UNIVERSITY OF CALIFORNIA
SANTA CRUZ

BELL JAR: AN AUTOMATED ANALYSIS TOOL FOR MOUSE BRAIN HISTOLOGY

A thesis submitted in partial satisfaction
of the requirements of the degree of

MASTER OF SCIENCE

in

MOLECULAR, CELL, AND DEVELOPMENTAL BIOLOGY

by

Alec L.R. Soronow

September 2024

The Thesis of Alec Soronow is
approved:

Professor Euiseok Kim

Professor David Feldheim

Professor Bin Chen

Peter Biehl
Vice Provost and Dean of Graduate Studies

TABLE OF CONTENTS

LIST OF FIGURES **iv**

ABSTRACT **v**

ACKNOWLEDGEMENTS **vi**

1 Introduction **1**
- 1.1 Current Problems with Tissue Alignment in Neuroanatomy 1
- 1.2 Developing the Platform 4

2 Materials and Methods **5**
- 2.1 Virus Preparation 5
- 2.2 Animal Surgery and Virus Injection 5
- 2.3 Histology and Image Analysis 6
- 2.4 Statistical Methods 7
- 2.5 Hardware Specifications 7
- 2.6 Analysis of ISH Signals 8
- 2.7 Model Generation and Training 8
- 2.8 Composite Transformation Routine for Alignment 9

3 Results 12
- 3.1 Predicting Slice Positions and Fine Tuning 13
- 3.2 Aligning Experimental Tissue to the Reference Brain Atlas 14
- 3.3 Detecting Signals of Interest 21

4 Conclusions and Discussion **24**
- 4.1 Bell Jar Advances Histological Analysis through Automation 24
- 4.2 Bell Jar Enhances Precision in Cell Detection 25
- 4.3 Implications for Neuroscience and Beyond 25
- 4.4 Future Directions 25
- 4.5 Limitations of the Current Work 26

BIBLIOGRAHY **28**

LIST OF FIGURES

3.1	Bell Jar is a pipeline for aligning histology and integrating datasets	13
3.2	Bell Jar accurately assigns anatomical labels to experimental tissues	17
3.3	Bell Jar uses object detection to detect cells automatically	21
3.4	Bell Jar cell counting is comparable to manual counting	23

ABSTRACT

Alec Soronow: Bell Jar: An Automated Analysis Tool for Mouse Brain Histology

For comprehensive anatomical analysis of a mouse brain, accurate and efficient registration of the experimental brain samples to a reference atlas is necessary. Here, I introduce Bell Jar, an automated solution that can align and annotate tissue sections with anatomical structures from a reference atlas and detect fluorescent signals with cellular resolution (e.g., cell bodies or nuclei). Bell Jar utilizes machine learning-based non-linear image registration to achieve precise alignments, even with damaged sample tissues. While user input remains required for fine-tuning section matches, the platform streamlines the process, aiding rapid analyses in high-throughput neuroanatomy studies. As a standalone desktop application with a user-friendly interface, Bell Jar's performance surpasses traditional manual and existing automated methods and can improve the reproducibility and throughput of histological analyses.

ACKNOWLEDGEMENTS

First and foremost, I would like to acknowledge my mentor and advisor, Prof. Euiseok Kim. He has been a constant source of encouragement, advice, and inspiration. He ensured I had everything I needed to succeed at every step of the way.

I would also like to thank Matt Jacobs for his friendship and commitment to rigorous science. I never wrote any code that did not go through Matt for the first round of testing. Additionally, I want to thank Richard Dickson for asking hard questions about my work and helping me foster a deeper understanding of my projects. He also generously provided many experiments to be used in developing my projects.

Last but not least, I want to thank J. Lu, D. Feldheim, B. Colquitt, and D. Turner-Evans for reading the manuscript, A. Shariati for reading the previous manuscript, and P. Pham for mouse husbandry and histology. I want to thank my committee D. Feldheim and B. Chen for their support. I also thank H. Lee, J. Ratliff, R. Marek, and L. Hermelin for testing the current and previous versions of Bell Jar and providing feedback. I thank R. Marek for providing experimental tissue images for testing. I acknowledge technical support from Benjamin Abrams, UCSC Life Sciences Microscopy Center, RRID: SCR_021135. I acknowledge support from the UCSC start-up fund, the Whitehall Foundation, the Hellman Fellows Program, the E. Matilda Ziegler Foundation for the Blind, NIH RF1MH132591 and R01NS128771 (E.J.K.), the Koret scholar program (A.L.R.S.), and an NIH IRACDA Postdoctoral Training Grant (R.G.D.).

1 Introduction

Assigning anatomical regions to mouse brain sections (known as alignment) based on a reference atlas is a challenging and necessary task for neuroscience research. Before the publication of digital reference atlases (e.g., the Allen Mouse Brain Reference Atlas[1]), alignment relied on paper atlases such as Paxinos and Franklin's *The Mouse Brain in Stereotaxic Coordinates*[2], which contains representative tissue section images using Nissl and Acetylcholinesterase (AChE) staining, and the corresponding line drawing of the anatomical boundaries of brain regions. The experimenter would then, by visual inspection, identify the reference image that is the closest to the experimental brain section and annotate the regions of interest accordingly. This method is time-consuming and is subject to inter-rater bias. Importantly, imperfect sectioning angles and other distortions introduced through histological processing or factors such as animal strain, age, sex, and size make the experimental tissue sections non-identical to the reference images. This may result in imprecise registration and annotation of brain structures in the experimental tissues. As neuroscience moves towards high-throughput experiments, automated methods are necessary to achieve accurate and reproducible results in a reasonable amount of time. Several tools developed to address this problem[3–7] rely on rigid, affine transforms of the digital atlas to assign regions but remain problematic when aligning damaged, deformed, or otherwise sub-optimal specimens often encountered in research. To overcome this obstacle, I created Bell Jar, a software tool that automatically aligns sections to a

reference atlas with a deformable transformation, enabling all types of tissue to be aligned.

1.1 Current Problems with Tissue Alignment in Neuroanatomy

Due to the technological advances of microscopy and computer systems, large image datasets are becoming increasingly common among neuroscience researchers[8–12], and there is a need to automate the analysis of this data. There can be hundreds of high-resolution brain tissue sections for analysis of whole-brain histology experiments with even a few animals[9–11]. Researchers then must interpret results manually to analyze experimental signals (e.g., count labeled cells) and demarcate anatomical regions (e.g., draw borders between different structures) to interpret results manually for each section. Manual annotation across large datasets is time-consuming and prone to inter-rater bias[3,13]. The bias is introduced in the traditional application of region boundaries by an expert with the aid of a paper atlas[14]. Each expert rater has to make a personal determination based on cues in the tissue section, such as cytoarchitecture or landmark structures. However, while these heuristics work for ideal tissue sections, they become difficult to apply for damaged or irregular tissue sections.

With advances in brain imaging and the collection of large datasets, digital reference atlases have become increasingly detailed and accessible. The Allen Brain Common Coordinate Framework[1] (CCF) provides high-resolution atlas volumes and corresponding region annotations (including a coordinate system and structure

ontologies) representing the average adult mouse brain. A program can align an experimental tissue section into the reference atlas with a high-resolution digital intermediate tissue section as a guide[3,4,6] , assuming they share key anatomical features. With the resulting alignment, the exact annotations of the digital tissue section can be used to determine the regions in the experimental tissue. With these boundaries, signals can be further analyzed within those regions via other software.

Several programs have been developed for alignment and signal detection[3,4,6,13] and have seen significant usage throughout the neuroscience community, but they face limitations in their ability to align experimental tissues. While implementation details vary, they use the same principle discussed above: each performs some user-defined affine[3,6,15] or user-defined spline-based transformation[15] of the digital atlas section to match the experimental tissue section. Affine transformations cannot accurately represent the extensive deformations arising from differences in the mounting and sectioning of individual experiments due to their inability to represent complex elastic deformations. Tools that feature manual landmark-based deformable transformations (user-defined spline-transformations) of alignments address this issue.

Still, the added time to manually tune hundreds of sections this way limits their utility. In some cases, this is a semi-automated user-driven process[3,6]. In others, it is automated based on deep learning or other heuristics[5,13]. Despite their efficacy, these tools still achieve limited accuracy on challenging tissues and cannot overcome

complex deformations in tissues. Many of them also require data to be two-photon microscopy[16,17] , which may be inaccessible or non-applicable to some neuroscience labs. Additionally, these tools are built on platforms that are consistently difficult to maintain and execute across platforms or as scripts in other software. The lack of maintenance for these platforms hinders their usage in research for labs that want to integrate high throughput analysis. Some of these tools are no longer usable on modern systems or not maintained altogether. I sought to create a platform that any researcher could learn and use, regardless of computational experience.

1.2 Developing the Platform

To address these limitations, I created Bell Jar. A standalone cross-platform desktop software combines the ability to automatically register experimental tissues to the Allen Brain CCF (common coordinate framework) and quantify experimental signals based on the alignment, with minimal requirement for computational expertise.

Bell Jar's UI (user interface) and backend were built on the Electron.js framework[18] to leverage web applications' ease of development, powerful frameworks, and cross-platform compatibility. Bell Jar's tools are written in Python as individual scripts. The software hosts its own standalone Python installation, keeping its dependencies separate from the host installations. These features let users download and start using the software with minimal configuration. Bell Jar provides various tools for histology

analysis, with its functionality divided into independent tools: alignment, detection, and counting.

2 Materials and Methods

2.1 Virus Preparation

All AAVs and EnvA+RV*dG* were produced by the Salk Viral Core GT3: AAVretro-nef-lox66/71-tTA ($1.77X10^{12}$ GC/ml), AAV8-TRE-DIO-oG-WPRE ($5.92X10^{12}$ GC/ml), AAV8-TRE-DIO-eGFP-T2A-TVA ($7.00X10^{13}$ GC/ml), AAV8-DIO-TVA66T-2A-eGFP-2A-oG ($5.06X10^{13}$ GC/ml), and EnvA+RV*dG*-mCherry ($1.07X10^{9}$ Infectious Unit (IU)/ml).

2.2 Animal Surgery for Virus Injection

For rabies tracing experiments, SepW1-Cre NP39 or Tlx3-Cre PL56 mice received AAV helper injections at postnatal day (P)80-P100. Mice were anaesthetized with 100 mg/kg of ketamine and 10 mg/kg of xylazine cocktail via intra-peritoneal injections and mounted in a stereotax (RWD instruments) for surgery and stereotaxic injections. 50nl of AAVretro-nef-lox66/71-tTA was injected into the center of medial secondary visual cortex (V2M), using the following coordinates: 1.8 mm caudal, 1.6 mm lateral relative to lambda and 0.5-0.7 mm ventral from the pia. A 50 nl mixture of AAV8-TRE-DIO-oG-WPRE and AAV8-TRE-DIO-eGFP-T2A-TVA or a 50 nl AAV8-DIO-TVA66T-2A-eGFP-2A-oG was injected into the center of the primary visual cortex (V1), using the following coordinates: 3.4 mm caudal, 2.6 mm lateral relative to

bregma and 0.5-0.7 mm ventral from the pia. We injected AAVs using air pressure by 1ml syringe with 18G tubing adaptor and tubing. To prevent virus backflow, the pipette was left in the brain for 5-10 minutes after completion of injection. Two or three weeks after AAV helper injection, 100-200 nl of EnvA+RV*dG*-mCherry were injected into the same site in V1 using 1ml syringe-mediated air pressure. Mice were housed for seven days to allow for trans-synaptic rabies spread and fluorescent protein expression.

2.3 Histology and Image Analysis

Brains were harvested after trans-cardiac perfusion using phosphate-buffered saline (PBS) followed by 4% paraformaldehyde (PFA). Brains were dissected out from skulls and post-fixed with 2% PFA and 15% sucrose in PBS at 4°C overnight, and then immersed in 30% sucrose in PBS at 4°C before sectioning. Using a freezing microtome, 50μm coronal brain sections were cut and stored in PBS with 0.01% sodium azide at 4°C. To enhance eGFP and dsRed signals, free-floating sections were incubated at 4°C for 16-48 hours with goat anti-GFP (1:1000; Rockland 600-101-215) and rabbit anti-dsRed (1:500; Clontech 632496) primary antibodies in PBS/0.5% normal donkey serum/0.1% Triton-X 100, followed by the appropriate secondary antibodies conjugated with Alexa 488 or 568 (Invitrogen). Sections were counterstained with 10μM DAPI in PBS for 30 min to visualize cell nuclei. Immunostained tissue sections were mounted on slides with Polyvinyl alcohol mounting medium containing DABCO and allowed to air-dry overnight.

All sections were scanned with a 10x objective on a Zeiss AxioImager Z2 Widefield Microscope. Scanned images were first processed in Zeiss ZEN software using their extended focal imaging algorithm. Subsequent image files were processed and analyzed by Bell Jar, ilastik or NIH ImageJ (FIJI).

2.4 Statistical Methods

Kruskal-Wallis one-way analysis from the SciPy[22] of variance was used to analyze the significance of the difference between the alignment methods' fit metrics, followed by post-hoc *Dunn* to find statistical significances. Pearson's product-moment correlation coefficient was used to calculate the correlation and significance for the various counting methods to manual data.

2.5 Hardware Specifications

I performed all network training on a Windows 11 machine with an AMD Ryzen 9 7950X processor and an Nvidia RTX A5000 GPU. The minimum specifications we recommend for running Bell Jar are: an 8th generation Intel or Ryzen 5 (2017 onwards) processor and a GTX 1060 (6GB) GPU. The minimum GPU memory required for Bell Jar is 4GB, but we recommend at least 6GB for optimal performance.

2.6 Analysis of ISH Signals

All images were collected from the Allen Mouse Brain Reference Atlas[20] transgenic characterization dataset via their public application programming interface. The experiments were individually aligned using Bell Jar's alignment workflow. Images were first inverted, then converted to 8-bit grayscale, and ISH signals were thresholded at a constant value (125) across all experiments to create binary masks of signal in each section. The number of thresholded pixels in each brain region was then quantified using the Bell Jar alignments.

2.7 Model Generation and Training

All YoloV8[26] models were trained using the large model weights for 200 epochs and with all recommended default hyperparameters. The predictor ResNet-101[21] architecture was based on the architecture described in the original work with some key differences. Inputs to the network first undergo Sobel filtering to enhance the edges of anatomical features. The network's final output dense layer is also modified to produce three normalized scalars representing our X-cut angle, Y-cut angle, and anterior-posterior position. I normalized these values to aid training and generalization. Normalization of the vectors was done by assuming -10 degrees to be the minimum cut angle in either dimension and 10 degrees to be the maximum cut angle in either dimension. Likewise, for the anterior-posterior position, I based this off the indices of the digitally coronally sectioned Allen Mouse Brain Reference

Atlas[1,20] 10μm Nissl volume, with 0 being the minimum position and 1324 being the maximum.

2.8 Composite Transformation Routine for Alignment

The composite transformation routine for aligning tissue sections to a reference atlas through a two-stage registration process, incorporating both affine and B-spline transformations. The SimpleITK framework, a toolkit for image processing, facilitates this process. Initially, the fixed (experimental tissue) and moving (digital reference tissue) images undergo padding, extended by a uniform margin of 50 pixels on all sides, to mitigate boundary effects during registration. Subsequently, these images are cast to a 32-bit floating-point format to ensure numerical precision in subsequent computations. Then, histogram matching gives the reference atlas the same intensity distribution as our sample. The function employs the SimpleITK framework to adjust the intensity distribution of a moving image to match that of a fixed image, a process known as histogram matching. This is achieved through the 'HistogramMatchingImageFilter' function, which modifies the pixel values of the moving image such that its histogram aligns with the histogram of the fixed image. The filter is configured to use 1024 histogram levels, ensuring a detailed representation of intensity distributions, and it matches the histograms based on 10 equally spaced intensity values (match points) that are representative of the entire intensity range. Additionally, by activating the 'ThresholdAtMeanIntensityOn' option, the algorithm modifies the intensity values of the moving image only above its mean

intensity, thereby preserving the lower intensity range. Afterwards I proceed with the first stage of registration.

The first stage involves an affine registration, where the moving image is aligned to the fixed image through an affine transformation. This is initialized using the 'CenteredTransformInitializer' function, which aligns the centers of the two images and estimates an initial transformation matrix. The affine registration employs the Mattes Mutual Information[24] metric, a statistical similarity measure, with 32 histogram bins to guide the optimization process. Optimization is performed using gradient descent, with a learning rate of 0.01, over 300 iterations, or until convergence is reached, as indicated by a minimum value change of 1×10^{-8} over a 20-iteration window. Physical shifts determine the optimizer scales to ensure uniform step sizes across dimensions. A multi-resolution strategy is employed by setting shrink factors and smoothing sigmas per level to [4, 2, 1] and [2, 1, 0], respectively, to enhance convergence. The transformation is interpolated using a linear method.
The affine transformation, denoted as T_{affine}, modifies the moving image to align with the fixed image based on translation, rotation, scaling, and shearing adjustments, formalized as:

$$T_{affine}(x) = Ax + b,$$

where A is a linear transformation matrix, and b is a translation vector applied to coordinates x of the moving image. When the affine transformation is found, I apply it to the fixed image and begin the second stage with that transformed image.

The second stage of registration is conducted using a B-spline transformation. This is initialized over a specified mesh size (4 x 4), determined by a uniform grid with dimensions proportional to the image size, to introduce local deformations for finer alignment. This stage also utilizes gradient descent optimization with the same parameters as the affine stage, with the exception of now using neighborhood correlation as its fit metric. It also runs over 300 iterations to refine the alignment based on local deformations within the control points defined by the mesh. The B-spline transformation, $T_{\text{B-spline}}$, allows for elastic deformation of the moving image to fit the fixed image more accurately, represented as:

$$T_{B-spline}(x) = \sum_{i}^{N} P_i B_i(x),$$

where P_i are the control points, $B_i(x)$ are the B-spline basis functions, and N is the number of control points. This method offers a flexible, high-degree-of-freedom approach to accommodate complex anatomical variations.

The final output is a composite transformation, combining the affine and B-spline transformations. This composite ensures that the initial global alignment provided by the affine transformation is refined through local adjustments via the B-spline transformation. The resulting composite transformation is applied to the moving image, achieving an aligned reference atlas section with respect to the fixed tissue

image. The same transform is also applied to the reference atlas annotations so the labels may be applied to the tissue section.

3 Results

I first dissected the general workflow into its constituent steps to automate the analysis workflow. There are four key steps in going from raw images to results for a given experiment: prediction, alignment, detection, and integration (Figure 3.1B-F). Each step outputs an analyzed component of the whole experiment that can provide a final result when combined with the others. Prediction determines the approximate location of a given section in the CCF. Alignment uses predictions to warp matching atlas borders onto the experimental tissue. Detection locates experimentally relevant signals in experimental tissue. Finally, integration combines the outputs of alignment and detection into usable experimental results (e.g., cell counts). An experiment can have any number of variations in its setup, so I begin our workflow assuming that the end-user has acquired serial sections of their tissue (Figure 3.1A) and imaged them with a background stain and some signal of interest that can be separated into two distinct channels. Importantly, these must be serial coronal sections as they are currently the only supported data format, although they need not be whole brain sections, as single hemispheres, single pieces of cortex, and midbrain are supported. However, I plan to implement more experimental modalities like sagittal sections in future versions (see Future Directions). I will now discuss how each step is conducted.

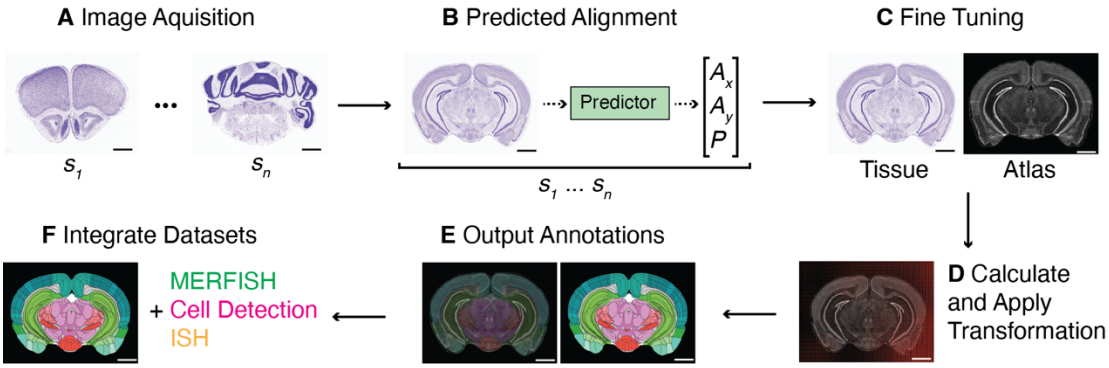

Figure 3.1 Bell Jar is a pipeline for aligning histology and integrating datasets.

(A) Images of the counterstained mouse coronal brain sections with Nissl are acquired in sequence anterior to posterior. **(B)** The predictor network processes each image to produce a preliminary X-cut angle, Y-cut angle, and anterior-to-posterior position (within the reference atlas). **(C)** Each prediction is evaluated for accuracy by a user and tuned for the best possible match. **(D)** An affine transformation followed by a B-spline transformation registering the atlas match onto the experimental tissue is found via gradient descent using SimpleITK[19]. **(E)** The transformation is applied to the annotations for the atlas match, resulting in the output annotation labeling the experimental tissue. **(F)** Annotations can be used to integrate a variety of datasets with the experimental tissue. Experimental tissue and atlas images were adapted from the Allen Mouse Brain Reference Atlas[1,20]. Scale bars are 1mm.

3.1 Predicting Slice Positions and Fine Tuning

Alignment of experimental tissue sections begins with predictor network processing. The Allen Brain CCF[1] is a 3D volume, and our experimental tissue must have a corresponding section for alignment somewhere within that space. Therefore, we must define a good guess for the atlas section that matches each experimental tissue. The predictor network selected images to estimate each section's X and Y cut angles and the anterior-posterior positions (or Z) (Figure 3.1B). The X and Y angles are the rotation angles in the XZ and YZ planes, respectively, and the anterior-posterior position Z is the depth in the coronal axis of the 3D volume. I include this step since

typical microtome or cryostat tissue collection will result in some variation in cut angle. These values approximate how the section was cut during harvesting. The predictor network is based on the ResNet-101 architecture[21]. The predictor was trained on a dataset of 1 million images generated from the Allen Mouse Brain Reference Atlas[1,20] 10μm Nissl staining volume by slicing at random angles and depths coronally (single or dual hemisphere at random, 50:50) to produce a generalizable network that gives strong initial predictions. It uses a 256x256 coronal section image as input and outputs three normalized scalars representing the X and Y cut angles and the anterior-posterior position. I repeat this process for all images in the experiment, then calculate the average predicted X and Y cut angles and apply them across all predicted sections. Since these initial predictions are not perfect, I present the predictions to the user and enable tweaking on a section-by-section basis (Figure 3.1C). I refer to this step as fine-tuning, and it is essential to incorporate human expert knowledge into the alignment process. The user examines each predicted section's cut angle and position, ensuring they represent the experimental tissue reasonably and accurately. Once all sections have been checked, the workflow proceeds to the alignment step.

3.2 Aligning Experimental Tissue

Registration of the experimental tissue to the Allen Brain CCF[1,20] (alignment) is non-trivial. Experimental tissue can have widely ranging imaging settings and collection techniques, leading to significant inter and intra-experimental differences in

brightness, contrast, resolution, evenness of illumination, artifacts, and tissue quality. Previous methods avoid these variations using standardized imaging techniques like two-photon microscopy[16,17] or deep learning approaches[5,13] to make a generalized system insensitive to such factors. However, I used an image registration approach to achieve superior alignments at a local scale (cellular resolution) with standard 2D imaging. The moving image is the digital average tissue from the Allen Brain CCF, and the fixed image is our experimental tissue of interest. By finding this transform, I simultaneously warp the digital average CCF tissue onto our experimental tissue and find a valid transform for transforming the CCF labels into the space of our experimental tissue since any transformation of the digital average tissue is also applicable to its labels. Transformations are found using a two-step composite registration process using the Python library for SimpleITK[19]. It relies on the inherent similarity between the gradient of the digital average Nissl image and the real tissue. Even though a user may not have background data that is Nissl stained, the digital average still represents a strong feature set to which I can compare against the experimental tissue. First, images are preprocessed by being resized to 360x360 pixels and converted from an intensity image into a gradient image via the Sobel operator[22,23]. The gradient image is then normalized to ensure gradients between the experimental tissue and the digital average image are on the same scale. An initial affine transformation is found via gradient descent using Mattes mutual information[19,24] as the fit metric. After the affine transformation is found, the moving image is resampled by the affine transform, which centers it relative to the fixed

image. A B-spline transformation is then found via gradient descent with the same metric, but this time, the resampled moving image is used to superimpose the digital average tissue on the fixed image (experimental tissue) accurately by matching gradient intensities (Figure 3.1D). The same transformation of the digital average tissue is applied to its labels to create annotations for the experimental tissue (Figure 3.1E). As a pixel-to-region mapping, such annotations can be applied to a variety of datasets, including fluorescently labeled signals to detect cells or molecules such as *in situ* hybridization (ISH) and multiplexed error-robust fluorescence *in situ* hybridization (MERFISH) or any dataset in the tissue section's coordinate space (Figure 3.1F).

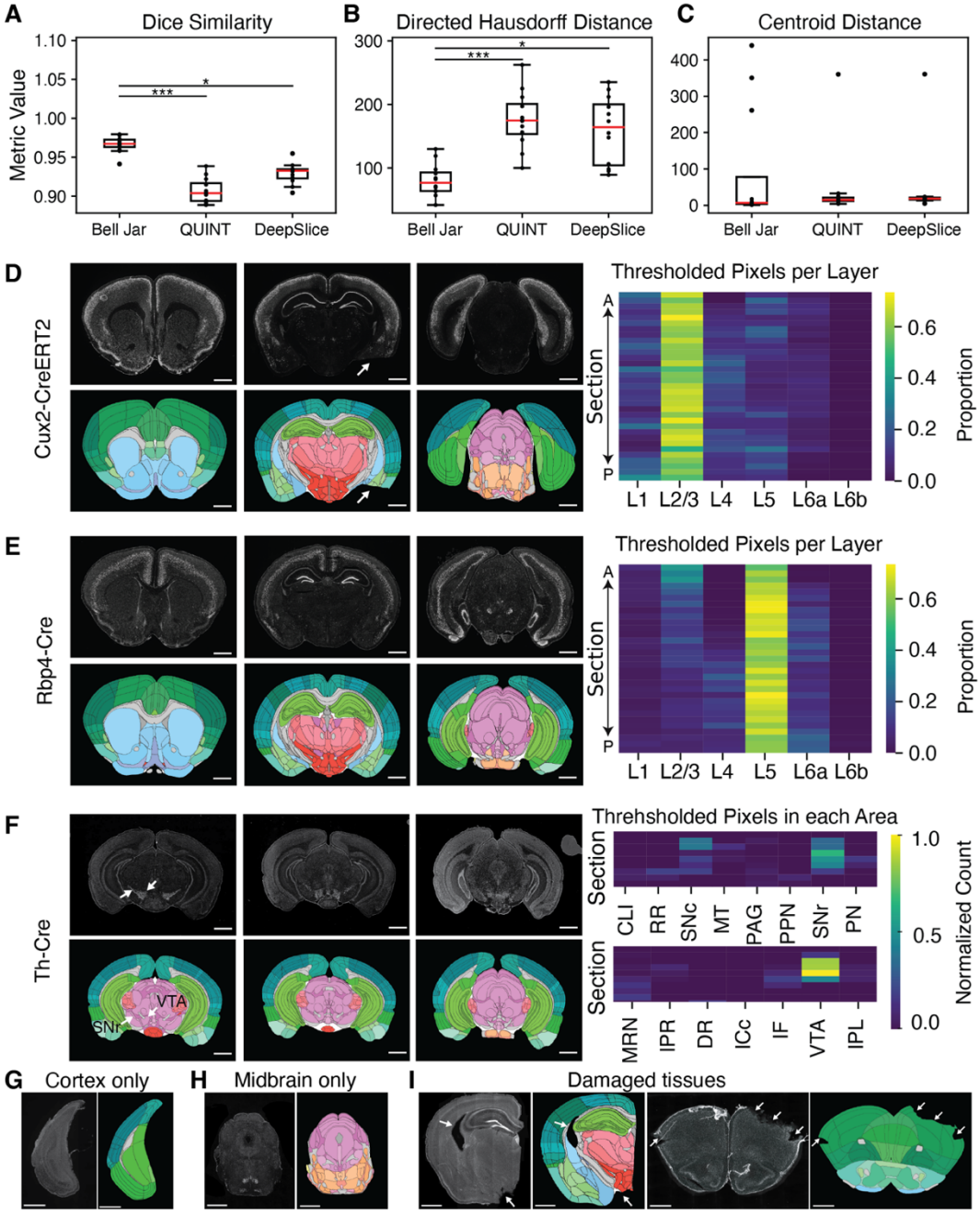

Figure 3.2 Bell Jar accurately assigns anatomical labels to experimental tissues.

(A) The Dice similarity coefficient calculated between experimental tissue sections and their alignments produced from each method. Bell Jar significantly outperformed QUINT[3] and DeepSlice[5] (p = 8.49e-07, 0.00297 *Kruskal-Wallis with post hoc Dunn*, n=12). **(B)** Directed Hausdorff distance between the contours of alignments and the aligned experimental tissue sections from each method. Bell Jar outperformed

QUINT and DeepSlice (p = 0.00012, 0.0011 *Kruskal-Wallis with post hoc Dunn*, n=12). **(C)** The distance between the centroids of experimental tissue sections and the alignments from each method. **(D)** (top) A selection of Cre ISH images from a Cux2-CreERT2 mouse provided by Allen Brain transgenic mouse characterization dataset[25]. Allen Mouse Brain Reference Atlas, connectivity.brain-map.org/transgenic/experiment/571261835 (bottom) examples of per-section alignment alignments by Bell Jar. A heatmap shows the proportion of thresholded pixels (detected Cux2-Cre ISH signals) in each layer by section. The arrows in the middle panels point the damaged section images. **(E)** (top) A selection of Cre ISH images from an alignment of an Rbp4-Cre mouse provided by Allen Brain transgenic mouse characterization dataset[25]. Allen Mouse Brain Reference Atlas, connectivity.brain-map.org/transgenic/experiment/117285137 (bottom) examples of per-section alignment alignments by Bell Jar. A heatmap shows the proportion of thresholded pixels (detected Rbp4-Cre ISH signals) in each layer by section. **(F)** (top) A selection of Cre ISH images from a Th-Cre mouse provided by Allen Brain transgenic mouse characterization dataset[25]. Allen Mouse Brain Reference Atlas, connectivity.brain-map.org/transgenic/experiment/304164559 (bottom) examples of per-section alignment alignments by Bell Jar. Arrows show two key regions detected in our analysis VTA (ventral tegmental area) and SNr (substantia nigra, reticular part). A heatmap shows the normalized count of thresholded pixels (detected Th-Cre ISH signals) I detected in midbrain regions consistent with the expected expression. **(G)** An example of a single piece of cortex without midbrain or hindbrain aligned with Bell Jar. **(H)** An example of a single piece of midbrain without cortex aligned with Bell Jar. **(I)** Examples of damaged experimental tissue aligned by Bell Jar using masking. Arrows in D and I show damaged regions that were compensated for by masking. Scale bars are 1mm. * = $p < 0.05$; *** = $p < 0.001$; *Kruskal-Wallis with post hoc Dunn.*

I compared the efficacy of Bell Jar's alignment against two contemporary tools: QUINT[3] and DeepSlice[5]. To measure accuracy, I used the Dice Similarity Coefficient (DSC), which quantifies the overlap between the aligned sections and their corresponding atlas annotations. I found that Bell Jar's performance in fitting anatomical labels to experimental tissue sections (Figure 3.2A-C) is superior, being consistently closer to the ground truth than that of QUINT or DeepSlice. Further comparison using the Directed Hausdorff distance, which assesses the maximum

distance of a set to the nearest point in the other set, reinforced these findings. Bell Jar achieved a significantly lower mean distance than QUINT and DeepSlice, indicative of more precise contour alignment (Figure 3.2B). To further assess performance, centroid distance measurements between the experimental tissue sections and the respective alignments corroborated the superior performance of Bell Jar, as it yielded the smallest deviations (Figure 3.2C).

To test Bell Jar's alignment performance, I utilized ISH signals of Cre recombinase mRNAs in Cux2-CreERT2 mice expressing Cre in the cerebral cortical layers 2 and 3 mostly[25]. I investigated whether Bell Jar can accurately detect Cre expression at the expected localization across different sections (Figure 3.2D). I first aligned all the sections using Bell Jar, and then used the resulting annotations to find the proportion of Cre ISH signals in the cortical layers of each section. A positive ISH signal was defined as pixels above an intensity threshold. As expected, the signal is predominantly in layers 2 and 3 (Figure 3.2D). I also used different Cre ISH expression images of two transgenic mice, Rbp4-Cre and Th-Cre. Rbp4-Cre mice express Cre specifically in the cortical layer 5, and Th-Cre mice in midbrain structures such as the ventral tegmental area (VTA) and substantia nigra (SNr)[25] (Figure 3.2E-F). Bell Jar successfully aligns the experiment brain tissue images from these two transgenic lines based on the precise demarcation of Cre+ cortical layers and midbrain subregions (Figure 3.2E-F).

I also investigated whether Bell Jar could handle challenging tissue types that can be produced when the posterior portions of the brain are coronally sectioned. As shown in Figure 3.2G-H, Bell Jar can align a single piece of cortex or midbrain section without midbrain or other forebrain tissues, respectively. It also showed versatility in aligning damaged tissues by employing masking techniques, removing portions of the atlas image to better match the experimental one, and ignoring areas of experimental artifact or damage (Figure 3.2I). These tests underscore the software's adaptability to various experimental conditions and its proficiency in maintaining alignment accuracy even in non-ideal samples.

3.3 Detecting Signals of Interest

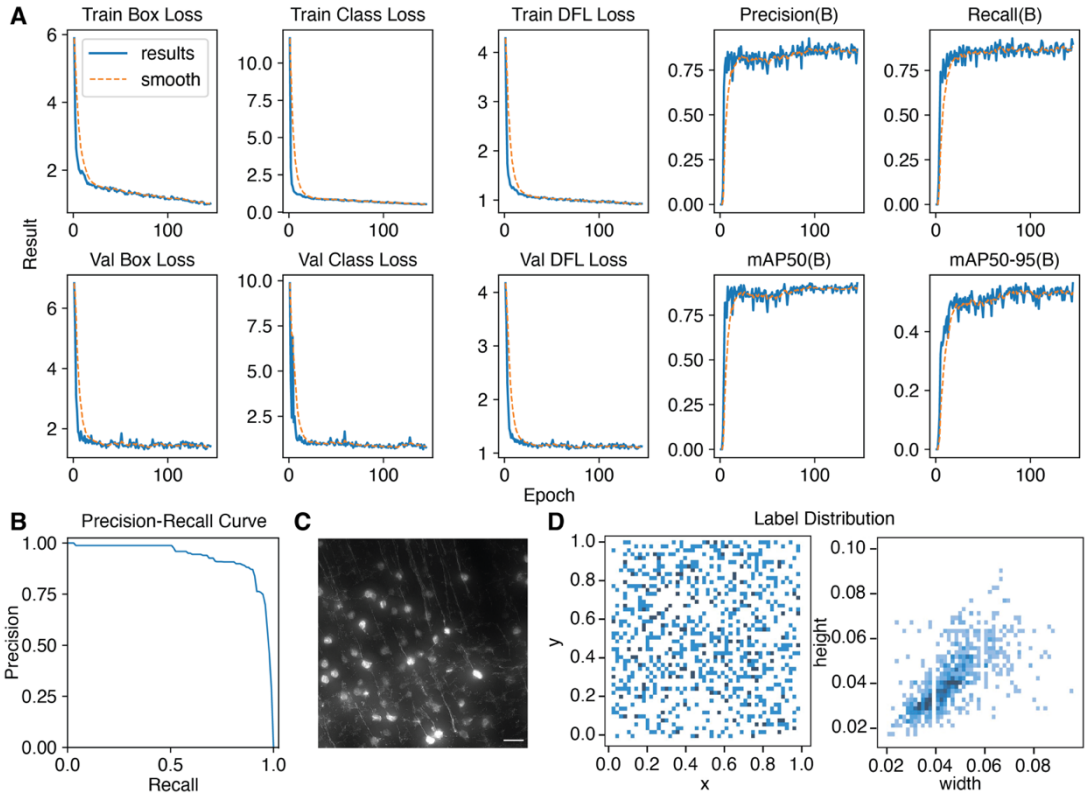

Figure 3.3 Bell Jar uses object detection to detect cells automatically.

(A) Training results for the YoloV8[26] object detector Bell Jar uses for cell detection. Each graph represents a measurement of model training progress. The final model demonstrates minimal loss and high precision. **(B)** The precision-recall curve for the cell detection model. **(C)** A representative microscopic image of the training data was used to create the cell detection model, showing RV*dG*-mCherry labeled neurons in the mouse visual cortex. The scale bar is 50µm. **(D)** The distribution of labels in space and by size in the training data set (left) normalized x and y coordinate heatmap of the cell labels (right) total image width and height normalized heatmap of cell label size.

Bell Jar provides integrated cell detection by object detection to identify cells in the experimental tissue. I leverage the YoloV8[26] object detection model to autonomously detect cell bodies, demonstrating a notable advantage in handling varying signal levels, a challenge that traditional thresholding methods often fail to address (Figure

3.3A). The precision-recall curve for the cell detection model indicates high reliability, showcasing the model's ability to effectively discern true positives from false positives (Figure 3.3B). The model was trained on a representative dataset of RV*dG*[20] (G-deleted rabies virus) traced neuronal cell bodies (Figure 3.3C). I curated a dataset of 41 images in which we labeled 700 neurons, ensuring broad coverage of cell body appearances and signal intensities (Figure 3.3C). The resulting label distribution encompasses a wide variation in relative cell position and size, helping generalize the model (Figure 3.3D). From our initial 700 cells, I further augmented the dataset with random brightness and saturation levels to simulate a variety of imaging parameters (the final training had 2600 labeled instances). In addition to these advantages, Bell Jar also uses slicing-aided hyperinferencing[27] (SAHI) to enhance its detections. It tiles high-resolution input images into small overlapping tiles (640x640, 50% overlap) and performs detection on each of them, then aggregates results by non-maximum suppression of overlapping detections. Using SAHI ensures that cells are detected at a contextually relevant scale since each cell body may be as little as 200px in a high-resolution microscopy image.

Bell Jar's cell counting performance was benchmarked against manual counts and standard automated methods, such as ImageJ[28] and ilastik[29]. I selected tissue sections from experiments previously imaged in our lab with rabies virus-traced neurons in the mouse cerebral cortex labeled with mCherry. All the sections were counted independently with each method, and the counts were compared to each other (Figure

3.4A). The results demonstrated that Bell Jar's deep learning object detection-based cell counting closely aligned with manual counts across ten representative sections, evidenced by significant correlation (ImageJ vs Manual: r=0.9650, p=6.31e-06*; ilastik vs Manual: r=0.8893, p=5.74e-04*; Bell Jar vs. Manual: r=0.9827, p=3.80e-07*; Figure 3.4B). As performed by Bell Jar, the neuron counts per section closely matched those of manual counting, outperforming the counts achieved by ImageJ and ilastik, which rely on intensity thresholding and machine learning approaches, respectively (Figure 3.4A and 4B). In contrast, counts obtained from ImageJ and ilastik, while still correlated with manual counts, showed lower correlation coefficients, suggesting less accuracy with the manual standard (Figures 4C and 4D), mainly because of ImageJ and ilastik mislabeled neurons (yellow arrowheads in Figure 3.4G and 4H). Direct comparison with these conventional methods suggests the robustness of Bell Jar's automated cell counting method.

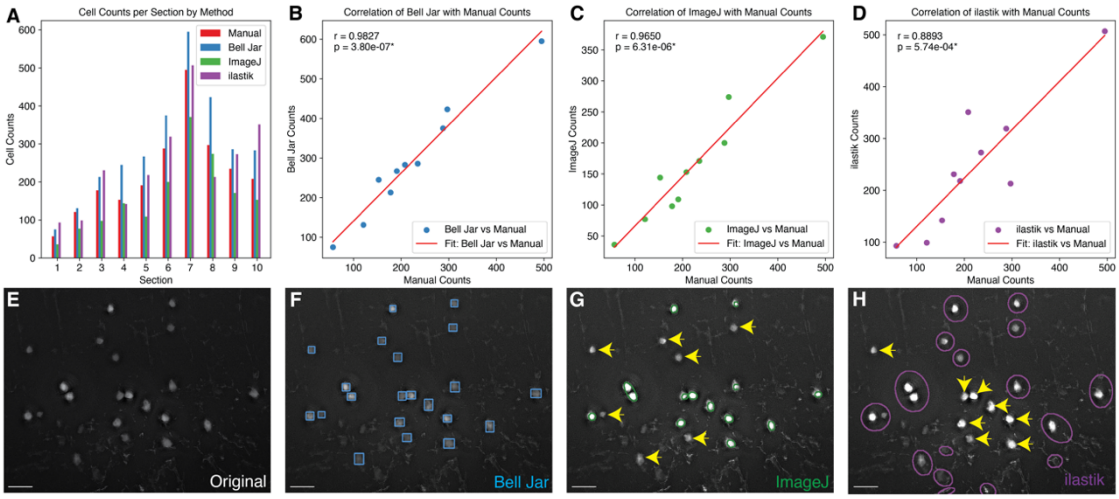

Figure 3.4. Bell Jar cell counting is comparable to manual counting

(A) Counts of neurons detected in ten representative coronal sections of a mouse with cortical cells traced by RV*dG* by each tested method. **(B)** Correlation of Bell Jar automatic counts and manual counting across ten representative sections (r=0.9827, p=3.80e-07*, *Pearson's Correlation*) **(C)** Correlation of ImageJ[28] automatic counts by thresholding and manual counting across ten representative sections (r=0.9650, p=6.31e-06*, *Pearson's Correlation*) **(D)** Correlation of ilastik[29] automatic counts and manual counting across ten representative sections (r=0.8893, p=5.74e-04*, *Pearson's Correlation*) **(E)** The original sample image of labeled mouse cortical neurons. **(F)** Bell Jar's detections of neurons are represented on the sample image as blue bounding boxes. **(G)** ImageJ's detections of neurons are represented in green circles on the sample image. **(H)** Ilastik's detections of neurons are represented in purple circles on the sample image. Yellow arrows indicate missing detections of cells in ImageJ and ilastik (G-H). The scale bars are 50μm.

4 Conclusions and Discussions

4.1 Bell Jar Advances Histological Analysis through Automation

The integration of Bell Jar into histological analysis represents a significant advancement in the automation of tissue alignment by using automated non-linear deformations for image registration. The alignment module's ability to accurately map experimental tissue to a reference atlas not only streamlines the initial stages of analysis but also ensures reproducibility. This yields a particularly notable advantage when considering the variance in signal levels and tissue integrity commonly encountered in experimental settings. Bell Jar's alignment outcomes, validated against other current state-of-the-art methods such as DeepSlice and QUINT[3,5], demonstrate its robustness and reliability. Together, these improvements enhance confidence in subsequent analytical processes.

4.2 Bell Jar Enhances Precision in Cell Detection

Applying the YoloV8[26] object detection algorithm within Bell Jar for cell counting is a methodological shift from traditional thresholding-based techniques to a more sophisticated, deep learning-based approach. This shift is substantiated by the high correlation of Bell Jar's cell counts with manual cell counts. These findings suggest that the object detection method employed by Bell Jar is sensitive enough to replicate the discernment of trained human eyes. Furthermore, the precision-recall balance achieved by the model underscores its efficiency in distinguishing true cell bodies from noise, a task that has been challenging for automated systems.

4.3 Implications for Neuroscience and Beyond

The implications of Bell Jar can extend into various fields of other biological research, where histological analysis is a cornerstone in other tissue systems or model organisms. The high-throughput and automated nature of Bell Jar's pipeline enables researchers to process large datasets with greater ease, facilitating more extensive studies that the laborious nature of manual analysis may have previously limited. In neuroscience, this could accelerate the precise mapping of neural circuits and quantifying neuron populations across different brain regions.

4.4 Future Directions

The Bell Jar platform offers a foundation for further methodological enhancements and broader applications. The adaptability of the software to various tissue types and

staining methods suggests its potential utility in a broader range of biological studies. The underlying algorithms could also be refined to include more advanced artificial intelligence techniques, improving accuracy and expanding the scope of detectable features within tissue sections. One of the most likely future uses for Bell Jar is its potential integration into spatial transcriptomic studies, where histological data can be combined with genomic data. This comprehensive approach could provide greater accuracy in the anatomical localization of transcriptomic data.

In conclusion, Bell Jar represents a step forward in automated histological analysis. Its ability to provide accurate, reproducible results lays a convenient platform for neuroscience research characterized by high-throughput image data analysis. Likewise, Bell Jar's open-source nature lends itself to cross-disciplinary integration. Bell Jar provides easy access for neuroscientists to analyze the mouse brain anatomy accurately and flexibly.

4.5 Limitations of the Current Work

Despite the strengths of the Bell Jar system, certain limitations must be acknowledged. Bell Jar requires images in both the counterstain, such as Nissl or DAPI (4',6-diamidino-2-phenylindole), and signal channels simultaneously, an imaging paradigm that may not align with the objectives or resources of every study. Researchers who only image specific regions of interest, such as a subregion of the hippocampal formation, may find this requirement burdensome, as it necessitates

comprehensive imaging beyond their targeted area. Moreover, obtaining images of the sample with consistent aspect ratios in both channels is critical for ensuring accurate alignment. This can pose a practical challenge, mainly when dealing with samples that are difficult to image in their entirety or when equipment limitations impact image consistency.

Additionally, while Bell Jar substantially reduces the need for expert input compared to traditional manual methods, it does not eliminate it. The initial setup and some aspects of the alignment process still require user intervention and a degree of expertise, while this user intervention provides the opportunity to correct errors and improve the qualities of the image alignment. This necessity for expert input, although reduced, could introduce user bias and inconsistency, especially in cases where multiple individuals are involved in the image processing workflow.

Lastly, while object-based cell detection offers a clear advantage over thresholding-based methods but can be prone to false positives, currently, Bell Jar can detect ~80% of cells reliably at a confidence threshold of 0.5. Lowering this threshold may detect more cells but at the risk of false positive detections. A key challenge is balancing confidence cutoffs for predictions to limit these false positives and having a sensible threshold to label most cells in an image.

BIBLIOGRAPHY

1. Wang, Q., Ding, S.-L., Li, Y., Royall, J., Feng, D., Lesnar, P., Graddis, N., Naeemi, M., Facer, B., Ho, A., et al. (2020). The Allen Mouse Brain Common Coordinate Framework: A 3D Reference Atlas. Cell *181*, 936-953.e20. https://doi.org/10.1016/j.cell.2020.04.007.

2. Paxinos, G., and Franklin, K.B.J. (2019). Paxinos and Franklin's The mouse brain in stereotaxic coordinates Fifth edition. (Elsevier, Academic Press).

3. Yates, S.C., Groeneboom, N.E., Coello, C., Lichtenthaler, S.F., Kuhn, P.-H., Demuth, H.-U., Hartlage-Rübsamen, M., Roßner, S., Leergaard, T., Kreshuk, A., et al. (2019). QUINT: Workflow for Quantification and Spatial Analysis of Features in Histological Images From Rodent Brain. Front. Neuroinform. *13*, 75. https://doi.org/10.3389/fninf.2019.00075.

4. Xiong, J., Ren, J., Luo, L., and Horowitz, M. (2018). Mapping Histological Slice Sequences to the Allen Mouse Brain Atlas Without 3D Reconstruction. Front. Neuroinform. *12*, 93. https://doi.org/10.3389/fninf.2018.00093.

5. Carey, H., Pegios, M., Martin, L., Saleeba, C., Turner, A.J., Everett, N.A., Bjerke, I.E., Puchades, M.A., Bjaalie, J.G., and McMullan, S. (2023). DeepSlice: rapid fully automatic registration of mouse brain imaging to a volumetric atlas. Nat Commun *14*, 5884. https://doi.org/10.1038/s41467-023-41645-4.

6. Lauridsen, K., Ly, A., Prévost, E.D., McNulty, C., McGovern, D.J., Tay, J.W., Dragavon, J., and Root, D.H. (2022). A Semi-Automated Workflow for Brain Slice Histology Alignment, Registration, and Cell Quantification (SHARCQ). eNeuro *9*, ENEURO.0483-21.2022. https://doi.org/10.1523/ENEURO.0483-21.2022.

7. Iqbal, A., Sheikh, A., and Karayannis, T. (2019). DeNeRD: high-throughput detection of neurons for brain-wide analysis with deep learning. Sci Rep *9*, 13828. https://doi.org/10.1038/s41598-019-50137-9.

8. Landhuis, E. (2017). Neuroscience: Big brain, big data. Nature *541*, 559–561. https://doi.org/10.1038/541559a.

9. Oh, S.W., Harris, J.A., Ng, L., Winslow, B., Cain, N., Mihalas, S., Wang, Q., Lau, C., Kuan, L., Henry, A.M., et al. (2014). A mesoscale connectome of the mouse brain. Nature *508*, 207–214. https://doi.org/10.1038/nature13186.

10. Harris, J.A., Mihalas, S., Hirokawa, K.E., Whitesell, J.D., Choi, H., Bernard, A., Bohn, P., Caldejon, S., Casal, L., Cho, A., et al. (2019). Hierarchical organization of cortical and thalamic connectivity. Nature *575*, 195–202. https://doi.org/10.1038/s41586-019-1716-z.

11. Zingg, B., Hintiryan, H., Gou, L., Song, M.Y., Bay, M., Bienkowski, M.S., Foster, N.N., Yamashita, S., Bowman, I., Toga, A.W., et al. (2014). Neural Networks of the Mouse Neocortex. Cell *156*, 1096–1111. https://doi.org/10.1016/j.cell.2014.02.023.

12. Kim, Y., Yang, G.R., Pradhan, K., Venkataraju, K.U., Bota, M., García Del Molino, L.C., Fitzgerald, G., Ram, K., He, M., Levine, J.M., et al. (2017). Brain-wide Maps Reveal Stereotyped Cell-Type-Based Cortical Architecture and Subcortical Sexual Dimorphism. Cell *171*, 456-469.e22. https://doi.org/10.1016/j.cell.2017.09.020.

13. Sadeghi, M., Ramos-Prats, A., Neto, P., Castaldi, F., Crowley, D., Matulewicz, P., Paradiso, E., Freysinger, W., Ferraguti, F., and Goebel, G. (2023). Localization and Registration of 2D Histological Mouse Brain Images in 3D Atlas Space. Neuroinform *21*, 615–630. https://doi.org/10.1007/s12021-023-09632-8.

14. Franklin, K.B.J., and Paxinos, G. (2008). The mouse brain in stereotaxic coordinates Compact 3. ed. (Elsevier Academic Press).

15. Chiaruttini, N. (2019). Registration of mouse brain slices with ABBA. https://github.com/BIOP/ijp-imagetoatlas.

16. Niedworok, C.J., Brown, A.P.Y., Jorge Cardoso, M., Osten, P., Ourselin, S., Modat, M., and Margrie, T.W. (2016). aMAP is a validated pipeline for registration and segmentation of high-resolution mouse brain data. Nat Commun *7*, 11879. https://doi.org/10.1038/ncomms11879.

17. Tyson, A.L., Vélez-Fort, M., Rousseau, C.V., Cossell, L., Tsitoura, C., Lenzi, S.C., Obenhaus, H.A., Claudi, F., Branco, T., and Margrie, T.W. (2022). Accurate determination of marker location within whole-brain microscopy images. Sci Rep *12*, 867. https://doi.org/10.1038/s41598-021-04676-9.

18. Electron JS (2022). https://www.electronjs.org.

19. Lowekamp, B.C., Chen, D.T., Ibáñez, L., and Blezek, D. (2013). The Design of SimpleITK. Front. Neuroinform. *7*. https://doi.org/10.3389/fninf.2013.00045.

20. Allen Institute for Brain Science Allen Reference Atlas - Mouse Brain. atlas.brain-map.org.

21. He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (IEEE), pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

22. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2.

23. Van Der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., and Yu, T. (2014). scikit-image: image processing in Python. PeerJ *2*, e453. https://doi.org/10.7717/peerj.453.

24. Mattes, D., Haynor, D.R., Vesselle, H., Lewellyn, T.K., and Eubank, W. (2001). Nonrigid multimodality image registration. In, M. Sonka and K. M. Hanson, eds., pp. 1609–1620. https://doi.org/10.1117/12.431046.

25. Lein, E.S., Hawrylycz, M.J., Ao, N., Ayres, M., Bensinger, A., Bernard, A., Boe, A.F., Boguski, M.S., Brockway, K.S., Byrnes, E.J., et al. (2007). Genome-wide atlas of gene expression in the adult mouse brain. Nature *445*, 168–176. https://doi.org/10.1038/nature05453.

26. Jocher, G., Chaurasia, A., and Qiu, J. (2023). YOLO by Ultralytics. Version 8.0.0.

27. Akyon, F.C., Cengiz, C., Altinuc, S.O., Cavusoglu, D., Sahin, K., and Eryuksel, O. (2021). SAHI: A lightweight vision library for performing large scale object detection and instance segmentation. (Zenodo). https://doi.org/10.5281/zenodo.5718950 https://doi.org/10.5281/zenodo.5718950.

28. Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al. (2012). Fiji: an open-source platform for biological-image analysis. Nat Methods *9*, 676–682. https://doi.org/10.1038/nmeth.2019.

29. Berg, S., Kutra, D., Kroeger, T., Straehle, C.N., Kausler, B.X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., et al. (2019). ilastik: interactive machine learning for (bio)image analysis. Nat Methods *16*, 1226–1232. https://doi.org/10.1038/s41592-019-0582-9.