Table of Contents

..... 1
Spontaneous and Task-related Activation of Neuronally Correlated Events (STANCE) 1
Turn off warnings 2
Select subject by index (originally there are 20 subjects to choose from) 2
Build activation regions by modelling reported results 5
Reslice volumes to functional space according to fMRI scan protocal 9
Load activation map from data files 15
Save, free up memory, and return 17


```
% author: Dr. Jason E. Hill (post-doc fellow with CNT at TTU)
% demo_3D_ex1    updated     28 SEPT 2016
```

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

Illustates how to define a simple finger tapping task activation map, both by modelling from values reported in the literature and from a data file (obtained from Neurosynth).

```
close all;
clear all; %#ok<CLALL>
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

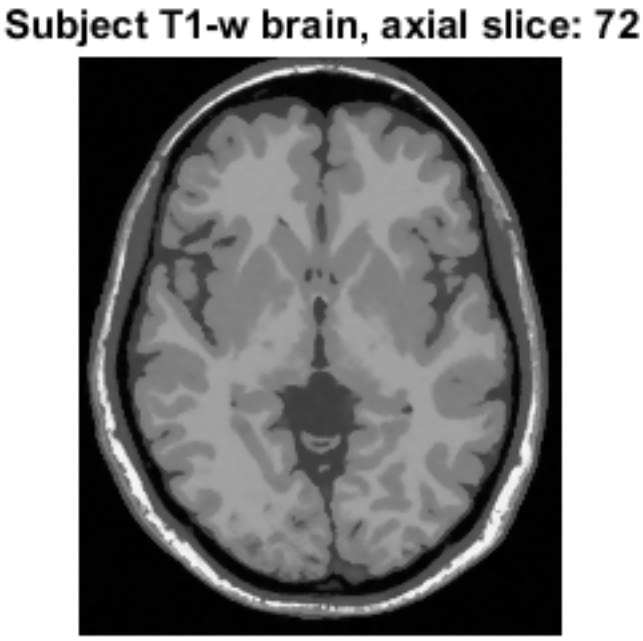
```
warning('off','MATLAB:opengl:StartupBlacklistedNoSetting');
warning('off', 'MATLAB:hg:AutoSoftwareOpenGL');
% ... finite warning
warning('off', 'MATLAB:FINITE:obsoleteFunction');
% ... NIFTI class warnings when loading SPM mat files
warning('off', 'MATLAB:unknownElementsNowStruc');
warning('off',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

Select subject by index (originally there are 20 subjects to choose from)

```
subject_brain = 20;
Now_sss = [1 1 1];
STANCE_new_session(1,1,1)
filepathOut = STANCE_genpath(Now_sss);
makeFMRI = true;

% show MNI volume conformed to BrainWEB dimensions
[V_MNI,Y_MNI] = STANCE_load_volume(filenameMNI);
display('The matrix dimensions of the MNI152 brain:')
MNI_dim = V_MNI.dim
display('The homographic matrix of the MNI152 brain:')
MNI_mat = V_MNI.mat
display('The origin (AC location) of the MNI152 brain:')
origin  = abs(V_MNI.mat(1:3,4))'

[~,I_max] = max(sum(sum(Y_MNI)));
showSlice = I_max(1);

% figure, imshow(imrotate(Y_MNI(:,:,showSlice),90),[]), drawnow;
% TITLE = ['MNI152 brain, A slice: ',num2str(showSlice)];
```

```
% title(TITLE)

% load the T1w data for subject, for display purposes
[V_T1w,Y_T1w] = STANCE_choose_subject(subject_brain,'T1');

display('The matrix dimensions of the T1-w image from header:')
T1w_dim = V_T1w.dim  % dimensions of T1-w volume
display('The homographic matrix of the T1-w image from header:')
T1w_mat = V_T1w.mat  % 4x4 homographic matrix relating indeces to
 real-world coordinates
f1 = figure;
subplot(2,1,2)
imshow(imrotate(Y_T1w(:,:,showSlice),90),[]), drawnow;
TITLE = ['Subject T1-w brain, axial slice: ',num2str(showSlice)];
title(TITLE)
truesize
movegui(f1,'northwest');

% retrieve transformation matrix mapping MNI152 to subjects' native
 spaces
display('The transformation matrix mapping MNI152 to the native spaces
 of the subjects:');
M = M_array(:,:,subject_brain)

[V_MNI_reg,Y_MNI_reg] = STANCE_register_MNI(V_T1w.fname,M);

% figure, imshow(imrotate(Y_MNI_reg(:,:,showSlice),90),[]), drawnow;
% TITLE = ['MNI152 registered to subject brain, A slice:
 ',num2str(showSlice)];
% title(TITLE)

display('The matrix dimensions of the T1-w image:')
dimensions = size(Y_T1w)
display('The origin (AC location) of the T1-w image:')
origin = round(abs(V_T1w.mat(1:3,4)))'

Erasing files in C:\spm\STANCE/fMRI/study001/subject0001/session001.

ans =

     1     1   305

The matrix dimensions of the MNI152 brain:

MNI_dim =

   181   217   181

The homographic matrix of the MNI152 brain:

MNI_mat =

     1     0     0   -91
     0     1     0  -127
```

```
     0     0     1   -73
     0     0     0     1

The origin (AC location) of the MNI152 brain:

origin =

    91   127    73

The matrix dimensions of the T1-w image from header:

T1w_dim =

   181   217   181

The homographic matrix of the T1-w image from header:

T1w_mat =

    1.0000         0         0  -90.7500
         0    1.0000         0 -126.7500
         0         0    1.0000  -73.2500
         0         0         0    1.0000

The transformation matrix mapping MNI152 to the native spaces of the
 subjects:

M =

    1.0000    0.0049   -0.0053    1.0808
   -0.0049    1.0000    0.0066   -0.7607
    0.0054   -0.0066    1.0000    1.3507
         0         0         0    1.0000

The matrix dimensions of the T1-w image:

dimensions =

   181   217   181

The origin (AC location) of the T1-w image:

origin =

    91   127    73
```



Build activation regions by modelling reported results

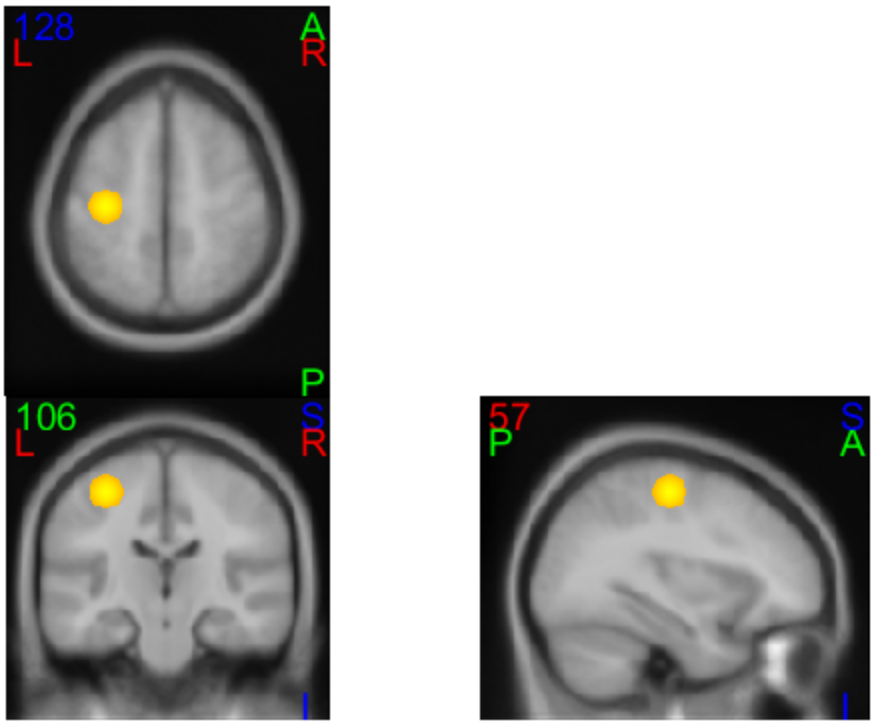
```
uiwait(msgbox('Demo example of a finger opposition task in the
 pSM.','Finger opposition task','modal'));

% see Tables 1. & 3. of "Functional Mapping of Human Sensorimotor
 Cortex with
% 3D BOLD fMRI Correlates Highly With H2150 PET rCBF" in Journal of
 Cerebral Blood Flow and Metabolism
% 16:755-764

% define Finger Opposition task stimulation of the primary
 sensorimotor cortex (PSM)
task.name = 'Finger opposition PSM';
task.activation.region = 'R PSMC';
task.activation.volume = 45.9*3.75^3; % estimated from Table 1 average
 of 45.9 voxels
task.activation.center = round(tal2mni([-33.8,-18.2,51.8]'))';  %
 activation for right-handed subjects, from mean in Table 3.
task.activation.rotation = [0,0,0];   % degrees
task.activation.shape = 'sphere';
task.activation.proportion = [1,1,1]; % aspect ratio
task.activation.falloff = 0.005;      % parameterizes exponential
 falloff about center, in [0,1]
```

```
task.activation.minimum   = 0.2;      % parameterizes exponential
 falloff minimum in [0,1]
% define signal amplitude
task.amplitude = 0.03; % 3% activation

simulations{Now_sss(1)}.name = 'Finger related tasks';
simulations{Now_sss(1)}.task{1} = task;

% left brain component of right-handed task
disp('Defining finger opposition task of PSM activation map...')
task.activation.map = STANCE_make_activation_map(dimensions, origin,
 task.activation);
task.map = task.activation.map;  % more useful when combining maps
%task.activation.map = [];        % clear memory

% find the ammount of gray matter volume for the activation map based
 on MNI tissue priors
task.GMvolume = STANCE_find_GM_volume(task);

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

% figure, imshow(imrotate(task.map(:,:,showSliceTA),90),[]), drawnow;
TITLE = {'Finger opposition task'; 'activation template in MNI'};
htask1 = STANCE_display_activation_slice(Y_MNI,task.map,[],[]);
title(TITLE)
movegui(htask1,'center');

% TITLE = ['Finger opposition task activation of the PSM, S slice'];
% title(TITLE)
% [~,I_max] = max(sum(sum(task.map,2),3));
% showSliceTS = I_max(1);
% h_task = STANCE_display_activation_slice(Y_MNI,task.map,[],1);
% title(TITLE)
%
% [~,I_max] = max(sum(sum(task.map),3));
% showSliceTC = I_max(1);
% TITLE = ['Finger opposition task activation of the PSM, C slice'];
% title(TITLE)
% h_task = STANCE_display_activation_slice(Y_MNI,task.map,[],2);
% title(TITLE)

[V_finger_opposition_reg,Y_finger_opposition_reg] =
 STANCE_register_activation(V_T1w.fname,task);

% figure,
 imshow(imrotate(Y_finger_opposition_reg(:,:,showSliceTA),90),[]),
 drawnow;
% title('Finger opposition task activation of the PSM registered')

%h_task_reg =
 STANCE_display_activation_slice(Y_MNI_reg,Y_finger_opposition_reg,
[]);
%title('Finger opposition task activation of the PSM registered')
```

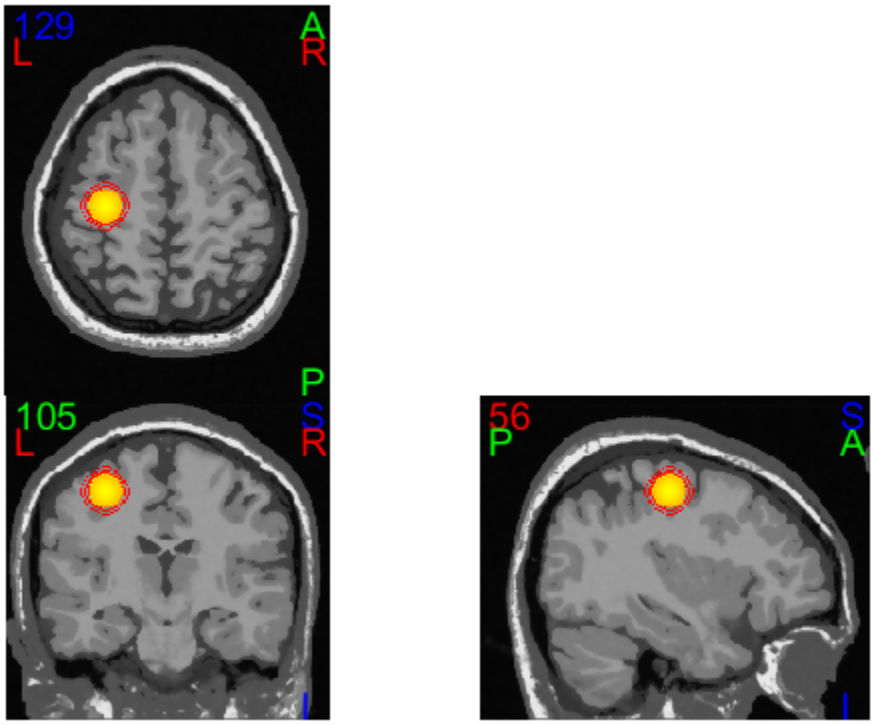
```
% Finger opposition task activation template of the PSM of subject
TITLE = {'Finger opposition task'; 'activation template in subject'};
htask1sub =
 STANCE_display_activation_slice(Y_T1w,Y_finger_opposition_reg,[],[]);
title(TITLE)
movegui(htask1sub,'center');

% make room in memory
if ~strcmp(V_T1w.fname(end-1:end),'gz')
    delete(V_T1w.fname);
end
clear('V_T1w','YT1w');
delete(V_MNI_reg.fname);
clear('V_MNI_reg','Y_MNI_reg')
task.map = [];

% save activation template
task.activation.map = int8(255*task.activation.map);
cd([STANCEroot,'/activations'])
save([task.name,'.mat'],'task')
cd(STANCEroot)
task.activation.map = [];        % clear memory
```

```
Defining finger opposition task of PSM activation map...
o Specifying spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
```

Finger opposition task activation template in MNI



Finger opposition task activation template in subject



Reslice volumes to functional space according to fMRI scan protocal

```
if makeFMRI == true
% default scan settings:
%     voxelSize    = [3 3 3];
%     new_dims     = [64 64 NaN];  % effectively [64 64 40] here
%     tiltAngle    = 15; % degrees
%     voxelSpacing = [0 0 0.6];
%     sumThreshold = 100;
%     TR = 2000 ms; % (whole volume)
%     TE = 30 ms;
% define scan protocal parameters (these are the default values)
scan.voxel.size    = [3 3 3]; % [3.75 3.75 3.75] in original
 experiment
scan.voxel.matrix  = [64 64 NaN];  % [64 50 24]; in original
 experiment
scan.voxel.spacing = [0 0 0.2*scan.voxel.size(3)]; % assume 20% Z
 spacing
scan.tiltAngle     = 0; % degrees
scan.TR            = 2400; % [ms]  %?TR = 24 ms; scan time = 6s (per
 slice?)
scan.TE            = 35;   % [ms]
scan.ES            = 0.51; % [ms] echo spacing
scan.FA            = 11;   % [degrees]
scan.BW            = 2232; % [Hz/Px]
scan.order         = 'SD'; % SD = sequential descending order
scan.KM0           = 2225; % fit to data with max of 909 at 3T and FA
 = 90 degree  (Siemens 3T ~4000)
scan.noise_method  = 'percent';
scan.noise         = 0;    % percent noise relative to peak
scan.attenuation   = 0;    % coil attenuation factor ~mm^-1
% FWHM ~4.5 mm

simulations{Now_sss(1)}.scan = scan;
save('simulations.mat','simulations')

% load tissue fuzzy memberships in subject's native space
[V_fuzzy,~] = STANCE_choose_subject(subject_brain,'fuzzy',true);
fn_tissue = [V_fuzzy(1).fname,'.gz'];

% generate the tissue fuzzy memberships in functional space
[V_reslice,Y_reslice] = STANCE_reslice_tissue(fn_tissue,scan,[],
[],false); % change the last arg to 'true' to show figures
sliceLimits =
 [V_reslice(1).sliceLimitLower,V_reslice(1).sliceLimitUpper];
[~,I_max] = max(sum(sum(Y_reslice(:,:,:,3))));
showSlice2 = I_max(1);
% figure, imshow(imrotate(Y_reslice(:,:,showSlice2,3),90),[]);
% TITLE = ['Reslice tissue priors - gray matter, A
 slice:',num2str(showSlice2TA)];
% title(TITLE)
```

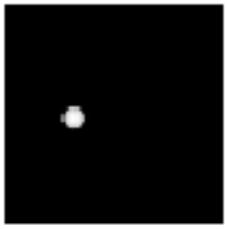
```
fn_fuzzy_reslice = V_reslice(1).fname;

% generate T2* baseline volume in functional space
[V_T2star_Map,Y_T2star_Map] =
 STANCE_make_parameter_map(fn_fuzzy_reslice,'T2star');
scrsz = get(groot,'ScreenSize');
positionVector2 = [scrsz(3)/2.5 scrsz(4)/2.5 scrsz(3)/5 scrsz(4)/3];
f2 = figure;
imshow(imrotate(Y_T2star_Map(:,:,showSlice2),90),[])
title('T2* baseline volume')
set(f2,'OuterPosition',positionVector2);
movegui(f2,'north')

% project activation map on to functional space
[V_finger_opposition_reslice,Y_finger_opposition_reslice] =
 STANCE_reslice_volume(V_finger_opposition_reg,scan,sliceLimits);
[~,I_max] = max(sum(sum(Y_finger_opposition_reslice)));
showSlice2TA = I_max(1);
TITLE = {'Finger opposition task,'; ['functional axial:
 ',num2str(showSlice2TA)]};
f3 = figure;
imshow(imrotate(Y_finger_opposition_reslice(:,:,showSlice2TA),90),[])
title(TITLE)
set(f3,'OuterPosition',positionVector2);
movegui(f3,'northeast')
delete(V_finger_opposition_reg.fname);

% mask with gray matter mask
[Y_finger_opposition_reslice,Y_GM] =
 STANCE_GM_mask(Y_finger_opposition_reslice,task.GMvolume,Now_sss);
%figure,
 imshow(imrotate(Y_finger_opposition_reslice(:,:,showSlice2TA),90),
[]);

subjectActivation3D{1} = Y_finger_opposition_reslice;

[~,I_max] = max(sum(sum(Y_finger_opposition_reslice)));
showSlice2TA = I_max(1);

cd(STANCE_genpath([],2))
save('STANCEsubject.mat','Now_sss','subject_brain','fn_tissue')
cd(STANCEroot)

% add activation to T2* baseline
[V_T2star_Map_Act,Y_T2star_Map_Act] =
 STANCE_add_activation(V_T2star_Map.fname,Y_finger_opposition_reslice,scan.TE,task
TITLE = {'T2* map w/ BOLD activation,', ['Axial slice:
 ',num2str(showSlice2TA)]};
f4 = figure;
imshow(imrotate(Y_T2star_Map_Act(:,:,showSlice2TA),90),[])
title(TITLE)
set(f4,'OuterPosition',positionVector2);
movegui(f4,'east')
```

```
% figure, imshow(imrotate(Y_T2star_Map_Act(:,:,showSlice2TA),90)-
imrotate(Y_T2star_Map(:,:,showSlice2TA),90),[]);
% TITLE = ['T2* map activation - baseline, A
 slice:',num2str(showSlice2TA)];
% title(TITLE)

% exact EPI baseline signal, no noise, no attenuation
[V_EPI0,Y_EPI0] =
 STANCE_EPI_signal(fn_fuzzy_reslice,Y_T2star_Map,scan);
% figure, imshow(imrotate(Y_EPI0(:,:,showSlice2TA),90),[]);
TITLE = {'Gray matter priors,', ['axial
 slice:',num2str(showSlice2TA)]};
f5 = figure;
imshow(imrotate(Y_GM(:,:,showSlice2TA),90),[])
title(TITLE)
set(f5,'OuterPosition',positionVector2);
movegui(f5,'southeast')

maxS = max(Y_EPI0(:).*Y_GM(:));
TITLE = {'Baseline signal volume,',['Axial slice:
 ',num2str(showSlice2TA)]};
f6 = figure;
imshow(imrotate(Y_EPI0(:,:,showSlice2TA),90),[0,maxS])
title(TITLE)
set(f6,'OuterPosition',positionVector2);
movegui(f6,'south')

% exact EPI signal, no noise, no attenuation, with rigid body motion
motion = [0.3333, 0.6667, 1, 1, 2, -2]; % [x-, y-,z-translation,
 rotation about x, y, z axis]
[V_EPI,Y_EPI] =
 STANCE_EPI_signal(fn_fuzzy_reslice,Y_T2star_Map_Act,scan,[],
[],motion);
maxS = max(Y_EPI(:).*Y_GM(:));
TITLE = {'Exact BOLD signal with motion,',['Axial slice:
 ',num2str(showSlice2TA)]};
f7 = figure;
imshow(imrotate(Y_EPI(:,:,showSlice2TA),90),[0,maxS])
title(TITLE)
set(f7,'OuterPosition',positionVector2);
movegui(f7,'southwest')

f8 = figure;
imshow(imrotate(Y_EPI(:,:,showSlice2TA),90)-
imrotate(Y_EPI0(:,:,showSlice2TA),90),[])
title('(BOLD - baseline) signal')
set(f8,'OuterPosition',positionVector2);
movegui(f8,'west')

% Finger opposition task activation of the PSM in subject
TITLE = {'Finger opposition task', 'BOLD signal in subject'};
htask1subfun =
 STANCE_display_activation_slice(Y_EPI,Y_finger_opposition_reslice,[],
[]);
```

```
title(TITLE)
movegui(htask1subfun,'center');

cd(STANCE_genpath)
save('STANCEscan.mat','scan','fn_fuzzy_reslice','sliceLimits','subjectActivation3D
cd(STANCEroot)

clear task;

else
    save('simulations.mat','simulations')
end
```

```
Reslicing gray matter fuzzy membership labels...
Reslicing background fuzzy membership labels...
Reslicing CSF fuzzy membership labels...
Reslicing white matter fuzzy membership labels...
Reslicing fat fuzzy membership labels...
Reslicing muscle fuzzy membership labels...
Reslicing skin fuzzy membership labels...
Reslicing skull fuzzy membership labels...
Reslicing blood vessels fuzzy membership labels...
Reslicing connective tissue fuzzy membership labels...
Reslicing dura matter fuzzy membership labels...
Reslicing bone marrow fuzzy membership labels...
o Writing C:\spm\STANCE/fMRI/study001/subject0001/
session001\EPI_BOLD_0001_001.nii
The maximum intensity of the simulated signal: 253.9648
o Writing C:\spm\STANCE/fMRI/study001/subject0001/
session001\EPI_BOLD_0001_001.nii
The maximum intensity of the simulated signal: 246.2837
```

Finger opposition task, functional axial: 34

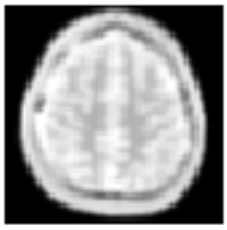


T2* map w/ BOLD activation, Axial slice: 34

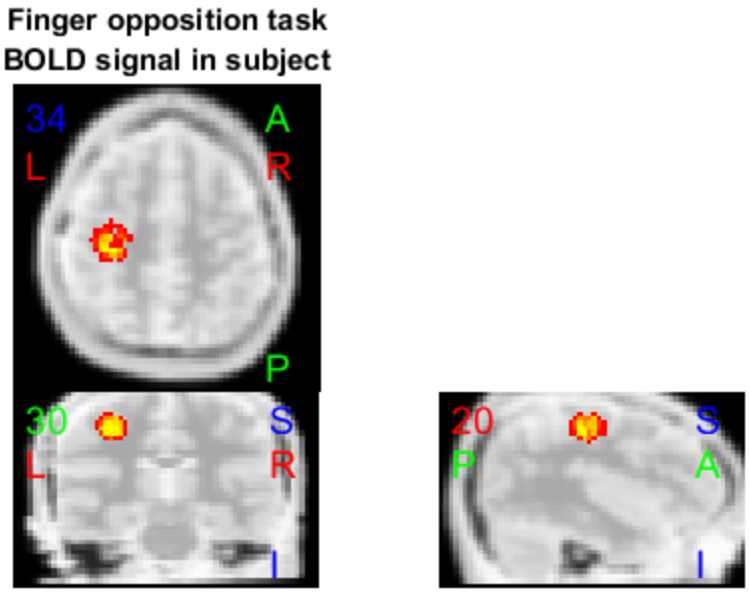
Gray matter priors, axial slice:34

Baseline signal volume,
Axial slice: 34

Exact BOLD signal with motion,
Axial slice: 34



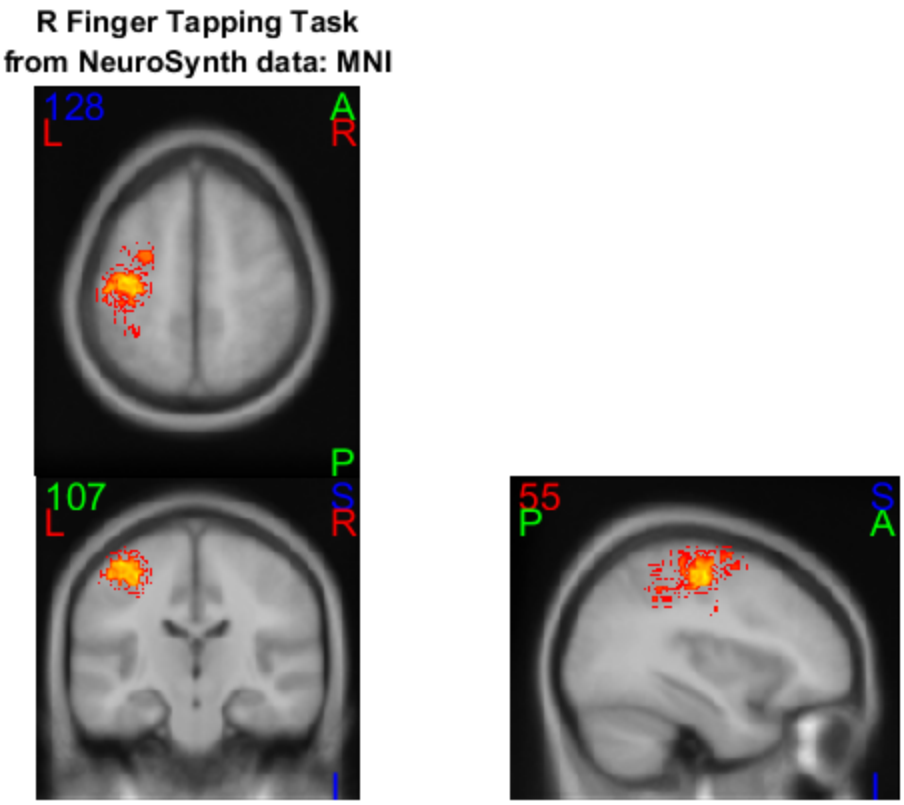
(BOLD - baseline) signal



Load activation map from data files

```
clear task;

uiwait(msgbox('Demo example of a finger related task data from
 NeuroSynth.','Finger meta-analysis data','modal'));

% NeuroSynth finger tapping example
task.name = 'Finger Tapping';
task.activation(1).region   = [STANCEroot,'/activations/finger
 tapping_pFgA_z_FDR_0.01.nii.gz'];
task.activation(1).shape    = 'data'; % data derived by forward
 inference
task.activation(1).volume   = [];        % the index for 4D data
task.activation(1).proportion = 5.0;     % the activation thresholds:
 [lower, saturation]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
task.activation(2).region   = [STANCEroot,'/activations/finger
 tapping_pAgF_z_FDR_0.01.nii.gz'];
task.activation(2).shape    = 'data'; % data derived by reverse
 inference
task.activation(2).proportion = 5.0;     % the activation thresholds:
 [lower, saturation]
task.combine{1} = {'OR','all'};
task.combine{2} = {'AND','flip'};
```

```
task.combine{3} = {'mask','L'};
% define signal amplitude
task.amplitude = 0.03; % 3% activation

if exist('simulations.mat','file')
    load('simulations.mat');
else
    % create simulations struct
end
simulations{Now_sss(1)}.task{2} = task;

task.activation(1).map =
 STANCE_load_map(task.activation(1).region,V_MNI,5.0);
task.activation(2).map =
 STANCE_load_map(task.activation(2).region,V_MNI,5.0);

disp('o Combining finger tapping task activation maps from data
 files...')
task.map = STANCE_parse_combine(task);
% % explictly this does the following
% task.map = STANCE_combine_maps('OR',task.activation(:).map);
% % combine activation in opposite hemispheres
% task.map = STANCE_combine_maps('AND',task.map,flipud(task.map));

% remove bright artefact on medial surface
task.map(ceil(0.38*dimensions(1)):end,:,:) = 0;
task.map(:,:,1:90) = 0;

% free up working memory (optional)
% task.activation(1).map = [];
% task.activation(2).map = [];

TITLE = {'R Finger Tapping Task','from NeuroSynth data: MNI'};
htask2 = STANCE_display_activation_slice(Y_MNI,task.map,[],[]);
title(TITLE)
movegui(htask2,'center');

% h_task_TS_R =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTS,1);
% title('R Finger Tapping Task from NeuroSynth: S')
%
% h_task_TC =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTC,2);
% title('R Finger Tapping Task from NeuroSynth: C')

o Loading map from data in C:\spm\STANCE/activations/finger
 tapping_pFgA_z_FDR_0.01.nii.gz.
o Loading map from data in C:\spm\STANCE/activations/finger
 tapping_pAgF_z_FDR_0.01.nii.gz.
o Combining finger tapping task activation maps from data files...
```



Save, free up memory, and return

```
save('simulations.mat','simulations')

% make room in memory
clear('V_MNI','Y_MNI')
task.map = [];

% save activation template
task.activation(1).map = int8(255*task.activation(1).map);
task.activation(2).map = int8(255*task.activation(2).map);
cd([STANCEroot,'/activations'])
save([task.name,'.mat'],'task')
cd(currentDir)
% free up working memory (optional)
task.activation(1).map = [];
task.activation(2).map = [];
```

Published with MATLAB® R2016a