Table of Contents


```
%Hub analysis

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 = '_conbound15/'; %data type
parcLabel = '214/'; % label for parcellation
load('/projects/sw49/Atlas/214COG.mat');

behav.variables = [3,4,8,10,11,12,70,31]; %may be altered in future

% load connectomes
[~,~,nodata,Cpre,Cpost] =
 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(:,6) = behav.data(:,6)*-1;
```

```
behav.data(:,6) = normal_transform(behav.data(:,6))*-1;
behav.data(:,7) = normal_transform(behav.data(:,7))*-1;

% sample size
SampSize = size(Cpre,3);
```

-------- Graph analysis

```
k = 0.15; %top 15%

for i = 1:SampSize
    % calculate k at individual level
    tmp = sort(sum(Cpre(:,:,i)),'descend');
    Klevel = tmp(round(length(tmp)*k));
    [ConCount(i,:),~,ConStren(i,:),HubMat] =
 find_hubs(Cpre(:,:,i),Klevel);
    HubPrint(:,:,i) = HubMat(:,:,1);

    % count the number of connections in each connection class.
    tmp = Cdiff(:,:,i)>0;

    for h = 1:3
        idx = logical(HubMat(:,:,h));
        HubDamage(i,h) = sum(tmp(idx))/2;
        HubDamageCont(i,h) = (sum(tmp(idx))/2)/sum(sum(idx)); %divided
 by number of total conn in class
    end
end

% communicability
for i = 1:SampSize
    for l = 1 % no difference across this parameter
        Cmcy.pre(i,l) =
 sum(sum(getCommunicability(Cpre(:,:,i),l,1)))/2;
        Cmcy.post(i,l) =
 sum(sum(getCommunicability(Cpost(:,:,i),l,1)))/2;
        Cmcy.diff(i,l) = Cmcy.pre(i,l) - Cmcy.post(i,l);
    end
end

% calculate lesion size
for i = 1:SampSize
    lesionfile = [basedir,'Lesions/',P_ID{i},'_interp.nii'];
    [~,data] = read(lesionfile);
    lesionSize(i,1) = sum(sum(sum(data)));
end

% linear regression - does hub connection damage predict
 communicability
% when controlling for lesion size & other types of damage?
X = [sqrt(lesionSize),HubDamage];
lm = fitlm(X,Cmcy.diff,'linear');
disp(lm)
```

```
%OK, so the degree within each connection type are so highly
 correlated
%that the multiple regression is probably not appopriate (predictors
 are
%too highly correlated).
```

Linear regression model:
y ~ 1 + x1 + x2 + x3 + x4

Estimated Coefficients:

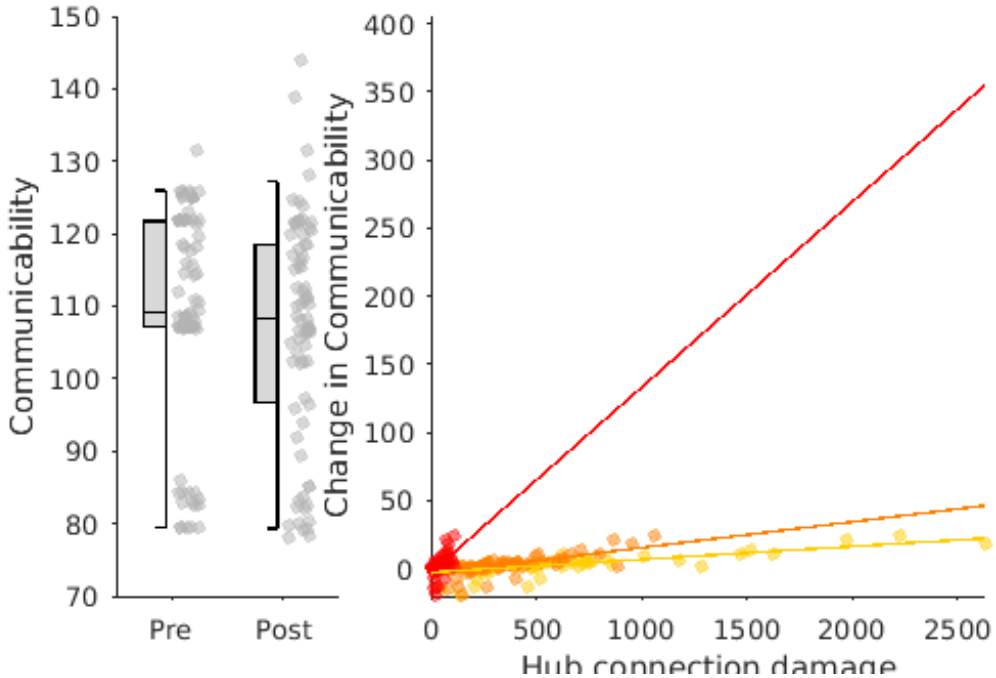
	Estimate	SE	tStat	pValue
(Intercept)	*-3.2884*	*0.8825*	*-3.7262*	*0.00039042*
x1	*0.029965*	*0.010803*	*2.7737*	*0.0070993*
x2	*0.028505*	*0.09518*	*0.29948*	*0.76546*
x3	*-0.0052345*	*0.019462*	*-0.26896*	*0.78875*
x4	*0.0055734*	*0.0050538*	*1.1028*	*0.27388*

Number of observations: 75, Error degrees of freedom: 70
Root Mean Squared Error: 4.28
R-squared: 0.601, Adjusted R-Squared 0.579
F-statistic vs. constant model: 26.4, p-value = 2.31e-13

Plots

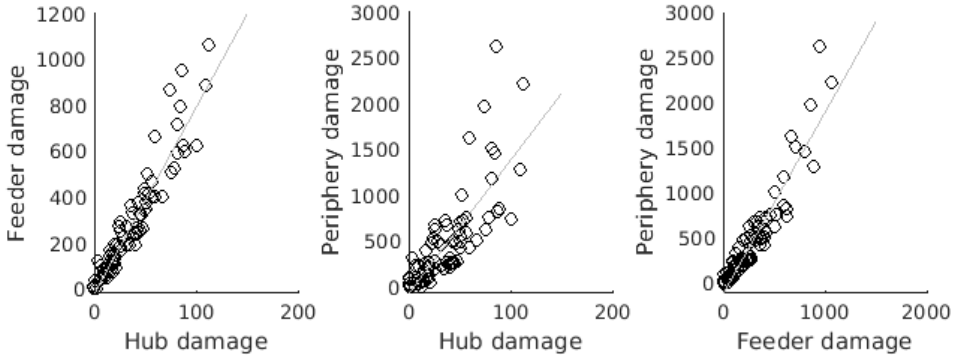
```
close all

figure('Color',[1 1 1],'pos',[1000 600 550 350]);
figparam.lw = 1;
figparam.alpha = 0.5;
figparam.col = [0.7 0.7 0.7];%[0.53 0.8 0.92];
figparam.s = 15;

%--------------------------
subplot(1,3,1)
box_and_scatterplot(Cmcy.pre,1,figparam.lw,figparam.s,...
    figparam.col,figparam.alpha); hold on
box_and_scatterplot(Cmcy.post,2,figparam.lw,figparam.s,...
    figparam.col,figparam.alpha); hold on

set(gca,'FontName', 'Helvetica','FontSize', 10,'Box','off',...
'TickDir','out','ygrid','off','XLim',[.5 2.5]);

ylabel('Communicability')
set(gca, 'XTick',[1,2],'XTickLabel',{'Pre','Post'});

%--------------------------
subplot(1,3,[2,3])
figparam.col = [1,.8,0];
scatter(HubDamage(:,3),Cmcy.diff,...
```

```
        'MarkerEdgeColor',figparam.col,...
        'LineWidth',figparam.lw,...
        'MarkerFaceAlpha',figparam.alpha,...
        'MarkerEdgeAlpha',figparam.alpha,...
        'MarkerFaceColor', figparam.col,...
        'SizeData',figparam.s); hold on

figparam.col = [1,0.5,0];
scatter(HubDamage(:,2),Cmcy.diff,...
        'MarkerEdgeColor',figparam.col,...
        'LineWidth',figparam.lw,...
        'MarkerFaceAlpha',figparam.alpha,...
        'MarkerEdgeAlpha',figparam.alpha,...
        'MarkerFaceColor', figparam.col,...
        'SizeData',figparam.s); hold on

figparam.col = [1,0,0];
scatter(HubDamage(:,1),Cmcy.diff,...
        'MarkerEdgeColor',figparam.col,...
        'LineWidth',figparam.lw,...
        'MarkerFaceAlpha',figparam.alpha,...
        'MarkerEdgeAlpha',figparam.alpha,...
        'MarkerFaceColor', figparam.col,...
        'SizeData',figparam.s); hold on

h = lsline;
set(h(3),'color',[1,.8,0],'LineWidth',1)
set(h(2),'color',[1,.5,0],'LineWidth',1)
set(h(1),'color',[1,0,0],'LineWidth',1)

set(gca,'FontName', 'Helvetica','FontSize', 10,'Box','off',...
'TickDir','out','ygrid','off','XLim',[0 max(max(HubDamage))]);

ylabel('Change in Communicability')
xlabel('Hub connection damage');
```



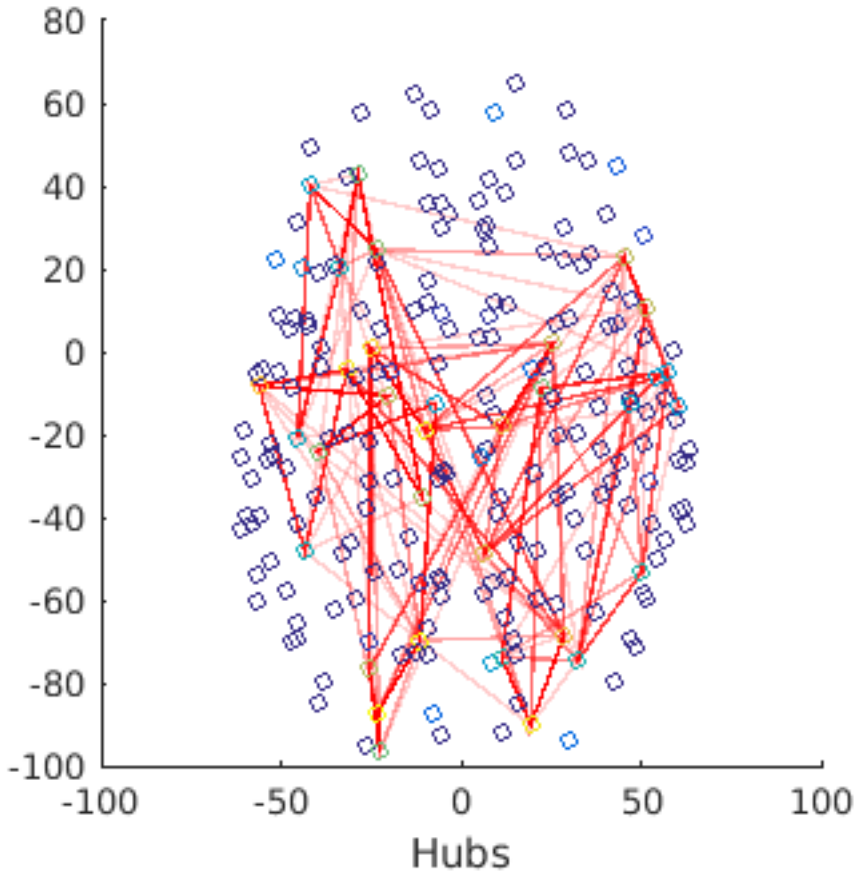
I think the high colinearity between the variables precludes any meaningingful assessment of each connection classes contribution to i) global processing loss and ii) behaviour. I guess we can conclude that strokes don't really attack structural hubs.

```
figure('Color',[1 1 1],'pos',[1000 600 750 250]);
subplot(1,3,1)
scatter(HubDamage(:,1),HubDamage(:,2),'k'); hold on
lsline
ylabel('Feeder damage')
xlabel('Hub damage');

subplot(1,3,2)
scatter(HubDamage(:,1),HubDamage(:,3),'k'); hold on
lsline
ylabel('Periphery damage')
xlabel('Hub damage');

subplot(1,3,3)
scatter(HubDamage(:,2),HubDamage(:,3),'k'); hold on
lsline
ylabel('Periphery damage')
xlabel('Feeder damage');
```



Print edge and node files

```
figure('Color',[1 1 1],'pos',[1000 600 350 350]);
draw_connectome(sum(HubPrint,3),COG,20,500)
xlabel('Hubs');
```



Published with MATLAB® R2015b