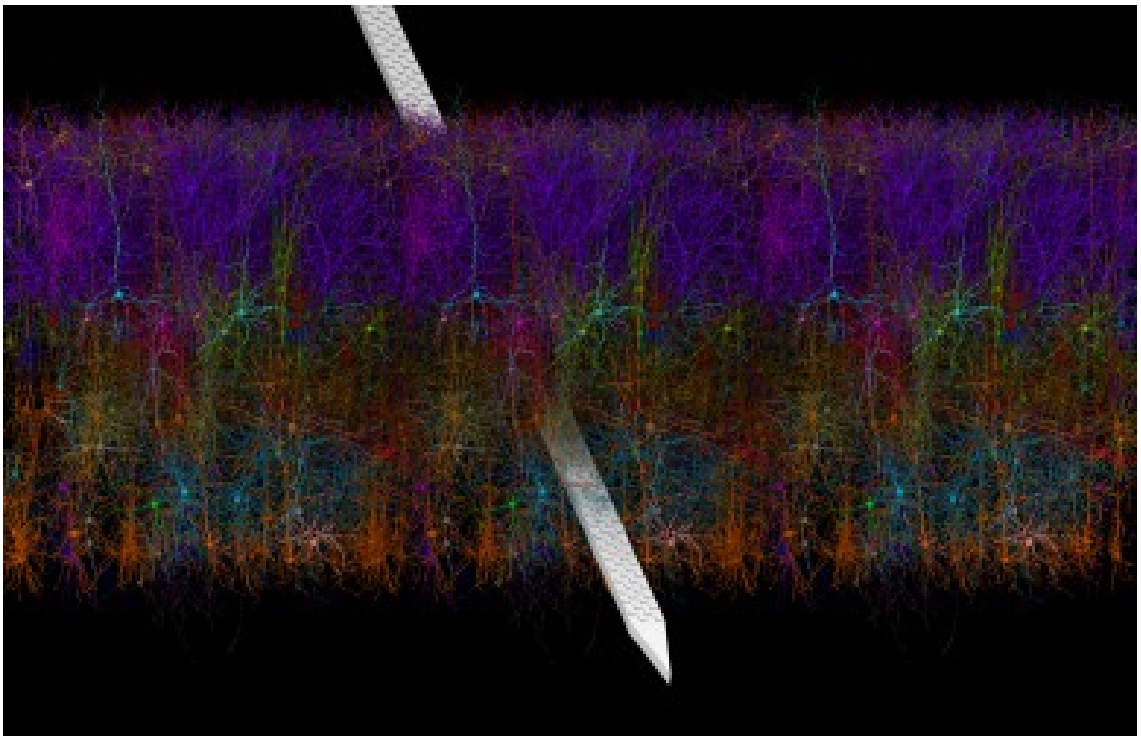
University of Illinois at Urbana-Champaign
Beckman Institute for Advanced Science and Technology
Theoretical and Computational Biophysics Group
Computational Biophysics Workshop

Visual Neuronal Dynamics Tutorial



Tutorial by Mariano Spivak, Barry Isralewitz, Jason Sharma, and John E. Stone

July 2023

A current version of this tutorial is available at
http://www.ks.uiuc.edu/Research/vnd/
Join the `tutorial-l@ks.uiuc.edu` mailing list for additional help.

Contents

1 Introduction

VND is a neuroscience-specialized adaptation and repackaging of the popular molecular visualization program VMD. VND supports the SONATA neuronal network modeling file format and can create visualizations of the structure and dynamics of neuronal simulations. Key features of VND include:

- Support for Linux.
- Support for multicore processors.
- Support for GPU-accelerated computation.

`http://www.ks.uiuc.edu/Research/vnd`

1.1 Downloading VND

Before staring the tutorial you need to download the current version of VND. This tutorial requires VND version 1.9.4a53 or later. VND supports Linux and macOS, and can be obtained from the VND homepage shown below. Follow the instructions online to install VND on your computer.

`http://www.ks.uiuc.edu/Research/vnd/vnd-1.9.4/files/alpha/`

1.2 Tutorial Topics and Files

The tutorial contains several sections, each section acts as an independent tutorial for a specific topic, with the section layout as shown in Contents. We suggest that readers with no prior VND experience work through the sections in the order they are presented. Readers already familiar with the basics of VND may selectively pursue sections of their interest. Several files have been prepared to accompany this tutorial. You will need to download these files at `http://www.ks.uiuc.edu/Training/Tutorials/vnd`.

1.3 BMTK and SONATA

The Brain Modeling Toolkit (BMTK) is a python-based software package for building, simulating and analyzing large-scale neural network models. It supports the building and simulation of models of varying levels-of-resolution; from multi-compartment biophysically detailed networks, to point-neuron models, to filter-based models, and even population-level firing rate models. For more details visit `https://alleninstitute.github.io/bmtk/`

The SONATA Data Format is a Scalable Open Data Format for multiscale neuronal network models and simulation output, jointly developed by the Allen Institute for Brain Science (AIBS) and the Blue Brain Project (BBP) of the École polytechnique fédérale de Lausanne (EPFL). The SONATA Data Format provides:

- Facilities for representing nodes (cells) and edges (synapses/junctions) of a network. It uses table-based data structures, hdf5 and csv, to represent nodes, edges and their respective properties.
- A JSON-based file format for configuring simulations, including specifying variables to record from, and stimuli to apply.

For more details visit `https://github.com/AllenInstitute/sonata`

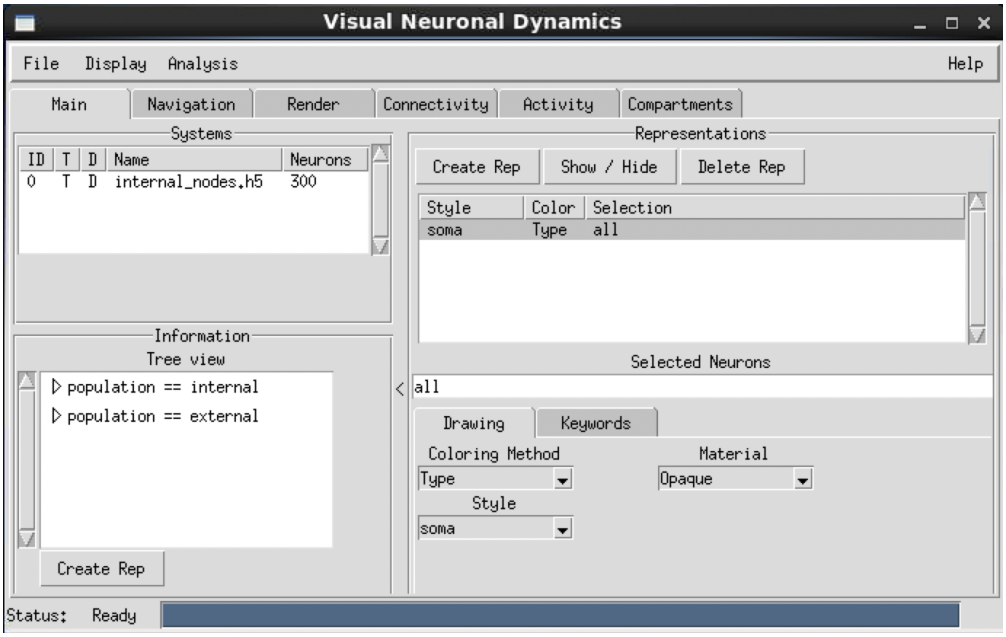


Figure 1: Visual Neuronal Dynamics window, showing the 'Main' tab with 'Systems', 'Information' and 'Representations' interfaces.

2 Working with a single model

In this section you will learn the basic functions of VND. We will start with loading a model and displaying the neuron(s), by adding graphical representations based on a selection. This section uses an example model from the SONATA Github repository (300_cells). Files are provided in the VND tutorial folder. This is a small model, comprised of 300 (internal) neurons and can help understand modeling with SONATA and BMTK.

2.1 Loading a Model

The first step is to load the model. Based on the SONATA file format, the **circuit_config.json** file contains the manifest of the files required to represent the model.

1. Start a VND session.
2. In the VND Main window, choose 'File' → 'Open File'.
3. Browse to the directory '300_cells' and select **circuit_config.json**. Click 'Open'.
4. Now the VND window should populate with details from the model (Fig. 1), which should appear with a default representation in the OpenGL window (Fig. 2).

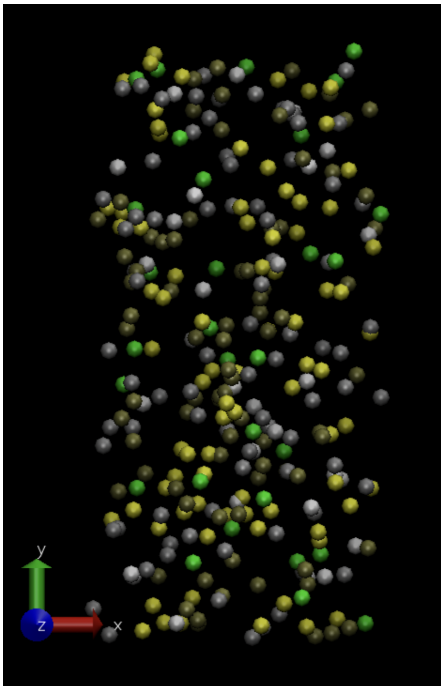


Figure 2: VND OpenGL window showing 300_cell example model. Default representation: selection 'all', style 'soma', coloring method 'Type' and material 'Opaque'.

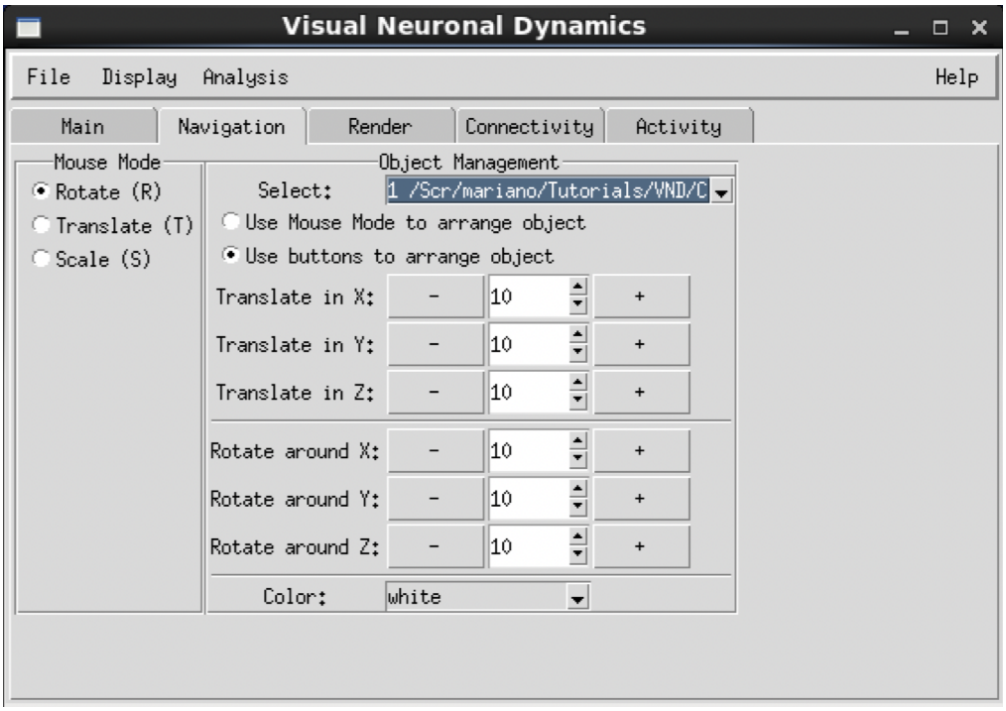


Figure 3: VND window showing the 'Navigation' tab, which contains the 'Mouse Mode' and 'Object Management' interfaces.

2.2 Navigation on the Display window (OpenGL)

In order to see the 3D structure of our model, we will use the mouse in multiple modes to change the viewpoint. VND allows users to rotate, scale and translate the viewpoint of your molecule.

1. In the OpenGL Display, press the left mouse button down and move the mouse. Explore what happens. This is the rotation mode of the mouse and allows you to rotate the molecule around an axis parallel to the screen.

2. If you hold down the right mouse button and repeat the previous step, the rotation will be done around an axis perpendicular to your screen. (For Mac users, the right mouse button is equivalent to holding down the command key while pressing the mouse button).

3. In the VND Main window, look at the 'Navigation' tab for the Mouse Mode (Fig. 3). Here, you will be able to switch the mouse mode from Rotation (R) to Translation (T) or Scale modes (S).

4. Choose the Translation mode and go back to the OpenGL Display. You can now move the molecule around when you hold the left mouse button down.

5. Go back to the Mouse Mode and choose the Scale mode this time. This will allow you to zoom in or out by moving the mouse horizontally while holding the left mouse button down. While in the OpenGL window, you can switch between the mouse modes pressing the corresponding hotkey (R, T or S) in your keyboard.

In the VND window, select the 'Display' → 'Reset View' menu item to return to the default view. You can also reset the view by pressing the "=" key when you are in the OpenGL Display window.

2.3 Representations of neurons

VND can display your model in various ways, using the Representations menu in the 'Main' tab (Fig. 1). Each representation is defined by four main parameters: the selection of neurons included in the representation, the drawing style, the coloring method, and the material. The selection determines which part of the model is drawn (e.g. 'all'), the drawing style defines how the neurons are represented (e.g. as spheres or full morphology), the coloring method gives the color of each part of the representation, and the material determines the effects of lighting, shading, and transparency on the representation. Let's first learn how to create, show/hide and delete representations.

In the VND window, select the 'Main' tab. On the right, a section called 'Representations' shows the controls to define the current representation displaying your model. In the top panel, three buttons control the creation of new representations, whether to hide or show them, and to delete unwanted representations. The middle table present the available representations with minimal details and whether they are active (shown, in black) or not (hidden, in red). Clicking on a representation from the table allows to change its selection, style, coloring method or material.

2.3.1 Create - show/hide - delete representations

1. Click on the representation that appears on the table (given by the characteristics: 'soma', 'Type', 'all'). The table highlights the selected representation. Then, click on the 'Show / Hide' button to toggle the view of this representation. Check that a hidden representation disappears from the OpenGL window and its font changes to red in the representation table, instead of black for shown representations. Click the button to show the representation on the OpenGL.

2. Create a new representation using the button 'Create rep'. By default, it copies the previously selected representation but we are going to explain how to make changes in the next sections. Having multiple representations provides mixture of different selections with different styles and colors, all displayed at the same time.

3. Finally, click and select a representation and then press the 'Delete Rep' button to remove it.

2.3.2 Displaying different selections

1. Let's see how to change the selection of a representation. Click on the representation in the table and then edit the 'Selected Neurons' entry to write 'soma x $<$ 0' and press enter. You should now see this change reflected in the OpenGL window, where only part of the neurons are represented (Fig 4, left). The full list of keywords for a selection query is provided in Appendix A.

2. Create another representation and change the selection to 'stride 100', which means draw 1 every 100 neurons. This is useful when working with large models or when looking to see individual neurons with full morphology. We are going to explore this drawing style soon.

3. Finally, create a new representation and change its selection to '(soma x $>$ 0) && (soma y $>$ 0)'. This is an example of how to add syntax to create complex selections. Now use the 'Show / Hide rep' button to hide the representations leaving only one visible each time, and compare the differences in the selections.

2.3.3 Browsing reserved and non-reserved attributes

1. Click on the 'Keywords' tab beneath the 'Selected Neurons' entry. Observe that there are 3 columns present: 'Attributes', 'Values', and 'Min/Max'

2. Scroll through the first column labeled 'Attributes' to observe various reserved and non-reserved properties. Reserved attributes apply to all neuronal models (files) as opposed to non-reserved which are model (file) specific.

3. Select an element in the 'Attrtibutes' column to display values of each selected attribute.

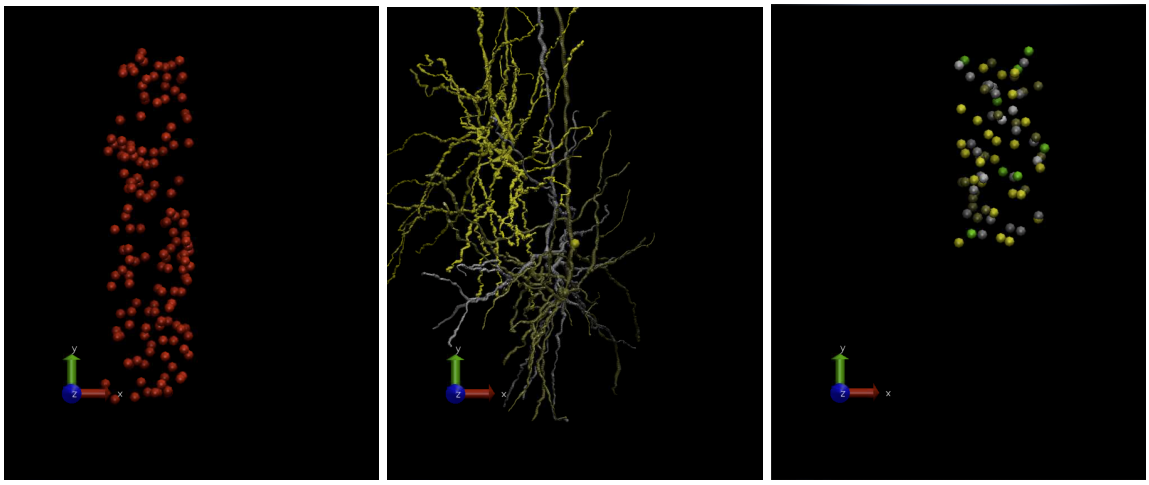


Figure 4: VND OpenGL window showing 300_cell example model. Modified representations: (left) selection 'soma x < 0', style 'soma', coloring method 'Red'. (middle) selection 'stride 100', style 'morphology_sph', coloring method 'Type'. (right) selection '(soma x > 0) && (soma y > 0)', style 'soma', coloring method 'Type'. All representations have material 'Opaque'.

4. Double-clicking an element in the 'Attributes' or 'Values' column appends that text to the 'Selected Neurons' entry. This assists query construction with quick and error-free addition of attribute names and values.

2.3.4 Exploring different coloring methods

In the current version of VND there are two coloring methods available for our representations.
The default coloring method is 'Type'. In this coloring method each individual neuron is colored by the type of neuron present in the data. The other coloring method is 'Color', which opens another menu to select the color from an extensive list.

1. Click on a visible (black font) representation and choose 'Coloring Method' → 'Color' and select 'Red'.
2. Try coloring each representation in a different color.

2.3.5 Exploring different drawing styles

In the current version of VND, there are two drawing styles for neurons. 'soma' depicts neurons as spheres, whereas 'morphology' shows full morphology details when available in the model. There are other drawing styles used to show neuron connectivity, but that will be covered in section 4.

1. In order to see details of the full morphology, let's work with a representation that shows a few neurons ('stride 100').
2. Select the 'stride 100' representation in the table and 'Style' → 'morphology' to change its style. In the current version, this drawing style is very computationally demanding and can take a little time to process depending of the number of neurons required.
3. Use the 'Show / Hide' button to toggle between the representations to have a better view of the different styles.

2.3.6 Exploring different materials

In the current version of VND, we provide a list of pre-defined options for materials. Each material interacts differently with light and gives representations different appearances.

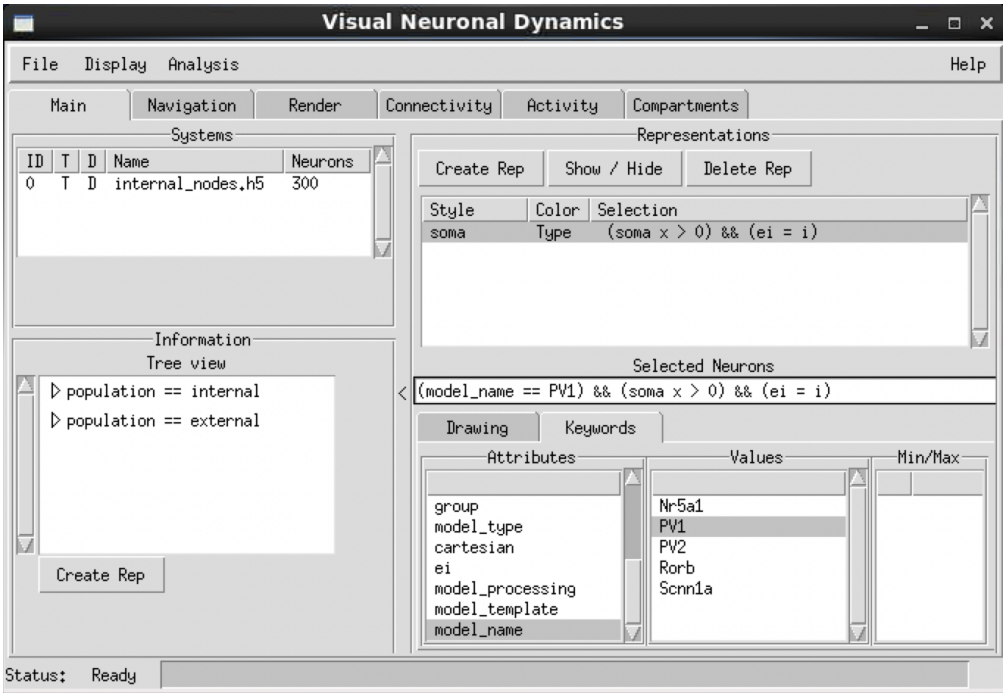


Figure 5: Double-clicking on elements in the 'Attributes' and 'Values' tables appends that item to the 'Selected Neurons' entry field.

1. Click in on a representation and select 'Material' → 'Transparent'. Now the surfaces defining that representation are clear or see-through.
2. Explore different materials for your representations. Take into account that some of them might not look that different in the OpenGL. This is because some materials (e.g. Glass, AOShiny) require to activate advanced render options before noting all its features. We are going to see the advanced render options in section 7.

2.4 Saving your work

The viewpoints and representations that you have created using VND can be saved as a VND state. This VND state contains all the information needed to reproduce the same VND session without losing what you have done.

1. Save the current VND state. Go to 'File' → 'Save Visualization State'. Choose a file name a click save.
2. Then, close VND and start a new VND session.
3. Load the previous VND state in the new instance. Go to 'File' → 'Load Visualization State'. Select the file you saved and click open. You should be able to see the same session you saved.

3 Adding objects to the scene

Creating a composite image is possible in VND. Geometric forms can be added to the scene as 'PLY' files. In the following example, we will add a pixel probe to our 300_cell example model.

3.1 Loading a PLY file

1. Go to 'File' → 'Add Object'. Browse to the 'neuropixels_probe' folder in the VND tutorial files, and select the file **shank_to_scale_shortened_c.ply**.

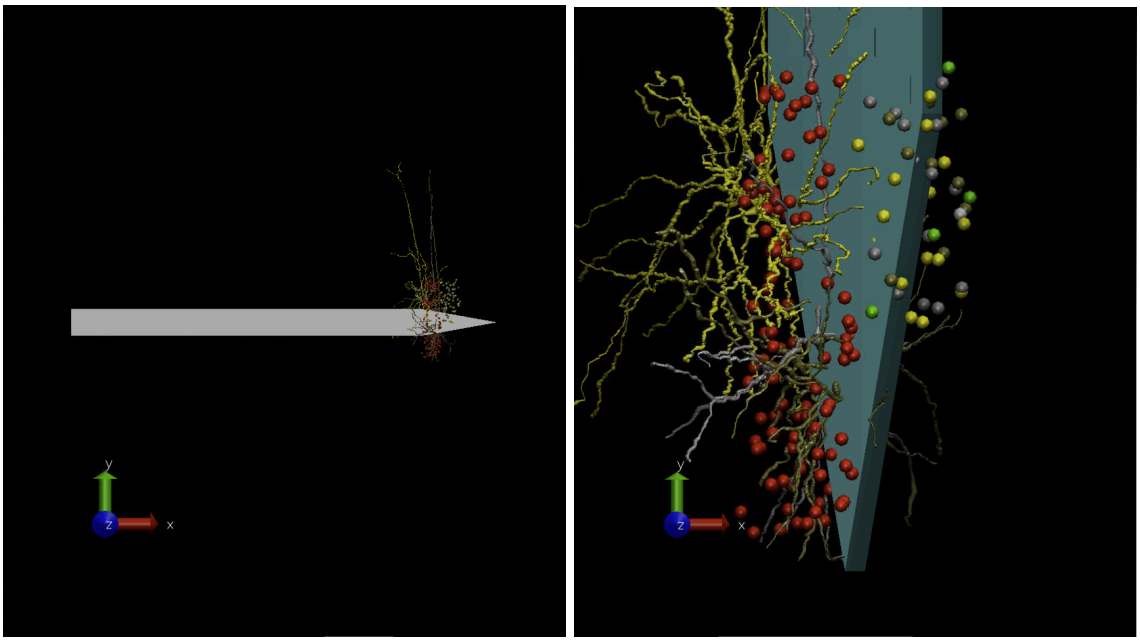


Figure 6: VND OpenGL window showing the 300_cell example model and a Neuropixels probe. (left) initial configuration after loading the object. (right) scene composition after object manipulation.

2. The initial position of the probe is given by the file, but VND allows you to move the shape using the 'Navigation' tab.

3.2 Object management

1. In the 'Navigation' tab select the object you want to interact with.
2. Let's use the buttons to translate and rotate the object so it fits close to our model.
3. In the 'Rotate around Z' line, edit the entry value to 90 and press the '-' button. This should rotate the probe -90^o on the Z axis, and make the probe parallel to the Y axis.
4. In the 'Translate in Y' line, edit the entry value to 60 and press the '+' button. This translates the probe 60 units along Y.
5. Rotate the probe -10^o on the Y axis to see some of the probe geometrical features.
6. Finally, change the color of the probe to 'cyan'

4 Displaying connections between neurons

In this section, we will learn how to draw the connections (edges) between neurons present in the model. We will continue using the '300_cell' example model and the same representations we created in section 2.

4.1 Loading Edges information

The connectivity information in the model is present in the same **circuit_config.json**. This time, we will load this data and visualize it.

1. Start a new VND session or go to 'File' → 'Reset VND'
2. Go to 'File' → 'Open File with Edges'.

3. Browse the directory and select **circuit_config.json**. Click 'Open'.
4. VND should load the model with the edges information and define the default 'all' representation.
5. Let's create the same representations we used in section 2. Here you can practice what you learnt, or load the visualization state with the representations' information provided by the file **loadreps.tcl**.
6. Delete the default 'all' representation if needed.

4.2 The Connectivity tab

The connectivity tab helps on the display of edges between neurons by guiding the user on the creation of connection representations. There are three sections in the tab: 'Source', 'Target' and 'Representation configuration'. The first two define the selection of neurons to evaluate connectivity in the model. The last one provides a way to configure the representation. Let's explore this tool with an example.

1. Click on the 'Connectivity' tab.
2. On the 'Source' section, use the drop-down menu to use the existing selection 'stride 100'.
3. On the 'Target' section, use the drop-down menu to use the existing selection '(soma x $>$ 0) && (soma y $>$ 0)'.
4. Change the color of the representation to 'white'.
5. Click the 'Create connection rep' button. You should be able to see edges drawn in white in the OpenGL window (Fig. 7, left).
6. In this version of VND, connection representations are shown in the 'Representations' table in the 'Main' tab. You can use the 'Show / Hide' and 'Delete Rep' buttons but you can't edit the connection rep using the options. If you want to change something of the connection rep, you need to create a new one in the 'Connectivity' tab.
7. Create another connection representation in color pink, but this time exchanging the source and target. Compare the two connection representations, use the 'Show / Hide' button in the 'Main' tab to help you. Figure 7 (middle) shows this representation.
8. So far we've been using the default style for connectivity representations, 'simple_edge', which draws a cylinder connecting the two neuron selections. There are other styles available that provide complementary information about the edges of the model. Table 1 provides the explanation for the styles available in VND.
9. Let's create another connection representation, in blue, with the same 'Source' and 'Target' but with the style 'source_soma'. Figure 7 (right) shows all the connection representations created in this tutorial together. This new representation helps to highlight in blue those neurons that do have a connection, among the those that were selected as a source (spheres in other colors).

5 Displaying spike activity

In this section, we will learn how to load and visualize spike activity data into our model within VND. We will continue using the '300_cell' example model.

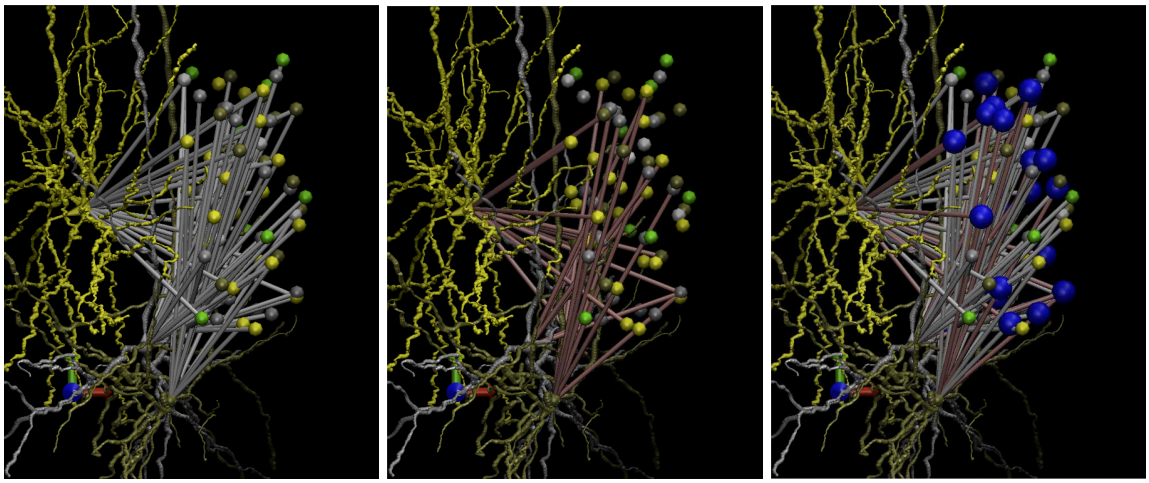


Figure 7: VND OpenGL window showing 300_cell example model with drawn edges. Connection representations: (left) source 'stride 100', target '(soma x > 0) && (soma y > 0)', coloring method 'white', style 'simple_edge'. (middle) source '(soma x > 0) && (soma y > 0)', target 'stride 100', coloring method 'pink', style 'simple_edge'. (right) All of the above and source '(soma x > 0) && (soma y > 0)', target 'stride 100', coloring method 'blue', style 'source_soma' and 'Sphere Scale' 5. All representations have material 'Opaque'.

Table 1: Explanation of connection representations styles available in 'Connectivity' tab in VND.

Style	Explanation
simple_edge	Draws cylinder connecting neuron selections
source_soma	Draws spheres on source neurons' somas
target_soma	Draws spheres on target neurons' somas
source_target_soma	Draws spheres on both source and target neurons' somas
simple_edge_swc	Draws cylinder connecting morphology synapse locations
source_sphere_swc	Draws spheres at source morphology synapse locations
target_sphere_swc	Draws spheres at target morphology synapse locations
source_target_sphere_swc	Draws spheres at source and target morphology synapse locations

5.1 Loading spike information

1. Start a new VND session or go to 'File' → 'Reset VND'
2. Go to 'File' → 'Open File'.
3. Browse the directory and select **circuit_config.json**. Click 'Open'.
4. VND should load the model and define the default 'all' representation. But we need to specify the file containing the spike data.
5. Go to 'File' → 'Add File with Spikes'.
6. Browse into the 'output' directory and select **spikes.h5** file. Click 'Open'.
7. We are going to use the 'Activity' tab to display animations of the spike activity data.

5.2 The Activity tab

The 'Activity' tab allows the user to visualize spike activity data of the neurons in the model (Fig. 8). In the current version of VND, it is possible to define a population of neurons and (optionally) a selection using

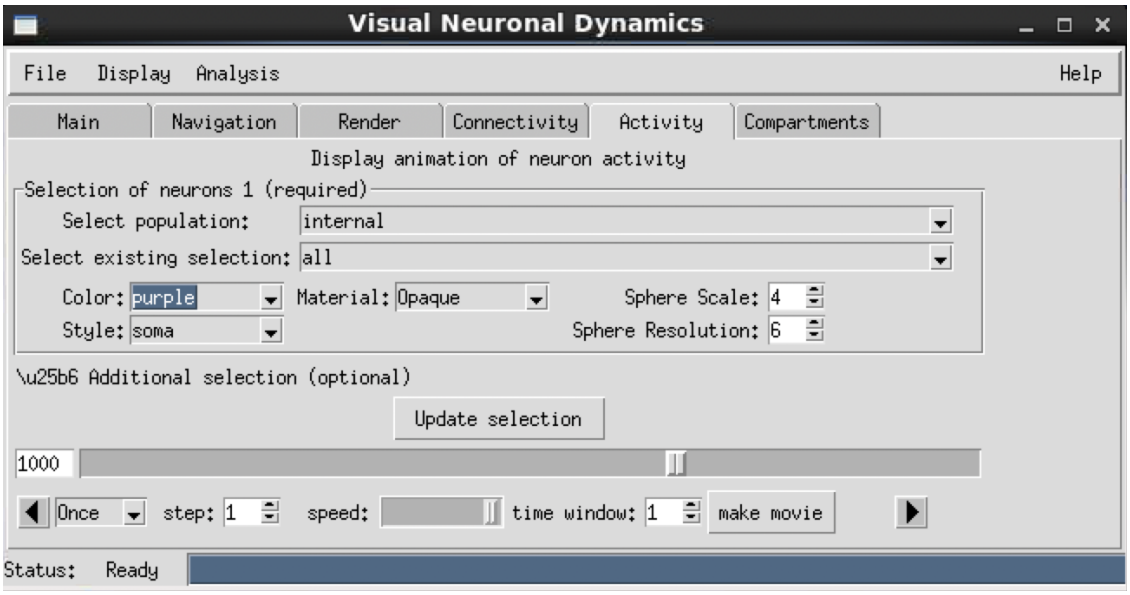


Figure 8: VND window with 'Activity' tab. Menus define selection of neurons for spike activity (e.g. 'all'), configuration (e.g. color 'purple', style 'soma') and animation playback (e.g. step '5', time window '5'). Now featuring a drop-down menu to have additional selections.

the selection language defined in Appendix A. The tab also presents a menu to configure the representation generated for the animation. Finally, the tab contains a playback menu with buttons to 'play' / 'pause' / 'play backwards' and options to control the 'step', 'speed' and 'time window' for the animation.

Let's do an example visualization:

1. Define the 'Population' entry as 'internal'
2. Since this is a simple model, let's use the default selection 'all'.
3. Click 'Update selection'. This generates the animation which then can be customize in the 'Representation configuration' menu. Any time we modify the 'Population' and / or 'Selection', we need to use the button.
4. Let's use the default style, color and material for now.
5. Click on the 'play' button on the bottom right side of the playback menu. You should be able to see the time increasing by the 'step' value.
6. Press 'pause' and change the color to 'purple'.
7. You can use the playback bar to scroll across the animation. Also, you can edit the entry showing the timestamp to go to a specific time in the animation. Go to the animation frame '1000'. Figure 9 shows the resulting animation frame.

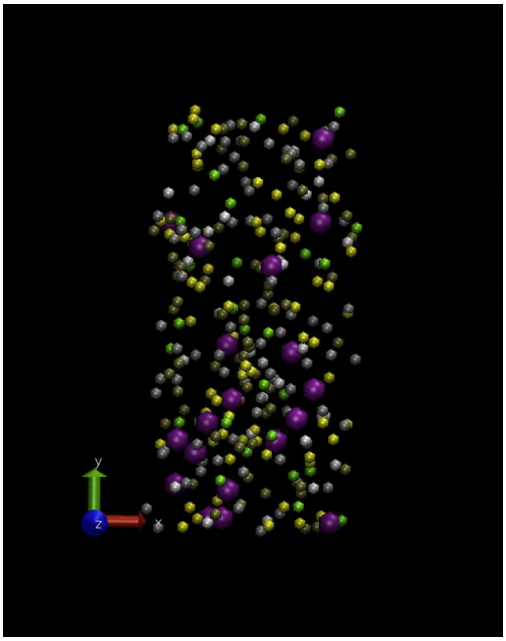


Figure 9: VND OpenGL window showing spike activity animation of 300_cell example model, corresponding to Fig. 8 details.

5.3 Analysis

The analysis of spike data can be achieved by loading a raster plot. This visualization technique displays individual neuron firing events over time, with coloring by neuron type. To load a raster plot:

1. Ensure the file circuit_config.json is loaded as described above.
2. Check that 'Add File with Spikes' has been performed.
3. Go to 'Analysis' → 'Raster Plot'
4. Select the 'internal' population
5. Choose selection 'all'
6. Pick color by 'Type'
7. A new window should appear, labeled 'MultiPlot' / 'Neuronal Spike Activity', containing the raster plot (see Figure 10)
8. Freely move the raster plot legend by clicking and dragging the legend to the chosen position.

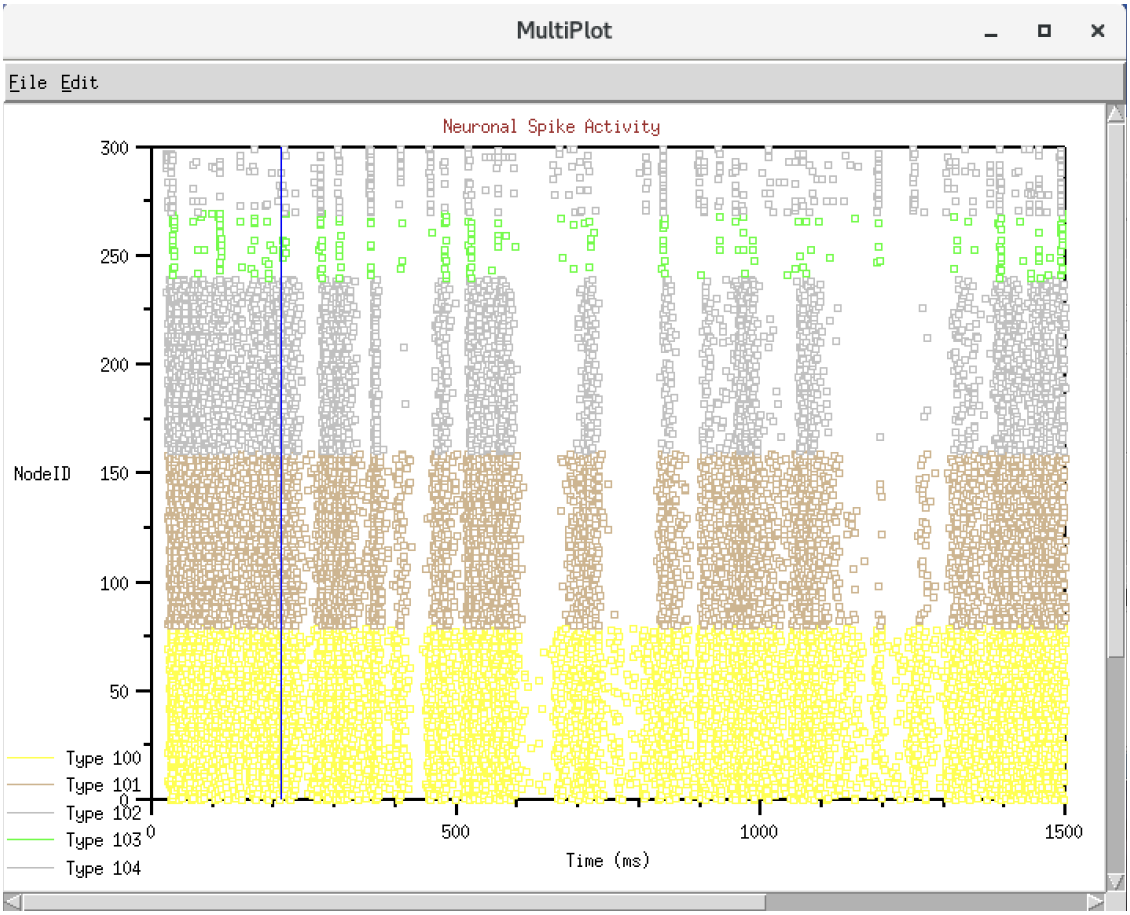


Figure 10: A raster plot shown of 'all' selected neurons. Each individual point is specified by nodeID plotted against time. The blue vertical line is an indicator of the current animation frame inside the VND engine.

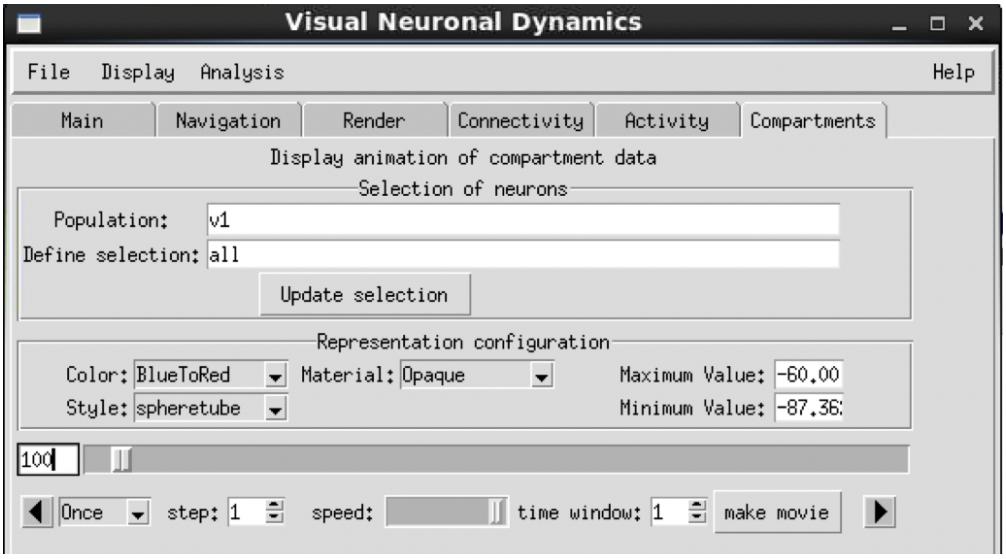


Figure 11: VND window with 'Compartment' tab. Menus define selection of neurons for compartment data (e.g. 'all'), configuration (e.g. color 'BlueToRed', style 'spheretube') and animation playback (e.g. step '5', time window '5').

6 Displaying compartment data

In this section, we will learn how to load and visualize compartment based data into our model within VND. We will use an example model consisting of two neurons, present in the folder '2_cells'.

6.1 Loading model and compartment data

1. Start a new VND session or go to 'File' → 'Reset VND'
2. Go to 'File' → 'Open File'.
3. Browse to the **2_cells** directory and select **config.network.json**. Click 'Open'.
4. VND should load the model and define the default 'all' representation. But we need to specify the file containing the compartment data.
5. Go to 'File' → 'Add File with Compartment Data'.
6. Browse into the **2_cells/output** directory and select **v_report.h5** file. Click 'Open'. This file contains data of voltage over time per compartment. (The file may take a minute or so to load.)
7. We are going to use the 'Compartment' tab to display animations of the aforementioned data.

6.2 The Compartment tab

The 'Compartments' tab allows the user to visualize compartment specific data of the neurons in the model (Fig. 11). In the current version of VND, it is possible to define a population of neurons and (optionally) a selection using the selection language defined in section A. The tab also present a menu to configure the representation generated for the animation. Finally, the tab contains a playback menu with buttons to 'play' / 'pause' / 'play backwards' and options to control the 'step', 'speed' and 'time window' for the animation.

Let's do an example visualization:

1. Define the 'Population' entry as 'v1'
2. Since this is a simple model, let's use the default selection 'all'.
3. Click 'Update selection'. This generates the animation which then can be customize in the 'Representation configuration' menu. Any time we modify the 'Population' and / or 'Selection', we need to use the button.
4. Let's use the default style, color and material for now.
5. Adjust the 'Maximum Value' to -60.00, improving the color scale changes for this particular example.
6. Select the Main tab, select the 'all' representation, and select 'Display' → 'Reset View'
7. Select the 'Compartments' tab again, change the 'step' value to 50 (double-click to type value into field), in order to glance over the data faster.
8. Click on the 'play' button on the bottom right side of the playback menu. You should be able to see the time increasing by the 'step' value.
9. You can use the playback bar to scroll across the animation. In the playback bar: drag with left mouse button; jump to a new position with the middle mouse button. Also, you can edit the entry showing the timestamp to go to a specific time in the animation. Go to the animation frame '100'. Figure 12 shows the resulting animation frame.

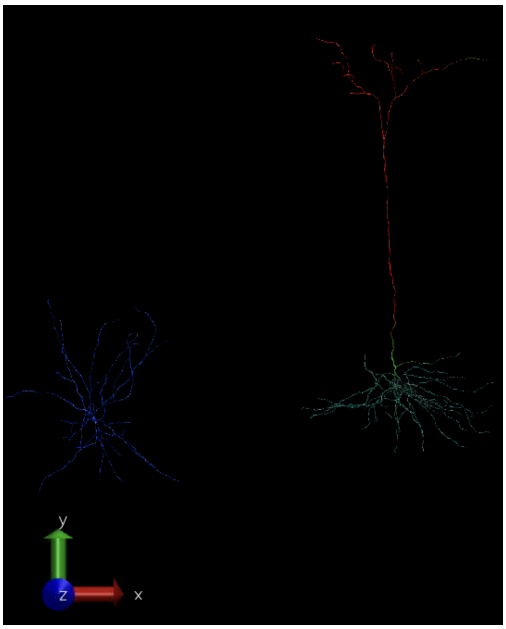


Figure 12: VND OpenGL window showing compartment data animation of two cell example model, corresponding to Fig. 11 details.

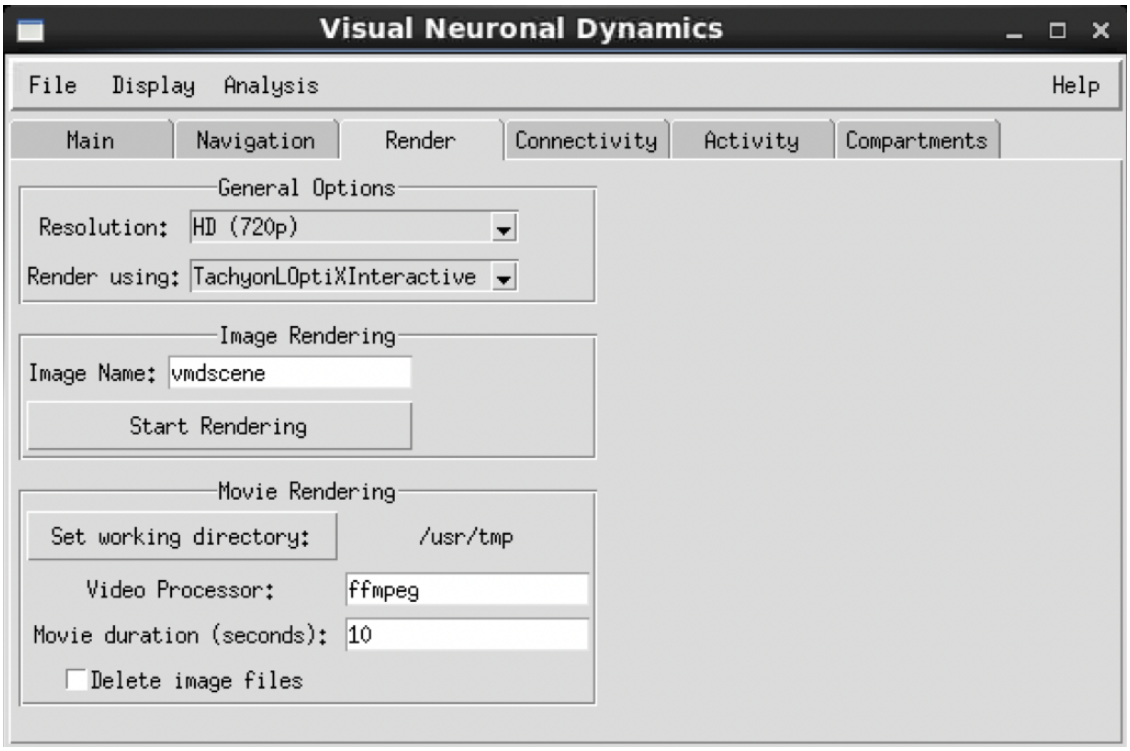


Figure 13: VND window with 'Render' tab. The options define OpenGL resolution and render selection to generate high quality images.

7 The Basics of VND Rendering

The current version of VND provides various options to customize the display. In the main VND window, the 'Display' menu allows to select between 'Perspective' or 'Orthographic' view, as well as other visualization effects. In particular, 'Display Settings' menu controls clipping planes and advanced visualization options like Ray Tracing, which are available while rendering an image.

7.1 The Render tab

The 'Render' tab provides controls to generate high quality rendering of your VND visualization. Depending on your available computational resources, different render options might be available. The 'Resolution' option gives a series of predefined resolutions for the OpenGL window. Any other window size can be achieved by extending / arranging the window itself. Let's explore the render options with an example:

1. Select 'HD (720p)' for resolution.
2. Select render using 'TachyonLOptixInteractive', or other 'Tachyon' option if available. (Other options: 'TachyonInternal', 'TachyonOpixLInternal', or screen-quality 'snapshot')
3. Choose a file name: for example 'model.png'.
4. Click 'Start Rendering'. If using 'TachyonLOptixInteractive', this will open an interactive ray tracer mode, which works similar to the OpenGL window to rotate, translate and scale. This advanced rendering supports options like 'shadows', 'ambient occlusion' and 'depth-of-field' present in the 'Display Settings' menu.
5. Once you are satisfied with the view, click 'Esc' in your keyboard. The final image will open with ImageMagick, close it to continue working with VND.

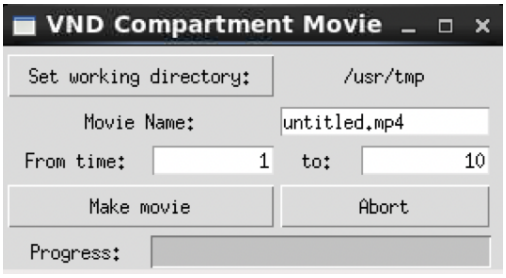


Figure 14: Example of the VND movie control window for the compartment data animation. The user can define the working directory, the root name of the files generated, the animation interval, and also monitor the progress of the rendering.

7.2 Movie rendering

VND provides a simple way of rendering the built-in animations into a series of images, that can be paste together using an external video processor (e.g. ffmpeg for linux and macos). The 'Render' tab provides a place to customize default parameters such as the working directory for creating the images, path of the video processor, the movie duration in seconds and the option to delete all the generated image files after the movie is completed.

In order to create the rendering for an animation, the 'Activity' and 'Compartments' tabs have a 'make movie' button, that once pressed pops-up a window called 'VND Movie control' (see Figure 14). This window controls the rendering process for the animation, by defining important parameters like the working directory, movie name, animation interval, and monitoring the progress of the movie. Let's try an example movie generation in VND.

1. If necessary, follow the steps to load the two neurons example referred to in section 6.
2. Define the appropriate population (v1) and selection (all), and click 'Update selection'.
3. Adjust the 'Maximum Value' to -60.00.
4. Select 'step' as 50.
5. Click 'make movie'. This will open the VND movie control.
6. If necessary, modify the working directory and movie name.
7. Select the animation time interval from 1 to 1000. Take into account that VND will use the 'step' defined in the animation controls.
8. Make sure the OpenGL window displays the model system.
9. Finally, click 'Make movie' again and wait till the rendering is finished.
10. Close the Movie control window.

A Appendix: Neuron selection language

A.1 Logic and grouping

Logical operators:
And: `&&`
Or: `||`
Not: `!`

Group search terms by using parenthesis: `(` , `)`.
Nesting is permitted.

Examples:

```
(soma x < 0) && (population == LGN)
(node_id > 100) || ( (soma x > 0) && (population == LGN) )
```

A.2 Neuron selection keywords

search term	definition	comparators	example
`soma x | y | z`	pos. of neuron center	`<, >, ==`	`soma x > 0`
`morphology x | y | z`	pos. of any segment of morphology	`<, >, ==`	`morphology x > 0`
`type`	node ID	`==`	`type == 2003`
`node_id`	node ID	`<, >, ==`	`node_id < 100`
`node`	node ID (synonym)	`<, >, ==`	`neuron_id < 100`
`population`	population name	`==`	`population == LGN`
`group`	group number	`<, >, ==`	`group == 3`
`fileset`	file set number in circuit config.	`<, >, ==`	`fileset == 2`
`stride`	select every Nth neuron	N/A	`stride 50`
`within`	within N units of a neuron	N/A	`within 10 of node_id 1073`
non-reserved attributes	per-model non-standard attributes	`==`	`ei == e`

Notes:
The comparators `<=`, `>=`, and `!=` are not yet permitted. Use `!(A == B)` instead of `(A != B)` .
The `within` term currently only works with `node_id`/`node`, e.g. `within 10 of node_id 1073`. Combine with `population` and similar terms to choose specific neurons in multi-population systems.

A.3 Neuron attribute query construction

Query construction can be performed by double-clicking elements in the columns located in the 'Keywords' tab on the 'Main' screen tab. Search terms and values can be paired with comparators to build a unique query. See section 2.3.3 for details.