

# Individualized functional topographic mapping with the BayesBrainMap R package and population-derived priors

Nohelia Da Silva Sanchez, Damon D. Pham, Ellyn Butler, Amanda F. Mejia

2025-05-07

## 0.1 1. Introduction

Bayesian brain mapping (BBM) is a technique for producing individualized functional brain topographic maps from existing group-level network maps or *templates*. Those templates take the form of either a group parcellation, a set of group ICA maps, or other types of network maps like PROFUMO or NNF. BBM is a generalization of template ICA (Mejia et al. 2020), in which the template is a set of group ICA maps. In BBM, the template is not used directly in the individual-level model, but rather forms is the basis of *population-derived priors*, which are used in a Bayesian model. The priors are derived from a training set of subjects from a representative population, either using holdout data from the focal study or using data from a publicly available neuroimaging repository such as the Human Connectome Project (HCP) (CITE) or Alzheimer’s Disease Neuroimaging Initiative (ADNI) (CITE).

The Bayesian model in BBM is a single-subject “hierarchical” source separation model, with parameters representing the spatial topography of different networks and their corresponding temporal activation profiles. The model is hierarchical in the sense that it includes population-derived priors on those parameters. While hierarchical models typically require multi-subject data to establish the shared prior parameters, in BBM the model is fit to a single subject using priors that are already established. This allows BBM to be highly pragmatic, computationally efficient, and potentially clinically applicable.

The use of population-derived priors in BBM reduces noise while retaining relevant signal. This results in more reliable individual-level network topography maps and the functional connectivity between them (Mejia et al. 2020) (also CITE Mejia 2022 and 2025). An important feature of the BBM priors on spatial topography is that they spatially vary within each network, which allows individual differences to be expressed where they exist while reducing noise in background regions. Furthermore, due to the continuous, overlapping nature of BBM network maps, noise reduction in background regions of one network indirectly contributes to estimation of signal in the same region in other networks. The powerful noise-reduction properties of BBM allows it to produce reliable maps of individual functional topography with moderate scan duration, without the need for “dense sampling” of individuals.

There are two steps to performing BBM, which are implemented in the `BayesBrainMap` R package: (1) *prior estimation* and (2) *model fitting* (Figure 1). Here, our primary focus is creating and sharing population-derived priors for using data from the HCP, based on various templates. These priors can be directly used to perform BBM in studies of healthy young adults. To perform BBM to study individuals from other populations, or using other templates, we provide and describe the code used to produce the HCP-derived priors, so that this workflow can be easily reproduced using other datasets. We also provide guidance for visual investigation of the priors to ensure quality.

Finally, we illustrate the model fitting step using an individual from the HCP and individuals from an external study, the midnight scan club (MSC) (CITE). The MSC focuses on a similar population as the HCP but differs in acquisition methods. Since the population-derived priors used in BBM encode the distribution of signal, not noise, priors derived from one study can be applied to other studies with differing noise properties, as long as the populations are similar. We will illustrate this through application of HCP-derived priors to

data from the MSC. Application to the MSC will also exemplify the ability of BBM to reveal individual features of functional topography from a single session of data, without the need for dense sampling.

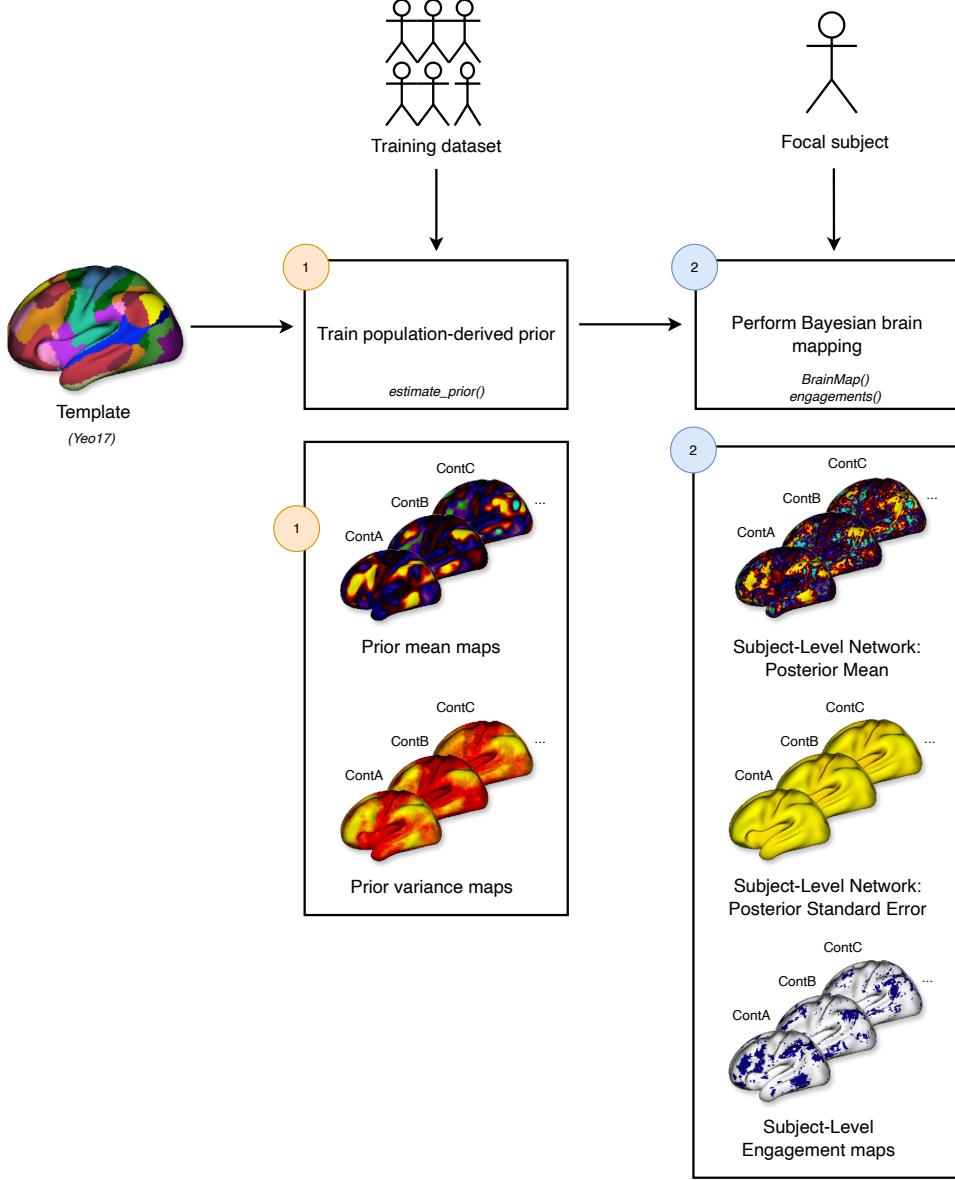


Figure 1: Overview of the Bayesian Brain Mapping Workflow

## 0.2 2. Methods

### 0.2.1 2.1 The Bayesian Brain Mapping Model

Here we provide a brief overview of the BBM statistical model. For a given subject and fMRI session, let  $y_{tv}$  be the preprocessed BOLD fMRI time series at voxel or vertex  $v$  and time point  $t$ . The BBM model assumes that the BOLD time series can be decomposed into contributions from a set of networks, similar to ICA. The first level of the model is given by

$$y_{tv} = \sum_{q=1}^Q a_{tq} s_{qv} + e_{tv} = \mathbf{a}_t^\top \mathbf{s}_v + e_{tv}, \quad e_{tv} \sim N(0, \tau_v^2),$$

where  $s_{qv}$  is the spatial engagement of network  $q$  at voxel or vertex  $v$ , and  $a_{tq}$  is the temporal activation of network  $q$  at time point  $t$ . The vectors  $\mathbf{a}_t$  and  $\mathbf{s}_v$  combine those values across all  $Q$  networks.

The second level of the model incorporates the population-derived prior on the spatial topography in  $s_{qv}$ , as described in [CITE Mejia 2020]:

$$s_{qv} = s_{qv}^0 + \delta_{qv}, \quad \delta_{qv} \sim N(0, \sigma_{qv}^2),$$

where  $s_{qv}^0$  and  $\sigma_{qv}^2$  are known via the prior, and the deviation terms  $\delta_{qv}$  represent individual differences between the subject and the population average. Optionally, spatial dependencies in  $\delta_{qv}$  can be modeled via a spatial prior for additional accuracy and power, though at a higher computational cost (CITE Mejia 2022). Note that structured noise may exist in the data that is not well-captured by the residual noise term. While in ICA it is common to model structured noise as components, in BBM these are not included in the model because can be removed beforehand as described in [CITE Mejia 2020] or via other methods like ICA-FIX.

The third level of the mdoel incorporates the population-derived prior on the functional connectivity between networks, which is represented by the covariance of  $\mathbf{a}_t$ . This is accomplished by assuming a multivariate Normal prior on  $\mathbf{a}_t$  with mean zero and covariance  $\mathbf{G}$ , and assuming a population-derived hyperprior on  $\mathbf{G}$ :

$$\mathbf{a}_t \sim N(\mathbf{0}, \mathbf{G}), \text{ where } \mathbf{G} \sim p(\mathbf{G}),$$

See [CITE Mejia 2025] for details on the population-derived prior on the functional connectivity. Briefly, two choices exist for this prior: the conjugate Inverse-Wishart distribution, or a novel Cholesky-based prior developed for this model, which involves sampling. The former is faster, while the latter has somewhat better performance but is more computationally intensive.

### 0.2.2 2.2 Building HCP-Derived Priors

Here, we build HCP-derived priors for Bayesian brain mapping using several different choices of template, including two different parcellations and group ICA maps (Table 1). Specifically, we use the 17-network Yeo parcellation (CITE), the MSC group parcellation (CITE and get the real name), and the HCP group ICA maps with 15, 25, and 50 components. For each template, we build priors with and without global signal regression (GSR), since whether or not to perform GSR is an important choice that remains the subject of debate.

Global Signal Regression	Group ICA			Parcellation Yeo 17
	15 HCP ICs	25 HCP ICs	50 HCP ICs	
With GSR	✓	✓	✓	✓
Without GSR	✓	✓	✓	✓

Table 1: Templates used for Bayesian brain mapping.

Since the functional MRI data in the HCP was acquired with two different phase-encoding directions (left-to-right, LR, and right-to-left, RL), we build the HCP-derived priors using both phase encoding directions separately, as well as a combined version. Comparing the LR and RL priors allows us to assess the impact of phase encoding direction on the priors, while the combined version provides a general purpose prior that is not specific to either the LR or RL acquisition.

Before estimating the BBM priors, we first select a high-quality, balanced subject sample to ensure high-quality and representative priors. Starting from the full HCP sample of  $N = 1206$  subjects, we apply several

filters to obtain the final sample for the priors. First, we exclude subjects with insufficient scan duration after motion scrubbing using a framewise displacement (FD) threshold of 0.5 mm and dropping the first 15 frames. Second, we exclude any related subjects. Finally, we balance sex within age groups. See Appendix B for details. After all filters, our final sample contains data from X [Nohelia, how many? Can be approximate if it differs across LR/RL/combined] subjects, which is used to estimate the priors. [Nohelia, we are missing the information about which runs were actually used for the LR, RL and combined priors. For LR I guess it must be REST1\_LR and REST2\_LR, and same for RL, but what about combined?]

### 0.2.3 2.3 Workflow Overview

#### Setup

To reproduce this workflow, first follow the setup process outlined in Appendix A.

## 0.3 3. Choosing Training Subjects

### 1. Filter Subjects by Sufficient fMRI Scan Duration

See Appendix B.1 and script: `1_fd_time_filtering.R`

### 2. Filter Unrelated Subjects

See Appendix B.2 and script: `2_unrelated_filtering.R`

### 3. Balance sex within age groups

See Appendix B.3 and script: `3_balance_age_sex.R`

The resulting subject list (`valid_combined_subjects_balanced.rds`) is used throughout the rest of the analysis.

## 0.4 4. Step 1: Estimate Priors using `estimate_prior()`

In this step, we estimate group-level statistical priors using the `estimate_prior()` function from the `BayesBrainMap` package.

## 0.5 4.1 Subject List and Scan Selection

The encoding parameter is set to `combined`, `LR`, and `RL` to use the final lists of subjects saved in Step 3.3 (`valid_combined_subjects_balanced.rds`, `valid_LR_subjects_balanced.rds`, and `valid_RL_subjects_balanced.rds`). The `combined` list includes individuals who passed motion filtering in both `LR` and `RL` directions for both sessions, were unrelated, and were sex-balanced within age groups. The `LR` and `RL` lists include subjects who met these criteria independently for each direction.

If encoding is `combined`, we include only REST1 sessions from both phase-encoding directions:

- `rfMRI_REST1_LR_Atlas_MSMAll_hp2000_clean.dtseries.nii`
- `rfMRI_REST1_RL_Atlas_MSMAll_hp2000_clean.dtseries.nii`

If encoding is `LR` or `RL`, we use both REST1 and REST2 sessions from the specified direction:

For `LR`:

- `rfMRI_REST1_LR_Atlas_MSMAll_hp2000_clean.dtseries.nii`

- rfMRI\_REST2\_LR\_Atlas\_MSMAll\_hp2000\_clean.dtseries.nii

For RL:

- rfMRI\_REST1\_RL\_Atlas\_MSMAll\_hp2000\_clean.dtseries.nii
- rfMRI\_REST2\_RL\_Atlas\_MSMAll\_hp2000\_clean.dtseries.nii

## 0.6 4.2 Temporal Preprocessing Parameters

To standardize scan duration and improve data quality, we apply both initial volume dropping and temporal truncation using parameters handled directly by the `estimate_prior()` function from the `BayesBrainMap` package.

Specifically:

- `drop_first = 15` removes the first 15 volumes from each scan to eliminate early signal instability and motion artifacts.
- `scrub` defines volumes to exclude after a target duration. In our case, we truncate data to the first 10 minutes (600 seconds), excluding any volumes beyond that point.

See Appendix C for more details.

## 0.7 4.3 Parcellation Choices

We consider two types of group-level parcellations for estimating priors:

- HCP GICA parcellation (`GICA15.dscalar.nii`, etc.), available in the `data_OSF/inputs` folder. These files were downloaded from the HCP website, specifically from the CIFTI Subject-specific ICA Parcellations dataset for 15-, 25-, and 50-dimensions.
- Yeo17 parcellation (CITE). For details on how this parcellation was processed and simplified for use, see Appendix D.

Each of these parcellations was used to estimate priors with and without global signal regression (GSR), resulting in eight total priors saved as `.rds` files. See Table Table 1 for a summary of the parcellations and GSR combinations.

```
# This script estimates and saves functional connectivity priors
# for both spatial topography and connectivity.
# It supports both GICA-based (15/25/50 ICs) and Yeo17 parcellations,
# with or without global signal regression (GSR).
# For priors using the "combined" subject list, it loads REST1-LR and REST1-RL
# scans for each subject,
# drops the first 15 volumes, and truncates each scan to approximately 10 minutes.
# Outputs:
# - Priors `*.rds` file saved in `dir_results`

source("5_estimate_prior.R")
```

## 0.8 4.4 Example Usage

Running `estimate_prior()` on the full "combined" subject list (~350 subjects) takes approximately 27 hours and uses 135 GB of memory.

For an example of how to run `estimate_prior()` and all relevant parameters, see Appendix E.

## 0.9 5. Visualization

In this section, we visualize both the parcellation maps and the priors outputs (mean and variance) for each parcellation scheme used in the study: Yeo17, 15 IC, 25 IC, and 50 IC using the `combined` list of subjects.

We also visualize their corresponding functional connectivity (FC) priors.

### 0.9.1 5.1 Generate and Save Parcellation Visualizations

#### 0.9.1.1 5.1.1 Yeo17 parcellation

Script: `8_visualization_Yeo17parcellations.R`

This script creates one PNG image per parcel (17 in total), where only the selected parcel is colored and all others are white. The parcellation used is Yeo17, created in Appendix D.

Images are saved in `data_0SF/outputs/parcellations_plots/Yeo17`.

#### 0.9.1.2 5.1.2 GICA Parcellations

Script: `9_visualization_GICAparchellations.R`

This script loops over all independent components for each parcellation dimensionality (`nIC = 15, 25, 50`) and generates two images per component:

- A cortical surface map (e.g., `GICA15_IC1.png`)
- A subcortical view (e.g., `GICA15_IC1_sub.png`)

The resulting images are saved in the following folders:

- `data_0SF/outputs/parcellations_plots/GICA15/`
- `data_0SF/outputs/parcellations_plots/GICA25`
- `data_0SF/outputs/parcellations_plots/GICA50`

Each pair of files corresponds to a specific ICA component and captures its spatial map across brain regions.

### 0.9.2 5.2 Visualize Prior Components

Script: `6_visualization_prior.R`

This script loads each estimated prior file from `priors_rds/` and plots both the mean and standard deviation components for all independent components (ICs).

All images are organized into folders by number of ICs, GSR setting, and corresponding list of subjects used, e.g.:

```

data_0SF/priors_plots/GICA15/combined/GSR/
data_0SF/priors_plots/GICA15/combined/noGSR/
data_0SF/priors_plots/GICA25/LR/noGSR/
data_0SF/priors_plots/Yeo17/RL/GSR/
...

```

### 0.9.3 5.3 Visual Summary of Priors

In this section, we present a comparative visual summary of the estimated group-level priors.

For each parcellation type Yeo17, 15 ICs, 25 ICs, and 50 IC, we display:

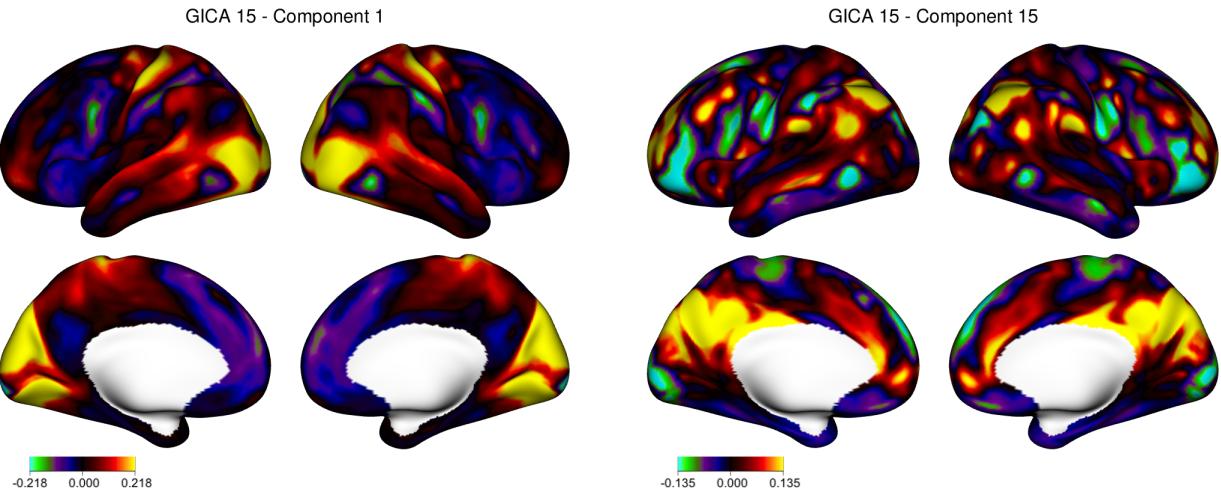
- First and Last Parcellation Map
- First and Last Component Mean
- First and Last Component Standard Deviation

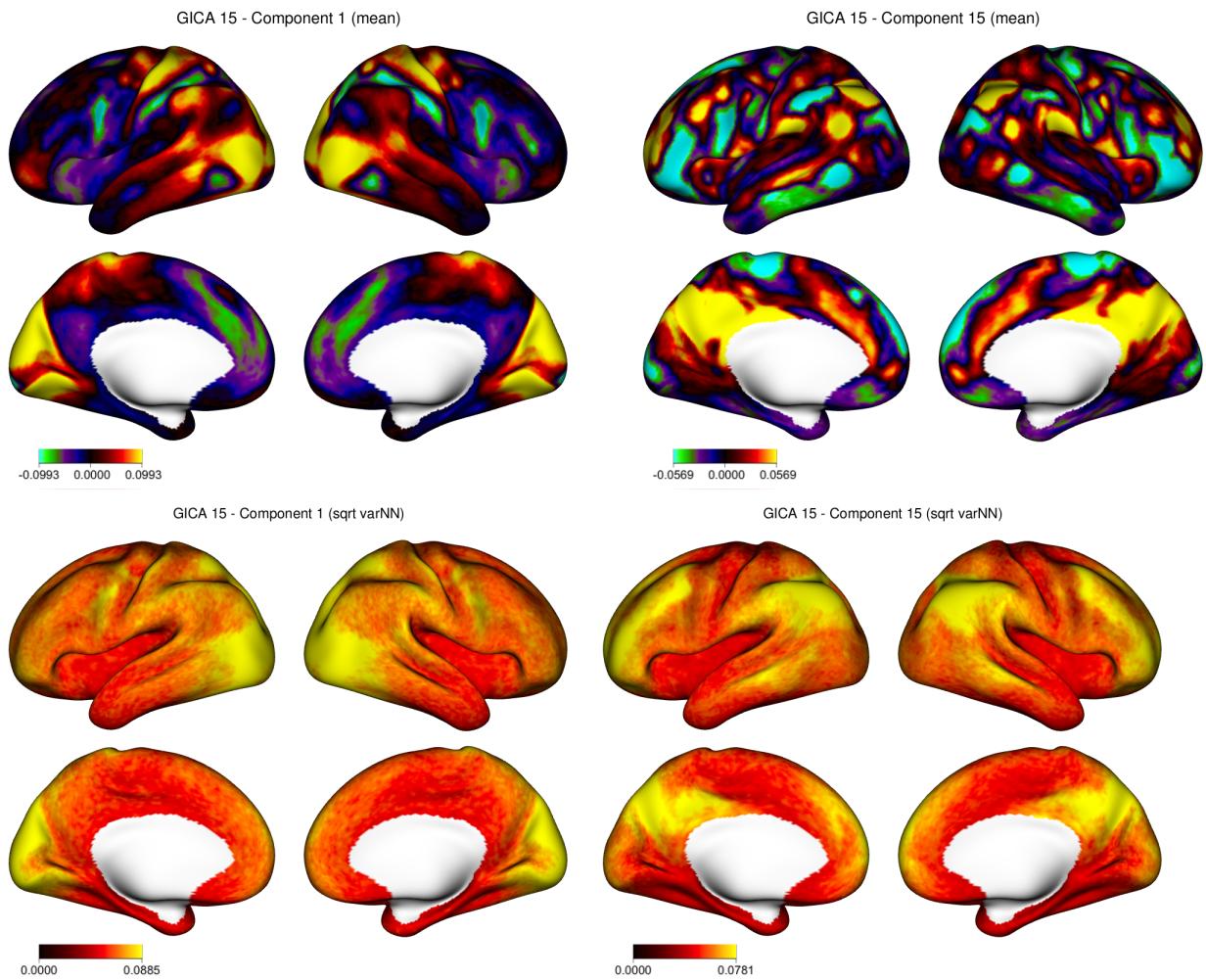
These summaries are shown in a 2-column grid layout per parcellation to highlight spatial structure and variability.

All images were generated using the scripts:

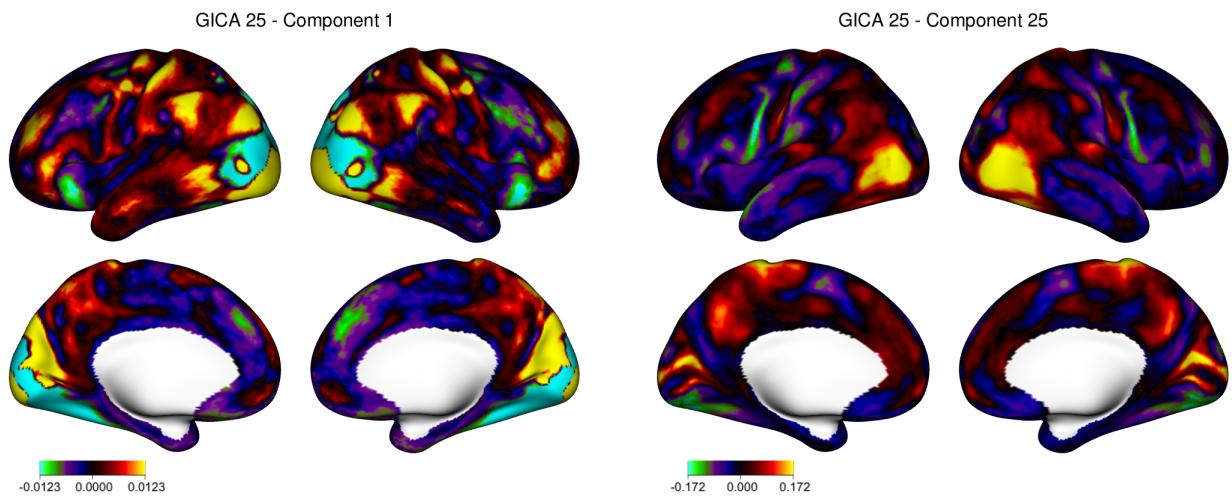
- 8\_visualization\_Yeo17parcellations.R
- 9\_visualization\_GICAparchellations.R
- 6\_visualization\_prior.R

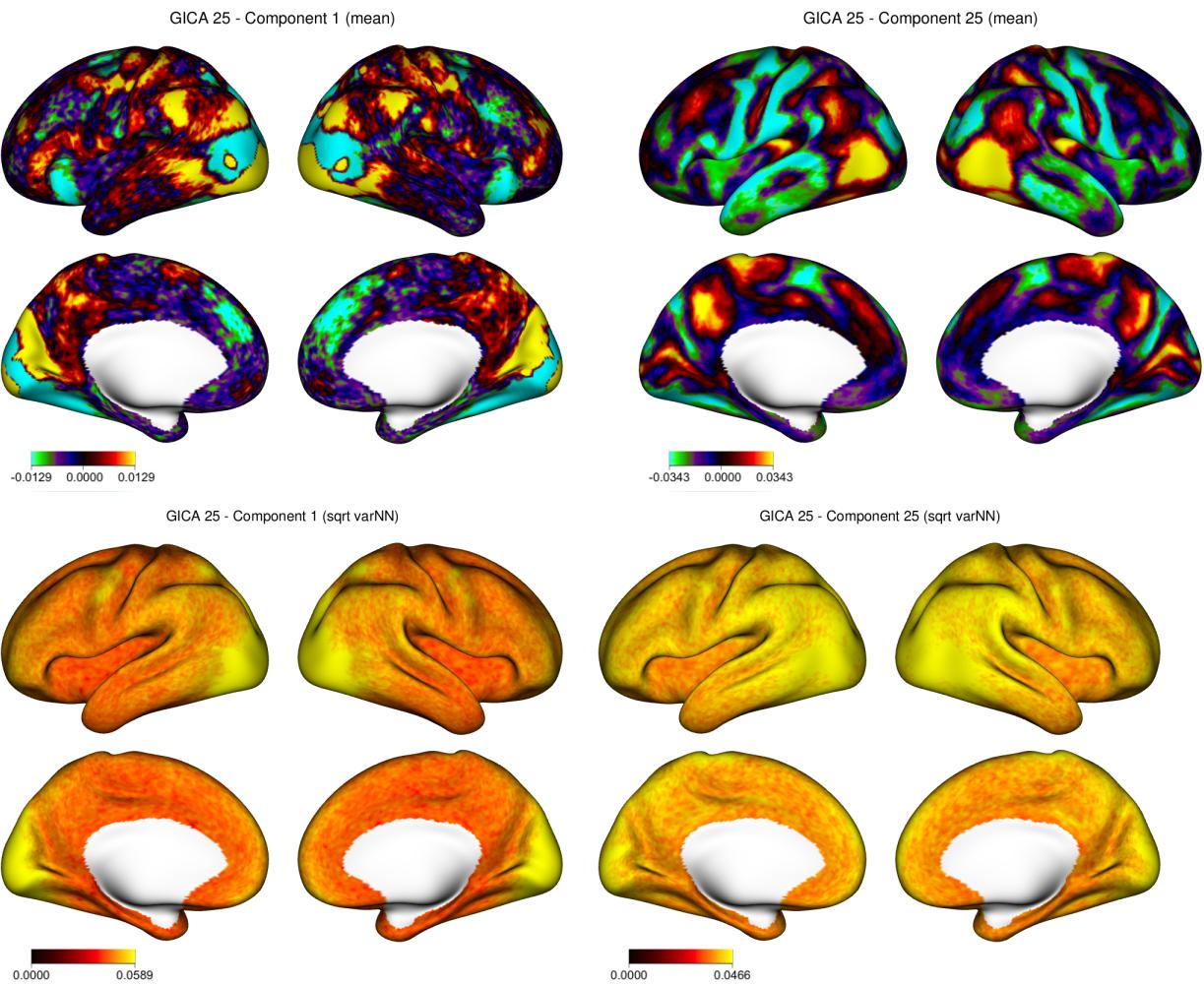
#### 0.9.3.1 5.3.1 15 ICs



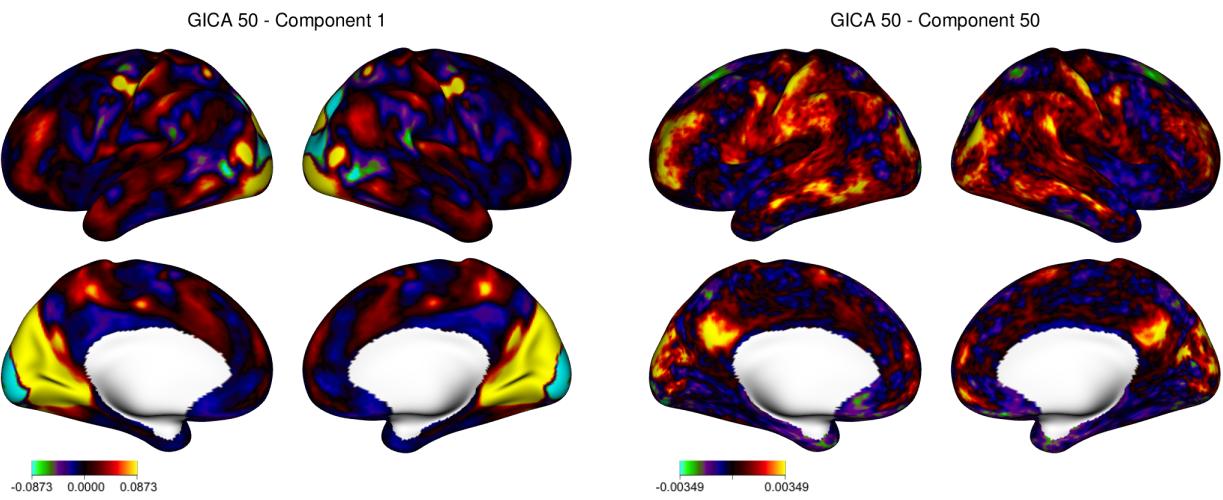


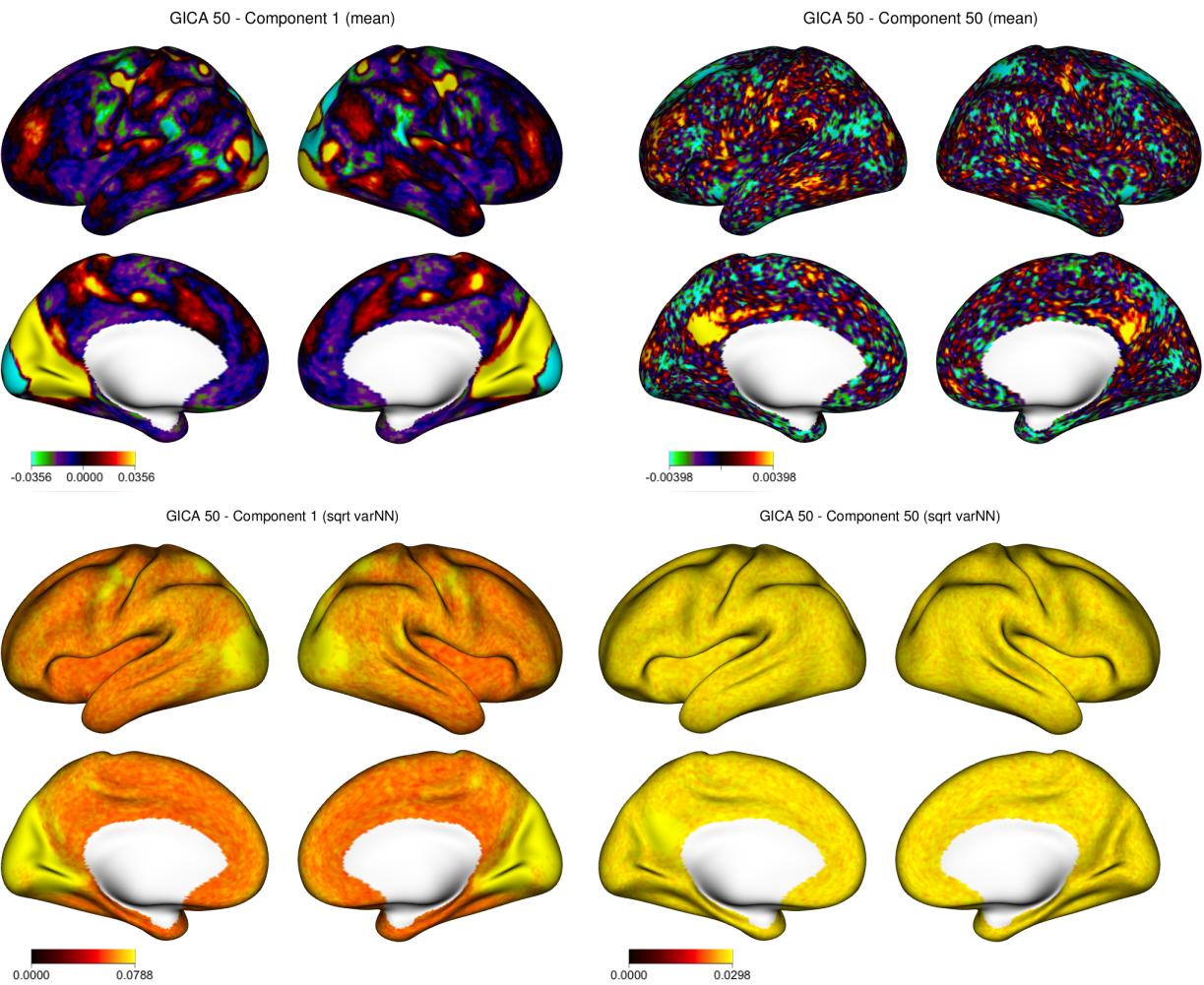
### 0.9.3.2 5.3.2 25 ICs





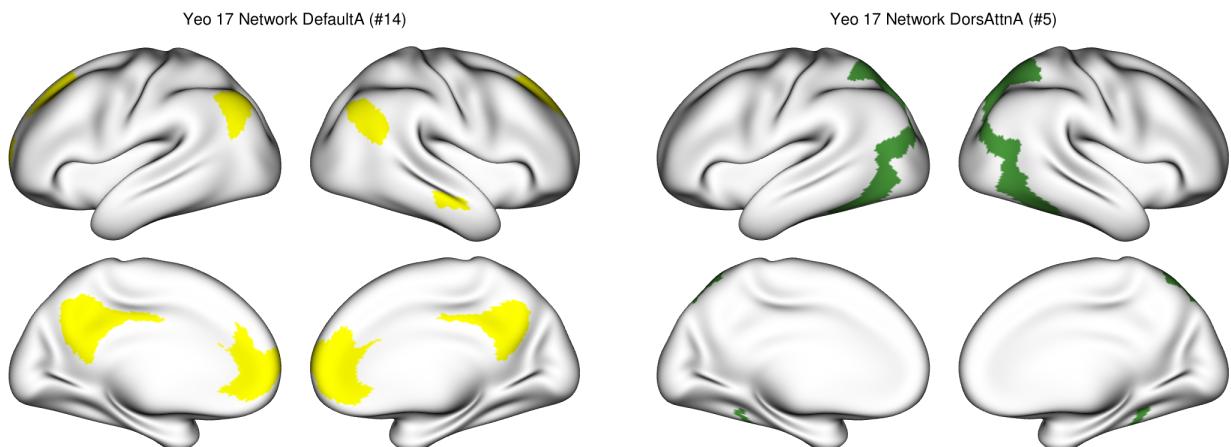
### 0.9.3.3 5.3.3 50 ICs

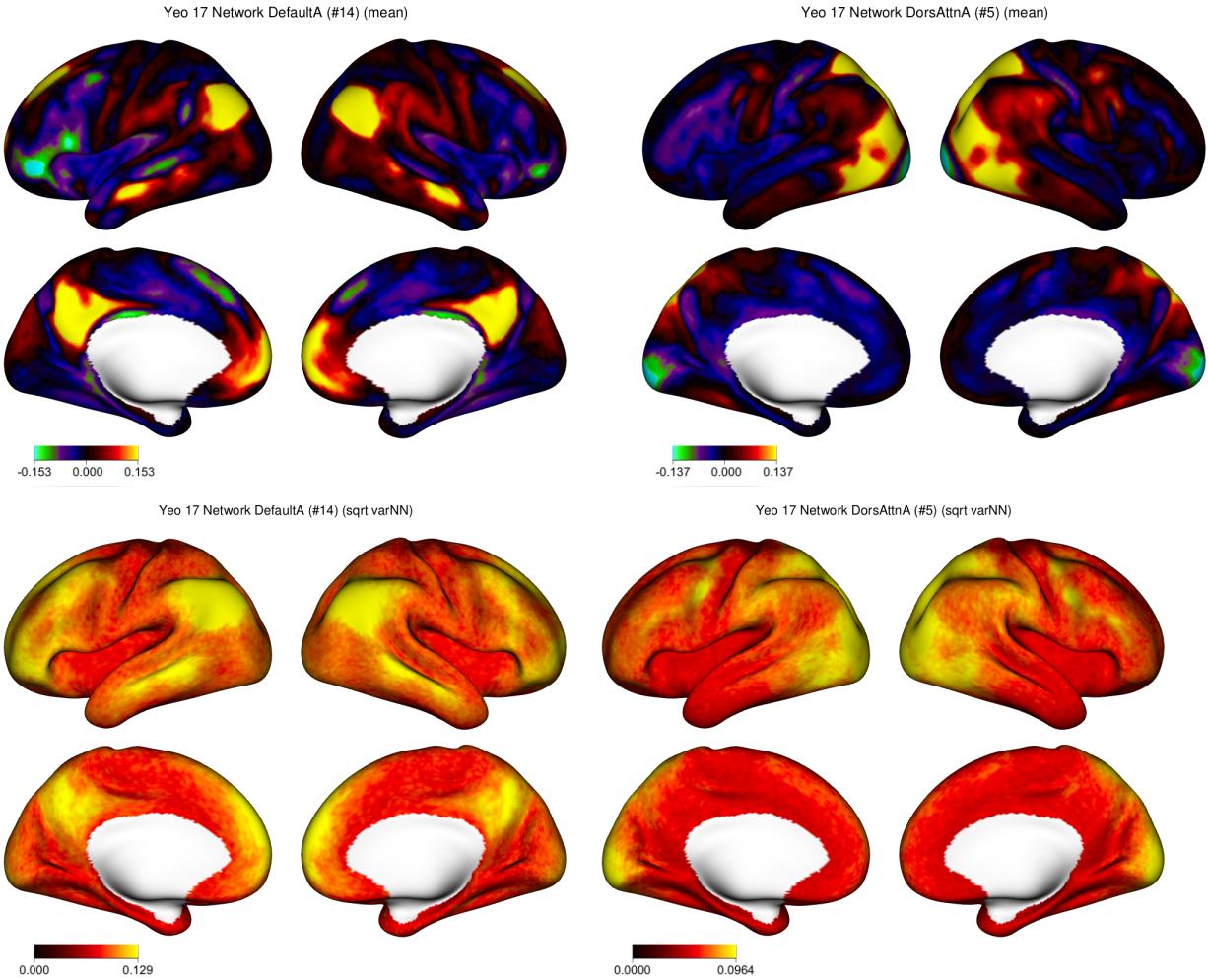




#### 0.9.3.4 5.3.4 Yeo17

For the Yeo17 parcellation, we show visualizations of the two main networks (`DefaultA` and `DorsAttnA`):





#### 0.9.4 5.4 Visualize Functional Connectivity Priors

Script: `7_visualization_FC.R`

This step visualizes the Functional Connectivity (FC) prior for each prior using both the Cholesky and Inverse-Wishart parameterizations. For each group-level prior in `priors/`, we compute and plot:

- Mean FC matrix (off-diagonal values only)
- Standard deviation of FC estimates (from the variance matrix)

For each prior, the following outputs are saved in the corresponding folder under:

`data_OSF/outputs/priors_plots/<parcellation>/<encoding>/FC/`

Where:

- = GICA15, GICA25, GICA50, or Yeo17
- = LR, RL, or combined

#### 0.9.4.1 PDF files (2 per prior)

- [prior\_name]\_FC\_Cholesky.pdf
- [prior\_name]\_FC\_InverseWishart.pdf

Each PDF includes:

- FC Prior Mean (Page 1)
- FC Prior Standard Deviation (Page 2)

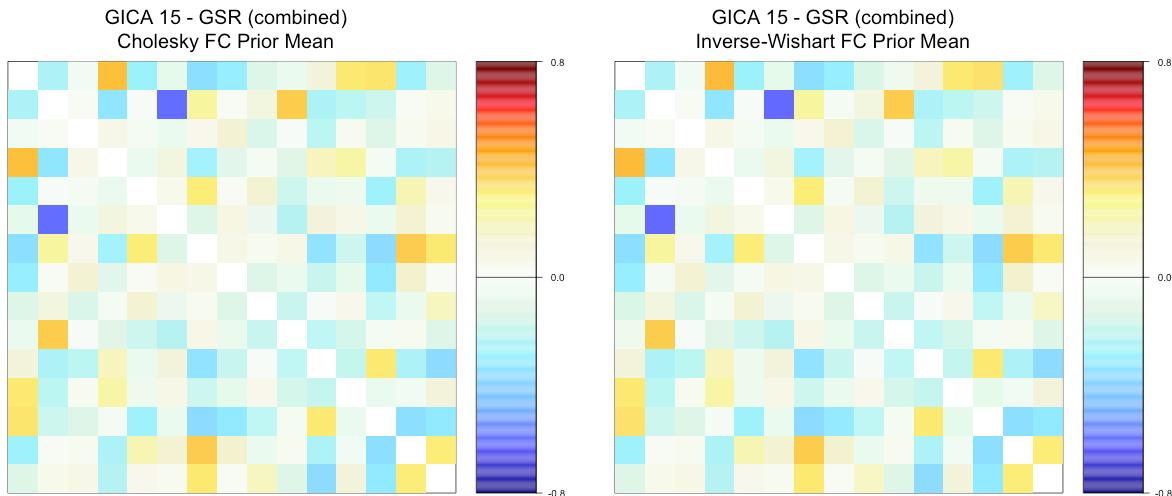
#### 0.9.4.2 PNG images (4 per prior)

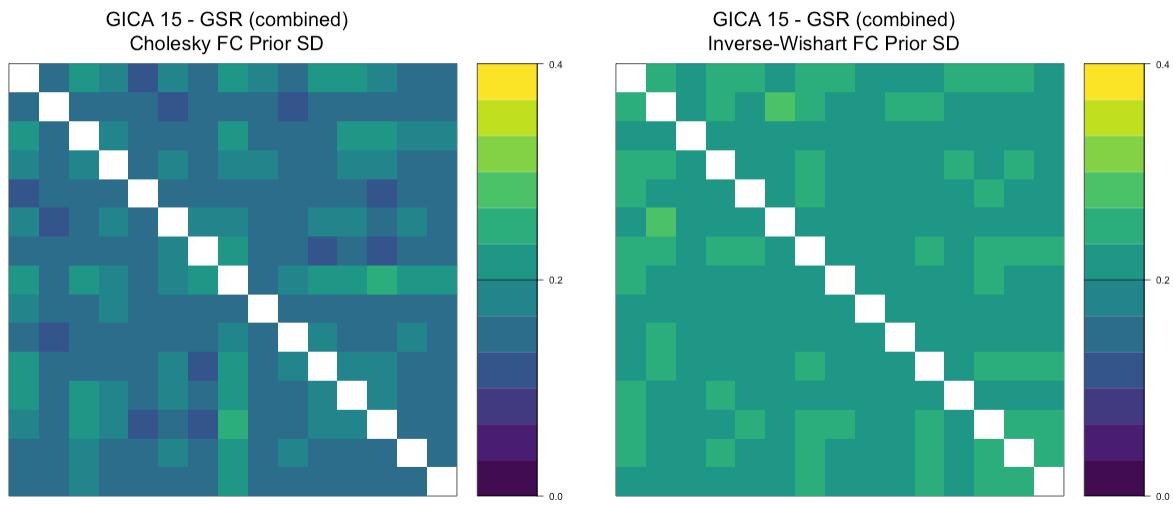
- [prior\_name]\_FC\_Cholesky\_mean.png
- [prior\_name]\_FC\_Cholesky\_sd.png
- [prior\_name]\_FC\_InverseWishart\_mean.png
- [prior\_name]\_FC\_InverseWishart\_sd.png

These visualizations allow for a direct comparison of spatial FC structure and uncertainty across priors and estimation methods.

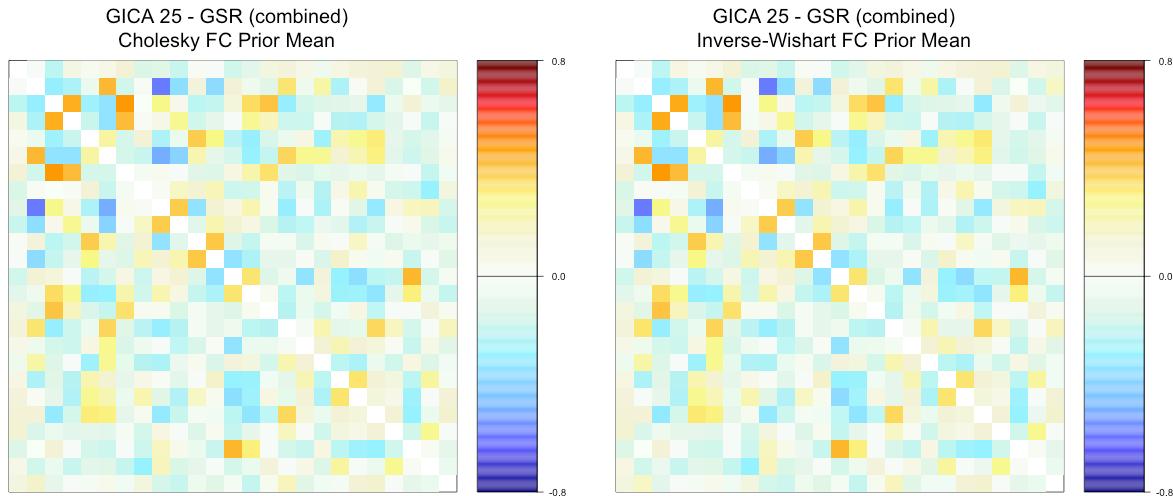
The figures below show the mean and standard deviation of FC priors for each parcellation (GICA15, GICA25, GICA50, Yeo17) using Cholesky and Inverse-Wishart methods. Only combined priors are shown.

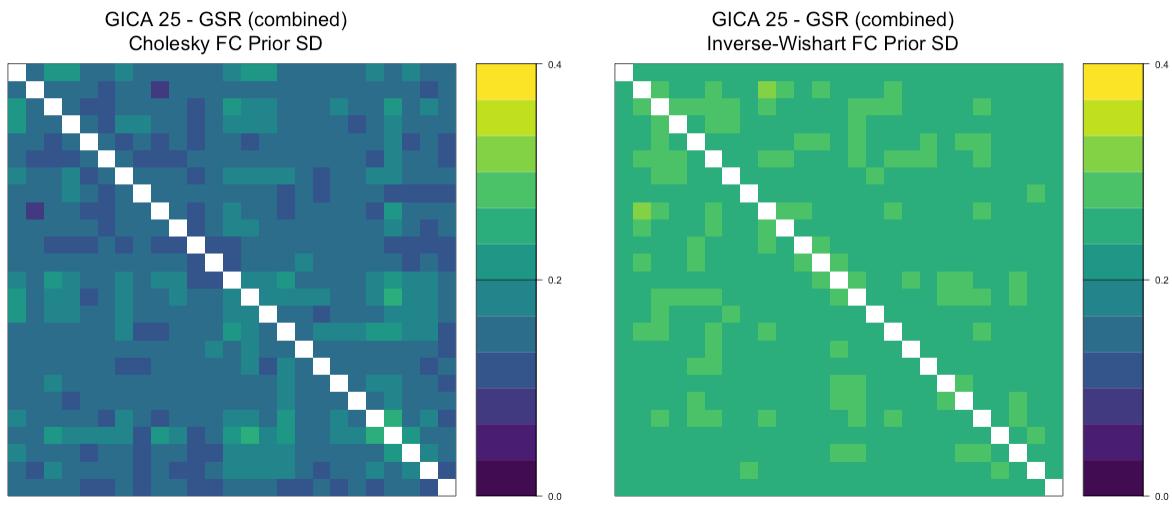
#### 0.9.4.3 5.4.1 GICA15



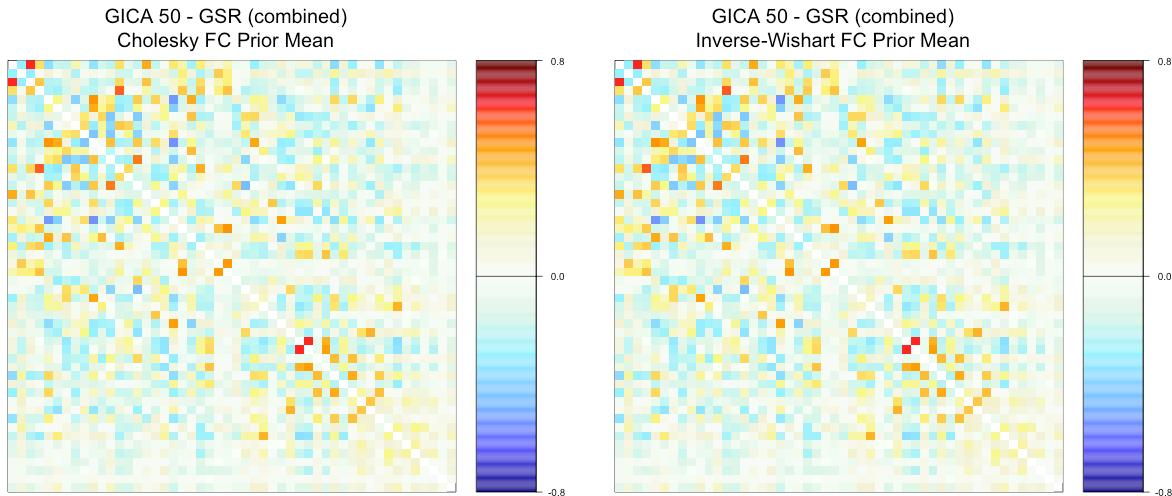


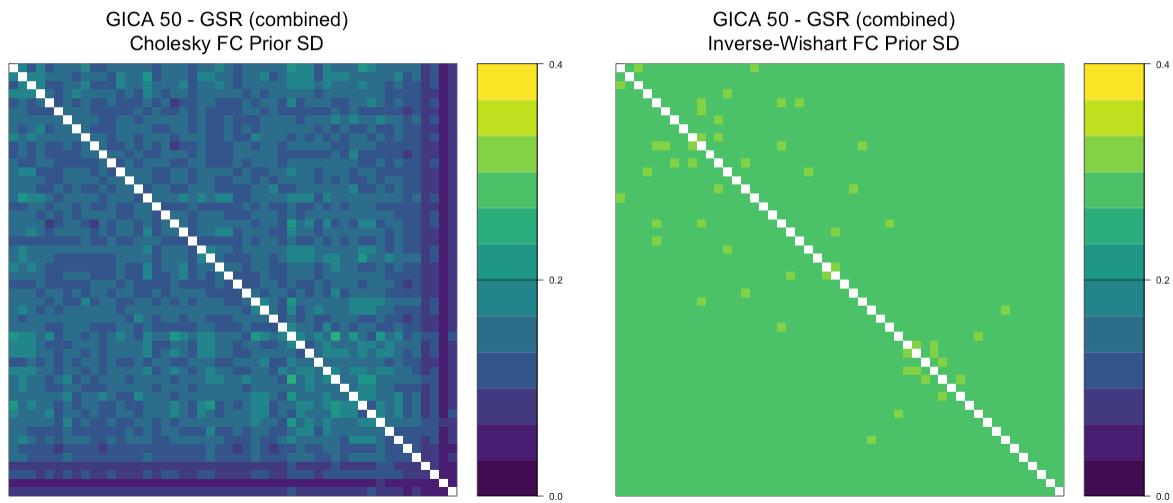
#### 0.9.4.4 5.4.2 25 ICs



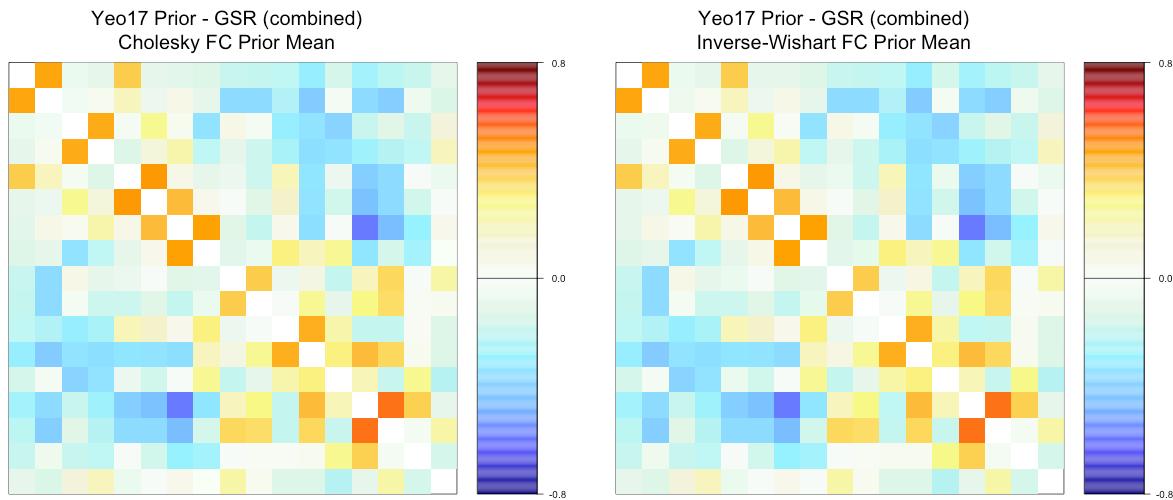


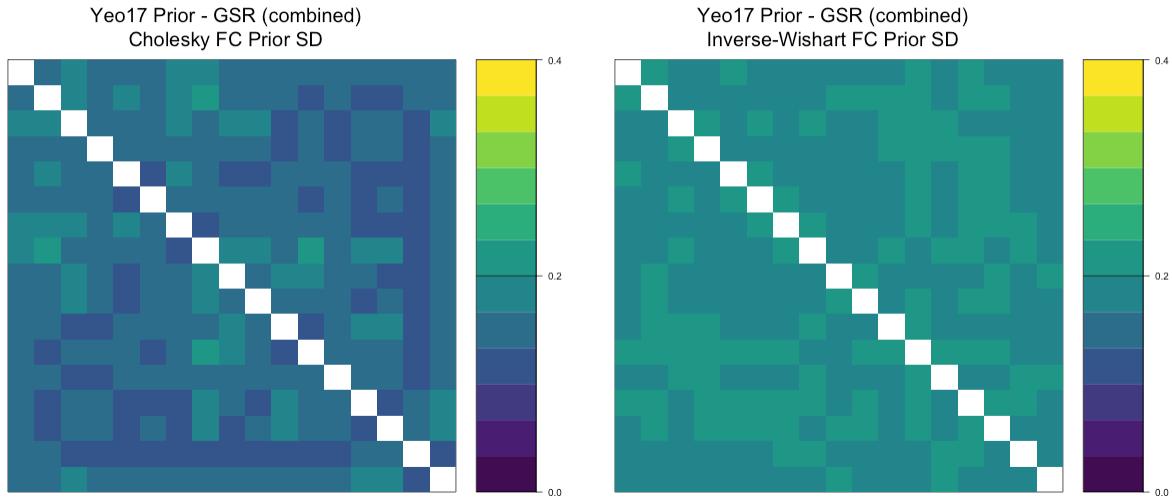
#### 0.9.4.5 5.4.3 50 ICs





#### 0.9.4.6 5.4.4 Yeo17





## 0.10 6. Step 2: Using Priors for Individual-Level Brain Mapping

In this section, we demonstrate how to apply the population-level priors estimated in Section 4 to perform subject-level analysis using the `BayesBrainMap` package.

The process involves two steps:

1. Fitting the Bayesian brain mapping model to subject data using a precomputed prior.
2. Identifying regions of significant deviation from the prior mean (i.e., areas of engagement).

This example uses:

- A prior based on Yeo17 template with global signal regression (GSR) and the combined list of subjects.
- One subject's resting-state data in CIFTI format. In this case, we use HCP subject 100206.

```
# Load population prior
prior <- readRDS("priors/Yeo17/prior_combined_Yeo17_GSR.rds")

# Load subject fMRI data (CIFTI format)
BOLD <- c(file.path(dir_data, "inputs",
                     "rfMRI_REST1_LR_Atlas_MSMAll_hp2000_clean.dtseries.nii"),
           file.path(dir_data, "inputs",
                     "rfMRI_REST1_RL_Atlas_MSMAll_hp2000_clean.dtseries.nii"))
```

**0.10.0.1 6.1 Load Subject-Level fMRI Data and Prior** The fMRI input must be a CIFTI, NIFTI, or matrix object compatible with the prior.

**0.10.0.2 6.2 Estimate Subject-Level Networks** Once the data is loaded, we fit the Bayesian brain mapping model to obtain individualized functional networks aligned to the prior components:

```
bMap <- BrainMap(  
  BOLD = BOLD,  
  prior = prior,  
  TR = 0.72,  
  drop_first = 15  
)
```

#### 0.10.0.3 6.3 Identify Engagement Maps

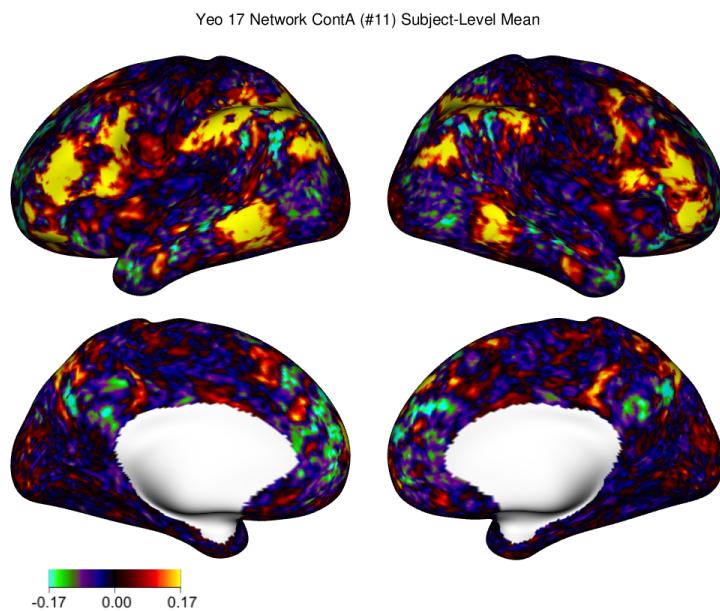
```
eng <- engagements(  
  bMap = bMap  
)
```

#### 0.10.0.4 6.4 Visualize the Results

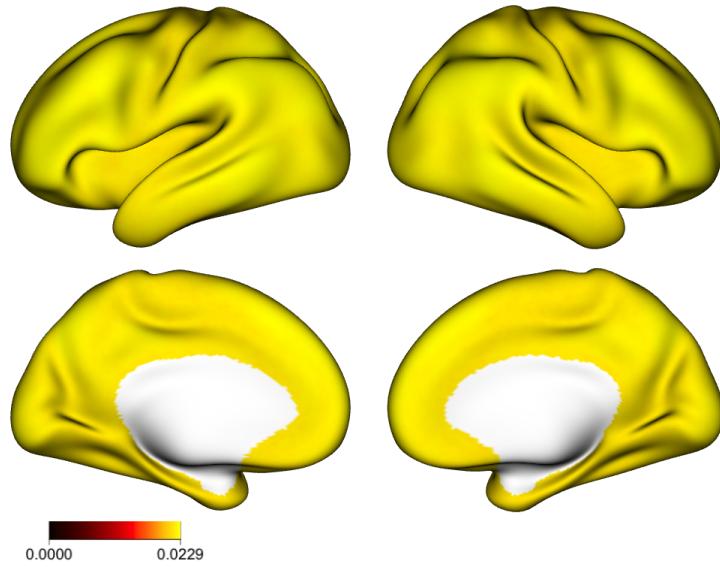
We now plot:

1. The subject-level networks estimated by `BrainMap()` (both mean and standard error).
2. The engagement maps, showing regions of deviation from the prior mean.

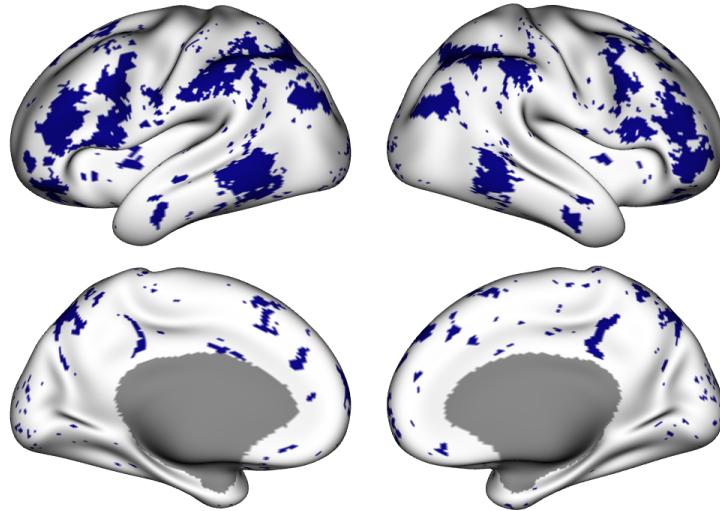
For all outputs, we only visualize ContA network from the Yeo 17 parcellation.



Yeo 17 Network ContA (#11) Subject-Level SE



Yeo 17 Network ContA (#11) Engagement Map



### Labels

- Medial Wall
- Not engaged
- Engaged:  $x > 0$

## .1 Appendix A: Setup

To reproduce this workflow, first clone the repository to your local machine or cluster:

```
git clone https://github.com/mandymejia/BayesianBrainMapping-Templates.git
cd BayesianBrainMapping-Templates
```

Next, download the required `data_OSF/` and `priors` folders from the following OSF link:

[https://osf.io/n3wk5/?view\\_only=0d95b31090a245eb9ef51fe262be60ef](https://osf.io/n3wk5/?view_only=0d95b31090a245eb9ef51fe262be60ef)

Once downloaded, unzip the folder and place in the folder in the GitHub directory with the same corresponding name. The folder structure should look like this:

This section initializes the environment by loading required packages, setting analysis parameters, and defining directory paths.

**Important:** Before running the workflow, you must review `0_setup.R` and install any necessary packages, ensure you have an installation of Connectome Workbench, and update the following variables to match your local or cluster environment:

- `dir_project` (path to the GitHub folder)
- `dir_HCP` (path to the HCP data)
- `HCP_unrestricted_fname` (path to the unrestricted HCP CSV if you have access to it)
- `HCP_restricted_fname` (path to the restricted HCP CSV if you have access to it)
- `wb_path` (location of the CIFTI Workbench on your system)

```
github_repo_dir <- getwd()
src_dir <- file.path(github_repo_dir, "src")
source(file.path(src_dir, "0_setup.R"))
```

## .2 Appendix B: Subject Filtering

### .2.1 Appendix B.1: Filter Subjects by Sufficient fMRI Scan Duration

We begin by filtering subjects based on the fMRI scan duration after motion scrubbing. For each subject, and for each session (REST1, REST2) and encoding direction (LR, RL), we compute framewise displacement (FD) using the `fMRIscrib` package. We use a lagged and filtered version of FD (CITE Pham Less is More and Power/Fair refs therein) appropriate for multiband data. FD is calculated from the `Movement_Regressors.txt` file available in the HCP data for each subject, encoding and session.

A volume is considered valid if it passes an FD threshold, and a subject is retained only if both sessions in both encodings have at least 10 minutes (600 seconds) of valid data.

The final subject lists include those who passed the filtering criteria separately for each encoding: LR, RL, and their intersection, referred to as `thecombed` list. The combined list includes only subjects who passed all criteria for both LR and RL encodings across both visits (REST1 and REST2), and is the one used throughout this project.

```
# -----
# @Nohelia, can you report the FD threshold used, both here and above in the
# Methods section where the filters are mentioned?
# -----

# This script filters subjects based on motion using framewise displacement (FD)
# from fMRIscrib.
```

Table 2: First rows of FD summary table

X	subject	session	encoding	mean_fd	valid_time_sec
1	100206	REST1	LR	0.1017240	858.24
2	100206	REST2	LR	0.1361220	858.96
3	100206	REST1	RL	0.0698779	864.00
4	100206	REST2	RL	0.0824894	863.28

```
# For each subject, encoding (LR/RL), and session (REST1/REST2), it computes FD
# and valid scan
# time after excluding high-motion volumes.
# Subjects with 10 minutes of valid data in both sessions are retained.
# Outputs (saved in dir_results):
# - Valid subject lists for LR, RL, and combined encodings (intersection)
# - FD summary per subject/session/encoding

#set up path, etc.
github_repo_dir <- getwd()
src_dir <- file.path(github_repo_dir, "src")
source(file.path(src_dir, "0_setup.R"))

#run script to exclude sessions based on head motion
source(file.path(src_dir,"1_fd_time_filtering.R"))
```

During this step, an FD summary table is generated with the following columns:

- subject: HCP subject ID
- session: REST1 or REST2
- encoding: LR or RL
- mean\_fd: mean framewise displacement
- valid\_time\_sec: total duration of valid data in seconds

```
# Read FD summary
fd_summary <- read.csv(file.path(dir_data, "outputs", "filtering", "fd_summary.csv"))

# Display the first 4 rows
knitr::kable(head(fd_summary, 4), caption = "First rows of FD summary table")
```

**.2.1.1 Preview of FD Summary Table** As shown above, subject 100206 qualifies for further analysis because each of the four sessions (REST1/REST2 × LR/RL) contains at least 600 seconds of valid data.

The script is currently designed to filter based on valid time only, but it can be easily adapted to apply additional constraints such as maximum mean FD thresholds if desired (e.g., mean\_fd < 0.1).

## .2.2 Appendix B.2: Filter Unrelated Subjects

Building on the previous step, we use the HCP restricted demographic data to exclude related individuals. This step helps ensure the statistical independence of subjects in the group-level priors estimation.

For the LR, RL, and combined lists of valid subjects derived in the previous step, we:

1. Subset the HCP restricted demographics to include only those subjects with at least 10 minutes remaining after scrubbing.
2. Filter by Family\_ID to retain a single individual per family.

Note: This step requires access to the HCP restricted data. If you do not have access, you can skip this step, resulting in some related subjects being included in your training data.

```
# This script filters subjects to retain only unrelated individuals, using Family ID
# information from the restricted HCP data.
# For each encoding (LR, RL, combined), it selects one subject per family from the
# FD-valid lists.
# Outputs (saved in dir_personal due to restricted data):
# - Unrelated subject lists for LR, RL, and combined encodings (intersection)

source(file.path(src_dir, "2_unrelated_filtering.R"))
```

## .2.3 Appendix B.3: Balance Sex Within Age Groups

In the final step of subject selection, we balance sex across age groups to reduce potential demographic bias in priors estimation.

For the LR, RL, and combined lists of valid subjects derived in the previous step, we:

- Subset the HCP unrestricted demographics to include only those subjects.
- Split subjects by age group and examine the sex distribution within each group.
- If both sexes are present but imbalanced, we randomly remove subjects from the overrepresented group to achieve balance.

Note: If the unrelated subject filtering step is skipped (e.g., due to lack of restricted data access), the code automatically falls back to using valid\_<encoding>\_subjects\_FD instead of valid\_<encoding>\_subjects\_unrelated.

The final list of valid subjects is saved in dir\_results as:

- valid\_<encoding>\_subjects\_balanced.csv
- valid\_<encoding>\_subjects\_balanced.rds (used in the prior estimation step)

```
# This script balances sex within each age group for subjects who passed FD and
# unrelated filtering.
# For each encoding (LR, RL, combined), it samples subjects to equalize the number
# of males and females per age group,
# unless an age group includes only one gender (in which case no balancing is applied).
# Uses age and gender information from the unrestricted HCP data.
# Outputs (saved in dir_personal):
# - Sex-balanced subject lists for LR, RL, and combined encodings (as .csv and .rds)

source(file.path(src_dir, "3_balance_age_sex.R"))
```

### .3 Appendix C: Scrubbing and Temporal Truncation

Given the HCP TR of 0.72 seconds, 10 minutes corresponds to:

```
T_total <- floor(600 / TR_HCP) # ~833 volumes
```

To define the volumes to scrub (i.e., exclude beyond 10 minutes), we compute:

```
T_scrub_start <- T_total + 1
scrub_BOLD1 <- replicate(length(BOLD_paths1), T_scrub_start:nT_HCP, simplify = FALSE)
scrub_BOLD2 <- replicate(length(BOLD_paths2), T_scrub_start:nT_HCP, simplify = FALSE)
scrub <- list(scrub_BOLD1, scrub_BOLD2)
```

Because `drop_first = 15` removes frames before truncation, the final retained time series per scan will be slightly shorter than 10 minutes. Approximately:

$(833 - 15) * 0.72 = \sim 589 \text{ seconds } (\sim 9.8 \text{ minutes})$

### .4 Appendix D: Prepare Yeo17 Parcellation for Prior Estimation

In this step, we load and preprocess a group-level cortical parcellation to be used as the template to estimate the priors in the next step. Specifically, we use the Yeo 17-network parcellation (`Yeo_17`) and perform the following operations:

- Simplify the labels by collapsing hemisphere-specific naming and removing subnetwork identifiers, grouping regions by their main network.
- Create a new `dlabel` object that maps each vertex to its corresponding network.
- Mask out the medial wall to exclude it from analysis.

The resulting parcellation is saved as `Yeo17_simplified_mwall.rds`.

```
# This script simplifies the Yeo 17-network parcellation by collapsing region
# labels and masking out the medial wall.
# It creates a cleaned version of the parcellation suitable for downstream analyses.
# Output:
# - Saved as RDS file in dir_data: "Yeo17_simplified_mwall.rds"

source(file.path(src_dir, "4_parcellations.R"))
```

We can visualize the Yeo17 networks and their corresponding labels:

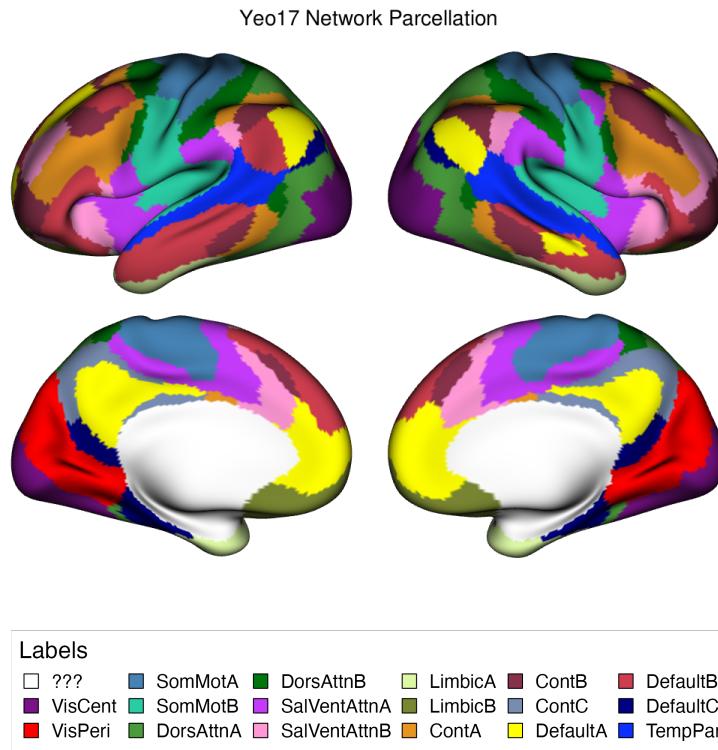
```
# Load libraries
library(ciftiTools)
library(rgl)
rgl::setupKnitr()

# Load the parcellation
yeo17 <- readRDS(file.path(dir_data, "outputs", "Yeo17_simplified_mwall.rds"))
yeo17 <- add_surf(yeo17)
```

```

view_xifti_surface(
  xifti = yeo17,
  widget = TRUE,
  title = "Yeo17 Network Parcellation",
  legend_ncol = 6,
  legend_fname = file.path(dir_data, "outputs", "parcellations_plots",
                            "Yeo17", "Yeo17_legend.png"),
  fname=file.path(dir_data, "outputs", "parcellations_plots", "Yeo17")
)

```



## .5 Appendix E: Example Function Call for Prior Estimation

```

# For detailed parameter descriptions, run: ?estimate_prior

estimate_prior(
  BOLD = BOLD_paths1,          # REST1 LR scans (list of file paths)
  BOLD2 = BOLD_paths2,         # REST2 LR scans (same subjects/order as BOLD)
  template = GICA,            # GICA 15-component parcellation (CIFTI dscalar file path)
  GSR = TRUE,                 # Apply global signal regression
  TR = 0.72,                  # Repetition time in seconds
  hpf = 0.01,                 # High-pass filter cutoff in Hz
  Q2 = 0,                     # No nuisance IC denoising
  drop_first = 15,            # Drop first 15 volumes
  scrub = scrub,              # Timepoints to scrub (list format)
  verbose = TRUE              # Print progress updates
)

```

## References

- Mejia, Amanda F, Yu Yue, David Bolin, Finn Lindgren, and Martin A Lindquist. 2020. “A Bayesian General Linear Modeling Approach to Cortical Surface fMRI Data Analysis.” *Journal of the American Statistical Association* 115 (530): 501–20.