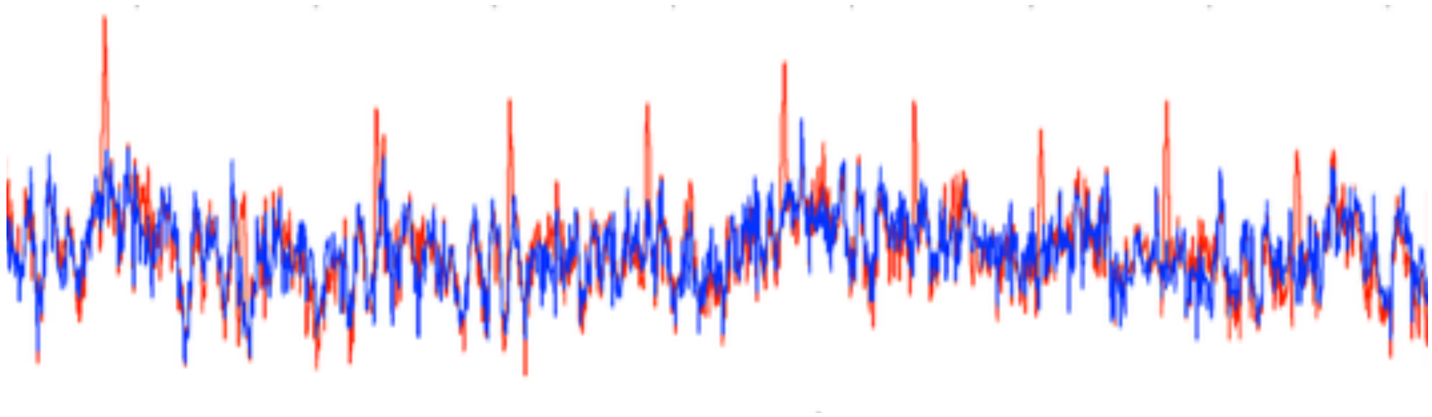
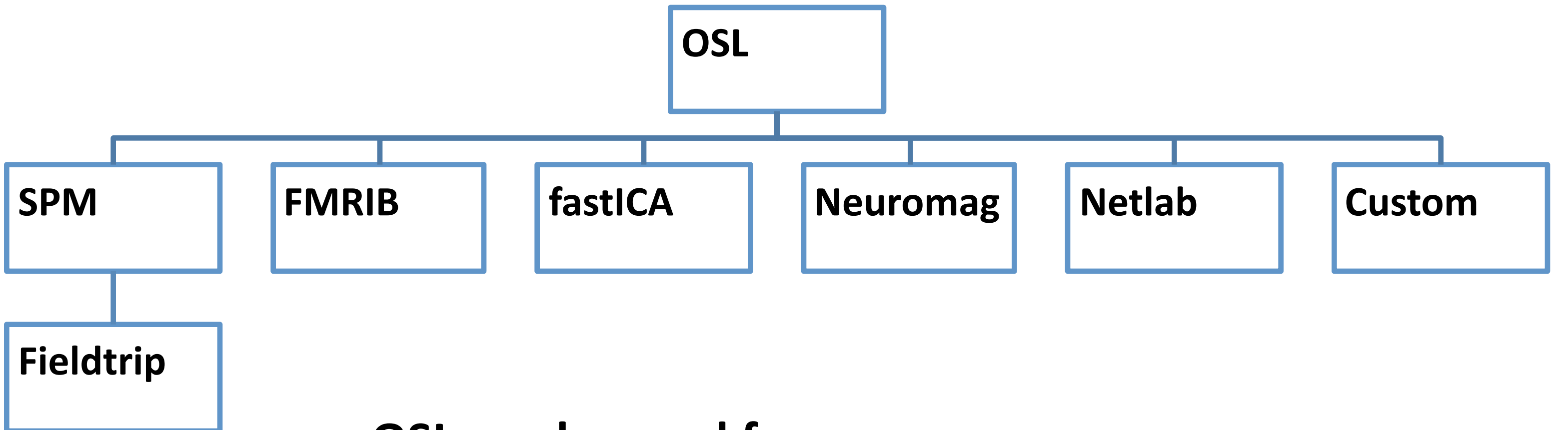
OSL Preprocessing



OHBA's Software Library



OSL can be used for

- **task and rest analyses**
- **preprocessing**
- **sensor space analysis**
- **source reconstruction**
- **statistics**

Overview

1. **Introduction to MEG artefacts**
2. **Manual Preprocessing**
   1. **Visual Inspection**

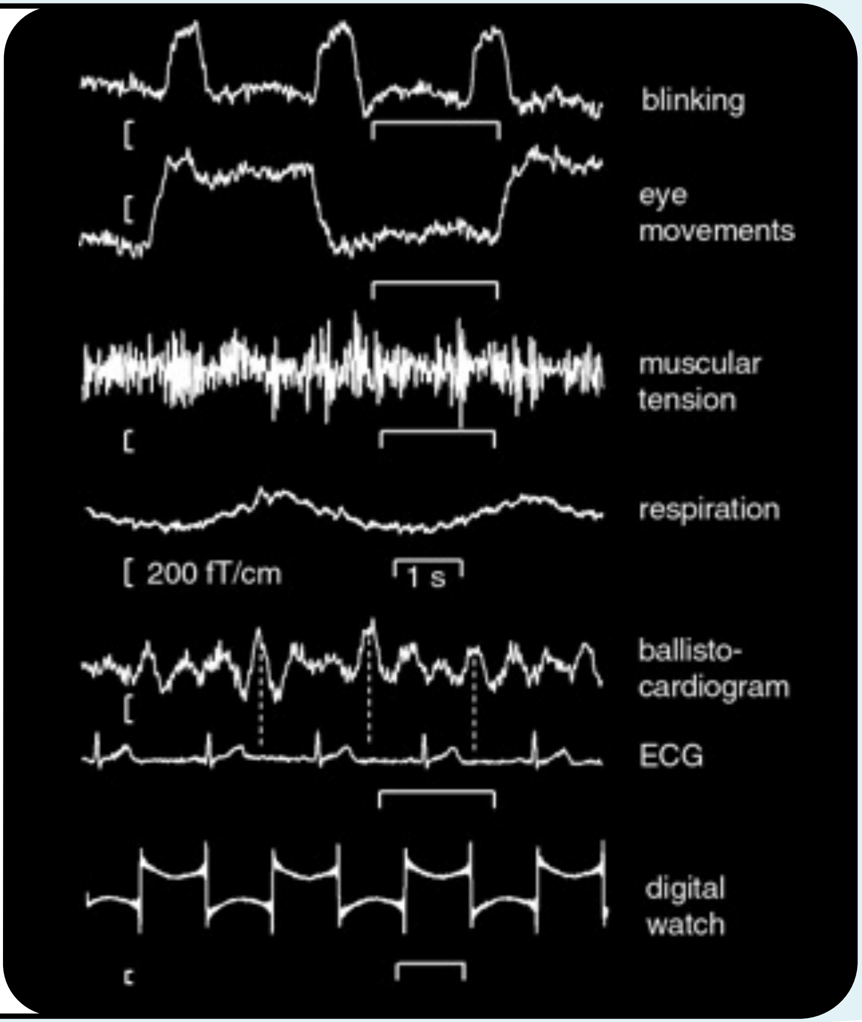
      Continuous Data using ***OSLview***
   2. **MaxFilter Artefacts – how to avoid them!**
   3. **De-noising using ICA**

      Intro to ICA and ***AFRICA***
3. **Automated Preprocessing (*Recommended approach)**

   OPT (OSL's Preprocessing Pipeline)

Artefacts - Know thy Enemy

- Biological artifacts
  - *Saccades, blinks, microsaccades*
  - *Muscular artefacts (high freq.)*
  - *Heartbeat*
  - *Respiration*
- Electrical/other
  - *50 Hz line noise*
  - *Scanner artifacts (jumps, spikes)*
  - *Channel saturation*
  - *MRI magnetisation*



Ultimate Strategy – Avoid Artefacts

- Good Experiment Design
  - Self-initiated trials, with preceding blink
  - Fixation prior to stimuli
  - Monitor subject and tell them if they blink
  - Frequent breaks
- Good Screening/Communication
  - No make-up, unsuitable clothing e.g. bras with under-wires. Let your subject know in advance….

ASK FOR HELP FROM EXPERIENCED SCANNERS

Backup Strategy – Record Artefacts

- We can't stop someone's heart beating – ***can we?***
  - Some artefacts can't be avoided (e.g. heart beat)
  - Recording these artefacts gives us a better chance to detect and remove them
- Record ECG, Eyetracker, EOG, (…EMG, Respiration)
  - This may be restricted by your specific experimental constraints.

THE MORE EXTERNAL SIGNALS THE BETTER!

Continuous vs. Trial-wise Data

Continuous

- MaxFilter
- Downsampling
- Visual inspection with ***oslview***
    - Reject "bad" channels
    - Flag *BadEpochs*
- De-noising with ***AfRICA***

Trial-wise

- MaxFilter
- Downsampling
- Visual inspection with ***oslview***
    - Reject "bad" channels
    - Flag *BadEpochs*
- De-noising with ***AfRICA***
- "Bad" channel and trial rejection & inspection

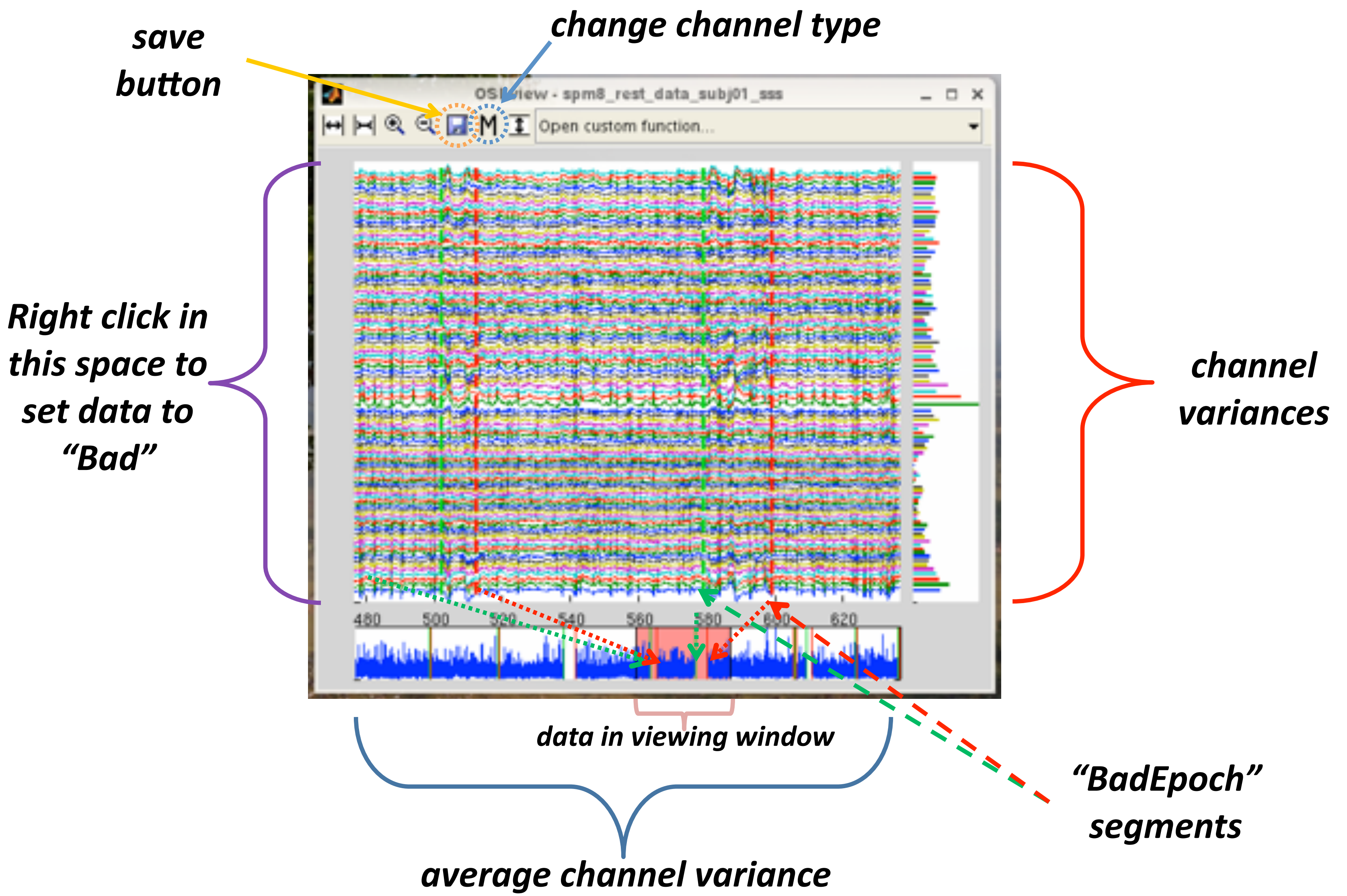
These are the recommended manual strategies.

Visual Inspection is **Essential**!

All clever artefact rejections tools fail at some point

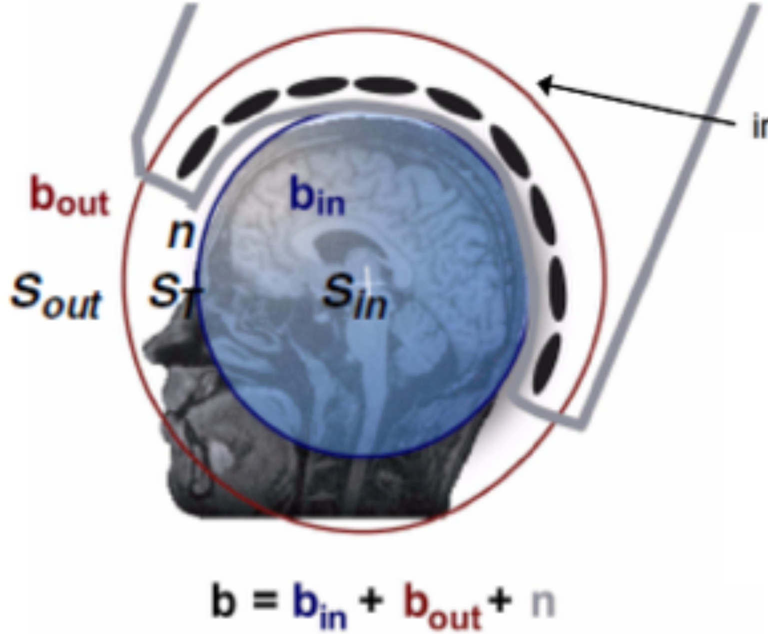
- If running the manual pipeline, you must check the output at each stage.
  – e.g. use ***OSLview*** to check pre-epoched data.
    - You will play with this today
- If running the automated pipeline (OPT) inspect the diagnostic output plots
  - See later

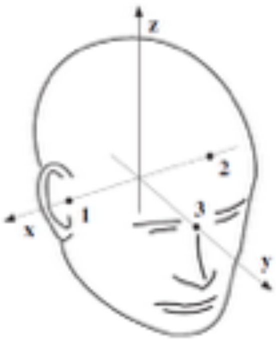
OSLview


save button
change channel type
OSLview - spm8_rest_data_subj01_sss
Open custom function...
Right click in this space to set data to "Bad"
channel variances
480
500
520
540
560
580
600
620
data in viewing window
"BadEpoch" segments
average channel variance

Maxfilter

- Maxfilter is a program provided by Elekta, which implements a spatial signal space separation (SSS) algorithm to remove the external noise (b_{out}):





Movement Compensation

- Maxfilter can use ***MaxMove*** to compensate for head movements by reprojecting the data onto the sensors as if it had been recorded with the head in a different position.

- This can be used in two ways:

    1) to continuously compensate for movements made within a recording session (*-movecomp* option)

        - requires that the HPI signal from the coils was recorded continuously during the MEG session

    2) to bring different sessions / subjects into a common frame, making the sensor-space results more comparable between sessions / subjects (*-trans* option)

Maxfilter

- Maxfilter can also:
  - detect bad channels
  - downsample data,
  - output log files for head position,
  - and other things besides - see the manual for the full set of options

- There is a function to call MaxFilter called *osl_call_maxfilter.m*

Double Maxfilter Procedure

- We advise you use the following "Double Maxfilter Procedure" when using MaxFilter.

1. Call ***osl_call_maxfilter*** without MaxFilter
   S.nosss = 1;
2. Convert to SPM and open in ***oslview***
3. Mark any channels with scanner artefacts as "*Bad*".
4. Call ***osl_call_maxfilter*** with MaxFilter & bad channels.
   S.nosss = 0; S.spmfile points to the SPM file from steps 2 & 3.

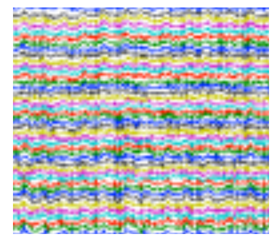
De-noising with AfRICA

Artefact **R**ejection using **I**ndependent **C**omponent **A**nalysis

Data driven method to split our MEG data

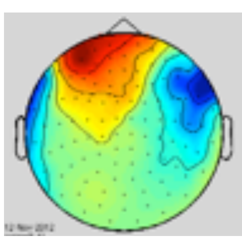
$N_{samples}$

$N_{channels}$ $\boldsymbol{Y}$



into a linear mixture of temporally

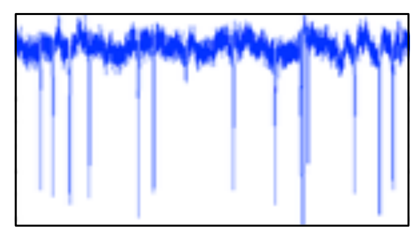
independent components and topographies.



$N_{components}$

$N_{channels}$ $\boldsymbol{A}$ **x** $N_{components}$ $\boldsymbol{S}$

$N_{samples}$



ICA – a brief introduction

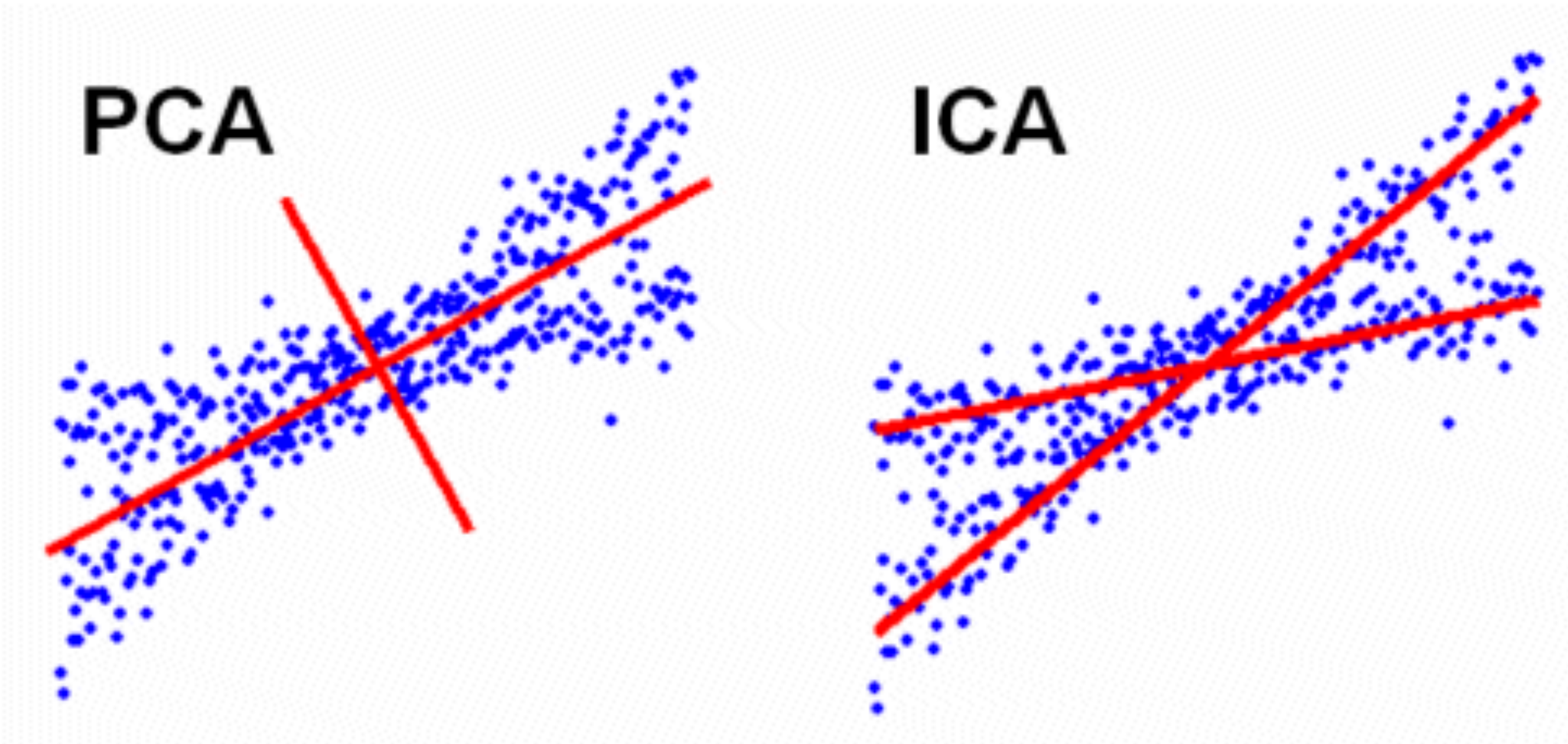
- A blind source separation technique for un-mixing.

$$\boldsymbol{Y} = \boldsymbol{A} \times \boldsymbol{S}$$

Data — ***Mixing Matrix*** — ***Underlying Sources***

- Our estimated independent time courses ***S*** are a linear mixture of our original data ***Y***.

ICA – a brief introduction



ICA – a brief introduction

- A blind source separation technique for un-mixing.

$$\boldsymbol{Y} \quad = \quad \boldsymbol{A} \quad \times \quad \boldsymbol{S}$$

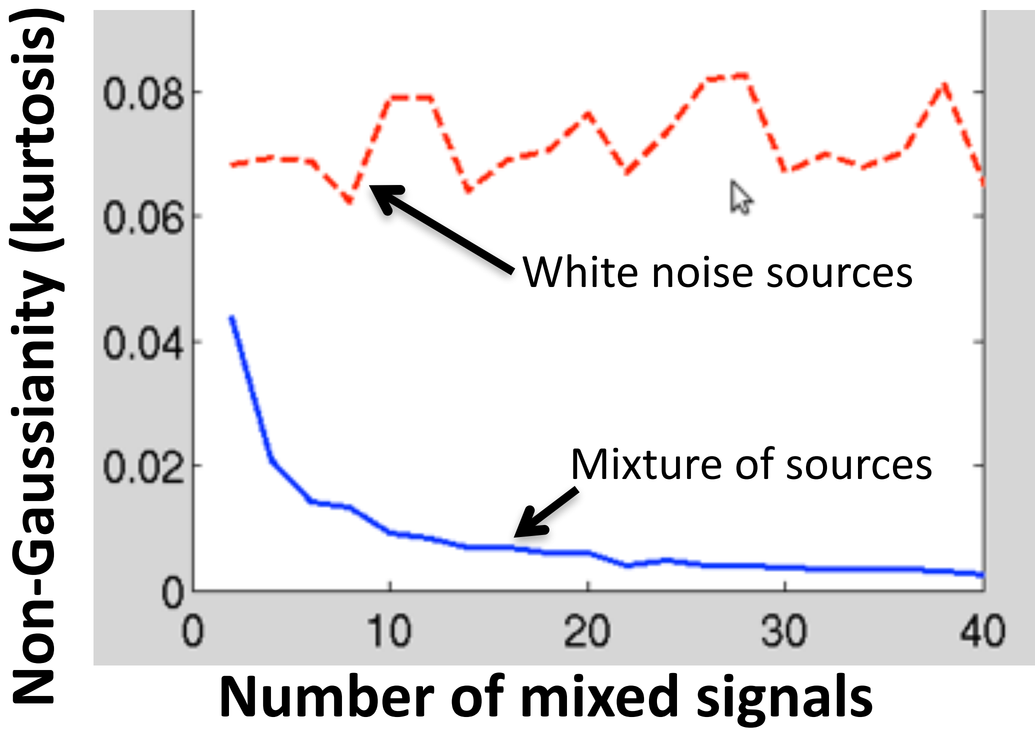
Data ***Mixing Matrix*** ***Underlying Sources***

- Because we don't know A or S the problem seems ill-posed
- We employ the **CENTRAL LIMIT THEOREM** to help us.

The Central Limit Theorem

A mixture of signals is always **more Gaussian** than the underlying signals.

As long as there are enough signals!



By searching for the set of maximally non-Gaussian signals we can reverse the mixing process and recover our unknown sources.

That's ICA!

Classifying Components

ICA un-mixes our MEG data but doesn't tell us which components are artefacts

AfRICA has two ways of helping you do this:

1.) Correlation with external signals

- If you have acquired ECG, Eyetracker, EOG etc

  AfRICA will flag components that match these.

2.) Extreme temporal kurtosis ("peakedness of the distribution")

- Extreme high and low kurtosis .

You can see both at work in today's practical!

Classifying Components

ICA un-mixes our MEG data but doesn't tell us which components are artefacts

AfRICA can be run in two modes:

1.) <u>Manual</u>

- In which you manually label components as artefacts (AFRICA can sort these components based on their correlation with artefact channels or their variance/kurtosis)

2.) <u>Automated</u>

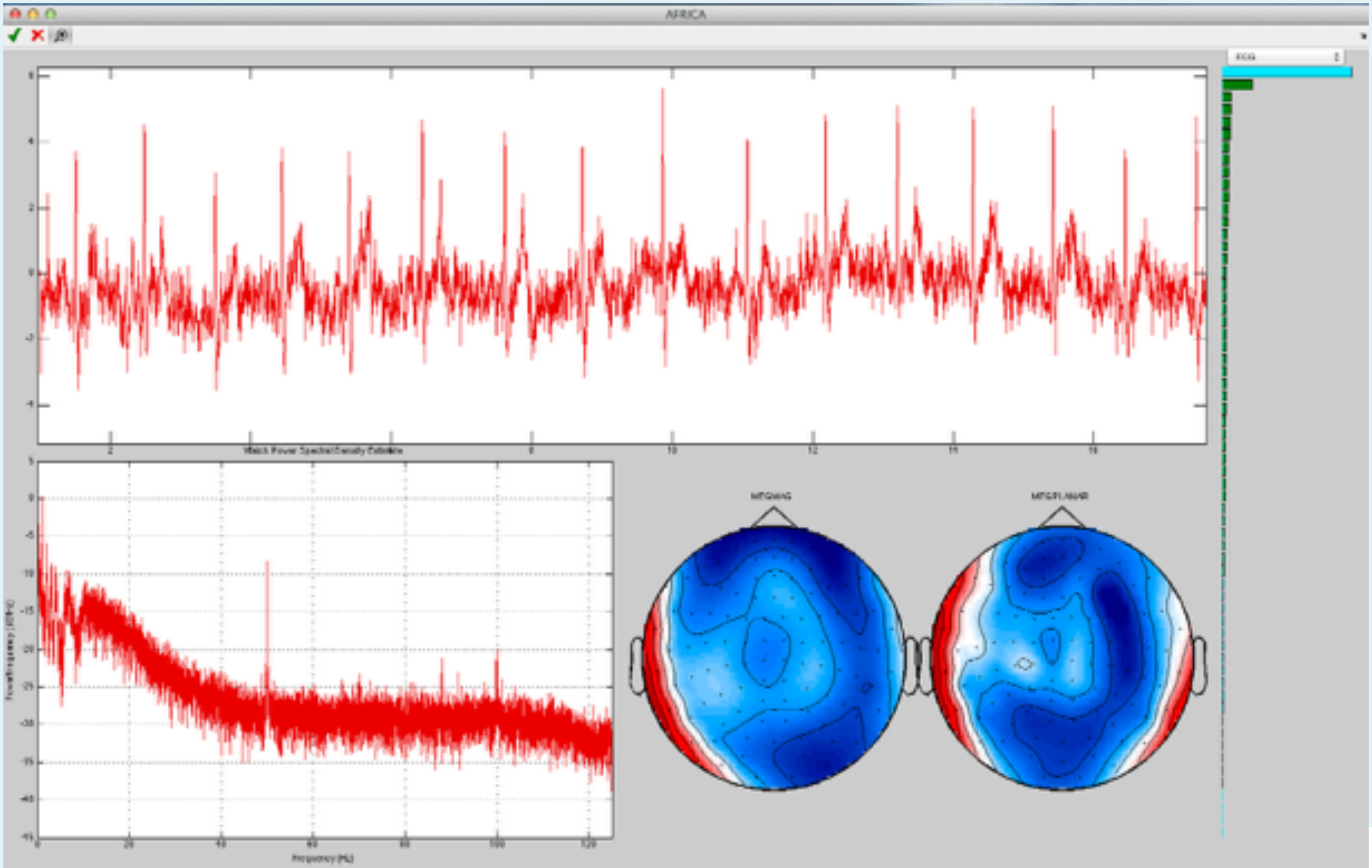
- AFRICA automatically thresholds artefact channel correlations and kurtosis (used by OPT)

Manual AFRICA

Set component as bad (or revert to good)

metrics (e.g. kurtosis, ECG, EOG, mains)

time course



power spectrum

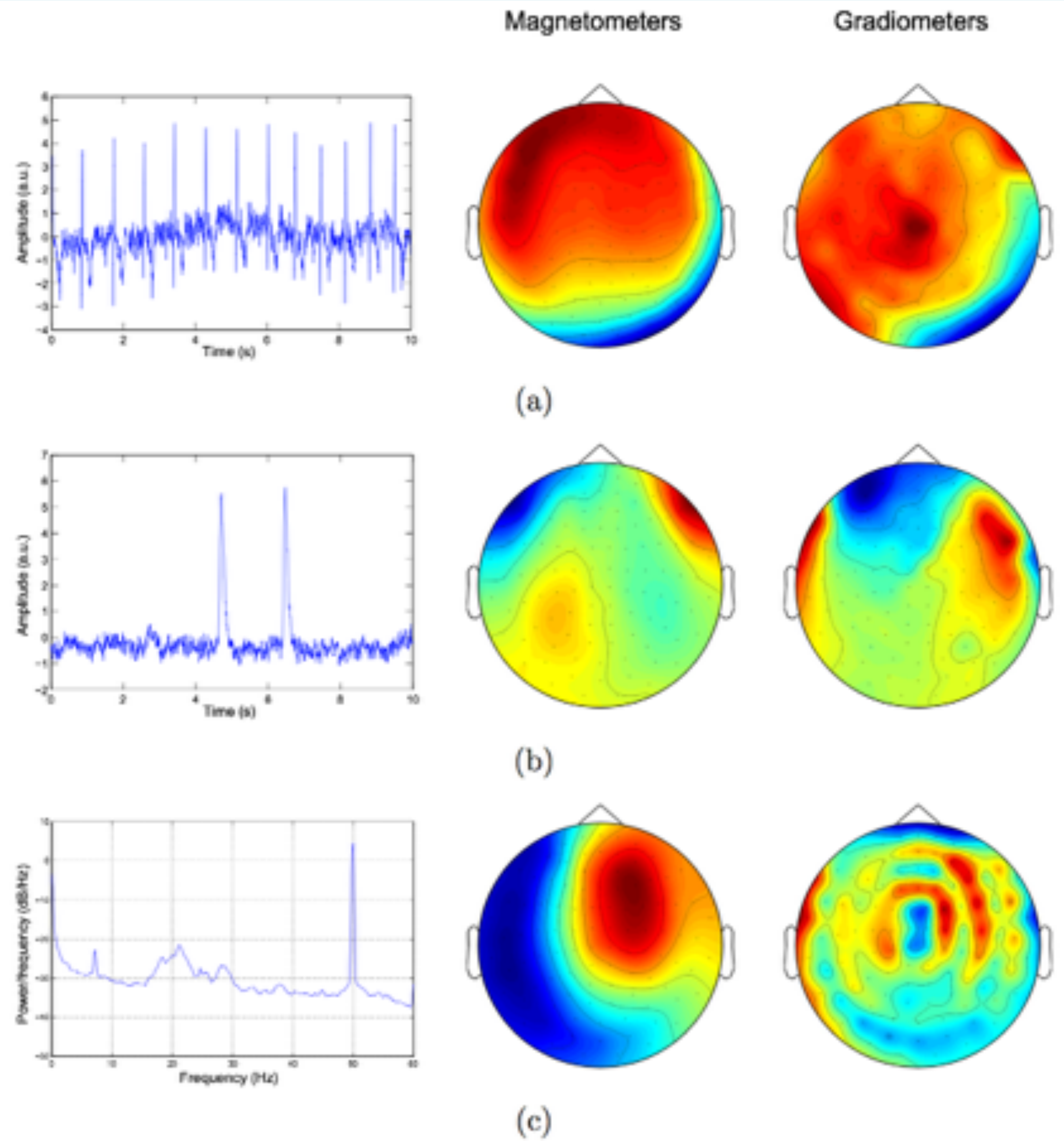
sensor topographies

Manual AFRICA

Cardiac component

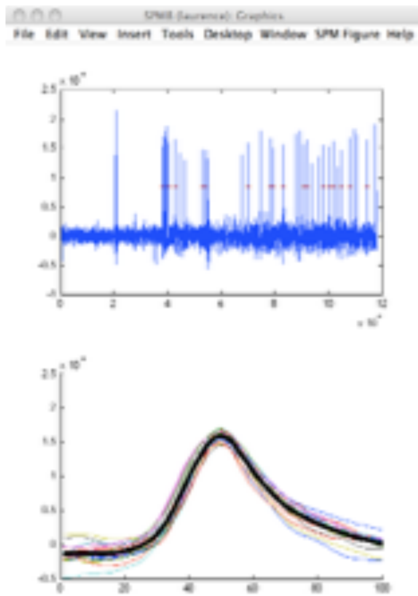
Eye blink component

Mains (50 Hz) component

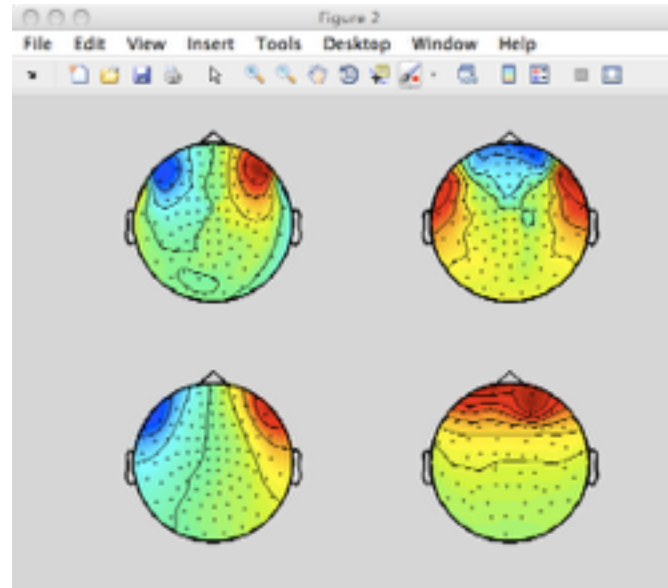


Alternative: PCA/Spatial Regression

- Isolate spatial topographies associated with artifact



1. Find some blinks in EOG, Create 'average MEG blink'



2. Run PCA of average blink to identify spatial topographies underlying blink

- Regress topography out of *raw, continuous data*

OPT (OSL Preprocessing Toolbox)

- Fully automated pipeline
- OPT runs through the following pipeline steps (any of which can be optionally turned off):
  1. *For Elekta Neuromag data*: Runs the "Double Maxfilter Procedure" (to help Maxfilter with detection of bad channels):
  2. Conversion of data into SPM format
  3. Downsampling
  4. High-pass filtering
  5. Automated AFRICA denoising
  6. Mark bad segments (using outlier rejection of dummy epochs)
  7. Coregistration (needed if intending to do subsequent analysis in source space)
  8. Epoching (If appropriate)
  9. Automated outlier trial and channel rejection

OPT - Data Input

- Data can be input as:

  Either (only for Elekta Neuromag data):

  - the full path of the raw fif files (pre-SSS) to pass to the Maxfilter

  Or:

  - the full path of the input files that will be passed to the SPM convert function (for Elekta Neuromag data this will be post-SSS .fif files

  Or:

  - the full path of the (already converted) SPM MEEG files

Using OPT

- Use osl_check_opt call to setup an OPT struct:
  - ➡ *opt= osl_check_opt(opt);*
  - ➡ *Requires limited mandatory settings*
  - ➡ *Fills other field with default values (which can then be adjusted before running)*

- Use osl_run_opt to run an OPT:
  - ➡ *opt=osl_run_opt(opt);*

OPT Output

- Results are stored in the directory specified in *opt.dirname,* with a *'.opt'* suffix
- *opt=osl_run_opt(opt)* also returns:
  - ➡ *opt.results*
- This contains:
  - ➡ *opt.results.logfile (file containing the matlab text output)*
  - ➡ *opt.results.report:* ***(Web page report with diagnostic plots)***
  - ➡ *opt.results.spm_files: (list of SPM MEEG object files for the continuous data, e.g. to pass into an OAT analysis)*
  - ➡ *opt.results.spm_files_epoched: (list of SPM MEEG object files for the epoched data, e.g. to pass into an OAT analysis)*

OPT Output

- It is highly recommended that you inspect both the *opt.results.logfile* and *opt.results.report*, to ensure that OPT has run successfully (See the practical).

Today's Practicals

- *Practicals + data are on the OSL Wiki*
- *Practical is in two parts:*

1) Manual Preprocessing Pipeline

2) Automated Preprocessing Pipeline (OPT)

Recommended Reading

Look at and use the OSL Wiki!

Independent Component Analysis (easy)
Independent Component Analysis – A Tutorial Introduction – James V. Stone

fastICA & ICASSO (advanced)
Hyvärinen, A., 1999. Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans. Neural Netw. 10 (3), 626–634.

ICA de-noising in MEG (relevant)
Mantini, D., et al. 2011. A Signal-Processing Pipeline for Magnetoencephalography Resting-State Networks. *Brain Connectivity*, *1*(1), 49–59.