OpenNanopore
# Table of Contents

**User Guide** .................................................................................................................. 1
  - How to use the analysis GUI ......................................................................................... 1
  - How to use the statistics GUI ....................................................................................... 10
  - Using the Scripts ........................................................................................................ 16

**Introduction to the Event Database** ........................................................................... 21
  - Event Database ........................................................................................................... 21

**Functions** .................................................................................................................. 27
  - List of Available Functions ......................................................................................... 27
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 information: https://www.youtube.com/watch?v=McKy_hNy7cg&feature=plcp
0) Open GUI.m and click on "run"
1) Load your signal and start the analysis

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

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

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

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)

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)

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

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.
Run statistics.m

Open statistics.m and click run.
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. Let's go through the different options.
1) Choose Database

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

![Color Picker](image)

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

![Filter List](image)

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.
1) Load your signal in the workspace

**FORMAT of input signal:** the input signal need to be in a double of \(<nx1>\) ionic current values in \(\text{nA}\).

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

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 \textit{main} in the Command Window.

4) look at the results

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 \textit{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.
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.

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

**Informations available**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllLevelFits</td>
<td>&lt;1x15 double&gt;</td>
</tr>
<tr>
<td>ConcatenatedStartCoordinates</td>
<td>50</td>
</tr>
<tr>
<td>EventType</td>
<td>'Standard Event'</td>
</tr>
<tr>
<td>Levels</td>
<td>[1.0.3998,0.4176,43850,43864,14]</td>
</tr>
<tr>
<td>NumberOfLevels</td>
<td>1</td>
</tr>
<tr>
<td>StartAndEndPoint</td>
<td>[43850,43864]</td>
</tr>
<tr>
<td>AreaCoulomb</td>
<td>5.4484e-14</td>
</tr>
</tbody>
</table>

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 recapitulates 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. **AreaCoulomb**: 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

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

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'.

```matlab
>> 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
```
Functions

List of Available Functions

This section lists the available functions.

[Images of function files]
Scripts

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