0. Setup

Initialization:

- FieldTrip top-level folder for ft_defaults.m
- MEGneto top level for megne2setup.m
- SPM T1 template visible
 - project path
- analysis name
- rawdata path
- mri path
- overwrite

MATLAB input

- Run ft_defaults.m to add the right. subfolders within FieldTrip.
- Create directories for project
- Initialize JSON config file

megne2setup.m

paths (struct)

out

fcp_1_TaskEpoching.m

1. Setup

- Read in config settings
- Load and match PIDs from MEG and MRI folders
- Create list of subjects with full sets of data

2. Plot triggers

- Generate and save plot of event triggers
- 3. Epoch continuous data into trials
- Set up trial definitions for epoching
- Epoch events list to define time windows corresponding to each trial and overall number of trials; save to config

4. Head motion

- · Identify trials with excessive head motion
- Reject those trials, save filtered version

5. Artifacts

- Check for muscle, jump artifacts
- · Reject those trials with artifacts, save filtered version

6. Bad channels

- Identify and save list of bad channels
- Throw warning if there are many

Outputs

- Plots:
 - Triggers for each event type
 - Head motion visualization
- Epochs (into fcp 1 output JSON):
 - All trials
 - Filtered for head motion (HM)
 - Filtered for HM, muscle, jump artifacts
- · List of bad channels (saved to its own JSON)

fcp 2 PreprocessingICA.m

1,2,3. Processing

1. Setup

• As in fcp 1

2. Noise reduction

 Load gradiometer info, 3rd order gradients from CTF to account for noise

4. ICA

- · Remove bad channel signals altogether from analysis (they are dependent on neighbor sensor signals, thus redundant to include in ICA)
- Run ICA

Outputs

ICA components for each participant

fcp 2 5 checkpoint.m

- Human to identify ICA components that are artifacts or too noisy.
- Back-project to clean ICA

Outputs

- Bad ICA components to JSON file
- ICA-denoised data

fcp 3 ChannelRepair.m

1. Setup

As in fcp 1

2. Bad channel repair

 Replace bad channel signal with average (or other option) of neighbours

Outputs

- Final preprocessed data output into *.mat
- fcp 2 output JSON tracking configurations, paths
- Bad ICA components to JSON file

fcp 4 beamforming.m

4. Beamforming

1. Setup

- Read in config settings
- Load and match PIDs from MEG and MRI folders
- Create list of subjects with full sets of data; note that participants may be removed after preprocessing due to insufficient number of trials leftover (e.g., too much noise)

2. Head model preparation

- Load and segment T1 template brain
- Construct head model and do necessary unit conversions
- Construct dipole grid in template brain
- Load desired atlas and create binary masks to define valid voxels within head model

3. Check alignment

- Load and segment participant MRIs, load preproc. MEG data
- Construct subject-specific head, source models
- Check alignment between subject and template head model
- Check alignment source model and head model
- Save images

4. Source reconstruction

- Compute lead field matrix
- Run source analysis, reduce to dominant orientation
- Interpolate functional data onto anatomical data
- Further interpolation onto AAL-defined regions
- Return these results

Outputs (for each participant)

- catmatrix = individual source timeseries in *.mat files: (timepoints in 1 trial) x (num. trials) x (num. AAL nodes)
- srate = sampling rate
- coords = coordinate of source points w/in
- Individual and template head model alignment, *.png
- Source and head model alignment, *.png

1. Setup

- As in fcp 1
- Create filter coefficients

2. Analysis

- Load participant source timeseries matrices (catmatrix)
- band

5. Connectivity Analysis

fcp_5_connectivity.m

- Return PLI or PLV connectivity between pairs of AAL nodes
- Repeat for each trial and frequency band
- Save individual adjacency matrix
- Then, repeat for all participants
- participant and assemble all-participant adjacency matrix

- Individual adjacency matrix (nodes x nodes x trial x freq. band)
- (nodes x nodes x participants x band)

- ft defaults
- path_generation
- save_to_json
- path_check

- load config
- load participants

- · ft artifact muscle
- ft artifact jump
- save_to_json

- ds pid match

- ft rejectartifact

- write match if not empty
- ft definetrial
- plotTriggers
- HeadMotionTool

- detectBadChannels

- load config
- load participants
- ds pid match write match if not empty
- save_to_json
- ft channelrepair
- ft channelselection
- ft_prepare_neighbours
- ft rejectcomponent

- ft resampledata
- ft selectdata
- ft topoplotIC inputBADchannels
- saveSensorsToFile
- ft_componentanalysis ft_denoise_synthetic
- ft_preprocessing

- load config
- load participants
- ds_pid_match
- write_match_if_not_empty
- ft convert units ft freganalysis
- ft plot mesh ft plot sens
- ft plot vol
- ft_prepare_headmodel
- ft_prepare_leadfield

- ft prepare sourcemodel
- · ft read atlas ft read mri
- ft sourceanalysis
- ft sourcedescriptives
- ft sourceinterpolate ft sourceplot
- ft timelockanalysis ft volumelookup
- ft volumesegment

- Apply Hilbert filter to isolate frequency
- · Take average across trials for each

- **Outputs**
- All participant adjacency matrix

load config

- load participants ds pid match
- write_match_if_not_empty
- ft connectivityanalysis
- 2020-01-09