Learning Machine Learning – Meeting 5

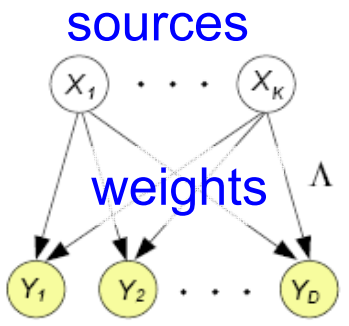
Independent Components Analysis (ICA): Theory and Application to MEG Data

Jason Taylor

MRC Cognition and Brain Sciences Unit

jason.taylor <at> mrc-cbu.cam.ac.uk

Blind Source Separation



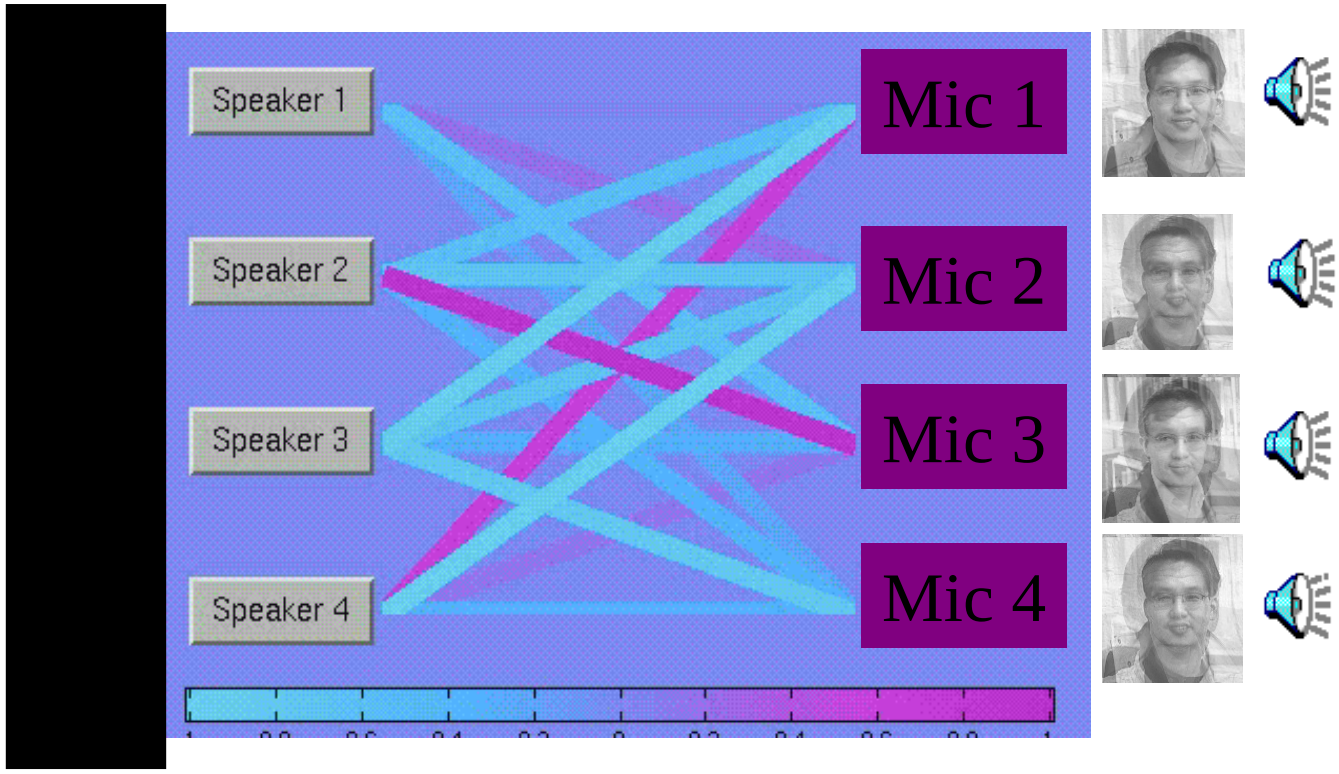
Adapted from Zoubin Ghahramani: http://learning.eng.cam.ac.uk/zoubin/ml06

Example: Audio decomposition

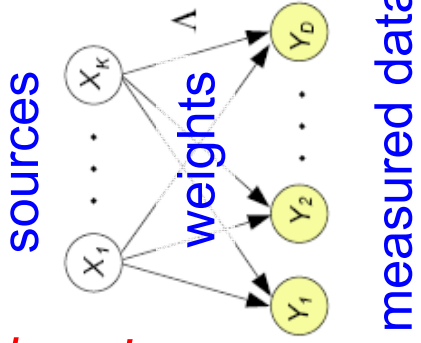
Adapted from Scott Makeig [http://sccn.ucsd.edu/~scott/icademo/index.html]

‘Source’ Speakers

Recorded Mixtures



(blind)



See Makeig’s website for a functional copy of the demo!

ICA: Logic & Assumptions

Measured data are a linear mixture of independent non-gaussian sources.

$y = \Lambda x$ (data = mixing matrix * sources)

ICA finds the 'unmixing' matrix (W) to recover sources

$x = Wy$ (sources = *un*mixing matrix * data)

$W = \Lambda^{-1}$

A mixture of non-gaussian sources tends towards gaussian (Central Limit Theorem)

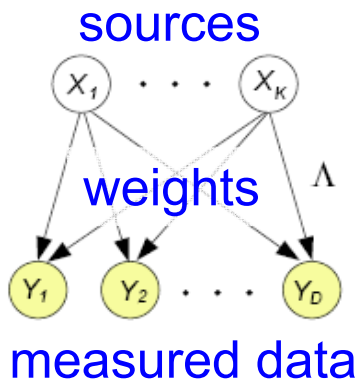
Iterative algorithm begins with random weights, computes estimate of x, adjusts weights to increase non-gaussianity of source estimates

Strategies:
- Increase (absolute) *Kurtosis*
- Increase *Entropy*
- Decrease *Mutual Information (infomax)*

$$\mathrm{kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2$$

$$H(Y) = -\sum_i P(Y = a_i) \log P(Y = a_i)$$

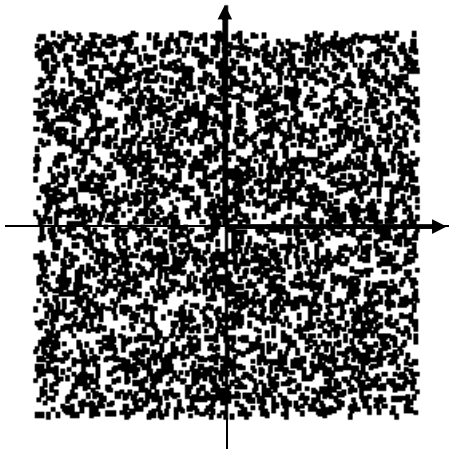
$$I(y_1, y_2, \ldots, y_m) = \sum_{i=1}^{m} H(y_i) - H(y).$$



ICA: Logic & Assumptions

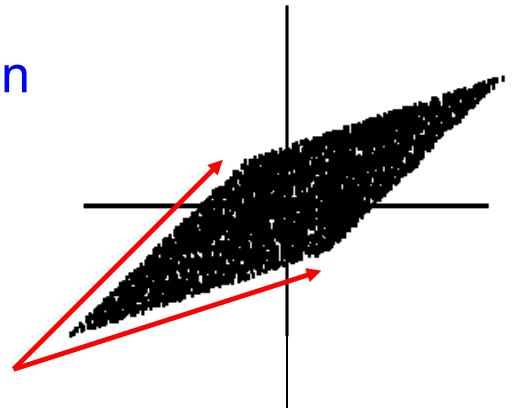
Aapo Hyvärinen:
http://www.cis.hut.fi/aapo/papers/IJCNN99_tutorialweb/IJCNN99_tutorial3.html

Two non-gaussian ‘sources’



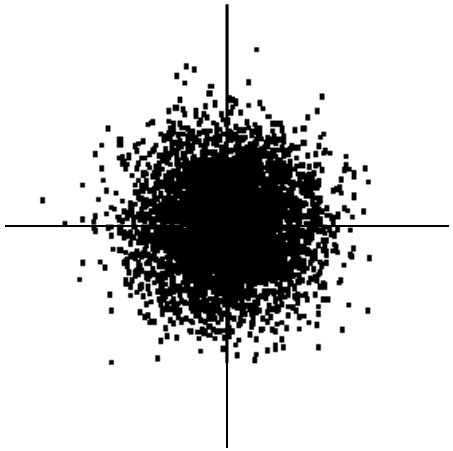
Linear transformation

$$\mathbf{A}_0 = \begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}$$

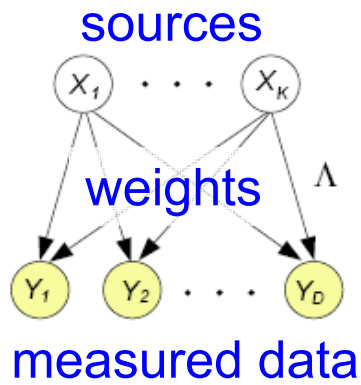


ICA

Two gaussian ‘sources’



Doesn’t work

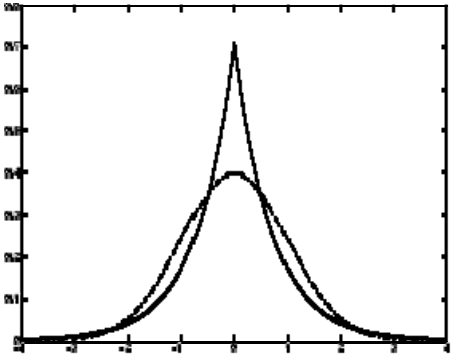


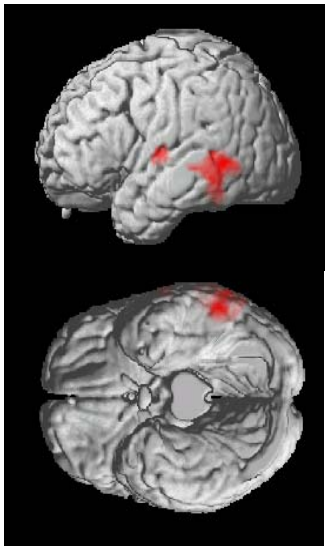
ICA: Logic & Assumptions

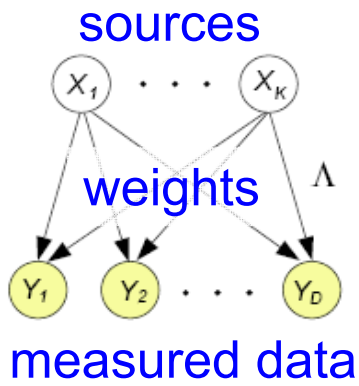
Time-varying (tICA) sources are assumed to be:

Independent (duh), non-gaussian, compact & spatially static

$$p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k)$$





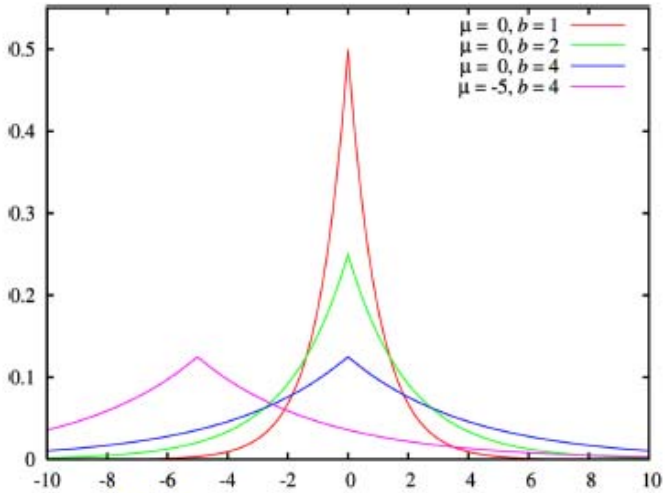


Kurtosis

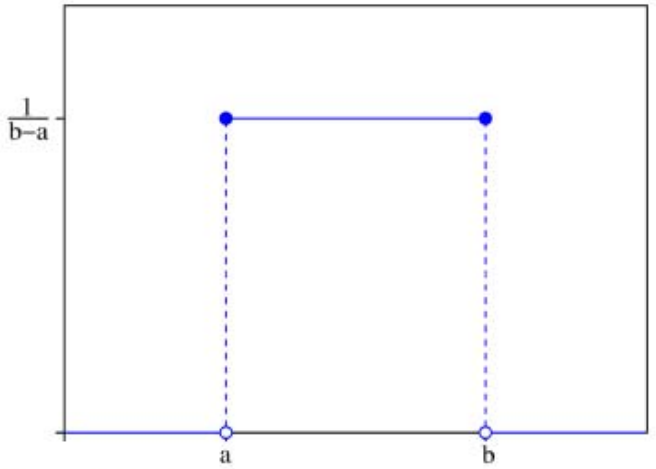
The kurtosis (or excess kurtosis) measures how "peaky" or "heavy-tailed" a distribution is.

$$K = \frac{E((x-\mu)^4)}{E((x-\mu)^2)^2} - 3$$

where $\mu = E(x)$ is the mean of x.
Gaussian distributions have zero kurtosis.



Heavy tailed distributions have positive kurtosis (leptokurtic).



Light tailed distributions have negative kurtosis (platykurtic).

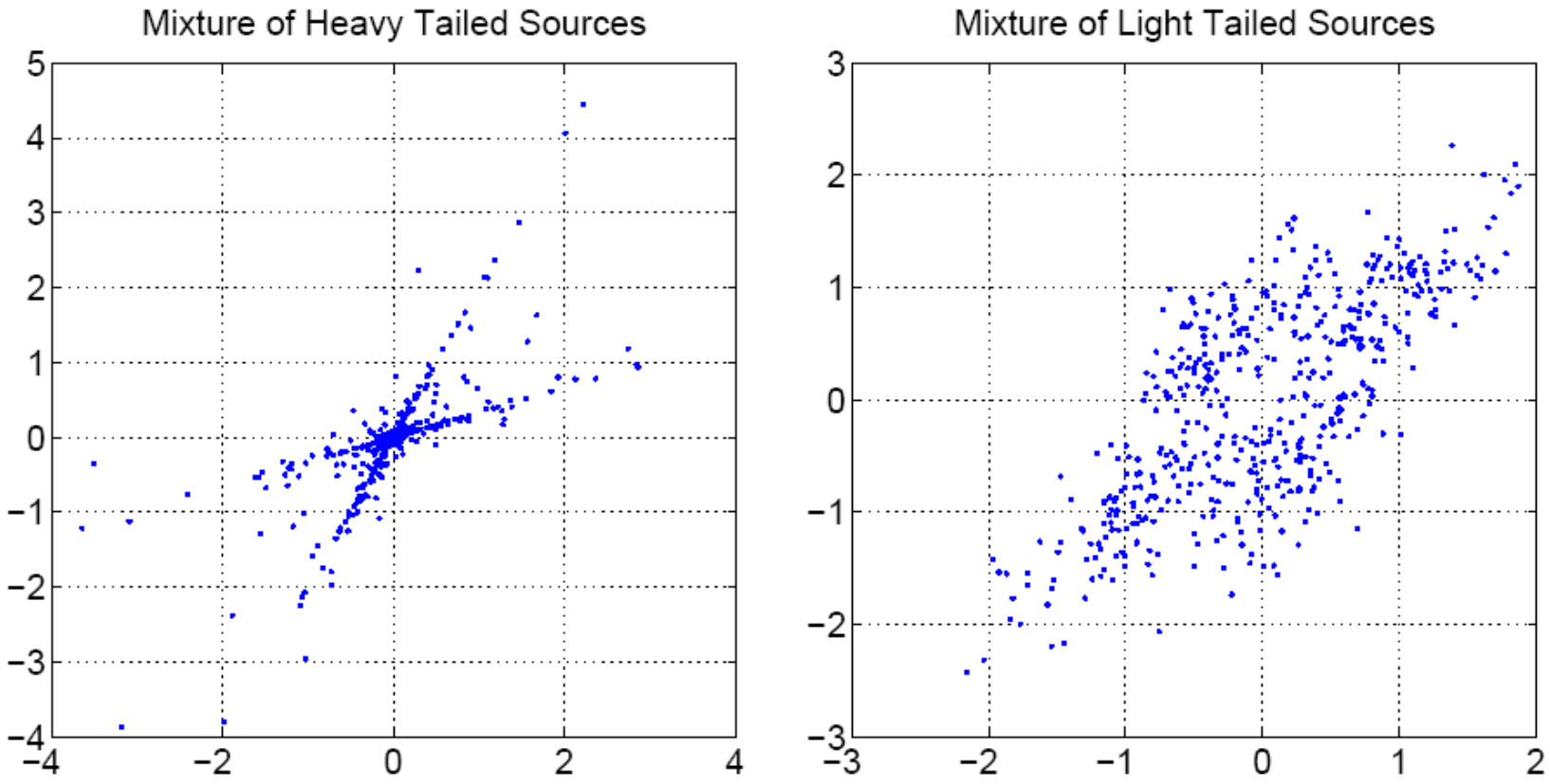
ICA models often use heavy-tailed distributions.

Why are heavy-tailed distributions interesting?

Slide from Zoubin Ghahramani: http://learning.eng.cam.ac.uk/zoubin/ml06

Generating data from an ICA model

To understand how ICA works it's useful to show data generated from it ($K = D = 2$).

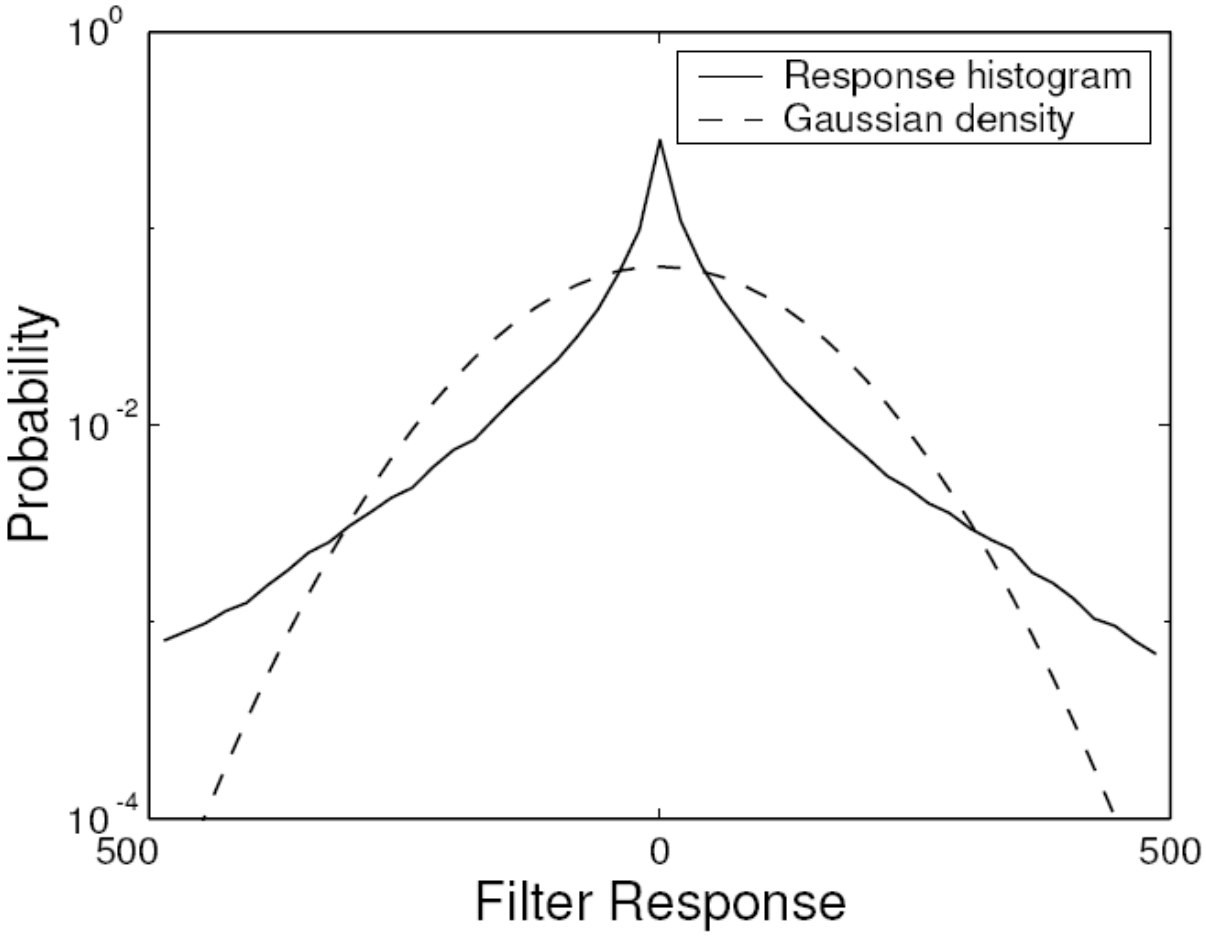


ICA (with heavy tailed noise) tries to find the directions with outliers.

Slide from Zoubin Ghahramani: http://learning.eng.cam.ac.uk/zoubin/ml06

Natural Scenes and Sounds

Experiment: take some local linear filter (e.g. Gabor wavelet) and run it on some natural sounds or images. Measure filter output.



Slide from Zoubin Ghahramani: http://learning.eng.cam.ac.uk/zoubin/ml06

Natural Scenes

Interesting fact: ICA models seem to learn representations ($\mathbf{x}$ given $\mathbf{y}$) that look very similar to responses of neurons in primary visual cortex of the brain.

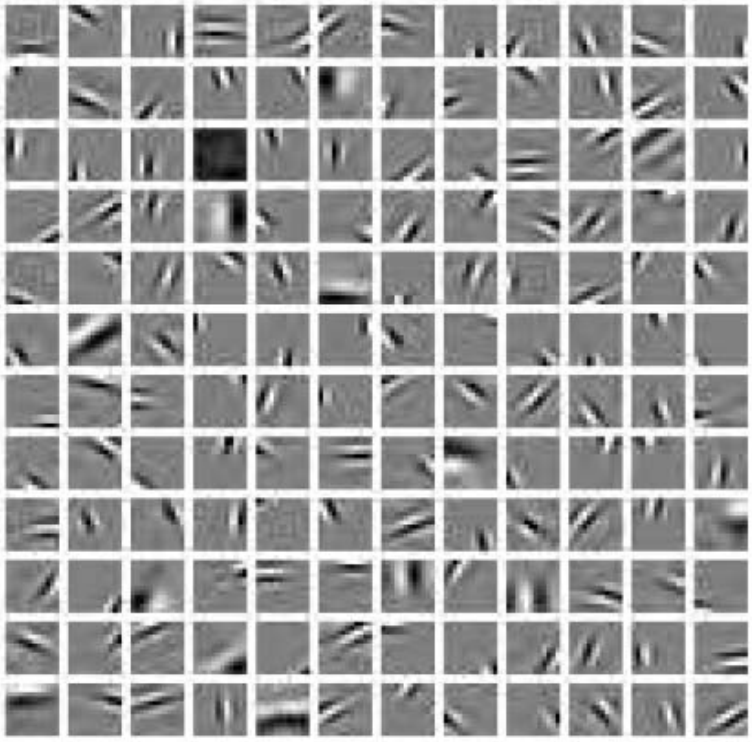


Figure 7: Example basis functions derived using sparseness criterion see (Olshausen & Field 1996).

Slide from Zoubin Ghahramani: http://learning.eng.cam.ac.uk/zoubin/ml06

ICA: Logic & Assumptions

Time-varying (tICA) sources are assumed to be:

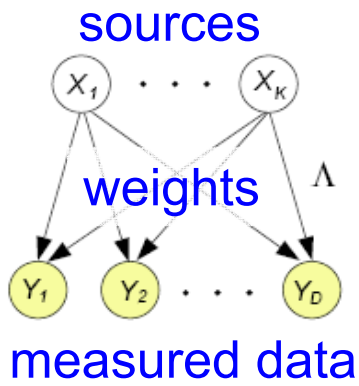
Independent (duh), non-gaussian, compact & spatially static

Limitations:

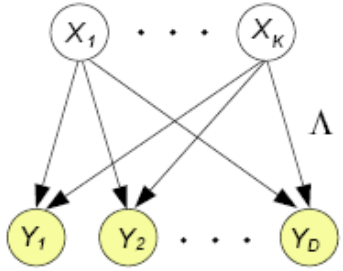
- Can't have all gaussian sources
- Can't have non-linear mixture
- Can only recover as many sources as you have observations
  (Beware: components can be wasted on modelling noise)

Quirks:

- Each component accounts for a small percentage of variance
  (typically 0-5%; unlike PCA)
- Order of output components is random
  (can order by e.g. PVA)
- Sign of output components is arbitrary
  (2 local minima, one +, one -)



Blind Source Separation



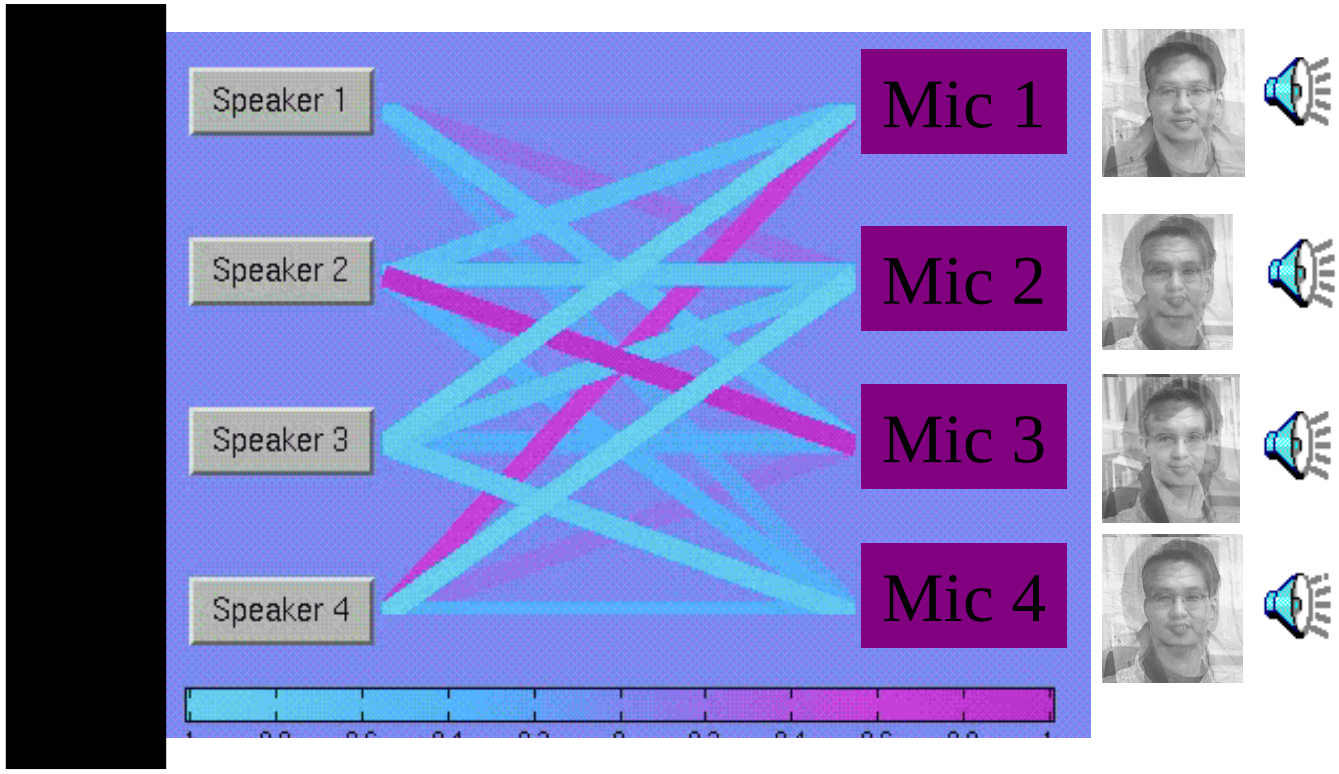
Adapted from Zoubin Ghahramani: http://learning.eng.cam.ac.uk/zoubin/ml06

Example: Audio decomposition

Adapted from Scott Makeig [http://sccn.ucsd.edu/~scott/icademo/index.html]

‘Source’ Speakers

Recorded Mixtures

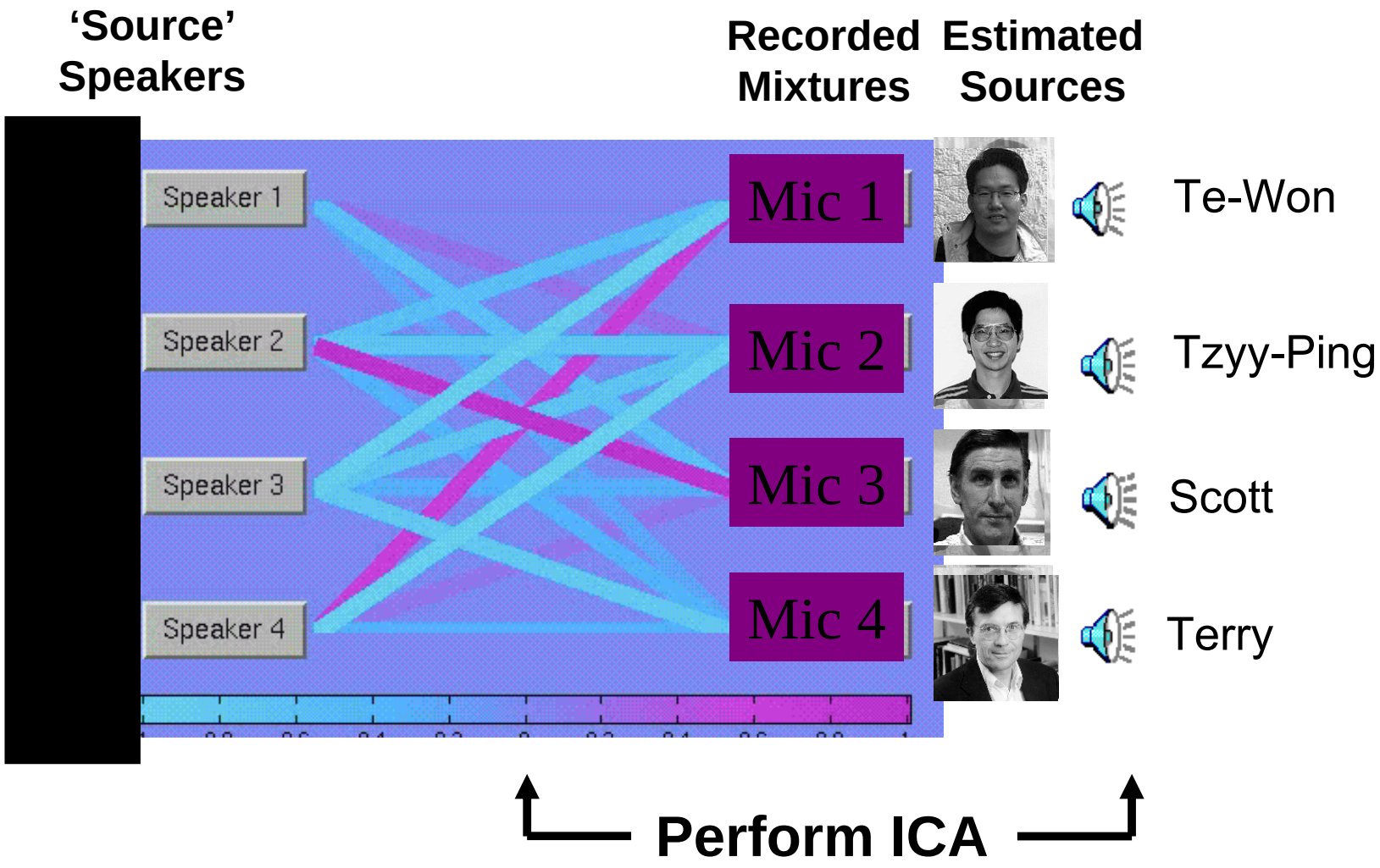


(blind)

See Makeig’s website for a functional copy of the demo!

Example: Audio decomposition

Adapted from Scott Makeig [http://sccn.ucsd.edu/~scott/icademo/index.html]



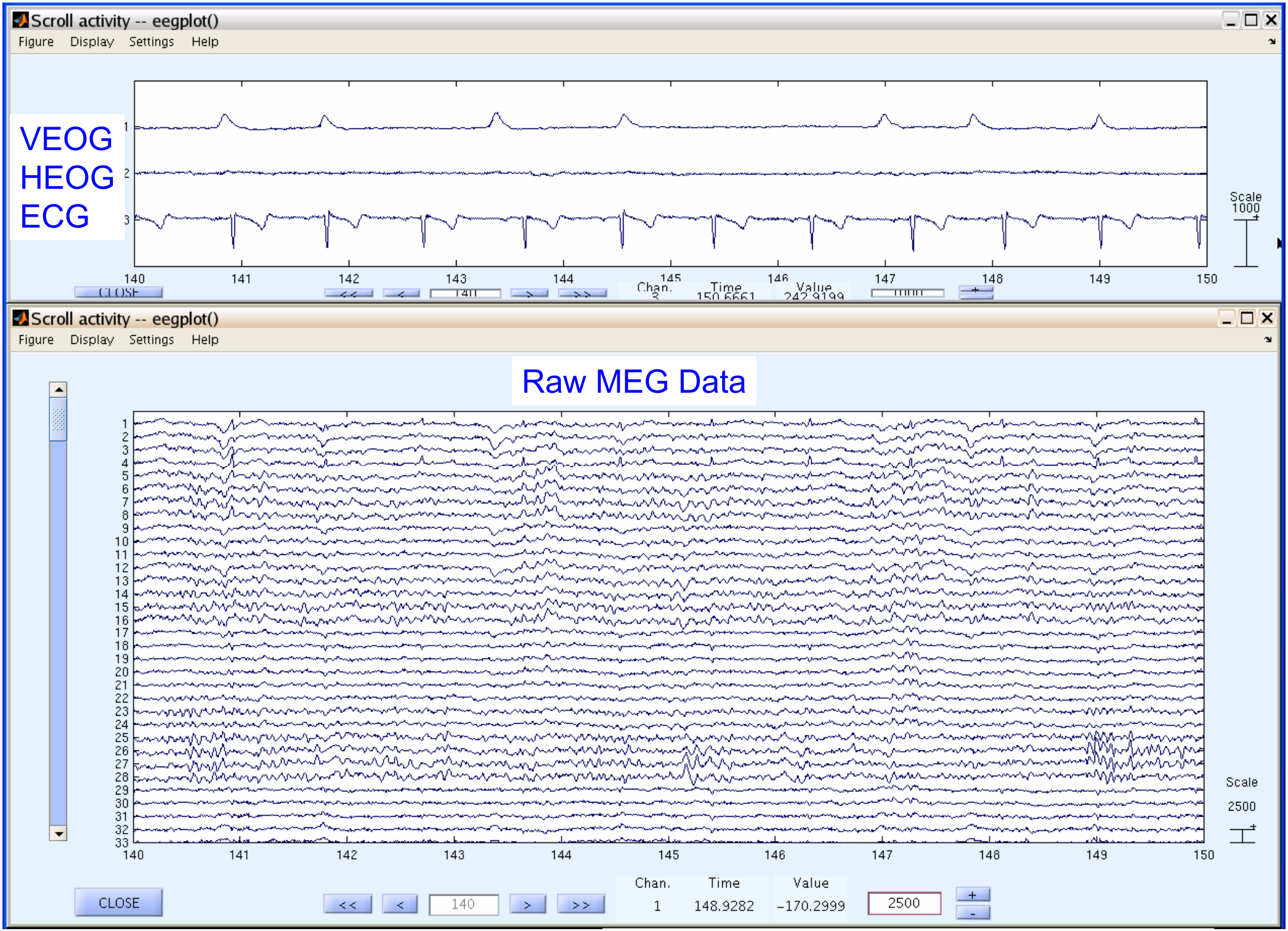
See Makeig's website for a functional copy of the demo!

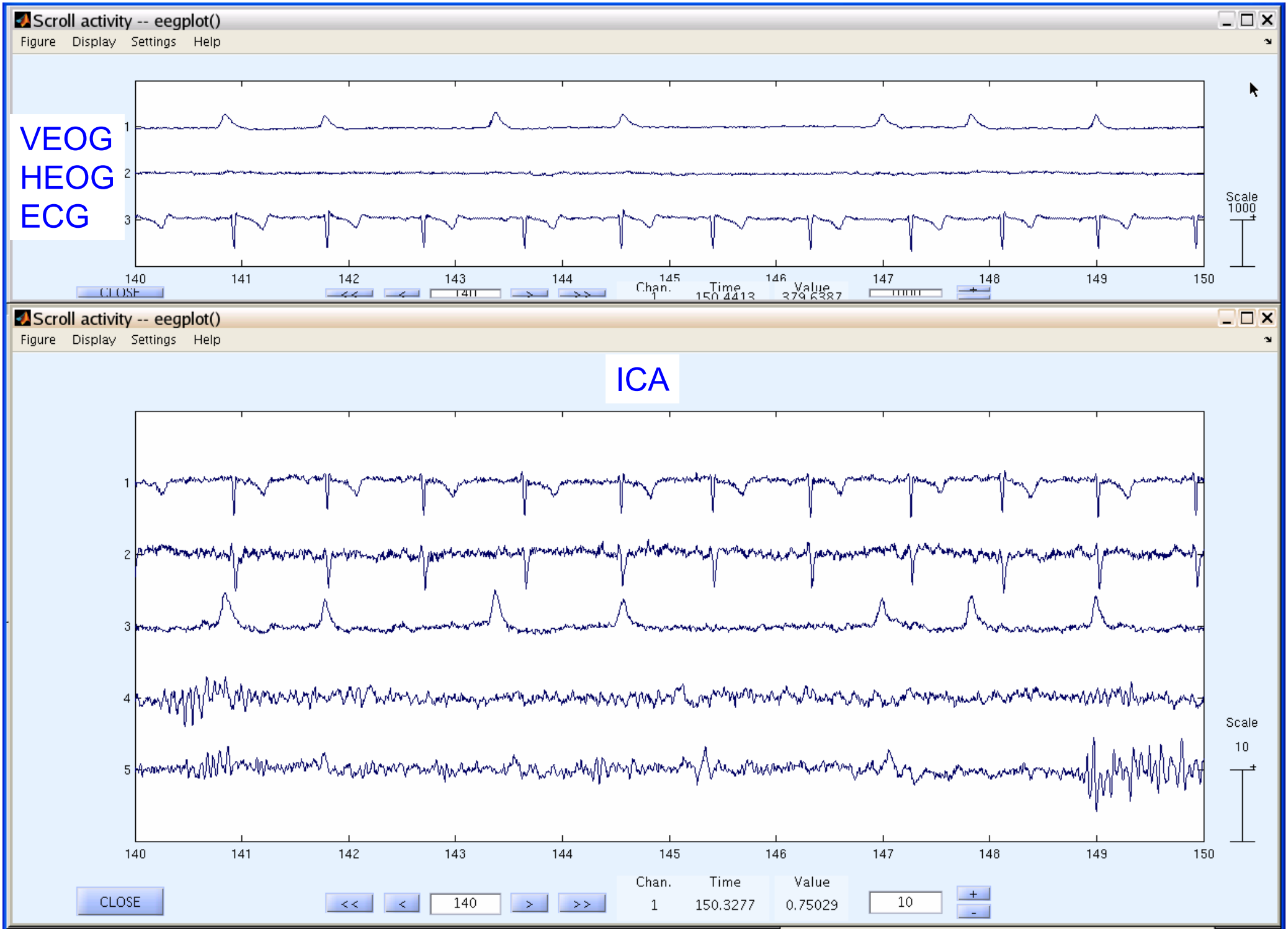
Applications of ICA and Related Methods

- Separating auditory sources
- Analysis of EEG data
- Analysis of functional MRI data
- Natural scene analysis
- ...

Slide from Zoubin Ghahramani: http://learning.eng.cam.ac.uk/zoubin/ml06

Examples from MEG data


Scroll activity -- eegplot()
Figure Display Settings Help
VEOG
HEOG
ECG
Scale
1000
140 141 142 143 144 145 146 147 148 149 150
CLOSE
Chan.
Time
Value
Scroll activity -- eegplot()
Figure Display Settings Help
Raw MEG Data
Scale
2500
CLOSE
<< < 140 > >>
Chan. Time Value
1 148.9282 -170.2999
2500


Scroll activity -- eegplot()
Figure Display Settings Help
VEOG
HEOG
ECG
1
2
3
140 141 142 143 144 145 146 147 148 149 150
Scale
1000
CLOSE
<< < 140 > >>
Chan. Time Value
1 150.4413 379.6387
Scroll activity -- eegplot()
Figure Display Settings Help
ICA
1
2
3
4
5
140 141 142 143 144 145 146 147 148 149 150
Scale
10
CLOSE
<< < 140 > >>
Chan. Time Value
1 150.3277 0.75029
10
+
-

ICA – 65 components (reduced from 306 by PCA)

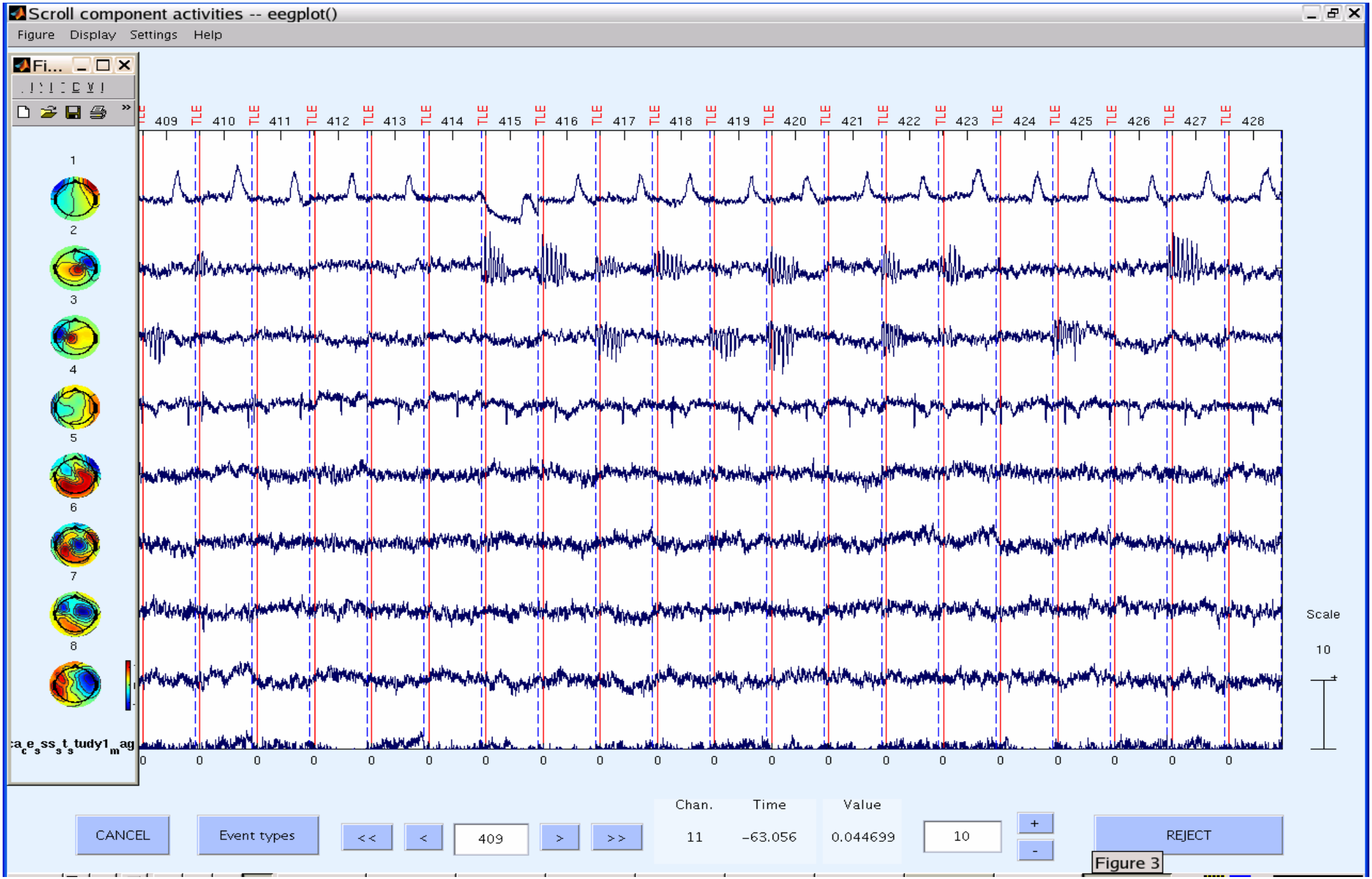
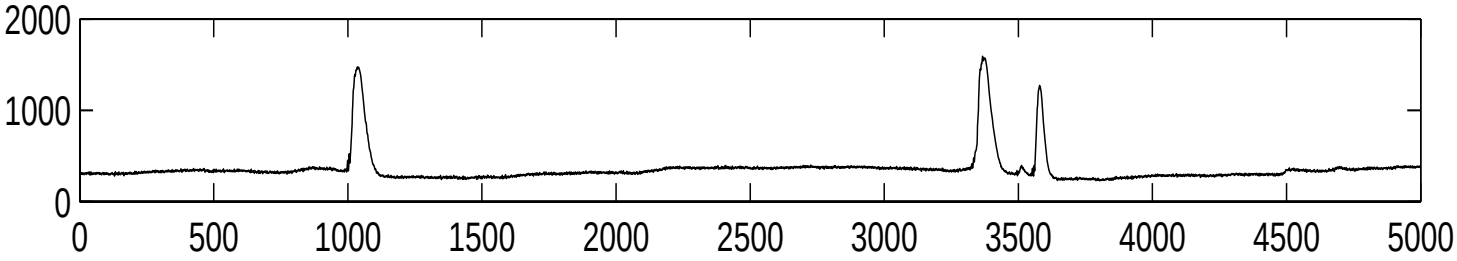
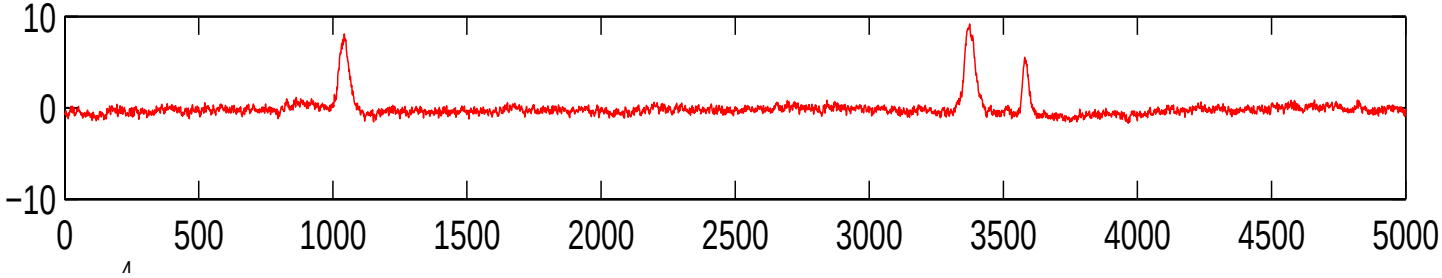


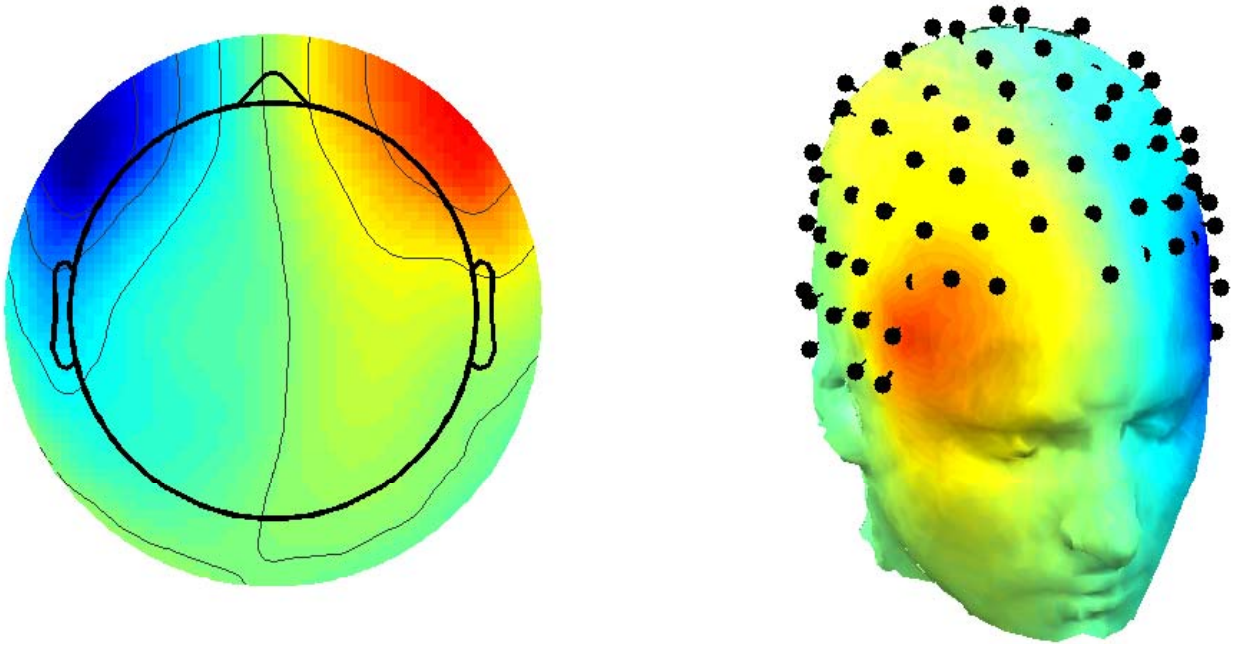
Figure 3

VEOG



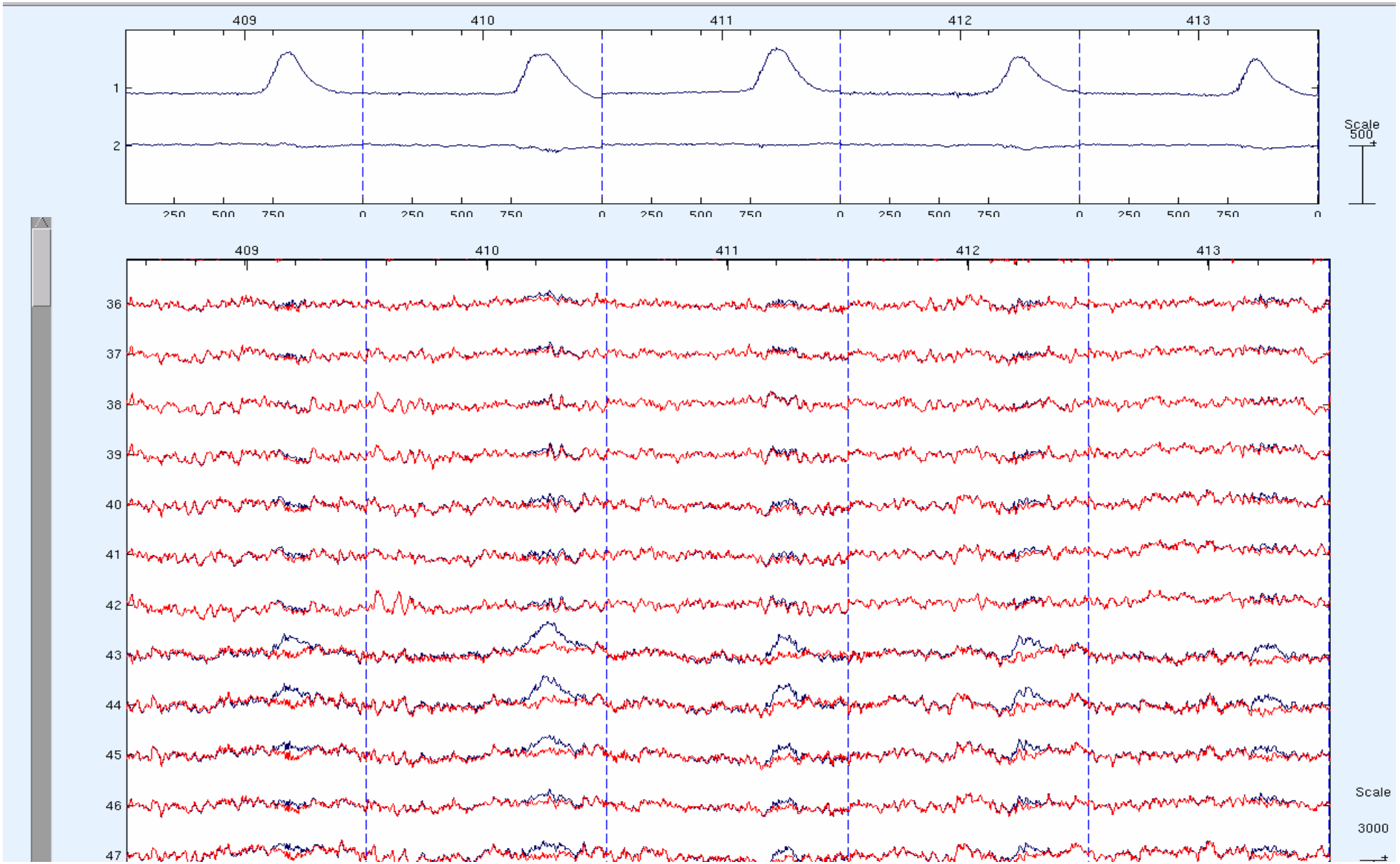
blink IC

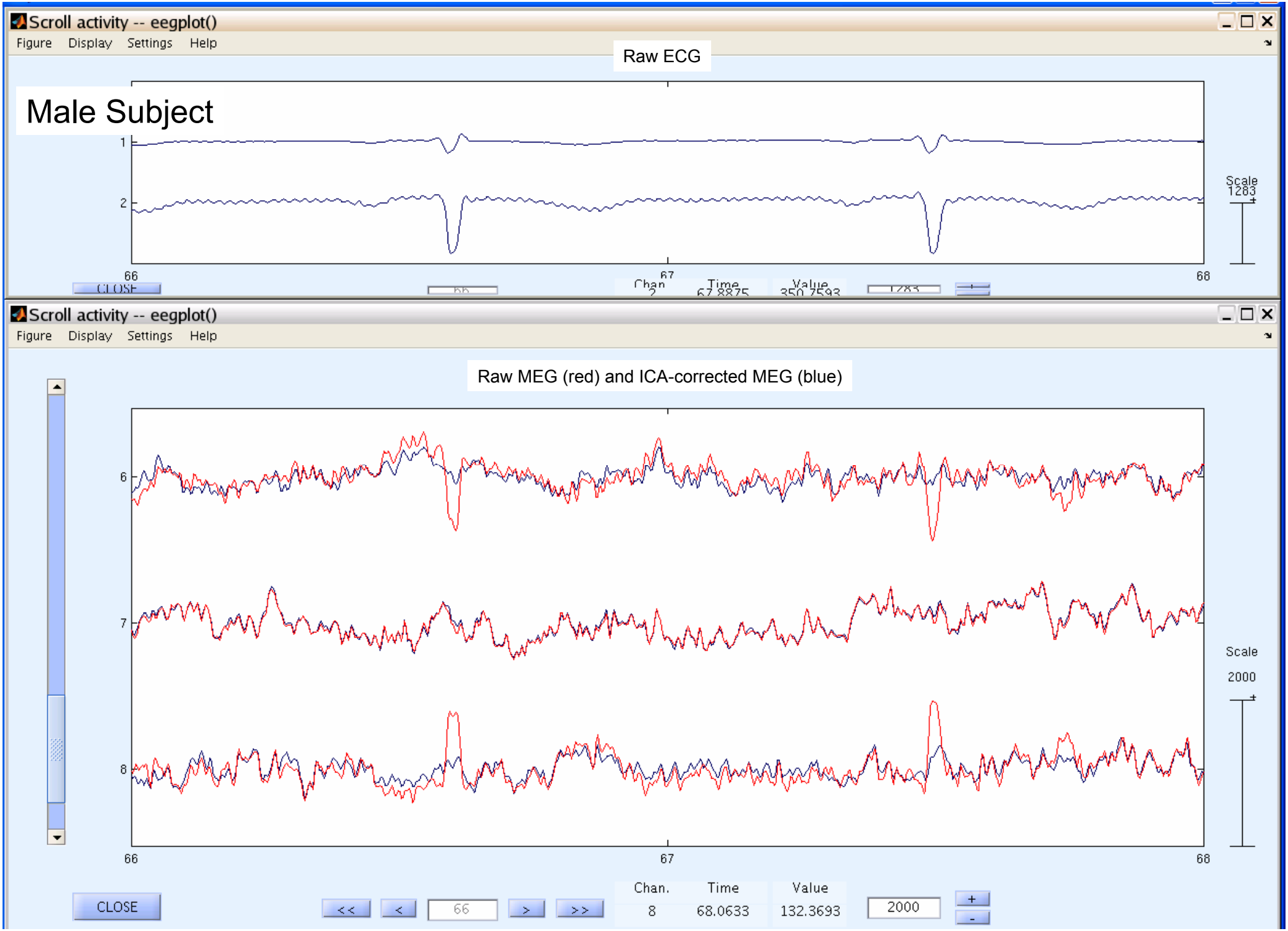




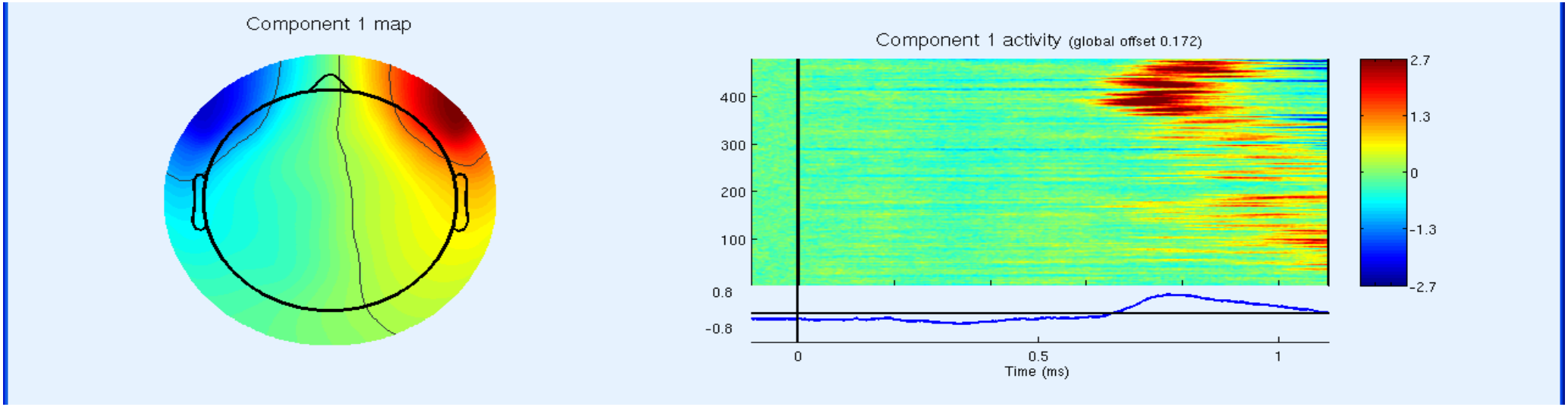
Scalp Topography of blink IC

Artefact Correction using ICA:
Raw epochs before/after blink component is removed

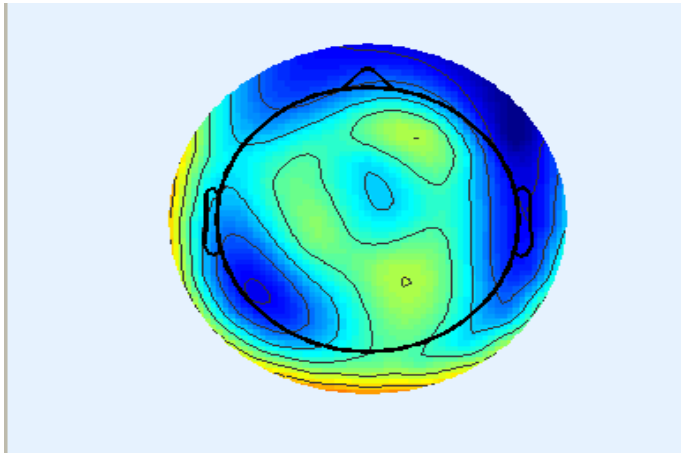


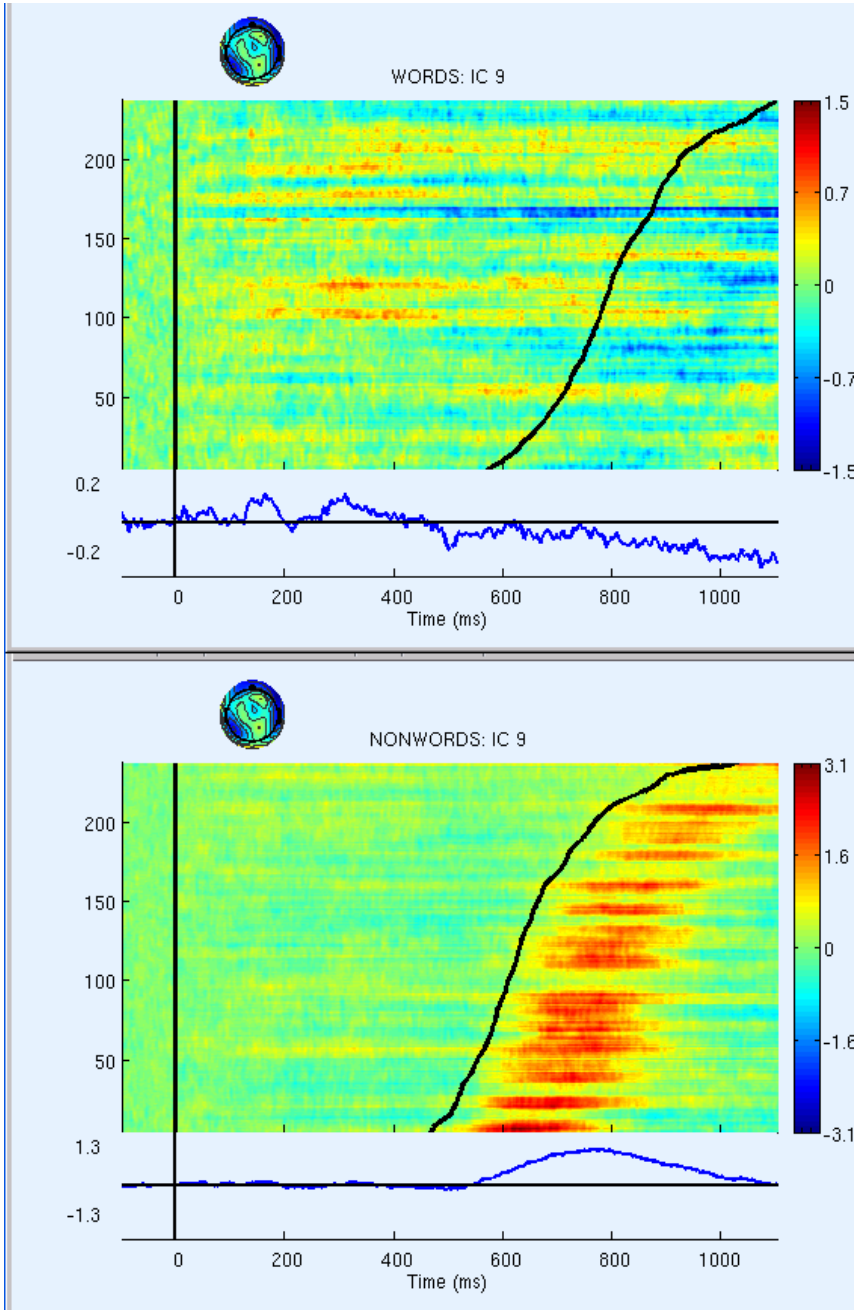

Scroll activity -- eegplot()
Figure Display Settings Help
Raw ECG
Male Subject
1
2
66
67
68
Scale
1283
CLOSE
Chan
Time
67.8875
Value
350.7593
Scroll activity -- eegplot()
Figure Display Settings Help
Raw MEG (red) and ICA-corrected MEG (blue)
6
7
8
66
67
68
Scale
2000
CLOSE
<<
<
66
>
>>
Chan.
Time
Value
8
68.0633
132.3693
2000
+
-

ICA – Event (Epoch) Related Plots



ICA – Event (Epoch) Related Plots

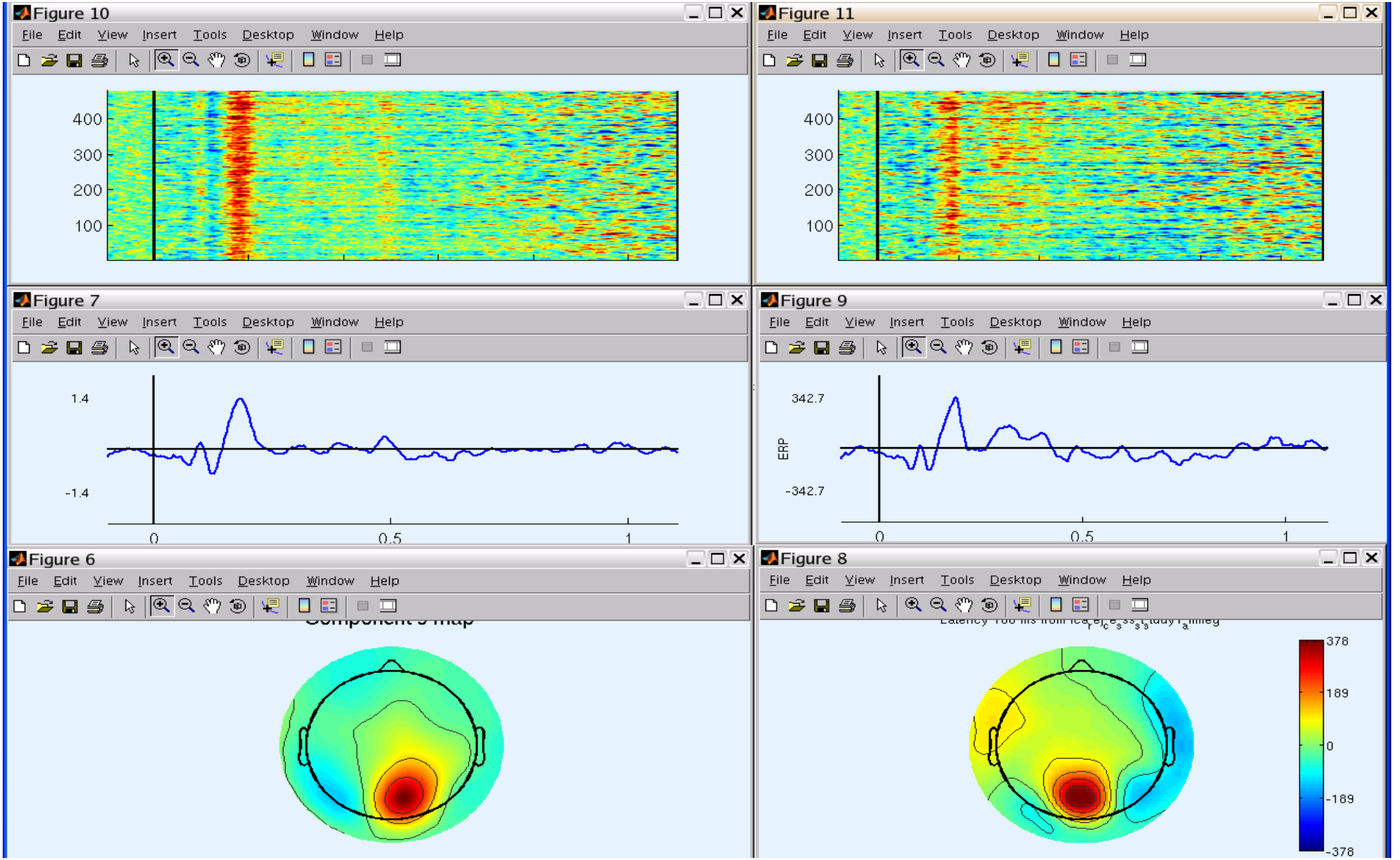




ICA – Event (Epoch) Related Plots

ICA – Component 9

MEG – Sensor 76 (2021) @ 186ms



Further Info:

A fantastic tutorial:
Aapo Hyvärinen:
http://www.cis.hut.fi/aapo/papers/IJCNN99_tutorialweb/IJCNN99_tutorial3.html

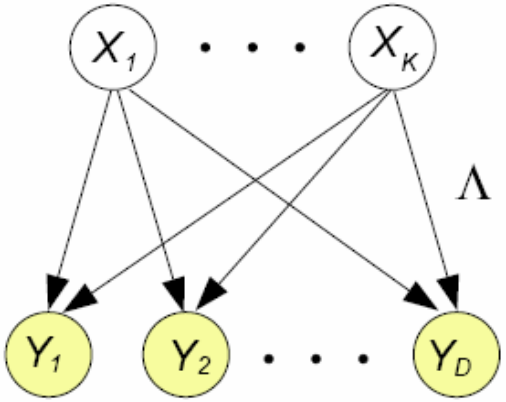
A demo of blind source separation applied to voice recordings:
Scott Makeig
http://sccn.ucsd.edu/~scott/icademo/index.html

Lecture 5 in this Machine Learning course deals with ICA:
Zoubin Ghahramani
http://learning.eng.cam.ac.uk/zoubin/ml06

Appendix

Slides from Zoubin Ghahramani: http://learning.eng.cam.ac.uk/zoubin/ml06

Independent Components Analysis



- Just like Factor Analysis, hidden factors in ICA are *independent*: $p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k)$
- **But**, their distribution $p(x_k)$ is *non-Gaussian*:

$$y_d = \sum_{k=1}^{K} \Lambda_{dk}\, x_k + \epsilon_d$$

- We can call the special case of $K = D$, with invertible Λ and zero observation noise, standard ICA. This was the originally proposed model (analogous to PCA) and has been studied extensively[1]:

$$\mathbf{y} = \Lambda\mathbf{x} \quad \text{which implies} \quad \mathbf{x} = W\mathbf{y} \quad \text{where} \quad W = \Lambda^{-1}$$

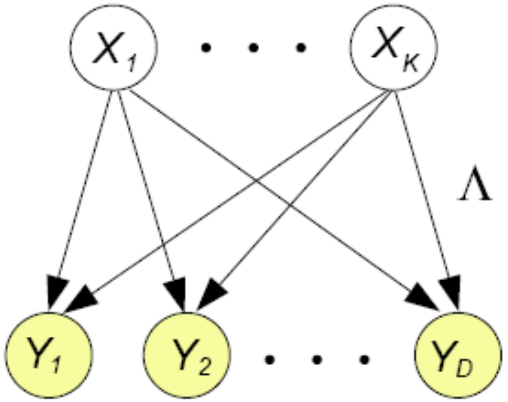
  where $\mathbf{x}$ are the independent components (factors), $\mathbf{y}$ are the observations, Λ is the mixing matrix, and W is the unmixing matrix.

- Inferring $\mathbf{x}$ given $\mathbf{y}$ and learning Λ is easy in standard ICA.

[1] See: `http://www.cnl.salk.edu/~tony/ica.html`

Slides from Zoubin Ghahramani: http://learning.eng.cam.ac.uk/zoubin/ml06

ICA: Choosing non-Gaussian hidden factor densities



- Just like Factor Analysis, hidden factors in ICA are *independent*: $p(\mathbf{x}) = \prod_{k=1}^{K} p(x_k)$
- **But**, their distribution $p(x_k)$ is *non-Gaussian*:

$$y_d = \sum_{k=1}^{K} \Lambda_{dk}\, x_k + \epsilon_d$$

There are many possible continuous non-Gaussian densities for the hidden factors $p(x_k)$ from which we can choose.

A major distinction between univariate distributions is whether they are heavy tailed or light tailed.

This is defined in terms of the kurtosis.

Slides from Zoubin Ghahramani: http://learning.eng.cam.ac.uk/zoubin/ml06

How ICA Relates to Factor Analysis and Other Models

- **Factor Analysis (FA)**: Linear latent variable model which assumes that the factors are Gaussian, and Gaussian observation noise.
- **Probabilistic Principal Components Analysis (pPCA)**: Assumes isotropic observation noise: $\Psi = \sigma^2 I$ (PCA: $\Psi = \lim_{\sigma^2 \to 0} \sigma^2 I$).
- **Independent Components Analysis (ICA)**: Assumes that the factors are non-Gaussian.
- **Mixture of Gaussians**: A single discrete-valued "factor": $x_k = 1$ and $x_j = 0$ for all $j \neq k$.
- **Linear Gaussian State-space Model (Linear Dynamical System)**: Time series model in which the factor at time t depends linearly on the factor at time $t-1$, with added Gaussian noise.

ICA can and has been extended in several ways: fewer sources than "microphones", time varying mixing matrices, combining with convolution with linear filters, discovering number of sources...

Slides from Zoubin Ghahramani: http://learning.eng.cam.ac.uk/zoubin/ml06

Appendix: Matlab Code for Standard ICA

```
% ICA using tanh nonlinearity and batch covariant algorithm
% (c) Zoubin Ghahramani
%
% function [W, Mu, LL]=ica(X,cyc,eta,Winit);
%
% X - data matrix (each row is a data point),    cyc - cycles of learning (default = 200)
% eta - learning rate (default = 0.2),           Winit - initial weight
%
% W - unmixing matrix,  Mu - data mean,          LL - log likelihoods during learning

function [W, Mu, LL]=ica(X,cyc,eta,Winit);

if nargin<2,  cyc=200; end;
if nargin<3,  eta=0.2; end;
[N D]=size(X);                    % size of data
Mu=mean(X); X=X-ones(N,1)*Mu;     % subtract mean
if nargin>3,    W=Winit;          % initialize matrix
else,   W=rand(D,D);  end;
LL=zeros(cyc,1);                  % initialize log likelihoods

for i=1:cyc,
  U=X*W';
  logP=N*log(abs(det(W)))-sum(sum(log(cosh(U))))-N*D*log(pi);
  W=W+eta*(W-tanh(U')*U*W/N);                    % covariant algorithm
  % W=W+eta*(inv(W)-X'*tanh(U)/N)';              % standard algorithm
  LL(i)=logP; fprintf('cycle %g log P= %g\n',i,logP);
end;
```

Slides from Zoubin Ghahramani: http://learning.eng.cam.ac.uk/zoubin/ml06