

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 refer-
²² ence atlases. The Advanced Normalization Tools Ecosystem (ANTsX) is a comprehensive
²³ open-source software image analysis toolkit with applicability to multiple organ systems,
²⁴ modalities, and animal species. Herein, we illustrate the utility of ANTsX for generating
²⁵ precision spatial mappings of the mouse brain of different developmental ages including the
²⁶ prerequisite preprocessing steps. Additionally, as a further illustration of ANTsX capabil-
²⁷ ities, we use these publicly available mouse brain atlases to generate a velocity flow-based
²⁸ mapping encompassing the entire developmental trajectory, which we also make available to
²⁹ the public.

³⁰ Introduction

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

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

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 (available upon request). Of the publicly available pack-
⁵³ ages, most, if not all rely on well-established package dependencies originally developed on
⁵⁴ human brain data. Another early tool was SPMMouse¹⁶ 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 in-
60 corporates the FMRIB Software Library (FSL)²¹ for brain extraction and DSISTudio²² for
61 DTI processing, the latter uses NiftySeg and multi-consensus labeling tools²³ for brain extrac-
62 tion and parcellation. In addition, MASMAT incorporates N4 bias field correction²⁴ from the
63 Advanced Normalization Tools Ecosystem (ANTsX)²⁵ as do the packages Multi-modal Image
64 Registration And Connectivity anaLysis (MIRACL),²⁶ Saamba-MRI,²⁷ and Small Animal
65 Magnetic Resonance Imaging (SAMRI).²⁸ However, whereas Saamba-MRI uses AFNI²⁹ for
66 image registration; MIRACL, SAMRI, and BrainsMap³⁰ all use ANTsX tools for comput-
67 ing image-based correspondences. Other packages use landmark-based approaches to image
68 registration including SMART—³¹an R package for semi-automated landmark-based regis-
69 tration and segmentation of mouse brain based on WholeBrain.³² FriendlyClearMap³³ uses
70 the landmark-based registration functionality of Elastix.³⁴ Finally, the widespread adop-
71 tion of deep learning techniques has also influenced development in mouse brain imaging
72 methodologies. For example, if tissue deformations are not considered problematic for a
73 particular dataset, DeepSlice can be used to determine affine mappings³⁵ with the optimal
74 computational efficiency associated 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.

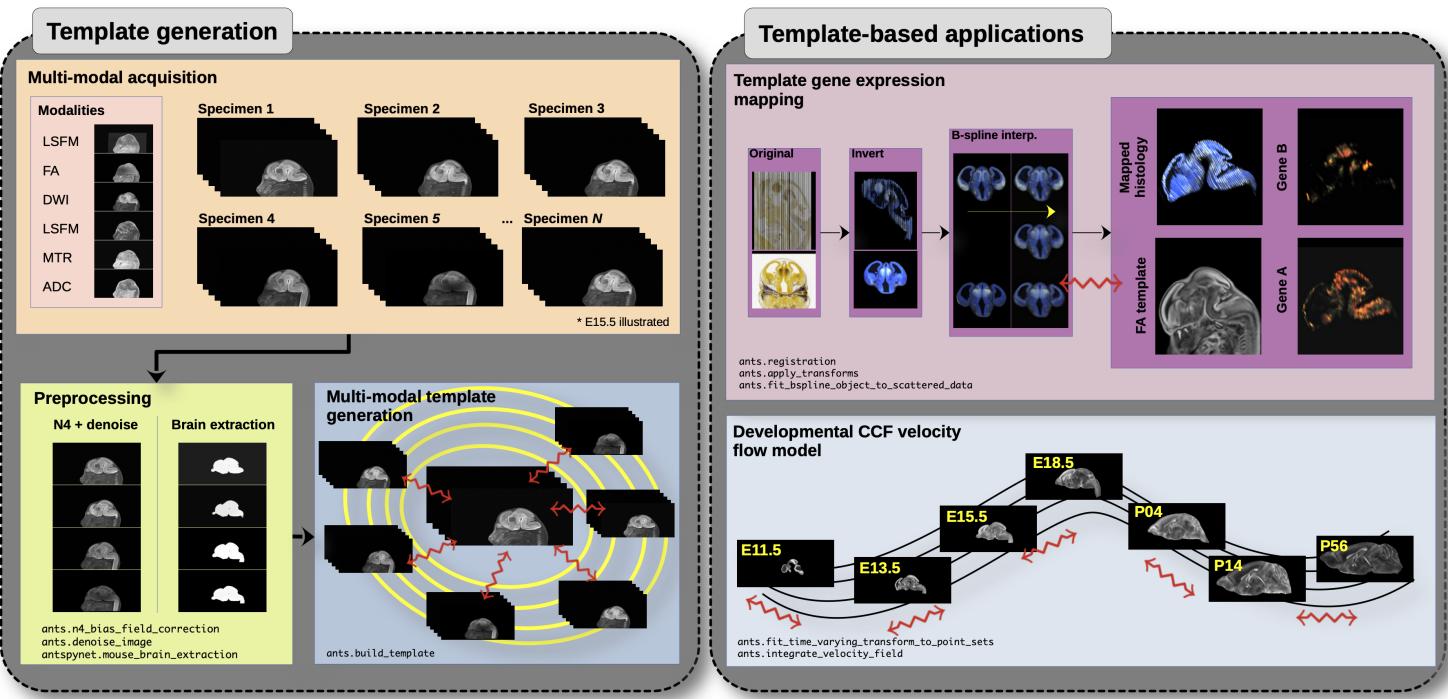


Figure 1: Illustration of a mouse brain template generation workflow and related template-based applications demonstrating the utility of different ANTsX tools. After imaging acquisition of the study population, various preprocessing steps are applied to the imaging data such as bias correction, denoising, 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. To further demonstrate the practical utility of the DevCCF, the
101 P56 template was integrated with the Allen CCFv3 for mapping spatial transcriptome cell-
102 type data. These processes, specifically template generation and multi-modal image map-
103 ping, were performed using ANTsX functionality in the presence of previously noted image
104 mapping difficulties (e.g., missing slices, tissue distortion) 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>).

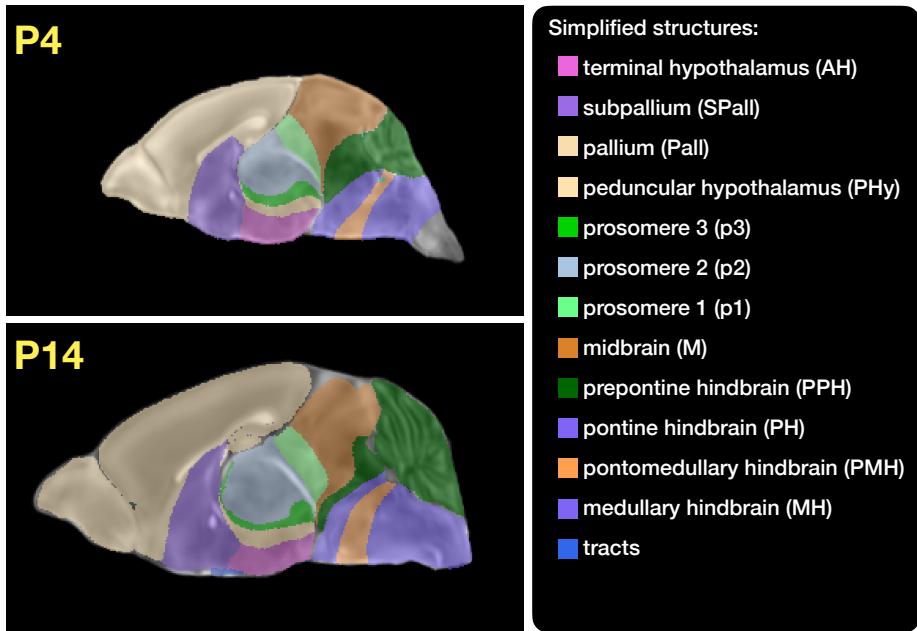


Figure 2: Annotated regions representing common labels across developmental stages which are illustrated for both P4 and P14.

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

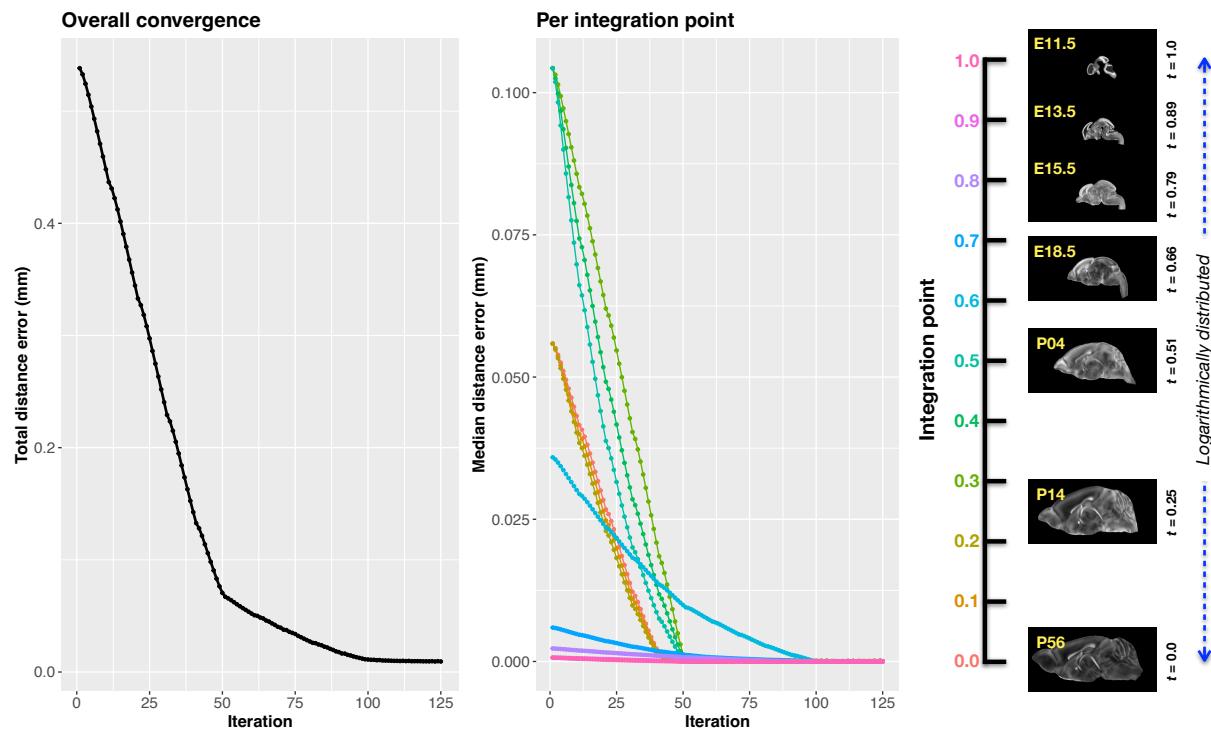


Figure 3: Convergence of the optimization of the velocity field for describing the transformation through the developmental stages from E11.5 through P56.

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.

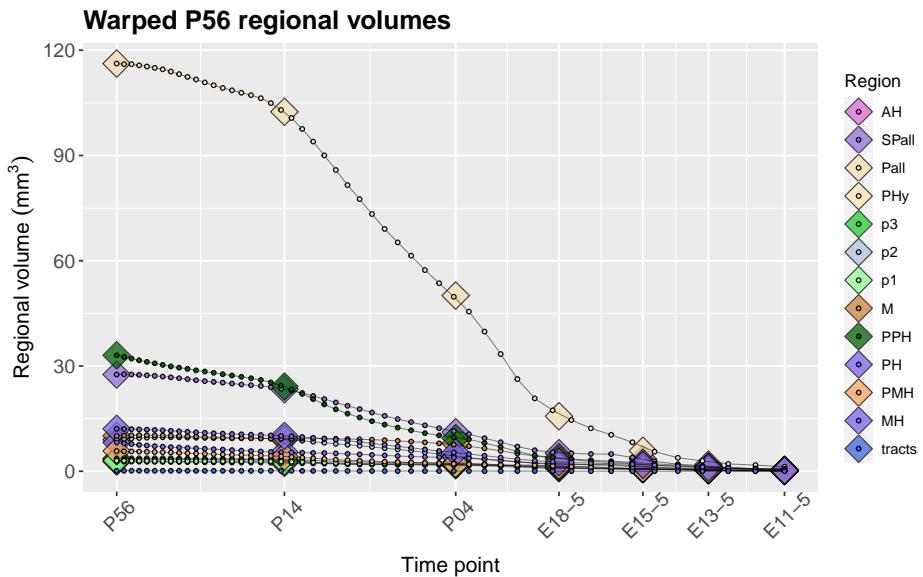


Figure 4: After the velocity field is generated, we can use it to warp the simplified labels of the P56 atlas continuously over the interval [0, 1] and plot the volumes of the atlas regions. Note how they compare with the volumes of the same regions in the other atlases.

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

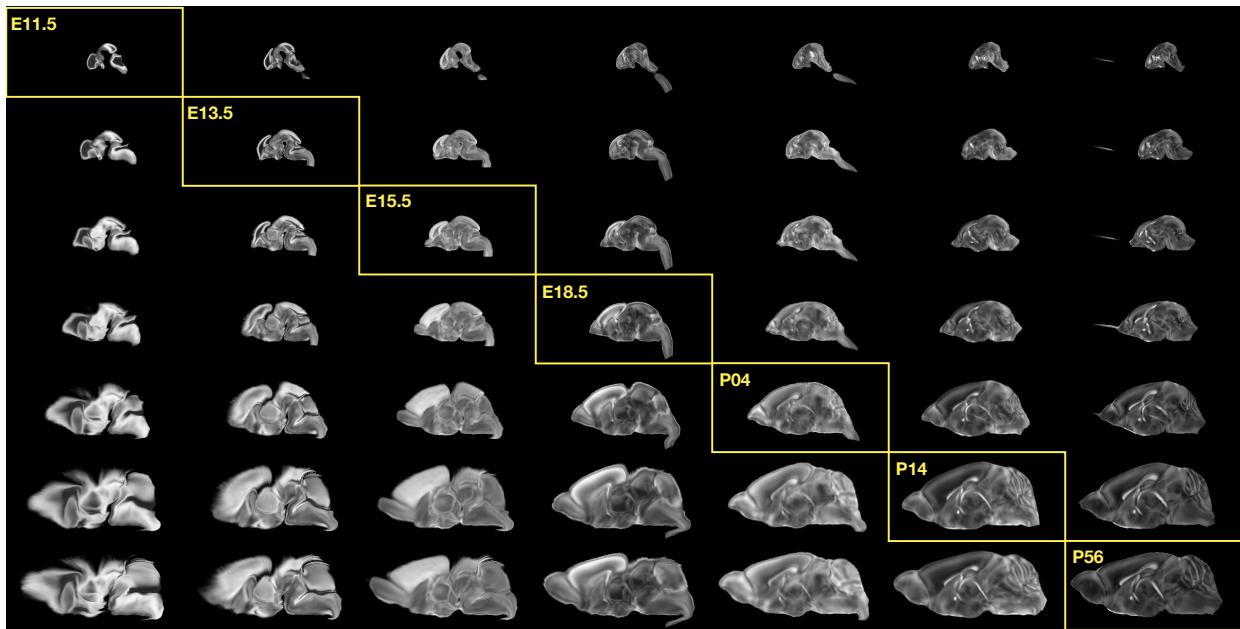


Figure 5: Mid-sagittal visualization of the effects of the transformation model in warping every developmental stage to the time point of every other developmental stage. The original images are located along the diagonal. Columns correspond to the warped original image whereas the rows represent the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

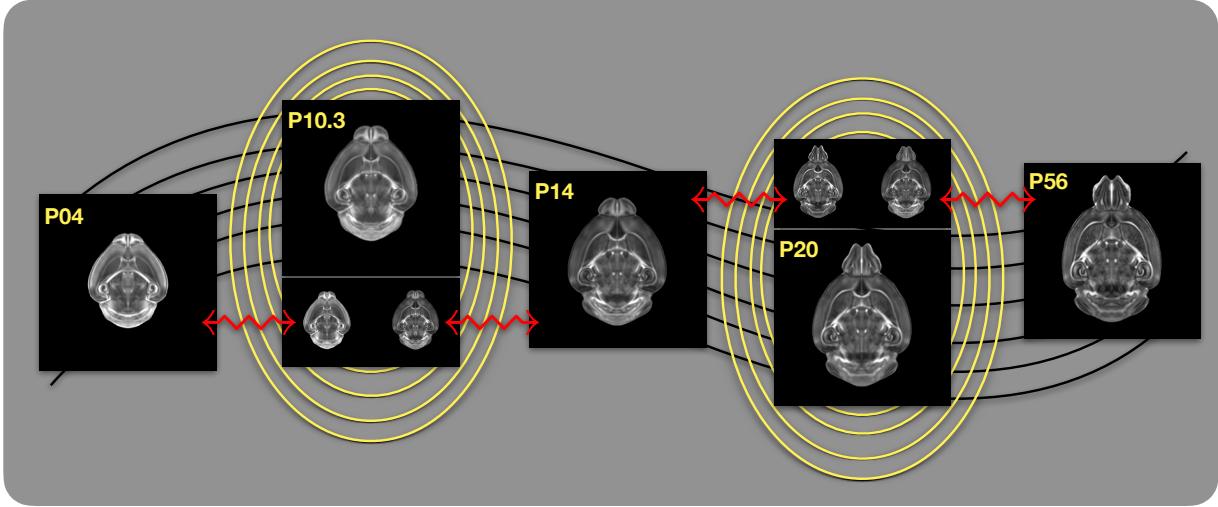


Figure 6: Illustration of the use of the velocity flow model for creating pseudo-templates at continuous time points not represented in one of the existing developmental stages. For example, FA templates at time point P10.3 and P20 can be generated by warping the existing temporally adjacent developmental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

195 structure 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 at
 201 that time point.

202 **Methods**

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

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

217 **ANTsXNet mouse brain applications**

218 *General notes regarding deep learning training.*

219 All network-based approaches described below were implemented and organized in the
220 ANTsXNet libraries comprising Python (ANTsPyNet) and R (ANTsRNet) analogs using the
221 Keras/Tensorflow libraries available as open-source in ANTsX GitHub repositories. For the
222 various applications, both share the identically trained weights for mutual reproducibility.
223 Training data was provided by manual labeling by various co-authors and expanded using
224 both intensity-based and shape-based data augmentation techniques.

225 Intensity-based data augmentation consisted of randomly added noise based on ITK
226 functionality, simulated bias fields based on N4 bias field modeling, and histogram
227 warping for mimicking well-known MRI intensity nonlinearities.^{25,54} These augmentation

techniques are available in ANTsXNet (only ANTsPyNet versions are listed): simulated bias field: `simulate_bias_field(...)`, image noise: `add_noise_to_image(...)`, and MRI intensity nonlinear characterization: `histogram_warp_image_intensities(...)`. Shape-based data augmentation used both random linear and nonlinear deformations. This functionality is also instantiated within ANTsXNet in terms of random spatial warping: `randomly_transform_image_data(...)`.

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

239 Brain extraction.

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

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

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

257 captured in these data using deep learning networks.

258 **Image registration**

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

260 Volumetric gene expression slice data was collated into 3-D volumes using . . . (ask Jeff).

261 Prior to mapping this volume to the corresponding structural data and, potentially, to the
262 appropriate template, alignment was improved using deformable registration on contiguous
263 slices. However, one of the complications associated with these image data was the un-
264 known number of missing slices, the number of consecutive missing slices, and the different
265 locations of these missing slices. To handle this missing data problem, we found that data
266 interpolation using the B-spline approximation algorithm cited earlier⁵² (ANTsPy function:
267 `fit_bspline_object_to_scattered_data(...)`). This provided sufficient data interpo-
268 lation fidelity to perform continuous slicewise registration. Other possible variants that
269 were considered but deemed unnecessary was performing more than one iteration cycling
270 through data interpolation and slicewise alignment. The other possibility was incorporating
271 the super-resolution technique described earlier. But again, our data did not require these
272 additional steps.

273 **Template generation**

274 ANTsX provides functionality for constructing templates from a set (or multi-modal sets) of
275 input images as originally described⁵⁰ and recently used to create the DevCCF templates.⁴⁷
276 An initial template estimate is constructed from an existing subject image or a voxelwise
277 average derived from a rigid pre-alignment of the image population. Pairwise registration
278 between each subject and the current template estimate is performed using the Symmetric
279 Normalization (SyN) algorithm.³⁹ The template estimate is updated by warping all subjects
280 to the space of the template, performing a voxelwise average, and then performing a “shape
281 update” of this latter image by warping it by the average inverse deformation, thus yielding

282 a mean image of the population in terms of both the intensity and shape.

283 **Continuous developmental velocity flow transformation model**

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

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

307 **Visualization**

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

312 **References**

- 313 1. Keller, P. J. & Ahrens, M. B. Visualizing whole-brain activity and development at
314 the single-cell level using light-sheet microscopy. *Neuron* **85**, 462–83 (2015).
- 315 2. La Manno, G. *et al.* Molecular architecture of the developing mouse brain. *Nature*
316 **596**, 92–96 (2021).
- 317 3. Wen, L. *et al.* Single-cell technologies: From research to application. *Innovation*
318 (*Camb*) **3**, 100342 (2022).
- 319 4. Oh, S. W. *et al.* A mesoscale connectome of the mouse brain. *Nature* **508**, 207–14
320 (2014).
- 321 5. Gong, H. *et al.* Continuously tracing brain-wide long-distance axonal projections in
322 mice at a one-micron voxel resolution. *Neuroimage* **74**, 87–98 (2013).
- 323 6. Li, A. *et al.* Micro-optical sectioning tomography to obtain a high-resolution atlas of
324 the mouse brain. *Science* **330**, 1404–8 (2010).
- 325 7. Ueda, H. R. *et al.* Tissue clearing and its applications in neuroscience. *Nat Rev*
326 *Neurosci* **21**, 61–79 (2020).
- 327 8. Ståhl, P. L. *et al.* Visualization and analysis of gene expression in tissue sections by
328 spatial transcriptomics. *Science* **353**, 78–82 (2016).
- 329 9. Burgess, D. J. Spatial transcriptomics coming of age. *Nat Rev Genet* **20**, 317 (2019).
- 330
- 331 10. MacKenzie-Graham, A. *et al.* A multimodal, multidimensional atlas of the C57BL/6J
332 mouse brain. *J Anat* **204**, 93–102 (2004).
- 333 11. Mackenzie-Graham, A. J. *et al.* Multimodal, multidimensional models of mouse brain.
334 *Epilepsia* **48 Suppl 4**, 75–81 (2007).
- 335 12. Dong, H. W. *Allen reference atlas. A digital color brain atlas of the C57BL/6J male*
336 *mouse.* (John Wiley; Sons, 2008).

- 337 13. Wang, Q. *et al.* The allen mouse brain common coordinate framework: A 3D reference
atlas. *Cell* **181**, 936–953.e20 (2020).
- 338
- 339 14. Johnson, G. A. *et al.* Waxholm space: An image-based reference for coordinating
340 mouse brain research. *Neuroimage* **53**, 365–72 (2010).
- 341 15. Oguz, I., Zhang, H., Rumple, A. & Sonka, M. RATS: Rapid automatic tissue segmen-
342 tation in rodent brain MRI. *J Neurosci Methods* **221**, 175–82 (2014).
- 343 16. Sawiak, S. J., Picq, J.-L. & Dhenain, M. Voxel-based morphometry analyses of in
344 vivo MRI in the aging mouse lemur primate. *Front Aging Neurosci* **6**, 82 (2014).
- 345 17. Ashburner, J. SPM: A history. *Neuroimage* **62**, 791–800 (2012).
- 346
- 347 18. Modat, M. *et al.* Fast free-form deformation using graphics processing units. *Comput
348 Methods Programs Biomed* **98**, 278–84 (2010).
- 349 19. Tyson, A. L. *et al.* Accurate determination of marker location within whole-brain
350 microscopy images. *Sci Rep* **12**, 867 (2022).
- 351 20. Pallast, N. *et al.* Processing pipeline for atlas-based imaging data analysis of struc-
352 tural and functional mouse brain MRI (AIDAmri). *Front Neuroinform* **13**, 42 (2019).
- 353 21. Jenkinson, M., Beckmann, C. F., Behrens, T. E. J., Woolrich, M. W. & Smith, S. M.
354 FSL. *Neuroimage* **62**, 782–90 (2012).
- 355 22. Yeh, F.-C., Wedeen, V. J. & Tseng, W.-Y. I. Generalized q-sampling imaging. *IEEE
356 Trans Med Imaging* **29**, 1626–35 (2010).
- 357 23. Jorge Cardoso, M. *et al.* STEPS: Similarity and truth estimation for propagated
358 segmentations and its application to hippocampal segmentation and brain parcelation.
Med Image Anal **17**, 671–84 (2013).
- 359 24. Tustison, N. J. *et al.* N4ITK: Improved N3 bias correction. *IEEE Trans Med Imaging*
360 **29**, 1310–20 (2010).

- 361 25. Tustison, N. J. *et al.* The ANTsX ecosystem for quantitative biological and medical
362 imaging. *Sci Rep* **11**, 9068 (2021).
- 363 26. Goubran, M. *et al.* Multimodal image registration and connectivity analysis for inte-
364 gration of connectomic data from microscopy to MRI. *Nat Commun* **10**, 5504 (2019).
- 365 27. Celestine, M., Nadkarni, N. A., Garin, C. M., Bougacha, S. & Dhenain, M. Sammba-
366 MRI: A library for processing SmAll-MaMmal BrAin MRI data in python. *Front
Neuroinform* **14**, 24 (2020).
- 367 28. Ioanas, H.-I., Marks, M., Zerbi, V., Yanik, M. F. & Rudin, M. An optimized regis-
368 tration workflow and standard geometric space for small animal brain imaging. *Neu-
roimage* **241**, 118386 (2021).
- 369 29. Cox, R. W. AFNI: What a long strange trip it's been. *Neuroimage* **62**, 743–7 (2012).
- 370
- 371 30. Ni, H. *et al.* A robust image registration interface for large volume brain atlas. *Sci
372 Rep* **10**, 2139 (2020).
- 373 31. Jin, M. *et al.* SMART: An open-source extension of WholeBrain for intact mouse
374 brain registration and segmentation. *eNeuro* **9**, (2022).
- 375 32. Fürth, D. *et al.* An interactive framework for whole-brain maps at cellular resolution.
376 *Nat Neurosci* **21**, 139–149 (2018).
- 377 33. Negwer, M. *et al.* FriendlyClearMap: An optimized toolkit for mouse brain mapping
378 and analysis. *Gigascience* **12**, (2022).
- 379 34. Klein, S., Staring, M., Murphy, K., Viergever, M. A. & Pluim, J. P. W. Elastix: A
380 toolbox for intensity-based medical image registration. *IEEE Trans Med Imaging* **29**,
196–205 (2010).
- 381 35. Carey, H. *et al.* DeepSlice: Rapid fully automatic registration of mouse brain imaging
382 to a volumetric atlas. *Nat Commun* **14**, 5884 (2023).

- 383 36. Bajcsy, R. & Broit, C. Matching of deformed images. in *Sixth International Conference on Pattern Recognition (ICPR'82)* 351–353 (1982).
- 384
- 385 37. Bajcsy, R. & Kovacic, S. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing* **46**, 1–21 (1989).
- 386
- 387 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).
- 388
- 389 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).
- 390
- 391 40. Klein, A. *et al.* Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration. *Neuroimage* **46**, 786–802 (2009).
- 392
- 393 41. Murphy, K. *et al.* Evaluation of registration methods on thoracic CT: The EMPIRE10 challenge. *IEEE Trans Med Imaging* **30**, 1901–20 (2011).
- 394
- 395 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).
- 396
- 397 43. Wang, H. *et al.* Multi-atlas segmentation with joint label fusion. *IEEE Trans Pattern Anal Mach Intell* **35**, 611–23 (2013).
- 398
- 399 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).
- 400
- 401 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).
- 402

- 403 46. McCormick, M., Liu, X., Jomier, J., Marion, C. & Ibanez, L. ITK: Enabling repro-
404
ducible research and open science. *Front Neuroinform* **8**, 13 (2014).
- 405 47. Kronman, F. A. *et al.* Developmental mouse brain common coordinate framework.
406 *bioRxiv* (2023) doi:[10.1101/2023.09.14.557789](https://doi.org/10.1101/2023.09.14.557789).
- 407 48. Joshi, S. C. & Miller, M. I. Landmark matching via large deformation diffeomor-
408 phisms. *IEEE Trans Image Process* **9**, 1357–70 (2000).
- 409 49. Christensen, G. E., Rabbitt, R. D. & Miller, M. I. Deformable templates using large
410 deformation kinematics. *IEEE Trans Image Process* **5**, 1435–47 (1996).
- 411 50. Avants, B. B. *et al.* The optimal template effect in hippocampus studies of diseased
412 populations. *Neuroimage* **49**, 2457–66 (2010).
- 413 51. Tustison, N. J. *et al.* Optimal symmetric multimodal templates and concatenated
414 random forests for supervised brain tumor segmentation (simplified) with ANTsR.
Neuroinformatics **13**, 209–25 (2015).
- 415 52. Tustison, N. J. & Amini, A. A. Biventricular myocardial strains via nonrigid regis-
416 tration of anatomical NURBS model [corrected]. *IEEE Trans Med Imaging* **25**, 94–112
(2006).
- 417 53. Manjón, J. V., Coupé, P., Martí-Bonmatí, L., Collins, D. L. & Robles, M. Adaptive
418 non-local means denoising of MR images with spatially varying noise levels. *J Magn
Reson Imaging* **31**, 192–203 (2010).
- 419 54. Nyúl, L. G., Udupa, J. K. & Zhang, X. New variants of a method of MRI scale
420 standardization. *IEEE Trans Med Imaging* **19**, 143–50 (2000).
- 421 55. Falk, T. *et al.* U-net: Deep learning for cell counting, detection, and morphometry.
422 *Nat Methods* **16**, 67–70 (2019).
- 423 56. Haris, M., Shakhnarovich, G. & Ukita, N. Deep back-projection networks for super-
424 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).

- 425 57. avants, brian *et al.* Concurrent 3D super resolution on intensity and segmentation
maps improves detection of structural effects in neurodegenerative disease. *medRxiv*
426 (2023) doi:[10.1101/2023.02.02.23285376](https://doi.org/10.1101/2023.02.02.23285376).