and select 'Sort by Event Num.'. At the bottom of the result table, the summary statistics for each column are shown.

Frequency Data Plots

On the table tab, clicking 'Plot Frequency Data' will bring up the events frequency plots (Fig. 17). Note that changing any of these options only changes the current display of the plot and not how the data is binned and presented in the table.

The data can be displayed as a histogram or cumulative probability plot, and any event parameter selected for display. If analysis is run in Batch Mode, any analyzed file can be selected for display. Finally, clicking ‘Plot Options’ allows customization of the plot, including removing the gridlines, setting marker and line color, size and X axis limits.

8. Action Potential Counting

Action Potential Counting analysis provides automatic detection of action potentials (APs) and calculation of the rheobase, first-spike latency, inter-spike interval and spike-frequency accommodation.

Run Spike Counting Analysis

Clicking the 'Count AP's button on the *Action Potential Counting* analysis panel will run the analysis (Fig. 18). By default, every record in the file is analyzed. To select specific records to analyze set the 'Records to Analyze' box at the bottom of the analysis panel.

To restrict the analysis to a specific time period, click 'Set Bounds' (Fig. 19). To change the spike-detection algorithm, use the drop-down menu in the 'Thresholding' box on the analysis panel. More detail on these detection algorithms is given below.

To calculate the rheobase, Im input is required. This can be initialized by clicking the 'Im' checkbox. See the *Input Resistance* section for more detail on Im measurement.

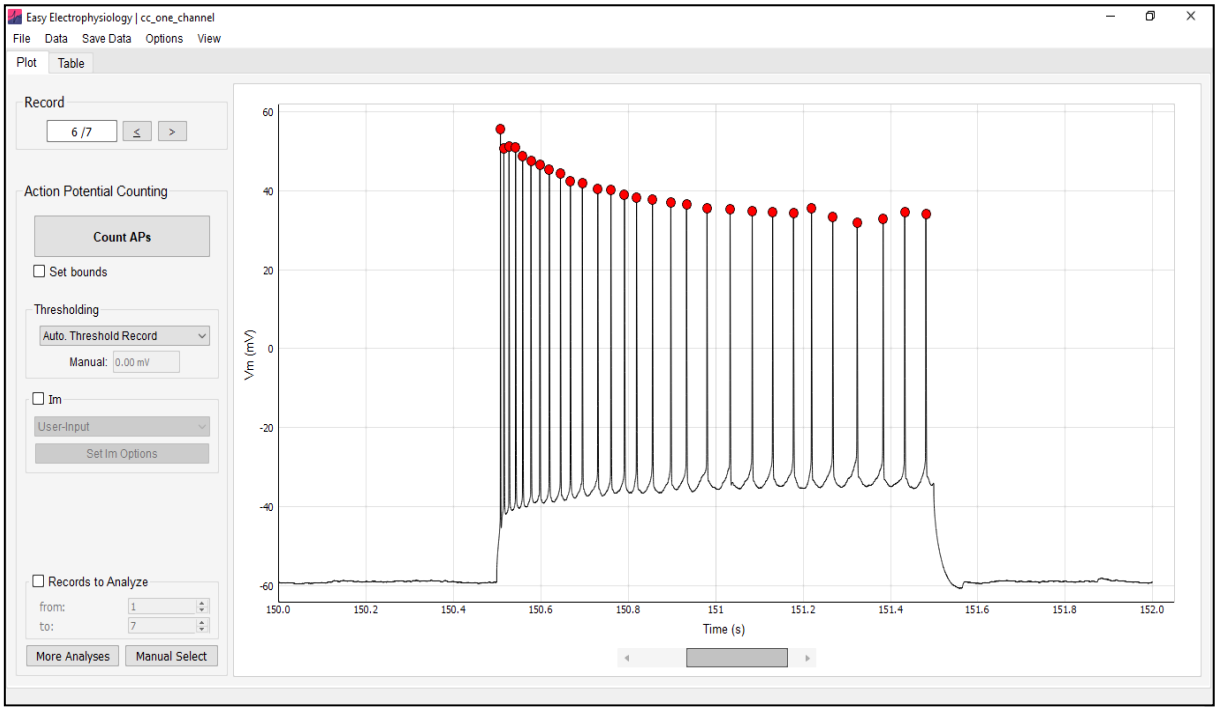

Figure 18. Action Potential Counting. Detected APs are indicated with a red filled circle at their peaks. Double clicking the AP peak will discard the AP from analysis. Undetected APs can be manually selected using the 'Manual Select' button (*bottom left*).

Spike Detection Methods

Three spike-detection algorithms are provided – *Auto-threshold Record*, *Auto-threshold Spike* and *Manual Threshold* which differ in their sensitivity and approach.

Automatic spike detection uses four parameters to detect spikes:

i. *Spike amplitude (mV)* – minimum amplitude of spike to be considered an action potential.
ii. *Positive first derivative (mV/ms)* – this captures the rising phase of the action potential. Spikes must exceed this threshold to be considered an action potential.
iii. *Negative first derivative (mV/ms)* – this captures the repolarizing phase of the action potential. This threshold criterion will only be applied if the positive first derivative threshold is passed within a user-defined time period (iv. Width).
iv. *Width (ms)* – Period to search for a sample crossing the negative first derivative threshold after the positive first derivative threshold is crossed.

These parameters can be adjusted through the 'Options' > 'Action Potential Counting Options' menu.

Auto-Threshold Spike

The option detects action potentials using the threshold parameters described above. This option is useful for dynamic spike waveforms and instances where non-linear trends in the peak amplitudes preclude use of a straight-line threshold.

Auto-Threshold Record

Auto-threshold record Fig. 18). provides a robust alternative to individual action potential detection, averaging across all detected spikes to generate an absolute threshold.

First, an *Auto-Threshold Spike* approach is used to detect action potentials. Then, the median amplitude mid-point Vm is calculated from all detected action potentials. This is used as a straight-line threshold over which any spike crossing is considered an action potential.

This method reduces the number of erroneously detected action potentials, particularly in noisy data as edge-case spikes are downweighed. This is the default setting for spike detection.

Manual Threshold

The *Manual Threshold* method is the simplest – if selected, a red threshold bar will appear on the plot (Fig. 19). Any spike which rises above then falls below this threshold will be counted as an action potential.

Manual Spike Selection / Editing

Following AP counting, detected events may be discarded or undetected events manually selected.

To select an AP, click the 'Manual Select' button on the *Action Potential Counting* panel. The cursor will change to a crosshair and click / dragging on the graph will draw a circle allowing selection of an AP peak. To allow maximum flexibility, these selections are not thresholded and the maximum Vm in the selected region will be taken as the new spike peak.

To discard detected APs, click twice on the filled red circle at the peak of an AP. The first click will prime the AP for deletion (the peak circle will turn blue) and the second click will delete the event from analysis.

Manual editing of detected action potentials will update all existing results (e.g. rheobase, first-spike latency, mean inter-spike interval, spike-frequency accommodation).

Action Potential Counting Options

Rheobase

Rheobase refers to the smallest current injection that elicits an action potential. Rheobase analysis is enabled when the Im checkbox is active (see *Input Resistance* for details on Im calculation in current-clamp mode).

The rheobase can be calculated as:

- *Record* – Im injection for the record at which the first AP is detected.
- *Exact* – Im at the exact time-point of the first detected AP peak.

First Spike Latency

The first spike latency is defined as the time from the start of the Im injection to the peak of the first action potential. If selected, users will be prompted to input the time (in seconds) that

the Im injection starts. First spike latency will be calculated for all records that contains action potentials.

Mean Inter-Spike Interval

The mean difference between the times of consecutive action potential peaks (i.e. the inter-spike-interval, ISI).

Spike Frequency Accommodation

Spike frequency accommodation (SFA) refers to a reduction in firing rate during a prolonged current injection. Two methods of SFA measurement are provided:

- *Divisor Method* – The first ISI of the spike train divided by the final ISI in the spike train.
- *Local Variance* – Introduced in Shinomoto et al., (2003), this measure captures the degree of irregularity of neuronal firing from the local variation of inter-spike intervals across the spike train (see *Appendix I*).

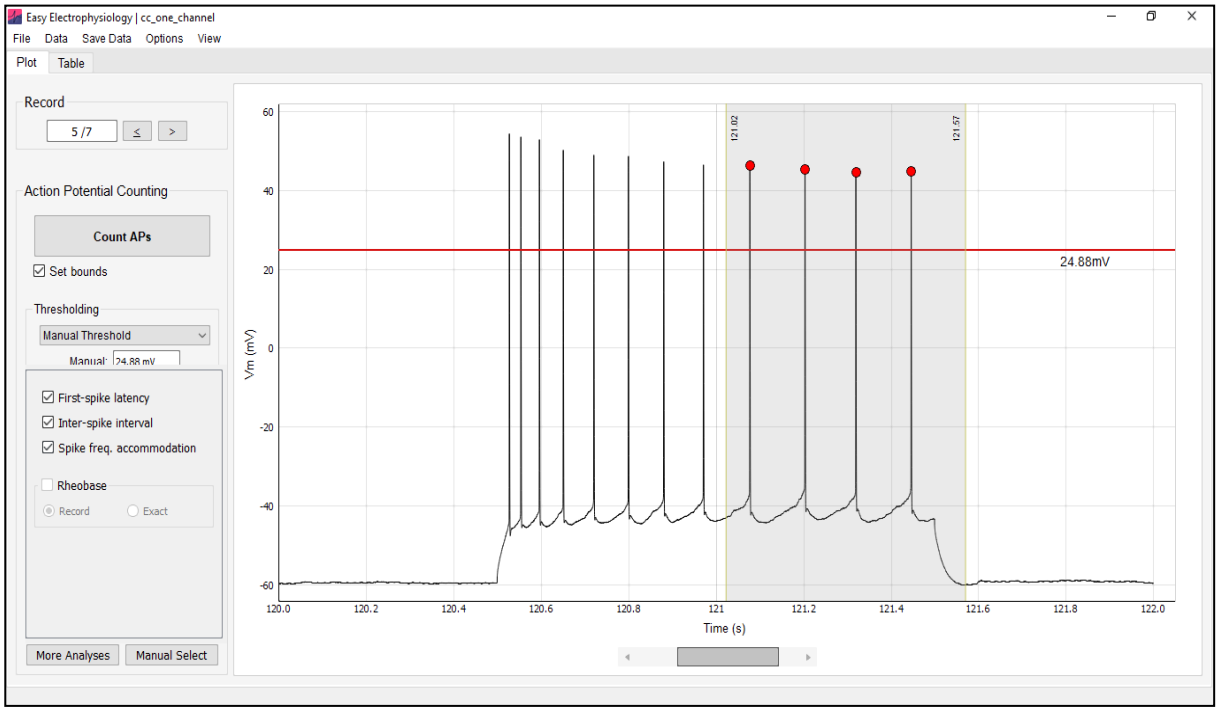

Figure 19. Action Potential Counting Options. *Action Potential Counting* analysis with a *manual threshold* (red line) and analysis restricted to a select region ('Set Bounds'). 'More Analysis' includes options for *first-spike latency*, *inter-spike interval* and *spike frequency accommodation*.

9. Input Resistance

Input Resistance analysis provides a convenient interface for measuring the input resistance and voltage sag / hump in current-clamp mode.

To calculate input resistance the Vm change (ΔVm) and Im change (ΔIm) must be determined from the data (in the case of ΔIm these can also be input manually). The input resistance is calculated as the slope of a linear ordinary least squares fit, X = ΔIm, y = ΔVm (calculated with SciPy's *linregress,* modelling the intercept*).* If only one record is analyzed, Ri is calculated as V / I.

For input resistance calculations, voltage is specified in mV and current in nA so that input resistance is in MOhms.

Options for analyzing the voltage sag (for negative Im injections) or hump (for positive Im injections) are also provided.

Calculating ΔVm and ΔIm

Two methods to calculate ΔVm and *ΔIm* from the data are provided:

From Data (Bounds)

The ΔVm and ΔIm are calculated as the difference between the data calculated within the 'measure and 'baseline' regions (Fig. 20). Baseline and measure regions are automatically displayed on the Vm and Im graphs. The baseline is calculated as the mean between the two boundaries of the baseline region – the baseline is displayed on the graph as a horizontal red line. The measured data (Im or Vm) is calculated as the mean of the data contained within the 'measure' boundary region.

Alternatively, the Im injection protocol start / stop times can be used to set the baseline and measure regions.

From Data (Start / Stop)

The known start and stop times of the current injection can be input directly (by clicking 'From Data (Start / Stop)' in the dropdown menu and 'Set Im'). The baseline is calculated as the average from the start of the record to the start of the Im injection. The measured data is calculated as the average from the start of the Im injection to the end. A padding of *n* samples around the Im injection start and stop times is included to account for the non-

instantaneous ΔIm. The number of samples to pad the calculation can be set by clicking ‘Options’ > ‘Misc. Options’ > ‘User-Im Protocol Options’ (default 10).

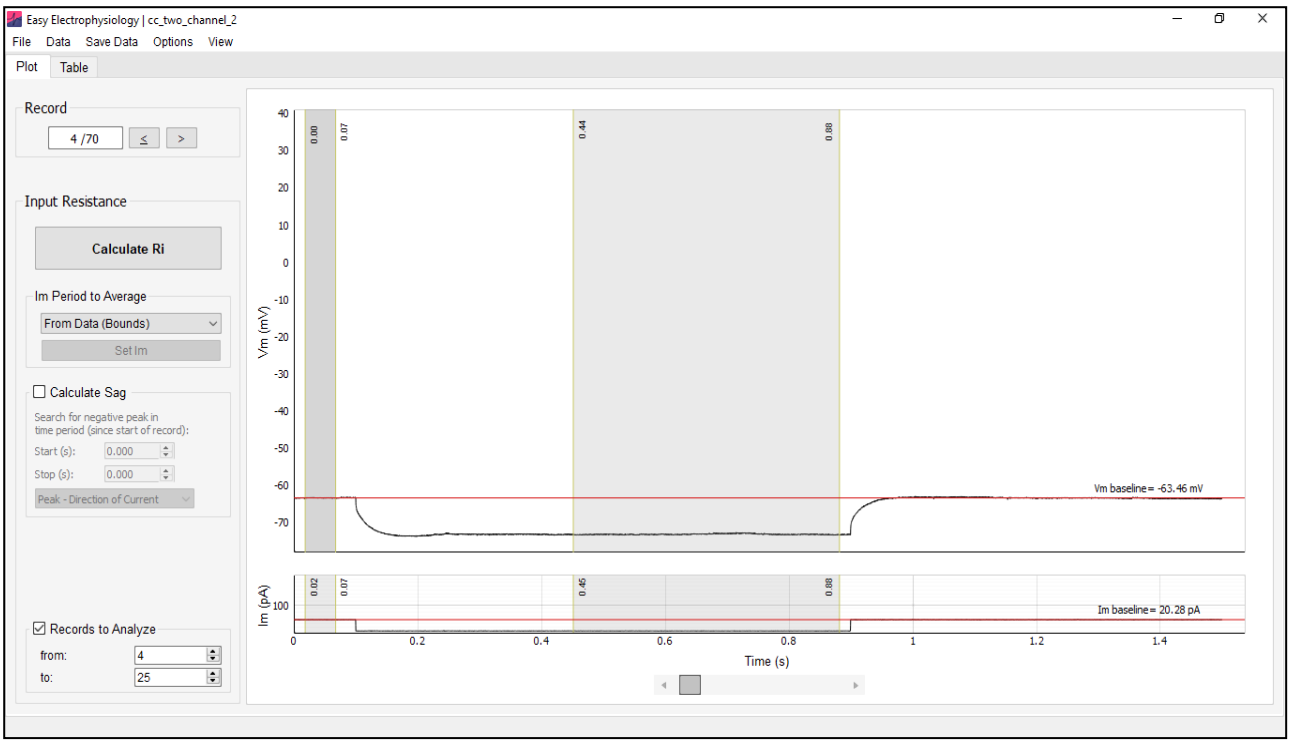

Figure 20. Input Resistance Analysis. Regions for calculating the ΔVm and ΔIm are displayed on the graph. The Δ is calculated as the difference between the baseline (mean of the data within the ‘baseline’ region, indicated with a red line) and Vm (mean of the data within the ‘experimental’ region). Here the *Records to Analyze* are selected to avoid records with action potentials.

User Input Im

User-Input

Im injection steps can be input directly (rather than measured from the data). This is particularly useful if only the Vm channel was recoded. Switching to the ‘User-Input’ option in the dropdown menu on the *Input Resistance* panel and clicking ‘Set Im’. The Im injection for each record can be input per row – the number of Im injections input must match the number of records to analyze.

For rheobase analysis (*Action Potential Counting),* a ramp protocol can also be specified for ‘exact’ rheobase calculation by clicking the ‘Ramp’ tab and inputting the current stimulation protocol.

Additional Options

Round Im Injections

If current is calculated from the data, the measured Im steps can be rounded based on the Im injection protocol using the drop-down menu on the 'Table' tab panel. Although the rounding algorithm tries to account for noise by incorporating the current injection protocol, it is recommended to cross-check these rounded values in case of stray Im measurements.

Link Im to Vm

When region boundaries are shown on both Vm and Im plots, their positions can be linked using the option 'Options' > 'Boundary Options' > 'Link Im to Vm'. In this mode, Im and Vm regions will be aligned and moving the Vm region boundaries will move the Im region boundaries, and vice-versa. By default, this option is turned off.

Link Across Records

This option will link the position of region boundaries across records. That is, moving a region boundary on one record (e.g. Record 1) will also move its position on another record (e.g. Record 2). If off, moving a region boundary on one record will not affect its position on another record. This option can be changed at 'Options' > 'Boundary Options' > 'Link Across records' and is turned on by default.

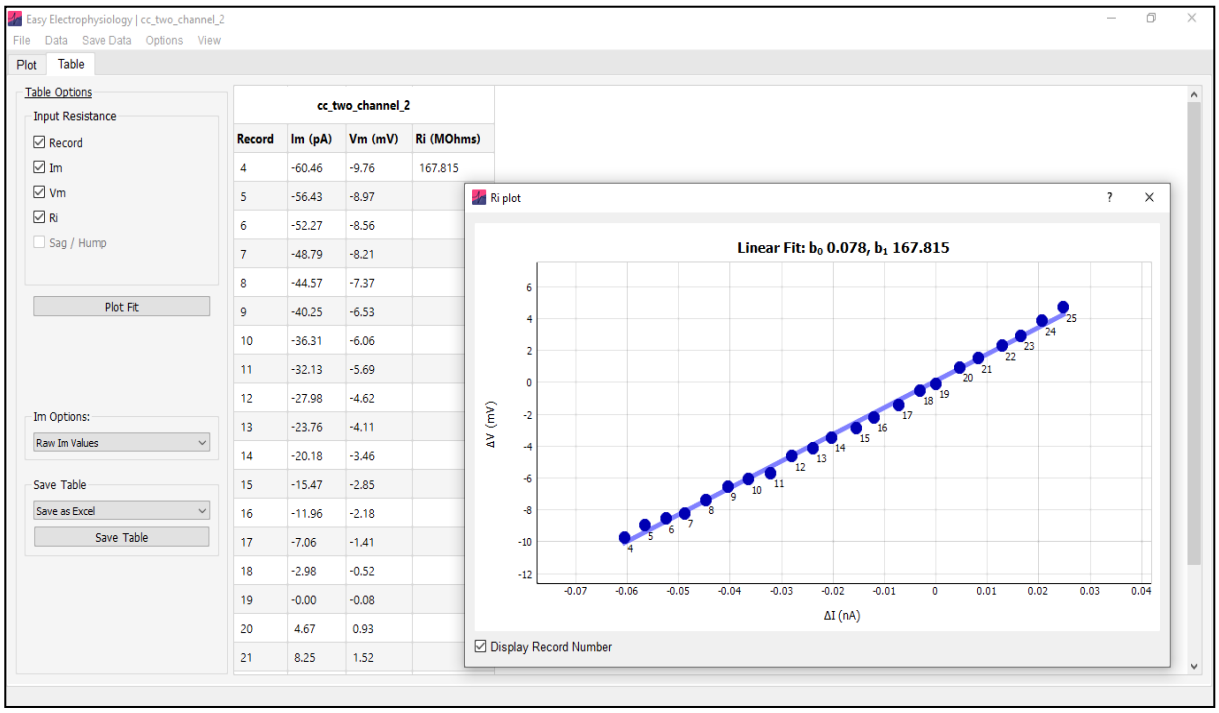

Figure 21. Input Resistance Results. The results for input analysis displayed on the 'Table' tab. Clicking 'Plot Fit' will display the linear fit used to calculate the input resistance along with coefficients of the line.

Sag / Hump Analysis

Many neuronal types display a pronounced Vm deflection following negative (sag) or positive (hump) current injection that decays to the steady-state response. Options are provided for finding the sag or hump peak within a specified time window (the ‘Calculate Sag / Hump’ box).

The peak detected can be in the same direction as the current injection (e.g. minimum if the Im injection is negative, maximum if positive) or forced always maximum / minimum. The desired option can be selected from the drop-down menu found in the ‘Calculate Sag’ box.

The Sag / Hump is position signified on the plot with a red circle. The sag or hump peak voltage deflection is calculated as the difference between peak Vm and steady state Vm response. The Sag / Hump ratio is defined as the sag (or hump) divided by the minimum (or maximum) voltage deflection from baseline.

Input Resistance Plot

A ΔI / ΔV plot with line of best fit and displayed coefficients is available by clicking the ‘Plot Ri’ on the Panel in the Table tab (Fig. 21). Note that this is only available if more than 1 record is analyzed.

10. Action Potential Kinetics

Action Potential Kinetics analysis calculates AP threshold, amplitude, half-width, rise and decay-time, fast- and medium- after-hyperpolarization (fAHP, mAHP respectively).

Options for automatic detection and manual selection of APs are available. For details on automatic spike detection, please see the *Action Potential Counting* section of this manual.

Action Potential Detection

The action potential kinetics algorithm works by first identifying the action potential peak, calculating the threshold, then calculating the additional measures (Fig. 22).

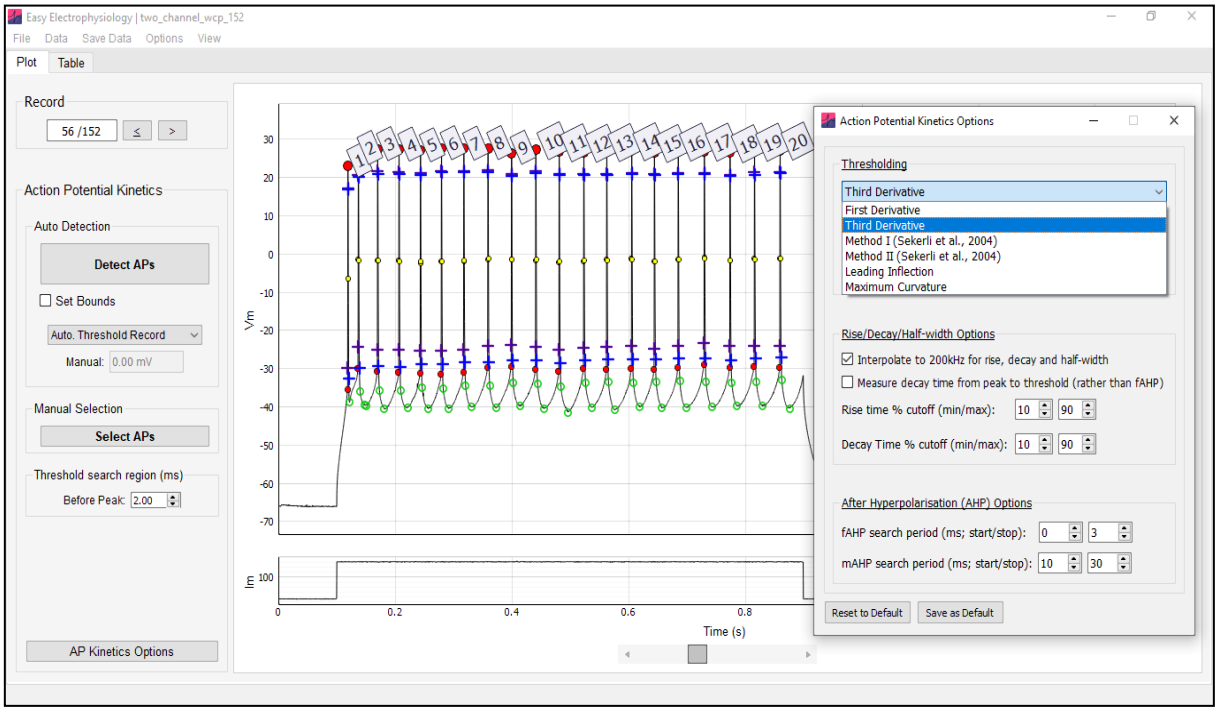

Figure 22. Action Potential Kinetics Analysis. Calculation of AP threshold, rise-time, decay-time, half-width and fAHP / mAHP with customizable analysis options.

Action Potential Kinetics Overview

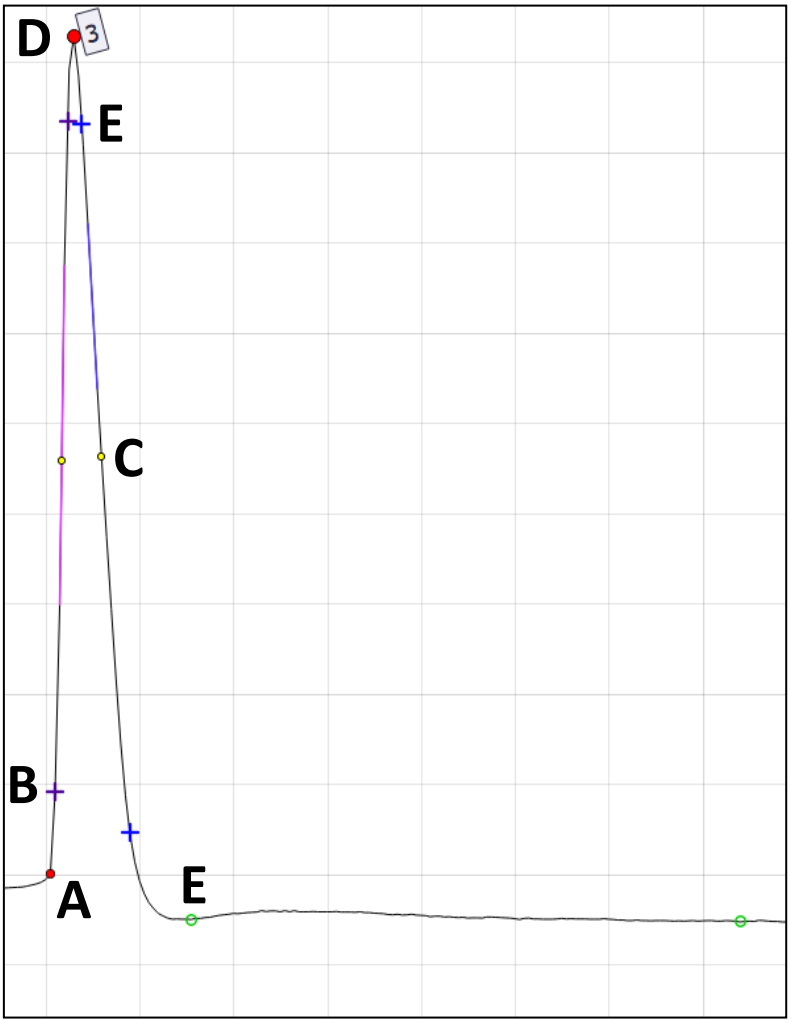

Figure 23. Action Potential Kinetics. Analyzed AP with 200 kHz interpolation for rise-time, decay-time and half-width.

A *Threshold* – The threshold value for the AP. The period to search for the threshold (prior to the peak) can be set in *Threshold Search Region (ms)*. The method used to detect the threshold can be set in *Threshold Detection Method*.

B *Rise-time* – The rise-time is calculated as the time between two samples specified as a percentage of the AP amplitude (displayed with the purple cross; default 10-90). The percentage cut-offs can be set using the *Rise-time % Cutoff* option.

C *Half-width* – This is the time between the two half-amplitude samples (on the rise and decay; displayed on the plot with black-bordered filled yellow circles). The nearest samples to the 'true' half-amplitude are used to calculate the FWHM; as such *Interpolate to 200 kHz for rise, decay and half-width* can greatly improve accuracy.

D *Peak* – The action potential peak

E *Decay-time* – The decay time is measured as the time between two samples specified as a percentage of AP amplitude (displayed with the blue cross, default 10-90). The percentage cutoffs can be specified with *Decay-time % cutoff* and the decay endpoint can be specified using *Measure decay time from peak to threshold (rather than fAHP)*.

F *Afterhyperpolarization* – The fast- (fAHP) and medium (mAHP) afterhyperpolarization is calculated as baseline minus fAHP or mAHP value. The fAHP or mAHP value (displayed using an open green circle) is the minimum value within a search region that can be specified using the *After-hyperpolarization (AHP) options*.

Action Potential Kinetics Options

Options can be accessed from the 'Action Potential Kinetics Options' button at the bottom of the *Action Potential Kinetics* panel or by clicking 'Options' > 'Action Potential Kinetics Options'.

Threshold Detection Method

A number of threshold detection methods are available from the *Thresholding* dropdown menu (see *Appendix I* for mathematical formulas). Once an AP peak is detected, the detection method is applied to a region prior to the peak with window size specified in *threshold search region.* The detected threshold is plot on the AP with a closed red circle. The threshold is calculated as the Vm which maximizes / minimizes the following functions (see *Appendix I* for details):

- *First Derivative* –
    - *Maximum* – The maximum of the first derivative.
    - *Cutoff* – The first sample that the first derivative passes the cutoff value. A value of 20 mV / ms usually works well.
- *Third Derivative* –
    - *Maximum* – The maximum of the third derivative.
    - *Cutoff* – The first sample that the third derivative passes the cutoff value.

- *Method I (Sekerli et al., 2004)* – Method I of the phase-space methods introduced in Sekerli et al., 2004). Samples with first derivative under the lower value are excluded as potential thresholds; the default value typically works well.
- *Method II (Sekerli et al., 2004).* First derivative lower bound may be specified similar to *Method I*.
- *Leading Inflection* – minimum of the first derivative.
- *Maximum Curvature* – introduced in Rossokhin and Saakian (1992).

Threshold Search Region (ms)

The period prior to the peak (in ms) to search the Vm data for the AP threshold .

Half-width, rise- and decay-times

The decay-time can be measured from the peak to fAHP (default) or to the closest sample in the repolarizing phase to the action potential threshold (when box checked).

Interpolate to 200 kHz for rise, decay and half-width

Interpolate the data (linear) to a sampling rate of 200 kHz prior to calculation of the half-width, rise and decay-times. This is recommended to improve accuracy of measured kinetics.

Measure decay-time from peak to threshold (rather than fAHP)

When this option is set, the decay amplitude is measured from the AP peak to the fAHP. By default (i.e. when this option is unchecked), the decay amplitude is measured from the AP peak to the first sample that the decay crosses the AP threshold. The *Decay-time % cutoff* is calculated on the decay amplitude.

Rise-time % Cutoff

By default, the 10-90 rise-time is calculated. The percentage cutoff values for the rise-time can be changed using this option.

Decay-time % Cutoff

By default, a 10-90 decay-time is calculated from the decay amplitude (time from peak to fAHP or threshold value, depending on settings). These percentage cutoff values for the decay-time can be changed using this option.

Calculate Max Slope

The maximum rise and decay for each AP will be calculated if this option is selected (it is not calculated by default as it can increase analysis time).

The maximum slope is calculated as a regression over n points. The number of samples to calculate the regression over can be specified separately for the rise and decay.

After-hyperpolarization (AHP) options

Set the time period to search for the fast- and medium- afterhyperpolarization (fAHP and mAHP respectively). The fAHP and mAHP are plot on the AP with open green circles.

The mAHP is defined as the period after the AP peak (to measure the voltage sag at end of a current injection to which mAHP sometimes refers, see *Input Resistance,* Sag / Hump analysis).

Phase Plot Analysis

Phase plot analysis of action potentials is available under *AP Kinetics* analysis. After an analysis has been run, the 'Phase Plot Analysis' button on the bottom-left of the *AP Kinetics* panel will bring up the phase plot analysis window.

The phase plot analysis window contains two panels (Fig. 24). On the left panel is displayed the currently selected AP, while the right panel contains the phase-space plot of the currently selected AP.

Selecting APs

All analyzed APs can be selected for phase-space analysis. APs can be selected by iterating through all APs using the '<' or '>' buttons, or directly inputting the Record / AP number into the Record / AP number boxes.

The window size (+/-) (ms) allows adjustment of the windowing for the selected AP. Increasing / decreasing the AP window will change both the AP plot and the phase-space plot, affecting the phase-plot results. In practice, this is often used to adjust the start-point of the analysis close to the AP threshold.

In theory, it is also possible to increase the window to create phase-space plots of multiple APs. However, the phase-space plot parameters will be calculated only once on a phase-space plot.

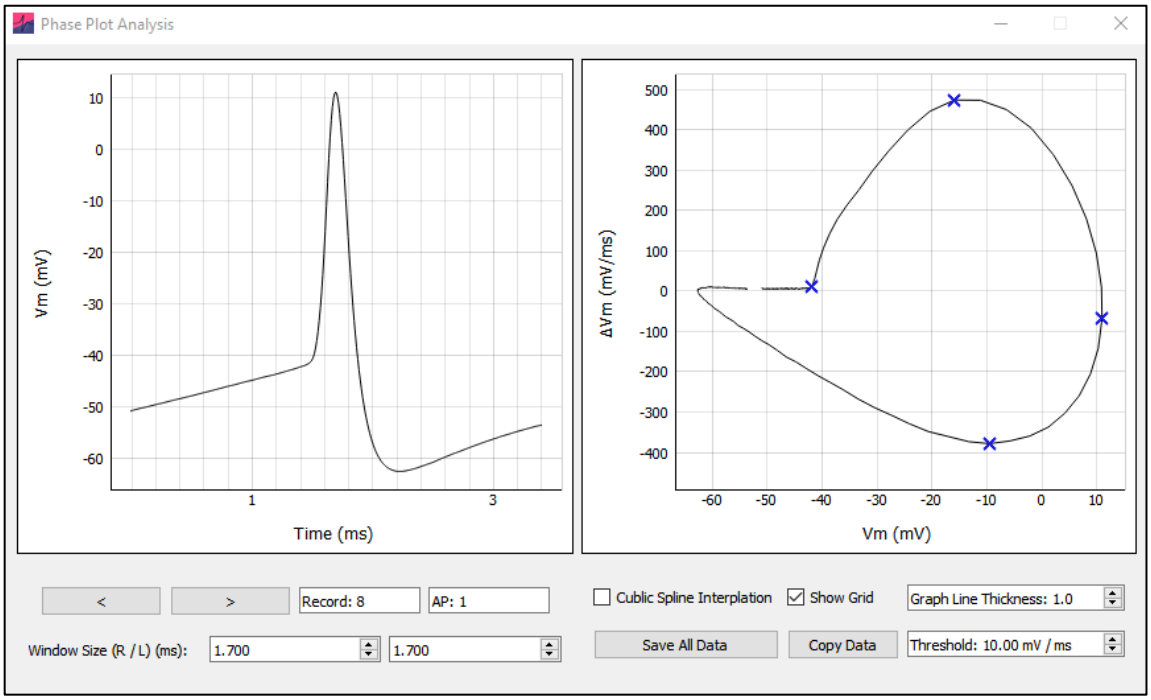

Figure 24. Phase Analysis Dialog. The AP is shown on the left, windowed +- 1.7 ms from the peak. On the right, the phase-space plot of the AP.

Phase Plot Analysis

Phase-space plot graphs the first derivative of the Vm (ΔVm, in mV/ms) (y-axis) against the Vm (x-axis). This is interpreted as the rate at which the Vm changes at each voltage level.

When the Vm increases (left-to-right on the x-axis) the positive first derivative indicates the rate-of-increase in the positive direction (i.e. the rising phase of the AP), and we can see the dynamics of the AP rise. The negative first derivative indicates the negative rate-of-change (i.e. the falling phase of the AP). For example, in Fig. 24, we can see the AP hyperpolarization, the negative rate-of Vm change occurring at voltages smaller than ~-40 mV.

Phase Plot Parameters

The calculated parameters from the phase-space plot are the Threshold (see above), maximum ΔVm, minimum ΔVm and maximum Vm. Note the threshold and maximum Vm are equivalent to the AP threshold and peak.

Results can be obtained in a number of ways:

<u>Right-click the plot, select 'Copy Parameters'</u> will copy the Record, AP number, AP peak time, and (ΔVm, Vm) coordinates (i.e. x, y coordinates) of the Threshold, maximum ΔVm, minimum ΔVm and maximum Vm to clipboard.

‘Copy Data’ will copy the above parameters, as well as the raw values of all plots (i.e. the AP (time and Vm) and phase-space plot (ΔVm and Vm). Note the Vm will be equivalent unless interpolated.

‘Save All Data’ will save the AP phase plot parameters, AP and phase-plot values for every analyzed AP into an excel or .csv file.

Analysis Options

Cubic Spline Interpolation: Interpolates the phase-space plot, in a smooth manner (ensuring up to 3rd derivatives at interpolated ‘knots’ are continuous, interpolation factor 100). This can be useful when the AP is not well sampled.

Threshold (mV/ms): A simple thresholding approach is used to calculate the AP threshold from the phase-space plot. The AP threshold is taken as the Vm at which the first derivative crosses this threshold setting (mV/ms).

Graph Options

Show Grid: Toggle on / off the AP plot and phase-plot grid

Graph Line Thickness: Adjust the thickness of the AP plot and phase-plot graph lines. Note due to a bug in the underlying graphing software, line thickness > 1 may reduce performance.

11. Data Tools

Data Tools provides methods for processing electrophysiologic recordings including filtering, resampling, detrending and record averaging.

Data Tools can be combined (e.g. interpolate followed by detrend) and run before carrying out analyses. When *Batch Mode* is off, running these methods will clear previous analysis from the graph and results table. When *Batch Mode* is on, running these methods will clear previous analysis from the graph but not from the results table (and running a new analysis will add new table columns rather than overwrite the previous results).

Interpolation and filtering methods utilize SciPy (https://www.scipy.org/) package routines.

Upsample

Data upsampling increases the resolution of the data by interpolating between samples. The following methods of upsampling are available:

- *Nearest Neighbor* – the simplest method of interpolation, in which the interpolants are chosen as the nearest datapoint.
- *Linear Interpolation* – a coarse interpolation method in which the interpolants are chosen to lie on a straight line between adjacent datapoints.
- *Cubic Spline Interpolation* – a sophisticated interpolation technique in which interpolants are chosen to lie on 3rd-order piecewise polynomials between adjacent data points. The polynomial coefficients are set to ensure the interpolated function is continuous at the first and second derivative.

For most purposes cubic spline interpolation is optimal as it produces the smoothest interpolation.

Downsample

Downsampling reduces the data resolution by discarding every nth sample (up to a factor of 12). Prior to downsampling, the primary channel data is low-pass filtered to the Nyquist frequency of the downsampled data. For two-channel data, both the primary and secondary channel are downsampled, but only the primary channel is filtered before downsampling. This is because it is assumed the secondary channel will contain Im / Vm step protocols. Options for choosing the low-pass filter are provided (default 8th order Bessel).

Filter

Options for low- and high-pass filtering with Bessel and Butterworth filters are provided (Fig. 25).

Zero-phase shift filtering is achieved by passing the data forwards and backwards through a linear digital filter. This process doubles the effective order of the filter, and this is accounted for in the 'Filter' options menu. For example, if an 8th order Bessel is specified, a 4th order filter is designed and the data passed through twice for an effective filter order of 8.

Convenient interfaces are provided for checking the results of filtering before applying to the data. The 'See Filtered Data' button will display the currently selected trace overlaid with the filtered trace in red. 'Show FFT' will show the Fourier-transformed data, with the power-spectrum of the filter data overload in red. Finally, clicking 'Apply' will filter all records in the file.

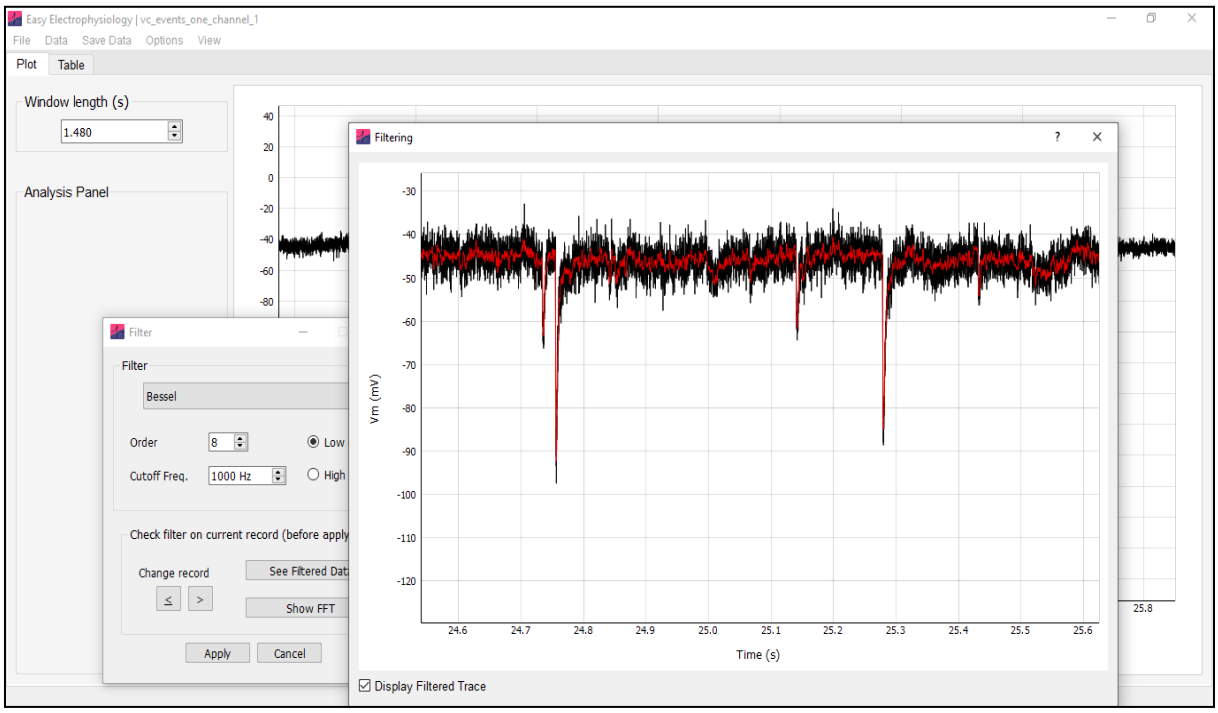

Figure 25. Filter Data. Data filtered at 1000 Hz with a low-pass 8th order Bessel. Clicking ' See Filtered Data' displays the data overlaid with the filtered data. This is useful for checking the effects of filtering prior to applying to the data.

Detrend

Linear and non-linear drifts can be removed from electrophysiological timeseries by detrending. This is particularly useful for long recordings (e.g. sEPSCs) that show some baseline variability.

An nth order polynomial (up to 20th order) is first fit to and then subtracted from the data (see *Appendix I*). In the *Detrend* analysis window, the type of detrending and polynomial order can be accessed through the drop-down menu and input box respectively. Note that the mean is added back to the data after detrending (to remove the mean, see *Remove Baseline*).

Clicking 'View Fit' will display the currently selected record overlaid with the fit polynomial in red. Use the 'View Detrended Trace' button to view the currently selected record after detrending (without applying to the data).

Finally, clicking 'Apply to All Records' will detrend all records in the dataset. A polynomial of the order specified will be fit and subtracted (with the mean added back) for every record iteratively. Only the primary data channel will be detrended.

Remove Baseline

Removing the baseline is accomplished by subtracting a baseline value from the data. The baseline is calculated as the average within a specified time window (indicated by the period within the boundary region that appears on screen).

Clicking 'Apply' will apply baseline removal for every record in the file. For each record, the baseline is calculated and subtracted separately.

If your file has two channels (i.e. Vm and Im) clicking the checkbox option will also calculate and remove the baseline from the second data channel (across the time window specified on the primary data channel).

Average Records

Averaging across records is useful for reducing noise in repeated-stimulation protocols. If set to 'Average All Records', every record in the file will be averaged into one record.

Alternatively, specific records can be selected for averaging by choosing the ‘Average specific records’ option from the dropdown menu. A single-column table will be displayed. The rows of the table can be used to group records for averaging. For example, in a 10-record file 1,2,3 could be entered to the first row, and 4,5,6 into the second. Clicking ‘Apply’ would generate with 2 records (record 1 - average of 1,2,3; record 2 - average of 4,5,6, with records 7,8,9,10 discarded).

Record Arithmetic

Record arithmetic (‘Data’ > ‘Record Arithmetic’) provides the option to perform arithmetic operations pointwise across records e.g. add two records together, sample-by-sample.

Record Arithmetic can be performed by a flexible interface that takes text input. Records are referred to with the “r” prefix, and permitted mathematical operations are the addition (“+”), subtraction (“-“), multiplication (“*”) and division (“/”), and brackets. For example, to add the first and second record: “r1+r2”. To calculate the average of the 2nd, 3rd and 4th records: “(r2+r3+r4)/3”.

Multiple record arithmetic expressions over sets of records can be performed by inputting an expression on each row of the input table. For example, Fig. 26 will result in a 2-record file, with the first record being the sum of the first and second record, and the second record being the sum of the second and third record.

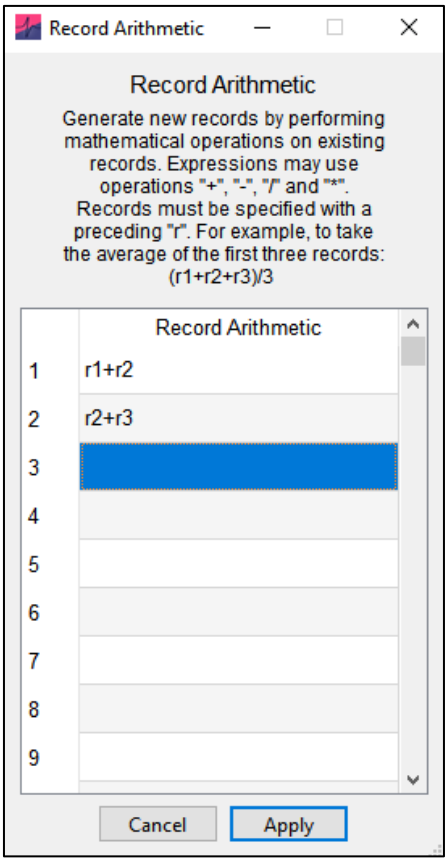

Figure 26. Record Arithmetic. Record arithmetic allows pointwise mathematical operations across records. In this example, the resulting dataset will consist of two records. The first record will be the sum of the first and second record in the original dataset, the second new record will be the sum of the second and third record in the original dataset.

Stimulus Artefact Removal

Electrophysiological recordings are often contaminated by artefacts from electrical stimuli used to elicit the membrane responses during experimental protocols (Fig. 27). *Stimulus Artefact Removal* can automatically remove such stimulus artefacts from the data.

The data with stimulus artefacts removed can be displayed by clicking the 'Preview' button, which will preview the trace with stimulus-artefact removed in orange. Clicking 'Apply' will apply the stimulus artefact removal to the data, closing the Stimulus Artefact window.

The default parameters typically work out of the box on most datasets, however the details of the algorithm are explored below in case the default parameters need to be adjusted.

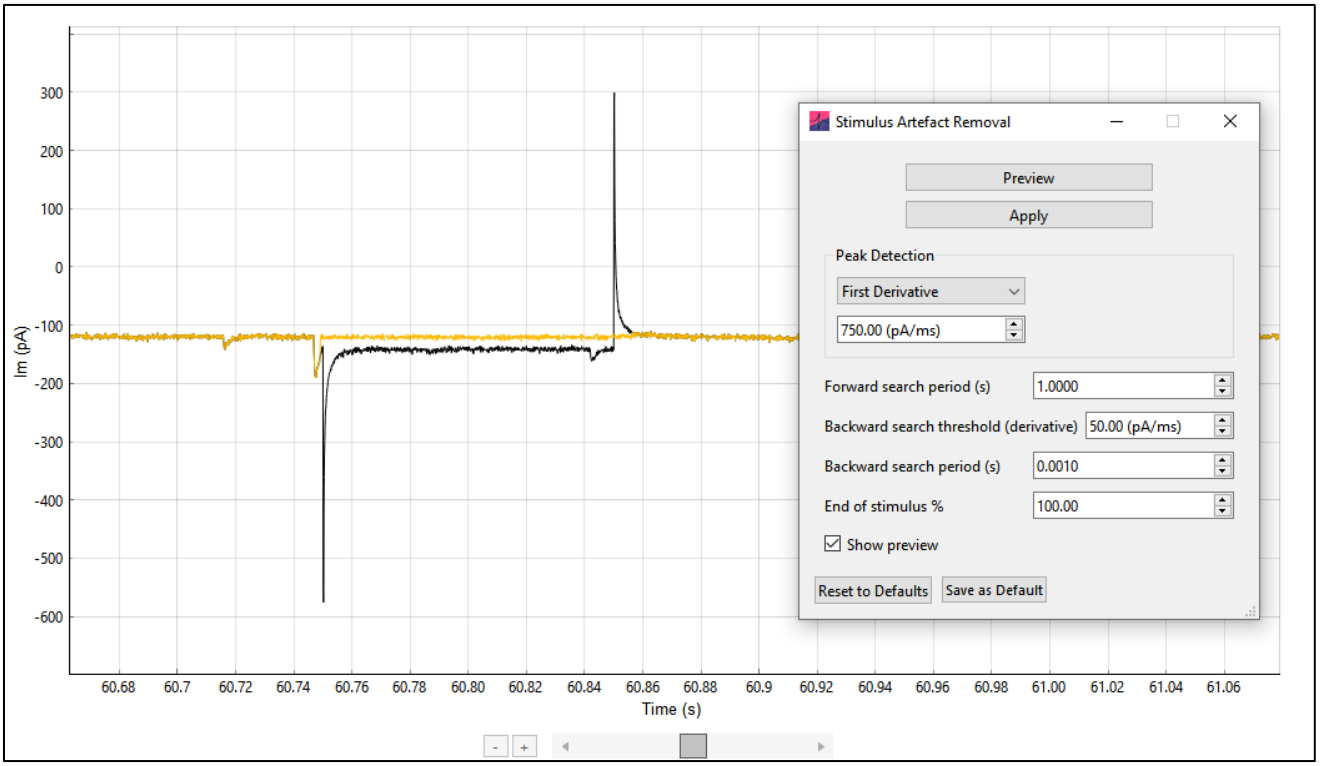

Figure 27. Stimulus Artefact Removal. In this example, periodic stimuli used to check the patch seal contaminate the recordings and may erroneously be detected as events. Stimulus Artefact Removal can automatically remove such stimuli. Here, the 'Preview' is displayed in orange. Clicking 'Apply' would remove the stimulus artefact from the trace.

The stimulus artefact removal algorithm

The first step is to detect the peaks (positive or negative) of the stimulus artefacts. The method used for detection is by default the first derivative. The first derivative is usually optimal because stimulus artefacts typically have a much faster rise times than physiological responses. Therefore, setting a high derivative threshold can selectively pick out stimulus artefact events. However, an absolute threshold is also available.

Next, the period from the stimulus artefact peak to the end of the artefact in the 'forward' direction (i.e. moving forward in time) is detected. The length of the window that is searched (from the peak to the end of the artifact) is set by *forward search period (s)*.

The value taken as the end of the stimulus artefact is determined by the *End of stimulus %* parameter. The sets the data value at which the stimulus artifact is determined to have reached baseline. This is detected by the crossing of a threshold which takes the value of the baseline signal plus some percentage of the noise of the signal (here, the noise is measured as the median difference between consecutive samples). The *End of stimulus %* sets this percentage. For example, if *End of stimulus %* is set to 200%, a datapoint will be considered returned to baseline once it is within twice the noise level.

Finally, the point where the stimulus artefact started is determined. The parameter b*ackward search period (s)* defines the size of the window, starting at the peak, which is searched for the stimulus artefact start. Because stimulus artefacts are characterized by a fast rise time, a derivative threshold is again used. The first sample that crosses the *backward search threshold (derivative)* is taken at the start of the stimulus artefact.

All datapoints within the region determined to contain the stimulus artefact are replaced with samples randomly drawn from a Gaussian distribution with mean and variance estimated from the data.

Reshape Records

Selecting *'Concatenate all records into a single record'* will combine multiple records into one long record. Note that where recorded time intervals between records are not continuous, because they reflect gaps in the recording schedule (e.g. Record 1, 0 – 2 s, Record 2, 3 – 5 s), they will be made continuous (e.g. record 1 and record 2 would be concatenated into a single record between time points 0 – 4 s).

Alternatively, a single-record dataset can be split into multiple records if the number of records is known by selecting *'Expand data from a single record to multiple records'*. This can be useful as in some instances .ABF files exported from other software (e.g. Igor, WinWCP) are concatenated into a single-record file, or to increase performance for long single-record files.

The option is provided to normalize time when reshaping records (so every record has the same time period, see *Normalize Time).*

If the number of samples does not divide without remainder into the number of records, a warning will show and the remainder samples will be removed from the end of the single record before reshaping.

Long single record files (containing millions of datapoints) can slow down the software due to the overhead associated with displaying the data on the graph. A simple workaround is to reshape the single-record into multiple, smaller records. The only difference this will make is for *Events* analysis, inter-event intervals (used for Frequency cumulative probability and histogram plots) are not calculated across records. As such, the number of inter-event intervals will be reduced by N, where N is the number of records. In large files with many hundreds or thousands of events this will make no difference to the results, in particular considering inter-event interval size is randomly distributed through the recording.

Normalize Time

The time in electrophysiological data may be recorded in two ways. If 'normalized', the times of each record are the absolute record length (e.g. for 2 second recordings, record 1: 0 – 2 s, record 2: 0 – 2 s). Alternatively time may be recorded relative to the start of recording, increasing cumulatively (e.g. record 1: 0 – 2 s, record 2: 2 – 4 s). *Normalize Time* applies the time vector from the first record for every record in the recording. Note that this assumes all records in the dataset are the same length.

Cut Trace Length

This give the option to set new record start / finish times (e.g. cut a trace that is 0 – 2 s long down to the time period 0.5 – 1 s). An offset will always be subtracted to ensure the first record starts at time = 0.

Clicking 'Apply' will change the record length for every record in the file. Options are provided for cumulatively increasing record times after cutting trace length (e.g. record 1: 0 – 0.5 s, record 2: 0.5 – 1 s) or normalizing time instead (see *Normalize Time*).

Reset to Raw Data

If you are unhappy with the results, data can be reset to raw data (as loaded from file).

12. Analysis

Kolmogorov-Smirnov Tests

Kolmogorov-Smirnov (KS) tests for comparing the distribution of one sample (against the normal distribution) or two samples (against each other) are available at 'Analysis' > 'Kolmogorov-Smirnov'. Raw data can be copied directly into the input data table (it is not necessary to calculate the cumulative distribution first).

The KS test compares two distributions by taking the maximum absolute difference of their cumulative (or empirical; Fig. 28) distribution functions (CDF), termed *D*. Hypothesis testing is performed against the Kolmogorov distribution.

For one-sample tests, the empirical CDF of the entered data is compared against the normal (i.e. Gaussian) CDF. Where the mean and standard deviation of this Gaussian in estimated from the data, Lilliefors correction is used.

To plot the empirical CDF, click 'Plot Data' and select 'Empirical CDF' from the drop-down menu.

Easy Electrophysiology uses the SciPy module 'kstest' for Kolmogorov-Smirnvov tests, except for the case in which Lilliefors correction, where the statsmodel package function 'lilliefors' is used.

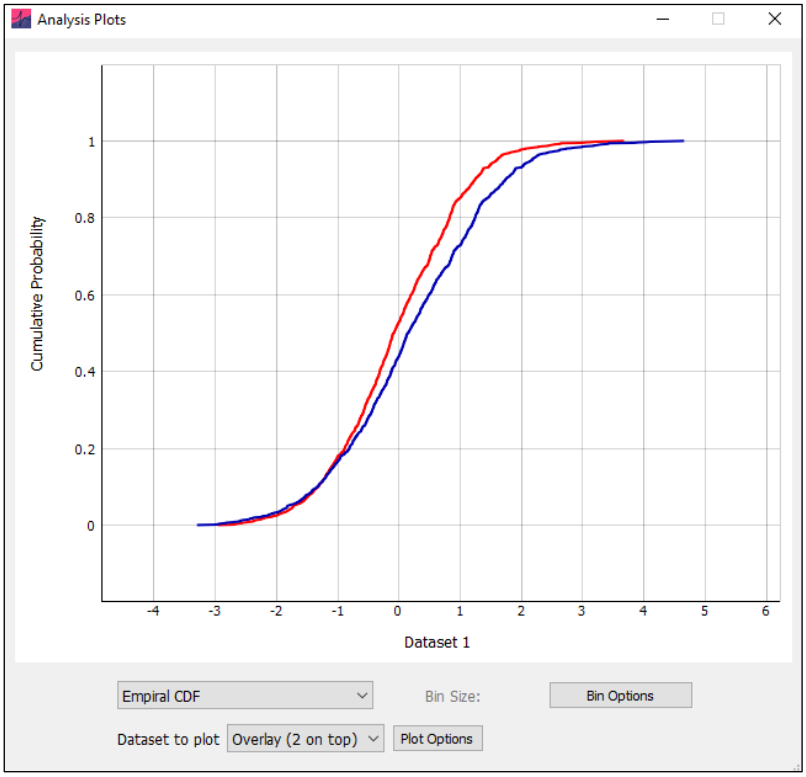

Figure 28. Empirical CDF Plot. Plot of the empirical CDF for two datasets.

Summary Statistics

Summary statistics can be calculated from data displayed on the results table. Clicking the ‘summary statistics’ checkbox on the table tab panel will display (by default) the mean, standard deviation and standard error of each parameter, for each file analyzed.

To change the calculated parameters, click ‘Analysis’ > ‘Table Options’. Parameters with meaningless summary statistics (e.g. record number, event times) are not calculated.)). In Addition, a ‘count’ column is shown that displays the number of rows that were summarized. For *Events* and *AP Kinetics,* this will equal the number of events / action potentials summarized. For all other analyses, this will equal the number of records summarized.

This window (Fig. 29) also provides the option to display results with analyzed files stacked vertically (‘per row’) rather than horizontally (‘per column’).

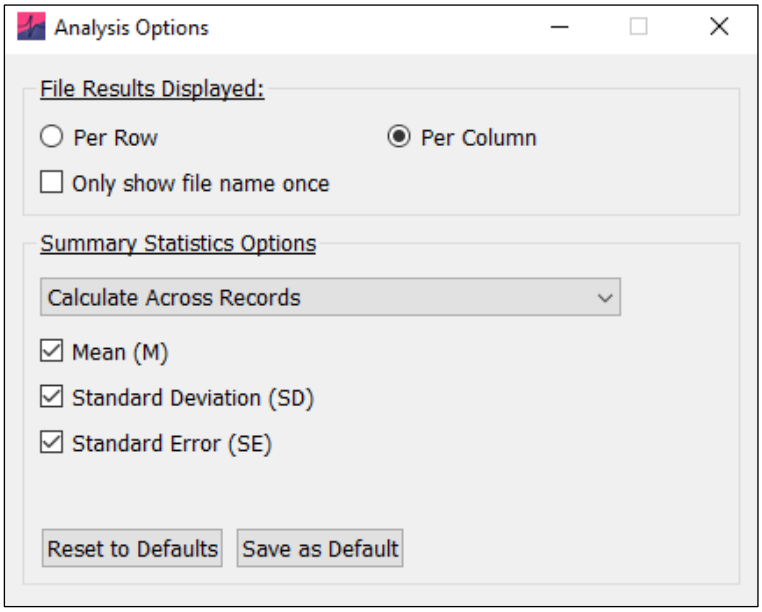

Figure 29. Options window for table analysis and display.

13. Burst Analysis

Burst Analysis is available for both Events (*Events Template, Events Thresholding)* and AP (*AP Kinetics*) analysis. After running an *Events* or *AP Kinetics* analysis, burst analysis can be accessed by ‘Analysis’ > ‘Burst Analysis’ from the top menu bar.

Two methods for Burst Analysis are provided, the LogISI and MaxInterval method, the two best performing methods as described in Cotterill et al. (2016).

Burst Analysis Parameters

‘Run Burst Analysis’ will auto-detect bursts in the data (Fig. 30) (see below for details of burst detection methods) and calculate burst summary.

When burst analysis is run, the main plot will update to display the detected bursts. To switch between main plot views, toggle *Show Burst on Main Plot.*

Note that in the software, a detected spike or event is called ‘spike’ in *AP Kinetics* mode (Fig. 31) and an ‘event’ in *Events* mode (Fig. 30). In this documentation, we will use the term ‘spike’ to cover both cases. The inter-spike interval (ISI) is defined as the time period between two consecutive spikes.

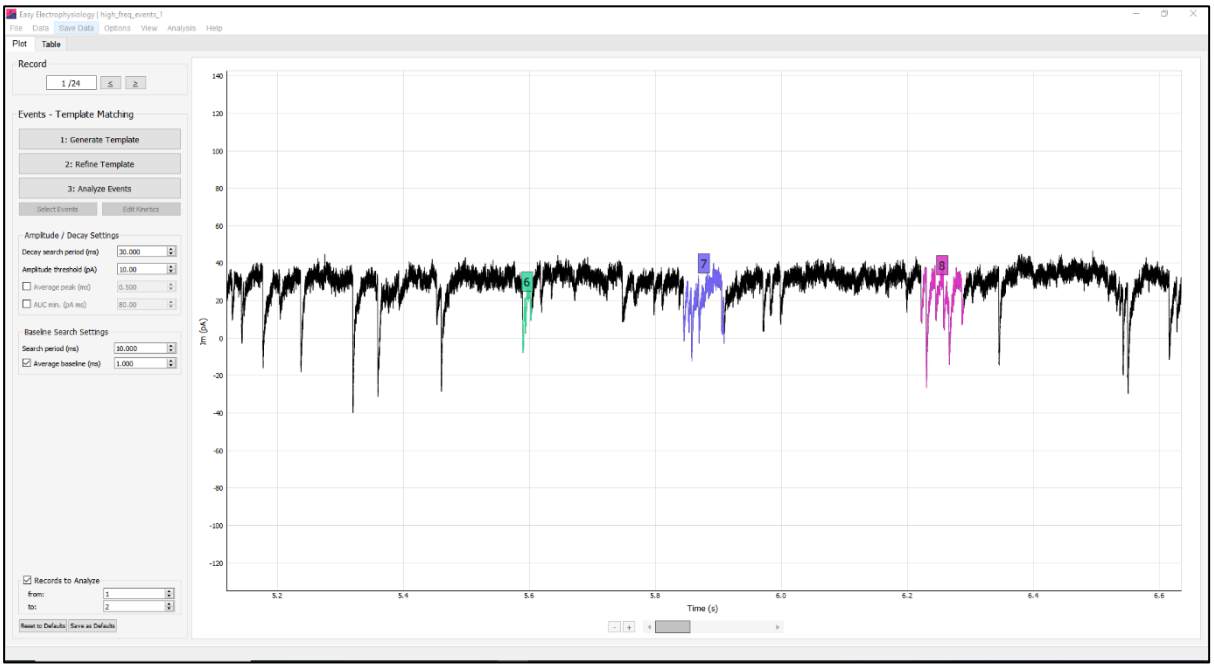

Figure 30. Burst Detection *Events*. Detected bursts are shown in different colours with auto-generated labels.

Summary Statistics

The burst summary statistics are displayed on the left-hand panel of the burst analysis window (Fig. 32), as well as in the results table ('Show Full Results Table').

Number of bursts: Total number of detected bursts in the dataset (where the dataset is the number of detected APs or events).

Fraction of spikes in burst: Proportion of detected spikes that are in a burst. Calculated as (total spikes in a burst / total detected spikes).

Averages

Number of spikes in burst: Average number of spikes in a burst (calculated as the average 'Num. Spikes in Burst' across all bursts).

Burst length (ms): Average length of a burst in milliseconds (calculated as the average 'Burst Length (ms)' across all bursts).

Within-burst inter-spike interval (ms): Average inter-spike interval of spikes found in a burst (calculated as the average 'Mean Inter-event Interval (ms)' across all bursts)

Inter-burst interval (ms): Average time interval between successive bursts (in milliseconds).

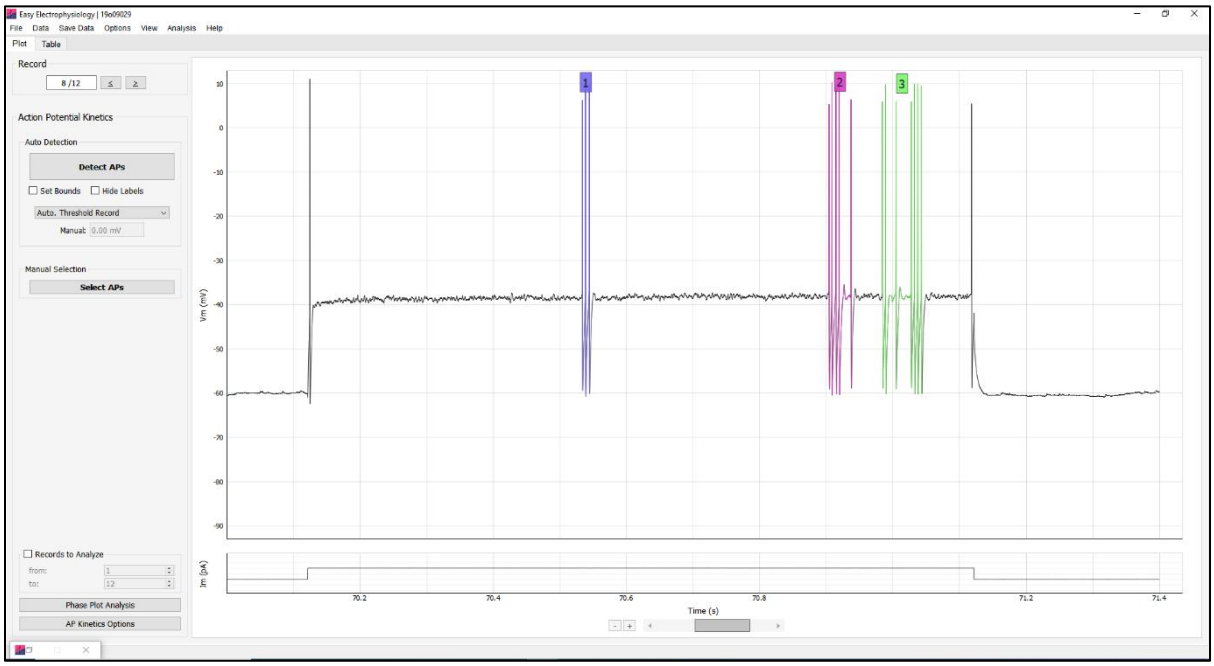

Figure 31. Burst Detection *AP Skinetics*.

Per-Burst Statistics

The parameters calculated for each burst are displayed in the results table. These include the number of spikes (Num. Spikes in Burst), burst length (Burst Length (ms), and average inter-spike interval within that burst (Mean Inter-event Interval (ms).

Further, the results table displays for each detected spike: the record, peak time and burst it belongs to ('In Burst', N/A if spike is not in a burst).

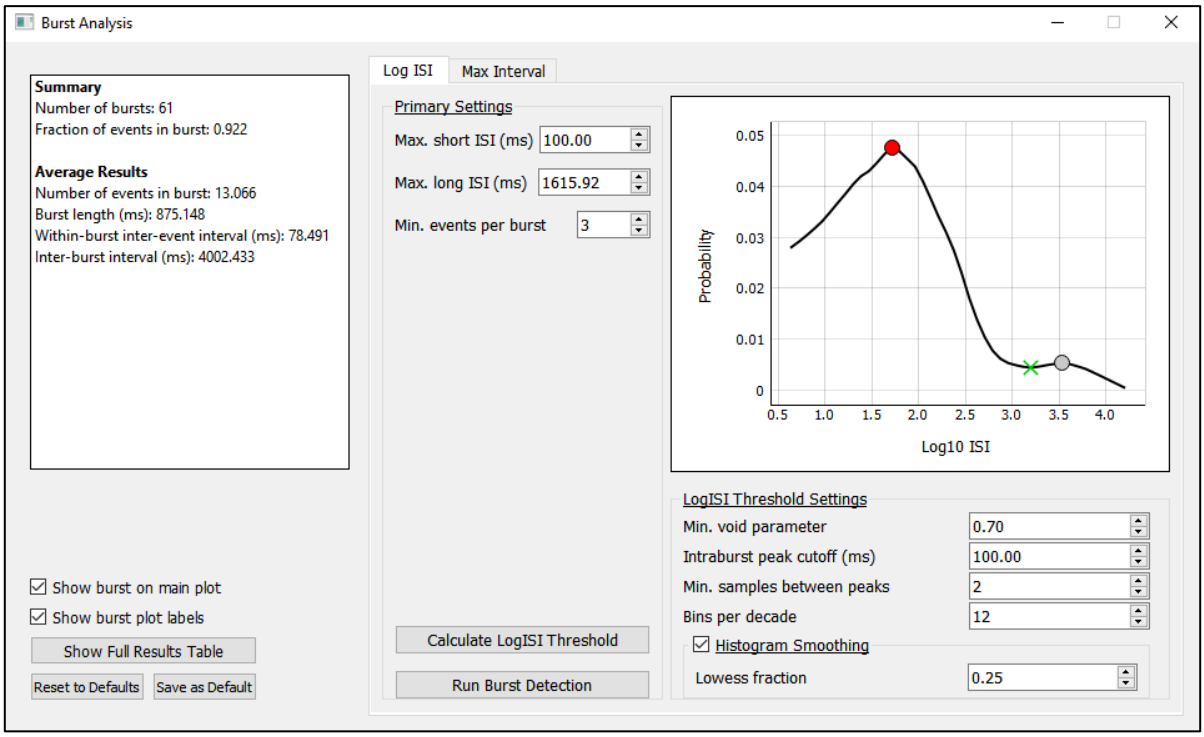

Figure 32. Burst Analysis Window. The burst analysis window, in Log ISI detection mode. On the left are the burst results summary window, and on the right is a logISI histogram (red peak: intra-burst peak, grey peak: other ISI peak, green cross: threshold used as Max long ISI (ms).

Burst Detection Methods

Log ISI

The Log ISI method (Pasquale et al., 2016) extends burst analysis to hard-to-detect burst boundaries by calculating an additional inter-spike interval threshold from the data, using the logarithmically-scaled histogram of ISIs.

The algorithm proceeds by using a manually set, short ISI ('Max. short ISI (ms)', typically 100 ms) and a long ISI ('Max. long ISI (ms)') generated from the logISI histogram. The short ISI is used to detect burst cores, while the long ISI is used to extend these cores to the burst boundaries. If the long ISI is smaller than the short ISI, the long ISI is used as both.

‘Calculate LogISI Threshold’ will calculate and plot the logISI histogram used for long ISI calculation (Fig. 32). The detected threshold will be displayed with a green cross and the long ISI updated to the thresholded value (see below for more details on threshold detection).

Primary Settings:

Max. short ISI (ms): The short ISI used to detect burst cores in the first-pass analysis. This is set manually, typical value used is 100 ms.

Max. long ISI (ms): The long ISI used to detect the burst boundaries, extending the burst cores. This is calculated from the data (logISI histogram), or can be set manually.

Min. spikes per burst: The minimum number of spikes in a burst. Bursts detected from ISI analysis with less than this number of spikes will be excluded.

LogISI Threshold Settings

The long ISI is calculated from the LogISI histogram using a peak / trough–finding algorithm. First, the intra-burst peak (expected to be dominated by the short ISI of spikes in bursts) is detected. This is expected to be the smallest peak (i.e. within-burst ISI), and so the search range is usually small (e.g. default 100 ms).

Next, all other peaks of the logISI histogram are found. A ‘void parameter’ is computed between the intra-burst peak and all other peaks, capturing the depth of the trough in between the two peaks. The trough with the highest void parameter (i.e. ISI that best separates the peaks) is taken as the new long ISI threshold (see Pasquale et al., (2016) for equations).

Min. void parameter: This sets the minimum acceptable void parameter, lowering it will increase chances of finding local minimum.

Intraburst peak cutoff (ms): The time window over which to search for the intraburst peak (the intraburst peak, consisting of the smallest ISIs, will be searched for between 0 and the intraburst peak cutoff (ms)).

Min. samples between peaks: The minimum number of samples used in the peak-detection algorithm when searching for all additional (i.e. non-intraburst) logISI histogram peaks.

Bins per decade: Sets the logISI histogram bin-size. Increasing the number of bins per decade increases the sampling of the histogram and can improve threshold detection.

Histogram smoothing: If set, LOWESS smoothing will be used to smooth the histogram.

Lowess fraction: Sets the number of points used in the local regression for LOWESS smoothing. Determines the smoothness (increasing will increase smoothness).

Max Interval

The ‘MaxInterval’ method, as it is commonly described in the literature, is an inter-spike interval thresholding approach proceeds based the following thresholds:

Max. Interval (ms): Maximum ISI at which the first ISI in a burst must not exceed.

Max. end interval (ms): Maximum ISI at which the last ISI in a burst must not exceed.

Min. burst interval (ms): Minimum ISI at which the inter-burst interval between two successive bursts must not exceed.

Min. burst duration (ms): Minimum duration in time a burst must be to be detected.

Min. spikes per burst: Minimum number of spikes a burst must have to be included in the analysis.

14. Analyzing in Batch Mode

When *Batch Mode* is off, loading a new file will clear all analyses from the previous file.

Turning *Batch Mode* on ('File' > 'Batch Mode' > 'On') causes a files analysis results to be saved in the 'Table' tab results table (Fig. 33). If analysis is run on a file then a new file is loaded, all results from this new file are appended to the end of the table. For each file, re-analysis will overwrite the results for that file only (unless a *Data Tools* method has been run between the two analyses in which case they are both displayed).

In batch mode, 'Records to Analyze' settings will not be changed when the next file is loaded (by default, they are set to the start / stop times of the new file).

In *Batch Mode* analysis can be deleted from the analysis results table by right-clicking the first row of the table and selected 'Delete analysis'.

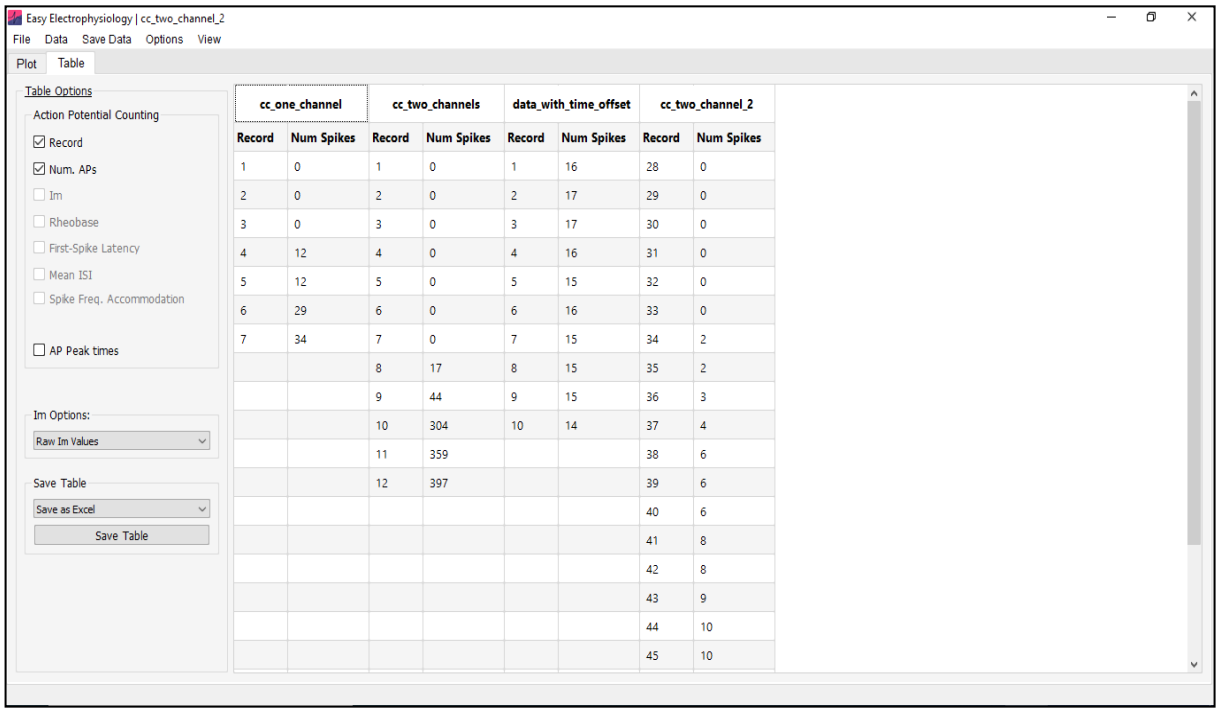

Figure 33. Analyzing in Batch Mode. Batch mode saves previous analysis on the 'Table' tab results table. Here *Action Potential Counting* analysis from 4 files run sequentially is displayed. Batch mode can be set using 'File' > ' Batch Mode' (*top left*).

15. Saving Data and Analysis Results

Copy / Save Data

Options to copy and save the displayed Vm and Im data are available under ‘Save Data’ on the top menubar.

Data from the currently selected record can be copied (‘Copy Im Trace’, ‘Copy Vm Trace’). Alternatively, all records in the file can be saved to file in .xlsx or .csv formats (‘Save All Records (Im)’, ‘Save All Records (Vm)’.

Saving to .csv is faster than to .xlsx and may significantly decrease saving times for large datasets.

Save Analysis

The displayed results in the ‘Table’ tab table can be saved to .xlsx or .csv. Saving to excel includes additional formatting around the cell borders (e.g. between different files when in batch mode). Typically with the analysis results table displayed data size is small enough that the difference in save times for .xlsx and .csv is negligible.

Data can also be copied directly from the analysis results table (select data to copy and right-click).

16. Saving and Resetting Default Options

Easy Electrophysiology has the convenient option to save the current analysis settings for later use. Default options can be overwritten using the ‘Save as Default’ button that appears on certain options panels and windows. Clicking this button will save the settings on this panel / window to file. These settings will be remembered across sessions (i.e. if you close and re-start the software).

Clicking ‘Reset to Default’ will reset all options on the relevant panel / window to their default settings (e.g. if they were changed after starting the software).

To reset settings to the Easy Electrophysiology defaults (that are set when the software is first downloaded), click ‘Options’ > ‘Reset to Factory Defaults’ and choose the options to reset.

17. Graph Options

The graphs have a number of customizable display features for clear and convenient data visualization. These can be selected through ‘View’ > ‘Graph Options’.

Y Axis Scaling

- *All Records* – the y axis is scaled to that of the record with the largest y-axis range. This ensures the y-axis scale is the same for every record in the file.
- *Displayed Record* – the y-axis is rescaled on every record change. This reduces performance when switching between records.

Plot Size Ratio and Position

The relative size of the upper and lower plots can be changed using the slider. Clicking the ‘Swap Plot Position’ button will switch the position of the upper and plot. This is simply aesthetic and will not change the primary data channel.

Show Gridlines

This will show / hide the gray gridlines on the upper / lower plot.

Graph Line Thickness

This will set the line thickness on the upper and lower plots. It is not recommended to set the plot thickness higher than the default value of 1.0 as this will significantly slow down graphics performance.

Limit View Range for Performance (Single-record files only).

For single-record voltage-clamp data the entire dataset can be displayed on the graph at once. However, depending on the length of recording time, this data can be very large and slow performance. ‘*Limit view range for performance’* will set a maximum window length so only a subset of the data is shown. This is set at 500k samples by default, but the maximum window size can be increased, decreased or this feature turned off.

18. Plot Exporter

All figures in Easy Electrophysiology are exportable, through right-clicking the plot and selecting the 'Export' option. This window provides the option to export the plot data as a .csv (comma-separated value file), many possible standard image formats (including .jpeg, .png, .tiff, .bmp and more), or as .svg (scalable vector graphics) format.

The larger width and height of the exported data, the better quality the exported image will be. While the high-quality and editable .svg format is suitable for small data plots (e.g. the Ri plot, Fig 34.) but not raw data plots (as displayed on the main plot) as the resulting files will be very large and take a long time to save.

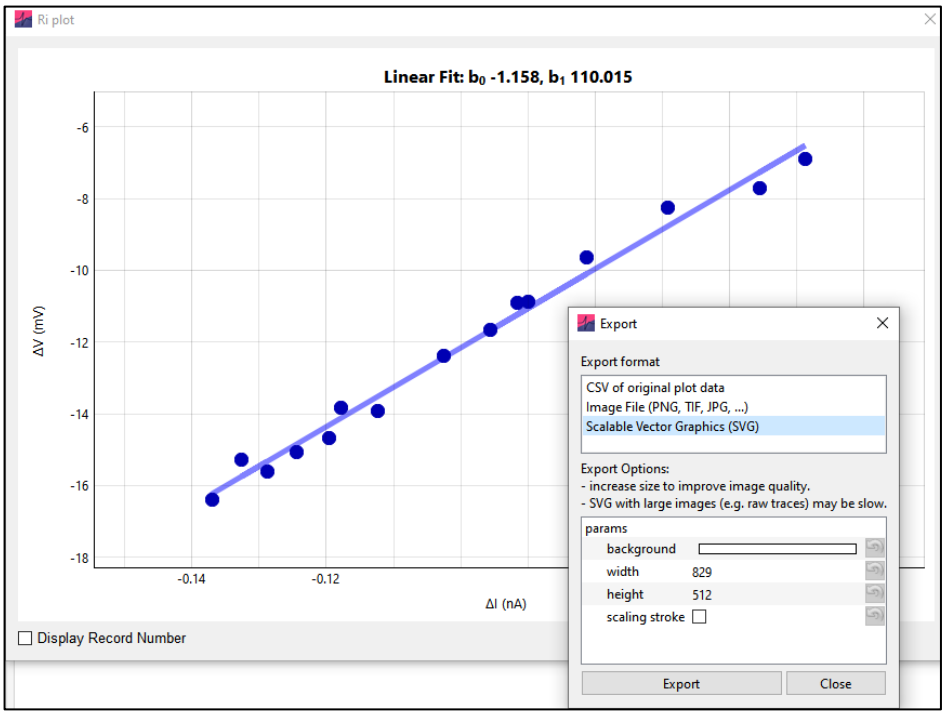

Figure 34. Plot Export Window. Right-clicking on any plot and choosing the 'Export' option will open the Export window.

19. Legacy Options

Options from previous versions (< v2.3.0) of Easy Electrophysiology are provided to allow backwards compatibility with previously analyzed datasets. This also functions to provide additional options that may work better in some specific situations.

Legacy baseline – no enhanced position based on rise time: This option estimates the event baseline, and is the first stage if baseline detection method in the current version. When this option is selected, the second stage will not be run (in which the event baseline is shifted to the event foot as calculated by Jonas et al., (1993), as the intersection between the baseline and a straight line drawn through the 20-80 rise time) (see *Baseline Options (Analyze windows only)*). This is how baseline detection was implemented prior to v2.3.0-beta.

Legacy baseline – no adjustment for prior events: Since v2.3.0-beta, the event baseline is adjusted if the previous event peak is after the current event baseline. This can sometimes occur for very closely spaced events. Selecting this option will turn this feature off.

Legacy decay detection – closest point to baseline: Selecting this option will use the previous decay endpoint detection method, in which the closest point to the baseline within the *decay search period (ms)* is used. No adjustment is made for the position of the next event.

Legacy frequency data bin-sizing: prior to version v2.3.0-beta, a data range slightly larger than the true range was used for cumulative probability data. This was removed to make bin-sizing clearer to the user.

20. Keyboard Shortcuts

ESC – will close any open window (except the main window)

U – undo the zoom

Z – Pan the graph one window right (useful when zoomed in)

X – Pan the graph one window left (useful when zoomed in)

A or S – hold while clicking plot zoom buttons for axis specific zooming (A: x-axis, S: y-axis)

Events Refine / Analysis Window

When an analysis has been run, keyboard shortcuts are available for scrolling through, moving to and deleting events. When the Refine / Events Analyze window is open:

Right arrow – scroll to the next event

Left arrow – scroll to the previous event

Up – go to the event (the main plot will be moved so that the selected event is in the center)

Space – delete the selected event from analysis

21. License

License is available at https://www.easyelectrophysiology.com/license and with the Easy Electrophysiology distribution

References

Clements, J. D., & Bekkers, J. M. (1997). Detection of spontaneous synaptic events with an optimally selected template. Biophysical Journal, 73(1), 220-229.

Cotterill E, Charlesworth P, Thomas CW, Paulsen O, Eglen SJ. (2016). A comparison of computational methods for detecting bursts in neuronal spike trains and their application to human stem cell-derived neuronal networks. Journal of Neurophysiology, 116(2):306-21.

Garcia, S., Guarino, D., Jaillet, F., Jennings, T., Pröpper, R., Rautenberg, P. L., Rodgers, C. C., Sobolev, A., Wachtler, T., Yger, P., & Davison, A. P. (2014). Neo: An object model for handling electrophysiology data in multiple formats. Frontiers in Neuroinformatics; 8:10.

Guzman, S, J., Schlögl, A., & Schmidt-Hieber, C. (2014). Stimfit: quantifying electrophysiological data with Python. Frontiers in Neuroinformatics; 8: 16.

Jonas, P., Major, G., Sakman, B., (1993) Quantal components of unitary EPSCs at the mossy fibre synapse on CA3 pyramidal cells of rat hippocampus. The Journal of Physiology; 472: 615-663.

Pasquale V, Martinoia S, Chiappalone M. (2010). A self-adapting approach for the detection of bursts and network bursts in neuronal cultures. Journal of Computational Neuroscience, 29, 213–229.

Pernia-Andrade, A., Goswami, S. P., Stickler, Y., Frobe, U. Schlogl, A., & Jonas, P. (2012). A Deconvolution-Based Method with High Sensitivity and Temporal Resolution for Detection of Spontaneous Synaptic Currents In Vitro and In Vivo. Biophysical Journal. 103(7), 1429-1439.

Rossokhin, A. V., & Saakian, I. Z. (1992). Dependence of the nerve pulse generation threshold on the ability of forming an input signal based on the Hodgkin-Huxley model. Biofizika; 37(6): 1071–1075.

Sekerli, M., Del Negro, C. A., Lee, R. H., & Butera, R. J. (2004). Estimating action potential thresholds from neuronal time-series: New metrics and evaluation of methodologies. IEEE Transactions on Biomedical Engineering; 51(9): 1665–1672.

Shinomoto. S., Shima, K., & Tanji, J., (2003). Differences in spiking patterns among cortical neurons. Neural Computation; 15, 2823–2842.

Appendix I – Mathematical Formulas

1 Curve Fitting Functions

1.1 Monoexponential

$$f(x) = b_0 + b_1 e^{\frac{-x}{\tau}}$$

1.2 Biexponential (decay)

$$f(x) = b_0 + b_1 e^{\frac{-x}{\tau_1}} + b_2 e^{\frac{-x}{\tau_2}}$$

1.3 Biexponential (event)

$$f(x) = b_0 + b_1 (1 - e^{\frac{-x}{rise}}) e^{\frac{-x}{decay}}$$

1.4 Triexponential

$$f(x) = b_0 + b_1 e^{\frac{-x}{\tau_1}} + b_2 e^{\frac{-x}{\tau_2}} + b_3 e^{\frac{-x}{\tau_3}}$$

where x is a vector of times, normalised by subtracting the first timepoint.

1.5 Starting Estimates for Exponential Fits

$$b0 = \frac{1}{n} \sum_{i=1}^{n} (D_i)$$

$$b1 = D_n - D_1$$

$$tau = \frac{t_{half}}{\log(2)}$$

where D is a $1...n$ vector of data-points to fit, and t_{half} is the time from D_1 to half-amplitude.
Biexponential (decay) taus are weighted: $tau1 = 0.1tau,\ tau2 = 0.9tau$
Triexponential taus are weighted: $tau1 = tau2 = tau3 = tau/3$.

2 Action Potential Threshold Methods

Where v is a vector of recorded Vm samples, Vm_{thr} is the estimated action potential threshold, and LB is a lower bound set by the user:

2.1 Method I (Sekerli et al., 2004)

$$Vm_{thr} = \arg\max_{v} f(v) \ where \ v' >= LB,$$

$$f(v) = \frac{v''}{v'}$$

2.2 Method II (Sekerli et al., 2004)

$$Vm_{thr} = \arg\max_{v} f(v) \ where \ v' >= LB,$$

$$f(v) = \frac{v'''v' - (v'')^2}{(v')^3}$$

2.3 Leading Inflection

$$Vm_{thr} = \arg\min_{v} v'$$

2.4 Maximum Curvature

Introduced in Rossokhin and Saakian (1992):

$$Vm_{thr} = \arg\max_{v} f(v),$$

$$f(v) = v''(1 + (v')^2)^{\frac{-3}{2}}$$

3 Polynomials for Detrending

The polynomial equations for *Detrending* and *Events* detection (e.g. *Curve* thresholding) are in the standard form:

$$a_0 + a_1 x ... \ a_n x^n$$

4 Spike Frequency Accommodation

Spike frequency accommodation (SFA) is calculated from the inter-spike intervals in a train of action potentials. ISI_{first} is the first ISI in the spike train, ISI_{last} is the final ISI of the spike train.

4.1 Divisor Method

$$SFA_{divisor} = \frac{ISI_{first}}{ISI_{last}}$$

4.2 Local Variance

The local variance (LV) method was introduced in Shinomoto et al., 2003:

$$SFA_{LV} = \frac{1}{n-1} \sum_{i=1}^{n-1} \frac{3(ISI_i - ISI_{i+1})^2}{(ISI_i + ISI_{i+1})^2}$$