Background

Precision functional mapping (PFM) refers to a suite of approaches for delineating functional brain areas and networks at the individual level.

Example PFM dataset

An example PFM style dataset (“ME01”, study author CJL) is available online at OpenNeuro.org (https://openneuro.org/datasets/ds005118/) in part for the purpose of providing example data for this tutorial. Please cite Lynch et al., 2024 (citation pending) when using this code, and [1] when publishing work that includes the ME01 dataset.

Usage

The goal of this tutorial is to describe step-by-step how PFM is performed in our lab. This tutorial focuses on the ME01 dataset, but in principle the code described below can be adapted to accommodate any fMRI dataset that has been mapped to `fs_LR_32k` surface space (CIFTI format, “.dtseries.nii” file with 64k vertices, 32k vertices per hemisphere when including medial wall). For more information regarding the CIFTI format, see Glasser et al. 2013 Neuroimage (https://pubmed.ncbi.nlm.nih.gov/23668970/).

If you have issues, please email me (Chuck Lynch; cjl2007@med.cornell.edu), and I will try my best to help you.

Before you begin.

Check that all of the necessary dependencies are available.

Start Matlab, open the `pfm_tutorial.m` script in the Matlab editor window, and run lines 5-6 to add all of the necessary dependencies to the Matlab search path.

```
3     %% Before you begin.
4
5     % add dependencies to Matlab search path
6  -  addpath(genpath([pwd '/PFM-Tutorial/Utilities']));
7
8     % define path to some software packages that will be needed
9  -  InfoMapBinary = '/home/charleslynch/miniconda3/bin/infomap'; % path to infomap binary; code tested on version 2.0.0
10 -  WorkbenchBinary = '/usr/local/workbench/bin_linux64/wb_command'; % path to workbench binary; code tested on version 1
11
12    % number of
13    % workers
14 -  nWorkers = 5;
```

You will also need to install Infomap (https://www.mapequation.org/infomap/#Install) and Connectome Workbench (https://www.humanconnectome.org/software/connectome-workbench). Once installed, run lines 8-10 to define two variables called `InfomapBinary` and `WorkbenchBinary`. These variables (string) contain the paths to the binary executables for each of these programs.

```
3     %% Before you begin.
4
5     % add dependencies to Matlab search path
6 -   addpath(genpath([pwd '/PFM-Tutorial/Utilities']));
7
8     % define path to some software packages that will be needed
9 -   InfoMapBinary = '/home/charleslynch/miniconda3/bin/infomap'; % path to infomap binary; code tested on version 2.0.0
10 -  WorkbenchBinary = '/usr/local/workbench/bin_linux64/wb_command'; % path to workbench binary; code tested on version 1
11
12    % number of
13    % workers
14 -  nWorkers = 5;
```

Next, run lines 12-14 to define the number of workers for parallelization of certain procedures. The appropriate number will depend on your particular computing environment.

```
3     %% Before you begin.
4
5     % add dependencies to Matlab search path
6 -   addpath(genpath([pwd '/PFM-Tutorial/Utilities']));
7
8     % define path to some software packages that will be needed
9 -   InfoMapBinary = '/home/charleslynch/miniconda3/bin/infomap'; % path to infomap binary; code tested on version 2.0.0
10 -  WorkbenchBinary = '/usr/local/workbench/bin_linux64/wb_command'; % path to workbench binary; code tested on version 1
11
12    % number of
13    % workers
14 -  nWorkers = 5;
```

Finally, the example dataset can be obtained from OpenNeuro.org (https://openneuro.org/datasets/ds005118/). Please note that this is a large download (~100 GB).

Step 1: Temporal Concatenation of fMRI data from all sessions.

The first step is to temporally concatenate all the individual denoised and fs_LR_32k surface-registered CIFTI (".dtseries.nii") files to obtain a single CIFTI file for the subsequent analysis.

For context, the validity and test-retest reliability of functional connectivity measurements and their derivatives (functional network parcellations) at the single-subject level increases rapidly with additional data. This is why PFM is often performed in individuals that have been scanned repeatedly over an extended period of time, for total scan durations of several hours or more. For example, ME01 was scanned 44 times over 10 imaging sessions (> 10 hours of functional MRI data total). In principle, however, PFM may be performed with far less data when specialized acquisitions are used to improve signal-to-noise.

In the `pfm_tutorial.m` script, run lines 18-20 to define the path to the subject's folder and identifier.

```
16    %% Step 1: Temporal Concatenation of fMRI data from all sessions.
17
18    % define subject directory and name;
19 -  Subdir = [pwd '/WCM-ME/derivatives/sub-ME01/'];
20 -  Subject = 'ME01';
21
22    % define & create
23    % the pfm directory;
24 -  PfmDir = [Subdir '/pfm/'];
25 -  mkdir(PfmDir);
```

Next, run lines 22-25 to define and create the folder (`PfmDir`) where all the outputs will be stored. Note that for convenience, the ME01 dataset is distributed online with this folder and all outputs pre-generated. Delete `PfmDir` before proceeding to start from scratch.

```
16  %% Step 1: Temporal Concatenation of fMRI data from all sessions.
17
18  % define subject directory and name;
19  Subdir = [pwd '/WCM-ME/derivatives/sub-ME01/'];
20  Subject = 'ME01';
21
22  % define & create
23  % the pfm directory;
24  PfmDir = [Subdir '/pfm/'];
25  mkdir(PfmDir);
```

Next, run lines 27-53 to iteratively load each of the denoised and fs_LR_32k surface-registered CIFTI (".dtseries.nii") files. The resting-state time courses will be extracted from the `.data` field of each CIFTI file, and demeaned and motion-censored (time points with framewise displacement > 0.3 mm are discarded) before being temporally concatenated into the `ConcatenatedData` variable.

```
27  % count the number of imaging sessions;
28  nSessions = length(dir([Subdir '/processed_restingstate_timecourses/ses-func*']));
29
30  % preallocate;
31  ConcatenatedData = [];
32
33  % sweep through
34  % the sessions;
35  for i = 1:nSessions
36
37      % count the number of runs in this session
38      nRuns = length(dir([Subdir '/processed_restingstate_timecourses/ses-func' sprintf('%02d',i) '/*run-*.dtseries.nii
39
40      % sweep
41      % through
42      % the runs;
43      for ii = 1:nRuns
44
45          % load the denoised & fs_lr_32k surface-registered CIFTI file for run "ii" from session "i"...
46          Cifti = ft_read_cifti_mod([Subdir '/processed_restingstate_timecourses/ses-func' sprintf('%02d',i) '/sub-' Su
47          Cifti.data = Cifti.data - mean(Cifti.data,2); % demean
48          Tmask = load([Subdir '/processed_restingstate_timecourses/ses-func' sprintf('%02d',i) '/sub-' Subject '_ses-f
49          ConcatenatedData = [ConcatenatedData Cifti.data(:,Tmask==1)]; % 1 (Low motion timepoints) == FD < 0.3mm, 0 (H
50
51      end
52
53  end
54
55  % make a single CIFTI containing
56  % time-series from all scans;
57  ConcatenatedCifti = Cifti;
58  ConcatenatedCifti.data = ConcatenatedData;
59
```

Finally, run lines 55-58 to create a single CIFTI file (`ConcatenatedCifti`) that contains all the motion-censored resting-state fMRI signals for this subject.

```
% count the number of imaging sessions;
nSessions = length(dir([Subdir '/processed_restingstate_timecourses/ses-func*']));

% preallocate;
ConcatenatedData = [];

% sweep through
% the sessions;
for i = 1:nSessions

    % count the number of runs in this session
    nRuns = length(dir([Subdir '/processed_restingstate_timecourses/ses-func' sprintf('%02d',i) '/*run-*.dtseries.nii

    % sweep
    % through
    % the runs;
    for ii = 1:nRuns

        % load the denoised & fs_lr_32k surface-registered CIFTI file for run "ii" from session "i"...
        Cifti = ft_read_cifti_mod([Subdir '/processed_restingstate_timecourses/ses-func' sprintf('%02d',i) '/sub-' Su
        Cifti.data = Cifti.data - mean(Cifti.data,2); % demean
        Tmask = load([Subdir '/processed_restingstate_timecourses/ses-func' sprintf('%02d',i) '/sub-' Subject '_ses-f
        ConcatenatedData = [ConcatenatedData Cifti.data(:,Tmask==1)]; % 1 (Low motion timepoints) == FD < 0.3mm, 0 (H

    end

end

% make a single CIFTI containing
% time-series from all scans;
ConcatenatedCifti = Cifti;
ConcatenatedCifti.data = ConcatenatedData;
```

Step 2: Create distance matrix, regress nearby cortical signals from subcortical structures.

The next step is to create a matrix summarizing the distance between all points in the brain. Geodesic and Euclidean space is used for cortico-cortical (vertex-to-vertex) and subcortical-cortical distance (voxel-to-vertex), respectively.

In the `pfm_tutorial.m` script in the Matlab editor window, and run lines 62-67 to define the path the subject's fs_LR_32k midthickness surfaces (the midpoint between the white and pial surfaces) and run the `pfm_make_dmat` function.

```
60 %% Step 2: Make a distance matrix.
61
62 % define fs_lr_32k midthickness surfaces;
63 MidthickSurfs{1} = [Subdir '/fs_LR/fsaverage_LR32k/' Subject '.L.midthickness.32k_fs_LR.surf.gii'];
64 MidthickSurfs{2} = [Subdir '/fs_LR/fsaverage_LR32k/' Subject '.R.midthickness.32k_fs_LR.surf.gii'];
65
66 % make the distance matrix;
67 pfm_make_dmat(ConcatenatedCifti,MidthickSurfs,PfmDir,nWorkers,WorkbenchBinary); %
68
69 % optional: regress adjacent cortical signal from subcortex to reduce artifactual coupling
70 % (for example, between cerebellum and visual cortex, or between putamen and insular cortex)
71 [ConcatenatedCifti] = pfm_regress_adjacent_cortex(ConcatenatedCifti,[PfmDir '/DistanceMatrix.mat'],20);
72
73 % write out the CIFTI file;
74 ft_write_cifti_mod([Subdir '/pfm/sub-ME01_task-rest_concatenated_32k_fsLR.dtseries.nii'],ConcatenatedCifti);
```

For context, this distance matrix is used in multiple different ways during PFM. As one example, spurious coupling between subcortical voxels and adjacent cortical tissue (e.g., inflated FC between occipital cortex and the cerebellum) can be mitigated by regressing the average time-series of cortical tissue within a specified distance from any subcortical voxel [2,3].

In the `pfm_tutorial.m` script in the Matlab editor window, run lines 69-74 to run the `pfm_regress_adjacent_cortex` function and save the resultant CIFTI file.

```
60    %% Step 2: Make a distance matrix.
61
62    % define fs_lr_32k midthickness surfaces;
63 -  MidthickSurfs{1} = [Subdir '/fs_LR/fsaverage_LR32k/' Subject '.L.midthickness.32k_fs_LR.surf.gii'];
64 -  MidthickSurfs{2} = [Subdir '/fs_LR/fsaverage_LR32k/' Subject '.R.midthickness.32k_fs_LR.surf.gii'];
65
66    % make the distance matrix;
67 -  pfm_make_dmat(ConcatenatedCifti,MidthickSurfs,PfmDir,nWorkers,WorkbenchBinary); %
68
69    % optional: regress adjacent cortical signal from subcortex to reduce artifactual coupling
70    % (for example, between cerebellum and visual cortex, or between putamen and insular cortex)
71 -  [ConcatenatedCifti] = pfm_regress_adjacent_cortex(ConcatenatedCifti,[PfmDir '/DistanceMatrix.mat'],20);
72
73    % write out the CIFTI file;
74 -  ft_write_cifti_mod([Subdir '/pfm/sub-ME01_task-rest_concatenated_32k_fsLR.dtseries.nii'],ConcatenatedCifti);
```

Step 3: Apply the desired amount of spatial smoothing.

In the Matlab editor window, specify a range of kernel sizes (in sigma) at line 80. The example range of sigma values specified below of 0.85, 1.7, and 2.55 correspond to a FWHM of 2mm, 4mm, and 6mm, respectively (FWHM ≈ 2.355 * sigma).

```
76    %% Step 3: Apply spatial smoothing.
77
78    % define a range of gaussian
79    % smoothing kernels (in sigma)
80 -  KernelSizes = [0.85 1.7 2.55];
81
82    % sweep a range of
83    % smoothing kernels;
84 -  for k = KernelSizes
85
86        % smooth with geodesic (for surface data) and Euclidean (for volumetric data) Gaussian kernels;
87 -      system([WorkbenchBinary ' -cifti-smoothing ' PfmDir '/sub-ME01_task-rest_concatenated_32k_fsLR.dtseries.nii '...
88        num2str(k) ' ' num2str(k) ' COLUMN ' PfmDir '/sub-ME01_task-rest_concatenated_smoothed' num2str(k) '_32k_fsLR.dts
89
90 -  end
```

Next, run lines 82-90 to apply the specified levels of spatial smoothing to the concatenated CIFTI file with geodesic (for cortical vertices) and Euclidean (for subcortical voxels) Gaussian kernels using Connectome Workbench command line utilities. Note that this step can be considered optional, but is recommended for most datasets to improve signal-to-noise.

```
76    %% Step 3: Apply spatial smoothing.
77
78    % define a range of gaussian
79    % smoothing kernels (in sigma)
80 -  KernelSizes = [0.85 1.7 2.55];
81
82    % sweep a range of
83    % smoothing kernels;
84 -  for k = KernelSizes
85
86        % smooth with geodesic (for surface data) and Euclidean (for volumetric data) Gaussian kernels;
87 -      system([WorkbenchBinary ' -cifti-smoothing ' PfmDir '/sub-ME01_task-rest_concatenated_32k_fsLR.dtseries.nii '...
88        num2str(k) ' ' num2str(k) ' COLUMN ' PfmDir '/sub-ME01_task-rest_concatenated_smoothed' num2str(k) '_32k_fsLR.dt
89
90 -  end
```

Step 4: Run infomap.

The Infomap community detection algorithm (https://www.mapequation.org/infomap/) is one of the most widely used approaches for delineating functional brain networks and their boundaries in individuals. The `pfm_infomap` function is a wrapper that encompasses multiple steps —

including creating and thresholding the functional connectivity (FC) matrix, calling the Infomap algorithm, and saving the resultant Infomap communities to a CIFTI file.

In the Matlab editor window, run line 95 to load the CIFTI file that you want to run Infomap on. In the example below, the CIFTI file with 2.55 sigma spatial smoothing is selected.

```
92   %% Step 4: Run infomap.
93
94   % load your concatenated resting-state dataset, pick whatever level of spatial smoothing you want
95   ConcatenatedCifti = ft_read_cifti_mod([PfmDir '/sub-ME01_task-rest_concatenated_smoothed2.55_32k_fsLR.dtseries.nii'])
96
97   % define inputs;
98   DistanceMatrix = [Subdir '/pfm/DistanceMatrix.mat']; % can be path to file
99   DistanceCutoff = 10; % in mm; usually between 10 to 30 mm works well.
100  GraphDensities = flip([0.0001 0.0002 0.0005 0.001 0.002 0.005 0.01 0.02 0.05]); %
101  NumberReps = 50; % number of times infomap is run;
102  BadVertices = []; % optional, but you could include regions to ignore, if you know there is bad signal there.
103  Structures = {'CORTEX_LEFT','CEREBELLUM_LEFT','ACCUMBENS_LEFT','CAUDATE_LEFT','PALLIDUM_LEFT','PUTAMEN_LEFT','THALAMU
104
105  % run infomap
106  pfm_infomap(ConcatenatedCifti,DistanceMatrix,PfmDir,GraphDensities,NumberReps,DistanceCutoff,BadVertices,Structures,r
107
108  % remove some intermediate files (optional)
109  system(['rm ' Subdir '/pfm/*.net']);
110  system(['rm ' Subdir '/pfm/*.clu']);
111  system(['rm ' Subdir '/pfm/*Log*']);
112
113  % define inputs;
114  Input = [PfmDir '/Bipartite_PhysicalCommunities.dtseries.nii'];
115  Output = 'Bipartite_PhysicalCommunities+SpatialFiltering.dtseries.nii';
116  MinSize = 50; % in mm^2
117
118  % perform spatial filtering
119  pfm_spatial_filtering(Input,PfmDir,Output,MidthickSurfs,MinSize,WorkbenchBinary);
120
```

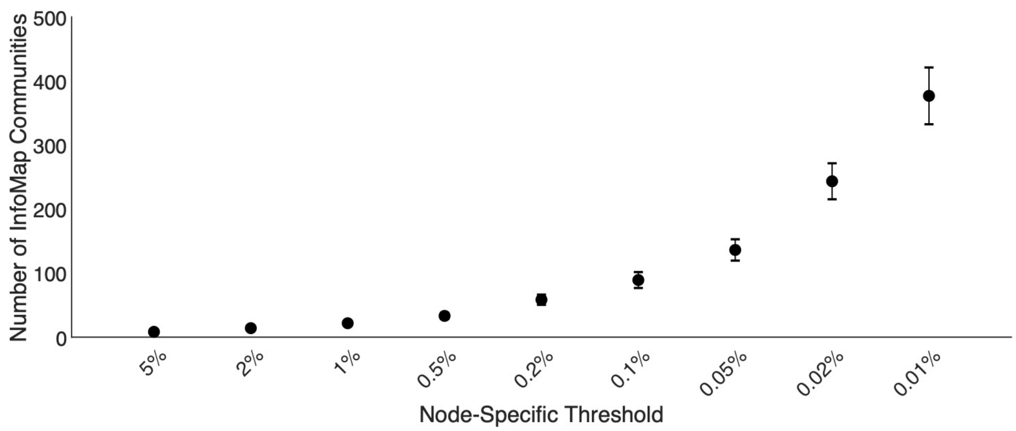
Next, in the Matlab editor window, the following inputs must be defined by the user at lines 97-103 in the `pfm_tutorial.m` script.


```
92   %% Step 4: Run infomap.
93
94   % load your concatenated resting-state dataset, pick whatever level of spatial smoothing you want
95   ConcatenatedCifti = ft_read_cifti_mod([PfmDir '/sub-ME01_task-rest_concatenated_smoothed2.55_32k_fsLR.dtseries.nii'])
96
97   % define inputs;
98   DistanceMatrix = [Subdir '/pfm/DistanceMatrix.mat']; % can be path to file
99   DistanceCutoff = 10; % in mm; usually between 10 to 30 mm works well.
100  GraphDensities = flip([0.0001 0.0002 0.0005 0.001 0.002 0.005 0.01 0.02 0.05]); %
101  NumberReps = 50; % number of times infomap is run;
102  BadVertices = []; % optional, but you could include regions to ignore, if you know there is bad signal there.
103  Structures = {'CORTEX_LEFT','CEREBELLUM_LEFT','ACCUMBENS_LEFT','CAUDATE_LEFT','PALLIDUM_LEFT','PUTAMEN_LEFT','THALAMU
104
105  % run infomap
106  pfm_infomap(ConcatenatedCifti,DistanceMatrix,PfmDir,GraphDensities,NumberReps,DistanceCutoff,BadVertices,Structures,r
107
108  % remove some intermediate files (optional)
109  system(['rm ' Subdir '/pfm/*.net']);
110  system(['rm ' Subdir '/pfm/*.clu']);
111  system(['rm ' Subdir '/pfm/*Log*']);
112
113  % define inputs;
114  Input = [PfmDir '/Bipartite_PhysicalCommunities.dtseries.nii'];
115  Output = 'Bipartite_PhysicalCommunities+SpatialFiltering.dtseries.nii';
116  MinSize = 50; % in mm^2
117
118  % perform spatial filtering
119  pfm_spatial_filtering(Input,PfmDir,Output,MidthickSurfs,MinSize,WorkbenchBinary);
120
```


`DistanceMatrix` is the path (string) to the distance matrix created earlier during Step 2. `DistanceCutoff` is the threshold (numeric, in mm) for removing short-distance correlations in the FC matrix. Correlations between nodes < `DistanceCutoff` from each other will be set to zero. This is done to mitigate the effects of spatial autocorrelation on the network structures identified.

`GraphDensities` is a numeric vector of graph densities — the percentage of the top connections retained by each node after thresholding. So for example, in a hypothetical 1000 x 1000 FC matrix, a graph density of 0.05 (5%) means that each node will retain its top 50 strongest connections.

By default the total number of communities identified by Infomap is data-driven, but can be controlled in part by how many connections are retained in the functional connectivity matrix after thresholding. For example, fewer communities are identified at the 5% threshold (on average, 8.28 ± 1.21) than at the 0.1% threshold (on average, 89.13 ± 8.04). We recommend running Infomap over a range of graph densities (e.g., 5% to 0.01%, as done in [4]).



`NumberReps` is a numeric value representing the number of times the Infomap algorithm is run before selecting the best solution.
`BadVertices` is an optional index of all points in the brain the user would like to omit from the analysis. For example, if the user knows that data quality is especially poor (low tSNR or test-retest reliability) in a particular set of brain regions. Otherwise, set to `[]` to include all vertices and voxels.
`BrainStructures` is a cell array of brain structures of interest from the `.brainstructurelabel` field of the `ConcatenatedCifti` file. The FC matrix will omit nodes from brain structures not included in this variable.

Run lines 105-106 in the `pfm_tutorial.m` script to run the `pfm_infomap` function.

```
92      %% Step 4: Run infomap.
93
94      % load your concatenated resting-state dataset, pick whatever level of spatial smoothing you want
95  -   ConcatenatedCifti = ft_read_cifti_mod([PfmDir '/sub-ME01_task-rest_concatenated_smoothed2.55_32k_fsLR.dtseries.nii'])
96
97      % define inputs;
98  -   DistanceMatrix = [Subdir '/pfm/DistanceMatrix.mat']; % can be path to file
99  -   DistanceCutoff = 10; % in mm; usually between 10 to 30 mm works well.
100 -   GraphDensities = flip([0.0001 0.0002 0.0005 0.001 0.002 0.005 0.01 0.02 0.05]); %
101 -   NumberReps = 50; % number of times infomap is run;
102 -   BadVertices = []; % optional, but you could include regions to ignore, if you know there is bad signal there.
103 -   Structures = {'CORTEX_LEFT','CEREBELLUM_LEFT','ACCUMBENS_LEFT','CAUDATE_LEFT','PALLIDUM_LEFT','PUTAMEN_LEFT','THALAMU
104
105     % run infomap
106 -   pfm_infomap(ConcatenatedCifti,DistanceMatrix,PfmDir,GraphDensities,NumberReps,DistanceCutoff,BadVertices,Structures,r
107
108     % remove some intermediate files (optional)
109 -   system(['rm ' Subdir '/pfm/*.net']);
110 -   system(['rm ' Subdir '/pfm/*.clu']);
111 -   system(['rm ' Subdir '/pfm/*Log*']);
112
113     % define inputs;
114 -   Input = [PfmDir '/Bipartite_PhysicalCommunities.dtseries.nii'];
115 -   Output = 'Bipartite_PhysicalCommunities+SpatialFiltering.dtseries.nii';
116 -   MinSize = 50; % in mm^2
117
118     % perform spatial filtering
119 -   pfm_spatial_filtering(Input,PfmDir,Output,MidthickSurfs,MinSize,WorkbenchBinary);
120
```

Optionally, users can discard implausibly small (i.e., smaller than the effective resolution of underlying data) patches of a community by specifying a minimum cluster size (in mm²). This procedure acts as a spatial filter — removing small objects without imposing additional spatial smoothing on the underlying data. The neighboring network identities will then be dilated one vertex at a time until the region is filled, as done in [5].

Run lines 113-119 in the pfm_tutorial.m script to define inputs and run the `pfm_spatial_filtering` function.

```
92      %% Step 4: Run infomap.
93
94      % load your concatenated resting-state dataset, pick whatever level of spatial smoothing you want
95  -   ConcatenatedCifti = ft_read_cifti_mod([PfmDir '/sub-ME01_task-rest_concatenated_smoothed2.55_32k_fsLR.dtseries.nii'])
96
97      % define inputs;
98  -   DistanceMatrix = [Subdir '/pfm/DistanceMatrix.mat']; % can be path to file
99  -   DistanceCutoff = 10; % in mm; usually between 10 to 30 mm works well.
100 -   GraphDensities = flip([0.0001 0.0002 0.0005 0.001 0.002 0.005 0.01 0.02 0.05]); %
101 -   NumberReps = 50; % number of times infomap is run;
102 -   BadVertices = []; % optional, but you could include regions to ignore, if you know there is bad signal there.
103 -   Structures = {'CORTEX_LEFT','CEREBELLUM_LEFT','ACCUMBENS_LEFT','CAUDATE_LEFT','PALLIDUM_LEFT','PUTAMEN_LEFT','THALAMU
104
105     % run infomap
106 -   pfm_infomap(ConcatenatedCifti,DistanceMatrix,PfmDir,GraphDensities,NumberReps,DistanceCutoff,BadVertices,Structures,r
107
108     % remove some intermediate files (optional)
109 -   system(['rm ' Subdir '/pfm/*.net']);
110 -   system(['rm ' Subdir '/pfm/*.clu']);
111 -   system(['rm ' Subdir '/pfm/*Log*']);
112
113     % define inputs;
114 -   Input = [PfmDir '/Bipartite_PhysicalCommunities.dtseries.nii'];
115 -   Output = 'Bipartite_PhysicalCommunities+SpatialFiltering.dtseries.nii';
116 -   MinSize = 50; % in mm^2
117
118     % perform spatial filtering
119 -   pfm_spatial_filtering(Input,PfmDir,Output,MidthickSurfs,MinSize,WorkbenchBinary);
120
```

Step 5: Algorithmic assignment of network identities to infomap communities.

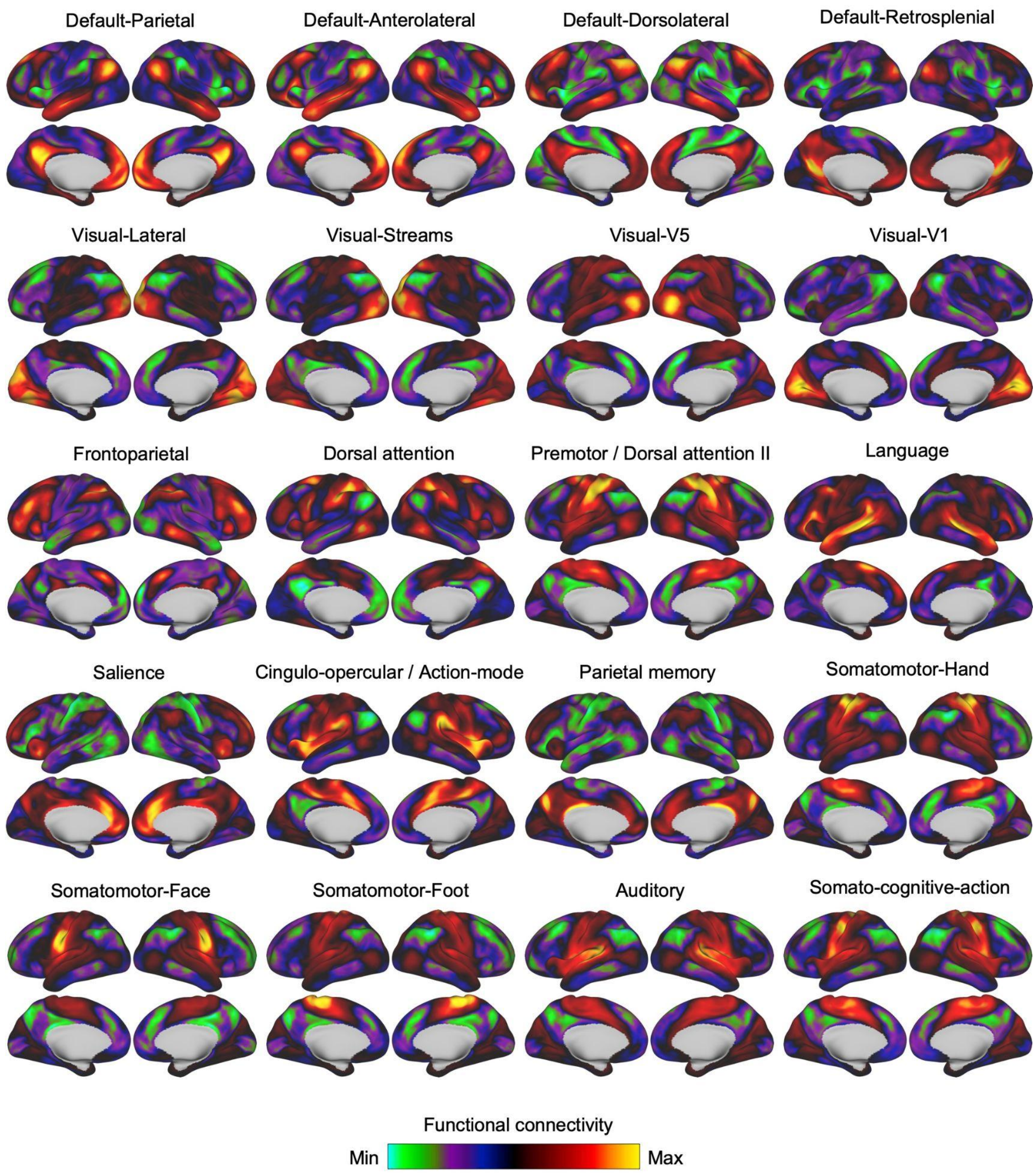
The main output of `pfm_infomap` is a CIFTI file called `“Bipartite_PhysicalCommunities.dtseries.nii”`, which contains the Infomap communities obtained at each graph density (each column represents a different graph density). These community labels are arbitrary — in other words, community number 1 will not represent the same functional network in different individuals.

We assign each Infomap community to one of 20 known functional network identities based on their spatial locations and functional connectivity. In principle, this could be accomplished manually by an expert familiar with functional network topography in individuals, but this can be time consuming and difficult to scale. To help accelerate and standardize the network identification process, we have created a semi-automated procedure for quantifying the likelihood of an Infomap community belonging to a particular functional brain network, and specifying the best match as the initial assignment. The entire procedure is implemented using the `pfm_identify_networks` function.

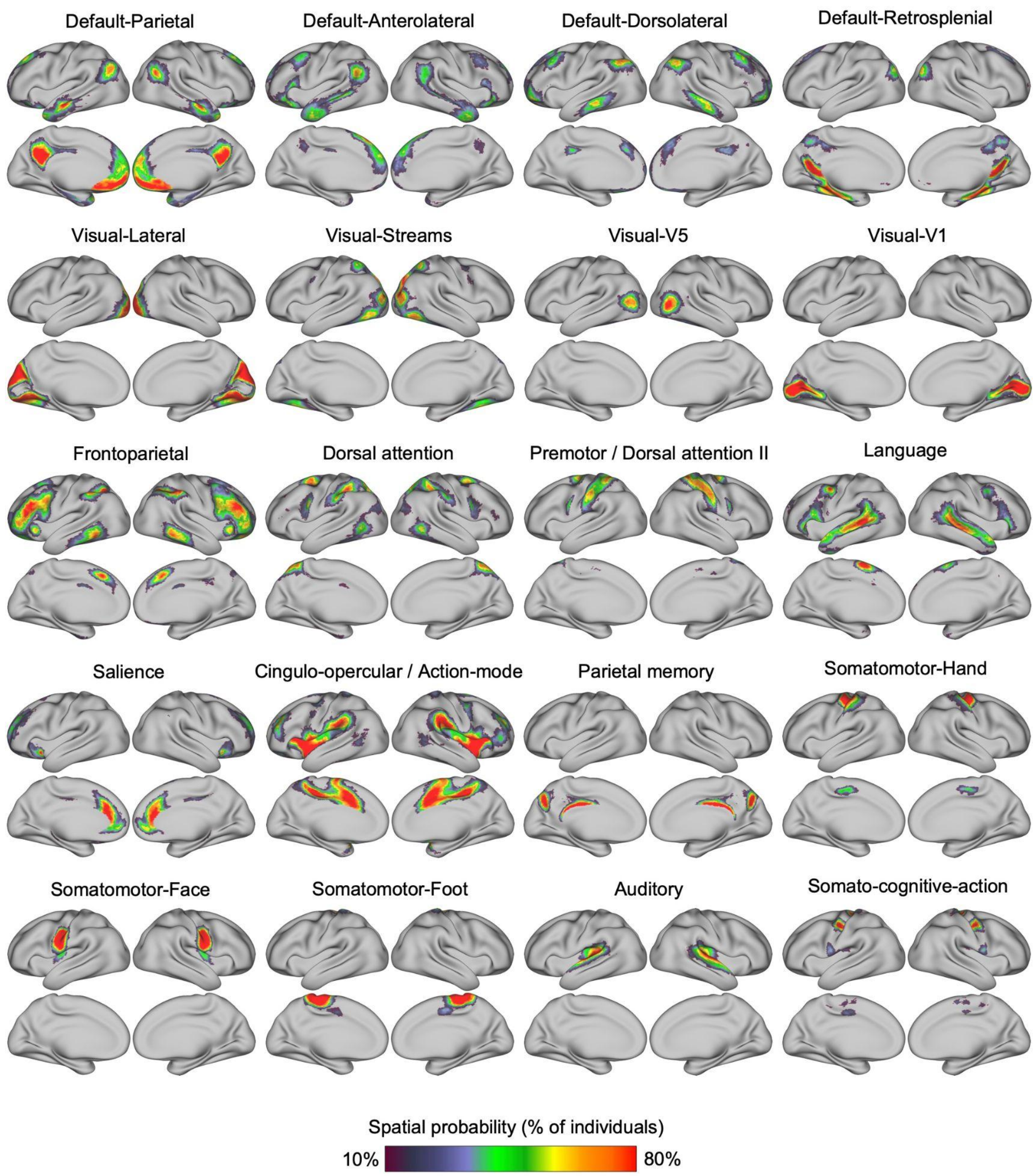
First, in the Matlab editor window, run lines 123-124 in the `pfm_tutorial.m` script to load the default priors for the network identification algorithm.

```
121     %% Step 5: Algorithmic assignment of network identities to infomap communities.
122
123     % load the priors;
124 -   load('priors.mat');
125
126     % define inputs;
127 -   Ic = ft_read_cifti_mod([PfmDir '/Bipartite_PhysicalCommunities+SpatialFiltering.dtseries.nii']);
128 -   Output = 'Bipartite_PhysicalCommunities+AlgorithmicLabeling';
129 -   Column = 6; % column 6, representing graph density 0.01% in this example.
130
131     % run the network identification algorithm;
132 -   pfm_identify_networks(ConcatenatedCifti,Ic,MidthickSurfs,Column,Priors,Output,PfmDir,WorkbenchBinary);
133
```

`Priors` is a structure array containing the average FC (`Priors.FC`) and spatial locations (`Priors.Spatial`) of 20 functional brain networks — (Default-Parietal, Default-Anterolateral, Default-Dorsolateral, Default-Retrosplenial, Visual-Lateral, Visual-Dorsal/Ventral Stream, Visual-V1, Visual-V5, Frontoparietal, Dorsal Attention, Premotor / Dorsal Attention II, Language, Salience, Cingulo-opercular / Action-mode, Parietal memory, Auditory, Somatomotor-Hand, Somatomotor-Face, Somatomotor-Foot, Auditory, or Somato-Cognitive-Action). The code will accept other sets of priors if they are organized in the same way.



Default-Parietal
Default-Anterolateral
Default-Dorsolateral
Default-Retrosplenial
Visual-Lateral
Visual-Streams
Visual-V5
Visual-V1
Frontoparietal
Dorsal attention
Premotor / Dorsal attention II
Language
Salience
Cingulo-opercular / Action-mode
Parietal memory
Somatomotor-Hand
Somatomotor-Face
Somatomotor-Foot
Auditory
Somato-cognitive-action
Functional connectivity
Min
Max



Next, in the Matlab editor window, run lines 126-129 in the `pfm_tutorial.m` script to define the other inputs for the `pfm_identify_networks` function.

```
121  %% Step 5: Algorithmic assignment of network identities to infomap communities.
122
123  % load the priors;
124  load('priors.mat');
125
126  % define inputs;
127  Ic = ft_read_cifti_mod([PfmDir '/Bipartite_PhysicalCommunities+SpatialFiltering.dtseries.nii']);
128  Output = 'Bipartite_PhysicalCommunities+AlgorithmicLabeling';
129  Column = 6; % column 6, representing graph density 0.01% in this example.
130
131  % run the network identification algorithm;
132  pfm_identify_networks(ConcatenatedCifti,Ic,MidthickSurfs,Column,Priors,Output,PfmDir,WorkbenchBinary);
133
```

`InfomapCommunities` is the CIFTI file created by `pfm_infomap`. It contains the Infomap communities obtained at each graph density (each column represents a different graph density). `Column` is the column of `InfomapCommunities` that the user wants to assign network identities. `Output` is the name for the output file.

Next, in the Matlab editor window, run lines 131-132 in the `pfm_tutorial.m` script to run the `pfm_identify_networks` function.


```
121  %% Step 5: Algorithmic assignment of network identities to infomap communities.
122
123  % load the priors;
124  load('priors.mat');
125
126  % define inputs;
127  Ic = ft_read_cifti_mod([PfmDir '/Bipartite_PhysicalCommunities+SpatialFiltering.dtseries.nii']);
128  Output = 'Bipartite_PhysicalCommunities+AlgorithmicLabeling';
129  Column = 6; % column 6, representing graph density 0.01% in this example.
130
131  % run the network identification algorithm;
132  pfm_identify_networks(ConcatenatedCifti,Ic,MidthickSurfs,Column,Priors,Output,PfmDir,WorkbenchBinary);
133
```


Step 6: Review algorithmic network assignments, adjust if needed.

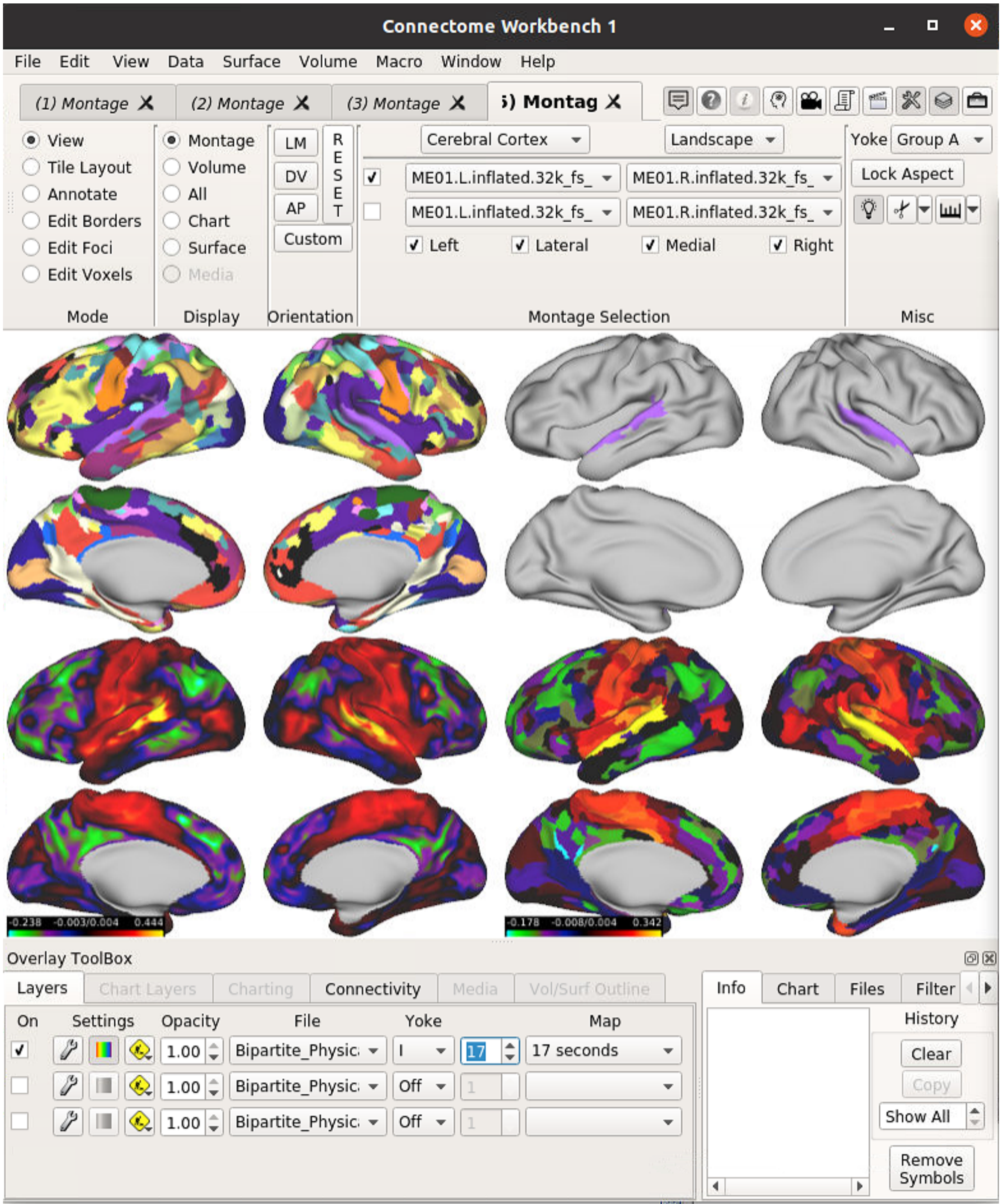
There are multiple outputs generated by the `pfm_identify_networks` function that we recommend reviewing carefully before proceeding.

First, there is an .XLS sheet containing the winning and runner-up network assignments for each Infomap community, as well as a brief summary of the information that influenced algorithmic assignments. Second, there are a series of CIFTI files that are highlighted and described below.

Name	Size	Type
Bipartite_PhysicalCommunities.dtseries.nii	3.6 MB	Binary
Bipartite_PhysicalCommunities+AlgorithmicLabeling.dlabel.nii	919.6 kB	Binary
Bipartite_PhysicalCommunities+AlgorithmicLabeling.L.border	582.2 kB	Marku
Bipartite_PhysicalCommunities+AlgorithmicLabeling.R.border	591.8 kB	Marku
Bipartite_PhysicalCommunities+AlgorithmicLabeling_FC_btwn_InfoMapCommunities.dtseries.nii	26.8 MB	Binary
Bipartite_PhysicalCommunities+AlgorithmicLabeling_FC_btwn_InfoMapCommunities.pdf	143.3 kB	Docum
Bipartite_PhysicalCommunities+AlgorithmicLabeling_FC_WholeBrain.dtseries.nii	26.8 MB	Binary
Bipartite_PhysicalCommunities+AlgorithmicLabeling_InfoMapCommunities.dlabel.nii	27.0 MB	Binary

`Bipartite_PhysicalCommunities+AlgorithmicLabeling.dlabel.nii` is a CIFTI file containing the initial algorithmic network assignments for all Infomap communities (set in Tab 1 in images below). The colors used to represent different functional brain networks are set by the RGB values in `Priors.NetworkColors`.
`Bipartite_PhysicalCommunities+AlgorithmicLabeling_InfoMapCommunities.dlabel.nii` is a CIFTI file containing the initial algorithmic network assignment for each Infomap community, stored separately in each column (set in Tab 2 in images below).
`Bipartite_PhysicalCommunities+AlgorithmicLabeling_FC_WholeBrain.dtseries.nii` is a CIFTI file containing the whole-brain functional connectivity of each Infomap community, stored separately in each column (set in Tab 3 in images below).
`Bipartite_PhysicalCommunities+AlgorithmicLabeling_FC_btwn_InfoMapCommunities.dtseries.nii` is a CIFTI file containing the functional connectivity of each Infomap community to all other Infomap communities, stored separately in each column (set in Tab 4 in images below).

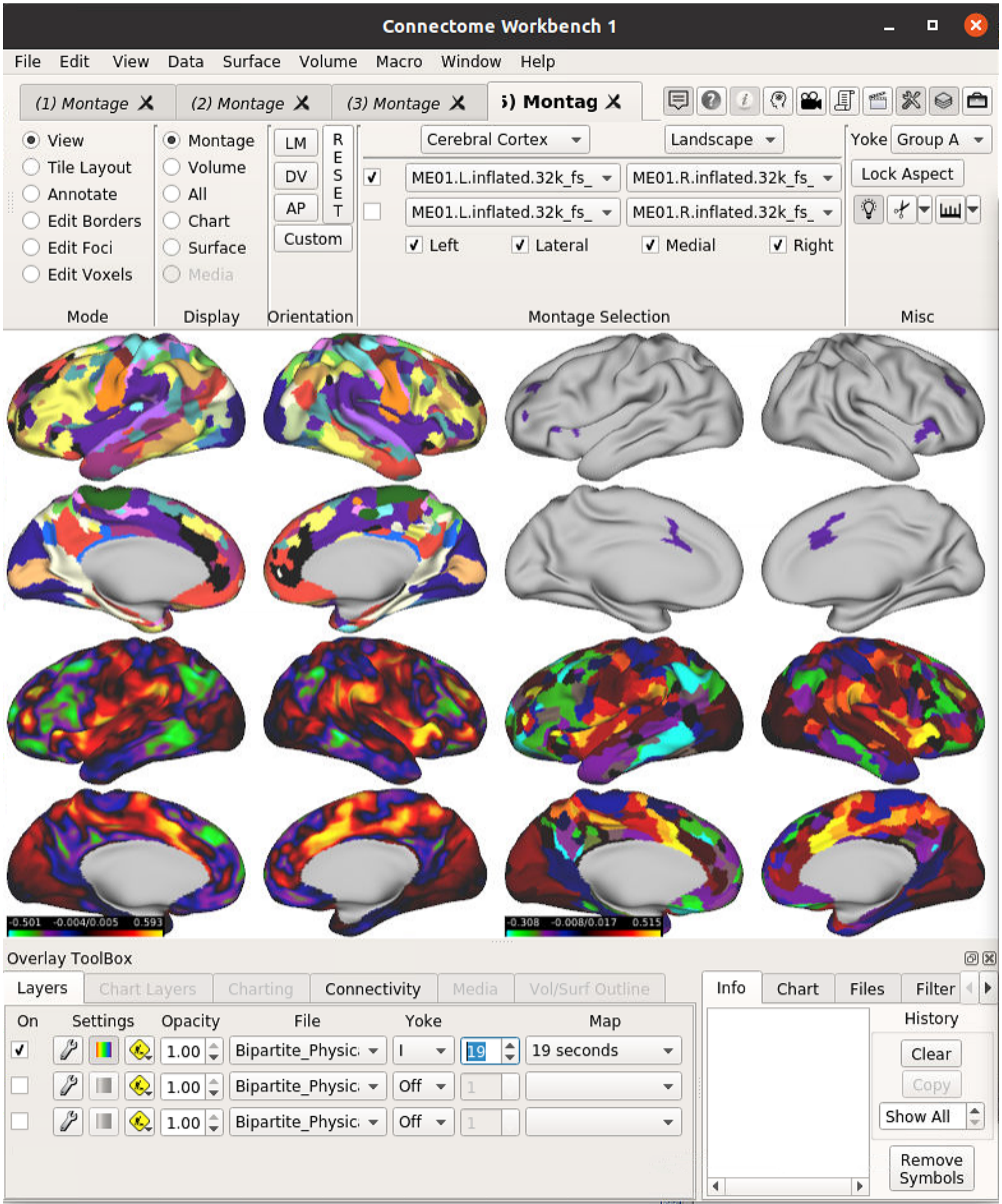
To review the algorithmic network assignments, load the subject's anatomical data and the four CIFTI files above into Connectome Workbench. We recommend creating a 2 x 2 tile tab configuration to simultaneously view all of the relevant information. Tabs 2-4 can be set to "Yoke I" in the *Overlay Toolbox*, so that when you flip through the columns of one file, the others will also update. This allows you to flip through individual communities and view their algorithmic assignment and spatial locations (Tab 2), as well as their functional connectivity with the rest of the brain (Tab 3) and with other communities (Tab 4).



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Community	Network	Network_ManualDecision	R	G	B	FC_Similarity	Spatial_Score	Confidence	Alt_1_Network	Alt_1_FC_Similarity	Alt_1_Spatial_Score	Alt_2_Network	Alt_2_FC_Similarity	Alt_2_Spatial_Score
2	1	Default_Dorsolateral		0.839215686	0.619607843	0.254901961	0.741194874	0.315544575	0.59138351	Frontoparietal	0.627108023	0.234355912	Default_Anterolateral	0.350683717	0.138700158
3	2	Visual_Lateral		0.105882353	0.007843137	0.57254902	0.861742951	0.460733496	0.480725045	Visual_Dorsal/VentralStream	0.752496129	0.356327085	Visual_V5	0.56645345	0.042205922
4	3	Default_Dorsolateral		0.839215686	0.619607843	0.254901961	0.824409303	0.33047391	0.586898797	Frontoparietal	0.667543126	0.257188475	Default_Parietal	0.391039042	0.161666304
5	4	Frontoparietal		0.996078431	1	0.329411765	0.808506474	0.474387302	5.61651861	Default_Dorsolateral	0.451976867	0.128253968	DorsalAttention	0.317082841	0.105752381
6	5	Visual_V1		1	0.709803922	0.388235294	0.842521205	0.8157277	6.237666855	Visual_Lateral	0.734929168	0.12920579	Default_Retrosplenial	0.083778525	0.00312989
7	6	Default_Retrosplenial		1	1	0.854901961	0.768957052	0.549095726	7.768371678	Visual_V1	0.254595786	0.18913852	Default_Parietal	0.579871248	0.040501299
8	7	Somatomotor_Hand		0.439215686	1	0.996078431	0.903785228	0.334304207	0.15268445	Premotor/DorsalAttentionII	0.796442649	0.329110803	CinguloOpercular/Action-mode	0.372021893	0.145292033
9	8	Default_Anterolateral		0.560784314	0.105882353	0.419607843	0.674348555	0.280761998	0.148461135	Language	0.560442939	0.294154198	Default_Parietal	0.597210426	0.182536179
10	9	Somatomotor_Foot		0.043137255	0.329411765	0.011764706	0.908714521	0.732879819	13.69738442	SomatoCognitiveAction	0.767282422	0.059056122	CinguloOpercular/Action-mode	0.395208713	0.10564059
11	10	Somatomotor_Face		1	0.501960784	0	0.911512329	0.706101477	8.001444999	SomatoCognitiveAction	0.890805526	0.080266538	CinguloOpercular/Action-mode	0.581643465	0.076316635
12	11	Visual_Lateral		0.105882353	0.007843137	0.57254902	0.832729355	0.702574756	3.185805458	Visual_V1	0.744006396	0.187862764	Visual_Dorsal/VentralStream	0.683994883	0.03083412
13	12	CinguloOpercular/Action-mode		0.274509804	0.02745098	0.576470588	0.831340674	0.697625179	12.39999669	Auditory	0.657300764	0.065846447	Premotor/DorsalAttentionII	0.608059432	0.045150215
14	13	CinguloOpercular/Action-mode		0.274509804	0.02745098	0.576470588	0.849826686	0.725513784	15.78884704	SomatoCognitiveAction	0.768785458	0.047769424	Somatomotor_Face	0.592046739	0.059131161
15	14	Salience		0	0	0	0.585783462	0.302410131	1.508982275	Default_Parietal	0.491794554	0.143566176	Default_Dorsolateral	0.588281345	0.116094771
16	15	Frontoparietal		0.996078431	1	0.329411765	0.778770019	0.465902266	3.277669574	Default_Dorsolateral	0.617928835	0.137264537	MedialParietal	0.349613949	0.04024024
17	16	Default_Parietal		0.917647059	0.2	0.137254902	0.708382861	0.346835238	0.424018065	Default_Anterolateral	0.702237192	0.24569252	Language	0.378996237	0.08321582
18	17	Auditory		0.666666667	0.329411765	0.97254902	0.880931112	0.569699136	5.073237907	Language	0.392928794	0.210306822	CinguloOpercular/Action-mode	0.674951339	0.113208222
19	18	Somatomotor_Hand		0.439215686	1	0.996078431	0.911957634	0.611176857	2.835696616	Premotor/DorsalAttentionII	0.796363556	0.182467675	SomatoCognitiveAction	0.743561062	0.046657901
20	19	CinguloOpercular/Action-mode		0.274509804	0.02745098	0.576470588	0.767146814	0.660395062	13.84839844	Salience	0.203871326	0.167358025	DorsalAttention	0.301488635	0.010592593
21	20	Language		0.250980392	0.6	0.6	0.56898731	0.332826187	1.24870953	Default_Anterolateral	0.402101221	0.209435992	Default_Parietal	0.276429466	0.131396957

For example, community 17 (highlighted above), was labeled *Auditory* (Column B). The spatial correlation of community 1 FC with the *Auditory* template FC was r = 0.88 (Column G). The average *Auditory* spatial probability value in community 1 was 0.57 (Column H). The total score for this winning assignment is the product of the FC_Similarity and Spatial_Score (0.88 * 0.57 = 0.50). The total score for the first runner-up assignment (*Language*) is also the product of the FC_Similarity and Spatial_Score (0.39 * 0.21 = 0.08). The “Confidence” of the winning assignment (0.59) is quantified in Column I, and is defined as the relative difference in total score associated with the winning and first runner-up network assignments ([0.502 - 0.083] / 0.083 = 5.07). In this case, the high confidence score and impression after visual inspection is that this is a sensible label for this community.

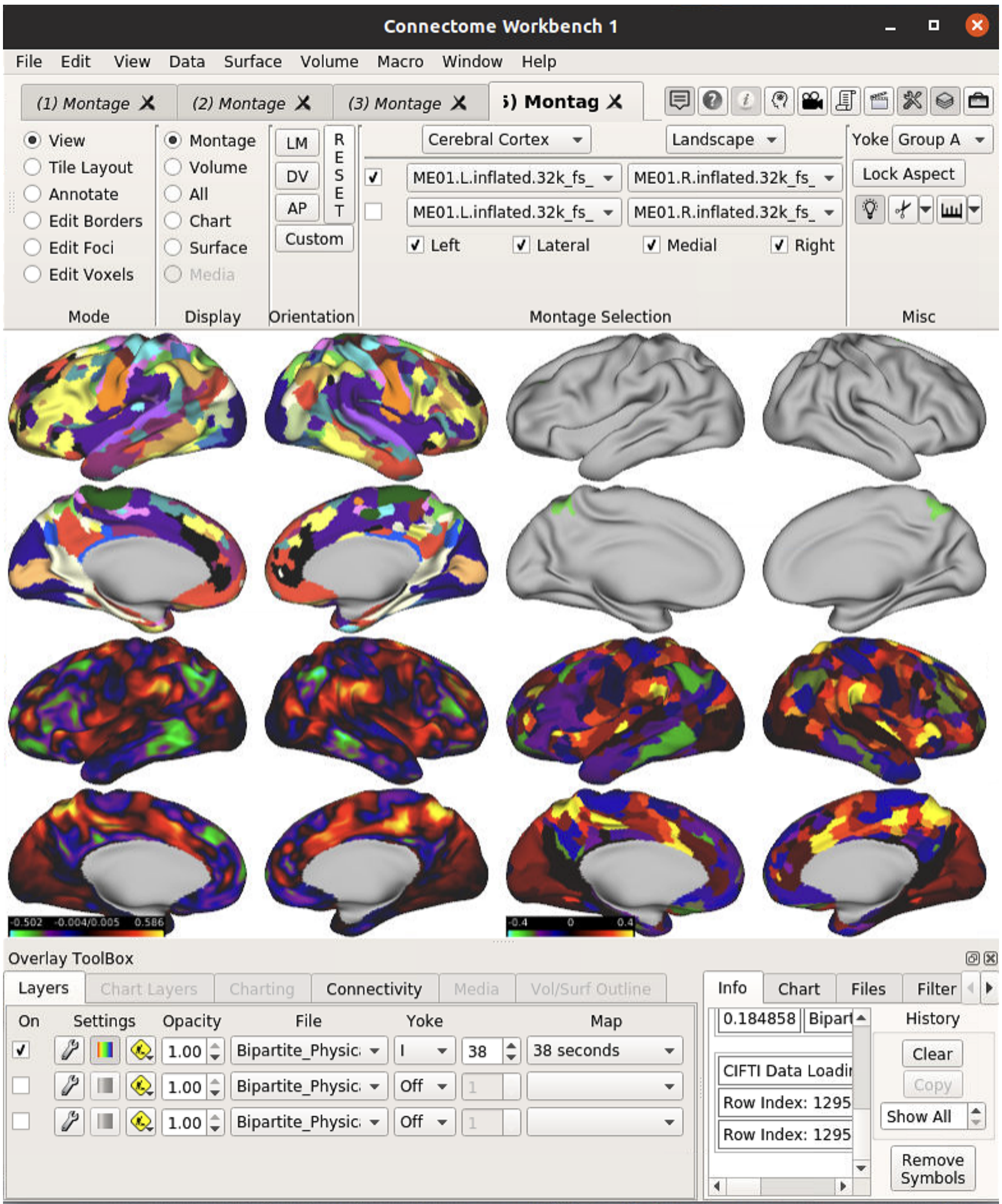
Another example is shown below. In this case, community 19 (highlighted below), was labeled *Cingulo-opercular / Action-mode.*



Community	Network	Network_ManualDecision	R	G	B	FC_Similarity	Spatial_Score
1	Default_Dorsolateral		0.839215686	0.619607843	0.254901961	0.741194874	0.315544575
2	Visual_Lateral		0.105882353	0.007843137	0.57254902	0.861742951	0.460733496
3	Default_Dorsolateral		0.839215686	0.619607843	0.254901961	0.824409303	0.33047391
4	Frontoparietal		0.996078431	1	0.329411765	0.808506474	0.474387302
5	Visual_V1		1	0.709803922	0.388235294	0.842521205	0.8157277
6	Default_Retrosplenial		1	1	0.854901961	0.768957052	0.549095726
7	Somatomotor_Hand		0.439215686	1	0.996078431	0.903785228	0.334304207
8	Default_Anterolateral		0.560784314	0.105882353	0.419607843	0.674348555	0.280761998
9	Somatomotor_Foot		0.043137255	0.329411765	0.011764706	0.908714521	0.732879819
10	Somatomotor_Face		1	0.501960784	0	0.911512329	0.706101477
11	Visual_Lateral		0.105882353	0.007843137	0.57254902	0.832729355	0.702574756
12	CinguloOpercular/Action-mode		0.274509804	0.02745098	0.576470588	0.831340674	0.697625179
13	CinguloOpercular/Action-mode		0.274509804	0.02745098	0.576470588	0.849826686	0.725513784
14	Salience		0	0	0	0.585783462	0.302410131
15	Frontoparietal		0.996078431	1	0.329411765	0.778770019	0.465902266
16	Default_Parietal		0.917647059	0.2	0.137254902	0.708382861	0.346835238
17	Auditory		0.666666667	0.329411765	0.97254902	0.880931112	0.569699136
18	Somatomotor_Hand		0.439215686	1	0.996078431	0.911957634	0.611176857
19	CinguloOpercular/Action-mode		0.274509804	0.02745098	0.576470588	0.767146814	0.660395062
20	Language		0.250980392	0.6	0.6	0.56898731	0.332826187

Community	Confidence	Alt_1_Network	Alt_1_FC_Similarity	Alt_1_Spatial_Score	Alt_2_Network	Alt_2_FC_Similarity	Alt_2_Spatial_Score
1	0.59138351	Frontoparietal	0.627108023	0.234355912	Default_Anterolateral	0.350683717	0.138700158
2	0.480725045	Visual_Dorsal/VentralStream	0.752496129	0.356327085	Visual_V5	0.56645345	0.042205922
3	0.586898797	Frontoparietal	0.667543126	0.257188475	Default_Parietal	0.391039042	0.161666304
4	5.61651861	Default_Dorsolateral	0.451976867	0.128253968	DorsalAttention	0.317082841	0.105752381
5	6.237666855	Visual_Lateral	0.734929168	0.12920579	Default_Retrosplenial	0.083778525	0.00312989
6	7.768371678	Visual_V1	0.254595786	0.18913852	Default_Parietal	0.579871248	0.040501299
7	0.15268445	Premotor/DorsalAttentionII	0.796442649	0.329110803	CinguloOpercular/Action-mode	0.372021893	0.145292033
8	0.148461135	Language	0.560442939	0.294154198	Default_Parietal	0.597210426	0.182536179
9	13.69738442	SomatoCognitiveAction	0.767282422	0.059056122	CinguloOpercular/Action-mode	0.395208713	0.10564059
10	8.001444999	SomatoCognitiveAction	0.890805526	0.080266538	CinguloOpercular/Action-mode	0.581643465	0.076316635
11	3.185805458	Visual_V1	0.744006396	0.187862764	Visual_Dorsal/VentralStream	0.683994883	0.03083412
12	12.39999669	Auditory	0.657300764	0.065846447	Premotor/DorsalAttentionII	0.608059432	0.045150215
13	15.78884704	SomatoCognitiveAction	0.768785458	0.047769424	Somatomotor_Face	0.592046739	0.059131161
14	1.508982275	Default_Parietal	0.491794554	0.143566176	Default_Dorsolateral	0.588281345	0.116094771
15	3.277669574	Default_Dorsolateral	0.617928835	0.137264537	MedialParietal	0.349613949	0.04024024
16	0.424018065	Default_Anterolateral	0.702237192	0.24569252	Language	0.378996237	0.08321582
17	5.073237907	Language	0.392928794	0.210306822	CinguloOpercular/Action-mode	0.674951339	0.113208222
18	2.835696616	Premotor/DorsalAttentionII	0.796363556	0.182467675	SomatoCognitiveAction	0.743561062	0.046657901
19	13.84839844	Salience	0.203871326	0.167358025	DorsalAttention	0.301488635	0.010592593
20	1.24870953	Default_Anterolateral	0.402101221	0.209435992	Default_Parietal	0.276429466	0.131396957

Note that this approach works well when functional networks are situated approximately in their “typical” locations. However, communities may be “incorrectly” labeled if they have a low Spatial_Score (i.e., minimal overlap with the typical network location). These cases tend to be accompanied by low Confidence scores. Users can sort the XLS sheet by the Confidence values (as done below) and visually examine low confidence assignments. We have found that a good rule of thumb is to focus on Confidence values of 0.33 or less (highlighted below), but it is useful to carefully review all of the assignments, if time permits.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Community	Network	Network_ManualDecision	R	G	B	FC_Similarity	Spatial_Score	Confidence	Alt_1_Network	Alt_1_FC_Similarity	Alt_1_Spatial_Score	Alt_2_Network	Alt_2_FC_Similarity	Alt_2_Spatial_Score
2	36	CinguloOpercular/Action-mode		0.274509804	0.02745098	0.576470588	0.800996366	0.412387387	0.01345979	Auditory	0.861795446	0.378203203	SomatoCognitiveAction	0.883511145	0.048798799
3	50	Visual_Dorsal/VentralStream		0.184313725	0.458823529	0.709803922	0.737521487	0.196483352	0.02760549	DorsalAttention	0.527263468	0.267452301	CinguloOpercular/Action-mode	0.614066292	0.107706697
4	72	Visual_V5		0.725490196	0.811764706	0.564705882	0.756270312	0.13768614	0.047348585	Visual_Dorsal/VentralStream	0.48742482	0.203970981	DorsalAttention	0.232888137	0.142038946
5	71	Salience		0	0	0	0.253622798	0.227817746	0.114468277	CinguloOpercular/Action-mode	0.170769567	0.303597122	Default_Anterolateral	0.340764685	0.122781775
6	8	Default_Anterolateral		0.560784314	0.105882353	0.419607843	0.674348555	0.280761998	0.148461135	Language	0.560442939	0.294154198	Default_Parietal	0.597210426	0.182536179
7	7	Somatomotor_Hand		0.439215686	1	0.996078431	0.903785228	0.334304207	0.15268445	Premotor/DorsalAttentionII	0.796442649	0.329110803	CinguloOpercular/Action-mode	0.372021893	0.145292033
8	62	DorsalAttention		0.388235294	0.839215686	0.247058824	0.747838317	0.41556971	0.213532077	Visual_Dorsal/VentralStream	0.65828925	0.389030432	Frontoparietal	0.184934476	0.077565464
9	40	Premotor/DorsalAttentionII		1	0.509803922	1	0.891416148	0.255105105	0.246582323	CinguloOpercular/Action-mode	0.662089718	0.275525526	Somatomotor_Foot	0.86877648	0.159647147
10	56	Somatomotor_Face		1	0.501960784	0	0.78975781	0.391477273	0.250308709	SomatoCognitiveAction	0.893852855	0.276641414	CinguloOpercular/Action-mode	0.785623011	0.190340909
11	38	DorsalAttention	CinguloOpercular/Action-mo	0.388235294	0.839215686	0.247058824	0.414434646	0.405845691	0.321867499	CinguloOpercular/Action-mode	0.636207924	0.2	Salience	0.205082247	0.050204169
12	42	Frontoparietal		0.996078431	1	0.329411765	0.406638408	0.281541704	0.362564157	Default_Anterolateral	0.463210167	0.181391147	Language	0.276124483	0.197259861
13	61	Default_Parietal		0.917647059	0.2	0.137254902	0.547932208	0.344413321	0.397468608	Default_Retrosplenial	0.543172004	0.248615002	Default_Dorsolateral	0.760979712	0.110364146
14	16	Default_Parietal		0.917647059	0.2	0.137254902	0.708382861	0.346835238	0.424018065	Default_Anterolateral	0.702237192	0.24569252	Language	0.378996237	0.08321582
15	35	Default_Anterolateral		0.560784314	0.105882353	0.419607843	0.819573224	0.334450172	0.463820417	Default_Parietal	0.788390874	0.237514318	Default_Dorsolateral	0.532250342	0.136597938
16	34	SomatoCognitiveAction		0.470588235	0.078431373	0.078431373	0.761364419	0.356140351	0.468354841	CinguloOpercular/Action-mode	0.754750039	0.244669366	Somatomotor_Hand	0.51097305	0.176608187
17	2	Visual_Lateral		0.105882353	0.007843137	0.57254902	0.861742951	0.460733496	0.480725045	Visual_Dorsal/VentralStream	0.752496129	0.356327085	Visual_V5	0.56645345	0.042205922
18	68	Visual_Dorsal/VentralStream		0.184313725	0.458823529	0.709803922	0.735867933	0.369124877	0.483115741	DorsalAttention	0.421402233	0.434611603	Premotor/DorsalAttentionII	0.740768217	0.082989184
19	64	Default_Anterolateral		0.560784314	0.105882353	0.419607843	0.780199687	0.302189781	0.490666923	Default_Dorsolateral	0.689342508	0.229440389	Default_Parietal	0.694147953	0.104298459
20	46	Default_Retrosplenial		1	1	0.854901961	0.443107426	0.203564467	0.567605918	Salience	0.620877888	0.092676135	MedialParietal	0.639607881	0.054525202
21	3	Default_Dorsolateral		0.839215686	0.619607843	0.254901961	0.824409303	0.33047391	0.586898797	Frontoparietal	0.667543126	0.257188475	Default_Parietal	0.391039042	0.161666304
22	33	Premotor/DorsalAttentionII		1	0.509803922	1	0.787971079	0.304456019	0.591147149	CinguloOpercular/Action-mode	0.804248484	0.187471065	DorsalAttention	0.469847815	0.288686343
23	1	Default_Dorsolateral		0.839215686	0.619607843	0.254901961	0.741194874	0.315544575	0.59138351	Frontoparietal	0.627108023	0.234355912	Default_Anterolateral	0.350683717	0.138700158
24	67	Premotor/DorsalAttentionII		1	0.509803922	1	0.699088694	0.414481898	0.686011367	DorsalAttention	0.75720938	0.226966292	CinguloOpercular/Action-mode	0.629576781	0.116604245
25	77	Default_Parietal		0.917647059	0.2	0.137254902	0.520860531	0.245446951	0.687846467	Default_Dorsolateral	0.778910076	0.097243108	Default_Retrosplenial	0.53513245	0.110944027
26	37	Default_Retrosplenial		1	1	0.854901961	0.439606748	0.226084373	0.706870047	DorsalAttention	0.39555319	0.147207368	MedialParietal	0.395659438	0.082174688
27	73	Premotor/DorsalAttentionII		1	0.509803922	1	0.902654975	0.427390181	0.736026187	Somatomotor_Face	0.816977587	0.272006891	Somatomotor_Hand	0.847987885	0.0625323
28	25	SomatoCognitiveAction		0.470588235	0.078431373	0.078431373	0.925296613	0.451342877	0.853852876	Somatomotor_Hand	0.855628349	0.263285617	Somatomotor_Foot	0.877094206	0.147511929
29	21	CinguloOpercular/Action-mode		0.274509804	0.02745098	0.576470588	0.776585229	0.44089277	1.059443659	Premotor/DorsalAttentionII	0.82613936	0.201242115	Somatomotor_Foot	0.797541585	0.153944687
30	60	Language		0.250980392	0.6	0.6	0.532571113	0.345489443	1.153980646	Default_Anterolateral	0.363668556	0.234890168	Default_Parietal	0.102909178	0.104371934
31	28	DorsalAttention		0.388235294	0.839215686	0.247058824	0.676656651	0.411414693	1.209341165	Frontoparietal	0.621809987	0.202641166	MedialParietal	0.261877622	0.04896782
32	20	Language		0.250980392	0.6	0.6	0.56898731	0.332826187	1.24870953	Default_Anterolateral	0.402101221	0.209435992	Default_Parietal	0.276429466	0.131396957
33	51	Premotor/DorsalAttentionII		1	0.509803922	1	0.897222414	0.420382395	1.318150177	DorsalAttention	0.559648234	0.290728716	Visual_Dorsal/VentralStream	0.437699331	0.087301587
34	58	Default_Parietal		0.917647059	0.2	0.137254902	0.74023567	0.390598291	1.34893986	Default_Anterolateral	0.682926669	0.180241327	Default_Dorsolateral	0.639964773	0.174409251
35	14	Salience		0	0	0	0.585783462	0.302410131	1.508982275	Default_Parietal	0.491794554	0.143566176	Default_Dorsolateral	0.588281345	0.116094771
36	74	Frontoparietal		0.996078431	1	0.329411765	0.598230482	0.294609461	1.522512723	DorsalAttention	0.365844074	0.190979098	Default_Dorsolateral	0.277173183	0.022662266
37	48	Default_Parietal		0.917647059	0.2	0.137254902	0.754837279	0.437003058	1.602059603	Default_Retrosplenial	0.693406969	0.182823649	Default_Dorsolateral	0.674080386	0.151172273
38	69	Frontoparietal		0.996078431	1	0.329411765	0.734621948	0.399435028	1.666751792	Salience	0.470625276	0.233804143	Default_Anterolateral	0.139395554	0.139642185
39	59	CinguloOpercular/Action-mode		0.274509804	0.02745098	0.576470588	0.826305791	0.548	1.752725984	Premotor/DorsalAttentionII	0.822079774	0.200098765	Somatomotor_Face	0.669336388	0.105135802
40	23	CinguloOpercular/Action-mode		0.274509804	0.02745098	0.576470588	0.650017852	0.429473684	1.890895128	DorsalAttention	0.649549278	0.148667966	Frontoparietal	0.217351764	0.178401559
41	49	Salience		0	0	0	0.645687453	0.249506903	1.905461582	Default_Anterolateral	0.388650523	0.142669297	Default_Parietal	0.362257685	0.118737673

To change a network assignment, copy the alternative assignment name into column C ("Network_ManualDecision"). For example, community 38 (shown above) was originally labeled *Dorsal Attention* but was changed to *Cingulo-opercular / Action-mode*. Note how Tabs 3-4 indicate strong FC with brain regions already established as belonging to *Cingulo-opercular / Action-mode*.

Once all of the assignments have been inspected and the XLS file containing the manual decisions has been saved, you can run `pfm_parse_manual_decisions` to incorporate the manual labels.

In the Matlab editor window, run lines 136-141 to define the inputs and run `pfm_parse_manual_decisions`.

```
134     %% Step 6: Review algorithmic network assignments, optionally adjust labels manually if needed.
135
136     % define inputs
137 -   XLS = [PfmDir '/Bipartite_PhysicalCommunities+AlgorithmicLabeling_NetworkLabels.xls']; % note that the XLS
138 -   Output = 'Bipartite_PhysicalCommunities+FinalLabeling';
139
140     % OPTIONAL: update network assignments according to manual decisions;
141 -   pfm_parse_manual_decisions(Ic,Column,MidthickSurfs,Priors,XLS,Output,PfmDir,WorkbenchBinary);
142
```

Step 7: Calculate size of each functional brain network.

The main output of `pfm_parse_manual_decisions` is a CIFTI file (called `"Bipartite_PhysicalCommunities+FinalLabeling.dlabel.nii")`. The `pfm_calculate_network_size` function can be used to calculate the size of each functional brain network (the percentage of cortical surface area it occupies).

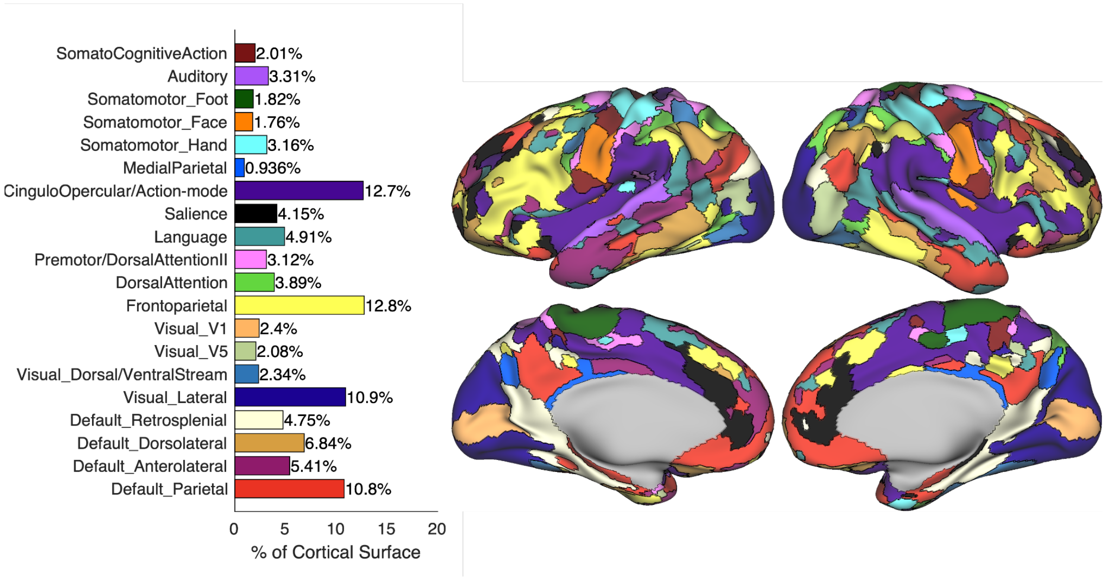
In the Matlab editor window, run lines 145-151 in the `pfm_tutorial.m` script to define the inputs and run the `pfm_calculate_network_size` function.

```
%% Step 7: Calculate size of each functional brain network

% define inputs
FunctionalNetworks = ft_read_cifti_mod([PfmDir '/Bipartite_PhysicalCommunities+FinalLabeling.dlabel.nii']);
VA = ft_read_cifti_mod([Subdir '/fs_LR/fsaverage_LR32k/' Subject '.midthickness_va.32k_fs_LR.dscalar.nii']);
Structures = {'CORTEX_LEFT','CORTEX_RIGHT'}; % in this case, cortex only.

% calculate the size of each functional brain network
NetworkSize = pfm_calculate_network_size(FunctionalNetworks,VA,Structures);
```

`FunctionalNetworks` is a CIFTI file containing the “final” (post manual review) functional network maps.
`VA` is a CIFTI file containing the surface area (in mm^2) each vertex is responsible for.
`Structures` is a cell array of brain structures. The network size calculation will be constrained to the structures in this variable.

In the Matlab editor window, run lines 153-175 to run the `pfm_calculate_network_size` function and visualize the results.

```
143 %% Step 7: Calculate size of each functional brain network
144
145 % define inputs
146 FunctionalNetworks = ft_read_cifti_mod([PfmDir '/Bipartite_PhysicalCommunities+FinalLabeling.dlabel.nii']);
147 VA = ft_read_cifti_mod([Subdir '/fs_LR/fsaverage_LR32k/' Subject '.midthickness_va.32k_fs_LR.dscalar.nii']);
148 Structures = {'CORTEX_LEFT','CORTEX_RIGHT'}; % in this case, cortex only.
149
150 % calculate the size of each functional brain network
151 NetworkSize = pfm_calculate_network_size(FunctionalNetworks,VA,Structures);
152
153 close all; % blank slate
154 H = figure; % prellocate parent figure
155 set(H,'position',[1 1 325 400]); hold;
156
157 % unique functional networks;
158 uCi = unique(nonzeros(FunctionalNetworks.data));
159
160 % sweep through
161 % the networks;
162 for i = 1:length(uCi)
163     Tmp = nan(1,length(Priors.NetworkLabels));
164     Tmp(i) = NetworkSize(i);
165     barh(Tmp,'FaceColor',Priors.NetworkColors(i,:));
166     text((NetworkSize(i)+0.1),i,[num2str(NetworkSize(i),3) '%']);
167 end
168
169 % make it pretty;
170 yticklabels(Priors.NetworkLabels);
171 yticks(1:length(uCi)); ylim([0 21]);
172 xlim([0 20]); xticks(0:5:20);
173 set(gca,'fontname','arial','fontsize',10,'TickLength',[0 0],'TickLabelInterpreter','none');
174 xlabel('% of Cortical Surface');
175 print(gcf,[PfmDir '/FunctionalNetworkSizes'],'-dpdf');
176
```



References

1. Lynch, C. J. *et al.* Rapid Precision Functional Mapping of Individuals Using Multi-Echo fMRI. *Cell Rep.* **33**, 108540 (2020).
2. Buckner, R. L., Krienen, F. M., Castellanos, A., Diaz, J. C. & Yeo, B. T. T. The organization of the human cerebellum estimated by intrinsic functional connectivity. *J. Neurophysiol.* **106**, 2322–2345 (2011).
3. Marek, S. *et al.* Spatial and Temporal Organization of the Individual Human Cerebellum. *Neuron* **100**, 977–993.e7 (2018).
4. Gordon, E. M. *et al.* Default-mode network streams for coupling to language and control systems. *Proc. Natl. Acad. Sci. U. S. A.* **117**, 17308–17319 (2020).
5. Gordon, E. M. *et al.* Precision Functional Mapping of Individual Human Brains. *Neuron* **95**, 791–807.e7 (2017).