Table of Contents

CCA analysis

loads behaviour and network data. Performs MCA on connectivity to reduce dimensionality. Prints the corresponding components and the variance accounted for. Enters 5 behavioural variables and 5 MCA components into the CCA. Two significant modes result: seem to be very robust across multiple parameters including the number of connectomes (10 ,15, 20), the atlas used (114,214,512), the number of connectons excluded, the number of components included as the dep variable

```
clearvars
close all

addpath('functions');
addpath(genpath('/projects/sw49/BCT/'));
addpath(genpath('/projects/sw49/FSLNets/'));
addpath(genpath('/home/lukehearne/R/'));

% inputs
basedir = '/scratch/sw49/1_LEADStrokeMapping/';
dataType = '_conbound10/'; %data type
parcLabel = '140/'; % label for parcellation
load('/projects/sw49/Atlas/140COG.mat');
load('/projects/sw49/Atlas/140parcellation_Yeo8Index.mat');

behav.variables = [3,4,8,10,11,12,70,31];
lesion_affection = 5;
comps = 5;
% load connectomes
[Cpre,Cpost,nodata] =
 load_connectomes([basedir,'connectomes',dataType,parcLabel]);
Nodes = size(Cpre,1);

% load behaviour
[data, key, P_ID] = load_stroke_behav;
behav.data = data(:,behav.variables);

% exclusions
exclude = sum(isnan(behav.data),2)>0; %missing behav data
exclude = exclude+nodata>0; %missing lesion data
behav.data(exclude,:) = [];
```

```
P_ID(exclude) = [];
Cpre(:,:,exclude) = [];
Cpost(:,:,exclude) = [];
Cdiff = Cpre-Cpost;

exclude = squeeze(sum(sum(Cdiff,1),2)==0); %missing connectivity data
behav.data(exclude,:) = [];
P_ID(exclude) = [];
Cpre(:,:,exclude) = [];
Cpost(:,:,exclude) = [];
Cdiff(:,:,exclude) = [];
% transform the spatial neglect variable
behav.data(:,7) = behav.data(:,7)*-1;
%behav.data(:,6) = normal_transform(behav.data(:,6));
%behav.data(:,7) = normal_transform(behav.data(:,7))*-1;
behav.data = normal_transform(behav.data);
% sample size
SampSize = size(Cpre,3);
```

set up CCA

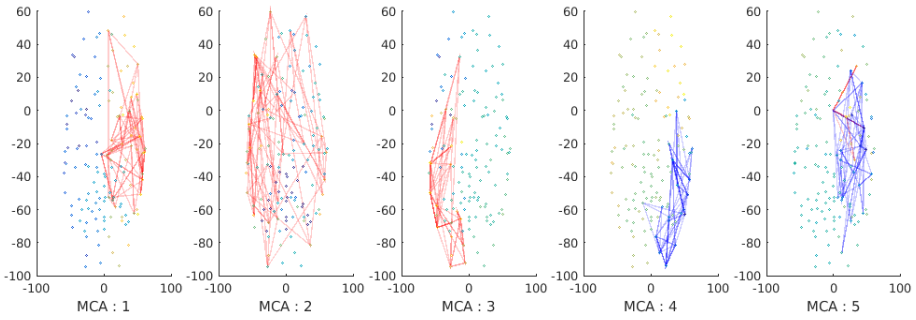
```
idx = sum(Cdiff>0,3)>lesion_affection; %index informative voxels
for i = 1:SampSize
    tmp = Cdiff(:,:,i)>0; %binarize
    Conn(i,:) = tmp(idx);
end

% behav side
x = behav.data(:,4:end);
```

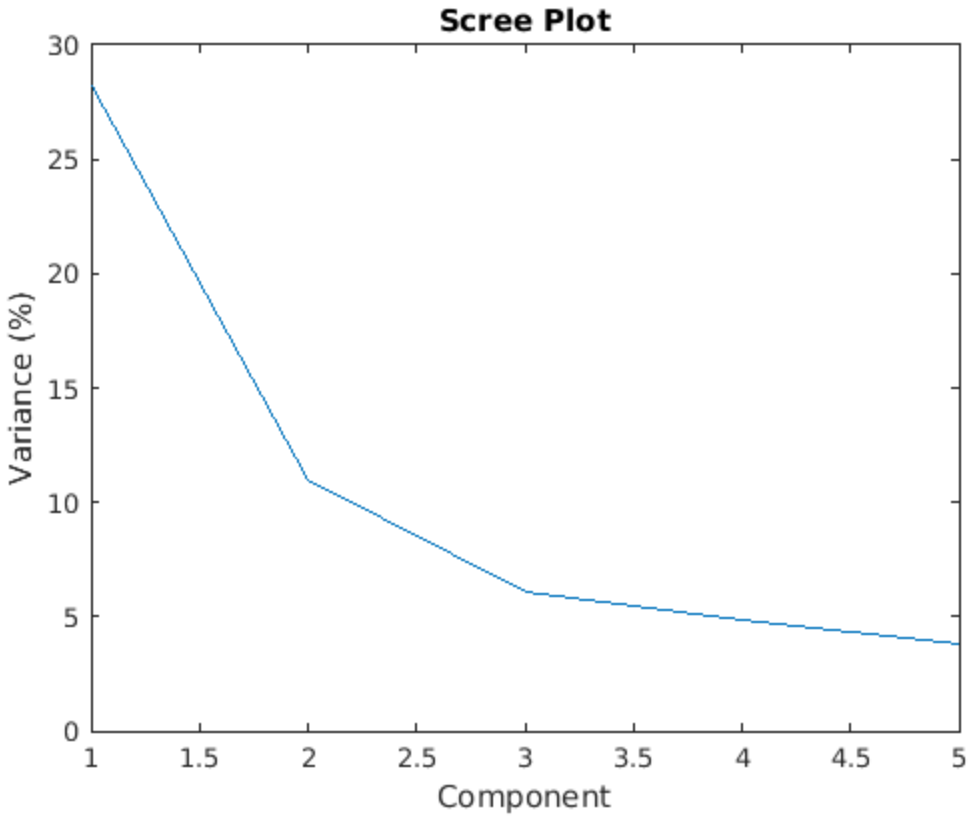
Performs MCA via R.

```
csvwrite('/projects/sw49/Results/MCA/MCAinput.csv',double(Conn))
csvwrite('/projects/sw49/Results/MCA/MCAcomp.csv',
[0,0;0,double(comps)]);

system('Rscript MCAR.R')
MCA.VarWeights = csvread('/projects/sw49/Results/MCA/
MCA_VarWeights.csv',1,1);
MCA.VarWeights = MCA.VarWeights(2:2:end,:)'; %unsure why this is
 represent 2:2
% print components
figure('Color',[1 1 1],'pos',[100 600 1200 350]);
nidx = find(idx);
for i = 1:comps
    subplot(1,comps,i)
    MAT = zeros(Nodes,Nodes);
    MAT(nidx) = MCA.VarWeights(i,:);
    draw_connectome(MAT,COG,1,50);
    xlabel(['MCA : ',num2str(i)])
end
```

```
% plot variance explained as scree plot
figure
MCA.Eigen = csvread('/projects/sw49/Results/MCA/
MCA_Eigenvalues.csv',1,1);
plot(MCA.Eigen(1:comps,2));
title('Scree Plot')
xlabel('Component');
ylabel('Variance (%)');
```

```
Loading required package: ggplot2
Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at
 https://goo.gl/13EFCZ
[1] "data loaded"
[1] "MCA finished"
Time difference of 3.785221 secs

ans =

     0
```





do the CCA

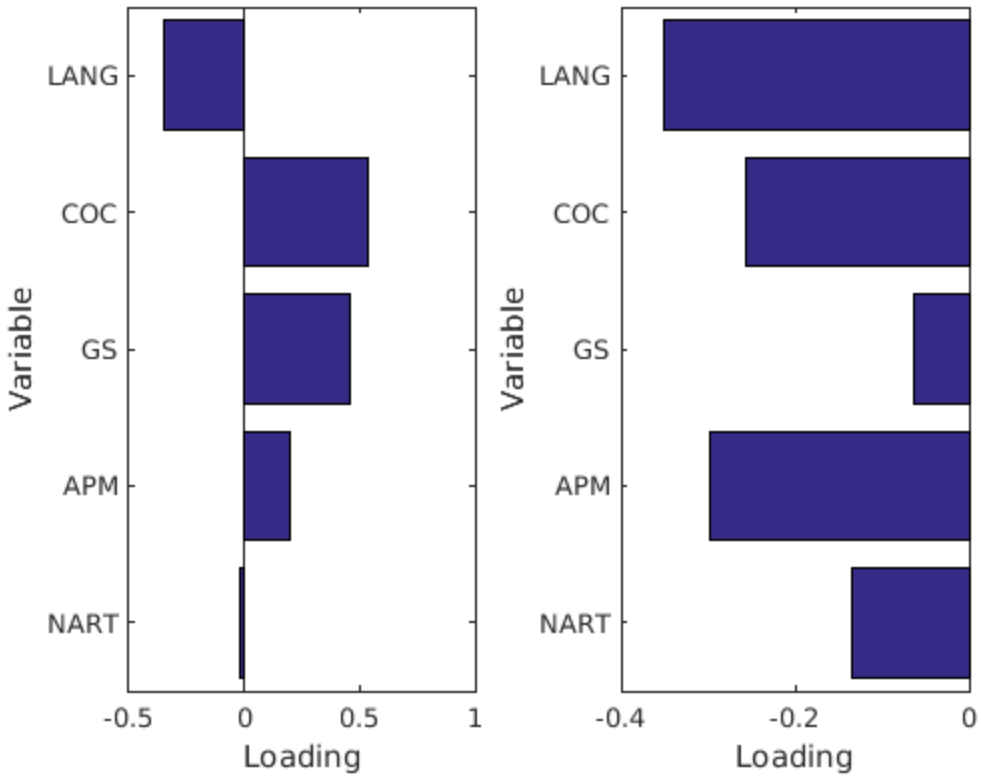
```
MCA.IndWeights = csvread('/projects/sw49/Results/MCA/
MCA_IndWeights.csv',1,1);
[grotA, grotB, grotR, grotU, grotV,
 grotstats]=canoncorr(x,MCA.IndWeights);
conload=corr(x,grotV);
disp(grotstats.p);

%Non-parametric test
% perms = 1000;
% for i = 1:perms
%     t = randperm(length(x));
%     [~,~,~,~,~, permstats]=canoncorr(x(t,:),MCA.IndWeights);
%     p_perm(i,:) = permstats.p;
% end
```

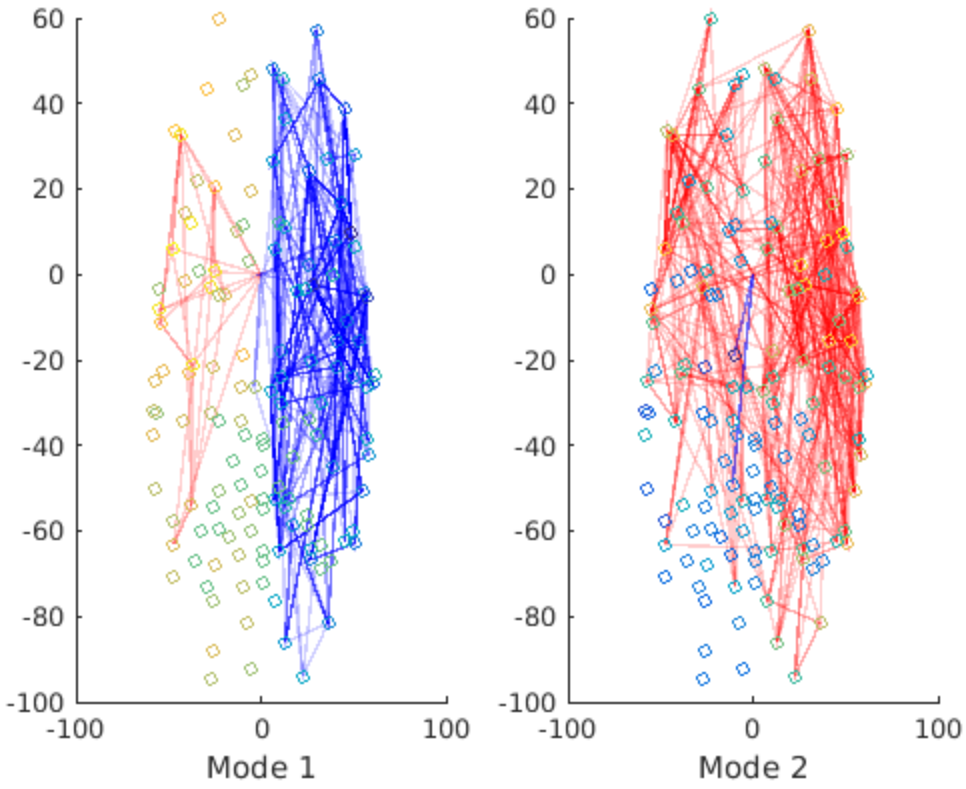
0.0000 0.0196 0.1745 0.3997 0.6440

generate some simple figures

```
figure
for i = 1:2
subplot(1,2,i)
```

```
barh(conload(:,i));
set(gca,'YTick',1:5,'YTickLabel', {'NART','APM','GS','COC','LANG'});
xlabel('Loading');
ylabel('Variable');

end

% print edges/nodes
figure
Att = ones(Nodes,1);
for i = 1:2
    subplot(1,2,i)
    Mode = corr(grotV(:,i),Conn)';
    MAT = zeros(Nodes,Nodes);
    MAT(nidx) = Mode;
    draw_connectome(MAT,COG,20,500);
    xlabel(['Mode ',num2str(i)]);
    out = ['/projects/sw49/Results/Mode_',num2str(i)];
    mat2brainnet(MAT,COG,Att,Att,out,out);
end
```



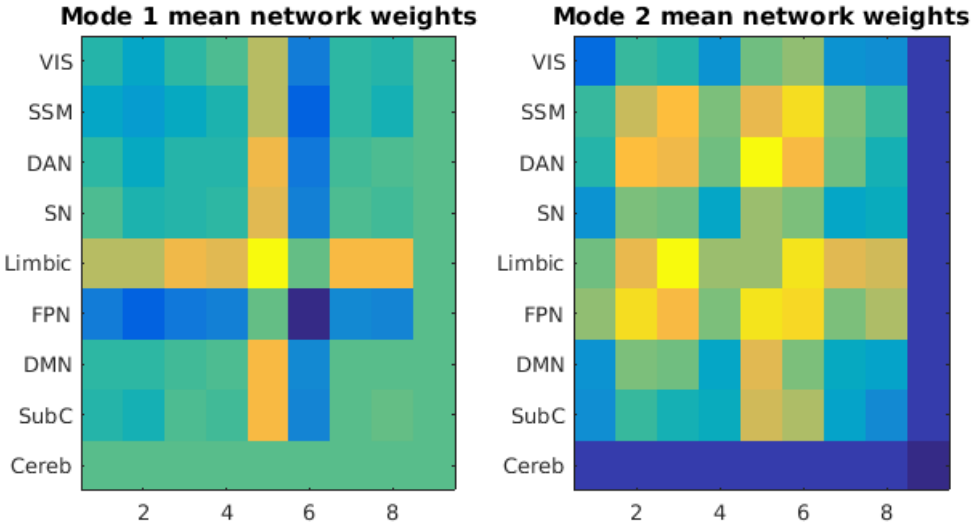


functional networks

mean weights within/across each functional network (yeo)

```
figure('pos',[100 600 700 350]);

for i = 1:2

    Mode = corr(grotV(:,i),Conn)';
    MAT = zeros(Nodes,Nodes);
    MAT(nidx) = Mode;
    MAT = MAT+MAT';

    NET = zeros(9);
    for j = 1:max(Yeo8Index)
        for k = 1:max(Yeo8Index)
            r = Yeo8Index==j;
            c = Yeo8Index==k;

            d = MAT(r,c);
            NET(j,k)=mean(d(:));
        end
    end

    labels =
{'VIS';'SSM';'DAN';'SN';'Limbic';'FPN';'DMN';'SubC';'Cereb'};
```

```
    subplot(1,2,i)
    imagesc(NET)
    set(gca,'YTick',1:9,'YTickLabel', labels);
    %set(gca,'XTick',1:9,'XTickLabel', labels);
    title(['Mode ',num2str(i), ' mean network weights']);
end
```



project the top 10 subj back into lesion space.

calculate lesion size

```
n = 20;
for i = 1 :2
    [~,subjTop] = sort(grotV(:,i),'descend');
    subjTop = subjTop(1:n);

    [~,subjBot] = sort(grotV(:,i),'ascend');
    subjBot = subjBot(1:n);

    lesionPos = zeros(181,217,181);
    lesionNeg = lesionPos;

    for s = 1:n
        lesionfile =
 [basedir,'Lesions/',P_ID{subjTop(s)},'_interp.nii'];
        [~,data] = read(lesionfile);
        lesionPos = lesionPos+data;

        lesionfile =
 [basedir,'Lesions/',P_ID{subjBot(s)},'_interp.nii'];
        [~,data] = read(lesionfile);
        lesionNeg = lesionNeg+data;
    end

    path = '/projects/sw49/Results/CCA/';
```

```
    mat2nii(lesionPos,
[path,'PosLesions_',num2str(n),'_Mode',num2str(i),'.nii'],...
        size(lesionPos),32,lesionfile);
    mat2nii(lesionNeg,
[path,'NegLesions_',num2str(n),'_Mode',num2str(i),'.nii'],...
        size(lesionNeg),32,lesionfile);

end
```

Conclusions

Results seem to indicate two modes - the first indicates an attention bias on the right and a language bias on the left, makes sense. The second is a little more interesting as ALL variables are negative loaded, suggesting a sort of 'global' deficit pattern associated with largely frontal-posterior & interhemispheric connectivity.

```
% it is actually a little hard to interpret negative weights, as they
% should represent a lack of lesion loading, which I suppose is a
% preservation of connectivity? Or would it be appropriate to "flip"
 the
% interpretation and say damage to negative weights is associated with
 the
% opposite weighted independent variables? I think if we only include
% connections that are damaged in individuals (i.e., no connections
 that
% are not damaged across the whole group) the second interpretation is
% viable.

%functional network mapping to Yeo's 7 networks (+subcortical and the
%cerebellum) show a nice limbic = lang, fpn = attention/fg effect in
 the
%first mode. The second mode is more global with some interesting
 links
%between limbic and FPN and DAN. Kind of makes sense, if you damage
 the
%limbic or FPN somewhat indepedently you find more indepedent
 cognitive
%deficets, but if you damage the connections between you find more
 global
%deficiets.
```

Published with MATLAB® R2015b