**List of mice (highlighted mice are what I think you should start with – for IZ27 and IZ33 just use the saline folders to avoid confusion with the CA3 manipulations)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Folder location** | **Probe type** | **Number of channels** | **Region** | **Priority rank** | **Other notes** |
| IZ11 | Ipsi\_mEC | 5 shank, dense with middle linear shank 100 um spaced | 64 | CA1 | Low |  |
| IZ12 | Ipsi\_mEC | 2 shanks, 20 um spacing (700 um depth) | 64 | CA1 | Mid |  |
| IZ13 | Ipsi\_mEC | 2 shanks, 20 um spacing (700 um depth) | 64 | CA1 | Mid |  |
| IZ15 | CA1\_mEC | 5 shank, dense with middle linear shank 100 um spaced | 64 | CA1 | Low |  |
| IZ16 | Ipsi\_mEC | 2 shanks, 20 um spacing (700 um depth) | 64 | CA1 | Mid |  |
| IZ17 | Ipsi\_mEC | 2 shanks, 20 um spacing (700 um depth) | 64 | CA1 | Mid |  |
| IZ18 | CA1\_mEC | 2 shanks, 20 um spacing (700 um depth) | 64 | CA1 | Mid |  |
| IZ20 | CA1\_mEC | 1 shank, 20 um spacing, 1300 um depth | 64 | CA1, CA3 | Mid |  |
| IZ21 | CA1\_mEC | 1 shank, 20 um spacing, 1300 um depth | 64 | CA1, CA3 | High | The CA1 manipulation was less effective in this mouse |
| IZ24 | Bilateral\_mEC | 2 shanks, 20 um spacing (700 um depth) | 64 |  | High |  |
| IZ25 | Bilateral\_mEC | 2 shanks, 20 um spacing (700 um depth) | 64 |  | High |  |
| IZ26 | Bilateral\_mEC | 2 shanks, 20 um spacing (700 um depth)  | 64 | CA1 | High | This mouse has bilateral recordings – one hemisphere has the 2 shank probe, the other a 1 shank probe |
| IZ27 | CA3\_mEC | 1 shank, 20 um spacing, 1300 um depth | 64 | CA1, DG | High |  |
| IZ28 | CA3\_mEC | 1 shank, 20 um spacing, 1300 um depth | 64 | CA1, CA3 | High |  |
| IZ29 | CA3\_mEC | 5 shank, dense with middle linear shank 100 um spaced | 64 | CA1 | Low |  |
| IZ30 | CA1\_mEC | 5 shank, dense with middle linear shank 100 um spaced | 64 | CA1 | Low |  |
| IZ31 | CA1\_mEC | 2 shanks, 20 um spacing (700 um depth) | 64 | CA1 | Mid |  |
| IZ32 | CA3\_mEC | 5 shank, dense with middle linear shank 100 um spaced | 64 | CA1 | Low |  |
| IZ33 | CA3\_mEC | 2 shanks, 20 um spacing (1300 um depth) | 128 | CA1,CA3 | High |  |
| IZ34 | CA3\_mEC | 2 shanks, 20 um spacing (1300 um depth) | 128 | CA1,CA2(?),CA3 | Low |  |

**Other useful .mat files and subfields**

1. **LFP**

Name: ‘sessionName.lfp’

Notes: This is special type of binary file that is read using the buzsaki lab code base. I’m not really sure how you would read it if you didn’t use our code, but maybe it’s the same as the peyrache dataset?

Sampling rate is 1250 Hz

1. **Metadata**

Name: sessionName.sessionInfo.mat

Notes: Contains a lots of information about the session but perhaps the most relevant for lfp is,

**sessionInfo.AnatGrps.** This is a cell array, with each cell containing the order of the channels for a shank (so the number of cells would be number of shanks +1). The last cell will contain channels for my analog inputs – you can ignore that.

1. **Stimulation timestamps**

Name: sessionName.pulses.events.mat

pulses.timestamps is a nx2 array of timestamps for pulses (start, stop). Some will be on the maze, others will be on the track. Note I usually had two track sessions in a day, and this will contain everything. Also note that these are all pulses (so could be ipsi mEC, alternate manipulation, depending on animal, and both ipsi mEC and alternate manipulation).

The logical to determine which timestamp corresponds to a manipulation is using the field pulses.stimComb. Here, 1 is alternate manipulation (so either CA1, contralateral mEC, or CA3), 2 is (always) ipsi mec manipulation, 3 is both together. BUT, the 3 is repeated twice so I use a second logical to extract that. The script I use is this –

 for rr = 1:3

 if rr <= 2

 pulTr = (pulses.stimComb==rr);

 else

 pulTr = (pulses.stimPerID'==1 & pulses.stimComb==rr);

 end

end

Sorry, confusing, I know. Hope that made sense. You could just start with pulses.stimComb == 2 if all you’re looking at is the ipsi mEC.

1. **Track manipulation and targeting information**

Name: sessionName.SessionPulses.events.mat

This will usually have two fields, each corresponding to a separate behavior session for that day. Within each field, it has the following subfields

Timestamps: ON timestamps for stimulation per trial, this is 0 if stimulation was not provided in a trial. I did blocks of 10 trials alternating between no stimulation and stimulation

Stim: 0 or 1 logical, on whether a trial was no stimulation or stimulation

Target: 1 or 2, 1 meant stimulation was targeted to the center arm, 2 meant stimulation was targeted to the side

Region: Manipulation – same logic as pulses – 1 is alternate manipulation, 2 is ipsi mEC, 3 is both.

Name: sessionName.SessionArmChoice.Events.mat

Similar to sessionPulses, and just has more information on the trial by trial metrics, i.e., whether is was a left or right trial, the animal’s choice etc. If you’re interested, let me know and I can walk you through it.

1. **Behavior**

Name: sessionName.Behavior.mat

It has all the extracted position data with the following fields,

Behavior.timestamps

Behavior.position.x – x position for each timestamp

Behavior.position.y – y position for each timestamp

Behavior.position.lin – linearized position for each timestamp

Behavior.events/trials – contains a lot of subfields for timestamps for behavior. Let me know if you are interested

Behavior.masks.trials – each data point is assigned a trial number corresponding to sessionArmChoice and sessionPulses.

Behavior.masks.recording - if there were multiple behavior sessions, which one each data point this corresponds to.

I know this is a lot! At this point I would suggest just extracting lfp, and looking at timestamps on the track during no stimulation and stimulation in different parts of the maze? Here is the script that I use to extract these timestamps for no stimulation and stimulation-

 efields = fieldnames(sessionPulses);

 for jj = 1:length(efields)

 region = sessionPulses.(efields{jj}).region; %1 is CA1/CA3, 2 is mec, 3 is both

 target = sessionPulses.(efields{jj}).target; %1 is stem, 2 is return

 rewardTS = sessionArmChoice.(efields{jj}).timestamps;

 startDelay = sessionArmChoice.(efields{jj}).delay.timestamps(1,:)';

 endDelay = sessionArmChoice.(efields{jj}).delay.timestamps(2,:)';

for zz = 1:6 % the maze is split in 3 zones – side, center and delay. So stim and no stim make 6 %conditions

 switch zz

 case 1 %First, no stim trials, return

 startTS = rewardTS(sessionPulses.(efields{jj}).stim(1:(end-1))==0);

 endTS = startDelay(sessionPulses.(efields{jj}).stim(1:(end-1))==0);

 events = [startTS'; endTS'];

 case 2 %No stim, stem

 startTS = endDelay(sessionPulses.(efields{jj}).stim(1:(end-1))==0);

 endTS = rewardTS(find(sessionPulses.(efields{jj}).stim(1:(end-1))==0)+1);

 events = [startTS';endTS'];

 case 3 %No stim, delay

 startTS = startDelay(sessionPulses.(efields{jj}).stim(1:(end-1))==0);

 endTS = endDelay(sessionPulses.(efields{jj}).stim(1:(end-1))==0);

 events = [startTS';endTS'];

 case 4 % Stim, return

 startTS = rewardTS(sessionPulses.(efields{jj}).stim(1:(end-1))==1);

 endTS = startDelay(sessionPulses.(efields{jj}).stim(1:(end-1))==1);

 events = [startTS';endTS'];

 case 5 % Stim, stem

 startTS = endDelay(sessionPulses.(efields{jj}).stim(1:(end-1))==1);

 endTS = rewardTS(find(sessionPulses.(efields{jj}).stim(1:(end-1))==1)+1);

 events = [startTS';endTS'];

 case 6 %stim, delay

 startTS = startDelay(sessionPulses.(efields{jj}).stim(1:(end-1))==1);

 endTS = endDelay(sessionPulses.(efields{jj}).stim(1:(end-1))==1);

 events = [startTS';endTS'];

 end

end

end