Eventer

Software for the detection of spontaneous synaptic events measured by electrophysiology or imaging

v1.1

Authors
Andrew Penn
Giles Winchester
Samuel Liu
Oliver G. Steele
Wajeeha Aziz

PLEASE CITE:
Winchester, G., Liu, S., Steele, O.G., Aziz, W. and Penn, A.C. (**2020**) *Eventer: Software for the detection of spontaneous synaptic events measured by electrophysiology or imaging*. http://doi.org/10.5281/zenodo.3991677

Table of Contents

1. Introduction

Eventer is a programme designed for the detection of spontaneous synaptic events measured by electrophysiology or imaging. The software combines deconvolution for detection and variable length template matching approaches for screening out false positive events. Eventer also includes a machine learning based approach, utilising a Random Forest algorithm that allows users to train a model fitting their own 'expert' selection criteria from exemplar data increasing the reproducibility of the analysis procedure. The software is a MATLAB app but it has also been compiled as standalone applications for 64-bit Windows, Mac and Linux. These standalone applications only require users to download and install freely available MATLAB runtime environments.

2. System Requirements

Eventer requires the freely available MATLAB runtime environment[1] installed in one of the following operating systems: Windows 64-bit, Linux 64-bit, Intel Mac 64-bit. Please see documentation for the MATLAB runtime environment for details on runtime-specific system requirements.

3. Installation

3.1 Windows & Linux

The installer for both Windows and Linux operating systems will detect whether the correct MATLAB runtime is installed and defined in the environmental variables, automatically downloading and installing the runtime if it is not. The version of the MATLAB runtime used to compile the standalone application is provided in the name of the installer. Note that installer for Eventer is not currently digitally signed and so you can expect Windows to raise a security concern when starting the installer.

3.2 Mac

For Mac OSX the .dmg contains both the application and an install.txt file containing instructions to download and install the correct runtime. The version of the MATLAB runtime used to compile the standalone application is provided in the name of the installer. Depending on your version of macOS, you may need to enable the installer to open in the Security and Privacy settings for it to proceed.

3.3 Linux

[1] https://uk.mathworks.com/products/compiler/matlab-runtime.html

Load a terminal in the directory of the install file and install Eventer using the following commands:

```
chmod a+x Eventer-v1.1.0-Linux64-R2019a.install
 ./Eventer-v1.1.0-Linux64-R2019a.install
```

You should be able to choose to install to any location within your home directory. Make a note of where you install Eventer and the MATLAB Runtime. If you intend to install to the default system directories you should use the sudo command version instead.

```
sudo ./Eventer-v1.1.0-Linux64-R2019a.install
```

Once installed, run eventer from the command line as follows:

```
path/to/eventer/application/run_Eventer.sh path/to/MATLAB_Runtime/v96
```

3.4 Tested Operating Systems

There are atleast some versions of Eventer compatible with:

Windows: 7 and 10
MacOS: High Sierra (10.13.6), Catalina (10.15.7), Big Sur (11.2.3)
Linux: Ubuntu 14.04.6 (Trusty Tahr), Ubuntu 20.0.4 (Focal Fossa)

4. Supported Data formats

4.1 Importing Data

Eventer can analyse data from episodic or continuous recordings. The following *binary* and *text* data formats can be loaded by Eventer:

Eventer analysis (*.evt)
ACQ4 binary (hdf5) files (*.ma)
Axon binary files 1 and 2 (*.abf)
Axograph binary file (*.axgx, *.axgd)
CED Signal binary files (*.cfs)
CED Spike2 binary files (*.smr)
ephysIO HDF5/MATLAB binary files (*.phy)
HEKA PatchMaster,Pulse and ChartMaster binary files (*.dat)
Igor binary wave files (*.ibw, *.bwav)
Igor Packed experiment binary files (*.pxp)
GINJ2 MATLAB binary files (*.mat)
LabVIEW Signal Express TDMS binary files (*.tdms)
Stimfit binary (hdf5) files (*.h5)
WinEDR binary file (*.EDR)
WaveSurfer binary (hdf5) files (*.h5)
WinWCP binary file (*.wcp)

Axon text files (*.atf)
Comma-separated values text files (*.csv)
Igor text files (*.itx, *.awav)
Tab-delimited text files (*.txt)

4.2 Exporting Data

Additionally, Eventer's output data can be saved in the following *binary* and *text* data formats:

Eventer analysis (*.evt)
ephysIO HDF5/MATLAB binary files (*.phy)
stimfit HDF5 binary files (*.h5)

Axon text files (*.atf)
Comma-separated values ASCII text files (*.csv)
Igor text files (*.itx, *.awav)
Tab-delimited ASCII text files (*.asc) (suitable for import into WinWCP or WinEDR)
Tab-delimited ASCII text files (*.txt)

Eventer can both automatically unzip and export files with GNU zip compression, thus allowing easier handling of larger datasets. However, this option is unavailable for the following exceptions:

- The ephysIO file format (*.phy), the native Matlab 7.3 (HDF5) data format of Eventer. It is a compact binary format.

- Eventer *analysis.evt* files contain analysis settings for a particular dataset. Loading an *analysis.evt* file loads all files and settings saved from an analysis using Eventer. Note that the *analysis.evt* file is saved together with a human readable *filepaths.txt* file. Both of these files are required to reload an analysis. *Analysis.evt* and *filepaths.txt* cannot be loaded if they are compressed.

- *Presets* files are matlab scripts (*.m) that contain only generic analysis settings. Loading a *presets* script loads only settings (not files). For more on *presets*, see chapter 5.10

4.3 Exporting Figures

Eventer's figures can be exported in the following raster, vector and other formats:

BMP (24-bit) (*.bmp)
PNG (24-bit) (*.png)
TIFF (24-bit, not compressed) (*.tif)
TIFF (24-bit, LZW compressed) (*.tif)

EMF Windows metafile (*.emf)
EPS Encapsulated postscript level 3 (colour) (*.eps)
SVG Scalable Vector Graphics (*.svg)

MATLAB figure (*.fig)

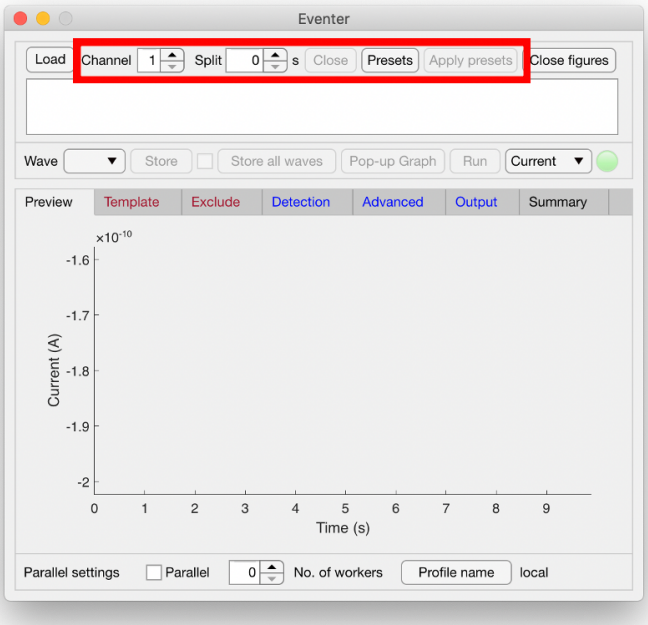
All raster images are saved at a resolution of 300 dpi (dots per inch). Data for all vector formats figures undergo data reduction before plotting and saving for more convenient figure editing by third party drawing applications.

5. Quick Start Guide

This chapter will show the user how to quickly analyse their data without the need for a detailed explanation of each feature. *Tip. Hover the mouse cursor over features of the GUI for tooltips.*

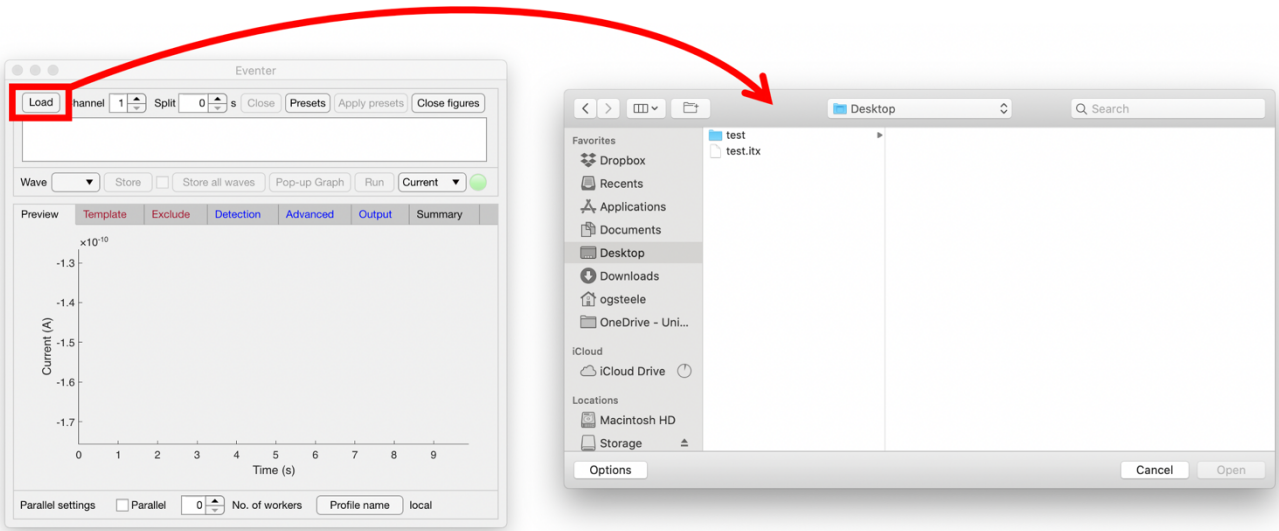
5.1 Preload settings

Choose your channel and split settings. Only some data formats allow for multiple recording channels, so unless you need to, the channel feature can be left at 1. Split should only be used if your data varies in length or if you want to split a continuous recording wave into smaller chunks for parallel processing.



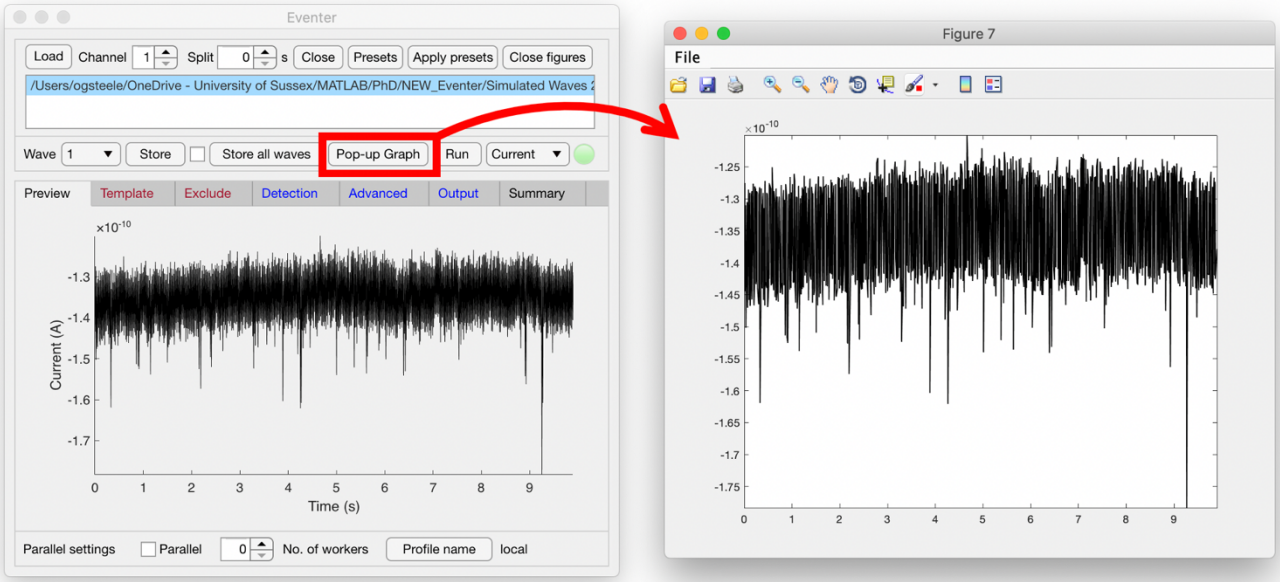
5.2 Loading your data

Choose which data you wish to analyse. Multiple data sets can be analysed at once. Eventer will merge these data sets together so ensure you are loading data that you want merged. To load multiple data sets to be merged, simply repeat the loading process shown below.



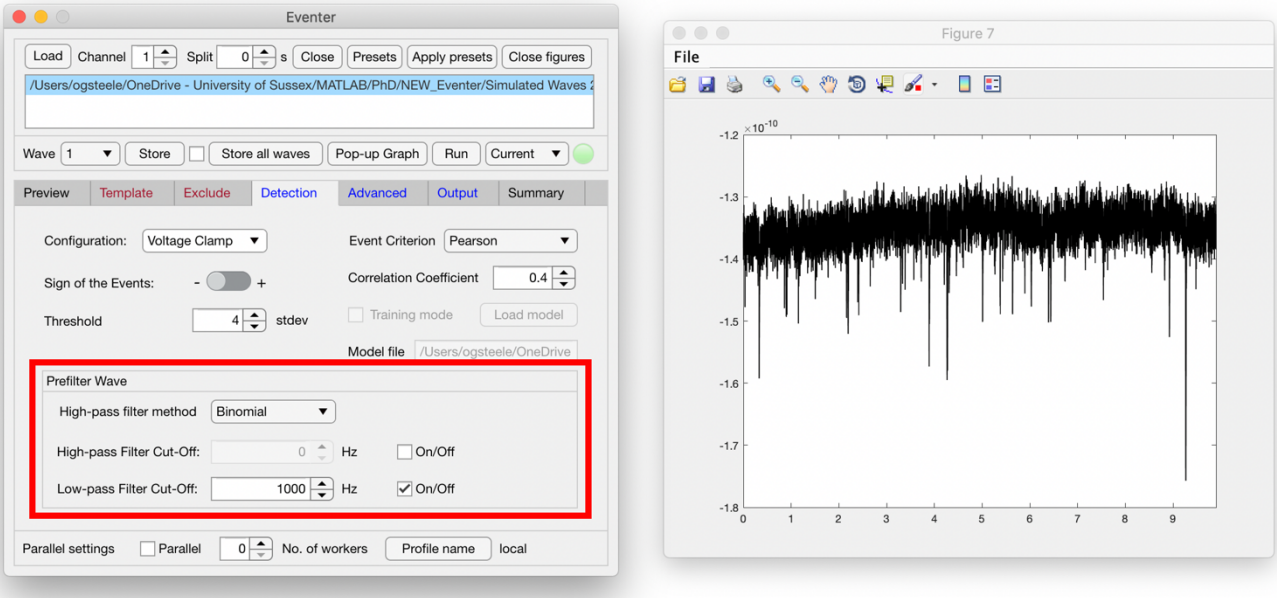
5.3 Visualising your data

Eventer includes a **preview tab** for immediate visualisation of your data upon loading. To inspect the data in more detail, the 'Pop-up Graph' should be used as shown here. This graph can then be zoomed and dragged to located individual events.



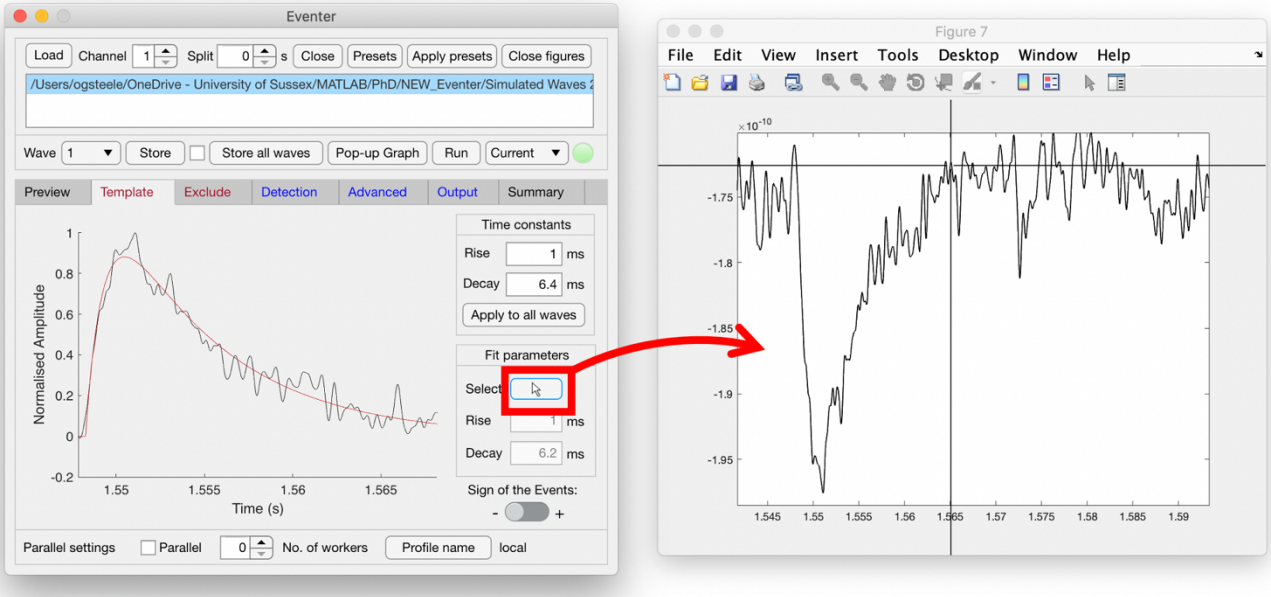
5.4 Prefiltering your data

It is possible that excess noise is still present in your data that may not have been isolated during data acquisition. To help visualise data, Eventer enables the user to pre-filter their data through high- and low-pass filter cut offs. Shown here, a 1 kHz low-pass filter is applied to the data in the **detection tab**.



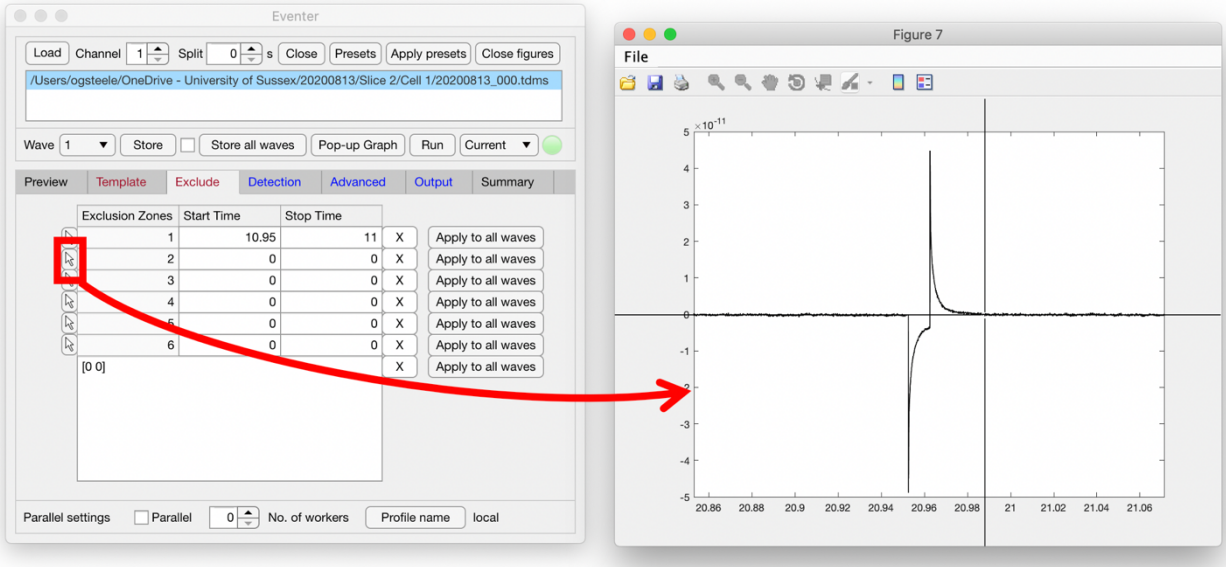
5.5 Defining your event template

In the template tab you can set a template event for the software to compare your data against. This can be set manually if the rise and decay parameters of your events are known, or can be done through opening the 'Pop-up Graph' as mentioned before and selecting the button highlighted here. This will allow the user to select two points; one before and one after the event of interest. The time constants will then be in the boxes below the highlighted button and should then be copied into boxes above before selecting the button 'Apply to all waves'.
It is worth locating an event in the 'Pop-up Graph' before defining the time constants and that the sign of your events is correctly stated in the detection panel.



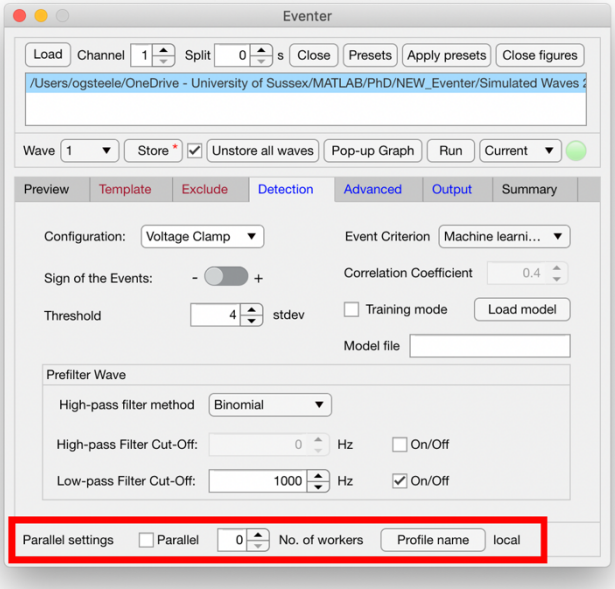
5.6 Excluding regions from your analysis

It is possible that there are regions of your data that you want to be excluded from the analysis. For example, this could be a particularly noisy section of recording or the inclusion of a test pulse as shown here. To exclude these regions, open the **exclude tab** and regions to be excluded can either be manually typed in or selected using the highlighted button. Selecting this button will bring up cursors similar to the template selection window.



5.7 Parallel Processing

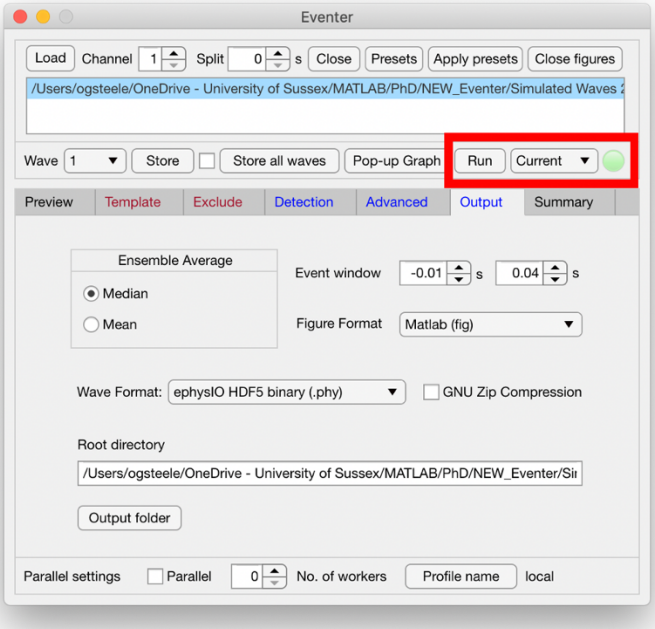
If you wish to use parallel processing which will speed up analysis on newer computers, you can choose the number of workers. The limit of how many workers you can have is dependent on how many cores and threads your CPU has. After the worker number has been chosen, check the parallel box and wait for the workers to initialize.



5.8 Running your analysis

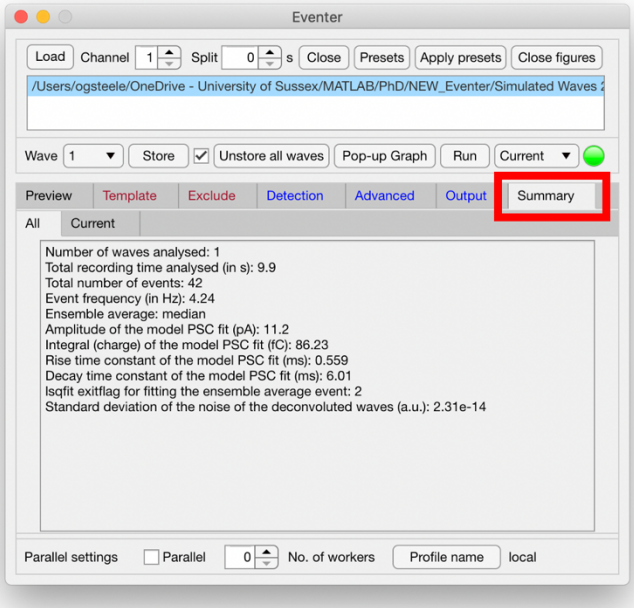
Once the user is happy with the settings for their analysis, open the **output tab** and ensure the data will be outputted into the desired output format and saved to the appropriate location. By default this will create a folder named 'eventer.output' in the current working directory that the data was located. Choose which waves you would like stored in the analysis by either going through each wave individually and selecting/deselecting the store checkbox, or click the store all waves button. Unless a new output folder is specified for new analysis on the same dataset, it will be overwritten when performing new analysis.

Changing the box on the right from 'Current' to 'Batch' will tell Eventer that the analysis can be performed on the batch of waves stored, rather than solely on the current wave. If the user has not loaded multiple files or split the recording then 'Current' and 'Batch' will both be the same. Having confirmed these settings the user can hit 'Run' and the light will turn green indicating analysis is in progress.



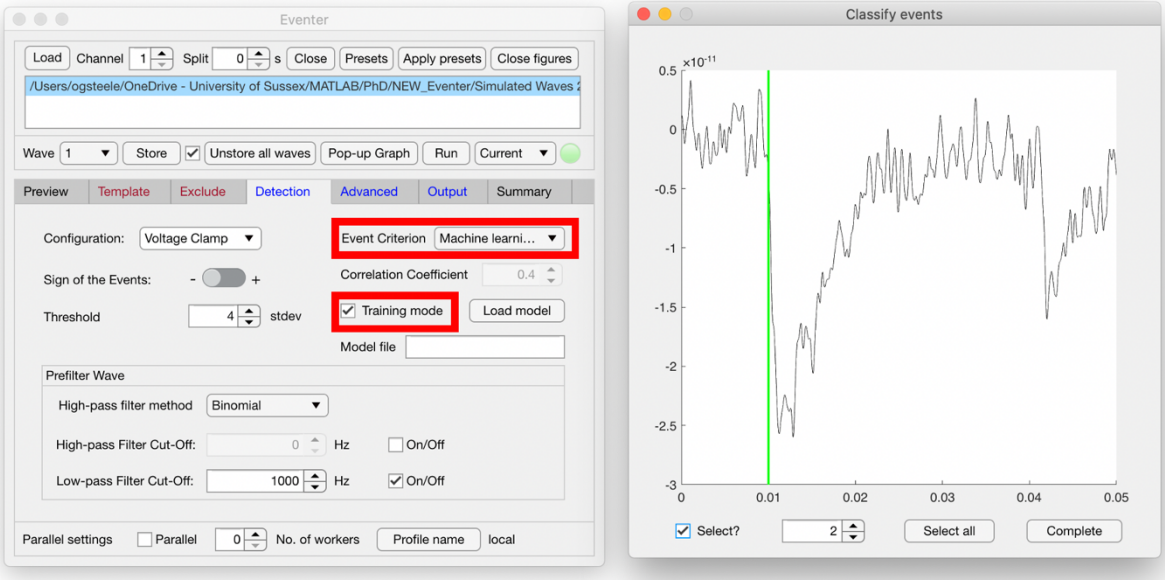
5.9 Accessing your results

After running the analysis, Eventer will open multiple figures (explained in 5.11.4). A quick summary of results is also available in the **summary tab** as shown here. The rest of your results will be saved in the root directory in the previously specified output folder.



5.10 Training a machine learning model

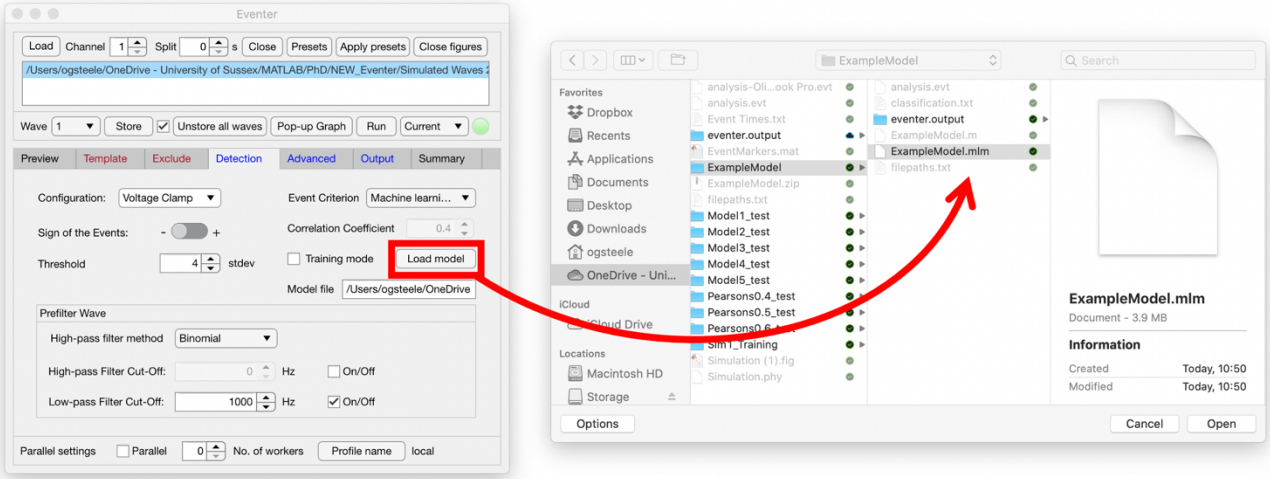
Eventer also includes an option to enable users train a machine learning model against an exemplary set of data. To do this, open the detection tab and change the event criterion to 'Machine Learning' before ticking the 'Training Mode' tickbox. Running the analysis now will open up a window that will ask users to classify events as either events or not. A pop-up window will allow the user to also name their model. Move through the events selecting or deselecting events where appropriate until all events are classified then click complete. When selecting whether an event is indeed actually an event, it is



important that the green line shown above is exactly at the start of where you perceive the event to be otherwise the model may become inaccurate. Upon completing your classification task, you will be presented with a plot showing the out-of-bag classification error which can briefly be described as a measure of the prediction error using bootstrap aggregating (bagging) to subsample data samples used for training. The prediction error should then stabilize well before reaching 128 trees (which is the default number of trees used by the implementation of Random Forests in Eventer).

5.11 Using your model

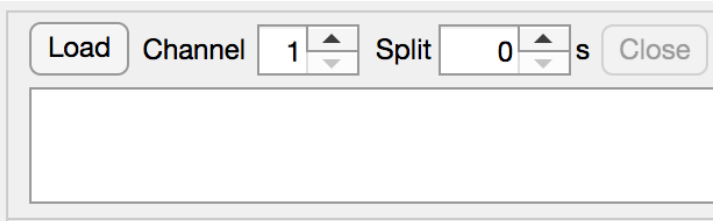
To then use the trained model on subsequent analysis, the user can then deselect the training mode and select load model in the detection tab. The trained model will be stored in the eventer output folder as a .mlm file as shown here. This model is now available for use on a new dataset, distinct from the set it was trained on.



6. GUI Feature Reference

6.1 Wave Settings

These settings can be located at the top of the Eventer graphical user interface (GUI). They will alter how the data is extracted upon loading. They have to be set before loading the data. The channel determines the recording channel number for file types that allow multiple recording channels. The split settings splits the waves found in the loaded data into specifically sized chunks.



6.1.1 Channel Settings

The channel option determines which recording channel is selected by Eventer from the file. By default, Eventer will attempt to load the first recording channel. Eventer will override this setting and channel numbers will be ignored for loaded file types that do not support multiple recording channels.

6.1.2 Split Value

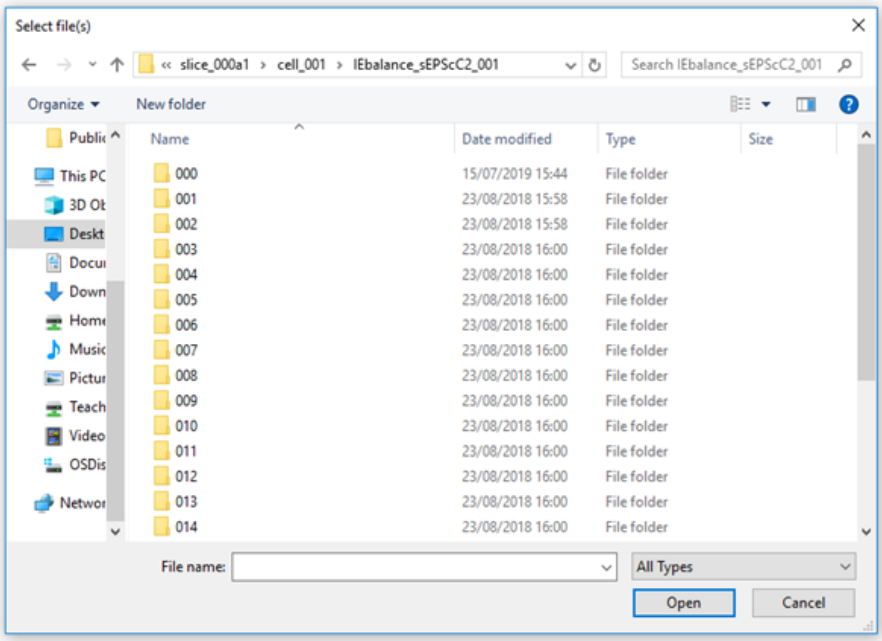
The split value option allows continuous data to be segmented into smaller equal lengths. This option should be utilized in two situations:

- When multiple sets of data are not of the same length to allow for batch analysis. **Eventer will not work for sets of data of varying length.**

- To split up long continuous data into smaller segments to allow for utilization of the parallel analysis feature. For more on parallel analysis see chapter 5.12

6.2 Loading Data

This feature can be located at the top-left side of the Eventer GUI. The load feature allows the user to locate data files for analysis. Pressing this button will open a file search dialogue box allowing for the user to search their directories for the data to be loaded. Additionally, files can be filtered by type (All Types by default). All the supported data formats can be found in chapter 4.

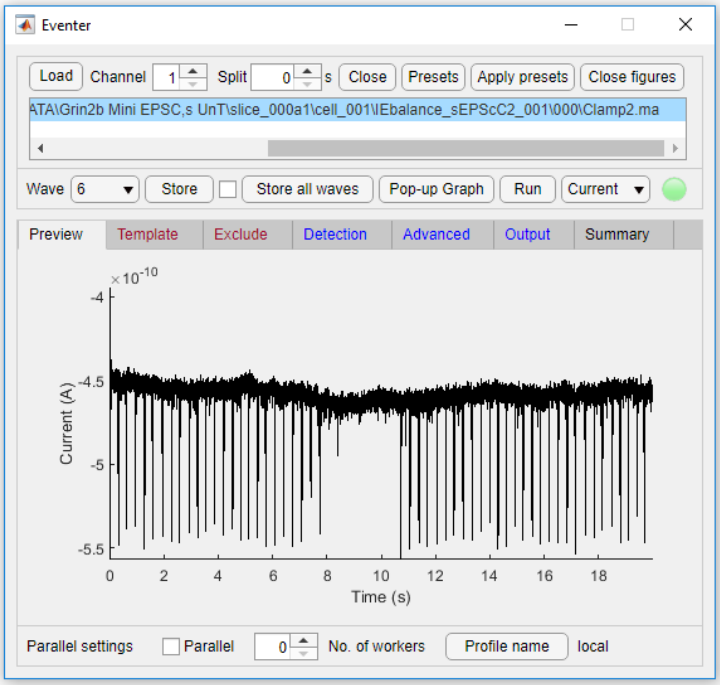


Load all of the data required before running event acquisition. Select the files you need and click open. Multiple files can be selected at once by clicking and dragging over them or clicking on each one while control is held. Each loaded piece of data will be shown in the file box. Each piece of data can have multiple waves within.

If the data is stored as ACQ4 binary (hdf5) files (*.ma), multiple pieces of data will be loaded if your first piece of data is in a file labelled 000, and your subsequent pieces of data are in files labelled 001, 002 and so on.

This is the format required for automatic .ma loading. Within each folder there is one piece of .ma data. Upon loading the data within the folder labelled 000, the data within each of the other folders will also be loaded.

If previous analysis has already been performed by eventer, this analysis can also be loaded the same way as new data. The previous analysis will be labelled *analysis.evt* and found in the same directory as the data it came from. This will load the analysis and all the settings that were used to analyse this data and such, adding onto this previous analysis is possible.



6.3 Preview

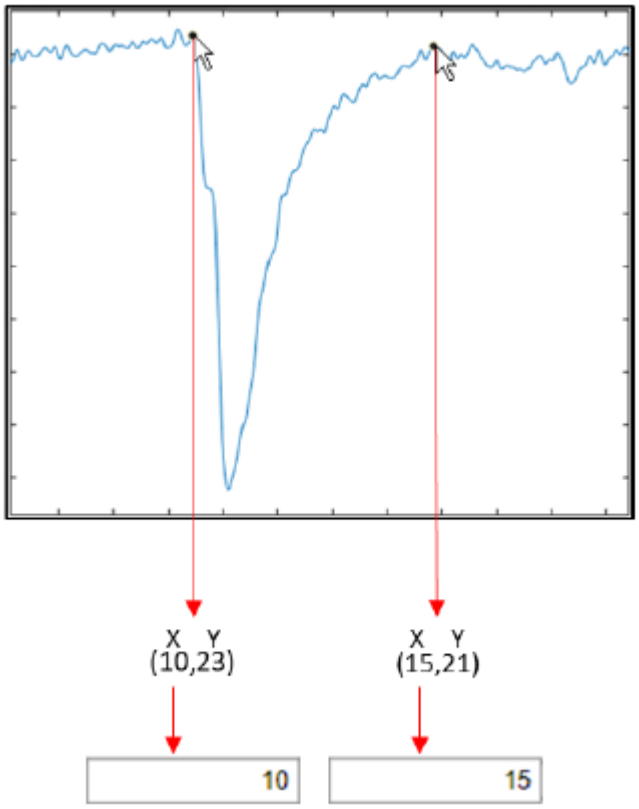
The **preview tab** allows the user to see the currently selected wave prior to running the event analysis. This can allow for the user to determine whether the wave is representative of the data and thus either included or ignored during event analysis by selecting or deselecting the store tickbox. Note that the preview image should only be used for a quick overview. If manipulation of the image is required, a faster, more detailed image can be loaded by clicking the pop-up graph button. For more on the pop-up graph see chapter 6.11.3

6.4 Template

The **template tab** allows the user to enter and refine their rise and decay time constant values. To enter time constant values simply enter their values (in ms) within the rise and decay numeric entry fields.

To further refine these constants a user should locate an exemplar event (see the exclude section of chapter 6.5 for steps to do this) using the pop-up graph. Once located, appropriate start and stop times for the event should be selected with the cursor button. Eventer will then run calculations for the appropriate time constants for this event and display them below the cursor button.

This process can be repeated as many times as desired until the user is satisfied with their time constants for their data.



The sign of the events switch allows the user to set the sign of event peak deflections as either *positive* (+) or *negative* (-). This will affect the application of analysis protocols so ensure the sign setting accurately reflects the data in question. By default, this setting is set to *negative*. Changing this will also change the switch found in the **detection tab** (chapter 6.6)

To save the template for a single wave, press the store button. Note this will also add the wave to the list of waves that will make up the merge. Templates can also be applied to all waves by pressing the apply to all waves button. Note this will also add all of the waves to the list of waves that will make up the merge.

6.5 Exclusion

The **exclusion tab** allows the user to select data ranges to remove from analysis by entering a start time and stop time within the exclusion zone table. For easier determination of ranges the arrow cursor can be used to click on the pop-up graph to visually select exclusion ranges. Exclusion zones can be removed at any time via the X button.

To save an Exclusion zone for a single wave, press the store button. Note this will also add the wave to the merge. Exclusion zones can also be applied to all waves by pressing the apply to all waves button, for cases where electrophysiology equipment causes artifacts at a constant point in recording this option should be employed. Note that this will also add all of the waves to the merge. Also note that if when the Exclusion zone is applied to all waves, it is also applied to the same time range on split waves meaning that on waves without an artifact, a non artifact zone will be excluded. It may be beneficial to the user to manually exclude each artifact if using the split function.

The Eventer UI allows for a total of 6 exclusion zones within the exclusion table, however, if more exclusion zones are required, they can be entered in the text box below in the form;

'[StartTime, EndTime]' , allowing for as many further zones as required.

6.6 Detection

The **detection tab** contains settings that change how the data is treated during analysis. Within this tab options for clamp configuration, sign of the events, detection threshold, event criterion type, correlation coefficient and prefiltering options can be found.

6.6.1 Configuration

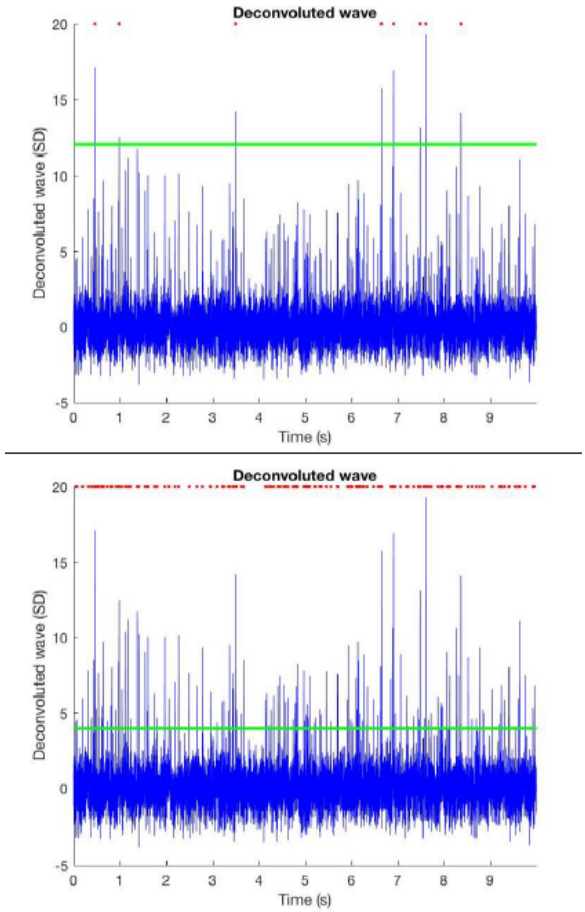
This setting allows the user to set the configuration of the recording wave to either *voltage clamp* (VC) or *current clamp* (CC). By default, waves will be treated as voltage clamp data.

6.6.2 Sign

Allows the user to set the sign of event peak deflections as either *positive* (+) or *negative* (-). This will affect the application of analysis protocols so ensure the sign setting accurately reflects the data in question. By default, this setting is set to *negative*. Changing this will also change the switch in the template tab (chapter 6.4).

6.6.3 Threshold

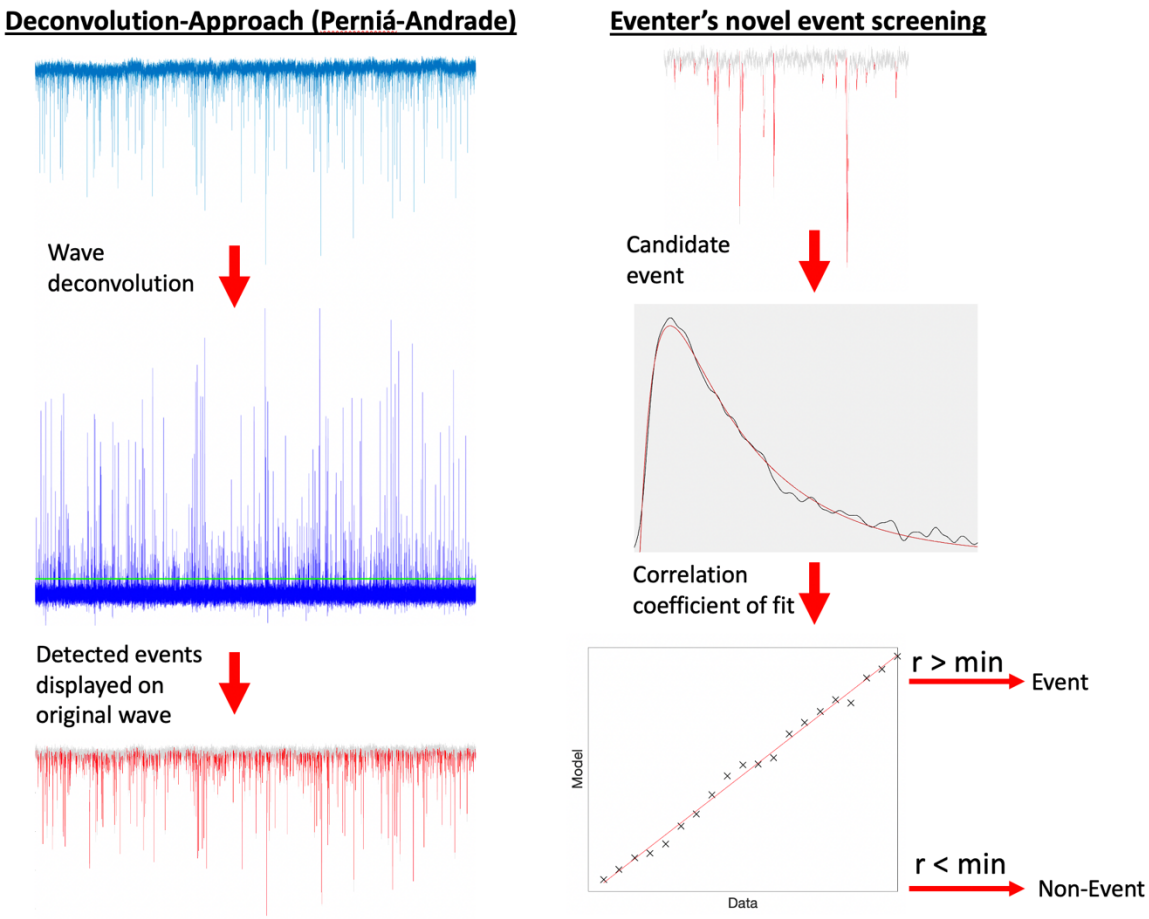
The threshold setting allows users to determine the value by which delta-like waves must exceed to classify as a candidate event. The value for this setting (SD) reflects the standard deviation (SD) of the noise of the deconvoluted wave multiplied by a scale factor (SF).



In the case of low-noise data this threshold can be reduced to detect low-magnitude events, increasing true positive (TP) detection rate of Eventer. Inversely, in the case of high-noise data this threshold can be increased to help reduce the inclusion of noise as candidate events, decreasing the false positive (FP) detection rate of Eventer.

By default, the threshold value is set to 4SD, immediate testing suggest appropriate values from 4-6SD achieve the best trade-off between TP and FP for normal noise-level data. If known, absolute values can also be entered.

6.6.4 Event Criterion



This option allows the user to choose between **Pearson's Correlation Coefficient** or **Machine Learning** to screen for true events from proposed candidate events.

Pearson's Correlation Coefficient

The Pearson's Correlation Coefficient option compares each candidate event determined via deconvolution analysis, with the template and gives a coefficient based on the deviation of the event from the template. A number closer to 1 indicates a higher correlation. The default minimum correlation for event confirmation is 0.4, this can be altered depending on how stringent template correlation needs to be.

Machine Learning

The Machine Learning option applies a Random Forest algorithm that acts as a classification criterion for candidate events determined via deconvolution analysis. The Random Forest algorithm operates from a model that has been trained on exemplar data, for best results exemplar data should be from the same recordings as those being analysed but not being analysed in of themselves

To begin creating a model for the Random Forest model enable the training mode tick box and choose a name for your model. Note that a lower SD threshold should be used to ensure all events are found during deconvolution analysis (Chapter 6.6.3). After running Eventer in training mode, a graph will appear displaying all detections, these detections will include false positives which will have to manually be screened. Click start and reject (shortcut 'r') all detections that should not be classified as an event. Upon finishing, click complete and a graph of learning will appear.

This graph uses out of bag classification error. Briefly, this can be described as a measure of the prediction error using bootstrap aggregating (bagging) to subsample data samples used for training. The prediction error should then stabilize well before reaching 128 trees (which is the default number of trees used by the implementation of Random Forests in Eventer).

The model will be saved as a Machine Learning Model file (*.mlm*) and be contained within a folder labelled with the name of the model. It can be found in the same directory as your training data. The *.mlm* model can then be loaded to analyse any further data by pressing the load model button, then selecting the *.mlm* file. This *.mlm* file also contains all the settings so no *preset* is required.

6.6.5 Prefiltering

Allows filtering of preview wave to allow for easier event selection/exclusion.

6.7 Advanced

The advanced tab contains numerous options for a more experienced user to further tailor their analysis to fit their dataset.

6.7.1 Number of Taus

The number of taus option allows users to determine the number of time constants after the peak of the template to use when fitting the template to the detected events.

By default, this value is set to 0.4, which reflects to the time for the template to decay by 25% of the peak. This value ensures that peak measurements are not compromised at high event frequencies where event overlap is common.

As such this setting should be changed to suit data if the event frequency is generally known.

6.7.2 Baseline Time

The baseline time setting sets the length of time that Eventer will use as the pre-event baseline for the template fit (ms).

6.7.3 Exmode

This setting tells Eventer what to do with the first event preceding each exclusion zone.
Mode 1 (default): IEIs (inter event intervals) are calculated for these events from the last event preceding the exclusion zone under the assumption that no events occurred during the exclusion zone.
Mode 2: events are assigned an IEI of 'NaN', and as such are excluded from the merge and will not appear in the *ALL_events* output directory.

6.7.4 Deconvoluted Wave Signal-Processing

This setting includes the cut-off (Hz) for the low-pass and high-pass filters.
High-pass Filter – sets the –3 dB cut-off (Hz) of the low-pass median filter applied to the deconvoluted wave, where the filtered wave is then subtracted from the deconvoluted wave. As such this is essentially a high-pass filter.

Low-pass Filter – sets the –3 dB cut-off (Hz) of the low-pass binomial filter applied to the deconvoluted wave.

6.7.5 Levenberg-Marquardt Settings

This setting determines the damping factor used in the Levenberg-Marquardt ordinary nonlinear least-squares fitting procedures. By default, lambda is set to 1, the higher the value the more robust the fitting procedure is to the initial values but the greater the number of iterations is will use.

6.8 Output

The output tab contains several options for the user to tailor the format of the output of the Eventer analysis.

6.8.1 Ensemble Average

This setting allows the user to choose between displaying ensemble merged event data in either a mean or median format.

6.8.2 Event Window

This setting allows the user to determine the event window limits (s) for the conversion of the continuous wave to episodic data.

6.8.3 Figure Format

This setting allows the user to determine the output format of the figures generated by Eventer.

6.8.4 Wave Format

This setting allows the user to determine what format they wish Eventer to export the episodic data of all detected events in.

6.8.5 GNU Zip Compression

The GNU Zip compression option once enabled will compress all output data into a zip file to reduce size, this option is highly recommended for large sets of data.

6.8.6 Save Directory

Eventer will save every time it is run, producing an *eventer.output* folder along with *analysis* and *filepaths*. By default it will save these three items into the root directory which is the same directory as where the first piece of loaded data is regardless of whether any waves from there end up being stored. You can also have eventer create a folder in the directory in which to put the three items in by clicking on the output folder button and entering a folder name. Note that only alphanumeric characters are accepted as folder names (a-z, 1,0) , underscores (_) and dashes (-).

6.9 Summary

The summary tab displays information about the data after performing a run.

6.9.1 All Tab

The all tab displays data about the merge graph (labelled figure 6 after performing a batch or current run). The information it displays are as follows:

Number of waves analysed
- How many waves are stored in the merge

Total recording time analysed (in s)
- How long the waves are altogether (excludes exclusion zones)

Total number of events
- Total number of events that fit the template to the detection threshold determined in the detection tab.

Event frequency (in Hz)
- Average number of events detected per second

Ensemble average
- The chosen method of averaging

Amplitude of the model PSC fit (pA)
- The peak amplitude of the model wave

Integral (charge) of the model PSC fit (fC)
- Estimate of the area under the curve of the fitted response

Rise time constant of the model PSC fit (ms)
- Time it takes for the model wave to reach peak amplitude

Decay time constant of the model PSC fit (ms):

- Time it takes for the model wave to decay from its peak amplitude

lsqfit exitflag for fitting the ensemble average event

- see Appendix 1

S*tandard deviation of the noise of the deconvoluted waves (a.u.)*

- Standard deviation of the detected noise

6.9.2 Current Tab

The current tab displays data about the current selected wave after doing a current run. The information it displays are as follows:

Filename

– name of the file being recorded

Channel

- The recording channel number

Wave number

- Should be 1

Wave name

- The wave name. Should be in the format YWave###

Total number of events detected

- Total number of events that fit the template to the detection threshold determined in the detection tab.

Duration of recording analysed (in s)

- How long the wave is (excluding exclusion zones)

Mean event amplitude (pA)

- Average amplitude of events

Event frequency (in Hz)

- Average number of events detected per second

High-pass filter cut-off on deconvoluted wave (at -3 dB, in Hz)

- The frequency at which the –3dB cut-off of the high-pass filter was applied to the deconvoluted wave

Low-pass filter cut-off on deconvoluted wave (at -3 dB, in Hz)

- The frequency at which the –3dB cut-off of the low-pass filter was applied to the deconvoluted wave

Vector of model template time constants (in ms)

- Displays the rise and decay constants in a vector : '[rise, decay]'

Standard deviation of the noise of the deconvoluted wave (a.u.)

- Spread of the data in the deconvoluted wave

Scale factor of noise standard deviations for threshold setting:

- How much the standard deviation of noise is multiplied by to generate a threshold for events

Theoretical false positive detection rate before applying event criterion (in Hz):

- Theoretical rate that the noise will pass the threshold and give a false positive per second.

Sign of the event peaks I

- Can be positive (+) or negative (-)

Criterion used for event screening

- Can be Pearson's Correlation Coefficient or Machine Learning.

Maximum time after peak (expressed in time constants) used for template fit

The amount of time after the peak of an event that will be included when fitting back to events expressed in time constants (*no. of taus * tau decay*)

Dead time from event start required for template fit (ms)

- Amount of time before an event that is used to generate its baseline.

Minimum acceptable correlation coefficient for the template fit

- Minimum correlation coefficient (Pearson's or Machine learning) that will be accepted as an event

Episodic data window limits centred around each event

- Limits of the graphs before and after the event.

Sample rate of the recording (in kHz)

- Rate at which the data was sampled during recording

lsqfit exitflag for fitting the noise peak

– see Appendix 1

Exclusion zones

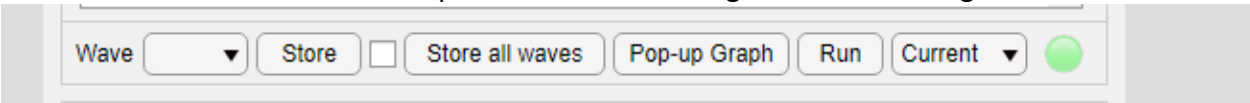
- Zones of the wave that are excluded from event detection

6.10 Presets

The presets button allows for presets to be loaded and saved. A preset will contain all settings that can be changed in eventer with the exception of any exclusion zones as these will vary wave to wave and potentially recording to recording. All preset files will have the extension *.m*

6.11 Session Settings

This bar contains buttons and drop-down boxes to change essential settings.



6.11.1 Wave

The wave drop-down box allows the current wave to be changed. After it has been changed, settings for that individual wave can be changed.

6.11.2 Store

The store button will save any changes made to that wave. If there are unsaved changes a red asterisk will appear in this box prompting the user to press it. Pressing this button will also check the store checkbox showing that the selected wave has been added to the merge list.

The store all waves button will check the checkbox for all waves loaded. Only the waves that want to be saved should be stored so ensure that all waves are valid before pressing the store all waves button.

6.11.3 Pop-Up Graph

The pop-up graph button will display a widow displaying the current wave. This should be used for manual detailed analysis of the wave and should also be used to zoom into a region of interest before selection (template event or exclusion zone)

6.11.4 Run

The run button will activate eventer with the current chosen settings. It can be run in Current mode where only the current wave will be analysed and added to the merge. Upon completion, 6 figures will appear;

Figure 1 will display a histogram of all the points of the deconvoluted wave (figure 2) along with the threshold line.

Figure 2 will display the deconvoluted wave across time along with the threshold line

Figure 3 will display the average shape of the events in blue as well as showing each event in grey.

Figure 4 will display a comparison between the template and the average event.

Figure 5 will display the original wave with the events highlighted

Figure 6 will display the average shape of the events of the whole merge in blue as well as showing each event of the whole merge in grey and a curve to fit the average curve in red. Note that this graph may take a while to load if there are many waves in the merge.

Eventer can also be run in batch mode where all the waves that have been selected to be stored will be analysed and added to the merge graph at once.

The results of the analysis will be stored in the folder where the first trace of data was loaded from. Within the generated folder called *eventer.output,* there will be:

ALL_events: This is where the data for the merge is.
Data_ch#.YWave###: This is where the data for wave ### is.

Within All_events there will be:

A copy of Figure 6 in the img folder
features, IEI (Inter Event Intervals), peak (Amplitude of events) in the txt folder
_offset (Should be hidden)
_parameters (Should be hidden)
Ensemble_average (Average trace of events)
Event_data (compressed figure 6, opable with ephysIO)
Fit smoothened out average
Residuals average minus smoothened
summary (A copy of the All summary)

Within each wave folder, there will be:

Copies of figures 1-5 in the img folder
features, IEI (Inter Event Intervals), peak (Amplitude of events), times (times of events) in the txt folder
_offset (Should be hidden)
_parameters (Should be hidden)
Ensemble_average (Average trace of events)
Event_data (compressed figure 6, opable with ephysIO)

summary (A copy of the Current summary)

6.12 Parallel

The parallel bar at the bottom allows for the use of parallel processing assuming your computer has the ability for parallel processing (multiple CPU cores), it will be faster to use this.
The number of workers is the number of workers (cores) that you wish to designate to event analysis. Set this before checking the Parallel processing box. Upon checking the parallel box, the workers will initialise which may take several minutes. After the workers have been initialised, any further batch runs will be significantly quicker (disk speed and memory dependant).
The profile name button allows for the use of clusters if a profile had previously been set in Matlab. The profile can be found and the cluster used.

Appendix

Appendix 1: ephysIO file format

The preferred ephysIO format (.phy) is a simple HDF5 format that is created for convenience using the matlab save command (Matlab file version 7.3). It is designed for efficient data storage, where the int16 data array format is suitable for most applications.

In the HDF5 file, the root group members are the names of the variables used to retrieve the data. These root members are:

```
/array  (int16 or int32) % Raw data (row major order, see below)
/start  (single)         % Start value for each wave of the actual data
/scale  (uint8)          % Power of two exponent for data scaling
/xdiff  (double)         % Sampling interval
/xunit  (uint16)         % Character of unit of the x-dimension wave
/yunit  (uint16)         % Character of unit of the y-dimension waves
/xname  (uint16)         % Character array of the name for the x-dimension
/names  (uint16)         % Character array of wave names (row major order)
/notes  (uint16)         % Character array of additional metadata
/saved  (double)         % Date and time that the file was saved
```

Note that Matlab 7.3 files use Unicode encoding of characters.

This simple file structure is capable of storing multiple fixed-length waves, which could represent multiple recording sweeps for a single recording channel. Different recording channels should be saved in separate files. In order to support variable-length recording sweeps, ephysIO performs padding on the waves to make them the same length.

The data array stored by ephysIO in the HDF5 files is the first derivative scaled, stored as integers and transposed. Loading the HDF5 files using ephysIO will return the proper (rescaled) data in the 'array' variable. If extracting the raw data directly from the HDF5 file then the following retransformation must be performed:

```
> scale=double(h5read('test.phy','/scale'));
> start=double(h5read('test.phy','/start'));
> array=double(h5read('test.phy','/array'));
> scale = 2.^(scale*ones(1,size(array,2)));
> array = array./scale;
> array = cat(2,start,array);
> array = transpose(cumsum(array,2));
```

Note that the above transformation is incompatible with NaN data values. Note also that on older versions of Matlab, the matlab variables will be saved with the native matlab file format instead of the HDF5 format.

Appendix 1: Exit flags for Levenberg-Marquardt (lsqfit)

1	Function converged to a solution x
2	Change in x was less than the specified tolerance
3	Change in the residual was less than the specified tolerance
4	Magnitude of search direction was smaller than the specified tolerance
0	Number of iteration exceeded options.MaxIterations or number of function evaluations exceeded options.MaxFunctionEvaluations
-1	Output function terminated the algorithm
-2	Problem is infeasible: the bounds lb and ub are inconsistent