

Neuron_Morpho plugin for ImageJ

Version 1.1.6 - May 25, 2007

Giampaolo D'Alessandro
Department of Mathematics, University of Southampton,
Southampton, SO17 1BJ, England, UK

1 Introduction

The Neuron_Morpho plugin for ImageJ allows the user to measure the x , y and z coordinates of a selected point of a stack of confocal images and the radius of the corresponding section of the neuron together with other information that can be used to reconstruct the neuron morphology. The measurement output is in *swc* format. It is possible to transform it in a *hoc* file for the program **neuron** (written by Ted Carnevale and Michael Hines, <http://www.neuron.yale.edu/neuron/newhome.html>) using the neuron morphology viewer **cvapp** (written in Java by Robert Cannon and available at <http://www.compneuro.org>).

2 Quick guide

Once the plugin has been installed (see the Downloading and installation section) we can measure the morphological data of a given neuron whose confocal images are stored in an ImageJ stack. The steps involved are (see following sections for more detailed explanations):

1. Load the stack of images using the *Open* entry in the *File* menu.
2. Push the straight line selection button in the tool bar of ImageJ. This step is not essential in order to use the plugin, but it makes its use easier.
3. Load the plugin from the *Plugins* menu of ImageJ.
4. Once the plugin window appears press the *Start* button.
5. Identify in the stack of images a section of the neuron whose coordinates you wish to measure.
6. Press the Shift key and keep it pressed for all the time that the measurement is taken. [**Note:** in some versions of ImageJ, pressing the shift key forces the straight line selection to be either horizontal or vertical. This has no effect on the plugin, but it makes visualisation of the measured section rather hard. In this case it may be more convenient to use the freehand line selection.]
7. Move the mouse to one side of the section; press the left button and drag the mouse across to the other side of the section. Release the left button. A row of morphological data including the coordinates and the radius of the section will appear in the table.
8. Repeat steps (5) to (7) for all the points of the neuron that you wish to identify.
9. Save the data to a file using the *Save* button of the Neuron_Morpho plugin.

3 Downloading and installation

In order to use the Neuron_Morpho plugin ImageJ must have been installed. It can be downloaded from <http://rsb.info.nih.gov/nih-image/>. If one wishes to compile the plugin then also a suitable version of Java must have been installed (see the ImageJ documentation for further information).

The plugin, this manual and all the files used in this example can be downloaded from the Neuron_Morpho plugin home page at the URL

<http://www.personal.soton.ac.uk/dales/morpho>

The plugin is written in Java: its source code is contained in the file “Neuron_Morpho.java”. You can run the plugin by downloading only this file and, eventually, the online documentation. However, it is probably easier to download the zipped archive “Neuron_Morpho.zip”, that contains the online documentation, the executables and the source code.

3.1 Easy install

1. Download the file “Neuron_Morpho.zip” in the “plugins” subdirectory of the ImageJ directory.
2. Unpack the archive using wkzip or unzip.
3. Next time ImageJ is started an item called “Neuron Morpho” will appear in the “Plugins” menu. By selecting it the plugin will be activated.

3.2 Source install

In order for this installation to work a suitable version of Java must have been installed (see the ImageJ documentation for further information).

1. Download the file “Neuron_Morpho.java” in the “plugins” subdirectory of the ImageJ directory.
2. Start ImageJ.
3. Pull down the “Plugins” menu and select “Compile and Run...”.
4. In the “Compile and Run...” window select “Neuron_Morpho.java” and press OK. The plugin will be compiled and, if the compilation is successful, it will start.
5. Next time you run ImageJ there is no need to compile the plugin again: an item called “Neuron Morpho” will appear in the “Plugins” menu and by selecting it the plugin will be activated.
6. In order for the online documentation to be available you must also follow the “Documentation Install” steps.

3.3 Documentation install

Unless you have chosen the “Easy install” option you must install the documentation by hand.

1. Download the zipped archive “Neuron_Morpho_doc.zip” in the “plugins” subdirectory of the ImageJ directory.
2. Unpack the archive using wkzip or unzip. A subdirectory called “morpho_doc” will appear: it contains a few image files and the HTML version of this instruction manual. This can be accessed using the Help button of the plugin.

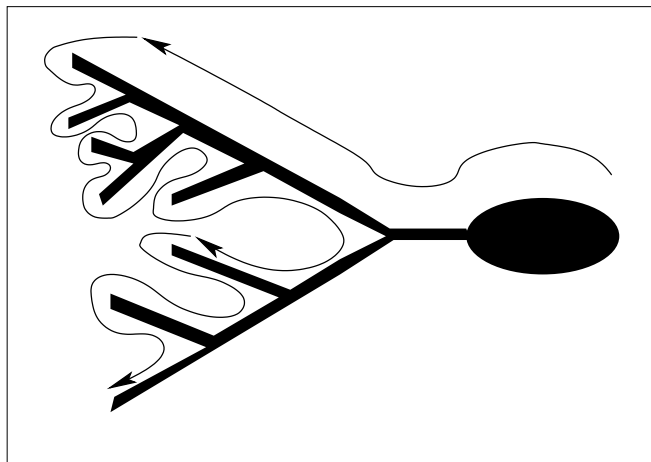


Figure 1: *Z projection of the average intensity of a stack of three slices that represent a toy neuron. The arrows indicate the direction in which the neuron is traverse when taking measurements.*

4 How to use the Neuron_Morpho plugin

4.1 Loading a stack

Load the stack of images either using the *Biorad reader* plugin (for biorad stacks) or the open command in the file menu of ImageJ (for more standard stacks of images). It is convenient to produce a z-projection (*Image/Stacks/Z project*, choose either average intensity or maximum intensity) of the neuron to use it as a map for later on (see Figure 1). If the neuron contains many small details it may be advantageous to print more detailed maps of smaller regions and/or to make projections of fewer slices.

4.2 The data format

The Neuron_Morpho plugin allows us to take measurements by drawing a straight line across the section of the neuron that we wish to represent: the plugin obtains the radius of the section by computing the length of the line and dividing it by two. The coordinates of the centre of the line are the x and y coordinates of the point. The z coordinate is the index of the current slice. The data for each represented point is in the following format (*swc format*):

$$n \quad T \quad x \quad y \quad z \quad R \quad P$$

In this format, n is an integer label (normally increasing by one from one line to the next) that identifies the point. T is an integer that represents the type of neuronal segment. The convention used by this plugin is given by the following table:

Value	Meaning	Value	Meaning
0	Undefined	4	Apical dendrite
1	Soma	5	Fork point
2	Axon	6	End point
3	Dendrite	7	Custom

The “custom” entry could be used to identify sections that have a particular meaning for a specific neuron. The user never sees the integers: the plugin automatically transforms them using the above

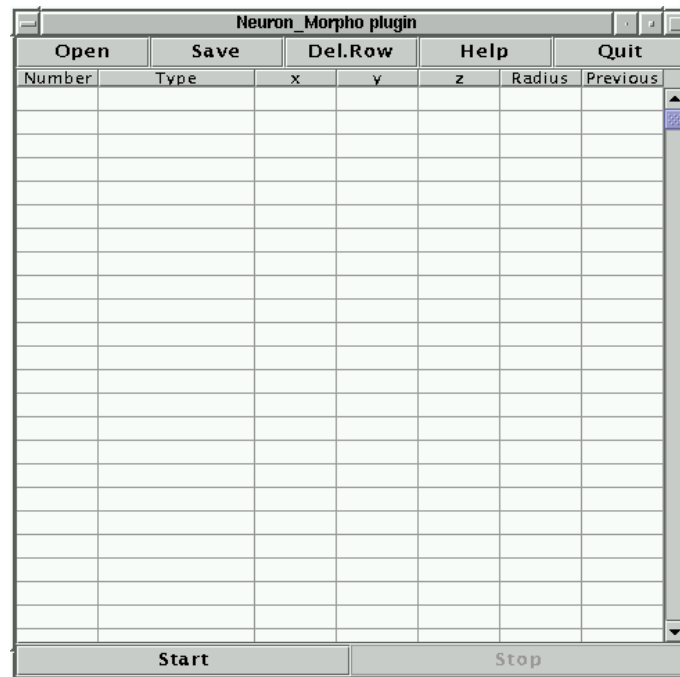


Figure 2: Window of the *Neuron_Morpho* plugin. The measurements will appear in the table.

table and prints the type of section in words. x , y and z are the three coordinates of the point of the neuron and R is the radius of the section. P is an integer that represents the point preceding the current one when moving away from the soma. It is usually the label of the point of the previous row except for the first row (where it set to -1) and at forks in the neuron (see later). The *swc* file may contain a header with comments. Each line of the header **must** start with a “#” (hash) character.

4.3 The plugin window

Start the *Neuron_Morpho* plugin by selecting it from the *Plugins* menu. A separate window should appear (see Figure 2). The centre of the window is a table whose columns correspond to the headings of the *swc* format. Each new measurement is added at the first empty line. The top row contains four buttons:

- Open** Open an *swc* file that contains previously made measurements. The data in the file will be shown in the table and the new measurements will be appended to them.
- Save** Save the measurements in an *swc* file. This file can either be loaded in *cvapp* to produce a hoc file for neuron or can be loaded again by the *Neuron_Morpho* plugin if more measurements are needed.
- Delete** This button deletes the selected row. A row in the table is selected by clicking the left mouse button on it. If more than one row is selected only the topmost row will be deleted.
- Help** This button loads the online help in a separate Help window. The plugin assumes that the help files are contained in a subdirectory of the “plugins” directory called “morpho_doc”. If this is not the case, the correct URL can be specified in the text field at the top of the Help Window.

Quit Quit the Neuron_Morpho plugin.

The bottom row contains two buttons:

Start Press this button when ready to start taking measurements.

Stop Press this button when wishing to stop taking measurements.

The start and stop buttons can be pressed as many times as one likes during a single measurement session. The new data will always be appended at the bottom of the table.

4.4 Taking measurements

The Neuron_Morpho plugin allows us to take measurements by drawing a straight line across the section of the neuron that we wish to represent. Moreover, measurements are taken **only** if the Shift key is pressed **all the time** that the mouse is dragged. This means that the image can be manipulated while taking measurements without affecting the measurement process. If we wish to have a visual aid when measuring, we should press the line selection button in the ImageJ tool bar [**Note:** in some versions of ImageJ, pressing the shift key forces the straight line selection to be either horizontal or vertical. This has no effect on the plugin, but it makes visualisation of the measured section rather hard. In this case it may be more convenient to use the freehand line selection.]. Finally, before starting taking measurements it is convenient to choose a path on the neuron along which to take the measurements. The Neuron_Morpho plugin assumes that the path chosen will be to follow a branch to its end, then backtrack to the last fork and follow the other branch to the end and so on (see Figure 1). If this strategy is followed (and the “fork” and “end points” are dutifully entered) then the plugin will be able to correctly guess the entries in the “Previous” column. However, we may take the points in whatever we want (even random): in this case we will have to set by hand the entries in the “Previous” column because it is quite likely that the suggestions provided by the plugin will be inappropriate.

For the purpose of this example we will follow the path indicated in Figure 1. The stack of images (down-loadable from the Neuron_Morpho plugin homepage) that represent the toy neuron has on only three slices. The soma is the ellipse on the right hand side and is located in the middle slice. Go to that slice. The first point of the neuron that we wish to represent is the rightmost point of the soma. Keep the Shift key pressed during the entire measurement. Go to the top right hand most corner of the soma, press the left mouse button and drag the mouse down to the bottom right hand most corner. Release the mouse button keeping the Shift key pressed. A line (of an undefined colour) should appear on the neuron to indicate the selection. We can increase the thickness of the line by using *Edit/Options/Line Width...* (see Figure 3: in producing this figure a line thickness of 6 has been used and the colour of the selection has been changed by hand. A smaller thickness is probably more appropriate when making “real” measurements.)

The following line should appear in the Neuron_Morpho plugin window (measurements taken by different people may be slightly different because they depend on the line drawn on the slice):

```
1 undefined 508.50 199.00 2.00 15.01 -1
```

The first number, 1, indicates that this is the first point, the second entry, `undefined` indicates that the plugin had no idea of what section has just been measured. This can be changed by clicking on the cell and choosing the appropriate entry from the list of options that will appear (see Figure 3).

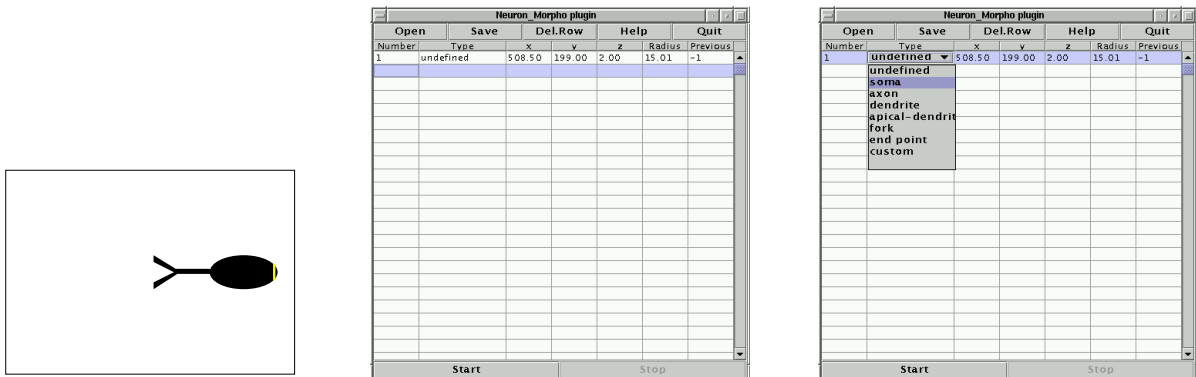


Figure 3: (Left) First measurement used to represent the neuron and corresponding entries in the plugin table. The centre panel shows the table entry immediately after the measurement has been made. The “Type” of section is undefined. This can be changed using a pull-down menu that becomes visible by clicking on the table cell.

From now on the plugin normally assumes that the type of neuron section is the same as that in the previous row. However, if the previous measurement corresponds to either a “fork” or an “end point” the plugin assumes that the type of the current point is “dendrite”. It is always possible to change the type of section by hand, if the suggestion offered by the plugin is inappropriate.

The third, fourth and fifth numbers of the line are the x , y and z coordinates of the point, the sixth number is the radius (all in pixels). The last number is the index of the previous point: since this is the first point this index is set to -1. Notice that the Neuron_Morpho plugin assumes that in general the “previous point” entry is the index of the point in the previous row. The only exception is when the point in the previous row is an “end point” in which case the “previous point” entry is the index of the most recently measured fork point that has only one branch. If this is not the case then the “previous point” entry may be edited by hand by clicking on the table cell. Any label can be entered, but there is also a pull down menu that contains all the fork points measured thus far (see Figure 4).

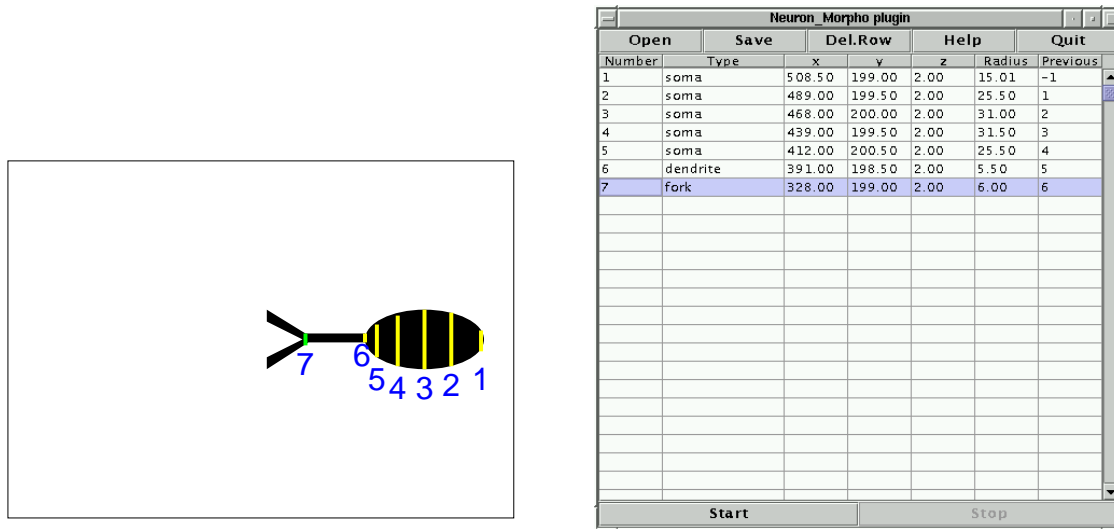


Figure 5: The first points selected. Note that at 7 the neuron forks and we must choose to follow one branch (in this case, the top one). The “Type” entry for point number 7 has been changed using the pull down menu. The green bars denote forks.

Number	Type	x	y	z	Radius	Previous
1	soma	508.50	199.00	2.00	15.01	-1
2	soma	489.00	199.50	2.00	25.50	1
3	soma	468.00	200.00	2.00	31.00	2
4	soma	439.00	199.50	2.00	31.50	3
5	soma	412.00	200.50	2.00	25.50	4
6	dendrite	391.00	198.50	2.00	5.50	5
7	fork	328.00	199.00	2.00	6.00	6
8	dendrite	265.50	163.00	1.00	8.01	7
9	fork	218.00	137.00	1.00	7.00	8
10	fork	157.50	106.00	1.00	7.02	9
11	fork	113.00	80.50	1.00	6.50	10
12	end point	65.00	55.50	1.00	6.50	11
13	end point	63.00	105.00	1.00	5.00	11

Figure 4: If the path indicated in Figure 1 is followed and the “fork” and “end points” are correctly indicated then the plugin will guess correctly the entries for the “previous point” columns. However, should these be incorrect we can edit the entry by hand or use the pull down menu. This contains the list of all the fork points measured thus far.

One should select points where the radius or the coordinates change appreciably (as in points 1 and 2 of Figure 5). The selected points on the middle slice are shown in Figure 5. At 7 the neuron branches and we have selected to follow the top branch. We follow this to the end (see Figure 6) and then get back to the bottom branch which we again follow sequentially to the end (see Figure 7).

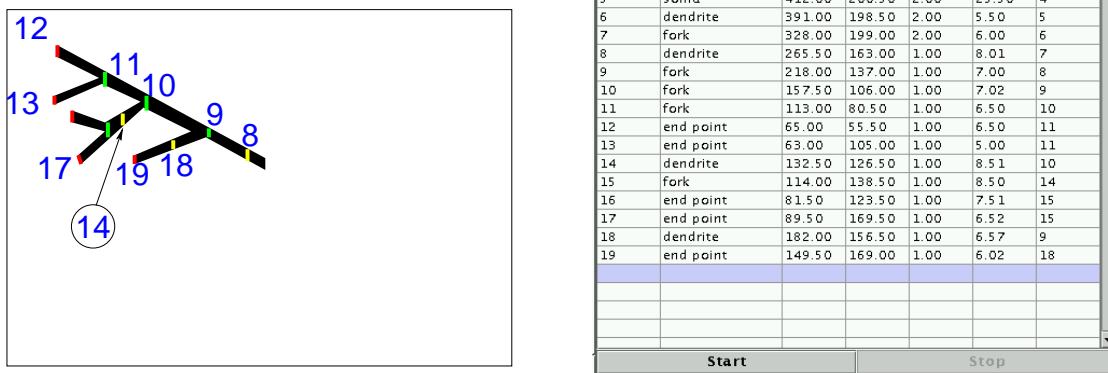


Figure 6: We follow the top branch completely before starting the bottom branch. The green bars denote forks, the red bars end points.

The measurement process can be interrupted at pleasure by pressing the stop button. It can be resumed by pressing the start button. Moreover, the data can be stored in a file at any stage of the measurement process (and it may be a good idea to save them regularly just in case the computer should melt under the strain). Should we wish to stop completely, we can save the data to a file, quit the plugin and close ImageJ. Next time we can continue the measurements from where we had stopped by loading the measurement file using the “Open” button.

5 cvapp and neuron

Once all the measurements have been made we can save them to a file, e.g. *morpho.swc* and read it using the program *cvapp*, written by Robert Cannon (see Figure 8). This program can be downloaded from <http://www.compneuro.org>. *Cvapp* allows the user to edit the neuron structure, to scale the x , y and z coordinates, to view the neuron in 3D using red and green filter and, finally, to write the morphological data in a hoc file that can then be read by the program *neuron*, written by Ted Carnevale and Michael Hines (<http://www.neuron.yale.edu/neuron/newhome.html>). The neuron as loaded by the program *neuron* is shown in Figure 9.

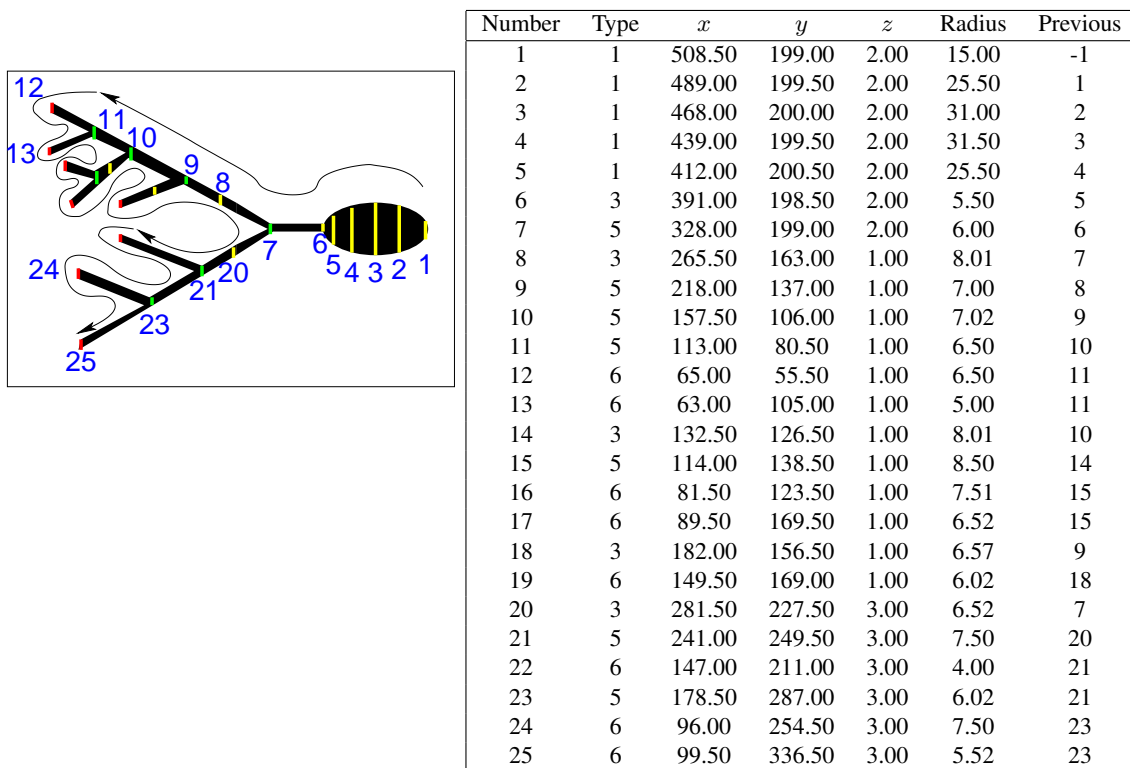


Figure 7: All the measurement points and their coordinates. After having followed the top branch to the end we come back to the bottom branch.

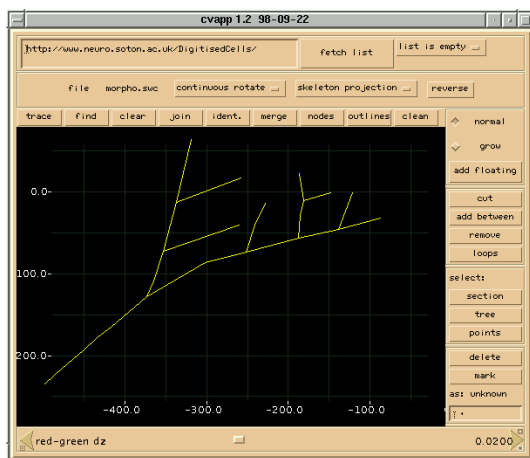


Figure 8: The swc file of the toy neuron used in this example has been loaded into cvapp.

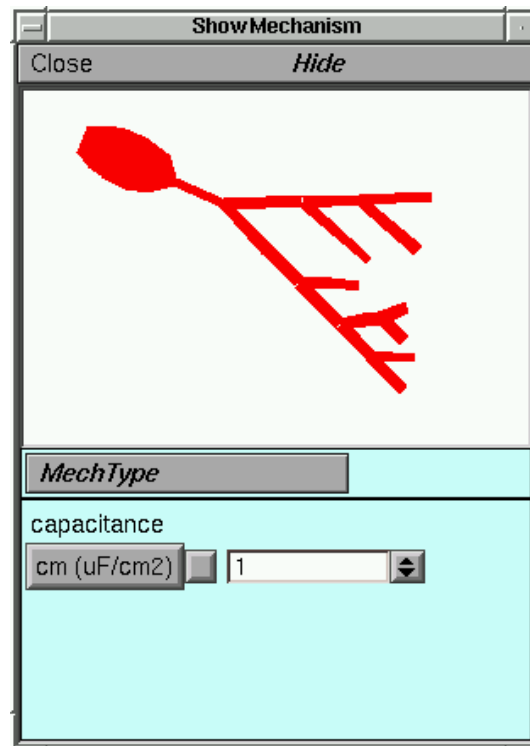


Figure 9: The hoc file produced by cvapp has been loaded by neuron. The figure shows the three dimensional reconstruction of the toy neuron.