Evaluating ICA Components

EEGLAB Workshop XXV
JAIST, Tokyo, Japan
Day 1
Independent Component Analysis

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

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

\[ u = \text{sources} \]

\[ W^{-1} (\text{scalp projections}) \]

\[ W^* x = u \]

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

ICA is a method to recover a version of the original sources by multiplying the data by an unmixing matrix,

While PCA simply decorrelates the outputs (using an orthogonal mixing matrix), ICA attempts to make the outputs *statistically independent*, while placing no constraints on the mixing matrix.
Finally: ICA 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 too small → too long… from data too large → wts blow up</td>
<td></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, amica, cudaica, beamica
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 ((1083)^2 = 133175 weights, Initial learning rate will be 0.001, block size
Learning rate will be multiplied by 0.98 whenever angledelta >= 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 ...
Spherering the data ...
Beginning ICA training ... first training step may be slow ...
step 1 - Irate 0.001000, wchange 16.85061324, angledelta 0.0 deg
step 2 - Irate 0.001000, wchange 0.26760405, angledelta 0.0 deg
step 3 - Irate 0.001000, wchange 0.79058323, angledelta 104.0 deg
step 4 - Irate 0.000888, wchange 0.667000031, angledelta 147.2 deg
step 5 - Irate 0.000841, wchange 0.73967355, angledelta 145.5 deg
step 6 - Irate 0.000822, wchange 0.73727229, angledelta 151.6 deg
step 7 - Irate 0.000904, wchange 0.74051387, angledelta 137.9 deg
step 8 - Irate 0.000886, wchange 0.74536137, angledelta 156.0 deg
step 9 - Irate 0.000868, wchange 0.72101402, angledelta 143.7 deg
step 10 - Irate 0.000851, wchange 0.14690114, angledelta 102.5 deg
step 11 - Irate 0.000834, wchange 0.11822100, angledelta 114.3 deg
step 12 - Irate 0.000817, wchange 0.75552966, angledelta 100.6 deg
step 13 - Irate 0.000801, wchange 0.26739750, angledelta 109.1 deg
step 14 - Irate 0.000785, wchange 0.12132351, angledelta 94.2 deg
step 15 - Irate 0.000769, wchange 0.10285606, angledelta 110.7 deg
step 16 - Irate 0.000754, wchange 0.09770439, angledelta 118.6 deg
step 17 - Irate 0.000739, wchange 0.09544428, angledelta 117.1 deg

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

EEGLAB Workshop XXV, Sep 26-29, 2017, Tokyo, Japan – John Iversen – Evaluating IC Components
Alternatives to runica

Infomax ICA

runica  matlab implementation
binica  compiled version; fast
cudaica  GPU version


AMICA

Best at extracting dipolar ICs
Multiple-model support
Source activation = unmixing * Channel data

Channel data = mixing (topo) * Source activation
Results of ICA Decomposition in EEG struct
Source activation = unmixing * Channel data

Channel data = mixing (topo) * Source activation

EEG.icaact = (EEG.icaweights*EEG.icasphere) * EEG.data

EEG.data = EEG.icawinv * EEG.icaact
Now what…?

Part 1
Getting an overview of your ICs

Part 2
Classifying/Evaluating ICs

Part 3
Detailed look at IC properties
  ERP
  Spectrum
  ERP images
  ERSP
IC Evaluation Practicum (Day 1)

- ICA Component Classifier Competition

- Traditional Practicum using faces_4.set
Now what…?

Part 1
Getting an overview of your ICs

Part 2
Classifying/Evaluating ICs

Part 3
Detailed look at IC properties
  ERP
  Spectrum
  ERP images
  ERSP
A convenient ‘trick’…

Use ‘Reject components by map’ to survey components

NB: A new plugin "viewprops" is available!
An interactive overview of ICs
Step 0: Quality of Decomposition

BAD ICA Components

ICA Components
Examining IC Properties
IC Properties

IC Topography
topoplot()

ERP Image & ERP
erpimage()

Power Spectrum
spectopo()
Many plot panels in EEGLAB will expand when clicked on!
Part 1
Getting an overview of your ICs

Part 2
Classifying/Evaluating ICs
- Eye Artifacts
- Muscle Artifacts
- Other Artifacts
- Brain ICs

Part 3
Detailed look at IC properties
- ERP
- Spectrum
- ERP images
- ERSP
Evaluating ICs

Over time, most EEGLAB users develop a heuristic sense of which ICs might be brain vs. artifact.

Heuristics are generally based on:

- Topography
- Component Activities (scroll)
- ERP
- Power Spectrum

IC Classification can be used to ‘clean’ data—study likely brain activity without artifacts

*There are new efforts to automate this process, but doing it by hand is a good place to start to build intuition*
Topography
IC 4 – eyeblink

Classic frontal eye-blink topography

Sporadic large biphasic pulses

That may yet be task related!

But: highly variable (erpalpha=0.01)
Plot → Component Activations (scroll)
IC 4 Activation – eyeblink

Sporadic large biphasic pulses
IC 9 – lateral eye movement

Classic frontal eye-movement topography
IC 9 Activation – lateral eye movement
IC 12, 18 – Muscle

Narrowly spaced dipolar topography (consistent with superficial source)

Noisy ERP/ERP Image

High frequencies dominate power spectrum
IC 12, 18 Activation – Muscle
IC 17, 25 – Bad channels

Punctate topography (single channel)

Sporadic epoch activity (sometimes just a single large spike)
IC 2, 7 – Cardiac

Unusual, peaky spectrum (often peaks ~5, 10 Hz)

Periodic spikes (~1 / sec)

Cardiac-like topographies:
Shallow gradient = extremely distant source
Artifacts

<table>
<thead>
<tr>
<th>Eye Muscle Cardiac Badchan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>29</td>
</tr>
</tbody>
</table>
Brain ICs

- Classic occipital topography
- Strongly task-related ERP Image & ERP
- 10 Hz Alpha peak
Dipole orientation matters
Brain ICs

Classic radial-dipole source topography

Strongly task-related ERP Image & ERP
Brain ICs

Classic tangential-dipole source topography

two peaks, not as closely spaced as muscle: deeper

Task-related ERP Image & ERP
IC Classification…so far

Eye
Muscle
Cardiac
Badchan
Brain
Now what…?

Part 1
Getting an overview of your ICs

Part 2
Classifying/Evaluating ICs

Part 3
Detailed look at IC properties
ERP
Spectrum
ERP images
ERSP
Component ERPs

Figure 2: Plot > Component ERPs > In rect. array -- plotdata()

Sternberg: Memorize epochs ERP

Component ERPs in rect. array -- pop_plotdata()
A step back: Electrode-level ERP

EEGLAB v13.1.1

Channel locations
- Channel data (scroll)
- Channel spectra and maps
- Channel properties
- Channel ERP image

Channel ERPs
- ERP map series
- Sum/Compare ERPs
- Component activations (scroll)
- Component spectra and maps
- Component maps
- Component properties
- Component ERP image
- Component ERPs
- Sum/Compare comp. ERPs

Data statistics
Time-frequency transforms

With scalp maps
In scalp/rect. array

Plotting time range (ms): [-500 1500]
Scalp map latencies (ms, NaN -> max-RMS): [112 380]
Plot title: ERP data and scalp maps
Scalp map options (see >> help topoplot): ['electrodes', 'off']
Traditional ERP: Time-locked activity at each channel

Topography at latency of two peaks

ERP at each channel

EEG peaks are sometimes called “components.” Not to be confused with ICAs Independent Components (ICs)!
ERP at two channels
Definition: The data envelope

Data (all channels)

Data envelope (max/min traces)
Definition: IC Envelope

IC 10 Topography

Data Envelope

IC Envelope

IC 10 Activation ERP back-projected to channels
Key: Scalp ERP peaks are often the sum of multiple independent source processes.

PVAF
“% Variance Accounted For”
The variance of scalp EEG accounted for by this component

*ppaf

This component accounts for all of the negative scalp ERP peak at ~150 ms, but only some of other ERP peaks

Net PVAF: 37%
Component ERP envelope

[Image of EEGLAB software interface showing menu options for analyzing ERP envelopes]
ERP peak- and IC Component-topographies
Component 3 ERP envelope

Channel ERP
Component 1 ERP envelope

Note: IC Envelope can exceed the data envelope: Other component(s) have opposite sign at this latency.
Component 1 + 3 ERP envelope

Else plot these component numbers only (Ex: 2:4,7): 1, 3
pvaftopo plugin (Makoto Miyakoshi)

Note: However, PVAF is calculated over entire signal duration. PVAF at times of peaks often higher, but still typically not 100%.

Max PVAF ~ 70%

Max PVAF ~ 40%
Top 6 IC contributions to data ERP envelope

Figure 3

Largest ERP components of faces_4 face epochs

Potential (uV)

Time (s)

ppaf 94.48%
Non-artifact IC contrib. to data ERP envelope

Else plot these component numbers only (Ex: 2, 4, 7).

Component numbers to remove from data before plotting:

Plot title:
Non-artifact IC contrib. to data ERP envelope

Largest non-artifact ERP components of faces_4 face epochs

```
pop_envtopo(EEG, [-500 1500] , 'limcontrib', [0 1500], ... 'compsplot',[6],'subcomps',[2 4 5 7 9 12 17 18 25], ... 'title', 'Largest non-artifact ERP components of faces_4 face epochs',..., 'electrodes','off');
```
Compare: Effect of removing artifacts
IC ERP difference

What is the IC ERP difference between these 2 conditions?

(Data: stern_125Hz.set)
IC ERP difference
IC ERP difference
Now what…?

Part 1
Getting an overview of your ICs

Part 2
Classifying/Evaluating ICs

Part 3
Detailed look at IC properties
  ERP
  Spectrum (see next lecture)
  ERP images
  ERSP
Plot component power spectrum

By default, plots topographies for frequency of largest peak.
Select the frequency for topographies

Component spectra and maps -- pop_spectopo()

Epoch time range to analyze [min_ms max_ms]:
Frequency (Hz) to analyze:
Electrode number to analyze (1=elec with max power; 0=whole scalp):
Percent data to sample (1 to 100):
Components to include in the analysis:
Number of largest-contributing components to map:
Else, map only these component numbers:
[Checked] Compute comp spectra; [Unchecked] (data-comp) spectra:
Plotting frequency range (min-max) Hz:
Spectral and scalp map options (see topplot):

Figure 2: spectopo()
Now what…?

Part 1
Getting an overview of your ICs

Part 2
Classifying/Evaluating ICs

Part 3
Detailed look at IC properties
   ERP
   Spectrum
   ERP images
   ERSP
Component ERP image
ERP Image basics

by default, sorted by time-on-task (1\textsuperscript{st} trial, 2\textsuperscript{nd} trial, ...)
ERP Image basics

Trial 1:
Trial 2:

No Smoothing
Smoothed across 10 Trials
ERP Images: smoothing across trials
Evoked response in ERP Image follows the time of the button press.

This can be obscured in the ERP because reaction times are smeared in time.
Component ERP Images: Sort by phase

Phase-sorted image

Alpha phase is initially random

After the stimulus, the phase is reset.

This phase alignment yields the ERP!
Component ERP Images: ITC

**Phase-sorted alpha power**

**ITC**

**ERSP**

- Component(s): 3
- Project to channel #: 
- Smoothing: 10
- Downsampling: 1
- Time limits (ms): -800 1000

**Sort/align trials by epoch event values**
- Epoch-sorting field 
- Event type(s) 
- Event type(s) 

**Sort trials by phase**
- Frequency (Hz minHz maxHz): 10 12
- Percent low-amp. trials: 
- Inter-trial coherence options
  - Frequency (Hz minHz maxHz): 10 12
  - Signif. level (<0.2): .01

**Other options**
- Plot spectrum (minHz maxHz): 
- Baseline ampl. (dB): 

**Figure title**
- ERP limits
- Color limits (see Help)

**Figure 3: erpimage()**

- File Edit View Insert Tools Desktop Window Help

- Phase-sorted Trials
- Comp. 3

- ITC
- ERSP
Component ERP Images: Sort by amplitude

Same data: Sorted by alpha amplitude

Phase-sorted alpha power

'mpsort' = [center_ms, prcnt, freq, maxfreq] Sort epochs by amplitude.

>> help erpimage

'ampsort' = [center_ms, prcnt, freq, maxfreq] Sort epochs by amplitude.
Component ERP Images: Amplitude vs. Activations

Same sorting order: Plotting Amplitude vs. Activations
Now what…?

Part 1
Getting an overview of your ICs

Part 2
Classifying/Evaluating ICs

Part 3
Detailed look at IC properties
   ERP
   Spectrum
   ERP images
   ERSP (see next lecture)
Definition: ERSP

Event Related Spectral Perturbation

Change in power in different frequency bands relative to a baseline. ERS, ERD
Plot IC ERSP

Time points removed at beginning and end of epoch to avoid edge effects of wavelet transform.
Plot IC ERSP

The image shows a plot of event-related spectral power (ERSP) for a specific component. The window displays parameters such as component number, sub epoch time limits, frequency limits, baseline limits, and wavelet cycles. The plot includes time-frequency representations with color-coded ERSP and ITC values.
Further Resources

Some attempts to automate the IC classification:

“Automatic Classification of Artifactual ICA-Components for Artifact Removal in EEG Signals”
Irene Winkler, Stefan Haufe and Michael Tangermann (2011)
http://www.behavioralandbrainfunctions.com/content/7/1/30

Bigdely-Shamlo’s EyeCatch (2013)
https://www.researchgate.net/publication/257602145_EyeCatch_Data-mining_over_half_a_million_EEG_independent_components_to_construct_a_fully-automated_eye-component_detector

Luca Pion-Tonachini (ongoing)
Crowd-sourcing heuristic knowledge about IC components to build automatic classifier
We’ll play the game later: http://reaching.ucsd.edu:8000
IC Evaluation Practicum (Day 1)

• ICA Component Classifier Competition

• Traditional Practicum using faces_4.set
IC Classification…so far

Eye
Muscle
Cardiac
Badchan
Brain
What is EEGLAB?

EEGLAB is an interactive Matlab toolbox for processing continuous and event-related EEG, MEG and other electrophysiological data incorporating independent component analysis (ICA), time/frequency analysis, artifact rejection, event-related statistics, and several useful modes of visualization of the averaged and single-trial data. EEGLAB runs under Linux, Unix, Windows, and Mac OS X.

EEGLAB provides an interactive graphic user interface (GUI) allowing users to flexibly and interactively process their high-density EEG and other dynamic brain data using independent component analysis (ICA) and/or time/frequency analysis (TFA), as well as standard averaging methods. EEGLAB also incorporates extensive tutorial and help windows, plus a command history function that eases users' transition from GUI-based data exploration to building and running batch or custom data analysis scripts. EEGLAB offers a wealth of methods for visualizing and modeling event-related brain dynamics, both at the level of individual EEGLAB 'datasets' and/or across a collection of datasets brought together in an EEGLAB 'studyset.'

For experienced Matlab users, EEGLAB offers a structured programming environment for storing, accessing, measuring, manipulating and visualizing event-related EEG data. For creative research programmers and methods developers, EEGLAB offers an extensible, open-source platform through which they can share new methods with the world research community by publishing EEGLAB 'plug-in' functions that appear automatically in the EEGLAB menu of users who download them. For example, novel EEGLAB plug-ins might be built and released to 'pick peaks' in ERP or time/frequency results, or to perform specialized import/export, data visualization, or inverse source modeling of EEG, MEG, and/or ECOG data.

EEGLAB Features

- Graphic user interface
- Multiformat data importing
- High-density data scrolling
- Interactive plotting functions
- Semi-automated artifact removal
- ICA & time/frequency transforms
- Event & channel location handling
- Forward/inverse head/source modeling
- Defined EEG data structure
- Many advanced plug-in/extension toolboxes

Q: I’ve used ICA to decompose my data – now how can I learn to recognize which independent component processes of EEG data represent brain sources activity and which capture activity from other non-brain sources?

Try using iCLabel to first learn about and then practice labelling EEG independent components (ICs). You can also help to create a more accurate automated IC classifier using machine learning algorithms on crowd-sourced data. The results are being incorporated into a self-updating EEGLAB plug-in that will become more accurate the more labels you and others contribute.
Automating IC Identification

Luca Pion-Tonachini (lpionton@ucsd.edu)

**Goal:** Create an automated, high confidence EEG component labeler.

**Motivation:** Typically we rely on expert knowledge to pick which components to work with, but can be very time consuming with large datasets or inconvenient / infeasible when automation is the goal (BCI).

**Plan:**
1. Aggregate Data
2. Gather Labels
3. Process Labels
4. Train Classifiers
   - Real-time and offline versions
Tutorial: EEG Independent Component Labeling

Overview

We would like you to help us label independent components from EEG datasets to create an automated classifier. For more information, see Why Help Us?

Steps to doing so

1. Register or Log In.
2. Look at the image presented. For help reading it, see How To Label. It is essential that you go over the instructions before you start.
3. For each component presented, try to decide what type of component you are looking at. To learn how to do this, see Telling Components Apart and perform Practice Labeling.
4. Click the appropriate button or buttons to label the component. For help with our categories, first read How To Label.
5. Then click on "Next" to view a new component.

That's it! Please read the text in all the links above and perform some Practice Labels. Then click Begin Labeling.

If you have any suggestions, please Leave A Comment. Also – we have a Leaderboard!
Practice First…

Practice

Example component images are provided below. Click on all the labels your feel follows:

- White: no labels
- Grey: correct but insufficient labels
- Green: all labels correct
- Red: one or more labels are incorrect

Marking "?" is ignored here as that category is user dependent.
Practicum

- **ALL**
  - Download then load faces_4.set, epoch on face

- **Novice, Intermediate**
  - From the GUI, open the ‘Reject component by map’ interface
  - Explore and classify several additional ICs: muscle, channel, brain
    ~ Justify your classification
  - Redo the "Plot → Component ERPs → With component maps" excluding your additional artifacts. What change do you observe?
  - Pick a brain IC. Plot an ERP Image
    ~ Try sorting by phase, is there any relationship to the IC activation pattern? What about power in a frequency band of choice?

- **Intermediate**
  - Plot ERP Image sorted by response latency
    ~ Figure out how to realign trials to response latency instead (Hint ‘Align’)
  - Plot ERSPs for selected ICs
    ~ Explore parameter options. Why is each useful?
  - Plot component cross-coherence for pairs of ICs

- **ALL (Time permitting)**
  - Create second dataset, epoched on object
  - Examine ERP differences between the conditions using "Plot → Component ERPs → With component maps (compare)"
  - For ICs most different between conditions, compare ERP Image, ERSP
Realigning Trials: Stimulus vs. Response


Nice method for generating dual ERPs