Software Manual Network Burst Detection

# Main Routine

resultCell = main\_MEA\_BurstAnalysis(varargin)

if nargin ~= 0

z = varargin{1};

end

z: in a time series, z defines the file that is used for data cleaning; clears wells/electrodes that do not meet defined conditions

## Load Files

The user selects files to evaluate and location to save the result; multiple file selection is ‘on’, so files of a time series can be loaded.

firstList = getList('instruction','insert baseline files','multiSelection','off',...

'folder',basePath,'file',baseInfo{z});

[2 Functions 2](#_Toc63768180)

[2.1 getList 2](#_Toc63768181)

[2.2 cleanData 2](#_Toc63768182)

# Functions

## getList

firstList = getList(varargin);

This is a function to read the .csv file containing spike list data by Axion. The single table column containing the timestamps for all spikes on all electrodes chronologically is sorted and rearranged returning the spike time stamps sorted to corresponding electrode in a column. Optionally, specific electrodes can selected by providing a list of electrode names (NameList)

The output variable **firstList** is a 1x1 struct containing the fields

|  |  |
| --- | --- |
| .experiment: | 1x1 cell array containing the name of the recording |
| .data: | n x m cell array, with n = number of max. spike number per electrode and m = number of electrodes; containing time stamps of spike per electrode |
| .nameElectrodes: | M x 1 cell array, with m = number of electrodes, containing names of electrodes |

Input of Name-Value-Pairs

|  |  |
| --- | --- |
| ‘instruction’ | ‘*text*’ - Adjust the instruction in user interface  Default: ‘insert file’ |
| ‘multiselection’ | ‘on’, ‘off’  Default: ‘off’ |
| ‘NameList’ | Cell array containing names of electrodes that have to be selected |
| ‘folder’ | Optional: ‘*text*’ – name of path |
| ‘file’ | Otional: ‘*text*’ – name of files |
| If ‘folder’ || ‘file’ are not provided, user is ask to select experiments be uigetfile | |
| ‘saveName’ | Optional: ‘text’ – name of path to save |
| ‘savePath’ | Optional: ‘text’ – filename for saving |
| If ‘saveName’ ||’savePath’ is not provided, flagSave = 0 (no saving) | |

## cleanData

[spikeListExcl, spikeName, wellNames] = cleanData(data, varargin)

This is a function selecting active/valid electrodes and valid wells/networks according to defined conditions:

Minimum firing rate: default is 0.1, can be passed as 2. Input variable;

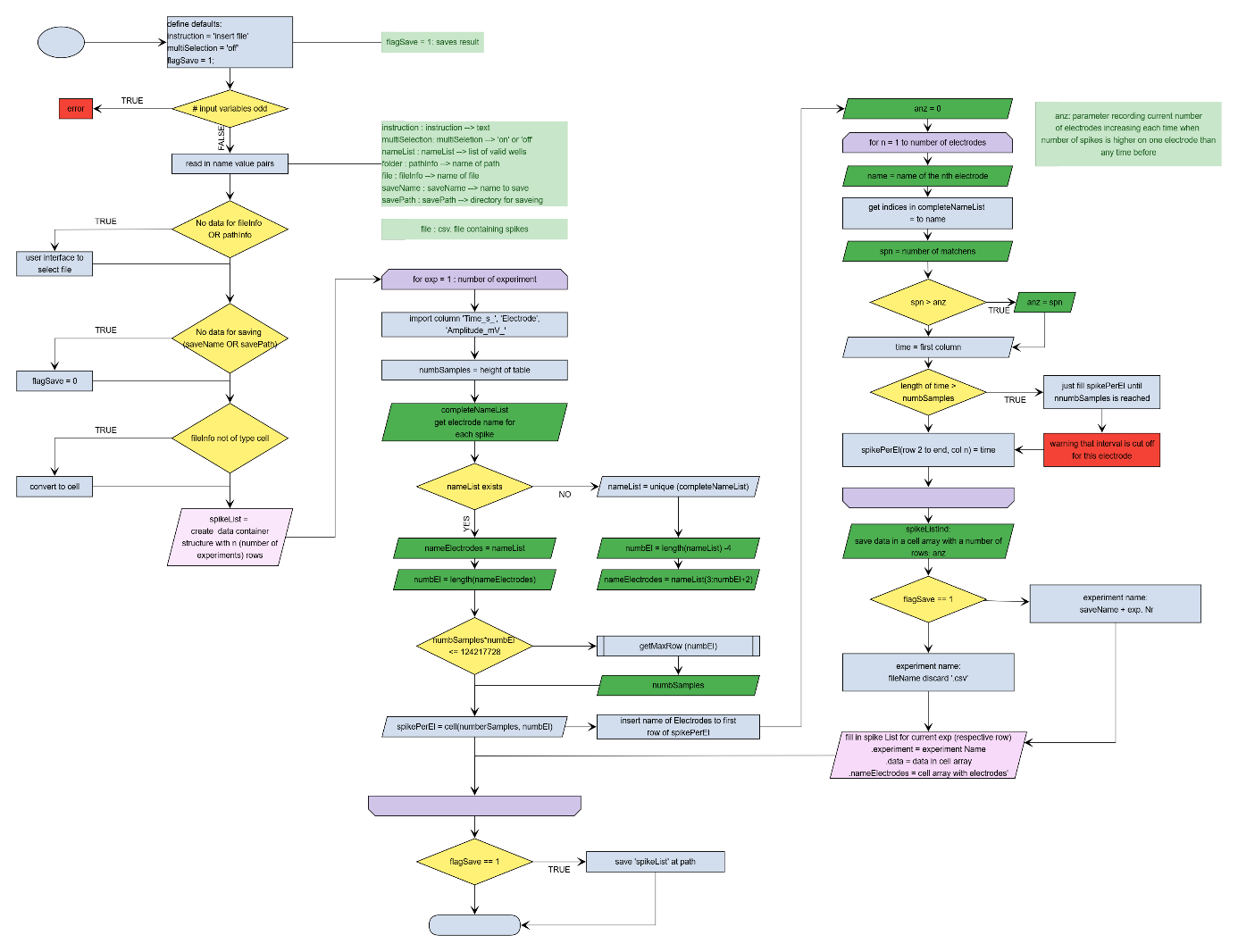
Minimum number of active electrodes: has to be set manually in getWells(QV);

Call of functions:

* conversion (2.3)
* getInterval (2.4)
* getSpikeNum (2.5)
* getWells

Output variables

|  |  |
| --- | --- |
| spikeListExl: | 1x1 cell array containing the name of the recording |
| spikeName | 1xn cell array, with n is number of valid electrodes, containing the names of valid electrodes |
| wellNames | 1xn cell array containing the list of valid names |



## conversion

[spikeArray,emptyIndex] = conversion(spikeList, decl)

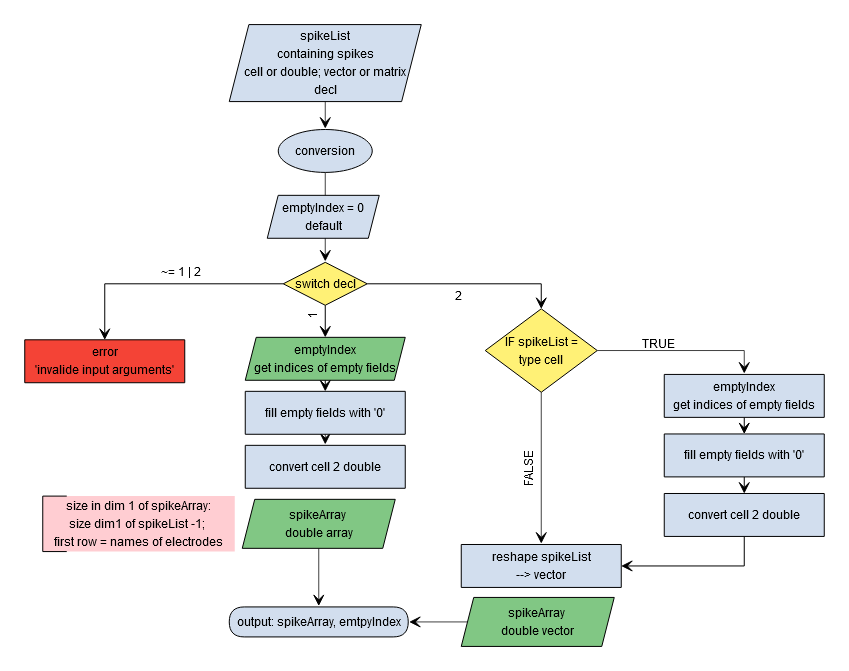
This function is for converting and rearrange array; before conversion from cell to double empty fields are detected in cell array and replaced by 0;

Input variables

|  |  |
| --- | --- |
| SpikeList | Cell array or double array |
| decl | Declare dimension of the output variable:  1: converts a cell array to double array; dimension remain the same  2: converts an array to a 1 dimensional array (vector) of type double. If input variable is cell type, it’s converted to double; |

Output variables

|  |  |
| --- | --- |
| spikeArray | Converted array  decl = 1: double array  decl = 2: double vector |
| emptyIndex | Empty indices in input array (used to replace with ‘0’ , to convert to double) |



## getInterval

[interval,start,stop] = getInterval(spikeVec)

This function will calculate the duration of the recorded bin; the function determines the first and the last time stamp values as start and stop and calculates interval as stop –start;

Input variables

|  |  |
| --- | --- |
| spikeVec | Data array containing time stamps of all electrodes;  In the case of dim ~1 (no vector) 🡪 run conversion (2.3) |

Output variables

|  |  |
| --- | --- |
| interval | Duration in seconds |
| start | Time point of first time stamp |
| stop | Time point of last time stamp |

## W:\psychiatrie\science\MolPsych\weissea\Flowcharts\MEA_Analysis2021\getInterval.png

## getSpikeNum

numbSpikes = getSpikeNum(spikeList, varargin)

This function calculates the number of spikes for each individual electrode. By default empty fields are determined – for each electrode

IF spikeList type cell

emptyIndex = cellfun(‘isempty’, spikeList),

ELSE

emptyIndex = spikeList == 0;

END

Second argument to pass list of empty fields;

Number of spikes for current electrode = max. spike number on any electrode – sum of empty fields on current electrode

Input variables

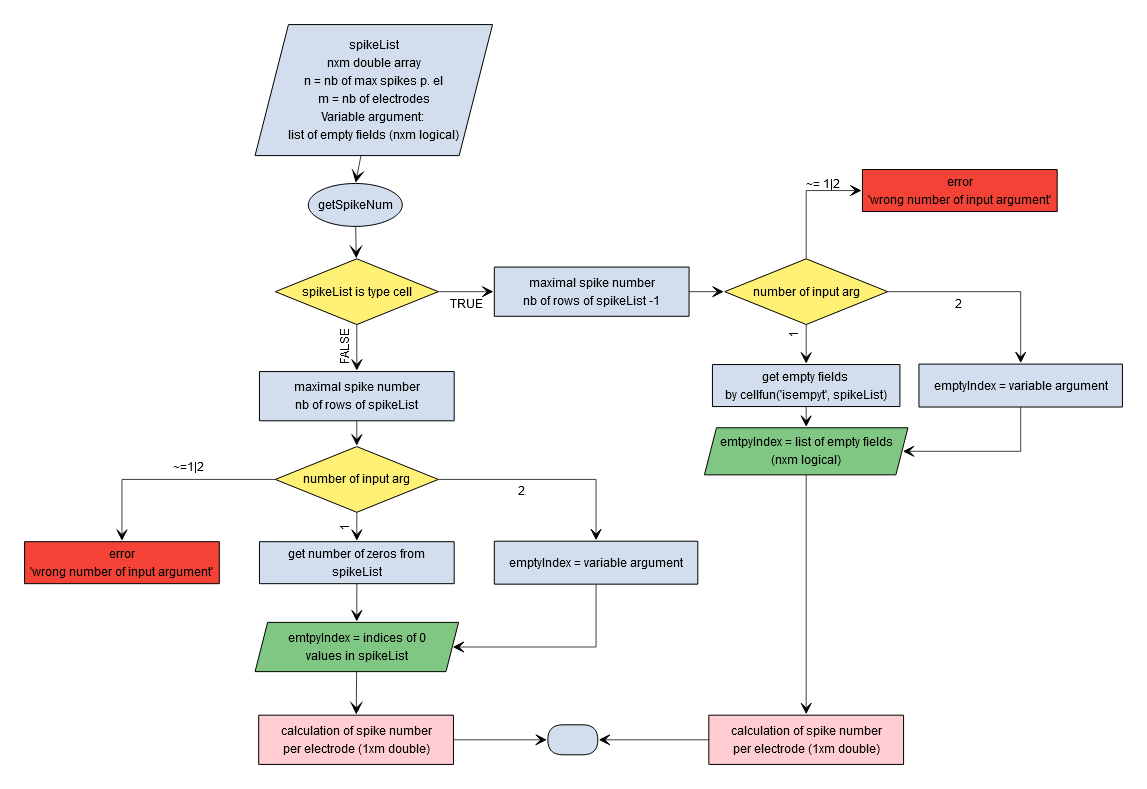
|  |  |
| --- | --- |
| Spike List | Cell or double array of spikes |

Output variables

|  |  |
| --- | --- |
| numbSpikes | Number of spikes for each electrode, 1 x n double array with n = number of electrodes |

Varargin

|  |  |
| --- | --- |
| 2°nd parameter | If narargin == 2  emptyIndex = varargin{1} |



## getWells

[list,wellNames] = getWells(data, varargin)

This function groups electrodes into wells and subsequently selects valid wells.

Input variables

|  |  |
| --- | --- |
| data | M x n Cell array containing spikes for each electrode with m = max number of + 1spikes and n = number of electrodes; first line contains names of electrodes |

Varargin

|  |  |
| --- | --- |
| 2nd parameter | If narargin >= 2  excluding = varargin{1};  if excluding == -1  clear excluding  end  end  Number of minimum active electrodes; excludes wells that do not meet this condition  -1: the variable excluding is cleared 🡪 no excluding parameter |
| 3th parameter | if nargin == 3  redList = varargin{2};  redList = num2cell(redList);  end  an already existing list of wells can be passed, associated electrodes are grouped to these wells;Δ  IF ~exist(redList)  redList = name list of all electrodes  extract first 2 characters (name of corresponding well) |
|  |  |

