

Data visualization tutorial for COCOAN101-L13

by Choong-Wan (Wani) Woo

Agenda

1. Run some of the analyses (one sample t-test) from L04 example code

```
% key functions: fmri_data, plot

basedir = '/Users/clinspsywoo/Dropbox/2011-yr/Teaching/COCOAN101/L04';
% datdir = fullfile(basedir, 'example_data/bmrk3_temp_avg');
datdir = '/Users/clinspsywoo/Dropbox/2011-yr/Teaching/R_advancedfmri/advancedfmrianalys

imgs = filenames(fullfile(datdir, 'sub*heat*nii'));

% dat = fmri_data(imgs);
% default mask: >> which('brainmask_canlab.nii')
% or you can also use different masks, e.g.,

dat_gray = fmri_data(imgs, which('gray_matter_mask.nii'));

loading mask. mapping volumes.
checking that dimensions and voxel sizes of volumes are the same.
Pre-allocating data array. Needed: 154924704 bytes
Loading image number: 198
경과 시간은 1.847960초입니다.
Image names entered, but fullpath attribute is empty. Getting path info.
Number of unique values in dataset: 31852256 Bit rate: 24.92 bits
```

```
tdat = ttest(dat_gray, 0.05, 'bfr');
```

One-sample t-test
Calculating t-statistics and p-values

Image 1
51 contig. clusters, sizes 1 to 12942
Positive effect: 29884 voxels, min p-value: 0.00000000
Negative effect: 23179 voxels, min p-value: 0.00000000

2. brain_activation_display.m in CocoanCORE

Recently upgraded from brain_activations_wani.m

Please use brain_activations_display, not _wani.m

```
close all;
% you need this to use brain_activations_display: https://github.com/cocoanlab/cocoanC
```

```
[out, o2] = brain_activations_display(region(tdat));
```

Grouping contiguous voxels: 51 regions

Using custom color maps.

Getting heat-mapped colors

Building color change function call

Loading surface image

Running color change.

eval: [c,alld] = getVertexColors_cocoan(xyz, p, mycolors, [.5 .5 .5], 3,'colorscale',actcolors,'fsavg_ri

Main color vertices: 163842 vertices. selecting: 53063

Calculating distance...

Finished!

Using custom color maps.

Getting heat-mapped colors

Building color change function call

Loading surface image

Running color change.

eval: [c,alld] = getVertexColors_cocoan(xyz, p, mycolors, [.5 .5 .5], 3,'colorscale',actcolors,'fsavg_le

Main color vertices: 163842 vertices. selecting: 53063

Calculating distance...

Finished!

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

sagittal montage: 1295 voxels displayed, 51768 not displayed on these slices

sagittal montage: 1402 voxels displayed, 51661 not displayed on these slices

sagittal montage: 1077 voxels displayed, 51986 not displayed on these slices

sagittal montage: 932 voxels displayed, 52131 not displayed on these slices

axial montage: 630 voxels displayed, 52433 not displayed on these slices

axial montage: 1134 voxels displayed, 51929 not displayed on these slices

axial montage: 1479 voxels displayed, 51584 not displayed on these slices

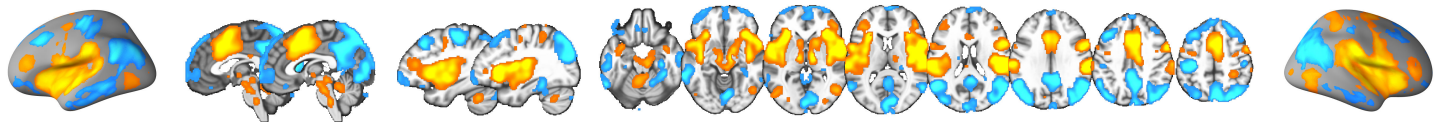
axial montage: 1434 voxels displayed, 51629 not displayed on these slices

axial montage: 1334 voxels displayed, 51729 not displayed on these slices

axial montage: 1120 voxels displayed, 51943 not displayed on these slices

axial montage: 1120 voxels displayed, 51943 not displayed on these slices

axial montage: 1119 voxels displayed, 51944 not displayed on these slices



```
help brain_activations_display
```

This function display brain activations on a inflated brain and few sagittal, axial slices. Cocoon style activation visualization.

:Usage:
::

```
[out, o2] = brain_activations_display(cl, varargin)
```

:Inputs:

****r:****
region object/activation map

:Optional Inputs:

****inflated:****
not recommended
use inflated brain. We use the 32k inflated brain surface from HCP connectome workbench. (Q1-Q6_R440.R.inflated.32k_fs_LR.surf.gii and Q1-Q6_R440.L.inflated.32k_fs_LR.surf.gii)

****very_inflated (default):****
recommended
use freesurfer inflated brain with Thomas Yeo group's RF_ANTs mapping from MNI to Freesurfer. (<https://doi.org/10.1002/hbm.24213>)

****very_inflated_workbench:****
use very inflated brain. We also use the 32k inflated brain surface from HCP connectome workbench.
(Q1-Q6_R440.R.very_inflated.32k_fs_LR.surf.gii and Q1-Q6_R440.L.very_inflated.32k_fs_LR.surf.gii)

****depth:****
depth for surface map, (e.g., 'depth', 4)
default is 3 mm

****color:****
if you want to use one color for blobs, you can specify color using this option.

****region_color (or region_colors):****
if you want to use one color for each region, you can specify region colors using this option. The input should have the same number of rows with the region, i.e., # region x 3

****custom_color (or custom_colors):****
if you want to define colors for all voxels on your own, you can specify voxel colors using this option. The input should have the same number of rows with the voxels, i.e., # voxel x 3

****axial_slice_range:****
followed by axial slice range in a cell
e.g., 'axial_slice_range', {[-10 30]}
You can also define spacing in the same cell.
e.g., 'axial_slice_range', {[-10 30], 6}
The default range is [-20 25] with the spacing of 10.

****outline:****
draw outline, default linewidth: 2

****surface_only:****
you can use this option to draw only surface. As a default, it will show only lateral view.

****surface_all:****
 You can use this option to draw lateral and also medial view of surface maps.

****all:****
 This will put 2 surface and 12 montage maps (4 sagittal and 8 axial) in one row. This will be good when you want to explore the activation maps.

****all2:****
 This will show 2 surface and 10 montage maps (4 sagittal and 6 axial) in one row. This will be good when you want to use the figure for publication.

****all_xyz****
 You can also specify x and z for sagittal and axial slices for "all" option using this options.
 E.g., 'all_xyz', [-5 2 -35 35 -30:12:60]
 first four will be used as x's and the eight numbers after that will be used as z. More than 12 numbers will be ignored.

****all2_xyz****
 You can also specify x and z for sagittal and axial slices for "all2" option using this options.
 E.g., 'all_xyz', [-5 2 -35 35 -30:12:60]
 first four will be used as x's and the six numbers after that will be used as z. More than 10 numbers will be ignored.

****montage_only:****
 you can use this option to draw only montage

****x1:****
 you can specify the sagittal slice numbers using this option
 e.g., 'x1', [-5 4] (default: [-5 5])

****x2:****
 you can specify the second set of sagittal slice numbers using this option. If you don't want to draw this, just leave it blank.
 e.g., 'x2', [-41 41] or 'x2', [] (default: [-37 37]);

****y:****
 you can specify the coronal slice numbers using this option.
 e.g., 'y', [-10 10] (default: []);

****z:****
 you can specify the axial slice numbers
 e.g., 'z', [-25 -15 -6 13 22]
 (default: slice range between z = -20 and 25 with spacing 10 mm)

****squeeze_x1:****
 you can specify squeeze percentage for x1 using this. default: 40
 e.g., 'squeeze_x1', 0 or 'squeeze_x1', 30

****squeeze_x2:****
 you can specify squeeze percentage for x2 using this. default: 50

****squeeze_y:****
 you can specify squeeze percentage for y using this. default: 30

****squeeze_z:****
 you can specify squeeze percentage for z using this. default: 20

****pruned:****
 if you have pruned version of map, you can use this option.

currently only works with (e.g., -3, -2, -1, 1, 2, 3)

```
**cmaprange:**
    you can use this option to specify cmaprange. (see help of
    addblob.m to see more details about cmaprange)

**o2:**
    if you want to reuse montages underlay that are already exist, you
    can simply provide o2 (fmridisplay object) as an input. It
    automatically check whether there is an input that is fmridisplay
    object, and reuse those montage.

**colorbar:**
    Show colorbar. default: false

**colorbar_fontsize:**
    Font size of colorbar. default: 14

**prioritize_last:**
    For determining colors of each vertex, prioritize the colors of
    the voxels that are drawn last. Without specifying this, colors
    are determined based on the colors of nearest voxels.
```

Examples:

```
% % Yeo 10 network, surface only
% gray_mask = fmri_data(which('Yeo_10networks_4mm.nii'));
% load(which('Schaefer_Net_Labels_r265.mat'));
% brain_activations_display(region(gray_mask, 'unique_mask_values'), 'surface_only', 'region_color', S

% % Yeo 10 network, surface and montage
% gray_mask = fmri_data(which('Yeo_10networks_4mm.nii'));
% load(which('Schaefer_Net_Labels_r265.mat'));
% brain_activations_display(region(gray_mask, 'unique_mask_values'), 'all2', 'region_color', Schaefer_M

% % SIIPS1 mask, surface only
% SIIPS1_mask = fmri_data(which('nonnoc_v11_4_137subjmap_weighted_mean.nii'));
% brain_activations_display(region(SIIPS1_mask, 'contiguous_regions'), 'surface_only', 'colorbar');

% % SIIPS1 mask, surface and montage
% SIIPS1_mask = fmri_data(which('nonnoc_v11_4_137subjmap_weighted_mean.nii'));
% brain_activations_display(region(SIIPS1_mask, 'contiguous_regions'), 'all2', 'colorbar');
```

% 'all2' option: close to publication-level brain figure

```
close all;
[out, o2] = brain_activations_display(region(tdat), 'all2', 'labels', 'fontsize', 30);
```

Grouping contiguous voxels: 51 regions

```
*****
You selected to 'all2' option. It will draw four sagittal slices and six axial slices with two surface map
If you want to specify x and z for sagittal and axial slices, you can use 'all2_xyz' option.
E.g., 'all2_xyz', [-5 2 -35 35 -30:12:60], first four will be used as x's and the six numbers after that
will be used as z. More than 10 numbers will be ignored.
```

```
*****
Using custom color maps.
Getting heat-mapped colors
Building color change function call
Loading surface image
Running color change.
eval: [c,alld] = getVertexColors_cocoan(xyz, p, mycolors, [.5 .5 .5], 3,'colorscale',actcolors,'fsavg_ric
Main color vertices: 163842 vertices. selecting: 53063
```

Calculating distance...
Finished!

Using custom color maps.

Getting heat-mapped colors

Building color change function call

Loading surface image

Running color change.

eval: [c,alld] = getVertexColors_cocoan(xyz, p, mycolors, [.5 .5 .5], 3,'colorscale',actcolors,'fsavg_le

Main color vertices: 163842 vertices. selecting: 53063

Calculating distance...

Finished!

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

sagittal montage: 1295 voxels displayed, 51768 not displayed on these slices

sagittal montage: 1402 voxels displayed, 51661 not displayed on these slices

sagittal montage: 1077 voxels displayed, 51986 not displayed on these slices

sagittal montage: 932 voxels displayed, 52131 not displayed on these slices

axial montage: 630 voxels displayed, 52433 not displayed on these slices

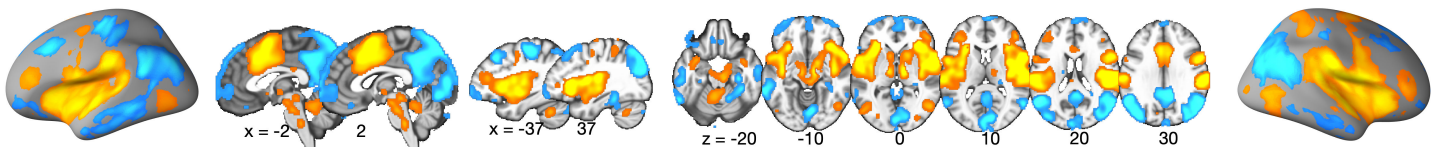
axial montage: 1134 voxels displayed, 51929 not displayed on these slices

axial montage: 1479 voxels displayed, 51584 not displayed on these slices

axial montage: 1434 voxels displayed, 51629 not displayed on these slices

axial montage: 1334 voxels displayed, 51729 not displayed on these slices

axial montage: 1120 voxels displayed, 51943 not displayed on these slices



```
close all;
```

```
[out, o2] = brain_activations_display(region(tdat), 'all2');
```

Grouping contiguous voxels: 51 regions

You selected to 'all2' option. It will draw four sagittal slices and six axial slices with two surface maps.

If you want to specify x and z for sagittal and axial slices, you can use 'all2_xyz' option.

E.g., 'all2_xyz', [-5 2 -35 35 -30:12:60], first four will be used as x's and the six numbers after that will be used as z. More than 10 numbers will be ignored.

Using custom color maps.

Getting heat-mapped colors

Building color change function call

Loading surface image

Running color change.

eval: [c,alld] = getVertexColors_cocoan(xyz, p, mycolors, [.5 .5 .5], 3,'colorscale',actcolors,'fsavg_rig

Main color vertices: 163842 vertices. selecting: 53063

Calculating distance...

Finished!

Using custom color maps.

Getting heat-mapped colors

Building color change function call

Loading surface image

Running color change.

eval: [c,alld] = getVertexColors_cocoan(xyz, p, mycolors, [.5 .5 .5], 3,'colorscale',actcolors,'fsavg_le

Main color vertices: 163842 vertices. selecting: 53063

Calculating distance...

Finished!

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

Load underlay. Define axes. Ready.

sagittal montage: 1295 voxels displayed, 51768 not displayed on these slices

sagittal montage: 1402 voxels displayed, 51661 not displayed on these slices

sagittal montage: 1077 voxels displayed, 51986 not displayed on these slices

sagittal montage: 932 voxels displayed, 52131 not displayed on these slices

axial montage: 630 voxels displayed, 52433 not displayed on these slices

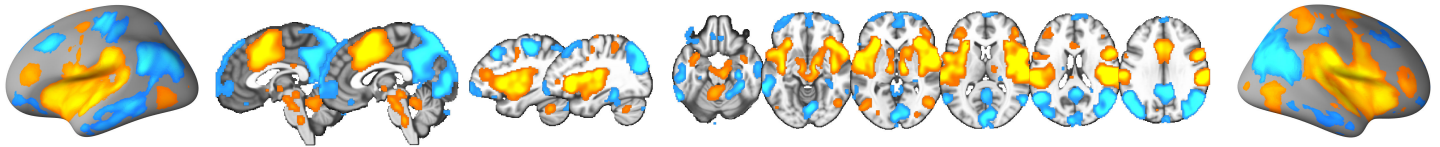
axial montage: 1134 voxels displayed, 51929 not displayed on these slices

axial montage: 1479 voxels displayed, 51584 not displayed on these slices

axial montage: 1434 voxels displayed, 51629 not displayed on these slices

axial montage: 1334 voxels displayed, 51729 not displayed on these slices

axial montage: 1120 voxels displayed, 51943 not displayed on these slices



```
% 'surface_only', 'surface_all'
```

```
close all;
```

```
[out, o2] = brain_activations_display(region(tdat), 'surface_only', 'surface_all');
```

Grouping contiguous voxels: 51 regions

Using custom color maps.

Getting heat-mapped colors

Building color change function call

Loading surface image

Running color change.

eval: [c,alld] = getVertexColors_cocoan(xyz, p, mycolors, [.5 .5 .5], 3,'colorscale',actcolors,'fsavg_le

Main color vertices: 163842 vertices. selecting: 53063

Calculating distance...

Finished!

Using custom color maps.

Getting heat-mapped colors

Building color change function call

Loading surface image

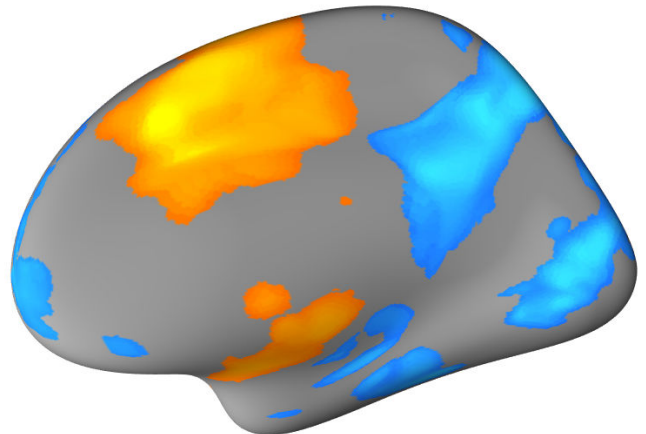
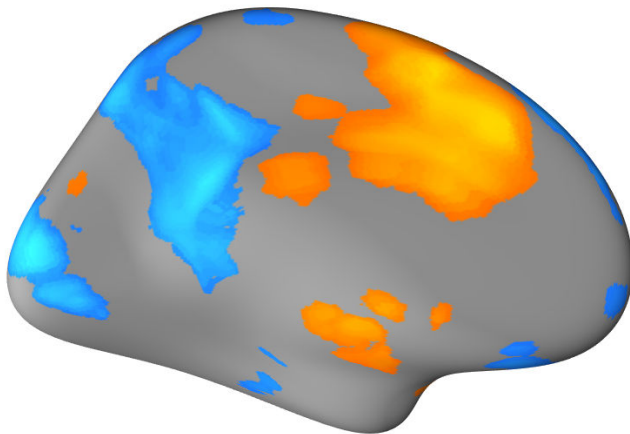
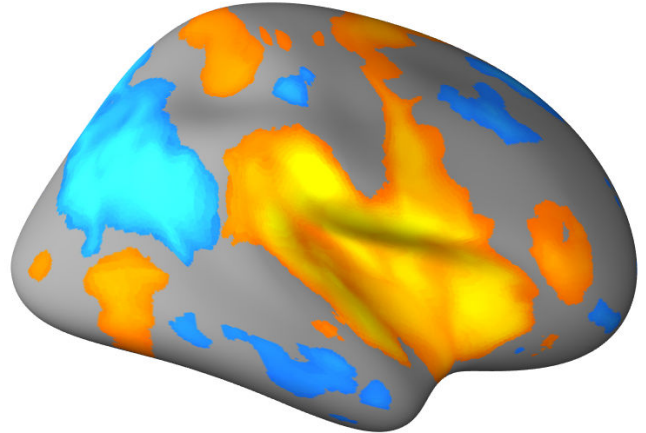
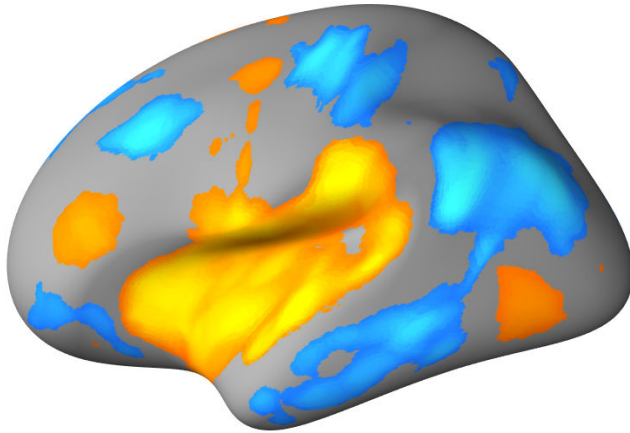
Running color change.

```
eval: [c,alld] = getVertexColors_cocoan(xyz, p, mycolors, [.5 .5 .5], 3,'colorscale',actcolors,'fsavg_rig
```

Main color vertices: 163842 vertices. selecting: 53063

Calculating distance...

Finished!



3. glass_brain_network.m in CocoanCORE

```
r = region(tdat);
```

Grouping contiguous voxels: 51 regions

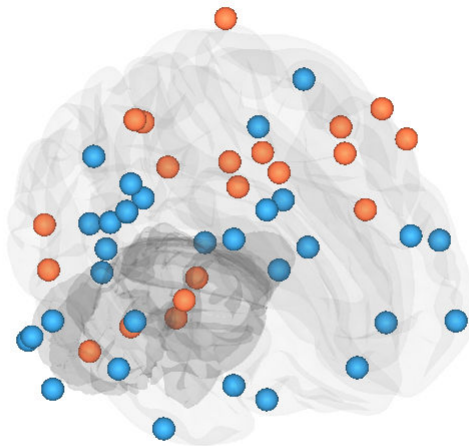
```
for i = 1:numel(r)  
    rsign(i,1) = sign(r(i).Z(1));
```



```
end
```

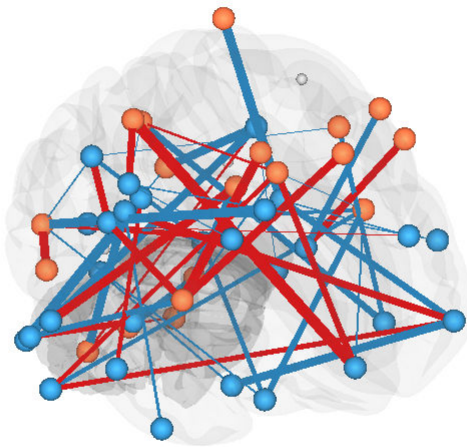
```
cluster_idx = double(rsign==-1) + double(rsign==1)*2;  
cols = [0.1961    0.5333    0.7412;  
        0.9569    0.4275    0.2627];  
close all;  
h = glass_brain_network(r, 'group', cluster_idx, 'colors', cols, 'radius', 4, 'sphere'
```

```
basecolor = 1x3  
    0.5000    0.5000    0.5000  
Main color vertices: Coords is empty. Nothing to plot.  
Running color change.  
basecolor = 1x3  
    0.5000    0.5000    0.5000  
Main color vertices: Coords is empty. Nothing to plot.  
Running color change.
```



```
% create a random positive and negative weights  
w = rand(numel(r));  
w = rand(numel(r)) .* (double(w>0.993)-double(w<0.007));  
w = reformat_r_new(w, 'symmetric_sum');  
close all;  
h = glass_brain_network(r, 'group', cluster_idx, 'colors', cols, 'sphere', 'edge_weight'  
  
basecolor = 1x3  
    0.5000    0.5000    0.5000  
Main color vertices: Coords is empty. Nothing to plot.  
Running color change.  
basecolor = 1x3  
    0.5000    0.5000    0.5000
```

Main color vertices: Coords is empty. Nothing to plot.
Running color change.



4. boxplot_wani_2016.m

```
% mask atlas: CANlab_combined_atlas_object_2018.mat
% key functions: select_atlas_subset,
% apply_mask or extract_roi_averages

load(which('CANlab_combined_atlas_object_2018.mat'));
thal = select_atlas_subset(atlas_obj, {'Thal'});
nac = select_atlas_subset(atlas_obj, {'NAC'});
amyg = select_atlas_subset(atlas_obj, {'Amygdala'});

thal = remove_empty(thal);
nac = remove_empty(nac);
amyg = remove_empty(amyg);

dat_thal = apply_mask(dat, thal);
dat_nac = apply_mask(dat, nac);
dat_amyg = apply_mask(dat, amyg);

% key functions: ttest, plot_specificity_box
temp_idx = repmat((1:6)', 1, 33);
temp_idx = temp_idx(:);

dat_thal_heatlv2 = dat_thal.dat(:, temp_idx==2);
```

```

dat_thal_heatlv5 = dat_thal.dat(:, temp_idx==5);

dat_nac_heatlv2 = dat_nac.dat(:, temp_idx==2);
dat_nac_heatlv5 = dat_nac.dat(:, temp_idx==5);

dat_amyg_heatlv2 = dat_amyg.dat(:, temp_idx==2);
dat_amyg_heatlv5 = dat_amyg.dat(:, temp_idx==5);

```

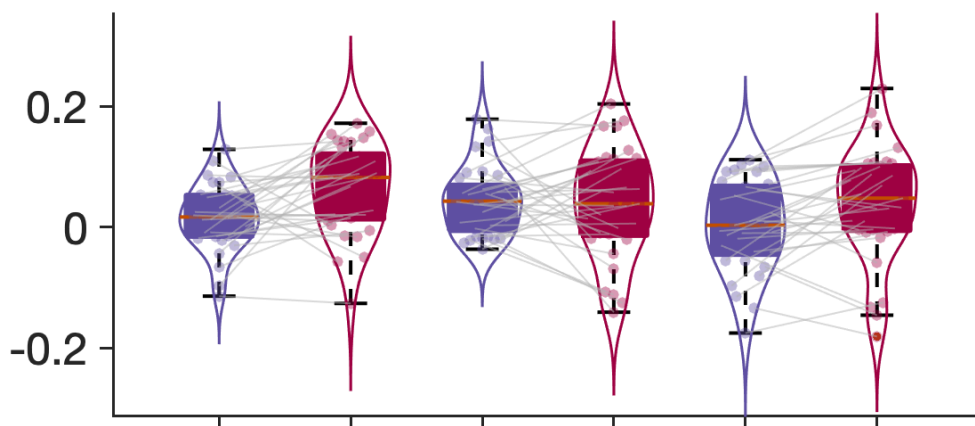
```

x = [mean(dat_thal_heatlv2)' mean(dat_thal_heatlv5)' mean(dat_nac_heatlv2)' mean(dat_nac_heatlv5)'];

col = [0.3686    0.3098    0.6353
       0.6196    0.0039    0.2588
       0.3686    0.3098    0.6353
       0.6196    0.0039    0.2588
       0.3686    0.3098    0.6353
       0.6196    0.0039    0.2588];

close all;
[handles, dot_locs] = boxplot_wani_2016(x, 'color', col, 'linewidth', 2, 'boxlinewidth', 2);
for i = 1:2:5
    hold on;
    line([dot_locs.x{i} dot_locs.x{i+1}]', [dot_locs.y{i} dot_locs.y{i+1}]', 'linewidth', 2);
end

```



```

drawnow; close all;

```

```

% only violin
x = [mean(dat_thal_heatlv2)' mean(dat_thal_heatlv5)' mean(dat_nac_heatlv2)' mean(dat_nac_heatlv5)'];

col = [0.3686    0.3098    0.6353
       0.6196    0.0039    0.2588
       0.3686    0.3098    0.6353
       0.6196    0.0039    0.2588];

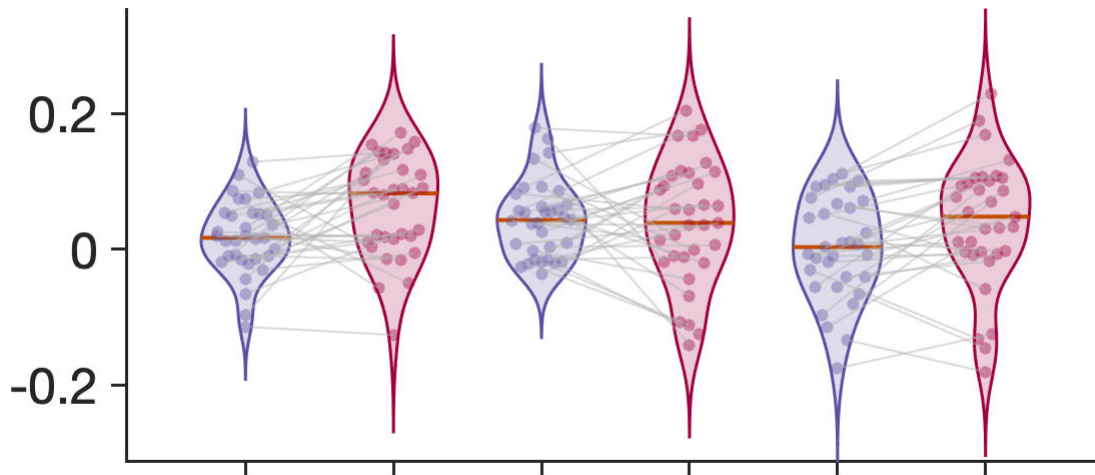
```

```

0.3686    0.3098    0.6353
0.6196    0.0039    0.2588];

close all;
[handles, dot_locs] = boxplot_wani_2016(x, 'color', col, 'linewidth', 2, 'boxlinewidth
for i = 1:2:5
    hold on;
    line([dot_locs.x{i} dot_locs.x{i+1}]' , [dot_locs.y{i} dot_locs.y{i+1}]', 'linewidth
end

```



```

drawnow; close all;

```

```

% how to save this as PDF run this again in the command window
close all;
[handles, dot_locs] = boxplot_wani_2016(x, 'color', col, 'linewidth', 2, 'boxlinewidth
for i = 1:2:5
    hold on;
    line([dot_locs.x{i} dot_locs.x{i+1}]' , [dot_locs.y{i} dot_locs.y{i+1}]', 'linewidth
end

figdir = '/Users/clinpsywoo/Dropbox/2011-yr/Teaching/COCOAN101/L13';
savename = fullfile(figdir, 'boxviolinplot.pdf');

pagesetup(gcf);
saveas(gcf, savename);

pagesetup(gcf);
saveas(gcf, savename);

```

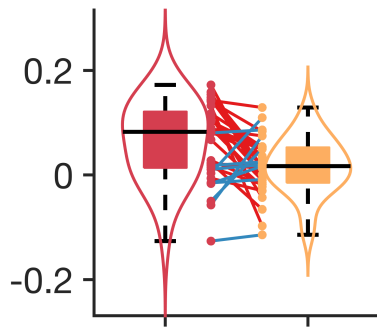
5. plot_specificity_box.m

```

[h, p, ci, tstat] = ttest((mean(dat_heatlv5)-mean(dat_heatlv2))');

```

```
out = plot_specificity_box(mean(dat_heatlv5)', mean(dat_heatlv2)');
```



```
% save PDF
close all;
out = plot_specificity_box(mean(dat_heatlv5)', mean(dat_heatlv2)');
figdir = '/Users/clinpsywoo/Dropbox/2011-yr/Teaching/COCOAN101/L13';
savename = fullfile(figdir, 'plotspecificity.pdf');

pagesetup(gcf);
saveas(gcf, savename);

pagesetup(gcf);
saveas(gcf, savename);
```

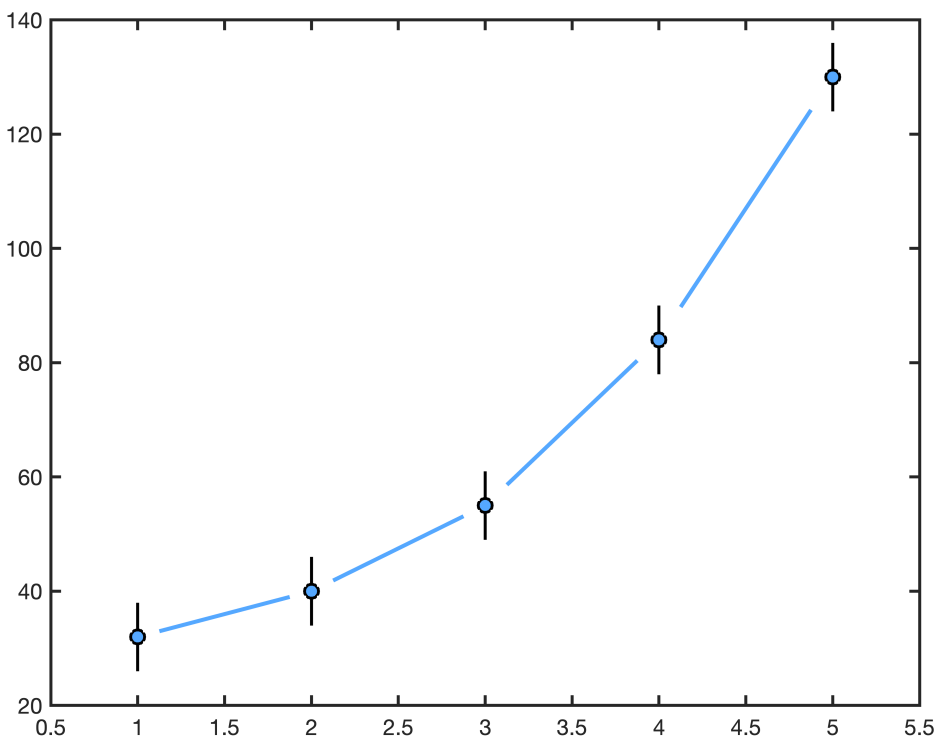
6. sepplot.m

```
x = 1:5; % x values
y = [32 40 55 84 130]; % mean
e = [6 6 6 6 6]; % standard error of the mean

col = [0.3333 0.6588 1.0000];
markercol = col-.2;

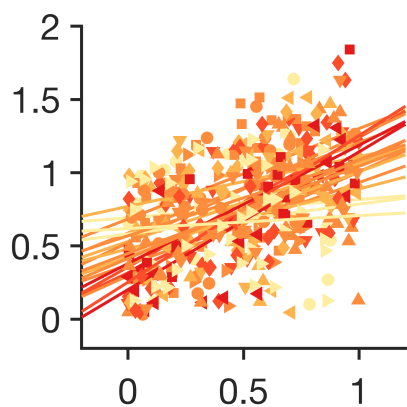
close all;
h = errorbar(x, y, e, 'o', 'color', 'k', 'linewidth', 1.5, 'markersize', 7, 'markerfacecolor', 'k');
hold on;
sepplot(x, y, .75, 'color', col, 'linewidth', 2);
h.CapSize = 0;

set(gca, 'xlim', [.5 5.5], 'linewidth', 1.5);
```



7. plot_y_yfit.m

```
% create fake data
for i = 1:20
    yval{i} = rand(20,1);
    yfit{i} = yval{i}.*rand(20,1) + rand(20,1);
end
out = plot_y_yfit(yval, yfit, 'xlim', [-0.2 1.2], 'ylim', [-0.2 2]);
```



```
% save PDF
% figdir = '/Users/clinsywoo/Dropbox/2011-yr/Teaching/COC0AN101/L13';
% savename = fullfile(figdir, 'plotyyfit.pdf');
%
```

```
% pagesetup(gcf);  
% saveas(gcf, savename);  
%  
% pagesetup(gcf);  
% saveas(gcf, savename);
```