Please read first carefully the paper Gullo et al., J. Neurosci. Meth. 2011, and afterwards read and follow the suggestions below:

Create the directory in "neurocode2"

We hypothesized that you will download all the following files in the directory "neurocode2" in hard disk D:

MEAdir.7z	this compressed file contains the source code in Python
timeneuro_mod.7z	this file contains the executable "timeneuro_mod.exe" file, please extract!
burst_detection_settings.txt	this text file (do not change name) contains all the parameter settings
experiment_to_process.txt	this text file (do not change its name) contains all the experiment "names"
321.7z	this compressed file contains a complete experiment called 321, please extract!

!!!! In the same directory in which you will download the executable, also the 2 text files should be present!!

In the file "experiment_to_process" you will see that there are many lines commented with a "#" , but there is only one not commented, namely "321". This is the sample experiment given for learning purposes.

Extracting the experiment

After extracting 321, you will see that there are 3 directories and 2 Origin files which contain a template file and a completely "full-of-data" file from experiment "321". This is a typical structure for any novel experiment.

The *.plx file (or *-nex) containing all the timestamps must have name "321" and will stay in directory "*MEAdata_plx*".
Moreover, the time segment list with their start and stop times must stay in directory "*MEAdata_settings*" and should have the format of the text file that you will find inside (check for no spaces after each line).
The Origin file "template for importing csv output files in ORIGIN graphs 20ott2011" is a template "data-free" file created for starting a novel analysis of an experiment (it will be seen by changing the horizontal window that contains many graphing windows).
The "Origin file of experiment 321_31ott2011" Origin file is full of results from the 321 experiment. Try to use the analysis program and import the result in the template file. At the end, your origin file should be identical to file "Origin file of experiment 321_31ott2011.OPJ".

A) Procedure to import csv files into origin spreadsheet files

Preliminary management of each data type

Remember that all of the results data will be in the directory "*MEAdata_cluster_results*". Notice that there are 2 sub-directories for the data computed either by using the neuron-by-neuron strategy or the Network-Burst strategy.

To visualize the data output (in *.csv files), all the output data had been organized in spreadsheets that can be automatically imported in the proper spreadsheets in a template file of ORIGIN (version PRO 7), in which each spreadsheet is linked to pre-organized windows [GRAPHs].

For each type of results accomplish the following 4 general rules:

1) by selecting the appropriate spreadsheet (notice that there is a legend with a name, remember it),
2) then CTRL+k,
3) select into directory "*MEAdata_cluster_results*" the csv file memorized before and click
4) the window near the spreadsheet will become full of data in each graph.

List of the available results in term of "statistical" properties

Import the file **clusters_data** (found in directory *single-neuron-burst-results)* into spreadsheet Data4 (upper-left corner) and you will see in window "cluster" data like BD, SN, ACF, FF, CV2, SR, etc.. Select the upper legend, go to "properties", select OK and in the legend you will read all the important properties of the experiment and the parameters used during the computation. Update x-scale!

Import the file **ACF_time-segment name** (found in directory *burst_data_extracted)* into spreadsheet ACF1 (middle-left) and you will see in window "ACF1e2" the waveshapes of the autocorrelograms. Select the upper legend, go to "properties", select OK and in the legend you will read all the important properties of the experiment and the parameters used during the computation.

Import the file **netstate_data2** (found in directory *Network-burst-results)* into spreadsheet state2 (middle-middle) and you will see in window "Gstates" (below) data like SN, Excitability, Normalized excitability, IBI, Neurons, SR, all computed by using the burst-by-burst strategy. Select the upper legend, go to "properties", select OK and in the legend you will read all the important properties of the experiment and the parameters used during the computation. Update x-scale!

Import the file **CumCount15_1** (found in subdirectory *counthistoNeurClasses* of directory *Network-burst-results)* into spreadsheet netstates1 (middle-middle) and you will see in window "SpHisSt15" (upper-middle) data of "cumFSH" computed by using the burst-by-burst strategy; there are 15 pairs (1st and 2nd states) of graphs for each histogram for the first time-segments (to see results from more than 15 segments go to right into spreadsheet netstates2 and window "nsSR30"). Select the upper legend, go to "properties", select OK and in the legend you will read all the important properties of the experiment and the parameters used during the computation. Update legends by selecting the indicated text!

Follow the same procedures for the following spreadsheets and windows (notice that the file has the property of being horizontally-extend at right):

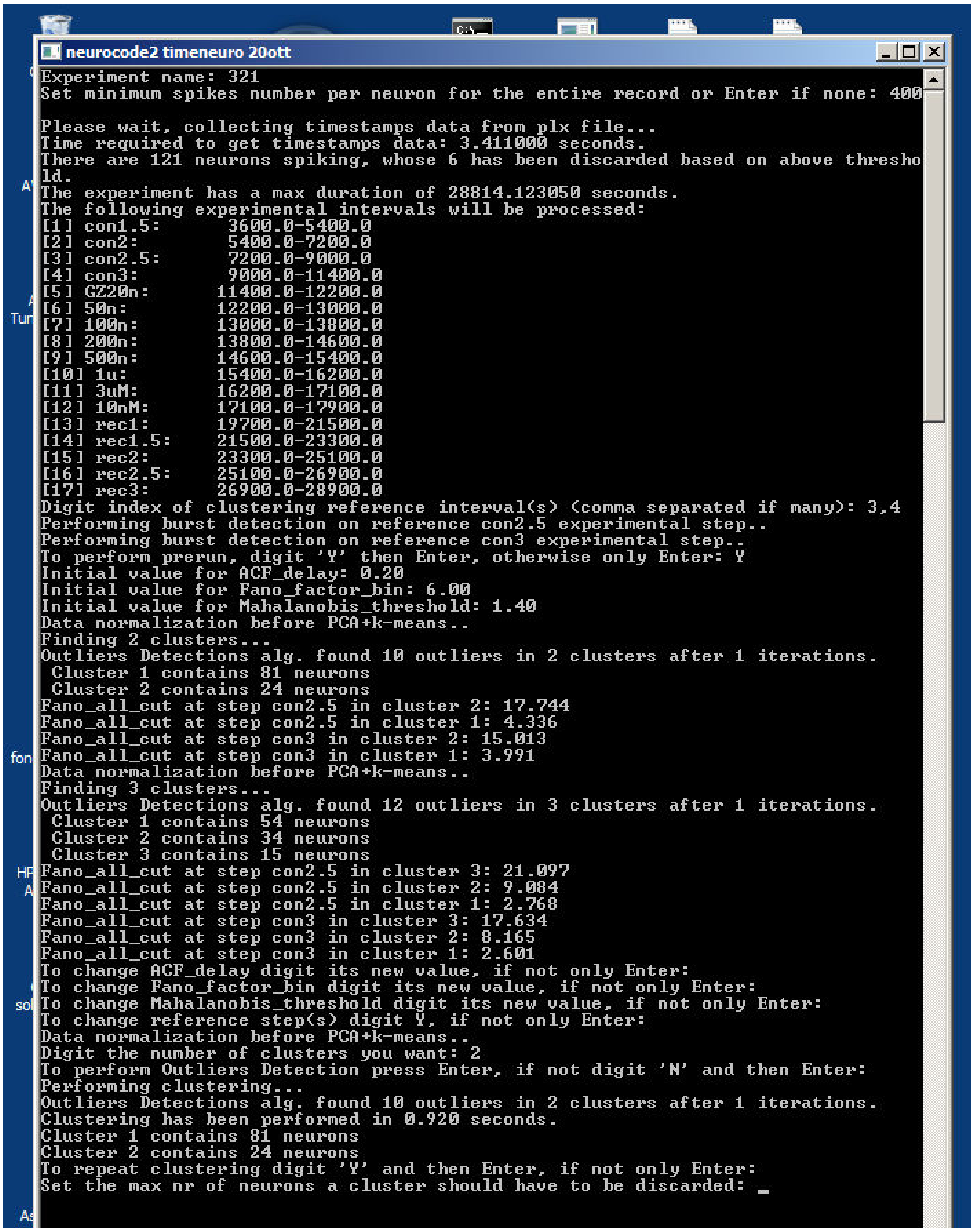
	ORIGIN	ORIGIN	histograms
Directory	**Spreadsheet**	**window**	**data**
nb_clusters	netstate1F	FFpeTH15	EXCTH of segments 1-15
nb_clusters	netstate2F	FFpeTH30	EXCTH of segments 16-30
nb_clusters	netstate1N	nsNN15	SN or NN TH of segments 1-15
nb_clusters	netstate2N	nsNN30	SN or NN TH of segments 16-30

B) Running "timeneuro_mod"

The timeneuro_mod program, besides running with a fixed series of parameters, allows also some preliminary user-controlled changes of parameters (Fano-factor (FF) window and ACF_delay) by input the proper values or reject some analyses (see below).

Remember to define time segments not larger than ~30 minutes because the computing time is increasing with a **power of 2** if you will use time segments longer than 30 min.

For running the analysis select the file "timeneuro_mod.exe" in directory "timeneuro_mod" and the following screeshot will appear:



The first decision is the eventual removal of some units with few timestamps (in the total experimental length), typically deleting units with a number of timestamps lower than 800 for an acquisition time of 4 hours is normal, because the statistics obtained from these units are dramatically poor! (input a number: say 200, or Enter). Here we used 400 and 6/121 units were discarded.

The 2nd decision is "<u>which time-segment should be considered as reference?</u>" Here we used intervals identified by 3,4 [see these numbers in square brackets]. The user should input one or more time segments that are considered a "CONTROL" and reference situation for the activity of the network. The program will respond with preliminary results on how many units were assigned into the two clusters and the Fano factor values are given for each of the cluster detected. In general, the more populated cluster (cluster1) is that belonging to the excitatory neurons, followed by cluster2 (inhibitory neurons). Since the former and the latter fire few or many spikes, respectively, the associated values of the Fano Factor are accordingly either small (1.5 – 3.5) or high values (7 - 25), here we found 17.7 and 4.3 in the time-segment "con2.5" (see in the window).

From dish-to-dish there are changes in the number of neurons and especially inhibitory neurons (Gullo et al., 2009, 2010). To finely and correctly adjust, in each experiment, the number of units assigned to the 2nd cluster (inhibitory cells) in such a way to reach the correct excitatory/inhibitory ratio to ~4, the user can operate as follows: 1) check the first results and if the number of 2nd cluster units is too low, slightly decrease FF (or ACF_delay if used) and a higher number of units will be observed, 2) if the number is too high operate inversely. If the number of outliers is too much, increase slightly the "<u>*Mahalanobis_threshold*</u>" from 1.4 to 1.6.

If you are not satisfied of the 2-cluster division there is the possibility to make a second choice by using <u>*the 3-cluster classification*</u> and disregard the smallest cluster outcome by setting the "max nr of neurons" to the this value: you will notice that the computation will start regularly with only 2 clusters.

If you want to use ACF instead of FF, remember that you have to modify the "burst_detection_settings.txt" file (namely, add 'ACFPC1' or 'halflife_ACF') in the square brackets of line 28. The program offers the possibility to change the maximal time delay in the ACF computation (name is "ACF delay"). Remember that now this time is fixed (0.2 in seconds) in line 65 of the settings file, but it can be automatically changed if you respond with a number the text line in the running window: <u>put a smaller or a larger value if you want to increase or decrease the number of inhibitory neurons</u>.

The program will continue the data analysis with the neuron-by-neuron strategy (Gullo et al., 2011, Fig. 1) and afterwards it starts the network burst (NB) controlled analysis up to end, but before ending completely there is the possibility to re-do this last NB-analysis with different parameter values (follow the definition according to what is listed in the file "burst_detection_settings.txt"). The novel computation is automatically saved in a new directory "*Network_burst_results*" stored in directory "*MEAdata_cluster_results*".

```
To exit digit N, to change settings Enter:
Input new trash_poor_states or Enter:
Input new Net_features(comma separated) or Enter:
Input new bin or Enter:
Input new binstates or Enter:
Input new delay or Enter:
Input new pvalue or Enter:
Input new nr_pvalue or Enter:
Collecting network burst data to find neurons classes..
Finding classes based on neurons histograms spike counts
Performing network burst analysis on step con1.5
Performing network burst analysis on step con2
Performing network burst analysis on step con2.5
Performing network burst analysis on step con3
Performing network burst analysis on step GZ20n
Performing network burst analysis on step 50n
Performing network burst analysis on step 100n
Performing network burst analysis on step 200n
Performing network burst analysis on step 500n
Performing network burst analysis on step 1u
Performing network burst analysis on step 3uM
Performing network burst analysis on step 10nM
Performing network burst analysis on step rec1
Performing network burst analysis on step rec1.5
Performing network burst analysis on step rec2
Performing network burst analysis on step rec2.5
Performing network burst analysis on step rec3
To exit digit N, to change settings Enter: N
Date: 02/11/2011  h: 18:05:15
Amount of time occurred for each processing step:
 Timestamps data retrieving: 3.997s
 Single neuron burst detection: 35.349s
 Results storage: 0.681s
 Network burst detection: 108.859s
Total amount of time occurred:  148.887s
-------------------------------------------------------------------------------
```

```
Set the max nr of neurons a cluster should have to be discarded:
Performing burst detection on con1.5 experimental step..
Performing burst detection on con2 experimental step..
Performing burst detection on GZ20n experimental step..
Performing burst detection on 50n experimental step..
Performing burst detection on 100n experimental step..
Performing burst detection on 200n experimental step..
Performing burst detection on 500n experimental step..
Performing burst detection on 1u experimental step..
Performing burst detection on 3uM experimental step..
Performing burst detection on 10nM experimental step..
Performing burst detection on rec1 experimental step..
Performing burst detection on rec1.5 experimental step..
Performing burst detection on rec2 experimental step..
Performing burst detection on rec2.5 experimental step..
Performing burst detection on rec3 experimental step..
Writing results data..
Collecting network burst data to find neurons classes..
Finding classes based on neurons histograms spike counts
Performing network burst analysis on step con1.5
Performing network burst analysis on step con2
Performing network burst analysis on step con2.5
Performing network burst analysis on step con3
Performing network burst analysis on step GZ20n
Performing network burst analysis on step 50n
Performing network burst analysis on step 100n
Performing network burst analysis on step 200n
Performing network burst analysis on step 500n
Performing network burst analysis on step 1u
Performing network burst analysis on step 3uM
Performing network burst analysis on step 10nM
Performing network burst analysis on step rec1
Performing network burst analysis on step rec1.5
Performing network burst analysis on step rec2
Performing network burst analysis on step rec2.5
Performing network burst analysis on step rec3
To exit digit N, to change settings Enter:
Input new trash_poor_states or Enter:
```

It is normal that during the NB computation phase the program could crash, during the analysis of a particular time-segment, producing an error (whose origin in the source-code will be saved in a text file into the proper directory). This could always happen if the number of *recognized bursts is smaller than 3*. In this case, please increase the length of the time segment in the file present in the directory "*MEAdata_settings*".

At the end of this last computation the program offer the possibility to change some of the parameters that are crucial to do the classification of the network states: see the settings file at lines 260-280 (those that have the mention "end-mod" can be changed immediately) and the associated results will be displayed in novel subdirectories (with appropriate names).