



HAL
open science

Exploring the Allen mouse connectivity experiments with new neuroinformatic tools for neurophotronics, diffusion MRI and tractography applications

Mahdi Abou Hamdan, Elise Cosenza, Sylvain Miraux, Laurent Petit, Joël
Lefebvre

► **To cite this version:**

Mahdi Abou Hamdan, Elise Cosenza, Sylvain Miraux, Laurent Petit, Joël Lefebvre. Exploring the Allen mouse connectivity experiments with new neuroinformatic tools for neurophotronics, diffusion MRI and tractography applications. SPIE BiOS 2023 - Neural Imaging and Sensing 2023, Jan 2023, San Francisco, United States. pp.123650A, 10.1117/12.2649029 . hal-04232388

HAL Id: hal-04232388

<https://hal.science/hal-04232388>

Submitted on 8 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Exploring the Allen Mouse Connectivity experiments with new neuroinformatic tools for neurophotonics, diffusion MRI and tractography applications

Mahdi Abou-Hamdan^{a,b,c}, Elise Cosenza^b, Sylvain Miraux^c, Laurent Petit^b, and Joël Lefebvre^d

^aPolytech Nancy, Université de Lorraine, Nancy, France

^bGIN-IMN, UMR5293 CNRS, Université de Bordeaux, Bordeaux, France

^cCRMSB, UMR5536 CNRS, Université de Bordeaux, Bordeaux, France

^dDépartement d'informatique, Université du Québec à Montréal, Montréal, Canada

ABSTRACT

The Allen Mouse Brain Connectivity Atlas (AMBCA) offers a high-resolution map of neural connections detailing axonal projections labeled by viral tracers. It is a unique tool for studying structural connectivity and better understanding the white matter pathways of the gene mouse brain. But, the analysis and comparison of these data are limited to a simple visualization on the Allen website and have no direct relationship with specific User data. Here, we propose a series of python-based tools to operate with AMBCA data in the User's data space. Our method is based on "back and forth" actions between Allen and User data using the Allen Software Development Kit (AllenSDK) to import data from the Allen Institute and the Python package ANTsPyX for registration. A transformation matrix is calculated with ANTsPyX to overlay, for instance, Allen's projection density maps with a diffusion MRI-based tractography in the User space. Conversely, applying the inverse transformation to a specific location along a white matter bundle within the User space allows us to recover which experiments were done at this particular location in the Allen Mouse brain Common Coordinate Framework (CCFv3). Thus, both data can be used in a natural interaction, e.g., by inspecting them in a visualization tool such as the MI-Brain software. This series of tools will offer an attractive solution for researchers with neural tracing and/or tractography data to be combined with the AMBCA. The code is available at: <https://github.com/linum-uqam/m2m>.

Keywords: Mouse Brain, Multimodal Registration, Tractography, Visualization, Multimodal Imaging, Neurophotonics

Further author information: lefebvre.joel@uqam.ca

1. INTRODUCTION

Knowledge about the neural connections among brain regions is critical for advancing our understanding of normal brain function and changes that occur with aging and disease. Researchers use various experimental techniques in the innovative connectomics field to map connections at different levels of granularity in rodent animal models. However, comparing and integrating multimodal connectomics results still is a challenging task. In their recent review, Leergaard and Bjaalie underlined that current brain atlases offer new opportunities for extensive data integration to improve our understanding of the organization of the brain.¹ Serial two-photon tomography has been a fundamental methodological innovation for tract-tracing at the microscopic scale.² Using the blockface acquisition of high-resolution microscopic images, the Allen Institute has successfully used this technology to generate large volumes of microscopic, fluorescent viral tracers 3D tract data showing brainwide connections in the mouse brain.^{3,4} In parallel, diffusion MRI (dMRI) and tractography can noninvasively image white matter fiber pathways across the

brain at the macroscopic level.⁵ To date, few approaches can bring together these two fields of connectomics. Here we provide a series of turn-key tools to combine these low-resolution macroscopic data with high-resolution microscopy data from more invasive studies (e.g., serial blockface histology, light-sheet microscopy, electrophysiology ...). Our goal was to create tools that allowed easy interaction between the user data and the existing Allen Institute datasets (see Fig. 1). To be able to work with both data seamlessly, multiple challenges had to be addressed, such as multimodal and multiresolution data registration, the bidirectional transformation between datasets (e.g., from the Allen to the user spaces and vice versa), and the integration with existing neuroinformatics toolkits used to analyze dMRI brain data. Combining macroscopic dMRI data with serial blockface histology (SBH)⁶ data would enable new neuroimaging experimental paradigms. It would allow, for example, to compare dMRI-derived white matter fiber bundles with cellular-level data originating from viral tracer injections to validate the tissue microstructure hypotheses often used to interpret dMRI data. The tools can also be used with serial blockface histology (SBH) to either plan the SBH experiments to target relevant regions of interest (ROIs) or to augment SBH during the acquisition with useful data helping the microscope operators to locate fiber bundles, brain regions, and other ROIs.

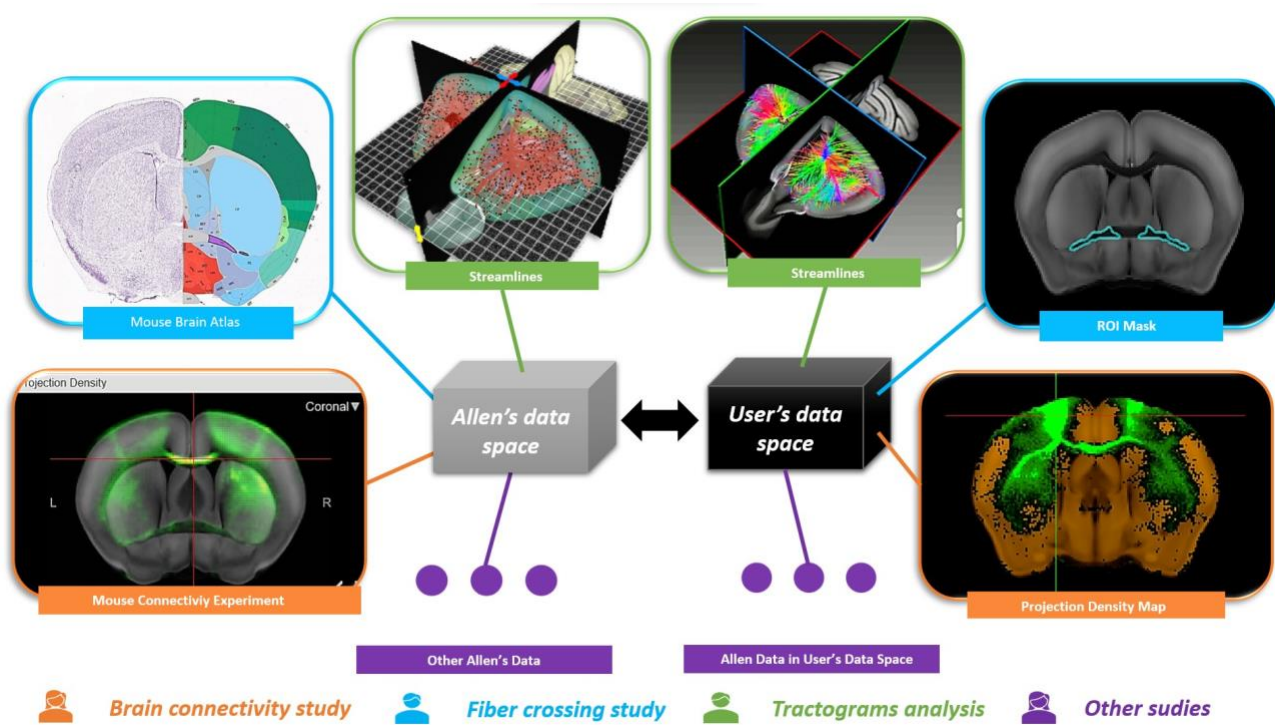


Figure 1. "Back and forth" interactions between the Allen Data Space (ADS) and the User's Data Space (UDS). Each pair of thumbnails represent an example of data provided by the Allen Institute (left side) and its counterpart in the UDS (right side). Each color represents an example of the data needed for a specific study.

We here introduce our **m2m** toolkit. We first describe the various multimodal and multiresolution mouse brain data used for this project. We used, in particular, the Allen Mouse Brain Connectivity Atlas⁴ and its associated datasets. We start by summarizing the Allen data acquisition and preprocessing necessary to prepare the data for the multimodal registration and transformation between Allen (ADS) and user data (UDS) spaces. The methods section also describes the data interaction and visualization tools used to allow neuroscientists to use their mouse brain data with the Allen Institute data, followed by some simple use cases of our toolkit. It includes downloading viral tracer projection maps, detecting fiber-crossing regions and brain structures, and creating custom white matter bundles combining dMRI-derived streamlines with microscope-derived brain structure ROIs.

2. METHODS

2.1 Data sources

We used the Allen Mouse Brain Connectivity Atlas (AMBCA), offering high-resolution maps of neural connections to describe axonal projections labeled by fluorescent viral tracer injections.⁴ The whole infected mouse brains have been imaged with serial 2-photon microscopy to generate 3D (viral) projection density maps. We also used the 3D spatial high-resolution mouse brain average template available in the AMBCA, named AVGT template. It consists of an average of 1,675 young adult mouse brains of the tissue autofluorescence measured with serial 2-photon tomography. The AMBCA is a unique tool for studying structural connectivity and better understanding the white matter pathways of the mouse brain at the mesoscopic scale. Fluorescent-derived streamlines can also be downloaded from the AMBCA. They consist of the shortest 3D path within the projection map that links a voxel to its injection volume. They differ from dMRI-derived streamlines in that the fluorescent-derived streamlines were generated by attempting to maximize the fluorescence along its path and without considering local orientation distribution functions (ODF). The fluorescent-derived streamlines dataset was made available through the Neuroinformatics-NL group*.

The diffusion MRI data used in the current study were acquired *ex vivo* in 20 C57BL/6J mouse brains on a 7T small-animal Biospec MRI system (Bruker BioSpin, Germany) with the Paravision 6.1 software. All experiments were approved by the local ethical committee (Animal Care and Use Institutional Ethics Committee of Bordeaux, #13152). Diffusion-weighted imaging (DWI) was accomplished using a 3D diffusion-weighted SpinEcho EPI sequence with a repetition time (TR) = 400 ms, 12 numbers of EPI shot, and echo time (TE) = 23.44 ms, and b value = 3000 s/mm². The acquisition matrix was 192 x 128 x 128 over a 20 x 13 x 13 mm field of view, resulting in 104 x 102 x 102 μ m voxels. The diffusion protocol included 60 diffusion directions and 5 non-diffusion-weighted (b0) measurements for a total scan duration of 11h for each mouse brain. The DWI processing consists of a series of tasks from raw DWI preprocessing steps to the diffusion tensor imaging (DTI) metrics and fiber orientation distribution function (fODF) metrics computation using FSL,⁷ MRtrix3,⁸ ANTs⁹ sequentially, and DIPY-based SCILPY** tools.¹⁰

DIPY-based SCILPY (scil compute local tracking.py) was used to build a whole-brain tractogram for each mouse brain using the fODF image, 100 seeds per voxel in the whole white matter mask, a step size of 0.05 mm, a maximum angle of 20°, and a streamline length between 2 and 20 mm. Streamlines of the right hemisphere between the sensory-motor cortex and the thalamus were extracted and concatenated from the 20 specimens to be used as a sample cortico-thalamic bundle in the current study.

2.2 Data acquisition and pre-processing

Our framework uses the Allen Software Development Kit (AllensDK, Version 0.16.3)*** to import data from the AMBCA, either by interrogating the Application Programming Interface (API) or by storing data in a cache to prevent unnecessary re-downloading. Note that all downloaded data with the AllensDK are in the Nearly Raw Raster Data (NRRD) format. When stored in the cache, no modifications are applied to them. Modifications are applied when saving resulting transformed data on the user's device (Sec. 2.3). Since the Neuroimaging Informatics Technology Initiative (NIFTI) format is the most common data format when working with MRI, we convert them from NRRD to NIFTI. Similarly, the Allen data are PIR+-oriented (PIR+ refers to Posterior Inferior Right neurosurgical reference space), while they must be RAS+-oriented (RAS+ refers to Right Anterior Superior neuroanatomic reference space) (Sec. 2.3.1) to interact and visualize data combinations in the MI-Brain software¹¹ (Sec. 2.3). Also, we only considered volumes with voxel resolutions from 25 μ m to 100 μ m among the ones offered by the AMBCA. The downloaded resolution

* <https://neuroinformatics.nl/HBP/allen-connectivity-viewer/streamline-downloader.html>

** <https://github.com/scilus/scilpy>

*** <https://allensdk.readthedocs.io/>

depends on the User's application and determines the bounding box. In addition, all coordinates in the Allen Mouse brain Common Coordinate Framework (CCFv3) are stored in microns. Unit conversion into voxels is done with the UDS (see the 3D-coordinates method in Sec. 2.3.3).

The Mouse Brain Average Template is mandatory to compute the transformation matrix between Allen and the UDS. Hence, it is acquired from the AllenSDK Reference Space API. Then, given an experiment identifier from the AMBCA, 3D viral projection maps from the Allen are downloaded, interrogating the AllenSDK Mouse Connectivity Cache module. Following a similar process, mouse brain structure masks are downloaded from the AllenSDK Reference Space API using a brain structure identifier. These identifiers are parts of the 840 structures list of the *Mouse Connectivity - Target Search* tool. They are also stored in the User's Cache.

Finally, Allen streamlines can also be downloaded, given a list of identifiers, and stored in the User's cache without modifications. They are aligned on the ADS, but the values of each coordinate in streamlines are in millimeters rather than microns. Hence, these values are converted into voxels when loading the pre-saved data from the cache.

2.3 Data registration and transformations

We use the Python package ANTsPyX, which is the Python version of the Advanced Normalization Tools (ANTs).⁹ Given two reference spaces (ADS and UDS), our method consists of an initial pre-processing basis transformation followed by an affine registration (Fig. 2). Thus the overall transformations can be done to points, volumes, or streamlines. They will be presented in the last part of this section (see 2.3.3).

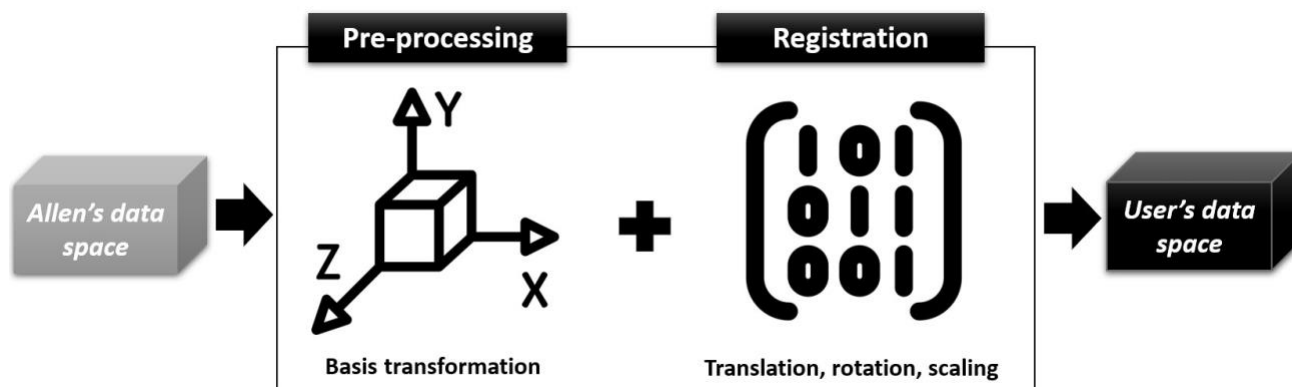


Figure 2. Steps of the data registration and transformation pipeline to link the Allen's Data Space (ADS) to the User's Data Space (UDS).

2.3.1 Pre-processing

Initial alignments illustrate that pre-processing is mandatory to achieve registration (pre-processing in Fig. 2). First, Allen's data need an orientation modification to match the User's own. For example, switching the PIR+-oriented ADS to a RAS+-oriented UDS. We generalized such pre-processing for any reference space transformation from ADS to UDS and conversely.

We use the Nibabel orientations methods to retrieve the orientation of UDS and ADS. Given the 4x4 affine matrix of a 3D volume, the method first retrieves the orientation axes codes ("positive end of voxel axes", axcodes); for example, "R", "A", "S". Second, it creates a two-column matrix storing the orientation of a particular axcode. The first column contains the axes indexes sorted by row, taking RAS+ as a default reference. For example, it is $[0, 1, 2]^T$ for RAS+ and $[1, 2, 0]^T$ for PIR+. The second column contains the axis orientations, taking the values 1 if it follows the reference direction, -1 otherwise. For example, $[-1, -1, 1]^T$ for PIR+, and $[1, 1, 1]^T$ for RAS+. Third, from these individual orientation two-column matrices (User's and Allen), this package provides a method that computes a similar matrix that stores the orientation values from A to B for any A and B. For example, the resulting matrix from PIR+ (Allen) to RAS+ (our data) is

$$\mathbf{T} = \begin{bmatrix} 1 & -1 \\ 2 & -1 \\ 0 & 1 \end{bmatrix} \quad (1)$$

for the direct transform and is

$$\mathbf{T}^{-1} = \begin{bmatrix} 2 & 1 \\ 0 & -1 \\ 1 & -1 \end{bmatrix} \quad (2)$$

for the reverse way (our data to Allen). Finally, such a back-and-forth transformation can be applied in many ways, either by using a method provided by the Nibabel package or manually, depending on the data type mentioned earlier. Further details are in Sec. 2.3.3.

2.3.2 Registration

The second step is to compute the affine transformation from the ADS to the UDS (registration in Fig. 2) using the ANTsPyX registration method. The registration parameters are set as follows: the user's data as the fixed image, Allen's data as the moving image, and the type of transform to "Affine". Affine means that the matrix is computed with rigid transformations and scaling. Whereas rigid is only about rotations and translation. However, since Allen's data comes in different resolutions (Sec. 2.2), a different matrix has to be computed for each. Computed matrices are saved in .mat format in the user's device and are reusable for the other scripts. Secondly, matrices are reused to align Allen's data with the user's data. For this purpose, ANTsPyX apply .transform function is used. Applying the transformation means that each voxel from the moving data is mapped to the corresponding voxel in the fixed data. Here, moving data can be any data downloaded from the AMBCA, and the fixed image is usually the user's template. The interpolator is set to "Nearest neighbor" by default because the values of each voxel are integers (e.g., in brain structure masks). However, the interpolator can also be set to "bSpline" for the interpolation of floating number values. It results in smooth renderings, usually better to distinguish nuances within MRI or fluorescent density signals within the visualization software. Finally, the inverse transform matrix is used to interrogate ADS from the UDS. Note that for points, we used the ANTsPyX apply .to point function.

2.3.3 Transformation

This part describes the algorithms used to achieve the complete transformation (pre-processing and registration, see Fig. 2) from the ADS to the UDS. There are three ways of performing the transformation depending on the initial data types, i.e., volumes, 3D coordinates, and streamlines.

Volume: The volume transformation method is applied to the viral-based AMBCA projection maps and the 3D brain structure masks (Fig. 3). The input is a PIR+-oriented data type within the ADS bounding box. Each point has a 3D coordinate expressed in voxel within the bounding box at the chosen resolution. The volumetric input data axes are first reoriented following the 2.3.1 process (Fig. 3A). Thus, the volume becomes UDS-oriented but is still within the ADS bounding box. So, by applying the ANTsPyX affine registration matrix on the reoriented volume, we get it aligned on the UDS within its appropriate bounding box (see 2.3.2 and Fig. 3B).

3D Coordinates: Fig. 4 presents the different steps to transform 3D coordinates. The goal is to allow the User to interact with their own and Allen's data naturally in their visualization software without considering the different spaces where these data reside (see examples in Sec. 3). To obtain PIR+-oriented coordinates in ADS (in microns) from RAS+-oriented coordinates in UDS (in voxels) (UDS to ADS, or $A \rightarrow B \rightarrow C$ in Fig. 4), the first step is to apply the ANTsPyX transformation matrix on the user-defined point itself (see 2.3.2 and Fig. 4A). One way to apply the transformation to a single point is to label it in a synthetic volume and use the volume transformation method

described above. An advantage of this approach is that it can generalize to other types of user-based annotations (e.g. spheres, ROIs, dense labels, etc.). Hence, a synthetic volume of dimensions corresponding to the ADS bounding box is created to compute in the ADS a point position selected by the user in their visualization software. All values are set to 0 except at the position of the selected point. The volume transformation is then applied to the synthetic annotation volume. The transformed point is extracted using the maximum value within the reoriented array. We thus obtain a PIR+-oriented point within the ADS bounding box in voxel coordinates. The second step is reorienting the axes from RAS+ to PIR+ (see 2.3.1 and Fig. 4B). The last step is simply to convert each component into microns (see Fig. 4C) by multiplying them by the Allen resolution value. To obtain RAS+-oriented coordinates in voxels within UDS bounding box from PIR+-oriented coordinates in microns within the ADS bounding box (ADS to UDS, or C → B → A in Fig. 4), the same process is inverted and starts from the end (conversion from microns to voxels, reorienting the axes to UDS with a synthetic volume and applying the inverse transformation matrix (see Fig. 4 C, B, and A, and see examples in Sec. 2.4 and 3.1).

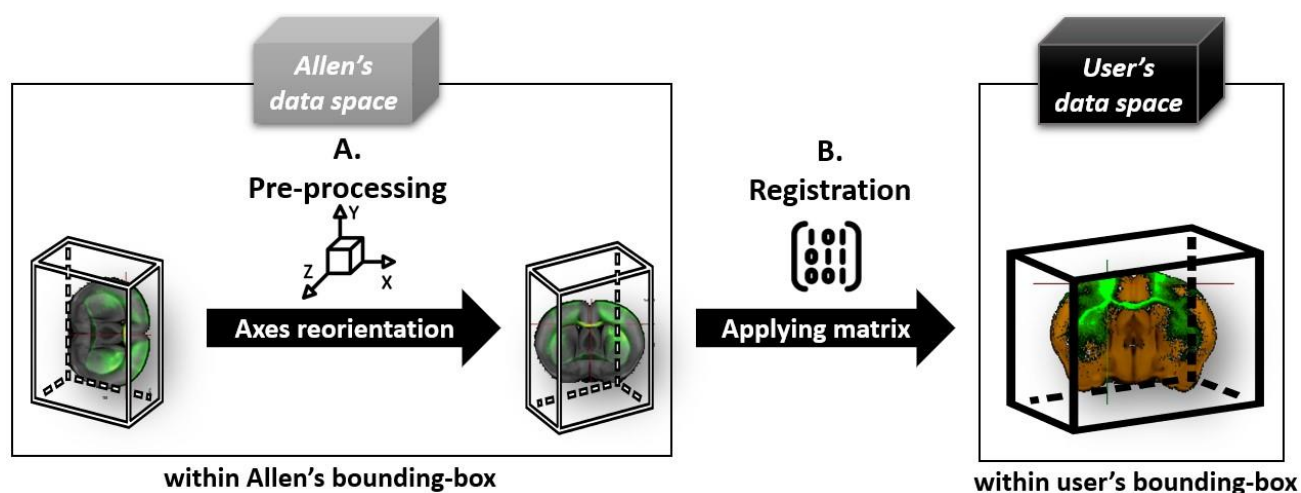


Figure 3. Two steps explaining the volumetric transformation algorithm between AMBCA (left) and UDS (right). Axes re-orientation (A) refers to Sec. 2.3.1, and Registration (B) refers to Sec. 2.3.2.

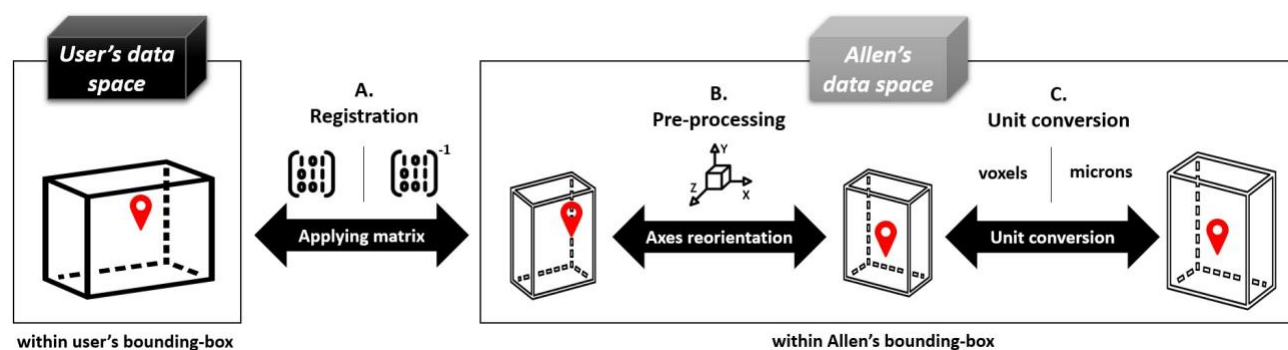


Figure 4. Three steps explaining the back-and-forth 3D-coordinates transformation algorithm between UDS (left) and AMBCA (right). Registration (A) refers to Sec. 2.3.2, Axes re-orientation (B) refers to Sec. 2.3.1, and Unit conversion (C).

Streamlines: By definition, streamlines correspond to an array of 3D coordinates. So here, the algorithm follows the same logic. However, tractograms contain multiple streamlines and a streamline itself can contain a lot of points. So, reorienting a tractogram using the volume transformation approach point by point could be computer time-consuming. Thus, we opted for a faster and more precise solution, also applicable for single 3D coordinates. In this case, the unit conversion (Fig. 4C) is not needed since the values of each coordinate are converted into voxels while loading streamlines from the user's cache. The difference is in the axis reorientation step (Fig. 4B). We opted to reorient points without a synthetic volume. Here, we use information about the Nibabel's orientation two-column matrix manually (Sec. 2.3.1). In other words, we change the indices of each coordinate using the first column of the matrix and change the coordinate direction using the other column of the matrix. Third, if a coordinate appears negative, we add the bounding box value of its axis to retrieve the actual offset point. The registration step remains the same (Fig. 4A). Thus, the whole tractogram is transformed by iterating on each point and each streamline. Each streamline is built as a list of transformed points.

2.4 Data interaction and visualization

Once the alignments are achieved, both Allen and User data can be used in natural interaction, e.g., by inspecting and analyzing both modalities with visualization software such as MI-Brain.¹¹ Note that this visualization software requires RAS+-oriented data to display them correctly. If the User Data Space is not in RAS+, data transformed by our tools will only be UDS-Oriented. Hence, an additional reorientation has to be done by the user if he/she wants to interact with this visualization software. With MI-Brain, a user can interact with 3D coordinates, 3D shapes such as spheres (displayed as a green circle in Fig. 5), tractograms, colors, and so on. The resulting data produced by our m2m toolkit take advantage of these interactions to produce additional data. For example, coordinates within the software can be retrieved by selecting a voxel or creating a region of interest (ROI) at a specific location.

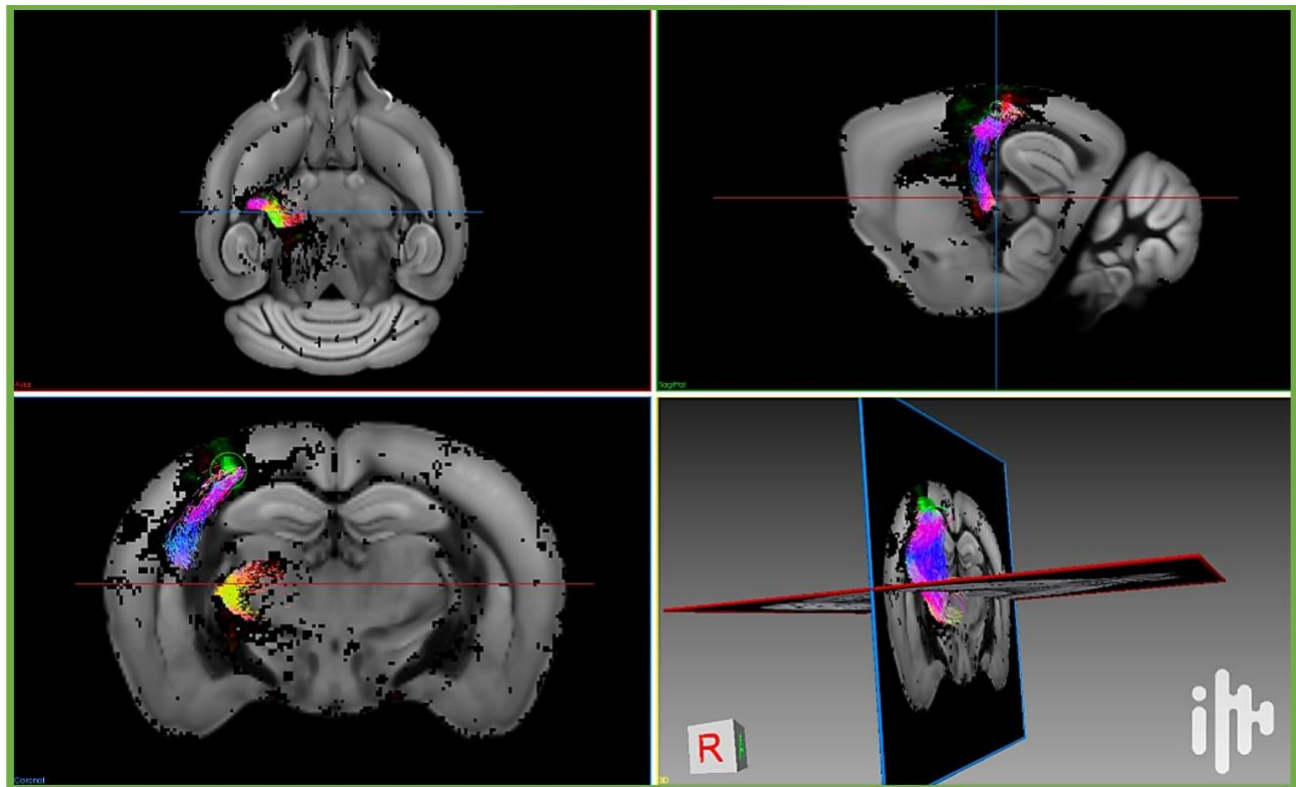
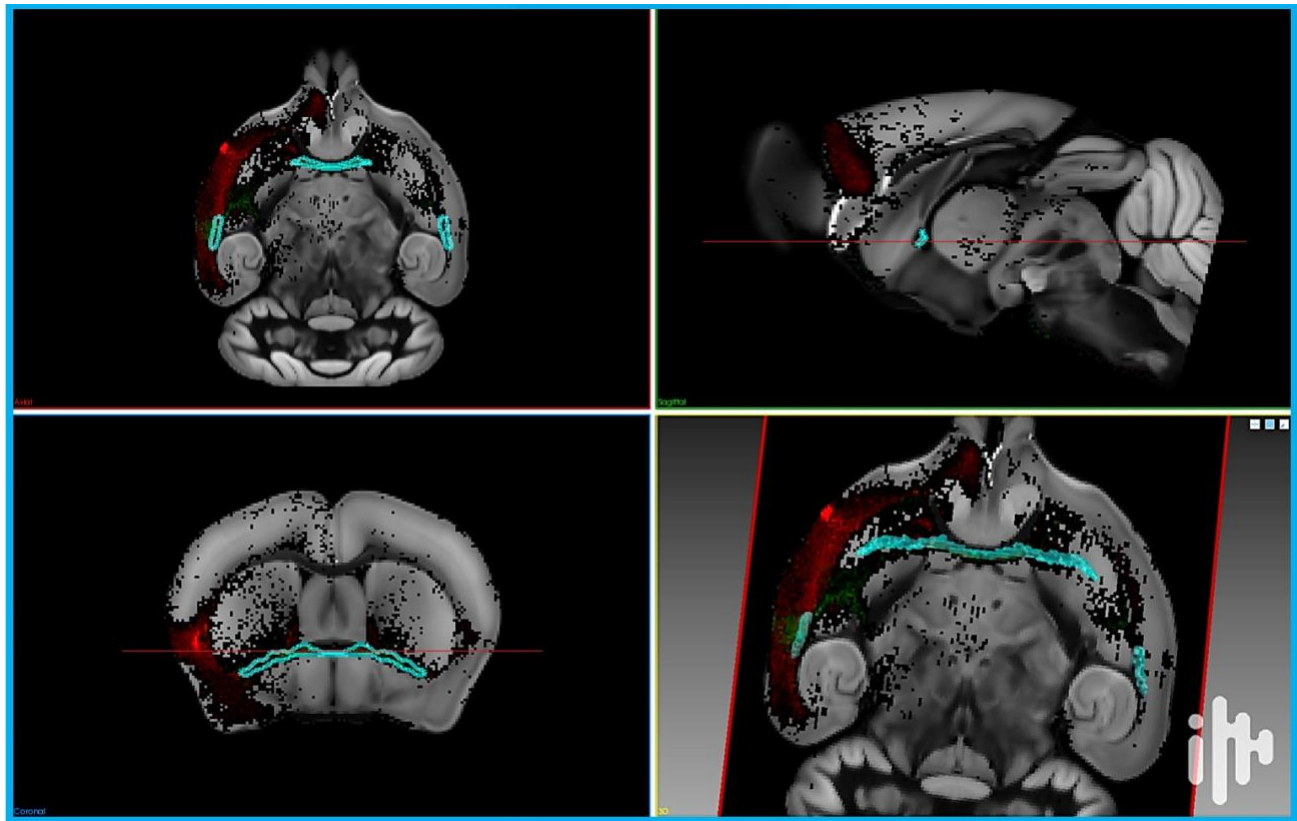


Figure 5. Visualization of a fiber tract analysis study using the MI-Brain software.¹¹ A cortico-thalamic bundle in the UDS overlaid with two Allen projection density maps originating from different injection locations (red and green). The grayscale image is the AVGT average mouse brain template.

Another example is the capacity to visualize overlapping points. For instance, voxels from multiple AMBCA experiment maps are stored in each color channel of the volumetric data to display. Red, green, and blue. Thus, green and red channels can appear in orange-to-yellowish tones depending on the green and red intensities. Combining this with visualization with coordinate and 3D shape interactions can better understand the overlays between various AMBCA experiments (example of ROI in blue Fig. 6). Finally, for tractograms, when the visualization software allows the filtering of streamlines that pass through an ROI, other possibilities for analyzing emerge. For example, while importing a projection map, the user can create a binarized mask upon a certain threshold (see 3.1) that can be used as an ROI to compute orientations of streamlines in interactions with other applications.



 Fiber crossing study

Figure 6. Visualization using the MI-Brain software¹¹ of a fiber crossing study with two Allen projection density maps (green and red) and a 3D mask (blue) located where the two experiments have a high co-localized density signal superior to a given threshold chosen by the user. The grayscale image is the AVGT average mouse brain template.

3. RESULTS

We here present three complementary use cases of applying our **m2m** toolkit.

3.1 Download AMBCA projection maps and useful information

This first use case consists of importing an AMBCA projection map given an experiment identifier and aligning it on the UDS (Fig. 5). It is achieved by applying the transformation matrix to the density volume. This functionality has some options. For example, it is possible to download information about the injection center of the experiment and represent it as a sphere. Itself displayable in the user's visualization tool. Also, injection coordinates (within UDS and ADS), as well as the identifier of the experiment, can be stored in a JSON file to interact with other functionalities. Finally, this functionality also allows the creation of a thresholded binary map of this experiment. Displaying it in the user visualization software can help identify a high connectivity signal.

3.2 Finding fiber-crossing regions with AMBCA projection maps and CCFv3 brain structures

In this second use case, the goal is to find potential fiber-crossing regions in the mouse brain. This functionality first downloads up to three AMBCA projection density maps with their respective injection sites or high-density locations. Injection sites can be retrieved from the first functionality. Others can also be chosen directly in the visualization tool (see 2.4). We can recover the corresponding locations in ADS by applying the inverse transformation matrix on these specifically chosen locations within the UDS. The corresponding projection density maps are then downloaded and aligned in the UDS. Secondly, it merges these previous maps in a colorbased volume, assigning each color channel to an experiment. It allows its visualization to consider overlapping points. Thirdly, this functionality finds brain structures where the density signal within the maps overlaps above a certain threshold chosen by the user. Given a list of 840 brain structures corresponding to the *Mouse Connectivity Target Search structures* from the Allen Mouse Brain Connectivity Atlas, we can iterate in this list and recover which region, in which hemisphere, contains a projection density signal superior to a given threshold for each experiment of the color-based volume. Once the density signal superior to a given threshold is verified in each map, we download the volumetric brain masks with the AllenSDK, combine them like the maps in a binary volume, and align them on the UDS (shown in blue in Fig.5). Also, identifiers of the experiments and brain structures are saved in a JSON file, like for the first functionality. We can, for example, use one of the identifiers to download information about a precise experiment within the combined ones. Furthermore, it can be interesting to investigate this potential crossing region by combining with tractography tools to make deeper analyses.

3.3 Importing CCFv3 tractograms and creating a custom bundle of streamlines

As a third use case, **m2m** offers the possibility to import fluorescent-derived tractograms from Allen's CCFv3. It can be done by giving a CSV list of one-to-n identifiers. If multiple identifiers are given, all the streamlines are combined in a single tractogram. They correspond to AMBCA experiments. Hence, the interaction with the two other use cases mentioned above is natural because they either take them as input or return them. So all of these functionalities can be used as a continuous back-and-forth process. Moreover, creating a custom bundle of streamlines is often useful for analyzing specific brain locations. This task can be done in interaction with results obtained with previous functionalities. It will create a bundle by extracting fluorescent-derived streamlines that pass through a given location. The latter can be a point around an injection region, an ROI from the UDS, or a CCFv3 brain structure mask.

4. CONCLUSION

Our **m2m** toolkit provides a turn-key series of python-based "back and forth" actions to combine AMBCA and User's data. The latter can be of any neuroimaging origin, e.g., anatomical, diffusion, and even functional MRI, tractography, serial blockface histology, optical coherence tomography, etc. M2M can thus be used for multiple other purposes with additional neuroimaging analyses highlighting further investigation of the mouse brain connectome. It is a unique tool in connectomics to link structural and functional data between macroscopic to mesoscopic scales.

ACKNOWLEDGMENTS

This work and M.A.H. summer internship scholarship was supported by a grant from the Qu'bec Bio-Imaging Network (QBIN Pilot Project PP18.07). The diffusion MRI and tractography data belong to a study conducted in the framework of the University of Bordeaux's IdEx "Investments for the Future" program RRI "IMPACT" that received financial support from the French government. The authors also wish to thank Maxime Descoteaux and his SCIL group from Universit'e Sherbrooke, and Fr'ed'eric Lesage and his LIOM group from Polytechnique Montr'eal for their useful advice and insights.

REFERENCES

- [1] Leergaard, T. B. and Bjaalie, J. G., "Atlas-based data integration for mapping the connections and architecture of the brain," *Science* **378**(6619), 488–492 (2022).
- [2] Ragan, T., Kadiri, L. R., Venkataraju, K. U., Bahlmann, K., Sutin, J., Taranda, J., Arganda-Carreras, I., Kim, Y., Seung, H. S., and Osten, P., "Serial two-photon tomography for automated ex vivo mouse brain imaging," *Nature Methods* **9**(3), 255–258 (2012).
- [3] Oh, S. W., Harris, J. A., Ng, L., Winslow, B., Cain, N., Mihalas, S., Wang, Q., Lau, C., Kuan, L., Henry, A. M., Mortrud, M. T., Ouellette, B., Nguyen, T. N., Sorensen, S. A., Slaughterbeck, C. R., Wakeman, W., Li, Y., Feng, D., Ho, A., Nicholas, E., Hirokawa, K. E., Bohn, P., Joines, K. M., Peng, H., Hawrylycz, M. J., Phillips, J. W., Hohmann, J. G., Wohnoutka, P., Gerfen, C. R., Koch, C., Bernard, A., Dang, C., Jones, A. R., and Zeng, H., "A mesoscale connectome of the mouse brain," *Nature* **508**(7495), 207–214 (2014).
- [4] Wang, Q., Ding, S.-L., Li, Y., Royall, J., Feng, D., Lesnar, P., Graddis, N., Naeemi, M., Facer, B., Ho, A., Dolbeare, T., Blanchard, B., Dee, N., Wakeman, W., Hirokawa, K. E., Szafer, A., Sunkin, S. M., Oh, S. W., Bernard, A., Phillips, J. W., Hawrylycz, M., Koch, C., Zeng, H., Harris, J. A., and Ng, L., "The Allen Mouse Brain Common Coordinate Framework: A 3D Reference Atlas," *Cell* **181**(4), 936–953.e20 (2020).
- [5] Calabrese, E., Badea, A., Cofer, G., Qi, Y., and Johnson, G. A., "A Diffusion MRI Tractography Connectome of the Mouse Brain and Comparison with Neuronal Tracer Data," *Cerebral Cortex* **25**(11), 4628–4637 (2015).
- [6] Lefebvre, J., Delafontaine-Martel, P., and Lesage, F., "A Review of Intrinsic Optical Imaging Serial Blockface Histology (ICI-SBH) for Whole Rodent Brain Imaging," *Photonics* **6**(2), 66 (2019).
- [7] Jenkinson, M., Beckmann, C. F., Behrens, T. E., Woolrich, M. W., and Smith, S. M., "FSL," *NeuroImage* **62**(2), 782–790 (2012).
- [8] Tournier, J.-D., Smith, R., Raffelt, D., Tabbara, R., Dhollander, T., Pietsch, M., Christiaens, D., Jeurissen, B., Yeh, C.-H., and Connelly, A., "MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation," *NeuroImage* **202**, 116137 (2019).
- [9] Avants, B., Tustison, N., and Song, G., "Advanced Normalization Tools: V1.0," (2009).
- [10] Garyfallidis, E., Brett, M., Amirbekian, B., Rokem, A., Walt, S. v. d., Descoteaux, M., Nimmo-Smith, I., and Contributors, D., "Dipy, a library for the analysis of diffusion MRI data," *Frontiers in Neuroinformatics* **8**, 8 (2014).
- [11] Rheault, F., Houde, J.-C., Goyette, N., Morency, F. C., and Descoteaux, M., "MI-Brain, a software to handle tractograms and perform interactive virtual dissection," in *[ISMRM Diffusion Study Group]*, (2016).