Analysis overview
Overview

- Analysis flow chart for all major scripts
- Figure plotting guide
- Pre-processing
- Physiology scripts
- Aligning events
Before any of the scripts will work, go into analysis->load_root.m and change…
- Folder directory
- EEGLAB directory (you must have this installed somewhere to run some operations)
- directory for R script outputs

Subject numbers
Separate pair learning: 1-30
Competitive pair learning: 31-60
Analysis flow chart

- Scripts in boxes require more explanation - see scripts themselves

```
Import data

StepOne

-sleepInt.set

subject notes logged in preprocessingLog

StepTwo

via sleepSMG

stagesJWA.mat

TIB.set

TIB_5s_epoch.set

reject.mat

StepThree

via 5s rejection

StepFour

CheckSoundStage aggregates stages and where cues fall

CheckBootstraps to control for multiple comparisons in time domain

UnivariateStuff to analyze time-locked physiology outputs

Folder outputs

Df4D

SO, spindle, SOphase scripts

StepsMeta runs all in yellow

XSubBehav / CompConds for across-subject behavioral analyses / plots

Behavior: run StepZero first

QuadrantSimulations for simulations on quadrant analysis

MovingAverage2D for item-competitor analyses
```
Summary of the .m files where you can find code for each plot:
Fig 1B: CompLinReg, uncomment lines at bottom
Fig 2A: CompConds
Fig 2B: CompConds
Fig 2C/D: XSubBehav, set vers=1 (SPL) or 2 (CPL)
Fig 3A: XSubBehav, set vers=1
Fig 3B: QuadrantSimulations, set vers=1
Fig 4B-G: MovingAverage2D, set vers=2
Fig 4H: XSubBehav, set vers=2
Fig 5A, left: UnivariateStuff, set vers=5, memversion=4, paredown=0
Fig 5B, left: UnivariateStuff, set vers=5, memversion=6, paredown=2
Fig 5C, left: UnivariateStuff, set vers=5, memversion=5, paredown=0
Fig 5, right: UnivariateStuff on vers=1 and vers=2, then set inty=1 and run middle section
Fig 6A, left: UnivariateStuff, set vers=5, memversion=6, paredown=2
Fig 6B, left: UnivariateStuff, set vers=5, memversion=5, paredown=0
Fig 6, right: UnivariateStuff on vers=1 and vers=2, then set inty=1 and run middle section
Table 1: CheckSoundStage, grab ‘exp’ variable
**Importing data**

- **Use StepOne**
  - Load files, downsample, select relevant channels
  - **Note:** two differences for Biosemi (from Neuroscan)
    - Electrode location files
    - Usually larger – need to load in steps and combine
    - See ‘StepOneBiosemi’
  - Best practice: save notes about each subject at the bottom in preprocessingLog
Event structure - note each of the variables

‘EEG.history’ → scripting

- Not perfect, and doesn’t log everything, but logs many things to run in batch scripts
- Best practices: script everything you can and only go to the GUI when necessary. This saves you time and also ensures any errors are (at least) not due to negligence on individual subjects.
I do two things:

- Graph spectrogram for every channel
  - `figure;pop_spectopo(EEG,1,[0 EEG.pnts], 'EEG','freqrange',[2 64], 'electrodes','off');`
- Skip through the recording every 500s to see if any channels go in and out for a decent length of time

If you run ICA, you must do this first WITHOUT the channels you need to interpolate.

- For instance, if you need to interpolate channel 57, use
  ```matlab
  'channelstoinclude=[1:56 58:68]; EEG = pop_runica(EEG,'icatype','runica','dataset',1,'chanind',channelstoinclude,'extended',1);
  ```
- Then reject eye blink components
- Then use `pop_interp` (example: electrodes 8 & 10)
  ```matlab
  EEG = pop_interp(EEG, [8 10], 'spherical');
  ```
FYI - I use the script, StepsMeta.m, to do interpolation and generally just run the other ‘Step’ scripts between the other tasks. It controls everything in the yellow box on the flow chart.
Use StepTwo

For CNTs:
- EEG & EOG files filtered 0.4-50 Hz
- EMG filtered 10-59 Hz (Delphine low-passed at 70)

Then data are re-referenced to linked mastoids and separated into 8 channels for the montage in ‘sleepSMG’ (‘SM.set’)

Filtering and separating files
Use sleepSMG

- Go to ‘SM.set’ directory
- Stores stages in .mat file, variable stageData.stages

Additional things you can do:
  - Back-project spindle / SO analyses to test how well the automatic scripts are doing
Epoching & artifact rejection

- Here we reject artifacts in 5s segments
  - My philosophy: reject anything that isn’t brain activity
- Use your judgment
  - If a single channel shoots up by 10,000 microvolts, probably should reject it
  - If it does this repeatedly, should probably go back and interpolate this channel and run through steps 2 & 3 again
  - There is constantly this tug and pull of the way to have optimal data. At some point you could interpolate small segments and do this forever and it would take forever, but you’d technically have better data. I see it best as being relatively conservative and time-saving with things, as I think the signal lost in not obsessing over data is relatively small.
Epoching & artifact rejection

- StepThree splits up data into 5s “epochs”
  - Why 5s chunks? It goes quickly, though if you are overly concerned about losing extra data you can split it up how you want in StepThree and beyond steps
- Open ‘TIB_Filt_Epoch_5s.set’ for the subject
  - EEGLAB → Tools → Reject Data Epochs → Reject By Inspection
  - Keep default checks (do NOT select the Reject button), mark epochs you want to reject and DO NOT FORGET TO CLICK ‘UPDATE MARKS’ WHEN YOU’RE DONE
  - Type reject = EEG.reject.rejmanual; save reject reject;
  - Basically we are MARKING epochs for rejection and saving a .mat file of a 0 or 1
    - Why do we do this? In case you have to go back and change something in pre-processing, we have a file externally saved so we won’t have to do it again with new .set file!
Physiology scripts

• These files will read stages and artifact rejected epochs for each subject
• Can run on all stages separately, I run on just NREM altogether
• Saves each in individual folders
• Spindle / SO: saves each individual spindle characteristics (frq, dur, pow, timing) as well as spitting out density / total #s
• SOphase - gives SO phase and power for a full continuous recording
Spindle algorithm

Frequency = \[
\frac{\text{#peaks}}{\text{length}}
\]

Duration = \text{End} - \text{Start} (must be 2.5-3s)

Power = area between RMS curve and threshold
Spindle algorithm

- SO script is similar
- Here you should try to walk through the script in bits and see how the algorithm works
Aligning physiology w/ cues

* Counted as spindle started between 0-1s (but you could cut it however you want).
Basic alignment approach

- StepFive.m
  - Loop through all relevant cues
    - Restrict to sleep cues, possibly only in one condition
  - Loop through all spindles in an electrode (or cluster of electrodes one by one)
  - If spindles occur within a certain time of cue, mark the characteristics for later
- DF4D.m / UnivariateStuff.m
  - DF4D does the above, but for voltage (ERP), sigma RMS, etc. data aligned to cues, post-cue spindles, or post-cue SOs. All relevant behavioral data are saved along with the data in the ‘ref’ field of the main ‘unwrap’ struct.
  - UnivariateStuff then compiles the above data depending on the contrast(s) of interest, plots them, and does some statistics. There are other scripts that come in later too that are explained in more detail in the comments, but this is the basics.
Using both ICA and sleep

- Huge issue: need to reject continuous data to run ICA, but need to score sleep data in 30s epochs
- Lots of time remapping needed to re-align the events
- Do this only if you need to
- Another option is to look for ICA components during sleep. This is something I’ve explored, but never published. For instance, there are nice spindles / SO components that come out.
Questions?

- Email james.ward.antony@gmail.com