Table of Contents


```
% author: Dr. Jason E. Hill (post-doc fellow with CNT at TTU)
% demo_3D_ex3    updated     20 SEPT 2016
```

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

This demo models the 8 main gustation (taste) activated regions.

```
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

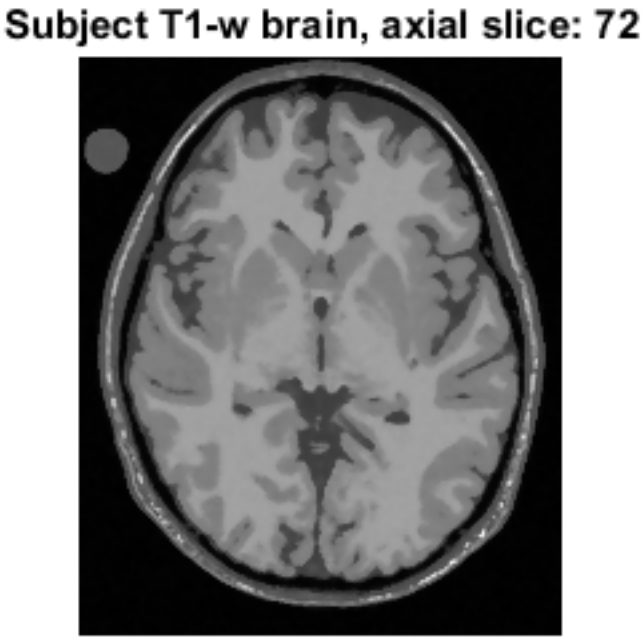
```
warning('off','MATLAB:opengl:StartupBlacklistedNoSetting');
warning('off', 'MATLAB:hg:AutoSoftwareOpenGL');
% ... finite warning
warning('off', 'MATLAB:FINITE:obsoleteFunction');
% ... NIFTI class warnings when loading SPM mat files
warning('off', 'MATLAB:unknownElementsNowStruc');
warning('off',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

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

```
subject_brain = 2;
Now_sss       = [3 1 1];
STANCE_new_session(3,1,1)
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)
% register the MNI152 brain to the 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/study003/subject0001/session001.

ans =

     1     1   309

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 =

    0.9998    0.0076   -0.0169    1.5680
   -0.0073    0.9998    0.0167    0.8990
    0.0171   -0.0166    0.9997   -1.2812
         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

```
% see Table 2 of
% "Functional MRI Detection of Activation in the Primary Gustatory
 Cortices in Humans"
% in Chemical Senses 30(7):583-92, September 2005.
% voxel size = 125/32 x 125/32 x 4 mm
voxelVolume = 1; % mm (NOTE: from group analysis in MNI space)

% define the gustation (taste) activated regions
task.name = 'Gustation activated regions';
task.activation(1).region     = 'L buried Fop';
task.activation(1).volume     = 92*voxelVolume; % from Table 2
task.activation(1).center     = [-51,26,4];     % L buried Fop, from
 Table 2
task.activation(1).rotation   = [0,0,0];        % [degrees]
task.activation(1).shape      = 'ellipsoid';
task.activation(1).proportion = [2,1,1];        % aspect ratio
task.activation(1).falloff    = 0.005;          % parameterizes
 exponential falloff about center, in [0,1]
task.activation(1).minimum    = 0.2;            % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
task.activation(2).region     = 'L Rop';
task.activation(2).volume     = 21*voxelVolume; % from Table 2
task.activation(2).center     = [-60,-18,24];   % L Rop
```

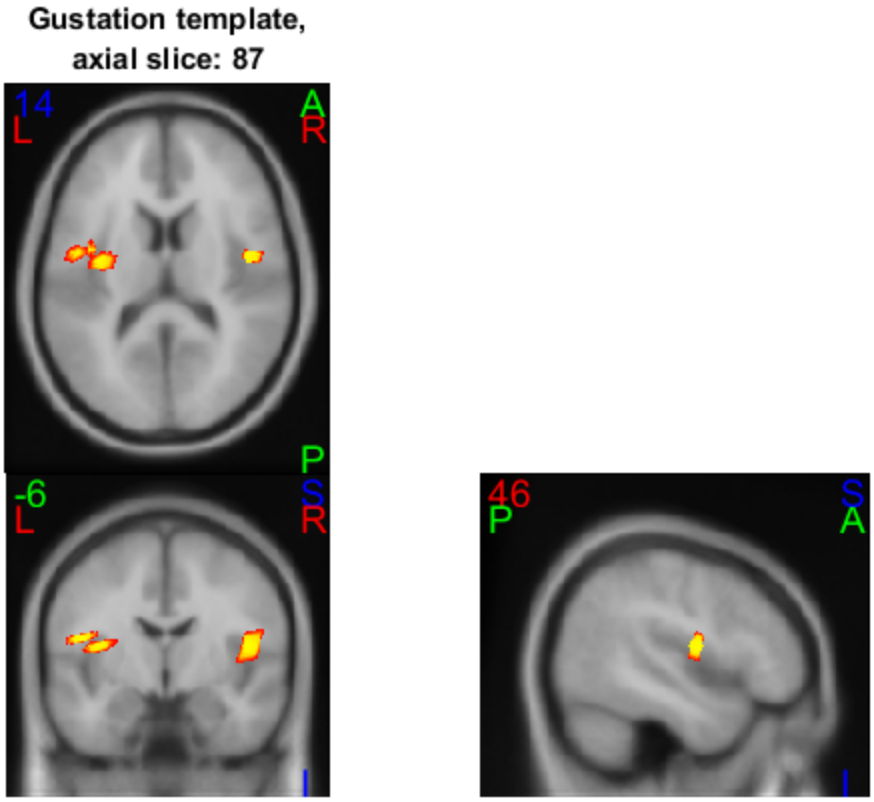
```
task.activation(2).rotation   = [0,0,0];        % [degrees]
task.activation(2).shape      = 'sphere';
task.activation(2).proportion = [1,1,1];        % aspect ratio
task.activation(2).falloff    = 0.005;          % parameterizes
 exponential falloff about center, in [0,1]
task.activation(2).minimum    = 0.2;            % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
task.activation(3).region     = 'L cs';
task.activation(3).volume     = 125*voxelVolume; % from Table 2
task.activation(3).center     = [-44,-2,12];     % L cs
task.activation(3).rotation   = [0,-30,0];       % [degrees]
task.activation(3).shape      = 'astroid';
task.activation(3).proportion = [3,3,2];         % aspect ratio
task.activation(3).falloff    = 0.005;           % parameterizes
 exponential falloff about center, in [0,1]
task.activation(3).minimum    = 0.2;             % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
task.activation(4).region     = 'R buried Pop';
task.activation(4).volume     = 119*voxelVolume; % from Table 2
task.activation(4).center     = [46,-6,12];      % R buried (area G)
 Pop
task.activation(4).rotation   = [-15,-15,-15];   % [degrees]
task.activation(4).shape      = 'cuboid';
task.activation(4).proportion = [2,1,3];         % aspect ratio
task.activation(4).falloff    = 0.005;           % parameterizes
 exponential falloff about center, in [0,1]
task.activation(4).minimum    = 0.2;             % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
task.activation(5).region     = 'L Pop';
task.activation(5).volume     = 106*voxelVolume; % from Table 2
task.activation(5).center     = [-38,-8,12];     % L Pop
task.activation(5).rotation   = [-15,-15,-15];   % [degrees]
task.activation(5).shape      = 'ellipsoid';
task.activation(5).proportion = [3,2,1];         % aspect ratio
task.activation(5).falloff    = 0.005;           % parameterizes
 exponential falloff about center, in [0,1]
task.activation(5).minimum    = 0.2;             % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
task.activation(6).region     = 'LL Pop';
task.activation(6).volume     = 106*voxelVolume; % from Table 2
task.activation(6).center     = [-48,-6,16];     % LL of Pop
task.activation(6).rotation   = [-15,-15,-15];   % [degrees]
task.activation(6).shape      = 'ellipsoid';
task.activation(6).proportion = [3,2,1];         % aspect ratio
task.activation(6).falloff    = 0.005;           % parameterizes
 exponential falloff about center, in [0,1]
task.activation(6).minimum    = 0.2;             % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
task.activation(7).region     = 'R SPI (area G)';
```

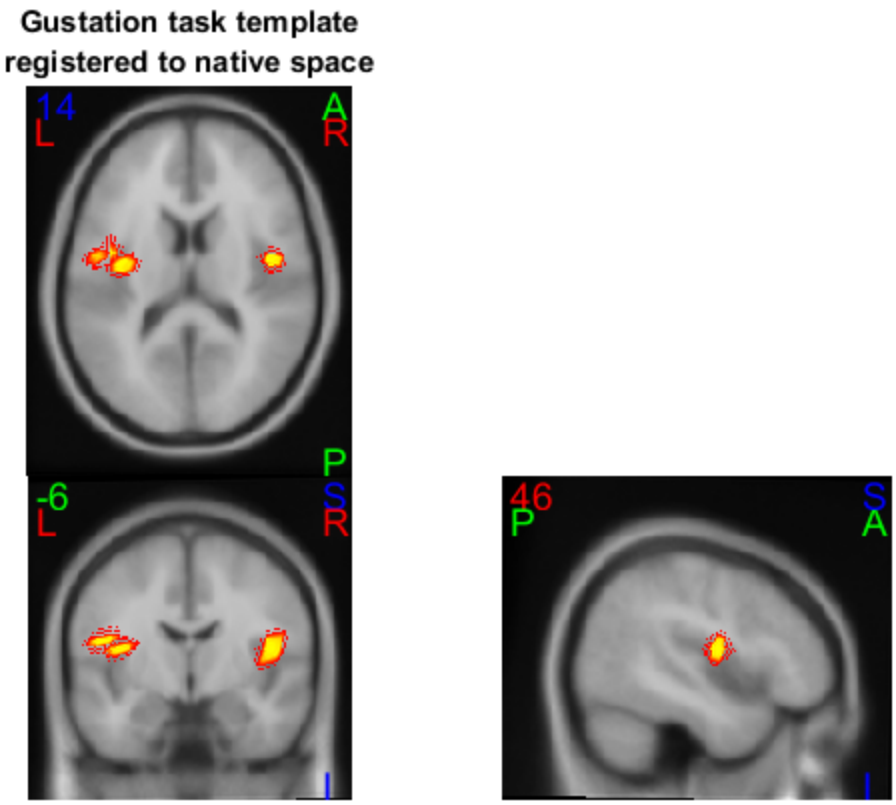
```
task.activation(7).volume     = 119*voxelVolume; % from Table 2
task.activation(7).center     = [46,-6,12];      % R Superior
 posterior (area G) Insula (SPI)
task.activation(7).rotation   = [0,0,0];         % [degrees]
task.activation(7).shape      = 'sphere';
task.activation(7).proportion = [1,1,1];         % aspect ratio
task.activation(7).falloff    = 0.005;           % parameterizes
 exponential falloff about center, in [0,1]
task.activation(7).minimum    = 0.2;             % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
task.activation(8).region     = 'L Insula';
task.activation(8).volume     = 106*voxelVolume; % from Table 2
task.activation(8).center     = [-38,-8,12];     % L Insula
task.activation(8).rotation   = [0,0,0];         % [degrees]
task.activation(8).shape      = 'sphere';
task.activation(8).proportion = [1,1,1];         % aspect ratio
task.activation(8).falloff    = 0.005;           % parameterizes
 exponential falloff about center, in [0,1]
task.activation(8).minimum    = 0.2;             % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% 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)}.name = 'Gustatory task study';
simulations{Now_sss(1)}.task{1} = task;

% define components
disp('Defining the gustation (taste) activated regions...')
disp('... activated region 1.')
task.activation(1).map = STANCE_make_activation_map(dimensions,
 origin, task.activation(1));
disp('... activated region 2.')
task.activation(2).map = STANCE_make_activation_map(dimensions,
 origin, task.activation(2));
disp('... activated region 3.')
task.activation(3).map = STANCE_make_activation_map(dimensions,
 origin, task.activation(3));
disp('... activated region 4.')
task.activation(4).map = STANCE_make_activation_map(dimensions,
 origin, task.activation(4));
disp('... activated region 5.')
task.activation(5).map = STANCE_make_activation_map(dimensions,
 origin, task.activation(5));
disp('... activated region 6.')
task.activation(6).map = STANCE_make_activation_map(dimensions,
 origin, task.activation(6));
disp('... activated region 7.')
```

```
task.activation(7).map = STANCE_make_activation_map(dimensions,
 origin, task.activation(7));
disp('... activated region 8.')
task.activation(8).map = STANCE_make_activation_map(dimensions,
 origin, task.activation(7));

% combine different task components
disp('o Combining task components.')
Nactivations = length(task.activation);
task.map = STANCE_combine_maps('OR',task.activation(:).map);

% clear out working memory (optional)
task_map_temp = [];

% find MNI gray matter volume of activation map
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 = {'Gustation template,',['axial slice:
 ',num2str(showSliceTA)]};
% title(TITLE)
h_task = STANCE_display_activation_slice(Y_MNI,task.map,[],[],origin);
title(TITLE)
movegui(h_task,'center');

[~,I_max] = max(sum(sum(task.map,3)));
showSliceTC = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTC,2,origin);
% TITLE = ['Gustation (taste) activated regions, C slice:
 ',num2str(showSliceTC)];
% title(TITLE)

[~,I_max] = max(sum(sum(task.map,3),2));
showSliceTS = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,task.map,showSliceTS,1,origin);
% TITLE = ['Gustation (taste) activated regions, S slice:
 ',num2str(showSliceTS)];
% title(TITLE)
% clear('V_MNI','Y_MNI')

[V_gustation_activated_reg,Y_gustation_activated_reg] =
 STANCE_register_activation(V_T1w.fname,task);

% figure,
 imshow(imrotate(Y_gustation_activated_reg(:,:,showSliceTA),90),[]),
 drawnow;
% title('Gustation (taste) activated regions registered')
```

```
h_task_reg =
 STANCE_display_activation_slice(Y_MNI_reg,Y_gustation_activated_reg,
[],[],origin);
title({'Gustation task template', 'registered to native space'})
movegui(h_task_reg ,'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(1).map = int8(255*task.activation(1).map);
task.activation(2).map = int8(255*task.activation(2).map);
task.activation(3).map = int8(255*task.activation(3).map);
task.activation(4).map = int8(255*task.activation(4).map);
task.activation(5).map = int8(255*task.activation(5).map);
task.activation(6).map = int8(255*task.activation(6).map);
task.activation(7).map = int8(255*task.activation(7).map);
task.activation(8).map = int8(255*task.activation(8).map);
cd([STANCEroot,'/activations'])
save([task.name,'.mat'],'task')
cd(STANCEroot)
task.activation(1).map = [];
task.activation(2).map = [];
task.activation(3).map = [];
task.activation(4).map = [];
task.activation(5).map = [];
task.activation(6).map = [];
task.activation(7).map = [];
task.activation(8).map = [];

Defining the gustation (taste) activated regions...
... activated region 1.
o Specifying spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... activated region 2.
o Specifying spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... activated region 3.
o Specifying superspherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... activated region 4.
o Specifying cubical template
o Performing affine transformation.
o Building activation map.
... activated region 5.
```

o Specifying spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... activated region 6.
o Specifying spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... activated region 7.
o Specifying spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... activated region 8.
o Specifying spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
o Combining task components.





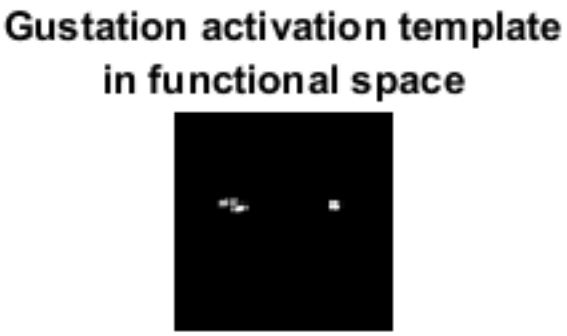
Reslice according to fMRI scan protocol specifications

```
if makeFMRI == true

% use default scan settings:
%     voxelSize    = [3 3 3];
%     new_dims     = [64 64 NaN];  % effectively [64 64 40];
%     tiltAngle    = 15; % degrees
%     voxelSpacing = [0 0 0.6];
%     sumThreshold = 100;
%     TR = 2000 ms; % (whole volume)
%     TE = 30 ms;
scan.voxel.size    = [3 3 3];      %[3.90625 3.90625 4] = 25x25cmx4mm
scan.voxel.matrix  = [64 64 NaN];  %[64 64 20]
scan.voxel.spacing = [0 0 0.2*scan.voxel.size(3)]; % assume 20% Z
 spacing
scan.tiltAngle     = 11; % degrees
scan.TR            = 3000; % [ms]
scan.TE            = 60;   % [ms]
scan.ES            = 0.51; % [ms] echo spacing
scan.FA            = 78;   % 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
scan.noise_method  = 'percent';
scan.noise         = 0;    % percent noise relative to peak
scan.attenuation   = 0;  % coil attenuation factor ~mm^-1

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

%showSlice2 = 19;

% 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,Now_sss); % change last to 'true' to show figure
sliceLimits =
 [V_reslice(1).sliceLimitLower,V_reslice(1).sliceLimitUpper];
[~,I_max] = max(sum(sum(Y_reslice(:,:,:,3))));
showSlice2TA = I_max(1);
% figure, imshow(imrotate(Y_reslice(:,:,showSlice2TA,3),90),[]);
% TITLE = ['Resliced 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');
% figure, imshow(imshow(imrotate(Y_T2star_Map(:,:,showSlice2TA),90),
[]);
% TITLE = ['T2* baseline map, A slice:',num2str(showSlice2TA)];
% title(TITLE)

% project activation map onto functional space
[V_gustation_activated_reslice,Y_gustation_activated_reslice] =
 STANCE_reslice_volume(V_gustation_activated_reg,scan,sliceLimits);

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

scrsz = get(groot,'ScreenSize');
positionVector2 = [scrsz(3)/2.5 scrsz(4)/2.5 scrsz(3)/5 scrsz(4)/3];
f2 = figure;
imshow(imrotate(Y_gustation_activated_reslice(:,:,showSlice2TA),90),
[]);
title({'Gustation activation template', 'in functional space'})
set(f2,'OuterPosition',positionVector2);
movegui(f2,'north')

delete(V_gustation_activated_reg.fname);
```

```
% mask with gray matter mask
[Y_gustation_activated_reslice,Y_GM] =
 STANCE_GM_mask(Y_gustation_activated_reslice,task.GMvolume,Now_sss);
%figure,
 imshow(imrotate(Y_gustation_activated_reslice(:,:,showSlice2TA),90),
[]);
%'Gustation activated regions (functional) masked w/ GM'

subjectActivation3D(1,:,:,:) = Y_gustation_activated_reslice;

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

```
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...
```



Add activation to T2* baseline volume

add activation to T2* baseline

```
[V_T2star_Map_Act,Y_T2star_Map_Act] =
 STANCE_add_activation(V_T2star_Map.fname,Y_gustation_activated_reslice,scan.TE,ta
% figure, imshow(Y_T2star_Map_Act(:,:,showSlice2TA),[]);
% TITLE = ['T2* map w/ BOLD activation, A
 slice:',num2str(showSlice2TA)];
% title(TITLE)
% figure, imshow(Y_T2star_Map_Act(:,:,showSlice2)-
Y_T2star_Map(:,:,showSlice2TA),[]);
```

```
% TITLE = ['T2* map activation - baseline, A
 slice:',num2str(showSlice2TA)];
% title(TITLE)

fn_reslice = V_reslice(1).fname;
% exact EPI signal, no noise, no attenuation, no activation
[V_EPI0,Y_EPI0] = STANCE_EPI_signal(fn_reslice,Y_T2star_Map,scan);
maxS = max(Y_EPI0(:).*Y_GM(:));

f3 = figure;
imshow(imrotate(Y_EPI0(:,:,showSlice2TA),90),[0,maxS]);
TITLE = {'Baseline signal volume,', ['axial slice:
 ',num2str(showSlice2TA)]};
title(TITLE)
set(f3,'OuterPosition',positionVector2);
movegui(f3,'northeast')
```

```
o Writing C:\spm\STANCE/fMRI/study003/subject0001/
session001\EPI_BOLD_0001_001.nii
The maximum intensity of the simulated signal: 721.4784
```

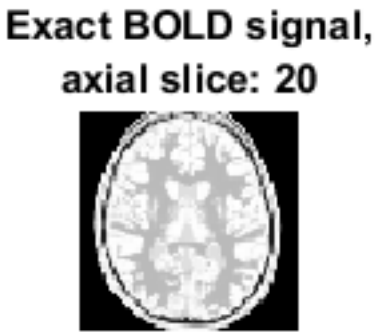
Generate pristine EPI signal

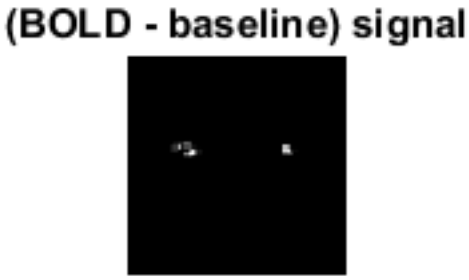
exact EPI signal, no noise, no attenuation

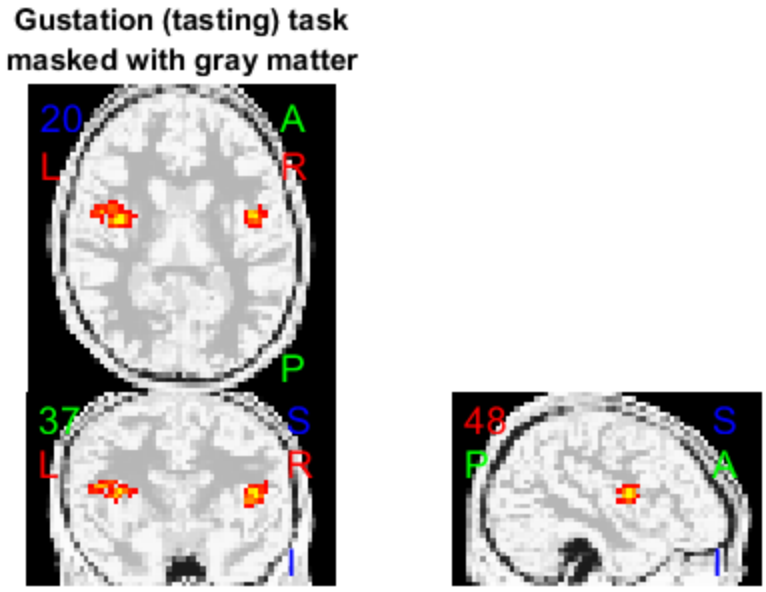
```
[V_EPI,Y_EPI] = STANCE_EPI_signal(fn_reslice,Y_T2star_Map_Act,scan);
maxS = max(Y_EPI(:).*Y_GM(:));
f4 = figure;
imshow(imrotate(Y_EPI(:,:,showSlice2TA),90),[0,maxS]);
TITLE = {'Exact BOLD signal,',['axial slice:
 ',num2str(showSlice2TA)]};
title(TITLE)
set(f4,'OuterPosition',positionVector2);
movegui(f4,'east')

f5 = figure;
imshow(imrotate(Y_EPI(:,:,showSlice2TA),90)-
imrotate(Y_EPI0(:,:,showSlice2TA),90),[]);
```

```
title('(BOLD - baseline) signal')
set(f5,'OuterPosition',positionVector2);
movegui(f5,'southeast')

htasksubfun =
 STANCE_display_activation_slice(Y_EPI,Y_gustation_activated_reslice,
[],[]);
title({'Gustation (tasting) task', 'masked with gray matter'})
movegui(htasksubfun,'center');
```

o Writing C:\spm\STANCE/fMRI/study003/subject0001/
session001\EPI_BOLD_0001_001.nii
The maximum intensity of the simulated signal: 729.4124







Save results, free up memory, and return

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

clear task;

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

Published with MATLAB® R2016a