

The minimum you need to know to classify electro-physiological brain signals

Jason Farquhar
Dept. Cognitive Artificial Intelligence (CAI)
Donders Center for Cognition
<J.Farquhar@donders.ru.nl>

Radboud University Nijmegen



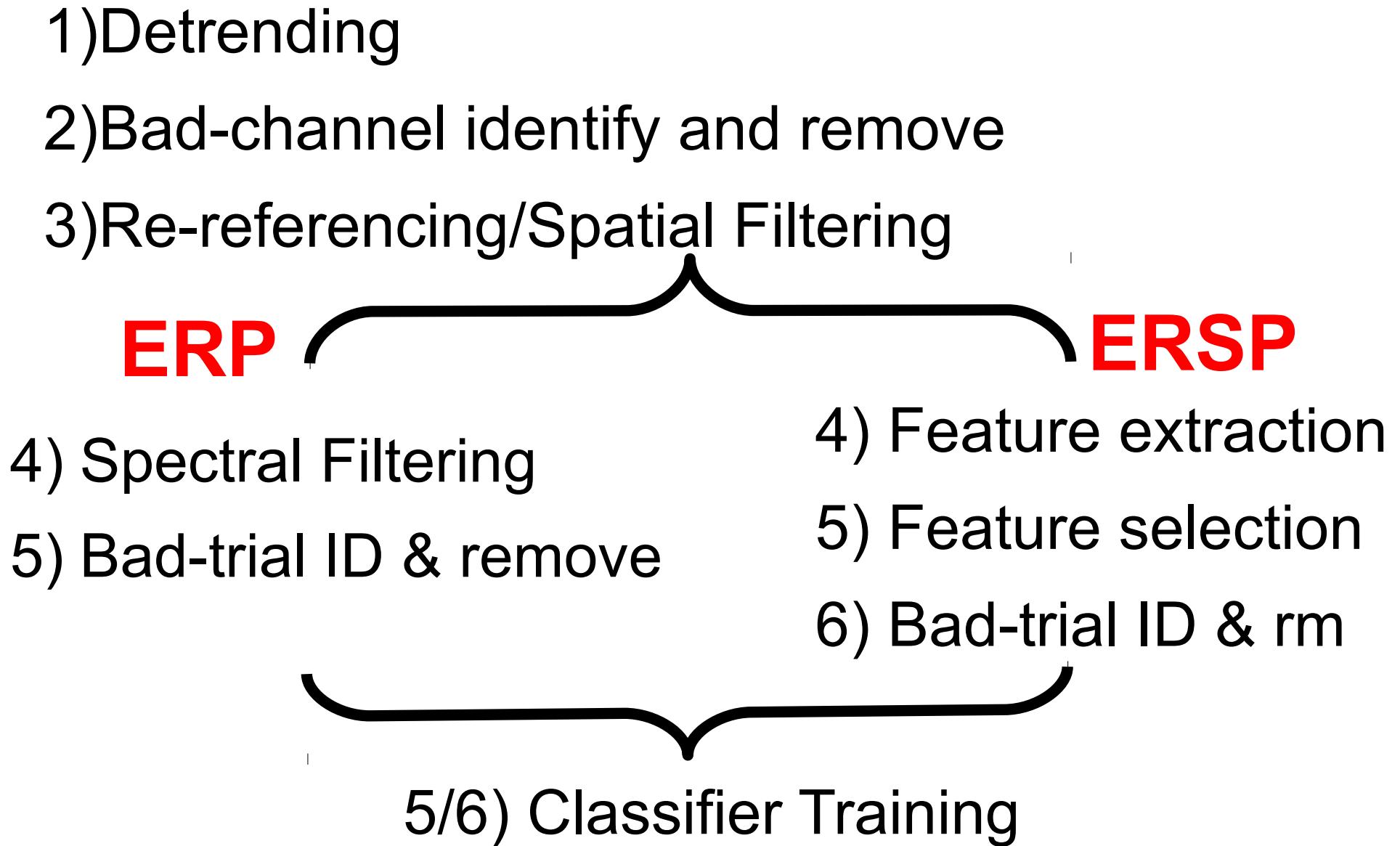
Learning Goals

- Understand:
 - How the on-line real-time nature of BCI necessitates slightly different methods than off-line analysis
 - The purpose of each of the steps needed to process electrophysiological data
 - The trade-offs inherent in skipping steps and/or using different methods at each step
- Know what the following terms mean:
 - Detrending, bad-channel identification, re-referencing, spectral-filtering, bad-trial identification, classifier-training

Today's Plan

- Overview : the BCI cycle
- Overview : 2 common BCI signals, ERPs and ERSPs
- Overview : Properties of electrophysiological data and artifacts
- A simple BCI signal analysis pipeline:
 - Detrending / High-pass filtering – What? Why? How?
 - bad-channel removal – What? Why? How?
 - re-referencing/spatial-filtering – What? Why? How?
 - Spectral-filtering – What? Why? How?
 - Bad-trial identification/removal – What? Why? How?
 - ERSP feature-extraction & selection – What? Why? How?
 - Classifier training – What? Why? How?
- Summary

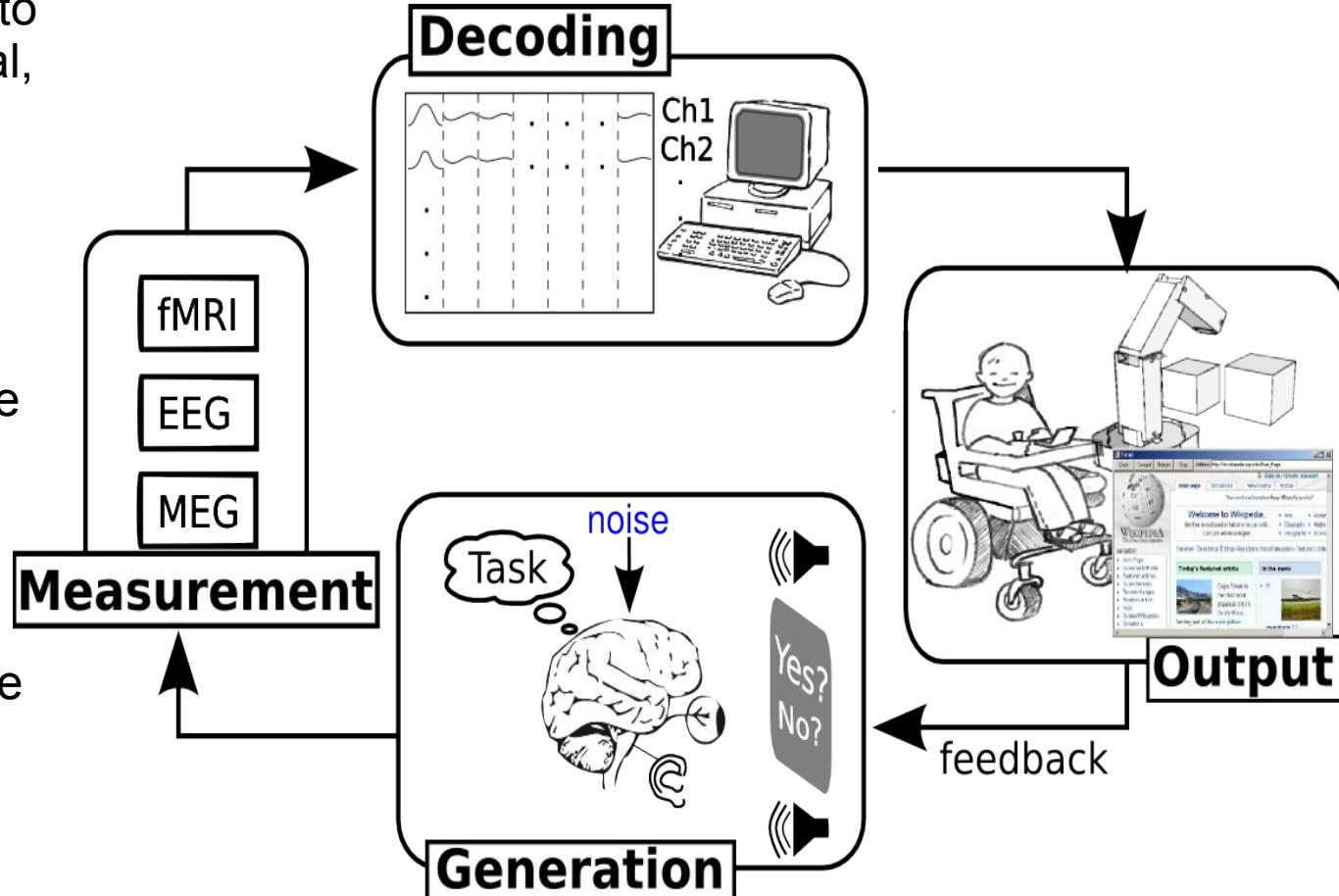
BCI signal analysis pipeline



The BCI Cycle

Fundamentally, BCI is a simple(?) engineering problem;

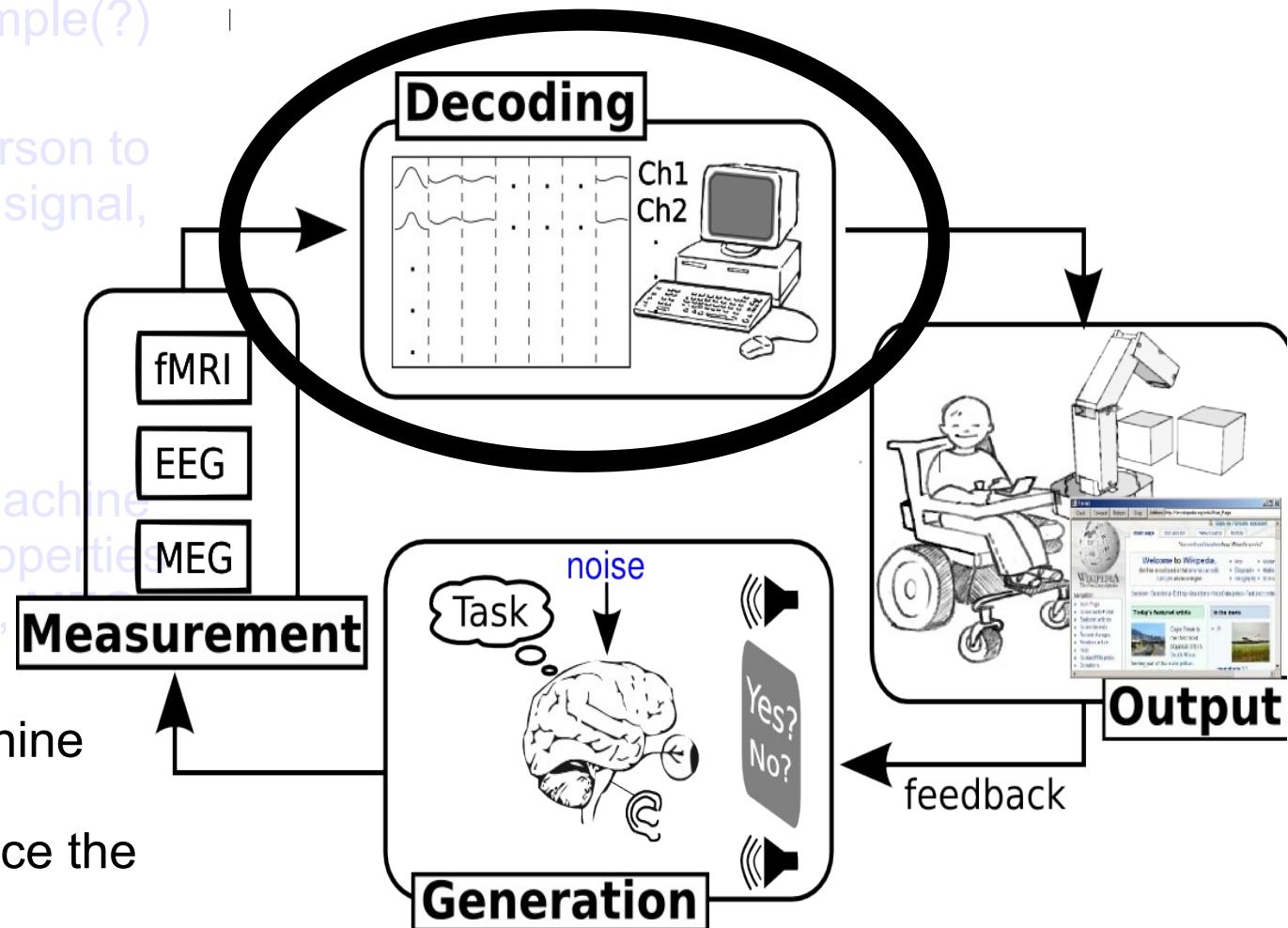
- 1) **Generation:** Get the person to produce a strong brain signal, either by performing an explicit mental-task, or through normal mental processes
- 2) **Measurement:** Build a machine able to measure the properties of their brain, e.g. EEG, MEG, fMRI
- 3) **Decoding:** Build a machine able to decode the measurements to deduce the users mental state
- 4) **Output:** Communicate the mental-state to the outside world.



The BCI Cycle

Fundamentally, BCI is a simple(?) engineering problem;

- 1) Generation: Get the person to produce a strong brain signal, either by performing an explicit mental-task, or through normal mental processes
- 2) Measurement: Build a machine able to measure the properties of their brain, e.g. EEG, fMRI, MEG
- 3) Decoding: Build a machine able to decode the measurements to deduce the users mental state
- 4) Output: Communicate the mental-state to the outside world.

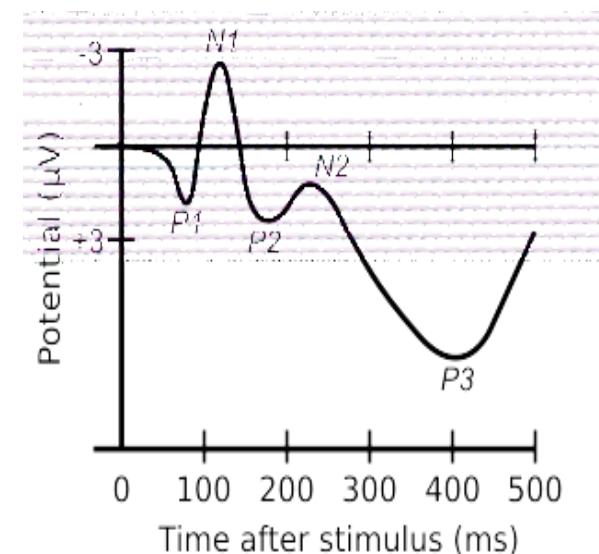


2-Types BCI signal : ERP, ERSP

- Event Related Potentia (ERP)
 - or Evoked response BCIs
- Event Related Spectral Potential (ERSP)
 - Or Induced response BCIs

ERP BCIs

- ERP – time-locked signal generated in response to stimulus
- Location of response depends on modality of stimulus: visual, tactile, auditory
- Can depend on more than raw stimulus properties, e.g.
 - Syntactic processing, semantic processing, semantic updating, error-processing, surprise, etc.
- P300 – 'oddball' response
 - Evoked by: unusual stimulus, absence of stimulus
 - Influenced by: uncertainty resolved by event, importance of the stimulus
 - P300a – stimulus response
 - P300b – cognitive response



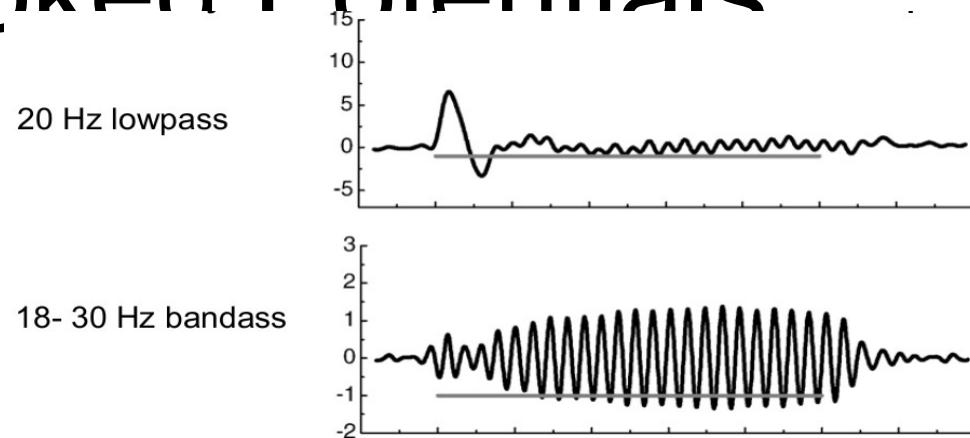
ERP BCIs (2)

- P300 - ERP applications
 - Cortically Coupled Computer Vision (CCCV/RSVP), Brain-fingerprinting
- BCI spelling (Farwell&Donchin) matrix speller
 - Parallel selective attention task
 - Each symbol has unique sequence of 'odd-ball' events
- Overt vs. covert attention
 - Most matrix speller performance based on eye-pointing **but** still works without it..

A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	1	2	3	4
5	6	7	8	9	←

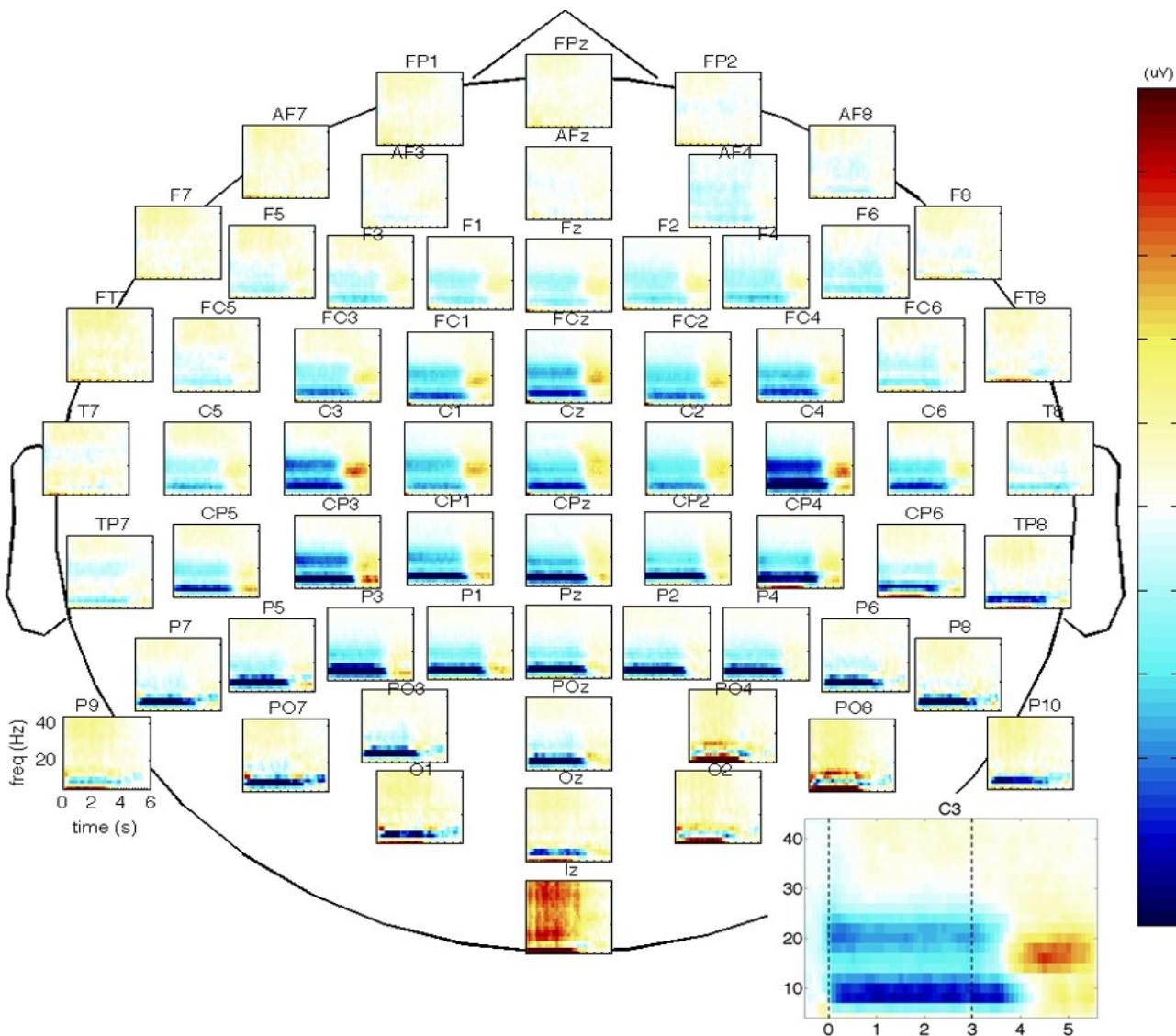
Special Case: Steady State Evoked Potentials

- Steady State Response
 - response to long-term periodic stimulus
- Modality dependent optimal frequency
 - Visual (VSSEP) – 16-20Hz, occipital cortex
 - Auditory (ASSR) – AM modulation ~40Hz, auditory cortex
 - Tactile (SSSEP) – AM ~21Hz, contralateral somatosensory cortex
 - Used to diagnose correct operation of the sensory system
- Response amplitude influenced by selective attention
 - Used for parallel/serial selective attention BCI



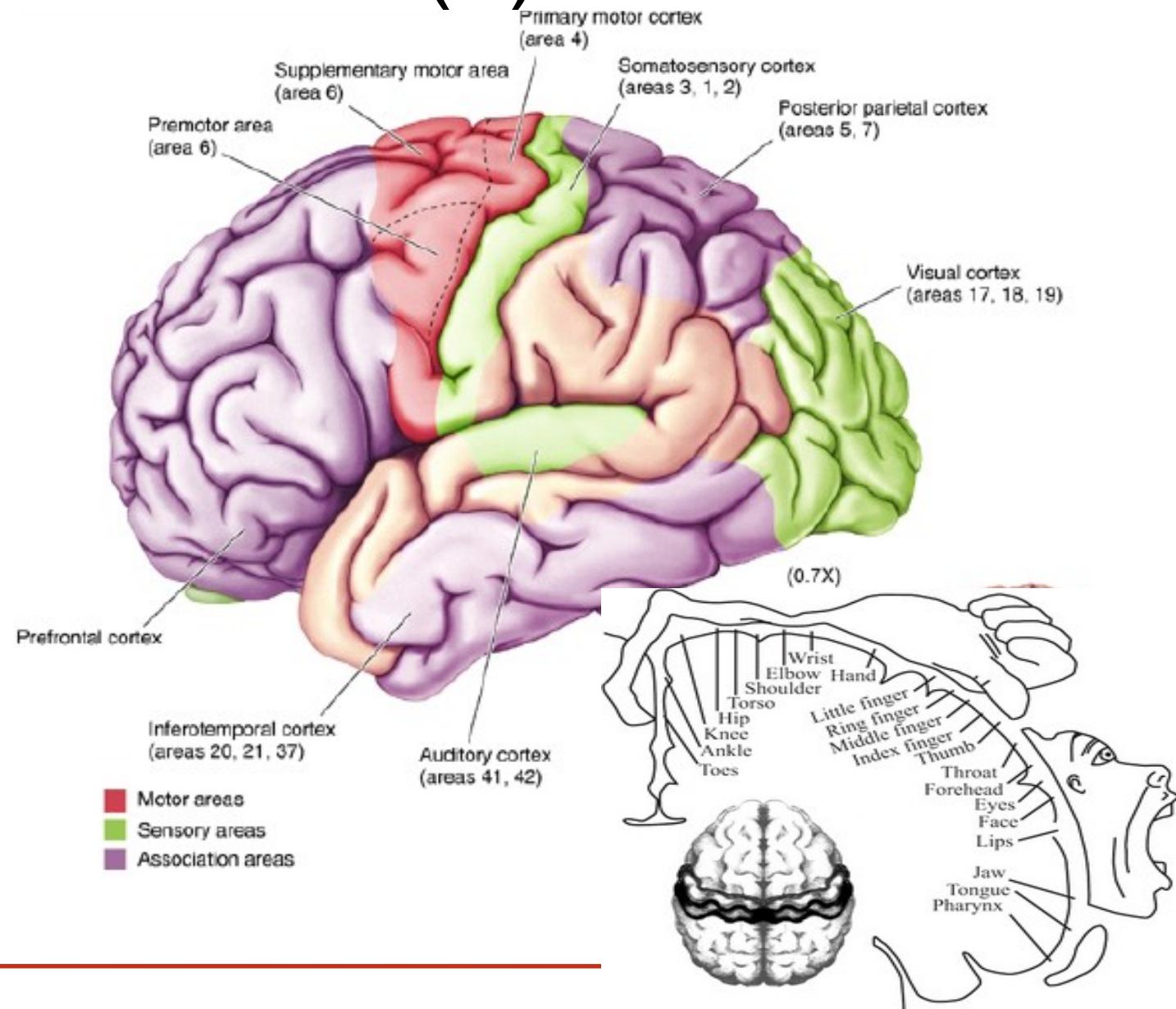
Induced Response / ERSP BCIs

- ERSP – task dependent change in the **power** of the signal in a particular **frequency range**
- Pseudonyms:
 - Event Related De-Synchronisation (ERD/ERS)



ERSPs (2)

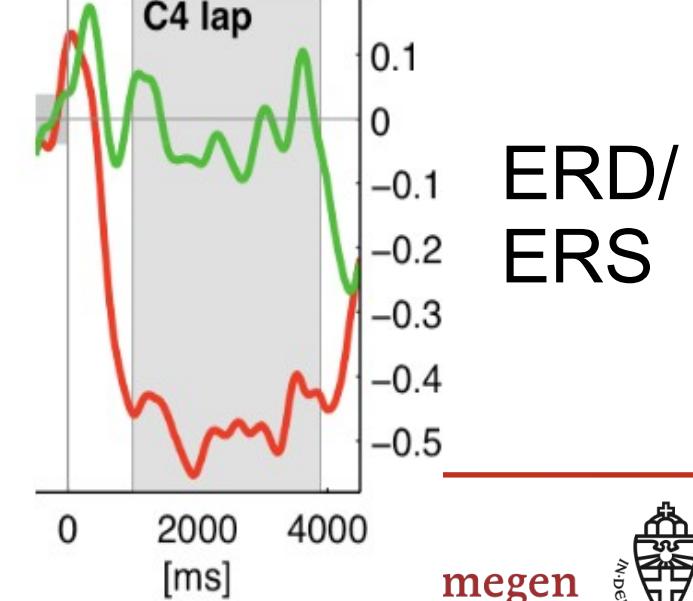
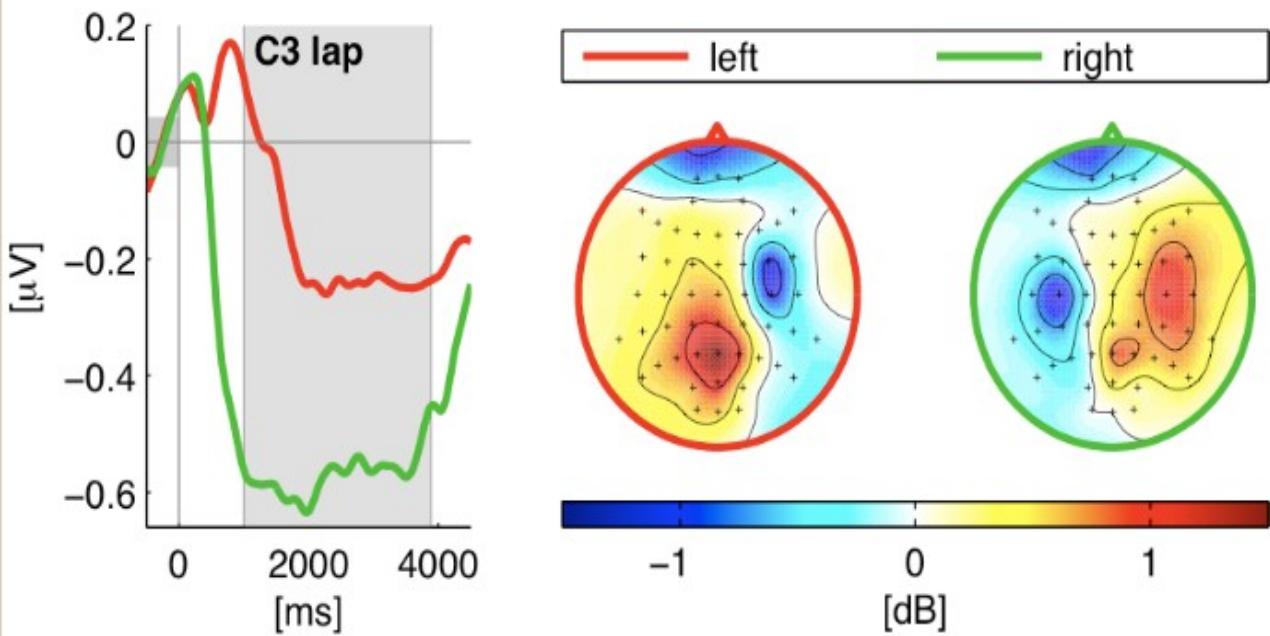
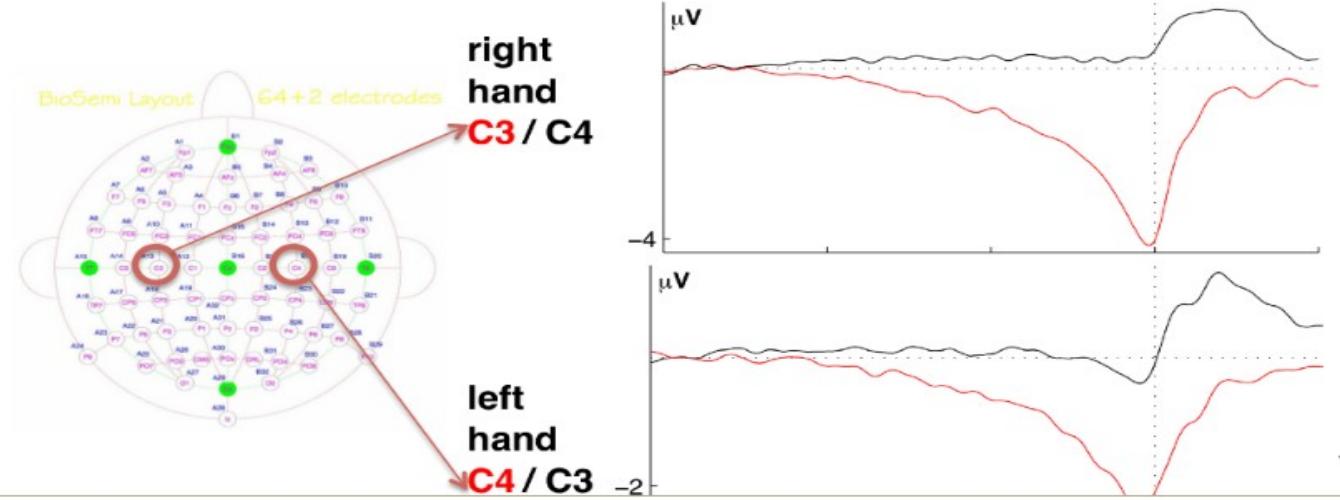
- Location and frequency of the response depends on the mental task the user is performing



ERSP BCIs – Imagined Movements

- Brain signatures of motor imagery

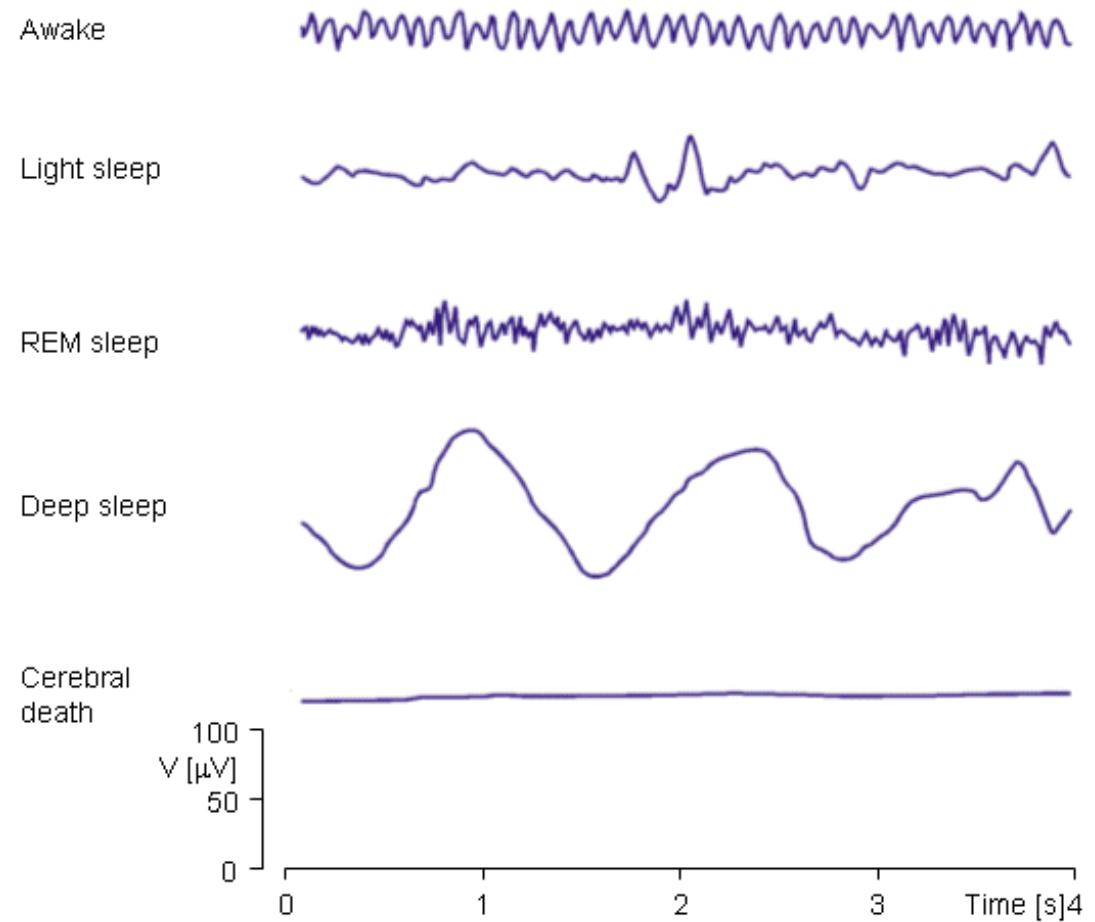
Lateralised
Readiness
Potential
(LRP)



Basic properties of EEG/MEG data and artifacts

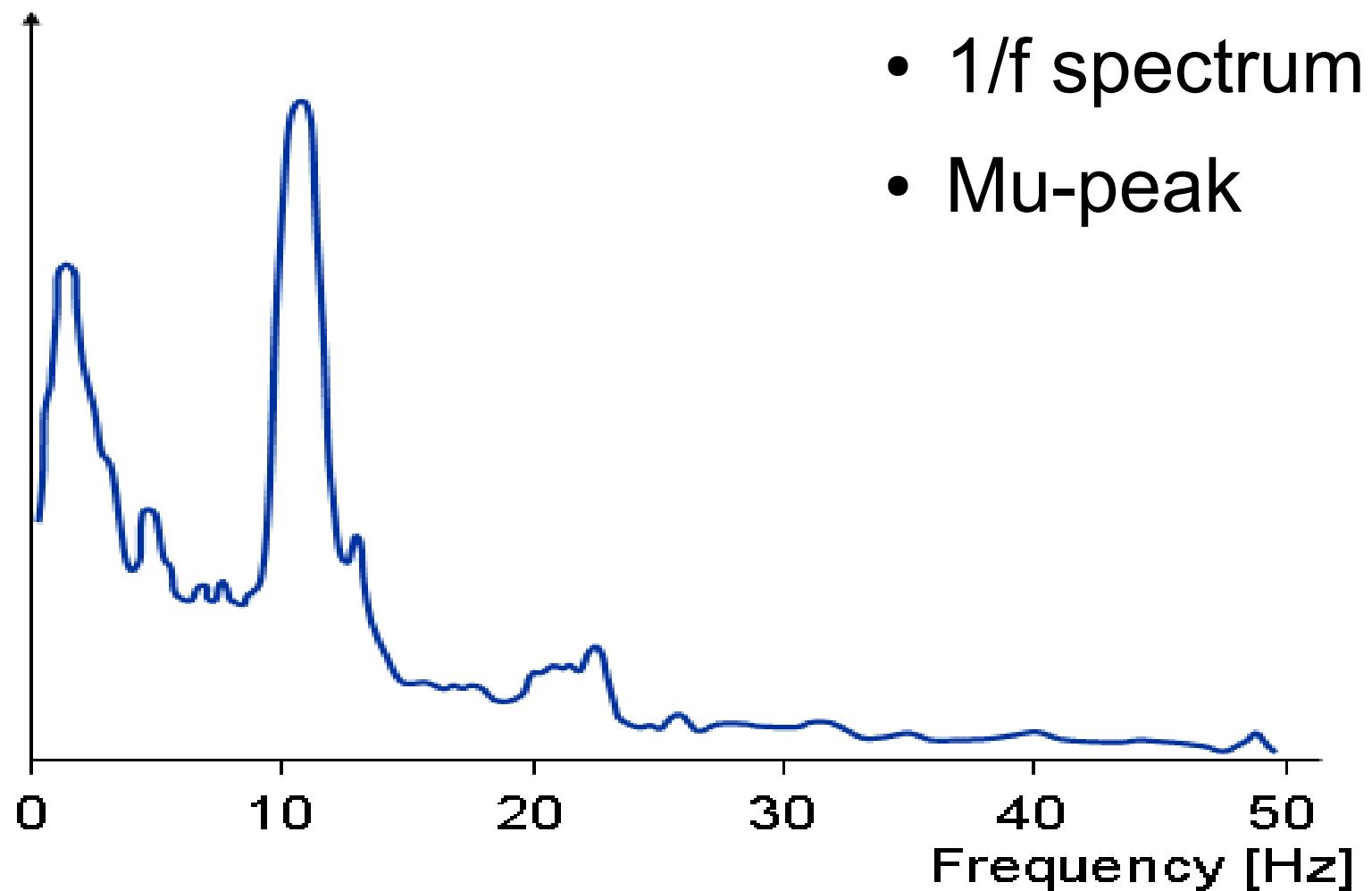
Basic Properties of the EEG

- Temporally
 - Fractal structure



EEG Properties - Spectrally

Relative amplitude



EEG properties - spectrally

Beta (β) 13-30 Hz

Frontally and
parietally



Alpha (α) 8-13 Hz

Occipitally



Theta (θ) 4-8 Hz

Children,
sleeping adults



Delta (δ) 0.5-4 Hz

Infants,
sleeping adults



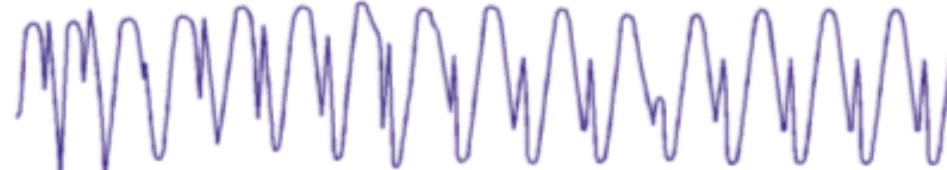
Spikes 3 Hz

Epilepsy -
petit mal

200

100

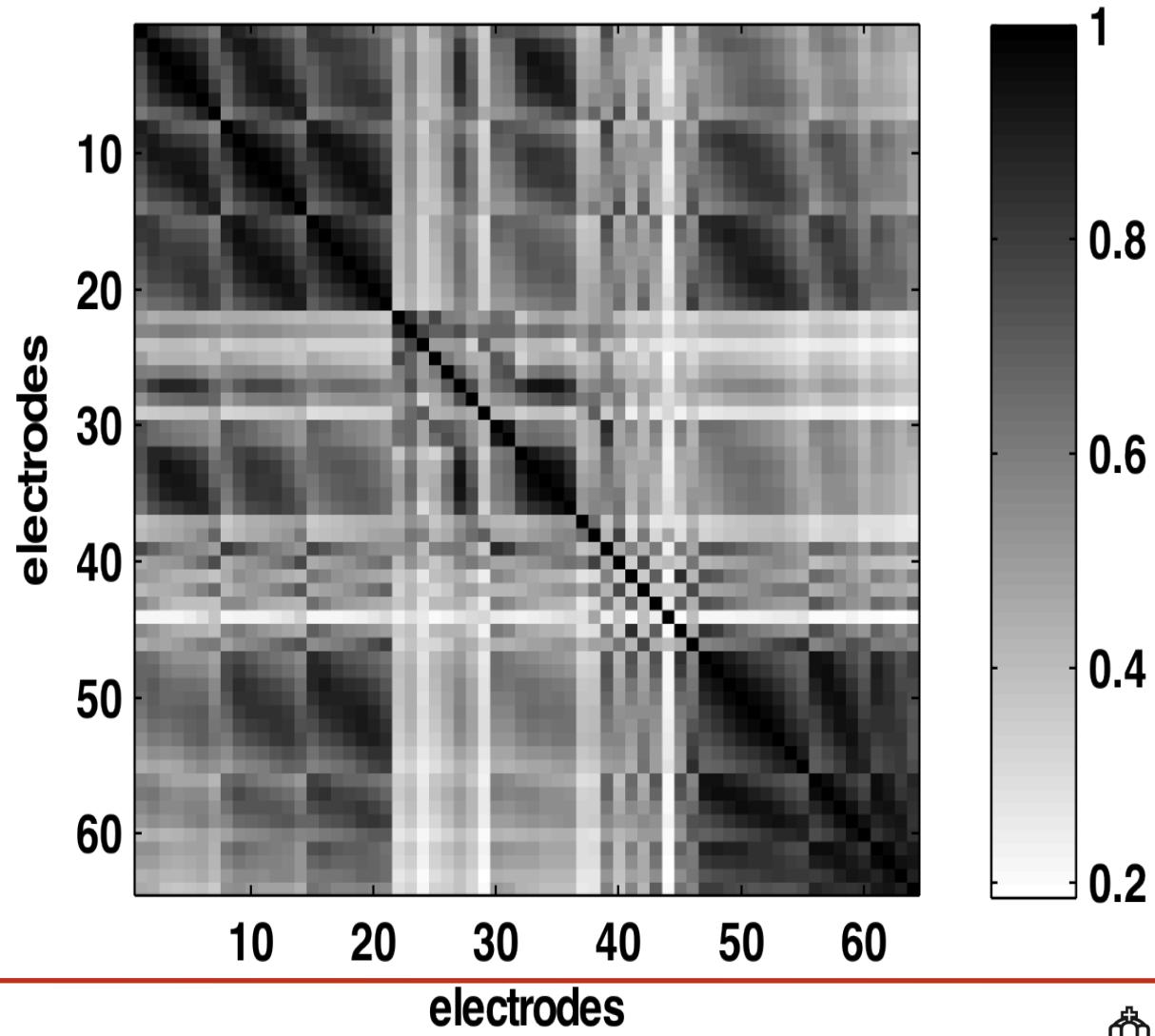
0



0 1 2 3 4 Time [s]

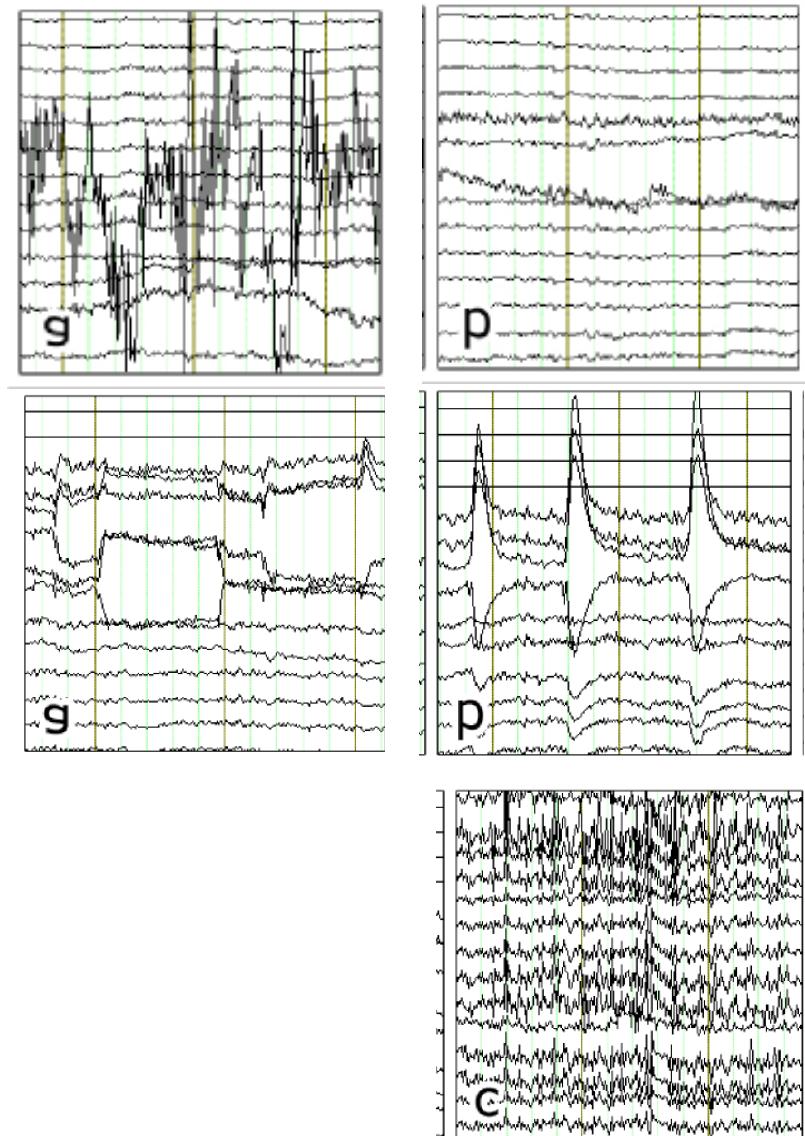
EEG Properties - Spatially

- High spatial correlation between electrodes



External Artifact Sources in EEG

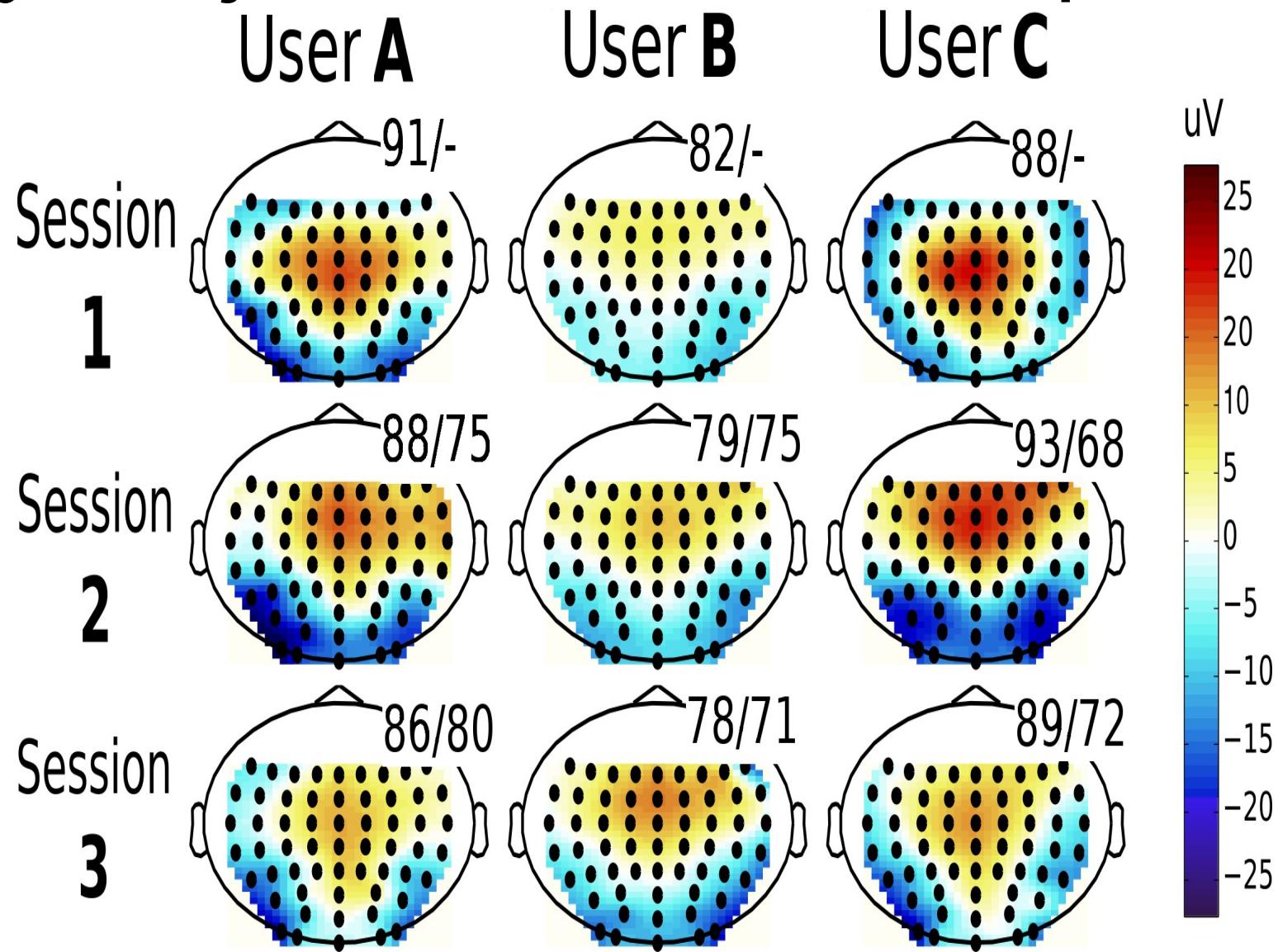
- Bad-channels
- 50Hz
- Slow electrode drifts
- Movement effects – head wiggling
- Eye-movement / Eye blinks
- Muscle movement
 - jaw-clenching
 - frowning
 - ear-wagging



Internal Artifact Sources in EEG

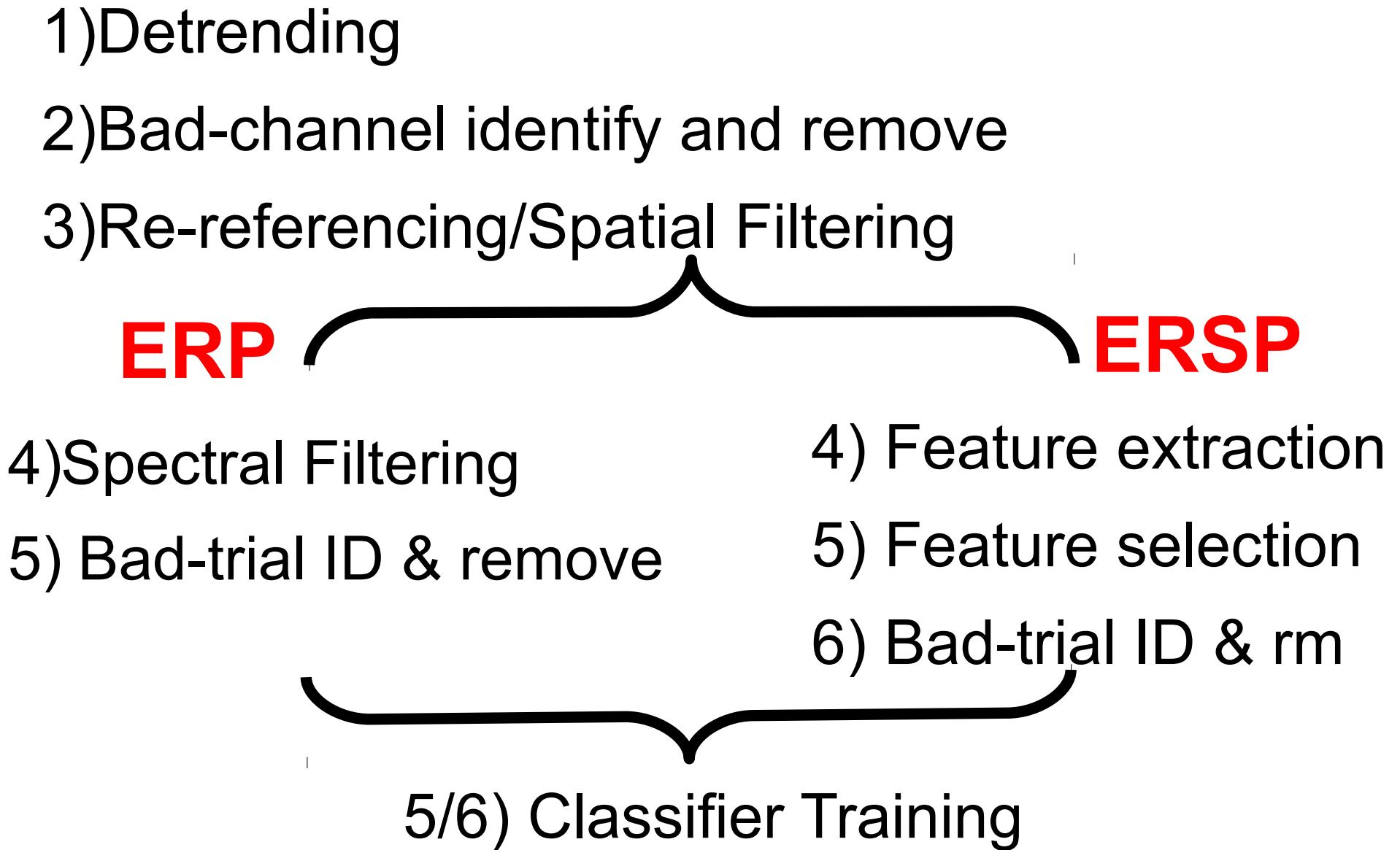
- Eyes open vs. Closed
- Relaxation (deep-breathing) vs. Stressed
- Hyperventilation

Very subject and session dependent

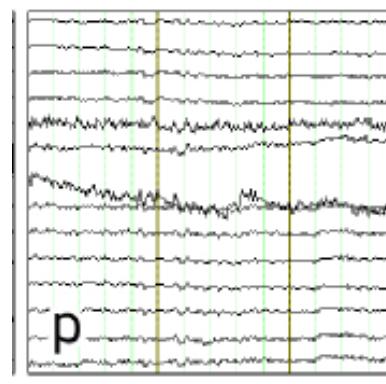


Basic BCI signal processing stages

BCI signal analysis pipeline



1) Detrending



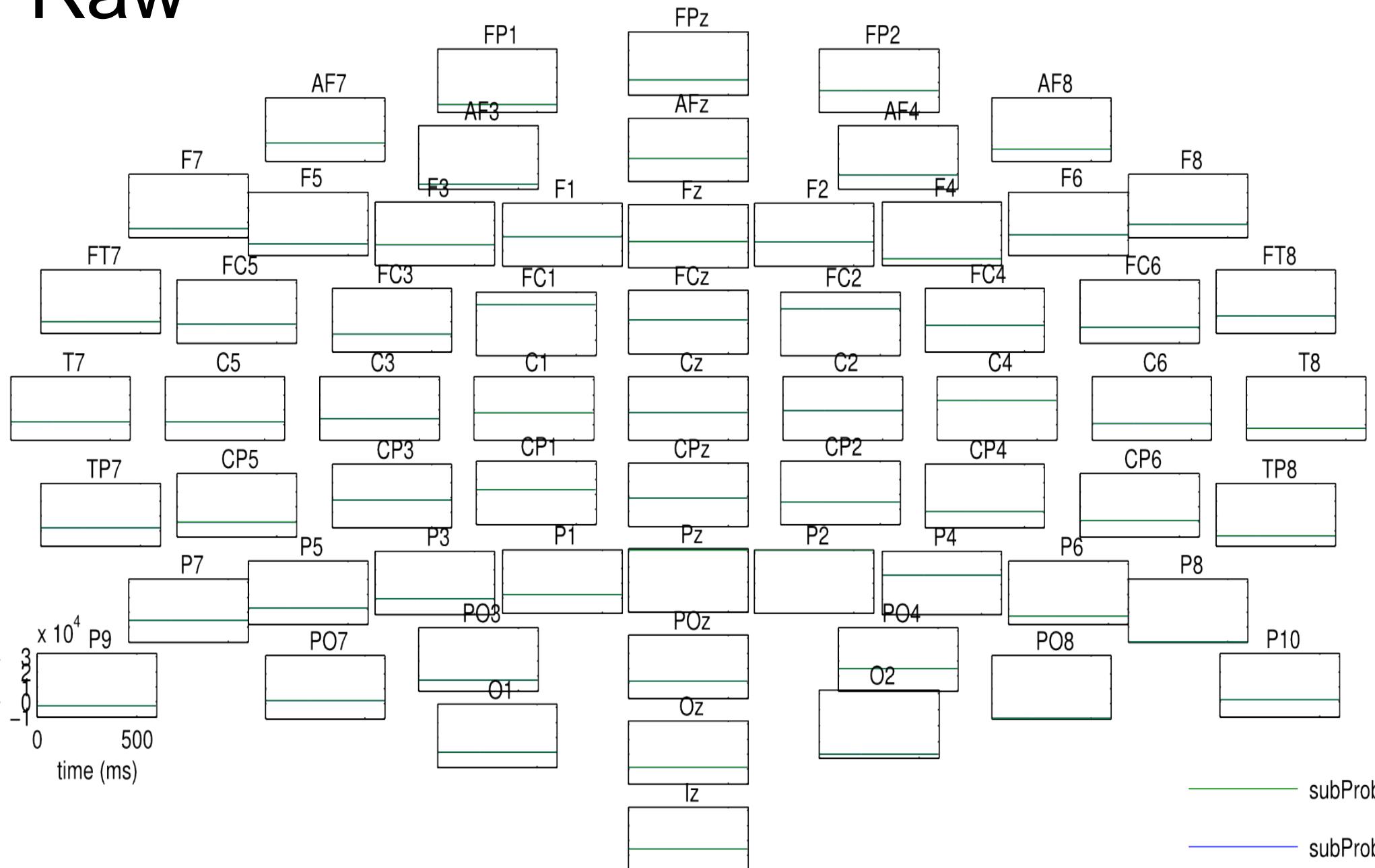
- Why?
 - Remove slow-drifts and arbitrary offsets in EEG data
- How?
 - Compute and subtract linear trends for each channel and epoch

Key Function :

- $X = \text{detrend}(X, \text{dim})$; %Detrend data X along dimension dim

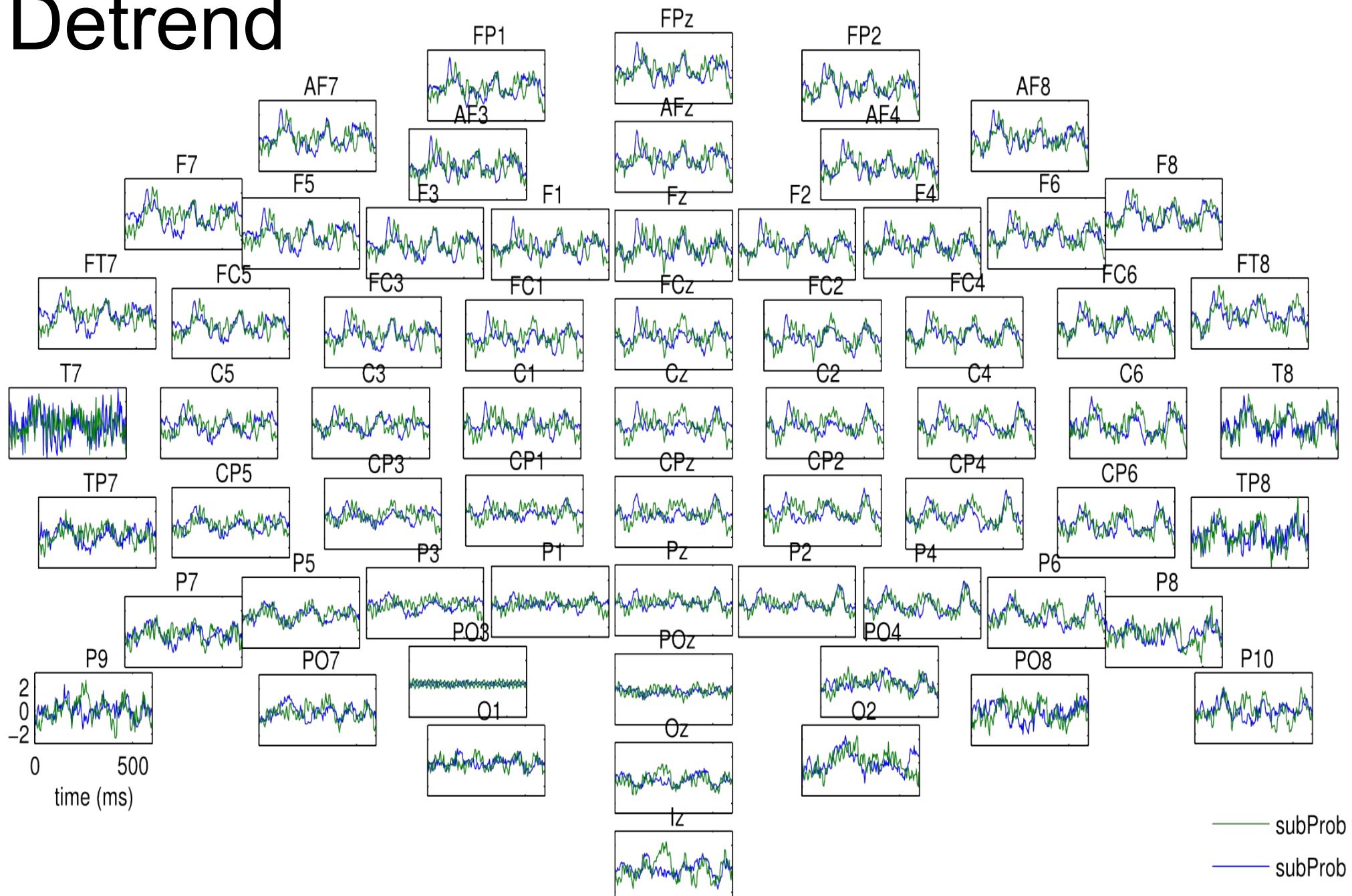
Raw data

Raw

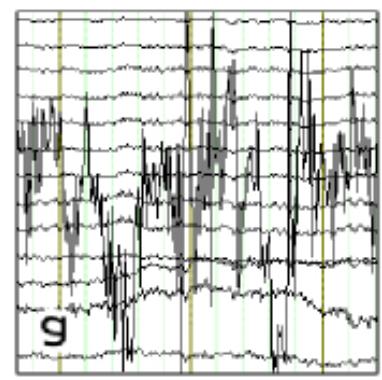


Detrend

Detrend



2) Bad channel identify and remove



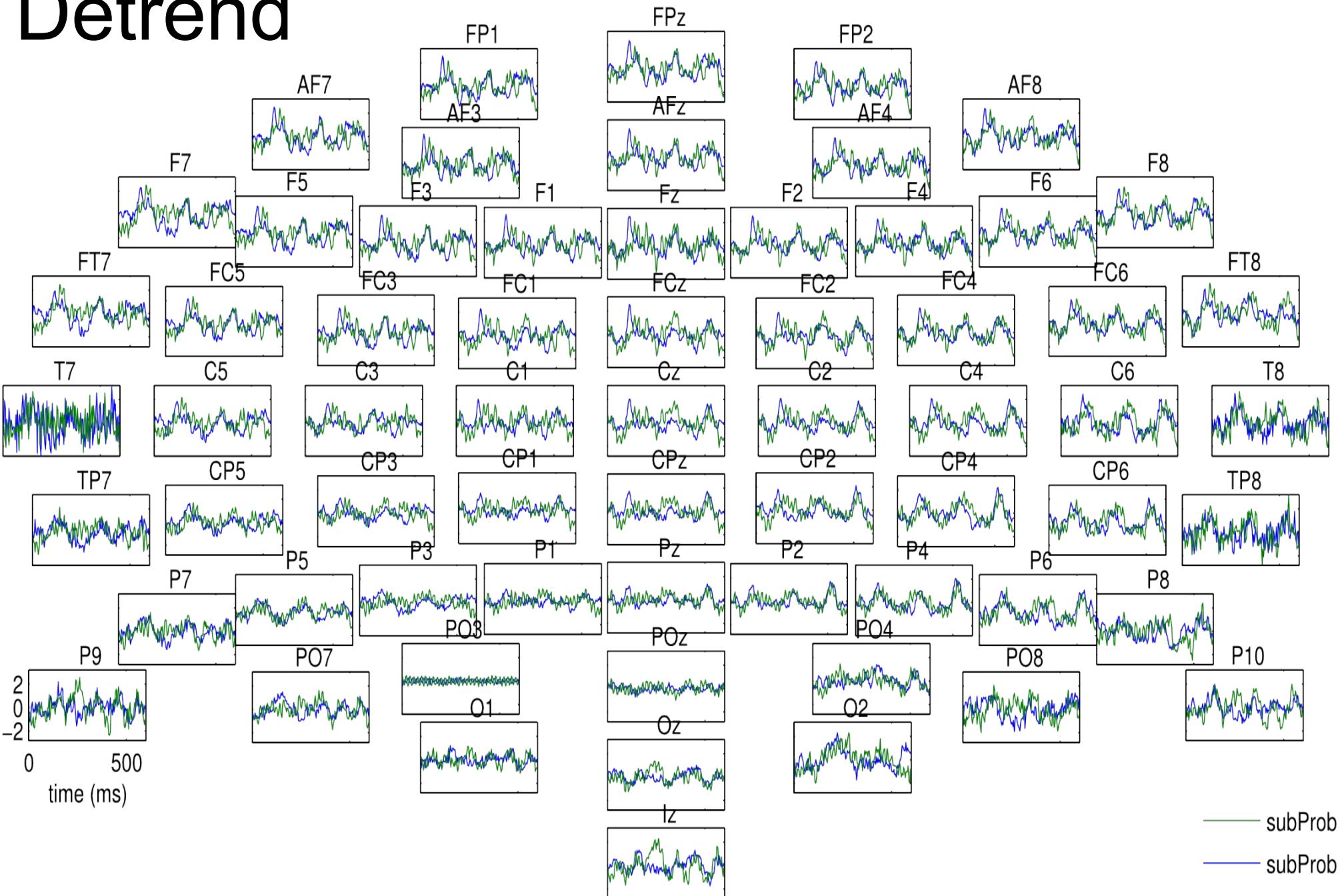
- Why?
 - Some channels don't connect well so only pickup noise. Leaving them in messes up the reset when spatial filtering.
- How?
 - Identify channels with excessively high power.
 - Specifically:
 - compute total power for each channel over all epochs
 - Compute mean channel power and variance in channel power
 - Remove any channels with power more than 3 std-dev from mean

Key Function :

- `badidx=idOutliers(X, dim, thresh);` % identify outlining elements of X along dim

Detrend

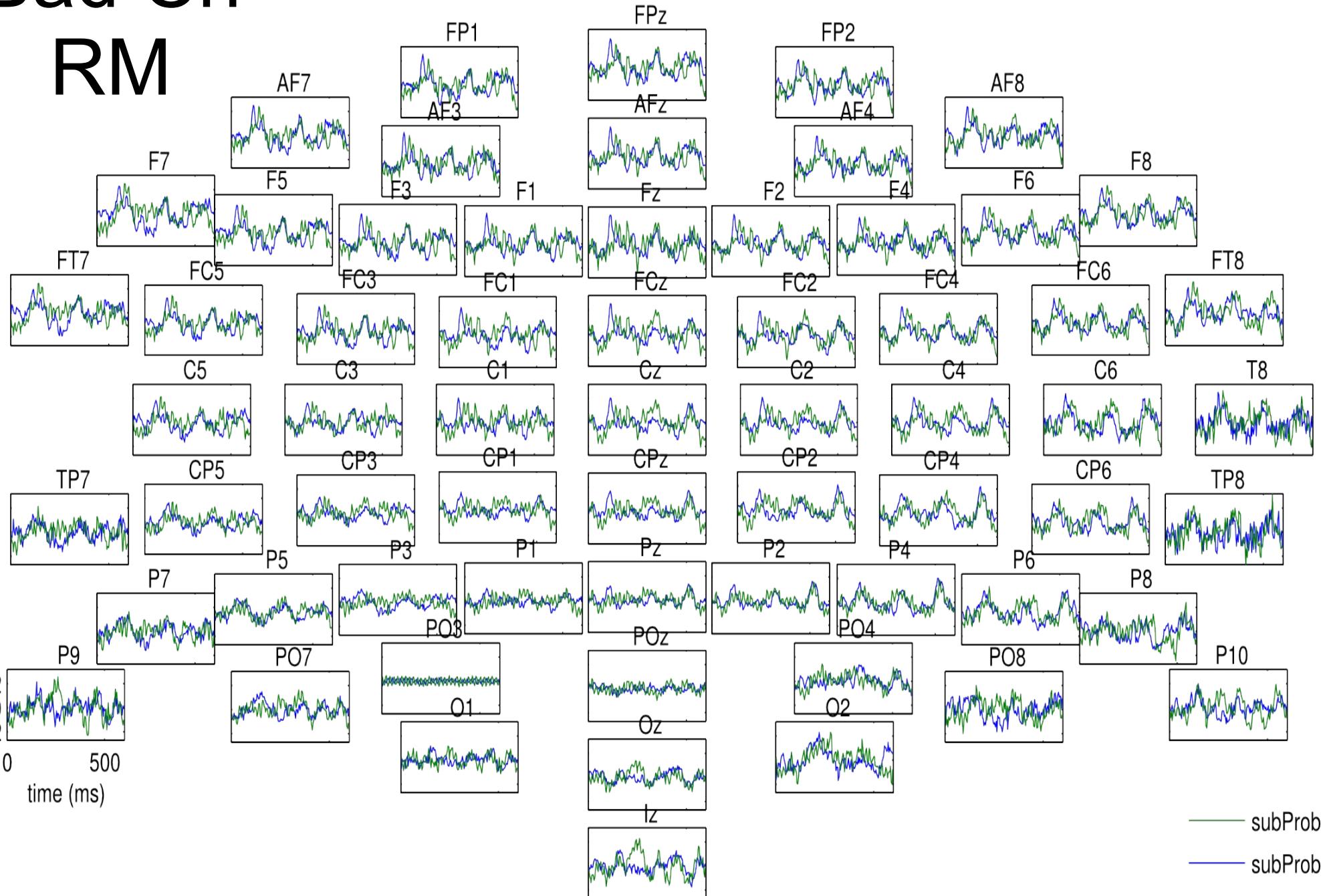
Detrend



Bad Ch

RM

Detrend + bad-ch rm



3) Spatial Filter

- Why?
 - EEG contains lots of signal from external noise sources – which is visible as a signal common to all channels
 - Volume conduction means nearby channels have high correlation
- How?
 - Common Average Reference (CAR) -- Remove common external signal by subtracting the average signal over channels from all channels
 - Surface Laplacian – Remove channel correlation (and common signal!) subtracting a local average signal (**default**)
 - Spatial Whiten – Remove channel correlation (and common signal!) using PCA to map to a co-ordinate set where channels are uncorrelated.

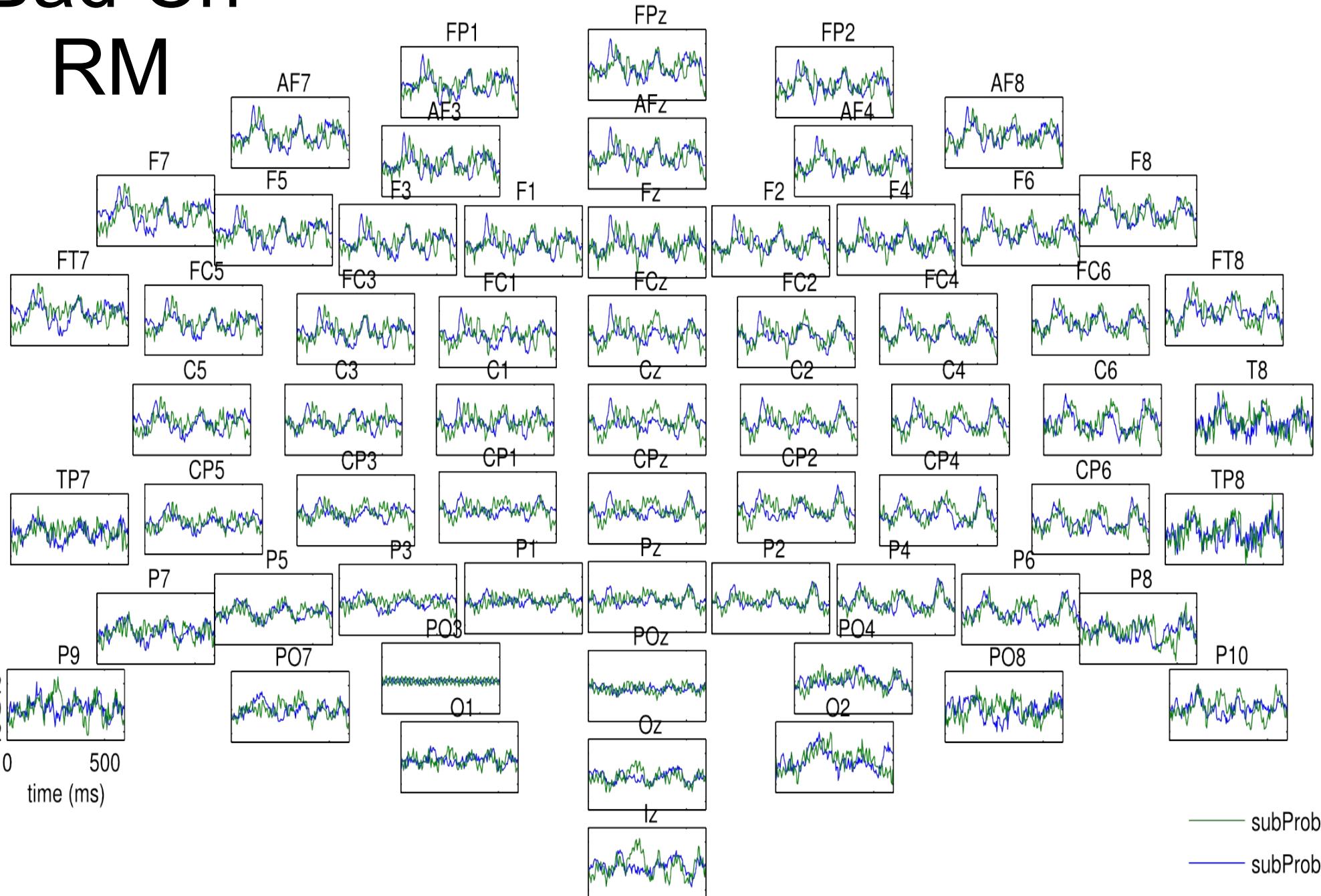
Key functions

- $X = \text{repop}(X, '-', \text{mean}(X, 1));$
 - subtract average over dimension 1
 - Alternative:
 - $X = X - \text{repmat}(\text{mean}(X, 1), [\text{size}(X, 1), 1, 1]);$
 - $R = \text{sphericalSplineInterpolate}(chpos, chpos, [], [], 'slap');$
 - compute Surface Laplacian (SLAP) spatial filter for electrodes at **3-d** positions $chpos$
 - R is a $[nCh \times nCh]$ spatial filter matrix
 - $X = \text{tprod}(X, [-1 2 3], R, [-1 1]);$
 - Apply the spatial filter matrix R to the data X
 - Alternative:
 - $X = \text{reshape}(R' * \text{reshape}(X, \text{size}(X, 1), []), [\text{size}(R, 2), \text{size}(X, 1), \text{size}(X, 2)])$
-
- Diagram illustrating the flow of operations:
- Apply filter**: Points to the code $X = \text{tprod}(X, [-1 2 3], R, [-1 1]);$
 - Make 2D**: Points to the code $X = \text{reshape}(R' * \text{reshape}(X, \text{size}(X, 1), []), [\text{size}(R, 2), \text{size}(X, 1), \text{size}(X, 2)])$
 - Make 3d again**: Points to the code $X = \text{reshape}(R' * \text{reshape}(X, \text{size}(X, 1), []), [\text{size}(R, 2), \text{size}(X, 1), \text{size}(X, 2)])$

Bad Ch

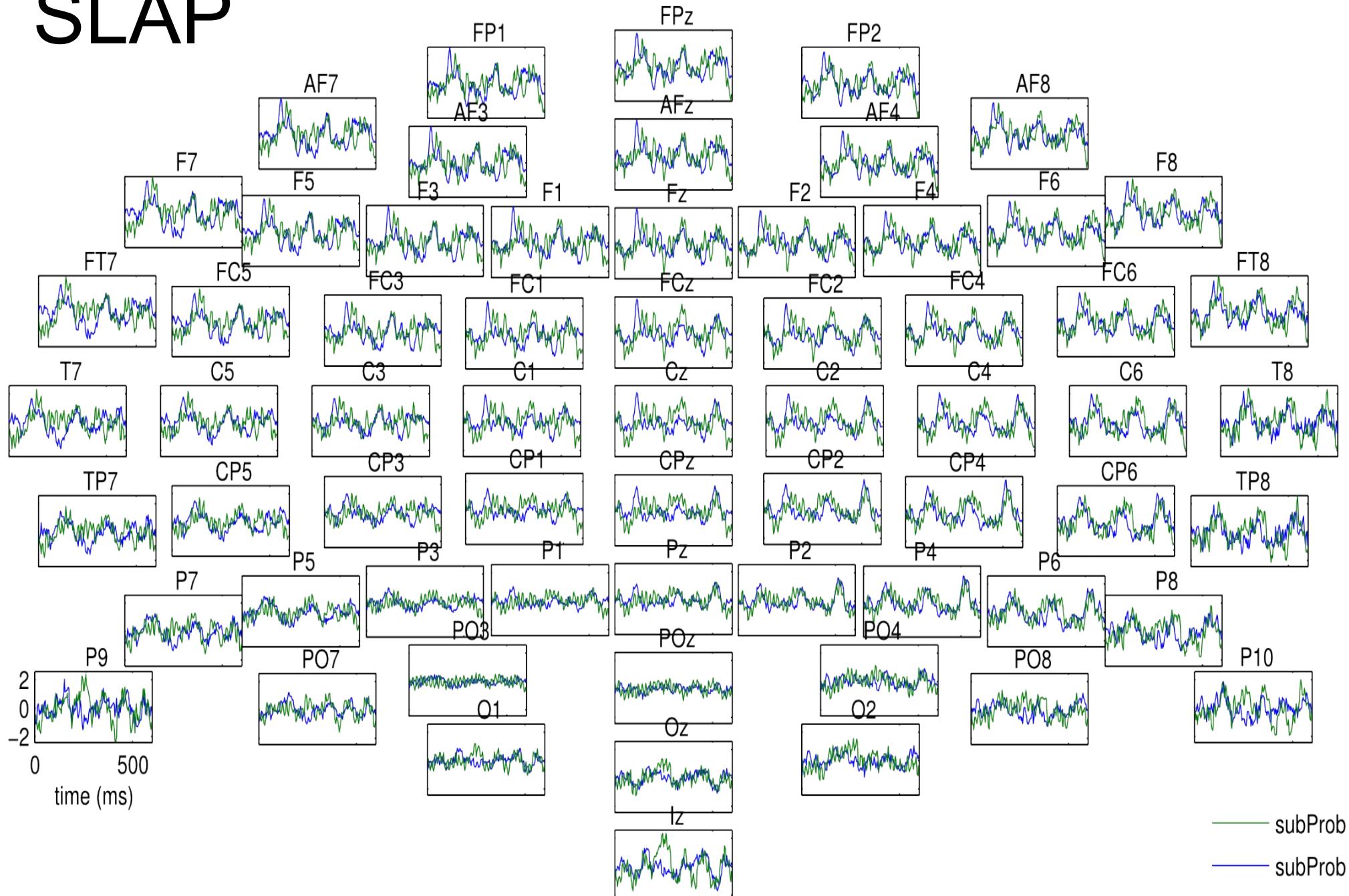
RM

Detrend + bad-ch rm

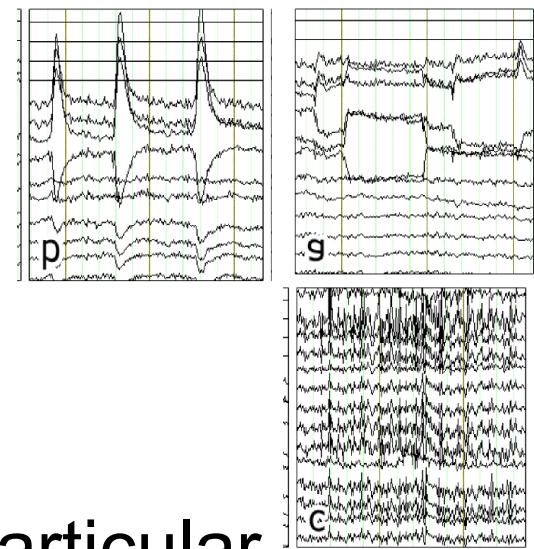


Detrended + bad-ch rm + SLAP

SLAP



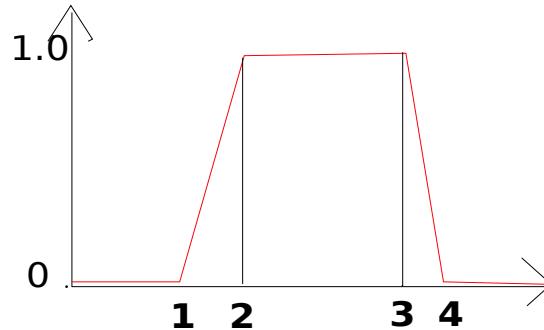
4) ERP Spectrally Filter



- Why?
 - Our signal of interest only occurs in a particular frequency range
 - Thus, any signal which occurs outside this frequency range **must** be noise, and should be removed
- How?
 - Apply a spectral filter to remove frequencies outside the range of interest

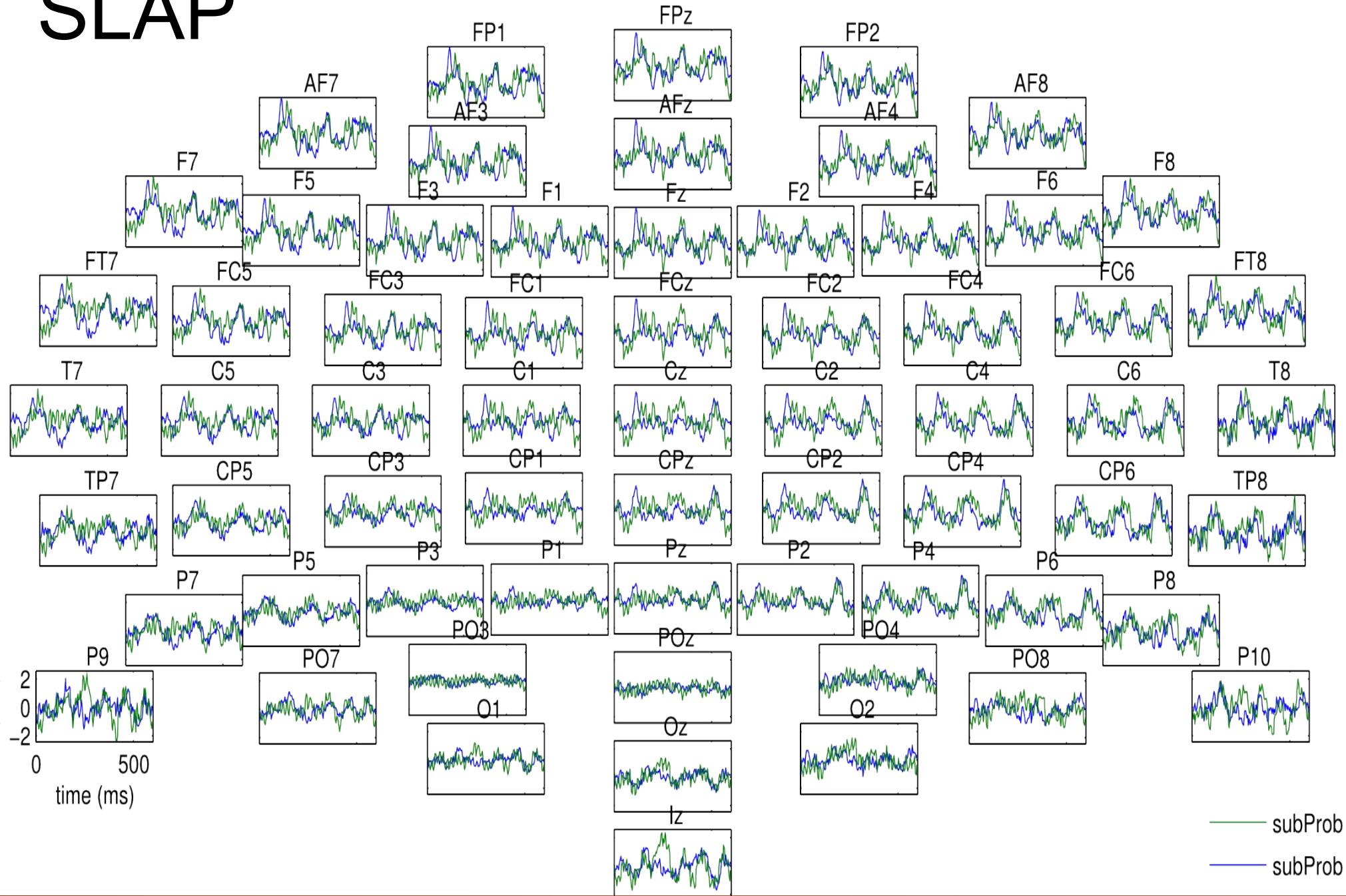
Key functions

- `filt=mkFilter(size(X,2)/2,bands,fs/size(X,2));`
 - Make a spectral domain filter with pass-band given by *bands*
 - *bands* has 4 elements
- To reduce filtering artifacts ramps should be >1hz wide
- `X=fftfilter(X,filt,[],2)`
 - Apply the spectral domain filter *filt* to X along dim 2



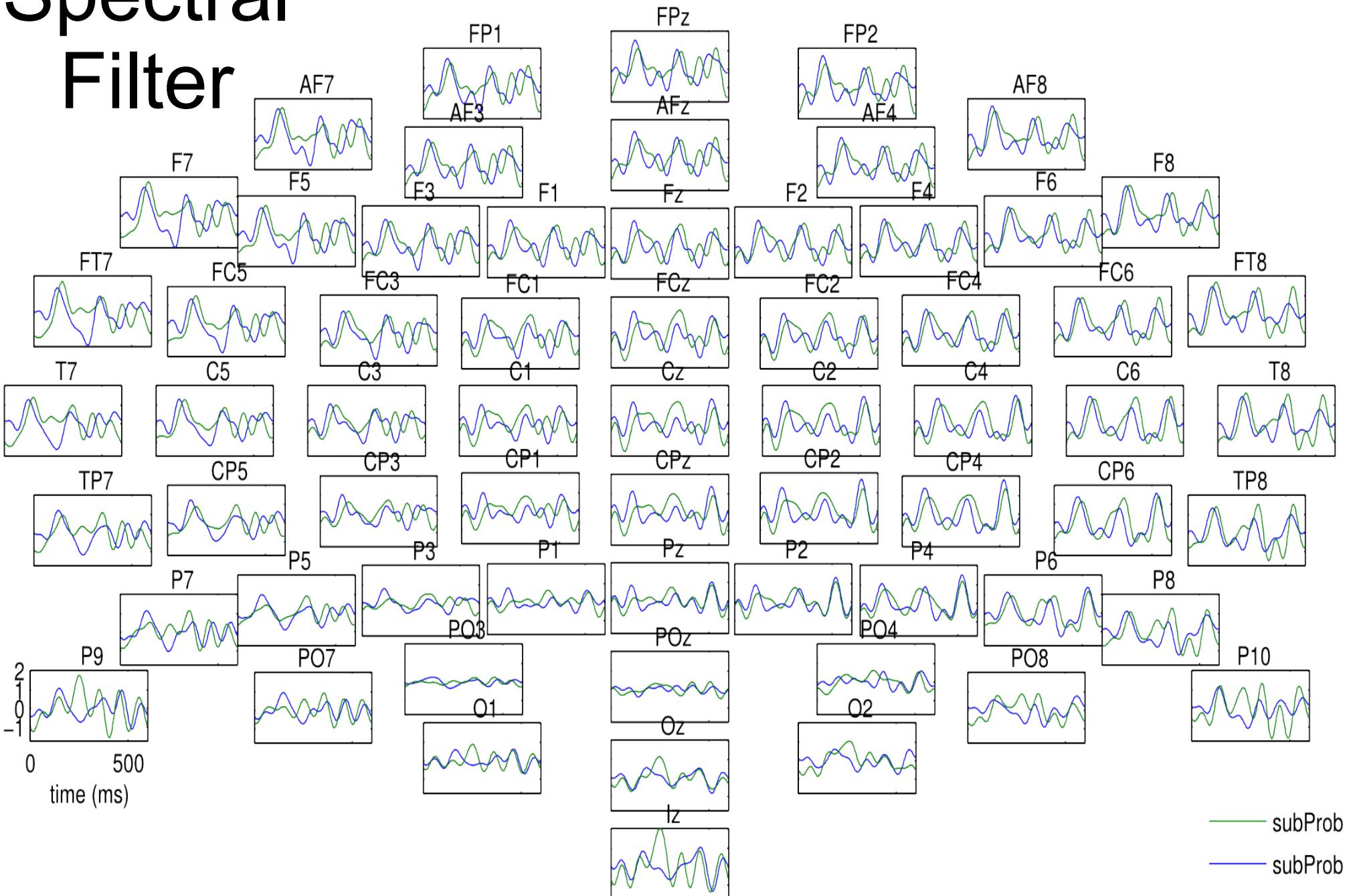
Detrended + bad-ch rm + SLAP

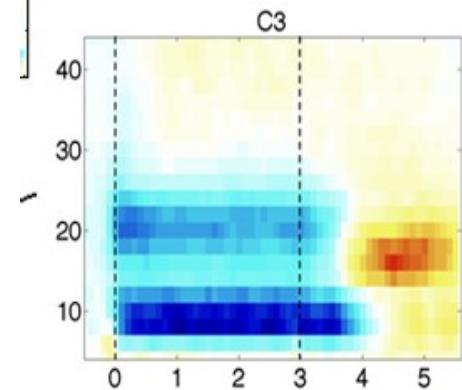
SLAP



Spectral Filter

Detrended + bad-ch rm + SLAP + spectral filter





4) ERSP Feature Extraction

- Why?
 - Our signal of interest is **not** time locked, but a change in power in a particular frequency range
 - Thus, it cannot be detected from the raw time-domain features by a linear classifier
- How?
 - Use Welch's method to compute a high quality estimate of the signals power spectrum for each epoch, i.e. power in each frequency bin
 - N.B. To make the distribution of powers more Gaussian distributed, we use **log power** as the classification feature

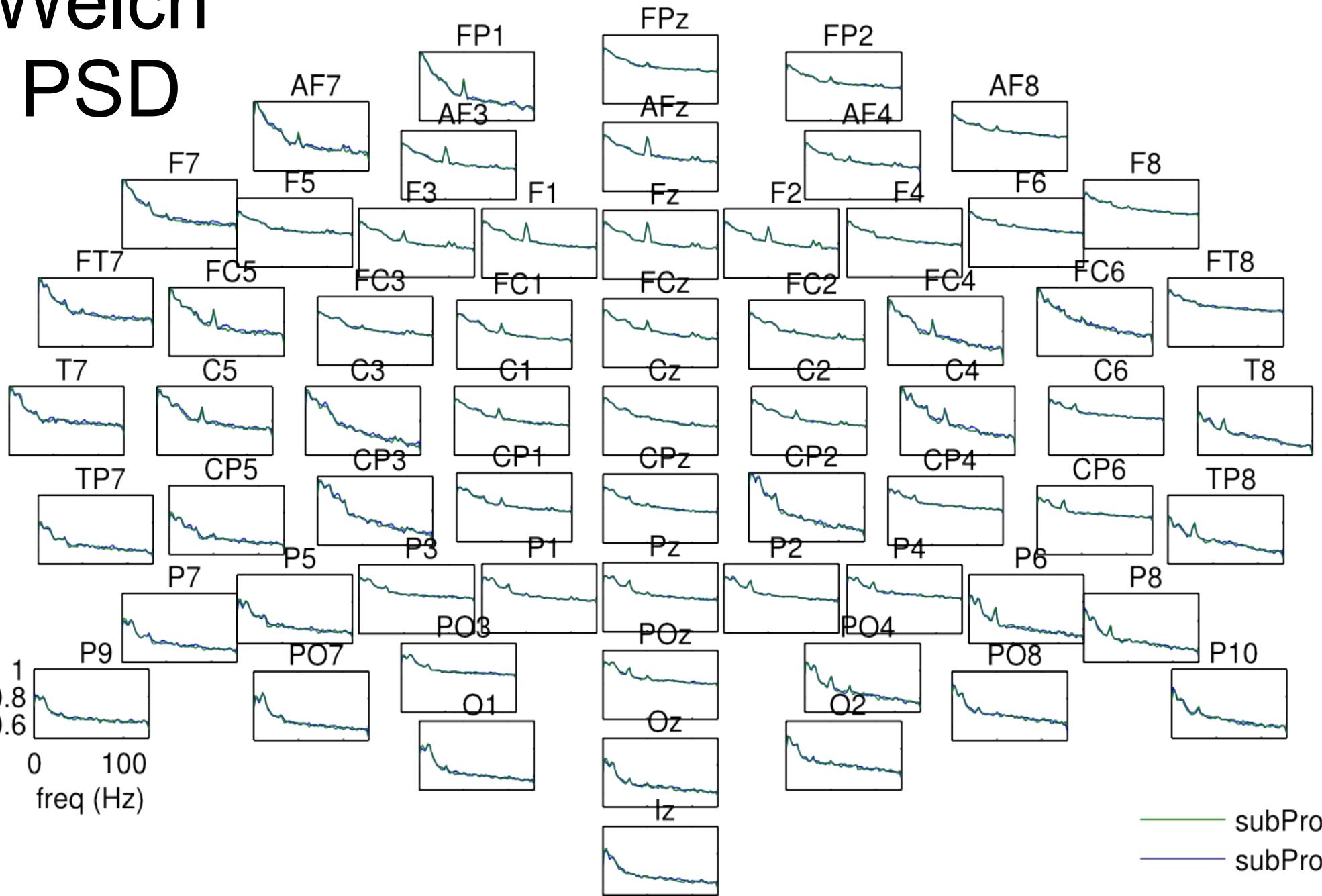
Key functions

- `X=welchpsd(X,2,'width_ms',500,'fs',fs);`
 - Compute the power-spectral-density (PSD) of X along dimension 2 using Welch's method[1] with *width_ms* sized windows.
 - N.B. Output spectral resolution = $1000/\text{width_ms}$ Hz

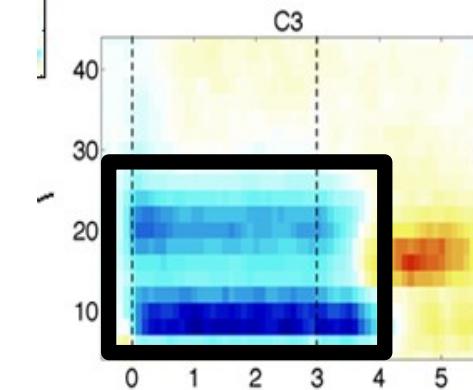
[1] see [en.wikipedia.org/wiki/Welch's_method](https://en.wikipedia.org/wiki/Welch%27s_method)

Welch PSD

Detrended + bad-ch rm + SLAP + welch



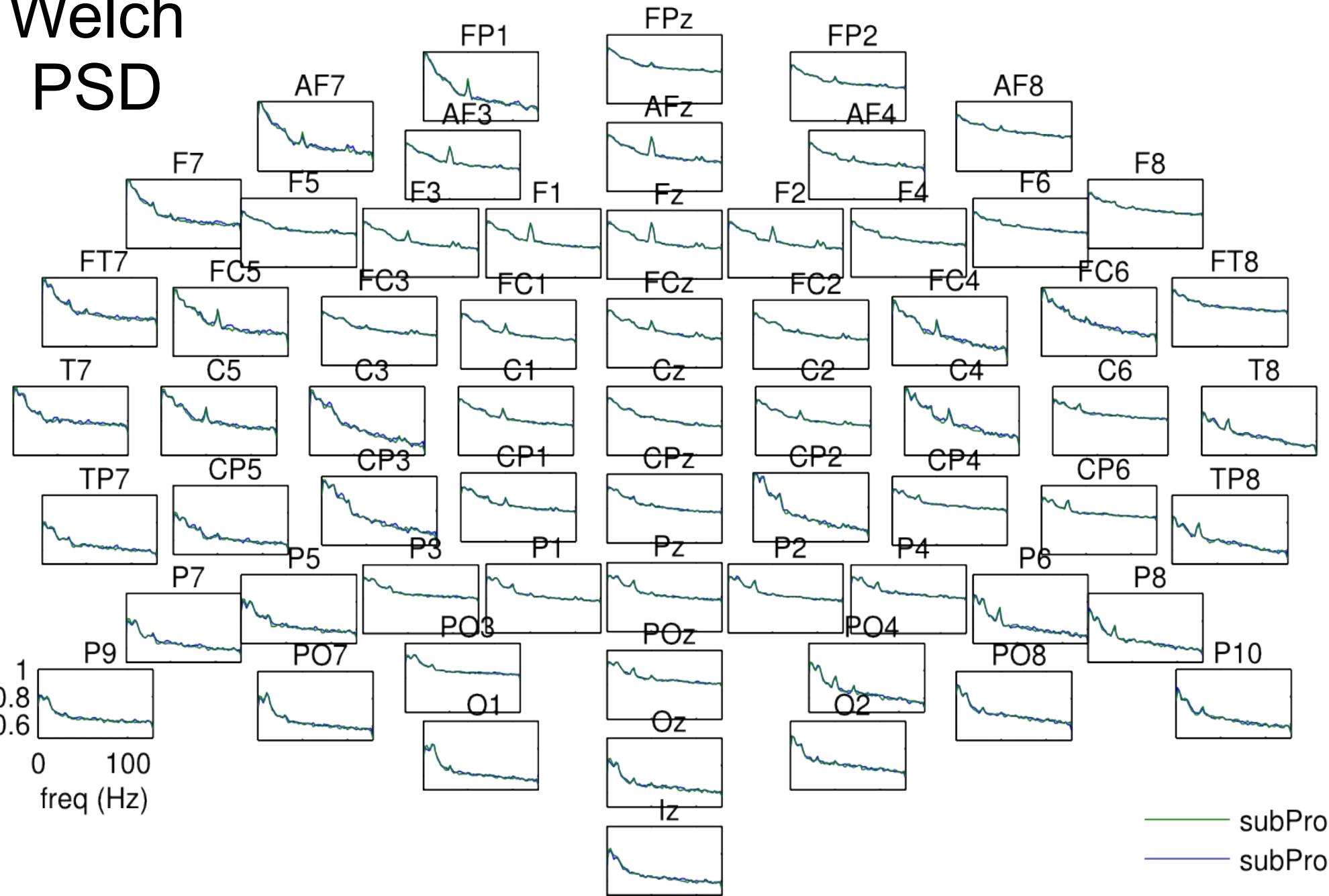
5) ERSP Feature Selection



- Why?
 - Our signal of interest only occurs in a particular frequency range
 - Thus, any signal which occurs outside this frequency range **must** be noise, and should be removed
- How?
 - Discard frequency bins from the power spectrum which occur outside the range of interest

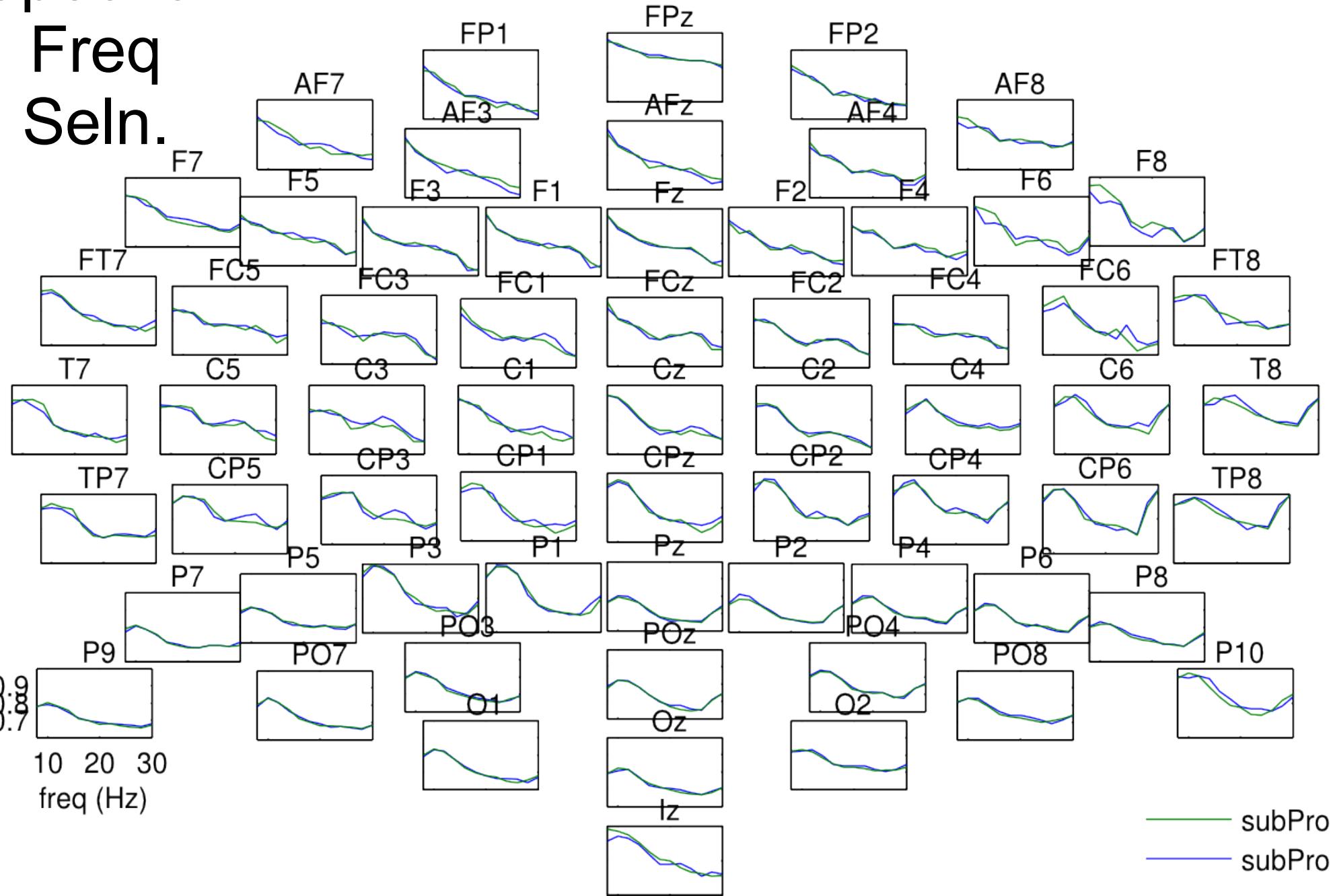
Detrended + bad-ch rm + SLAP + welch

Welch PSD

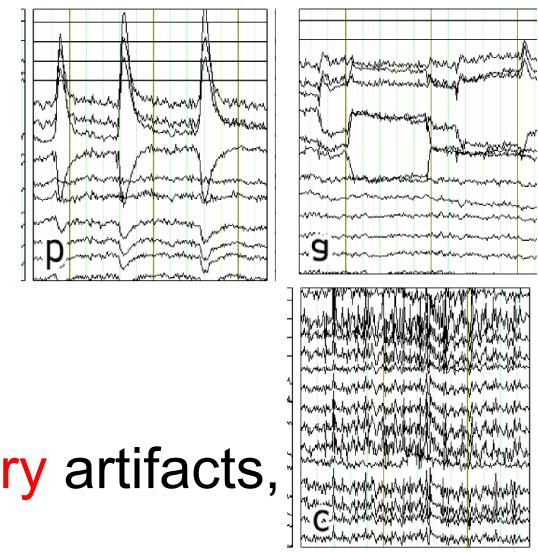


Spectral

Freq Seln.



5/6) Bad trial identify and remove



- Why?

- Some trials (blocks of time) are just corrupted by **tempory** artifacts, e.g. eye-blanks.
- Leaving these in will cause the classifier to produce garbage which leads to output mistakes.
- Generally Better to identify produce a null-output than make a mistake... (particularly in sequence based BCIs like visual spellers)

- How?

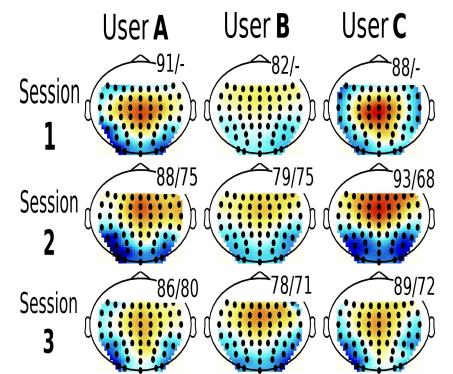
- Identify time points with excessively high power.
- Specifically:
 - compute total power for each trial (time-block)
 - Compute mean trial power and variance in power
 - Remove any channels with power more than 3 std-dev **greater** than

Key Function :

- `badiidx=idOutliers(X,dim,thresh);` % identify outlining elements of X along dim

6/7) Train Classifier

- Why?
 - Account for high variability in signal and noise properties, by building a special purpose detector for this subject on this day.
- How?
 - Train a linear logistic regression classifier with quadratic regularisation on the gathered examples
 - With 10-fold cross validation to select the optimal regularisation strength
 - N.B. Many other classifiers possible... doesn't seem to matter much so long as its regularised



Key functions

- `X=cvtrainLinearClassifier(X,Y,Cs,nFold);`
 - Train a linear classifier on data X for labels Y
 - default *quadratically (L_2) regularised [1] logistic regression[2]*
 - estimate regularisation parameter using *nFold* cross-validation [3]
 - Cs is the set of regularisation parameters to test (automatically set if empty)
- Note: specify alternative classifiers using:
 - `cvtrainLinearClassifier(X,Y,[],10,'objFn','objFn')`
 - Where *objFn* is the name of a classifier, e.g. '`lr_cg`', '`l2svm_cg`', '`kls_cg`', '`rklis_cg`', etc.
 - N.B. Use option '`compKernel`', 1 for **kernel** methods [4], SVM, KLR

[1] see en.wikipedia.org/wiki/Regularization_%28mathematics%29

[2] see en.wikipedia.org/wiki/Logistic_regression

[3] see [en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))

[4] see en.wikipedia.org/wiki/Kernel_methods



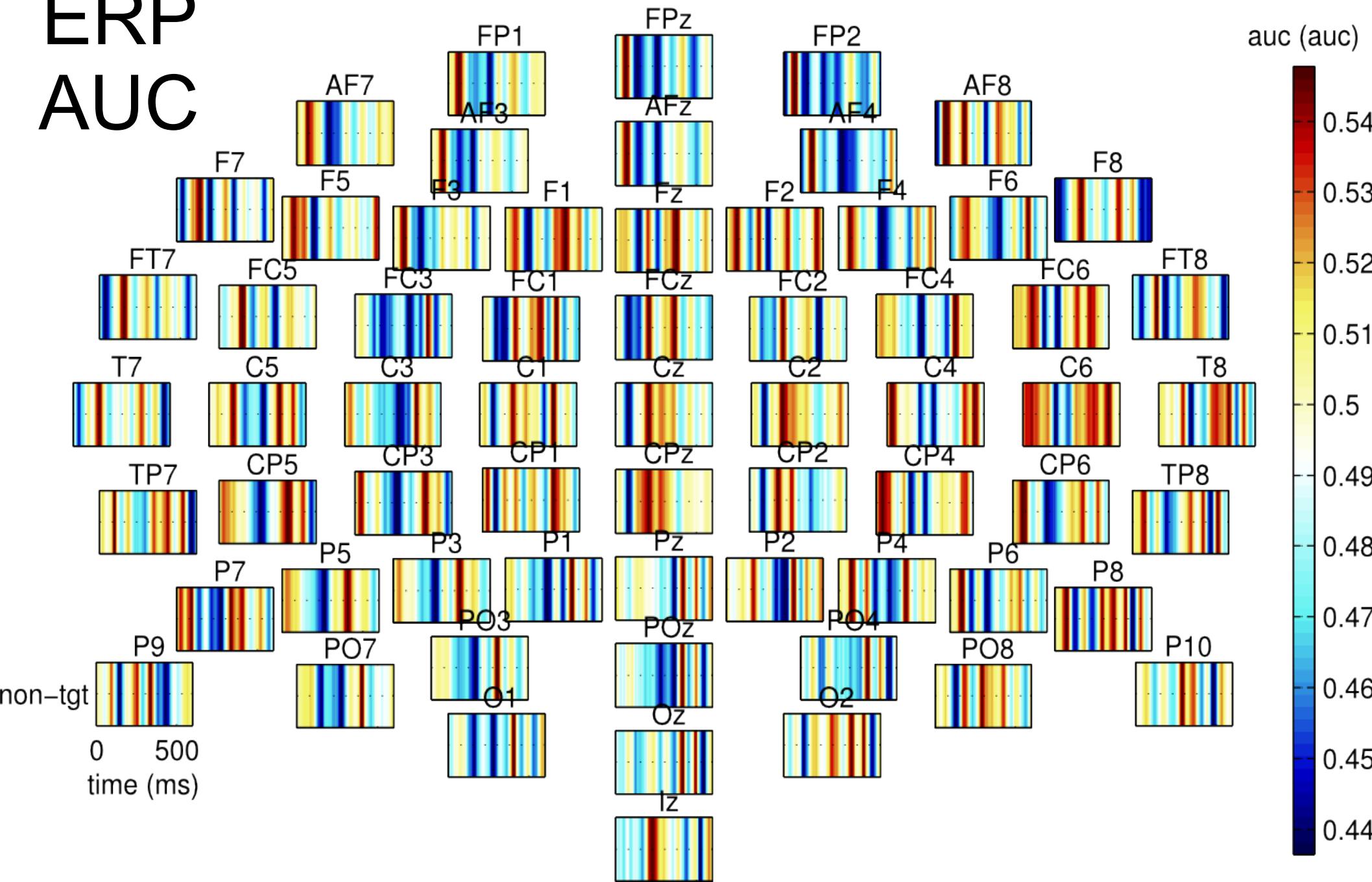
Classifier Output

Regularisation Strength

	High					low
outer fold, (all data)	(out)	0.51/NA	0.93/NA	0.95/NA	0.95/NA	0.95/NA
	(1)	0.50/0.50	0.93/0.91	0.95/0.95	0.95/0.95	0.95/0.95
	(2)	0.50/0.50	0.93/0.92	0.95/0.94	0.95/0.94	0.96/0.94
	(3)	0.50/0.50	0.92/0.96	0.95/0.99	0.95/0.99	0.95/1.00
	(4)	0.50/0.50	0.93/0.93	0.95/0.97	0.95/0.97	0.95/0.98
	(5)	0.50/0.50	0.92/0.93	0.96/0.94	0.95/0.93	0.96/0.93
	(6)	0.50/0.50	0.93/0.91	0.95/0.96	0.95/0.97	0.95/0.97
	(7)	0.50/0.50	0.92/0.94	0.95/0.96	0.95/0.96	0.95/0.96
	(8)	0.50/0.50	0.93/0.91	0.96/0.94	0.95/0.94	0.95/0.94
	(9)	0.50/0.50	0.92/0.94	0.95/0.98	0.95/0.97	0.95/0.97
Folds	(10)	0.50/0.50	0.94/0.89	0.96/0.89	0.96/0.89	0.96/0.89
	(ave)	0.50/0.50	0.93/0.92	0.95/0.95	0.95/0.95	0.95/0.95
				Training Performance	Testing Performance	

Detrended + bad-ch rm + SLAP + spectral filter

ERP
AUC



Key functions

- `clsfr=train_erp_clsfr(X,Y,...);`
 - Apply default ERP pipeline to X and train a linear classifier.
 - *Clsfr* contains the classifier and pipeline info
- `f=apply_erp_clsfr(X,clsfr)`
 - Apply ERP pipeline and classifier in *clsfr* to new data X, return classifier prediction *f* as a **decision value**
- `clsfr=train_ersp_clsfr(X,Y,...);`
 - Apply default ERSP pipeline to X and train a linear classifier
 - *Clsfr* contains classifier and pipeline info
- `f=apply_ersp_clsfr(X,clsfr)`
 - Apply ERSP pipeline and classifier in *clsfr* to new data X, return classifier prediction *f* as a **decision value**

Summary : Decoding

- 2 main types of BCI signal : **ERP**, **ERSP**
- EEG **artifacts** and **noise** mean need particular processing steps to correct.
- Simple pipeline of processing removes these artifacts:
 - Detrend / high-pass filter
 - Identify and remove bad channels
 - Spatial filter / reference
 - Spectral filter (ERP)
 - Compute power spectrum and select frequencies (ERSP)
 - Identify and remove bad trials
 - Regularised linear classifier training / application