Multi compartmental Cerebellar Granule cell running on NSG.

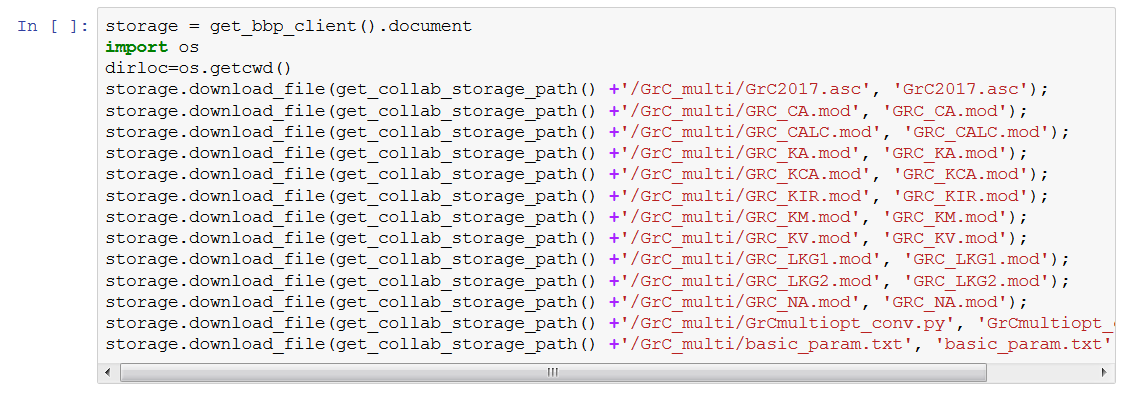
Multi compartmental Cerebellar Granule cell optimization, performed with BluePyOpt, direcly on the collab (Code and testing - Stefano Masoli PhD).

The ionic channels mechanisms, the passive properties and the morphology were taken and adapted from a previous published granule cell model (Diwakar et al., 2006) (<https://senselab.med.yale.edu/modeldb/showModel.cshtml?model=116835>) and used in the automatic reconstruction and validation of the granule cells, using the BBP Optimizer Framework (Masoli et al., 2017). Information about the generation of the morphology can be found in the paper.

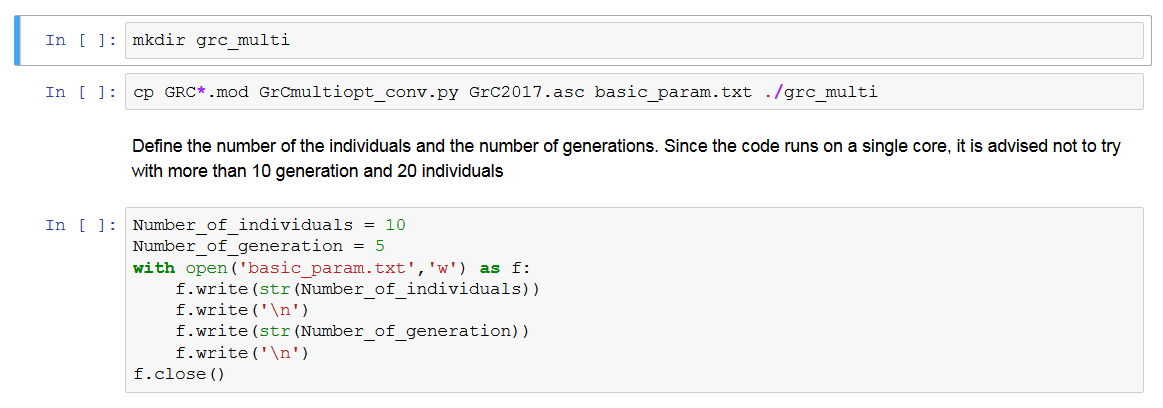
The code is commented, step by step, to define what is done in each ipython cell.

Note - Each notebook cell has two square bracket on the left.

1. When are empty, it means that the code was never started before.
2. When there is an asterisk [\*], it means that the code is running.
3. When there is a number, it means that the code was run. A progressive numbering scheme define the order in which the cells were selected.



* Fig1 – The optimization requires a certain amount of files. The morphology, the ionic channels, the python file, which is nearly identical to the one used for the granule cell optimization on the collab and a text file that contains the values for the numbers of individuals and generations.



* Fig2 – A requirement of NSG is that all files need to be placed into a directory, which later will be compressed into a zip file.

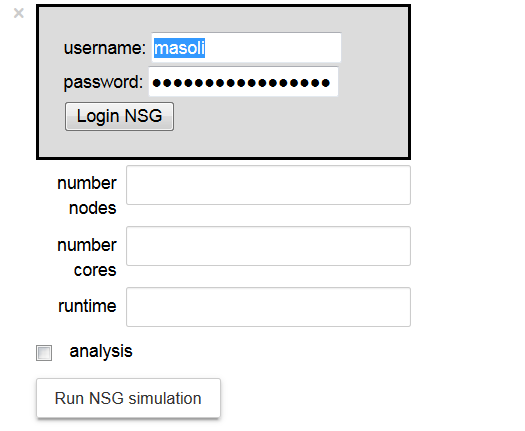
At this point it can be set the number of generations and individual. The two parameters will be saved in a text file used by the main python file.



* Fig3 – Compression of the files into the zip file.



Fig4a – The code in this part is in charge of a GUI.



* Fig4b – This is used to set the number of nodes (in this case only 1), of cores (again 1) and the runtime. The analysis checkbox is not functional. The procedure requires to login to NSG, with a valid account, that is not preset in the code itself. After pushing the run button, another button will appear to check the status of the queue and if the simulation is running.
* IMPORTANT. After sending a job to NSG, it cannot be:

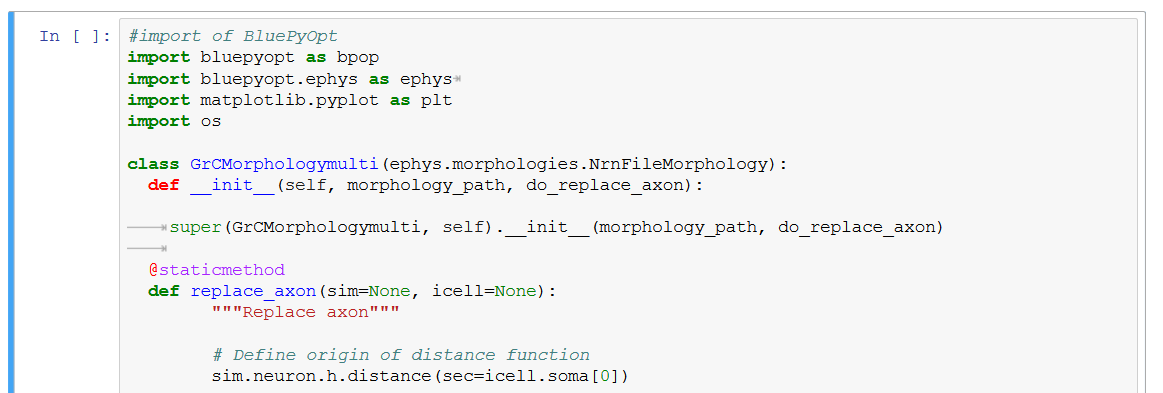
1) Closed the collab page

2) The page

3) The browser

4) If the connection is interrupted, the code cannot recover the results.

5) The page must remain always active to retrieve the results from NSG.



* Fig5 – The rest of the cells is a reduced version of the code used to run the optimization on the collab. It is used to plot the best individual obtained at the end of the optimization. Any modification to this code can destroy the entire process.