

1

2 **The ANTsX Ecosystem for Spatiotemporal 3 Mapping of the Developmental Mouse Brain 4 Common Coordinate Framework**

5 Nicholas J. Tustison¹, Min Chen², Fae N. Kronman³, Jeffrey T. Duda², Clare Gamlin⁴,
6 Lydia Ng⁴, Yongsoo Kim³, and James C. Gee²

7 ¹Department of Radiology and Medical Imaging, University of Virginia, Charlottesville, VA

8 ²Department of Radiology, University of Pennsylvania, Philadelphia, PA

9 ³Department of Neural and Behavioral Sciences, Penn State University, Hershey, PA

10 ⁴Allen Institute for Brain Science, Seattle, WA

11

12 Corresponding author:

13 Nicholas J. Tustison, DSc

14 Department of Radiology and Medical Imaging

15 University of Virginia

16 ntustison@virginia.edu

¹⁷ **Abstract**

¹⁸ Precision mapping techniques coupled with high resolution image acquisition of the mouse
¹⁹ brain permit the study of the spatial organization of gene activity and their mutual in-
²⁰ teraction for a comprehensive view of salient structural/functional relationships. Such re-
²¹ search is facilitated by standardized anatomical coordinate systems, such as the well-known
²² Allen Common Coordinate Framework , and the ability to spatially map to such standard-
²³ ized spaces. The Advanced Normalization Tools Ecosystem (ANTsX) is a comprehensive
²⁴ open-source software image analysis toolkit, which includes template building and mapping
²⁵ functionality, with applicability to multiple organ systems, modalities, and animal species.
²⁶ Herein, we illustrate the utility of ANTsX for generating precision spatial mappings of the
²⁷ mouse brain using the recently proposed Developmental Common Coordinate Framework.
²⁸ These longitudinal, discretely sampled atlases are used to generate a velocity flow-based
²⁹ mapping spanning the spatiotemporal domain of the developmental trajectory with future
³⁰ work accommodating the introduction of additional developmental time points.

³¹ **Introduction**

³² Over the past two decades there have been significant advancements in mesoscopic anal-
³³ ysis of the mouse brain. It is now possible to track single cell neurons in mouse brains,¹
³⁴ observe whole brain developmental changes on a cellular level,² associate brain regions and
³⁵ tissues with their genetic composition,³ and locally characterize neural connectivity.⁴ Much
³⁶ of this scientific achievement has been made possible due to breakthroughs in high resolution
³⁷ imaging techniques that permit submicron, 3-D imaging of whole mouse brains. Associated
³⁸ research techniques such as micro-optical sectioning tomography,⁶ tissue clearing,^{1,7} spatial
³⁹ transcriptomics⁹ are all well-utilized in the course of scientific investigations of mesoscale
⁴⁰ relationships in the mouse brain.

⁴¹ An important component of this research is the ability to map the various image data to
⁴² anatomical reference frames¹¹ for inferring spatial relationships between structures, cells,
⁴³ and genetics. This has motivated the development of detailed structural image atlases of the
⁴⁴ mouse brain. Notable examples include the Allen Brain Atlas and Coordinate Frameworks¹³
⁴⁵ and the Waxholm Space.¹⁴ Despite the significance of these contributions, challenges still
⁴⁶ exist in large part due to the wide heterogeneity in associated study-specific image data. For
⁴⁷ example, variance in the acquisition methods can introduce artifacts such as tissue distor-
⁴⁸ tion, holes, bubbles, folding, tears, and missing slices. These severely complicate assumed
⁴⁹ correspondence for conventional spatial mapping approaches.

⁵⁰ To address such challenges, several software packages have been developed over the years
⁵¹ comprising solutions of varying comprehensibility, sophistication, and availability. An early
⁵² contribution to the community was the Rapid Automatic Tissue Segmentation (RATS)
⁵³ package¹⁵ for brain extraction. Of the publicly available packages, most, if not all have well-
⁵⁴ established package dependencies originally developed on human brain data. SPMMouse,¹⁶
⁵⁵ for example, is based on the well-known Statistical Parametric Mapping (SPM) software
⁵⁶ package.¹⁷ The automated mouse atlas propagation (aMAP) tool is largely a front-end for
⁵⁷ the NiftyReg image registration package¹⁸ applied to mouse data which is currently avail-
⁵⁸ able as a Python module.¹⁹ NiftyReg is also used by the Atlas-based Imaging Data Anal-
⁵⁹ ysis (AIDA) MRI pipeline²⁰ as well as the Multi Atlas Segmentation and Morphometric

60 Analysis Toolkit (MASMAT). Whereas the former also incorporates the FMRIB Software
61 Library (FSL)²¹ for brain extraction and DSISTudio²² for DTI processing, the latter uses
62 NiftySeg and multi-consensus labeling tools²³ for brain extraction and parcellation. In ad-
63 dition, MASMAT incorporates N4 bias field correction²⁴ from the Advanced Normalization
64 Tools Ecosystem (ANTsX)²⁵ as do the packages Multi-modal Image Registration And Con-
65 nectivity anaLysis (MIRACL),²⁶ Saamba-MRI,²⁷ and Small Animal Magnetic Resonance
66 Imaging (SAMRI).²⁸ However, whereas Saamba-MRI uses AFNI²⁹ for image registration;
67 MIRACL, SAMRI, and BrainsMapi³⁰ all use ANTsX registration tools. Other packages
68 use landmark-based approaches to image registration including SMART—³¹an R package
69 for semi-automated landmark-based registration and segmentation of mouse brain based
70 on WholeBrain.³² FriendlyClearMap³³ uses the landmark-based registration functionality of
71 Elastix.³⁴ Finally, the widespread adoption of deep learning techniques has also influenced
72 development in mouse brain imaging methodologies. For example, if tissue deformations
73 are not considered problematic for a particular dataset, DeepSlice can be used to determine
74 affine mappings³⁵ with the optimal computational efficiency associated with neural networks.

75 The ANTsX Ecosystem

76 As noted previously, many of the existing approaches for processing of mouse brain image
77 data use ANTsX tools for core processing steps in various workflows, particularly its pair-
78 wise, intensity-based image registration tools and bias field correction. Historically, ANTsX
79 development is originally based on fundamental approaches to image mapping,^{36–38} partic-
80 ularly in the human brain, which has resulted in core contributions to the field such as
81 the well-known and highly-vetted Symmetric Normalization (SyN) algorithm.³⁹ Since its
82 development, various independent platforms have been used to evaluate ANTsX image reg-
83 istration capabilities in the context of different application foci which include multi-site brain
84 MRI data,⁴⁰ pulmonary CT data,⁴¹ and most recently multi-modal brain registration in the
85 presence of tumors.⁴²

86 Apart from its registration capabilities, ANTsX comprises additional functionality such as
87 template generation, general data approximation, and deep learning networks specifically

Table 1: Sampling of ANTsX functionality

<i>ANTsPy: Preprocessing</i>	
bias field correction	<code>n4_bias_field_correction(...)</code>
image denoising	<code>denoise_image(...)</code>
<i>ANTsPy: Registration</i>	
image registration	<code>registration(...)</code>
template generation	<code>build_template(...)</code>
landmark registration	<code>fit_transform_to_paired_points(...)</code>
time-varying landmark reg.	<code>fit_time_varying_transform_to_point_sets(...)</code>
integrate velocity field	<code>integrate_velocity_field(...)</code>
invert displacement field	<code>invert_displacement_field(...)</code>
<i>ANTsPy: Segmentation</i>	
General segmentation	<code>atropos(...)</code>
Joint label fusion	<code>joint_label_fusion(...)</code>
diffeomorphic thickness	<code>kelly_kapowski(...)</code>
<i>ANTsPy: Miscellaneous</i>	
Regional intensity statistics	<code>label_stats(...)</code>
Regional shape measures	<code>label_geometry_measures(...)</code>
B-spline approximation	<code>fit_bspline_object_to_scattered_data(...)</code>
Visualize images and overlays	<code>plot(...)</code>
<i>ANTsPyNet: Mouse-specific</i>	
brain extraction	<code>mouse_brain_extraction(...modality="t2"...)</code> <code>mouse_brain_extraction(...modality="ex5"...)</code>
foreground extraction	<code>mouse_histology_brain_mask(...)</code>
midline segmentation	<code>mouse_histology_hemispherical_coronal_mask(...)</code>
cerebellum segmentation	<code>mouse_histology_cerebellum_mask(...)</code>
super resolution	<code>mouse_histology_super_resolution(...)</code>

ANTsX provides state-of-the-art open-science functionality for processing 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>).

88 trained for mouse data (see Table 1). The collective use of the toolkit has demonstrated
 89 superb performance in multiple application areas (e.g., consensus labeling,⁴³ brain tumor
 90 segmentation,⁴⁴ and cardiac motion estimation⁴⁵). Importantly, ANTs is built on the In-
 91 sight Toolkit (ITK)⁴⁶ deriving benefit from the open-source community of scientists and
 92 programmers and providing a visible, open-source venue for algorithmic contributions.

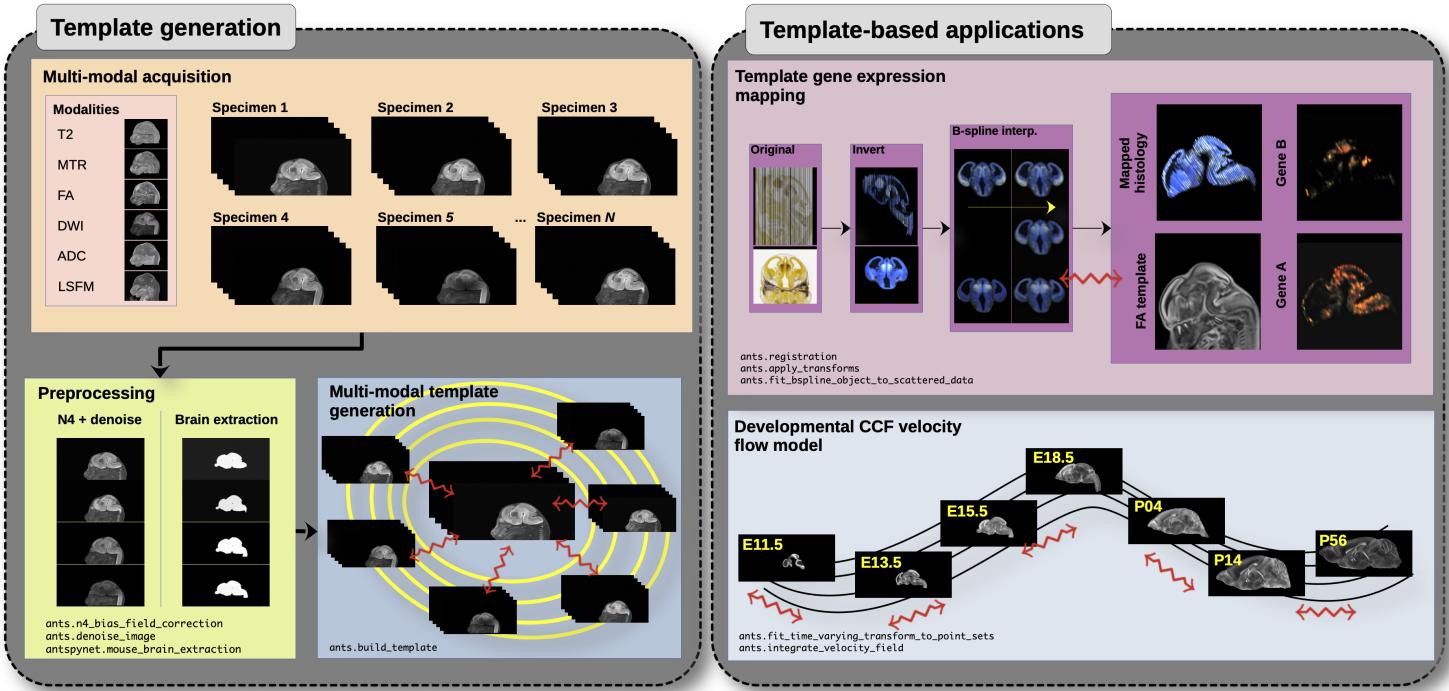


Figure 1: Illustration of a mouse brain template generation workflow and related template-based applications demonstrating the utility of different ANTsX tools. After imaging acquisition of the study population, various preprocessing steps are applied to the imaging data such as bias correction, denoising, and brain extraction as dictated by the needs of the study protocol. In the specific case of the DevCCF, potential applications include gene expression mapping and the generation of the associated velocity flow model for continuous spatiotemporal mapping in the temporal domain spanned by the DevCCF.

93 Recently, the developmental common coordinate framework (DevCCF) was introduced to
 94 the mouse brain research community as a public resource.⁴⁷ These symmetric atlases, com-
 95 prising both multimodal image data and anatomical segmentations defined by developmental
 96 ontology, sample the mouse embryonic days (E) 11.5, E13.5, E15.5, E18.5 and postnatal day
 97 (P) 4, P14, and P56. Modalities include light sheet fluorescence microscopy (LSFM) and at
 98 least four MRI contrasts per developmental stage. Anatomical parcellations are also available

99 for each time point and were generated from ANTsX-based mappings of gene expression and
100 other cell type data. The P56 template was integrated with the Allen CCFv3 to further in-
101 crease the practical utility of the DevCCF. These processes, specifically template generation
102 and multi-modal image mapping, were performed using ANTsX functionality in the presence
103 of previously noted image mapping difficulties (e.g., missing slices, tissue distortion).

104 Given the temporal gaps in the discrete set of developmental atlases with the potential for
105 additional interpolative time points, we discuss the strategy of the current DevCCF tem-
106 plate generation⁴⁷ and provide additional information for the interested reader. Related, we
107 also provide an open-source framework, through ANTsX, for inferring correspondence within
108 the temporally continuous domain sampled by the existing set of embryonic and postnatal
109 atlases of the DevCCF. Although alternative approaches are possible for interpolating be-
110 tween time points, this recently developed ANTsX functionality permits the generation of
111 a diffeomorphic velocity flow transformation model,⁴⁸ influenced by previous work.⁴⁹ The
112 resulting time-parameterized velocity field spans the stages of the DevCCF where mappings
113 between any two continuous time points within the span bounded by the E11.5 and P56
114 atlases is determined by integration of the optimized velocity field. This functionality is
115 available through ANTsX (via R and Python ANTsX packages) with a dedicated GitHub
116 repository that contains all data, scripts, and other guidance necessary to both reproduce
117 what is described below and to illustrate how future researchers can incorporate additional
118 atlases into a more densely sampled model in a straightforward manner.

₁₁₉ **Results**

₁₂₀ **Template building**

₁₂₁ Template building using ANTsX tools was first described in the context of hippocampal
₁₂₂ studies.⁵⁰ Multi-modal and symmetrical variants were subsequently described as part
₁₂₃ of a proposed brain tumor segmentation approach based on random forests.⁵¹ Tem-
₁₂₄ plate building capabilities are available in both ANTsPy (`ants.build_template(...)`)
₁₂₅ and ANTsR (`buildTemplate(...)`) as well as part of the core ANTs package (e.g.,
₁₂₆ `antsMultivariateTemplateConstruction.sh`).

₁₂₇ **Data preparation**

₁₂₈ Multi-modal symmetric template construction is performed separately for each develop-
₁₂₉ mental stage. Prior to optimization, preprocessing can include several steps not all of
₁₃₀ which are required but are dependent on the data and the particular requirements of the
₁₃₁ study. For MRI scans, inhomogeneity correction is often necessary and can be performed
₁₃₂ using the ANTsPy function `ants.n4_bias_field_correction(...)` which is a wrapper
₁₃₃ for the N4 algorithm.²⁴ Denoising is another preprocessing step that can potentially im-
₁₃₄ prove template quality results. The ANTsPy function `ants.denoise_image(...)` is an
₁₃₅ implementation of a well-known denoising algorithm.⁵² For a typical image, both of these
₁₃₆ steps takes approximately on the order of a couple minutes. In ANTsX, due to legacy
₁₃₇ code issues, only bias correction is wrapped with template building so one need not per-
₁₃₈ form this step prior to optimization. In addition, brain extraction has demonstrated im-
₁₃₉ proved performance in the context of human brain normalization⁵³ and is similarly used
₁₄₀ in mouse brain registration to maximize alignment. Various approaches within ANTs are
₁₄₁ possible including a template-based approach `antsBrainExtraction.sh` or using deep learn-
₁₄₂ ing `antspynet.mouse_brain_extraction(...)`. Additionally, it is important to ensure a
₁₄₃ standardized orientation, similar to the Dicom standard for human brain imaging. A study
₁₄₄ requirement of template bilateral symmetry is also an important consideration prior to tem-
₁₄₅ plate generation. This can be performed by either flipping all the input images contralaterally

146 such that all input specimens are represented twice or one can generate an initial asymmetric
147 template, flipping it contralaterally, and using the two asymmetric templates in a subsequent
148 template generation call to create a single symmetric template. For multi-modal templates,
149 all the images for a single specimen need to be mutually aligned in the same image space
150 prior to optimization. After selecting the target image space for a particular specimen
151 (e.g., T2-weighted MRI), this can be performed with a rigid transform registration call us-
152 ing `ants.registration(...)`. It should be noted that for most applications, the general
153 heuristic of ≈ 10 randomly sampled specimens is sufficient for a satisfactory template.

154 In the case of the DevCCF, bias correction was employed in generating the multiple stage
155 templates using the shell script `antsMultivariateConstruction.sh`. Brain extraction was
156 applied to the postnatal images. Template symmetrization employed the original and con-
157 tralateral versions of all specimen images.

158 Optimization

159 Template generation is initialized with either a user-provided image or a bootstrapped ini-
160 tialization template constructed from the input data. If the latter is selected, the voxelwise
161 averaged image for each modality is constructed followed by a linear registration of each
162 specimen to this template initialization which refines the estimate. The former option is
163 often used where computational considerations are important. For example, this initial tem-
164 plate can be generated using low resolution input data or only a subset of the input cohort.
165 This higher quality initial estimate can then be further refined using the entire data set at
166 full resolution.

167 Following template initialization, each specimen is registered to the current template es-
168 timate, which can be performed in parallel. After the current round of registrations is
169 complete, a voxelwise average of each modality is performed with optional Laplacian sharp-
170 ening followed by a “shape update” step. This shape update step is used to warp the current
171 estimate of the template so that its shape is closer to the mean shape of the input data.
172 Implementation-wise this is done by averaging each displacement field that points from the
173 template to the affinely warped specimen. This average displacement field is then used to

¹⁷⁴ deform the voxelwise-averaged template. Shape and intensity template convergence typically
¹⁷⁵ occurs in four deformable iterations.

¹⁷⁶ The DevCCF Velocity Flow Model

¹⁷⁷ To continuously interpolate transformations between the different stages of the De-
¹⁷⁸ vCCF atlases, a velocity flow model was constructed using Dev-CCF derived data and
¹⁷⁹ ANTsX functionality recently introduced into both the ANTsR and ANTsPy packages.
¹⁸⁰ Both platforms include a complete suite of functions for determining dense correspon-
¹⁸¹ dence from sparse landmarks based on a variety of transformation models ranging from
¹⁸² standard linear models (i.e., rigid, affine) to deformable diffeomorphic models (e.g,
¹⁸³ symmetric normalization).³⁹ The latter set includes velocity flow models for both the
¹⁸⁴ pairwise scenario (`ants.fit_transform_to_paired_points(...)`) and for multiple
¹⁸⁵ sets (`ants.fit_time_varying_transform_to_point_sets(...)`), as in the case of the
¹⁸⁶ DevCCF. Several self-contained tutorials illustrating usage for these functions are available
¹⁸⁷ at <https://tinyurl.com/antsxtutorial>.

¹⁸⁸ ANTsX, being built on top of ITK, uses an ITK image data structure for the 4-D velocity
¹⁸⁹ field where each voxel contains the x , y , z components of the field at that point. Field
¹⁹⁰ regularization is provided by a B-spline scattered data approximation technique⁴⁹ which
¹⁹¹ permits individual point weighting. Both field regularization and integration of the velocity
¹⁹² field are built on ITK functions contributed from ANTsX development.

¹⁹³ Data preparation

¹⁹⁴ Labeled annotations are available as part of the original DevCCF and reside in the space
¹⁹⁵ of each developmental template which range in resolution from $31.5 - 50\mu\text{m}$. Across all
¹⁹⁶ atlases, the total number of labeled regions exceeds 2500. From these labels, a common set
¹⁹⁷ of 26 labels (13 per hemisphere) across all atlases were used for optimization and evaluation.
¹⁹⁸ These regions are illustrated for the P4 and P14 stages in Figure 2.

¹⁹⁹ Prior to velocity field optimization, all data were rigidly transformed to a common space.

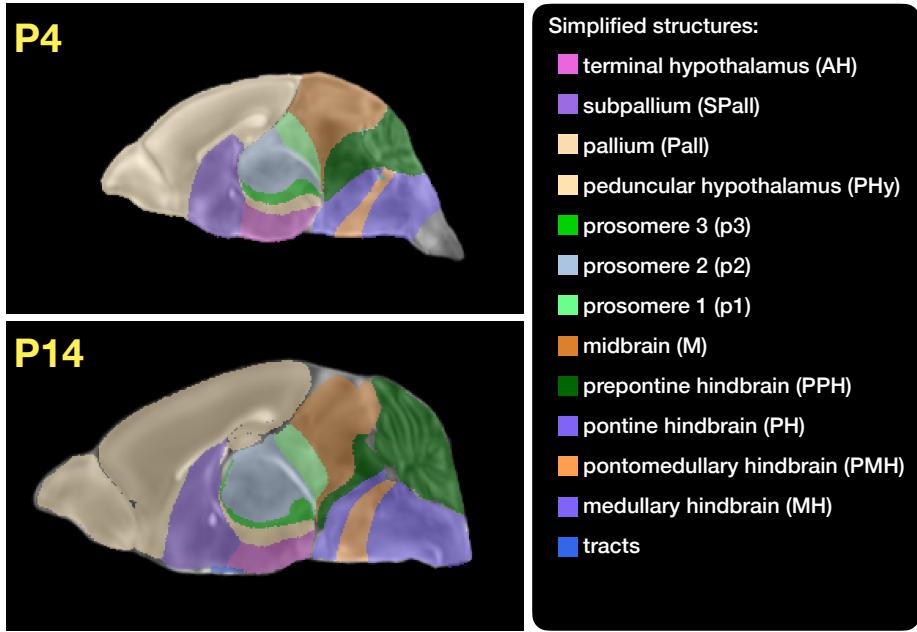


Figure 2: Annotated regions representing common labels across developmental stages which are illustrated for both P4 and P14.

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 regional points during optimization.

214 **Optimization**

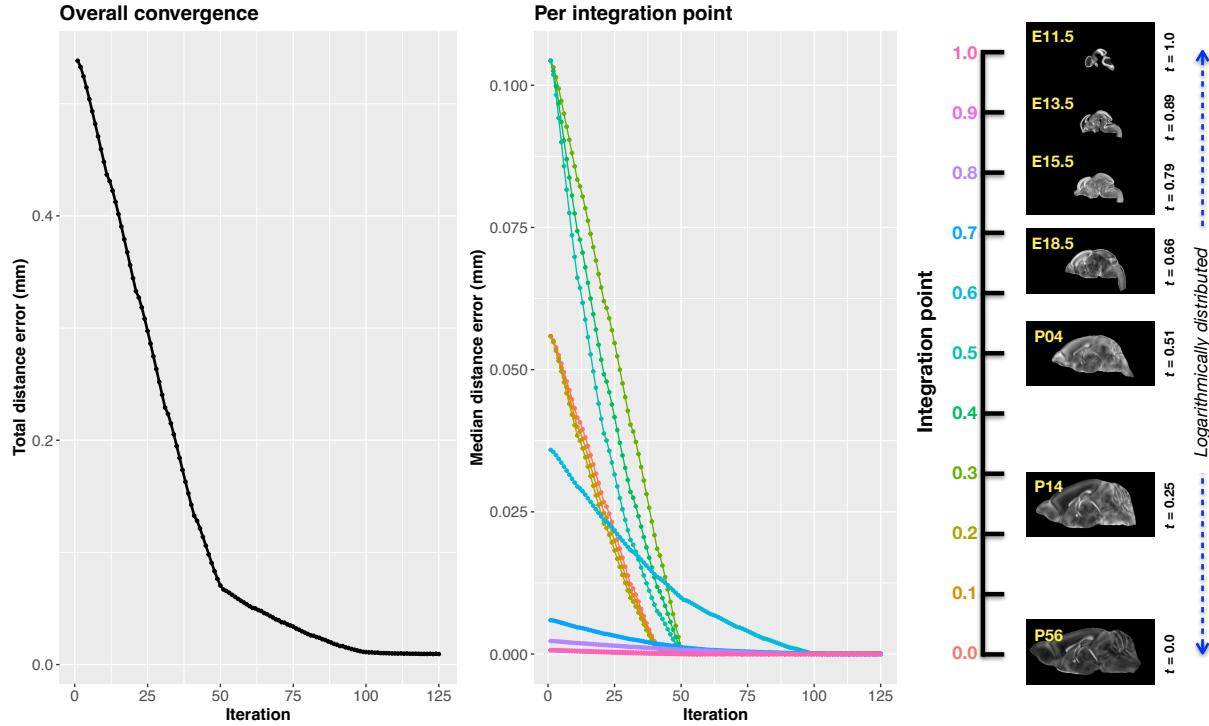


Figure 3: Convergence of the optimization of the velocity field for describing the transformation through the developmental stages from E11.5 through P56.

215 `ants.fit_time_varying_transform_to_point_sets(...)` from the ANTsPy package was
 216 used to optimize the velocity field. Input comprised the seven corresponding point sets and
 217 their associated weight values, the selected number of integration points for the velocity
 218 field ($N = 11$), and the parameters defining the geometry of the spatial dimensions of the
 219 velocity field. Thus, the optimized velocity field described here is of size [256, 182, 360] (50 μm
 220 isotropic) $\times 11$ integration points for a total compressed size of a little over 2 GB. This choice
 221 represented weighing the trade-off between tractability, portability, and accuracy. However,
 222 all data and code to reproduce the results described are available in a dedicated GitHub
 223 repository (<https://github.com/ntustison/DevCCF-Velocity-Flow>).

224 The normalized time point scalar value for each atlas/point-set in the temporal domains [0, 1]
 225 was also defined. Given the increasingly larger gaps in the postnatal timepoint sampling, we
 226 made two adjustments. Based on known mouse brain development, we used 28 days for the
 227 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. 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. Using multithreading, each iteration took about six minutes.

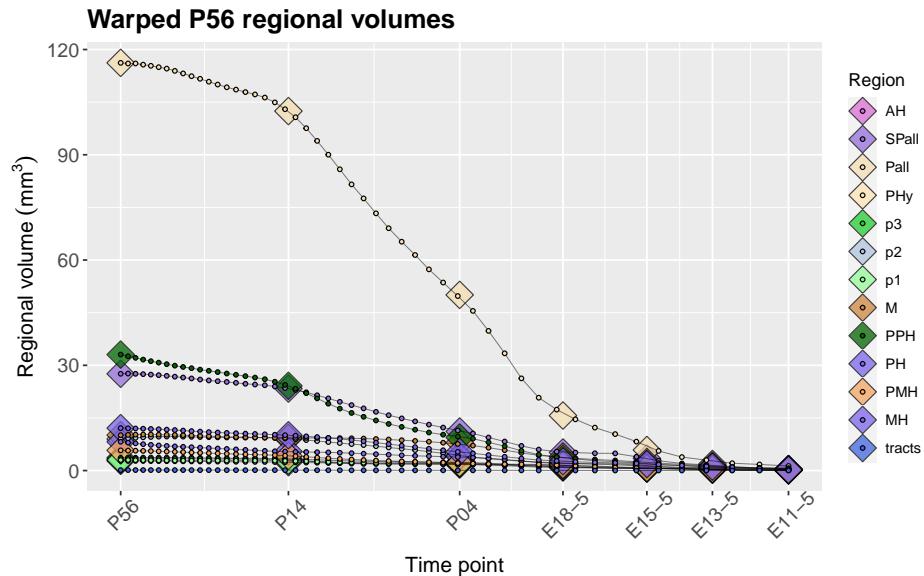


Figure 4: After the velocity field is generated, we can use it to warp the simplified labels of the P56 atlas continuously over the interval [0, 1] and plot the volumes of the atlas regions. Note how they compare with the volumes of the same regions in the other atlases.

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. After optimization, we use the velocity field to warp the P56 set of labels to each of the other atlas time points to compare the volumes of the different simplified annotated regions. This is shown in Figure 4.

244 The DevCCF transform model

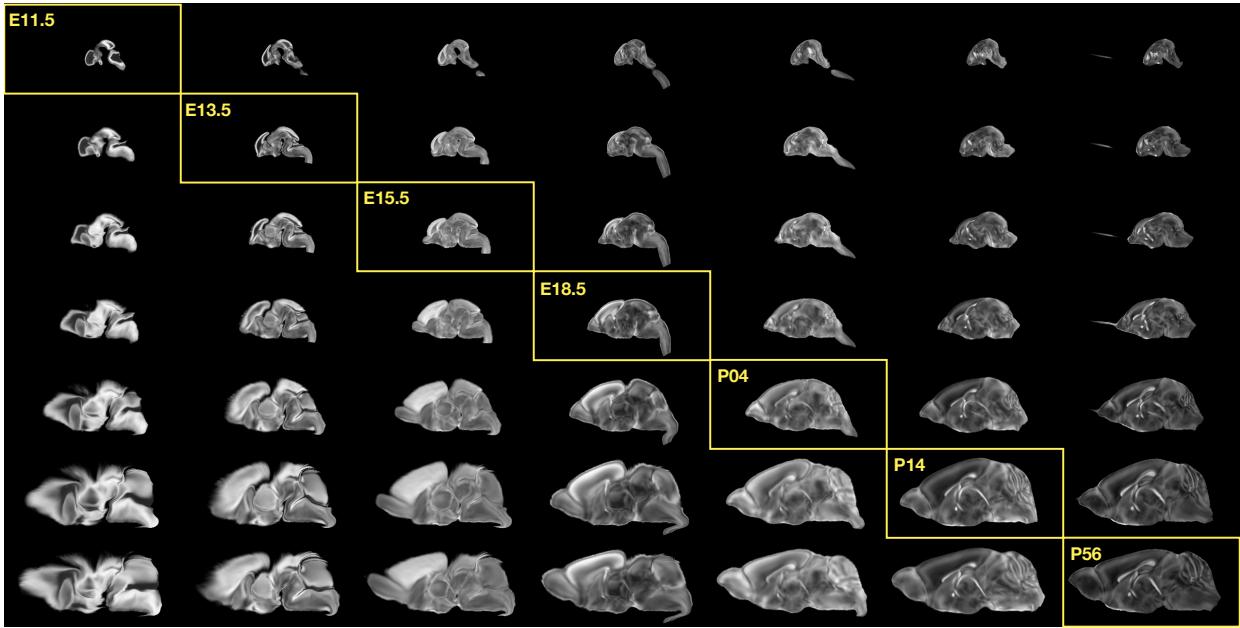


Figure 5: Mid-sagittal visualization of the effects of the transformation model in warping every developmental stage to the time point of every other developmental stage. The original images are located along the diagonal. Columns correspond to the warped original image whereas the rows represent the reference space to which each image is warped.

245 Once optimized, the resulting velocity field can be used to generate the deformable transform
 246 between any two continuous points within the time interval bounded by E11.5 and P56. In
 247 Figure 5, we transform each atlas to the space of every other atlas using the DevCCF
 248 transform model. Additionally, one can use this transformation model to construct virtual
 249 templates in the temporal gaps of the DevCCF. This is illustrated in Figure 6 where we used
 250 the optimized velocity field to construct virtual-templates at time point P10.3 and P20—
 251 arbitrarily chosen simply to demonstrate the concept. After situating these time points
 252 within the normalized time point interval, the existing adjacent DevCCF atlases on either
 253 chronological side can be warped to the desired time point. A subsequent call to one of the
 254 ANTsX template building functions then permits the construction of the template at that
 255 time point. Note that both of these usage examples can be found on the GitHub repository
 256 given above.

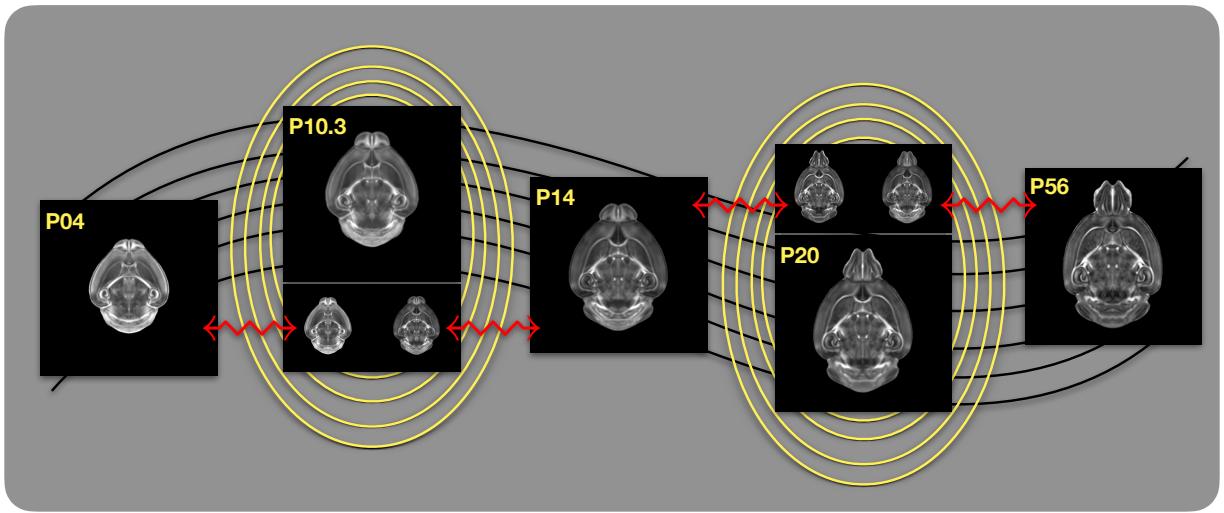


Figure 6: Illustration of the use of the velocity flow model for creating virtual templates at continuous time points not represented in one of the existing DevCCF time points. For example, FA templates at time point P10.3 and P20 can be generated by warping the existing temporally adjacent developmental templates to the target time point and using those images in the ANTsX template building process.

257 **Discussion**

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

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

277 Despite the significant expansion of available developmental age templates beyond what pre-
278 viously existed (e.g., Allen CCFv3), there still exist temporal gaps in the DevCCF. However,
279 pioneering work involving diffeomorphic transformations allowed us to continuously situate
280 the existing templates within a time-varying velocity flow model. This allows one to deter-
281 mine the diffeomorphic transformation from any one temporal location to any other temporal
282 location within the time span defined by the E11.5 and P56 templates. This functionality
283 is built on multiple components from the Insight Segmentation and Registratiton Toolkit
284 including the B-spline scattered data approximation technique for field regularization and
285 velocity field integration using fourth order Runge-Kutta. This velocity field model permits

²⁸⁶ intra-template comparison and the construction of virtual templates where a template can
²⁸⁷ be estimated at any continuous time point within the temporal domain. This novel appli-
²⁸⁸ cation can potentially enhance our understanding of intermediate developmental stages. To
²⁸⁹ increase its impact and reproduce the results shown previously, we have made the data and
²⁹⁰ code publicly available at <https://github.com/ntustison/DevCCF-Velocity-Flow>.

²⁹¹ Although ANTsX is quite evolved in its development and functionality, there are several areas
²⁹² which are currently under active development or consideration for further expansion. Most
²⁹³ notably, as in our human applications, deep learning has had a significant impact in steering
²⁹⁴ our attention. Core functionality, such as brain extraction for mouse brain mapping, would
²⁹⁵ benefit from increasing the number of available modalities. Additionally, as with much deep
²⁹⁶ learning development, such work will require additional data but is significantly facilitated
²⁹⁷ by the tools that we have created in both ANTsPyNet and ANTsRNet.

²⁹⁸ **Methods**

²⁹⁹ The following methods are all available as part of the ANTsX ecosystem with analogous
³⁰⁰ elements existing in both ANTsR (ANTs in R) and ANTsPy (ANTs in Python) with and
³⁰¹ ANTs/ITK C++ core. However, most of the development for the work described below was
³⁰² performed using ANTsPy. For equivalent calls in ANTsR, please see the ANTsX tutorial at
³⁰³ <https://tinyurl.com/antsxtutorial>.

³⁰⁴ **Preprocessing: bias field correction and denoising**

³⁰⁵ As in human studies, bias field correction and image denoising are standard preprocessing
³⁰⁶ steps in improving overall image quality in mouse brain images. The bias field, a gradual
³⁰⁷ spatial intensity variation in images, can arise from various sources such as magnetic field in-
³⁰⁸ homogeneity or acquisition artifacts, leading to distortions that can compromise the quality
³⁰⁹ of brain images. Correcting for bias fields ensures a more uniform and consistent repre-
³¹⁰ sentation of brain structures, enabling accurate quantitative analysis. Additionally, brain
³¹¹ images are often susceptible to various forms of noise, which can obscure subtle features
³¹² and affect the precision of measurements. Denoising techniques help mitigate the impact
³¹³ of noise, enhancing the signal-to-noise ratio and improving the overall image quality. The
³¹⁴ well-known N4 bias field correction algorithm²⁴ has its origins in the ANTs toolkit which
³¹⁵ was implemented and introduced into the ITK toolkit. Similarly, ANTsX contains an im-
³¹⁶ plementation of a well-performing patch-based denoising technique⁵² and is also available as
³¹⁷ an image filter to the ITK community.

³¹⁸ **ANTsXNet mouse brain applications**

³¹⁹ *General notes regarding deep learning training.*

³²⁰ All network-based approaches described below were implemented and organized in the
³²¹ ANTsXNet libraries comprising Python (ANTsPyNet) and R (ANTsRNet) analogs using the
³²² Keras/Tensorflow libraries available as open-source in ANTsX GitHub repositories. For the

323 various applications, both share the identically trained weights for mutual reproducibility.
324 Training data was provided by manual labeling by various co-authors and expanded using
325 both intensity-based and shape-based data augmentation techniques.

326 Intensity-based data augmentation consisted of randomly added noise based on
327 ITK functionality, simulated bias fields based on N4 bias field modeling, and his-
328 togram warping for mimicking well-known MRI intensity nonlinearities.^{25,55} These
329 augmentation techniques are available in ANTsXNet (only ANTsPyNet versions are
330 listed): simulated bias field: `antspynet.simulate_bias_field(...)`, image noise:
331 `antspyhet.add_noise_to_image(...)`, and MRI intensity nonlinear characteriza-
332 tion: `antspynet.histogram_warp_image_intensities(...)`. Shape-based data
333 augmentation used both random linear and nonlinear deformations. This func-
334 tionality is also instantiated within ANTsXNet in terms of random spatial warping:
335 `antspynet.randomly_transform_image_data(...)`.

336 For all GPU training, we used Python scripts for creating custom batch generators. As such
337 batch generators tend to be application-specific, we store them in a separate GitHub reposi-
338 tory for public availability (<https://github.com/ntustison/ANTsXNetTraining>). In terms of
339 GPU hardware, all training was done on a DGX (GPUs: 4X Tesla V100, system memory:
340 256 GB LRDIMM DDR4).

341 *Brain extraction.*

342 Similar to human neuroimage processing, brain extraction is a crucial preprocessing step for
343 accurate brain mapping. Within ANTsXNet, we have created several deep learning networks
344 for brain extraction for several image modalities (e.g., T1, FLAIR, fractional anisotropy).
345 Similarly, for the developmental brain atlas work⁴⁷ we developed similar functionality for
346 mouse brains of different modalities and developmental age. All networks use a conven-
347 tional 2-D U-net architecture⁵⁶ and perform prediction in a slice-wise fashion given the
348 limitations of the acquisition protocols (e.g., missing slices, slice thickness). Currently,
349 coronal and sagittal networks are available for both E13.5 and E15.5 data and coronal
350 network for T2-weighted MRI. In ANTsPyNet, this functionality is available in the pro-
351 gram `antspynet.mouse_brain_extraction(...)`. Even when physical brain extraction is

352 performed prior to image acquisition, artifacts, such as bubbles or debris, can complicate
353 subsequent processing. Similar to the brain extraction networks, a 2-D U-net architecture⁵⁶
354 was created to separate the background and foreground.

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

356 To further enhance the data prior to designing mapping protocols, additional networks were
357 created. A well-performing deep back projection network⁵⁷ was ported to ANTsXNet and
358 expanded to 3-D for various super-resolution applications,⁵⁸ including mouse data. Finally,
359 features of anatomical significance, namely the cerebellum and hemispherical midline were
360 captured in these data using deep learning networks.

361 **Intra-slice image registration with missing slice imputation**

362 Volumetric gene expression slice data was collated into 3-D volumes. Prior to mapping
363 this volume to the corresponding structural data and, potentially, to the appropriate tem-
364 plate, alignment was improved using deformable registration on contiguous slices. How-
365 ever, one of the complications associated with these image data was the unknown num-
366 ber of missing slices, the number of consecutive missing slices, and the different locations
367 of these missing slices. To handle this missing data problem, we found that data in-
368 terpolation using the B-spline approximation algorithm cited earlier⁵⁴ (ANTsPy function:
369 `ants.fit_bspline_object_to_scattered_data(...)`). This provided sufficient data in-
370 terpolation fidelity to perform continuous slicewise registration. Other possible variants that
371 were considered but deemed unnecessary was performing more than one iteration cycling
372 through data interpolation and slicewise alignment. The other possibility was incorporating
373 the super-resolution technique described earlier. But again, our data did not require these
374 additional steps.

375 **Image registration**

376 The ANTs registration toolkit is a complex framework permitting highly tailored solu-
377 tions to pairwise image registration scenarios.⁵⁹ It includes innovative transformation mod-

378 els for biological modeling^{39,49} and has proven capable of excellent performance.^{40,60} Various
379 parameter sets targeting specific applications have been packaged with the different
380 ANTsX platforms, specifically ANTs, ANTsPy, and ANTsR.²⁵ In ANTsPy, the function
381 `ants.registration(...)` is used to register a pair of images or a pair of image sets where
382 `type_of_transform` is a user-specified option that invokes a specific parameter set. For
383 example `type_of_transform='antsRegistrationSyNQuick[s]'` is an oft-used parameter
384 set.

385 Initially, linear optimization is initialized with center of (intensity) mass alignment typically
386 followed by optimization of both rigid and affine transforms using the mutual information
387 similarity metric. This is followed by diffeomorphic deformable alignment using symmetric
388 normalization (SyN) with Gaussian³⁹ or B-spline regularization⁴⁹ where the forward trans-
389 form is invertible and differentiable. The similarity metric employed at this latter stage
390 is typically either neighborhood cross-correlation or mutual information. Note that these
391 parameter sets are robust to input image type (i.e., LSFM, Nissl staining, and the various
392 MRI modalities) and are adaptable to mousing image geometry scaling. Further details can
393 be found in the various documentation sources for these ANTsX packages.

394 Template generation

395 ANTsX provides functionality for constructing templates from a set (or multi-modal sets) of
396 input images as originally described⁵⁰ and recently used to create the DevCCF templates.⁴⁷
397 An initial template estimate is constructed from an existing subject image or a voxelwise
398 average derived from a rigid pre-alignment of the image population. Pairwise registration
399 between each subject and the current template estimate is performed using the Symmetric
400 Normalization (SyN) algorithm.³⁹ The template estimate is updated by warping all subjects
401 to the space of the template, performing a voxelwise average, and then performing a “shape
402 update” of this latter image by warping it by the average inverse deformation, thus yielding
403 a mean image of the population in terms of both intensity and shape.

404 **Continuous developmental velocity flow transformation model**

405 Given multiple, linearly or non-linearly ordered point sets where individual points across are
406 in one-to-one correspondence, we developed an approach for generating a velocity flow trans-
407 formation model to describe a time-varying diffeomorphic mapping as a variant of the inexact
408 landmark matching solution. Integration of the resulting velocity field can then be used to
409 describe the displacement between any two time points within this time-parameterized do-
410 main. Regularization of the sparse correspondence between point sets is performed using a
411 generalized B-spline scattered data approximation technique,⁵⁴ also developed by the ANTsX
412 developers and contributed to ITK.

413 To apply this methodology to the developmental templates,⁴⁷ we coalesced the manual par-
414 cellations of the developmental templates into 26 common anatomical regions (13 per hemi-
415 sphere). We then used these regions to generate invertible transformations between succe-
416 ssive time points. Specifically each label was used to create a pair of single region images
417 resulting in 26 pairs of “source” and “target” images. The multiple image pairs were used
418 to iteratively estimate a diffeomorphic pairwise transform. Given the seven atlases E11.5,
419 E13.5, E15.5, E18.5, P4, P14, and P56, this resulted in 6 sets of transforms between succe-
420 ssive time points. Given the relative sizes between atlases, on the order of 10^6 points were
421 randomly sampled labelwise in the P56 template space and propagated to each successive
422 atlas providing the point sets for constructing the velocity flow model. Approximately 125
423 iterations resulted in a steady convergence based on the average Euclidean norm between
424 transformed point sets. Ten integration points were used and point sets were distributed
425 along the temporal dimension using a log transform for a more evenly spaced sampling.

426 **Visualization**

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

431 **Data availability.** All data used in this work are publicly available. The DevCCF atlas is
432 available at <https://kimlab.io/brain-map/DevCCF/>. Additionally, all software discussed is
433 publicly available. ANTsPy and ANTsR are available through GitHub at the ANTsX Ecosys-
434 tem (<https://github.com/ANTsX>). A GitHub repository specific to the work discussed in the
435 manuscript was created and is available at <https://github.com/ntustison/DevCCF-Velocity->
436 **Flow.**

437 **References**

- 438 1. Keller, P. J. & Ahrens, M. B. Visualizing whole-brain activity and development at
439 the single-cell level using light-sheet microscopy. *Neuron* **85**, 462–83 (2015).
- 440 2. La Manno, G. *et al.* Molecular architecture of the developing mouse brain. *Nature*
441 **596**, 92–96 (2021).
- 442 3. Wen, L. *et al.* Single-cell technologies: From research to application. *Innovation*
443 (*Camb*) **3**, 100342 (2022).
- 444 4. Oh, S. W. *et al.* A mesoscale connectome of the mouse brain. *Nature* **508**, 207–14
445 (2014).
- 446 5. Gong, H. *et al.* Continuously tracing brain-wide long-distance axonal projections in
447 mice at a one-micron voxel resolution. *Neuroimage* **74**, 87–98 (2013).
- 448 6. Li, A. *et al.* Micro-optical sectioning tomography to obtain a high-resolution atlas of
449 the mouse brain. *Science* **330**, 1404–8 (2010).
- 450 7. Ueda, H. R. *et al.* Tissue clearing and its applications in neuroscience. *Nat Rev*
451 *Neurosci* **21**, 61–79 (2020).
- 452 8. Ståhl, P. L. *et al.* Visualization and analysis of gene expression in tissue sections by
453 spatial transcriptomics. *Science* **353**, 78–82 (2016).
- 454 9. Burgess, D. J. Spatial transcriptomics coming of age. *Nat Rev Genet* **20**, 317 (2019).
455
- 456 10. MacKenzie-Graham, A. *et al.* A multimodal, multidimensional atlas of the C57BL/6J
457 mouse brain. *J Anat* **204**, 93–102 (2004).
- 458 11. Mackenzie-Graham, A. J. *et al.* Multimodal, multidimensional models of mouse brain.
459 *Epilepsia* **48 Suppl 4**, 75–81 (2007).
- 460 12. Dong, H. W. *Allen reference atlas. A digital color brain atlas of the C57BL/6J male*
461 *mouse.* (John Wiley; Sons, 2008).

- 462 13. Wang, Q. *et al.* The allen mouse brain common coordinate framework: A 3D reference
463 atlas. *Cell* **181**, 936–953.e20 (2020).
- 464 14. Johnson, G. A. *et al.* Waxholm space: An image-based reference for coordinating
465 mouse brain research. *Neuroimage* **53**, 365–72 (2010).
- 466 15. Oguz, I., Zhang, H., Rumple, A. & Sonka, M. RATS: Rapid automatic tissue segmen-
467 tation in rodent brain MRI. *J Neurosci Methods* **221**, 175–82 (2014).
- 468 16. Sawiak, S. J., Picq, J.-L. & Dhenain, M. Voxel-based morphometry analyses of in
469 vivo MRI in the aging mouse lemur primate. *Front Aging Neurosci* **6**, 82 (2014).
- 470 17. Ashburner, J. SPM: A history. *Neuroimage* **62**, 791–800 (2012).
- 471
- 472 18. Modat, M. *et al.* Fast free-form deformation using graphics processing units. *Comput
473 Methods Programs Biomed* **98**, 278–84 (2010).
- 474 19. Tyson, A. L. *et al.* Accurate determination of marker location within whole-brain
475 microscopy images. *Sci Rep* **12**, 867 (2022).
- 476 20. Pallast, N. *et al.* Processing pipeline for atlas-based imaging data analysis of struc-
477 tural and functional mouse brain MRI (AIDAmri). *Front Neuroinform* **13**, 42 (2019).
- 478 21. Jenkinson, M., Beckmann, C. F., Behrens, T. E. J., Woolrich, M. W. & Smith, S. M.
479 FSL. *Neuroimage* **62**, 782–90 (2012).
- 480 22. Yeh, F.-C., Wedeen, V. J. & Tseng, W.-Y. I. Generalized q-sampling imaging. *IEEE
481 Trans Med Imaging* **29**, 1626–35 (2010).
- 482 23. Jorge Cardoso, M. *et al.* STEPS: Similarity and truth estimation for propagated
483 segmentations and its application to hippocampal segmentation and brain parcelation.
Med Image Anal **17**, 671–84 (2013).
- 484 24. Tustison, N. J. *et al.* N4ITK: Improved N3 bias correction. *IEEE Trans Med Imaging
485* **29**, 1310–20 (2010).

- 486 25. Tustison, N. J. *et al.* The ANTsX ecosystem for quantitative biological and medical
487 imaging. *Sci Rep* **11**, 9068 (2021).
- 488 26. Goubran, M. *et al.* Multimodal image registration and connectivity analysis for inte-
489 gration of connectomic data from microscopy to MRI. *Nat Commun* **10**, 5504 (2019).
- 490 27. Celestine, M., Nadkarni, N. A., Garin, C. M., Bougacha, S. & Dhenain, M. Sammba-
491 MRI: A library for processing SmAll-MaMmal BrAin MRI data in python. *Front
Neuroinform* **14**, 24 (2020).
- 492 28. Ioanas, H.-I., Marks, M., Zerbi, V., Yanik, M. F. & Rudin, M. An optimized regis-
493 tration workflow and standard geometric space for small animal brain imaging. *Neu-
roimage* **241**, 118386 (2021).
- 494 29. Cox, R. W. AFNI: What a long strange trip it's been. *Neuroimage* **62**, 743–7 (2012).
495
- 496 30. Ni, H. *et al.* A robust image registration interface for large volume brain atlas. *Sci
497 Rep* **10**, 2139 (2020).
- 498 31. Jin, M. *et al.* SMART: An open-source extension of WholeBrain for intact mouse
499 brain registration and segmentation. *eNeuro* **9**, (2022).
- 500 32. Fürth, D. *et al.* An interactive framework for whole-brain maps at cellular resolution.
501 *Nat Neurosci* **21**, 139–149 (2018).
- 502 33. Negwer, M. *et al.* FriendlyClearMap: An optimized toolkit for mouse brain mapping
503 and analysis. *Gigascience* **12**, (2022).
- 504 34. Klein, S., Staring, M., Murphy, K., Viergever, M. A. & Pluim, J. P. W. Elastix: A
505 toolbox for intensity-based medical image registration. *IEEE Trans Med Imaging* **29**,
196–205 (2010).
- 506 35. Carey, H. *et al.* DeepSlice: Rapid fully automatic registration of mouse brain imaging
507 to a volumetric atlas. *Nat Commun* **14**, 5884 (2023).

- 508 36. Bajcsy, R. & Broit, C. Matching of deformed images. in *Sixth International Conference on Pattern Recognition (ICPR'82)* 351–353 (1982).
- 509
- 510 37. Bajcsy, R. & Kovacic, S. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing* **46**, 1–21 (1989).
- 511
- 512 38. 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).
- 513
- 514 39. 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).
- 515
- 516 40. Klein, A. *et al.* Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration. *Neuroimage* **46**, 786–802 (2009).
- 517
- 518 41. Murphy, K. *et al.* Evaluation of registration methods on thoracic CT: The EMPIRE10 challenge. *IEEE Trans Med Imaging* **30**, 1901–20 (2011).
- 519
- 520 42. 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).
- 521
- 522 43. Wang, H. *et al.* Multi-atlas segmentation with joint label fusion. *IEEE Trans Pattern Anal Mach Intell* **35**, 611–23 (2013).
- 523
- 524 44. 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](https://doi.org/10.1007/s12021-014-9245-2).
- 525
- 526 45. 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).
- 527

- 528 46. McCormick, M., Liu, X., Jomier, J., Marion, C. & Ibanez, L. ITK: Enabling repro-
529
ducible research and open science. *Front Neuroinform* **8**, 13 (2014).
- 530 47. Kronman, F. A. *et al.* Developmental mouse brain common coordinate framework.
531 *bioRxiv* (2023) doi:[10.1101/2023.09.14.557789](https://doi.org/10.1101/2023.09.14.557789).
- 532 48. Beg, M. F., Miller, M. I., Trouvé, A. & Younes, L. Computing large deformation
533 metric mappings via geodesic flows of diffeomorphisms. *International Journal of
Computer Vision* **61**, 139–157 (2005).
- 534 49. Tustison, N. J. & Avants, B. B. Explicit B-spline regularization in diffeomorphic image
535 registration. *Front Neuroinform* **7**, 39 (2013).
- 536 50. Avants, B. B. *et al.* The optimal template effect in hippocampus studies of diseased
537 populations. *Neuroimage* **49**, 2457–66 (2010).
- 538 51. Tustison, N. J. *et al.* Optimal symmetric multimodal templates and concatenated
539 random forests for supervised brain tumor segmentation (simplified) with ANTsR.
Neuroinformatics **13**, 209–25 (2015).
- 540 52. Manjón, J. V., Coupé, P., Martí-Bonmatí, L., Collins, D. L. & Robles, M. Adaptive
541 non-local means denoising of MR images with spatially varying noise levels. *J Magn
Reson Imaging* **31**, 192–203 (2010).
- 542 53. Klein, A. *et al.* Evaluation of volume-based and surface-based brain image registration
543 methods. *Neuroimage* **51**, 214–20 (2010).
- 544 54. Tustison, N. J. & Amini, A. A. Biventricular myocardial strains via nonrigid registra-
545 tion of anatomical NURBS model [corrected]. *IEEE Trans Med Imaging* **25**, 94–112
(2006).
- 546 55. Nyúl, L. G., Udupa, J. K. & Zhang, X. New variants of a method of MRI scale
547 standardization. *IEEE Trans Med Imaging* **19**, 143–50 (2000).
- 548 56. Falk, T. *et al.* U-net: Deep learning for cell counting, detection, and morphometry.
549 *Nat Methods* **16**, 67–70 (2019).

- 550 57. Haris, M., Shakhnarovich, G. & Ukita, N. Deep back-projection networks for super-
resolution. in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recog-*
551 *nition* 1664–1673 (2018). doi:[10.1109/CVPR.2018.00179](https://doi.org/10.1109/CVPR.2018.00179).
- 552 58. Avants, B. B. *et al.* Concurrent 3D super resolution on intensity and segmentation
maps improves detection of structural effects in neurodegenerative disease. *medRxiv*
553 (2023) doi:[10.1101/2023.02.02.23285376](https://doi.org/10.1101/2023.02.02.23285376).
- 554 59. Avants, B. B. *et al.* The Insight ToolKit image registration framework. *Front Neu-*
555 *roinform* **8**, 44 (2014).
- 556 60. Avants, B. B. *et al.* A reproducible evaluation of ANTs similarity metric performance
557 in brain image registration. *Neuroimage* **54**, 2033–44 (2011).