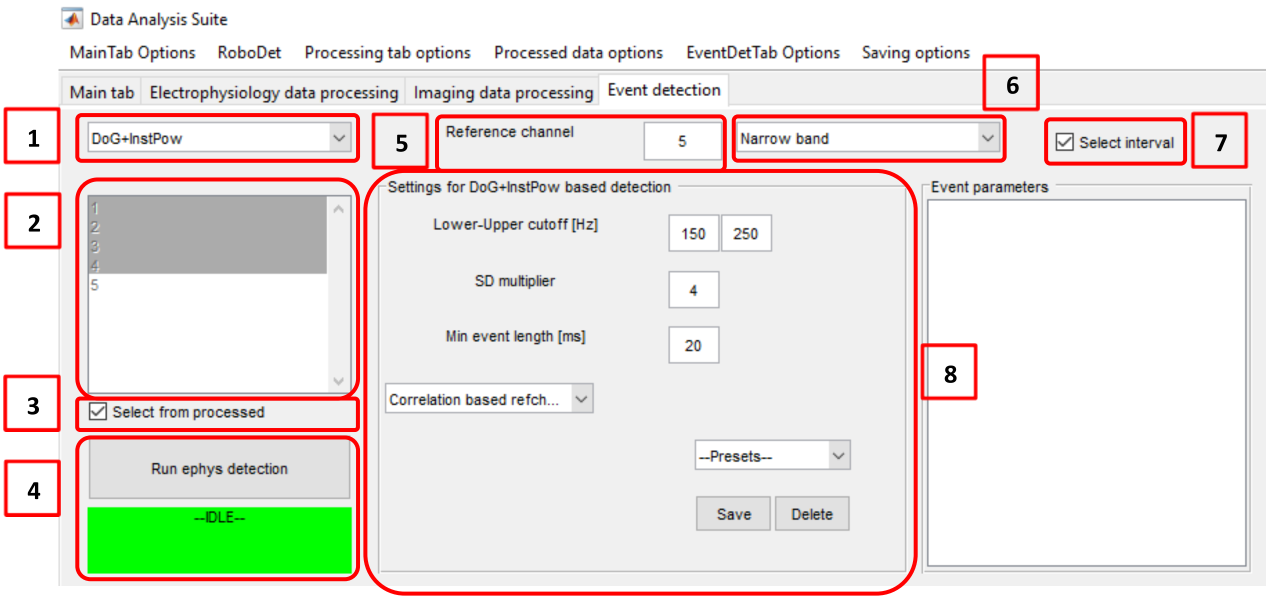
Benedek Szmola – December 2022

# Electrophysiology event detection algorithms of DAS

## Contents



## General detection setup UI in DAS




1. Drop down menu for selecting the electrophysiology (ephys) detection algorithm
2. Listbox for selection of channels to use in the detection
    - When 3 is enabled this element becomes inactive
3. Checkbox to indicate that you want to select previously processed channels for detection
    - These previously processed channels are the ones which can be found in the data processing tab
4. Button to launch the detection algorithm (upper) + status indicator (lower)
5. Entry field to specify the number of the reference channel(s)

- Reference channel(s) (here and in the whole document) means channels which should not contain the signals of interest (e.g. located in unrelated tissue)
- Accepted formats:
    i. Single channel: 5
    ii. Multiple channels method 1: 1,2,3,4,5
    iii. Multiple channels method 2: 1-5

6. Drop down menu for selecting a preprocessing method for the selected raw data
    - If 3 is selected this is overridden
7. Checkbox for discarding certain intervals
    - Discarding intervals means they will not be taken into consideration during the various computational steps during the detection
    - Discarded intervals will not be deleted from memory
8. Panel for setting up the selected detection algorithm
    - Specific settings will be explained separately for the algorithms in the following sections

Using the menu "EventDetTab Options" → "Show extra detection figures" additional visualization can be requested about the detection results or for some algorithms the step by step functioning.

## commDetAlg

This is a **comm**on **det**ection **alg**orithm used by multiple other detection algorithms.

- Looping over the provided channels
- Intervals above the provided threshold (computed by the calling detection algorithms) are extracted
    - Minimal above threshold duration here is computed as follows:
        - min(10 ms , minimum event length setting)
- If interval discarding was used
    - Every extracted interval is tested whether it overlaps with discarded segments
    - If the overlap is over 75% → discard the extracted interval
- Peak detection
    - Find the maximal value and its index in the extracted intervals
    - Which data transformation this peak detection is executed on depends on the caller detection script, see later
    - If reference channel validation was requested
        - Time stamp based
            - If reference channel was above threshold in the extracted interval, the interval is flagged

            - Correlation based
                - Correlation coefficient is computed between extracted interval and the analogue segment on the reference data, if it is above .5 → flag
- Intervals (putative events) flagged by reference validation can be manually inspected
    - The user can request the flagged events to be kept, otherwise they will be discarded
        - This is done via a separate popup GUI
- Event borders are extended towards the extending threshold if it was provided by the caller script
- When two subsequent events are separated by less than half the minimal event length setting they are merged
    - Separated here means difference between first event's ending border and the second event's starting border
- Using the specified minimal event length short events are discarded
- Outputs are the:
    - Peak indices
    - Border indices

## detParamMiner

This script is used by the detection algorithms to extract event parameters.

The parameters are the following (all of them describe the data within given event's borders):

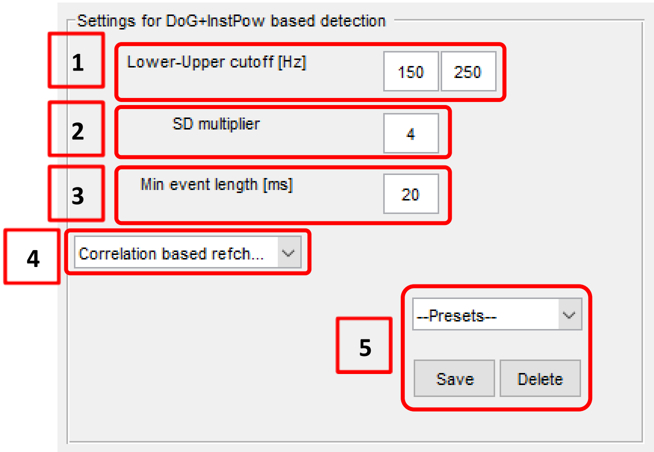
- RawAmplitudePeakT
    - Timestamp of the peak value (absolute value) on the raw data
- RawAmplitudeP2P
    - Peak to peak value on the raw data, i.e. difference between lowest and highest value
- BpAmplitudePeakT
    - Analogue to above, but on bandpass filtered data
- BpAmplitudeP2P
    - Analogue to above, but on bandpass filtered data
- Length
    - Duration of the event in seconds (ending border – starting border)
- Frequency
    - Frequency at which the event's wavelet transform showed the highest energy

- NumCycles
  - Number of cycles in the event's duration
  - Computed by counting the number of peaks in the interval
- AUC
  - Area under the curve
  - Computed by the numerical integral of the event
- RiseTime
  - Time from start of the event to its peak
- RiseTime2080
  - Time from 20% to 80% of the peak value (before peak)
- DecayTime
  - Time from event peak to event end
- DecayTime8020
  - Time from 80% to 20% of the peak value (after peak)
- DecayTau
  - Exponent of exponential fit to the 80%-20% portion of the event
- FWHM
  - Full width at half maximum (peak value / 2)
- NumInComplex
  - Place of the event in a complex
  - Complex means multiple events following each other within a given separation
    - It is hardcoded at the moment at 100 ms
    - Can be changed in the detParamMiner script, variable is called "maxSepInComplex"
  - The script also returns a separate variable which stores information about every event complex
- NumSimultEvents
  - Number of coinciding events from the other data type (talking about ephys this means coinciding imaging events)

Benedek Szmola – December 2022

# DoG+InstPow

## Setup UI




1. Setting the frequency range of interest (specify in [Hz])
2. Threshold standard deviation multiplier
3. Minimal length that an event should have (specify in [ms])
4. Drop down menu for selecting how the preliminary events should be tested against the reference channel
5. Preset tool: here you can call up previously saved presets, save the current settings or delete a preset

These inputs will be later referred to using the following format: setting#1 (the frequency range)

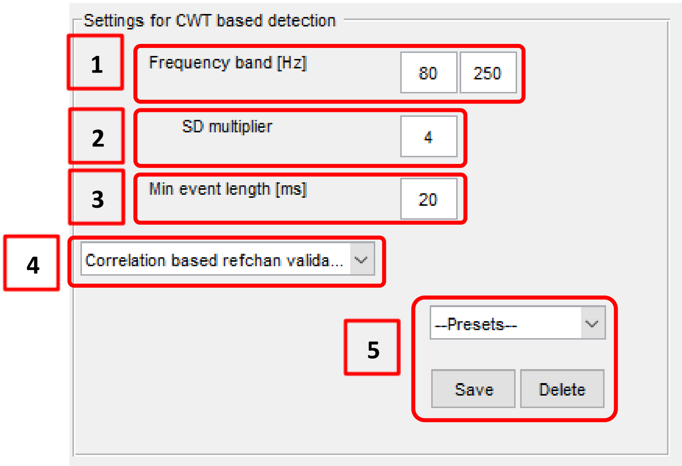
## Algorithm description

- Utilized data transformations:
    - Difference of Gaussians (DoG) bandpass filtering
    - Instantaneous power (InstPow)
    - Both are from BuzsákiLab (https://github.com/buzsakilab/buzcode)
- DoG and InstPow are computed for all channels
    - For the actual detection mainly InstPow is used
- If interval discarding was selected, first the undesired segments are removed from the InstPow
- Selection of baseline segments
    - Compute a threshold on the InstPow according to:
        - median(InstPow) + std(InstPow)
    - Every segment below this threshold is considered baseline (referred to as "quiet" in the code)
- Detection threshold is computed using the baseline segments

    - median(InstPow_baseline) + setting#2 * std(Instpow_baseline)
- A further threshold is computed for extending the borders of the detected events
    - median(InstPow_baseline) + std(InstPow_baseline)
- Common detection algorithm is called
    - InstPow is fed in as the detection data
- Event parameter computing algorithm is called

## CWT based

### Setup UI




1. Setting the frequency range of interest (specify in [Hz])
2. Threshold standard deviation multiplier
3. Minimal length that an event should have (specify in [ms])
4. Drop down menu for selecting how the preliminary events should be tested against the reference channel
5. Preset tool: here you can call up previously saved presets, save the current settings or delete a preset

These inputs will be later referred to using the following format: setting#1 (the frequency range)

### Algorithm description

- Utilized data transformations
  - Wavelet transform instantaneous energy (InstE)
    - Compute continuous wavelet transform (cwt)
    - At each time index take the numerical integral over the desired frequencies → InstE
  - DoG is also used, not directly for detection
- If interval discarding was selected, first the undesired segments are removed from the InstE
- Selection of baseline segments
  - Compute a threshold on the InstE according to:
    - median(InstE) + std(InstE)
  - Every segment below this threshold is considered baseline (referred to as “quiet” in the code)
- Detection threshold is computed using the baseline segments

- median(InstE _baseline) + setting#2 * std(InstE _baseline)
- A further threshold is computed for extending the borders of the detected events
  - median(InstE _baseline) + std(InstE _baseline)
- Common detection algorithm is called
  - InstE is fed in as the detection data
- Event parameter computing algorithm is called

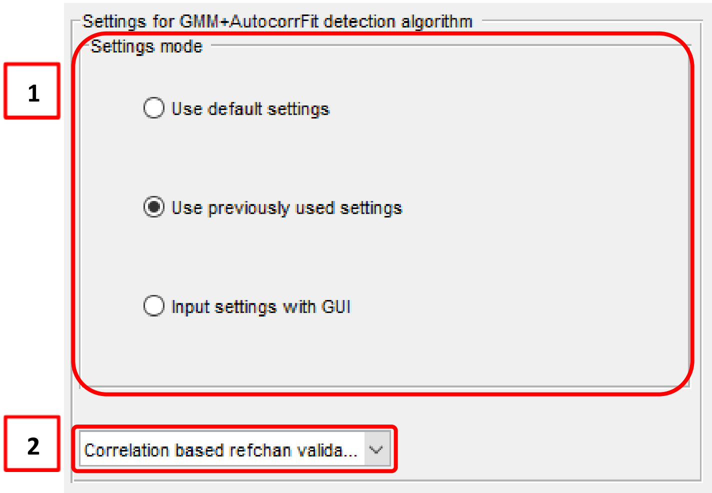
Benedek Szmola – December 2022

# GMM+Autocorrfit

### General overview

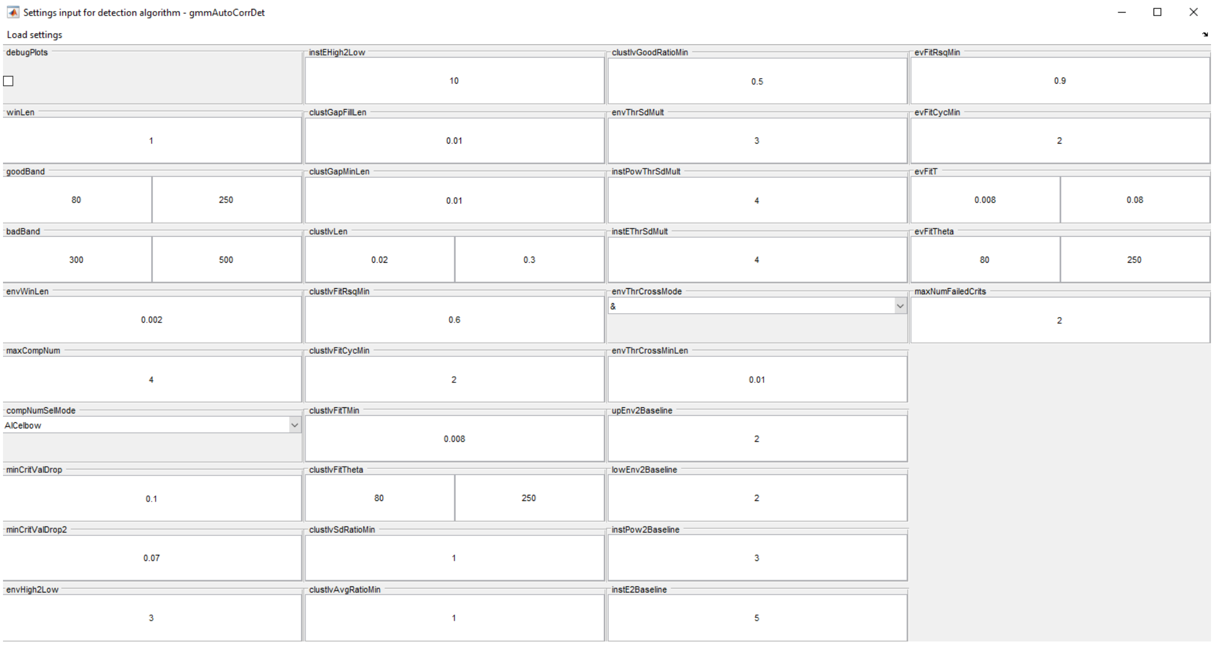
This algorithm utilizes Gaussian mixture modelling (GMM) to separate baseline activity from events using a sliding, non-overlapping window. After the clustering step various criteria are evaluated, prominent among them the curve fitting to the interval's autocorrelogram. Based on the results the intervals are classified as events and others. Inspiration from [Tsakanikas, P., Sigalas, C., Rigas, P. *et al.* High-Throughput Analysis of *in-vitro* LFP Electrophysiological Signals: A validated workflow/software package. *Sci Rep* **7**, 3055 (2017). https://doi.org/10.1038/s41598-017-03269-9].

### Setup UI in DAS




1. Selection panel for how you want to specify the algorithm settings
2. Drop down menu for selecting how the preliminary events should be tested against the reference channel

## Separate setup UI




The settings for this algorithm will be explained in the description, with the setting names highlighted with **bold font**. While configuring the settings in the window depicted above, a tooltip is available for each field by hovering the cursor over it.

## Algorithm description

- Utilized data transformations
  - DoG
  - Upper and lower envelope from DoG
    - Peaks are extracted with MATLAB's findpeaks
      - MinPeakDistance parameter is set to **envWinLen**
        - Peaks closer than **envWinLen** are disregarded
    - Peaks are interpolated to get the envelope using "makima" function
  - InstPow
    - Computed separately for frequency band of interest (**goodBand**) and frequency band which is expected to contain confounding noise (**badBand**)
  - InstE
    - Computed separately for frequency band of interest (**goodBand**) and frequency band which is expected to contain confounding noise (**badBand**)
- Sliding along the time axis
  - Window size given by **winLen**
  - If interval discarding option was selected
    - Sliding occurs in the continuous non discarded intervals

  - There are some conditions which can alter the normal sliding behavior
    - When at the end of the time axis is closer than half of the window size (**winLen**), the window gets extended to the end
    - If in the last 50 ms of the window there is above threshold activity → window gets extended by 100 ms
      - Thresholding: median + std
      - The next iteration takes this shift into consideration such that the non-overlapping nature is preserved
- Data matrix is prepared for GMM, columns as follows:
  - DoG upper envelope, DoG lower envelope, InstPow (frequencies of interest), InstPow (frequencies of noise), InstE (frequencies of interest), InstE (frequencies of noise)
- GMM is called
  - Fit GMMs with MATLAB's "fitgmdist" using different number of components according to **maxCompNum**
  - Each GMM is stored, they are evaluated according to **compNumSelMode**
    - These refer to different criterions: silhouette, BIC, AIC
  - Using the criterion, the best GMM is selected
    - It also has to pass an additional test which checks by how much the criterion value dropped (**minCritValDrop**)
  - If a 1 component GMM was chosen there is an extra round of checks to make sure some activity was not missed
    - If 3 tests are passed the 2 component GMM is selected
      - Ratio of variances between the low and high cluster's upper envelope (**envHigh2Low**)
      - Ratio of variances between the low and high cluster's InstE envelope (**instEHigh2Low**)
      - Criterion drop  larger than **minCritValDrop2**
  - The low cluster is determined as the cluster with the lowest mu (mean) values
- If the chosen GMM has more than 1 component another script is called which evaluates various tests in order to determine whether there are any events of interest in the given window
  - 4 thresholds are computed
    - upThr: mean(low cluster upper envelope) + **envThrSdMult** * std(low cluster upper envelope)
    - lowThr: mean(low cluster lower envelope) - **envThrSdMult** * std(low cluster lower envelope)
    - iPThr: mean(low cluster InstPow) + **instPowThrSdMult** * std(low cluster InstPow)
    - instEThr: mean(low cluster InstE) + **instEThrSdMult** * std(low cluster InstE)

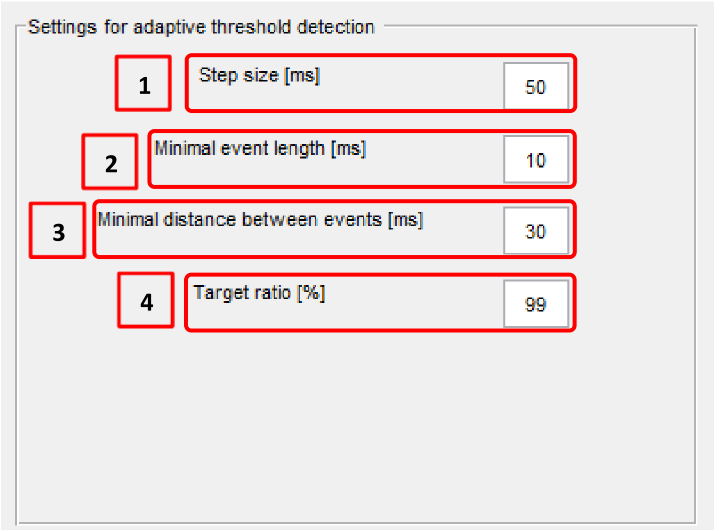
- Extract the clustered intervals
  - Gaps in these intervals smaller than **clustGapFillLen** are filled
  - Only intervals which lie in the range given by **clustIvLen** are further processed
- If the interval is not above the upper or lower envelope thresholds for long enough (**envThrCrossMinLen** / 2) → skipped
- Autocorrelation is computed for the interval in DoG form
- An exponentially decaying cosine function is fitted to the autocorrelogram
  - 4 criteria are computed
    - $R^2$ < **clustIvFitRsqMin**
    - Number of cycles until the .1 level is reached < **clustIvFitCycMin**
    - Time until .1 level is reached < **clustIvFitTMin**
    - Theta of fitted cosine outside range of **clustIvFitTheta**
  - If more than 2 are true → interval skipped
  - If more than 0 is true → further investigation
    - Interval is segmented at gaps longer than **clustGapMinLen**
    - Same procedure is repeated
- Std and mean of InstE is compared between frequency bands of interest and noise (**clustIvAvgRatioMin** & **clustIvSdRatioMin**)
  - If fail → interval skipped
- If ratio of interval where bad frequency InstE is higher than good frequency InstE > **clustIvGoodRatioMin** → interval skipped
- If bad InstE has higher AUC than good InstE (ratio > .75) → interval skipped
- Narrow down the cluster interval
  - Compute sub intervals of interest separately using the previously computer thresholds (upThr, lowThr, iPThr, instEThr) and the specified minimal event length (**envThrCrossMinLen**)
  - Unify the sub intervals (potential events)
  - First test for the potential events
    - Ratio between event and the low cluster (i.e. baseline activity) for upper, lower envelope, InstPow, InstE
    - Compare with (**upEnv2Baseline, lowEnv2Baseline, instPow2Baseline, instE2Baseline**)
    - If it fails more than 2 → skip
  - Once again fit the decaying cosine
    - Same comparisons as above, but here with: **evFitT, evFitCycMin, evFitTheta, evFitRsqMin**)
    - The events which pass all the tests get classified as confident events

        - The events which pass less than **maxNumFailedCrits** are classified as dubious events (later these can be confirmed manually)
- The extracted events get extended using the two thresholds:
    - mean(low cluster upper envelope) + std(low cluster upper envelope)
    - mean(low cluster lower envelope) + std(low cluster lower envelope)
    - The threshold which gives the wider event is accepted
- The user has the option to review dubious events, and events which were discarded because of comparison with the reference data
- Events separated by less than **clustGapFillLen** are merged
- detParamMiner script is called

## Adaptive threshold

This algorithm has not been maintained for a longer time, so proceed with caution. It is based on: [Patel J. Using an Adaptive Thresholding Algorithm to Detect CA 1 Hippocampal Sharp Wave Ripples. 2013]

### Setup UI




1. Sliding window step size (specify in [ms])
2. Minimal event length (specify in [ms])
3. Minimal temporal separation between two events (specify in [ms])
4. Percent of time bins which should be below the iteratively updated threshold

These inputs will be later referred to using the following format: setting#1

### Algorithm description

- Utilized data transformations
  - DoG
  - DoG envelope computed with Hilbert transformation
- Z score the DoG
- Compute Hilbert transform from z-scored DoG
- Smooth it with a 20 ms Gaussian window
- Find peaks using MATLAB's "findpeaks"
- Create a histogram of the peaks using a sliding window
  - Window length given by setting#1 * 2
  - Step size = setting#1
- Compute sliding average with same windowing
- Iteratively raise threshold until setting#4 percent of data points are below it
- Extract above threshold events

## Abbreviations

- (G)UI: (Graphical) User Interface
- std: standard deviation
- ephys: electrophysiology
- DoG: Difference of Gaussians
- InstPow: Instantaneous Power
- InstE: Instantaneous Energy
- cwt: continuous wavelet transform
- GMM: Gaussian Mixture Modelling