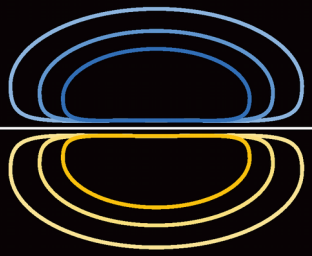
Karolinska Institutet

Seminar on:

Open data – Standardized analysis pipeline for Magnetoencephalography (MEG)

By Lau M. Andersen



NatMEG

The Swedish National Facility for Magnetoencephalography

PROTOCOLS ARTICLE

Front. Neurosci., 22 January 2018 | https://doi.org/10.3389/fnins.2018.00006

Group Analysis in MNE-Python of Evoked Responses from a Tactile Stimulation Paradigm: A Pipeline for Reproducibility at Every Step of Processing, Going from Individual Sensor Space Representations to an across-Group Source Space Representation

Lau M. Andersen*

NatMEG, Department of Clinical Neuroscience, Karolinska Institutet, Stockholm, Sweden

MNE
MEG + EEG ANALYSIS & VISUALIZATION

Open-source Python software for exploring, visualizing, and analyzing human neurophysiological data: MEG, EEG, sEEG, ECoG, and more.

PROTOCOLS ARTICLE

Front. Neurosci., 01 May 2018 | https://doi.org/10.3389/fnins.2018.00261

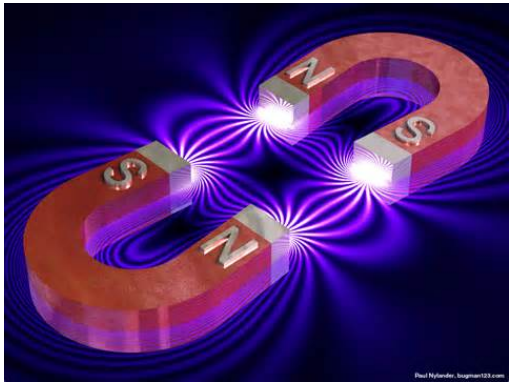
Group Analysis in FieldTrip of Time-Frequency Responses: A Pipeline for Reproducibility at Every Step of Processing, Going From Individual Sensor Space Representations to an Across-Group Source Space Representation

Lau M. Andersen*

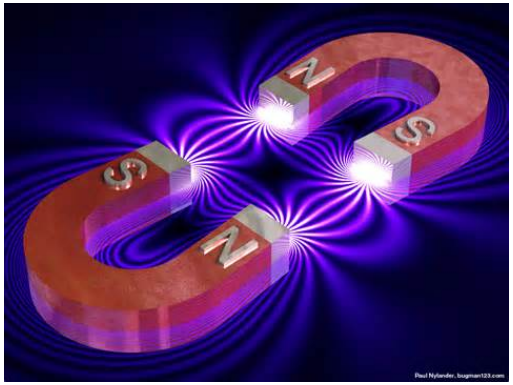
NatMEG, Department of Clinical Neuroscience, Karolinska Institutet, Stockholm, Sweden

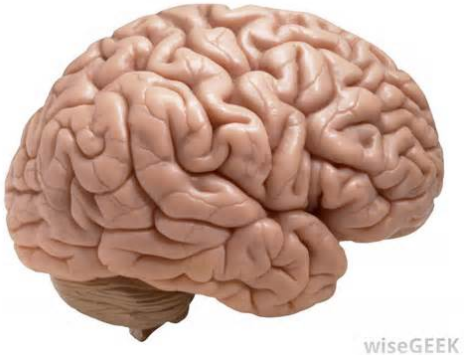
Magneto·encephalo·graphy

Magneto·encephalo·graphy



Magneto·encephalo·graphy





Writing out the *magnetic fields* of the *brain*

Goals for today

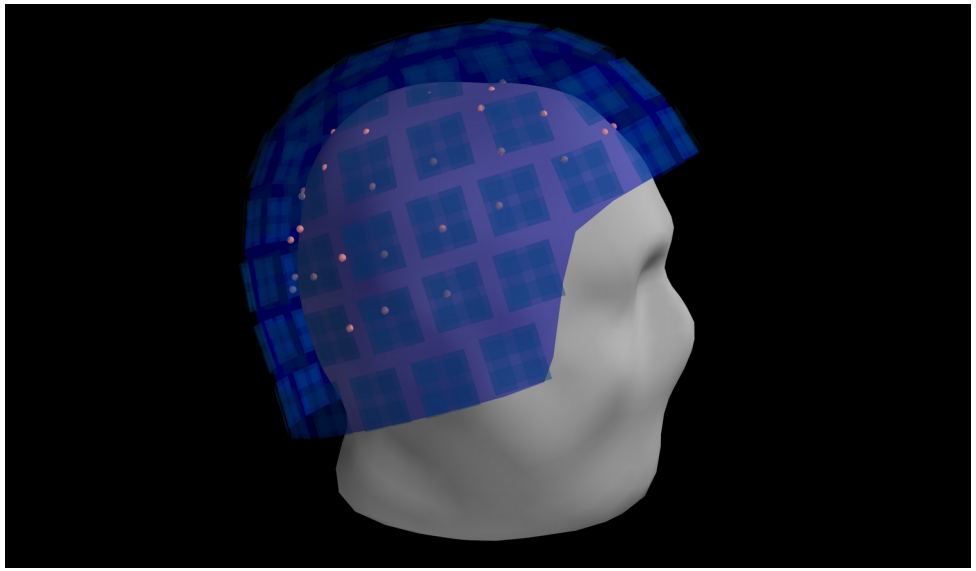
1) (*What is MEG?*) You are to be able to recognize MEG signals

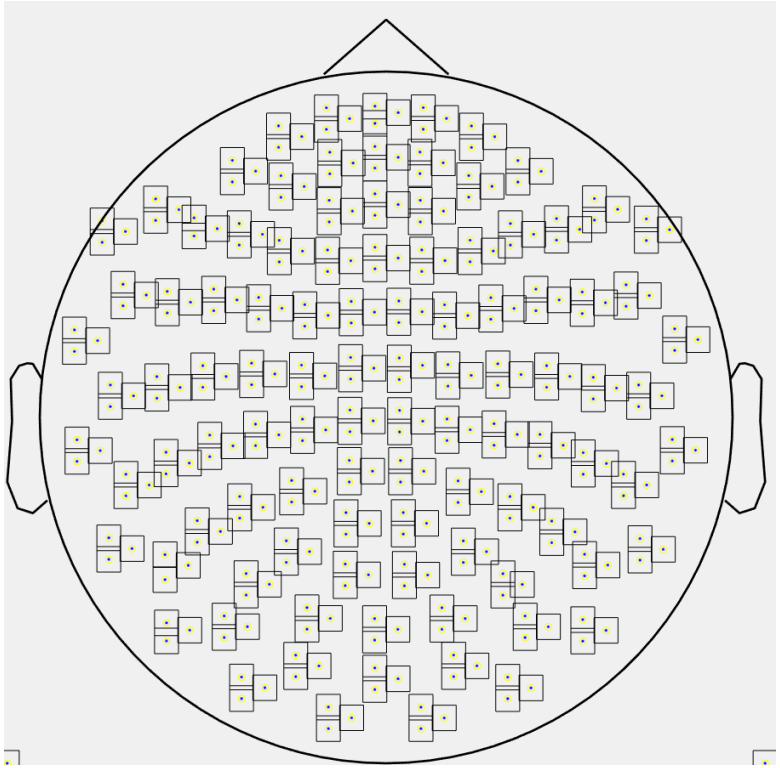
2) (*How to analyse MEG*) You are to be able to find relevant resources for group level analysis, so you can get started yourself

3) (How to decide about MEG) You are to know about statistical tools for answering questions about MEG

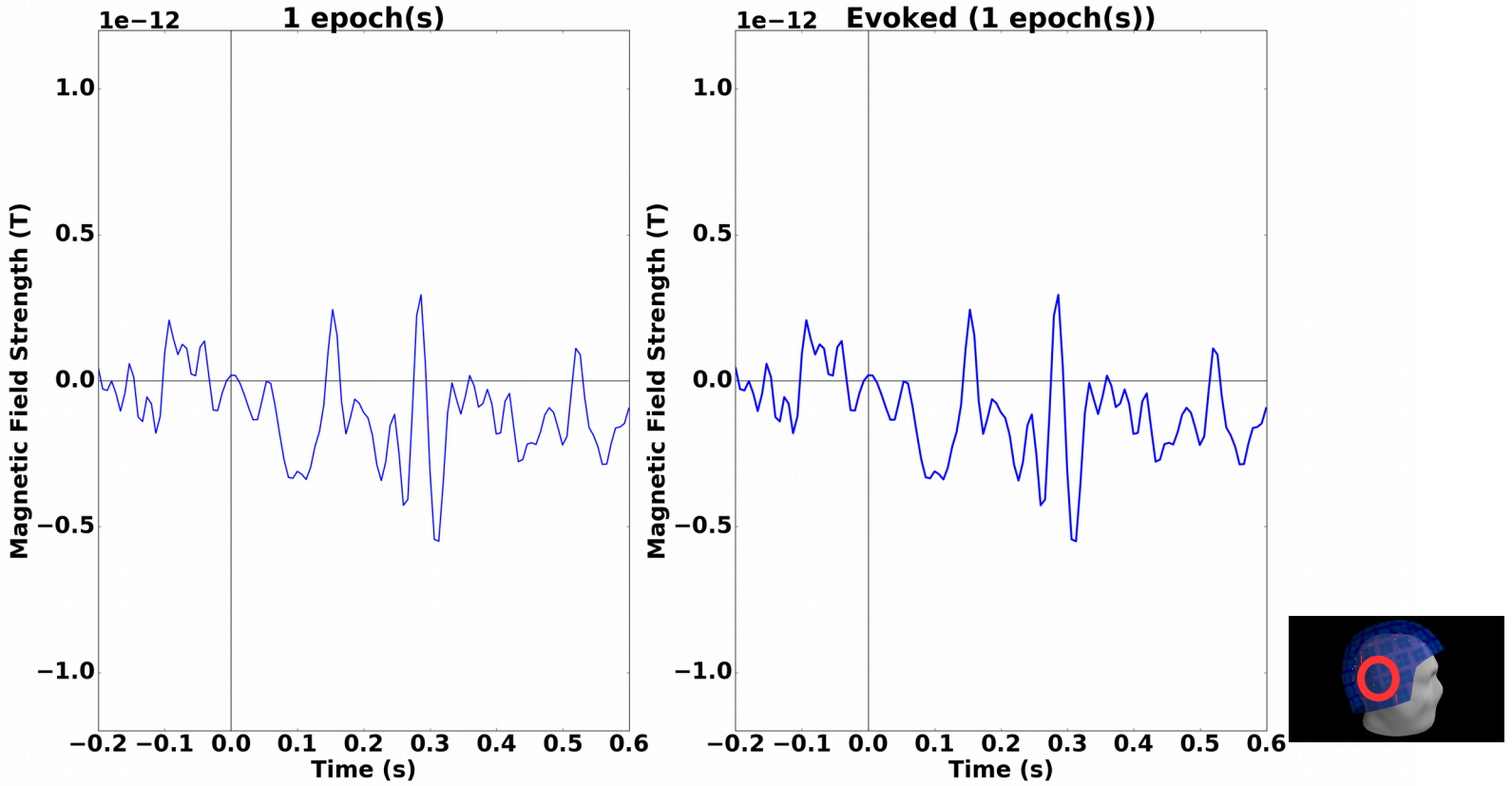
How do we record it?

How do we record it?

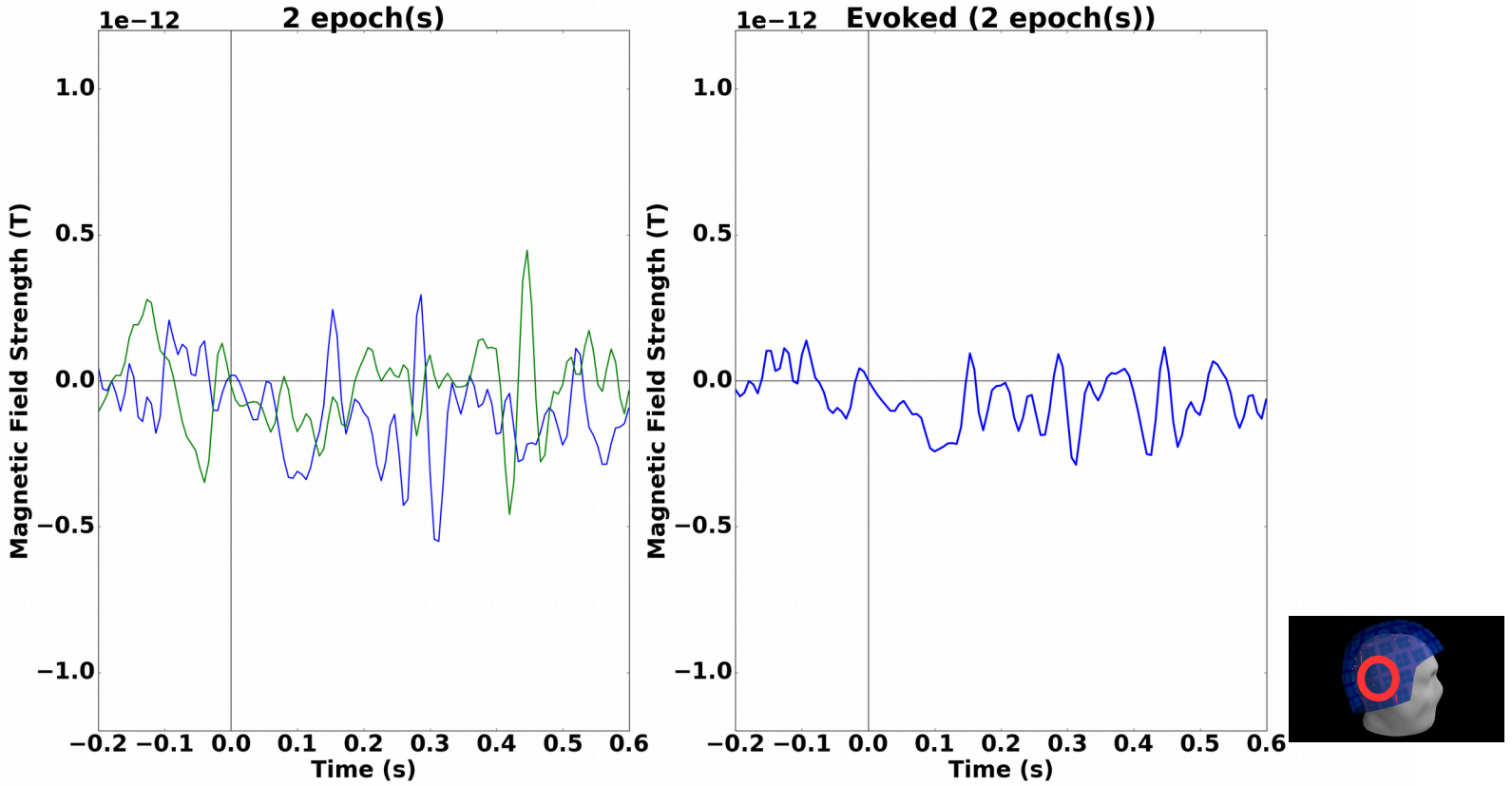




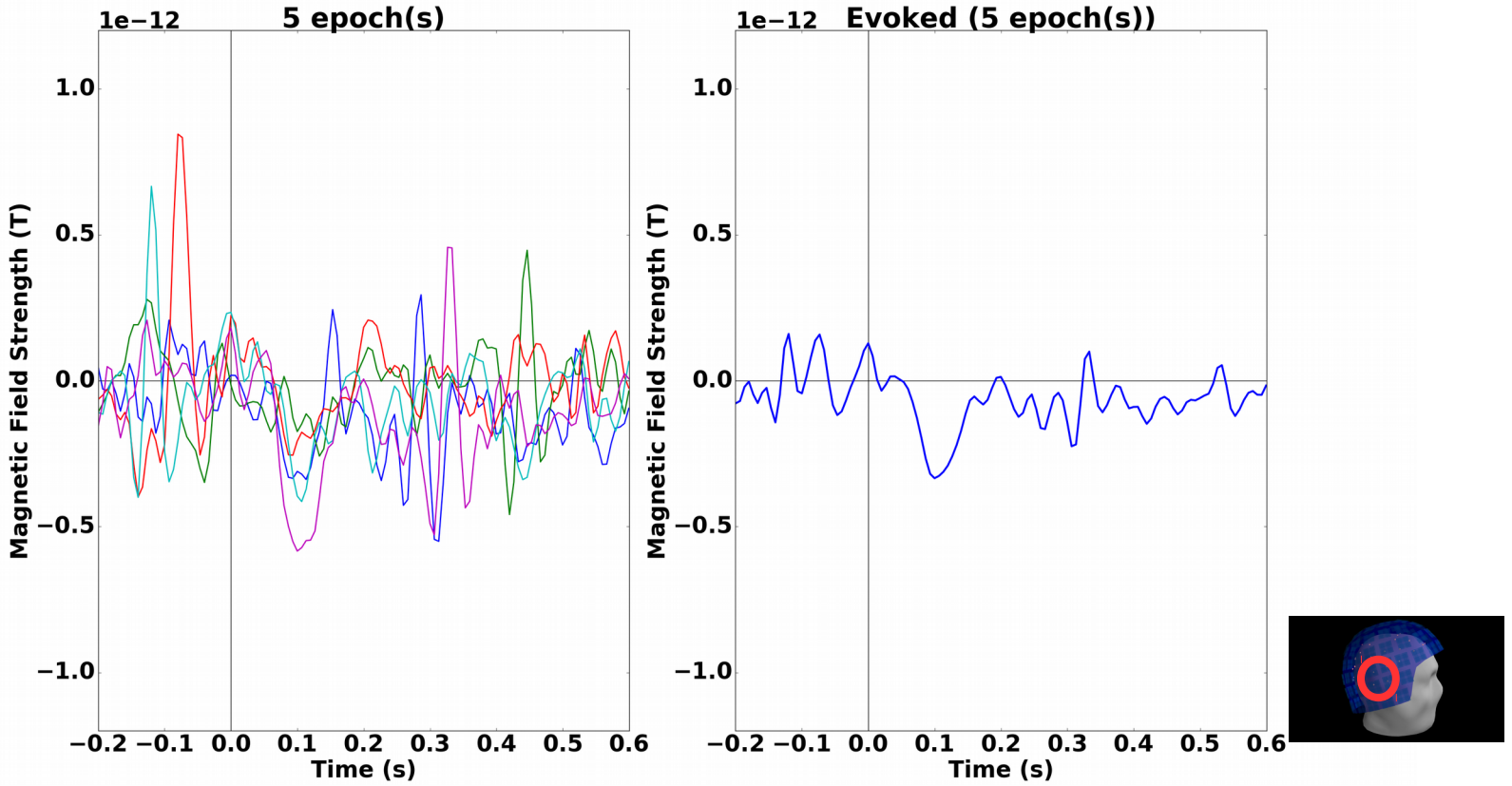
EPOCHS AND EVOKED



EPOCHS AND EVOKED

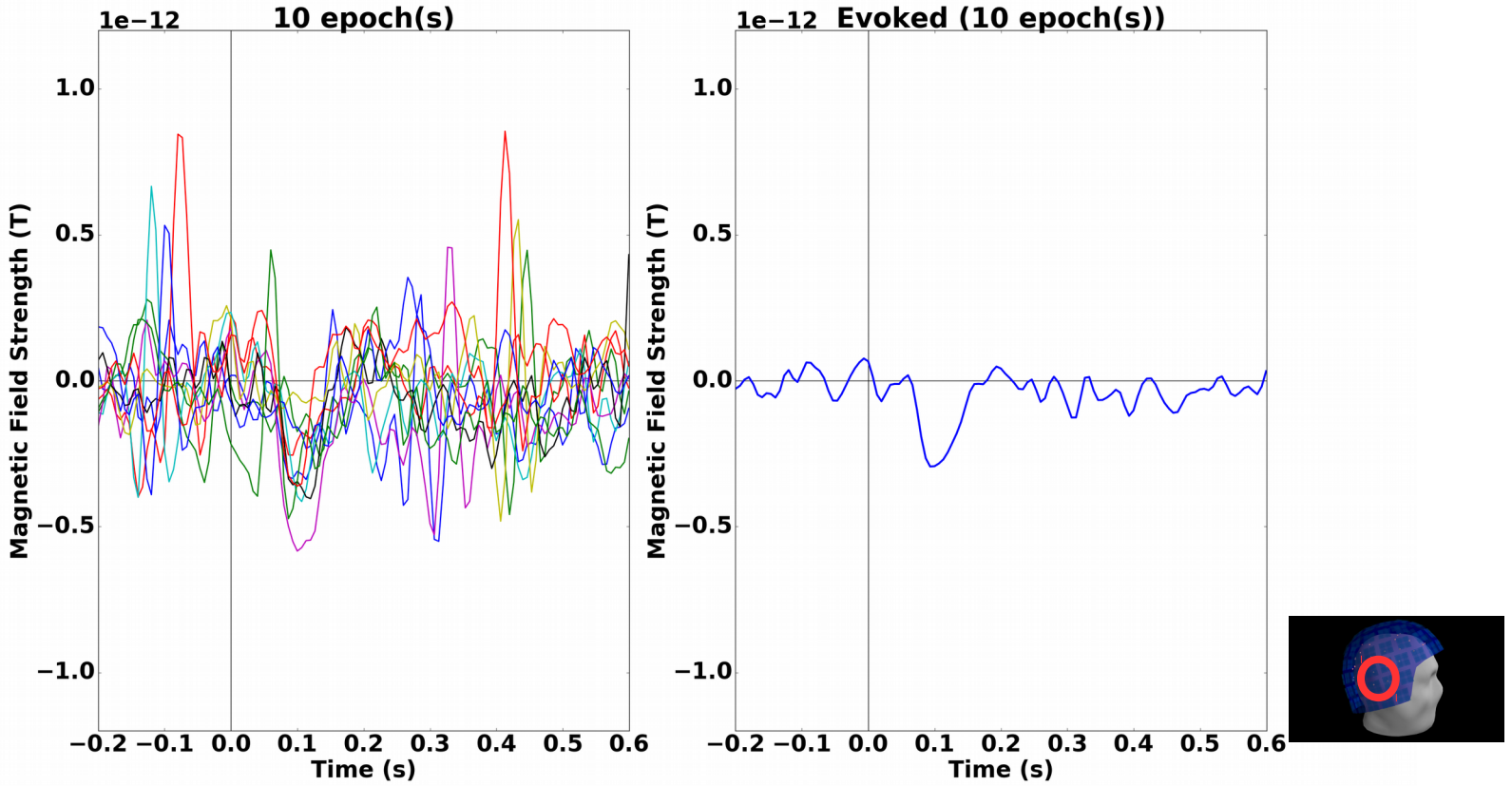


EPOCHS AND EVOKED



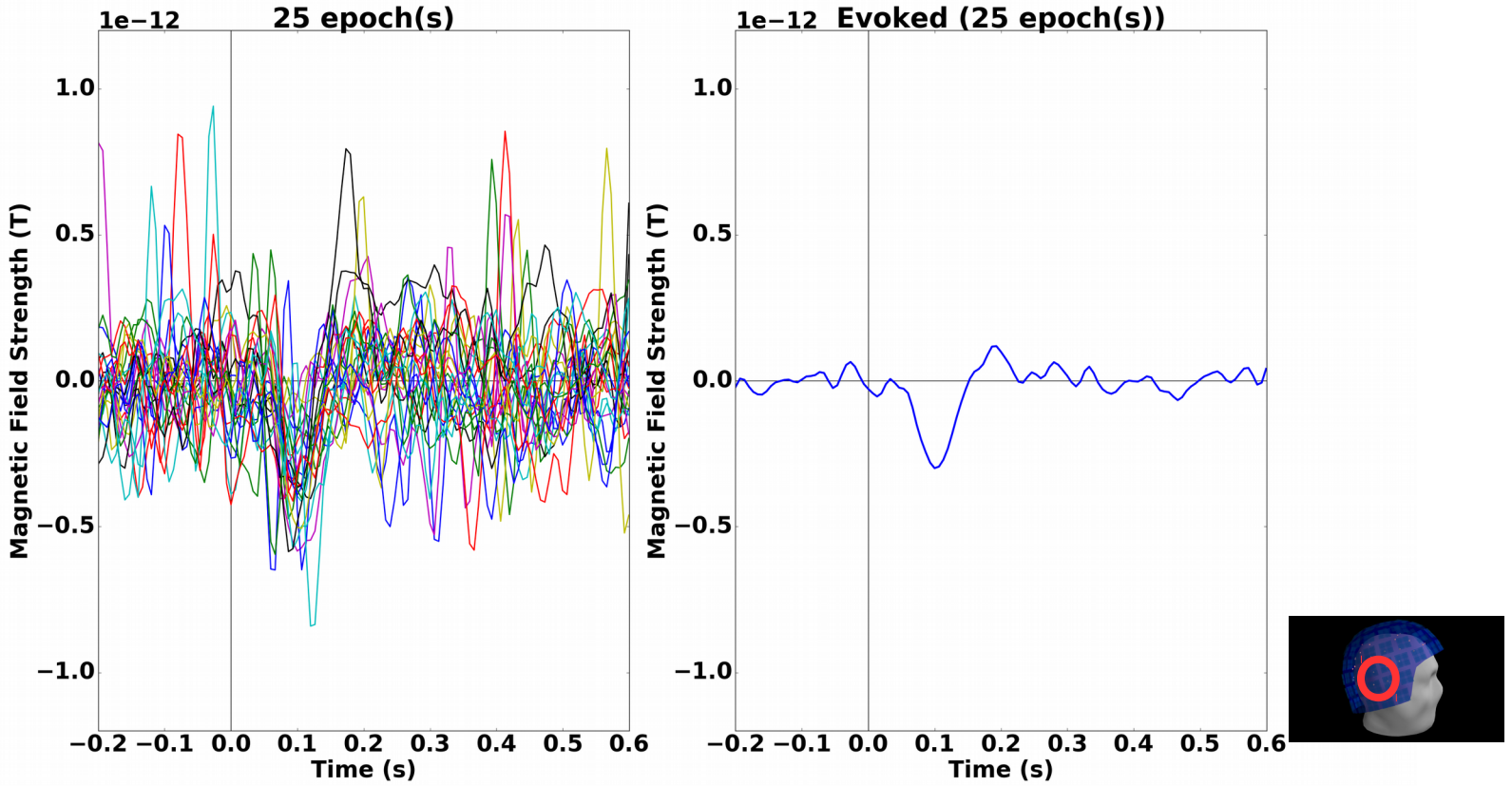
Karolinska Institutet

EPOCHS AND EVOKED



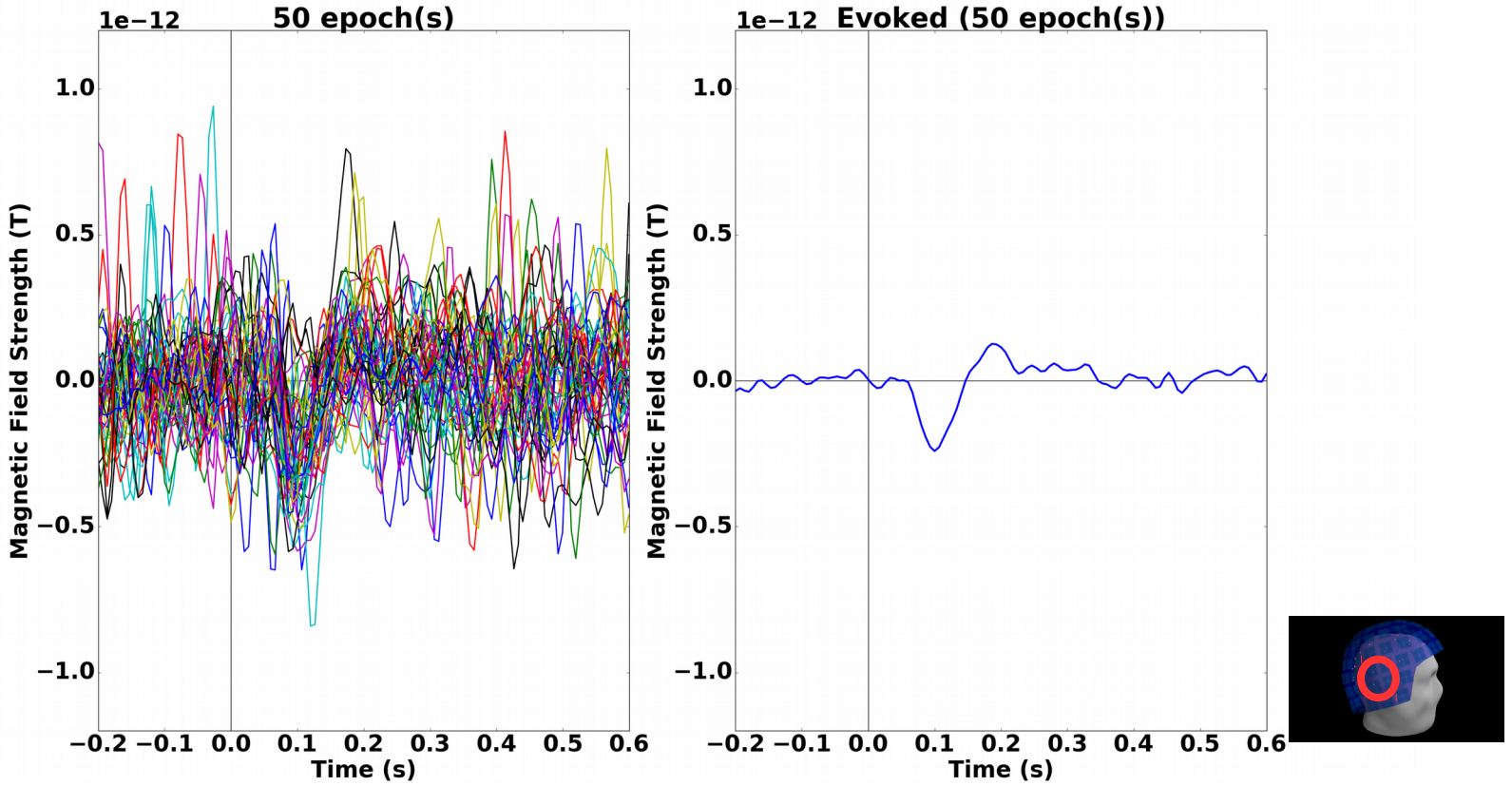
Karolinska Institutet

EPOCHS AND EVOKED



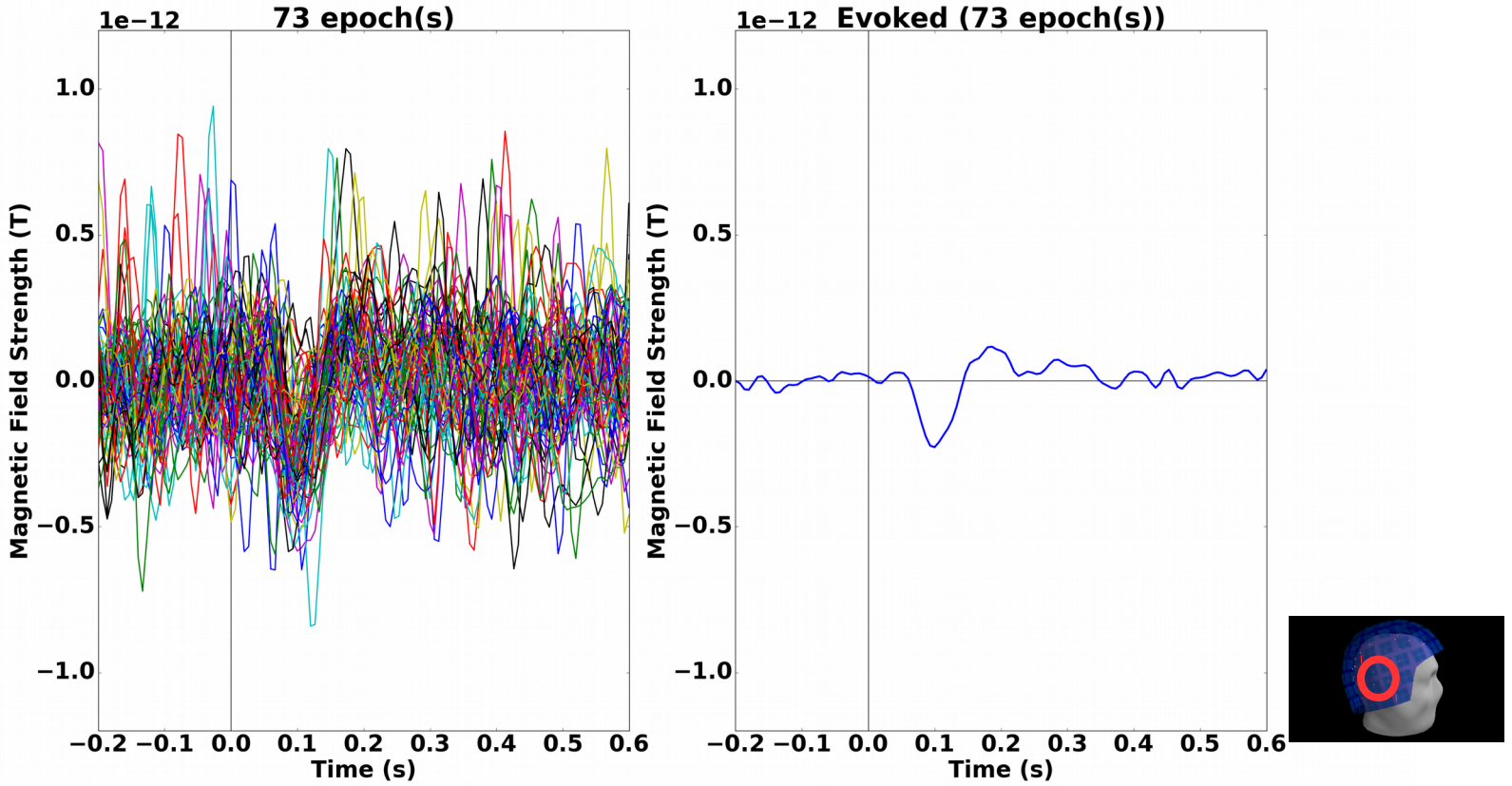
Karolinska Institutet

EPOCHS AND EVOKED



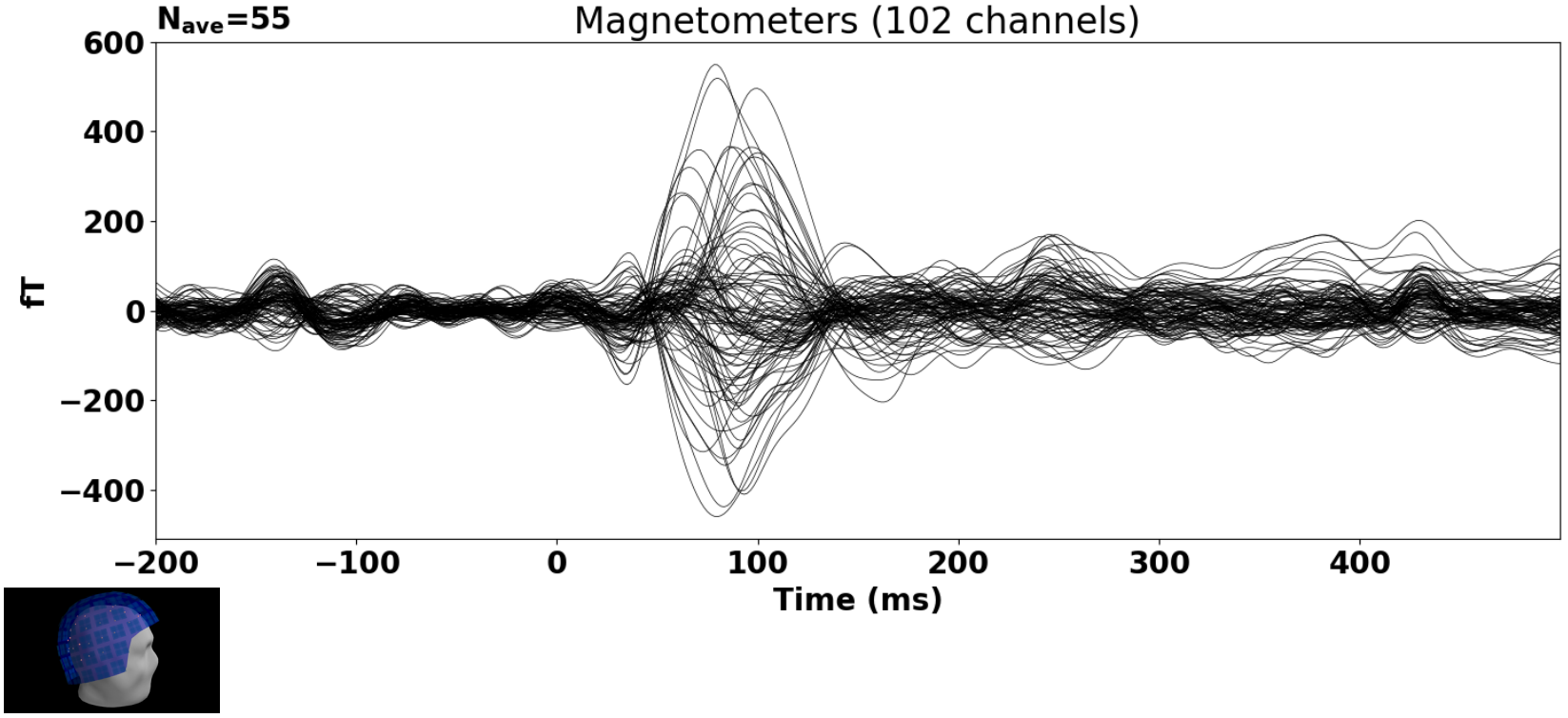
Karolinska Institutet

EPOCHS AND EVOKED



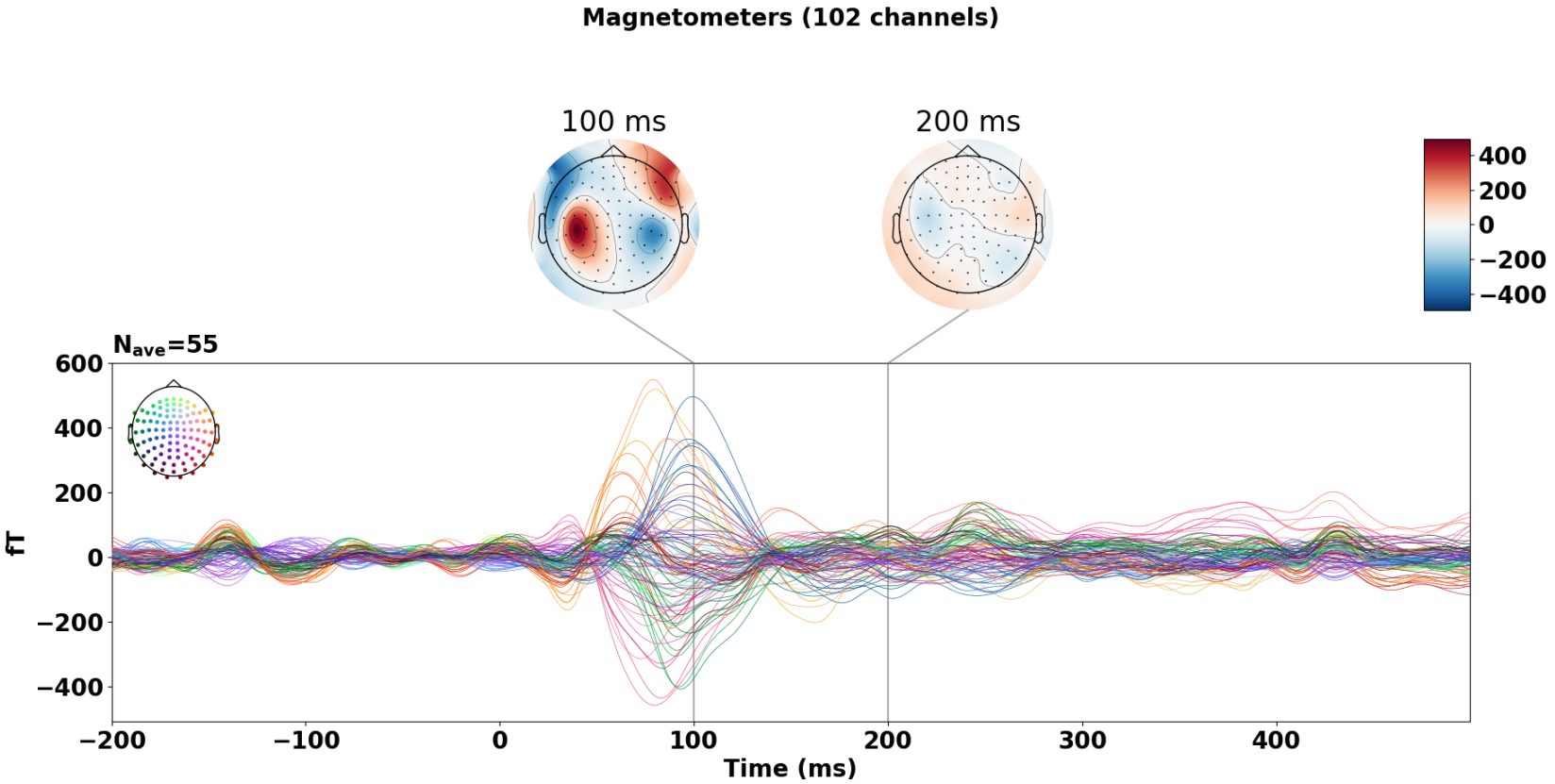
Karolinska Institutet

What does the signal look like?

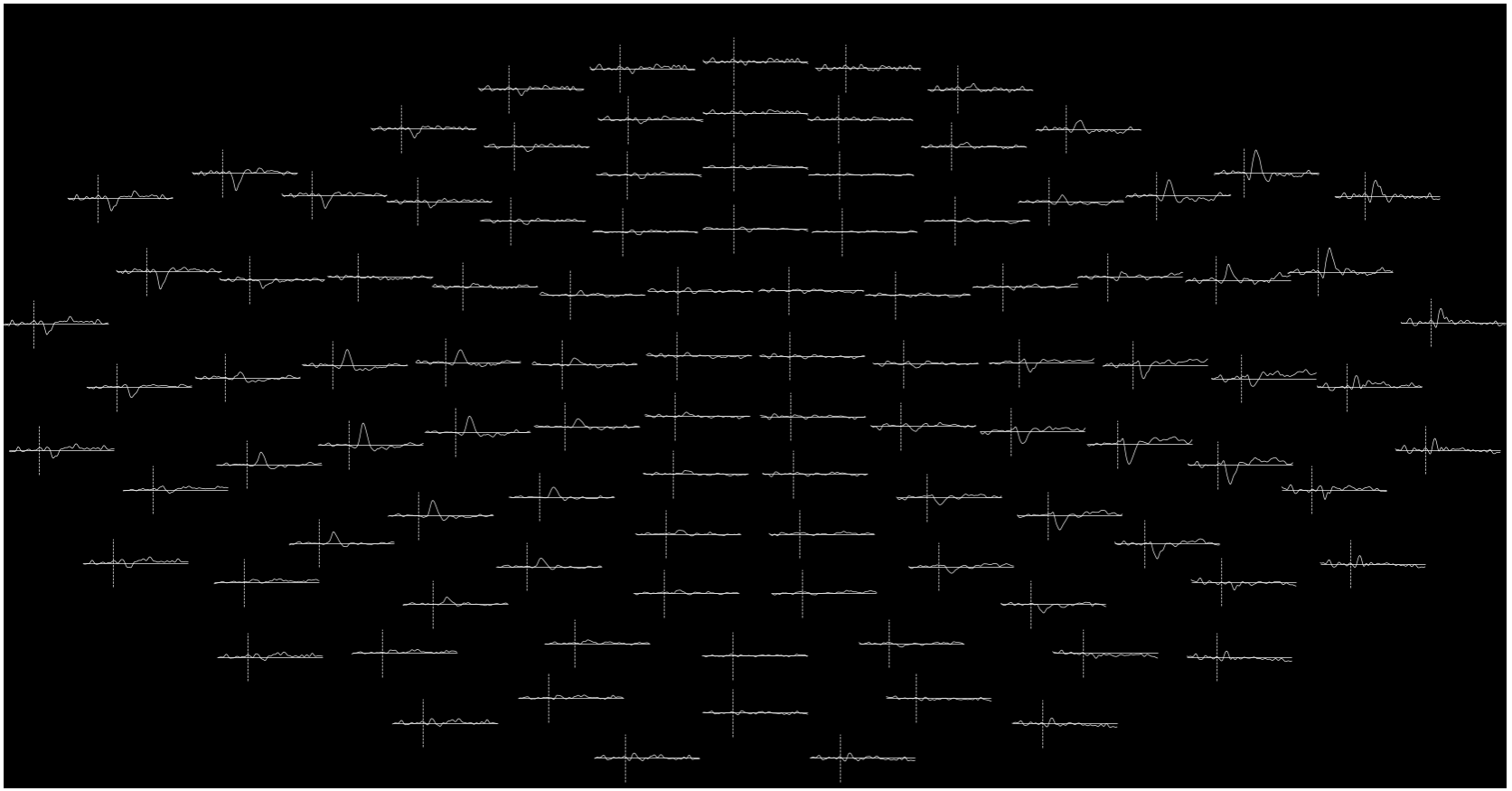


Karolinska Institutet

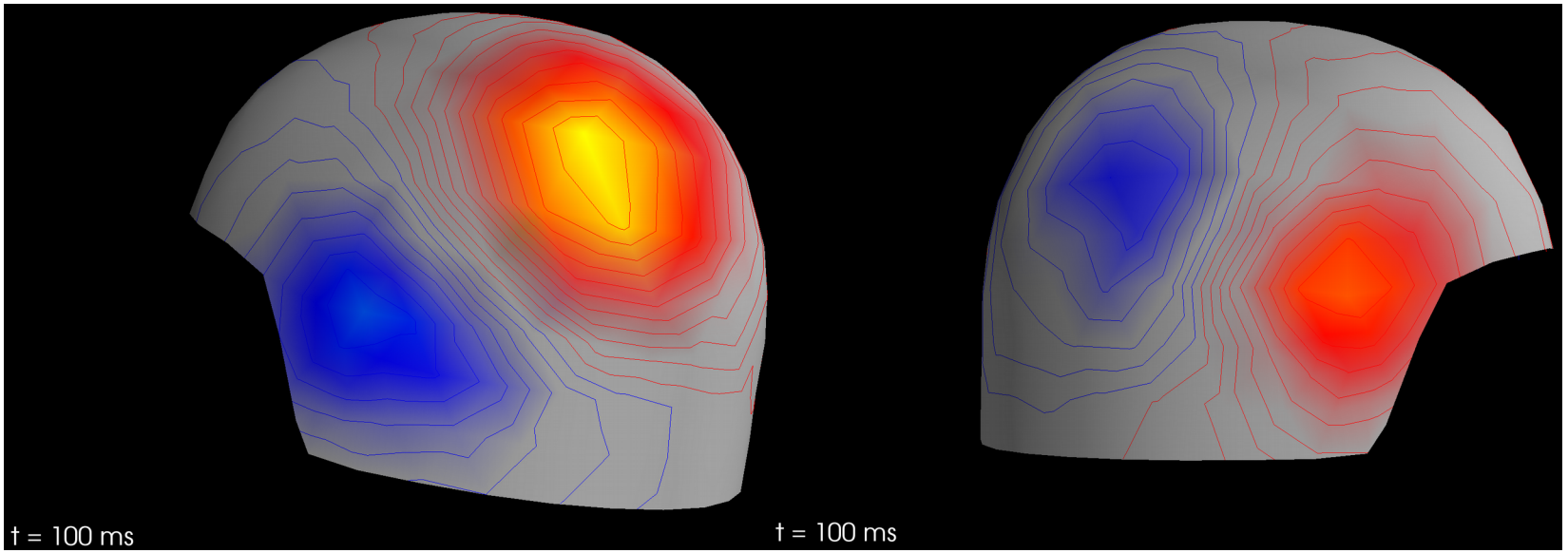
What does the signal look like?

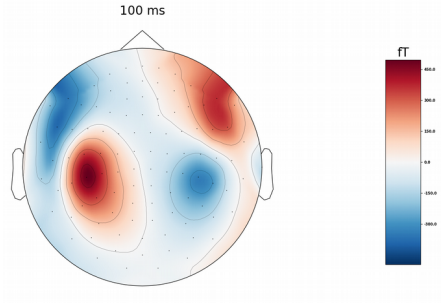


What does the signal look like?



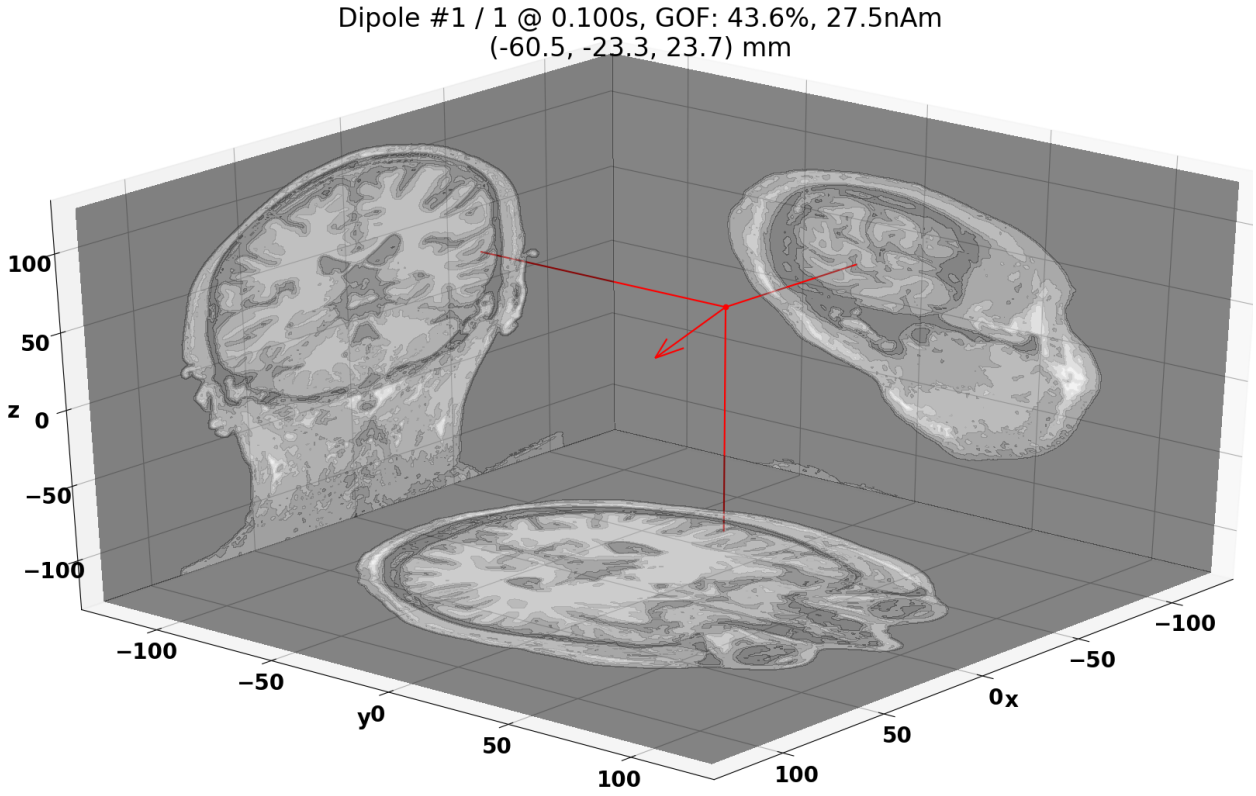
What does the signal look like?



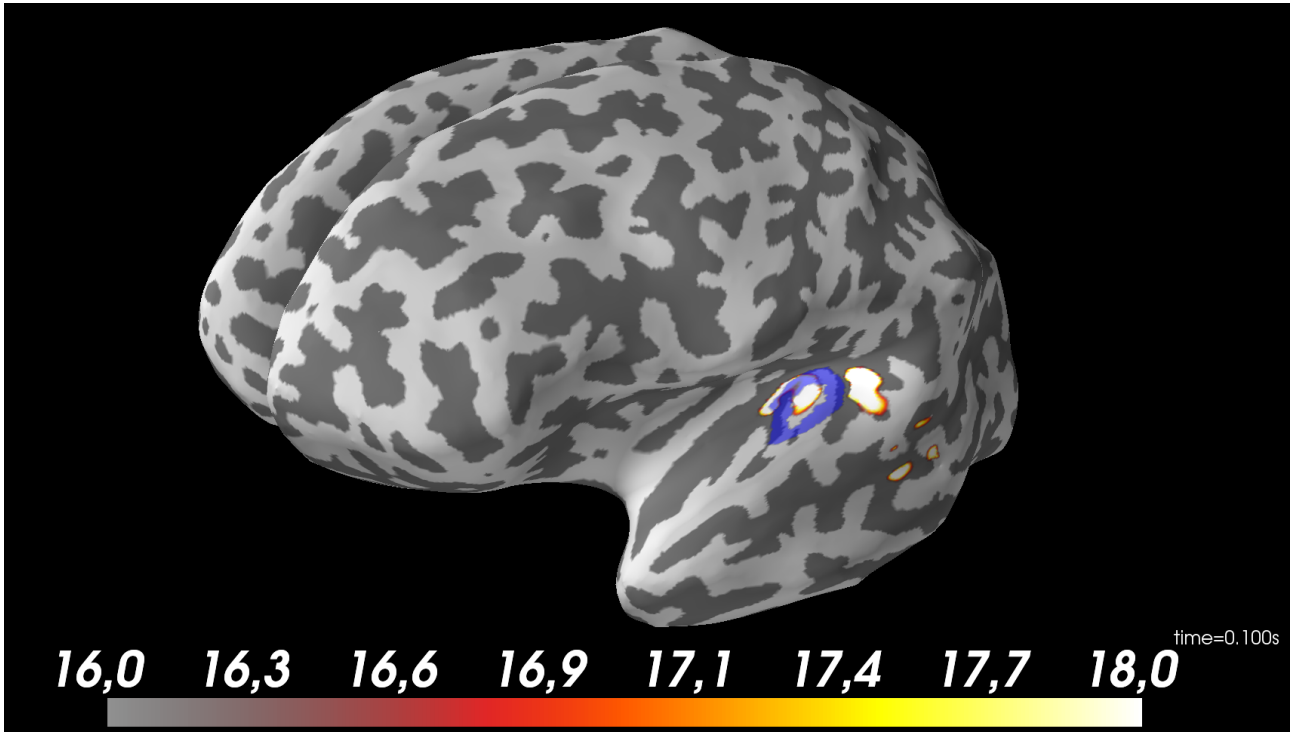


Karolinska Institutet

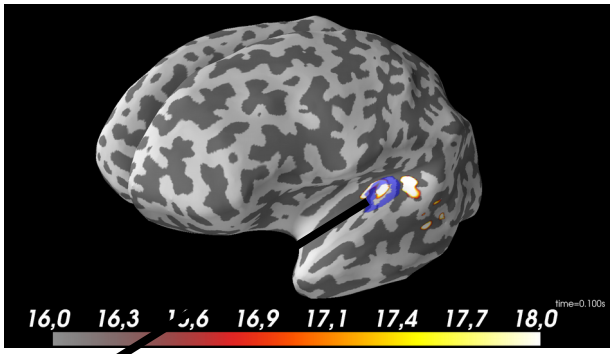
What is its origin?

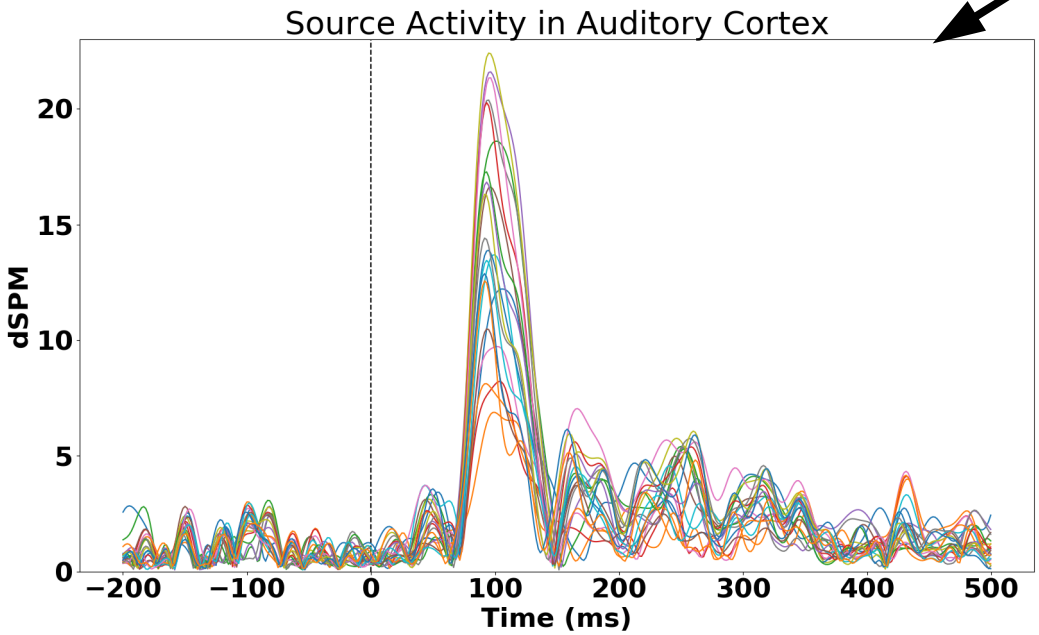


What is its origin?

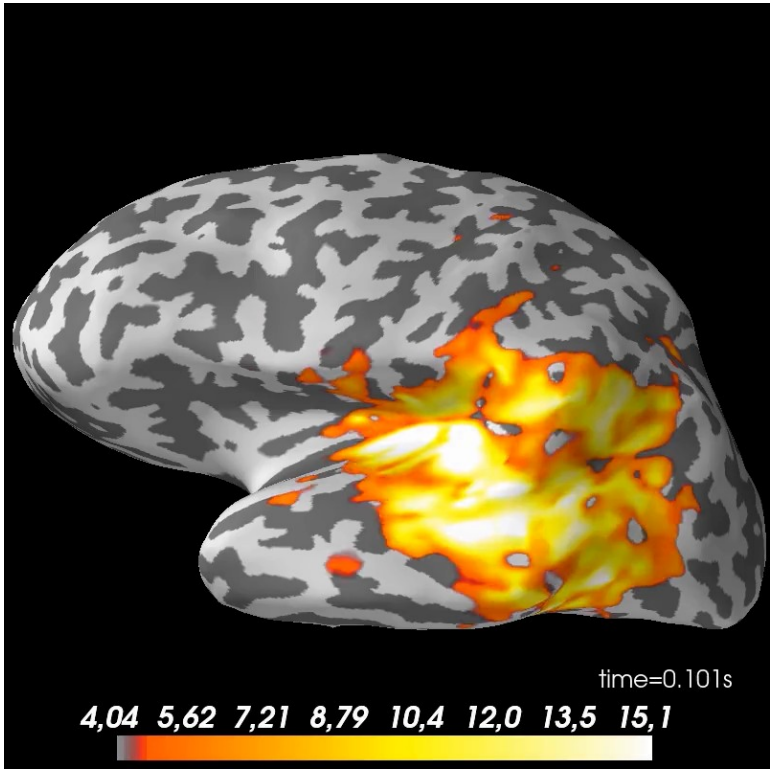


What is its origin?





What is its origin? (video)



Some history

Credit: Matti Hämäläinen

Some history

2016 – what existed?

- Single subject tutorials
- Recommendations paper

Karolinska Institutet

Single subject tutorials

Compare evoked responses for different conditions

In this example, an Epochs object for visual and auditory responses is created. Both conditions are then accessed by their respective names to create a sensor layout plot of the related evoked responses.

```
# Authors: Denis Engemann <denis.engemann@gmail.com>
#          Alexandre Gramfort <alexandre.gramfort@telecom-paristech.fr>

# License: BSD (3-clause)


import matplotlib.pyplot as plt
import mne

from mne.viz import plot_evoked_topo
from mne.datasets import sample

print(__doc__)

data_path = sample.data_path()
```

Set parameters

```
raw_fname = data_path + '/MEG/sample/sample_audvis_filt-0-40_raw.fif'
event_fname = data_path + '/MEG/sample/sample_audvis_filt-0-40_raw-eve.fif'
event_id = 1
tmin = -0.2
tmax = 0.5

#   Setup for reading the raw data
raw = mne.io.read_raw_fif(raw_fname)
events = mne.read_events(event_fname)

#   Set up pick list: MEG + STI 014 - bad channels (modify to your needs)
include = []  # or stim channels ['STI 014']
# bad channels in raw.info['bads'] will be automatically excluded

#   Set up amplitude-peak rejection values for MEG channels
reject = dict(grad=4000e-13, mag=4e-12)

# pick MEG channels
picks = mne.pick_types(raw.info, meg=True, eeg=False, stim=False, eog=True,
                       include=include, exclude='bads')
```

Download Python source code: plot_topo_compare_conditions.py

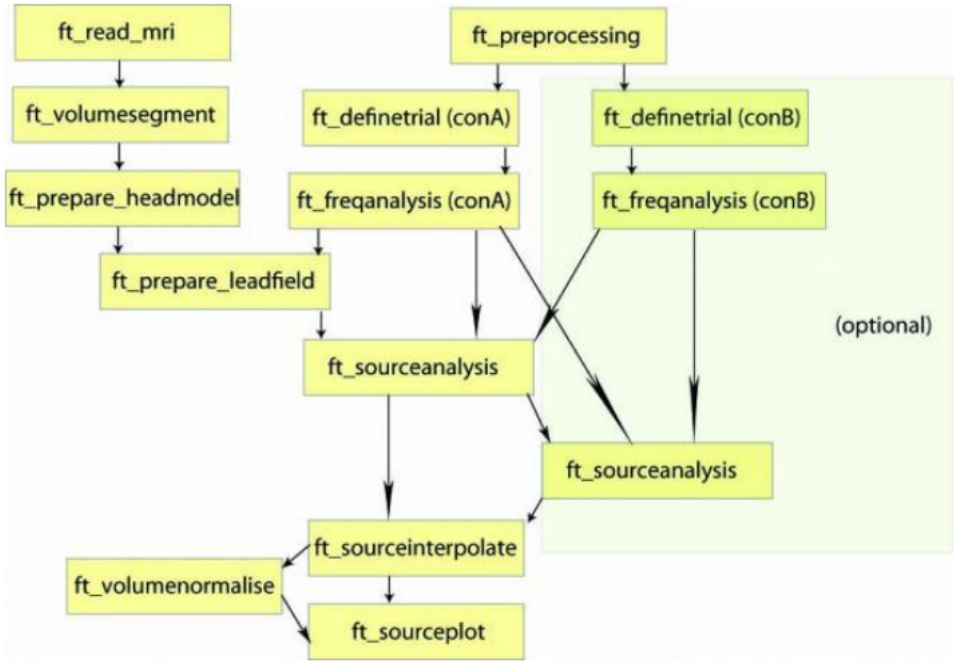
Download Jupyter notebook: plot_topo_compare_conditions.ipynb

MEG + EEG ANALYSIS & VISUALIZATION

Open-source Python software for exploring, visualizing, and analyzing human neurophysiological data: MEG, EEG, sEEG, ECoG, and more.

Karolinska Institutet

Single subject tutorials



FieldTrip

Karolinska Institutet

Recommendations paper

NeuroImage

Volume 65, 15 January 2013, Pages 349-363

Comments and Controversies

Good practice for conducting and reporting MEG research

Joachim Gross [a], Sylvain Baillet [b], Gareth R. Barnes [c], Richard N. Henson [d], Arjan Hillebrand [e], Ole Jensen [f], Karim Jerbi [g], Vladimir Litvak [c], Burkhard Maess [h], Robert Oostenveld [f], Lauri Parkkonen [i, j], Jason R. Taylor [d], Virginie van Wassenhove [k, l, m], Michael Wibral [n], Jan-Mathijs Schoffelen [f, o]

Show more

https://doi.org/10.1016/j.neuroimage.2012.10.001

Some history

2016 – what was missing?

- Group level tutorials
- Open data on which to apply them

How to analyse MEG?

How to analyse MEG?

Karolinska Institutet

Research Topic

From raw MEG/EEG to publication: how to perform MEG/EEG group analysis with free academic software.

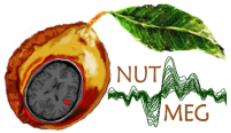
Submission closed.

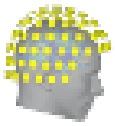
Comment 0

4 28 1 2 38

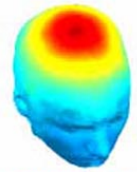
Overview | 19 Articles | 72 Authors | Impact | Comments

VIEWS 42,827





Brainstorm



EEGLAB

FieldTrip

MNE

MEG + EEG ANALYSIS & VISUALIZATION

ource Python software for exploring, visualizing, and analyzing
neurophysiological data: MEG, EEG, sEEG, ECoG, and more.

How to analyse MEG?

Karolinska Institutet

SCIENTIFIC DATA

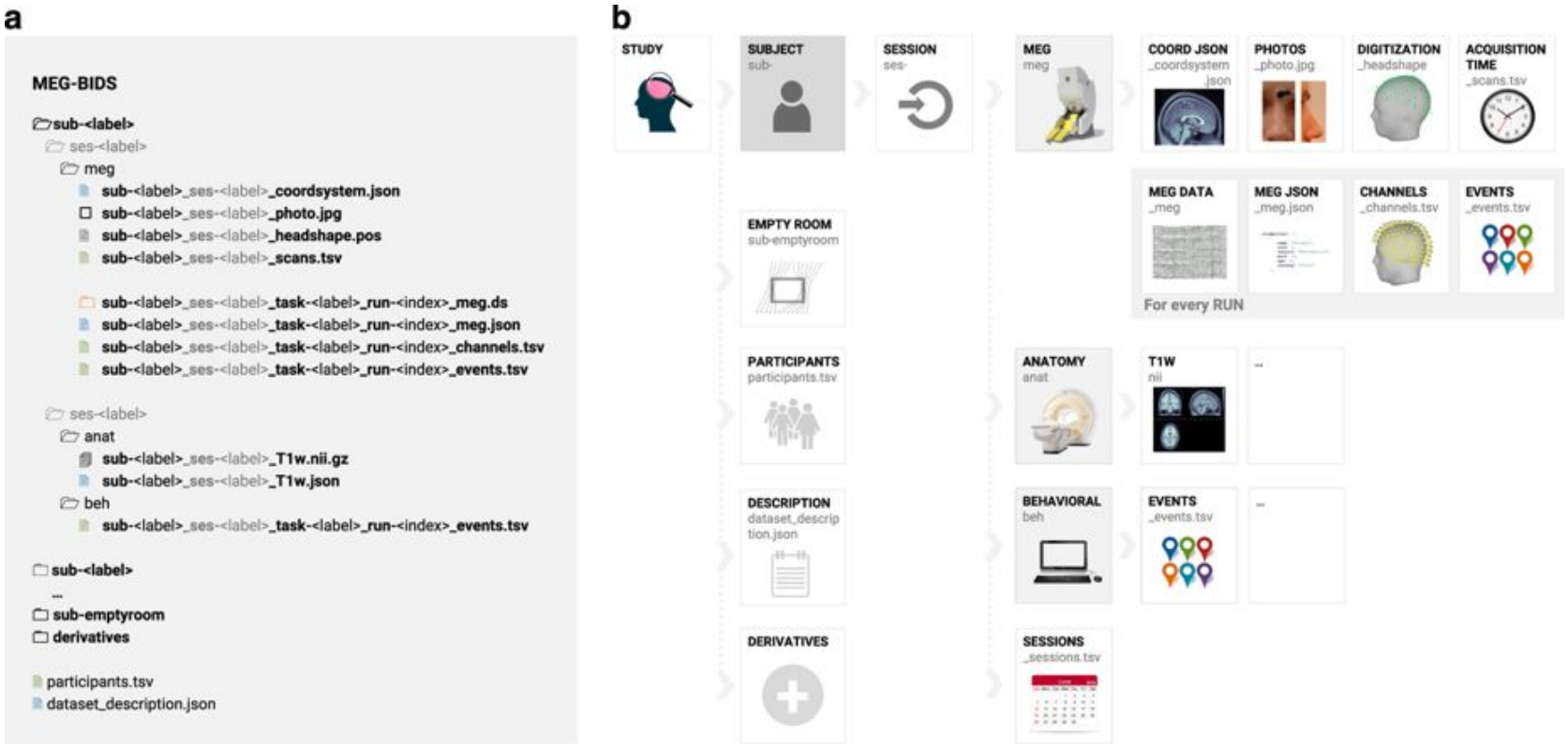
Comment | OPEN | Published: 19 June 2018

MEG-BIDS, the brain imaging data structure extended to magnetoencephalography

Guiomar Niso, Krzysztof J. Gorgolewski, Elizabeth Bock, Teon L. Brooks, Guillaume Flandin, Alexandre Gramfort, Richard N. Henson, Mainak Jas, Vladimir Litvak, Jeremy T. Moreau, Robert Oostenveld, Jan-Mathijs Schoffelen, Francois Tadel, Joseph Wexler & Sylvain Baillet ✉

Scientific Data **5**, Article number: 180110 (2018) | Download Citation ⬇

MEG-BIDS

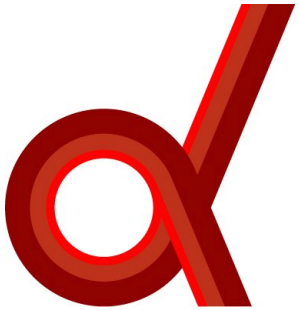


Niso et al. 2018, *Scientific data*

Karolinska Institutet

DATA REPOSITORIES

zenodo



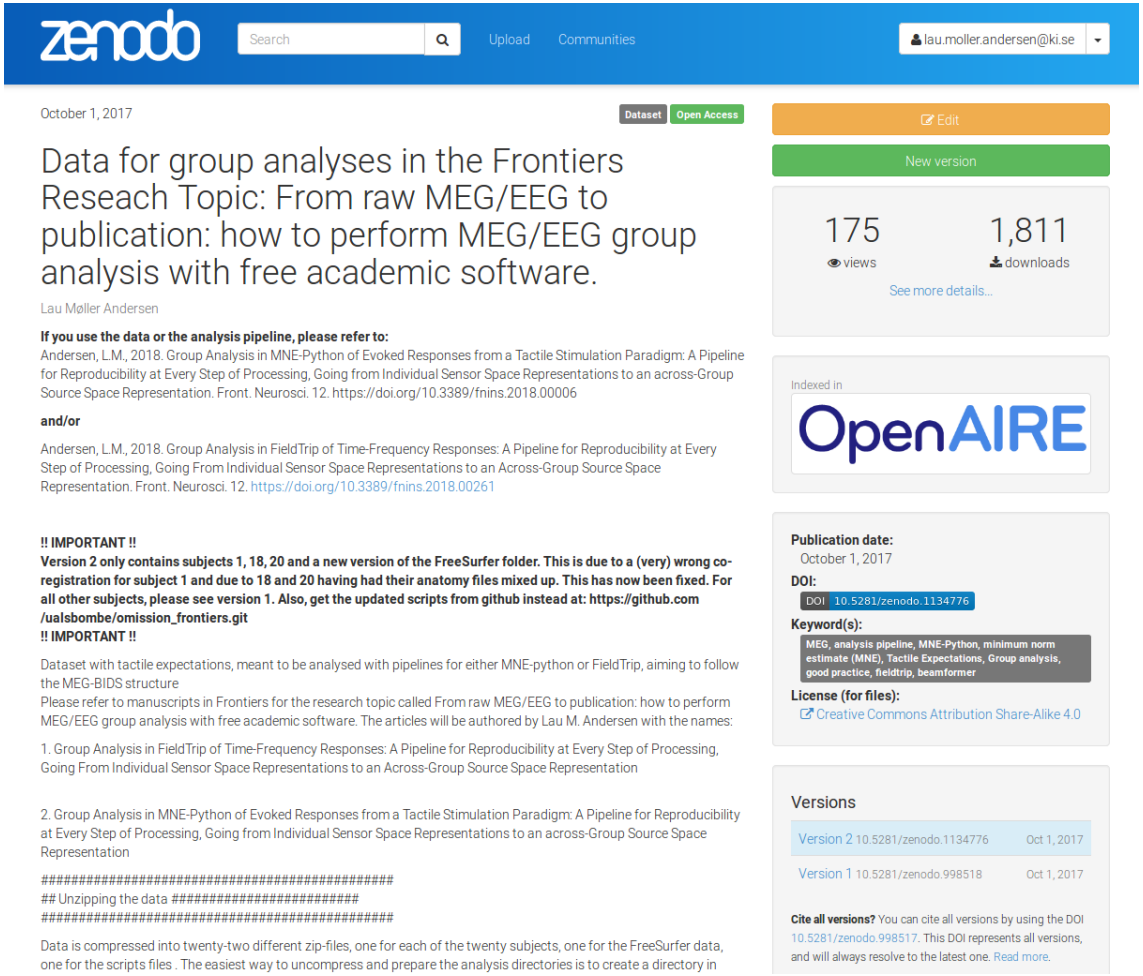
DONDERS

OpenNEURO BETA

A free and open platform for analyzing and sharing neuroimaging data

Karolinska Institutet

DATA



zenodo

Search

Upload Communities

lau.moller.andersen@ki.se

October 1, 2017

Dataset Open Access

Data for group analyses in the Frontiers Reseach Topic: From raw MEG/EEG to publication: how to perform MEG/EEG group analysis with free academic software.

Lau Møller Andersen

If you use the data or the analysis pipeline, please refer to:
Andersen, L.M., 2018. Group Analysis in MNE-Python of Evoked Responses from a Tactile Stimulation Paradigm: A Pipeline for Reproducibility at Every Step of Processing, Going from Individual Sensor Space Representations to an across-Group Source Space Representation. Front. Neurosci. 12. https://doi.org/10.3389/fnins.2018.00006

and/or

Andersen, L.M., 2018. Group Analysis in FieldTrip of Time-Frequency Responses: A Pipeline for Reproducibility at Every Step of Processing, Going From Individual Sensor Space Representations to an Across-Group Source Space Representation. Front. Neurosci. 12. https://doi.org/10.3389/fnins.2018.00261

!! IMPORTANT !!
Version 2 only contains subjects 1, 18, 20 and a new version of the FreeSurfer folder. This is due to a (very) wrong co-registration for subject 1 and due to 18 and 20 having had their anatomy files mixed up. This has now been fixed. For all other subjects, please see version 1. Also, get the updated scripts from github instead at: https://github.com /ualsbombe/omission_frontiers.git
!! IMPORTANT !!

Dataset with tactile expectations, meant to be analysed with pipelines for either MNE-python or FieldTrip, aiming to follow the MEG-BIDS structure
Please refer to manuscripts in Frontiers for the research topic called From raw MEG/EEG to publication: how to perform MEG/EEG group analysis with free academic software. The articles will be authored by Lau M. Andersen with the names:

1. Group Analysis in FieldTrip of Time-Frequency Responses: A Pipeline for Reproducibility at Every Step of Processing, Going From Individual Sensor Space Representations to an Across-Group Source Space Representation

2. Group Analysis in MNE-Python of Evoked Responses from a Tactile Stimulation Paradigm: A Pipeline for Reproducibility at Every Step of Processing, Going from Individual Sensor Space Representations to an across-Group Source Space Representation

```
##############################################
## Unzipping the data ########################
##############################################
```

Data is compressed into twenty-two different zip-files, one for each of the twenty subjects, one for the FreeSurfer data, one for the scripts files . The easiest way to uncompress and prepare the analysis directories is to create a directory in

Edit

New version

175 views

1,811 downloads

See more details...

Indexed in OpenAIRE

Publication date:
October 1, 2017

DOI:
DOI 10.5281/zenodo.1134776

Keyword(s):
MEG, analysis pipeline, MNE-Python, minimum norm estimate (MNE), Tactile Expectations, Group analysis, good practice, fieldtrip, beamformer

License (for files):
Creative Commons Attribution Share-Alike 4.0

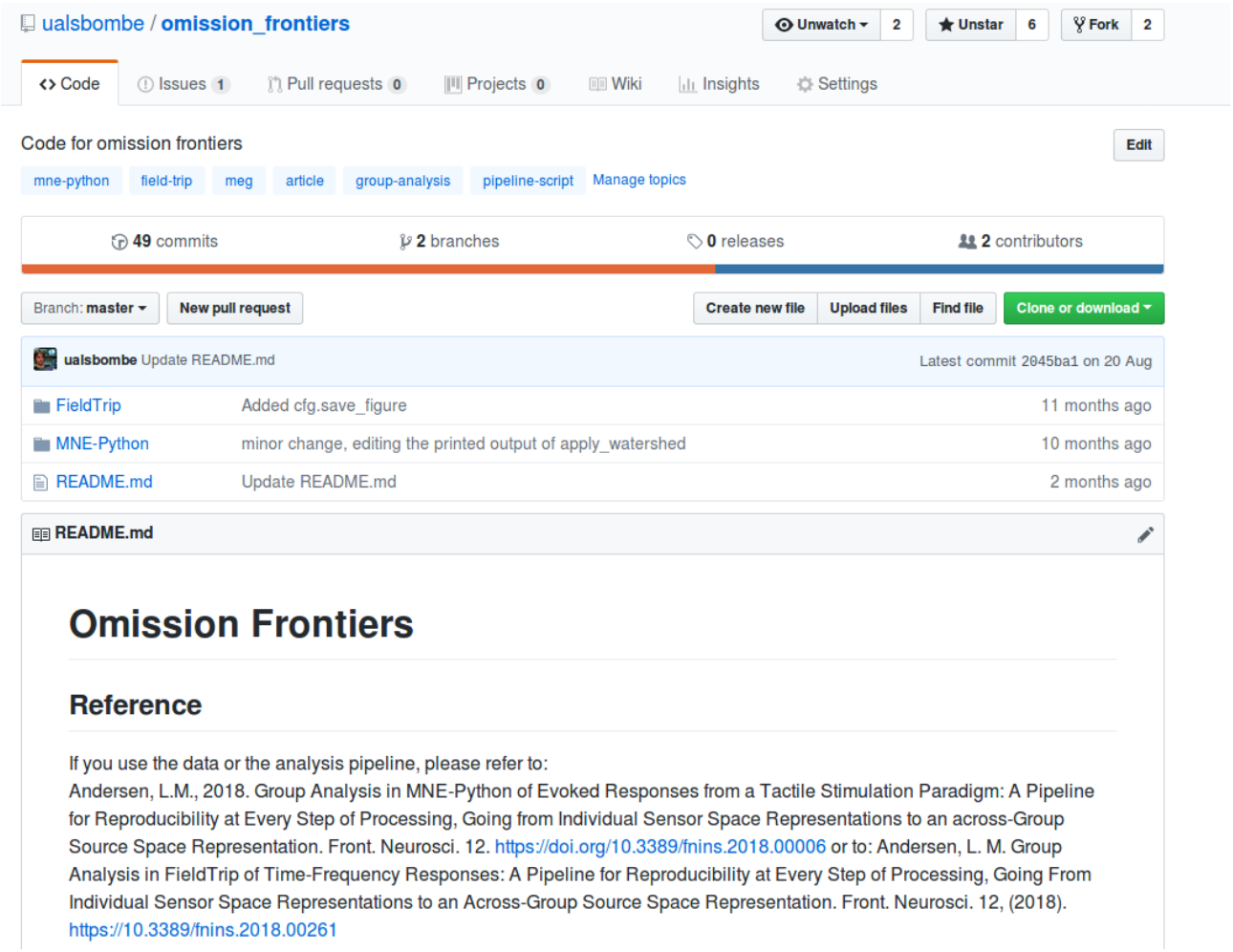
Versions

Version 2 10.5281/zenodo.1134776 Oct 1, 2017

Version 1 10.5281/zenodo.998518 Oct 1, 2017

Cite all versions? You can cite all versions by using the DOI 10.5281/zenodo.998517. This DOI represents all versions, and will always resolve to the latest one. Read more.

CODE



ualsbombe / **omission_frontiers**

Unwatch 2 | Unstar 6 | Fork 2

Code | Issues 1 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

Code for omission frontiers

Edit

mne-python | field-trip | meg | article | group-analysis | pipeline-script | Manage topics

49 commits | 2 branches | 0 releases | 2 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

ualsbombe Update README.md — Latest commit 2045ba1 on 20 Aug

FieldTrip	Added cfg.save_figure	11 months ago
MNE-Python	minor change, editing the printed output of apply_watershed	10 months ago
README.md	Update README.md	2 months ago

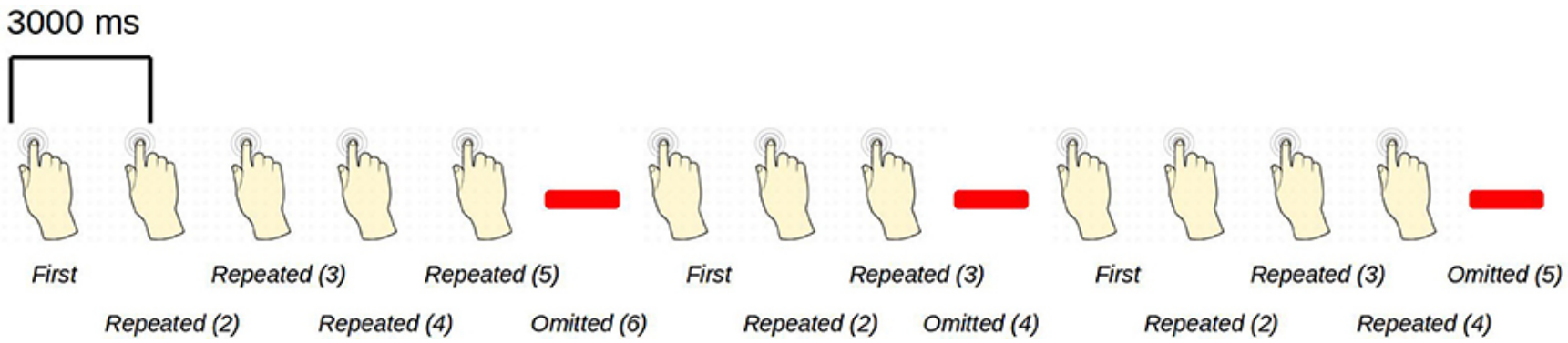
README.md

Omission Frontiers

Reference

If you use the data or the analysis pipeline, please refer to:
Andersen, L.M., 2018. Group Analysis in MNE-Python of Evoked Responses from a Tactile Stimulation Paradigm: A Pipeline for Reproducibility at Every Step of Processing, Going from Individual Sensor Space Representations to an across-Group Source Space Representation. Front. Neurosci. 12. https://doi.org/10.3389/fnins.2018.00006 or to: Andersen, L. M. Group Analysis in FieldTrip of Time-Frequency Responses: A Pipeline for Reproducibility at Every Step of Processing, Going From Individual Sensor Space Representations to an Across-Group Source Space Representation. Front. Neurosci. 12, (2018). https://10.3389/fnins.2018.00261

PARADIGM



Andersen 2018, Frontiers in Neuroscience

QUESTION

Do *Stimulations* give rise to more activity than *Non-Stimulations* in the primary somatosensory cortex?

How to analyse MEG?

ELSEVIER

NeuroImage

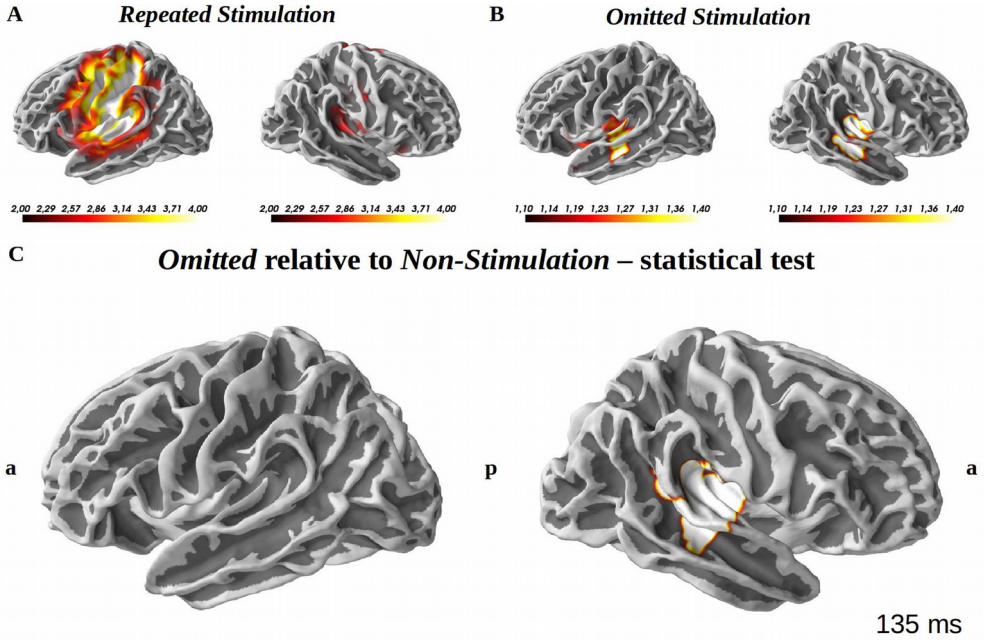
Volume 184, 1 January 2019, Pages 78-89

Somatosensory responses to nothing: An MEG study of expectations during omission of tactile stimulations

Lau M. Andersen, Daniel Lundqvist

Show more

https://doi.org/10.1016/j.neuroimage.2018.09.014



Karolinska Institutet

SUBJECTS

```
subjects = [
                'sub-01',
                'sub-02',
                'sub-03',
                'sub-04',
                'sub-05',
                'sub-06',
                'sub-07',
                'sub-08',
                'sub-09',
                'sub-10',
                'sub-11',
                'sub-12',
                'sub-13',
                'sub-14',
                'sub-15',
                'sub-16',
                'sub-17',
                'sub-18',
                'sub-19',
                'sub-20'
            ]
```

4 GUIDING PRINCIPLES FOR A PIPELINE

1. Should divide output files into folders belonging to the respective subjects and recordings

2. Should apply operations across a group of subjects

3. Should make it possible to (re)start the analysis at any intermediate point by saving output for each intermediate point

4. Should allow for plotting the results in a way that allows for changing the figures in a principled, but flexible manner

TWO PIPELINES

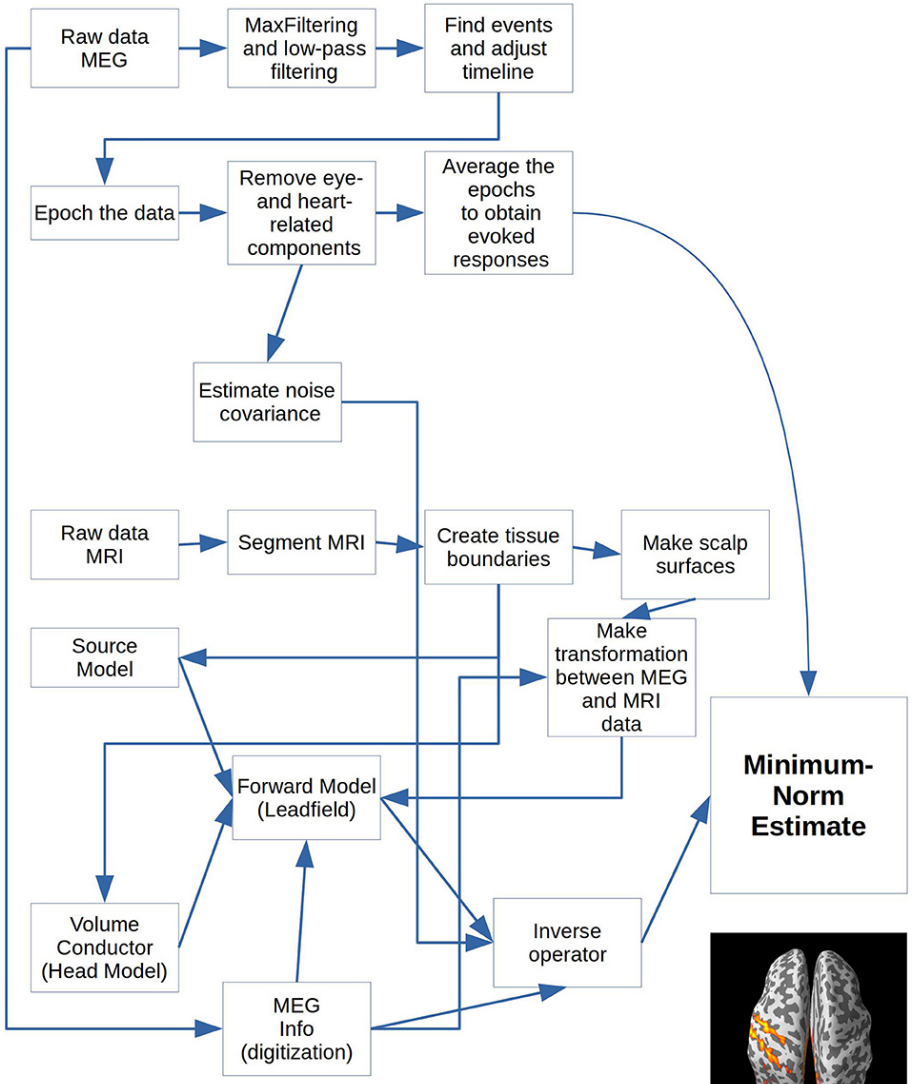
MNE

MEG + EEG ANALYSIS & VISUALIZATION

Open-source Python software for exploring, visualizing, and analyzing human neurophysiological data: MEG, EEG, sEEG, ECoG, and more.

Karolinska Institutet

MNE-PYTHON PIPELINE



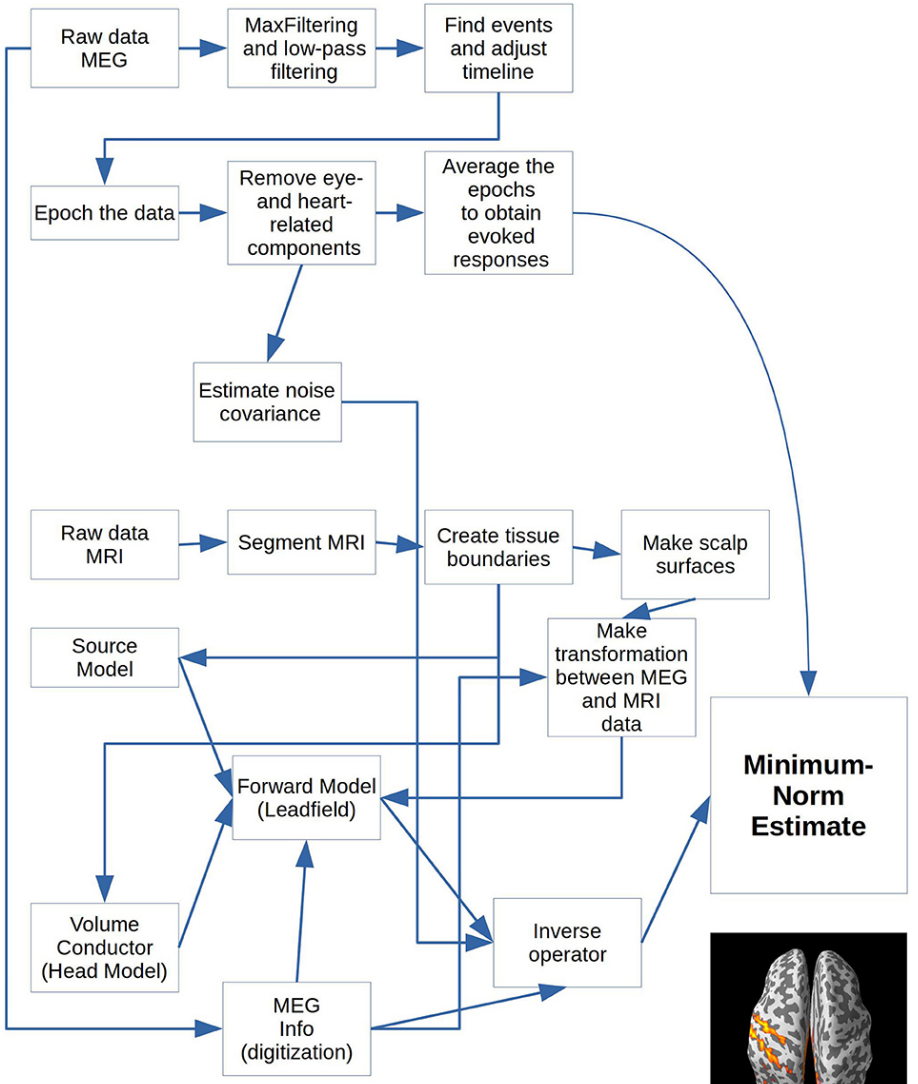
Andersen 2018, Frontiers in Neuroscience

Karolinska Institutet

OPERATIONS

```
## sensor space operations
filter_raw=0,
find_events=0,
epoch_raw=0,
run_ica=0,
apply_ica=0,
get_evokeds=0,
```

```
## source space operations
import_mri=0,
segment_mri=0, # long process (>6 h)
apply_watershed=0,
make_source_space=0,
make_dense_scalp_surfaces=0,
make_bem_solutions=0,
create_forward_solution=0,
estimate_noise_covariance=0,
create_inverse_operator=0,
source_estimate=0,
morph_to_fsaverage=0,
```



Andersen 2018, Frontiers in Neuroscience

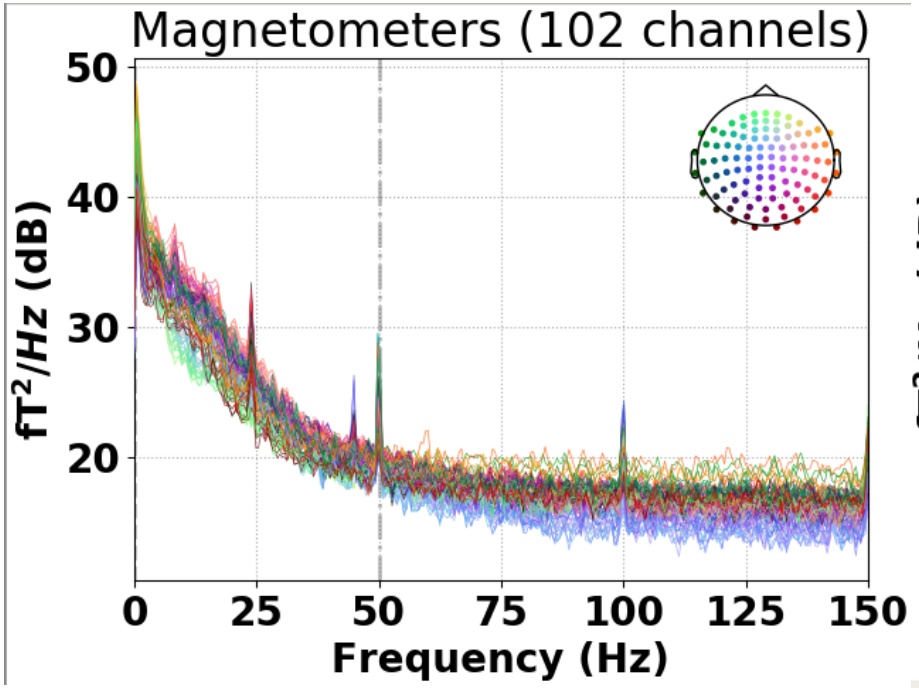
Karolinska Institutet

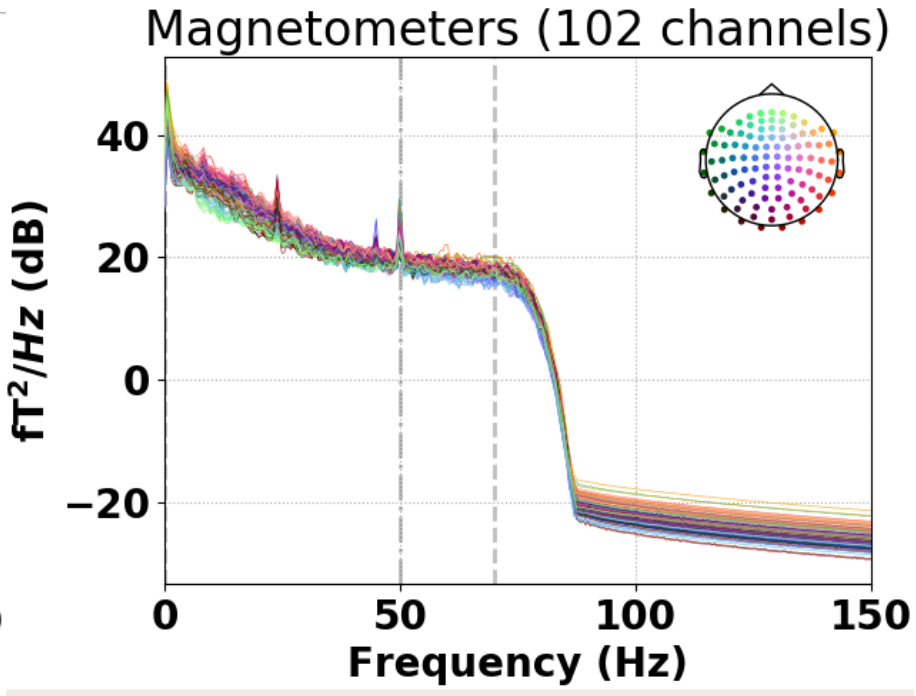
OPERATIONS

```
## sensor space operations
filter_raw=0,
find_events=0,
epoch_raw=1,
run_ica=0,
apply_ica=0,
get_evokeds=0,
```

```
def epoch_raw(name, save_dir, lowpass, event_id, tmin, tmax,
              baseline, reject, bad_channels, decim, overwrite):

    epochs_name = name + filter_string(lowpass) + '-epo.fif'
    epochs_path = join(save_dir, epochs_name)
    if overwrite or not isfile(epochs_path):

        events = io.read_events(name, save_dir)
        raw = io.read_filtered(name, save_dir, lowpass,)
        raw.info['bads'] = bad_channels
        picks = mne.pick_types(raw.info, meg=True, eog=True, ecg=True,
                               exclude='bads')

        epochs = mne.Epochs(raw, events, event_id, tmin, tmax, baseline,
                            reject=reject, preload=True, picks=picks,
                            decim=decim)

        epochs.save(epochs_path)

    else:
        print('epochs file: '+ epochs_path + ' already exists')
```

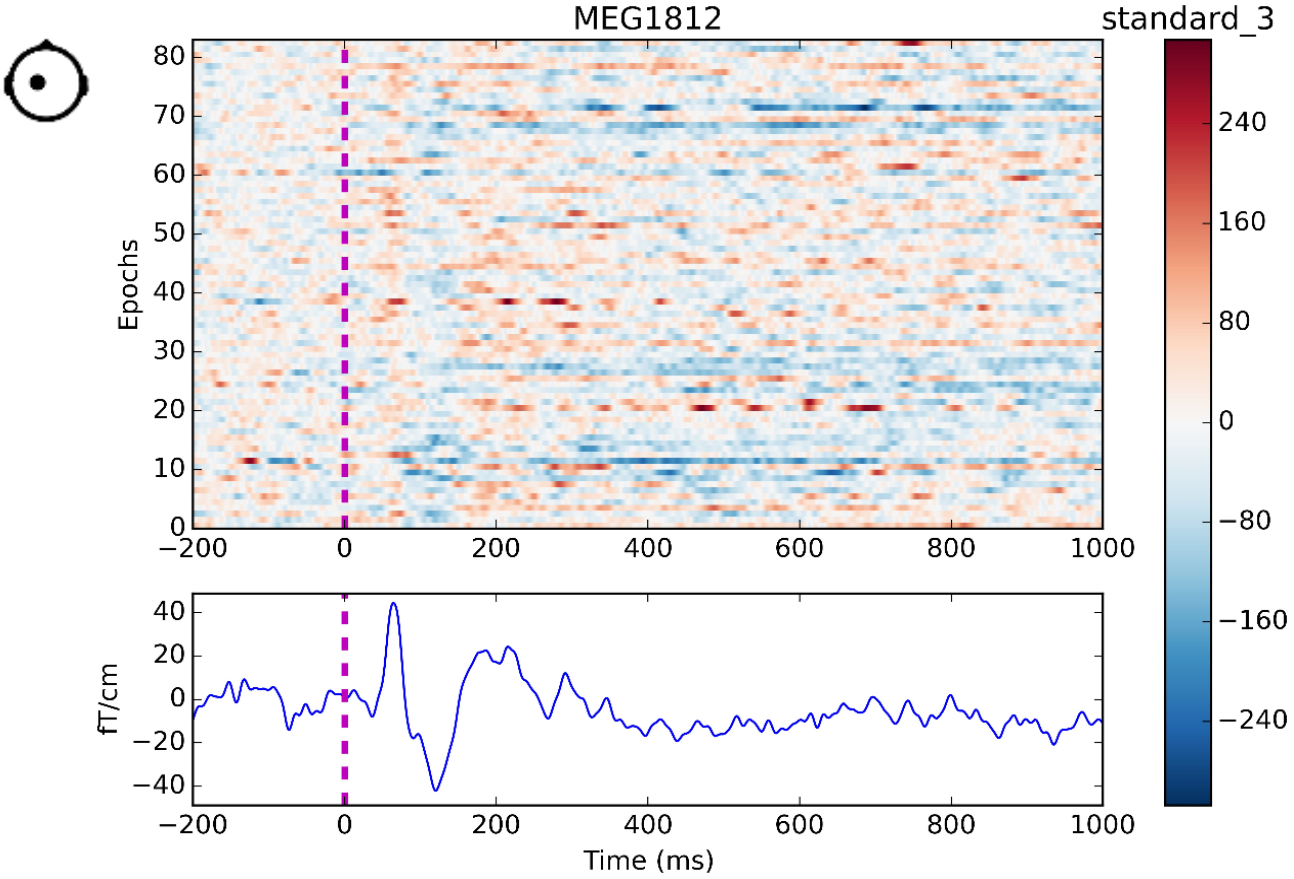
OPERATIONS – FILTERING



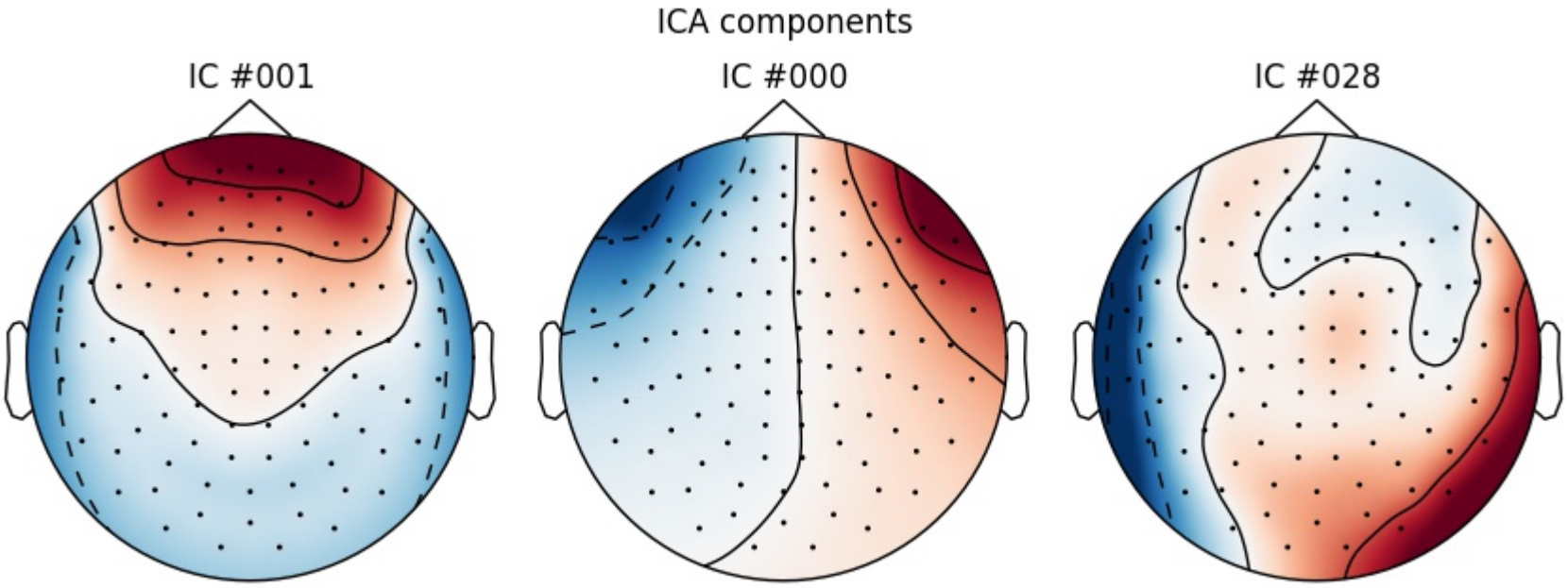


Karolinska Institutet

OPERATIONS – EPOCHING

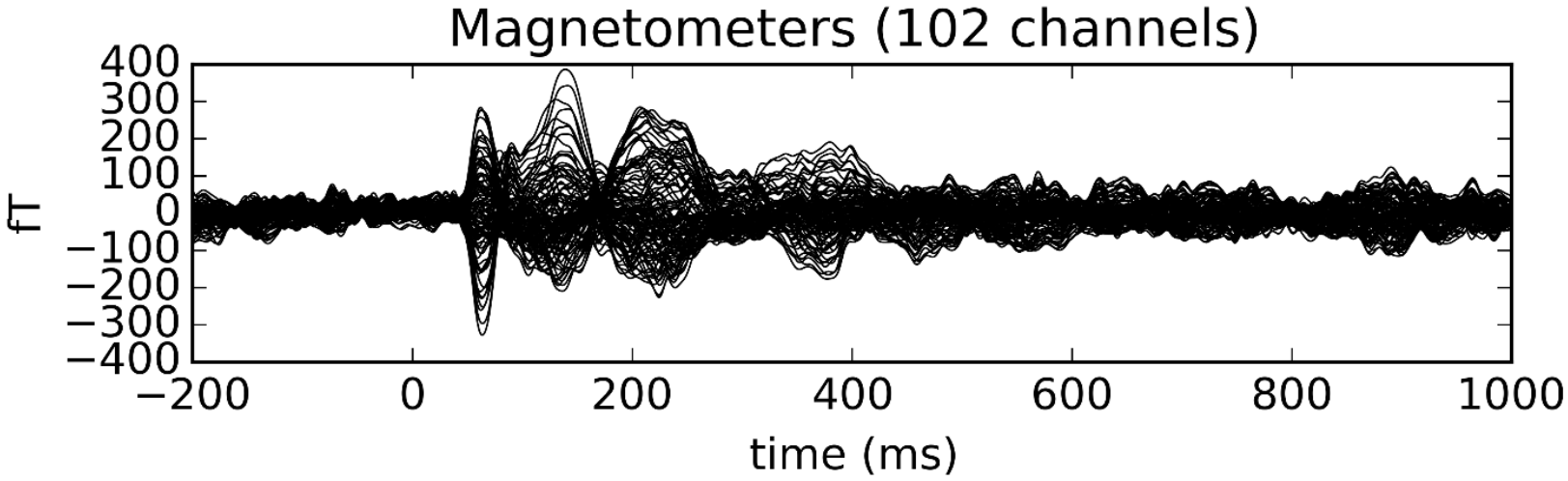


OPERATIONS – ICA



PHYSIOLOGICAL NOISE, *EYE BLINKS* AND *HEART BEATS*

OPERATIONS – EVOKEDS



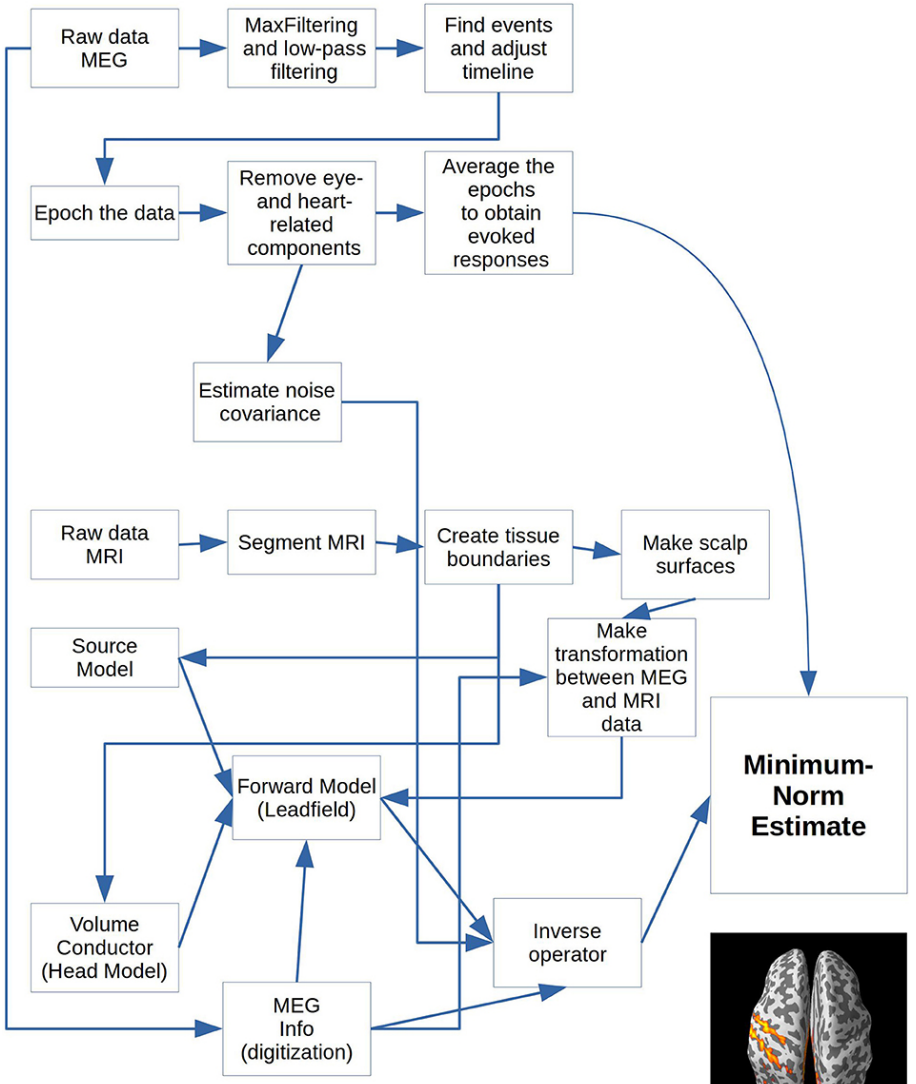
THESE ARE THE ONES WE WANT TO SOURCE RECONSTRUCT

Karolinska Institutet

OPERATIONS

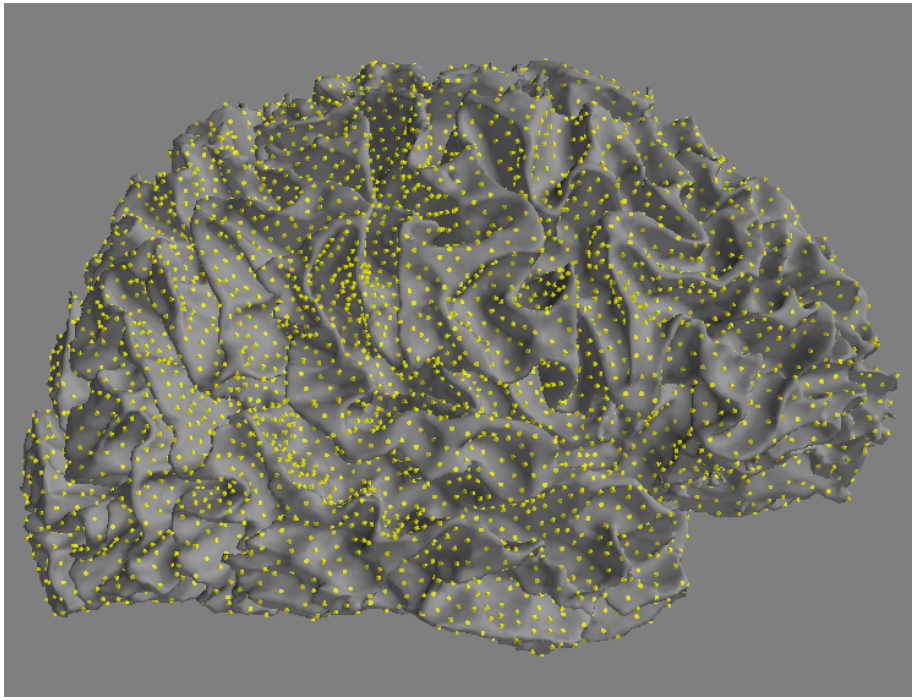
```
## sensor space operations
filter_raw=0,
find_events=0,
epoch_raw=0,
run_ica=0,
apply_ica=0,
get_evokeds=0,
```

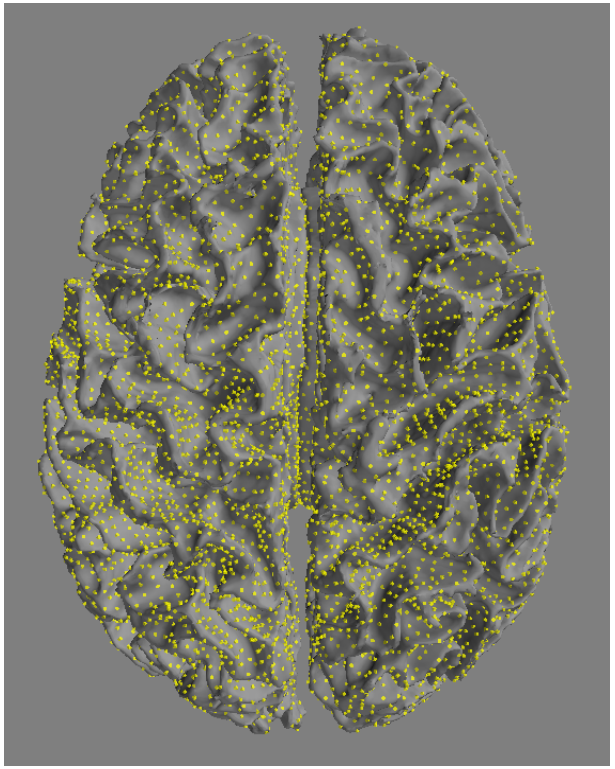
```
## source space operations
import_mri=0,
segment_mri=0, # long process (>6 h)
apply_watershed=0,
make_source_space=0,
make_dense_scalp_surfaces=0,
make_bem_solutions=0,
create_forward_solution=0,
estimate_noise_covariance=0,
create_inverse_operator=0,
source_estimate=0,
morph_to_fsaverage=0,
```



Andersen 2018, Frontiers in Neuroscience

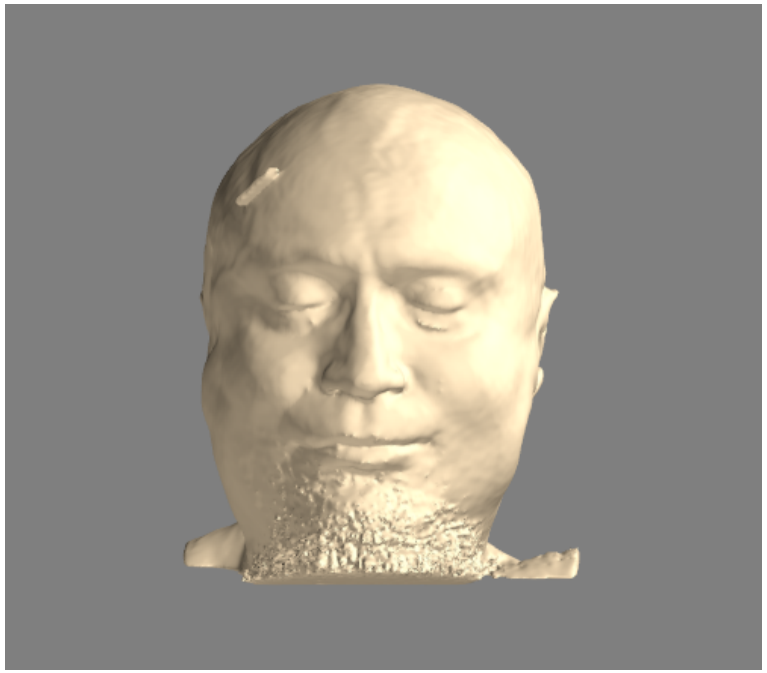
OPERATIONS – MAKE SOURCE SPACE





THE POSITIONS OF THE SOURCES

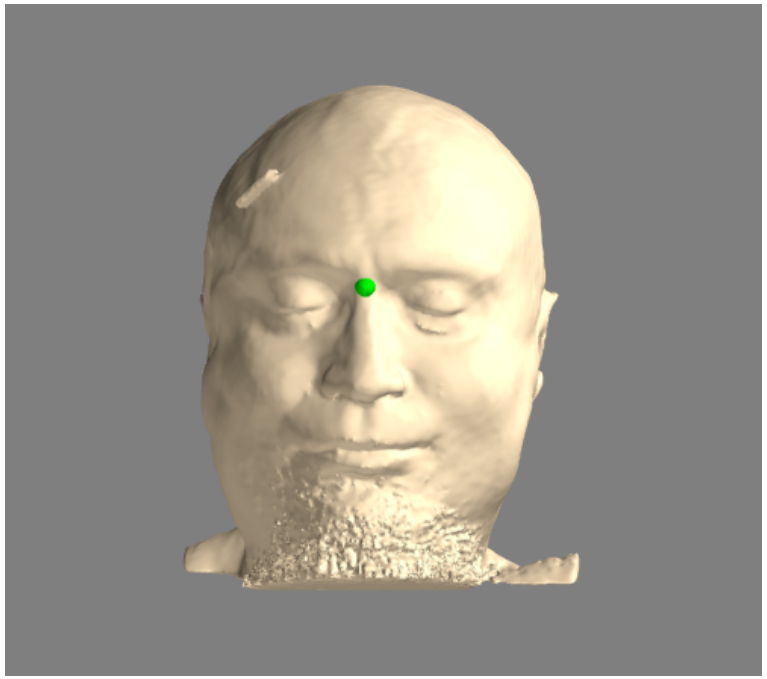
OPERATIONS – CO-REGISTER MEG AND MRI



CONSTRUCT A HEAD MODEL BASED ON THE MRI

OPERATIONS – CO-REGISTER MEG AND MRI

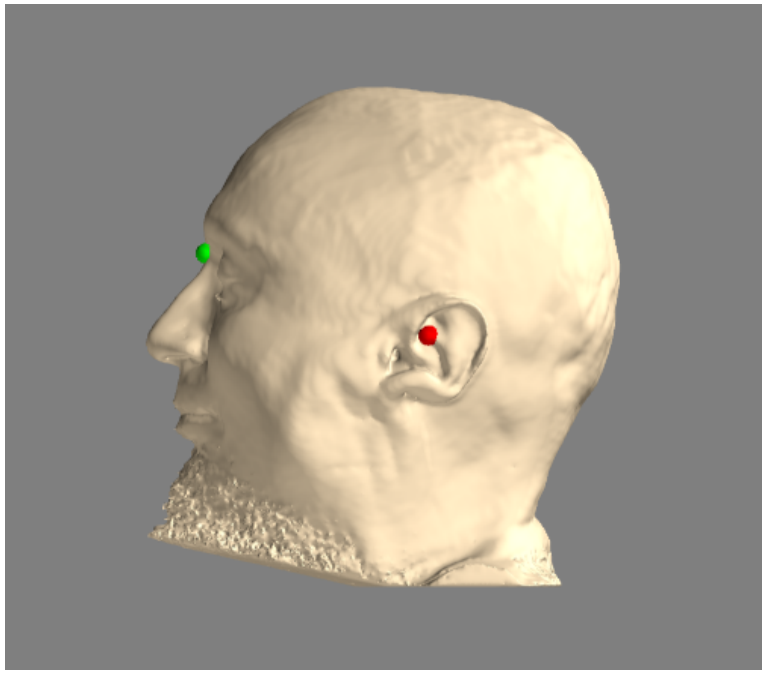
1) NASION



PLOT IN FIDUCIAL POINTS

OPERATIONS – CO-REGISTER MEG AND MRI

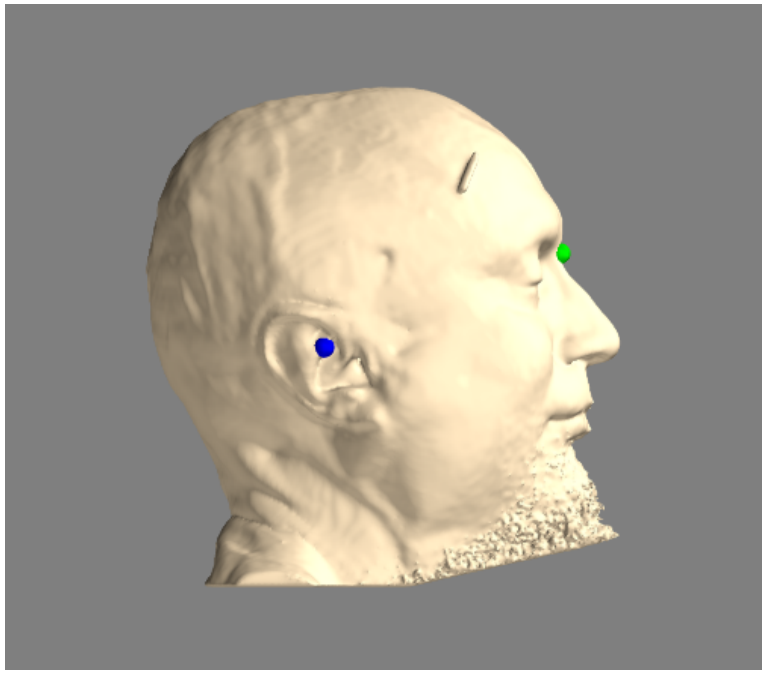
1) NASION
2) LPA



PLOT IN FIDUCIAL POINTS

OPERATIONS – CO-REGISTER MEG AND MRI

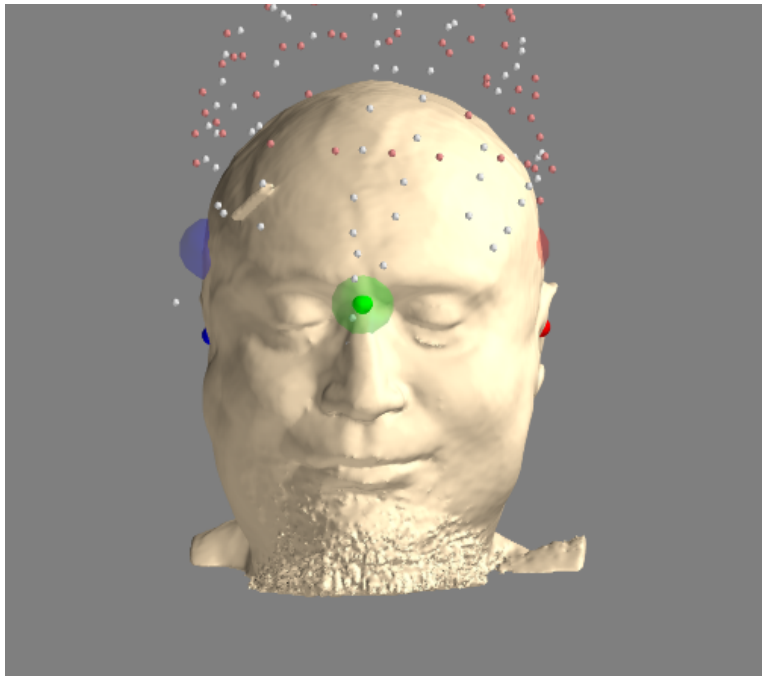
1) NASION
2) LPA
3) RPA



PLOT IN FIDUCIAL POINTS

OPERATIONS – CO-REGISTER MEG AND MRI

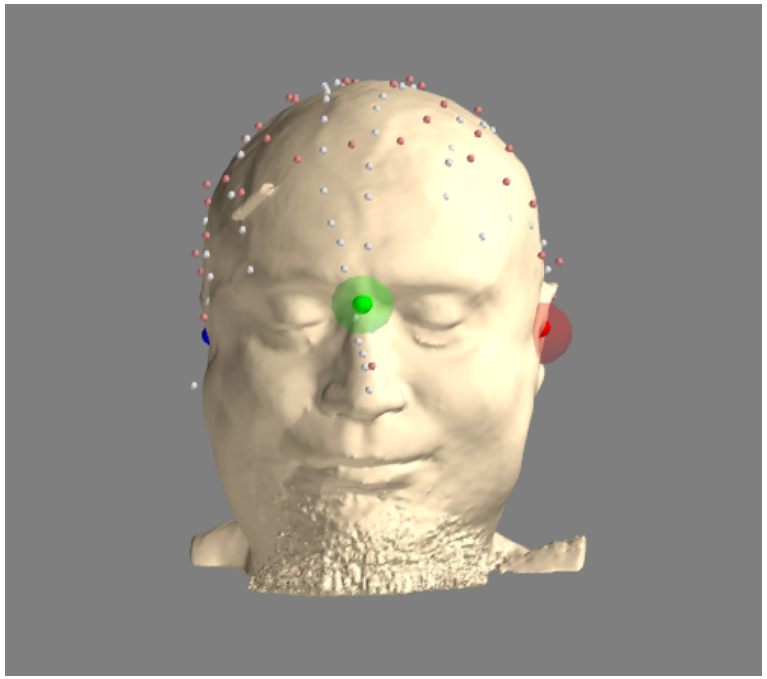
1) NASION
2) LPA
3) RPA



PLOT IN EXTRA HEAD POINTS

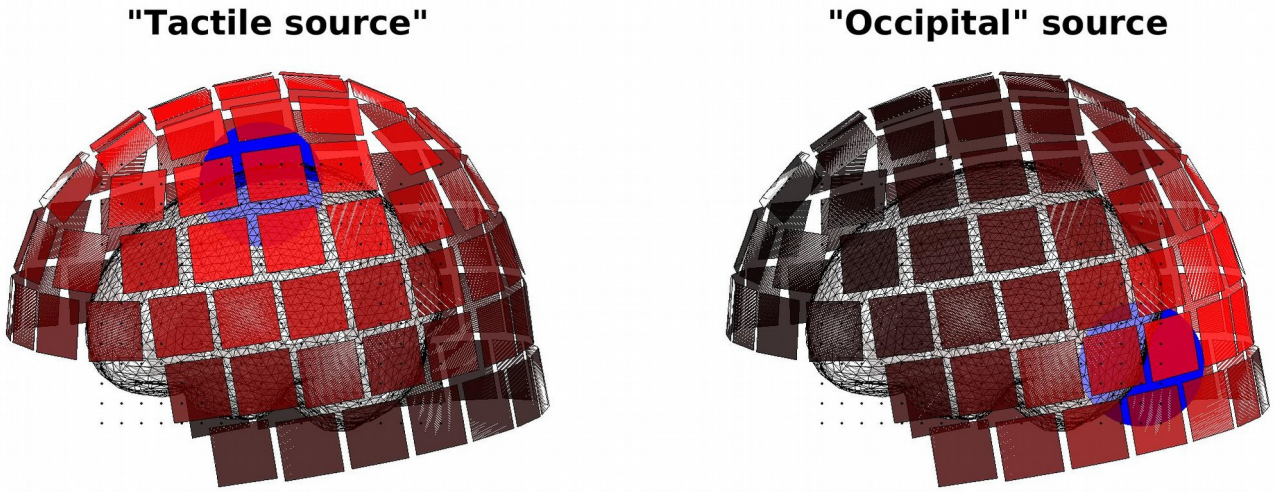
OPERATIONS – CO-REGISTER MEG AND MRI

1) NASION
2) LPA
3) RPA



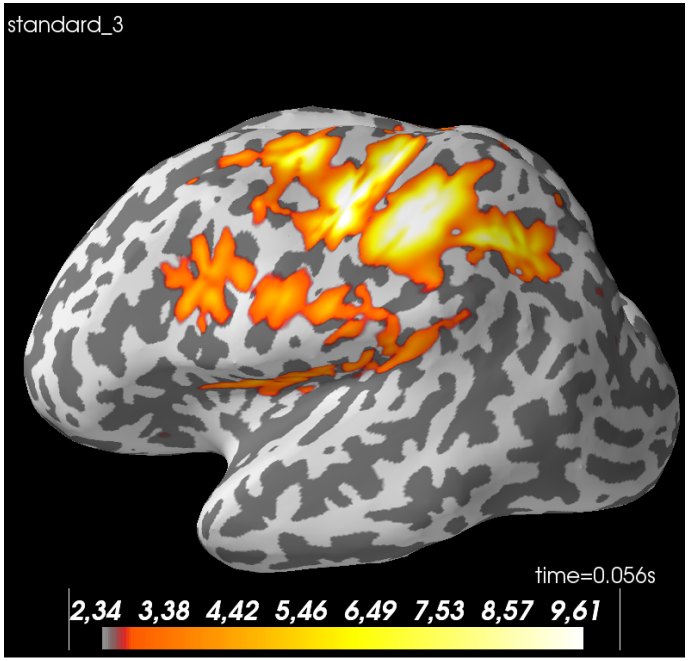
MINIMIZE DISTANCE BETWEEN HEAD POINTS AND HEAD MODEL

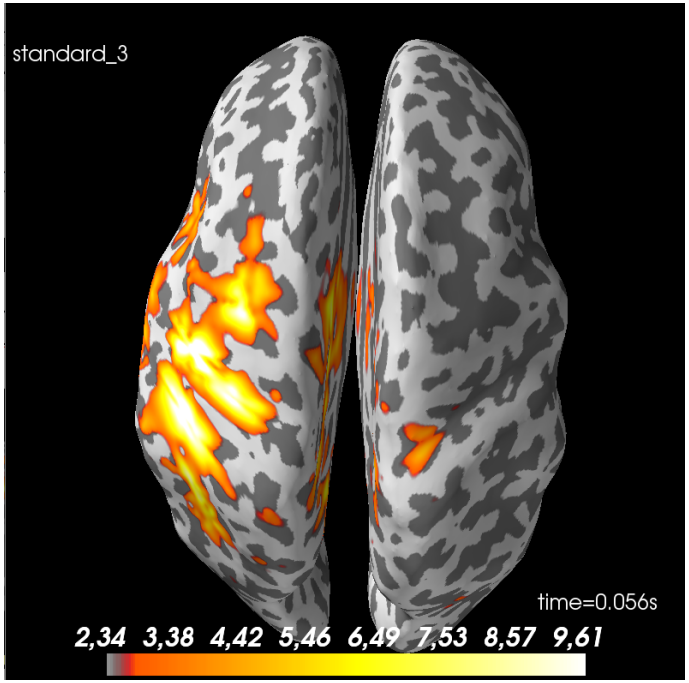
OPERATIONS – FORWARD MODEL



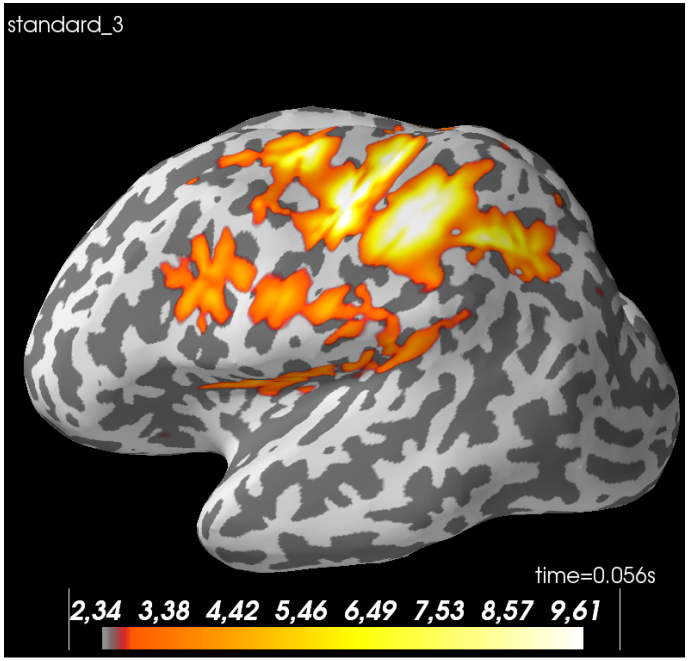
THE LINK BETWEEN SOURCES AND CHANNELS

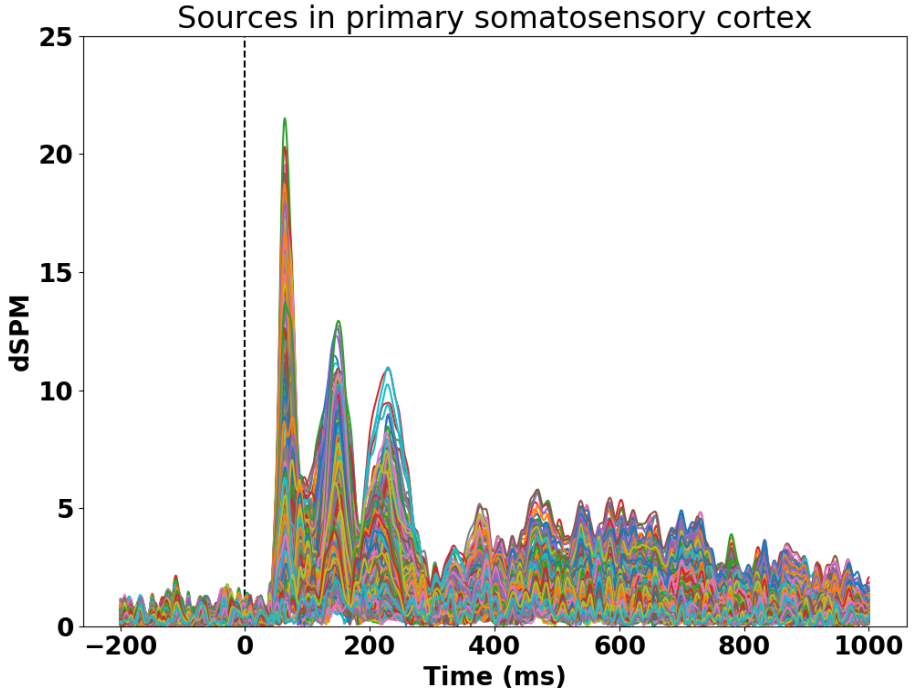
OPERATIONS – SOURCE RECONSTRUCTION



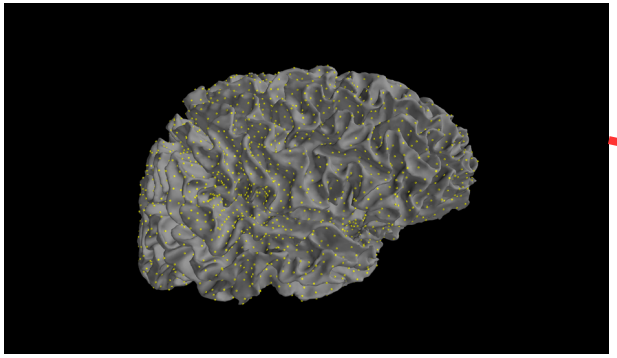


OPERATIONS – SOURCE RECONSTRUCTION

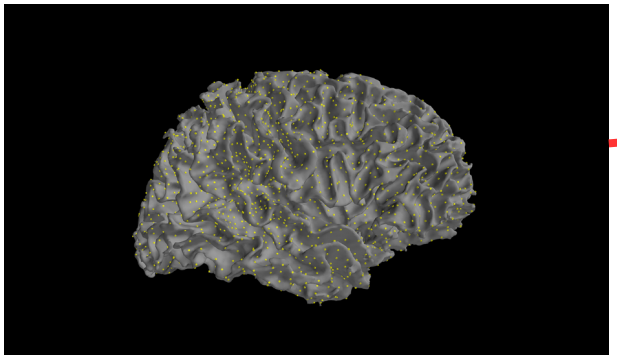




OPERATIONS – MORPH TO TEMPLATE



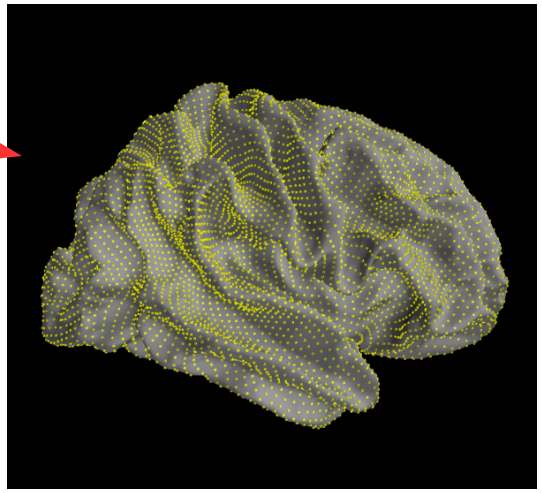
n_1



n_2

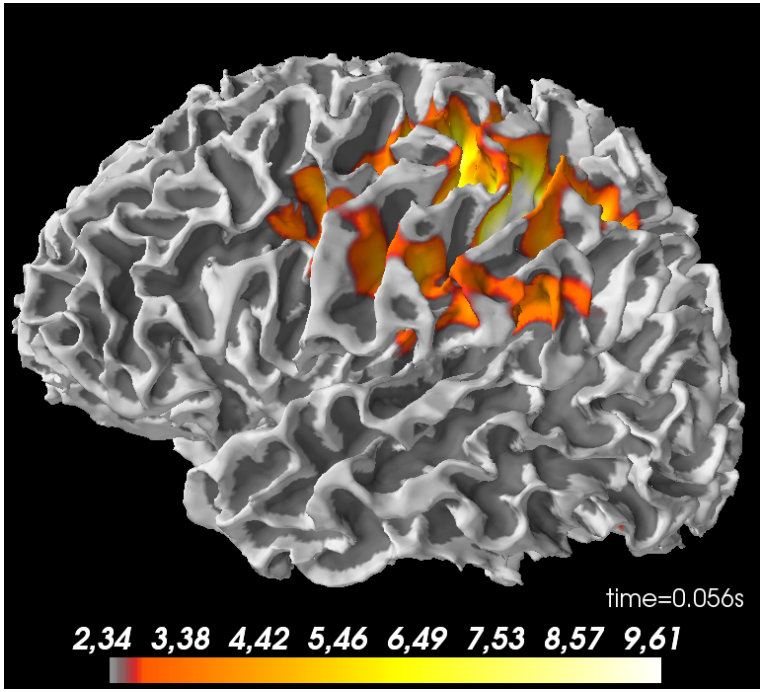
...

n_{20}

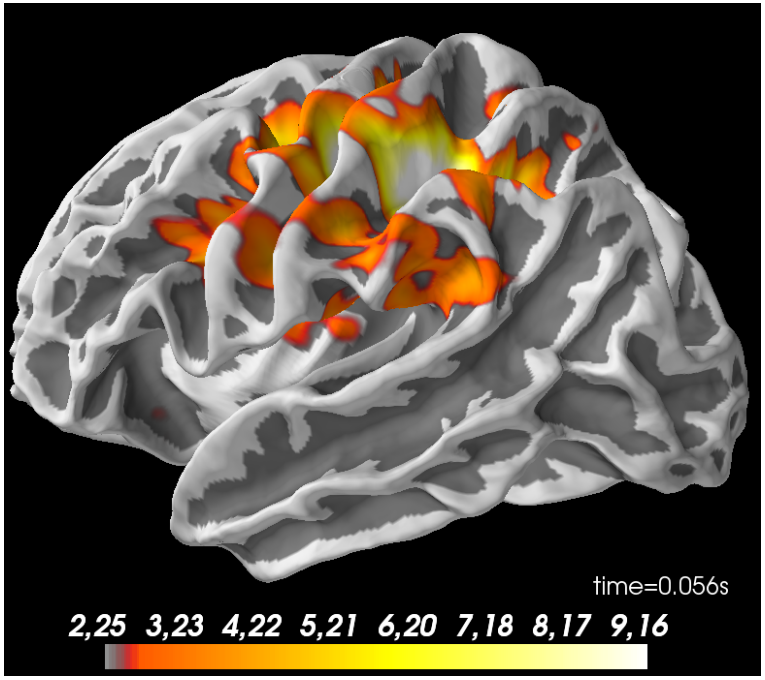


fsaverage template from the FreeSurfer suite, but many other anatomies exist

OPERATIONS – MORPH TO TEMPLATE

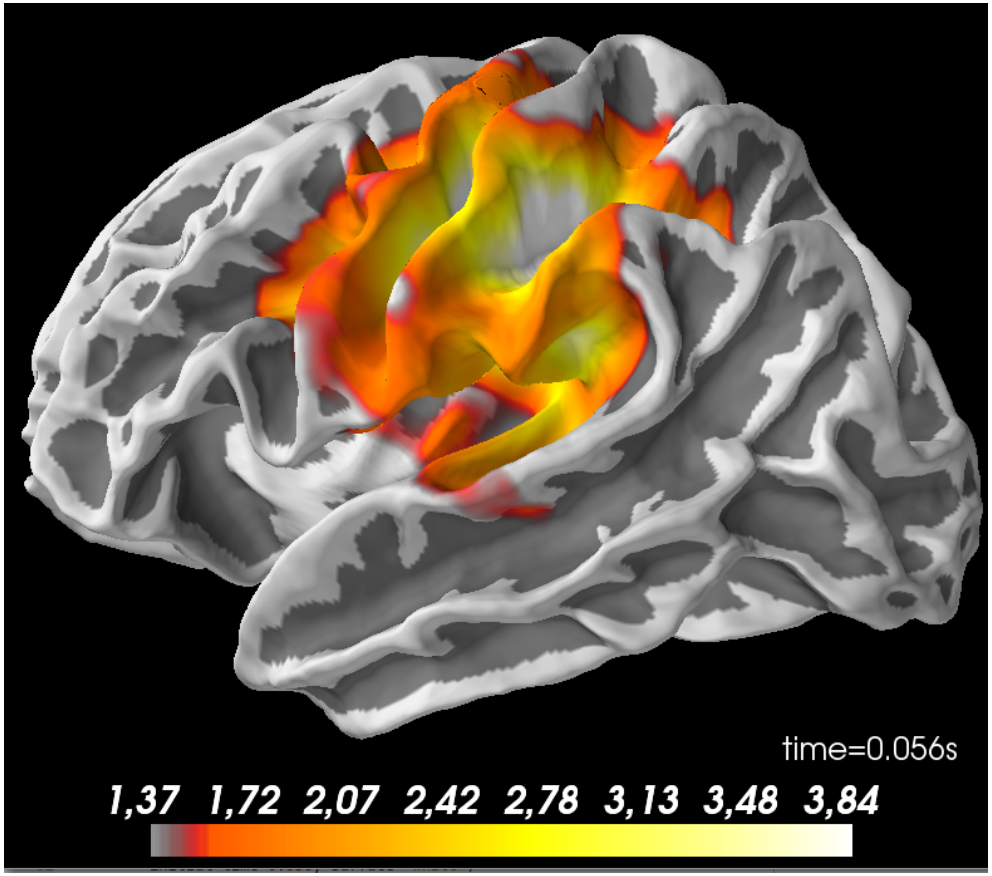


INDIVIDUAL'S ANATOMY



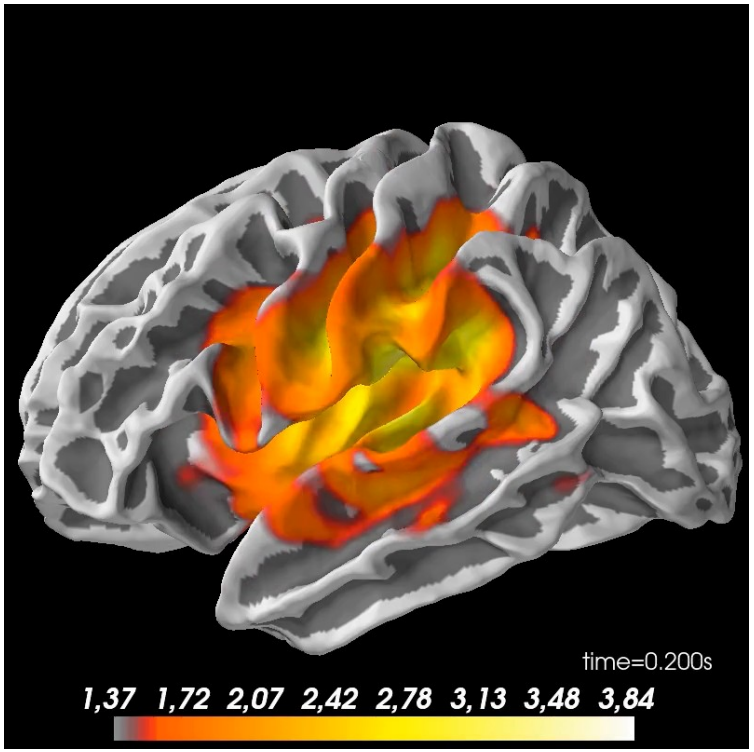
FSAVERAGE TEMPLATE

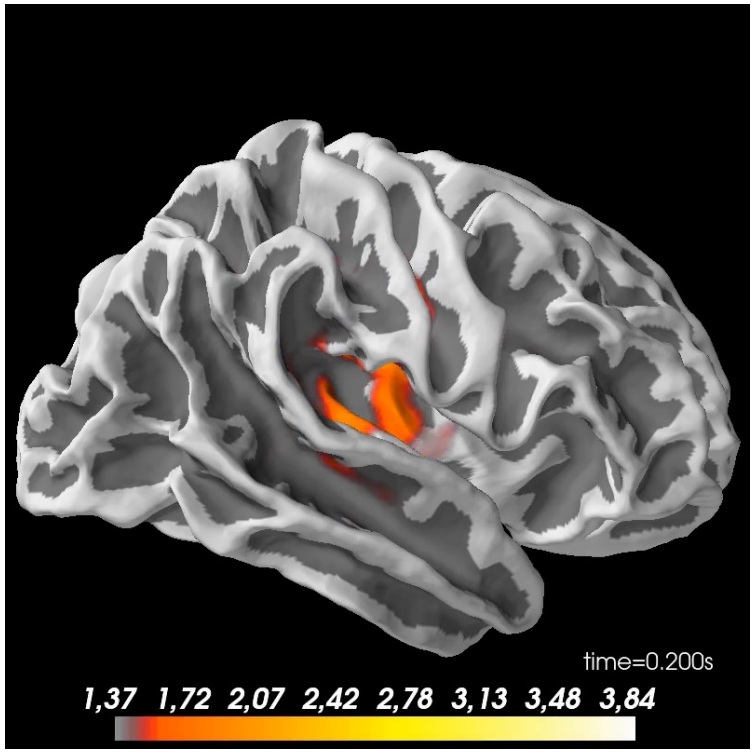
OPERATIONS – GRAND AVERAGE



FSAVERAGE TEMPLATE

OPERATIONS – GRAND AVERAGE VIDEOS





OPERATIONS – STATISTICS

Do *Stimulations* give rise to more activity than *Non-Stimulations* in the primary somatosensory cortex?

How do we decide?

How do we decide?

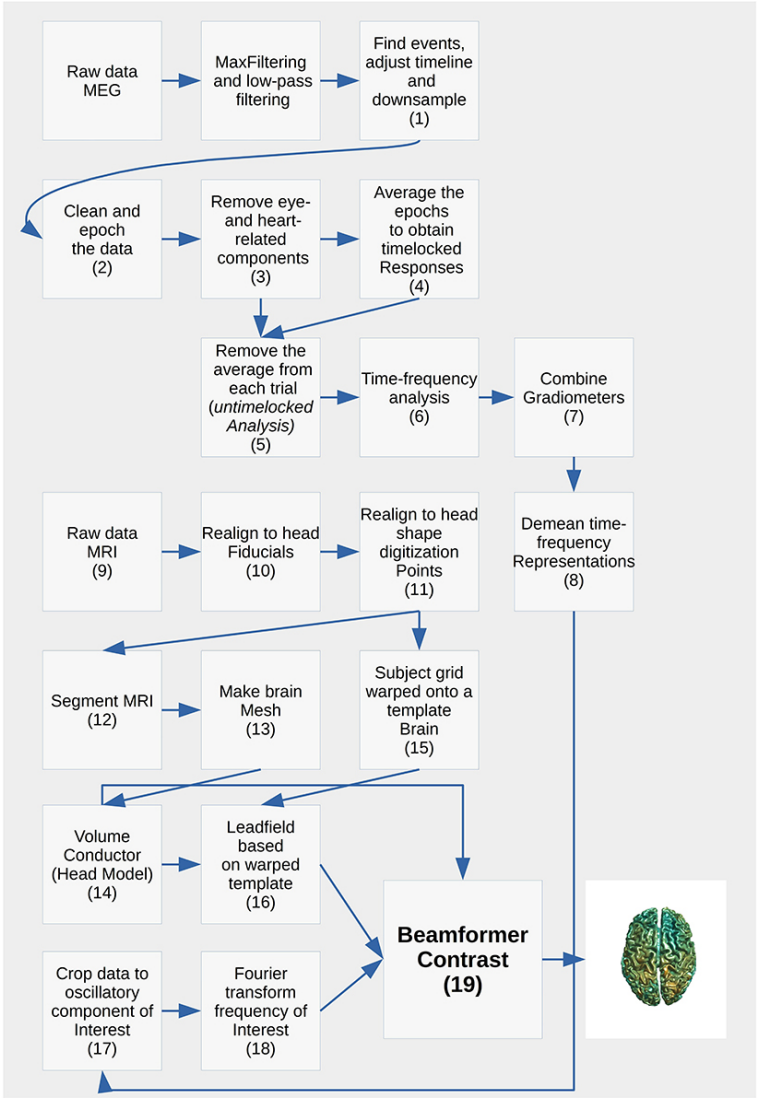
We'll postpone this question until we have gone through the next analysis pipeline

Karolinska Institutet

FIELDTRIP PIPELINE

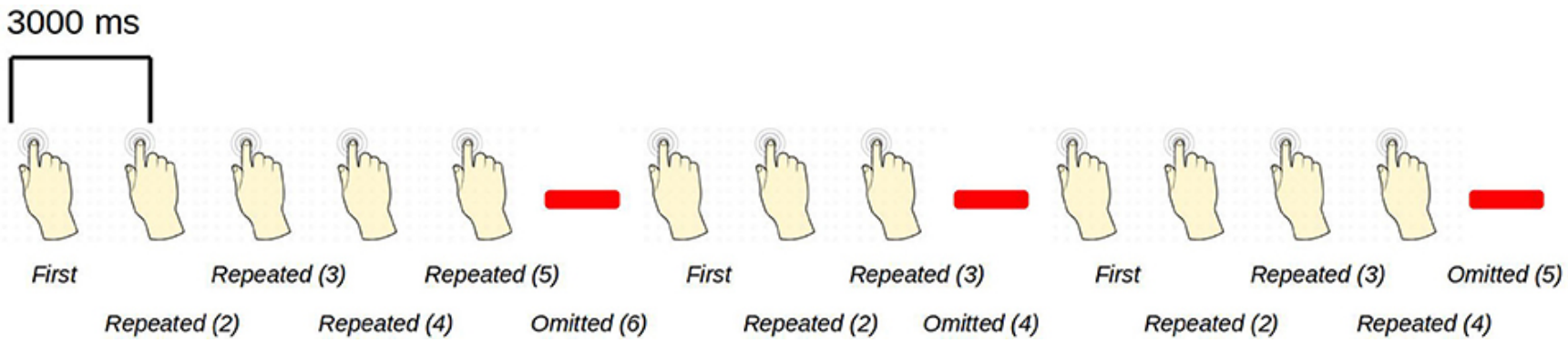
Argument	Purpose
subjects	Subject IDs indicating the directory name of the subject
data_dir	Whether data is MEG or MRI data
function_name	The function that should be applied to all subjects
cfg	Configuration structure, as known from FieldTrip
output	A cell array of name(s) of the output file(s)
input	A cell array of name(s) of the input file(s)
figures_dir	Where figures should be stored (leave empty, [], if no figures are produced)
overwrite	Whether existing output files should be overwritten

Arguments for the function *loop_through_subjects*, which structures input and output of all operations done on single subjects.



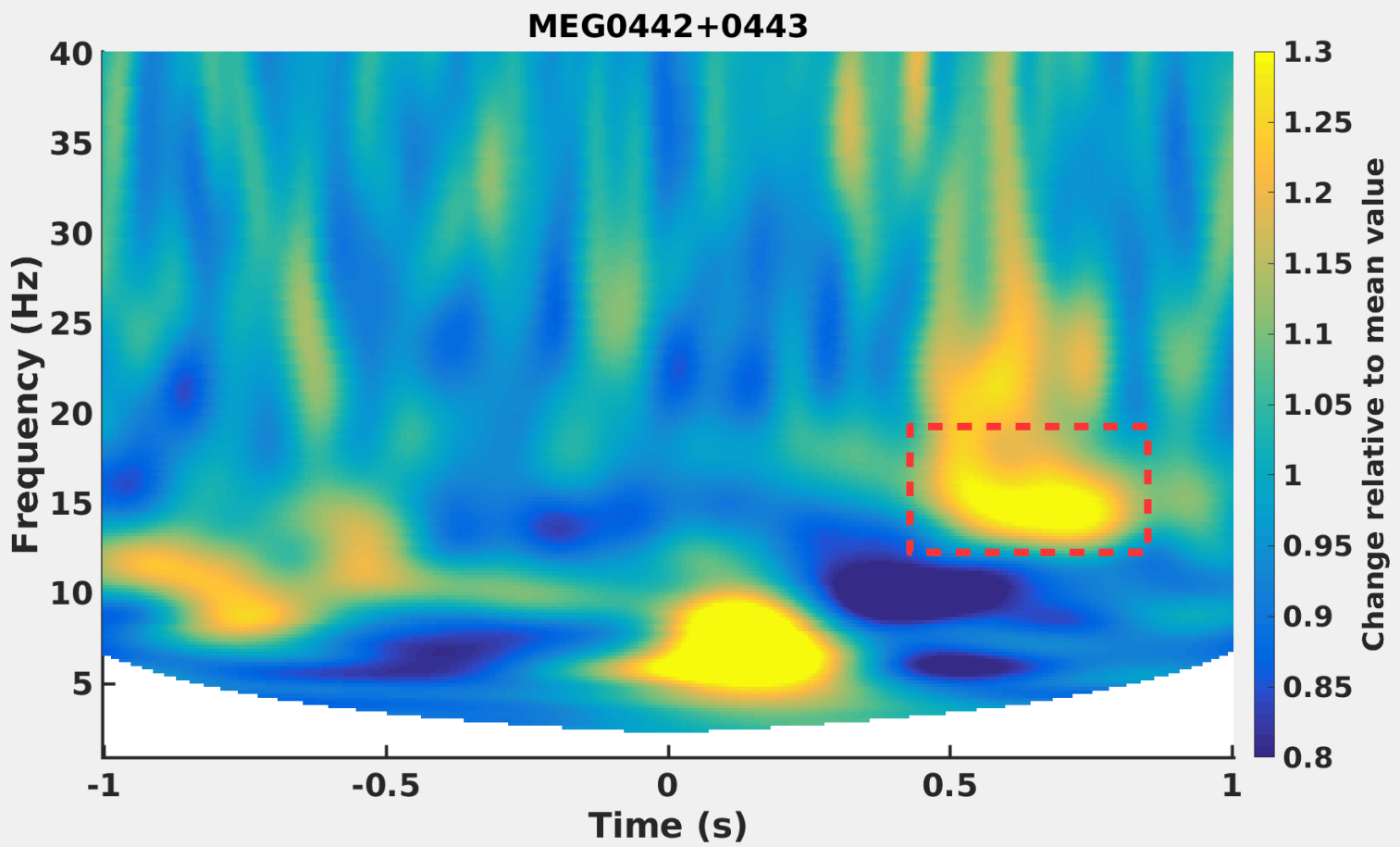
Andersen 2018, Frontiers in Neuroscience

SAME PARADIGM



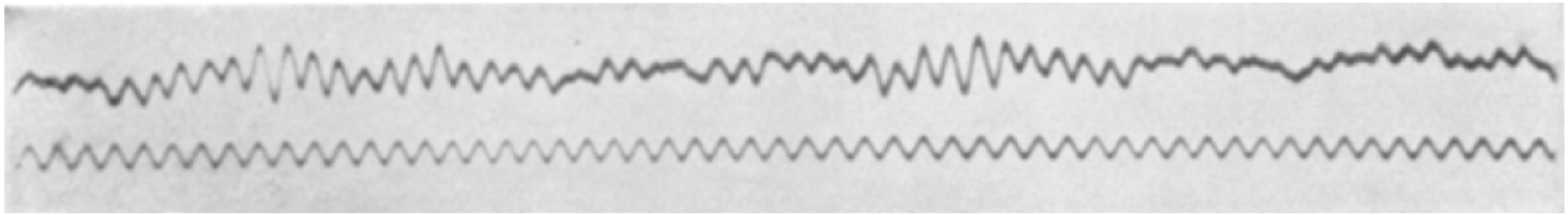
Andersen 2018, Frontiers in Neuroscience

BETA REBOUND



OSCILLATIONS

Berger 1929
Archives für Psychiatrie



First EEG recorded:

Upper row: EEG

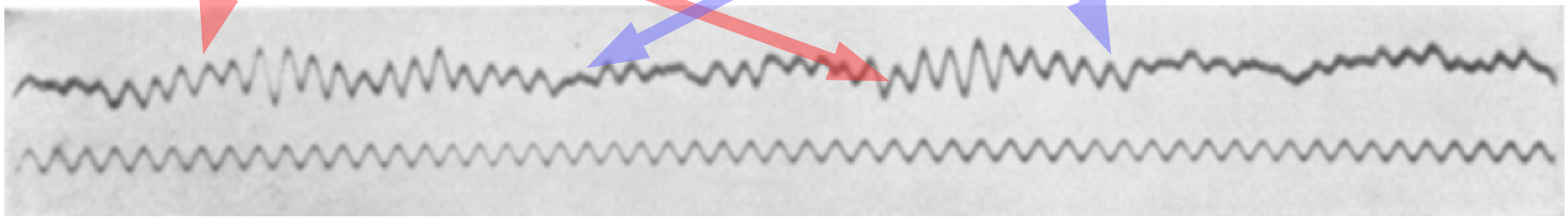
Lower row: 10 Hz (alpha) rhythm

OSCILLATIONS

Subject closed his eyes

Subject opened his eyes

Berger 1929
Archives für Psychiatrie

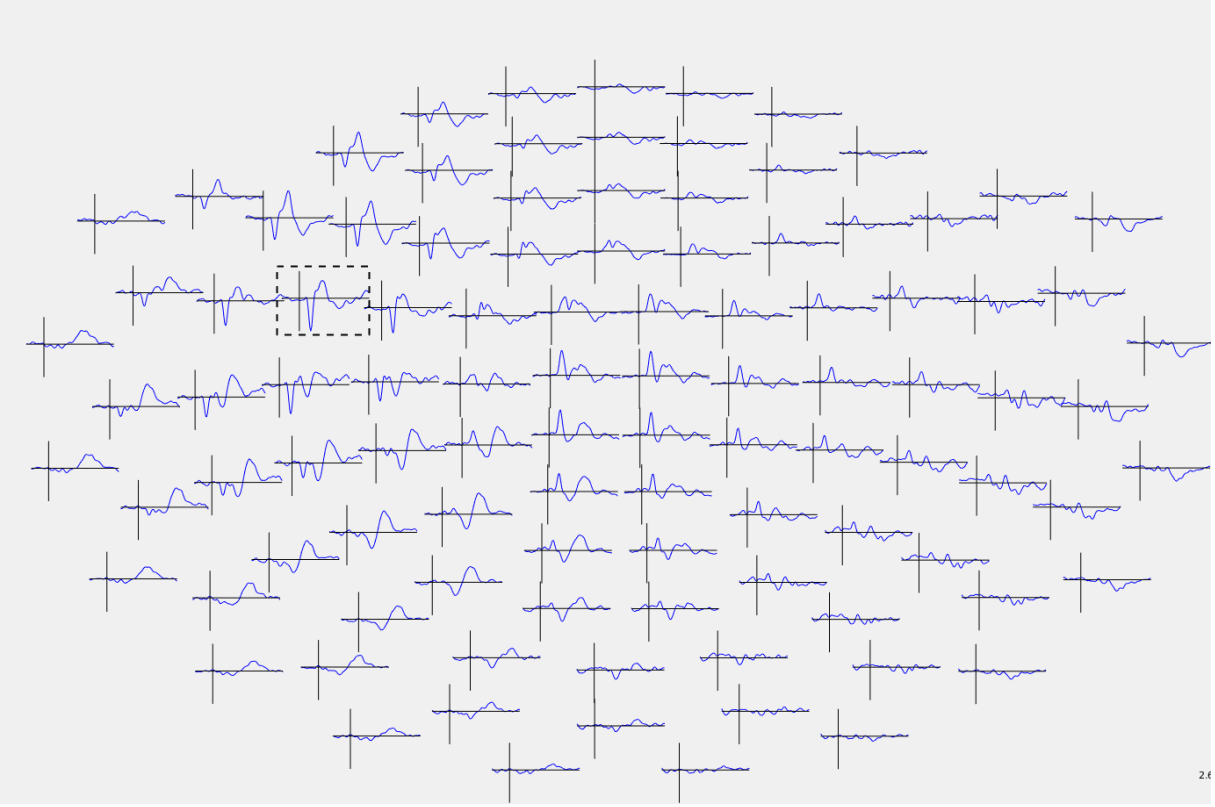


First EEG recorded:

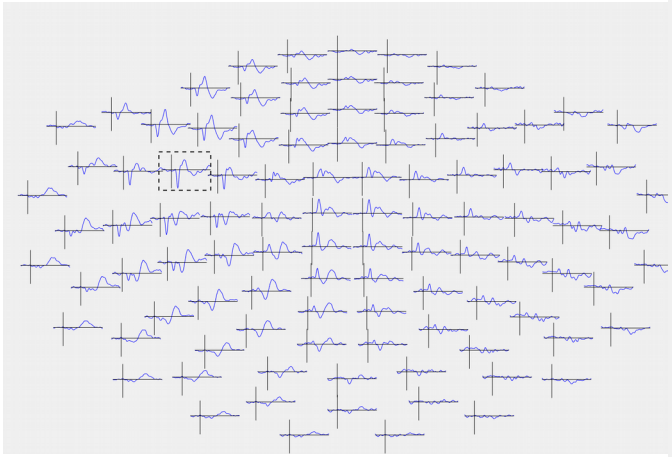
Upper row: EEG

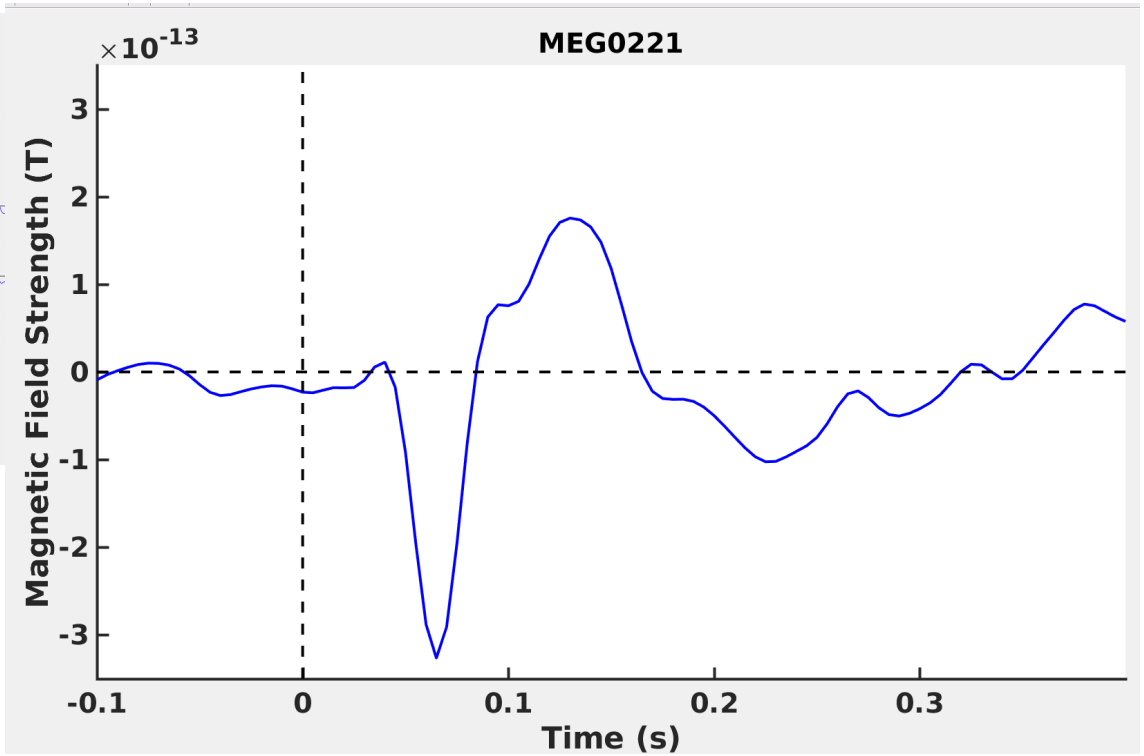
Lower row: 10 Hz (alpha) rhythm

OPERATIONS – TIMELOCKEDS (EVOKED)

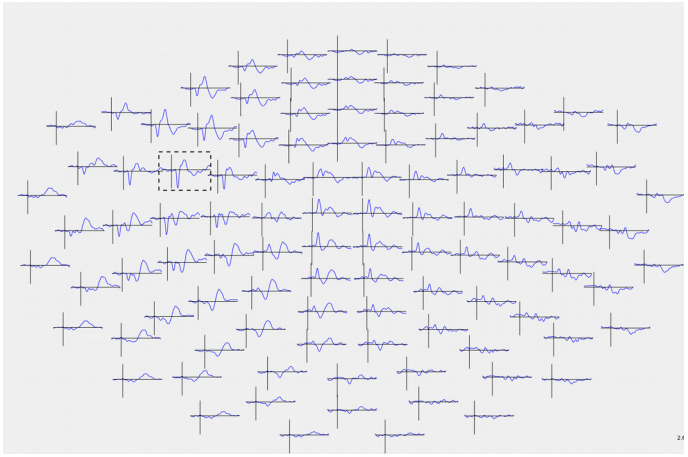


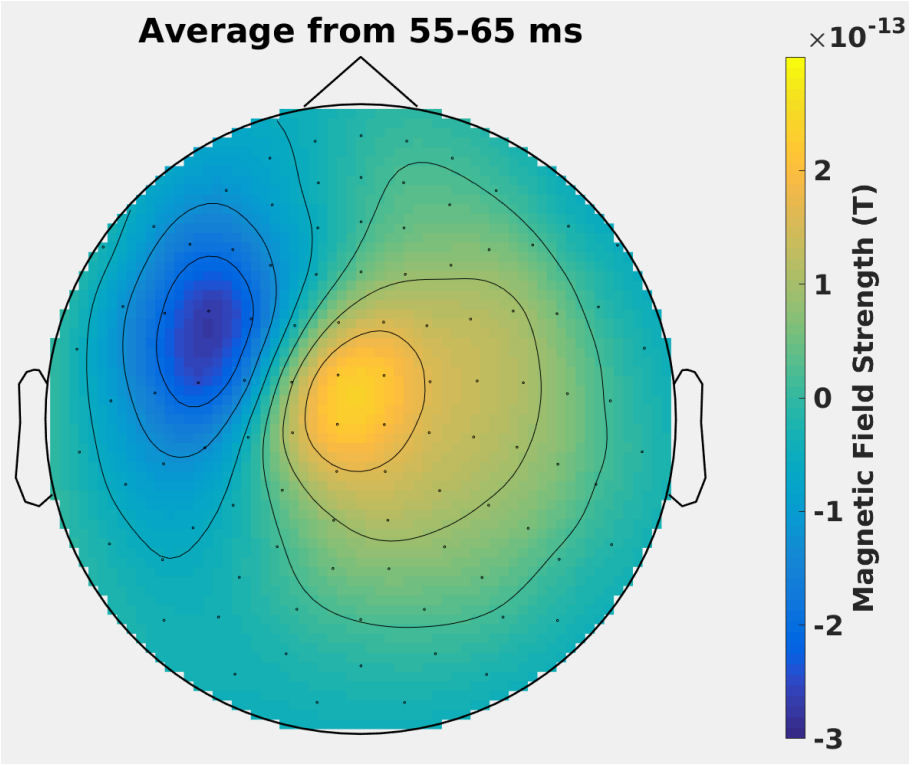
OPERATIONS – TIMELOCKEDS (EVOKED)





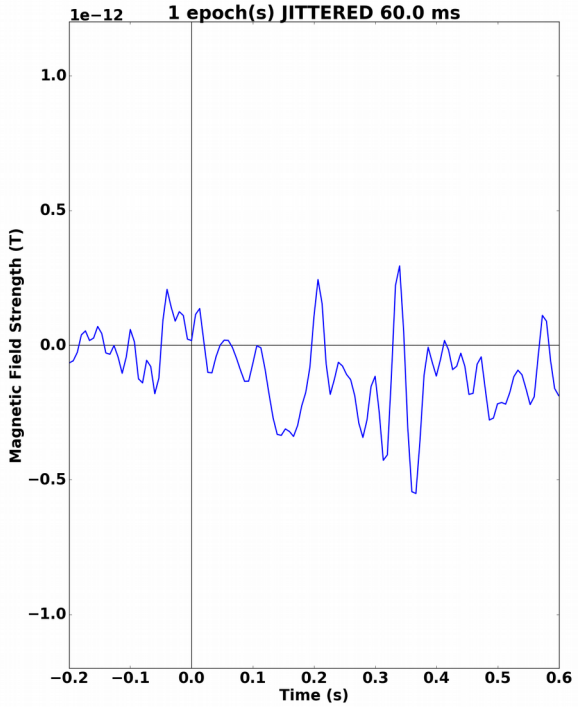
OPERATIONS – TIMELOCKEDS (EVOKED)

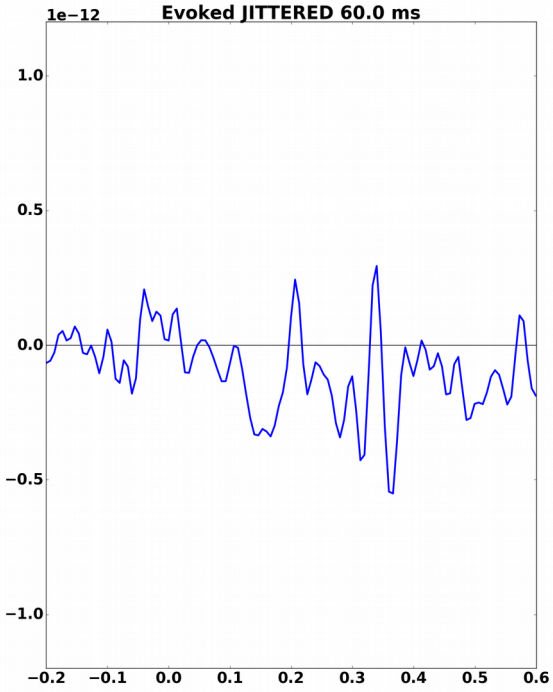


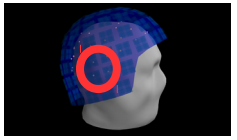


WE SEE NO OSCILLATIONS

EPOCHS AND EVOKED (JITTERED 60 MS)

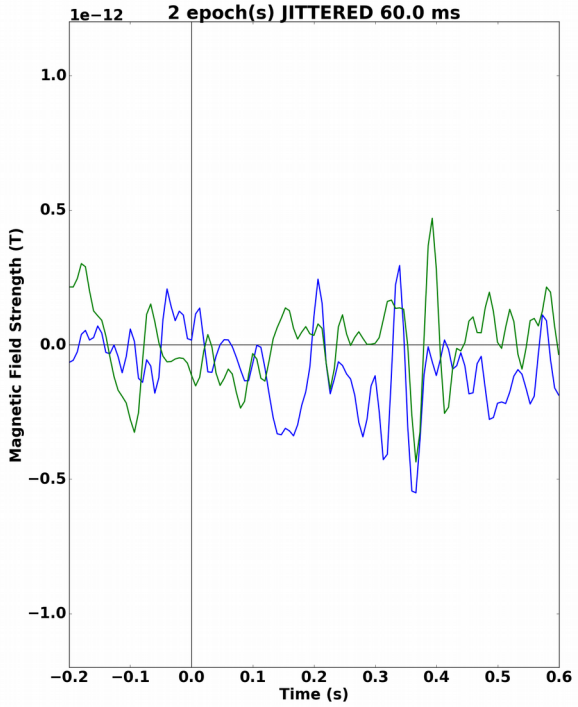


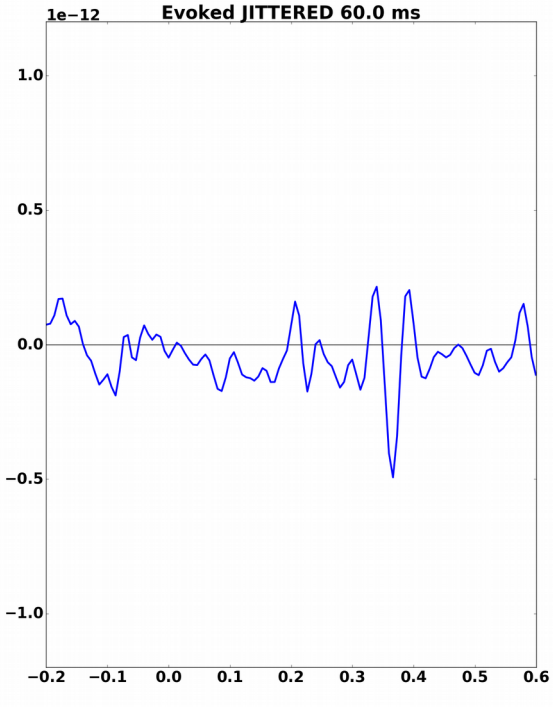


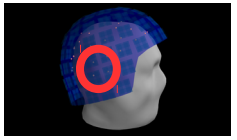


Karolinska Institutet

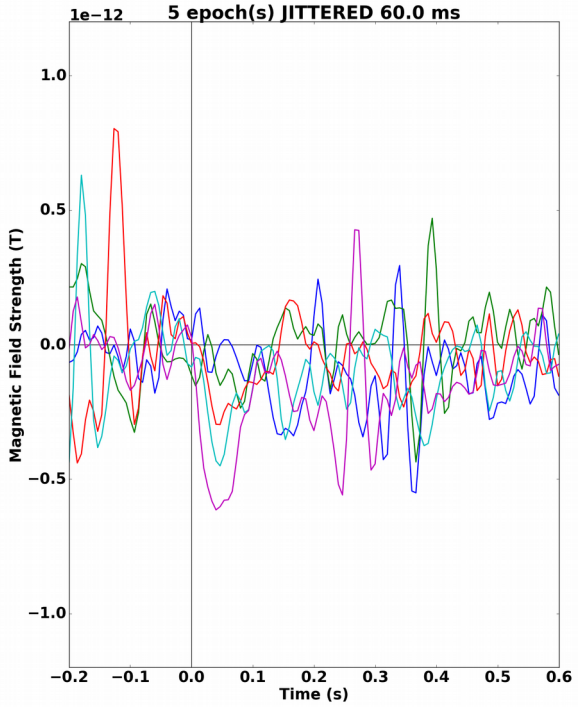
EPOCHS AND EVOKED (JITTERED 60 MS)

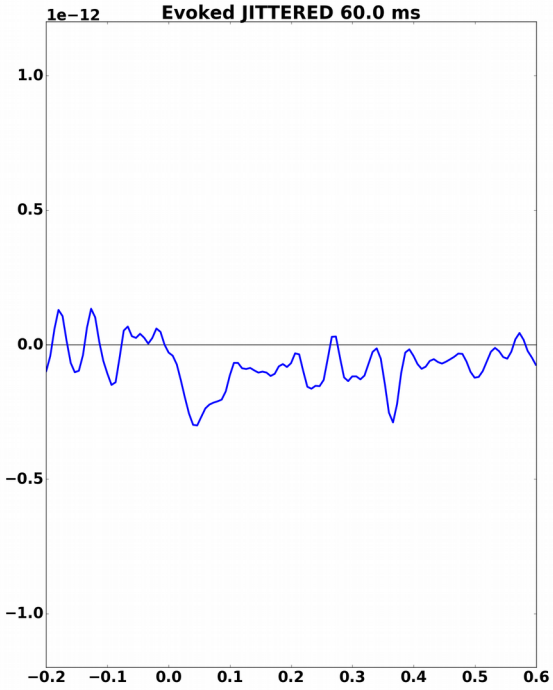


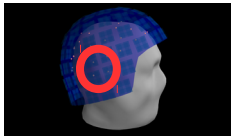




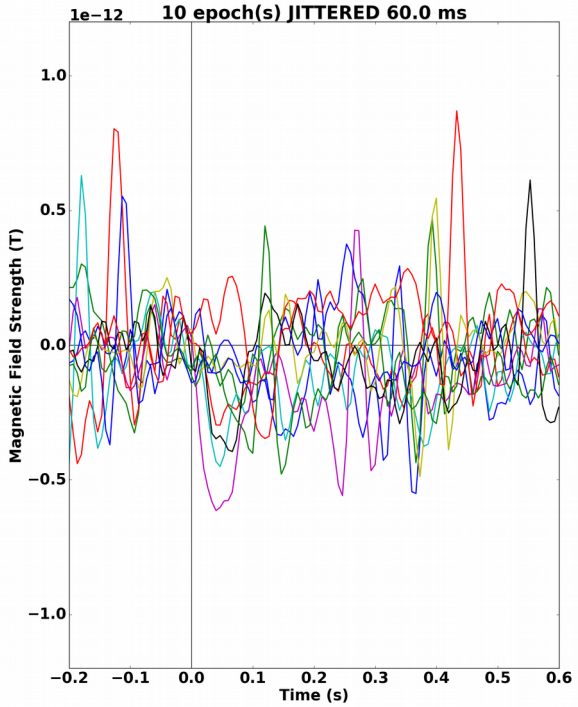
EPOCHS AND EVOKED (JITTERED 60 MS)

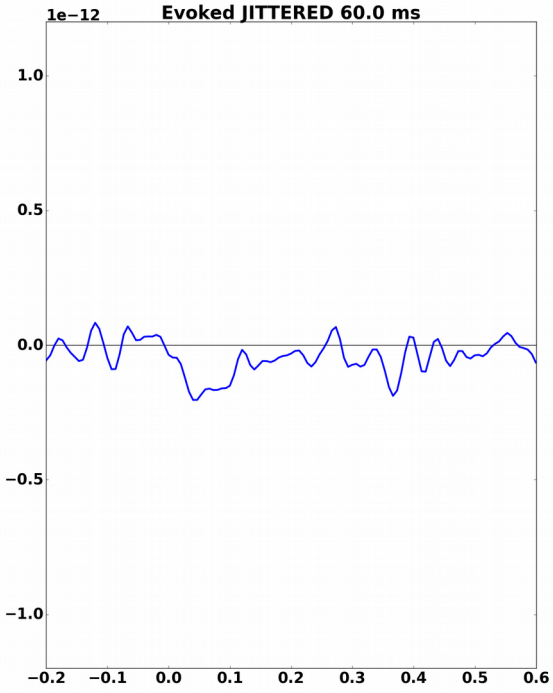


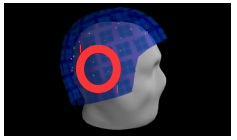




EPOCHS AND EVOKED (JITTERED 60 MS)

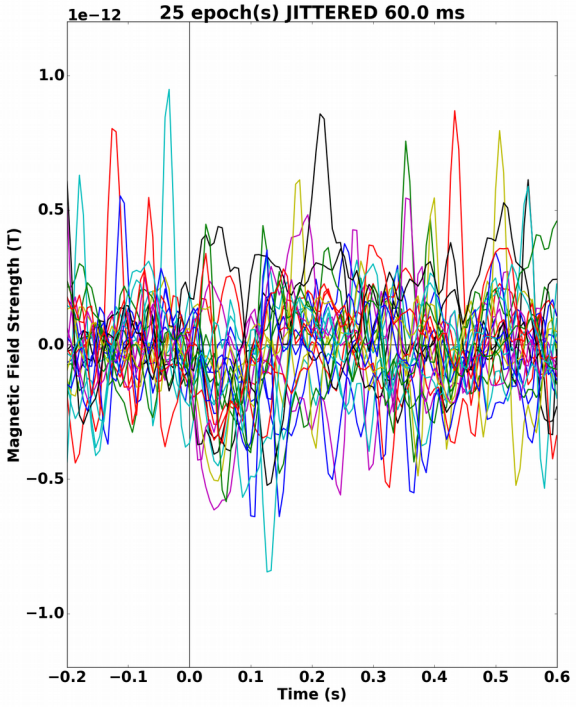


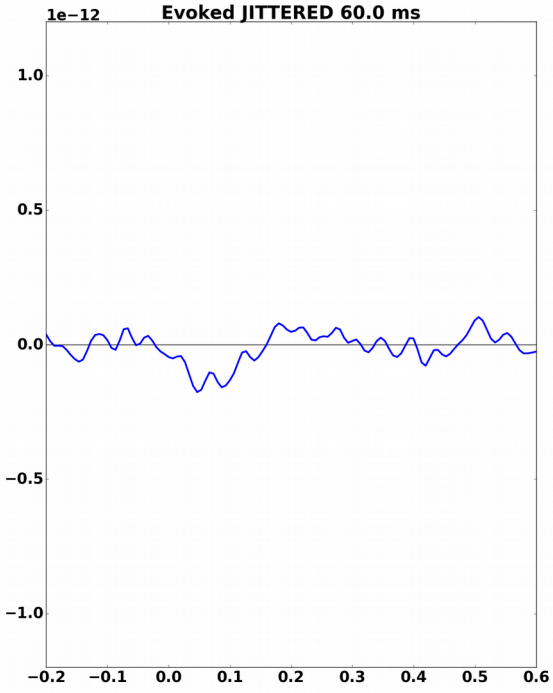


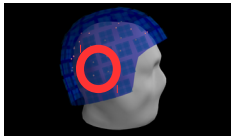


Karolinska Institutet

EPOCHS AND EVOKED (JITTERED 60 MS)

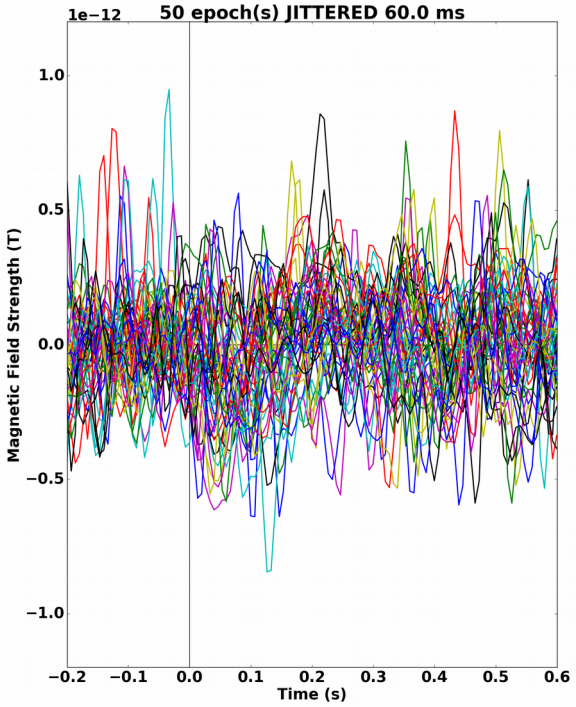


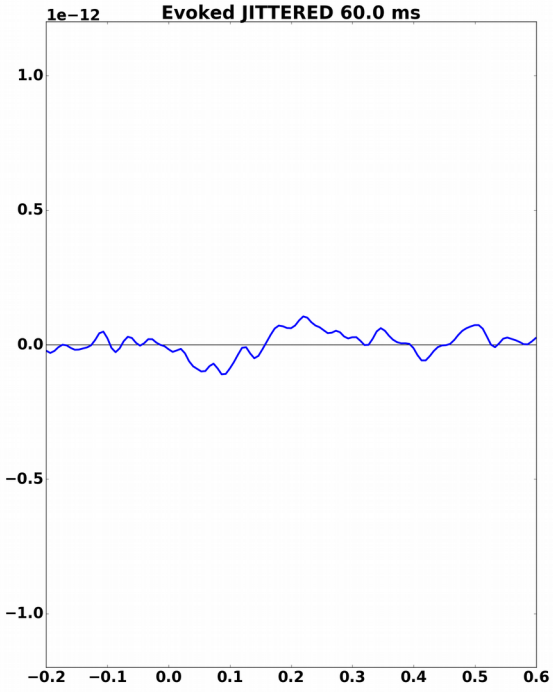


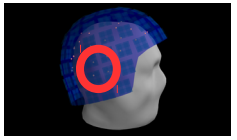


Karolinska Institutet

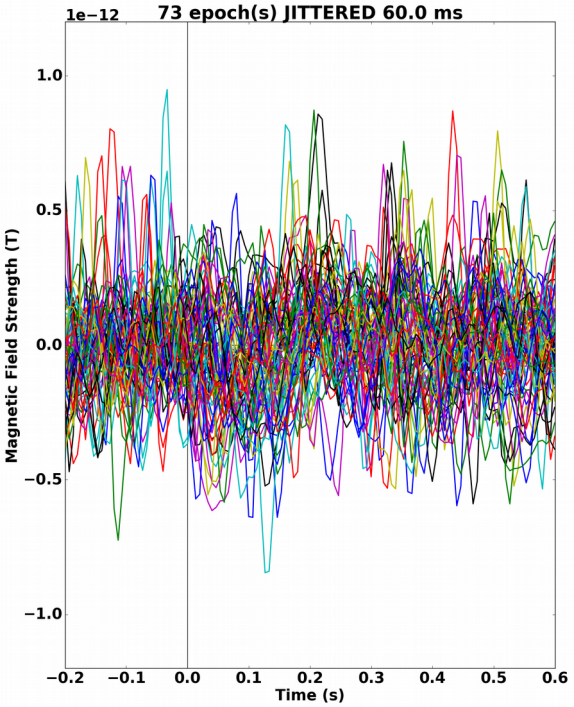
EPOCHS AND EVOKED (JITTERED 60 MS)

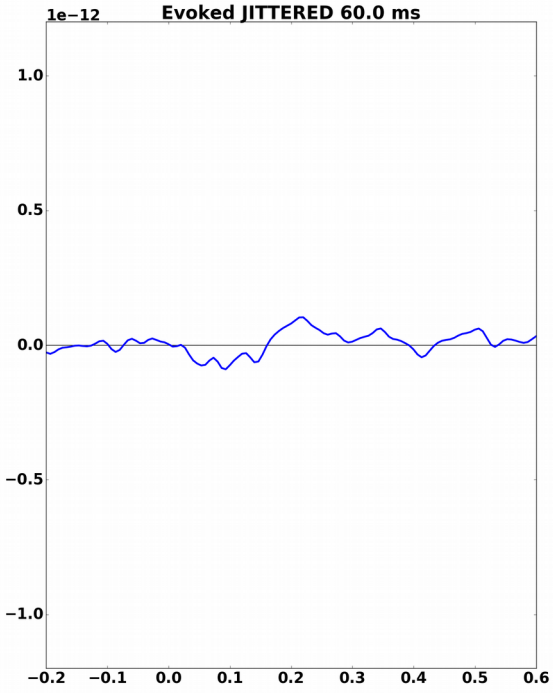


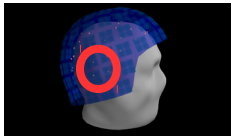




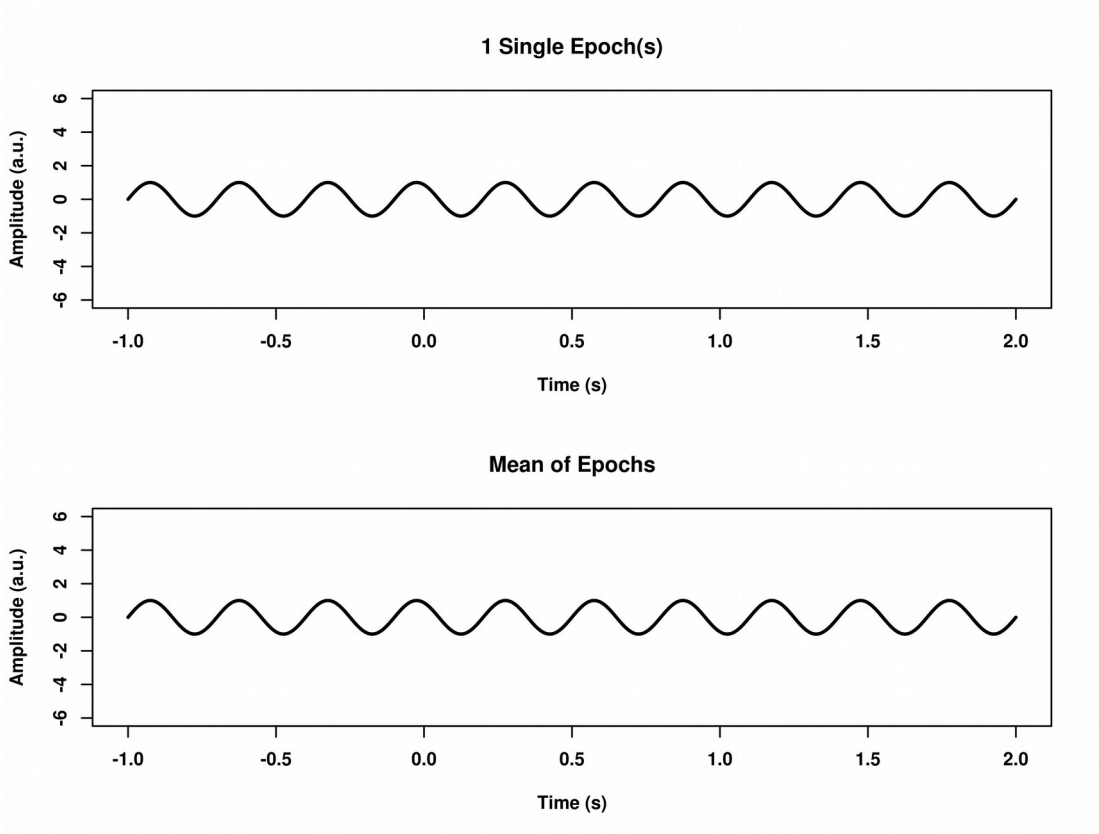
EPOCHS AND EVOKED (JITTERED 60 MS)







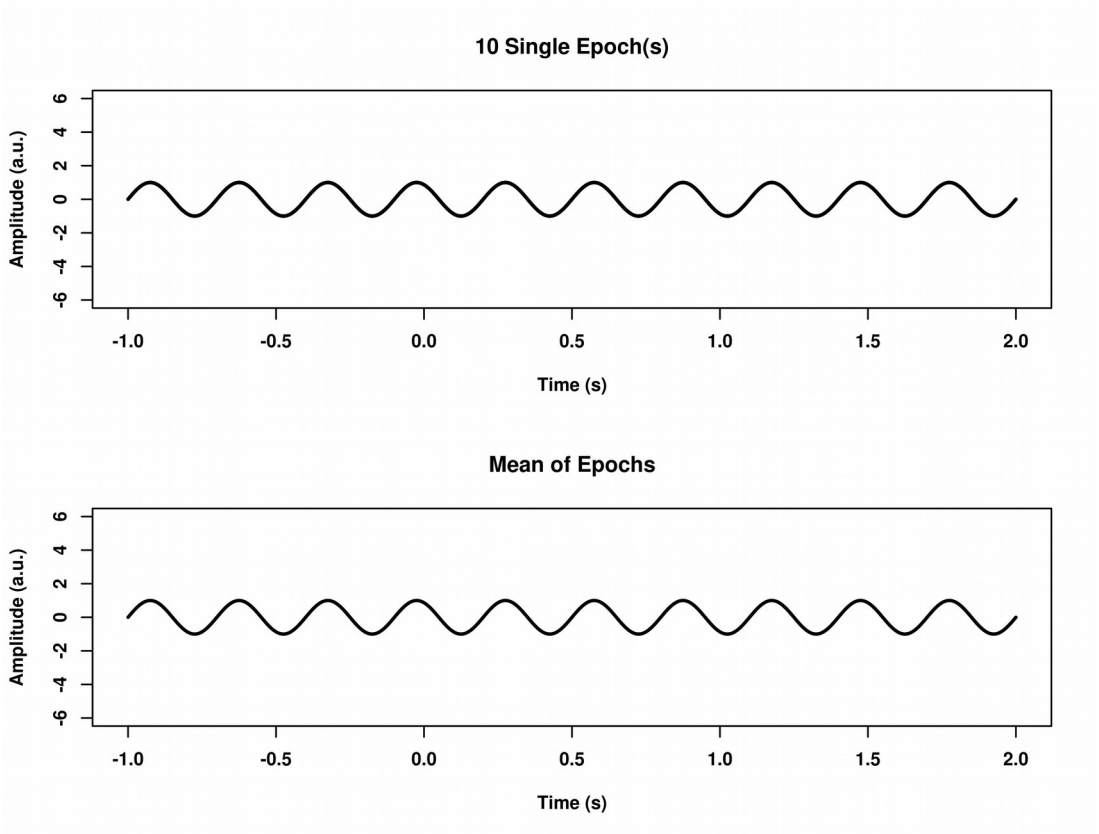
IN-PHASE OR OUT-OF-PHASE



IN PHASE, IDEAL,

Real Mean = 1.0

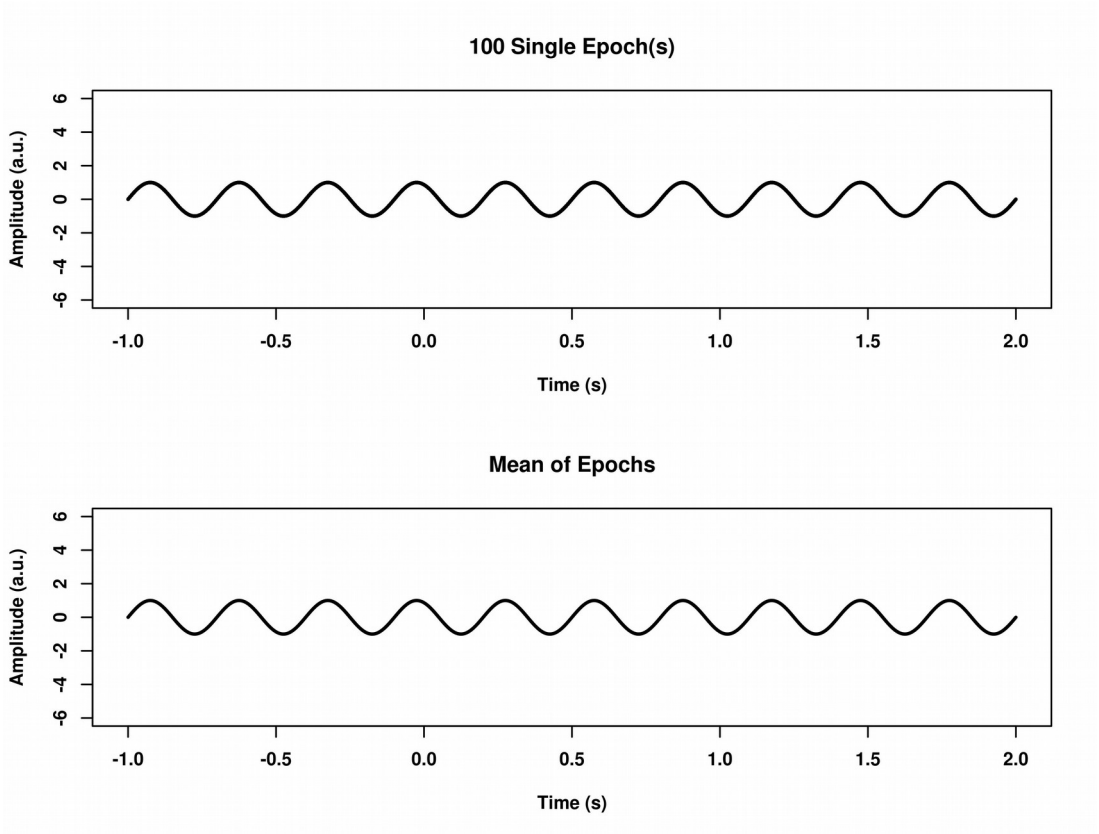
IN-PHASE OR OUT-OF-PHASE



IN PHASE, IDEAL,

Real Mean = 1.0

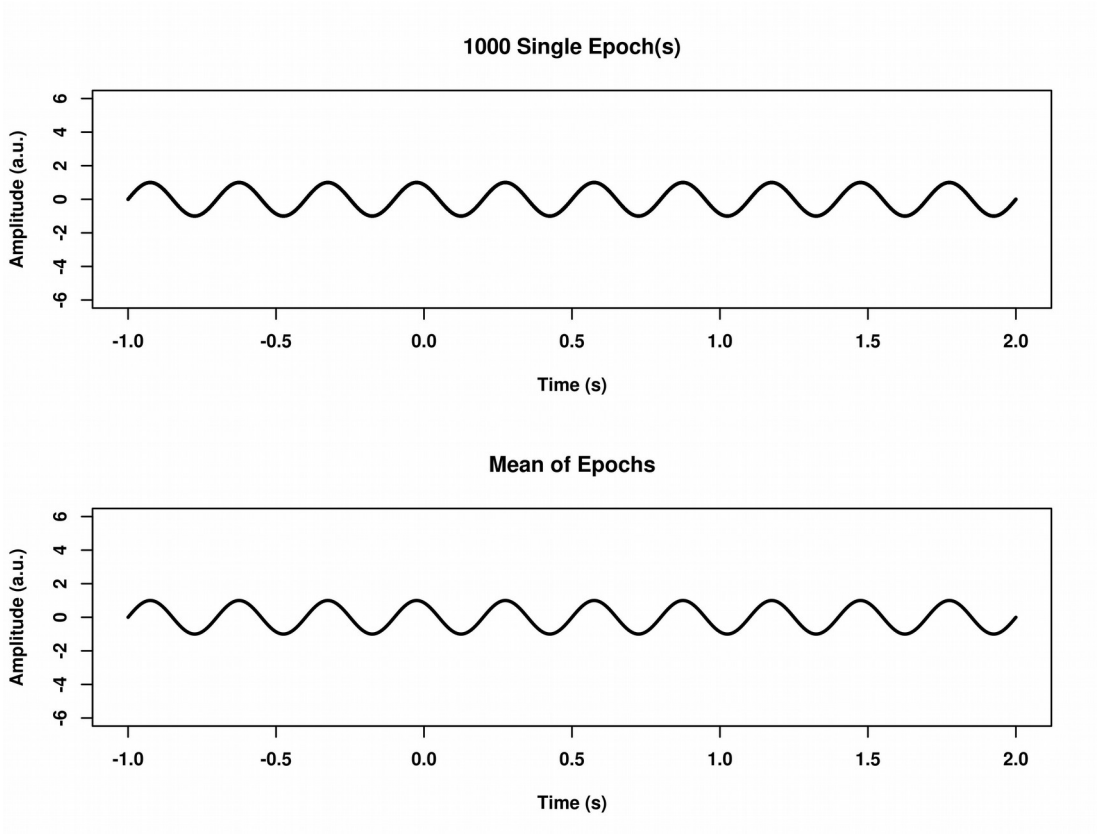
IN-PHASE OR OUT-OF-PHASE



IN PHASE, IDEAL,

Real Mean = 1.0

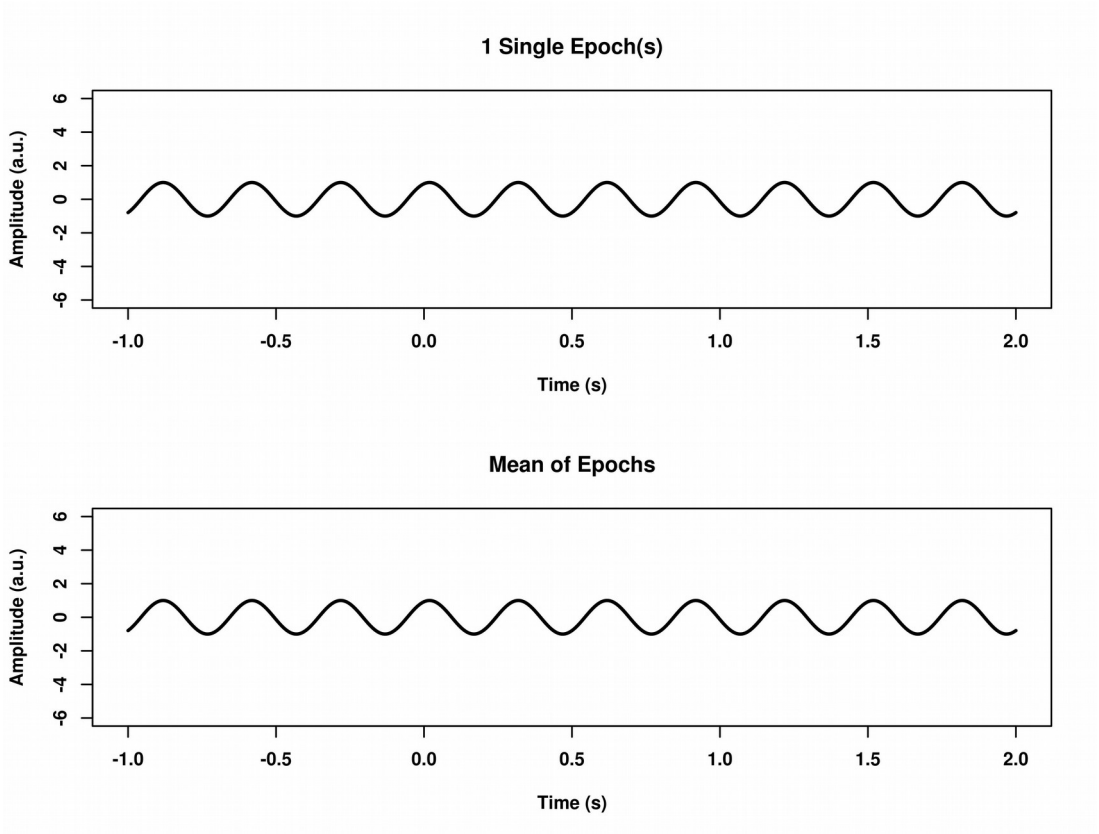
IN-PHASE OR OUT-OF-PHASE



IN PHASE, IDEAL,

Real Mean = 1.0

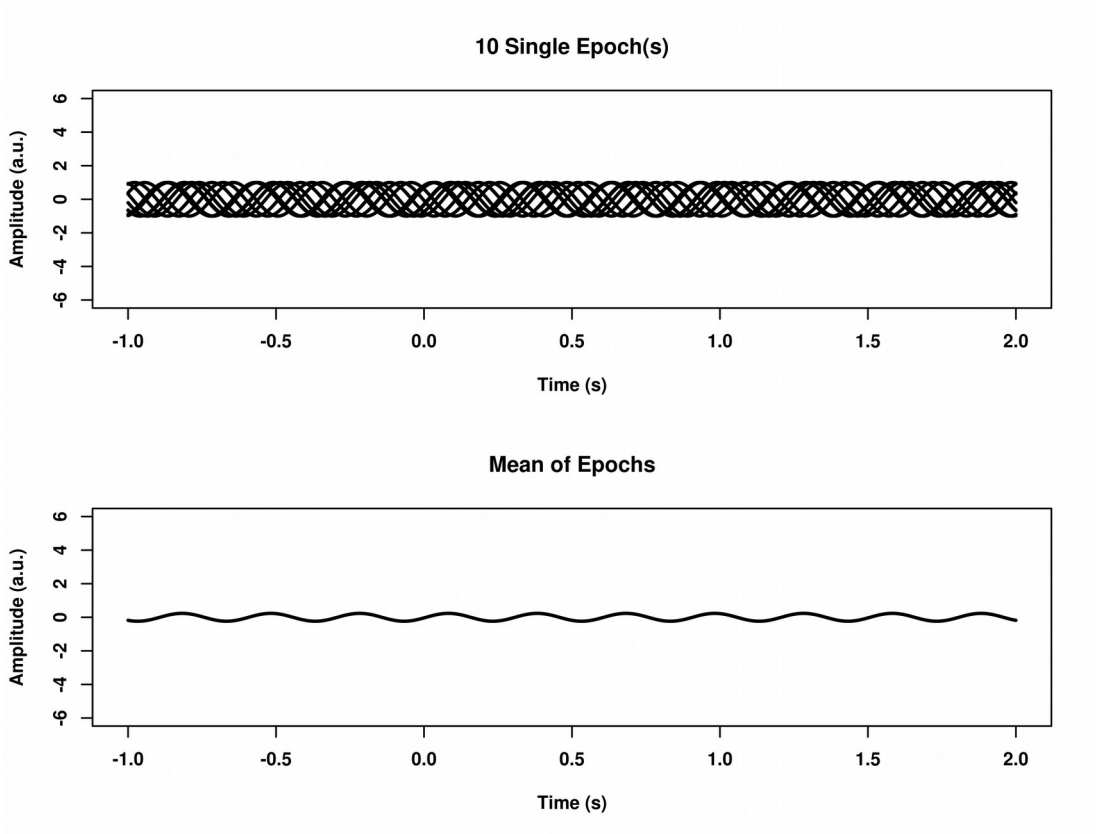
IN-PHASE OR OUT-OF-PHASE



OUT OF PHASE,

IDEAL, Real Mean = 1.0

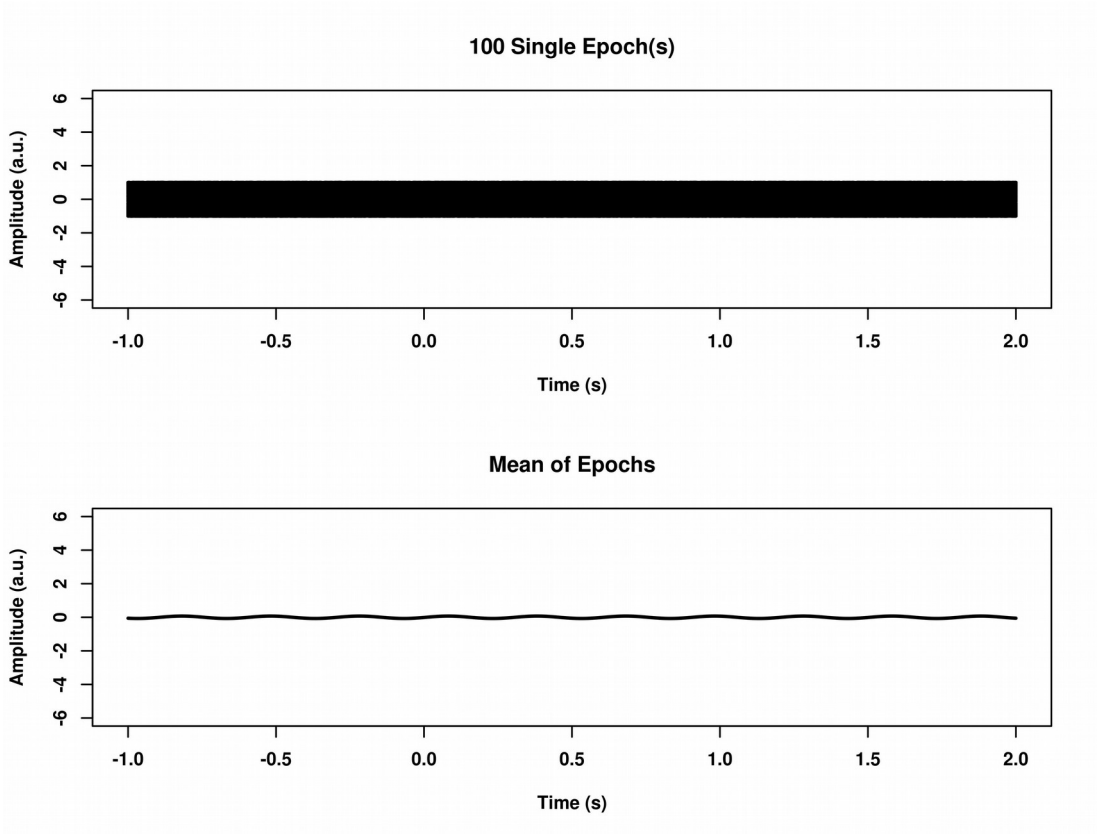
IN-PHASE OR OUT-OF-PHASE



OUT OF PHASE,

IDEAL, Real Mean = 1.0

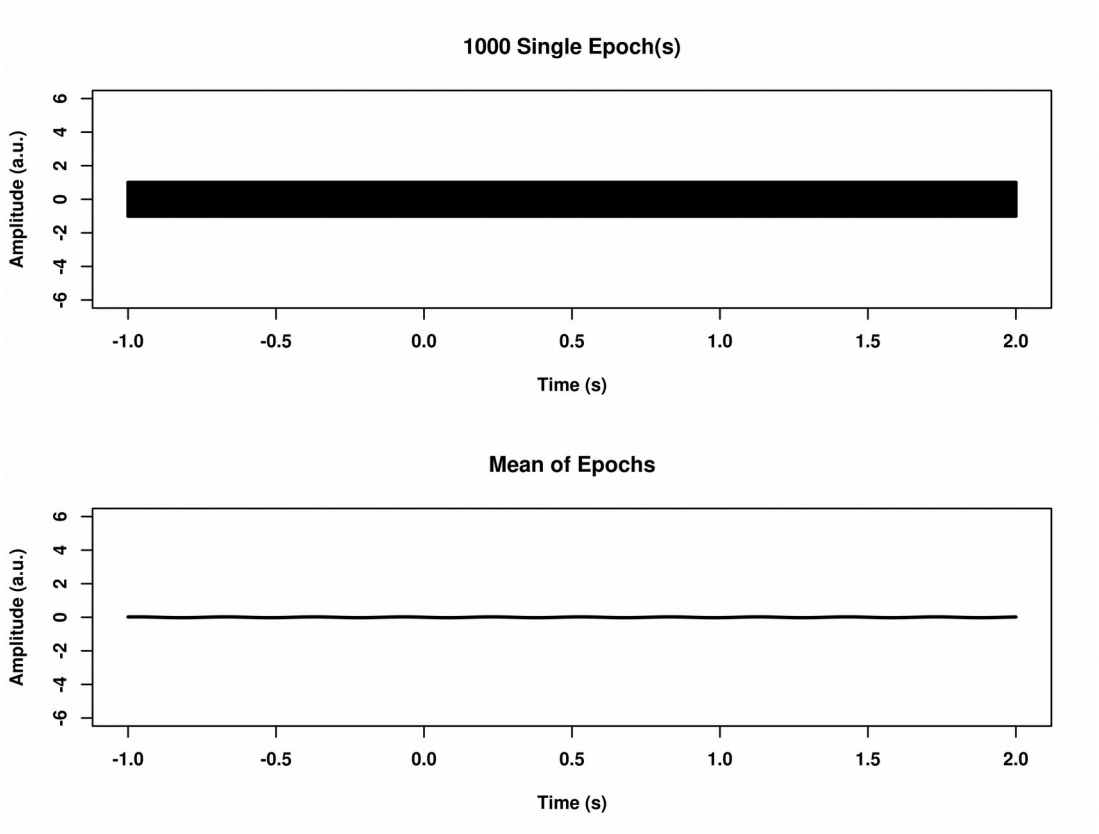
IN-PHASE OR OUT-OF-PHASE



OUT OF PHASE,

IDEAL, Real Mean = 1.0

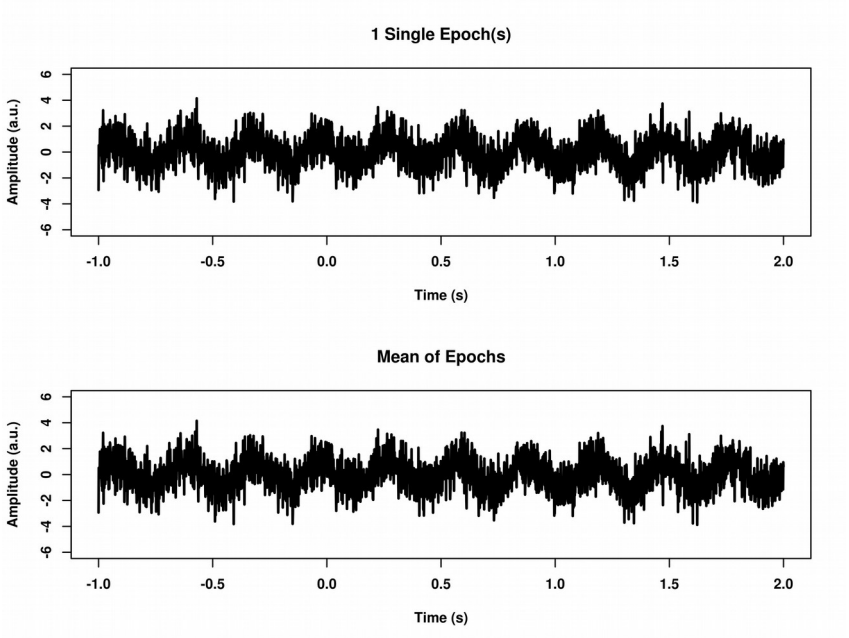
IN-PHASE OR OUT-OF-PHASE



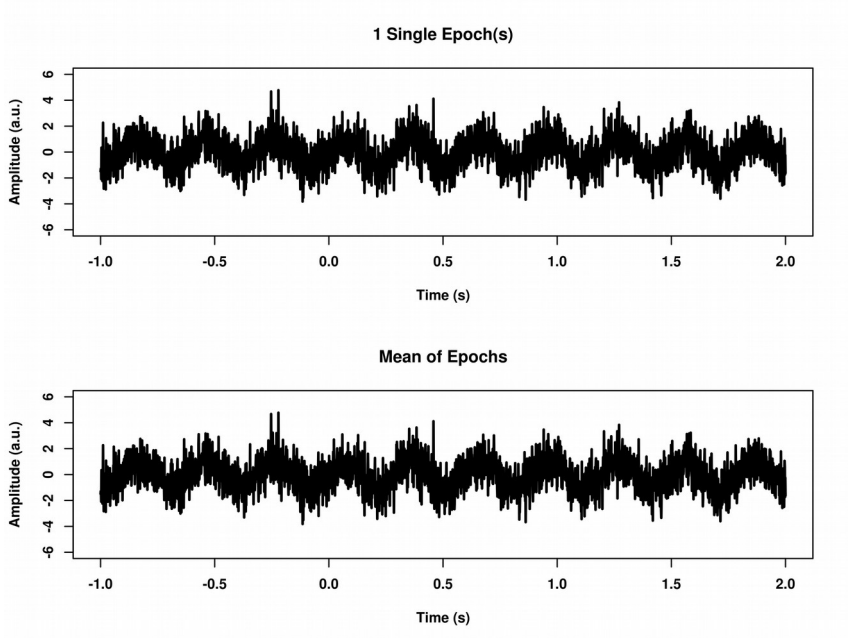
OUT OF PHASE,

IDEAL, Real Mean = 1.0

IN-PHASE OR OUT-OF-PHASE WITH NOISE



IN PHASE

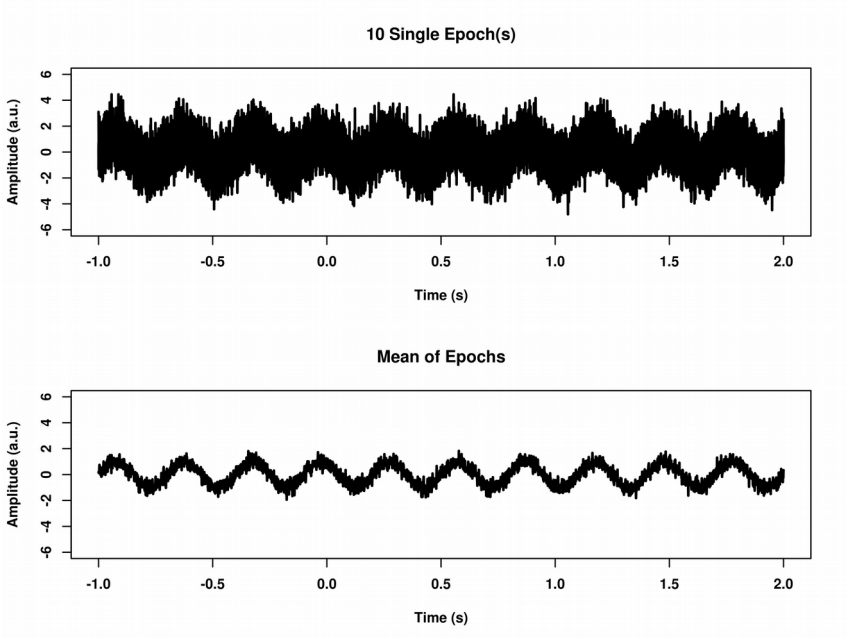


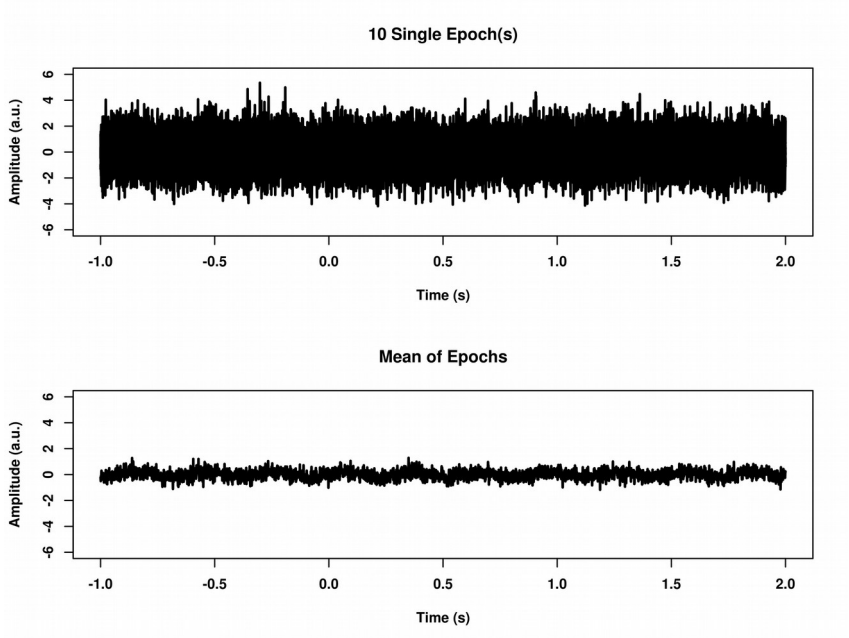
OUT OF PHASE

More realistic

Real Mean = 1.0

IN-PHASE OR OUT-OF-PHASE WITH NOISE





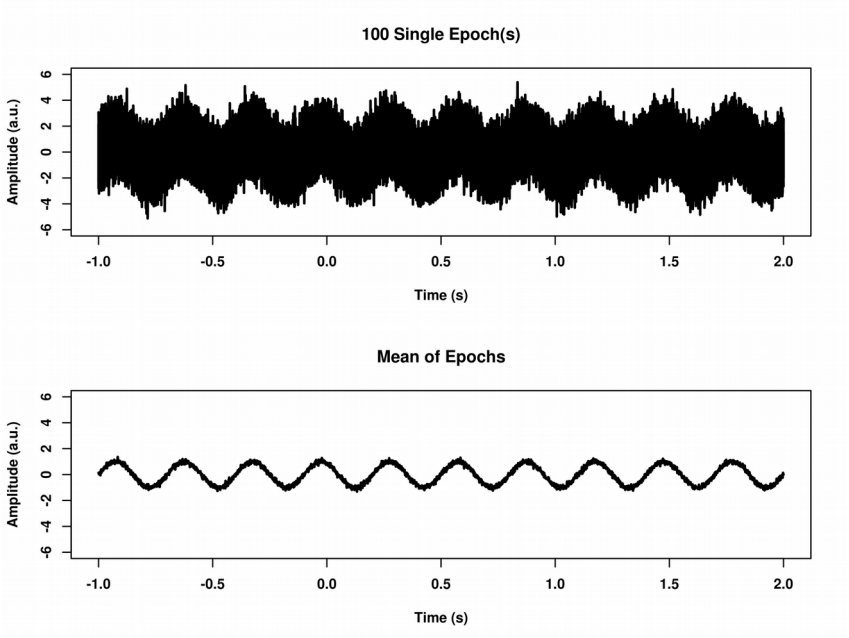
IN PHASE

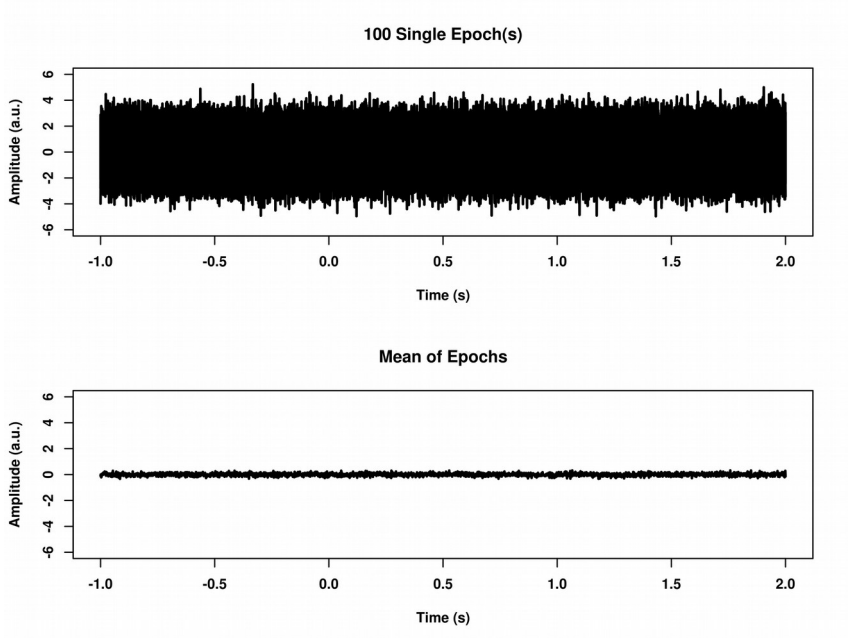
OUT OF PHASE

More realistic

Real Mean = 1.0

IN-PHASE OR OUT-OF-PHASE WITH NOISE





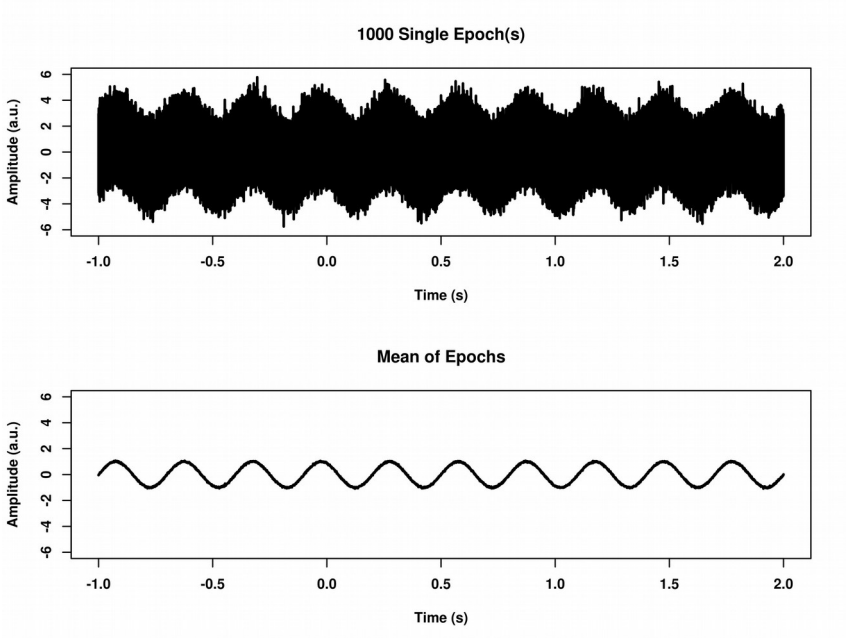
IN PHASE

OUT OF PHASE

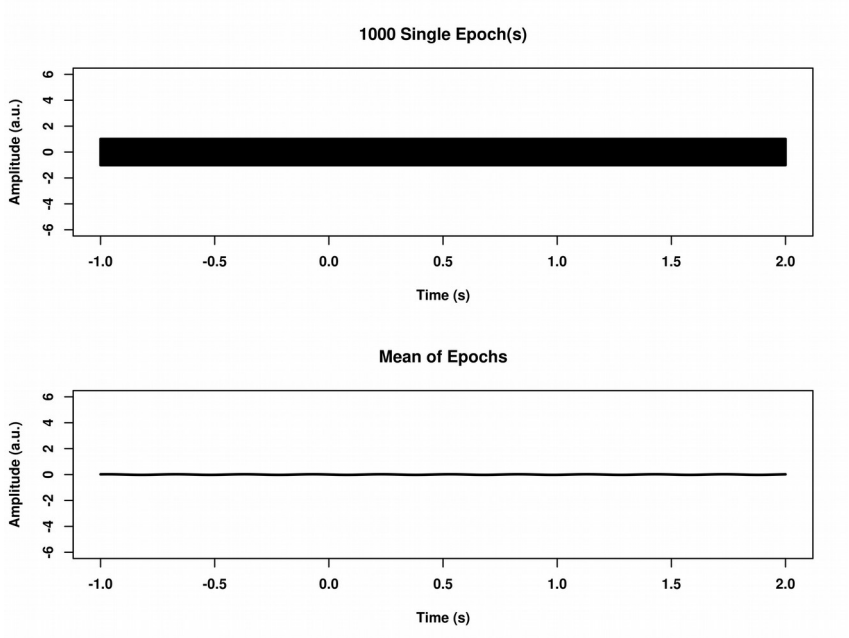
More realistic

Real Mean = 1.0

IN-PHASE OR OUT-OF-PHASE WITH NOISE



IN PHASE

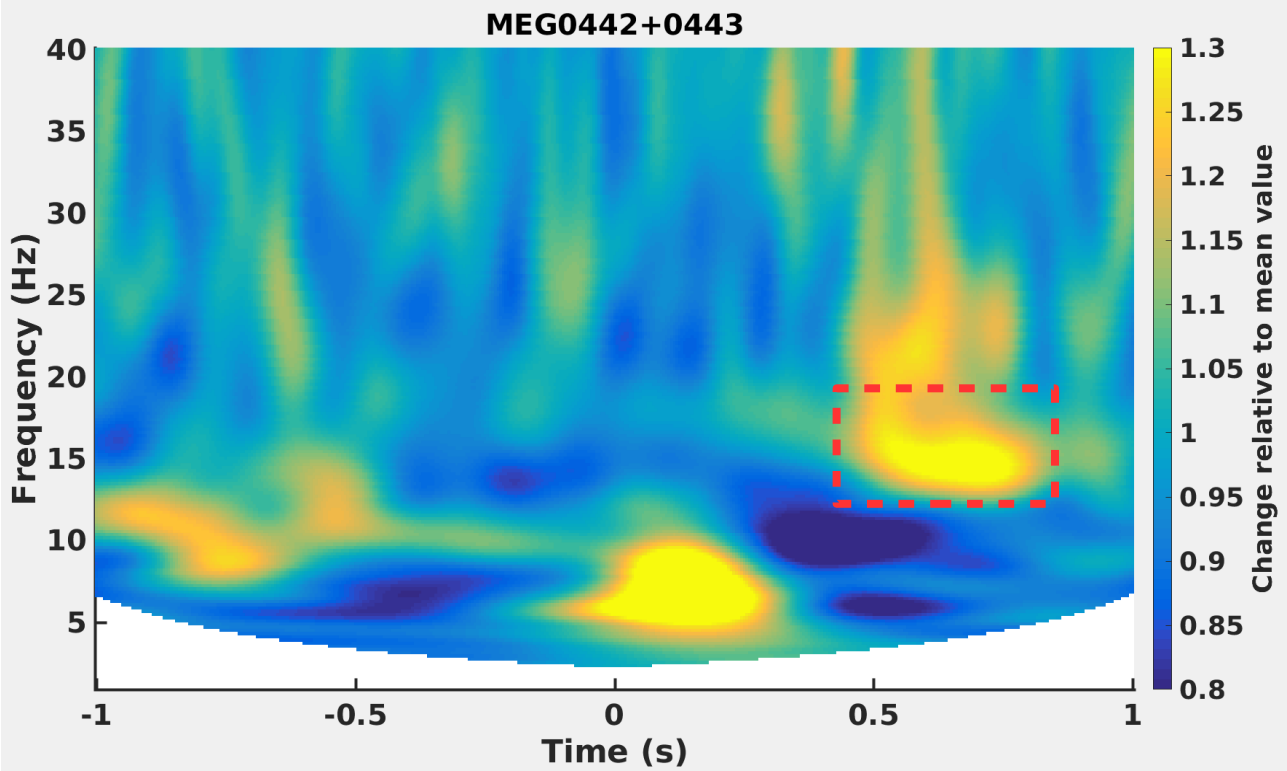


OUT OF PHASE

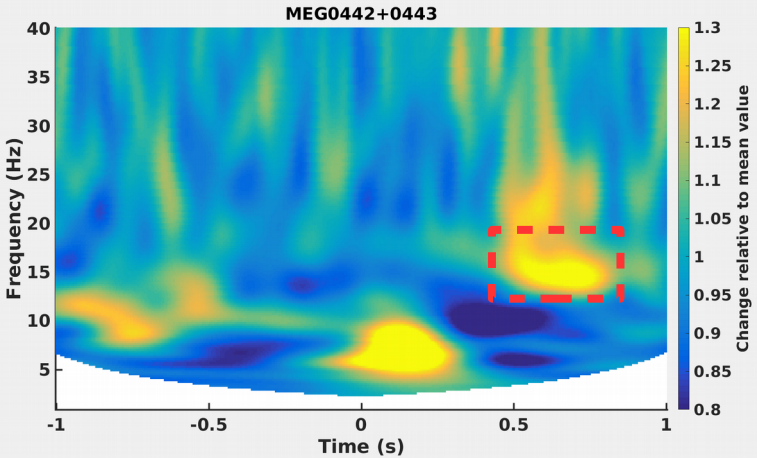
More realistic

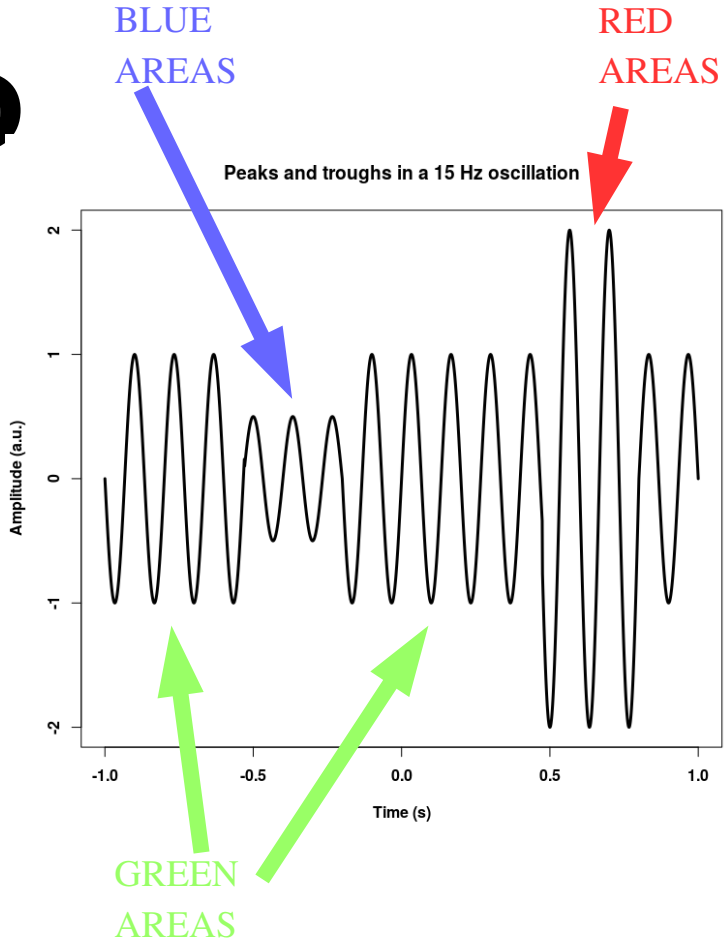
Real Mean = 1.0

BETA REBOUND – TIME-FREQUENCY REPRESENTATION (TFR)



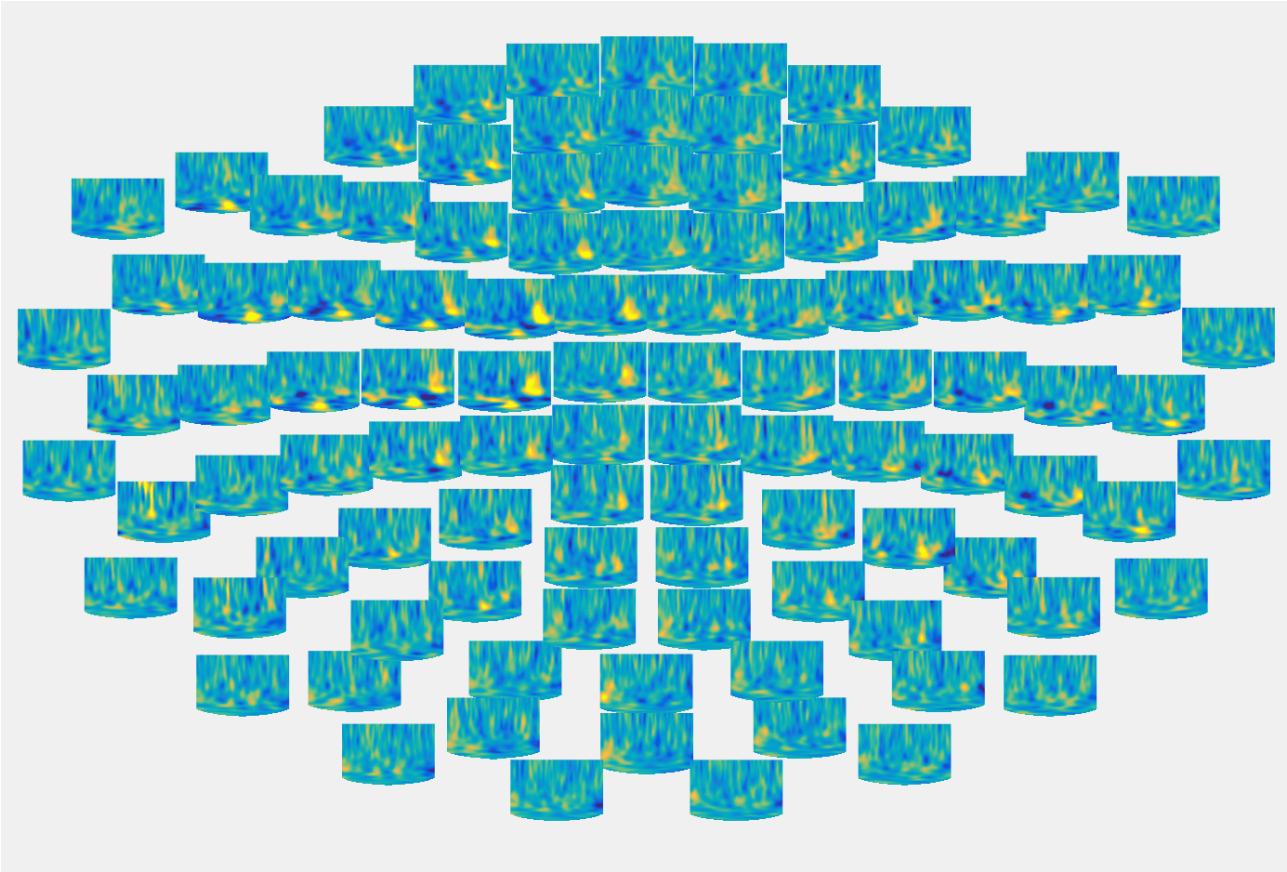
BETA REBOUND



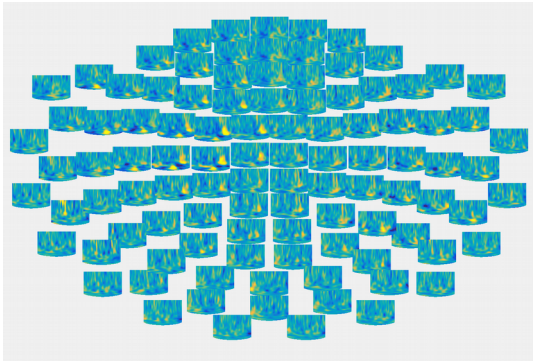


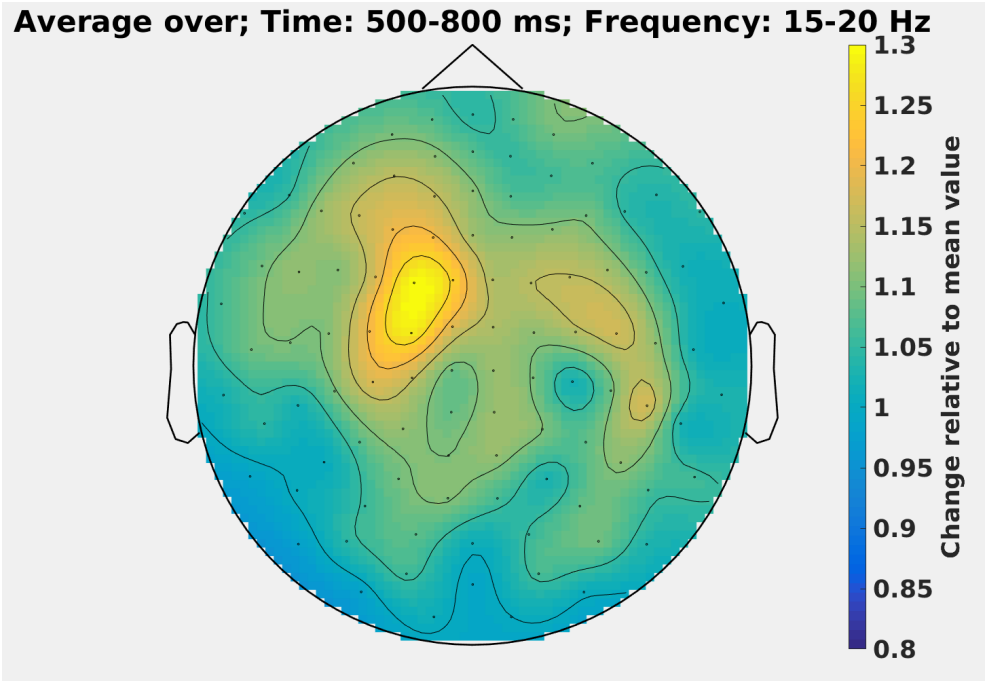
THESE CAN BE CREATED FOR ALL FREQUENCY BANDS

OPERATIONS – TFR



OPERATIONS – TFR

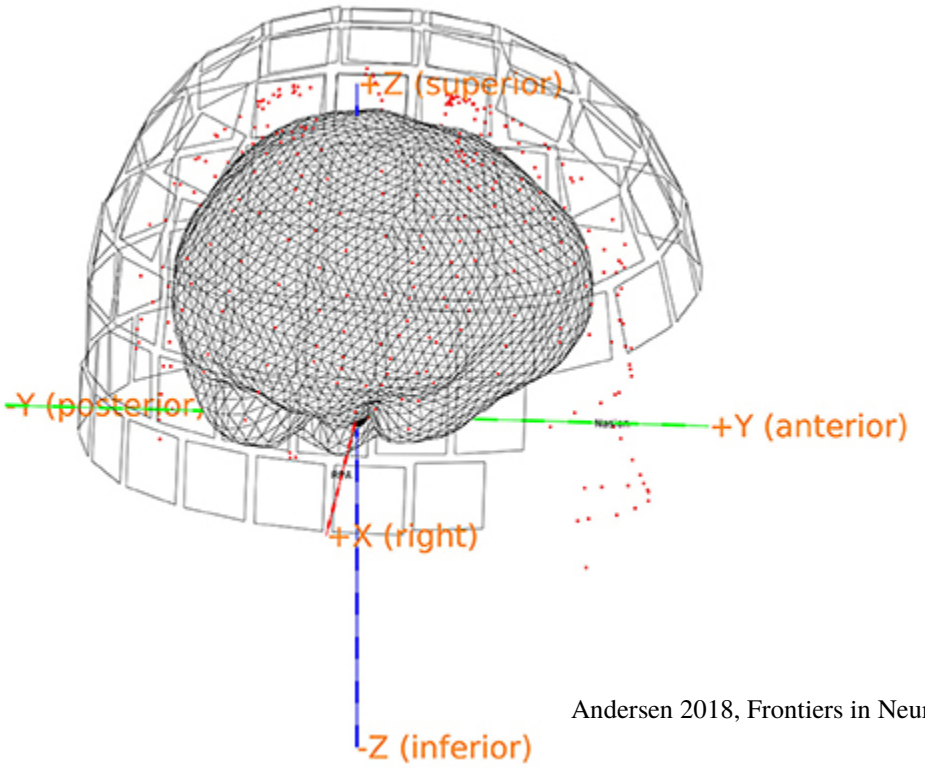




NEW QUESTION

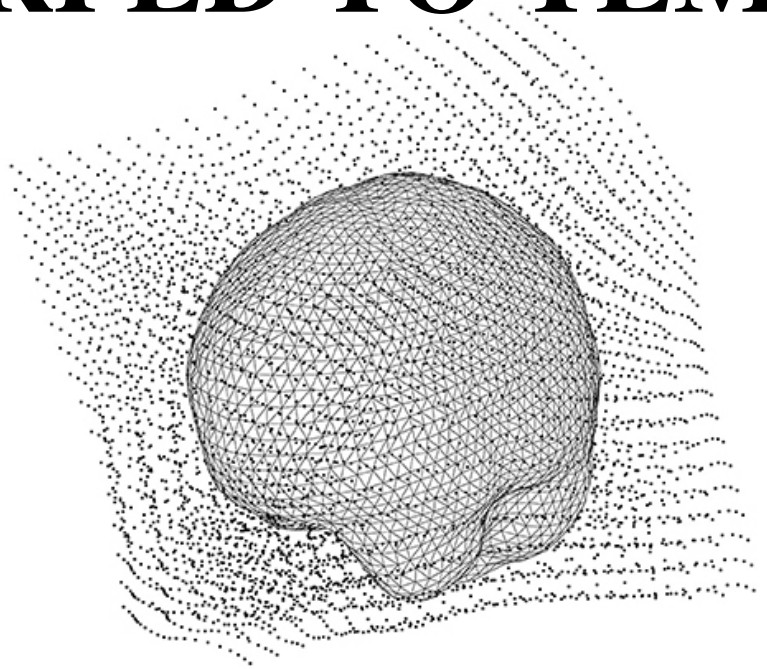
Where does the beta rebound localize to?

OPERATIONS – CO-REGISTRATION



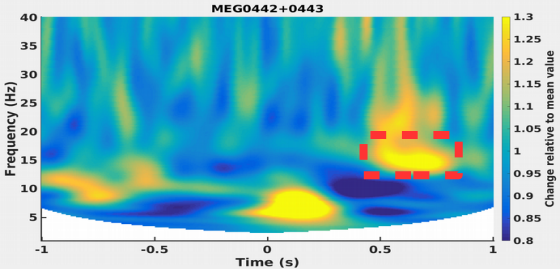
Andersen 2018, Frontiers in Neuroscience

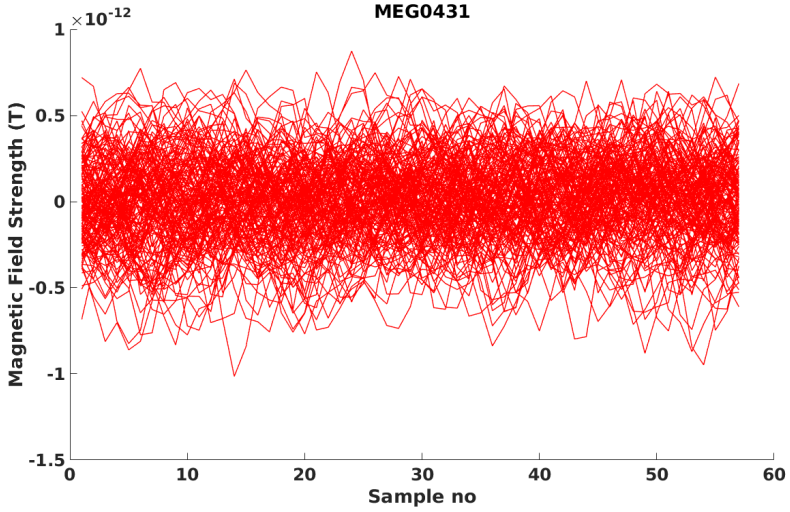
OPERATIONS – LEADFIELD (FORWARD MODEL) WARPED TO TEMPLATE



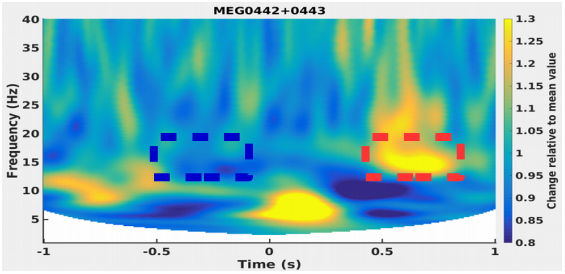
Andersen 2018, Frontiers in Neuroscience

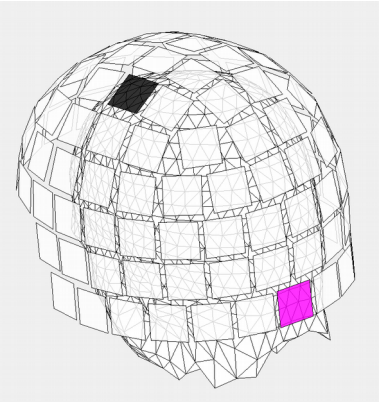
OPERATIONS – SIGNAL TO SOURCE RECONSTRUCT

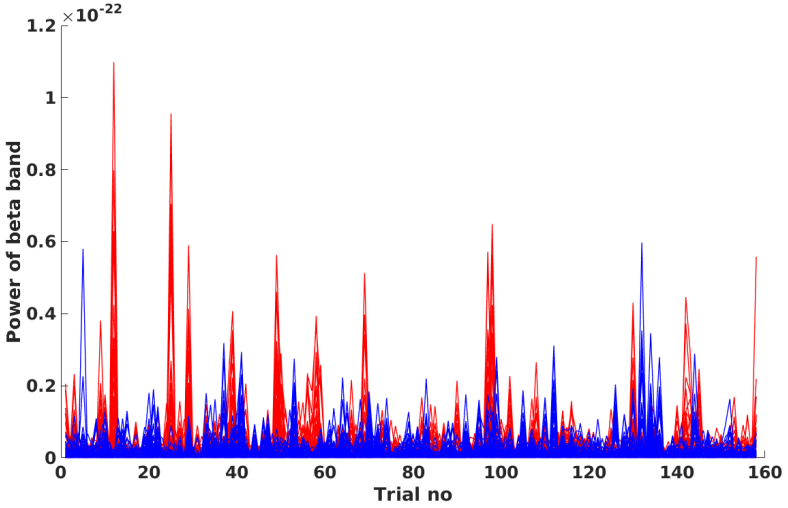


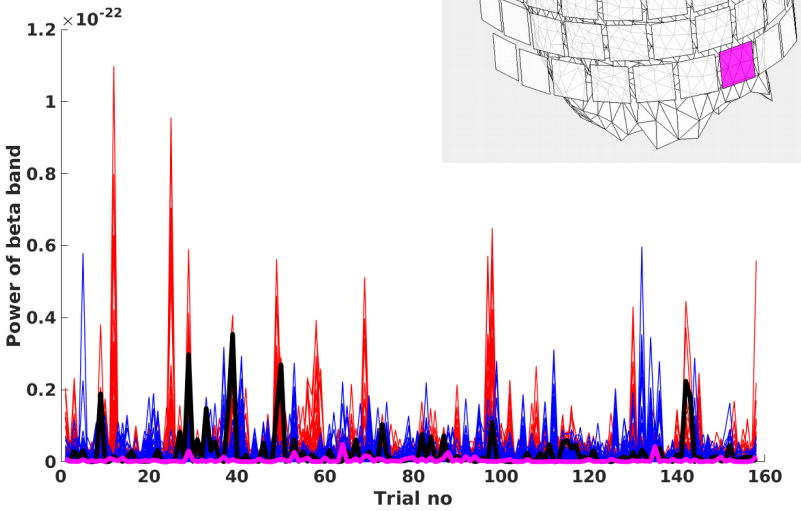


OPERATIONS – SIGNAL TO SOURCE RECONSTRUCT

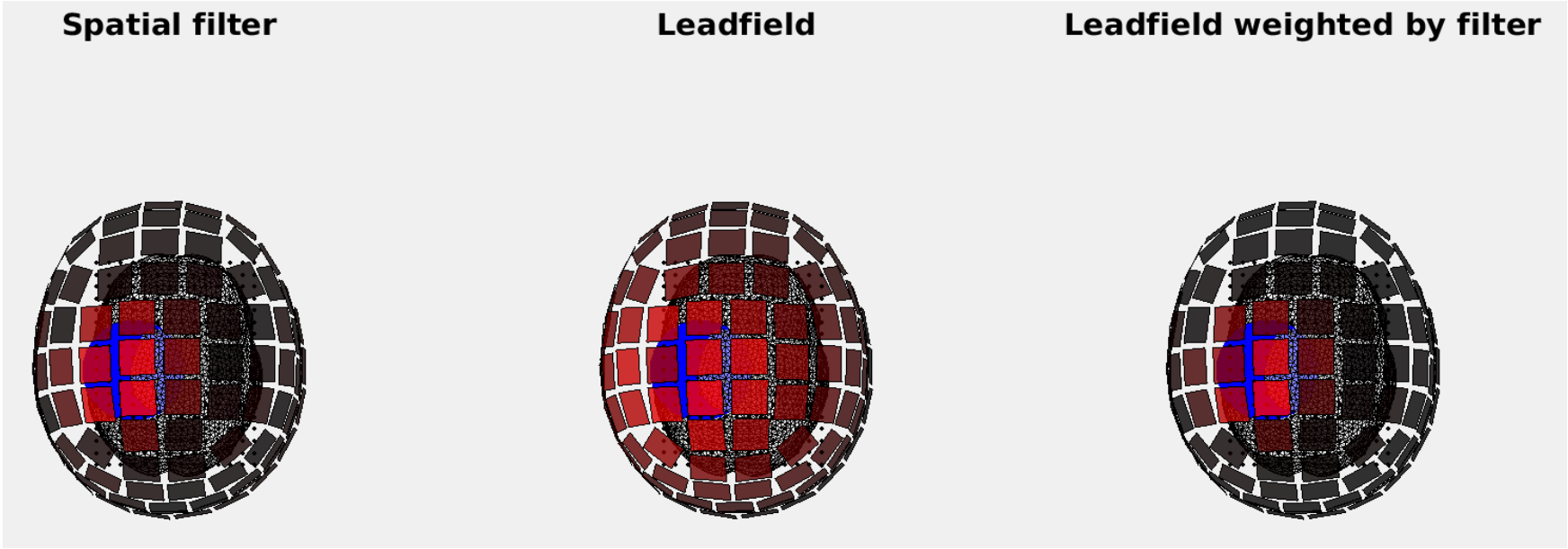




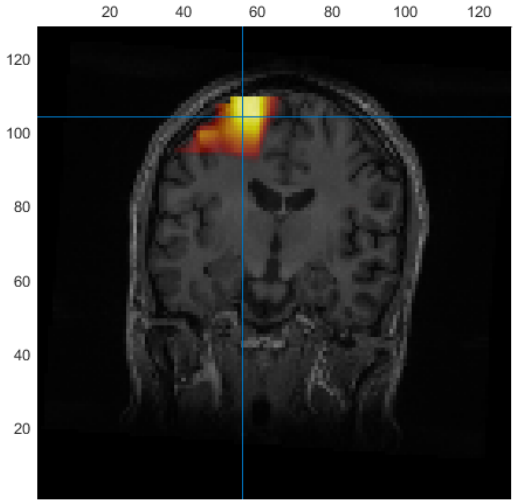


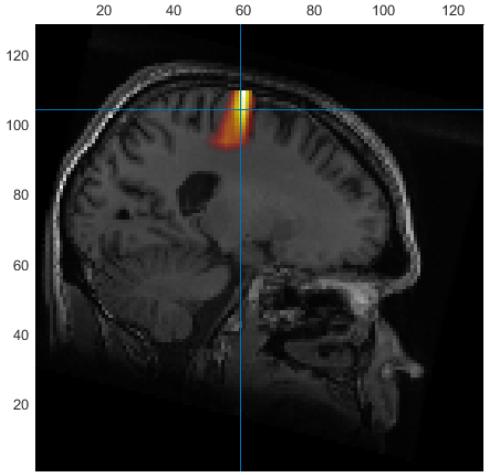


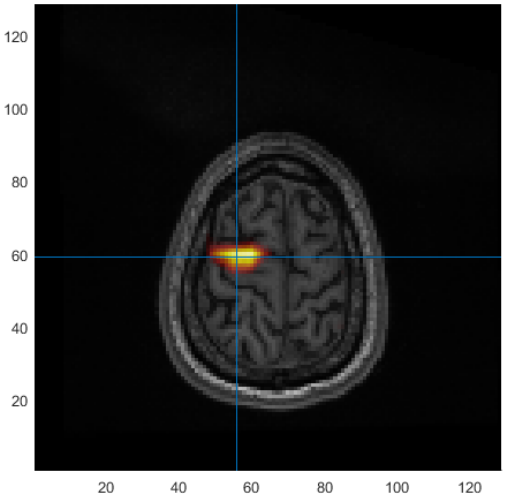
OPERATIONS – LEADFIELD AND SPATIAL FILTER



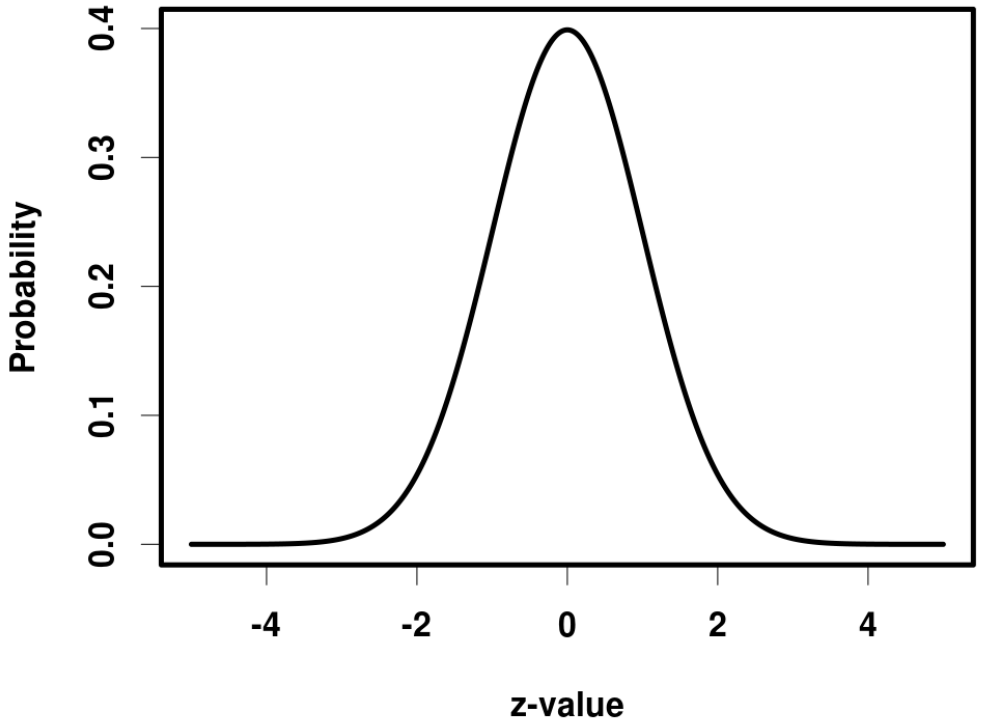
OPERATIONS – BEAMFORMER





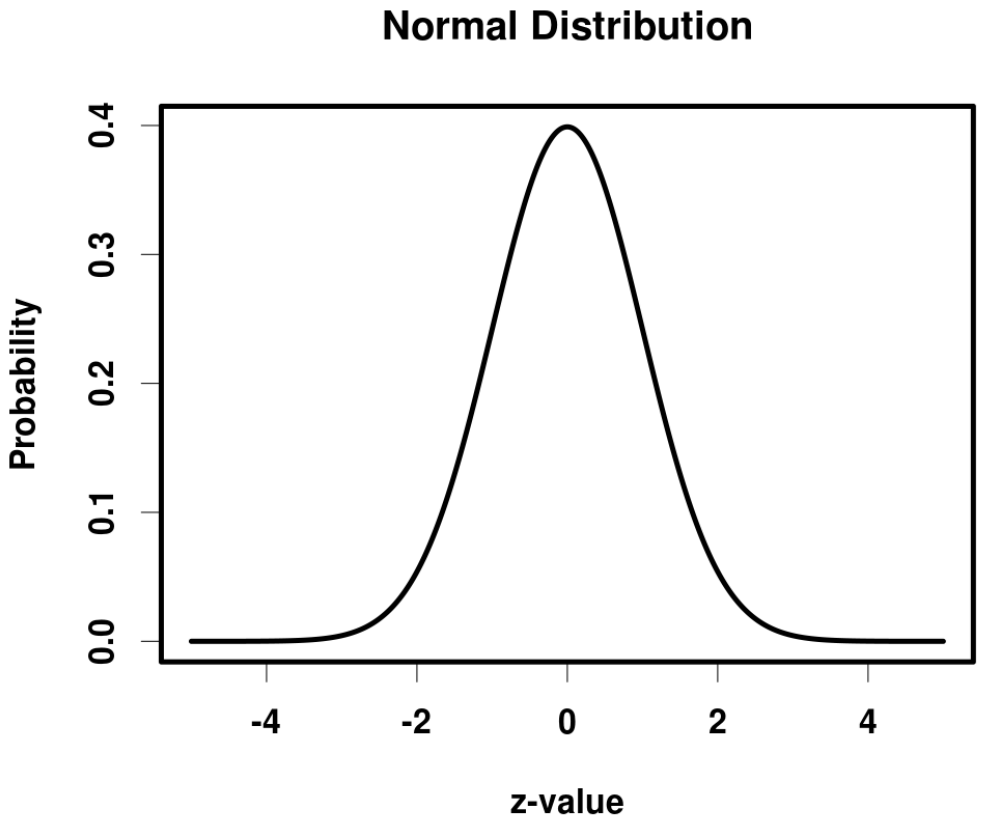


Statistics



What is this?

Statistics

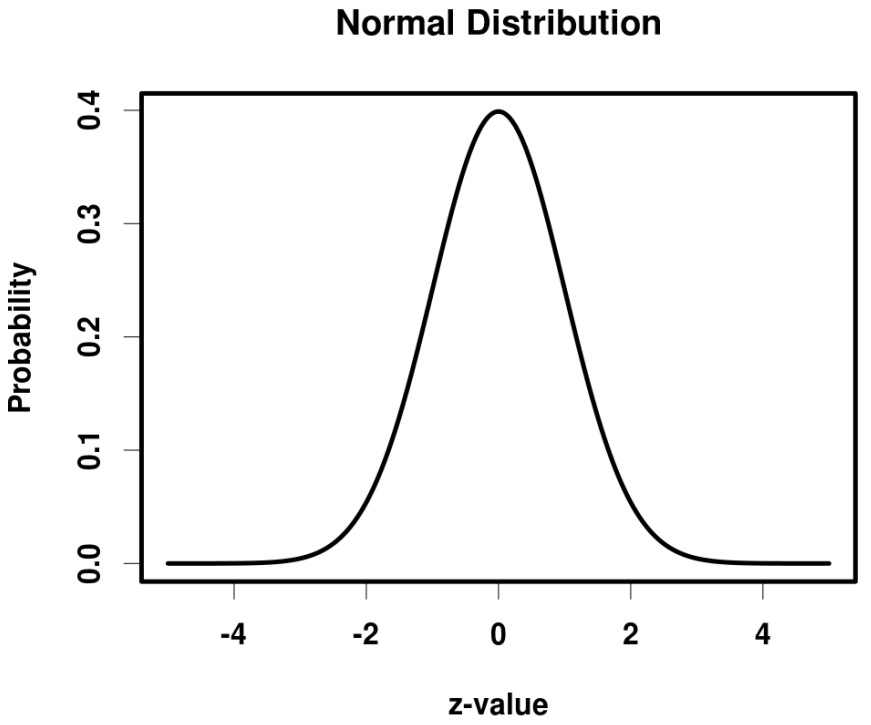

Normal Distribution
Probability
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
z-value

Statistics

$$z = \frac{\bar{X} - P}{SE}$$

$$SE = \frac{\sigma}{\sqrt{n}}$$

$\bar{X}$: *sample mean*
P : *population mean*
SE : *standard error*
σ : *standard deviation*
n : *number of observations*



Statistics

$$z = \frac{\bar{X} - P}{SE}$$

$$SE = \frac{\sigma}{\sqrt{n}}$$

$\bar{X}$: *sample mean*
P : *population mean*
SE : *standard error*
σ : *standard deviation*
n : *number of observations*

If we are testing the null hypothesis, then what is *z*?

Statistics

$$z = \frac{\bar{X} - P}{SE}$$

$$SE = \frac{\sigma}{\sqrt{n}}$$

$\bar{X}$: *sample mean*
P : *population mean*
SE : *standard error*
σ : *standard deviation*
n : *number of observations*

If we are testing the null hypothesis, then what is *z*?

$$P = 0$$

$$z = \frac{\bar{X}}{SE}$$

Statistics

What do we call the probability of observing a *z*-value or a *z*-value that is more extreme?

Statistics

What do we call the probability of observing a *z*-value or a *z*-value that is more extreme?

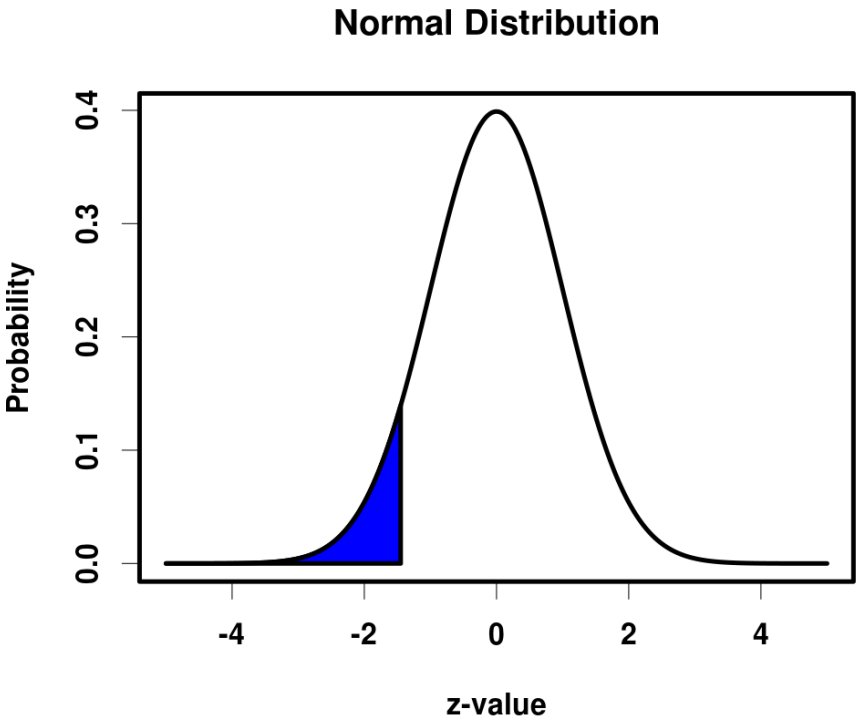
p

Statistics

What do we call the probability of observing a *z*-value or a *z*-value that is more extreme?

p

If *z* = -1.45, what is *p* then?



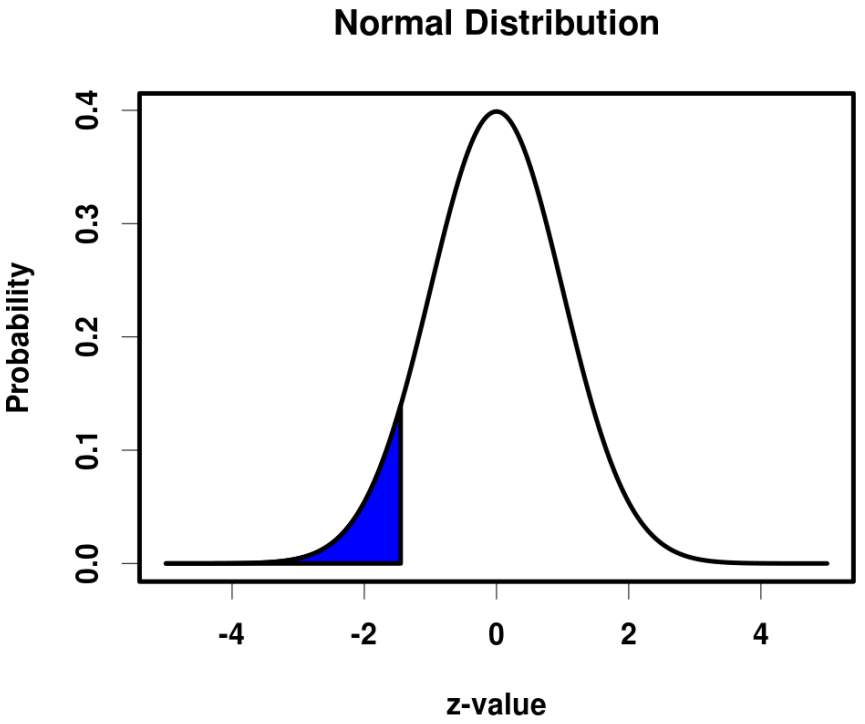
Statistics

What do we call the probability of observing a *z*-value or a *z*-value that is more extreme?

p

If *z* = -1.45, what is *p* then?

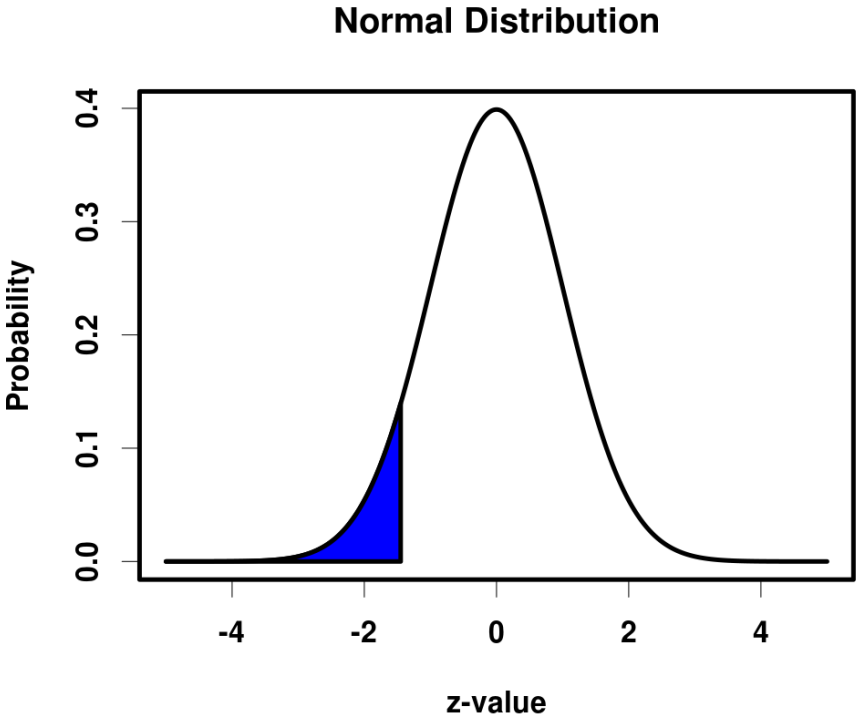
The area of the blue curve, *p* = 0.0735



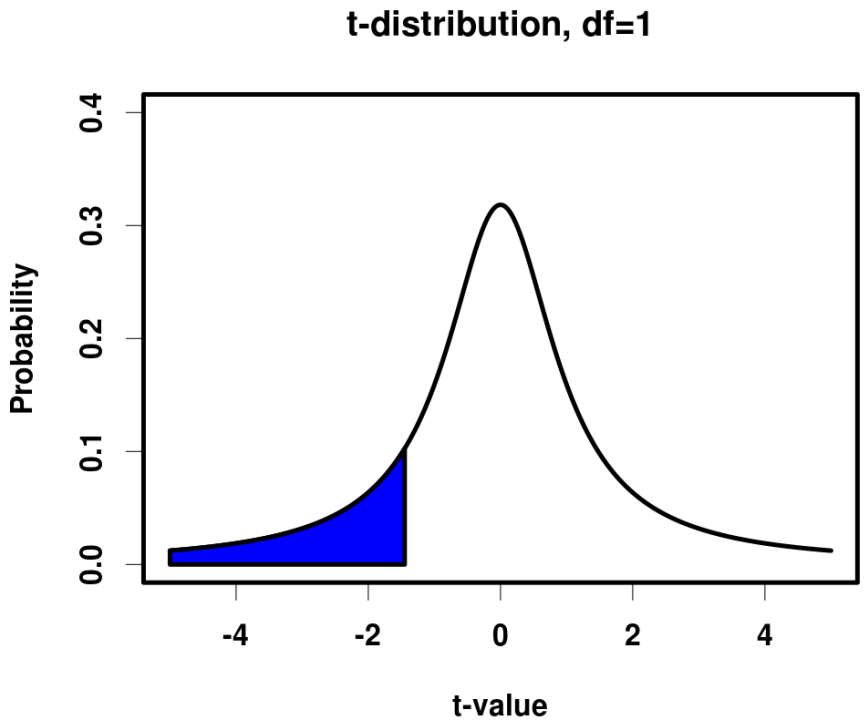
Statistics

Crucial assumption when you are performing a *z*-test is that the population variance is known. When this is not known a *t*-test is more appropriate

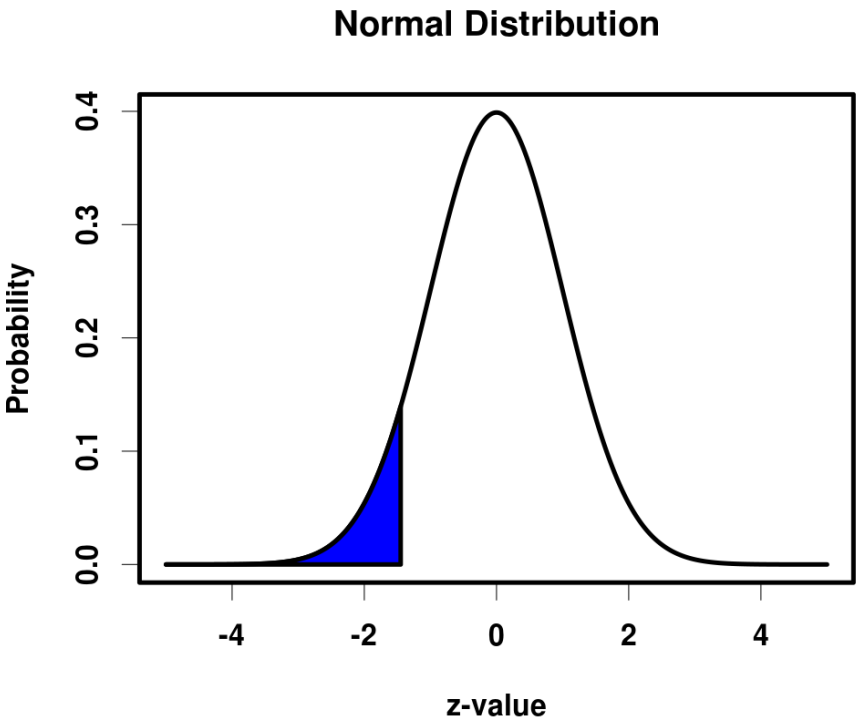
We normally do not know the population variance



Statistics – *t*-distribution

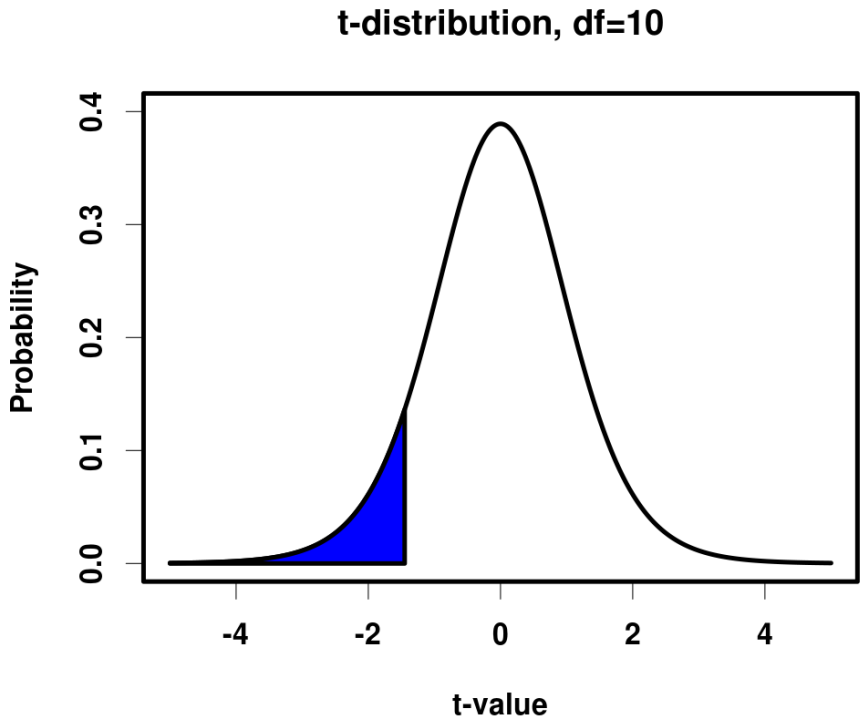


$p = 0.192$

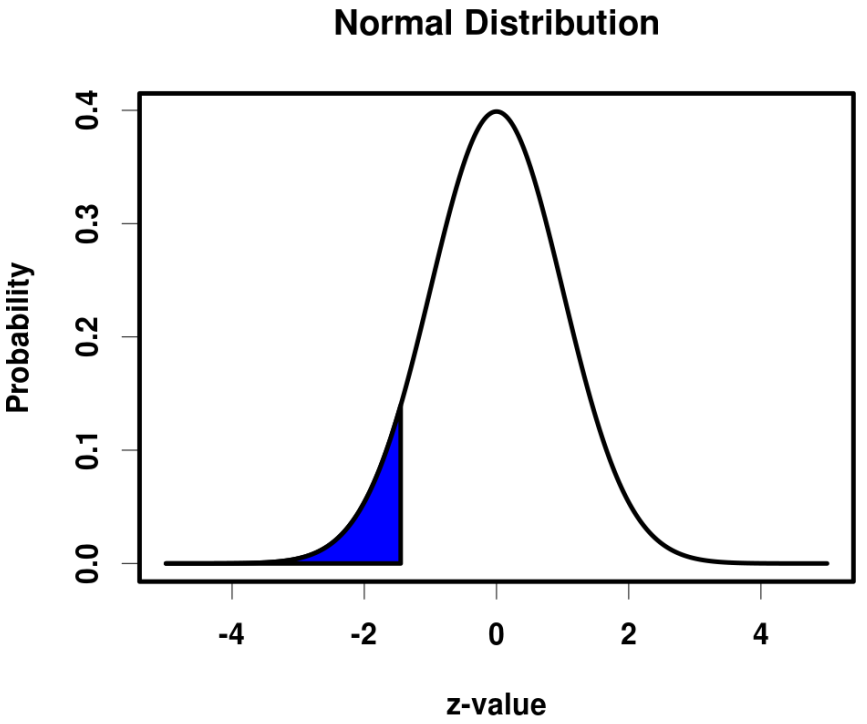


$p = 0.0735$

Statistics – *t*-distribution

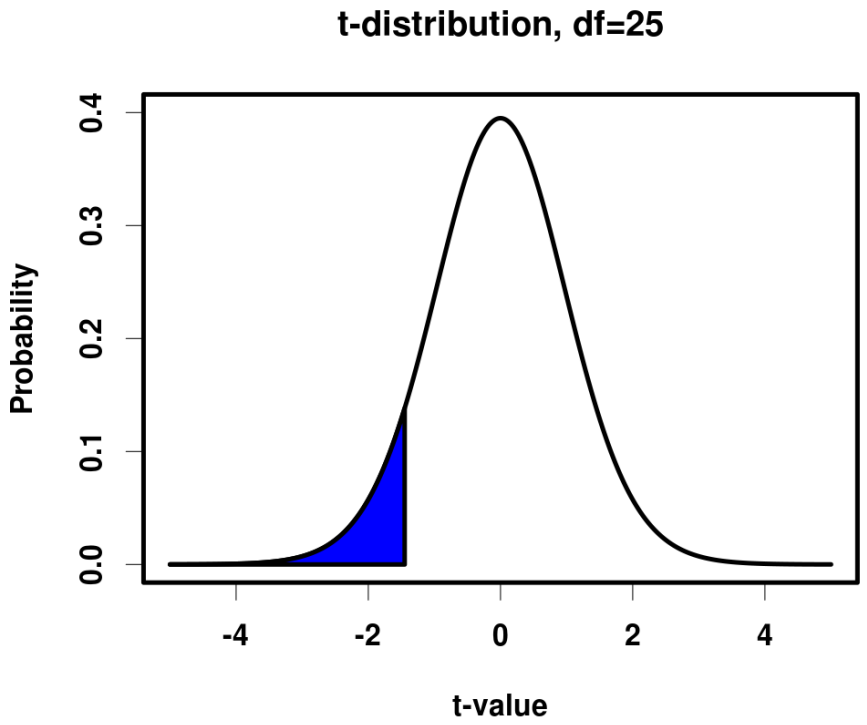


$p = 0.0888$

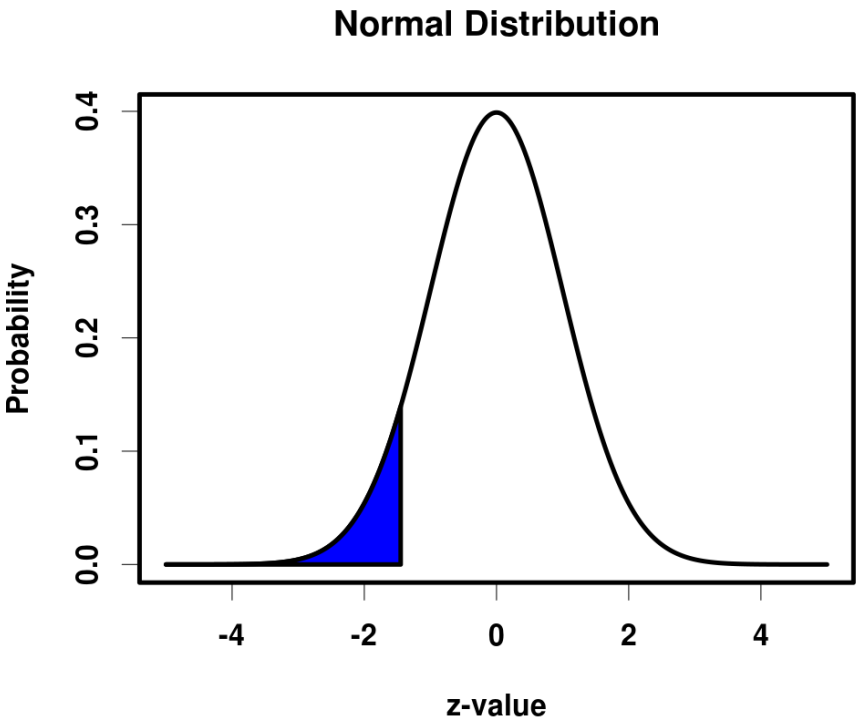


$p = 0.0735$

Statistics – *t*-distribution

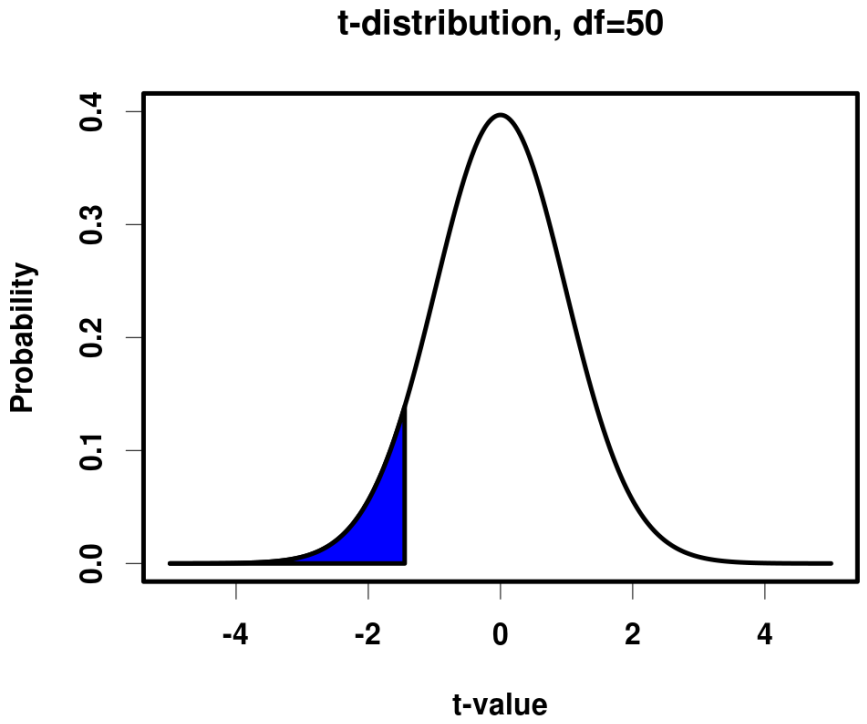


$p = 0.0797$

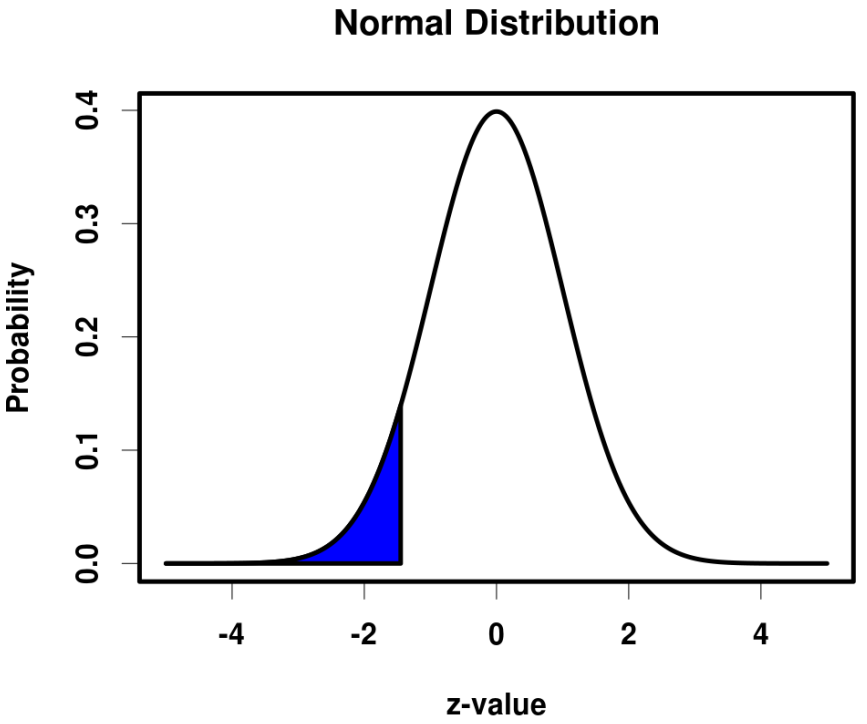


$p = 0.0735$

Statistics – *t*-distribution

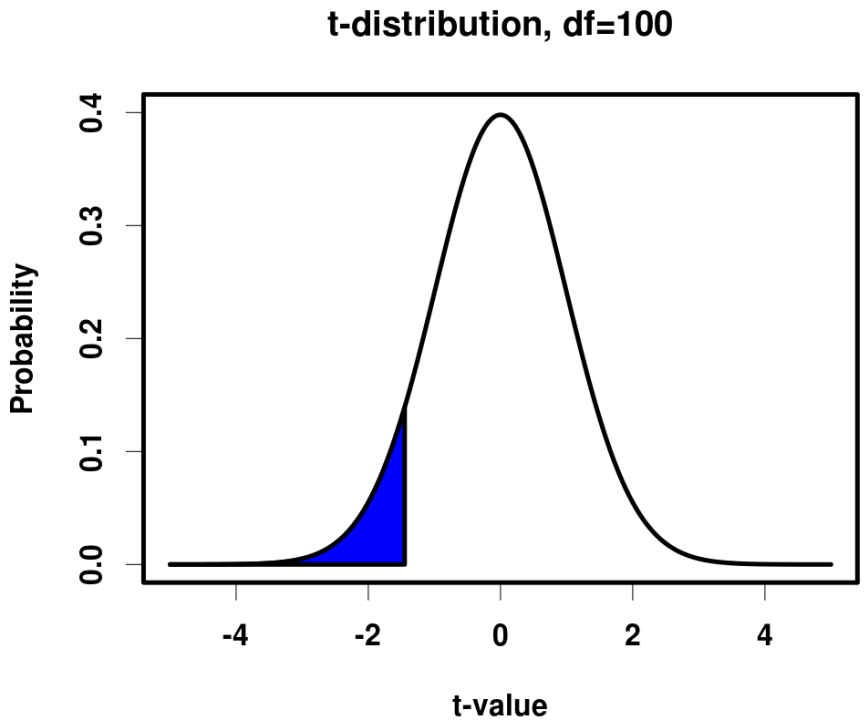


$p = 0.0767$

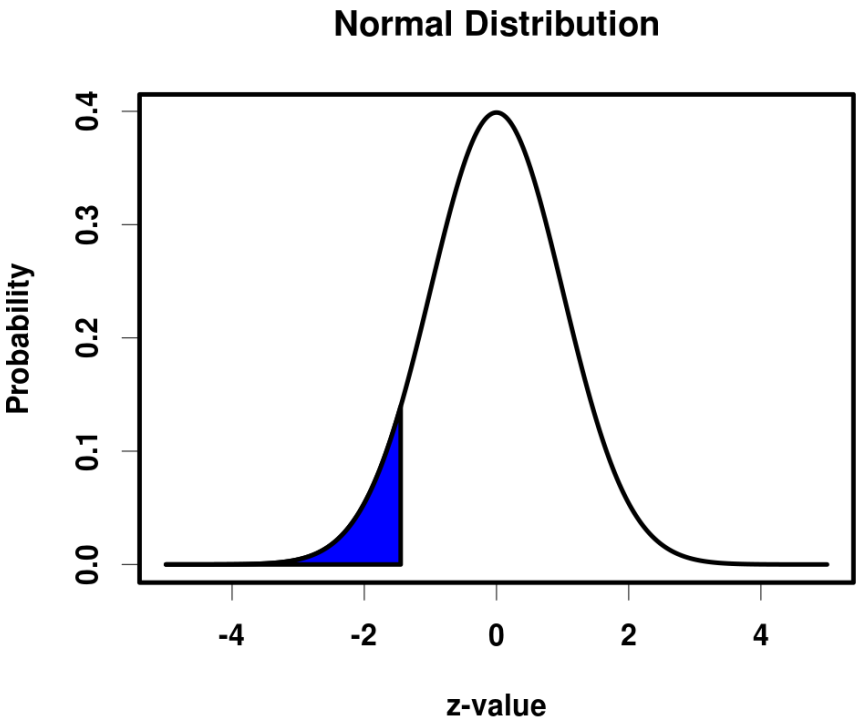


$p = 0.0735$

Statistics – *t*-distribution

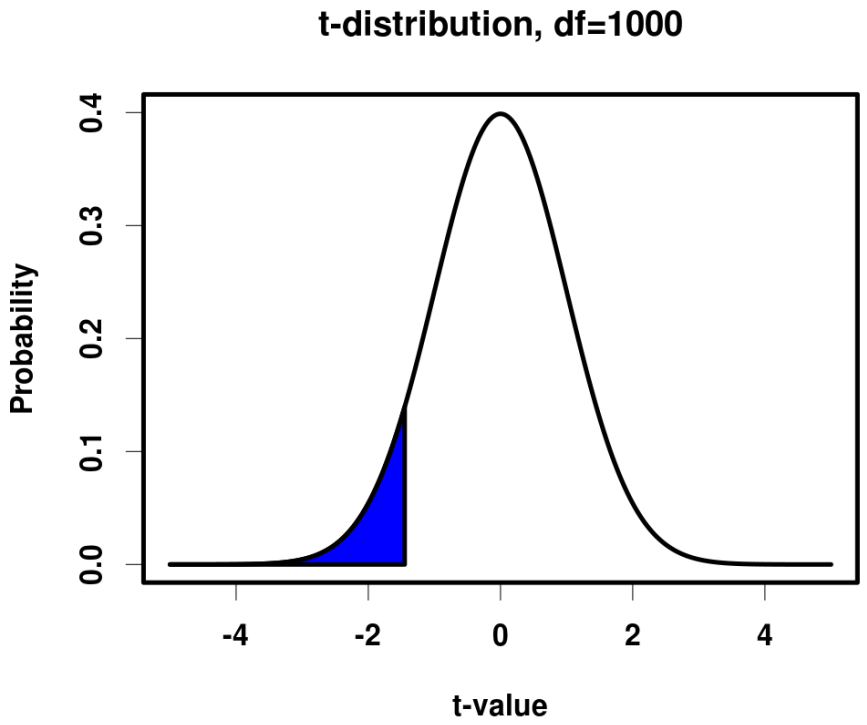


$p = 0.0751$

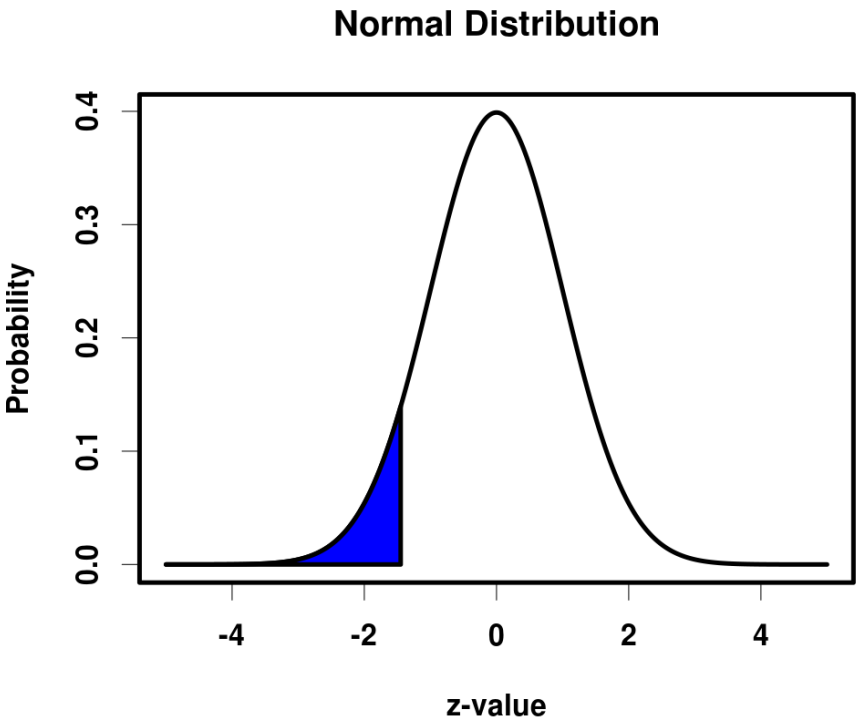


$p = 0.0735$

Statistics – *t*-distribution



$p = 0.0737$



$p = 0.0735$

Two questions

- What does the *t*-distribution converge to when *df* goes to infinity?
- How are the *t*-value and the *z*-value related when doing null hypothesis testing?

Two questions

- What does the *t*-distribution converge to when *df* goes to infinity?

  The *z*-distribution

- How are the *t*-value and the *z*-value related when doing null hypothesis testing?

  $t = z$

Two questions

- What does the *t*-distribution converge to when *df* goes to infinity?

  The *z*-distribution

- How are the *t*-value and the *z*-value related when doing null hypothesis testing?

$$t = z = \frac{\bar{X}}{SE}$$

Two questions

- What does the *t*-distribution converge to when *df* goes to infinity?

  The *z*-distribution

- How are the *t*-value and the *z*-value related when doing null hypothesis testing?

$$t = z = \frac{\bar{X}}{SE}$$

Thus, *t*-values are independent of *df*. It is the *p*-value associated with the *t*-value that is dependent on *df*

α

α signifies the cut-off point for *p*-values, which is used to decide whether to reject the null hypothesis and accept the alternative hypothesis, or whether to conclude that the null hypothesis could not be rejected

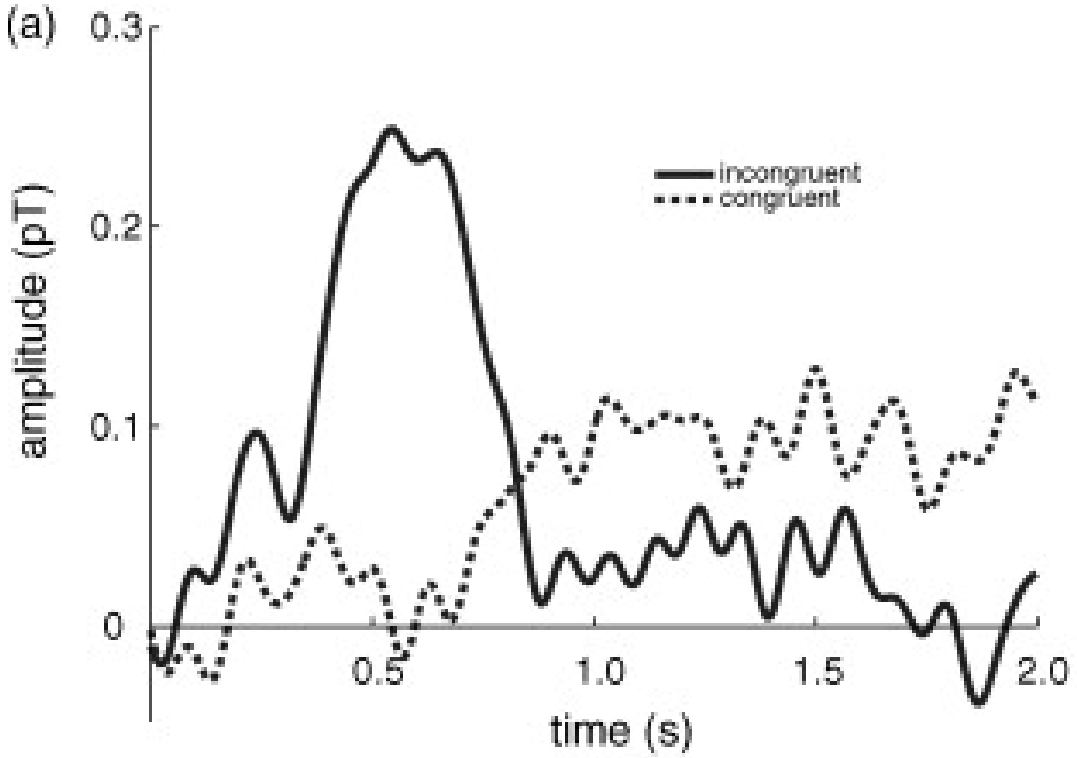
α also signifies the risk of making a Type-I error, i.e. the error of accepting a false positive, i.e. rejecting the null hypothesis when it is actually true, every time a test is made

α is canonically set at 0.05 for reasons unbeknownst to man

Statistics – *t*-distribution

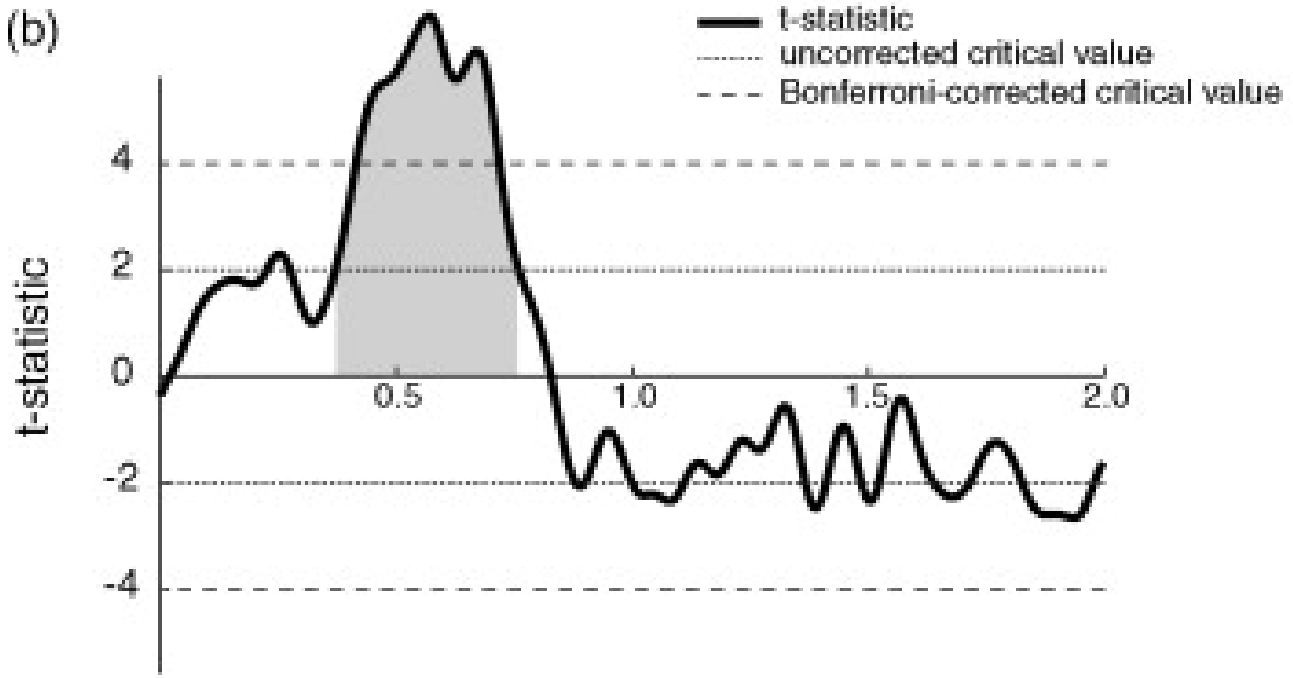
The *t*-test works fine for many things, for example testing the quality of beer batches (Guinness), which it is what is was originally developed for

Evoked example



Maris & Oostenveld, 2007

Evoked example

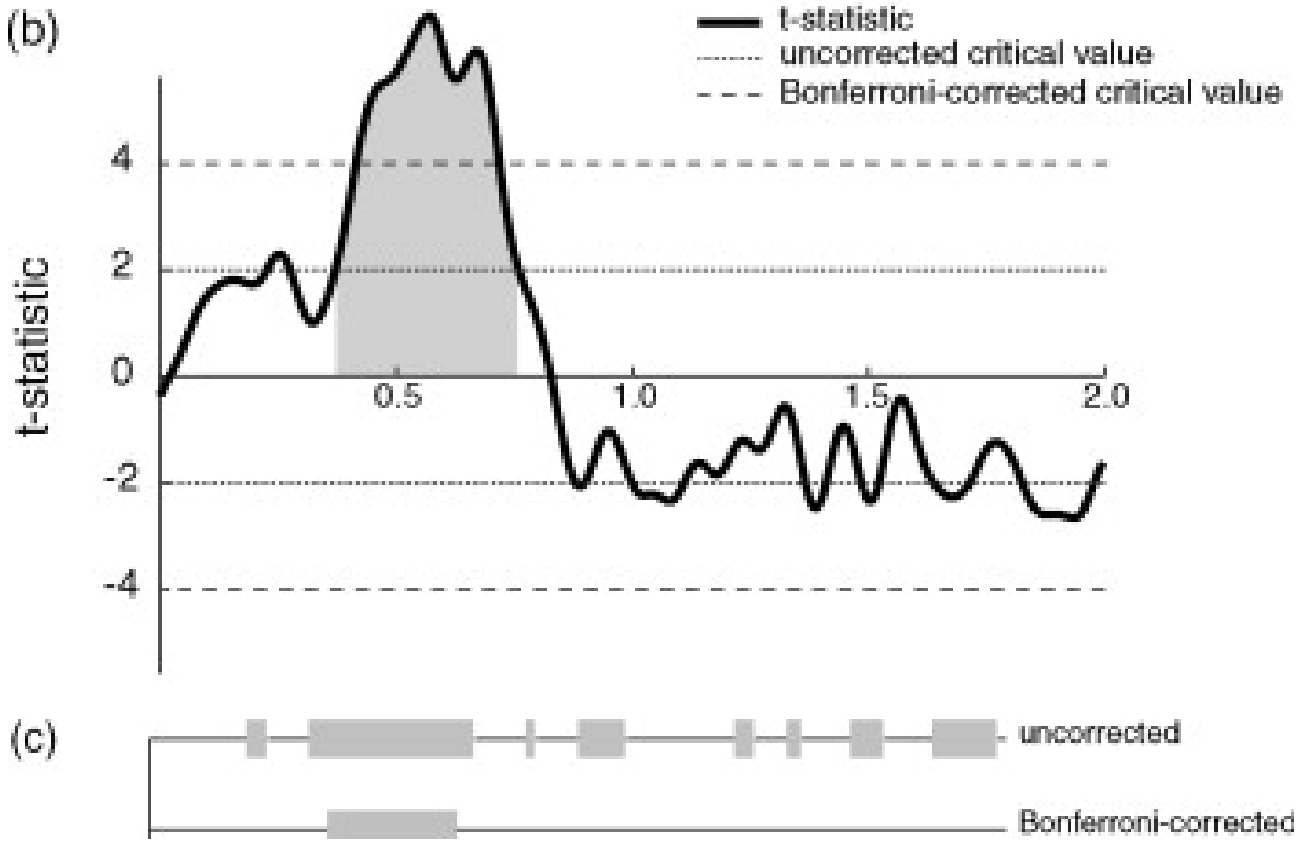


α also signifies the risk of making a Type-I error, i.e. the error of accepting a false positive, i.e. rejecting the null hypothesis when it is actually true, every time a test is made

This leads to the Multiple Comparisons Problem

Maris & Oostenveld, 2007

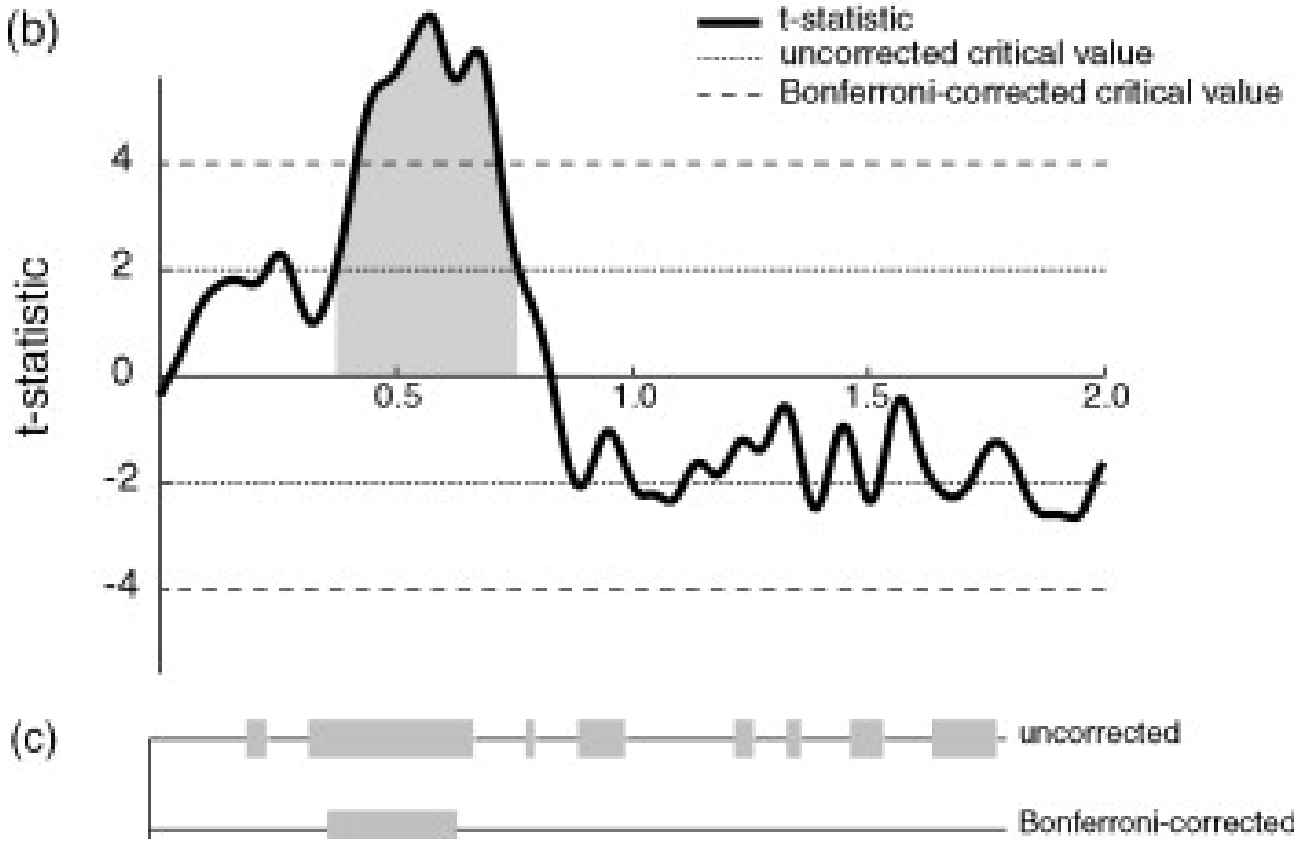
Evoked example



$$\alpha_{BONF} = \frac{\alpha}{n\,tests}$$

Maris & Oostenveld, 2007

Evoked example



$$\alpha_{BONF} = \frac{\alpha}{n\,tests}$$

Goes well enough with one sensor

Maris & Oostenveld, 2007

What about when we have 306 MEG channels and 128 electrodes each sampled 1000 times per second?

What about when we have 306 MEG channels and 128 electrodes each sampled 1000 times per second?

That amounts to 434,000 possible tests per second

Uncorrected testing will lead to way too many false positives

Bonferroni-corrected test will lead to no true positives

Statistics – *t*-distribution

Can anyone name an assumption of the *t*-test that makes it less than optimal for MEEG data?

Statistics – *t*-distribution

Can anyone name an assumption of the *t*-test that makes it less feasible for MEEG data?

It requires that tested samples are independent from one another, which MEEG samples are evidently not. (MEEG data show strong spatio-temporal correlations)

Permutation tests

They don’t have strong assumptions about the underlying structure of the data

ELSEVIER

Journal of Neuroscience Methods 164 (2007) 177–190

JOURNAL OF NEUROSCIENCE METHODS

www.elsevier.com/locate/jneumeth

Nonparametric statistical testing of EEG- and MEG-data☆,☆☆

Eric Maris[a,b,*], Robert Oostenveld[b]

[a] *NICI, Biological Psychology, Radboud University Nijmegen, Nijmegen, The Netherlands*

[b] *F.C. Donders Center for Cognitive Neuroimaging, Radboud University Nijmegen, Nijmegen, The Netherlands*

Received 7 January 2007; received in revised form 19 March 2007; accepted 29 March 2007

Abstract

In this paper, we show how ElectroEncephaloGraphic (EEG) and MagnetoEncephaloGraphic (MEG) data can be analyzed statistically using nonparametric techniques. Nonparametric statistical tests offer complete freedom to the user with respect to the test statistic by means of which the experimental conditions are compared. This freedom provides a straightforward way to solve the multiple comparisons problem (MCP) and it allows to incorporate biophysically motivated constraints in the test statistic, which may drastically increase the sensitivity of the statistical test. The paper is written for two audiences: (1) empirical neuroscientists looking for the most appropriate data analysis method, and (2) methodologists interested in the theoretical concepts behind nonparametric statistical tests. For the empirical neuroscientist, a large part of the paper is written in a tutorial-like fashion, enabling neuroscientists to construct their own statistical test, maximizing the sensitivity to the expected effect. And for the methodologist, it is explained why the nonparametric test is formally correct. This means that we formulate a null hypothesis (identical probability distribution in the different experimental conditions) and show that the nonparametric test controls the false alarm rate under this null hypothesis.

Keywords: Nonparametric statistical testing; Hypothesis testing; EEG; MEG; Multiple comparisons problem

Overview of logic

1) Create a design matrix indicating how data are organized and what conditions should be compared against one another
2) Do a *t*-tests on all samples of interest
3) Find the *T*-values for each cluster of *t*-values passing the analytic significance test
4) Permute the condition labels (between 1000-10000 times) using a random mechanism and find the *T*-values for each of the permutations.
5) Find the distribution of permutation-based *T*-values
6) Compare the *T*-value based on the design matrix with the permutation distribution and use the *p*-value together with a pre-set alpha to decide whether or not to reject the null hypothesis

Design matrix

Unit variable	Independent variable
1	1
2	1
...	...
19	1
20	1
1	2
2	2
...	...
19	2
20	2

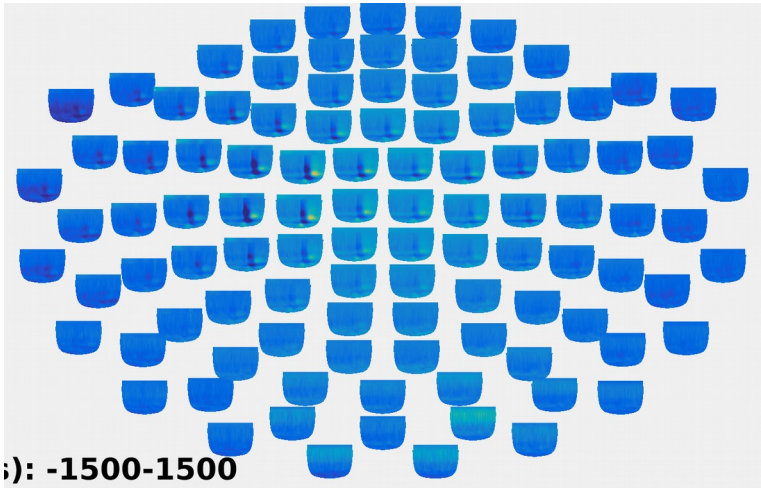
```
cfg = [];
cfg.latency = [1.500 1.500];
cfg.frequency = [1 100];
cfg.parameter = 'powspctrm';
cfg.method = 'analytic'; %% do a mass-univariate
test
cfg.alpha = 0.05; %% critical value around ± 2.09
cfg.statistic = 'depsamplesT'; % use a dependent
samples t-test
cfg.design(1, :) = [1:n_subjects 1:n_subjects];
cfg.design(2, :) = [ones(1, n_subjects) 2 * ones(1,
n_subjects)];
cfg.uvar = 1; % first row of cfg.design, containing
the (u)nits (subjects)
cfg.ivar = 2; % second row of cfg.design, the
(i)ndependent events (3&15)
stat = ft_freqstatistics(cfg, dataset_1, dataset_2);
```

Dependent variable is the frequency data

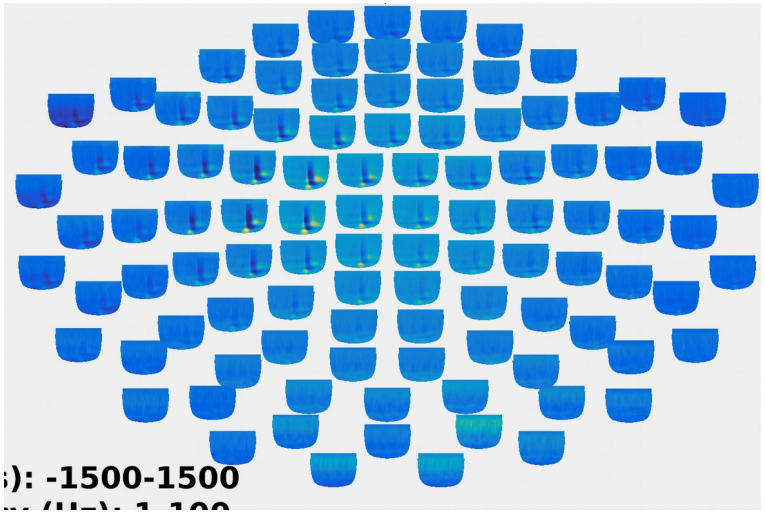
Design matrix

Unit variable	Independent variable
1	1
2	1
...	...
19	1
20	1
1	2
2	2
...	...
19	2
20	2

Dependent variable is the frequency data

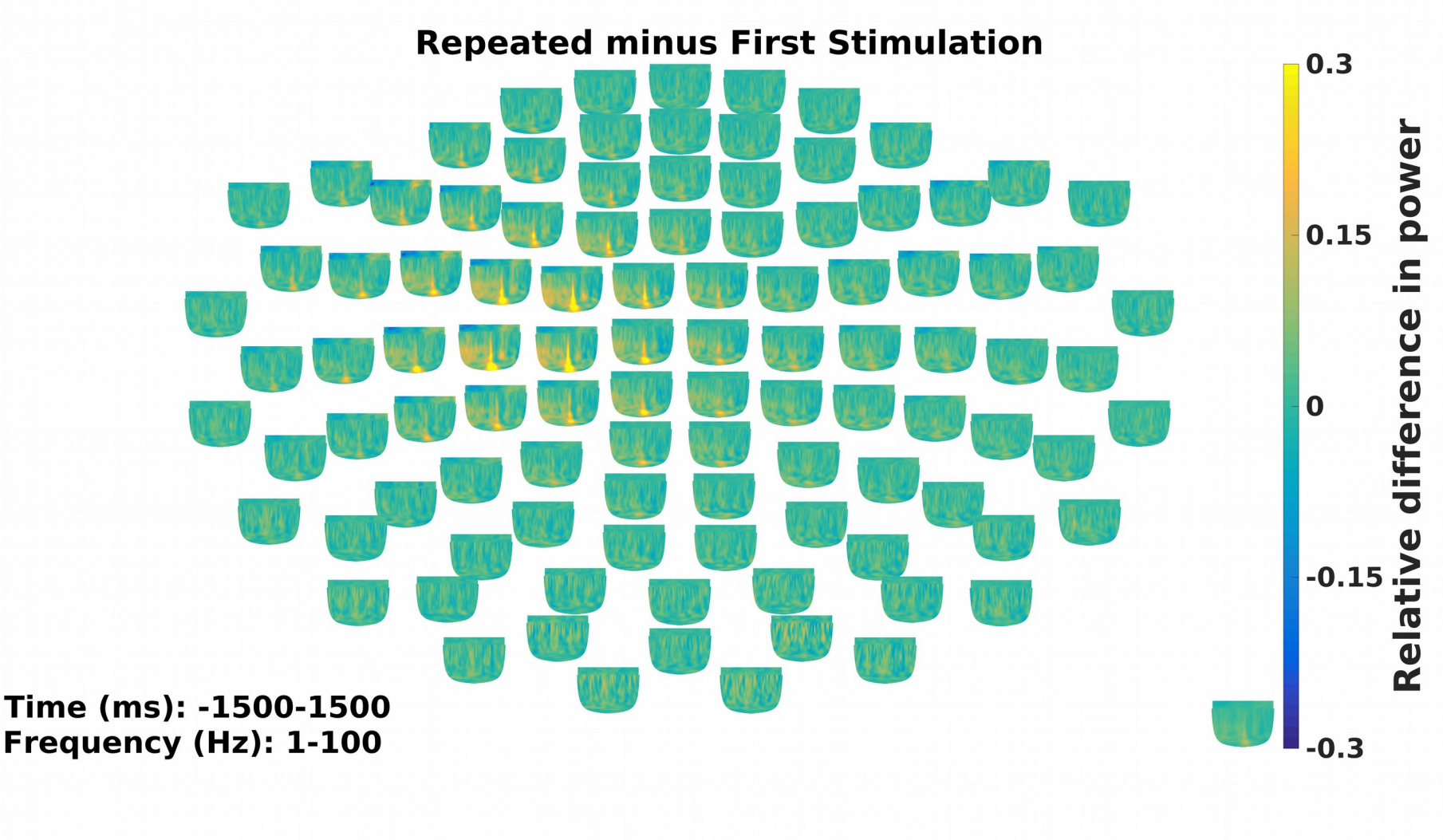


First Stimulation

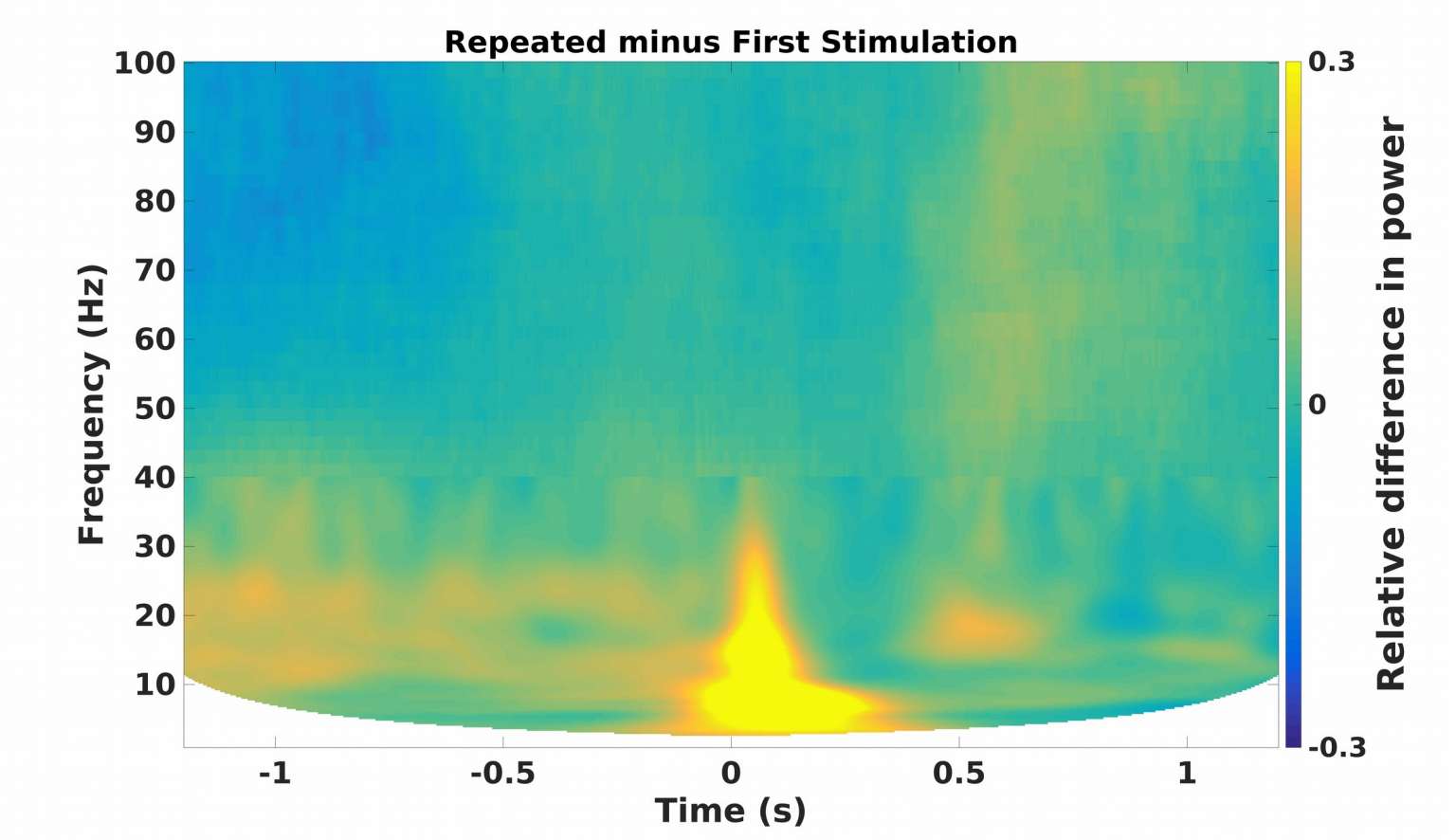


Repeated Stimulation

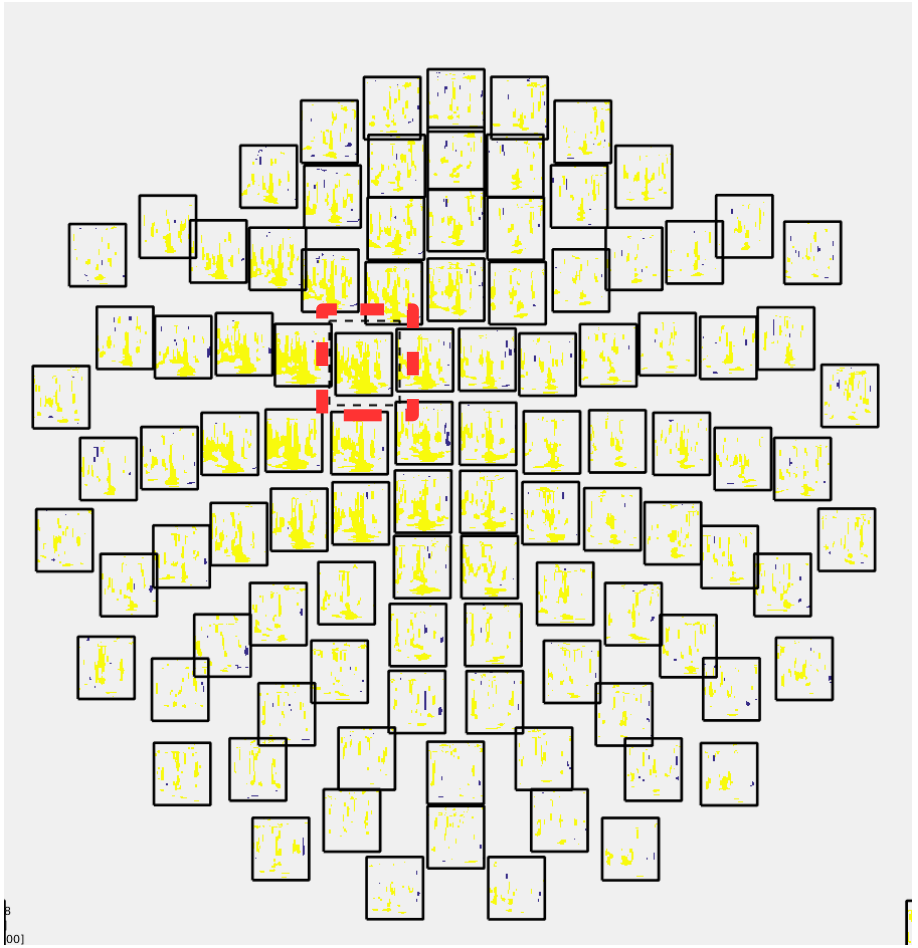
Difference map

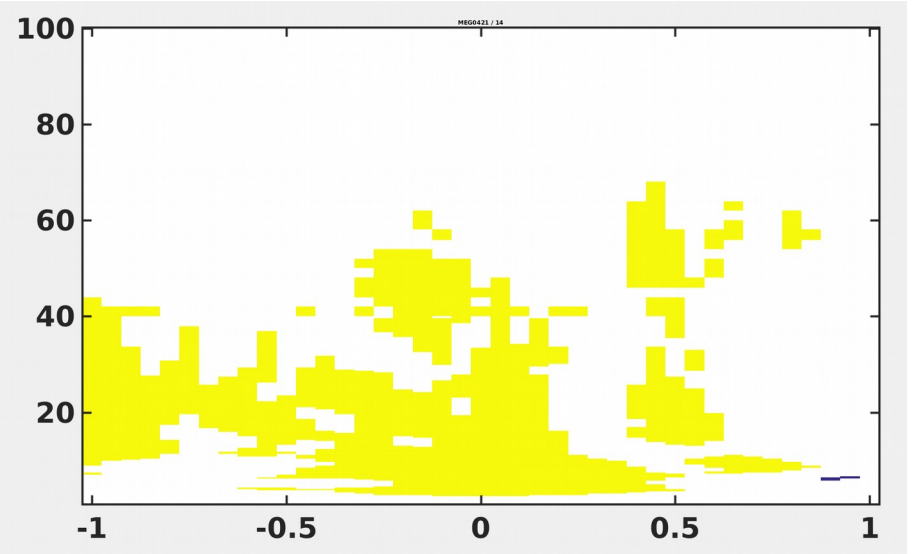


Difference-map – single channel



t-value-map – masked





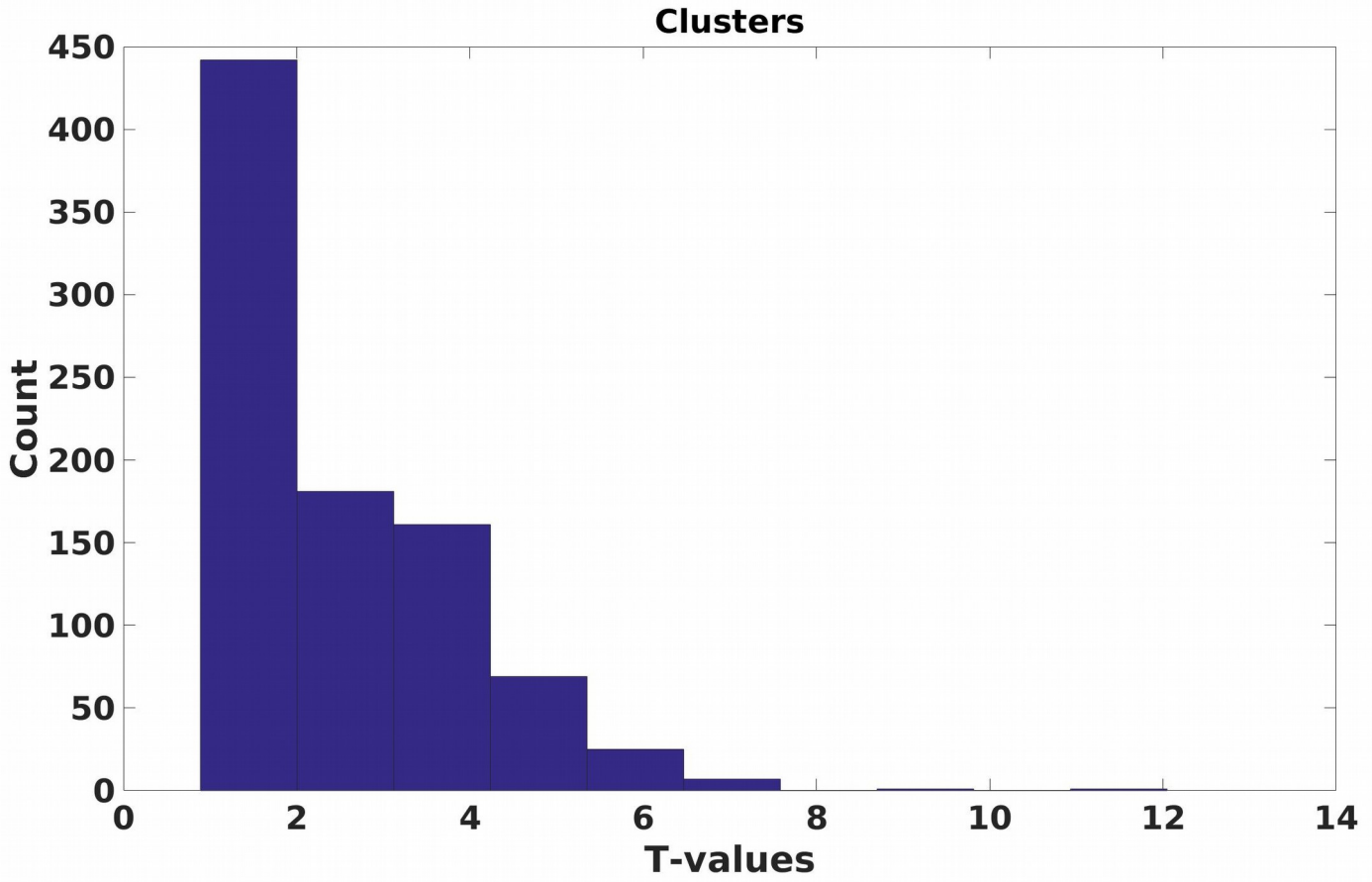
Cluster analysis

Unit variable	Independent variable
1	1
2	1
...	...
19	1
20	1
1	2
2	2
...	...
19	2
20	2

Dependent variable is the frequency data

```
cfg = [];
cfg.latency = [1.500 1.500];
cfg.frequency = [1 100];
cfg.parameter = 'powspctrm';
cfg.method = 'montecarlo'; %% do a permutation test
cfg.alpha = 0.05; %% critical value around ± 2.09
cfg.statistic = 'depsamplesT'; % use a dependent
samples t-test
cfg.design(1, :) = [1:n_subjects 1:n_subjects];
cfg.design(2, :) = [ones(1, n_subjects) 2 * ones(1,
n_subjects)];
cfg.uvar = 1; % first row of cfg.design, containing
the (u)nits (subjects)
cfg.ivar = 2; % second row of cfg.design, the
(i)ndependent events

cfg.correctm = 'cluster'; % correct with cluster
method
cfg.clusteralpha = 0.025; % alpha for clusters
cfg.clusterstatistic = 'maxsum'; % way to calculate
T
cfg.clustertail = 0;
cfg.neighbours = neighbours; % a neighbour
structure, see ft_prepare_neighbours
cfg.numrandomization = 2000; % n randomizations

stat = ft_freqstatistics(cfg, dataset_1, dataset_2);
```

(Log)-*T*-values



Now we have some *T*-values. But we do not have a distribution to compare it against, like the normal distribution or a *t*-distribution

Cluster analysis – permutation 1

Unit variable	Independent variable
1	1
2	2
...	...
19	2
20	1
1	2
2	1
...	...
19	1
20	2

Dependent variable is the frequency data

```
cfg = [];
cfg.latency = [1.500 1.500];
cfg.frequency = [1 100];
cfg.parameter = 'powspctrm';
cfg.method = 'montecarlo'; %% do a permutation test
cfg.alpha = 0.05; %% critical value around ± 2.09
cfg.statistic = 'depsamplesT'; % use a dependent
samples t-test
cfg.design(1, :) = [1:n_subjects 1:n_subjects];
cfg.design(2, :) = [ones(1, n_subjects) 2 * ones(1,
n_subjects)];
cfg.uvar = 1; % first row of cfg.design, containing
the (u)nits (subjects)
cfg.ivar = 2; % second row of cfg.design, the
(i)ndependent events

cfg.correctm = 'cluster'; % correct with cluster
method
cfg.clusteralpha = 0.025; % alpha for clusters
cfg.clusterstatistic = 'maxsum'; % way to calculate
T
cfg.clustertail = 0;
cfg.neighbours = neighbours; % a neighbour
structure, see ft_prepare_neighbours
cfg.numrandomization = 2000; % n randomizations

stat = ft_freqstatistics(cfg, dataset_1, dataset_2);
```

Cluster analysis – permutation 2

Unit variable	Independent variable
1	2
2	1
...	...
19	1
20	2
1	1
2	2
...	...
19	2
20	1

Dependent variable is the frequency data

```
cfg = [];
cfg.latency = [1.500 1.500];
cfg.frequency = [1 100];
cfg.parameter = 'powspctrm';
cfg.method = 'montecarlo'; %% do a permutation test
cfg.alpha = 0.05; %% critical value around ± 2.09
cfg.statistic = 'depsamplesT'; % use a dependent
samples t-test
cfg.design(1, :) = [1:n_subjects 1:n_subjects];
cfg.design(2, :) = [ones(1, n_subjects) 2 * ones(1,
n_subjects)];
cfg.uvar = 1; % first row of cfg.design, containing
the (u)nits (subjects)
cfg.ivar = 2; % second row of cfg.design, the
(i)ndependent events

cfg.correctm = 'cluster'; % correct with cluster
method
cfg.clusteralpha = 0.025; % alpha for clusters
cfg.clusterstatistic = 'maxsum'; % way to calculate
T
cfg.clustertail = 0;
cfg.neighbours = neighbours; % a neighbour
structure, see ft_prepare_neighbours
cfg.numrandomization = 2000; % n randomizations

stat = ft_freqstatistics(cfg, dataset_1, dataset_2);
```

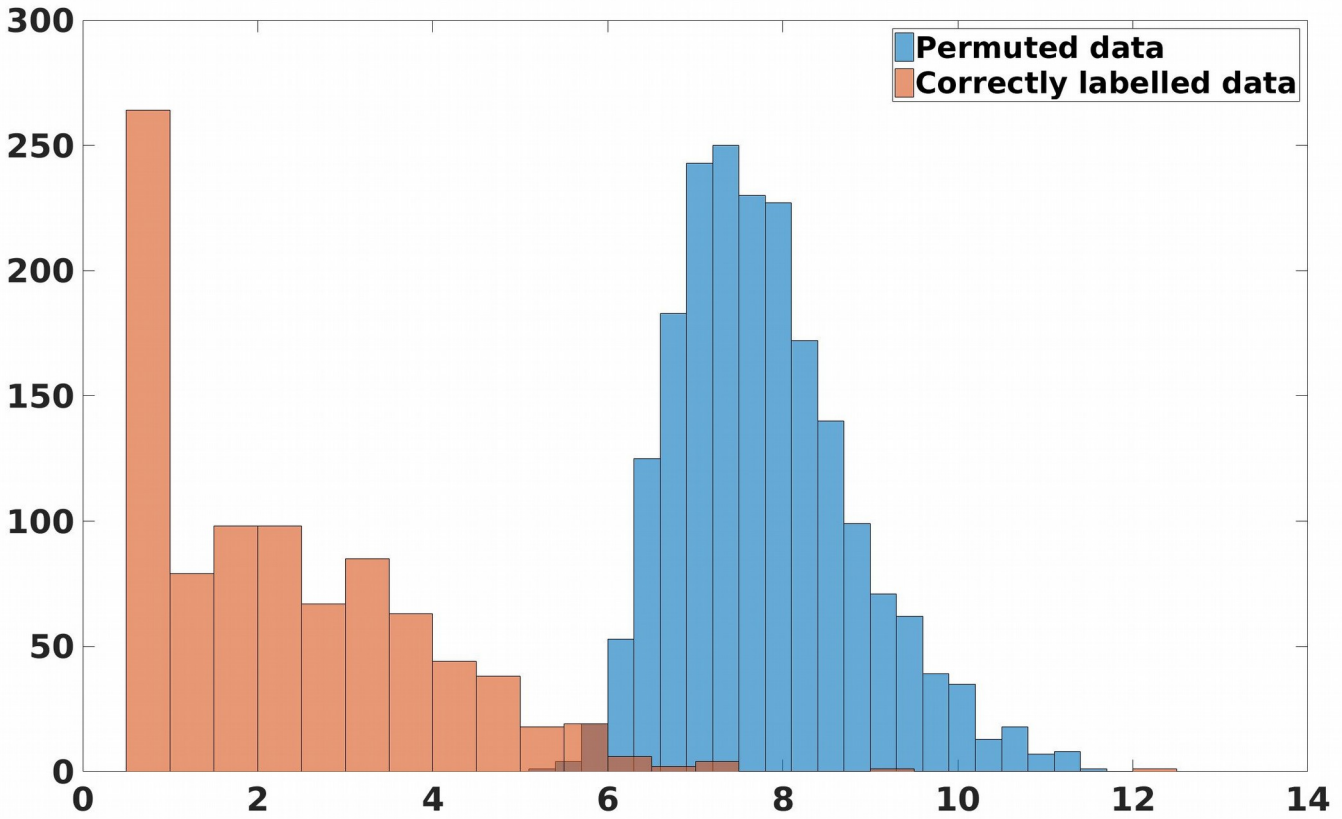
Cluster analysis – permutation ... 2000

Unit variable	Independent variable
1	1
2	2
...	...
19	1
20	1
1	2
2	1
...	...
19	2
20	2

Dependent variable is the frequency data

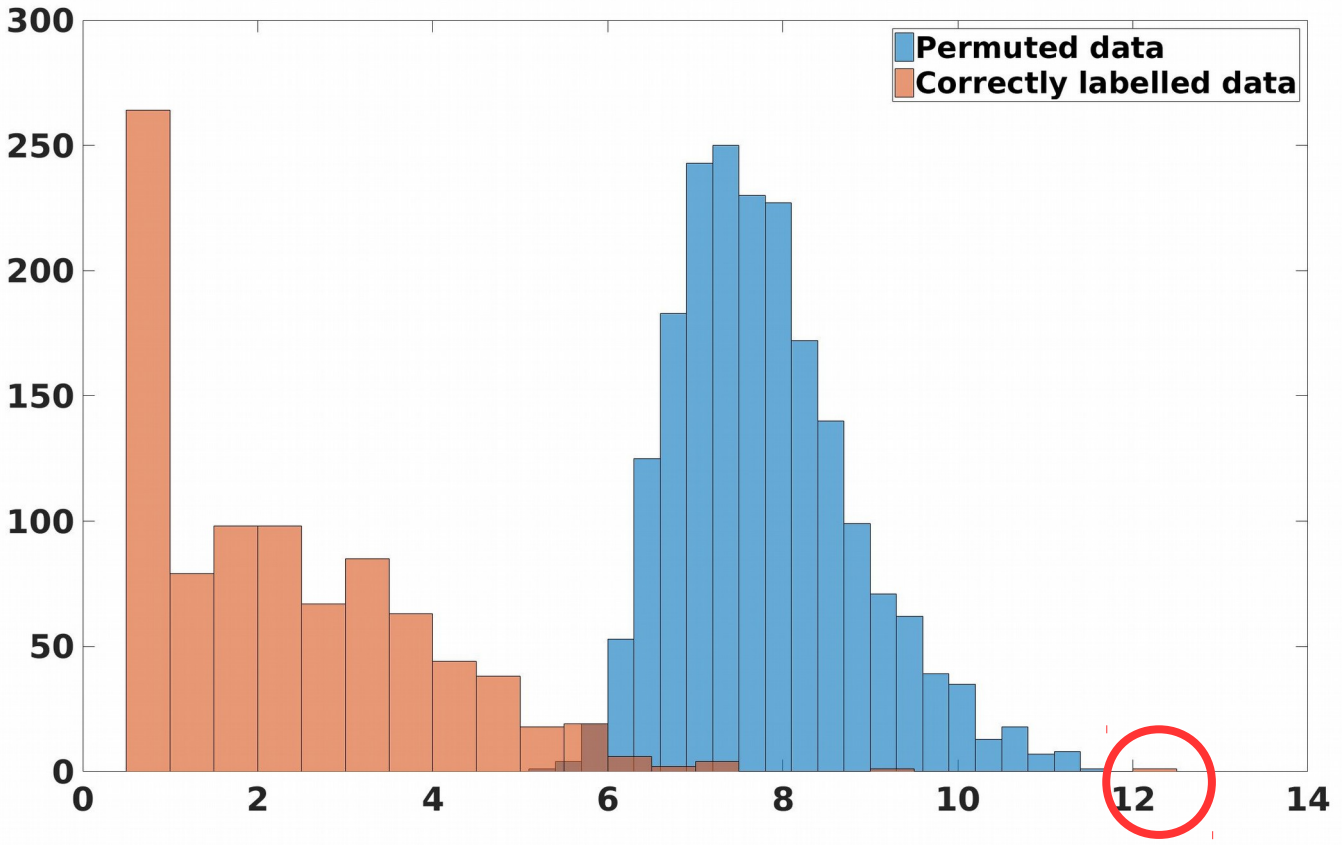
```
cfg = [];
cfg.latency = [1.500 1.500];
cfg.frequency = [1 100];
cfg.parameter = 'powspctrm';
cfg.method = 'montecarlo'; %% do a permutation test
cfg.alpha = 0.05; %% critical value around ± 2.09
cfg.statistic = 'depsamplesT'; % use a dependent
samples t-test
cfg.design(1, :) = [1:n_subjects 1:n_subjects];
cfg.design(2, :) = [ones(1, n_subjects) 2 * ones(1,
n_subjects)];
cfg.uvar = 1; % first row of cfg.design, containing
the (u)nits (subjects)
cfg.ivar = 2; % second row of cfg.design, the
(i)ndependent events

cfg.correctm = 'cluster'; % correct with cluster
method
cfg.clusteralpha = 0.025; % alpha for clusters
cfg.clusterstatistic = 'maxsum'; % way to calculate
T
cfg.clustertail = 0;
cfg.neighbours = neighbours; % a neighbour
structure, see ft_prepare_neighbours
cfg.numrandomization = 2000; % n randomizations

stat = ft_freqstatistics(cfg, dataset_1, dataset_2);
```

(Log)-*T*-values



Let’s run (many) permutations to sample the *T*-distribution

(Log)-*T*-values



log(*T*) = 12, *p* = 0.0005

What is the null hypothesis when doing permutation based test?

The null hypothesis

Unit variable	Independent variable
1	1
2	1
...	...
19	1
20	1
1	2
2	2
...	...
19	2
20	2

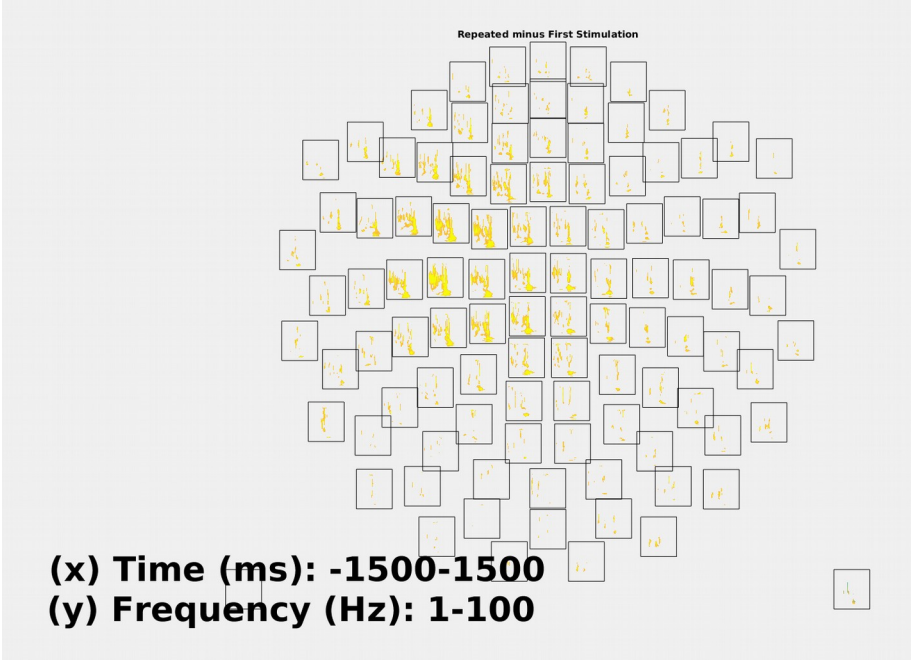
Null hypothesis: data is exchangeable → more informally: the way that you label your conditions do not matter

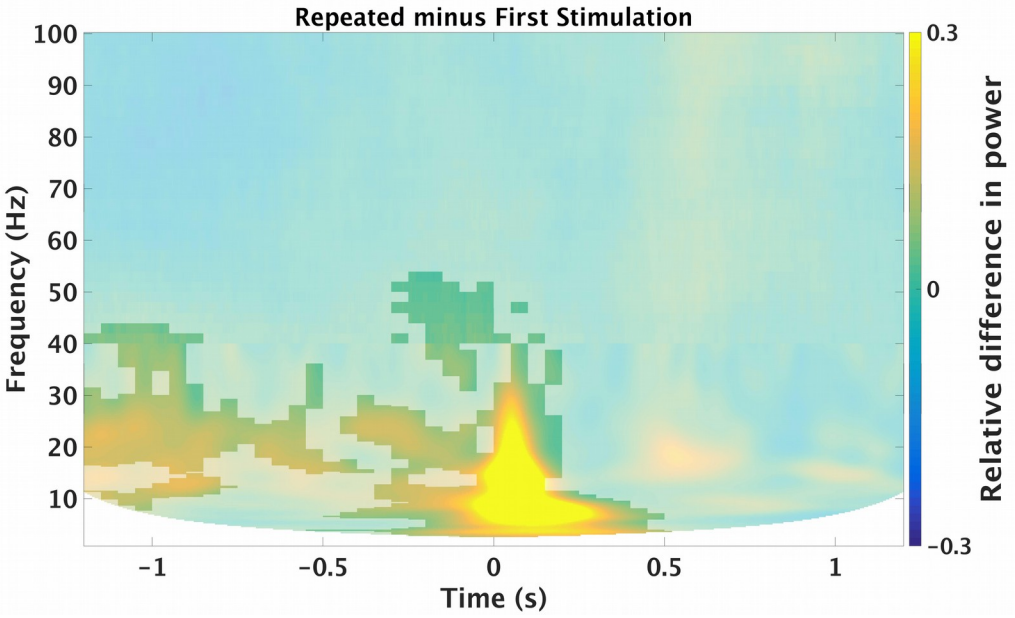
Alternative hypothesis informally: your way of labelling the conditions matter

This also means that it is nonsensical to speak of *statistically significant clusters*

(It doesn’t mean that clusters are not interesting to investigate further)

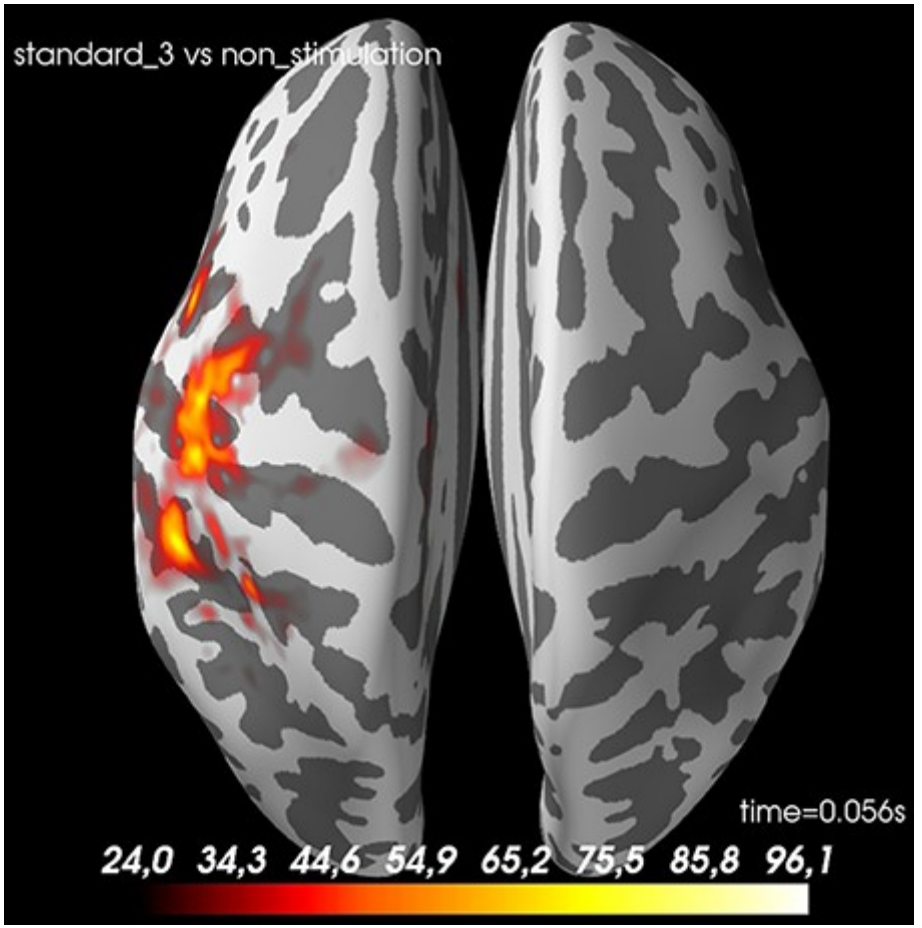
Clusters


Repeated minus First Stimulation
(x) Time (ms): -1500-1500
(y) Frequency (Hz): 1-100


Repeated minus First Stimulation
Frequency (Hz)
100
90
80
70
60
50
40
30
20
10
-1
-0.5
0
0.5
1
Time (s)
0.3
0
-0.3
Relative difference in power

Clusters

Clusters can also be used for source space



Thank you!