

A vertical blue line of medium thickness, positioned to the left of the word 'SPYCODE'.

SPYCODE

Version 4 User Reference

INTRODUCTION

SpyCode is a custom software built in Matlab®, developed for processing multi-channel electrophysiological signals. The software provides a working environment able to perform efficient data management and processing of MEA signal, since it incorporates a very rich repertoire of standard and advanced signal analysis tools.

This User Guide presents the main functionalities of the software.

TABLE OF CONTENTS

DATA CONVERSION	1
Multiple Converter	1
HyBrainWare Spikes Conversion.....	2
PRE-PROCESSING	3
Filtering Data	3
SALPA and ARTIFACTS.....	3
SPIKE DETECTION	4
Threshold Computation	4
Spike Detection.....	5
Join and Split Recording Phases.....	6
PLOT TOOLS	7
Raw Data	7
Raster Plot	8
POST STIMULUS TIME HISTOGRAM	9
SPIKE ANALYSIS	10
BURST ANALYSIS	13
Burst Detection.....	13
Burstiness Index (BI).....	14
Burst Analysis.....	14
ADDITIONAL TOOLS	20
Avalanches Analysis GUI.....	20
Information Theory GUI.....	23
Shuffling Methods.....	23
MULTIPLE ANALYSIS	23

DATA CONVERSION

Multiple Converter

In SpyCode, data formats from Multi Channel Systems (MCD Files) and the Panasonic MED64 system (MED Files) are supported for conversion to the format recognized by Matlab (MAT Files). Data are converted to the internal format where single channels are stored in single files.

To start conversion, select the correspondent option on menu Data Conversion (Multiple MCD/MED Converter), and then select the folder which contains the files. Figure 1 shows a custom GUI opened for MCD file conversion. *Help* button contains additional information.

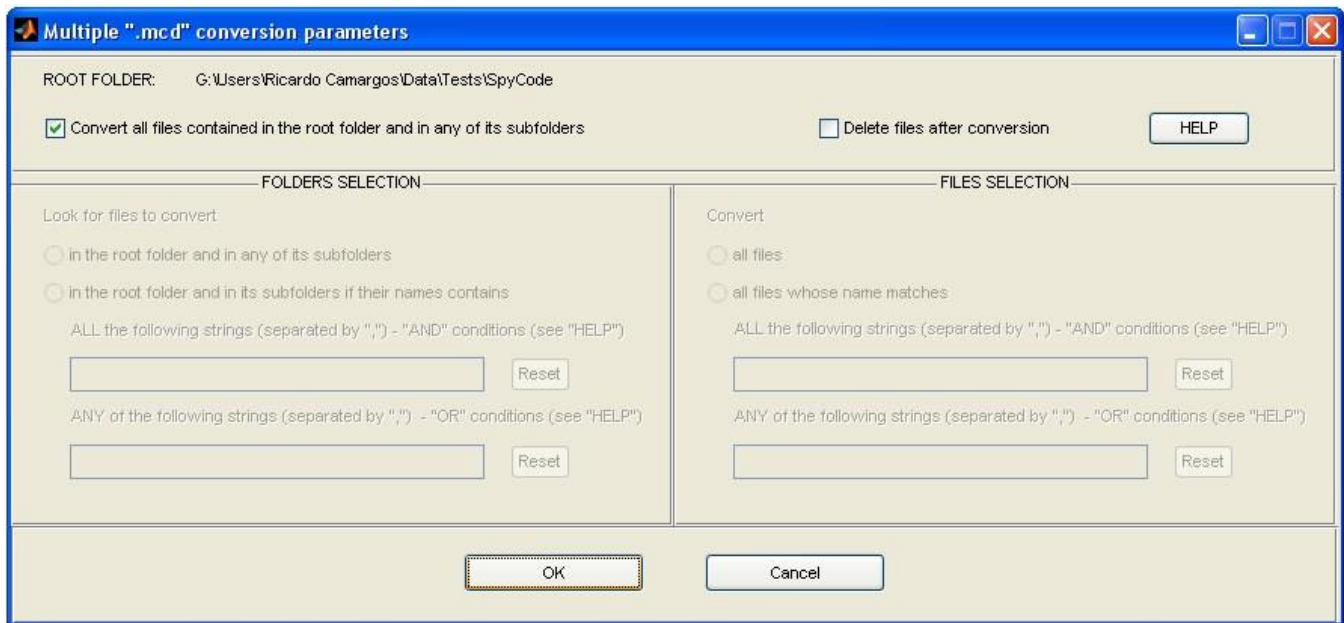


Figure 1. MCD Conversion

The window shown in Figure 2 will return all files recognized with the selected format. Select any file to determine specific properties, like experiment name, output folder or digital filter settings. To select all files, press the button Select All, and then click in the button OK.

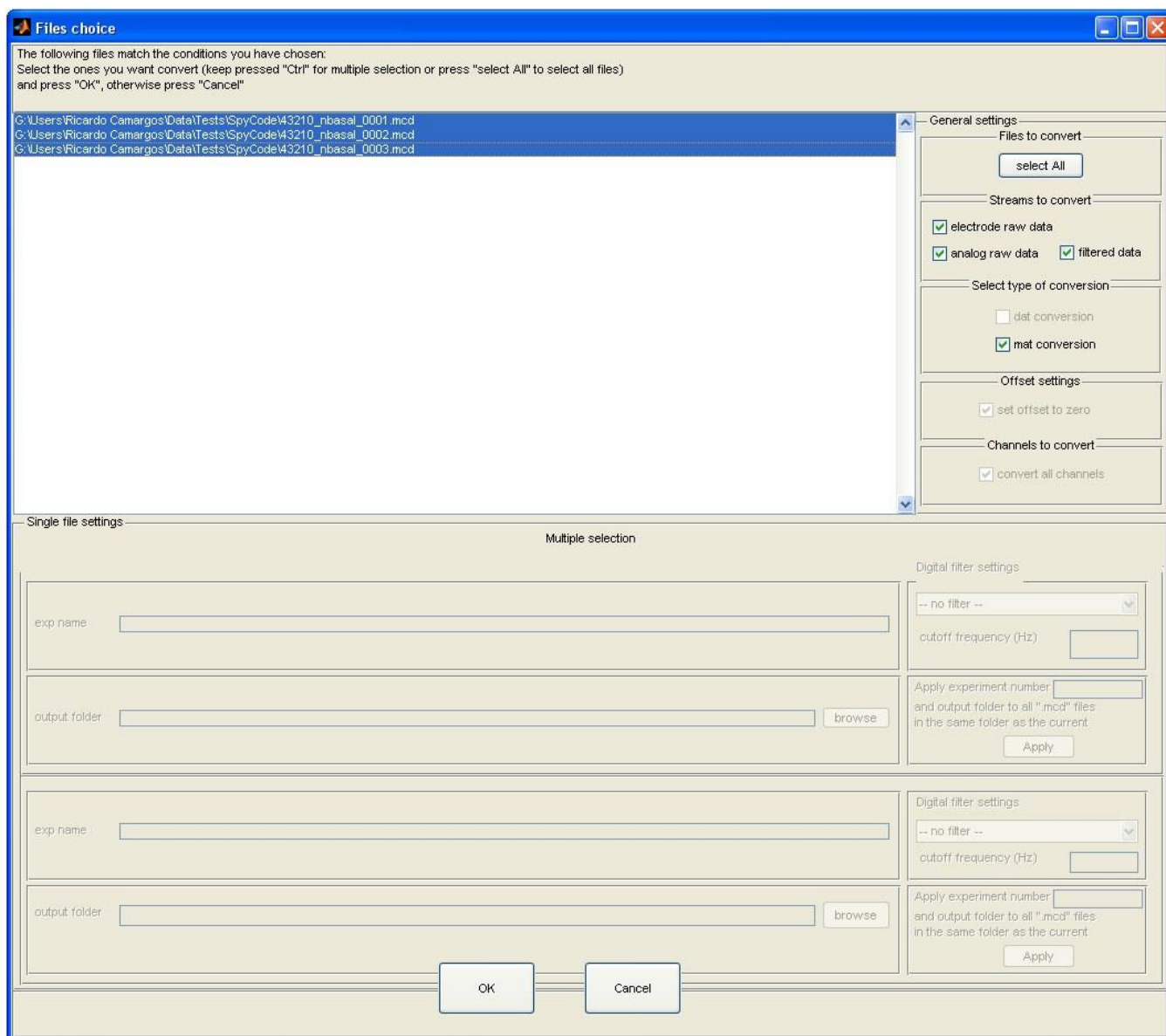


Figure 2. Files Choice Dialog

HyBrainWare Spikes Conversion

This tool, is used to extract the peak trains from the Files generated by HyBrainWare software, working similarly with the function described above.

PRE-PROCESSING

Filtering Data

After the conversion step, there is the possibility of filter the raw data, through band-pass filter, in which a Butterworth filtering is implemented.

In Pre-Processing menu, select the option *Filter Raw Data*, select the root folder and then in the options window (Figure 3) select the Bandwidth [Hz], the Sampling Rate [Hz] the Voltage Range [mV] selected in McRack, the Gain of the filter in use and its Resolution [bits]. The results will be saved in the folder *_FilteredData*.

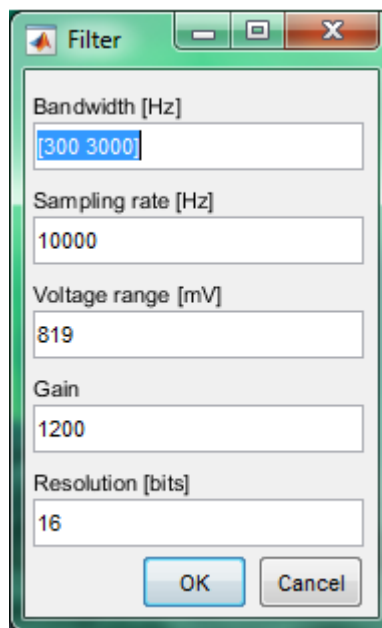


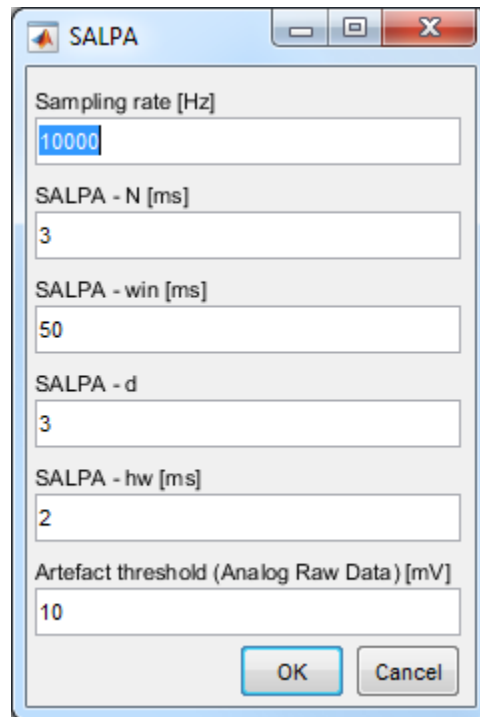
Figure 3. Filtering

SALPA and ARTIFACTS

The algorithm SALPA, described by Wagenaar & Potter (Wagenaar & Potter, 2002), is here implemented. A model of the artifact based on locally fitted cubic polynomials is subtracted from the recording, yielding a flat baseline amenable to spike detection by voltage thresholding.

In Pre-Processing menu, select the option *Suppress artifact (SALPA)*, select the root folder and then in the options window (Figure 4) select the Sampling Rate [Hz] the N value [ms] (For every timepoint n in the recording from a given electrode, a three degree polynomial is fitted to a segment of $2N+1$

samples centred around n. 3ms is the optimal value for N), the Window (win) [ms], the degree of the polynomial function (d), the Hardware Blanking, (hw) [ms] and the Artefact Threshold [mV].



The image shows a Windows-style dialog box titled "SALPA". It contains several input fields with the following labels and values:

- Sampling rate [Hz]: 10000
- SALPA - N [ms]: 3
- SALPA - win [ms]: 50
- SALPA - d: 3
- SALPA - hw [ms]: 2
- Artefact threshold (Analog Raw Data) [mV]: 10

At the bottom right, there are two buttons: "OK" and "Cancel".

SPIKE DETECTION

Threshold Computation

The threshold used for the spike detection procedure is calculated as a multiple of the standard deviation of the raw data noise. In order to do that, a period of the recording in which no spike or burst are present must be selected appropriately.

In menu *Detection > Compute Threshold*, it's available a custom GUI (Figure 4), which shows the traces of every channel. Single channels can be explored individually and the chunks of recordings to be used can be selected independently.

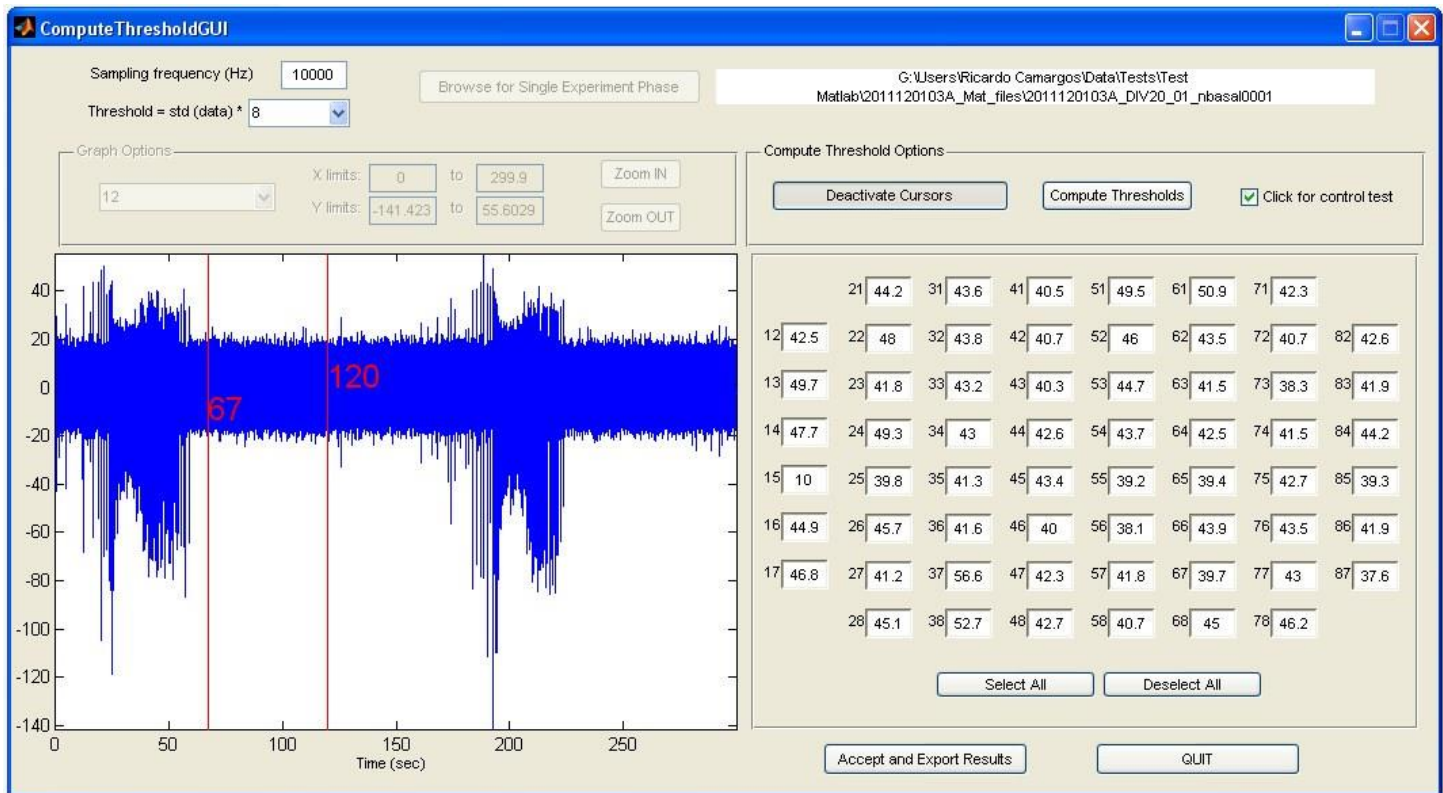


Figure 4. Compute Threshold GUI

Determine the parameter *Sampling Frequency* and the standard deviation multiplier, which can be changed empirically or can be let as is suggested.

In panel *Graph Options* there are graph tools that refer to the analyzed signal. Select any channel in the listbox to show its signal, and then adjust the axis limits to better visualize it.

In panel *Compute Threshold Options*, click on the button *Activate Cursors*, then select two points delimiting where there isn't any peak and then click in the button *Compute Thresholds*. The values will appear in all textboxes near each channel number. If there is one or more values in red color, select all values except the ones in red (Click in button *Select All* and then click in all red values to deselect them). Choose in the listbox a channel that corresponds to one of the red values, and then repeat the process until all values are written in black color.

Spike Detection

Two different algorithms are implemented to detect spikes, both based on the computation of a differential threshold:

The first one, SDDT (Spike Detection Differential Threshold), splits up the signal in consecutive windows fitted to hold at most one single spike and assign a spike whenever the absolute Max-Min difference within a window overtakes the threshold.

The second one, PTSD (Precision Timing Spike Detection), basically looks for all Relative Maximum/Minimum (RMM) within the signal and evaluates spikes firstly by comparing RMMs position in terms of spike expected duration and refractory period and finally by excluding underthreshold outputs.

To make the process, select the algorithm in *Spike Detection* menu and insert the parameters to be used, shown in Figure 5, in which there are suggested values.

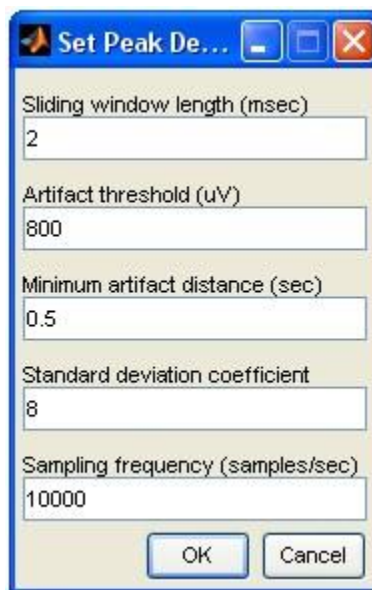


Figure 5. Set Peak Detection parameters

Join and Split Recording Phases

It's possible to split spike trains into arbitrarily long chunks and similarly to join different spike trains into a single one. This allows the user to perform analysis on very long experiments (whose raw data had been necessarily split because of PC's resources management) with no need of combining results from different chunks a posteriori.

To make the process of joining phases, select *Spike Detection > Join Recording Phases*. In the first textbox, set the name of the Peak Detection folder, in the second one write the name of each phase, as it is shown in the example {'PhaseName1'; 'PhaseName2'; 'PhaseName3'}, and then in the last one set the name of the sub-folders. If there isn't any sub-folder, write {'PeakDetectionMAT'}, as is shown in Figure 6.

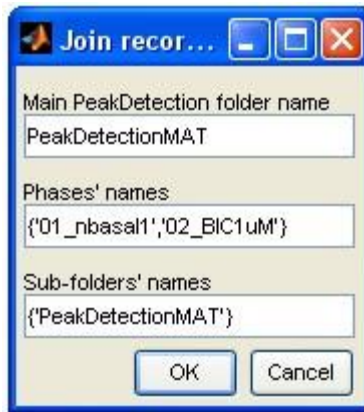


Figure 6. Join Recording Phases

PLOT TOOLS

Raw Data

There are some options to plot the results generated by SpyCode. It's possible to plot the raw data in menu *Plot>Raw Data* and select specific channels, as is shown in Figure 7.

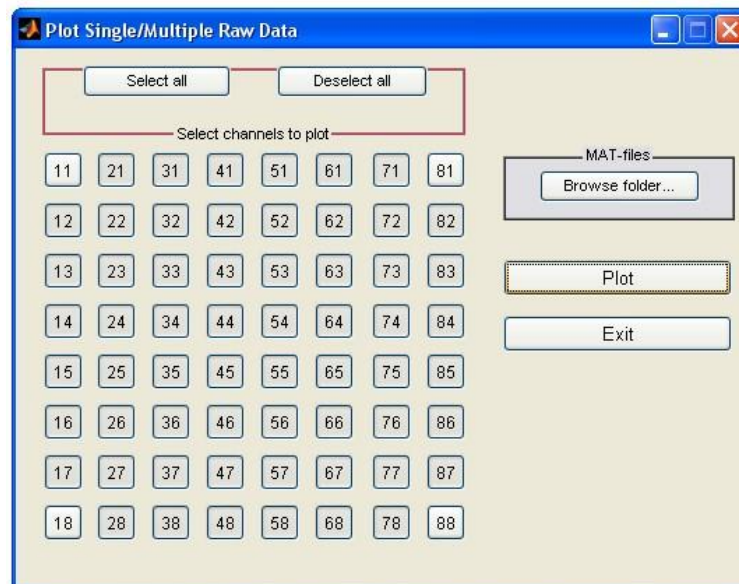


Figure 7. Plot Raw Data

The results are saved in JPEG format for each channel separately (as it's shown in the example in Figure 8a). Figure 8b shows the same raw data, but plotted with together with the corresponding peaks.

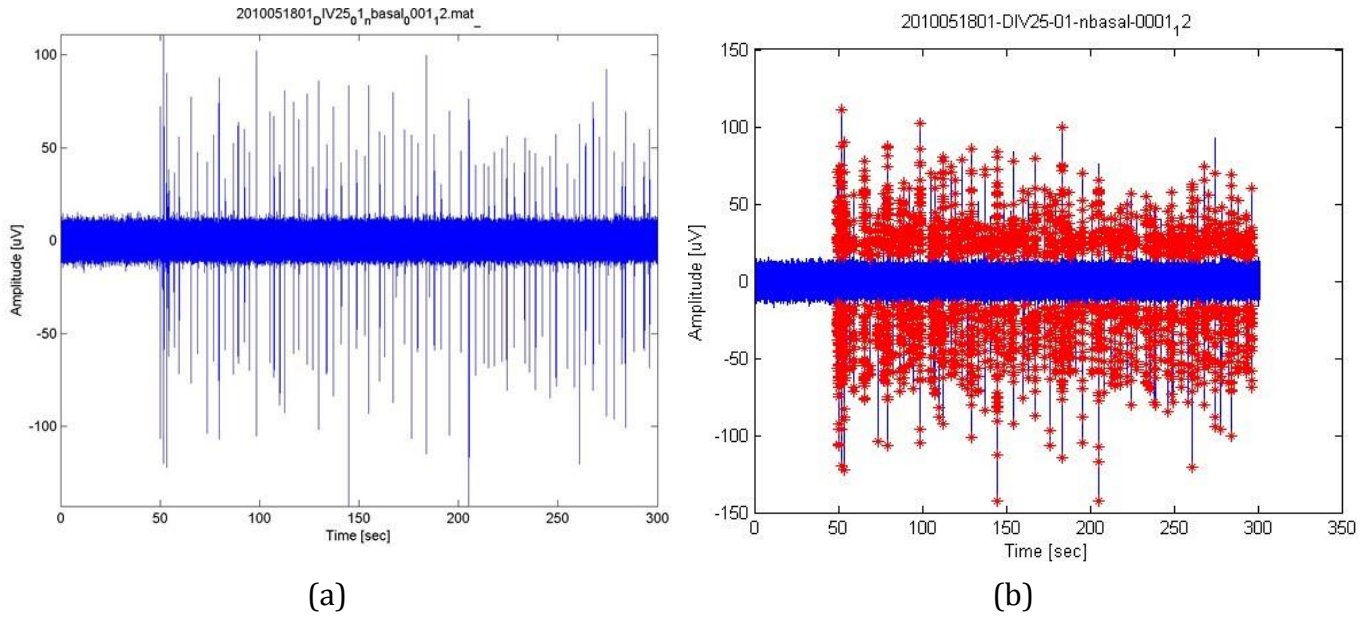
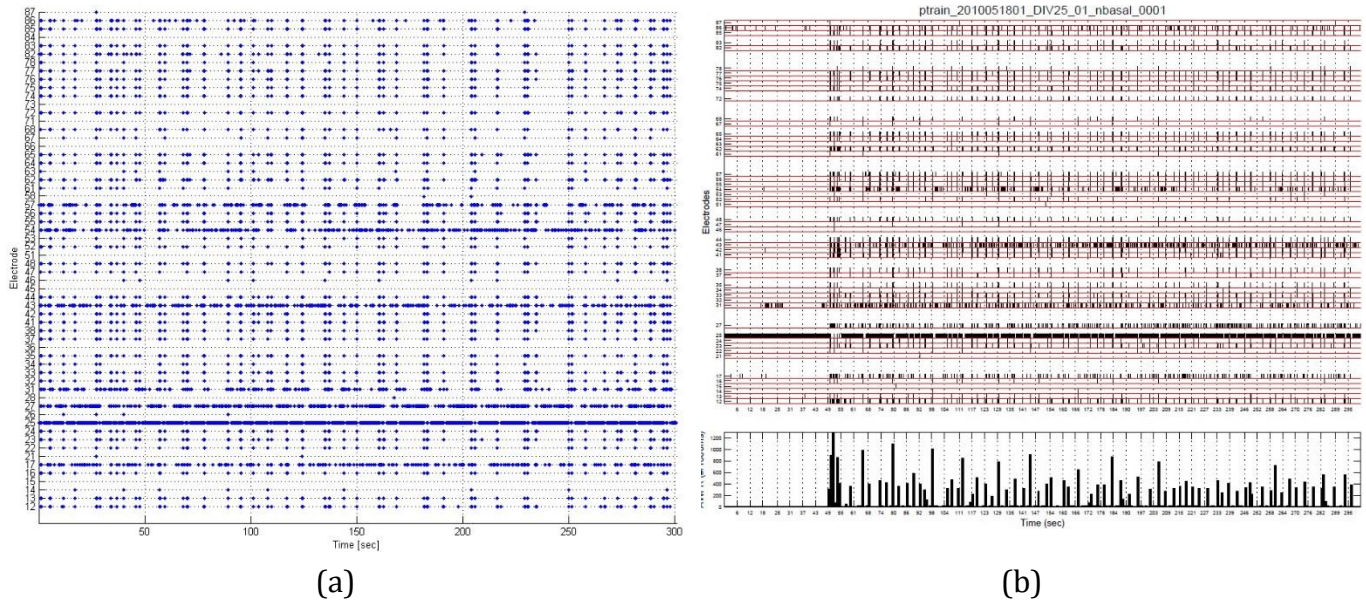
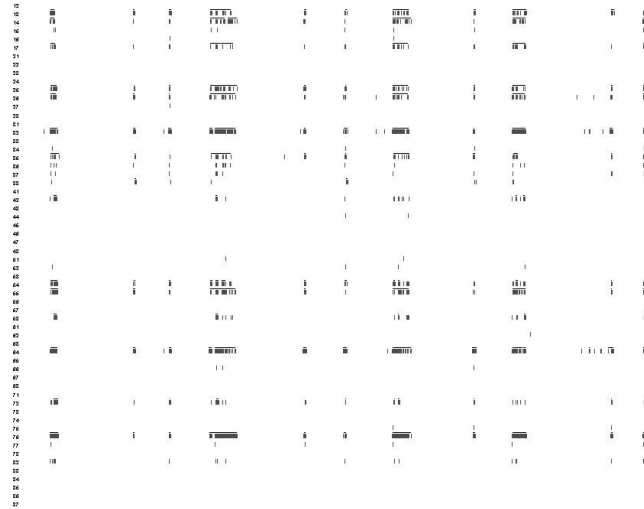


Figure 8. (a) Raw Data - one channel; (b) Raw Data with Peaks plotted

Raster Plot

Raster Plot is a dot matrix data structure that shows a visualization of the peak trains of each channel vs. time. It can be accessed in menu *Plot>Raster Plot*, in which its results are saved in the folder *_RasterPlotMAT* (Figure 9a). It's possible also to plot the Raster Plot with corresponding Average Firing Rate (Figure 9b), in menu *Raster Plot + AFR*, which results are saved in folder *_RasterPlotPdf*, and Burst Trains (Figure 9c), in menu *Plot Raster + Burst trains + Network Burst Train*.





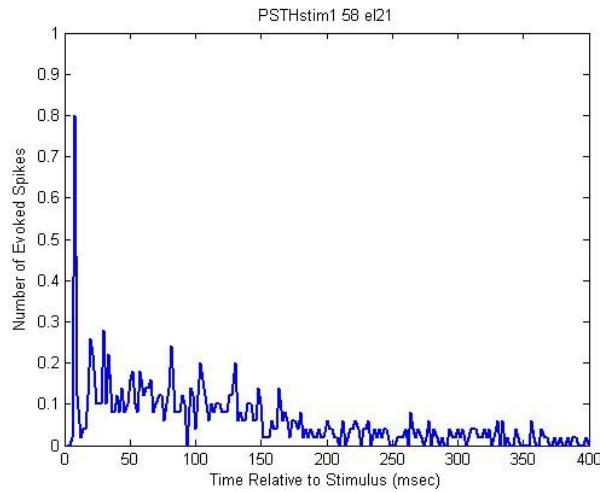
(c)

Figure 9. (a) Raster Plot of all channels; (b) Raster Plot with Average Firing Rate; (c) Burst Train Plot

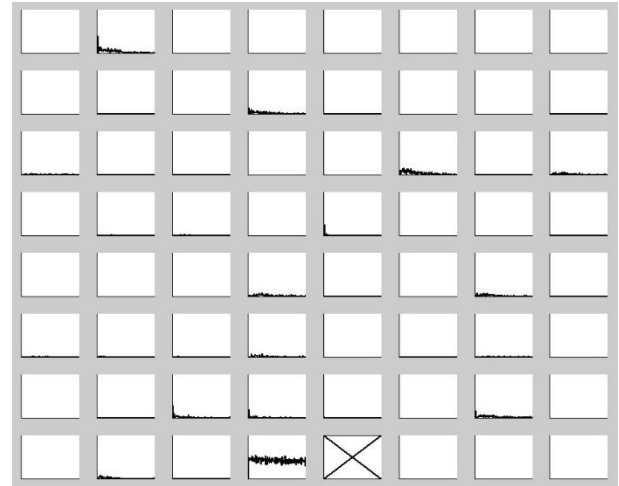
POST STIMULUS TIME HISTOGRAM

Post Stimulus Time Histogram (PSTH) allows computing the response to the stimuli (PSTH area and latency) of all channels by inserting the appropriate parameters for artifact blanking. The function computes the number of spikes in the considered window and once the blanking given to artifact has been performed. It also allows plotting the histograms of single channels individually or together with all the others (i.e., a multiple plot in which the PSTH of all the electrodes are plotted together). All the generated plots are saved in both the .fig and .jpg formats.

In menu Plot PSTH, it's possible to plot a single channel, all channels or an 8x8 map, as it is shown in Figure 10. The results are saved in the folder "..._PSTHresults".



(a)



(b)

Figure 10. Post Stimulus Time Histogram. (a) Single Channel Plot; (b) 8x8 plot

SPIKE ANALYSIS

The Spike Analysis allows the user to compute the Inter-spike interval (ISI), generated for each channel separately, with the selected parameters for the computation and plot the distribution of all the ISI histograms in a single plot. It will result an image that includes a graph and a spectrogram with the Firing Rates, as it is shown in the example of Figure 11, which signal is divided in two phases, and is saved in the folder *IFR*.

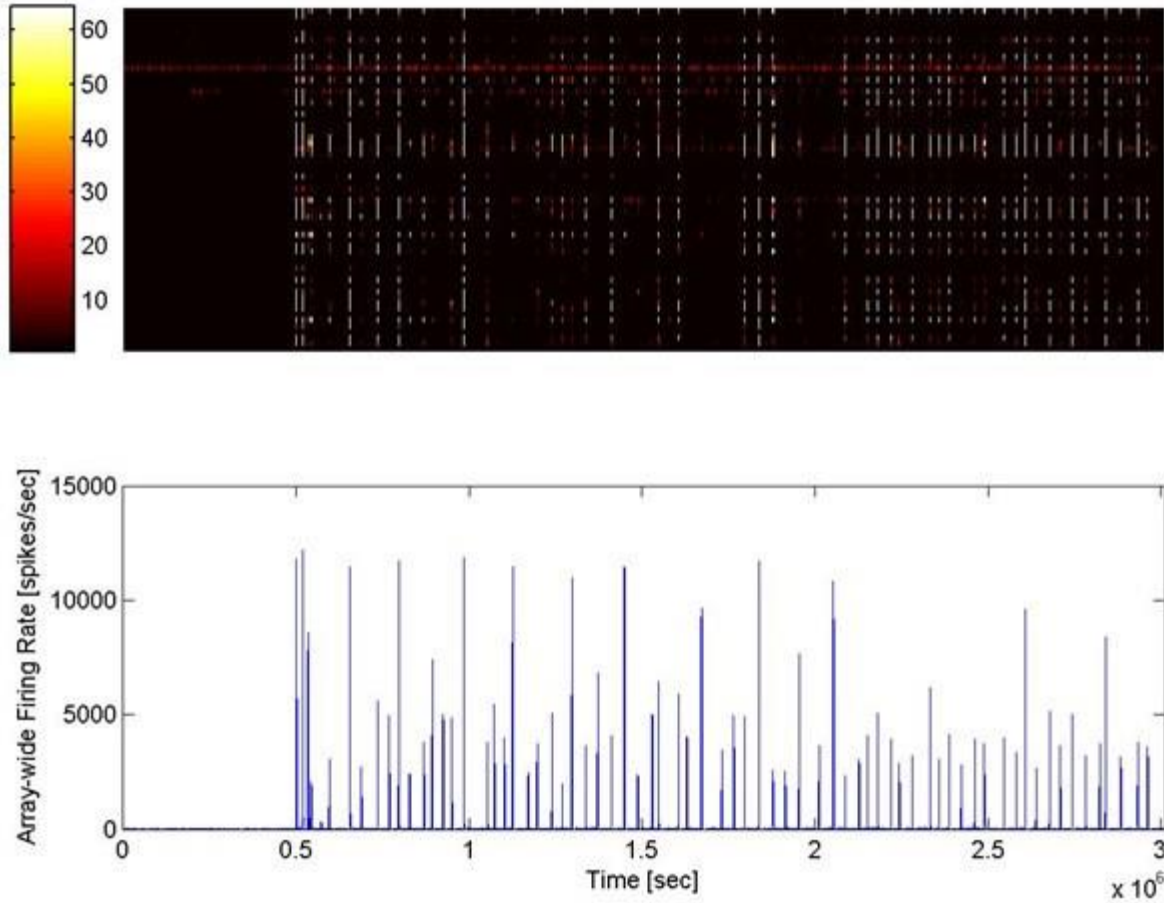
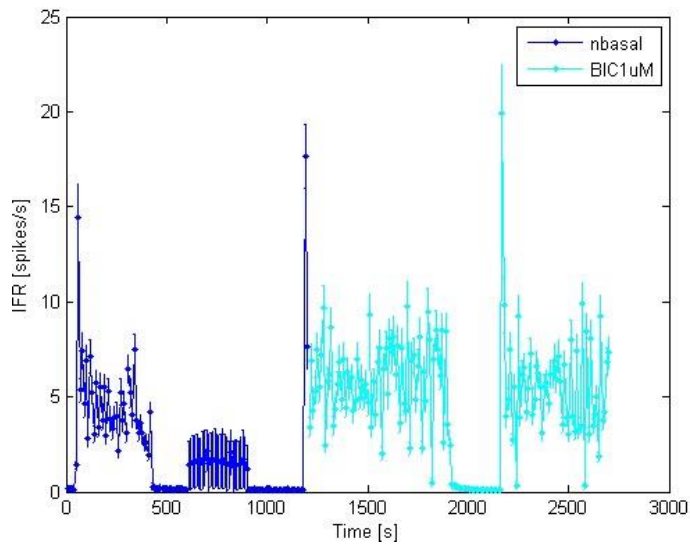
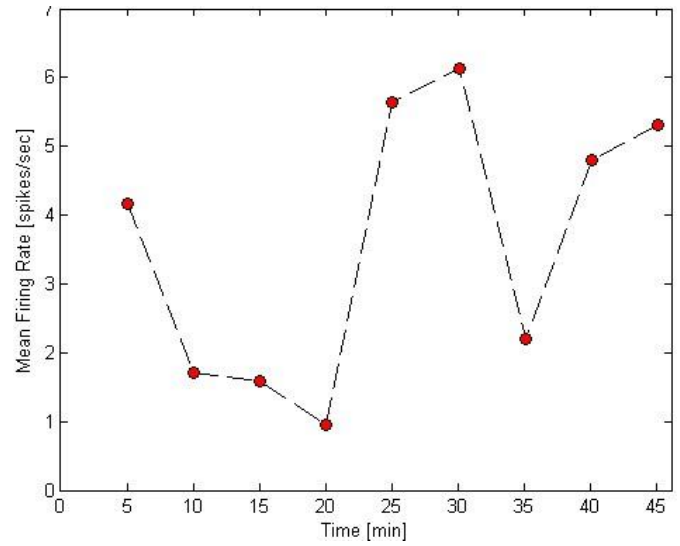


Figure 11. Instantaneous Firing Rate

The Average Firing Rate (Figure 12a) divides the signal in bin-size specified segments to calculate its values, which results are saved in folder *_SpikeAnalysis/_AFR*. The Mean Firing Rate, which is the number of spikes per second of each channel, is shown in Figure 12b. Its results are saved in folder *_SpikeAnalysis/_MeanFiringRate*.



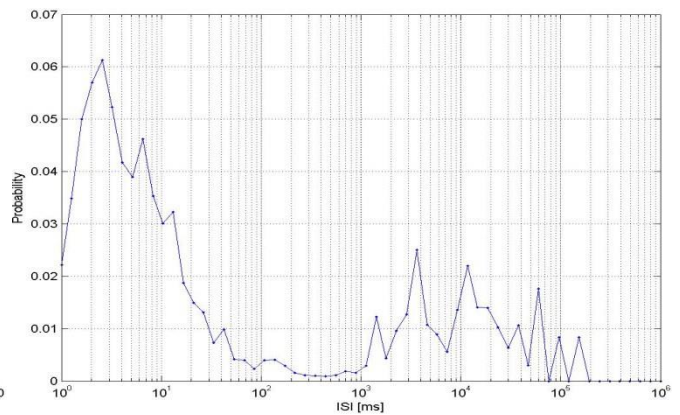
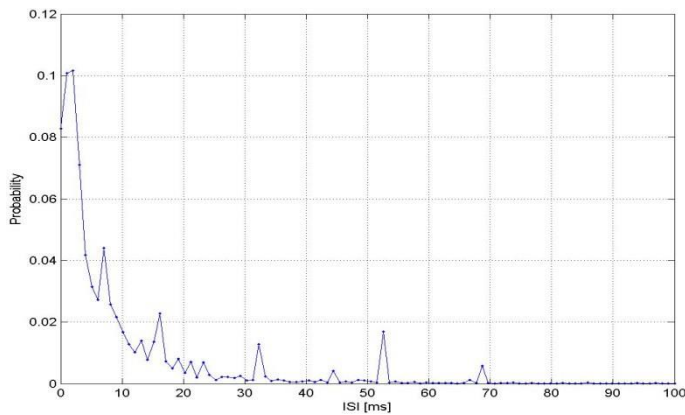
(a)



(b)

Figure 12. (a) Average Instantaneous Firing Rate; (b) Mean Firing Rate

In menu ISI Computation, it's possible to obtain a histogram of the inter spikes intervals. With this tool, the software can generate the histogram of probability \times ISI for each channel (Figure 13a), average of all channels (Figure 13b), smoothed histograms (Figure 13c) and statistical data in matlab files, saved in the folder *_SpikeAnalysis/..._AFR*.



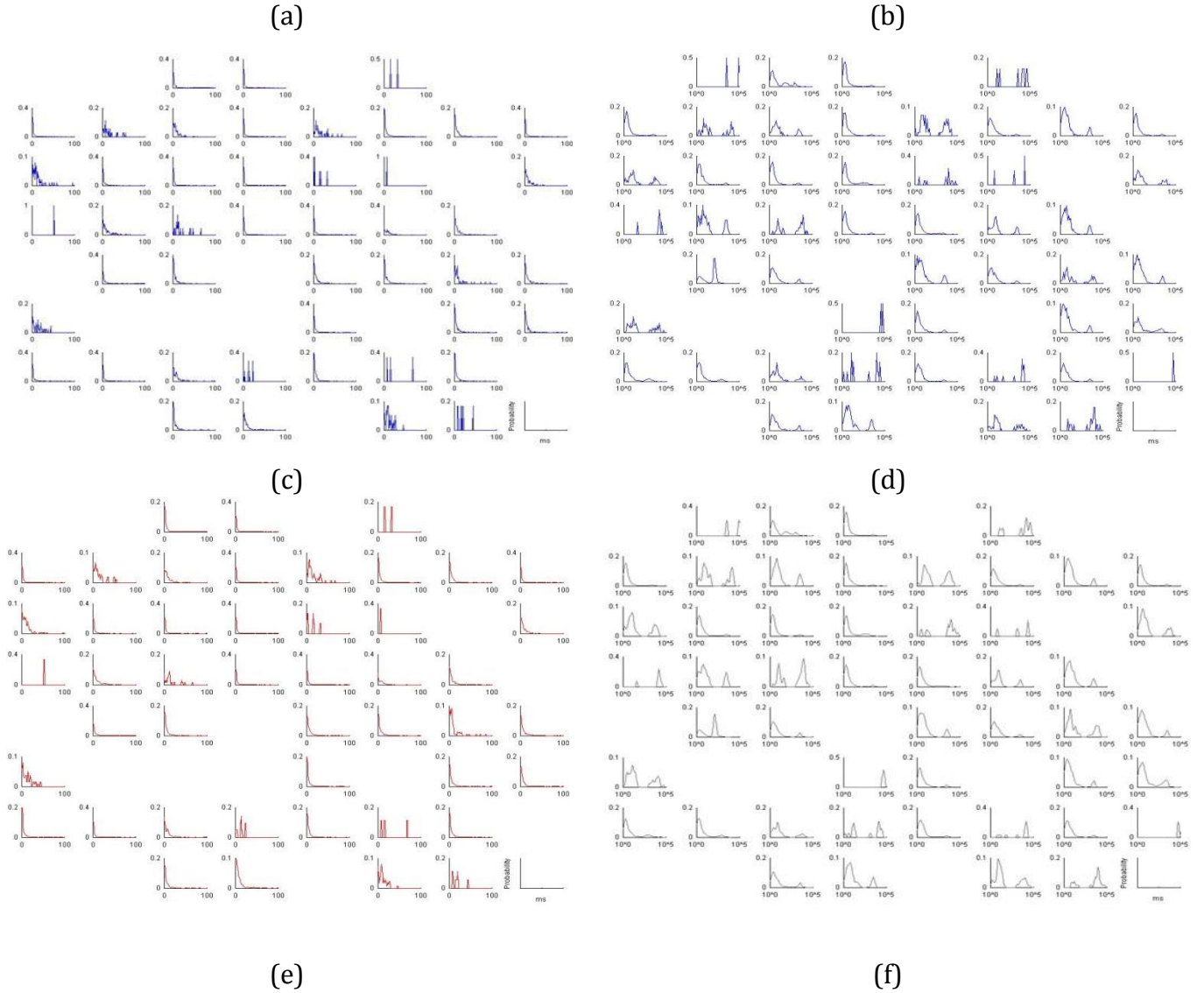


Figure 13. ISI Histograms: (a) Linear Network Average ISI; (b) Logarithmic Network Average ISI; (c) Linear 8x8; (d) Logarithmic 8x8 (e) Smoothed Linear 8x8; (f) Logarithmic Linear 8x8;

BURST ANALYSIS

To make the analysis of burst parameters, is necessary firstly make the detection of bursts, available in menu *Burst Analysis>Burst Detection*. There are two different algorithms to make it (CH and PA).

Burst Detection

The Burst Detection (BD) allows searching for bursts in the recorded activity by following two different approaches. The first one, Burst Detection v1-CH, allows to detect bursts as sequences of

at least N consecutive spikes (default value = 5) spaced less than a convenient time threshold (default value 100 ms).

The second one, Burst Detection v2-PA is a more reliable algorithm capable of detecting bursts and self-adapting to different experimental conditions. Basically, the new algorithm evaluates the logSIH in order to look for the best threshold (SIth) between intra-burst (i.e. within bursts) and inter-burst (i.e. between bursts and/or outside bursts) activity. If SIth is lower than 100 ms, the new BD uses the SIth value to determine the maximum ISI allowed within a burst. On the contrary, if SIth is higher than 100 ms, the new BD uses two different thresholds: the first one (i.e. 100 ms) is used for detecting burst cores, whereas the second one (i.e. SIth) is used to extend burst cores at the boundaries to include all spikes whose ISI is lower than SIth.

Burstiness Index (BI)

Here is implemented the algorithm as described by Wagenaar et al. (Wagenaar *et al.*, 2006), in which they counted the array-wide number of spikes in non-overlapping 1 s windows, and determined what fraction of the total number of spikes was contained in the 15% most active windows. Since bursts always occupied fewer than 10% of 1 s windows, this number, f_{15} , is close to one if most spikes occur in bursts. Conversely, if spikes are evenly spaced in time, f_{15} is close to 0.15. They then defined our burstiness index as $BI = (f_{15} - 0.15)/0.85$. Thus, BI is normalized between 0 (no bursts) and 1 (all spikes in bursts).

Burst Analysis

Once the burst detection has been performed, either following v1 or v2, it's possible to compute and save in .txt and .mat files the main statistics related to the burst analysis (e.g., number of bursts, mean bursting rate, mean frequency intra-burst, burst duration, etc.) by calling the functions *Statistics Report* or *Statistic Report Mean*. The results are saved in the folder *_BurstAnalysis* and subfolders *_MeanStatReportBURST*, *_MeanStatReportSPIKEinBURST* and *_StatisticsReport*, and are available in MAT and TXT Files.

Visual analysis can be generated by plotting a single channel Inter Burst Interval, available in Menu *Burst Analysis>Plot Single Channel IBI*. The resulting figure is saved in folder *_SingleChannelIBI* and is shown Figure 14.

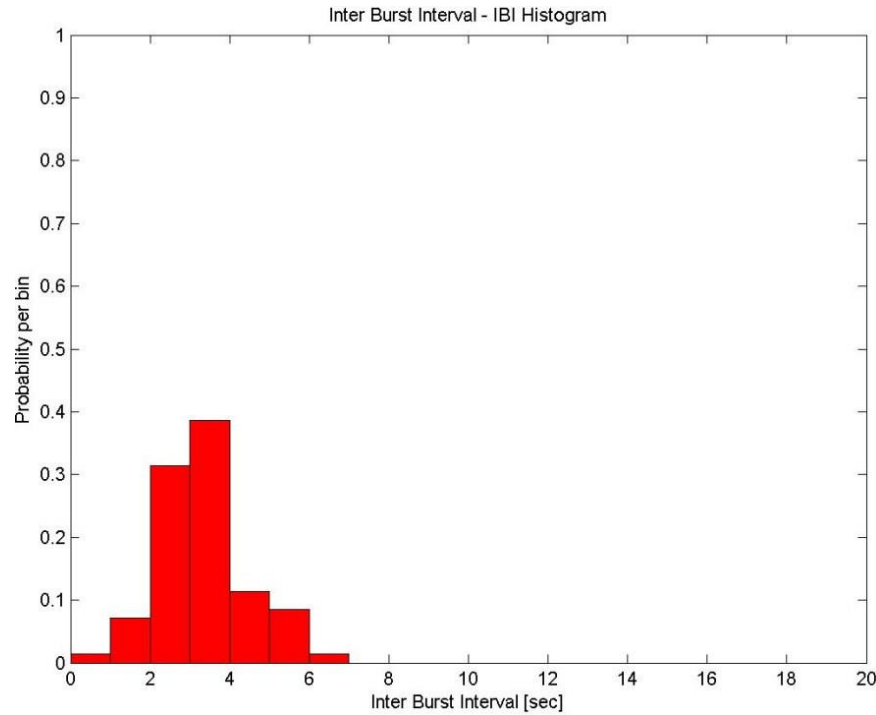
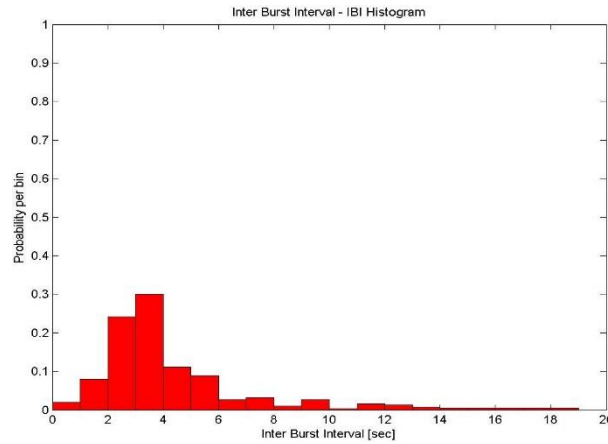


Figure 14. Single Channel IBI histogram

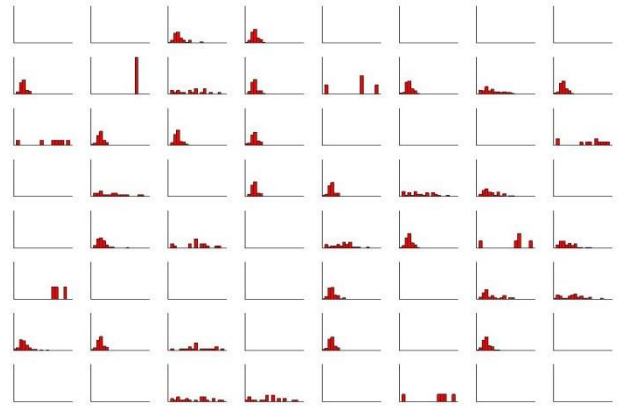
There is the possibility to detect network bursts from the burst trains (i.e. the output of the burst detection procedure). The algorithm considers the cumulative burst event train, namely the sequence of all the initial spikes of each single channel burst in the array and computes the logarithmic interburst event interval histogram (logIBeIH) by binning data in equally spaced logarithmic bins. In this way, it's possible to highlight the presence of two or more distinct peaks, the first one corresponding to short intervals within network bursts and the others to long intervals between network bursts. Hence, the same approach developed for the detection of spike bursts is applied to the NBD, starting from the cumulative burst event train: the only operation needed is setting two parameters, firstly the maximum inter-burst event interval for burst events within a network burst (maxIBeI) and secondly the minimum percentage of recording electrodes involved in a network burst (usually set at 20% the total number of active channels). The maxIBeI is set according to the IBeI threshold extracted from the logIBeIH, while the minimum percentage of recording electrodes is set by the user.

The Network Histogram is generated in menu *Burst Analysis>Plot IBI (network)*, resulting in the figure shown in Figure 15a. It is also possible to generate an array that shows all channels, in menu *Burst Analysis>Plot IBI (8x8 map)*, as is shown in Figure 15b.

The function “Network Burst Detection – van Pelt” is under revision and will be available in a future version of SpyCode.



(a)



(b)

Figure 15. (a) 8x8 IBI Histogram; (b) Network IBI Histogram

CROSS CORRELOGRAM (CC)

A dedicated GUI for cross correlation computation and analysis are available in SpyCode. Through the GUI, it's possible to select the time bin and the temporal window of the correlogram, together with different methods for the normalization of the correlogram itself. The possibility to define a threshold for the 'strong' connections is given in order to plot the connectivity maps also starting from the CC algorithm. The correlation can be calculated both in spike trains (i.e. the output of the spike detection procedure) and in burst event trains (i.e. the output of the burst detection algorithm).

The GUI is available in menu *Cross Correlogram*>*Cross Correlogram GUI*, as is shown in Figure 16.

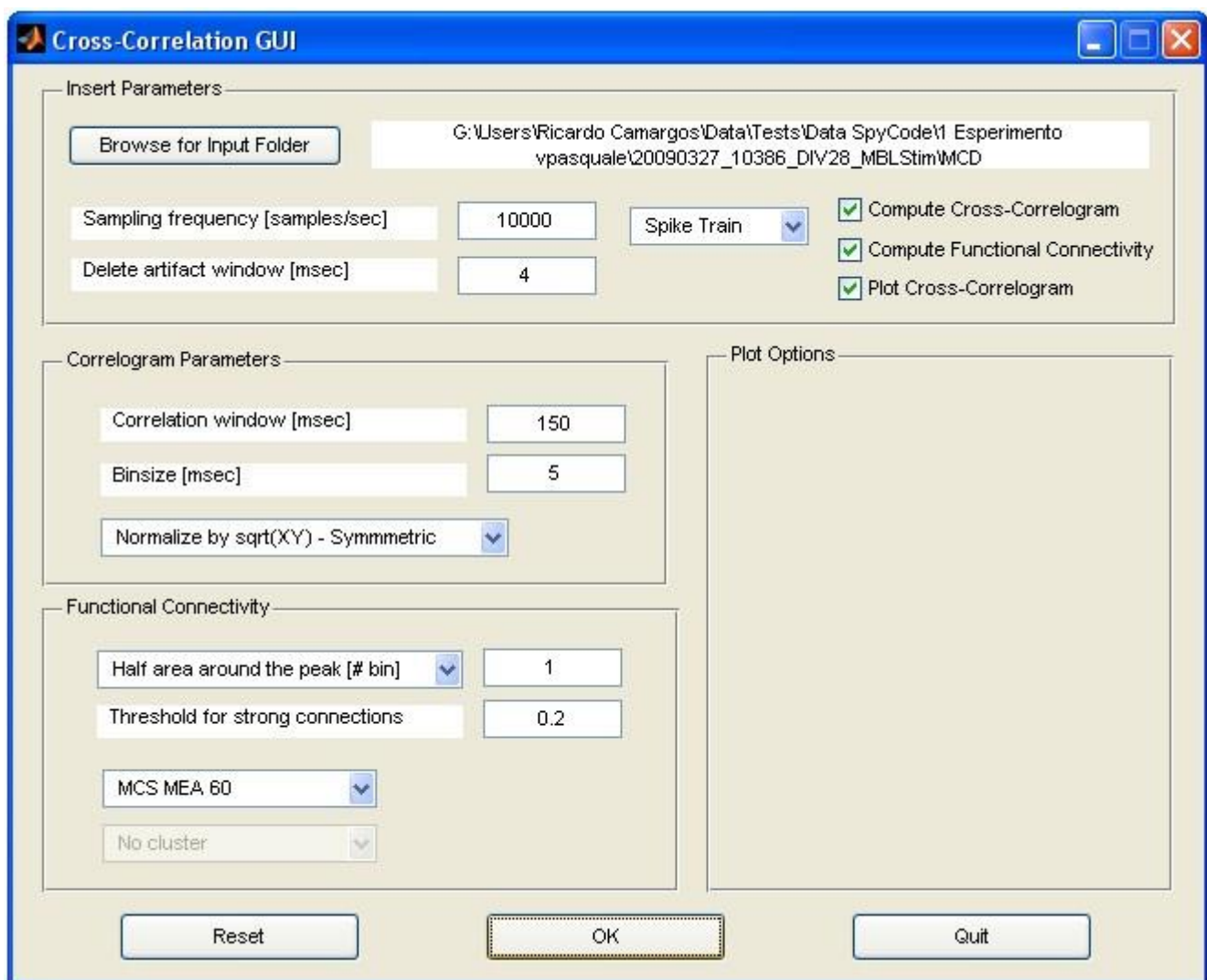


Figure 16. Cross-Correlation GUI

In GUI generated after CC computation, it can be plotted clicking in menu *Plot Correlogram*. The resulting figures are saved in the folder *_CCorr..._Correlogram*. Figure 17 shows an example of a correlogram of the channels 13 and 17 of an experiment.

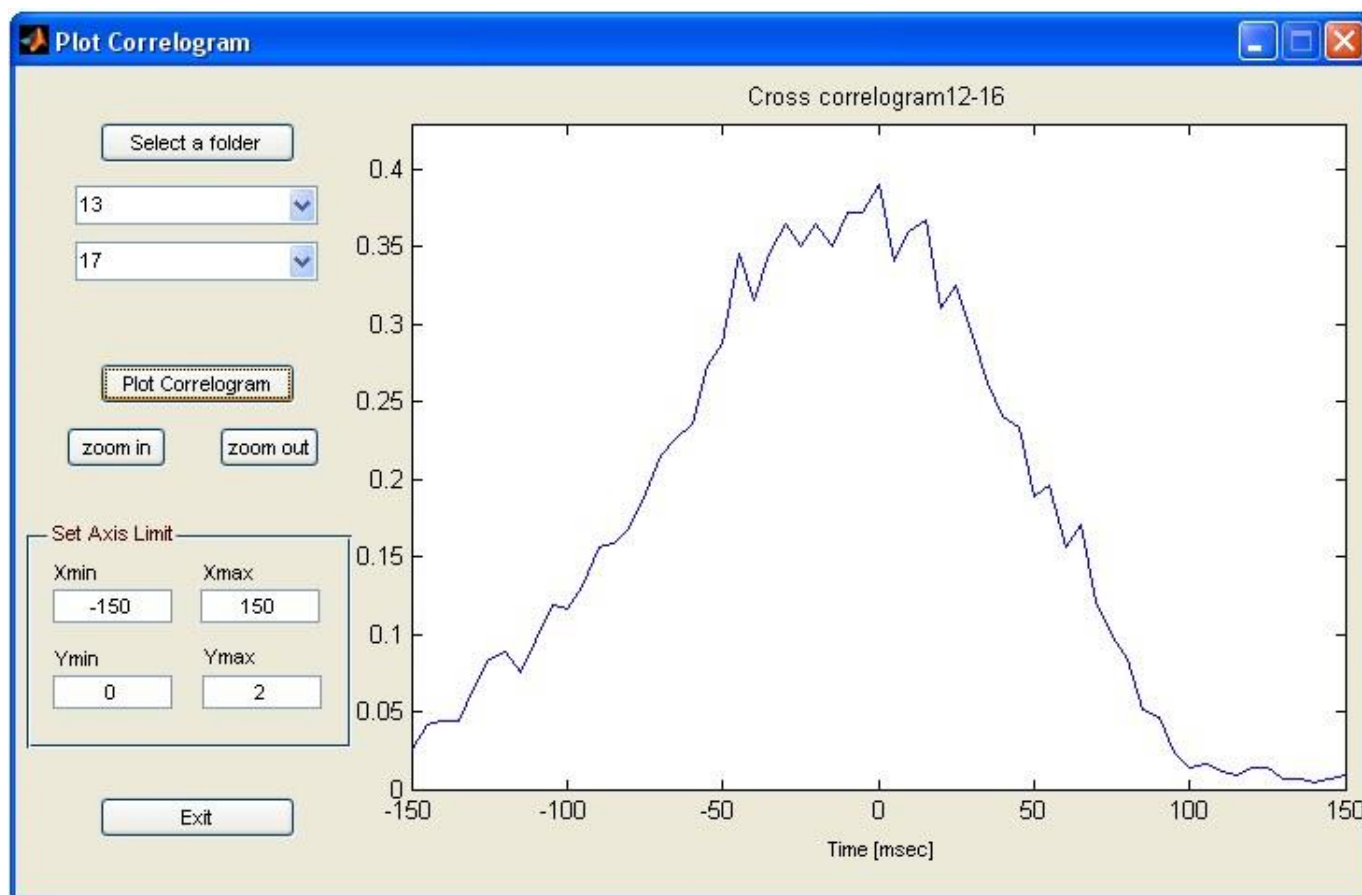


Figure 17. Correlogram plot

Three dimensional correlograms can also be generated, in menu *Cross Correlogram> Plot 3D Correlogram*. The results (Figure 18) are saved in the folder *_CCorr_...3DPlot\3DCorrelogram*.

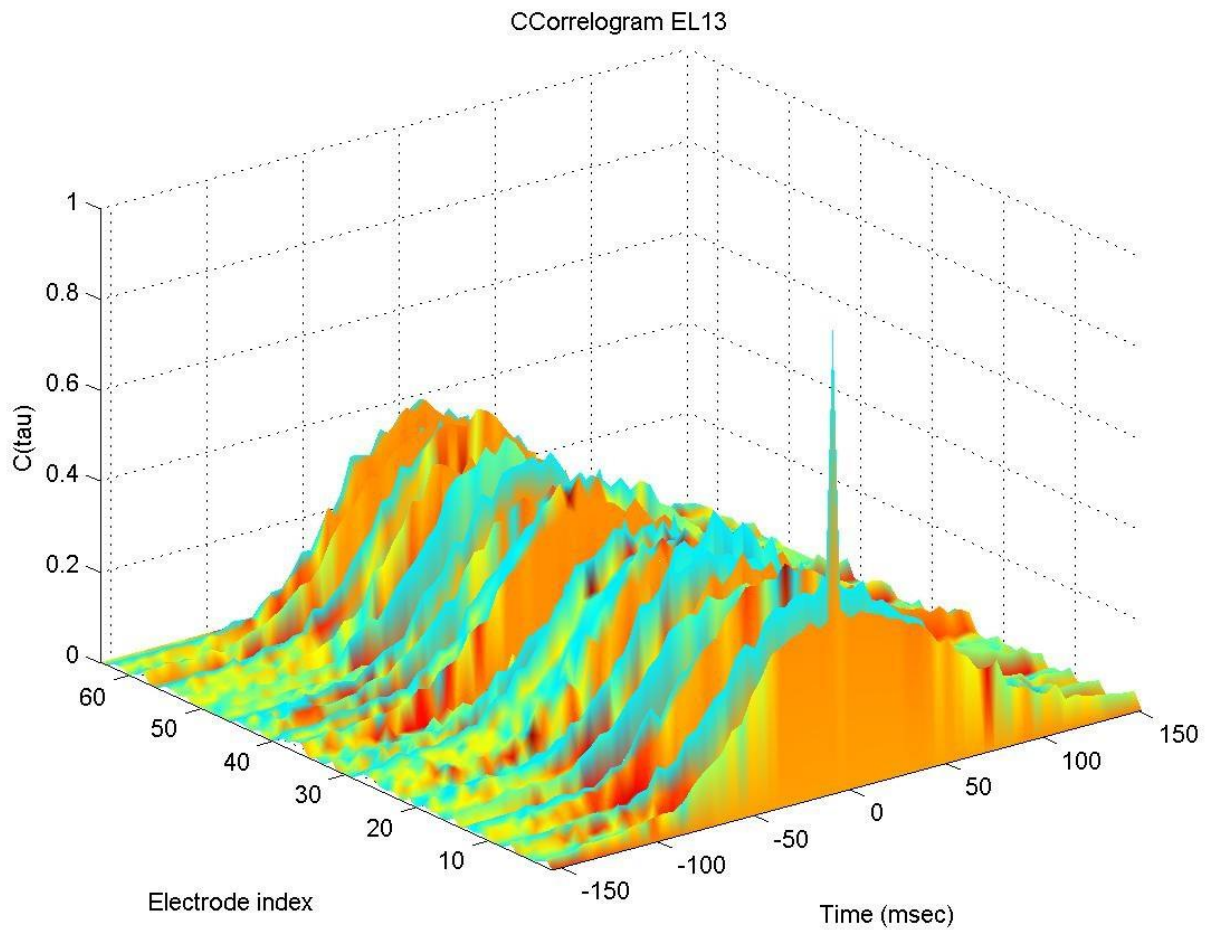


Figure 18. 3D Correlogram

Another way of visualize data is plotting the mean correlogram, in menu *Cross Correlogram*> *Plot Mean Correlogram*. Figure 19 shows an example of the mean correlograms in an 8x8 array, which is saved in the folder *CCorr_...MeanPlot*.

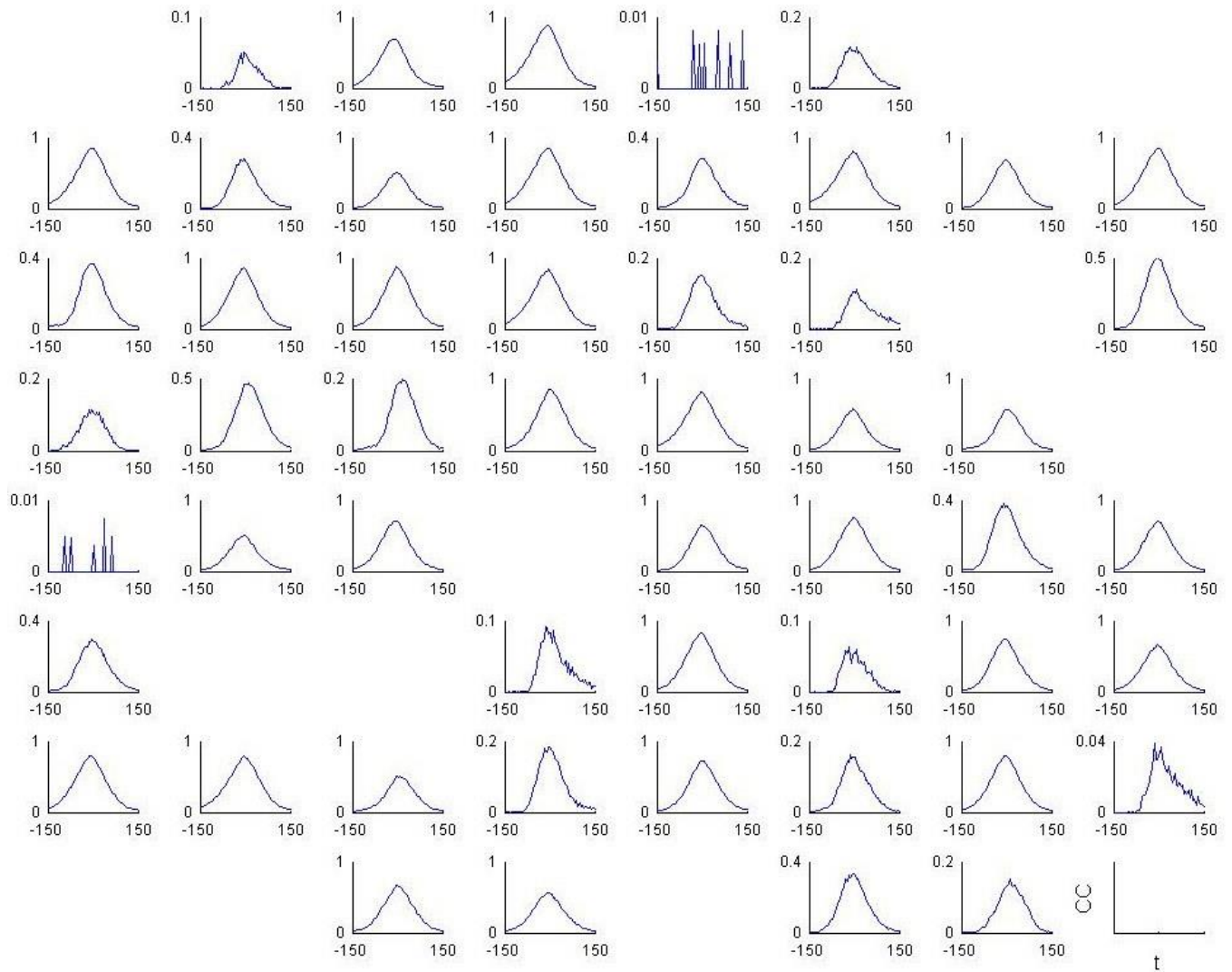


Figure 19. Mean Correlogram

ADDITIONAL TOOLS

Avalanches Analysis GUI

A dedicated GUI allows launching the neuronal avalanches detection and analysis. It's possible to set the time resolution for detecting avalanches and decide whether apply a logarithmic binning. The same GUI offers the possibility to apply the neuronal avalanches analysis to other MEA layouts and visualize the results' figures.

The guide, showed in Figure 20, allows choosing a specific algorithm to detect the avalanches. By clicking in button Load figure(s) is possible to visualize the resulting figures directly in Matlab.

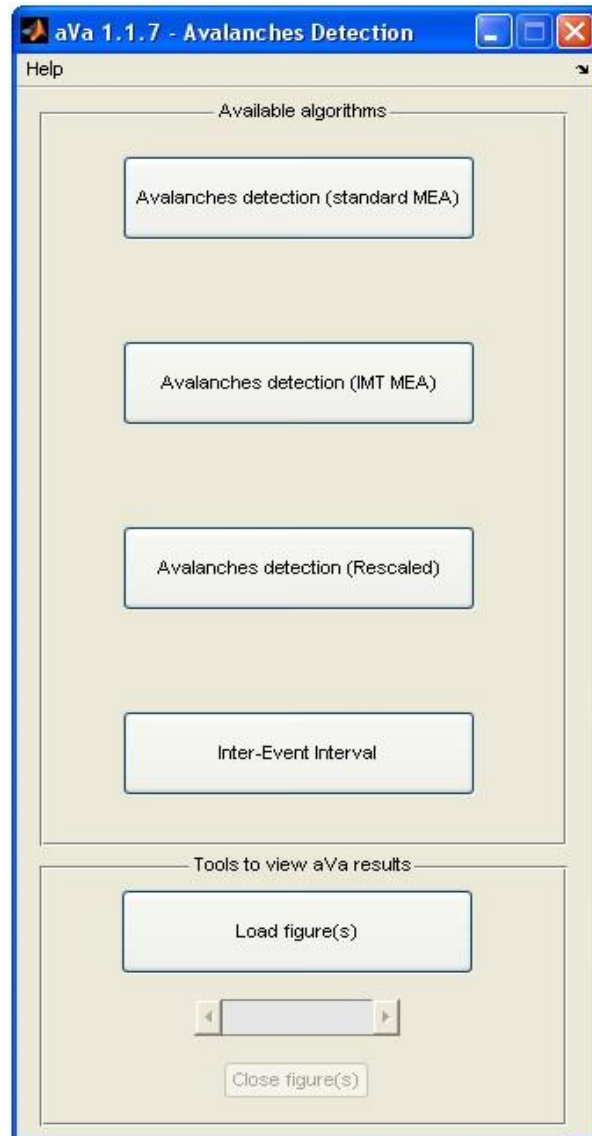


Figure 20. Avalanches Detection GUI

Figure 21 shows the parameters, like the bin widths, that shall be chosen by the user to perform the selected algorithm.

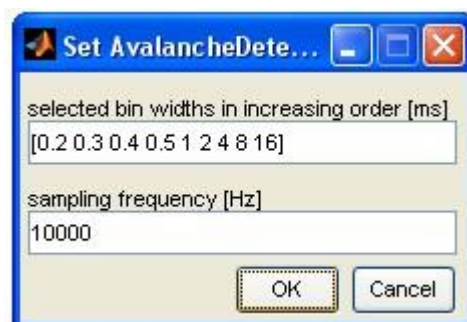


Figure 21. Avalanches Detection Parameters

Figure 22 shows an example of combined results performed by avalanches detection, and Figure 23 shows individual avalanches detection, performed for different bin widths. All results are saved in folder *_aVaAnalysis*.

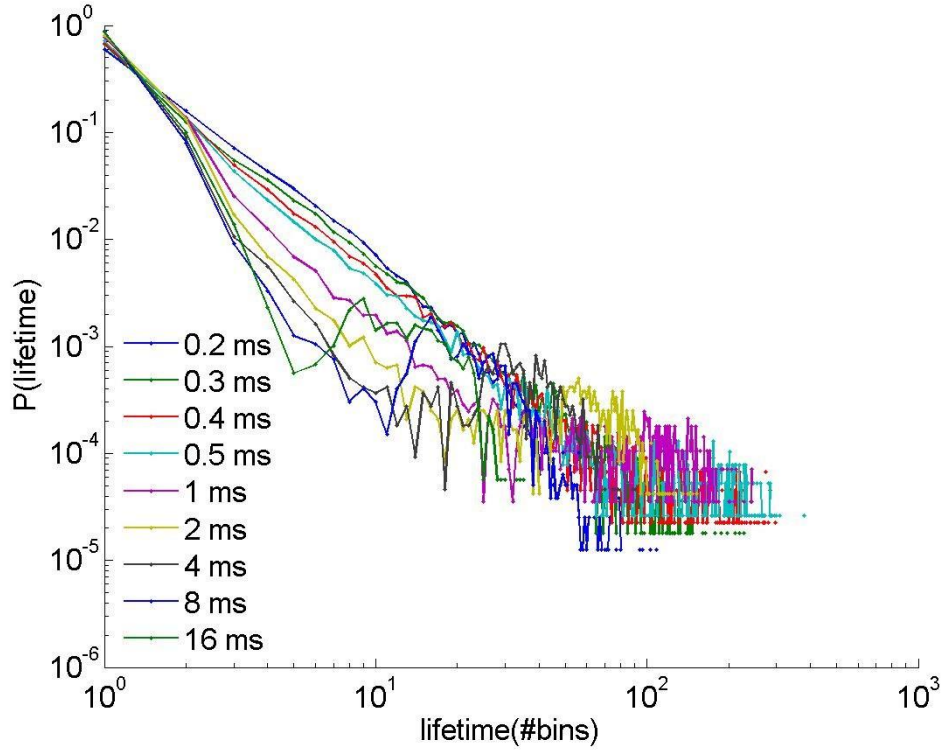
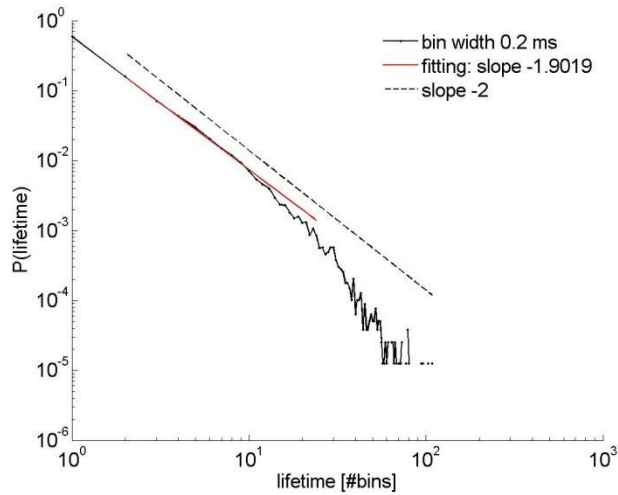
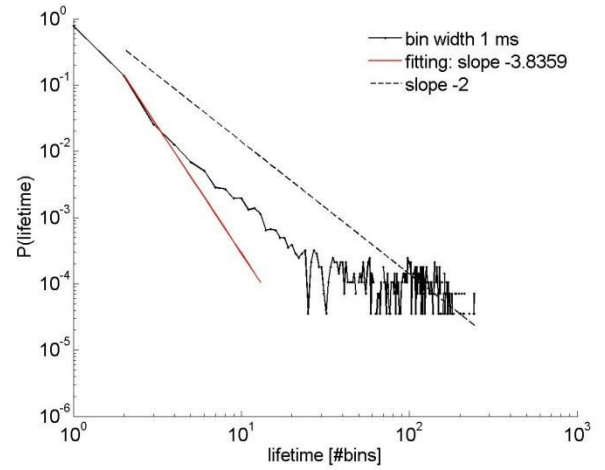


Figure 22. Avalanche Detection



(a)



(b)

Figure 23. Avalanche Detection for different bin widths: (a) 0.2 ms; (b) 1 ms

Information Theory GUI

This function will be available in a future version of SpyCode.

Shuffling Methods

SpyCode permits also shuffling Peak Train channels, separately (menu *Single Electrode*) or for all channels (menu *All Electrodes*). The results are saved in a chosen folder by the user.

There is a tool in SpyCode that allows shuffling the peak trains obtained after detection. Briefly, the method allows binning the spiking activity into appropriately wide temporal windows and then permuting the order of the bins either separately for each channel or for all channels at the same time. This procedure can be useful, for example, to validate the results obtained from the avalanche analysis or for calculating the excess amount of spike correlation.

MULTIPLE ANALYSIS

The ‘multiple analysis’ GUI, shown in Figure 24, relies upon the directory tree structure generated by the software itself. The latter is engineered so to create different folders for the different analysis performed. The two most important folders are the ones containing raw data in .mat format and the one containing the spike trains, built after the spike detection was performed.

Given the directory tree generated by the program and the common practice of archiving different experiments with different names and in different folders, the philosophy underlying the multiple analysis, can be outlined with the following steps:

1. Choose the root folder from which to start the search for the experiments to analyze;
2. Give the sequences of characters that the name of the experiments to be analyzed must fulfill;
3. Choose the analysis to perform on the experiments and the parameters to be used;
4. Launch the analysis.

The image shows a software window titled "Multiple Analysis" with a blue title bar and standard Windows window controls. The window contains several tabs: "Common Parameters", "Peak Detection", "Plot", "PSTH", "Spike Analysis", "Burst Analysis", and "Cross Correlation". The "Peak Detection" tab is currently selected. Inside this tab, there is a checked checkbox labeled "perform peak detection". Below this, there are two radio buttons: "PDDT" (which is selected) and "RTSD". To the right of the "PDDT" radio button, there are three input fields: "Sliding window length (msec)" with the value "2", "Peak lifetime period (msec)" with the value "2.0", and "Standard deviation coefficient" with the value "8". To the right of the "RTSD" radio button, there are two input fields: "Refractory period (msec)" with the value "1.0" and "Standard deviation coefficient" with the value "8". At the bottom of the dialog, there are two buttons: "OK" and "Cancel".

Parameter	Value
perform peak detection	<input checked="" type="checkbox"/>
PDDT	<input checked="" type="radio"/>
RTSD	<input type="radio"/>
Sliding window length (msec)	2
Peak lifetime period (msec)	2.0
Standard deviation coefficient (PDDT)	8
Refractory period (msec)	1.0
Standard deviation coefficient (RTSD)	8

Figure 24. Multiple Analysis Guide

CONTACT INFORMATION

**Department of Neuroscience and Brain
Technologies** Italian Institute of Technology
www.iit.it

Department of Biophysical and Electronic Engineering
University of Genova

SpyCode Authors

Michela Chiappalone; Valentina Pasquale; Luca Leonardo Bologna
michela.chiappalone@iit.it

User Guide Authors

Ricardo de Camargos Lopes; Luca Leonardo Bologna; Alberto Averna

References

- Wagenaar, D.A., Pine, J. & Potter, S.M. (2006) An extremely rich repertoire of bursting patterns during the development of cortical cultures. *BMC neuroscience*, **7**, 11.
- Wagenaar, D.A. & Potter, S.M. (2002) Real-time multi-channel stimulus artifact suppression by local curve fitting. *Journal of neuroscience methods*, **120**, 113-120.