* 1. **Read in Data**
     1. final\_GetPlexData.m
        1. for a .pl2 file, takes 5-10" and reduces from 900mb -> 50mb (~18x)
        2. for a .plx file, takes 120-130" and reduces from 950mb -> 45mb (~21x)
        3. Uses Plexon's Matlab Offline Files SDK to load data from .plx or .pl2 files
        4. Gives a warning for files with multiple fragments
     2. final\_RearrangeData.m
        1. Takes 15-25" per frequency per region per mouse
        2. Re-arranges the data across all recordings into single files where each row is a 60" trial of 20"-

off, 20"-on, 20"-off

* 1. **Check Data Quality - These automated detection scripts are not fool-proof, so always visually inspect potential problems**
     1. final\_CheckLFPtraces.m
        1. Creates and saves plots of the original recording traces, with the trials, SLOs, and clipping annotated
        2. Useful for checking bad channels, trials, clipping, etc.
        3. Also saves a preliminary \*\_countSLOs\_prelim.mat file
        4. Saves the figures
     2. final\_RunQC1.m
        1. Operates on the individual recording files (\_A,B,C,D)
        2. Checks the sum of the LFP mV values to test for duplicate channels
        3. Checks the sum of the Spike TS values to test for duplicate units
        4. Outputs results to console and as a QC1\_\*.txt file
     3. final\_RunQC2.m
        1. Operates on the \_adj files
        2. Compares laser timing to expected to find deviations
        3. Compares LFP duration to expected to check for drops in data (e.g. fragments)
        4. Checks for clipping in the LFP
        5. Outputs results to console and as a QC2\_\*.txt file
     4. final\_CheckSpikesDuringClips
        1. Check for clipping and whether spikes occur during them, and if so how many
     5. final\_CheckWaveforms.m
        1. Plot the spike unit waveforms from all frequencies and recordings and manually inspect for bad units
        2. Saves the figures
        3. Can also be used to re-check specific bad waveforms and just save those unit names

**After you decide which channels, trials, and units to drop, you can use the following script to make the \*\_filt.mat version of the file for all subsequent analysis**

* + 1. final\_FilterData.m
       1. Manually define badChannels, badTrials, & badUnits
       2. Drop the bad channels, trials, and units
       3. Also re-orients the Poly2 electrode sites so channel 1 is at the tip (they have a reversed site map)
       4. Save the new file as \*\_filt.mat
  1. **Analysis & Plots - LFP**
     1. final\_PlotLFPspectrograms\_byTrial.m
        1. Plots the LFP for individual LFP channels & trials (generally before filtering)
        2. Saves the plots
     2. final\_PlotLFPspectrograms\_Averaged.m
        1. Plots the LFP for individual LFP channels, averaged across trials
        2. Saves the plots
     3. final\_PlotLFPaverages\_bySite
        1. Plot the LFP traces averaged across trials by electrode site
        2. Saves the plots
     4. final\_GetLFPaverages\_byGroup.m
        1. Gets the average LFP trace for all animals
        2. Saves the data structure to LFPaverages.mat
     5. final\_PlotLFPspectrograms\_byGroup.m
        1. Plots the LFP for individual LFP channels, averaged across animals in a group
        2. Can use either the LFP trace averages, or preferably the PSD averages
        3. Saves the plots
     6. final\_GetLFPamplitude\_byGroup.m
        1. Extracts the average LFP amplitude for pre-, peri-, and post-stimulation after passband filtering the stimulation frequency
        2. Also extracts the unfiltered amplitudes
        3. Saves the data structure to LFPamplitude.mat
     7. final\_PlotLFPamplitude\_byGroup.m
        1. Plots the increase in amplitude by group, region, frequency, and channel
        2. Saves the plots
     8. final\_PlotLFPamplitude\_byMouse.m
        1. Plots the averaged LFP amplitude increase by mouse with SEM by trial and the group medians and means
        2. Can also do it for both frequencies combined
        3. Saves the plots
     9. final\_GetLFPpeaks.m
        1. Slices out the first peak of the LFP trace and averages across trials
        2. Saves the data structure and the range used to LFPpeaks.mat
     10. final\_PlotLFPpeaks.m
         1. Generates a plot for each channel and frequency showing the first LFP peaks for all groups and regions
         2. Saves the plots
     11. final\_PlotLFPpeaks\_Zoomed.m
         1. Generates a plot for each channel and frequency showing the first LFP peaks for all groups and regions, but zoomed in on the first peak
         2. Saves the plots
     12. final\_GetLFPdelays.m
         1. Counts delay between onset of stimulation and first LFP peak
         2. Can set a sensitivity (e.g. 80% of the first peak) since some have multiple peaks and the first may not be the biggest
         3. Saves a \*\_LFPdelays.mat output file with the results
     13. final\_PlotLFPdelays.m
         1. Plots the LFP delays across all channels for all groups at the different thresholds tested
         2. Saves the plots
  2. **Analysis & Plots - Spiking**
     1. final\_TestSpikes.m
        1. Bin spiking data and test for significant increases/decreases
        2. Saves a \*\_spikeTest.mat output file with the p-threshold and bin size, as well as the following info for each unit
           1. Firing Rates
           2. Logical matrix for INC, DEC, and NC units
           3. Average, std, and SEM across trials by bin
           4. Average, std, and SEM across trials by period (pre, stim, post)
     2. final\_PlotRaster.m
        1. Plot the events for each trial on top, and the histogram on the bottom, with the average firing rate per period overlaid on top of the histogram
     3. final\_PlotFiringRates.m
        1. Plots the firing rates for the neurons separated by response type, as well as all together
        2. Saves the plots
     4. final\_PlotStackedBar.m
        1. Plot a stacked bar graph with different colors representing how many neurons responded for all groups
        2. Saves the figure
     5. final\_PlotStackedbar\_byChannel.m
        1. Plots a separate StackedBar figure for each channel separately
        2. Saves the figures
     6. final\_PlotSpikeCount\_byChannel.m
        1. Plots the number of each type of response by channel for the different groups
        2. Saves the figures
     7. final\_PlotSpikeTest.m
        1. Plot the firing rates during each period for all neurons split by region, colored by their response type. Different plots for each group and frequency
        2. Can also do it split by group instead
        3. Save the plots
     8. final\_PlotSpikeTest\_bySite.m
        1. Plot the firing rates during each period for all neurons separated by electrode site, colored by their response type
        2. Saves the plots
     9. final\_PlotSpikeTest\_byChannel.m
        1. Plot the firing rates during each period for all neurons separated by electrode site and colored by their response type, with all units from the same group on one plot
        2. Saves the plots
     10. final\_PlotSpikeTest\_Rates.m
         1. Plots histograms of the firing rates for different groups
         2. Saves the plots
     11. final\_PlotSpikeTest\_Rates\_Dist.m
         1. Plots distributions of the firing rate changes for the different groups
         2. Saves the plots
     12. final\_GetSpikeDelays.m
         1. Counts delay between onset of stimulation and first or all spikes per trial
         2. Saves a \*\_spikeDelays.mat output file with the results
     13. final\_PlotSpikeDelays.m
         1. Plots all of the spike delays as a histogram at the group/region level
         2. Saves the plots
     14. final\_PlotSpikeDelays\_bySite.m
         1. Plots the spike delays as a histogram for each channel, with the bars colored by the response type
         2. Saves the plots
     15. final\_PlotSpikeDelays\_byChannel
         1. Plots a separate set of SpikeDelays figures for each channel separately
         2. Saves the plots
     16. final\_SortWaveforms.m
         1. Sorts the figures from final\_CheckWaveforms.m into different subfolders for comparison
         2. Some units not sorting properly and ending up in a dummy folder. Some due to being dropped, other I'm unsure about