EEGLAB Plugins/Extensions
Starting EEGLAB

>> eeglab
  eeglab: options file is /Volumes/donnees/data/STUDYstern/eeg_options.m
  Adding path to all EEGLAB functions
  Adding path to eeglab/external/bioelectromagnetism_light
  Adding path to eeglab/external/biosig-partial
  Adding path to eeglab/external/fieldtrip-partial
  Adding path to eeglab/external/fieldtrip-partial subfolders
  EEGLAB: adding plugin function "eegplugin_VisEd"
  EEGLAB: adding "eepimport1.02" plugin (see >> help eegplugin_eepimport)
  EEGLAB: adding "bdfimport" plugin (see >> help eegplugin_bdfimport)
  EEGLAB: adding "brainmovie0.1b" plugin (see >> help eegplugin_brainmovie)
  EEGLAB: adding "ctfimport1.03" plugin (see >> help eegplugin_ctfimport)
  EEGLAB: adding "dipfit2.2" plugin (see >> help eegplugin_dipfit)
  EEGLAB: adding "EEG toolbox ERP plotting" plugin (see >> help eegplugin_eeg_toolbox)
  EEGLAB: adding "erpssimport1.00" plugin (see >> help eegplugin_erpssimport)
  EEGLAB: adding "fmrib1.21" plugin (see >> help eegplugin_fmrib)
  EEGLAB: adding "iirfilt1.01" plugin (see >> help eegplugin_iirfilt)
  EEGLAB: adding "eepimport1.02" plugin (see >> help eegplugin_ascinstep)
  EEGLAB: adding "loreta1.0" plugin (see >> help eegplugin_loreta)
  EEGLAB: adding "Butter1.0" plugin (see >> help eegplugin_ERPLAB_filters)
  EEGLAB: adding "Measure_Product1.0" plugin (see >> help eegplugin_mp_clustering)
  EEGLAB: adding plugin function "eegplugin_miclust"
  EEGLAB: adding "4dneuroimaging1.00" plugin (see >> help eegplugin_4dneuroimaging)
  >>
**EEGLAB plugins**

<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eepimport1.02</td>
<td>Data importing for EEprobe data (Oostenveld &amp; ANT company)</td>
</tr>
<tr>
<td>bva_io1.30</td>
<td>Brain vision analyzer import/export plugin (Widmann &amp; Delorme)</td>
</tr>
<tr>
<td>ctfimport1.01</td>
<td>MEG CTF import plugin (Carver, Weber &amp; Delorme)</td>
</tr>
<tr>
<td>dipfit2.0</td>
<td>4-shell and BEM (Oostenveld &amp; Delorme)</td>
</tr>
<tr>
<td>fmrib1.2b</td>
<td>Removal of artifact from simultaneously EEG/fMRI recording (Niazi)</td>
</tr>
<tr>
<td>icaclust1.00</td>
<td>Clustering ICA components (Serby, Delorme, Makeig)</td>
</tr>
<tr>
<td>iirfilt1.0</td>
<td>Non-linear IIR filtering (Pozdin)</td>
</tr>
<tr>
<td>loreta1.0</td>
<td>Interface to LORETA-KEY (Delorme)</td>
</tr>
<tr>
<td>newtimefreq1.00</td>
<td>Time-freq. decomposition (Delorme)</td>
</tr>
</tbody>
</table>

**Matlab toolboxes interfaced**

- **BIOSIG**: Data importing for rare data binary format (Schloegl)
- **Fieldtrip**: Source localization and time-freq. decompositions (Oostenveld)
- **ICALAB**: 20 ICA algorithms (automatically detected by EEGLAB)
- **SPM2**: Spatial normalization of anatomical MRI
Plugin list process

<table>
<thead>
<tr>
<th>Plugin name</th>
<th>Version</th>
<th>Short plugin description</th>
<th>Link</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>brainmovie</td>
<td>0.1</td>
<td>Brainmovies (command line only)</td>
<td>Download</td>
<td>User comments</td>
</tr>
<tr>
<td>corrmap</td>
<td>1.03</td>
<td>Import BIOPAC data files</td>
<td>Download</td>
<td>User comments</td>
</tr>
<tr>
<td>eeg_toolbox</td>
<td>1.0</td>
<td>Interface EEG toolbox functions for ERP peak detection</td>
<td>Download</td>
<td>User comments</td>
</tr>
<tr>
<td>ERPLABfilters</td>
<td>1.00</td>
<td>Interface ERPLAB filters (requires separate ERPLAB installation)</td>
<td>Download</td>
<td>User comments</td>
</tr>
<tr>
<td>fMRIb</td>
<td>1.21</td>
<td>Remove fMRI artifacts from EEG</td>
<td>Download</td>
<td>User comments</td>
</tr>
<tr>
<td>MP_clustering</td>
<td>1.00</td>
<td>Measure projection clustering of ICA components</td>
<td>Download</td>
<td>User comments</td>
</tr>
<tr>
<td>MutualInfoClustering</td>
<td>1.00</td>
<td>Mutual information clustering</td>
<td>Download</td>
<td>User comments</td>
</tr>
<tr>
<td>StudyEnvtopo</td>
<td>0.9</td>
<td>Add envtopo capabilities to STUDY</td>
<td>Download</td>
<td>User comments</td>
</tr>
<tr>
<td>VisEd</td>
<td>1.04</td>
<td>Add/Edit dataset events</td>
<td>Download</td>
<td>User comments</td>
</tr>
<tr>
<td>ADJUST</td>
<td>1.21</td>
<td>Automatic artifact rejection</td>
<td>Download</td>
<td>User comments</td>
</tr>
<tr>
<td>irfit</td>
<td>1.02</td>
<td>Non linear filtering</td>
<td>Download</td>
<td>User comments</td>
</tr>
<tr>
<td>loreta</td>
<td>1.0</td>
<td>Export and import data to/from LORETA software</td>
<td>Download</td>
<td>User comments</td>
</tr>
<tr>
<td>BERGEN</td>
<td>1.1</td>
<td>Removal of fMRI-related gradient artifacts from simultaneous EEG-fMRI data</td>
<td>Download</td>
<td>User comments</td>
</tr>
</tbody>
</table>

Add your plugin to the list

You may add your plugin to the list so users can download it automatically from within EEGLAB. There are 5 tabs:

- **Plugin name**: this tab should contain the abbreviated name of your plugin and if necessary a link to the plugin documentation. The plugin documentation may be stored on this wiki.
- **Version**: this tab should contain the version of your plugin. The version listed on this page and the one made available in the eegplugin_vxx.m file must be consistent. This allows EEGLAB to automatically check for newer versions of your plugin.
- **Short plugin description**: this tab should contain a short plugin description (no more than one line). Additional documentation may be provided as a link in tab 1.
Writing EEGLAB plugins

- Assuming that you have a signal processing function called `xxxxxx`
  - a `pop_xxxxx` function will interface your signal processing function
  - a `eegplugin_xxxxx` function will add the menu to the main interface (and history etc…)

```
Process any Input data
```

```
Process EEG structure
```

```
Timef()
```

```
Pop_timef()
```
Pop functions

- Called with the EEG structure only pop_xxxxx(EEG), they pop-up a GUI asking for more arguments
- Called with enough arguments, the simply call the signal processing function

```matlab
function [EEG, com] = pop_sample( EEG, param1 );

com = ''; % empty history
if nargin < 2
    % pop up window if less than 2 arguments
    result = inputdlg({'Enter the parameter:' }, 'Title of window', 1, { '0' })
    if length( result ) == 0 return; end;

    param1 = eval( [ ' [' result{1} ' ]' ] ); % the brackets allow to process matlab arrays
end;

sample( EEG.data, param1); % run sample function

com = sprintf('pop_sample(EEG, %d );', param1); % return history
```
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

2. ALLEEG - vector of loaded EEG datasets

3. CURRENTSET - index in ALLEEG of current EEG dataset

4. STUDY - root ‘studyset’ structure
   .cluster - component clustering substructure
EEG =

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

data:384
nbchan:32
trials:80
srate:128
xmin:-1
xmax:1.0922

data:[32x384x80 double]
icawinv:[32x32 double]
icasphere:[32x32 double]
icaweights:[32x32 double]
icact:[32x364x80 double]
event:[1x157 struct]
epoch:[1x60 struct]
chanlocs:[1x32 struct]
comments:[8x150 char]
averf:'no'
eventdescription:[1x5 cell]
ePOCHdescription:[]

ICA scalp maps
ICA activity
Epoch/event information
Channel location

Number of data points per trial
Number of channels
Number of trials
Sampling rate
Time limits
Data

EEG structure
The EEG structure can be extended to include new fields to store information for future access.

```
EEG =
    setname: 'Epoched from "ee114 continuous"'
    filename: 'ee114squaresepocheeg'
    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]
    icacact: [32x364x80 double]
    event: [1x157 struct]
    epoch: [1x80 struct]
    chanlocs: [1x32 struct]
    comments: [8x150 char]
    averef: 'no'
    eventdescription: [1x6 cell]
    epochdescription: {}
    specdata: []
    specicaact: []
      reject: [1x1 struct]
      stats: [1x1 struct]
      spinefile: []
        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
Continuous data

\[
\text{EEG.data} = \begin{bmatrix}
    2.1 & 3.8 & 4.9 & 5.1 & 4.8 & 3.9 \\
    -1.3 & -2.4 & -0.5 & -0.3 & 1.4 & 2.5 \\
    5.2 & 4.7 & 3.3 & 1.2 & 0.7 & 1.3 \\
\end{bmatrix}
\]
Data epochs

\[
\begin{bmatrix}
2.1 & 3.8 & 4.9 & 5.1 & 4.8 & 3.9 \\
-1.3 & -2.4 & -0.5 & -0.3 & 1.4 & 2.5 \\
5.2 & 4.7 & 3.3 & 1.2 & 0.7 & 1.3
\end{bmatrix}
\]

Trials 1: EEG.data(:,:,1)

\[
\begin{bmatrix}
2.1 & 3.8 & 4.9 & 5.1 & 4.8 & 3.9 \\
-1.3 & -2.4 & -0.5 & -0.3 & 1.4 & 2.5 \\
5.2 & 4.7 & 3.3 & 1.2 & 0.7 & 1.3
\end{bmatrix}
\]

Trials 2: EEG.data(:,:,2)

\[
\begin{bmatrix}
2.1 & 3.8 & 4.9 & 5.1 & 4.8 & 3.9 \\
-1.3 & -2.4 & -0.5 & -0.3 & 1.4 & 2.5 \\
5.2 & 4.7 & 3.3 & 1.2 & 0.7 & 1.3
\end{bmatrix}
\]

Trials 3: EEG.data(:,:,3)

Plot ERP for your data

\[
>> \text{figure; plot(mean(EEG.data,3))';}
\]

\[
>> \text{figure; plot(EEG.times, mean(EEG.data,3)')};
\]
eegplugin functions

• eegplugin_xxxxx function

% eegplugin_erp() - plot ERP plugin

function eegplugin_erp( fig, try_strings, catch_strings);

% create menu
plotmenu = findobj(fig, 'tag', 'plot'); % find plot menu

% create submenu
uimenu( plotmenu, 'label', 'ERP plugin', ...
   'callback', 'figure; plot(EEG.times, mean(EEG.data,3));');
eegplugin functions

>> eeglab
eeglab: adding "BIOSIGv0.86" plugin
eeglab: adding "eepimport1.02" plugin (see >> help eegplugin_eepimport)
eeglab: adding "bva_io1.30" plugin (see >> help eegplugin_bva_io)
eeglab: adding "ctfimport1.01" plugin (see >> help eegplugin_ctfimport)
eeglab: adding "dipfit2.0" plugin (see >> help eegplugin_dipfit2_0)
eeglab: adding plugin function "eegplugin_erp"
eeglab: adding "fmrib1.2b" plugin (see >> help eegplugin_fmrib)
eeglab: adding "iicalust1.00" plugin (see >> help eegplugin_iicalust)
eeglab: adding "iirfilt1.0" plugin (see >> help eegplugin_iirfilt)
eeglab: adding "loreta1.0" plugin (see >> help eegplugin_loreta)
eeglab: adding "newtimefreq1.00" plugin (see >> help eegplugin_ne
PCA plugin

function vers = eegplugin_pca(fig, trystrs, catchstrs)

    vers = 'pca1.00';
    if nargin < 3, error('eegplugin_pca requires 3 arguments'); end;

    % add icaclust folder to path
    if ~exist('eegplugin_pca')
        p = which('eegplugin_pca');
        p = p(1:findstr(p,'eegplugin_pca.m')-1);
        addpath( p );
    end;

    % find tools menu
    menu = findobj(fig, 'tag', 'tools');

    % PCA command
    cmd = ['[tmp1 EEG.icawinv] = runpca(EEG.data(:,:));' '];
    cmd = [ cmd 'EEG.icaweights = pinv(EEG.icawinv);' '];
    cmd = [ cmd 'EEG.icasphere = eye(EEG.nbchan);' '];
    cmd = [ cmd 'clear tmp1;' '];

    % create menu
    uimenu( menu, 'Label', 'Run PCA', 'CallBack', cmd, 'separator', 'on');
EEGLAB documentation

EEGLAB Home Page  sccn.ucsd.edu/eeglab/
EEGLAB Tutorial Index  sccn.ucsd.edu/wiki/EEGLAB

- 200 pages of tutorial (including “how to” for plugins) WEB or PDF
- Function documentation (next slide)
- Send questions to the mailing list eeglablist@sccn.ucsd.edu
  (or search mailing list archive using google)
- Bug submission http://sccn.ucsd.edu/eeglab/bugzilla
- Email us (suggestions) eeglab@sccn.ucsd.edu
- Workshop with practicum every year
Out of Memory

• Solution
  – Buy more RAM
  – Use option “use memory mapped arrays”
  – Close all programs, remove Windows services (Adobe etc…), reboot
  – Try different memory manager “start Matlab from the DOS command line with `matlab –memmgr fast` option
  – Use older Matlab versions (how old?)
  – Change of OS (Windows 10 or 8 vs 7)

• OSx and Unix/Linux
  - Buy more RAM
  - Matlab cannot allocated inactive memory. To free it, type `du -sx /`
Figure problem and export

For crashes, freezes, etc… use a non Open-GL renderer

At startup type in `set(0,'defaultfigurerenderer','zbuffer' )` or `set(0,'defaultfigurerenderer','painter' )`. Note that these cannot handle transparency and 3-D graphics or type “opengl software”

To export figures for publication, use .eps format (postscript) and edit for instance with adobe illustrator. Use “`set(gcf, ‘renderer’, ‘painter’)`” before exporting complex figures. Use the “plot2svg” matlab toolbox to export figure for transparency.
Exercice

Write a plugin to plot ERPs