These instructions apply only to this customized version of xraylarch that has been modified to run with python 3.X using Anaconda3. Begin by installing the software:

(Anaconda Prompt) *python setup.py install*

If additional modifications are made to the source code, this command will need to be re-run for those changes to take effect.

Launch the top level GUI with:

(Anaconda Prompt) *larch\_gui*

This GUI provides a Larch specific command prompt along with a hierarchical display of built-in functions and documentation. To analyze an XRF spectrum, we begin by loading the spectrum at the Larch prompt in the ‘data’ variable. We will use the included ‘Brass.txt’ file as an example.

(Larch) *data = read\_ascii(“Brass.txt”)*

Notice that ‘data’ appears at the bottom of the Larch command hierarchy on the main window. Unfortunately, the data provided by the Horiba XRF must be converted to a useful format before analysis. We do this by creating an ‘MCA’ data structure using the built-in command ‘create\_mca’ and saving it as an MCA file.

(Larch) *brass\_mca = create\_mca(counts=data.data[:, 1],offset=float(data.attrs.offset), slope=float(data.attrs.xperchan))*

(Larch) *brass\_mca.save\_mcafile(“Brass.mca”)*

Now that we have a properly formatted MCA file, we will analyze it using the Larch XRF spectrum viewer. Launch this from the Larch main window by selecting Applications > XRF Spectrum Viewer. Now open the newly created “Brass.mca”: File > Read MCA Spectra File. A blue spectrum should appear in the plot on the right side of the window with an approximately correct energy calibration. In order to improve the energy calibration, elemental lines must first be identified. Click on ‘Cu’ in the periodic table on the left. The characteristic Cu lines are added to the plot to assist in identification. Begin by zooming into the most intense line. This is the Cu K-alpha line. Position the cursor half-way down the left side of the line and left-click. A green cursor should appear and text at the bottom of the plot should indicate x-ray energy and number of counts. Now right-click on the right side of the Cu K-alpha line. An energy window from approximately 7.8 keV to 8.2 keV has now been defined. Label this as the Cu K-alpha line by entering “Cu Ka” in the next box directly below “Region of Interest”. Click “Add”. The line is now plotted in red and “Cu Ka” has been added to the list of regions of interest (ROIs). Continue by adding the Cu K-beta (“Cu Kb”), Zn K-alpha (“Zn Ka”), and Zn K-beta lines (“Zn Kb”). Note: the text of the ROI labels is extremely important and must be the elemental symbol followed by a space followed by the line identifier. Four ROIs have now been identified. Save the MCA file with File > Save MCA File.

Perform an energy calibration with Analysis > Calibrate Energy. The names of the ROIs have been used to pull the know line energies from the x-ray database for use in the calibration procedure. These are seen in the “Predicted Energy” column. These values likely disagree slightly with those in the column labeled “Center”. Perform the calibration by pressing “Compute Calibration” and save the results with “Use New Calibration”. Save the MCA file.

The final task is to obtain the counts in each peak. Click Analysis > Fit Peaks. Most of the settings shown are not yet used. On the “Fit & Background Settings” tab, ensure that the minimum and maximum energies bracket the lines of interest and that “Fit Background-Subtracted Spectra” is selected. The “Filters and Attenuation” tab is unused. Changes on the “XRF Peaks” tab should not be required. Click “Fit Peaks”. The spectrum has now been background subtracted and fit with a Gaussian peak for each line analyzed. The parameters of those Gaussian fits along with the estimated errors are printed at the Anaconda Prompt. Click “Done”. Notice that the spectral fit (background + Gaussian lines) is now plotted on the behind the experimental spectrum.