Open-source EEG/MEG analysis software: FieldTrip toolbox in detail

Amit Jaiswal, *DSc (Tech.)*

Senior Clinical Scientist & Application Lead

MEGIN Oy, Espoo, Finland

FieldTrip

MNE
MEG + EEG ANALYSIS & VISUALIZATION

Brainstorm
http://neuroimage.usc.edu/brainstorm

sccn/eeglab

12

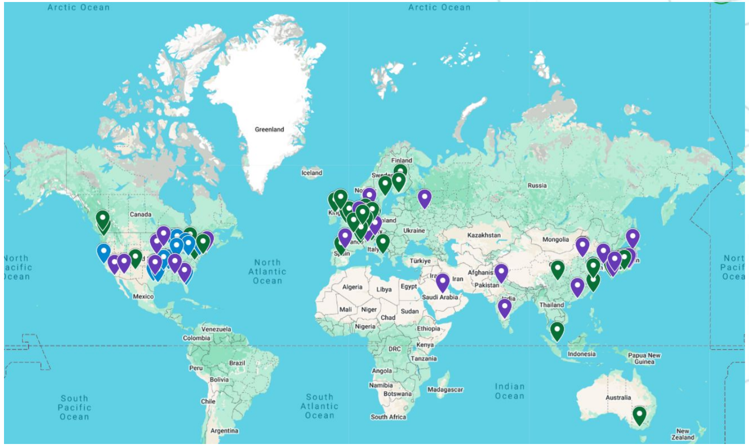
FreeSurfer

MEGIN

Greeting from MEGIN, Finland

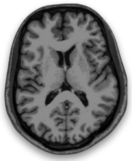
- Providing MEG device and surgical mapping solutions for neuroscience research and clinical applications for more than 35 years.
- More than 120 MEGIN MEG systems located across globe, contributing to research and medical diagnosis.
- Learn more about MEGIN at megin.com

MEGIN head office, Espoo, Finland

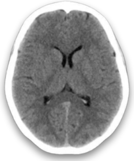


Human brain and neuroimaging

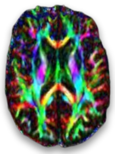
Structural imaging techniques



MRI
(Magnetic resonance imaging)

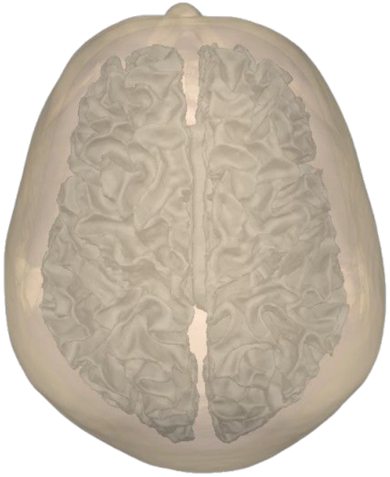


CT
(Computed tomography)



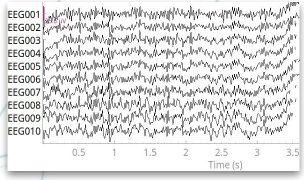
DTI
(Diffusion tensor imaging)

…

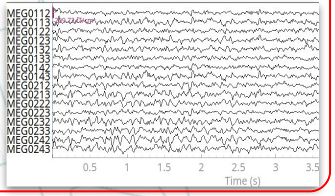


Functional imaging techniques

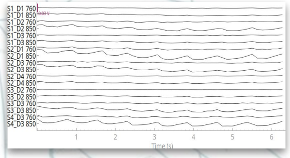
EEG
(Electroencephalography)



MEG
(Magnetoencephalography)



fNIRS
(Functional
near-infrared spectroscopy)

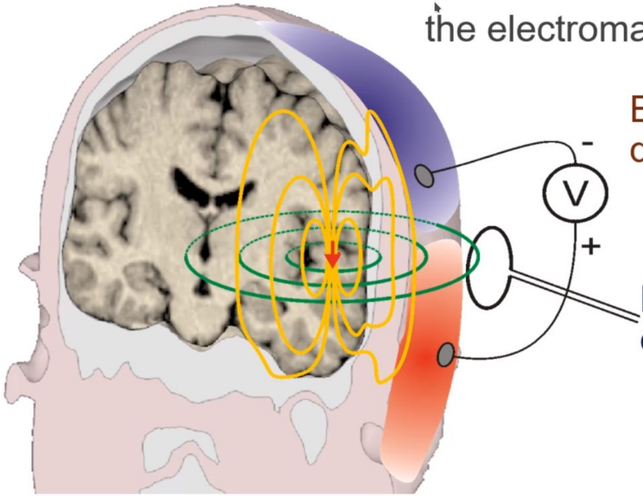


fMRI
(Functional
magnetic resonance imaging)

…

MEG/EEG signals measurement

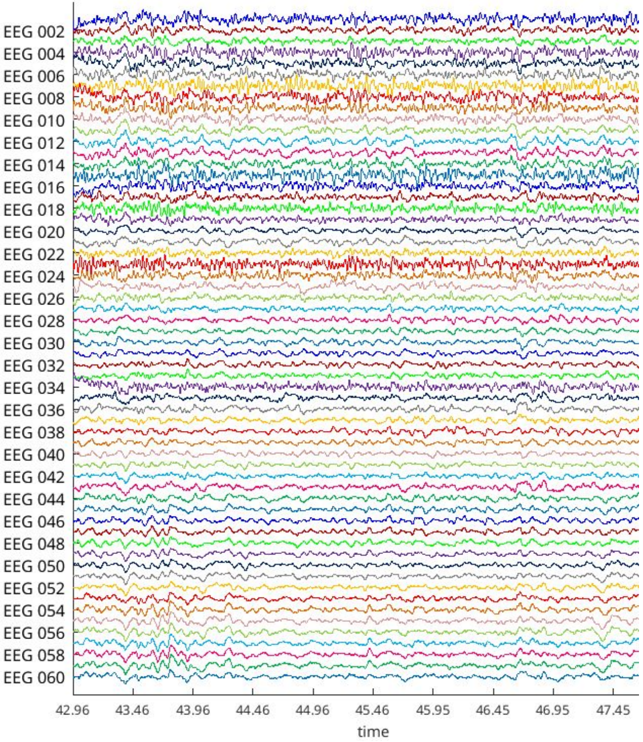
MEG and EEG track electric brain activity by measuring the electromagnetic fields generated by neurons



EEG = measuring the potential differences on the scalp

MEG = measuring neuromagnetic fields outside of the head

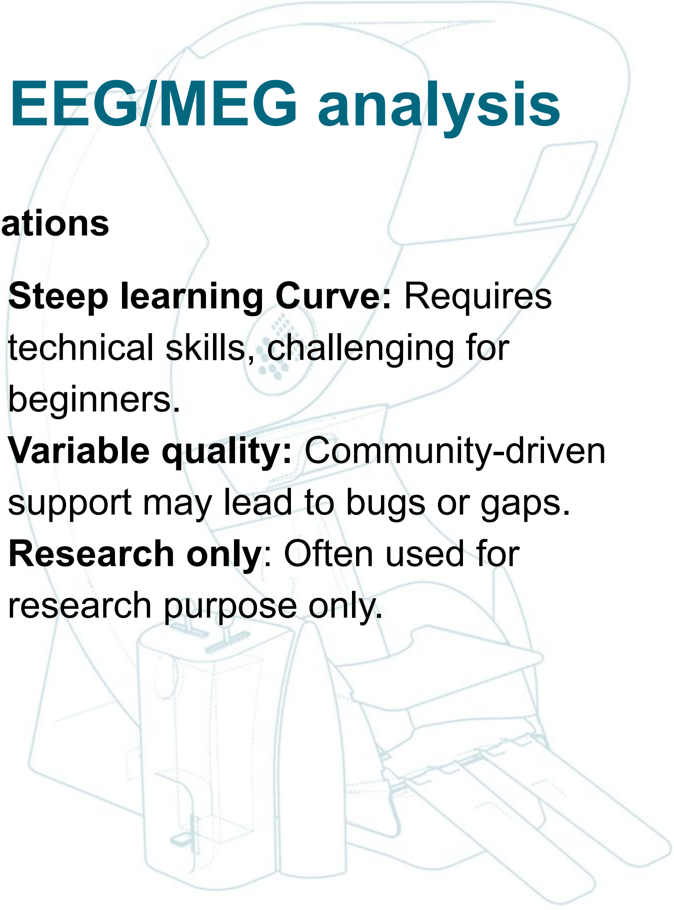
EEG data, analysis methods and software



- Evoked responses
- Induced responses
- Oscillatory responses
- Time-frequency/ spectral methods
- Cluster-based permutation test
- Source reconstruction
- Functional connectivity
- Microstates
- Signal complexities
- …

FieldTrip
MNE_Python
LORETA
SPM12
EEGLAB
MNE
Curry
Brainstorm
BESA
NUTMEG
MEGreview

Why to use open-source software for EEG/MEG analysis



Advantages

- **Unlimited access:** Available to all, removing financial barriers to researchers.
- **Customizable:** Source code can be modified to suit specific research needs.
- **Collaborative:** Encourages global contributions, fostering innovation.
- **Up-to-date:** Quickly adapts to new methods and analysis techniques.
- **Transparent:** Open code ensures reproducibility and algorithm validation.
- **Well-documented:** Often includes tutorials, forums, and version control.

Limitations

- **Steep learning Curve:** Requires technical skills, challenging for beginners.
- **Variable quality:** Community-driven support may lead to bugs or gaps.
- **Research only**: Often used for research purpose only.

Major open-source EEG/MEG analysis software

- **FieldTrip** https://www.fieldtriptoolbox.org/
  - MATLAB-based scripting.
  - Easy-to-learn but require some coding skills.
- **MNE-Python** https://mne.tools/stable/index.html
  - Python-based; requires Python scripting knowledge.
- **EEGLAB** https://sccn.ucsd.edu/eeglab/
  - Mainly GUI-based; scripting possible.
- **Brainstorm** https://neuroimage.usc.edu/brainstorm/
  - GUI-based; easy-to-use.
  - Scripting possible.
- **SPM12** https://www.fil.ion.ucl.ac.uk/spm/software/spm12/
  - Structural-functional analysis possible; scripting as well as GUI-based.

FieldTrip

MNE
MEG + EEG ANALYSIS & VISUALIZATION

sccn/eeglab

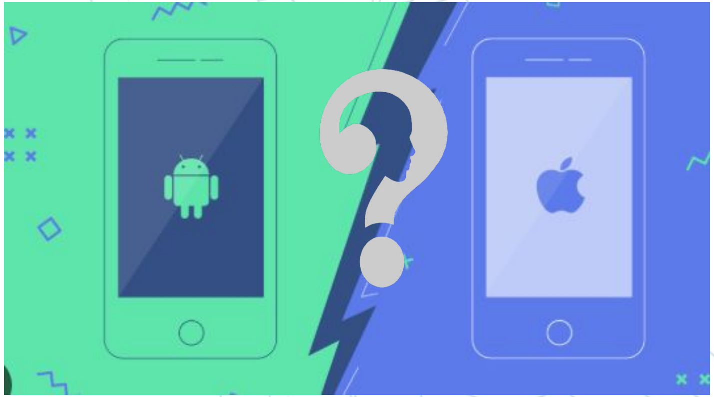
Brainstorm
http://neuroimage.usc.edu/brainstorm

12

Differences and similarities

- Jaiswal, A., Nenonen, J., Stenroos, M., Oostenveld, R., ... & Parkkonen, L. (2020). Comparison of beamformer implementations for MEG source localization. *NeuroImage*, *216*, 116797.

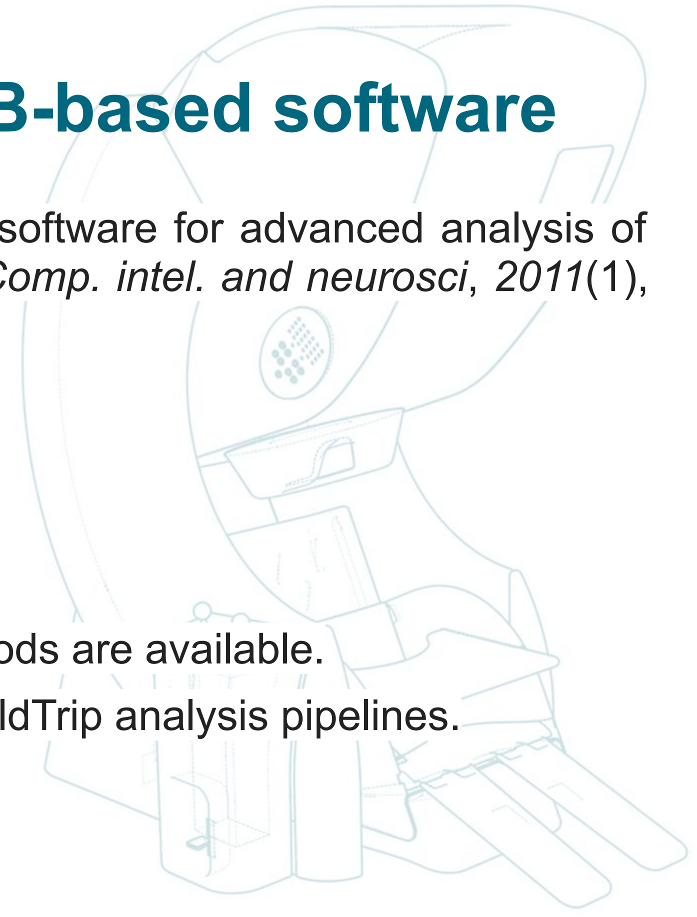
- **Platform:** MATLAB or Python or C/C++
- **Interface:** Coding or Graphical-UI.
- **Usability:** Easy or tedious -to-use.
- **Integration with other s/w**: Simple or complex.
- **Reproducibility:** Similar or different.



FieldTrip: an open-source MATLAB-based software

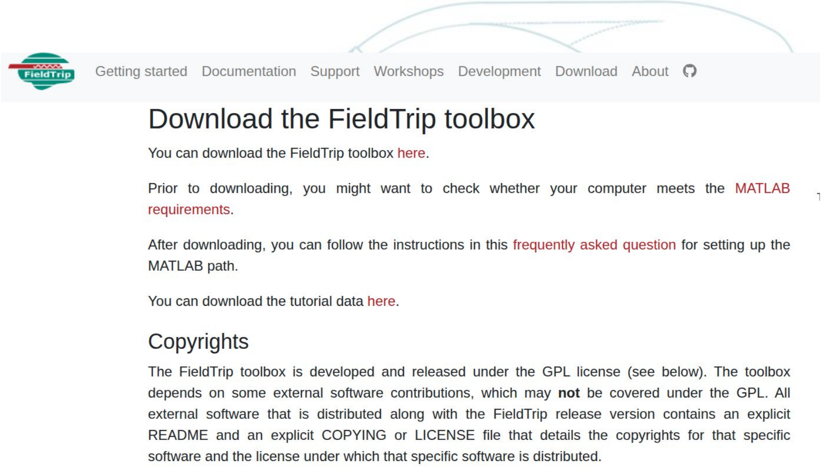
- Oostenveld, R. et al. (2011). FieldTrip: open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Comp. intel. and neurosci*, *2011*(1), 156869.

- **Why FieldTrip:**
    - Easy to learn. Simple and well structured code.
    - MATLAB is almost free for academia.
    - Many conventional and advanced analysis methods are available.
    - Several plugins provide easy integration with FieldTrip analysis pipelines.
    - Active developers community.



Starting with FieldTrip

- Where to get it: https://github.com/fieldtrip/fieldtrip
- How to install:
  - *addpath('fieldtrip directory on your drive')*
  - *ft_defaults*
- Where to start it:
  - https://www.fieldtriptoolbox.org/documentation/
  - https://www.fieldtriptoolbox.org/getting_started/
- What can you do with it?
  - Time-domain analysis, spectral analysis, clustering, source reconstruction, connectivity analysis, and many more.


FieldTrip
Getting started Documentation Support Workshops Development Download About

Download the FieldTrip toolbox

You can download the FieldTrip toolbox here.

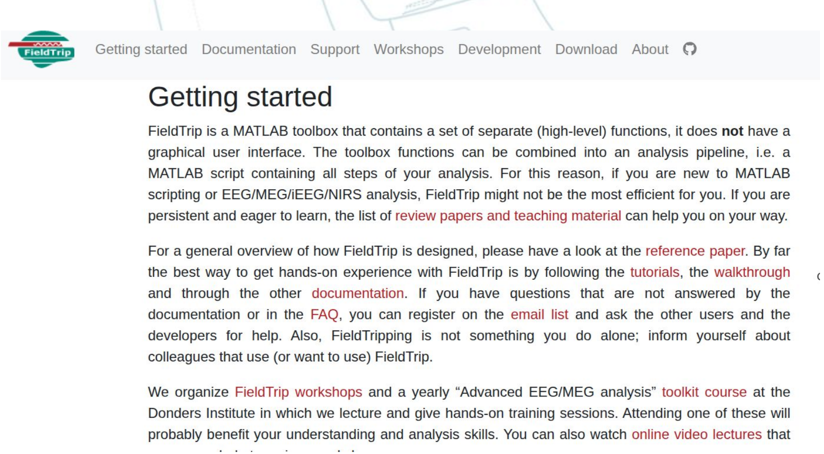
Prior to downloading, you might want to check whether your computer meets the MATLAB requirements.

After downloading, you can follow the instructions in this frequently asked question for setting up the MATLAB path.

You can download the tutorial data here.

Copyrights

The FieldTrip toolbox is developed and released under the GPL license (see below). The toolbox depends on some external software contributions, which may **not** be covered under the GPL. All external software that is distributed along with the FieldTrip release version contains an explicit README and an explicit COPYING or LICENSE file that details the copyrights for that specific software and the license under which that specific software is distributed.


FieldTrip
Getting started Documentation Support Workshops Development Download About

Getting started

FieldTrip is a MATLAB toolbox that contains a set of separate (high-level) functions, it does **not** have a graphical user interface. The toolbox functions can be combined into an analysis pipeline, i.e. a MATLAB script containing all steps of your analysis. For this reason, if you are new to MATLAB scripting or EEG/MEG/iEEG/NIRS analysis, FieldTrip might not be the most efficient for you. If you are persistent and eager to learn, the list of review papers and teaching material can help you on your way.

For a general overview of how FieldTrip is designed, please have a look at the reference paper. By far the best way to get hands-on experience with FieldTrip is by following the tutorials, the walkthrough and through the other documentation. If you have questions that are not answered by the documentation or in the FAQ, you can register on the email list and ask the other users and the developers for help. Also, FieldTripping is not something you do alone; inform yourself about colleagues that use (or want to use) FieldTrip.

We organize FieldTrip workshops and a yearly "Advanced EEG/MEG analysis" toolkit course at the Donders Institute in which we lecture and give hands-on training sessions. Attending one of these will probably benefit your understanding and analysis skills. You can also watch online video lectures that

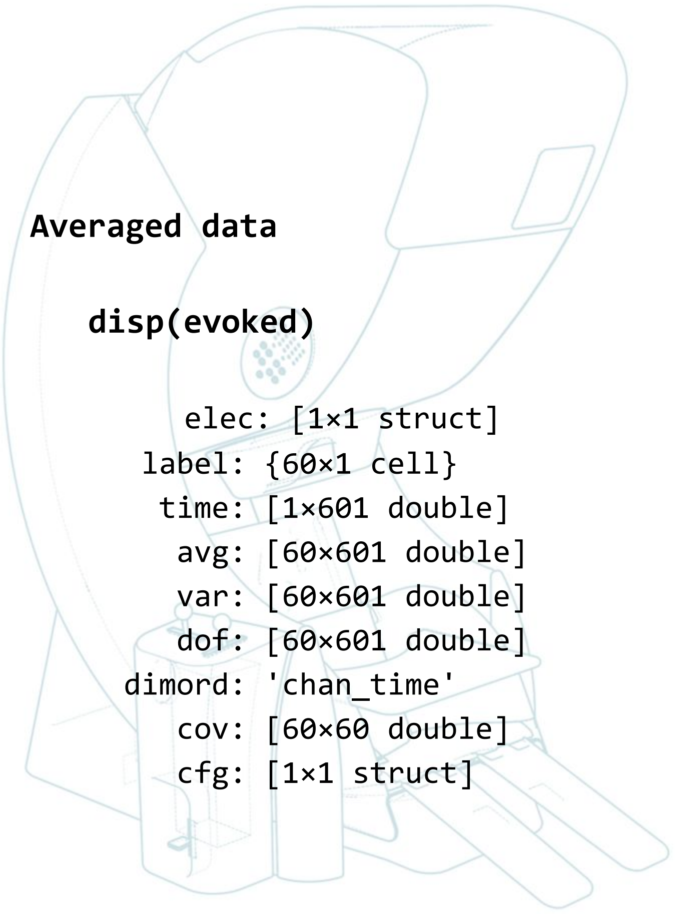
FieldTrip code structure

- *output = function_name(cfg)*
- *output = function_name(cfg, input)*

process configuration

```
Example:
cfg                = [];
cfg.continuous = 'yes';
cfg.channel    = 'all';
cfg.dataset    = filename;
raw = ft_preprocessing(cfg);
```

Data structures



Continuous data

```
disp(raw)

       hdr: [1×1 struct]
     label: {70×1 cell}
      time: {1×1 cell}
     trial: {[70×166800 double]}
   fsample: [600.6150]
sampleinfo: [1 166800]
      elec: [1×1 struct]
       cfg: [1×1 struct]
```

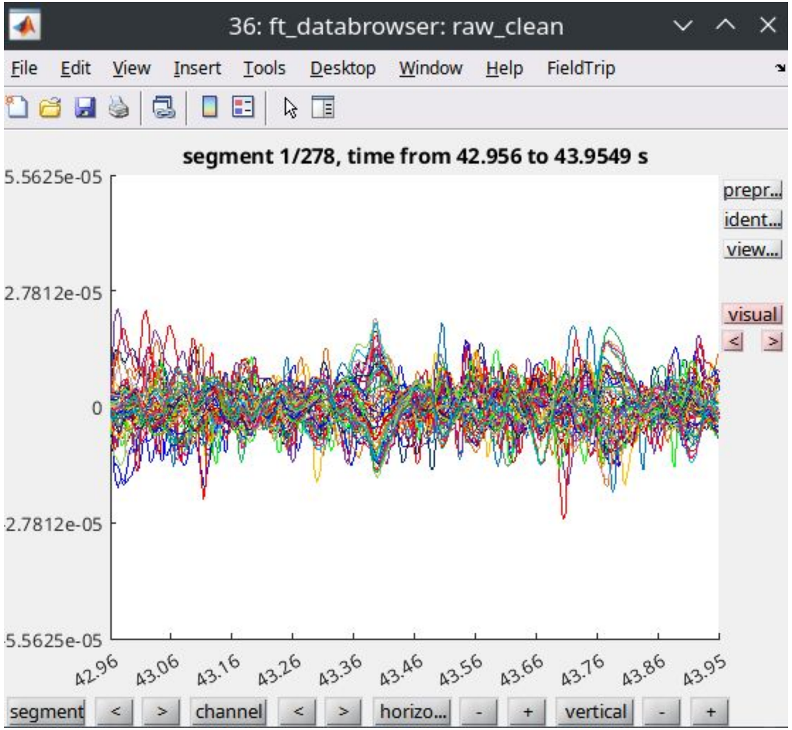
Segmented data

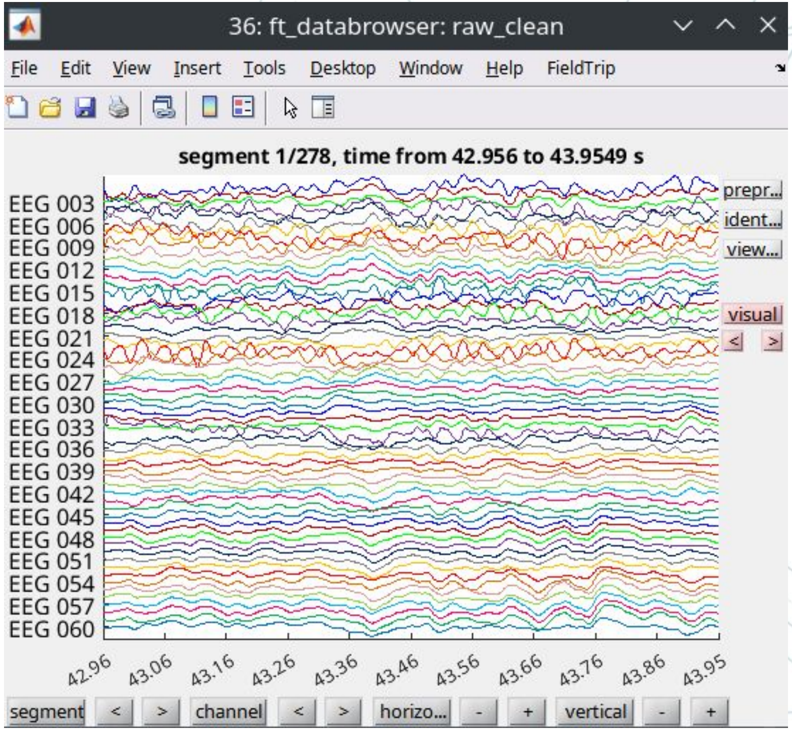
```
disp(epochs)

      elec: [1×1 struct]
   fsample: 600.6150
     label: {60×1 cell}
       cfg: [1×1 struct]
       hdr: [1×1 struct]
     trial: {1×71 cell}
      time: {1×71 cell}
 trialinfo: [71×1 double]
sampleinfo: [71×2 double]
```

Averaged data

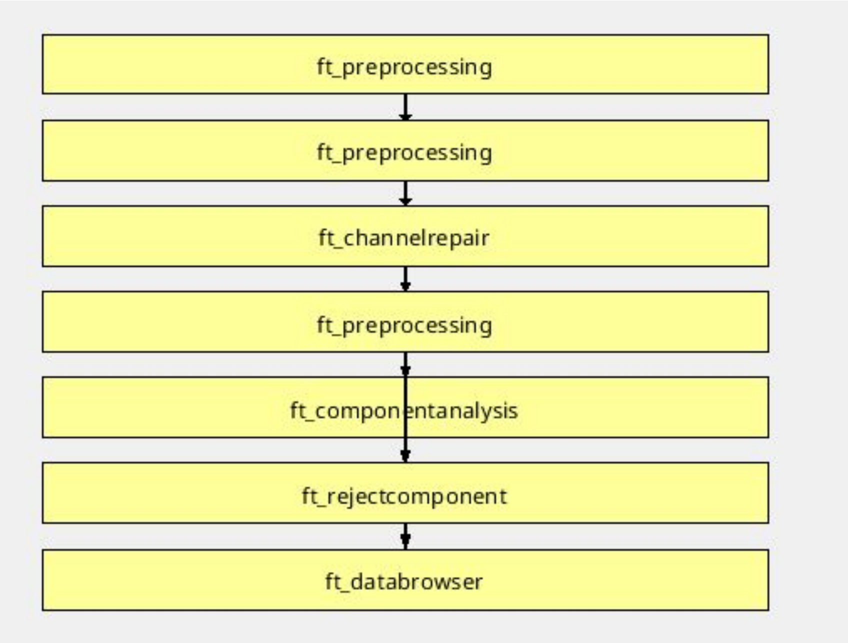
```
disp(evoked)

    elec: [1×1 struct]
   label: {60×1 cell}
    time: [1×601 double]
     avg: [60×601 double]
     var: [60×601 double]
     dof: [60×601 double]
  dimord: 'chan_time'
     cov: [60×60 double]
     cfg: [1×1 struct]
```

Data browser or reviewing tool





Processing pipelines



ft_preprocessing

ft_preprocessing

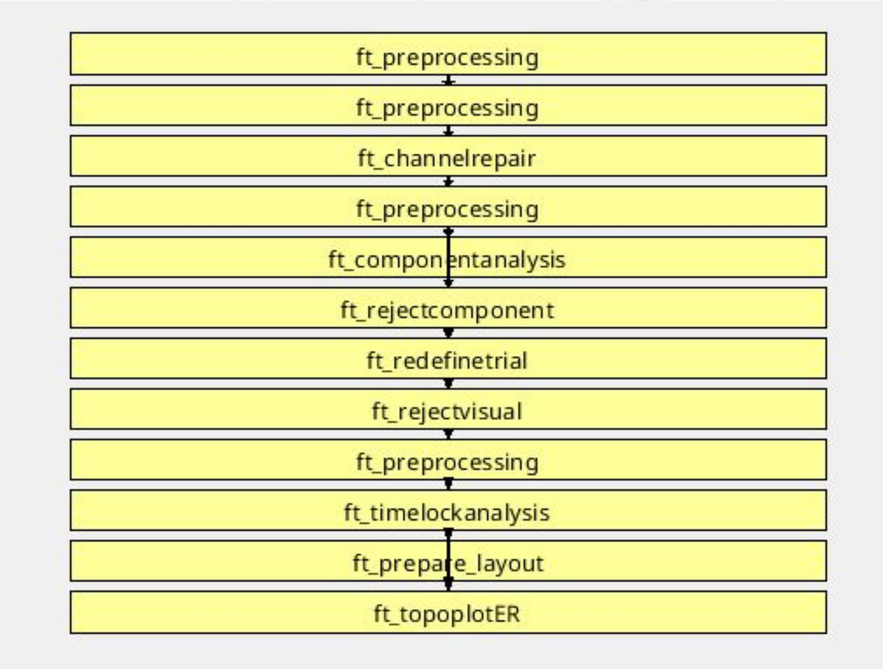
ft_channelrepair

ft_preprocessing

ft_componentanalysis

ft_rejectcomponent

ft_databrowser



ft_preprocessing

ft_preprocessing

ft_channelrepair

ft_preprocessing

ft_componentanalysis

ft_rejectcomponent

ft_redefinetrial

ft_rejectvisual

ft_preprocessing

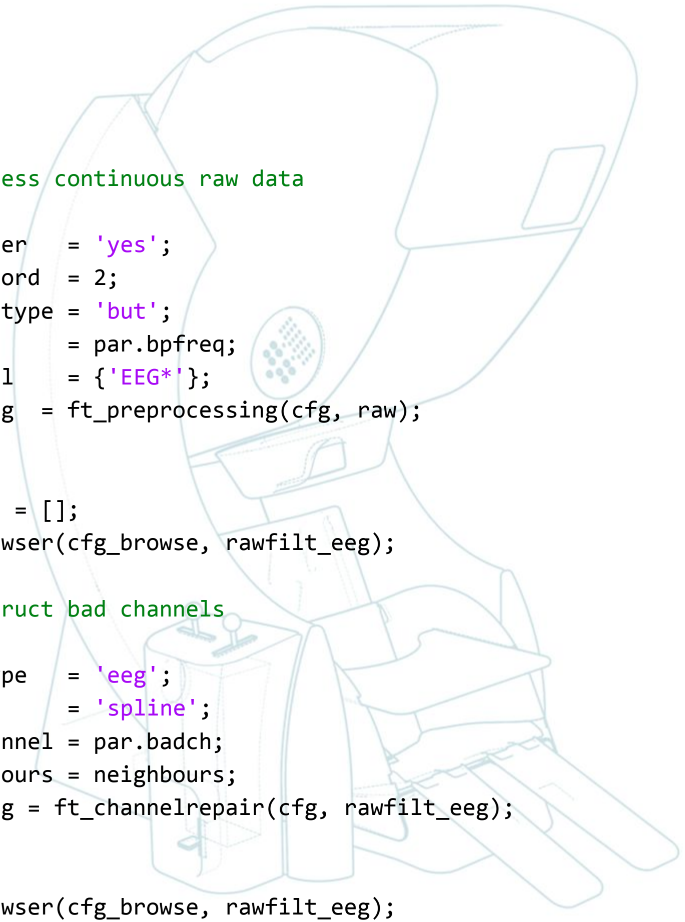
ft_timelockanalysis

ft_prepare_layout

ft_topoplotER

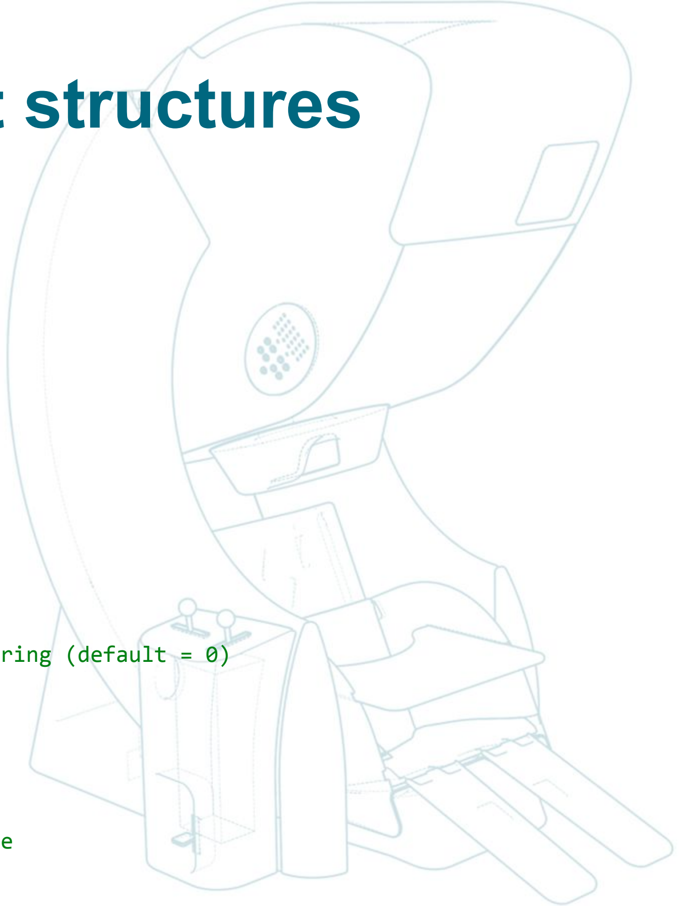
Sample scripts

```
addpath('..//..//TPTools//fieldtrip//)
ft_defaults

%% Set data directory and other parameters
data_dir = '..//..//..//Workshop_IITMandi/';
filename = [data_dir,
'sample_audvis_raw_eeg.fif'];
par            = [];
par.bpfreq     = [1 45];

%% Read the continuous data
cfg            = [];
cfg.continuous = 'yes';
cfg.channel    = 'all';
cfg.dataset    = filename;
raw        = ft_preprocessing(cfg);
disp(raw)

%% Review
cfg = [];
cfg.preproc.demean   = 'yes';
artf = ft_databrowser(cfg, raw);

%% Prepare and plot EEG layout/montage
cfg = [];
cfg.elec    = raw.elec;
cfg.channel = {'EEG*'};
lay2D = ft_prepare_layout(cfg, raw);
```



```
%% Preprocess continuous raw data
cfg = [];
cfg.bpfilter   = 'yes';
cfg.bpfiltord  = 2;
cfg.bpfilttype = 'but';
cfg.bpfreq     = par.bpfreq;
cfg.channel    = {'EEG*'};
rawfilt_eeg  = ft_preprocessing(cfg, raw);

%% Review
cfg_browse = [];
ft_databrowser(cfg_browse, rawfilt_eeg);

%% Reconstruct bad channels
cfg = [];
cfg.senstype   = 'eeg';
cfg.method     = 'spline';
cfg.badchannel = par.badch;
cfg.neighbours = neighbours;
rawfilt_eeg = ft_channelrepair(cfg, rawfilt_eeg);

%% Review
ft_databrowser(cfg_browse, rawfilt_eeg);
```

Function-specific details and input structures



```
function [data] = ft_preprocessing(cfg, data)
% FT_PREPROCESSING reads MEG and/or EEG data according to user-specified trials
% and applies several user-specified preprocessing steps to the signals.
%
% Use as
%   [data] = ft_preprocessing(cfg)
% or
%   [data] = ft_preprocessing(cfg, data)
%
% The first input argument "cfg" is the configuration structure, which contains all
% details for the dataset filename, trials and the preprocessing options.
%
% If you are calling FT_PREPROCESSING with only the configuration as first input
% argument and the data still has to be read from file, you should specify
%   cfg.dataset      = string with the filename
%   cfg.trl          = Nx3 matrix with the trial definition, see FT_DEFINETRIAL
%   cfg.padding      = length (in seconds) to which the trials are padded for filtering (default = 0)
%   cfg.padtype      = string, type of padding (default: 'data' padding or
%                      'mirror', depending on feasibility)
%   cfg.continuous   = 'yes' or 'no' whether the file contains continuous data
%                      (default is determined automatic)
%
% Instead of specifying the dataset in the configuration, you can also explicitly
% specify the name of the file containing the header information and the name of the
% file containing the data, using
%   cfg.datafile     = string with the filename
```

Thanks for your attention!

MEGIN