Table of Contents


```
% authors: Dr. Jason E. Hill (post-doc fellow with CNT at TTU), Neloy
 R. Shome & Prethom Shome
% demo_3D_RSNs_models    updated     12 SEP 2016
```

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

Models the complicated shapes of the 3D baseline signals for some important and well studied Resting State Networks (RSNs). This proof-of-principle demo uses reported coordinates and volumes to model the volumes.

```
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 ...

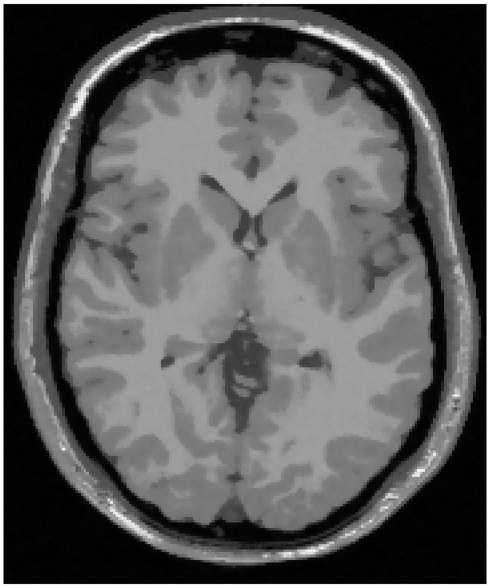
... OpenGl warnings

```
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 = 3;

% 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);
```

```
% 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');

T1w_dim = V_T1w.dim;  % dimensions of T1-w volume
T1w_mat = V_T1w.mat;  % 4x4 homographic matrix relating indeces to
 real-world coordinates
figure, imshow(imrotate(Y_T1w(:,:,showSlice),90),[]);
TITLE = ['Subject T1-w brain, A slice: ',num2str(showSlice)];
title(TITLE), drawnow;

% retreive transfromation matrix mapping MNI152 to subject's native
 space
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)

dimensions = size(Y_T1w);
origin = round(abs(V_T1w.mat(1:3,4)))';
```



Load activation map from data files of Basal ganglia network

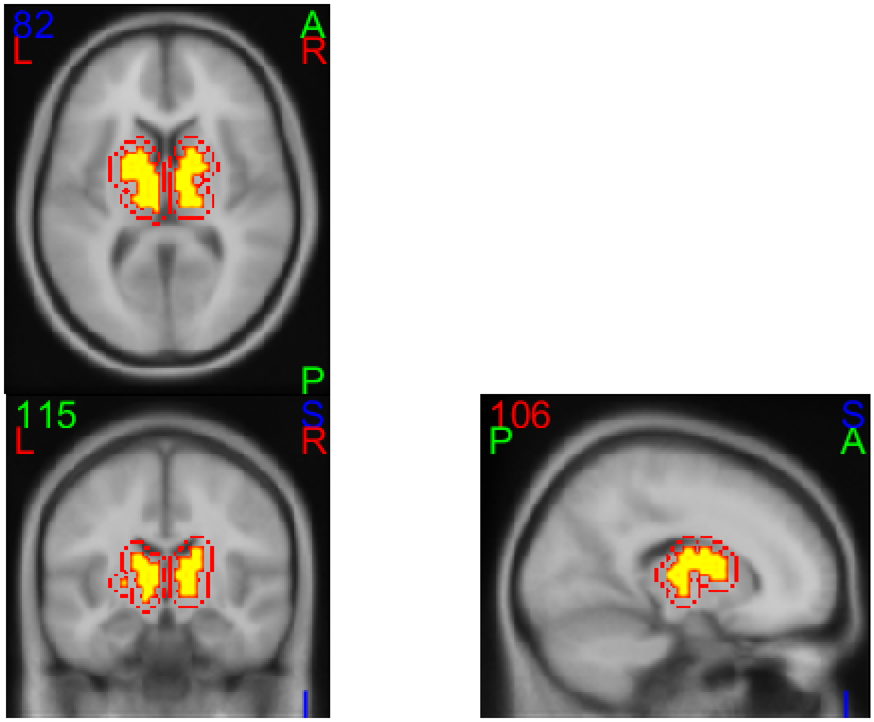
```
uiwait(msgbox('Demo example of the Basal Ganglia Network.','Resting
 state: BGN','modal'));
```

```
clear task;
% Basal Ganglia Network ICs example
rest_BGN_RSN.name = 'Basal Ganglia Network: RSN ICs 11-15';
rest_BGN_RSN.activation(1).region   = [STANCEroot,'/activations/
RSN.nii.gz'];
rest_BGN_RSN.activation(1).shape    = 'data';      % data derived by
 independent component analysis
rest_BGN_RSN.activation(1).volume   = 11;          % the index for 4D
 data
rest_BGN_RSN.activation(1).map =
 STANCE_load_map(rest_BGN_RSN.activation(1).region,V_MNI,11);
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_BGN_RSN.activation(2).region   = [STANCEroot,'/activations/
RSN.nii.gz'];
rest_BGN_RSN.activation(2).shape    = 'data';       % data derived by
 independent component analysis
rest_BGN_RSN.activation(2).volume   = 12;          % the index for 4D
 data
rest_BGN_RSN.activation(2).map =
 STANCE_load_map(rest_BGN_RSN.activation(2).region,V_MNI,12);
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_BGN_RSN.activation(3).region   = [STANCEroot,'/activations/
RSN.nii.gz'];
rest_BGN_RSN.activation(3).shape    = 'data';       % data derived by
 independent component analysis
rest_BGN_RSN.activation(3).volume   = 13;          % the index for 4D
 data
rest_BGN_RSN.activation(3).map =
 STANCE_load_map(rest_BGN_RSN.activation(2).region,V_MNI,13);
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_BGN_RSN.activation(4).region   = [STANCEroot,'/activations/
RSN.nii.gz'];
rest_BGN_RSN.activation(4).shape    = 'data';       % data derived by
 independent component analysis
rest_BGN_RSN.activation(4).volume   = 14;          % the index for 4D
 data
rest_BGN_RSN.activation(4).map =
 STANCE_load_map(rest_BGN_RSN.activation(2).region,V_MNI,14);
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_BGN_RSN.activation(5).region   = [STANCEroot,'/activations/
RSN.nii.gz'];
rest_BGN_RSN.activation(5).shape    = 'data';       % data derived by
 independent component analysis
rest_BGN_RSN.activation(5).volume   = 15;          % the index for 4D
 data
rest_BGN_RSN.activation(5).map =
 STANCE_load_map(rest_BGN_RSN.activation(2).region,V_MNI,15);

% combine different component with fuzzy logic OR
disp('o Combining activation components.')
rest_BGN_RSN.map =
 STANCE_combine_maps('OR',rest_BGN_RSN.activation(:).map);
% patch center line shift
```

```
rest_BGN_RSN.map = circshift(rest_BGN_RSN.map,-7);

% free up working memory (optional)
rest_BGN_RSN.activation(1).map = [];
rest_BGN_RSN.activation(2).map = [];
rest_BGN_RSN.activation(3).map = [];
rest_BGN_RSN.activation(4).map = [];
rest_BGN_RSN.activation(5).map = [];

% [~,I_max] = max(sum(sum(rest_BGN_RSN.map)));
% showSliceA = I_max(1);
% h_rest_BGN_RSN =
 STANCE_display_activation_slice(Y_MNI,rest_BGN_RSN.map,showSliceA,3);
% title('Basal Ganglia Network: RSN ICs 11-15: A')
%
% [~,I_max] = max(sum(sum(rest_BGN_RSN.map,2),3));
% showSliceS = I_max(1);
% h_rest_BGN_RSN_TS_R =
 STANCE_display_activation_slice(Y_MNI,rest_BGN_RSN.map,showSliceS,1);
% title('Basal Ganglia Network: RSN ICs 11-15: S')
%
% [~,I_max] = max(sum(sum(rest_BGN_RSN.map),3));
% showSliceC = I_max(1);
% h_rest_BGN_RSN_TC =
 STANCE_display_activation_slice(Y_MNI,rest_BGN_RSN.map,showSliceC,2);
% title('Basal Ganglia Network: RSN ICs 11-15: C')

h_rest_BGN_RSN =
 STANCE_display_activation_slice(Y_MNI,rest_BGN_RSN.map,[],[]);
title('Basal Ganglia Network: RSN ICs 11-15');
movegui(h_rest_BGN_RSN,'east');
```

```
o Loading map from component data in C:\spm\STANCE/activations/
RSN.nii.gz.
o Loading map from component data in C:\spm\STANCE/activations/
RSN.nii.gz.
o Loading map from component data in C:\spm\STANCE/activations/
RSN.nii.gz.
o Loading map from component data in C:\spm\STANCE/activations/
RSN.nii.gz.
o Loading map from component data in C:\spm\STANCE/activations/
RSN.nii.gz.
o Combining activation components.
```

Basal Ganglia Network: RSN ICs 11-15



Build activation regions of Basal ganglia network

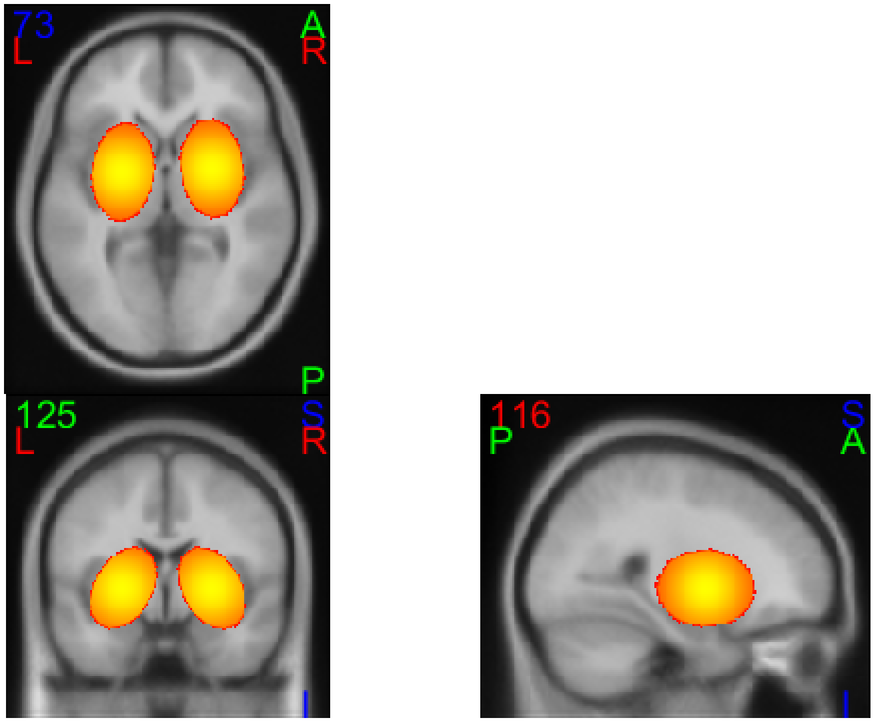
```
% see Table 2 of "A baseline for the multivariate comparison
% of resting-state networks" in Frontiers in Systems Neuroscience
 February
% 2011, Volume 5, Article 2.
% GICA voxel volume = 3 x 3 x 3 mm
voxelVolume = 3*3*3;

% define the Basal Ganglia Network (IC 21) activated regions
rest_BGN_IC21.name = 'Basal Ganglia Network: IC 21';
rest_BGN_IC21.activation(1).region = 'R putamen';
rest_BGN_IC21.activation(1).volume = 1454*voxelVolume; % from Table 2
rest_BGN_IC21.activation(1).center = [25,-1,0];        % L superior
 temporal gyrus
rest_BGN_IC21.activation(1).rotation = [+10,+10,+10];  % [degrees]
rest_BGN_IC21.activation(1).shape = 'sphere';
rest_BGN_IC21.activation(1).proportion = [4,6,5];      % aspect ratio
rest_BGN_IC21.activation(1).falloff = 0.0015;          % parameterizes
 exponential falloff about center, in [0,1]
rest_BGN_IC21.activation(1).minimum   = 0.1;           % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
rest_BGN_IC21.activation(2).region = 'L putamen';
rest_BGN_IC21.activation(2).volume = 1407*voxelVolume; % from Table 2
rest_BGN_IC21.activation(2).center = [-25,-3,0];       % L superior
 temporal gyrus
rest_BGN_IC21.activation(2).rotation = [+10,-10,-10];  % [degrees]
rest_BGN_IC21.activation(2).shape = 'sphere';
rest_BGN_IC21.activation(2).proportion = [4,6,5];      % aspect ratio
rest_BGN_IC21.activation(2).falloff = 0.0015;          % parameterizes
 exponential falloff about center, in [0,1]
rest_BGN_IC21.activation(2).minimum   = 0.1;           % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% define components
disp('Defining the Basal Ganglia Network (IC 21) activated
 regions...')
disp('... L putamen.')
rest_BGN_IC21.activation(1).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_BGN_IC21.activation(1));
disp('... R putamen.')
rest_BGN_IC21.activation(2).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_BGN_IC21.activation(2));

% combine different task components
disp('o Combining activation components.')
NactivationsBGN = length(rest_BGN_IC21.activation);
rest_BGN_IC21.map =
 STANCE_combine_maps('OR',rest_BGN_IC21.activation(:).map);

% clear out working memory (optional)
rest_BGN_IC21.activation(1).map = [];
rest_BGN_IC21.activation(2).map = [];

% find MNI gray matter volume of activation map
rest_BGN_IC21.GMvolume = STANCE_find_GM_volume(rest_BGN_IC21);

% define signal amplitude
rest_BGN_IC21.amplitude = 0.98*0.03; % 3% activation

% % display activation templates
% [~,I_max] = max(sum(sum(rest_BGN_IC21.map)));
% showSliceRA = I_max(1);
% % figure, imshow(imrotate(rest_BGN_IC21.map(:,:,showSliceRA),90),
[]), drawnow;
% TITLE = ['BGN (IC 21) activation regions, A slice:
 ',num2str(showSliceRA)];
% % title(TITLE)
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_BGN_IC21.map,showSliceRA,3,origin);
% title(TITLE)
%
% [~,I_max] = max(sum(sum(rest_BGN_IC21.map,3)));
```

```
% showSliceRC = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_BGN_IC21.map,showSliceRC,2,origin);
% TITLE = ['BGN (IC 21) activation regions, C slice:
 ',num2str(showSliceRC)];
% title(TITLE)
%
% [~,I_max] = max(sum(sum(rest_BGN_IC21.map,3),2));
% showSliceRS = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_BGN_IC21.map,showSliceRS,1,origin);
% TITLE = ['BGN (IC 21) activation regions, S slice:
 ',num2str(showSliceRS)];
% title(TITLE)

TITLE = 'Modeled BGN (IC 21) activation regions';
h_rest_BGN_IC21 =
 STANCE_display_activation_slice(Y_MNI,rest_BGN_IC21.map,[],[]);
title(TITLE)
movegui(h_rest_BGN_IC21,'west');
```

```
Defining the Basal Ganglia Network (IC 21) activated regions...
... L putamen.
o Specifying spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... R putamen.
o Specifying spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
o Combining activation components.
```

Modeled BGN (IC 21) activation regions



Auditory network ...

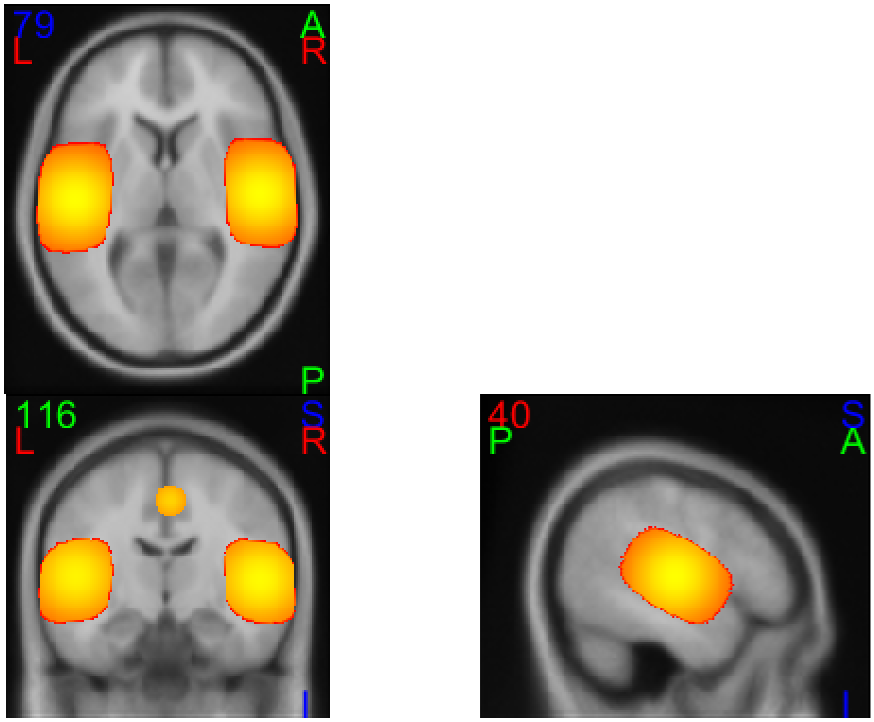
```
uiwait(msgbox('Demo example of the Auditory Network.','Resting state:
 Auditory','modal'));

% define the Auditory Network (IC 17) activated regions
rest_AN_IC17.name = 'Auditory Network: IC 17';
rest_AN_IC17.activation(1).region = 'L superior temporal gyrus';
rest_AN_IC17.activation(1).volume = 2374*voxelVolume; % from Table 2
rest_AN_IC17.activation(1).center = [-51,-18,7];      % Bi precuneus
rest_AN_IC17.activation(1).rotation = [-30,-5,0];     % [degrees]
rest_AN_IC17.activation(1).shape = {'superellipsoid',[3,4]};
rest_AN_IC17.activation(1).proportion = [2,3,2];      % aspect ratio
rest_AN_IC17.activation(1).falloff = 0.001;           % parameterizes
 exponential falloff about center, in [0,1]
rest_AN_IC17.activation(1).minimum   = 0.1;           % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_AN_IC17.activation(2).region = 'R superior temporal gyrus';
rest_AN_IC17.activation(2).volume = 2257*voxelVolume; % from Table 2
rest_AN_IC17.activation(2).center = [52,-15,5];       % Bi precuneus
rest_AN_IC17.activation(2).rotation = [-30,+5,0];     % [degrees]
rest_AN_IC17.activation(2).shape = {'superellipsoid',[3,4]};
rest_AN_IC17.activation(2).proportion = [2,3,2];      % aspect ratio
```

```
rest_AN_IC17.activation(2).falloff = 0.001;           % parameterizes
 exponential falloff about center, in [0,1]
rest_AN_IC17.activation(2).minimum   = 0.1;           % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_AN_IC17.activation(3).region = 'R middle cingulate cortex';
rest_AN_IC17.activation(3).volume = 165*voxelVolume;        % from
 Table 2
rest_AN_IC17.activation(3).center = [2,-4,49];  % Bi precuneus
rest_AN_IC17.activation(3).rotation = [0,0,0];   % [degrees]
rest_AN_IC17.activation(3).shape = 'sphere';
rest_AN_IC17.activation(3).proportion = [1,1,1];      % aspect ratio
rest_AN_IC17.activation(3).falloff = 0.005;      % parameterizes
 exponential falloff about center, in [0,1]
rest_AN_IC17.activation(3).minimum   = 0.2;        % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% define components
disp('Defining the Auditory Network (IC 17) activated regions...')
disp('... activated region 1.')
rest_AN_IC17.activation(1).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_AN_IC17.activation(1));
disp('... activated region 2.')
rest_AN_IC17.activation(2).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_AN_IC17.activation(2));
disp('... activated region 3.')
rest_AN_IC17.activation(3).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_AN_IC17.activation(3));

% combine different task components
disp('o Combining task components.')
NactivationsAN = length(rest_AN_IC17.activation);
rest_AN_IC17.map =
 STANCE_combine_maps('OR',rest_AN_IC17.activation(:).map);

% clear out working memory (optional)
rest_AN_IC17.activation(1).map = [];
rest_AN_IC17.activation(2).map = [];
rest_AN_IC17.activation(3).map = [];


% % display activation templates
% [~,I_max] = max(sum(sum(rest_AN_IC17.map)));
% showSliceRA = I_max(1);
% % figure, imshow(imrotate(rest_AN_IC17.map(:,:,showSliceRA),90),[]),
 drawnow;
% % TITLE = ['AN: IC 17 activation regions, A slice:
 ',num2str(showSliceRA)];
% % title(TITLE)
```

```
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_AN_IC17.map,showSliceRA,3,origin);
% title(TITLE)
%
% [~,I_max] = max(sum(sum(rest_AN_IC17.map,3)));
% showSliceRC = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_AN_IC17.map,showSliceRC,2,origin);
% TITLE = ['AN: IC 17 activation regions, C slice:
 ',num2str(showSliceRC)];
% title(TITLE)
%
% [~,I_max] = max(sum(sum(rest_AN_IC17.map,3),2));
% showSliceRS = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_AN_IC17.map,showSliceRS,1,origin);
% TITLE = ['AN: IC 17 activation regions, S slice:
 ',num2str(showSliceRS)];
% title(TITLE)

TITLE = 'Modeled AN (IC 17) activation regions';
h_rest_AN_IC17 =
 STANCE_display_activation_slice(Y_MNI,rest_AN_IC17.map,[],[]);
title(TITLE);
movegui(h_rest_AN_IC17,'south');
```

```
Defining the Auditory Network (IC 17) activated regions...
... activated region 1.
o Specifying superspherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... activated region 2.
o Specifying superspherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... activated region 3.
o Specifying spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
o Combining task components.
```

Modeled AN (IC 17) activation regions



Default mode network - the first ROI ...

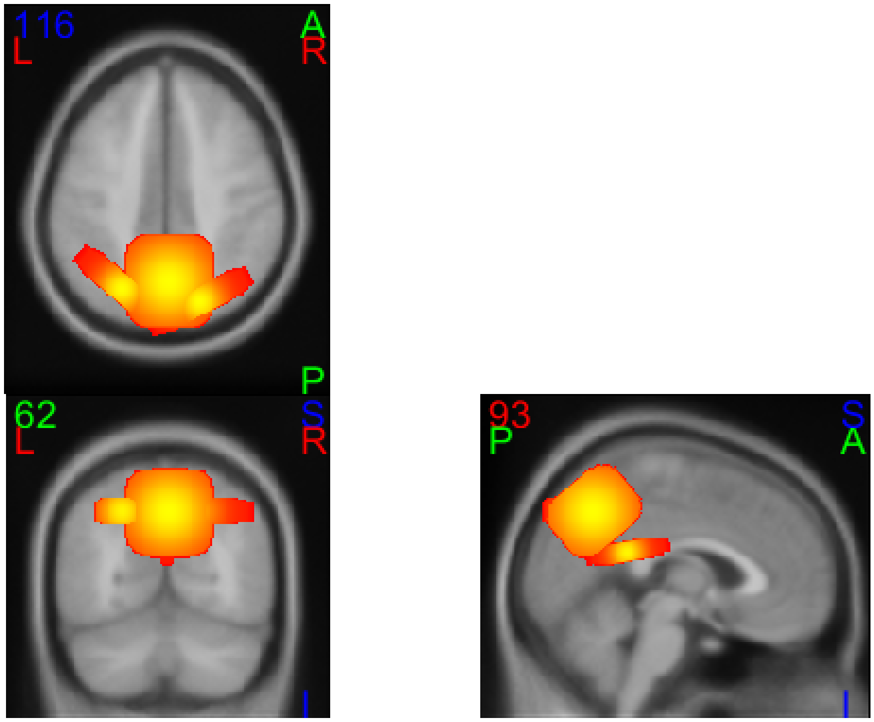
```
uiwait(msgbox('Demo example of the Default Mode Networks.','Resting
 state: DMNs','modal'));

% define the Default Mode Network (IC 50) activated regions
rest_DMN_IC50.name = 'Default-Mode Network: IC 50';
rest_DMN_IC50.activation(1).region = 'Bi precuneus';
rest_DMN_IC50.activation(1).volume = 2425*voxelVolume; % from Table 2
 = 2902 - 677 = 301 each + 75  + overlap
rest_DMN_IC50.activation(1).center = [1,-64,43];       % Bi precuneus
rest_DMN_IC50.activation(1).rotation = [40,0,0];       % [degrees]
rest_DMN_IC50.activation(1).shape = 'squircle';
rest_DMN_IC50.activation(1).proportion = [5,5,4];      % aspect ratio
rest_DMN_IC50.activation(1).falloff = 0.0015;          % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC50.activation(1).minimum   = 0.1;           % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_DMN_IC50.activation(2).region = 'L precuneus';
rest_DMN_IC50.activation(2).volume = 301*voxelVolume; % from Table 2
 = 2902 - 677 = 301 each + 75  + overlap
rest_DMN_IC50.activation(2).center = [-25,-68,43];    % L precuneus
rest_DMN_IC50.activation(2).rotation = [0,0,-40];     % [degrees]
rest_DMN_IC50.activation(2).shape = 'squircle';
```

```
rest_DMN_IC50.activation(2).proportion = [4,1,1];     % aspect ratio
rest_DMN_IC50.activation(2).falloff = 0.005;          % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC50.activation(2).minimum   = 0.1;          % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_DMN_IC50.activation(3).region = 'R precuneus';
rest_DMN_IC50.activation(3).volume = 301*voxelVolume; % from Table 2
 = 2902 - 677 = 301 each + 75  + overlap
rest_DMN_IC50.activation(3).center = [+18,-75,43];    % R precuneus
rest_DMN_IC50.activation(3).rotation = [0,0,+28];     % [degrees]
rest_DMN_IC50.activation(3).shape = 'squircle';
rest_DMN_IC50.activation(3).proportion = [4,1,1];     % aspect ratio
rest_DMN_IC50.activation(3).falloff = 0.005;          % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC50.activation(3).minimum   = 0.1;          % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_DMN_IC50.activation(4).region = 'A precuneus';
rest_DMN_IC50.activation(4).volume = 75*voxelVolume; % from Table 2  =
 2902 - 902 = 401 each + 100
rest_DMN_IC50.activation(4).center = [0,-45,20];     % A precuneus
rest_DMN_IC50.activation(4).rotation = [10,0,0];     % [degrees]
rest_DMN_IC50.activation(4).shape = 'squircle';
rest_DMN_IC50.activation(4).proportion = [1,4,1];    % aspect ratio
rest_DMN_IC50.activation(4).falloff = 0.009;         % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC50.activation(4).minimum   = 0.1;         % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% define components
disp('Defining the Default Mode Network (IC 50) activated regions...')
rest_DMN_IC50.activation(1).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC50.activation(1));
rest_DMN_IC50.activation(2).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC50.activation(2));
rest_DMN_IC50.activation(3).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC50.activation(3));
rest_DMN_IC50.activation(4).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC50.activation(4));

% combine different task components
disp('o Combining activation components.')
NactivationsDMN_IC50 = length(rest_DMN_IC50.activation);
rest_DMN_IC50.map =
 STANCE_combine_maps('OR',rest_DMN_IC50.activation(:).map);

% clear out working memory (optional)
rest_DMN_IC50.activation(1).map = [];
```

```
rest_DMN_IC50.activation(2).map = [];
rest_DMN_IC50.activation(3).map = [];
rest_DMN_IC50.activation(4).map = [];

% [~,I_max] = max(sum(sum(rest_DMN_IC50.map)));
% showSliceRA = I_max(1);
% % figure, imshow(imrotate(rest_DMN_IC50.map(:,:,showSliceRA),90),
[]), drawnow;
% % TITLE = ['DMN:IC 50 activation regions, A slice:
 ',num2str(showSliceRA)];
% title(TITLE)
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC50.map,showSliceRA,3,origin);
% title(TITLE)
%
% [~,I_max] = max(sum(sum(rest_DMN_IC50.map,3)));
% showSliceRC = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC50.map,showSliceRC,2,origin);
% TITLE = ['DMN:IC 50 activation regions, C slice:
 ',num2str(showSliceRC)];
% title(TITLE)
%
% [~,I_max] = max(sum(sum(rest_DMN_IC50.map,3),2));
% showSliceRS = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC50.map,showSliceRS,1,origin);
% TITLE = ['DMN:IC 50 activation regions, S slice:
 ',num2str(showSliceRS)];
% title(TITLE)

TITLE = 'Modeled DMN (IC 50) activation regions';
h_rest_DMN_IC50 =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC50.map,[],[]);
title(TITLE);
movegui(h_rest_DMN_IC50 ,'northwest');
```

```
Defining the Default Mode Network (IC 50) activated regions...
o Specifying superspherical template
o Performing affine transformation (this may take a while).
o Building activation map.
o Specifying superspherical template
o Performing affine transformation (this may take a while).
o Building activation map.
o Specifying superspherical template
o Performing affine transformation (this may take a while).
o Building activation map.
o Specifying superspherical template
o Performing affine transformation (this may take a while).
o Building activation map.
o Combining activation components.
```

Modeled DMN (IC 50) activation regions



The second default mode network ROI

```
rest_DMN_IC53.name = 'Default-Mode Network: IC 53';
rest_DMN_IC53.activation(1).region = 'B PCC';
rest_DMN_IC53.activation(1).volume = 2387*voxelVolume; % from Table 2
rest_DMN_IC53.activation(1).center = [0,-52,22];       % Bi posterior
 cingulate cortex (PCC)
rest_DMN_IC53.activation(1).rotation = [+15,0,0];      % [degrees]
rest_DMN_IC53.activation(1).shape = 'ellipsoid';
rest_DMN_IC53.activation(1).proportion = [5,3,4];      % aspect ratio
rest_DMN_IC53.activation(1).falloff = 0.001;           % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC53.activation(1).minimum   = 0.1;           % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_DMN_IC53.activation(2).region = 'L angular gyrus';
rest_DMN_IC53.activation(2).volume = 332*voxelVolume; % from Table 2
rest_DMN_IC53.activation(2).center = [-43,-69,33];    % L angular
 gyrus
rest_DMN_IC53.activation(2).rotation = [+15,+5,15];   % [degrees]
rest_DMN_IC53.activation(2).shape = 'ellipsoid';
rest_DMN_IC53.activation(2).proportion = [5,3,3];     % aspect ratio
rest_DMN_IC53.activation(2).falloff = 0.0015;         % parameterizes
 exponential falloff about center, in [0,1]
```

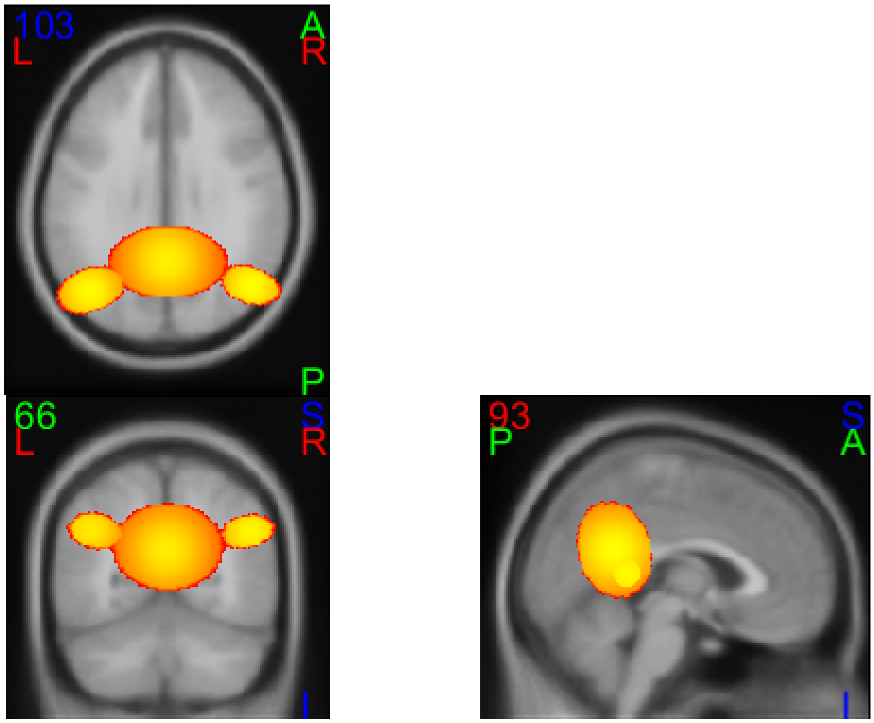
```
rest_DMN_IC53.activation(2).minimum   = 0.1;          % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_DMN_IC53.activation(3).region = 'R angular gyrus';
rest_DMN_IC53.activation(3).volume = 194*voxelVolume; % from Table 2
rest_DMN_IC53.activation(3).center = [47,-66,32];     % R angular
 gyrus
rest_DMN_IC53.activation(3).rotation = [+15,-5,-15];  % [degrees]
rest_DMN_IC53.activation(3).shape = 'ellipsoid';
rest_DMN_IC53.activation(3).proportion = [5,3,3];     % aspect ratio
rest_DMN_IC53.activation(3).falloff = 0.0015;         % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC53.activation(3).minimum   = 0.1;          % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_DMN_IC53.activation(4).region = 'B MFG';
rest_DMN_IC53.activation(4).volume = 61*voxelVolume; % from Table 2
rest_DMN_IC53.activation(4).center = [-1,-45,9];     % Bi medial
 frontal gyrus (MFG)
rest_DMN_IC53.activation(4).rotation = [0,0,0];      % [degrees]
rest_DMN_IC53.activation(4).shape = 'sphere';
rest_DMN_IC53.activation(4).proportion = [1,1,1];    % aspect ratio
rest_DMN_IC53.activation(4).falloff = 0.0015;        % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC53.activation(4).minimum   = 0.1;         % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% define components
disp('Defining the Default Mode Network (IC 53) activated regions...')
disp('... activated region 1.')
rest_DMN_IC53.activation(1).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC53.activation(1));
disp('... activated region 2.')
rest_DMN_IC53.activation(2).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC53.activation(2));
disp('... activated region 3.')
rest_DMN_IC53.activation(3).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC53.activation(3));
disp('... activated region 4.')
rest_DMN_IC53.activation(4).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC53.activation(4));

% combine different task components
disp('o Combining activation components.')
NactivationsDMN_IC53 = length(rest_DMN_IC53.activation);
rest_DMN_IC53.map =
 STANCE_combine_maps('OR',rest_DMN_IC53.activation(:).map);

rest_DMN_IC53.activation(1).map = [];
```

```
rest_DMN_IC53.activation(2).map = [];
rest_DMN_IC53.activation(3).map = [];
rest_DMN_IC53.activation(4).map = [];

% [~,I_max] = max(sum(sum(rest_DMN_IC53.map)));
% showSliceRA = I_max(1);
% % figure, imshow(imrotate(rest_DMN_IC53.map(:,:,showSliceRA),90),
[]), drawnow;
% % TITLE = ['DMN:IC 53 activation regions, A slice:
 ',num2str(showSliceRA)];
% % title(TITLE)
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC53.map,showSliceRA,3,origin);
% title(TITLE)
%
% [~,I_max] = max(sum(sum(rest_DMN_IC53.map,3)));
% showSliceRC = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC53.map,showSliceRC,2,origin);
% TITLE = ['DMN:IC 53 activation regions, C slice:
 ',num2str(showSliceRC)];
% title(TITLE)
%
% [~,I_max] = max(sum(sum(rest_DMN_IC53.map,3),2));
% showSliceRS = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC53.map,showSliceRS,1,origin);
% TITLE = ['DMN:IC 53 activation regions, S slice:
 ',num2str(showSliceRS)];
% title(TITLE)

TITLE = 'Modeled DMN (IC 53) activation regions';
h_rest_DMN_IC53 =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC53.map,[],[]);
title(TITLE);
movegui(h_rest_DMN_IC53 ,'northeast');
```

```
Defining the Default Mode Network (IC 53) 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 spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... activated region 4.
o Specifying spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
```

o Combining activation components.

Modeled DMN (IC 53) activation regions



The third default mode network ROI

```
rest_DMN_IC25.name = 'Default-Mode Network: IC 25';
rest_DMN_IC25.activation(1).region = 'B ACC';
rest_DMN_IC25.activation(1).volume = 3126*voxelVolume; % from Table 2
rest_DMN_IC25.activation(1).center = [0,41,4];         % Bi anterior
 cingulate cortex (ACC)
rest_DMN_IC25.activation(1).rotation = [-5,-5,0];      % [degrees]
rest_DMN_IC25.activation(1).shape = 'squircle';
rest_DMN_IC25.activation(1).proportion = [1,1,1];      % aspect ratio
rest_DMN_IC25.activation(1).falloff = 0.005;           % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC25.activation(1).minimum   = 0.2;           % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_DMN_IC25.activation(2).region = 'L MCC';
rest_DMN_IC25.activation(2).volume = 358*voxelVolume; % from Table 2
rest_DMN_IC25.activation(2).center = [1,-30,41];      % L middle
 cingulate cortex (MCC)
rest_DMN_IC25.activation(2).rotation = [-10,0,0];     % [degrees]
rest_DMN_IC25.activation(2).shape = 'squircle';
rest_DMN_IC25.activation(2).proportion = [1,1,1];     % aspect ratio
rest_DMN_IC25.activation(2).falloff = 0.0015;         % parameterizes
 exponential falloff about center, in [0,1]
```

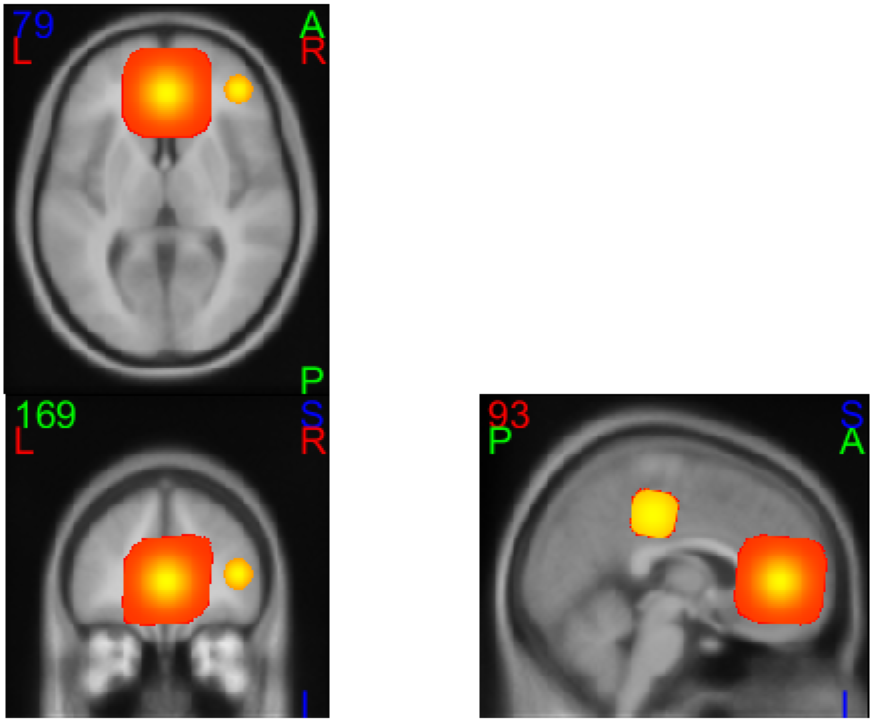
```
rest_DMN_IC25.activation(2).minimum   = 0.1;          % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_DMN_IC25.activation(3).region = 'R IFG';
rest_DMN_IC25.activation(3).volume = 93*voxelVolume; % from Table 2
rest_DMN_IC25.activation(3).center = [32,22,-15];    % R inferior
 frontal gyrus (IFG)
rest_DMN_IC25.activation(3).rotation = [0,0,0];      % [degrees]
rest_DMN_IC25.activation(3).shape = 'sphere';
rest_DMN_IC25.activation(3).proportion = [1,1,1];    % aspect ratio
rest_DMN_IC25.activation(3).falloff = 0.008;         % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC25.activation(3).minimum   = 0.1;         % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_DMN_IC25.activation(4).region = 'R MFG';
rest_DMN_IC25.activation(4).volume = 63*voxelVolume; % from Table 2
rest_DMN_IC25.activation(4).center = [40,43,8];      % R middle
 frontal gyrus (MFG)
rest_DMN_IC25.activation(4).rotation = [0,0,0];      % [degrees]
rest_DMN_IC25.activation(4).shape = 'sphere';
rest_DMN_IC25.activation(4).proportion = [1,1,1];    % aspect ratio
rest_DMN_IC25.activation(4).falloff = 0.008;         % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC25.activation(4).minimum   = 0.1;         % parameterizes
 exponential falloff floor in [0,1]

% define components
disp('Defining the Default Mode Network (IC 25) activated regions...')
disp('... activated region 1.')
rest_DMN_IC25.activation(1).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC25.activation(1));
disp('... activated region 2.')
rest_DMN_IC25.activation(2).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC25.activation(2));
disp('... activated region 3.')
rest_DMN_IC25.activation(3).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC25.activation(3));
disp('... activated region 4.')
rest_DMN_IC25.activation(4).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC25.activation(4));

% combine different task components
% combine different task components
disp('o Combining task components.')
NactivationsDMN_IC25 = length(rest_DMN_IC25.activation);
rest_DMN_IC25.map =
 STANCE_combine_maps('OR',rest_DMN_IC25.activation(:).map);

% clear out working memory (optional)
```

```
rest_DMN_IC25.activation(1).map = [];
rest_DMN_IC25.activation(2).map = [];
rest_DMN_IC25.activation(3).map = [];
rest_DMN_IC25.activation(4).map = [];

% [~,I_max] = max(sum(sum(rest_DMN_IC53.map)));
% showSliceRA = I_max(1);
% % figure, imshow(imrotate(rest_DMN_IC25.map(:,:,showSliceRA),90),
[]), drawnow;
% % TITLE = ['DMN:IC 25 activation regions, A slice:
 ',num2str(showSliceRA)];
% % title(TITLE)
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC25.map,showSliceRA,3,origin);
% title(TITLE)
%
% [~,I_max] = max(sum(sum(rest_DMN_IC25.map,3)));
% showSliceRC = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC25.map,showSliceRC,2,origin);
% TITLE = ['DMN:IC 25 activation regions, C slice:
 ',num2str(showSliceRC)];
% title(TITLE)
%
% [~,I_max] = max(sum(sum(rest_DMN_IC25.map,3),2));
% showSliceRS = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC25.map,showSliceRS,1,origin);
% TITLE = ['DMN:IC 25 activation regions, S slice:
 ',num2str(showSliceRS)];
% title(TITLE)

TITLE = 'Modeled DMN (IC 25) activation regions';
h_rest_DMN_IC25 =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC25.map,[],[]);
title(TITLE);
movegui(h_rest_DMN_IC25 ,'southeast');
```

```
Defining the Default Mode Network (IC 25) activated regions...
... activated region 1.
o Specifying superspherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... activated region 2.
o Specifying superspherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... activated region 3.
o Specifying spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
... activated region 4.
o Specifying spherical template
o Performing affine transformation (this may take a while).
```

```
o Building activation map.
o Combining task components.
```

Modeled DMN (IC 25) activation regions



The fourth default mode network ROI

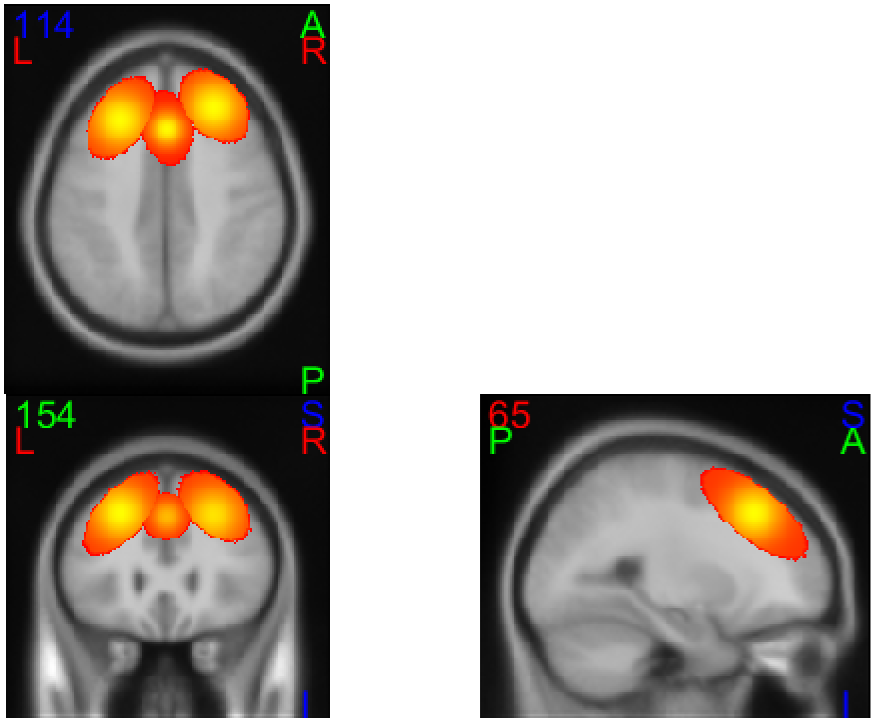
```
rest_DMN_IC68.name = 'Default-Mode Network: IC 68';
rest_DMN_IC68.activation(1).region = 'L MFG';
rest_DMN_IC68.activation(1).volume = 1490*voxelVolume; % from Table 2
rest_DMN_IC68.activation(1).center = [-26,26,42];      % L middle
 frontal gyrus (MFG)
rest_DMN_IC68.activation(1).rotation = [-45,-15,+15];  % [degrees]
rest_DMN_IC68.activation(1).shape = 'sphere';
rest_DMN_IC68.activation(1).proportion = [3,5,2];      % aspect ratio
rest_DMN_IC68.activation(1).falloff = 0.004;           % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC68.activation(1).minimum   = 0.2;           % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_DMN_IC68.activation(2).region = 'R MFG';
rest_DMN_IC68.activation(2).volume = 1210*voxelVolume; % from Table 2
rest_DMN_IC68.activation(2).center = [26,33,41];       % R middle
 frontal gyrus (MFG)
rest_DMN_IC68.activation(2).rotation = [-45,+10,-10];  % [degrees]
rest_DMN_IC68.activation(2).shape = 'sphere';
rest_DMN_IC68.activation(2).proportion = [3,5,2];      % aspect ratio
```

```
rest_DMN_IC68.activation(2).falloff = 0.004;           % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC68.activation(2).minimum   = 0.2;           % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
rest_DMN_IC68.activation(3).region = 'B MCC';
rest_DMN_IC68.activation(3).volume = 450*voxelVolume; % from Table 2
rest_DMN_IC68.activation(3).center = [0,21,40];       % Bi middle
 cingulate cortex
rest_DMN_IC68.activation(3).rotation = [0,0,+10];     % [degrees]
rest_DMN_IC68.activation(3).shape = 'ellipsoid';
rest_DMN_IC68.activation(3).proportion = [2,3,2];     % aspect ratio
rest_DMN_IC68.activation(3).falloff = 0.01;           % parameterizes
 exponential falloff about center, in [0,1]
rest_DMN_IC68.activation(3).minimum   = 0.1;          % parameterizes
 exponential falloff floor in [0,1]
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% define components
disp('Defining the Default Mode Network (IC 25) activated regions...')
disp('... activated region 1.')
rest_DMN_IC68.activation(1).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC68.activation(1));
disp('... activated region 2.')
rest_DMN_IC68.activation(2).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC68.activation(2));
disp('... activated region 3.')
rest_DMN_IC68.activation(3).map =
 STANCE_make_activation_map(dimensions, origin,
 rest_DMN_IC68.activation(3));

% combine different task components
disp('o Combining activation components.')
NactivationsDMN_IC68 = length(rest_DMN_IC68.activation);
rest_DMN_IC68.map    =
 STANCE_combine_maps('OR',rest_DMN_IC68.activation(:).map);

% clear out working memory (optional)
rest_DMN_IC68.activation(1).map = [];
rest_DMN_IC68.activation(2).map = [];
rest_DMN_IC68.activation(3).map = [];


% [~,I_max] = max(sum(sum(rest_DMN_IC68.map)));
% showSliceRA = I_max(1);
% % figure, imshow(imrotate(rest_DMN_IC68.map(:,:,showSliceRA),90),
[]), drawnow;
% % TITLE = ['DMN:IC 68 activation regions, A slice:
 ',num2str(showSliceRA)];
% % title(TITLE)
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC68.map,showSliceRA,3,origin);
```

```
% title(TITLE)
%
% [~,I_max] = max(sum(sum(rest_DMN_IC68.map,3)));
% showSliceRC = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC68.map,showSliceRC,2,origin);
% TITLE = ['DMN:IC 68 activation regions, C slice:
 ',num2str(showSliceRC)];
% title(TITLE)
%
% [~,I_max] = max(sum(sum(rest_DMN_IC68.map,3),2));
% showSliceRS = I_max(1);
% h_task =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC68.map,showSliceRS,1,origin);
% TITLE = ['DMN:IC 68 activation regions, S slice:
 ',num2str(showSliceRS)];
% title(TITLE)

TITLE = 'Modeled DMN (IC68) activation regions';
h_rest_DMN_IC68 =
 STANCE_display_activation_slice(Y_MNI,rest_DMN_IC68.map,[],[]);
title(TITLE);
movegui(h_rest_DMN_IC68 ,'southwest');
```

```
Defining the Default Mode Network (IC 25) 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 spherical template
o Performing affine transformation (this may take a while).
o Building activation map.
o Combining activation components.
```

Modeled DMN (IC68) activation regions



Free up memory and return

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

```
cd(STANCE_genpath)
```

```
cd(currentDir)
```

Published with MATLAB® R2016a