

Bell Jar Guide (version v6.0.0)

0. Installation

<https://github.com/asoronow/belljar/releases>

Download the zip file for your operating system, unzip it, and run the belljar file.

For intel-based mac, download darwin-x64.

For apple-silicon based mac, download Darwin-arm64.

For windows download, win32-x64.

For linux, download the deb for your architecture.

(Installation of Bell Jar will take approximately 60 minutes assuming at least 10 MB/s download speeds and requires ~60Gb of storage).

Successful Installation of Bell Jar will show three main options (Preprocessing, Prediction, Output) once Bell Jar is opened.

Go to section 4 for a quick example run through on a sample dataset!

1. Preprocessing

The preprocessing tools are helpful runtimes to prepare images for the detection and alignment steps in 'Prediction'.

a. Max Projection

It takes a directory of z-stacked single-channel images and flattens it into a single plane by taking only the pixels of the greatest intensity. This function offers a fast way of "focusing" images with focal depth.

Usage:

1. Create a new directory for the output of the process. (e.g., Brain42_maxprojections)
2. Select the input path to your z-stacked images. They must be in the tiff (ome-tiff works as well) file format.
3. Select the output path as the directory you created.
4. Click 'Run' to initiate the progress and monitor results by the loading bar on the bottom of main window or the output folder contents.

Outputs: Max projected tiff files in your output path directory.

b. Top Hat Filter

It takes a directory of flat images and applies a top hat filter of the selected kernel size to each image. This process helps remove haloing and even illumination artifacts that occurred during EFI (Extended Focus Imaging) processing and max projection.

Usage:

1. Create a new directory for the output of the process. (e.g., Brain42_tophatfiltered)
2. Select the input path to your flattened images. They must be in the tiff (ome-tiff works as well) file format.
3. Select the output path as the directory you created.
4. Input a filter size that will define the kernel size of the filter. A good starting point is 10px; higher values may begin to create undesirable artifacts but may be necessary on very bright images.
5. Set a gamma correction value if desired. By default, this will be calculated automatically for each image.
6. Click 'Run' to initiate the program and monitor results by the loading bar on the bottom of the main window or the output folder contents.

Outputs: Top hat filtered tiff files in your output path directory.

2. Prediction

The prediction tools comprise the main components of Bell Jar. They can align tissue section images to the Allen brain CCFv3 (<http://atlas.brain-map.org/>), detect labeled neurons in tissue sections, and integrate these two sets of data into detailed counts of cells by brain region in an experiment.

a. Cell Detection

'Cell Detection' tool finds cells (cell bodies) in flattened (single plane) tissue sections. The detector was trained on the fluorescently labeled cell bodies traced by G-deleted engineered rabies tracer (see Method sections of Soronow et al., TBA).

Usage:

1. Create a new directory for the outputs of the process (e.g., Brain42_detections).
2. Prepare your input images in anterior to posterior order along the coronal sections with ascending section indicator names (e.g., Brain42_section1,

Brain42_section2); the ordering and naming ensure that the predictions are processed in the same order as the other analysis steps.

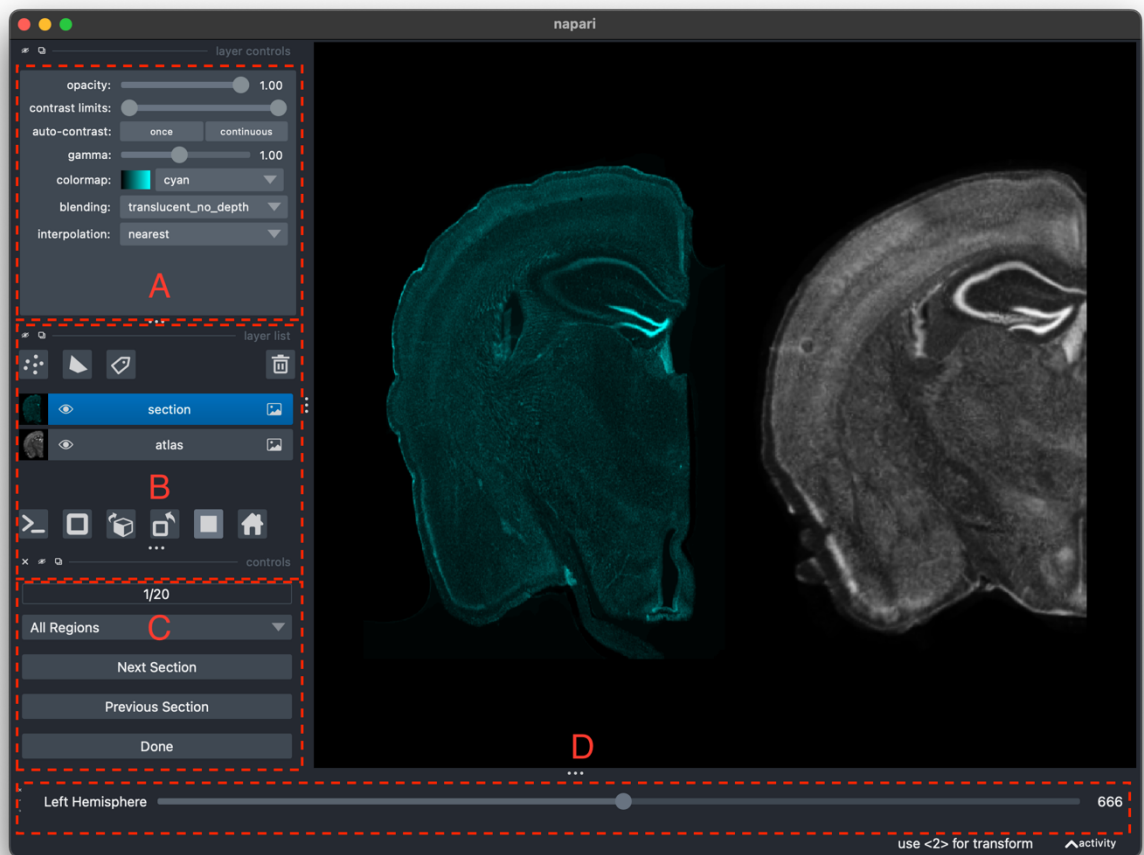
3. Select the input path to your preprocessed images with cells to detect.
4. Select the output path to the new directory you created.
5. (Optional) Under advanced settings, you can specify a custom YOLOv5 model. These advanced settings allow you to adjust the tile size and threshold.
6. Click 'Run' to initiate the program and monitor the results by the loading bar or the output folder contents.

Outputs: Detection points maps (for use with traditional segmentation analysis tools), bounding box images (to verify detections), and raw predictions (saved as .pkl files, used in subsequent analysis steps) in your output path directory.

b. Align

'Align' tool takes in a directory of DAPI or Nissl-stained section images and assists the user in aligning them to the Allen Brain atlas CCFv3. This function preserves the aspect ratio between the background-stained images and the Allen Brain atlas reference images and your cell images to ensure accurate counts.

Necessary: The ordered inputs here should correspond directly to the prediction images (e.g., Brain42_section1 should be the name of both the DAPI/Nissl section (background staining) and the image with cells present (the experimental image). They must be **.PNG** format 8 bit and no larger than 2MB each.



Align Tissue

Alignment takes your experimental tissue sections and helps align them to the Allen Brain Atlas. A predictor network will evaluate your tissue and suggest the best fitting sections. Afterwards you will have to fine tune these predictions to best match your tissue.

Input Path ⓘ

E

Click to select directory or cancel to enter manually

Output Path ⓘ

F

Click to select directory or cancel to enter manually

Alignment Method ⓘ

G

Both Hemispheres

Angle Override ⓘ

H

Enter

I

Back

Run

Usage:

1. Create a directory for the outputs of the process. (e.g., Brain42_alignment)
2. Prepare your section images in anterior to posterior order with ascending section indicator names (e.g., Brain42_section1, Brain42_section2); the ordering and naming ensure that the alignments are processed in the same order as the other analysis steps. (section images should be 8-bit pngs)
3. Select the input directory to your section image (E)
4. Select the output path to the new directory you created (F). (e.g., Brain42_alignment)
5. Select whether the input images are brain sections containing 'both hemispheres' or a 'single hemisphere' (G) (if you select 'single hemisphere', your section images must be in a normal coronal left hemisphere orientation).
6. (Optional) If you had poor results with the automated predictions, possibly due to the off angle of your section images from the atlas reference, you may manually select the cut angle (up to 10 degree in positive or negative direction) of the brain with the 'Angle Override' option (H).
7. Click 'Run' to initiate the program.
8. During execution, the script will first predict the cut angle of your samples. Afterward, it will select approximate atlas matches to your section images, and then a new window will pop up, prompting you to fine-tune (if you do not see this window, it may be hidden or minimized. Check the taskbar/dock).
9. (Optional) Set the 'atlas' image in the bottom left layer selector (B) to a different color and adjust the opacity (top left under 'layer controls', A) until you have reached a visibility you are comfortable with and can easily be compared with your section images. You may also use the eye icon to hide each section image while doing your comparison.
10. You must tune each atlas image to best match your section image by using the horizontal scroll bar at the bottom of the window (D) to move anterior to posterior through the reference atlas.
11. (Optional) If the cerebral cortex in any section is separated from the midbrain, you may check the 'Separated?' box above the 'Done' button to apply a filter that will help better fit them to the atlas (C).
12. Once satisfied with your selected match, click 'Next' to move on to the next section.
13. Repeat this process until you arrive at your final section image, and then click 'Done' to write the results (C).
14. Monitor the output folder to check the progress of the resulting output. The warping will take 10-30 minutes per section depending on how fast your computer is (the average PC will take 30 minutes per section).

Outputs: Reference atlas slices, Area maps for each section, and Raw annotation data (as .pkl files for subsequent analysis) in your output path directory (e.g., Brain42_alignment).

c. Adjustments

‘Adjustments’ tool takes in a directory containing the annotations and DAPI/Nissl section images from alignment and allows users to adjust the alignments. Two modes are available. The first is “Affine,” which enables transformations of the annotation to better fit the tissue. The second is “Paint,” which lets the user directly paint new region boundaries onto the annotation.

Important: This routine directly modifies the annotations. Please follow the setup instructions. Additionally, a mouse (not a trackpad) is **STONGLY** advised.

Adjustments

Allows you to adjust your alignments with a simple brush tool. Add annotation pkls and DAPI files to a folder and select it as your input. The annotation and DAPI files should have the same names as they did in the original alignment step. Right click selects the hovered region, left click paints the selected region over any other region. Z clears all changes. Q advances to the next section. ESC quits the process. WARNING: Annotation files are modified directly.

Input Path ? **J**

Click to select directory or cancel to enter manually

Affine or Paint ?

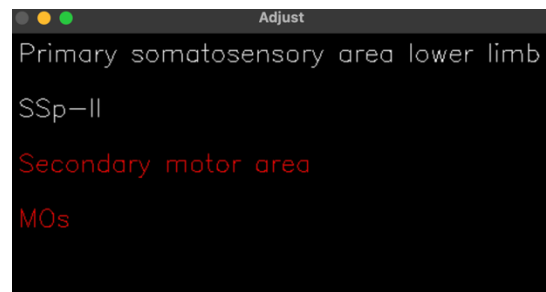
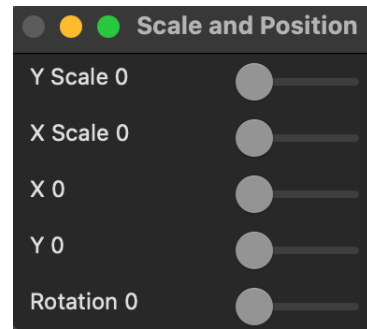
Affine ▾ **K**

BackRun

Usage:

1. Create a new directory clearly labeled as adjustments (e.g., Brain42_adjusted).
2. Make a copy of the annotations (.pkl files from the alignment step) in the new directory (preserve naming).
3. Make a copy of the DAPI/Nissl section images in the new directory (preserve naming).
4. Select the input path to the directory you have now configured (J).
5. Select whether you want to adjust in “Affine” or “Paint” mode (K).

6. Run the program.
7. (Affine Mode) When you adjust the alignment using the provided sliders in affine mode, the Q key will save and advance to the next section, and ESC will end the process. Y and X scale stretch your alignment in their respective directions. X and Y move your alignment in their respective directions. Rotation lets you rotate your alignment.
8. (Paint Mode) When in paint mode, you adjust the alignment by painting. A separate window will appear like the screenshot. The white text represents the region your cursor is currently over. The red text is the region you have selected to paint. To select a region to paint, hover over the map until you locate the desired region to modify (will appear in white text on separate window), and right-click (in Mac, two fingers on the trackpad) to select it (should show up in red text below the white). Now while keep pressing your left click, you can move your cursor to paint new boundaries for the selected region anywhere on the map. The Q key will save and advance to the next section saving any changes permanently, Z will undo any changes, and ESC will end the process.
9. Navigate and edit all the image sections as needed, and the process will terminate at the end of all images or upon pressing ESC.



Outputs: Modified annotation .pkl files in the selected input directory.

3. Output

The output tools produce useful output from the Bell Jar prediction routines. Currently, it consists of one primary tool 'Count Brain' we have developed that counts all the cells in an experiment processed with the prediction tools. 'Intensity by Region' and 'Collate Counts' will be experimental features.

a. Count brain

Takes in the raw cell location predictions and raw annotations to produce a count of predicted labeled cells per region in the experimental brain. Counts are output by section and as totals in a CSV file.

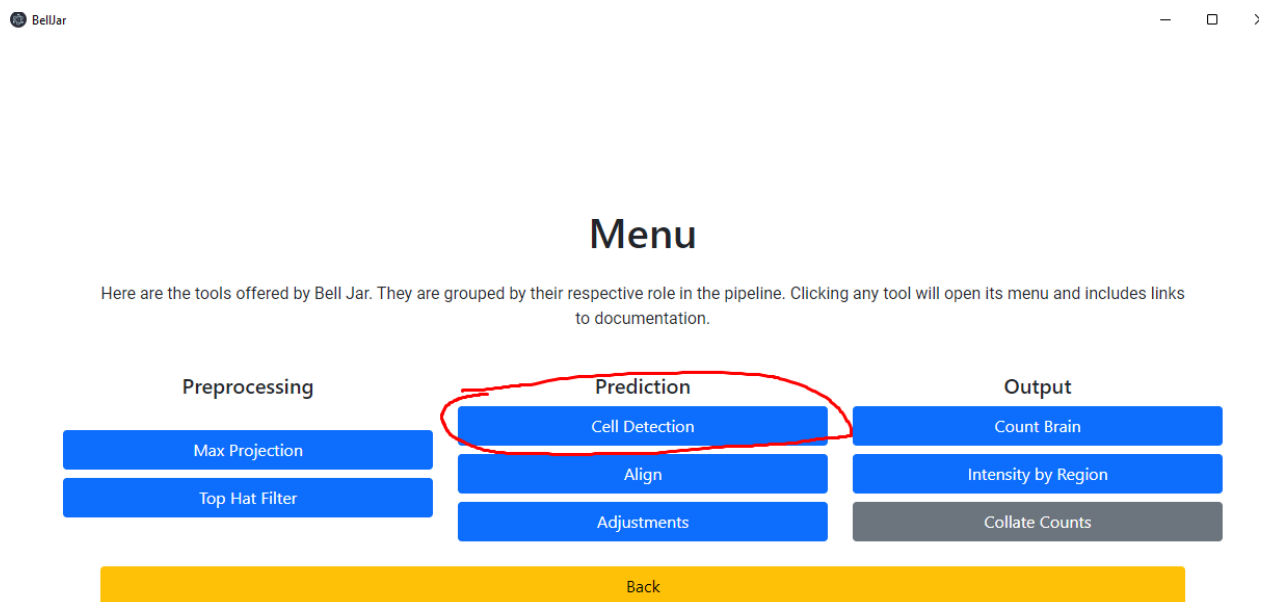
Usage:

1. Create a new directory for the outputs of the process. (e.g., Brain42_count).
2. Select the directory where your raw prediction pkls are located as the predictions path (e.g., Brain42_detections).
3. Select the directory where your raw annotations pkls are located as the annotations path (if you adjusted your annotations, select the adjustments directory, e.g., Brain42_alignment or Brain42_adjusted).
4. Select the new directory you created as the output path.
5. Run the program and check the loading bar or monitor the output folder for the result.

4. Processing the sample dataset

We have provided a minimal example case of using Bell Jar to align the sections to the Allen brain CCFv3 and count the cells by region in some experimental tissue. Please first download the sample dataset from [here and unzip it](#). All directories have been created for you, and the sample images have been preprocessed.

Click 'Menu' on the homepage.



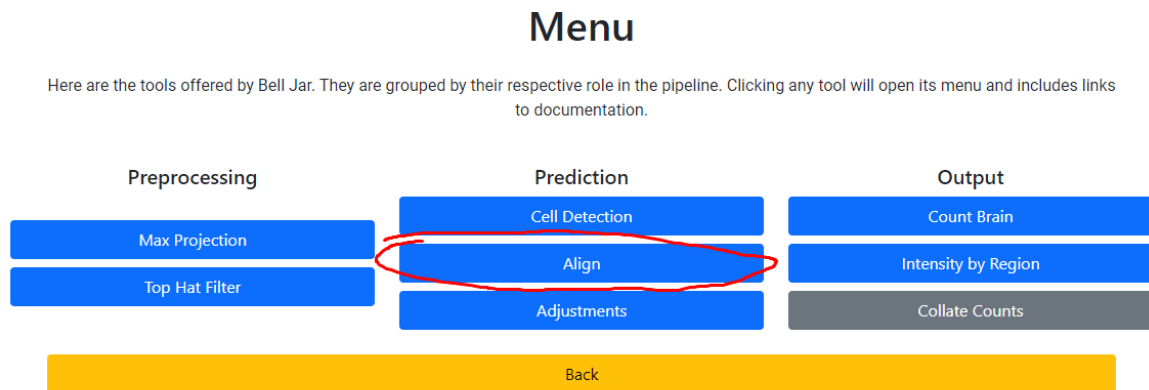
1. Begin by clicking to run 'Cell Detection' on the "Cell Images" in 'belljar-sample' folder provided to obtain detections. Specify Input Path as a folder where the

image files are located (e.g. ~/belljar-sample/Cell Images) and Output path as a folder to store the processed images (e.g. ~/belljar-sample/Detections).

2. If cell detection is completed, move to 'Aligning' step (#3).

 BellJar

— □ ×



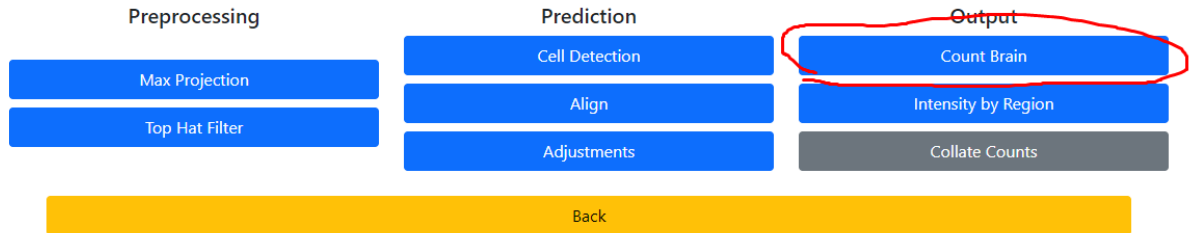
3. Run the aligning step on the provided “DAPI images” to obtain an alignment. Alignment takes your experimental tissue sections and helps align them to the Allen Brain Atlas CCFv3. A predictor network will evaluate your tissue and suggest the best fitting sections. Afterwards you will have to fine tune these predictions to best match your tissue.

Input Path = ~/belljar-sample/DAPI mages (or your experimental DAPI images)
Output Path = ~/belljar-sample/Alignment

The Alignment method you choose depends on how you processed your tissue. The sample data is 'single hemisphere' so it should be set as such, if your experimental data is whole brain then choose 'Both hemispheres' option.

Menu

Here are the tools offered by Bell Jar. They are grouped by their respective role in the pipeline. Clicking any tool will open its menu and includes links to documentation.



4. Finally, run the 'Count Brain' routine to obtain cell counts by region for the brain.

Predictions Path = belljar-sample/Detection

Annotations Path = belljar-sample/Alignment or belljar-sample/Adjustment if you made any adjustment

Output Path = belljar-sample/Counts

Feel free to compare counts against the bounding box and map images of the sample detections and alignment. count_results.csv will be found in the 'Counts' folder, thus Output path.