IDENTIFYING WHITE MATTER FIBER CROSSING AREAS IN THE MOUSE BRAIN USING A FLASK BASED WEB INTERFACE AND THE ALLEN SDK

Philippe Lemieux, Joël Lefebvre

Computer sciences department
Université du Québec à Montréal, QC, Canada

ABSTRACT

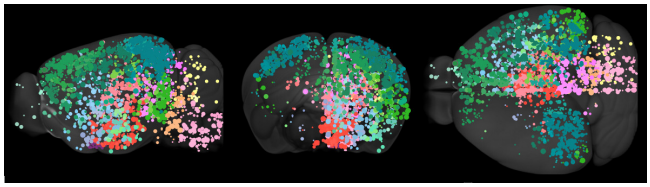
The Allen Mouse Brain Connectivity Atlas offers data for around 3,000 experiments where a mouse brain is injected with a fluorescent virus which can be followed throughout the mouse neurons. Using the Allen software development kit (SDK), we created a web interface for studying the topology of fibers crossing in mouse brains. This application lets users browse and query experiment data from the Mouse Connectivity Project, as well as finding potential structures for fiber crossing. Making Allen SDK data more accessible with this interface could facilitate the acquisition of such data for our research in mouse brain fibers crossing.

***Index Terms*—** Neuroinformatics;

1. INTRODUCTION

"The Allen Mouse Brain Connectivity Atlas is a three-dimensional, high-resolution map of neural connections in the mouse brain. Axonal projections from a broad range of brain regions were mapped on a standardized platform to generate a comprehensive database of neural projections"[1], all of which can be browsed and queried using some of the many modules of the Allen software development kit (SDK). Exploiting the neuronal projection paths toward the injection site, we could identify potential fibers crossing areas. Let's say, we perform an experiment where one injection is in the left hemisphere and another is in the right hemisphere. Using the density of signal in the target brain structure as a ratio of pixels with signal over all pixels in the structure, if the density for one structure is high for both experiments, the odds of seeing fibers crossing by overlapping both injections are high. Identifying these regions of interests (ROI) would be more efficient than trial and error. Also, the mouse connectivity project injections cover most of the right side of the mouse brain, which could hopefully be mirrored, reducing the need for further experiments for the acquisition of data (See Fig. 1). This would decrease the number of experiments needed on mouse specimens.

In this paper, we explore the use of the Allen SDK with our Web interface to visualize and download the mouse connectivity experiments and find ROI to investigate for observing fibers crossings. The developed application also serves as a bridge to distribute some of the AllenSDK data, such as volumes and streamlines, to further investigate the detected ROI.



horizontal, coronal, sagital view of mouse brain

Fig. 1. All the experiments injections coordinates in the mouse average brain from the Allen Mouse Connectivity Project[2]

2. METHODOLOGY

2.1. Application

Since the Allen SDK is in Python, we used the framework Flask[3] to develop the web interface. This way, we can easily import the SDK modules. In particular, we have used the following modules:

- `allensdk.core.mouse_connectivity_cache` (MCC): for displaying the experiments based on client filters. Filters are user sided, all in Javascript.
- `allensdk.api.queries`: for giving to the user the option to download brain slices or volume for an experiment.
- `allensdk.api.queries.mouse_connectivity_api` (MCA): a little bonus which lets the user search by correlation, injection coordinate, or source.

Flask will generate our templates which will be serve to client as HTML pages. In particular, our page for displaying all the experiments in a table. Each row is an experiment and

This work was funded by a QBIN pilot project grant (35450)

the columns are: experiment id, injection structures, product, injection volume, injection coordinate (x, y, z), transgenic line, specimen name, gender and genetically modified (cre). Rows can be filtered by column values with include and exclude filters.

2.2. Query volumes and streamlines

Allen SDK serves volumes and tractograms in a different way than ours with nrrd files. From reading the header of one of the served template nrrd file, the allen brain format is in the Left-Posterior-Superior space. Preferring nifti files for such information, we change the nrrd files to nifti with the Python library NiBabel[4]. For this, we create an affine matrix to align this brain to the Right-Anterior-Superior space and in mm. Since we don't host the nrrd files, we need to download each of the nrrd volumes on the application host computer before creating a nifti file from them.

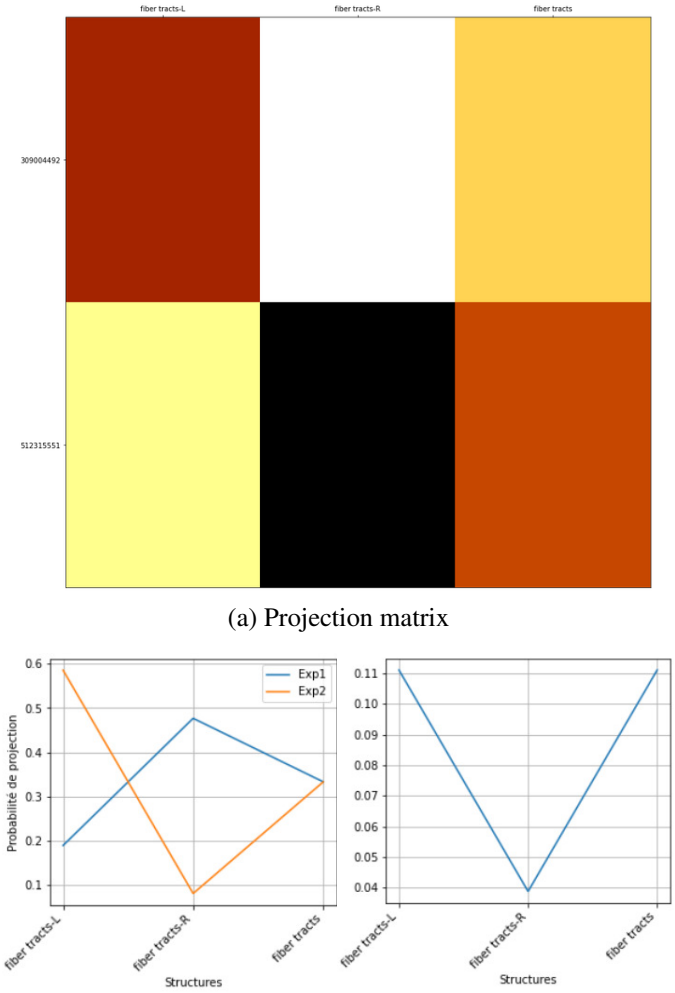
2.3. Average projection density volume

We offer to the client to get the average projection density volume to download by summing the average of each filtered experiments projection density volume into one. Creating this volume requires the download of each experiment volume on the application host for processing. For visualizing this volume, we also serve the average template volume of the mouse brain.

2.4. Detecting fiber crossing ROIs

As an example, using the experiments 309004492, 512315551 and the fiber tracts structures. Using the module MCC, we get the projection matrix with each row being an experiment and each column a structure (Fig.2.a). Each case indicates the fluorescent projection density in a given brain structure for an experiment. We normalise each column in a row with the sum of the density for this row to get a density probability (Fig. 2.b). From there, multiplying the density probability for a structure of an experiment with another one gives us our probability of seeing fibers crossing (Fig. 2.c).

From this result, we can identify the structures "fiber tracts" and its left part as the ROI.



(a) Projection matrix

(b) Normalised densities

(c) Probabilities multiplied

Fig. 2. Density probability graphs

3. RESULTS AND DISCUSSION

3.1. Query experiment data

We can offer the user to download a CSV file with the filtered experiments. Same goes for serving streamlines. Downloading slices and experiment volume can be slow if the resolution is high, like 10 microns but we will get the nifti file eventually. We don't truly host these volume files. When needed, the application host has to download them temporarily from the Allen Brain website. Operations such as creating an average mouse brain volume based on filtered experiments and getting streamlines requires to download of the nrrd volumes to compute. This can take a while and ends up taking a lot of disk space. Caching some of the files and avoiding to demand too much memory from the application host would help, but we haven't reached that point yet.

3.2. Searching for ROI

This search mode doesn't suffer from the query issue described above and may well be used to facilitate the search for finding instances of fibers crossing. One issue could be when we used with too many experiments in our search. Let's say we observe a high density for all of our experiments except one in our projection matrix, and that its density is 0, then the search will indicate no possibility of fiber crossing in this structure since all the experiment rows in the projection

matrix are multiplied.

4. CONCLUSION

We made an Web application to make the Allen SDK data for mouse connectivity more accessible. We can download nifti volumes, average projection density volume, streamlines, uses some of the search functions offered by the MCA and identify ROI for some experiments in which fibers crossing could be observed. While identifying ROI using the projection matrix worked fine, the average volume and streamlines functionalities are too slow at the moment to be used with many experiments. Some important topics that could be addressed in future works are: caching experiments volume and deploying the application. This application has already been used for identifying experiment in a conference submitted to Photonics West 2021 and we intend to use it in research about fibers crossing.

5. REFERENCES

[1] Allen Brain Atlas Team, "Overview :: Allen brain atlas: Mouse connectivity," .

[2] Allen Brain Atlas Team, "Projection :: Allen brain atlas: Mouse connectivity," .

[3] Flask Team, "Flask — the pallets projects," .

[4] Michael Hanke (MH) Marc-Alexandre Côté (MC) Ben Cipollini (BC) Paul McCarthy (PM) Chris Cheng (CC) Yaroslav Halchenko (YOH) Satra Ghosh (SG) Eric Larson (EL) Demian Wassermann Stephan Gerhard Matthew Brett (MB), Chris Markiewicz (CM) and Ross Markello (RM), "Neuroimaging in python - nibabel 3.1.1 documentation," .