Collect high-density EEG data (>30 chan) → Import into EEGLAB → Import event markers and channel locations

Re-reference/down-sample (if necessary) → High pass filter (~0.5 – 1 Hz) → Examine raw data

Reject bad channels → Reject large artifact time points → Run ICA

EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started
Dense-array EEG
Pre-processing pipeline

1. Collect high-density EEG data (>30 chan)
2. Import into EEGLAB
3. Import event markers and channel locations
4. Re-reference/down-sample (if necessary)
5. High pass filter (~.5 – 1 Hz)
6. Examine raw data
7. Reject bad channels
8. Reject large artifact time points
9. Run ICA
EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started

EEGLAB Matlab toolbox

main graphic interface

```
/home/arno> matlab -nodesktop

< MATLAB >
Copyright 1984-2002 The MathWorks, Inc.
Version 6.5.0.180913a Release 13
Jun 18 2002

Using Toolbox Path Cache. Type "help toolbox_path_cache" for

To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.

>> eeglab

- Create a new or load an existing dataset:
  Use "File > Import data" (new)
  Or "File > Load existing dataset" (old)
- If new,
  "File > Import epoch info" (data epochs) else
  "File > Import event info" (continuous data)
  "Edit > Dataset info" (add/edit dataset info)
  "File > Save dataset" (save dataset)
- Prune data: "Edit > Select data"
- Reject data: "Tools > Reject continuous
- Epoch data: "Tools > Extract epochs"
- Remove baseline: "Tools > Remove
- Run ICA: "Tools > Run ICA"
```
Importing a dataset

EEGLAB supports many different raw data formats...
Imported EEG data

EEGLAB GUI displays dataset basics
Load an existing dataset

Load dataset(s) -- pop_loadset()
Pre-processing pipeline

1. Collect high-density EEG data (>30 chan)
2. Import into EEGLAB
3. Import event markers and channel locations
4. Re-reference/down-sample (if necessary)
5. High pass filter (~.5 – 1 Hz)
6. Examine raw data
7. Reject bad channels
8. Reject large artifact time points
9. Run ICA
Import data events

- Import events from Matlab array or ASCII file
- Import events from data channel
- Import from Presentation event file
- Import from Neuroscan file

Often imported automatically during data import
Appearance of an event channel in raw data
Imported data events

>> EEG.event

ans =

1x1303 struct array with fields:
  Trial
  Event_Type
  type
  latency
  TTime
  Uncertainty
  Duration
  Uncertainty2
  ReqTime
  ReqDur
  init_index
  init_time
  urevent
  duration
  load
  rt

If event import was successful, you will see an appropriate number here
Review event values

Insert event BEFORE current event

Most relevant fields

Number of event fields is unlimited

Delete CURRENT event

Append event AFTER current event

To resort: first select Main sorting field
Import channel locations

Several file formats supported (Polhemus, BESA, El Guide ...)

EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started
Import channel locations

EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started
Imported channel locations

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Filename:</strong></td>
<td>none</td>
</tr>
<tr>
<td><strong>Channels per frame</strong></td>
<td>70</td>
</tr>
<tr>
<td><strong>Frames per epoch</strong></td>
<td>610133</td>
</tr>
<tr>
<td><strong>Epochs</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Events</strong></td>
<td>1303</td>
</tr>
<tr>
<td><strong>Sampling rate (Hz)</strong></td>
<td>250</td>
</tr>
<tr>
<td><strong>Epoch start (sec)</strong></td>
<td>0.000</td>
</tr>
<tr>
<td><strong>Epoch end (sec)</strong></td>
<td>2440.528</td>
</tr>
<tr>
<td><strong>Reference</strong></td>
<td>CZ</td>
</tr>
<tr>
<td><strong>Channel locations</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>ICA weights</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Dataset size (Mb)</strong></td>
<td>349</td>
</tr>
</tbody>
</table>
**Pre-processing pipeline**

1. Collect high-density EEG data (>30 chan)
2. Import into EEGLAB
3. Import event markers and channel locations
4. Re-reference/down-sample (if necessary)
5. High pass filter (~0.5 – 1 Hz)
6. Examine raw data
7. Reject bad channels
8. Reject large artifact time points
9. Run ICA
Re-reference data (if necessary/desired)

For example, average reference

Current data reference state is: unknown

- **Compute average reference**
- **Re-reference data to channel(s):**
- **Retain old reference channels in data**
- **Exclude channel indices (EMG, EOG)**
- **Add current reference channel back to the data**

Optional:

LEYE  REYE

EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started
Re-reference data (if necessary/desired)

OR, re-reference to (i.e.) 'linked mastoids'

$\text{EEG} = \text{pop\_reref}(\text{EEG}, 39)$;

EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started
Save new dataset, keep old one

[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1, 'setname', ...
'Sternberg Continuous -- Reref''d');

EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started
Multiple active datasets (ALLEEG)

#1: Sternberg Continuous Data
- Filename: ...\STUDY\S04stern.set
- Channels per frame: 71
- Frames per epoch: 810133
- Epochs: 1
- Events: 1303
- Sampling rate (Hz): 250
- Epoch start (sec): 0.000
- Epoch end (sec): 2440.528
- Reference: unknown
- Channel locations: Yes
- ICA weights: Yes
- Dataset size (Mb): 698

#2: Sternberg Continuous
- Filename: none
- Channels per frame: 71
- Frames per epoch: 610133
- Epochs: 1
- Events: 1303
- Sampling rate (Hz): 250
- Epoch start (sec): 0.000
- Epoch end (sec): 2440.528
- Reference: average
- Channel locations: Yes
- ICA weights: No
- Dataset size (Mb): 351.3
Pre-processing pipeline

1. Collect high-density EEG data (>30 chan)
2. Import into EEGLAB
3. Import event markers and channel locations
4. Re-reference/down-sample (if necessary)
5. High pass filter (~0.5 – 1 Hz)
6. Examine raw data
7. Reject bad channels
8. Reject large artifact time points
9. Run ICA
Filter the data (if necessary/desired)

Lower cut off frequencies require longer stretches of continuous data

High-pass needed for ICA
Pre-processing pipeline

1. Collect high-density EEG data (>30 chan)
2. Import into EEGLAB
3. Import event markers and channel locations
4. Re-reference/down-sample (if necessary)
5. High pass filter (~0.5 – 1 Hz)
6. Examine raw data
7. Reject bad channels
8. Reject large artifact time points
9. Run ICA
Scroll channel data

>> pop_eegplot(EEG,1,1,1);
Scroll channel data

- Channels
- Time
- Events
- sec/epoch
- Scroll buttons
- Event markers
- Scaling

EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started
Pre-processing pipeline

1. Collect high-density EEG data (>30 chan)
2. Import into EEGLAB
3. Import event markers and channel locations
4. Re-reference/down-sample (if necessary)
5. High pass filter (~0.5 – 1 Hz)
6. Examine raw data
7. Reject bad channels
8. Reject large artifact time points
9. Run ICA
Remove channel

1) Identify bad channel
Remove channel(s)
Removing channel(s)

If not checked, will result in dataset with one channel
Channel removed

Channel data without 'F6' (see supplementary material for interpolation)
Pre-processing pipeline

1. Collect high-density EEG data (>30 chan)
2. Import into EEGLAB
3. Import event markers and channel locations
4. Re-reference/down-sample (if necessary)
5. High pass filter (~.5 – 1 Hz)
6. Examine raw data
7. Reject bad channels
8. Reject large artifact time points
9. Run ICA

EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started
Reject continuous data

Equivalent
Reject continuous data

Click and drag with mouse over noisy data to reject
Rejecting data for ICA

To prepare data for ICA:
- Keep stereotyped artifacts (like eye blinks)
- Reject large muscle or otherwise strange events...

EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started
Pre-processing pipeline

1. Collect high-density EEG data (>30 chan)
2. Import into EEGLAB
3. Import event markers and channel locations
4. Re-reference/down-sample (if necessary)
5. High pass filter (~.5 – 1 Hz)
6. Examine raw data
7. Reject bad channels
8. Reject large artifact time points
9. Run ICA

EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started
Independent Component Analysis

\[ x = \text{scalp EEG} \]

\[ W = \text{unmixing matrix} \]

\[ W^* x = u \]

\[ u = \text{sources} \]

\[ x = W^{-1} * u \]

\[ u = \text{sources} \]

\[ W^{-1} (\text{scalp projections}) \]
“Secrets” to a good ICA decomposition

➤ Garbage in... garbage out (it's not magic)

➤ Remove large, non-stereotyped artifacts

➤ Do you have enough data? (based mostly on time, not frames)
  * ~30 min of data for 60-70 channels, ~60 min for > 200 channels

➤ High-pass filter to remove slow drifts
  * low-pass/ notch filters usually unnecessary

➤ Remove bad channels

➤ Data must be in double precision (not single)
## Runica options

<table>
<thead>
<tr>
<th>Option</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘extended’</td>
<td>0</td>
<td>1 is recommended to find sub-gaussians</td>
</tr>
<tr>
<td>‘stop’</td>
<td>1e-7</td>
<td>final weight change → stop</td>
</tr>
<tr>
<td>‘lrate’</td>
<td>determined from data</td>
<td>too small → too long… too large → wts blow up</td>
</tr>
<tr>
<td>‘maxsteps’</td>
<td>512</td>
<td>more channels → more steps</td>
</tr>
<tr>
<td>‘pca’</td>
<td>0 or EEG.nbchan</td>
<td>Decompose only a principal data subspace</td>
</tr>
</tbody>
</table>

Other algorithms: binica, sobi, acsobiro
EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started

Runica progress...

Input data size [33,133175] = 33 channels, 133175 frames/nFinding 33 ICA components using extended ICA. Kurtosis will be calculated initially every 1 blocks using 6000 data points. Decomposing 122 frames per ICA weight \((1089)^2 = 133175\) weights. Initial learning rate will be 0.001, block size Learning rate will be multiplied by 0.98 whenever angledelta \(\geq 60\) deg. More than 32 channels: default stopping weight change 1E-7 Training will end when wchange < 1e-07 or after 512 steps. Online bias adjustment will be used, Removing mean of each channel ...

Final training data range: -171,806 to 179,094
Computing the spherering matrix...
Starting weights are the identity matrix ...
Sphering the data ...
Beginning ICA training ... first training step may be slow ...
step 1 - lrate 0.0001000, wchange 16.85061324, angledelta 0.0 deg
step 2 - lrate 0.0010000, wchange 0.26760405, angledelta 0.0 deg
step 3 - lrate 0.0010000, wchange 0.79058323, angledelta 104.0 deg
step 4 - lrate 0.0005800, wchange 0.66700031, angledelta 147.2 deg
step 5 - lrate 0.000960, wchange 0.62849071, angledelta 145.6 deg
step 6 - lrate 0.000941, wchange 0.73967955, angledelta 150.7 deg
step 7 - lrate 0.000922, wchange 0.73727229, angledelta 151.6 deg
step 8 - lrate 0.000904, wchange 0.74051387, angledelta 137.9 deg
step 9 - lrate 0.000886, wchange 0.74536137, angledelta 156.0 deg
step 10 - lrate 0.000868, wchange 0.72101402, angledelta 143.7 deg
step 11 - lrate 0.000851, wchange 0.14630114, angledelta 102.5 deg
step 12 - lrate 0.000834, wchange 0.11822100, angledelta 114.3 deg
step 13 - lrate 0.000817, wchange 0.75552986, angledelta 100.6 deg
step 14 - lrate 0.000801, wchange 0.26737950, angledelta 109.1 deg
step 15 - lrate 0.000785, wchange 0.12123251, angledelta 94.2 deg
step 16 - lrate 0.000769, wchange 0.10285606, angledelta 110.7 deg
step 17 - lrate 0.000754, wchange 0.09770499, angledelta 118.6 deg
step 18 - lrate 0.000739, wchange 0.09544428, angledelta 117.1 deg

Sorting components in descending order of mean projected variance ...
Permuting the activation wave forms ...

Press Button to interrupt Runica
Interrupt
ICA weights in EEG structure
Pre-processing pipeline (review)

Collect high-density EEG data (>30 chan) → Import into EEGLAB → Import event markers and channel locations

Re-reference/down-sample (if necessary) → High pass filter (~0.5 – 1 Hz) → Examine raw data

Reject bad channels → Reject large artifact time points → Run ICA
The example data: Sternberg working memory

File  .../SampleData/stern.set
Data  Continuous data (not epoched), ref’d to right mastoid
Task
between 3 and 7 letters to memorize (colored black),
between 1 and 5 letters to ignore (colored green),
8 letters presented during each trial
50% chance of probe letter being ‘in-set’

Fixation
(5 sec)

Memorize
Ignore

SOA
(1.4 sec)

Maintenance
(2-4 sec)  Probe

See 'SternbergTaskExplanation.pdf' on wiki for more task details.

Was this letter in the memorized set?

RESPONSE
Epoch on EEG.event type

Memorize letters: capital letters
Ignore letters: ‘g’ preceding capital letter (e.g., ‘gB’)
Probe letters: ‘r’ preceding capital letter (e.g., ‘rB’)

>> EEG = pop_epoch( EEG, {'B','C','D','F','G','H','J','K','L','M','N','P'...'Q' 'R' 'S' 'T' 'V' 'W'...'X' 'Y' 'Z' }, [-1 2], 'newname',...
'Sternberg Memorize letter epochs',...
'epochinfo', 'yes');
Extract epochs

Dataset info -- pop_newset()

What do you want to do with the new dataset?

- Name it: Sternberg Memorize epochs
- Save it as file:

Epoch baseline removal -- pop_rmbase()

Baseline latency range (min_ms max_ms) (1 whole epoch):

-1000

Else, baseline points vector (ex: 1:56)
(overwrite by latency range above)

EEGLAB v10.2.4.4b

#2: Sternberg Memorize epochs

Filename: none
Channels per frame: 71
Frames per epoch: 750
Epochs: 500
Events: 1000
Sampling rate (Hz): 250
Epoch start (sec): -1.000
Epoch end (sec): 1.996
Reference: unknown
Channel locations: No
ICA weights: Yes
Dataset size (Mb): 433.1
Exercise

- **ALL**
  - Load stern.set (continuous data)
  - Do not save your changes under the same filename!

- **Novice**
  - Scroll channel data and explore plotting options under 'Settings'.
  - Reject noisy time points by visual inspection
  - Import standard channel locations
  - Practice preprocessing steps described in this lecture

- **Intermediate / Advanced** (requires supplementary material)
  - Remove a channel and then replace it by interpolation
    - Compare this signal with the original when you do this with a 'clean' channel
  - Epoch data even of interest, plot Channel ERPs from Plot menu
  - Try different filter methods and cut-offs, compare results
Supplementary lessons
Auto-detection of noisy channels

\[ \text{EEG} = \text{pop_rejchan}(\text{EEG}, \ '\text{elec}', [1:71], \ '\text{threshold}', 5, \ldots \ '\text{norm}', \ '\text{on}', \ '\text{measure}', \ '\text{prob}') \]
Auto-detected noisy channel
Interpolate bad channel

Choose a channel from other dataset

Auto-select deleted channel from other dataset
Interpolated channel

Channel order changes, but scalp location is correct
Merge (append) datasets

- Select data
- Append datasets
- Delete dataset(s)

Dataset indices to merge:
1
2

Preserve ICA weights of the first dataset? Yes

ICA weights: Yes
Dataset size (Mb): 67.6
Merged datasets

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filename</td>
<td>none</td>
</tr>
<tr>
<td>Channels per frame</td>
<td>71</td>
</tr>
<tr>
<td>Frames per epoch</td>
<td>375</td>
</tr>
<tr>
<td>Epochs</td>
<td>900</td>
</tr>
<tr>
<td>Events</td>
<td>900</td>
</tr>
<tr>
<td>Sampling rate (Hz)</td>
<td>250</td>
</tr>
<tr>
<td>Epoch start (sec)</td>
<td>-0.500</td>
</tr>
<tr>
<td>Epoch end (sec)</td>
<td>0.996</td>
</tr>
<tr>
<td>Reference</td>
<td>unknown</td>
</tr>
<tr>
<td>Channel locations</td>
<td>Yes</td>
</tr>
<tr>
<td>ICA weights</td>
<td>Yes</td>
</tr>
<tr>
<td>Dataset size (Mb)</td>
<td>199.9</td>
</tr>
</tbody>
</table>
1) input original 'type' code
2) input new 'type' code
3) Keep/delete all other events
Renaming events

EEGLAB Workshop, June 16-18, 2012, Beijing, China: Julie Onton – Getting started
Analysis of channel ERPs

```matlab
>> pop_timtopo(EEG, [-200 500], NaN, 'ERP data and scalp maps');
```
Analysis of channel ERPs
Channel ERP in rectangular array
Analysis of channel ERPs

```
pop_topoplot(EEG,1,[0:25:275],'Memorize',[3 4],0,'electrodes','off');
```
Compare ERPs across conditions

How do 'Memorize' and 'Ignore' ERPs differ?
Compare ERPs across conditions

>> pop_comperp(ALLEEG,1,[2 3],[],'addavg','off','addstd','off',...  
  'addall','on','diffavg','off','diffstd','off','lowpass',20, ... 
  'tplotopty','ydir',1));
Compare ERPs across conditions

Click on an axis to see larger image
Analysis of ERP differences

Plot difference between two conditions

>> `pop_comperp(ALLEEG,1, 2, 3,'addavg','off',...'addstd','off', 'diffavg','on','diffstd','off', ...'lowpass',20, 'tplotopt',{'ydir',1});`
Analysis of ERP differences

ERP difference between 2 conditions
Event durations

Color denotes event duration
Comments in EEGLAB structure

>> EEG.comments

About this dataset

Data acquired by: Julie Onton
Data acquired on: Oct 15, 1974

Task ('eventname'):
- Every trial preceded by 5 sec fixation ('fix') - 3,5,7 letters to memorize (in black, ex, 'B')
- 5,3,1 letters to ignore (in green, ex, 'gB')
- 8-letters total per trial
- 2-4 sec variable maintenance period ('WM')
- Probe letter (in red, ex, 'rB') prompted a button press response
- Subject was to indicate by left or right mouse button (with right hand) whether the probe letter was in-set (left,'in') or out-of-set (right,'out')
- If the response was correct, subject heard one tone ('correct'), if incorrect, a different tone ('wrong')

Reference electrode placed on right mastoid

(Subject indicated sleepiness toward the end of session)
Memory options

Set when loading a STUDY