

# Individualized functional topographic mapping with the BayesBrainMap R package and Human Connectome Project-derived priors

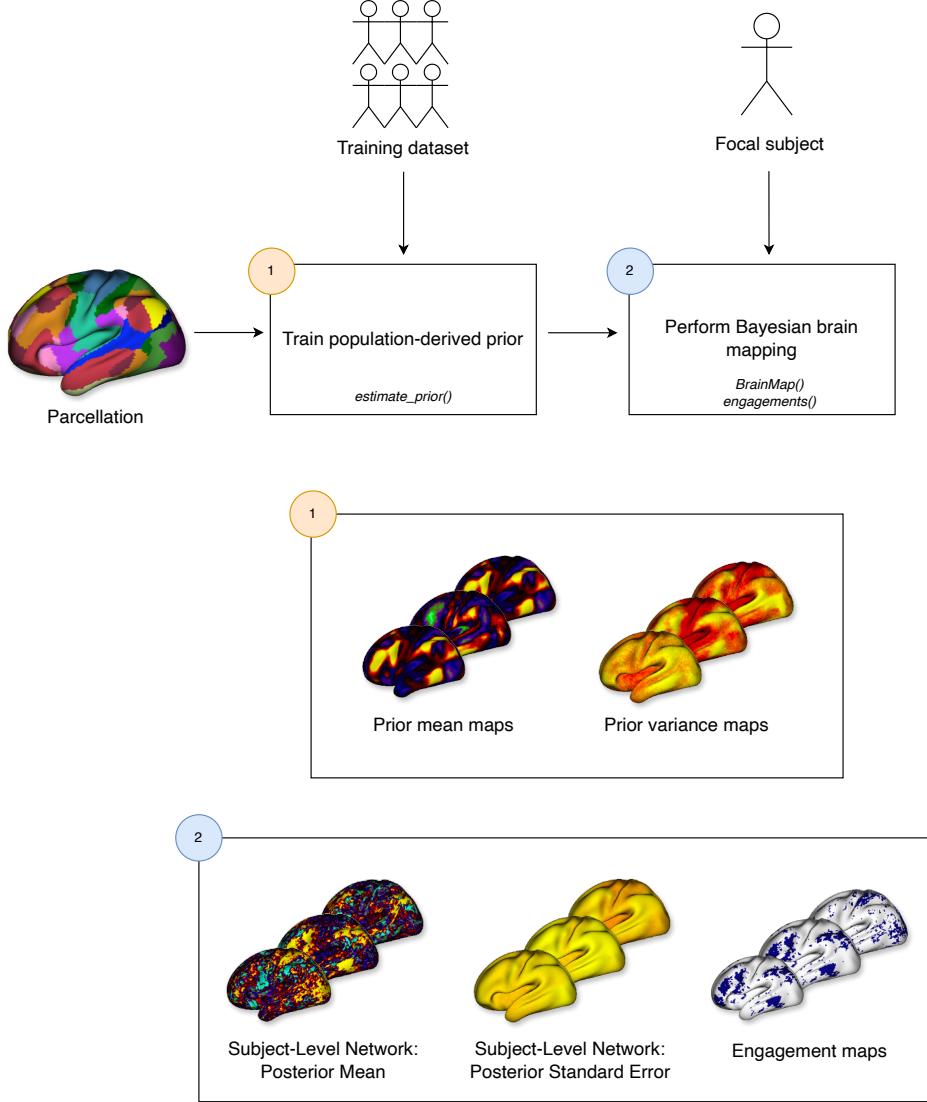
Nohelia Da Silva Sanchez, Damon D. Pham, Ellyn Butler, Amanda F. Mejia

2025-05-07

## 1. Introduction

Bayesian brain mapping (BBM) is a technique for producing individualized functional brain topographic maps from existing group-level network maps. Group-level network maps are commonly group ICA maps or group average parcellations, but other types of networks maps such as non-negative matrix factorization and PROFUMO can be used. In the case of ICA, BBM is known as template ICA (Mejia et al. 2020). BBM is a hierarchical source separation model, with priors on the spatial topography and, optionally, on the functional connectivity. The priors are estimated a-priori, so the model can be fit to individual-level fMRI data. The noise-reduction properties of the population-derived priors result highly accurate and reliable individual network topography maps and the functional connectivity between them. Importantly, the subject-level network maps are matched to the corresponding group networks from the template (i.e. parcellations or group ICA maps). Because BBM is applicable to individual-level analysis, it is computationally convenient and has potential clinical utility.

Once a set of group-level network maps has been chosen, there are two steps to performing BBM. Both are implemented in the `BayesBrainMap` R package.



Here, we perform Step 1 using data from the Human Connectome Project (HCP). Specifically, we train CIFTI-format Bayesian brain mapping priors using a variety of ICA and parcellation-based templates, listed below. The population-derived priors described here are available for use for individual-level Bayesian brain mapping. For analysis of individuals from other populations, it is often desirable to train the prior on a set of training subjects representative of that population. To facilitate this, we also provide and describe the code used to produce the HCP-derived priors, so that this workflow can be easily reproduced in other datasets. Finally, we illustrate the use of

For the choice of group-level network maps, we provide several options:

- Group ICA maps from the HCP at resolutions from 15 to 50 (CITE)
- The 17 Yeo networks (CITE)

## 2. Setup

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

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.

### 3. Choosing Training Subjects

Before estimating the BBM priors, we first select a high-quality, balanced subject sample to ensure reliable, representative priors. Starting from the full HCP sample of 1206, we apply the following filtering steps:

#### 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.

### 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.

#### 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`

## 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.

## 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")
```

## 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.

## 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.

### 5.1 Generate and Save Parcellation Visualizations

#### 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_OSF/outputs/parcellations_plots/Yeo17`.

#### 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_OSF/outputs/parcellations_plots/GICA15/`
- `data_OSF/outputs/parcellations_plots/GICA25/`
- `data_OSF/outputs/parcellations_plots/GICA50/`

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

### 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_OSF/priors_plots/GICA15/combined/GSR/  
data_OSF/priors_plots/GICA15/combined/noGSR/  
data_OSF/priors_plots/GICA25/LR/noGSR/  
data_OSF/priors_plots/Yeo17/RL/GSR/  
...
```

### 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:

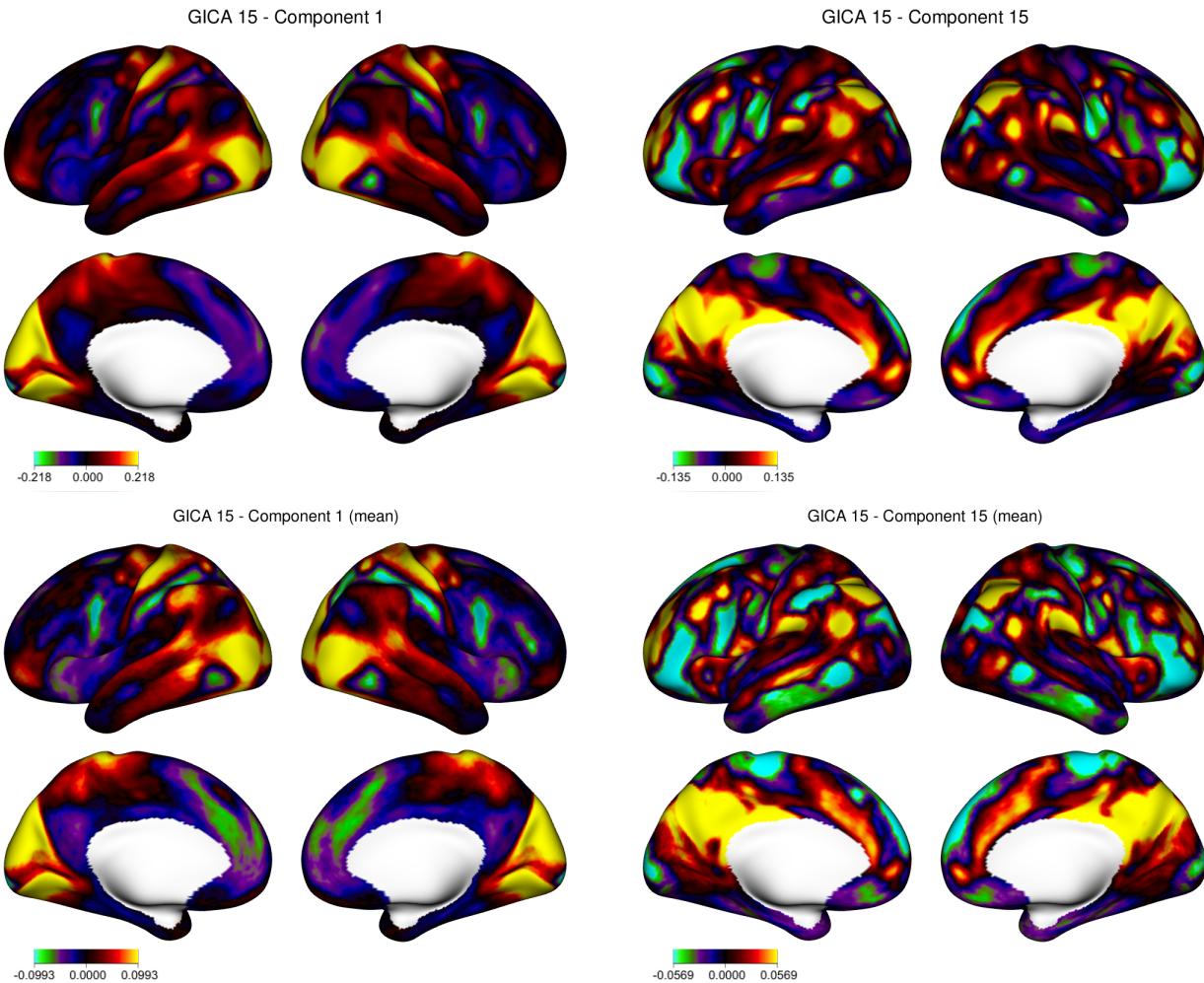
- First and Last Parcellation Map
- First and Last Component Mean
- First and Last Component Standard Deviation

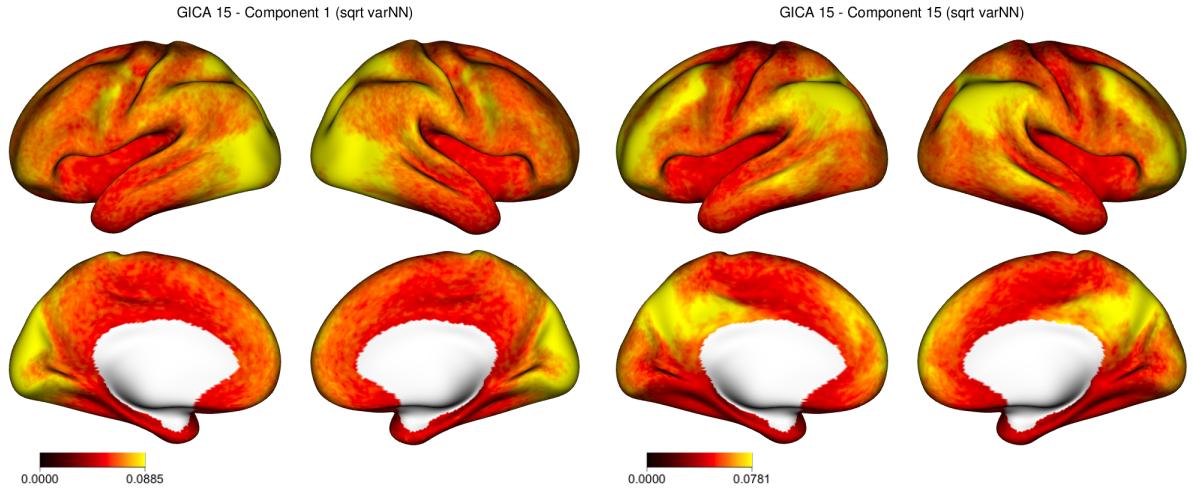
These summaries are shown in a 2-column grid layout per parcellation to highlight spatial structure and variability.

All images were generated using the scripts:

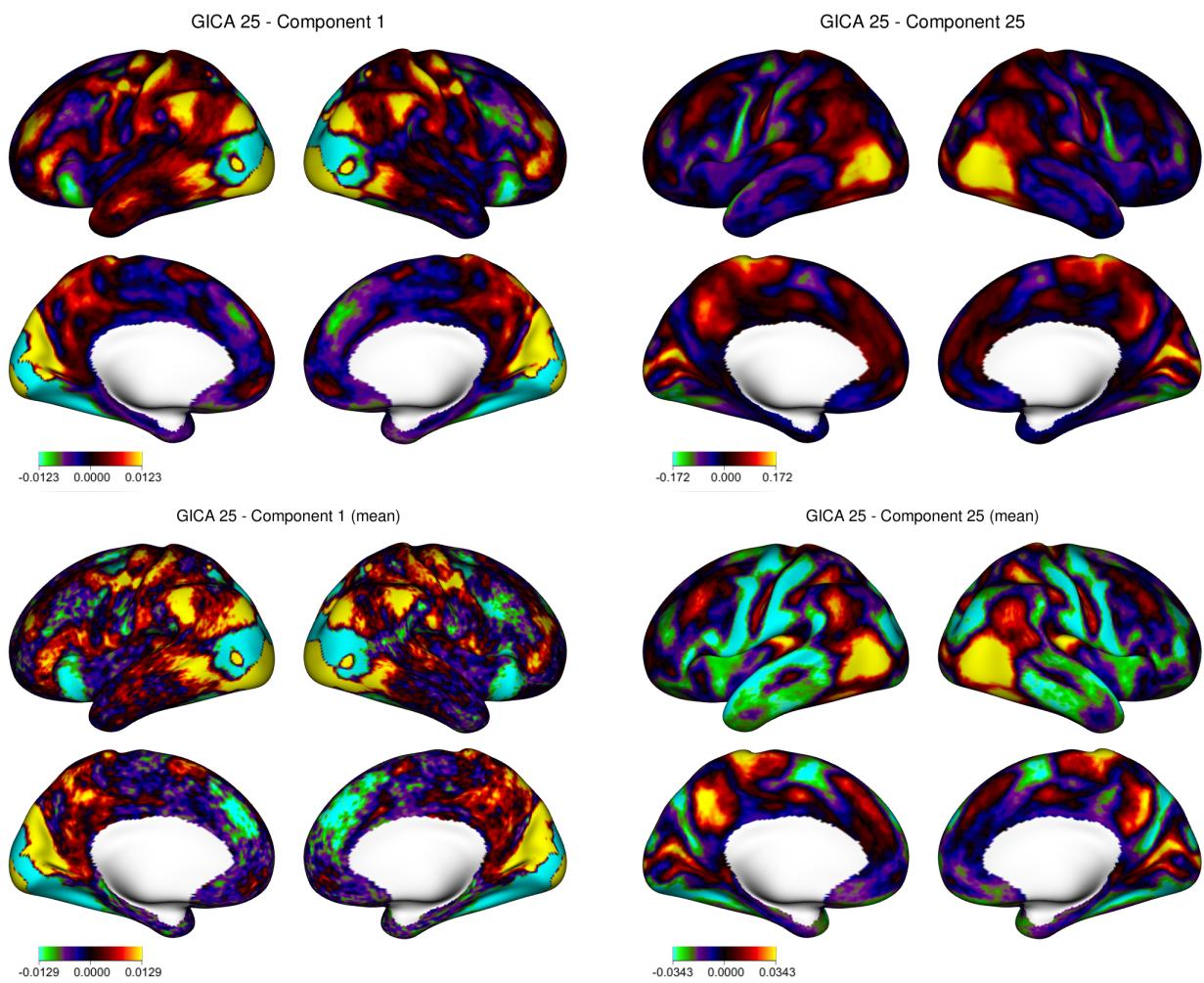
- `8_visualization_Yeo17parcellations.R`
- `9_visualization_GICAparchellations.R`
- `6_visualization_prior.R`

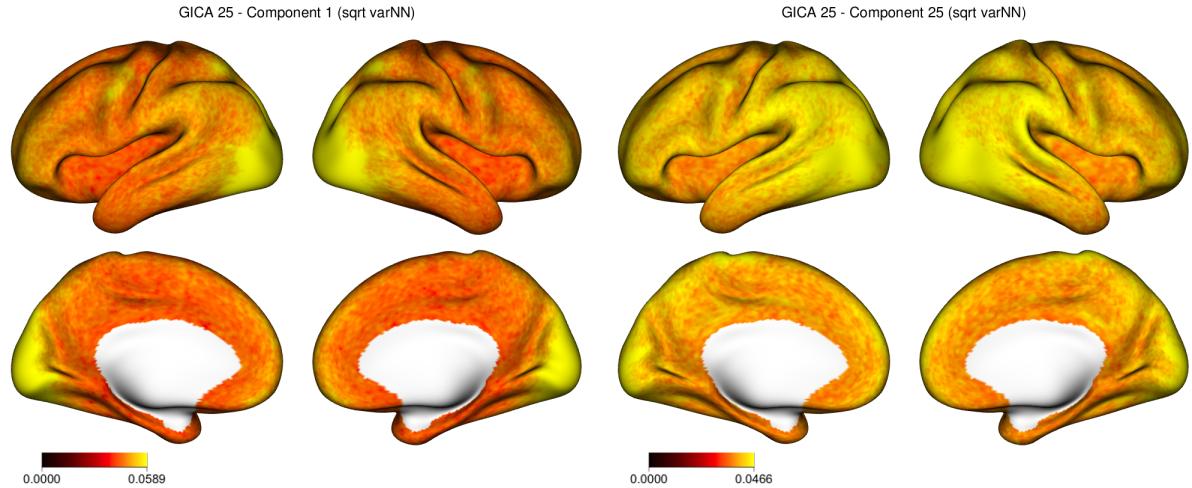
#### 5.3.1 15 ICs



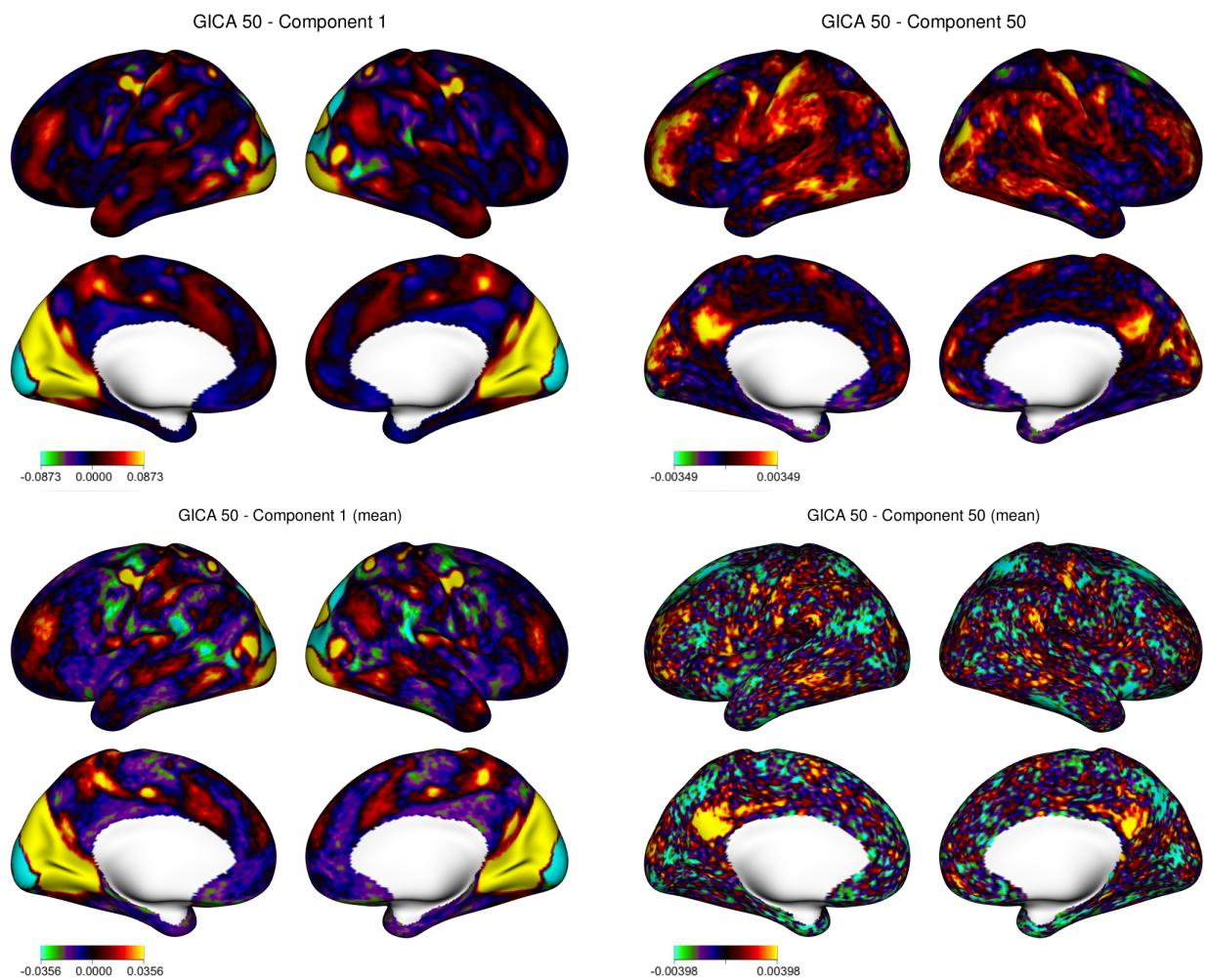


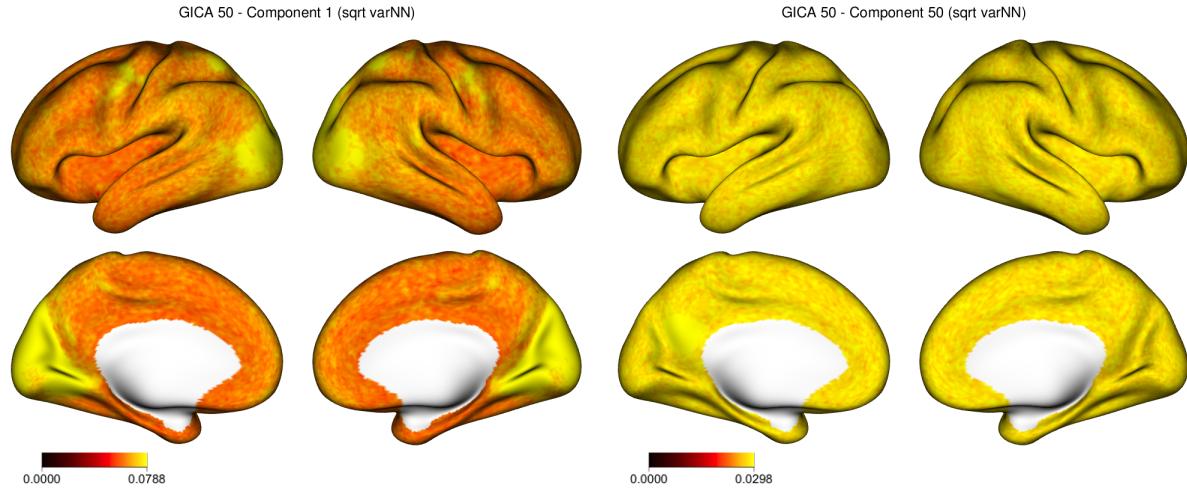
### 5.3.2 25 ICs





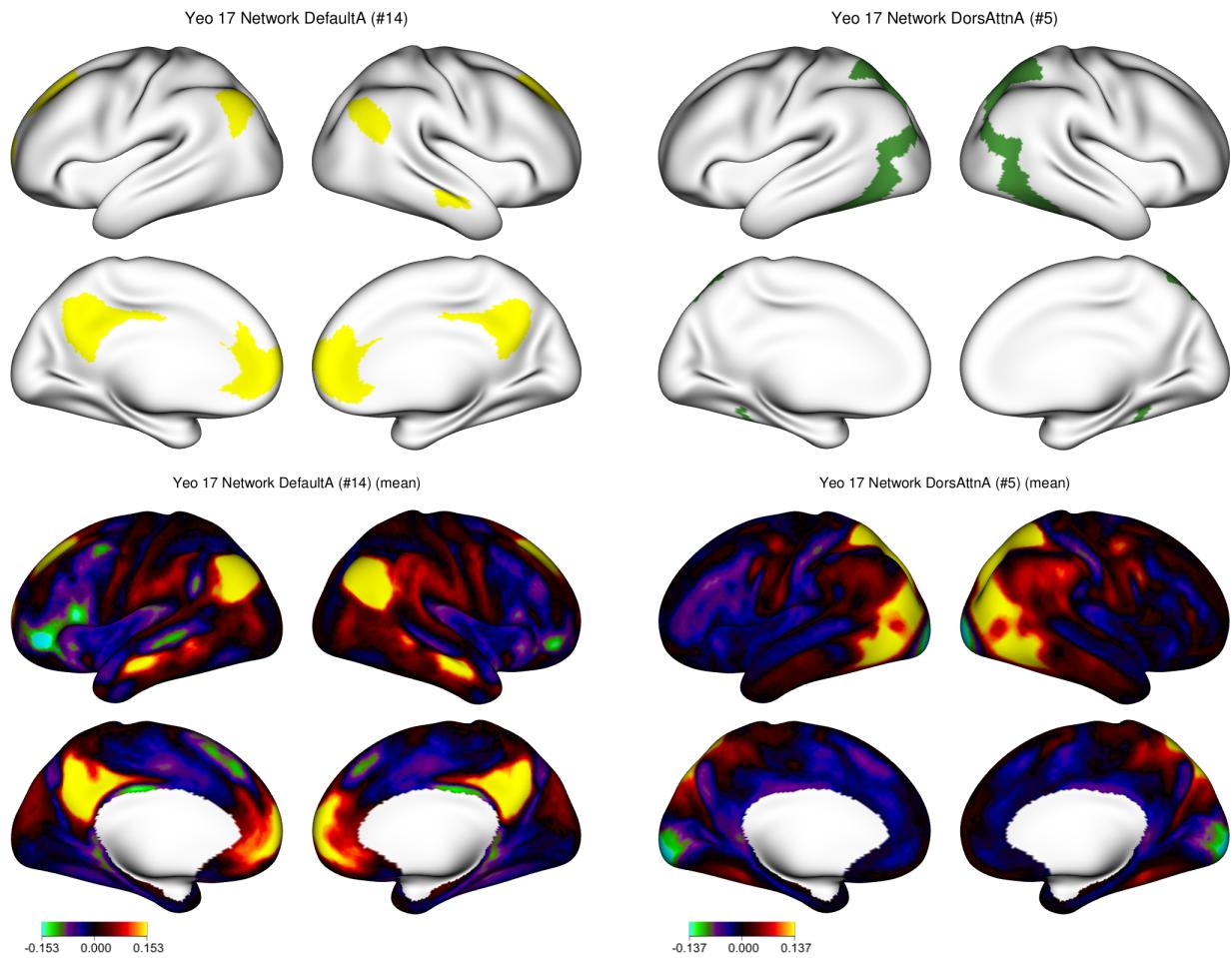
### 5.3.3 50 ICs

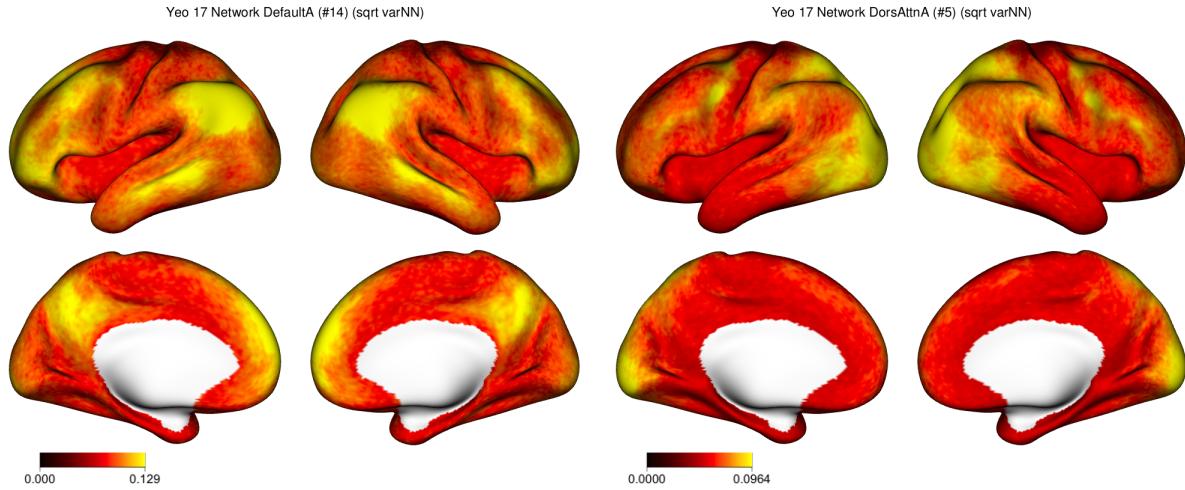




### 5.3.4 Yeo17

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





## 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

### 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)

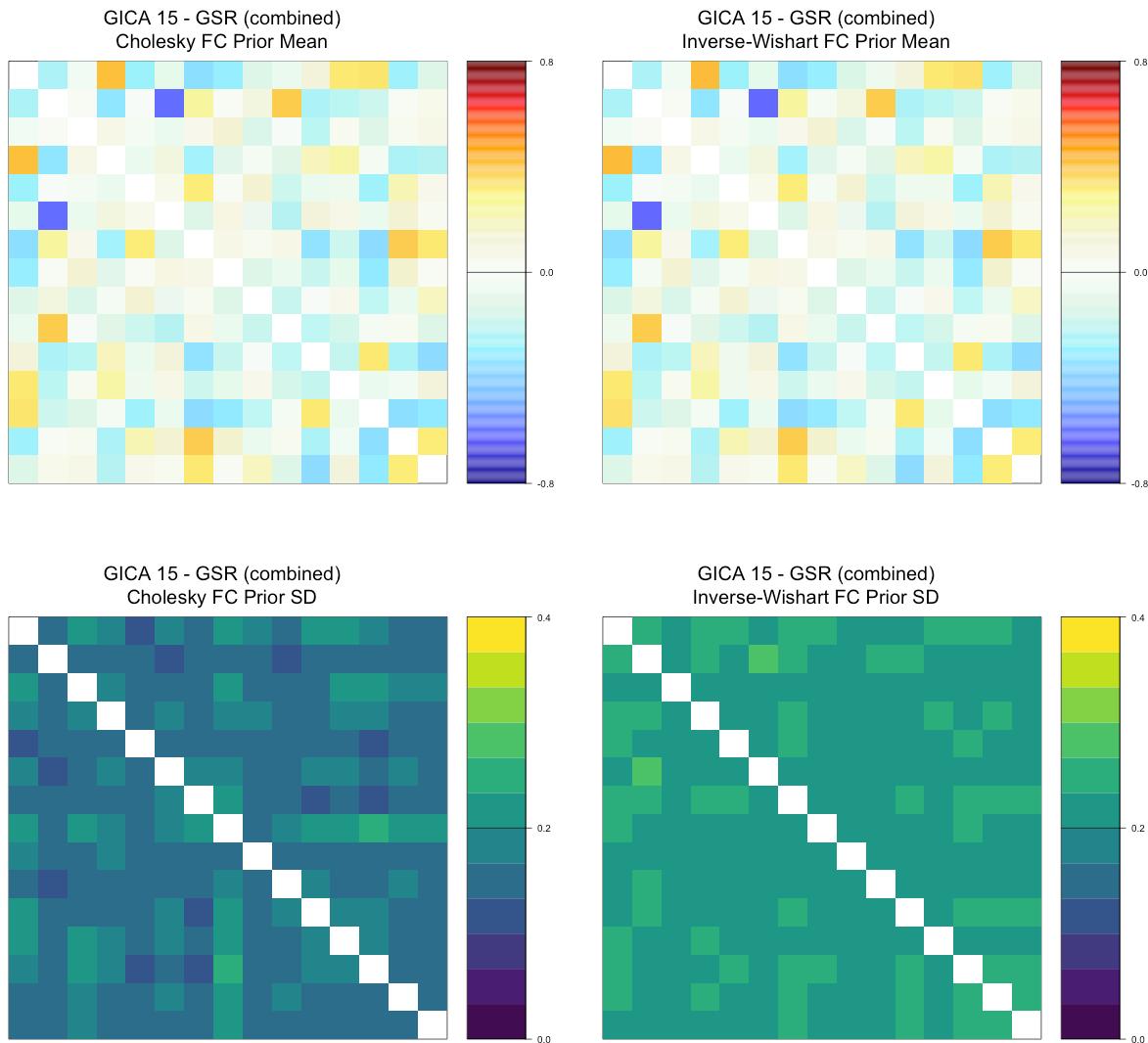
## PNG images (4 per prior)

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

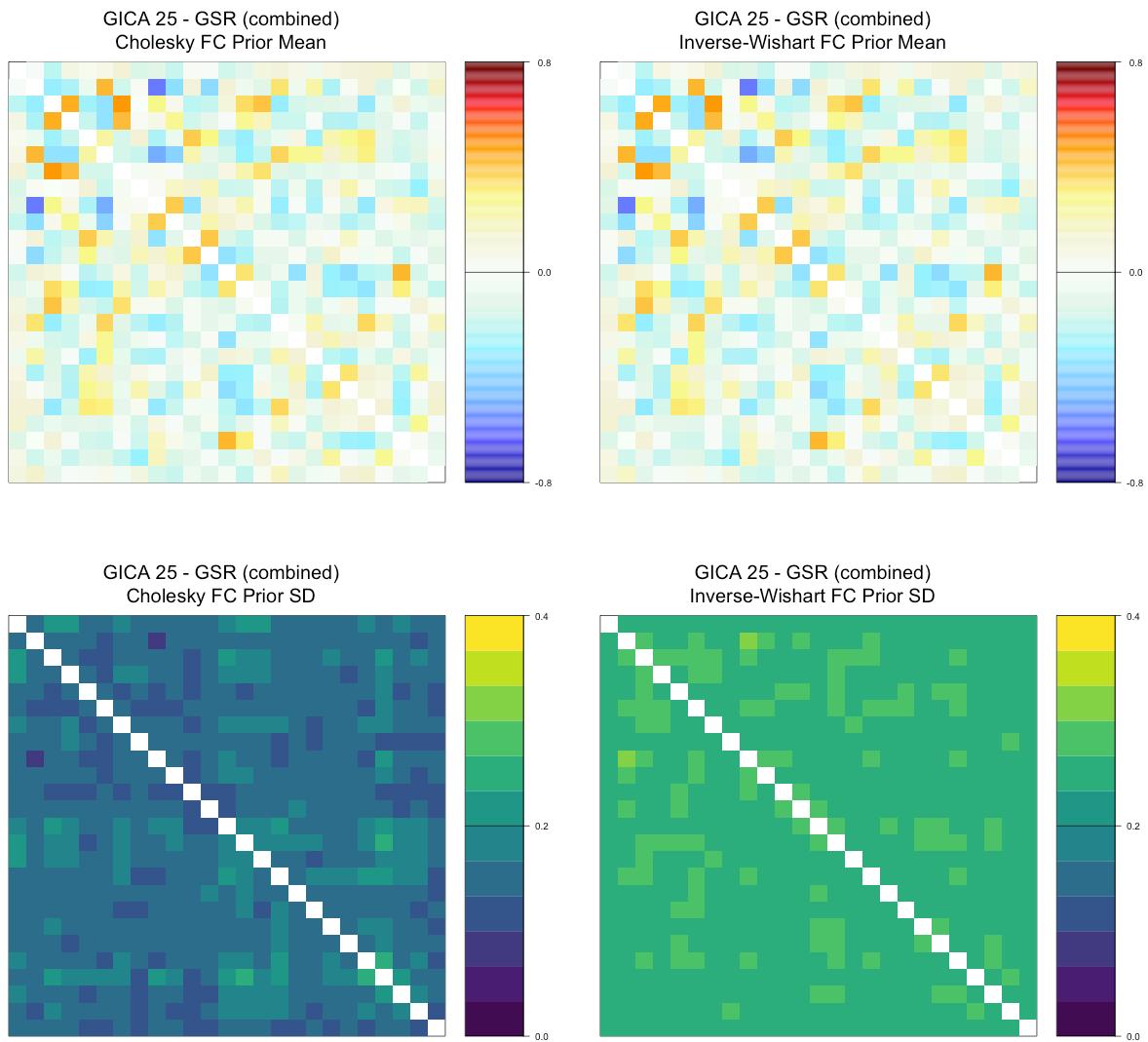
These visualizations allow for a direct comparison of spatial FC structure and uncertainty across priors and estimation methods.

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.

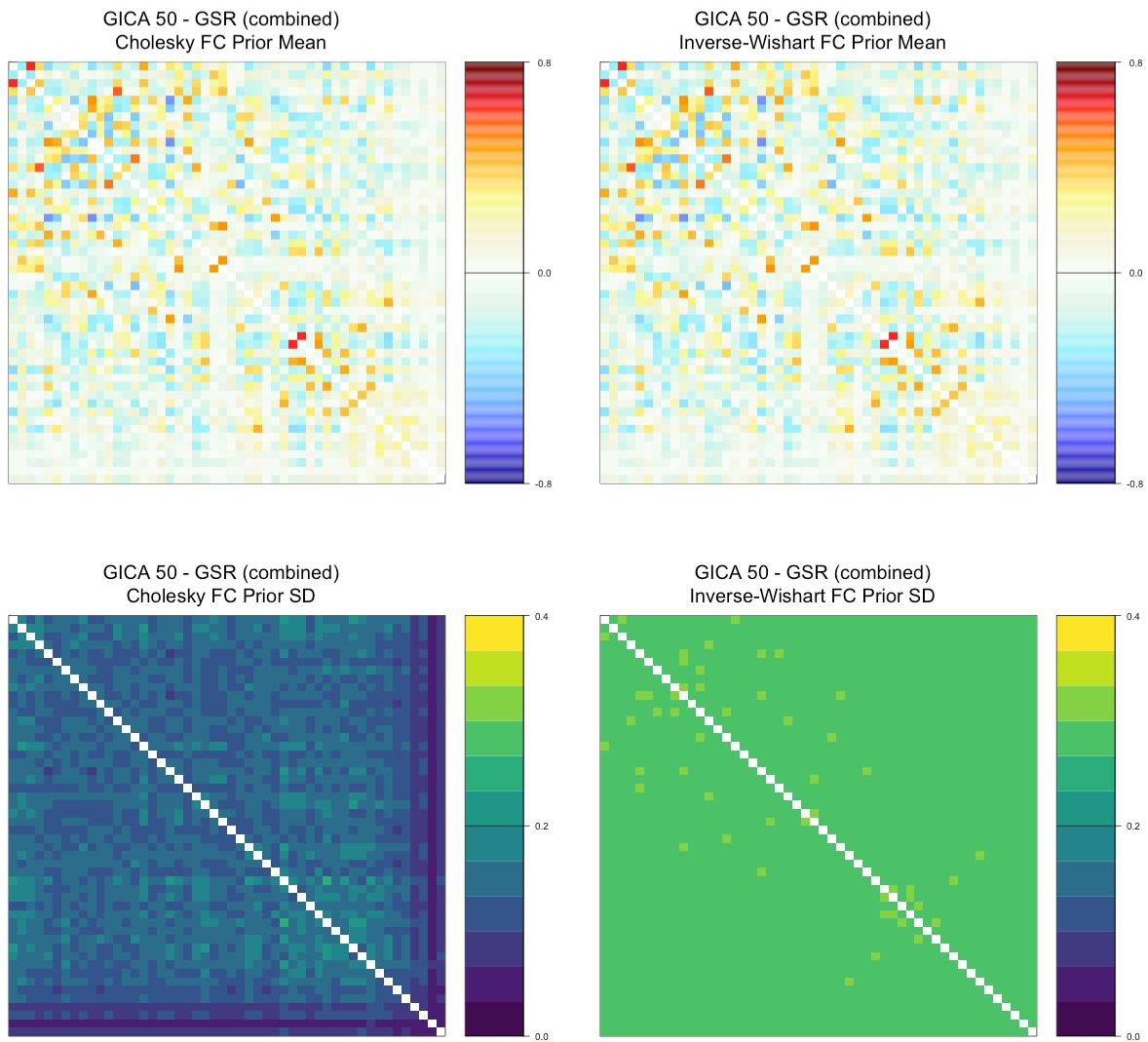
### 5.4.1 GICA15



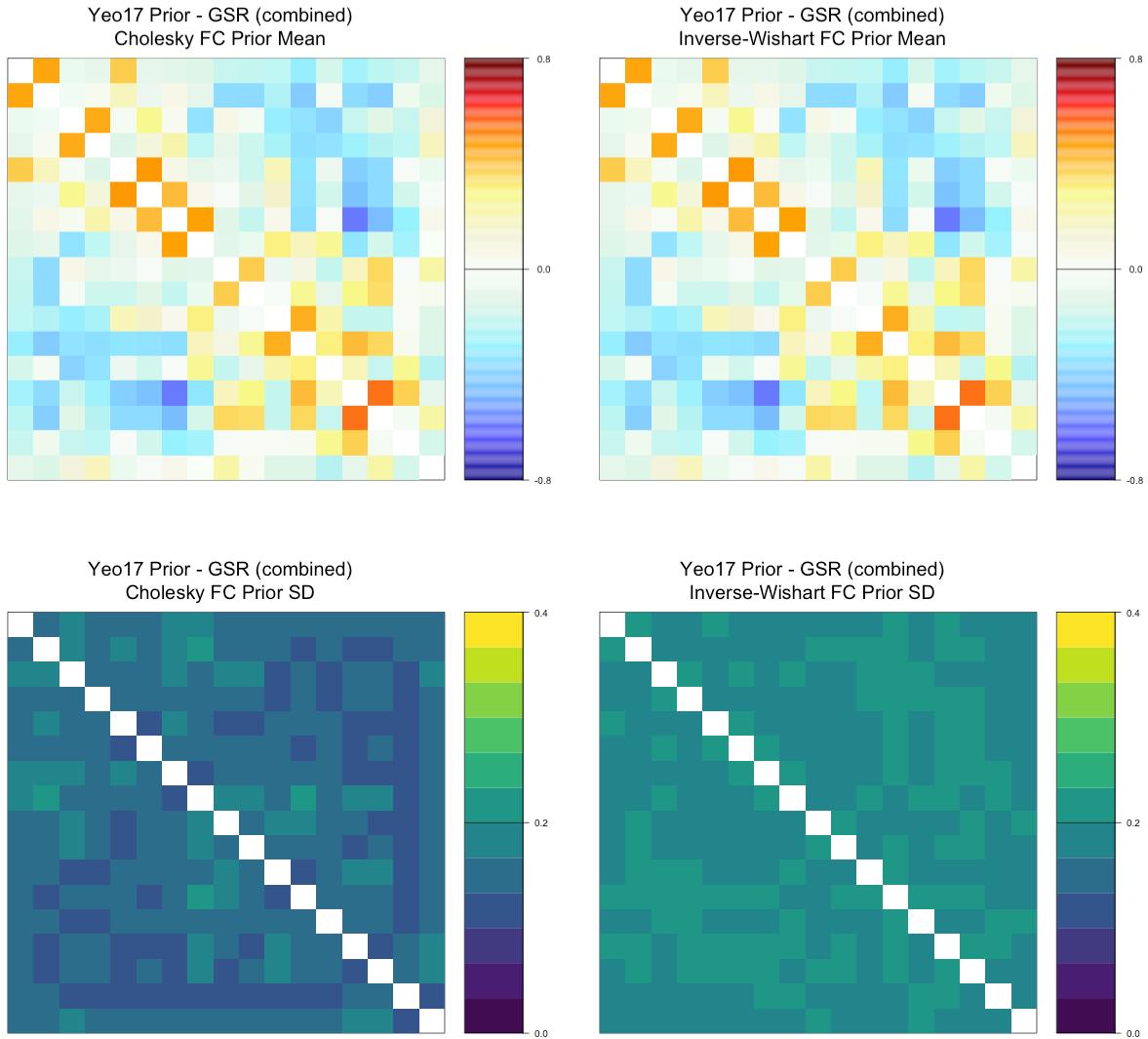
#### 5.4.2 25 ICs



#### 5.4.3 50 ICs



#### 5.4.4 Yeo17



## 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"))

```

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

**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
)

```

### 6.3 Identify Engagement Maps

```

eng <- engagements(
  bMap = bMap
)

```

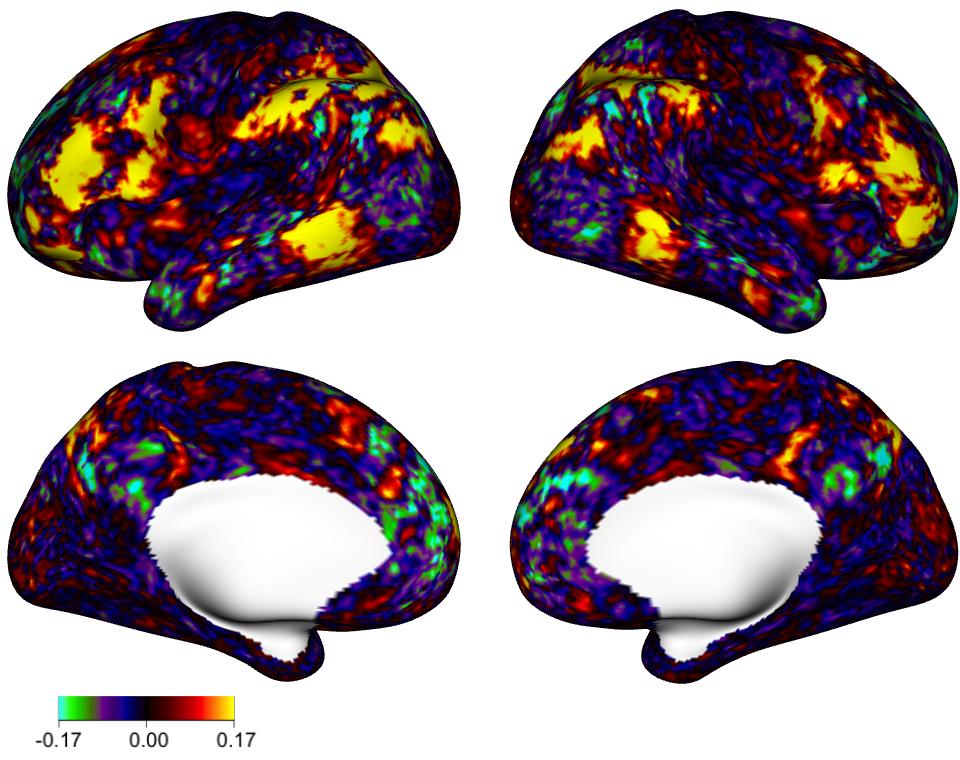
### 6.4 Visualize the Results

We now plot:

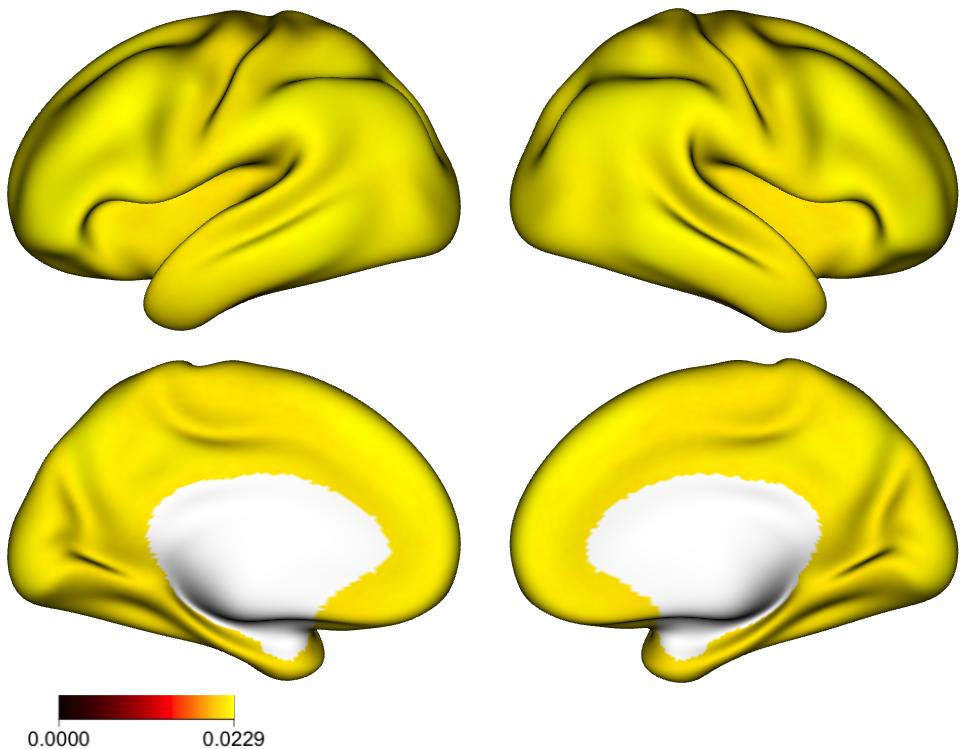
1. The subject-level networks estimated by `BrainMap()` (both mean and standard error).
2. The engagement maps, showing regions of deviation from the prior mean.

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

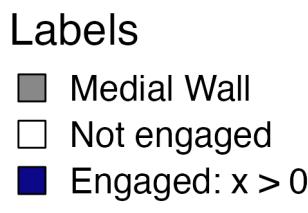
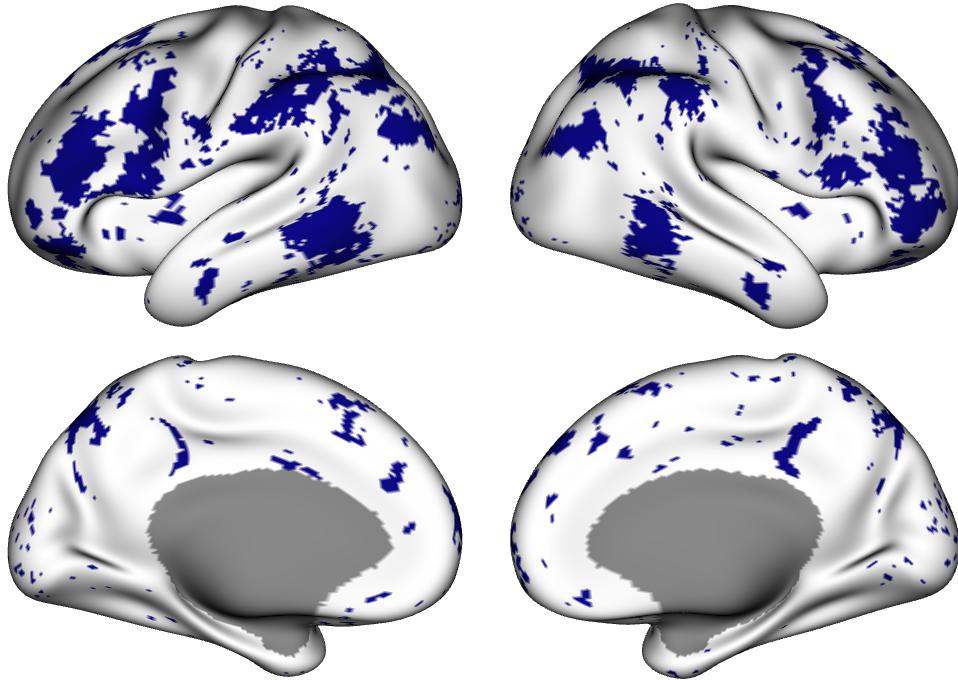
Yeo 17 Network ContA (#11) Subject-Level Mean



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



Yeo 17 Network ContA (#11) Engagement Map



## 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"))
```

## Appendix B: Subject Filtering

### 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 `fMRIscrub` 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 the `combined` 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.

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

### Preview of FD Summary Table

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

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 < 0.2).

### 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"))

```

### 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"))

```

### 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:

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

## 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:

\*\* todo: visualize better in pdf \*\*

```

# 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)

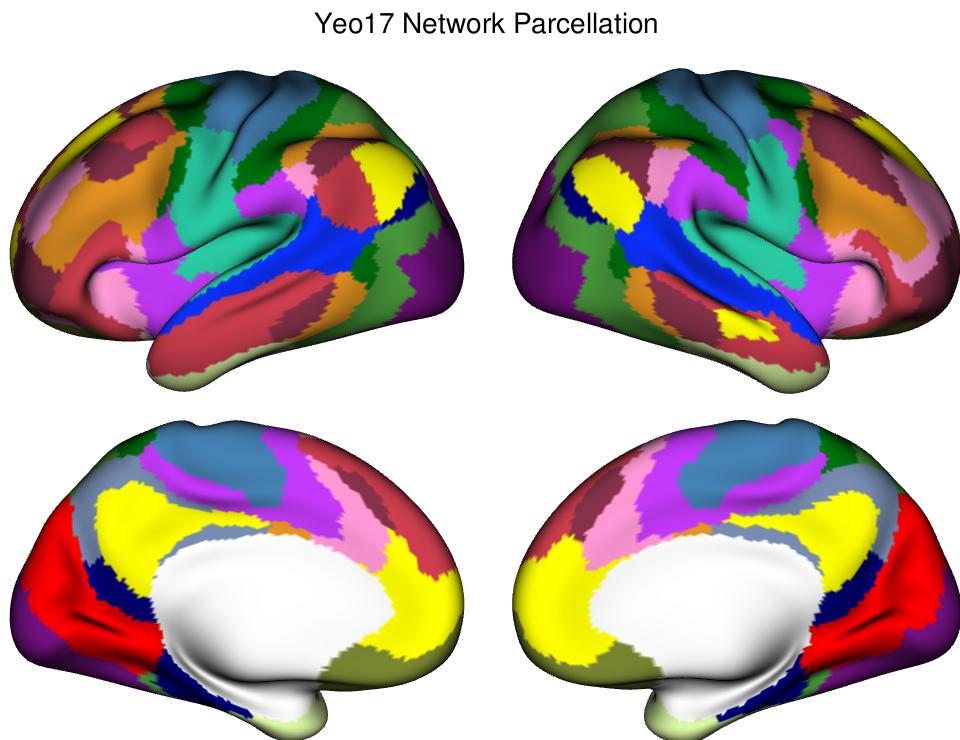
view_xifti_surface(
  xifti = yeo17,
  widget = TRUE,
  title = "Yeo17 Network Parcellation",
  legend_ncol = 6,
  legend_fname = file.path(dir_data, "outputs", "parcellations_plots",

```

```

    "Yeo17", "Yeo17_legend.png"),
  fname=file.path(dir_data, "outputs", "parcellations_plots", "Yeo17")
)

```



Labels									
□ ???	SomMotA	DorsAttnB	LimbicA	ContB	DefaultB				
■ VisCent	SomMotB	SalVentAttnA	LimbicB	ContC	DefaultC				
■ VisPeri	DorsAttnA	SalVentAttnB	ContA	DefaultA	TempPar				

## 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.