CS463/516

Lecture 11

BOLD fMRI pre-processing:

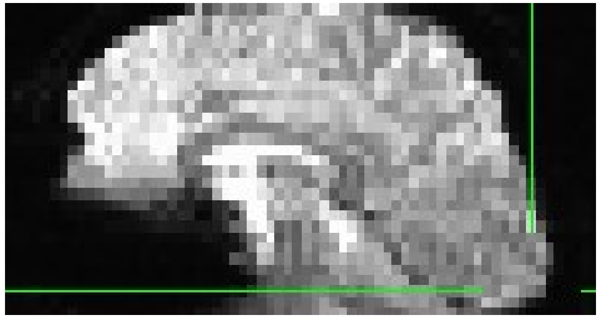
Nuisance regression

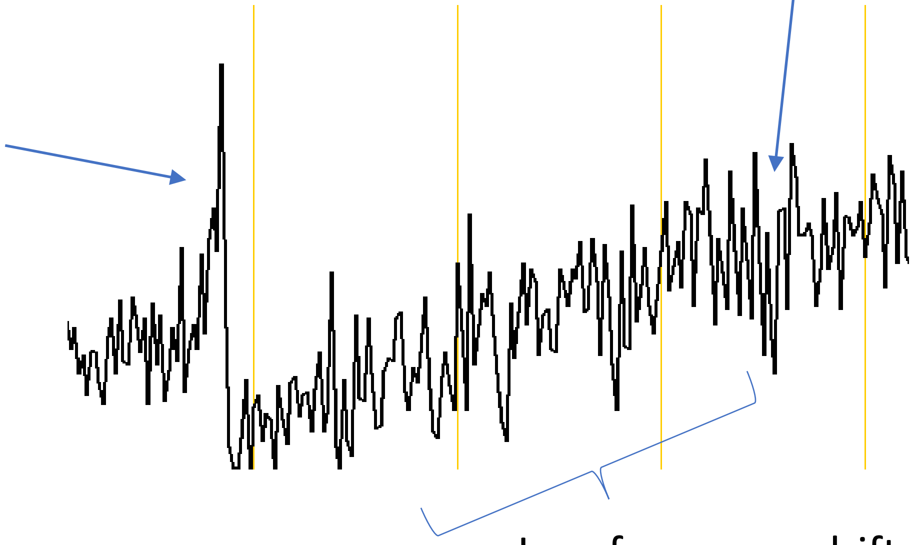
Bandpass filtering

BOLD fMRI pre-processing

- BOLD images suffer from artifacts which can impact the analysis to varying degrees
- Artifacts must be removed in a specific order and using specific tools
- Processing pipeline:
- 1) EPI distortion artifact correction
- 2) rigid body head motion correction
- 3) regress nuisances in time series, including:
  - Low-frequency drift
  - Residual spikes due to head motion
  - Cardiac and respiratory effects
- Finally, images are typically smoothed to remove random noise and aligned with anatomy for subsequent analysis
  - Alignment with anatomy can be done at any point in the analysis (order is not important)

Some artifacts:





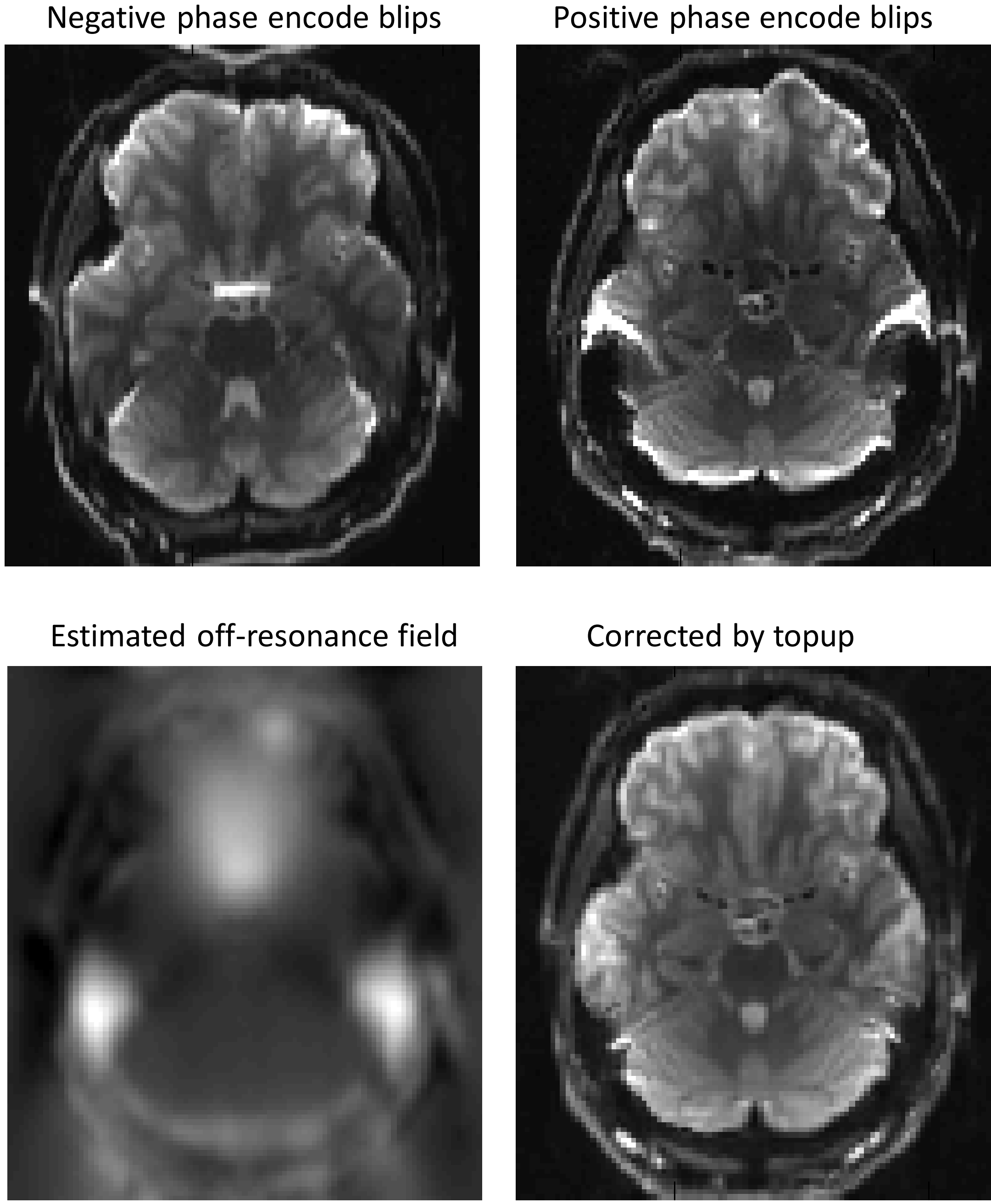
Cardiac pulsation

Head motion spike

Low frequency drift

EPI distortion correction

- Echo-planar imaging (EPI) causes geometric distortions stemming from inhomogeneities in the underlying B0 field, which is in turn due to varying magnetic susceptibilities of air, bone, and tissue
- EPI distortions alter anatomic shape of images, making anatomically accurate measurements difficult and complicating multimodal investigations
- *topup* is a tool for estimating and correcting susceptibility-induced distortions (from FSL libraries)
- topup can be applied to both diffusion and BOLD fMRI images, (both use echo-planar imaging (EPI))
- **topup**:
  - 1) acquire two extra images during scan time, with reversed phase-encoding gradients.
    - Distortions in each image are identical but in opposite directions
  - 2) apply symmetric registration with transformations only allowed along phase-encode direction
    - This warps images to each other to 'meet in the middle' thereby obtaining distortion corrected images with no signal loss
- Some researchers don't do this step (but they should!)



Negative phase encode blips

Positive phase encode blips

Estimated off-resonance field

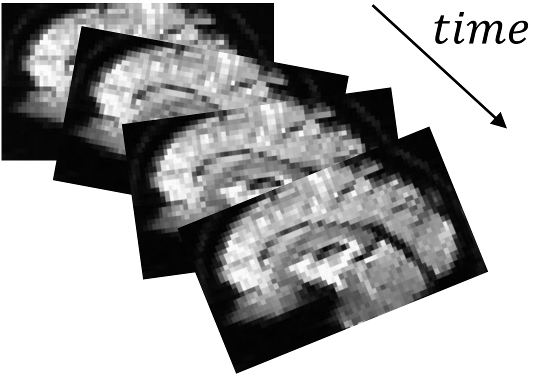
Corrected by topup

N.B. - above images are diffusion MRI, but the principle is same for BOLD

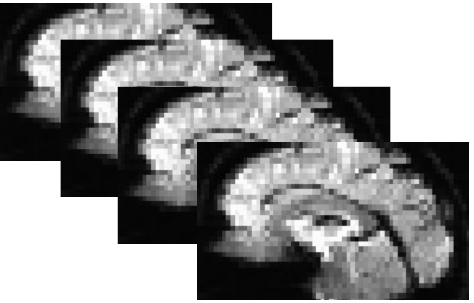
Rigid body motion correction

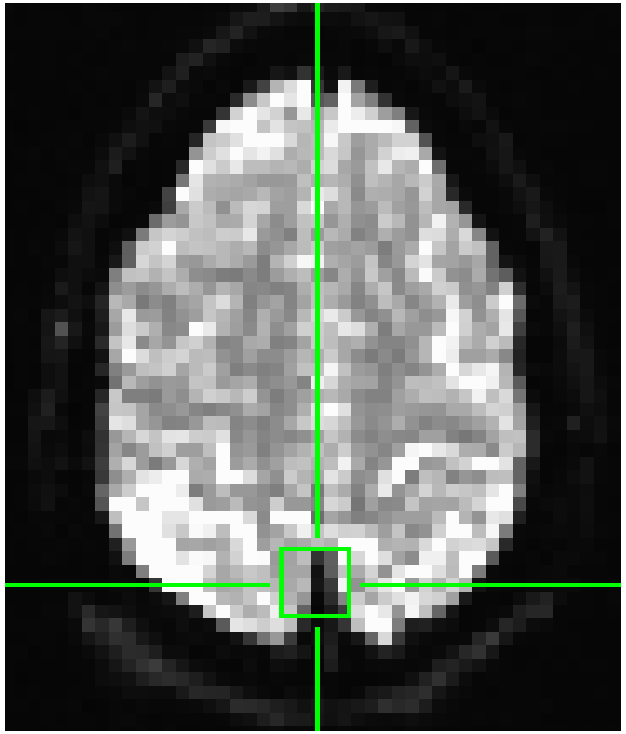
- During the scan, subjects will move their head, causing large artifacts in BOLD time series especially in areas with high contrast difference (edge of brain, near ventricles, etc.)
- Can reduce these artifacts by applying rigid body motion correction to every volume in the BOLD time series
    - Typically, the first or last volume in the time series is selected as the reference image, and all other volumes are aligned to this reference volume

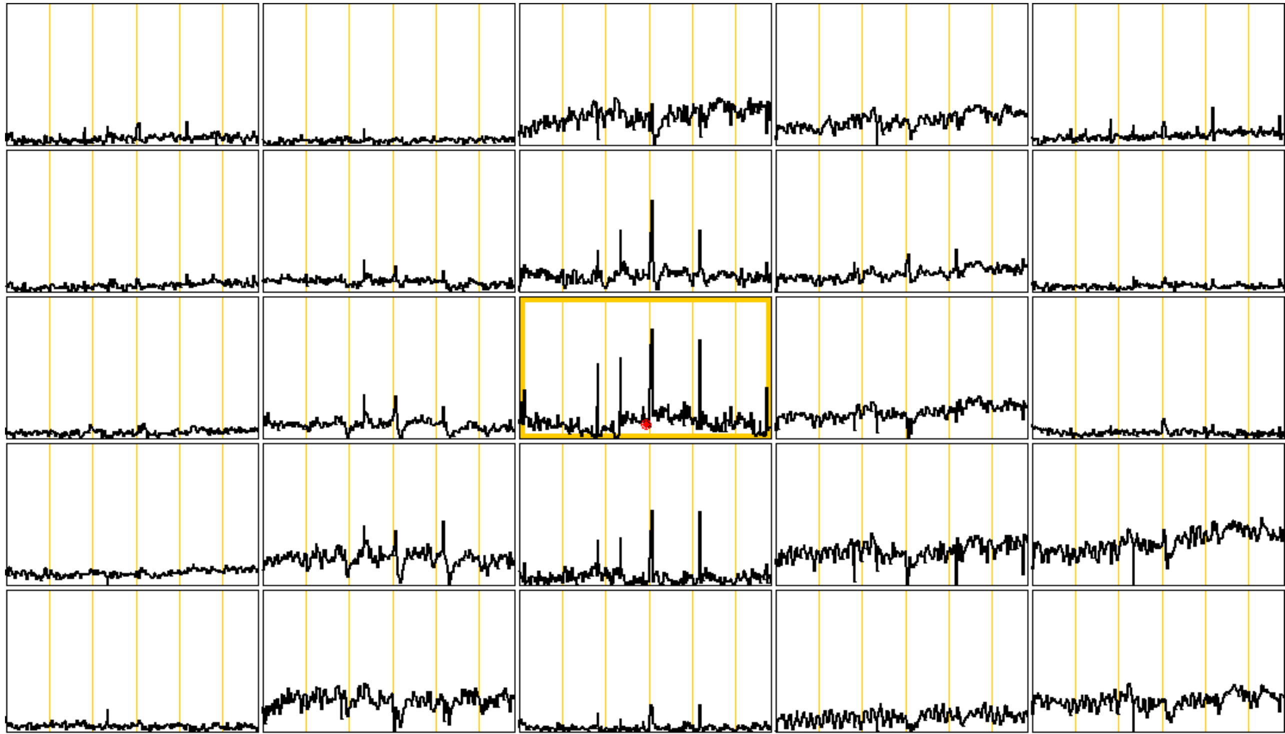
Not motion corrected



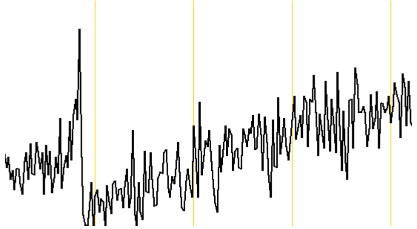
motion corrected







Nuisances in time series

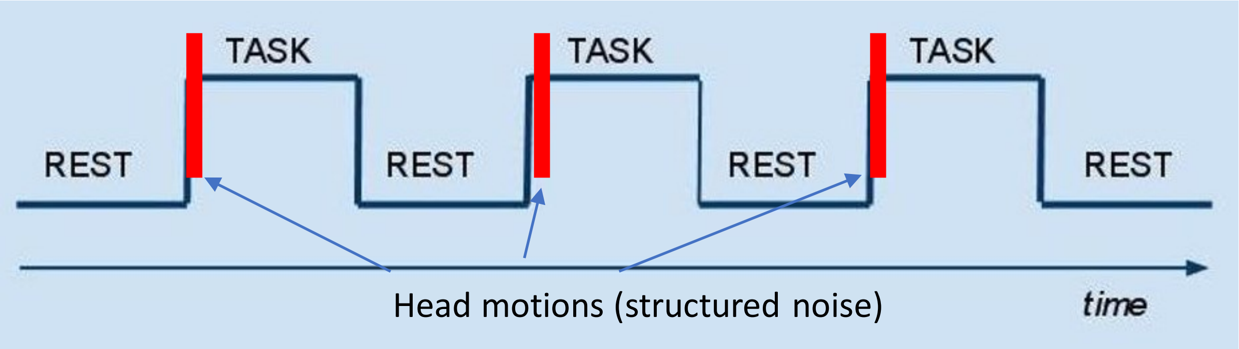


- Many artifacts in time series, both physiological (caused by participant) and non-physiological (caused by scanner hardware)
- These ‘nuisances’ present strong confounds for any type of BOLD analysis
- Confounds for event-related analysis (a)
  - Stimulus-locked head motion, respiratory changes, heartrate changes, etc.
- Confounds for resting-state analysis (resting state functional connectivity (RSFC))
  - Generally, neuronal changes in the resting state are quite weak
  - Physiological artifacts are likely to dominate the signal, and hence, RSFC measures
  - b) correlation of respiration belt with BOLD in each voxel

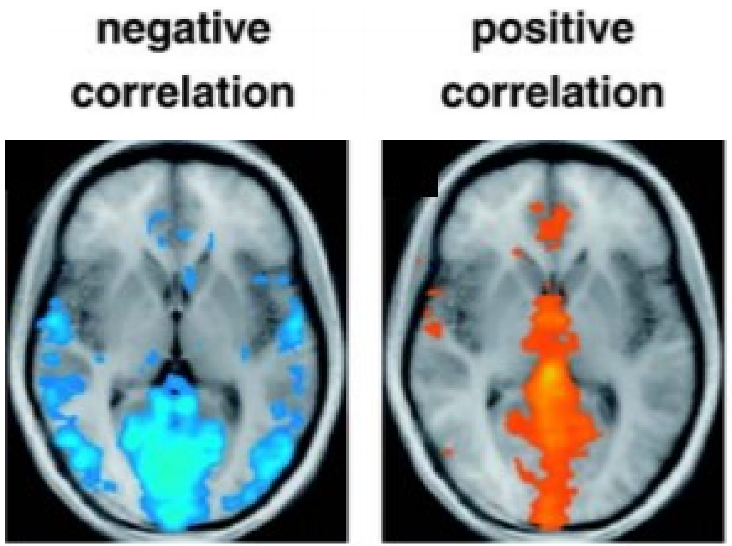
a) **Typical event-related design for BOLD fMRI**

What happens if subject moves head every time task appears?

Will result in correlations between head motion and task!



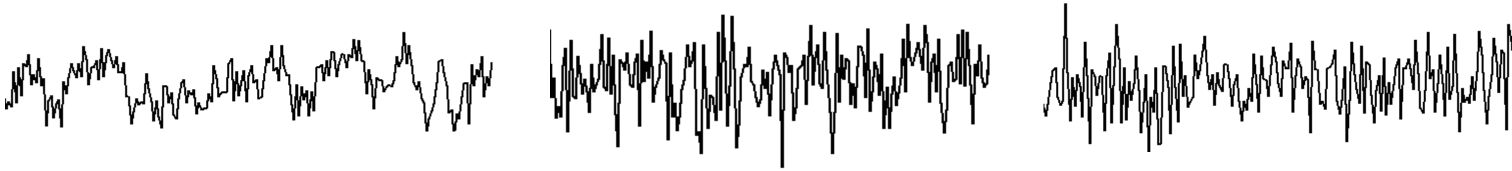
b)

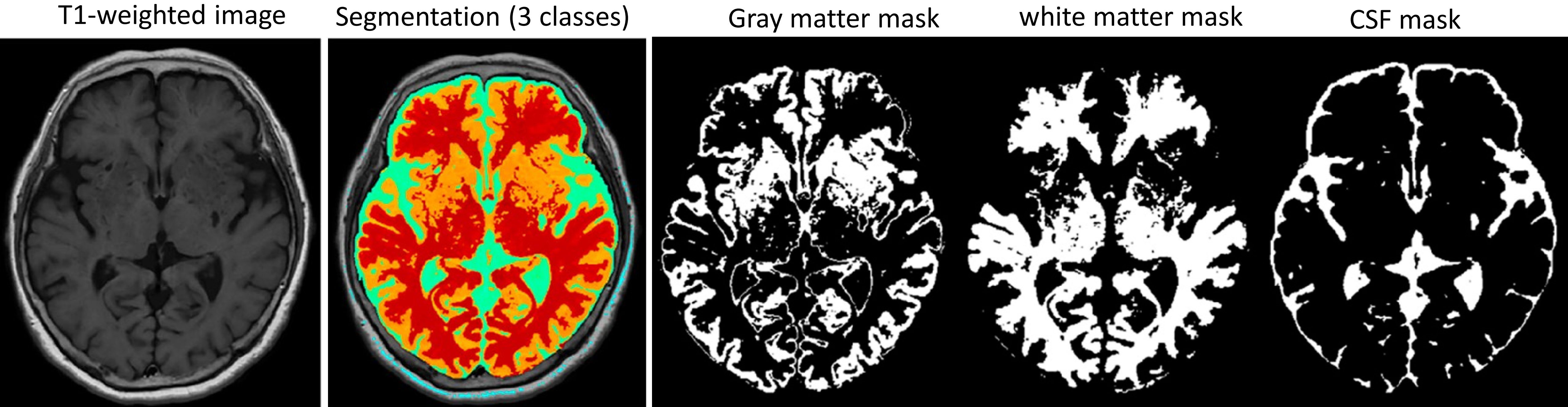


Nuisance regressors

- How to get the nuisance signals (respiration, heartbeat, motion, etc.)
- One common approach is to *regress out* the average signal in specific tissue type (usually white matter)
    - White matter used because it has far less veins, so BOLD signal in white matter is mostly noise
- Another approach is to record the signals directly using heartrate monitor, regress out heartrate
- A third approach is to regress out the average signal across entire brain (global signal)

Time series from single voxel in each of the masks (typically, we get an average time series by combining all voxels in mask) →





T1-weighted image

Segmentation (3 classes)

Gray matter mask

white matter mask

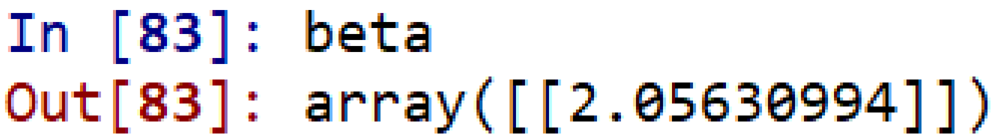
CSF mask

Nuisance regression

- How to remove the nuisance signals from the overall signal?
- Use the *general linear model:*
- $Y = X\beta + e$
- Where statistical assumptions and requirements are as follows:
  - 1) system must be linear
  - 2) X is a design matrix containing linearly independent explanatory variables
  - 3) Y is (linearly) dependent on explanatory variables contained in X through weights β, these weights are the model parameters
  - 4) the model is complete, such that explanatory variables explain deterministic variance in Y leaving only residual errors. These errors should ideally be estimates of e
  - 5) true errors e are independent and identically distributed, and have constant variance.
- Typically, design matrix X is formed from nuisance regressors reflecting head motion and physiological noise sources (heartbeat, respiration) while the observations Y are the BOLD fMRI time series data
- The denoised time series is defined as the residual of the model fit
- $\hat{\beta} = (X^T X)^{-1} X^T Y$, where $\hat{\beta}$ is the approximation to β

Nuisance regression example

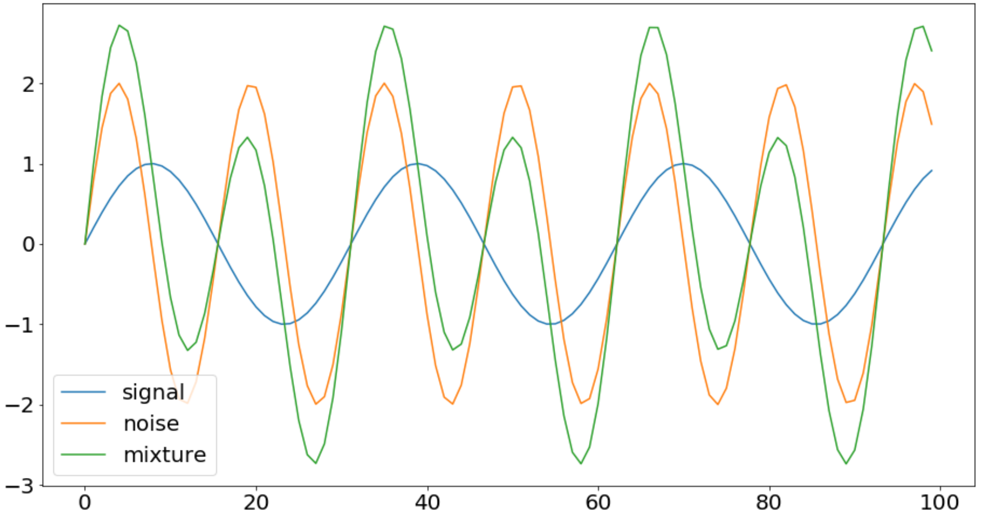
```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams.update({'font.size': 20})

signal = np.expand_dims(np.sin(np.linspace(0,20,100)),axis=1)
noise = np.expand_dims(np.sin(np.linspace(0,40,100)),axis=1)

mixture = signal + noise*2

plt.plot(signal); plt.plot(noise*2); plt.plot(mixture);
plt.legend(['signal','noise','mixture'])

beta = np.dot(np.dot(np.linalg.inv(np.dot(noise.T,noise)),noise.T),mixture)

denoised = mixture - np.dot(noise,beta)
```



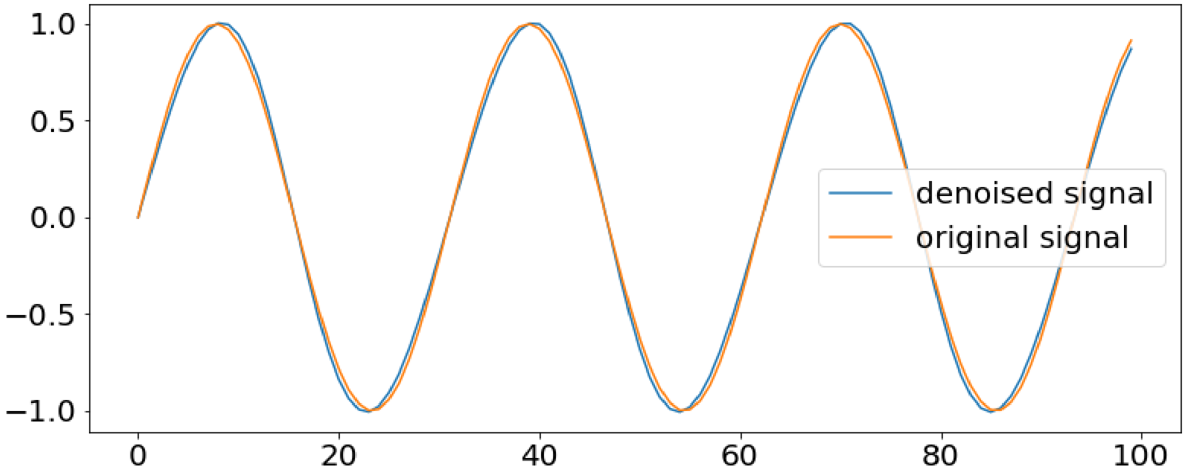
```
In [83]: beta
Out[83]: array([[2.05630994]])
```

Create a couple sinewaves (signal and noise), mix them with different amplitude coefficients, and then get back the coefficient using leastsq and use it to recover the original signal

$$\hat{\beta} = (X^TX)^{-1}X^TY\,,$$



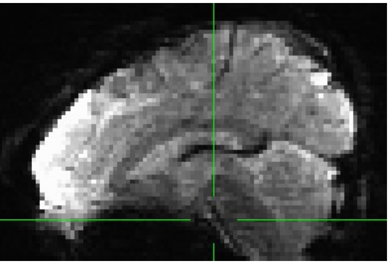
```
plt.plot(denoised); plt.plot(signal)
plt.legend(['denoised signal','original signal'])
```

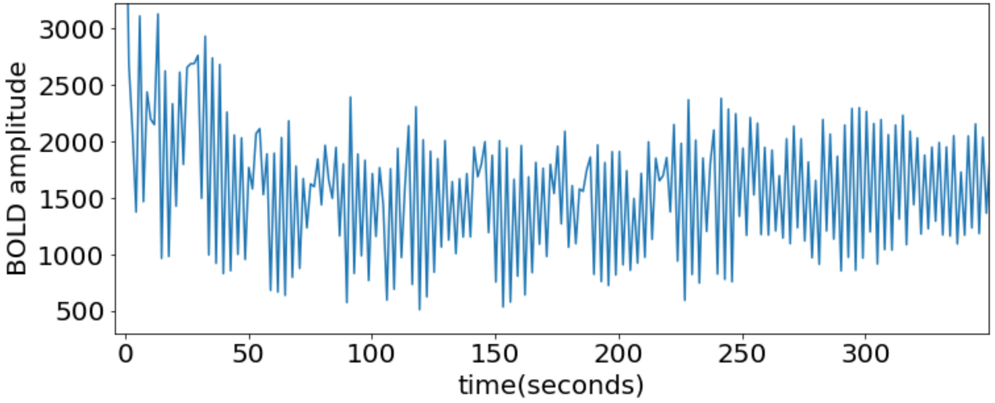


Frequency content of BOLD signals

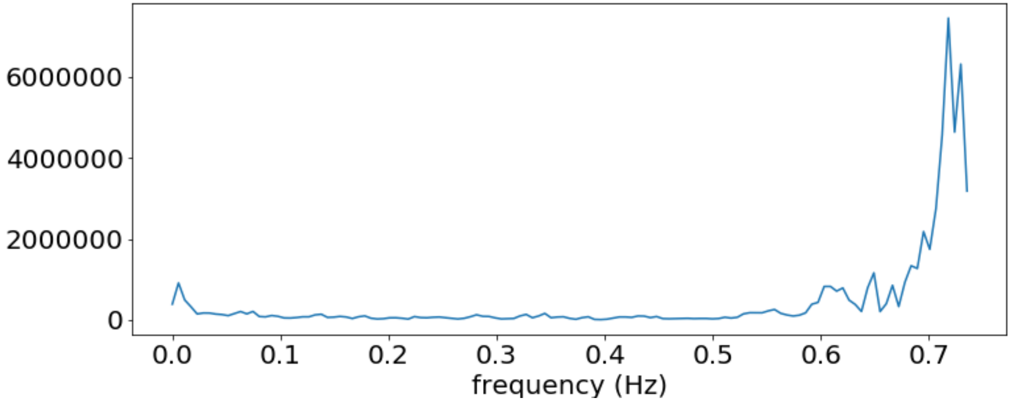
- BOLD images typically sampled every ~1 second (but can be as fast as every 500ms, or as slow as every 3.5-4 seconds).
- Frequency content of signal can be quite different depending on what brain area we look at:

Voxel in neck artery

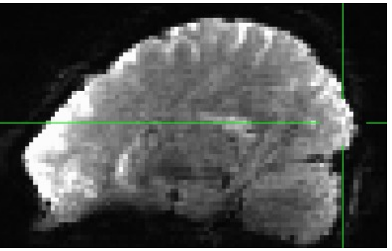


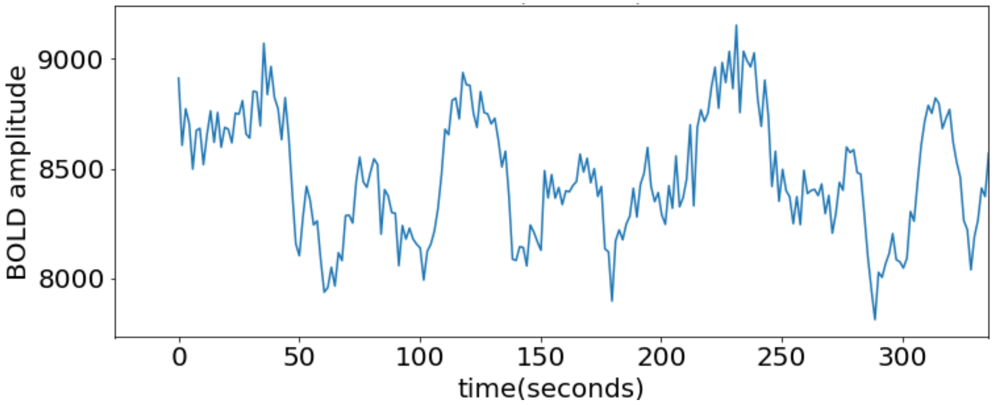


FFT →

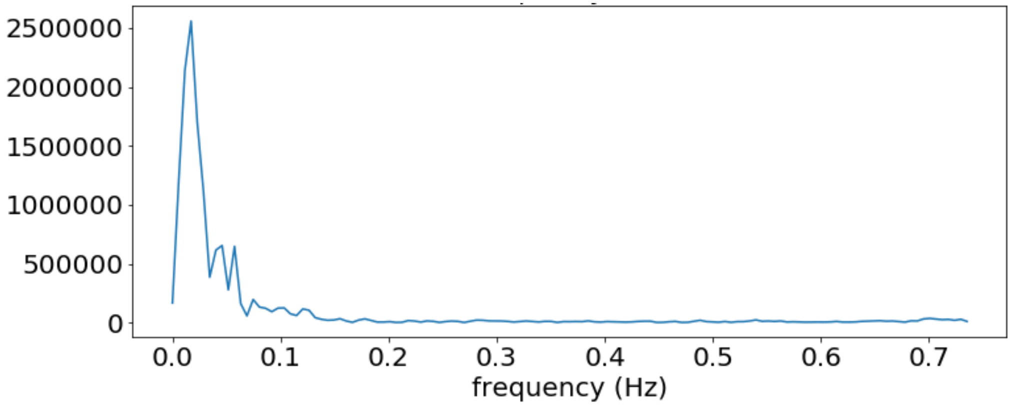


Gray matter voxel



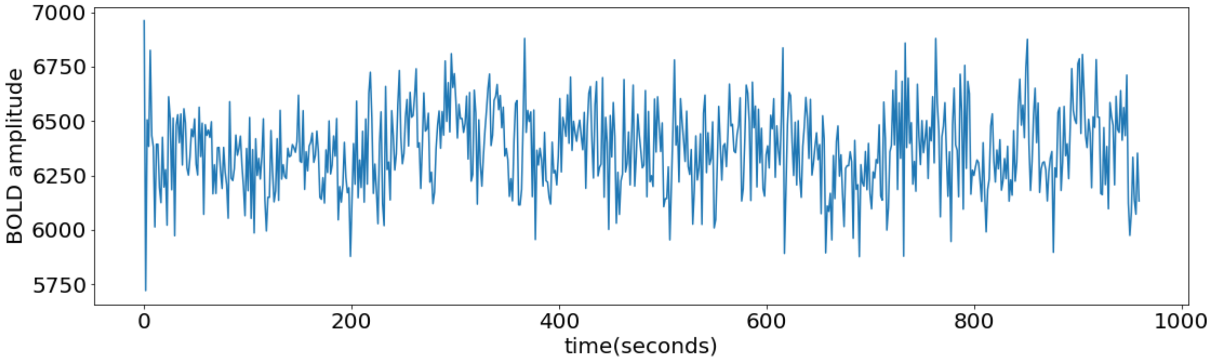


FFT →



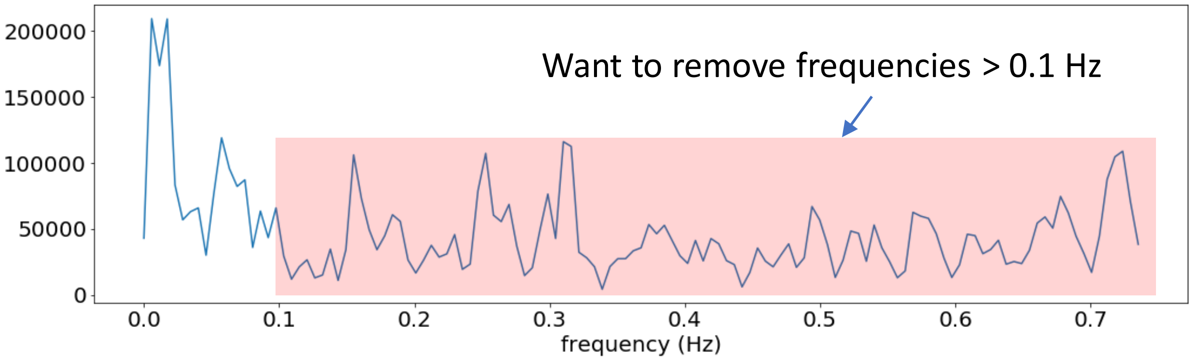
Bandpass filtering (basic approach)

Time series (before applying cutoff)



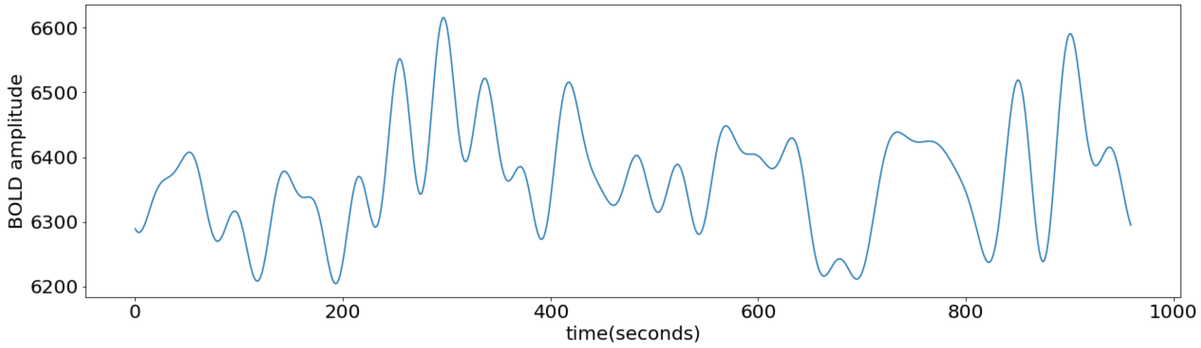
FFT

Power spectrum (before applying cutoff)

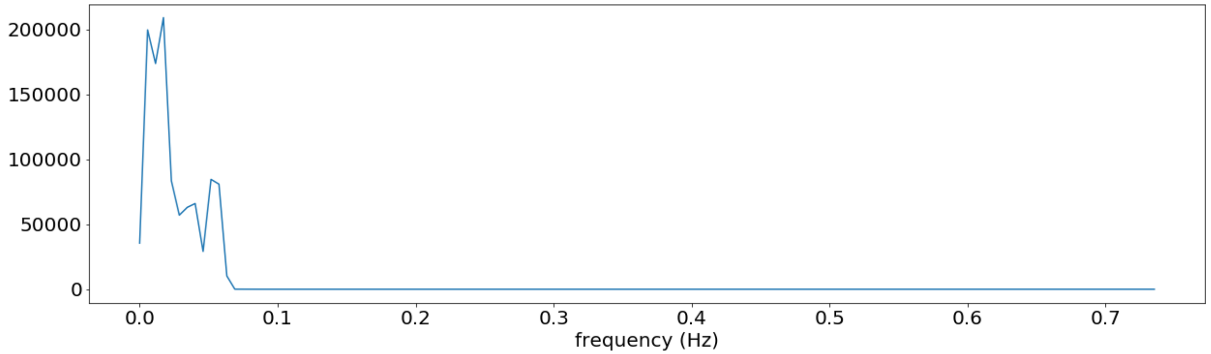


Want to remove frequencies > 0.1 Hz

Time series (after applying cutoff)



Power spectrum (after applying cutoff)

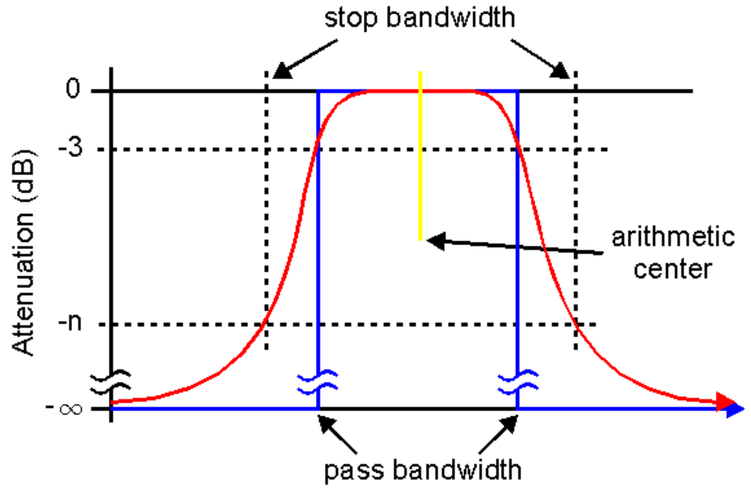


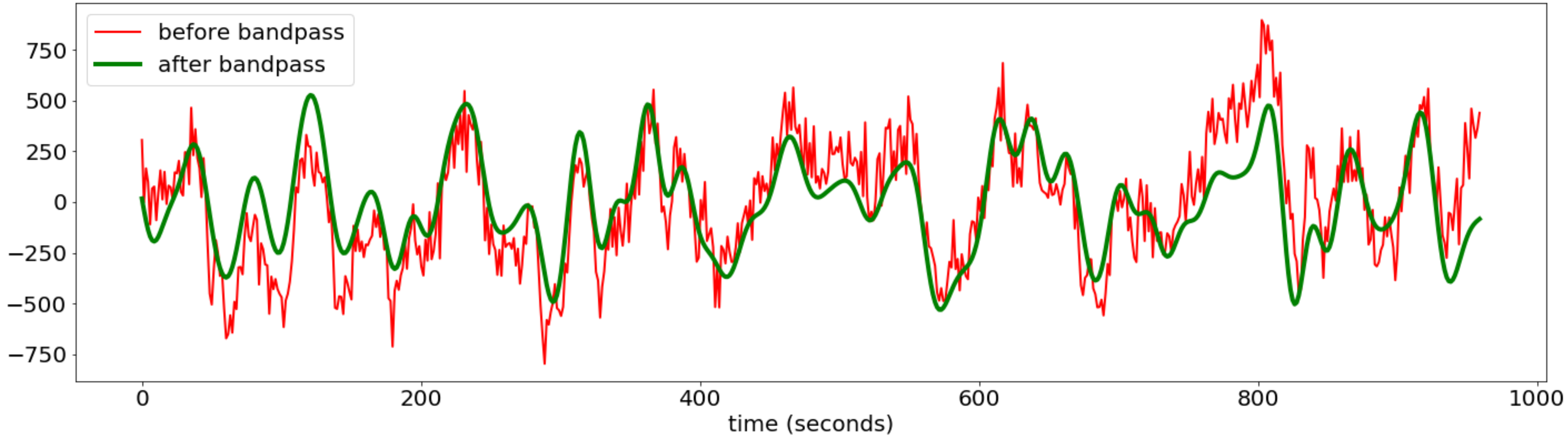
Bandpass filtering using butterworth filter

- Because our signals are not infinite in length, we don’t use box filter as in previous slide (produces ‘gibbs ringing’)
- Instead, better to use a *butterworth filter*

“an ideal filter should not only completely reject the unwanted frequencies, but should also have uniform sensitivity for the wanted frequencies” –S Butterworth

```
from scipy.signal import butter, lfilter

def butter_bandpass(lowcut, highcut, fs, order=5):
    nyq = 0.5 * fs
    low = lowcut / nyq
    high = highcut / nyq
    sos = butter(order, [low, high], btype='band',output='sos')
    return sos

sos = butter_bandpass(0.01,0.1, 1/0.68,order=5)
ts = img[-57,86,31,:]                  # the BOLD time series
y_sos = signal.sosfiltfilt(sos, ts) # filtered signal
```





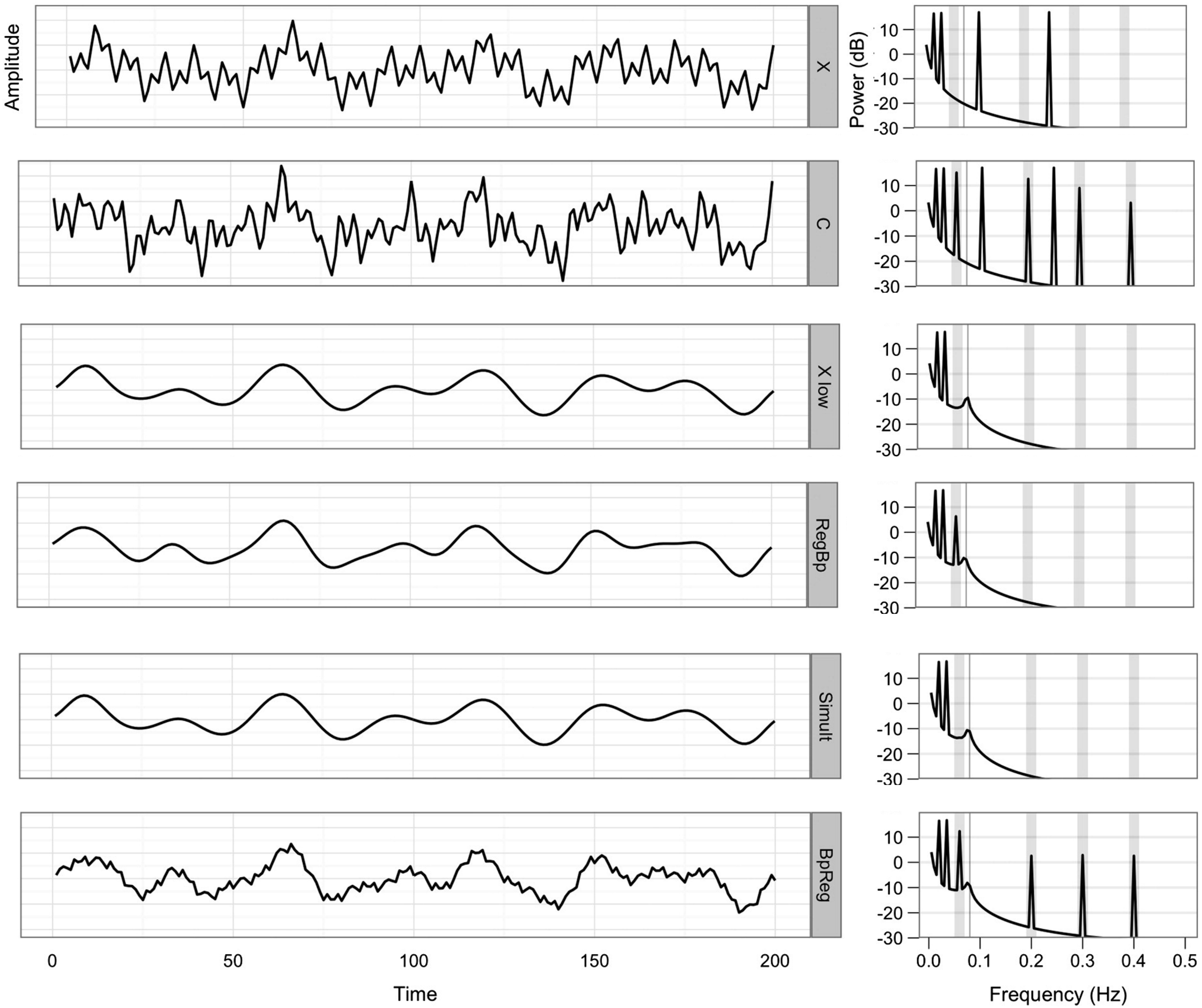
Typically, in analyzing BOLD signals, a bandpass from **0.01 to 0.1** Hz is applied, meaning all frequencies below 0.01 Hz and all frequencies above 0.1 Hz are removed from signal

Combining bandpass filtering and nuisance regression

- Bandpass filtering and nuisance regression intended to reduce noise in BOLD
- When BOLD data are filtered but nuisance regressors are not, noise is poorly controlled
- This approach ↑ also reintroduces synchronous noise into BOLD data
- *Simultaneous* bandpass filtering and regression eliminates this source of bias. Why?
- Bandpass filtering and nuisance regression can both be conceptualized as linear filters of the data
  - Filtered signal y is a linear transformation of original signal x: $y_t = F_t\{x\}$, where the transformation operator F linearly maps every value of x to a transformed value of y
  - When two linear filters are orthogonal to each other, the output signal is the same regardless of which filter is applied first
  - However, nuisance regressors (motion parameters, heartbeat/respiratory parameters) and bandpass filter coefficients are unlikely to be orthogonal because noise sources typically include power in frequencies in the stopband (frequencies to be suppressed)
  - This leads to statistical dependence between spectral filter and nuisance regressors, thus the order in which bandpass filtering and nuisance regression are applied is important

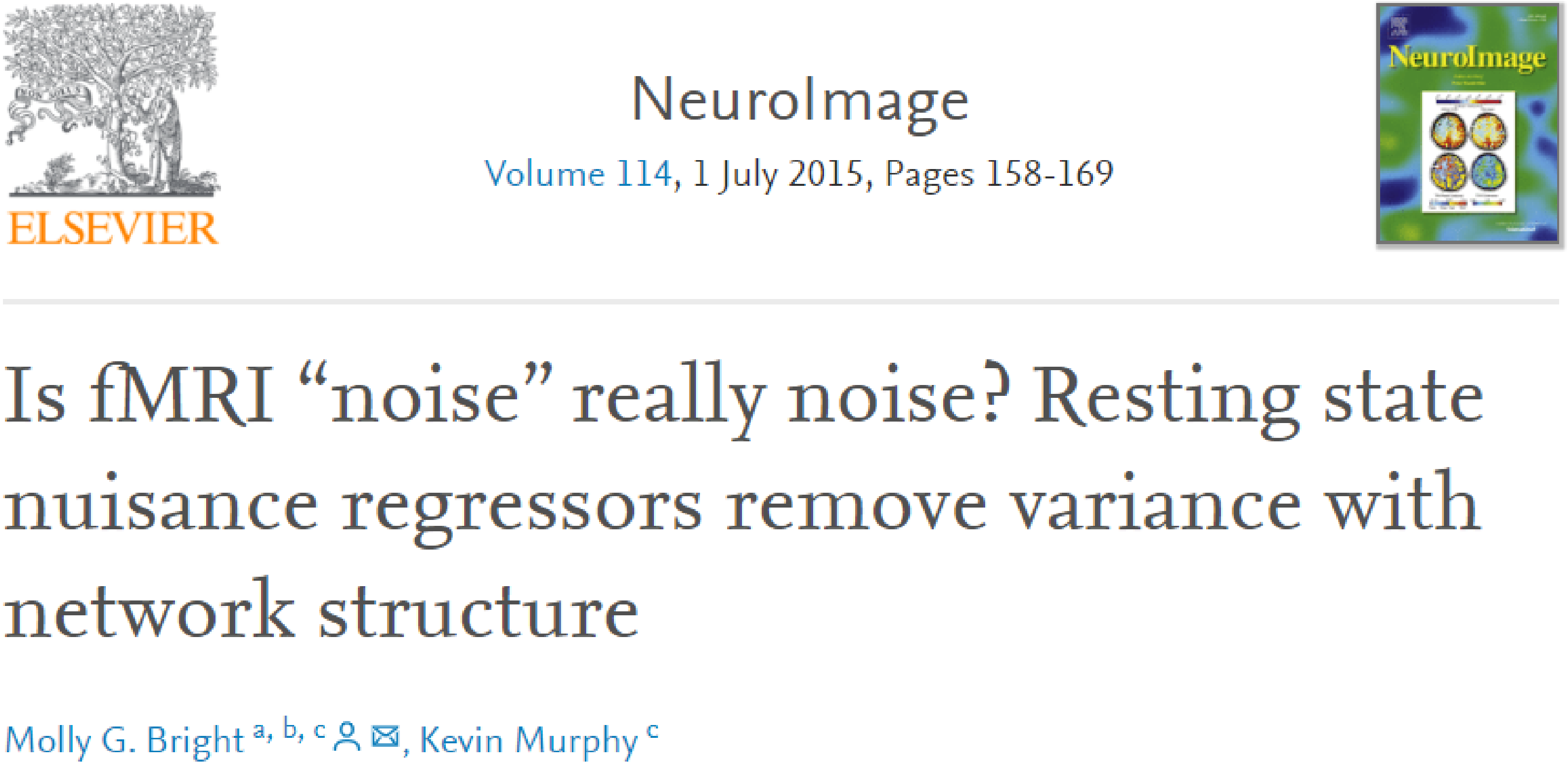
Why nuisance regression after bandpass filtering is bad

- Regression of bandpass filtered BOLD on full bandwidth nuisance regressors results in distortion of BOLD signal:
- Consider a synthetic BOLD time series X composed of four sinusoidal components at 0.02, 0.035, 0.11, and 0.25 Hz:
- $X = \sum_{t=1}^{n} \sin\left(2\pi t \frac{4}{n}\right) + \sin\left(2\pi t \frac{7}{n}\right) + \cos\left(2\pi t \frac{22}{n}\right) + \cos(2\pi t \frac{50}{n})$
- Conventional processing pipeline would first apply a bandpass filter from 0.01 to 0.1 Hz, retaining the two low-frequency components at 0.02 and 0.035 and removing 0.11 and 0.25 Hz (X_{low})
- Next, consider periodic signal M representing participant motion during the acquisition, composed at sinusoidal frequencies at 0.06, 0.2, 0.3 and 0.4 Hz:
- $M = \sum_{t=1}^{n} \cos\left(2\pi t \frac{12}{n}\right) + \cos\left(2\pi t \frac{40}{n}\right) + \sin\left(2\pi t \frac{60}{n}\right) + \cos(2\pi t \frac{80}{n})$
- This motion signal mixed with original signal to yield motion-corrupted BOLD signal:
- $C = X + 0.8M_{0.6\ Hz} + 0.6M_{0.2\ Hz} + 0.4M_{0.3Hz} + 0.2M_{0.4Hz}$
- This represents what is typically seen in motion-corrupted BOLD data
- To eliminate bivariate association between M and C, use ordinary least-squares regression: $\boldsymbol{Y}_t = \boldsymbol{X}_{t \times p}\beta_{p \times 1} + \varepsilon_t$
  - Where there are t observations (time points), p nuisance signals, and the residual represents discrepancy between model estimate of Y_t and its observed value $\varepsilon_t = \boldsymbol{Y}_t - \hat{\boldsymbol{Y}}_t$
- **Three approaches tested on this synthetic dataset:**
- 1) BpReg – bandpass filtering of C followed by regression on M
- 2) RegBp – regression of C on M followed by bandpass filtering
- 3) Simult – simultaneous regression and bandpass filtering
- Full motion removal is only achieved with ‘Simult’ approach



Nuisance regression is controversial

- Despite many years of work on nuisance regression, the controversies persist
- Example: recent article showing how the noise maps which are removed by nuisance regression actually contain important network structure



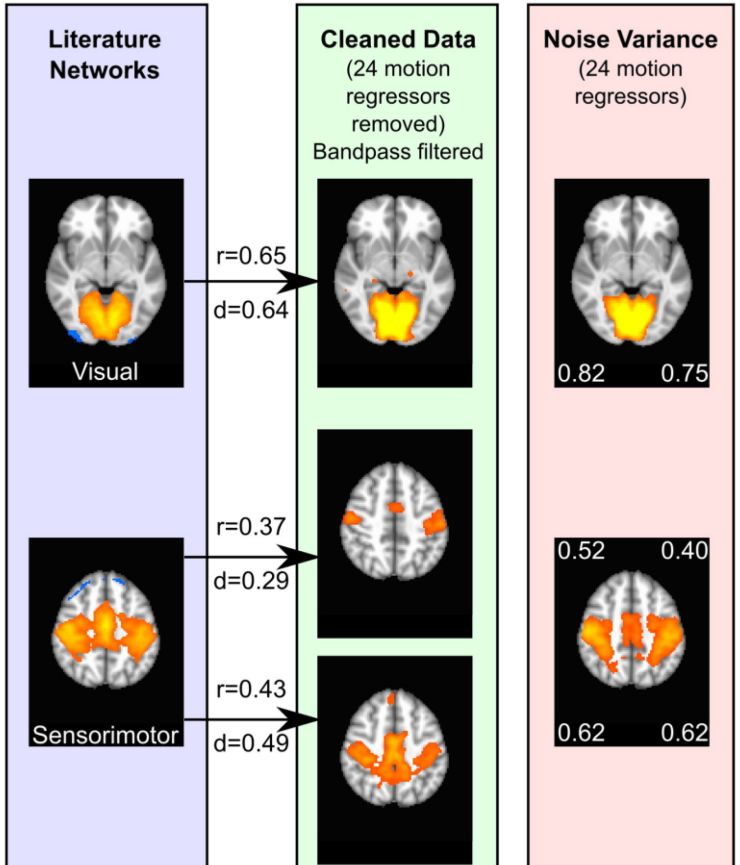
ELSEVIER

NeuroImage

Volume 114, 1 July 2015, Pages 158-169

Is fMRI "noise" really noise? Resting state nuisance regressors remove variance with network structure

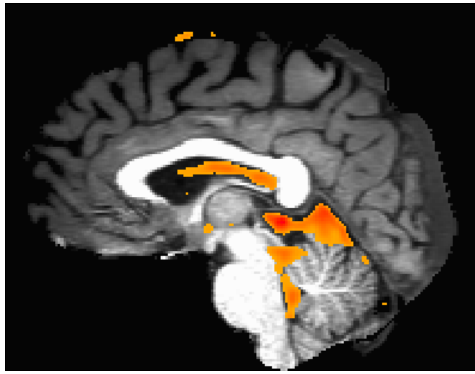
Molly G. Bright [a], [b], [c], Kevin Murphy [c]

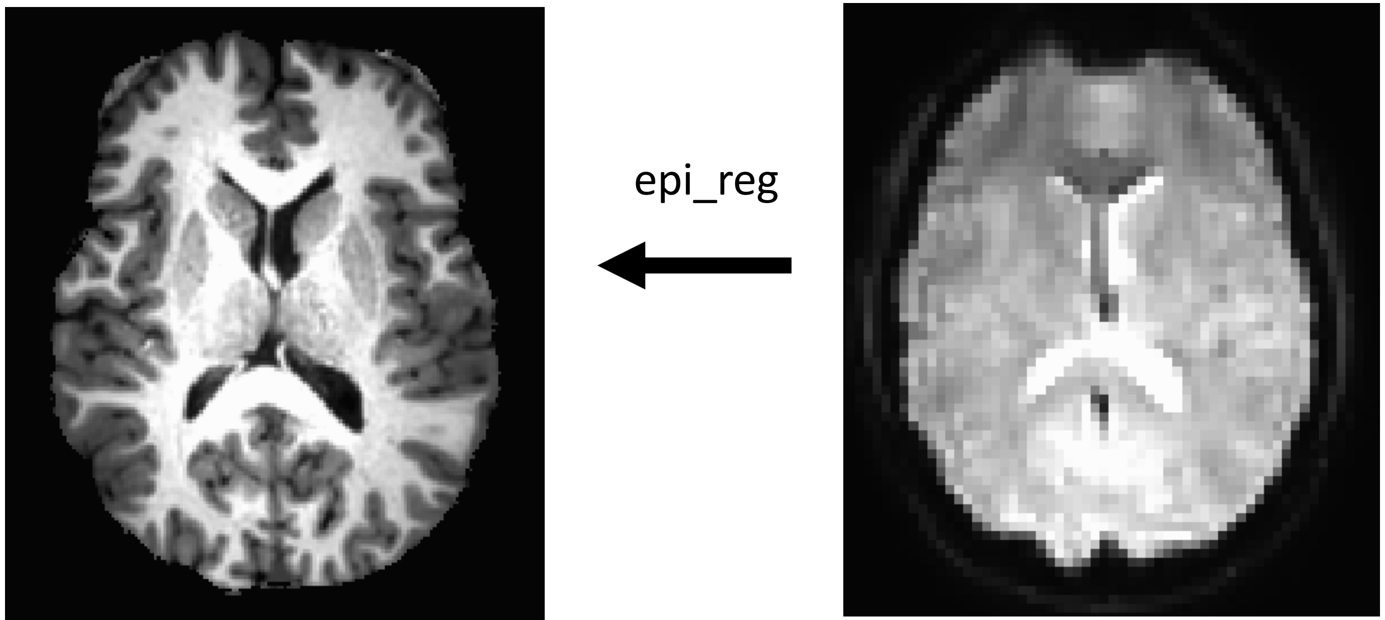

Literature Networks
Cleaned Data (24 motion regressors removed) Bandpass filtered
Noise Variance (24 motion regressors)
Visual
Sensorimotor
r=0.65
d=0.64
r=0.37
d=0.29
r=0.43
d=0.49
0.82
0.75
0.52
0.40
0.62
0.62

Alignment of BOLD to anatomy

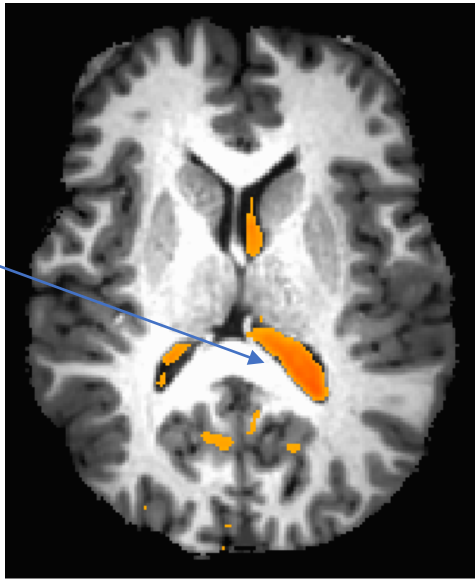
Why align BOLD to anatomy? Because once pre-processing is finished, we will use anatomy to define regions of interest and be more precise about where in the brain the BOLD signals are coming from.

- Aligning BOLD image to T1-weighted image is a challenging multi-modal image registration problem
  - Different resolution
  - Different contrast (T1 (anatomy) vs T2* (BOLD))
  - Distortions in the BOLD
- Several tools exist, but FSL's epi_reg seems to perform best
- When registration is done, overlay the BOLD on the T1 and check the alignment in the ventricles by increasing threshold on the BOLD overlay

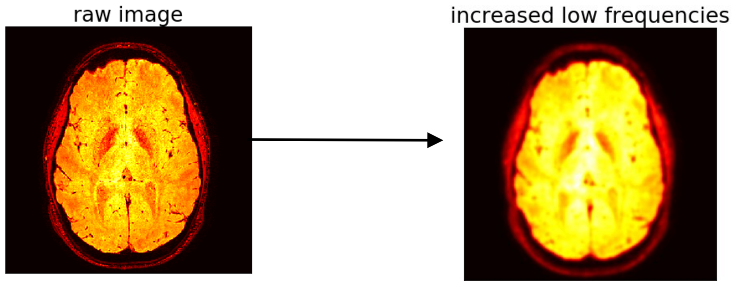




Example of a good alignment: ventricles match well
orange = BOLD
gray = T1

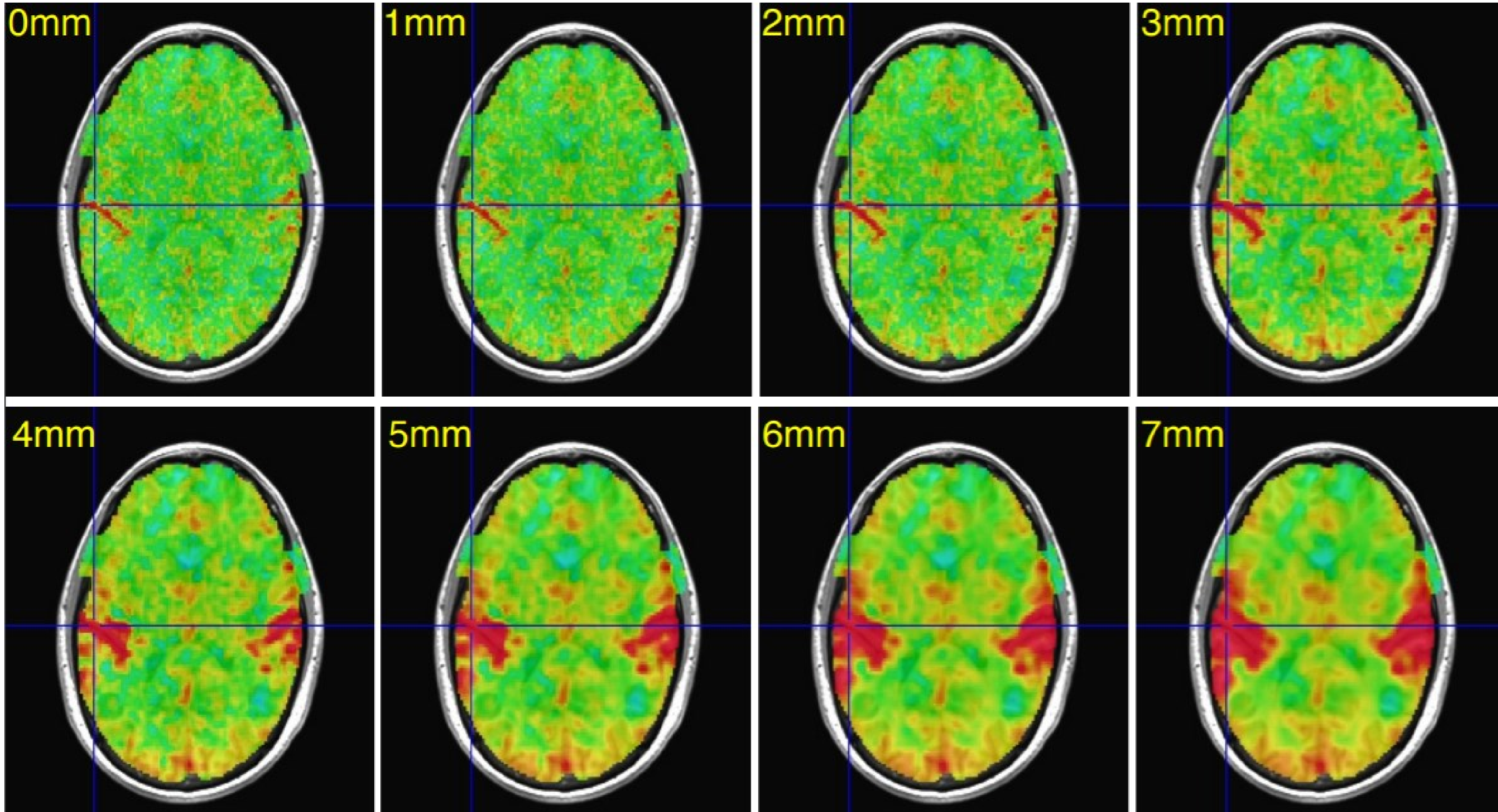


Spatial smoothing



- One final step which is often performed (but sometimes not) is *spatial smoothing*
- Spatial smoothing helps to reduce noise and can also aid in multi-subject analysis
- Example using 8 different gaussian smoothing kernels with increasing full-width half-maximum (FWHM)

- Averaging locally cancels out noise and increases coherent signals
- Must use smoothing *after* tissue-based signal extraction (so we don’t mix the signal with the noise)
- Example: resting state functional connectivity map in a single subject (seed voxel at crosshair) for different levels of spatial smoothing
    - More smoothing = more widespread correlations
    - However, this is not *necessarily* a good thing (could indicate increased contribution of physiological artifacts)



Full example

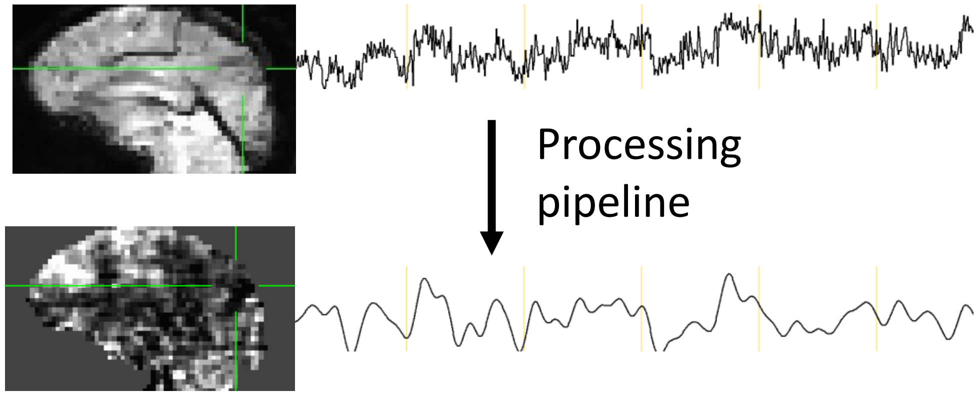
Data taken from sub-01 of the MB-EPI comparison dataset at Openneuro:

https://openneuro.org/datasets/ds002603/versions/1.0.0

Recap:

1) Epi distortion correction (not shown here)
2) Register to anatomy
3) rigid body head motion correction
4) Nuisance regression + bandpass
5) Spatial smoothing
6) Your analysis of choice!

- Let’s look at a small shell script (.sh) using the AFNI and FSL toolboxes (assignment 3 part 1)



```
# must have both 'afni' and 'fsl' software packages installed to run

# despike - remove large spikes in time series
3dDespike -prefix despike.nii.gz bold.nii.gz
# motion correction - align all volumes
3dvolreg -prefix volreg.nii.gz -1Dfile motion_params.1D bold.nii.gz
# Brain Extraction Tool (bet) - removes skull
bet t1.nii.gz bet.nii.gz -f 0.3
# FMRIBs Automated Segmentation Tool (fast) - segment tissue types
fast -g -o fast bet.nii.gz
3dTstat -prefix mean_bold.nii.gz volreg.nii.gz # get mean of BOLD (for epi_reg)
# epi_reg - does the multimodal registration between BOLD image and T1 (uses white matter from 'fast')
epi_reg --epi=mean_bold.nii.gz --t1=t1.nii.gz --t1brain=bet.nii.gz --out=epireg --wmseg=fast_seg_2.nii.gz
# invert the bold->T1 matrix (so can bring white matter into BOLD space
convert_xfm -inverse -omat t1_2_epi.m epireg.mat
# bring white matter (from fast) into BOLD space:
flirt -in fast_seg_2.nii.gz -ref mean_bold.nii.gz -applyxfm -init t1_2_epi.m -interp nearestneighbour -out
white_matter_in_bold.nii.gz
# get the average BOLD signal in the white matter mask:
3dmaskave -quiet -mask white_matter_in_bold.nii.gz volreg.nii.gz > white_matter_signal.1D
# create a mask over the entire brain (for input to 3dTproject below)
3dAutomask -prefix mask.nii.gz mean_bold.nii.gz
# final step: regress out nuisance (white matter signal, motion) and do bandpass filtering + blurring
3dTproject -prefix clean_bold.nii.gz -input volreg.nii.gz -ort white_matter_signal.1D -ort motion_params.1D -passband 0.01 0.1 -
mask mask.nii.gz -blur 4
```