

Osort Tutorial

Automatic spike detection and sorting

Ueli Rutishauser
California Institute of Technology

<urut@caltech.edu>

Spike sorting

- Steps in spike sorting:

(1) Read raw data

(2) Detect spikes, extract their raw waveform (**detection**)

(3) Find the peak of the waveform, align (**alignment**)

(4) Determine to which cluster the spike belongs (**sorting**)

OSort

- Osort is an implementation of a template based, unsupervised online spike sorting algorithm.
- Online: sorting is done spike-by-spike. As soon as a spike is detected, it is sorted. Thus, this algorithm can be used for realtime processing of data.
- Unsupervised: The algorithm and its implementation can be run automatically on large data sets. Figures (png) of all clusters are automatically created for later visual inspection.
- Supervised: After the algorithm finishes, human operator needs to decide which clusters are valid, which are artificial splits (need to be merged), which are SUA/MUA.

Supported methods

- Apart from the core sorting algorithm (our own), Osort implements several other methods for spike detection and alignment (see references for details).
- Detection:
 - energy threshold: based on a local energy signal, similar to a TEO operator (teager energy operator)
 - CDW: spike detection based on a discrete wavelet decomposition
- Alignment:
 - MTEO alignment: multiscale teager energy operator

Supported raw data

- Neuralynx binary files (Ncs). Both Analog Cheetah and Digital Cheetah Variants. Sampling rates are fixed: 25000 Hz (Analog Cheetah Files) and 32556 (Digital Cheetah).
- Text files. Variable sampling rate, but $> 20\text{kHz}$ is recommended. Text files are slow for large data amounts.
- Other formats can be incorporated easily:
 - i) convert them to Ncs (freely available matlab functions for writing Ncs files are available).
 - ii) modify the code to directly read your format. Modify the functions `getRawData.m` / `getRawTimestamps.m`
 - iii) convert to textfile (see `readme.txt` for format description).

The Graphical User Interface - Overview

Paths & File Formats

File Details

Data In/Out

Raw data path Channels to process data in format analog cheetah digital cheetah text

Data out path Ground channels sampling rate

Timespamps file

Which Figures to produce

Figure properties

make figures of clusters figures path figure format:

display figures (off=only exported them) Block Nr(s) for raw fig:

make spike extraction figure figure label prefix

Spike detection

Processing Methods/Parameters

Detection

execute spike detection method amplitude thresholding positive amplitude thresholding negative amplitude thresholding absolute power signaling thresholding wavelet

extraction threshold e.g. 5

kernel size

Peak alignment

Peak finding

method none findPeak peak of power MTEO peak

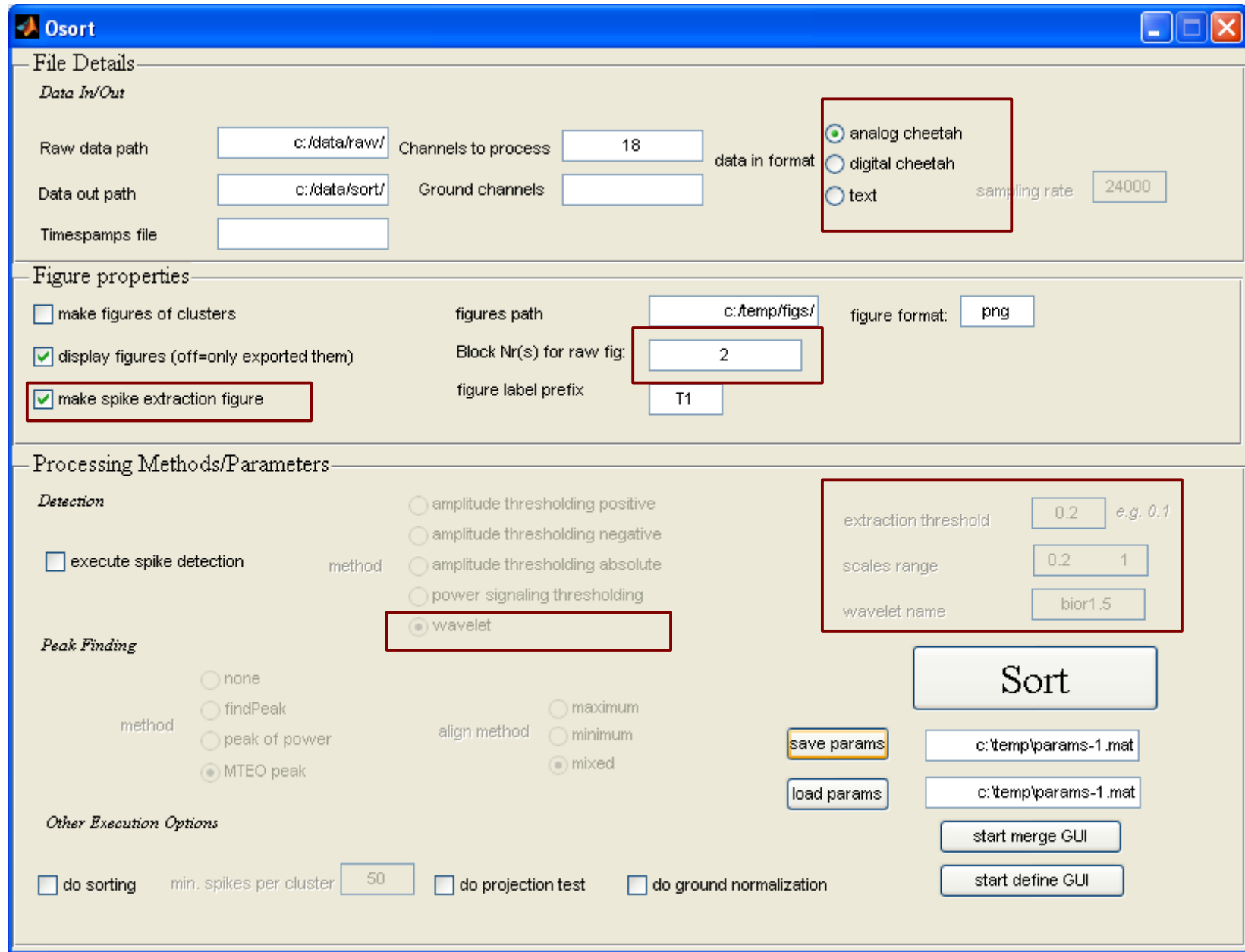
align method maximum minimum mixed

Other Execution Options

do sorting min. spikes per cluster do projection test do ground normalization

(see the readme.txt file for a description of each field & suggested default values).

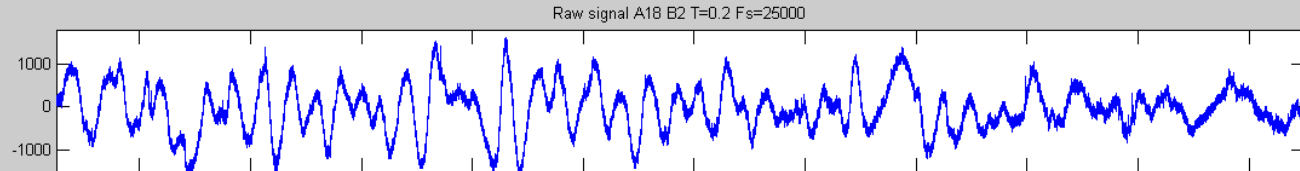
Step-by-Step: 1) plot raw data for inspection (GUI)



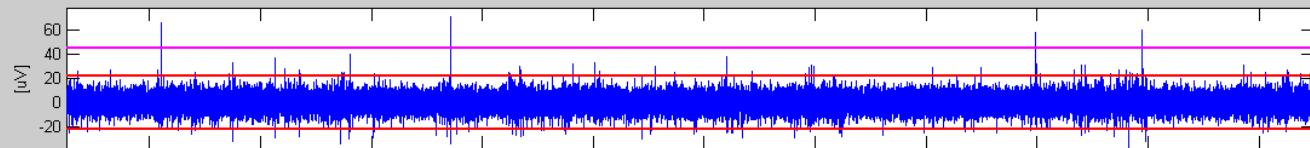
Produce a figure illustrating the spike extraction process of channel 18, second block (of 20s). Detecting spikes using wavelet with 0.2 as threshold, 0.2-1 as scales and the bior1.5 wavelet. These parameters are used even if “execute spike detection” is disabled. Enable/disable the option to set the parameters if you only want to produce the plot and not extract spikes of the entire data file.

Step-by-Step: 1) plot raw data for inspection (result)

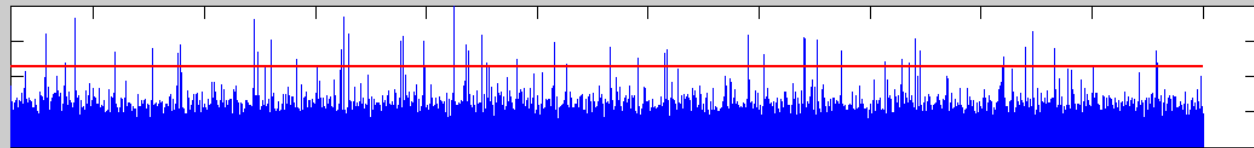
Raw signal



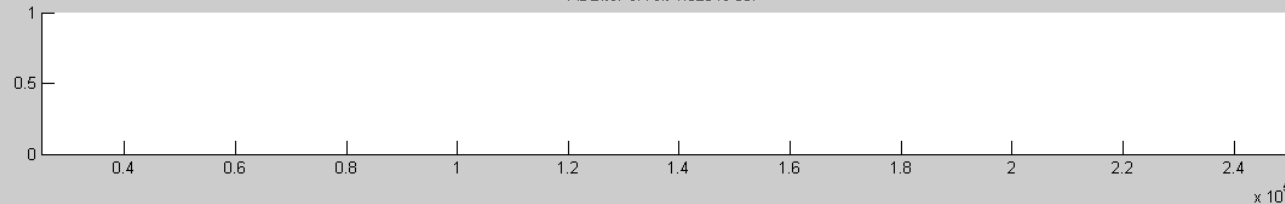
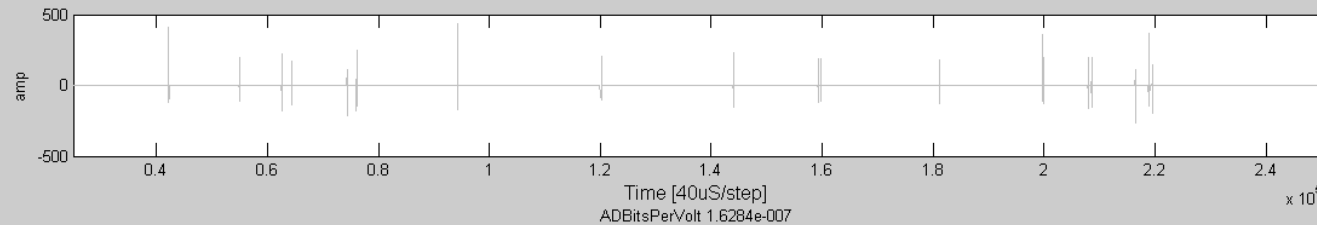
Bandpass filtered



Energy



Extracted spikes



Step-by-Step: 2) detect spikes & sort (GUI)

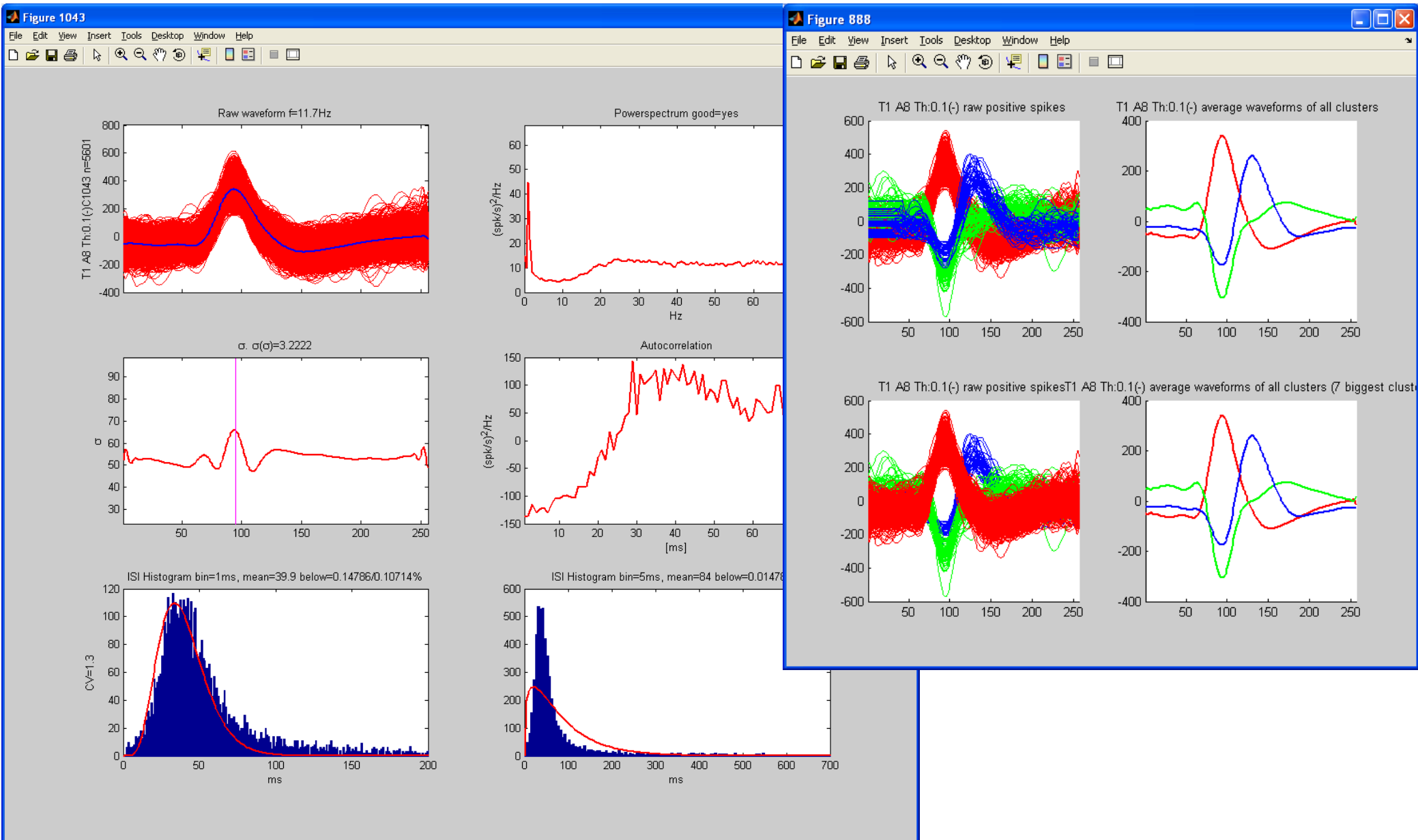
The screenshot shows the Osort GUI with the following configuration:

- File Details:**
 - Raw data path: c:/data/raw2/
 - Data out path: c:/data/sort/
 - Channels to process: 20
 - Ground channels: (empty)
 - Timespamps file: (empty)
 - data in format: analog cheetah (selected)
 - digital cheetah (unselected)
 - text (unselected)
 - sampling rate: 24000
- Figure properties:**
 - make figures of clusters: (highlighted)
 - display figures (off=only exported them):
 - make spike extraction figure:
 - figures path: c:/temp/figs/
 - figure format: png
 - Block Nr(s) for raw fig: 2
 - figure label prefix: T1
- Processing Methods/Parameters:**
 - Detection:**
 - execute spike detection: (highlighted)
 - method: wavelet (selected)
 - amplitude thresholding positive (unselected)
 - amplitude thresholding negative (unselected)
 - amplitude thresholding absolute (unselected)
 - power signaling thresholding (unselected)
 - extraction threshold: 0.1 (e.g. 0.1)
 - scales range: 0.2 to 1
 - wavelet name: bior1.5
 - Peak Finding:**
 - method: MTEO peak (selected)
 - align method: mixed (selected)
 - none (unselected)
 - findPeak (unselected)
 - peak of power (unselected)
 - maximum (unselected)
 - minimum (unselected)
 - Other Execution Options:**
 - do sorting: (highlighted)
 - min. spikes per cluster: 100 (highlighted)
 - do projection test:
 - do ground normalization:

Buttons: Sort, save params, load params, start merge GUI, start define GUI

Min nr clusters: clusters which have less than this number of spikes are considered “noise”. Lower if you have very sparsely firing neurons!

Step-by-Step: 2) detect spikes & figures of clusters (results)



Plot illustrating one particular cluster. Left column: raw waveforms, variance at each point, ISI (zoomed). Right column: powerspectrum, autocorrelation, ISI (0-700ms).

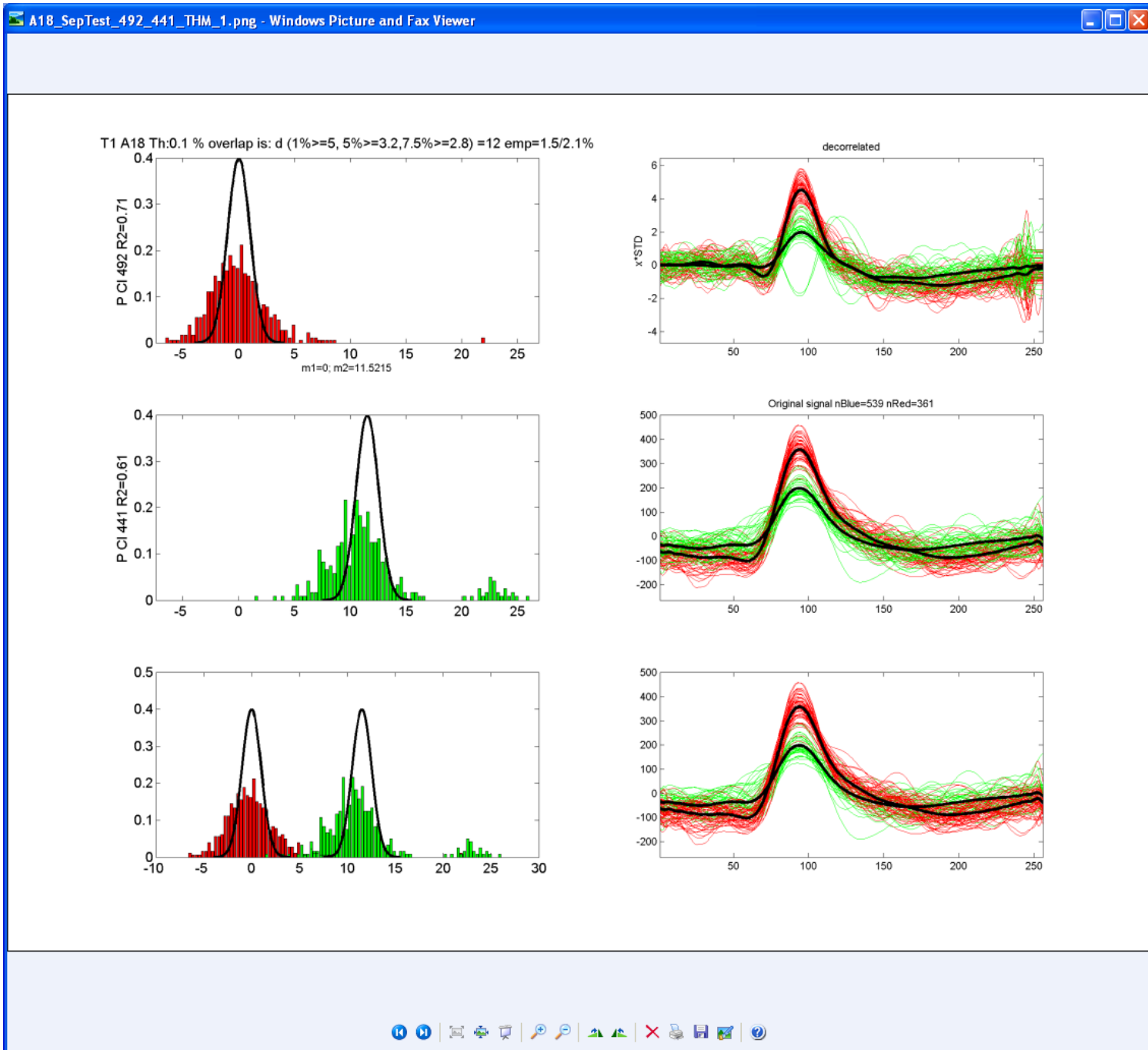
Step-by-Step: 3) projection test figures (GUI)

The screenshot shows the Osort GUI with the following configuration:

- File Details:**
 - Raw data path: c:/data/raw2/
 - Data out path: c:/data/sort/
 - Channels to process: 18
 - Ground channels: (empty)
 - data in format: analog cheetah (selected)
 - text (unselected)
 - digital cheetah (unselected)
 - sampling rate: 24000
 - Timesamps file: (empty)
- Figure properties:**
 - make figures of clusters:
 - display figures (off=only export them):
 - make spike extraction figure:
 - figures path: c:/temp/figs/
 - figure format: png
 - Block Nr(s) for raw fig: 2
 - figure label prefix: T1
- Processing Methods/Parameters:**
 - Detection:**
 - execute spike detection:
 - method:
 - amplitude thresholding positive (unselected)
 - amplitude thresholding negative (unselected)
 - amplitude thresholding absolute (unselected)
 - power signaling thresholding (unselected)
 - wavelet (selected)
 - extraction threshold: 0.1 (e.g. 0.1)
 - scales range: 0.2 1
 - wavelet name: bior1.5
 - Peak Finding:**
 - method:
 - none (unselected)
 - findPeak (unselected)
 - peak of power (unselected)
 - MTEO peak (selected)
 - align method:
 - maximum (unselected)
 - minimum (unselected)
 - mixed (selected)
 - Other Execution Options:**
 - do sorting: min. spikes per cluster: 50
 - do projection test: (highlighted with a red box)
 - do ground normalization:

Buttons on the right side: Sort, save params, load params, start merge GUI, start define GUI.

Step-by-Step: 3) projection test figures (result)



Example: how to process text file

The screenshot shows the Osort software interface with the following configuration:

- File Details**
 - Data In/Out*
 - Raw data path:
 - Data out path:
 - Timespamps file:
 - Channels to process:
 - Ground channels:
 - data in format: analog cheetah, digital cheetah, text
 - sampling rate:
- Figure properties**
 - make figures of clusters
 - display figures (off=only exported them)
 - make spike extraction figure
 - figures path:
 - figure format:
 - Block Nr(s) for raw fig:
 - figure label prefix:
- Processing Methods/Parameters**
 - Detection*
 - execute spike detection
 - method: amplitude thresholding positive, amplitude thresholding negative, amplitude thresholding absolute, power signaling thresholding, wavelet
 - extraction threshold: e.g. 5
 - kernel size:
 - Peak Finding*
 - method: none, findPeak, peak of power, MTEO peak
 - align method: maximum, minimum, mixed
 - Other Execution Options*
 - do sorting
 - min. spikes per cluster:
 - do projection test
 - do ground normalization

Buttons: (highlighted), (highlighted), , ,

A red box highlights the "data in format" and "sampling rate" options, with an arrow pointing to it from the right.

Note on automation

- Many files can be processed in a sequence: enter any number of channel numbers in “channels to process”
- Everything can be done in one go (select spike detection, sorting, make figures, make projection test) or selectively. Each of these steps can be executed independently, but some will only run if previous steps have been run already (i.e. Running sorting before spike detection won't work; if multiple options are selected, the order is automatically correct).
- Raw figures for several blocks can be generated: enter a list of block numbers in the “Block nr(s) for figs” field.

Filestructure and content

- Three main directories:
 - raw
 - figures (“figs” in the following)
 - data output (“sort” in the following)
- For each extraction threshold (field “extraction threshold”) Osort will create a subdirectory in figs and sort. Files will be stored in there. This is so that different thresholds/detection methods can be tried easily.
- For each channel & threshold, spike detection creates a file “Ax_sorted_new.mat”. This file contains the spike waveforms (variable: newSpikesNegative) the cluster number they are assigned to (variable: assignedNegative).
- There are two more GUIs: the merge GUI and the define usable cluster GUI. They are not covered here, but can be used to further (manually) refine the sorting (merge GUI) and to define what result should be used and the rest is assigned to noise.

References & Resources

- The spike sorting itself as well as the findPeak method of alignment:

Rutishauser, U., E.M. Schuman, A.N. Mamelak (2006). "Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo." J Neurosci Methods 154(1-2): 204-24.

- The wavelet based spiked detection method is described in:

Nenadic, Z. and J. W. Burdick (2005). "Spike detection using the continuous wavelet transform." IEEE Transactions on Biomedical Engineering 52(1): 74-87

- The MTEO peak finding method:

Choi, J. H., H. K. Jung, et al. (2006). "A new action potential detector using the MTEO and its effects on spike sorting systems at low signal-to-noise ratios." IEEE Transactions on Biomedical Engineering 53(4): 738-746.

- For an example of real data sorted using our method:

Rutishauser, U., A.N. Mamelak, E.M. Schuman (2006). "Single-trial learning of novel stimuli by individual neurons of the human hippocampus-amygdala complex." Neuron 49(6): 805-13.

Copyright

We make this source code freely available in the spirit of academic freedom and reproducibility of research. **However, it comes with no warranties whatsoever. Use at your own risk. Backup your data before you use it with this software.** We can't guarantee support. You are welcome to modify the code, but please send us updates if you fix a bug or add a useful feature. We might integrate it into the distribution.

OSort was initially written by Ueli Rutishauser (Caltech); The GUI was written by Matthew McKinley (MIT). All the code in 'code/3rdParty' was written by others and is included for convenience.

Copyright (c) by Ueli Rutishauser and the California Institute of Technology. This Research was conducted in collaboration with E.M. Schuman and A.N. Mamelak and funded by the Gimbel Discovery Fund as well as the Howard Hughes Medical Institute, through the laboratory of E.M. Schuman at Caltech.