**Software Requirements:**

ImageJ (FIJI)

Jupyter Notebook (Anaconda Package-Python ver2.7)

Package Dependencies:

-scipy

-skimage

-tifffile

-tqdm

**Image Acquisition:**

SensorFRET analysis requires imaging of the same sample volume at two different excitation frequencies (ex1 and ex2: 405 and 458nm for the example provided). The two images acquired need to be spatially registered in both the image plane and focal height. The sensorFRET software includes an image registration function to account for lateral drift between images. The focal height at the two excitation frequencies can differ slightly due to drift or chromatic aberration, and therefore, depending on the quality of the objective lens and the time between acquiring each image, may require refocusing. Additionally, the pinhole size should be maintained between frequencies so that photons are collected from identical volumes of the sample.

Power should be adjusted at each excitation frequency such that the best signal to noise is achieved while sample bleaching must be minimized. Detector gain should be kept constant.

Spectral image stacks must be acquired at the same emission wavelengths as the calibration. The example provided was acquired with 32 wavelength channels ranging from 416nm to 718nm, which is the full range available to the Zeiss 710 LSM used. In theory, only 2 channels are needed to unmix the donor and acceptor fluorophore signals of the FRET pair, however higher spectral resolution greatly reduces the uncertainty of the unmixing process and also allows the autofluorescence contribution to be removed.

**Masking:**

Masking can be carried out in ImageJ, Photoshop, or another image processing program. For each pair of images a single channel 8-bit binary image mask should be generated, with 0 value pixels excluded from the analysis and 255 value pixels included in the analysis, greatly reducing the time and memory required to perform the analysis. If you wish to analyze the whole image, simply make an image the same size as the spectral images with all pixels having a 255 value. Choosing the masking method is dependent on the experiment and up to the researcher as shown by example masks below. If desired, separate masks for ex1 and ex2 can be generated to ensure the features analyzed are apparent at both excitation frequencies. The pySensor program will calculate the product of these masks when importing the images, thereby analyzing only the pixels which are included in both masks.

Any contiguous mask region can be analyzed as separate observations in the SensorFRET software, allowing, for example, FRET to be determined on a per cell basis when there are many cells in a single image.

A

B

C

D

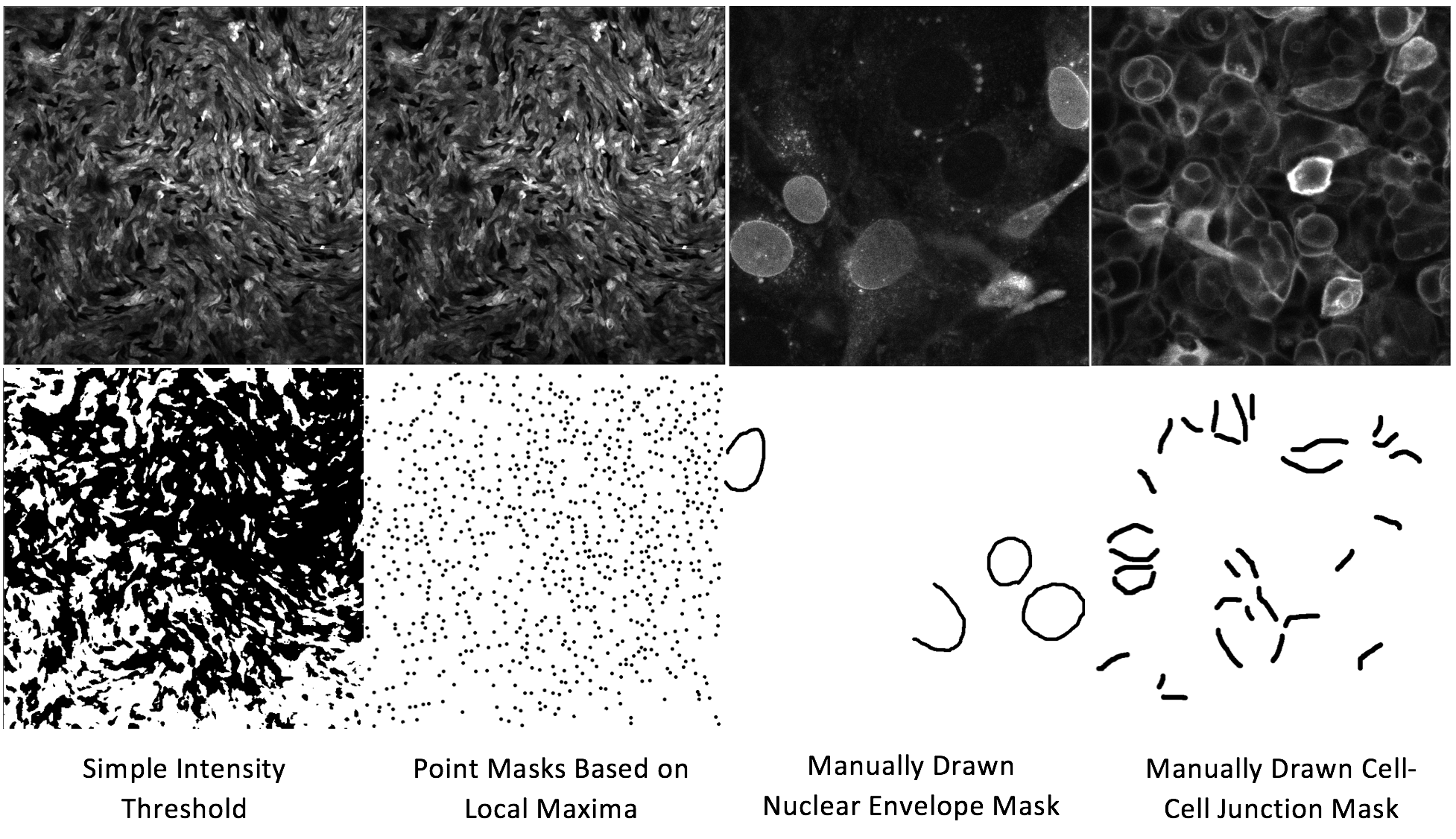
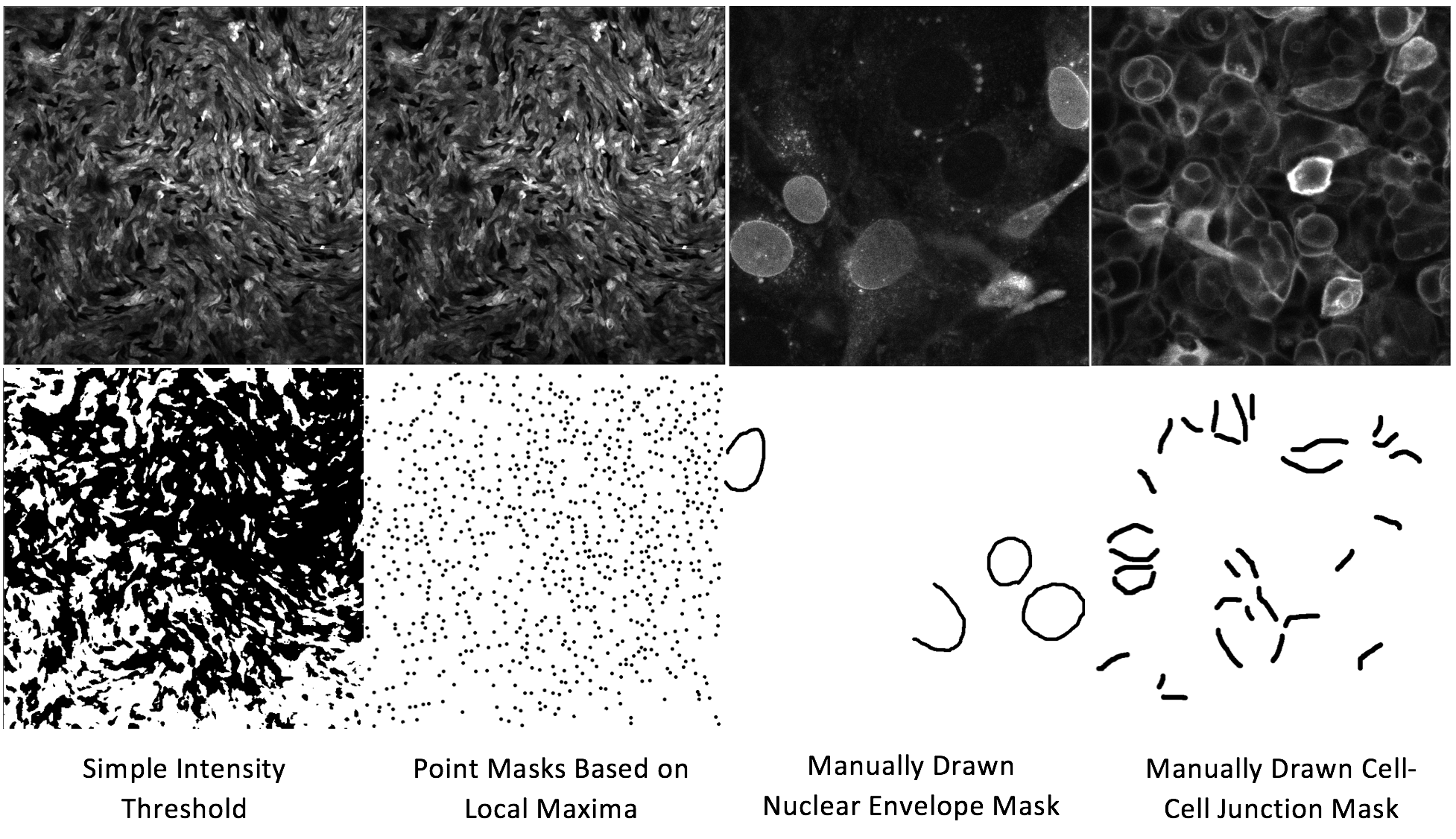
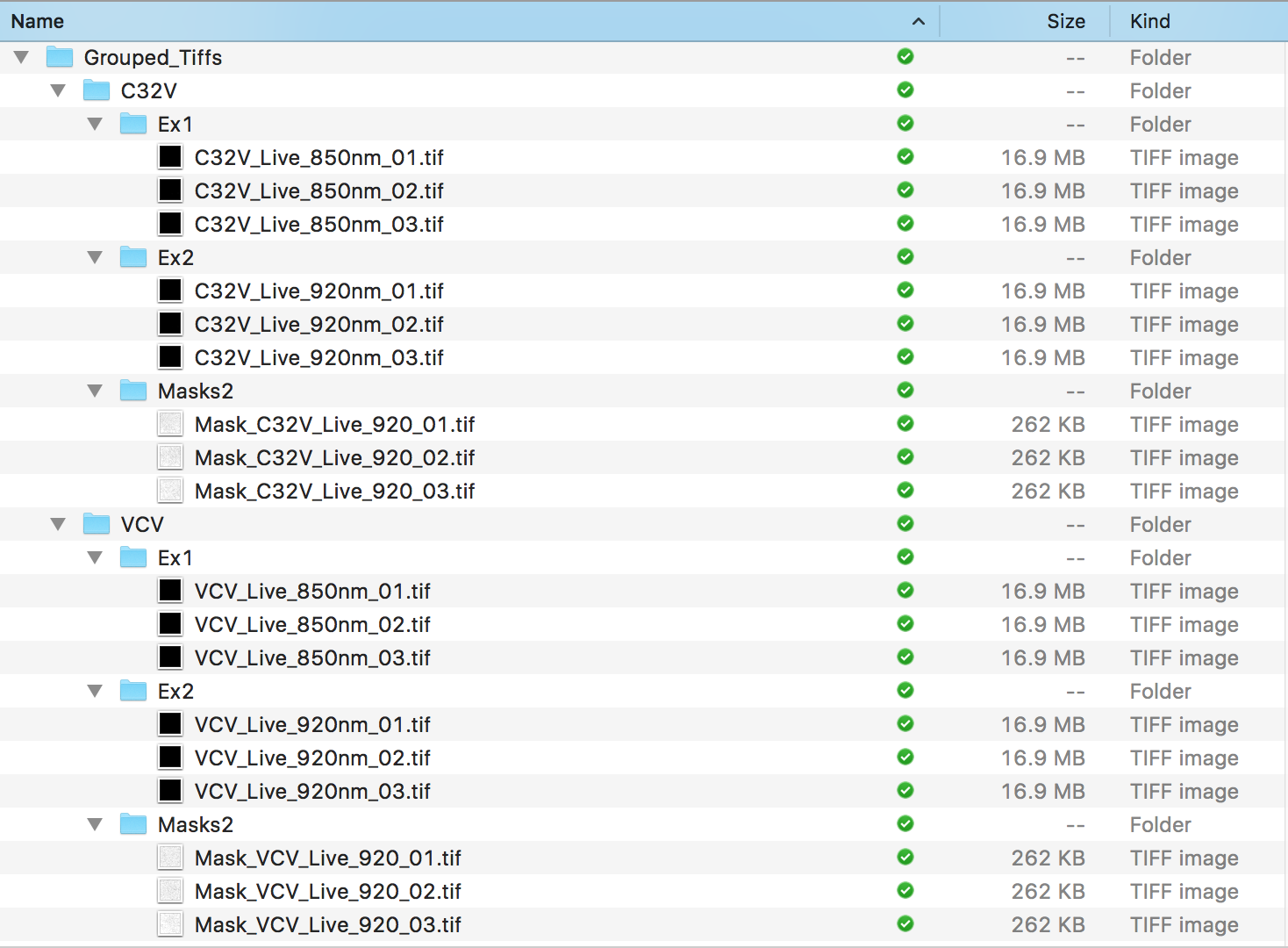


Figure 1: Example Masks: A) Intensity threshold (useful for excluding oversaturated and low signal pixels), B) Point masks (useful for sampling a large number of cells when the morphology doesn’t matter), C&D) manually drawn masks (required for isolating particular features within an image, such as the nuclear envelope or cell-cell junctions).

**File Structure:**

The pySensor software requires a specific file structure within the ‘Grouped\_Tiffs’ folder in order for the data to be read in correctly. Within the Grouped\_Tiffs directory there should be a subdirectory for each experimental group to be compared (C32V and VCV in the example shown below). The names of the first layer of subdirectories are used for labeling and legend entries within the software. Within each of these group subdirectories there are folders for Ex1, Ex2, Mask1, and an optional Mask2. The Ex1 and Ex2 folders contain the spectral image stacks in .tif format for the lower and higher wavelength excitations, respectively (405 and 458 in the example case). The Mask1 and Mask2 folders contains the single channel binary images used to tell the program which pixels to analyze. In cases where the masks are manually drawn such as Figure 1 C&D, a single mask for both excitation wavelengths is often sufficient and the Mask2 folder can be omitted. In some cases, such as when saturated pixels need to be excluded in both spectral images, it is advantageous to create masks for both excitation frequencies and then the program will only analyze pixels which are included in both mask files.



Low Wavelength Spectral Images

High Wavelength Spectral Images

8-bit Binary Masks

(additional mask folder is optional)

Second Experimental Group with Identical File Structure

Figure 2: Required File Structure for the Grouped\_Tiffs folder

First Experimental Group

**Running the Notebook:**

After properly masking and organizing the image data in the Grouped\_Tiffs folder, the pySensor.ipynb notebook should be opened (see Figure 3, Anaconda Navigator ->Jupyter Notebook -> pySensor.ipynb). Within the pySensor notebook, update the ‘User Input’ cell (Figure 3 #3) with the current experiment information (parameters explained below).

Name=string: Experiment Name which dictates the exported file names,

Calib\_File\_Name=string: filepath to the appropriate calibration file for your fluorophore and excitation frequency pairings

Interpolation=[X,Y] or False: Allows images to be resized to X by Y dimensions or False uses the native image resolution

Register\_to\_Ex1=Boolean: Determines which image is stationary during the image registration (eg. if masks were drawn using Ex1 this should be set to True)

Central Tendency= ’mean’ or ‘median’: Chooses which averaging function you want to use. Usually the differences are small but the median is preferable as long as the sample size is reasonably large. This is because ratio based quantities (such as FRET assuming the donor and acceptor are normally distributed) are Cauchy distributed which does not have a stable mean value.

Export\_Data=[list]: the export data list determines which observation dataframes (‘group’, ‘image’, ‘region’, and/or ‘pix’) are exported to the .xlsx spreadsheet at the end of the notebook. There is no harm in exporting all types of observations but exporting more than 50,000 pixel observations can take a considerable amount of time.

Once these parameters are chosen, the whole analysis can be run by selecting ‘Kernel’->’Restart & Run All’ (highlighted in Figure 3 #4). Alternatively pressing ‘Shift-Enter’ can be used to run cells individually and progress through the notebook. There are many settings available for changing the output of the plotting functions which are explained in the markdown cells of the notebook.

**Output:**

4

3

2

1

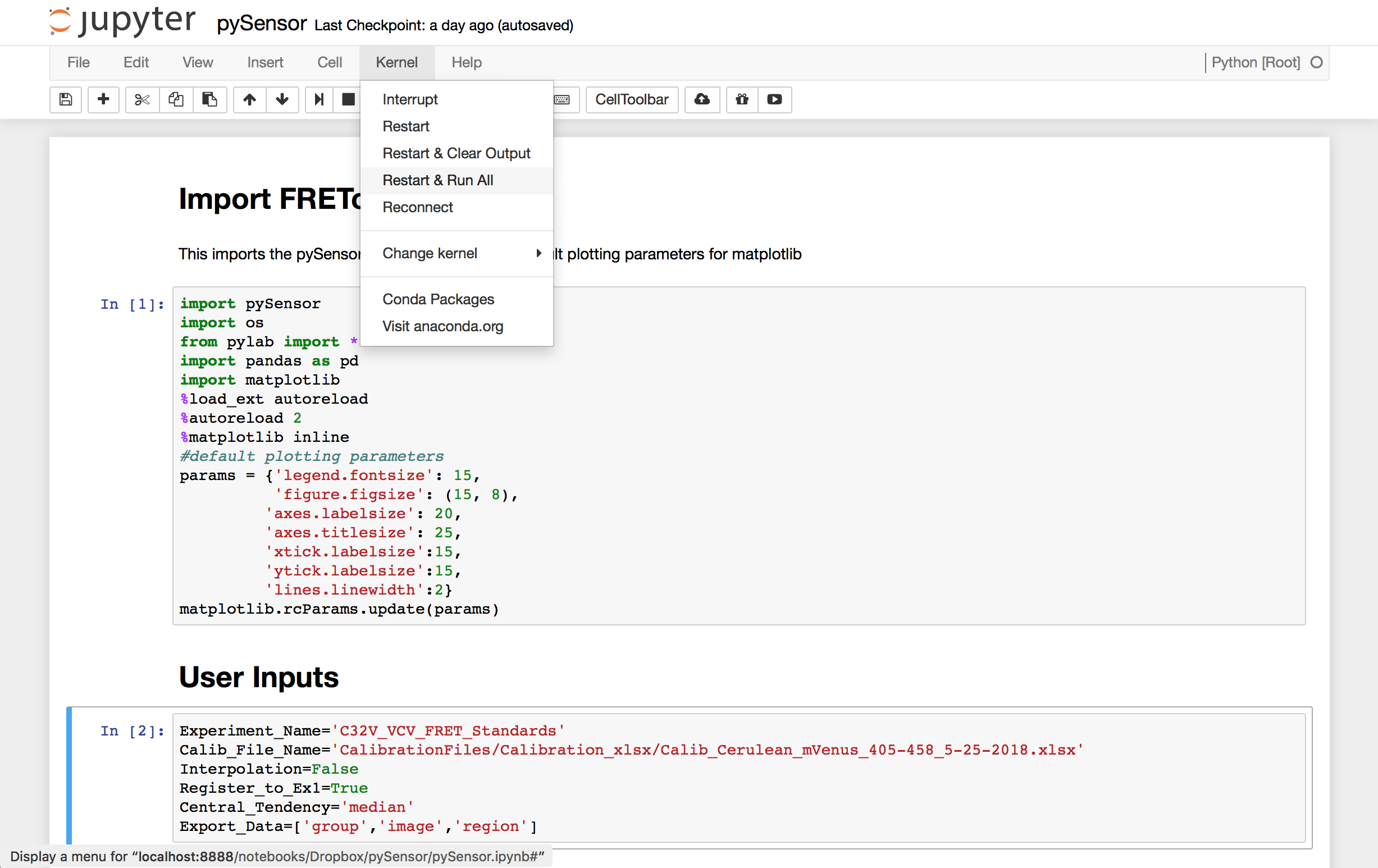
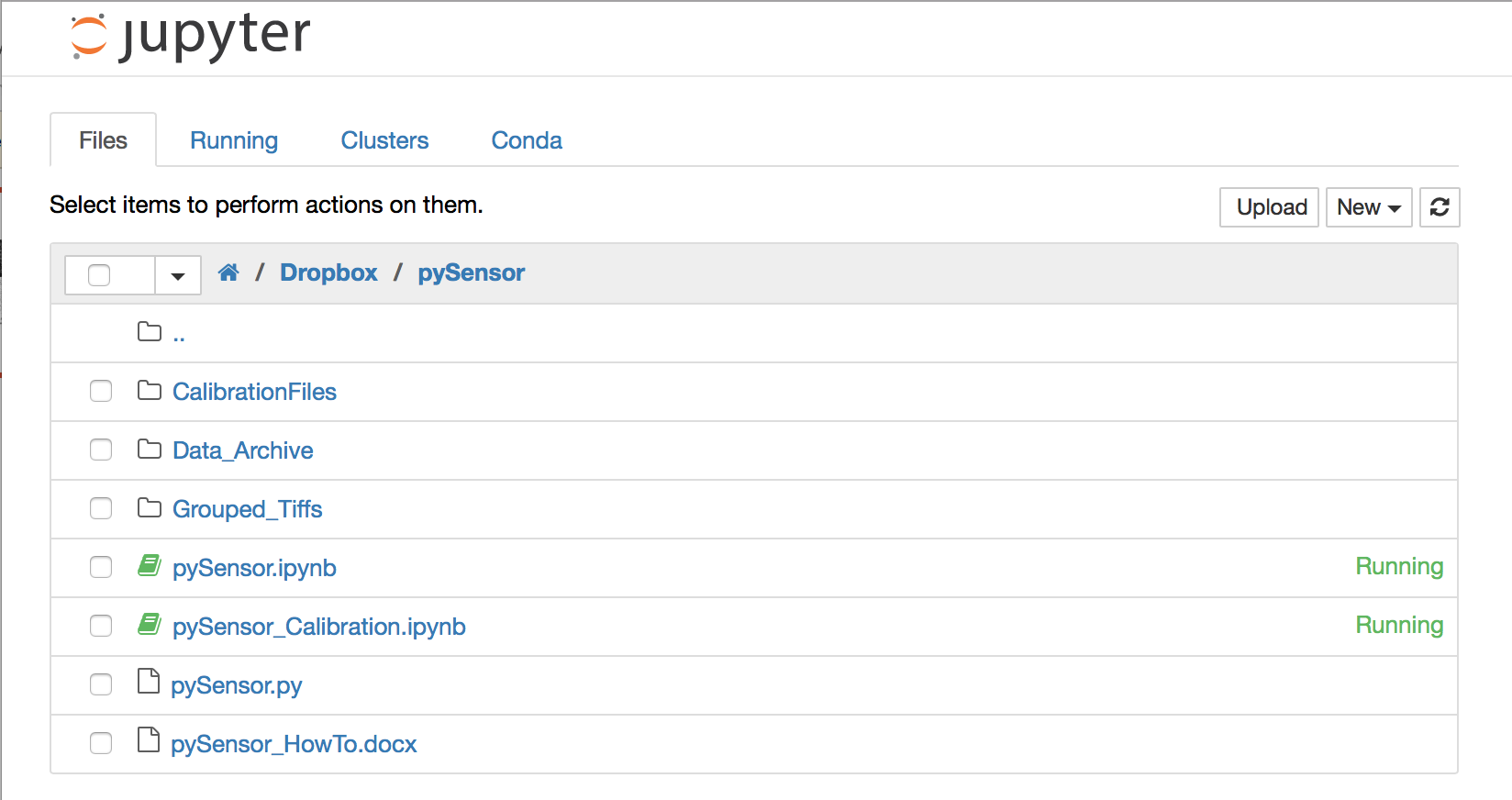
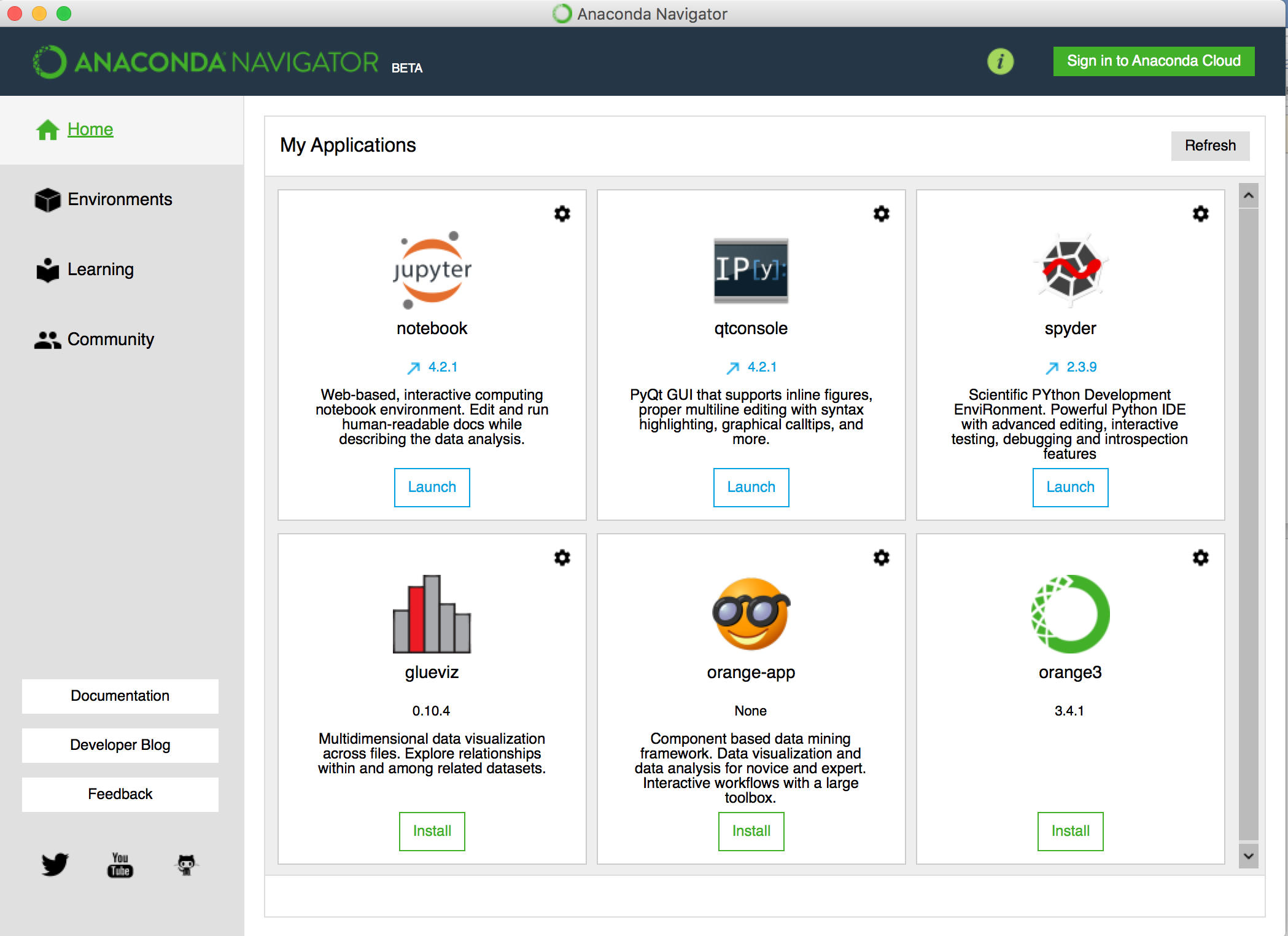


Figure 3: Opening the Notebook through the Anaconda interface.

After running the notebook the results of the analysis are stored within two files saved within the “Data\_Archive” folder. The first is an exported .html file, which is a copy of the whole notebook including any embedded plots and images that were generated (individual plots and images needed for generating publication figures can then be copy and pasted from this .html). The second is an excel spreadsheet containing the analysis settings used on the first page of the workbook (calibration information, image size settings, etc.) and the numerical data calculated based on the sensorFRET analysis (fluorophore magnitudes, acceptor direct excitation, FRET efficiency, etc.) for each observation type on the subsequent pages. Each of the calculated parameters are described in the Jupyter notebook. From the excel spreadsheet, custom plotting or analysis can be performed in the software of your choice (excel, R, matlab, etc).