STUDY analysis

Task 1

Cluster ERP image (IC polarity)

Task 2

Cluster cross coherence

Task 3

Dipole density

Exercise...

STUDY analysis

Task 1

Cluster ERP image (IC polarity)

Task 2

Cluster cross coherence

Task 3

Dipole density

Exercise...

Cluster ERP image

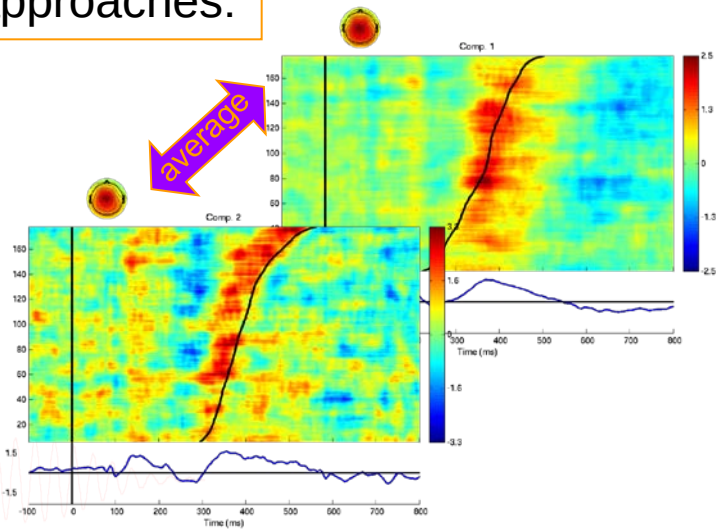
Purpose of ERP image:

- Observe single-trial dynamics of an IC activation (or power)

Purpose of *CLUSTER* ERP image:

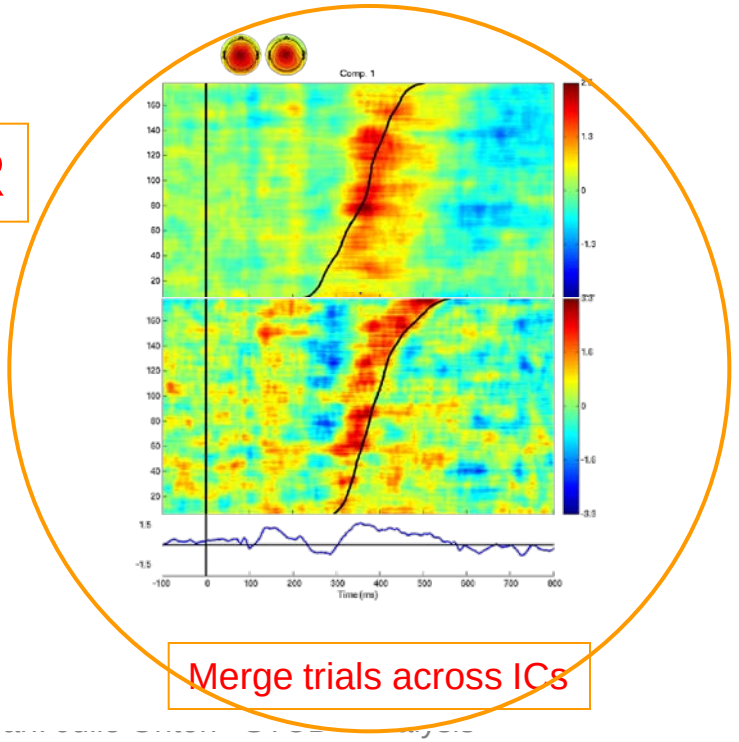
- Observe single-trial dynamics of multiple *matched* ICs from several subjects

Two approaches:



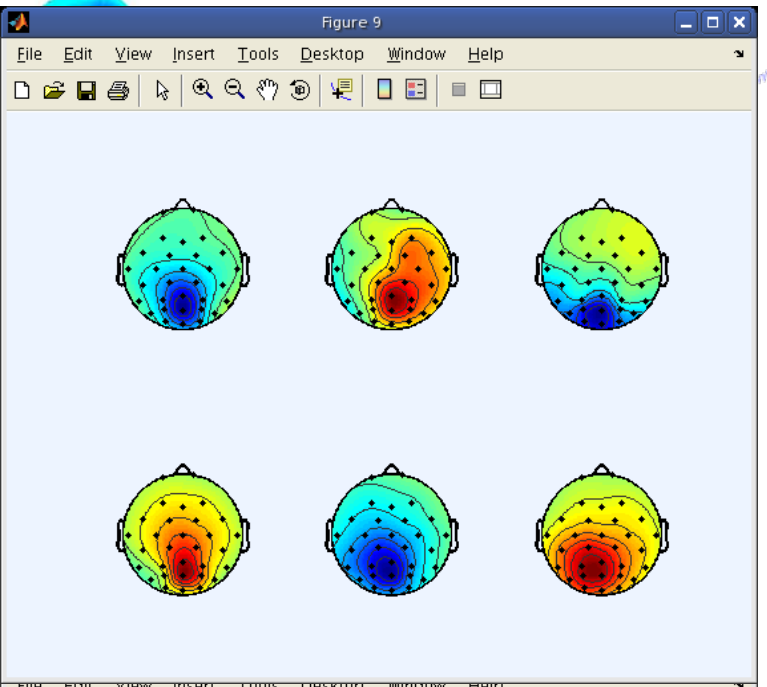
Average ERP images across ICs

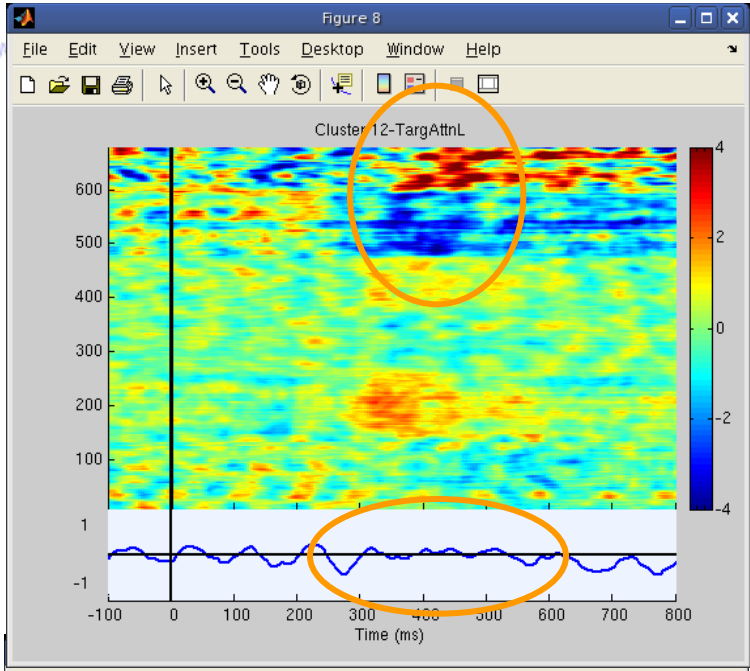
OR



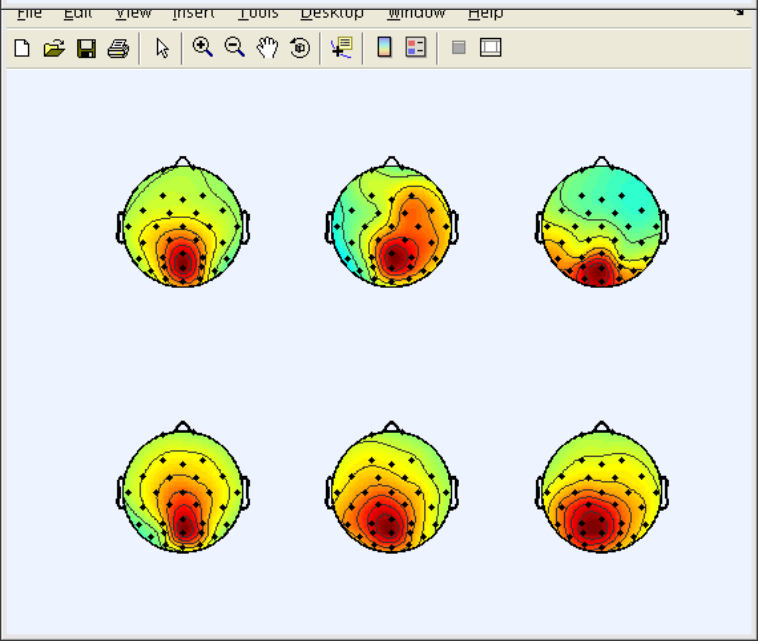
Merge trials across ICs

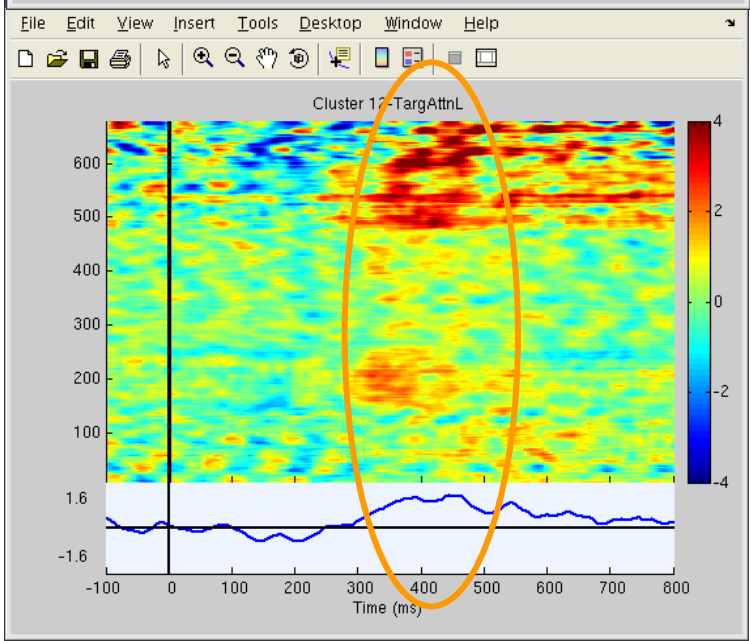
Cluster ERP image: match polarity





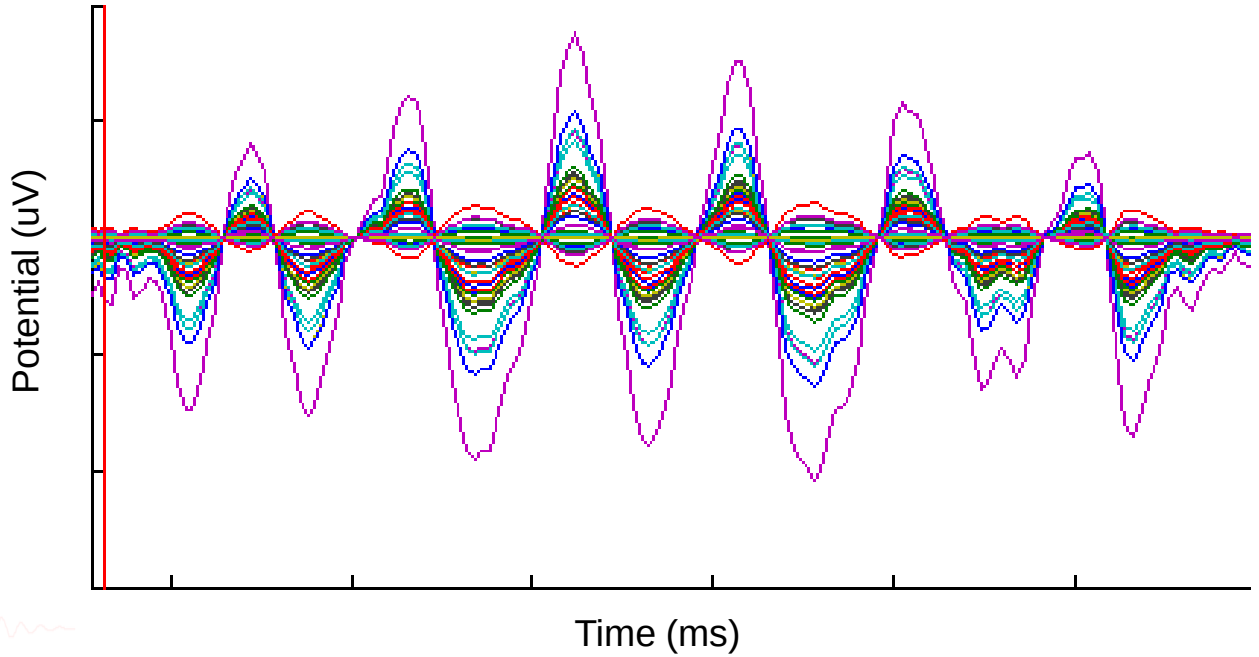
reversed polarities reflect mismatched scalp maps



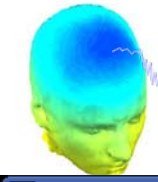


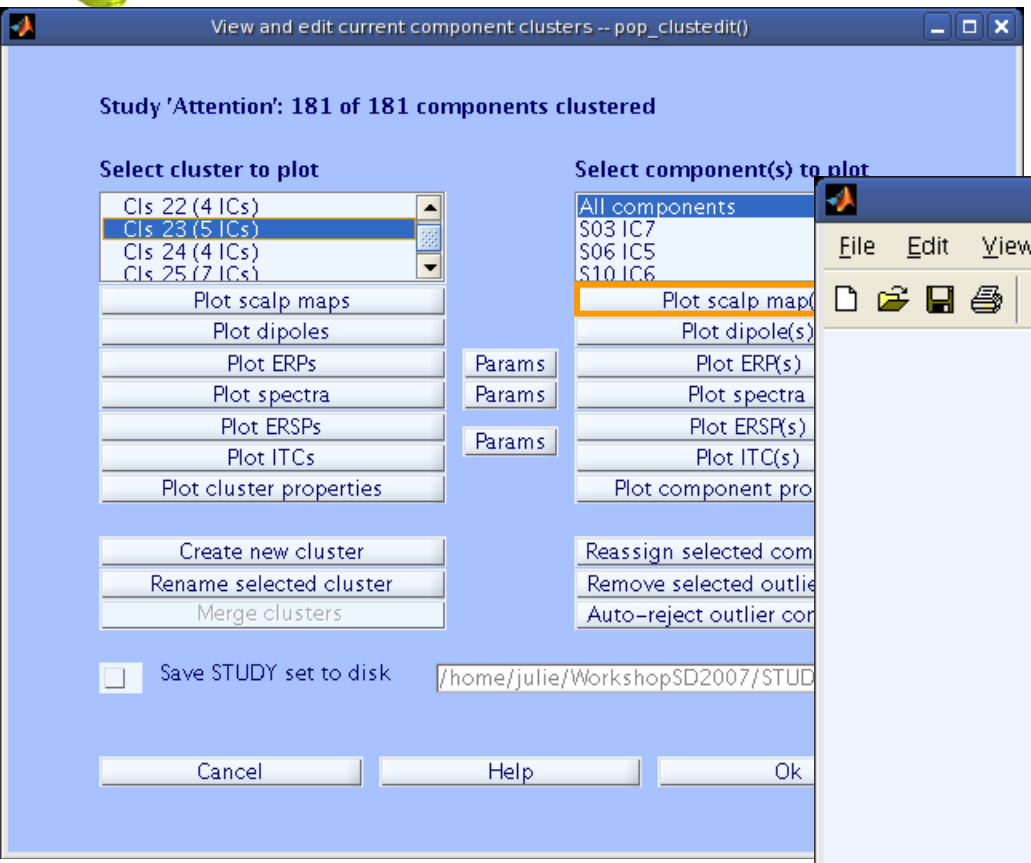
reorienting maps and activations gives a more coherent picture

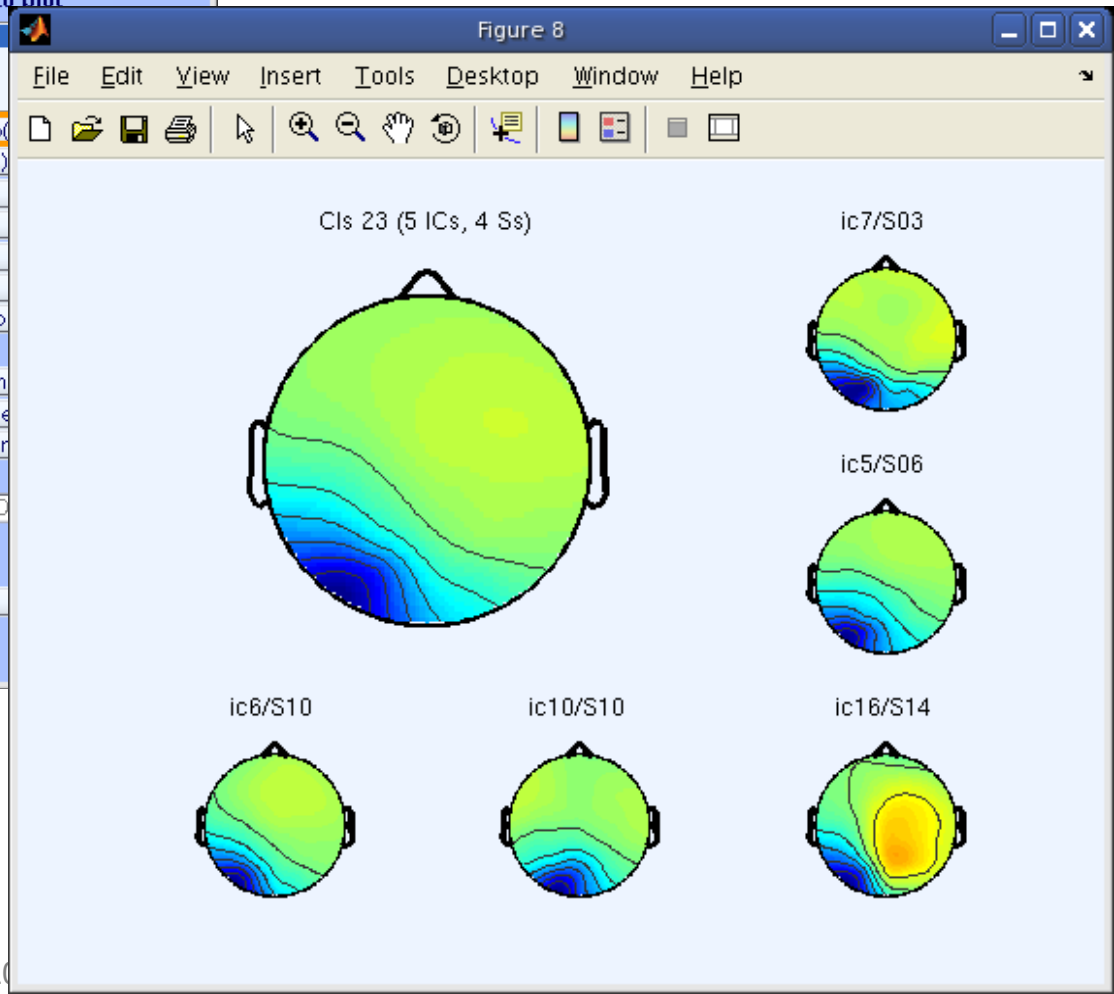
Movie of IC scalp map over time



Matching activation polarity



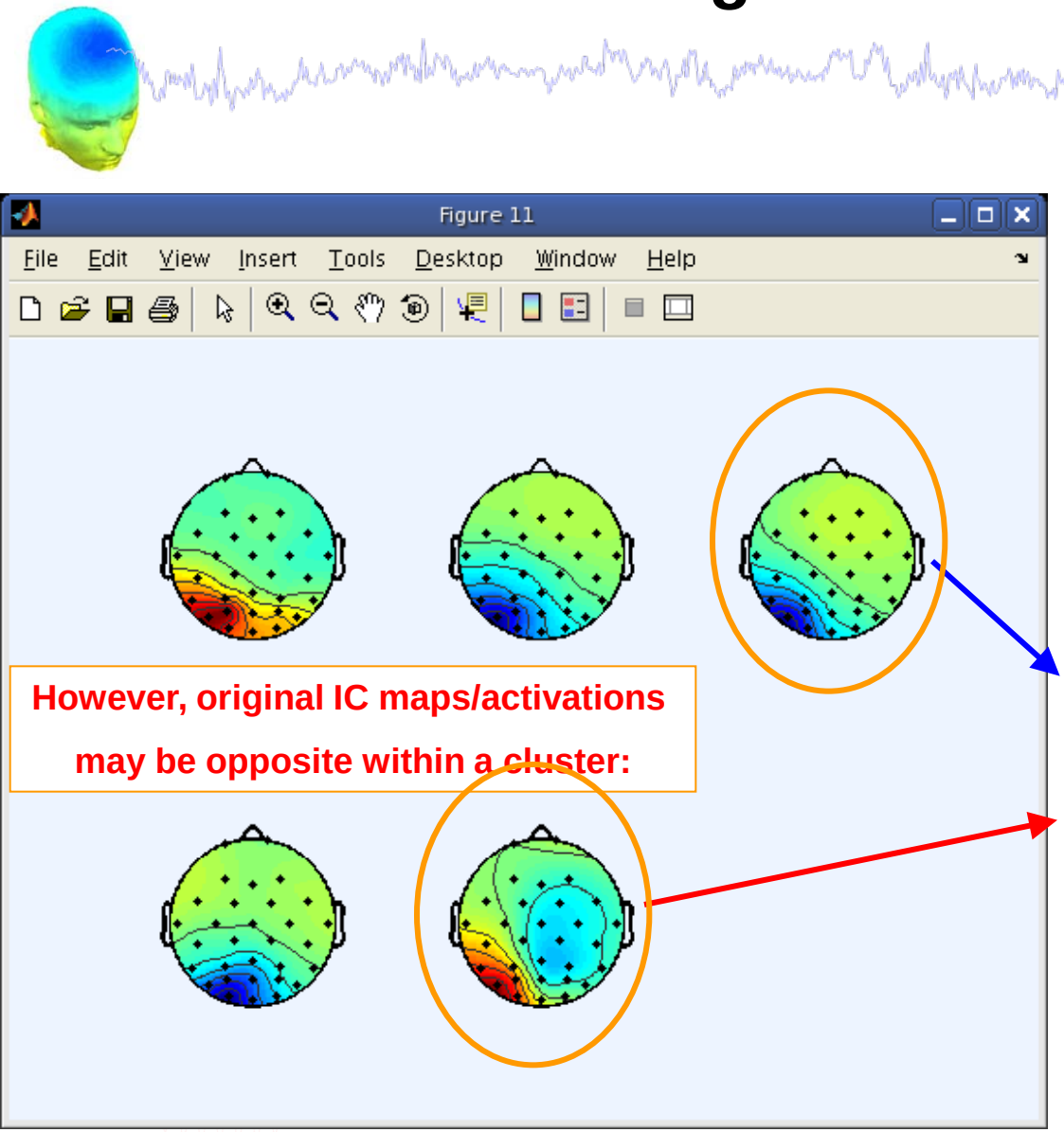




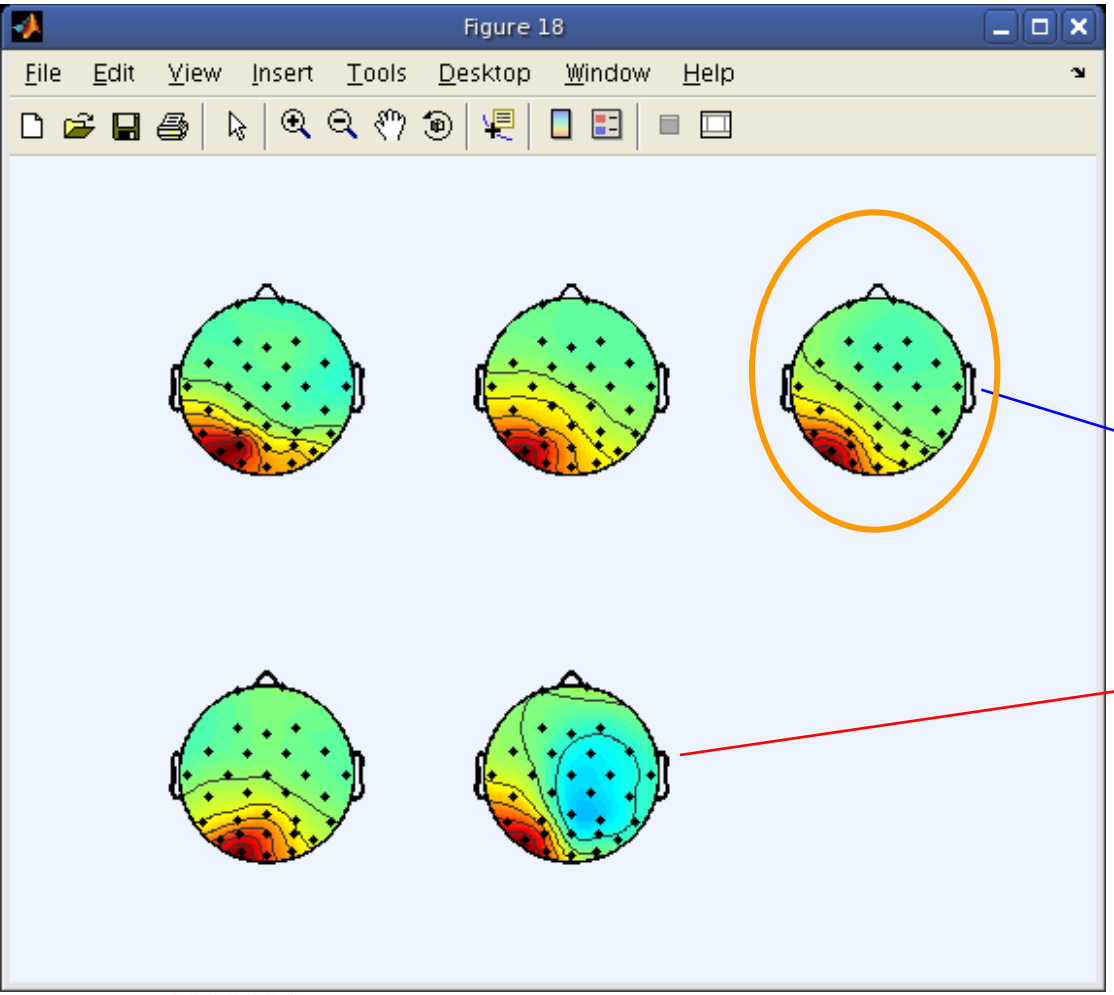
EEGLAB STUDY

matches polarities for you

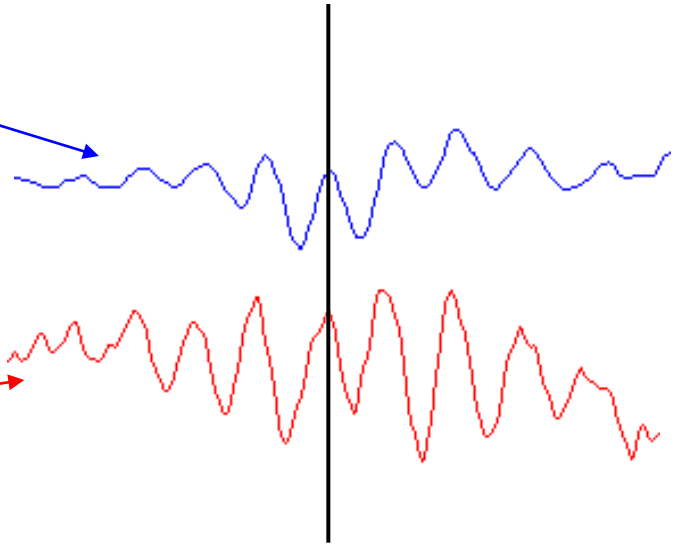
Matching activation polarity



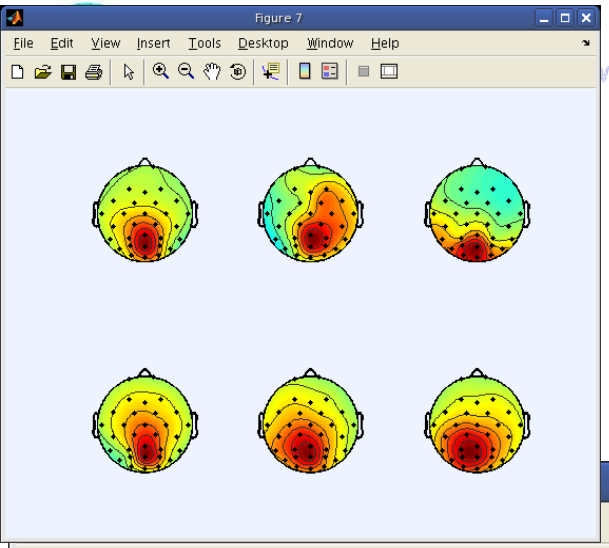
Matching activation polarity

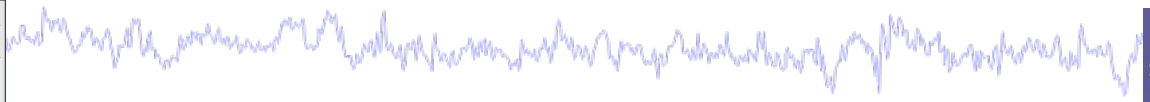


Reorient map AND activation of one IC to align



Cluster ERP image: RT sort



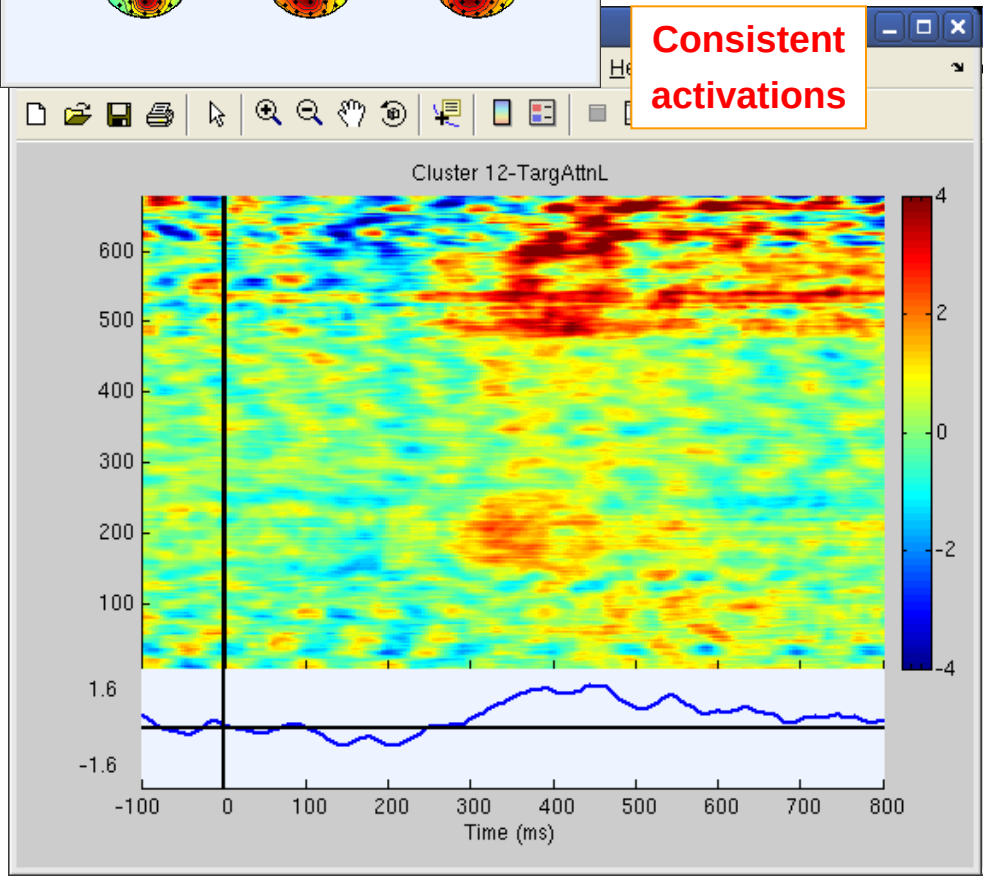


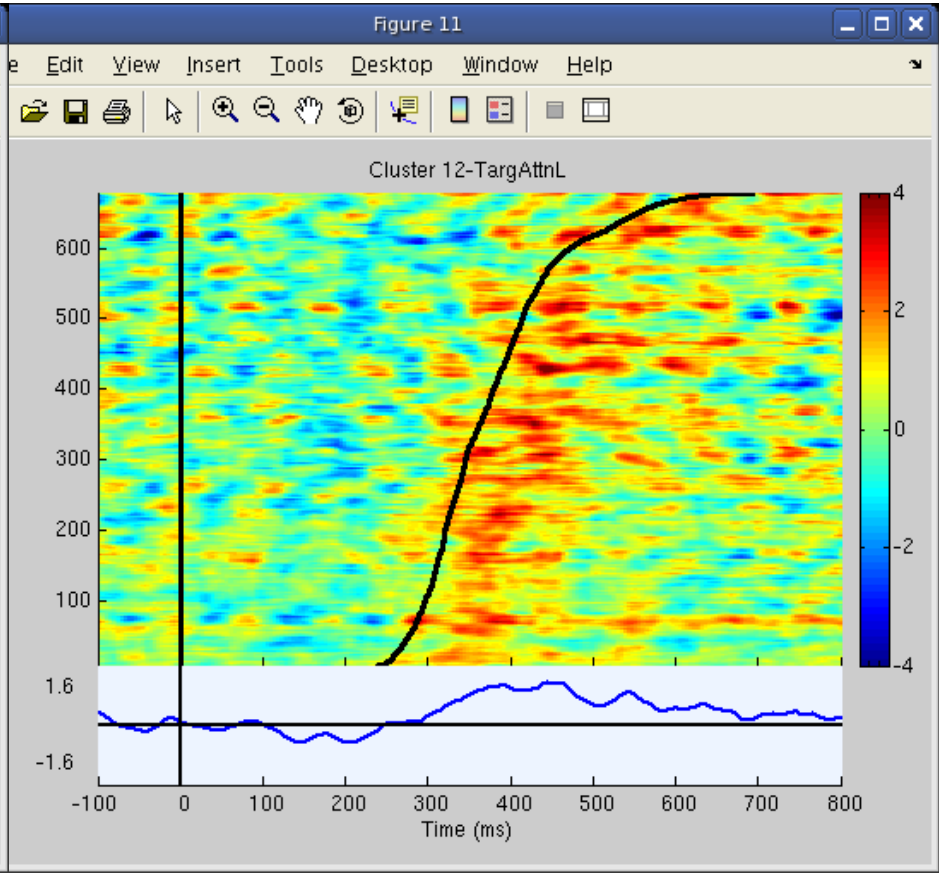
Swartz Center for Computational Neuroscience

Consistent scalp maps

Consistent activations

Sort cluster ERP image by response time





STUDY ERP image

```
 % plot all mean maps to get topo polarity:
STUDY = std_topoplot(STUDY,ALLEEG,'clusters',[2:length(STUDY.cluster)]);

clust = 7;  % choose a cluster
cond = 3; % choose a condition (from STUDY.condition)

(% requires memory options set to pre-calculate ica activations)
```

STUDY ERP image

```
% collect activations (correctly oriented) for all cluster ICs:
CURRENTSTUDY = 1; EEG = ALLEEG; CURRENTSET = [1:length(EEG)];
rts = []; clear winvs acts chans
for ic = 1:length(STUDY.cluster(clust).comps)
    setidx = STUDY.cluster(clust).sets(cond,ic);
    comp = STUDY.cluster(clust).comps(ic);
    [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET,…
    'retrieve',setidx,'study',CURRENTSTUDY);
    for ep = 1:length(EEG.epoch)
        pos = find(ismember(EEG.epoch(ep).eventtype,'rt'));
        if ~isempty(pos)
            rts = [rts EEG.epoch(ep).eventlatency{pos}];
        else
            rts = [rts 0]; % if no rt, make rt 0 ms
        end;
    end;
    rmsuv = sqrt(mean(ALLEEG(setidx).icawinv(:,comp).^2));% RMS at scalp
    winvs{ic} = ALLEEG(setidx).icawinv(:,comp)*STUDY.cluster(clust).topopol(ic);
    acts{ic} = ALLEEG(setidx).icaact(comp,:,:)*rmsuv*STUDY.cluster(clust).topopol(ic);
    chans{ic} = ALLEEG(setidx).chanlocs;
end;
```

STUDY ERP image

```
allacts = zeros(1,length(ALLEEG(1).times),0);
for ic = 1:length(STUDY.cluster(clust).comps)
    allacts(:,:,end+1:end+size(acts{ic},3)) = acts{ic};
end;
allacts = squeeze(allacts); % makes a frames x trials matrix
if strcmp(sortby,'rt')
    sortvar = rts;  % reaction times
else
    sortvar = ones(1,size(allacts,2)); % no sort
end
% PLOT activations:--------------
figure;[outdata,outvar,outtrials,limits,axhndls,erp,amps,cohers,cohs
ig,ampsig,outamps,phsangls,phsamp,sortidx,erpsig] = …
erpimage( allacts, sortvar, linspace(EEG.xmin*1000, EEG.xmax*1000,
EEG.pnts), ['Cluster ',int2str(clust),'-',condttl], smoothby, 1
,'yerplabel','','erp','limits',[tmlims NaN NaN NaN NaN NaN NaN]
,'cbar','caxis',[-ACTcolrlim ACTcolrlim],'coher', [9 12 .01]);
```

STUDY analysis

Task 1

Cluster ERP image (IC polarity)

Task 2

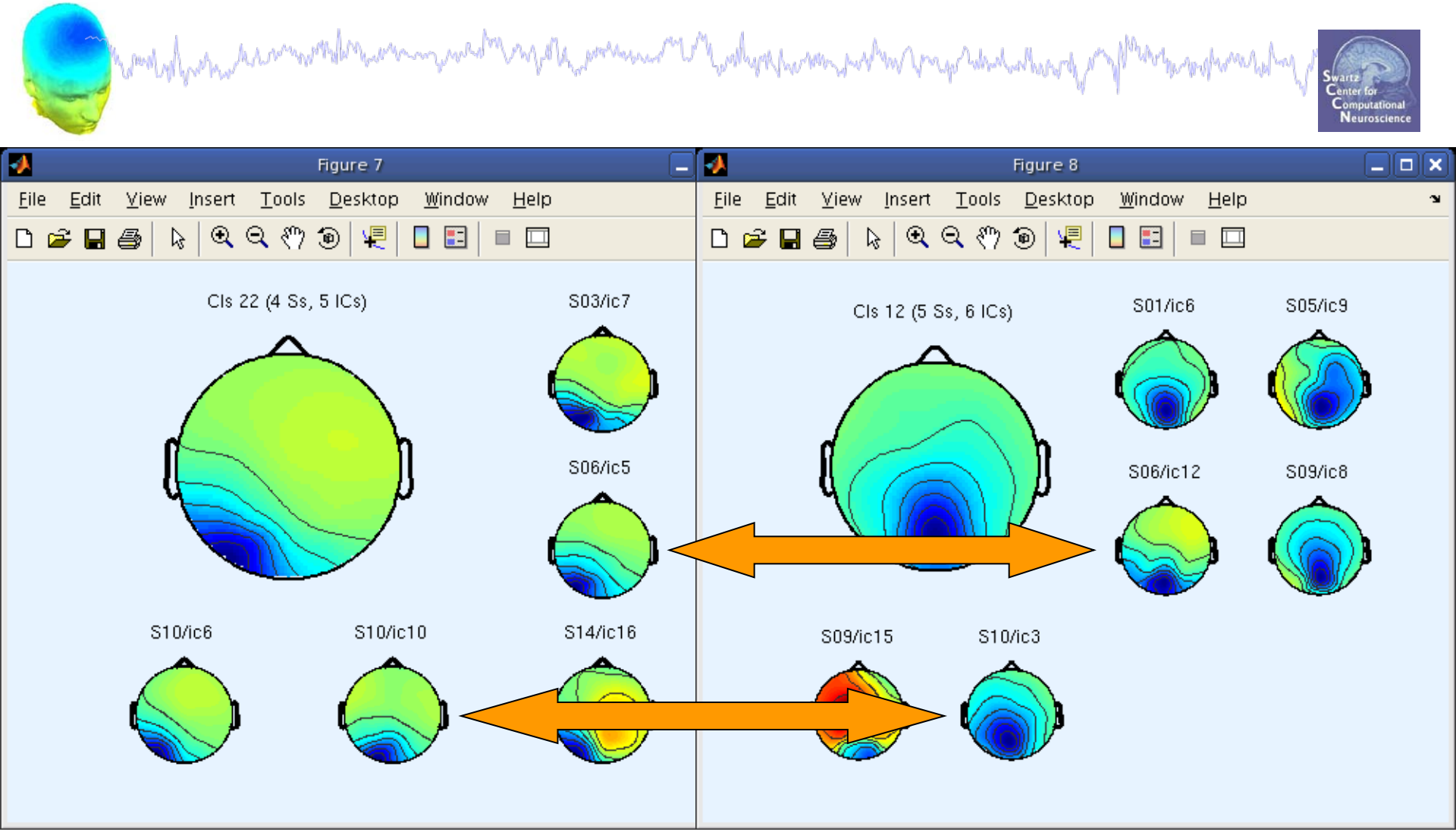
Cluster cross coherence

Task 3

Dipole density

Exercise...

Cross coherence between clusters



cross coherence between clusters
requires 2 clusters with common subjects

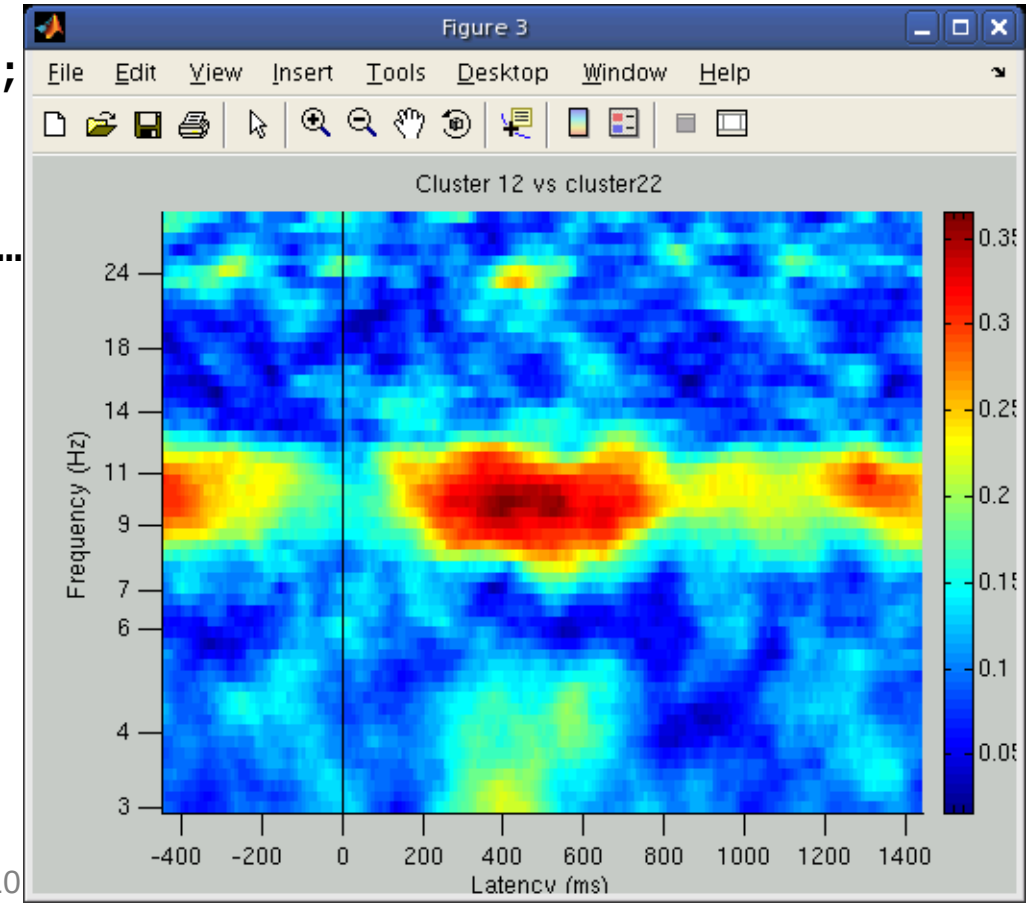
Cross coherence between clusters

```
figure;
imagesclogy(times,freqs,mean(coh,3));
set(gca,'ydir','norm');hold on;
plot([0 0],[get(gca,'ylim')],'k-');

title(['Cluster ',…
int2str(clusts(1)),' vs cluster ',…
int2str(clusts(2))]);

% include a colorbar
% for coh values:
cbar;
```

Gives average phase coherence between members of two different IC clusters



STUDY analysis

Task 1

Cluster ERP image (IC polarity)

Task 2

Cluster cross coherence

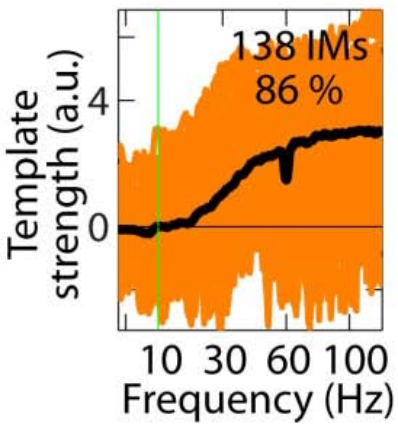
Task 3

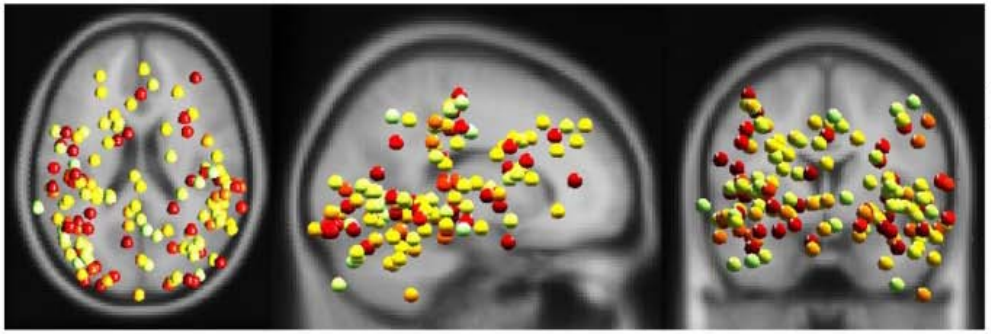
Dipole density

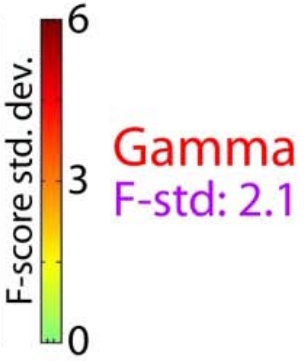
Exercise...

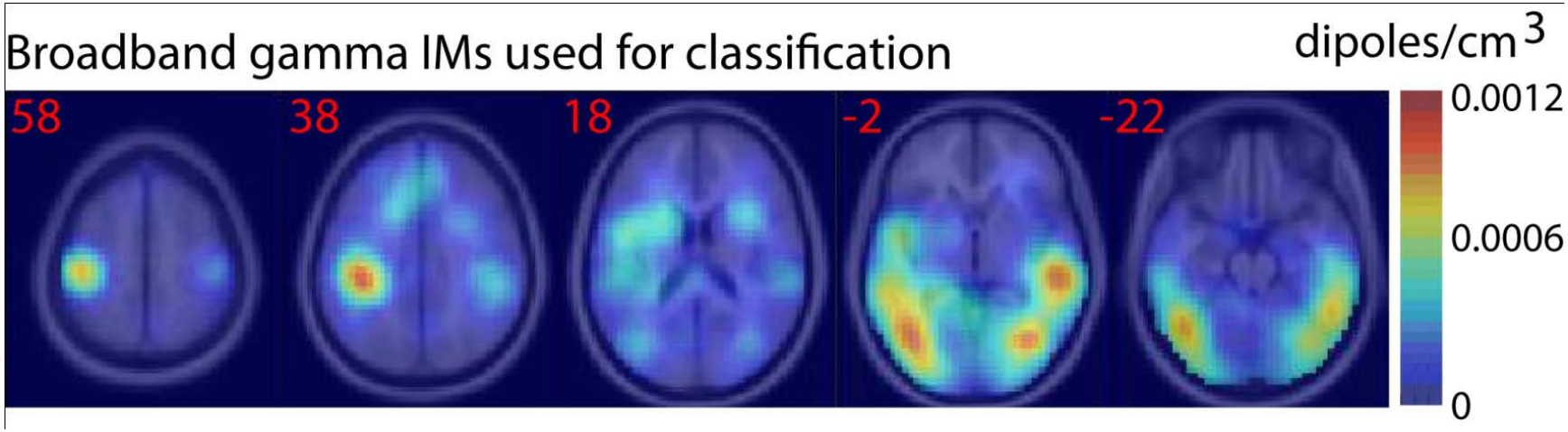
Dipole density plotting

PURPOSE: to visualize distributions of dipoles in 'MRI-esque' way





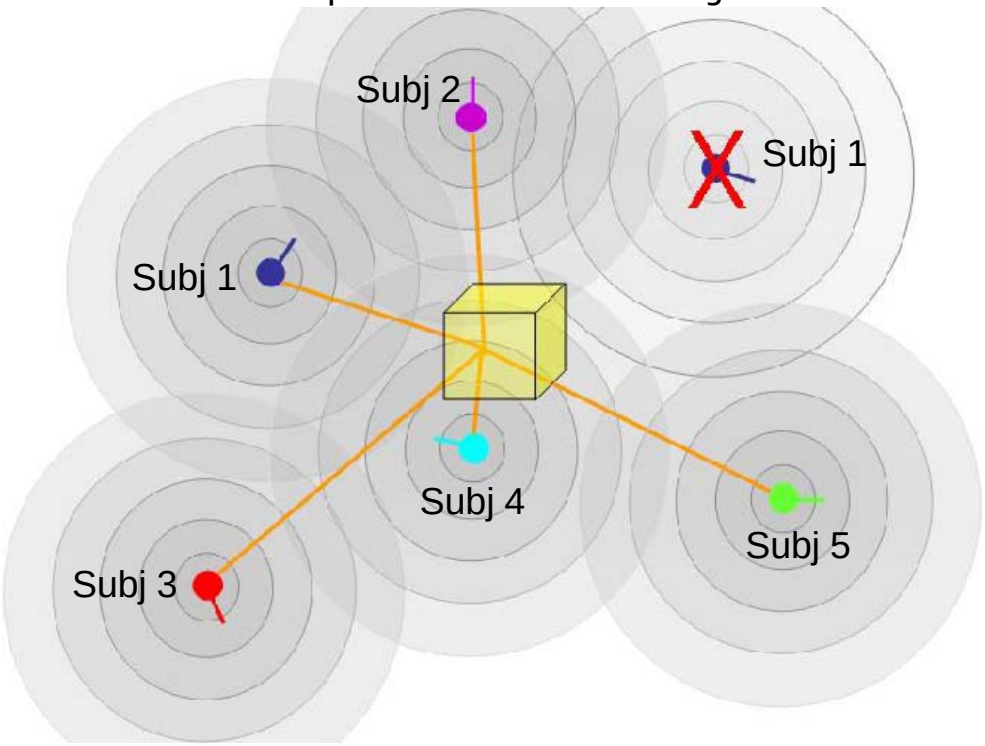




Dipole density plotting

```
'method'   - ['alldistance'|'distance'|'entropy'|'relentropy
             'alldistance' - {default} takes into account the gaussian-weighted
                        distances from each voxel to all the dipoles. See
                        'methodparam' (below) to specify a standard deviation
                        (in mm) for the gaussian weight kernel.
             'distance' - takes into account only the distances to the nearest
                          dipole for each subject. See 'methodparam' (below).
```

Explanation of 'method' argument ('distance')



Dipole density plotting – commandline only

```
cond = 1;   clust = 3;
dipsources = struct('posxyz',[],'momxyz',[],'rv',[]);   n = 1;
nowidx = 0; % initialize
for ic = 1:length(STUDY.cluster(clust).comps)
  setidx = STUDY.cluster(clust).sets(cond,ic);
  comp = STUDY.cluster(clust).comps(ic);
  if setidx ~= nowidx % don't call in if already active
    [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET, …
      'retrieve',setidx, 'study',CURRENTSTUDY);  nowidx = setidx;
  end;
  model = EEG.dipfit.coordformat;
  dipsources(1,n).posxyz = EEG.dipfit.model(comp).posxyz;
  dipsources(1,n).momxyz = EEG.dipfit.model(comp).momxyz;
  dipsources(1,n).rv = EEG.dipfit.model(comp).rv;  n = n + 1;
end;
dipoledensity(dipsources , 'method','alldistance','methodparam',10,...
   'coordformat',model);
```

Exercise

- **ALL**
  - Load workshop STUDY

- **Advanced**

  - Plot ERP image for a cluster sorting for response time (Probe).

  - Try a dipole density plot for one or more clusters of interest
    - try plotting different MRI slices to better view cluster densities

** All scripts for Intermediate/Advanced exercises can be found in …/Scripts/Tutorial_9_STUDYanalysis_II.m

STUDY cross coherence

```
clust1 = 6;

clust2 = 23;

% Crossf parameters:-------------------------------

type = 'phasecoher';

alpha = .01;

cycles = [3 0.5]; % wavelet cycles

freqscale = 'log';

frqlim = [3 30]; % calculation frequency limits in Hz

tmlims = [-100 1000]; % [min max] times in ms for window

cond = 1; % memorize only
```

STUDY cross coherence

```
allrts = cell(1,0); p=1;
for ic = 1:length(STUDY.cluster(clust1).comps)
    setidx = STUDY.cluster(clust1).sets(cond,ic);
    comp1 = STUDY.cluster(clust1).comps(ic);
    [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, CURRENTSET,…
    'retrieve',setidx, 'study',CURRENTSTUDY);
    subjmatch = find(STUDY.cluster(clust2).sets(cond,:) == setidx);
    for c = 1:length(subjmatch)
        comp2 = STUDY.cluster(clust2).comps(subjmatch(c));
        [coh(:,:,p),mcoh,times,freqs,cohboot(:,p),cohang(:,:,p)] = …
            newcrossf(EEG.icaact(comp1,:) ,EEG.icaact(comp2,:),…
            EEG.pnts,[EEG.xmin*1000 EEG.xmax*1000], EEG.srate, cycles,…
            'alpha', alpha,'winsize',EEG.srate,'newfig','off',…
            'type',type,'freqs',frqlim,'freqscale' ,freqscale,…
            'savecoher',0 , 'plotamp' ,'off','plotphase' ,'off' );
        p = p+1;
    end;
end;
```