💻

Data Processing Pipeline

Personal data

Data source	REDCap
Main script	`PersonalData_main.r`
Functions script	`PersonalData_functions.r`
Input file	[ID]_screening.csv
Output file	[ID]_personal-data.csv

Procedure

1. Verify exclusion criteria:
    a. no consent → error
    b. any exclusion criteria questions ‘maybe’ → check
    c. AUDIT score: if > 15 → error
    d. PSQI score: if > 4 → error
    e. uMCTQ score: if < 01:30 or > 06:00 → error

Output variables

- **age**: number in years
- **sex**: 1=female, 2=male, 3=non-binary, 0=prefer not to say
- **gender_match**: 1=matching, 0=non-matching / no info
- **handedness**: 1=right-handed, 2=ambidextrous, 3=left-handed
- **education**: 0="Early childhood education", 1="Primary education", 2="Lower secondary education", 3="Upper secondary education", 4="Post-secondary, non-

tertiary education", 5="Short-cycle tertiary education", 6="Bachelors or equivalent level", 7="Masters or equivalent level", 8="Doctoral or equivalent level”

- **time_lab_arrival:** HHMMSS
- **uMCTQ_score**: HHMMSS
- **PSQI_score**: number

Sleep data

Data sources	GSQS in REDCap; EEG recordings
Main scripts	`EEGData_main_control.py EEGData_main_experimental.py`
Functions script	`EEGData_functions.py`
Input files	[ID]_session[02/03]_raw-EEG.eeg [ID]_sleep-quality.csv
Output files	[ID]_[control/experimental]_sleep-data.csv

Procedure

1. Pre-process EEG data:
    a. Channel subset: C3, C4, LEOG, REOG, LEMG, REMG
    b. Re-reference to average of A1+A2
    c. Resample to 100 Hz
    d. Bandpass filter: 0.1 - 45 Hz
2. Run sleep staging algorithm (YASA):
    a. Right-sided channels as inputs (C4, REOG, REMG), with left-sided ones as back-up, correspondingly (C3, LEOG, LEMG)
    b. Additional inputs, if available: age, sex
    c. Outputs: spectrogram, hypnogram, sleep metrics

Output variables

- SOL: sleep onset latency (min) (for control condition, subtract 10 min, during which participants are asked to stay awake by design)
- TST: total sleep time (min)
- WASO: wake after sleep onset (min)
- %N1, %N2, %N3, %REM: time spent in each sleep stage as percentage of TST (%)
- GSQS_sum: GSQS sum score (extracted from REDCap file)

EEG data

Data sources	EEG recordings
Main scripts	`EEGData_main_control.py EEGData_main_experimental.py`
Functions script	`EEGData_functions.py`
Input files	[ID]_session[01/02/03]_raw-EEG.edf [ID]_session[01/03]_annotations.edf
Output files	[ID]_[control/experimental]_PSD-output-metrics.csv [ID]_[control/experimental]_SSVEP-output-metrics.csv
Outputs for group plots	[ID]_[control/experimental]_PSD-output-spectra.csv [ID]_[control/experimental]_SSVEP-output-curves.csv

Procedure

1. Load raw data in EDF format
    a. Channel subset: PO3, PO4, POz, O1, O2, Oz
    b. Re-reference to average of A1+A2
    c. Mark bad channels (previously identified visually)
2. Display recording information as sanity check
    a. recording duration, sampling rate, filters, channels
3. Import stimulation triggers from separate EDF files, sessions 01 and 03 (necessary because of conversion from Nihon Kohden format to EDF)
    a. For control condition, merge the two trigger sets (for calculation of PSD and SSVEP metrics in the absence of stimulation, to compare with experimental

condition)

   b. For experimental condition, keep sessions separate

4. Run YASA algorithm for automated sleep scoring
   a. Stages: 0=wake, 1=N1, 2=N2, 3=N3, 4=REM (N1 not used for analyses)
5. Create 30 sec epochs assigned to stages
   a. Reject epochs scored at < 50 % confidence (estimated by YASA)
   b. Reject epochs with < 25 / 30 sec of stimulation
   c. Reject epochs with > 1 mV peak to peak amplitude (likely to include movement artifacts)
6. PSD analysis by stage
   a. Compute a PSD spectrum for each 30 sec epoch using a hamming window
   b. Compute SNR spectra with a kernel using the mean of neighbouring frequencies as ‘noise’
   c. Average PSD and SNR spectra across epochs and channels; transform average PSD spectrum to dB
   d. Determine exact frequency with highest SNR within [39.5-40.5 Hz] (to allow for some degree of variability / potential flicker timing inconsistencies)
   e. For this frequency, extract the PSD value (dB) and SNR value
7. SSVEP analysis per stage
   a. Divide data into 25 ms segments following a trigger
   b. Reject segments with a peak-to-trough amplitude > 100 uV
   c. Average segments and compute peak-to-trough amplitude
   d. Compute SNR by shuffling the data points of the segments, then computing the peak-to-peak amplitude of the averaged shuffled “SSVEP”; repeat over 100 iterations → SNR = true amplitude / shuffled amplitude

Output variables

- PSD_ntrials_[W/N2/N3/REM]: Nr. of PSD trials included per stage (30 s epochs)

- PSD_40Hz_[W/N2/N3/REM]: Mean 40 Hz power (dB) per stage
- PSD_SNR_[W/N2/N3/REM]: SNR of PSD value at 40 Hz per stage
- SSVEP_ntrials_[W/N2/N3/REM]: Nr. of SSVEP trials included per stage (25 ms segments)
- SSVEP_PTA_[W/N2/N3/REM]: Mean SSVEP peak-to-trough amplitude (uV) per stage
- SSVEP_SNR_[W/N2/N3/REM]: SNR of SSVEP per stage
- For group plots: average PSD spectra, PSD-SNR spectra, and SSVEP curves per stage