

1

2 The ANTsX Ecosystem for Spatiotemporal 3 Mapping of the Mouse Brain

4 Nicholas J. Tustison¹, Min Chen², Fae N. Kronman³, Jeffrey T. Duda², Clare Gamlin⁴,
5 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 version 3 (CCFv3), and the ability to map to such reference
₂₂ atlases. The Advanced Normalization Tools Ecosystem (ANTsX) is a comprehensive open-
₂₃ source software image analysis toolkit, including template building and mapping functional-
₂₄ ity, with applicability to multiple organ systems, modalities, and animal species. Herein, we
₂₅ illustrate the utility of ANTsX for generating precision spatial mappings of the mouse brain
₂₆ of different developmental ages including the prerequisite preprocessing steps. Additionally,
₂₇ as a further illustration of ANTsX capabilities, we use these publicly available mouse brain
₂₈ atlases to generate a velocity flow-based mapping encompassing the entire developmental
₂₉ trajectory, which we also make available to the public.

³⁰ Introduction

³¹ Over the past two decades there have been significant advancements in mesoscopic anal-
³² ysis of the mouse brain. It is now possible to track single cell neurons in 3-D across full
³³ mouse brains,¹ observe whole brain developmental changes on a cellular level,² associate
³⁴ brain regions and tissues with their genetic composition,³ and locally characterize neural
³⁵ connectivity.⁴ Much of this scientific achievement has been made possible due to break-
³⁶ throughs in high resolution imaging techniques that permit submicron, 3-D imaging of whole
³⁷ mouse brains. Associated research techniques such as micro-optical sectioning tomography,⁶
³⁸ tissue clearing,^{1,7} spatial transcriptomics⁹ are all well-utilized in the course of scientific in-
³⁹ vestigations of mesoscale relationships in the mouse brain.

⁴⁰ An important component of this research is the ability to map the various image data to
⁴¹ anatomical reference frames¹¹ for inferring spatial relationships between structures, cells,
⁴² and genetics. This has motivated the development of detailed structural image atlases of the
⁴³ mouse brain. Notable examples include the Allen Brain Atlas and Coordinate Frameworks¹³
⁴⁴ and the Waxholm Space.¹⁴ Despite the significance of these contributions, challenges still
⁴⁵ exist in large part due to the wide heterogeneity in associated study-specific image data. For
⁴⁶ example, variance in the acquisition methods can introduce artifacts such as tissue distor-
⁴⁷ tion, holes, bubbles, folding, tears, and missing slices. These severely complicate assumed
⁴⁸ correspondence for conventional registration approaches.

⁴⁹ To address such challenges, several software packages have been developed over the years
⁵⁰ comprising solutions of varying comprehensibility, sophistication, and availability. An early
⁵¹ contribution to the community was the Rapid Automatic Tissue Segmentation (RATS)
⁵² package¹⁵ for brain extraction which is available upon request. Of the publicly available
⁵³ packages, most, if not all have well-established package dependencies originally developed
⁵⁴ on human brain data. SPMMouse,¹⁶ for example, is based on the well-known Statistical
⁵⁵ Parametric Mapping (SPM) software package.¹⁷ The automated mouse atlas propagation
⁵⁶ (aMAP) tool is largely a front-end for the NiftyReg image registration package¹⁸ applied
⁵⁷ to mouse data which is currently available as a Python module.¹⁹ NiftyReg is also used by
⁵⁸ the Atlas-based Imaging Data Analysis (AIDA) MRI pipeline²⁰ as well as the Multi Atlas

59 Segmentation and Morphometric Analysis Toolkit (MASMAT). Whereas the former also
60 incorporates the FMRIB Software Library (FSL)²¹ for brain extraction and DSISTudio²²
61 for DTI processing, the latter uses NiftySeg and multi-consensus labeling tools²³ for brain
62 extraction and parcellation. In addition, MASMAT incorporates N4 bias field correction²⁴
63 from the Advanced Normalization Tools Ecosystem (ANTsX)²⁵ as do the packages Multi-
64 modal Image Registration And Connectivity anaLysis (MIRACL),²⁶ Saamba-MRI,²⁷ and
65 Small Animal Magnetic Resonance Imaging (SAMRI).²⁸ However, whereas Saamba-MRI uses
66 AFNI²⁹ for image registration; MIRACL, SAMRI, and BrainsMap³⁰ all use ANTsX registra-
67 tion tools. Other packages use landmark-based approaches to image registration including
68 SMART—³¹an R package for semi-automated landmark-based registration and segmenta-
69 tion of mouse brain based on WholeBrain.³² FriendlyClearMap³³ uses the landmark-based
70 registration functionality of Elastix.³⁴ Finally, the widespread adoption of deep learning tech-
71 niques has also influenced development in mouse brain imaging methodologies. For example,
72 if tissue deformations are not considered problematic for a particular dataset, DeepSlice can
73 be used to determine affine mappings³⁵ with the optimal computational efficiency associated
74 with neural networks.

75 The ANTsX Ecosystem

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

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</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>).

general data approximation, and deep learning networks specifically trained for mouse data (see Table 1). The collective use of the toolkit has demonstrated superb performance in multiple application areas (e.g., consensus labeling,⁴³ brain tumor segmentation,⁴⁴ and cardiac motion estimation).⁴⁵ Importantly, ANTs is built on the Insight Toolkit (ITK)⁴⁶ deriving benefit from a very capable open-source community of scientists and programmers as well as providing a visible, open-source venue for algorithmic contributions.

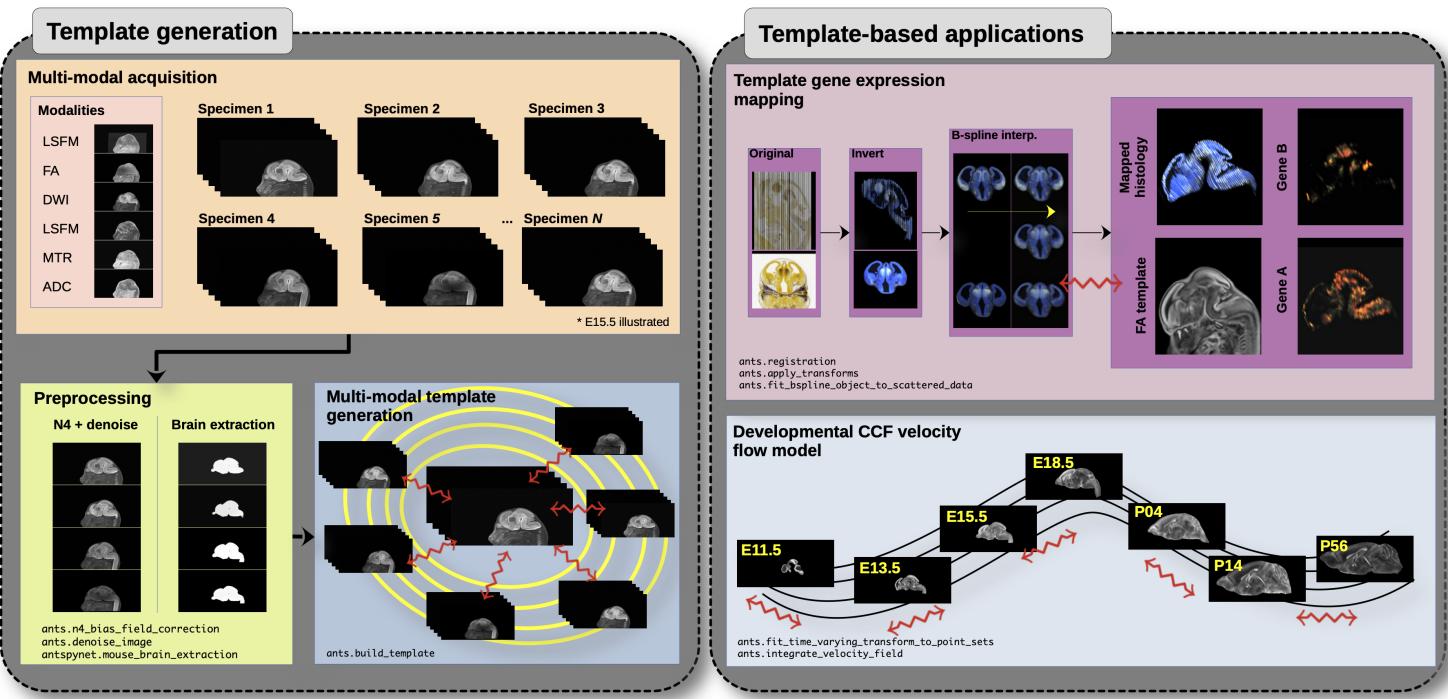


Figure 1: Illustration of a mouse brain template generation workflow and related template-based applications demonstrating the utility of different ANTsX tools. After imaging acquisition of the study population, various preprocessing steps are applied to the imaging data such as bias correction, denoising, and brain extraction as dictated by the needs of the study protocol. Not shown is the possibility of template symmetrization by contralaterally flipping the image data associated with each specimen. In the case of the DevCCF, applications include gene expression mapping and the associated velocity flow model for pseudo-template generation.

Recently, the developmental common coordinate framework (DevCCF) was introduced to the mouse brain research community as a public resource.⁴⁷ These symmetric atlases, comprising both multimodal image data and anatomical segmentations defined by developmental ontology, span the mouse embryonic days (E) 11.5, E13.5, E15.5, E18.5 and postnatal day

97 (P) 4, P14, and P56. Modalities include at least four MRI contrasts and light sheet flores-
98 cence microscopy (LSFM) per developmental stage. Gene expression and other cell type
99 data were mapped to the corresponding developmental time point to guide the associated
100 anatomical parcellations. The P56 template was integrated with the Allen CCFv3 to further
101 increase the practical utility of the DevCCF. These processes, specifically template gener-
102 ation and multi-modal image mapping, were performed using ANTsX functionality in the
103 presence of previously noted image mapping difficulties (e.g., missing slices, tissue distortion)
104 illustrated in Figure 1.

105 Given the temporal gaps in the discrete set of developmental atlases, we augment the tem-
106 plate generation explanation previously given⁴⁷ from a developer’s perspective. We hope that
107 this will provide additional information for the interested reader for potential future template
108 generation. Related, we also provide a complementary strategy for inferring correspondence
109 and mapping information within the temporally continuous domain spanned and sampled by
110 the existing set of embryonic and postnatal atlas brains of the DevCCF. Recently developed
111 ANTsX functionality include the generation of a diffeomorphic velocity flow transformation
112 model⁴⁸ spanning developmental stages where mappings between any two continuous time
113 points within the span bounded by the E11.5 and P56 atlases is determined by integration
114 of the generated time-varying velocity field.⁴⁹ Such transformations permit the possibility of
115 “pseudo” templates generated between available developmental stages.

₁₁₆ **Results**

₁₁₇ **Template building**

₁₁₈ Template building using ANTsX tools was first described in.⁵⁰ Subsequently, multi-modal and
₁₁₉ symmetrical variants were more explicitly described as part of the brain tumor segmentation
₁₂₀ approach.⁵¹

₁₂₁ **The DevCCF Velocity Flow Model**

₁₂₂ To continuously link the DevCCF atlases, a velocity flow model was constructed using Dev-
₁₂₃ CCF derived data and ANTsX functionality available in both ANTsR and ANTsPy. Al-
₁₂₄ though many implementations optimize variations of this transformation model (and others)
₁₂₅ using various image intensity similarity metrics, we opted to implement a separate de-
₁₂₆ termination of iterative correspondence and transformation optimization. This decision was
₁₂₇ based on existing ANTsX functionality and wanting complementary utility for the toolkit.

₁₂₈ ANTsX, being built on top of ITK, uses an ITK image data structure for the 4-D velocity
₁₂₉ field where each voxel contains the x , y , z components of the field at that point. Field
₁₃₀ regularization is provided by a novel B-spline scattered data approximation technique⁵² which
₁₃₁ permits individual point-based weighting. Both field regularization and integration of the
₁₃₂ velocity field are built on ITK functions written by ANTsX developers.

₁₃₃ The optimized velocity field described here is of size [256, 182, 360] (or $50\mu\text{m}$ isotropic) $\times 11$
₁₃₄ integration points for a total compressed size of a little over 2 GB. This choice represented
₁₃₅ weighing the trade-off between tractability, portability, and accuracy. However, all data
₁₃₆ and code to reproduce the results described are available in a dedicated GitHub repository
₁₃₇ (<https://github.com/ntustison/DevCCF-Velocity-Flow>).

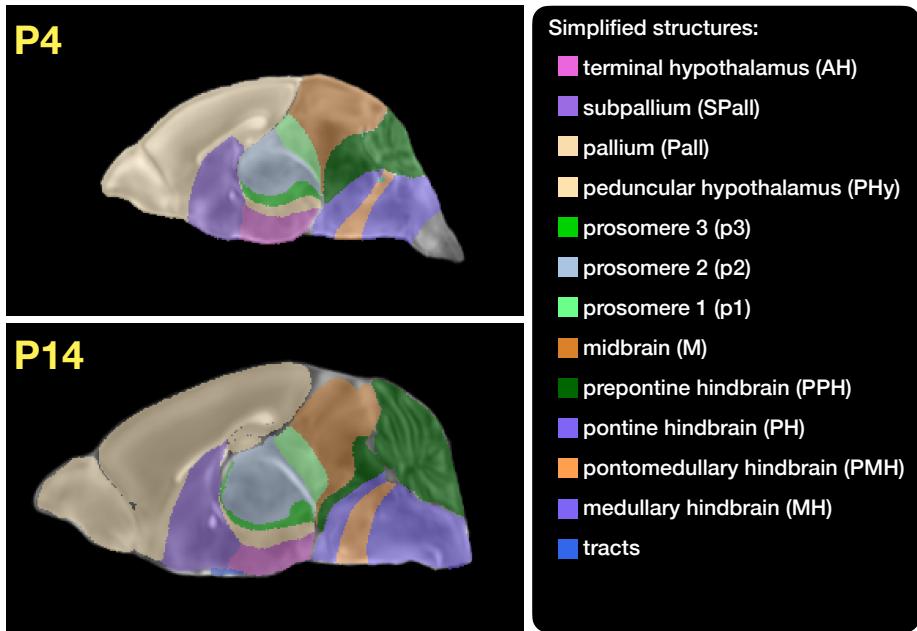


Figure 2: Annotated regions representing common labels across developmental stages which are illustrated for both P4 and P14.

¹³⁸ Data preparation

¹³⁹ Labeled annotations are available as part of the original DevCCF and reside in the space
¹⁴⁰ of each developmental template which range in resolution from $31.5 - 50\mu\text{m}$. Across all atlases,
¹⁴¹ the total number of labels exceeded 2500 without taken into account per hemispherical
¹⁴² enumeration. From this set of labels, there were a common set of 24 labels (12 per hemisphere)
¹⁴³ across all atlases that were used for optimization and evaluation. These regions are
¹⁴⁴ illustrated for the P4 and P14 stages in Figure 2.

¹⁴⁵ Prior to velocity field optimization, the data was rigidly transformed to a common
¹⁴⁶ space. Using the centroids for the common label set of each CCFDev atlas, the ANTsPy
¹⁴⁷ `ants.fit_transform_to_paired_points(...)` function was used to warp each atlas
¹⁴⁸ to the space of the P56 atlas and then downsampled to $50\mu\text{m}$ isotropic resolution. In
¹⁴⁹ order to determine the common point sets across stages, the multi-metric capabilities
¹⁵⁰ of `ants.registration(...)` were used. Instead of performing intensity-based pairwise
¹⁵¹ registration directly on these multi-label images, each label was used to construct a separate
¹⁵² fixed and moving image pair resulting in a multi-metric registration optimization scenario

153 involving 24 binary image pairs (each label weighted equally) for optimizing correspondence
154 between neighboring atlases using the mean squares metric and the SyN transform.

155 To provide the common point sets across all seven developmental atlases, the label boundaries
156 and whole regions were sampled in the P56 atlas and then propagated to each atlas using
157 the transformations derived from the pairwise registrations. Based on previous experience
158 as both the developers of users of these tools, we selected a sampling rate of 10% for the
159 contour points and 1% for the regional points for a total number of points being per atlas
160 being 173303 ($N_{contour} = 98151$ and $N_{region} = 75152$). Boundary points were weighted twice
161 as those of regional points for the B-spline data approximation optimization.

162 Optimization

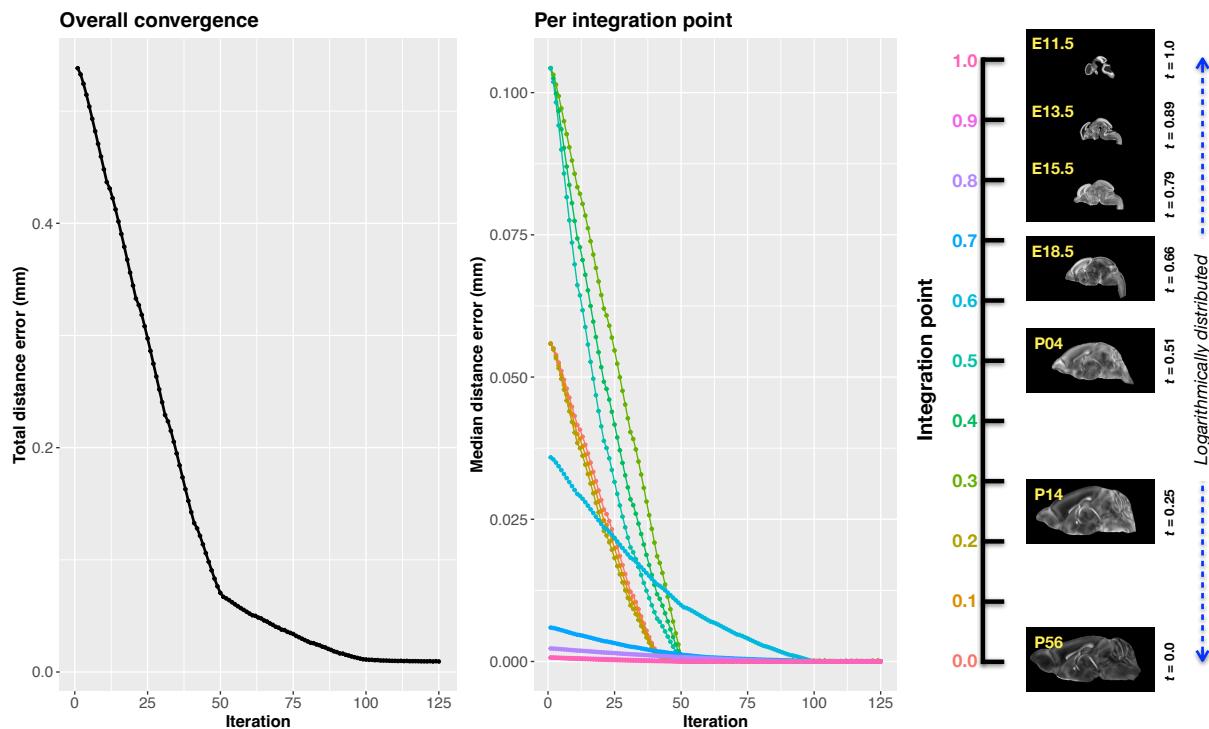


Figure 3: Convergence of the optimization of the velocity field for describing the transformation through the developmental stages from E11.5 through P56.

163 `fit_time_varying_transform_to_point_sets(...)` from the ANTsPy package was used
164 to optimize the velocity field. Input comprised the seven corresponding point sets and
165 their associated weight values, the selected number of integration points for the velocity

field ($N = 11$), and the parameters defining the geometry of the spatial dimensions of the velocity field (same as the downsampled P56 atlas noted above). In addition, the normalized time point for each atlas/point-set was also defined. Given the increasingly larger gaps in the postnatal timepoint sampling, we made two adjustments. Based on known mouse brain development, we used 28 days for the P56 data. We then computed the log transform of the adjusted set of time points prior to normalization between 0 and 1 (see the right side of Figure 3.) This log transform, as part of the temporal normalization, significantly improved data spacing.

The max number of iterations was set to 200. 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.

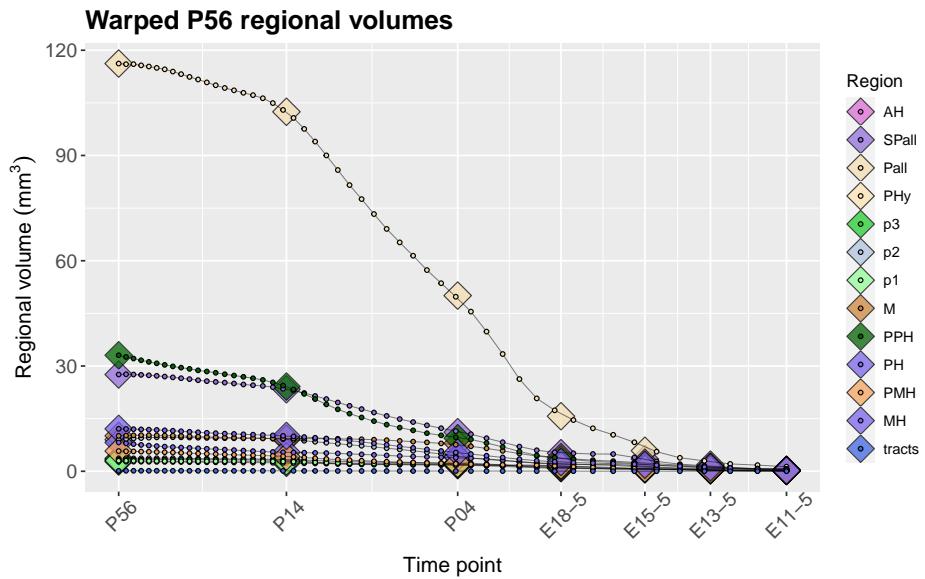


Figure 4: After the velocity field is generated, we can use it to warp the simplified labels of the P56 atlas continuously over the interval [0, 1] and plot the volumes of the atlas regions. Note how they compare with the volumes of the same regions in the other atlases.

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 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.

188 The DevCCF transform model

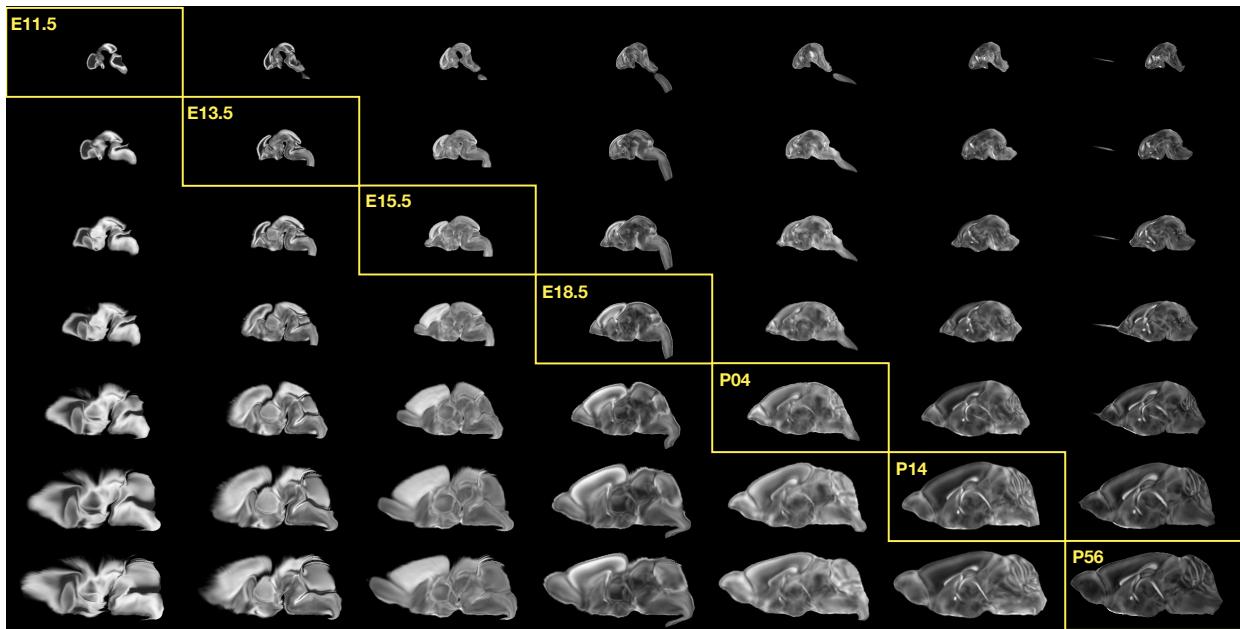


Figure 5: Mid-sagittal visualization of the effects of the transformation model in warping every developmental stage to the time point of every other developmental stage. The original images are located along the diagonal. Columns correspond to the warped original image whereas the rows represent the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

Once optimized, the resulting velocity field can be used to generate the deformable transform between any two continuous points within the time interval bounded by E11.5 and P56. So, for example, one can transform each atlas to the space of every other atlas. This is illustrated in Figure 5 where we render a mid-sagittal location for each atlas and the results of warping every atlas to that space.

One potential application for this particular transformation model is facilitating the con-

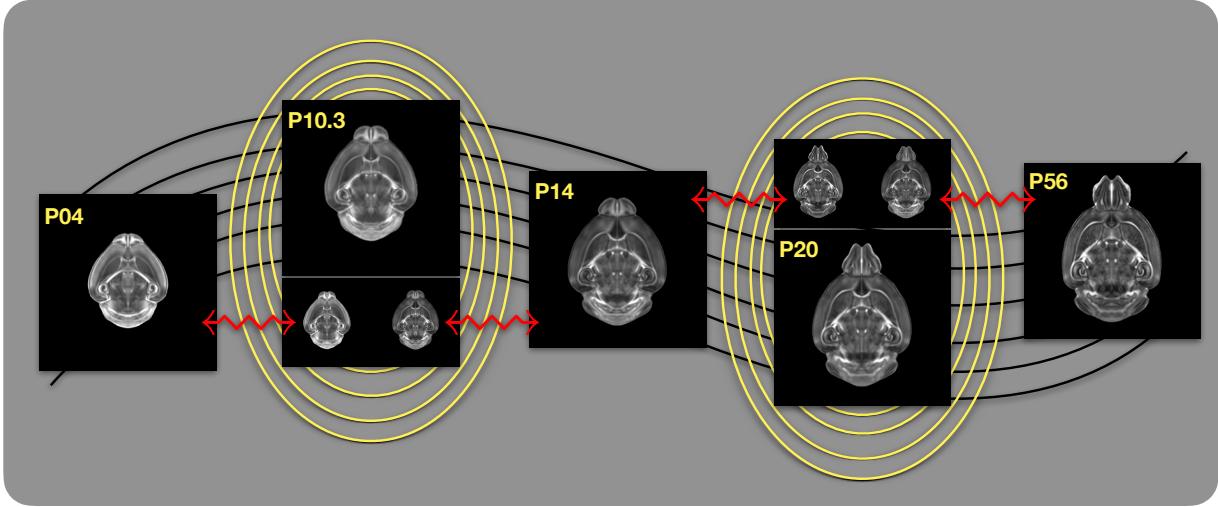


Figure 6: Illustration of the use of the velocity flow model for creating pseudo-templates at continuous time points not represented in one of the existing developmental stages. For example, FA templates at time point P10.3 and P20 can be generated by warping the existing temporally adjacent developmental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

195 construction of pseudo-templates in the temporal gaps of the DevCCF. This is illustrated in
 196 Figure 6 where we used the optimized velocity field to construct pseudo-templates at time
 197 point P10.3 and P20—arbitrarily chosen simply to demonstrate the concept. After situating
 198 these time points within the normalized time point interval, the existing adjacent DevCCF
 199 atlases on either side can be warped to the desired time point. A subsequent call to one
 200 of the ANTsX template building functions then permits the construction of the template
 201 at that time point. Note that both of these usage examples can be found on the GitHub
 202 repository given above.

203 **Discussion**

204 In this study, we employed precision mapping techniques and high-resolution image acquisition
205 to investigate the spatial organization of gene activity and their interactions in the mouse
206 brain. Leveraging the Advanced Normalization Tools Ecosystem (ANTsX), an open-source
207 software image analysis toolkit, we demonstrated its utility in generating precise spatial
208 mappings of the mouse brain at different developmental ages. The focus of this discussion
209 is on the construction and application of the DevCCF Velocity Flow Model, showcasing its
210 versatility and potential implications.

211 The velocity flow model was constructed to continuously link the developmental atlases,
212 providing a dynamic representation of the mouse brain's transformation across different
213 developmental stages. ANTsX, built on the Insight Segmentation and Registration Toolkit
214 (ITK), facilitated the creation of a 4-D velocity field, employing a B-spline scattered data
215 approximation technique for field regularization. The optimization of the velocity field was
216 achieved through a unique approach, involving iterative correspondence determination and
217 transformation optimization.

218 The choice of a velocity field size of [256, 182, 360] with 11 integration points struck a balance
219 between tractability, portability, and accuracy. The resulting velocity field, though compact,
220 encapsulated the intricate changes across developmental stages, ensuring the representative-
221 ness of the model. The entire process, including data preparation and optimization, is
222 well-documented and accessible through a dedicated GitHub repository.

223 Data preparation involved the transformation of labeled annotations to a common space,
224 enabling the establishment of a common point set across all developmental atlases. The op-
225 timization process included the rigid transformation of data, determination of common point
226 sets, and the subsequent generation of the velocity field. The convergence of the optimization
227 process, illustrated in Figure 3, highlighted the efficiency of the employed methodology.

228 The velocity field was further validated through the warping of simplified labels, providing
229 visual insights into the consistency and accuracy of the model. The availability of the warped
230 P56 atlas volumes, as depicted in Figure 4, enables a comprehensive comparison of annotated

231 regions across developmental stages.

232 The DevCCF transform model, once optimized, offers a powerful tool for generating de-
233 formable transformations between any two continuous points within the developmental time
234 interval. This capability is exemplified in Figure 5, where each atlas is transformed to the
235 space of every other atlas, providing a detailed view of the developmental trajectory.

236 An intriguing application of this transformation model is the creation of pseudo-templates
237 in temporal gaps within the DevCCF. As depicted in Figure 6, the velocity flow model
238 enables the generation of templates at continuous time points not represented in existing
239 developmental stages. This novel application can significantly enhance our understanding of
240 intermediate developmental states.

241 In conclusion, the presented DevCCF Velocity Flow Model, constructed and optimized using
242 ANTsX, represents a valuable contribution to the field of mouse brain mapping. The ability
243 to continuously link developmental atlases, visualize transformations, and generate pseudo-
244 templates opens avenues for in-depth explorations of gene activity and structural-functional
245 relationships during brain development. The accessibility of the methodology and results
246 through the provided GitHub repository encourages further exploration and collaboration
247 within the scientific community.

²⁴⁸ **Methods**

²⁴⁹ The following methods are all available as part of the ANTsX ecosystem with analogous
²⁵⁰ elements existing in both ANTsR (ANTs in R) and ANTsPy (ANTs in Python) with and
²⁵¹ ANTs/ITK C++ core. However, most of the development for the work described below was
²⁵² performed using ANTsPy. For equivalent calls in ANTsR, please see the ANTsX tutorial at
²⁵³ <https://tinyurl.com/antsxtutorial>.

²⁵⁴ **Preprocessing: bias field correction and denoising**

²⁵⁵ As in human studies, bias field correction and image denoising are standard preprocessing
²⁵⁶ steps in improving overall image quality in mouse brain images. The bias field, a gradual
²⁵⁷ spatial intensity variation in images, can arise from various sources such as magnetic field in-
²⁵⁸ homogeneity or acquisition artifacts, leading to distortions that can compromise the quality
²⁵⁹ of brain images. Correcting for bias fields ensures a more uniform and consistent representa-
²⁶⁰ tion of brain structures, enabling accurate quantitative analysis. Additionally, brain images
²⁶¹ are often susceptible to various forms of noise, which can obscure subtle features and affect
²⁶² the precision of measurements. Denoising techniques help mitigate the impact of noise, en-
²⁶³ hancing the signal-to-noise ratio and improving the overall image quality. The well-known
²⁶⁴ N4 bias field correction algorithm²⁴ has its origins in the ANTs toolkit which was imple-
²⁶⁵ mented and introduced into the ITK toolkit. Similarly, ANTsX contains an implementation
²⁶⁶ of a well-performing patch-based denoising technique⁵³ and is also available as a image filter
²⁶⁷ to the ITK community.

²⁶⁸ **ANTsXNet mouse brain applications**

²⁶⁹ *General notes regarding deep learning training.*

²⁷⁰ All network-based approaches described below were implemented and organized in the
²⁷¹ ANTsXNet libraries comprising Python (ANTsPyNet) and R (ANTsRNet) analogs using the
²⁷² Keras/Tensorflow libraries available as open-source in ANTsX GitHub repositories. For the

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

276 Intensity-based data augmentation consisted of randomly added noise based on ITK
277 functionality, simulated bias fields based on N4 bias field modeling, and histogram
278 warping for mimicking well-known MRI intensity nonlinearities.^{25,54} These augmentation
279 techniques are available in ANTsXNet (only ANTsPyNet versions are listed): simulated
280 bias field: `simulate_bias_field(...)`, image noise: `add_noise_to_image(...)`, and
281 MRI intensity nonlinear characterization: `histogram_warp_image_intensities(...)`.
282 Shape-based data augmentation used both random linear and nonlinear deformations. This
283 functionality is also instantiated within ANTsXNet in terms of random spatial warping:
284 `randomly_transform_image_data(...)`.

285 For all GPU training, we used Python scripts for creating custom batch generators. As such
286 batch generators tend to be application-specific, we store them in a separate GitHub reposi-
287 tory for public availability (<https://github.com/ntustison/ANTsXNetTraining>). In terms of
288 GPU hardware, all training was done on a DGX (GPUs: 4X Tesla V100, system memory:
289 256 GB LRDIMM DDR4).

290 *Brain extraction.*

291 Similar to human neuroimage processing, brain extraction is a crucial preprocessing step for
292 accurate brain mapping. Within ANTsXNet, we have created several deep learning networks
293 for brain extraction for several image modalities (e.g., T1, FLAIR, fractional anisotropy).
294 Similarly, for the developmental brain atlas work⁴⁷ we developed similar functionality for
295 mouse brains of different modalities and developmental age. All networks use a conventional
296 2-D U-net architecture⁵⁵ and perform prediction in a slice-wise fashion given the limitations of
297 the acquisition protocols (e.g., missing slices, slice thickness). Currently, coronal and sagittal
298 networks are available for both E13.5 and E15.5 data and coronal network for T2-weighted
299 MRI. In ANTsPyNet, this functionality is available in the program `brain_extraction(...)`.
300 Even when physical brain extraction is performed prior to image acquisition, artifacts, such
301 as bubbles or debris, can complicate subsequent processing. Similar to the brain extraction

302 networks, a 2-D U-net architecture⁵⁵ was created to separate the background and foreground.

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

304 To further enhance the data prior to designing mapping protocols, additional networks were
305 created. A well-performing deep back projection network⁵⁶ was ported to ANTsXNet and
306 expanded to 3-D for various super-resolution applications,⁵⁷ including mouse data. Finally,
307 features of anatomical significance, namely the cerebellum and hemispherical midline were
308 captured in these data using deep learning networks.

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

310 Volumetric gene expression slice data was collated into 3-D volumes using ... (ask Jeff).
311 Prior to mapping this volume to the corresponding structural data and, potentially, to the
312 appropriate template, alignment was improved using deformable registration on contiguous
313 slices. However, one of the complications associated with these image data was the un-
314 known number of missing slices, the number of consecutive missing slices, and the different
315 locations of these missing slices. To handle this missing data problem, we found that data
316 interpolation using the B-spline approximation algorithm cited earlier⁵² (ANTsPy function:
317 `fit_bspline_object_to_scattered_data(...)`). This provided sufficient data interpo-
318 lation fidelity to perform continuous slicewise registration. Other possible variants that
319 were considered but deemed unnecessary was performing more than one iteration cycling
320 through data interpolation and slicewise alignment. The other possibility was incorporating
321 the super-resolution technique described earlier. But again, our data did not require these
322 additional steps.

323 **Image registration**

324 The ANTs registration toolkit is a complex framework permitting highly tailored so-
325 lutions to pairwise image registration scenarios.⁵⁸ It includes innovative transformation
326 models for biological modeling^{39,59} and has proven capable of excellent performance.^{40,60}
327 Various parameter sets targeting specific applications have been packaged with the

328 different ANTsX platforms, specifically ANTs, ANTsPy, and ANTsR.²⁵ In ANTsPy,
329 the function `ants.registration(...)` is used to register two (sets) of images where
330 `type_of_transform` is a user-specified option that invokes a specific parameter set. For
331 example `type_of_transform='antsRegistrationSyNQuick[s]'` is an oft-used parameter
332 set.

333 Initially, linear optimization is initialized with center of (intensity) mass alignment typically
334 followed by optimization of both rigid and affine transforms using the mutual information
335 similarity metric. This was followed by diffeomorphic deformable alignment using symmetric
336 normalization (SyN) with Gaussian³⁹ or B-spline regularization⁵⁹ where the forward trans-
337 form is invertible and differentiable. The similarity metric employed at this latter stage is
338 typically either neighborhood cross-correlation or mutual information similarity metric. Note
339 that these parameter sets are robust to input image type (i.e., LSFM, Nissl staining, and the
340 various MRI modalities) and are adaptable to mousing image geometry scaling. Further
341 details can be found in the various documentation sources for these ANTsX packages.

342 Template generation

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

³⁵² **Continuous developmental velocity flow transformation model**

³⁵³ Given multiple, linearly or non-linearly ordered point sets where individual points across
³⁵⁴ are in one-to-one correspondence, we developed an approach for generating a velocity flow
³⁵⁵ transformation model to describe a time-varying diffeomorphic mapping as a variant of the
³⁵⁶ inexact landmark matching solution of Joshi and Miller.⁴⁸ Integration of the resulting velocity
³⁵⁷ field can then be used to describe the displacement between any two time points within this
³⁵⁸ time-parameterized domain. Regularization of the sparse correspondence between point sets
³⁵⁹ is performed using a generalized B-spline scattered data approximation technique,⁵² also
³⁶⁰ developed by the ANTsX developers and contributed to ITK.

³⁶¹ To apply this methodology to the developmental templates,⁴⁷ we coalesced the manual par-
³⁶² cellations of the developmental templates into 26 common anatomical regions (13 per hemi-
³⁶³ sphere). We then used these regions to generate invertible transformations between succe-
³⁶⁴ ssive time points. Specifically each label was used to create a pair of single region images
³⁶⁵ resulting in 26 pairs of “source” and “target” images. The multiple image pairs were used
³⁶⁶ to iteratively estimate a diffeomorphic pairwise transform. Given the seven atlases E11.5,
³⁶⁷ E13.5, E15.5, E18.5, P4, P14, and P56, this resulted in 6 sets of transforms between succe-
³⁶⁸ ssive time points. Given the relative sizes between atlases, on the order of 10^6 points were
³⁶⁹ randomly sampled labelwise in the P56 template space and propagated to each successive
³⁷⁰ atlas providing the point sets for constructing the velocity flow model. Approximately 200
³⁷¹ iterations resulted in a steady convergence based on the average Euclidean norm between
³⁷² transformed point sets. Ten integration points were used and point sets were distributed
³⁷³ along the temporal dimension using a log transform for a more evenly spaced sampling.
³⁷⁴ Further details including links to data and scripts to reproduce our reported results is found
³⁷⁵ in the associated GitHub repository.

³⁷⁶ **Visualization**

³⁷⁷ To complement the well-known visualization capabilities of R and Python, e.g., ggplot2
³⁷⁸ and matplotlib, respectively, image-specific visualization capabilities are available in the

379 `ants.plot(...)` (Python) and `plot.antsImage(...)` (R). These are capable of illustrating
380 multiple slices in different orientations with both other image overlays as well as label images.

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

387 **References**

- 388 1. Keller, P. J. & Ahrens, M. B. Visualizing whole-brain activity and development at
389 the single-cell level using light-sheet microscopy. *Neuron* **85**, 462–83 (2015).
- 390 2. La Manno, G. *et al.* Molecular architecture of the developing mouse brain. *Nature*
391 **596**, 92–96 (2021).
- 392 3. Wen, L. *et al.* Single-cell technologies: From research to application. *Innovation*
393 (*Camb*) **3**, 100342 (2022).
- 394 4. Oh, S. W. *et al.* A mesoscale connectome of the mouse brain. *Nature* **508**, 207–14
395 (2014).
- 396 5. Gong, H. *et al.* Continuously tracing brain-wide long-distance axonal projections in
397 mice at a one-micron voxel resolution. *Neuroimage* **74**, 87–98 (2013).
- 398 6. Li, A. *et al.* Micro-optical sectioning tomography to obtain a high-resolution atlas of
399 the mouse brain. *Science* **330**, 1404–8 (2010).
- 400 7. Ueda, H. R. *et al.* Tissue clearing and its applications in neuroscience. *Nat Rev*
401 *Neurosci* **21**, 61–79 (2020).
- 402 8. Ståhl, P. L. *et al.* Visualization and analysis of gene expression in tissue sections by
403 spatial transcriptomics. *Science* **353**, 78–82 (2016).
- 404 9. Burgess, D. J. Spatial transcriptomics coming of age. *Nat Rev Genet* **20**, 317 (2019).
405
- 406 10. MacKenzie-Graham, A. *et al.* A multimodal, multidimensional atlas of the C57BL/6J
407 mouse brain. *J Anat* **204**, 93–102 (2004).
- 408 11. Mackenzie-Graham, A. J. *et al.* Multimodal, multidimensional models of mouse brain.
409 *Epilepsia* **48 Suppl 4**, 75–81 (2007).
- 410 12. Dong, H. W. *Allen reference atlas. A digital color brain atlas of the C57BL/6J male*
411 *mouse.* (John Wiley; Sons, 2008).

- 412 13. Wang, Q. *et al.* The allen mouse brain common coordinate framework: A 3D reference
413 atlas. *Cell* **181**, 936–953.e20 (2020).
- 414 14. Johnson, G. A. *et al.* Waxholm space: An image-based reference for coordinating
415 mouse brain research. *Neuroimage* **53**, 365–72 (2010).
- 416 15. Oguz, I., Zhang, H., Rumple, A. & Sonka, M. RATS: Rapid automatic tissue segmen-
417 tation in rodent brain MRI. *J Neurosci Methods* **221**, 175–82 (2014).
- 418 16. Sawiak, S. J., Picq, J.-L. & Dhenain, M. Voxel-based morphometry analyses of in
419 vivo MRI in the aging mouse lemur primate. *Front Aging Neurosci* **6**, 82 (2014).
- 420 17. Ashburner, J. SPM: A history. *Neuroimage* **62**, 791–800 (2012).
- 421
- 422 18. Modat, M. *et al.* Fast free-form deformation using graphics processing units. *Comput
423 Methods Programs Biomed* **98**, 278–84 (2010).
- 424 19. Tyson, A. L. *et al.* Accurate determination of marker location within whole-brain
425 microscopy images. *Sci Rep* **12**, 867 (2022).
- 426 20. Pallast, N. *et al.* Processing pipeline for atlas-based imaging data analysis of struc-
427 tural and functional mouse brain MRI (AIDAmri). *Front Neuroinform* **13**, 42 (2019).
- 428 21. Jenkinson, M., Beckmann, C. F., Behrens, T. E. J., Woolrich, M. W. & Smith, S. M.
429 FSL. *Neuroimage* **62**, 782–90 (2012).
- 430 22. Yeh, F.-C., Wedeen, V. J. & Tseng, W.-Y. I. Generalized q-sampling imaging. *IEEE
431 Trans Med Imaging* **29**, 1626–35 (2010).
- 432 23. Jorge Cardoso, M. *et al.* STEPS: Similarity and truth estimation for propagated
433 segmentations and its application to hippocampal segmentation and brain parcelation.
Med Image Anal **17**, 671–84 (2013).
- 434 24. Tustison, N. J. *et al.* N4ITK: Improved N3 bias correction. *IEEE Trans Med Imaging*
435 **29**, 1310–20 (2010).

- 436 25. Tustison, N. J. *et al.* The ANTsX ecosystem for quantitative biological and medical
imaging. *Sci Rep* **11**, 9068 (2021).
- 437
- 438 26. Goubran, M. *et al.* Multimodal image registration and connectivity analysis for inte-
gration of connectomic data from microscopy to MRI. *Nat Commun* **10**, 5504 (2019).
- 439
- 440 27. Celestine, M., Nadkarni, N. A., Garin, C. M., Bougacha, S. & Dhenain, M. Sammba-
MRI: A library for processing SmAll-MaMmal BrAin MRI data in python. *Front
Neuroinform* **14**, 24 (2020).
- 441
- 442 28. Ioanas, H.-I., Marks, M., Zerbi, V., Yanik, M. F. & Rudin, M. An optimized regis-
tration workflow and standard geometric space for small animal brain imaging. *Neu-
roimage* **241**, 118386 (2021).
- 443
- 444 29. Cox, R. W. AFNI: What a long strange trip it's been. *Neuroimage* **62**, 743–7 (2012).
- 445
- 446 30. Ni, H. *et al.* A robust image registration interface for large volume brain atlas. *Sci
Rep* **10**, 2139 (2020).
- 447
- 448 31. Jin, M. *et al.* SMART: An open-source extension of WholeBrain for intact mouse
brain registration and segmentation. *eNeuro* **9**, (2022).
- 449
- 450 32. Fürth, D. *et al.* An interactive framework for whole-brain maps at cellular resolution.
Nat Neurosci **21**, 139–149 (2018).
- 451
- 452 33. Negwer, M. *et al.* FriendlyClearMap: An optimized toolkit for mouse brain mapping
and analysis. *Gigascience* **12**, (2022).
- 453
- 454 34. Klein, S., Staring, M., Murphy, K., Viergever, M. A. & Pluim, J. P. W. Elastix: A
toolbox for intensity-based medical image registration. *IEEE Trans Med Imaging* **29**,
196–205 (2010).
- 455
- 456 35. Carey, H. *et al.* DeepSlice: Rapid fully automatic registration of mouse brain imaging
to a volumetric atlas. *Nat Commun* **14**, 5884 (2023).
- 457

- 458 36. Bajcsy, R. & Broit, C. Matching of deformed images. in *Sixth International Conference on Pattern Recognition (ICPR'82)* 351–353 (1982).
- 459
- 460 37. Bajcsy, R. & Kovacic, S. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing* **46**, 1–21 (1989).
- 461
- 462 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).
- 463
- 464 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).
- 465
- 466 40. Klein, A. *et al.* Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration. *Neuroimage* **46**, 786–802 (2009).
- 467
- 468 41. Murphy, K. *et al.* Evaluation of registration methods on thoracic CT: The EMPIRE10 challenge. *IEEE Trans Med Imaging* **30**, 1901–20 (2011).
- 469
- 470 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).
- 471
- 472 43. Wang, H. *et al.* Multi-atlas segmentation with joint label fusion. *IEEE Trans Pattern Anal Mach Intell* **35**, 611–23 (2013).
- 473
- 474 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).
- 475
- 476 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).
- 477

- 478 46. McCormick, M., Liu, X., Jomier, J., Marion, C. & Ibanez, L. ITK: Enabling repro-
479
ducible research and open science. *Front Neuroinform* **8**, 13 (2014).
- 480 47. Kronman, F. A. *et al.* Developmental mouse brain common coordinate framework.
481 *bioRxiv* (2023) doi:[10.1101/2023.09.14.557789](https://doi.org/10.1101/2023.09.14.557789).
- 482 48. Joshi, S. C. & Miller, M. I. Landmark matching via large deformation diffeomor-
483 phisms. *IEEE Trans Image Process* **9**, 1357–70 (2000).
- 484 49. Christensen, G. E., Rabbitt, R. D. & Miller, M. I. Deformable templates using large
485 deformation kinematics. *IEEE Trans Image Process* **5**, 1435–47 (1996).
- 486 50. Avants, B. B. *et al.* The optimal template effect in hippocampus studies of diseased
487 populations. *Neuroimage* **49**, 2457–66 (2010).
- 488 51. Tustison, N. J. *et al.* Optimal symmetric multimodal templates and concatenated
489 random forests for supervised brain tumor segmentation (simplified) with ANTsR.
Neuroinformatics **13**, 209–25 (2015).
- 490 52. Tustison, N. J. & Amini, A. A. Biventricular myocardial strains via nonrigid regis-
491 tration of anatomical NURBS model [corrected]. *IEEE Trans Med Imaging* **25**, 94–112
(2006).
- 492 53. Manjón, J. V., Coupé, P., Martí-Bonmatí, L., Collins, D. L. & Robles, M. Adaptive
493 non-local means denoising of MR images with spatially varying noise levels. *J Magn
Reson Imaging* **31**, 192–203 (2010).
- 494 54. Nyúl, L. G., Udupa, J. K. & Zhang, X. New variants of a method of MRI scale
495 standardization. *IEEE Trans Med Imaging* **19**, 143–50 (2000).
- 496 55. Falk, T. *et al.* U-net: Deep learning for cell counting, detection, and morphometry.
497 *Nat Methods* **16**, 67–70 (2019).
- 498 56. Haris, M., Shakhnarovich, G. & Ukita, N. Deep back-projection networks for super-
499 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).

- 500 57. Avants, B. B. *et al.* Concurrent 3D super resolution on intensity and segmentation
maps improves detection of structural effects in neurodegenerative disease. *medRxiv*
501 (2023) doi:[10.1101/2023.02.02.23285376](https://doi.org/10.1101/2023.02.02.23285376).
- 502 58. Avants, B. B. *et al.* The Insight ToolKit image registration framework. *Front Neu-*
503 *roinform* **8**, 44 (2014).
- 504 59. Tustison, N. J. & Avants, B. B. Explicit B-spline regularization in diffeomorphic image
505 registration. *Front Neuroinform* **7**, 39 (2013).
- 506 60. Avants, B. B. *et al.* A reproducible evaluation of ANTs similarity metric performance
507 in brain image registration. *Neuroimage* **54**, 2033–44 (2011).