MATLAB code to extract LFP

/hpc/comco/lopez.l/electrophysiology/OrganizeBehaviour\_for\_python.mat

* Code to extrcat the LFP and the organize behaviour and structure them to open them in MATLAB
* Reads the .txt file from work/comco/kilavik.b/Data\_VisuoMotorLaminar/Lists&Documentation/{monkey}\_NEWList2020\_BetaBands

Note : I could get the code running – for now only extracted two Tomy sessions but I modified it such that it will give the behavior and the LFP output in RAWData in the same name format at Laura uses.

I am only removing the elitrials as mentioned by Bjorg in the NewBetaBands List and not including the ones removed extra by Laura

Output Folder - /Volumes/work/comco/nandi.n/LFP\_timescales/RAWData/{monkey}/ LFPmat

LFP Unipolar extraction of power and sites

main.py – This function is made by Laura which requires an excel file to run caled ephydataset\_info.xlsx (I need to understand this and modify it accordingly )

changed it to main\_modified.py

included name of monkey in the function as an input since I modified the path functions later

4 paths to give – path of analog signal (HandEyeMovements from bjorg), path of the mat data (LFP and behaviour- saved by me in RAWData), output path (to store the unipolar power), and the excel file (to read the session names, probes etc) .

Also added the src/prepoc\_tools script which has all function used inside the main.py code

* Searches for the session name given by the function.
* Then searches for the bad channels present in either of the probes in that session
* Loads the behavious and behaviour\_analog as dict from mat files
* good\_trials\_mask = behaviour['OrigCorrTrials'] – 1 – why ??? also, since I am using only the elitrials by Bjorg in the behaviour file and not including Laura’s, do I need to change somehow the excel file so that tha analog signal matches – I don’t need to use this mask anyway since I am aligned on SEL and also yes, I am only removing the bad trials from Bjorg’s list
* if there are more than 1 probe, then for that session, concatenate the channels and LFPSs from both the probes using concatenate\_probes subfunc from prepoc\_tools. – Here if both the probes are in PMd , then I am calling the second site as PMd2. The LFP and channel data are read from previously saved mat file in RAWData
* then filter the LFPs in each channel usign notch filter
* Then you cut the LFP in each channel based on markers, SEL, SC1 .. etc. The alignment of trials is given from outside the function. For now Alignment = SEL; for this you get the eventmarkers based on the alighnment from behaviour\_analog and using good\_trials\_mask

Saves 3 output structures for this session – epoch, annot, metadata.pkl

Note : need to give 2PMd if there are 2 probes in PMd in any session. If I give PMd2 then its not wrking correctly, it doesn’t separately pick up the names of the two probes

Also had to add the number of channels in each probe else the unipolar extraction code does not work correctly in the line where the layers\_mask are created.

Now changed to premotor since in the psd.computation code – area\_mask = [i\_ch for i\_ch, channel in enumerate(LFP\_epochs.ch\_names) if area in channel] in this line; it does not separate PMd and PMD

LFP Bipolar extraction

bipolar\_reference\_computation.py

Changed to bipolar\_reference\_computation\_modified.py

Need to give the path for the unipolar extracted LFP and the LFP file name

* Searches for the session and the unipolar extracted LFP and file name
* Reads - LFP-[session name] -Sel-full-epo.fif (if the alignment was Sel) using mne.read\_epochs
* Reads the metadata and checks first if the probe spacing is 150 or not ; If yes then she creates new channels by interpolation which are spaced at 50 microns , then computes the bipolar referencing between channels at width of 400 microns… that is, each bipolar channels has information from the channel 200 micrns before it and 200 microns after it
* For the bipolar referencing , depending on whether the spacing between channels is 100/200, she defines how many channels you need to take for the difference. Then for two channels separated at 400 micron, you take the difference of their signal which is the bipolar LFP signal and also store the depth of this newly created bipolar signal .
* Then outside the function you find the actual depth at which the layers were changing and see where your bipolar channels lie. Accordingly update the layers at which the new bipolar channels are created.

For now the bipolar output is stored in ../Unipolar\_sites. After Laura is done with the two sessions I will put them in Bipolar\_sites for each session

Also if each session has 2 probes then they are stored together ., as PMd- bip- ch(idx) and M1-bip-ch(idx)

Need to update to git

Git Committed July 26

* Added path output in this code which saves the data inside the /Bipolar\_sites folder
* Fixed the np.digitize(bip, depth\_layers) to correctly put SUP layers in bin 0, and so on – it would work even if there are 2 layers – this is in bipolar referencing code

**\*\*Note**

#To read the anot file do annot = mne.read\_annotations(path) . and then just annot.onset, annot.duration etc – basically the names that were saved

PSD Computation

psd\_computation.py

* Searches for the session and the bipolar extracted LFP and file name
* For each channel and each trial, PSD is computed – Using Welch method , sampling freq = 1000; freqs =[0.1 150] , window – Hammilton (default), overlap = 0.5 sec and window size = 1 sec. – these paramaters are kept the same as mentioned in the Gao et al paper – elife, 2020

Output – psd – array containing has shape (trials, channels, freq at which power is computed)

* Redefines the layers and the depths as L2/3, L5 , L6 and mixed. Mixed are those channels having the bipolar referencing between channels belonging to separate layers

FOOOF Laminar Timescales

Git Committed July 26

fooof\_laminar\_timescales.py

freqs for fooof - [1,150]

Output paths

path\_plots = server + f'/comco/nandi.n/LFP\_timescales/Results/Plots/{session}'

path\_fooof = server + f'/comco/nandi.n/LFP\_timescales/Results/FOOOF/{session}'

* added new subfuncion called compute\_plot\_foof\_layerwise(psd, area\_label, session\_label).

This is used to first average across the trials for each channel, and then across channels having the same label – L23, L5 and L6 (excluding mixed), then do the fooof on this signal and estimate the knee freq

So for each layer from each session you get an output of one timescale and one knee frequency

Finally saving the knee, tau, psd, layer, area , psd from that session are all saved in one pickle file in the session folder