USER GUIDE

*BRAINCELL* *1.0.*

Brain cell *in* *silico*

© University College London, MIT licence

2023

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

Table of Contents

[INTRODUCTION 4](#_Toc130903799)

[Key system and software requirements 4](#_Toc130903800)

[Basic version 4](#_Toc130903801)

[Full version 4](#_Toc130903802)

[The strategy of building the model: summary 5](#_Toc130903803)

[The outlines of experimental data or approximations required to create a realistic brain cell model. 6](#_Toc130903804)

[GETTING STARTED 6](#_Toc130903805)

[Installing and running BRAINCELL 6](#_Toc130903806)

[Setting up and launching 6](#_Toc130903807)

[File structure in Host computer (under Windows) 7](#_Toc130903808)

[Preparing BRAINCELL system files 7](#_Toc130903809)

[Astro/ Base. Setting up and running BRAINCELL: Astrocyte configuration. 8](#_Toc130903810)

[GENERATING COMPLETE ASTROCYTE MORPHOLOGY 8](#_Toc130903811)

[Generating / downloading astrocyte stem tree 9](#_Toc130903812)

[Generating astroglial morphology on the nanoscale 11](#_Toc130903813)

[Geometry of nanoscopic processes 11](#_Toc130903814)

[Populating astrocyte tree with nanoscopic processes 11](#_Toc130903815)

[Tissue-filling properties of astroglial morphology 12](#_Toc130903816)

[Nano Astro. Download previously created astrocyte morphology. 14](#_Toc130903817)

[Neuron / Base. Setting up and running BRAINCELL: Astrocyte module. 15](#_Toc130903818)

[GENERATING COMPLETE NEURON MORPHOLOGY 15](#_Toc130903819)

[Nano Neuron. Download previously created Neuron morphology. 16](#_Toc130903820)

[Nano geometry modification. 17](#_Toc130903821)

[The arrangement of spines. 17](#_Toc130903822)

[The spine head and neck geometry. 17](#_Toc130903823)

[Manage distribution mechanisms. 18](#_Toc130903824)

[Adjust the spatial distribution of mechanisms. 19](#_Toc130903825)

[Synapse distribution. 19](#_Toc130903826)

# 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.**

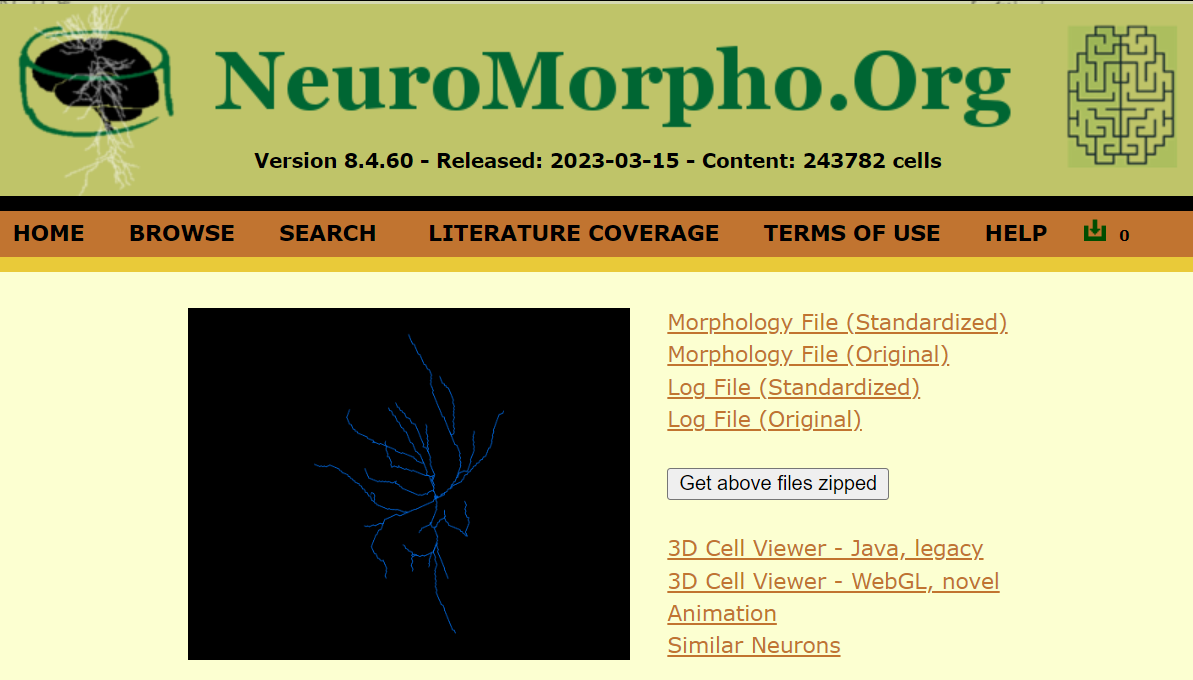
4. The file ***init.hoc*** opens three windows: System window (***cmd.exe***) (a), a window panel to define the gross astrocyte geometry (stem tree,b), and a menu panel to set the density for higher orders of nanoscopic processes ('**Leaf** **number'** c) and number of nanostructures per dendtire (**‘Max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

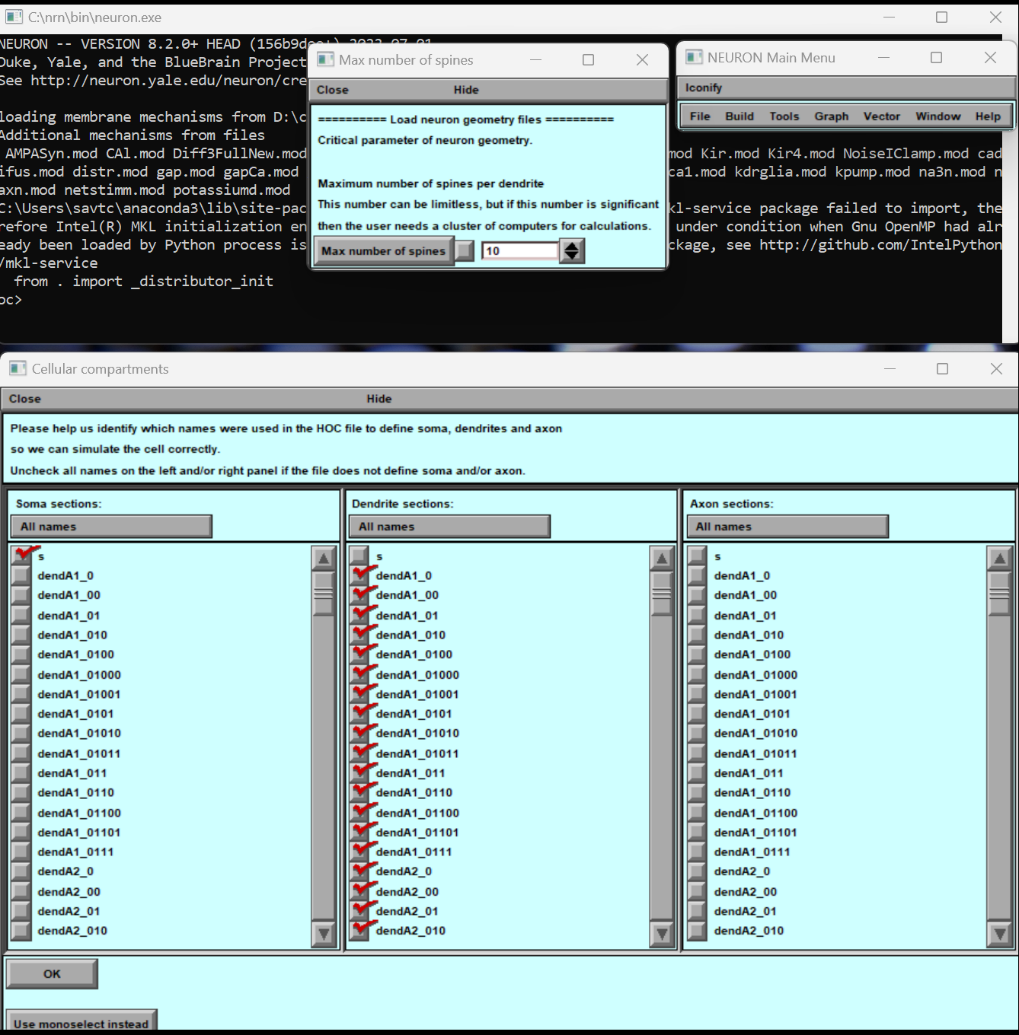
To proceed with simulation and management of biophysical mechanisms, select the astrocyte with Nano structure. This will take you to a new option for simulation and management..

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.

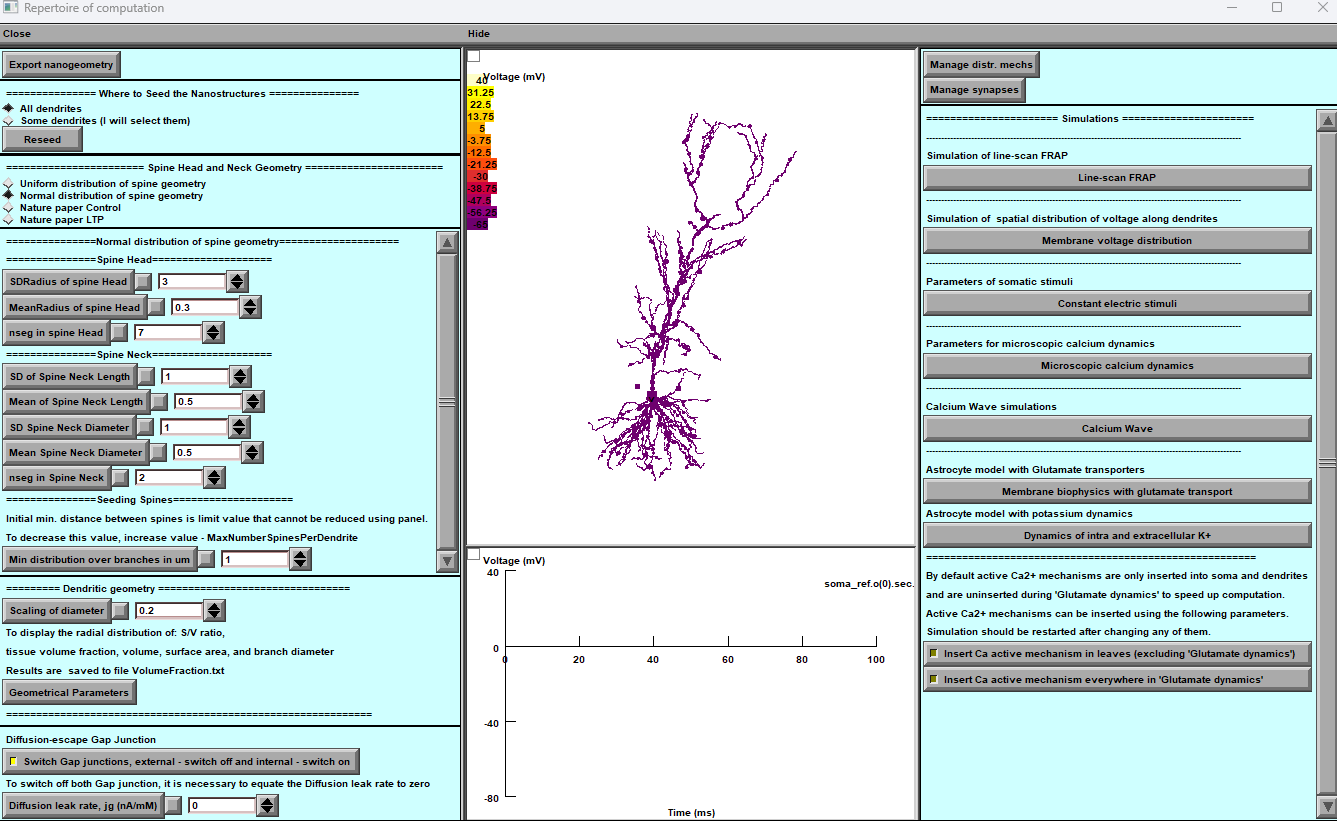
## GENERATING COMPLETE NEURON MORPHOLOGY

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.



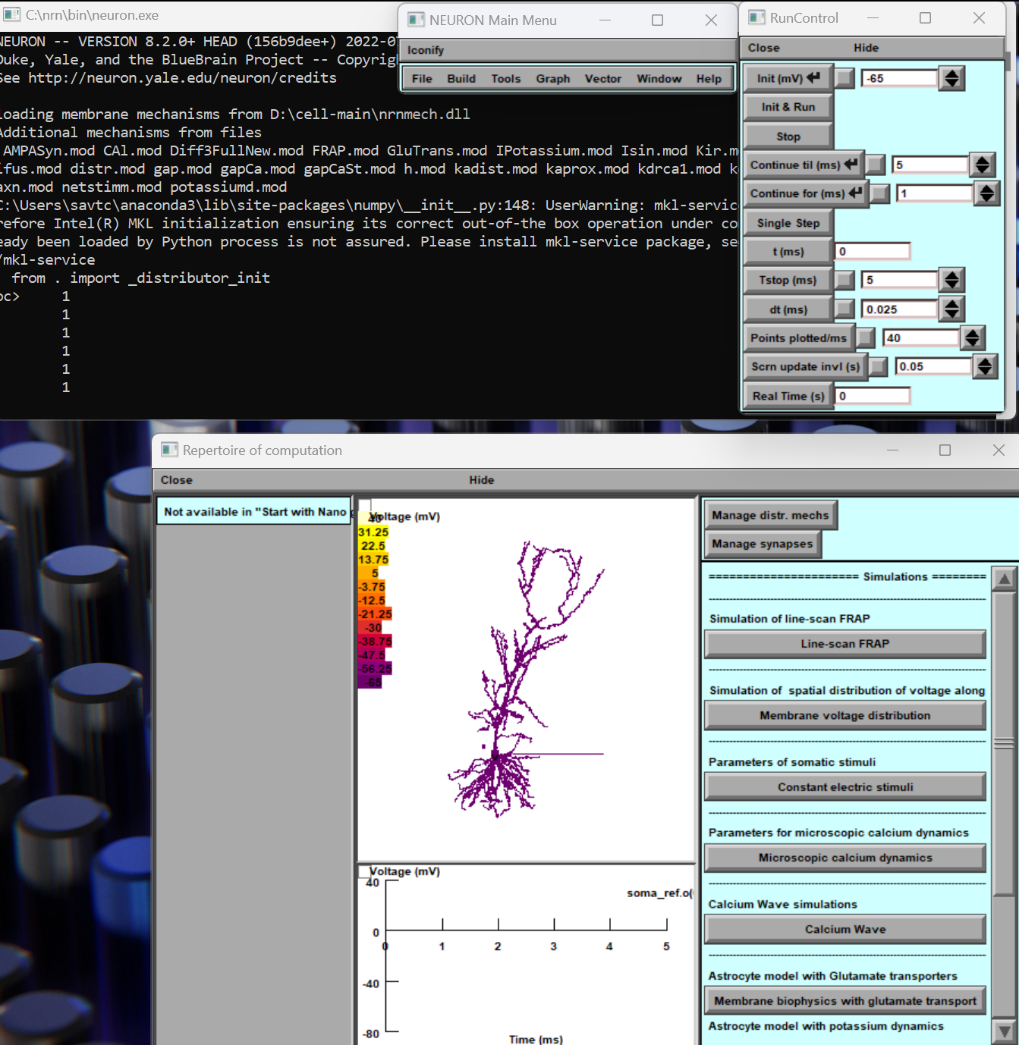
**Fig. Main windows for neuron simulation**

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 our 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 mode offers a visualization sections and segment distribution across all neuronal cell. You can control how ofter your neuronal cell model is discretised.

**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.

**Simulation modes: Users can choose from a variety of simulation modes to simulate the neuron's behavior accurately.**

**Linescan** **FRAP** **experiment:** **probing** **intracellular** **connectivity** **of** **astroglia**

Fluorescence recovery after photobleaching (FRAP) applied with respect to water-soluble intracellular indicators assesses effective diffusivity across the cellular compartments adjacent to the bleached area. FRAP experiments and the corresponding simulations (launched from the main ASTRO window; Fig. 18c) could be used therefore to test whether the modelled astrocyte morphology reproduces intracellular diffusivity (connectivity) properties documented empirically. The default values correspond to the data obtained in CA1 astroglia using linear (**linescan**) photobleaching tests with Alexa Fluor 594 or Alexa Fluor 488.

This test simulates FRAP with a photobleaching area represented by a segment (**linescan**) in the XY plane. Pressing **'Linescan** **FRAP'** opens its control window (Fig. 21):

The initiation panel (Fig. 21a) provides the following settings:

**'Linescan** **width'** sets the linescan width: in two-photon excitation mode, the point-spread function of the focused laser beam is on average 1-1.5 µm depending on the system optics; this parameter is therefore adjustable (within a narrow range).

**'Angle'** and **'Y** **at** **x=0'** set, respectively, the inclination angle and the y-axis intersect of the photobleaching (scanning) segment.

**'Initial** **concentration** **(mM)'** sets the basal concentration of the photo-bleachable dye molecules, to match the experimental value (in configuration of whole-cell dialysis).

25

**'Photobleaching** **rate'** sets the rate of photobleaching (free parameter that can be used to match simulated and experimental data).

**'Dfree'** sets the intracellular diffusivity of the dye molecules based on empirical estimates.

**'Bleaching** **recovery** **interactions'** sets the number of bleaching-recovery periods per trial.

a b



c

**Figure** **21.** **Window** **panels** **to** **control** **and** **monitor** **simulated** **linescan** **FRAP.** **a**, Parameter settings panel. **b,** Concentration dynamics of non-bleached molecules, colour-coded and mapped on cell morphology. **c**, Concentration space profile sample.

The default duration of FRAP trials is 8000 ms, with photobleaching occurring every 1000 ms. In addition to the colour-coded shape map display (Fig. 21b), the results of FRAP simulations could be plotted and saved using standard NEURON functions: for instance by selecting and identifying the cell branch / area of interest ('dendrite' in NEURON format) using **PointProcessGroupManager,** and next configuring the output display and store using the **Graph** menu (Fig. 21c).

**Probing** **membrane** **mechanisms** **of** **astroglia**

This part of ASTRO deals with simulations of membrane mechanisms (passive electrical properties, voltage-dependent channels, ion exchange, receptor current, etc.) and intracellular processes (Ca2+

26

entry, buffering, diffusion, and removal mechanisms) acting in morphologically realistic astrocytes. Most of the programming routines involved are an inherent part of the NEURON environment, with the full guidance available in the NEURON documentation [https://www.neuron.yale.edu/neuron/docs.](https://www.neuron.yale.edu/neuron/docs)

Therefore the sections below provide only brief information relevant to astroglial probing.

**Membrane** **voltage** **landscape**

In the main control panel (Fig. 18), pressing **'Compute** **the** **spatial** **voltage** **distribution'** opens window panels (Fig. 22) which report membrane voltage across selected cell processes upon a step current injection at the soma (a common electrophysiological test scenario). The default resting potential is -85 mV.

a b



c

**Figure** **22.** **Window** **panel** **to** **monitor** **membrane** **voltage** **distribution** **including** **a** **gap** **junctions** **feature.**

**a,** Settings panel. **b,** Dynamic voltage landscape mapped on the cell morphology. **c,** The corresponding voltage profile plot along individual processes ('dendrites' in NEURON).

The initiation panel (Fig. 22a) provides the following settings:

**'Stimulus** **amplitude** **(nA)'**, amplitude of constant depolarising current injected into the soma. **'Duration** **(ms)**', duration of constant depolarising current.

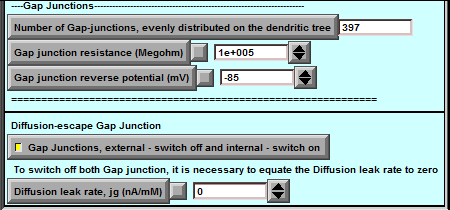
**'Computation** **time** **(ms)**', simulation run time (time of display). **'Run',** to start simulation.

27

**Simulating** **membrane** **voltage** **in** **response** **to** **local** **current** **hotspots**

This ASTRO function can be performed by modifying settings in the ASTRO menu section below titled **'Simulating** **astroglial** **glutamate** **transporters'**. The latter provides full monitoring of the dynamic membrane voltage landscape mapped onto the cell morphology.

**Gap** **junctions**

ASTRO has basic modelling provisions for the two types of gap junctions, which simulate the bulk diffuse (intracellular and extra cellular connections) and electrical connections between adjacent astroglia (control panel in Fig. 18a, bottom).

**Gap** **junction** **menu** **(fragment** **of** **Fig.** **18a,** **bottom).**

*The* *first* *type* is a local electrical connection (current sink), which will affect local membrane potential *Vm*. It sets the local transmembrane current as (*Vg-Vm*)/*r*g where *Vg* and *rg* are the gap junction reverse potential and resistance, respectively.

**'Number** **of** **Gap** **junctions,** **evenly** **distributed** **on** **the** **dendritic** **tree’:** indicates the total number of gap junctions scattered uniformly across astrocyte processes ('dendrites' in NEURON terminology). **'Gap** **Junction** **Resistance** ***rg*** **(MOhm)':** a default value of 1010 MOhm, equivalent to having zero gap junction current.

**'Gap** **Junction** **Reverse** **potential** ***Vg*** **(mV)'**: the default value is set at -85 mV.

*The* *second* *type* (**'Diffusion** **escape** **gap** **junction'**) is a local diffusion leak. It is determined by the rate of *jg*(*C*g-*C*0) where *jg*, *C*g, and *C*0 are the leak constant, local intracellular ion/molecule concentration, and intracellular ion/molecule concentration on the other side of gap junction (resting concentration by default), respectively. The default value of *jg* is zero, reflecting no gap junction diffusion leak.

**'Diffusion** **leak** **rate,** **jg** **(nA/mM)':** sets *jg* value.

**‘Gap** **Junctions,** **external** **–** **switch** **off** **and** **internal** **–** **switch** **on’:** switches between inter-cellular (external) and internal gap junctions. The flux of internal (autaptic) gap junction is *jg*(*C*g-*C*g1), where the Cg1 is the ion concentration at the connected part of astrocyte ; the connecting compartment is currently set to be chosen randomly from the nanoscopic processes in the area; the internal gap junction procedure is to be further adjusted in accord with additional further experimental constraints.

**Modelling** **intracellular** **calcium** **dynamics**

ASTRO provides simulation and analyses of the intracellular Ca2+ dynamics - including entry, buffering, diffusion, regenerative waves, and removal - across the entire astrocyte morphology. This type of simulation may require significant computational resources. Shorter-duration (1-1000 ms) trials could be feasible on the Host computer alone. Simulating longer events over large cell areas will probably require a dedicated Worker computer / cluster (see below for detail).

The current ASTRO version enables simulation of the two following scenarios: (i) macroscopic Ca2+ dynamics over the entire cell morphology, and (ii) oscillatory, single-channel type Ca2+ entry dynamics on the nanoscopic or microscopic scale within one or two selected processes.

28

**Ca2+** **wave** **simulations**

Pressing **'Calcium** **wave'** key on the main control panel (Fig. 18c) opens new windows (Fig. 23) for setting, controlling, and displaying intracellular Ca2+ dynamics in response to a local increase in the concentration of the Ca2+ channel ligand IP3.

ASTRO employs the standard NEURON-integrated mathematical formulism of Ca2+ reaction-diffusion kinetics including buffering and removal, which has been tested and validated in numerous studies; its detailed description can be found in the ***cadifus.mod*** file. The basic set of equations of calcium dynamics is as follows (built-in NEURON functions and forma are used)**.**

*d* *dta*] *Dca* *d*2[*Ca*] - (2*FARADAYam* , where [Ca] is Ca2+ concentration, *Dca* - diffusion coefficient, diam is a local diameter, ica is a sum of voltage dependent potassium currents and the pump current *ipump* is

2

*dt*

[*C*

*ca* *pmp*

*i* - *i* )*di*

*ipmp* 2 *FARADAY* ( *fflux* - *bflux* ), *where* *ffluxand* *bflux* are the forward and reverse fluxes

*area*

Two types of buffers are used, mobile "*bufm*" and endogenous "*bufs*" with corresponding kinetic constants kf and KD

~ [*Ca*] *bufs* - *cabufs* (*kfs*, *KDs**kfs*)

~ [*Ca*] *bufm* - *cabufm* (*kfm*, *KDm**kfm*)

SERCA pump, channel, and leak with alpha, relative abundance of SERCA mechanism

*pump* *current* : *d*[*Ca*] (-*v*max [*Ca*[*Ca*]2 *Kp*2 )

2

] 

*dt*

*channel* :~ *hc* - *ho* (*kon**Kinh*, *kon**ca*), jchnl is Ca flux releases from SERCA to cytoplasm *d*[*Ca*] ( *alpha**j*max (1- [*Ca*])( *ip*3*ip*3*i* *ip*3[*Ca*]*ho* )3 )

*i* *K* *ca* *Kact*

*dt* *caer*

*leak* : *d*[*Ca*] (*alpha**L*(1- [*Ca*])), L = vmax[*Ca*]2 / [*Ca*[*Ca*]2 *Kp*2 /(1- [*Ca*])

2

] 

*dt* *caer*

*caer*

29

a b c



d

**Figure** **23.** **Window** **panels** **to** **control** **and** **monitor** **Ca2+** **waves** **dynamics.** **a***,* Ca2+ wave trigger settings panel. **b**, Colour-coded, dynamic [Ca2+] landscape mapped onto cell morphology (top) and readout plot of [Ca2+] dynamics the soma (by default; bottom). **c**, Parameter setting panel for Ca2+ reaction-diffusion processes. **d,** Window panel to monitor Ca2+ homeostasis and dynamics at individual processes ('dendrites' in NEURON nomenclature).

**'Ca** **wave'** panel (Fig. 23a) has controls as follows:

**'IP3** **increase** **onset** **(ms)'**, sets the onset of a step increase in the IP3 concentration; **'IP3** **peak** **concentration** **(mM)'**, sets amplitude (0.005 mM shown).

**'Run** **time'** sets simulation run time (default 2000 s).

The window parameter setting panel for Ca2+ reaction-diffusion (Fig. 23c) displays the corresponding explanations above the setting keys. Concentration and kinetic parameters of the Ca2+ indicator

(Fluo-4) are constrained by experimental measurements whereas the default values for Ca2+ pumps and channels correspond to estimates found in the literature. These parameters should be further constrained by the user based on specific experimental tests.

A new simulation run can start once the current run has ended. **'Graph'** panel (Fig. 23b, bottom) provides [Ca2+] time course in the soma. The current ASTRO version deals with IP3 rises at the soma but it can be adapted to 'release' IP3 locally at any place in the astrocyte.

30

**Choosing** **between** **IP3-dependent** **mechanisms** **of** **Ca2+** **signalling**

In addition to the NEURON-default IP3-dependent Ca2+ mechanism, one can use two alternatives developed earlier specifically for astroglia: one incorporating glutamate-dependence of IP3 concentration (De Pitta et al, 2009, *J* *Biol* *Phys* 35:383-411), and one involving a specific IP3-receptor dynamics (Fink et al, 2000, *Biophys* *J* 79:163-83). To use the De Pitta mechanism, (i) set IP3 concentration to zero (Fig. 23a panel), and (ii) invoke ***cadifus*** mechanism in the standard '**MechType'** NEURON menu: glutamate concentration set by the '**ModelStim\_cadifus'** button. To run the Fink model, (i) copy all mod-files from directory *...Astro-master\neuronSims\* *ExtraMechanism* *fink2000\* to the main directory *...Astro-master\neuronSims\,* and (ii) compile these files in NEURON using the standard ***mknrndll.exe*** command.

**Simulating** **microscopic** **Ca2+** **events**

Pressing **'Calcium** **dynamics'** key (Fig. 18c) prompts four new windows (Fig. 24):

a b c



**Figure** **24.** **Window** **panels** **to** **control** **and** **monitor** **microscopic** **[Ca2+]** **dynamics.** **a**, Parameter settings for single-channel Ca2+ dynamics ('sparks' and 'puffs') in two selected 'active' branches (bottom: 'dendrite' numbers 30 and 31 are shown, in NEURON nomenclature). **b**, Visualisation of internal [Ca2+] dynamics mapped on cell morphology (top), and [Ca2+ ] time course in two selected dendrites (bottom; dendrites d1 and d2 are shown); yellow circle depicts the area of interest ('dendrites' 30 and 31). **c**, Parameter setting panel for Ca2+ reaction-diffusion processes (as in Fig. 23c).

Window panel **'Calcium** **stimulation** **parameters'** (Fig. 24a) provides parameter settings for microscopic Ca2+ entry kinetics. The latter is assumed to consist of a series of stochastic Ca2+ channel openings (individual events) including higher-frequency bursts. Note that many of the parameters

31

involved are currently unknown and therefore represent the subject of investigative exploration and probing.

**'Single** **channel** **calcium** **entry** **flux**' sets the rate of calcium entry during an individual event. **'Mean** **interval** **between** **two** **calcium** **events'** is the average time interval between events. **'Basal** **Ca2+'** sets is basal [Ca2+ ] which is reverse concentration for the linear Ca2+ efflux. **'Rise'** and **'Decay'** set the rise and decay time of Ca2+ flux during a single channels opening. **'Events** **per** **burst'** sets the number of calcium channels opening per burst.

**'Randomness'** is set to 0 for uniformly random channel opening (Poisson process) with, or to 1 for cyclic channel opening.

**'Burst** **onset'** sets the time of the event burst.

**'IP3** **concentration'** sets the initial concentration of IP3 ions.

**'Active** **dendrite** **1'** **and** **'Active** **dendrite** **2'** are astroglial processes ('dendrites' in NEURON terminology) where active Ca2+ entry is enabled.

**'Stimulation** **time'** sets the run time of simulations. In astroglia, Ca2+ dynamics is a relatively slow process and the simulation trial normally requires needs at least 100 seconds.

**Simulating** **astroglial** **glutamate** **transporters**

Astroglial plasma membranes are enriched in high-affinity glutamate transporters which generate rapid inward current upon glutamate binding. In ASTRO main panel, pressing **'Glutamate** **transporters'** key (Fig. 18c, bottom) opens the menu (Fig. 25), which enables simulations of the dynamic membrane voltage landscape mapped onto cell morphology, in response to volume-limited application ('uncaging') of glutamate at a selected area of the cell.

**'Uncaging** **glutamate'** panel (Fig. 25a, top) provides self-explanatory parameter settings for volume-limited glutamate application (uncaging) within a round area of the cell. The *glutamate* *transporter* *kinetics* (Bergels and Jahr, 1997 Neuron 19: 1297-1308) includes six independent states (*Ci*, where *i* =1,…,6). The detailed description can be found the *GluTrans.mod* file. The basic relationships are

*dC*1 *C*1[*Glu*]*o* *k*12 *u*(*v*,-0.1) *C*2*k*21 Parameters

*dC*2 *C*2[*Na*]*o* *k*23*u*(*v*,0.5) *C*3*k*32 *k*12 0.015 (*mM* *ms*)11 *dC*3 *C*3*k*34 *u*(*v*,0.4) *C*4*k*43 *k*34 0.2 *ms*-1

*dt*

*dt*

*dt*

-

*k* 20 (*mM* *ms*)

-

23

*k*45 4 *ms*-1 *dt*4 *C*4*k*45 *C*5*k*54[*Glu*]*in* *k*56 1 *ms*-1

*dC*

16

*k* 

*dt*5 *C*5*k*56 *u*(*v*,0.6) *C*6 *k*65[*Na*]*in* [*Na*]*in*0.0016 (*mM* *ms*)-1 *dC*6 *C*6[*K*]*in* *k*61 *C*1*k*16*u*(*v*,0.6)[*K*]*o*[*K*]*in* 120 *mM*

*dt*

*dC*

15 *mM*

*where* *u*(*th*,*x*) exp(*th* *x* / (2 26.7)) [*Glu*- ]*in* 0.3 *mM*

*k*21 0.1 *ms*-1 *k*32 0.5 *ms*-1 *k*43 0.6 *ms*-1

*k*54 10 (*mM* *ms*)-1 *k*65 0.1 (*mM* *ms*)-1

*k*61 2104 (*mM* *ms*)-1 [*Na*]*o* 150 *mM*

[*K*]*o* 3 *mM*

[*Glu*- ]*o* 20106 *mM*

NOTE: The onset of uncaging must be >3 ms, to ensure stable membrane kinetics. The user can add another, simultaneously occurring uncaging spot specifying the distance between the two spots at the distance X from the first one.

Transporter currents are computed in somatic voltage clamp configuration: the clamp parameters are indicated on the panel **'Somatic** **voltage** **clamp'** (Fig. 25b) including electrode resistance, clamp

32

voltage and duration. The monitoring panel (Fig. 25c) display colour-coded membrane current landscape mapped onto cell morphology; the graph panel (Fig. 25d) displays the time course of extracellular glutamate and clamp current at selected cell processes.

a b



c

d

**Figure** **25.** **Window** **panels** **with** **parameter** **settings** **and** **readout** **plots,** **to** **simulate** **the** **dynamic** **membrane** **voltage** **landscape** **mapped** **onto** **cell** **morphology,** **in** **response** **to** **volume-limited** **application** **('uncaging')** **of** **glutamate.** **a,** Glutamate uncaging parameter settings. **b,** Voltage-clamp parameter settings including electrode positioning. **c,** Visualisation panel. **d,** Time course graph panel.

Window panel **'Glu** **setting'** (Fig. 24a) provides parameters:

**'X,** **Y** **and** **Z** **coordinate** **(um)** **'** set respectively x, y and z-coordinates of centre of uncaging circle. **'Uncaging** **radius** **(um)** **'** sets the radius of uncaging circle.

33

**'Uncaging** **onset** **(ms)** **'** sets the onset of uncaging.

**'Glutamate** **concentration** **Max** **(mM)'** sets the maximum glutamate concentration during uncaging. **'Rise'** **and** **'Decay** **tau** **(ms)'** set the dynamics of glutamate uncaging.

**Simulating** **potassium** **dynamics** **inside** **and** **outside** **astroglia**

This option is to simulate the dynamics of intracellular [K+] resulting from local K+ input and intracellular redistribution and efflux / leakage (Fig. 26). From the main panel of ASTRO (Fig. 18) the user can activate the potassium dynamics by clicking the button **'Potassium** **Dynamics'** (Fig. 18c, bottom) which prompts the relevant windows (Fig. 26).

A standard NEURON panel (Fig. 26a) is to control simulation and visualise the outcome. The **'Potassium** **settings'** panel (Fig. 26b) provides self-explanatory parameter settings for volume-limited potassium application (local current) within a round area of the cell. Here, [K+] dynamics are computed in somatic voltage clamp configuration: the clamp parameters are indicated on the panel **'Somatic** **voltage** **clamp'** (Fig. 26a, bottom) including electrode resistance, clamp voltage and

duration. a b potassium

Intracellular

buffering is

modelled with the linear approximation current *IK* *Kp* ([*K*]*n* 1), where



]

*i*

[*K*



0

*jK* is current density, [K+]in and [K+]0 are the present and basal intracellular K+ concentration ,*Kp* (mA/cm2) is the pump rate.

**Figure** **26.** **Window** **panels** **to** **control** **and** **monitor** **intracellular** **K+** **dynamics** **and** **extracellular** **K+**

c d

**steady-state** **distribution.** **a,** Parameter settings for monitoring [K+]in and [K+]o landscape mapped onto cell

morphology. **b,** Parameter settings pertinent to volume-limited [K+]in entry and [K+]o distribution. **c,** Shape visualisation with voltage-clamp positioning. **d,**

Somatic voltage clamp applied (bottom), for the sake of simplicity.

NOTE: The present simulation feature explores the diffusion landscape of intracellular K+ while approximating the K+ entry and extrusion kinetics with first-order reactions that are independent of membrane voltage, for the sake of clarity. A more detailed simulation paradigm could include further 34

experimental constrains based on known (voltage- and concentration-dependent) K+ pumps and channels, extracellular K+ dynamics, and unclamped membrane voltage. Such membrane mechanisms, including astroglial Kir4.1 channels, are available through the standard NEURON

The monitoring panel (Fig. 27a) displays colour-coded [K+]in concentration landscapes mapped onto cell morphology. The corresponding plot (Fig. 27c) shows the time course of intracellular [K+]in at a selected location (dendrite [51]). The panel (Fig.27b) shows the steady state distribution of extracellular [K+] onto cell morphology.

a b



**Figure** **27.** **Window** **panels** **displaying** **simulated** **dynamics** **of** **intracellular** **[K+]in** **and** **extracellular** **[K+]o** **steady-state** **distribution** **mapped** **onto** **cell** **morphology** **(a)** **and** **(b),** **and** **[K+]in** **time** **course** **(c)** **at** **a** **selected** **branch** **(dendrite),** **in** **response** **to** **volume-limited** **injection** **of** **potassium** **current** **(as** **in** **Fig.** **26).**

C

Window panel **'Potassium** **setting'** provides parameter settings

**'X,** **Y** **and** **Z** **coordinate** **[K]in** **(um)'** sets centroid co-coordinates for the K+ entry area.

**'Potassium** **[K]in** **input** **radius** **(um)** **'** sets the radius of K+ entry area. **'Potassium** **[K]in** **input** **onset** **(ms)'** sets the onset of K+ entry.

**'Duration** **of** **potassium** **input** **(ms)'** sets the duration of K+ entry.

**'Peak** **K+** **current** **(mA/cm2)'** sets maximum amplitude of K+ current entry.

**'Diffusion** **coefficient** **(****m2/ms)'** sets the intracellular K+ diffusion coefficient. **'K+** **leak** **rate** **(mA/cm2)'** sets *Kp*, the K+ extrusion current density.

**'Basal** **[K+]in** **concentration** **(mM)'** sets the resting intracellular [K+] .

**'Basal** **[K+]in** **concentration** **(mM)'** sets the basic potassium intracellular concentration.

Window panel “**Extracellular** **potassium** **distribution**”

**'X,** **Y** **and** **Z** **coordinate** **[K]o** **(um)** **'** sets the centroid co-coordinates for the extracellular region of the [K+ ] fluctuation.

35

**'Potassium** **[K]o** **distribution** **radius** **(um)'** sets the radius of the extracellular K+ region. **'Potassium** **[K]o** **inside** **circle** **(mM)'** sets extracellular [K+] inside the region.

**'Basic** **[K]o** **outside** **circle** **(mM)'** sets extracellular [K+] outside the region.

36

**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