# Anaconda3 Workstation

## Setting Up

Go to <https://www.anaconda.com/download/> and Download Python, version 3.7, or later for Windows. Do not download Python 2.7. **If your workstation already has Python 2.7 or Anaconda2 installed, be sure to uninstall it.** This is so you can access the Anaconda prompt from Anaconda3 / Python 3.7.

*Note: If you have Anaconda2 installed and you do not want to uninstall it, then simply run the following lines on the Anaconda Prompt terminal:* **$ conda create -n py36 python=3.6 anaconda  
 $ conda activate py36**

## Installing the Required Packages

Some of the libraries and packages used in this UI need to be installed separately. This includes:

### PIP

Pip is an installer used to install other required packages or libraries. Anaconda2 and Anaconda3 already have a version of pip installed, however this version is likely out of date. To update pip, run **python -m pip install --upgrade pip** in the Anaconda Terminal.

### PLOTLY

Plotly is a handy plotting library that comes with many weird and wonderful widgets and tools we can leverage. Plotly outputs also generally tend to be very visually appealing and neatly organized. To install plotly, run **pip install plotly** in the Anaconda Terminal. Note: It will take a few minutes to fully install.

### COLORLOVER

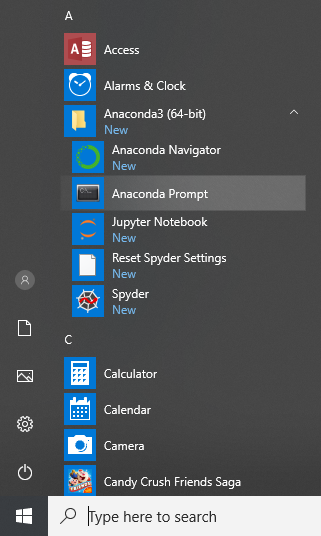
Colorlover is a neat colour scale library that is leveraged to apply colours to some of the legends and figures output in this UI. To install colorlover, run **pip install colorlover** in the Anaconda Terminal.

### OPENCV

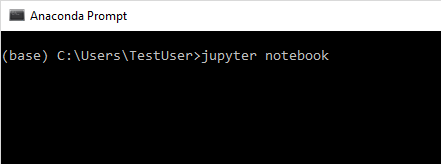
OpenCV is a library we use to import images. To install, run **pip install opencv-python** in the Anaconda Terminal.

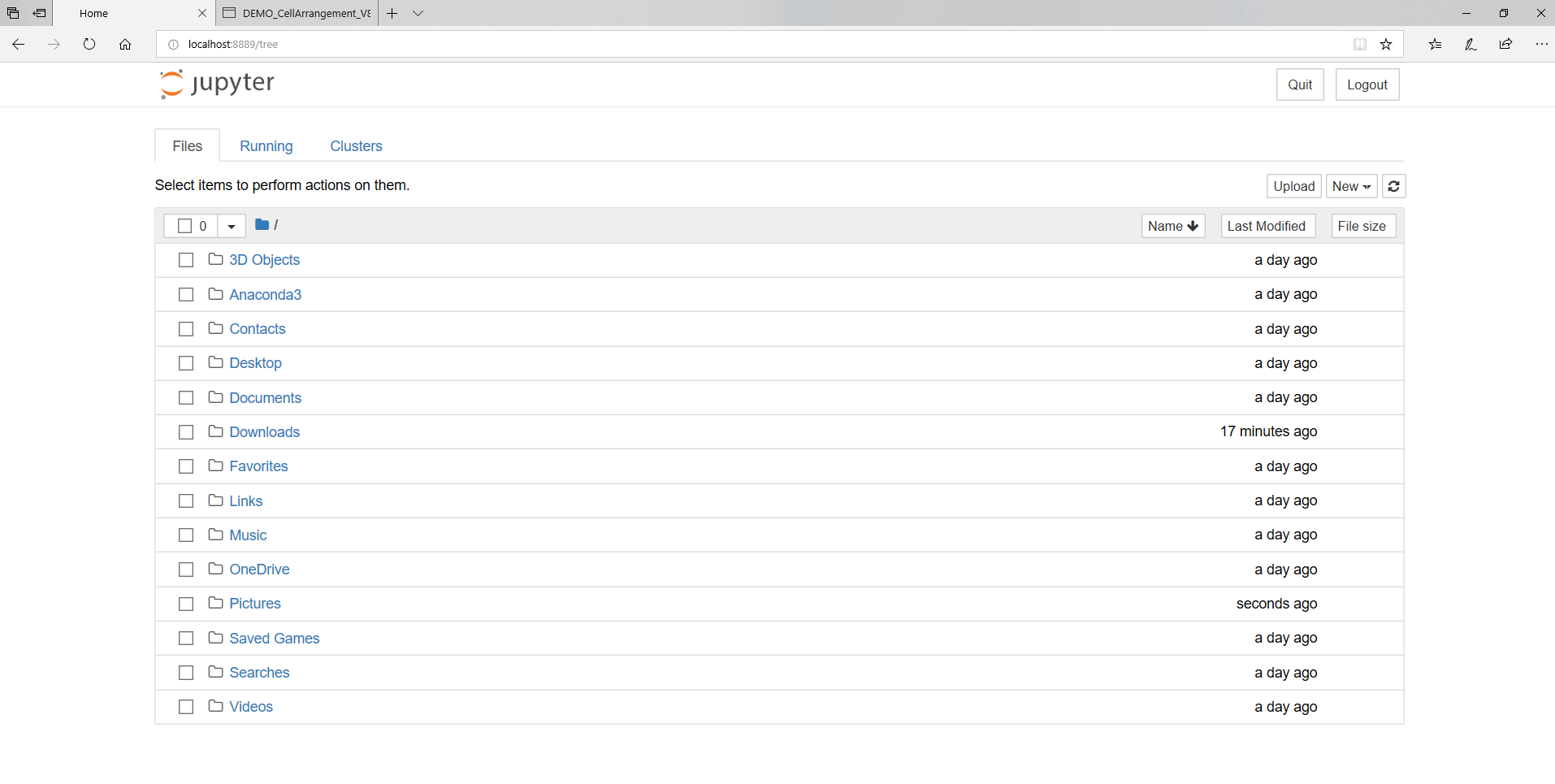
## Opening Jupyter Notebook

1. Download all the required files from the designated distributable folder, **DAPI\_Distributable** in the **Z:\C7Disk\Dima\DAPI\_Distributable**, to the desired folder. **The files must be saved to the local drive under the running user account:** C:\Users\<your user account here>.
2. Select the Windows icon in the bottom left of the screen, and search for the folder **Anaconda3 (64-bit)**. Open the folder and navigate for **Anaconda Prompt.**

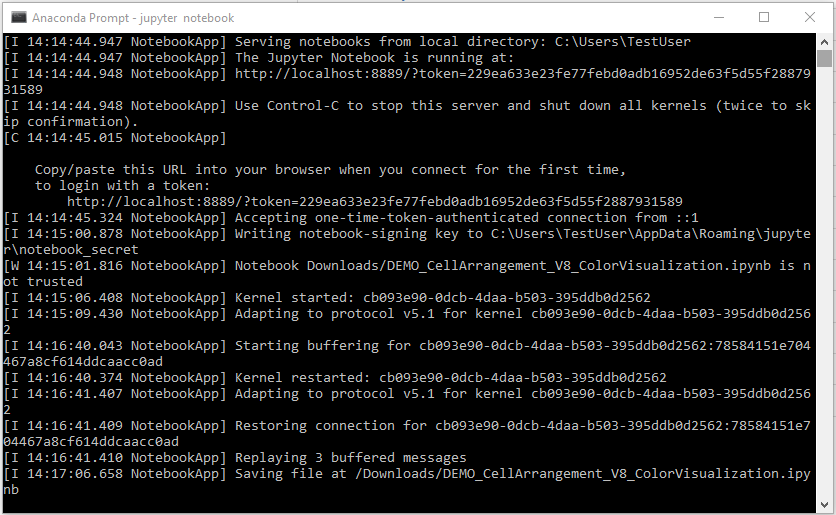
****

1. When selected, a black terminal will pop up. In this command prompt, simply type **jupyter notebook** (space included, no capital letters), and an instance of Jupyter Notebook will open up in the default browser. You will notice that although the interface will look like a webpage, the URL will read **localhost:8889/tree.** Jupyter notebook runs offline and does not need internet to run.

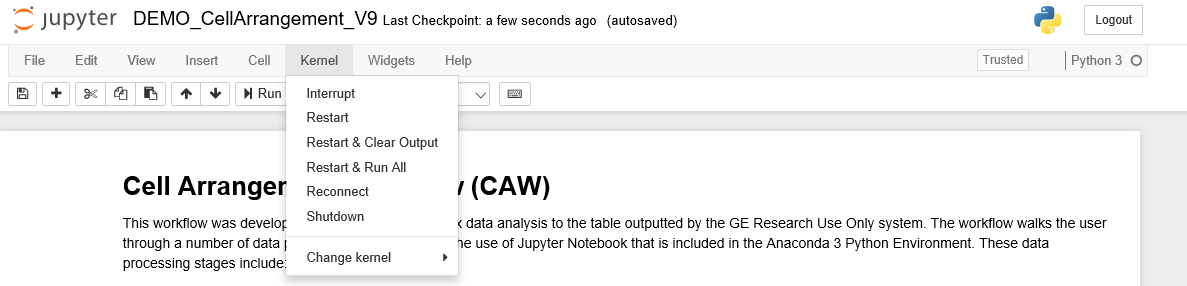




1. You will also notice that the previous Anaconda Prompt panel now logs some of the backend workings of the notebook (i.e. saving the notebook or clearing the terminal will be noted in the Anaconda Prompt terminal with a timestamp). **Do not close the Anaconda Prompt window, Jupyter notebook cannot run without the terminal running in the background.**



1. Navigate to wherever you saved the distributable folder in Step 1 of this section, and select the file titled **DEMO\_CellArrangement\_V9.ipynb.** It will open in a new tab.
2. In the top bad, select **Kernel** and select **Restart & Clear Output** to clear the workspace. Doing this between every run will ensure a clean and smooth workflow, but it is not necessary.



**You should now be fully set-up to use the interface!**

**Follow the next steps for a thorough step-by-step navigation of the workflow.**

# Cell Arrangement Workflow (CAW) v8.1

## Expected Inputs / Parameters

The User is expected to input a series of files, commands, & proposed file names to the UI in Steps 2 - 6.

### 2) Input Session Information

The User is prompted to enter the proposed information:

* User,
* Date of analysis session,
* Project Name,
* Project Notes,

The Date of analysis session is auto-filled, but the User may change this date if they desire. The User may also manually enter a User (i.e. username) or Project Notes, this is optional. **The User must input a Project Name. The Project Name may only consist of alphanumeric characters.**

### 3) Assign and Load the Files

The User is required to upload the following information:

* A GE table with the required data (i.e. cell coordinates),
* A corresponding DAPI image to the GE data file,
* A designated folder where the UI will output all results to,

**The uploaded GE file must be in a pivot table format (.tab extension).** The corresponding image file can be any valid image file type (.png, .jpg, .tif). **The designated folder may not contain a folder within, with a folder name that matches the proposed Project Name, outlined in (2).**

### 4) Select the Biomarkers to be Analysed

The User is prompted to check all relevant biomarkers they would like to include in the analysis. By default, all the biomarkers are selected.

### 5) Cell Quality Control Parameters

The User can restrict cell selection based on parameters such as:

* Cell Nuclear Area: the size of the cell’s nucleus, in pixels.
* Cell Membrane Area: the size of the cell’s membrane, in pixels.
* Total Cell Area: the size of the whole cell, in pixels.
* Total Cell Perimeter: the perimeter of the whole cell, in pixels.
* Total Cell Eccentricity: symmetry of cell. An eccentricity value of 1 indicates a spherical cell.
* Total Cell Major Axis Length: length of the longest side of an oblong cell.
* Total Cell Minor Axis Length: length of the shortest side of an oblong cell.
* Quality Control Score: this value is provided by GE Layers, it indicates the presence of the cell in all scanning iterations (stain-scan-bleach-repeat). A QC score of 1 is desirable.

**Note: you will notice two normal distributions in Cell Eccentricity.** This is due to a high population of very small cells with an area of 4 or 9 pixels often being classed as “perfectly spherical” (eccentricity = 1). To mitigate this, you can restrict the lower bound of the Total Cell Area parameter so as to not exclude large, perfectly spherical cells from your analysis.

### 6) Cell Biomarker Expression Data Settings

The User is now prompted to select a labelling and classification method. They may either use:

* **Hard Thresholding by Biomarker:** this is selected if the user knows how many distinct groups to separate the data into, and what threshold values to separate the data at.
* **K-Means by Biomarker**: this is selected if the user knows how many distinct groups to separate the data into but does not know what threshold values to separate at.
* **K-Means Grouped Biomarker**: this is selected if the user does not know how many distinct groups to separate the data into nor the threshold values to separate at.

## Next Steps

There have been numerous suggestions that were made to me in the last few weeks of my term, that I was not able to implement in my time. I have noted them all and will list them below for any future developers to consider when undertaking this project and developing it further:

* implement 3\*n image preview panels
* K means by grouped biomarker: this section of the code in (6) above was developed by a previous developer who has since left. When I inherited the code, I didn’t have time to play around with or test if this part of the code even works. Troubleshooting this to make it work would be nice, although not needed. Talk to the users to see if this is a feature they would like to have implemented.
* The DAPI workflow is able to take multiple file inputs and concatenate them into one massive file. However, when I wrote the concat library, I wrote 2 distinct ones for vHE and DAPI, and did not have time to fully implement the DAPI concat library. Look into implementing the appropriate library into the DAPI workflow (ConcatFunctions4.py)
* To leverage full panning and zooming plotly features on the visualization, it would be nice to flatten the matplotlib image output as a flat image file (points correctly plotted on matplotlib) and then plot this image onto plotly. While it would be ideal to figure out how to accurately plot points on an image in plotly, for some reason plotly misbehaves when I tried implementing this over my term. (I imagine it’s due to high data points).
* In section 6 where the UI suggests threshold values, be sure to round off suggested values to an appropriate number of significant figures.
* Output the legend in section 7 as csv. (The legend is the table that explains what each cell type is made up of.)
* Prompt user to input vHE image, and plot points against vHE instead of DAPI.
* Label all axes in histograms.
* Implement an additional cell with radioboxes to allow user to select which biomarkers they want to be plot against the DAPI/vHE background. This way they can create their own unique outputs and look into the biomarkers they are interested in.

### References

If you have any questions or concerns, feel free to reach out to me by phone or email at:

Email: [dimaismail79@gmail.com](mailto:dimaismail79@gmail.com)

Phone: (647) 770-2412

Cheers, and best of luck!  
Dima Ismail