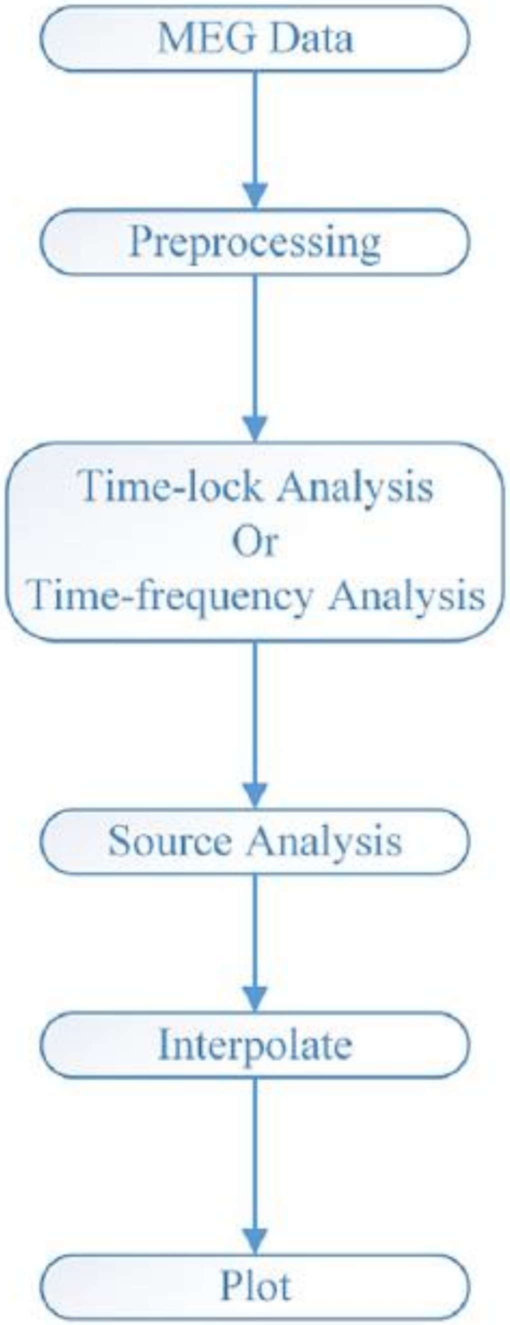
Objective

- Familiarize with EasyMEG

(*EasyMEG: An Easy-To-Use Toolbox for MEG Analysis: Comput Methods Programs Biomed 2020 186:105199*)

- Use EasyMEG for preprocessing and event-related analysis of MEG data
- Use EasyMEG for source analysis (minimum-norm estimate) of event-related MEG data


MEG Data

Preprocessing

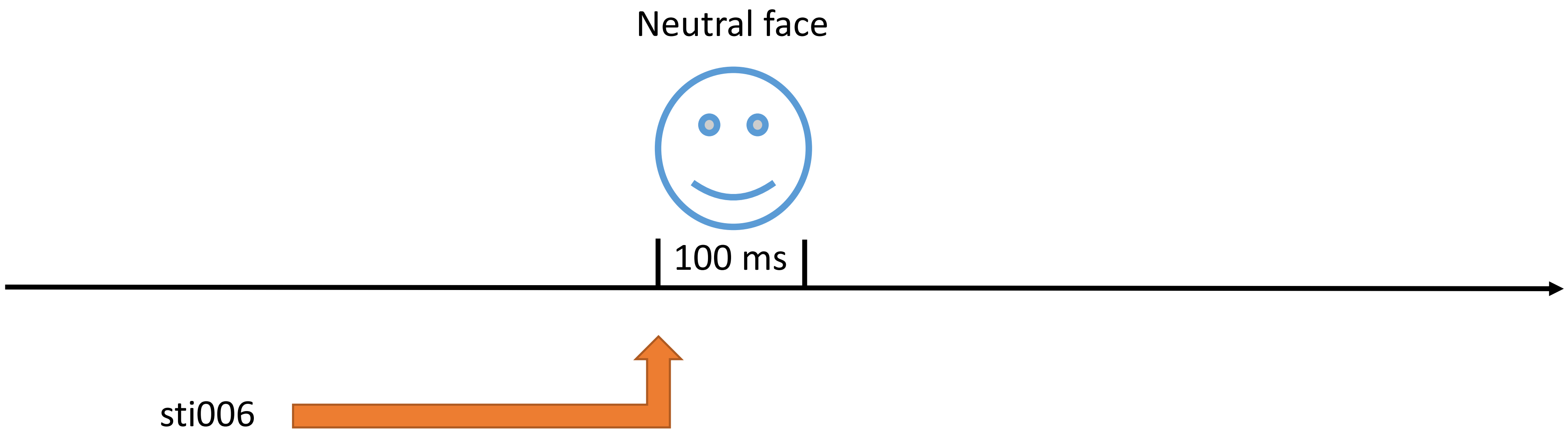
Time-lock Analysis
Or
Time-frequency Analysis

Source Analysis

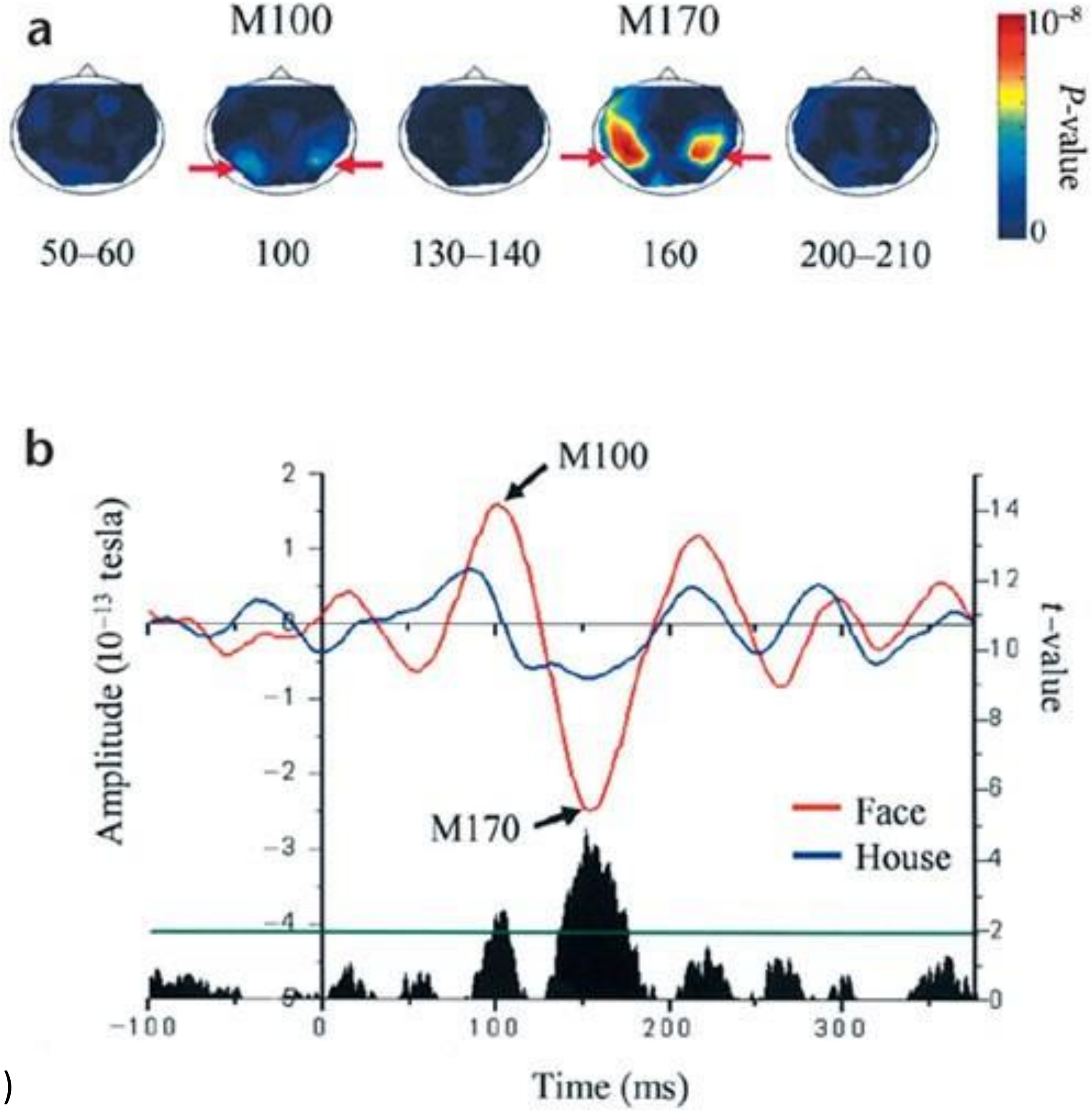
Interpolate

Plot

Sample dataset



Ideal results



Liu et al. (2002)

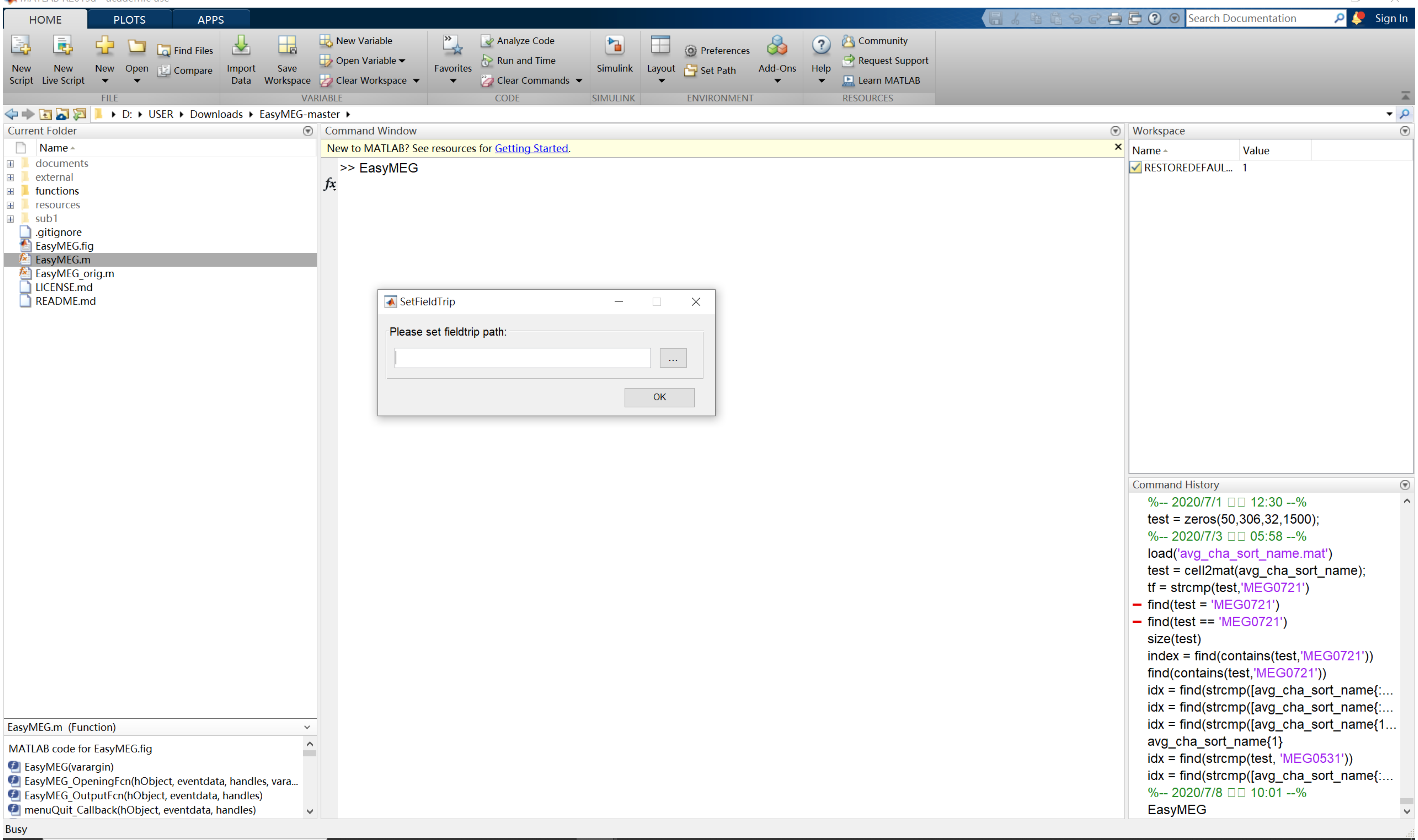
Modifications

```
% dataDir = uigetdir('.','Pick a Neuromag data directory');
[filename, pathname]  = uigetfile('*.fif', 'Pick a Neuromag dataset');
dataDir = fullfile(pathname, filename);
```

```
%     data = ft_redefinetrial(cfg,data);
    data = ft_preprocessing(cfg,data);
```

```
%     cfg.coordsys = 'ctf';
cfg.coordsys   = 'neuromag';
```

```
if meg
      cfg.megscale = meg;
      cfg.alim     = 5e-11;
```



Current Folder

- Name
- documents
- external
- functions
- resources
- sub1
- .gitignore
- EasyMEG.fig
- EasyMEG.m
- EasyMEG_orig.m
- LICENSE.md
- README.md

EasyMEG.m (Function)

MATLAB code for EasyMEG.fig

```
EasyMEG(varargin)
EasyMEG_OpeningFcn(hObject, eventdata, handles, vara...
EasyMEG_OutputFcn(hObject, eventdata, handles)
menuQuit_Callback(hObject, eventdata, handles)
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> EasyMEG
```

SetFieldTrip

Please set fieldtrip path:

[ ] ...

OK

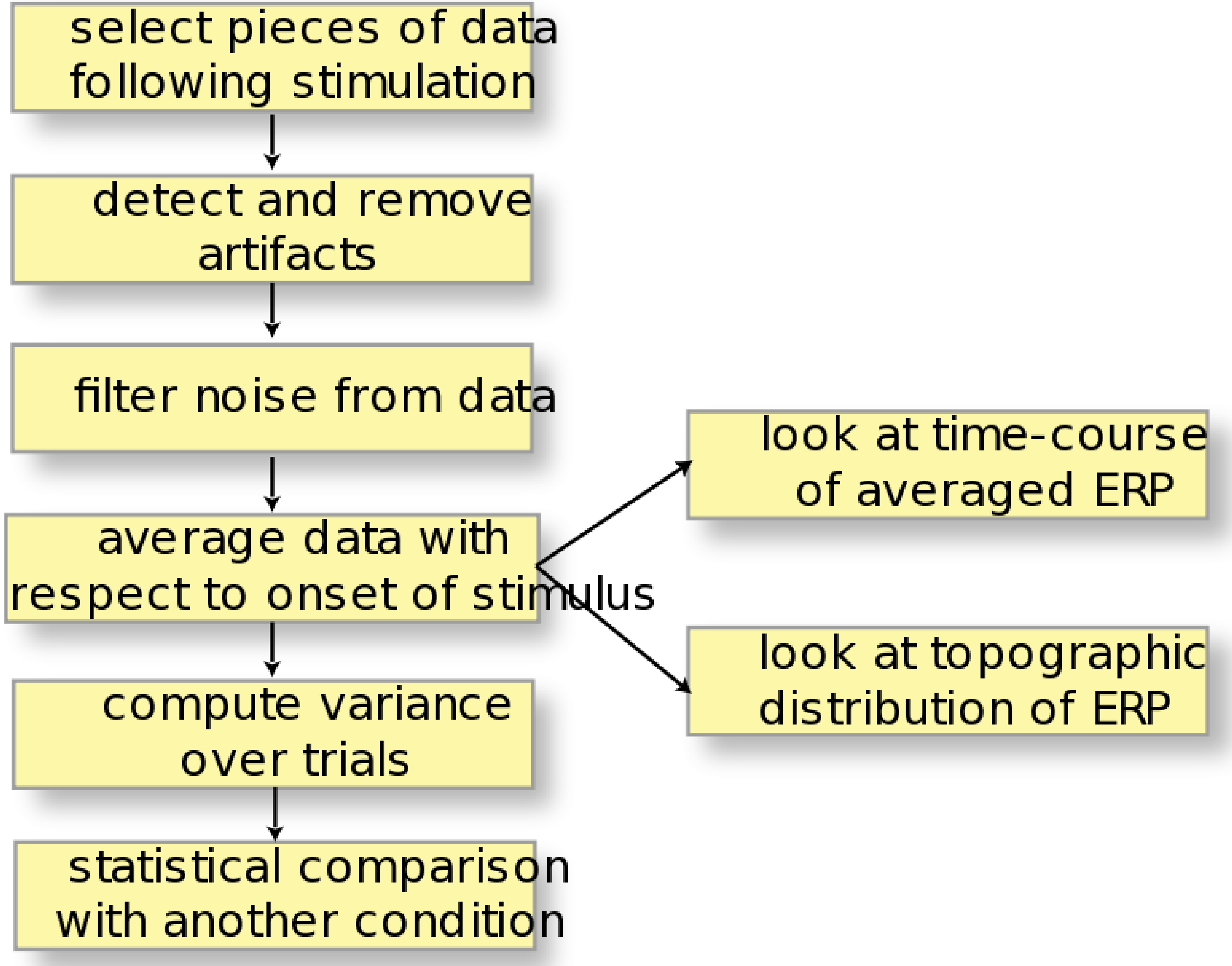
Workspace

Name	Value
RESTOREDEFAUL...	1

Command History

```
%-- 2020/7/1 □□ 12:30 --%
test = zeros(50,306,32,1500);
%-- 2020/7/3 □□ 05:58 --%
load('avg_cha_sort_name.mat')
test = cell2mat(avg_cha_sort_name);
tf = strcmp(test,'MEG0721')
find(test = 'MEG0721')
find(test == 'MEG0721')
size(test)
index = find(contains(test,'MEG0721'))
find(contains(test,'MEG0721'))
idx = find(strcmp([avg_cha_sort_name{:...
idx = find(strcmp([avg_cha_sort_name{:...
idx = find(strcmp([avg_cha_sort_name{1...
avg_cha_sort_name{1}
idx = find(strcmp(test, 'MEG0531'))
idx = find(strcmp([avg_cha_sort_name{:...
%-- 2020/7/8 □□ 10:01 --%
EasyMEG
```

ERF pipeline



select pieces of data following stimulation

detect and remove artifacts

filter noise from data

average data with respect to onset of stimulus

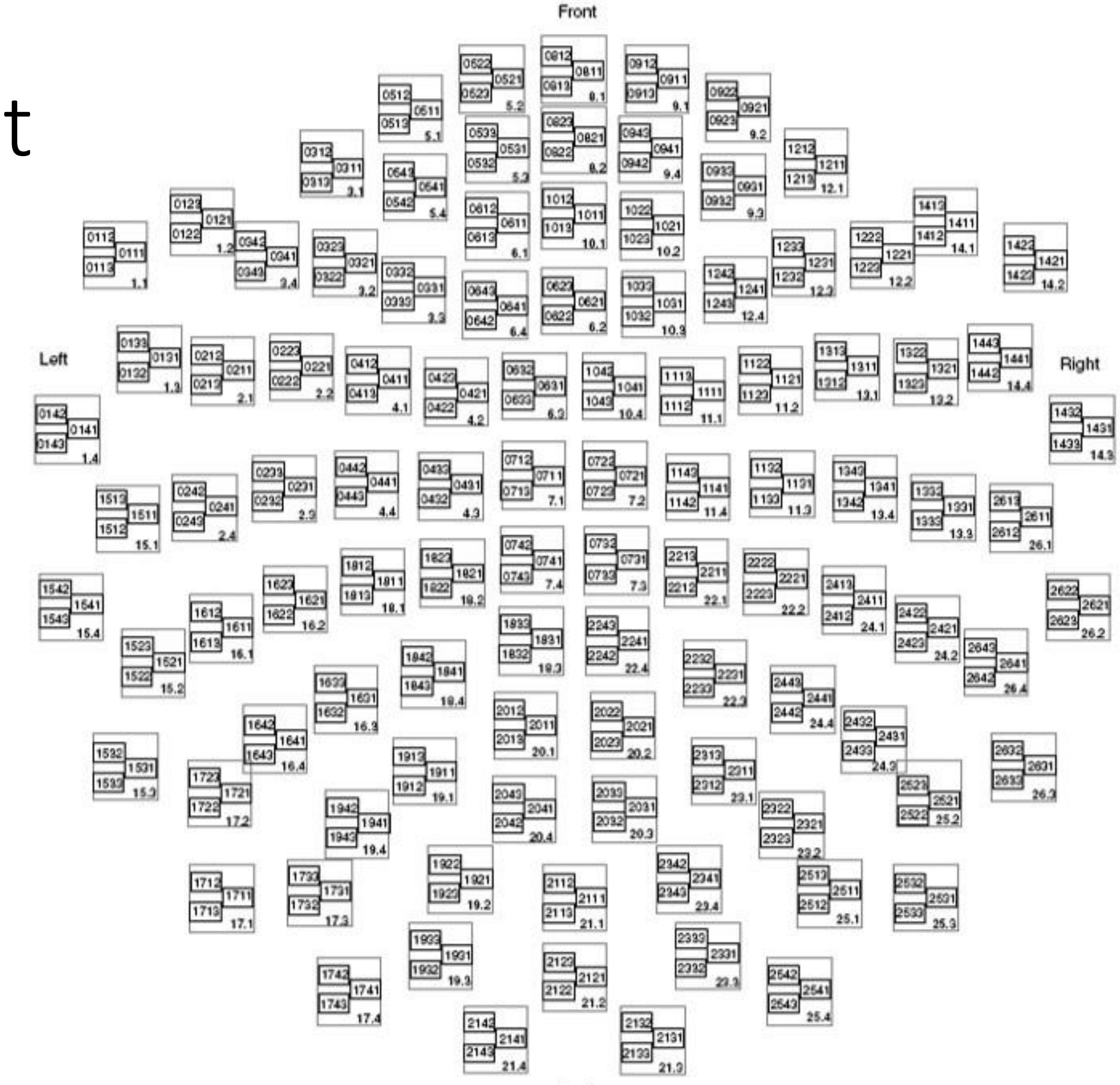
compute variance over trials

statistical comparison with another condition

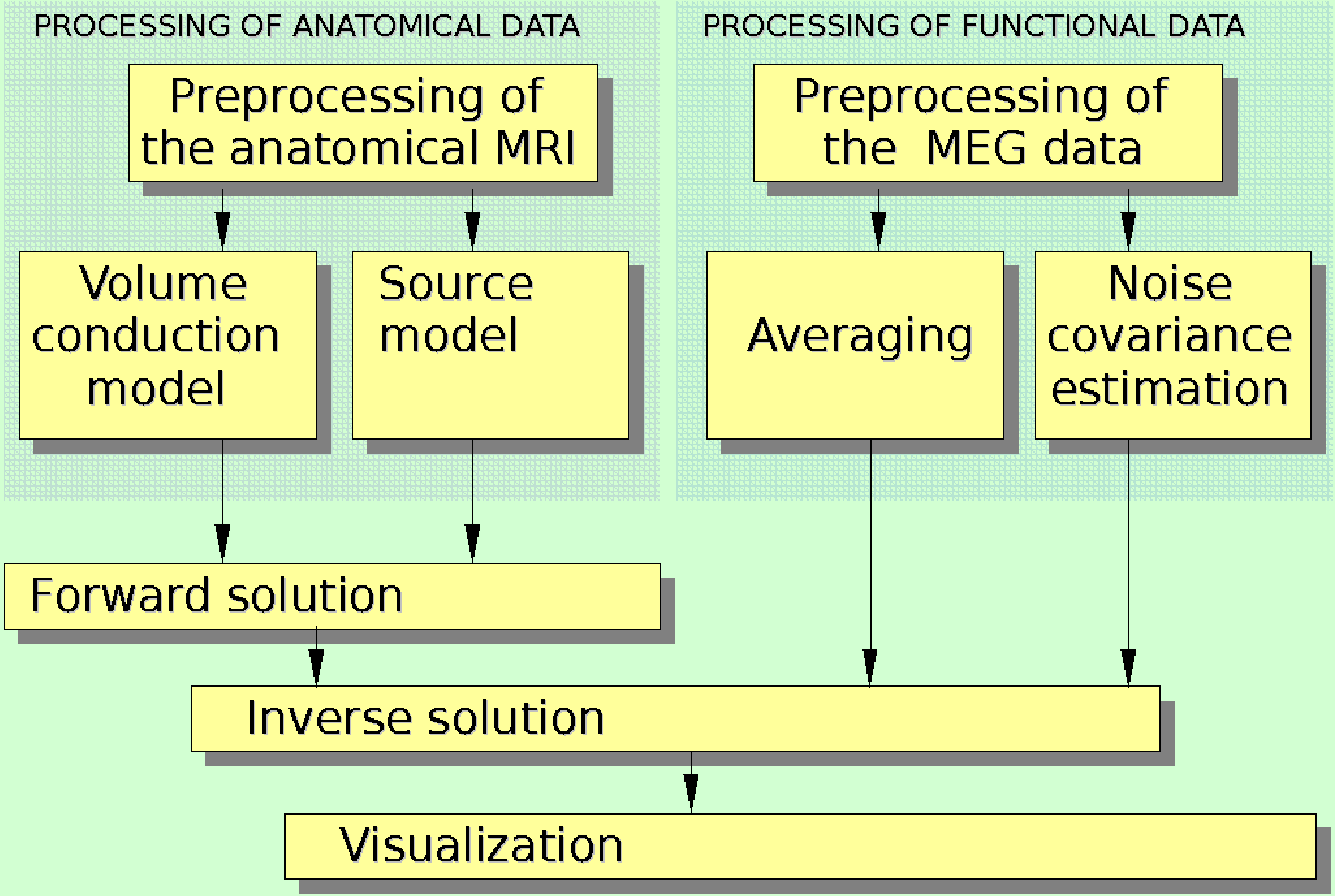
look at time-course of averaged ERP

look at topographic distribution of ERP

Neuromag layout



MNE, SOURCE RECONSTRUCTION PIPELINE



PROCESSING OF ANATOMICAL DATA

Preprocessing of the anatomical MRI

Volume conduction model

Source model

PROCESSING OF FUNCTIONAL DATA

Preprocessing of the MEG data

Averaging

Noise covariance estimation

Forward solution

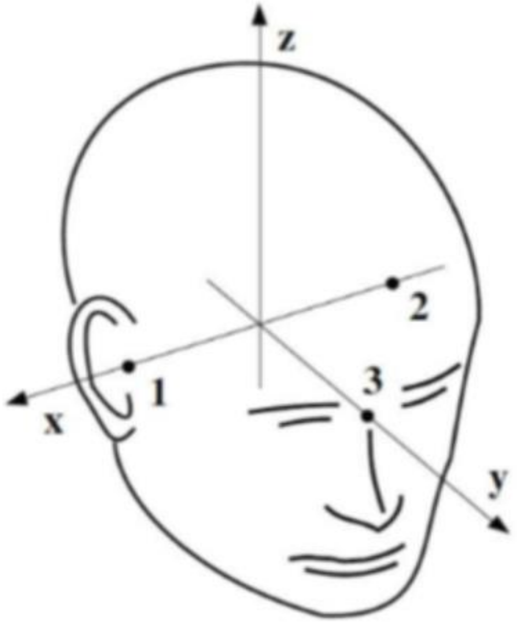
Inverse solution

Visualization

Details of the Neuromag coordinate system

The **Neuromag** coordinate system is expressed in meter, with the principal (X, Y, Z) axes going through external landmarks (i.e. fiducials). The details are

- X-axis from the origin towards the RPA point (exactly through)
- Y-axis from the origin towards the nasion (exactly through)
- Z-axis from the origin upwards orthogonal to the XY-plane
- Origin: Intersection of the line through LPA and RPA and a line orthogonal to L passing through the nasion.



```
sourcemodel = ft_read_headshape('D:\fieldtrip-20200224\template\sourcemodel\cortex_5124.surf.gii');
template_grid = sourcemodel;
mri= dataSet{1,1}.mri;

cfg           = [];
cfg.warpmni   = 'yes';
cfg.template  = template_grid;
cfg.nonlinear = 'yes';
cfg.mri       = mri;
cfg.unit      ='mm';
sourcemodel         = ft_prepare_sourcemodel(cfg);

headmodel = dataSet{1,1}.headmodel;
grad = dataSet{1,1}.timelock.grad;
label = dataSet{1,1}.timelock.label;

cfg         = [];
cfg.grad    = grad;   % sensor information
cfg.channel = label;  % the used channels
cfg.grid    = sourcemodel;   % source points
cfg.headmodel = headmodel;   % volume conduction model
cfg.singleshell.batchsize = 5000; % speeds up the computation
leadfield  = ft_prepare_leadfield(cfg);
```

```
timelock = dataSet{1,1}.timelock;
cfg              = [];
cfg.method        = 'mne';
cfg.grid          = leadfield;
cfg.headmodel     = headmodel;
cfg.mne.prewhiten = 'yes';
cfg.mne.lambda    = 3;
cfg.mne.scalesourcecov = 'yes';
source          = ft_sourceanalysis(cfg,timelock);

%%
m=source.avg.pow(:,671);
ft_plot_mesh(source, 'vertexcolor', m);
% view([180 0]); h = light; set(h, 'position', [0 1 0.2]);

cfg = [];
cfg.projectmom = 'yes';
sd  = ft_sourcedescriptives(cfg,source);

cfg = [];
cfg.funparameter = 'pow';
ft_sourcemovie(cfg,sd);
```