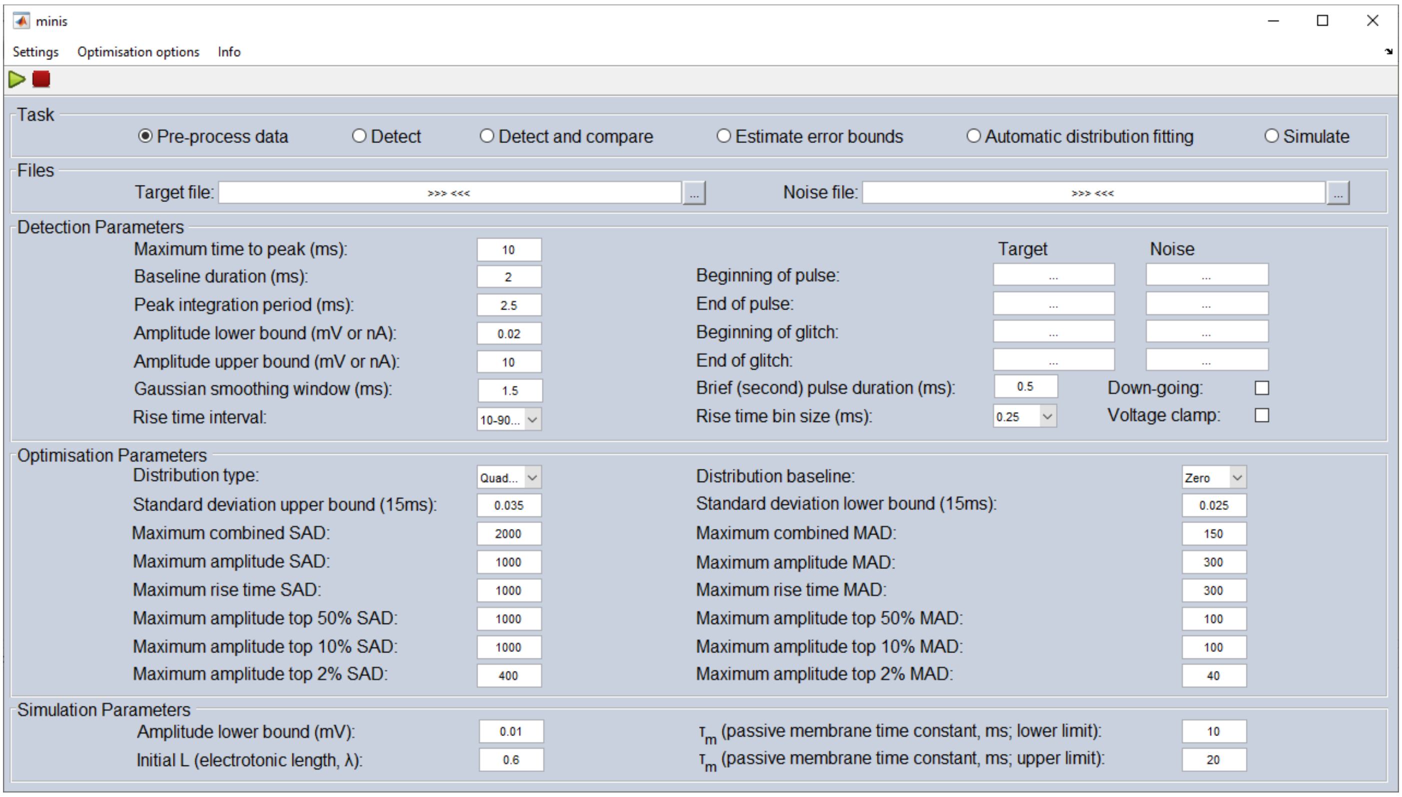
‘minis’ documentation



Task

● Pre-process data ○ Detect ○ Detect and compare ○ Estimate error bounds ○ Automatic distribution fitting ○ Simulate

Files

Target file: >>> <<< ...
Noise file: >>> <<< ...

Detection Parameters

Maximum time to peak (ms):	10
Baseline duration (ms):	2
Peak integration period (ms):	2.5
Amplitude lower bound (mV or nA):	0.02
Amplitude upper bound (mV or nA):	10
Gaussian smoothing window (ms):	1.5
Rise time interval:	10-90...

	Target	Noise
Beginning of pulse:	...	...
End of pulse:	...	...
Beginning of glitch:	...	...
End of glitch:	...	...

Brief (second) pulse duration (ms): 0.5
Rise time bin size (ms): 0.25
Down-going: □
Voltage clamp: □

Optimisation Parameters

Distribution type:	Quad...
Standard deviation upper bound (15ms):	0.035
Maximum combined SAD:	2000
Maximum amplitude SAD:	1000
Maximum rise time SAD:	1000
Maximum amplitude top 50% SAD:	1000
Maximum amplitude top 10% SAD:	1000
Maximum amplitude top 2% SAD:	400
Distribution baseline:	Zero
Standard deviation lower bound (15ms):	0.025
Maximum combined MAD:	150
Maximum amplitude MAD:	300
Maximum rise time MAD:	300
Maximum amplitude top 50% MAD:	100
Maximum amplitude top 10% MAD:	100
Maximum amplitude top 2% MAD:	40

Simulation Parameters

Amplitude lower bound (mV):	0.01
Initial L (electrotonic length, λ):	0.6
τ_m (passive membrane time constant, ms; lower limit):	10
τ_m (passive membrane time constant, ms; upper limit):	20

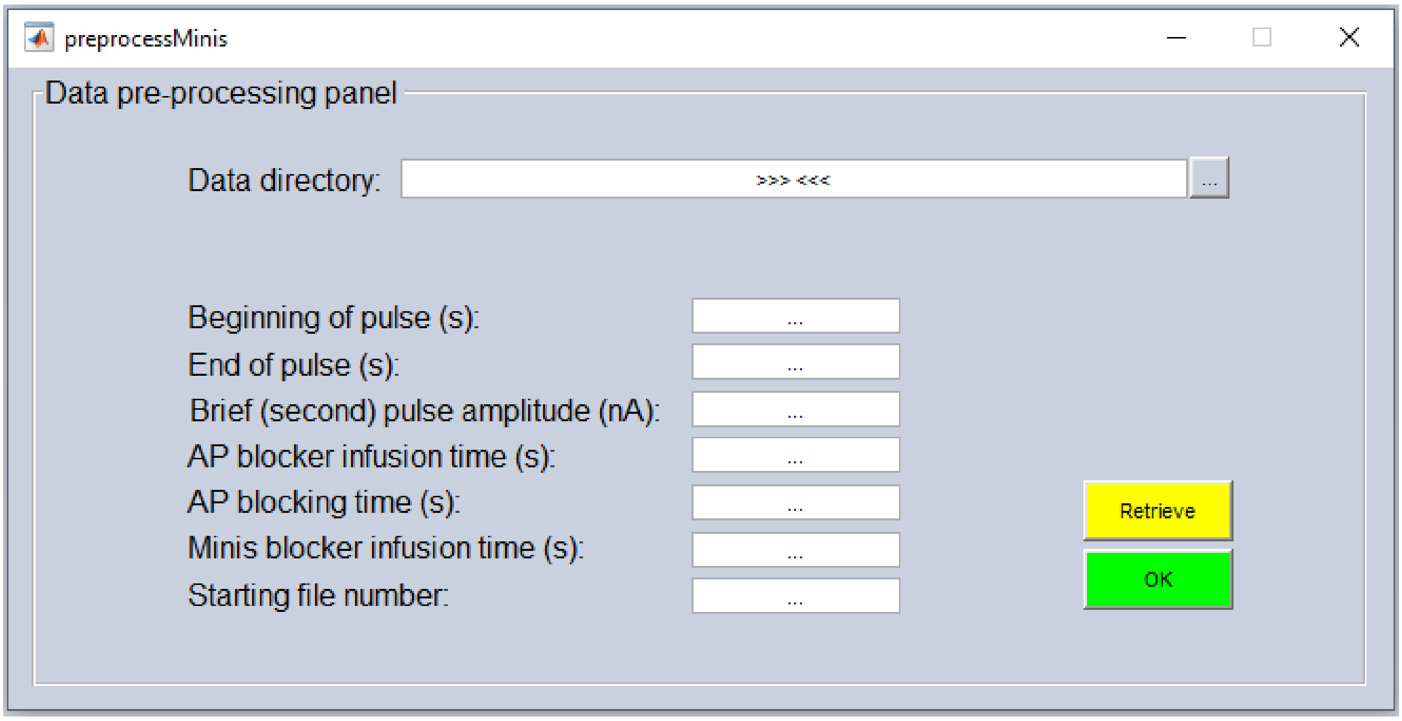
Figure 1: The main window

In the Task panel (Figure 1) you can choose how you want to run the ‘minis’ software. Five different options are available:

- Pre-process data
- Detect
- Detect and compare
- Estimate error bounds
- Automatic distribution fitting
- Simulate

Pre-process data

If the Pre-process data option is highlighted, when pressing the green play button located in the top left corner of the main window, the Data pre-processing panel window (Figure 2) will appear.



Data directory: >>> <<<

Beginning of pulse (s): ...

End of pulse (s): ...

Brief (second) pulse amplitude (nA): ...

AP blocker infusion time (s): ...

AP blocking time (s): ...

Minis blocker infusion time (s): ...

Starting file number: ...

Figure 2: Data pre-processing panel

The very top entry allows you to enter a path to the data directory containing the ABF files with whole cell patch clamp recordings. Alternatively, by clicking the button on the right-hand side of the entry, you can navigate to the data directory. The files in the directory should be named and ordered consecutively time-wise. They should include recordings prior to the application of any drugs followed by recordings after the application of action potential blockers, and, finally, by noise recordings following the application of minis blockers.

You should also indicate the beginning and end of the pulse (the second and the third entries, respectively) used for estimating membrane decay time constant, membrane capacitance, and the neuron's input resistance. At the beginning of each sweep there should be three pulses (Figure 3): a long depolarising one and two brief (0.5 ms) depolarising and hyperpolarising ones, respectively. If you are using the packaged minisMatlab app, you can use the plotabf function to visualise data saved in the ABF format.

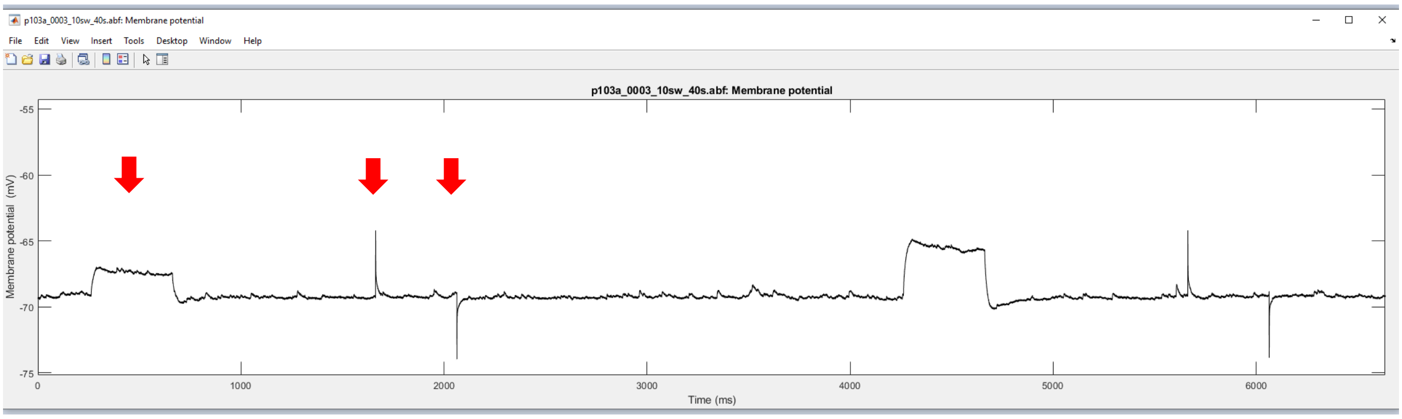
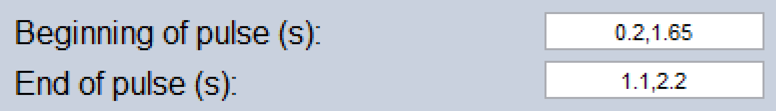


Figure 3: Membrane potential trace at the beginning of a typical whole-cell patch clamp recording

You should indicate the beginning of the first and the second pulses and the end of the first and the third pulses just like it is shown in Figure 4. The entered values should be separated by commas to indicate which of the pulses they correspond to. As you can see in the Figure 4, the entered values correspond to the recording example shown in Figure 3.



Beginning of pulse (s): 0.2,1.65

End of pulse (s): 1.1,2.2

Figure 4: Entering the beginning and end times for the pulses

If you wish, you can also supply the amplitude of the second/third pulse (the fourth entry). If you don't supply the amplitude, 'minis' will estimate this amplitude from the current data in the ABF file.

The first pulse is used for estimating the bridge balance error. The properties of this long pulse are estimated automatically.

You also have to enter the action potential blocker infusion time (the fifth entry), the time when you think action potentials are blocked (the sixth entry), and the minis blocker infusion time (the seventh entry).

Finally, you can enter the starting file number (the seventh entry). If you leave it empty, then the file starting number, by default, will be 1.

Once you have entered all the required information and are happy with it, press the green OK button to accept it and carry out the data pre-processing analyses. During the pre-processing analyses membrane potential traces corresponding to individual ABF files are produced and saved in a figures folder inside the data folder if a user chooses to do so. The workspace variables can also be saved in a mat file. The user is also asked if a text file describing recording properties in every ABF file should also be saved.

The text file contains two tables. The first table has the following headers: File (file parts: sweeps), Top 20% mean (amplitude mean of the top 20% amplitude events in mV detected by the 'minis' detection algorithm), Top 20% median (amplitude median of the top 20% amplitude events in mV), Top 10% mean (amplitude mean of the top 10% amplitude events in mV), Top 10% median (amplitude median of the top 10% amplitude events in mV), Top 5% mean (amplitude mean of the top 5% amplitude events in mV), Top 5% median (amplitude mean of the top 5% amplitude events in mV), Top 2% mean (amplitude mean of the top 2% amplitude events in mV), Top 2% median (amplitude median of the top 2% amplitude events in mV), Top 1% mean (amplitude mean of the top 1% amplitude events in mV), Top 1% median (amplitude median of the top 1% amplitude events in mV), 100ms SD (membrane potential mean standard deviation averaged over 100 ms duration windows in mV), 100ms smoothed SD (membrane potential mean standard deviation averaged over 100 ms duration windows in mV with membrane potential data being Gaussian-smoothed by the 'minis' detection algorithm), 15ms-mean SD (membrane potential mean standard deviation averaged over 15 ms duration windows in mV), 15ms-mean smoothed SD (membrane potential mean standard deviation averaged over 15 ms duration windows in mV with membrane potential data being Gaussian-smoothed by the 'minis' detection algorithm), 15ms-med SD (membrane potential median standard deviation averaged over 15 ms duration windows in mV), 15ms-med smoothed SD (membrane potential median standard deviation averaged over 15 ms duration windows in mV with membrane potential data being Gaussian-smoothed by the 'minis' detection algorithm), 100ms BL (membrane potential mean baseline averaged over 100 ms duration windows in mV), 100ms smoothed BL (membrane potential mean baseline averaged over 100 ms duration windows in mV with membrane potential data being Gaussian-smoothed by the 'minis' detection algorithm), 15ms BL (membrane potential mean baseline averaged over 15 ms duration windows in mV), 15ms smoothed BL (membrane potential mean baseline averaged over 100 ms duration windows in mV with membrane potential data being Gaussian-smoothed by the 'minis' detection algorithm). The second table has the following headers: File (file parts: halves), Top 10% tau_m (effective passive membrane time constant in ms estimated using top 10% amplitude events detected by the 'minis' detection algorithm), Impulse tau_m (passive membrane time constant in ms estimated by fitting a line to the slowest membrane potential response exponential decay part following brief current injections corresponding to the second and third pulses in Figure 3 and averaged across all such pulses in the recording), Effective tau_m (passive membrane time constant in ms estimated in a similar way to the impulse tau_m but instead of fitting the line a membrane potential drop of 1/e is used), Capacitance (in pF estimated based on impulse tau_m and (Major et al., 1993)), Effective capacitance (in pF estimated based on effective tau_m and (Major et al., 1993)), Pseudo series R (the bridge balance error in MΩ).

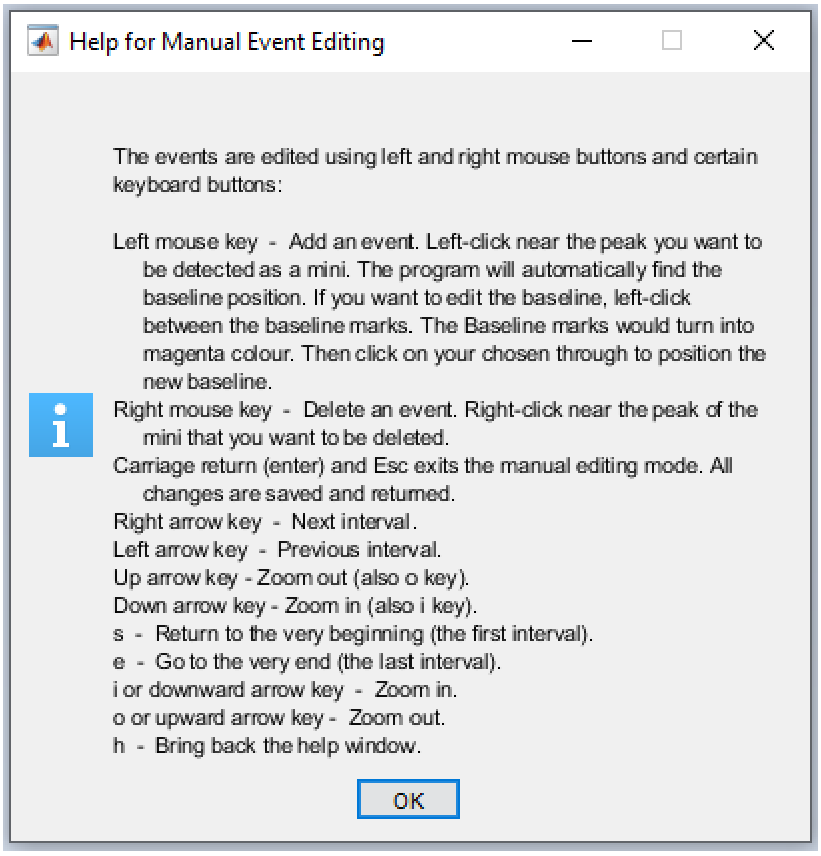
Besides the membrane potential trace figures and the summary text file, summary figures will also be produced. These figures can be saved manually by the user. The following figures will be produced: Amplitudes of the top 10% of events detected using the 'minis' detection algorithm, effective passive membrane time constant estimated using top 10% amplitude events detected using the 'minis' detection algorithm, membrane potential mean standard deviation averaged over 100 ms duration windows, membrane potential mean standard deviation averaged over 15 ms duration windows, membrane potential mean baseline averaged over 100 ms duration windows, impulse-based passive membrane time constant, total capacitance, and neuron's input resistance (the bridge balance error).

If you already ran the analyses in the past and now only want to display the summary figures, you can simply press the yellow Retrieve button. In this case you don't need to enter any of the details in the Data pre-processing panel. You will be prompted to provide a mat file containing the workspace variables that was saved at the end of the previous pre-processing analysis session.

Detect minis

In order to detect minis in your current/voltage clamp recording, highlight the Detect option in the Task panel of the main window (Figure 1). Also make sure that you specify the target file in the Files panel of the main window. You can either enter the full path to the file or you can navigate and select the file by pressing the button on the right-hand side of the target file entry. If you are happy with the default detection parameters shown in the Detection Parameters panel of the main window, you can press the green play button to start the minis detection process. Initially, you will be asked if you want to filter regular (systemic) noise. If you chose yes, the signal power spectrum will be displayed and you will be prompted to choose band-stop frequencies to be filtered out of the signal. Multiple band-stop frequencies can be separated by commas.

The results will be displayed immediately afterwards and you will be asked if you want to edit the detected events. If you choose to edit, you will see the help window (Figure 5) prior to starting the editing. The help window explains the meaning of button keys used during the editing process.



The events are edited using left and right mouse buttons and certain keyboard buttons:

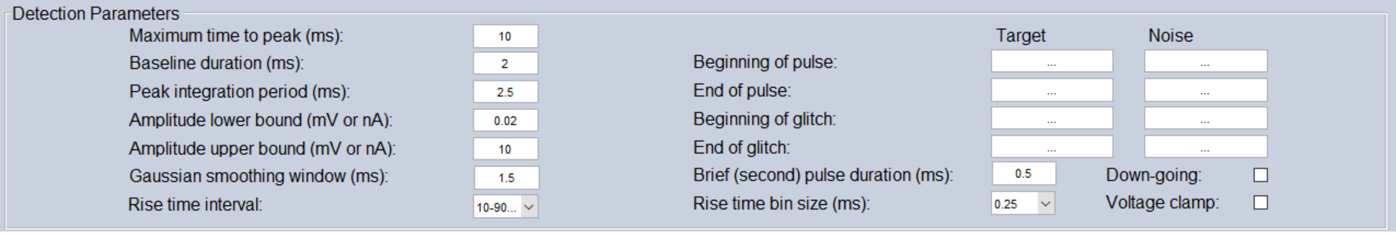
Left mouse key - Add an event. Left-click near the peak you want to be detected as a mini. The program will automatically find the baseline position. If you want to edit the baseline, left-click between the baseline marks. The Baseline marks would turn into magenta colour. Then click on your chosen through to position the new baseline.
Right mouse key - Delete an event. Right-click near the peak of the mini that you want to be deleted.
Carriage return (enter) and Esc exits the manual editing mode. All changes are saved and returned.
Right arrow key - Next interval.
Left arrow key - Previous interval.
Up arrow key - Zoom out (also o key).
Down arrow key - Zoom in (also i key).
s - Return to the very beginning (the first interval).
e - Go to the very end (the last interval).
i or downward arrow key - Zoom in.
o or upward arrow key - Zoom out.
h - Bring back the help window.

Figure 5: Help window for the manual editing of detected events

If you choose not to edit the detection results or after you have edited them already and would like to proceed, you will be asked if you want to estimate the average waveform of detected events. If you choose yes, the average waveform figure will be displayed. Subsequently, 1-dimensional and 2-dimensional amplitude and rise time histograms will be displayed and you will be asked if you would like to save an event log file and the initial figure showing the detected events. Finally, you will be asked if you want to save the remaining open figures like the event distribution graphs, the signal power spectrum, and the average waveform.

You can carry out the detection or any other type of operation using voltage clamp recordings instead and you can also look for inhibitory events. You would need to indicate this in Down-going and Voltage clamp boxes (Figure 6).

'minis' detection algorithm



Detection Parameters

Maximum time to peak (ms): 10
Baseline duration (ms): 2
Peak integration period (ms): 2.5
Amplitude lower bound (mV or nA): 0.02
Amplitude upper bound (mV or nA): 10
Gaussian smoothing window (ms): 1.5
Rise time interval: 10-90...

	Target	Noise
Beginning of pulse:	...	...
End of pulse:	...	...
Beginning of glitch:	...	...
End of glitch:	...	...

Brief (second) pulse duration (ms): 0.5
Rise time bin size (ms): 0.25
Down-going: □
Voltage clamp: □

Figure 6: Detection Parameters panel

New and original minis detection algorithm is one of the core features offered by 'minis' software. Here is a step by step outline of the algorithm:

1. During the initialisation of the detection process the user is prompted to enter stopband frequencies if one wishes to carry out the band-stop filtering of the membrane potential data in order to remove periodic noise. The procedure uses Butterworth filter with a stopband attenuation of 10 Db and passband ripple of 0.05 Db. The stopband size is 1 Hz and the passband being the entire frequency range except 6 Hz window surrounding the stopband frequency.
2. Subsequently the membrane potential data obtained during the electrophysiological recording is smoothed. This procedure is optional and fully controlled by the user. The smoothing parameter is set using the Gaussian smoothing window entry located in the Detection Parameters panel (Figure 6) of the main interface window (Figure 1). The Gaussian smoothing window corresponds to a standard deviation in milliseconds of a Gaussian function used for low-pass filtering the recorded data in order to remove the high frequency noise. Under this procedure each data point is convolved with a Gaussian function and in this way each data point is essentially a weighted average of itself and neighbouring data points.
3. The detection algorithm is based on peak detection in the recording data and a subsequent application of various selection criteria for accepting them as mini-like events (Figure 7). The first criterion of acceptance is the absence of higher peaks in the vicinity of the peak of interest. The duration of the vicinity period is a user-defined parameter that is set in the Peak integration period entry of the Detection Parameters panel (Figure 6) in milliseconds and it corresponds to the period before and after the peak.
4. The second selection criterion is whether the amplitude of the peak is within the range of acceptable amplitudes (Figure 7). This parameter is controlled by the user by entering the lower and upper amplitude bound values in the Amplitude lower bound and Amplitude upper bound boxes in the Detection Parameters panel (Figure 6) of the interface main window.
5. The period for finding a trough before the peak for positioning the baseline is set by the user in the Maximum time to peak box in the Detection Parameters panel (Figure 6) of the main interface window.
6. The baseline of the peak is positioned at the lowest point (trough) before the peak and its duration is as well controlled by the user (Figures 6 and 7). The end of the baseline extends 20% of the entire baseline duration ahead of the detected through with the rest of the 80% falling before the through. However, the duration of the baseline in certain situations is shorter than was set by the user. This happens when the baseline overlaps with an earlier peak and, therefore, is reduced so that it starts where the previous peak ends without affecting the end of the baseline.
7. However, the baseline, 10-90% rise time, and/or the peak may be recalculated in certain circumstances (Figure 7). First, the algorithm compares the intervals of 10-50% and 50-90% rise time. If the first interval is larger than the second one by a factor of more than 5, the algorithm reduces the period for finding a through before the peak so that it starts at the end of the original baseline. In this way a new baseline is found and the 10-90% rise time is re-calculated. The second circumstance occurs when the ratio of the first period divided by the second one is smaller or equal to 0.2. In this case the peak might be shifted to an earlier local maximum after the end of the baseline if such exists, and the 10-90% rise time would be re-calculated. The

third circumstance occurs when the end of the baseline deviates from the 10% rise time point by half of the baseline duration. If this happens, the algorithm reduces the period for finding a through before the peak so that it starts at half of the baseline duration before the 10% rise time point. Thus, the baseline is re-positioned and the 10-90% rise time is recalculated. However, if the amplitude of the new re-calculated minis-like event does not fall within the range of acceptable minis amplitudes, this event is rejected (see the step 4).

8. Finally, the user has a decision whether the detection algorithm uses 10-90% or 20-80% rise time by choosing between the two options in the Rise time interval box (Figure 6).

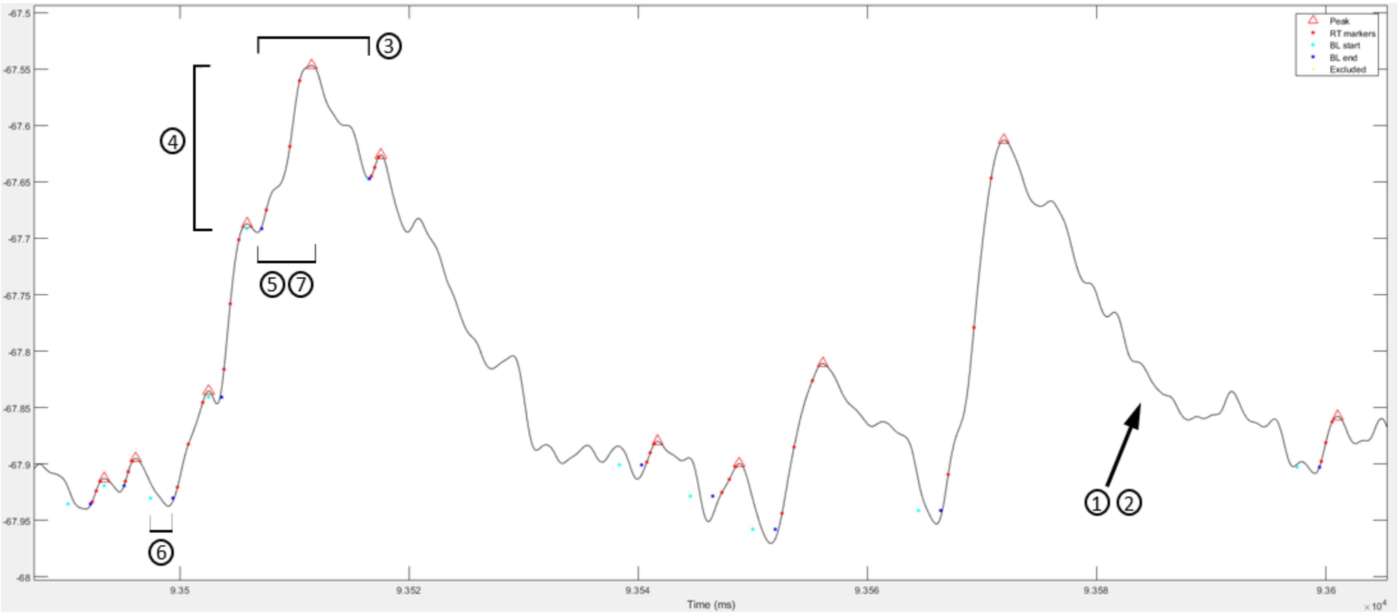


Figure 7: Steps of the 'minis' detection algorithm: (1) Band-stop filtering; (2) membrane potential trace Gaussian smoothing; (3) detection of well-defined peaks; (4) rejection of peaks with amplitude outside of acceptable range; (5) initial estimation of the rise time; (6) initial estimation of the baseline; (7) further correction of the baseline and the rise time if needed. Legend: red triangle – peak, red dot – rise time markers (10, 50, and 90% or 20, 50, 80% marks), cyan dot – baseline start, blue dot – baseline end, yellow dot – excluded times.

In addition to the above detection parameters, there are extra entries in the Detection Parameters panel (Figure 6) that affect data processing. User should set times when current is injected, so that these time periods could be excluded from any data analyses (Beginning of pulse and End of pulse entries). These times are relative to the beginning of each sweep. Multiple pulse times should be separated by commas. Moreover, the user can specify glitch times that should also be excluded (Beginning of glitch and End of glitch entries). The times are relative to the beginning of the first sweep. Multiple glitch times should be separated by commas. The panel also has the Brief (second) pulse duration entry in ms (see the Pre-process Data section for more information about this entry). Finally, Rise time bin size in ms could also be specified. This parameter is used any time the software is estimating the minis rise time distribution.

Detect and compare minis

The third mode of software operation allows the user to compare two recordings. Minis detection is carried out on two recordings that are specified in the Files panel of the main window (Figure 1). The same detection parameters are used for the two recordings and are specified in the Detection Parameters panel (Figure 6). The pulse and glitch times are individually specified for each file. Once the parameters are set, the Detect and Compare option should be selected in the Task panel and the green play button should be pressed.

For the most part, Detect and Compare option is executed in identical manner to the Detect mode with the minis detection procedure carried out on the two files. There are a few exceptions, however. First of all, the average waveform is only produced for the target recording. Second, amplitude and rise time distributions and power spectra for the target and noise files are shown simultaneously to facilitate comparison of the two recordings.

Estimating error bounds for distribution fitting

The second core functionality of the ‘minis’ software is the estimation of the properties of the minis distribution in terms of amplitudes and rise times using the Matlab’s genetic algorithm (the ‘minis’ optimisation algorithm). Two types of recordings are required for this purpose. The first type of recording is the membrane potential trace after the application of action potential and inhibitory synaptic activity blockers (or excitatory synaptic activity blockers depending on which type of activity interests you). It should contain minis on top of the background noise and is referred to here as the target recording. The second type of recording should contain only the background noise and is obtained after all types of synaptic transmission are blocked. It is referred to here as the noise recording. The ‘minis’ optimisation algorithm simulates minis based on the passive cable theory and adds these simulated events on top of the noise recording. The goal is to match the target recording closely enough in terms of its amplitude and rise times distributions derived after subjecting the recording to the ‘minis’ detection algorithm. The meaning of ‘closely enough’ is defined by the error bound estimation procedure that the software carries out when the user selects Estimate error bounds option in the Task panel of the main window (Figure 1).

The software estimates two types of error bounds and how they change depending on the size of target files. The first type of error bounds is based on the variability in the sum of absolute deviations (SAD) between the amplitude and rise time distributions of target files. The second type of error bounds is based on the variability in the maximum absolute deviation (MAD) between the amplitude and rise time distributions of target files.

The theoretical model that describes the increase in the SAD and MAD scores as the size of the comparison files is increased is based on the notion of a random variable and the relation between the standard deviation (SD) of a random variable and the average absolute deviation of a random variable. The SAD and MAD scores essentially are random variables that have a normal probability density distribution. This is true for files of a single sweep length and, as the sum of identically normally distributed variables is also normally distributed, so this would also hold for files longer than a single sweep length. Furthermore, according to the probability theory, the variance of a sum of two random variables is equal to the sum of their variances. That is,

$$\sigma_{(X+Y)}^2 = \sigma_X^2 + \sigma_Y^2,$$

where σ^2 is a variance of a random variable and X and Y are random variables. The standard deviation is then given by

$$\sigma_{(X+Y)} = \sqrt{\sigma_X^2 + \sigma_Y^2}.$$

The standard deviation of the sum of two identical variables is given by

$$\sigma_{(X+Y)} = \sqrt{2}\sigma_{X,Y}$$

and for more than two variables it is given by

$$\sigma_{(X+Y+\cdots+N)} = \sqrt{n}\sigma_{X,Y,\ldots,N},$$

where n is the number of random variables and N is the subscript of the nth variable. Now, it is known that the standard deviation of a normal distribution is related to its mean absolute deviation by a ratio SD/mean absolute deviation = $\sqrt{2/\pi}$. The mean absolute deviation is an expected value of an absolute value of a random variable and is given by

$$E(|x|) = \int_{-\infty}^{\infty} |x|\Phi(x)dx,$$

where x is a random variable and $\Phi(x)$ is a normal probability density. Since both absolute value and normal density functions are even,

$$E(|x|) = 2\int_{0}^{\infty} |x|\Phi(x)dx.$$

Integration by substitution gives

$$E(|x|) = -\sigma\sqrt{\frac{2}{\pi}}[e^{-\frac{x^2}{2\sigma^2}}]_0^{\infty} = \sigma\sqrt{\frac{2}{\pi}}.$$

Hence, we show how this relation is derived. Moreover, this relation extends to SAD because mean absolute deviation is equal to SAD/number of histogram bins. Thus,

$$SAD = number\ of\ histogram\ bins \times \sigma\sqrt{\frac{2}{\pi}}.$$

The equation above holds in general. In the case where we have the sum of random variables, we write

$$SAD_n = number\ of\ histogram\ bins \times \sqrt{n}\sigma_{X,Y,\dots,N}\sqrt{\frac{2}{\pi}}$$

$$SAD_n = \sqrt{n}SAD_{X,Y,\dots,N}.$$

This is the model that describes how SAD changes with increasing the length of comparison files. The length of the file in this case is exactly the same as the number of random variables n. Finally, the same model applies to MAD because the MAD score is directly proportional to the SAD score and on average the following should hold with increased variability

$$MAD_n = \sqrt{n}MAD_{X,Y,\dots,N}.$$

However, this model in the above form applies only to comparison files of equal length. In situations where the two files have different length, the SAD and MAD score of one of the files would be scaled up or down in order to compensate for the length discrepancy. The average of SAD of the two files is then given by

$$SAD_{L|S} = (\frac{l}{s}\sqrt{s}SAD_{X,Y,\dots,N} + \sqrt{l}SAD_{X,Y,\dots,N})/2$$

$$SAD_{L|S} = \left(\frac{l}{\sqrt{s}}SAD_{X,Y,\dots,N} + \sqrt{l}SAD_{X,Y,\dots,N}\right)/2,$$

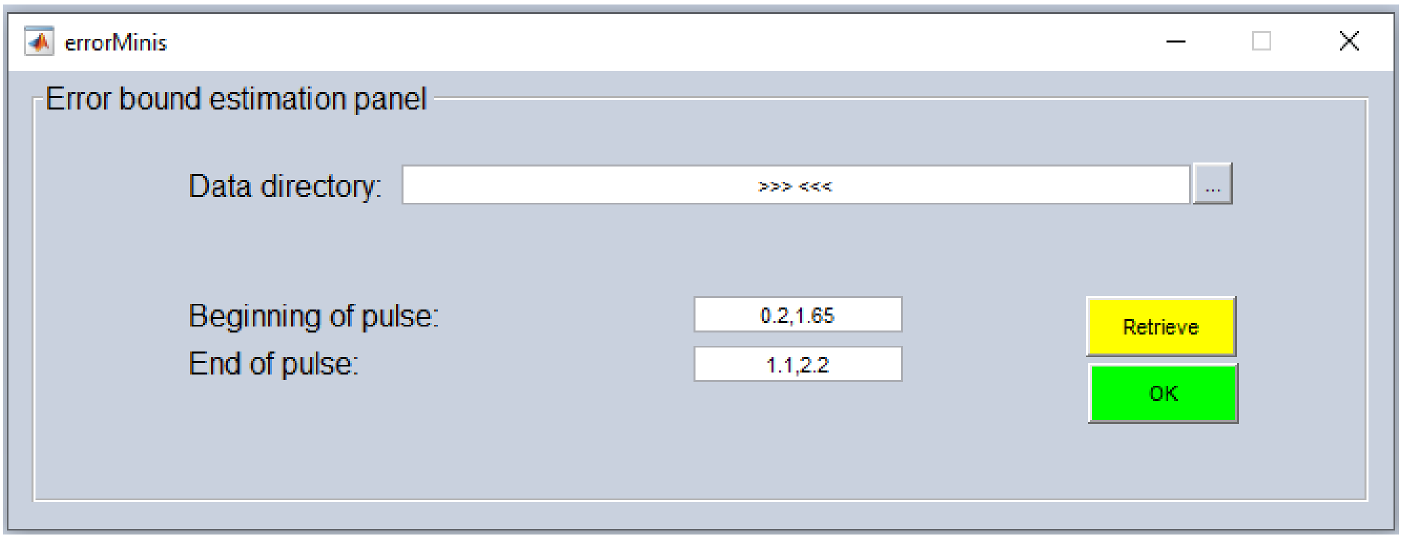
where l is the length of a target file and s is the length of a file scaled to match the target file. Extending this relation to MAD scores finishes the description of how the software estimates and models cross-file differences in the SAD and MAD:

$$MAD_{L|S} = \left(\frac{l}{\sqrt{s}}MAD_{X,Y,\dots,N} + \sqrt{l}MAD_{X,Y,\dots,N}\right)/2.$$

Ideally, there should be a non-stationarity correction to the model in the error bound estimation procedure. The correction should be achieved by replacing the $\sqrt{n}$ term of the model expression with $n^{estimated}$, where the estimated power is in the range [0.5, 1] giving the best fit to the data in terms of the least square. It has to be noted that this correction does not have a mathematical proof yet and is used here as a heuristic. The correct way of accounting for non-stationarity in the data is to use the information contained in the overall standard deviation of the recording trace. This information provides a way for explicit mathematical description of the correction procedure. It is also more powerful than the heuristic above because it allows estimating the non-stationarity associated with each different file length rather than a single estimate for the entire model. The correction procedure may be implemented in the future in order to improve the optimisation routine.

When estimating the error bounds, the 'minis' software calculates SAD and MAD score averages for the target file size of interest. It also calculates the error scaling factor for cases when noise and target recording files are of different length. Moreover, rather than taking the mean or the median differences between detected event amplitude and rise time distributions, the software calculates the 6-score 50th centile based on six difference distributions. These are all amplitudes distributions, top 50% amplitudes distribution, top 10% amplitudes distribution, top 2% amplitudes distribution, rise time distribution, and two-dimensional amplitude and rise time distribution. The SAD and MAD scores based on 6-score 50th centile are the minimal SAD and MAD scores that satisfy 50% difference in all six difference distributions simultaneously.

Once the error bound estimation is initiated, the software opens the Error bound estimation panel (Figure 8). The user has to supply a path to the folder containing the target files in the Data directory entry. Beginning and End of the pulse entries should also be specified. Multiple pulses should be separated by commas. Once this is done, pressing OK button initiates minis detection procedure in all of the target files. When minis detection procedure is finished, minis amplitude and rise time histograms for all target files are displayed. The user is then prompted to save the intermediate results that could be loaded by the software in the future in case one needs to resume the error bound estimation procedure. The intermediate results can be loaded after pressing the yellow Retrieve button and the error bound estimation procedure would resume from this point onwards.



errorMinis

Error bound estimation panel

Data directory: >>> <<<

Beginning of pulse: 0.2,1.65

End of pulse: 1.1,2.2

Figure 8: Error bound estimation panel

During the second stage of error bound estimation, EPSP and IPSP mean, median and 6-score 50th centile SAD and MAD graphs are displayed. The user is then asked if the SAD and MAD model fitting graphs should also be displayed. Subsequently, the user is also asked whether to estimate error bounds for files of particular size. The estimated error bounds can also be saved in a text file. At this stage the user is also asked if the SAD and MAD model fitting graphs should be saved. If the user chooses to save them, the figures are saved in a designated folder error_bounds. The user is then prompted for closing the figures or keeping them open for inspection. Finally, the user is prompted for saving the error bound estimation results to be used later by the 'minis' optimisation algorithm.

The text file that is saved is rather large and contains key and a lot of auxiliary information on error bounds for the optimisation part. Minis detection parameters that were used when detecting minis-like events in the target files are saved at the top of the file. This is followed by SAD and MAD 6-score 50th centiles for all detected depolarising and hyperpolarising events in terms of amplitude, rise time, and two-dimensional difference distributions, then by corresponding SAD and MAD medians and means, and by maximal and minimal SAD and MAD values. Lines of stars are used to separate sections of different types of data. There are SAD and MAD scores for events having the top 50%, 10%, and 2% of amplitudes.

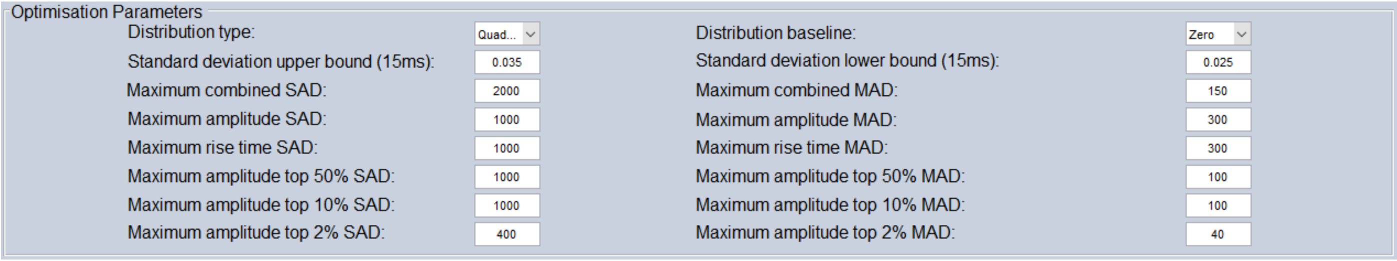
The empirical data is followed by modelling results. First there are sections showing the actual fitted values of the model various amplitude configurations. The associated least square errors are also printed out with the corresponding line number on the left. Confidence-weighted fits weigh the least square error by the file size in sweeps. The model equations are also provided.

The end of the file contains recording sweep and file information. Recording sweep duration factors are printed out for each of the sweep. The shorter the sweep, the larger the duration factor. This is followed by file names, file lengths in sweep numbers, duration factors for each of the files, as well as standard deviations based on various window sizes. Finally, SAD and MAD score corrections for different size noise and target files are provided based on the corrections requested by the user (if requested at all). Both corrections based on empirical comparisons, as well as on modelling predictions are provided.

Automated distribution fitting

The full functionality and potential of the 'minis' software is revealed when the user choses to execute the software in the Automated distribution fitting mode (Figure 1). In this mode two of the core features of the software are employed: 'minis' detection algorithm and 'minis' optimisation algorithm. It takes a long time and requires a lot of computing resources in order to run the software in this mode. Automated distribution fitting estimates the real minis distribution in terms of their amplitude and rise time in your current clamp recording of cell membrane potential.

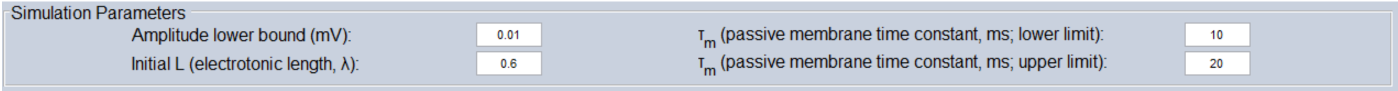
Two prerequisites are essential for estimating the minis distribution using 'minis' optimisation algorithm. First of all, the user has to have the cell's membrane potential recorded in a specific way where recording of minis and noise is present in one recording and only noise is present in the second recording. What one has to do is to initially block all the action potentials and inhibitory (excitatory) synaptic transmission in the recording brain slice for some time and record the membrane potential. The second recording is obtained after the excitatory synaptic transmission is blocked and is expected to contain only the membrane potential noise. Typically a single continuous recording is obtained which is later divided into segments of target (noise + minis) and noise data. The second prerequisite is that the user is expected to do all the required data pre-processing analysis in order to estimate error bounds for distribution fitting (see the previous section). Once these requirements are met and the user has filled in the entries in the Optimisation Parameters (Figure 9) and Simulation Parameters (Figure 10) panels of the main window, the 'minis' optimisation algorithm is ready to be executed.



Optimisation Parameters

Distribution type:	Quad...	Distribution baseline:	Zero
Standard deviation upper bound (15ms):	0.035	Standard deviation lower bound (15ms):	0.025
Maximum combined SAD:	2000	Maximum combined MAD:	150
Maximum amplitude SAD:	1000	Maximum amplitude MAD:	300
Maximum rise time SAD:	1000	Maximum rise time MAD:	300
Maximum amplitude top 50% SAD:	1000	Maximum amplitude top 50% MAD:	100
Maximum amplitude top 10% SAD:	1000	Maximum amplitude top 10% MAD:	100
Maximum amplitude top 2% SAD:	400	Maximum amplitude top 2% MAD:	40

Figure 9: Optimisation Parameters panel



Simulation Parameters

Amplitude lower bound (mV):	0.01	τ_m (passive membrane time constant, ms; lower limit):	10
Initial L (electrotonic length, λ):	0.6	τ_m (passive membrane time constant, ms; upper limit):	20

Figure 10: Simulation Parameters panel

The 'minis' optimisation algorithm works by taking an analytically defined distribution of minis in terms of amplitudes and rise times and then randomly drawing a limited number of membrane potential events from this distribution to be simulated and added to the noise membrane potential recording. The shape of the analytical distribution is controlled by the Matlab's genetic algorithm which repeatedly generates a sample of distributions that are evolved over a number of generations in order to match the distribution of the detected events in the noise + simulated minis membrane potential recording to the distribution of the detected events in the noise + real minis distribution in terms of amplitudes and rise times after the application of the 'minis' detection algorithm. The key aspects of the algorithm are the parameter space and fitness function of the optimisation algorithm. Each of these are discussed in turn below.

The user has an option to choose the type of minis distribution to be fitted in the Distribution type box of the Optimisation Parameters panel (Figure 9). Available distribution types are listed in Figure 11. The default is the quadrimodal normal distribution. The difference between Normal and Gaussian is the way the two types of distributions are generated. The Normal distribution uses Matlab's pseudorandom normal number generator function randn, whereas the Gaussian distribution is simply a rounded analytical distribution. The Distribution baseline box allows the user to add to the simulated distribution all the positive events after subtracting the two-dimensional amplitude and rise time noise distribution from the target distribution (Subtracted option). The default is Zero which does not add anything.

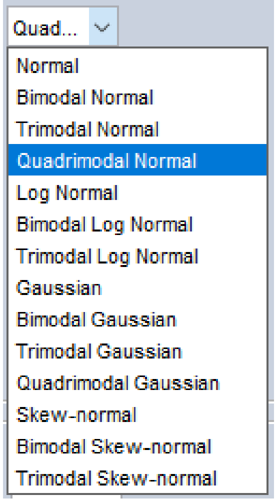
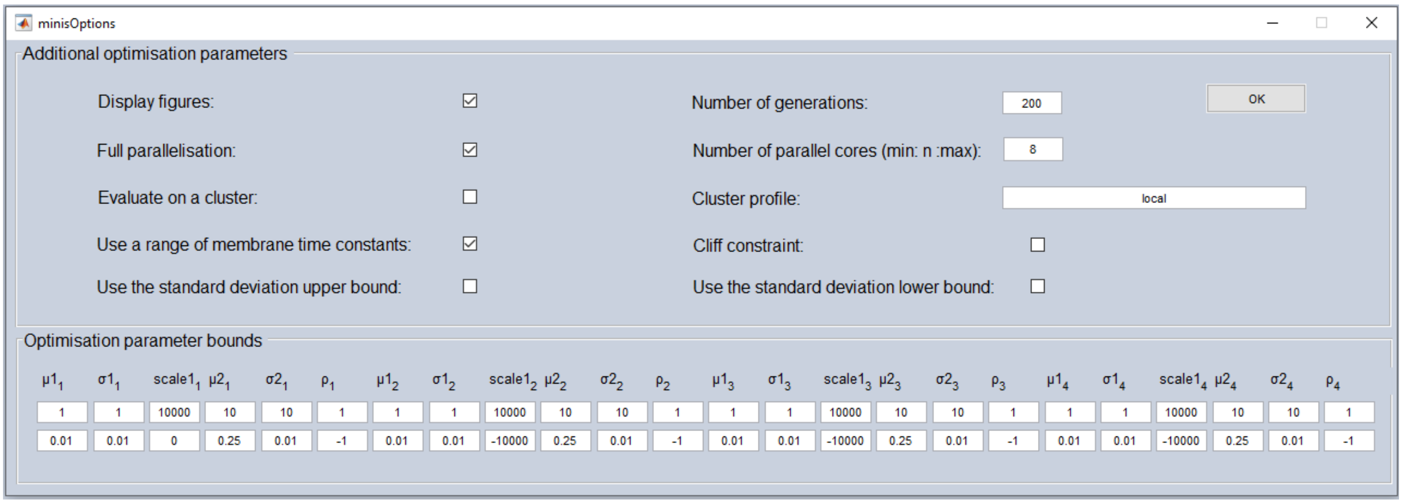


Figure 11: Available simulated minis distribution for minis distribution fitting

Standard deviation upper and lower bounds based on 15 ms window average can also be specified. The upper bound value should be based on the mean estimate of this value in target files. These values are available in the error bounds text files. The lower bound value should be based on the mean estimate of this value in noise files. An alternative approach is to leave these entries empty and simply allow the ‘minis’ software to estimate these values based on the target and noise files you use for optimisation. These files are indicated in the Files panel of the main window (Figure 1). The latter approach has been taken in the algorithm development study.

The remaining entries of the Optimisation Parameters panel should be entered based on the error bound text file. They all are various SAD and MAD scores and can either be based on fitted estimates to the empirical data or be direct estimates based on the empirical data itself. The latter approach has been taken in the algorithm development study. All of these entries should be based on the 6-score 50th centile SAD values.

The user can bring up the optimisation options window (Figure 12) by going to the Optimisation options menu at the top menu of the main window and selecting Change options. The window has two panels: Additional optimisation parameters and Optimisation parameter bounds. The Additional optimisation parameters panel allows the user to choose whether optimisation figures should be displayed (Display figures box), whether to use full parallelisation (Full parallelisation box) or instead limit the parallelisation to a specific number of cores (Number of parallel cores box), whether to evaluate on a computing cluster (Evaluate on a cluster box; currently not implemented), as well as to specify the name of the cluster profile (Cluster profile box). When entering the number of parallel cores, instead of giving a number, the user can enter min or max which would give the minimum or maximum available cores on the profile, respectively. Moreover, the number of generations of the Matlab genetic algorithm (Number of generations entry) can also be specified by the user. The number of generations should not be smaller than 2.



Additional optimisation parameters

Display figures: ☑
Full parallelisation: ☑
Evaluate on a cluster: ☐
Use a range of membrane time constants: ☑
Use the standard deviation upper bound: ☐
Number of generations: 200
Number of parallel cores (min: n :max): 8
Cluster profile: local
Cliff constraint: ☐
Use the standard deviation lower bound: ☐

Optimisation parameter bounds

$\mu1_1$	$\sigma1_1$	$scale1_1$	$\mu2_1$	$\sigma2_1$	ρ_1	$\mu1_2$	$\sigma1_2$	$scale1_2$	$\mu2_2$	$\sigma2_2$	ρ_2	$\mu1_3$	$\sigma1_3$	$scale1_3$	$\mu2_3$	$\sigma2_3$	ρ_3	$\mu1_4$	$\sigma1_4$	$scale1_4$	$\mu2_4$	$\sigma2_4$	ρ_4
1	1	10000	10	10	1	1	1	10000	10	10	1	1	1	10000	10	10	1	1	1	10000	10	10	1
0.01	0.01	0	0.25	0.01	-1	0.01	0.01	-10000	0.25	0.01	-1	0.01	0.01	-10000	0.25	0.01	-1	0.01	0.01	-10000	0.25	0.01	-1

Figure 12: Optimisation options window

The user can also specify to use a range of membrane time constants (Use a range of membrane time constants box). If this is the case, the genetic algorithm adds the passive membrane time constant to the parameters that it controls. The lower limit on the passive membrane time constant is 0.95 times the τ_m (passive membrane time constant, ms; lower limit) entry of the Simulation Parameters panel (Figure 10), whereas the upper limit is set in the corresponding upper limit entry. Alternatively, the user can choose to base the lower limit on the 0.95 times the τ_m estimate derived from current pulses and the upper limit on the effective τ_m of detected minis-like events in the target file. If lower and upper limit entries on τ_m are left empty, the 'minis' optimisation algorithm would estimate τ_m based on current pulses and simply take the range to be between 0.95 times and 1.05 times the τ_m estimate. If the range of membrane option box in the Additional optimisation parameters panel is unticked, the 'minis' optimisation algorithm estimates τ_m based on current pulses and leaves this value constant.

The Cliff constraint option of the Additional optimisation parameters panel allows the user to penalise any generated minis distributions that have a sharp edge. Sharp edges located next to any distribution bins between the top 50th to 100th percentile bin values get penalised depending of the height of the bin. Two-dimensional and one-dimensional amplitude histograms are both considered. The distribution is penalised only once with the priority given to the one-dimensional amplitude distribution. That is, if the one-dimensional amplitude distribution does not have a sharp edge, only then the two-dimensional distribution is considered. Finally, the user also has the option to use the standard deviation upper and lower bounds (Use the standard deviation upper and lower bound boxes) which are set in the Optimisation parameters panel (Figure 9) of the main window.

Optimisation parameter bounds panel contains all of the distribution bounds controlled by the Matlab genetic algorithm. They vary depending on the chosen simulated minis distribution (Figure 11). In the example shown in Figure 12 the bounds correspond to the quadrimodal normal distribution and they limit the variation range of 26 distribution parameters describing amplitude and rise time means (μ1 and μ2, respectively), amplitude and rise time standard deviations (σ1 and σ2, respectively), and scaling factors (scale1) of distribution modes indicated by subscript numbers. The number and type of these distribution shape parameters vary depending on the number of modes and distribution type. In addition to these parameters, the 'minis' optimisation algorithm also sets bounds on the synaptic rise constant τ_{sy1} (0.15-3 ms) without an option for the user. As discussed earlier, it may also set the bounds on the passive membrane time constant τ_m if the user chooses so. Once the user is happy with the choice of optimisation parameters, pressing OK button updates the internal program variables and the Optimisation options window is closed.

The Amplitude lower bound entry (in millivolts) of the Simulation Parameters panel sets the lower amplitude bound on simulated minis. Smaller amplitude minis are not generated by the minis distribution fitting algorithm. The Initial L (electrotonic length, l/λ) entry sets the smallest dendritic length used in minis simulations by the fitting algorithm. This parameter is the physical length of the dendrite normalised by the space constant of a dendritic segment and, therefore, is dimensionless. This essentially finishes the discussion of the parameter space of the optimisation algorithm.

The user initiates the distribution fitting process by selecting the Automatic distribution fitting option in the Task panel of the main window (Figure 1). Once started, the software then asks if the passive membrane time constant should be estimated based on current pulses. If the user chooses so, the upper and lower limit values entered in the Simulation Parameters panel (Figure 10) are ignored and the range is taken to be between 0.95 times and 1.05 times the τ_m estimate. The user is also given an option to band-stop filter the certain noise frequencies, supply data for optimisation, and resume an earlier optimisation. Once the genetic algorithm starts, a number of minis distribution candidates is generated. The exact number of candidates is either ten times larger than the number of optimised parameters or 40, whichever number is larger. Each candidate is then taken in turn and individual minis are drawn from the distribution and simulated. The current is taken to be a double exponential with a voltage response given by the lumped terms solution in Appendix 2 of (Major et al., 1993) with 100 terms ($n = 100$). The simulated minis are overlaid on the noise recording and the 'minis' detection algorithm is applied to the resultant voltage trace. The distribution of detected events is then compared to the target distribution.

The quality of the match is determined by a complex fitness function that has 31 parameters in total. The first 15 are range parameters. The range of target distribution values for each bin is established during the error bound estimation. When the user is starting the distribution fitting, the software asks to supply optimisation data which has been saved during the error estimation procedure. At the initial stage the optimisation algorithm aims to obtain a minis distribution that fits within the range. The 15 range parameters are 6-score 50th centile values of the 2-D SAD, full amplitude SAD, rise time SAD, 2-D MAD, full amplitudes MAD, rise time MAD, top 50% amplitudes SAD, top 50% amplitudes positive SAD, top 50% amplitudes MAD, top 10% amplitudes SAD, top 10% amplitudes positive SAD, top 10% amplitudes MAD, top 2% amplitudes SAD, top 2% amplitudes positive SAD, and top 2% amplitudes MAD. The positive SAD scores for amplitudes are based only on positive bin values after subtracting the noise + simulated minis distribution from the target distribution. It is specifically aimed at obtaining a better fit of the top amplitude distribution as it was observed that not having this fitness value often results in final fitted distributions that have a considerably smaller count of top amplitude events than the target distribution. The optimisation algorithm fits these parameters one by one proceeding in the same order as they are listed here. Once the range parameters are fitted, the optimisation algorithm proceeds to fit the specific target distribution rather than a range of them. The same 6-score 50th centile SAD and MAD scores are used but with the aim of minimising the scores pertaining to the specific comparison of interest. The last parameter to fit is the standard deviation of the voltage signal that is optional. Its range is set by the user in the Standard deviation upper and lower bounds entries of the Optimisation Parameters panel (Figure 9). It can be removed by unticking the Use the standard deviation upper and lower bound boxes of the Optimisation Options window (Figure 12).

The fitness score is calculated first by creating a cost basis vector. The cost basis vector is a 31-element vector starting with 60 times the Maximum combined SAD entry value of the Optimisation Parameters panel and finishing with 0 in steps of 2 times the Maximum combined SAD entry value. Then the 31 fitness parameter scores are calculated and their deviations from target values are taken and normalised by the corresponding target values and added to the cost basis vector. The first parameter with the deviation of more than zero then serves as the basis for fitness calculation. The fitness of the simulated minis distribution is, thus, the corresponding entry of the basis vector plus the normalised deviation of the corresponding fitness parameter.

There is one caveat to the fitness calculation. Once the initial fitness score for the simulated minis distribution is calculated, the optimisation algorithm may then attempt to improve this fitness score. A new minis distribution would then be created on the basis of the existing one by adding the top 10% amplitudes serving as the basis for the top 10% positive SAD. The simulated minis drawn from this amplitude tail distribution are added on top of the existing noise + simulated minis voltage trace and the fitness of the new distribution is then evaluated. If its score is better than the score of the initial distribution, the new distribution is kept instead. This step can only occur once the 15 range parameters are fitted.

Once the fitness values of all the candidate distributions within a single generation are calculated, a number of the best performing candidates is chosen, crossbred by the Matlab's genetic algorithm, and the next generation of candidates is created. The optimisation algorithm stops once all 31 (or 30) fitness parameters are satisfied or once the maximum number of generations is reached. The latter is specified in the Number of generations entry of the Optimisation options window (Figure 12).

While optimisation algorithm is running, its performance is displayed in a graph showing the fitness value distribution and other metrics of minis candidate distributions. This graph also allows the user to pause or stop the minis distribution fitting. The optimisation algorithm is paused or stopped once it finishes evaluating every candidate distribution in the current generation. The state of the algorithm is saved in the initVars.mat file in the created data folder. This file can be loaded into the algorithm in the future and the optimisation can be resumed. If the algorithm is running in a serial fashion, intermediate figures are produced. These include 1-D amplitude and rise time fits for the best candidate, as well as a 2-D amplitude and rise time fit. The best simulated minis distribution is also displayed, as well as distribution power spectra and the full range of target 1-D amplitude and rise time distributions.

Finally, figures showing the performance of the best candidate on every fitness parameter are displayed. The group comparison optimisation summary figure shows the performance on the 15 range parameters. The performance on the remaining specific comparison parameters is displayed in the Single comparison optimisation summary. The circle indicates the fitness score for a particular comparison of interest while the downward-pointing triangle indicates the best possible comparison. The target and range values are indicated in every figure panel with range values calculated by running all possible between target file comparisons. If the user is running the optimisation algorithm in a parallel mode, the intermediate figures are not displayed and only the final figures are displayed that show the performance of the final best candidate distribution.

Simulating minis

The final task that you can run is Simulate (Figure 1). In this mode minis are simulated and overlaid at pseudo-random locations over the noise recording. The detection algorithm is then applied to simulated trace and the detection performance is evaluated according to the signal detection theory as described in (Dervinis and Major, 2022). The simulated minis are drawn from the distribution indicated in the Distribution type box (Figure 11). The distribution parameters are taken from the Optimisation options menu (Figure 12): Optimisation parameter bound values are averaged resulting in a single value per parameter. The simulated smoothed and unsmoothed traces are saved in ABF type files in addition to the mat file containing detection performance variables. The following detection performance variables are saved:
'sensitivity' - true positive (hit) rate.
'specificity' - specificity or 1 - false positive (false alarm) rate.
'FPR' - false positive rate.
'dPrime' - sensitivity index.
'performance' - a row matrix containing logical indices corresponding to

row 1: simulated event positions (peaks);
row 2: hits (detected positions) + misses;
row 3: hits (detected positions) + false alarms;
row 4: hits (detected positions);
row 5: misses;
row 6: false alarms;
row 7: correct rejections.

'falseI' - locations (indices) of prominent noise events.
'falseT' - times of prominent noise events
'detectionParameters' - a structure variable containing parameters controlling the detection task.
'simulationParameters' - a structure variable containing parameters controlling simulation of minis.
'optimisationParameters' - a structure variable containing parameters controlling the automated distribution fitting.
'classificationParameters' - a structure variable containing parameters controlling distribution binning.
'simulatedEventInfo' - a matrix with rows corresponding to simulated events and columns corresponding to (1) minis count, (2) amplitude, (3) rise time, (4) electrotonic charge input site distance from the measuring site, (5) electrotonic length of the dendritic cylinder, (6) onset index, (7) onset time, (8) peak index, (9) peak time.

Controlling 'minis' via the application programming interface

Graphical interface is only one way how 'minis' can be controlled. Programming interface is another way to control 'minis' that offers more flexibility. Programming interfaces exist for both Matlab and Python. Examples of how to execute 'minis' in Matlab and Python can be found in testMatlab.m, testMatlab_preload.m, testPython.py, and testPython_preload.py files.

References

Dervinis M, Major G (2022) Novel method for reliably measuring spontaneous postsynaptic potentials/currents in whole- cell patch clamp recordings in the central nervous system. bioRxiv:2022.03.20.485046 Available at: http://biorxiv.org/content/early/2022/03/21/2022.03.20.485046.abstract.

Major G, Evans JD, Jack JJ (1993) Solutions for transients in arbitrarily branching cables: I. Voltage recording with a somatic shunt. Biophys J 65:423–449.