EEGLAB Data Structures

1. EEG - root ‘dataset’ structure
   .data - the dataset data (2-D, 3-D matrix)
   .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

setname: "Epoched from "ee114 continuous"
filename: "ee114squaresepochs.set"
filepath: "/home/amo/ee114/

trials: 80
srate: 128
xmin: -1
xmax: 1.9922

data: [32x384x80 double]
icawinv: [32x32 double]
icasphere: [32x32 double]
icaweights: [32x32 double]
icact: [32x364x80 double]
event: [1x157 struct]
epoch: [1x60 struct]
chaniocs: [1x32 struct]
comments: [8x150 char]
aref: 'no'

 eventdata: [1x5 cell]
epochdescription: {}
specdata: {}
specificact: []
reject: [1x1 struct]
stats: [1x1 struct]
splinefile: []
ref: 'common'
history: [7x138 char]
urevent: [1x154 struct]
times: [1x384 double]
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

File: "...ulie/workshop06/faces_4.set"
Channels per frame: 33
Frames per epoch: 1,331,750
Epochs: 1
Events: 731
Sampling rate (Hz): 250
Epoch start (sec): 0.000
Epoch end (sec): 532.696
Average reference: No
Channel locations: Yes
ICA weights: Yes
Dataset size (Mb): 35.4
% 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...

#1: filenamex

#2: faces_4_epochs

Filename: none
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
Task 1: Plot an ERP image...
Task 1: Resulting figure
Script task 1 using 'eegh'

Write a script to do this:

>> eegh
Script task 1 using 'eegh'

```matlab
>> 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
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'