The ANTsX Ecosystem for Mapping the Mouse Brain

Nicholas J. Tustison[1], Min Chen[2], Fae N. Kronman[3], Jeffrey T. Duda[2], Clare Gamlin[4], Mia G. Tustison, Lydia Ng[4], Yongsoo Kim[3], and James C. Gee[2]

[1]Department of Radiology and Medical Imaging, University of Virginia, Charlottesville, VA
[2]Department of Radiology, University of Pennsylvania, Philadelphia, PA
[3]Department of Neural and Behavioral Sciences, Penn State University, Hershey, PA
[4]Allen Institute for Brain Science, Seattle, WA

Corresponding author:
Nicholas J. Tustison, DSc
Department of Radiology and Medical Imaging
University of Virginia
ntustison@virginia.edu

Abstract

Precision mapping techniques coupled with high resolution image acquisition of the mouse brain permit the study of the spatial organization of gene activity and their mutual interaction for a comprehensive view of salient structural/functional relationships. Such research is facilitated by standardized anatomical coordinate systems, such as the well-known Allen Common Coordinate Framework (AllenCCFv3), and the ability to spatially map to such standardized spaces. The Advanced Normalization Tools Ecosystem (ANTsX) is a comprehensive open-source software toolkit for generalized quantitative imaging, which includes template building and mapping functionality, with applicability to multiple organ systems, modalities, and animal species. Herein, we illustrate the utility of ANTsX for generating precision spatial mappings of the mouse brain and potential subsequent quantitation. First, we provide ANTsX-based protocols for mapping MERFISH, fMOST, and lightsheet datasets to AllenCCFv3 accounting for common artefacts and other confounds. Novel contributions include recently developed ANTsX functionality for generating a velocity flow-based mapping spanning the spatiotemporal domain of a longitudinal trajectory which we apply to the Developmental Common Coordinate Framework (DevCCF). Additionally, we present an automated structural morphological pipeline for determining volumetric and cortical thickness measurements analogous to the well-utilized ANTsX pipeline for human neuroanatomical structural morphology. This latter development also illustrates a more general open-source ANTsX framework for determining tailored brain parcellations using the AllenCCFv3 and DevCCF templates.

1 Introduction

Over the past two decades there have been significant advancements in mesoscopic analysis of the mouse brain. It is currently possible to track single cell neurons in mouse brains,[1] observe whole brain developmental changes on a cellular level,[2] associate brain regions and tissues with their genetic composition,[3] and locally characterize neural connectivity.[4] Much of this scientific achievement has been made possible due to breakthroughs in high resolution imaging techniques that permit submicron, 3-D imaging of whole mouse brains. Associated research techniques such as micro-optical sectioning tomography,[6] tissue clearing,[1,7] spatial transcriptomics[9] are all well-utilized in the course of scientific investigations of mesoscale relationships in the mouse brain.

An important component of this research is the ability to map the various image data to anatomical reference frames[11] for inferring spatial relationships between structures, cells, and genetics. This has motivated the development of detailed structural image atlases of the mouse brain. Notable examples include the Allen Brain Atlas and Coordinate Frameworks (AllenCCFv3),[13] the Waxholm Space,[14] and more recently, the Developmental Common Coordinate Framework (DevCCF).[15] Despite the significance of these contributions, challenges still exist in large part due to the wide heterogeneity in associated study-specific image data. For example, variance in the acquisition methods can introduce artifacts such as tissue distortion, holes, bubbles, folding, tears, and missing slices. These complicate assumed correspondence for conventional spatial mapping approaches.

1.1 Mouse-specific brain mapping software

To address such challenges, several software packages have been developed over the years comprising solutions of varying comprehensibility, sophistication, and availability. An early contribution to the community was the Rapid Automatic Tissue Segmentation (RATS) package[16] for brain extraction. More recently, several publicly available packages comprise well-established package dependencies originally developed on human brain data. SPMMouse,[17] for example, is based on the well-known Statistical Parametric Mapping

(SPM) Matlab-based toolset.[18] The automated mouse atlas propagation (aMAP) tool is largely a front-end for the NiftyReg image registration package[19] applied to mouse data which is currently available as a Python module.[20] NiftyReg is also used by the Atlas-based Imaging Data Analysis (AIDA) MRI pipeline[21] as well as the Multi Atlas Segmentation and Morphometric Analysis Toolkit (MASMAT). Whereas the former also incorporates the FMRIB Software Library (FSL)[22] for brain extraction and DSIStudio[23] for DTI processing, the latter uses NiftySeg and multi-consensus labeling tools[24] for brain extraction and parcellation. In addition, MASMAT incorporates N4 bias field correction[25] from the Advanced Normalization Tools Ecosystem (ANTsX)[26] as do the packages Multi-modal Image Registration And Connectivity anaLysis (MIRACL),[27] Sammba-MRI,[28] and Small Animal Magnetic Resonance Imaging (SAMRI).[29] However, whereas Saamba-MRI uses AFNI[30] for image registration; MIRACL, SAMRI, and BrainsMapi[31] all use ANTsX registration tools. Other packages use landmark-based approaches to image registration including SMART—[32]an R package for semi-automated landmark-based registration and segmentation of mouse brain based on WholeBrain.[33] FriendlyClearMap[34] uses the landmark-based registration functionality of Elastix.[35] Finally, the widespread adoption of deep learning techniques has also influenced development in mouse brain imaging methodologies. For example, if tissue deformations are not considered problematic for a particular dataset, DeepSlice can be used to determine affine mappings[36] with the optimal computational efficiency associated with neural networks.

1.2 The ANTsX Ecosystem for mouse brain mapping

As noted previously, many of the existing packages designed for processing mouse brain image data use ANTsX tools for core processing steps in various workflows, particularly its pairwise, intensity-based image registration capabilities and bias field correction. Historically, ANTsX development is originally based on fundamental approaches to image mapping,[37–39] particularly in the human brain, which has resulted in core contributions to the field such as the well-known Symmetric Normalization (SyN) algorithm.[40] Since its development, various independent platforms have been used to evaluate ANTsX image registration capabilities in the context of different application foci which include multi-site brain MRI data,[41] pulmonary CT data,[42] and most recently, multi-modal brain registration in the presence of tumors.[43]

Apart from its registration capabilities, ANTsX comprises additional functionality such as template generation,[44] intensity-based segmentation,[45] preprocessing,[25,46] deep learning networks,[26] and other miscelleneous utilties (e.g.,)[47] (see Table 1). The comprehensive use of the toolkit has demonstrated superb performance in multiple application areas (e.g., consensus labeling,[48] brain tumor segmentation,[49] and cardiac motion estimation[50]). Importantly, ANTs is built on the Insight Toolkit (ITK)[51] deriving benefit from the open-source community of scientists and programmers and providing an important resource for algorithmic development, evaluation, and improvement.

1.3 ANTsX-based open-source contributions

Consistent with previous ANTsX development, the newly introduced capabilities introduced below are available through ANTsX (specifically, via R and Python ANTsX packages), and illustrated through self-contained examples in the ANTsX tutorial (https://tinyurl.com/antsxtutorial) with a dedicated GitHub repository specific to this work (https://github.com/ntustison/ANTsXMouseBrainMapping).

1.3.1 The DevCCF velocity flow model

Recently, the Developmental Common Coordinate Framework (DevCCF) was introduced to the mouse brain research community as a public resource[15] comprising symmetric atlases of multimodal image data and anatomical segmentations defined by developmental ontology. These templates sample the mouse embryonic days (E) 11.5, E13.5, E15.5, E18.5 and postnatal day (P) 4, P14, and P56. Modalities include light sheet flourescence miscroscopy (LSFM) and at least four MRI contrasts per developmental stage. Anatomical parcellations are also available for each time point and were generated from ANTsX-based mappings of gene expression and other cell type data. Additionally, the P56 template was integrated

Table 1: Sampling of ANTsX functionality

ANTsPy: Preprocessing	
bias field correction	`n4_bias_field_correction(...)`
image denoising	`denoise_image(...)`
ANTsPy: Registration	
image registration	`registration(...)`
image transformation	`apply_transforms(...)`
template generation	`build_template(...)`
landmark registration	`fit_transform_to_paired_points(...)`
time-varying landmark reg.	`fit_time_varying_transform_to_point_sets(...)`
integrate velocity field	`integrate_velocity_field(...)`
invert displacement field	`invert_displacement_field(...)`
ANTsPy: Segmentation	
MRF-based segmentation	`atropos(...)`
Joint label fusion	`joint_label_fusion(...)`
diffeormorphic thickness	`kelly_kapowski(...)`
ANTsPy: Miscellaneous	
Regional intensity statistics	`label_stats(...)`
Regional shape measures	`label_geometry_measures(...)`
B-spline approximation	`fit_bspline_object_to_scattered_data(...)`
Visualize images and overlays	`plot(...)`
ANTsPyNet: Mouse-specific	
brain extraction	`mouse_brain_extraction(...modality="t2"...)`
	`mouse_brain_extraction(...modality="ex5"...)`
brain parcellation	`mouse_brain_parcellation(...)`
cortical thickness	`mouse_cortical_thickness(...)`
super resolution	`mouse_histology_super_resolution(...)`

ANTsX provides state-of-the-art functionality for processing biomedical image data. Such tools, including deep learning networks, support a variety of mapping-related tasks. A more comprehensive listing of ANTsX tools with self-contained R and Python examples is provided as a gist page on GitHub (https://tinyurl.com/antsxtutorial).

with the Allen CCFv3 to further increase the practical utility of the DevCCF. These processes, specifically template generation and multi-modal image mapping, were performed using ANTsX functionality in the presence of image mapping difficulties such as missing data and tissue distortion.

Given the temporal gaps in the discrete set of developmental atlases, we also provide an open-source framework for inferring correspondence within the temporally continuous domain sampled by the existing set of embryonic and postnatal atlases of the DevCCF. This recently developed functionality permits the generation of a diffeomorphic velocity flow transformation model,[52] influenced by previous work.[53] The resulting time-parameterized velocity field spans the stages of the DevCCF where mappings between any two continuous time points within the span bounded by the E11.5 and P56 atlases is determined by integration of the optimized velocity field.

1.3.2 Structural morphology and cortical thickness in the mouse brain

One of the most frequently utilized pipelines in the ANTsX toolkit is that of estimating cortical thickness maps in the human brain. Beginning with the Diffeomorphic Registration-based Cortical Thickness (DiReCT) algorithm,[54] this was later expanded to include a complete processing framework for human brain cortical thickness estimation for both cross-sectional[55] and longitudinal[56] data using T1-weighted MRI. These pipelines were later significantly refactored using deep learning innovations.[26]

In contrast to the pipeline development in human data,[26] no current ANTsX tools exist to create adequate training data for the mouse brain. In addition, mouse brain data acquisition often has unique issues, such as lower data quality or sampling anisotropy which limits its applicability to high resolution resources (e.g., AllenCCFv3, DevCCF), specifically with respect to the corresponding granular brain parcellations derived from numerous hours of expert annotation leveraging multimodal imaging resources.

Herein, we introduce a mouse brain cortical thickness pipeline for T2-weighted MRI comprising two novel deep learning components: two-shot learning brain extraction from data augmentation of two ANTsX templates generated from two open datasets[57,58] and single-shot brian parcellation derived from the AllenCCFv3 labelings mapped to the corresponding DevCCF P56 T2-w component. Although we anticipate that this cortical thickness pipeline will be beneficial to the research community, this work demonstrates more generally how one can leverage ANTsX tools for developing tailored brain parcellation schemes using these publicly available resources. Evaluation is performed on an independent open data set[59] comprising longitudinal acquisitions of multiple specimens.

2 Results

2.1 The DevCCF Velocity Flow Model

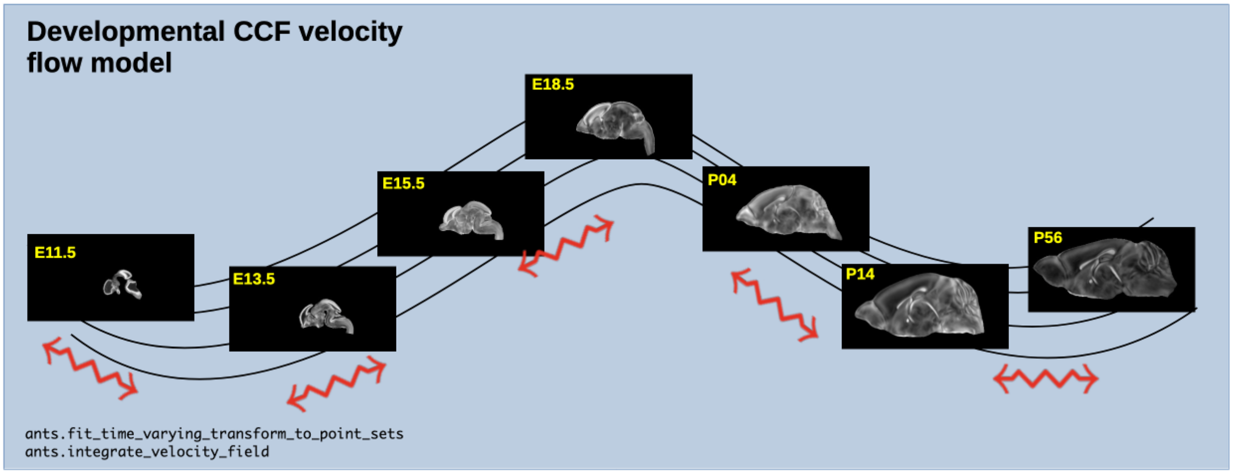

Developmental CCF velocity flow model
E11.5
E13.5
E15.5
E18.5
P04
P14
P56
ants.fit_time_varying_transform_to_point_sets
ants.integrate_velocity_field

Figure 1: The spatial transformation between any two time points within the DevCCF longitudinal developmental trajectory is available through the use of ANTsX functionality for generating a velocity flow model.

To continuously interpolate transformations between the different stages of the DevCCF atlases, a velocity flow model was constructed using DevCCF derived data and functionality recently introduced into both the ANTsR and ANTsPy packages. Both platforms include a complete suite of functions for determining dense correspondence from sparse landmarks based on a variety of transformation models ranging from standard linear models (i.e., rigid, affine) to deformable diffeomorphic models (e.g, symmetric normalization).[40] The latter set includes transformation models for both the pairwise scenario and for multiple sets, as in the case of the DevCCF. ANTsX, being built on top of ITK, uses an ITK image data structure for the 4-D velocity field where each voxel contains the x, y, z components of the field at that point.

2.1.1 Data preparation

Labeled annotations are available as part of the original DevCCF and reside in the space of each developmental template which range in resolution from $31.5 - 50\mu$m. Across all

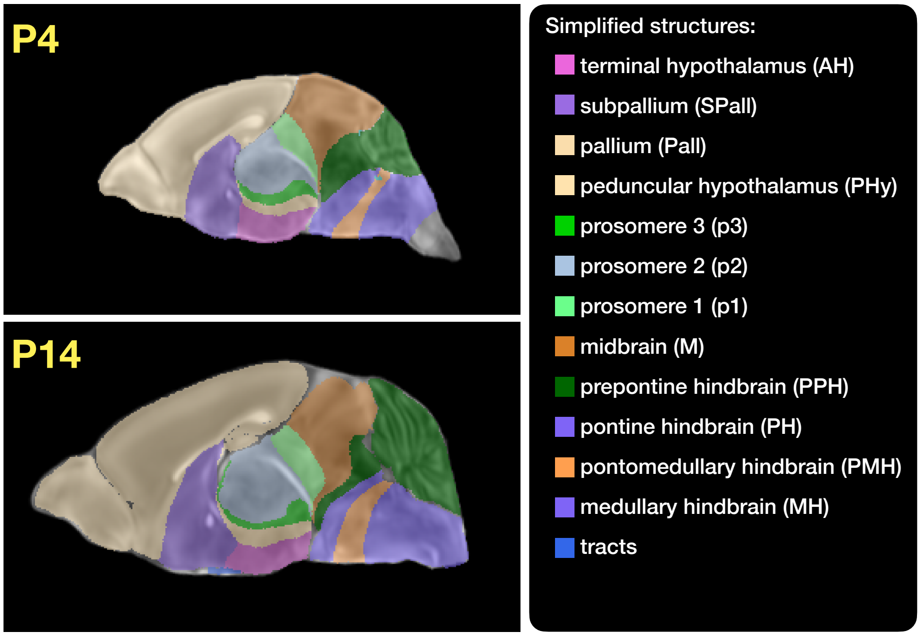


Figure 2: Annotated regions representing common labels across developmental stages which are illustrated for both P4 and P14.

atlases, the total number of labeled regions exceeds 2500. From these labels, a common set of 26 labels (13 per hemisphere) across all atlases were used for optimization and evaluation. These regions are illustrated for the P4 and P14 stages in Figure 2.

Prior to velocity field optimization, all data were rigidly transformed to a common space. Using the centroids for the common label set of each DevCCF atlas, each atlas was rigidly aligned to the space of the P56 atlas. In order to determine the landmark correspondence across DevCCF stages, the multi-metric capabilities of `ants.registration(...)` were used. Instead of performing intensity-based pairwise registration directly on these multi-label images, each label was used to construct a separate fixed and moving image pair resulting in a multi-metric registration optimization scenario involving 24 binary image pairs (each label weighted equally) for optimizing diffeomorphic correspondence between neighboring time point atlases using the mean squares metric and the symmetric normalization transform.

To generate the set of common point sets across all seven developmental atlases, the label boundaries and whole regions were sampled in the P56 atlas and then propagated to each atlas using the transformations derived from the pairwise registrations. We selected a sampling rate of 10% for the contour points and 1% for the regional points for a total number of points being per atlas being 173303 ($N_{contour} = 98151$ and $N_{region} = 75152$). Regional boundary points were weighted twice as those of non-boundary points during optimization.

2.1.2 Optimization

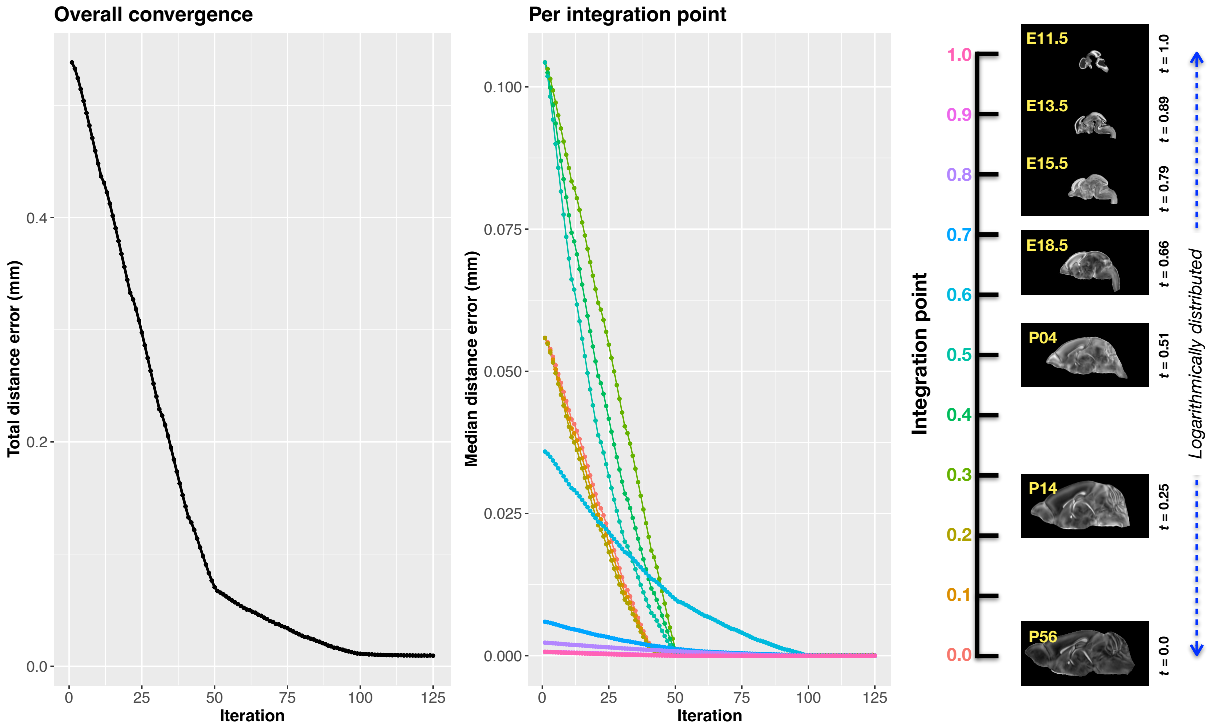


Figure 3: Convergence of the optimization of the velocity field for describing the transformation through the developmental stages from E11.5 through P56.

The velocity field was optimized using the input composed of the seven corresponding point sets and their associated weight values, the selected number of integration points for the velocity field ($N = 11$), and the parameters defining the geometry of the spatial dimensions of the velocity field. Thus, the optimized velocity field described here is of size $[256, 182, 360]$ (50μm isotropic) $\times 11$ integration points for a total compressed size of a little over 2 GB. This choice represented weighing the trade-off between tractability, portability, and accuracy. However, all data and code to reproduce the results described (with possible variation in the input parameters) are available in the dedicated GitHub repository.

The normalized time point scalar value for each atlas/point-set in the temporal domains $[0, 1]$ was also defined. Given the increasingly larger gaps in the postnatal timepoint sampling, we made two adjustments. Based on known mouse brain development, we used 28 days for the P56 data. We then computed the log transform of the adjusted set of time points prior to normalization between 0 and 1 (see the right side of Figure 3). This log transform, as part of the temporal normalization, significantly improved data spacing.

The max number of iterations was set to 200 with each iteration taking six minutes. At each iteration we looped over the 11 integration points. At each integration point, the velocity field estimate was updated by warping the two immediately adjacent point sets to the integration time point and determining the regularized displacement field between the two warped point sets. As with any gradient-based descent algorithm, this field was multiplied by a small step size ($\delta = 0.2$) before adding to the current velocity field. Convergence is determined by the average displacement error over each of the integration points. As can be seen in the left panel of Figure 3, convergence occurred around 125 iterations when the average displacement error over all integration points is minimized. The median displacement error at each of the integration points also trends towards zero but at different rates.

2.1.3 The transformation model

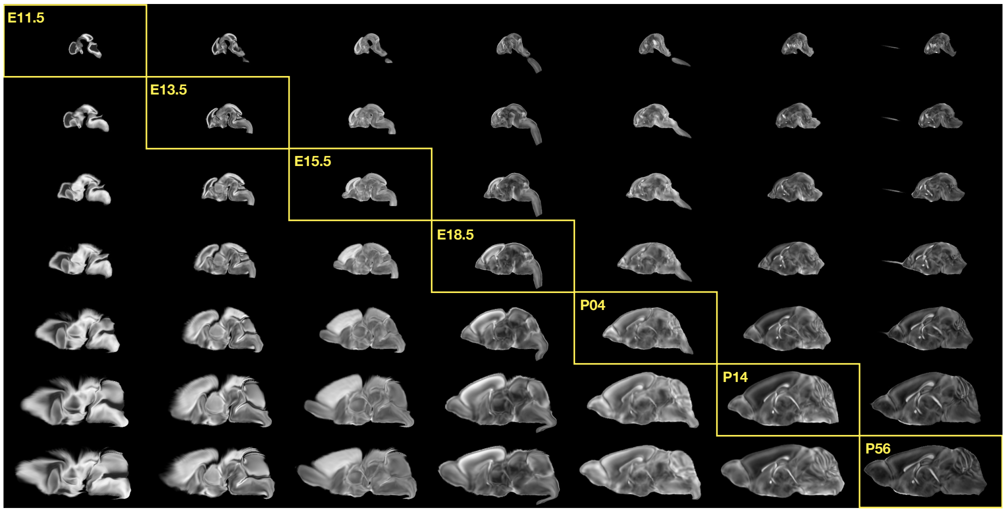


Figure 4: Mid-sagittal visualization of the effects of the transformation model in warping every developmental stage to the time point of every other developmental stage. The original images are located along the diagonal. Columns correspond to the warped original image whereas the rows represent the reference space to which each image is warped.

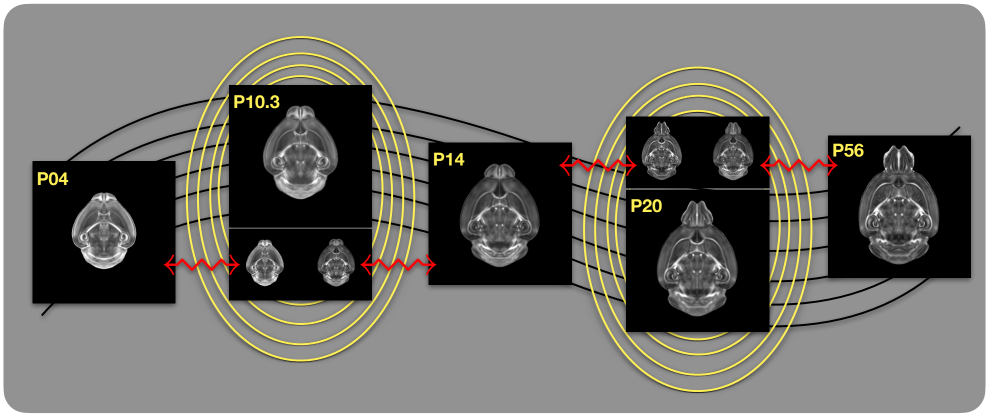


Figure 5: Illustration of the use of the velocity flow model for creating virtual templates at continuous time points not represented in one of the existing DevCCF time points. For example, FA templates at time point P10.3 and P20 can be generated by warping the existing temporally adjacent developmental templates to the target time point and using those images in the ANTsX template building process.

Once optimized, the resulting velocity field can be used to generate the deformable transform between any two continuous points within the time interval bounded by E11.5 and P56. In Figure 4, we transform each atlas to the space of every other atlas using the DevCCF transform model. Additionally, one can use this transformation model to construct virtual templates in the temporal gaps of the DevCCF. This is illustrated in Figure 5 where we used the optimized velocity field to construct virtual templates at time point P10.3 and P20—arbitrarily chosen simply to demonstrate the concept. After situating these time points within the normalized time point interval, the existing adjacent DevCCF atlases on either chronological side can be warped to the desired time point. A subsequent call to one of the ANTsX template building functions then permits the construction of the template at that time point. Note that both of these usage examples can be found in the GitHub repository previously given.

2.2 The Mouse Cortical Thickness Pipeline

One of the most well-utilized pipelines in the ANTsX toolkit is the generation of cortical thickness maps in the human brain from T1-weighted MRI. Starting with the novel Diffeomorphic Registration-based Cortical Thickness (DiReCT) algorithm,[54] a complete al-

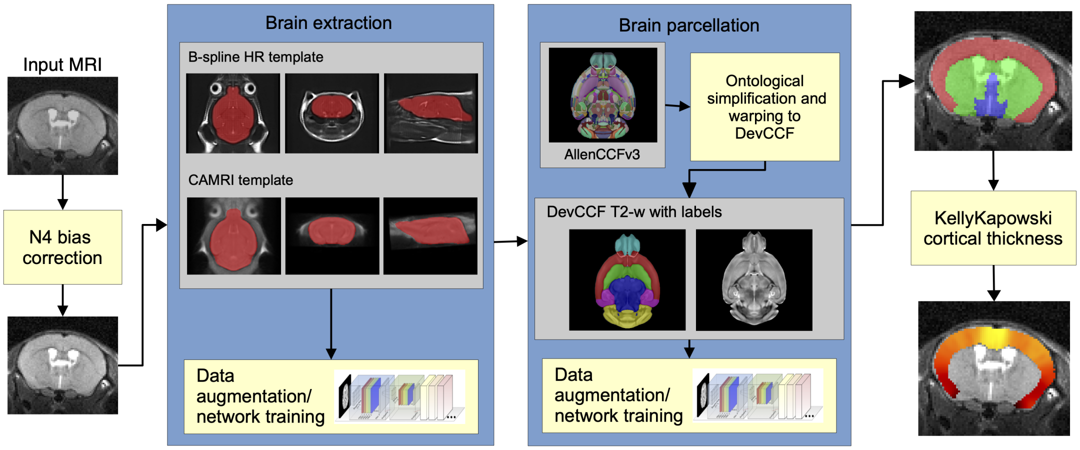

Input MRI
N4 bias correction
Brain extraction
B-spline HR template
CAMRI template
Data augmentation/ network training
Brain parcellation
AllenCCFv3
Ontological simplification and warping to DevCCF
DevCCF T2-w with labels
Data augmentation/ network training
KellyKapowski cortical thickness

Figure 6: The mouse brain cortical thickness pipeline integrating two deep learning components for brain extraction and brain parcellation prior to estimating cortical thickness. Both deep learning networks rely heavily on data augmentation on templates built from open data and provide an outline for further refinement and creating alternative parcellations for tailored research objectives.

gorithmic workflow was developed for both cross-sectional[55] and longitudinal[56] T1-weighted MR image data. This contribution was later refactored using deep learning[26] leveraging the earlier results[55] for training data.

In the case of the mouse brain, the lack of training data and/or tools to generate training data make a similar developmental trajectory difficult. In addition, mouse data is often characterized by unique issues such as frequent anisotropic sampling which are often in sharp contrast to the high resolution resources available within the community, e.g., AllenCCFv3 and DevCCF. Using ANTsX and other publicly available data resources, we developed a complete mouse brain structural morphology pipeline as illustrated in Figure 6 and detailed below.

2.2.1 Two-shot mouse brain extraction network

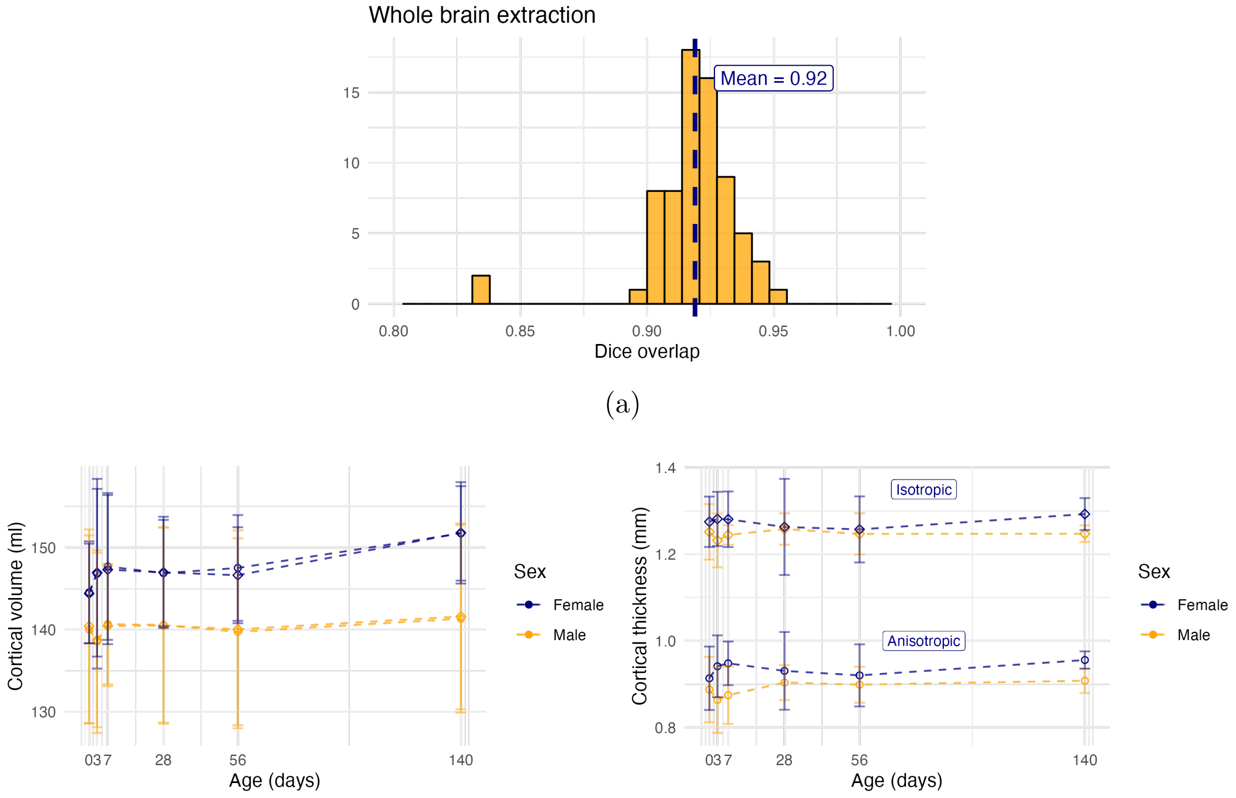
In order to create a generalized mouse brain extraction network, we built whole-head templates from two publicly available datasets. The Center for Animal MRI (CAMRI) dataset[57] from UNC consists of 16 T2-weighted MRI of voxel resolution $0.16 \times 0.16 \times 0.16 mm^3$. The second high-resolution data set[58] comprises 88 specimens each with three spatially aligned canonical views with in-plane resolution of $0.08 \times 0.08mm^2$ with a slice thickness of 0.5 mm. These three orthogonal views were used to reconstruct a single high-resolution volume per subject using a B-spline fitting algorithm developed in ANTsX.[47] From these two datasets, two symmetric isotropic ANTsX templates[44] were generated having different defacing aesthetics analogous to the publicly available ANTsX human brain templates used in previous research.[55] Bias field simulation, intensity histogram warping, noise simulation, random translation and warping, and random anisotropic resampling in the three canonical directions were used for data augmentation in creating a T2-weighted brain extraction network.

2.2.2 Single-shot mouse brain parcellation network

To create the network for generating a brain parcellation consistent with cortical thickness estimation, we used the AllenCCFv3 and the associated `allensdk` Python library. Using `allensdk`, a gross parcellation labeling was generated from the fine Allen CCFv3 labeling which includes the cerebral cortex, cerebral nuclei, brain stem, cerebellum, main olfactory bulb, and hippocampal formation. This labeling was mapped to the P56 component of the DevCCF. Both the T2-w P56 DevCCF and labelings, in conjunction with the data augmentation described previously for brain extraction, was used to create a brain parcellation network.

2.2.3 Evaluation

For evaluation, we used an additional publicly available dataset[59] which is completely independent from the data used in training the brain extraction and parcellation networks. Data includes 12 specimens each imaged at seven time points (Day 0, Day 3, Week 1, Week 4, Week 8, Week 20) with available brain masks. In-plane resolution is $0.1 \times 0.1mm^2$ with a slice thickness of $0.5mm$. Since the training data is isotropic and data augmentation includes downsampling in the canonical directions, each of the two networks learns mouse brain-specific interpolation such that one can perform prediction on thick-sliced images, such as these evaluation data, and return isotropic probability and thickness maps (a choice avail-



(a)

(b)

(c)

Figure 7: Evaluation of the ANTsX mouse brain extraction, parcellation, and cortical thickness pipeline on an independent dataset consisting of 12 specimens × 7 time points = 84 total images. (a) Dice overlap comparisons with the provided brain masks provide generally good agreement with the brain extraction network. (b) Cortical volume measurements show similar average quantities over the developmental trajectory between the original anisotropic data and interpolated isotropic data. These results contrast with the cortical thickness measurements which show that cortical thickness estimation in anisotropic space severely underestimates the actual values.

able to the user). Figure 7 summarizes the results of the evaluation and comparison between isotropic and anisotropic cortical measurements in male and female specimens.

Discussion

The ANTsX ecosystem is a powerful framework that has demonstrated applicability to multiple species and organ systems, including the mouse brain. This is further evidenced by the many other software packages that use various ANTsX components in their own mouse-specific workflows. In and of itself, the extensive functionality of ANTsX makes it possible to create complete processing pipelines without requiring the integration of multiple packages. These open-source components not only perform well but are available across multiple platforms which facilitates the construction of tailored pipelines for individual study solutions. These components are also supported by years of development not only by the ANTsX development team but by the larger ITK community.

In the case of the development of the DevCCF, ANTsX was crucial in providing necessary functionality for yielding high quality output. For the generation of the individual developmental stage multi-modal, symmetric templates, ANTsX is unique amongst image analysis software packages in providing existing solutions for template generation which have been thoroughly vetted, including being used in several studies over the years, and which continue to be under active refinement. At its core, computationally efficient and quality template generation requires the use of precision pairwise image mapping functionality which, historically, is at the origins of the ANTsX ecosystem. And these mapping capabilities extend beyond template generation to the mapping of other image data (e.g., gene expression maps) to a selected template for providing further insight into the mouse brain.

With respect to the DevCCF, despite the significant expansion of available developmental age templates beyond what existed previously, there are still temporal gaps in the DevCCF which can be potentially sampled by future research efforts. However, pioneering work involving time-varying diffeomorphic transformations allow us to continuously situate the existing templates within a velocity flow model. This allows one to determine the diffeomorphic transformation from any one temporal location to any other temporal location within the time span defined by the temporal limits of the DevCCF. This functionality is built on multiple components from the Insight Segmentation and Registration Toolkit including the B-spline scattered data approximation technique for field regularization and velocity field integration. This velocity field model permits intra-template comparison and the construction of virtual templates where a template can be estimated at any continuous time point within the temporal domain. This novel application can potentially enhance our understanding of intermediate developmental stages.

We also presented a mouse brain pipeline for brain extraction, parcellation, and cortical thickness using single-shot and two-shot learning with data augmentation. This approach attempts to circumvent (or at least minimize) the typical requirement of large training datasets as with the human ANTsX pipeline analog. However, even given our initial success on independent data, we fully anticipate that refinements will be necessary. In fact, a current parallel study with a separate collaborator using private data yielded three brain extraction failures (out of 89 specimens). Given that the ANTsX toolkit is a dynamic effort undergoing continual improvement, we manually correct such cases and use them for future training and refinement of network weights as we have done for our human-based networks. Generally, these approaches provide a way to bootstrap training data for manual refinement and future generation of more accurate deep learning networks in the absence of corresponding non deep learning-based tools.

Methods

The following methods are all available as part of the ANTsX ecosystem with analogous elements existing in both ANTsR (ANTs in R) and ANTsPy (ANTs in Python) with and ANTs/ITK C++ core. However, most of the development for the work described below was performed using ANTsPy. For equivalent calls in ANTsR, please see the ANTsX tutorial at https://tinyurl.com/antsxtutorial.

Preprocessing: bias field correction and denoising

Bias field correction and image denoising are standard preprocessing steps in improving overall image quality in mouse brain images. The bias field, a gradual spatial intensity variation in images, can arise from various sources such as magnetic field inhomogeneity or acquisition artifacts, leading to distortions that can compromise the quality of brain images. Correcting for bias fields ensures a more uniform and consistent representation of brain structures, enabling more accurate quantitative analysis. Additionally, brain images are often susceptible to various forms of noise, which can obscure subtle features and affect the precision of measurements. Denoising techniques help mitigate the impact of noise, enhancing the signal-to-noise ratio and improving the overall image quality. The well-known N4 bias field correction algorithm[25] has its origins in the ANTs toolkit which was implemented and introduced into the ITK toolkit, i.e. `ants.n4_bias_field_correction(...)`. Similarly, ANTsX contains an implementation of a well-performing patch-based denoising technique[46] and is also available as an image filter to the ITK community, `ants.denoise_image(...)`.

Image registration

The ANTs registration toolkit is a complex framework permitting highly tailored solutions to pairwise image registration scenarios.[60] It includes innovative transformation models for biological modeling[40,53] and has proven capable of excellent performance.[41,61] Various parameter sets targeting specific applications have been packaged with the different

ANTsX platforms, specifically ANTs, ANTsPy, and ANTsR.[26] In ANTsPy, the function `ants.registration(...)` is used to register a pair of images or a pair of image sets where `type_of_transform` is a user-specified option that invokes a specific parameter set. For example `type_of_transform='antsRegistrationSyNQuick[s]'` encapsulates an oft-used parameter set for quick registration whereas `type_of_transform='antsRegistrationSyN[s]'` is a more detailed alternative. Transforming images using the derived transforms is performed via the `ants.apply_transforms(...)` function.

Initially, linear optimization is initialized with center of (intensity) mass alignment typically followed by optimization of both rigid and affine transforms using the mutual information similarity metric. This is followed by diffeomorphic deformable alignment using symmetric normalization (SyN) with Gaussian[40] or B-spline regularization[53] where the forward transform is invertible and differentiable. The similarity metric employed at this latter stage is typically either neighborhood cross-correlation or mutual information. Note that these parameter sets are robust to input image type (i.e., LSFM, Nissl staining, and the various MRI modalities) and are adaptable to mousing image geometry scaling. Further details can be found in the various documentation sources for these ANTsX packages.

Template generation

ANTsX provides functionality for constructing templates from a set (or multi-modal sets) of input images as originally described[44] and recently used to create the DevCCF templates.[15] An initial template estimate is constructed from an existing subject image or a voxelwise average derived from a rigid pre-alignment of the image population. Pairwise registration between each subject and the current template estimate is performed using the Symmetric Normalization (SyN) algorithm.[40] The template estimate is updated by warping all subjects to the space of the template, performing a voxelwise average, and then performing a "shape update" of this latter image by warping it by the average inverse deformation, thus yielding a mean image of the population in terms of both intensity and shape. The corresponding ANTsPy function is `ants.build_template(...)`.

Continuous developmental velocity flow transformation model

Given multiple, linearly or non-linearly ordered point sets where individual points across are in one-to-one correspondence, we developed an approach for generating a velocity flow transformation model to describe a time-varying diffeomorphic mapping as a variant of the inexact landmark matching solution. Integration of the resulting velocity field can then be used to describe the displacement between any two time points within this time-parameterized domain. Regularization of the sparse correspondence between point sets is performed using a generalized B-spline scattered data approximation technique,[47] also developed by the ANTsX developers and contributed to ITK.

To apply this methodology to the developmental templates,[15] we coalesced the manual annotations of the developmental templates into 26 common anatomical regions (13 per hemisphere). We then used these regions to generate invertible transformations between successive time points. Specifically each label was used to create a pair of single region images resulting in 26 pairs of "source" and "target" images. The multiple image pairs were used to iteratively estimate a diffeomorphic pairwise transform. Given the seven atlases E11.5, E13.5, E15.5, E18.5, P4, P14, and P56, this resulted in 6 sets of transforms between successive time points. Given the relative sizes between atlases, on the order of 10^6 points were randomly sampled labelwise in the P56 template space and propagated to each successive atlas providing the point sets for constructing the velocity flow model. Approximately 125 iterations resulted in a steady convergence based on the average Euclidean norm between transformed point sets. Ten integration points were used and point sets were distributed along the temporal dimension using a log transform for a more evenly spaced sampling. For additional information see the help menu for the ANTsPy function `ants.fit_time_varying_transform_to_point_sets(...)`.

ANTsXNet mouse brain applications

General notes regarding deep learning training.

All network-based approaches described below were implemented and organized in the

ANTsXNet libraries comprising Python (ANTsPyNet) and R (ANTsRNet) analogs using the Keras/Tensorflow libraries available as open-source in ANTsX GitHub repositories. For the various applications, both share the identically trained weights for mutual reproducibility. For all GPU training, we used Python scripts for creating custom batch generators. As such batch generators tend to be application-specific, we store them in a separate GitHub repository for public availability (https://github.com/ntustison/ANTsXNetTraining). In terms of GPU hardware, all training was done on a DGX (GPUs: 4X Tesla V100, system memory: 256 GB LRDIMM DDR4).

Data augmentation is crucial for generalizability and accuracy of the trained networks. Intensity-based data augmentation consisted of randomly added noise (i.e., Gaussian, shot, salt-and-pepper), simulated bias fields based on N4 bias field modeling, and histogram warping for mimicking well-known MRI intensity nonlinearities.[26,62] These augmentation techniques are available in ANTsXNet (only ANTsPyNet versions are listed with ANTsRNet versions available) and include:

- image noise: `ants.add_noise_to_image(...)`,
- simulated bias field: `antspynet.simulate_bias_field(...)`, and
- nonlinear intensity warping: `antspynet.histogram_warp_image_intensities(...)`.

Shape-based data augmentation used both random linear and nonlinear deformations in addition to anisotropic resampling in the three canonical orientations to mimic frequently used acquisition protocols for mice brains:

- random spatial warping: `antspynet.randomly_transform_image_data(...)` and
- anisotropic resampling: `ants.resample_image(...)`.

Brain extraction.

Similar to human neuroimage processing, brain extraction is a crucial preprocessing step for accurate brain mapping. Within ANTsXNet, we have created several deep learning networks for brain extraction for several image modalities (e.g., T1, FLAIR, fractional anisotropy). Similarly, for the developmental brain atlas work[15] we developed similar functionality for mouse brains of different modalities and developmental age. All networks use a conventional U-net architecture.[63] Whereas T2-weighted brain extraction is volumetric-based for both isotropic and anisotropic data, coronal and sagittal networks are available for both E13.5 and E15.5 data. In ANTsPyNet, this functionality is available in the program `antspynet.mouse_brain_extraction(...)`. Even when physical brain extraction is performed prior to image acquisition, artifacts, such as bubbles or debris, can complicate subsequent processing. Similar to the brain extraction networks, a 2-D U-net architecture[63] was created to separate the background and foreground. For the two-shot T2-weighted brain extraction network, two brain templates were generated along with their masks. Although both template/mask pairs are available in the GitHub repository associated with this work, the synthesize volumetric B-spline T2-weighted pair is available within ANTsXNet through the calls:

- template: `antspynet.get_antsxnet_data("bsplineT2MouseTemplate")` and
- mask: `antspynet.get_antsxnet_data("bsplineT2MouseTemplateBrainMask")`.

Brain parcellation.

The T2-weighted brain parcellation network is also based on a 3-D U-net architecture and the T2-w DevCCF P56 template component with extensive data augmentation, as described previously. Intensity differences between the template and any brain extracted input image are minimized through the use of the rank intensity transform (`ants.rank_intensity(...)`). Shape differences are reduced by the additional preprocessing step of warping the brain extracted input image to the template. Additional input channels include the prior probability images created from the template parcellation. These images are also available through the ANTsXNet interface:

- template: `antspynet.get_antsxnet_data("DevCCF_P56_MRI-T2_50um")` and
- parcellation: `antspynet.get_antsxnet_data("DevCCF_P56_MRI-T2_50um_BrainParcellationNic`

Miscellaneous networks: Super-resolution, cerebellum, and hemispherical masking.

To further enhance the data prior to designing mapping protocols, additional networks were created. A well-performing deep back projection network[64] was ported to ANTsXNet and expanded to 3-D for various super-resolution applications,[65] including mouse data. Finally, features of anatomical significance, namely the cerebellum and hemispherical midline were captured in these data using deep learning networks.

Intra-slice image registration with missing slice imputation

Volumetric gene expression slice data was collated into 3-D volumes. Prior to mapping this volume to the corresponding structural data and, potentially, to the appropriate template, alignment was improved using deformable registration on contiguous slices. However, one of the complications associated with these image data was the unknown number of missing slices, the number of consecutive missing slices, and the different locations of these missing slices. To handle this missing data problem, we found that data interpolation using the B-spline approximation algorithm cited earlier[47] (ANTsPy function: `ants.fit_bspline_object_to_scattered_data(...)`). This provided sufficient data interpolation fidelity to perform continuous slicewise registration. Other possible variants that were considered but deemed unnecessary was performing more than one iteration cycling through data interpolation and slicewise alignment. The other possibility was incorporating the super-resolution technique described earlier. But again, our data did not require these additional steps.

Synthesizing isotropic image volumes from orthogonal views

Similar to the isotropic , the multi-plane, high resolution data[58] was used to create single isotropic volumes using the B-spline fitting algorithm.[47] This algorithm is encapsulated in `ants.fit_bspline_object_to_scattered_data(...)` where the input is the set of voxel intensity values and associated physical location. Since each point can be assigned a confidence weight, we use the the normalized gradient value to more heavily weight edge regions.

Visualization

To complement the well-known visualization capabilities of R and Python, e.g., ggplot2 and matplotlib, respectively, image-specific visualization capabilities are available in the `ants.plot(...)` (Python) and `plot.antsImage(...)` (R). These are capable of illustrating multiple slices in different orientations with both other image overlays as well as label images.

Data availability. All data and doftware used in this work are publicly available. The DevCCF atlas is available at https://kimlab.io/brain-map/DevCCF/. ANTsPy, ANTsR, ANTsPyNet, and ANTsRNet are available through GitHub at the ANTsX Ecosystem (https://github.com/ANTsX). Training scripts for all deep learning functionality in ANTsXNet can also be found on GitHub (https://github.com/ntustison/ANTsXNetTraining). A GitHub repository specific to the work discussed in the manuscript was created and is available at https://github.com/ntustison/ANTsXMouseBrainMapping.

References

1. Keller, P. J. & Ahrens, M. B. Visualizing whole-brain activity and development at the single-cell level using light-sheet microscopy. *Neuron* **85**, 462–83 (2015).

2. La Manno, G. *et al.* Molecular architecture of the developing mouse brain. *Nature* **596**, 92–96 (2021).

3. Wen, L. *et al.* Single-cell technologies: From research to application. *Innovation (Camb)* **3**, 100342 (2022).

4. Oh, S. W. *et al.* A mesoscale connectome of the mouse brain. *Nature* **508**, 207–14 (2014).

5. Gong, H. *et al.* Continuously tracing brain-wide long-distance axonal projections in mice at a one-micron voxel resolution. *Neuroimage* **74**, 87–98 (2013).

6. Li, A. *et al.* Micro-optical sectioning tomography to obtain a high-resolution atlas of the mouse brain. *Science* **330**, 1404–8 (2010).

7. Ueda, H. R. *et al.* Tissue clearing and its applications in neuroscience. *Nat Rev Neurosci* **21**, 61–79 (2020).

8. Ståhl, P. L. *et al.* Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science* **353**, 78–82 (2016).

9. Burgess, D. J. Spatial transcriptomics coming of age. *Nat Rev Genet* **20**, 317 (2019).

10. MacKenzie-Graham, A. *et al.* A multimodal, multidimensional atlas of the C57BL/6J mouse brain. *J Anat* **204**, 93–102 (2004).

11. Mackenzie-Graham, A. J. *et al.* Multimodal, multidimensional models of mouse brain. *Epilepsia* **48 Suppl 4**, 75–81 (2007).

12. Dong, H. W. *Allen reference atlas. A digital color brain atlas of the C57BL/6J male mouse.* (John Wiley; Sons, 2008).

13. Wang, Q. *et al.* The allen mouse brain common coordinate framework: A 3D reference atlas. *Cell* **181**, 936–953.e20 (2020).

14. Johnson, G. A. *et al.* Waxholm space: An image-based reference for coordinating mouse brain research. *Neuroimage* **53**, 365–72 (2010).

15. Kronman, F. A. *et al.* Developmental mouse brain common coordinate framework. *bioRxiv* (2023) doi:10.1101/2023.09.14.557789.

16. Oguz, I., Zhang, H., Rumple, A. & Sonka, M. RATS: Rapid automatic tissue segmentation in rodent brain MRI. *J Neurosci Methods* **221**, 175–82 (2014).

17. Sawiak, S. J., Picq, J.-L. & Dhenain, M. Voxel-based morphometry analyses of in vivo MRI in the aging mouse lemur primate. *Front Aging Neurosci* **6**, 82 (2014).

18. Ashburner, J. SPM: A history. *Neuroimage* **62**, 791–800 (2012).

19. Modat, M. *et al.* Fast free-form deformation using graphics processing units. *Comput Methods Programs Biomed* **98**, 278–84 (2010).

20. Tyson, A. L. *et al.* Accurate determination of marker location within whole-brain microscopy images. *Sci Rep* **12**, 867 (2022).

21. Pallast, N. *et al.* Processing pipeline for atlas-based imaging data analysis of structural and functional mouse brain MRI (AIDAmri). *Front Neuroinform* **13**, 42 (2019).

22. Jenkinson, M., Beckmann, C. F., Behrens, T. E. J., Woolrich, M. W. & Smith, S. M. FSL. *Neuroimage* **62**, 782–90 (2012).

23. Yeh, F.-C., Wedeen, V. J. & Tseng, W.-Y. I. Generalized q-sampling imaging. *IEEE Trans Med Imaging* **29**, 1626–35 (2010).

24. Jorge Cardoso, M. *et al.* STEPS: Similarity and truth estimation for propagated segmentations and its application to hippocampal segmentation and brain parcelation. *Med Image Anal* **17**, 671–84 (2013).

25. Tustison, N. J. *et al.* N4ITK: Improved N3 bias correction. *IEEE Trans Med Imaging* **29**, 1310–20 (2010).

26. Tustison, N. J. *et al.* The ANTsX ecosystem for quantitative biological and medical imaging. *Sci Rep* **11**, 9068 (2021).

27. Goubran, M. *et al.* Multimodal image registration and connectivity analysis for integration of connectomic data from microscopy to MRI. *Nat Commun* **10**, 5504 (2019).

28. Celestine, M., Nadkarni, N. A., Garin, C. M., Bougacha, S. & Dhenain, M. Sammba-MRI: A library for processing SmAll-MaMmal BrAin MRI data in python. *Front Neuroinform* **14**, 24 (2020).

29. Ioanas, H.-I., Marks, M., Zerbi, V., Yanik, M. F. & Rudin, M. An optimized registration workflow and standard geometric space for small animal brain imaging. *Neuroimage* **241**, 118386 (2021).

30. Cox, R. W. AFNI: What a long strange trip it's been. *Neuroimage* **62**, 743–7 (2012).

31. Ni, H. *et al.* A robust image registration interface for large volume brain atlas. *Sci Rep* **10**, 2139 (2020).

32. Jin, M. *et al.* SMART: An open-source extension of WholeBrain for intact mouse brain registration and segmentation. *eNeuro* **9**, (2022).

33. Fürth, D. *et al.* An interactive framework for whole-brain maps at cellular resolution. *Nat Neurosci* **21**, 139–149 (2018).

34. Negwer, M. *et al.* FriendlyClearMap: An optimized toolkit for mouse brain mapping and analysis. *Gigascience* **12**, (2022).

35. Klein, S., Staring, M., Murphy, K., Viergever, M. A. & Pluim, J. P. W. Elastix: A toolbox for intensity-based medical image registration. *IEEE Trans Med Imaging* **29**, 196–205 (2010).

36. Carey, H. *et al.* DeepSlice: Rapid fully automatic registration of mouse brain imaging to a volumetric atlas. *Nat Commun* **14**, 5884 (2023).

37. Bajcsy, R. & Broit, C. Matching of deformed images. in *Sixth International Conference on Pattern Recognition (ICPR'82)* 351–353 (1982).

38. Bajcsy, R. & Kovacic, S. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing* **46**, 1–21 (1989).

39. Gee, J., Sundaram, T., Hasegawa, I., Uematsu, H. & Hatabu, H. Characterization of regional pulmonary mechanics from serial magnetic resonance imaging data. *Acad Radiol* **10**, 1147–52 (2003).

40. Avants, B. B., Epstein, C. L., Grossman, M. & Gee, J. C. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Med Image Anal* **12**, 26–41 (2008).

41. Klein, A. *et al.* Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration. *Neuroimage* **46**, 786–802 (2009).

42. Murphy, K. *et al.* Evaluation of registration methods on thoracic CT: The EMPIRE10 challenge. *IEEE Trans Med Imaging* **30**, 1901–20 (2011).

43. Baheti, B. *et al.* The brain tumor sequence registration challenge: Establishing correspondence between pre-operative and follow-up MRI scans of diffuse glioma patients. (2021).

44. Avants, B. B. *et al.* The optimal template effect in hippocampus studies of diseased populations. *Neuroimage* **49**, 2457–66 (2010).

45. Avants, B. B., Tustison, N. J., Wu, J., Cook, P. A. & Gee, J. C. An open source multivariate framework for n-tissue segmentation with evaluation on public data. *Neuroinformatics* **9**, 381–400 (2011).

46. Manjón, J. V., Coupé, P., Martí-Bonmatí, L., Collins, D. L. & Robles, M. Adaptive non-local means denoising of MR images with spatially varying noise levels. *J Magn Reson Imaging* **31**, 192–203 (2010).

47. Tustison, N. J. & Amini, A. A. Biventricular myocardial strains via nonrigid registration of anatomical NURBS model [corrected]. *IEEE Trans Med Imaging* **25**, 94–112 (2006).

48. Wang, H. *et al.* Multi-atlas segmentation with joint label fusion. *IEEE Trans Pattern Anal Mach Intell* **35**, 611–23 (2013).

49. Tustison, N. J. *et al.* Optimal symmetric multimodal templates and concatenated random forests for supervised brain tumor segmentation (simplified) with *ANTsR*. *Neuroinformatics* (2014) doi:10.1007/s12021-014-9245-2.

50. Tustison, N. J., Yang, Y. & Salerno, M. Advanced normalization tools for cardiac motion correction. in *Statistical atlases and computational models of the heart - imaging and modelling challenges* (eds. Camara, O. et al.) vol. 8896 3–12 (Springer International Publishing, 2015).

51. McCormick, M., Liu, X., Jomier, J., Marion, C. & Ibanez, L. ITK: Enabling reproducible research and open science. *Front Neuroinform* **8**, 13 (2014).

52. Beg, M. F., Miller, M. I., Trouvé, A. & Younes, L. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision* **61**, 139–157 (2005).

53. Tustison, N. J. & Avants, B. B. Explicit B-spline regularization in diffeomorphic image registration. *Front Neuroinform* **7**, 39 (2013).

54. Das, S. R., Avants, B. B., Grossman, M. & Gee, J. C. Registration based cortical thickness measurement. *Neuroimage* **45**, 867–79 (2009).

55. Tustison, N. J. *et al.* Large-scale evaluation of ANTs and FreeSurfer cortical thickness measurements. *Neuroimage* **99**, 166–79 (2014).

56. Tustison, N. J. *et al.* Longitudinal mapping of cortical thickness measurements: An Alzheimer's Disease Neuroimaging Initiative-based evaluation study. *J Alzheimers Dis* (2019) doi:10.3233/JAD-190283.

57. Hsu, L.-M. *et al.* CAMRI mouse brain MRI data.

58. Reshetnikov, V. *et al.* High-resolution MRI data of brain C57BL/6 and BTBR mice in three different anatomical views.

59. Rahman, N., Xu, K., Budde, M. D., Brown, A. & Baron, C. A. A longitudinal microstructural MRI dataset in healthy C57Bl/6 mice at 9.4 tesla. *Sci Data* **10**, 94 (2023).

60. Avants, B. B. *et al.* The Insight ToolKit image registration framework. *Front Neuroinform* **8**, 44 (2014).

61. Avants, B. B. *et al.* A reproducible evaluation of ANTs similarity metric performance in brain image registration. *Neuroimage* **54**, 2033–44 (2011).

62. Nyúl, L. G., Udupa, J. K. & Zhang, X. New variants of a method of MRI scale standardization. *IEEE Trans Med Imaging* **19**, 143–50 (2000).

63. Falk, T. *et al.* U-net: Deep learning for cell counting, detection, and morphometry. *Nat Methods* **16**, 67–70 (2019).

64. Haris, M., Shakhnarovich, G. & Ukita, N. Deep back-projection networks for super-resolution. in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 1664–1673 (2018). doi:10.1109/CVPR.2018.00179.

65. Avants, B. B. *et al.* Concurrent 3D super resolution on intensity and segmentation maps improves detection of structural effects in neurodegenerative disease. *medRxiv* (2023) doi:10.1101/2023.02.02.23285376.