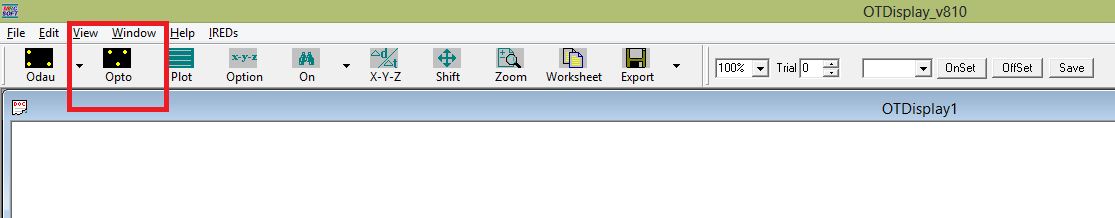
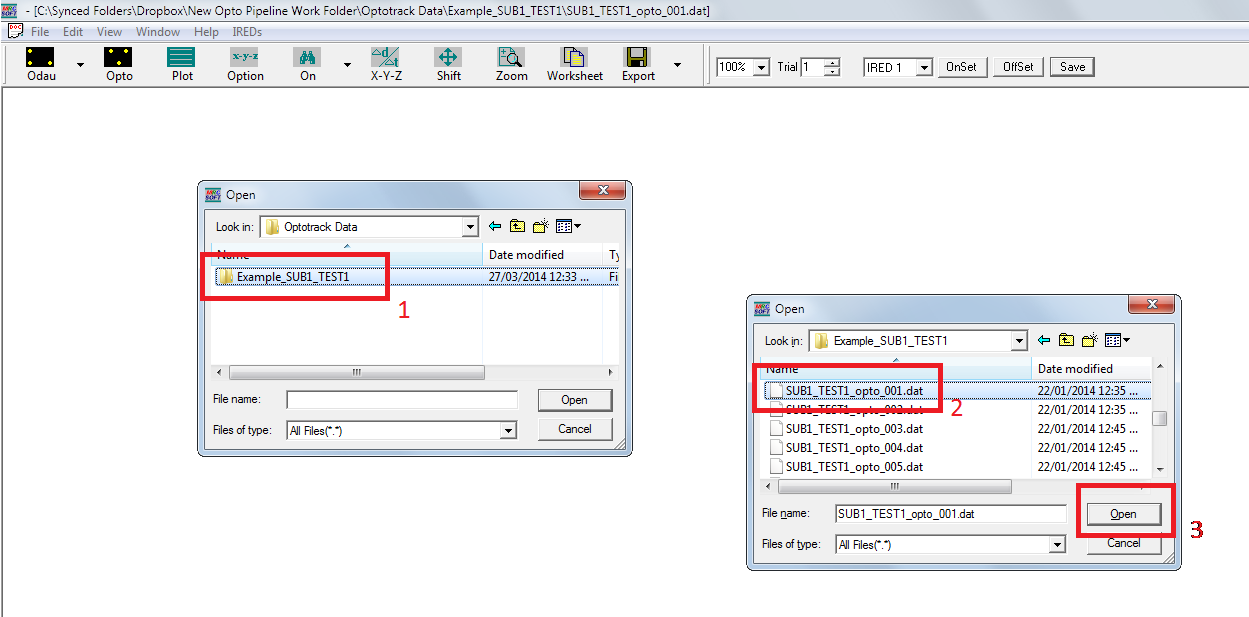
\*Before you begin, make a copy of the folder found in this dropbox to work in. Please do not work in the shared folder.

### Export opto data to txt files (This step cannot be performed on a Mac)

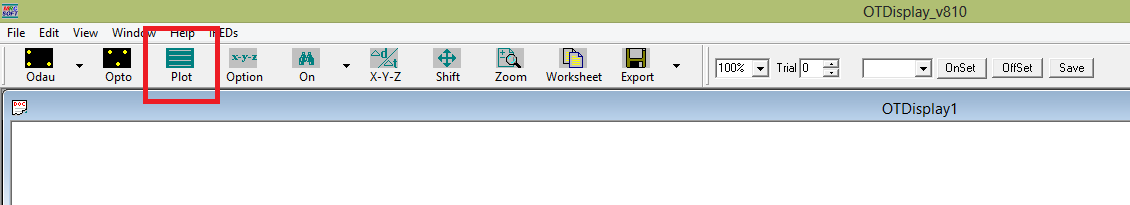
* 1. Open OTDisplay (v810)
  2. Click “Opto”



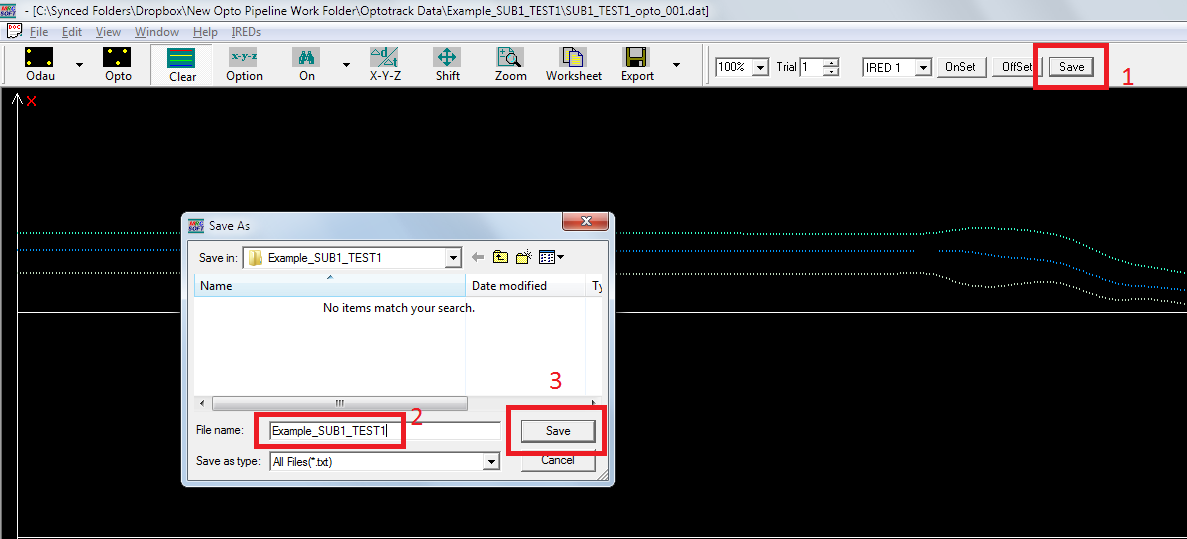
* 1. Navigate to where you have stored your optotrak data. Open the first file, which will have a name ending in “\_001.dat”.



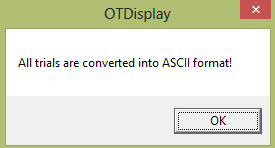
* 1. Click “Plot”. This step is required even for just saving data to text files.



* 1. Click “Save” in the top right corner (do not use File->Save As), enter a filename, and save. **Note: this filename will be used to name all subsequent files. You may think of it as the ID for the analysis. In this example, I called the analysis “Example\_SUB1\_TEST1”. Spaces and underscores are allowed, as are numbers.**

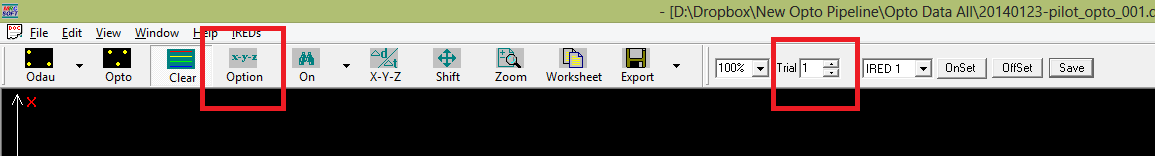


* 1. OTDisplay will cycle through all available trials (e.g., 001-142), saving them as filename\_###.txt (e.g., Example\_SUB1\_TEST1\_001.txt). Files are saved to the same folder as the dat files. A dialogue box will notify when complete.

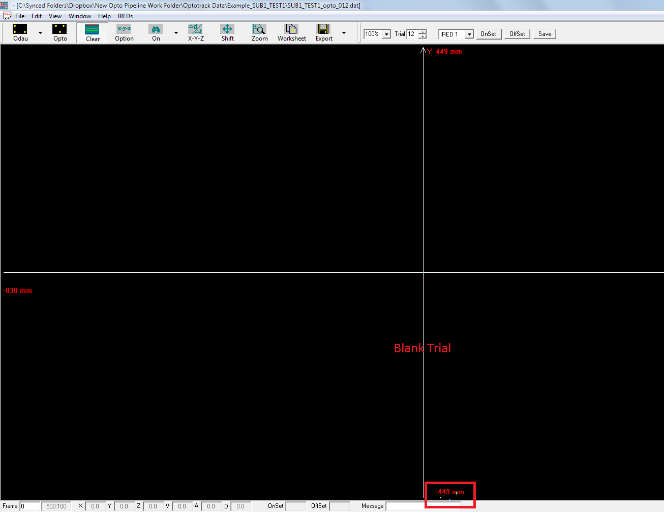
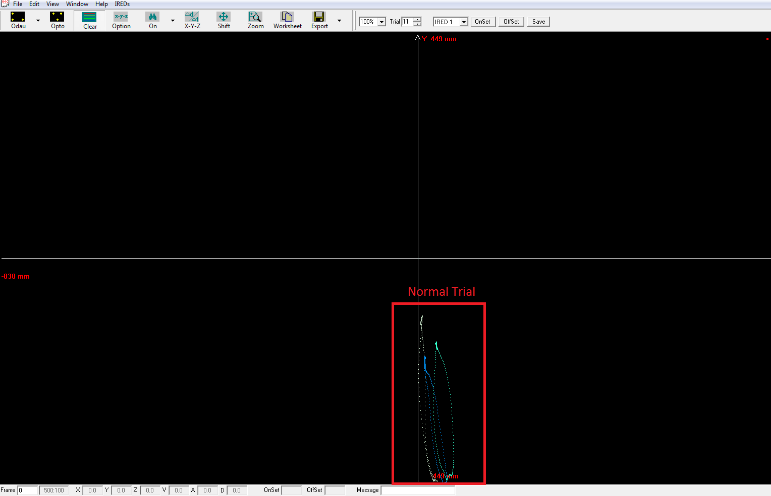


### If there are more files than there were trials, you will have to figure out which files need to be ignored.

* 1. Open up OTDisplay again. Load in the 001 dat file. Click plot.
  2. Use the arrow keys or Up/Down button to cycle through trials searching for the extra trials. It may be useful to change the view by pressing Option. Extra trials typically contain no motion or very bizarre motion (e.g., from scratching ones nose) making them easy to spot.



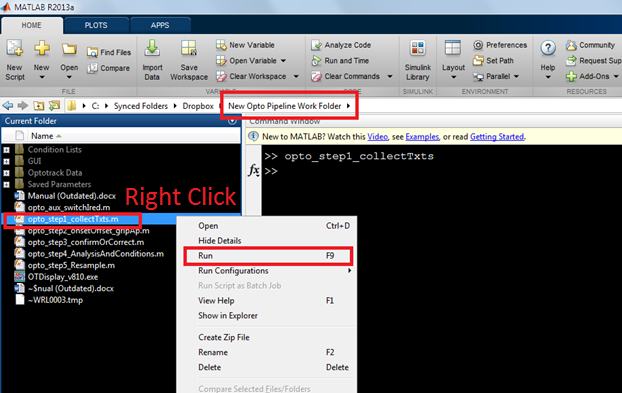
* 1. Delete the txt file for any extra trials. Leave the dat file – this serves as a backup. In this example, I deleted the txt files 006 and 012, leaving 140 files. To be safe, delete only txt files, leaving the dat files as a backup copy.

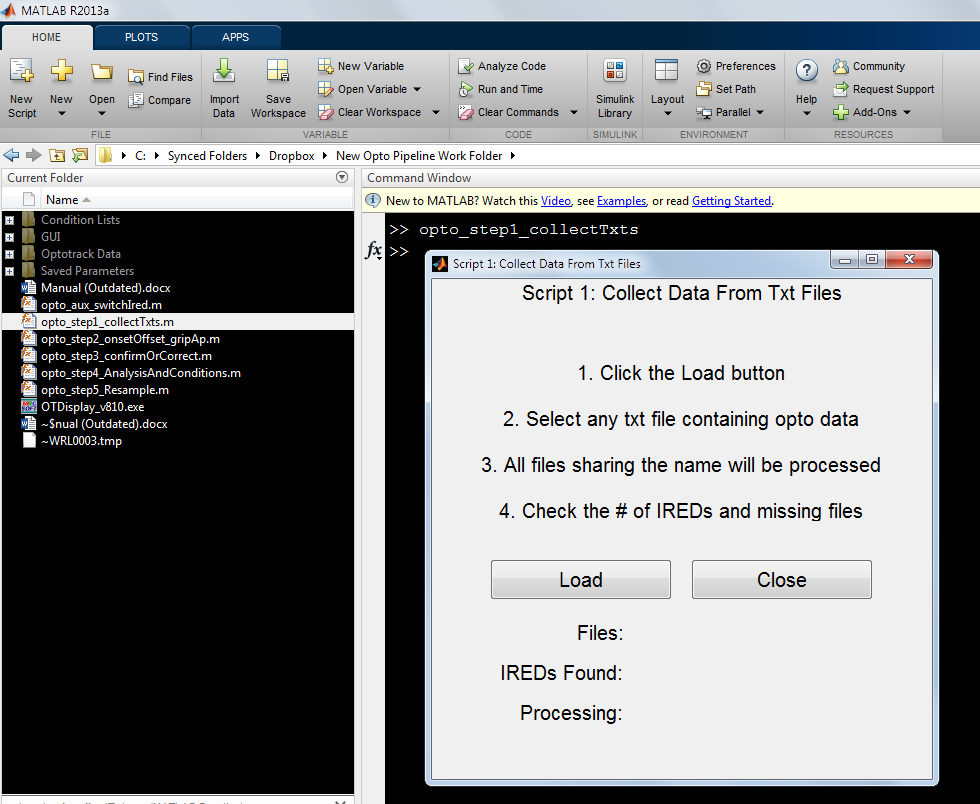


* 1. You should now have exactly as many txt files as collected trials, though the numbers on the files may not line up (e.g., trial 140 is called Example\_SUB1\_TEST1\_142, but it is the 140th file alphanumerically).

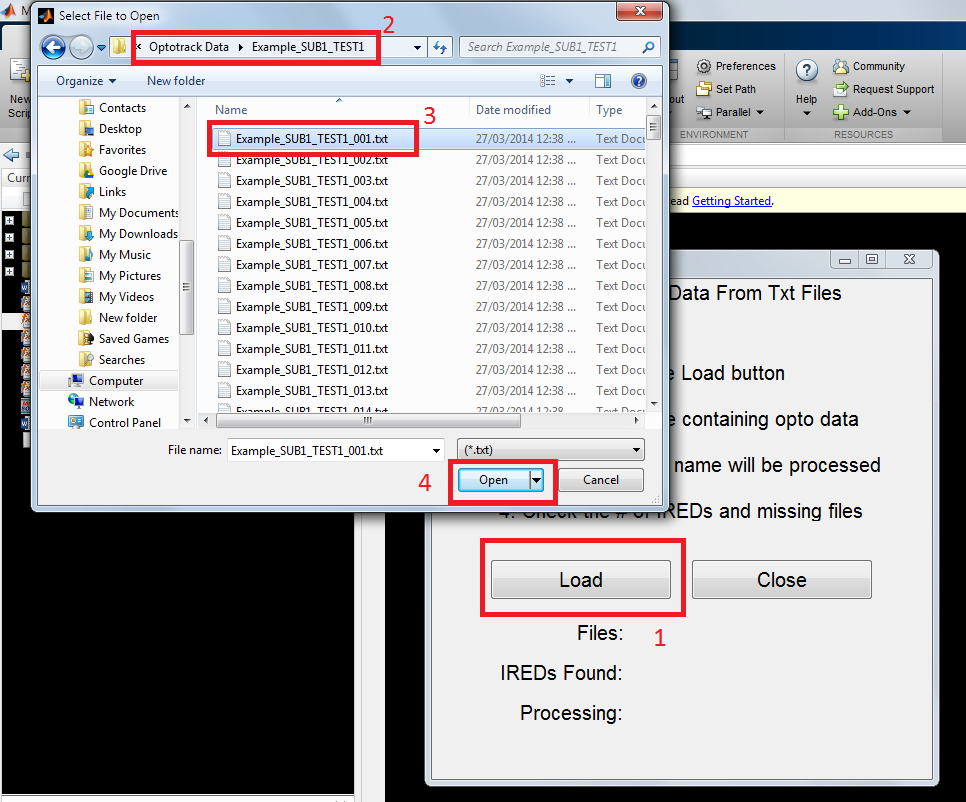
### Script 1: Load txt files into matlab

* 1. Open matlab and run “opto\_step1\_collectTxts”

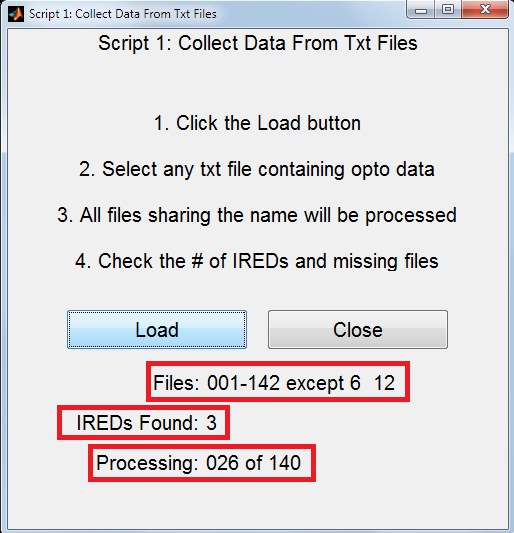




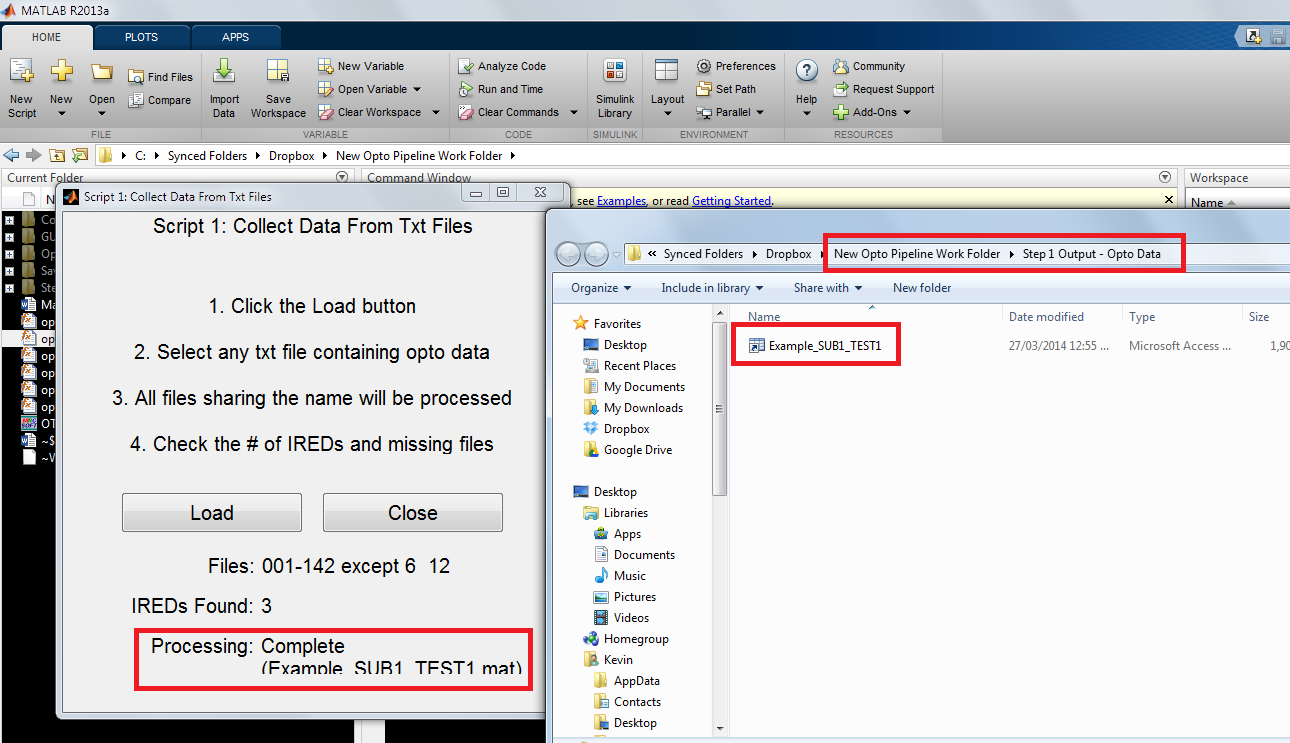
* 1. Click Load and select any one of the txt files from the previous step (it doesn’t matter which – this is done simply to get the filepath and naming convention)



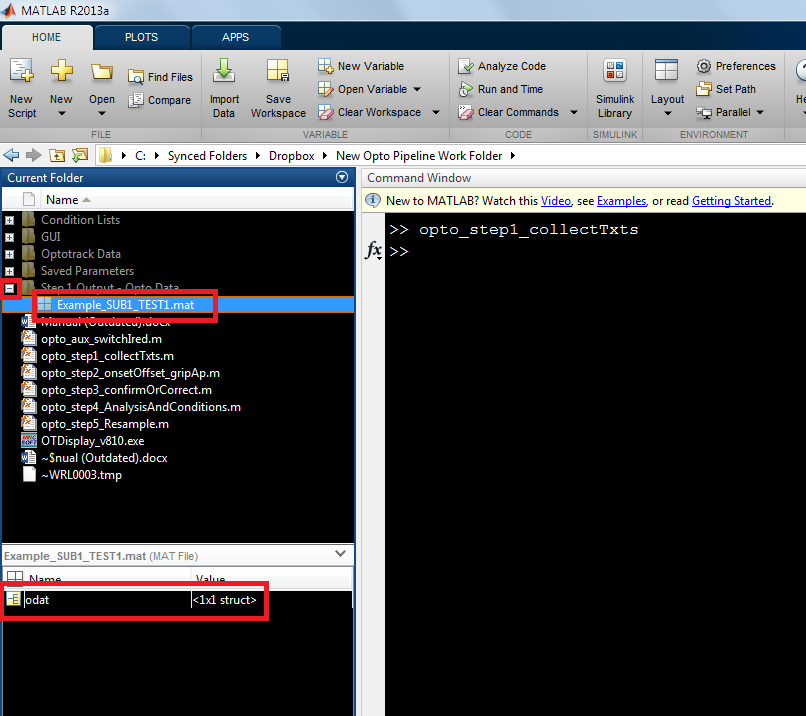
* 1. The GUI will indicate which files were found (in this case, numbers 001 to 142) and which files within this range (i.e., 1-142) were missing (in this case, 6 and 12). The number of IREDs found will also be displayed as well as progress.



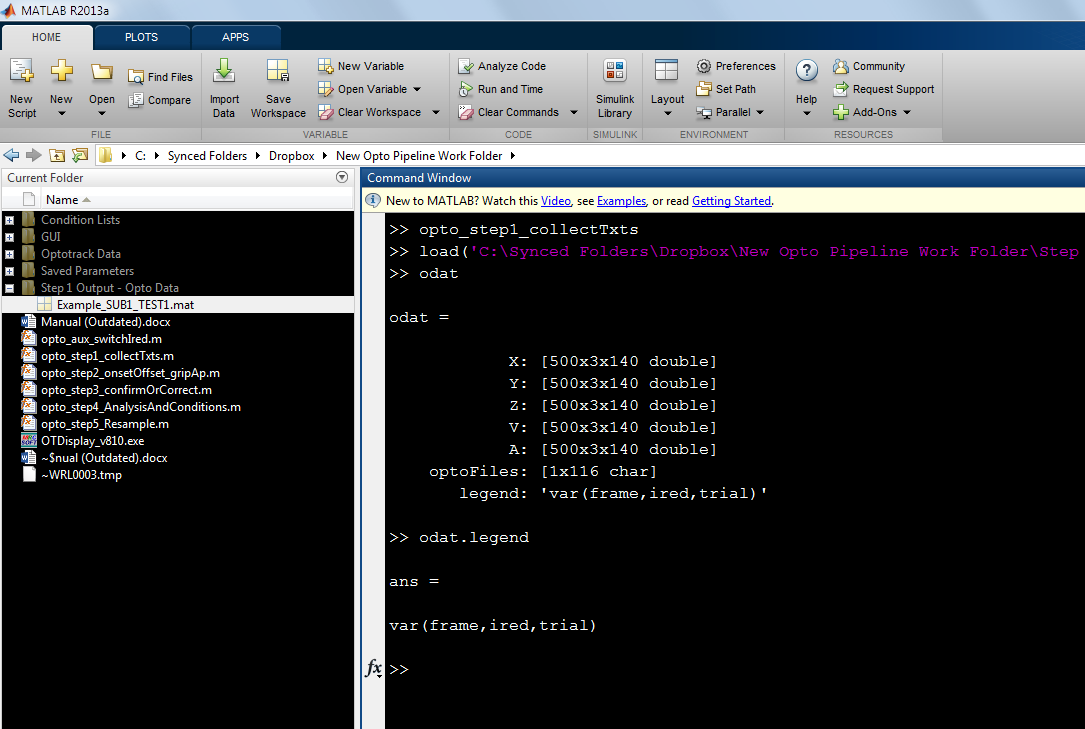
* 1. Once complete, the GUI will indicate completion as well as the filename of the output file. The folder containing the output file will also be opened to verify success. Note that the folder “Step 1 Output – Opto Data” will be created if it does not exist.



* 1. You should now have mat files that contain a data structure called odat.



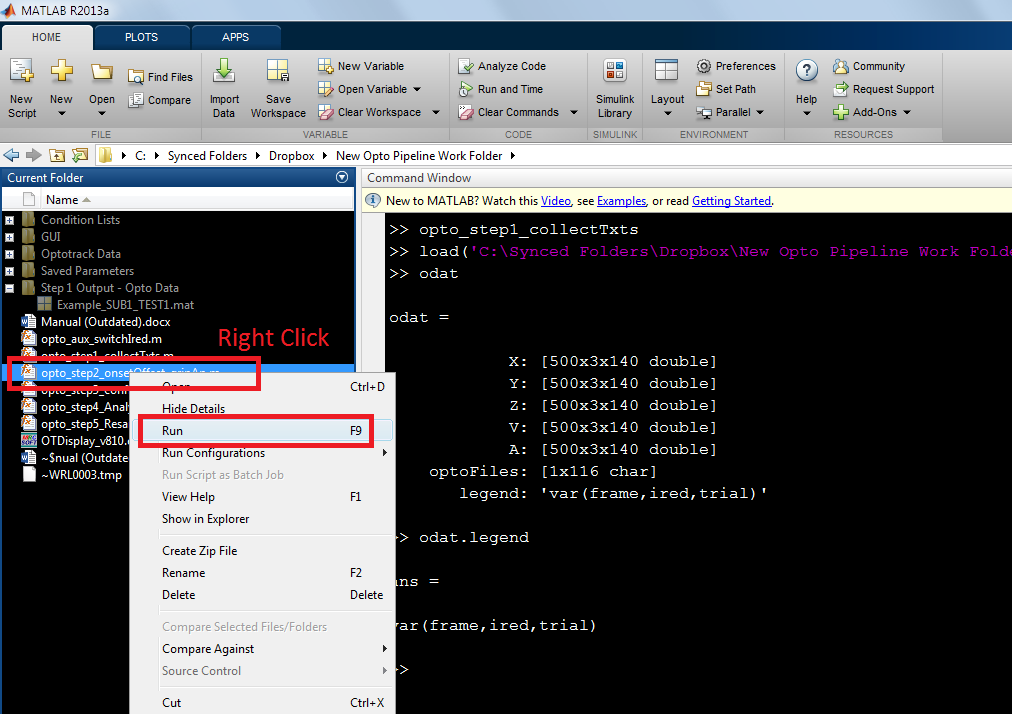
* 1. The structure, odat, contains XYZ as well as velocity (V) and acceleration (A) for each frame (500 here) for each ired (3 here) for each trial (140 here).

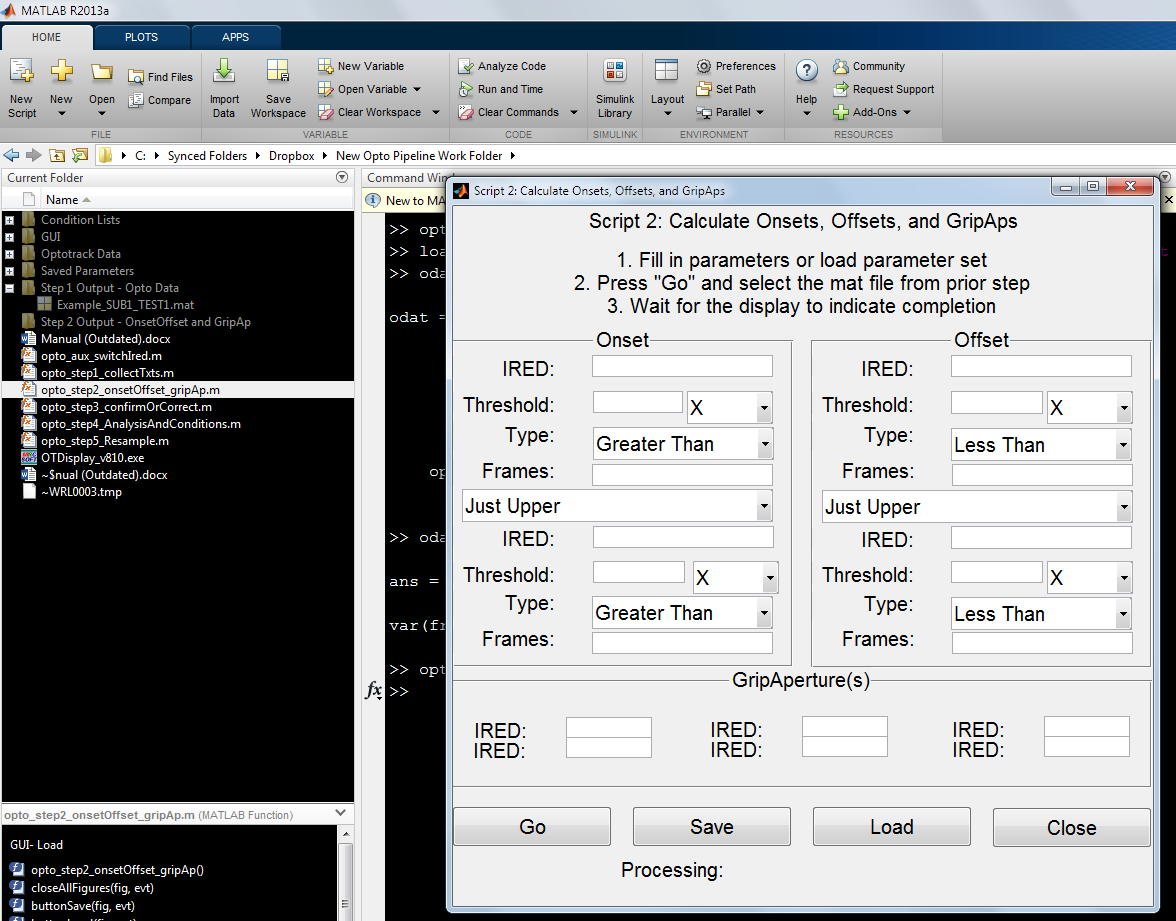


* 1. If IRED locations were accidentally switched between participants, run “opto\_aux\_switchIred”, select the mat file created by script 1, and enter the two IREDs you wish to switch. To switch more than 2 IREDs around, you can do pairwise switching.
     1. If IRED1 and IRED3 and switched, 1 becomes 3 and 3 becomes 1.
     2. A new mat file is created with “\_Switched#And#” added to the end. Use this file in script 2.

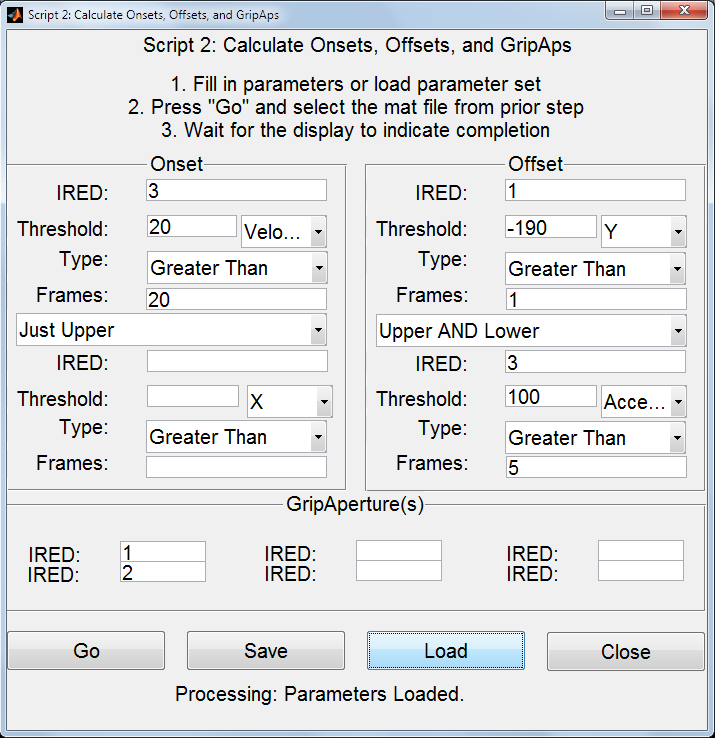
### Script 2: Automated onset/offset frame and grip aperture(s)

* 1. In matlab, run “opto\_step2\_onsetOffset\_gripAp”





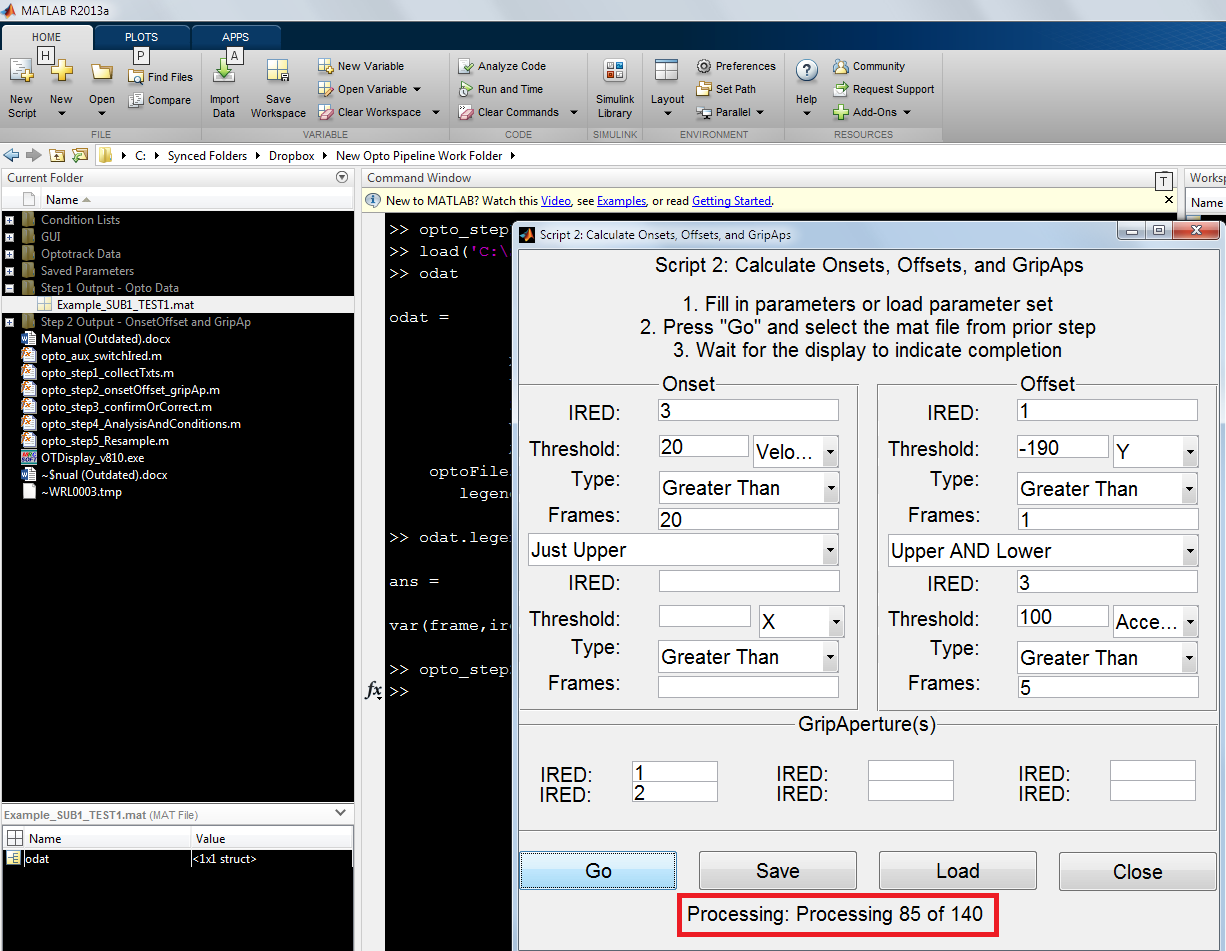
* 1. Fill in all parameters or “Load” pre-existing set. All text fields accept a single number as input. Do not include any characters or spaces.
     1. For each onset and offset, up to two criteria may be used.
     2. For the example parameter set below:
        1. **Onset:** velocity of 20+ for at least 20 consecutive frames in IRED #3
        2. **Offset:** BOTH A) Y location greater than -190 in IRED 1 for at least 1 frame (indicates reach in progress) AND B) acceleration greater than 100 for at least 5 consecutive frames in IRED #3
           1. **NOTE:** this offset parameter set is abnormal. Typically, stopping at a velocity of <20 in the wrist for at least 20 frames is sufficient. However, if the wrist does not come to a complete stop, you will have to get creative with the offset parameters.
        3. **Grip Aperture:** distances between both IREDs 1&2 will be calculated (here this means thumb-index). Up to three IRED pairs can be entered.



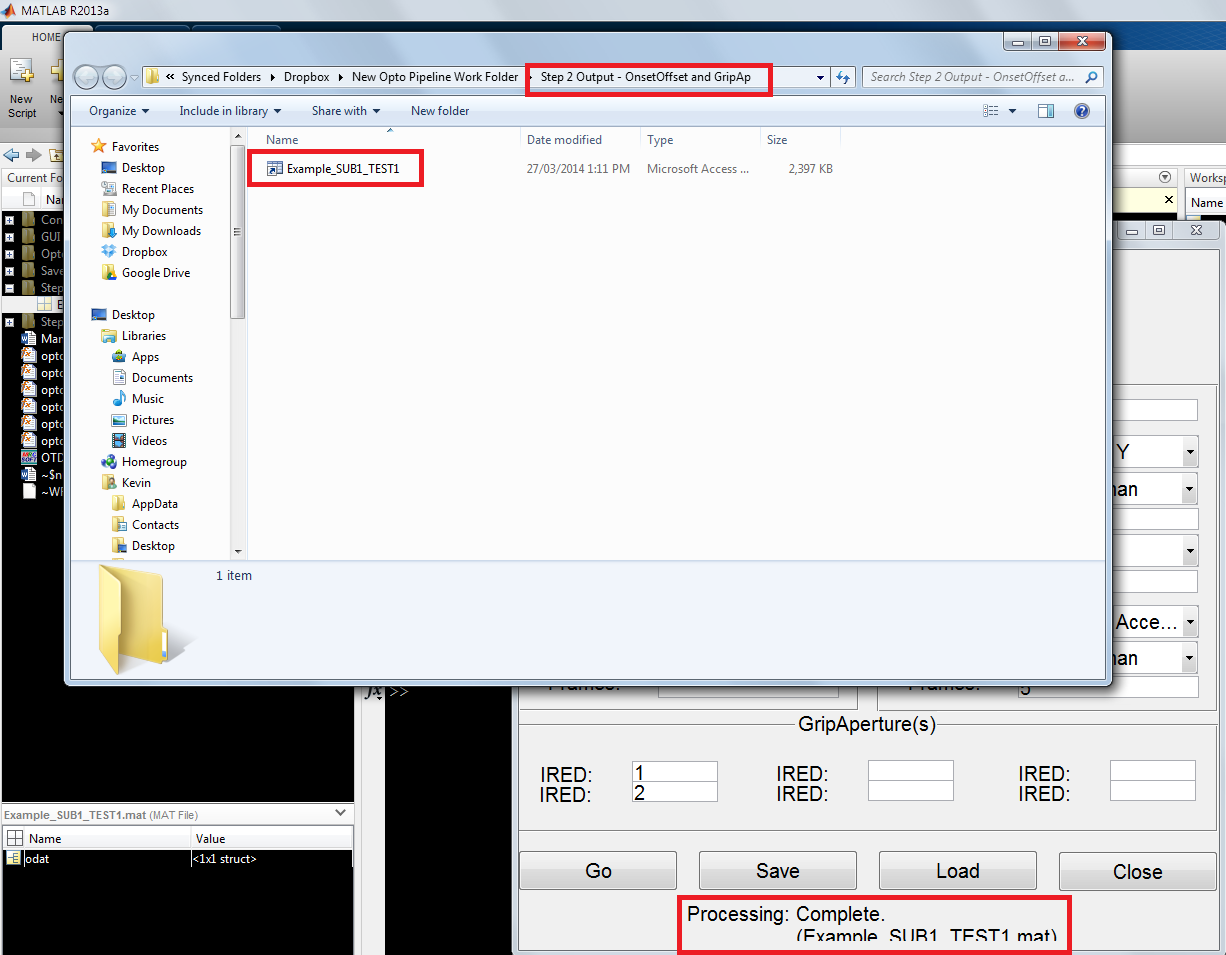
* 1. ADDED NOV28 2014: There is now a checkbox for each criteria which allows you to use the absolute value of a measure. For example, you could use the absolute X-Velocity (instead of positive/negative X-Velocity) by checking the “ABS” checkbox. The checkbox is only visible when a measure that can absolute is selected (e.g., checkbox is not visible when “Velocity” is selected because velocity is never negative). ABS is also not available for XYZ as these are positions.
  2. Once parameters are entered, you may save the set if you intend to use it later. Save parameter sets will be placed in a folder called “Saved Parameters”.
  3. Press Go and select the mat file from step1 that you wish to process.



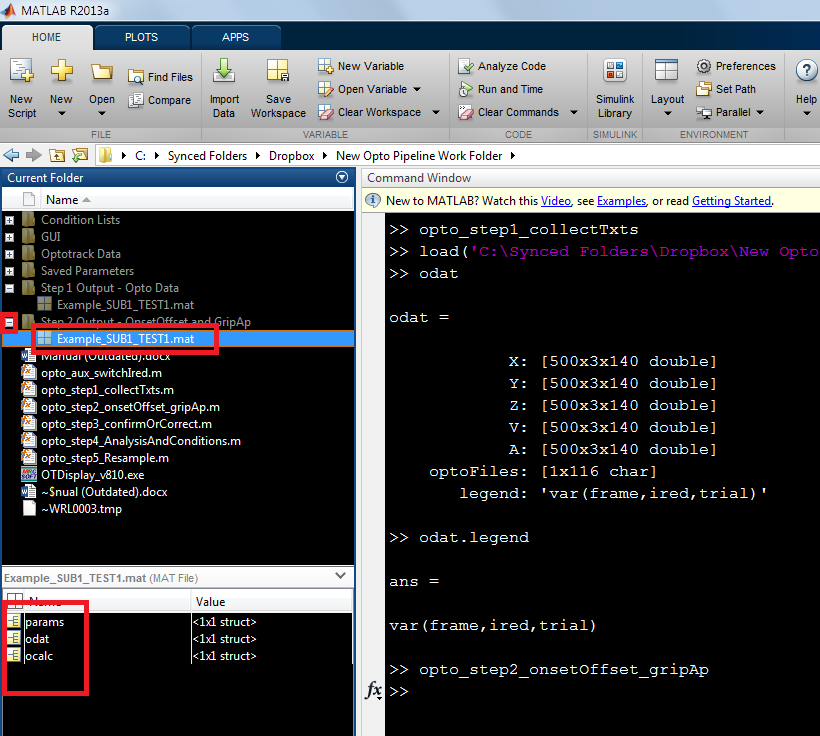
* 1. Progress is indicated at the bottom of the GUI.



* 1. Similar to step1, the GUI will indicate when processing is complete and the resulting file will be showing in explorer to verify. This output is saved to a folder called “Step 2 Output - OnsetOffset and GripAp”.

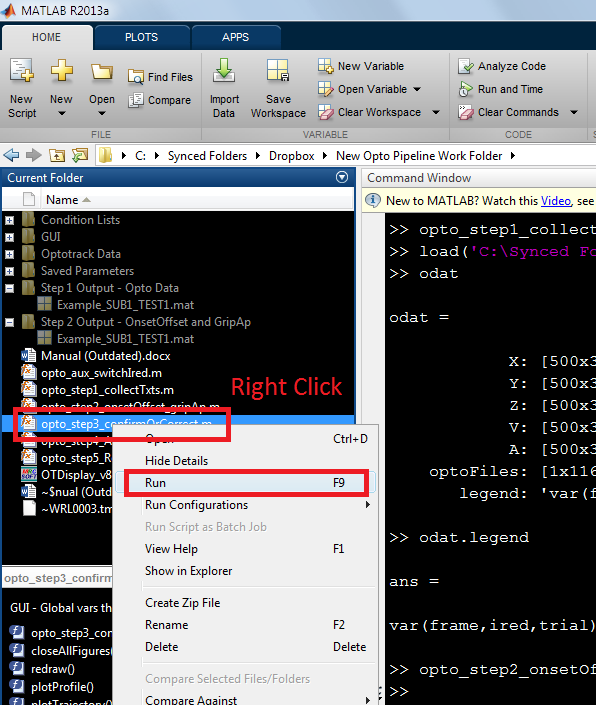


* 1. The output mat file contains a copy of the odat that was used as input (result of step 1). The file also contains the parameters entered in a structure called params. The onsets, offsets, and grip aperture(s) are stored in the structure called ocalc.

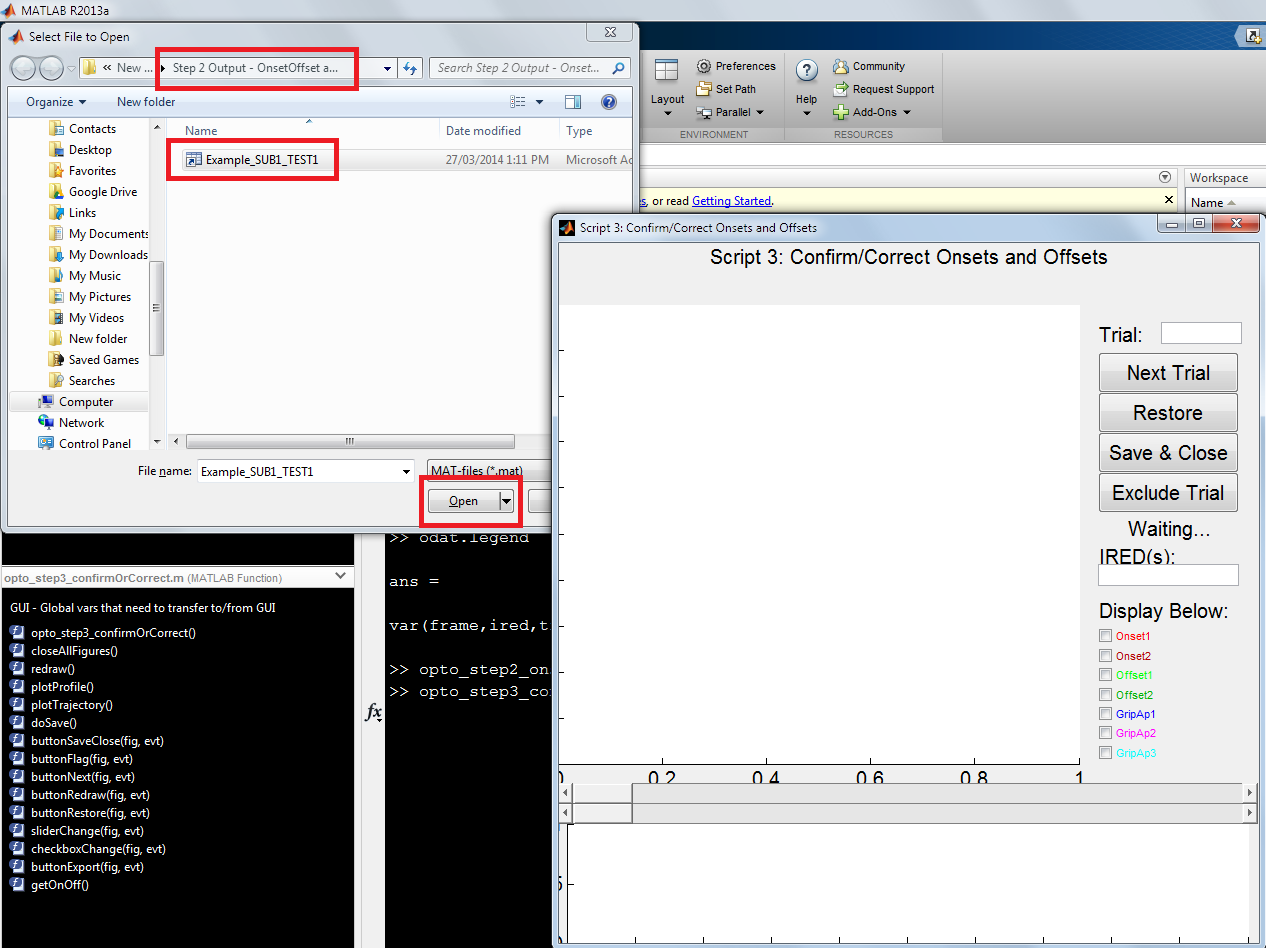


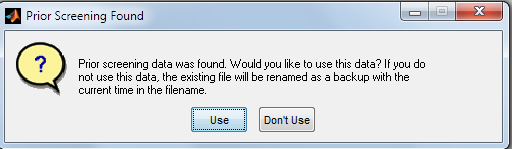
### Script 3: Manual Checking/Correcting and Visualization

* 1. In matlab, run “opto\_step3\_confirmOrCorrect”

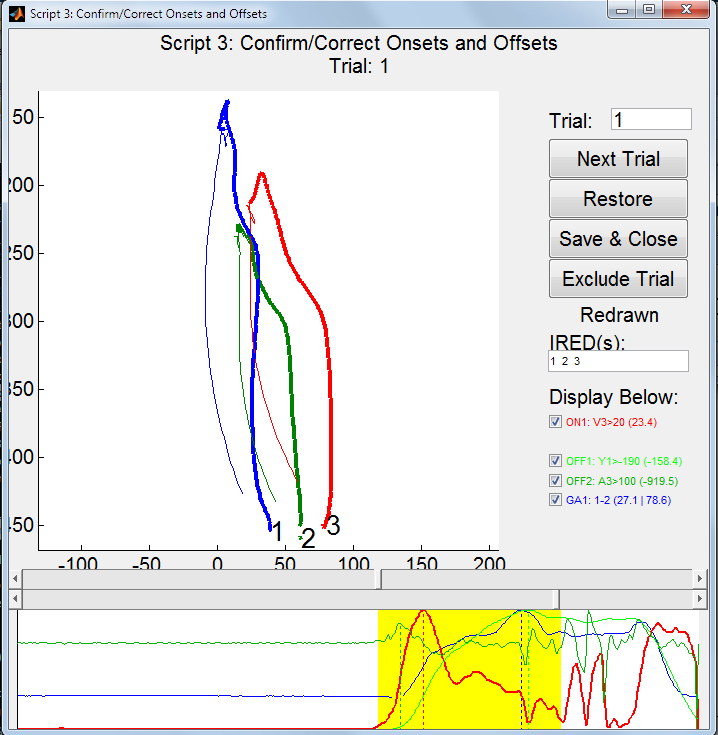


* 1. Select the mat file output from script 2. Note that you will have to rerun this script for subsequent data sets (i.e., there is no “Load” button). This is to ensure that global variables do not become mixed-up.
     1. If prior onset/offset changes have been made using this script, a popup will ask you if you wish to load those changes for this session. If no, the old changes will be backed-up.





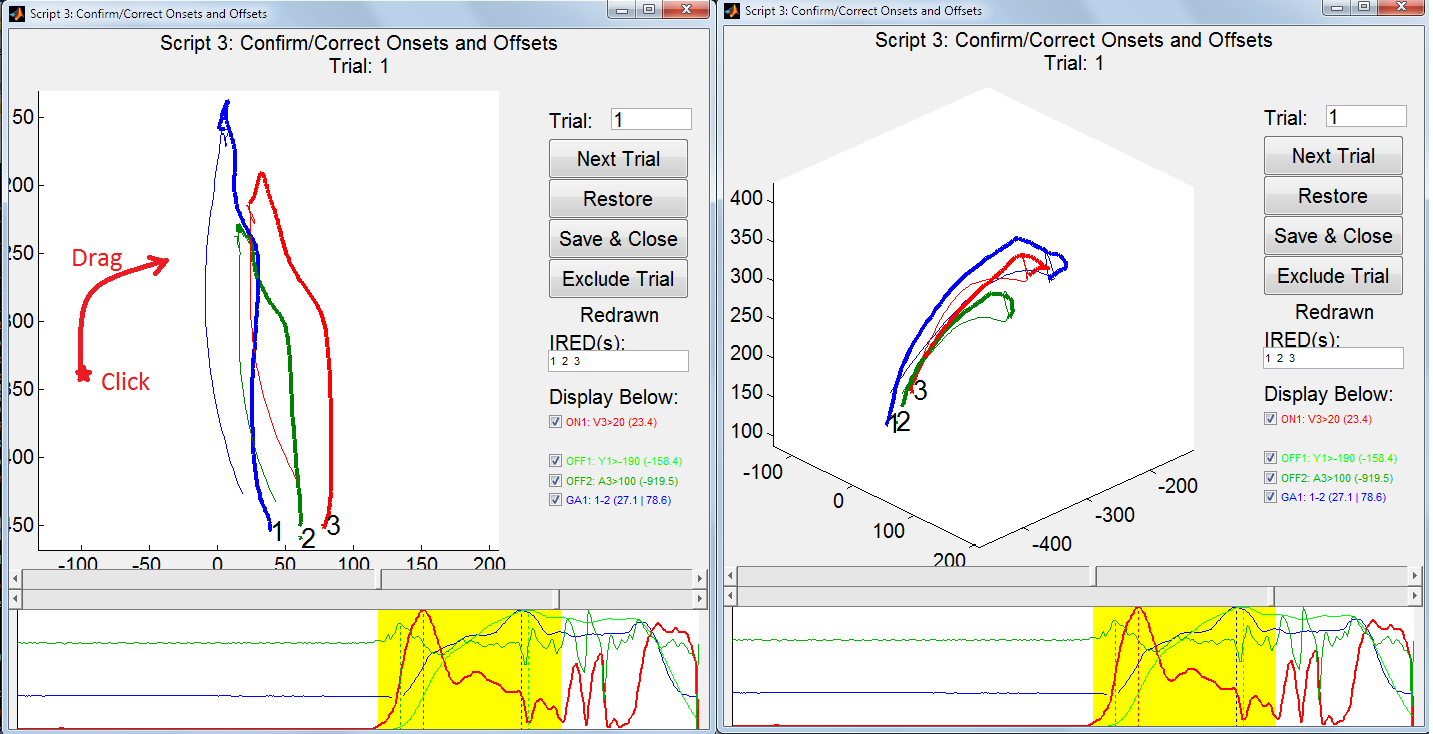
* 1. Starting with trial 1, the trajectory and profile are displayed.
     1. Any interface items that are not present in the dataset will be automatically removed. For example, this dataset did not contain a second onset criteria or second/third grip aperture pair so these checkboxes are removed.



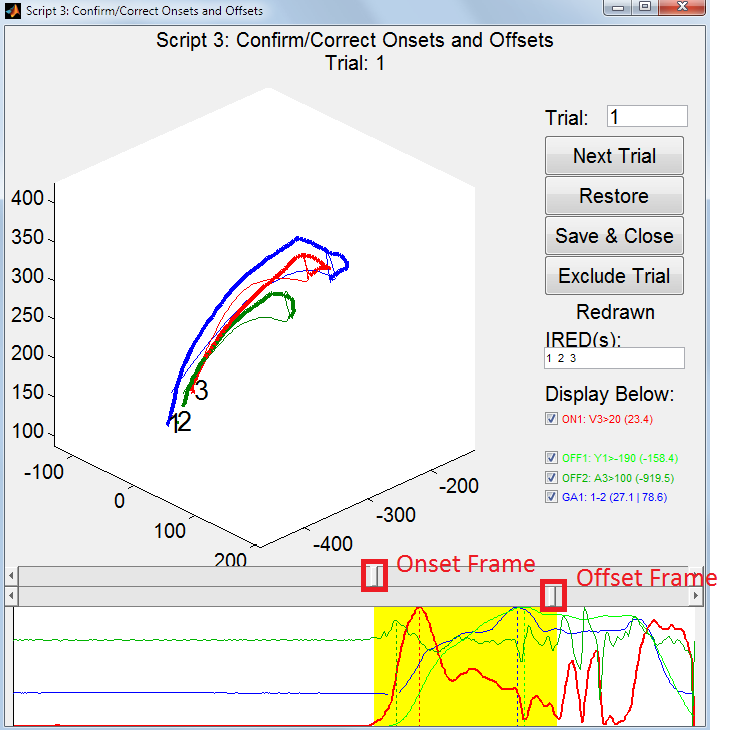
* 1. Understanding The Interface: Trajectory Plot
     1. The upper plot indicates the actual 3D trajectory of the trial**. Bolded segments of each line indicate frames from the selected onset to offset**. Each IRED is indicated with it’s number and displayed in a unique colour (**these colours are arbitrary**, unlike colours in the lower plot).



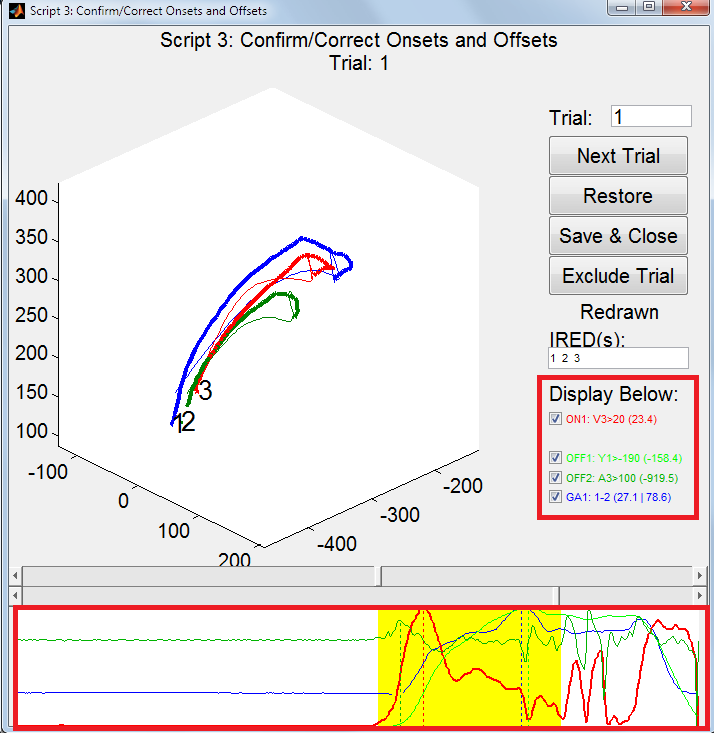
* + 1. You may select which IREDs to display using the indicated (above) textbox.
       1. If there are many IREDs, you can enter them as:
          1. “1:5 9:13” (shows IRED 1 2 3 4 5 9 10 11 12 13)
       2. Press enter or click elsewhere to apply changes.
    2. Click and drag the axes to rotate the trajectory in 3D space (below).



* 1. Understanding The Interface: Onset/OffSet Sliders
     1. The frames of onset and offset are indicated by the slider bars circled below. The leftmost position is frame #1 and the rightmost position is the final frame.
     2. **Changes to these sliders (dragging/clicking to change their positions) will change the recorded onset/offset frame.** These changes will be reflected in all interface elements. Press “Restore” to undo manual changes to the current trial.



* 1. Understanding The Interface: Profile Plot and Checkboxes
     1. Checkboxes indicate which items will be displayed on the profile plot (lower)
     2. The names of each checkbox correspond with the criteria specified in Step 2
     3. In parenthesis,
        1. For onsets (ON1 and ON2), the value of the criteria (e.g., V3 – velocity in IRED 3) at the current onset frame is displayed.
        2. For offsets (OFF1 andOFF2), the value of the criteria at the current offset frame is displayed.
        3. For grip apertures (GA1, GA2, and GA3), the value at both onset and offset is displayed separated by a line (“|”).



* + 1. The checkbox colours are the same colours used for each item in the profile plot
       1. For example, GripAp1 is blue in both the checkbox’s font and in the plot.
    2. The yellow section of the profile plot indicates the onset/offset of motion as defined by the slider bars above.
    3. The left/right scale of the profile plot matches that of the sliders above (first frame to last frame)
    4. In recent versions, a vertical indicator (dotted line) is presented for each plot line indicating the maximum value within the motion frames (yellow).
    5. The y-axis is 0-to-max individually for each plot.
    6. A break in a line indicates that a requisite IRED was blocked during the missing frames.
  1. Understanding The Interface: Buttons and Trial Box

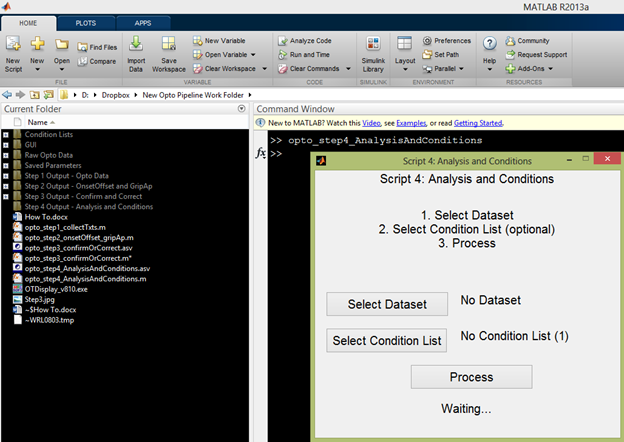


* + 1. The current trial is indicated under the title.
    2. “Next Trial” will save the current trial and move to the next unless there is no next trial.
       1. To change the present trial to a non-sequential trial, enter a new value in the text box and press enter.
    3. Pressing “Restore” will undue all manual changes to the *present trial*.
    4. “Exclude Trial” will flag a trial so that it will not be used in later steps. Additionally, the background will change to red when a flagged trial is displayed.
       1. Note: this button is new, not reflected in prior screenshots
    5. When you are done, press “Save & Close” to finish.
  1. Saving creates a new mat file in a subfolder called “Step 3 Output - Confirm and Correct”, which is created if it does not exist. The save filename is the same as the loaded file.

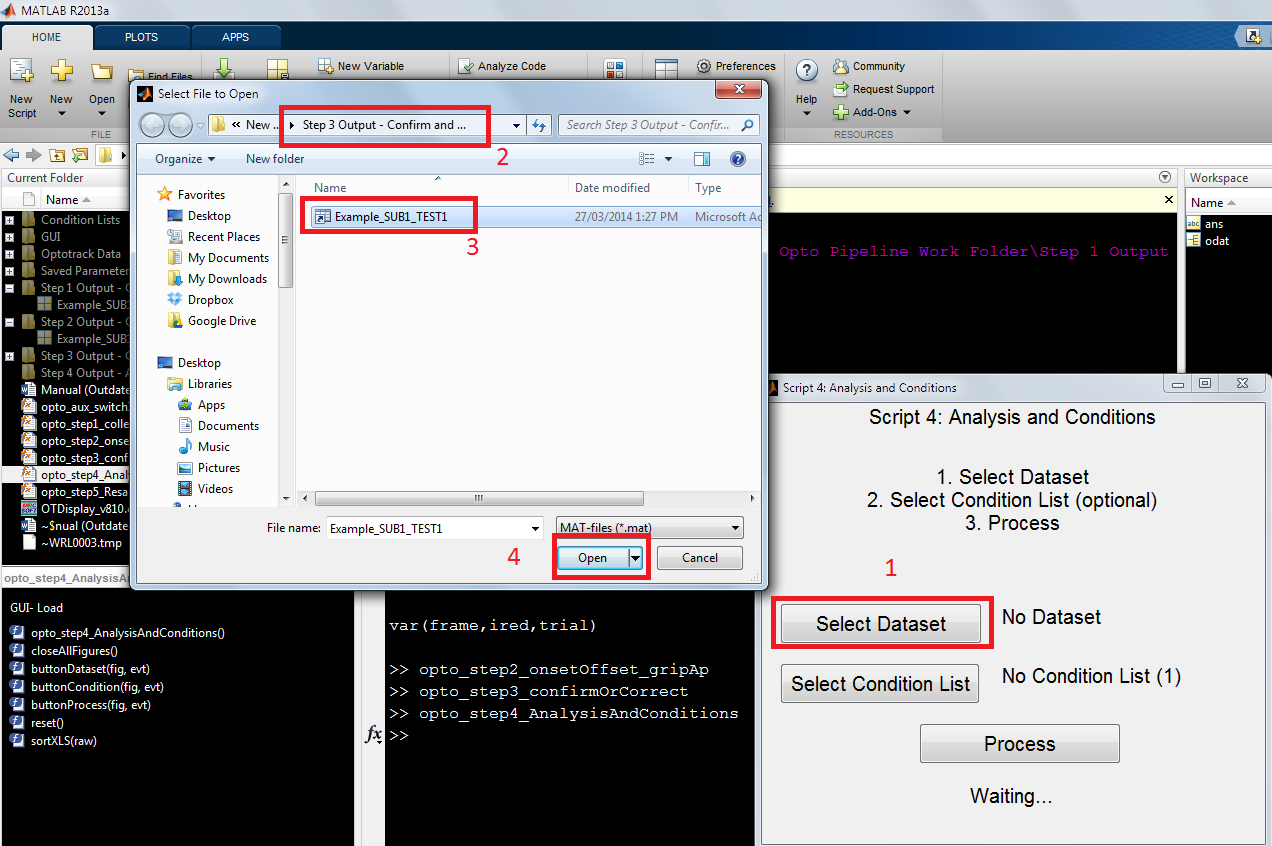
### Script 4: Analysis and Conditions

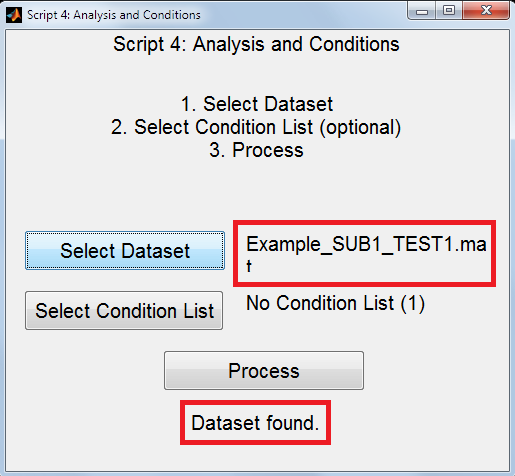
* 1. In matlab, run “opto\_step4\_AnalysisAndConditions”



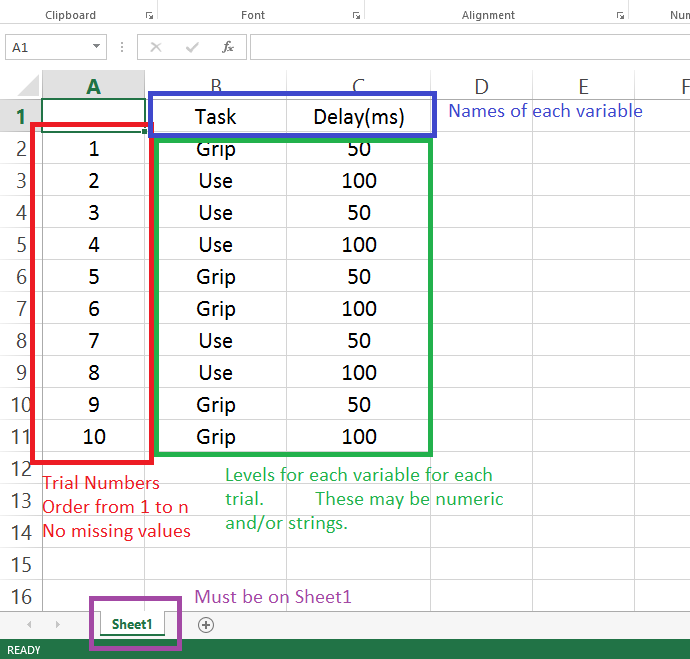


* 1. Press “Select Dataset” and choose the output file from the step 3 script.

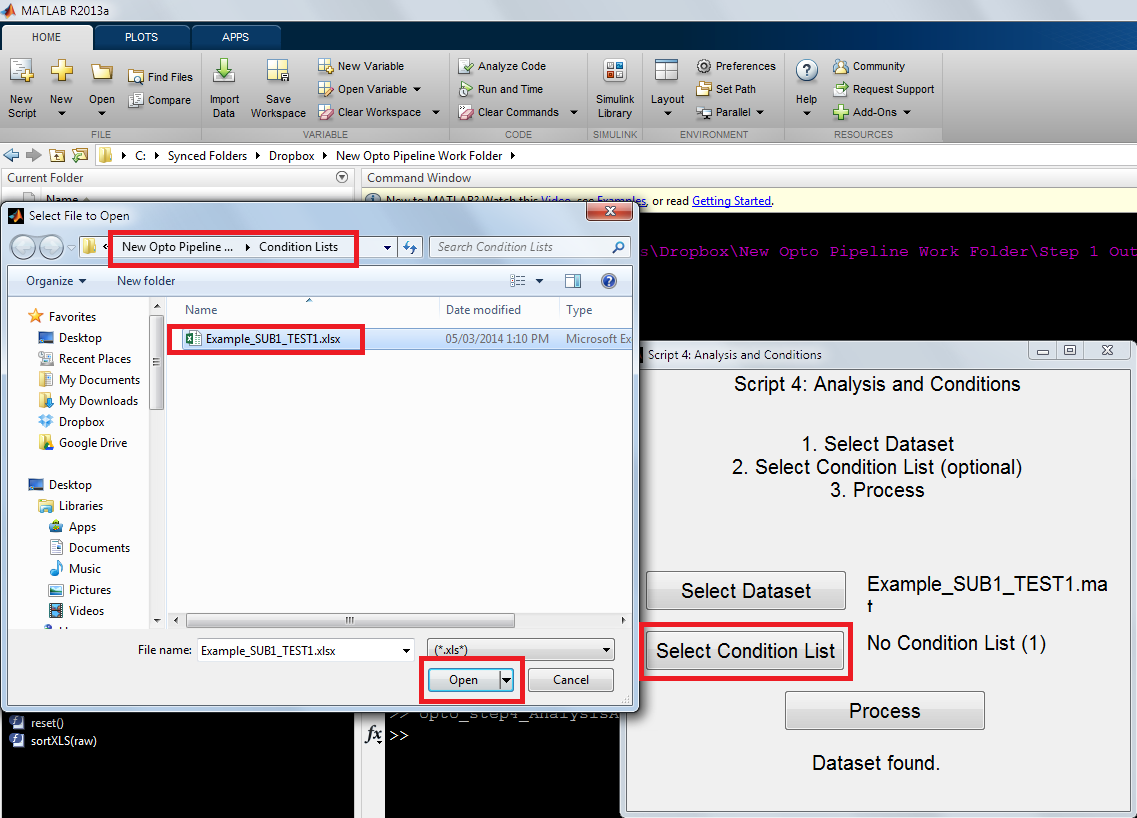




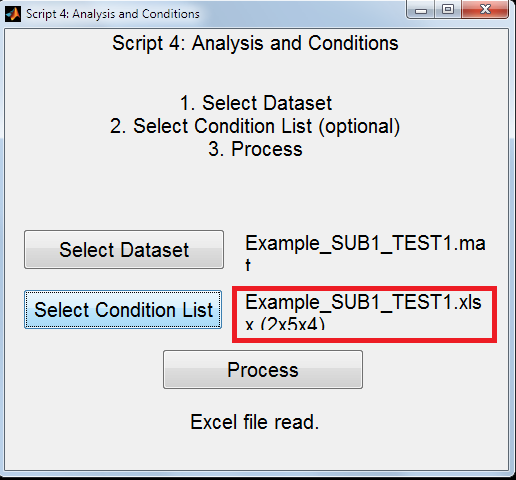
* 1. At this point, condition information may be provided. If no condition information is provided, all trials will be assigned to a condition called “AllConditions”.
     1. To include condition information, you must first create an excel file to be placed in a local folder called “Condition Lists”. This folder is automatically created the first time you run this script if it does not yet exist.
     2. Files may be of .xls or .xlsx filetype.
     3. Files **must** be of the following format:



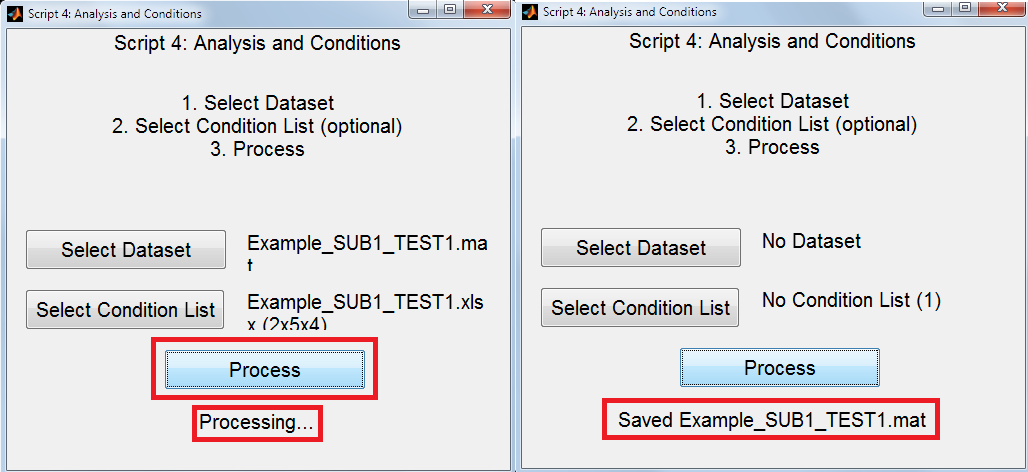
* + 1. There is no limit to the number of variables or to the number of levels each variable may have (i.e., a 2x2 is shown above, but anything is allowed).
    2. Variable and level names may be made up of any characters that can be used in a filename (i.e., no colons, question marks, etc) including numbers.
    3. Press “Select Condition List” and choose the excel file.



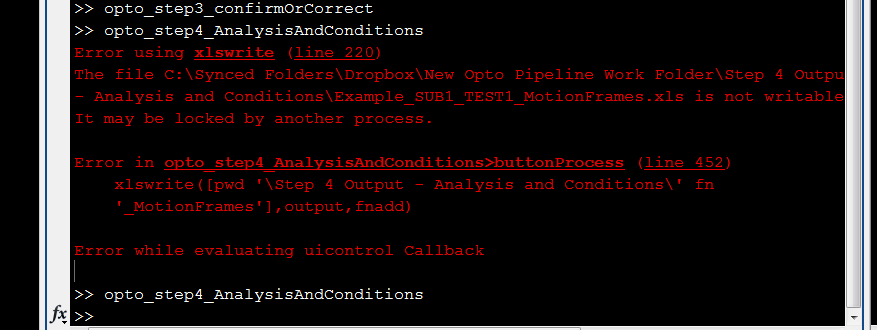
* + 1. Ensure that the correct filename is now displayed beside the button. Additionally, the number of levels of each variable are indicated in parenthesis. In this example, the first variable has 2 levels, the second variable has 5 levels, and the third variable has 4 levels.



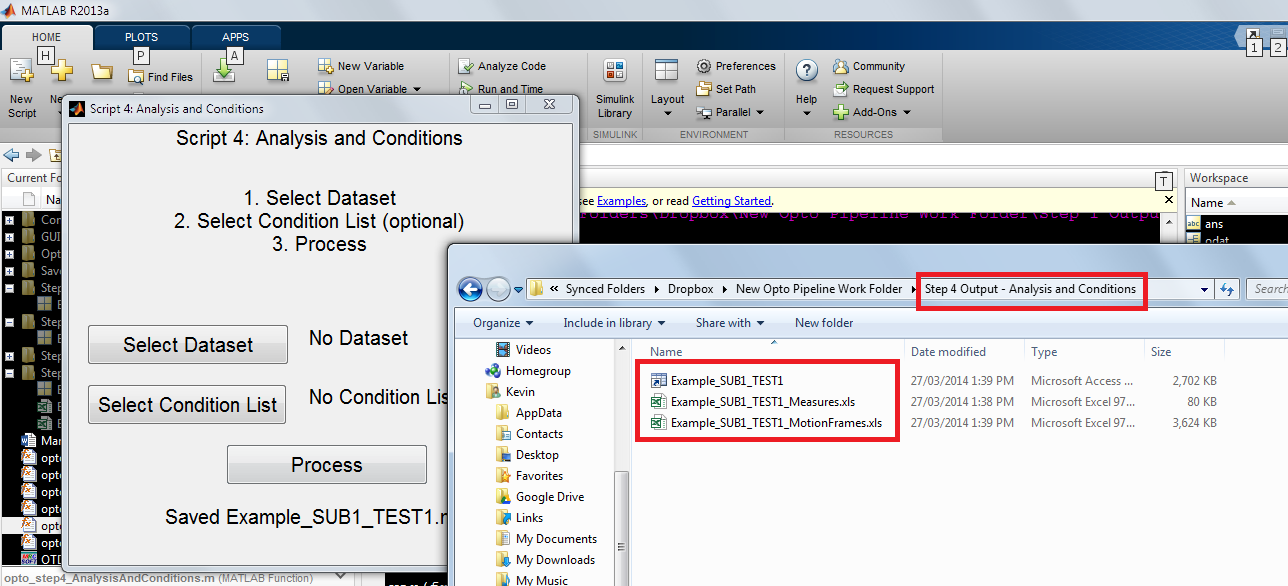
* 1. Press “Process”. This process can be lengthy (up to a few minutes?) if there is a lot of data or many variables/levels.



* + 1. If there is an error like the one shown below, try again. Sometimes the actxserver for Excel has trouble and needs a do-over. This error may be more common on older machines.



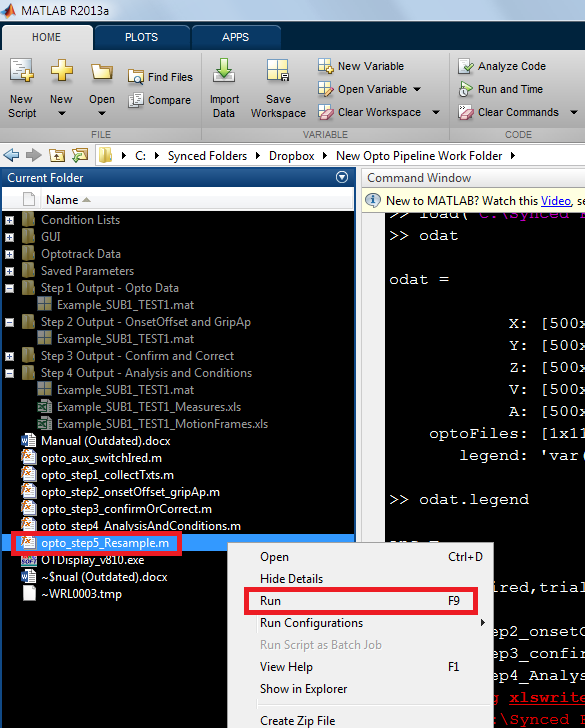
* 1. A new mat file is saved in a subfolder called “Step 4 Output - Analysis and Conditions”, which is created if it does not exist. The save filename is the same as the loaded file.

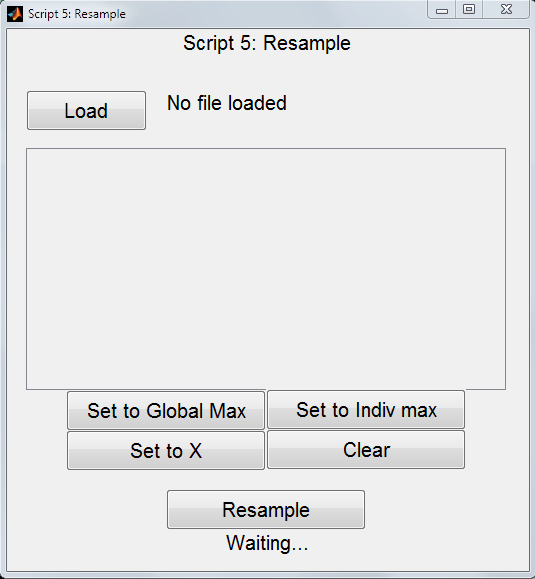


* 1. In addition to the mat file (used for script 5), excel files are created. The measures file contains all measures such as peak velocities for each IRED. The motion frames file contains XYZ and grip aperture(s) for each trial during its motion frames. **In both files, conditions are placed on separate tabs.**
  2. Statistics may be done using the mat file instead of the excel files if desired. The main structure is “data.cond(#).trial(#)”.

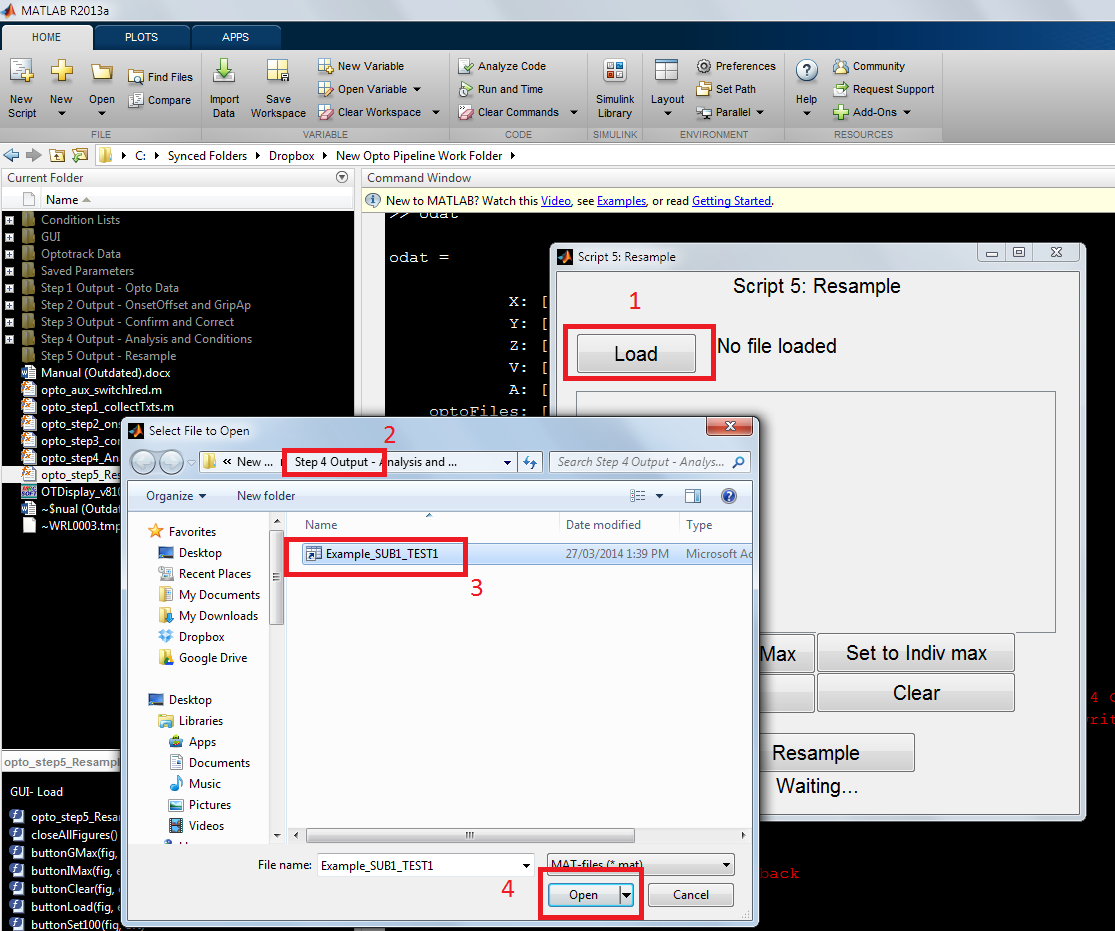
### Script 5: Resample Motion Frames (optional)

* 1. In matlab, run “opto\_step5\_Resample”

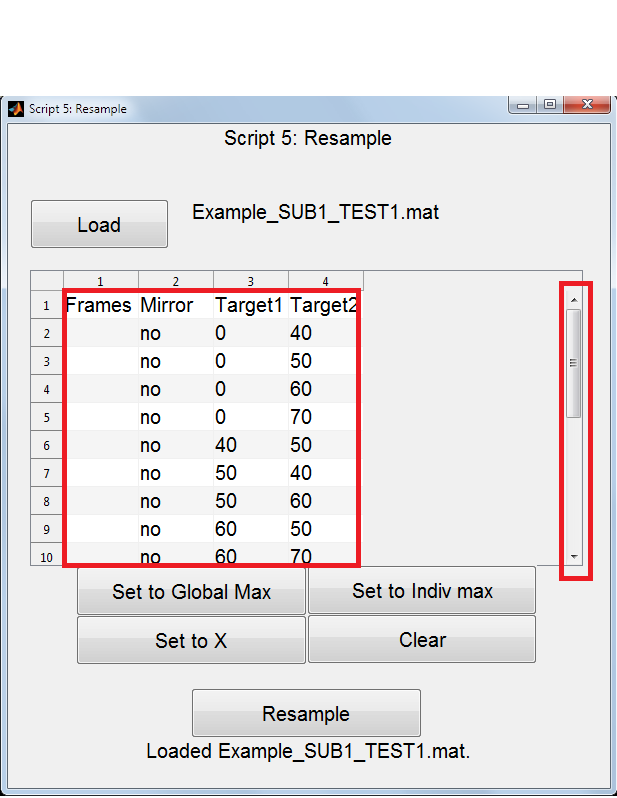




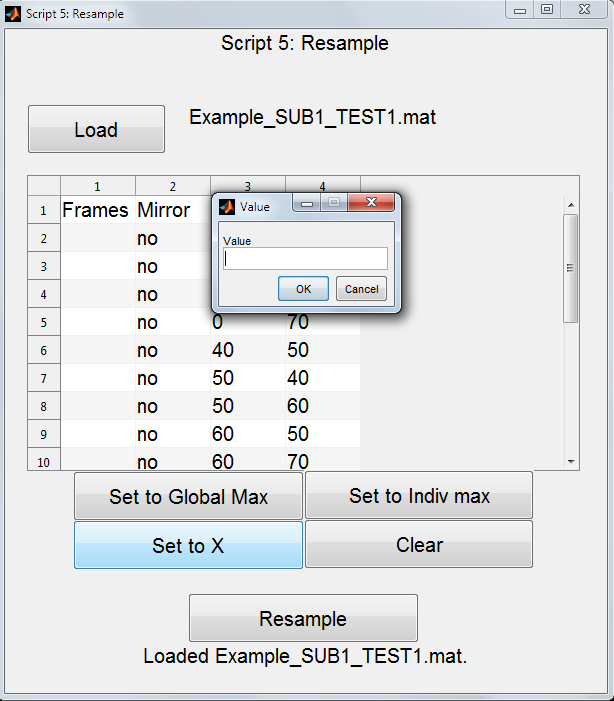
* 1. Click “Load” and select the mat file from the previous script.



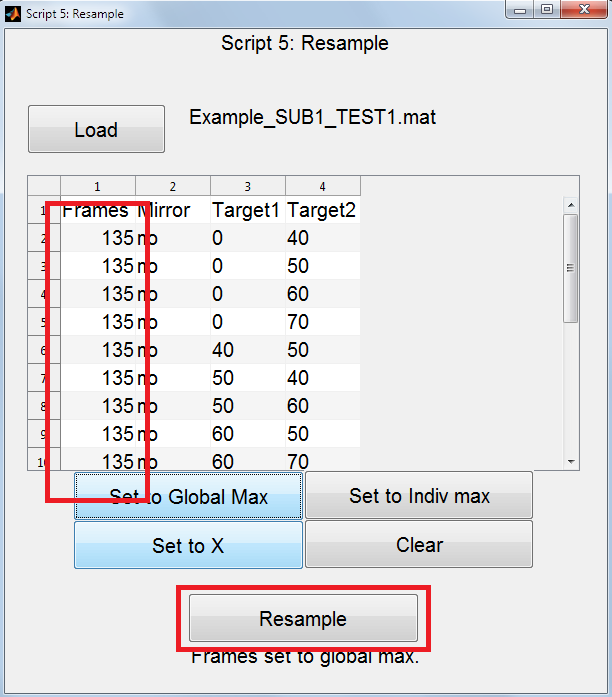
* 1. All conditions present in the excel file used in script 4 will be read into a table.
     1. In this example, the excel file had 2x5x4 levels, but only 20 of the possible 40 conditions were used. As such, only those 20 conditions (not all 40) are displayed in script 5.
     2. Note that there is a scroll bar.



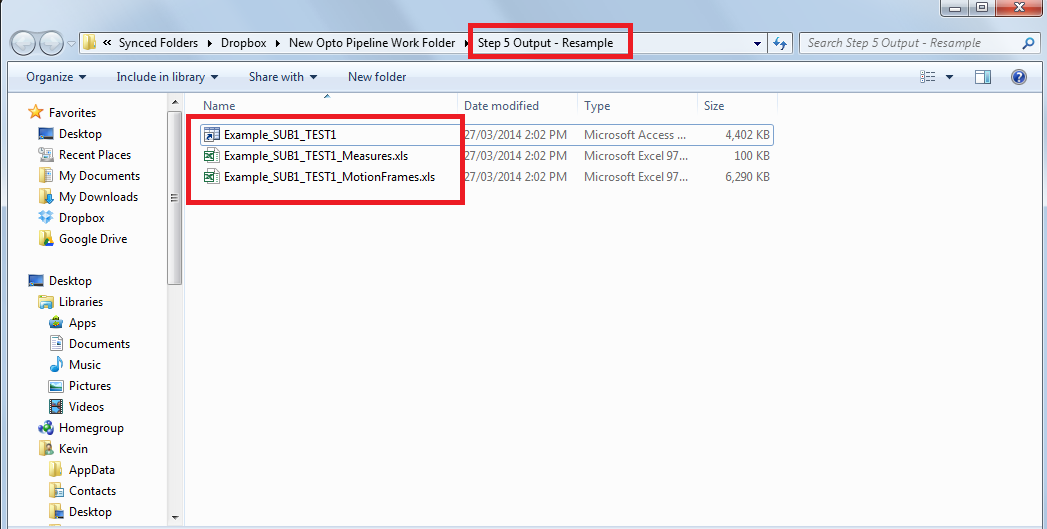
* 1. For each condition, you may now choose how many frames to resample the motion frames too. It is recommend that you not downsample unless you have good reason to.
  2. You can enter resample frame numbers in one of 4 ways:
     1. **Manually:** for each condition, you can write in a number of frames to resample all trials of that condition to
     2. **“Set to Global Max”**: clicking this button will determine the longest motion span (largest number of motion frames) amongst all trials and enter this value for all conditions
     3. **“Set to Indiv Max”:** click this button will determine the longest motion span within each condition and set the values accordingly
     4. **“Set to X”:**  you will be prompted (see image below) to enter a value which will be used for all conditions – this can be useful for resampling to a round number like 100 or 1000
     5. The clear button clears all selections.



* 1. Once all conditions have a value to resample to, press Resample.
     1. This step can be time consuming depending on the amount of data and resample values (potentially a few minutes longer than script 4).
     2. If an excel error like the one mentioned for script 4 occurs, try again.



* 1. The excel output is the same as for script 4 except that some measures are now given in percentages instead of frame numbers.



* 1. Additionally, a final mat file is provided that may be used if you wish to do your statistics in matlab instead of excel. This file contains a structure called “data” with many sub-structures. “data.data.cond(#).trial(#)” is where most final data is stored.

### How to use input your own motion onset/offset frames

* 1. If you already know the onset/offset times, you can bypass the automated onset/offset calculation in Step 2. This will require that you know the framerate that you recorded Optotrak at (could be anywhere from 100hz to 2000hz or more).
  2. First, convert your onset/offset times into frames using the known framerate.
     1. For example, 1.768 seconds at 1500hz is frame 2652.
        1. Matlab in-line: onset\_frame = round(onset\_time \* framerate);
     2. Remember to round your frames so that you do not try to reference non-integer frames.
  3. Second, run step 2. Enter the IREDs for grip apertures. Enter anything at all in the onset/offset criteria.
     1. These onsets/offsets will be overwritten.
  4. Third, load the output file from step 2.
  5. Fourth, change the onset frames located in “ocalc.onset.onsetFrame(trial#)” and the offset frames located in “ocalc.offset.offsetFrame(trial#)”.
  6. Fifth, save the mat file. You can change the filename from the original if you like.
  7. Sixth, continue with step 3 using the file you just saved.
  8. The lines in the graph at the bottom of step 3 will be based on the overwritten criteria, but the actual onset/offset frames will be the ones you entered.

### Auxiliary Script: Endpoint Locations (Work In Progress)

* 1. I am currently working on a script which uses output files from this pipeline to determine endpoint locations in another plane at motion offset (e.g., the plane of a targeted object). We have scripts that can do this already, but they have been designed on a project-by-project basis. This script will be general enough to apply to nearly any project.
  2. The script requires either:
     1. 3 IREDs which define the target plane, OR
     2. 1+ IRED(s) which moves within the target plane