Introduction to the EEG toolbox
Joshua Jacobs

If using UNIX-based systems, the connection to the rhino file serve r using can be initiated through the terminal using the following command:

```
ssh user@rhino.psych.upenn.edu -X
```

To use macfuse and macfusion on snow leopard to mount the rhino file systems on your local machine:

1, download macfuse and macfusion at the following url:
http://code.google.com/p/macfuse/downloads/list
http://www.macfusionapp.org/
2 install them and quit macfusion
3 open System Preferences and then open the MacFUSE pane. Check the "Show Beta Versions" box and click "Check For Updates". Go ahead and update MacFUSE.
4,open up a terminal and do the following: rm /Applications/Macfusion.app/Contents/PlugIns/sshfs.mfplugin/Contents/Resources/sshnodelay.so
you may need to change to address to where macfusion is installed.In this case it is Applications.
5 open macfusion and input the username password host. Click on mount and now you got it!

We use two different file libraries, each of which has a different function:

1. iEEGDatabase, used for getting lists of participants who performed in each task, and their meta data.
2. EEG toolbox, used for manipulating data from each patient.

(Each of these libraries can be obtained from SVN.)

To setup SVN for use on the rhino server directly, add these lines to your .bashrc file in your home directory:

```
export SVNROOT=file://localhost/home/svn
export SVN_EDITOR="emacs -nw"
```

To setup SVN for use on a different machine, add these lines to your .profile file:

```
export SVNROOT=svn+ssh://USERNAME@rhino.psych.upenn.edu/home/svn/
export SVN_EDITOR="emacs -nw"
```

(replace “USERNAME” with your actual rhino username)

> Note: When running SVN using this procedure to access rhino remotely, it will ask you for your rhino password. You can follow this procedure to have it automatically log in and eliminate the need for typing your password:
> http://www.gotidea.net/Linux/sshowto.htmlGrieb01a

then run to get a copy of each of these directories

```
svn checkout $SVNROOT/eeg/iEEGDataBase
svn checkout $SVNROOT/eeg/eeg_toolbox/trunk eeg_toolbox
```

Finally, add the directory iEEGDataBase (and its subdirectories) and the directory eeg_tolbox/trunk (and its subdirectories) to your matlab path. (Type the command `matlab` to run matlab.)

Using the iEEGDatabase to get a list of patients who performed in the Sternberg working-memory task:

```
subs=get_subs('pymms');
```

You can also get subjects who only ran in a particular variant of the Sternberg task. This example provides the subset of the subjects that ran in a version of the task that has a repeated item:

```
subs=get_subs('pymms','hasRepeat');
```

You can use `get_sub_expInfo` to get a list of all the queryable properties of the experiment:

```
expProperties=get_sub_expInfo('pymms','TJ030');
```

Or, use ‘pyFR’ to get the list of patients who performed in the py Free Recall task:

```
subs=get_subs('pyFR');
```

Get data on an individual subject:

```
sub='UP017';
[events,eventDetails]=get_sub_events('pymms',sub);
```

```
events =

1x3240 struct array with fields:
    eegfile
    eegoffset
    subject
    mode
    rt
    type
    item
    istarget
    listlen
    probe_pos
    cueitem
    correct
    artifacts
    mstime
    session
    trial



eventDetails =

         leads: [1x57 double]
      leadNames: {57x1 cell}
    sampleRate: 512
     noiseFreq: 60
           tal: [1x57 struct]
```

eventDetails contains the information about the subject overall:

- leads= a list of all the electrodes in this patient
- leadNames=the 'name' of each electrode as designated by the clinicians
- sampleRate=the sampling frequency of the recording
- noiseFreq=the frequency of power noise (60hz in america, 50Hz in europe)
- tal=information about the specific location of each recording electrode.

events gives information about each occurrence in the task:

- eegfile = the filename where the raw eeg for this event is stored
- eegoffset = the sample offset in this file that corresponds to the timepoint of the event
- subject = the name of the subject (first two characters are the hospital)
- mode = font/case manipulation (only for some versions of the task)
- rt = reaction time in responding in that trial
- type = type of event.
  - PRES1 = presentation of the first item in the list (likewise for PRES2)
  - CUE = presentation cue (or probe) item
  - RESP = the subject presses a key to respond
- item = the actual item that was presented
- istarget = whether the cue item in that list did or did not match a studied item (1 or 0, respectively)
- listlen = length of the presented list
- probe_pos = position of the cue in the presented list
- cueitem= name of the cue item in the presented list
- correct=whether the patient responded correctly in the trial
- artifact = whether there was an EEG artifact in near this event (this is generally blank...)
- mstime = time in milliseconds at onset of event
- session = number of the session where this event took place
- trial = trial number within this session

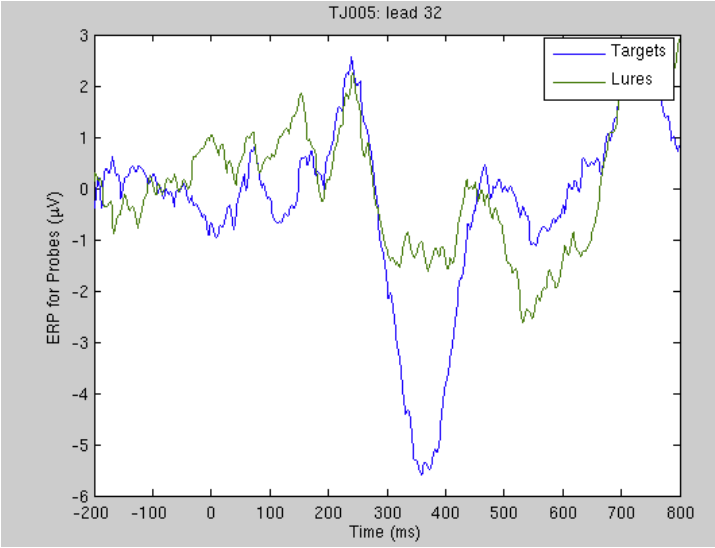
Using events for neural data analysis with the EEG toolbox:

Common functions:

- filterStruct - used for picking out the specific events that you want
- gete_ms - gets the raw EEG data at the timepoint of each event
- gethilbertphase- gets the phase and amplitude of the EEG signal in a particular prespecified EEG band
- getphasepow - does wavelet power spectrum analyses

Calculate an event-related potential (ERP) for target and lure cue items using gete_ms:

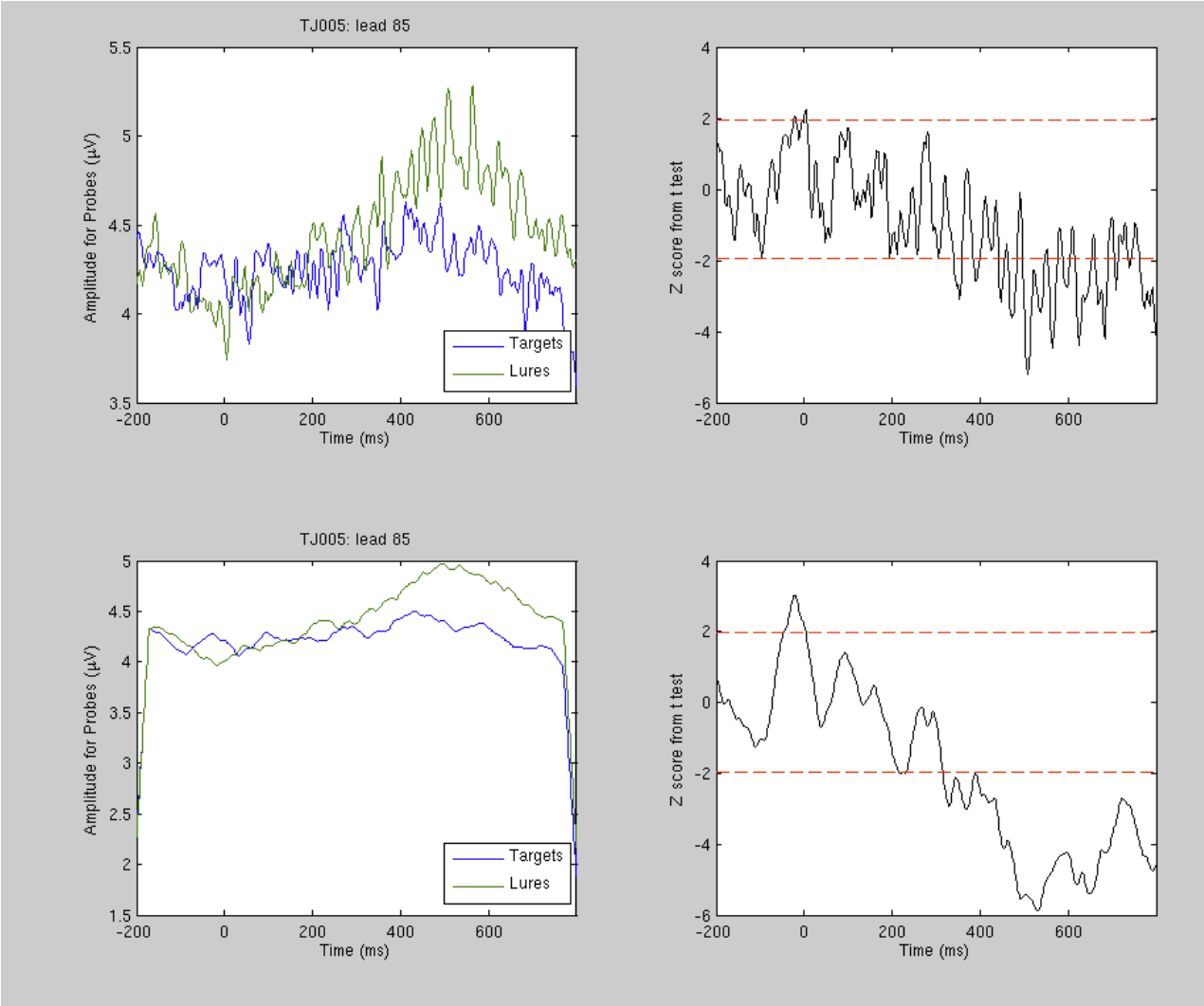
```
sub='TJ005';
lead=32
duration=1000;offset=-200;buffer=200; resamp=500;
times=offset:(1000/resamp):offset+duration-1;

[events,eventDetails]=get_sub_events('pymms',sub);

targs=filterStruct(events,'strcmp(type,''CUE'')&istarget==1');
lures=filterStruct(events,'strcmp(type,''CUE'')&istarget==0');

targ_d=gete_ms(lead,targs,duration,offset,buffer,[59 ],'low',1,resamp,[offset 0]);
lure_d=gete_ms(lead,lures,duration,offset,buffer,[59 ],'low',1,resamp,[offset 0]);

targ_m=mean(targ_d);
lure_m=mean(lure_d);

plot(times,targ_m,times,lure_m);
ylabel('ERP for Probes (\muV)'); xlabel('Time (ms)');
title(sprintf('%s: lead %d',sub,lead));
legend('Targets','Lures');
```



Calculating EEG amplitude using the bandpass filtering and the Hilbert transform. The amplitude of the Hilbert transform is calculated for the band between 40 and 100Hz.

```
sub='TJ005';  lead=85;

duration=1000;offset=-200;buffer=200; resamp=500;
times=offset:(1000/resamp):offset+duration-1;

[events,eventDetails]=get_sub_events('pymms',sub);

targs=filterStruct(events,'strcmp(type,''CUE'')&istarget==1');
lures=filterStruct(events,'strcmp(type,''CUE'')&istarget==0');

band=[40 100];
[~,targ_amp]=gethilbertphase(lead,targs,duration,offset,buffer,band,eventDetails.noiseFreq,resam
p);
[~,lure_amp]=gethilbertphase(lead,lures,duration,offset,buffer,band,eventDetails.noiseFreq,resam
p);

m=2;n=2;
```

```
subplot(m,n,1);
targ_m=mean(targ_amp);lure_m=mean(lure_amp);
plot(times,targ_m,times,lure_m);
ylabel('Amplitude for Probes (\muV)'); xlabel('Time (ms)');
title(sprintf('%s: lead %d',sub,lead));
legend('Targets','Lures','Location','SouthEast');
xlim(times([1 end]));


subplot(m,n,2);
[~,p]=ttest2(targ_amp,lure_amp,.05,'left'); %goes across first dimension
plot(times,norminv(p),'k-',...
     times([1 end]),norminv(.025)+[0 0],'r--',...
     times([1 end]),-norminv(.025)+[0 0],'r--');
ylabel('Z score from t test'); xlabel('Time (ms)');
xlim(times([1 end]));

%%%%%%%%%%%%%%
% Now do all the same plots, but with smoothing
%%%%%%%%%%%%%%

windSize=30; %150 ms at a 500Hz sampling rate
targ_amp2=smoothRows(targ_amp,windSize);
lure_amp2=smoothRows(lure_amp,windSize);

subplot(m,n,3);
targ_m2=mean(targ_amp2);lure_m2=mean(lure_amp2);
plot(times,targ_m2,times,lure_m2);
ylabel('Amplitude for Probes (\muV)'); xlabel('Time (ms)');
title(sprintf('%s: lead %d',sub,lead));
legend('Targets','Lures','Location','SouthEast');
xlim(times([1 end]));

subplot(m,n,4);
[~,p2]=ttest2(targ_amp2,lure_amp2,.05,'left'); %goes across first dimension
plot(times,norminv(p2),'k-',...
     times([1 end]),norminv(.025)+[0 0],'r--',...
     times([1 end]),-norminv(.025)+[0 0],'r--');
ylabel('Z score from t test'); xlabel('Time (ms)');
xlim(times([1 end]));
```



Whereas the HIlbert transform (through gethilbertphase) looks at a specific band, the wavelet transform (through getphasepow) provides the EEG power spectra for a wide range of frequencies: it provides the EEG power spectrum in a time-frequency space.

The getphasepow function needs, as inputs, the specific frequencies and the width that will be used for the wavelet calculation. Two possible ways exists to pass this input to getphasepow:

1) directly, as inputs in the command line, e.g.:

```
[~,targ_pow]=getphasepow(lead,targs,duration,offset,buffer,'resampledrate',resamp,...
     'freqs', (2^(1/8)).^(8:56), 'width', 6);
freqs = (2^(1/8)).^(8:56);  % this is the vector of frequencies to be used in the plots
```

2) (preferred) indirectly, having getphasepow retrieve these parameters from a text file (eeganalparams.txt), e.g.:

```
[~,targ_pow]=getphasepow(lead,targs,duration,offset,buffer,'resampledrate',resamp);
freqs=eeganalparams('freqs'); % this is the vector of frequencies to be used in the plots
```

The eeganalparams.txt file needs to be placed in a folder named ‘eeg’ (path: ~/eeg/) and its contents will be, for example:

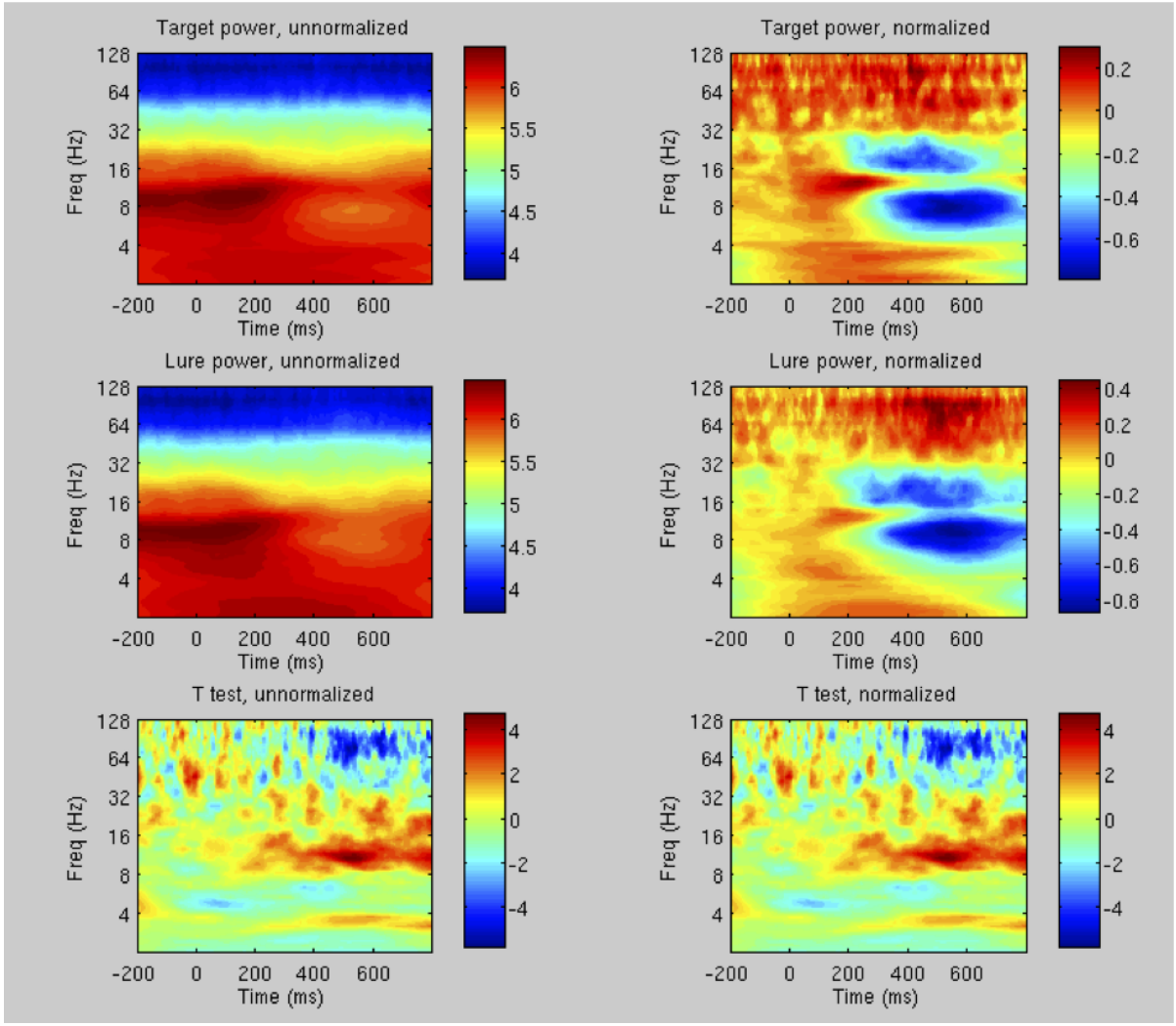
```
freqs (2^(1/8)).^(8:56)
width 6
```

Here is an example using the (preferred) second approach:

```
sub='TJ005';  lead=85;

duration=1000;offset=-200;buffer=200; resamp=500;
times=offset:(1000/resamp):offset+duration-1;
```

```
[events,eventDetails]=get_sub_events('pymms',sub);

targs=filterStruct(events,'strcmp(type,''CUE'')&istarget==1');
       lures=filterStruct(events,'strcmp(type,''CUE'')&istarget==0');

[~,targ_pow]=getphasepow(lead,targs,duration,offset,buffer,'resampledrate',resamp);
[~,lure_pow]=getphasepow(lead,lures,duration,offset,buffer,'resampledrate',resamp);
freqs=eeganalparams('freqs');

%log transform
targ_pow=log10(targ_pow);
lure_pow=log10(lure_pow);

%normalize relative to power at time zero
baseline=[targ_pow(:,:,times==0); lure_pow(:,:,times==0)];
targ_powN=bsxfun(@rdivide,bsxfun(@minus,targ_pow,mean(baseline)),std(baseline));
lure_powN=bsxfun(@rdivide,bsxfun(@minus,lure_pow,mean(baseline)),std(baseline));

m=3;n=2;
plotLevels=40;

%%%%%%%%%%%%%
%plot targets

subplot(m,n,1);
y=squeeze(mean(targ_pow,1));
contourf(times,freqs,y,plotLevels); shading flat;
%other 2d plotting commands: pcolor, imagesc
ylabel('Freq (Hz)'); xlabel('Time (ms)');
set(gca,'yscale','log','ytick',2.^[1:7]);
title('Target power, unnormalized');
colorbar;

subplot(m,n,2);
y=squeeze(mean(targ_powN,1));
contourf(times,freqs,y,plotLevels); shading flat;
ylabel('Freq (Hz)'); xlabel('Time (ms)');
set(gca,'yscale','log','ytick',2.^[1:7]);
title('Target power, normalized');
colorbar;

%%%%%%%%%%%%%
%plot lures

subplot(m,n,3);
y=squeeze(mean(lure_pow,1));
contourf(times,freqs,y,plotLevels); shading flat;
%other 2d plotting commands: pcolor, imagesc


ylabel('Freq (Hz)'); xlabel('Time (ms)');
set(gca,'yscale','log','ytick',2.^[1:7]);
title('Lure power, unnormalized');
colorbar;
```

```
subplot(m,n,4);
y=squeeze(mean(lure_powN,1));
contourf(times,freqs,y,plotLevels); shading flat;
ylabel('Freq (Hz)'); xlabel('Time (ms)');
set(gca,'yscale','log','ytick',2.^[1:7]);
title('Lure power, normalized');
colorbar;

subplot(m,n,5);
[~,p]=ttest2(targ_pow,lure_pow,.05,'left');
z=squeeze(norminv(p));
contourf(times,freqs,z,plotLevels); shading flat;
ylabel('Freq (Hz)'); xlabel('Time (ms)');
set(gca,'yscale','log','ytick',2.^[1:7]);
title('T test, unnormalized');
colorbar;

subplot(m,n,6);
z=squeeze(norminv(p));
[~,p]=ttest2(targ_powN,lure_powN,.05,'left');
contourf(times,freqs,z,plotLevels); shading flat;
ylabel('Freq (Hz)'); xlabel('Time (ms)');
set(gca,'yscale','log','ytick',2.^[1:7]);
title('T test, normalized');
colorbar;
```


Target power, unnormalized
Target power, normalized
Lure power, unnormalized
Lure power, normalized
T test, unnormalized
T test, normalized
Freq (Hz)
Time (ms)
128
64
32
16
8
4
-200
0
200
400
600
6
5.5
5
4.5
4
0.2
0
-0.2
-0.4
-0.6
0.4
-0.8
2
-2
-4