

1

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

5      Nicholas J. Tustison<sup>1</sup>, Min Chen<sup>2</sup>, Fae N. Kronman<sup>3</sup>, Jeffrey T. Duda<sup>2</sup>, Clare Gamlin<sup>4</sup>,  
6      Lydia Ng<sup>4</sup>, Yongsoo Kim<sup>3</sup>, and James C. Gee<sup>2</sup>

7      <sup>1</sup>Department of Radiology and Medical Imaging, University of Virginia, Charlottesville, VA

8      <sup>2</sup>Department of Radiology, University of Pennsylvania, Philadelphia, PA

9      <sup>3</sup>Department of Neural and Behavioral Sciences, Penn State University, Hershey, PA

10     <sup>4</sup>Allen Institute for Brain Science, Seattle, WA

11

---

12     Corresponding author:

13     Nicholas J. Tustison, DSc

14     Department of Radiology and Medical Imaging

15     University of Virginia

16     [ntustison@virginia.edu](mailto:ntustison@virginia.edu)

<sup>17</sup> **Abstract**

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

<sup>31</sup> **Introduction**

<sup>32</sup> Over the past two decades there have been significant advancements in mesoscopic anal-  
<sup>33</sup> ysis of the mouse brain. It is now possible to track single cell neurons in mouse brains,<sup>1</sup>  
<sup>34</sup> observe whole brain developmental changes on a cellular level,<sup>2</sup> associate brain regions and  
<sup>35</sup> tissues with their genetic composition,<sup>3</sup> and locally characterize neural connectivity.<sup>4</sup> Much  
<sup>36</sup> of this scientific achievement has been made possible due to breakthroughs in high resolution  
<sup>37</sup> imaging techniques that permit submicron, 3-D imaging of whole mouse brains. Associated  
<sup>38</sup> research techniques such as micro-optical sectioning tomography,<sup>6</sup> tissue clearing,<sup>1,7</sup> spatial  
<sup>39</sup> transcriptomics<sup>9</sup> are all well-utilized in the course of scientific investigations of mesoscale  
<sup>40</sup> relationships in the mouse brain.

<sup>41</sup> An important component of this research is the ability to map the various image data to  
<sup>42</sup> anatomical reference frames<sup>11</sup> for inferring spatial relationships between structures, cells,  
<sup>43</sup> and genetics. This has motivated the development of detailed structural image atlases of the  
<sup>44</sup> mouse brain. Notable examples include the Allen Brain Atlas and Coordinate Frameworks<sup>13</sup>  
<sup>45</sup> and the Waxholm Space.<sup>14</sup> Despite the significance of these contributions, challenges still  
<sup>46</sup> exist in large part due to the wide heterogeneity in associated study-specific image data. For  
<sup>47</sup> example, variance in the acquisition methods can introduce artifacts such as tissue distor-  
<sup>48</sup> tion, holes, bubbles, folding, tears, and missing slices. These severely complicate assumed  
<sup>49</sup> correspondence for conventional spatial mapping approaches.

<sup>50</sup> To address such challenges, several software packages have been developed over the years  
<sup>51</sup> comprising solutions of varying comprehensibility, sophistication, and availability. An early  
<sup>52</sup> contribution to the community was the Rapid Automatic Tissue Segmentation (RATS)  
<sup>53</sup> package<sup>15</sup> for brain extraction. Of the publicly available packages, most, if not all have well-  
<sup>54</sup> established package dependencies originally developed on human brain data. SPMMouse,<sup>16</sup>  
<sup>55</sup> for example, is based on the well-known Statistical Parametric Mapping (SPM) software  
<sup>56</sup> package.<sup>17</sup> The automated mouse atlas propagation (aMAP) tool is largely a front-end for  
<sup>57</sup> the NiftyReg image registration package<sup>18</sup> applied to mouse data which is currently avail-  
<sup>58</sup> able as a Python module.<sup>19</sup> NiftyReg is also used by the Atlas-based Imaging Data Anal-  
<sup>59</sup> ysis (AIDA) MRI pipeline<sup>20</sup> as well as the Multi Atlas Segmentation and Morphometric

60 Analysis Toolkit (MASMAT). Whereas the former also incorporates the FMRIB Software  
61 Library (FSL)<sup>21</sup> for brain extraction and DSISTudio<sup>22</sup> for DTI processing, the latter uses  
62 NiftySeg and multi-consensus labeling tools<sup>23</sup> for brain extraction and parcellation. In ad-  
63 dition, MASMAT incorporates N4 bias field correction<sup>24</sup> from the Advanced Normalization  
64 Tools Ecosystem (ANTsX)<sup>25</sup> as do the packages Multi-modal Image Registration And Con-  
65 nectivity anaLysis (MIRACL),<sup>26</sup> Saamba-MRI,<sup>27</sup> and Small Animal Magnetic Resonance  
66 Imaging (SAMRI).<sup>28</sup> However, whereas Saamba-MRI uses AFNI<sup>29</sup> for image registration;  
67 MIRACL, SAMRI, and BrainsMapi<sup>30</sup> all use ANTsX registration tools. Other packages  
68 use landmark-based approaches to image registration including SMART—<sup>31</sup>an R package  
69 for semi-automated landmark-based registration and segmentation of mouse brain based  
70 on WholeBrain.<sup>32</sup> FriendlyClearMap<sup>33</sup> uses the landmark-based registration functionality of  
71 Elastix.<sup>34</sup> Finally, the widespread adoption of deep learning techniques has also influenced  
72 development in mouse brain imaging methodologies. For example, if tissue deformations  
73 are not considered problematic for a particular dataset, DeepSlice can be used to determine  
74 affine mappings<sup>35</sup> with the optimal computational efficiency associated with neural networks.

## 75 The ANTsX Ecosystem

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

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

Table 1: Sampling of ANTsX functionality

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

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

88 trained for mouse data (see Table 1). The collective use of the toolkit has demonstrated  
 89 superb performance in multiple application areas (e.g., consensus labeling,<sup>43</sup> brain tumor  
 90 segmentation,<sup>44</sup> and cardiac motion estimation<sup>45</sup> ). Importantly, ANTs is built on the In-  
 91 sight Toolkit (ITK)<sup>46</sup> deriving benefit from the open-source community of scientists and  
 92 programmers and providing a visible, open-source venue for algorithmic contributions.

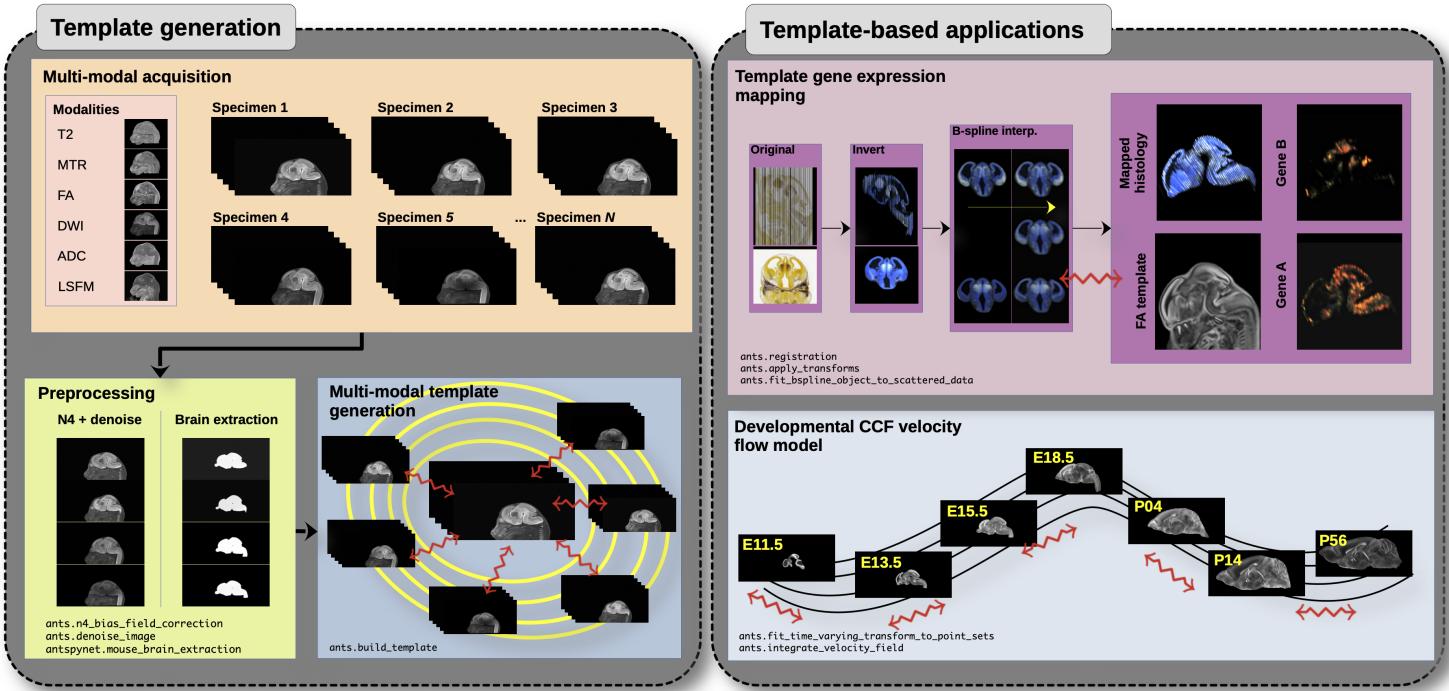


Figure 1: Illustration of a mouse brain template generation workflow and related template-based applications demonstrating the utility of different ANTsX tools. After imaging acquisition of the study population, various preprocessing steps are applied to the imaging data such as bias correction, denoising,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

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

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

104 Given the temporal gaps in the discrete set of developmental atlases with the potential for  
105 additional interpolative time points, we discuss the strategy of the current DevCCF tem-  
106 plate generation described in<sup>47</sup> and provide additional information for the interested reader.  
107 Related, we also provide an open-source framework, through ANTsX, for inferring correspon-  
108 dence within the temporally continuous domain sampled by the existing set of embryonic  
109 and postnatal atlases of the DevCCF. This recently developed ANTsX functionality per-  
110 mits the generation of a diffeomorphic velocity flow transformation model,<sup>48</sup> with variation  
111 based on the ideas introduced previously.<sup>49</sup> The resulting time-parameterized velocity field  
112 spans the stages of the DevCCF where mappings between any two continuous time points  
113 within the span bounded by the E11.5 and P56 atlases is determined by integration of the  
114 optimized velocity field. This functionality is available through ANTsX (via R and Python  
115 ANTsX packages) with a dedicated GitHub repository that contains all data, scripts, and  
116 other guidance necessary to both reproduce what is described below and to illustrate how  
117 future researchers can incorporate additional atlases into a more densely sampled model in  
118 a straightforward manner.

<sub>119</sub> **Results**

<sub>120</sub> **Template building**

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

<sub>127</sub> **Data preparation**

<sub>128</sub> Multi-modal symmetric template construction is performed separately for each develop-  
<sub>129</sub> mental stage. Prior to optimization, preprocessing can include several steps not all of  
<sub>130</sub> which are required but are dependent on the data and the particular requirements of the  
<sub>131</sub> study. For MRI scans, inhomogeneity correction is often necessary and can be performed  
<sub>132</sub> using the ANTsPy function `ants.n4_bias_field_correction(...)` which is a wrapper  
<sub>133</sub> for the N4 algorithm.<sup>24</sup> Denoising is another preprocessing step that can potentially im-  
<sub>134</sub> prove template quality results. The ANTsPy function `ants.denoise_image(...)` is an  
<sub>135</sub> implementation of a well-known denoising algorithm.<sup>52</sup> For a typical image, both of these  
<sub>136</sub> steps takes approximately on the order of a couple minutes. In ANTsX, due to legacy  
<sub>137</sub> code issues, only bias correction is wrapped with template building so one need not per-  
<sub>138</sub> form this step prior to optimization. In addition, brain extraction has demonstrated im-  
<sub>139</sub> proved performance in the context of human brain normalization<sup>53</sup> and is similarly used  
<sub>140</sub> in mouse brain registration to maximize alignment. Various approaches within ANTs are  
<sub>141</sub> possible including a template-based approach `antsBrainExtraction.sh` or using deep learn-  
<sub>142</sub> ing `antspynet.mouse_brain_extraction(...)`. Additionally, it is important to ensure a  
<sub>143</sub> standardized orientation, similar to the Dicom standard for human brain imaging. A study  
<sub>144</sub> requirement of template bilateral symmetry is also an important consideration prior to tem-  
<sub>145</sub> plate generation. This can be performed by either flipping all the input images contralaterally

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

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

## 158 Optimization

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

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

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

## <sup>176</sup> The DevCCF Velocity Flow Model

<sup>177</sup> To continuously link the DevCCF atlases, a velocity flow model was constructed using Dev-  
<sup>178</sup> CCF derived data and ANTsX functionality available in both ANTsR and ANTsPy. Al-  
<sup>179</sup> though many implementations optimize variations of this transformation model (and others)  
<sup>180</sup> using various image intensity similarity metrics, we opted to implement a separate de-  
<sup>181</sup> termination of iterative correspondence and transformation optimization. This decision was  
<sup>182</sup> based on existing ANTsX functionality and wanting complementary utility for the toolkit.

<sup>183</sup> ANTsX, being built on top of ITK, uses an ITK image data structure for the 4-D velocity  
<sup>184</sup> field where each voxel contains the  $x$ ,  $y$ ,  $z$  components of the field at that point. Field  
<sup>185</sup> regularization is provided by a novel B-spline scattered data approximation technique<sup>54</sup> which  
<sup>186</sup> permits individual point-based weighting. Both field regularization and integration of the  
<sup>187</sup> velocity field are built on ITK functions written by ANTsX developers.

<sup>188</sup> The optimized velocity field described here is of size [256, 182, 360] ( $50\mu\text{m}$  isotropic)  $\times$  11  
<sup>189</sup> integration points for a total compressed size of a little over 2 GB. This choice represented  
<sup>190</sup> weighing the trade-off between tractability, portability, and accuracy. However, all data  
<sup>191</sup> and code to reproduce the results described are available in a dedicated GitHub repository  
<sup>192</sup> (<https://github.com/ntustison/DevCCF-Velocity-Flow>).

## <sup>193</sup> Data preparation

<sup>194</sup> Labeled annotations are available as part of the original DevCCF and reside in the space  
<sup>195</sup> of each developmental template which range in resolution from  $31.5 - 50\mu\text{m}$ . Across all at-  
<sup>196</sup> lases, the total number of labels exceeded 2500 without taken into account per hemispherical  
<sup>197</sup> enumeration. From this set of labels, there were a common set of 24 labels (12 per hemi-  
<sup>198</sup> sphere) across all atlases that were used for optimization and evaluation. These regions are  
<sup>199</sup> illustrated for the P4 and P14 stages in Figure 2.

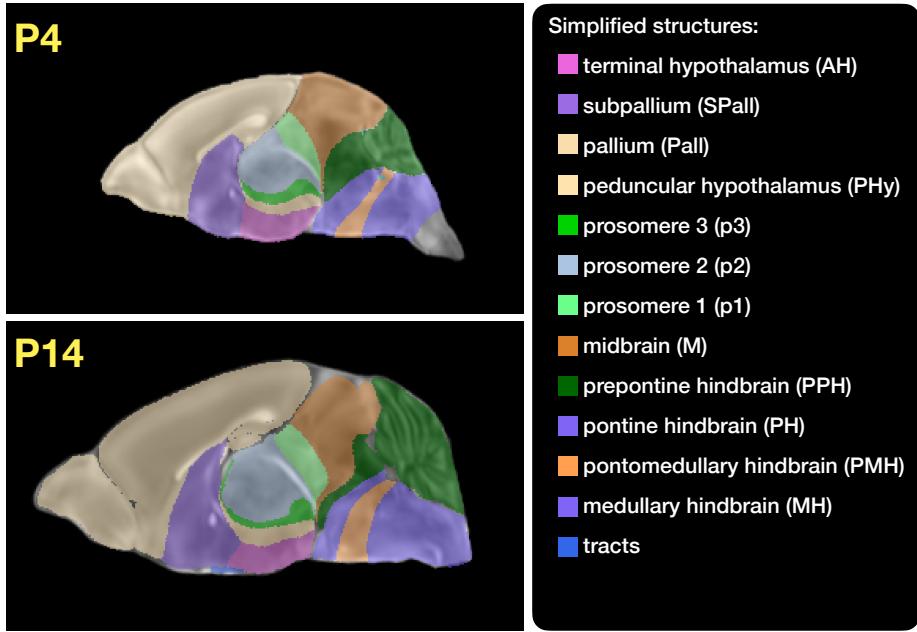


Figure 2: Annotated regions representing common labels across developmental stages which are illustrated for both P4 and P14.

200 Prior to velocity field optimization, the data was rigidly transformed to a common  
 201 space. Using the centroids for the common label set of each CCFDev atlas, the ANTsPy  
 202 `ants.fit_transform_to_paired_points(...)` function was used to warp each atlas  
 203 to the space of the P56 atlas and then downsampled to  $50\mu\text{m}$  isotropic resolution. In  
 204 order to determine the common point sets across stages, the multi-metric capabilities  
 205 of `ants.registration(...)` were used. Instead of performing intensity-based pairwise  
 206 registration directly on these multi-label images, each label was used to construct a separate  
 207 fixed and moving image pair resulting in a multi-metric registration optimization scenario  
 208 involving 24 binary image pairs (each label weighted equally) for optimizing correspondence  
 209 between neighboring atlases using the mean squares metric and the SyN transform.

210 To provide the common point sets across all seven developmental atlases, the label boundaries  
 211 and whole regions were sampled in the P56 atlas and then propagated to each atlas using  
 212 the transformations derived from the pairwise registrations. Based on previous experience  
 213 as both the developers of users of these tools, we selected a sampling rate of 10% for the  
 214 contour points and 1% for the regional points for a total number of points being per atlas  
 215 being 173303 ( $N_{contour} = 98151$  and  $N_{region} = 75152$ ). Boundary points were weighted twice

as those of regional points for the B-spline data approximation optimization.

## Optimization

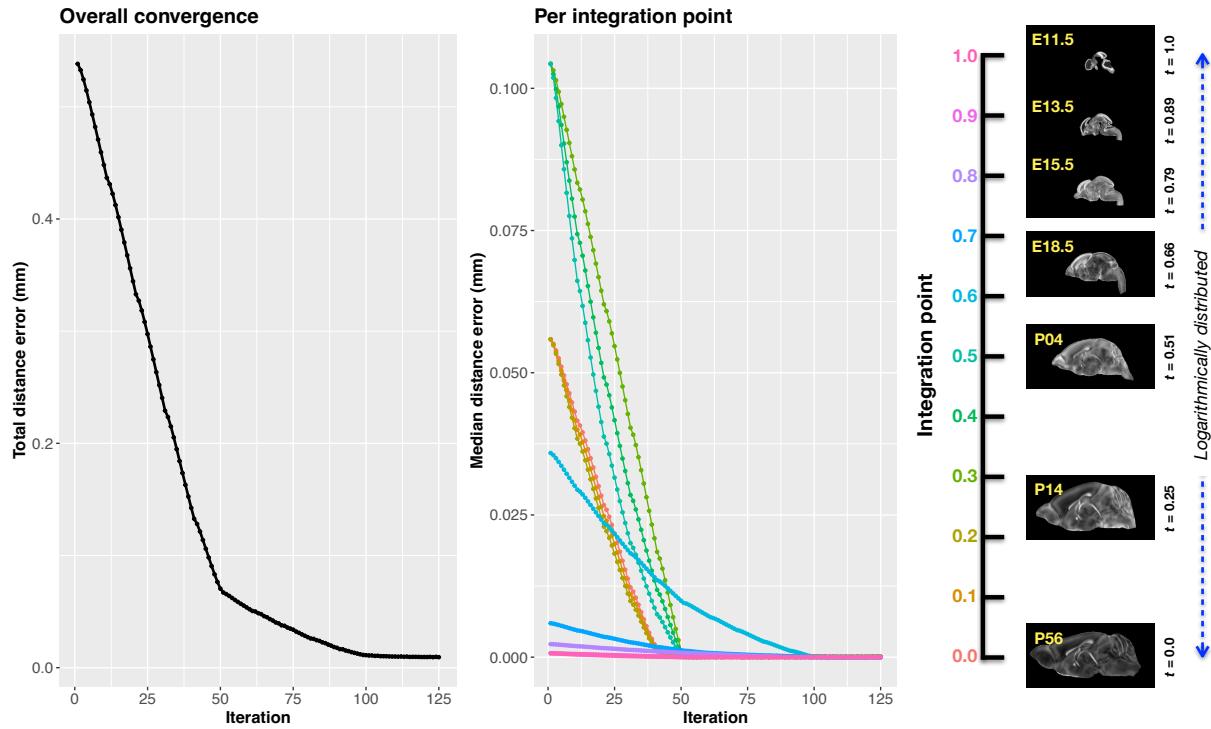


Figure 3: Convergence of the optimization of the velocity field for describing the transformation through the developmental stages from E11.5 through P56.

`ants.fit_time_varying_transform_to_point_sets(...)` from the ANTsPy package was used to optimize the velocity field. Input comprised the seven corresponding point sets and their associated weight values, the selected number of integration points for the velocity field ( $N = 11$ ), and the parameters defining the geometry of the spatial dimensions of the velocity field (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.

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

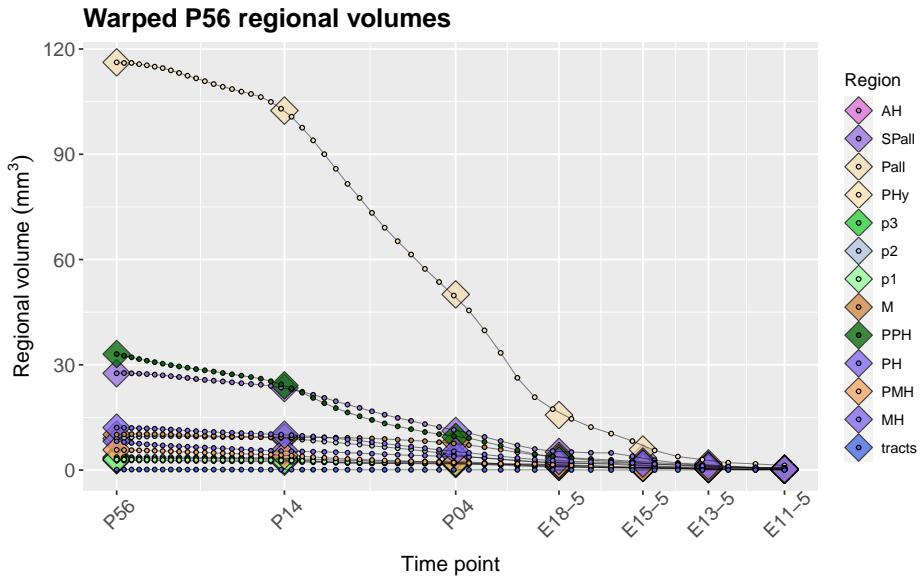


Figure 4: After the velocity field is generated, we can use it to warp the simplified labels of the P56 atlas continuously over the interval [0, 1] and plot the volumes of the atlas regions. Note how they compare with the volumes of the same regions in the other atlases.

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

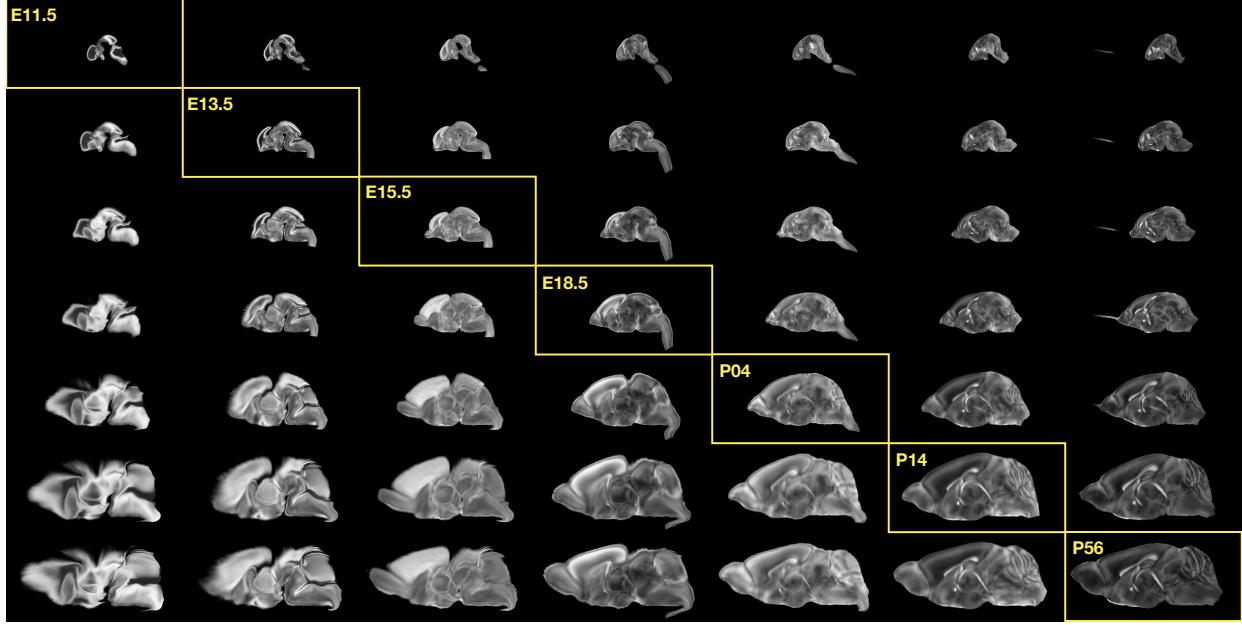


Figure 5: Mid-sagittal visualization of the effects of the transformation model in warping every developmental stage to the time point of every other developmental stage. The original images are located along the diagonal. Columns correspond to the warped original image whereas the rows represent the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

#### 243 The DevCCF transform model

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

249 One potential application for this particular transformation model is facilitating the con-  
250 struction of virtual-templates in the temporal gaps of the DevCCF. This is illustrated in  
251 Figure 6 where we used the optimized velocity field to construct virtual-templates at time  
252 point P10.3 and P20—arbitrarily chosen simply to demonstrate the concept. After situating  
253 these time points within the normalized time point interval, the existing adjacent DevCCF  
254 atlases on either side can be warped to the desired time point. A subsequent call to one  
255 of the ANTsX template building functions then permits the construction of the template  
256 at that time point. Note that both of these usage examples can be found on the GitHub

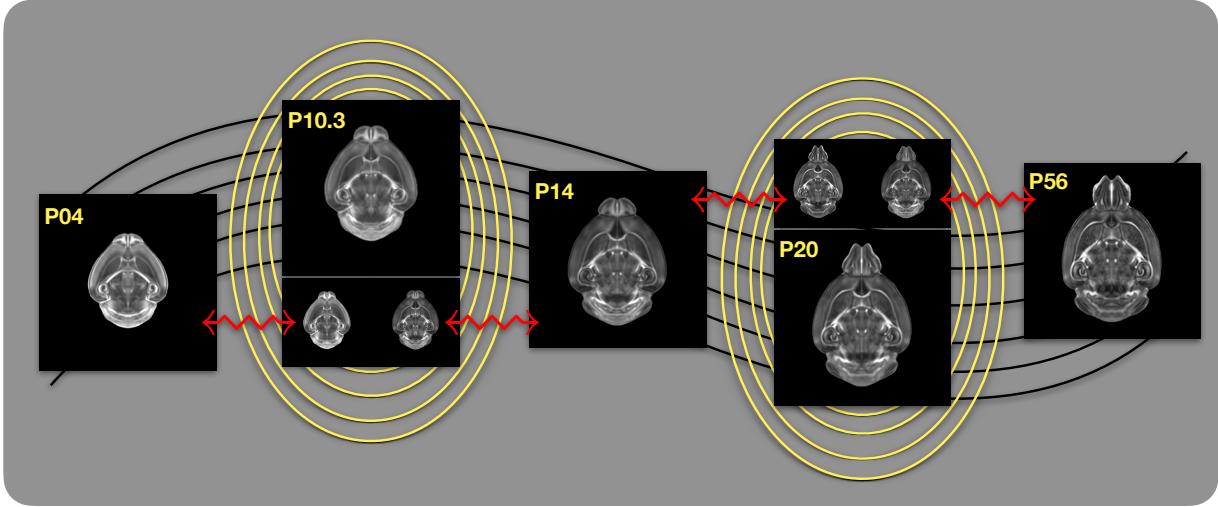


Figure 6: Illustration of the use of the velocity flow model for creating virtual templates at continuous time points not represented in one of the existing DevCCF time points. For example, FA templates at time point P10.3 and P20 can be generated by warping the existing temporally adjacent developmental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

<sup>257</sup> repository given above.

258 **Discussion**

259 The ANTsX ecosystem is a powerful framework that has demonstrated  
260 applicability to multiple species and organ systems, including the mouse brain.  
261 This has been demonstrated in many ways including the significant number of external  
262 software packages that use various ANTsX components in their own mouse-specific work-  
263 flows. The extensive functionality of ANTsX makes it possible to create complete processing  
264 pipelines without requiring multiple packages which often have limited cross-compatibility  
265 evaluation.

266 These open-source ANTsX components not only perform well but are available across mul-  
267 tiple popular platforms (e.g., R and Python) which facilitates the construction of tailored  
268 pipelines for individual study solutions. These components are also supported by years of  
269 development not only by the ANTsX development team but by the larger ITK community.

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

280 Despite the significant expansion of available developmental age templates beyond what pre-  
281 viously existed (e.g., Allen CCFv3), there still exist temporal gaps in the DevCCF. However,  
282 pioneering work involving diffeomorphic transformations allowed us to continuously situate  
283 the existing templates within a time-varying velocity flow model. This allows one to deter-  
284 mine the diffeomorphic transformation from any one temporal location to any other temporal  
285 location within the time span defined by the E11.5 and P56 templates. This functionality  
286 is built on multiple components from the Insight Segmentation and Registratiton Toolkit

287 including the B-spline scattered data approximation technique for field regularization and  
288 velocity field integration using fourth order Runge-Kutta. This velocity field model permits  
289 intra-template comparison and the construction of virtual templates where a template can  
290 be estimated at any continuous time point within the temporal domain. This novel appli-  
291 cation can potentially enhance our understanding of intermediate developmental states. To  
292 increase its impact and reproduce the results shown previously, we have made the data and  
293 code publicly available at <https://github.com/ntustison/DevCCF-Velocity-Flow>.

294 Although ANTsX is quite evolved in its development and functionality, there are several  
295 areas which are currently under active development or consideration for further expan-  
296 sion. Most notably, as in our human applications, deep learning has had a significant  
297 impact in steering our attention. Core functionality, such as brain extraction for mouse  
298 brain mapping, would benefit from increasing the number of available modalities. As with  
299 much deep learning development, such work will require additional data but is significantly  
300 facilitated by the tools that we have created in both ANTsPyNet and ANTsRNet. Re-  
301 lated would be the utility of the development of mouse brain parcellation tools such as our  
302 `antspynet.desikan_killiany_tourville_labeling(...)` tool.

303 **Methods**

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

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

310 As in human studies, bias field correction and image denoising are standard preprocessing  
311 steps in improving overall image quality in mouse brain images. The bias field, a gradual  
312 spatial intensity variation in images, can arise from various sources such as magnetic field in-  
313 homogeneity or acquisition artifacts, leading to distortions that can compromise the quality  
314 of brain images. Correcting for bias fields ensures a more uniform and consistent representa-  
315 tion of brain structures, enabling accurate quantitative analysis. Additionally, brain images  
316 are often susceptible to various forms of noise, which can obscure subtle features and affect  
317 the precision of measurements. Denoising techniques help mitigate the impact of noise, en-  
318 hancing the signal-to-noise ratio and improving the overall image quality. The well-known  
319 N4 bias field correction algorithm<sup>24</sup> has its origins in the ANTs toolkit which was imple-  
320 mented and introduced into the ITK toolkit. Similarly, ANTsX contains an implementation  
321 of a well-performing patch-based denoising technique<sup>52</sup> and is also available as a image filter  
322 to the ITK community.

323 **ANTsXNet mouse brain applications**

324 *General notes regarding deep learning training.*

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

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

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

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

346 *Brain extraction.*

347 Similar to human neuroimage processing, brain extraction is a crucial preprocessing step for  
348 accurate brain mapping. Within ANTsXNet, we have created several deep learning networks  
349 for brain extraction for several image modalities (e.g., T1, FLAIR, fractional anisotropy).  
350 Similarly, for the developmental brain atlas work<sup>47</sup> we developed similar functionality for  
351 mouse brains of different modalities and developmental age. All networks use a conven-  
352 tional 2-D U-net architecture<sup>56</sup> and perform prediction in a slice-wise fashion given the  
353 limitations of the acquisition protocols (e.g., missing slices, slice thickness). Currently,  
354 coronal and sagittal networks are available for both E13.5 and E15.5 data and coronal  
355 network for T2-weighted MRI. In ANTsPyNet, this functionality is available in the pro-  
356 gram `antspynet.mouse_brain_extraction(...)`. Even when physical brain extraction is

<sup>357</sup> performed prior to image acquisition, artifacts, such as bubbles or debris, can complicate  
<sup>358</sup> subsequent processing. Similar to the brain extraction networks, a 2-D U-net architecture<sup>56</sup>  
<sup>359</sup> was created to separate the background and foreground.

<sup>360</sup> *Miscellaneous networks: Super-resolution, cerebellum, and hemispherical masking.*

<sup>361</sup> To further enhance the data prior to designing mapping protocols, additional networks were  
<sup>362</sup> created. A well-performing deep back projection network<sup>57</sup> was ported to ANTsXNet and  
<sup>363</sup> expanded to 3-D for various super-resolution applications,<sup>58</sup> including mouse data. Finally,  
<sup>364</sup> features of anatomical significance, namely the cerebellum and hemispherical midline were  
<sup>365</sup> captured in these data using deep learning networks.

## <sup>366</sup> **Intra-slice image registration with missing slice imputation**

<sup>367</sup> Volumetric gene expression slice data was collated into 3-D volumes using ... (ask Jeff).  
<sup>368</sup> Prior to mapping this volume to the corresponding structural data and, potentially, to the  
<sup>369</sup> appropriate template, alignment was improved using deformable registration on contiguous  
<sup>370</sup> slices. However, one of the complications associated with these image data was the un-  
<sup>371</sup> known number of missing slices, the number of consecutive missing slices, and the different  
<sup>372</sup> locations of these missing slices. To handle this missing data problem, we found that data  
<sup>373</sup> interpolation using the B-spline approximation algorithm cited earlier<sup>54</sup> (ANTsPy function:  
<sup>374</sup> `ants.fit_bspline_object_to_scattered_data(...)`). This provided sufficient data in-  
<sup>375</sup> terpolation fidelity to perform continuous slicewise registration. Other possible variants that  
<sup>376</sup> were considered but deemed unnecessary was performing more than one iteration cycling  
<sup>377</sup> through data interpolation and slicewise alignment. The other possibility was incorporating  
<sup>378</sup> the super-resolution technique described earlier. But again, our data did not require these  
<sup>379</sup> additional steps.

## <sup>380</sup> **Image registration**

<sup>381</sup> The ANTs registration toolkit is a complex framework permitting highly tailored so-  
<sup>382</sup> lutions to pairwise image registration scenarios.<sup>59</sup> It includes innovative transformation

models for biological modeling<sup>39,49</sup> and has proven capable of excellent performance.<sup>40,60</sup> Various parameter sets targeting specific applications have been packaged with the different ANTsX platforms, specifically ANTs, ANTsPy, and ANTsR.<sup>25</sup> In ANTsPy, the function `ants.registration(...)` is used to register two (sets) of images where `type_of_transform` is a user-specified option that invokes a specific parameter set. For example `type_of_transform='antsRegistrationSyNQuick[s]'` is an oft-used parameter set.

Initially, linear optimization is initialized with center of (intensity) mass alignment typically followed by optimization of both rigid and affine transforms using the mutual information similarity metric. This was followed by diffeomorphic deformable alignment using symmetric normalization (SyN) with Gaussian<sup>39</sup> or B-spline regularization<sup>49</sup> where the forward transform is invertible and differentiable. The similarity metric employed at this latter stage is typically either neighborhood cross-correlation or mutual information similarity metric. Note that these parameter sets are robust to input image type (i.e., LSFM, Nissl staining, and the various MRI modalities) and are adaptable to mousing image geometry scaling. Further details can be found in the various documentation sources for these ANTsX packages.

## Template generation

ANTsX provides functionality for constructing templates from a set (or multi-modal sets) of input images as originally described<sup>50</sup> and recently used to create the DevCCF templates.<sup>47</sup> An initial template estimate is constructed from an existing subject image or a voxelwise average derived from a rigid pre-alignment of the image population. Pairwise registration between each subject and the current template estimate is performed using the Symmetric Normalization (SyN) algorithm.<sup>39</sup> The template estimate is updated by warping all subjects to the space of the template, performing a voxelwise average, and then performing a “shape update” of this latter image by warping it by the average inverse deformation, thus yielding a mean image of the population in terms of both the intensity and shape.

409 **Continuous developmental velocity flow transformation model**

410 Given multiple, linearly or non-linearly ordered point sets where individual points across  
411 are in one-to-one correspondence, we developed an approach for generating a velocity flow  
412 transformation model to describe a time-varying diffeomorphic mapping as a variant of the  
413 inexact landmark matching solution of Joshi and Miller.<sup>61</sup> Integration of the resulting velocity  
414 field can then be used to describe the displacement between any two time points within this  
415 time-parameterized domain. Regularization of the sparse correspondence between point sets  
416 is performed using a generalized B-spline scattered data approximation technique,<sup>54</sup> also  
417 developed by the ANTsX developers and contributed to ITK.

418 To apply this methodology to the developmental templates,<sup>47</sup> we coalesced the manual par-  
419 cellations of the developmental templates into 26 common anatomical regions (13 per hemi-  
420 sphere). We then used these regions to generate invertible transformations between suc-  
421cessive time points. Specifically each label was used to create a pair of single region images  
422 resulting in 26 pairs of “source” and “target” images. The multiple image pairs were used  
423 to iteratively estimate a diffeomorphic pairwise transform. Given the seven atlases E11.5,  
424 E13.5, E15.5, E18.5, P4, P14, and P56, this resulted in 6 sets of transforms between suc-  
425cessive time points. Given the relative sizes between atlases, on the order of  $10^6$  points were  
426 randomly sampled labelwise in the P56 template space and propagated to each successive  
427 atlas providing the point sets for constructing the velocity flow model. Approximately 200  
428 iterations resulted in a steady convergence based on the average Euclidean norm between  
429 transformed point sets. Ten integration points were used and point sets were distributed  
430 along the temporal dimension using a log transform for a more evenly spaced sampling.  
431 Further details including links to data and scripts to reproduce our reported results is found  
432 in the associated GitHub repository.

433 **Visualization**

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

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

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

444 **References**

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

- 469 13. Wang, Q. *et al.* The allen mouse brain common coordinate framework: A 3D reference  
atlas. *Cell* **181**, 936–953.e20 (2020).
- 470
- 471 14. Johnson, G. A. *et al.* Waxholm space: An image-based reference for coordinating  
mouse brain research. *Neuroimage* **53**, 365–72 (2010).
- 472
- 473 15. Oguz, I., Zhang, H., Rumple, A. & Sonka, M. RATS: Rapid automatic tissue segmen-  
474 tation in rodent brain MRI. *J Neurosci Methods* **221**, 175–82 (2014).
- 475 16. Sawiak, S. J., Picq, J.-L. & Dhenain, M. Voxel-based morphometry analyses of in  
476 vivo MRI in the aging mouse lemur primate. *Front Aging Neurosci* **6**, 82 (2014).
- 477 17. Ashburner, J. SPM: A history. *Neuroimage* **62**, 791–800 (2012).
- 478
- 479 18. Modat, M. *et al.* Fast free-form deformation using graphics processing units. *Comput  
480 Methods Programs Biomed* **98**, 278–84 (2010).
- 481 19. Tyson, A. L. *et al.* Accurate determination of marker location within whole-brain  
482 microscopy images. *Sci Rep* **12**, 867 (2022).
- 483 20. Pallast, N. *et al.* Processing pipeline for atlas-based imaging data analysis of struc-  
484 tural and functional mouse brain MRI (AIDAmri). *Front Neuroinform* **13**, 42 (2019).
- 485 21. Jenkinson, M., Beckmann, C. F., Behrens, T. E. J., Woolrich, M. W. & Smith, S. M.  
486 FSL. *Neuroimage* **62**, 782–90 (2012).
- 487 22. Yeh, F.-C., Wedeen, V. J. & Tseng, W.-Y. I. Generalized q-sampling imaging. *IEEE  
488 Trans Med Imaging* **29**, 1626–35 (2010).
- 489 23. Jorge Cardoso, M. *et al.* STEPS: Similarity and truth estimation for propagated  
490 segmentations and its application to hippocampal segmentation and brain parcelation.  
*Med Image Anal* **17**, 671–84 (2013).
- 491 24. Tustison, N. J. *et al.* N4ITK: Improved N3 bias correction. *IEEE Trans Med Imaging*  
492 **29**, 1310–20 (2010).

- 493 25. Tustison, N. J. *et al.* The ANTsX ecosystem for quantitative biological and medical  
imaging. *Sci Rep* **11**, 9068 (2021).
- 494
- 495 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).
- 496
- 497 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).
- 498
- 499 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).
- 500
- 501 29. Cox, R. W. AFNI: What a long strange trip it's been. *Neuroimage* **62**, 743–7 (2012).
- 502
- 503 30. Ni, H. *et al.* A robust image registration interface for large volume brain atlas. *Sci  
Rep* **10**, 2139 (2020).
- 504
- 505 31. Jin, M. *et al.* SMART: An open-source extension of WholeBrain for intact mouse  
brain registration and segmentation. *eNeuro* **9**, (2022).
- 506
- 507 32. Fürth, D. *et al.* An interactive framework for whole-brain maps at cellular resolution.  
*Nat Neurosci* **21**, 139–149 (2018).
- 508
- 509 33. Negwer, M. *et al.* FriendlyClearMap: An optimized toolkit for mouse brain mapping  
and analysis. *Gigascience* **12**, (2022).
- 510
- 511 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).
- 512
- 513 35. Carey, H. *et al.* DeepSlice: Rapid fully automatic registration of mouse brain imaging  
to a volumetric atlas. *Nat Commun* **14**, 5884 (2023).
- 514

- 515 36. Bajcsy, R. & Broit, C. Matching of deformed images. in *Sixth International Conference on Pattern Recognition (ICPR'82)* 351–353 (1982).
- 516
- 517 37. Bajcsy, R. & Kovacic, S. Multiresolution elastic matching. *Computer Vision, Graphics, and Image Processing* **46**, 1–21 (1989).
- 518
- 519 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).
- 520
- 521 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).
- 522
- 523 40. Klein, A. *et al.* Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration. *Neuroimage* **46**, 786–802 (2009).
- 524
- 525 41. Murphy, K. *et al.* Evaluation of registration methods on thoracic CT: The EMPIRE10 challenge. *IEEE Trans Med Imaging* **30**, 1901–20 (2011).
- 526
- 527 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).
- 528
- 529 43. Wang, H. *et al.* Multi-atlas segmentation with joint label fusion. *IEEE Trans Pattern Anal Mach Intell* **35**, 611–23 (2013).
- 530
- 531 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).
- 532
- 533 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).
- 534

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

- 557 57. Haris, M., Shakhnarovich, G. & Ukita, N. Deep back-projection networks for super-  
558 resolution. in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recog-*  
558 *nition* 1664–1673 (2018). doi:[10.1109/CVPR.2018.00179](https://doi.org/10.1109/CVPR.2018.00179).
- 559 58. Avants, B. B. *et al.* Concurrent 3D super resolution on intensity and segmentation  
560 maps improves detection of structural effects in neurodegenerative disease. *medRxiv*  
560 (2023) doi:[10.1101/2023.02.02.23285376](https://doi.org/10.1101/2023.02.02.23285376).
- 561 59. Avants, B. B. *et al.* The Insight ToolKit image registration framework. *Front Neu-*  
562 *roinform* **8**, 44 (2014).
- 563 60. Avants, B. B. *et al.* A reproducible evaluation of ANTs similarity metric performance  
564 in brain image registration. *Neuroimage* **54**, 2033–44 (2011).
- 565 61. Joshi, S. C. & Miller, M. I. Landmark matching via large deformation diffeomor-  
566 phisms. *IEEE Trans Image Process* **9**, 1357–70 (2000).