

Help

Overview:

NeuroCube simulates realistic extracellular recordings with different electrode configurations. This tutorial shows how to define the characteristics of the neural tissue and obtain synthetic datasets using the Graphic User Interface.

Getting started:

All the files needed to run the simulator are in the folder NeuroCube, where the whole structure of subfolders is included. First of all, open Matlab and make NeuroCube/mat your current folder. Type *neurocube* to run the interface shown in Figure 1.

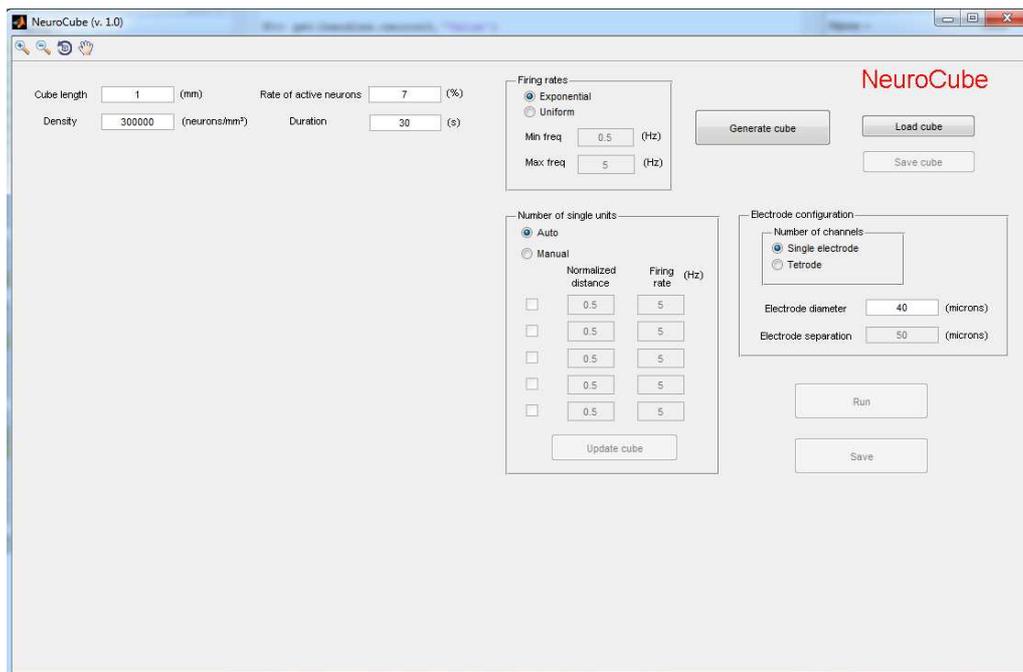


Figure 1

Generation of the cube:

In the interface you can set several parameters that describe the configuration of the cube: 'Cube length', 'Density', 'Rate of active neurons', 'Duration' and 'Firing rates'. 'Cube length' will define the size of the cube, while the number of simulated neurons will be given by both the density of neurons in the tissue and the rate of active neurons in %. You can choose between two different distributions of firing rates: exponential or uniform, indicating also the minimum and maximum frequency for the uniform distribution. The number of spikes generated by each neuron will be given by both the firing rates and the duration of the

simulation. After setting all these parameters, click ‘Generate cube’ and the distribution of neurons will be created. Figure 2 shows the interface after clicking ‘Generate cube’.

Instead of clicking ‘Generate cube’, you could also have clicked ‘Load cube’ if you wanted to load a neuron configuration that you had previously saved. Either if you generate the cube or if you load it, you can see a 3D plot of the neuron distribution together with the electrodes in your tissue. In order to make the plot clear enough, only a small proportion of the neurons is represented (by default, 1%). In this plot, the electrodes are represented as black circles of the size specified by the user, while the close-by and distant neurons are represented as red and blue triangles, respectively. You can click ‘Generate cube’ as many times as you want, obtaining different random distributions of neurons. Whenever you are happy with the result, you can save the scenario by clicking ‘Save cube’. A structure called *Neurons* will be saved in a *.mat file with the coordinates and identification (model used to describe it) of all the neurons, together with the firing rates. Only for the close-by neurons (those represented in red), the spike times will also be saved.

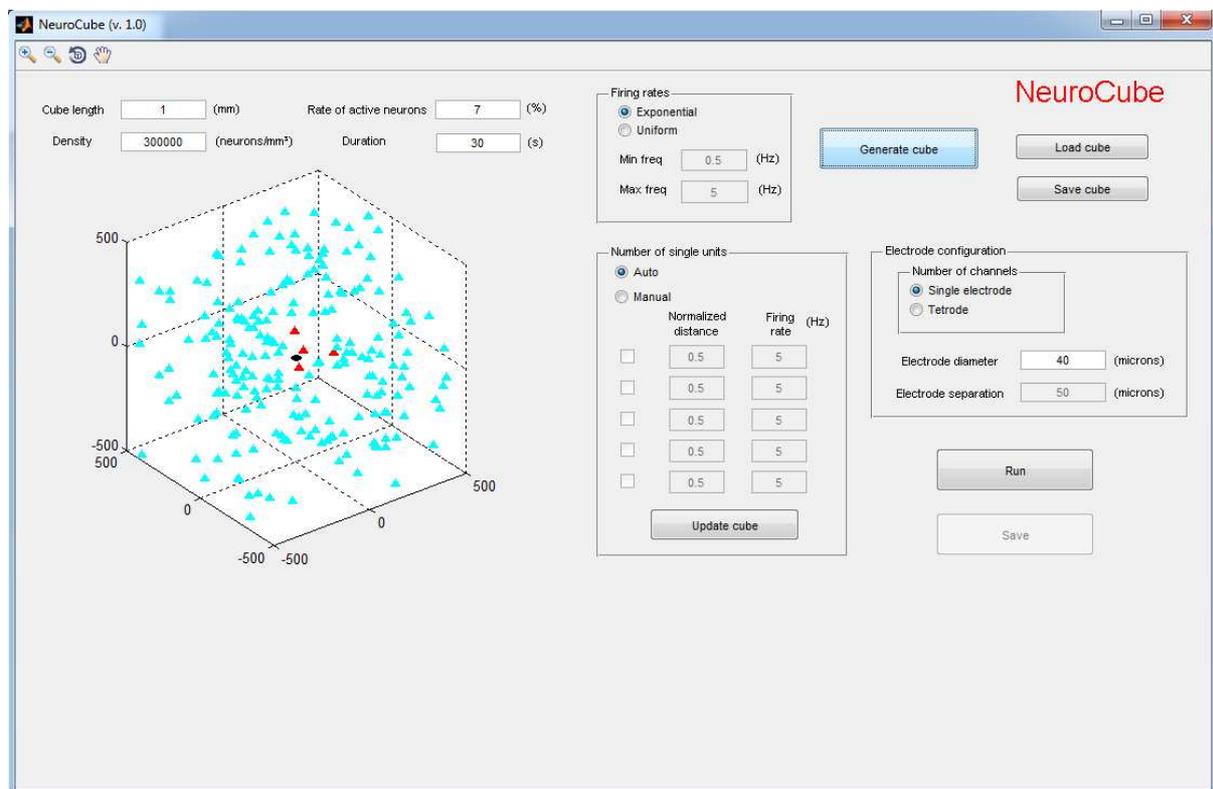


Figure 2

Manual selection of single units:

After the cube has been generated, you can decide whether to keep the automatic result or to decide manually the number of single units. If you want to control precisely the number of single units in your simulation, you have to select ‘Manual’, and activate up to 5 different neurons. Then, all the neurons that had been generated automatically within a fixed distance to the electrode are deleted, and the neurons selected by you are included. For each selected neuron, you can introduce the firing rate and the normalized distance (0 for the closest

possible distance to the electrode, and 1 for the limit of the volume considered for single units). After specifying all the single neurons and their properties, click ‘Update cube’ and the new neuron configuration will be plotted. Figure 3 shows the effect of selecting manual mode with two different single units.

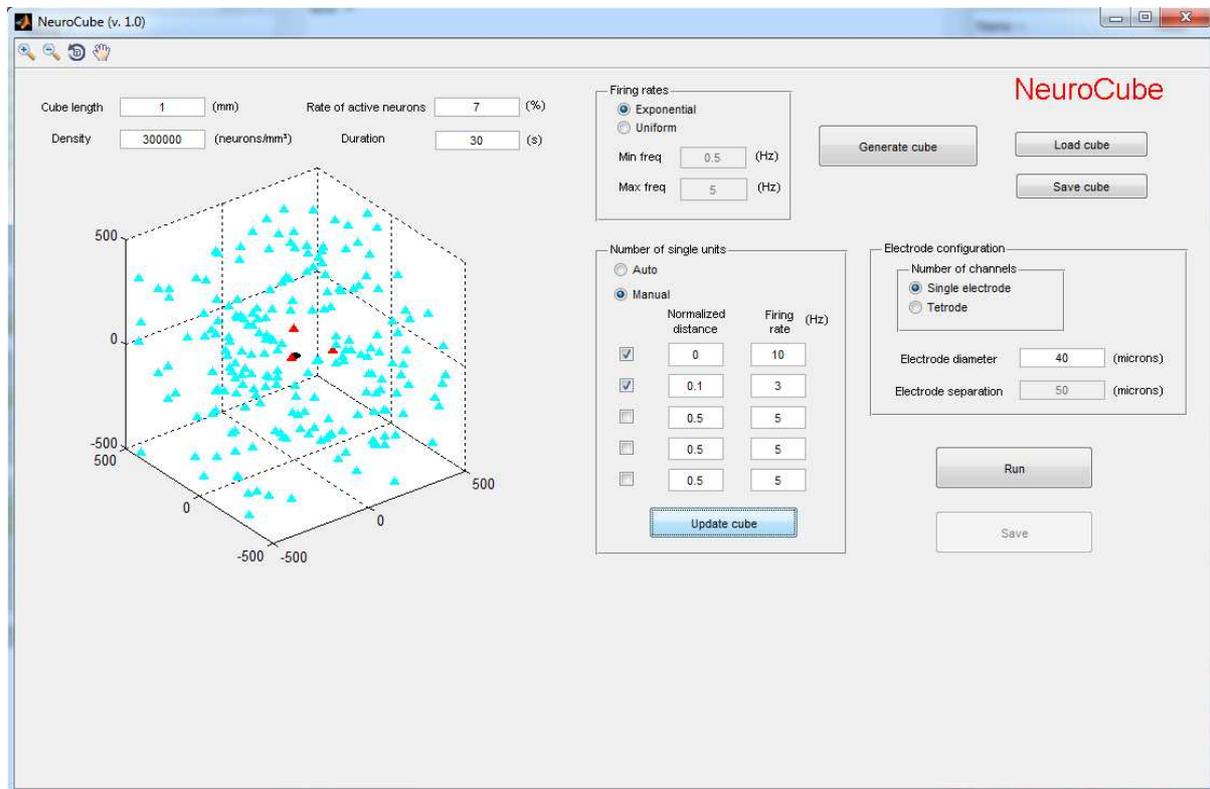


Figure 3

Electrode configuration:

After finishing with the neuron distribution, you can also change the properties of the electrode: you can choose either a single channel or a tetrode, and you can select the diameter of each electrode and the separation between them (if you choose a tetrode). Figure 4 shows the effect of selecting a tetrode with 40 microns of diameter and 50 microns of separation. Figure 5 shows the effect of increasing the separation to 70 microns.

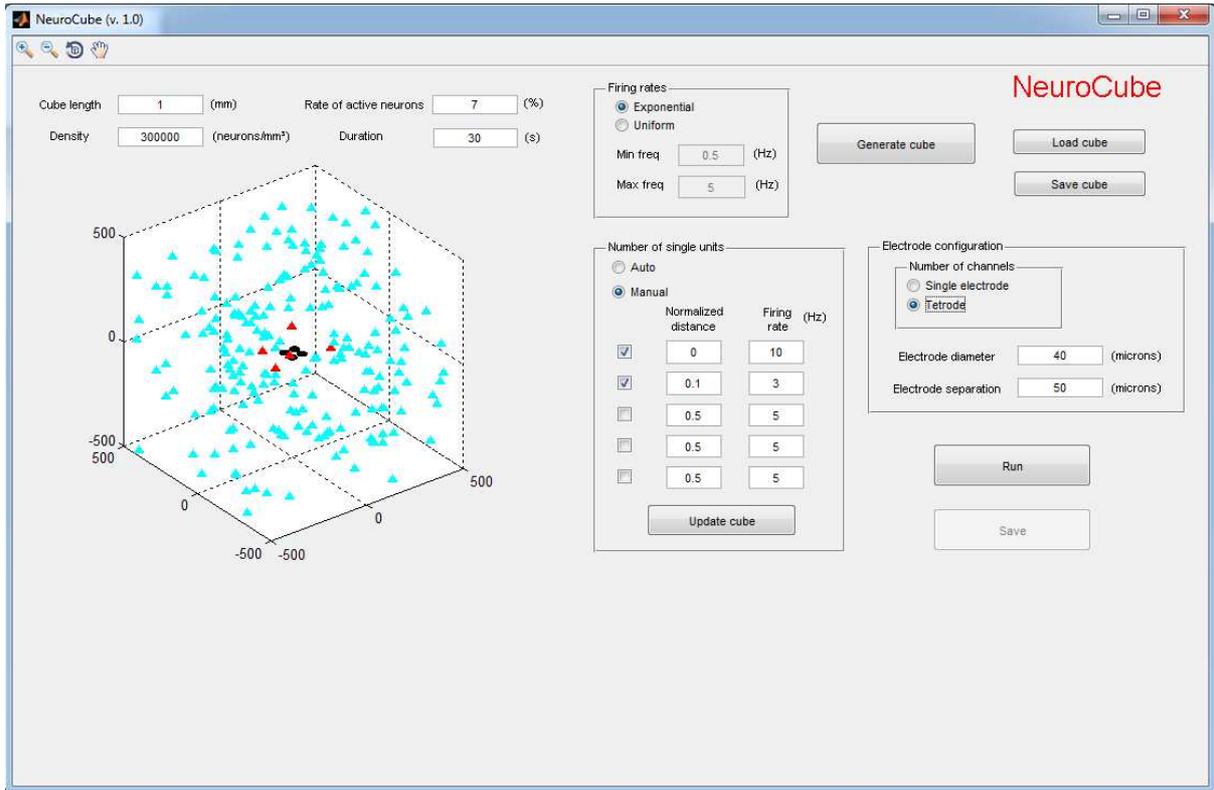


Figure 4

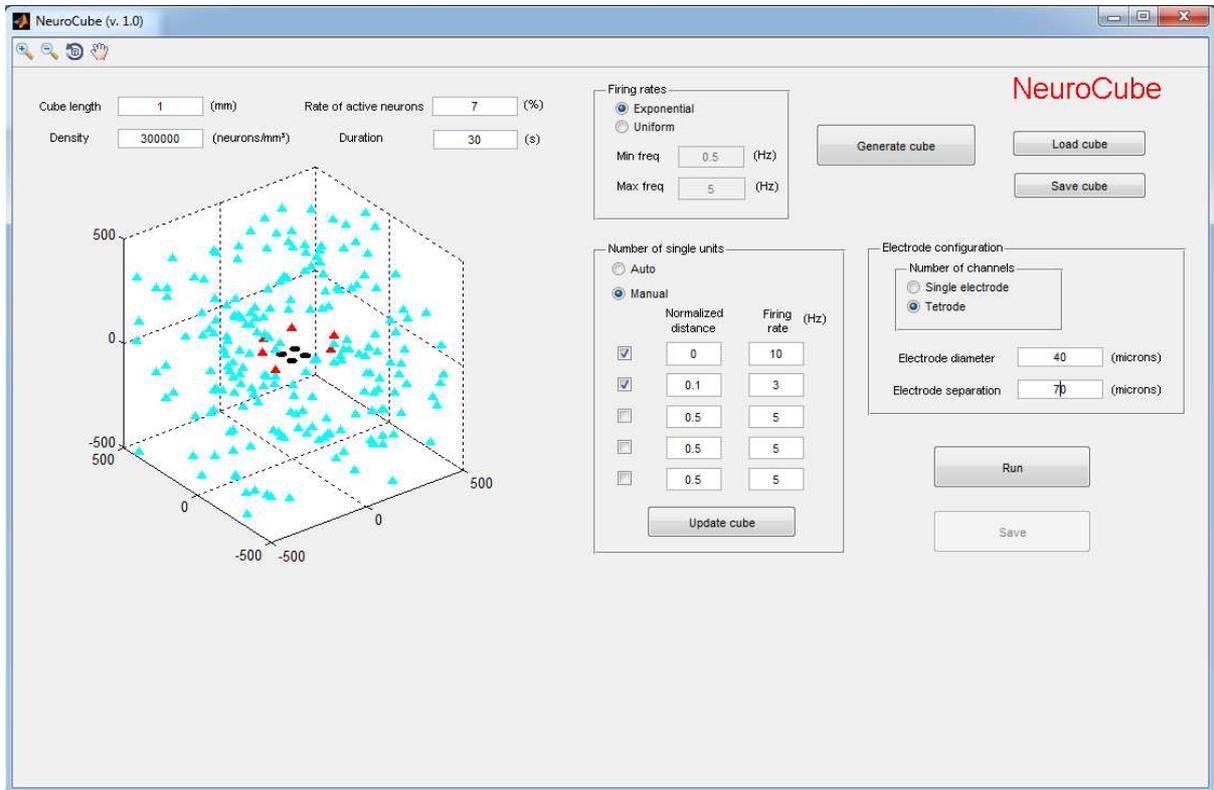


Figure 5

Running the simulation:

When both the distribution of neurons and the configuration of the electrode are set, you can click 'Run' and the simulation will start running. After that, in the bottom-left corner you will be able to follow the state of the simulation. First, it will calculate the spike shapes for close-by neurons (as shown in Figure 6 for the original distribution of neurons in Figure 2). After that, it will calculate the spike shapes for distant neurons (as shown in Figure 7). And finally, it calculates the spike times (as shown in Figure 8) and builds the final recording. When the simulation is ready, a section of 30 seconds of the continuous data is plotted as shown in Figure 9. After that, you can click 'Save' and a *.mat file will be generated. That file will include the continuous signal (or signals if more than one electrode is selected), the coordinates and identification of every neuron included in the cube, the spike times associated to each neuron and the spike shapes generated by the close-by neurons.

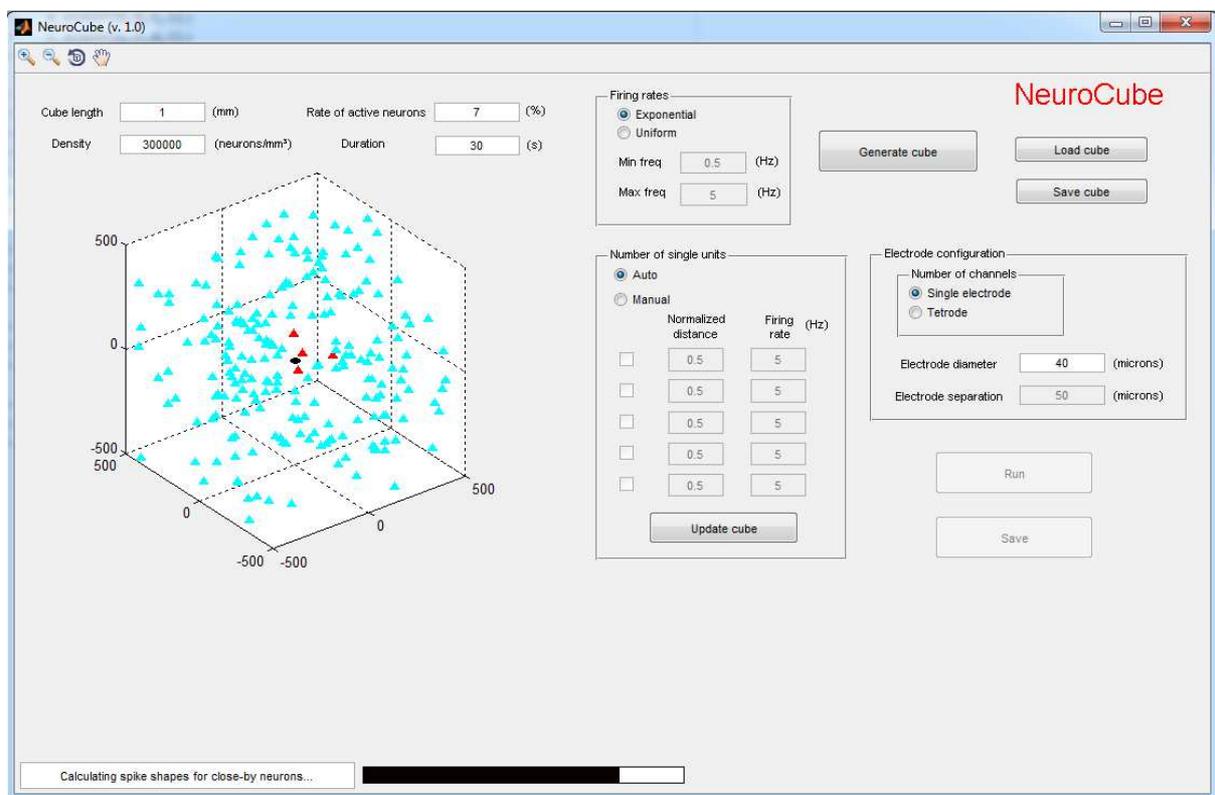


Figure 6

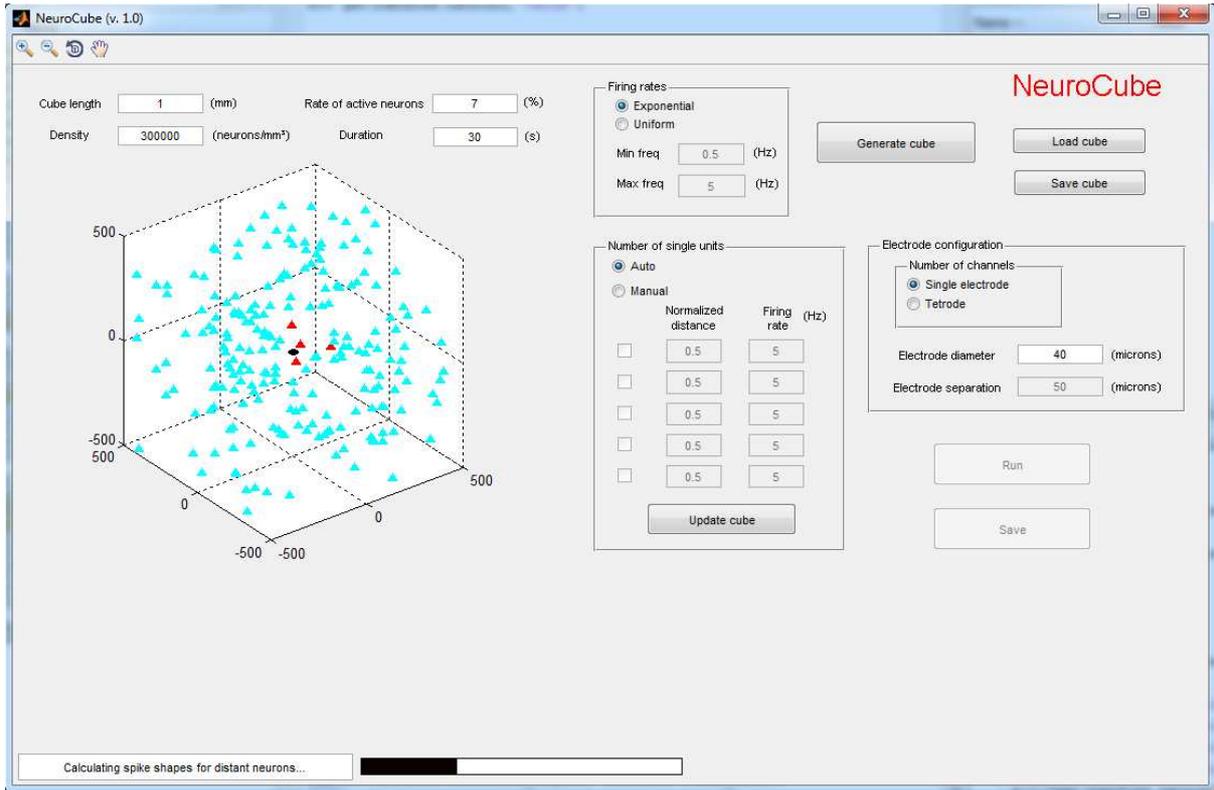


Figure 7

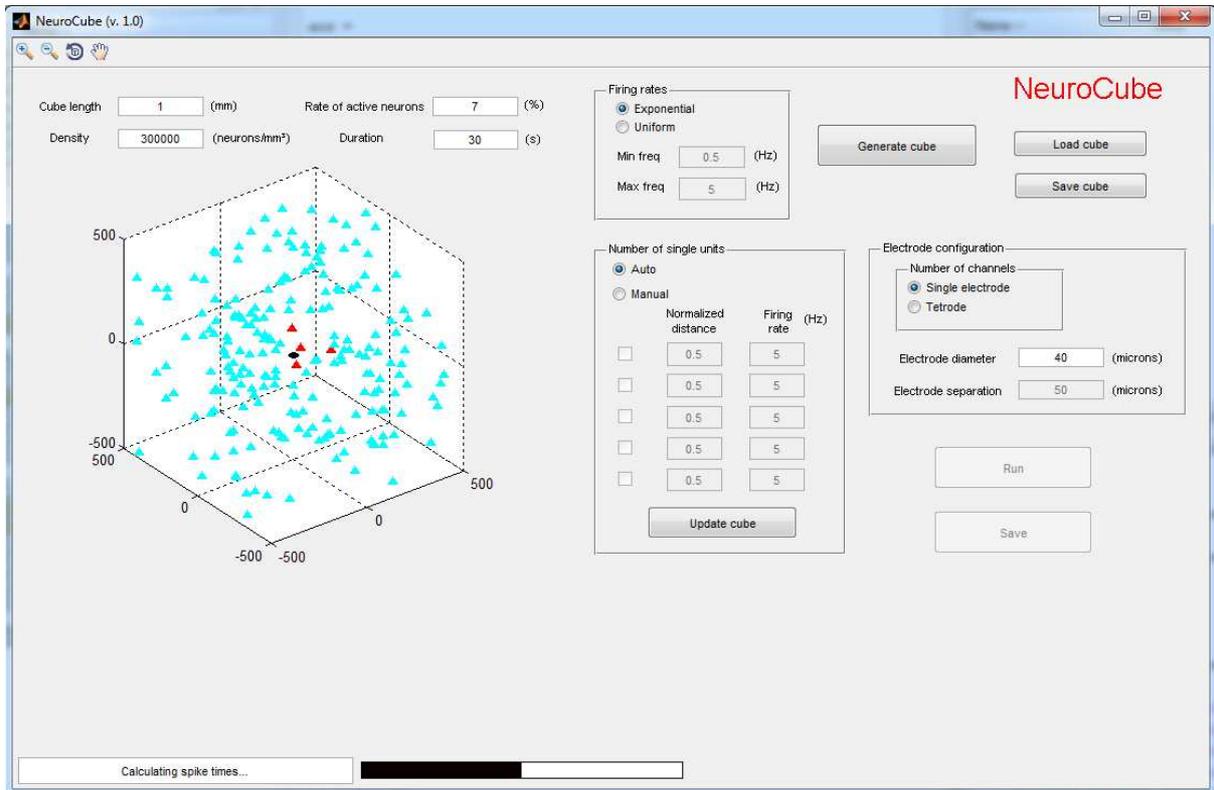


Figure 8

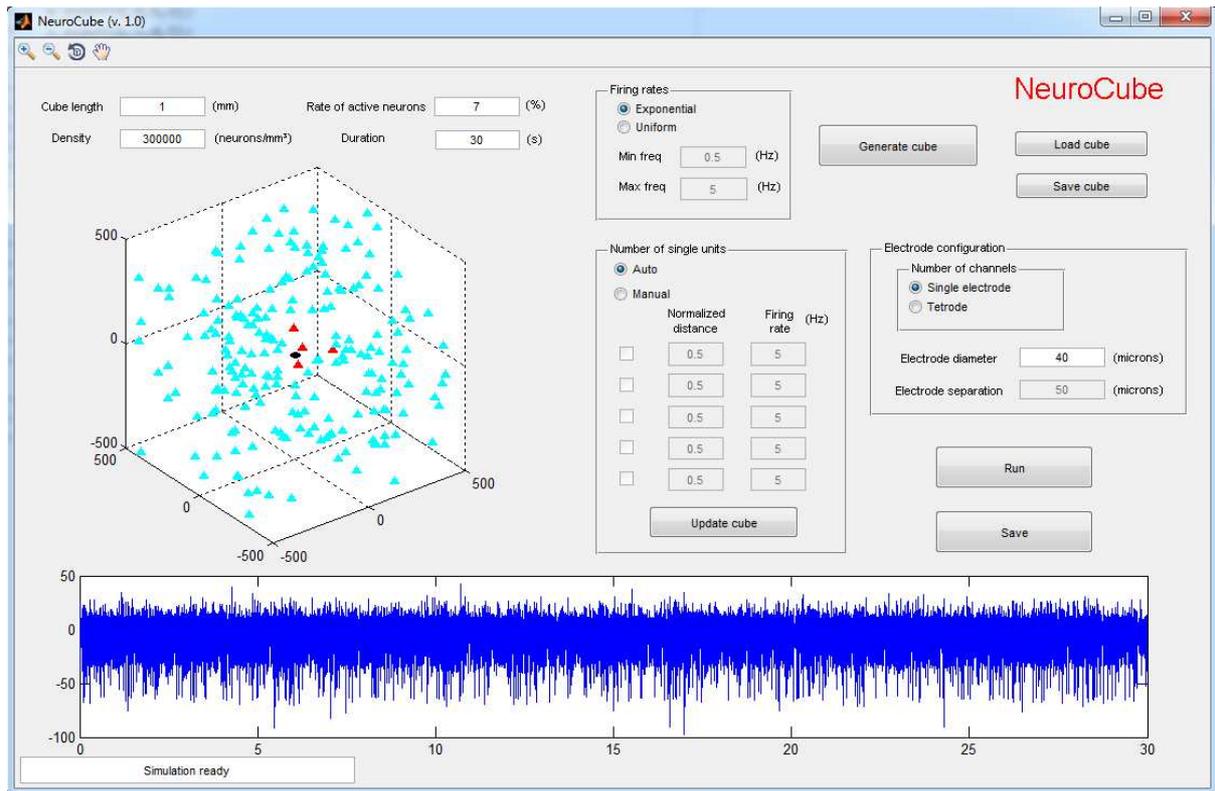


Figure 9

Internal parameters:

Apart from the parameters that can be set in the GUI, there are more advanced parameters that can be changed by the user inside the code.

- Parameters of the cube:
 - o `Par_cube.firing_rates_params`: it can be used to define the parameters of the distribution of firing rates for the neurons in the cube.
 - o `Par_cube.margin_electrode`: all the neurons whose soma is inside a sphere around the centre of the electrode are deleted. This parameter indicates the distance between the edge of the electrode and that sphere.
 - o `Par_cube.prop_plot`: percentage of neurons that are represented in the 3D plot.
- Parameters of the electrode:
 - o `Electrode.nchannels`: number of electrode channels.
 - o `Electrode.tetrode`: coordinates of the different electrodes when more than one is used.
- Parameters of the simulation:
 - o `Par_sim.sr`: sampling rate of the synthetic data.
 - o `Par_sim.refract_period`: refractory period in milliseconds.
 - o `Par_sim.dlim`: only the neurons inside a sphere with radius `dlim` around the electrode are considered close-by neurons and are simulated in detail.

- `Par_sim.dsingle`: when manual mode is selected, all the neurons inside a sphere with radius `dlim` around the electrode are deleted, so that only those single neurons inserted manually are present.
- `Par_sim.std_th_noise`: standard deviation of thermal noise.
- `Par_sim.grid_npoint`: Number of points in the grid used to calculate spike shapes across the electrode surface.