Introduction to EEG, MEG and analysis with the FieldTrip toolbox

Robert Oostenveld

Donders Institute, Radboud University, Nijmegen, NL
NatMEG, Karolinska Institute, Stockholm, SE

What is FieldTrip

a MATLAB toolbox for the analysis of MEG, EEG and animal electrophysiology data

can import data from many different file formats

contains algorithms for spectral analysis, source reconstruction, statistics, connectivity, …

Talk outline

What kind of signals are generated in the brain

How do we record those signals

Analyzing those signals with FieldTrip

Background on the FieldTrip toolbox

Talk outline

What kind of signals are generated in the brain

How do we record those signals

Analyzing those signals with FieldTrip

Background on the FieldTrip toolbox

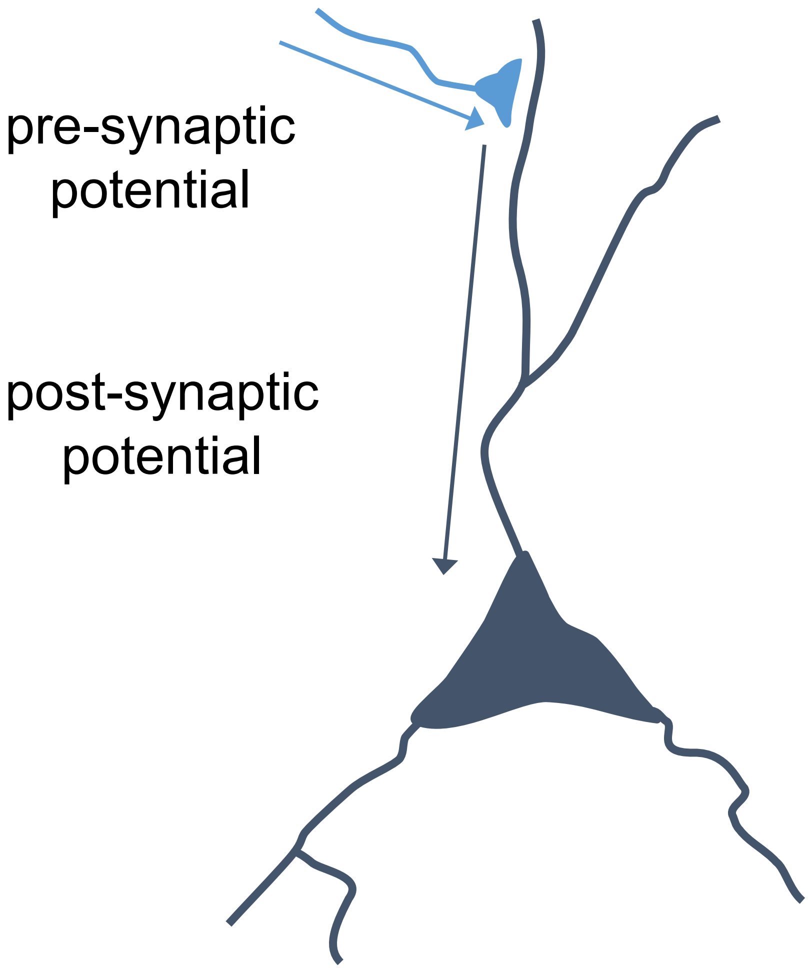
What kind of signals are generated in the brain

- We measure the scalp potentials or field associated with post-synaptic potentials in pyramidal neurons

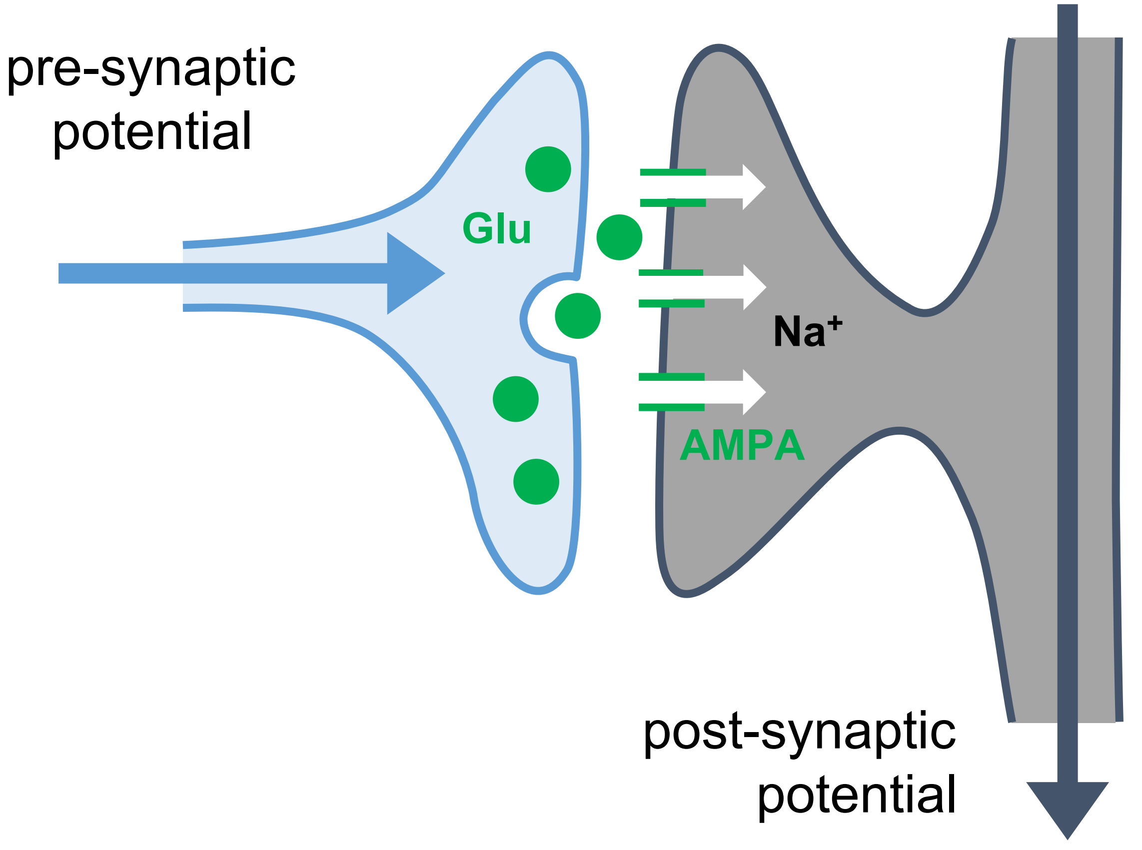
- These PSPs represent the exitatory and inhibitory input that these neurons receive

- Usually we study this neuronal input following the presentation of a stimulus or following a cognitive event

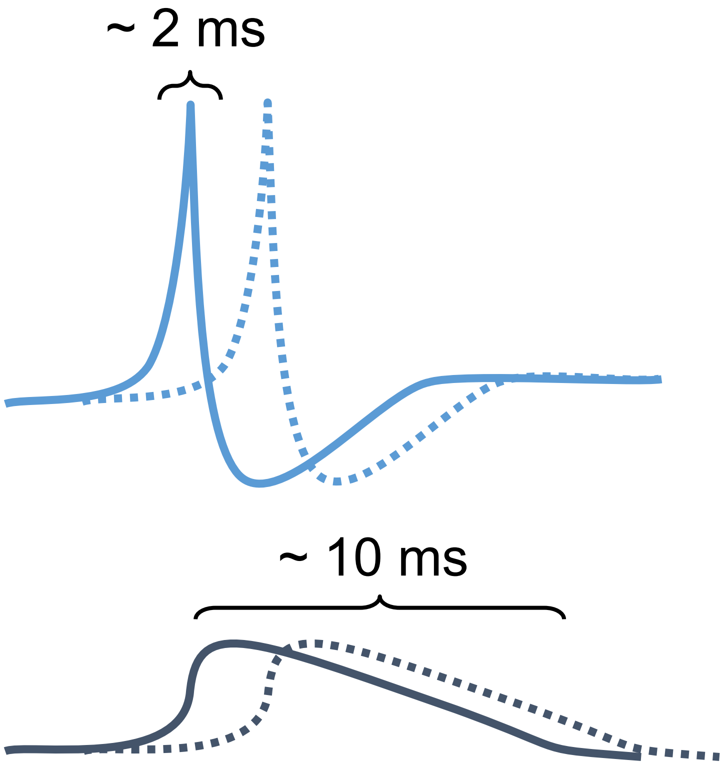
What generates the currents and fields



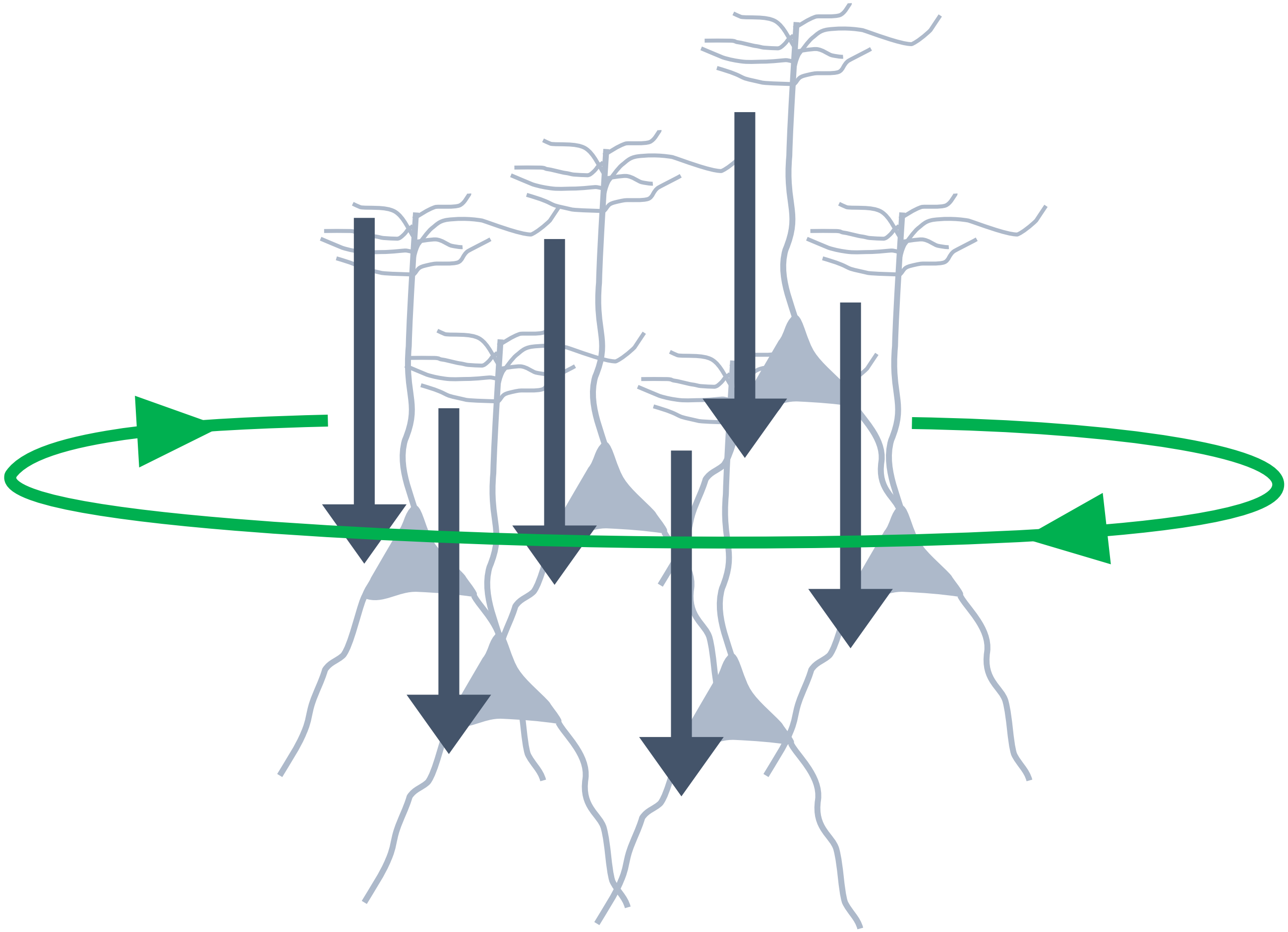
What generates the currents and fields



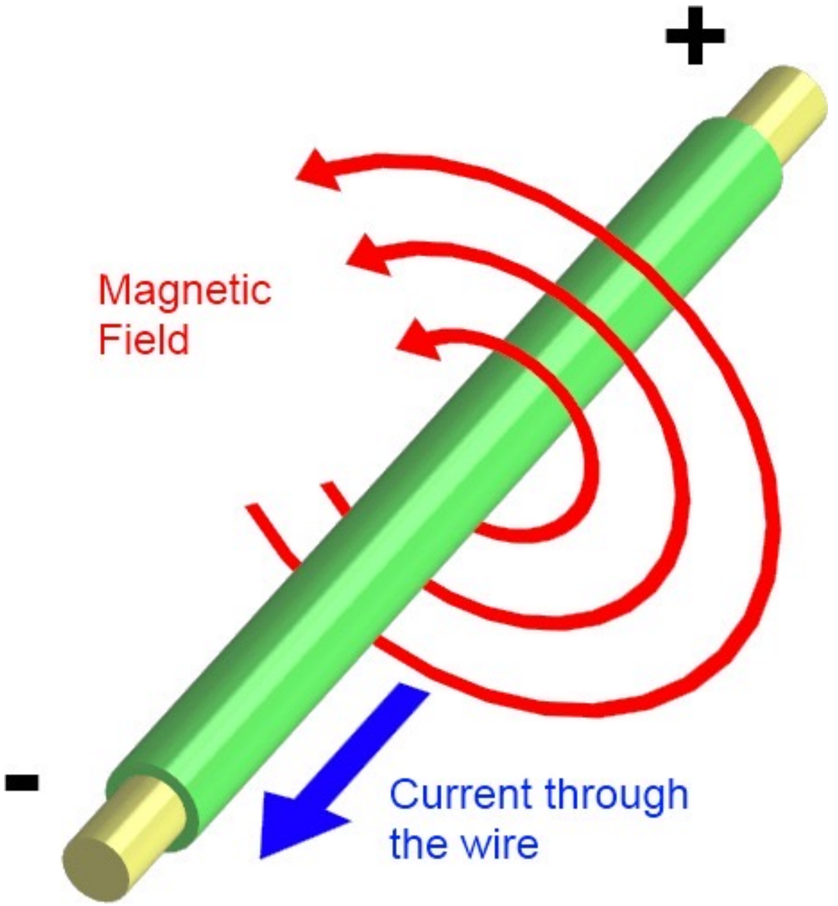
What generates the currents and fields



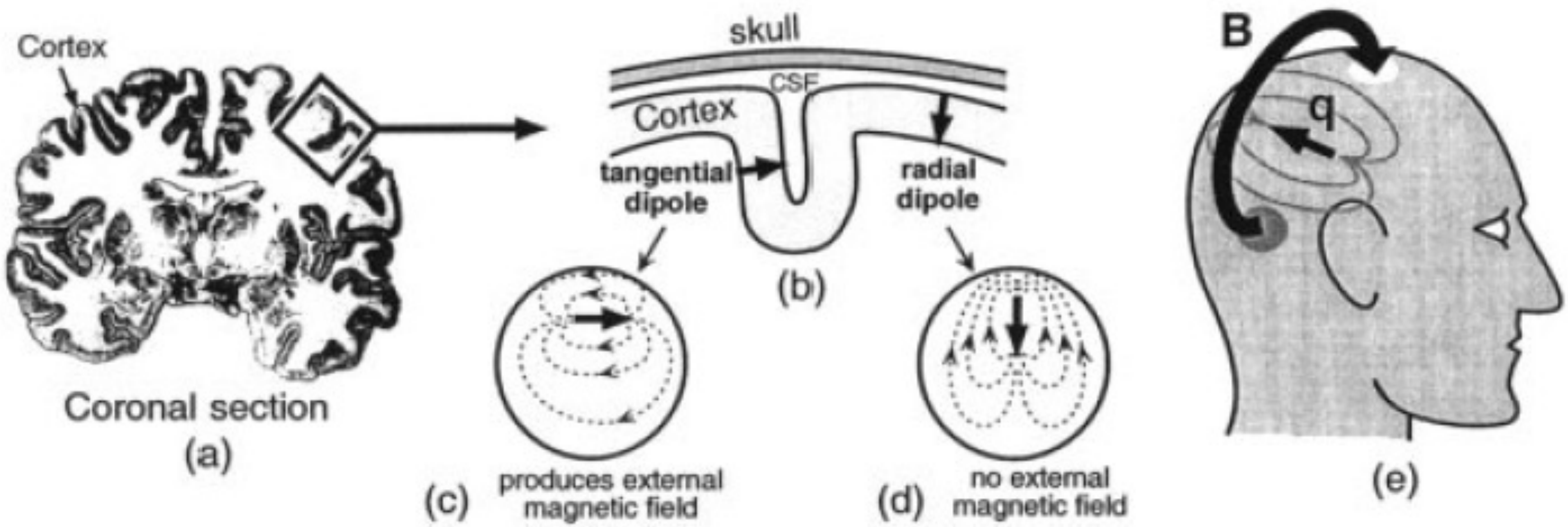
What generates the currents and fields



What generates the currents and fields

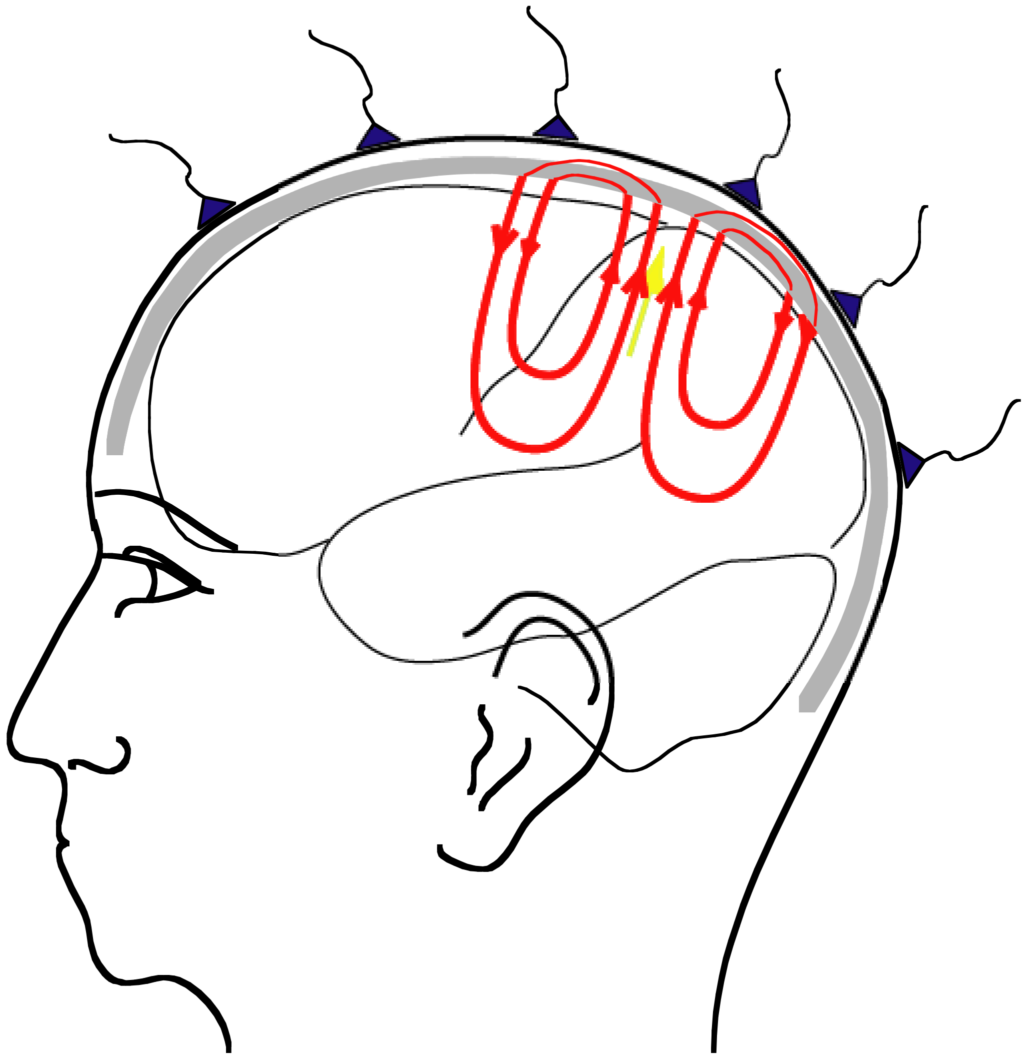


What generates the currents and fields

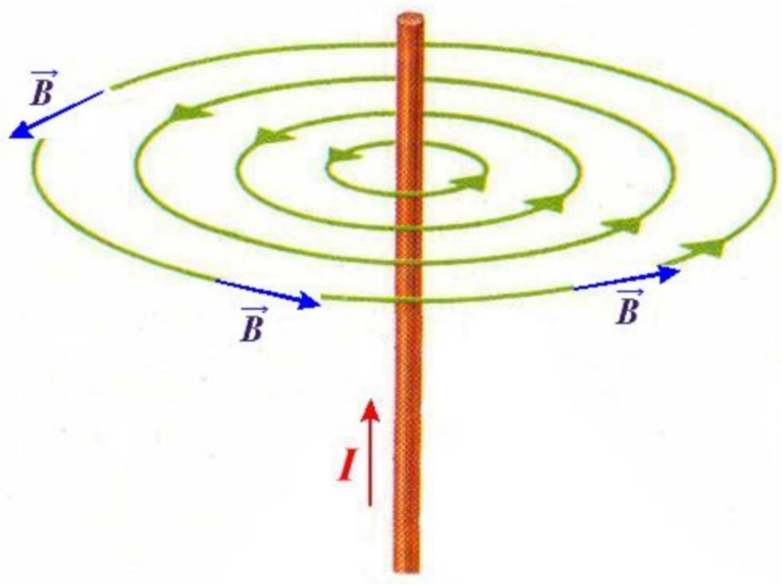


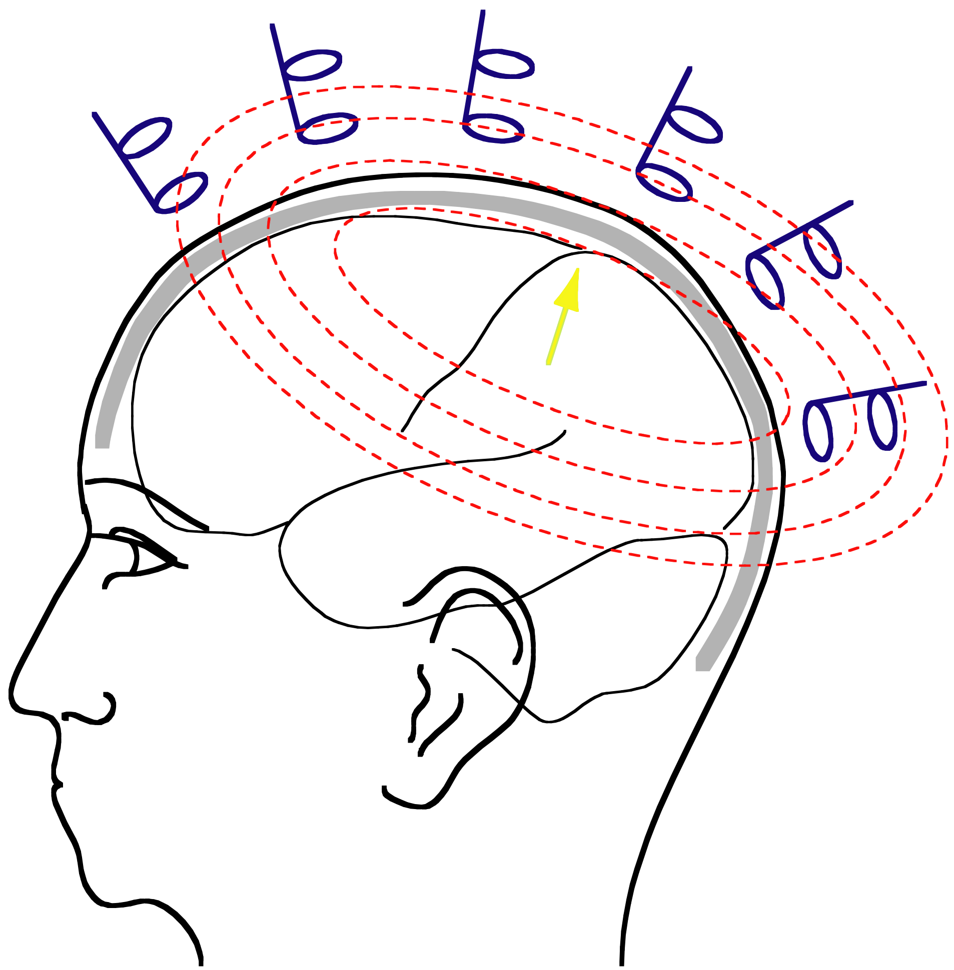
Vrba, 2001

EEG volume conduction



Electric current → magnetic field





Talk outline

What kind of signals are generated in the brain

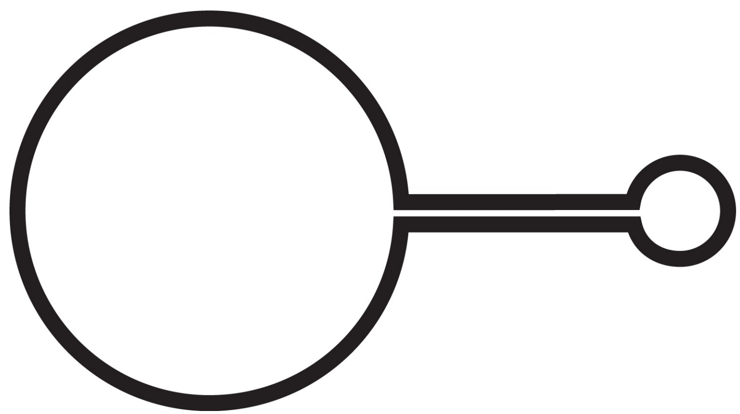
How do we record those signals

Analyzing those signals with FieldTrip

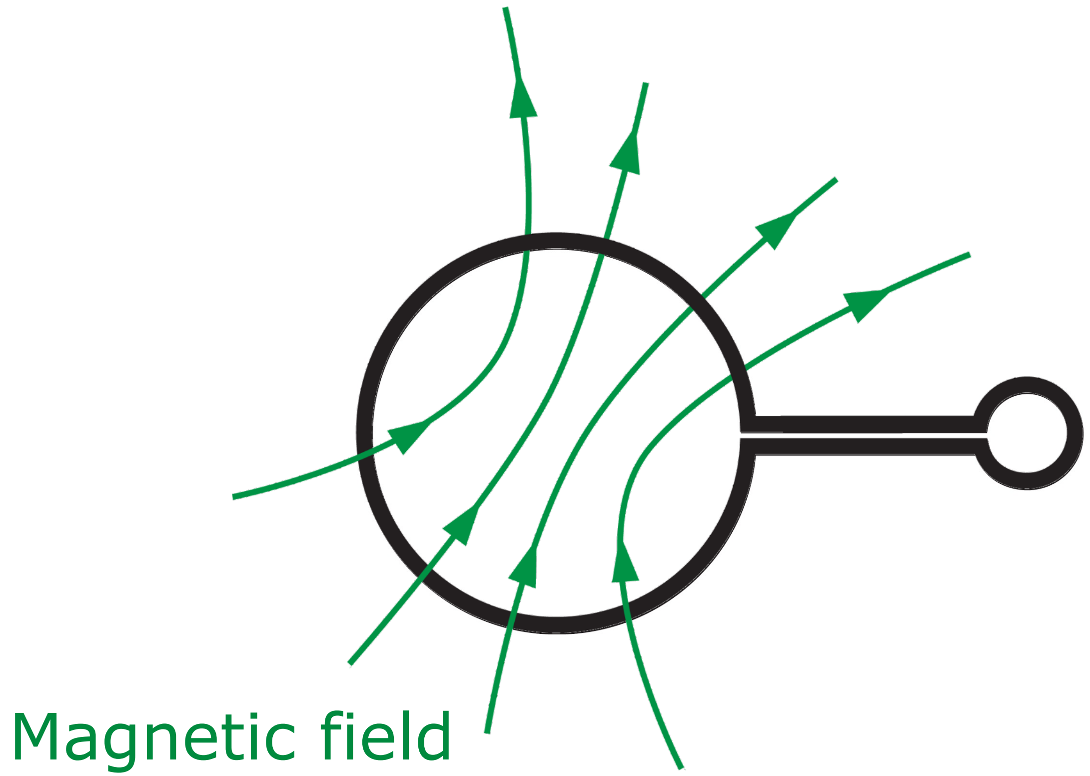
Background on the FieldTrip toolbox

How do we record these signals from the brain

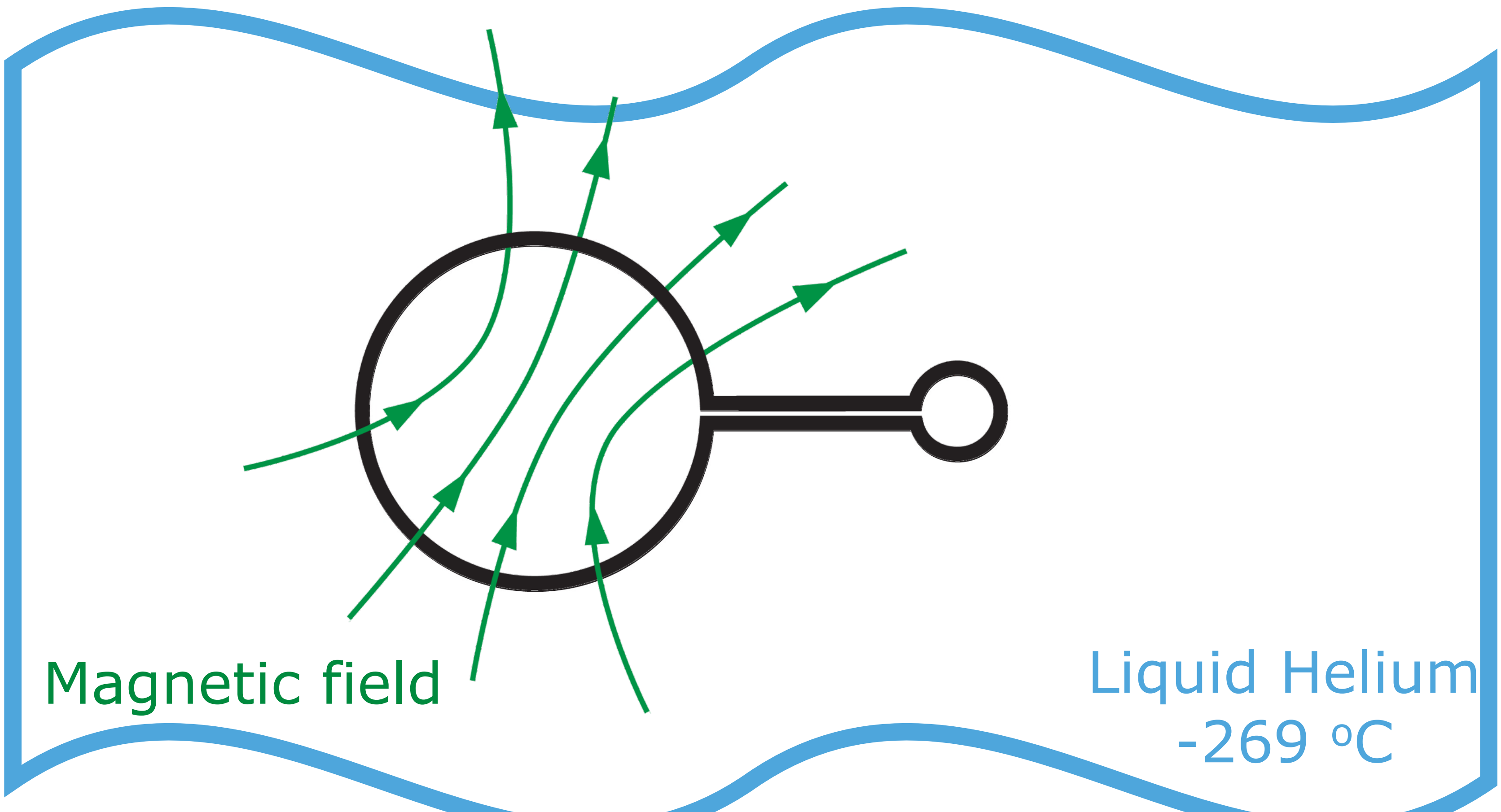
Recording small magnetic fields using SQUIDs



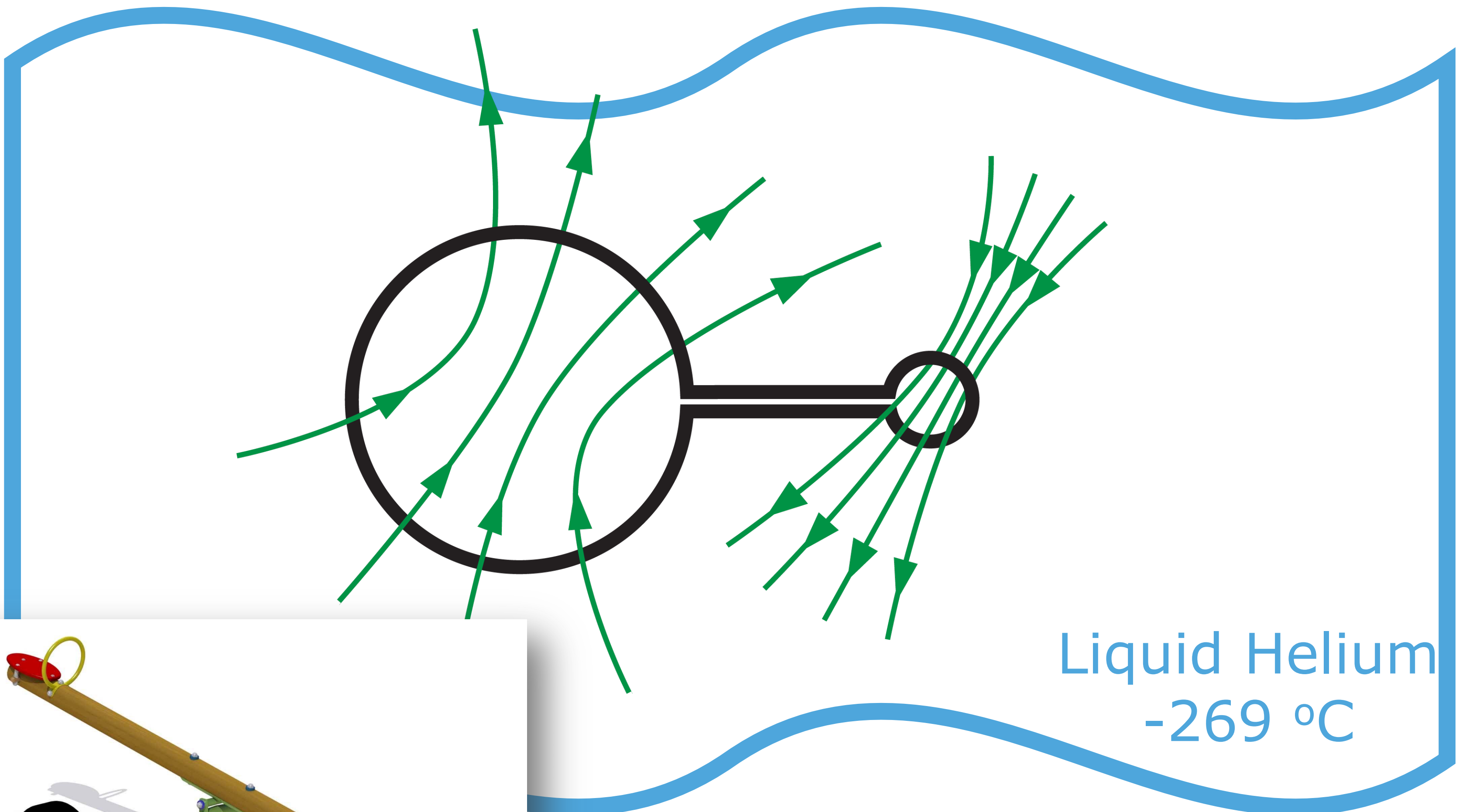
Recording small magnetic fields using SQUIDs



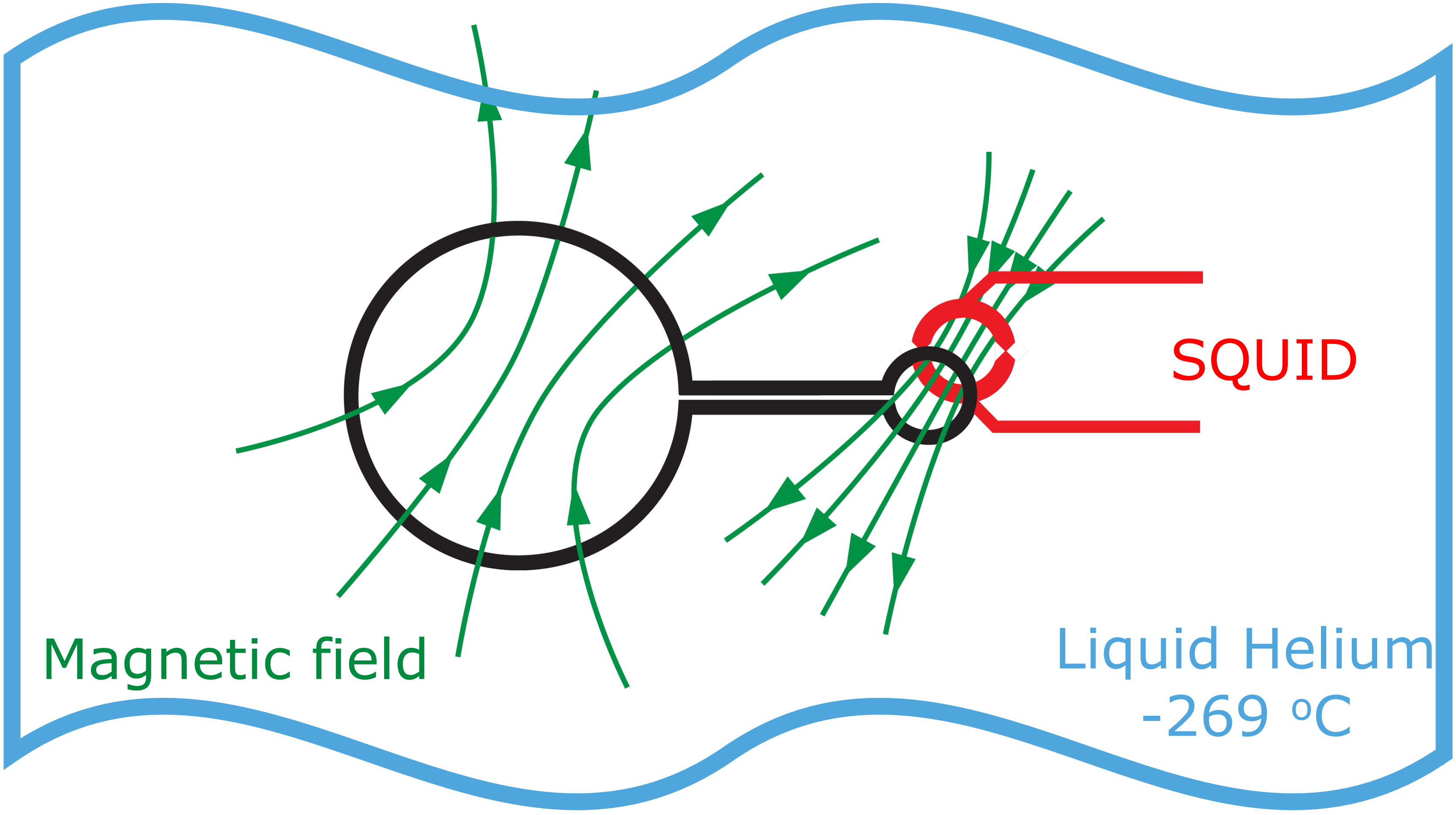
Recording small magnetic fields using SQUIDs



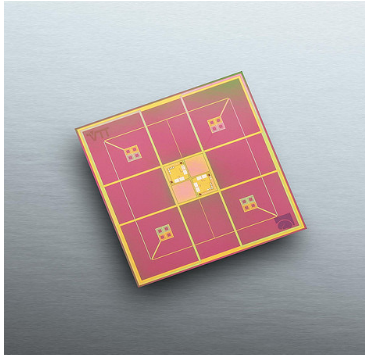
Recording small magnetic fields using SQUIDs

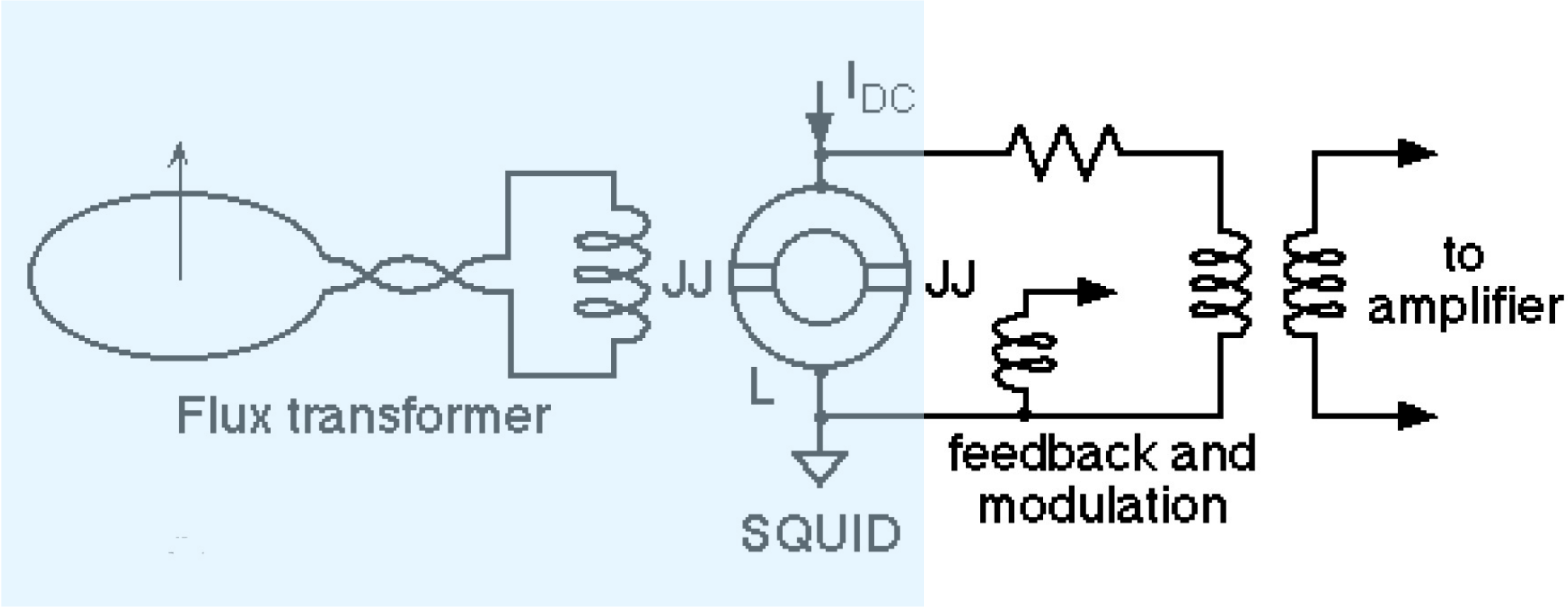


Recording small magnetic fields using SQUIDs



Magnetic field detectors
Superconducting **QU**antum **I**nterference **D**evice





Magnetic field strength - compared

10^{0}	1.5 - 3 T	MRI systems
10^{-3}	5 mT	typical refrigerator magnet
10^{-6}	30-60 uT	earth's magnetic field
10^{-9}	0.1 -10 nT	heliosphere
10^{-12}	0.1 - 1.0 pT	spontaneous brain activity
10^{-15}	10-100 fT	ERF differences

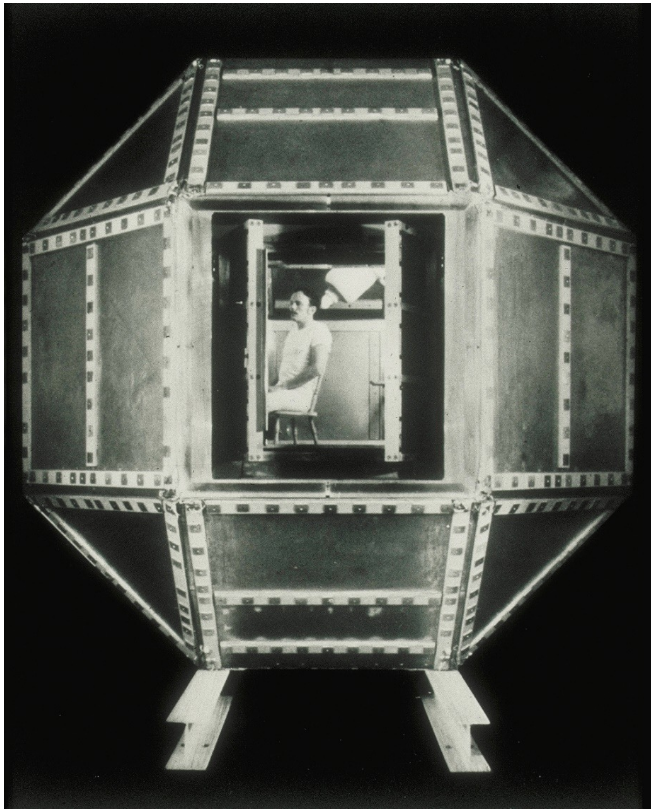
Technical challenges of MEG

Requires sensitive magnetic detectors

Deal with environmental noise

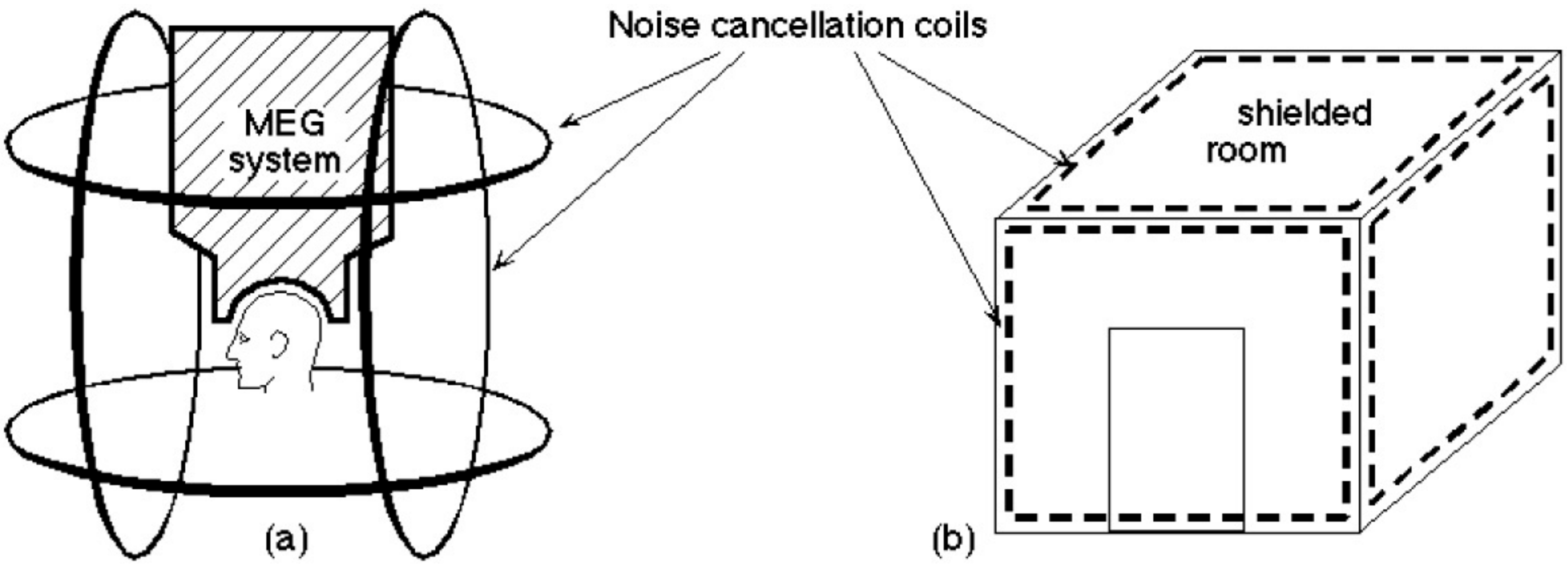
- shielding
- sensor design
- reference sensors for noise subtraction

Shielding - passive

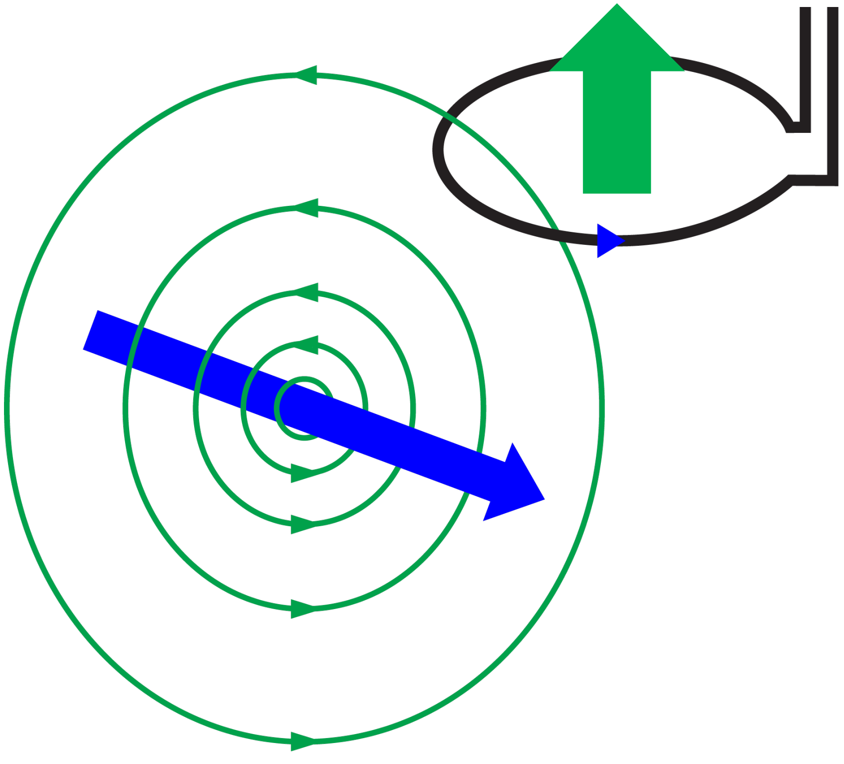


The magnetically shielded room built by David Cohen at MIT's Francis Bitter National Magnet Laboratory in 1969.

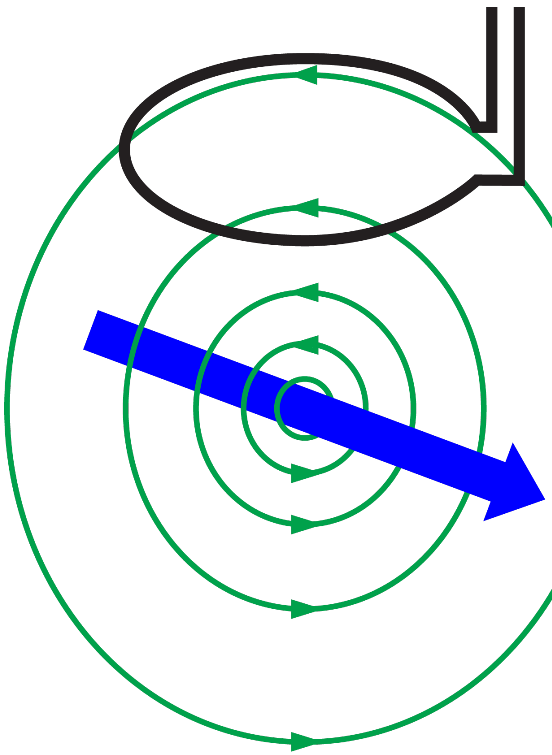
Shielding - active



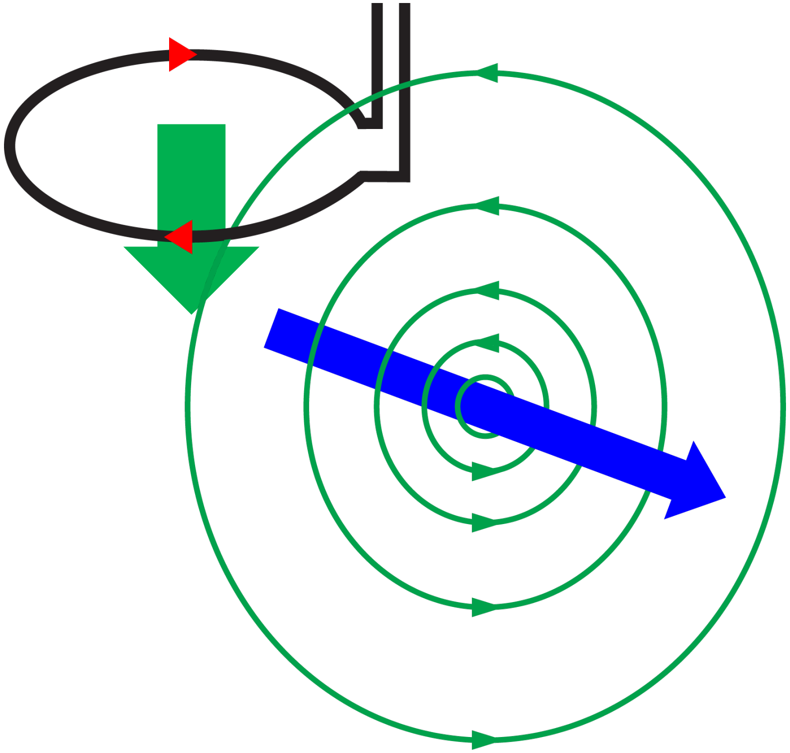
Magnetometer



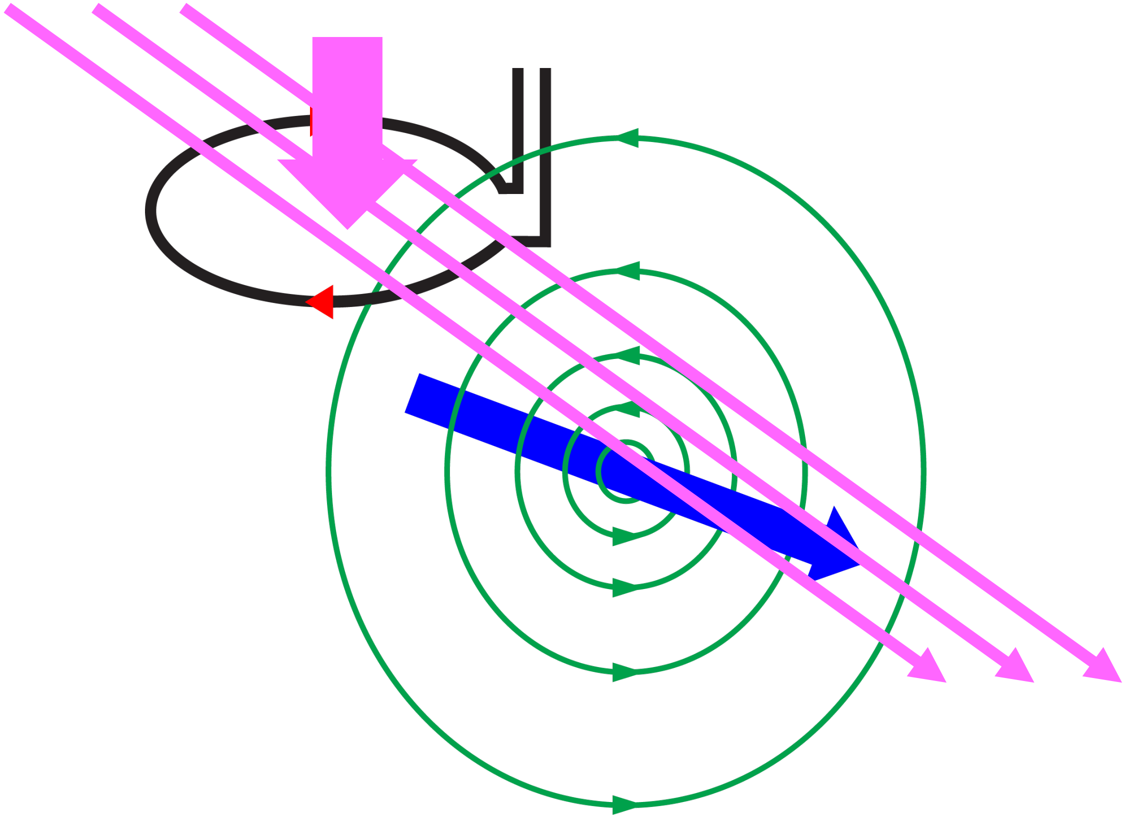
Magnetometer



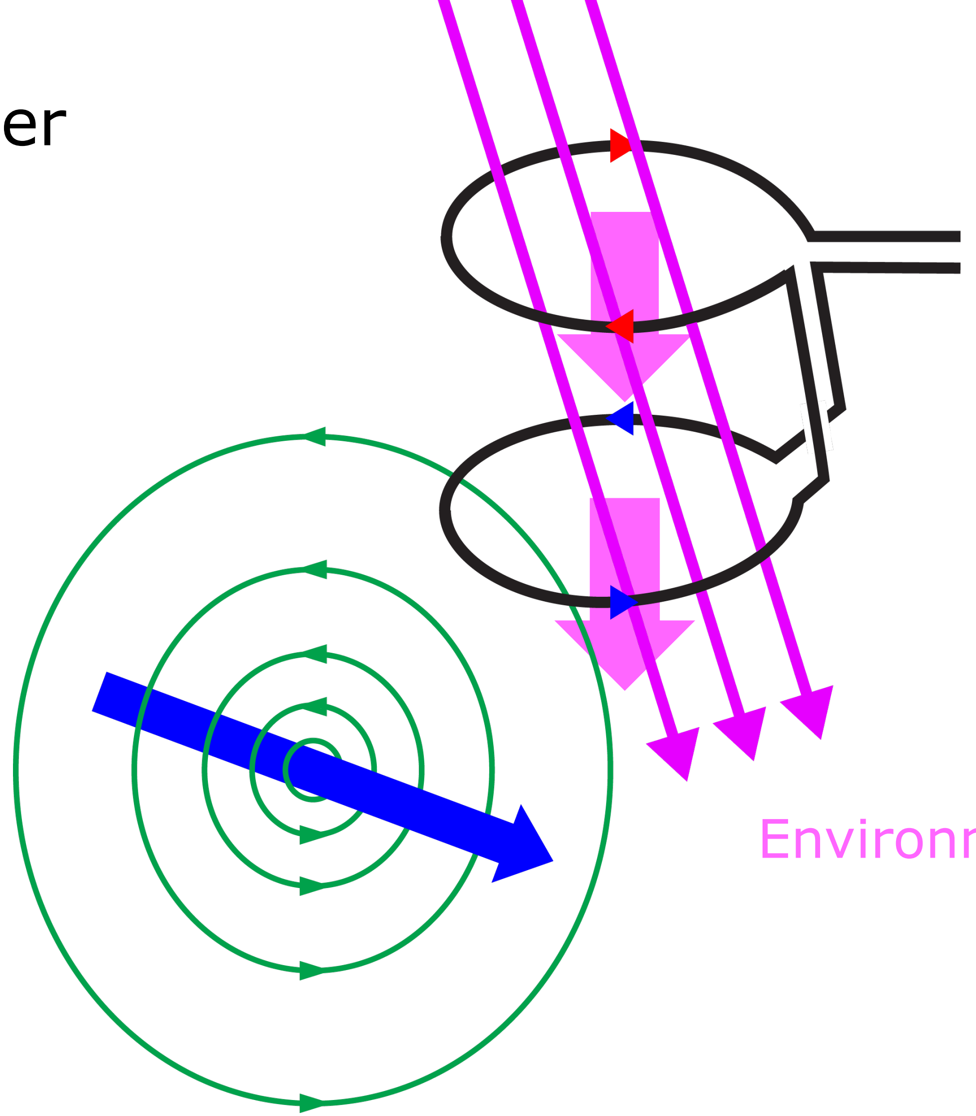
Magnetometer



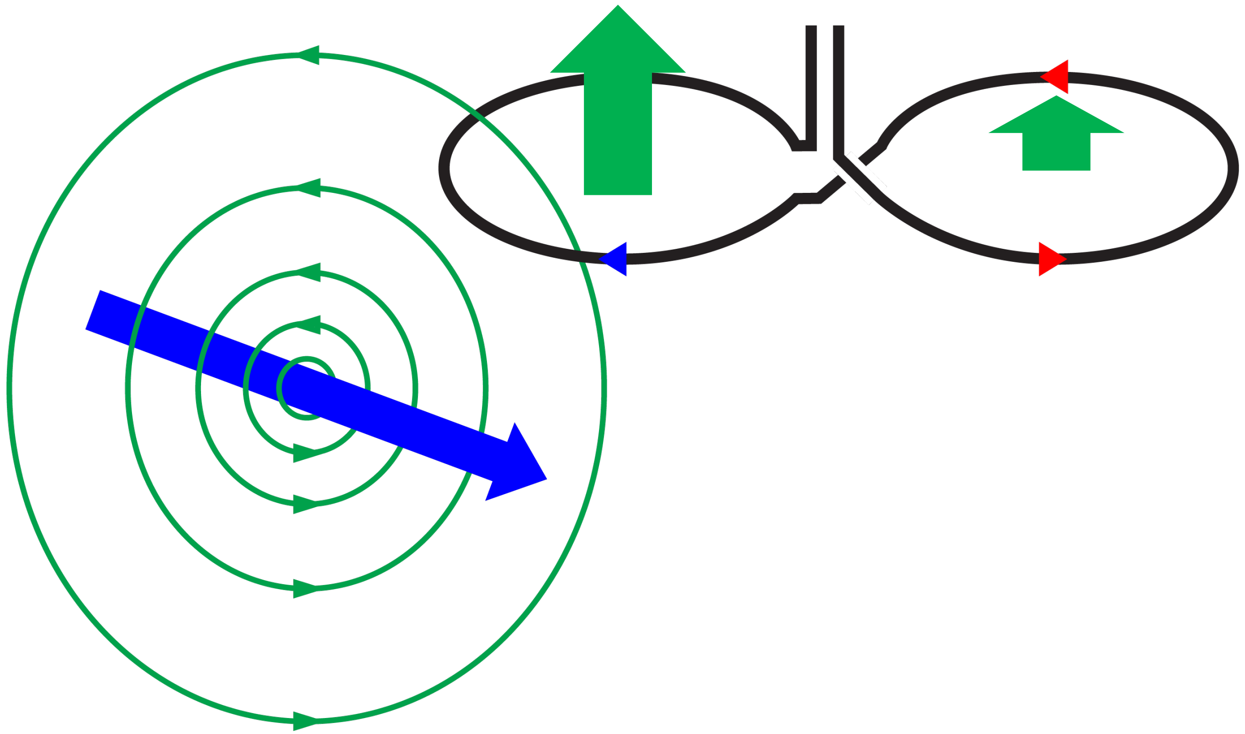
Magnetometer



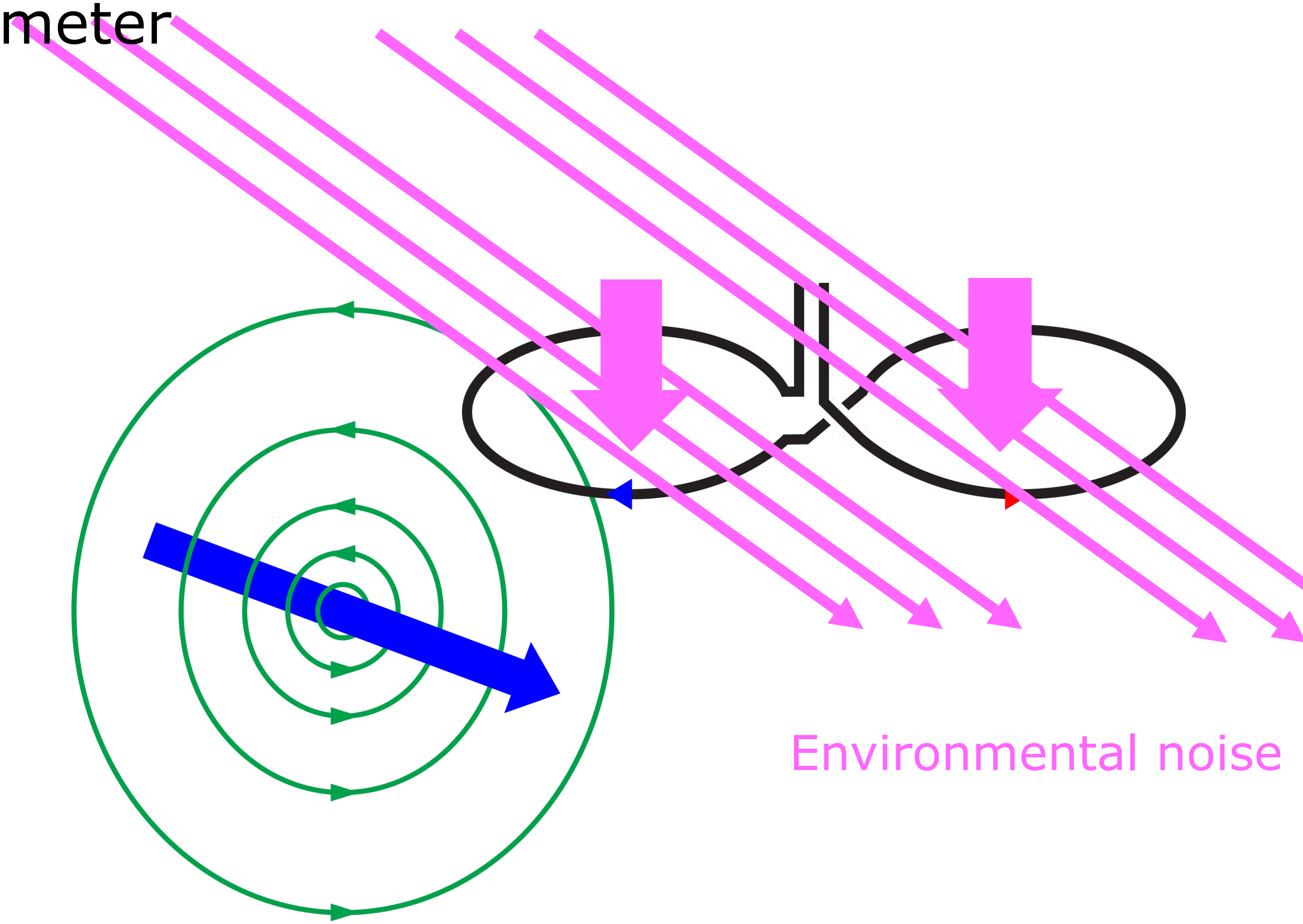
Axial gradometer



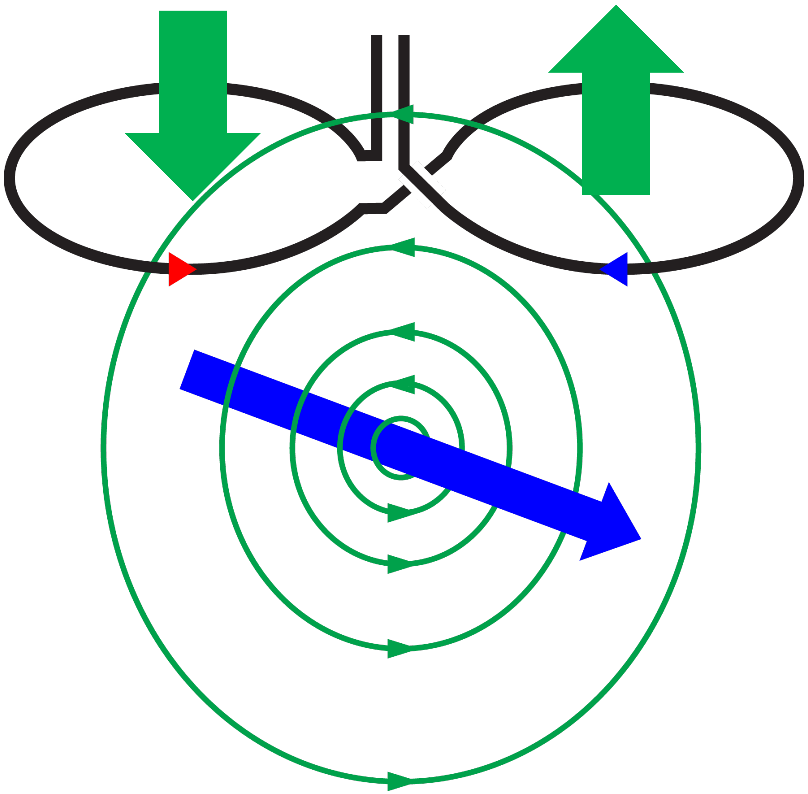
Planar gradiometer



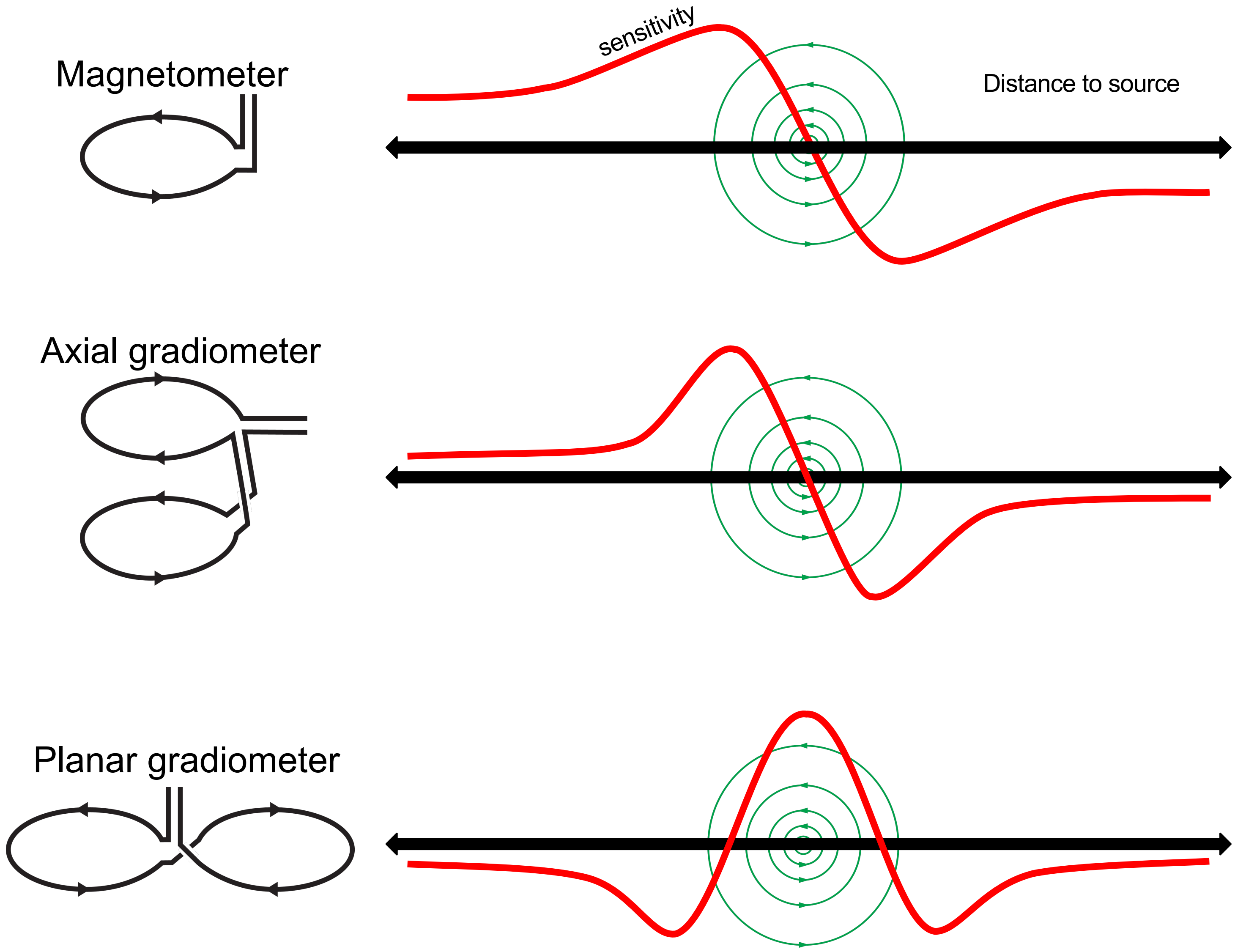
Planar gradiometer



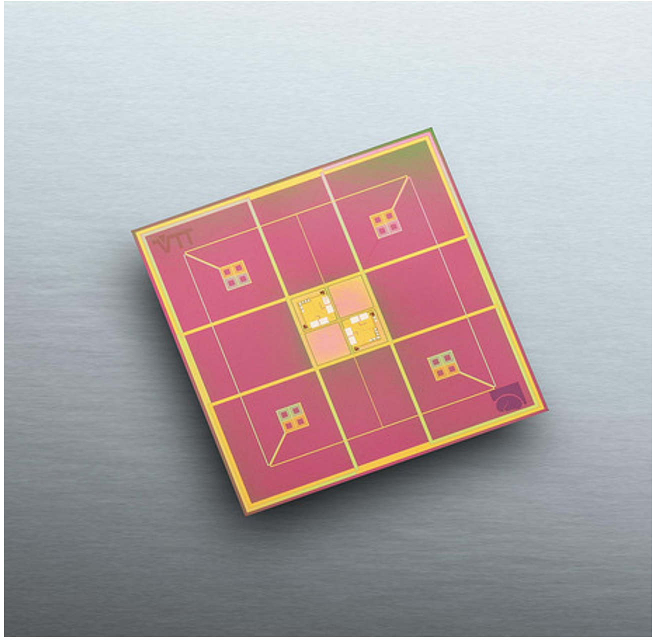
Planar gradiometer



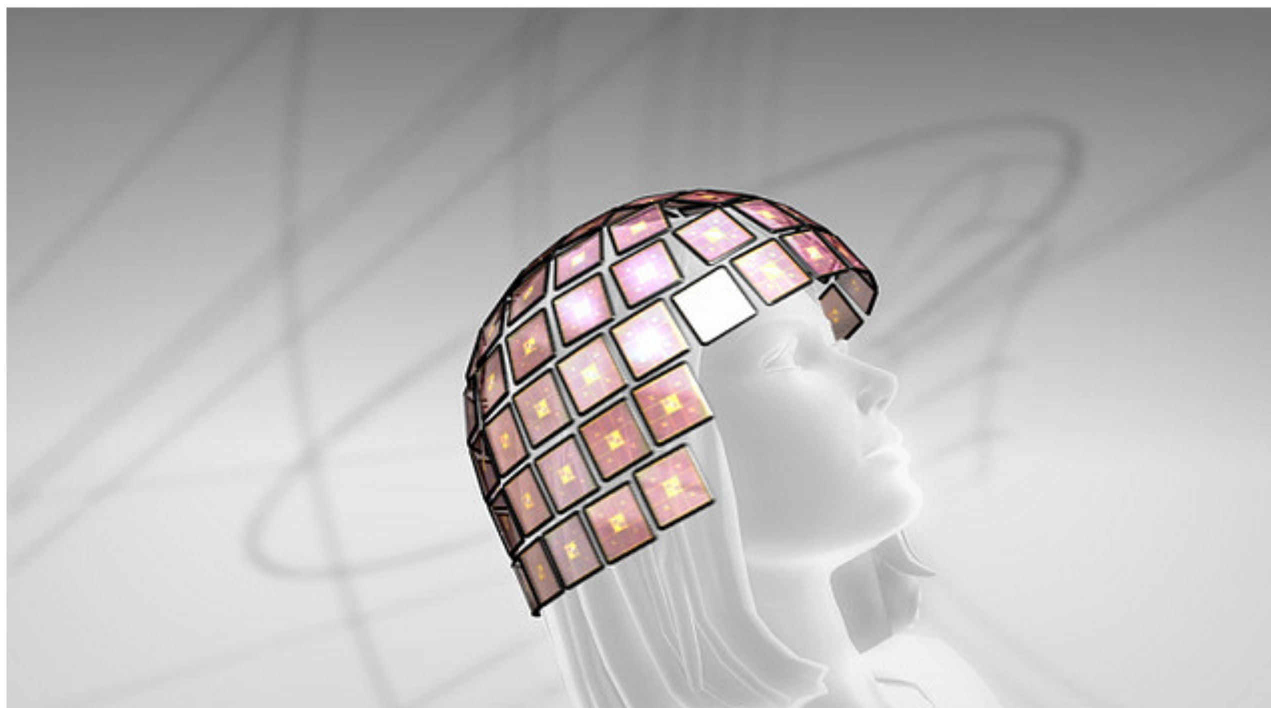
MEG sensor – sensitivity profile



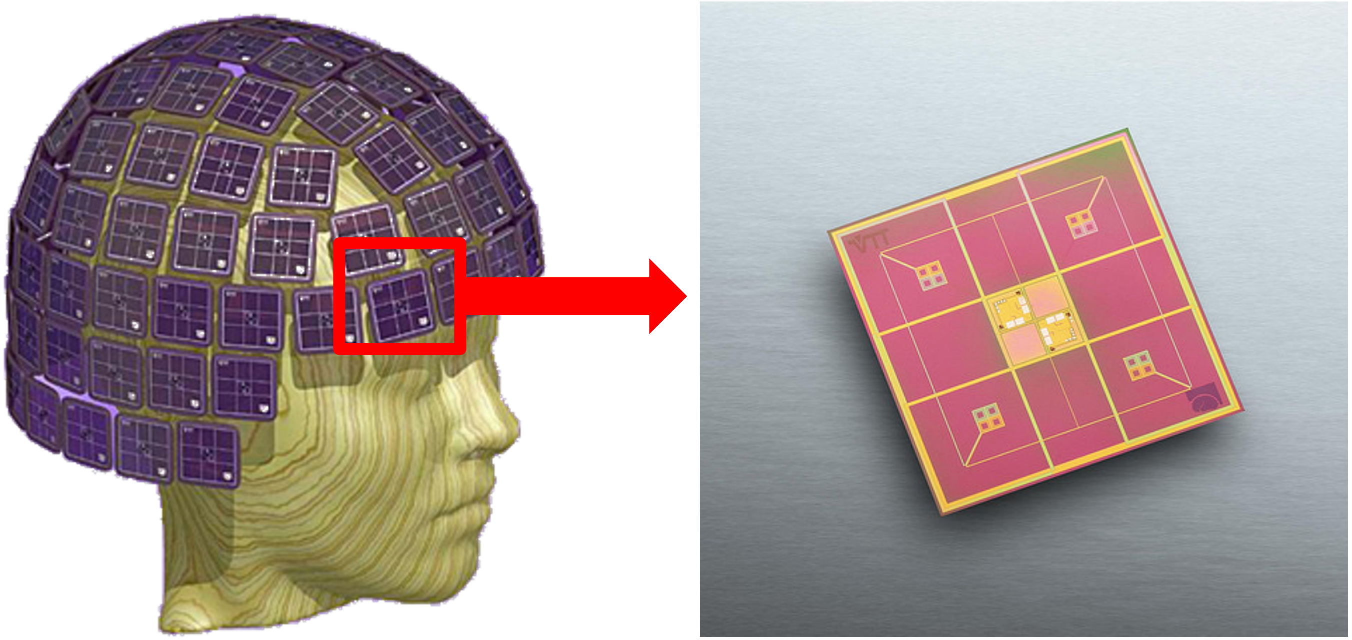
Neuromag/Elekta/Megin



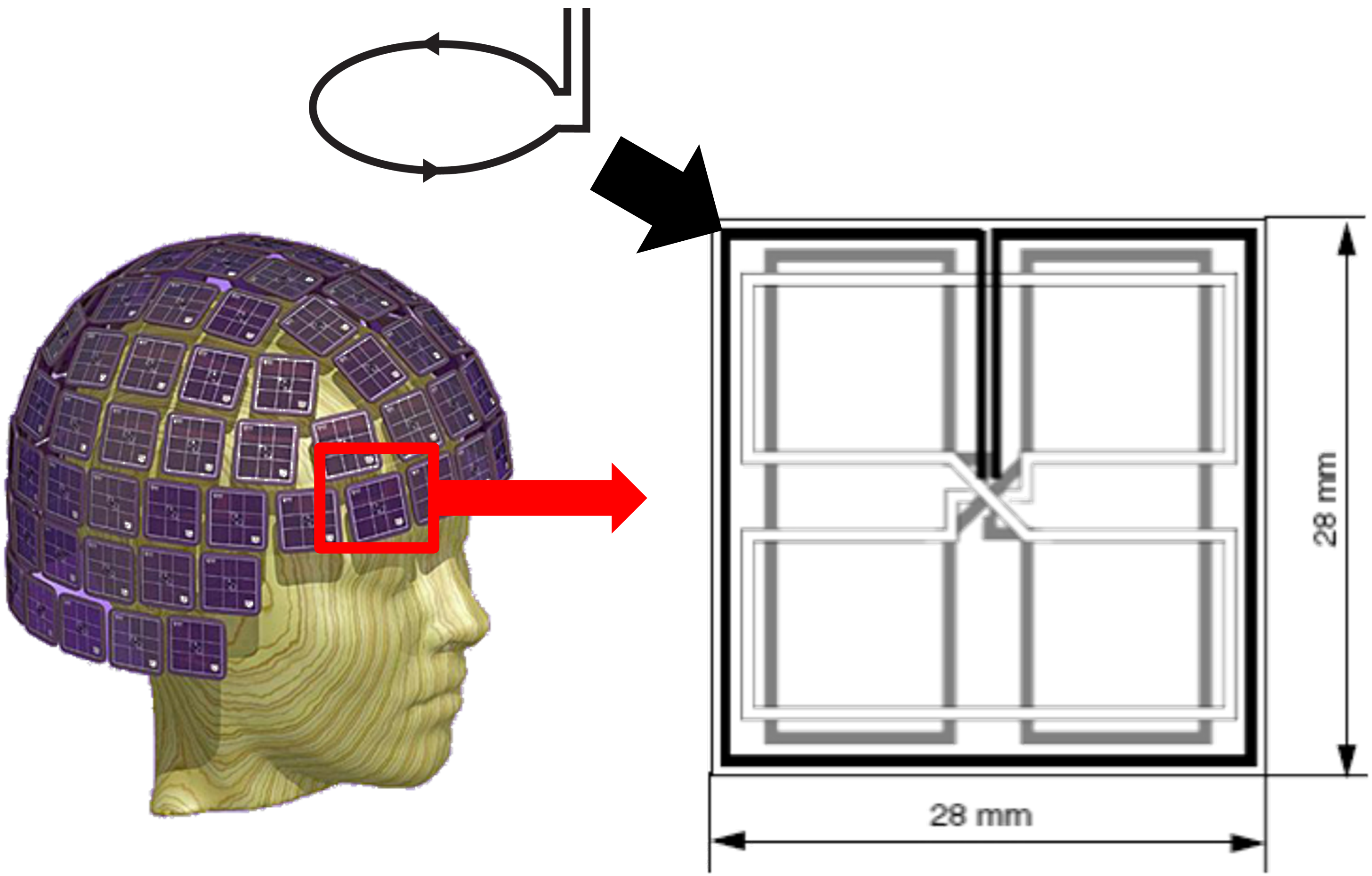
102 magnetometers
204 planar gradiometers
306 channels total



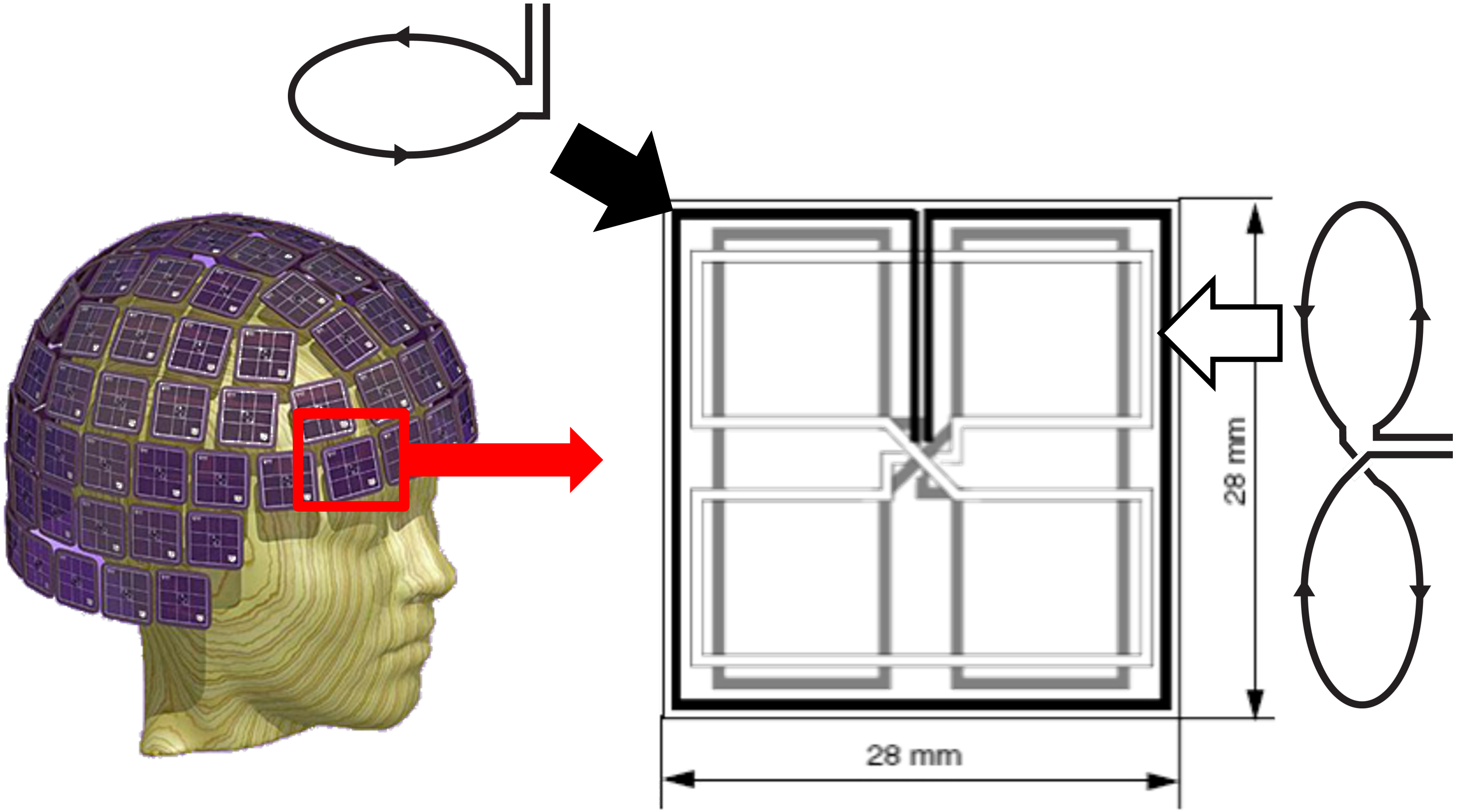
Neuromag/Elekta/Megin



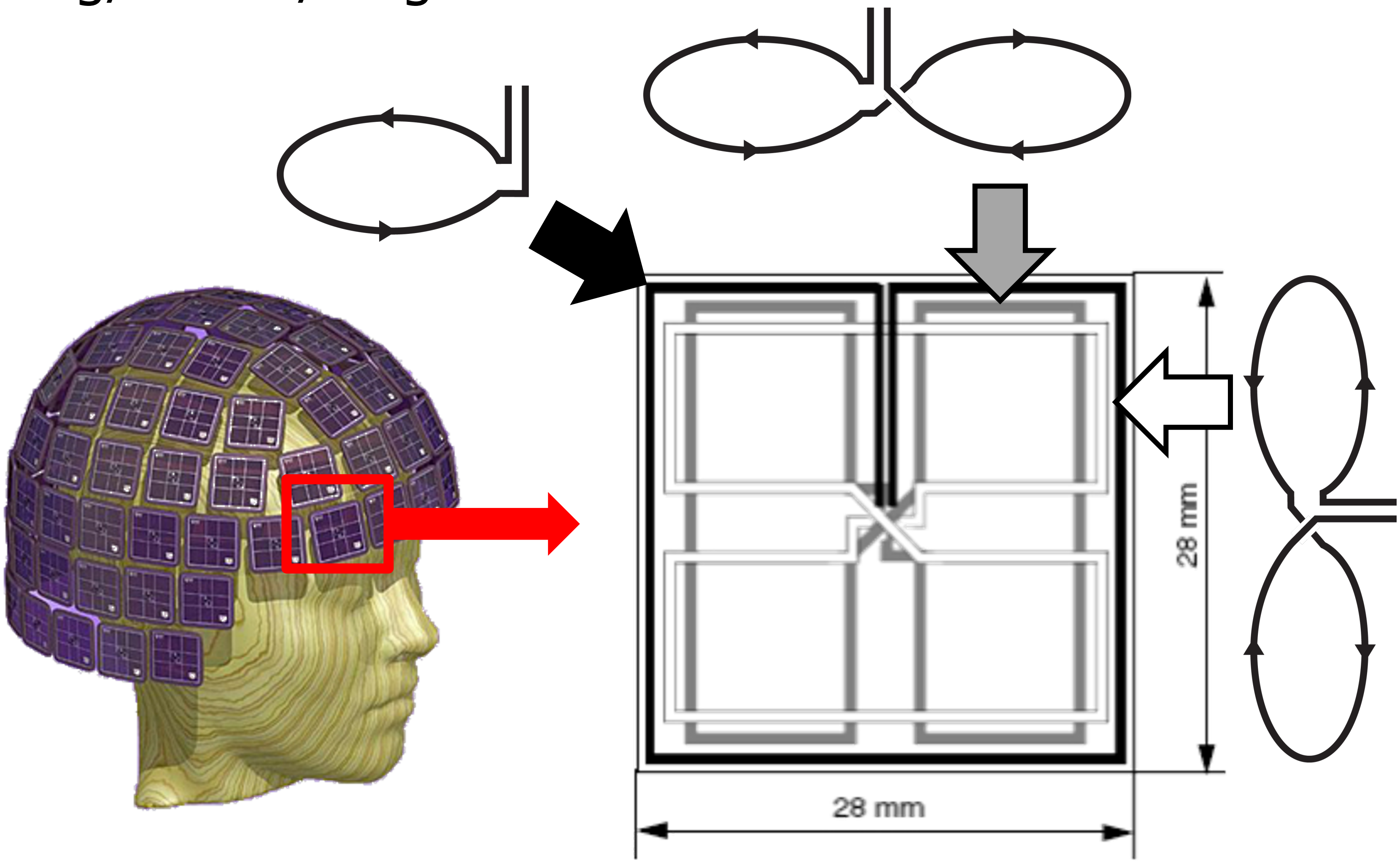
Neuromag/Elekta/Megin



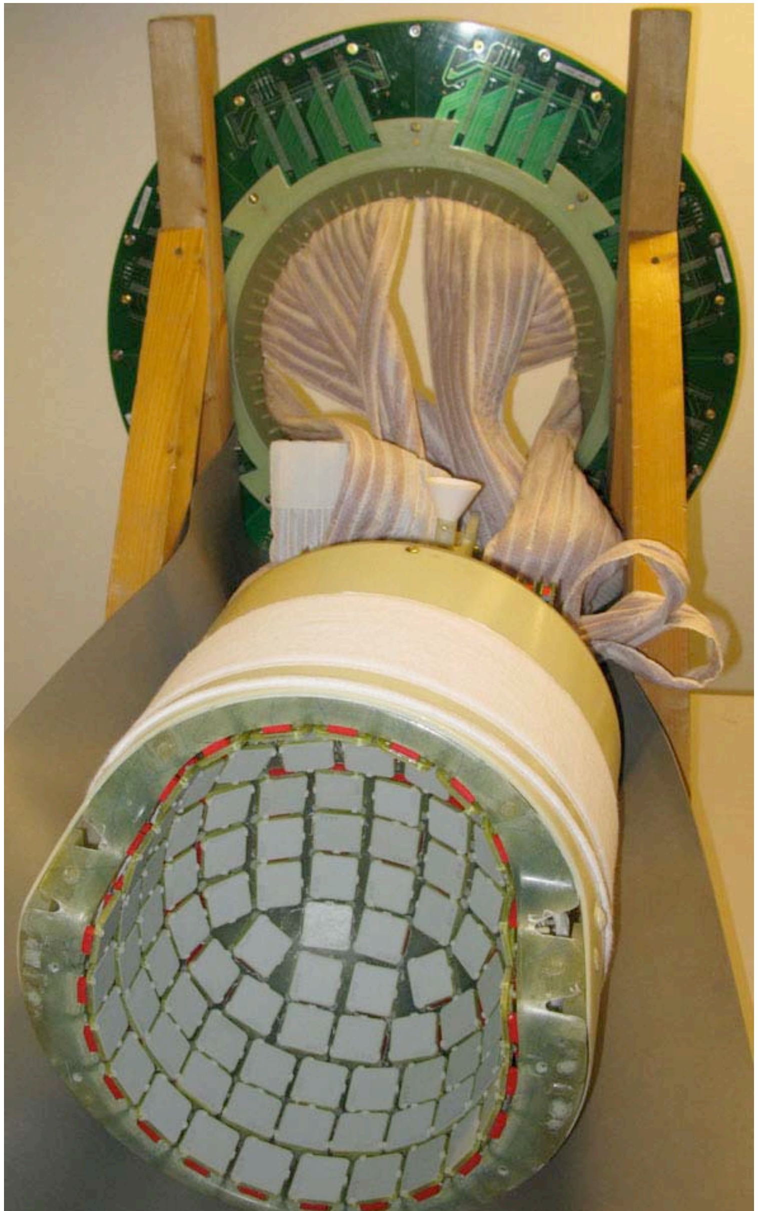
Neuromag/Elekta/Megin

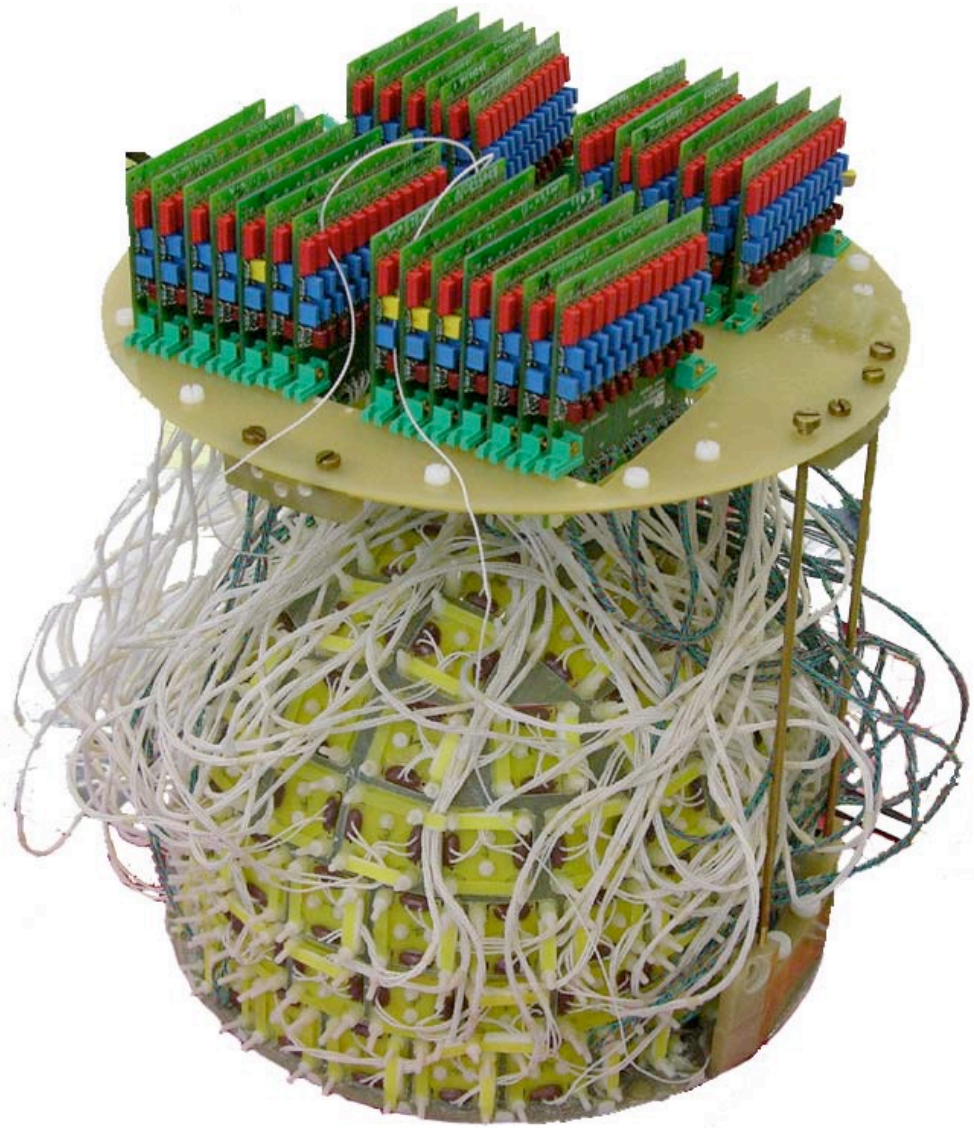


Neuromag/Elekta/Megin

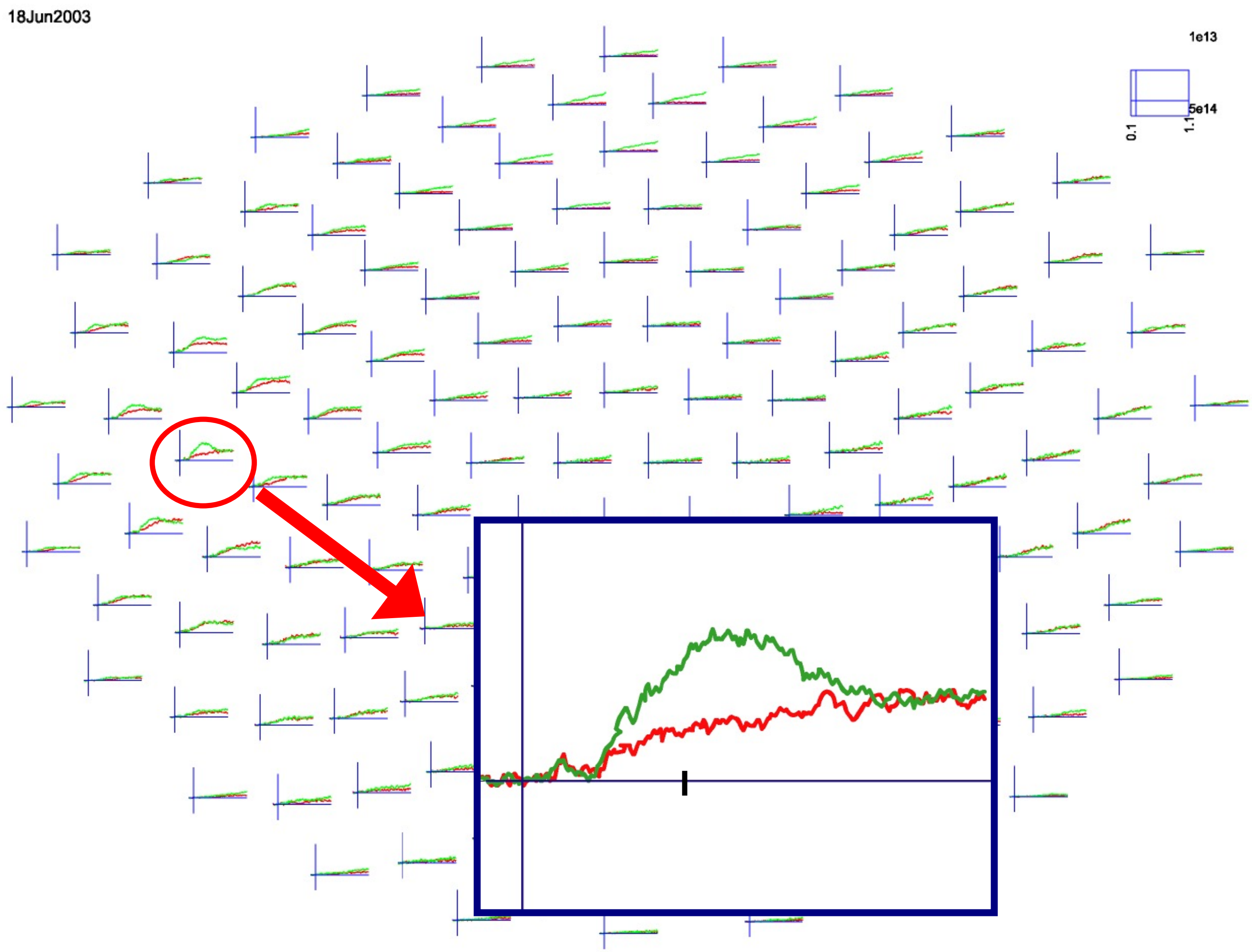


Neuromag/Elekta/Megin

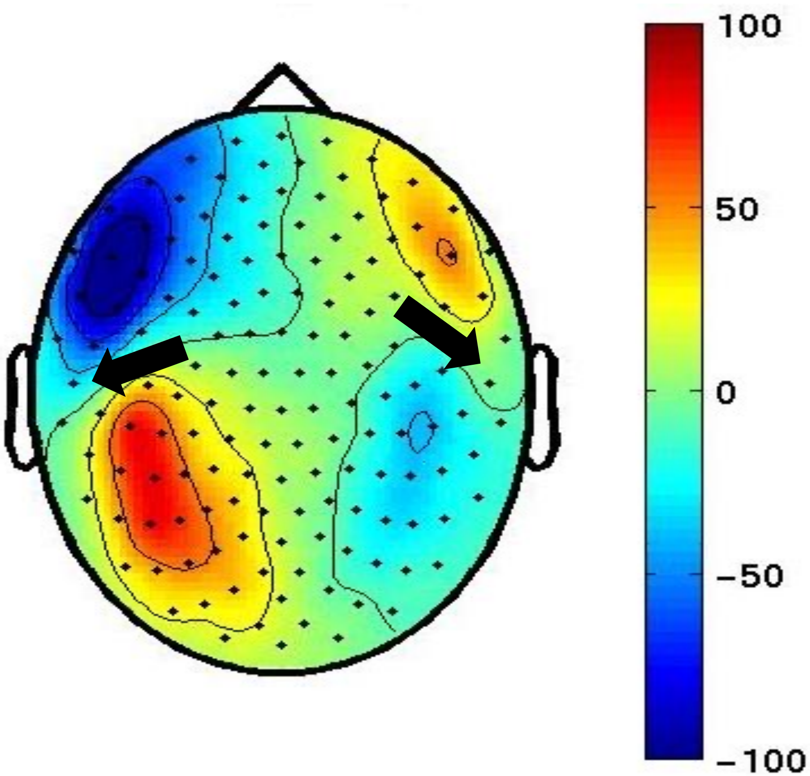




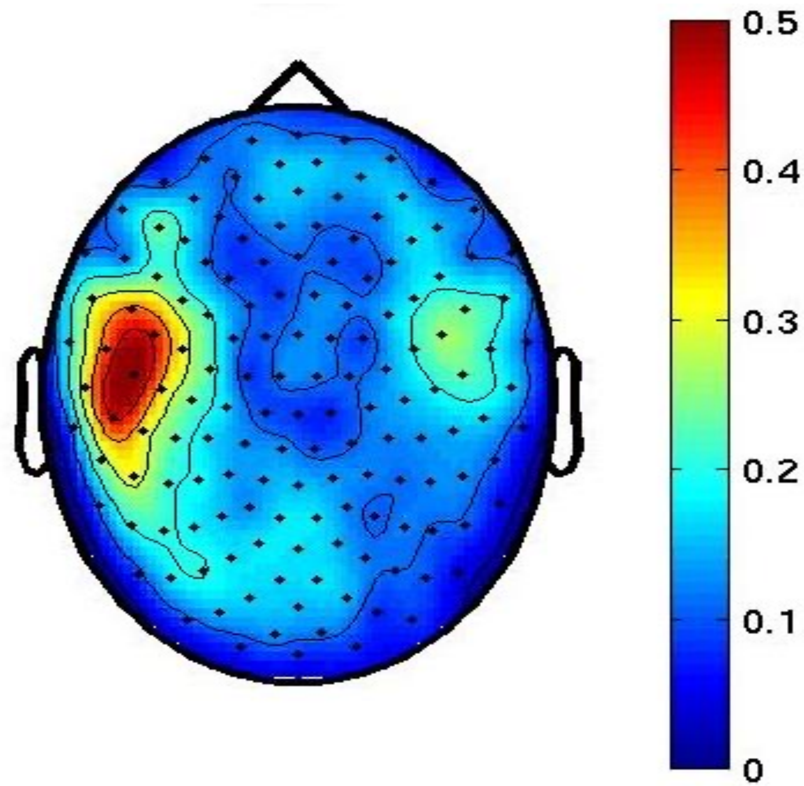
N400 response in MEG



N400 response - compared between MEG systems

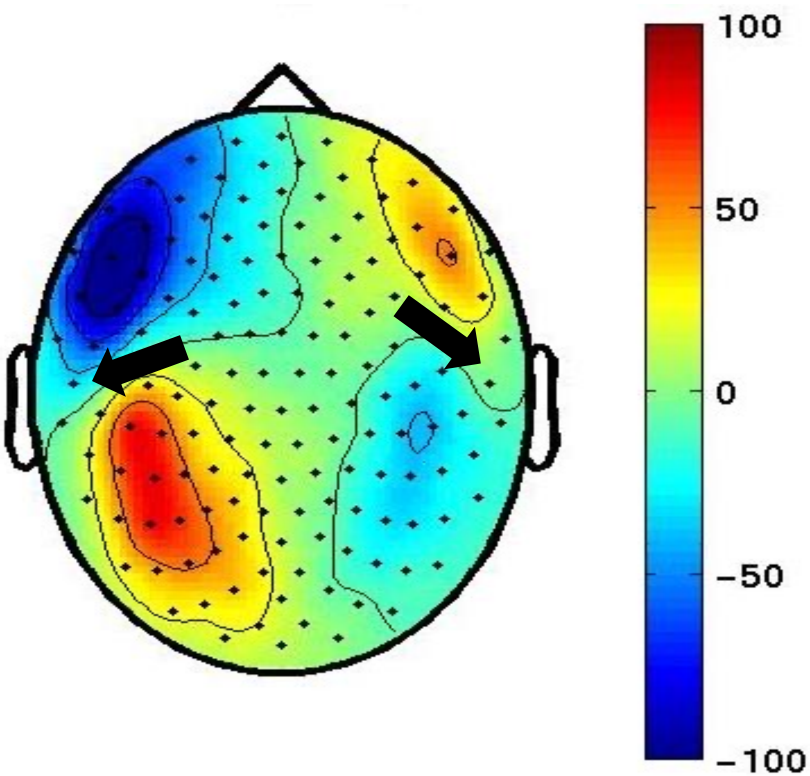


axial magnetic field

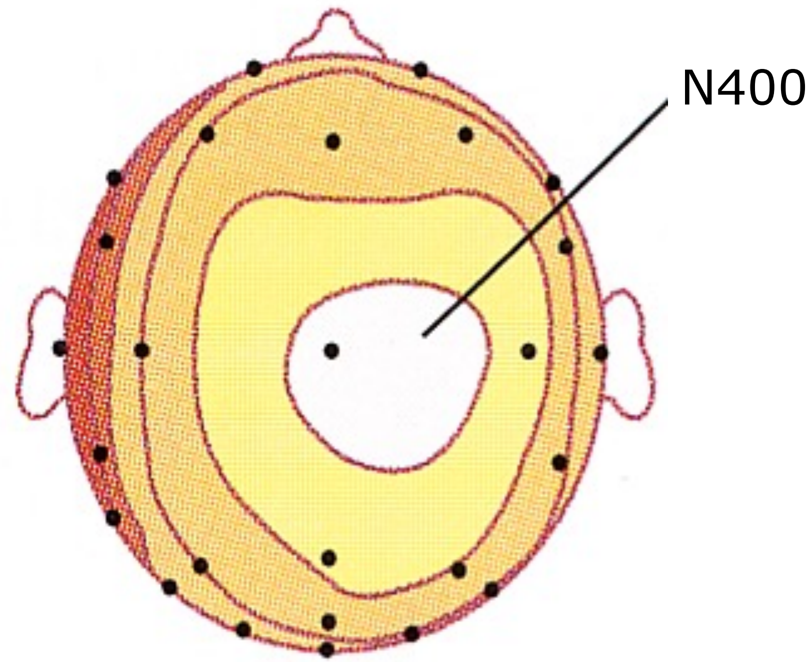


planar gradient

N400 response - compared between MEG and EEG



axial magnetic field



potential distribution

LETTER

doi:10.1038/nature26147

Moving magnetoencephalography towards real-world applications with a wearable system

Elena Boto[1]*, Niall Holmes[1]*, James Leggett[1]*, Gillian Roberts[1]*, Vishal Shah[2], Sofie S. Meyer[3,4], Leonardo Duque Muñoz[3], Karen J. Mullinger[1,5], Tim M. Tierney[3], Sven Bestmann[3,6], Gareth R. Barnes[3]§, Richard Bowtell[1]§ & Matthew J. Brookes[1]§

Imaging human brain function with techniques such as magnetoencephalography[1] typically requires a subject to perform tasks while their head remains still within a restrictive scanner. This artificial environment makes the technique inaccessible to many people, and limits the experimental questions that can be addressed. For example, it has been difficult to apply neuroimaging to investigation of the neural substrates of cognitive development in babies and children, or to study processes in adults that require unconstrained head movement (such as spatial navigation). Here we describe a magnetoencephalography system that can be worn like a helmet, allowing free and natural movement during scanning. This is possible owing to the integration of quantum sensors[2,3], which do not rely on superconducting technology, with a system for nulling background magnetic fields. We demonstrate human electrophysiological measurement at millisecond resolution while subjects make natural movements, including head nodding, stretching, drinking and playing a ball game. Our results compare well to those of the current state-of-the-art, even when subjects make large head movements. The system opens up new possibilities for scanning any subject or patient group, with myriad applications such as characterization of the neurodevelopmental connectome, imaging subjects moving naturally in a virtual environment and investigating the pathophysiology of movement disorders.

Magnetoencephalography[1] (MEG) allows direct imaging of human brain electrophysiology by measurement of magnetic fields generated at the scalp by neural currents. Mathematical analysis of those fields enables the generation of 3D images that show the formation and dissolution of brain networks in real time. MEG measurements of brain activity are currently made using an array of superconducting sensors placed around the head[1,4]. These cryogenically cooled sensors have femtotesla (fT) sensitivity, which is needed to detect the weak magnetic fields produced by the brain. Unfortunately, the requirement

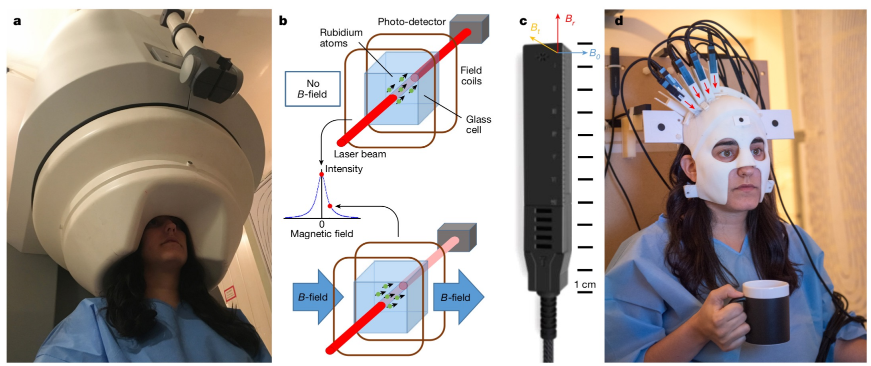
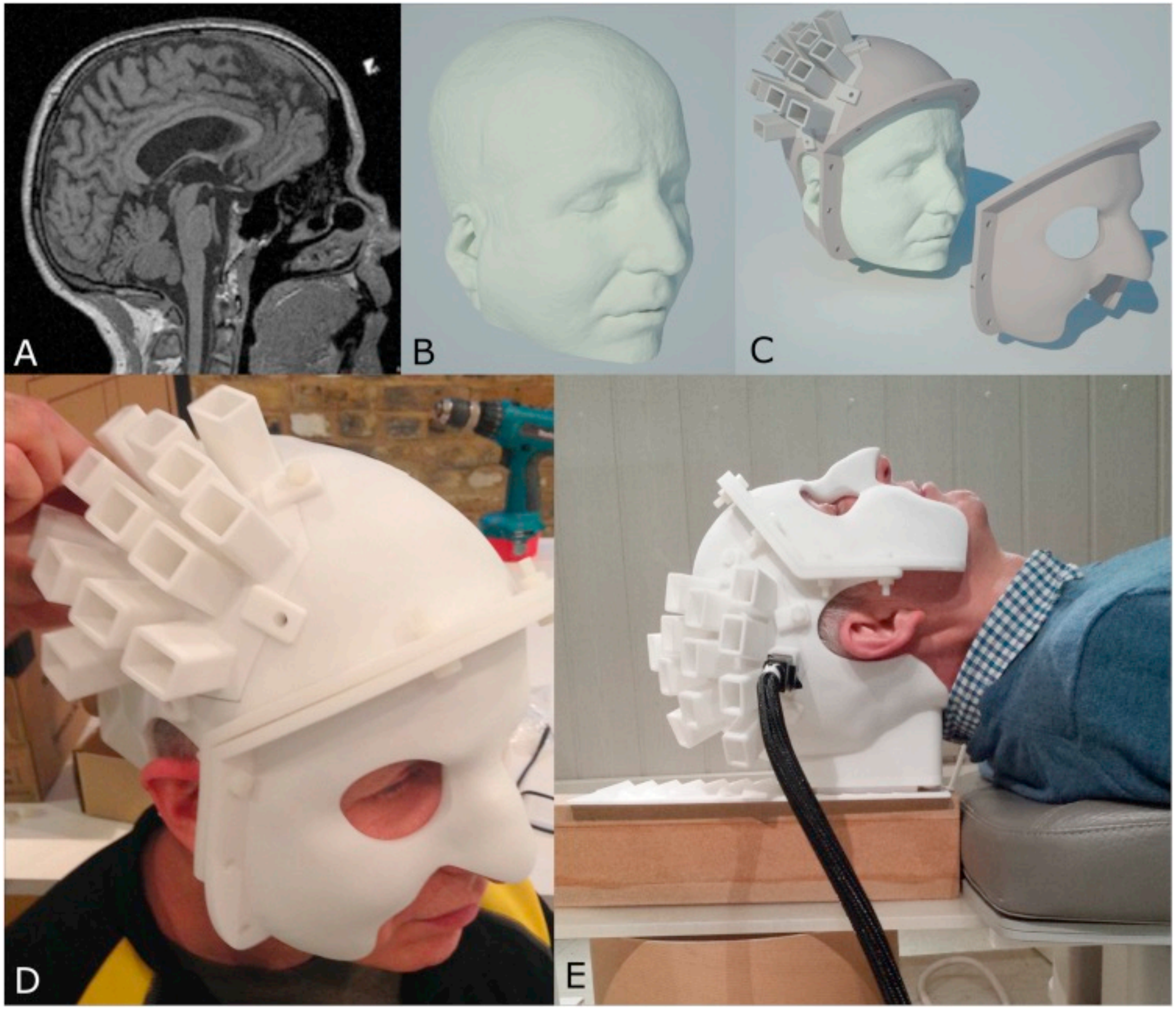


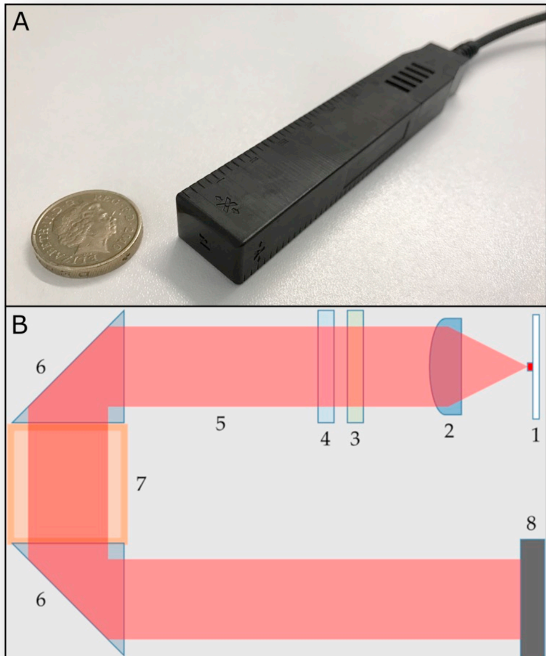
Figure 1 | A new generation of MEG system. a, A conventional 275-channel cryogenic MEG system. Weighing about 450 kg, the system is fixed and cumbersome and subjects must remain still relative to the fixed sensor array. **b**, Schematic illustration of zero-field resonance in an OPM sensor. Top, operation in zero-field; bottom, Larmor precession when an external field (*B*-field) impinges on the cell and the transmitted light intensity is reduced. **c**, A commercial OPM sensor made by QuSpin. The geometry used is illustrated by the coloured axes where B_r is the radial field component, B_t the tangential field component and B_o the direction along which the laser beam is oriented. **d**, Our prototype OPM-MEG system helmet. The helmet weighs 905 g and is customized so that the sensors (which in this prototype cover only the right sensorimotor cortex) are directly adjacent to the scalp surface. The subject is free to move their head. The measured radial field direction for the sensors is illustrated by the red arrows.

[1]Sir Peter Mansfield Imaging Centre, School of Physics and Astronomy, University of Nottingham, University Park, Nottingham NG7 2RD, UK. [2]QuSpin Inc., 331 South 104th Street, Suite 130, Louisville, Colorado 80027, USA. [3]Wellcome Centre for Human Neuroimaging, UCL Institute of Neurology, University College London, 12 Queen Square, London WC1N 3BG, UK. [4]Institute of Cognitive Neuroscience, University College London, 17–19 Queen Square, London WC1N 3AZ, UK. [5]Centre for Human Brain Health, School of Psychology, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK. [6]Sobell Department for Motor Neuroscience and Movement Disorders, UCL Institute of Neurology, University College London, Queen Square House, Queen Square, London WC1N 3BG, UK.
*These authors contributed equally to this work.
§These authors jointly supervised this work.

https://www.nature.com/articles/nature26147



https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5562927/



https://www.cercamagnetics.com
https://www.kernel.com
https://fieldlineinc.com
https://quspin.com

https://youtu.be/6QizMOa4ZJM

Talk outline

What kind of signals are generated in the brain

How do we record those signals

Analyzing those signals with FieldTrip

Background on the FieldTrip toolbox

M/EEG signal characteristics considered during analysis

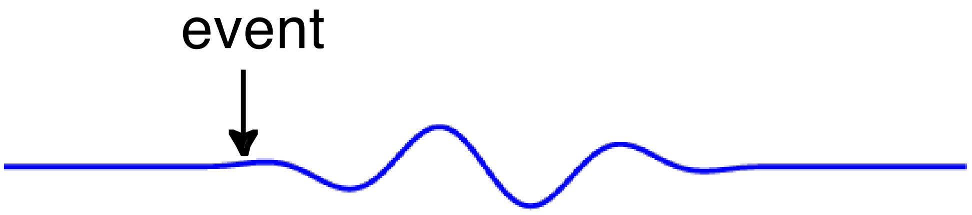
timecourse of activity
-> ERP

spectral characteristics
-> power spectrum

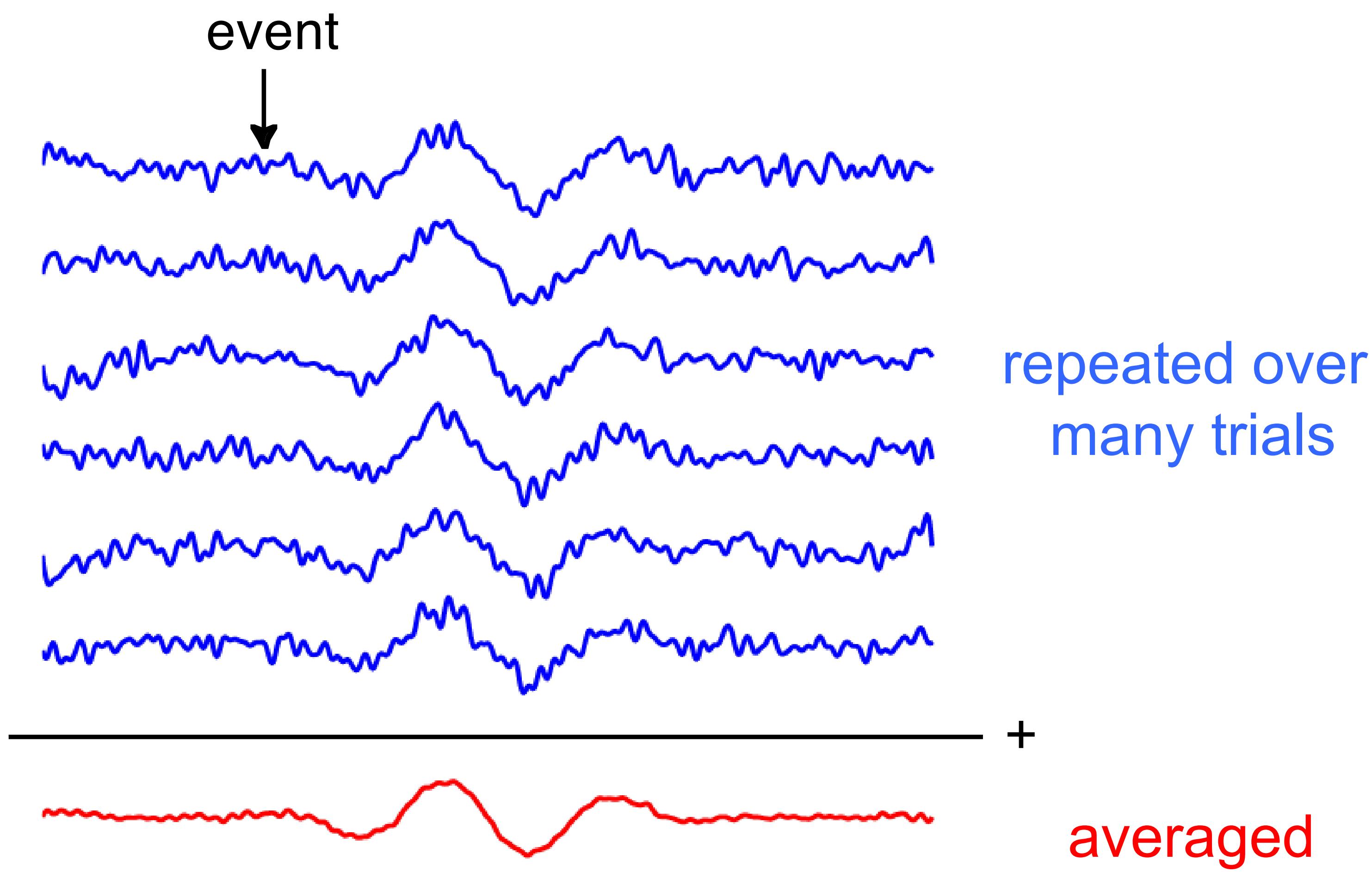
temporal changes in power
-> time-frequency response (TFR)

spatial distribution of activity over the head
-> source reconstruction

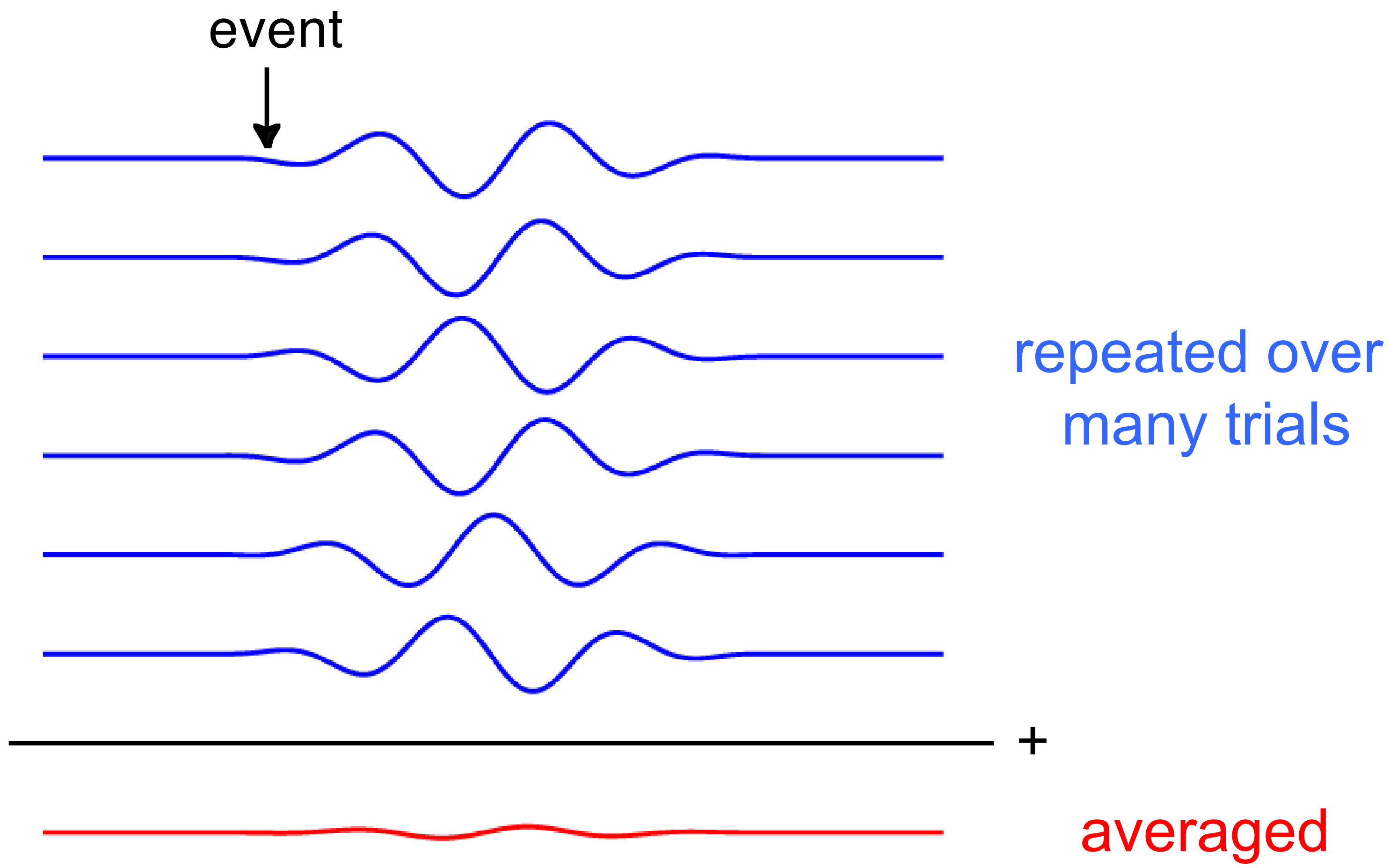
Evoked activity



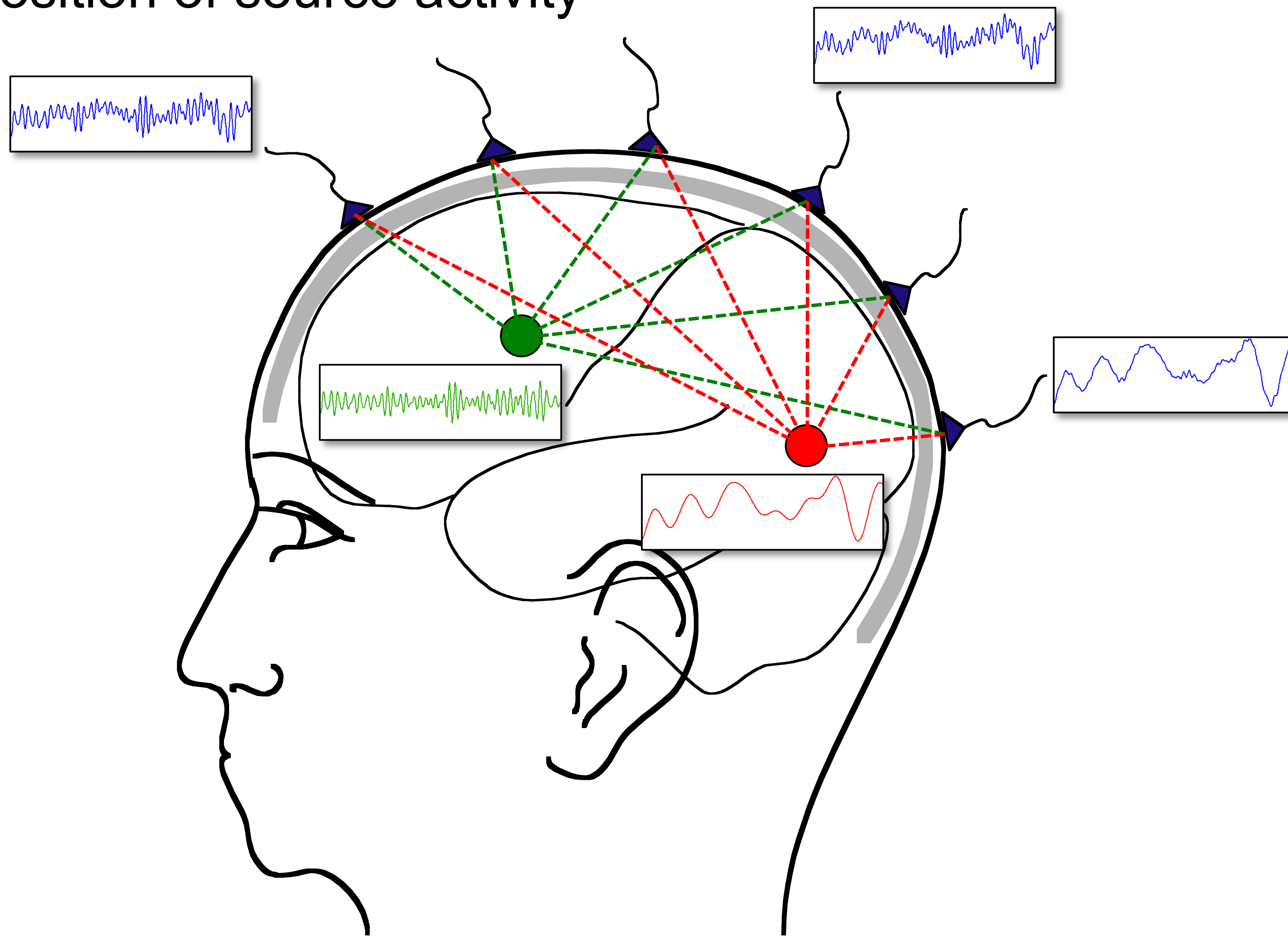
Evoked activity



Induced activity



Superposition of source activity



Separating activity of sources

- Use the temporal aspects of the data at the channel level
  - ERF latencies
  - ERF difference waves
  - Filtering the time-series
  - Spectral decomposition

- Use the spatial aspects of the data
  - Volume conduction model of head
  - Estimate source model parameters

Talk outline

What kind of signals are generated in the brain

How do we record those signals

Analyzing those signals with FieldTrip

Background on the FieldTrip toolbox

Some FieldTrip basics

dataout = functionname(cfg, datain, …)

functionname(

dataout = func

the “cfg” argument is a configuration structure, e.g.

cfg.channel = {‘C3’, C4’, ‘F3’, ‘F4’}
cfg.foilim = [1 70]

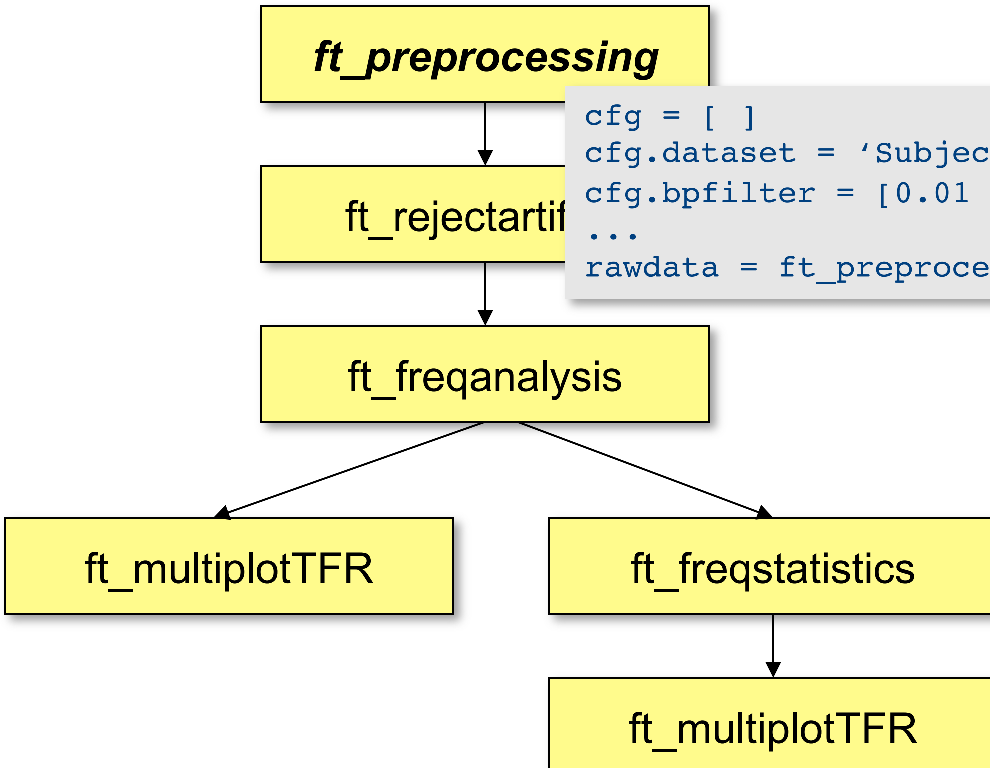
FieldTrip v.s. default Matlab

```
dataout = functionname(cfg, datain, …)

        cfg.key1 = value1
        cfg.key2 = value2

dataout = functionname(datain, key1, value1, …)
```

Using functions in an analysis protocol

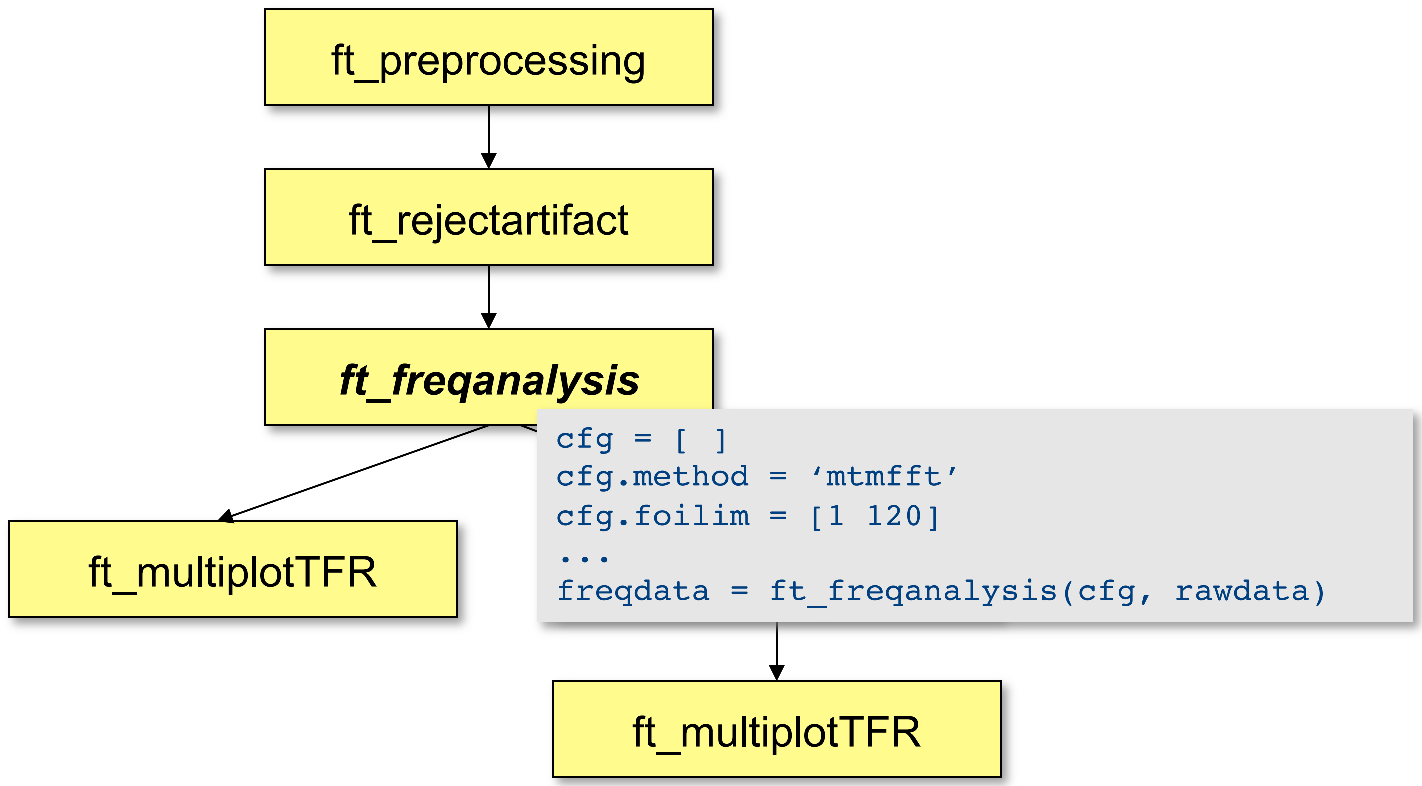

```
FT_PREPROCESSING reads MEG and/or EEG data according to user-specified
trials and applies several user-specified preprocessing steps to the
signals.

Use as
  [data] = ft_preprocessing(cfg)
or
  [data] = ft_preprocessing(cfg, data)

The first input argument "cfg" is the configuration structure, which
contains all details for the dataset filenames, trials and the
preprocessing options. You can only do preprocessing after defining the
segments of data to be read from the file (i.e. the trials), which is for
example done based on the occurence of a trigger in the data.

...
```

Using functions in an analysis protocol




```
cfg = [ ]
cfg.dataset = 'Subject01.ds'
cfg.bpfilter = [0.01 150]
...
rawdata = ft_preprocessing(cfg)
```

Using functions in an analysis protocol



Raw data structure

```
rawData =
       label: {151x1 cell}
       trial: {1x80 cell}
        time: {1x80 cell}
     fsample: 300
         hdr: [1x1 struct]
         cfg: [1x1 struct]
```

Event related response

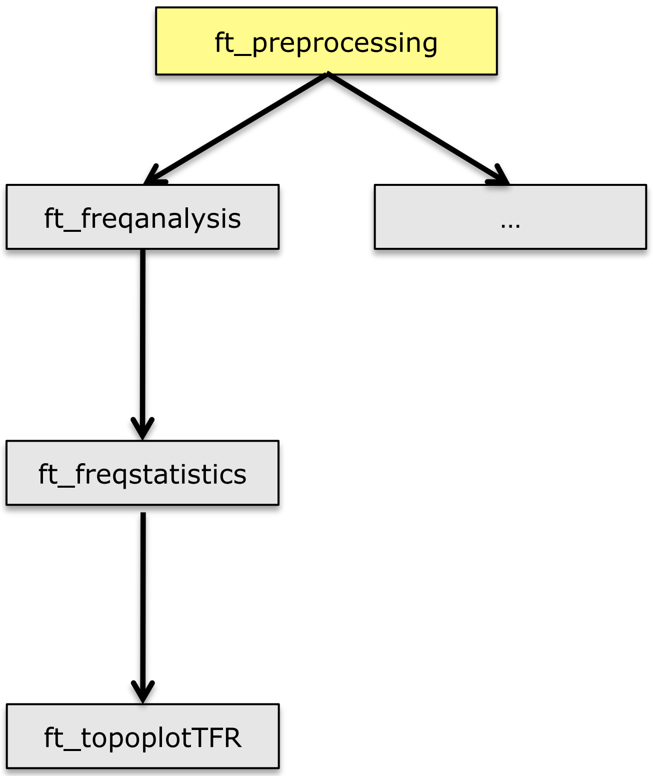
```
timelockData =
           label: {151x1 cell}
             avg: [151x900 double]
             var: [151x900 double]
            time: [1x900 double]
          dimord: 'chan_time'
             cfg: [1x1 struct]
```

Example use in scripts

```
cfg = []
cfg.dataset = 'Subject01.ds'
cfg.bpfilter = [0.01 150]
...
rawdata = ft_preprocessing(cfg)
```

```
cfg = []
cfg.method = 'mtmfft'
cfg.foilim = [1 120]
...
freqdata = ft_freqanalysis(cfg, rawdata)
```

```
cfg = []
cfg.method = 'montecarlo'
cfg.statistic = 'indepsamplesT'
cfg.design = [1 2 1 2 2 1 2 1 1 2 ... ]
...
freqstat = ft_freqstatistics(cfg, freqdata)
```

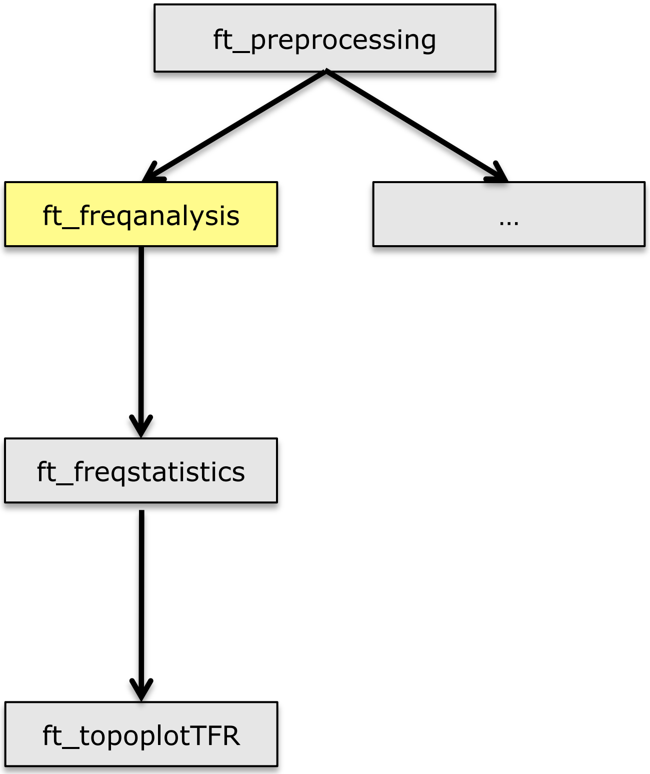


Example use in scripts

```
cfg = []
cfg.dataset = 'Subject01.ds'
cfg.bpfilter = [0.01 150]
...
rawdata = ft_preprocessing(cfg)
```

```
cfg = []
cfg.method = 'mtmfft'
cfg.foilim = [1 120]
...
freqdata = ft_freqanalysis(cfg, rawdata)
```

```
cfg = []
cfg.method = 'montecarlo'
cfg.statistic = 'indepsamplesT'
cfg.design = [1 2 1 2 2 1 2 1 1 2 ... ]
...
freqstat = ft_freqstatistics(cfg, freqdata)
```

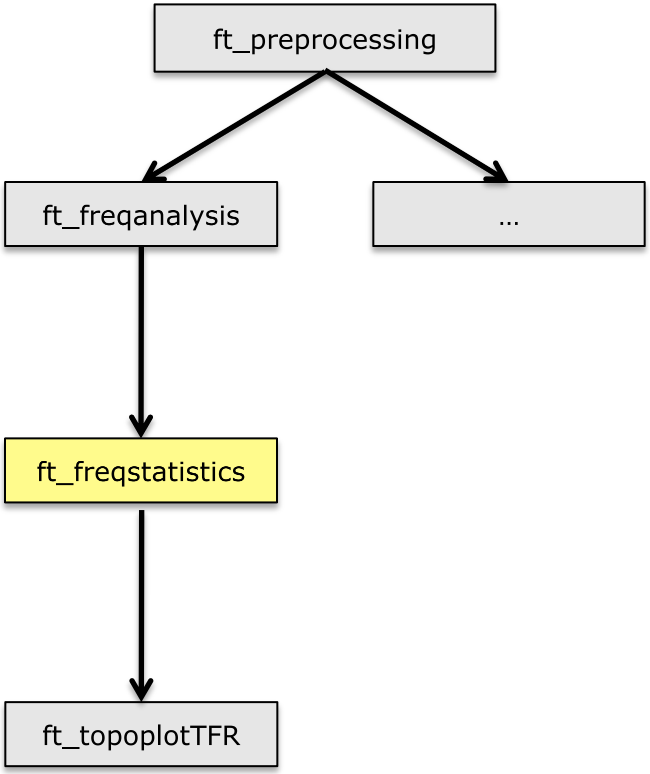


Example use in scripts

```
cfg = []
cfg.dataset = 'Subject01.ds'
cfg.bpfilter = [0.01 150]
...
rawdata = ft_preprocessing(cfg)
```

```
cfg = []
cfg.method = 'mtmfft'
cfg.foilim = [1 120]
...
freqdata = ft_freqanalysis(cfg, rawdata)
```

```
cfg = []
cfg.method = 'montecarlo'
cfg.statistic = 'indepsamplesT'
cfg.design = [1 2 1 2 2 1 2 1 1 2 ... ]
...
freqstat = ft_freqstatistics(cfg, freqdata)
```

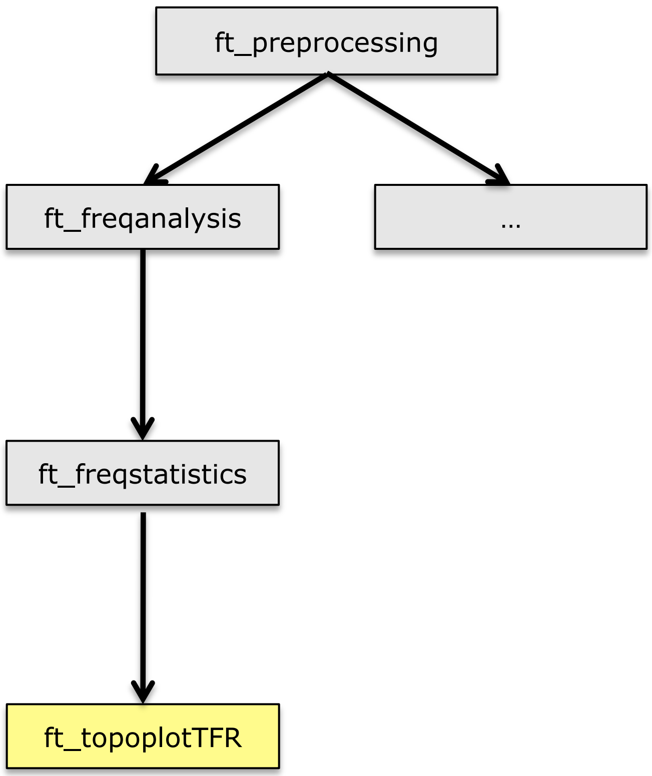


Example use in scripts

```
cfg = []
cfg.dataset = 'Subject01.ds'
cfg.bpfilter = [0.01 150]
...
rawdata = ft_preprocessing(cfg)
```

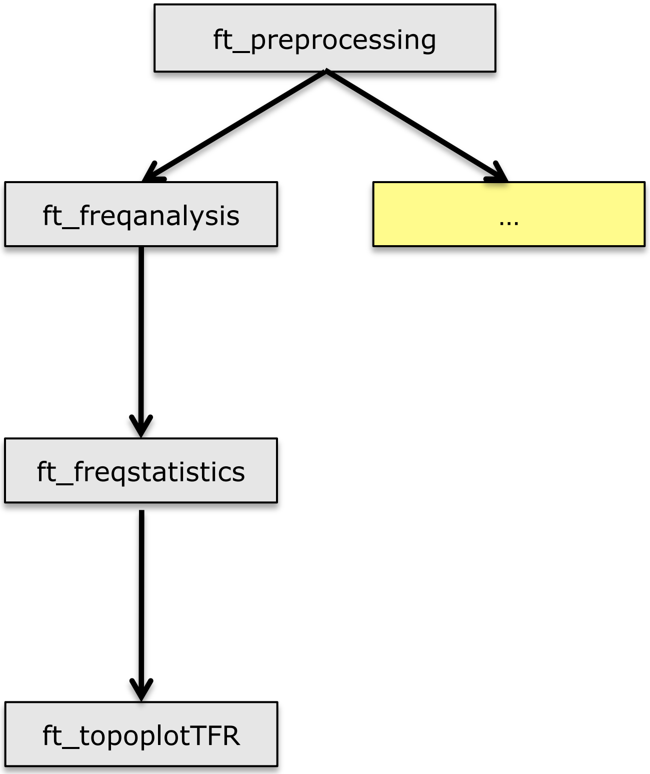
```
cfg = []
cfg.method = 'mtmfft'
cfg.foilim = [1 120]
...
freqdata = ft_freqanalysis(cfg, rawdata)
```

```
cfg = []
cfg.method = 'montecarlo'
cfg.statistic = 'indepsamplesT'
cfg.design = [1 2 1 2 2 1 2 1 1 2 ... ]
...
freqstat = ft_freqstatistics(cfg, freqdata)
```



Example use in scripts

```
cfg = []
cfg.dataset = 'Subject01.ds'
cfg.bpfilter = [0.01 150]
...
rawdata = ft_preprocessing(cfg)

cfg = []
cfg.method = 'mtmfft'
cfg.foilim = [1 120]
...
freqdata = ft_freqanalysis(cfg, rawdata)

cfg = []
cfg.method = 'montecarlo'
cfg.statistic = 'indepsamplesT'
cfg.design = [1 2 1 2 2 1 2 1 1 2 ... ]
...
freqstat = ft_freqstatistics(cfg, freqdata)
```



Example use in scripts

```
subj = {'S01.ds', 'S02.ds', …}
trig = [1 3 7 9]

for s=1:nsubj
for c=1:ncond

  cfg = []
  cfg.dataset  = subj{s}
  cfg.trigger  = trig(c)
  rawdata{s,c} = ft_preprocessing(cfg)

  cfg = []
  cfg.method    = 'mtmfft'
  cfg.foilim    = [1 120]
  freqdata{s,c} = ft_freqanalysis(cfg, rawdata{s,c})

end
end
```

Example use in scripts

```
subj = {'S01.ds', 'S02.ds', …}
trig = [1 3 7 9]

for s=1:nsubj
for c=1:ncond

  cfg = []
  cfg.dataset  = subj{s}
  cfg.trigger  = trig(c)
  rawdata      = ft_preprocessing(cfg)


  filename = sprintf('raw%s_%d.mat', subj{s}, trig(c));
  save(filename, 'rawdata')


end
end
```

Example use in distributed computing

```
subj = {'S01.ds', 'S02.ds', …}
trig = [1 3 7 9]

for s=1:nsubj
for c=1:ncond

  cfgA{s,c} = []
  cfgA{s,c}.dataset    = subj{s}
  cfgA{s,c}.trigger    = trig(c)
  cfgA{s,c}.outputfile = sprintf('raw%s_%d.mat', subj{s}, trig(c))

  cfgB{s,c} = []
  cfgB{s,c}.dataset    = subj{s}
  cfgB{s,c}.trigger    = trig(c)
  cfgB{s,c}.inputfile  = sprintf('raw%s_%d.mat', subj{s}, trig(c));
  cfgB{s,c}.outputfile = sprintf('freq%s_%d.mat', subj{s}, trig(c));

end
end

dfeval(@ft_preprocessing, cfgA)
dfeval(@ft_freqanalysis,  cfgB)
```

Example use in distributed computing

```
subj = {‘S01.ds’, ‘S02.ds’, …}
trig = [1 3 7 9]

for s=1:nsubj
for c=1:ncond

  cfgA{s,c} = []
  cfgA{s,c}.dataset    = subj{s}
  cfgA{s,c}.trigger    = trig(c)
  cfgA{s,c}.outputfile = sprintf(‘raw%s_%d.mat’, subj{s}, trig(c))

  cfgB{s,c} = []
  cfgB{s,c}.dataset    = subj{s}
  cfgB{s,c}.trigger    = trig(c)
  cfgB{s,c}.inputfile  = sprintf(‘raw%s_%d.mat’, subj{s}, trig(c));
  cfgB{s,c}.outputfile = sprintf(‘freq%s_%d.mat’, subj{s}, trig(c));

end
end

qsubcellfun(@ft_preprocessing, cfgA)
qsubcellfun(@ft_freqanalysis,  cfgB)
```

Talk outline

What kind of signals are generated in the brain

How do we record those signals

Analyzing those signals with FieldTrip

Background on the FieldTrip toolbox

FieldTrip is a toolbox

- the data and the separate functions are in your hands
- the scripts depend on the data properties, your computer and on your programming skills and style
- scripts correspond to analysis protocols
  - scripts can be reviewed by supervisors
  - scripts are often shared with colleagues
  - scripts can be published/released

Finding your way around in the FieldTrip toolbox

Matlab

help functionname

edit functionname

Website

http://www.fieldtriptoolbox.org

Email discussion list

Expertise in your local group