About this document

This document explains the general structure of the code (and data) used to create the majority of the photometry data plots (all photometry plots except ED figure 5 O-Y and ED figure 12 plots).

In general, if you have the repro data folder all you should need to run is the individual figure plotting files (named e.g. fig2DE.py). All plotting functions now work on the repro data. Most have an if statement something like: if os.path.exists(repro_file): load that, else: load something else and maybe do some analysis.

If you are just running from the **reproducing_figures dataset** (I will call this **repro data** in this document) you will need to change **reproduce_figures_path** in **set_global_params.py** to where you have saved this data.

I also include some instructions for if you wanted to run it all from scratch or see how these files were generated. I do this by explaining what each file in the repo does, and for the plotting files, their dependencies. If a file is mostly involved in restructuring data to make it easier to plot/ do other analyses on, I also explain the rough structure of the data that is saved out.
I have tried to list all dependencies of each file but it is worth checking any file that is a dependency to see if that also has a dependency.

Please change the directory **all_data_path** in **set_global_params.py** to where you have saved the full data set. In this document I will refer to some paths as those that are defined in **set_global_params.py** so refer to this document for file locations.

For any files that are formatted with the mouse, date in the file or folder name, all mouse IDs are in the csv **‘exp_records_APE_paper.csv’** along with dates and information about what type of experiment was performed (e.g. silence, psychometric, value change, omissions and large rewards, state change). Dates are all formatted YYYYMMDD. This csv is also the full record of the photometry experiments carried out in this paper (excluding figures ED fig5O-Y and ED fig12).

Requirements

This code was run using Python 3.8 and the packages stated in the requirements.txt file.

General Workflow

All analysis (if not reproducing figures from repro files) assumes you at least have the demodulated and smoothed photometry files and restructured bpod data file (files saved in processed_data_path/mouse with format mouse_date_smoothed_signal.npy and mouse_date_restructured_data.pkl respectively).

Generally there will be 1-2 stages of processing between generating those files and making the figure. The general work flow tends to be: extract behavioral events of interest and traces, save these out in an object or dataframe for further analysis. Sometimes dopamine response sizes are also calculated and saved out as part of the object or dataframe, unfortunately sometimes these are in a dataframe column called ‘APE peaks’ which is obviously not well named for VS data, but these are the dopamine responses. These objects/dataframes are then loaded in by further analysis scripts for regression/ plotting.

I have included a description of what the attributes of the objects and columns in these dataframes correspond to in the descriptions of the files that generate them below.

Data (full data set)

- All photometry recording sessions have (mouse and date are proxies):
    1) A bpod file: this is the behavioral data as it comes out of bpod.
       behavioural_data_path\mouse\protocol\Session Data\mouse_protocol_YYYYMMDD_HHMMSS
       Protocol is normally Two_Alternative_Choice, a few mice were initially trained on Two_Alternative_Choice_CentrePortHold
    2) A raw analog input file: AI.tdms - this is the raw photometry signal and the camera trigger times
       photometry_data_path\mouse\YYYYMMDD_HH_MM_SS
    3) A demodulated, smoothed photometry signal file:
       processed_data_path/mouse/mouse_YYYYMMDD_smoothed_signal.npy
       (this is created by *data_preprocessing/pre_processing.py*)
    4) A restructured bpod behavior file:
       processed_data_path/mouse/mouse_YYYYMMDD_restructured_data.pkl
       (this is created by *data_preprocessing/pre_processing.py*)
- It is much easier to just work from 3) and 4) so most analysis just uses these files. Occasionally code will load in 1) and 2) to sync up with the tracking data.
- All behavior, including sessions where there was no photometry is saved in, allowing you to plot learning curves if you so wish:
  behavioural_data_path/ mouse/Two_Alternative_Choice/Data_Analysis/mouse_dataframe.pkl

- Many figures also require one of the following files at some point in their workflow:
    1) **processed_data_path/mouse/mouse_date_aligned_traces.p**
        - These files are included in the full data set
        - This is created by ***figure3/save_out_aligned_traces_for_session.py*** for sessions in normal training
        - Or by ***data_preprocessing/extract_aligned_traces_general.py*** for special experiments (such as silence)

2) **processed_data_path/for_figure/mouse_date_aligned_traces_for_figure.p** (these are used in figure 2 because this file also include traces aligned to times of entering choice port but getting no reward, which are only used in this figure)
    - These files are included in the full data set
    - This is created by ***figure2/format_first_three_sessions_for_heatmap.py***

  For details about how the data is structured in these files see: section 'Aligned data traces files' below.

- Tracking data can be found in:
  raw_tracking_path or for some mice **old_raw_tracking_path** (see individual scripts for which one)
- The experiment with mice running in a box without a task in ED fig5 HIJ data is in:
  running_in_box_dir and **running_in_box_tracking_dir**

Aligned traces files (session_data objects)

In the full data set there are files called aligned_traces, or aligned_traces_for_fig or aligned_traces_for_psychometric. These are pickled objects that have the general structure detailed below. **Note the params attribute that details exactly which parameters were used to generate the ZScoredTraces object within the SessionData object.**

Algined_traces_for_fig and aligned_traces_for_psychometric have outcome_data.reward_data and outcome_data.no_reward_data
Aligned_traces has reward_data.ipsi_data and reward_data.contra_data (only correct trials).

All of these files use the **object class SessionData** (defined in ***utils/individual_trial_analysis_utils.py***).

Often the SessionData object is loaded in as **session_data**:

- Session_data has attributes: **session_data.choice_data**, **session_data.cue_data**, **session_data.outcome_data**, **session_data.reward_data**
    - choice_data then has attributes **ipsi_data** and **contra_data**, so does cue_data and reward_data (reward data is only correct trials, look at outcome_data if you want both correct and incorrect)
    - outcome_data has attributes **reward_data** and **no_reward_data**

- Ipsi_data, contra_data, reward_data and no_reward_data are all of class **ZScoredTraces** (which is also defined in ***utils/individual_trial_analysis_utils.py***) and all have attributes:
    - **Params:** this is particularly useful if you change any of the default params - it tells you what params the objects were generated with (see utils.individual_trial_analysis_utils.HeatMapParams class for full details on what all of these are). Mostly these filter behavioral events and decide whether to align to beginning or end of states from the bpod state machine. There are also parameters relating to whether to include events if the animal repeated them -

this is particularly important early in learning as animal behavior is very messy. Because APE is a movement-related signal, animals moving back and forth between ports early in learning make for very messy signals so we tend to select trials where the animals go center port -> hear cue -> make choice -> get reward (or not) to make the photometry easier to interpret. To try to be conservative in how much we filter, we do this per event rather than per trial because if we align to reward it is less of a problem if the animal has done multiple choice movements than if we align to movement.

- **Sorted_traces:** these are the photometry traces aligned to the event, trials sorted by either trial number or the duration of the event (this is evident from the trial_nums or reaction_times (duration of event) attribute.
- **Trial_nums:** the trial numbers for the traces, sorted in the same way as for all the other attributes
- **Reaction_times:** this is generally the duration of the behavioural event - it’s mostly only interesting for choice_data.ipsi_data or choice_data.contra_data as these are the only events with variable reaction times that I align to to save out. These are the white dots in the heatmaps.
- **Mean_trace:** the mean across all traces for this behavioral event for the session
- **Time_points:** the time points (in seconds) that correspond to the traces. 0 is the time of the behavioral event. There are normally 6-8 seconds either side saved out (this is the window used to z-score the data).
- **State_name:** tells you which bpod state the data was aligned to (can be start or end you’ll need to look in params for that)
- **Outcome_times:** time of next reward (or not if they got the trial wrong)
- **Sorted_next_poke:** the beginning of the next trial
- **Event_times:** the actual time in seconds (in the session, from the beginning) of the behavioral event (i.e. the event that is at 0s in the trace)

Behavior

A note on trial selection for analysis:

- This study looks at photometry very early in learning, often within the first 3-5 times the animal was placed in the task.
- This behavior is a self paced behavior with some shaping.
- This means there is a huge amount of variability in exactly what is happening on any given trial (especially when the mouse has not learned the structure of the task).
- To be able to align to some behavioral events we sometimes pre select for trials where the animals did not experience a bpod state (see below) multiple times to look mostly at trials where the trial structure was cue -> choice -> reward.
- We do this selection per state to still include choice movements where they maybe heard the cue multiple times, to include as much data as possible given the huge amount of behavioral variability.
- The files **processed_data_path/mouse/mouse_date_aligned_traces.p** and **processed_data_path/for_figure/mouse_date_aligned_traces_for_figure.p** have a

record of the parameters used to select behavioral events and have the trial numbers, event times, traces, bpod state durations saved in them.

Bpod

The bpod is a state machine: it has various behavioral task states and these correspond to stages in the trial, e.g. wait for center poke or wait for choice. Changes in state are triggered by the animal poking into ports, or other task events.

Experiencing states multiple times in a task:

The task is structured such that the mice have to put their head in the center port to hear a cue. They then have to hold their heads there for a certain period to be able to make a choice. Early in training this period is sometimes shorter to help encourage the mice to engage in the task. If the mice withdraw from the port early, they often try to make choice movements but they won't count (they won't be able to get reward) because they didn't hold their heads in long enough.

Shaping protocol:

Also part of the shaping protocol early in training are several sessions where the mice can make an incorrect choice and correct it and still get the reward. This again helps encourage engagement. Later in training an incorrect choice triggers a timeout (punishment). To look at the first incorrect choice in the non-punished sessions I add in a bpod state into the processed data called state 5.5 that is not in the raw bpod data that corresponds to the first incorrect choice made by the mice in these early sessions. As the state of the bpod only changes when the animal makes the correct choice, to get the time of correct choice we can still align to the end of state 5 (wait for choice).

Figure 2

General note: most of the files require **aligned_traces_for_fig** files which are produced by *figure2/format_first_three_sessions_for_heatmap.py* - **these files are included in the full data set**

figure2/ fig2DE.py

Makes:

- fig 2D (heatmaps)
    - Requires repro data or files generated by *figure2/format_first_three_sessions_for_heatmap.py*
- fig2E (average traces)
    - Requires either repro data or files generated by f*igure2/per_mouse_average_traces.py* (which requires *figure2/format_first_three_sessions_for_heatmap.py*)

figure2/format_first_three_sessions_for_heatmap.py

- Aligns photometry trace to behavioral events for fig 2D and E.
- Creates object **session_data** with the general structure (one per session). I will also refer to this as **aligned_traces_for_fig**:

    - Session_data has attributes: **session_data.choice_data**, **session_data.cue_data**, **session_data.outcome_data**, **session_data.reward_data**
    - choice_data then has attributes **ipsi_data** and **contra_data**, so does cue_data and reward_data (reward data is only correct trials, look at outcome_data if you want both correct and incorrect)
    - outcome_data has attributes **reward_data** and **no_reward_data**
    - See section ‘Aligned traces files’ on page 3 for more details
    - Objects saved out to: data_dir = os.path.join(processed_data_path, 'for_figure', mouse)
      filename = mouse + '_' + date + '_' + 'aligned_traces_for_fig.p'

figure2/Per_mouse_average_traces.py

- Loads in the saved out objects from format_first_three_sessions_for_heatmap.py and creates an average trace per mouse across sessions for ipsi choices, contra choices, cues, rewards and no rewards.
- Saves out averages to: dir = os.path.join(processed_data_path, 'for_figure')
  file_name = 'group_data_avg_across_sessions_' + site +'_new_mice_added_with_cues.npz'
  Where site is nacc, tail (or for ED figure ant_tail)

figure2/kernel_regression/fig2FG.py

Makes:

- Fig 2F (kernels)
    - Requires either repro data or running: 1) *get_time_stamps_for_regression_all_cues_matched_trials.py* and 2) *linear_regressions_all_session_different_shifts.py* (both in *figure2/kernel_regression*)
- Fig 2G (explained variances)
    - Requires either repro data or running: 1) *get_time_stamps_for_regression_all_cues_matched_trials.py*, 2) *linear_regressions_all_session_different_shifts.py* and 3) *percentage_variance_explained.py* (all files in *figure2/kernel_regression*)

figure2/kernel_regression/get_time_stamps_for_regression_all_cues_matched_trials.py

- Gets the times of behavioral events to include in the kernel regression
- Cue events, choice events (leaving center port and moving to side port), reward (and no reward) events.
- These are very similar to the events aligned to in figure 2D and E but we are slightly more lenient with the events included as we allow for repeated behavioral repeats (e.g. mouse moving back and forth between ports or hearing the cue a few times in a row if they don’t hold their heads in the center port for long enough) as the kernel regression should be able to disentangle overlapping behavioral events better than just plotting averages. This is actually one of the advantages of such analysis.
- We then filter events for trials where there is at least one cue, choice and outcome event, to make sure numbers of these events are the same (to prevent this biasing the regression).

- The objects created are very similar in structure to session_data objects but they do not have any photometry data in them.
- Creates object **session_events** with the general structure (one per session):
    - Session_events has attributes: session_events.choice_events, session_events.cue_events, session_events.reward_events
    - choice_events then has attributes ipsi_data and contra_data, so does cue_events
    - reward_events has attributes reward_data and no_reward_data
    - Ipsi_data, contra_data, reward_data and no_reward_data all have attributes:
        - Params: see above (for session_data objects). As mentioned we tend to allow trials with repeated events in the regressions, unlike for the averages, as the regression is better able to pull these events apart.
        - Trial_nums: the trial numbers for the evens, sorted in the same way as for all the other attributes
        - Reaction_times: this is generally the duration of the behavioural event - it's mostly only interesting for choice_data.ipsi_data or choice_data.contra_data as these are the only events with variable reaction times that I align to to save out.
        - Time_points: the time points (in seconds) that correspond to the traces. 0 is the time of the behavioral event. There are normally 6-8 seconds either side saved out (this is the window used to z-score the data).
        - State_name: tells you which bpod state the data was aligned to (can be start or end you'll need to look in params for that)
        - Sorted_next_poke: the beginning of the next trial
        - Event_times: the actual time in seconds (in the session, from the beginning) of the behavioral event (i.e. the event that is at 0s in the trace)
        - Trial_starts: time in seconds (in the session, from the beginning) that the trial started
        - Trial_ends: time in seconds (in the session, from the beginning) that the trial ended
- Events are saved into processed_data_path/mouse/linear_regression
  filename = mouse + '_' + date + '_' +
  'behavioural_events_with_no_rewards_all_cues_matched_trials.p'

figure2/kernel_regression/linear_regression_all_sessions_different_shifts.py

- Performs the regression using the (demodulated and smoothed) photometry trace and the session_event objects produced by get_time_stamps_for_regression_all_cues_matched_trials.py
- Saves out kernels that are loaded by fig2FG.py into processed_data_path/mouse

figure2/kernel_regression/percentage_variance_explained.py

- Requires 1) *get_time_stamps_for_regression_all_cues_matched_trials.py* and 2) *linear_regressions_all_session_different_shifts.py* (loads in kernels from processed_data_path/mouse)

- Calculates the percentage variance explained by the kernel regression model/ each event type (by removing the predicted signal due to a given event type and seeing how much the variance explained changes)
- Saves variance explained for all mice (for a given recording site) in processed_data_path/linear_regression_data/site + '_explained_variances_all_cues.p', where site is tail or nacc
- These files are loaded by fig2FG (and there is a copy of them in the repro data)

Figure 3

General note: most of the files require **aligned_traces files** which are produced by *figure3/save_out_aligned_traces_for_session.py* - **these files are included in the full data set**

figure3/fig3AE.py
Makes:

- Fig3 A Example average traces per 200 trials over learning for a VS mouse (cue aligned)
- Fig3 E Example average traces per 200 trials over learning for a TS mouse (choice aligned)
    - Requires repro data or running:
      1) *figure3/save_out_aligned_traces_for_session.py* (for all sessions) and then
      2) *figure3/change_over_time_get_peaks.py* run with window_for_binning=200

figure3/fig3BF.py

- Fig3 B Example dopamine response size over learning for a VS mouse (cue aligned)
- Fig3 F Example dopamine response size over learning over learning for a TS mouse (choice aligned)
    - Requires repro data or running:
      1) *figure3/save_out_aligned_traces_for_session.py* (for all sessions) and then
      2) *figure3/change_over_time_get_peaks.py* run with window_for_binning=40

figure3/fig3CG.py

- Fig3 C Average across VS mice dopamine response size over learning (cue aligned)
- Fig3 G Average across TS mice dopamine response size over learning over learning (choice aligned)
    - Requires repro data or running:
      1) *figure3/save_out_aligned_traces_for_session.py* (for all sessions) and then
      2) *figure3/change_over_time_get_peaks.py* run with window_for_binning=50

figure3/fig3DH.py
Makes:

- Fig3 D VS beginning and end comparison plots
- Fig3 H TS beginning and end comparison plots

Requires repro data or:

- files of format (mouse is e.g. SNL_photo17): behavioural_data_path/ mouse/Two_Alternative_Choice/Data_Analysis/mouse_dataframe.pkl

    - These are the summary behavior files for all training sessions (including those without photometry), used to get actual trial numbers from start of training, not just within session.
- Running:
  1) *figure3/save_out_aligned_traces_for_session.py* (for all sessions)

figure3/fig3JKM.py

Makes:

- Fig3 J Dopamine traces at choice for example TS mouse depending on last choice ipsi or contra
- Fig3 K regression coefs for effect of choice ipsi or contra on current trial -1 to -5 on size of current trial APE response when taking a contra choice
- Fig3 M Dopamine traces at cue for example VS mouse depending on last choice ipsi or contra
    - All three plots require repro data or running:
      1) *figure3/save_out_aligned_traces_for_session.py* (for all sessions) and then
      2) *figure3/format_data_for_trial_history_analysis.py*

figure3/fig3PQRS.py

Makes:

- Fig3 P Tone/ white noise traces for TS example mouse
    - Requires either repro data or:
        - Smoothed and demodulated photometry and restructured behaviour files for the session (files saved in processed_data_path/mouse with format mouse_date_smoothed_signal.npy and mouse_date_restructured_data.pkl respectively)
- Fig3 Q Dopamine response comparison TS mice at time of choice
    - Requires either repro data or running:
      figure3/reformat_state_change_data.py, which makes files:
      processed_data_dir = os.path.join(processed_data_path, 'state_change_data')
      state_change_data_file = os.path.join(processed_data_dir, 'state_change_data_{}_mice_only_correct_py36.p'.format(site)) for site = tail
- Fig3 R Tone/ white noise traces for VS example mouse
    - Requires either repro data or:
        - Smoothed and demodulated photometry and restructured behaviour files for the session (files saved in processed_data_path/mouse with format mouse_date_smoothed_signal.npy and mouse_date_restructured_data.pkl respectively)
- Fig3 S Dopamine response comparison VS mice at time of cue
    - Requires either repro data or running:
      figure3/reformat_state_change_data.py, which makes files:
      processed_data_dir = os.path.join(processed_data_path, 'state_change_data')
      state_change_data_file = os.path.join(processed_data_dir, 'state_change_data_{}_mice_only_correct_py36.p'.format(site)) for site = nacc

figure3/save_out_aligned_traces_for_session.py

- This is very similar to *figure2/format_first_three_sessions_for_heatmap.py* but instead of having outcome_data.reward_data and outcome_data.no_reward_data it has reward_data.ipsi_data and reward_data.contra_data (we don't align to times of no reward at outcome for any analysis in figure 3)
- See section 'Aligned traces files' on page 3 for more details about the objects)
- This takes in the demodulated and smoothed photometry and restructured behavior files and aligns photometry traces to behavioral events
- This is run for every session for all the mice that we have continuous recordings over the course of learning
- Objects saved out to: data_dir = os.path.join(processed_data_path, 'for_figure', mouse)
  filename = mouse + '_' + date + '_' + 'aligned_traces.p'
- I will also refer to the files created as **aligned_traces** files
- These files are included in the full data set

figure3/change_over_time_get_peaks.py

- Loads objects created and pickled by *figure3/save_out_aligned_traces_for_session.py* takes the mean trace per window_for_binning=50 trials and finds the first peak after cue (for VS) and leaving center port (for TS). Defaults are align_to=cue for VS and align_to=movement for TS. Default side=contra
- Saves these peak sizes out to: save_path = os.path.join(processed_data_path, 'peak_analysis')
  mouse + '_binned_' + str(window_for_binning) + '_average_then_peaks_peaks.npz'
- Also has option to align to non default things (not plotted in paper and these will produce files: mouse + '_binned_' + str(window_for_binning) +'_average_then_peaks_peaks_{}_aligned_to_{}.npz'

figure3/format_data_for_trial_history_analysis.py

- This reformats the photometry and behavioural data over the whole course of training into a dataframe that is easier to work with than those objects. This is the same data as is used for the change over time plots for the TS, just saved in a different format. For the VS these are DA responses at time of reward not cue.
- Requires running: *figure3/save_out_aligned_traces_for_session.py*

figure3/reformat_state_change_data.py

- This reformats the photometry and behavioral data for the state change (tone, white noise) experiment.
- Gets dopamine peak responses for each trial at time of cue (nacc) or choice (tail)
- Requires: Smoothed and demodulated photometry and restructured behaviour files for the session (files saved in processed_data_path/mouse with format mouse_date_smoothed_signal.npy and mouse_date_restructured_data.pkl respectively)
- Saves out data to:
  processed_data_dir = os.path.join(processed_data_path, 'state_change_data')
  state_change_data_file = os.path.join(processed_data_dir, 'state_change_data_{}_mice_only_correct_py36.p'.format(site))

Figure 4

figure4/PsychometricAnalysis/fig4HJ.py

Makes:

- Fig4 H Bias caused by last trial large or small dopamine response size TS
- Fig4 J Bias caused by last trial large or small dopamine response size VS
    - Require repro data or running:
      1) *figure4/extract_aligned_traces_psychometric.py* needed only for VS mice (gets traces aligned to correct and incorrect at time of outcome) **files included in full data set**
      2) aligned traces (from running *data_preprocessing/extract_aligned_traces_general.py*) on psychometric days for TS mice **Files included in full data set**
      3) *figure4/reformat_psychometric_data.py* which produces 3 files:
      save_path = os.path.join(processed_data_path, 'psychometric_data')
      File1 = os.path.join(save_path, "all_trial_data_Nacc_contra_ipsi_last_trial_confidence_and_traces_no_tracking_reward_aligned_pk5.pkl")
      File2 = os.path.join(save_path, "all_trial_data_tail_contra_ipsi_last_trial_confidence_and_traces_no_tracking_choice_aligned_old_data_pk5.pkl")
      File3 = os.path.join(save_path, "all_trial_data_tail_contra_ipsi_last_trial_confidence_and_traces_no_tracking_choice_aligned_pk5.pkl")
      (tail data was acquired in two batches that were processed separately)
      2) *save_relevant_data_nacc.py* and *save_relevant_data_tail.py* that produces:
      processed_data_path/psychometric_data/all_tail_data_for_paper.csv and
      processed_data_path/psychometric_data/nacc_data_for_paper.csv

figure4/PsychometricAnalysis/fig4IK.ipynb

Makes:

- Fig4 I Regression coefficients for logistic regression TS
- Fig4 K Regression coefficients for logistic regression VS
    - Require repro data or running (same as for fig4HJ):
      1) *reformat_psychometric_data.py* which produces 3 files:
      save_path = os.path.join(processed_data_path, 'psychometric_data')
      File1 = os.path.join(save_path, "all_trial_data_Nacc_contra_ipsi_last_trial_confidence_and_traces_no_tracking_reward_aligned_pk5.pkl")
      File2 = os.path.join(save_path, "all_trial_data_tail_contra_ipsi_last_trial_confidence_and_traces_no_tracking_choice_aligned_old_data_pk5.pkl")
      File3 = os.path.join(save_path, "all_trial_data_tail_contra_ipsi_last_trial_confidence_and_traces_no_tracking_choice_aligned_pk5.pkl")

(tail data was acquired in two batches that were processed separately)

2) *save_relevant_data_nacc.py* and *save_relevant_data_tail.py* that produces: processed_data_path/psychometric_data/all_tail_data_for_paper.csv and processed_data_path/psychometric_data/nacc_data_for_paper.csv

figure4/fig4LM.ipynb

Makes:

- Fig4 L Frechet distance example trajectories
- Fig4 M Frechet distance regression coefs
- ED Fig 10T difference in turn angle regression coefs
    - Require repro data or running:

      1) *figure3/format_data_for_trial_history_analysis.py*

      Which produces (one_per_mouse):
      bias_file = os.path.join(bias_path, 'pre_processing_bias_{}.pkl'.format(mouse))

      2) ED_figure6/movement_vs_trial_num_regression/ extract_movement_properties_and_APE_for_all_sessions.py

      Which produces files:
      save_out_folder = os.path.join(raw_tracking_path, mouse, date)
      movement_param_file = os.path.join(save_out_folder, 'APE_tracking{}_{}.pkl'.format(mouse, date))

      3) *figure4/add_movement_to_bias_data.py*

      Which produces files (one per mouse):
      bias_file = os.path.join(bias_path,
      'pre_processing_movement_bias_{}.pkl'.format(mouse))

figure4/reformat_psychometric_data.py

- Reformats the psychometric task data with the different perceptual uncertainties.
- The function that is called was originally written to combine tracking, photometry and bpod data but this feature was not eventually used so we call it with get_movement=False
- Requires *figure4/extract_aligned_traces_psychometric.py*
- Produces df with columns:
    - **APE peaks:** dopamine response - for tail it is APE but obviously for the VS it is not. This is the peak at time of choice for tail and cue for VS.
    - **trial type:** psychometric trial type (1-7 denotes different percentages of high (1) and low (7) tones)
    - **trial numbers:** trial number
    - **traces:** these are the photometry traces that correspond to the peaks (the event happens mid way through the trace)
    - **side:** ipsi or contra choice (on this trial)
    - **last trial type:** same as trial type but for the previous trial
    - **last outcome:** correct, incorrect or missed on last trial (0 incorrect, 1 correct, 3 missed)
    - **last choice:** same as side but for the previous trial
    - **next trial type:** same as trial type but for the next trial

    - **next outcome:** same as outcome but for the next trial
    - **next choice:** same as side but for the next trial
    - **stay or switch:** was the trial a stay or switch trial
    - **outcome:** correct, incorrect or missed (0 incorrect, 1 correct, 3 missed)
    - **APE quantile:** not used for this analysis but splits the dopamine responses into quartiles
    - **mouse:** mouse ID
    - **session:** date of session
    - **fiber side:** left or right (location of fiber in brain)
    - **norm APE:** dopamine response(see APE peaks) normalised per session
- Saves out df into these 3 files: save_path = os.path.join(processed_data_path, 'psychometric_data')
  File1 = os.path.join(save_path, "all_trial_data_Nacc_contra_ipsi_last_trial_confidence_and_traces_no_tracking_reward_aligned_pk5.pkl")
  File2 = os.path.join(save_path, "all_trial_data_tail_contra_ipsi_last_trial_confidence_and_traces_no_tracking_choice_aligned_old_data_pk5.pkl")
  File3 = os.path.join(save_path, "all_trial_data_tail_contra_ipsi_last_trial_confidence_and_traces_no_tracking_choice_aligned_pk5.pkl")
  (tail data was acquired in two batches that were processed separately)

figure4/save_relevant_data_nacc.py and figure4/save_relevant_data_tail.py

- Loads in files produced by *reformat_psychometric_data.py*
- Saves out columns: 'mouse', 'session', 'fiber side', 'trial numbers', 'trial type', 'side', 'outcome', 'last trial type', 'last choice', 'last outcome', 'next trial type', 'next choice', 'next outcome', 'norm APE', 'stay or switch' (gets rid of traces as they are large and take a long time to load and won’t be used for next analysis stage)
- Combines the two tail dfs
- Saves out dfs to:
  processed_data_path/psychometric_data/all_tail_data_for_paper.csv and
  processed_data_path/psychometric_data/nacc_data_for_paper.csv

figure4/add_movement_to_bias_data.py

- Loads in files produced by *figure3/format_data_for_trial_history_analysis.py* bias_file = os.path.join(bias_path, 'pre_processing_bias_{}.pkl'.format(mouse))
- Loads in movement parameter files produced by *ED_figure6/movement_vs_trial_num_regression/extract_movement_properties_and_APE_for_all_sessions.py*
- Adds movement parameters to photometry/behavior df
- Requires:
    - Running *ED_figure6/movement_vs_trial_num_regression/extract_movement_properties_and_APE_for_all_sessions.py* which creates the files:
      save_out_folder = os.path.join(raw_tracking_path, mouse, date)

movement_param_file = os.path.join(save_out_folder, 'APE_tracking{}_{}.pkl'.format(mouse, date))
This gets movement params for trials throughout learning.
    - bias_file = os.path.join(bias_path, 'pre_processing_bias_{}.pkl'.format(mouse))

figure4/extract_aligned_traces_psychometric.py

- Extracts aligned traces including correct and incorrect data at time of reward (only run for VS mice)
- Saves out:
  Folder = os.path.join(processed_data_path, 'for_psychometric', mouse)
  aligned_filename = experiment['mouse_id'] + '_' + experiment['date'] + '_' + 'aligned_traces_for_psychometric.p'

ED Figure 4

ED_figure4/ED_fig4CDE.py

Makes:

- ED_fig4 C: TS and VS photometry aligned to cue
- ED_fig4 D: TS and VS photometry aligned to movement
- ED_fig4 E: TS and VS photometry aligned to reward
    - Require repro data or running:
        1) figure2/per_mouse_average_traces.py which produces dir = os.path.join(processed_data_path, 'for_figure')
           file_name = 'group_data_avg_across_sessions_' + site +'_new_mice_added_with_cues.npz'

ED_figure4/ED_fig4F.py

Makes:

- ED_fig4F: heatmaps of TS and VS example mice example session aligned to cue
    - Requires repro data or running:
      1) *figure2/format_first_three_sessions_for_heatmap.py*
      saving_folder = os.path.join(processed_data_path, 'for_figure', example_mouse)
      aligned_filename = example_mouse + '_' + example_date + '_' + 'aligned_traces_for_fig.p'

ED_figure4/ED_fig4G.py

Makes:

- ED_fig4 G: Rise time box blot for VS and TS
    - Requires repro data or running:
      1) *figure2/format_first_three_sessions_for_heatmap.py*
      saving_folder = os.path.join(processed_data_path, 'for_figure', example_mouse)
      aligned_filename = example_mouse + '_' + example_date + '_' + 'aligned_traces_for_fig.p'
      2) *ED_figure4/get_peak_and_rise.py* (saves files in original_dir = os.path.join(processed_data_path,'for_figure'), filename='peak_times_and_time_to_slope_ipsi_and_contra_{}_with_means.npz'.format(site), where site is tail or nacc)

ED_figure4/ED_fig4HI.py

Makes:

- ED_fig4 H: Contra and ipsi choice photometry traces for TS separated by high and low cues
- ED_fig4 I: Contra and ipsi choice photometry traces for TS separated by high and low cues
    - Require repro data or running:
      1) *ED_figure4/get_average_trace_high_low_cues.py* produces files
      dir = os.path.join(processed_data_path, 'for_figure')
      file_name = 'group_data_avg_across_sessions_' + site +'_new_mice_added_high_low_cues_ipsi_contra.npz'
      Where site is nacc or tail

ED_figure4/ED_fig4JK.py

Makes:

- ED_fig4 J: return to centre ipsi contra photometry average traces TS example mouse
- ED_fig4 K: return to centre ipsi contra photometry average traces TS across mice
    - Require repro data or running:
      1) *ED_figure4/return_to_centre.py* with short_turns=True which produces files:
      save_dir = os.path.join(processed_data_path, 'return_to_centre', mouse)
      save_file = '{}_{}_return_to_centre_traces_aligned_to_movement_start_turn_ang_thresh_{}frame_window.npz'.format(mouse, date, timeframe)

ED_figure4/ED_fig4NO.py

Makes:

- ED_fig4 N: average traces for anterior to the tail mice aligned to choice
- ED_fig4 O: average traces for anterior to the tail mice aligned to reward
    - Require repro data or running:
    1) *figure2/per_mouse_average_traces.py*

ED_figure4/regression/ED_fig4L.py

Makes:

- ED_fig4 L: kernel regression for return to centre for TS (average across mice)
    - Requires repro data or running:
      1) *ED_figure4/return_to_centre.py* with short_turns=False which produces files:
      dlc_save_dir = os.path.join(processed_data_path, 'return_to_centre', mouse)
      time_stamp_save_file = '{}_{}_return_to_centre_movement_onset_times_300frame_window_long_turns.npz'.format(mouse, date)
      2) *figure2/kernel_regression/get_time_stamps_for_regression_all_cues_matched_trials.py* which produces files:
      events_folder = os.path.join(processed_data_path, mouse, 'linear_regression')
      aligned_filename = mouse + '_' + date + '_' + 'behavioural_events_with_no_rewards_all_cues_matched_trials.p'

3) *ED_figure4/regression/linear_regression_with_return_to_centre.py* which produces files:
data_dir = os.path.join(processed_data_path, mouse)
filename = mouse + '_' + date + '_' + 'linear_regression_kernels_return_to_centre_300frames_long_turns.p'
And var_exp_filename = os.path.join(processed_data_path,'_'.join(mouse_ids) + '_var_exp_with_return_to_centre_300frames_long_turns.p')

ED_figure4/regression/ED_fig4M.py

Makes:

- ED_fig4 M: full model variance explained for TS with different conditions
  - Requires repro data or running:
    1) *ED_figure4/regression/linear_regression_with_return_to_centre.py*
    2) *ED_figure4/regression/linear_regression_with_return_to_centre_trimmed_traces.py*
    3) *ED_figure4/regression/linear_regression_without_return_to_centre_trimmed_traces.py*
    4) *figure2/kernel_regression/linear_regression_all_sessions_different_shifts.py*

ED_figure4/get_average_trace_high_low_cues.py

- Separates traces by high and low cues
- Requires:
  - *figure2/format_first_three_sessions_for_heatmap.py*
    Which produces files
    data_dir = os.path.join(processed_data_path, 'for_figure', mouse)
    filename = mouse + '_' + date + '_' + 'aligned_traces_for_fig.p'
  - Also needs restructured data files

ED_figure4/get_peak_and_rise.py

- Calculates rise times for first 3 session data for VS and TS mice
- Requires:
  - *figure2/format_first_three_sessions_for_heatmap.py*
    Which produces files
    data_dir = os.path.join(processed_data_path, 'for_figure', mouse)
    filename = mouse + '_' + date + '_' + 'aligned_traces_for_fig.p'

ED_figure4/return_to_centre.py

short_turns=True for averages in ED_fig4J and K, and false for regression in panel L. This is to increase the number of events in the regression and the regression should be able to handle behavioral events overlapping in time better than just averaging traces from trials with different movement durations.

Requires:

file_path = os.path.join(raw_tracking_path,
'{}\\{}\\cameraDLC_resnet50_train_network_with_more_miceMar2shuffle1_800000.h5'.format(

```
            mouse, date)) or
file_path = os.path.join(old_raw_tracking_path,
'{}_{}DLC_resnet50_two_acMay10shuffle1_600000.h5'.format(mouse, date))
```

There were two batches of deep lab cut

Currently also requires the original raw daq AI.tdms files which are not included in the database but are available on request.

Saves out:

```
 if short_turns:
 save_file =
'{}_{}_return_to_centre_traces_aligned_to_movement_start_turn_ang_thresh_{}frame_window.npz'.format(mouse, date, timeframe)
        time_stamp_save_file =
'{}_{}_return_to_centre_movement_onset_times_{}frame_window.npz'.format(mouse, date, timeframe)
    else:
        save_file =
'{}_{}_return_to_centre_traces_aligned_to_movement_start_turn_ang_thresh_{}frame_window_long_turns.npz'.format(
            mouse, date, timeframe)
        time_stamp_save_file =
'{}_{}_return_to_centre_movement_onset_times_{}frame_window_long_turns.npz'.format(mouse, date, timeframe)
    np.savez(os.path.join(save_dir, save_file), contra_movement=contra_movement_traces,
ipsi_movement=ipsi_movement_traces)
```

ED_figure4/regression/linear_regression_with_return_to_centre.py

- Performs the kernel regression with return to center without trimmed traces
- Requires:
  1) *ED_figure4/return_to_centre.py* with short_turns=False which produces files:
  dlc_save_dir = os.path.join(processed_data_path, 'return_to_centre', mouse)
  time_stamp_save_file =
  '{}_{}_return_to_centre_movement_onset_times_300frame_window_long_turns.npz'.format(mouse, date)
  2)
  figure2/kernel_regression/get_time_stamps_for_regression_all_cues_matched_trials.py
  which produces files:
  events_folder = os.path.join(processed_data_path, mouse, 'linear_regression')
  aligned_filename = mouse + '_' + date + '_' +
  'behavioural_events_with_no_rewards_all_cues_matched_trials.p'

ED_figure4/regression/linear_regression_with_return_to_centre_trimmed_traces.py

- Performs the kernel regression with return to centre and with trimmed traces (trimmed traces = explained variance is only calculated on portions of the photometry trace where there are behavioral events)
- Requires:
  1) *ED_figure4/return_to_centre.py* with short_turns=False which produces files:
  dlc_save_dir = os.path.join(processed_data_path, 'return_to_centre', mouse)
  time_stamp_save_file = '{}_{}_return_to_centre_movement_onset_times_300frame_window_long_turns.npz'.format(mouse, date)
  2) *figure2/kernel_regression/get_time_stamps_for_regression_all_cues_matched_trials.py* which produces files:
  events_folder = os.path.join(processed_data_path, mouse, 'linear_regression')
  aligned_filename = mouse + '_' + date + '_' + 'behavioural_events_with_no_rewards_all_cues_matched_trials.p'

ED_figure4/regression/linear_regression_without_return_to_centre_trimmed_traces.py

- Performs the kernel regression without return to centre events but with trimmed traces (trimmed traces = explained variance is only calculated on portions of the photometry trace where there are behavioral events)
- Requires:
  1) *figure2/kernel_regression/get_time_stamps_for_regression_all_cues_matched_trials.py* which produces files:
  events_folder = os.path.join(processed_data_path, mouse, 'linear_regression')
  aligned_filename = mouse + '_' + date + '_' + 'behavioural_events_with_no_rewards_all_cues_matched_trials.p'

ED Figure 5

ED_figure5/silence/ED_fig5A.ipynb

Makes:

- ED_fig5 A: percentage complete trials silence experiment
  - Requires repro data or:
  - Original smoothed and demodulated photometry traces and restructured bpod files

ED_figure5/silence/ED_fig5BCD.ipynb

Makes:

- ED_fig5 B: Example TS mouse silence and tone photometry traces
- ED_fig5 C: Average across mice silence and tone photometry traces
- ED_fig5 D: Response size quantification silence vs tones
  - Require repro data or:

        - Original smoothed and demodulated photometry traces and restructured bpod files

ED_figure5/silence/ED_fig5E.py

Makes:

- ED_fig5 E: number of silence poking pairings before the silence experiment (this happens due to mice poking in the timeout period after incorrect trials)
    - Requires repro data or running:
      1) *ED_figure5/silence/get_silence_pokes_in_punishment.py*
      which produces file (which has been copied into the repro data)
      filename = os.path.join(processed_data_path, 'num_pokes_in_punishment.pkl')
        - Running this script requires the original bpod .mat files which are not included in the data set to keep the size down but are available on request.

ED_figure5/silence/ED_fig5FG.ipynb

Makes:

- ED_fig5 F: speed for tones and silence trials
- ED_fig5 G: turn angle for tones and silence trials
    - Require repro data or running:
      1) *ED_figure5/silence/extract_movement_properties_for_silence_exp.py*

ED_figure5/movement_analysis_out_of_task/ED_fig5IJ.ipynb

Makes:

- ED_fig5 I: Example TS mouse ipsi contra movement outside of task photometry traces
- ED_fig5 J:Average TS mice ipsi contra movement outside of task photometry traces
    - Require repro data or running:
      1)
      ED_figure5/movement_analysis_out_of_task/get_tracking_and_head_angle.py
      which produces files (there is a copy of these files in repro data):
      save_dir = os.path.join(running_in_box_dir, processed_data)
      data = np.load(os.path.join(save_dir,
      'preprocessed_speed_by_neurons_transformed_tracking_{}.npz'.format(mouse)),
      allow_pickle=True)
      file = open(os.path.join(save_dir,'tracking_data_{}.p'.format(mouse)), 'rb')

ED_figure5/movement_analysis_in_task/ED_fig5KLMN.py

Makes:

- ED_fig5 K: Mean TS dopamine (example mouse) split by APE response quartile (proof of good split)
- ED_fig5 L: Mean turn angle (example mouse) split by APE response quartile
- ED_fig5 M: Example correlation between APE response size and turn angle (split by APE quartile)
- ED_fig5 N: Regression coefficients (average per mouse across sessions) for analysis shown in example in panel M for TS and VS mice
    - Require repro data or running:

1) *ED_figure5/movement_analysis_in_task/extract_first_three_sessions_head_angles.py*
which produces the files:
Directory = reproduce_figures_path/ED_fig5/movement_inside_task
TS mice

1) contra_APE_tracking_first_3_sessions_SNL_photo16_SNL_photo17_SNL_photo18_SNL_photo21_SNL_photo22_SNL_photo26_with_shuffles.csv
2) contra_APE_tracking_first_3_sessions_SNL_photo16_SNL_photo17_SNL_photo18_SNL_photo21_SNL_photo22_SNL_photo26.csv

VS mice

1) contra_APE_tracking_first_3_sessions_SNL_photo28_SNL_photo30_SNL_photo31_SNL_photo32_SNL_photo33_SNL_photo34_SNL_photo35_with_shuffles.csv
2) contra_APE_tracking_first_3_sessions_SNL_photo28_SNL_photo30_SNL_photo31_SNL_photo32_SNL_photo33_SNL_photo34_SNL_photo35.csv

As well as the files:
save_out_folder = os.path.join(post_processed_tracking_data_path, mouse)
movement_param_file = os.path.join(save_out_folder, 'contra_APE_tracking{}_{}.pkl'.format(mouse, date))
(these are one per mouse and session, the example plots are from example_mouse='SNL_photo26', example_date='20200810')

ED_figure5/silence/get_silence_pokes_in_punishment.py

- Gets the number of silence-choice pairings experienced by the mice before the silence experiment. There is a timeout 'punishment' used in most of training during which the mice still try to make centre, and then choice pokes.
- which produces file (which has been copied into the repro data)
  filename = os.path.join(processed_data_path, 'num_pokes_in_punishment.pkl')
    - Running this script requires the original bpod .mat files which are not included in the data set to keep the dataset file size down but are available on request.

ED_figure5/silence/extract_movement_properties_for_silence_exp.py

- Gets tracking data for choice movements and formats it into a df with photometry and behavioral data for subsequent analysis
- **Utils.tracking_analysis.velocity_utils.format_movement_params_into_df and format_tracking_data_and_photometry** are the main functions used to extract and format movement, photometry and behavioral data, see them for more details
- Requires:
  1) having run *data_preprocessing/extract_aligned_traces_general.py* on the dates for silence experiments (these files are provided in processed_data_path/mouse and have format saving_folder = os.path.join(processed_data_path, mouse)

aligned_filename = os.path.join(saving_folder, mouse + '_' + date + '_' + 'aligned_traces.p'))
2) Tracking files: filename = os.path.join(raw_tracking_path, '{}\\{}\\cameraDLC_resnet50_train_network_with_more_miceMar2shuffle1_800000.h5'.format(mouse, date))
3) restructured bpod data (provided in full data set)

- And camera triggers (needs raw AI files)

ED_figure5/movement_analysis_out_of_task/get_tracking_and_head_angle.py

- Transforms the tracking data (the camera is at an angle for these experiments so transform is needed to get tracking in real coordinates)
- Calculates head_angular_velocity, head_ang_accel, speed, move_dir, acceleration, head_angles
- Saves out data (with photometry) to:
  save_dir = os.path.join(running_in_box_dir, 'processed_data')
  movement_and_photometry _file = os.path.join(save_dir, 'preprocessed_speed_by_neurons_transformed_tracking_{}.npz'.format(mouse)) and tracking_coords_file = os.path.join(save_dir, 'tracking_data_{}.p'.format(mouse)
- Requires:
  1) photometry files:
  (smoothed demodulated files, in os.path.join(running_in_box_dir, 'processed_data', mouse))
  2) dlc output files:
  file_path = os.path.join(running_in_box_tracking_dir, '{}\\{}\\{}_cameraDLC_resnet50_heading_angleMar23shuffle1_1030000.h5'.format(mouse, date, mouse))

ED_figure5/movement_analysis_in_task/extract_first_three_sessions_head_angles.py

- Gets tracking data for first three sessions (the sessions used in figure 2)
- Finds movement properties for choice movements and saves these to a df
- Calculates APE quartiles and classifies trials into quartiles
- Performs the regression in ED figure 5 M and N and saves out coefs
- Performs a shuffle of APE quartile label on a copy of that df
- Saves out coefs of regression on shuffled data too
- **Utils.tracking_analysis.velocity_utils.format_movement_params_into_df and format_tracking_data_and_photometry** are the main functions used to extract and format movement, photometry and behavioral data, see them for more details
- Produces the files:
  Directory = reproduce_figures_path/ED_fig5/movement_inside_task
  TS mice
  3) contra_APE_tracking_first_3_sessions_SNL_photo16_SNL_photo17_SNL_photo18_SNL_photo21_SNL_photo22_SNL_photo26_with_shuffles.csv
  4) contra_APE_tracking_first_3_sessions_SNL_photo16_SNL_photo17_SNL_photo18_SNL_photo21_SNL_photo22_SNL_photo26.csv
  VS mice

3) contra_APE_tracking_first_3_sessions_SNL_photo28_SNL_photo30_SNL_photo31_SNL_photo32_SNL_photo33_SNL_photo34_SNL_photo35_with_shuffles.csv
4) contra_APE_tracking_first_3_sessions_SNL_photo28_SNL_photo30_SNL_photo31_SNL_photo32_SNL_photo33_SNL_photo34_SNL_photo35.csv

As well as the files:
save_out_folder = os.path.join(post_processed_tracking_data_path, mouse)
movement_param_file = os.path.join(save_out_folder, 'contra_APE_tracking{}_{}.pkl'.format(mouse, date))

ED Figure 6

ED_figure6/movement_vs_trial_num_regression/ED_fig6CtoM.ipynb
Makes:

- ED fig6 C & D: correlating movement parameters (throughout training in C and D) with TS dopamine example mouse
- ED fig6 E: correlating movement parameters in first 3 sessions with TS dopamine across mice (single trial version of quartile analysis in ED fig 5KLMN)
- ED fig6 F & G: correlating movement parameters over training (binned similarly to dopamine responses shown in figure 3) for an example mouse.
- ED fig6 H: same as for E but for all sessions in all of training
- ED fig6 I-L: Analysis to show log trial number predicts TS dopamine even when movement parameter-related signal has been subtracted. Regression model predicting dopamine response on a single trial level for all trials over training from speed and turn angle. The predicted signal is subtracted from the actual dopamine response per trial. The residual DA response is then plotted and log trial number is used to predict the residuals. Log trial number coefficients for all TS mice shown in panel L.
- ED fig6 M: full model with all speed, turn angle and log trial number explained variance by each regressor for all mice
    - Require repro data or running:
      1) *ED_figure6/movement_vs_trial_num_regression/extract_movement_properties_and_APE_for_all_sessions.py*
        - This file requires aligned_data files produced by **figure3/save_out_aligned_traces_for_session.py** (provided in full data set)

      2) *ED_figure6/movement_vs_trial_num_regression/first_3_sessions_single_trial_movement_APE_correlation.py*
      3) *ED_figure6/movement_vs_trial_num_regression/get_regression_slopes_for_turn_angle_speed_trial_number_vs_APE.py*

ED_figure6/ED_fig6N.ipynb

Makes:

- ED fig6 N: Between sessions vs across sessions dopamine responses
    - Requires repro data or running:
    - *ED_figure6/movement_vs_trial_num_regression/get_regression_slopes_for_turn_angle_speed_trial_number_vs_APE.py*
    - Loads files df_save_dir = r'{}{}\turn_angle_over_time'.format(processed_data_path, mouse)
    - df_save_file = os.path.join(df_save_dir, 'movement_params_all_trials_vs_APE_{}.pkl'.format(mouse))
    - These files are the ones used for the movement vs trial number regression but they also just have the dopamine responses per trial and session in a nice format, which is why they are loaded here.

ED_figure6/state_change/ED_fig6ORST.ipynb

Makes:

- ED fig6ORST: Behavioral measures during the state change (white noise) experiment
    - Requires: only restructured data files

ED_figure6/state_change/ED_fig6PQ.ipynb

Makes:

- ED fig6 PQ: movement parameters on tone and white noise trials
    - Requires repro data or running:

      1*) ED_figure6/state_change/extract_movement_properties_for_state_change.py*
        - This requires the aligned_traces from these sessions (made by running *figure3/save_out_aligned_traces_for_session.py* on for given mice and dates). However, the photometry data is not actually used in these plots so these are not included in the data set). They are just used to get the trial numbers that go into the photometry plots. If you want to make the aligned data files, you can run *figure3/save_out_aligned_traces_for_session.py* for the given mice and dates for the state change experiment (as shown in the experiment_record csv).

      2) dlc output files: file_path = os.path.join(raw_tracking_path, '{}\\{}\\cameraDLC_resnet50_train_network_with_more_miceMar2shuffle1_800000.h5'.format(mouse, date))

ED_figure6/value_change/ED_fig6UVWXY.ipynb

Makes:

- ED fig6UVWXY: Behavioral measures during the value change experiment (reward amount changes in blocks to change cue value)
    - Requires: only restructured data files

ED_figure6/movement_vs_trial_num_regression/extract_movement_properties_and_APE_for_all_sessions.py

- Reformats data shown in figure 3 into a dataframe with movement parameters, trial numbers and dopamine responses

- Requires:
  1) The aligned_traces from these sessions (made by running *figure3/save_out_aligned_traces_for_session.py*) which are included in the full data set in processed_data_path/mouse
  2) dlc output file: file_path = os.path.join(raw_tracking_path, '{}\\{}\\cameraDLC_resnet50_train_network_with_more_miceMar2shuffle1_800000.h5'.format(mouse, date))
- Saves out df to:
  save_out_folder = os.path.join(raw_tracking_path, mouse, date)
  movement_param_file = os.path.join(save_out_folder, 'APE_tracking{}_{}.pkl'.format(mouse, date))

ED_figure6/movement_vs_trial_num_regression/first_3_sessions_single_trial_movement_APE_correlation.py

- Creates dfs (if they don’t exist) for first 3 sessions (data shown in figure 2) with photometry and choice movement parameters (including speed and turn angle - which is called 'abs fitted max cumsum ang vel')
- Performs a single trial regression for effect for speed and turn angle on TS choice dopamine response
- Saves out (if they don’t exist):
  save_out_folder = os.path.join(raw_tracking_path, mouse, date)
  movement_param_file = os.path.join(save_out_folder, 'APE_tracking{}_{}.pkl'.format(mouse, date))
- Also saves out:
  all_mice_df_save_dir = os.path.join(processed_data_path, 'turn_angle_over_time')
  all_mice_df_save_file = os.path.join(all_mice_df_save_dir, 'movement_params_first_3_sessions_vs_APE_regression_coefs_and_pvals.pkl')

ED_figure6/movement_vs_trial_num_regression/get_regression_slopes_for_turn_angle_speed_trial_number_vs_APE.py

- Gets movement for all trials in change over time plots in figure3 and then performs regression including movement and log trial number
- Either reads in dfs with photometry and movement and trial number made by *ED_figure6/movement_vs_trial_num_regression/extract_movement_properties_and_APE_for_all_sessions.py,* or makes them and save out here:
  save_out_folder = os.path.join(raw_tracking_path, mouse, date)
  movement_param_file = os.path.join(save_out_folder, 'APE_tracking{}_{}.pkl'.format(mouse, date))
- Combines these for all sessions (if not existing already) into a df (one per mouse) with all photometry, movement and trial number data and save out here:
  df_save_dir = os.path.join(processed_data_path, mouse,'turn_angle_over_time')
  df_save_file = os.path.join(df_save_dir, 'movement_params_all_trials_vs_APE_{}.pkl'.format(mouse))
- Performs a regression predicting APE responses (TS dopamine at time of choice) from a) speed, b) turn angle, c) trial number once movement has been accounted for
- Calculates r-squared for a full model with speed, turn angle an log trial number in it

- Saves out coefs, p-values and r-squared values to:
  all_mice_df_save_dir = os.path.join(processed_data_path, 'turn_angle_over_time')
  all_mice_df_save_file = os.path.join(all_mice_df_save_dir, 'movement_params_all_trials_vs_APE_regression_coefs_pvals_r2_and_trial_num_correlation_and_full_model.pkl')

ED_figure6/state_change/extract_movement_properties_for_state_change.py

- Gets movement properties for choice movements during silence experiment
- This requires the aligned_traces from these sessions (made by running *figure3/save_out_aligned_traces_for_session.py* on for given mice and dates). However, the photometry data is not actually used in these plots so these are not included in the data set). They are just used to get the trial numbers that go into the photometry plots. If you want to make the aligned data files, you can run *figure3/save_out_aligned_traces_for_session.py* for the given mice and dates for the state change experiment (as shown in the experiment_record csv).
- Saves out:
  save_out_folder = os.path.join(raw_tracking_path, mouse, date)
  movement_param_file = os.path.join(save_out_folder, 'APE_tracking{}_{}.pkl'.format(mouse, date))

ED Figure 7

ED_figure7/ED_fig7CDEF.py

Makes:

- ED fig7 CDEF: example and average mice data for large reward omission experiment
  - Requires repro data or running:
    1) *ED_figure7/reformat_unexpected_large_rewards_omissions.py*
    2) ED_figure7/downsample_traces_large_rewards_omissions.py

ED_figure7/ED_fig7IJKL.py

Makes:

- ED fig7 IJKL: example and average mice data for value change (block wise change of reward amount)
  - Requires repro data or running:
    1) *ED_figure7/reformat_value_change.py*
    2) *ED_figure7/downsample_traces_value_change.py*

ED_figure7/reformat_unexpected_large_rewards_omissions.py

- Gets data at time of outcome for unexpected large rewards and omissions experiments
- Requires:
  - Smoothed, demodulated photometry data and restructured bpod data
- Saves out:
  processed_data_dir = os.path.join(processed_data_path, 'large_rewards_omissions_data')
  processed_data_file = os.path.join(processed_data_dir, 'all_{}_reward_change_data.pkl'.format(site))
- Data format will be very similar to that of reward_data for aligned_traces files (see *figure2/format_first_three_sessions_for_heatmap.py)*

ED_figure7/reformat_value_change.py

- Gets data at time of cue (VS) and choice (TS) for values change experiment
- Requires:
    - Smoothed, demodulated photometry data and restructured bpod data
- Saves out:
  processed_data_dir = os.path.join(processed_data_path, 'value_change_data')
  block_data_file = os.path.join(processed_data_dir, 'value_change' + '_' + site + '.p')
- Data format will be very similar to that of choice_data or cue_data for aligned_traces files (see *figure2/format_first_three_sessions_for_heatmap.py)*

ED_figure7/downsample_traces_large_rewards_omissions.py

- Calculates response size to large rewards, normal rewards and omissions
- Downsamples and clips traces for example mouse for file size reasons (and saves to repro data)
- Requires:
    - Running *ED_figure7/reformat_unexpected_large_rewards_omissions.py*
- Saves out:
  repro_path = os.path.join(reproduce_figures_path, 'ED_fig7', 'omissions_large_rewards')
    1) repro_file = os.path.join(repro_path, f'omissions_large_rewards_downsampled_traces_peaks_{site}.pkl')

  mouse_name = large_reward_omission_example_mice[site]
    2) repro_example_file = os.path.join(repro_path, f'omissions_large_rewards_downsampled_traces_example_{site}_{mouse_name }.pkl')

ED_figure7/downsample_traces_value_change.py

- Gets dopamine response at time of cue or choice
- Downsamples and clips traces for example mouse for file size reasons (and saves to repro data)
- Requires:
    - Running *ED_figure7/reformat_value_change.py*
- Saves out:
  repro_path = os.path.join(reproduce_figures_path, 'ED_fig7', 'value_change')
    1) repro_file = os.path.join(repro_path, f'value_change_downsampled_traces_peaks_{site}.pkl')

  mouse_name = value_change_example_mice[site]
    2) repro_example_file = os.path.join(repro_path, f'value_change_downsampled_traces_example_{site}_{mouse_name}.pkl')