

Comparison of Neuron Radius and Length Scaling Ratios from Angicart++ Image Reconstructions and Existing Morphological Reconstruction Data

Samhita Mallavarapu, UID: 505-205-013

CS Bio 199 Final Paper, Fall 2021

1 Introduction

Neurons serve as the messengers of information throughout the brain and the nervous system. From studies and examinations of the brain, we can see that these neurons are connected through an extensive network of axons and dendrites. Such networks work in conjunction with each other to relay information and drive an organism to respond to its surrounding stimulus. We can learn information about a neuron by taking a closer look at its structure and function. Then, we can compare these properties between networks of various neuron types to better comprehend the association between the structure and the function of a neural network.

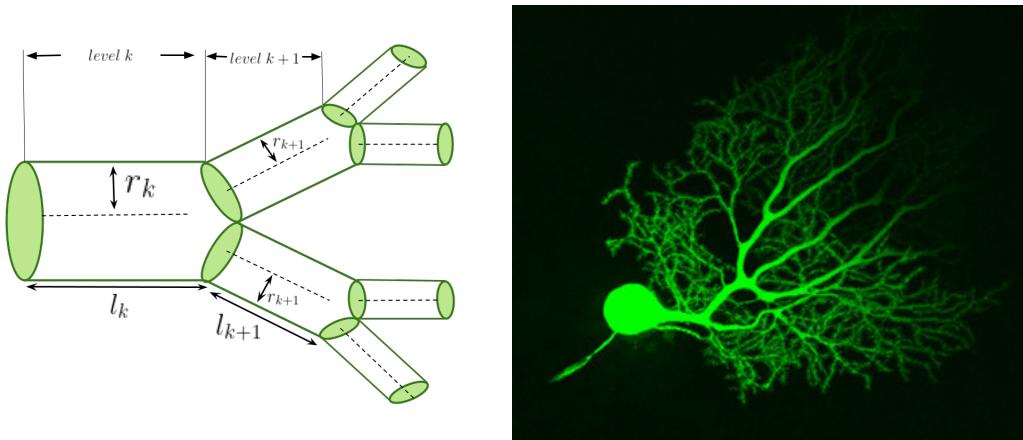
The best way to accomplish this mission is to build a model that takes into account the theoretical biophysical properties of the network, which can be done utilizing cost functions and considering constraints. We then take the functions and solve for the theoretical predictions for the radius scaling ratios which comes out to be $\frac{r_{k+1}}{r_k}$ and the length scaling ratio which comes out to be $\frac{l_{k+1}}{l_k}$. Then, we validate these models by comparing them to other models constructed of the neuron types.

To conduct this analysis, we use two different methods: the SWC method and the Angicart++ method. Both methods involve utilizing data that was downloaded from the Allen Brain Atlas. The SWC method involves taking the SWC files provided by the Allen Brain Atlas and conducting an analysis on these data sets. These SWC files are structured as pixel-by-pixel data so it is necessary to utilize the python script `swc-to-ratios.py` in order to organize the data by branch, as well as extracting the average radius and length of each branch.

The Angicart++ method takes the raw data images that can be downloaded from the Allen Brain Atlas and runs an analysis to extract the data from the images. Although the software was

primarily intended to extract information such as radius, length, and connectivity from images of blood vessels, our group noticed the structural similarities between blood vessels and branching neurons and decided to apply program to the neural network and work on validating the model for this purpose.

Once it is run, Angicart++ outputs the data in a manner that it is already organized by branch. We then loop through the outputs and calculate the scaling ratios utilizing the python script getratios.py. To get the radius scaling ratio, this code looks for the parent and daughter branches and takes the ratio of daughter radius to parent radius. Then, it does the same thing, this time focusing on obtaining the scaling ratio for the lengths of the branches. After this, we create histograms to show distributions of these ratios with the intention to compare the means in the data to the means in the theoretical prediction that were calculated from the SWC files. If the outputs of Angicart++ are relatively similar to the SWC calculations, it would validate the use of Angicart++ on neurons.



2 Data

The Allen Brain Atlas is a database that provides information about various neurons. It functions as a census of singular cells and their biological cell data. While the information within the Atlas is derived from both humans and mice, our team focused on the human cells that are found in the database. There is electrophysiological, morphological, and transcriptomic data for each of the cells available. According to the documentation that can be found within the database, the staining methods for the cells were programmed using the intelliPATH FLX® protocol run plan and were rinsed on an autostainer using 1X TBS wash buffer.

The Allen Brain Atlas database is comprised of image stacks that can be utilized for 3D analysis as well as standard SWC files. In order to test the validity of our Angicart++ software, we can compare the scaling ratio distributions between the histograms we make from the SWC files and

the histograms we make utilizing the Angiocart++ software.

We utilized the Cell Feature Search function of the Allen Brain Atlas to focus on the cells from the Middle Temporal Gyrus (MTG) layer 6. The MTG is the region of the brain that controls functions such as motion observation, deductive reasoning, language processing, and producing facial expressions [Desai-Chowdhry et al., 2021]. We found our primary subject, an eighteen year old male patient, by looking through the database and focusing on those cell images that displayed a clear branching pattern. For the SWC method of analysis, we simply downloaded the SWC files of this cell, extracted the radius and length ratios using the python code `swc-to-ratios.py`, and then created histograms out of these ratios from the output using the python code `histogram.py`.

Meanwhile, for the Angicart++ method, we started by downloading the image stacks from the Allen Brain Atlas and inverted the colors of the images using the python code `invert.py`. Then, we found that each image in the stack was 7580×11264 pixels however this image size is incompatible with Angicart++. This led us to downsampling the images until the dimensions were 470×701 pixels which took four cycles of running the downsampling code. We also normalized the data by removing unnecessary images, giving us a more accurate set of pictures to run the Angicart++ analysis on.

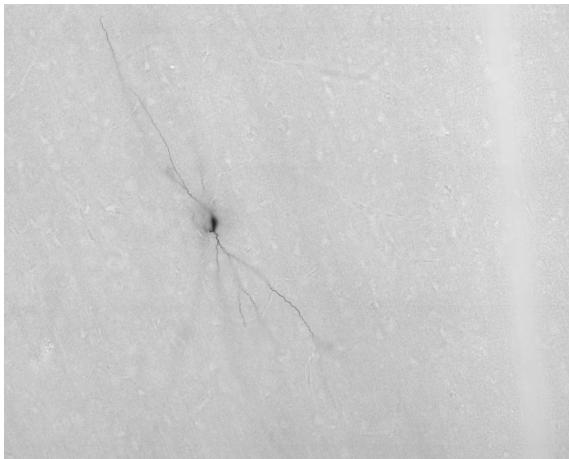


Figure 2: This is a picture of a raw image from Allen Brain Atlas, before any of the pre-processing steps we took to get it ready for Angicart++.

3 Methods: Existing Reconstruction Data Analysis

The SWC files that are downloaded from the Allen Brain Atlas that are in the format of pixel-by-pixel data. Because of this, it becomes necessary to use the python script `swc-to-ratios.py` to extract the average radius and length of each branch and organize this data by branch. The script `swc-to-ratios.py` works by first counting the number of pixels in the file, after which it finds the ID of the pixel in which the branching occurs. Then, the script identifies the IDs of the pixels of

the parent and child branches which it uses to separate the branches. From here, the script breaks up the parent branches into levels and creates an output file that specifies the pixel ID labels for each point, the x,y,z spatial coordinates, the radius at each of the points, and parent pixel IDs. It also takes these values, and calculates the radius and length of each of the branches using the equations below.

The following formula is used to average the radius values in each branch. In this equation, the k represents each branch, where the pixels i range from 1 to N_k. The N_k is the last pixel of each branch.

$$r_k = \frac{1}{N_k} \sum_{i=1}^{N_k} r_i$$

The following formula is used to extract the length of each of the branches. This is done by extracting the Euclidean distances between each point within the branch and using the formula. The Euclidean distance is the distance between points, in this case the pixels, being measured.

$$l_k = \sqrt{\sum_{i=1}^{N_k} (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}$$

Once the radius and length of each of the branches are calculated, the values are used to compute the scaling ratios. Obtaining the radius scaling ratios is done by dividing the daughter radius by the parent radius. These radius scaling ratios are then listed and outputted into a .dat file. The same process is repeated but this time with the intention of obtaining the length scaling ratios of the branches. We then take the radius scaling ratios .dat file and use the python script histogram.py to create a histogram from the values. We do the same for the length scaling ratios .dat file. These histograms contain information such as the mean and the Standard Error of the Mean (SEM), which are both listed and displayed by solid lines on the plots. These histograms were compared with the Angicart++ outputs that will be discussed in the next section.

We also wanted to create a more complete distributions that take into account variations within samples and we did this by downloading the SWC files of two more samples of the same neuron type that also had clear images. We ran the same python script swc-to-ratios.py on these files individually and used the python script bind.py to combine the radius ratios list .dat files of all three neuron samples and ran the python script histogram.py to create a histogram from the combined values. We then followed the same process for the length ratios list .dat files.

4 Methods: Image Processing with Angicart++ and Analysis

As mentioned previously, the first step in utilizing the Angicart++ method to conduct the analysis was obtaining the data on which to conduct the analysis on. For this, we downloaded the image stacks from the Allen Brain Atlas which were in .jpg files. Next, we numbered the images to make them ready for the python script invert.py. This script called for the python program OpenCV, which led to us importing the package cv2, as this is an import name for OpenCV. OpenCV is utilized for computer vision, machine learning, and image processing, and is an open-source library full of algorithms to accomplish these tasks.

We used the invert.py script to invert each of the downloaded .jpg image files and convert them into .png files. We also found that the image files were 7580×11264 pixels. To use Angicart++ most efficiently, both of the dimensions should be less than 1000, motivating us to use the downsampling script 4 times to make our images this size. We then removed the images that did not appear to contain valuable information before running the images through Angicart++ at various threshold values. Running the Angicart++ program at multiple threshold values allowed us to conduct a threshold analysis that will be discussed later.

Angicart++ works by taking the image stacks and analyzing them as a 3D image. Then, the software classifies the voxels in the 3D image as part of the vessel, or in our case the neuron, based on an intensity threshold. Next, we manually find a threshold value by trial-and-error method relative to the visual of the vessels. This results in a network mask upon which we use spatial criteria to find the endpoints of the network. Then, we find the centerline and branching points of the network. Then, the software removes the voxels that can be removed without disrupting the network. From here, the measurements of the network are quantified. [Newberry et al, 2015]

Unlike the output of the SWC file analysis, the output of Angicart++ is organized by branch and is structured in two .tsv files per chosen threshold. One of the output files is with roots while the other output file does not contain roots. We focused on the file that presented the roots. Because we were conducting a threshold analysis to determine which would be the best for our comparison, we ran the analysis for 17 different thresholds that fell within the range from 0.255 to 0.575.

Once we got the .tsv files, we extracted four of the data columns and put them into their own .dat files, one column per file. The columns extracted were the parent names, the vessel names, the vessel radius (we chose to look at the observed radius), and the vessel length. Then, we use the python script replacena.py to replace any "N/A"s in the parentname.dat file with a 0 in order to normalize the data so that the next steps of the analysis could be done.

Next, we took the data and looped through its outputs in order to calculate the scaling ratios.

This was done utilizing the python code `getratios.py`. This script imported the four .dat files and used them to look for the parent and daughter branches. Then, it took the ratio of the daughter radius to the parent radius to get the radius scaling ratio distribution. Following this, it took the ratio of the daughter length to parent length to get the length scaling ratio distribution and printed both of these lists as separate .dat files.

Once the scaling ratios had been found, the same `getratios.py` script created histogram distributions for each of the thresholds, meaning that this process occurred 17 times during this threshold analysis. Once the 34 histograms were made, we wrote out a table that included a column for the threshold value, the mean radius, and the mean length. These were then turned into 3 .dat files that were then imported into the python script `threshold.py`. This was used to make the threshold analysis plots from which we picked the threshold that gave us the histograms to which we could best compare to the SWC histograms. If the outputs of Angicart++ turned out to be relatively similar to the SWC calculations, it would validate the use of Angicart++ on neurons.

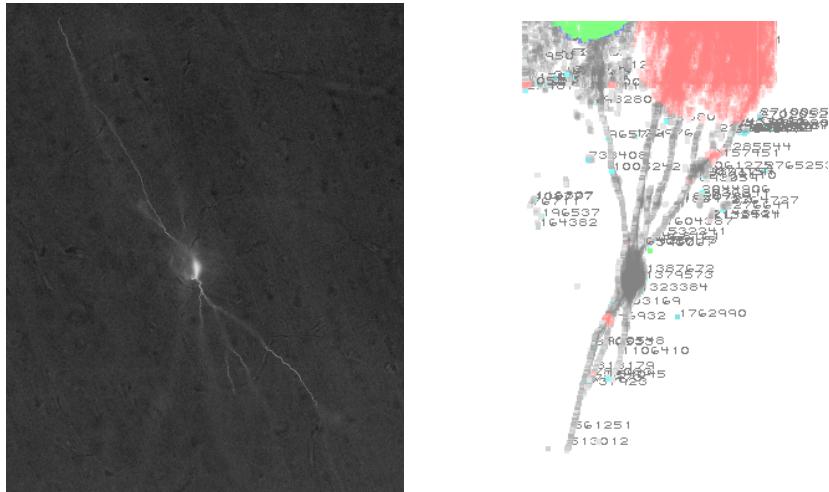


Figure 3: The left picture is the inverted picture of a raw image from Allen Brain Atlas. The right picture is an image of the Angicart++ software as it is reconstructing the neuron from the image stack.

5 Results

We had four sets of results that came from the work on this project. The first set of results is from the combined radius and length scaling ratio analysis. The second set of results is from the threshold analysis. The third set of results is from the comparison of the radius between the SWC files and the Angicart++ data, while the fourth set of results is from the comparison of the length between the SWC files and the Angicart++ data. The results of these projects will be discussed in the following subsections.

5.1 Combined Radius and Length Scaling Ratios

To get more data points to make a more representative distribution with a smaller standard of error, we downloaded the SWC files of three neurons and combined their data to create a histogram of their combined radius and a histogram of their combined branch length. This combination of 3 data sets allowed for more continuous histogram distributions rather than the gaps in the data that we were seeing from the histogram distributions that were constructed from the single SWC file.

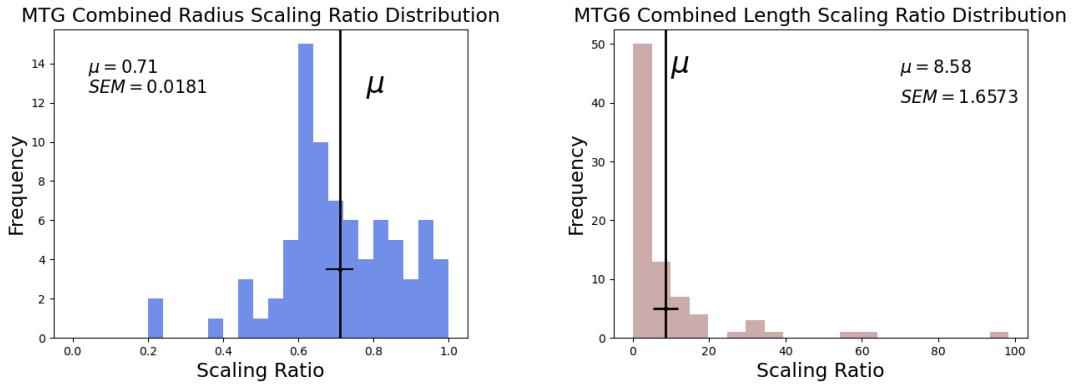


Figure 4: The left histogram is the MTG Combined Radius Scaling Ratio distribution. The right histogram is the MTG Combined Length Scaling Ratio distribution.

As we can see, the distribution for the radius scaling ratio is uni-modal and slightly right skewed. The mean radius scaling ratio is 0.71 and there is an error of 0.0181 with few outliers. The range of the radius scaling ratio also falls between 0.2 and 1.0. Meanwhile the distribution of the length scaling ratio is uni-modal and noticeably right skewed with a mean length scaling ratio of 8.58 and an error of 1.6573. The range of the length scaling ratio is much wider and falls between 0 to 100.

5.2 Threshold Analysis

Before we could choose which Angicart++ output to use in our comparisons, we had to conduct a threshold analysis. To do this successfully, we took 17 different thresholds that fell within the range from 0.255 to 0.575. We then plotted the results into the plots shown below.

We choose the threshold where the data plateaus from these plots. In this case, we choose the threshold of 0.260 based on these plots as there is a clear plateau at this threshold value in the radius plot and a slight plateau at this threshold value in the length plot.

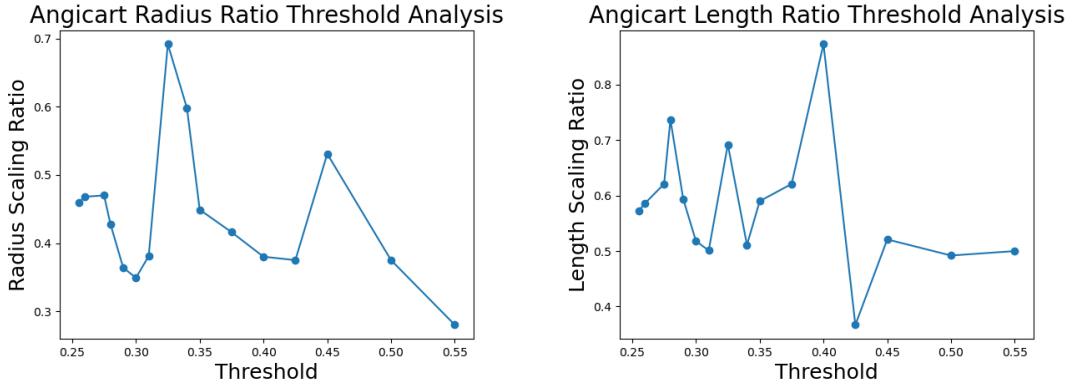


Figure 5: The left histogram is the Angicart++ Radius Ratio Threshold Analysis. The right histogram is the Angicart++ Length Ratio Threshold Analysis.

5.3 Comparison of the Radius Scaling Ratios Between the SWC Files and Angicart++ Data

As we can see, the distribution for the radius scaling ratio from the SWC files is uni-modal and left skewed. The mean radius scaling ratio is 0.66 and there is an error of 0.0525 with few outliers. The range of the radius scaling ratio also falls between 0.2 and 1.0. There are also many gaps in the distribution of the data. This all leads to a rather long error bar (the horizontal line). Meanwhile the radius scaling ratio from Angicart++ is uni-modal and extremely right skewed. The mean radius scaling ratio is 0.47. While we did not have an exact calculation of the error, we notice that the distribution is rather continuous over its range between 0.0 and 1.0. This leads to a much smaller error bar for the Angicart++ Radius Scaling Ratio distribution.

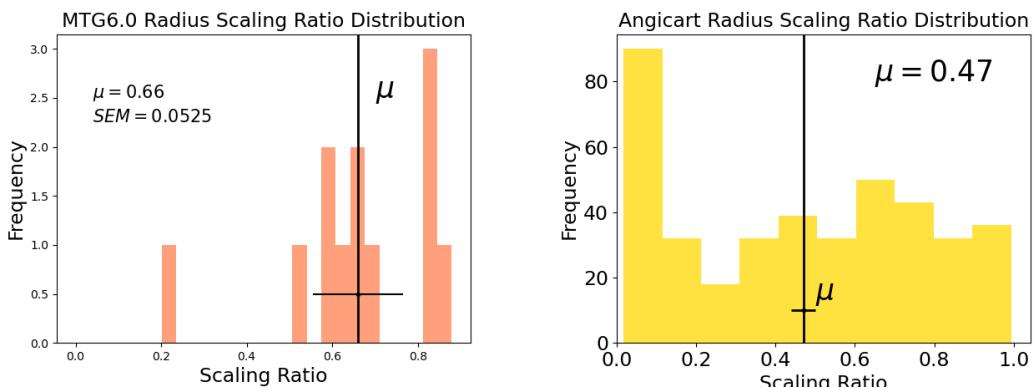


Figure 6: The left histogram is the SWC Radius Ratio Scaling Distribution. The right histogram is the Angicart++ Radius Ratio Distribution.

5.4 Comparison of the Length Scaling Ratios Between the SWC Files and Angicart++ Data

As we can see, the distribution for the length scaling ratio from the SWC files is uni-modal and left skewed. The mean length scaling ratio is 16.49 and there is an error of 6.2694 with outliers. The range of the length scaling ratio also falls between 0 and 60. There are also many gaps in the distribution of the data. This all leads to a rather long error bar (the horizontal line). Meanwhile the length scaling ratio from Angicart++ is uni-modal and left skewed. The mean radius scaling ratio is 0.59. While we did not have an exact calculation of the error, we notice that the distribution is rather continuous over its range between 0.0 and 1.0. This leads to a much smaller error bar for the Angicart++ Length Scaling Ratio distribution.

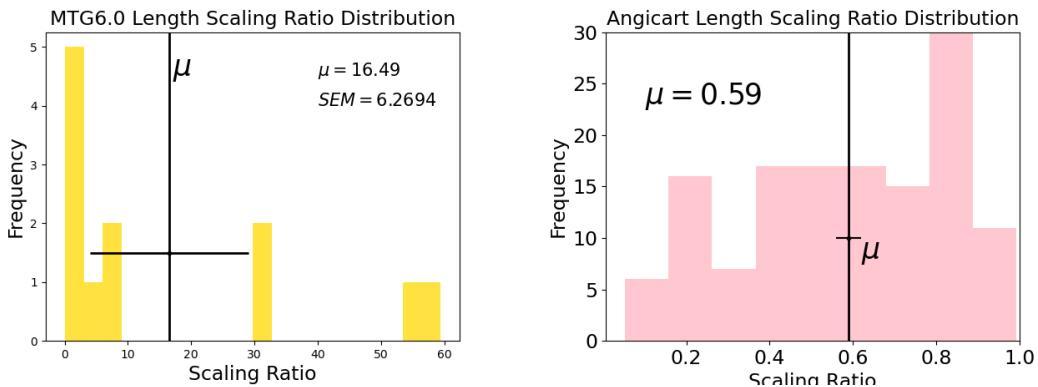


Figure 7: The left histogram is the SWC Length Ratio Scaling Distribution. The right histogram is the Angicart++ Length Ratio Distribution.

6 Conclusion

We looked at the combination of three SWC files as opposed to the singular SWC file because we wanted to see a more continuous and representative distribution with a smaller error. If we compare the combined histograms to the singular SWC histograms, we see that the error bars are much smaller in both the radius and length combined graphs than the singular file graphs, giving us a higher accuracy with these combinations.

When looking at the combined histograms, the radius scaling ratio distribution is uni-modal and slightly skewed to the right with only a few outliers. Meanwhile the distribution of the length scaling ratio is uni-modal and noticeably right skewed with many outliers. This could be occurring because the variance within the length is much larger than that of the radius. Additionally, as more branching occurs, the radius of the daughter branches will be smaller than the parent branch. This is not necessarily the same when looking at the length of the branch.

Next, we had to do a threshold analysis to pick a threshold. To achieve this, we decided to

conduct an analysis using 17 different threshold values and plotting their radius onto one graph and their length onto another graph in order to see where the outputs would plateau on both graphs. After examining the plots, we determined that of the threshold values we chose, 0.26 was the optimal value to conduct our comparisons with.

When we take a look at the comparison of the radius scaling ratios between the SWC Files and Angicart++ Data, we see that there appear to be significant differences between the two histogram distributions such as gaps within the SWC file distribution. This distribution is also left skewed while the more continuous Angicart++ distribution is right skewed. The means are also different, with a higher mean for the SWC file. The error bars also vary in length, with a longer one for the SWC file distribution. These discrepancies could be due to the fact that the SWC files have a smaller number of data points to derive their ratios from while Angicart++ takes a higher number of data points. Additionally, it could be because the bulk of the data of the Angicart++ distribution is collected from the earlier images because we removed some of the data of the later images when we determined that the margins were not clear.

When we take a look at the comparison of the length scaling ratios between the SWC Files and Angicart++ Data, we see that there appear to be significant differences between the two histogram distributions. There appear to be gaps within the SWC file distribution. This distribution is also right skewed while the more continuous Angicart++ distribution is left skewed. The means are also different but this is because the scaling ratios appear to be on a different scale altogether. Additionally, the SWC file distribution has a longer error bar than the Angicart++ distribution. Again, these discrepancies could be due to the fact that the SWC files have a smaller number of data points to derive their ratios from while Angicart++ takes a higher number of data points because of where the bulk of the data of the Angicart++ distribution is collected from.

7 Future Directions

We only took a look at a singular neuron from the Allen Brian Atlas. We should take a wider sample if we want to get a better understanding of how similar the outputs of the SWC file method are to the outputs of the Angicart++. I plan to take other samples from the Allen Brain Atlas and conduct a similar analysis to the one described above. This will help me to validate the Angicart++ outputs, giving us a tool to better understand the neuron as a whole, which would put forth the opportunity to ask and get answers to other questions. One main question that our group hopes to answer is to understand why neurons with certain functions have the structure that they have. In other words, why do different neuron types have a different branching network? This validation of Angicart++ and its outputs would be a first steps in answering that question.

References

- [1] Allen Institute for Brain Science. Allen Human Brain Atlas. 2010 [cited 2021 December 9]. Available from: <http://celltypes.brain-map.org/experiment/morphology/613396614>.
- [2] Desai-Chowdhry P, Brummer A, Savage V. How Axon and Dendrite Branching Are Governed by Time, Energy, and Spatial Constraints . bioRxiv 2021.07.15.452445; doi: <https://doi.org/10.1101/2021.07.15.452445>.
- [3] Newberry MG, Ennis DB, Savage S, Cox VM. Testing Foundations of Biological Scaling Theory Using Automated Measurements of Vascular Networks. PLoS Computational Biology. 2015; 11(8): e1004455.
- [4] OpenCV Documentation. OpenCV Open Source Computer Vision. [cited 2021 December 9]. Available from: <https://docs.opencv.org/4.x/d1/dfb/intro.html>.
- [5] Savage VM, Deeds EJ, Fontana W. Sizing up Allometric Scaling Theory. PLoS Computational Biology. 2008; 4(9):e1000171.