Bell Jar Guide

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.

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.
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.
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.
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. This 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 tissue sections as your background-stained 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)).

Usage:

1. Create a directory for the outputs of the process.
2. Prepare your input 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.
3. Select the output path to the new directory you created.
4. Select whether these are whole brain sections or single hemispheres (if you select hemisphere, your images must be in a normal coronal left hemisphere orientation).
5. (Optional) If you had poor results with the automated predictions or your tissue is particularly degraded, you may manually select the cut angle of the brain with the Angle Override option.
6. Click ‘Run’ to initiate the program.
7. During execution, the script will first predict the cut angle of your samples. Afterward, it will select approximate atlas matches to your tissue sections, 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).
8. You must tune each section to the best matching atlas section by using the horizontal scroll bar to move anterior to posterior through the reference atlas.

9. (Optional) If the cortex in any section is separated from the midbrain, you may check the “Separated?” box to apply a filter that will help better fit them to the atlas.
10. Once satisfied with your selected match, click “Next” to move on to the next section.
11. Repeat this process until you arrive at your final section, and then click “Done” to write the results.
12. Monitor the output folder to check the progress of the resulting output. The warping will take 20-60 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.

c. Adjustments

‘Adjustments’ tool takes in a directory containing the annotations and DAPI/Nissl sections 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.

Usage:

1. Create a new directory clearly labeled as adjustments.
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 images in the new directory (preserve naming).
4. Select the input path to the directory you have now configured.
5. Select whether you want to adjust in “Affine” or “Paint” mode.
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.
8. (Paint Mode) When in paint mode, you adjust the alignment by painting. To select a region to paint, hover over the map until you locate the desired region

to modify, and right-click to select it. Now with your left click, you can paint new boundaries for the selected region anywhere on the map. The Q key will save and advance to the next section, Z will undo any changes, and ESC will end the process.

9. Navigate and edit all the 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 we have developed that counts all the cells in an experiment processed with the prediction tools.

a. Count brain

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

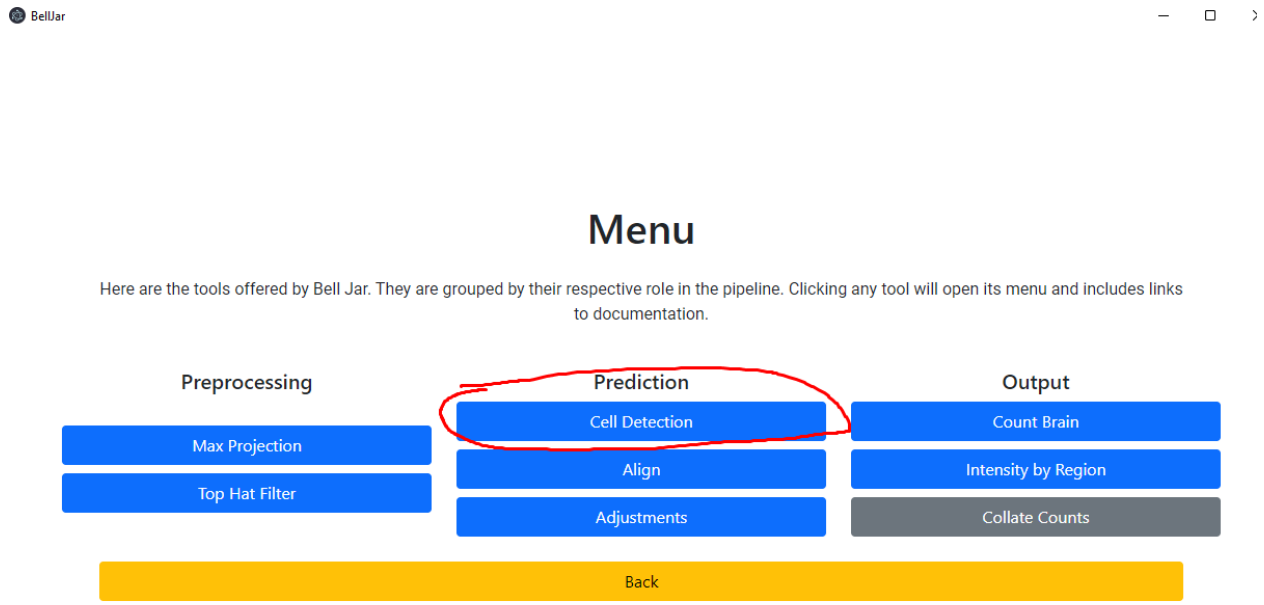
Usage:

1. Create a new directory for the outputs of the process.
2. Select the directory where your raw prediction pkls are located as the predictions path.
3. Select the directory where your raw annotations pkls are located as the annotations path (if you adjusted your annotations, select the adjustments directory).
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.

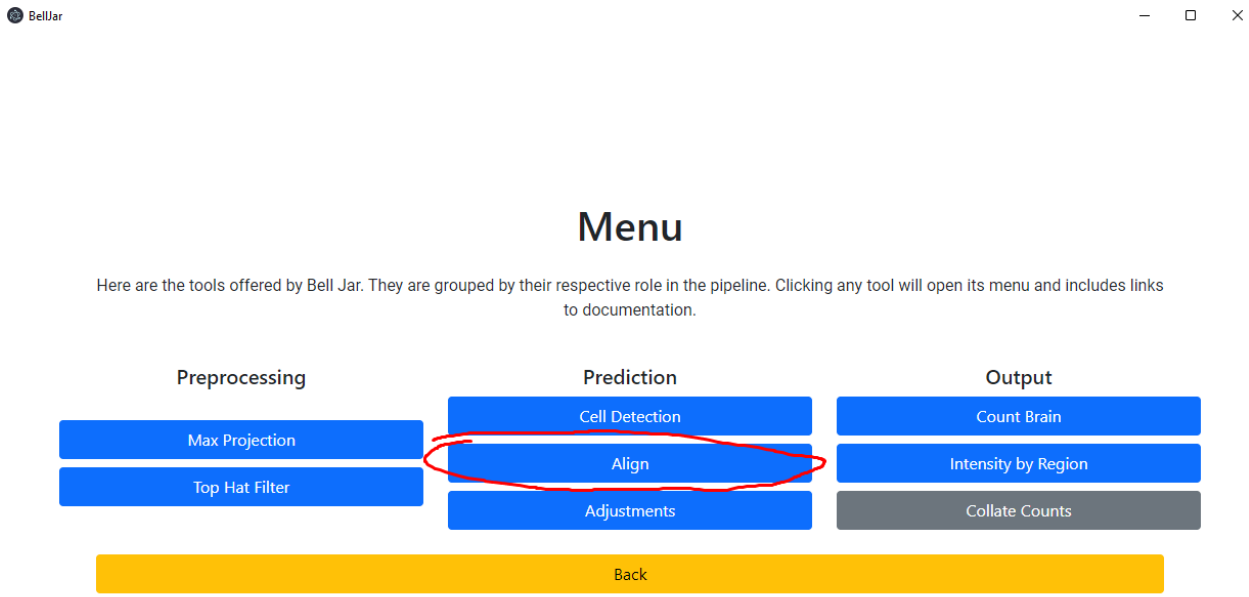
4. Processing the sample dataset

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 homescreen.



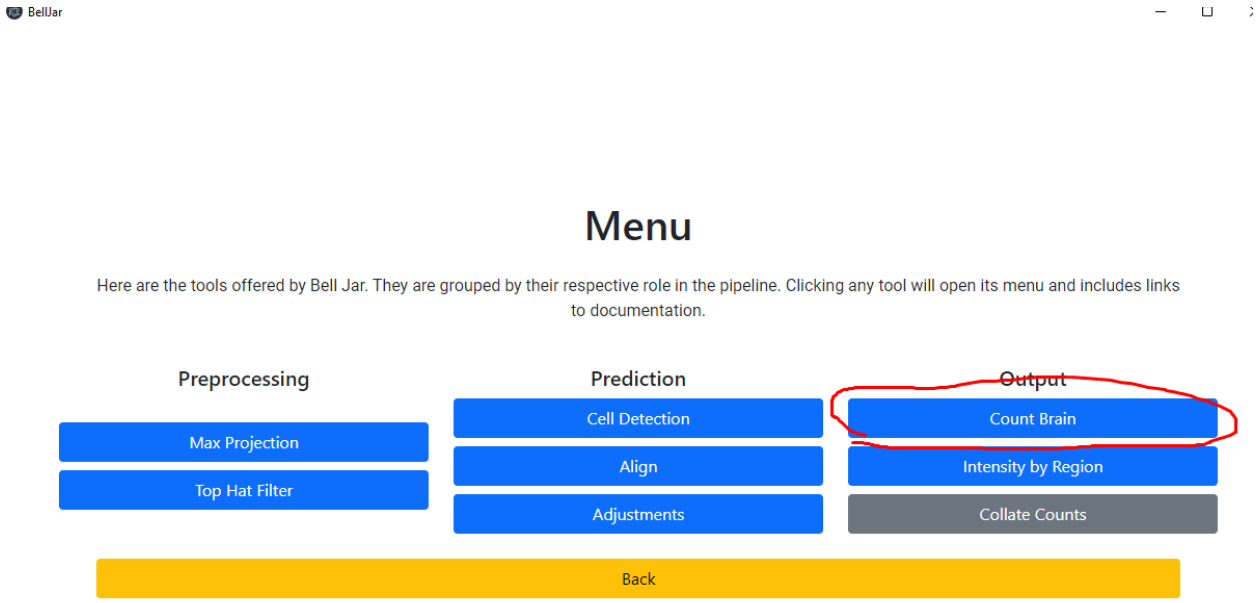
1. Begin by clicking to run ‘Cell Detection’ on the “Cell Images” in 'belljar-sample’ folder provided to obtain detections. How should I select Input path and output path? Specify Input Path as a folder where the image files are located (e.g. ~/belljar-sample/Cell Images) and Output path as a folder to store the processed images (e.g. ~/belljar-sample/Detections).
2. If cell detection is completed, move to ‘Aligning’ step (#3).
   ➔ Explain advanced settings (Tile Size, Confident Threshold, Custom Model).



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. 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 hemispheres so it should be set as such, if your experimental data is whole brain then choose that option. Angel override lets you force the brain to be aligned at a fixed Angel regardless of the predictors output. This is useful if you are getting strange predictions or just want to use the standard 0 angel atlas.



4. Finally, run the ‘Count Brain’ routine to obtain counts by region for the brain. Feel free to compare counts against the bounding box and map images of the sample detections and alignment.