BCI Research: EEG Mapping as Inverse Problem

HCI

Fall 2016

What is EEG mapping

- To establish correspondence between the EEG in discrete points on the head surface and the activation zones in the brain
- These zones are responsible for (presumably) corresponding to certain ideas, functions, emotions, etc.
- The zones are considered to be *voxels* as sections of a 3D brain model.

Task formulation of the research under consideration

1. Experimental research
   - Using eMotiv or any similar headset collect EEG data for different cognitive and emotional states
   - Can be done in both temporal and frequency domains
   - Collected data are interpreted using ML (as we did last spring) for two emotional modes
2. Theoretical research
   1. Create a forward model – from activated voxels to the boundary surface potentials
   2. Associate voxel with specific cognitive and emotional states (setting *f-labels*)
   3. Collect modeled data: voxels activation mapped to the surface potential
   4. Create the ML model mapping a given surface potentials onto *f-labels*

Experimental research

- Many things were done in spring 2015 and can be continued now
- The projects covered mostly such areas as applications for educational use and recognition of emotions
- Collection of project reports can be found here:

BCI projects S15 HCI on the Content page of CSUDH site HCI 2016

- More specifically, there are the following subjects covered:
  - ❖Emotiv Xavier SDK (by Patrick Bailey)
  - ❖A BCI Learning Engine (by Adrian Mirabel, et al.)
  - ❖BCI based Slide show presentation (by Anusha Karur, et al.)

Theoretical research

- Inverse problems solved by Pasqual-Marqui (sLoreta)
- Forward Monte Carlo neural activity simulation and construction of ML model

Pasqual-Marqui

- Solving inverse problem
- Issues: sensitivity to noise, ill posed matrix, undetermined equations

Standardized low resolution brain electromagnetic tomography

*s*LORETA

Exact low resolution brain electromagnetic tomography

*e*LORETA

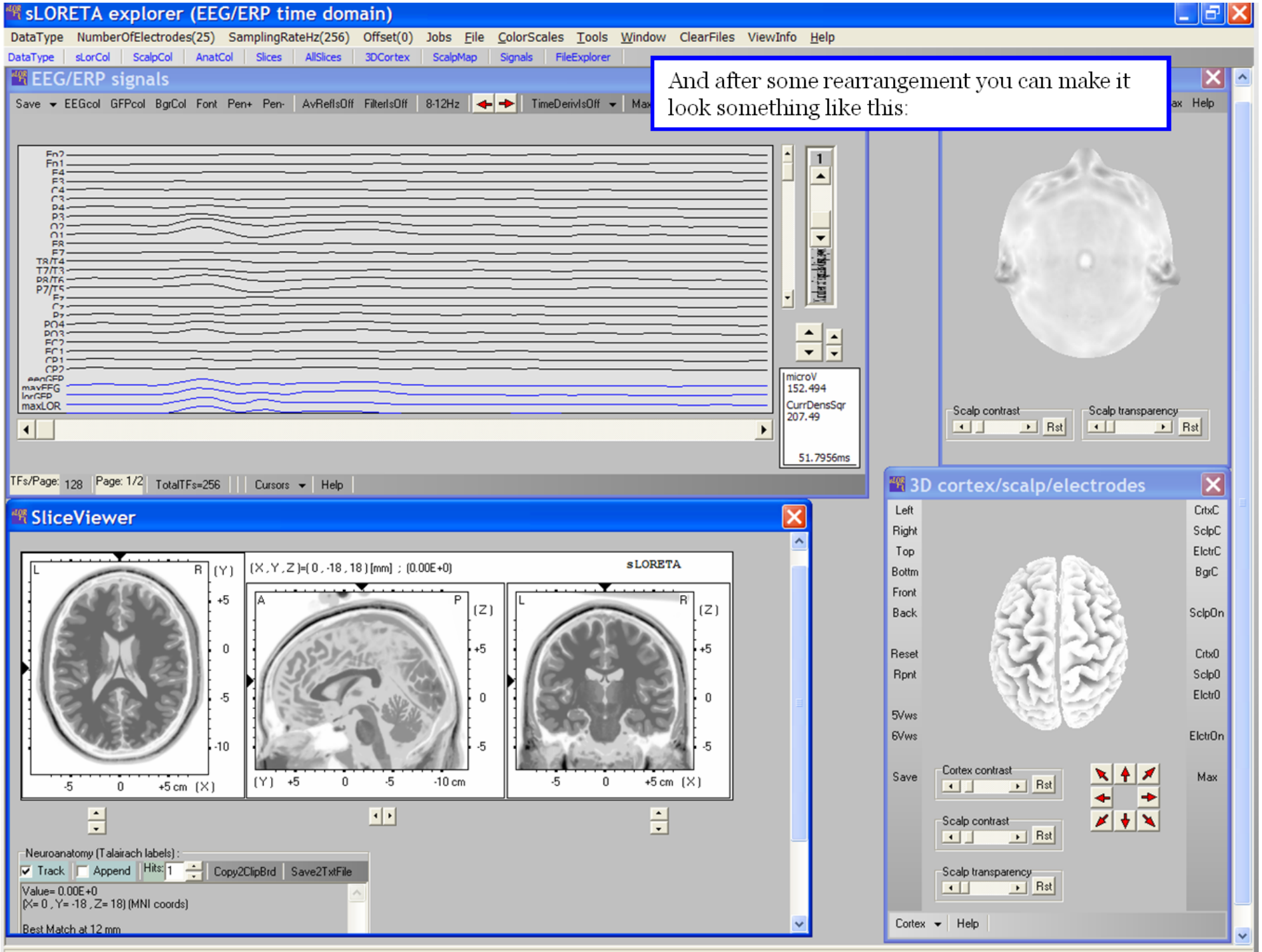
zero error, like it or not

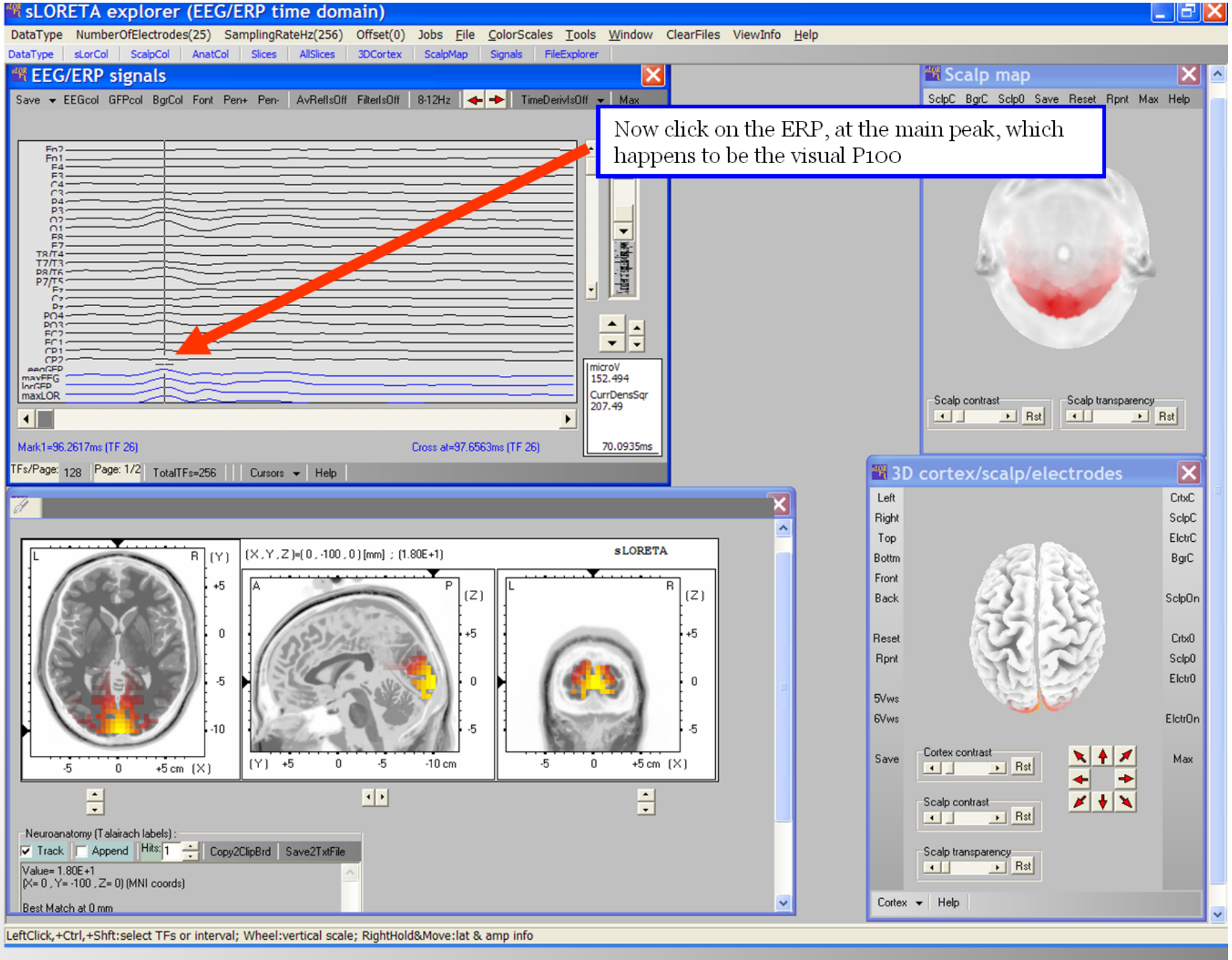
Functional connectivity

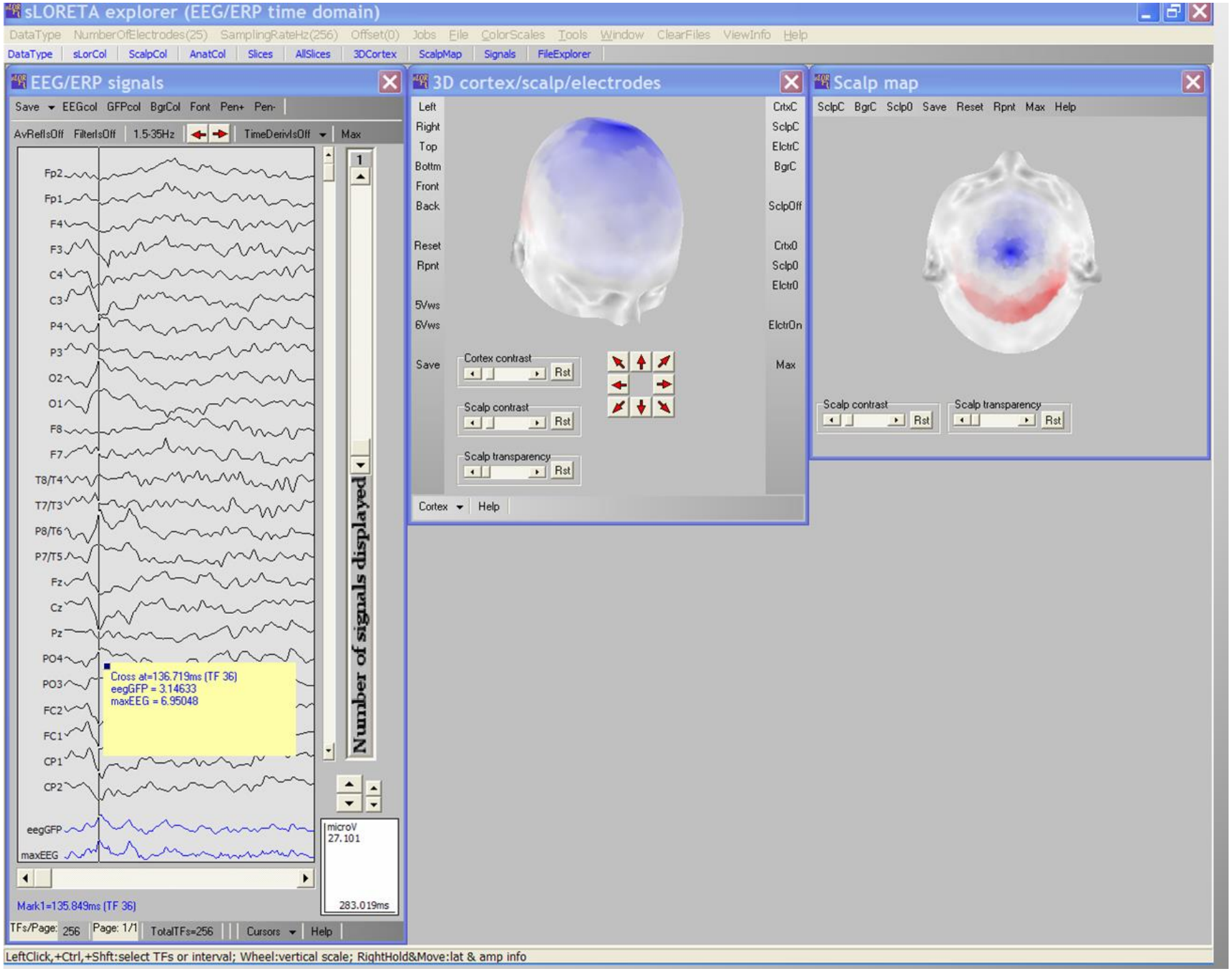
Lagged physiological coherence and phase synchronization, functional ICA (fICA), isolated effective coherence (iCoh)

Roberto D. Pascual-Marqui
The KEY Institute for Brain-Mind Research
University Hospital of Psychiatry, Zurich Swiyzerland
p a s c u a l m @ key . u z h . c h
www.keyinst.uzh.ch/loreta.htm

Guest Professor at Department of Neuropsyquiatry
Kansai Medical University, Osaka, Japan


sLORETA explorer (EEG/ERP time domain)
DataType NumberOfElectrodes(25) SamplingRateHz(256) Offset(0) Jobs File ColorScales Tools Window ClearFiles ViewInfo Help
DataType sLorCol ScalpCol AnatCol Slices AllSlices 3DCortex ScalpMap Signals FileExplorer
EEG/ERP signals
Save EEGcol GFPcol BgrCol Font Pen+ Pen- AvRefIsOff FilterIsOff 8-12Hz TimeDerivIsOff Help
maxLOR
microV
152.494
CurrDensSqr
207.49
51.7956ms
TFs/Page: 128 Page: 1/2 TotalTFs=256 Cursors Help
Scalp contrast Rst
Scalp transparency Rst
And after some rearrangement you can make it look something like this:
SliceViewer
(X,Y,Z)=(0 , -18 , 18) [mm] ; (0.00E+0)
sLORETA
Neuroanatomy (Talairach labels) :
Track Append Hits: 1 Copy2ClipBrd Save2TxtFile
Value= 0.00E+0
(X= 0 , Y= -18 , Z= 18) (MNI coords)
Best Match at 12 mm
3D cortex/scalp/electrodes
Left Right Top Bottm Front Back Reset Rpnt 5Vws 6Vws Save
CrtxC SclpC ElctrC BgrC SclpOn CrtxO SclpO ElctrO ElctrOn Max
Cortex contrast Rst
Scalp contrast Rst
Scalp transparency Rst
Cortex Help


sLORETA explorer (EEG/ERP time domain)
DataType NumberOfElectrodes(25) SamplingRateHz(256) Offset(0) Jobs File ColorScales Tools Window ClearFiles ViewInfo Help
DataType sLorCol ScalpCol AnatCol Slices AllSlices 3DCortex ScalpMap Signals FileExplorer
EEG/ERP signals
Save EEGcol GFPcol BgrCol Font Pen+ Pen- AvRefIsOff FilterIsOff 8-12Hz TimeDerivIsOff Max
Now click on the ERP, at the main peak, which happens to be the visual P100
microV 152.494
CurrDensSqr 207.49
70.0935ms
Mark1=96.2617ms (TF 26)
Cross at=97.6563ms (TF 26)
TFs/Page: 128 Page: 1/2 TotalTFs=256 Cursors Help
Scalp map
SclpC BgrC Sclp0 Save Reset Rpnt Max Help
Scalp contrast Rst
Scalp transparency Rst
(X,Y,Z)=(0,-100,0) [mm] ; (1.80E+1)
sLORETA
Neuroanatomy (Talairach labels):
Track Append Hits: 1 Copy2ClipBrd Save2TxtFile
Value= 1.80E+1
(X= 0 , Y= -100 , Z= 0) (MNI coords)
Best Match at 0 mm
3D cortex/scalp/electrodes
Left Right Top Bottm Front Back Reset Rpnt 5Vws 6Vws Save
CrtxC SclpC ElctrC BgrC SclpOn Crtx0 Sclp0 Elctr0 ElctrOn Max
Cortex contrast Rst
Scalp contrast Rst
Scalp transparency Rst
Cortex Help
LeftClick,+Ctrl,+Shft:select TFs or interval; Wheel:vertical scale; RightHold&Move:lat & amp info


sLORETA explorer (EEG/ERP time domain)
DataType NumberOfElectrodes(25) SamplingRateHz(256) Offset(0) Jobs File ColorScales Tools Window ClearFiles ViewInfo Help
DataType sLorCol ScalpCol AnatCol Slices AllSlices 3DCortex ScalpMap Signals FileExplorer
EEG/ERP signals
Save EEGcol GFPcol BgrCol Font Pen+ Pen-
AvRefIsOff FilterIsOff 1.5-35Hz TimeDerivIsOff Max
Fp2
Fp1
F4
F3
C4
C3
P4
P3
O2
O1
F8
F7
T8/T4
T7/T3
P8/T6
P7/T5
Fz
Cz
Pz
PO4
PO3
FC2
FC1
CP1
CP2
eegGFP
maxEEG
Cross at=136.719ms (TF 36)
eegGFP = 3.14633
maxEEG = 6.95048
Number of signals displayed
microV
27.101
283.019ms
Mark1=135.849ms (TF 36)
TFs/Page: 256 Page: 1/1 TotalTFs=256 Cursors Help
3D cortex/scalp/electrodes
Left Right Top Bottm Front Back Reset Rpnt 5Vws 6Vws Save
CrtxC SclpC ElctrC BgrC SclpOff Crtx0 Sclp0 Elctr0 ElctrOn Max
Cortex contrast Rst
Scalp contrast Rst
Scalp transparency Rst
Cortex Help
Scalp map
SclpC BgrC Sclp0 Save Reset Rpnt Max Help
Scalp contrast Rst
Scalp transparency Rst
LeftClick,+Ctrl,+Shft:select TFs or interval; Wheel:vertical scale; RightHold&Move:lat & amp info

Identifying Emotions on the Basis of Neural Activation (Carnegie-Mellon U.)

"We attempt to determine the discriminability and organization of neural activation corresponding to the experience of specific emotions.

Method actors were asked to self-induce nine emotional states (anger, disgust, envy, fear, happiness, lust, pride, sadness, and shame) while in an fMRI scanner.

Using a Gaussian Naıve Bayes pooled variance classifier, we demonstrate the ability to identify specific emotions experienced by an individual at well over chance accuracy on the basis of:

1) neural activation of the same individual in other trials,

2) neural activation of other individuals who experienced similar trials, and

3) neural activation of the same individual to a qualitatively different type of emotion induction.

These results suggest a structure for neural representations of emotion and inform theories of emotional processing."

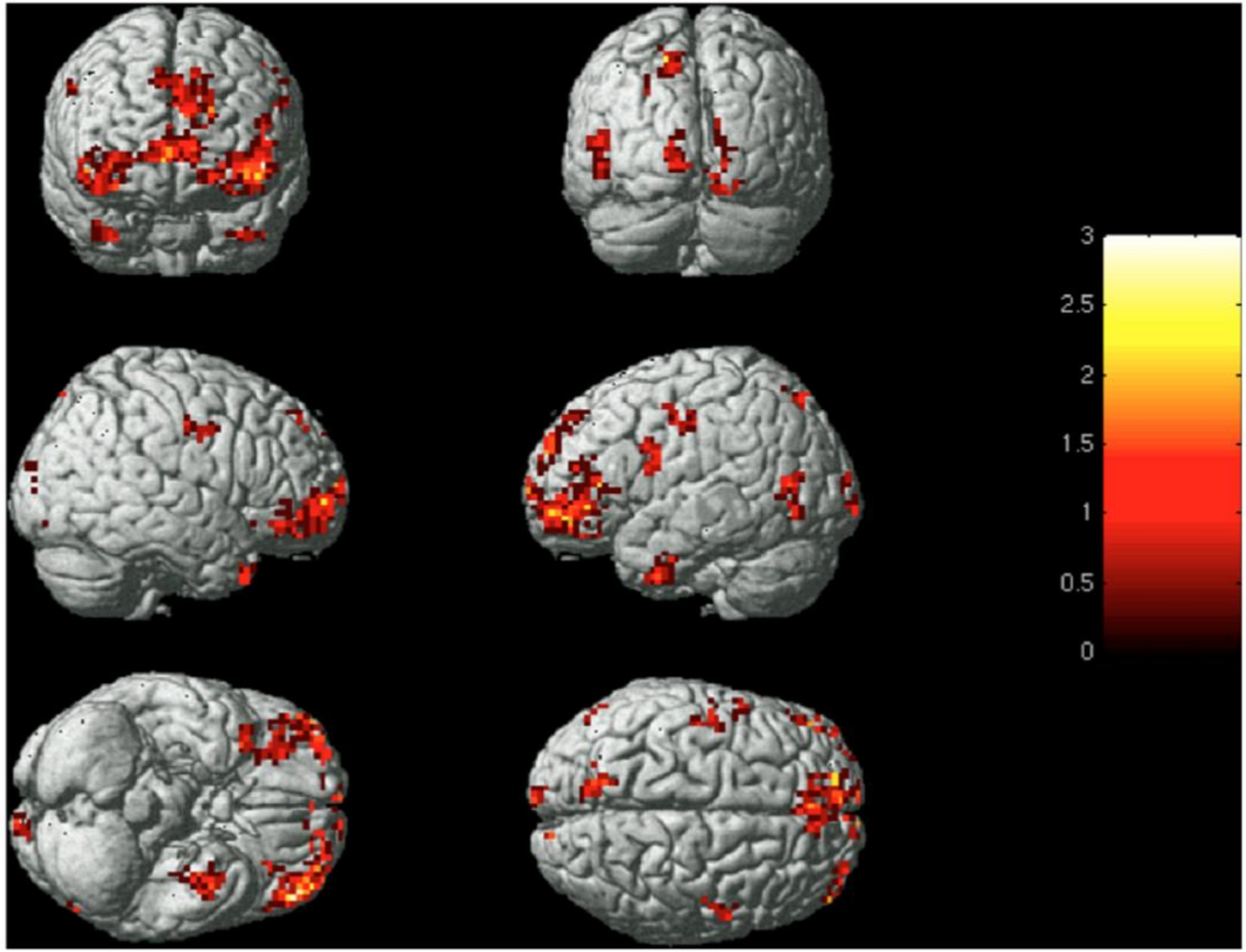
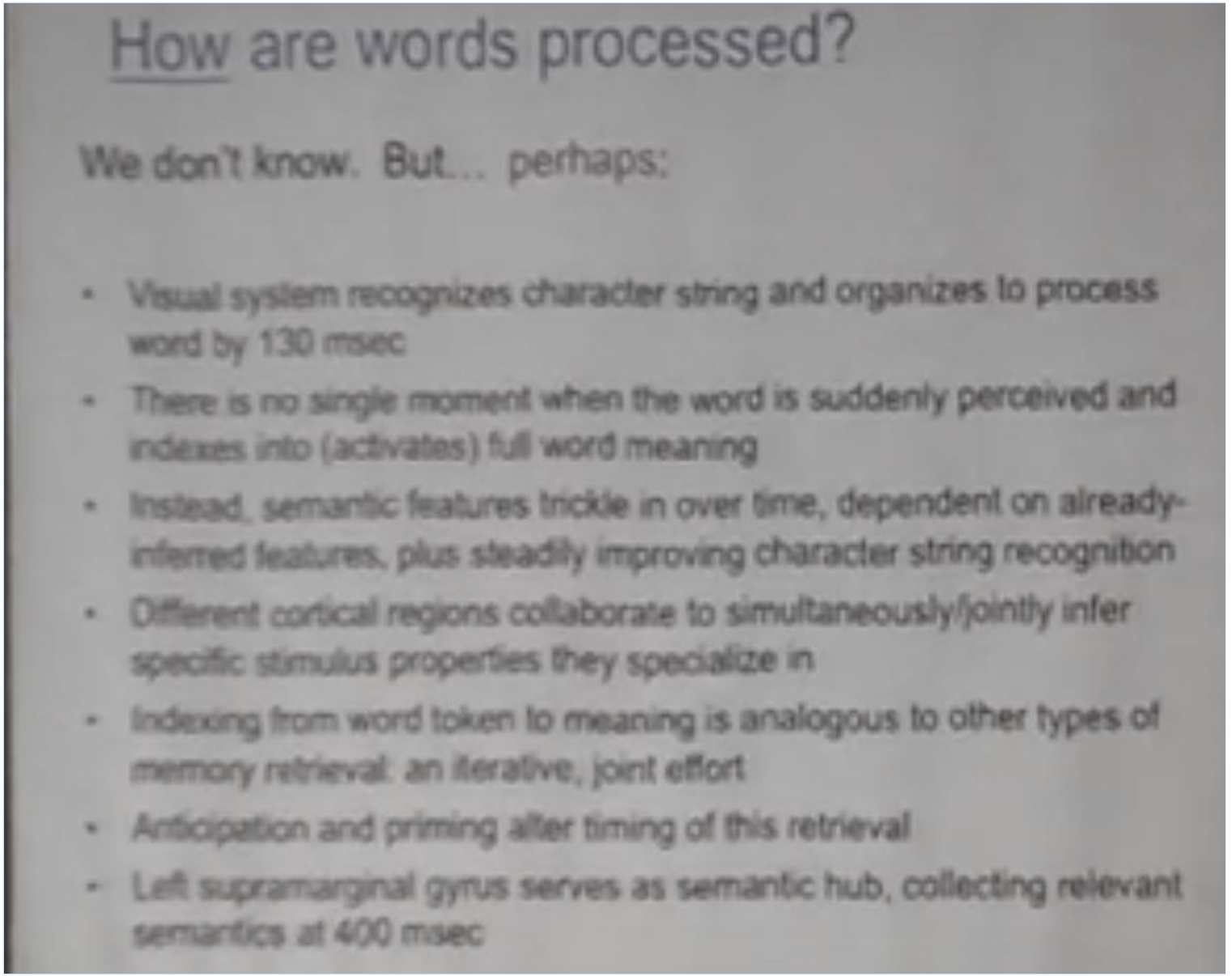


Figure 3. Group-level image of voxels used for within-subject classification. Images of the 240 most stable voxels across six presentations of emotion words were superimposed on one another, with resulting clusters of 25 or more voxels depicted. Color intensity reflects the number of participants for whom the voxel was among the 240 highest in stability.
doi:10.1371/journal.pone.0066032.g003

Tom Mitchel: How are word processed in the brain



Berkley semantic map

Natural speech reveals the semantic maps that tile human cerebral cortex *by* Alexander G. Huth, Wendy A. de Heer, Thomas L. Griffiths, Frederic E. Theunissen & Jack L. Gallant (Nature, 2016)

extremetech.com/extreme/227498-uc-berkeley-team-built-a-semantic-atlas-of-the-human-brain

http://www.nature.com/nature/journal/v532/n7600/full/nature17637.html

From authors

Our goal in this study was to map how the brain represents the meaning (or “semantic content”) of language. Most earlier studies of language in the brain have used isolated words or sentences. We used natural, narrative story stimuli because we wanted to map the full range of semantic concepts in a single study. This made it possible for us to construct a semantic map for each individual, which shows which brain areas respond to words with similar meaning or semantic content. Another aim of this study was to create a semantic atlas by combining data from multiple subjects, showing which brain areas represented similar information across subjects

What's new there?

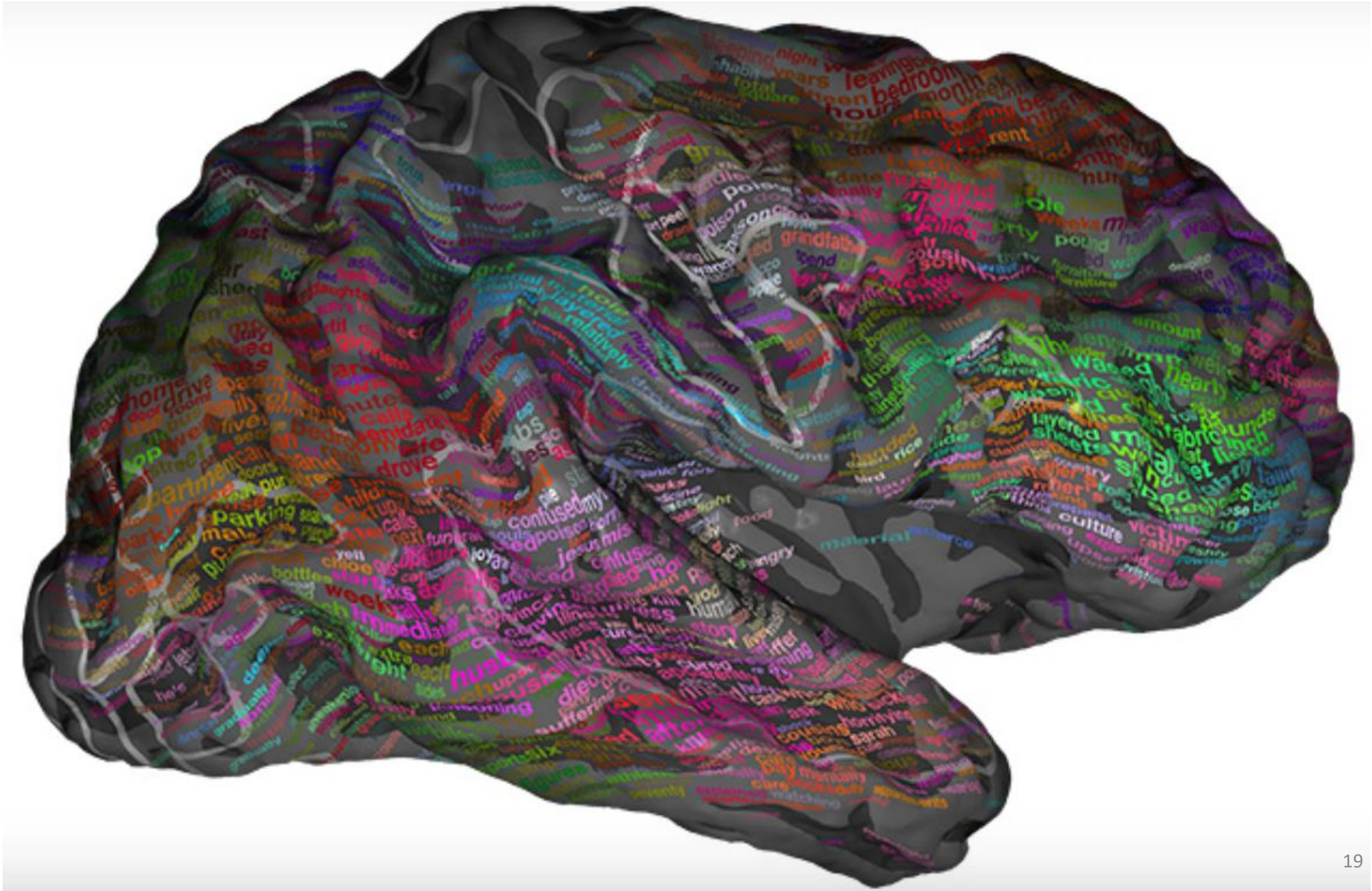
- We took a data driven approach rather than the conventional method of point-null hypothesis testing.
- We used natural narrative language rather than simple words presented in isolation.
- We used cross-validation on a separate dat a set to test model prediction performance and generalization.
- We used voxel-wise modeling to construct a separate semantic tuning curve for each voxel in each participant.
- We used a probabilistic and generative model of areas tiling cortex (PrAGMATiC) to construct the semantic atlas.

What were the main conclusions of the study?

Our study did not set out to test a single hypothesis or to ask a simple question. Instead, we sought to exhaustively map the representation of the meaning, or semantic information, in narrative language, across the entire cerebral cortex. The resulting maps show that semantic information is represented in rich patterns that are distributed across several broad regions of cortex. Furthermore, each of these regions contains many distinct areas that are selective for particular types of semantic information, such as people, numbers, visual properties, or places. **We also found that these cortical maps are quite similar across people, even down to relatively small details.**

How does this study change the way that we think about language and the brain?

These semantic maps give us, for the first time, a detailed map of how meaning is represented across the human cortex. Rather than being limited to a few brain areas, we find that language engages very broad regions of the brain. We also find that these representations are highly bilateral: responses in the right cerebral hemisphere are about as large and as varied as responses in the left hemisphere. This challenges the current dogma (inherited from studies of language production, as opposed to language comprehension as studied here) holding that language involves only the left hemisphere.



flowing
bowl
weight
washed
fabric
pounds
cloth
inch
lighter
metal
sheets
worn
shaped
holes
sheet
husband
children
family
school
apartment
parking
seat
male
purse
convinced
confused
cousin
poisoning
human
sickness
illness
mother
afterwards
cured
death
suffering
family
who
consisted
white
hose
fluid
tightly
onto
underneath
length
weight
onto
slide
thousand
horrified
thousand
quarter
six

Our approach to investigate

- Generate activation voxels and calculate the potential on the surface
- Collect data to create a ML model: EEG to f-label
- Use of Monte-Carlo, Latin cube, etc.
- Avoid ill-posed matrix operations

Our approach - why

- Using EEG to identify what's "on your mind" is faster (real time) than MRI and fMRI used in the many published works.
- Opportunities to take an instantaneous electrical signals (EEG) from the scalp and map them onto the *f-labels* (emotions, words, etc.)

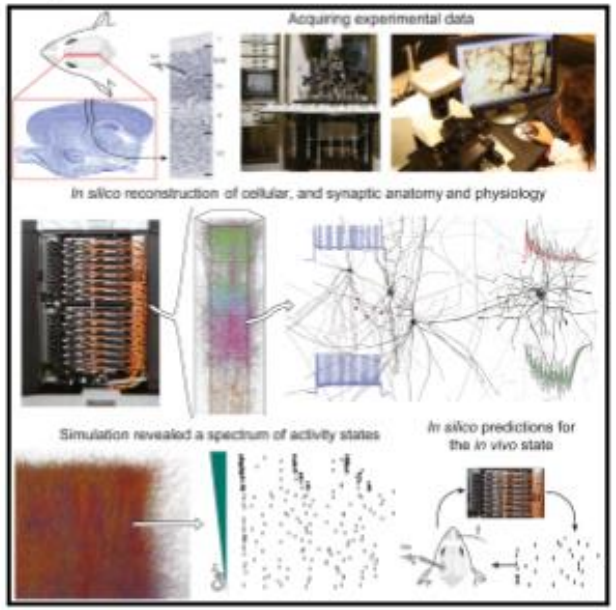
Forward model

- Neural network level
- Can be huge – really Big Data model
- Currently model of mice brains introduced with more than 30 million synapses
- For our purpose can be smaller

Cell

Reconstruction and Simulation of Neocortical Microcircuitry

Graphical Abstract



Authors

Henry Markram, Eilif Muller, Srikanth Ramaswamy, Michael W. Reimann, ..., Javier DeFelipe, Sean L. Hill, Idan Segev, Felix Schürmann

Correspondence

henry.markram@epfl.ch

In Brief

A digital reconstruction and simulation of the anatomy and physiology of neocortical microcircuitry reproduces an array of in vitro and in vivo experiments without parameter tuning and suggests that cellular and synaptic mechanisms can dynamically reconfigure the state of the network to support diverse information processing strategies.

Highlights

- The Blue Brain Project digitally reconstructs and simulates a part of neocortex
- Interdependencies allow dense in silico reconstruction from sparse experimental data
- Simulations reproduce in vitro and in vivo experiments without parameter tuning
- The neocortex reconfigures to support diverse information processing strategies

Markram et al., 2015, Cell 163, 456–492
October 8, 2015 ©2015 Elsevier Inc.
http://dx.doi.org/10.1016/j.cell.2015.09.029

Forward model: Method of Fundamental Solutions

The brain electrical field *u* can be modeled as an elliptical equation:

$$Lu = f(x, y, z), \qquad (x, y, z) \in G,$$
$$u = g(x, y, z); \qquad (x, y, z) \in \partial G$$

where *G* is the domain (the brain);

$f(x, y, z)$ is the source function representing voxel's level of electrical activity, and $g(x, y, z)$ is the boundary conditions (the potentials on the surface of the brain).

MFS

Using the method of fundamental solutions means to construct solution's approximation, $u^*(x,y,z)$, as the linear combination of the basis functions of the elliptical equation (functions of the distance between the source points and the collocation points on the boundary where the electrodes are located):

$$u^*(x,y,z) = \sum_{i=1}^{N} \alpha_i \varphi(r_i)$$

$$r_i = \|(x,y,z) - (sx_i, sy_i, sz_i)\|$$

where (sx_i, sy_i, sz_i) is the location of the source (the center of the voxel).

Basis functions

In the 3D case the basis function

$$\varphi(r_i) = 1/r_i$$

The MFS representation is equivalent to the model used in LORETA, that is a set of local electrical sources.

Inverse problem approach

- The Monte-Carlo method is suggested because it is supposed to avoid the instability of the inverse method that leads to the operations with the ill-imposed matrices.
- The algorithm is a sequence of the direct solutions – from the sources to the boundary points.
- The values of the potentials in the sources are selected randomly.

Algorithm

- The solutions are registered at the boundary points.
- The solutions can be presented as the look-up table.
- This table is used in the real time mapping of the electrode potentials to the voxels intensity. The measured vector of scalp potentials is compared with the closest match using some norm (the max(.) norm may work better).

OR, which is what we are going to study

- A Machine Learning model should be created to map the scalp EEG potentials onto the f-labels
- f-labels are functions (such as emotions, words, etc) associated with specific voxels

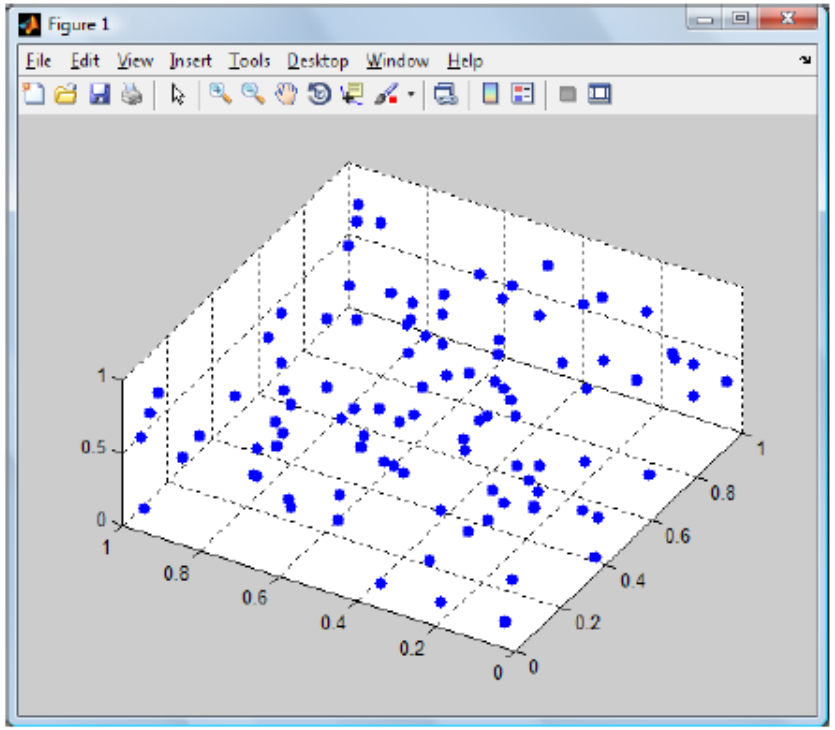
Latin Hypercube

- The Monte Carlo method is based on propagating probability distributions.
- Given a region in design space, we can assign a uniform distribution to the region and sample points.
- It is likely, though, that some regions will be poorly sampled.
- For example, in 5-dimensional space, with 32 sample points, the chance that all *n*- dimensional quadrants (orthants) will be occupied is (31/32)(30/32)(1/32) …(1/32) =$1.8e^{-13}$.

Latin Hypercube sampling

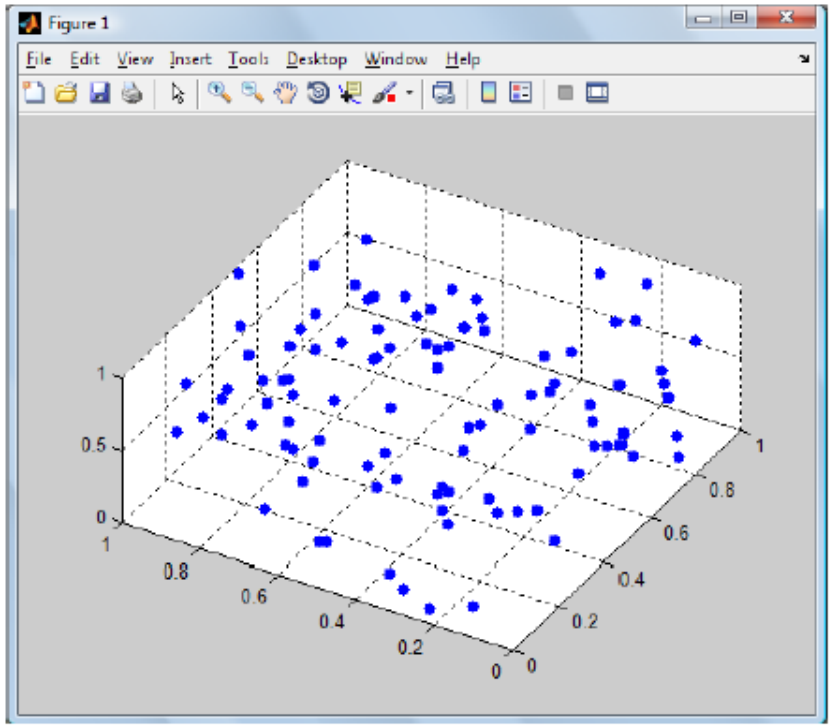
- To scan the parametric space deterministically, a prohibitively high number of calculations is needed.
- Monte Carlo simulation: the chance is that the intervals are sampled with higher density because the same random sample can contribute to more than one parameter's sampling.
- It can be a waste of simulation runs if the clusters are formed in less important parts of parametric space
- Latin Hypercube, orthogonal sampling, uniform space filing methods and others can be used.
- Applied to both sensitivity analysis and parametric interval control.
- Each variable range divided into *n* equal probability intervals.

MATLAB models of LHC



1st run

```
scatter3(XX(1,:),XX(2,:),XX(3,:),
'filled'),view(-60,60)
```



2nd run

```
scatter3(XX(1,:),XX(2,:),XX(3,:),
'filled'),view(-60,60)
```

MATLAB Halton sequences

- Can be used for Monte Carlo simulation of neural activities as well as LHS
- They are deterministic but behave like being random
- Supported by MATLAB
- Only low dimensional problems can be used reliably

Conclusion: Any efficient Monte Carlo procedure is to be tested for modeling neural activities

Technical issues

- Graphics
- Linear algebra
- ML models

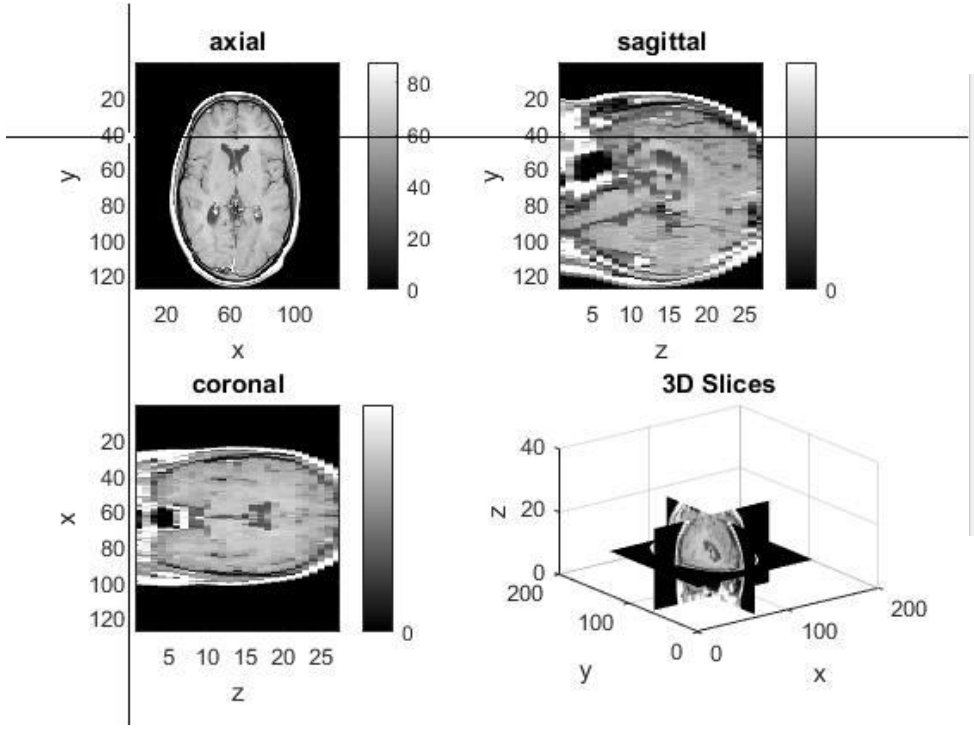
Software to use

- MATLAB
- EEGLab

Examples of what is under research currently

- mriplot.m
- isocap.m
- mri.m
- DistanceMatrix.m
- EEG_1.m

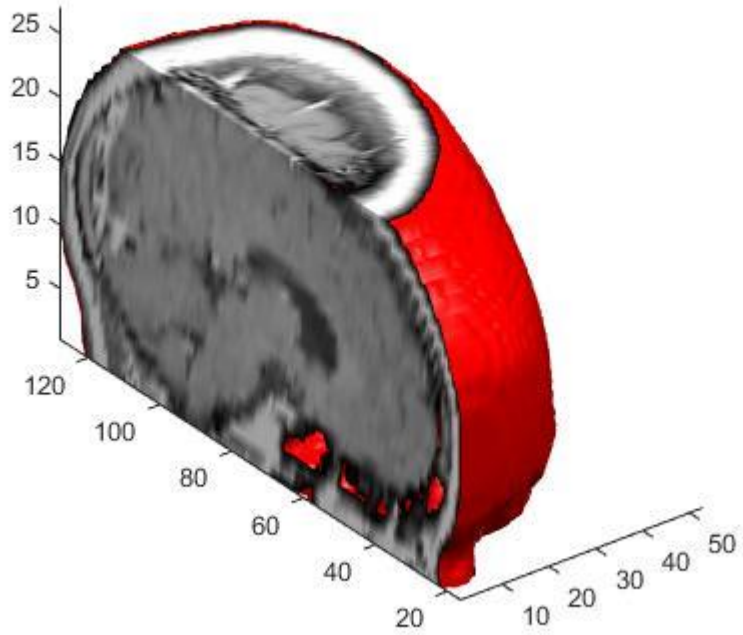
mriplot.m




```
load mri.mat;
D1=double(squeeze(D));
DIM = size(D1);
[X,Y,Z]=meshgrid(1:DIM(2),1:DIM(1),1:DIM(3));
h1=subplot(2,2,1);imagesc(D1(:,:,round(DIM(3)/2)),[min(D1(:)) max(D1(:))]);colormap(gray);title('axial');colorbar;
xlabel('x');ylabel('y')
h2=subplot(2,2,2);imagesc(squeeze(D1(:,round(DIM(2)/2),:)),[min(D(:)) max(D(:))]);colormap(gray);title('sagittal');colorbar;
xlabel('z');ylabel('y')
h3=subplot(2,2,3);imagesc(squeeze(D1(round(DIM(1)/2),:,:)),[min(D(:)) max(D(:))]);colormap(gray);title('coronal');colorbar;
xlabel('z');ylabel('x')
subplot(2,2,4);slice(X,Y,Z,D1,64,64,14);colormap(gray);shading flat;title('3D Slices')
xlabel('x');ylabel('y');zlabel('z');


x=round(DIM(2)/2);y=round(DIM(1)/2);z=round(DIM(3)/2);
button = 0;

while(1)
```

isocap.m

```
load mri.mat;
D1=double(squeeze(D));
DIM = size(D1);
[X,Y,Z]=meshgrid(1:DIM(2),1:DIM(1),1:DIM(3));
h1=subplot(2,2,1);imagesc(D1(:,:,round(DIM(3)/2)),[min(D1(:)) max(D1(:))]);colormap(gray);title('axial');colorbar;
xlabel('x');ylabel('y')
h2=subplot(2,2,2);imagesc(squeeze(D1(:,round(DIM(2)/2),:)),[min(D(:)) max(D(:))]);colormap(gray);title('sagittal');colorbar;
xlabel('z');ylabel('y')
h3=subplot(2,2,3);imagesc(squeeze(D1(round(DIM(1)/2),:,:)),[min(D(:)) max(D(:))]);colormap(gray);title('coronal');colorbar;
xlabel('z');ylabel('x')
subplot(2,2,4);slice(X,Y,Z,D1,64,64,14);colormap(gray);shading flat;title('3D Slices')
xlabel('x');ylabel('y');zlabel('z');


x=round(DIM(2)/2);y=round(DIM(1)/2);z=round(DIM(3)/2);
button = 0;

while(1)
```



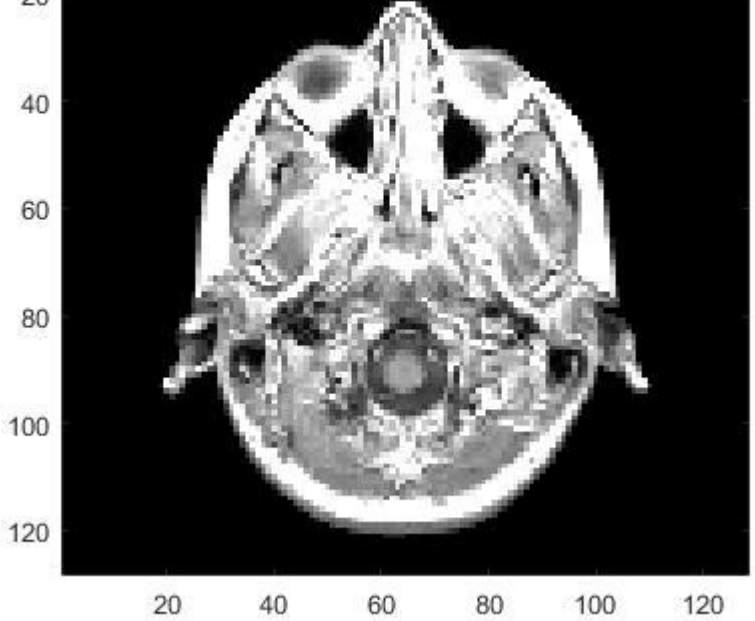
DistanceMatrix.m

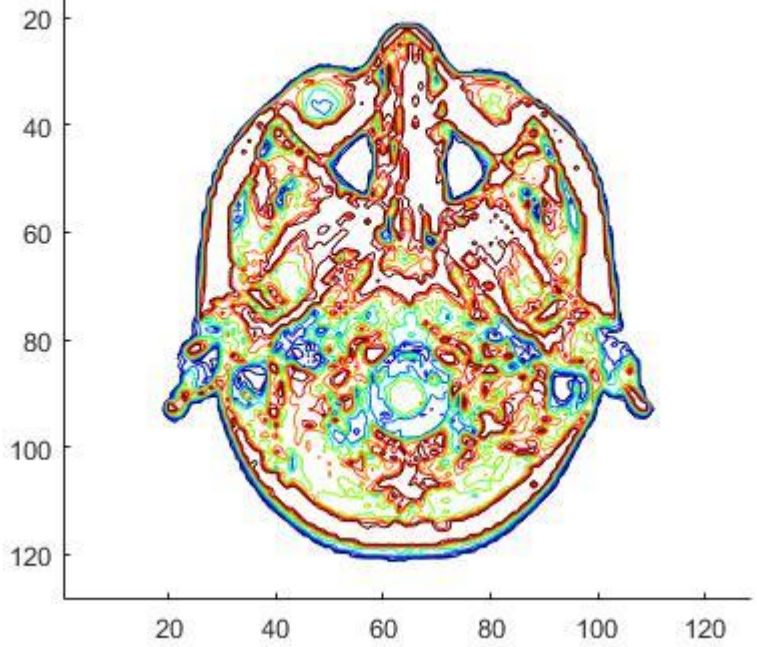
```
% DM = DistanceMatrix(dsites,ctrs)
% Forms the distance matrix of two sets of points in R^s,
% i.e., DM(i,j) = || datasite_i - center_j ||_2.
% Input
%   dsites: Mxs matrix representing a set of M data sites in
R^s
%              (i.e., each row contains one s-dimensional
point)
%   ctrs:   Nxs matrix representing a set of N centers in
R^s
%              (one center per row)
% Output
%   DM:     MxN matrix whose i,j position contains the
Euclidean
%              distance between the i-th data site and j-th
center
  function DM = DistanceMatrix(dsites,ctrs)
  [M,s] = size(dsites); [N,s] = size(ctrs);
  DM = zeros(M,N);
  % Accumulate sum of squares of coordinate differences
  % The ndgrid command produces two MxN matrices:
  %   dr, consisting of N identical columns (each containing
  %       the d-th coordinate of the M data sites)
  %   cc, consisting of M identical rows (each containing
  %       the d-th coordinate of the N centers)
  for d=1:s
     [dr,cc] = ndgrid(dsites(:,d),ctrs(:,d));
     DM = DM + (dr-cc).^2;
  end
  DM = sqrt(DM);
```

mri.m

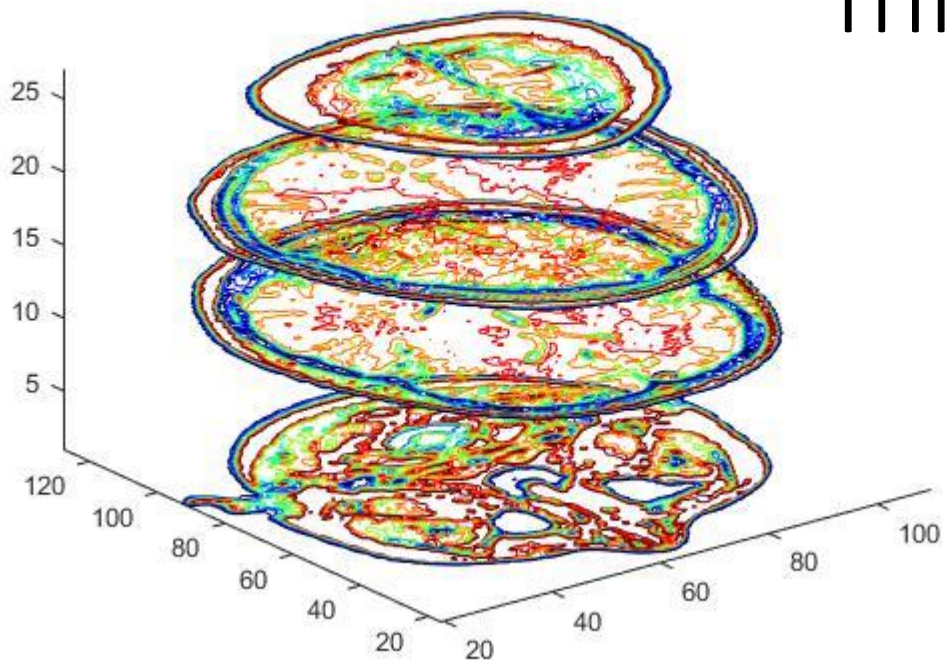
```
load mri
D = squeeze(D);
% Displaying Images of MRI Data
figure
colormap(map)
image_num = 2;
image(D(:,:,image_num))
axis image
x = xlim;
y = ylim;
% Displaying a 2-D Contour Slice
cm = brighten(jet(length(map)),-.5);
figure
colormap(cm)
contourslice(D,[],[],image_num)
axis ij
xlim(x)
ylim(y)
daspect([1,1,1])
% Displaying 3-D Contour Slices
figure
colormap(cm)
contourslice(D,[],[],[1,12,19,27],8);
view(3);
axis tight
```

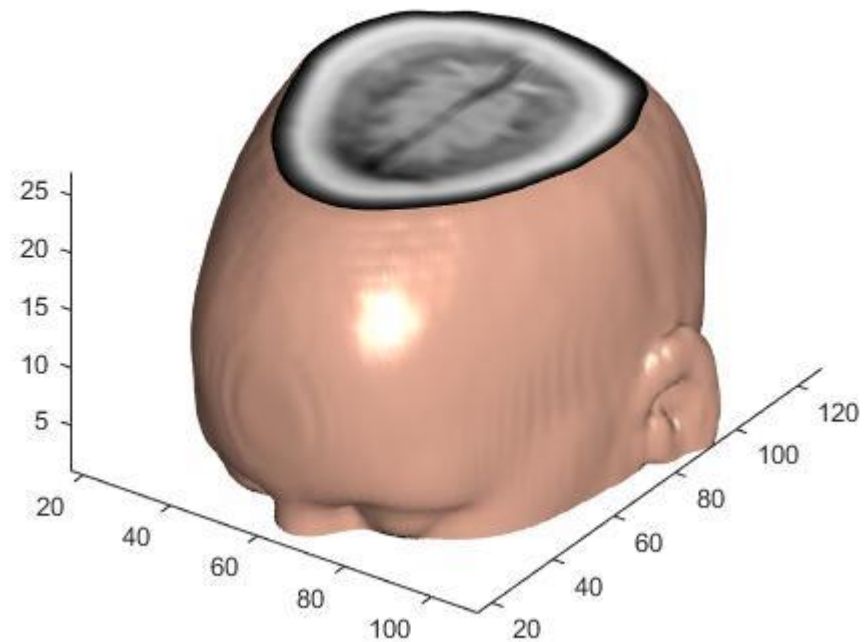
```
% Applying an Isosurface to the MRI Data
figure
%D(:,1:40,:) = [];
colormap(map)
Ds = smooth3(D);
hiso = patch(isosurface(Ds,5),...
    'FaceColor',[1,.75,.65],...
    'EdgeColor','none');
    isonormals(Ds,hiso)

hcap = patch(isocaps(Ds,5),...
    'FaceColor','interp',...
    'EdgeColor','none');
% Defining the View
view(35,30)
axis tight
daspect([1,1,.4])
% Add Lighting
lightangle(45,30);
lighting gouraud
hcap.AmbientStrength = 0.6;
hiso.SpecularColorReflectance = 0;
hiso.SpecularExponent = 50;
```


20
40
60
80
100
120


20
40
60
80
100
120

mri.m


25
20
15
10
5
120
100
80
60
40
20
20
40
60
80
100

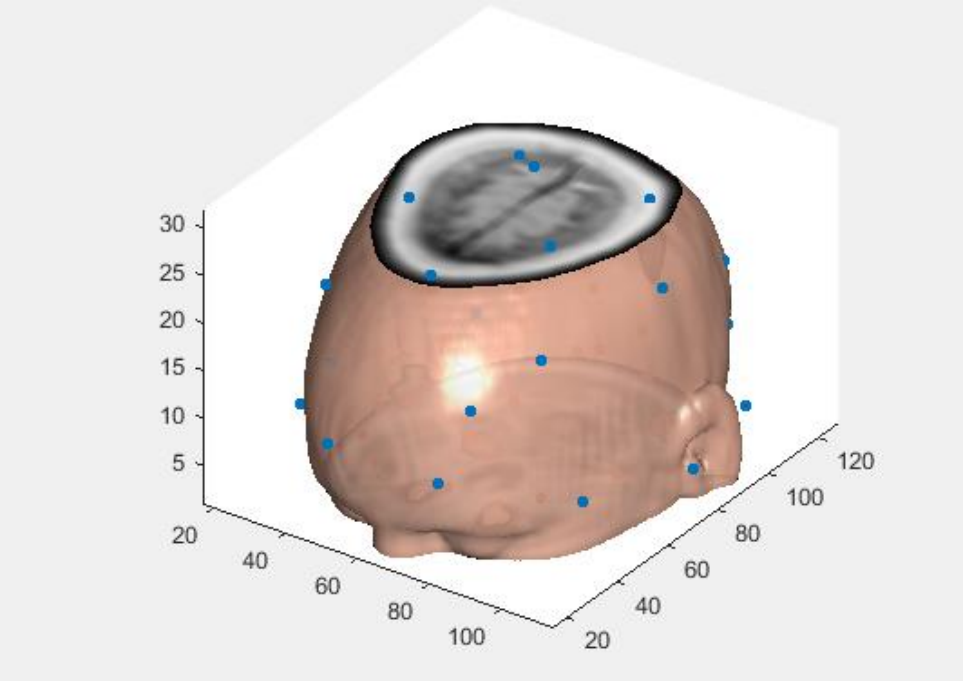

25
20
15
10
5
20
40
60
80
100
120
100
80
60
40
20

EEG_1.m

```
load('Electrode_coords.txt', '-ascii');
load('Flower_105.txt', '-ascii');
load('Flower_421.txt', '-ascii');
load('Flower_1000.txt', '-ascii');
load('Grey_105.txt', '-ascii');
load('Grey_421.txt', '-ascii');
load('Grey_1000.txt', '-ascii');
mm = size(Electrode_coords,1);
bmin = min(Electrode_coords,[],1);
bmax = max(Electrode_coords,[],1);
rhs = [ones(mm,1)]; % right hand side vector (=1)
% Map Pasqua electrode to MRI head image
El_delta=bmax-bmin;
MRI_min=[20,15,1];
MRI_max=[104,128,27];
MRI_delta=MRI_max-MRI_min;
MRI_to_El=MRI_delta./El_delta;
MRI_to_El_offset=MRI_min-bmin;
El_coords_mapped(:,1)=(Electrode_coords(:,1)-bmin(1))*MRI_to_El(1)+MRI_min(1);
El_coords_mapped(:,2)=(Electrode_coords(:,2)-bmin(2))*MRI_to_El(2)+MRI_min(2);
El_coords_mapped(:,3)=(Electrode_coords(:,3)-bmin(3))*MRI_to_El(3)+MRI_min(3);
%
x=Electrode_coords(:,1);
y=Electrode_coords(:,2);
z=Electrode_coords(:,3);
x_mapped=1.1*El_coords_mapped(:,1);
y_mapped=El_coords_mapped(:,2);
z_mapped=El_coords_mapped(:,3)+5;
plot3(x_mapped,y_mapped,z_mapped,'.','markersize',14)
hold on
```

```
% Measured potentials inside cortex
k = 0.7; % Points Inside Cortex relative to electrodes in Pasqua coords
x_cortexP = k * x;
y_cortexP = k * y;
z_cortexP = k * z;
cortexP = [x_cortexP, y_cortexP, z_cortexP];
x_cortex=(cortexP(:,1)-bmin(1))*MRI_to_El(1)+MRI_min(1);
y_cortex=(cortexP(:,2)-bmin(2))*MRI_to_El(2)+MRI_min(2);
z_cortex=(cortexP(:,3)-bmin(3))*MRI_to_El(3)+MRI_min(3);
DM_data = DistanceMatrix(Electrode_coords,cortexP);
IM = 1./DM_data; % collocation matrix of 3D fundamental solutions (1/r)
  T = inv(IM); % inverse of collocation matrix
  Pf = T*Flower_421'; % strength of internal sources
  r = rcond(T); % how good the solution is

  Check=IM*Pf; % Check equals to Flower_421
%Normalize data
Pf_min=min(Pf);
Pf_max=max(Pf);
Pf_delta=Pf_max-Pf_min;
Pf_normalized=(Pf-Pf_min)./Pf_delta;
Flower_min=min(Flower_421);
Flower_max=max(Flower_421);
Flower_delta=Flower_max-Flower_min;
Flower_421_normalized=(Flower_421-Flower_min)./Flower_delta;
%Plot of points (electrodes and cortex) and interpolated head surface
plot3(x_cortex,y_cortex,z_cortex,'.','markersize',14)
hold on
axis([bmin(1) bmax(1) bmin(2) bmax(2) bmin(3) bmax(3)])
% Surface out of the Electrode_coords
xlin = linspace(min(x),max(x),33);
ylin = linspace(min(y),max(y),33);
[X,Y] = meshgrid(xlin,ylin);
f = scatteredInterpolant(x,y,z, 'linear','linear');
Z = f(X,Y);
```

EEG_1.m – Model with electrodes and sources inside



EEGLab

- Developed at UCSD
- Used widely
- MATLAB toolbox
- To be tried in this project