

## Supplementary Tables

**Supplementary Table 1. Download links for online resources by Allen Institute for Brain Science**

File name/contents	Description
average_template_25.nrrd, average_template_50.nrrd, average_template_100.nrrd	<p>Average template (AT) of the mouse brain in NRRD 16-bit UINT format (<a href="http://teem.sourceforge.net/nrrd/">http://teem.sourceforge.net/nrrd/</a>) with spatial resolution of 25, 50, and 100 <math>\mu\text{m}</math> isovoxel. Volume dimensions are <math>528 \times 320 \times 456</math>, <math>264 \times 160 \times 228</math>, and <math>132 \times 80 \times 114</math> in PIR (+x = posterior, +y = -inferior, +z = right) coordinate, respectively, with an image-origin at anterior upper left corner long unit in micrometers.</p> <p>DL: <a href="http://download.alleninstitute.org/informatics-archive/current-release/mouse_ccf/average_template/">http://download.alleninstitute.org/informatics-archive/current-release/mouse_ccf/average_template/</a></p> <p>Ref: <a href="http://help.brain-map.org/download/attachments/2818169/MouseCCF.pdf?version=1&amp;modificationDate=1432933021016&amp;api=v2">http://help.brain-map.org/download/attachments/2818169/MouseCCF.pdf?version=1&amp;modificationDate=1432933021016&amp;api=v2</a></p>
annotation_10.nrrd, annotation_25.nrrd, annotation_50.nrrd, annotation_100.nrrd	<p>Annotation volume (AV) of the mouse brain in NRRD 32-bit UINT format with spatial resolution of 10, 25, 50, and 100 <math>\mu\text{m}</math> isovoxel. Volume dimensions are <math>1320 \times 800 \times 1140</math>, <math>528 \times 320 \times 456</math>, <math>264 \times 160 \times 228</math>, and <math>132 \times 80 \times 114</math> in PIR coordinates. AVs with higher spatial resolution include additional brain structures [acronym (ID) for a brain structure]: c (10), tspd (1), and RSPd4 (2).</p> <p>DL: <a href="http://download.alleninstitute.org/informatics-archive/current-release/mouse_ccf/annotation/ccf_2017/">http://download.alleninstitute.org/informatics-archive/current-release/mouse_ccf/annotation/ccf_2017/</a></p> <p>Ref: <a href="http://help.brain-map.org/display/mouseconnectivity/API">http://help.brain-map.org/display/mouseconnectivity/API</a></p>
1.json	<p>A text file in JSON format that defines annotation ontology (AO) of the mouse brain (Structure graph ID: 1).</p> <p>DL: <a href="http://api.brain-map.org/api/v2/structure_graph_download/1.json">http://api.brain-map.org/api/v2/structure_graph_download/1.json</a></p> <p>Ref: <a href="http://help.brain-map.org/display/api/Atlas+Drawings+and+Ontologies">http://help.brain-map.org/display/api/Atlas+Drawings+and+Ontologies</a></p>
Gene expression energy	<p>Volume data for gene expression energy for approximately 20,000 genes in the adult-mouse brain in MetaImage 32-bit FLOAT format (<a href="https://github.com/Kitware/MetaIO">https://github.com/Kitware/MetaIO</a>) with spatial resolution of 200 <math>\mu\text{m}</math> isovoxel. Their respective volume dimensions are <math>67 \times 41 \times 58</math> in PIR coordinates.</p> <p>Ref: <a href="http://help.brain-map.org/display/mousebrain/API">http://help.brain-map.org/display/mousebrain/API</a></p>
Axonal projection density	<p>Volume data for axonal projection density in the brain in NRRD 32-bit FLOAT format with spatial resolutions of 10, 25, 50, and 100 <math>\mu\text{m}</math> isovoxel. Their respective volume dimensions are <math>1320 \times 800 \times 1140</math>, <math>528 \times 320 \times 456</math>, <math>264 \times 160 \times 228</math>, and <math>132 \times 80 \times 114</math> in PIR coordinates.</p> <p>Ref: <a href="http://help.brain-map.org/display/mouseconnectivity/API">http://help.brain-map.org/display/mouseconnectivity/API</a></p>

DL, download link; Ref, reference site. Data were retrieved June 12, 2019.

**Supplementary Table 2. Steps to construct FAA of the mouse brain**

Step	Code/procedure	Input	Primary output
0	Prepare_AObaseAVbase.ipynb	Original resources at AIBS <ul style="list-style-type: none"> <li>- 1.json (anatomical ontology text-file)</li> <li>- annotation_100.nrrd (annotation volume)</li> </ul> <p>This “preprocessing” eliminates <i>destructive</i> brain structures (nodes) in the original resources, performed by a Python code without user inputs.</p>	FAAbase comprises <ul style="list-style-type: none"> <li>- AObase.json</li> <li>- AVbase.nrrd</li> </ul>
1	Combine brain structures	- AObase.json	- AObase_c.json
<p>This first step combines leaf-nodes to obtain a new leaf-node with larger volume while maintaining anatomical hierarchy by manually editing a JSON-formatted text file, AObase.json. Specifically, copy AObase.json and rename it to AObase_c.json. Then, to combine all descendent nodes of an <i>inner</i> node, delete all contents within brackets [] of a key “children” for the inner node. This would be facilitated by a text editor such as Vim (<a href="https://www.vim.org/">https://www.vim.org/</a>) with functionality to jump to matching brackets. To support this further, a zoomable plot of an anatomical hierarchy in AObase_c.json is provided as an HTML file using D3.js (<a href="https://d3js.org/">https://d3js.org/</a>) (e.g. Fig. 1d). To open this HTML-file, it is recommend that “Web Server for Chrome” (<a href="https://github.com/kzahel/web-server-chrome">github.com/kzahel/web-server-chrome</a>) be used. It is available at the Chrome web store because direct access to a local file is prohibited for security reasons in a recent web-browser, e.g. Firefox after ver. 68.0.</p>			
2	Divide_nodes.ipynb	A user-specified text-based information on <ul style="list-style-type: none"> <li>- <b>AObase_c.json</b></li> <li>- <b>Target_ROI_IDs</b> (brain structures) defined in AObase_c.json for dividing nodes</li> <li>- <b>ExpID</b> defined at AIBS to specify a gene of interest</li> <li>- <b>Acronyms</b> for brain structures to specify source- and target-node for axonal projection</li> </ul> <p>This second step divides leaf nodes based on gene expression and axonal fiber projection using a Python code with a user-specified text-based information, resulting in a flexible annotation atlas (FAA) that comprises an annotation ontology (AO) text-file and an annotation volume (AV). Five additional modifications are performed during this step: 1) assigning different IDs for homotopic nodes in the right and left side of the brain to make annotation atlas bilateral, e.g. Fig. 2c; 2) remapping IDs for brain structures in the original AO and AV to be in the range of 16-bit UINT; 3) transforming a format of the original AV from NRRD to NIfTI-1 (<a href="https://nifti.nimh.nih.gov/">https://nifti.nimh.nih.gov/</a>) because some programs such as MRICron and SPM do not support NRRD format; 4) modifying image orientation from posterior-inferior-right (PIR) to right-anterior-superior (RAS) as is widely used in NIfTI standard; and 5) setting spatial origin of AV and AT to the bregma referring to the mouse brain atlas (Paxinos and Franklin, 2001), from that at top left corner of a volume image. This step enables specification of the location of the brain structure using mm-coordinates, as in human MRI study. There are 6 nodes located only in the midline: Medulla, behavioral state related (MY-sat), its child nodes (nucleus raphe magnus (RM), pallidus (RPA), and obscurus (RO)), vascular organ of the lamina terminalis (OV), and Edinger-Westphal nucleus (EW). These nodes were assigned to the right side of the brain during the first step. For reconstruction of FAA, share the text-based information shown in bold face above: AObase_c.json, Target_ROI_IDs, ExpID, and Acronyms (Fiber_from and Fiber_to).</p>	FAA comprises <ul style="list-style-type: none"> <li>- <b>AO_LR_remapID.json</b></li> <li>- <b>AV_LR_remapID_RAS.nii</b></li> </ul>

**Supplementary Table 3. Subfunctions in Prepare\_AObaseAVbase.ipynb for preprocessing**

Step	Code	Input	Output
1	Add_VC_to_AO.ipynb	- 1.json - annotation_100.nrrd	- 1_VC.json
	A voxel count (VC) information for a brain structure in AV is appended to an anatomical ontology file.		
2	Prune_leaf_ROI_wo_VC_in_AO.ipynb	- 1_VC.json	- 1_VC_pruned.json
	Destructive leaf nodes (leaf nodes without voxel counts in AV) are deleted from AO.		
3	Divide_internal_ROI_with_VC_in_AO.ipynb	- 1_VC_pruned.json	- 1_VC_pruned_divided.json - dividedIDs.csv
	Destructive inner nodes (inner nodes with voxel counts in AV) are divided, leading to a new leaf-node with new ID. Its name and acronym are the same as the original ones suffixed respectively with “_peripheral” and “_peri”.		
4	Update_ID_in_AV_to_reflect_divided_AO.ipynb	- annotation_100.nrrd - dividedIDs.csv	- AVbase.nrrd
	IDs in AV is updated according to an anatomical ontology file, 1_VC_pruned_divided.json.		
5	Add_VC_to_AO.ipynb	- AVbase.nrrd - 1_VC_pruned_divided.json	- AObase.json
	Update voxel-count information in the AO according to AVbase.nrrd.		

**Supplementary Table 4. Subfunctions in Divide\_nodes.ipynb for dividing leaf-nodes**

Step	Code	Input	Output
0-1	Get_ID_parentID_pairs.ipynb This prepares pairs of IDs for a node and its parent node in AO.	- AObase.json	- ID_parentID_AObase.csv
0-2	Replace_ID_with_its_parent_ID_to_reflect_combined_AO.ipynb  IDs in AV is updated to reflect AObase.	- AVbase.nrrd - <b>AObase_c.json</b> - ID_parentID_AObase.csv	- AVbase_c.nrrd
1-1	Divide_ROI_with_gene_expression_data.ipynb  A selected node is divided to two based on the gene expression density.	- <b>ExpID</b> (74881161 for <i>Wfs1</i> gene) - <b>Target_ROI_ID</b> (382 for hippocampal CA1) - AVbase_c.nrrd	- AV_target_ROI_ID_382_gene_74881161.nrrd - figures for gene expression
1-2	Update_AV_according_to_gene_expression.ipynb  ID for a divided node with higher gene expression is modified.	- AV_target_ROI_ID_382_gene_74881161.nrrd	- AVbase_c_g.nrrd
1-3	Update_AO_according_to_gene_expression.ipynb  AObase_c.json is updated reflecting AVbase_c_g.nrrd. Node with the maximum ID is regarded as a node with high gene expression. Acronym for a node with high or low gene expression is suffixed with “_geneH” or “_geneL”. Node name for high gene expression is suffixed with “_gene+ExpID”.	- ExpID - Target_ROI_ID - AObase_c.json - AVbase_c_g.nrrd	- AObase_c_g_woVC.json
1-4	Add_VC_to_AO.ipynb	- AVbaes_c_g.nrrd - AObase_c_g_woVC.json	- AObase_c_g.json

2-1	Divide_ROI_with_fiber_innervation.ipynb	<ul style="list-style-type: none"> <li>- <b>Target_ROI_ID</b> (672 for Caudoputamen)</li> <li>- <b>Fiber_from</b> (AI, agranular insula)</li> <li>- <b>Fiber_to</b> (CP, caudoputamen)</li> <li>- AVbase_d_g.nrrd</li> </ul>	<ul style="list-style-type: none"> <li>- AV_target_ROI_ID_672_fiber_from_AI_to_CP.nrrd</li> <li>- figures for fiber innervation</li> </ul>
A node is divided depending on the density of fiber innervation from a source to a destination node. Injection into the right hemisphere is used for this study.			
2-2	Update_AV_according_to_fiber_innervation.ipynb	<ul style="list-style-type: none"> <li>- Target_ROI_ID (672)</li> <li>- Fiber_from (AI)</li> <li>- Fiber_to (CP)</li> <li>- AVbase_c_g.nrrd</li> <li>- AV_target_ROI_ID_672_fiber_from_AI_to_CP.nrrd</li> </ul>	<ul style="list-style-type: none"> <li>- AVbase_c_g_f.nrrd</li> </ul>
This updates ID for a node with higher fiber innervation in AV.			
2-3	Update_AO_according_to_fiber_innervation.ipynb	<ul style="list-style-type: none"> <li>- Target_ROI_ID (672)</li> <li>- Fiber_from (AI)</li> <li>- Fiber_to (CP)</li> <li>- AObase_c_g.json</li> <li>- AVbase_c_g_f.nrrd</li> </ul>	<ul style="list-style-type: none"> <li>- AObase_c_g_f_woVC.json</li> </ul>
AObase_c_g.json is updated reflecting AVbase_c_g_f.nrrd. Node with the maximum ID is regarded as a node with high fiber innervation. Acronym for a node with high or low fiber innervation is suffixed with “_fiberH” or “_fiberL”. Node name for high fiber innervation is suffixed with “_Fiber_from_Fiber_to”.			
2-4	Add_VC_to_AO.ipynb	<ul style="list-style-type: none"> <li>- AVbase_c_g_f.nrrd</li> <li>- AObase_c_g_f_woVC.json</li> </ul>	<ul style="list-style-type: none"> <li>- AObase_c_g_f.json</li> </ul>
3-1	Divide_left_right_AV	<ul style="list-style-type: none"> <li>- AVbase_c_g_f.nrrd</li> </ul>	<ul style="list-style-type: none"> <li>- AVbase_c_g_f_LR.nrrd</li> </ul>
This makes an annotation volume bilateral by adding a constant value to IDs on the right side of the brain.			
3-2	Prepare_AO_LR.ipynb	<ul style="list-style-type: none"> <li>- AObase_c_g_f.json</li> </ul>	<ul style="list-style-type: none"> <li>- AO_L.json</li> <li>- AO_R.json</li> </ul>
Anatomical ontology text file is updated to be bilateral. Node name is suffixed with “_L” or “_R”.			

3-3	Merge_AO_LR.ipynb	- AO_L.json - AO_R.json - AO_LR_wo_VC_TEMPLATE.json	- AO_LR_wo_VC.json
This merges json files for the right and left side of the brain. Nodes “root_peri_L” and “root_peri_R” were dismissed from an annotation ontology json-file because they were not assigned to any structures such as grey, fiber tracts, or ventricular systems.			
3-4	Add_VC_to_AO.ipynb	- AO_LR_wo_VC.json - AVbase_c_g_f_LR.nrrd	- AO_LR.json
4-1	Get_ID_parentID_pairs.ipynb	- AO_LR.json	ID_parentID_LR.csv
4-2	Reassign_ID.ipynb	- AO_LR.json - ID_parentID_LR.csv	- AO_LR_remapID.json - remapIDpairs.csv
IDs in AO is remapped so that the range decreases from 32bit-UINT to 16bit-UINT.			
4-3	Update_ID_in_AV_to_reflect_reassignedID.ipynb	- AVbase_c_g_f_LR.nrrd - remapIDpairs.csv	- AV_LR_remapID.nrrd
5	Transform_AllenImage_from_NRRD_toNifti.ipynb	- AV_LR_remapID.nrrd	- AV_LR_remapID_RAS.nii
Image orientation was modified from PIR to RAS by assigning s-form and q-form code in Nifti-header. The length unit was changed from micrometers to millimeters. The scale dimension unit was $\times 10$ of naive space. ITK-snap reads only q-form, and SPM outputs only s-form. For preparation of FAAssegment without dividing nodes and without making it double-sided, skip steps 1-1 to 3-3 and perform only steps 0-1, 0-2, 3-4 to 5. In this case, rename AObase_c.json and AVbase_c.nrrd to AO_LR_wo_VC.json and AVbase_c_g_f_LR.nrrd, respectively, before performing a step 3-4. See FAA reconstruction information at <a href="https://github.com/ntakata/flexible-annotation-atlas/tree/master/FAAs/FAAsegment/">https://github.com/ntakata/flexible-annotation-atlas/tree/master/FAAs/FAAsegment/</a> .			