**OVERVEIW / how to run PPSeq**

1. get a recording (ephys, behaviour and cameras)
2. transfer this data onto ceph
3. kilosort the spikes
4. run the DLC video tracking on the cameras
5. align the behavior and cameras to the ephys "Postprocess\_Neuropixel"
6. put all of these processed data together into a specific file structure that my code is built for (see file structure)
7. prep for PPSeq: extract spikes which will be used for ppseq
8. run ppseq on awake
9. extract sleep spikes for ppseq
10. run ppseq on sleep
11. run my PPseq postprocessing scripts to extract replays
12. analyse the replay

**TTLS/Triggers for aligment**

The bpod is the central clock.

cameras

bpod

ephys

It sends TTLS to the cameras and the ephys.

During the task the TTL goes high at each trial start (when the bpod loads the trial) and goes low at first poke (when the mouse pokes into the first task relevant port, eg. Port 2 in the 5 step sequence: 2,1,6,3,7)

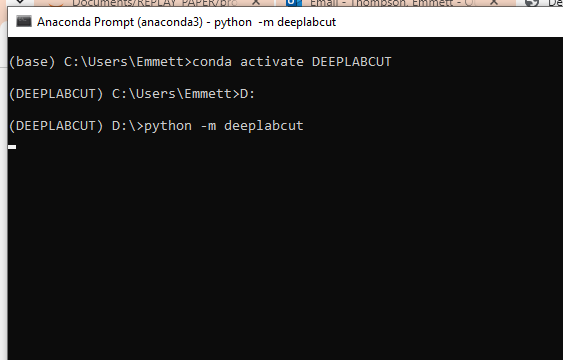
During sleep (TTL\_pulse thingy) the bpod sends out a regular 1s TTL pulse every 30s.

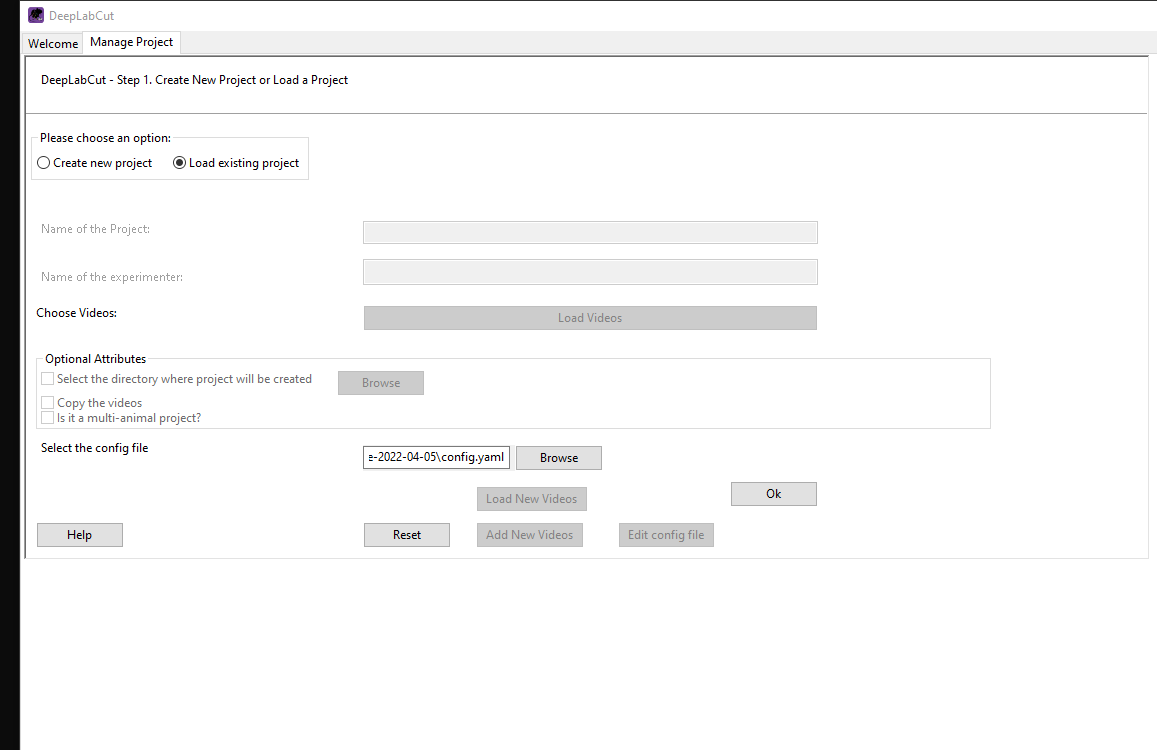
For each of these bpod also sends a barcode of short TTLS at the start so you can find the start of each period (task and sleep periods)

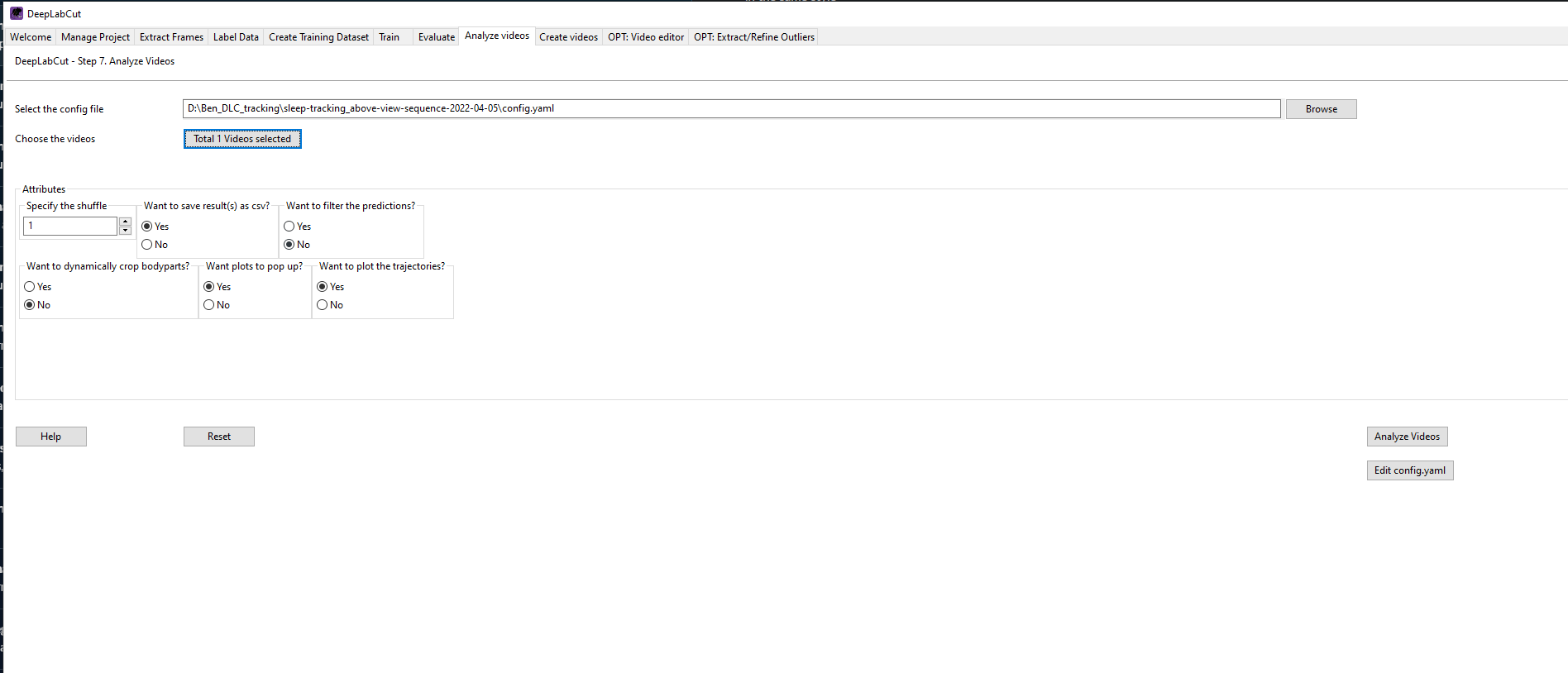
Note: on the setup, the ephys also sends TTLS from the neuropixel processor to the NI-DAQ so that the ephys can align the neural recordings with any triggers it reads in (the ‘events’) (open ephys global timestamps stuff)

**4. Video tracking**  
this is just incase you need to run this when i'm not here. You will have to run this locally on my computer.

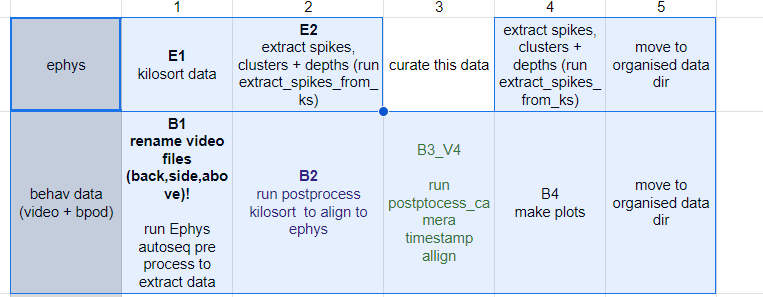
1. open anaconda prompt and activate DLC env, move to D: drive, start deeplabcut (see photo)
2. copy any videos you want to run into this directory: D:\Ben\_DLC\_tracking\
3. there are 3 tracking stages: above (sleep), back (the task) and back ports (to know where the ports are)
4. each has a different trained network
5. load the relevant config file for the network you want to use
6. go to analyse videos and select the ones you want to analyse (you can do multiple) and click run

****

****



**5. Postprocess\_Neuropixel**

****

These are the steps from left to right.

Ignore curate.

There is also a file called V3 you need to run (this aligns the pre/post sleep cameras to the ephys)

These files are under other/Postprocess\_Neuropixel here:

<https://github.com/EmmettJT/procedural_replay_paper_2024/tree/main>

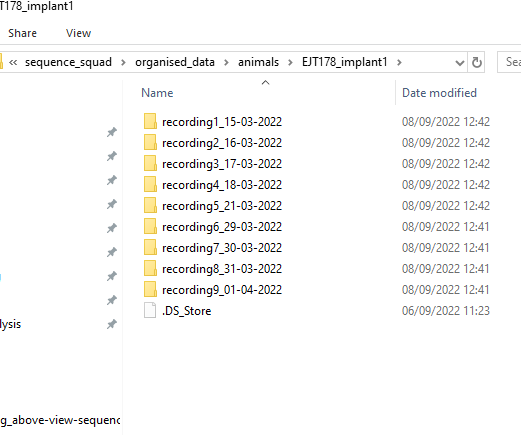
**6. file structure**

See here for examples:

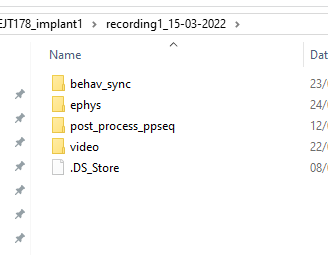
Z:\projects\sequence\_squad\organised\_data\animals

Each mouse\_implant has a file

In these files exist all the recordings for that mouse\_implant



Each recording has these files:



See this file structure on ceph to know what goes in each file

1. **Prepping for PPseq on awake:**
2. create a folder called 'post\_process\_ppseq' inside your recording folder for each animal
3. inside this create a txt file called "Time\_intervales.txt" ('intervales' isnt a spelling mistake....well it is, but its a legacy spelling mistake that is embedded throughout the code).
4. add some example timespans into the txt file: eg. Awake,[[400,800]] Post\_sleep,[[5000,5500],[7600, 8000]]
5. run ppseq\_behaviour\_scoring: this takes the sleep tracking file and returns the animals velocity during the sleep period (for selecting ppseq sleep run time points). it also needs to return something called Performance\_score.csv which will help you select time points that mouse was doing the task well. You will need to also add something about which of the 2 sequences the mouse was doing here. This doesn't need to be complex, just something that tells us roughly how well the mouse was doing for each sequence. We can talk about this tomorrow
6. Finally run Prepare\_data. this prepares the data for a ppseq run. It will create a plot showing where the selected ppseq timespan was across the whole recording, so we can adjust it based on the performance and mouse velocity etc.  Probably this script will need a tiny bit of editing to work with your data, but hopefully not too much
7. Run PPSeq - not that complicated...but I will save it for another instructional chunk

**8. Running PPseq awake**

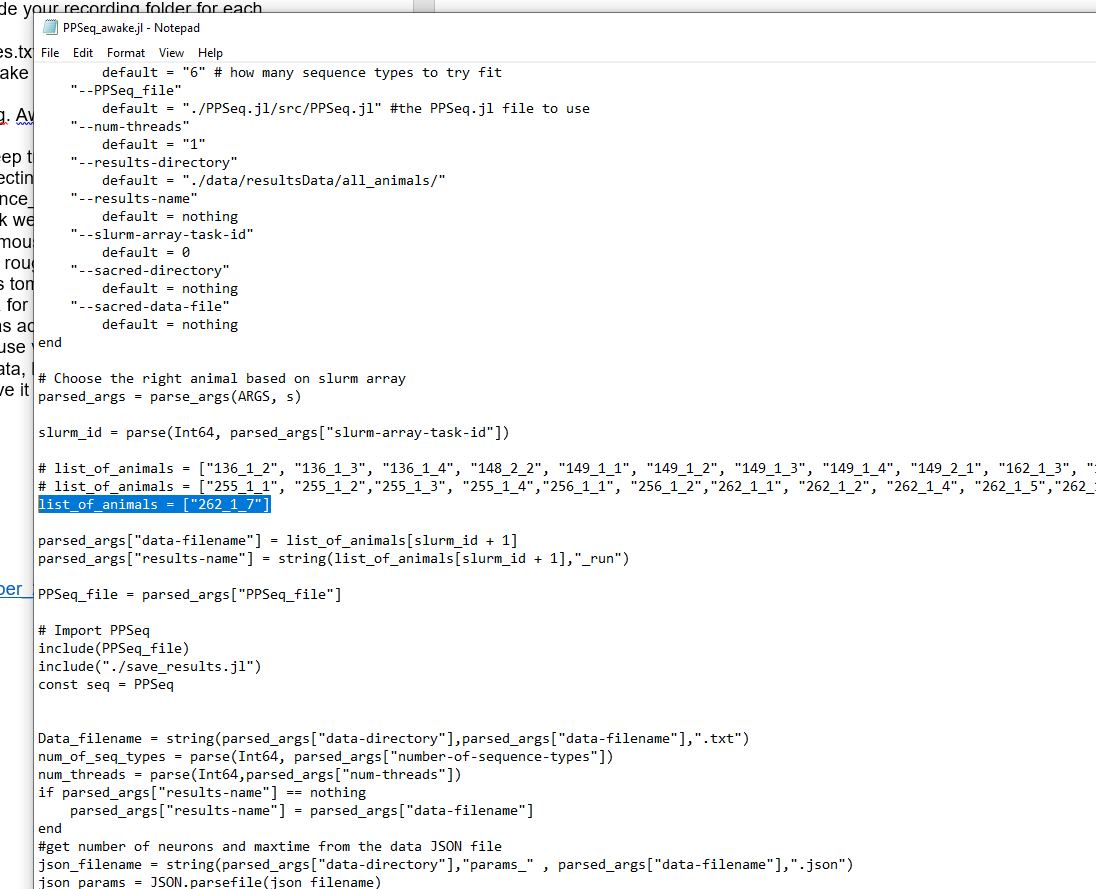
Clone this onto your hpc (cluster) account

<https://github.com/EmmettJT/procedural_replay_paper_2024/tree/main>

in other/ppseq you will find the ppseq files.

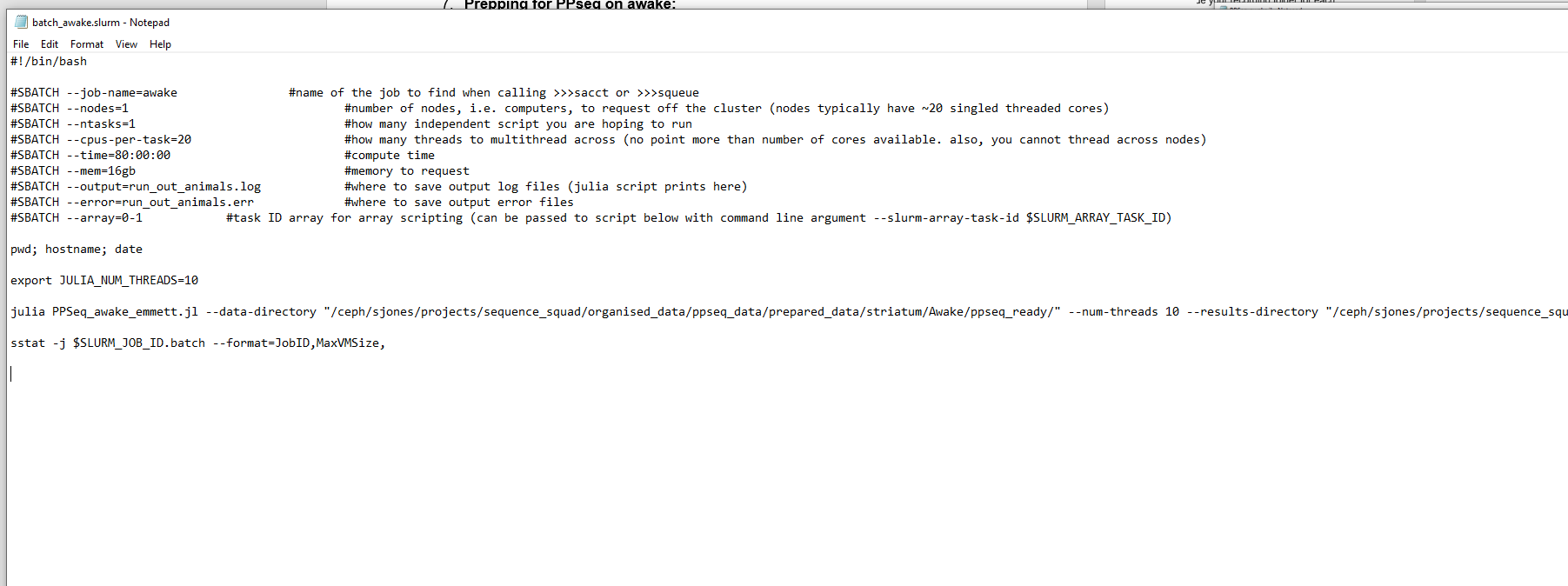
Open ppseq\_awake.jl and in ‘list\_of\_animals’ add in the names of the recordings you want to analyse – In the format mouse\_implant\_recording

Save this and copy it to your cloned folder on the cluster



Open the batch file for awake

Update the paths so that they point to your prepared data

****

**9. Processing the awake PPseq output**

In Here<https://github.com/EmmettJT/procedural_replay_paper_2024/tree/main>, under other\replay\_analysis\

Run this notebook:

PPseq\_awake\_postprocess.ipynb

This will pre-processes the awake output and allow you to reorder and recolour the sequences (manually)

This script:

PPseq\_awake\_alignment\_clean.ipynb

Is similar but slightly cleaner for producing nicer plots after you have done the processing.

**10. Running PPSeq sleep**

Prepare the spike in the same was as step 7. But choose sleeping time periods and int eh notebook change the time period to say something like ‘Post\_sleep’ instead of awake

Run the same way as step 8 except you need to change PPseq\_sleep.jl to add the mouse\_implant\_recordings and the sleep batch file needs to have something called

--sacred-directory “PATH TO YOUR AWAKE PPSEQ OUTPUT”

Point this to your awake output – it will use the params saved out from the awake to fit to the sleep data – ie. Search for the awake fit sequences during sleep.

**11. Processing the sleep PPseq output**

In Here<https://github.com/EmmettJT/procedural_replay_paper_2024/tree/main>, under other\replay\_analysis\replay\_analysis\_2024\

You will find some notebooks that are labelled part 1,2,3,4,5

1 and 2 are the preporcessing scripts. If you update your paths these should extract the replay and some other fetaures of replay that are useful.

3 is also a processing file but I think its not needed for your purposes.

4 and 5 are analysis scripts that I used to do plotting.

Something you will need to do is work out what the task ordering is so that you know whether the replays are in order or not. I think I did this manually and created a data frame – it will become more clear when you get to this stage.