Donders Institute
for Brain, Cognition and Behaviour

Source reconstruction using beamformers

Tzvetan Popov
MEEG toolkit 21.04.2015

Donders Institute for Brain, Cognition and Behaviour
Radboud University Nijmegen, The Netherlands

t.popov@donders.ru.nl

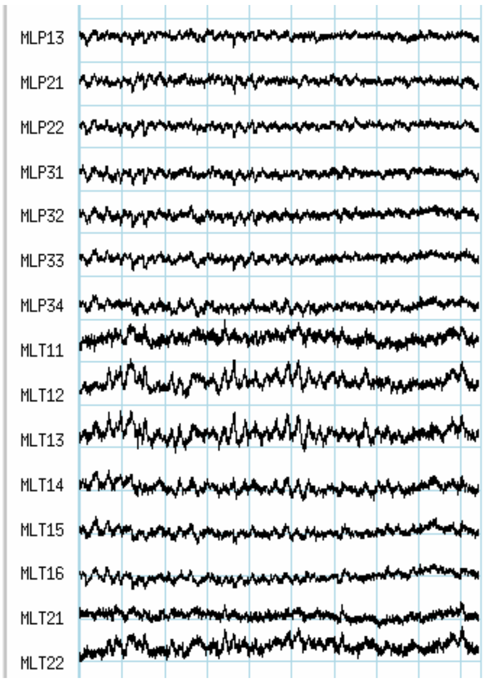
Radboud University Nijmegen

Separating sources

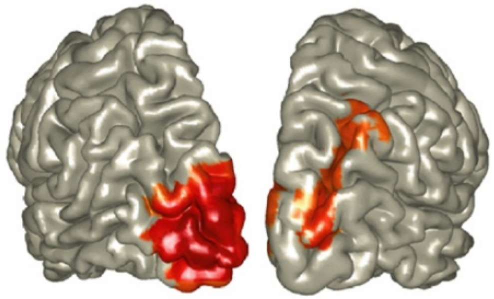
- Use the temporal aspects of the data at the channel level
  - ERF latencies
  - (ERF difference waves)
  - Filtering the time-series
  - Spectral decomposition
- Use the spatial aspects of the data

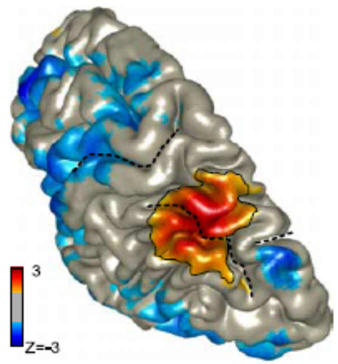
How did the brain get these red and blue blobs?

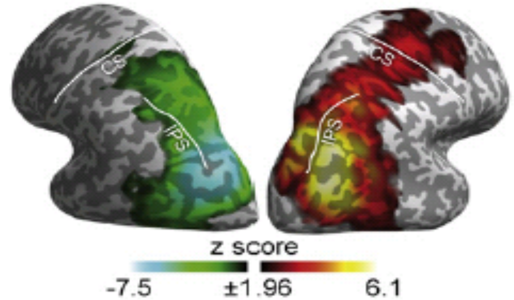
- This lecture will explain how you get from:



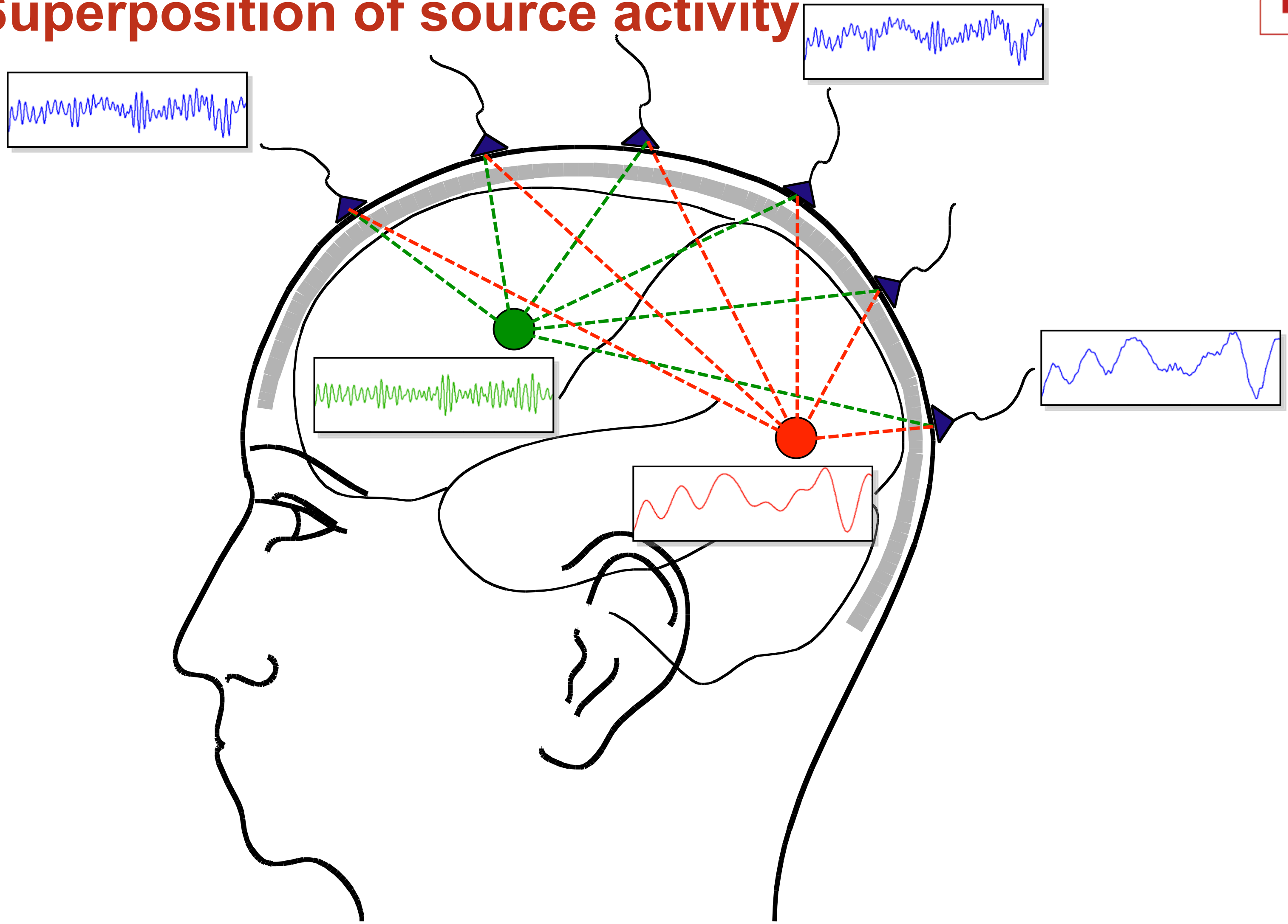
to







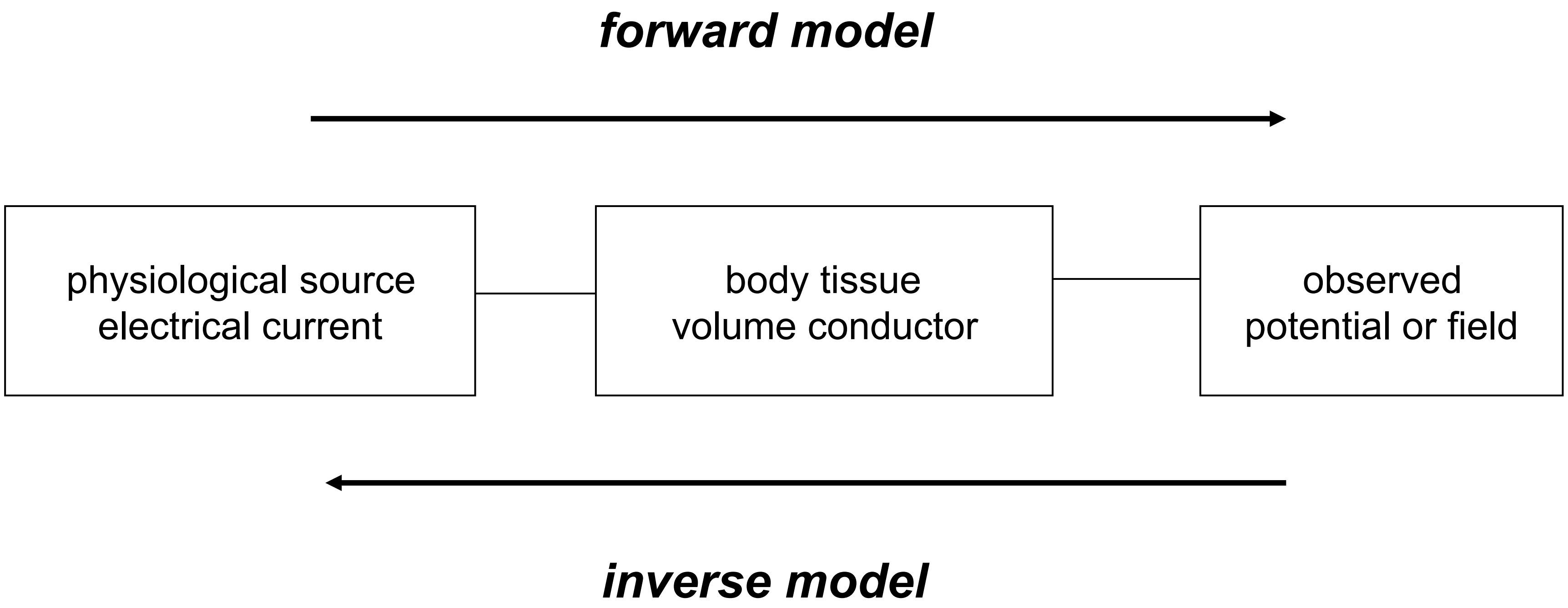
Superposition of source activity



Superposition of source activity

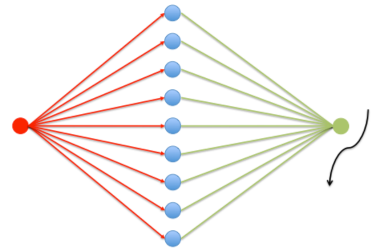
- Varying “visibility” of each source to each channel
- Timecourse of each source contributes to each channel
- The contribution of each source depends on its “visibility”
- Activity on each channel is a superposition of all source activity

Source modelling: overview

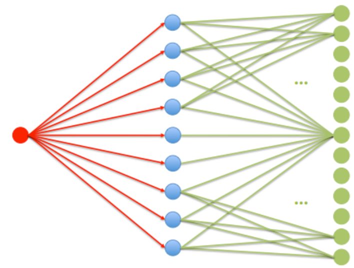


Inverse methods

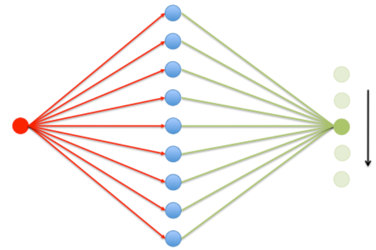
- Single and multiple dipole models
  - *Assume a small number of sources*
  - *Where (& how many) are the strongest sources?*



- Distributed dipole models
  - *Assume activity everywhere*
  - *What is the distribution of activity over the brain?*



- Spatial filtering
  - *Assume that the time-courses of different sources are uncorrelated*
  - *What is the likelihood for activity at a given brain location?*



Data model

$$X = h_1 s_1 + h_2 s_2 + \ldots + h_n s_n + \text{noise}$$

Inverse methods

- Single and multiple dipole models
  - *Minimize error between model and measured potential/field*

Data model

$$X = h_1 s_1 + h_2 s_2 + \dots + h_n s_n + \text{noise}$$

n is typically small.

$$(X - h_1 s_1 - h_2 s_2 - \dots - h_n s_n) = \text{noise}$$

Inverse methods

- Single and multiple dipole models
  - *Minimize error between model and measured potential/field*

- Distributed dipole models
  - *Perfect fit of model to the measured potential/field*
  - *Additional constraint on sources*
  - *Maximal smoothness (LORETA)*
  - *Minimum power (L2)*
  - *Minimum amplitude (L1)*

Data model

$$X = h_1 s_1 + h_2 s_2 + \dots + h_n s_n + \text{noise}$$

n is typically large. (> # channels)

$$X = h_1 s_1 + h_2 s_2 + \dots + h_n s_n + \text{noise}$$
$$H^{-1}(X - \text{noise}) = S$$

Inverse methods

- Single and multiple dipole models
  - *Minimize error between model and measured potential/field*
- Distributed dipole models
  - *Perfect fit of model to the measured potential/field*
  - *Additional constraint on sources*
  - *Maximal smoothness (LORETA)*
  - *Minimum power (L2)*
  - *Minimum amplitude (L1)*
- Spatial filtering
  - *Compute the filter output at every location*
  - *Scan the whole brain with a single dipole*
  - *Beamforming (e.g. LCMV, SAM, DICS)*
  - *Multiple Signal Classification (MUSIC)*

Data model

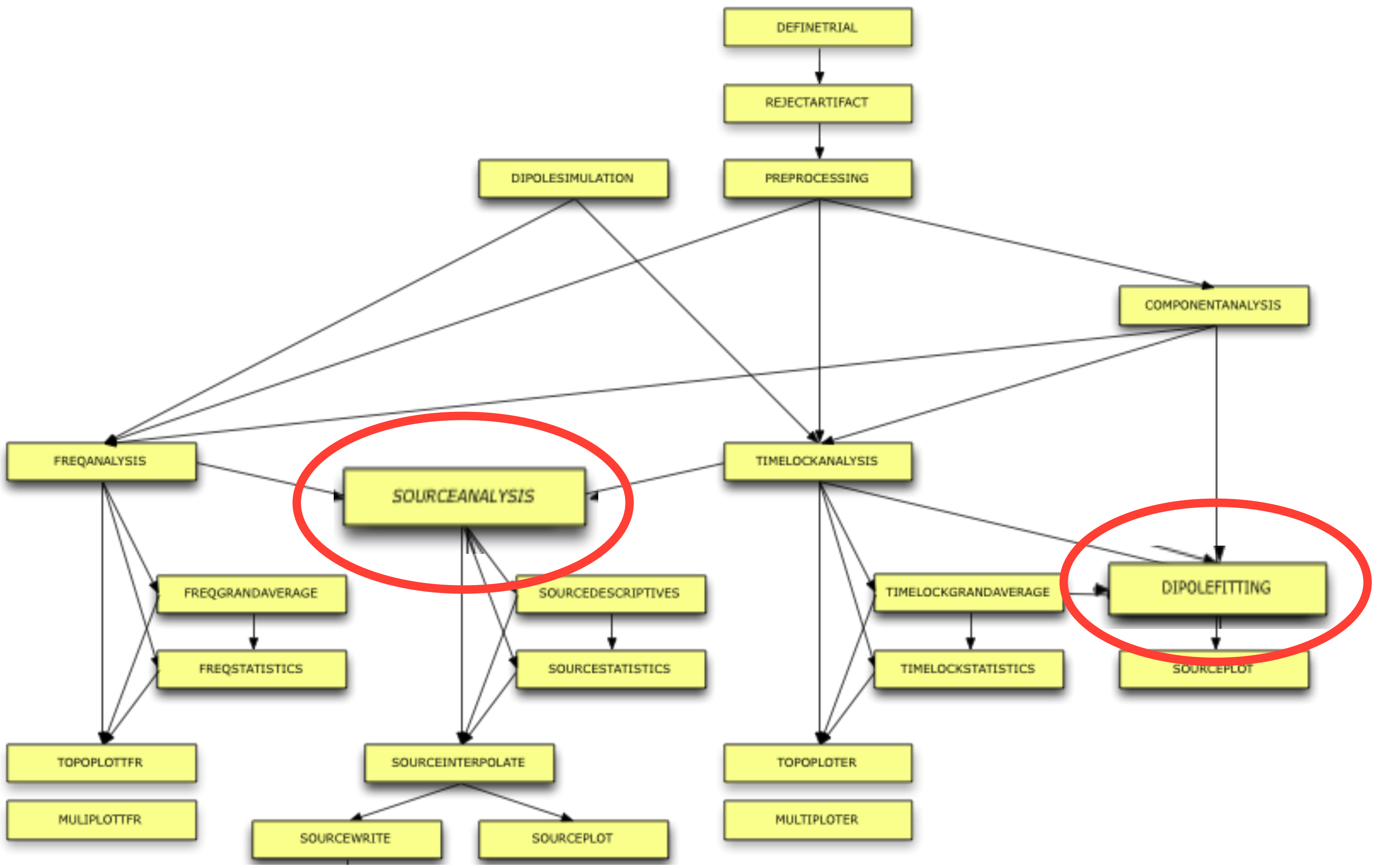
$$X = h_1 s_1 + h_2 s_2 + \ldots + h_n s_n + \text{noise}$$

$$X = (h_1 s_1 + h_2 s_2 + \ldots) + h_n s_n + (\text{noise})$$

$$X = h_n s_n + N$$

N = all activity not coming from n,
assuming N uncorrelated with s_n

Fieldtrip functions for source localization



Source localization

- Different algorithms on the market
- Several of these are implemented in FieldTrip

```
cfg   = [];
      .
source = ft_dipolefitting(cfg, data);
```

```
cfg   = [];
cfg.method = ‘mne’;
      .
source = ft_sourceanalysis(cfg, data);
```

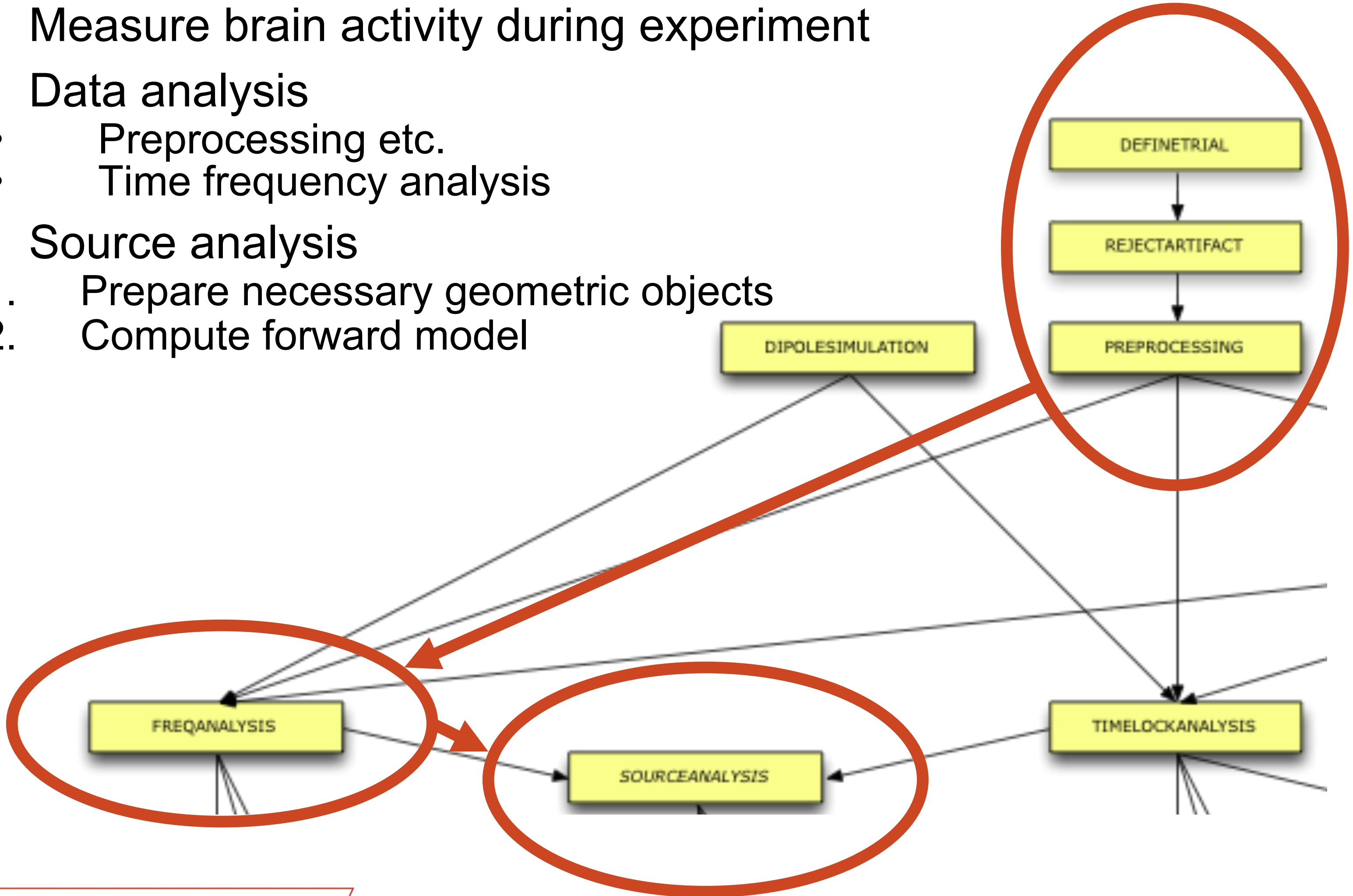
```
cfg   = [];
cfg.method = ‘lcmv’;
      .
source = ft_sourceanalysis(cfg, data);
```

```
cfg   = [];
cfg.method = ‘dics’;
      .
      .
      .
source = ft_sourceanalysis(cfg, freq);
```

Beamformers

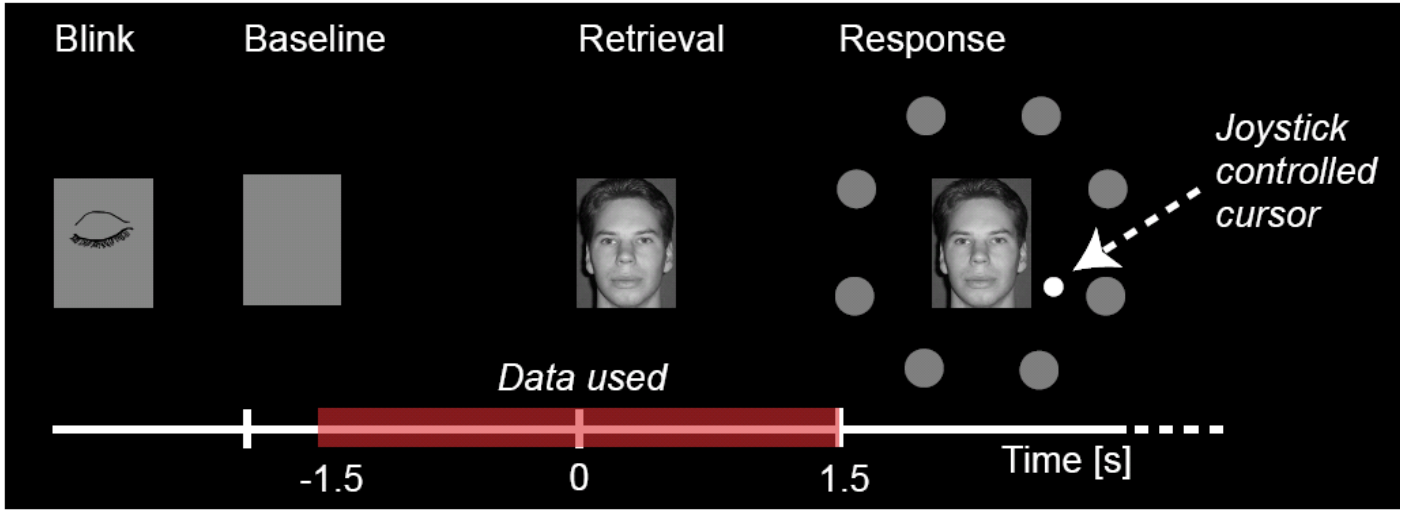
Procedure beamforming of oscillatory activity

1. Design experiment
2. Measure brain activity during experiment
3. Data analysis
   - Preprocessing etc.
   - Time frequency analysis
4. Source analysis
   1. Prepare necessary geometric objects
   2. Compute forward model



Stage 1: Design experiment

- Baseline recommendable
- Sufficient length of stationary signal
  - Delayed response
- Avoid artifacts
  - Eyeblink stimulus
  - Experiment not too long, or introduce breaks (muscle artifacts)

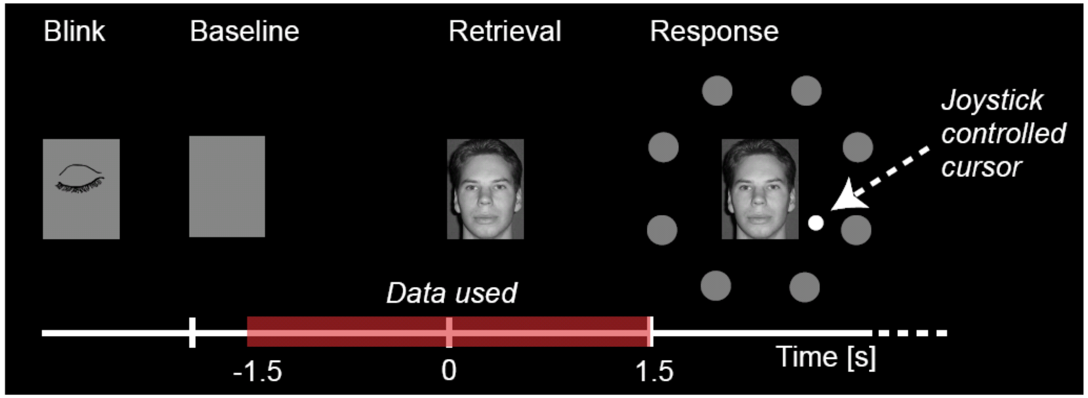


Stage 2: Measuring brain activity

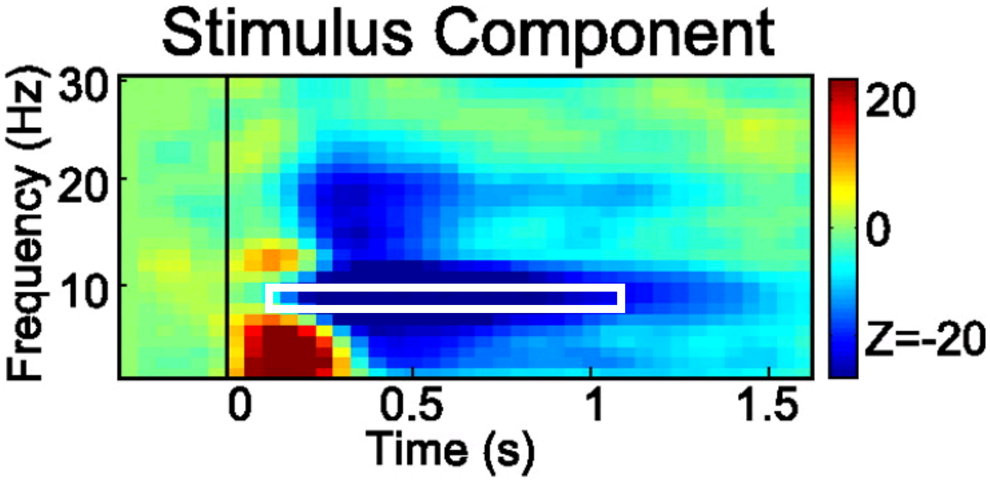
- Record EOG and ECG to remove artifacts
- Measure positions sensors/electrodes in relation to head
- Reduce head movement (MEG)
- Make anatomical MRI scan for realistic head model and optimal normalization over subjects
- Perform if applicable and possible localizer task

Stage 3: Data analysis: Preprocessing

- Data segmentation
- Artifact removal



Stage 3: Data analysis: Time frequency analysis

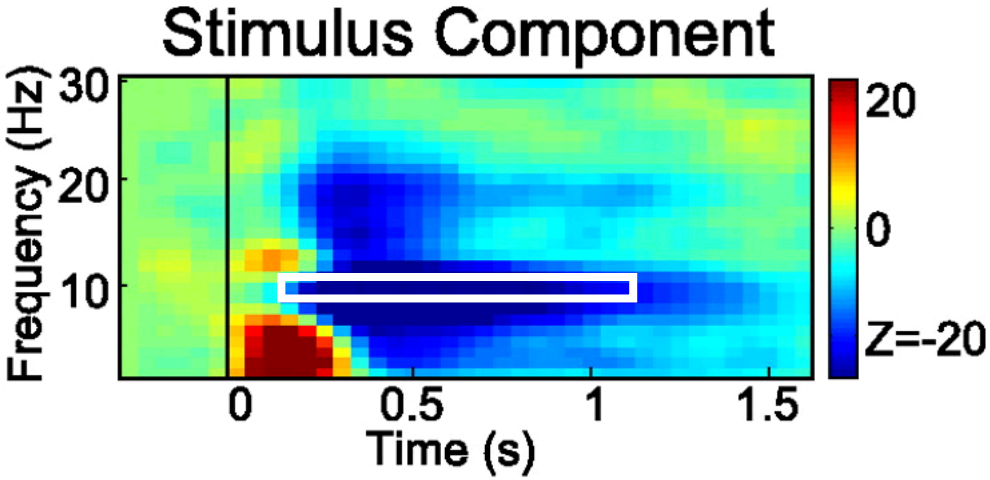


"Beam" this time-frequency tile

0.1 to 1.1 s

~10 Hz

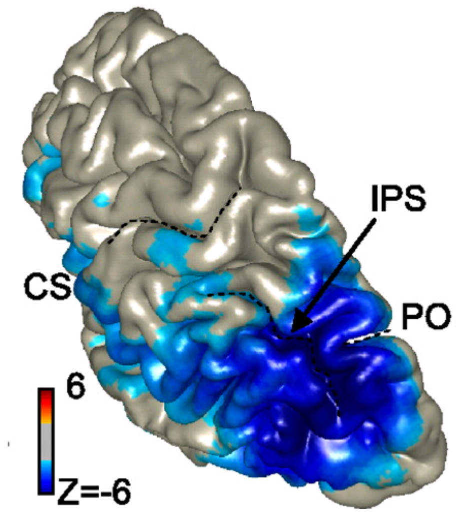
Stage 3: Data analysis: Time frequency analysis



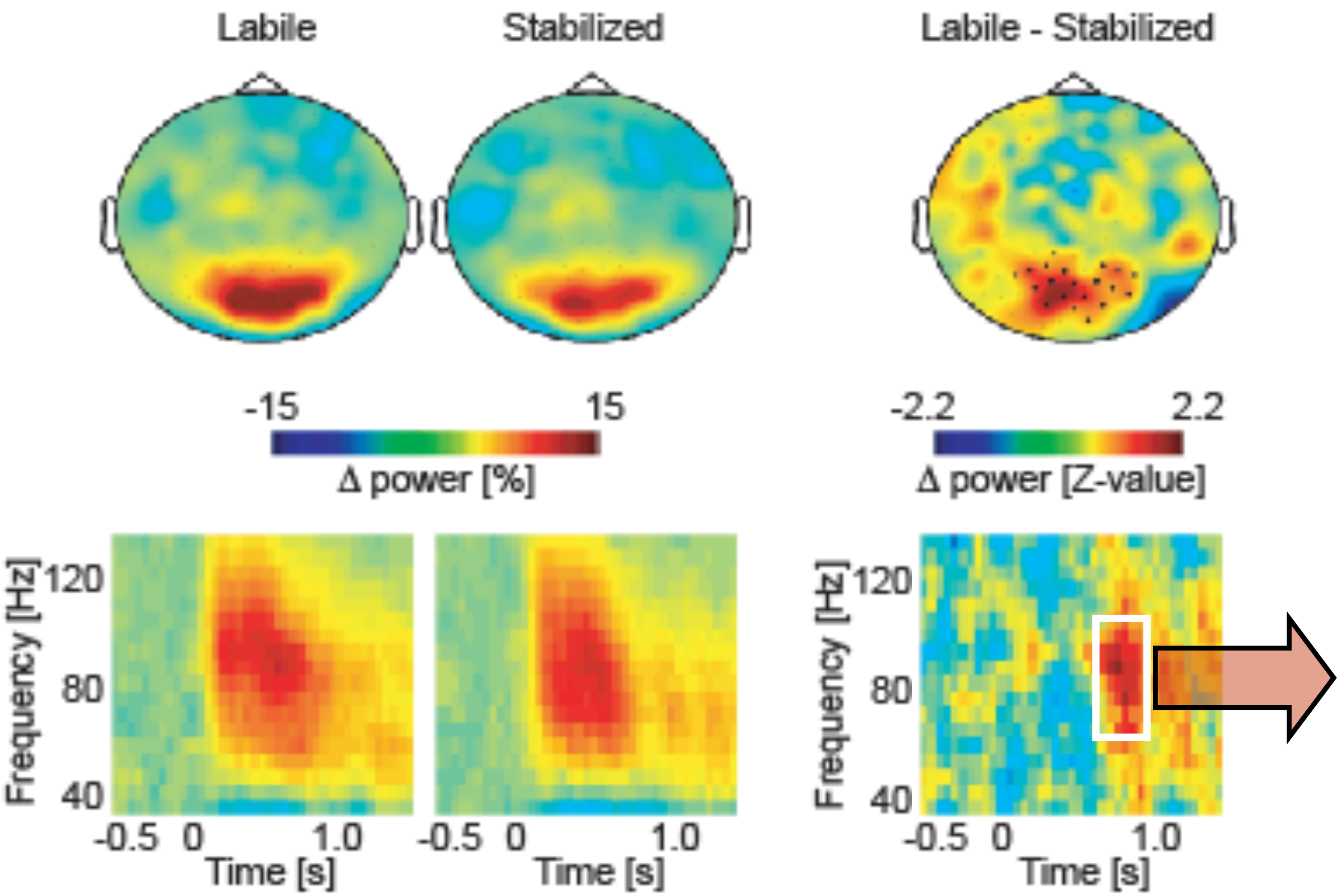
Time window of 1 second:

Frequency resolution 1 Hz

Bandwidth: 9.5 – 10.5 Hz



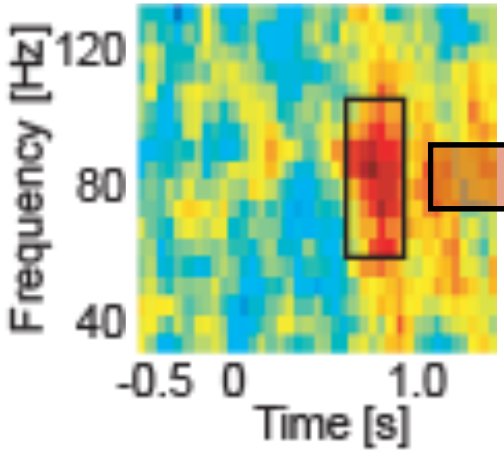
Stage 3: Data analysis: Time frequency analysis



"Beam" this time-frequency tile

0.7 to 0.9 s
70 to 90 Hz

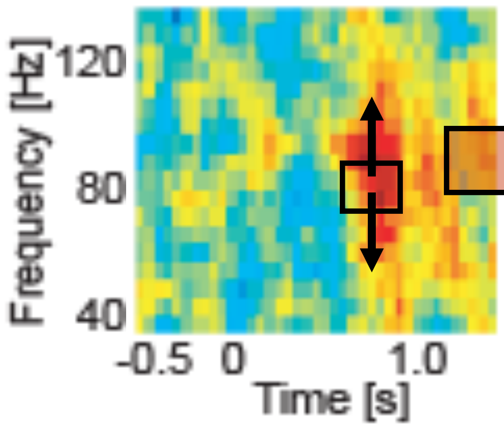
Stage 3: Data analysis: Time frequency analysis



I want:

"Beam" this time-frequency tile

0.7 to 0.9 s
70 to 90 Hz



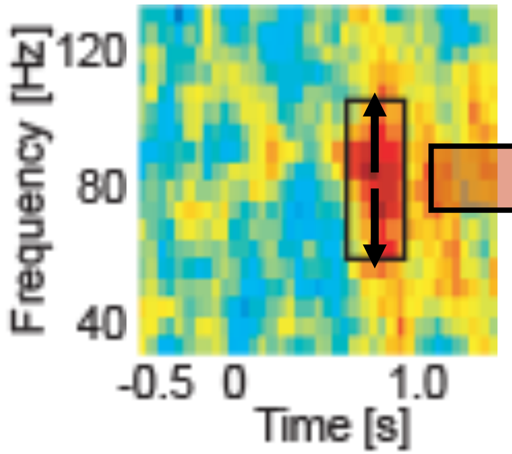
I get:

0.2 s
↓
5 Hz resolution
↓
77.5 - 82.5 Hz

Increase frequency smoothing without changing length time window: multitapers

Recap: multitapers

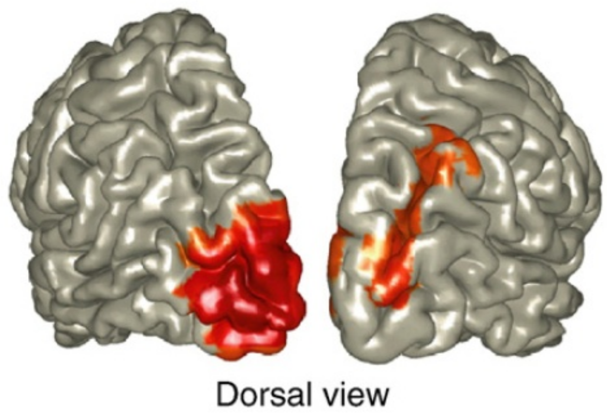
- More tapers for a given time window will result in more spectral smoothing
- Several orthogonal tapers are used for the time window, subsequently the power (and phase) is calculated for each tapered data segment and then combined.



0.2 s

5 tapers

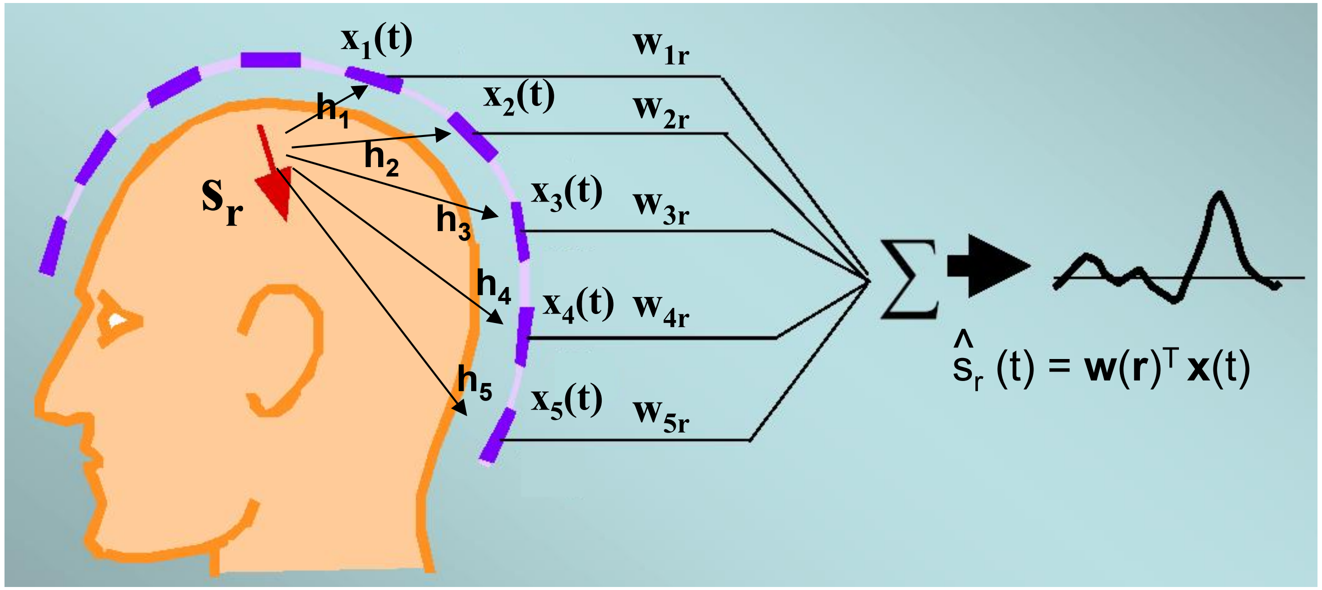
65 – 95 Hz



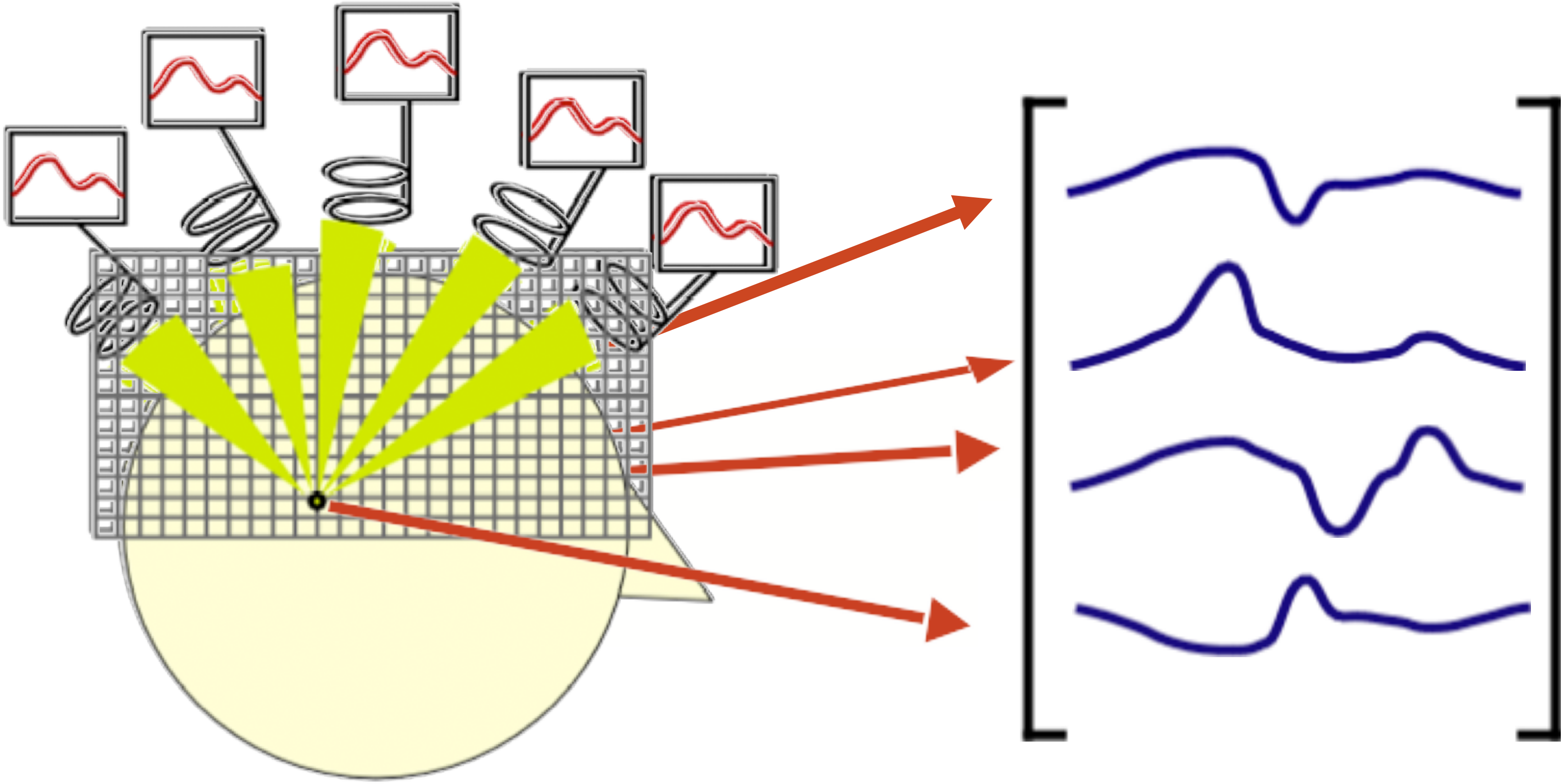

```
cfg  = [];
cfg.method = 'mtmconvol';
cfg.output = 'powandcsd';
cfg.toi     = 0.8;
cfg.foi     = 80;
cfg.t_ftimwin = 0.2;
cfg.tapsmofrq = 15;
freq = ft_freqanalysis(cfg, data);
```

Beamformer: the question

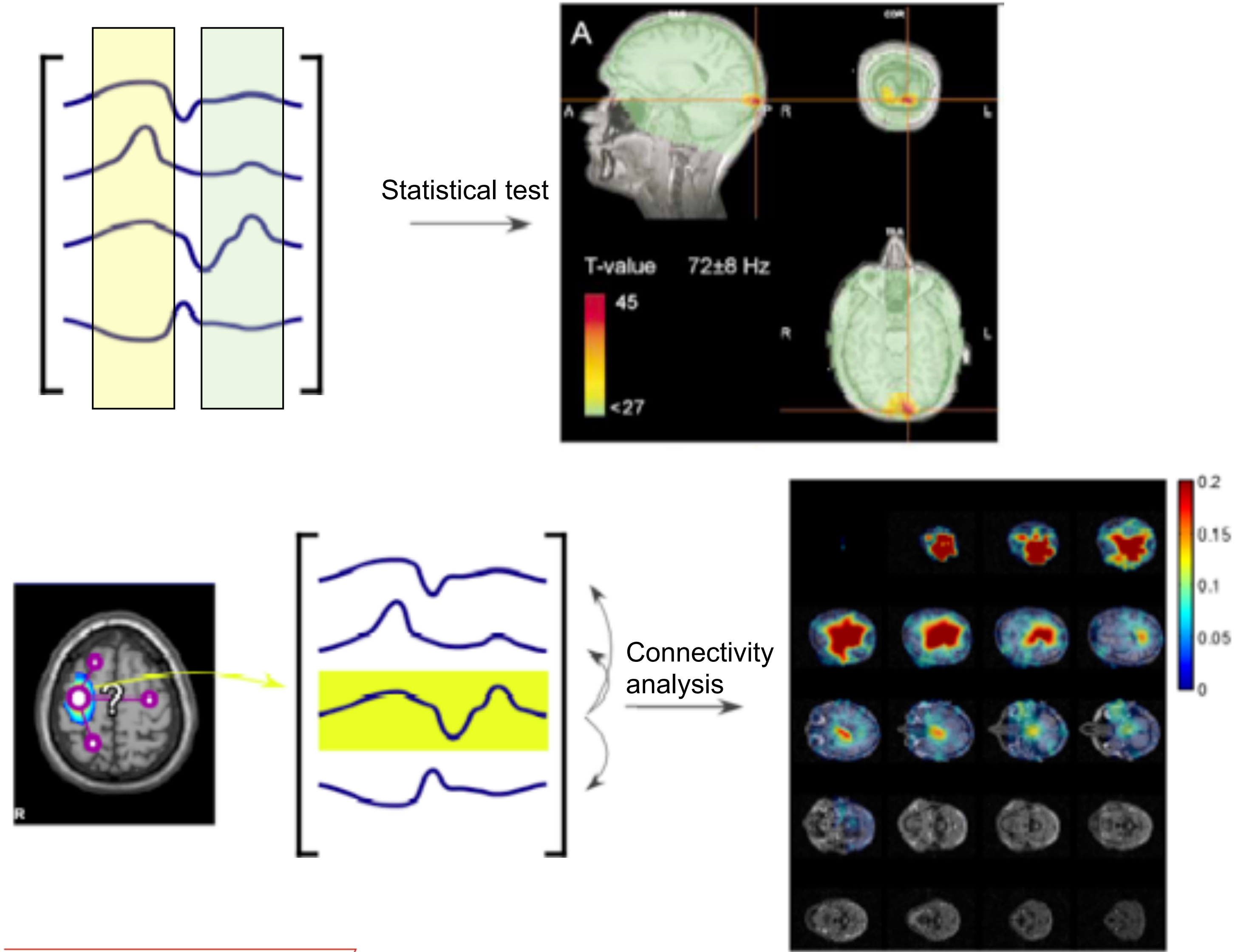
- What is the activity of a source **s**, at a location **r**, given the data **x**?
- Note: the explanation is in the time domain, because that is more intuitive
- We estimate **s** with a spatial filter **w**



Beamformers: the concept



And then: creating the blobs

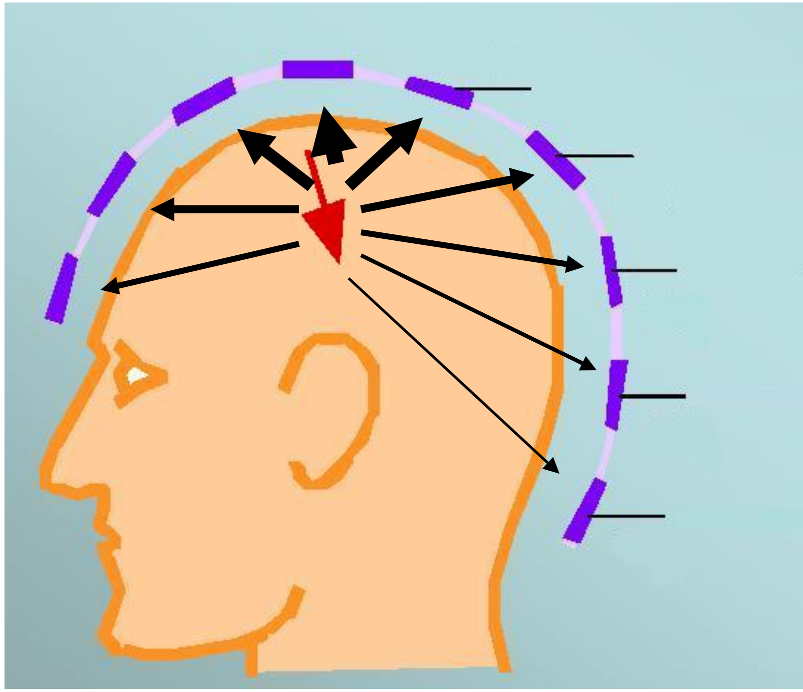


Beamformer ingredients (how to compute w)

- **Forward model**
  - Predict the data from a source at a given location
  - Ensures specificity in space (spotlight)
- **Experimental data**
  - Experimental contrast / active versus baseline
  - Ensures selectivity for effect of interest

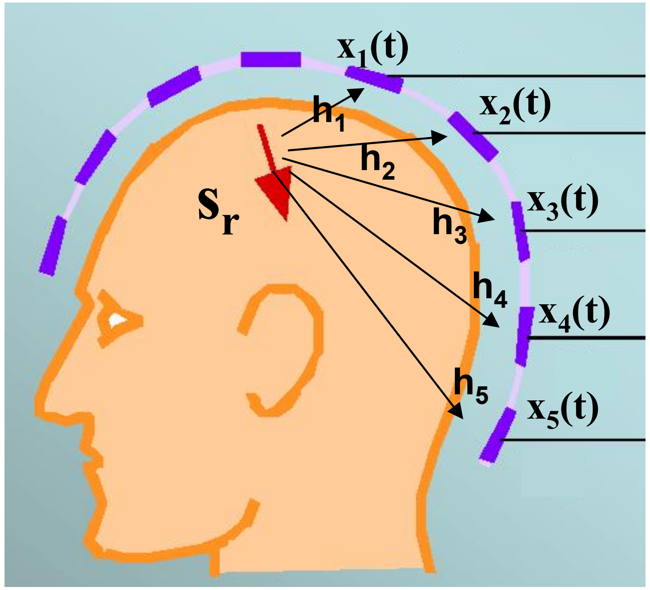
Beamformer ingredients: forward model

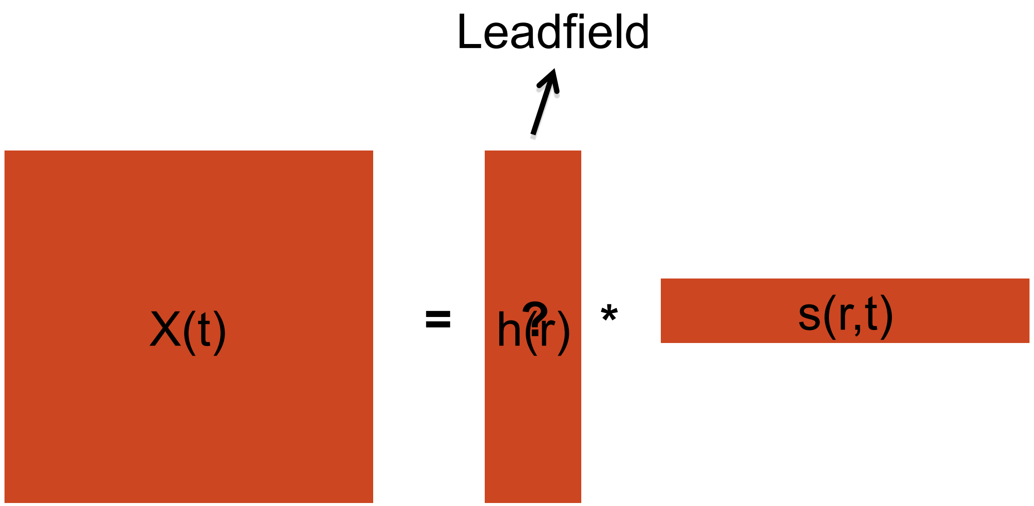
- **Forward model**
  - How is a source ‘seen’ by the sensor-array?
  - Given a source **s** at location **r** (and orientation η), what is the data **x**?



Beamformer ingredients: forward model

- **Forward model**
  - How is a source 'seen' by the sensor-array?
  - Given a source **s** at location **r** (and orientation η), what is the data **x**?

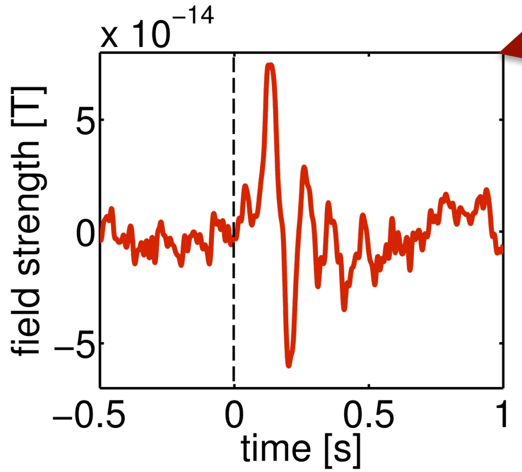


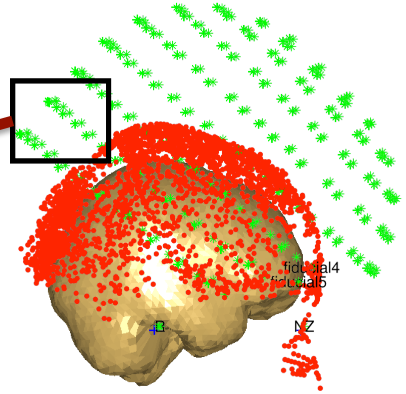


$$X = h * s$$

Beamformer ingredients: forward model

- **Sensor positions**
  - Where is the brain with respect to the sensors?

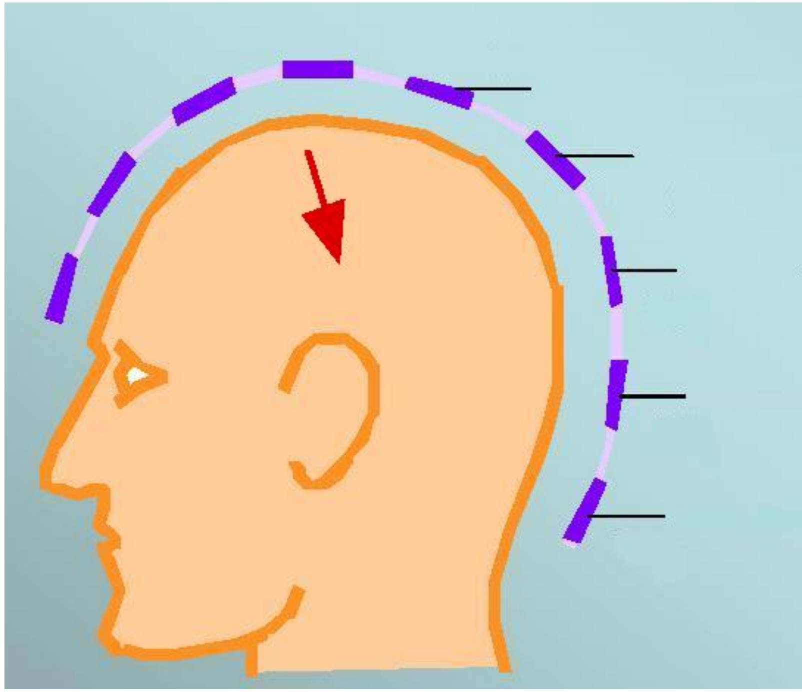




```
figure;
hold
ft_plot_sens(data.grad);
ft_plot_vol(vol);
hs = ft_read_headshape('hs_file');
ft_plot_headshape(hs);
```

Beamformer ingredients: forward model

- **Sensor positions**
  - Where is the brain with respect to the sensors?
  - Position of the head in the MEG-helmet

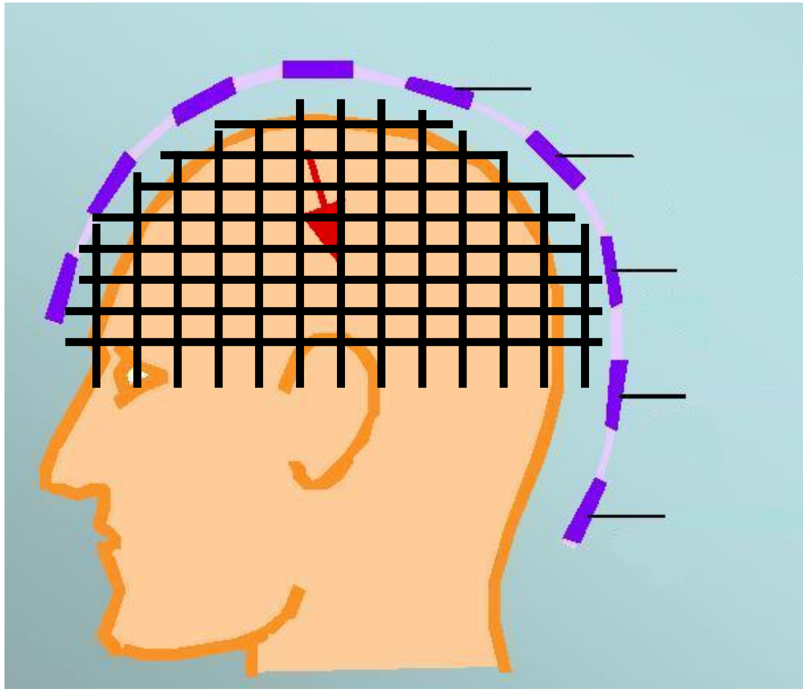


```
cfg  = [];
cfg.method = 'dics';
       .
       .
       .
       .
source = ft_sourceanalysis(cfg, freq);
```

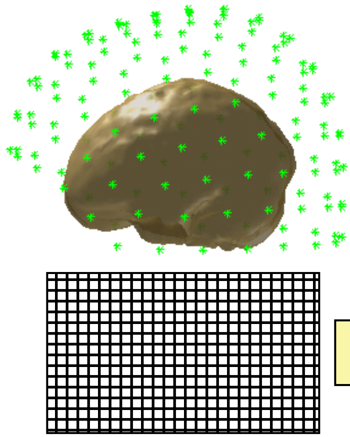
```
freq.grad

    freq.grad.coilpos = [Mx3]
    freq.grad.coilori = [Mx3]
    freq.grad.label   = {Nx1}
    freq.grad.tra     = [NxM]
```

Beamformer ingredients: forward model

- **Positions of the potential sources**
  - Which locations do you want to 'scan'?



```
cfg    = [];
cfg.method = 'dics';
cfg.grid    = sourcemodel; %create one
      .               %yourself, or
      .               %let fieldtrip
      .               %do it for you
source = ft_sourceanalysis(cfg, freq);
```

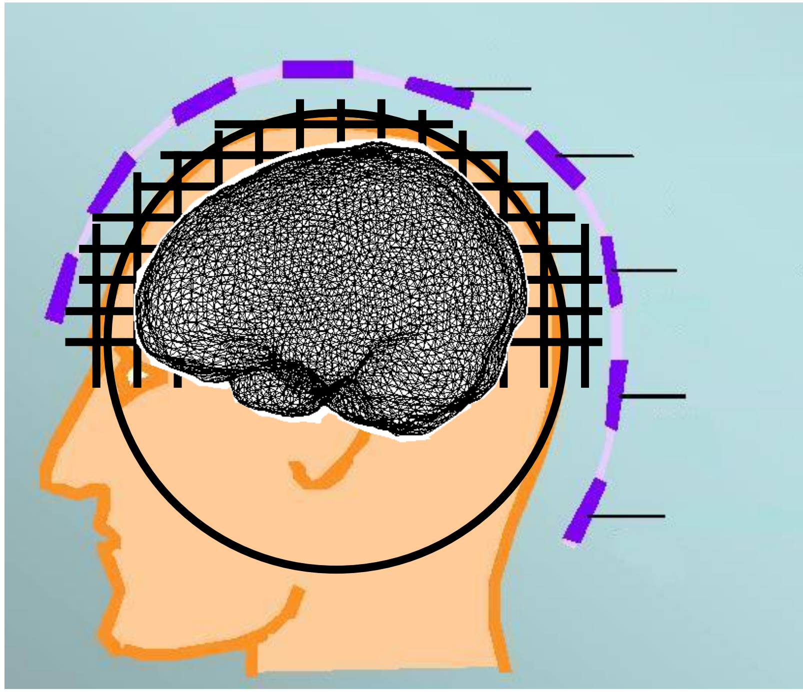


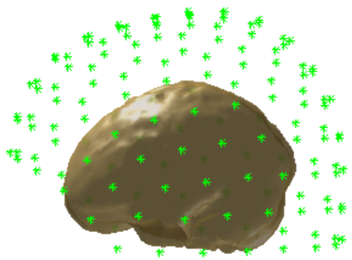
`ft_prepare_sourcemodel`

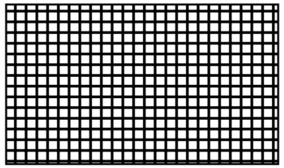
Beamformer ingredients: forward model

- **Volume conductor model**
  - What is the shape of the volume in which current is flowing?

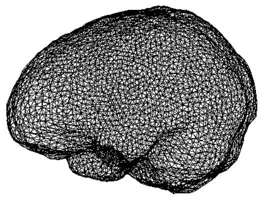
```
cfg  = [];
cfg.method = 'dics';
cfg.grid   = sourcemodel;
cfg.vol    = headmodel;  %create one
      .                  %yourself
      .
source = ft_sourceanalysis(cfg, freq);
```







ft_prepare_sourcemodel



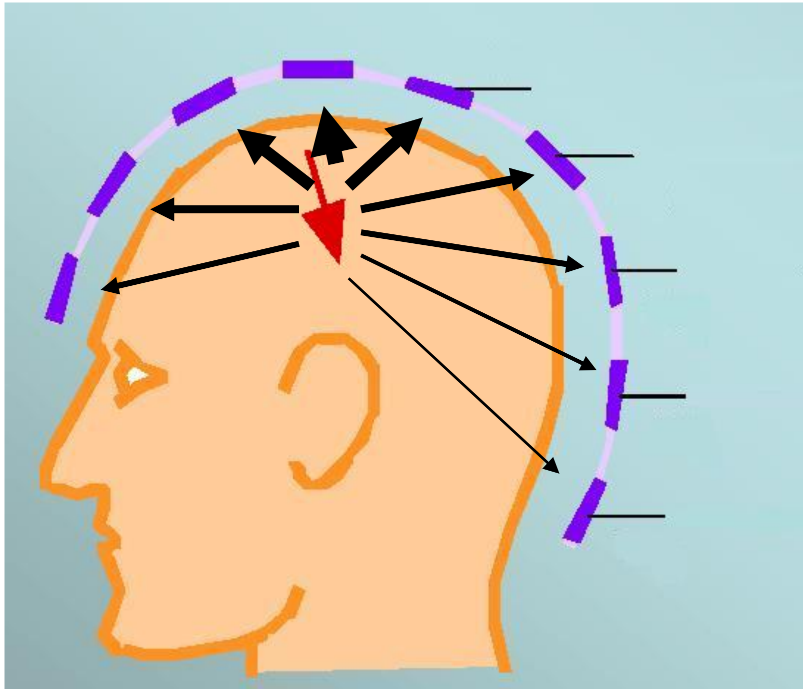
ft_prepare_headmodel

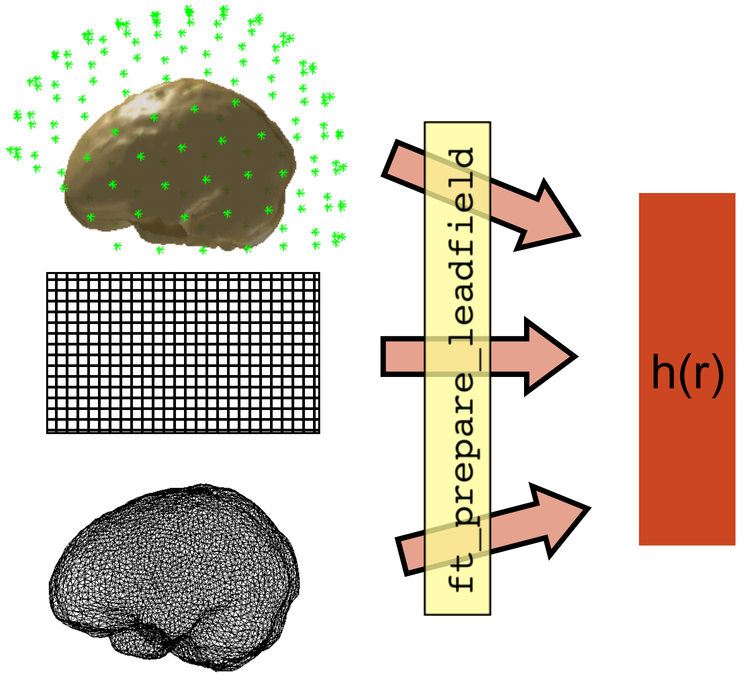
Beamformer ingredients: forward model

- **Forward model**

h = leadfield matrix

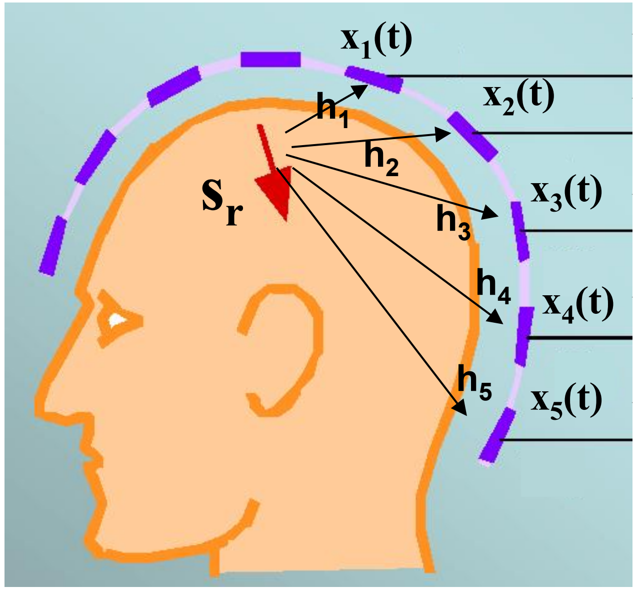
```
cfg   = [];
cfg.method = 'dics';
cfg.grid    = sourcemodel;
cfg.vol     = headmodel;  %create one
      .                   %yourself
      .
source = ft_sourceanalysis(cfg, freq);
```



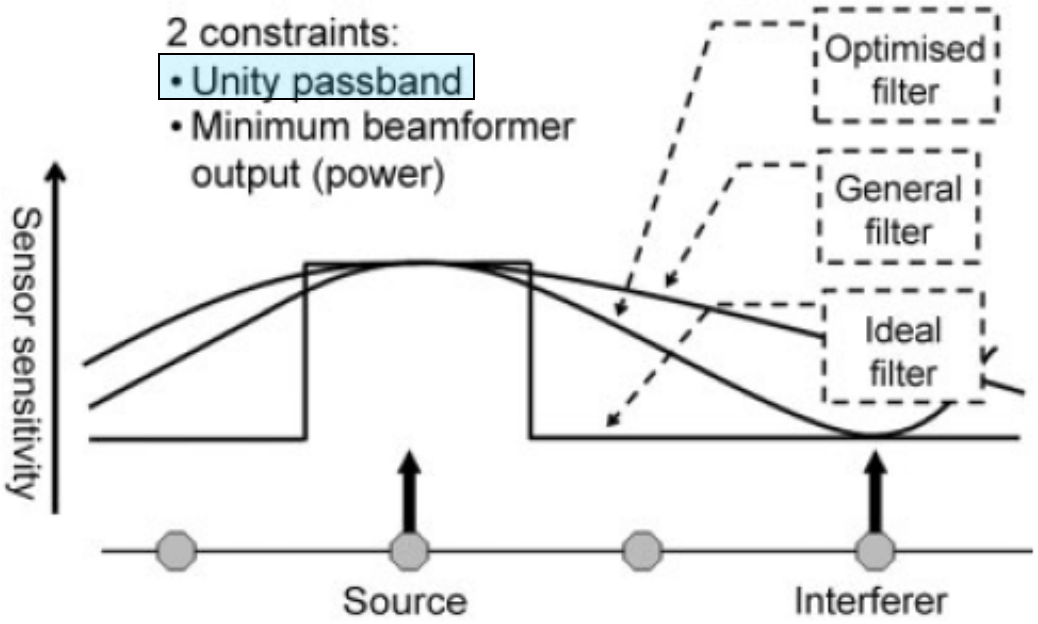


Beamformer: the question revisited

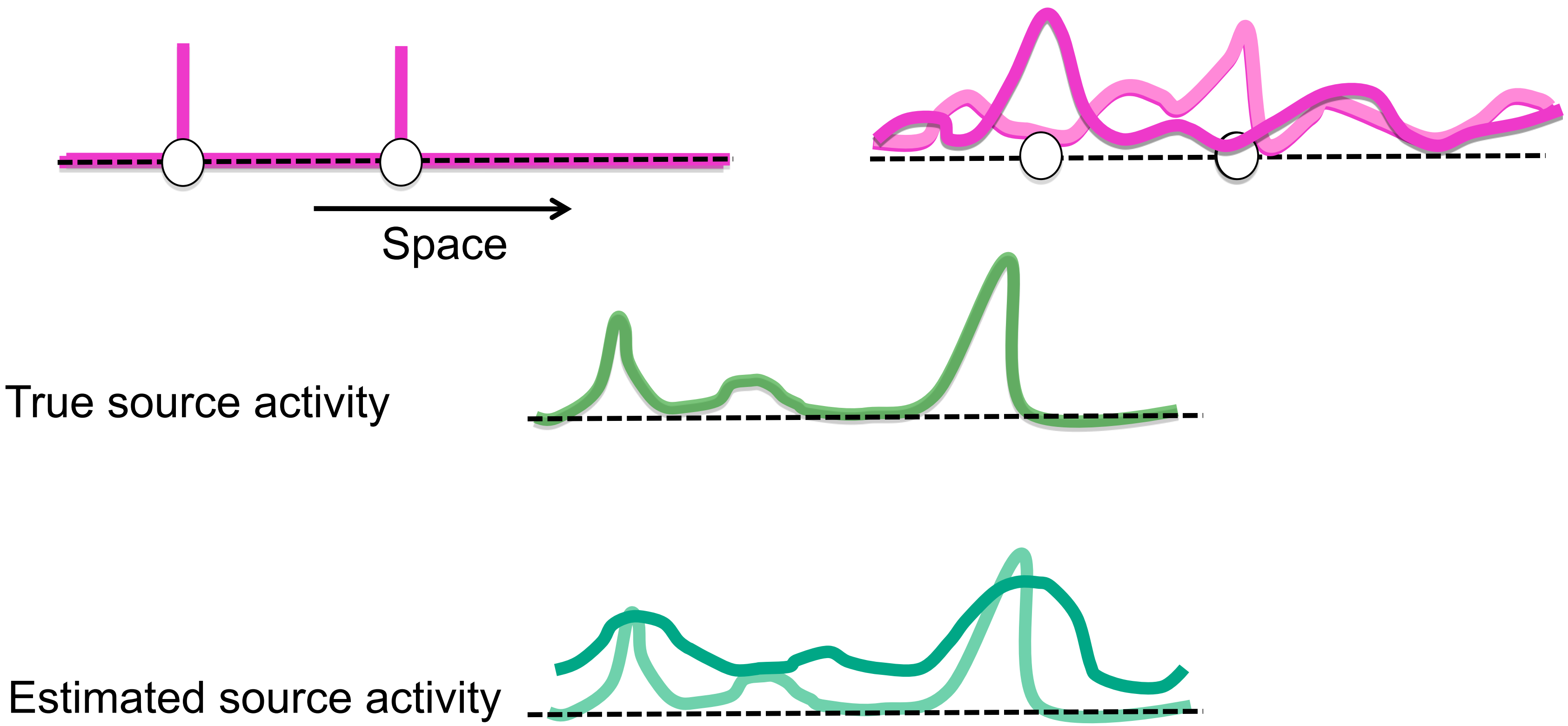
- What is the activity of a source **s**, at a location **r**, given the data **X**?
- We know how to get from source to data: $\mathbf{X} = \mathbf{h} * \mathbf{s}$
- We want to go from data to source: $\mathbf{w}^T * \mathbf{X} = \hat{\mathbf{s}}$
- $\mathbf{w}^T$ is called a spatial filter



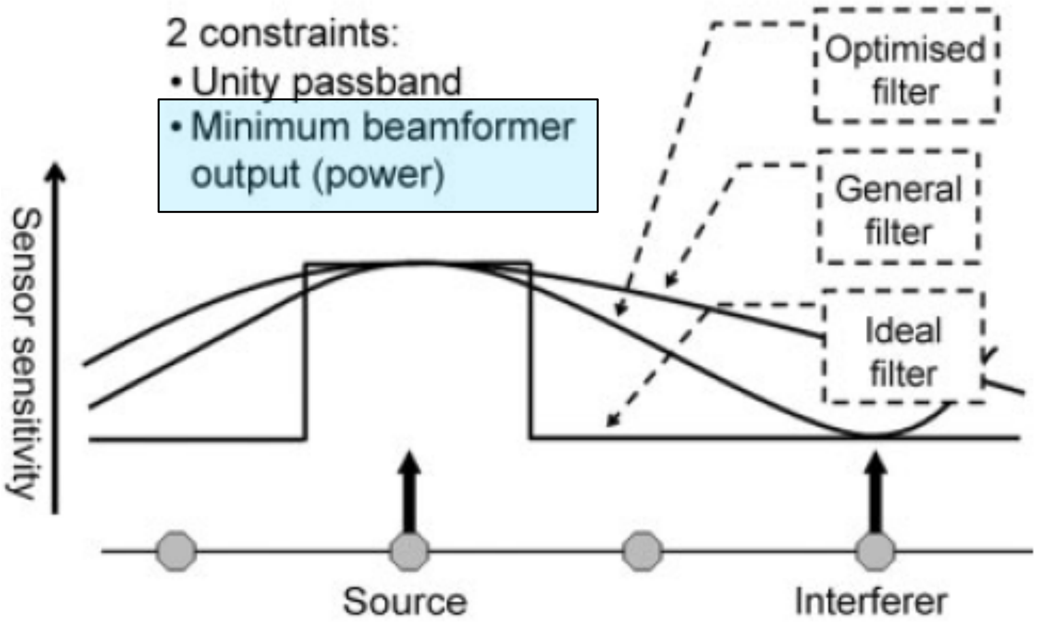
Sensitivity of a spatial filter



Concept of a spatial filter

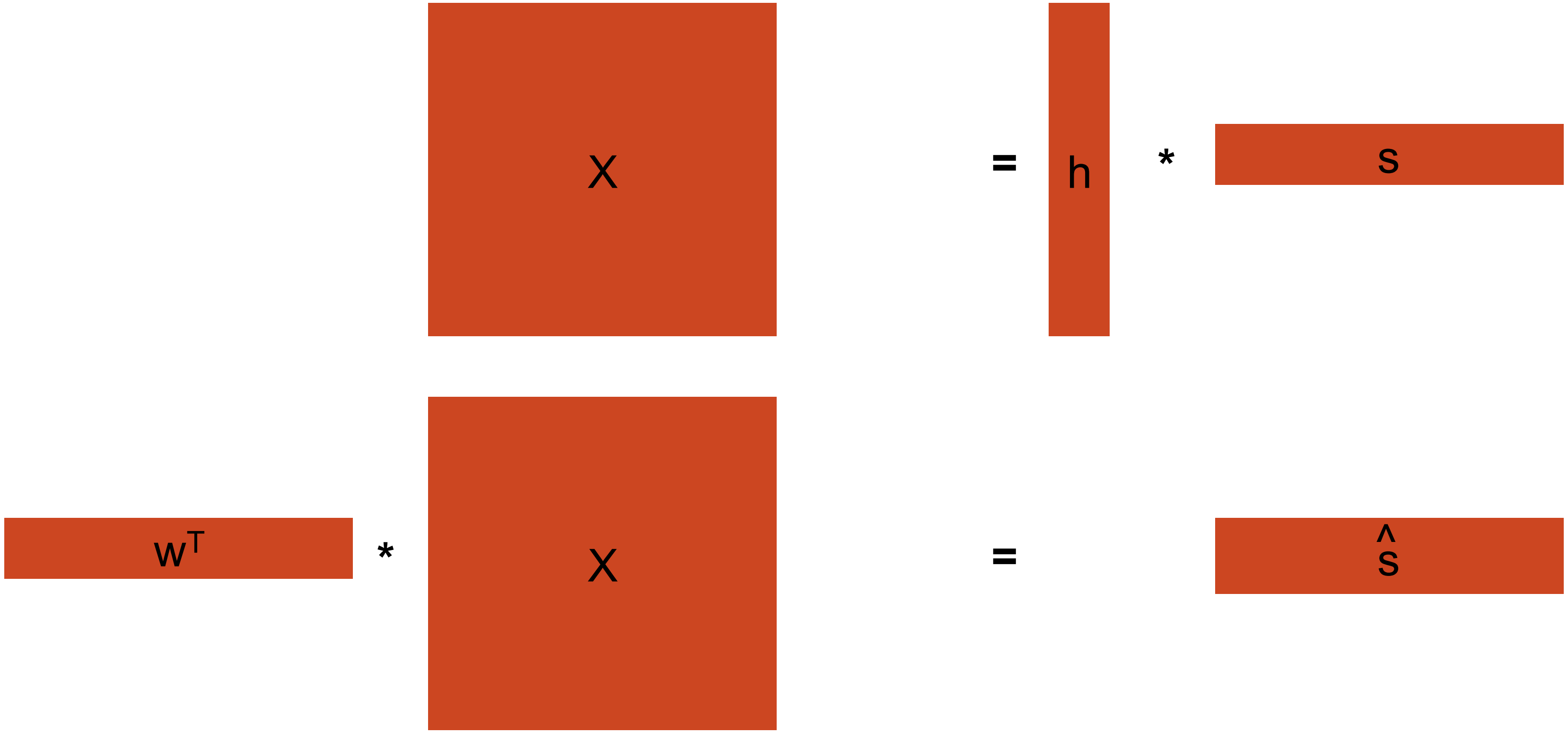


Sensitivity of a spatial filter



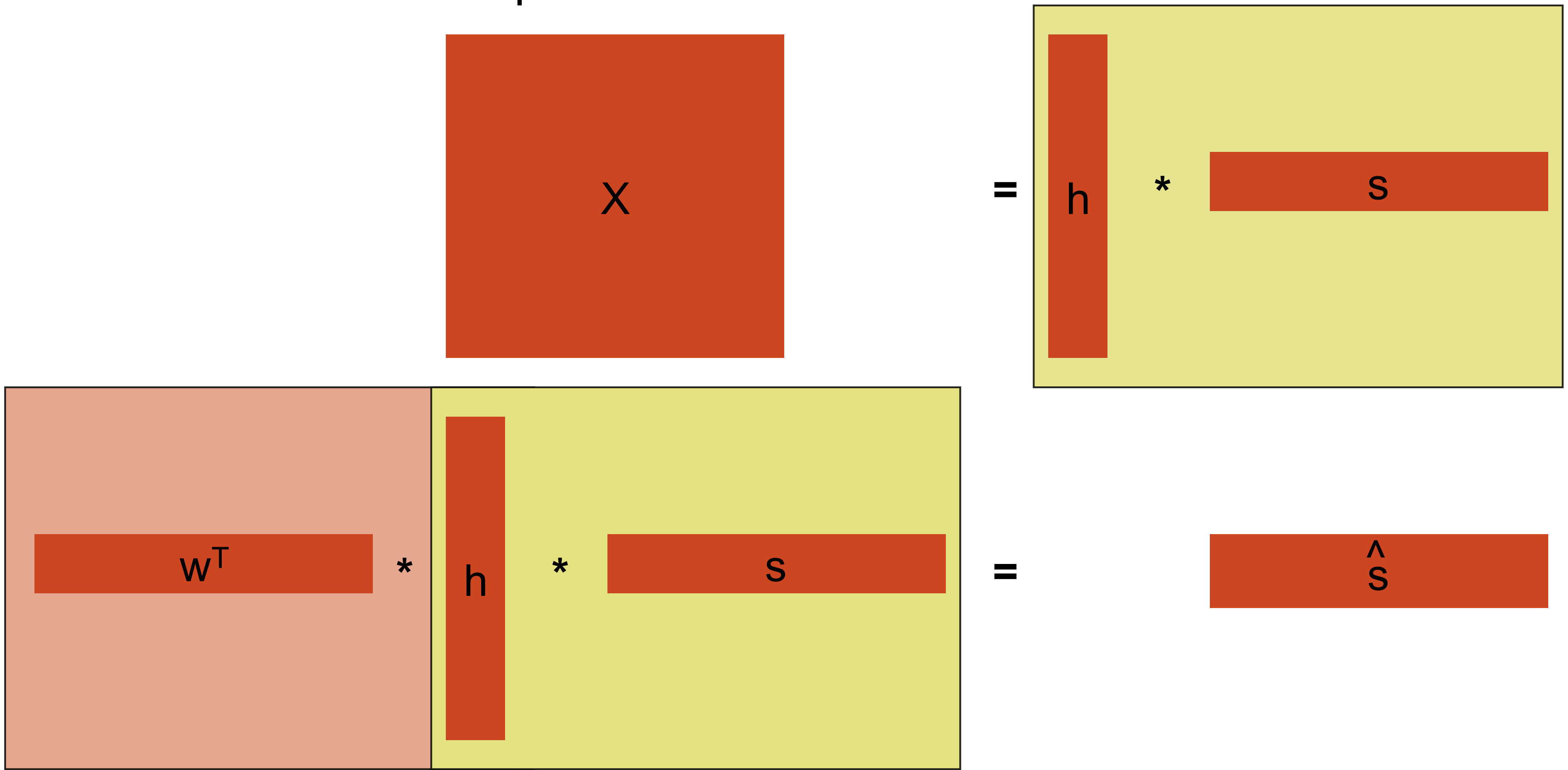
Beamformer: the question revisited

- What is the activity of a source **s**, at a location **r**, given the data **X**?
- We know how to get from source to data: $\mathbf{X} = \mathbf{h} * \mathbf{s}$
- We want to go from data to source: $\mathbf{w}^T * \mathbf{X} = \hat{\mathbf{s}}$
- $\mathbf{w}^T$ is called a spatial filter



Beamformer: the question revisited

- What is the activity of a source **s**, at a location **r**, given the data **X**?
- We know how to get from source to data: $\mathbf{X} = \mathbf{h} * \mathbf{s}$
- We want to go from data to source: $\mathbf{w}^T * \mathbf{X} = \hat{\mathbf{s}}$
- $\mathbf{w}^T$ is called a spatial filter

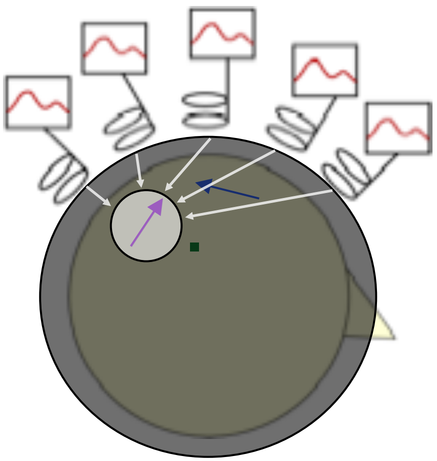


What would we like a spatial filter to do?

$w^T * h = 1$

$w^T * h_1 = 1$

$w^T * h = 0$

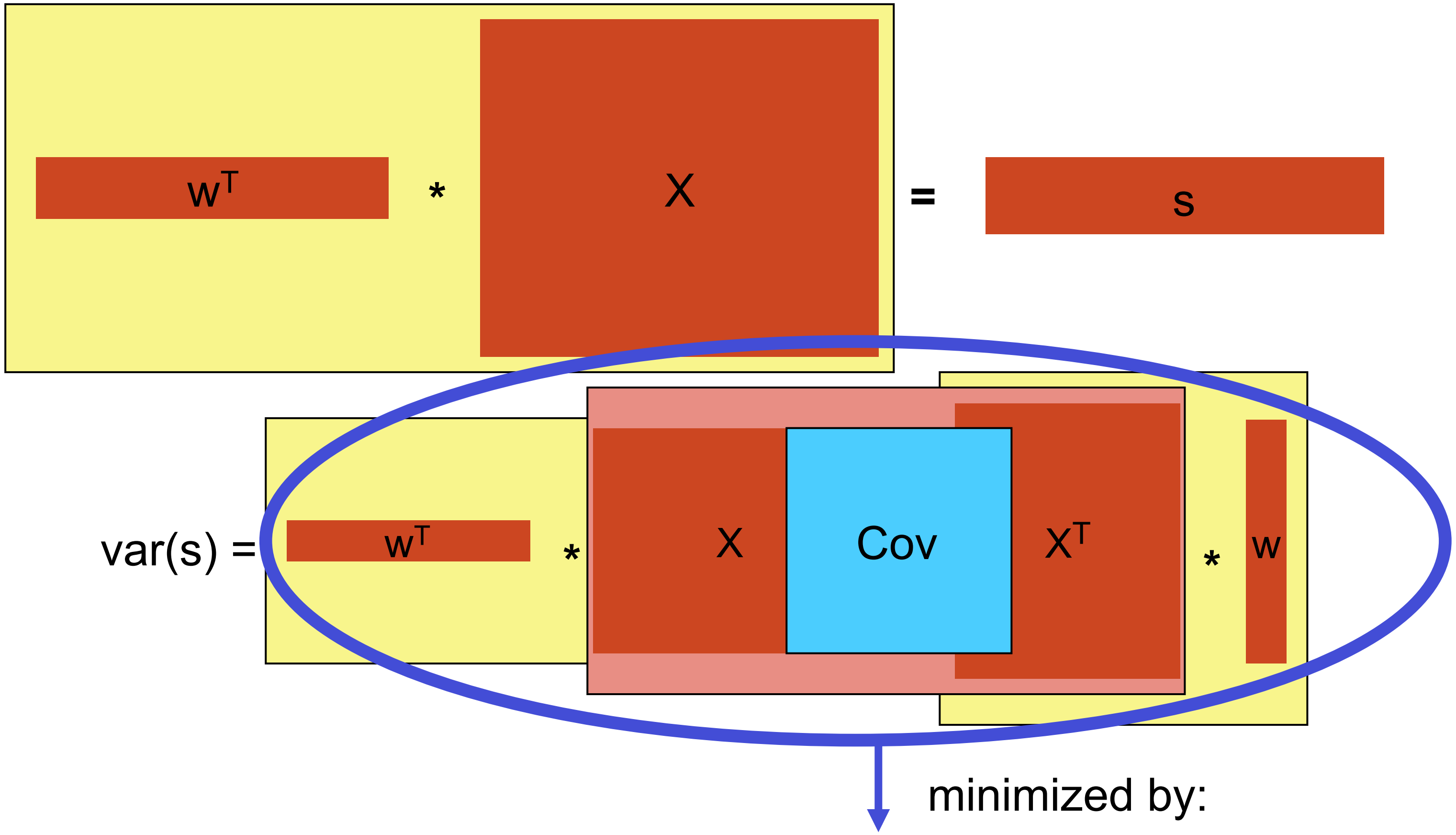


$w^T * h = 0$

$w^T * h = 0$

- $w^T_i h_i = 1$: unit gain constraint
- $w^T_i h_k = 0$: Cannot generally be fulfilled
- Minimize the variance of the filter output

Adaptive spatial filter: minimum variance constraint

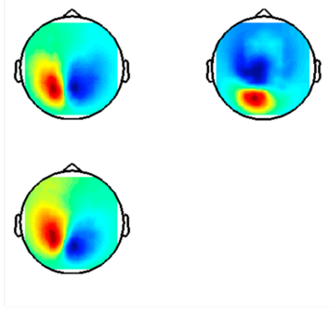


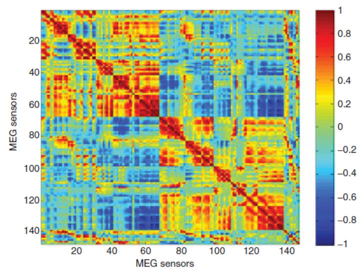
$$w^T = [h^T\, Cov^{-1} h]^{-1}\, h^T\, Cov^{-1}$$

Beamformer ingredients

- Forward model
  - Volume conduction model (typically using MRI)
  - Sensor positions
  - Points to 'scan': regular grid, cortical sheet, etc.
- Experimental data
  - Time domain: covariance
  - Frequency domain: cross-spectral density

Leadfield for each source





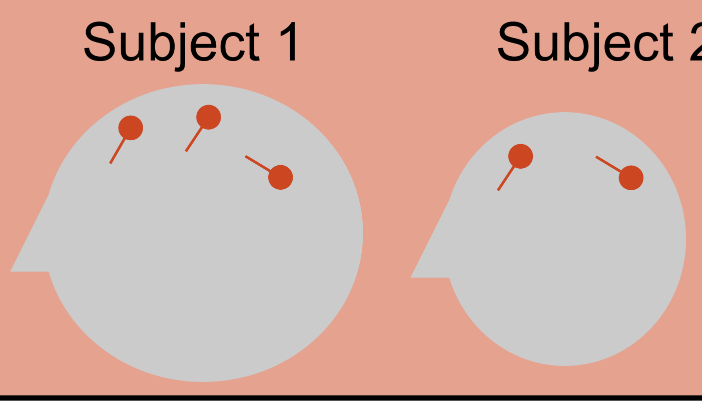

```
freq

        freq.freq
        freq.crsspctrm
        freq.powspctrm
        freq.labelcmb
        freq.label
```

```
freq.grad

    freq.grad.coilpos = [Mx3]
    freq.grad.coilori = [Mx3]
    freq.grad.label   = {Nx1}
    freq.grad.tra     = [NxM]
```

```
cfg   = [];
cfg.method = 'dics';
cfg.grid    = sourcemodel;
cfg.vol     = headmodel;
      .
      .
source = ft_sourceanalysis(cfg, freq);
```

Strengths of beamforming

Easier to average over subjects (compared to dipole methods)



Suitable for SPM-like statistics

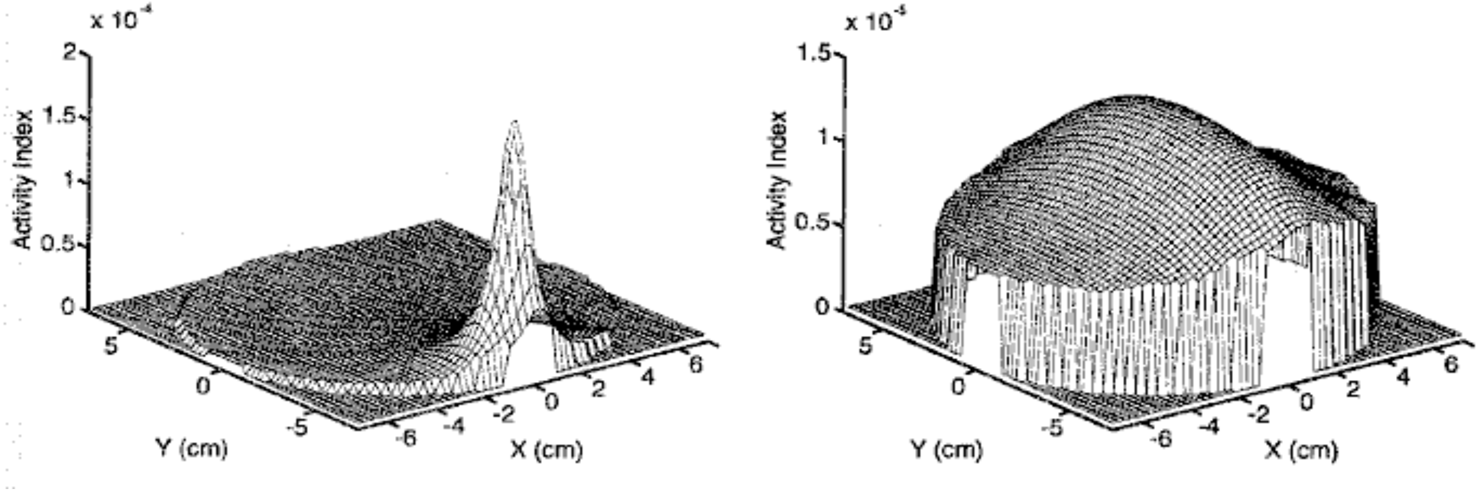
Because source estimation at each point independent of other points

(Most often) beamforming more spatially focal than distributed source (min norm) methods

No a priori assumptions about amount of sources or locations of sources

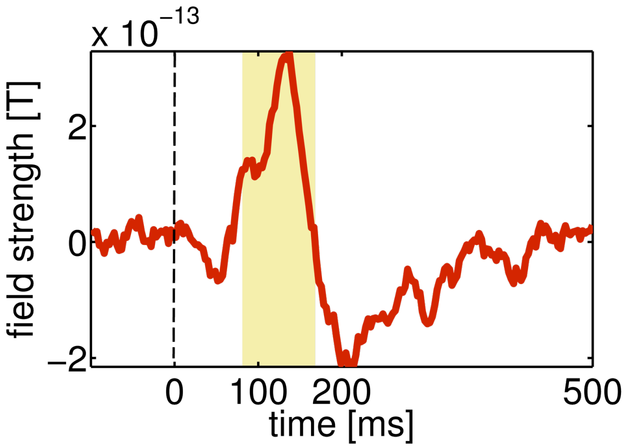
Limitation of beamforming

Sources should not be too correlated



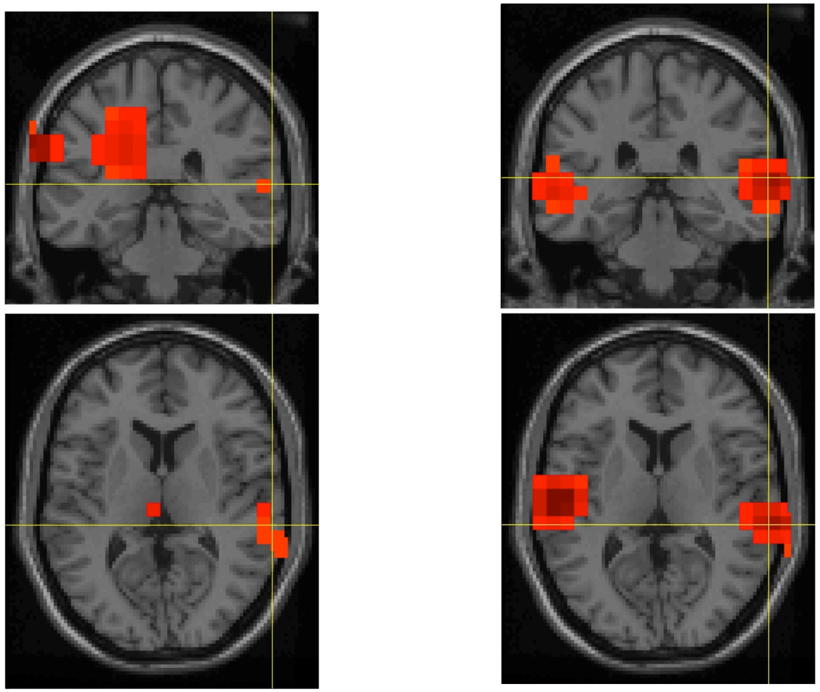
uncorrelated sources, van Veen et al. (1997)
mildly correlated sources
perfectly correlated sources

Limitation of beamforming




```
cfg = [];
cfg.covariance='yes';
cfg.covariancewindow = [-.2 .2];
avg = ft_timelockanalysis(cfg, tlk);
```

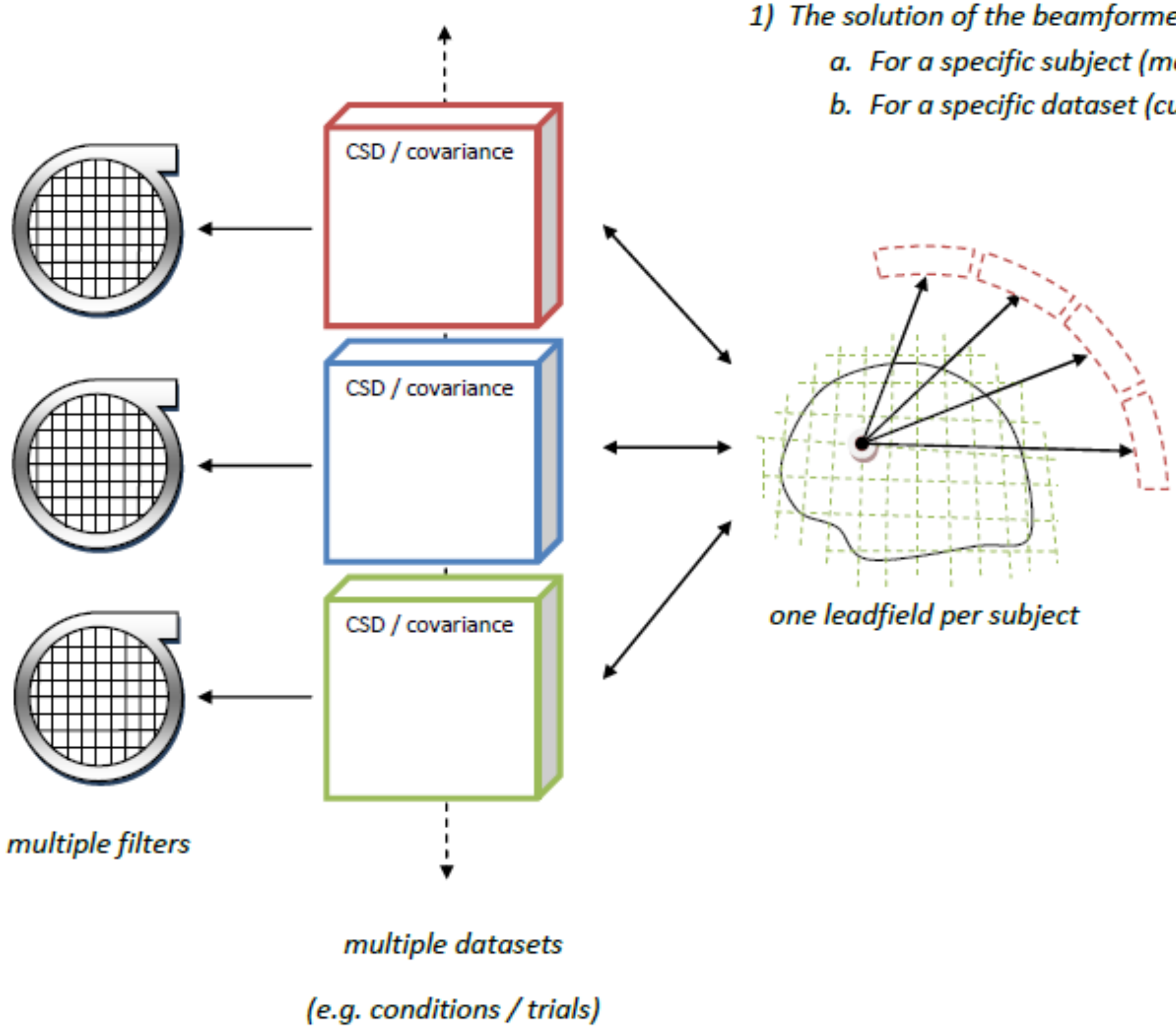
```
cfg   = [];
cfg.method = 'lcmv';
        .
        .
source=ft_sourceanalysis(cfg, avg);
```



Contrasting conditions with beamforming

Q: *How can I compare different datasets on the source level?*

1) *The solution of the beamformer is a unique spatial filter*
    a. *For a specific subject (morphology)*
    b. *For a specific dataset (current-source density or covariance)*



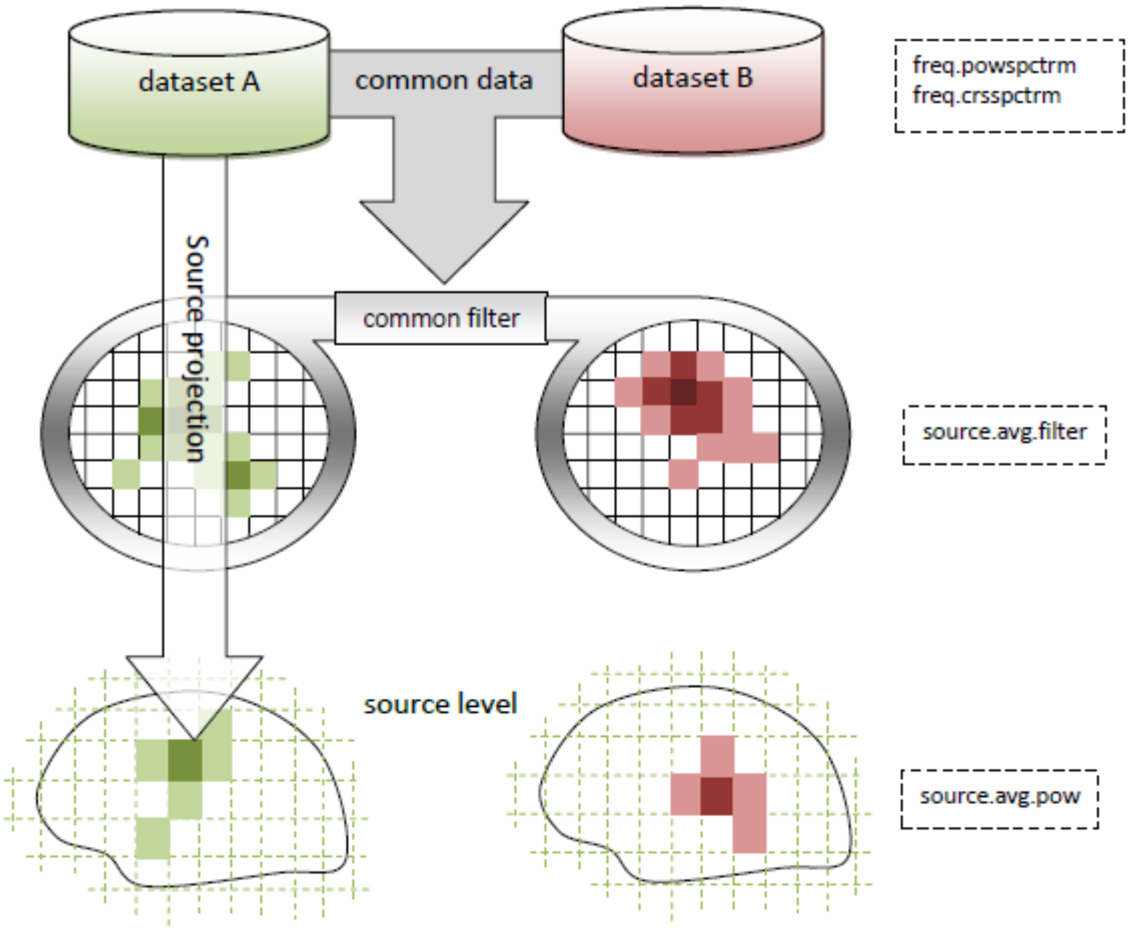
Slide courtesy Stephen Whitmarsh

Contrasting conditions with beamforming

Q: How can I compare different datasets on the source level?

1) *The solution of the beamformer is a unique spatial filter*
2) *Although you can use the same data to make the filter and project to source level*
   a. *often a common filter is calculated across conditions (using the same dataset)*
   b. *only the output of different datasets through the same filter are compared statistically*

```
cfg.grid = grid;
cfg.method = 'dics';
cfg.keepfilter = 'yes'
source = ft_sourceanalysis(cfg,freq)
% source.avg.filter = spatial filter
```



Slide courtesy Stephen Whitmarsh

Summary

Beamforming

- Scanning method, each point is estimated independently
- Inverse modeling by spatial filter

  Unifies two constraints:

  (1) pass all activity at location of interest while

  (2) suppressing as much activity (i.e. noise, other sources) as possible

  Makes use of covariance of data, and forward model
- Both possible in time and frequency domain
- No a priori assumptions about source configurations
- Applicable in very many scenarios

  Except when you have good reason to expect strongly correlated sources