USER GUIDE

*BRAINCELL* *1.0.*

Brain cell *in* *silico*

© University College London, MIT licence

2023

**TABLE** **OF** **CONTENTS**

Table of Contents

[INTRODUCTION 4](#_Toc131151691)

[Key system and software requirements 4](#_Toc131151692)

[Basic version 4](#_Toc131151693)

[Full version 4](#_Toc131151694)

[The strategy of building the model: summary 5](#_Toc131151695)

[The outlines of experimental data or approximations required to create a realistic brain cell model. 6](#_Toc131151696)

[GETTING STARTED 6](#_Toc131151697)

[Installing and running BRAINCELL 6](#_Toc131151698)

[Setting up and launching 6](#_Toc131151699)

[File structure in Host computer (under Windows) 7](#_Toc131151700)

[Preparing BRAINCELL system files 7](#_Toc131151701)

[Astro/ Base. Setting up and running BRAINCELL: Astrocyte configuration. 8](#_Toc131151702)

[GENERATING COMPLETE ASTROCYTE MORPHOLOGY 8](#_Toc131151703)

[Generating / downloading astrocyte stem tree 9](#_Toc131151704)

[Generating astroglial morphology on the nanoscale 11](#_Toc131151705)

[Geometry of nanoscopic processes 11](#_Toc131151706)

[Populating astrocyte tree with nanoscopic processes 11](#_Toc131151707)

[Tissue-filling properties of astroglial morphology 12](#_Toc131151708)

[Nano Astro. Download previously created astrocyte morphology. 14](#_Toc131151709)

[Neuron / Base. Setting up and running BRAINCELL: Astrocyte module. 16](#_Toc131151710)

[GENERATING COMPLETE NEURON MORPHOLOGY 16](#_Toc131151711)

[Nano Neuron. Download previously created Neuron morphology. 17](#_Toc131151712)

[Nano geometry modification. 18](#_Toc131151713)

[The arrangement of spines. 18](#_Toc131151714)

[The spine head and neck geometry. 18](#_Toc131151715)

[Manage distribution mechanisms. 19](#_Toc131151716)

[Adjust the spatial distribution of mechanisms. 20](#_Toc131151717)

[Synapse distribution. 20](#_Toc131151718)

[Adjust the stochastic distribution of mechanisms. 23](#_Toc131151719)

[Synaptic manager upper part. 26](#_Toc131151720)

[Synaptic manager middle part 27](#_Toc131151721)

[Synaptic manager bottom part. 27](#_Toc131151722)

# INTRODUCTION

Welcome to the user manual for BRAINCELL! BRAINCELL is a simulation tool designed to evaluate the multi-scale morphology of astroglia and neurons and create realistic multi-compartmental biophysical models of brain cells. With BRAINCELL, you can explore these models using the NEURON/Python computational environment, which allows you to incorporate and test a wide range of biophysical, cellular mechanisms. The primary goal of using BRAINCELL is to assist with the mechanistic interpretation of experimental observations in brain cells.

This manual will guide you through the features and functionality of BRAINCELL, enabling you to use this powerful tool to enhance your understanding of brain cell behaviour.

## Key system and software requirements

The current version of BRAINCELL can be downloaded directly from

<https://github.com/LeonidSavtchenko/BrainCellNew>

The present User Manual is to be regularly updated. Its current version can be downloaded from the exact location.

## Basic version

Running BRAINCELL without full-scale simulations on Cluster requires:

1. Host computer must have MATLAB (2012 version or later, <https://uk.mathworks.com/products/matlab.html>) and NEURON (7.2 or later, [https://neuron.yale.edu/neuron/download)](https://neuron.yale.edu/neuron/download) installed under Windows 7-12.

## Full version

Simulating full extracellular ionic dynamics on top of other biophysical mechanisms is highly resource-consuming and should typically require:

1. Two computers: the Host computer operating under Windows and Worker (remote) computer cluster operating under Linux.

2. Basic preinstalled software: MPIC++ (Worker) [(https://www.open-mpi.org/software/ompi/v3.0/)](https://www.open-mpi.org/software/ompi/v3.0/), MATLAB not older than 2013 (Worker and Host) and NEURON 7.0 (Worker and Host) [https://neuron.yale.edu/neuron/download.](https://neuron.yale.edu/neuron/download)

3. Platform: Linux and Windows. Optional modes of operation: sequential and parallel (MPI) computing.

**NOTE**: Instead of the fully-fledged MATLAB version, the user can install MATLAB Runtime, which is a free package that can be downloaded from here: [https://www.mathworks.com/products/compiler/mcr/.](https://www.mathworks.com/products/compiler/mcr/)

## The strategy of building the model: summary

Creating a cell model using BRAINCELL can be a complex process, but here are some general instructions to get started:

1. Basic 3D cell morphology. Go to The NeuronMorpho website (https://neuromorpho.org) and search for the specific type of brain cell you want to model. Once you have found the cell you want, download its 3D structure in the appropriate file format (such as SWC or OBJ).
2. Open a BRAINCELL and import the 3D structure file.
3. Adjust the scale of the model to the appropriate size for your needs. This may involve resizing, repositioning, or rotating the model to match your desired dimensions.
4. Once you have created a basic 3D model of the brain cell, you can start adding nanostructures to the model using either an experiment or computer simulation.
5. If you want to add nanostructures to astrocyte model using an experiment, you will need to use specialized software Astro in MATLAB to manipulate at the nanoscale level.

<https://github.com/LeonidSavtchenko/Astro>

1. Alternatively, you can use computer simulations to add nanostructures to the 3D cell. These simulations can help you to understand how the nanostructures interact with the brain cell and how they affect its function.
2. Once you have added the nanostructures to the model, you can use the 3D modeling software to visualize the changes and understand how they affect the overall structure and function of the brain cell.
3. Finally, you can refine and optimize the model as necessary to achieve your desired level of accuracy and detail.
4. Overall, creating a model of a brain cell using BRAINCELL is a complex process that requires both specialized software and expertise in both 3D modeling and nanoscale science. By following these instructions and utilizing the appropriate tools and techniques, however, you can create a highly accurate and detailed model of a brain cell that can be used for a wide range of scientific and educational purposes.

5

# The outlines of experimental data or approximations required to create a realistic brain cell model.

Here are the details:

1. It is preferable to have a 3D reconstructed tree of main cell processes that can be imported from ***https://neuromorpho.org*** in any format. Alternatively, an artificially generated cell arbour can be used, with the branching pattern and branch diameters representing the average (typical) cell from the population of interest.
2. Astrocyte nanostructures are essential, and a sample (20-50) of nanoscopic astroglial processes that have been reconstructed using 3D (serial-section) EM is necessary. The sample should have rendered surface coordinates and will be used to obtain statistical properties of the ultrathin processes to be generated in the model.
3. Neuron nanostructures are also needed, and "BRAINCELL" can automatically generate synaptic spines with different distribution densities, geometries, and contacts with synapses. Synapses can be located both on the spines and directly on the dendrites. The user can select all parameters and also control the geometry complexity of the spines.
4. The average tissue volume fraction occupied by astroglia and neurons, as distributed radially from the soma to the cell edges, is also required. This data set can be obtained from two-photon excitation measurements in situ (or from published data).
5. It is necessary to have the mean membrane surface density and surface-to-volume fraction values, which can be obtained from 3D reconstructions of nanoscopic processes.
6. The characteristic I-V curve for the cell of interest, obtained through somatic patch-clamp with square-pulse current injections, is essential. Other available functional data, such as electrical responses to neurotransmitter uncaging or changes in extracellular ion and intracellular calcium wave speed, are optional but helpful.

# GETTING STARTED

## Installing and running BRAINCELL

### Setting up and launching

The latest installation version can be downloaded from [(https://github.com/LeonidSavtchenko/BrainCellNew)](https://github.com/LeonidSavtchenko/BrainCellNew).

On the website front page (Fig. \*a), to download BrainCellNew, press the green key **'Clone** **or** **download'** and save Download.Zip at any place of the personal computer. Then the archive must be opened and its content saved on the Host computer (Windows/MacOs) keeping the folder structure as described (Fig. \*b).

To get started with BRAINCELL, the Host computer must have NEURON (7.0 or later) installed.

a b



**Figure 1. Screenshot of the BRAINCELL download GitHub page (a), and folder structure or BRAINCELL 1.0 on the Host computer (b).**

## File structure in Host computer (under Windows)

This section explains initial steps to launch and run the NEURON environment adapted for brain cell modelling, on the Host computer under Windows.

### Preparing BRAINCELL system files

1. Set the path to NEURON on the Host computer using the batch file ***INIT.bat***; by default, it is set as *NEURON\_HOME\_WIN="C:\nrn\bin\neuron.exe"*

2. Execute the ***init.hoc*** file located in the host computer directory …*\init.hoc*' or use the button 'NEURON simulations' from the start menu panel (Figs. 2-3).

3. Activate  ***build\_mechs.bat*** to trigger compilation of the NEURON \*.mod files automatically.



**Figure** **2.** **Introductory** **menu.**

**Introductory** **menu**: Simulation Cell Configuration

The menu in the simulation interface provides the option to choose between two main cell types: "**Astrocyte**" and "**Neuron**." Each cell type has two configuration options: "**Base**" or "**Nano**."

Selecting the "**Base**" configuration option allows users to build a cell with different geometries. Users can modify the geometry of the cell as needed.

Alternatively, selecting the "**Nano**" configuration option allows users to load a previously created cell that contains nanostructures. It is important to note that changing the loaded cell’s geometry is impossible. This option is designed to save time for users who do not want to create a new cell every time they use the simulation.

The user can choose from 4 loading scenarios.

1. [**Astro / Base**](#BaseAstro)**.**
2. [**Astro / Nano**](#NanoAstro)**.**
3. [**Neuron / Base**](#BaseNeuron)**.**
4. [**Neuron / Nano**](#NanoNeuron)**.**

# Astro/ Base. Setting up and running BRAINCELL: Astrocyte configuration.

## GENERATING COMPLETE ASTROCYTE MORPHOLOGY

a b



c

**Figure 3.** **Control** **windows** **initiated** **by** **launching** **NEURON** **in** **the** **ASTRO** **environment,** **as** **detailed** **in** **the** **text.**

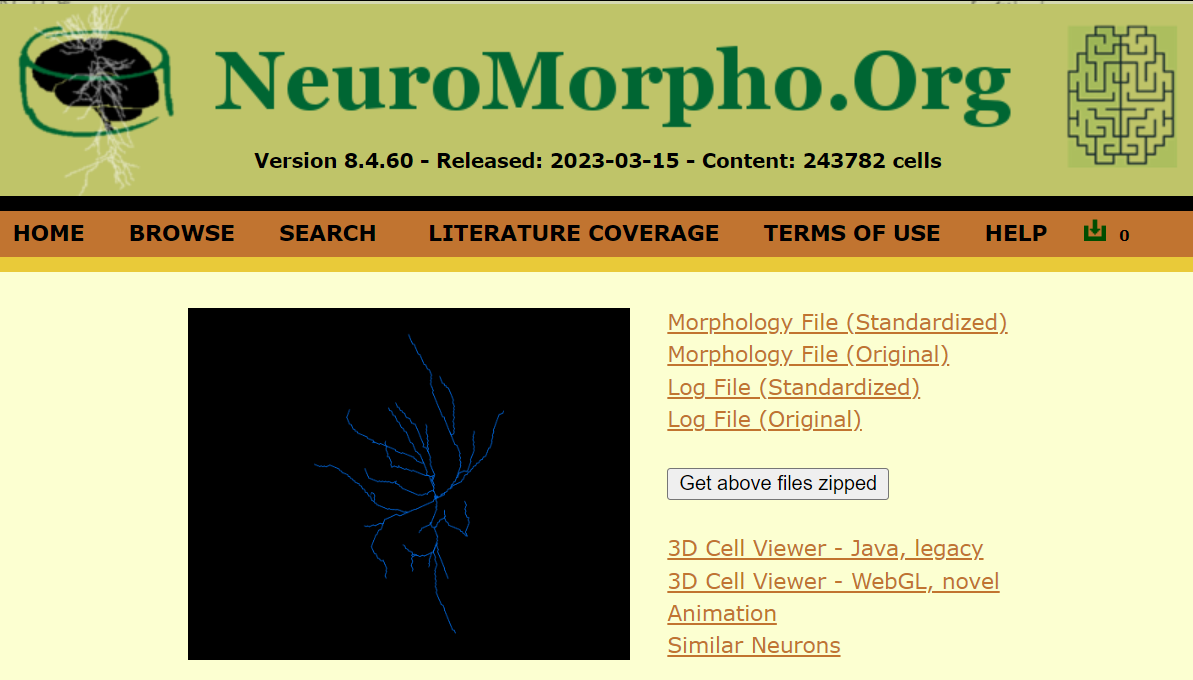
4. The file ***init.hoc*** opens three windows: System window (***cmd.exe***) (a), a window panel to define the gross astrocyte geometry (stem tree,b), and a menu panel to set the density for higher orders of nanoscopic processes ('**Leaf** **number'** c) and number of nanostructures per dendtire (**‘Max number of stalks’**, C).

## Generating / downloading astrocyte stem tree

To design a new astrocyte model, the user has to define the basic structure of dendritic tree using three different options:

**Option** **1**: Click on "Select Library Stem Tree".

Choose to import 3D files in general zip format from the database http://neuromorpho.com. You can upload as many files as you like, but for convenience, it is recommended that you place them in the directory .../Geometry/Astrocyte/New Style.

**Figure. Import 3D structure. A) Web page of Nanostructure, B) Operational** **window** **panels** **pertinent** **to** **choose** **astroglial** **morphology.**

Alternatively, select a file from the in-house directory in SCW or HOC format.

If you have selected a specific file with a 3D structure, you will proceed to the next window to view this 3D geometry.

If you like the structure, click the "Use this" button to proceed. If not, select another structure using the "Import another" button.

Once you have selected the desired cell, you can proceed to the next option.



**Figure.** **Operational** **window** **panels** **pertinent** **to** **the** **creation** **of** **gross** **astroglial** **morphology.**

where OriginalDendrite sets the number of branches (dendrites in NEURON terminology) on the stem tree. The database ***NeuroMorpho.org*** can be used as a guide to the ASTRO-compatible file format. Upon selection, a window panel is activated displaying the selected stem tree (Fig. \*b).

**Option** **2:** **'Select** **stem** **tree** **with** **endfoot’** is similar to Option 1, but with the endfoot structure, which is stochastically generated (Fig. \*c). Here, an additional window panel is activated (Fig. \*c, left), providing a menu to set the morphology of the main and the secondary endfoot branches, and to set up the local biophysical mechanisms.

**Option** **3:** **'Select** **reconstructed** **stem** **tree’** loads the 3D-reconstructed stem tree file. An example in ***RealAstrocyteSkeleton1.hoc*** (the directory …*/Geometry*) shows the reconstructed stem tree of the CA1 astrocyte using the Vaa3D software (Allen Institute, available from [http://www.alleninstitute.org/what-we-do/brain-science/research/products-tools/vaa3d/)](http://www.alleninstitute.org/what-we-do/brain-science/research/products-tools/vaa3d/). This option also prompts an additional window panel (Fig. 17d), providing setting for geometrical scaling and the centring of the astrocyte structure at the coordinate origin (to facilitate positioning of selected cell compartments). The corresponding menu buttons thus include '**X-Y** **scale** **(pixel/μm)**', '**Z** **scale** **(pixel** **μm)**', and '**X-Y** s**hift** **(μm)**'. This window will disappear after any parameter change.

**NOTE:** The user must upload the cell stem tree geometry before initiating any further design of the model

## Generating astroglial morphology on the nanoscale

### Geometry of nanoscopic processes

Once the stem tree has been downloaded, the next stage is the nanostructure of astrocyte. The user has two options.

**Option** **1:** To download the default nanostructure prepared in advance. Pressing the button **'Diameter** **distribution** **for** **nano-geometry'** (Fig. \*b) prompts the use to download a file with the statistics of process diameters produced by the '**Nano** **(Geometry)'** module from the sampled 3D-reconstructed astroglial processes (see above). By default, this option downloads the file ***testshape.dat\_radii\_dist.txt*** (characterising astroglial processes in CA1 *stratum* *radiatum*). After that, the user presses the '**Start** **Astro’** button (Fig. 16b).

**Option** **2:** To press '**Start** **Astro’** button, in which case ASTRO generates nanoscopic processes automatically using the built-in tools.

In both cases, the user can repeatedly adjust key morphometric features of the generated nanostructures. See further details in the chapter **Simulating** **Astrocyte** **Physiology**.

### Populating astrocyte tree with nanoscopic processes

The '**Start** **Astro’** button (Fig. 16b) prompts the main window panel '**Repertoire** **of** **computation**' (Fig. 18), which is key to the modelling of complete astrocyte morphology, as described in the sections below.

a b c



**Figure** **18.** **Main** **window** **of** **ASTRO.** **a,** Control panel providing detailed settings of astrocyte geometry including a gap junction feature (bottom). **b,** Simulated variable mapped onto astrocyte morphology (top; membrane voltage shown), with selected digital output plot (bottom). **c,** Computational scenarios with parameter setting.

Panel '**Leaf** **Geometry'** (Fig. 18a, top) provides an option to set up the distribution of cylindrical compartments (leaves) of nanoscopic processes as evenly random (with lower and upper limits) when the experimental statistics on 3D reconstructed processes are not available.

NOTE: This section is to be ignored when the latter have already been loaded (see previous section).

Panel '**Stalk** **Geometry'** (Fig. 18a, middle) sets upper and lower limits for the uniform distribution of transitional cylinders of nano geometry. These parameters determine how densely the tissue is to be filled with nanoscopic astroglial processes.

Panel **'Specific** **membrane** **conductance'** sets this value at the button '**Gm** **(mS/cm2**)', which takes into account all exposed surfaces of the cylindrical compartments. Resting potential of the current is -85 mV. This parameter is defined on the built-in NEURON panel “Distributed mechanism”.

Panel **'Dendritic** **Geometry'** (Fig. 18a, bottom) currently includes **'Branch** **diameter** **scaling',** which sets the scaling coefficient for the stem tree branch diameters as a function of distance from the soma, according to the average experimental trend. The empirically established formula for the branch diameter *d* is *d*~(*S*(*r+1*))-1/2 where **'scalingDiam'** value S and *r* is the distance to the soma.

NOTE: This panel has to be ignored if a 3D-reconstructed stem tree has been uploaded.

Panel **'Gap** **junctions'** is explained in the **Gap** **Junctions** section below (chapter **Simulating** **Astroglial** **Function**).

### Tissue-filling properties of astroglial morphology

The tissue volume-filling properties and the surface-to-volume ratios of the nanoscopic processes will be determined by the shapes and the effective density of simulated nanoscopic processes, as described in the previous section. Tissue volume filling and other geometry features of the model can be monitored by pressing **'Geometrical** **parameters'** key (Fig. 18c, top): this opens several window panels displaying various parameters of the modelled cell geometry (Fig. 19). The displayed data are automatically saved to the file ***…\neuronSims\Text*** ***results\VolumFraction.txt***.



**Figure** **19.** **Window** **panels** **providing** **readout** **of** **the** **volumetric** **characteristics** **for** **modelled** **astroglia** **(launched** **by** **'Geometrical** **parameters'** **button).** From top left: surface-to-volume ratio distribution, tissue volume fraction, total cell volume (cumulative value with the distance from the soma), total cell surface area, diameters of main processes.

The morphometric characteristics of the simulated astrocyte (Fig. 19) are to be compared with the corresponding empirical data obtained using 3D EM reconstructions and two-photon excitation imaging data for the astroglia of interest (Fig. 1). The user is free to evaluate the mismatch and adjust the density of nanoscopic processes (using '**Stalk** **Geometry'** and '**Dendritic** **Geometry'** options where relevant; Fig. 18) correspondingly, until an acceptable match is produced. The windows depicting critical geometrical parameters (Fig. 19) can be viewed at any time during modelling.

At the end of this stage, the modelled astroglial morphology is complete (see 'FRAP experiments' below for further subtle morphological adjustments). The user can begin to simulate various functions of astroglia while also implementing a variety of membrane and intracellular biophysical mechanisms, as briefly explained in the sections below.

**Generating** **and** **exploring** **pre-determined** **astrocyte** **models**

There are currently two pre-set models of the astroglia stem tree (main processes): one is obtained from a 3D reconstructed cell, stored in file ***GeometryAstrocyteCA1.hoc***, and the other is a 'typical' CA1 astrocyte (i.e. the cell whose macroscopic and nanoscopic features represent average values over a sample of CA1 astrocytes), stored in file ***AstrocyteBasicGeometry.hoc***. The pre-set

24

nanoscopic geometry representing the main features of nanoscopic astroglial processes in area CA1 (obtained from 3D EM reconstructions) is stored in file ***testshape.dat\_radii\_dist.txt***.

To download and work with one of the pre-set astrocyte models:

1. In the main NEURON menu (Fig. 16), set **'Leaf** **number'** option (Fig. 16c) up to 50. 2. Press **'Select** **basic** **geometry** **uploaded...'** menu button (Fig. 16b).

3. In the pop-up menu go to the *.../Geometry* and select ***GeometryAstrocyteCA1.hoc*** for a 'real CA1 astrocyte' geometry or ***AstrocyteBasicGeometry.hoc*** for 'typical CA1 astrocyte' geometry.

4. Press **'Select** **diameter** **distribution** **for** **the** **Nano** **geometry'** menu button (Fig. 16b). 5. In the pop-up Load menu select ***testshape.dat\_radii\_dist.txt*** file***.***

6. Press **'Start** **program'** (Fig. 16b).

The program will prompt the main control panel (**'Repertoire** **of** **computation'**, Fig. 18) displaying the newly-generated multi-scale model of an astrocyte (Fig. 18b).

Subsequent versions of ASTRO will include additional pre-set models of the astrocyte tree stem and its nanoscopic geometries.

# Nano Astro. Download previously created astrocyte morphology.



Combination Key "Astro + Nano" to create a panel. Once the panel appears, you can proceed with the following steps and open NEURON Basic Panel:

The NEURON basic panel will allow you to locate the previously prepared astrocyte with Nano structure.

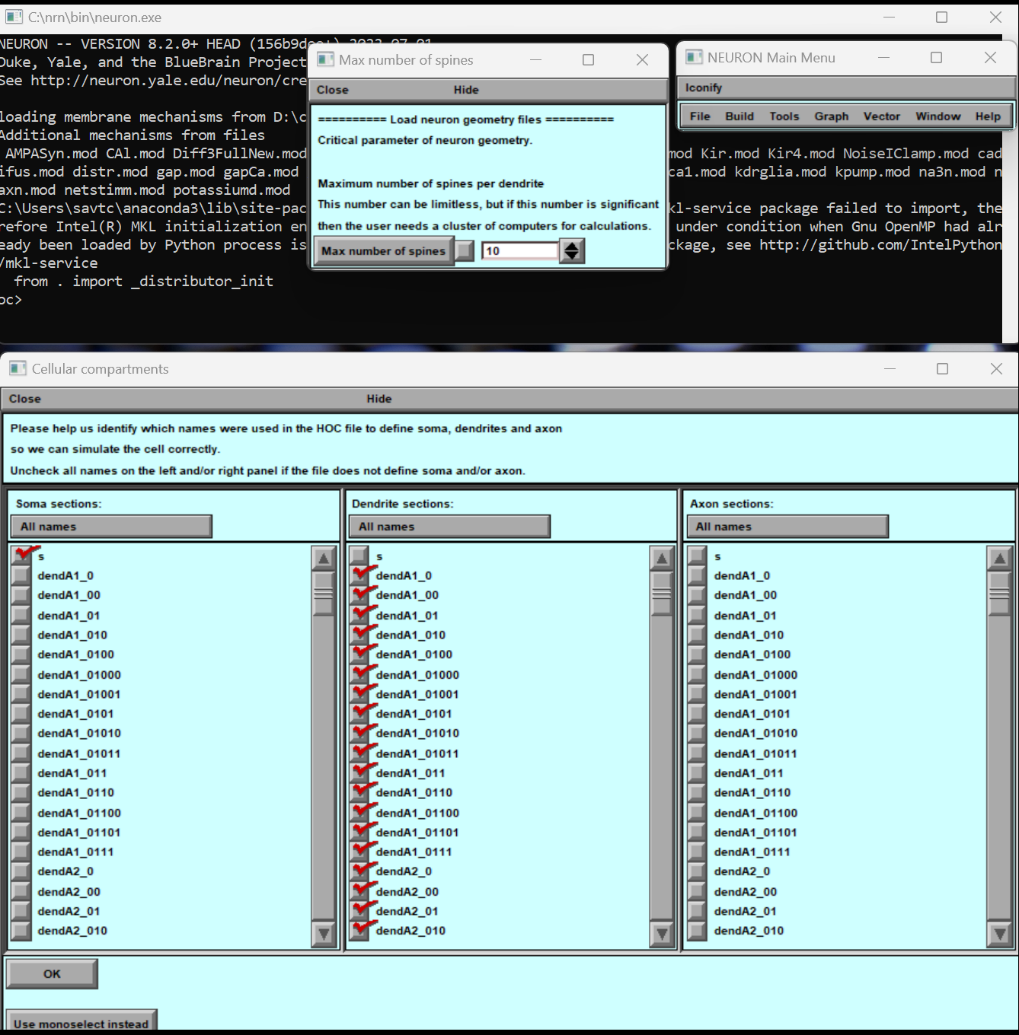
To proceed with simulation and management of biophysical mechanisms, select the astrocyte with Nano structure. This will take you to a new option for simulation and management..

At this stage, you will be able to simulate and manage the biophysical mechanisms of the selected astrocyte with Nano structure. Please note that you will not be able to change the geometry at this stage.

**Figure**. **Operational** **window** **panels** **uploaded previously created**  **astroglial** **morphology.**

# Neuron / Base. Setting up and running BRAINCELL: Astrocyte module.

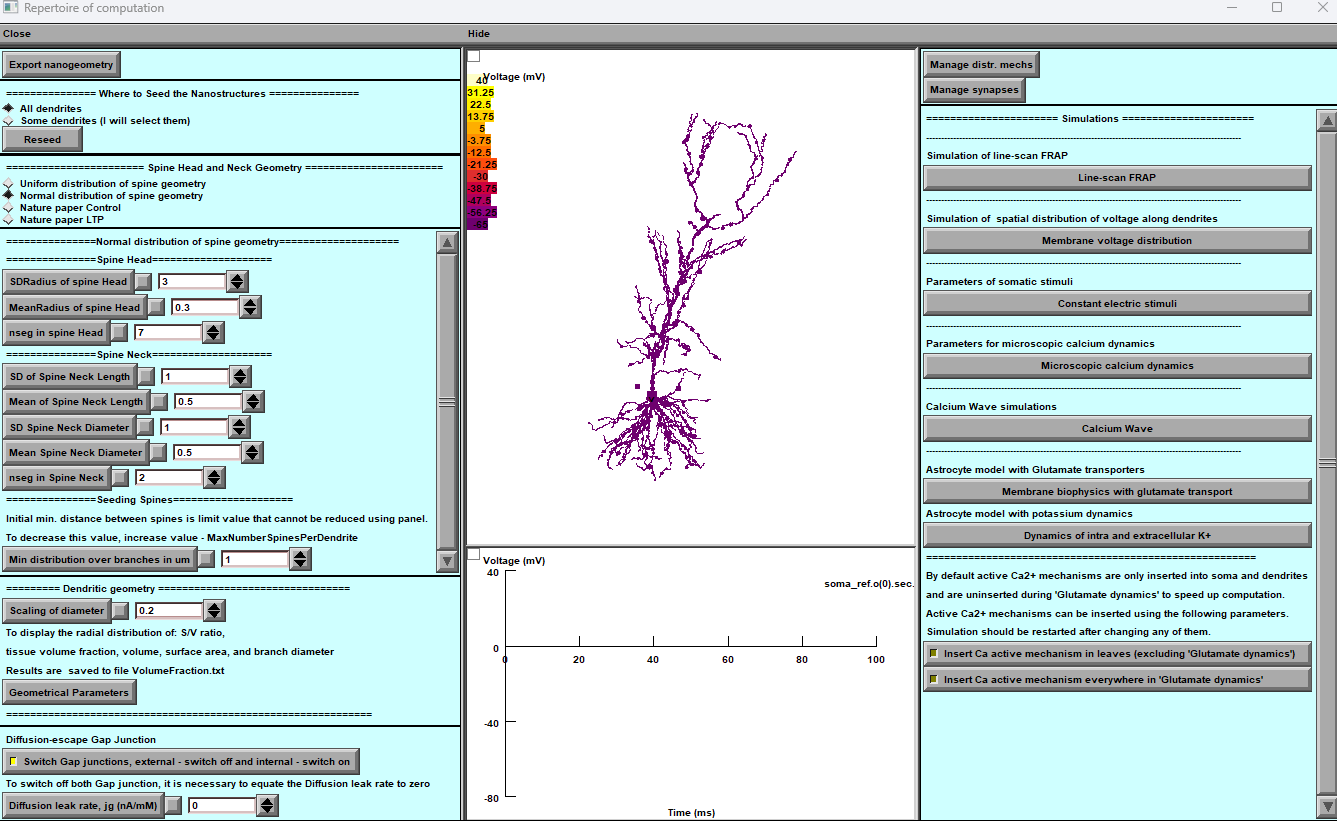
## GENERATING COMPLETE NEURON MORPHOLOGY

This part describes a BRAINCELL part for constructing 3D model of neuron, which is different from astrocytes. When loading a 3D neuron, users are presented with two windows that allow them to determine key features of the neuron's structure. The first window enables the user to set the maximum number of spines on the dendrite, which is a critical function. However, this number can be changed in the future, along with the geometry of the spines. The second window is much larger and allows users to determine the soma, dendrites, and axon of the neuron. If the axon is not defined initially, the program will generate a basic axon that can be modified later.

It should be noted that distinct components of a neuron cannot share the same name. The software will give an error message right away.

By selecting the given configuration and clicking OK, the programme renders the given form and allows the user to use it for further neuron model assembly or to try another form from the database.

If the user decides to try another neuron morphology, the procedure must be repeated, as previously described. If the user decides to stop at this one, the new window will appear by pressing the "Use this one" button.



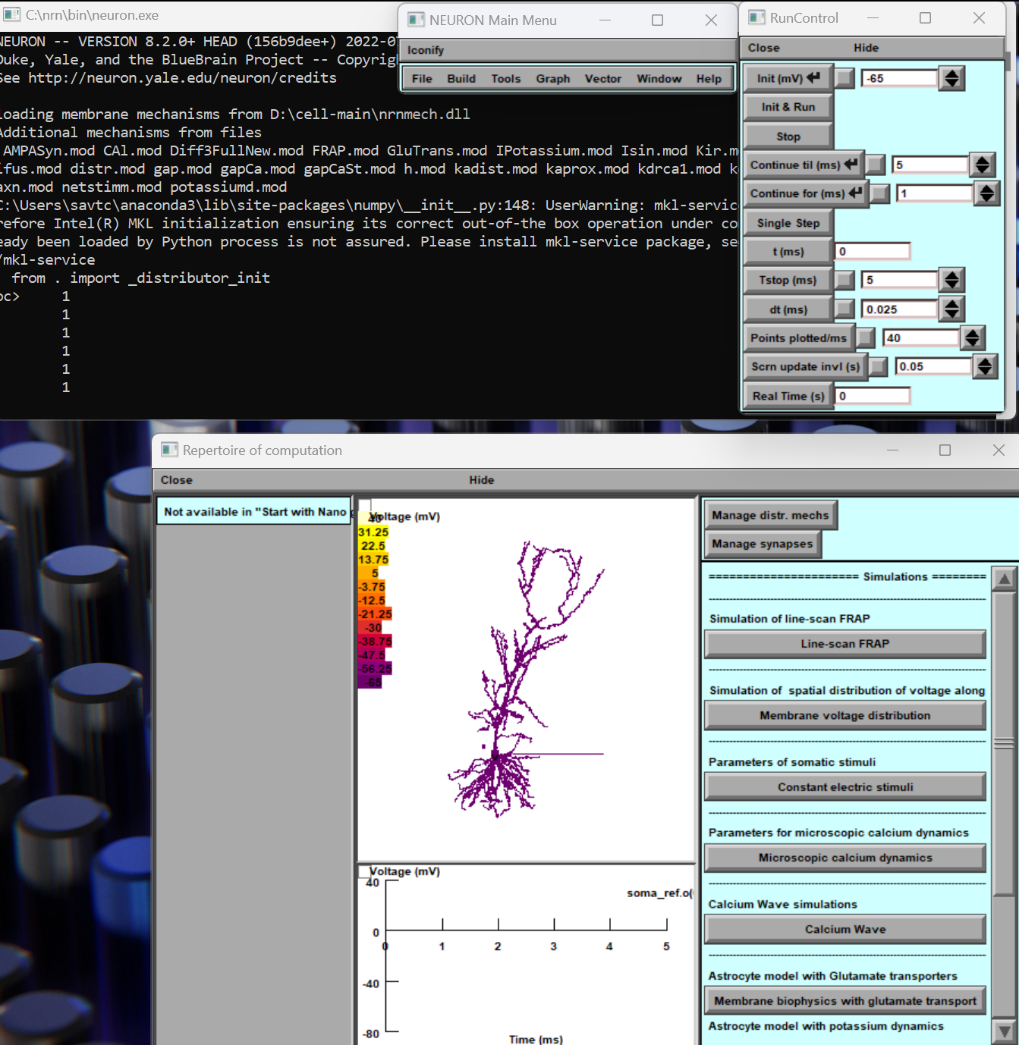
**Fig. Main windows for neuron simulation**

The main window of the tool provides several options to simulate neuron behavior effectively. Below are the key features of the main window:

1. [**Nano geometry modification**](#NanoGeometry): Users can add and modify spines' nano geometry to study the impact of structural changes on neuron behavior.
2. [**Biophysical mechanisms addition**](#BiophysicalMechanisms): The tool enables users to add and change various biophysical mechanisms to neurons to investigate their impact on neuron behavior.
3. [**Synapse distribution**](#SynapseDistribution): The tool facilitates the addition and distribution of different types of synapses on the dendritic tree, enabling the study of the neuron's connectivity and behavior.
4. [**Simulation modes**](#SimulationModes): Users can choose from a variety of simulation modes to simulate the neuron's behavior accurately. The simulation modes include voltage clamp, current clamp, and dynamic clamp.

The neuron simulation tool's main window provides the necessary features to simulate and analyze neuron behavior effectively.

## Nano Neuron. Download previously created Neuron morphology.

Combination Key "Neuron + Nano" to create a panel. Once the panel appears, you can proceed with the following steps and open NEURON Basic Panel:

The NEURON basic panel will allow you to locate the previously prepared Neuron with Nano structure.

To proceed with simulation and management of biophysical mechanisms, select the neuron with Nano structure. This will take you to a new option for simulation and management..

At this stage, you will be able to simulate and manage the biophysical mechanisms of the selected neuron with Nano structure.

***Please note that you will not be able to change the geometry at this stage.***

## Nano geometry modification.

Users can add and modify spines' nano geometry to study the impact of structural changes on neuron behavior.

### The arrangement of spines.

The "Where to Seed the Nanostructures" panel allows the user to distribute the spines throughout the neuron as well as on individual dendrites.

If the user selects, distribute the spines on individual dendrites, a window with the 3D structure of the neuron will appear, and the user can use the mouse to specify only those dendrites where the spines should be distributed.

### The spine head and neck geometry.

This part proposes a tool that allows users to analyze the geometry distribution of dendritic spines. The tool offers two distribution options: normal and uniform, which the user can select. The user can set numerical parameters for each distribution via a window upon selection. To make it convenient for the user, the tool provides pre-established distribution types of spine geometry (published in Tønnesen, J., Katona, G., Rózsa, B. et al. Spine neck plasticity regulates compartmentalization of synapses. Nat Neurosci 17, 678–685 (2014). https://doi.org/10.1038/nn.3682), which users can click on to obtain experiment-measured parameters.

Another crucial factor in determining spines is their complexity, which affects calculation speed. The number of segments that make up the spines determines their complexity, and this parameter can be adjusted with a minimum of two and no maximum limit.

The tool also considers the minimum distance between synapses in a dendritic tree as an essential parameter for synapse distribution. The shorter the distance, the more spines on the tree. However, this parameter is stochastic, meaning that the tool considers the element of chance in its calculations.

## Manage distribution mechanisms.

The tool enables users to add and change various biophysical mechanisms to neurons to investigate their impact on neuron behavior.

**Upon clicking** the "**Manage the distance of Mechanisms**" button located in the upper right corner (**Figure.** [**Main Window**](#MainWindowNeuronsimulation)), the user shall be presented with two significant panels, as depicted in the (**Figure**). The first window (A) enables the insertion and removal of mechanisms into various pre-defined regions of the brain cell. Within this window, all the mechanisms loaded into the "Mechanisms" folder, as well as the previously constructed neuron components, are displayed. The window comprises two modes, the initial mode illustrates the mechanisms present in each segment of the neuron model. The secondary mode, toggled via a button situated at the bottom, displays the location of each mechanism within the neuron segment. The usage of these windows is straightforward, whereby the user selects the required mechanisms for the neuron segment by ticking the corresponding checkboxes.



**Figure. Panels for dealing with biophysical mechanisms. A) control panel B) Panel for insert or remove mechanisms.**

The second window (Fig. A) serves a crucial role in facilitating cell-part operations. Through this window, users can subgroup, merge, or rename the different parts of a cell. Additionally, the second section of the window provides a means of interacting with the biophysical mechanisms themselves.

Within this section, users can access several panels that enable them to manipulate the mechanisms in various ways. These include the ability to insert or remove mechanisms, adjust the spatial distribution of mechanisms, and review mechanisms that display spatial inhomogeneity. Moreover, users can analyze stochastic mechanisms through this window.

### Adjust the spatial distribution of mechanisms.

The main panel (**Figure**) to adjust spatial distribution of mechanism across any part of cell allows the user different option to define the mathematical formula for the spatial distribution.



## Synapse distribution.

The tool facilitates the addition and distribution of different types of synapses on the dendritic tree, enabling the study of the neuron's connectivity and behavior.

**Figure. The main panel to edit mechanisms**.

The main window is designed to work seamlessly with any mechanisms built into different parts of neurons. Each piece of the neuron is located in its window within the main interface. Here, you can find all the biophysical mechanisms included in each part.

To begin editing any mechanism, click on it. This will bring up a menu where you can select what you want to edit. The menu includes options for global variables, state variables, and parameters. You can access another window that provides more detailed information about the selected variable by clicking on any of these options.

In the new window, you'll see the current value of the variable if it's spatially homogeneous. If you want to make it spatially inhomogeneous, click the "**Define as a function of distance**" bar. Doing this will open a new window called the Heterogeneity Editor.

In the **Heterogeneity Editor**, you can define the variable as a function of distance. This allows you to customize the behaviour of the variable based on the spatial location. Once you've described it, you can save your changes and edit other mechanisms.

Spatial Inhomogeneity of Biophysical Mechanisms Editor**.**

This editor allows you to work with segments of neuron parts in order to determine the heterogeneity of mechanisms.

The editor's main window is divided into three parts. The upper part of the panel is where you can work with segments of neuron parts. This is a crucial part of the software because heterogeneity is determined not in each physical coordinate of the neuron but in each segment.

For example, if your dendrite has only one segment, no matter how long it is, it will be spatially non-uniform. Therefore, you must increase the number of segments for more detailed spatial heterogeneity. However, it should be noted that the more segments you have, the longer the calculation will take.

**The spatial inhomogeneity specification feature of software.**

This feature is designed to allow you to specify the spatial inhomogeneity of a selected mechanism in your simulation. To use this feature, navigate to the central part of the window. Here, you will find the panel that offers five different modes for specifying the spatial inhomogeneity of your mechanism. Let's take a look at each of these modes:

**Simple Model**:

This mode offers several options to specify the spatial inhomogeneity of your mechanism. You can choose a constant value for the spatial inhomogeneity or specify a linear, quadratic, or polynomial function with two, three, or more parameters. You can also choose an exponential function for the spatial inhomogeneity.

**Custom Function**:

If you have a specific function in mind that is not covered by the options in the Simple Model mode, you can use the Custom Function mode. Here, you can enter any function you like, as long as it is written in either the neuron language or Python. To set your custom function, simply use the pop-up window.

**Custom Function from File**:

In addition to the Custom Function mode, you can also specify a custom function from a file. This option allows you to define your function in a separate file, written in either Neuron or Python. Once you have created your file, you can specify it in the software and use it to specify the spatial inhomogeneity of your mechanism.

**Table Function**:

If you have experimental data that you would like to use to specify the spatial inhomogeneity of your mechanism, you can use the Table Function mode. This mode allows you to download your data either manually or as a text file and use it to specify the spatial inhomogeneity.

**Special Function**:

Finally, the "Special Function" option displays sections and the distribution of segments within the neuron cell. You have control over the cell model's discretization stage. The discretization is depicted on the cell's spatial graph. The segments are marked differently for better visualisation.

**Visualization of Spatial Heterogeneity**

This part provides two different ways to visualize spatial heterogeneity. To access these options, navigate to the lower part of the main panel.

**Option 1**: Heterogeneity as the Distance from the Soma with this option, you can visualize the degree of spatial heterogeneity related to the Distance from the Soma. This visualization can help you identify areas of the neuron that exhibit high or low heterogeneity. To use this option, simply select it from the main panel.

**Option 2**: Spatial Colour map of Inhomogeneity on a cell. This option allows you to visualize spatial heterogeneity using a colour gradient representing the degree of inhomogeneity on the cell. The colours range from cool to warm, with cooler colours indicating lower levels of inhomogeneity and warmer colours indicating higher levels. To use this option, select it from the main panel.

Both of these visualization options can provide valuable insights into the spatial heterogeneity of your neuron. Choose the option that best suits your needs and explore the heterogeneity of your neuron in new and insightful ways.

### Adjust the stochastic distribution of mechanisms.

The main panel (**Figure**) to adjust stochastic properties of mechanism across any part of cell allows the user different option to define the mathematical formula for the stochastic distribution



To begin editing any mechanism, click on it. This will bring up a menu where you can select what you want to edit. The menu includes options for global variables, state variables, and parameters. You can access another window that provides more detailed information about the selected variable by clicking on any of these options.

In the new window, you'll see the current value of the variable if it's spatially homogeneous and “nan” is spatially non – uniform. If you want to make these mechanism stochastic, click the "**Define as a stochastic model**" bar. Doing this will open a new window called the **Stochasticity Editor**.

In the **Stochasticity Editor**, you can define the variable as a different stochastic function. This allows you to customize the behaviour of the variable as a stochastic function in time. Once you've described it, you can save your changes and edit other mechanisms.

The **Stochasticity Editor** window has four parts to help you model and visualize different types of stochastic distributions for a given variable. Let's go through each part in detail.

1. **Upper Part**:
   1. *Left part* : This part shows information about the state, “**Inhomogeneous**” of stochasticity of this variable/parameter, “**Mechanisms**”, its location on the cell, ”**Compartment**”, and full name with units, “**PARAMETER**”. It also indicates whether this variable/parameter is spatially inhomogeneous.
   2. *Right part “****Bounding****”*: Here you can select the parametric domain of (**DF**) density function of the stochastic variable, either non-infinite “**Keep as is**”, from above “**Bound from above**”, from below “**Bound from below”**, or from both sides, “**Bound from both sides**”. The full definition of any mode is on the right of panel.
2. **Stochastic Model**: This part allows you to select type of stochasticity either “**White noise**” or “**Coloured noise**” with PSD of different alpha correlation.
   1. *Simple Model*: This part includes standard models such as “**Uniform**”, “**Normal**”, “**Logarithmic**”, “**Exponential**”, “**Erlang**”, and “**Weibull**” distributionы. You can select any of these models. The selection of any distribution is accompanied by a pop-up panel containing the distribution's corresponding parameters.
   2. *Custom expression*: You can add your various stochastic functions by writing in line. **(For premium users)**
   3. *Custom expression*: You can add your various stochastic functions by writing and including them in "hoc"-files. **(For premium users)**
   4. *Table and linear interpolation*. You can add the experimental; data in table. **(For premium users)**
   5. *Special Functions*: The user is given a special set of simple functions to test stochastic variables. The function (**1,0,0,…**) is numerical delta function with uniform distribution of spectrum. The function (**1,1,1, …**) is function with zero frequency. The function (**1,0,-1,1,0,-1,** …) is a basic periodic function , cosine with half-Nyquist frequency. The function (**1,-1,1,-1,** …) is a basic periodic function , cosine with Nyquist frequency. The function **Foo** is a basic periodic function.
3. **Visualize and Model Part**: This part allows you to model and visualize the selected type of stochastic distribution. You can:
   1. Visualize the distribution density (**DF**) function for an infinite number of trials.
   2. Generate a sample of random numbers according to the chosen function. Number of simple indicated on the panel.
   3. Visualize the density function (**DF**) for this sample along with the ideal function.
   4. Build the autocorrelation function(**ACF**) of this sample.
   5. Build the power spectral density (**PSD**) function of this sample.
4. Apply or Disable Part: In the end, two buttons allow you to accept or refuse the stochasticity of this variable.

**Very important**. **The computation of the stochastic dynamics can be done using special run control panel “AltRunControl” provided be “BrainCell”**.



**Manager of synapses**.

Graphical user interface, application

Description automatically generated

**Figure. The synaptic panel and panel for the synaptic and stochastic simulations.**

The primary function of this panel is to generate and disseminate unique synaptic inputs onto a dendritic tree. The number of spines meticulously determines the number of synapses on a given dendritic tree, with neither excess nor deficiency. This information is crucial for individuals seeking to create synaptic inputs. Altering the distribution and density of synapses on the panel requires modification of the nanostructures on the primary panel. It is noteworthy that the spines on the synaptic panel are unchangeable.

At present, users can only adjust the location of the synapses, either on the spine itself or in its vicinity on the dendrite. The distribution of synapses may differ in proportions, with some distributed over the spines and others situated close to them. When a synapse is in proximity to a spine, the conductance of the spine becomes zero, effectively disappearing in an electrical sense. Conversely, if the synapse is located on a spine, it functions as a full-fledged conductor. When a spine is located next to the synapse, it is no longer a conductor.

This **Synaptic manager** is designed to provide you with a comprehensive understanding of the synapses on spines or dendrites. To get started, look at the upper portion of the panel. You'll find important information and tools to manipulate the location of synapses. The **Synaptic manager** window has three parts to help you model and visualize different types of synaptic distributions and efficacy. Let's go through each part in detail.

[**Synaptic manager** **upper part**](#SynManagerUpperPart)**.**

[**Synaptic manager** **middle part**](#SynManagerMiddlePart)**.**

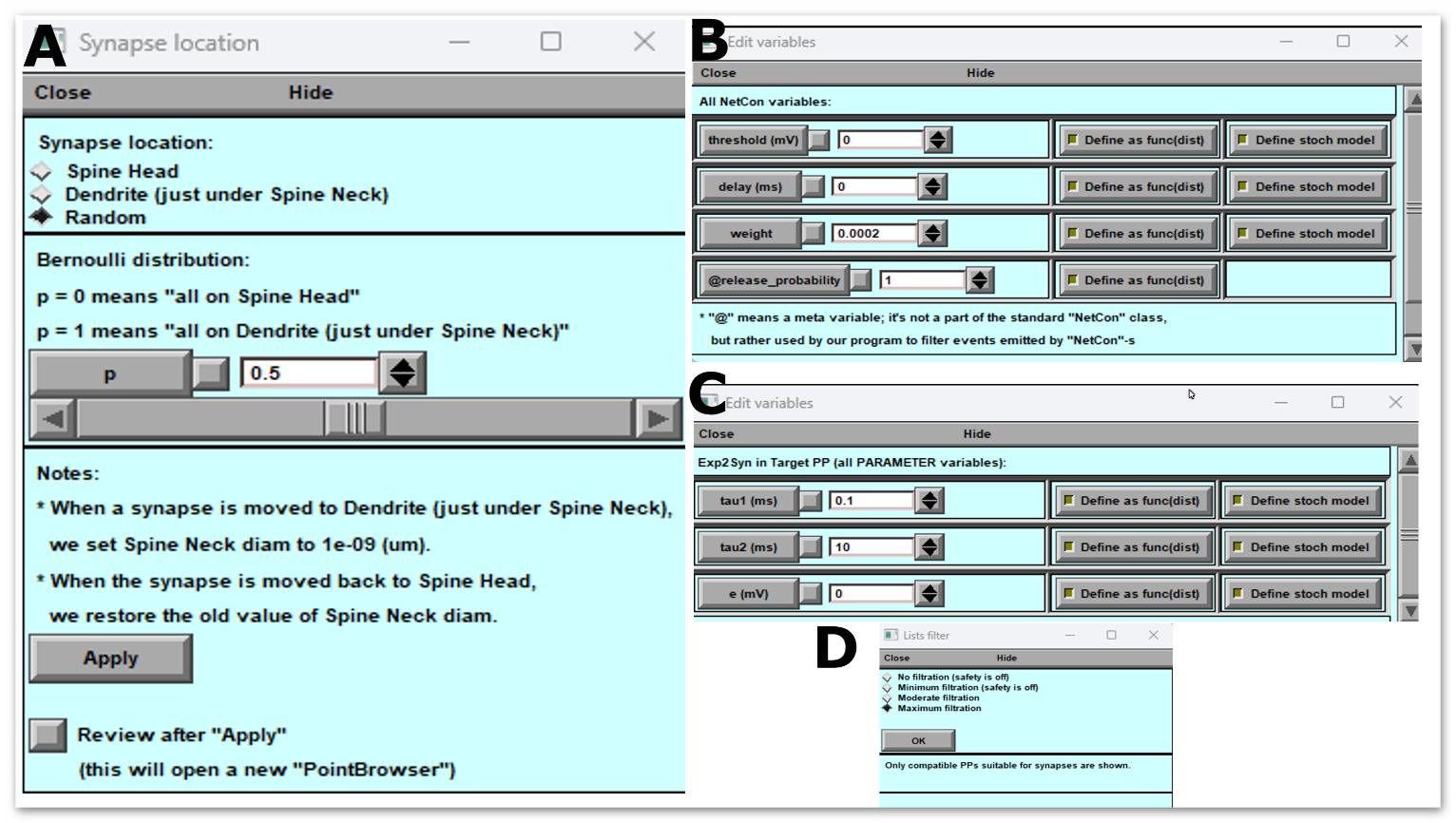
[**Synaptic manager** **bottom part**](#SynManagerBottomPart)**.**

#### Synaptic manager “upper part”.

The information section located in the upper portion of the panel is where you can find useful data about the synapses on dendritic tree. This section will indicate the number of synapses distributed throughout the dendritic tree, including the spatial range, and whether the parameters are spatially homogeneous or not. Additionally, it will provide you with stochastic variables of its parameters, which is crucial for understanding the behavior of the synapses.

To change the location of the synapses on spines or dendrites, use the tools located in the upper right portion of the panel. By manipulating these tools, you can adjust the position of the synapses to your desired location.

Clicking on the “**Synapse location**” button opens a pop-up panel (**Fig.\*\*\*A**).



**Figure: Pop-Up Panels of Synaptic Manager. A**) shows the manager of synaptic location, which allows users to visualize the spatial distribution of synapses. **B**) displays the manager of presynaptic signals, which provides information about the presynaptic neuron's activity. **C**) presents the manager of synaptic parameters, where users can modify synaptic properties such as strength and plasticity. **D**) shows the manager of presynaptic and postsynaptic mechanisms, located in the mechanism directory, which enables users to customize the synaptic transmission and reception mechanisms.

This part guide you through this panel's upper and lower (**Fig.\*\*\*A**) parts and help you understand their functionalities.

**Upper Part:** The upper part of the “**Synapse location**” panel enables you to modify the location of the synapses. There are three options available on this panel that you can use to change the location of synapses:

1. Place all synapses on spines
2. Place synapses on dendrites at the location of spines
3. Place synapses on dendrites and spines in different proportions

You can select any of the above options according to your requirements, and the software will make the necessary changes accordingly.

**Lower Part**: The lower part of the Synapse panel contains informational data and two buttons: "**Apply**" and "**Review after Apply**"

The informational data in the lower part provides details about the synapses' current location and density. It also shows the changes made to the synapse location after selecting any options mentioned in the upper part of the panel.

The " **Apply** " button confirms the changes made to the synapse location and saves them in the software. If you do not wish to save the changes, click the "Cancel" button.

The " **Review after Apply** " button lets you see how the program changed the synapse location. This option lets you verify the changes and ensure they align with your requirements.

**Synaptic structure**.

This part is for creating synaptic structures in the neuronal model. Synapses are two types in software "**Neuron**", one that requires a presynaptic signal and the other that works without a synaptic signal. The second type turns on depending on the "**onset**" parameter.

The panel offers two synaptic structures to accommodate these two types. The first structure includes presynaptic and postsynaptic mechanisms (**Source PP 🡪 NetCon 🡪 Target PP 🡪 Section**), while the second includes only postsynaptic mechanisms (**Source PP 🡪 Section**). Selecting one of these scenarios allows you to get either postsynaptic or presynaptic and postsynaptic mechanisms in the middle panel.

Please note that if you choose only postsynaptic mechanisms, you cannot use the "probability of release" parameter.

#### Synaptic manager “middle part”

This middle section of the panel has been specifically designed to allow you to organize and manipulate the presynaptic and postsynaptic functions of synapses, with a variety of parameters to choose from. Here, you can choose different presynaptic mechanisms (**left panel**) on the board and connect them to the synaptic mechanism (**right panel**). Once you have selected the appropriate presynaptic and postsynaptic mechanisms, you can start modifying the set of parameters just by clicking on the buttons labelled “**Edit source PP vars**”, “**Edit target PP vars**”.

The modification of the spatial distribution of Point base (**Synaptic**) mechanisms follows the same rules as the modification of the spatial distribution of standard density mechanisms, as explained in the section. [Manage distribution mechanisms.](#_Manage_distribution_mechanisms.)

The randomness of synaptic parameters changes specifically when the presynaptic signal is generated, as opposed to the randomness of spatially distributed mechanisms. This means that stochastic events in the synapse only occur either when the presynaptic signal is generated or when the "onset" time parameter value is reached. However, the panel for controlling stochasticity is no different from the panel for spatial mechanisms [Adjust the stochastic distribution of mechanisms.](#_Adjust_the_stochastic).

#### Synaptic manager “bottom part”.

In order to manipulate the "[NetCon](https://www.neuron.yale.edu/neuron/static/py_doc/modelspec/programmatic/network/netcon.html)" mechanism, which connects the presynaptic and synapse functions via four parameters - "**threshold**", "**delay**", "**weight**", and "**release probability**" - we have designed a specific section at the bottom of the panel, featuring a button labeled "**Edit NetCon vars**" (Fig.B). This section is crucial to ensure seamless coordination between the presynaptic and postsynaptic functions.

In addition, you can use the "**Adjust list filter**" button (Fig.D) to select and add various synaptic mechanisms that are not yet included in the Middle panel. This feature allows you to add your own biophysical mechanism and use it to form new synaptic connections, unlike the standard synaptic mechanisms of “**Neuron**”. Please note that any new mechanisms should be uploaded to the "Mechanisms" section/directory.

To accept all modifications with synaptic connections, simply click the "**Apply**" button located in the same bottom section.

**Simulation modes: Users can choose from a variety of simulation modes to simulate**

**SYSTEM** **PREPARATIONS** **FOR** **HIGH-END** **CALCIUM** **SIMULATIONS**

**Preparing** **Worker** **computer** **/** **cluster** **(HPC,** **OS** **Linux)** **for** **Ca2+** **simulations**

1. The remote cluster for computation of ASTRO is prepared only once. Thereafter, for future computations only the host computer will be used. However, if the cluster code of ASTRO will require substantial modification the user needs to change of C++ and HOC code in HPC directory.

2. Before installation the kernel of ASTRO on the cluster, one need to be sure that “mpic++” (***Message*** ***Passing*** ***Interface***) compiler is present in the system path. The free version of mpic++ is here [https://www.open-mpi.org/software/ompi/v2.0/.](https://www.open-mpi.org/software/ompi/v2.0/) Almost all modern clusters operating under Linux have this compiler in the system.

3. Download the folder HPC with all its content from https://github.com/LeonidSavtchenko/ASTRO. This folder must be saved to the place shared between cluster nodes. For example, it can be saved in the directory “/home/<***username***>”. To share files between a host computer and a remote cluster user can use free software “WinSCP” [https://winscp.net/eng/download.php.](https://winscp.net/eng/download.php)

4. Install NEURON on the cluster. The latest version can be downloaded from the official site. We recommend using the installation from source code taking the sources from here <https://www.neuron.yale.edu/neuron/download/getstd>and following steps 1-5 of the next instruction [https://www.neuron.yale.edu/neuron/download/compile\_linux.](https://www.neuron.yale.edu/neuron/download/compile_linux)

**--with-paranrn** **option** **should** **be** **added** to the configure command for NEURON installation to enable distributed computations.

**To** **create** **the** **parallel** **version** **of** **astrocyte** **the** **code** **was** **taken** **from** **here** **:**

[https://senselab.med.yale.edu/ModelDB/ShowModel.cshtml?model=97985&file=/multisplit/#](https://senselab.med.yale.edu/ModelDB/ShowModel.cshtml?model=97985&file=/multisplit/#tabs-2) [tabs-2](https://senselab.med.yale.edu/ModelDB/ShowModel.cshtml?model=97985&file=/multisplit/#tabs-2)

NEURON GUI is not required by cluster simulation. If you want to remove it from installation, do not download *iv-mm.tar.gz* archive and replace *--with-iv=$HOME/neuron/iv* with *--without-iv* when calling *configure* for NEURON installation.

The user can use PuTTY and WinSCP programs to work with console and file system of the remote cluster. Both programs are free.

The structure of the directors, in GitHub, necessary for working with the cluster has the form:

37

The cluster setup is done.

**NOTE**: All files from directory HPC should be downloaded on the cluster, keeping the structure of directories unchanged. All files from directory HOST should be downloaded on the local computer (OS Windows), keeping the structure of directories unchanged.

**Preparing** **Host** **computer** **(client,** **OS** **Windows)** **for** **Ca2+** **simulation**

1. Open *clusterCaSim/host/scripts/win-lin/params.bat* and set your cluster connection parameters including the path to the *hpc* folder located in the cluster. Below is the corresponding fragment of the ***params.bat*** file.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* set HEADNODEIP=144.82.46.83

set LOGIN=my\_login

set PASSWORD=my\_password

set HEADNODEWORKERDIR=/home/\*\*\*\*\*\*\*/hpc \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2. Open *clusterCaSim/host/core/BasicParams.m* and add “*avaliableNodes*” variable with names of the user cluster nodes.

Here it is the corresponding fragment of code

% Whether to conduct simulation on remote HPC cluster remoteHPC = true;

% Whether to ZIP I/O data files before transferring through network zipDataFiles = false;

defaultGeometry = 'default - AstroGeometry';

% Names of all cluster nodes you may want to use in the simulation

% (must be kept in sync with those ones defined in "hpc\hostfile\_BusyMaster and hpc\hostfile\_IdleMaster")

availableNodes = {'tuxmaster', ...

Also, fill *clusterCaSim/hpc/hostfile\_BusyMaster* *and* *hostfile\_IdleMaster* files with node names in the following manner: each line should contain the name of the node followed by ' max\_slots=1' without quotes.

Contents of both files should be the same except that *hostfile\_IdleMaster* should not include the master node (only slaves).

3. After launching *start.m* and pressing the key “Calcium dynamics on cluster” the user will have two options:

38



a. Option 1. The user can upload from the cluster and analyse the previously obtained results.

b. Option 2. The user can start a new simulation.

As soon as user press option 2 the following window appears

With this panel, the user can specify

39

1. The parameters of simulations (dt – time step of computation, step-per-ms – time step of data visualization and InitSeed – seed of random number generator)



2. The parameters of Nano geometry of astrocyte. 3. The parameters of calcium dynamics.

The definition of all these parameters are the same as for the astrocyte model calculated on the local computer. The user can find a description of these parameters in this manual (See the figs 18, 23 and 24).

Also the user can upload the astrocyte geometry on the cluster using the structure of *hoc*-files, the same structure of files as for the geometry of ASTRO describe previously (see part: ***Generating*** ***/*** ***downloading*** ***astrocyte*** ***stem*** ***tree)***.

The default basic of astrocyte geometry is defined by the file AstroGeometry.hoc.

With following panel the user can modify parameters on the *HPC* cluster:

40

On this panel there are three important options:

**scalTest** – check this parameter if the scalability test is useful here. This test will show how well execution time scales relative to the number of processes (see the parameter **np**).

**remoteHpc** – uncheck if ASTRO will be run on the client PC only without cluster. The user should also modify NRNDIR and HPCDIR parameters that point to the NEURON and *hpc* folder locations on client PC in the *clusterCaSim/host/scripts/win-win/params.bat* and *params.sh*.

**procScheme** – processor distribution scheme. This parameter allows to include to the calculation either only the master computer, or only the slaves, or all together.

**np** – number of processors for any slave computer.

If the user had compiled the executables to run the simulation, then the user should recompile them after changing any parameters in Matlab files using *clusterCaSim/host/BUILD\_AllHostExecutables.m.*

**Technical** **Notes**

**Nano** **geometry**

In data sets comprising 3D-reconstructed nanoscopic processes all serial sections should be represented by (10-20) XYZ points scattered along the section circumvent, for diffusion simulations to work properly.

Because of a significant amount of computations in complex diffusion simulation Matlab cannot handle stopping or restarting it by pressing a button.

**Calcium** **dynamics** **(Cluster)**

There are situations when selected geometry cannot be split into the specified number of processors. In this case, the user will see an MPI error before the computations begin. To solve the problem, the user can simply increase or decrease the number of processors.

Dendrites (astroglial processes will be called dendrites in the original NEURON environment) should be connected to the soma only in the 1 position. Otherwise cell splitting fails. Examples:

Good: soma[0] connect dendrite[125](0), 1 Wrong: soma[0] connect dendrite[125](0), 0.5 Wrong: soma[0] connect dendrite[125](0), 0.1 Wrong: soma[0] connect dendrite[125](0), 0

41

This version of ASTRO assumes that the user will work on own computer cluster. The new ASTRO will include a pre-installed version on the Amazon clouds for general usage.

Follow the updates on the GitHub/LeonidSavtchenko/Astro

42