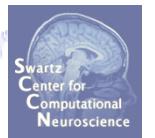


EEGLAB Data Structures



1. EEG
 - .data - root 'dataset' structure
 - .chanlocs - channel locations substructure
 - .event - data events substructure
 - .epoch - data epochs substructure
3. ALLEEG - vector of loaded EEG datasets
4. CURRENTSET - index in ALLEEG of current EEG dataset
5. STUDY
 - root 'studyset' structure
 - .cluster - component clustering substructure



EEG structure

EEG =

```
setname:'Epoched from "ee114 continuous"'  
filename:'ee114squaresepochechs.set'  
filepath:'/home/arno/ee114/'  
pnts:384  
nbchan:32  
trials:80  
srate:128  
xmin:-1  
xmax:1.9922  
data:[32x384x80 double]  
icawinv:[32x32 double]  
icasphere:[32x32 double]  
icaweights:[32x32 double]  
icaact:[32x384x80 double]  
event:[1x157 struct]  
epoch:[1x80 struct]  
chanlocs:[1x32 struct]  
comments:[8x150 char]  
averref:'no'  
rt:[]  
eventdescription:{1x5 cell}  
epochdescription:{}  
specdata:[]  
specicaact:[]  
reject:[1x1 struct]  
stats:[1x1 struct]  
splinofile:[]  
ref:'common'  
history:[7x138 char]  
urevent:[1x154 struct]  
times:[1x384 double]
```

Number of data points per trial

Number of channels

Number of trials

Sampling rate

Time limits

Data

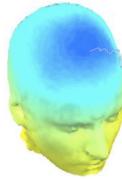
ICA scalp maps

ICA activity

Epoch/event information

Channel location

3 levels of functions



Administrative functions: handle EEG and ALLEEG structures

eeglab(), eeg_checkset(), pop_delset(), ...

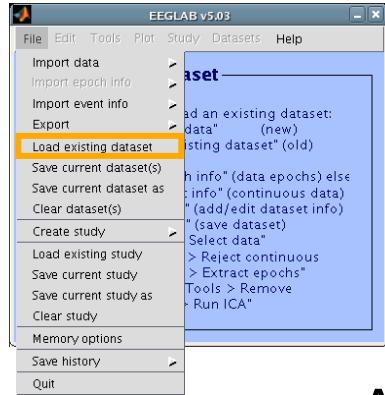
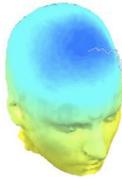
Pop functions: interactive functions using EEG structure

pop_erpimage(), pop_topoplot(), pop_envtopo(), ...

Signal processing functions: perform signal processing

erpimage(), topoplot(), envtopo(), ...

Command line tools



(Menus write both dataset and global history)

- Automated processing on groups of subjects (possibly on several processors).
- Richer options for plotting and processing functions (time-frequency decompositions, ...)
- Selecting data/epoch based on event context
- Custom processing...

Using EEGLAB history for basic scripting



EEG.history → useful information

Task 1

Create simple script using 'eegh'

Task 2

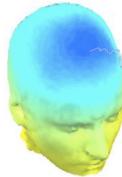
Eye-blink correction

Create a new EEG field

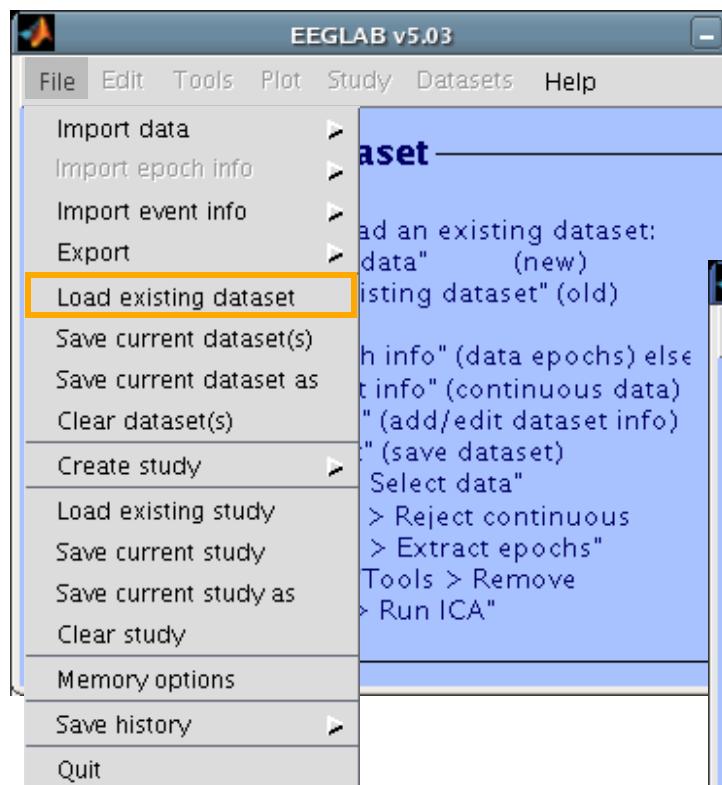
Exercise...



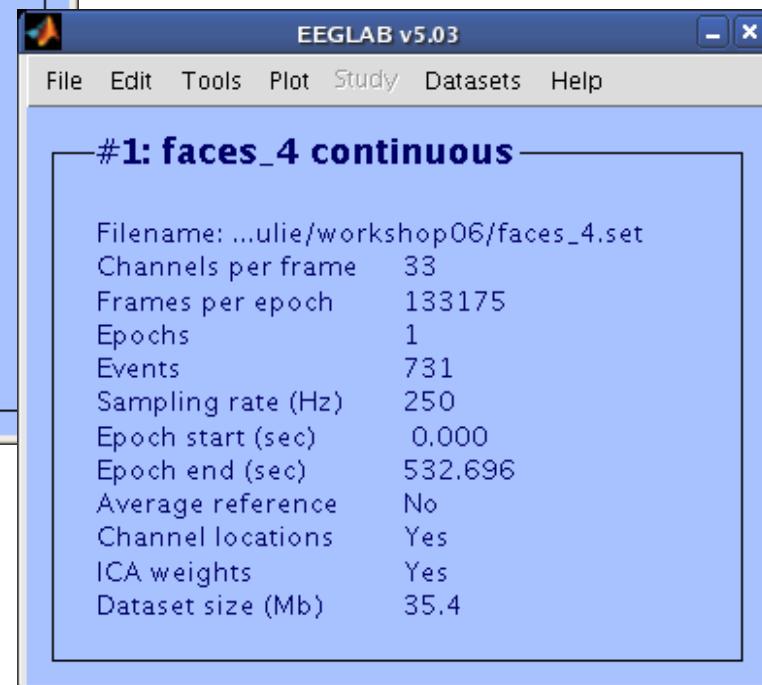
Task 1: Retrieve dataset history



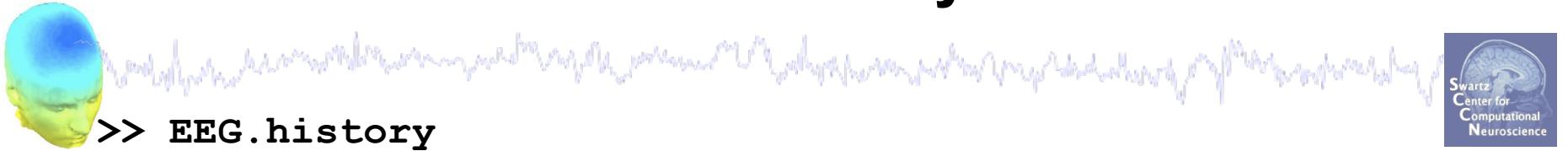
Retrieve information about the processing history of a data set:



>> EEG.history



EEG.history

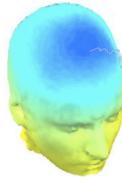


```
>> EEG.history

ans =
EEG = pop_loadbv('.../data/rawdata/', 'faces_4.vhdr');
EEG.setname='faces_4_continuous';
EEG = eeg_checkset( EEG );
EEG.chanlocs=pop_chanedit(EEG.chanlocs, 'load',[],...
'load',{ '.../wsporto/data/chan_locs.elp', 'filetype',...
'besa (elp)' }, 'eval',...
'chantmp = pop_chancenter( chantmp, [],[]);');

EEG = pop_saveset( EEG, 'faces_4.set', '.../workshop/');
EEG = pop_multifit(EEG, [1:33] , 'settings',{},'threshold',...
40, 'plotopt',{ 'normlen', 'on', 'image','fullmri'}));
pop_topoplot(EEG,0, [1:12] , 'faces_4 continuous',...
[3 4] ,0, 'electrodes', 'off', 'masksurf', 'on');
% no history for manual DIPFIT dipole localization
EEG = pop_saveset( EEG, 'faces_4.set', '.../workshop/');
```

Using EEGLAB history for basic scripting



EEG.history → useful information

Task 1

Create simple script using 'eegh'

Task 2

Eye-blink correction

Create a new EEG field

Exercise...



Task 1: plot an ERP image...



EEGLAB v5.03

File Edit Tools Plot Study Datasets Help

#1: faces_4.set

- #1: faces_4.set
- Filename: faces_4.set
- Channels per frame: 33
- Frames per epoch: 250
- Epochs: 364
- Events: 725
- Sampling rate (Hz): 250
- Epoch start (sec): -0.200
- Epoch end (sec): 0.796
- Average reference: No
- Channel locations: Yes
- ICA weights: Yes
- Dataset size (Mb): 24.5

Tools

- Change sampling rate
- Filter the data
- Re-reference
- Reject continuous data by eye
- Extract epochs**
- Remove baseline
- Reject data epochs
- Run ICA
- Remove components
- Reject data using ICA
- Locate dipoles using BESA
- Locate dipoles using DIPFIT 1.x
- Locate dipoles using DIPFIT 2.x
- Laplacian
- FMRIB Tools
- Grand average datasets
- Locate dipoles using LORETA
- PCA plugin

Task 1: Plot an ERPimage...



EEGLAB v5.03

#2: faces

File Edit Tools Plot Study Datasets Help

Channel locations >

Channel data (scroll)

Channel spectra and maps

Channel properties

Channel ERP image

Channels per frame

Frames per epoch

Epochs

Event types

Sampling rate

Epoch length

Epoch duration

Average

Channel selection

ICA with

Datasources

Channel

Smoothing

Downsampling

Time limits (ms)

21

10

1

-200 800

Figure title

Plot scalp map

Plot ERP

Plot colorbar

ERP limits (uV)

Color limits (see Help)

Sort/align trials by epoch event values

Epoch-sorting field

Event type(s)

Event time range

Rescale

Align

Don't sort by value

Don't plot values

Sort trials by phase

Frequency (Hz | minHz maxHz)

Percent low-amp. trials to ignore

Window center (ms)

Wavelet cycles

3

Inter-trial coherence options

Frequency (Hz | minHz maxHz)

Signif. level (<0.20)

Amplitude limits (dB)

Coher limits (<=1)

Image amps
(Requires signif.)

Other options

Plot spectrum (minHz maxHz)

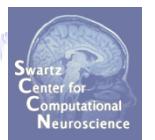
Baseline ampl. (dB)

Mark times (ms)

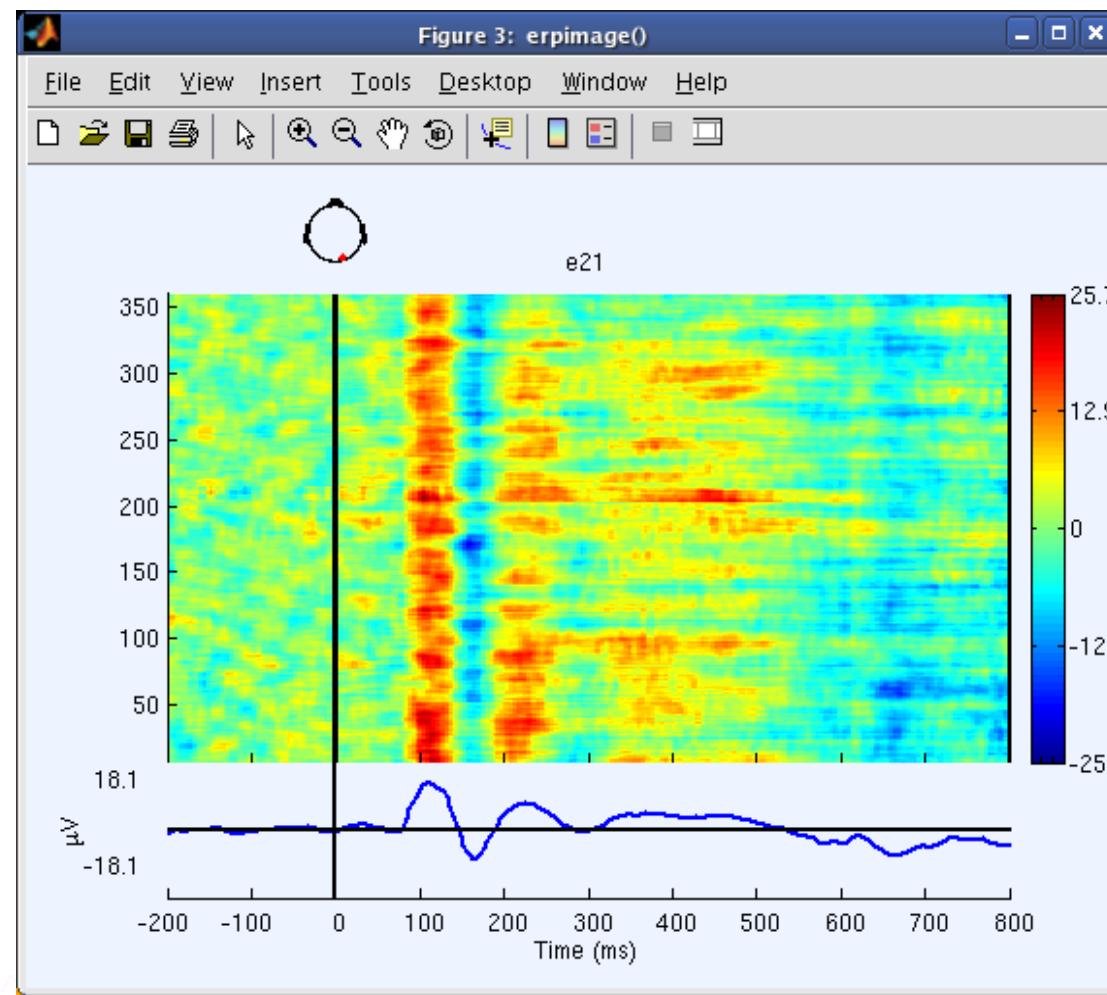
More options (see >> help erpimage)

Cancel Help Ok

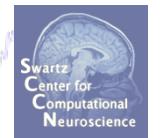
Channel ERP image -- pop_erpimage()



Task 1: Resulting figure



Script task 1 using 'eegh'



Write a script to do this:

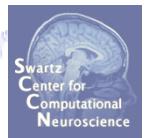
```
>> eegh
```



Script task 1 using 'eegh'



```
>> eegh  
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;  
  
EEG = pop_loadset('filename','faces_4.set','filepath',...  
'...\data\');  
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 0);  
  
EEG = pop_epoch( EEG, { 'face' 'object' }, [-0.2 0.8],...  
'newname', 'faces_4 epochs', 'epochinfo', 'yes');  
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1);  
EEG = pop_rmbase( EEG, [-200 0]);  
[ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);  
  
figure; pop_erpimage(EEG,1, [21],[],'e21',10,1,{},[],...  
'', 'yerplabel', '\muV', 'topo',...  
{ [21] EEG.chanlocs EEG.chaninfo } , 'erp' , 'cbar');
```



Using EEGLAB history for basic scripting



EEG.history → useful information

Task 1

Create simple script using 'eegh'

Task 2

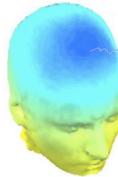
Eye-blink correction

Create a new EEG field

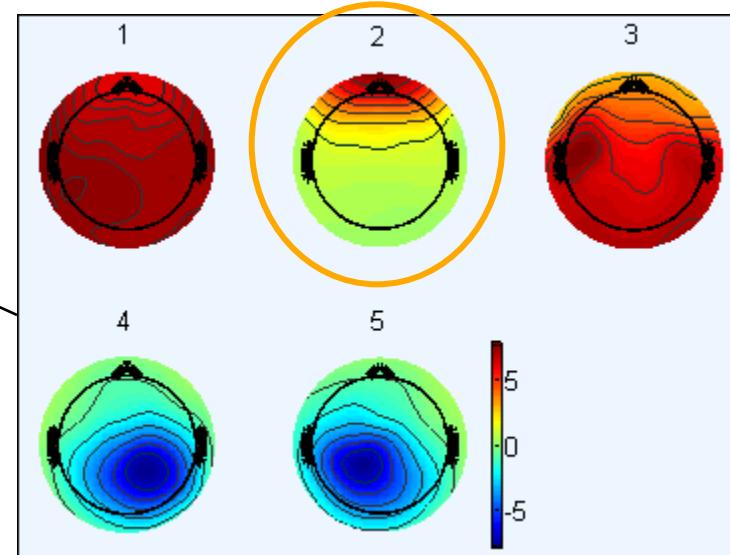
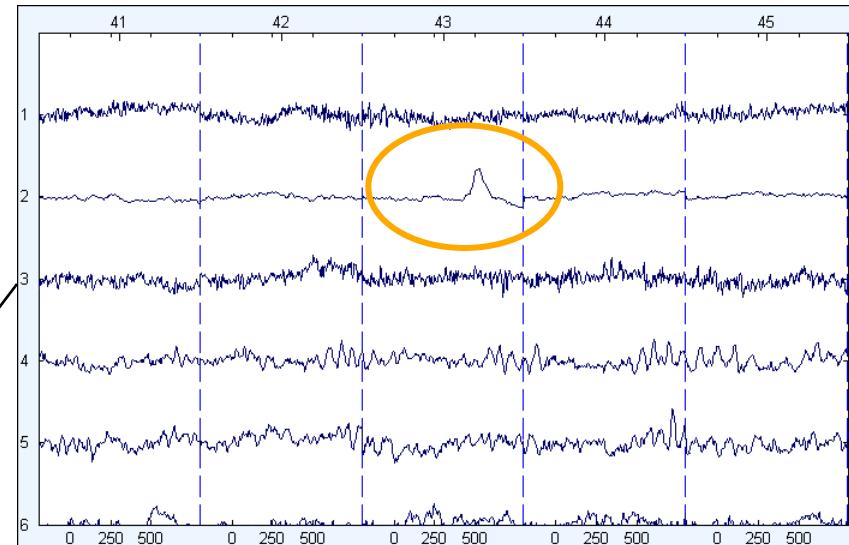
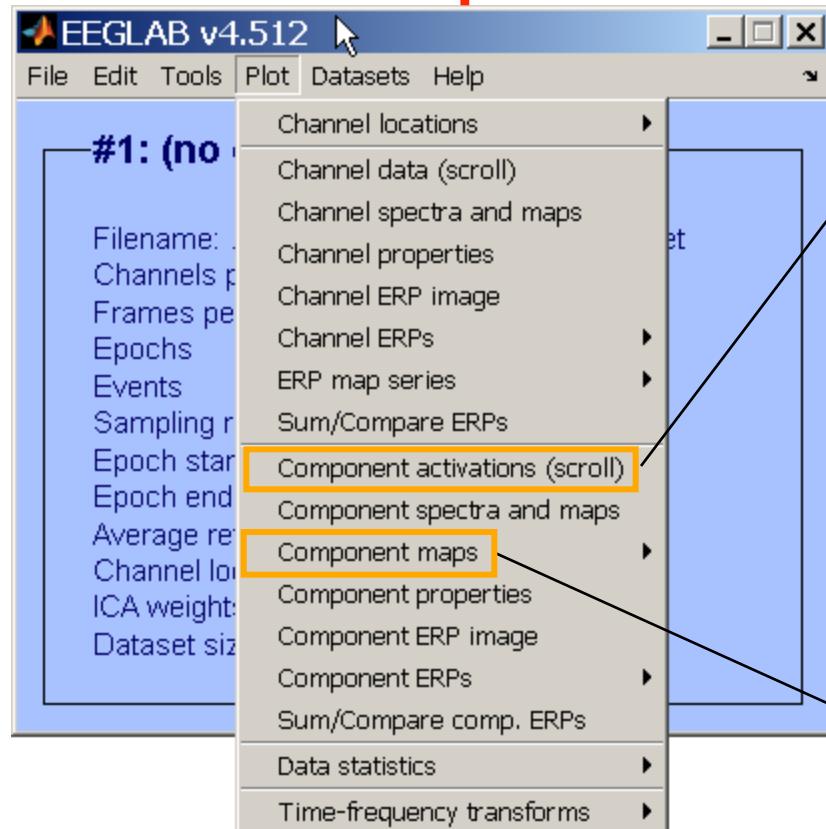
Exercise...



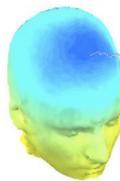
Eye blink correction



Identify eye-blink
components:

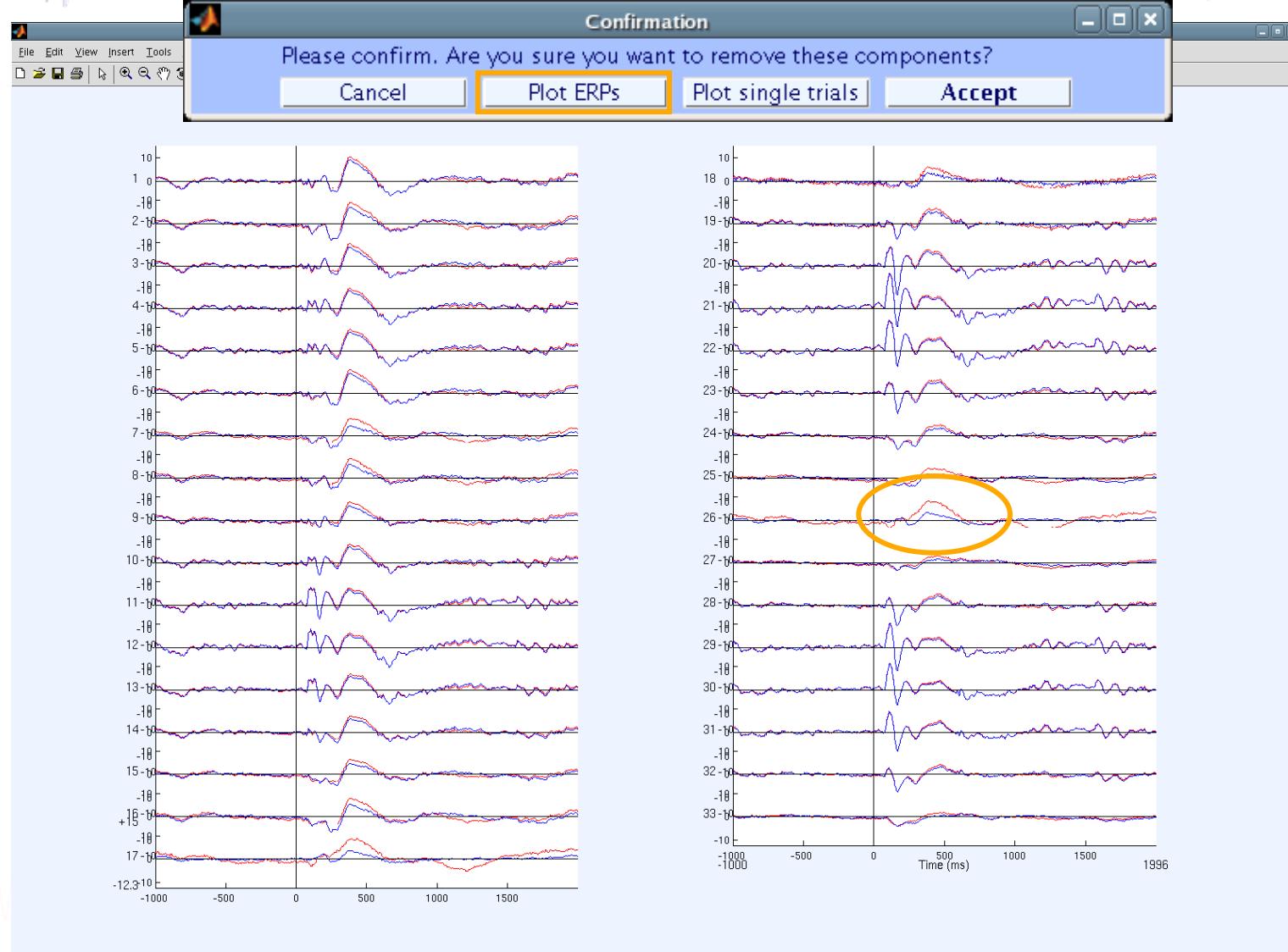


Eye blink correction

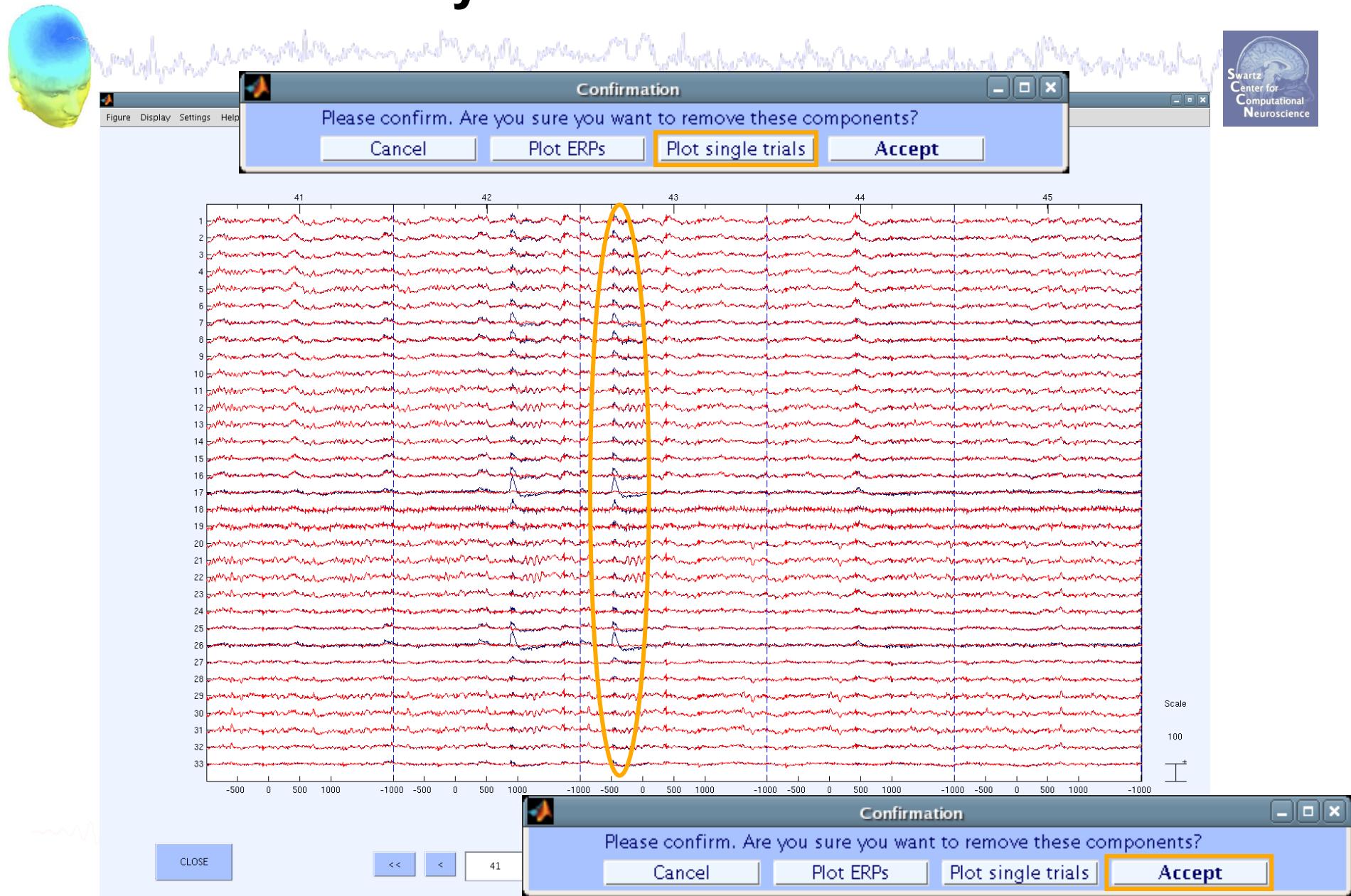


The image shows the EEGLAB v6.0b software interface. The main window title is "EEGLAB v6.0b". The "Tools" menu is open, displaying various processing options. The "Remove components" option is highlighted with a blue selection bar. Below the main window, a smaller dialog box titled "Remove components from data -- pop_subcomp()" is displayed. It contains two input fields: "Component(s) to remove from data:" and "Component(s) to retain (overwrites 'Component(s) to remove')". The number "2" is typed into the second field. This field is circled in orange. At the bottom of the dialog box are three buttons: "Cancel", "Help", and "Ok".

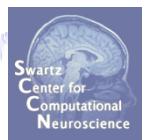
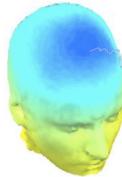
Eye blink correction



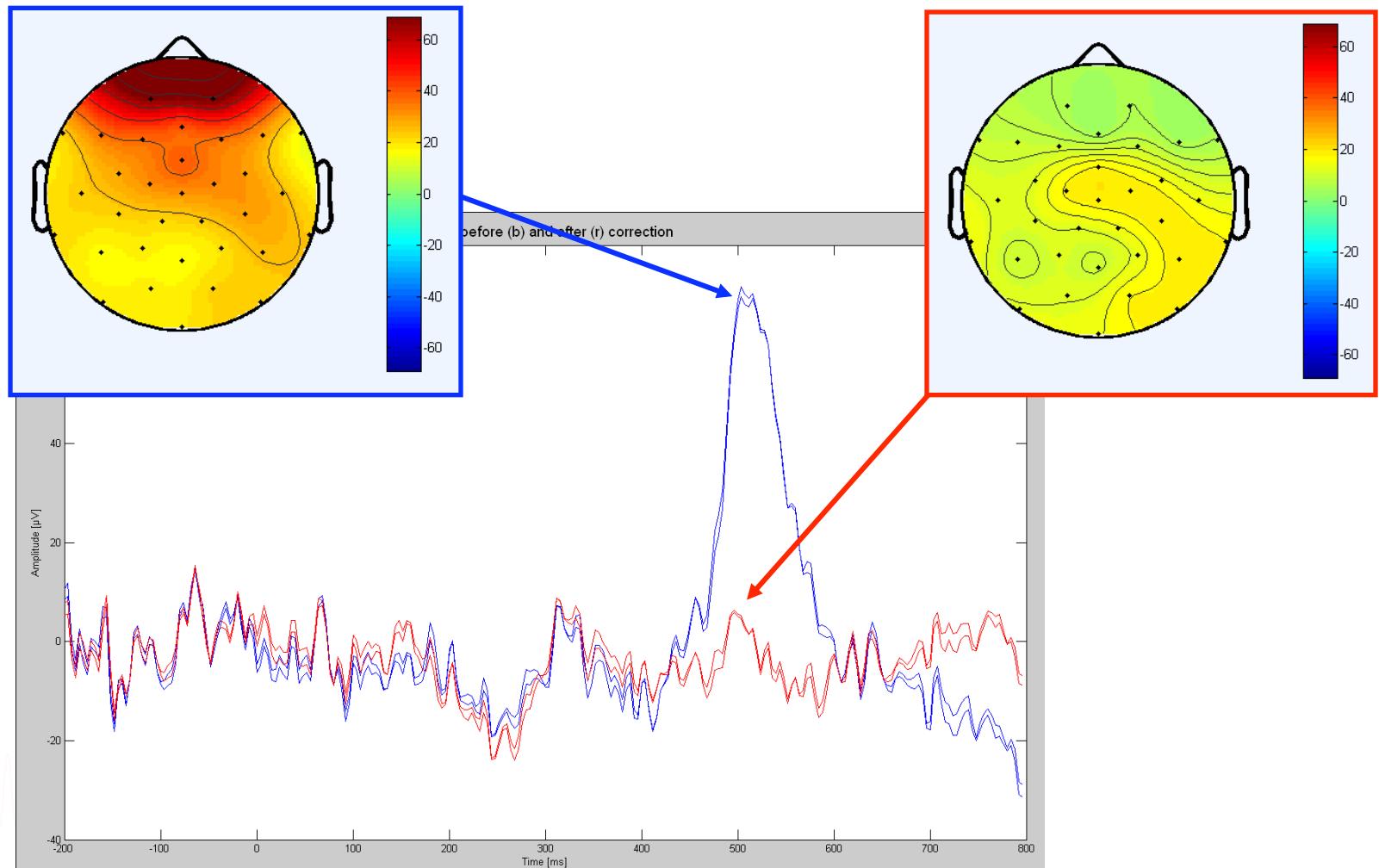
Eye blink correction



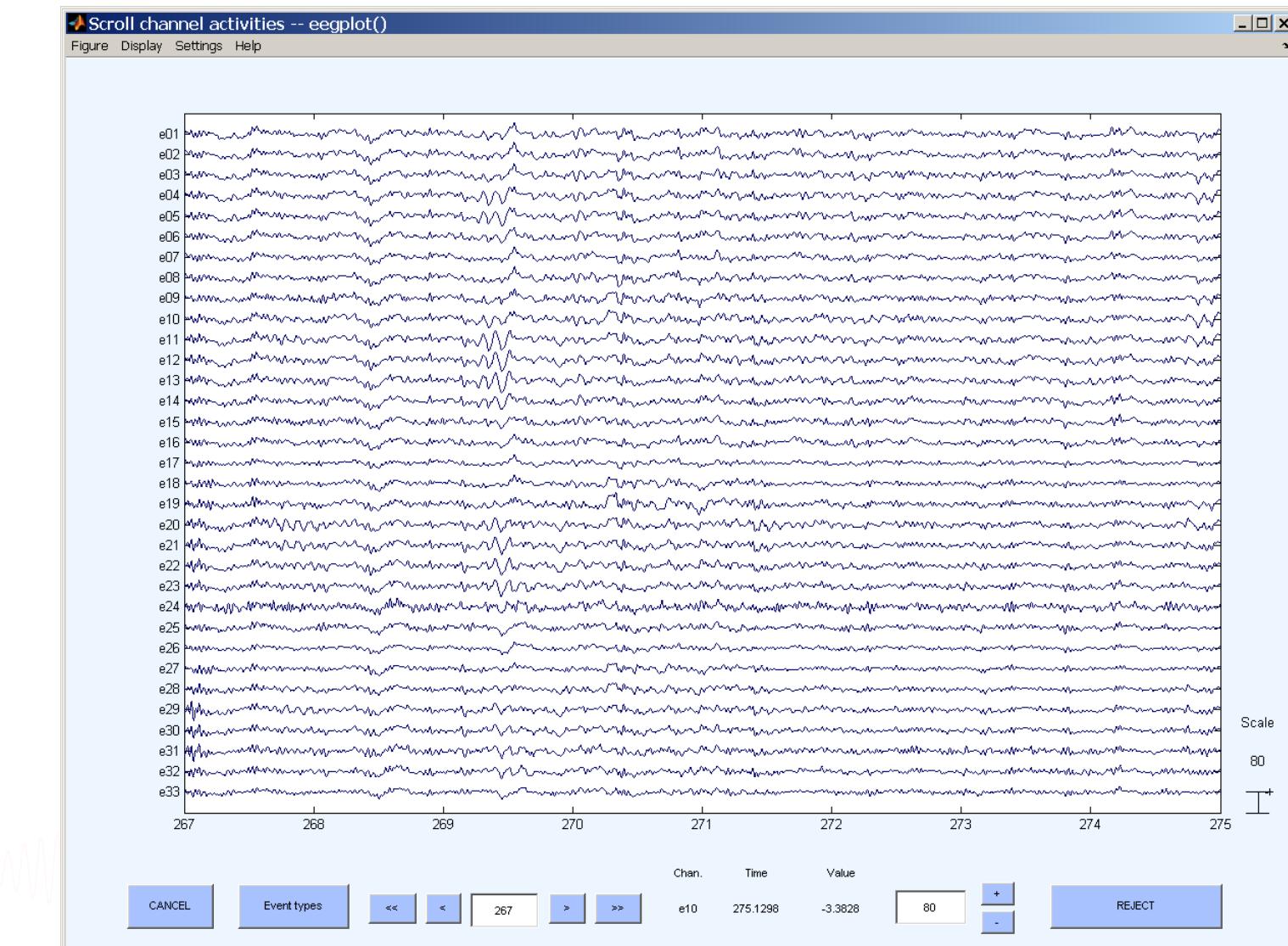
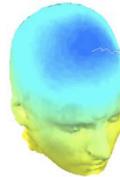
Eye blink correction



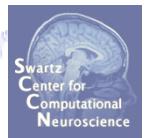
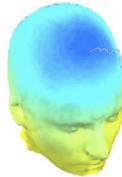
Trial 43: Fp1/2 before (b) and after (r) correction



Task 2: Eye blink correction



Task 2: Script an addition to EEG structure



- EEG structure can be extended to include new fields
 - store information for future access
- Task:
 - write a semi-automatic script to save eye blink IC index as '**EEG.blink**'



Create initial script from 'eegh'



```
>> eeglab

% using GUI:
% load dataset,
% plot component maps in 2D
% save current dataset as... (force a resave)

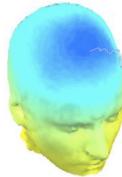
>> eegh

% open Matlab editor:
>> edit

% copy & paste eegh results into a new
% file and save it as faces2.m
```



Using EEGLAB history for basic scripting



EEG.history → useful information

Task 1

Create simple script using 'eegh'

Task 2

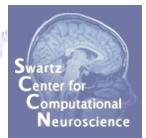
Eye-blink correction

Create a new EEG field

Exercise...



Exercise



Script it yourself!

- **Novice**

Script using eegh:

load a dataset, epoch on 'face' and 'object', plot erpimage for any channel or component, copy eegh results to a script file and run your script!

- **Intermediate**

Script a semi-automatic script to add an EEG.blink field into the EEG structure.

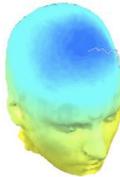
** All scripts for exercises can be found at

<http://sccn.ucsd.edu/wiki/EEGLAB09ASPET>

(ALL "DATPATH" VARIABLES MUST BE CHANGED TO POINT
TO THE DATA ON YOUR COMPUTER!)



Exercise: NOVICE



```
>> eeglab
```

```
% repeat all steps of task 1:  
% load dataset,  
% epoch on 'face' and 'object'  
% plot erpimage for any channel
```

```
>> eegh
```

```
% open Matlab editor:  
>> edit
```

```
% copy & paste eegh results into a new  
% file and save it (../faces1.m)
```

```
>> clear  
>> close all  
>> faces1  
>> eeglab redraw
```

For reference: example script
saved as: '...\\scripts\\practicum_4.m'

