<u>Progress Report on 8 September 2015</u>

- **Cross-correlations: Rationale for subtraction of mean values**

Cross-correlation is a measure of similarity of two series as a function of the lag of one relative to the other (i.e. measuring the ***correlation*** between two series with varying lag)

Observing the ***derivation of correlation***, the most familiar measure of dependence between two quantities is the ***Pearson product-moment correlation coefficient***. It is obtained by dividing the ***covariance*** of the two variables by the product of their standard deviations.

In probability theory and statistics, ***covariance*** is a measure of how much two random variables change together. If the greater values of one variable mainly correspond with the greater values of the other variable, and the same holds for the smaller values, i.e., the variables tend to show similar behavior, the covariance is positive.

The ***covariance*** between two jointly distributed real-valued random variables X and Y with ***finite second moments*** is defined as[2]

$$\sigma(X,Y) = \mathrm{E}\left[(X - \mathrm{E}[X])(Y - \mathrm{E}[Y])\right],$$

In mathematics, a ***moment is a specific quantitative measure***, used in both mechanics and statistics, of the ***shape of a set of points***.

The n-th moment of a real-valued continuous function f(x) of a real variable about a value c is

$$\mu_n = \int_{-\infty}^{\infty} (x-c)^n \, f(x)\, dx.$$

It is possible to define moments for random variables in a more general fashion than moments for real values. The moment of a function, without further explanation, usually refers to the above expression with c = 0.

For the second and higher moments, the central moments (moments about the mean, with c being the mean) are usually used rather than the moments about zero, because they provide clearer information about the distribution's shape.

- New Function Prototypes:
  - Updated extractionscript.m

```matlab
clear

for k = 1:10 %need to include dirSize, run with steps to acquire xcorr 'r' for single channel i

    for i = 1:8

fileName1 = getFiles(2*k+1);
fileName2 = getFiles(2*k+2);

data1 = extractData(fileName1);
data2 = extractData(fileName2);

%applying band-pass filter to delta, theta, alpha, beta, gamma freq band
delta1 = Band_pass_filter(data1,1,4,128);
delta2 = Band_pass_filter(data2,1,4,128);
theta1 = Band_pass_filter(data1,4,8,128);
theta2 = Band_pass_filter(data2,4,8,128);
alpha1 = Band_pass_filter(data1,8,12,128);
alpha2 = Band_pass_filter(data2,8,12,128);
beta1 = Band_pass_filter(data1,15,30,128);
beta2 = Band_pass_filter(data2,15,30,128);
gammaU1 = Band_pass_filter(data1,30,60,128);
gammaU2 = Band_pass_filter(data2,30,60,128);
gamma1 = Notch_filter(gammaU1, 50, 128); %apply Notch filter at 50Hz due to power line noise
gamma2 = Notch_filter(gammaU2, 50, 128);

%extracting single channel data for cross-correlation
singleD1 = delta1(:,i);
singleD2 = delta2(:,i);
singleT1 = theta1(:,i);
singleT2 = theta2(:,i);
singleA1 = alpha1(:,i);
singleA2 = alpha2(:,i);
singleB1 = beta1(:,i);
singleB2 = beta2(:,i);
singleG1 = gamma1(:,i);
singleG2 = gamma2(:,i);

%compute xcorr & extract variables for each freq band
[d(:,1),d(:,2)] = xcorr(singleD1-mean(singleD1), singleD2-mean(singleD2),'coeff');
[t(:,1),t(:,2)] = xcorr(singleT1-mean(singleT1), singleT2-mean(singleT2),'coeff');
[a(:,1),a(:,2)] = xcorr(singleA1-mean(singleA1), singleA2-mean(singleA2),'coeff');
[b(:,1),b(:,2)] = xcorr(singleB1-mean(singleB1), singleB2-mean(singleB2),'coeff');
[g(:,1),g(:,2)] = xcorr(singleG1-mean(singleG1), singleG2-mean(singleG2),'coeff');

%compute max(xcorr) for each freq band
deltamax = max(d(:,1));
thetamax = max(t(:,1));
alphamax = max(a(:,1));
betamax = max(b(:,1));
gammamax = max(g(:,1)); %getting 0.1477 for all values

%computing inter-brain density (IBD) values, using algorithm from
%brain_plot.m & compcorr.m for delta only
adj = compcorr(delta1, delta2); %different synchrony measure from xcorr?
value123=sum(adj(:)==1);
result13=value123/(14^2); %rationale?

    end;
end;
```

  - getFiles.m : to load files from chosen directory

```matlab
function fileName = getFiles(k)

d=dir('/Users/Jon/Desktop/Social_Neuroscience/recordings/1v1physical');

fileName = d(k).name;
```

- Notch_filter.m : to remove power line artefacts at 50Hz

```
function EEG_signal_filt = Notch_filter(EEG_signal,Cut_off_freq,Fs)

% Function to implement notch filter using IIR filter.
%
% Usage: EEG_signal_filt = Notch_filter(EEG_signal,Cut_off_freq,Fs)
% Input:
%       EEG_signal - Referenced EEG signal
%       Cut_off_freq - cut-off frequencies in linear scale (in hertz)
%       Fs  - sampling frequency
% Output:
%       EEG_signal_filt - filtered EEG signal

%% Preamble
Nyquist_freq = Fs/2;
[Num_chan temp] = size(EEG_signal);


%% IIR notch filter of Order-2
% Normalized frequencies

Wo =  Cut_off_freq/Nyquist_freq; BW = Wo/35;
[Filter_coeffs_num Filter_coeffs_den] = iirnotch(Wo,BW);

for i=1:Num_chan,
   EEG_signal_filt(i,:) = filter(Filter_coeffs_num,Filter_coeffs_den,EEG_signal(i,:));
end
```

- Further Pre-processing Steps:
  - Sample data for reference:

30720x8 double

	1	2	3	4	5	6	7	8
1	4.2031e+03	4.3836e+03	4.1913e+03	4.4595e+03	3.9964e+03	4.3472e+03	4.5795e+03	4.4138e+03
2	4.2010e+03	4.3826e+03	4.1867e+03	4.4590e+03	3.9954e+03	4.3508e+03	4.5769e+03	4.4118e+03
3	4.2092e+03	4.3903e+03	4.1892e+03	4.4559e+03	4.0072e+03	4.3528e+03	4.5846e+03	4.4103e+03
4	4.2092e+03	4.3918e+03	4.1882e+03	4.4564e+03	4.0103e+03	4.3492e+03	4.5867e+03	4.4113e+03
5	4.1995e+03	4.3877e+03	4.1841e+03	4.4590e+03	3.9938e+03	4.3477e+03	4.5815e+03	4.4123e+03
6	4.2056e+03	4.3913e+03	4.1954e+03	4.4595e+03	3.9903e+03	4.3472e+03	4.5856e+03	4.4149e+03
7	4.2133e+03	4.3944e+03	4.2015e+03	4.4605e+03	4.0036e+03	4.3446e+03	4.5887e+03	4.4149e+03
8	4.2046e+03	4.3913e+03	4.1897e+03	4.4590e+03	4.0010e+03	4.3415e+03	4.5841e+03	4.4103e+03
9	4.2046e+03	4.3903e+03	4.1903e+03	4.4554e+03	3.9897e+03	4.3410e+03	4.5846e+03	4.4108e+03
10	4.2185e+03	4.3959e+03	4.2031e+03	4.4559e+03	3.9944e+03	4.3446e+03	4.5913e+03	4.4205e+03
11	4.2195e+03	4.4031e+03	4.2031e+03	4.4610e+03	4.0010e+03	4.3487e+03	4.5923e+03	4.4236e+03
12	4.2138e+03	4.4021e+03	4.1995e+03	4.4631e+03	3.9969e+03	4.3508e+03	4.5908e+03	4.4205e+03
13	4.2138e+03	4.3974e+03	4.2010e+03	4.4615e+03	3.9949e+03	4.3523e+03	4.5923e+03	4.4221e+03
14	4.2138e+03	4.3969e+03	4.2000e+03	4.4600e+03	3.9985e+03	4.3579e+03	4.5928e+03	4.4231e+03
15	4.2144e+03	4.3990e+03	4.2010e+03	4.4595e+03	4.0036e+03	4.3600e+03	4.5913e+03	4.4241e+03
16	4.2159e+03	4.3985e+03	4.2036e+03	4.4590e+03	4.0062e+03	4.3554e+03	4.5892e+03	4.4267e+03
17	4.2138e+03	4.3985e+03	4.2026e+03	4.4595e+03	4.0005e+03	4.3508e+03	4.5897e+03	4.4231e+03
18	4.2133e+03	4.3985e+03	4.2015e+03	4.4610e+03	3.9944e+03	4.3497e+03	4.5892e+03	4.4179e+03
19	4.2169e+03	4.3974e+03	4.1995e+03	4.4615e+03	4.0005e+03	4.3477e+03	4.5897e+03	4.4231e+03
20	4.2174e+03	4.3969e+03	4.1985e+03	4.4600e+03	4.0031e+03	4.3405e+03	4.5913e+03	4.4221e+03
21	4.2149e+03	4.3944e+03	4.1974e+03	4.4579e+03	3.9928e+03	4.3390e+03	4.5882e+03	4.4138e+03

- Referencing (Common Average):

normalisedData | referencedData

30720x8 double

	1	2	3	4	5	6	7	8
1	-37.5958	10.3887	-0.4955	-15.1754	-35.8822	1.9445	-26.3369	-30.7989
2	-39.6471	9.3630	-5.1109	-15.6882	-36.9079	5.5342	-28.9010	-32.8502
3	-31.4420	17.0553	-2.5468	-18.7651	-25.1130	7.5855	-21.2087	-34.3887
4	-31.4420	18.5938	-3.5724	-18.2523	-22.0361	3.9958	-19.1575	-33.3630
5	-41.1856	14.4912	-7.6750	-15.6882	-38.4463	2.4573	-24.2857	-32.3374
6	-35.0317	18.0810	3.6070	-15.1754	-42.0361	1.9445	-20.1831	-29.7733
7	-27.3394	21.1579	9.7609	-14.1498	-28.7028	-0.6196	-17.1062	-29.7733
8	-36.0574	18.0810	-2.0340	-15.6882	-31.2669	-3.6965	-21.7216	-34.3887
9	-36.0574	17.0553	-1.5212	-19.2780	-42.5489	-4.2094	-21.2087	-33.8759
10	-22.2112	22.6963	11.2993	-18.7651	-37.9335	-0.6196	-14.5421	-24.1323
11	-21.1856	29.8758	11.2993	-13.6369	-31.2669	3.4829	-13.5164	-21.0554
12	-26.8266	28.8502	7.7096	-11.5857	-35.3694	5.5342	-15.0549	-24.1323
13	-26.8266	24.2348	9.2481	-13.1241	-37.4207	7.0727	-13.5164	-22.5938
14	-26.8266	23.7220	8.2224	-14.6626	-33.8310	12.7137	-13.0036	-21.5682
15	-26.3138	25.7733	9.2481	-15.1754	-28.7028	14.7650	-14.5421	-20.5425
16	-24.7753	25.2605	11.8122	-15.6882	-26.1387	10.1496	-16.5933	-17.9784
17	-26.8266	25.2605	10.7865	-15.1754	-31.7797	5.5342	-16.0805	-21.5682
18	-27.3394	25.2605	9.7609	-13.6369	-37.9335	4.5086	-16.5933	-26.6964
19	-23.7497	24.2348	7.7096	-13.1241	-31.7797	2.4573	-16.0805	-21.5682
20	-23.2369	23.7220	6.6840	-14.6626	-29.2156	-4.7222	-14.5421	-22.5938
21	-25.8010	21.1579	5.6583	-16.7139	-39.4720	-6.2607	-17.6190	-30.7989

- Normalisation:

normalisedData | referencedData | fullextractedData

30720x8 double

	1	2	3	4	5	6	7	8
1	-0.0032	8.8075e-04	-4.2011e-05	-0.0013	-0.0030	1.6485e-04	-0.0022	-0.0026
2	-0.0034	7.9380e-04	-4.3330e-04	-0.0013	-0.0031	4.6919e-04	-0.0025	-0.0028
3	-0.0027	0.0014	-2.1592e-04	-0.0016	-0.0021	6.4310e-04	-0.0018	-0.0029
4	-0.0027	0.0016	-3.0287e-04	-0.0015	-0.0019	3.3876e-04	-0.0016	-0.0028
5	-0.0035	0.0012	-6.5069e-04	-0.0013	-0.0033	2.0833e-04	-0.0021	-0.0027
6	-0.0030	0.0015	3.0581e-04	-0.0013	-0.0036	1.6485e-04	-0.0017	-0.0025
7	-0.0023	0.0018	8.2753e-04	-0.0012	-0.0024	-5.2532e-05	-0.0015	-0.0025
8	-0.0031	0.0015	-1.7244e-04	-0.0013	-0.0027	-3.1339e-04	-0.0018	-0.0029
9	-0.0031	0.0014	-1.2896e-04	-0.0016	-0.0036	-3.5687e-04	-0.0018	-0.0029
10	-0.0019	0.0019	9.5796e-04	-0.0016	-0.0032	-5.2532e-05	-0.0012	-0.0020
11	-0.0018	0.0025	9.5796e-04	-0.0012	-0.0027	2.9528e-04	-0.0011	-0.0018
12	-0.0023	0.0024	6.5362e-04	-9.8223e-04	-0.0030	4.6919e-04	-0.0013	-0.0020
13	-0.0023	0.0021	7.8405e-04	-0.0011	-0.0032	5.9962e-04	-0.0011	-0.0019
14	-0.0023	0.0020	6.9710e-04	-0.0012	-0.0029	0.0011	-0.0011	-0.0018
15	-0.0022	0.0022	7.8405e-04	-0.0013	-0.0024	0.0013	-0.0012	-0.0017
16	-0.0021	0.0021	0.0010	-0.0013	-0.0022	8.6048e-04	-0.0014	-0.0015
17	-0.0023	0.0021	9.1448e-04	-0.0013	-0.0027	4.6919e-04	-0.0014	-0.0018
18	-0.0023	0.0021	8.2753e-04	-0.0012	-0.0032	3.8224e-04	-0.0014	-0.0023
19	-0.0020	0.0021	6.5362e-04	-0.0011	-0.0027	2.0833e-04	-0.0014	-0.0018
20	-0.0020	0.0020	5.6667e-04	-0.0012	-0.0025	-4.0035e-04	-0.0012	-0.0019
21	-0.0022	0.0018	4.7971e-04	-0.0014	-0.0033	-5.3078e-04	-0.0015	-0.0026

- Filtering
    - Band Pass filter (theta, alpha, beta, gamma with Notch Filter at 50Hz) sample code:

```
%applying band-pass filter to delta, theta, alpha, beta, gamma freq band
delta1 = Band_pass_filter(data1,1,4,128);
delta2 = Band_pass_filter(data2,1,4,128);
theta1 = Band_pass_filter(data1,4,8,128);
theta2 = Band_pass_filter(data2,4,8,128);
alpha1 = Band_pass_filter(data1,8,12,128);
alpha2 = Band_pass_filter(data2,8,12,128);
beta1 = Band_pass_filter(data1,15,30,128);
beta2 = Band_pass_filter(data2,15,30,128);
gammaU1 = Band_pass_filter(data1,30,60,128);
gammaU2 = Band_pass_filter(data2,30,60,128);
gamma1 = Notch_filter(gammaU1, 50, 128); %apply Notch filter at 50Hz due to power line noise
gamma2 = Notch_filter(gammaU2, 50, 128);
```

- Synchrony Measure — Matrix of inter-brain synchrony
    - Cross-Correlation: Max(xcorr); xcorr2

```
%compute xcorr & extract variables for each freq band
[d(:,1),d(:,2)] = xcorr(singleD1-mean(singleD1), singleD2-mean(singleD2),'coeff');
[t(:,1),t(:,2)] = xcorr(singleT1-mean(singleT1), singleT2-mean(singleT2),'coeff');
[a(:,1),a(:,2)] = xcorr(singleA1-mean(singleA1), singleA2-mean(singleA2),'coeff');
[b(:,1),b(:,2)] = xcorr(singleB1-mean(singleB1), singleB2-mean(singleB2),'coeff');
[g(:,1),g(:,2)] = xcorr(singleG1-mean(singleG1), singleG2-mean(singleG2),'coeff');

%compute max(xcorr) for each freq band
deltamax = max(d(:,1));
thetamax = max(t(:,1));
alphamax = max(a(:,1));
betamax = max(b(:,1));
gammamax = max(g(:,1)); %getting 0.1477 for all values
```

- Matrix of inter-brain synchrony —> Estimate Inter Brain density —> Single value representing IBD

```
%computing inter-brain density (IBD) values, using algorithm from
%brain_plot.m & compcorr.m for delta only
adj = compcorr(delta1, delta2); %different synchrony measure from xcorr?
value123=sum(adj(:)==1);
result13=value123/(14^2); %rationale?
```

Comments:

- Need to determine rationale for IBD algorithm
- Future steps:
    - Statistical mean; Test for gaussianity
    - Compare p-values (ranks test) between scenarios