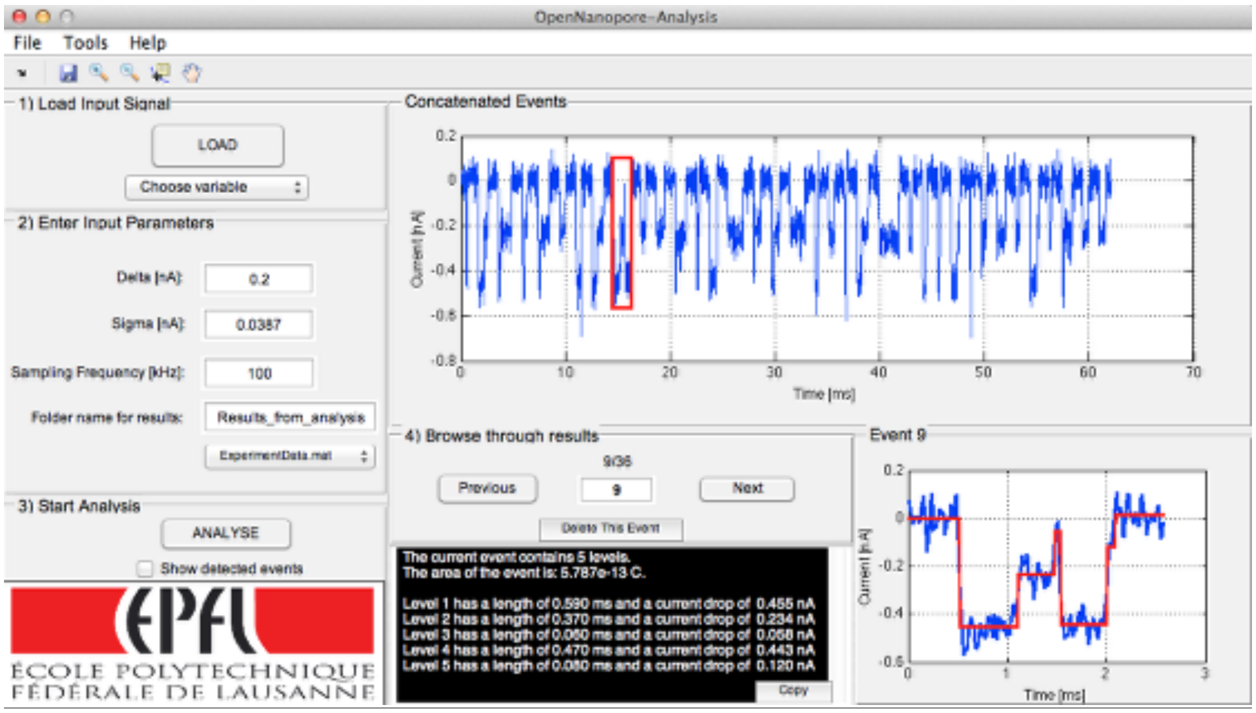
OpenNanopore

Table of Contents

User Guide

How to use the analysis GUI

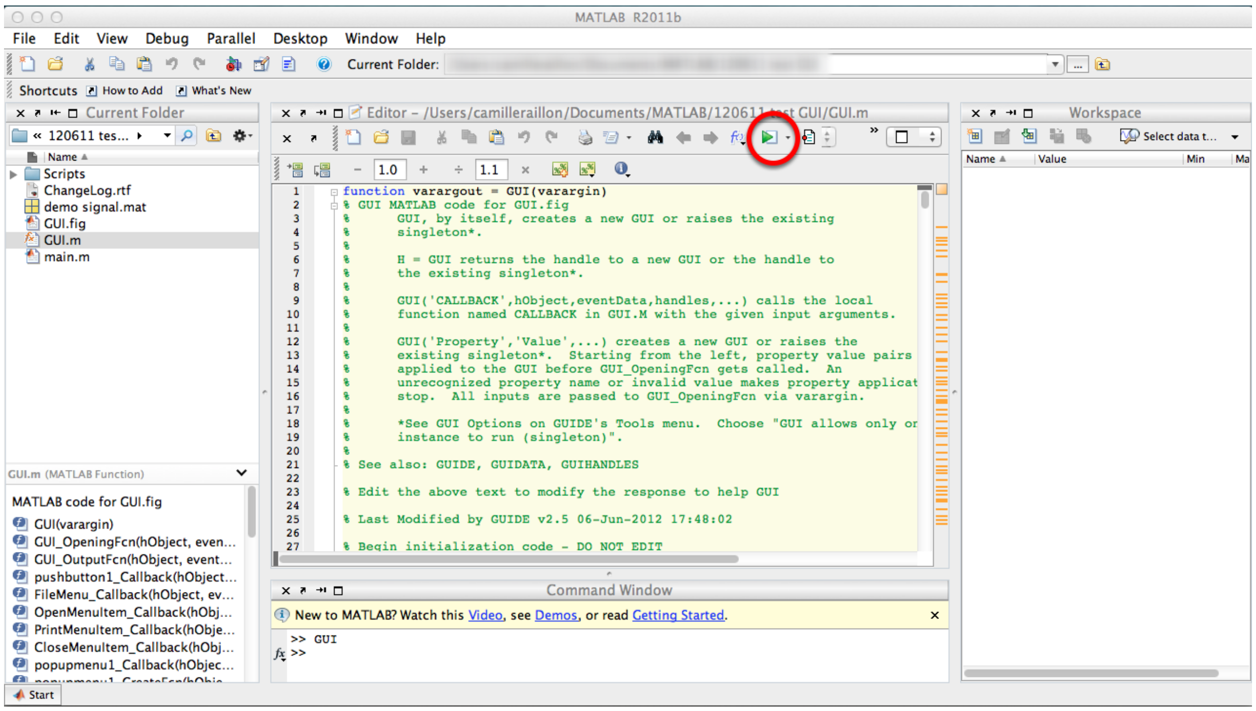
In this section you will learn how to use the graphical user interface. Please visit the youtube tutorial for more informations: https://www.youtube.com/watch?v=McKy_hNy7cg&feature=plcp



Delta [nA]: 0.2
Sigma [nA]: 0.0387
Sampling Frequency [kHz]: 100
Folder name for results: Results_from_analysis
ExperimentData.mat

```
The current event contains 5 levels.
The area of the event is: 5.787e-13 C.

Level 1 has a length of 0.590 ms and a current drop of  0.455 nA
Level 2 has a length of 0.370 ms and a current drop of  0.234 nA
Level 3 has a length of 0.060 ms and a current drop of  0.058 nA
Level 4 has a length of 0.470 ms and a current drop of  0.443 nA
Level 5 has a length of 0.080 ms and a current drop of  0.120 nA
```

0) Open GUI.m and click on "run"

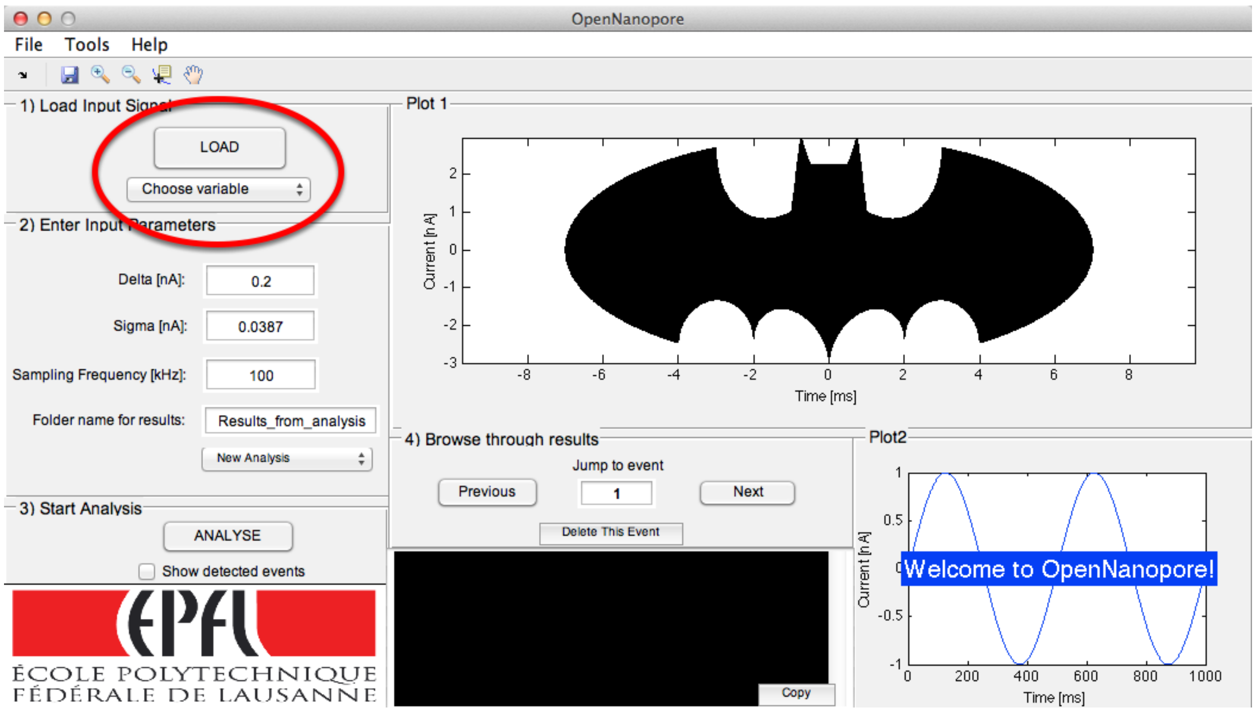

MATLAB R2011b
File Edit View Debug Parallel Desktop Window Help
Current Folder:
Shortcuts
How to Add
What's New
Current Folder
120611 tes...
Name
Scripts
ChangeLog.rtf
demo signal.mat
GUI.fig
GUI.m
main.m
GUI.m (MATLAB Function)
MATLAB code for GUI.fig
GUI(varargin)
GUI_OpeningFcn(hObject, even...
GUI_OutputFcn(hObject, event...
pushbutton1_Callback(hObject...
FileMenu_Callback(hObject, ev...
OpenMenuItem_Callback(hObj...
PrintMenuItem_Callback(hObje...
CloseMenuItem_Callback(hObj...
popupmenu1_Callback(hObjec...
Editor – /Users/camilleraillon/Documents/MATLAB/120611 test GUI/GUI.m
1.0
1.1
Workspace
Select data t...
Name
Value
Min
Ma
Command Window
Start

```
function varargout = GUI(varargin)
% GUI MATLAB code for GUI.fig
%      GUI, by itself, creates a new GUI or raises the existing
%      singleton*.
%
%      H = GUI returns the handle to a new GUI or the handle to
%      the existing singleton*.
%
%      GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI.M with the given input arguments.
%
%      GUI('Property','Value',...) creates a new GUI or raises the
%      existing singleton*.  Starting from the left, property value pairs
%      applied to the GUI before GUI_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property applicat
%      stop.  All inputs are passed to GUI_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only or
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI

% Last Modified by GUIDE v2.5 06-Jun-2012 17:48:02

% Begin initialization code - DO NOT EDIT
```

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
>> GUI
>>
```

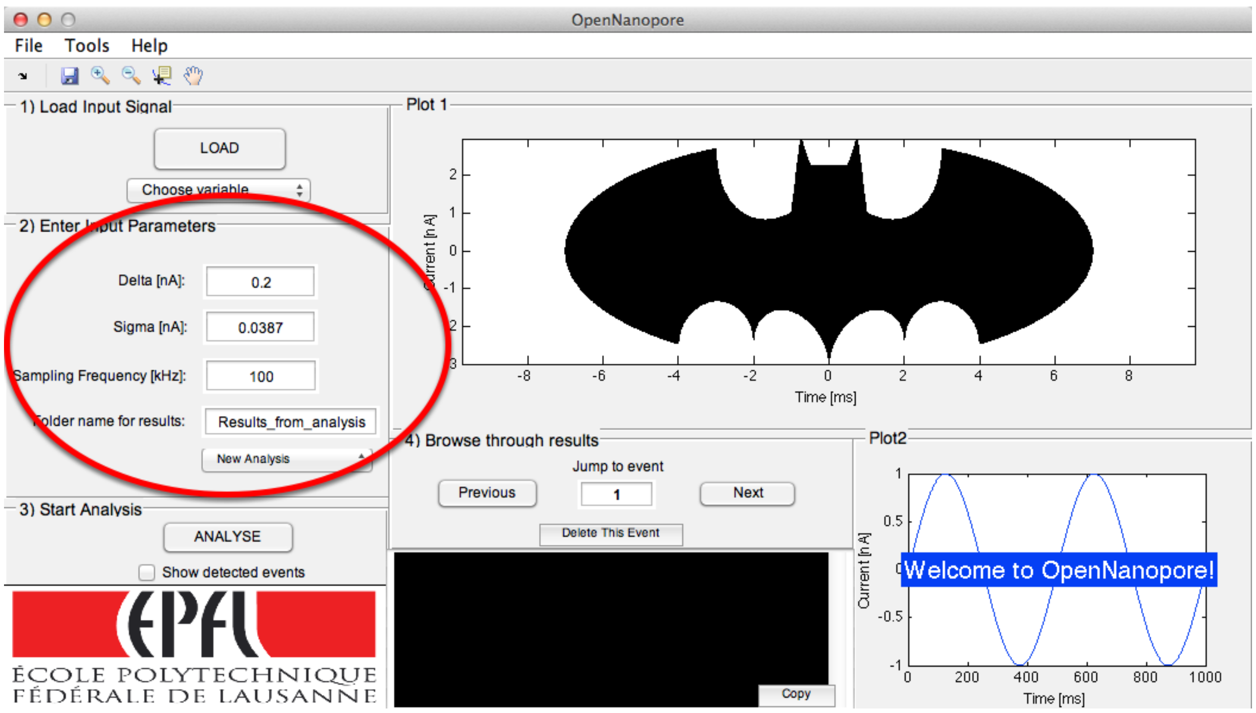
1) Load your signal and start the analysis



Delta [nA]: 0.2
Sigma [nA]: 0.0387
Sampling Frequency [kHz]: 100
Folder name for results: Results_from_analysis
New Analysis
Jump to event: 1

Prepare your signal in a .mat file. See User Guide scripts if you have any problem with the input signal.

2) Enter Input Parameters



2) Enter Input Parameters

Delta [nA]: 0.2

Sigma [nA]: 0.0387

Sampling Frequency [kHz]: 100

Folder name for results: Results_from_analysis

New Analysis

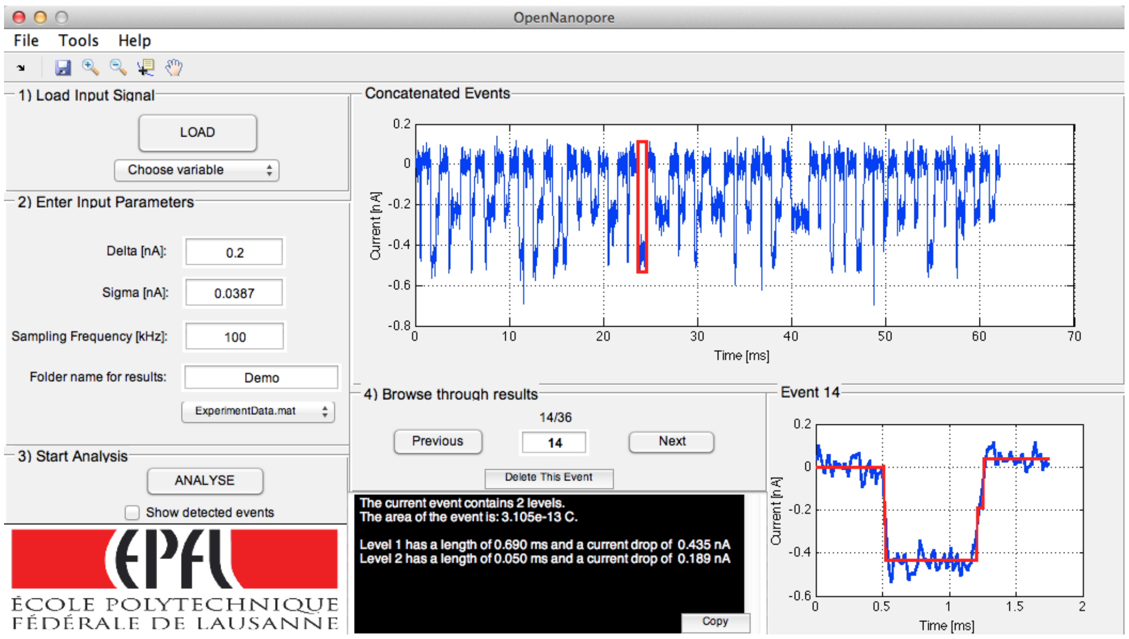
a) input the value of **delta**: most likely current drop in nA that you want to detect in an efficient manner (in the demo case 0.2 nA is a typical current blockage value for □ DNA translocation)

b) input the value of **sigma**: RMS current noise value, can be calculated using *std()* function of MATLAB on 1000 points of baseline or more

c) input the **sampling frequency** used when recording the signal in Hz

e) give a name to the **folder** in which your results will be stored

3) Validate Results

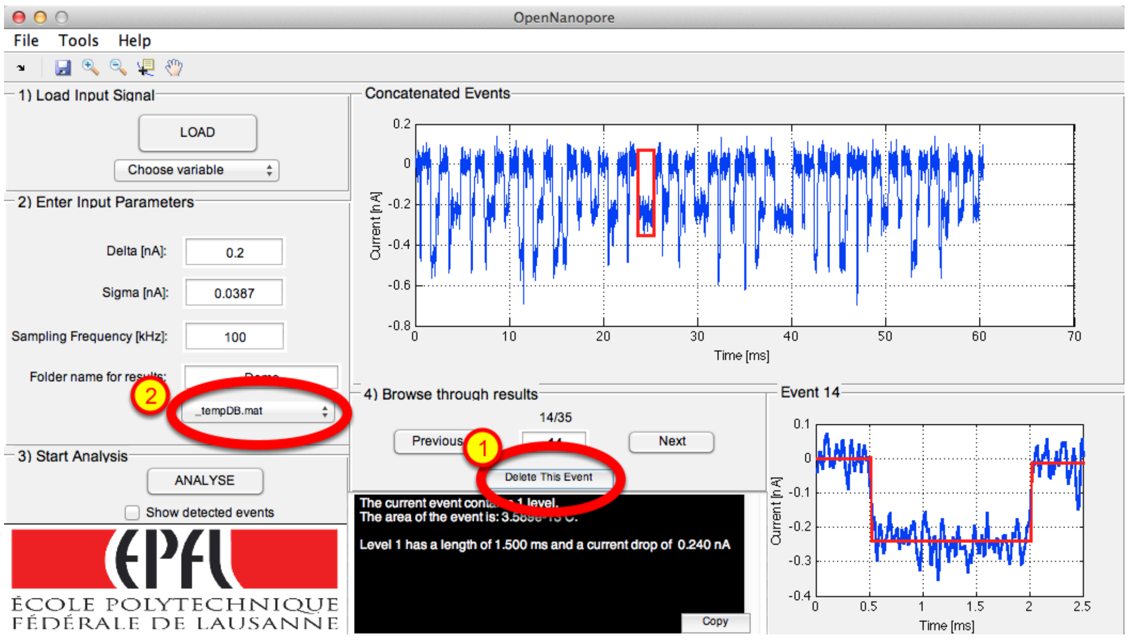


Delta [nA]: 0.2
Sigma [nA]: 0.0387
Sampling Frequency [kHz]: 100
Folder name for results: Demo
ExperimentData.mat

```
The current event contains 2 levels.
The area of the event is: 3.105e-13 C.

Level 1 has a length of 0.690 ms and a current drop of 0.435 nA
Level 2 has a length of 0.050 ms and a current drop of 0.189 nA
```

Look at the fitted events using the "previous" and "next" buttons, check that the recursive filter found all events and that the CUSUM algorithm fitted all events correctly. Alternatively to the buttons you can use the left and right arrow key of your keyboard.

4) Delete unwanted events

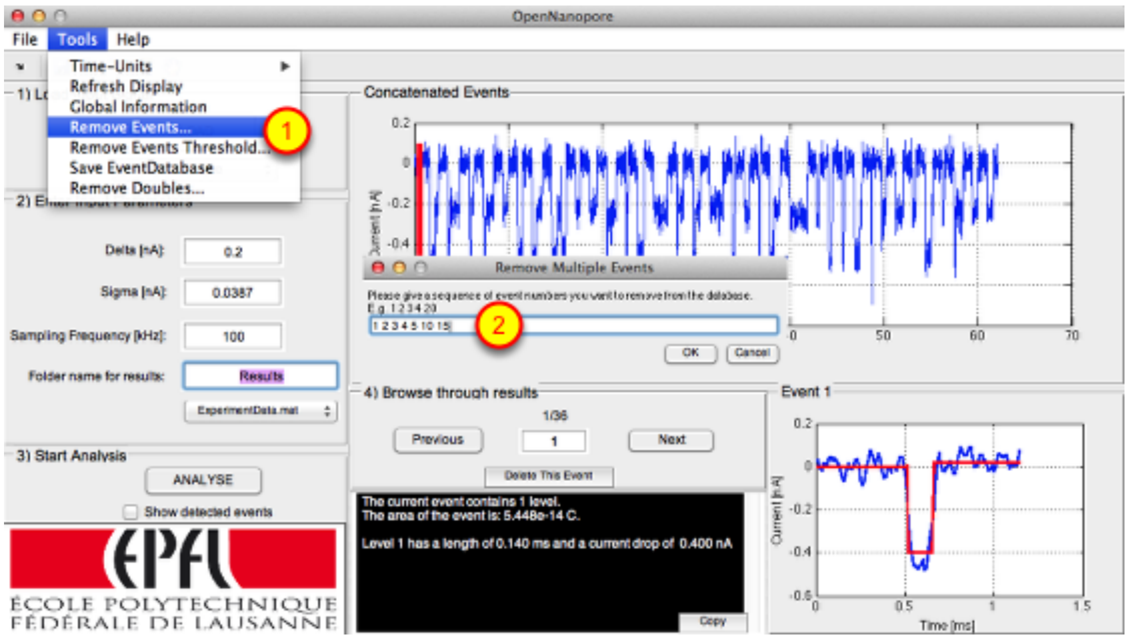


The current event contains 1 level.
The area of the event is: 3.589e-13 C.

Level 1 has a length of 1.500 ms and a current drop of 0.240 nA

If you want to delete an event from the database you can do so by using the delete button or the backspace button of your keyboard. A new temporary database is created. It will be deleted when the program is closed.

5) Remove multiple events (1)



Delta [nA]: 0.2

Sigma [nA]: 0.0387

Sampling Frequency [kHz]: 100

Folder name for results: Results

ExperimentData.mat

Please give a sequence of event numbers you want to remove from the database.
E.g. 1 2 3 4 20

1 2 3 4 5 10 15

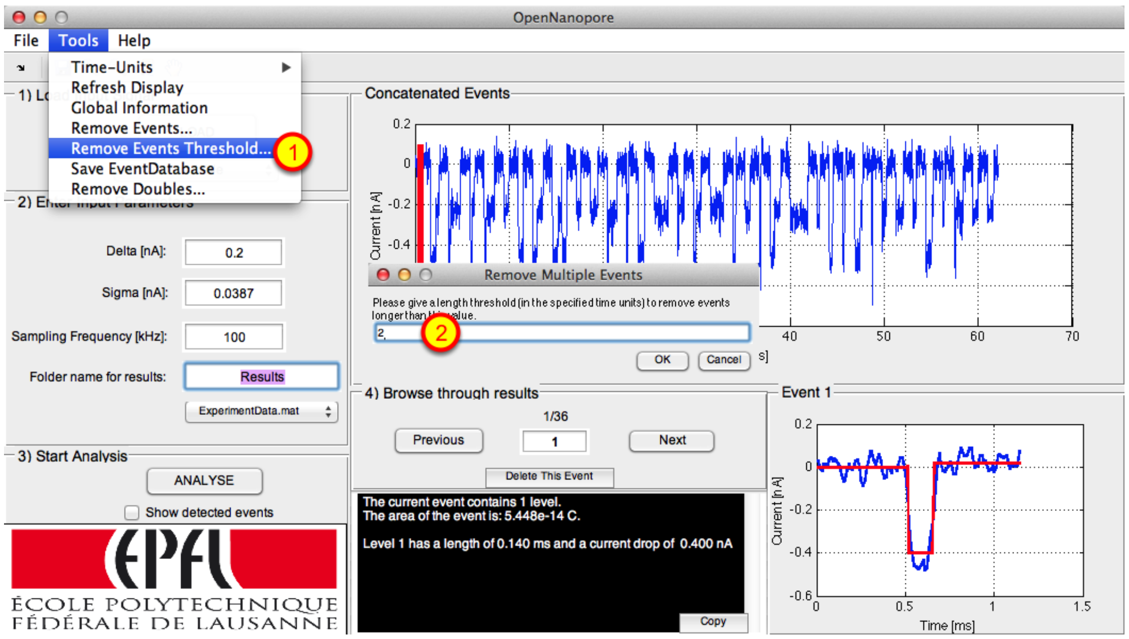
1/36

1

```
The current event contains 1 level.
The area of the event is: 5.448e-14 C.

Level 1 has a length of 0.140 ms and a current drop of 0.400 nA
```

You can remove multiple unwanted events by specifying their event numbers. Click on **Remove Events** in the **Tools** menu. The specify the event numbers you want to remove in the pop-up window, e.g. 1 2 3 4 5 10 15

6) Remove multiple events (2)



Please give a length threshold (in the specified time units) to remove events longer than this value.

2

Delta [nA]: 0.2

Sigma [nA]: 0.0387

Sampling Frequency [kHz]: 100

Folder name for results: Results

ExperimentData.mat

1/36

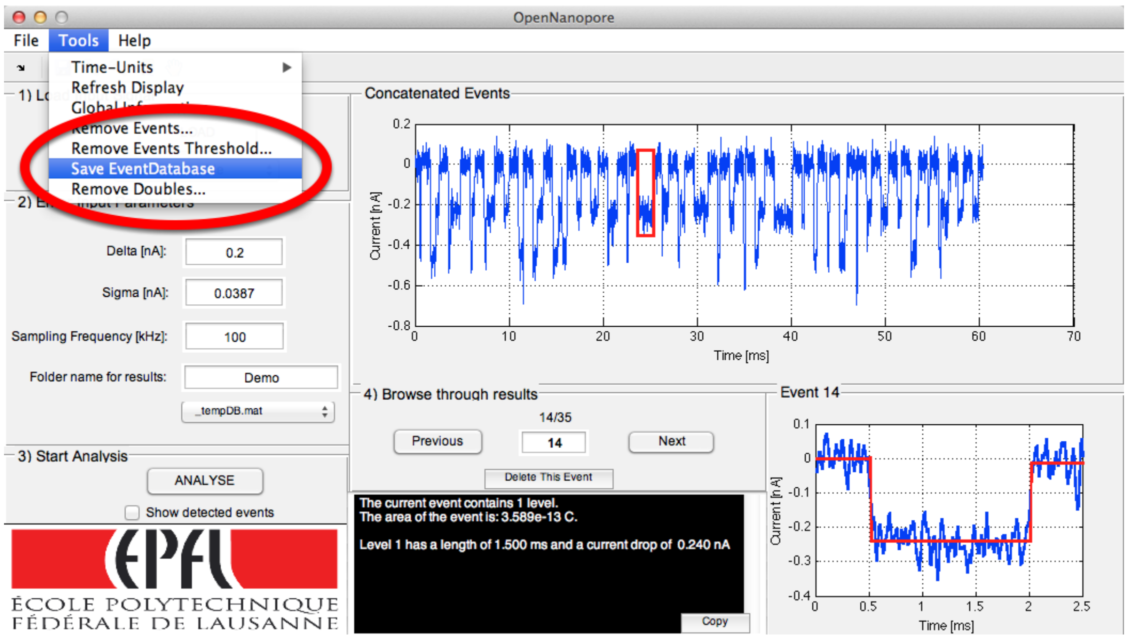
1

The current event contains 1 level.
The area of the event is: 5.448e-14 C.

Level 1 has a length of 0.140 ms and a current drop of 0.400 nA

Alternatively you can use a time threshold deleting all events longer than the specified threshold. In this case the threshold has to be specified in the used units, e.g. ms. Click on **Remove Events Threshold** in the **Tools** menu and specify the length threshold, e.g. **2** for 2ms.

7) Save a database



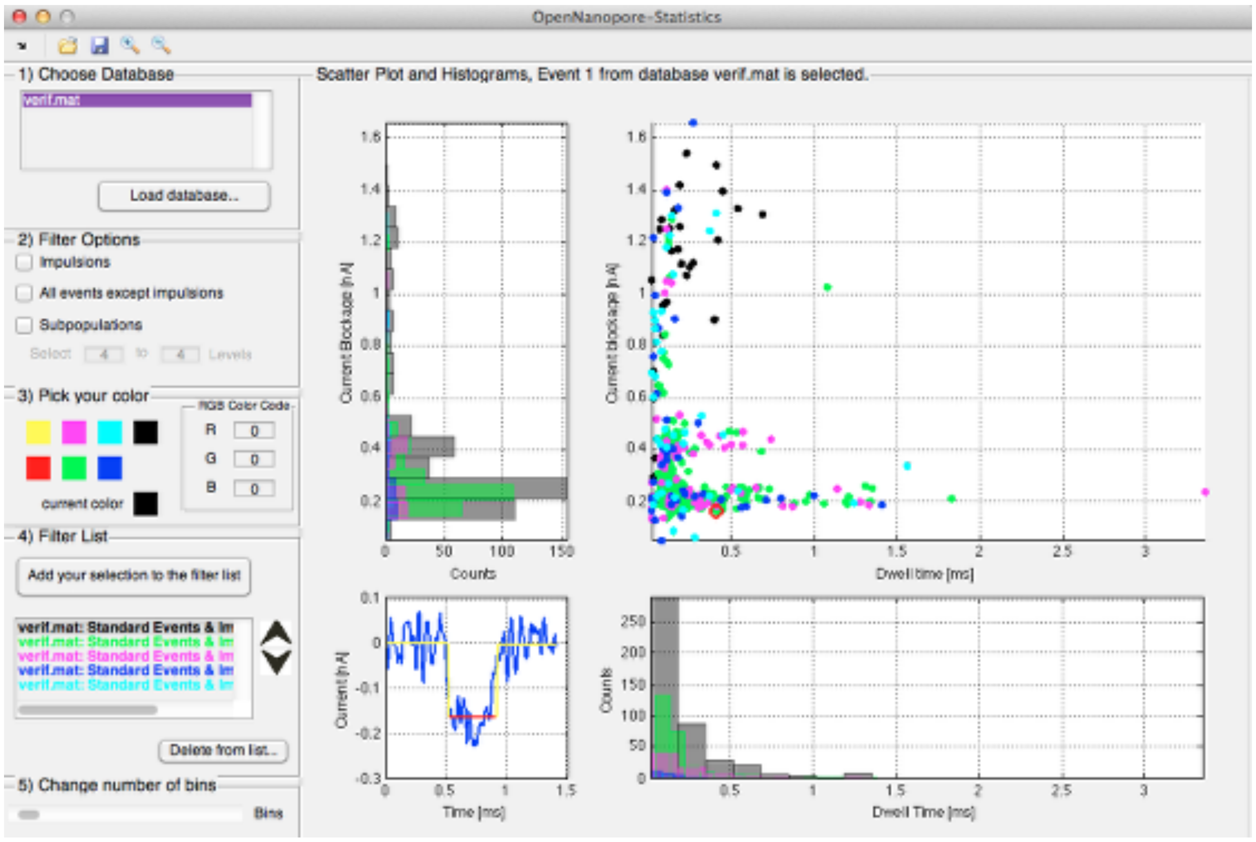
The current event contains 1 level.
The area of the event is: 3.589e-13 C.

Level 1 has a length of 1.500 ms and a current drop of 0.240 nA

If you want to save a modified database use the menu item *Save EventDatabase* in the menu *Tools*. A pop up window will ask you to specify the name of this new database. It is then stored inside your analysis-folder.

How to use the statistics GUI

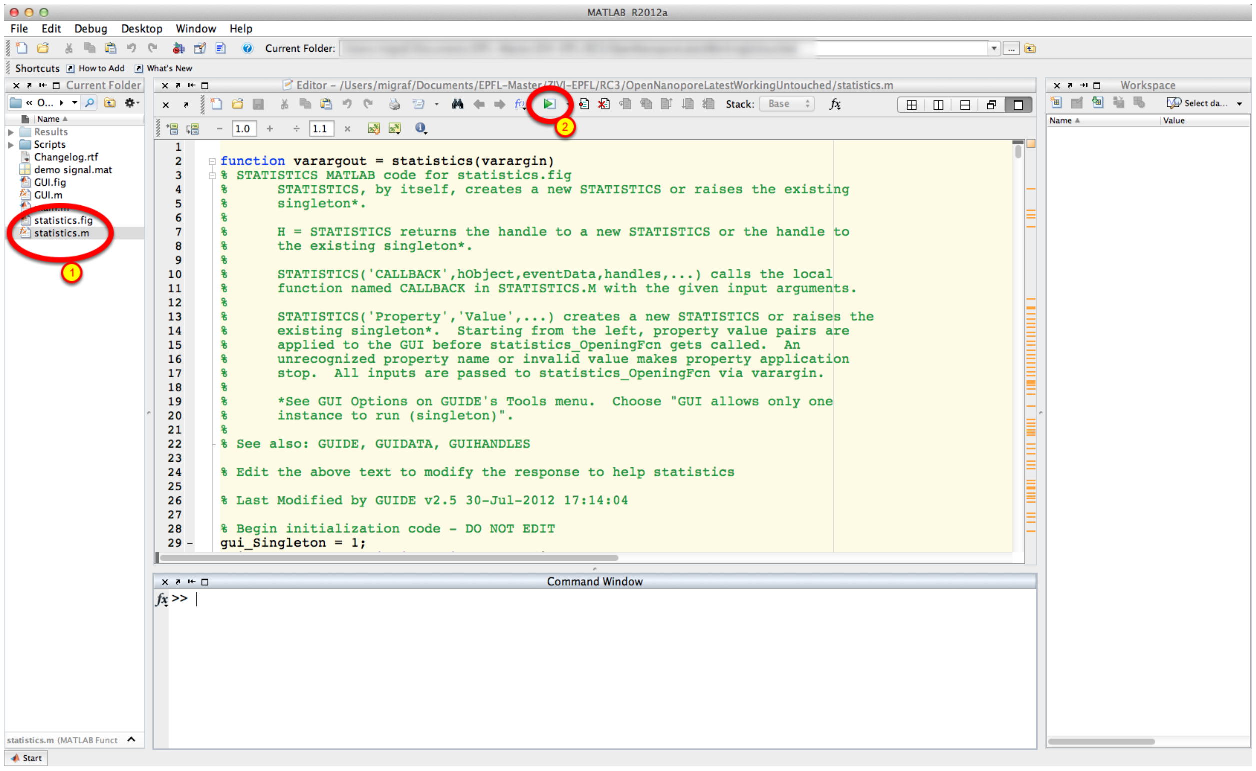
In this section the statistics GUI is introduced.



Scatter Plot and Histograms, Event 1 from database verif.mat is selected.

verif.mat: Standard Events & Im
verif.mat: Standard Events & Im
verif.mat: Standard Events & Im
verif.mat: Standard Events & Im
verif.mat: Standard Events & Im

Run statistics.m

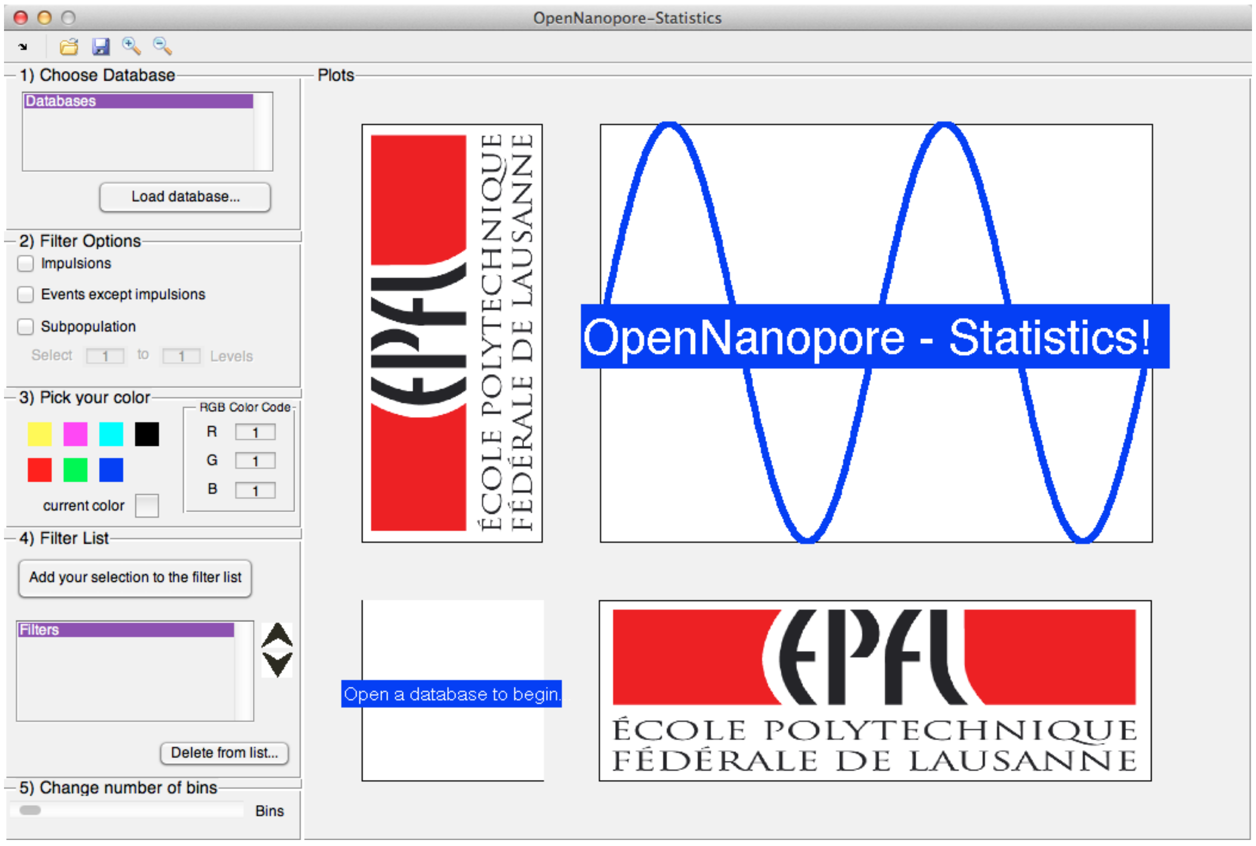


```

function varargout = statistics(varargin)
% STATISTICS MATLAB code for statistics.fig
%      STATISTICS, by itself, creates a new STATISTICS or raises the existing
%      singleton*.
%
%      H = STATISTICS returns the handle to a new STATISTICS or the handle to
%      the existing singleton*.
%
%      STATISTICS('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in STATISTICS.M with the given input arguments.
%
%      STATISTICS('Property','Value',...) creates a new STATISTICS or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before statistics_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to statistics_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help statistics

% Last Modified by GUIDE v2.5 30-Jul-2012 17:14:04

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
```

```
>> |
```

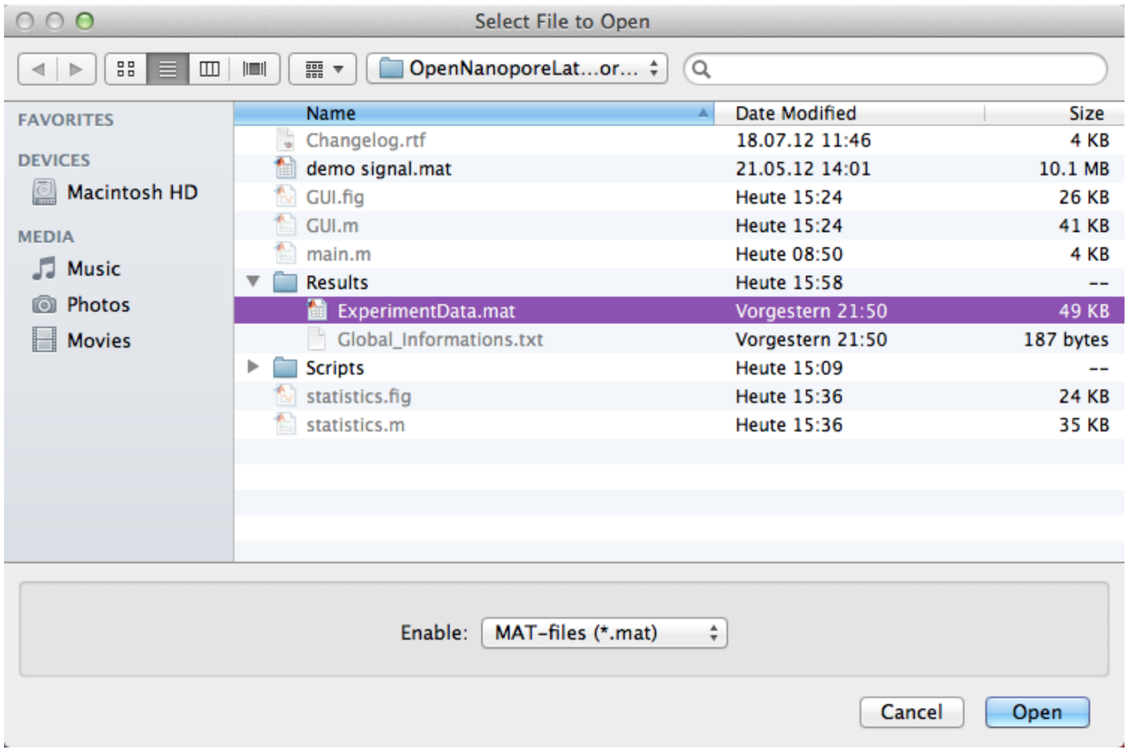
Open *statistics.m* and click run.

Start Screen



This is the start screen of the statistics gui. The idea of the statistics gui is to visualize and inspect different combinations of subpopulations of your events. Panel 1 to 3 allows you to choose different combinations of databases, subpopulations and colors. Lets go through the different options.

1) Choose Database

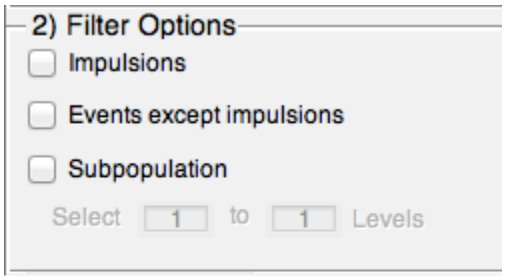


Name	Date Modified	Size
Changelog.rtf	18.07.12 11:46	4 KB
demo signal.mat	21.05.12 14:01	10.1 MB
GUI.fig	Heute 15:24	26 KB
GUI.m	Heute 15:24	41 KB
main.m	Heute 08:50	4 KB
Results	Heute 15:58	--
ExperimentData.mat	Vorgestern 21:50	49 KB
Global_Informations.txt	Vorgestern 21:50	187 bytes
Scripts	Heute 15:09	--
statistics.fig	Heute 15:36	24 KB
statistics.m	Heute 15:36	35 KB

Enable: MAT-files (*.mat)

In this panel you can load different event databases generated by the analysis GUI. Just click on **Load database** and choose the desired event database. Only event databases created by the analysis GUI or the scripts can be read.

2) Filter Options

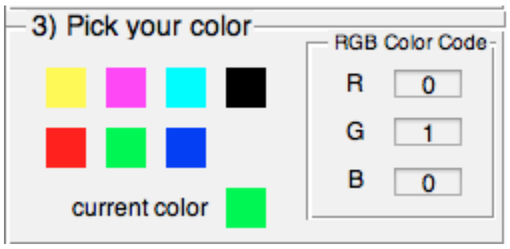


The second panel lets you decide which subpopulations you want to visualize:

- Events that have been detected as being impulsions

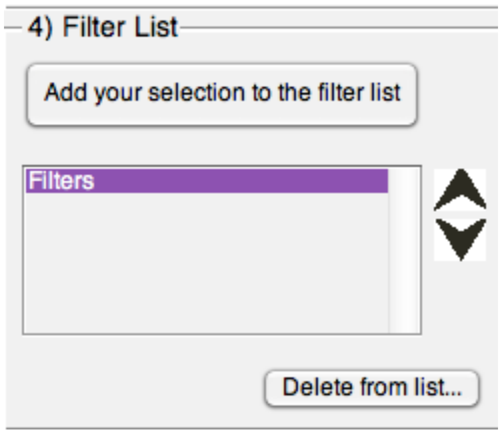
- All events except impulsions
- Subpopulations based on their number of levels

3) Color Picker



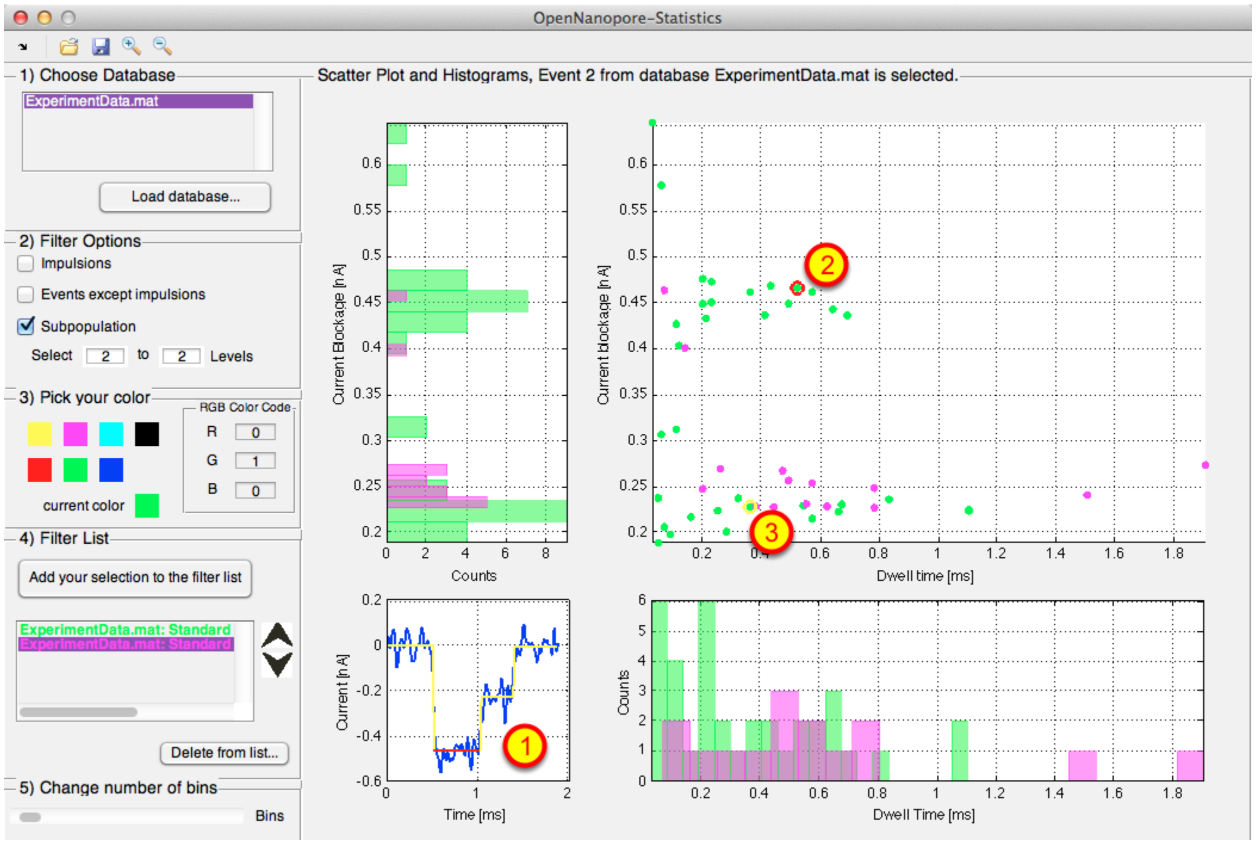
The third panel lets you assign a color to the previously chosen settings. Just pick one of the predefined colors or enter an RGB code in the fields on the right hand. Please note that the values have to been between 0 and 1.The current color square shows you the currently selected color.

4) Filter List



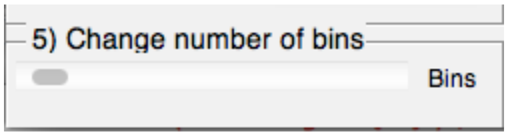
Clicking on the **Add your selection to the filter list** button adds the combination of panels 1 to 3 to your filter list. If you want to delete an item from the list use the **Delete from list** button. The up-and down arrows allow reordering of the list and consequently change the look of the plots.

Visualization



In this example two settings have been chosen. The pink color corresponds to one-level events and the green color to two-level events. You can click on points of the scatter plot to visualize (1) the corresponding event. Selected dots are circled in red (2), whereas dots corresponding to other levels of the same event are circled in yellow (3).

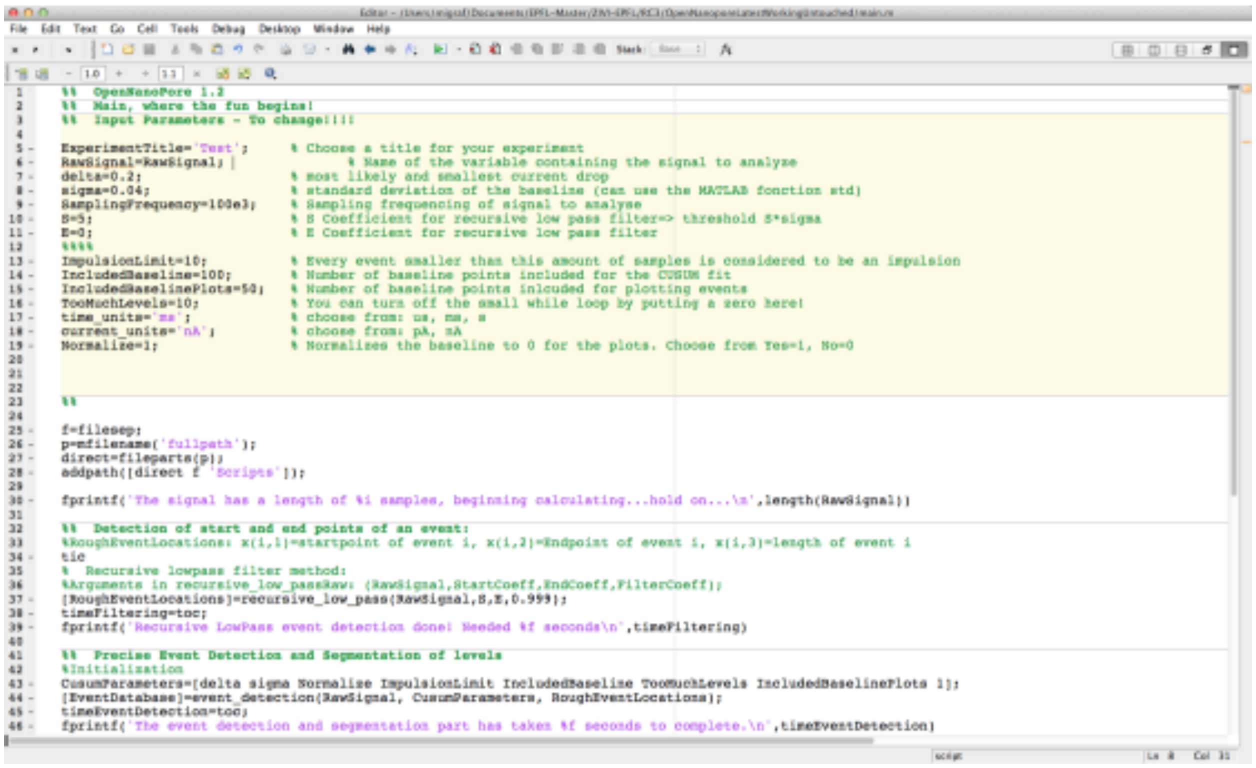
Histogram settings



You can change the number of bins used to plot the histograms with the help of this slider.

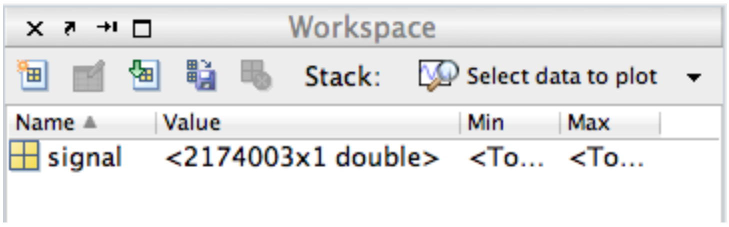
Using the Scripts

In this section you will learn how to run an analysis using the scripts.



```
%% OpenNanoPore 1.2
%% Main, where the fun begins!
%% Input Parameters - To change!!!!

ExperimentTitle='Test';        % Choose a title for your experiment
RawSignal=RawSignal;                   % Name of the variable containing the signal to analyze
delta=0.2;                     % most likely and smallest current drop
sigma=0.04;                    % standard deviation of the baseline (can use the MATLAB fonction std)
SamplingFrequency=100e3;       % Sampling frequencing of signal to analyse
S=5;                           % S Coefficient for recursive low pass filter=> threshold S*sigma
E=0;                           % E Coefficient for recursive low pass filter
%%%%
ImpulsionLimit=10;             % Every event smaller than this amount of samples is considered to be an impulsion
IncludedBaseline=100;          % Number of baseline points included for the CUSUM fit
IncludedBaselinePlots=50;      % Number of baseline points inlcuded for plotting events
TooMuchLevels=10;              % You can turn off the small while loop by putting a zero here!
time_units='ms';               % choose from: us, ms, s
current_units='nA';            % choose from: pA, nA
Normalize=1;                   % Normalizes the baseline to 0 for the plots. Choose from Yes=1, No=0



%%

f=filesep;
p=mfilename('fullpath');
direct=fileparts(p);
addpath([direct f 'Scripts']);

fprintf('The signal has a length of %i samples, beginning calculating...hold on...\n',length(RawSignal))

%% Detection of start and end points of an event:
%RoughEventLocations: x(i,1)=startpoint of event i, x(i,2)=Endpoint of event i, x(i,3)=length of event i
tic
% Recursive lowpass filter method:
%Arguments in recursive_low_passRaw: (RawSignal,StartCoeff,EndCoeff,FilterCoeff);
[RoughEventLocations]=recursive_low_pass(RawSignal,S,E,0.999);
timeFiltering=toc;
fprintf('Recursive LowPass event detection done! Needed %f seconds\n',timeFiltering)

%% Precise Event Detection and Segmentation of levels
%Initialization
CusumParameters=[delta sigma Normalize ImpulsionLimit IncludedBaseline TooMuchLevels IncludedBaselinePlots 1];
[EventDatabase]=event_detection(RawSignal, CusumParameters, RoughEventLocations);
timeEventDetection=toc;
fprintf('The event detection and segmentation part has taken %f seconds to complete.\n',timeEventDetection)
```

1) Load your signal in the workspace

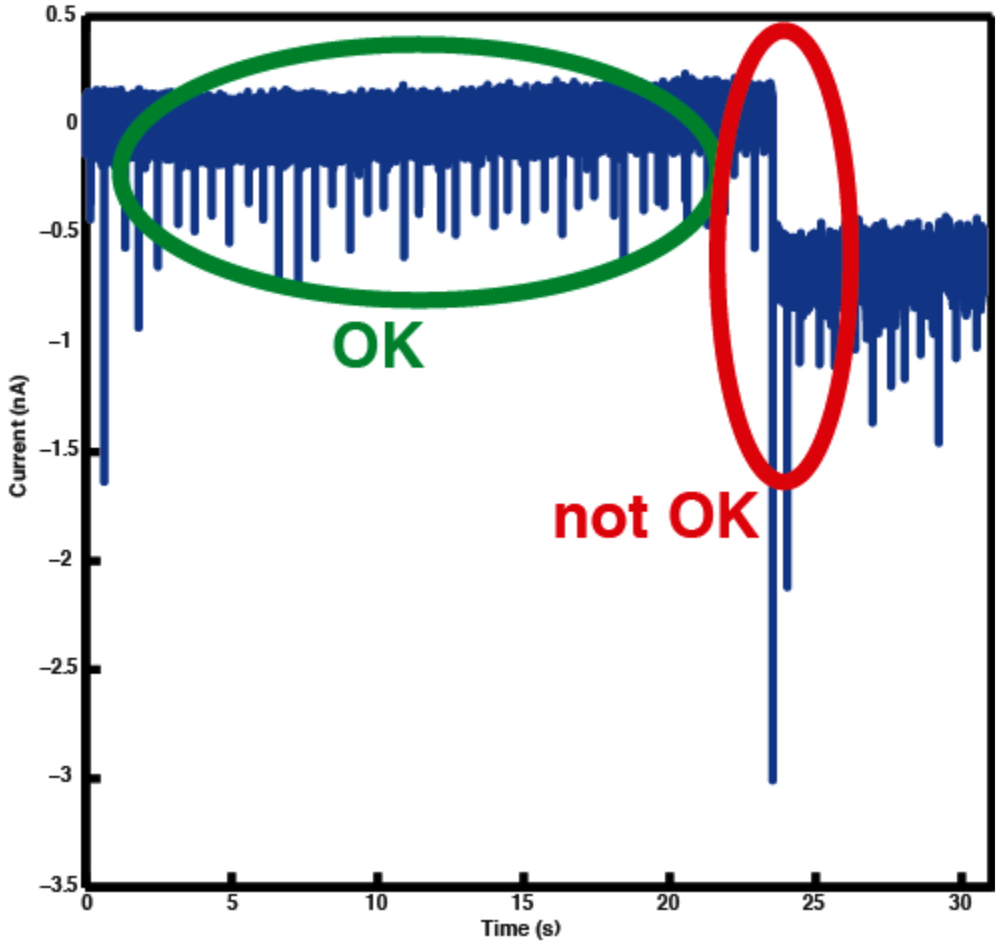


Name ▲	Value	Min	Max
signal	<2174003x1 double>	<To...	<To...

FORMAT of input signal: the input signal need to be in a double of <nx1> ionic current values in **nA.**

Your input signal **MUST** contain baseline information, at least 100 points of baseline before and after each event.

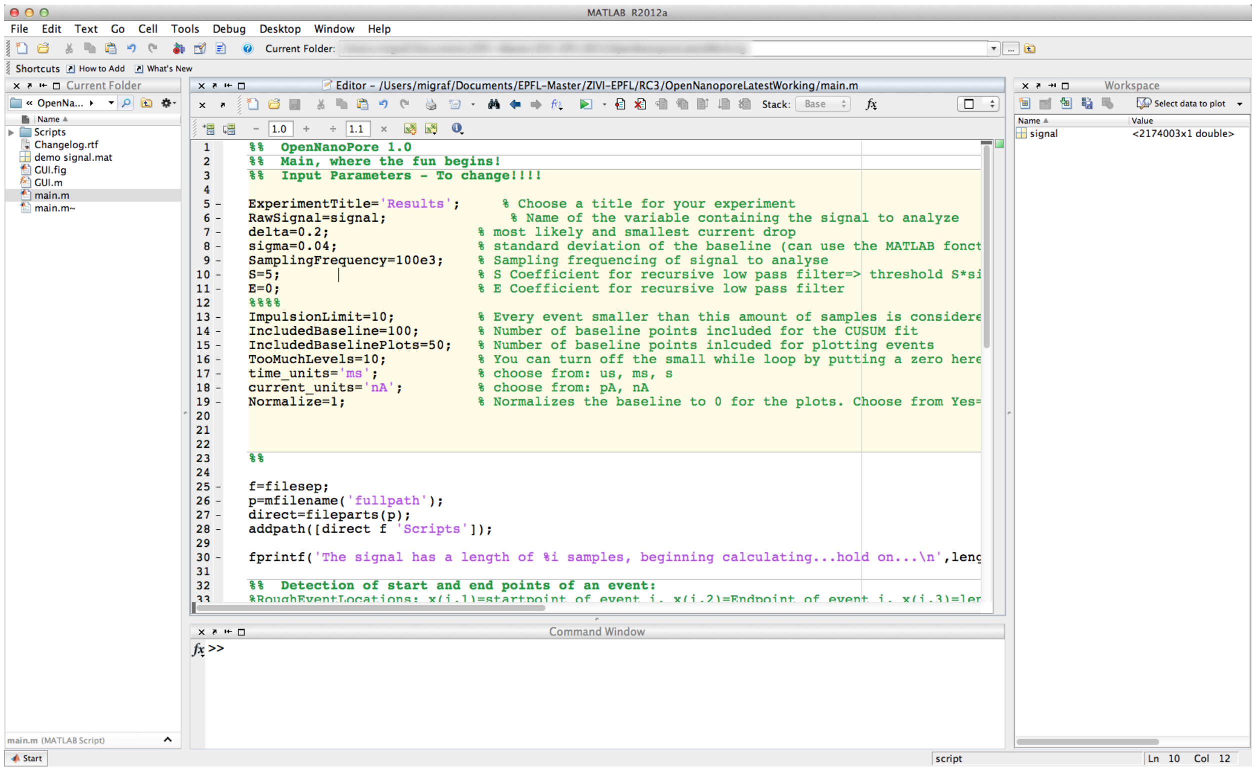
2) Check your signal by plotting it



Note that the baseline shouldn't contain any sudden jump...

If you have a problem loading the signal please email us! See address on website: http://lben.epfl.ch/page-79460-en.html

2) Open main.m and fill in the input parameters



Editor – /Users/migraf/Documents/EPFL-Master/ZIVI-EPFL/RC3/OpenNanoporeLatestWorking/main.m

```
1     %%  OpenNanoPore 1.0
2     %%  Main, where the fun begins!
3     %%  Input Parameters - To change!!!!
4
5  -  ExperimentTitle='Results';     % Choose a title for your experiment
6  -  RawSignal=signal;               % Name of the variable containing the signal to analyze
7  -  delta=0.2;                    % most likely and smallest current drop
8  -  sigma=0.04;                   % standard deviation of the baseline (can use the MATLAB fonct
9  -  SamplingFrequency=100e3;      % Sampling frequencing of signal to analyse
10 -  S=5;                          % S Coefficient for recursive low pass filter=> threshold S*si
11 -  E=0;                          % E Coefficient for recursive low pass filter
12    %%%%
13 -  ImpulsionLimit=10;            % Every event smaller than this amount of samples is considere
14 -  IncludedBaseline=100;         % Number of baseline points included for the CUSUM fit
15 -  IncludedBaselinePlots=50;     % Number of baseline points inlcuded for plotting events
16 -  TooMuchLevels=10;             % You can turn off the small while loop by putting a zero here
17 -  time_units='ms';              % choose from: us, ms, s
18 -  current_units='nA';           % choose from: pA, nA
19 -  Normalize=1;                  % Normalizes the baseline to 0 for the plots. Choose from Yes=
20
21
22
23    %%
24
25 -  f=filesep;
26 -  p=mfilename('fullpath');
27 -  direct=fileparts(p);
28 -  addpath([direct f 'Scripts']);
29
30 -  fprintf('The signal has a length of %i samples, beginning calculating...hold on...\n',leng
31
32    %%  Detection of start and end points of an event:
33    %RoughEventLocations: x(i,1)=startpoint of event i, x(i,2)=Endpoint of event i, x(i,3)=ler
```

Command Window

```
>>
```

Name	Value
signal	<2174003x1 double>

line 5: give a name to the **folder** in which your results will be stored

line 6: input the name of the **signal** you just loaded in the Workspace

line 7: input the value of **delta**: most likely current drop in nA that you want to detect in an efficient manner (in the demo case 0.2 nA is a typical current blockage value for DNA translocation)

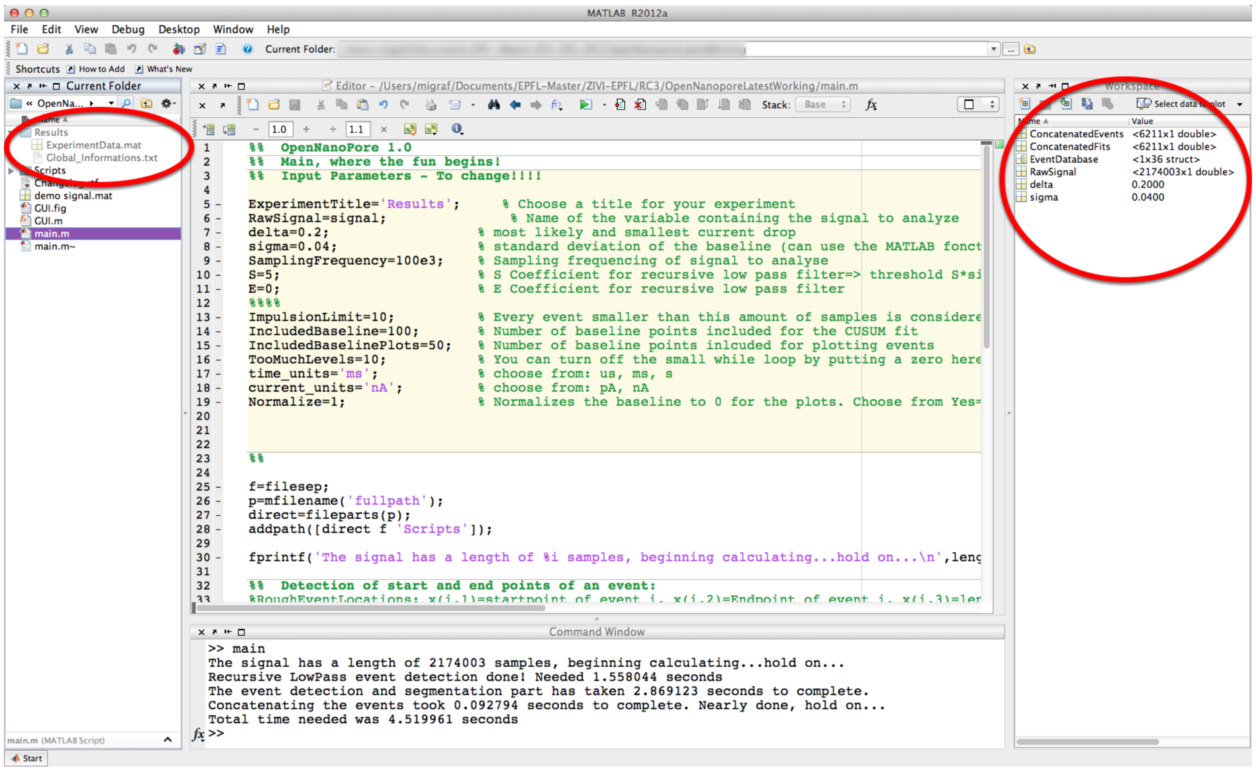
line 8: input the value of **sigma**: RMS current noise value, can be calculated using *std()* function of MATLAB on 1000 points of baseline or more

line 9: input the **sampling frequency** used when recording the signal in Hz

3) run the main

You can either press the button shown above or write *main* in the Command Window.

4) look at the results



```
%%  OpenNanoPore 1.0
%%  Main, where the fun begins!
%%  Input Parameters - To change!!!!

ExperimentTitle='Results';      % Choose a title for your experiment
RawSignal=signal;                % Name of the variable containing the signal to analyze
delta=0.2;                     % most likely and smallest current drop
sigma=0.04;                    % standard deviation of the baseline (can use the MATLAB fonct
SamplingFrequency=100e3;       % Sampling frequencing of signal to analyse
S=5;                           % S Coefficient for recursive low pass filter=> threshold S*si
E=0;                           % E Coefficient for recursive low pass filter
%%%%
ImpulsionLimit=10;             % Every event smaller than this amount of samples is considere
IncludedBaseline=100;          % Number of baseline points included for the CUSUM fit
IncludedBaselinePlots=50;      % Number of baseline points inlcuded for plotting events
TooMuchLevels=10;              % You can turn off the small while loop by putting a zero here
time_units='ms';               % choose from: us, ms, s
current_units='nA';            % choose from: pA, nA
Normalize=1;                   % Normalizes the baseline to 0 for the plots. Choose from Yes=



%%

f=filesep;
p=mfilename('fullpath');
direct=fileparts(p);
addpath([direct f 'Scripts']);

fprintf('The signal has a length of %i samples, beginning calculating...hold on...\n',leng

%%  Detection of start and end points of an event:
%RoughEventLocations: x(i,1)=startpoint of event i, x(i,2)=Endpoint of event i, x(i,3)=ler
```

```
>> main
The signal has a length of 2174003 samples, beginning calculating...hold on...
Recursive LowPass event detection done! Needed 1.558044 seconds
The event detection and segmentation part has taken 2.869123 seconds to complete.
Concatenating the events took 0.092794 seconds to complete. Nearly done, hold on...
Total time needed was 4.519961 seconds
>>
```

Name	Value
ConcatenatedEvents	<6211x1 double>
ConcatenatedFits	<6211x1 double>
EventDatabase	<1x36 struct>
RawSignal	<2174003x1 double>
delta	0.2000
sigma	0.0400

A .mat file called ExperimentData is created containing all results of an analysis:

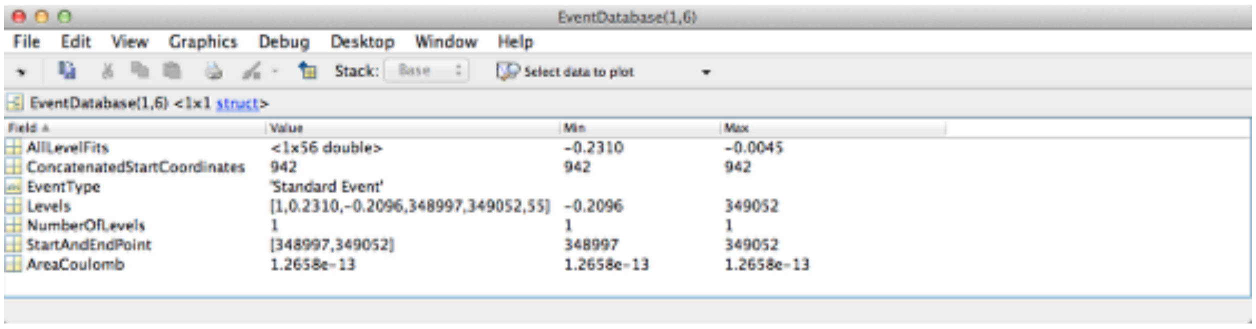
EventDatabase, ConcatenatedEvents and ConcatenatedFits and the settings of the analysis.

EventDatabase is a structure array with level and dwell time information for each event (see Event Database for more information) ConcatenatedEvents is a concatenation of all the detected events with 100 points of baseline between each event, ConcatenatedFits are its corresponding fits.

Introduction to the Event Database

Event Database

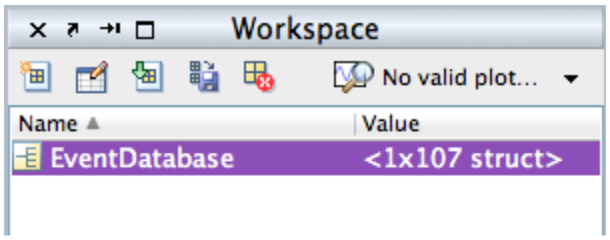
This section contains information on how to use the EventDatabase generated by the program.



EventDatabase(1,6) <1x1 struct>

Field	Value	Min	Max
AllLevelFits	<1x56 double>	-0.2310	-0.0045
ConcatenatedStartCoordinates	942	942	942
EventType	'Standard Event'		
Levels	[1,0.2310,-0.2096,348997,349052,55]	-0.2096	349052
NumberOfLevels	1	1	1
StartAndEndPoint	[348997,349052]	348997	349052
AreaCoulomb	1.2658e-13	1.2658e-13	1.2658e-13

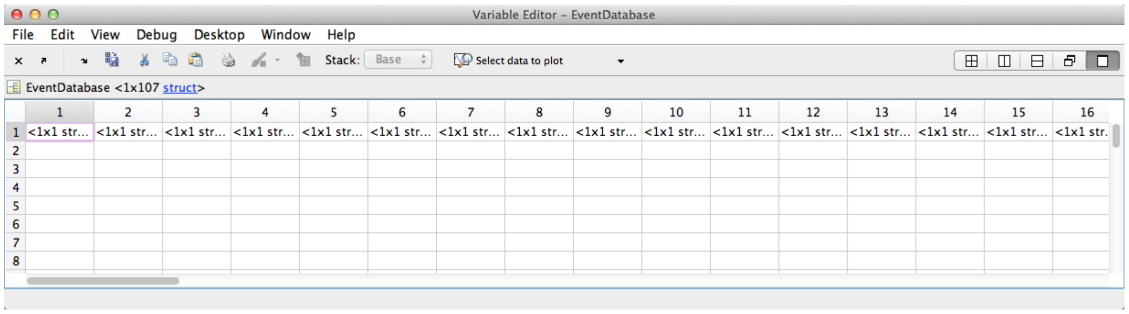
Structure (1)



Name	Value
EventDatabase	<1x107 struct>

The event database generated by the program is an array (vector) of structures. A structure is a data type that groups related data using data containers called fields. Each field can contain data of any type or size. Such types of variables are very readable and the desired information can be found very quickly.

Structure (2)

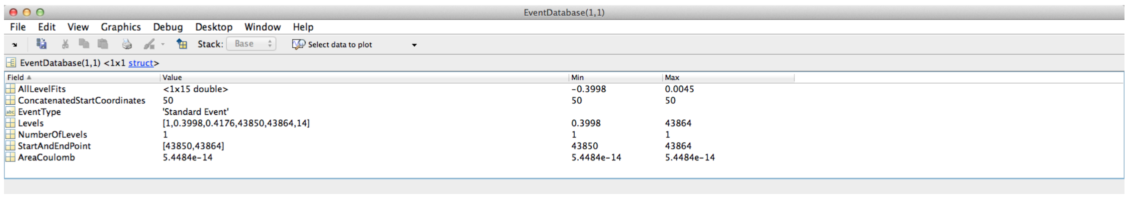

Variable Editor – EventDatabase
File Edit View Debug Desktop Window Help
Stack: Base
Select data to plot

EventDatabase <1x107 struct>

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str...	<1x1 str.
2																
3																
4																
5																
6																
7																
8																

Double-click on the EventDatabase variable opens the variable editor. Each field in this array corresponds to one event. The row number is thus the number of the event as it occured in the raw signal.

Structure (3)

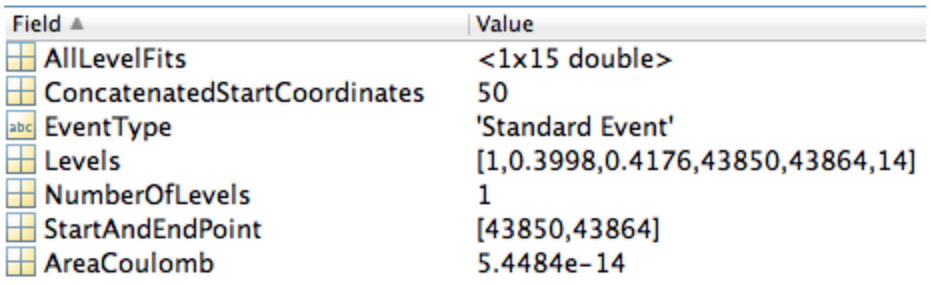


EventDatabase(1,1) <1x1 struct>

Field	Value	Min	Max
AllLevelFits	<1x15 double>	-0.3998	0.0045
ConcatenatedStartCoordinates	50	50	50
EventType	'Standard Event'		
Levels	[1,0.3998,0.4176,43850,43864,14]	0.3998	43864
NumberOfLevels	1	1	1
StartAndEndPoint	[43850,43864]	43850	43864
AreaCoulomb	5.4484e-14	5.4484e-14	5.4484e-14

Another double click on one of the structure-fields reveals the informations about the given event.

Informations available



Field ▲	Value
AllLevelFits	<1x15 double>
ConcatenatedStartCoordinates	50
EventType	'Standard Event'
Levels	[1,0.3998,0.4176,43850,43864,14]
NumberOfLevels	1
StartAndEndPoint	[43850,43864]
AreaCoulomb	5.4484e-14

1. **AllLevelFits**: A vector containing the values of the fit. Please note that all these values are relative to the baseline before the event and do not reflect the actual current drop. Level information on the absolute current drop can be found in the Levels category.

2. **ConcatenatedStartCoordinates**: This value is the location of the event in the corresponding concatenated signal.

3. **EventType:** The type of the event is stored in this variable. This is either 'Standard Event', 'Too many levels', 'Impulsion'or'Impulsion not detected by CUSUM'.

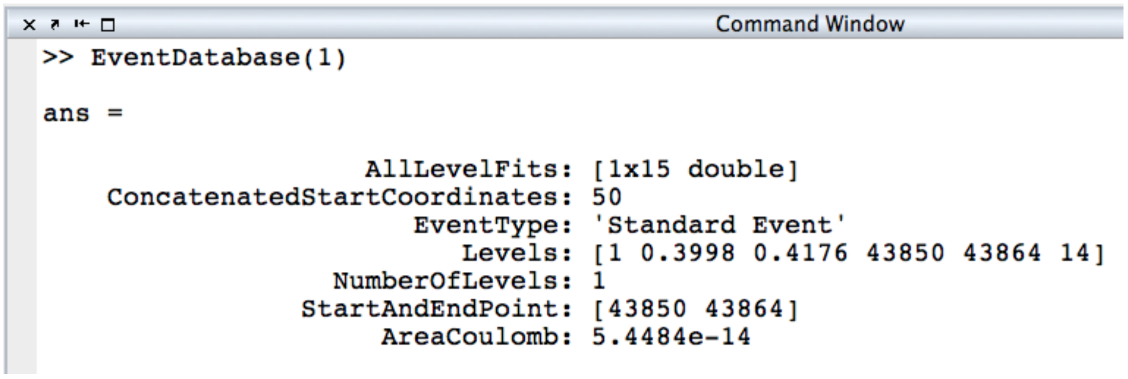
4. **Levels:** This variable contains the most important informations. It recapit- ulates the informations about each level of the event. This variable is a matrix which contains in its rows the different levels of the event. The columns give the following information: [LevelNumber, Relative Fit, Absolute Fit, Start-Point, End-Point, Length of Level]

5. **NumberOfLevels:** The number of levels this event contains

6. **StartAndEndPoint:** Contains the location of the event in the raw signal.

7. **AreaCoulob:** The area of the event (integration of the raw signal)

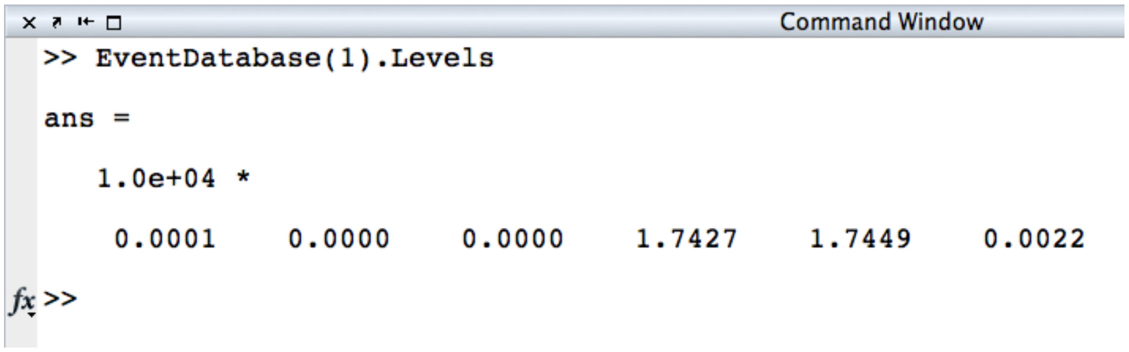
Show all the information about one event



```
>> EventDatabase(1)

ans =

                     AllLevelFits: [1x15 double]
    ConcatenatedStartCoordinates: 50
                        EventType: 'Standard Event'
                           Levels: [1 0.3998 0.4176 43850 43864 14]
                   NumberOfLevels: 1
                 StartAndEndPoint: [43850 43864]
                      AreaCoulomb: 5.4484e-14
```

The command EventDatabase(i), where i is the desired event number, displays all the fields of this structure.

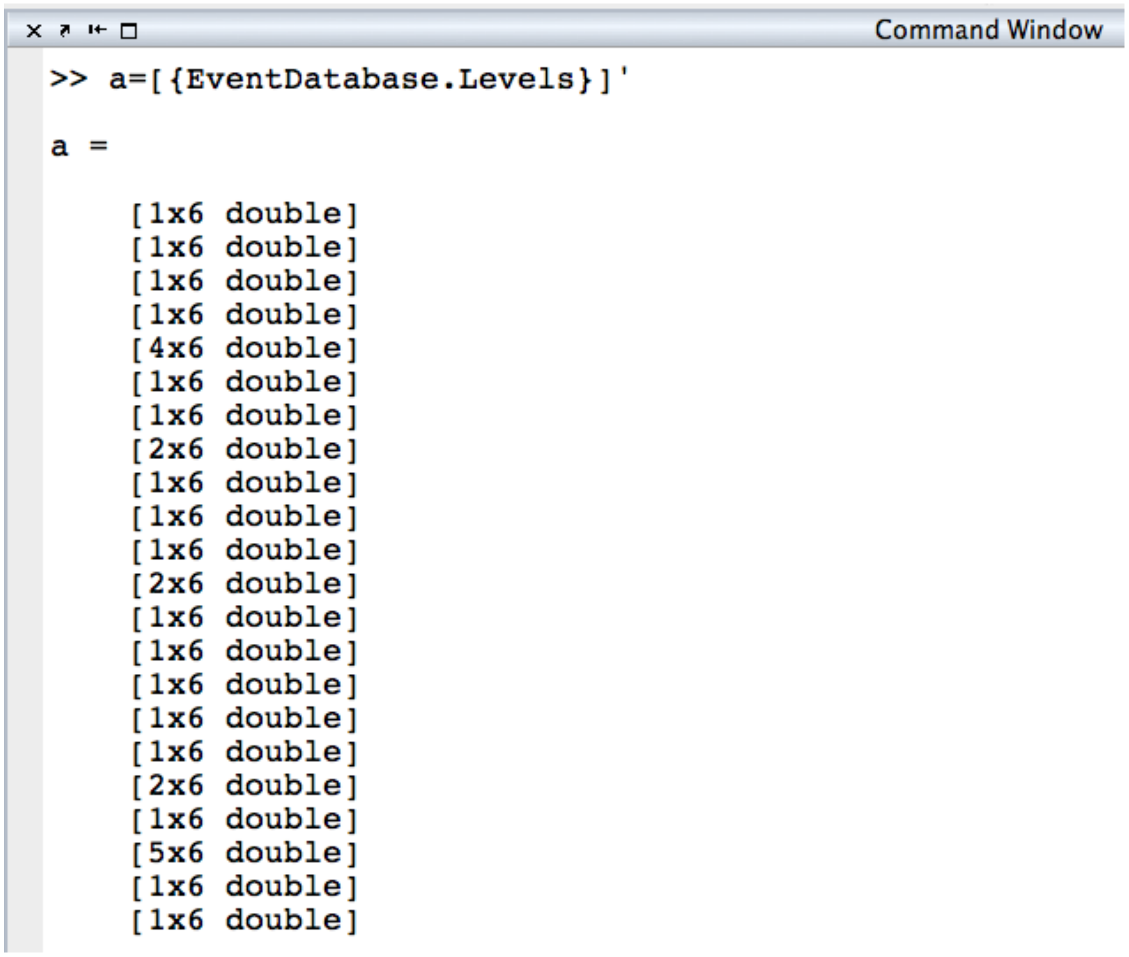
Show specific informations



```
>> EventDatabase(1).Levels

ans =

   1.0e+04 *

    0.0001    0.0000    0.0000    1.7427    1.7449    0.0022

>>
```

The command EventDatabase(i).Fieldname gives direct access to the desired information. Remember Fieldname has to be one of the variables specified before.

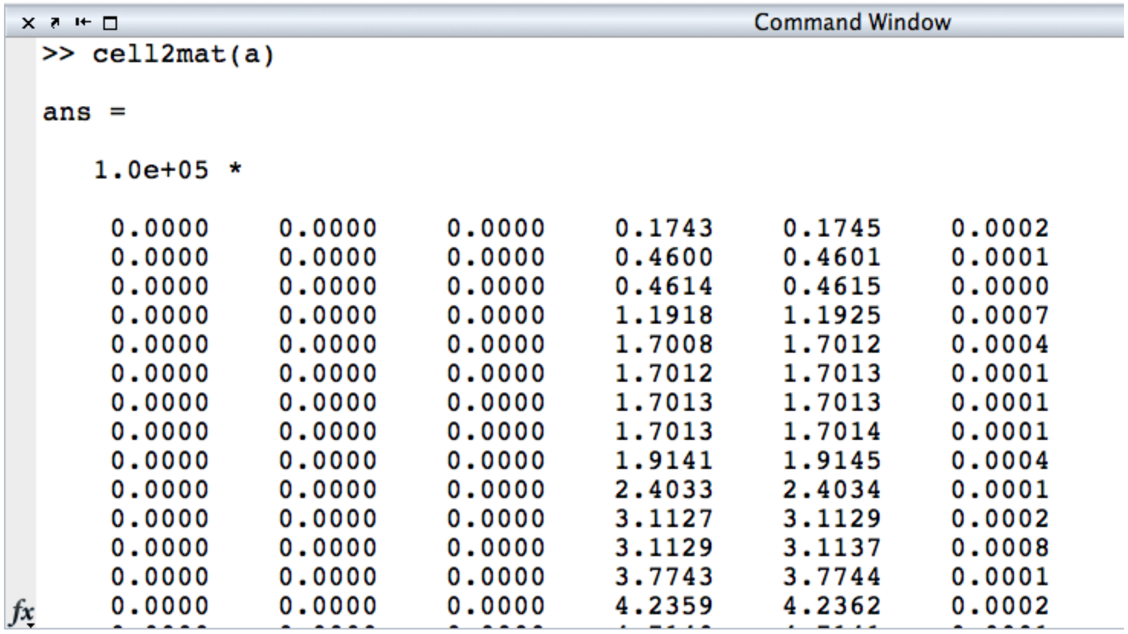
Combine several events



```
>> a=[{EventDatabase.Levels}]'

a =

    [1x6 double]
    [1x6 double]
    [1x6 double]
    [1x6 double]
    [4x6 double]
    [1x6 double]
    [1x6 double]
    [2x6 double]
    [1x6 double]
    [1x6 double]
    [1x6 double]
    [2x6 double]
    [1x6 double]
    [1x6 double]
    [1x6 double]
    [1x6 double]
    [1x6 double]
    [2x6 double]
    [1x6 double]
    [5x6 double]
    [1x6 double]
    [1x6 double]
```

The command a=[{EventDatabase.Levels}]' exports the level informations of all the events to the variable a. If you want to export only a fraction of the events, you can use: a=[{EventDatabase(1-10).Levels}]', which in this case exports only events 1 to 10. The same syntax applies to other fields.

Convert to single matrix



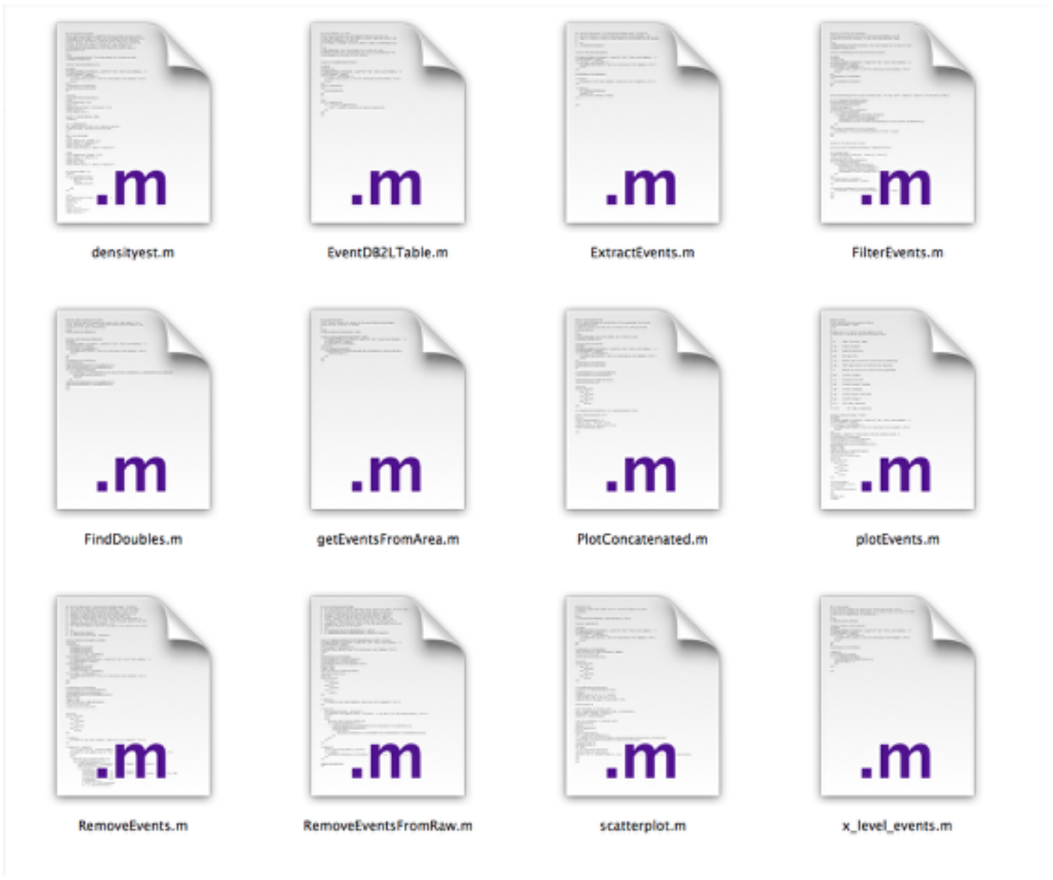
```
>> cell2mat(a)

ans =

   1.0e+05 *

    0.0000    0.0000    0.0000    0.1743    0.1745    0.0002
    0.0000    0.0000    0.0000    0.4600    0.4601    0.0001
    0.0000    0.0000    0.0000    0.4614    0.4615    0.0000
    0.0000    0.0000    0.0000    1.1918    1.1925    0.0007
    0.0000    0.0000    0.0000    1.7008    1.7012    0.0004
    0.0000    0.0000    0.0000    1.7012    1.7013    0.0001
    0.0000    0.0000    0.0000    1.7013    1.7013    0.0001
    0.0000    0.0000    0.0000    1.7013    1.7014    0.0001
    0.0000    0.0000    0.0000    1.9141    1.9145    0.0004
    0.0000    0.0000    0.0000    2.4033    2.4034    0.0001
    0.0000    0.0000    0.0000    3.1127    3.1129    0.0002
    0.0000    0.0000    0.0000    3.1129    3.1137    0.0008
    0.0000    0.0000    0.0000    3.7743    3.7744    0.0001
    0.0000    0.0000    0.0000    4.2359    4.2362    0.0002
```

The command cell2mat(a) can convert the before created cell array to one single matrix containing the level informations of all the events. Conversion to a single matrix is only possible for fields with constant row-size, i.e. all the fields except 'AllLevelFits' and 'EventType'.

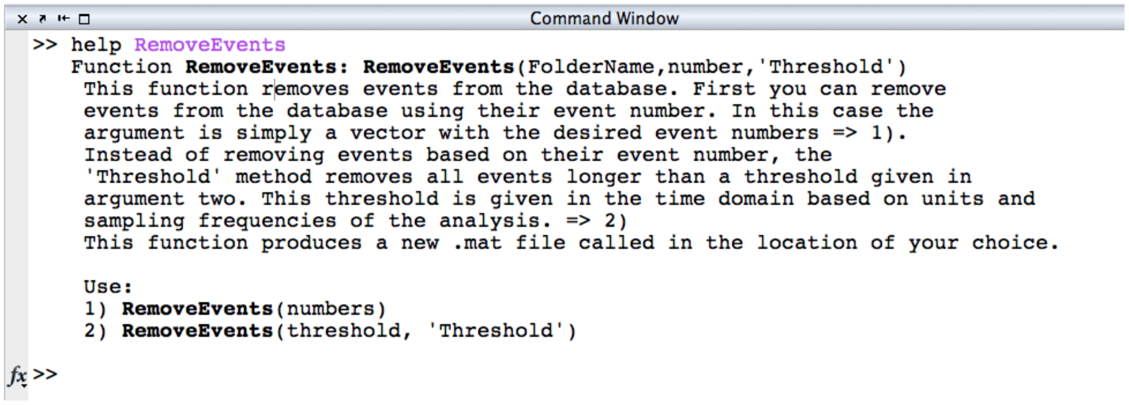
Functions

List of Available Functions

This section lists the available functions.



Scripts


Command Window

```
>> help RemoveEvents
  Function RemoveEvents: RemoveEvents(FolderName,number,'Threshold')
   This function removes events from the database. First you can remove
   events from the database using their event number. In this case the
   argument is simply a vector with the desired event numbers => 1).
   Instead of removing events based on their event number, the
   'Threshold' method removes all events longer than a threshold given in
   argument two. This threshold is given in the time domain based on units and
   sampling frequencies of the analysis. => 2)
   This function produces a new .mat file called in the location of your choice.

   Use:
   1) RemoveEvents(numbers)
   2) RemoveEvents(threshold, 'Threshold')

fx >>
```

The folder scripts is automatically added to your Matlab path when GUI.m is launched or main.m executed, i.e. you do not need to be in the actual directory in order to call the functions.

Here, only a list of the available functions is provided. If you need more information about a function or how to use it you can type the command **help function_name** in the Matlab command window.

List of scripts

- densityest
- EventDB2LTable
- FilterEvents
- histogram
- PlotConcatenated
- RemoveEvents
- RemoveEventsFromRaw
- scatterplot
- x_level_events

- plotEvents
- FindDoubles
- ExtractEvents
- getEventsFromArea