IMCO 2021 Project - Age, Sex, and Accuracy

Erica Baller

7/28/2021

#install.packages("knitr")  
#library(knitr)  
knitr::opts\_chunk$set(echo = TRUE)  
  
### Set up your paths  
#homedir = '/Users/eballer/BBL/imco/pmacs/PMACS\_remote'  
homedir = '/project/imco'  
mydirec = '/baller'  
subListPath = '/subjectLists'  
scriptsPath = '/scripts'  
  
## include external functions  
source(paste0(homedir, mydirec, scriptsPath, '/imco\_functions.R'))

## Loading required package: visreg

## Loading required package: mgcv

## Loading required package: nlme

## This is mgcv 1.8-33. For overview type 'help("mgcv-package")'.

## Loading required package: tableone

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:nlme':  
##   
## collapse

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## IMCO Project

Welcome to IMCO 2021! This project is a culmination of many years of work and a beautiful collaboration between Ted Satterthwaite and Taki Shinohara. A big thank you to the incredible support from Azeez Adebimpe, Bart Larsen for advising on modeling, and Adam Pines for his tireless help with taking my graphics to the next level. Thank you to Val Sydnor and Jakob Seidlitz for providing maps for our fancier visualization. In the remaining part of this document, I will take you on a step-by-step journey through the analysis, code, and visualizations.

Git for this project can be found at: <https://github.com/PennLINC/IntermodalCoupling>

\*\* Notes before you get started \*\*

Please make sure you have appropriately specified your home directory as well as all supporting directories. File system paths will be specific to each person’s specific directory structure/cluster.

### IMCO Analysis overview

IMCO, or intermodal coupling (or “coupling”), was a project that begun in 2018. It is based on work by Simon ’s work in 2016 (Paper: Vandekar, S. N., Shinohara, R. T., Raznahan, A., Hopson, R. D., Roalf, D. R., Ruparel, K., … Satterthwaite, T. D. (2016). Subject-level Measurement of Local Cortical Coupling. NeuroImage, 133, 88–97.), which looked at the relationship between cortical thickness and sulcal depth, and demonstrated regional differences, non-linear trajectories, as well as sex differences. His work used surface gray matter maps. The general process was the brain was divided into vertices and a neighborhood around each vertex was defined. For each neighborhood, the relationship between cortical thickness and sulcal depth was calculated using local weighted regression, with a variety of covariates.

The steps for coupling are as follows -

1. Sample construction
2. Taking each subject’s alff and cbf voxelwise maps and projecting to the surface
   * In fsaverage5 space (10242 vertices per hemisphere). Image type: .mgh
3. Running the locally weighted regression to obtain 1 coupling map per hemisphere per person (each person will end up with a left and right hemisphere map). Image type: .mgh
4. Converting these .mgh files to .asc
5. Reading in the .asc files into R, as well as demographic and accuracy information, and running coupling ~ age, sex, motion, and accuracy regressions
6. Correcting and masking these findings for display
7. Displaying in matlab
8. Spin testing mean coupling, gam age, sex and accuracy results to test whether # vertices per yeo network is significantly different in these analysis
9. Plotting the spin test results

For reference, I have included the wikis used to make the coupling maps found below.

##### Part 1 – Coupling Map Generation

**Relevant Source Code and Coupling Wikis:**

1. Volume-to-surface Projection: <https://github.com/PennBBL/tutorials/wiki/3D-Volume-to-Surface-Projection-(FS)>
2. how to generate 2D coupling maps: <https://github.com/PennBBL/tutorials/wiki/Surface-Coupling>
3. Simon Vandekar’s Coupling Map Generation (with setup instructions): <https://bitbucket.org/simonvandekar/coupling/src/master/>

\*\*\* Be aware that there are a few steps left out of these wikis, including some manual moving of files from one format to another that makes running this coupling in series impossible – it really depends on your file system.

Step 1) Surface projection:

*script: vol2surf.sh*

* + Surface projection scripts (vol2surf)
  + One step to consider adding to the end of the projection scrips is a change in the names of the files. Coupling needs files names rh.cbf rather than rh.cbf.mgh (which is what comes out of surface projection)
  + To accomplish this, a simple: cp rh.cbf.mgh rh.cbf works just fine
* Step 2) Coupling (to write to a user specified directory rather than frozen) \*all done in bash, this code can be directly copied):
  + *script:* coupling\_v2.R
* Step 3) Make sure you change these files in coupling\_v2.R (example included here)

## check these specifications are correct  
 # -m MEASURES cbf,alff -   
 # -t fsaverage5  
 # -fwhm 15  
  
 ##make these directory changes   
 scripts="/data/joag/aadebimpe/surfCoupling/R" #this contains the coupling\_v2.R and kth\_neighbor\_v3.R, change full path for kth\_neightbor script  
 export SUBJECT\_DIR=/data/jag/aadebimpe/subjecdir #this is the directory you make

* Step 4) Change SUBJECT\_DIR in bash profile (example included here)

gedit ~/.bash\_profile  
export SUBJECT\_DIR=/data/jag/aadebimpe/subjecdir #this is the directory you make

##### Part 2 – Group Coupling Analyses

For this next section, I’ll walk you through sample construction, analysis, and visualization.

**Sample Construction**

*script: n831\_alff\_cbf\_makeSample.R*

Code:

###########################################  
#### PMACS Adapted sample construction ####  
###########################################  
  
## Replicated LB sample construction from 2018 ##  
#### Date: 01/14/2021  
###### Author: Erica Baller  
  
### pre: pnc QA, health, demographics files:   
  
# 1) /project/imco/pnc/neuroimaging/asl/n1601\_PcaslQaData\_20170403.csv  
# 2) /project/imco/pnc/neuroimaging/rest/n1601\_RestQAData\_20170714.csv  
# 3) /project/imco/pnc/neuroimaging/t1struct/n1601\_t1QaData\_20170306.csv  
# 4) /project/imco/pnc/health/n1601\_health\_20170421.csv  
# 5) /project/imco/pnc/demographics/n1601\_demographics\_go1\_20161212.csv  
  
### post:   
  
# 1) final sample .csv [/project/imco/baller/subjectLists/n831\_alff\_cbf\_finalSample.csv]  
# 2) image order .csv [/project/imco/baller/subjectLists/n831\_alff\_cbf\_finalSample\_imageOrder.csv]  
  
### uses:  
  
## This script takes the PNC 1601 sample, and goes through a variety of exclusions to get the final n.   
# 1) removes subjects with poor QA data (t1Exclude, pcaslVoxelwiseExclude, restExcludeVoxelwise  
# 2) uses ltnExcludev2 for health - "Particiapants with no medical co-morbidity, abnormal brain structure on radiology read, or current use of psychoactive medications (Excludes those with medical rating 3/4, major incidental findings that distort anatomy, psychoactive medical and psych medications, inpatient hospitalizations)" 1=Yes; 0=No Calculated Field using data from goassess A few medications that were previously missed or not categorized in the right medical class were updated on 10/4/16 that resulted in version change.  
  
## It then factorizes race and sex, and adds a new column datexscanid - this is great because some subjects have multiple scans  
  
### dependencies: Any R should do. We used 3.2.5  
  
###########set homedir so you can use local or remote  
#1) set homedir before you run this  
#2) set my\_path to be your own path  
  
#homedir = '/Users/eballer/BBL/imco/pmacs/PMACS\_remote'  
#homedir = '/project/imco'  
#mydirec = '/baller/subjectLists'  
  
# Load in files   
pcaslQA <- read.csv(paste0(homedir, "/pnc/neuroimaging/asl/n1601\_PcaslQaData\_20170403.csv"))  
restQA <- read.csv(paste0(homedir, "/pnc/neuroimaging/rest/n1601\_RestQAData\_20170714.csv"))  
t1QA <- read.csv(paste0(homedir, "/pnc/neuroimaging/t1struct/n1601\_t1QaData\_20170306.csv"))  
healthQA <- read.csv(paste0(homedir, "/pnc/health/n1601\_health\_20170421.csv"))  
demos <- read.csv(paste0(homedir, "/pnc/demographics/n1601\_demographics\_go1\_20161212.csv"))  
  
#these two are premade, do not need to be done for r, but do need to be done for fsGLM, can be commented out  
datexscan <- read.csv(paste0(homedir, mydirec, subListPath, "/n1601\_bblid\_datexscanid.csv"))  
order <- read.csv(paste0(homedir, mydirec, subListPath, "/n831\_imageOrder.csv"))  
  
#merge  
df <- merge(t1QA, healthQA, by = "bblid")  
df <- merge(df, pcaslQA, by = c("bblid"))  
df <- merge(df, restQA, by = c("bblid", "scanid"))  
df <- merge(df, demos, by = "bblid")

## Warning in merge.data.frame(df, demos, by = "bblid"): column names 'scanid.x',  
## 'scanid.y' are duplicated in the result

df <- merge(df, datexscan, by = "bblid")

## Warning in merge.data.frame(df, datexscan, by = "bblid"): column names  
## 'scanid.x', 'scanid.y' are duplicated in the result

#do excludions  
df\_exclude <- df[which(df$t1Exclude == 0),] #n=1540, lost 61  
df\_exclude <- df\_exclude[which(df\_exclude$restExcludeVoxelwise == 0),] #n = 1090, lost 450  
df\_exclude <- df\_exclude[which(df\_exclude$pcaslVoxelwiseExclude == 0),] #n = 1036, lost 54  
final <- df\_exclude[which(df\_exclude$ltnExcludev2 == 0),] #n = 831, lost 205  
  
  
## Clean demos  
final$ageAtScan1 <- (final$ageAtScan1)/12  
final$sex <- as.factor(final$sex)  
final$race <- as.factor(final$race)  
final$race2 <- as.factor(final$race2)  
final <- subset(final, select = -(scanid.y))  
names(final)[names(final) == "scanid.x"] <- "scanid"  
  
# Get correct image order for GLM  
finalOrdered <- merge(order,final,by=c("bblid","datexscanid")) #only if you want to do this  
  
# Write out final demos for sample  
write.csv(final, paste0(homedir, mydirec, subListPath, "/n831\_alff\_cbf\_finalSample\_erica\_sampling.csv"), row.names=FALSE, quote = FALSE) #added  
write.csv(finalOrdered,paste0(homedir, mydirec, subListPath, "/n831\_alff\_cbf\_finalSample\_imageOrder\_erica\_sampling.csv"),row.names=FALSE, quote = FALSE)  
  
#LB's initial exclusions were a little more involved (more lines, but still accurate). I redid these for simplicity's sake, and for the ability to know how many we lost at each step. The original version is still present in the R script /project/imco/baller/scripts/from\_chead/n831\_alff\_cbf\_makeSample.R  
  
#for the analysis, we used the following subject lists -   
#/n831\_alff\_cbf\_finalSample\_sampling.csv"  
#n831\_alff\_cbf\_finalSample\_imageOrder\_sampling.csv

Demographics summary

make\_demographics\_table(finalOrdered)

##   
## level Overall   
## n 831   
## Age (mean (SD)) 15.62 (3.36)   
## Sex (%) Female 478 (57.5)   
## Male 353 (42.5)   
## Race (%) Caucasian 366 (44.0)   
## Non-caucasian 465 (56.0)   
## Maternal Ed (mean (SD)) 14.28 (2.48)

Mean Coupling results:

#set home directory, switch this depending on whether running from PMACS or from home directory  
#homedir <- "/Users/eballer/BBL/imco/pmacs/PMACS\_remote/"  
#homedir <- "/project/imco/"  
  
thresh = 3.09  
  
#read in maps  
lh\_mean\_map <- read.csv(paste0(homedir, "/baller/results/mean\_maps/n831\_lh.coupling\_coef\_alff\_cbf.fwhm15.fsaverage5.asc"), header = F)  
rh\_mean\_map <- read.csv(paste0(homedir, "/baller/results/mean\_maps/n831\_rh.coupling\_coef\_alff\_cbf.fwhm15.fsaverage5.asc"), header = F)  
  
#bind them together  
overall\_mean\_map <- rbind(lh\_mean\_map, rh\_mean\_map)  
  
#take the mean of the T values  
mean\_t <- mean(overall\_mean\_map$V1)  
  
#convert the T to p - one-sided  
p.value = pt(q=mean\_t, df=(length(overall\_mean\_map$V1) - 1), lower.tail = FALSE)  
print(p.value)

## [1] 0.0006555209

#threshold at 3.09 (p < 0.001)  
lh\_mean\_map\_thresh\_boolean <- ifelse(abs(lh\_mean\_map$V1) < thresh, 0, 1)  
lh\_mean\_map\_thresh <- lh\_mean\_map\_thresh\_boolean \* lh\_mean\_map$V1  
  
rh\_mean\_map\_thresh\_boolean <- ifelse(abs(rh\_mean\_map$V1) < thresh, 0, 1)  
rh\_mean\_map\_thresh <- rh\_mean\_map\_thresh\_boolean \* rh\_mean\_map$V1  
  
#save  
write.table(x=lh\_mean\_map\_thresh, file = paste0(homedir, "/baller/results/mean\_maps/n831\_lh.coupling\_coef\_alff\_cbf.fwhm15.fsaverage5\_THRESH\_3.09.csv"), quote = F, row.names = F, col.names = F)  
write.table(x=rh\_mean\_map\_thresh, file = paste0(homedir, "/baller/results/mean\_maps/n831\_rh.coupling\_coef\_alff\_cbf.fwhm15.fsaverage5\_THRESH\_3.09.csv"), quote = F, row.names = F, col.names = F)

**Coupling regressions**

This takes a really long time, and I recommend using some sort of cluster submission (as we have done here, bsub)

cd /project/imco/baller/scripts/coupling  
xbash  
source coupling\_accuracy\_fx\_T\_bsub.sh

*script: /project/imco/baller/scripts/coupling/coupling\_accuracy\_fx\_T\_bsub.sh*

bsub -o /project/imco/baller/scripts/coupling/logfiles/outputlogjob.out -e /project/imco/baller/scripts/coupling/logfiles/outputlogjob.error -R "rusage[mem=128G]" < /project/imco/baller/scripts/coupling/coupling\_accuracy\_fx\_T\_wrapper.sh

*script: /project/imco/baller/scripts/coupling/coupling\_accuracy\_fx\_T\_wrapper.sh*

#!/bin/bash  
echo LSB\_JOB\_REPORT\_MAIL=N >> ~/.bashrc  
Rscript coupling\_accuracy\_fx\_T.R

*script: /project/imco/baller/scripts/coupling/coupling\_accuracy\_fx\_T.R*

##################################################  
### Accuracy Scripts 02/09/2021 ###  
##################################################  
  
####################  
##### Summary ######  
####################  
  
#input: asc files, and pnc demographics, cnb, clinical files  
#output: 10242 length vector csvs with T and/or p values for vertex-wide regression  
#uses: goes vertex by vertex and does regression (coupling by age, sex, cognition, etc), pulls out T and p from these values and sticks it in a vector. The vector can then be used for visualization in matlab  
#dependencies: R (3.6.3 is my current default in pmacs)  
  
####################  
### Libraries ###  
####################  
  
library(mgcv)  
library(dplyr)  
library(ggplot2)  
library(visreg)  
  
#####################################################################################  
#### Makes the 831x10242 matrices, both left and right ####  
#####################################################################################  
  
#set home directory, switch this depending on whether running from PMACS or from home directory  
#homedir <- "/Users/eballer/BBL/imco/pmacs/PMACS\_remote/"  
homedir <- "/project/imco/"  
  
# read in demos  
subjDemos <- read.csv(paste0(homedir, "/baller/subjectLists/n831\_alff\_cbf\_finalSample\_imageOrder.csv"))  
  
#some verification preprocessing  
subjDemos$sex <- as.factor(subjDemos$sex)  
subjDemos$race <- as.factor(subjDemos$race)  
subjDemos$race2 <- as.factor(subjDemos$race2)  
  
#add osex category for use in gam later  
subjDemos$osex <- ordered(subjDemos$sex)  
  
#add psych bifactor scores  
psych <- read.csv(paste0(homedir, "/pnc/clinical/n1601\_goassess\_itemwise\_bifactor\_scores\_20161219.csv"), header = TRUE)  
  
#remove 4factorv2 from title  
names(psych) <-gsub("\_4factorv2", "", names(psych))  
  
#merge  
subjDemos <- merge(subjDemos, psych, by = "bblid")  
  
#cognitive data  
cog <- read.csv(paste0(homedir, "/pnc/cnb/n1601\_cnb\_factor\_scores\_tymoore\_20151006.csv"))  
accuracy <- subset(cog, select = c("bblid", "F1\_Exec\_Comp\_Res\_Accuracy"))  
  
#merge  
subjDemos <- merge(subjDemos, accuracy, by = "bblid")  
  
#drop the scanid.y  
subjDemos <- subset(subjDemos, select = -scanid.y)  
  
#rename scanid.x to scanid  
names(subjDemos) <- gsub("scanid.x", "scanid", names(subjDemos))   
  
#make list of bblid/scanid  
bblid\_scanid <- paste0(subjDemos$bblid, "\_", subjDemos$datexscanid)  
  
#####################  
##### Left Side #####  
#####################  
  
#initiate matrix for storage  
numrows <- dim(subjDemos)[1]  
lh\_matrix <- matrix(nrow = numrows, ncol = 10242)  
  
#go through each subject, grab 5th column in asc, transpose and stick in matrix  
# output is 831 x 10242 matrix  
for (subj in 1:831) {  
   
 bblid <- subjDemos$bblid[subj]  
 datexscanid <- subjDemos$datexscanid[subj]  
 file\_path <- paste0(homedir, "/couplingSurfaceMaps/alffCbf/lh/stat/", bblid, "\_", datexscanid, "\_lh.coupling\_coef\_alff\_cbf.fwhm15.fsaverage5.asc")  
 alff\_data <- read.table(file\_path, stringsAsFactors = FALSE)  
 lh\_matrix[subj,] <- t(alff\_data$V5)  
   
}  
  
#append with demographics  
subjDemos\_with\_lh\_matrix <- cbind(subjDemos, lh\_matrix)  
  
#write output  
write.table(subjDemos\_with\_lh\_matrix, file = paste0(homedir, "/baller/results/coupling\_accuracy/subjDemos\_with\_lh\_", numrows, "x10242.csv"), sep = ",")  
  
#####################  
#### Right Side #####  
#####################  
#initiate matrix for storage  
numrows <- dim(subjDemos)[1]  
rh\_matrix <- matrix(nrow = numrows, ncol = 10242)  
  
#go through each subject, grab 5th column in asc, transpose and stick in matrix  
# output is 831 x 10242 matrix  
for (subj in 1:831) {  
   
 bblid <- subjDemos$bblid[subj]  
 datexscanid <- subjDemos$datexscanid[subj]  
 file\_path <- paste0(homedir, "/couplingSurfaceMaps/alffCbf/rh/stat/", bblid, "\_", datexscanid, "\_rh.coupling\_coef\_alff\_cbf.fwhm15.fsaverage5.asc")  
 alff\_data <- read.table(file\_path, stringsAsFactors = FALSE)  
 rh\_matrix[subj,] <- t(alff\_data$V5)  
   
}  
  
#append with demographics  
subjDemos\_with\_rh\_matrix <- cbind(subjDemos, rh\_matrix)  
  
#write output  
write.table(subjDemos\_with\_rh\_matrix, file = paste0(homedir, "/baller/results/coupling\_accuracy/subjDemos\_with\_rh\_", numrows, "x10242.csv"), sep = ",")  
  
  
####-----------------------------------------------------------------------------####  
####---------------------------End of Part 1- Making matrices---\_----------------####  
####-----------------------------------------------------------------------------####  
  
#####################################################################################  
#### Run regression, both left and right ####  
#####################################################################################  
  
#make easier to reference names  
lh\_imco <- subjDemos\_with\_lh\_matrix #can also read directly from files if you'd like  
rh\_imco <- subjDemos\_with\_rh\_matrix  
  
  
#####################################################  
# lm/gams #  
#####################################################  
  
#initialize vectors for models  
  
hemis <- c("lh", "rh") #hemispheres  
models <- c("age", "sex", "exec\_accuracy")  
  
coeffs <- c("p", "t") #p or t value  
corrs <- c("uncor", "fdr") #correction  
  
for (hemi in hemis){  
 for (model in models) {  
 for (coeff in coeffs) {  
 for (corr in corrs) {  
 vector\_init\_cmd <- paste0(hemi, "\_gam\_", model, "\_", coeff, "\_", corrs, " <- vector(length = 10242)")  
 print(vector\_init\_cmd)  
 eval(parse(text=as.name(vector\_init\_cmd)))  
 }  
 }  
 }  
}  
  
  
#make linear models as well  
#lm\_models <- c("age", "accuracy", "speed", "efficiency")  
lm\_models <- c("age", "sex","exec\_accuracy")  
for (hemi in hemis) {  
 for (model in lm\_models) {  
 for (coeff in coeffs) {  
 for (corr in corrs) {  
 vector\_init\_cmd <- paste0(hemi, "\_lm\_", model, "\_", coeff, "\_", corrs, "<- vector(length= 10242)")  
 eval(parse(text=as.name(vector\_init\_cmd)))  
 }  
 }  
 }  
}  
  
  
#######################  
######## Left #########  
#######################  
  
#get # of items in df for calculation of column)  
numcolumns <- dim(lh\_imco)[2]  
#run gams models and store info in respective vectors  
for (i in 1:10242) {  
 curcol = (numcolumns - 10242 + i) # will start you counting at the right part of the df  
 age\_sex\_model <- gam(lh\_imco[,curcol] ~ pcaslRelMeanRMSMotion + restRelMeanRMSMotion +  
 osex + s(ageAtScan1, k = 4, fx = T), data=lh\_imco)  
   
 ## accuracy  
   
 exec\_accuracy\_model <- gam(lh\_imco[,curcol] ~ pcaslRelMeanRMSMotion + restRelMeanRMSMotion +  
 osex + s(ageAtScan1, k = 4, fx = T) + F1\_Exec\_Comp\_Res\_Accuracy, data=lh\_imco)  
   
 #lm  
 age\_lm\_model <- lm(lh\_imco[,curcol] ~ pcaslRelMeanRMSMotion + restRelMeanRMSMotion + ageAtScan1,   
 data=lh\_imco)  
 sex\_lm\_model <- lm(lh\_imco[,curcol] ~ pcaslRelMeanRMSMotion + restRelMeanRMSMotion + osex,   
 data=lh\_imco)  
   
 ## acc  
   
 exec\_accuracy\_lm\_model <- lm(lh\_imco[,curcol] ~ pcaslRelMeanRMSMotion + restRelMeanRMSMotion + ageAtScan1 + F1\_Exec\_Comp\_Res\_Accuracy,   
 data=lh\_imco)  
   
  
   
 #put pvalue in it's appropriate lm  
 lh\_gam\_age\_p\_uncor[i] <- summary(age\_sex\_model)$s.table[1,4] #smooth term for ageAtScan1  
 lh\_gam\_sex\_p\_uncor[i] <- summary(age\_sex\_model)$p.table[4,4] #linear term  
   
 lh\_gam\_exec\_accuracy\_p\_uncor[i] <- summary(exec\_accuracy\_model)$p.table[5,4] #accuracy term  
   
 #lm to assess directionality  
 lh\_lm\_age\_p\_uncor[i] <- summary(age\_lm\_model)$coeff[4,4]  
 lh\_lm\_sex\_p\_uncor[i] <- summary(sex\_lm\_model)$coeff[4,4]  
   
 lh\_lm\_exec\_accuracy\_p\_uncor[i] <- summary(exec\_accuracy\_lm\_model)$coeff[5,4]  
   
 #put tvalue in it's appropriate lm  
 lh\_gam\_age\_t\_uncor[i] <- summary(age\_sex\_model)$s.table[1,3] #smooth term for ageAtScan1  
 lh\_gam\_sex\_t\_uncor[i] <- summary(age\_sex\_model)$p.table[4,3] #linear term  
   
 lh\_gam\_exec\_accuracy\_t\_uncor[i] <- summary(exec\_accuracy\_model)$p.table[5,3] #accuracy term  
  
 #lm to assess directionality  
 lh\_lm\_age\_t\_uncor[i] <- summary(age\_lm\_model)$coeff[4,3]  
 lh\_lm\_sex\_t\_uncor[i] <- summary(sex\_lm\_model)$coeff[4,3] #linear term  
   
 lh\_lm\_exec\_accuracy\_t\_uncor[i] <- summary(exec\_accuracy\_lm\_model)$coeff[5,3]  
  
}  
  
#####################  
###### RIGHT ########  
#####################  
#get # of items in df for calculation of column)  
numcolumns <- dim(rh\_imco)[2]  
#run gams models and store info in respective vectors  
for (i in 1:10242) {  
 curcol = (numcolumns - 10242 + i) # will start you counting at the right part of the df  
 age\_sex\_model <- gam(rh\_imco[,curcol] ~ pcaslRelMeanRMSMotion + restRelMeanRMSMotion +  
 osex + s(ageAtScan1, k = 4, fx = T), data=rh\_imco)  
   
 ## accuracy  
 exec\_accuracy\_model <- gam(rh\_imco[,curcol] ~ pcaslRelMeanRMSMotion + restRelMeanRMSMotion +  
 osex + s(ageAtScan1, k = 4, fx = T) + F1\_Exec\_Comp\_Res\_Accuracy, data=rh\_imco)  
   
   
   
   
 #lm  
 age\_lm\_model <- lm(rh\_imco[,curcol] ~ pcaslRelMeanRMSMotion + restRelMeanRMSMotion + ageAtScan1,   
 data=rh\_imco)  
 sex\_lm\_model <- lm(rh\_imco[,curcol] ~ pcaslRelMeanRMSMotion + restRelMeanRMSMotion + osex,   
 data=rh\_imco)  
   
 ## acc  
  
 exec\_accuracy\_lm\_model <- lm(rh\_imco[,curcol] ~ pcaslRelMeanRMSMotion + restRelMeanRMSMotion + ageAtScan1 + F1\_Exec\_Comp\_Res\_Accuracy,   
 data=rh\_imco)  
   
   
   
 #put pvalue in it's appropriate lm  
 rh\_gam\_age\_p\_uncor[i] <- summary(age\_sex\_model)$s.table[1,4] #smooth term for ageAtScan1  
 rh\_gam\_sex\_p\_uncor[i] <- summary(age\_sex\_model)$p.table[4,4] #linear term  
   
 rh\_gam\_exec\_accuracy\_p\_uncor[i] <- summary(exec\_accuracy\_model)$p.table[5,4] #accuracy term  
   
 #lm to assess directionality  
 rh\_lm\_age\_p\_uncor[i] <- summary(age\_lm\_model)$coeff[4,4]  
 rh\_lm\_sex\_p\_uncor[i] <- summary(sex\_lm\_model)$coeff[4,4]  
   
 rh\_lm\_exec\_accuracy\_p\_uncor[i] <- summary(exec\_accuracy\_lm\_model)$coeff[5,4]  
   
 #put tvalue in it's appropriate lm  
 rh\_gam\_age\_t\_uncor[i] <- summary(age\_sex\_model)$s.table[1,3] #smooth term for ageAtScan1  
 rh\_gam\_sex\_t\_uncor[i] <- summary(age\_sex\_model)$p.table[4,3] #linear term  
   
 rh\_gam\_exec\_accuracy\_t\_uncor[i] <- summary(exec\_accuracy\_model)$p.table[5,3] #accuracy term  
  
 #lm to assess directionality  
 rh\_lm\_age\_t\_uncor[i] <- summary(age\_lm\_model)$coeff[4,3]  
 rh\_lm\_sex\_t\_uncor[i] <- summary(sex\_lm\_model)$coeff[4,3] #linear term  
   
 rh\_lm\_exec\_accuracy\_t\_uncor[i] <- summary(exec\_accuracy\_lm\_model)$coeff[5,3]  
   
}  
  
  
  
  
  
  
  
  
#################################################################################  
#################################################################################  
  
#####################################################  
# results #  
#####################################################  
  
#### FDR correction ####  
for (hemi in hemis) {  
 for (model in models) {  
 hemi\_model\_p\_unc <- paste0(hemi, "\_gam\_", model, "\_p\_uncor")   
 hemi\_model\_p\_fdr <- paste0(hemi, "\_gam\_", model, "\_p\_fdr")   
   
 print(hemi\_model\_p\_unc)  
   
 #correct p values  
 pfdr <- eval(substitute(p.adjust(i, method="fdr"), list(i = as.name(hemi\_model\_p\_unc))))  
   
 #figure out which values are < 0.05 and add to pfdr matrix  
 pfdr <- as.data.frame(pfdr)  
 pfdr$sig <- ifelse(pfdr<0.05, 1, 0)  
 pfdr$sig\_noNA <- ifelse(is.na(pfdr$sig), 0, pfdr$sig)  
 names(pfdr) <- c("pfdr", "sig05", "sig05\_noNA")  
 hemi\_model\_p\_fdr <- as.data.frame(pfdr[,1]) #sig05  
   
   
   
 #multiply T values by fdr vector to get the list of Ts that are fdr corrected  
 hemi\_model\_t\_unc <- paste0(hemi, "\_gam\_", model, "\_t\_uncor")  
 hemi\_model\_t\_fdr <- paste0(hemi, "\_gam\_", model, "\_t\_fdr")  
 t\_df <- eval(substitute(as.data.frame(i), list(i = as.name(hemi\_model\_t\_unc))))  
 names(t\_df) <- c("tval")  
 t\_df$tfdr <- pfdr[,3] \* t\_df$tval  
 hemi\_model\_t\_fdr <- as.data.frame(t\_df[,2])  
   
   
 #######################  
 #### write tables #####  
 #######################  
   
 ## uncorrected ##  
   
 ### p  
 filename <- paste0(homedir, "/baller/results/coupling\_accuracy/", hemi\_model\_p\_unc, ".csv")  
 write\_table\_command <- paste0("write.table(x = ", hemi\_model\_p\_unc, ", file = \"", filename,"\", row.names = FALSE, col.names = FALSE)")  
 eval(parse(text=write\_table\_command))  
   
 ### t  
   
 filename <- paste0(homedir, "/baller/results/coupling\_accuracy/", hemi\_model\_t\_unc, ".csv")  
 write\_table\_command <- paste0("write.table(x = ", hemi\_model\_t\_unc, ", file = \"", filename,"\", row.names = FALSE, col.names = FALSE)")  
 eval(parse(text=write\_table\_command))  
   
 ## corrected ##  
   
 ### p  
 filename <- paste0(homedir, "/baller/results/coupling\_accuracy/", hemi, "\_gam\_", model, "\_p\_fdr05.csv")  
 write\_table\_command <- paste0("write.table(x = ", hemi\_model\_p\_fdr, ", file = \"", filename,"\", row.names = FALSE, col.names = FALSE)")  
 eval(parse(text=write\_table\_command))  
   
 ### t  
 filename <- paste0(homedir, "/baller/results/coupling\_accuracy/", hemi, "\_gam\_", model, "\_t\_fdr05.csv")  
 write\_table\_command <- paste0("write.table(x = ", hemi\_model\_t\_fdr, ", file = \"", filename,"\", row.names = FALSE, col.names = FALSE)")  
 eval(parse(text=write\_table\_command))  
   
 }  
}  
  
###### linear model alone #####  
  
#### FDR correction ####  
for (hemi in hemis) {  
 for (model in lm\_models) {  
 hemi\_model\_p\_unc <- paste0(hemi, "\_lm\_", model, "\_p\_uncor")   
 hemi\_model\_p\_fdr <- paste0(hemi, "\_lm\_", model, "\_p\_fdr")   
   
 print(hemi\_model\_p\_unc)  
   
 #correct p values  
 pfdr <- eval(substitute(p.adjust(i, method="fdr"), list(i = as.name(hemi\_model\_p\_unc))))  
   
 #figure out which values are < 0.05 and add to pfdr matrix  
 pfdr <- as.data.frame(pfdr)  
 pfdr$sig <- ifelse(pfdr<0.05, 1, 0)  
 pfdr$sig\_noNA <- ifelse(is.na(pfdr$sig), 0, pfdr$sig)  
 names(pfdr) <- c("pfdr", "sig05", "sig05\_noNA")  
 hemi\_model\_p\_fdr <- as.data.frame(pfdr[,1]) #sig05  
   
   
   
 #multiply T values by fdr vector to get the list of Ts that are fdr corrected  
 hemi\_model\_t\_unc <- paste0(hemi, "\_lm\_", model, "\_t\_uncor")  
 hemi\_model\_t\_fdr <- paste0(hemi, "\_lm\_", model, "\_t\_fdr")  
 t\_df <- eval(substitute(as.data.frame(i), list(i = as.name(hemi\_model\_t\_unc))))  
 names(t\_df) <- c("tval")  
 t\_df$tfdr <- pfdr[,3] \* t\_df$tval  
 hemi\_model\_t\_fdr <- as.data.frame(t\_df[,2])  
   
   
 #######################  
 #### write tables #####  
 #######################  
   
 ## uncorrected ##  
   
 ### p  
 filename <- paste0(homedir, "/baller/results/coupling\_accuracy/", hemi\_model\_p\_unc, ".csv")  
 write\_table\_command <- paste0("write.table(x = ", hemi\_model\_p\_unc, ", file = \"", filename,"\", row.names = FALSE, col.names = FALSE)")  
 eval(parse(text=write\_table\_command))  
   
 ### t  
   
 filename <- paste0(homedir, "/baller/results/coupling\_accuracy/", hemi\_model\_t\_unc, ".csv")  
 write\_table\_command <- paste0("write.table(x = ", hemi\_model\_t\_unc, ", file = \"", filename,"\", row.names = FALSE, col.names = FALSE)")  
 eval(parse(text=write\_table\_command))  
   
 ## corrected ##  
   
 ### p  
 filename <- paste0(homedir, "/baller/results/coupling\_accuracy/", hemi, "\_lm\_", model, "\_p\_fdr05.csv")  
 write\_table\_command <- paste0("write.table(x = ", hemi\_model\_p\_fdr, ", file = \"", filename,"\", row.names = FALSE, col.names = FALSE)")  
 eval(parse(text=write\_table\_command))  
   
 ### t  
 filename <- paste0(homedir, "/baller/results/coupling\_accuracy/", hemi, "\_lm\_", model, "\_t\_fdr05.csv")  
 write\_table\_command <- paste0("write.table(x = ", hemi\_model\_t\_fdr, ", file = \"", filename,"\", row.names = FALSE, col.names = FALSE)")  
 eval(parse(text=write\_table\_command))  
 }  
}  
  
###########  
## Plots ##  
###########  
  
  
### Find Gam age FDR corrected  
### just for age  
lh\_gam\_age\_t\_fdr05 <- read.csv(paste0(homedir, '/baller/results/coupling\_accuracy/lh\_gam\_age\_t\_fdr05.csv'), header = F)  
rh\_gam\_age\_t\_fdr05 <- read.csv(paste0(homedir, '/baller/results/coupling\_accuracy/rh\_gam\_age\_t\_fdr05.csv'), header = F)  
  
#turn 0s into na  
lh\_gam\_to\_keep <- lh\_gam\_age\_t\_fdr05$V1  
lh\_gam\_to\_keep[lh\_gam\_to\_keep==0] <- NA  
  
rh\_gam\_to\_keep <- rh\_gam\_age\_t\_fdr05$V1  
rh\_gam\_to\_keep[rh\_gam\_to\_keep==0] <- NA  
  
#concatenate rows to keep  
lh\_and\_rh\_to\_keep <- c(lh\_gam\_to\_keep, rh\_gam\_to\_keep)  
columns\_to\_drop <- which(is.na(lh\_and\_rh\_to\_keep))  
  
# concatenate 831x10242 right and left matrices, and drop the columns with NA  
lh\_and\_rh\_matrix <- data.frame(cbind(lh\_matrix, rh\_matrix))  
lh\_and\_rh\_matrix\_fdr\_corrected = subset(lh\_and\_rh\_matrix, select = -(columns\_to\_drop))   
  
lh\_and\_rh\_rowmeans <- rowMeans(lh\_and\_rh\_matrix\_fdr\_corrected) #make vector of row means  
age = subjDemos$ageAtScan1  
#age\_x2\_for\_plotting <- c(subjDemos$ageAtScan1, subjDemos$ageAtScan1) #make vector of ages  
  
#plot and save  
jpeg(paste0(homedir,'baller/results/images/Mean\_coupling\_by\_age\_rplot\_fdr.jpg'))  
plot(age, lh\_and\_rh\_rowmeans, ylab = "Coupling (Z)", xlab = "Age (In Years)", main = "Mean Coupling by Age")  
abline(lm(lh\_and\_rh\_rowmeans~age), col = 'red')  
dev.off()  
  
### Find Gam age uncorrected  
### just for age  
lh\_gam\_age\_t\_uncor <- read.csv(paste0(homedir, '/baller/results/coupling\_accuracy/lh\_gam\_age\_t\_uncor.csv'), header = F)  
rh\_gam\_age\_t\_uncor <- read.csv(paste0(homedir, '/baller/results/coupling\_accuracy/rh\_gam\_age\_t\_uncor.csv'), header = F)  
  
#turn 0s into na  
lh\_gam\_to\_keep <- lh\_gam\_age\_t\_uncor$V1  
lh\_gam\_to\_keep[lh\_gam\_to\_keep==0] <- NA  
  
rh\_gam\_to\_keep <- rh\_gam\_age\_t\_uncor$V1  
rh\_gam\_to\_keep[rh\_gam\_to\_keep==0] <- NA  
  
#concatenate rows to keep  
lh\_and\_rh\_to\_keep <- c(lh\_gam\_to\_keep, rh\_gam\_to\_keep)  
columns\_to\_drop <- which(is.na(lh\_and\_rh\_to\_keep))  
  
# concatenate 831x10242 right and left matrices, and drop the columns with NA  
lh\_and\_rh\_matrix <- data.frame(cbind(lh\_matrix, rh\_matrix))  
lh\_and\_rh\_matrix\_uncorrected = subset(lh\_and\_rh\_matrix, select = -(columns\_to\_drop))   
  
lh\_and\_rh\_rowmeans <- rowMeans(lh\_and\_rh\_matrix\_uncorrected) #make vector of row means  
age = subjDemos$ageAtScan1  
#age\_x2\_for\_plotting <- c(subjDemos$ageAtScan1, subjDemos$ageAtScan1) #make vector of ages  
  
#plot and save  
jpeg(paste0(homedir,'baller/results/images/Mean\_coupling\_by\_age\_rplot\_uncor.jpg'))  
plot(age, lh\_and\_rh\_rowmeans, ylab = "Coupling (Z)", xlab = "Age (In Years)", main = "Mean Coupling by Age")  
abline(lm(lh\_and\_rh\_rowmeans~age), col = 'red')  
dev.off()  
  
#visreg  
jpeg(paste0(homedir,'baller/results/images/Mean\_coupling\_by\_age\_rplot\_uncor\_visreg.jpg'))  
df <- data.frame(cbind(lh\_and\_rh\_rowmeans, age))  
names(df) <- c("Means", "Age")  
fit <- lm(Means~Age, data=df)  
visreg(fit, "Age")  
dev.off()

Plots for Age x Coupling including GAM *script: /project/imco/baller/scripts/coupling/scatter\_plots.R*

\*make sure you install.packages(“devtools”)

###########  
## Plots ##  
###########  
  
  
###pre: gam age results, both fdr and uncorrected. rh and lh matrices  
###post: jpegs of linear and gam modesl  
###uses: makes the scatter plot of age by couping means for Figure 2  
### dependencies: R 3.6.3  
  
library(mgcv)  
library(dplyr)  
library(ggplot2)  
library(visreg)  
  
#set home directory, switch this depending on whether running from PMACS or from home directory  
#homedir <- "/Users/eballer/BBL/imco/pmacs/PMACS\_remote/"  
#homedir <- "/project/imco/"  
  
numrows = 831  
  
  
# read in lh and rh matrices, which have demos appended  
lh\_df <- read.table(paste0(homedir, "/baller/results/coupling\_accuracy/subjDemos\_with\_lh\_", numrows, "x10242.csv"), sep = ",")  
rh\_df <- read.table(paste0(homedir, "/baller/results/coupling\_accuracy/subjDemos\_with\_rh\_", numrows, "x10242.csv"), sep = ",")  
  
#just get the coupling values  
lh\_matrix <- lh\_df[,(dim(lh\_df)[2]-10241):dim(lh\_df)[2]]  
rh\_matrix <- rh\_df[,(dim(rh\_df)[2]-10241):dim(rh\_df)[2]]  
  
### Find Gam age FDR corrected  
### just for age  
lh\_gam\_age\_t\_fdr05 <- read.csv(paste0(homedir, '/baller/results/coupling\_accuracy/lh\_gam\_age\_t\_fdr05.csv'), header = F)  
rh\_gam\_age\_t\_fdr05 <- read.csv(paste0(homedir, '/baller/results/coupling\_accuracy/rh\_gam\_age\_t\_fdr05.csv'), header = F)  
  
#turn 0s into na  
lh\_gam\_to\_keep <- lh\_gam\_age\_t\_fdr05$V1  
lh\_gam\_to\_keep[lh\_gam\_to\_keep==0] <- NA  
  
rh\_gam\_to\_keep <- rh\_gam\_age\_t\_fdr05$V1  
rh\_gam\_to\_keep[rh\_gam\_to\_keep==0] <- NA  
  
#concatenate rows to keep  
lh\_and\_rh\_to\_keep <- c(lh\_gam\_to\_keep, rh\_gam\_to\_keep)  
columns\_to\_drop <- which(is.na(lh\_and\_rh\_to\_keep))  
  
# concatenate 831x10242 right and left matrices, and drop the columns with NA  
lh\_and\_rh\_matrix <- data.frame(cbind(lh\_matrix, rh\_matrix))  
lh\_and\_rh\_matrix\_fdr\_corrected = subset(lh\_and\_rh\_matrix, select = -(columns\_to\_drop))   
  
lh\_and\_rh\_rowmeans <- rowMeans(lh\_and\_rh\_matrix\_fdr\_corrected) #make vector of row means  
age = lh\_df$ageAtScan1  
  
#plot and save  
jpeg(paste0(homedir,'baller/results/images/Mean\_coupling\_by\_age\_rplot\_fdr05.jpg'))  
plot(age, lh\_and\_rh\_rowmeans, ylab = "Coupling (Z)", xlab = "Age (In Years)", main = "Mean Coupling by Age")  
abline(lm(lh\_and\_rh\_rowmeans~age), col = 'red')  
dev.off()  
  
#visreg  
  
jpeg(paste0(homedir,'baller/results/images/Mean\_coupling\_by\_age\_rplot\_fdr05\_visreg.jpg'))  
fit <- lm(Means~Age, data=df)  
visreg(fit, "Age", ylab = "Mean Coupling (Z)", xlab = "Age (in Years)",  
 line=list(col="red"),  
 points=list(col="black"))  
dev.off()  
  
jpeg(paste0(homedir, 'baller/results/images/Mean\_coupling\_by\_age\_rplot\_fdr05\_visreg\_gam.jpg'))  
fit\_gam <- gam(Means~s(Age, k = 4, fx=T), data=df)  
visreg(fit\_gam, "Age", ylab = "Mean Coupling (Z)", xlab = "Age (in Years)",  
 line=list(col="red"),  
 points=list(col="black"))  
dev.off()  
  
#################  
## Derivatives ##  
#################  
d<-derivatives(fit\_gam,n=1000)  
d\_plot <- draw(d)  
print(d\_plot)  
ggsave(plot = d\_plot,filename = paste0(homedir, "baller/results/images/derivative\_plot\_age\_gam\_fdr\_corrected.png"),device = "png",width = 180,height = 120,units = "mm")  
  
d<- d %>%  
 mutate(sig = !(0 >lower & 0 < upper)) #Ages where the CI does not include zero  
cat(sprintf("\nSignificant change: %1.2f - %1.2f\n",min(d$data[d$sig==T]),max(d$data[d$sig==T]))) #this only work  
  
######  
  
  
  
### Find Gam age uncorrected  
### just for age  
lh\_gam\_age\_t\_uncor <- read.csv(paste0(homedir, '/baller/results/coupling\_accuracy/lh\_gam\_age\_t\_uncor.csv'), header = F)  
rh\_gam\_age\_t\_uncor <- read.csv(paste0(homedir, '/baller/results/coupling\_accuracy/rh\_gam\_age\_t\_uncor.csv'), header = F)  
  
#turn 0s into na  
lh\_gam\_to\_keep <- lh\_gam\_age\_t\_uncor$V1  
lh\_gam\_to\_keep[lh\_gam\_to\_keep==0] <- NA  
  
rh\_gam\_to\_keep <- rh\_gam\_age\_t\_uncor$V1  
rh\_gam\_to\_keep[rh\_gam\_to\_keep==0] <- NA  
  
#concatenate rows to keep  
lh\_and\_rh\_to\_keep <- c(lh\_gam\_to\_keep, rh\_gam\_to\_keep)  
columns\_to\_drop <- which(is.na(lh\_and\_rh\_to\_keep))  
  
# concatenate 831x10242 right and left matrices, and drop the columns with NA  
lh\_and\_rh\_matrix <- data.frame(cbind(lh\_matrix, rh\_matrix))  
lh\_and\_rh\_matrix\_uncorrected = subset(lh\_and\_rh\_matrix, select = -(columns\_to\_drop))   
  
lh\_and\_rh\_rowmeans <- rowMeans(lh\_and\_rh\_matrix\_uncorrected) #make vector of row means  
age = lh\_df$ageAtScan1  
#age\_x2\_for\_plotting <- c(subjDemos$ageAtScan1, subjDemos$ageAtScan1) #make vector of ages  
  
#plot and save  
jpeg(paste0(homedir,'baller/results/images/Mean\_coupling\_by\_age\_rplot\_uncor.jpg'))  
plot(age, lh\_and\_rh\_rowmeans, ylab = "Coupling (Z)", xlab = "Age (In Years)", main = "Mean Coupling by Age")  
abline(lm(lh\_and\_rh\_rowmeans~age), col = 'red')  
dev.off()  
  
#visreg  
df <- data.frame(cbind(lh\_and\_rh\_rowmeans, age))  
names(df) <- c("Means", "Age")  
  
jpeg(paste0(homedir,'baller/results/images/Mean\_coupling\_by\_age\_rplot\_uncor\_visreg.jpg'))  
fit <- lm(Means~Age, data=df)  
visreg(fit, "Age", ylab = "Mean Coupling (Z)", xlab = "Age (in Years)",  
 line=list(col="red"),  
 points=list(col="black"))  
dev.off()  
  
jpeg(paste0(homedir, 'baller/results/images/Mean\_coupling\_by\_age\_rplot\_uncor\_visreg\_gam.jpg'))  
fit\_gam <- gam(Means~s(Age, k = 4, fx=T), data=df)  
visreg(fit\_gam, "Age", ylab = "Mean Coupling (Z)", xlab = "Age (in Years)",  
 line=list(col="red"),  
 points=list(col="black"))  
dev.off()  
  
######## Deriatives ########  
d<-derivatives(fit\_gam,n=1000)  
d\_plot <- draw(d)  
print(d\_plot)  
ggsave(plot = d\_plot,filename = paste0(homedir, "baller/results/images/derivative\_plot\_age\_gam\_uncorrected.png"),device = "png",width = 180,height = 120,units = "mm")  
  
d<- d %>%  
 mutate(sig = !(0 >lower & 0 < upper)) #Ages where the CI does not include zero  
cat(sprintf("\nSignificant change: %1.2f - %1.2f\n",min(d$data[d$sig==T]),max(d$data[d$sig==T])))

**Making actual visualization**

The visualizations are done in matlab, unfortunately, but I have made a bash script that can call matlab, so you no longer have to go into matlab and do it manually. Log in to PMACS, navigate to PBP\_graphics directory, which contains all the scripts I have made to make pretty visualizations. This will get the right version of matlab for you as well.

cd /project/imco/baller/scripts/PBP\_graphics  
source ../load\_matlab #contains the following line of code: bsub -Is -XF -q matlab\_interactive 'bash'  
sh run\_visualization\_scripts\_from\_command\_line.sh

The actual code in run\_visualization\_scripts\_from\_command\_line.sh is below:

#!/bin/sh  
  
##################################  
### PBP visualizations Wrapper ###  
##################################  
  
###### Author: Erica Baller ######  
###### Date: 2/26/2021 ######  
  
## pre: commands\_for\_matlab (in same directory as PBP scripts  
 #- contains all the commands we want to pass to matlab to run visualizations automaticall  
## post: images in /project/imco/baller/results/images/pbp  
## uses: I was getting tired of having to open matlab and run each of these commands manually. This script takes a bunch of commands and just feeds them to matlab, no muss, no fuss  
## dependencies: Matlab 2020b, please run "source /project/imco/baller/scripts/load\_matlab before starting this  
  
  
#set directories  
#homedir = '/project/imco'  
#set homedir = '/Users/eballer/BBL/imco/pmacs/PMACS\_remote/'  
wking\_dir='/baller/scripts/PBP\_graphics'  
command\_file="$homedir/$wking\_dir/commands\_for\_matlab"  
echo $command\_file  
  
#initialize matlab  
  
  
#######  
### loop ###  
while IFS= read -r line; do  
 echo "$line"  
 matlab -nosplash -nodesktop -nodisplay -r "$line; exit"  
  
done < $command\_file

The command files contain paths to the PBP\* visualization scripts. At any time, you can run one or all of them by changing the inputs in /project/imco/baller/scripts/PBP\_graphics/commands\_for\_matlab. Currently, these are the I recommend putting in that file. Of note, if you ever want to run a command independently, just go into matlab, copy the command, and run it within matlab. This just takes forever. Here are the final commands for visualization that I ran (I made individual scripts that included different color options).

PBP\_vertWiseEffect\_Erica\_mean\_coupling\_plasma\_results\_outpath('/project/imco/baller/results/mean\_maps/n831\_lh.coupling\_coef\_alff\_cbf.fwhm15.fsaverage5\_THRESH\_3.09.csv','/project/imco/baller/results/mean\_maps/n831\_rh.coupling\_coef\_alff\_cbf.fwhm15.fsaverage5\_THRESH\_3.09.csv','coupling\_mean\_thresh\_3\_09\_plasma')  
PBP\_vertWiseEffect\_Erica\_gam\_age\_parula\_results\_outpath('/project/imco/baller/results/coupling\_accuracy/lh\_gam\_with\_lm\_sign\_age\_t\_fdr05.csv','/project/imco/baller/results/coupling\_accuracy/rh\_gam\_with\_lm\_sign\_age\_t\_fdr05.csv','coupling\_gam\_w\_lm\_sign\_age\_parula\_fdr05\_grey\_mid')  
PBP\_vertWiseEffect\_Erica\_Ts\_pos\_and\_neg\_results\_outpath('/project/imco/baller/results/coupling\_accuracy/lh\_gam\_sex\_t\_fdr05.csv','/project/imco/baller/results/coupling\_accuracy/rh\_gam\_sex\_t\_fdr05.csv','coupling\_gam\_sex\_red\_and\_blue\_fdr05')  
PBP\_vertWiseEffect\_Erica\_Ts\_pos\_and\_neg\_results\_outpath('/project/imco/baller/results/coupling\_accuracy/lh\_gam\_exec\_accuracy\_t\_fdr05.csv','/project/imco/baller/results/coupling\_accuracy/rh\_gam\_exec\_accuracy\_t\_fdr05.csv','coupling\_gam\_exec\_accuracy\_red\_and\_blue\_fdr05')

Final images:

Coupling mean:

A picture containing chart

Description automatically generated

Coupling gam age

A picture containing chart

Description automatically generated

Coupling gam sex

A picture containing several

Description automatically generated

Coupling gam exec accuracy

A picture containing icon

Description automatically generated

Sample code (from PBP\_vertWiseEffect\_Erica\_Ts\_results\_outpath.m)

function PBP\_vertWiseEffect\_Erica(LH,RH,name)% pretty picture code, AAB 4/2018 - AP 5/1/20 - Updated to threshold according to inp  
ut - 1/12/21  
% data should be vectors, 10242 in length if fsaverage5 is used  
% if using higher resolution, then change accordingly  
% depencies include: matlab freesrufer functions, subaxis.m (matlab central), inferno color scale (matlab central - for Sam ;)  
  
%%% SET THRESHOLDS AS DESIRED HERE: only fill in each threshold as needed (no need to set both if you only want to threshold one e  
nd)  
% Values at or above this set to gray  
 %Uthresh=-2;  
% Values at or below this set to gray  
 LThresh=2;  
%%%  
  
  
addpath(genpath('/appl/freesurfer-6.0.0/matlab/'));  
addpath(genpath('/project/imco/baller/scripts/subaxis/'));  
addpath(genpath('/project/imco/baller/scripts/Colormaps/Colormaps (5)/Colormaps/'));  
%{  
addpath(genpath('/cbica/projects/alpraz\_EI/scripts/tools/'));  
ProjectFolder = '/cbica/projects/pinesParcels/data/SingleParcellation';  
SubjectsFolder = '/cbica/software/external/freesurfer/centos7/5.3.0/subjects/fsaverage5';  
%}  
  
plot\_text='';  
[vertices, faces] = freesurfer\_read\_surf('/project/imco/surfaces/fsaverage5/surf/lh.inflated');  
%using lh.gray will make more anatomical looking plot but harder to see into sulci  
right = readtable(RH,'TreatAsEmpty','NA','ReadVariableNames',false);  
datar = table2array(right);  
left = readtable(LH,'TreatAsEmpty','NA','ReadVariableNames',false);  
datal = table2array(left);  
  
%left=load(LHvec);  
%right=load(RHvec);  
%datal=left;  
%datar=right;  
minval = min(min(datal),min(datar)) %useful for colorbar later   
  
%set NaN to 0  
%I generally have the midcut region set to NaN  
%in the csv files that I read in  
indexNaNrh = find(isnan(datar));  
indexNaNlh = find(isnan(datal));  
datar(indexNaNrh)=0;  
datal(indexNaNlh)=0;  
datalr=[datal; datar];  
%invoke thresholding 1/12/21  
if exist('Uthresh','Var') == 1;  
 AboveThresh= datalr > Uthresh;  
 datalr(AboveThresh)=0;  
end  
if exist('LThresh','Var') ==1;  
 BelowThresh= datalr < LThresh;  
 datalr(BelowThresh)=0;  
end  
%%% set color scale  
% 1/12/21 - for p values, visualizing 1/p might be more effective. comment out line below and uncomment subsequent line to nix th  
is approach.  
%datalr=1./datalr;  
  
  
% 12/1/21 tiny bit of code to deal with 1/0 in matlab  
InfIndex=find(datalr==Inf);  
% Infinity values to 0  
datalr(InfIndex)=0;  
  
%AP% set to make white zero on all maps  
maxabs=prctile(abs(datalr),88);  
%mincol= minval-.00001   
%maxcol=maxabs  
%mincol=-maxabs  
maxcol=max(datalr)  
mincol=min(datalr)  
%change above to set max/min manually or by other means  
%custommap=colormap('plasma'); %or whatever  
% for white at 0  
%custommap=colormap(b2r(-1,1));  
%custommap=colormap('jet');  
custommap=colormap('plasma')  
custommap(1,:)=[0.75 0.75 0.75];  
  
  
data=datalr(1:10242);  
asub = subaxis(4,2,1, 'sh', 0, 'sv', 0, 'padding', 0, 'margin', 0);  
%asub = subplot(4,2,1)  
% note use of subaxis is to ged rid of white space around brains   
% if you don't care about that, it's faster and less likely to cause  
% issues if you use subplot instead  
% if so, bet rid of all of the posnew stuff below  
  
aplot = trisurf(faces, vertices(:,1), vertices(:,2), vertices(:,3),data)  
view([90 0]);  
colormap(custommap)  
caxis([mincol; maxcol]);  
daspect([1 1 1]);  
axis tight;  
axis vis3d off;  
lighting gouraud; %phong;   
material metal %shiny %metal;   
shading flat;  
camlight;  
alpha(1)  
  
asub = subaxis(4,2,3, 'sh', 0.00, 'sv', 0.00, 'padding', 0, 'margin', 0);  
aplot = trisurf(faces, vertices(:,1), vertices(:,2), vertices(:,3),data)  
view([90 0]);  
rotate(aplot, [0 0 1], 180)  
colormap(custommap)  
caxis([mincol; maxcol]);  
daspect([1 1 1]);  
axis tight;  
axis vis3d off;  
lighting gouraud; %phong;   
material metal %shiny %metal;   
shading flat;  
camlight;  
alpha(1)  
set(gcf,'Color','w')  
  
asub = subaxis(4,2,5, 'sh', 0.0, 'sv', 0.0, 'padding', 0, 'margin', 0);  
aplot = trisurf(faces, vertices(:,1), vertices(:,2), vertices(:,3),data)  
view([90 0]);  
rotate(aplot, [0 0 1], 225)  
colormap(custommap)  
caxis([mincol; maxcol]);  
daspect([1 1 1]);  
axis tight;  
axis vis3d off;  
lighting gouraud; %phong;   
material metal %shiny %metal;   
shading flat;  
camlight;  
alpha(1)  
set(gcf,'Color','w')  
  
asub = subaxis(4,2,7, 'sh', 0.0, 'sv', 0.0, 'padding', 0, 'margin', 0, 'MT', 0.0);  
aplot = trisurf(faces, vertices(:,1), vertices(:,2), vertices(:,3),data)  
view([90 0]);  
axis vis3d off;  
rotate(aplot, [0 1 0], 270)  
colormap(custommap)  
caxis([mincol; maxcol]);  
daspect([1 1 1]);  
axis tight;  
lighting gouraud; %phong;   
material metal %shiny %metal;   
shading flat;  
camlight;  
alpha(1)  
set(gcf,'Color','w')  
  
 pos = get(asub, 'Position');  
 posnew = pos; posnew(2) = posnew(2) + 0.04; set(asub, 'Position', posnew);  
 %white space again  
  
%plot title   
title(plot\_text)  
set(get(gca,'title'),'Position',[332 119 3])  
  
%%% right hemisphere  
data=datalr(10243:20484);  
  
[vertices, faces] = freesurfer\_read\_surf('/project/imco/surfaces/fsaverage5/surf/rh.inflated');  
  
asub = subaxis(4,2,2, 'sh', 0.0, 'sv', 0.0, 'padding', 0, 'margin', 0);  
aplot = trisurf(faces, vertices(:,1), vertices(:,2), vertices(:,3),data)  
view([90 0]);  
rotate(aplot, [0 0 1], 180)  
colormap(custommap)  
caxis([mincol; maxcol]);  
%caxis([NAval; max\_data])  
daspect([1 1 1]);  
axis tight;  
axis vis3d off;  
lighting phong; %gouraud  
material metal %shiny %metal;   
shading flat;  
camlight;  
alpha(1)  
%colormap(mycol)  
  
  
 pos = get(asub, 'Position');  
 posnew = pos; posnew(1) = posnew(1) - 0.22; set(asub, 'Position', posnew);  
  
asub = subaxis(4,2,4, 'sh', 0.0, 'sv', 0.0, 'padding', 0, 'margin', 0);  
aplot = trisurf(faces, vertices(:,1), vertices(:,2), vertices(:,3),data)  
view([90 0]);  
colormap(custommap)  
caxis([mincol; maxcol]);  
daspect([1 1 1]);  
axis tight;  
axis vis3d off;  
lighting gouraud; %phong;   
material metal %shiny %metal;   
shading flat;  
camlight;  
alpha(1)  
set(gcf,'Color','w')  
 pos = get(asub, 'Position');  
 posnew = pos; posnew(1) = posnew(1) - 0.22; set(asub, 'Position', posnew);  
  
asub = subaxis(4,2,6, 'sh', 0.0, 'sv', 0.0, 'padding', 0, 'margin', 0);  
  
aplot = trisurf(faces, vertices(:,1), vertices(:,2), vertices(:,3),data)  
view([90 0]);  
rotate(aplot, [0 0 1], -45)  
colormap(custommap)  
caxis([mincol; maxcol]);  
daspect([1 1 1]);  
axis tight;  
axis vis3d off;  
lighting gouraud; %phong;   
material metal %shiny %metal;   
shading flat;  
camlight;  
alpha(1)  
set(gcf,'Color','w')  
 pos = get(asub, 'Position');  
 posnew = pos; posnew(1) = posnew(1) - 0.22; set(asub, 'Position', posnew);  
  
  
%%%  
asub = subaxis(4,2,8, 'sh', 0.0, 'sv', 0.0, 'padding', 0, 'margin', 0);  
  
aplot = trisurf(faces, vertices(:,1), vertices(:,2), vertices(:,3),data)  
view([90 0]);  
axis vis3d off;  
rotate(aplot, [0 1 0], 270)  
rotate(aplot, [1 0 0], 180)  
colormap(custommap)  
caxis([mincol; maxcol]);  
daspect([1 1 1]);  
axis tight;  
lighting gouraud; %phong;   
material metal %shiny %metal;   
shading flat;  
camlight;  
alpha(1)  
%set(gcf,'Color',[.2 .2 .2])  
set(gcf,'Color',[1,1,1])  
 pos = get(asub, 'Position');  
 posnew = pos; posnew(2) = posnew(2) + 0.04; set(asub, 'Position', posnew);  
 pos = get(asub, 'Position');  
 posnew = pos; posnew(1) = posnew(1) - 0.22; set(asub, 'Position', posnew);  
%%%  
  
  
acbar = colorbar('EastOutside')  
set(acbar, 'position', [0.40 0.270 0.02 0.20])  
  
  
% going lower rez for now, but giant vector rendering was beaut  
print('-dpng','-r600',['/project/imco/baller/results/images/pbp/' char(name)])

#### Spin Test

In reviewing our results, it looked very clear that the sex and executive accuracy maps followed very closely onto Yeo 7 networks. For interpretability and for graphs, we decided to do a variation on the spin test. Overall, the goal was to take the vertices (10242 for each hemisphere), and spin. We would next ask how many vertices would randomly and by chance fall within certain yeo networks as compared to what we actually saw. A challenge of this spin is that when we spin the vertices, we include medial wall which is guaranteed to be 0. In order to account for this, we calculated the proportion of vertices within each network minus the ones in the medial wall. We then used ggplot2 to make violin plots for display.

Unfortunatley, this was another step where we required both matlab and R. Lucky for you, I have a bash wrapper that calls matlab!

First, I made trinarized yeo masks, 1, 0, -1. In order to run permutation analyses on the Yeo networks, I need to trinarize my fdr corrected maps. If a vertex is corrected, it will get a 1. If not, 0. If it is within the medial wall, it will get a -1

script: /project/imco/baller/scripts/spin\_test/make\_trinarized\_maps\_for\_spin\_test.R

###########################################  
### Make Trinarzied Maps for Spin Test ###  
###########################################  
  
### Author: Erica Baller  
### Date 3/10/21  
  
### pre: statistical maps (l and r), each 10242 vertices  
### post: 6 matrices with 1s if value fdr corrected, 0 if not, and -1 if vertex in medial wall of yeo map  
 # lh and rh pos and neg together, positive alone, and negative alone  
### uses: In order to spin my vertices in a way that allows me to later assess their yeo membership, I need all of my fdr values indicated, as well as which values are in the medial wall. This script takes lh and rh fdr corrected vectors generated in previous scripts, booleanizes them into in/out of fdr map. Then, -1s are added for vertices in the medial wall.   
### dependencis: and R will do. I used 3.2.5  
  
## set abs and relative paths. Must toggle before running locally/on cluster  
#homedir = "/Users/eballer/BBL/imco/pmacs/PMACS\_remote"  
#homedir = "/project/imco"  
  
#### set # parcels in case I want to do 7 or 17 or something else in the future  
parcel\_type = "Yeo"   
parcel\_num = 7   
input\_parcel\_array\_length = 10242  
  
#will loop through each of these  
analyses <- c("coupling\_accuracy") #, "cbf\_accuracy", "alff\_accuracy")   
models <- c("gam\_age", "gam\_sex", "lm\_sex", "lm\_exec\_accuracy", "gam\_exec\_accuracy") #gam\_age", "lm\_age", "gam\_sex", "lm\_sex", "lm\_accuracy", "lm\_exec\_accuracy")  
corrs <- c("fdr05") #correction  
  
parcel\_mapping <- get\_parcel\_mapping\_yeo(7)  
lh\_numerical\_map <- parcel\_mapping[[1]]  
rh\_numerical\_map <- parcel\_mapping[[2]]  
  
for (analysis in analyses) {  
 for (model in models) {  
 for (corr in corrs) {   
   
 ### set results path  
 stat\_path <- paste0("/", analysis, "/")  
 print(stat\_path)  
 result\_path <- paste0(model, "\_t\_", corr)  
 print(result\_path)  
   
 ### set paths  
 ## input  
 lh\_stat\_map <- read.csv(paste0(homedir, "/baller/results/", stat\_path, "lh\_", result\_path, ".csv"), header = F)  
 rh\_stat\_map <- read.csv(paste0(homedir, "/baller/results/", stat\_path, "rh\_", result\_path, ".csv"), header = F)  
   
 ## output  
 lh\_outdir <- paste0(homedir, "/baller/results/", stat\_path, "lh\_", result\_path, "\_", parcel\_type, parcel\_num, "\_1\_0\_-1.csv")  
 rh\_outdir <- paste0(homedir, "/baller/results/", stat\_path, "rh\_", result\_path, "\_", parcel\_type, parcel\_num, "\_1\_0\_-1.csv")  
   
 lh\_outdir\_pos <- paste0(homedir, "/baller/results/", stat\_path, "lh\_pos\_", result\_path, "\_", parcel\_type, parcel\_num, "\_1\_0\_-1.csv")  
 rh\_outdir\_pos <- paste0(homedir, "/baller/results/", stat\_path, "rh\_pos\_", result\_path, "\_", parcel\_type, parcel\_num, "\_1\_0\_-1.csv")  
   
 lh\_outdir\_neg <- paste0(homedir, "/baller/results/", stat\_path, "lh\_neg\_", result\_path, "\_", parcel\_type, parcel\_num, "\_1\_0\_-1.csv")  
 rh\_outdir\_neg <- paste0(homedir, "/baller/results/", stat\_path, "rh\_neg\_", result\_path, "\_", parcel\_type, parcel\_num, "\_1\_0\_-1.csv")  
   
 #convert stat map to boolean  
 lh\_stat\_boolean <- ifelse(lh\_stat\_map == 0, 0, 1)  
 rh\_stat\_boolean <- ifelse(rh\_stat\_map == 0, 0, 1)  
   
 lh\_stat\_boolean\_pos <- ifelse(lh\_stat\_map > 0, 1, 0)  
 rh\_stat\_boolean\_pos <- ifelse(rh\_stat\_map > 0, 1, 0)  
   
 lh\_stat\_boolean\_neg <- ifelse(lh\_stat\_map < 0, 1, 0)  
 rh\_stat\_boolean\_neg <- ifelse(rh\_stat\_map < 0, 1, 0)  
   
 #convert NAs to 0  
 lh\_stat\_boolean[is.na(lh\_stat\_boolean)] <- 0  
 rh\_stat\_boolean[is.na(rh\_stat\_boolean)] <- 0  
   
 lh\_stat\_boolean\_pos[is.na(lh\_stat\_boolean\_pos)] <- 0  
 rh\_stat\_boolean\_pos[is.na(rh\_stat\_boolean\_pos)] <- 0  
   
 lh\_stat\_boolean\_neg[is.na(lh\_stat\_boolean\_neg)] <- 0  
 rh\_stat\_boolean\_neg[is.na(rh\_stat\_boolean\_neg)] <- 0  
   
   
 #substitute -1 for those locations with medial wall stuff  
 lh\_medial\_wall\_nums <- which(lh\_numerical\_map == 8)  
 rh\_medial\_wall\_nums <- which(rh\_numerical\_map == 8)  
   
 lh\_stat\_boolean[lh\_medial\_wall\_nums] <- -1  
 rh\_stat\_boolean[rh\_medial\_wall\_nums] <- -1  
   
 lh\_stat\_boolean\_pos[lh\_medial\_wall\_nums] <- -1  
 rh\_stat\_boolean\_pos[rh\_medial\_wall\_nums] <- -1  
   
 lh\_stat\_boolean\_neg[lh\_medial\_wall\_nums] <- -1  
 rh\_stat\_boolean\_neg[rh\_medial\_wall\_nums] <- -1  
   
   
 #write output  
 write.table(x = lh\_stat\_boolean, file = lh\_outdir, quote = F, row.names = F, col.names = F)  
 write.table(x = rh\_stat\_boolean, file = rh\_outdir, quote = F, row.names = F, col.names = F)  
   
 write.table(x = lh\_stat\_boolean\_pos, file = lh\_outdir\_pos, quote = F, row.names = F, col.names = F)  
 write.table(x = rh\_stat\_boolean\_pos, file = rh\_outdir\_pos, quote = F, row.names = F, col.names = F)  
   
 write.table(x = lh\_stat\_boolean\_neg, file = lh\_outdir\_neg, quote = F, row.names = F, col.names = F)  
 write.table(x = rh\_stat\_boolean\_neg, file = rh\_outdir\_neg, quote = F, row.names = F, col.names = F)  
   
 }  
 }  
}

\*\* Call matlab wrapper\*\*

#runs whatever command is in the commands\_for\_matlab file. Currently, it is set to: calc\_spinTest\_EB  
sh run\_spin\_test\_from\_command\_line.sh

\*\* Code in matlab wrapper \*\*

/project/imco/baller/scripts/spin\_test/run\_spin\_test\_from\_command\_line.sh

#!/bin/sh  
  
##################################  
### Spin Test Wrapper ###  
##################################  
  
###### Author: Erica Baller ######  
###### Date: 2/26/2021 ######  
  
## pre: commands\_for\_matlab   
 #- trinarized maps   
## post: spin results in /project/imco/baller/results/coupling\_accuracy/spin\_test\_results  
## uses: I was getting tired of having to open matlab and run each of these commands manually. This script takes a bunch of commands and just feeds them to matlab, no muss, no fuss  
## dependencies: Matlab 2020b, please run "source /project/imco/baller/scripts/load\_matlab before starting this  
  
  
#set directories  
#homedir = '/project/imco'  
#set homedir = '/Users/eballer/BBL/imco/pmacs/PMACS\_remote/'  
wking\_dir='/baller/scripts/spin\_test'  
command\_file="$homedir/$wking\_dir/commands\_for\_matlab"  
echo $command\_file  
  
#initialize matlab  
  
  
#######  
### loop ###  
while IFS= read -r line; do  
 echo "$line"  
 matlab -nosplash -nodesktop -nodisplay -r "$line; exit"  
  
done < $command\_file

\*\* commands for matlab example \*\* file: /project/imco/baller/scripts/spin\_test/commands\_for\_matlab

SpinPermuFS('/project/imco/baller/results/coupling\_accuracy/lh\_gam\_age\_t\_fdr05\_Yeo7\_1\_0\_-1.csv','/project/imco/baller/results/coupling\_accuracy/rh\_gam\_age\_t\_fdr05\_Yeo7\_1\_0\_-1.csv',1000,'/project/imco/baller/results/coupling\_accuracy/spin\_test\_results/gam\_age\_spin\_results\_yeo\_1\_0\_-1\_output','gam\_sex'). SpinPermuFS('/project/imco/baller/results/coupling\_accuracy/lh\_gam\_sex\_t\_fdr05\_Yeo7\_1\_0\_-1.csv','/project/imco/baller/results/coupling\_accuracy/rh\_gam\_sex\_t\_fdr05\_Yeo7\_1\_0\_-1.csv',1000,'/project/imco/baller/results/coupling\_accuracy/spin\_test\_results/gam\_sex\_spin\_results\_yeo\_1\_0\_-1\_output','gam\_sex')  
SpinPermuFS('/project/imco/baller/results/coupling\_accuracy/lh\_gam\_exec\_accuracy\_t\_fdr05\_Yeo7\_1\_0\_-1.csv','/project/imco/baller/results/coupling\_accuracy/rh\_gam\_exec\_accuracy\_t\_fdr05\_Yeo7\_1\_0\_-1.csv',1000,'/project/imco/baller/results/coupling\_accuracy/spin\_test\_results/gam\_exec\_accuracy\_spin\_results\_yeo\_1\_0\_-1\_output','gam\_exec\_accuracy')

The contents of SpinPermuFS (of note, I added an extra variable, model, so I could change what is actually called)

function SpinPermuFS(readleft,readright,permno,wsname,model)  
% Compute designated # of permutations/spins of the input surface data  
% in FreeSurfer fsaverage5.  
% FORMAT SpinPermuFS(readleft,readright,permno)  
% readleft - the filename of left surface data to spin   
% readright - the filename of right surface data to spin   
% permno - the number of permutations  
% wsname - the name of a workspace file including all spun data to be saved  
% model - gam\_accuracy, lm\_accuracy, gam\_sex, etc  
% Example SpinPermuFS('../data/depressionFSdataL.csv','../data/depressionFSdataR.csv',100,'../data/rotationFS.mat')  
% will spin prebuilt data, neurosynth map associated with 'depression', 100  
% times, and save the workspace file of all spun data in ../data/rotationFS.mat  
% Aaron Alexander-Bloch & Siyuan Liu   
% SpinPermuFS.m, 2018-04-22  
% The implementation of generating random rotations originally described in our paper —   
% rotating the coordinates of vertices at angles uniformly chosen between zero and 360 degrees  
% about each of the x (left-right), y (anterior-posterior) and z (superior-inferior) axes —  
% introduces a preference towards oversampling certain rotations.   
% Thus, we modified the code to incorporate an approach, Lefèvre et al. (2018),   
% that samples uniformly from the space of possible rotations. The updated  
% uniform sampling prodcedure does not require AxelRot.m anymore.  
% Updated on 2018-07-18  
  
  
%outdir= '/project/imco/baller/results/coupling\_accuracy/spin\_test\_results/';  
  
%Set up paths  
fshome = getenv('FREESURFER\_HOME');  
fsmatlab = sprintf('%s/matlab',fshome);  
path(path,fsmatlab);  
%read the data saved in csv  
datal=importdata(readleft);  
datar=importdata(readright);  
  
  
%%extract the correspoding sphere surface coordinates for rotation  
% AP changed the filepath 10/26/20, i don't know how cubic managed to set up a fshome path not amenable to this  
[verticesl, ~] = freesurfer\_read\_surf(fullfile('/project/imco/surfaces/fsaverage5/surf/lh.sphere'));  
[verticesr, ~] = freesurfer\_read\_surf(fullfile('/project/imco/surfaces/fsaverage5/surf/rh.sphere'));  
  
  
rng(0);  
%Use rng to initialize the random generator for reproducible results.  
%initialize variables to save rotation  
bigrotl=[];  
bigrotr=[];  
distfun = @(a,b) sqrt(bsxfun(@minus,bsxfun(@plus,sum(a.^2,2),sum(b.^2,1)),2\*(a\*b)));  
%function to calculate Euclidian distance  
I1 = eye(3,3);  
I1(1,1)=-1;  
bl=verticesl;  
br=verticesr;  
%permutation starts  
for j=1:permno  
 j  
 %the updated uniform sampling procedure  
 A = normrnd(0,1,3,3);  
 [TL, temp] = qr(A);  
 TL = TL \* diag(sign(diag(temp)));  
 if(det(TL)<0)  
 TL(:,1) = -TL(:,1);  
 end  
 %reflect across the Y-Z plane for right hemisphere  
 TR = I1 \* TL \* I1;  
 bl =bl\*TL;  
 br = br\*TR;   
   
 %Find the pair of matched vertices with the min distance and reassign  
 %values to the rotated surface.  
 distl=distfun(verticesl,bl');  
 distr=distfun(verticesr,br');  
 [~, Il]=min(distl,[],2);  
 [~, Ir]=min(distr,[],2);  
 %save rotated data  
 bigrotl=[bigrotl; datal(Il)'];  
 bigrotr=[bigrotr; datar(Ir)'];  
 % it is also feasible to save Il Ir and apply them to different datasets  
 % for repeated use  
 %If annotation file is used, annotation file for each rotation could be  
 %saved by write\_annotation.m of FreeSurfer  
end  
save(wsname,'bigrotl','bigrotr')  
writematrix(bigrotl, [outdir, '/lh\_spintest\_', model, '\_output.csv']);  
writematrix(bigrotr, [outdir, '/rh\_spintest\_', model, '\_output.csv']);  
%save bigrotl and bigrotr in a workspace file for the null distribution  
%use it in pvalvsNull.m to caclulate pvalue

\*\* Here we actually generate the proportion matrices and plot \*\*

Script: /project/imco/baller/scripts/spin\_test/spin\_proportion\_calculations\_and\_plots.R

##############################################  
########### Spin Test Distribution for #######  
##########Intermodal Coupling Paper ##########  
  
####### Author: Erica Baller  
#### Date: 3/9/2021  
  
#######  
##pre: right and left 10242 x 1000 matrices from matlab SpinPermuFS, yeo R & L assignments  
##post: 2 7 x 1000 matrices (r & l) that contain the proportion of vertices within a network divided by the total number of vertices, and plots  
## uses: Takes output of spin test, and calcualted the number of vertices within each of yeo's 7 networks out of the number of total possible vertices within the network  
 #### 1) Read in the yeo network assignments and calculate total number of vertices per network  
 #### 2) Multiply the yeo networks x the matrices (so every value is 1 -7 if they were within the mask, -1--7 if they were medial wall, and 0 otherwise)  
 #### 3) Foreach permutation (r and l separately), and for each network, calculate the (# of vertices with a 1) divided(/) by the (number of total vertices within network minus number of negative vertices  
 #### 4) Store  
 #### 5) Plot  
  
### dependencies: ggplot2, bigmemory, vroom  
  
  
#library(bigmemory.sri)  
library(ggplot2)  
library(tidyr)  
  
  
  
#################  
### set home directory  
#homedir <- "/Users/eballer/BBL/imco/pmacs/PMACS\_remote"  
homedir <- "/project/imco"  
  
source(paste0(homedir, "/baller/scripts/imco\_functions.R"))  
  
#initialize  
hemis <- c("lh", "rh")  
permNum <- 1000  
yeo\_num <- 7  
models = c("gam\_sex", "pos\_gam\_sex", "neg\_gam\_sex", "lm\_exec\_accuracy", "pos\_lm\_exec\_accuracy", "neg\_lm\_exec\_accuracy", "gam\_exec\_accuracy", "pos\_gam\_exec\_accuracy", "neg\_gam\_exec\_accuracy")  
#models = c("lm\_exec\_accuracy", "pos\_lm\_exec\_accuracy", "neg\_lm\_exec\_accuracy")  
################  
### Read in matrices   
for (model in models) {  
 #{lh and rh}\_gam\_sex\_t\_fdr05 -> actual results  
 lh\_t\_fdr05\_results <- read.table(paste0(homedir, "/baller/results/coupling\_accuracy/lh\_", model, "\_t\_fdr05\_Yeo7\_1\_0\_-1.csv"))  
 rh\_t\_fdr05\_results <- read.table(paste0(homedir, "/baller/results/coupling\_accuracy/rh\_", model, "\_t\_fdr05\_Yeo7\_1\_0\_-1.csv"))  
   
 #spins  
 lh\_spin <- t(read.table(paste0(homedir, "/baller/results/coupling\_accuracy/spin\_test\_results/lh\_spin\_test\_", model,"\_output.csv"), sep = ","))  
 rh\_spin <- t(read.table(paste0(homedir, "/baller/results/coupling\_accuracy/spin\_test\_results/rh\_spin\_test\_", model,"\_output.csv"), sep = ","))  
   
 #bring together, with original values as first column  
 lh\_act\_results\_and\_spin <- cbind(lh\_t\_fdr05\_results, lh\_spin)  
 rh\_act\_results\_and\_spin <- cbind(rh\_t\_fdr05\_results, rh\_spin)  
   
 #grab list of yeo 7 networks in fsaverage5 space  
 yeo\_networks <- get\_parcel\_mapping\_yeo(yeo\_num)  
   
 #separate into right and left  
 lh\_yeo\_network <- yeo\_networks[[1]]  
 rh\_yeo\_network <- yeo\_networks[[2]]  
   
 #count up number of vertices per network  
 lh\_yeo\_network\_count\_table <- table(lh\_yeo\_network)  
 rh\_yeo\_network\_count\_table <- table(rh\_yeo\_network)  
   
 #multiply yeo network x spin test  
 lh\_spinxyeo <- lh\_act\_results\_and\_spin\*lh\_yeo\_network  
 rh\_spinxyeo <- rh\_act\_results\_and\_spin\*rh\_yeo\_network  
   
 #proportions  
 #go through each hemisphere, go through each perm, and go through each network  
   
 lh\_hemi\_spin\_proportions <- data.frame(matrix(nrow = yeo\_num, ncol = (permNum + 1)))  
 rh\_hemi\_spin\_proportions <- data.frame(matrix(nrow = yeo\_num, ncol = (permNum + 1)))  
 for (hemi in hemis){  
  
 for (perm in 1:(permNum + 1)){  
   
 for (network in 1:yeo\_num){  
   
 #to evaluate  
   
 #number of vertices within network that are fdr corrected  
 num\_pos\_to\_parse<- paste0("length(which(", hemi, "\_spinxyeo[", perm, "] == ", network, "))")  
   
 num\_vertices\_in\_spin <- eval(parse(text = as.character(num\_pos\_to\_parse)))  
   
   
 #number of vertices within network that are negative (i.e., medial wall)  
 num\_neg\_to\_parse <- paste0("length(which(", hemi, "\_spinxyeo[", perm, "] == -", network, "))")  
   
 num\_neg <- eval(parse(text = as.character(num\_neg\_to\_parse)))  
   
   
 #total number of vertices in normal network  
 total\_possible\_to\_parse <- paste0(hemi, "\_yeo\_network\_count\_table[", network, "]")  
   
 total\_possible <- eval(parse(text = as.character(total\_possible\_to\_parse)))  
   
   
 #proportion of vertices within network , with denominator being total possible by # in medial wall  
 proportion\_potential\_vertices <- num\_vertices\_in\_spin/(total\_possible - num\_neg)  
   
   
 #store in matrix  
 storing\_to\_parse <- paste0(hemi, "\_hemi\_spin\_proportions[", network, ",", perm, "] = ", proportion\_potential\_vertices)  
  
 eval(parse(text = as.character(storing\_to\_parse)))  
 }  
 }  
 }  
   
 write.table(lh\_hemi\_spin\_proportions, file = paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/lh\_spin\_test\_", model, "\_proportions.csv"), sep = ",", col.names = F, row.names = F)  
 write.table(rh\_hemi\_spin\_proportions, file = paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/rh\_spin\_test\_", model, "\_proportions.csv"), sep = ",", col.names = F, row.names = F)  
#then plot  
}

\*\* violin plotting the spin tests and getting stats \*\*

From /project/imco/baller/scripts/spin\_test/violin\_plots.R . This provides a platform for analysis and visualization of the permuted results. It calls a few functions from the imco\_functions.R script, so you must source it (uncomment that line) in this section if running independtly. Otherwise, it should load if running this document start to finish.

####### Violin Plots ###  
  
  
#### Author: Erica Baller  
  
### 3/16/2021  
  
###pre: requires that you have run the spin tests for your models, including the positive and negative spin tests  
###post: 2 violin plots per model. Violin plot x axis - yeo networks, y axis - proportion of vertices within networks. dotted lines - mean from permutations  
 #### plot 1 will contain a thick black line for the \*actual values from your analysis so you can compare how far above or below your value is from permuted  
 #### plot 2 will contain red lines detailing where the actual values from your analysis from the POSITIVE domain, blue for NEGATIVE  
 #### You will also get a table that tells you the p value (uncorrected) for each network within each model.  
 #### for my final plot, I added a \* to these manually  
### uses: Creates violin plots and analyses to help make sense of the results from permutation spin tests. The question we are asking is:  
 ### what is the likelihood that the number of vertices within a network is significant rather than due to chance  
### dependencies: Using R 3.6.3 but any R will do. Libraries to include listed below  
  
library(tidyr)  
library(ggplot2)  
library(reshape)  
  
#################  
### set home directory  
homedir <- "/Users/eballer/BBL/imco/pmacs/PMACS\_remote"  
#homedir <- "/project/imco"  
source(paste0(homedir, '/baller/scripts//imco\_functions.R'))  
  
#outdir\_name <- "spin\_stats\_gam\_sex\_lm\_exec\_acc\_all\_pos\_neg"  
outdir\_name <- "spin\_stats\_gam\_sex\_gam\_exec\_acc\_all\_pos\_neg"  
  
models\_for\_stats = c("gam\_age", "pos\_gam\_age", "neg\_gam\_age", "gam\_sex", "pos\_gam\_sex", "neg\_gam\_sex","lm\_exec\_accuracy", "pos\_lm\_exec\_accuracy", "neg\_lm\_exec\_accuracy", "gam\_exec\_accuracy", "pos\_gam\_exec\_accuracy", "neg\_gam\_exec\_accuracy", "mean\_coupling")  
models\_for\_plots = c("gam\_age", "gam\_sex", "lm\_exec\_accuracy", "gam\_exec\_accuracy")  
  
network\_names <- c("VIS", "MOT", "DA", "VA", "LIM", "FP", "DM")  
num\_spins = 2000  
#get Yeo colors from function, these values were set manually - this can be obtained through https://surfer.nmr.mgh.harvard.edu/fswiki/CorticalParcellation\_Yeo2011  
 #I typed the RGB values into a rgb -> hex converter and stored the values here. Works!  
  
yeo\_colors <- get\_yeo7\_colors()  
  
#for storing statistics at the end  
stats <- data.frame(matrix(nrow = 7, ncol = length(models\_for\_stats)))  
names(stats) <- models\_for\_stats  
row.names(stats) <- network\_names  
  
for (model in models\_for\_stats) {  
   
 if (model == "mean\_coupling") {  
 lh\_spin\_df <- data.frame(read.table(paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/lh\_spin\_test\_mean\_coupling.csv"), sep = ","))  
 rh\_spin\_df <- data.frame(read.table(paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/rh\_spin\_test\_mean\_coupling.csv"), sep = ","))  
 } else {  
 lh\_spin\_df <- data.frame(read.table(paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/lh\_spin\_test\_", model, "\_proportions.csv"), sep = ","))  
 rh\_spin\_df <- data.frame(read.table(paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/rh\_spin\_test\_", model, "\_proportions.csv"), sep = ","))  
 }  
   
   
 #take means of left and right  
 actual\_results <- (lh\_spin\_df[,1] + rh\_spin\_df[,1])/2  
   
 #dataframes for all spins  
 spin\_without\_target\_col <- cbind(lh\_spin\_df[,2:1001],rh\_spin\_df[,2:1001])  
   
 ####### Stats ########  
 for (i in 1:7){  
 #store p values  
 #equivalen to: stats$model[i] <- (length(which(spin\_without\_target\_col[i,] > actual\_results[i]))/2000)  
 eval(parse(text=as.character(paste0("stats$", model, "[", i ,"] <- (length(which(spin\_without\_target\_col[",i, ",] > actual\_results[", i, "]))/", num\_spins, ")"))))  
 }  
}  
  
print(stats)  
write.csv(stats, paste0(homedir, "/baller/results/coupling\_accuracy/spin\_test\_results/", outdir\_name, ".csv"))  
  
violin\_plot\_means\_mean\_coupling(homedir = homedir, network\_names = network\_names, num\_spins = num\_spins)  
violin\_plot\_pos\_and\_neg\_lines(homedir = homedir, models = models\_for\_plots, network\_names = network\_names, num\_spins = num\_spins)  
violin\_plot\_means(homedir = homedir, models = models\_for\_plots, network\_names = network\_names, num\_spins = num\_spins)

Mean coupling violin plots

A picture containing diagram

Description automatically generated

Gam age violin plots

Chart

Description automatically generated with medium confidence

Gam sex violin plots

Shape

Description automatically generated

Gam exec accuracy plots

Chart, histogram

Description automatically generated

Lastly, here is the content for the imco functions script. Enjoy!

\*\* imco\_functions.R \*\*

require('visreg')  
require('mgcv')  
require('tableone')  
require('dplyr')  
  
make\_demographics\_table<- function(data\_frame) {  
   
 #subset demographics  
 listVars <- c("Age", "Sex", "Race", "Maternal Ed") #Race 1 = caucasian, Maternal Ed = years, age = years  
 demo <- data.frame(data\_frame$ageAtScan1, data\_frame$sex, data\_frame$race2, data\_frame$medu1)  
 names(demo) <- c(listVars)  
   
 #Change categorical values to have names  
 demo$Race <- ifelse(demo$Race == 1, "Caucasian", "Non-caucasian")  
 demo$Sex <- ifelse(demo$Sex == 1, "Male", "Female")  
   
 #Define Categorical Variables  
 cat\_variables <- c("Sex", "Race")  
 title <- paste0("IMCO Demographics, n = ", dim(demo)[1])  
   
 #create demographics table  
 demo\_table <- CreateTableOne(vars = listVars, data = demo, factorVars = cat\_variables)  
 print(demo\_table, showAllLevels = TRUE)  
}  
  
get\_parcel\_mapping\_yeo <- function(parcel\_num){  
   
 #pre: input parcel #, either 7 or 17  
 #post: list lh and rh yeo networks that map onto code #s  
 #uses: easy way to translate the weird numerical maps in fsaverage 5 space into something we are more familiar with  
 #dependencies: Any R will do, I used 3.2.5  
   
 ## Set Yeo info  
 #### set # parcels in case I want to do 7 or 17 or something else in the future  
 parcel\_type = "Yeo"   
 parcel\_num = parcel\_num   
 input\_parcel\_array\_length = 10242  
   
 # read in yeo fsaverage5 vectors  
 parcelID <- read.csv(paste0(homedir, "/baller/processed\_data/yeo\_network\_data/NetworkIDnumbers", parcel\_type, parcel\_num, ".csv"), header = F)  
 parcelName <- t(read.csv(paste0(homedir, "/baller/processed\_data/yeo\_network\_data/NetworkNames", parcel\_type, parcel\_num, ".csv"), header = F))  
   
 lh\_parcel\_nums <- read.csv(paste0(homedir, "/baller/processed\_data/yeo\_network\_data/lh\_", input\_parcel\_array\_length, "\_vertex\_nums\_", parcel\_type, parcel\_num, ".csv"), header = F)  
 rh\_parcel\_nums <- read.csv(paste0(homedir, "/baller/processed\_data/yeo\_network\_data/rh\_", input\_parcel\_array\_length, "\_vertex\_nums\_", parcel\_type, parcel\_num, ".csv"), header = F)  
   
 # map Yeo numbers to parcels  
 #make a column of numbers for mapping  
 parcelID$network\_num <- c(1:dim(parcelID)[1])  
   
 #add extra row to parcelID, not clear why this didn't come from Yeo labels, maybe cerebellum?... 8 will equal 65793  
 # comment this out if not using yeo   
 parcelID<- rbind(parcelID, c(65793, 8))  
   
 #make vector for lh and rh with mapping  
 lh\_numerical\_map <- lh\_parcel\_nums  
 rh\_numerical\_map <- rh\_parcel\_nums  
   
 #foreach vertex, which contains a bunch of numbers, match it to the appropriate column, and take the network num (i.e. yeo 2, which would correspond to Motor), associated with it  
 lh\_numerical\_map[] <- lapply(lh\_parcel\_nums, function(x) parcelID$network\_num[match(x, parcelID$V1)])  
 rh\_numerical\_map[] <- lapply(rh\_parcel\_nums, function(x) parcelID$network\_num[match(x, parcelID$V1)])  
  
 lh\_and\_rh\_numerical\_map\_list <- list(lh\_numerical\_map$V1, rh\_numerical\_map$V1)  
 return(lh\_and\_rh\_numerical\_map\_list)  
   
}  
  
melt\_df\_for\_violin\_plot\_yeo7 <- function(df, network\_names, num\_spins){  
 #melt dataframe so it is in a good format for violin plotting.  
 #melt df so it is in a good position to be plotted  
 melted\_df\_network\_name <- rep(x = network\_names, each = num\_spins)  
 melted\_df\_network\_num <- rep(x = seq(1:7), each = num\_spins)  
 melted\_df\_spin\_results <- rbind(t(df[1,]),   
 t(df[2,]),  
 t(df[3,]),  
 t(df[4,]),  
 t(df[5,]),  
 t(df[6,]),  
 t(df[7,]))  
   
 melted\_df <- as.data.frame(cbind(melted\_df\_network\_name, melted\_df\_network\_num,melted\_df\_spin\_results))  
 names(melted\_df) <- c("network\_name", "network\_num", "spin")  
 melted\_df$spin <- as.numeric(as.character(melted\_df$spin))  
 return(melted\_df)  
}  
  
violin\_plot\_pos\_and\_neg\_lines <- function (homedir, models, network\_names, num\_spins){  
 for (model in models) {  
  
 print(model)  
 #for storing statistics at the end  
 lh\_spin\_df <- data.frame(read.table(paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/lh\_spin\_test\_", model, "\_proportions.csv"), sep = ",")  
 )  
 lh\_spin\_pos <-data.frame(read.table(paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/lh\_spin\_test\_pos\_", model, "\_proportions.csv"), sep = ",")  
 )  
 lh\_spin\_neg <-data.frame(read.table(paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/lh\_spin\_test\_neg\_", model, "\_proportions.csv"), sep = ",")  
 )  
 rh\_spin\_df <- data.frame(read.table(paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/rh\_spin\_test\_", model, "\_proportions.csv"), sep = ",")  
 )  
 rh\_spin\_pos <-data.frame(read.table(paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/rh\_spin\_test\_pos\_", model, "\_proportions.csv"), sep = ",")  
 )  
 rh\_spin\_neg <- data.frame(read.table(paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/rh\_spin\_test\_neg\_", model, "\_proportions.csv"), sep = ",")  
 )  
   
 #take means of left and right  
 actual\_results <- (lh\_spin\_df[,1] + rh\_spin\_df[,1])/2  
 actual\_results\_pos <- ((lh\_spin\_pos[,1] + rh\_spin\_pos[,1])/2)  
 actual\_results\_neg <- ((lh\_spin\_neg[,1] + rh\_spin\_neg[,1])/2)  
   
 # print(actual\_results)  
 # print(actual\_results\_pos)  
 # print(actual\_results\_neg)  
 #dataframes for all spins, as well as positive and negative  
 spin\_without\_target\_col <- cbind(lh\_spin\_df[,2:1001],rh\_spin\_df[,2:1001])  
 # print(min(spin\_without\_target\_col))  
 # print(spin\_without\_target\_col[4,5:10])  
   
 melted\_df <- melt\_df\_for\_violin\_plot\_yeo7(spin\_without\_target\_col, network\_names, num\_spins)  
   
 # print(melted\_df[1:4,2:3])  
 #with mean lines, different fonts  
 #save images  
 plot\_violin <- ggplot(melted\_df, aes(x = factor(network\_name, level = network\_names), y = spin, fill = network\_name)) +   
 scale\_fill\_manual(values=yeo\_colors) +   
 geom\_violin(trim = TRUE) +   
 xlab("Yeo 7 Network") + ylab(paste0("Proportion")) +  
 ylim(0,NA) +   
 geom\_violin(trim=FALSE) +   
 theme\_classic() +   
 theme(legend.position = "none",  
 legend.title = element\_blank(),  
 axis.text.x = element\_text(size = 10, colour = "black"),  
 axis.text.y = element\_text(size = 10, colour = "black"),  
 axis.title.y = element\_text(size = 10),  
 axis.title.x = element\_blank(),  
 plot.title = element\_text(size = 10)) +  
 stat\_summary(fun.y = mean, geom = "errorbar",   
 aes(ymax = ..y.., ymin = ..y.., group = factor(network\_name)),  
 width = 0.5, linetype = "dashed", position = position\_dodge(0.9)) +   
 geom\_segment(aes(x = 0.5, y = actual\_results\_pos[1], xend = 1.5, yend = actual\_results\_pos[1]), color="red") +   
 geom\_segment(aes(x = 1.5, y = actual\_results\_pos[2], xend = 2.5, yend = actual\_results\_pos[2]), color="red") +  
 geom\_segment(aes(x = 2.5, y = actual\_results\_pos[3], xend = 3.5, yend = actual\_results\_pos[3]), color="red") +  
 geom\_segment(aes(x = 3.5, y = actual\_results\_pos[4], xend = 4.5, yend = actual\_results\_pos[4]), color="red") +  
 geom\_segment(aes(x = 4.5, y = actual\_results\_pos[5], xend = 5.5, yend = actual\_results\_pos[5]), color="red") +  
 geom\_segment(aes(x = 5.5, y = actual\_results\_pos[6], xend = 6.5, yend = actual\_results\_pos[6]), color="red") +  
 geom\_segment(aes(x = 6.5, y = actual\_results\_pos[7], xend = 7.5, yend = actual\_results\_pos[7]), color="red") +  
 geom\_segment(aes(x = 0.5, y = actual\_results\_neg[1], xend = 1.5, yend = actual\_results\_neg[1]), color="blue") +   
 geom\_segment(aes(x = 1.5, y = actual\_results\_neg[2], xend = 2.5, yend = actual\_results\_neg[2]), color="blue") +  
 geom\_segment(aes(x = 2.5, y = actual\_results\_neg[3], xend = 3.5, yend = actual\_results\_neg[3]), color="blue") +  
 geom\_segment(aes(x = 3.5, y = actual\_results\_neg[4], xend = 4.5, yend = actual\_results\_neg[4]), color="blue") +  
 geom\_segment(aes(x = 4.5, y = actual\_results\_neg[5], xend = 5.5, yend = actual\_results\_neg[5]), color="blue") +  
 geom\_segment(aes(x = 5.5, y = actual\_results\_neg[6], xend = 6.5, yend = actual\_results\_neg[6]), color="blue") +  
 geom\_segment(aes(x = 6.5, y = actual\_results\_neg[7], xend = 7.5, yend = actual\_results\_neg[7]), color="blue")  
 # ggtitle(paste0("Spin Test Perm: ", model))  
 ggsave(plot=plot\_violin, filename = paste0(homedir, "/baller/results/images/spin\_", model, "\_pos\_and\_neg\_lines\_t\_fdr05.png"), width = 4.81, height = 4.81)  
 }  
}  
  
violin\_plot\_means <- function (homedir, models, network\_names, num\_spins){  
 for (model in models) {  
 #for storing statistics at the end  
 lh\_spin\_df <- data.frame(read.table(paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/lh\_spin\_test\_", model, "\_proportions.csv"), sep = ",")  
 )  
 rh\_spin\_df <- data.frame(read.table(paste0(homedir, "/baller/results/coupling\_accuracy//spin\_test\_results/rh\_spin\_test\_", model, "\_proportions.csv"), sep = ",")  
 )  
   
   
 #take means of left and right  
 actual\_results <- (lh\_spin\_df[,1] + rh\_spin\_df[,1])/2  
   
 #dataframes for all spins  
 spin\_without\_target\_col <- cbind(lh\_spin\_df[,2:1001],rh\_spin\_df[,2:1001])  
 melted\_df <- melt\_df\_for\_violin\_plot\_yeo7(spin\_without\_target\_col, network\_names, num\_spins)  
   
 #with mean lines, different fonts  
 #save images  
 plot\_violin <- ggplot(melted\_df, aes(x = factor(network\_name, level = network\_names), y = spin, fill = network\_name)) +   
 scale\_fill\_manual(values=yeo\_colors) +   
 geom\_violin(trim = TRUE) +   
 xlab("Yeo 7 Network") + ylab(paste0("Proportion")) +  
 geom\_violin(trim=FALSE) +   
 ylim(0,NA) +   
 theme\_classic() +   
 theme(legend.position = "none",  
 legend.title = element\_blank(),  
 axis.text.x = element\_text(size = 10, colour = "black"),  
 axis.text.y = element\_text(size = 10, colour = "black"),  
 axis.title.y = element\_text(size = 10),  
 axis.title.x = element\_blank(),  
 plot.title = element\_text(size = 10)) +  
 stat\_summary(fun.y = mean, geom = "errorbar",   
 aes(ymax = ..y.., ymin = ..y.., group = factor(network\_name)),  
 width = 0.5, linetype = "dashed", position = position\_dodge(0.9)) +   
 #geom\_boxplot(width = 0.15, position = position\_dodge(0.9)) +   
 geom\_segment(aes(x = 0.5, y = actual\_results[1], xend = 1.5, yend = actual\_results[1])) +   
 geom\_segment(aes(x = 1.5, y = actual\_results[2], xend = 2.5, yend = actual\_results[2])) +  
 geom\_segment(aes(x = 2.5, y = actual\_results[3], xend = 3.5, yend = actual\_results[3])) +  
 geom\_segment(aes(x = 3.5, y = actual\_results[4], xend = 4.5, yend = actual\_results[4])) +  
 geom\_segment(aes(x = 4.5, y = actual\_results[5], xend = 5.5, yend = actual\_results[5])) +  
 geom\_segment(aes(x = 5.5, y = actual\_results[6], xend = 6.5, yend = actual\_results[6])) +  
 geom\_segment(aes(x = 6.5, y = actual\_results[7], xend = 7.5, yend = actual\_results[7])) +  
 # ggtitle(paste0("Spin Test Perm: ", model))  
 ggsave(plot=plot\_violin, filename = paste0(homedir, "/baller/results/images/spin\_", model, "\_t\_fdr05.png"), width = 4.81, height = 4.81)  
 }  
}  
  
get\_yeo7\_colors <- function() {  
 yeo\_colors <- c(  
 `VIS` = "#781286",  
 `MOT` = "#4682b4",  
 `DA` = "#00760e",  
 `VA` = "#c43afa",  
 `LIM` = "#dcf8a4",  
 `FP` = "#e69422",  
 `DM` = "#cd3e56")  
 return(yeo\_colors)  
}