LFP Analysis

1. Map channels for each specific probe paradigm (e.g. Neuronexus vs Cambridge), to the representative amplifier channel number. Use this to write a probe mapping file in MATLAB
   1. CODE – util for LFP analysis
      1. intan\_to\_site\_map = probe\_site\_mapping(probe\_type)
      2. For Neuronexus H64LP 8x8 use ‘NN8x8’ as the probe\_type
      3. To create a caption for plots, use –
         1. naming\_convention; % this is a script to create a workspace variable for monopolar power lfps – when plotting from ‘monopolarpower’ files.
         2. naming\_conventinon\_diffs; % this is a script to create a workspace variable for the diffs – when plotting from ‘diffpower’ files.
2. Review Histology according to channel number to verify location of each site
   1. Include AP, ML, DV coordinates
   2. Verify atlas locations of each site (e.g. VM, VA, etc)
   3. Create an excel sheet that is MATLAB friendly (e.g. no notes). The notes are helpful to have though for future training purposes.
   4. Include a column that references the specific amplifier channel associated with the respective channels (mapped from 1 above)
      1. Note: Intan Amplifier channels are labelled 0-63 and the Intan records data from 0-63 but MATLAB reads in the data as 1-64. Keep this in mind when checking data.
3. Review amplifier.dat data using Neuroscope
   1. Create a file or tab within excel data sheets to specify good or bad channels (include each date as a separate column; channels might be good one day and bad the next)
      1. Artifacts like high or low voltage can be accommodated within the code itself
      2. Current ‘codes’ for individual good vs bad **lfp** channels are:
         1. 0 = good channel
         2. 1 = bad channel
         3. 2 = has some issues but also good spots, verify the data post analysis
      3. For the column header, the dates are highlighted to show whether the file could be analyzed for spikes (though by eye these spikes may be very small at best for NeuroNexus probes tested from 2020 to 2021 by JM)
         1. Green >80% of the file is good (minus a few bad individual channels etc)
         2. Yellow 50-80% of the file should be good. SPI cable was attached for at least half the file from the beginning of the recording.
         3. Red <50% of the file had solid SPI connection, the SPI cable came off before 30 min of recording, or the file started without a refresh of the software (thus had a lot of Intan issues with recording a line as solely that line without interference).
4. Analyze the differences between neighboring probe sites
   1. E.g. for a NeuroNexus 8x8 probe, analyze site 1 vs site 8 on shank 1. Include in the code specifications to NOT analyze the dorsal site from one shank and the ventral site from the next (e.g. data that happen to be functionally near each other in the ‘\*.mat’ files.)
   2. Use the intan\_to\_site\_map function to help with coding in the site mapping for the Neuronexus probe
   3. lfp\_NNsite\_diff = diff\_probe\_site\_mapping(lfp\_fname, probe\_type) can be used to assess the differences based on this site mapping.
   4. Verify the math for a few channels - LFP\_check\_diffs.m
5. Extract trials by type – at least to gather/sort trials by type for later analysis. Will be helpful information when trying to extract lfps by correct trial type, etc.
   1. Run the Choice\_task\_intan\_workflow to generate the trials structure
   2. Run getTrialEventParams.m to select for which trials match the type of event you are interested
      1. E.g. ‘correct go’
   3. Run extractTrials.m to get an index of the trials with the event type you’re interested (from getTrialEventParams.m in step b).
   4. Run periEventTrialTs.m for desired eventFieldnames (e.g. cueOn and centerIn).
      1. This file uses the trials structure from the Choice\_task\_intan\_workflow.
      2. Start with a tWindow of ~2 seconds. A tWindow of 1s is good to get data of interest from either the original (reorangized for NNsite) or the lfp\_diff (differentials), but need a little extra data around the tWindow to account for data processing issues.
      3. Use trIdx in the function execution to only pick out the trials of interest (from running extractTrials.m in step c)
         1. trialRanges = periEventTrialTs(trials(trIdx),[-2 2],eventFieldnames)
         2. The trials struct timestamps will show you the relevant eventFieldnames associated with that particular trial --- e.g. if selecting ‘correctgo’ from getTrialEventParams, the trIdx will show you which actual trials from the trials structure are ‘correctgo’. Within the trials structure, find that particular trial number and click on ‘timestamps’. This will give timestamps for that particular trial number including the relevant eventFieldnames. This can include: cueOn, centerIn, centerOut, sideIn, sideOut, etc.
            1. eventFieldnames = {‘cueOn’, ‘noseIn’};