Analysis pipeline

Colizoli, O., de Gee, J. W., Urai, A. E. & Donner, T. H.

Task-evoked pupil responses reflect internal belief states. *Scientific Reports* 8, 13702 (2018).

Note: In this manual, only files and functions that are important for the current analysis are described. It is not a complete summary of the entire pipeline. For questions, email: olympia.colizoli@gmail.com

Matlab psychtoolbox 2AFC task output

For each run, there are the following files:

- **Behav_.txt** columns:
    - coherence (smaller is more difficult)
    - hemifield (-1 is left, 1 is right)
    - correct response (90 up, 270 down)
    - response (90 up, 270 down)
    - accuracy (1 correct, 0 incorrect)
    - RT (seconds)
    - arousal condition (1 is tone, 2 is white noise)
- **Phases_.txt** columns:
- Timestamps (s) with respect to BEGIN variable = start of run (results.outputPhase-BEGIN)
    - phase 1 - baseline period ([.5,5] random)
    - phase 2 - pre stim onset period (wait for trigger) (2 s max)
    - phase 3 - stimulus on period (750 ms)
    - phase 4 - stim offset -> response (max 2250 ms)
    - phase 5 - response -> feedback ([3.5 5.5 7.5 9.5 11.5] uniform)
    - phase 6 - ITI after feedback ([3.5 5.5 7.5 9.5 11.5] uniform)
- **Geld_.txt** amount of money earned on that run (averaged with previous runs)

% Correct	<=50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
Reward	€ 0	€ 1	€ 2	€ 3	€ 4	€ 5	€ 6	€ 7	€ 8	€ 9	€ 10

- **.MAT** files variables:
    - 'setup', 'window1', 'stim', 'dots', 'fix', 'results', 'audio', 'sound', 'flip'
    - Due to human error, the distance and screen width were incorrect in the code presented during the experiments. Therefore, the following variables in the .MAT files are incorrect:
        - window1.distance
        - window1.width
        - dots.radius
        - dots.innerspace
        - dots.speed
        - dots.size
        - dots.density
    - The correct parameters (reported in the methods) were calculated using the script **CalculateParameters_2AFC.m**
- **Folders containing EDF files per run**

File Structure

home > bin > edf2asc (converting EDF file formats)
raw > main_task (experiment data)
raw > irf_task (control experiment 1 data)
raw > control_task (control experiment 2)
home > analysis (scripts)

- ***first_level.py*** (list of raw files, preprocessing data)
  - ***pupil_preprocessing.py*** (called by *first_level.py* for pupil subject level)
- ***second_level.py*** (group level analysis)
  - ***pupil_data_analysis.py*** (called by *second_level.py* for group level pupil
- Functions that cannot be found in any of these files are probably in the directory: analysis/other_scripts.

home > data (output from the above files for main experiment)
home > data > pupil_2AFC (output from the pupil preprocessing. MSG files [in raw folder] in here have been edited with a Matlab function to fit the necessary format for the pipeline).
home > measure_irf > pupil_measureIRF (output from the pupil preprocessing. MSG files [in raw folder] in here have been edited with a Matlab function to fit the necessary format for the pipeline).
home > feedback > pupil_feedback (output from the pupil preprocessing. MSG files OK)

first_level.py

(list of raw files, preprocessing data)
Note: behavioral variables of interest are extracted from EDF files

runWholeSession(rDA, session)

- the session class here refers to ***session.py***, see analysis/other_scripts folder.

pupilPreprocessSession = pupil_preprocessing.pupilPreprocessSession()
EDF files are converted to MSG, GAZE using **edf2asc**.

- **.import_raw_data()**
  - grabs EDF files and copies them into the raw folder in the data folder
- **.convert_edfs()**
  - converts edfs to gaze and msg files
- **.delete_hdf5()**
  - When need to rerun the pupil preprocessing, always delete exisiting HDF5 file (this is the structure holding the varaibles extracted from the EDF fies)

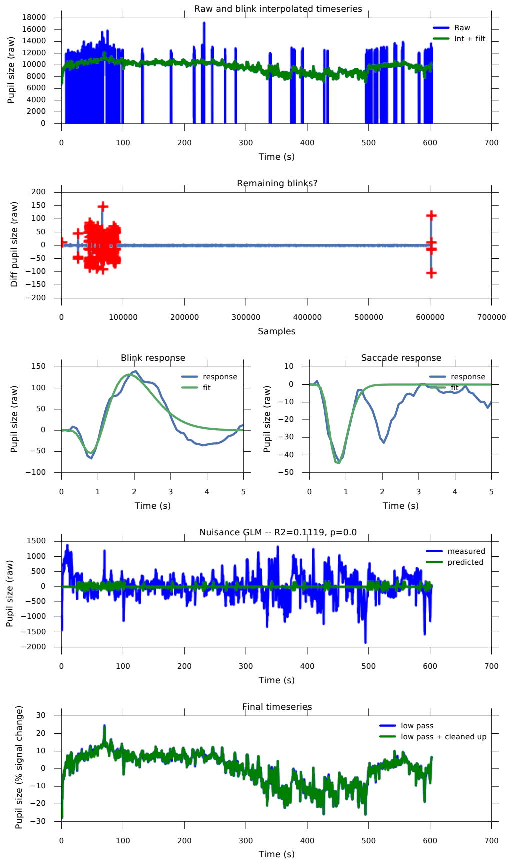
MSG files need to be reformatted in the following way to run in the python pipeline:
HERE is when the 3 Matlab scripts should be run, reformatting the MSG files before proceeding. Ideally, these MSG files should be corrected once, in the raw folder, before setupFiles is run, however, they were fixed in 2 steps as the analysis progressed.
In directory pupil_2AFC, run in the following order:

1) **MSG_FindReplace.m**
   a. Add colon to lines containing parameters
   b. Trial number must start with 0, not 1
   c. Missing a line before phase 1 for 'trial started at'
2) **MSG_insert_parameters.m**
   a. Adds parameter for easy/difficult trials
      i. Difficult = 1 (cohlevel1)
      ii. Easy = 2 (cohlevel2)
   b. Replaces coherence and RT values with non-scientific notation

MEASURE IRF experiment

1) **MeasureIRF_MSG_FindReplace.m**
   - **a.** Add trial / start stop and numbers (start with 0, not 1)
   - **b.** Add parameters
   - **c.** Add phases 1 for sound, 2 for response

- **.import_all_data(aliases)**
  - linear interpolation around blinks (time window 0.1 s before until 0.1 s after blink)
  - bandpass filtering (third-order Butterworth, passband: 0.01-6Hz)
  - removing responses to blink and saccade events using linear regression (responses estimated by deconvolution)
  - convert to percent signal change with respect to the mean of the pupil time series
  - Calls **HDFOperator**, which calls the **EyeSignalOperator:** self.ho.edf_gaze_data_to_hdf()
  - Produces plots “pupil_preprocess_2AFC_2AFC_1_1_0_L.pdf”
  - output in Pupil_2AFC > subj > processed
  - Adds preprocessed pupil data to subjects’ HDF5 file



- **.process_runs(process_runs()**
  - Adds parameters of interest to data structure
  - calls compute_omission_indices(): can specify extra omission conditions (missed trials)
  - trial_params(): calculates number of omissions, blinks, saccades, velocity etc. and adds as parameters to data structure
- Define phase indices (**fMRI_2AFC.m**):
  - self.baseline_times = ['trial_phase_index'] == **1**

    - self.cue_times = ['trial_phase_index'] == **3**
    - self.choice_times = ['trial_phase_index'] == **5**
    - self.feedback_times = ['trial_phase_index'] == **6**
- **.process_across_runs()**
    - creates scalar values for time periods of interest:
    - use the bandpassed pupil
    - **pupil_b** is pupil baseline before cue [-0.5,0]
    - **pupil_b_feed** is pupil baseline before feedback [-0.5,0]
    - **pupil_resp_d_clean** is pupil dilation locked to choice [3,6]
    - **pupil_d_feed_clean** is pupil dilation locked to feedback [3,6]

across.py

(group level analysis, pupil)

Data stored in CSV format (or numpy arrays) in each dataframe function

data_frames

- **anova_output**
- **before_omissions**
- **pupil**

higherLevel = **pupil_data_analysis**.**higherLevel**()

- **.dataframe_pupil_subjects()**
    - Outputs one dataframe for all trials for all subjects (pupil data only)
    - Extra omissions defined here
    - Regresses RT out of pupil data (scalars)
    - Should be run once without omissions excluded (saved in before_omissions folder)
- **.dataframe_processITI()**
    - ITIs and other condition information that was not sent to the eyetracker is added to the pupil data frame
    - To get the ITIs, need to run the matlab script: **process_behavior_local.m**
- **.dataframe_process_motionEnergy()**
    - Adds motion energy values to the pupil data frame
    - See motion energy scripts in other_scripts (matlab)
- **.dataframe_pupil_event_related_subjects()**
    - Time series data: per event of interest (resp-locked or feed-locked), outputs one numpy array containing all trials for all subject
- **.dataframe_pupil_higher()**
    - Creates higher level dataframes for ANOVA input
- **.dataframe_pupil_event_related_higher()**
    - Creates event related dataframes for the subjects' means
- **dataframe_pupil_event_related_higher_regressRT()**
    - removes RT from the time courses via linear regression
- **dataframe_pupil_event_related_higher_regressME()**
    - regression of motion energy onto pupil, saves betas

Anova

- **.run_anova_pupil()**
    - Runs a repeated measures ANOVA for all pupil DVs and RT for all conditions
    - Outputs an HTML table of results and CSV file containing statistics

figures_ScientificReports.py
(model, statistics, figures)

f = figures.figureClass()

- **.sdt_models()**
- **.plot_sdt_models()**
  - *Figure 2:*
  - Model predictions (group level)
- **.sdt_models_subjects()**
- **.plot_sdt_models_subject_predictions()**
  - *Supplementary Figure 5*
  - Model predictions (single subjects)
- **.urai_feedback_pupil_median()**
  - *Supplementary Figure 4*
- **.behavior()**
  - *Supplementary Figure 1*
- **.pupil_results()**
  - *Figure 3*
- **.RT_pupil_correlation()**
  - Correlations between RT and pupil in pre-feedback interval
- **.plot_sdt_models_fits_coherence()**
  - *Figure 5a,b*
  - *Note correction Figure 5b (corr and p-values)*
- **.plot_sdt_models_fits_motion_energy()**
  - *Figure 5c,d*
- **.control_exp()**
  - *Supplementary Figure 2*
- **.pupil_results_motion_energy()**
  - *Figure 4b*
- **.pupil_event_related_betas ()**
  - *Figure 4a*
- **.regress_pupil_motion_energy ()**
  - Single trial regression results