

BayesianBrainMapping-Templates

authors

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 (CITE 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.

- Step 1. Training a population-derived prior (mean and variance) for each network using a training dataset.
- Step 2. Fitting the Bayesian Brain Mapping model to fMRI data from an individual subject, using those priors.

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 clone the repository to your local machine or cluster:

```
# git clone https://github.com/mandy whole/BayesianBrainMapping-Templates.git  
# cd BayesianBrainMapping-Templates
```

Next, download the required data/ folder from the following OSF link and place it inside the cloned repository:

https://osf.io/n3wk5/?view_only=0d95b31090a245eb9ef51fe262be60ef

Once downloaded, unzip the contents so the folder structure looks 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:

[@Nohelia: Can you briefly describe the role of each directory below, like you did for the HCP csv and `wb_path`?]

- `dir_project`
- `dir_data`
- `dir_results`
- `dir_personal`
- `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"))
```

```
## Using this Workbench path: '/Users/nohelia/Downloads/workbench/bin_macosxub/wb_command'.
```

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 [@Nohelia, insert starting number], we apply the following filtering steps:

3.1 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. [@Nohelia, for template estimation, did we truncate the scan duration at 10 min? And was scrubbing implemented in template estimation?]

The final subject list includes only those who passed the filtering criteria in both LR and RL encodings. [@Nohelia -- and for both visits also?] This list is referred to as the `combined` list and is the one used throughout this project.

```
#@Nohelia please comment here very briefly what this script does, what/where it writes out
# 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("~/Documents/StatMIND/Data/fd_summary.csv")
#@Nohelia is there a reason this is saved locally and not in the Github? That would be more convenient
#so people running this have access to it. We can share it, so that they can still use/view it even if
#don't run the script.

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

Preview of FD Summary Table

Table 1: 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`).

3.2 Filter Unrelated Subjects

Building on the previous step, we use the HCP restricted demographic data to exclude related individuals. [@Nohelia very important that we don't share the list of final subjects included, because that could reveal which subjects are related. Can you confirm we do not share that information?] This step helps ensure the statistical independence of subjects in the group-level template estimation.

For the `combined` list 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.

```
#@Nohelia please comment here very briefly what this script does, what/where it writes out
#source(file.path(src_dir, "2_unrelated_filtering.R"))
```

3.3 Filter Subjects to Balance Sex Within Age Groups

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

For the `combined` list of valid and unrelated subjects, 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 you are not applying the unrelated subject filtering step (3.2), you can modify the code to subset based on `valid_combined_subjects_FD` instead of `valid_combined_subjects_unrelated`.

The final list of valid subjects is saved [`@Nohelia -- where?`] as:

- `valid_combined_subjects_balanced.csv`
- `valid_combined_subjects_balanced.rds` (used in the template estimation step)

```
#@Nohelia please comment here very briefly what this script does, what/where it writes out
#source(file.path(src_dir, "3_balance_age_sex.R"))
```

4. Prepare Group-Level Parcellations

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`.

```
#@Nohelia please comment here very briefly what this script does, what/where it writes out
# source(file.path(src_dir, "4_parcellations.R"))
```

[Mandy is here]

We can visualize the Yeo17 networks and their corresponding labels:

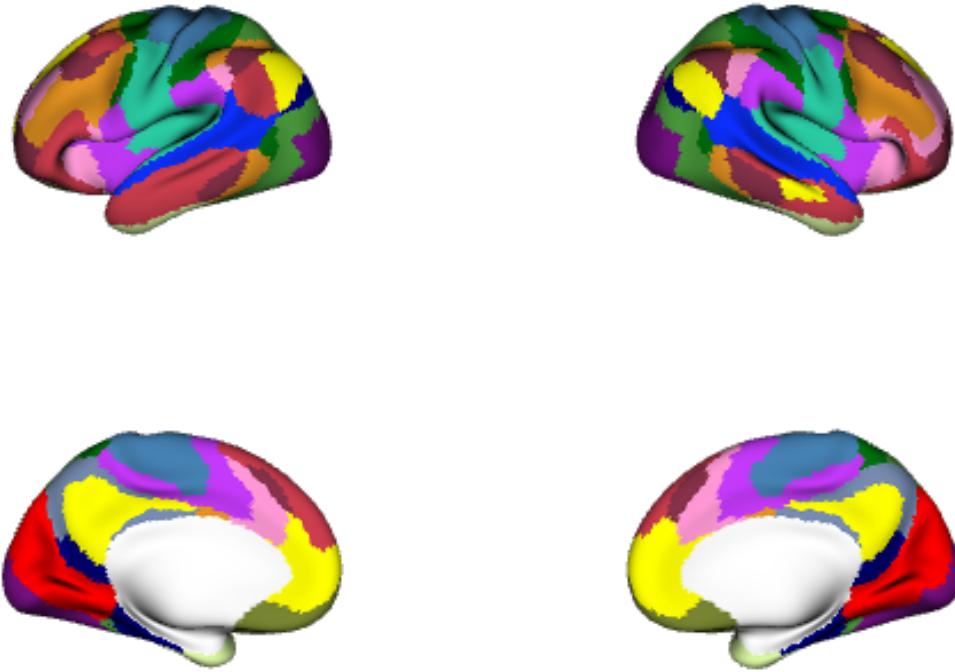
```
# Load libraries
library(ciftiTools)
library(rgl)
rgl::setupKnitr()

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

view_xifti_surface(
  xifti = yeo17,
  widget = TRUE,
  title = "Yeo17 Network Parcellation",
  legend_ncol = 6,
  legend_fname = "yeo17_legend.png",
)

## Warning in snapshot3d(scene = x, width = width, height = height): webshot =
## TRUE requires the webshot2 package and Chrome browser; using rgl.snapshot()
## instead
```

Yeo17 Network Parcellation



```
# Show legend
knitr::include_graphics("yeo17_legend.png")
```

Labels														
<input type="checkbox"/> ???	<input type="checkbox"/> SomMotA	<input type="checkbox"/> DorsAttnB	<input type="checkbox"/> LimbicA	<input type="checkbox"/> ContB	<input type="checkbox"/> DefaultB									
<input type="checkbox"/> VisCent	<input type="checkbox"/> SomMotB	<input type="checkbox"/> SalVentAttnA	<input type="checkbox"/> LimbicB	<input type="checkbox"/> ContC	<input type="checkbox"/> DefaultC									
<input type="checkbox"/> VisPeri	<input type="checkbox"/> DorsAttnA	<input type="checkbox"/> SalVentAttnB	<input type="checkbox"/> ContA	<input type="checkbox"/> DefaultA	<input type="checkbox"/> TempPar									

5. Estimate Templates

In this step, we estimate group-level statistical templates using the `estimate_template()` function from the `templateICAr` package.

The helper function `estimate_and_export_template()` wraps the full procedure, handling subject selection, BOLD file path construction, parcellation selection, and output saving.

The encoding parameter is set to `combined` to use the final list of subjects saved in Step 3.3 (`valid_combined_subjects_balanced.rds`), which includes individuals who passed motion filtering in both LR and RL directions, were unrelated, and were sex-balanced within age groups.

For these subjects, we include REST1 sessions only, but from both encodings:

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

The nIC parameter determines which parcellation is used:

- If nIC = 15, 25, or 50, the corresponding HCP GICA parcellation (GICA_15IC.dscalar.nii, etc.) is used from the data/ folder. These files were downloaded from the HCP website, specifically from the CIFTI Subject-specific ICA Parcellations dataset for 15-, 25-, 50-, and 100-dimensions.

Templates are saved as .rds files. For GICA-based runs, additional outputs are exported using `export_template()`.

In total, we estimate 8 templates varying the parcellation type and GSR inclusion:

- GICA parcellation with 15 components: GSR = TRUE / FALSE
- GICA parcellation with 25 components: GSR = TRUE / FALSE
- GICA parcellation with 50 components: GSR = TRUE / FALSE
- Yeo 17-network parcellation: GSR = TRUE / FALSE

```
# Load function
# source("5_estimate_template.R")
```

5.1 Example Run on 2 subjects

Running `estimate_template()` on the full "combined" subject list (~350 subjects) takes approximately 27 hours and uses 135 GB of memory. These templates were estimated on Quartz, a high-performance computing cluster.

To illustrate the process, we demonstrate a minimal example using 2 subjects, with:

- The GICA 15-component parcellation
- GSR = TRUE
- REST1 session from both LR and RL encodings

```
# test_subjects <- c("100307", "100206")
#
# BOLD_paths1 <- file.path("~/Desktop/", test_subjects, "rfMRI_REST1_LR_Atlas_MSMAll_hp2000_clean.dtseries.nii")
#
# BOLD_paths2 <- file.path("~/Desktop/", test_subjects, "rfMRI_REST2_LR_Atlas_MSMAll_hp2000_clean.dtseries.nii")
#
# GICA <- read_cifti(file.path(dir_data, "GICA_15IC.dscalar.nii"))
#
# template <- estimate_template(
#   BOLD = BOLD_paths1,
#   BOLD2 = BOLD_paths2,
#   GSR=TRUE,
#   TR = 0.72,
#   hpf = 0.01,
```

```

#      Q2 = 0,
#      Q2_max = NULL,
#      verbose=TRUE,
#      GICA=GICA
#  )

```

Note: This example is for demonstration purpose only. Estimating templates with so few subjects may produce unstable results, but it is useful for visualization, debugging, and understanding the pipeline.

6. Visualization

In this section, we visualize both the parcellation maps and the template outputs (mean and variance) for each parcellation scheme used in the study: Yeo17, 15 IC, 25 IC, and 50 IC. We also visualize their corresponding functional connectivity (FC) templates.

6.1 Generate and Save Parcellation Visualizations

6.1.1 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_simplified_mwall.rds`, created in Step 4.

Images are saved in `data/parcellations_plots/Yeo17`.

6.1.2 GICA Parcellations (15, 25, 50 ICs) Script: 9_visualization_GICAparcellations.R

This script defines a helper function that 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., `GICA_15_IC1.png`)
- A subcortical view (e.g., `GICA_15_IC1_sub.png`)

The resulting images are saved in the following folders:

- `data/parcellations_plots/15IC/`
- `data/parcellations_plots/25IC/`
- `data/parcellations_plots/50IC/`

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

6.2 Visualize Template ICA Components

Script: 6_visualization_template.R

TODO: where to upload the `template_rds` (too big for github)

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

For each parcellation type, we display:

- The first and last parcellation map
- The first and last mean map
- The first and last SD map

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

```
data/templates/15IC/GSR=F/
data/templates/25IC/GSR=T/
data/templates/50IC/GSR=F/
data/templates/yeo17/GSR=T/
```

These visualizations provide a detailed look at the spatial distribution and variability of each ICA component across the brain.

6.3 Visual Summary of Templates

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

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_GICAparcellations.R
- 6_visualization_template.R

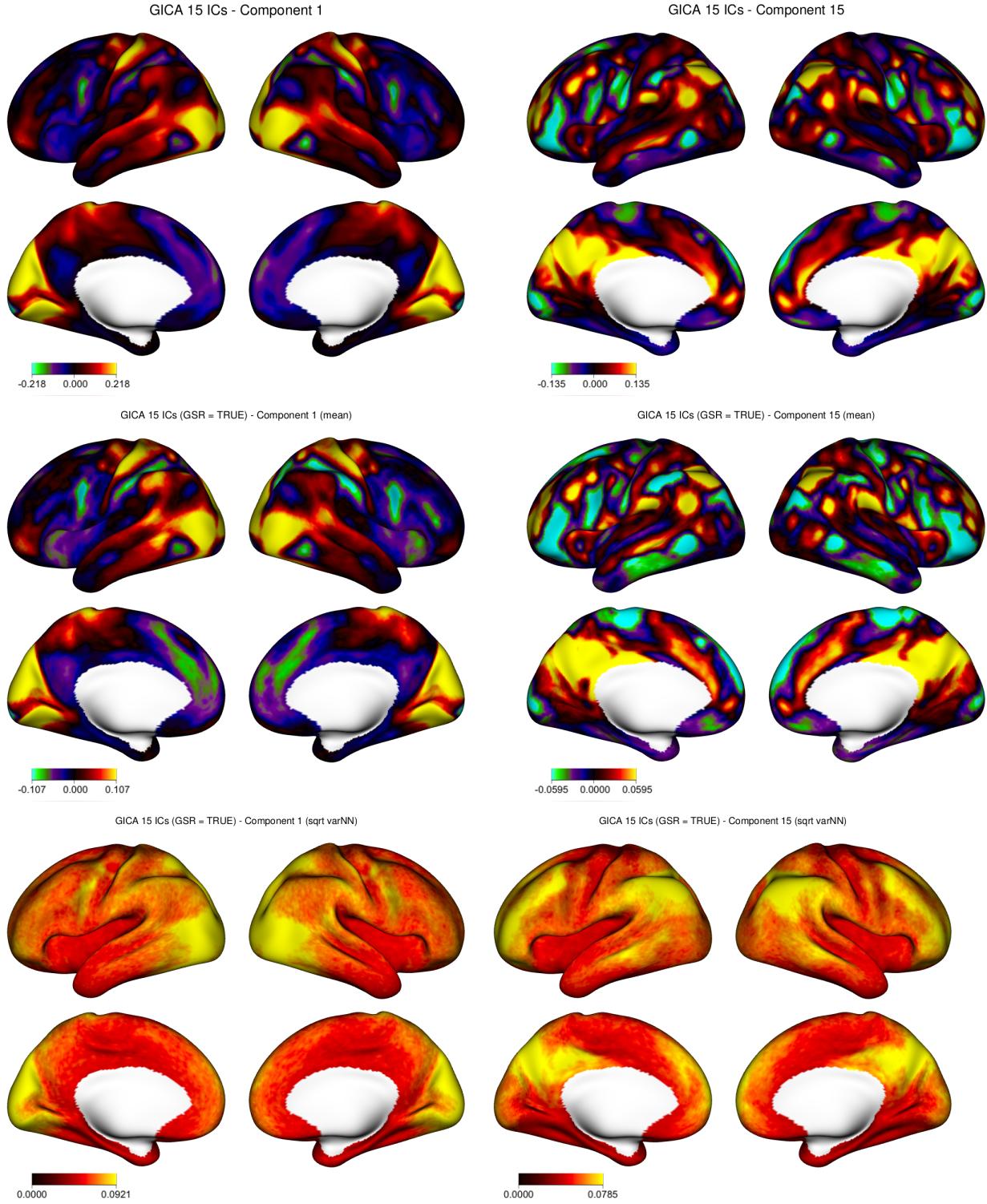
6.3.1 15 ICs For the 15 IC parcellation, we show visualizations of the first and last components (IC 1 and IC 15):

TODO: dont do plot but add image instead? regenerate them with titles

```
knitr:::include_graphics(file.path(dir_data, "parcellations_plots", "15IC", "GICA_15_IC1.png"))
knitr:::include_graphics(file.path(dir_data, "parcellations_plots", "15IC", "GICA_15_IC15.png"))

knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "15IC", "GSR=T", "template_com
knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "15IC", "GSR=T", "template_com

knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "15IC", "GSR=T", "template_com
knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "15IC", "GSR=T", "template_com
```



6.3.2 25 ICs For the 25 IC parcellation, we show visualizations of the first and last components (IC 1 and IC 25):

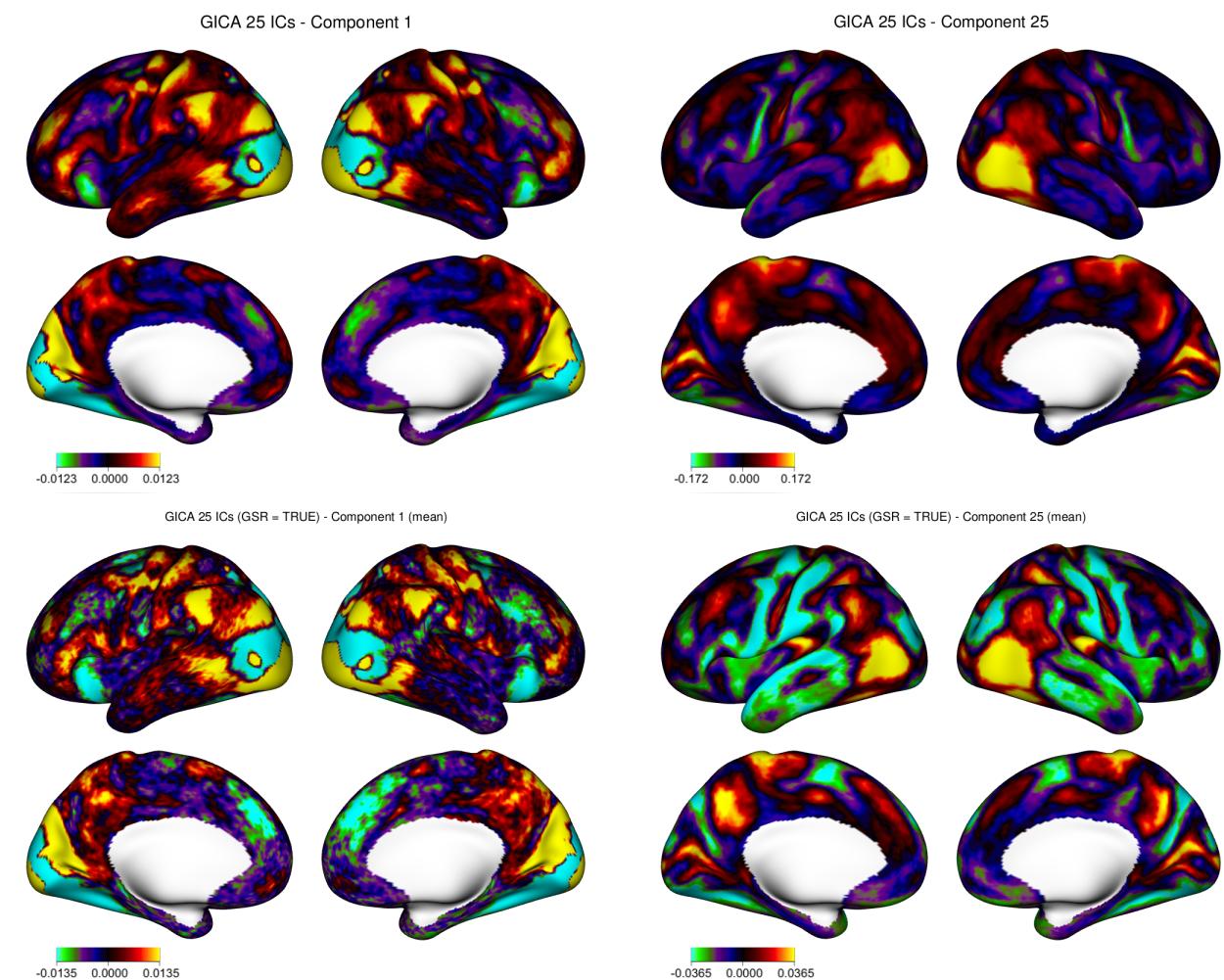
```

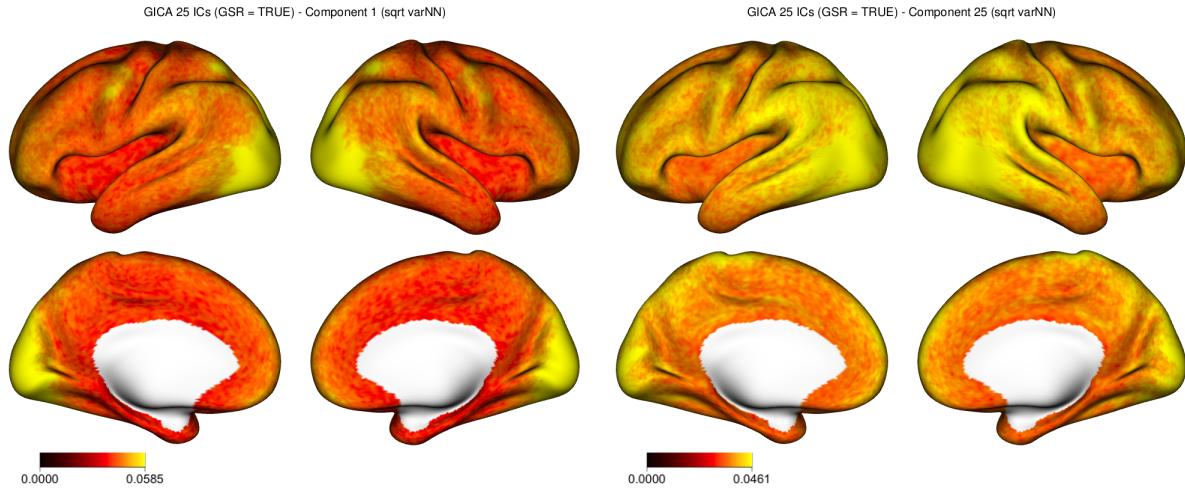
knitr:::include_graphics(file.path(dir_data, "parcellations_plots", "25IC", "GICA_25_IC1.png"))
knitr:::include_graphics(file.path(dir_data, "parcellations_plots", "25IC", "GICA_25_IC25.png"))

knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "25IC", "GSR=T", "template_com
knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "25IC", "GSR=T", "template_com

knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "25IC", "GSR=T", "template_com
knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "25IC", "GSR=T", "template_com

```



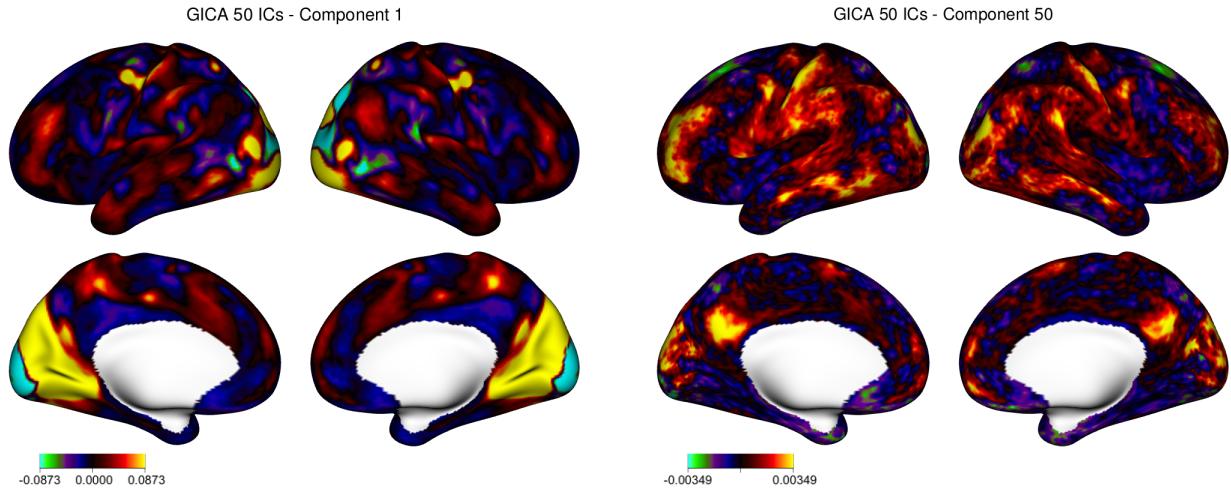


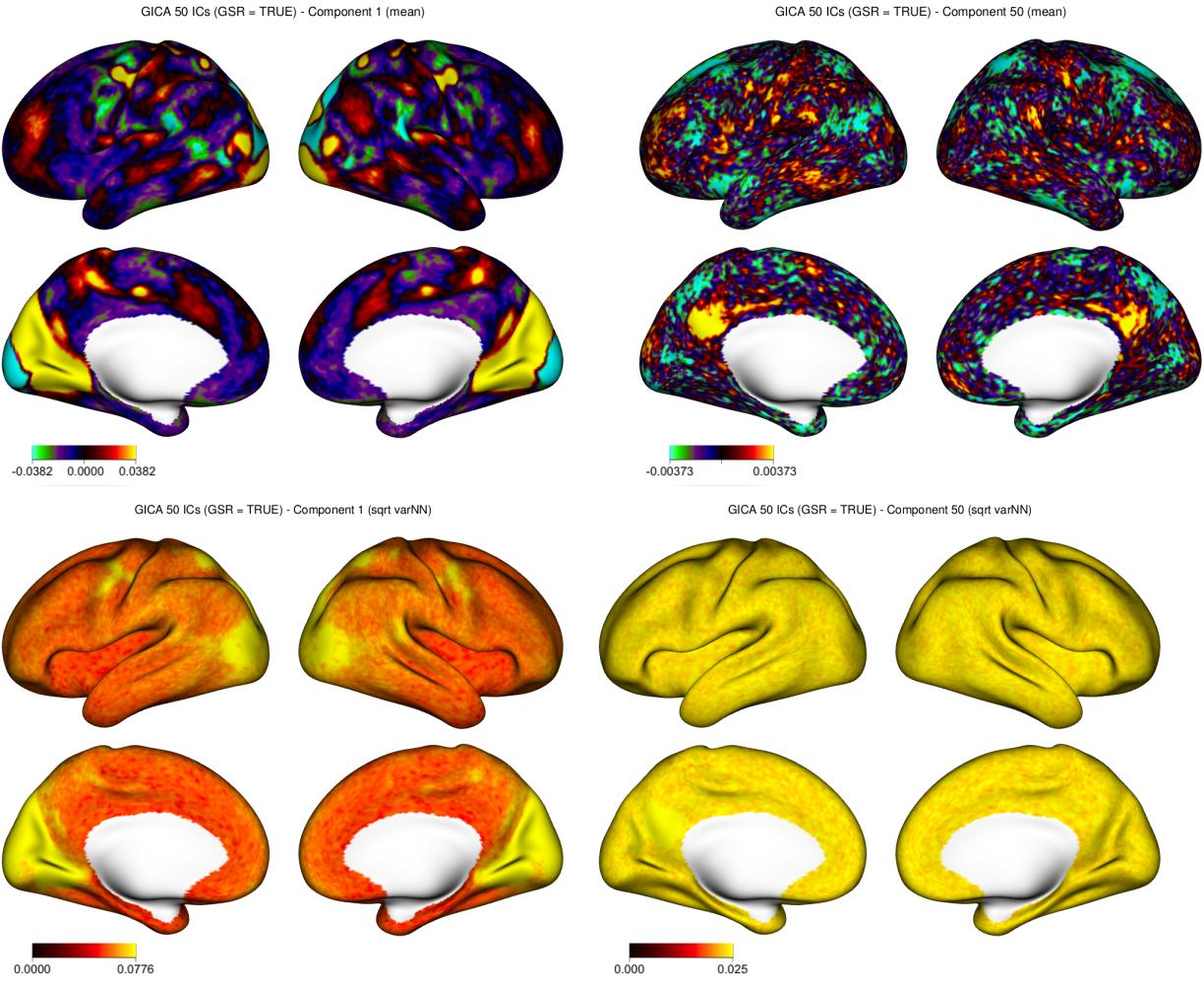
6.3.3 50 ICs For the 50 IC parcellation, we show visualizations of the first and last components (IC 1 and IC 50):

```
knitr::include_graphics(file.path(dir_data, "parcellations_plots", "50IC", "GICA_50_IC1.png"))
knitr::include_graphics(file.path(dir_data, "parcellations_plots", "50IC", "GICA_50_IC50.png"))

knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "50IC", "GSR=T", "template_com...
knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "50IC", "GSR=T", "template_com...

knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "50IC", "GSR=T", "template_com...
knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "50IC", "GSR=T", "template_com...
```





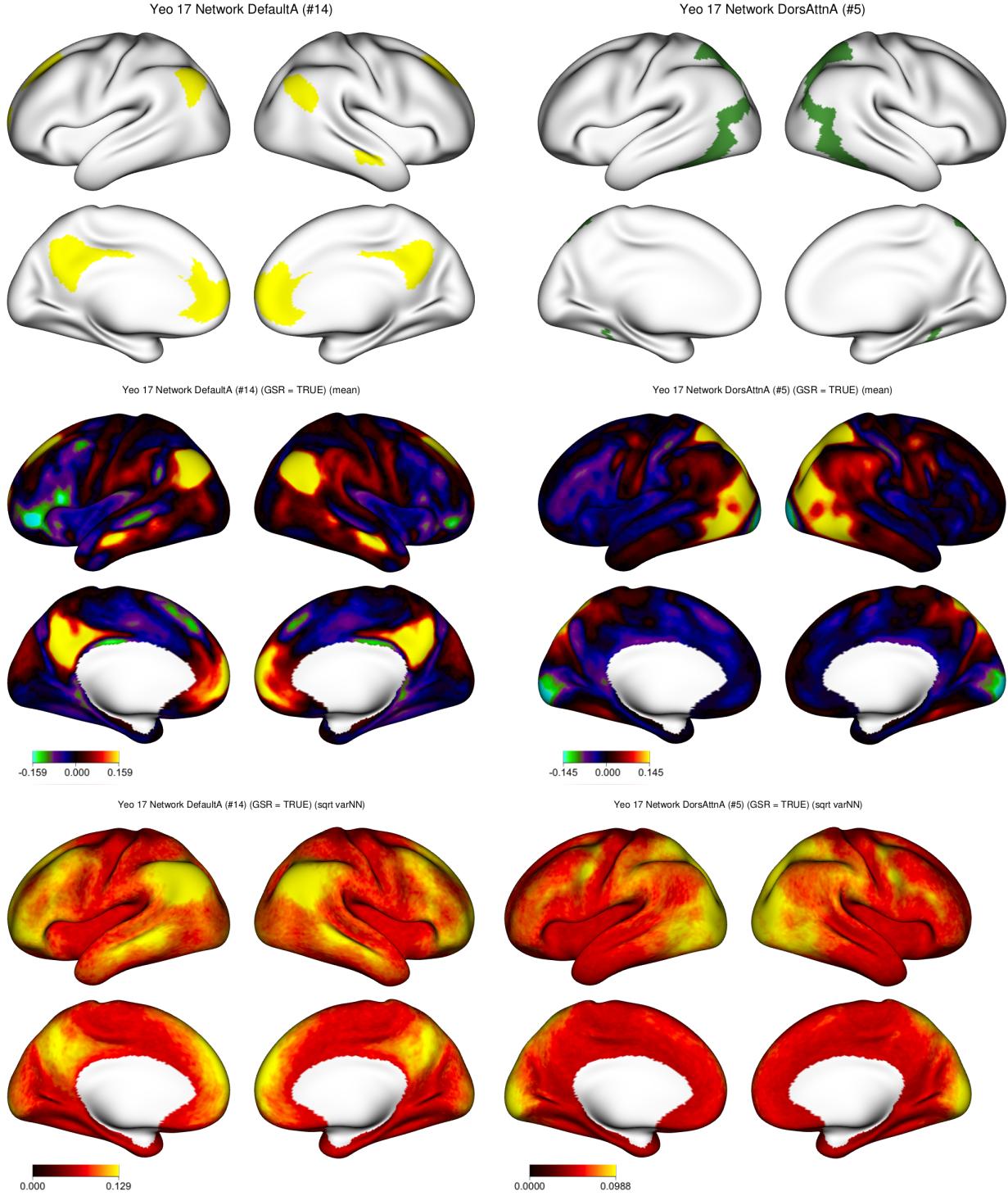
6.3.4 Yeo17 For the Yeo17 parcellation, we show visualizations of the two main networks (DefaultA and DorsAttnA):

TODO: ADD NAMES TO PARCELLATION PLOTS NO NUMBERS (ALSO LOWER CASE)

```
knitr:::include_graphics(file.path(dir_data, "parcellations_plots", "Yeo17", "Yeo17_DefaultA.png"))
knitr:::include_graphics(file.path(dir_data, "parcellations_plots", "Yeo17", "Yeo17_DorsAttnA.png"))

knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "Yeo17", "GSR=T", "template_co
knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "Yeo17", "GSR=T", "template_co

knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "Yeo17", "GSR=T", "template_co
knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "Yeo17", "GSR=T", "template_co
```



6.4 Visualize Template Functional Connectivity

Script: `7_visualization_FC.R`

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

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

For each template, the following outputs are saved in the `data/template_plots/combined/FC` folder:

PDF files (2 per template)

- `[template_name]_FC_Cholesky.pdf`
- `[template_name]_FC_InverseWishart.pdf`

Each PDF includes:

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

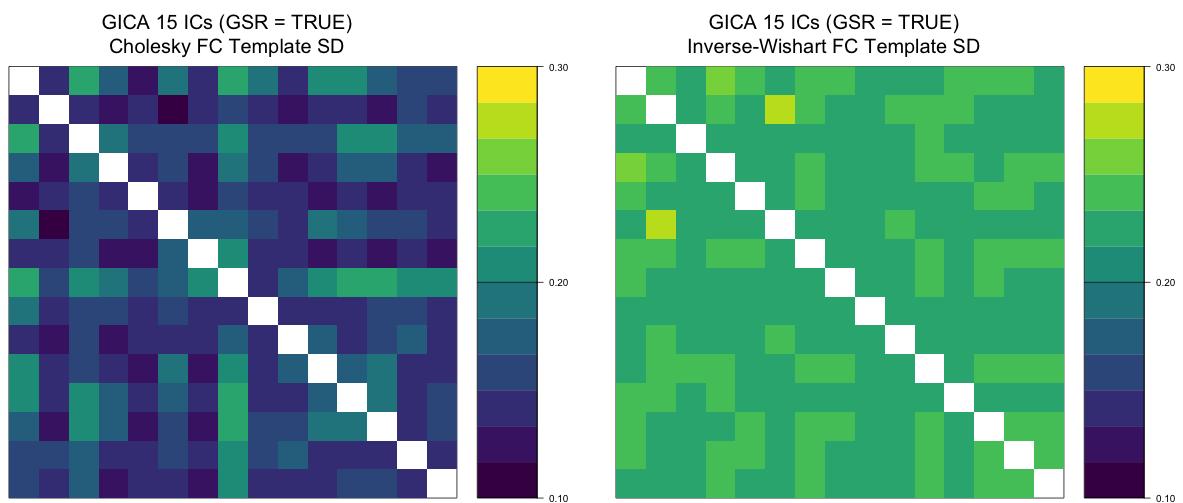
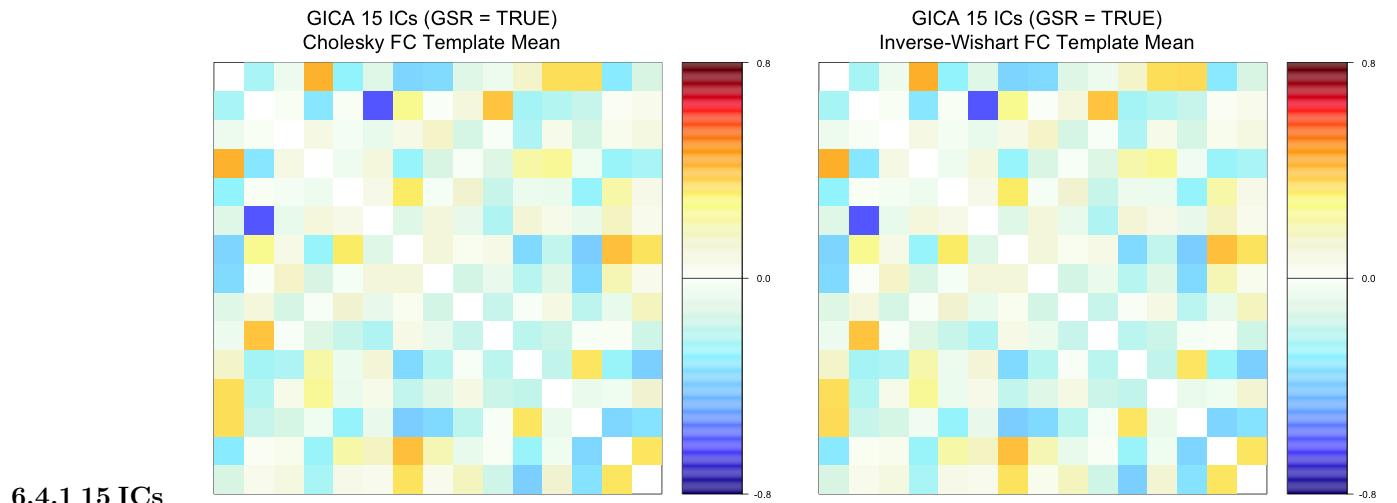
PNG images (4 per template)

- `[template_name]_FC_Cholesky_mean.png`
- `[template_name]_FC_Cholesky_sd.png`
- `[template_name]_FC_InverseWishart_mean.png`
- `[template_name]_FC_InverseWishart_sd.png`

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

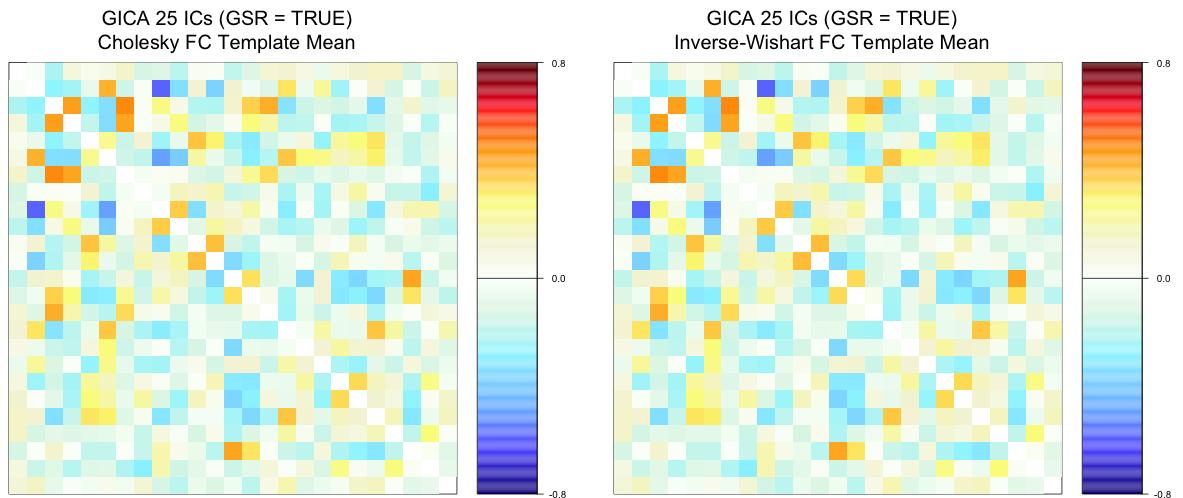
```
knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_15IC")
knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_15IC")

knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_15IC")
knitr:::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_15IC")
```

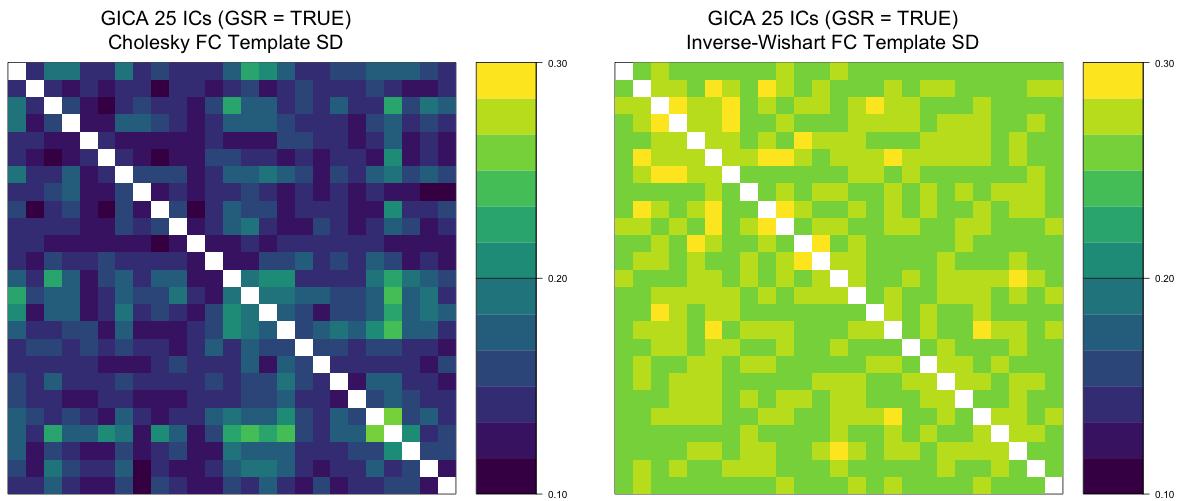


```
knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_25ICs"))
knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_25ICs"))

knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_25ICs"))
knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_25ICs"))
```



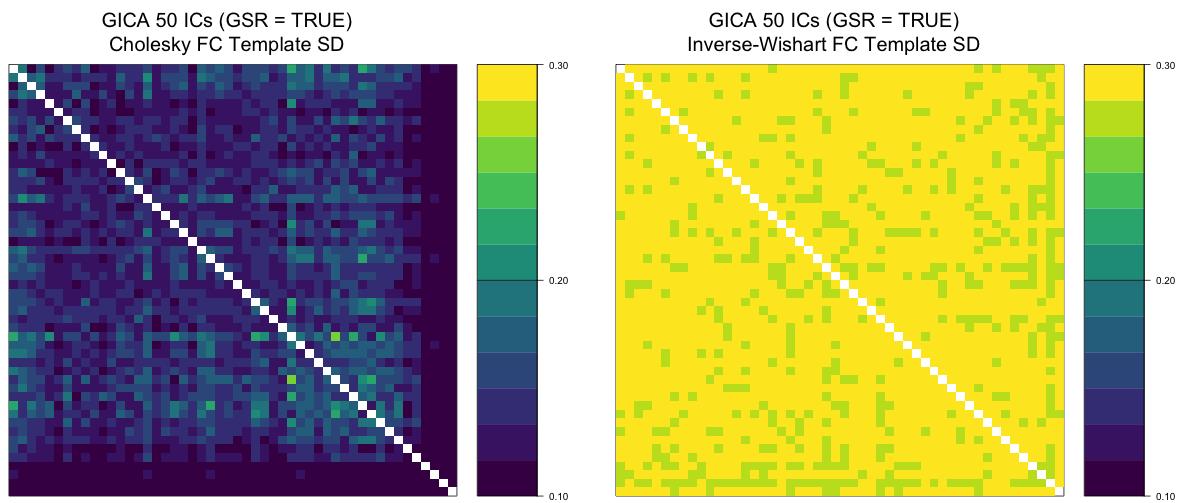
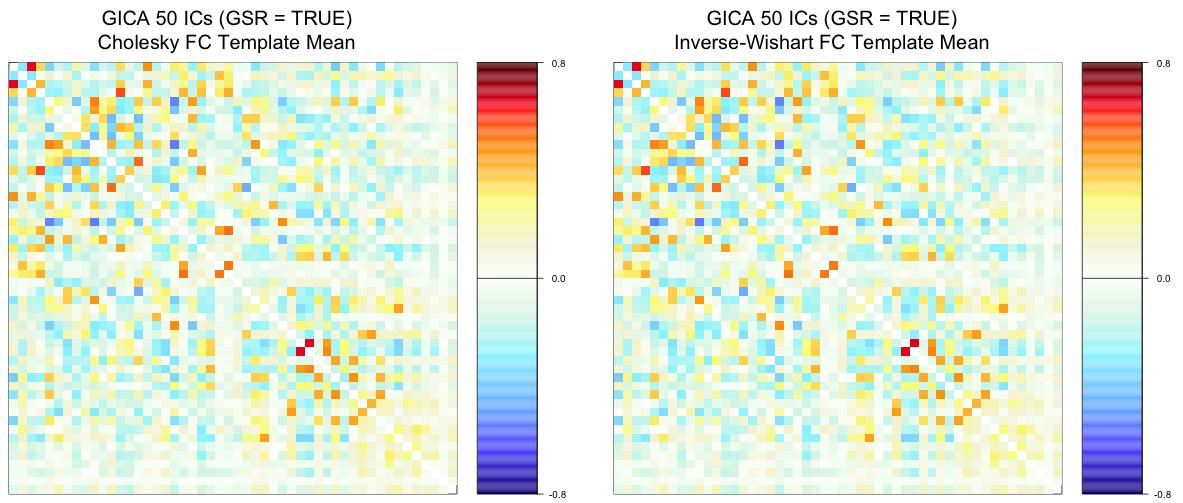
6.4.2 25 ICs



```
knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_50IC")
knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_50IC"))

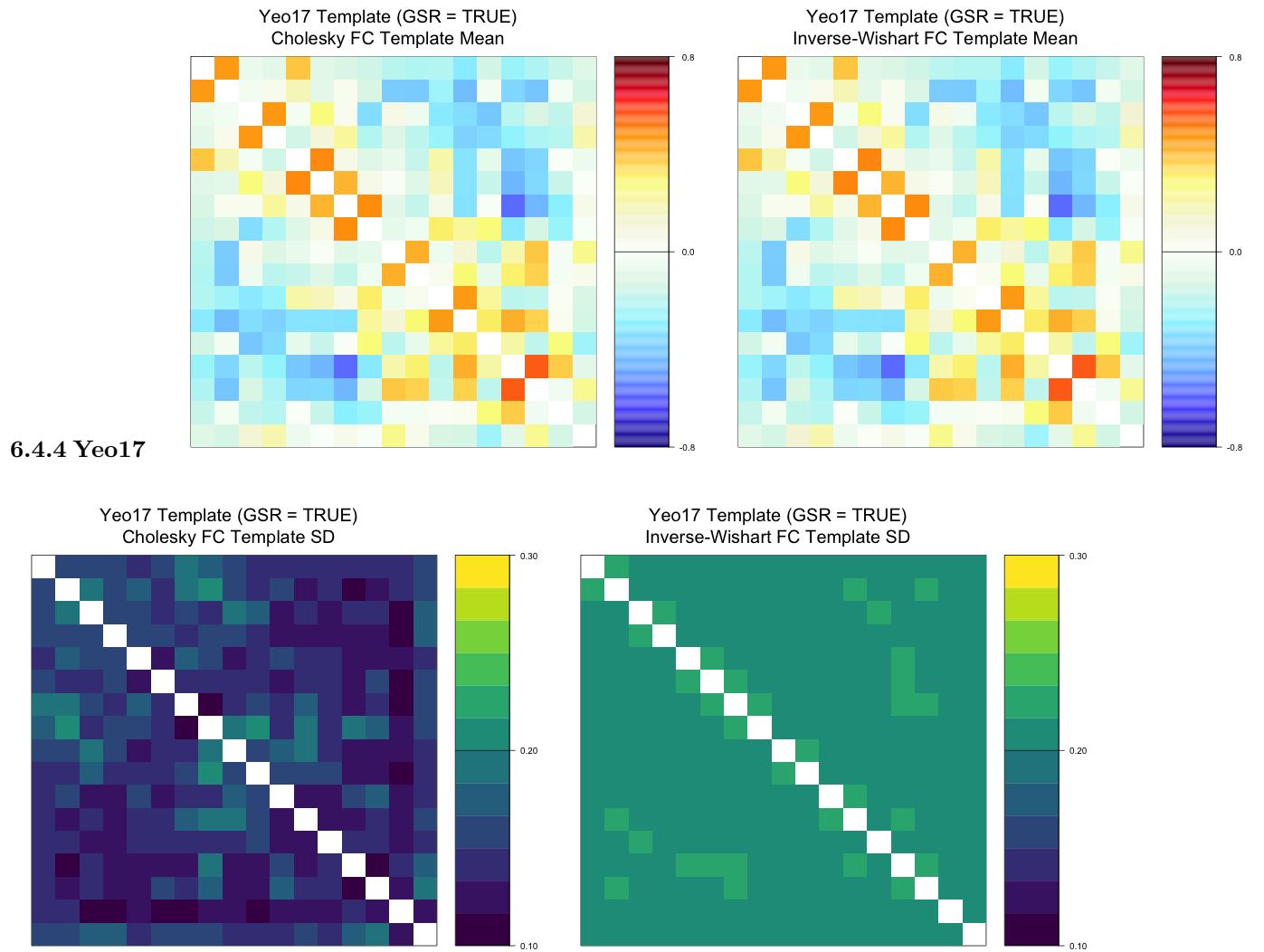
knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_50IC")
knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_50IC"))
```

6.4.3 50 ICs



```
knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_yeo1"))
knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_yeo1"))

knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_yeo1"))
knitr::include_graphics(file.path(dir_data, "template_plots", "combined", "FC", "template_combined_yeo1"))
```



7. Group-Level Analysis (individual?)

example on how to use the templates TEMAPLATE ICA VISUALIZATION