Source Reconstruction in MEG & EEG

~ From Brain-Waves to Neural Sources ~
Workshop

Karolinska Institutet
June 16th 2017

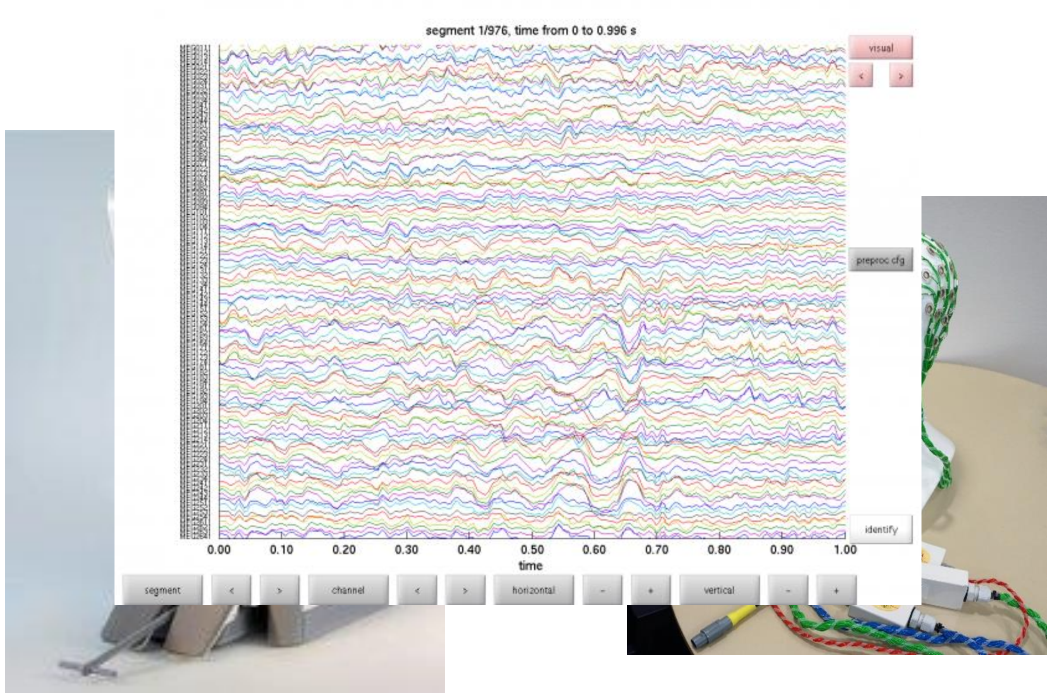
Program for today

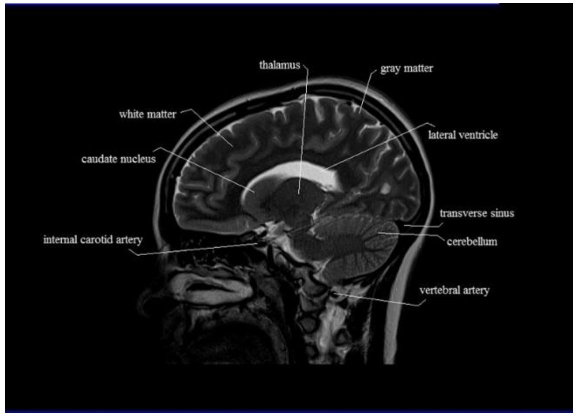
- Intro
  - Overview of a source reconstruction pipeline
  - Overview of toolboxes
  - Quick intro to FieldTrip
- Dipole fit
- Minimum-Norm Estimates
- Beamformer

Overview of (a general) pipeline

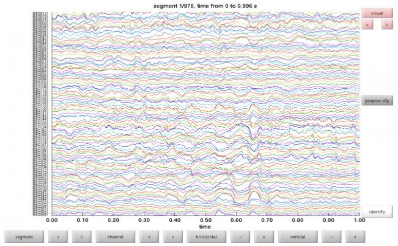
Ingredients

- MEG data
- EEG data
- Structural MRI

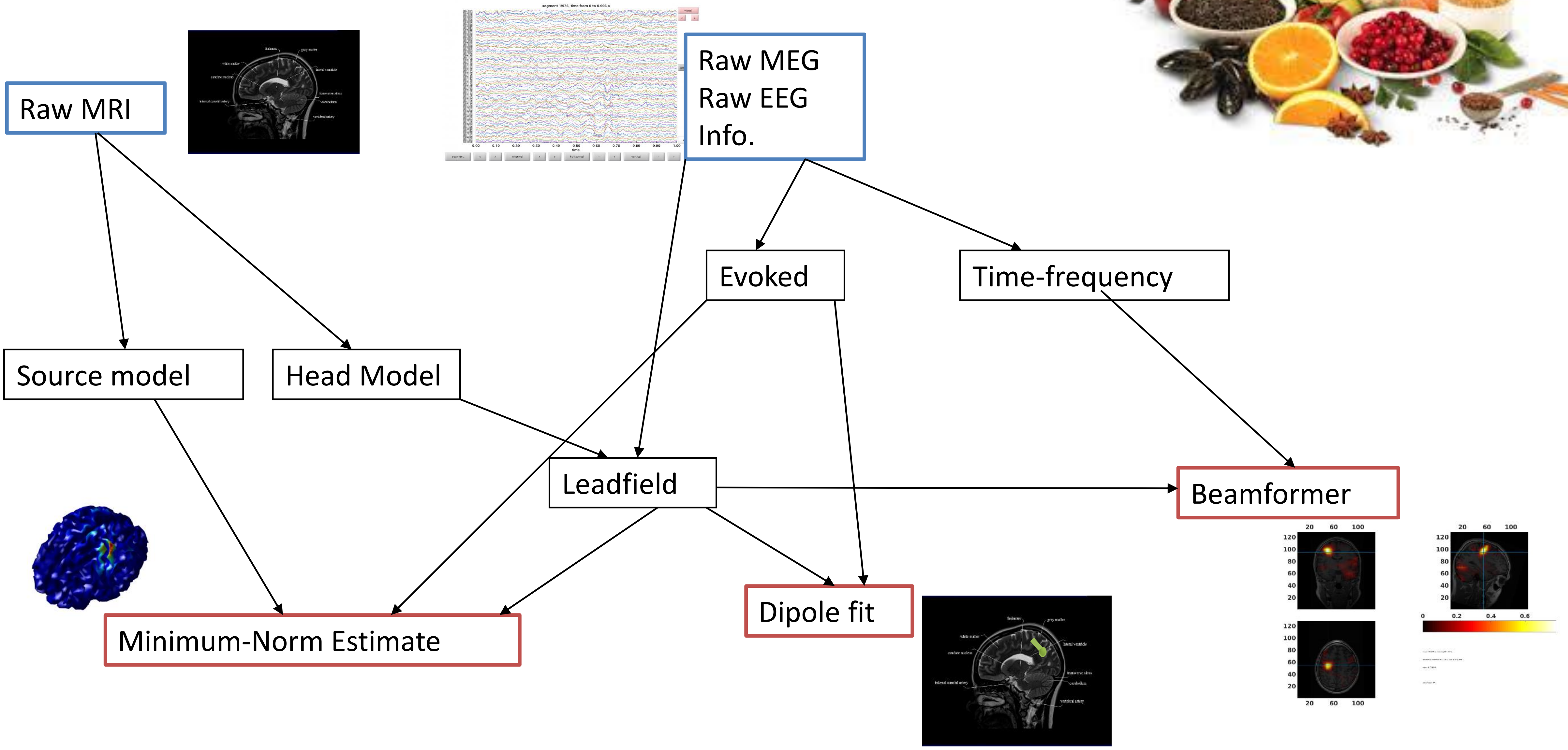




Procedure



Procedure



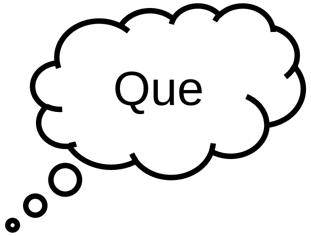
Raw MRI
Raw MEG
Raw EEG
Info.
Evoked
Time-frequency
Source model
Head Model
Leadfield
Beamformer
Minimum-Norm Estimate
Dipole fit

Procedure

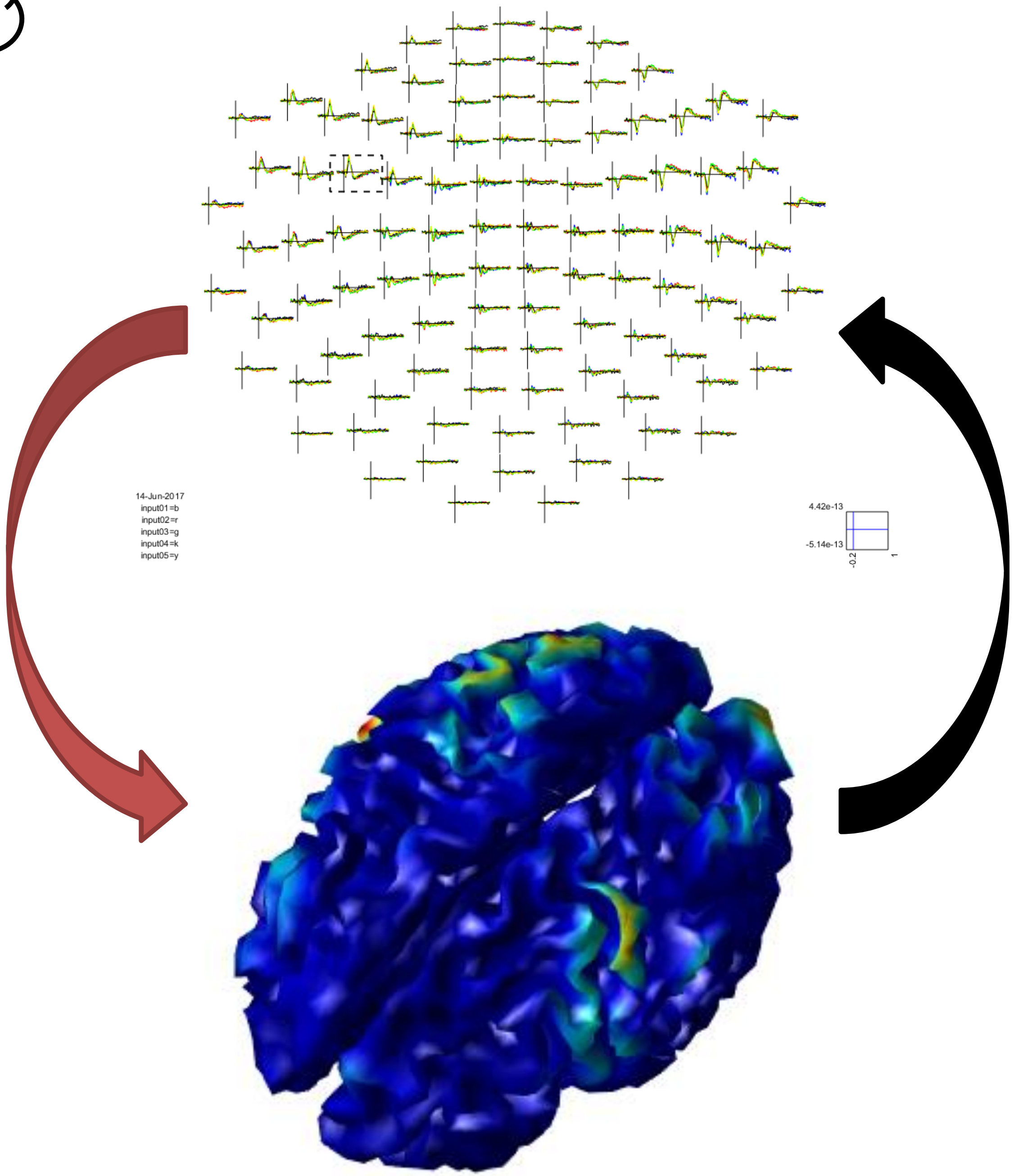
General steps

- **Process MEG/EEG data**: Extract relevant features/time-series
- **Process MRI data**: Create a volume model of the head
- **Make leadfield**: How to project from sensor to source
- **Do source inversion**

Quick terminology



- **Sensor space**
- **Source space**
- **Source model**
- **Leadfield/Forward model**
- **Head model/volume model**

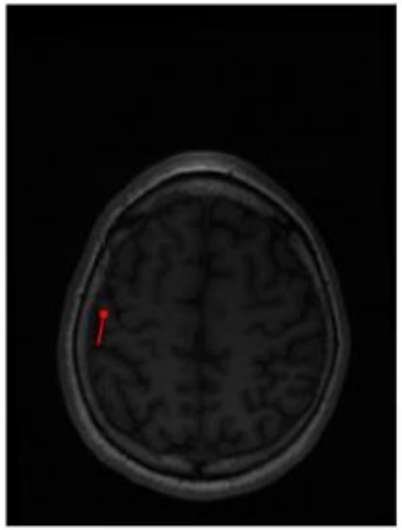
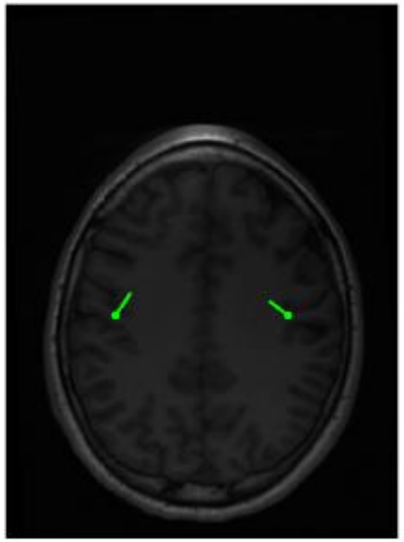
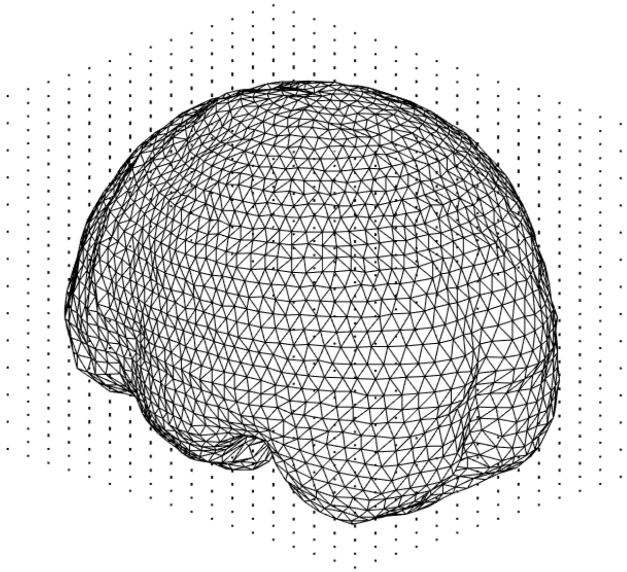
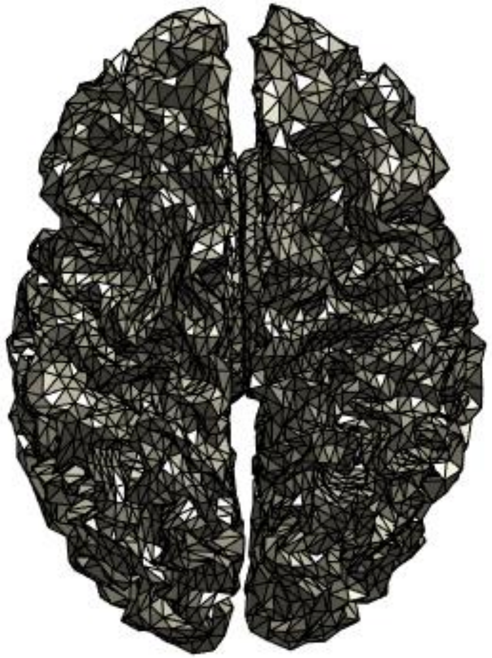


Source models

What we assume about the sources of the signals

Types of source models

- A single (or few) point(s)
- Evenly distributed
- Distributed according to anatomy

Doing source reconstruction

Where to start?

Toolboxes

Open source MATLAB toolboxes
- FieldTrip
- Brainstorm
- SPM
- EEGlab

Open source
- MNE/MNE-Python

Commercial software
- BESA
- Curry

Quick intro to FieldTrip

What is FieldTrip?

An open-source MATLAB toolbox for processing and analysing MEG and EEG data

- Data-processing
- Analysis of evoked and induced responses
- Source analysis
- Connectivity
- Statistics

References

- Oostenveld, R., Fries, P., Maris, E., & Schoffelen J-M (2011). "**FieldTrip: Open Source Software for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data**", *Computational Intelligence and Neuroscience, vol. 2011,*
- www.fieldtriptoolbox.org/references_to_implemented_methods

Set up FieldTrip

Download FieldTrip:

http://www.fieldtriptoolbox.org/download

- Sign up
- Find current date
- Download
- Put in easy to access folder

GitHub:

http://github.com/fieldtrip/fieldtrip

(use the version found with the workshop material)

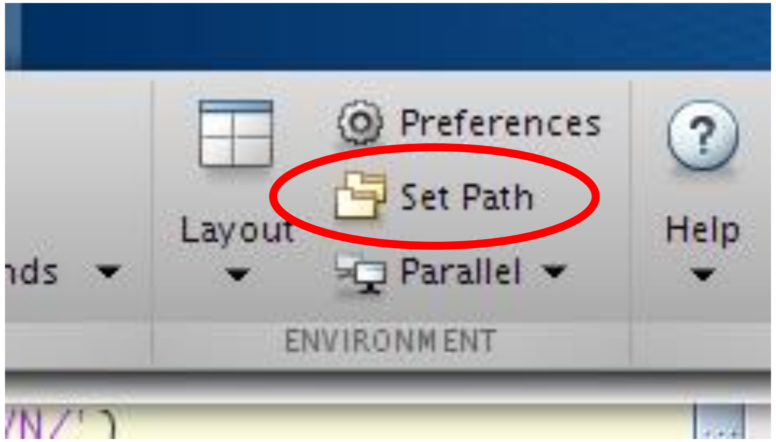
Set up FieldTrip

MATLAB

- ➢addpath '~/mypath/fieldtrip'
- ➢ft_defaults
- ➢cd '/my_working_directory'

NB: If you have SPM as a default path, remove before setting up FieldTrip.

- ➢restoredefaultpath;



FieldTrip functions

data_out = **ft_functionname**(cfg)
data_out = **ft_functionname**(cfg, data_in, …)

”cfg” is configuration structure

cfg.key1 = value1
cfg.key2 = value2
...etc.

Need more help?

Get documentation for functions for help, cfg options, etc, use the help function in MATLAB:

- **help** ft_functionname

Online tutorial, examples and documentation:
http://www.fieldtriptoolbox.org/

FieldTrip data structures (example)

data =

hdr:	[1x1 struct]	*Header info*
label:	{306x1 cell}	*Channel names*
time:	{1x600 cell}	*Time axis for each trial*
trial:	{1x600 cell}	***Trial data** [channels x timepoints]*
fsample:	200	*Sampling frequency (Hz)*
sampleinfo:	[600x2 double]	*[Start end] of each trial in raw data*
trialinfo:	[600x3 double]	*Trial bookkeeping*
grad:	[1x1 struct]	*Gradiometer positions, etc*
elec:	[1x1 struct]	*Electrode positions, etc*
cfg:	[1x1 struct]	*Previous configuration (for bookkeeping)*

What is in the data structures?

Data in the tutorial material

Head models

- Specifying what is brain, skin and skull of the subject

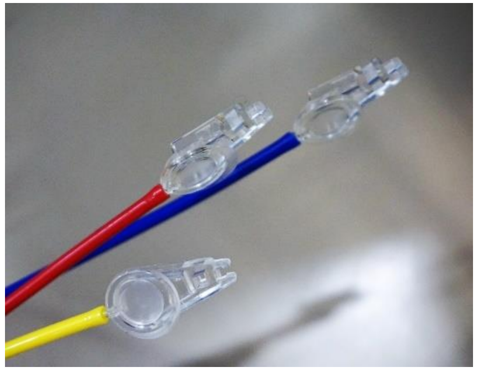
Timelocked data

- Brain activity that unfolds similarly *temporally* over many repetitions of a stimulation

Time-frequency representations

- Brain activity that unfolds similarly in terms of *frequencies* over many repetitions of a stimulation

Description of data

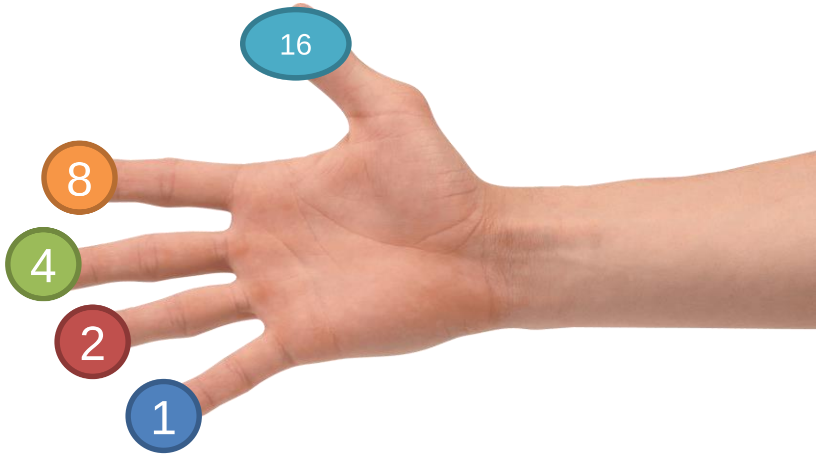


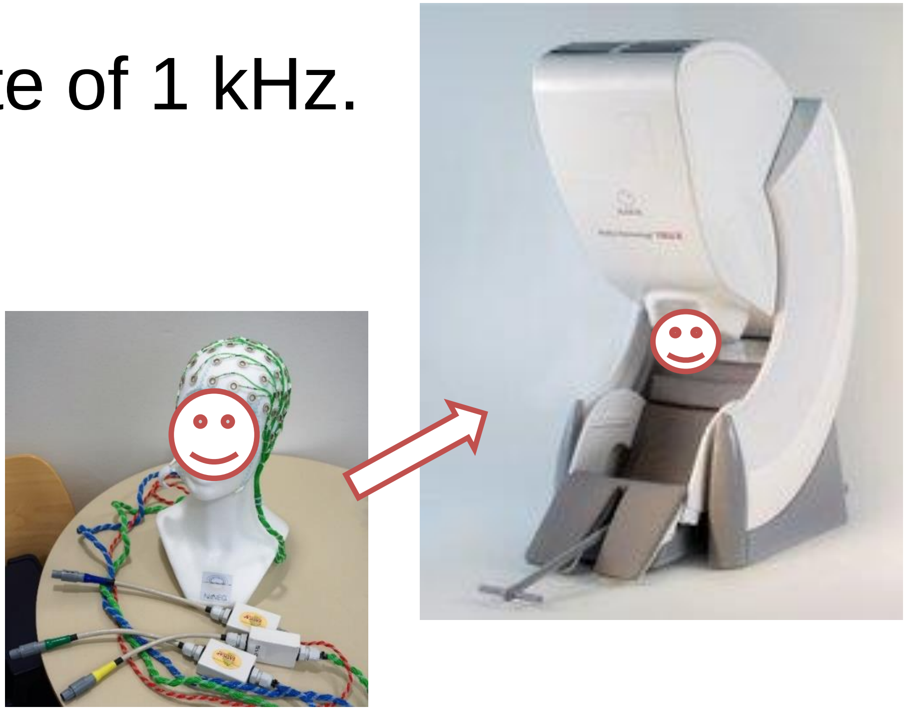
Taks

- Continous tactile stimulation of all fingertips, one at the right hand with a constant frequency of 0.3 Hz

Recordings

- MEG and EEG recorded with a sample rate of 1 kHz.
- Structual MRI





Description of data

Pre-processing

- MaxFilter (MEG)
- Epoched relatives to triggers (*1,2,4,8,16*).
- Bad EEG channels identified, removed, and interpolated, then re-referenced to average of all channels.
- Bad trials containing artefacts removed.
- Independent component analysis (ICA) used to remove eye blinks and eye movements
- Downsampled

See tutorial ***Preprocessing MEG and EEG data***

Description of data

Timelocked by averageing all trials per condition from -0.2 s to 1.0 s relatve to stimulation

- Low-pass filterd at 70 Hz
- Baseline corrected with baseline from -0.2 s to 0.0 s.
- Noise-covariance eastimated from baseline

See tutorial ***Preprocessing MEG and EEG data***

Warmup exercise

Let’s get started…

What is in the data structures?

- Open MATLAB and setup FieldTrip
  - ➢addpath '~/mypath/fieldtrip'
  - ➢ft_defaults
- Change directory to the folder containg the data
  - ➢ cd path
- Open the file containg MEG and EEG data
  - ➢load timelockeds.mat

What is in the data structure?

What is in the data structures?

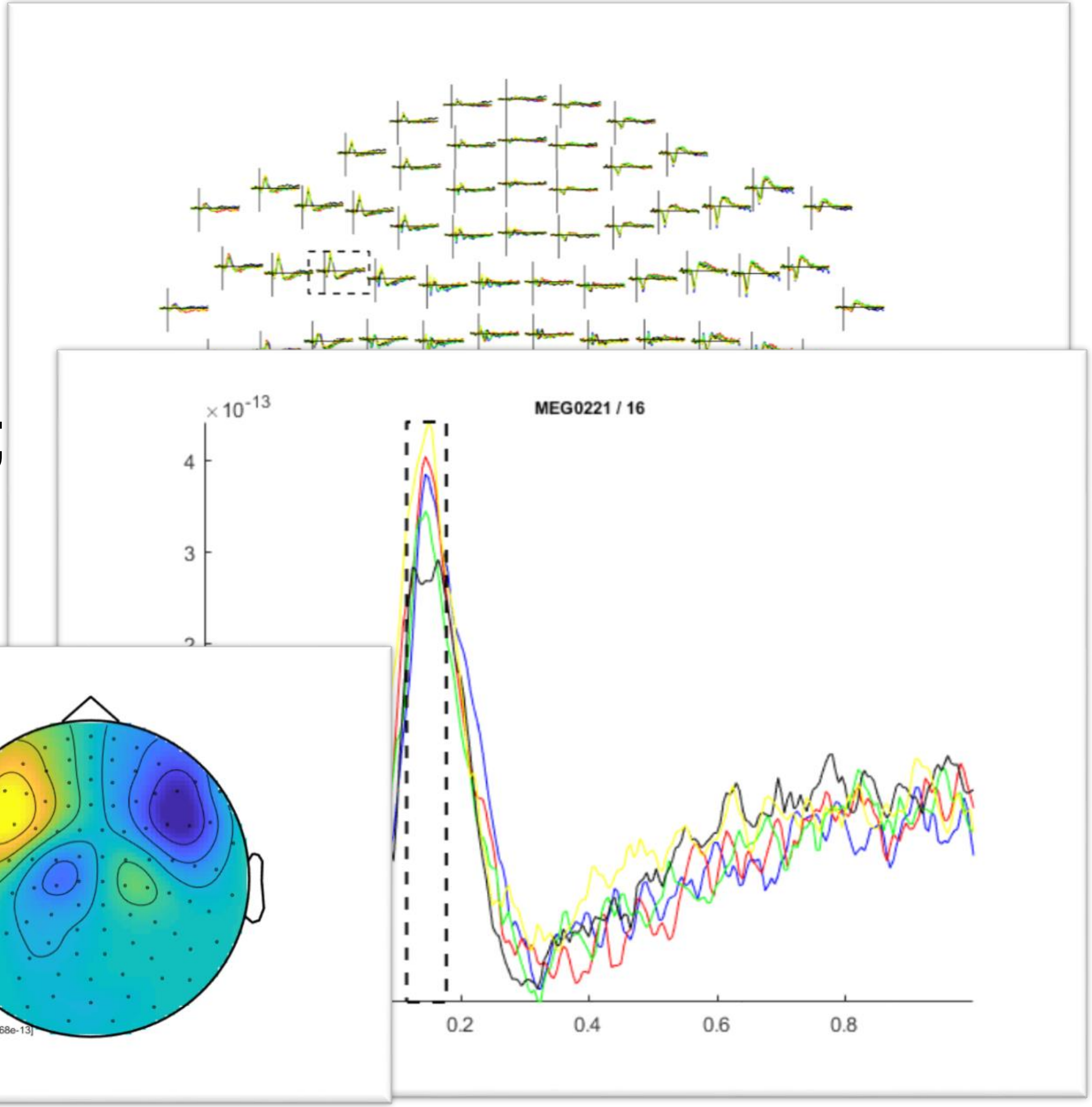
timelockeds{1} =

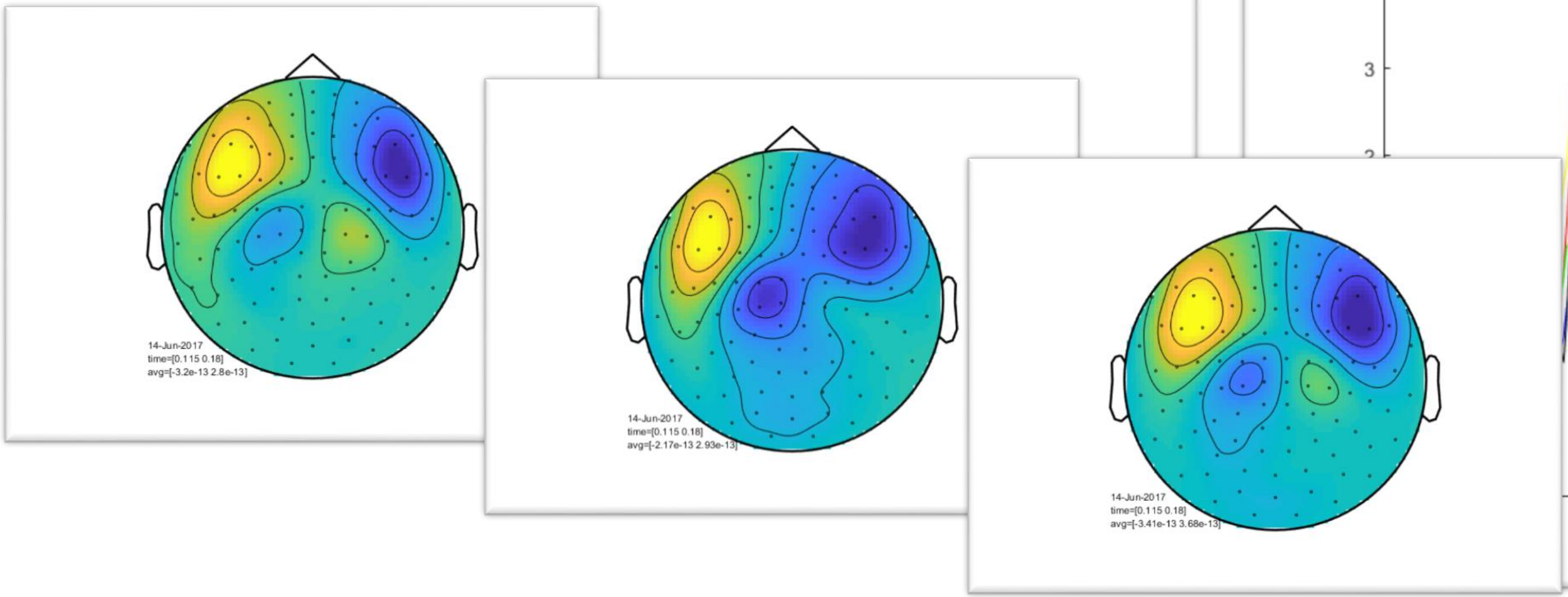
avg:	[434×241 double]	Averaged data per channel
var:	[434×241 double]	Variance
time:	[1×241 double]	labels for time axis
dof:	[434×241 double]	“degrees of freedom”
label:	{434×1 cell}	Channel names
dimord:	'chan_time‘	Dimension order of data (channels x time)
cov:	[434×434 double]	Covariance
elec:	[1×1 struct]	Electrode positions
grad:	[1×1 struct]	Gradiometer positions
cfg:	[1×1 struct]	Call to get this data structure.

What is in the data structures?

Look at the timelocked MEG data

- cfg = [];
- cfg.layout = 'neuromag306mag.lay‘;
- **ft_multiplotER**(cfg, timelockeds{:});

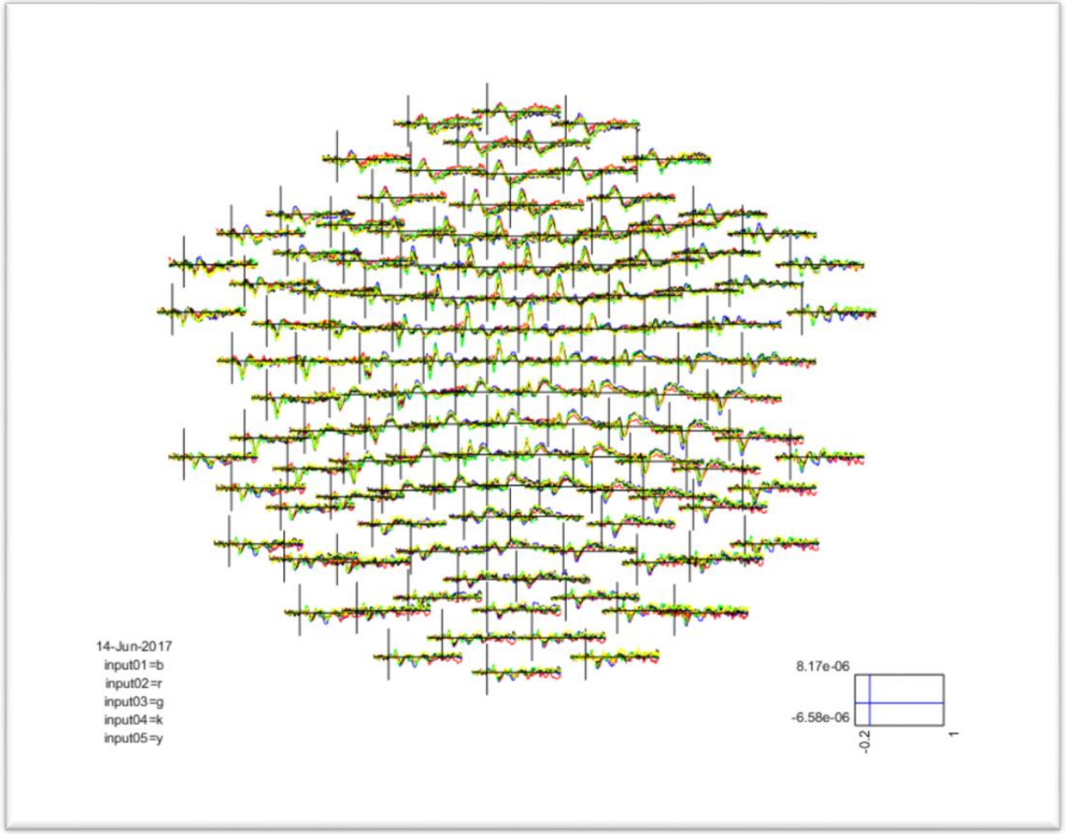




What is in the data structures?

Look at the timelocked MEG data

- cfg = [];
- cfg.layout = 'neuromag306mag.lay‘;
- **ft_multiplotER**(cfg, timelockeds{:});



Look at the timelocked EEG data

- cfg = [];
- cfg.layout = 'neuromag306eeg1005_natmeg.lay'
- **ft_multiplotER**(cfg, timelockeds{:});

Source reconstruction tutorials

Let's get started…

Source reconstruction tutorial

Go to the webpage:
natmeg.se/activities/source_reconstruction/workshop_material.html

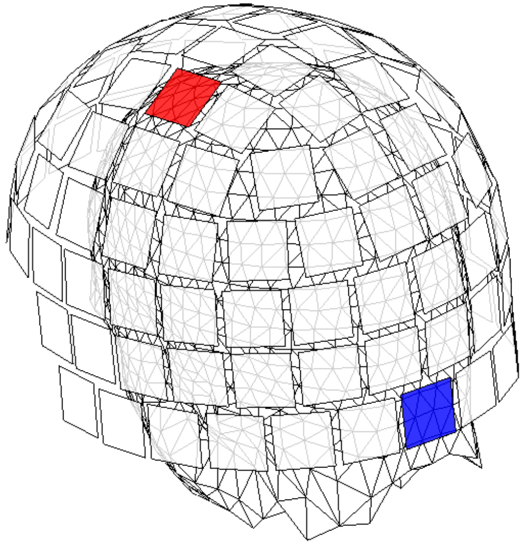
Tutorials

- Pre-processing MEG and EEG data (example)
- Preparing headmodels from MRI (example)
- Dipole fit **Run**
- Prepare source model for MNE (example)
- Minimum-Norm Estimates **Run**
- Beamformer **Run**

Preparing head models

Co-registering, preparing volume, etc.

HEAD MODEL: How was the brain positioned inside the scanner helmet and inside the head?



FieldTrip head model structures

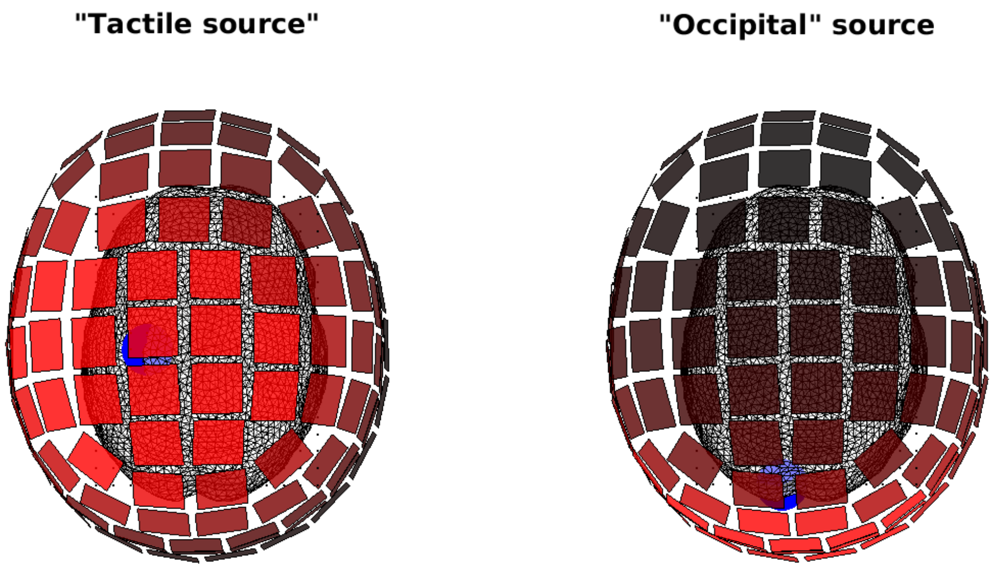
headmodel =

bnd:	[1x3 struct]	the boundaries of the tissues (skin, skull, brain)
cond:	[0.3333 0.0167 0.3333]	The conductivities of the different tissue
skin_surface:	3	which tissue is the skin
source:	1	which tissue is the brain
mat:	[1000x6000 double]	
type:	'bemcp'	the head model type
unit:	'mm'	the unit of the coordinates in the bnd
cfg:	[1x1 struct]	the configuration used

Leadfield

How do the sensors see the sources?

LEADFIELD: How *would* a sensor see a source *if it were* active?



FieldTrip leadfield/grid structures

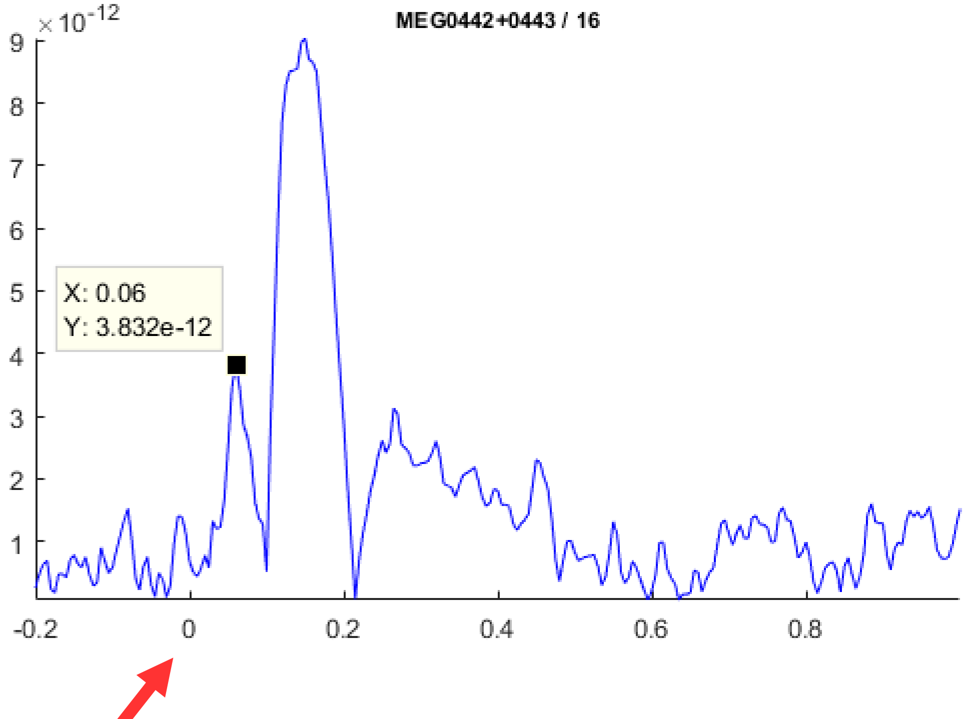
leadfield/grid =

xgrid:	[-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7]	*x* coordinates for the grid of sources
ygrid:	[-8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9]	*y* coordinates for the grid of sources
zgrid:	[0 1 2 3 4 5 6 7 8 9 10 11 12 13]	z coordinates for the grid of sources
dim:	[14 18 14]	*xyz*-dimensions
pos:	[3528x3 double]	coordinates of the grid points
unit:	'cm'	the unit
inside:	[3528x1 logical]	which points are inside the brain
cfg:	[1x1 struct]	the configuration used
leadfield:	{1x3528 cell}	the leadfields for each of the grid points, giving the field moment for each of the sensors in *label* below for a source with unit strength
label:	{102x1 cell}	the labels of the sensor
leadfielddimord:	'{pos}_chan_ori'	the dimension ordering of what is in the leadfield field

Evoked data

Pre-processing, clean, and average

EVOKED DATA: Time-locked activity



1. 0 msec is onset of stimulation
2. An average of ~150 trials
3. Two clear peaks showing activity that are stable across many trials

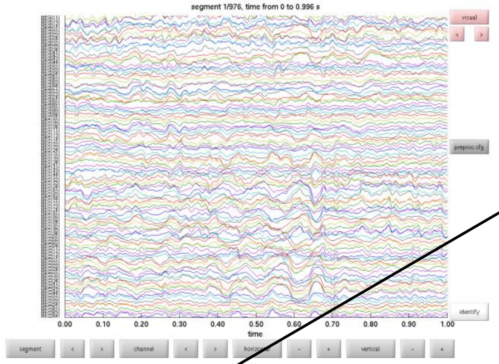
FieldTrip evoked structures

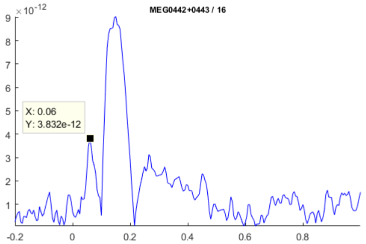
```
evoked =
    avg:    [434x241 double]     %% averages for chan_time
    var:    [434x241 double]     %% variance for chan_time
    time: [1x241 double]         %% time points
    dof:    [434x241 double]     %% degrees of freedom
    label: {434x1 cell}          %% labels of sensors
    dimord: 'chan_time'          %% order of dimensions
    cov:    [434x434 double]     %%covariance between sensors
    elec:   [1x1 struct]         %% electrode specification
    grad:  [1x1 struct]          %% MEG sensor specification
    cfg:    [1x1 struct]         %% the configuration used
```

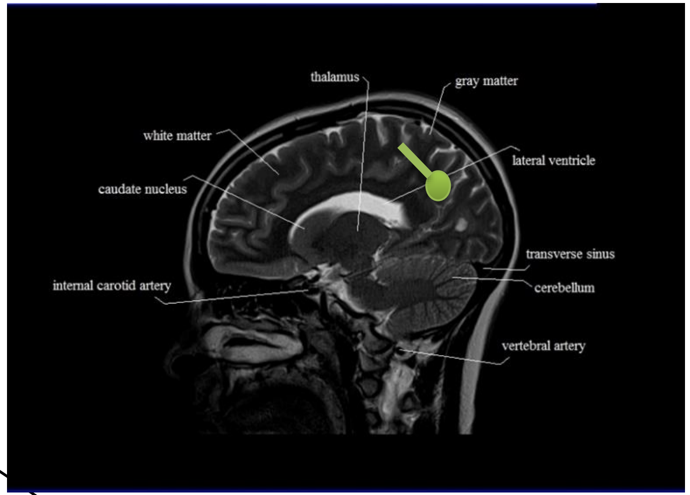
Dipole fits

Single source of peak activity

Overview of pipeline







Raw MEG

Pre-processing

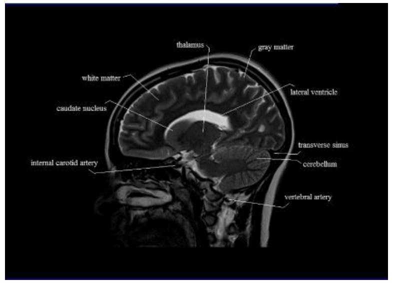
Evoked

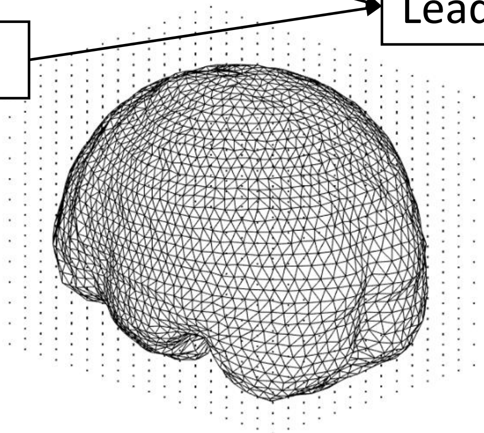
Dipole fit

Leadfield

Raw MRI

Head Model

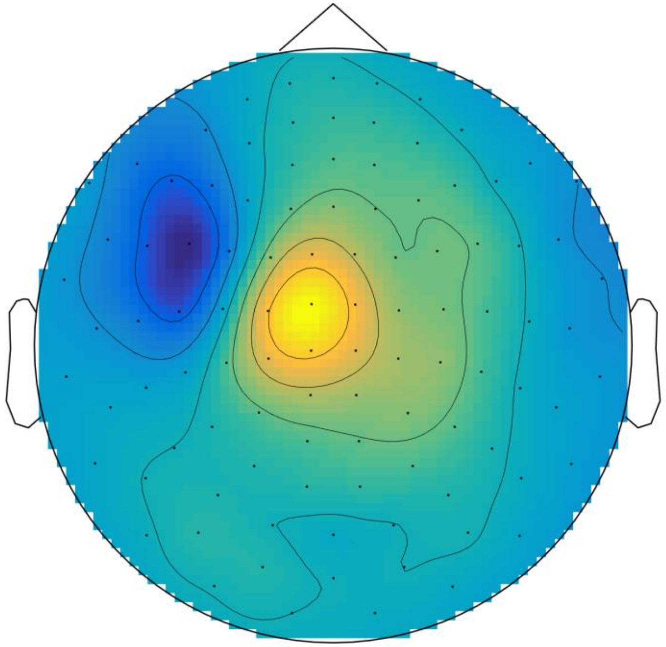


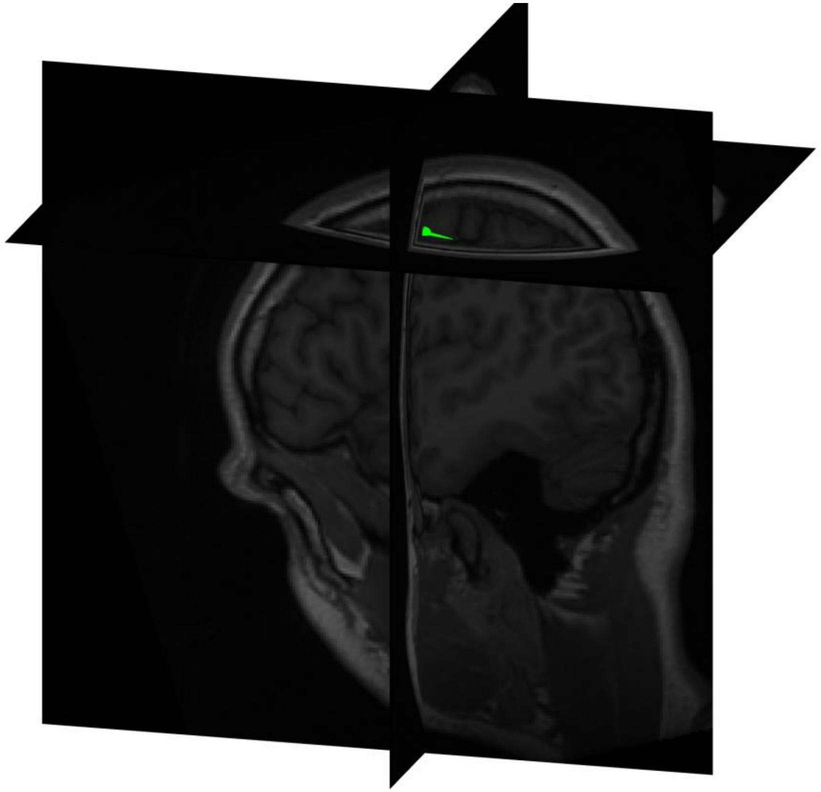


```
cfg = [];
cfg.latency          = [0.040 0.060];      %% latency in sec
cfg.numdipoles       = 1;                  %% number of dipoles
cfg.gridsearch       = 'yes';              %% whether to search for optimal starting position in grid
cfg.grid             = leadfield_mag;      %% the grid
cfg.headmodel        = headmodel_meg;      %% the head model
cfg.dipfit.metric    = 'rv';               %% the metric optimized (residual variance)
cfg.model            = 'regional';         %% the kind of model
cfg.senstype         = 'meg';              %% sensor type
cfg.channel          = 'megmag';           %% sensor specification
cfg.nonlinear        = 'yes';              %% whether the search is non-linear

dipole = ft_dipolefitting(cfg, timelocked_data);
```

Model the brain activity with a single or a few number of dipolar sources





FieldTrip dipole structures

```
dipole =
   label:     {102x1 cell}       %% the labels of the sensors
   dip:       [1x1 struct]       %% the dipoles
   Vdata:     [102x5 double]     %% the data time courses
   Vmodel:    [102x5 double]     %% the model time courses
   time: [0.1150 0.1200 0.1250 0.1300 0.1350] %% time points modelled
   dimord:    'chan_time'        %% order of dimensions
   cfg:       [1x1 struct]       %% the configuration used
```

Minimum-Norm Estimates

Distributed source model

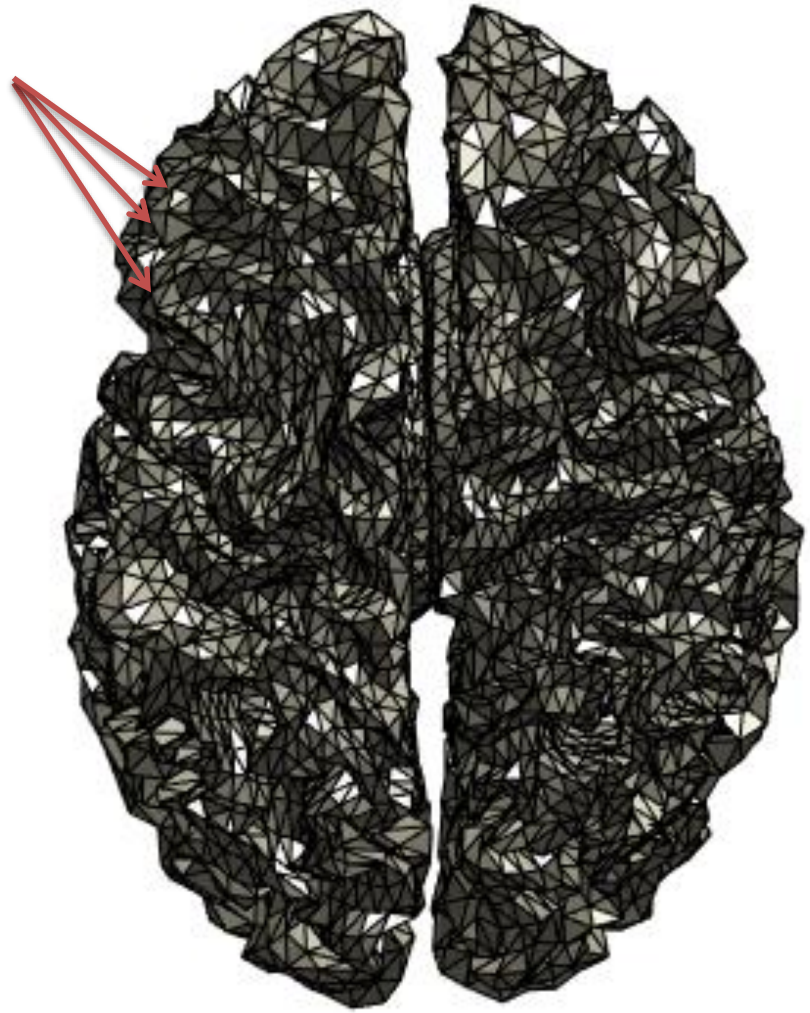
MNE

Distributed mesh of dipoles on cortical surface

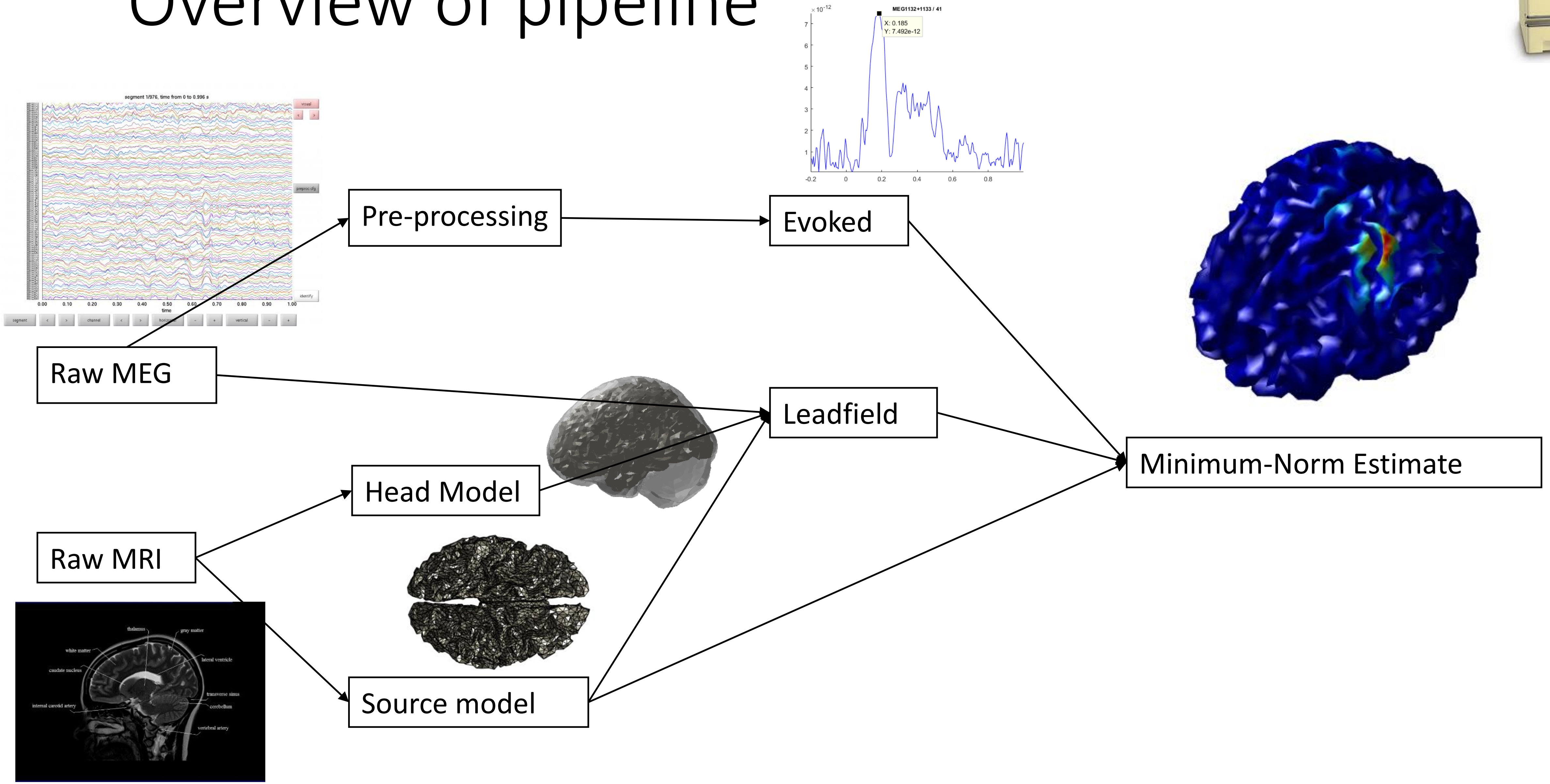
Find source activation pattern that minimises error with measured data

Not iterative

Each vertex represents a dipole



Overview of pipeline

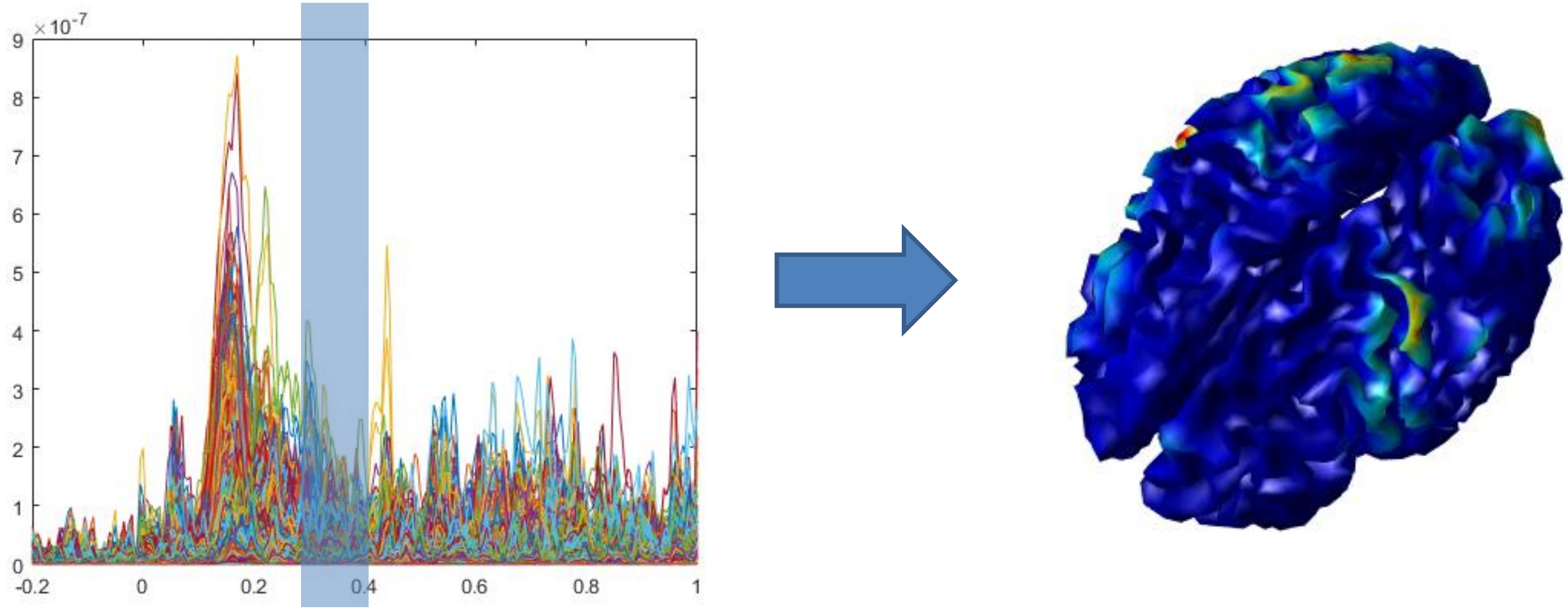


```
cfg = [];
cfg.grad             = data_meg.grad;                 % sensor positions
cfg.channel          = 'meggrad';                     % the used channels
cfg.senstype         = 'meg';                         % Which sensor type?
cfg.grid.pos         = sourcemodel.pos;               % source points
cfg.grid.inside      = 1:size(sourcemodel.pos,1);     % all source points are inside the brain
cfg.headmodel        = headmodel_mne_meg;             % volume conduction model

leadfield_mne_meg = ft_prepare_leadfield( cfg, data_meg);
```

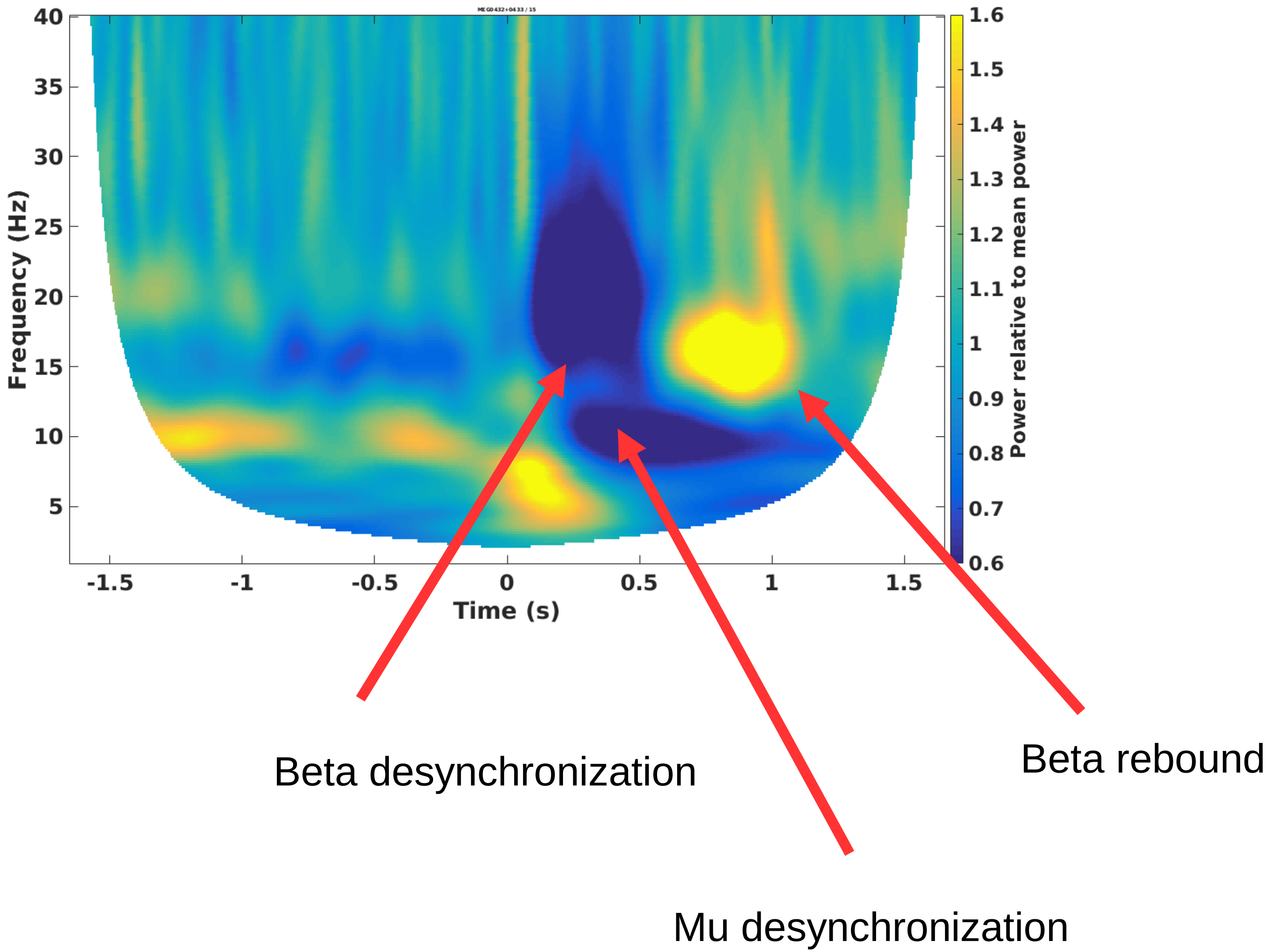
```
cfg                       = [];
cfg.method                = 'mne';                % Tell to use MNE
cfg.channel               = 'meggrad';            % the used channels
cfg.senstype              = 'meg';                % Which sensortype?
cfg.grid                  = leadfield_mne_meg;    % The leadfield
cfg.headmodel             = headmodel_mne_meg;    % The headmodel
cfg.mne.prewhiten         = 'yes';                % If we should prewhiten the lead
cfg.mne.lambda            = 3;                    % Scaling factor for noise (constant)
cfg.mne.scalesourcecov    = 'yes';                % If we should scale?

source_mne_meg  = ft_sourceanalysis(cfg, data_meg);
```


×10^-7
9
8
7
6
5
4
3
2
1
0
-0.2
0
0.2
0.4
0.6
0.8
1

Time-frequency representation

Time varying power differences over different frequencies



FieldTrip tfr structures

```
tfr =
    label:     {332x1 cell}              %% labels of sensors
    dimord:    'chan_freq_time'          %% order of dimensions
    freq:      [1x201 double]            %% frequencies represented
    time:      [1x661 double]            %% time points
    powspctrm: [332x201x661 double]      %% power spectrum by channel x
                                            frequency x time points
    elec:      [1x1 struct]              %% electrode specification
    grad:      [1x1 struct]              %%  MEG sensors specification
    cfg:       [1x1 struct]              %% the configuration used
```

What is in the data structures?

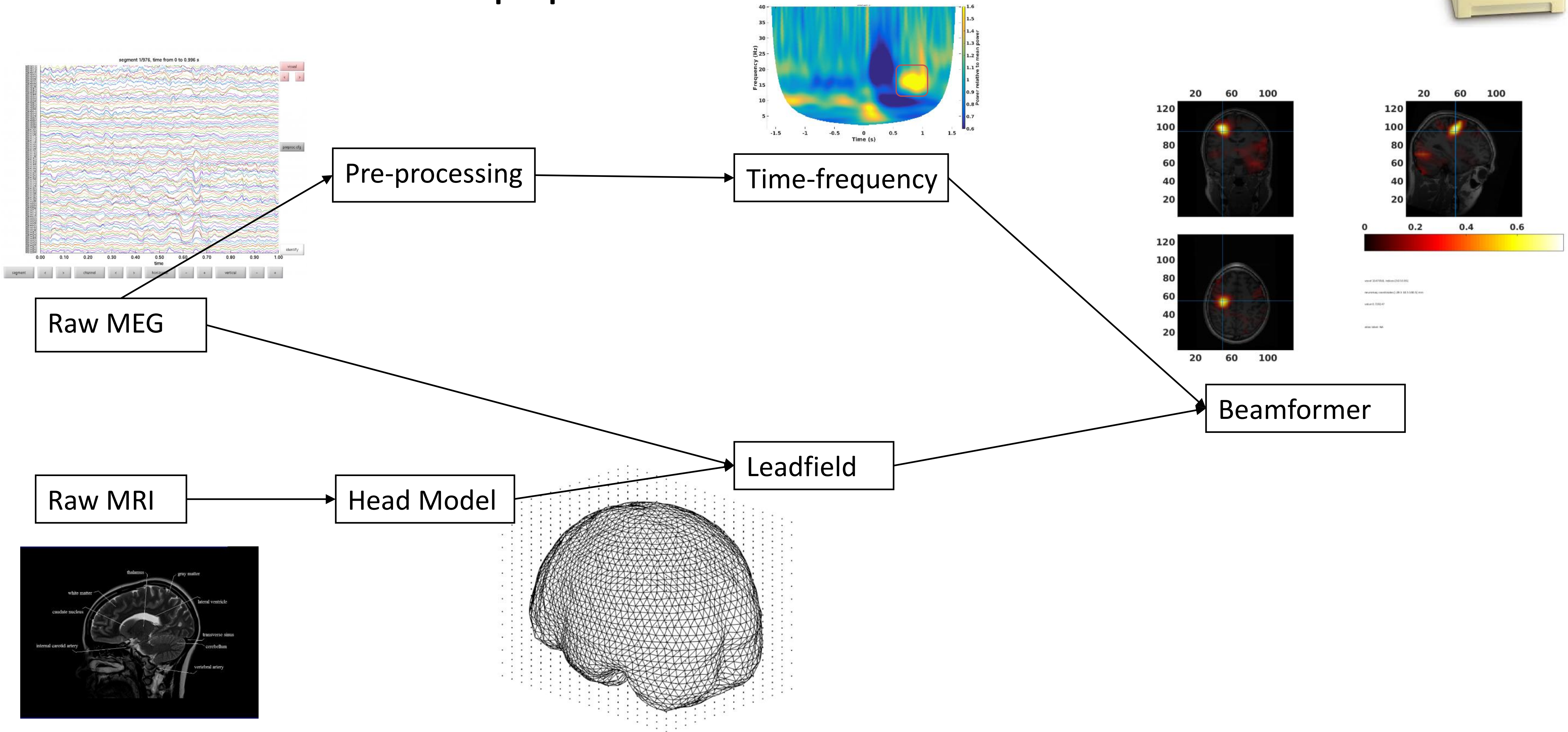
Look at the TFR data

- cfg = [];
- cfg.layout = 'neuromag306mag.lay';
- cfg.baseline = [-inf inf];
- cfg.baselinetype = 'relative';
- **ft_multiplotTFR**(cfg, combined_tfrs{1});

Beamformer

Locating oscillating activity

Overview of pipeline

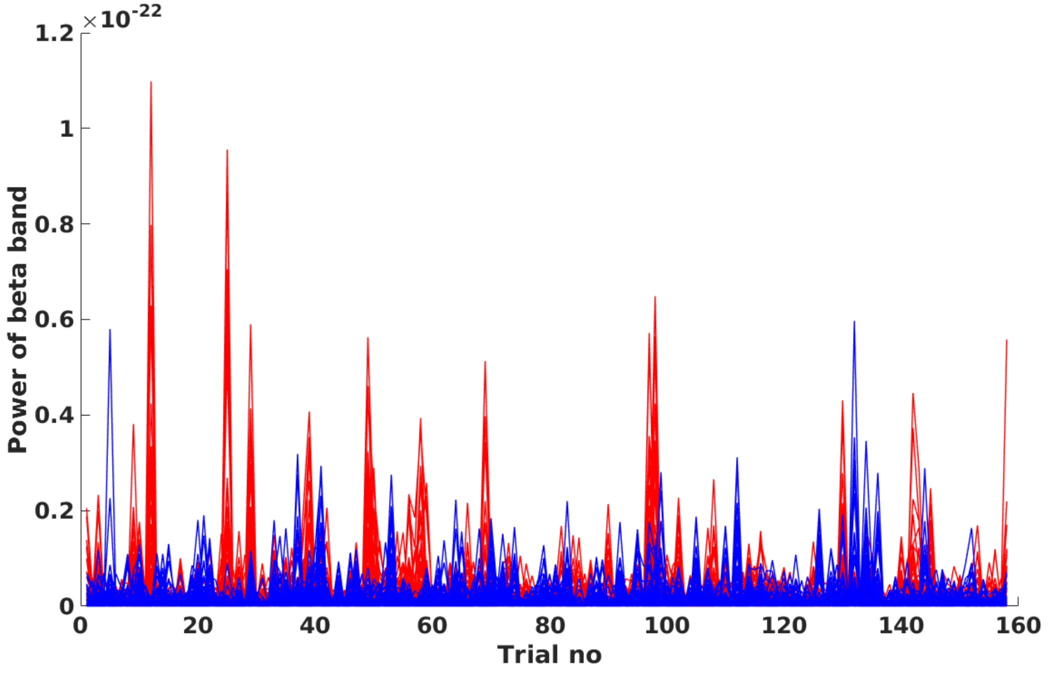


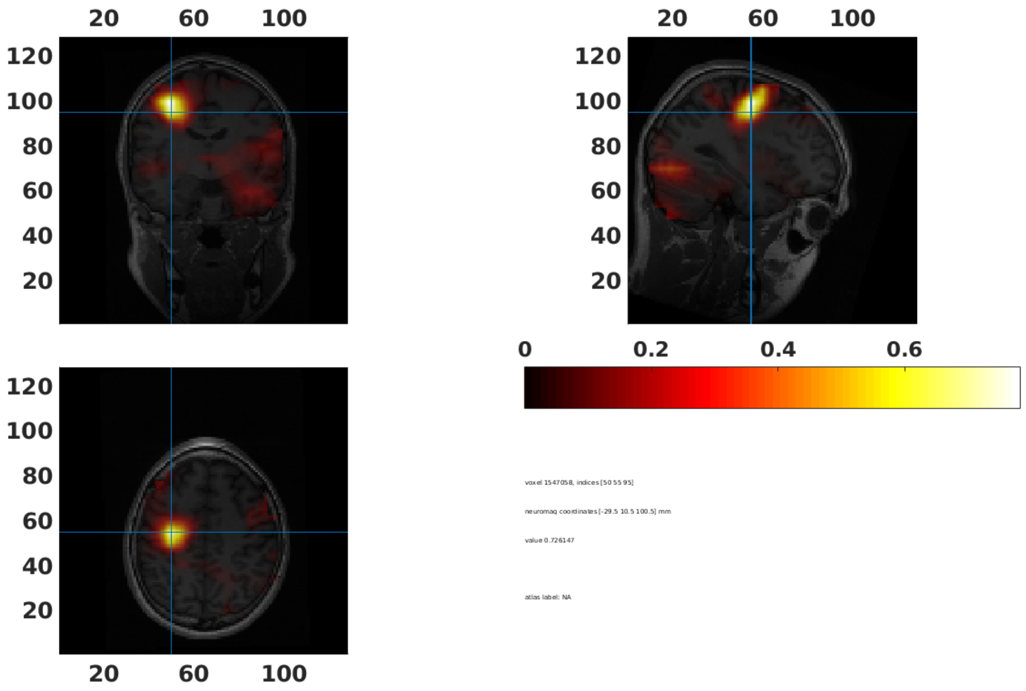
```
cfg = [];
cfg.method             = 'dics';        % Dynamic Imaging of Coherent Sources (Gross et al. 2001)
cfg.frequency          = freq;          %% the frequency from the Fourier analysis (16 Hz)
cfg.grid               = leadfield;     %% our grid and the leadfield
cfg.headmodel          = headmodel;     %% our headmodel (tells us how the magnetic field/electrical
                                         potential is propagated)
cfg.dics.projectnoise  = 'yes';         %% estimate noise
cfg.dics.lambda        = '10%';         %% how to regularise
cfg.dics.keepfilter    = 'yes';         %% keep the spatial filter in the output
cfg.dics.realfilter    = 'yes';         %% retain the real values
cfg.channel            = channels;
cfg.senstype           = sensor_type;
cfg.grad               = baseline_data.grad;
cfg.elec               = baseline_data.elec;

beamformer = ft_sourceanalysis(cfg, frequency_data);
```

Estimate the power for all the channels and each trial for a given band

Reconstruct the origin of these power changes in the brain





FieldTrip dipole structures

```
beamformer =
   freq:          15.6250            %% frequency reconstructed
   cumtapcnt:     [316x1 double]     %% ??
   dim:           [14 18 14]         %% dimension of the grid
   inside: [3528x1 logical] %% which grid positions are inside the brain
   pos:           [3528x3 double]    %% grid point positions
   method:        'average'          %% method used
   avg:           [1x1 struct]       %% the averages
   cfg:           [1x1 struct]       %% the configuration used
```