Table of Contents

Spontaneous and Task-related Activation of Neuronally Correlated Events (STANCE)

Showcases various usage options for modelling and defining activation maps.

author: Dr. Jason E. Hill (post-doc fellow with CNT at TTU) demo_3D_define_act updated 28 MAR 2017

```
close all;
clear all;
currentDir = pwd;
if strcmp(currentDir(end-2:end),'GUI')
    % GUI instance of initialization
    cd ../
    STANCEroot = pwd;
    cd(currentDir)
elseif strcmp(currentDir(end-5:end),'STANCE')
    STANCEroot = pwd;
elseif strcmp(currentDir(end-16:end),'scripts_for_demos')
    cd ../
    STANCEroot = pwd;
else
    hSTANCE = msgbox('Please select the STANCE directory');
    uiwait(hSTANCE);
    currPath = fileparts(mfilename('fullpath'));
    STANCEroot = uigetdir(currPath, 'Add STANCE filepath');
end
cd(STANCEroot)
addpath(genpath(pwd));

% Load STANCE globals ...
if ~exist('STANCE.mat','file')
    STANCE_initialize_STANCE;
    load('STANCE.mat');
else
    load('STANCE.mat');
end
% NOTE: Must add SPM version to filepath prior to usage
addpath(SPMpath);
if exist(spm('Dir'),'dir')
```

```
    display('o SPM installation found.')
else
    warning('SPM installation not found. Please add to MATLAB filepath
 or install.')
    warning('SPM8 installation: http://www.fil.ion.ucl.ac.uk/spm/
software/spm8/')
    exit
end
```

```
o SPM installation found.
```

Turn off warnings ...

... OpenGl warnings

```
warning('off','MATLAB:opengl:StartupBlacklistedNoSetting');
warning('off', 'MATLAB:hg:AutoSoftwareOpenGL');
% ... finite warning
warning('off', 'MATLAB:FINITE:obsoleteFunction');
% ... NIFTI class warnings when loading SPM mat files
warning('off', 'MATLAB:unknownElementsNowStruc');
warning('off', 'MATLAB:dispatcher:ShadowedMEXExtension');
warning('off', 'MATLAB:pfileOlderThanMfile');
% ... removing files from path
warning('off', 'MATLAB:RMDIR:RemovedFromPath');
```

Examples of 2D activation maps

```
uiwait(msgbox('Demo examples of geometric shapes.','Shapes','modal'));

% define target space
dimensions2D = [101 101];
origin2D = [50 50];

% square example
task.name = 'usage example';
task.activation.region   = 'Square';
task.activation.shape    = 'square';
task.activation.proportion    = 25;      % no volume -> use as length
 of side

task.map = STANCE_make_activation_map(dimensions2D, origin2D,
 task.activation);

f1 = figure;
imshow(imrotate(task.map,90),[]), drawnow;
TITLE = 'Square shape';
title(TITLE)
squareMap = task.map;
movegui(f1,'northwest');

% Rectangle example
task.name = 'usage example';
```

```
task.activation.region = 'Rectangle';
task.activation.rotation = +30;  % degrees
task.activation.shape = 'rectangle';
task.activation.proportion = [25 15]; % no volume -> use as length of
 sides
task.map = STANCE_make_activation_map(dimensions2D, origin2D,
 task.activation);

f2 = figure;
imshow(imrotate(task.map,90),[]), drawnow;
TITLE = 'Rectangle shape';
title(TITLE)
movegui(f2,'north');

clear task
% Oval example
task.name = 'usage example';
task.activation.region = 'Oval';
task.activation.volume = 1000;
task.activation.center = [-10 +10]; % relative to the origin
task.activation.shape = 'ellipse';
task.activation.proportion = [5 3]; % ratios of major and minor axes
task.activation.falloff = 0.005;    % Gaussian falloff from center, in
 [0,1]
task.activation.minimum = 0.25;     % falloff minimum value, in [0,1]
task.map = STANCE_make_activation_map(dimensions2D, origin2D,
 task.activation);

f3 = figure;
imshow(imrotate(task.map,90),[]), drawnow;
TITLE = 'Oval shape';
title(TITLE)
ovalMap = task.map;
movegui(f3,'northeast');

clear task
% Astroid example
task.name = 'usage example';
task.activation.region = 'Astroid';
task.activation.volume = 1000;      % no volume -> use lengths
task.activation.shape = 'astroid';
task.activation.proportion = [1 1]; % ratios of major and minor axes
task.map = STANCE_make_activation_map(dimensions2D, origin2D,
 task.activation);

f4 = figure;
imshow(imrotate(task.map,90),[]), drawnow;
TITLE = ['Astroid shape'];
title(TITLE)
movegui(f4,'east');

clear task
% Squircle example
task.name = 'usage example';
```

```
task.activation.region   = 'Squircle';
task.activation.volume   = 1000;
task.activation.shape = 'squircle';
task.activation.proportion = [1 1]; % aspect ratios of major and minor
 axes
task.map = STANCE_make_activation_map(dimensions2D, origin2D,
 task.activation);

f5 = figure;
imshow(imrotate(task.map,90),[]), drawnow;
TITLE = ['Squircle shape'];
title(TITLE)
movegui(f5,'southeast');

clear task
% Diamond example
task.name = 'usage example';
task.activation.region = 'Diamond';
task.activation.volume = 1000;
task.activation.shape = 'diamond';
task.activation.proportion = [3 1]; % ratios of major and minor axes
task.map = STANCE_make_activation_map(dimensions2D, origin2D,
 task.activation);

f6 = figure;
imshow(imrotate(task.map,90),[]), drawnow;
TITLE = 'Diamond  shape';
title(TITLE)
movegui(f6,'south');

clear task
% Superellipse example
task.name = 'usage example';
task.activation.region = 'Superellipse';
task.activation.volume = 1000;
task.activation.shape = {'superellipse',1.5};
task.activation.proportion = [2 1];           % ratios of major and
 minor axes
task.map = STANCE_make_activation_map(dimensions2D, origin2D,
 task.activation);

f7 = figure;
imshow(imrotate(task.map,90),[]), drawnow;
TITLE = 'Superellipse (n = 1.5)  shape';
title(TITLE)
movegui(f7,'southwest');
```

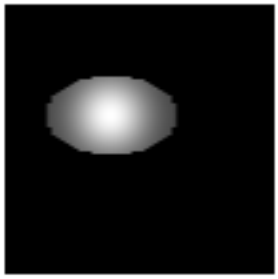
```
o Specifying square template
o Performing affine transformation.
o Building activation map.
o Specifying square template
o Performing affine transformation.
o Building activation map.
o Specifying circular template
```

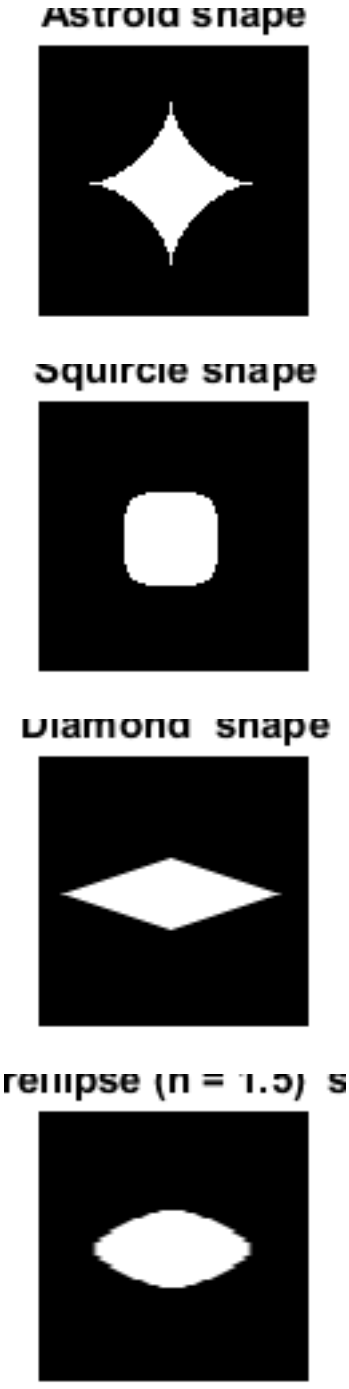
o Performing affine transformation.
o Building activation map.
o Specifying circular template
o Performing affine transformation.
o Building activation map.
o Specifying circular template
o Performing affine transformation.
o Building activation map.
o Specifying circular template
o Performing affine transformation.
o Building activation map.
o Specifying circular template
o Performing affine transformation.
o Building activation map.

Square shape

Rectangle shape

Oval shape





Combinations via fuzzy logical operators

```
uiwait(msgbox('Demo examples of combos from logical
 operations.','Combos','modal'));

% fuzzy logical NOT
map = (1-ovalMap);
map = STANCE_combine_maps('NOT',ovalMap);
```

```
f8 = figure;
imshow(map,[])
title('NOT oval'), drawnow;
movegui(f8,'west');


maps(1,:,:) = squareMap;
maps(2,:,:) = ovalMap;
mapsNOT2(1,:,:) = squareMap;
mapsNOT2(2,:,:) = (1-ovalMap);
mapsNOT1(1,:,:) = (1-squareMap);
mapsNOT1(2,:,:) = ovalMap;
mapsXOR(1,:,:)  = squareMap.*(1-ovalMap);
mapsXOR(2,:,:)  = ovalMap.*(1-squareMap);

% fuzzy logical OR
map = squeeze(max(maps));
map = STANCE_combine_maps('OR',squareMap,ovalMap);
f9 = figure;
imshow(imrotate(map,90),[])
title('oval OR square'), drawnow;
movegui(f9,'northwest');

% fuzzy logical XOR
map = squeeze(max(mapsXOR));
map = STANCE_combine_maps('XOR',squareMap,ovalMap);
f10 = figure;
imshow(imrotate(map,90),[])
title('oval XOR square'), drawnow;
movegui(f10,'north');

% fuzzy logical AND
map = squeeze(min(maps));
map = STANCE_combine_maps('AND',squareMap,ovalMap);
f11 = figure;
imshow(imrotate(map,90),[])
title('oval AND square'), drawnow;
movegui(f11,'northeast');

% fuzzy logical NAND on left
map = squeeze(min(mapsNOT1));
f13 = figure;
imshow(imrotate(map,90),[])
title('oval NAND square'), drawnow;
movegui(f13,'east');

% fuzzy logical NAND on right
map = squeeze(min(mapsNOT2));
f14 = figure;
imshow(imrotate(map,90),[])
title('square NAND oval'), drawnow;
movegui(f14,'southeast');

% fuzzy logical NAND implemented
```

```
map = (1 - squeeze(max(mapsNOT1)));
map = STANCE_combine_maps('NAND',squareMap,ovalMap);
f15 = figure;
imshow(imrotate(map,90),[])
title('square NAND oval'), drawnow;
movegui(f15,'southeast');

% fuzzy logical exclude other
map = (1 - squeeze(max(mapsNOT2)));
map = STANCE_combine_maps('NAND',ovalMap,squareMap);
f16 = figure;
imshow(imrotate(map,90),[]),
title('oval NAND square'), drawnow;
movegui(f16,'east');
```

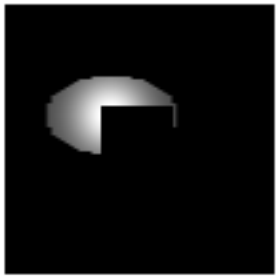
NOT oval

oval OR square

oval XOR square

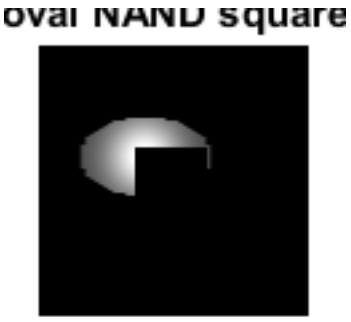
oval AND square

oval NAND square



square NAND oval

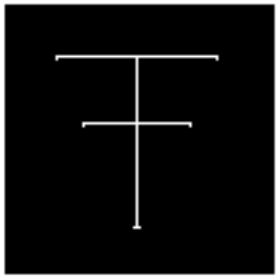
square NAND oval



Custom activation by manually building list of coordinates

```
uiwait(msgbox('Demo example of user-defined custom
 map.','Custom','modal'));

% define the voxels coordinates with list
centers(1:65,1)    = 51;
centers(1:65,2)    = 19:83;
centers(65:125,1)  = 21:81;
centers(65:125,2)  = 83;
centers(126:166,1) = 31:71;
centers(126:166,2) = 58;
centers(167,1)     = 50;
centers(167,2)     = 19;
centers(168,1)     = 52;
centers(168,2)     = 19;
centers(169,1)     = 31;
centers(169,2)     = 57;
centers(170,1)     = 71;
centers(170,2)     = 57;
centers(171,1)     = 21;
centers(171,2)     = 82;
centers(172,1)     = 81;
centers(172,2)     = 82;
centers(:,1) = centers(:,1) - 51; % subtract off origin
centers(:,2) = centers(:,2) - 51; % subtract off origin

clear task;
% mask example
task.name = 'usage example';
task.activation.region   = 'TT';
task.activation.center   = centers;
task.activation.shape    = 'mask';
task.map = STANCE_make_activation_map(dimensions2D, origin2D,
 task.activation);

h_custom  = figure;
imshow(imrotate(task.map,90),[]), drawnow;
TITLE = 'User-specified custom activation map';
```

```
title(TITLE), drawnow;
movegui(h_custom,'center');

o Building raw activation map.
```



Load MNI brain volume

```
uiwait(msgbox('Demo example building activation regions from
 atlas.','Atlas ROIs','modal'));

% show MNI volume conformed to BrainWEB dimensions
[V_MNI,Y_MNI] = STANCE_load_volume(filenameMNI);
MNI_dim = V_MNI.dim;
MNI_mat = V_MNI.mat;
origin  = abs(V_MNI.mat(1:3,4))';

[~,I_max] = max(sum(sum(Y_MNI)));
showSlice = I_max(1);
%
% imshow(imrotate(Y_MNI(:,:,showSlice),90),[]);
% TITLE = ['MNI152 brain, A slice: ',num2str(showSlice)];
% title(TITLE), drawnow;
```

Build activation regions from atlas ROIs

```
dimensions = size(Y_MNI);
origin = round(abs(V_MNI.mat(1:3,4)))';
clear task;
% AAL: Precuneus_L example
task.name = 'Precuneus_L';
task.activation.region   = 'aal';   % the name of the atlas
task.activation.volume   = 67;      % the ROI label
task.activation.shape    = 'atlas';
task.map = STANCE_make_activation_map(dimensions, origin,
 task.activation);

[~,I_max] = max(sum(sum(task.map)));
showSliceTA = I_max(1);

%figure, imshow(imrotate(task.map(:,:,showSliceTA),90),[]), drawnow;
```

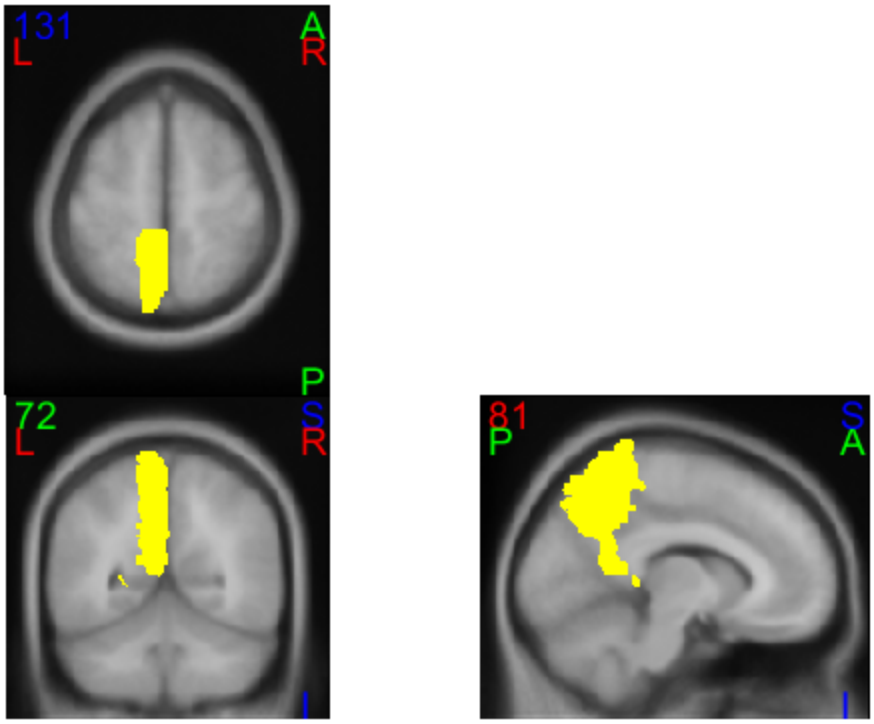
```
%TITLE = ['Activation of the L Precuneus: axial slice
 ',num2str(showSliceTA)];
%title(TITLE)

% h_task =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTA,3);
% title(TITLE)
%
% [~,I_max] = max(sum(sum(task.map,2),3));
% showSliceTS = I_max(1);
% h_task_TS_R =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTS,1);
% TITLE = ['Activation of the L Precuneus: sagittal slice
 ',num2str(showSliceTS)];
% title(TITLE)
%
% [~,I_max] = max(sum(sum(task.map),3));
% showSliceTC = I_max(1);
% h_task_TC =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTC,2);
% TITLE = ['Activation of the L Precuneus: coronal slice
 ',num2str(showSliceTC)];
% title(TITLE)

TITLE = {'Left Precuneus from the AAL'; 'activation template in MNI'};
h_task_AAL_67 = STANCE_display_activation_slice(Y_MNI,task.map,[],[]);
title(TITLE)
movegui(h_task_AAL_67,'west');

% Brodmann: BA10 example
clear task;
task.name = 'BA10';
task.activation.region   = 'brodmann'; % the name of the atlas
task.activation.volume   = 10;         % the ROI label
task.activation.shape    = 'atlas';
task.map = STANCE_make_activation_map(dimensions, origin,
 task.activation);

[~,I_max] = max(sum(sum(task.map)));
showSliceTA = I_max(1);

% figure, imshow(imrotate(task.map(:,:,showSliceTA),90),[]), drawnow;
% TITLE = ['Activation of the BA10: axial slice
 ',num2str(showSliceTA)];
% title(TITLE)

% h_task =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTA,3);
% title(TITLE)
%
% [~,I_max] = max(sum(sum(task.map,2),3));
% showSliceTS = I_max(1);
% h_task_TS_R =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTS,1);
```

```
% TITLE = ['Activation of the BA10: sagittal slice
 ',num2str(showSliceTS)];
% title(TITLE)
%
% [~,I_max] = max(sum(sum(task.map),3));
% showSliceTC = I_max(1);
% h_task_TC =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTC,2);
% TITLE = ['Activation of the BA10: coronal slice
 ',num2str(showSliceTC)];
% title(TITLE)


h_task_BA10 = STANCE_display_activation_slice(Y_MNI,task.map,[],[]);
TITLE = {'Brodmann area # 10'; 'activation template in MNI'};
title(TITLE)
movegui(h_task_BA10,'east');

% HarvardOxford: L Amygdala example
task.name = 'L Amygdala';
task.activation.region     = 'HarvardOxford'; % the name of the atlas
task.activation.volume     = -18;             % the ROI label,
 negative -> subcortical
task.activation.shape      = 'atlas';
task.activation.proportion = 25;              % probability threshold
task.map = STANCE_make_activation_map(dimensions, origin,
 task.activation);

% [~,I_max] = max(sum(sum(task.map)));
% showSliceTA = I_max(1);
%
% figure, imshow(imrotate(task.map(:,:,showSliceTA),90),[]), drawnow;
% TITLE = ['Activation of the L Amygdala: axial slice
 ',num2str(showSliceTA)];
% title(TITLE)
%
% h_task =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTA,3);
% title(TITLE)
%
% [~,I_max] = max(sum(sum(task.map,2),3));
% showSliceTS = I_max(1);
% h_task_TS_R =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTS,1);
% TITLE = ['Activation of the L Amygdala: sagittal slice
 ',num2str(showSliceTS)];
% title(TITLE)
%
% [~,I_max] = max(sum(sum(task.map),3));
% showSliceTC = I_max(1);
% h_task_TC =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTC,2);
% TITLE = ['Activation of the L Amygdala: coronal slice
 ',num2str(showSliceTC)];
```

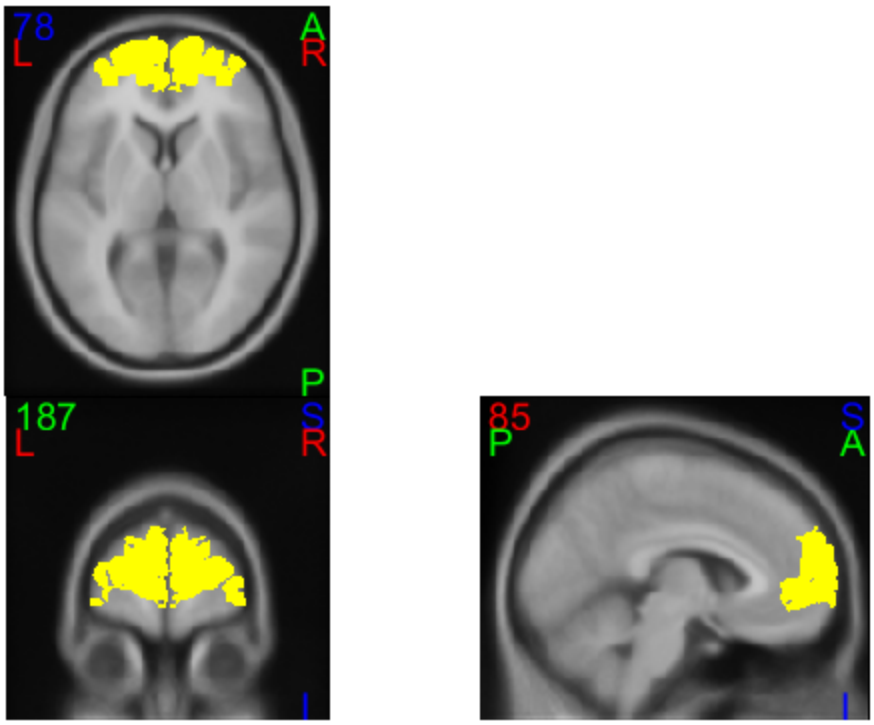
```
% title(TITLE)

h_task_HO25sc18 = STANCE_display_activation_slice(Y_MNI,task.map,[],
[]);
TITLE = {'Left Amygdala (Harvard Oxford 25%)'; 'activation template in
 MNI'};
title(TITLE)
movegui(h_task_HO25sc18,'south');


% Brainnetome: example
task.name = 'L rostral cuneus gyrus of the MedioVentral Occipital
 Cortex';
task.activation.region     = 'Brainnetome'; % the name of the atlas
task.activation.volume     = 191;           % the ROI label, negative
 -> subcortical
task.activation.shape      = 'atlas';
task.activation.proportion = 25;            % probability threshold
 (only choice at this time)
task.map = STANCE_make_activation_map(dimensions, origin,
 task.activation);


h_task_Brainnetome25_191 =
 STANCE_display_activation_slice(Y_MNI,task.map,[],[]);
TITLE = {'L rCunG of MVOcC (Brainnetome)'; 'activation template in
 MNI'};
title(TITLE)
movegui(h_task_Brainnetome25_191,'north');


% Craddock: example
task.name = 'L Somato-motor Cortex';
task.activation.region     = 'Craddock'; % the name of the atlas
task.activation.volume     = 90;         % the ROI label, negative ->
 subcortical
task.activation.shape      = 'atlas';
task.activation.proportion = 200;        % here the number of ROI (can
 also be 400)
task.map = STANCE_make_activation_map(dimensions, origin,
 task.activation);


h_task_CC200_90 = STANCE_display_activation_slice(Y_MNI,task.map,[],
[]);
TITLE = {'Left Somato-motor (Craddock 200)'; 'activation template in
 MNI'};
title(TITLE)
movegui(h_task_CC200_90,'center');

o Loading atlas ROI mask.
o Loading atlas ROI mask.
o Loading atlas ROI mask.
o Loading atlas ROI mask.
```

o Loading atlas ROI mask.

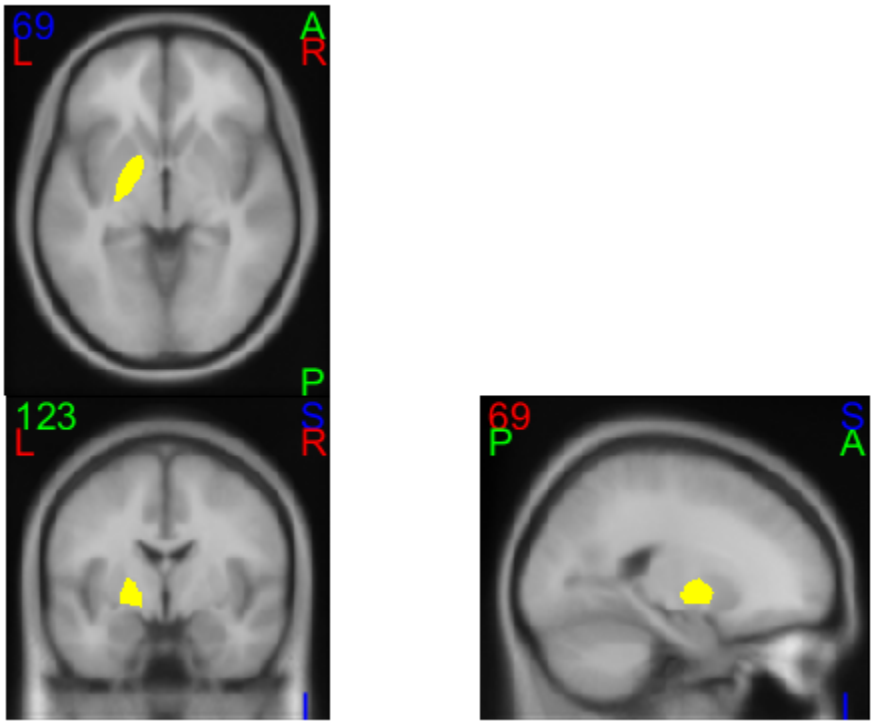
Left Precuneus from the AAL activation template in MNI



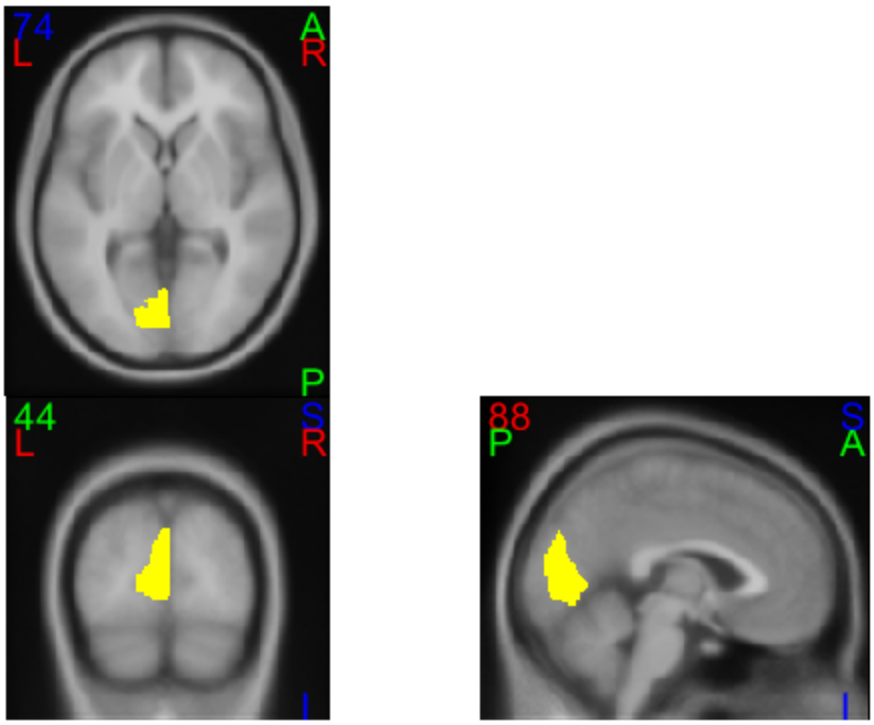
Brodmann area # 10
activation template in MNI



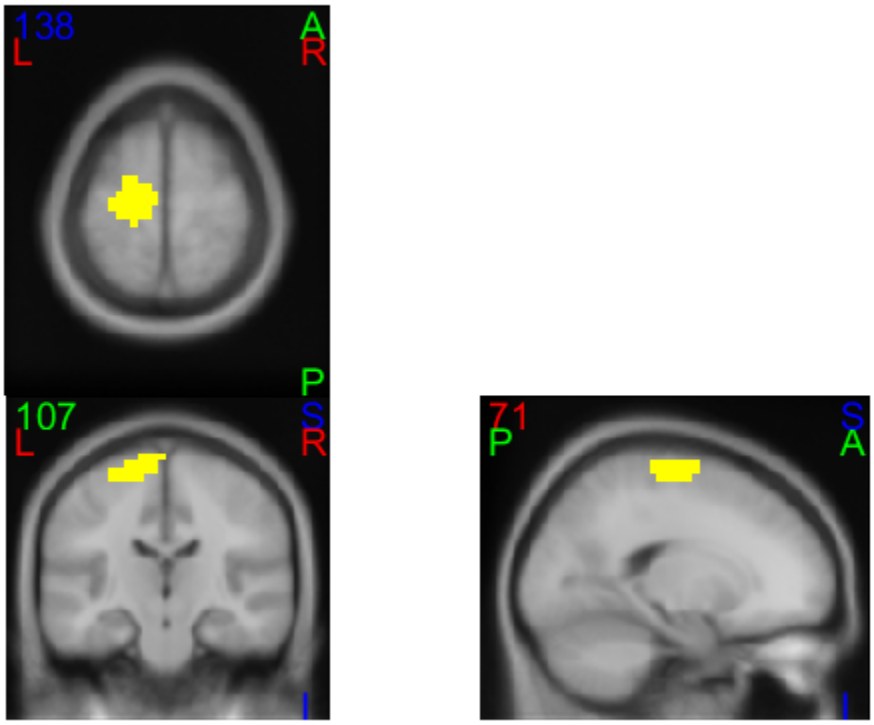
Left Amygdala (Harvard Oxford 25%)
activation template in MNI



L rCunG of MVOcC (Brainnetome) activation template in MNI



Left Somato-motor (Craddock 200) activation template in MNI



Clean up and return

```
clear('V_MNI','Y_MNI')
cd(currentDir)
```

Published with MATLAB® R2016a