Mapping the Brain: An Introduction to Connectomics Progress Report: Streamlining Membrane Segmentation Pipeline

Thomas Keady, Albert Lee, Augusto Ramirez

January 16, 2016

1 Summary

In the past week, the neuroXplorer project to streamline the MACHO pipeline has been updated to instead streamline the membrane segmentation pipeline. Currently, the project is running a few days behind the expected deadlines presented in the proposal. The latter is mostly due to technical difficulties with running the original pipeline and collecting data (at the beginning). We resolved this after meeting with Jordan, who explained ndio (which now allows us to retrieve membrane ROI (regions of interest) annotation and image data from the Open Connectome Project (OCP) server. Will guided us through running the pipeline by using the CAJAL application programming interface in Matlab. Also, we recently found out that we should be collecting data from the cv_kasthuri11_membrane_2014 token instead of kasthuri2015_ramon_v1. This new change has allowed us to make progress in the python implementation for running watershed but has also presented some issues (ndio is missing the RAMONVolume functions or a simple OCPQuery module). We are working on watershed segmentation using scikit-image and managed to run it through python on the collected data successfully. The progress with watershed has allowed us to move on with correcting issues (image processing output) presented in the new batch of data (cv_kasthuri11_membrane_2014).

2 Updated Goals

As mentioned earlier, the project has evolved from streamlining MACHO, with a vague sense of impact to quantifiable results, to streamlining the membrane segmentation pipeline's image processing steps. Working with watershed we now have a clearer project with a more defined goal - to enhance parameters to lower splitting and merging error rates. These new results from our streamlining experiment using watershed and python will lead to a documented optimal way to run the membrane segmentation pipeline. Our results can be passed to gala next, making the project potentially relevant to streamlining the i2g pipeline.

2.1 Updated Timeline

- 1/16/16: Get guidance from Will/Greg and/or Jordan on running OCPQuery (imageDense and probDense) from CAJAL on ndio.

- 1/17/16: Compare and contrast at least two sets of images through our pipeline and the current pipeline and most importantly, understand the results.

- 1/18/16: Have Matlab segments of the pipeline converted to working and integrated Python.

- 1/20/16: Begin implementing Kaynig gap completion code in Python as another step in the pre-pipeline processing.

- 1/22/16: Be able to run full pipeline plus our addition(s).

- End of class: Further reduce inaccuracies.