

1

2 **The ANTsX Ecosystem for Mapping the
3 Mouse Brain**

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

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

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

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

9 ⁴Allen Institute for Brain Science, Seattle, WA

10

11 Corresponding author:

12 Nicholas J. Tustison, DSc

13 Department of Radiology and Medical Imaging

14 University of Virginia

15 ntustison@virginia.edu

¹⁶ **Abstract**

¹⁷ Precision mapping techniques coupled with high resolution image acquisition of the mouse
¹⁸ brain permit the study of the spatial organization of gene activity and their mutual interac-
¹⁹ tion for a comprehensive view of salient structural/functional relationships. Such research
²⁰ is facilitated by standardized anatomical coordinate systems, such as the well-known Allen
²¹ Common Coordinate Framework (AllenCCFv3), and the ability to spatially map to such
²² standardized spaces. The Advanced Normalization Tools Ecosystem (ANTsX) is a compre-
²³ hensive open-source software toolkit for generalized quantitative imaging, which includes
²⁴ template building and mapping functionality, with applicability to multiple organ systems,
²⁵ modalities, and animal species. Herein, we illustrate the utility of ANTsX for generating
²⁶ precision spatial mappings of the mouse brain. First, we provide ANTsX-based protocols for
²⁷ mapping MERFISH, fMOST, and lightsheet datasets to AllenCCFv3 accounting for com-
²⁸ mon artefacts and other confounds. Additionally, recently developed ANTsX functionality
²⁹ permits the generation of velocity flow-based mappings for serial data. Using the recently in-
³⁰ troduced Developmental Common Coordinate Framework, we evaluate and describe the pub-
³¹ licly available ANTsX-based protocols for generating a velocity flow-based mapping spanning
³² the spatiotemporal domain of the developmental trajectory. Possible future work includes
³³ the introduction of additional developmental time points and application to histological slice
³⁴ stacking.

³⁵ Introduction

³⁶ Over the past two decades there have been significant advancements in mesoscopic analysis of the mouse brain. It is now 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 transcriptomics⁹ are all well-utilized in the course of scientific investigations of mesoscale relationships in the mouse brain.

⁴⁵ An important component of this research is the ability to map the various image data to anatomical reference frames¹¹ 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 severely complicate assumed correspondence for conventional spatial mapping approaches.

⁵⁵ 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) package¹⁶ for brain extraction. Of the many publicly available packages, most, if not all have well-established package dependencies originally developed on human brain data. SPMMouse,¹⁷ for example, is based on the well-known Statistical Parametric Mapping (SPM) software package.¹⁸ The automated mouse atlas propagation (aMAP) tool is largely a front-end for the NiftyReg image registration package¹⁹ applied to mouse data which is currently available as a Python module.²⁰ NiftyReg is also used by the Atlas-based Imaging Data

64 Analysis (AIDA) MRI pipeline²¹ as well as the Multi Atlas Segmentation and Morphometric
65 Analysis Toolkit (MASMAT). Whereas the former also incorporates the FMRIB Software
66 Library (FSL)²² for brain extraction and DSISTudio²³ for DTI processing, the latter uses
67 NiftySeg and multi-consensus labeling tools²⁴ for brain extraction and parcellation. In ad-
68 dition, MASMAT incorporates N4 bias field correction²⁵ from the Advanced Normalization
69 Tools Ecosystem (ANTsX)²⁶ as do the packages Multi-modal Image Registration And Con-
70 nectivity anaLysis (MIRACL),²⁷ Sammba-MRI,²⁸ and Small Animal Magnetic Resonance
71 Imaging (SAMRI).²⁹ However, whereas Saamba-MRI uses AFNI³⁰ for image registration;
72 MIRACL, SAMRI, and BrainsMapi³¹ all use ANTsX registration tools. Other packages
73 use landmark-based approaches to image registration including SMART—³²an R package
74 for semi-automated landmark-based registration and segmentation of mouse brain based
75 on WholeBrain.³³ FriendlyClearMap³⁴ uses the landmark-based registration functionality of
76 Elastix.³⁵ Finally, the widespread adoption of deep learning techniques has also influenced
77 development in mouse brain imaging methodologies. For example, if tissue deformations
78 are not considered problematic for a particular dataset, DeepSlice can be used to determine
79 affine mappings³⁶ with the optimal computational efficiency associated with neural networks.

80 The ANTsX Ecosystem

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

91 Apart from its registration capabilities, ANTsX comprises additional functionality such as

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>	
MRF-based 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>).

92 template generation,⁴⁴ point set data approximation,⁴⁵ and deep learning networks specifically trained for mouse data (see Table 1). The comprehensive use of the toolkit has demonstrated superb performance in multiple application areas (e.g., consensus labeling,⁴⁶ brain tumor segmentation,⁴⁷ and cardiac motion estimation⁴⁸). Importantly, ANTs is built on the Insight Toolkit (ITK)⁴⁹ deriving benefit from the open-source community of scientists and programmers and providing an open-source venue for algorithmic development, evaluation, and improvement.

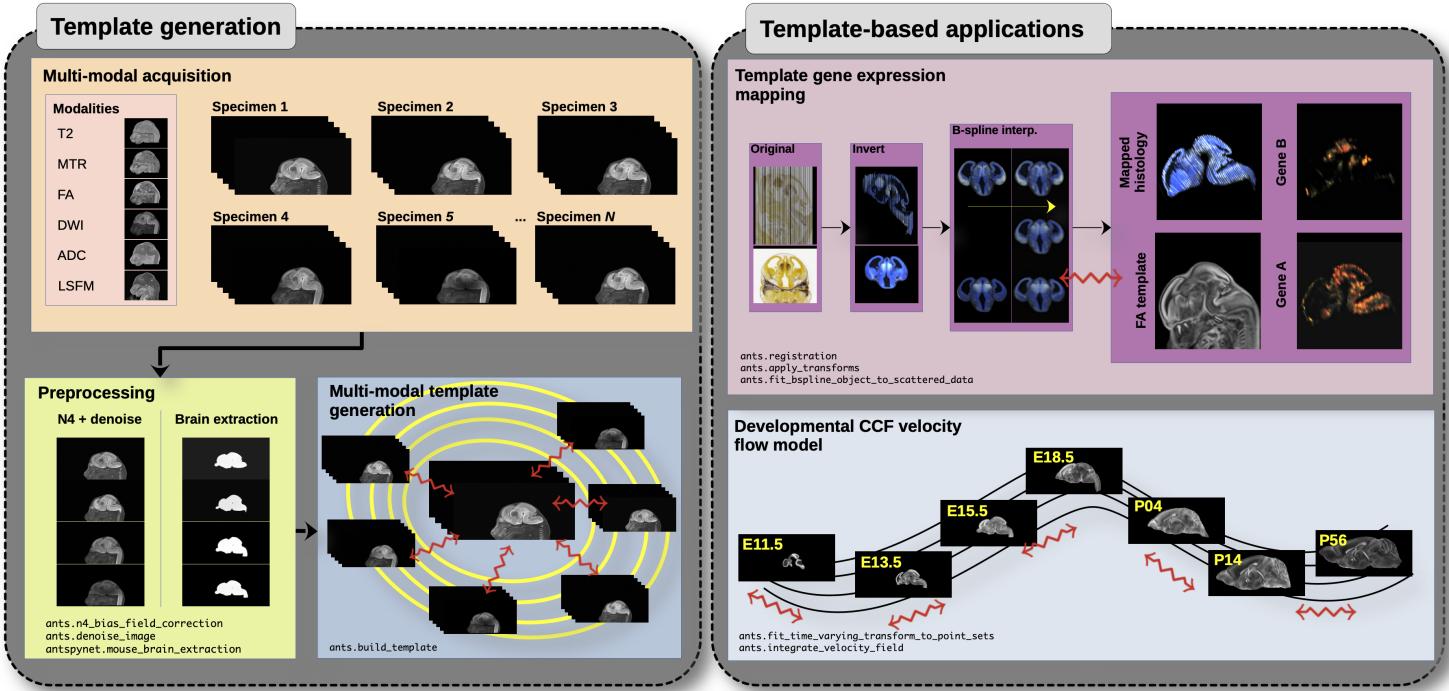


Figure 1: Illustration of a mouse brain template generation workflow and related template-based applications demonstrating the utility of different ANTsX tools. After imaging acquisition of the study population, various preprocessing steps are applied to the imaging data such as bias correction, denoising,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

99 Recently, the developmental common coordinate framework (DevCCF) was introduced to
 100 the mouse brain research community as a public resource.¹⁵ These symmetric atlases, com-
 101 prising both multimodal image data and anatomical segmentations defined by developmental
 102 ontology, sample the mouse embryonic days (E) 11.5, E13.5, E15.5, E18.5 and postnatal day

¹⁰³ (P) 4, P14, and P56. Modalities include light sheet floourescence miscroscopy (LSFM) and at
¹⁰⁴ least four MRI contrasts per developmental stage. Anatomical parcellations are also available
¹⁰⁵ for each time point and were generated from ANTsX-based mappings of gene expression and
¹⁰⁶ other cell type data. The P56 template was integrated with the Allen CCFv3 to further in-
¹⁰⁷ crease the practical utility of the DevCCF. These processes, specifically template generation
¹⁰⁸ and multi-modal image mapping, were performed using ANTsX functionality in the presence
¹⁰⁹ of previously noted image mapping difficulties (e.g., missing slices, tissue distortion).

¹¹⁰ Given the temporal gaps in the discrete set of developmental atlases with the potential for
¹¹¹ additional interpolative time points, we discuss the strategy of the current DevCCF tem-
¹¹² plate generation¹⁵ and provide additional information for the interested reader. Related, we
¹¹³ also provide an open-source framework, through ANTsX, for inferring correspondence within
¹¹⁴ the temporally continuous domain sampled by the existing set of embryonic and postnatal
¹¹⁵ atlases of the DevCCF. Although alternative approaches are possible for interpolating be-
¹¹⁶ tween time points, this recently developed ANTsX functionality permits the generation of
¹¹⁷ a diffeomorphic velocity flow transformation model,⁵⁰ influenced by previous work.⁵¹ The
¹¹⁸ resulting time-parameterized velocity field spans the stages of the DevCCF where mappings
¹¹⁹ between any two continuous time points within the span bounded by the E11.5 and P56
¹²⁰ atlases is determined by integration of the optimized velocity field. This functionality is
¹²¹ available through ANTsX (via R and Python ANTsX packages) with a dedicated GitHub
¹²² repository that contains all data, scripts, and other guidance necessary to both reproduce
¹²³ what is described below and to illustrate how future researchers can incorporate additional
¹²⁴ atlases into a more densely sampled model in a straightforward manner.

₁₂₅ **Results**

₁₂₆ **Template building**

₁₂₇ Template building using ANTsX tools was first described in the context of hippocampal
₁₂₈ studies.⁴⁴ Multi-modal and symmetrical variants were subsequently described as part
₁₂₉ of a proposed brain tumor segmentation approach based on random forests.⁵² Tem-
₁₃₀ plate building capabilities are available in both ANTsPy (`ants.build_template(...)`)
₁₃₁ and ANTsR (`buildTemplate(...)`) as well as part of the core ANTs package (e.g.,
₁₃₂ `antsMultivariateTemplateConstruction.sh`).

₁₃₃ **Data preparation**

₁₃₄ Multi-modal symmetric template construction is performed separately for each develop-
₁₃₅ mental stage. Prior to optimization, preprocessing can include several steps not all of
₁₃₆ which are required but are dependent on the data and the particular requirements of the
₁₃₇ study. For MRI scans, inhomogeneity correction is often necessary and can be performed
₁₃₈ using the ANTsPy function `ants.n4_bias_field_correction(...)` which is a wrapper
₁₃₉ for the N4 algorithm.²⁵ Denoising is another preprocessing step that can potentially im-
₁₄₀ prove template quality results. The ANTsPy function `ants.denoise_image(...)` is an
₁₄₁ implementation of a well-known denoising algorithm.⁵³ For a typical image, both of these
₁₄₂ steps takes approximately on the order of a couple minutes. In ANTsX, due to legacy
₁₄₃ code issues, only bias correction is wrapped with template building so one need not per-
₁₄₄ form this step prior to optimization. In addition, brain extraction has demonstrated im-
₁₄₅ proved performance in the context of human brain normalization⁵⁴ and is similarly used
₁₄₆ in mouse brain registration to maximize alignment. Various approaches within ANTs are
₁₄₇ possible including a template-based approach `antsBrainExtraction.sh` or using deep learn-
₁₄₈ ing `antspynet.mouse_brain_extraction(...)`. Additionally, it is important to ensure a
₁₄₉ standardized orientation, similar to the Dicom standard for human brain imaging. A study
₁₅₀ requirement of template bilateral symmetry is also an important consideration prior to tem-
₁₅₁ plate generation. This can be performed by either flipping all the input images contralaterally

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

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

164 Optimization

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

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

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

182 The DevCCF Velocity Flow Model

183 To continuously interpolate transformations between the different stages of the De-
184 vCCF atlases, a velocity flow model was constructed using Dev-CCF derived data and
185 ANTsX functionality recently introduced into both the ANTsR and ANTsPy packages.
186 Both platforms include a complete suite of functions for determining dense correspon-
187 dence from sparse landmarks based on a variety of transformation models ranging from
188 standard linear models (i.e., rigid, affine) to deformable diffeomorphic models (e.g,
189 symmetric normalization).⁴⁰ The latter set includes velocity flow models for both the
190 pairwise scenario (`ants.fit_transform_to_paired_points(...)`) and for multiple
191 sets (`ants.fit_time_varying_transform_to_point_sets(...)`), as in the case of the
192 DevCCF. Several self-contained tutorials illustrating usage for these functions are available
193 at <https://tinyurl.com/antsxtutorial>.

194 ANTsX, being built on top of ITK, uses an ITK image data structure for the 4-D velocity
195 field where each voxel contains the x , y , z components of the field at that point. Field
196 regularization is provided by a B-spline scattered data approximation technique⁵¹ which
197 permits individual point weighting. Both field regularization and integration of the velocity
198 field are built on ITK functions contributed from ANTsX development.

199 Data preparation

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

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

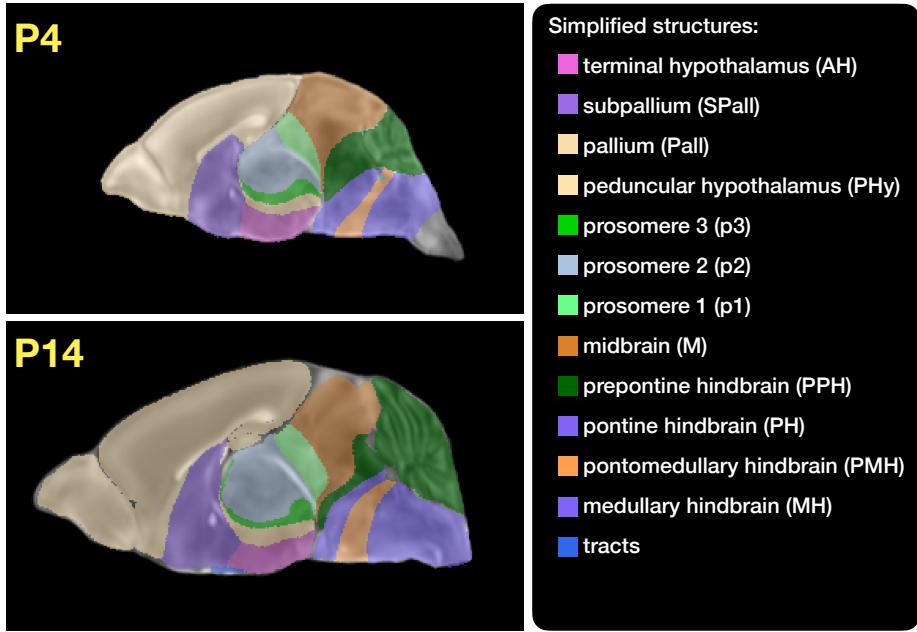


Figure 2: Annotated regions representing common labels across developmental stages which are illustrated for both P4 and P14.

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.

220 Optimization

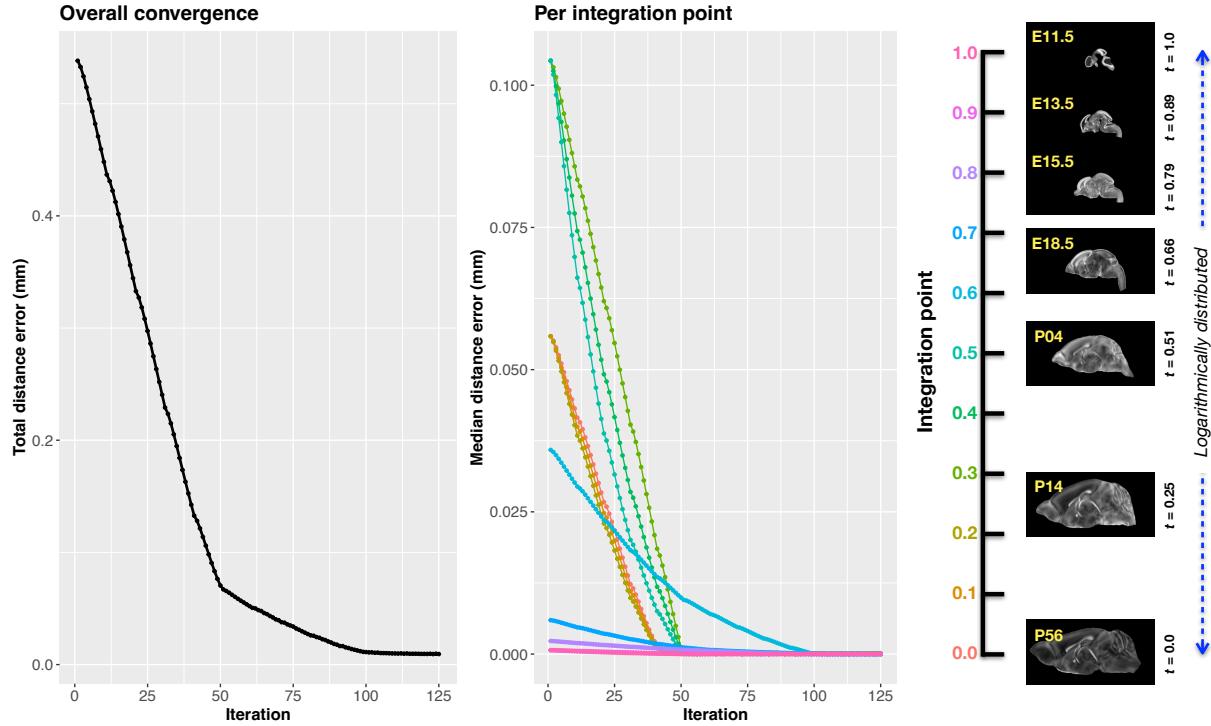


Figure 3: Convergence of the optimization of the velocity field for describing the transformation through the developmental stages from E11.5 through P56.

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

230 The normalized time point scalar value for each atlas/point-set in the temporal domains [0, 1]
 231 was also defined. Given the increasingly larger gaps in the postnatal timepoint sampling, we
 232 made two adjustments. Based on known mouse brain development, we used 28 days for the
 233 P56 data. We then computed the log transform of the adjusted set of time points prior to

²³⁴ normalization between 0 and 1 (see the right side of Figure 3). This log transform, as part
²³⁵ of the temporal normalization, significantly improved data spacing.

²³⁶ The max number of iterations was set to 200. At each iteration we looped over the 11
²³⁷ integration points. At each integration point, the velocity field estimate was updated by
²³⁸ warping the two immediately adjacent point sets to the integration time point and deter-
²³⁹ mining the regularized displacement field between the two warped point sets. As with any
²⁴⁰ gradient-based descent algorithm, this field was multiplied by a small step size ($\delta = 0.2$)
²⁴¹ before adding to the current velocity field. Using multithreading, each iteration took about
²⁴² six minutes.

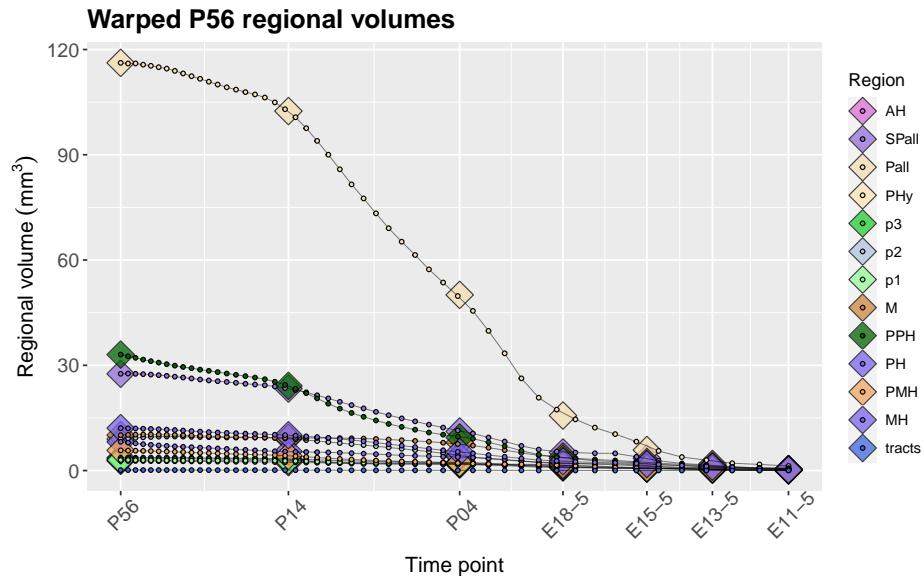


Figure 4: After the velocity field is generated, we can use it to warp the simplified labels of the P56 atlas continuously over the interval [0, 1] and plot the volumes of the atlas regions. Note how they compare with the volumes of the same regions in the other atlases.

²⁴³ Convergence is determined by the average displacement error over each of the integration
²⁴⁴ points. As can be seen in the left panel of Figure 3, convergence occurred around 125
²⁴⁵ iterations when the average displacement error over all integration points is minimized. The
²⁴⁶ median displacement error at each of the integration points also trends towards zero but at
²⁴⁷ different rates. After optimization, we use the velocity field to warp the P56 set of labels
²⁴⁸ to each of the other atlas time points to compare the volumes of the different simplified
²⁴⁹ annotated regions. This is shown in Figure 4.

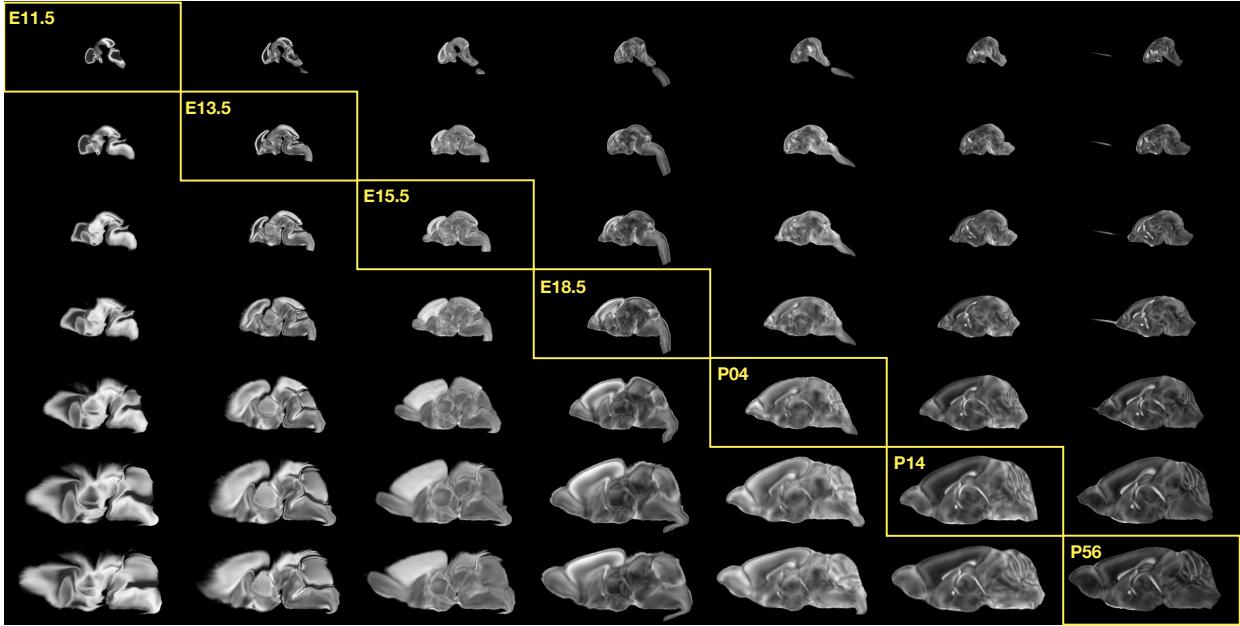
250 **The DevCCF transform model**

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.

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

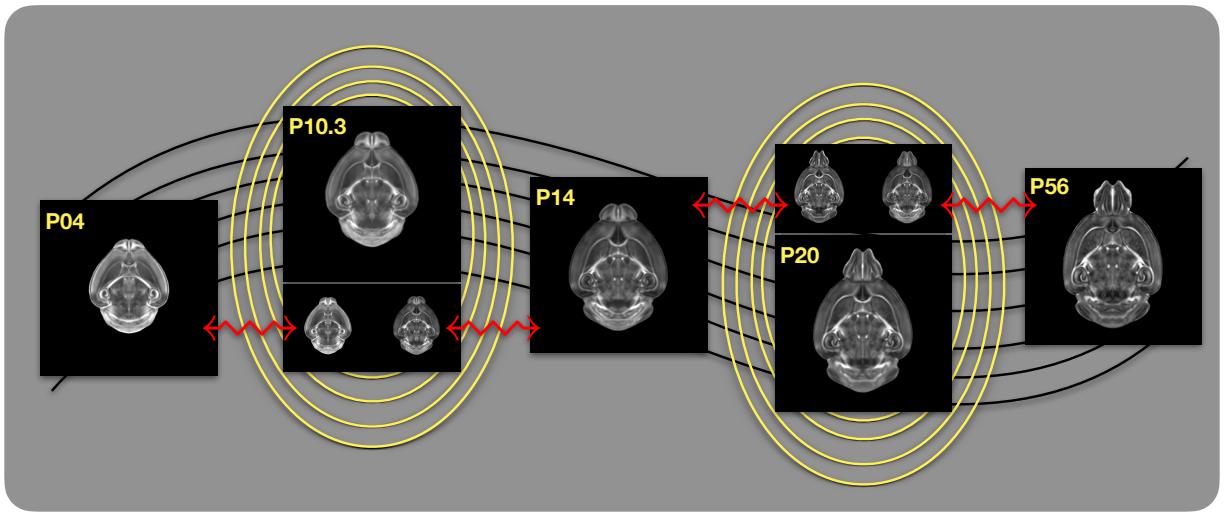


Figure 6: Illustration of the use of the velocity flow model for creating virtual templates at continuous time points not represented in one of the existing DevCCF time points. For example, FA templates at time point P10.3 and P20 can be generated by warping the existing temporally adjacent developmental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

263 **Discussion**

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

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

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

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

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

³⁰⁴ **Methods**

³⁰⁵ The following methods are all available as part of the ANTsX ecosystem with analogous
³⁰⁶ elements existing in both ANTsR (ANTs in R) and ANTsPy (ANTs in Python) with and
³⁰⁷ ANTs/ITK C++ core. However, most of the development for the work described below was
³⁰⁸ performed using ANTsPy. For equivalent calls in ANTsR, please see the ANTsX tutorial at
³⁰⁹ <https://tinyurl.com/antsxtutorial>.

³¹⁰ **Preprocessing: bias field correction and denoising**

³¹¹ As in human studies, bias field correction and image denoising are standard preprocessing
³¹² steps in improving overall image quality in mouse brain images. The bias field, a gradual
³¹³ spatial intensity variation in images, can arise from various sources such as magnetic field in-
³¹⁴ homogeneity or acquisition artifacts, leading to distortions that can compromise the quality
³¹⁵ of brain images. Correcting for bias fields ensures a more uniform and consistent repre-
³¹⁶ sentation of brain structures, enabling accurate quantitative analysis. Additionally, brain
³¹⁷ images are often susceptible to various forms of noise, which can obscure subtle features
³¹⁸ and affect the precision of measurements. Denoising techniques help mitigate the impact
³¹⁹ of noise, enhancing the signal-to-noise ratio and improving the overall image quality. The
³²⁰ well-known N4 bias field correction algorithm²⁵ has its origins in the ANTs toolkit which
³²¹ was implemented and introduced into the ITK toolkit. Similarly, ANTsX contains an im-
³²² plementation of a well-performing patch-based denoising technique⁵³ and is also available as
³²³ an image filter to the ITK community.

³²⁴ **ANTsXNet mouse brain applications**

³²⁵ *General notes regarding deep learning training.*

³²⁶ All network-based approaches described below were implemented and organized in the
³²⁷ ANTsXNet libraries comprising Python (ANTsPyNet) and R (ANTsRNet) analogs using the
³²⁸ Keras/Tensorflow libraries available as open-source in ANTsX GitHub repositories. For the

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

332 Intensity-based data augmentation consisted of randomly added noise based on
333 ITK functionality, simulated bias fields based on N4 bias field modeling, and his-
334 togram warping for mimicking well-known MRI intensity nonlinearities.^{26,55} These
335 augmentation techniques are available in ANTsXNet (only ANTsPyNet versions are
336 listed): simulated bias field: `antspynet.simulate_bias_field(...)`, image noise:
337 `antspyhet.add_noise_to_image(...)`, and MRI intensity nonlinear characteriza-
338 tion: `antspynet.histogram_warp_image_intensities(...)`. Shape-based data
339 augmentation used both random linear and nonlinear deformations. This func-
340 tionality is also instantiated within ANTsXNet in terms of random spatial warping:
341 `antspynet.randomly_transform_image_data(...)`.

342 For all GPU training, we used Python scripts for creating custom batch generators. As such
343 batch generators tend to be application-specific, we store them in a separate GitHub reposi-
344 tory for public availability (<https://github.com/ntustison/ANTsXNetTraining>). In terms of
345 GPU hardware, all training was done on a DGX (GPUs: 4X Tesla V100, system memory:
346 256 GB LRDIMM DDR4).

347 *Brain extraction.*

348 Similar to human neuroimage processing, brain extraction is a crucial preprocessing step for
349 accurate brain mapping. Within ANTsXNet, we have created several deep learning networks
350 for brain extraction for several image modalities (e.g., T1, FLAIR, fractional anisotropy).
351 Similarly, for the developmental brain atlas work¹⁵ we developed similar functionality for
352 mouse brains of different modalities and developmental age. All networks use a conven-
353 tional 2-D U-net architecture⁵⁶ and perform prediction in a slice-wise fashion given the
354 limitations of the acquisition protocols (e.g., missing slices, slice thickness). Currently,
355 coronal and sagittal networks are available for both E13.5 and E15.5 data and coronal
356 network for T2-weighted MRI. In ANTsPyNet, this functionality is available in the pro-
357 gram `antspynet.mouse_brain_extraction(...)`. Even when physical brain extraction is

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

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

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

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

368 Volumetric gene expression slice data was collated into 3-D volumes. Prior to mapping
369 this volume to the corresponding structural data and, potentially, to the appropriate tem-
370 plate, alignment was improved using deformable registration on contiguous slices. How-
371 ever, one of the complications associated with these image data was the unknown num-
372 ber of missing slices, the number of consecutive missing slices, and the different locations
373 of these missing slices. To handle this missing data problem, we found that data in-
374 terpolation using the B-spline approximation algorithm cited earlier⁴⁵ (ANTsPy function:
375 `ants.fit_bspline_object_to_scattered_data(...)`). This provided sufficient data in-
376 terpolation fidelity to perform continuous slicewise registration. Other possible variants that
377 were considered but deemed unnecessary was performing more than one iteration cycling
378 through data interpolation and slicewise alignment. The other possibility was incorporating
379 the super-resolution technique described earlier. But again, our data did not require these
380 additional steps.

381 **Image registration**

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

384 els for biological modeling^{40,51} and has proven capable of excellent performance.^{41,60} Various
385 parameter sets targeting specific applications have been packaged with the different
386 ANTsX platforms, specifically ANTs, ANTsPy, and ANTsR.²⁶ In ANTsPy, the function
387 `ants.registration(...)` is used to register a pair of images or a pair of image sets where
388 `type_of_transform` is a user-specified option that invokes a specific parameter set. For
389 example `type_of_transform='antsRegistrationSyNQuick[s]'` is an oft-used parameter
390 set.

391 Initially, linear optimization is initialized with center of (intensity) mass alignment typically
392 followed by optimization of both rigid and affine transforms using the mutual information
393 similarity metric. This is followed by diffeomorphic deformable alignment using symmetric
394 normalization (SyN) with Gaussian⁴⁰ or B-spline regularization⁵¹ where the forward trans-
395 form is invertible and differentiable. The similarity metric employed at this latter stage
396 is typically either neighborhood cross-correlation or mutual information. Note that these
397 parameter sets are robust to input image type (i.e., LSFM, Nissl staining, and the various
398 MRI modalities) and are adaptable to mousing image geometry scaling. Further details can
399 be found in the various documentation sources for these ANTsX packages.

400 **Template generation**

401 ANTsX provides functionality for constructing templates from a set (or multi-modal sets) of
402 input images as originally described⁴⁴ and recently used to create the DevCCF templates.¹⁵
403 An initial template estimate is constructed from an existing subject image or a voxelwise
404 average derived from a rigid pre-alignment of the image population. Pairwise registration
405 between each subject and the current template estimate is performed using the Symmetric
406 Normalization (SyN) algorithm.⁴⁰ The template estimate is updated by warping all subjects
407 to the space of the template, performing a voxelwise average, and then performing a “shape
408 update” of this latter image by warping it by the average inverse deformation, thus yielding
409 a mean image of the population in terms of both intensity and shape.

410 **Continuous developmental velocity flow transformation model**

411 Given multiple, linearly or non-linearly ordered point sets where individual points across are
412 in one-to-one correspondence, we developed an approach for generating a velocity flow trans-
413 formation model to describe a time-varying diffeomorphic mapping as a variant of the inexact
414 landmark matching solution. Integration of the resulting velocity field can then be used to
415 describe the displacement between any two time points within this time-parameterized do-
416 main. Regularization of the sparse correspondence between point sets is performed using a
417 generalized B-spline scattered data approximation technique,⁴⁵ also developed by the ANTsX
418 developers and contributed to ITK.

419 To apply this methodology to the developmental templates,¹⁵ we coalesced the manual par-
420 cellations of the developmental templates into 26 common anatomical regions (13 per hemi-
421 sphere). We then used these regions to generate invertible transformations between succe-
422 ssive time points. Specifically each label was used to create a pair of single region images
423 resulting in 26 pairs of “source” and “target” images. The multiple image pairs were used
424 to iteratively estimate a diffeomorphic pairwise transform. Given the seven atlases E11.5,
425 E13.5, E15.5, E18.5, P4, P14, and P56, this resulted in 6 sets of transforms between succe-
426 ssive time points. Given the relative sizes between atlases, on the order of 10^6 points were
427 randomly sampled labelwise in the P56 template space and propagated to each successive
428 atlas providing the point sets for constructing the velocity flow model. Approximately 125
429 iterations resulted in a steady convergence based on the average Euclidean norm between
430 transformed point sets. Ten integration points were used and point sets were distributed
431 along the temporal dimension using a log transform for a more evenly spaced sampling.

432 **Visualization**

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

437 **Data availability.** All data and software used in this work are publicly available. The
438 DevCCF atlas is available at <https://kimlab.io/brain-map/DevCCF/>. ANTsPy, ANTsR,
439 ANTsPyNet, and ANTsRNet are available through GitHub at the ANTsX Ecosystem
440 (<https://github.com/ANTsX>). A GitHub repository specific to the work discussed in the
441 manuscript was created and is available at <https://github.com/ntustison/DevCCF-Velocity->
442 **Flow.**

443 **References**

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

- 468 13. Wang, Q. *et al.* The allen mouse brain common coordinate framework: A 3D reference
469 atlas. *Cell* **181**, 936–953.e20 (2020).
- 470 14. Johnson, G. A. *et al.* Waxholm space: An image-based reference for coordinating
471 mouse brain research. *Neuroimage* **53**, 365–72 (2010).
- 472 15. Kronman, F. A. *et al.* Developmental mouse brain common coordinate framework.
473 *bioRxiv* (2023) doi:[10.1101/2023.09.14.557789](https://doi.org/10.1101/2023.09.14.557789).
- 474 16. Oguz, I., Zhang, H., Rumple, A. & Sonka, M. RATS: Rapid automatic tissue segmen-
475 tation in rodent brain MRI. *J Neurosci Methods* **221**, 175–82 (2014).
- 476 17. Sawiak, S. J., Picq, J.-L. & Dhenain, M. Voxel-based morphometry analyses of in
477 vivo MRI in the aging mouse lemur primate. *Front Aging Neurosci* **6**, 82 (2014).
- 478 18. Ashburner, J. SPM: A history. *Neuroimage* **62**, 791–800 (2012).
- 479
- 480 19. Modat, M. *et al.* Fast free-form deformation using graphics processing units. *Comput
481 Methods Programs Biomed* **98**, 278–84 (2010).
- 482 20. Tyson, A. L. *et al.* Accurate determination of marker location within whole-brain
483 microscopy images. *Sci Rep* **12**, 867 (2022).
- 484 21. Pallast, N. *et al.* Processing pipeline for atlas-based imaging data analysis of struc-
485 tural and functional mouse brain MRI (AIDAmri). *Front Neuroinform* **13**, 42 (2019).
- 486 22. Jenkinson, M., Beckmann, C. F., Behrens, T. E. J., Woolrich, M. W. & Smith, S. M.
487 FSL. *Neuroimage* **62**, 782–90 (2012).
- 488 23. Yeh, F.-C., Wedeen, V. J. & Tseng, W.-Y. I. Generalized q-sampling imaging. *IEEE
489 Trans Med Imaging* **29**, 1626–35 (2010).
- 490 24. Jorge Cardoso, M. *et al.* STEPS: Similarity and truth estimation for propagated
491 segmentations and its application to hippocampal segmentation and brain parcelation.
Med Image Anal **17**, 671–84 (2013).

- 492 25. Tustison, N. J. *et al.* N4ITK: Improved N3 bias correction. *IEEE Trans Med Imaging*
493 **29**, 1310–20 (2010).
- 494 26. Tustison, N. J. *et al.* The ANTsX ecosystem for quantitative biological and medical
495 imaging. *Sci Rep* **11**, 9068 (2021).
- 496 27. Goubran, M. *et al.* Multimodal image registration and connectivity analysis for inte-
497 gration of connectomic data from microscopy to MRI. *Nat Commun* **10**, 5504 (2019).
- 498 28. Celestine, M., Nadkarni, N. A., Garin, C. M., Bougacha, S. & Dhenain, M. Sammba-
499 MRI: A library for processing SmAll-MaMmal BrAin MRI data in python. *Front
Neuroinform* **14**, 24 (2020).
- 500 29. Ioanas, H.-I., Marks, M., Zerbi, V., Yanik, M. F. & Rudin, M. An optimized regis-
501 tration workflow and standard geometric space for small animal brain imaging. *Neuro-
image* **241**, 118386 (2021).
- 502 30. Cox, R. W. AFNI: What a long strange trip it's been. *Neuroimage* **62**, 743–7 (2012).
503
- 504 31. Ni, H. *et al.* A robust image registration interface for large volume brain atlas. *Sci
505 Rep* **10**, 2139 (2020).
- 506 32. Jin, M. *et al.* SMART: An open-source extension of WholeBrain for intact mouse
507 brain registration and segmentation. *eNeuro* **9**, (2022).
- 508 33. Fürth, D. *et al.* An interactive framework for whole-brain maps at cellular resolution.
509 *Nat Neurosci* **21**, 139–149 (2018).
- 510 34. Negwer, M. *et al.* FriendlyClearMap: An optimized toolkit for mouse brain mapping
511 and analysis. *Gigascience* **12**, (2022).
- 512 35. Klein, S., Staring, M., Murphy, K., Viergever, M. A. & Pluim, J. P. W. Elastix: A
513 toolbox for intensity-based medical image registration. *IEEE Trans Med Imaging* **29**,
196–205 (2010).

- 514 36. Carey, H. *et al.* DeepSlice: Rapid fully automatic registration of mouse brain imaging
515 to a volumetric atlas. *Nat Commun* **14**, 5884 (2023).
- 516 37. Bajcsy, R. & Broit, C. Matching of deformed images. in *Sixth International Conference on Pattern Recognition (ICPR'82)* 351–353 (1982).
- 517
- 518 38. Bajcsy, R. & Kovacic, S. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing* **46**, 1–21 (1989).
- 519
- 520 39. Gee, J., Sundaram, T., Hasegawa, I., Uematsu, H. & Hatabu, H. Characterization
521 of regional pulmonary mechanics from serial magnetic resonance imaging data. *Acad Radiol* **10**, 1147–52 (2003).
- 522
- 523 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).
- 524
- 525 41. Klein, A. *et al.* Evaluation of 14 nonlinear deformation algorithms applied to human
brain MRI registration. *Neuroimage* **46**, 786–802 (2009).
- 526
- 527 42. Murphy, K. *et al.* Evaluation of registration methods on thoracic CT: The EMPIRE10
challenge. *IEEE Trans Med Imaging* **30**, 1901–20 (2011).
- 528
- 529 43. Baheti, B. *et al.* The brain tumor sequence registration challenge: Establishing corre-
spondence between pre-operative and follow-up MRI scans of diffuse glioma patients.
(2021).
- 530
- 531 44. Avants, B. B. *et al.* The optimal template effect in hippocampus studies of diseased
populations. *Neuroimage* **49**, 2457–66 (2010).
- 532
- 533 45. Tustison, N. J. & Amini, A. A. Biventricular myocardial strains via nonrigid regis-
tration of anatomical NURBS model [corrected]. *IEEE Trans Med Imaging* **25**, 94–112
(2006).
- 534
- 535 46. Wang, H. *et al.* Multi-atlas segmentation with joint label fusion. *IEEE Trans Pattern
Anal Mach Intell* **35**, 611–23 (2013).

- 536 47. 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).
- 537
- 538 48. Tustison, N. J., Yang, Y. & Salerno, M. Advanced normalization tools for cardiac mo-
tion correction. in *Statistical atlases and computational models of the heart - imaging*
and modelling challenges (eds. Camara, O. et al.) vol. 8896 3–12 (Springer Interna-
tional Publishing, 2015).
- 539
- 540 49. McCormick, M., Liu, X., Jomier, J., Marion, C. & Ibanez, L. ITK: Enabling repro-
ducible research and open science. *Front Neuroinform* **8**, 13 (2014).
- 541
- 542 50. 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).
- 543
- 544 51. Tustison, N. J. & Avants, B. B. Explicit B-spline regularization in diffeomorphic image
registration. *Front Neuroinform* **7**, 39 (2013).
- 545
- 546 52. Tustison, N. J. *et al.* Optimal symmetric multimodal templates and concatenated
random forests for supervised brain tumor segmentation (simplified) with ANTsR.
Neuroinformatics **13**, 209–25 (2015).
- 547
- 548 53. 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).
- 549
- 550 54. Klein, A. *et al.* Evaluation of volume-based and surface-based brain image registration
methods. *Neuroimage* **51**, 214–20 (2010).
- 551
- 552 55. 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).
- 553
- 554 56. Falk, T. *et al.* U-net: Deep learning for cell counting, detection, and morphometry.
Nat Methods **16**, 67–70 (2019).
- 555

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