

(Step #1): Run `writeToExcel.ijm` in ImageJ

- Before running, make sure to change the location as to where you want .xlsx file to be saved

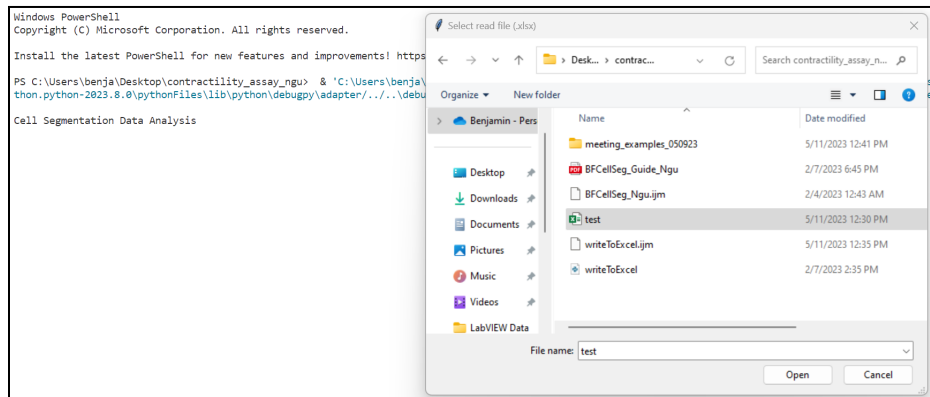
```
function processFile(input, input_image, output, file, image_file)
{
    // open file
    open(input_image + File.separator + image_file);
    print("Opening Image: " + input_image + File.separator + image_file);
    open(input + File.separator + file);
    print("Opening ROI: " + input + File.separator + file);

    // Do the processing here by adding your own code.
    roiManager("Measure");
    // Change location to save .xlsx file to
    run("Read and Write Excel", "file=[C:/Users/benja/Desktop/contractility_assay_ngu/test.xlsx]");
    run("Close");
}
```

(Step #2): Run `writeToExcel.py`

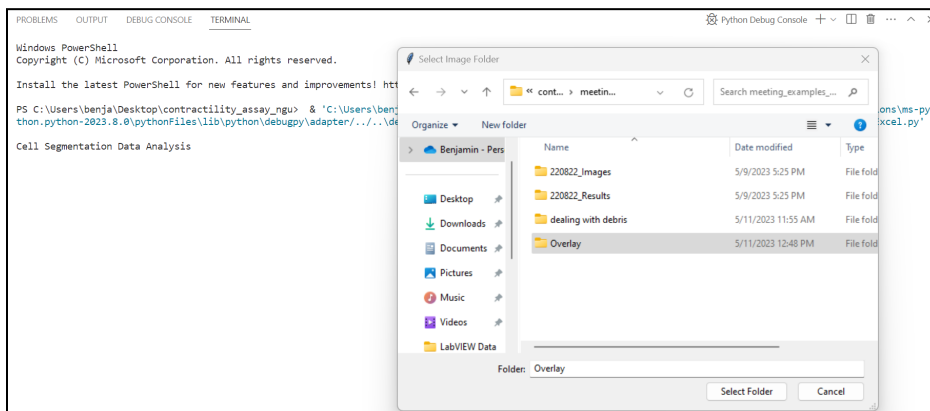
Example run done in Visual Studio Code (Make sure you have newest version of Python 3 downloaded)

- ❖ Select the .xlsx file generated from `writeToExcel.ijm`



- ❖ Select folder containing images

➤ Each image should be renamed in `treatment#_timept` format (ex: `1.1_0min`)



- ❖ Enter the name you want generated .xlsx file to be called



Example from command terminal (if you do not have an IDE)

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\benja> cd Desktop
PS C:\Users\benja\Desktop> cd contractility_assay_ngu
PS C:\Users\benja\Desktop\contractility_assay_ngu> python writeToExcel.py
Cell Segmentation Data Analysis
```

