PFAS and the Metabolome

Amber M Hall and Elvira Fleury

2023-08-18

Introduction

This code was written for the paper "Associations of a Prenatal Serum Per- and Polyfluoroalkyl Substance Mixture with the Cord Serum Metabolome in the HOME Study" by Hall et al. Details for this study can be found here (**Paper under review**) In brief, the purpose of this research was to evaluate associations between a mixture of 4 PFAS (PFOS, PFOA, PFNA, and PFHxS) and their corresponding metabolites. The metabolites were identified using a non-targeted analysis (NTA) approach. Additionally, we ran a sensitivity analysis for all analyses where we included MeFOSAA to evaluate the impact of this PFAS on our results.

This research was conducted at Brown University under the mentorship of Dr. Joseph Braun. The data used for this study was from the Health Outcomes and Measures of the Environment (HOME) Study (Braun et al.)

The code provided here is the corresponding code for our analyses. This code does the following:

---

1. Data Import and Cleaning
Cleans and structures the data for analyses.

2. Metabolome Wide Association Study (MWAS)
a. Conducts a metabolome-wide association study (MWAS) using quantile-based g-computation. This evaluates associations between this 4-PFAS mixture and each metabolite in the mixture.
b. Conducts an MWAS using linear regression to identify individual associations between our PFAS and each metabolite. The purpose of this was to observe the potential impact of the individual metabolites on the mixture. c. Conducts an MWAS using quantile-based g-computation to evaluate associations between the 5-PFAS mixture (4PFAS mixture plus MeFOSAA) and each metabolite in the mixture. This was a sensitivity analysis to determine the impact of MeFOSAA on our final results.

3. Pathway Enrichment Analysis (PEA)
a. Performs a PEA using mummichog via Metaboanalyst 5.0 to identify the association between our 4-PFAS mixture and specific metabolic pathways. This dataset is created

during the MWAS.
b. Performs a PEA using mummichog via Metaboanalyst 5.0 to identify the association between each individual PFAS and specific metabolic pathways. This dataset is created during the MWAS. The purpose of this was to observe the potential impact of the individual metabolites on the mixture.
c. Performs a PEA using mummichog via Metaboanalyst 5.0 to identify the association between a 5-PFAS mixture and specific metabolic pathways. This dataset is created during the MWAS. This was a sensitivity analysis to determine the impact of MeFOSAA on our final results.

---

1. Data Import and Data Cleaning

1.1. Needed Libraries

Importing all needed libraries If you do not have these in you library already, you will need to install theme before you can run this code. An example of code to install a library is as follows:

install.packages("tidyverse")

where "tidyverse" can be replaced with any of these packages except MetaboanalystR. MetaboanalystR installation instructions can be found here (MetaboAnalystR).

```
# Import needed libraries
# Load necessary libraries with suppressed messages

#Creating an object named libraries with all of the libraries I want to load
libraries <- c("knitr", "tidyverse", "writexl", "qgcomp", "MetaboAnalystR",
"fitdistrplus", "RJSONIO")

#Loading the libraries and suppressing all error messages and the function
return value
invisible(sapply(libraries, library, character.only = TRUE))

#Removing the object with the needed libraries
rm(libraries)
```

1.2. Importing Needed Data

For this code chunk I have 2 sets of files:

Files

1. The main C18 and HILIC files (named C18 and HILIC)
2. The map C18 and HILIC files (named C18_map and HILIC_map)

The main files contain the specific ion abundances for each participant. The map files contain the participant IDs (assigned to participants in the study) and a set of metabolomic specific IDs. The metabolomic specific IDs are used in the main C18 and HILIC files where all other imported files use the participant IDs.

For the main files, the headers are as follows*:

Main File Headers

1. mz: the mass to charge ratio.
2. time: retention time.
3. mode: labeled 1 for C18 and 2 for HILIC files 4+. a column for each participant containing the ion abundance for each metabolite (referred to as a 'feature'). The header name for each row is the participant ID. An example of a participant ID is 'C18_1'

*The number of columns in the main files are equal to the number of participants + 3

For the map files, the headers are as follows:

Map File Headers

1. participant_ID. The assigned participant ID for the data in this study.
2. "HILIC_ID" for HILIC data and "C18_ID" for C18 data. IDs that are specific only to the metabolomics data

---

Although not shown in this code, I have compared the C18 and HILIC map files to ensure they contain the same participant IDs.

For this study, both HILIC and C18 datasets have already:

1. Been batch corrected using WaveICA 2.0
2. Had values removed if CV >30% (within triplicate)
3. Had values removed if non-detect intensities >20%
4. Had missing values imputed using the minimum value (per feature)/sqrt(2)

To run this code, you will need to edit these files to include your own dataset.

```
# Importing the C18 map file that identifies the IDs
C18_map <- read.csv("Data/C18_map_clean.csv", header = TRUE) # INSERT YOUR
FILE PATHWAY HERE

# Importing the map file that identifies the IDs
```

```
HILIC_map <- read.csv("Data/HILIC_map_clean.csv", header = TRUE) # INSERT
YOUR FILE PATHWAY HERE

# Importing the batch-corrected C18 data
C18 <- read.csv ('Data/C18_clean.csv', header = TRUE) # INSERT YOUR FILE
PATHWAY HERE

# Importing the batch-corrected HILIC data
HILIC <- read.csv ('Data/HILIC_clean.csv', header = TRUE) # INSERT YOUR FILE
PATHWAY HERE
```

For this code chunk I import the PFAS data. This data will be named 'pfas_only'.

For 'pfas_only', the headers are as follows:

PFAS Dataset Headers

1. 'participant_ID': The assigned participant ID for the data in this study. This will be used to merge with the metabolomics data 'participant ID later.
2. pfoa_log2: log-2 transformed serum PFOA concentrations (ng/mL)
3. pfos_log2: log-2 transformed serum PFOS concentrations (ng/mL)
4. pfna_log2: log-2 transformed serum PFNA concentrations (ng/mL)
5. pfhxs_log2: log-2 transformed serum PFHxS concentrations (ng/mL)
6. me_pfosa_acoh_log2: log-2 transformed serum MeFOSAA concentrations (ng/mL)

To run this code, you will need to edit these files to include your own dataset.

```
# Importing the PFAS file
pfas_only <- read.csv("Data/pfas_clean.csv", header = TRUE)   %>% # INSERT
YOUR FILE PATHWAY HERE
    mutate(participant_ID = as.character(participant_ID)) #Converting this to
a character variable
```

For this code chunk I import the covariate data. This data will be named 'cov'.

For 'cov', the headers are as follows:

Covariate Dataset Headers

1. 'participant_ID': The assigned participant ID for the data in this study.
2. momagedeliv: maternal age at delivery (years)
3. ct_16w: log10 transformed serum cotinine concentrations measured at 16 weeks' gestation (ng/mL)
4. mom_race: maternal race (coded as "White, not-Hispanic" and "Other race")

5. midincome: family income in USD (continuous).
6. parity: parity (coded as “Nulliparous” and “Parous”)

To run this code, you will need to edit these files to include your own dataset.

```
# Importing covariate data and converting used factor variables to factors
cov <- read.csv("Data/covariates_clean.csv", header = TRUE) # INSERT YOUR
FILE PATHWAY HERE
```

1.3. Creating Master Datasets

1.3.1. Exposure Variables and Covariates

The purpose of this code is to create a master dataset named ‘pfas’. I do this by merging the following datasets into a single dataset:

1. pfas_only: PFAS data
2. cov: Covariate data
3. C18_map: Contains metabolomics specific IDs and the study participant IDs for the C18 data.
4. HILIC_map: Contains metabolomics specific IDs and the study participant IDs for the HILIC data.

```
# Creating a 'master' PFAS dataset called 'pfas' with both pfas and covariate
information
pfas <- pfas_only %>%
  merge(cov, by = "participant_ID") %>%
  merge(C18_map, by = "participant_ID") %>%
  merge(HILIC_map, by = "participant_ID")

# Removing unnecessary dataframes from the global environment
    #map files are used later on so we will keep these for now.
rm(pfas_only, cov)
```

1.3.2. Feature Table

The purpose of this chunk is to create a dataframe named ‘feature_table’. feature_table will contain only participants with both PFAS and covariate data (i.e. will be a subset of the full dataframe). This feature table will also contain both C18 and HILIC data stacked on top of each other. This new feature table will have the following columns

---

Feature Table Headers

1. mz: the mass to charge ratio.
2. time: retention time.
3. mode: labeled 1 for C18 and 2 for HILIC files 4+. a column for each participant containing the ion abundance for each metabolite (referred to as a ‘feature’). The

header name for each row is the participant ID. An example of a participant ID is 'C18_1'

The number of columns in the new dataframe will be the number of participants + 3

```
# Creating a 'participant' file that is a list of participants.
participants <- unique(pfas["participant_ID"])

# Join the C18_map data frame to the participants data frame
participants <- inner_join(participants, C18_map, by = "participant_ID")

# Join the HILIC_map data frame to the participants data frame
participants <- inner_join(participants, HILIC_map, by = "participant_ID")

#Creating a string of characters called 'C18_ID' that contain all of the
metabolomic-specific IDs for C18 as well as mz, rt, and mode
C18_ID <- c("mz", "time", "mode", participants$C18_ID)

# Selecting only participants from the C18 file that contain both pfas and
covariate information
C18 <- subset(C18, select = C18_ID)

#Creating a general metabolomics-specific ID rather than one specific for C18
data
    #As of now participant 1 will correspond to both C18_1 and HILIC_1. Here
I drop the C18 portion and replace it with ID changing the value to ID_1.
This will allow me to stack the HILIC and C18 data ontop of one another.
colnames(C18) <- gsub("C18","ID", colnames(C18))

#Creating a list called 'HILIC_ID' that contain all of the metabolomic-
specific IDs for HILIC as well as mz, rt, and mode
HILIC_ID <- c("mz", "time", "mode", participants$HILIC_ID)

# Selecting only participants from the HILIC file that contain both pfas and
covariate information
HILIC <- subset(HILIC, select = HILIC_ID)

#Creating a general metabolomics-specific ID rather than one specific for
HILIC data
colnames(HILIC) <- gsub("HILIC","ID", colnames(HILIC))

# Creating a 'master' dataframe called 'feature' with both the C18 and HILIC
data stacked on top of one another
feature_table <- bind_rows(C18, HILIC)

#Dropping unneeded datafiles and values
rm(participants, C18_ID, HILIC_ID, C18_map, HILIC_map, C18, HILIC)
```

1.4. Splitting the Dataset for Analysis

The purpose of this chunk is to create a list that has split each individual feature into their own tibble. Each individual tibble will contain the ion abundance for that metabolite as well as the pfas and covariate data. The purpose of doing this is so that we can run each individual tibble through the analyses individually in our analyses steps.

```
# Adding a feature new column named 'feature_id'. This columns contains
numbers 1,2,3,....N. This column will be used later for splitting the feature
table
feature_table <- cbind(feature_id = 1:nrow(feature_table), feature_table)

# I pivot the dataset to a 'long file'. This contained 6 values (feature_ID,
mz, time, mode, name [metabolomic-specific participant_ID], and value [ion
abundance]).
long_df <- feature_table %>% pivot_longer(!c(1:4))

# Replacing the C18 value names to a general ID to match the long_df
pfas <- pfas  %>%
  mutate(C18_ID = str_replace(C18_ID, "C18", "ID")) #Dropping the C18 to
merge with the long dataset

# Creating a long feature table to analyze for both the cardiometabolic and
the PFAS data
analysis_pfas <- inner_join(long_df, pfas, by= c("name"= "C18_ID"))

split_pfas <- split(analysis_pfas, f=analysis_pfas$feature_id)

# Removing unnecessary dataframes
rm(long_df, analysis_pfas, feature_table, pfas)
```

2. Metabolome Wide Association Study (MWAS)

2.1. Quantile-based G-computation (4 PFAS)

This chunk of code is our primary Metabolome Wide Association Study (MWAS) analysis. Here, we evaluate whether the joint effect of our 4 PFAS of interest (PFOS, PFOA, PFHxS, PFNA) are associated with the ion abundance in each feature. To accomplish this, we run each feature through a quantile-based g-computation model adjusted for parity, family income, maternal age, maternal race, and cotinine concentrations.

Quantile-based g-computation is discussed in the paper (Keil et.al).

Additional resources on quantile-based g-computation model are located in the Comprehensive R Archive Network (Cran) (link to Cran website).

The results of these models are outputted to 3 files:

Files Outputted from the 4-PFAS MWAS Code

1. a full or overall file that contains all the results*
2. a significant file that contains significant values (i.e. those with FDR values<.20)*
3. a metaboanalyst file that contains only the variables needed to import into metaboanalyst (mz, time, p-value, mode).

*contains 9 columns. These columns are: a. mz: the mass to charge ratio. b. time: retention time. c. psi: effect estimate for the joint effects of these 4 PFAS. For this model, psi represents the difference in feature intensity resulting from a simultaneous, one-quantile increase of all PFAS in the mixture d. 95%CIs: 95% confidence intervals e. p-value: unadjusted p-value (before FDR correction) f. mode: negative (C18) or positive (HILIC) g. FDR: False discovery rate (FDR) corrected p-value

---

You will need to specify your own file pathways to export the results. Furthermore, if you are adjusting for covariates, you will need to update the variables in the model.

```
# Creating a list of exposure names
Xnm <- c("pfoa_log2","pfos_log2","pfna_log2","pfhxs_log2")

# Writing a function that loops through each individual features and runs
quantile-based g-computation, adjusting for covariates.
find_psi_4PFAS <- function(list) {
  num_metabolites <- length(list)
  res_mat <- matrix(NA, nrow = num_metabolites, ncol = 8)
  colnames(res_mat) <- c("m.z", "r.t", "psi", "SE", "ci_lower", "ci_upper",
"p.value", "mode")
  for (i in 1:num_metabolites) {
    data_matrix <- as.matrix(list[[i]])
    # Quantile-based G-computation model
    res <- qgcomp.noboot(value ~ midincome + mom_race + momagedeliv + ct_16w
+ parity +
                           pfoa_log2 + pfos_log2 + pfna_log2 + pfhxs_log2,
                         expnms = Xnm, data = list[[i]], family = gaussian())
    # Outputting needed values
    res_mat[i, 1] <- as.numeric(list[[i]][1, 2]) # mz
    res_mat[i, 2] <- as.numeric(list[[i]][1, 3]) # time ie. r.t
    res_mat[i, 3] <- as.numeric(res$psi) # psi
    res_mat[i,4] <- as.numeric((res$ci[2]-res$ci[1])/3.92) #SE
    res_mat[i, 5:6] <- res$ci # upper and lower confidence interval
    res_mat[i, 7] <- as.numeric(res$pval[2]) # p-value
    res_mat[i, 8] <- as.numeric(list[[i]][1, 4]) # mode
  , eval=FALSE}
  # Applying the False Discovery Rate (FDR) correction
  res_df <- data.frame(res_mat)
  res_df$FDR <- p.adjust(res_df$p.value, method = 'BH')
  # Re-coding mode to 'negative' and 'positive'
  res_df$mode[res_df$mode == 1] <- "negative"
```

```
  res_df$mode[res_df$mode == 2] <- "positive"
  return(res_df)
, eval=FALSE}

# Creating files
qg_comp_4_FULL <- suppressWarnings(as.data.frame(find_psi_4PFAS(split_pfas)))
qg_comp_4_SIG <- qg_comp_4_FULL[qg_comp_4_FULL$FDR< 0.2,]
qg_comp_4_METABOANALYST <- subset(qg_comp_4_FULL, select=c("m.z",
"p.value","r.t", "mode"))

# Writing the files out
write.csv(qg_comp_4_FULL, 'Final Datasets/Full/QG_comp_full_4PFAS.csv',
row.names = TRUE)
write.csv(qg_comp_4_SIG, 'Final Datasets/Significant
only/QG_comp_sig_4PFAS.csv', row.names = TRUE)
write.csv(qg_comp_4_METABOANALYST, 'Final
Datasets/Metaboanalyst/QG_comp_full_METABOLOMICS_4PFAS.csv', row.names =
TRUE)

# Removing unneeded files and functions from the global environment
rm(Xnm, qg_comp_4_FULL, qg_comp_4_METABOANALYST, qg_comp_4_SIG,
find_psi_4PFAS)
```

2.2. Linear Regression Models for Individual PFAS

The purpose of this chunk is to evaluate whether each individual PFAS is association with ion abundance for each feature. I wrote a function named 'find_linear_pfas' that loops through each feature and runs a linear regression model, adjusted for covariates. This models evaluates each PFAS individually (PFOS, PFOA, PFNA, PFHxS, and MeFOSAA [MeFOSAA was for a sensitivity analysis]) with relation to ion abundance, adjusting for parity, family income, maternal age, maternal race, and cotinine concentrations. As a reminder, all PFAS have already been log2 transformed.

Files Outputted from the Single Pollutant PFAS MWAS Code

1. a full or overall file that contains all the results*
2. a significant file that contains significant values (i.e. those with FDR values<.20)*
3. a metaboanalyst file that contains only the variables needed to import into metaboanalyst (mz, time, p-value, mode).

*contains 9 columns. These columns are: a. mz: the mass to charge ratio. b. time: retention time. c. psi: effect estimate for the joint effects of these 4 PFAS. For this model, psi represents the difference in feature intensity resulting from a simultaneous, one-quantile increase of all PFAS in the mixture d. 95%CIs: 95% confidence intervals e. p-value: unadjusted p-value (before FDR correction) f. mode: negative (C18) or positive (HILIC) g. FDR: False discovery rate (FDR) corrected p-value

You will need to specify your own file pathways to export the results. Furthermore, if you are adjusting for covariates, you will need to update the variables in the model.

```
# Create the function
find_linear_pfas <- function(list, PFAS) {
  num_metabolites <- length(list)
  res_mat <- matrix(NA, nrow = num_metabolites, ncol = 8) # Add one more
column for standard error
  colnames(res_mat) <- c("m.z", "r.t", "Beta", "SE", "ci_lower", "ci_upper",
"p.value", "mode")
  for (i in 1:num_metabolites) {
    data_matrix <- as.matrix(list[[i]])
    # Fit linear regression model with specified PFAS
    formula_str <- paste0("value ~ ", PFAS, " + midincome + mom_race +
momagedeliv + ct_16w + parity")
    res <- lm(formula_str, data = list[[i]])
    # Fill in results matrix
    res_mat[i, 1] <- as.numeric(list[[i]][1, 2]) #mz
    res_mat[i, 2] <- as.numeric(list[[i]][1, 3]) #time
    res_mat[i, 3] <- res$coefficients[PFAS] #Beta estimate
    res_mat[i, 4] <- summary(res)$coefficients[PFAS, "Std. Error"] #standard
error
    res_mat[i, 5:6] <- confint(res, PFAS, level = 0.95) #upper and lower
confidence interval
    res_mat[i, 7] <- summary(res)$coefficients[PFAS, "Pr(>|t|)"] #p-value
    res_mat[i, 8] <- as.numeric(list[[i]][1, 4]) #mode (i.e. direction)
  , eval=FALSE}
  # Convert matrix to data frame and add FDR column
  res_df <- as.data.frame(res_mat)
  colnames(res_df) <- colnames(res_mat)
  res_df$FDR <- p.adjust(res_df$p.value, method = 'BH')
  # Recode mode column
  res_df$mode[res_df$mode == 1] <- "negative" #C18
  res_df$mode[res_df$mode == 2] <- "positive" #HILIC
  #Returns your wanted values
  return(res_df)
, eval=FALSE}

# Creating the datafiles
  # PFOA
  PFOA_Full <- find_linear_pfas(split_pfas, "pfoa_log2")
  PFOA_SIG <- PFOA_Full[PFOA_Full$FDR< 0.2,]
  PFOA_METABOANALYST <- subset(PFOA_Full, select=c("m.z", "p.value","r.t",
"mode"))

  # PFOS
  PFOS_Full <- find_linear_pfas(split_pfas, "pfos_log2")
  PFOS_SIG <- PFOS_Full[PFOS_Full$FDR< 0.2,]
```

```
  PFOS_METABOANALYST <- subset(PFOS_Full, select=c("m.z", "p.value","r.t",
"mode"))

  # PFNA
  PFNA_Full <- find_linear_pfas(split_pfas, "pfna_log2")
  PFNA_SIG <- PFNA_Full[PFNA_Full$FDR< 0.2,]
  PFNA_METABOANALYST <- subset(PFNA_Full, select=c("m.z", "p.value","r.t",
"mode"))

  # PFHxS
  PFHxS_Full <- find_linear_pfas(split_pfas, "pfhxs_log2")
  PFHxS_SIG <- PFHxS_Full[PFHxS_Full$FDR< 0.2,]
  PFHxS_METABOANALYST <- subset(PFHxS_Full, select=c("m.z", "p.value","r.t",
"mode"))

  # MEFOSAA
  MEFOSAA_Full <- find_linear_pfas(split_pfas, "me_pfosa_acoh_log2")
  MEFOSAA_SIG <- MEFOSAA_Full[MEFOSAA_Full$FDR< 0.2,]
  MEFOSAA_METABOANALYST <- subset(MEFOSAA_Full, select=c("m.z",
"p.value","r.t", "mode"))

# Writing the files out
  # PFOA
  write.csv(PFOA_Full, 'Final Datasets/Full/Individual PFAS/PFOA_Full.csv',
row.names = TRUE)
  write.csv(PFOA_SIG, 'Final Datasets/Significant only/Individual
PFAS/PFOA_SIG.csv', row.names = TRUE)
  write.csv(PFOA_METABOANALYST, 'Final Datasets/Metaboanalyst/Individual
PFAS/PFOA_METABOANALYST.csv',   row.names = TRUE)

  # PFOS
  write.csv(PFOS_Full, 'Final Datasets/Full/Individual PFAS/PFOS_Full.csv',
row.names = TRUE)
  write.csv(PFOS_SIG, 'Final Datasets/Significant only/Individual
PFAS/PFOS_SIG.csv', row.names = TRUE)
  write.csv(PFOS_METABOANALYST, 'Final Datasets/Metaboanalyst/Individual
PFAS/PFOS_METABOANALYST.csv', row.names = TRUE)
  # PFNA
  write.csv(PFNA_Full, 'Final Datasets/Full/Individual PFAS/PFNA_Full.csv',
row.names = TRUE)
  write.csv(PFNA_SIG, 'Final Datasets/Significant only/Individual
PFAS/PFNA_SIG.csv', row.names = TRUE)
  write.csv(PFNA_METABOANALYST, 'Final Datasets/Metaboanalyst/Individual
PFAS/PFNA_METABOANALYST.csv', row.names = TRUE)
# PFHxS
  write.csv(PFHxS_Full, 'Final Datasets/Full/Individual PFAS/PFHxS_Full.csv',
row.names = TRUE)
  write.csv(PFHxS_SIG, 'Final Datasets/Significant only/Individual
PFAS/PFHxS_SIG.csv', row.names = TRUE)
```

```
  write.csv(PFHxS_METABOANALYST, 'Final Datasets/Metaboanalyst/Individual
PFAS/PFHxS_METABOANALYST.csv', row.names = TRUE)

# MEFOSAA
  write.csv(MEFOSAA_Full, 'Final Datasets/Full/Individual
PFAS/ME_PFOSA_ACOH_Full.csv', row.names = TRUE)
  write.csv(MEFOSAA_SIG, 'Final Datasets/Significant only/Individual
PFAS/ME_PFOSA_ACOH_SIG.csv', row.names = TRUE)
  write.csv(MEFOSAA_METABOANALYST, 'Final Datasets/Metaboanalyst/Individual
PFAS/ME_PFOSA_ACOH_METABOANALYST.csv', row.names = TRUE)

# Removing unnecessary files and functions
rm(PFOA_Full, PFOA_SIG, PFOA_METABOANALYST,
   PFOS_Full, PFOS_SIG, PFOS_METABOANALYST,
   PFNA_Full, PFNA_SIG, PFNA_METABOANALYST,
   PFHxS_Full, PFHxS_SIG, PFHxS_METABOANALYST,
   MEFOSAA_Full, MEFOSAA_SIG, MEFOSAA_METABOANALYST,
  find_linear_pfas)
```

2.3. Sensitivity Analysis: Quantile-based G-computation (5 PFAS)

This is a sensitivity analysis to evaluate the impact of MeFOSAA on the original 4 PFAS mixture. Here, we evaluate whether the joint effect of these 5 PFAS of interest (PFOS, PFOA, PFHxS, PFNA, MeFOSAA) are associated with the ion abundance in each feature. To accomplish this, we run each feature through a quantile-based g-computation model adjusted for parity, family income, maternal age, maternal race, and cotinine concentrations.

Quantile-based g-computation is discussed in the paper (Keil et.al).

Additional resources on quantile-based g-computation model are located in the Comprehensive R Archive Network (Cran) (link to Cran website).

The results of these models are outputted to 3 files:

Files Outputted from the 5-PFAS MWAS Code

1. 'full' or overall file that contains all the results*
2. a significant file that contains significant values (i.e. those with FDR values<.20)*
3. a 'metaboanalyst' file that contains only the variables needed to import into metaboanalyst (mz, time, p-value, mode).

*contains 9 columns. These columns are: 1. mz: the mass to charge ratio. 2. time: retention time. 3. psi: effect estimate for the joint effects of these 4 PFAS. Psi represents the difference in feature intensity resulting from a simultaneous, one-quantile increase of all PFAS in the mixture 4. 95%CIs: 95% confidence intervals 5. p-value: unadjusted p-value (before FDR correction) 6. mode: negative (C18) or positive (HILIC) 7. FDR: False discover rate (FDR) corrected p-value

You will need to specify your own file pathways to export the results. Furthermore, if you are adjusting for covariates, you will need to update the variables in the model.

```
# Create a exposure list of the names of each of the 5 PFAS
Xnm <- c("pfoa_log2","pfos_log2","pfna_log2","pfhxs_log2",
"me_pfosa_acoh_log2")

# Writing a function that loops through each individual features and runs
quantile-based g-computation, adjusting for covariates.
find_psi_4PFAS_and_ME_PFOSA <- function(list) {
  num_metabolites <- length(list)
  res_mat <- matrix(NA, nrow = num_metabolites, ncol = 8)
  colnames(res_mat) <- c("m.z", "r.t", "psi", "SE", "ci_lower", "ci_upper",
"p.value", "mode")
  for (i in 1:num_metabolites) {
    data_matrix <- as.matrix(list[[i]])
    # Quantile G-comp model
    res <- qgcomp.noboot(value ~ midincome + mom_race + momagedeliv + ct_16w
+ parity +
                           pfoa_log2 + pfos_log2 + pfna_log2 + pfhxs_log2 +
                           me_pfosa_acoh_log2,
                         expnms = Xnm, data = list[[i]], family = gaussian())
    # Outputting needed values
    res_mat[i, 1] <- as.numeric(list[[i]][1, 2]) # mz
    res_mat[i, 2] <- as.numeric(list[[i]][1, 3]) # time ie. r.t
    res_mat[i, 3] <- as.numeric(res$psi) # psi
    res_mat[i,4] <- as.numeric((res$ci[2]-res$ci[1])/3.92) #SE
    res_mat[i, 5:6] <- res$ci # upper and lower confidence interval
    res_mat[i, 7] <- as.numeric(res$pval[2]) # p-value
    res_mat[i, 8] <- as.numeric(list[[i]][1, 4]) # mode
  , eval=FALSE}
  # Applying the FDR correction
  res_df <- data.frame(res_mat)
  res_df$FDR <- p.adjust(res_df$p.value, method = 'BH')
  # Recoding mode to 'megative' and 'positive'
  res_df$mode[res_df$mode == 1] <- "negative"
  res_df$mode[res_df$mode == 2] <- "positive"
  return(res_df)
, eval=FALSE}

# Creating files
qg_comp_5_FULL <-
suppressWarnings(as.data.frame(find_psi_4PFAS_and_ME_PFOSA(split_pfas)))
qg_comp_5_SIG <- qg_comp_5_FULL[qg_comp_5_FULL$FDR< 0.2,]
qg_comp_5_METABOANALYST <- subset(qg_comp_5_FULL, select=c("m.z",
"p.value","r.t", "mode"))

# Writing the files out
```

```
write.csv(qg_comp_5_FULL, 'Final
Datasets/Full/QG_comp_full_4PFAS_and_ME_PFOSA.csv', row.names = TRUE)
write.csv(qg_comp_5_SIG, 'Final Datasets/Significant
only/QG_comp_sig_4PFAS_and_ME_PFOSA.csv', row.names = TRUE)
write.csv(qg_comp_5_METABOANALYST, 'Final
Datasets/Metaboanalyst/QG_comp_full_METABOLOMICS_4PFAS_and_ME_PFOSA.csv',
row.names = TRUE)

# Removing unneeded files and functions from the global environment
rm(Xnm, qg_comp_5_FULL, qg_comp_5_METABOANALYST, qg_comp_5_SIG,
find_psi_4PFAS_and_ME_PFOSA, split_pfas)
```

3. Pathway Enrichment Analysis (PEA)

About MetaboanalystR

MetaboAnalystR is the R package associated with MetaboAnalyst. Information on Metaboanalyst can be found at (MetaboAnalyst). Additionally, information on MetaboAnalystR can be found at (MetaboanalystR).

3.1. Specifying Specific Adducts

For our pathway enrichment analysis (PEA), we first specify the adducts allowed for the study. This determination was made by our lead chemist Dr. Kate Manz. The purpose of adduct restriction is to restrict our results to the adducts that could potentially form based on our mobile phases and internal standards.

A full list of the adducts available for MetaboAnalyst can be found here (link to adduct list).

```
# Checking the version of MetaboAnalystR we are running.
#packageVersion("MetaboAnalystR")

# Creating a vector that contains the customized vectors for MetaboAnalyst.
add.vec <- c("M+FA-H [1-]","M-H [1-]","2M-H [1-]","M-H+O [1-]","M(C13)-H [1-
]",
             "2M+FA-H [1-]","M-3H [3-]","M-2H [2-]","M+ACN-H [1-]",
             "M+HCOO [1-]","M+CH3COO [1-]","M-H2O-H [1-]","M [1+]","M+H
[1+]",
             "M+2H [2+]","M+3H [3+]","M+H2O+H [1+]","M-H2O+H [1+]",
             "M(C13)+H [1+]","M(C13)+2H [2+]","M(C13)+3H [3+]","M-NH3+H
[1+]",
             "M+ACN+H [1+]","M+ACN+2H [2+]","M+2ACN+2H [2+]","M+3ACN+2H
[2+]",
             "M+NH4 [1+]","M+H+NH4 [2+]","2M+H [1+]","2M+ACN+H [1+]")
```

3.2. PEA for the 4-PFAS Mixture

Using the results from the 4-PFAS Mixture MWAS using quantile-based q-computation (Section 2.1), we conducted a *mummichog* PEA to identify enriched pathways using MetaboAnalystR version 3.2. The specifics of this analysis are described in detail in our

paper (**paper under review**). More information on *mummichog* can be found here (Li et al.).

In brief, the analysis was run in mixed mode (for C18 and HILIC simultaneously) and restricted to adducts that could potentially form based on our mobile phases and internal standards (the specific aduts are detailed below). This PEA was conducted using the human MFN network, a mass tolerance of 5ppm, 10,000 permutations and a p-value cutoff <0.05 to delineate between significantly enriched and non-significantly enriched features. The analysis was restricted to metabolite data sets containing at least 3 entries. A p(gamma) < 0.05 was considered statistically significant.

Adducts Restricted to in Our Anaysis

a. Negative Ion Mode M+FA-H [1-], M-H [1-], 2M-H [1-], M-H2O-H [1-], M-H+O [1-], M(C13)-H [1-], 2M+FA-H [1-], M-3H [3-], M-2H [2-], M+ACN-H [1-], M+HCOO [1-], and M+CH3COO [1-].

b. Positive Ion Mode M [1+], M+H [1+], M+2H [2+], M+3H [3+], M+H2O+H [1+], M-H2O+H [1+], M(C13)+H [1+], M(C13)+2H [2+], M(C13)+3H [3+], M-NH3+H [1+], M+ACN+H [1+], M+ACN+2H [2+], M+2ACN+2H [2+], M+3ACN+2H [2+], M+NH4 [1+], M+H+NH4 [2+], 2M+H [1+], and 2M+ACN+H [1+].

You will need to specify your own file pathways to export the results. These files were created in and exported in Section 2.1.

```
# Creating an object for storing data for mummichog
mSet4 <- InitDataObjects("mass_all", "mummichog", FALSE)

# Ranking the peaks by their p-value
mSet4 <- SetPeakFormat(mSet4, "rmp")

# Specifying
    #a. a mass tolerance of 5.0
    #b. mixed mode
    #c.not enforcing primary ions as this is exploratory
mSet4 <- UpdateInstrumentParameters(mSet4, 5.0, "mixed", "no");

# Reading in the specific peak list
mSet4 <- Read.PeakListData(mSet4, "Final
Datasets/Metaboanalyst/QG_comp_full_METABOLOMICS_4PFAS.csv"); # INSERT YOUR
PATHWAY HERE

# Performing a sanity check to ensure the data is in a suitable format for
further analysis
mSet4 <- SanityCheckMummichogData(mSet4)
```

```
# Adding in the adduct data
mSet4 <- Setup.AdductData(mSet4, add.vec);

# Specifying both positive and negative adducts
mSet4 <- PerformAdductMapping(mSet4, "mixed")

# Running the mummichog algorithm using selected adducts and version 2 of the
mummichog algorithm
mSet4 <- SetPeakEnrichMethod(mSet4, "mum", "v2")
mSet4 <- SetMummichogPval(mSet4, .05) #Specifying a p-value of 0.05

# Selecting the human MFN network, selecting the current human MFN library,
restricting to metabolite datasets with at least 3 entries, and running
10,000 permutations.
mSet4 <- PerformPSEA(mSet4, "hsa_mfn", "current", 3 , 10000)

# Storing the results as a dataframe
mummi_results_4pfas <- as.data.frame(mSet4$mummi.resmat)

# Restricting the results to those with a p(gamma) <0.05
mummi_results_4pfas <- mummi_results_4pfas[mummi_results_4pfas$Gamma< 0.05, ]

# Storing the results as a .csv
write.csv(mummi_results_4pfas, "/Users/amber/Desktop/mummi_4pfas_res.csv")

# Removing unneeded datasets and values
rm(all.mzsn, mdata.all, mdata.siggenes, metaboanalyst_env, mSet4,
mummi_results_4pfas, anal.type, api.base, meta.selected, module.count,
msg.vec, primary.user, smpdbpw.count, url.pre)
```

3.3. PEA for Individual PFAS

Using the results from the single pollutant MWAS models using linear regression (Section 2.1), we conducted a *mummichog* PEA to identify enriched pathways using MetaboAnalystR version 3.2. The specifics of this analysis are described in detail in our paper (**paper under review**). More information on *mummichog* can be found here (Li et al.).

Identical to the 4-PFAS analysis (detailed in 3.2), the analysis was run in mixed mode (for C18 and HILIC simultaneously) and restricted to adducts that could potentially form based on our mobile phases and internal standards (the specific aduts are detailed below). This PEA was conducted using the human MFN network, a mass tolerance of 5ppm, 10,000 permutations and a p-value cutoff <0.05 to delineate between significantly enriched and non-significantly enriched features. The analysis was restricted to metabolite data sets containing at least 3 entries. A p(gamma) < 0.05 was considered statistically significant.

Adducts Restricted to in Our Anaysis

a. Negative Ion Mode M+FA-H [1-], M-H [1-], 2M-H [1-], M-H2O-H [1-], M-H+O [1-], M(C13)-H [1-], 2M+FA-H [1-], M-3H [3-], M-2H [2-], M+ACN-H [1-], M+HCOO [1-], and M+CH3COO [1-].

b. Positive Ion Mode M [1+], M+H [1+], M+2H [2+], M+3H [3+], M+H2O+H [1+], M-H2O+H [1+], M(C13)+H [1+], M(C13)+2H [2+], M(C13)+3H [3+], M-NH3+H [1+], M+ACN+H [1+], M+ACN+2H [2+], M+2ACN+2H [2+], M+3ACN+2H [2+], M+NH4 [1+], M+H+NH4 [2+], 2M+H [1+], and 2M+ACN+H [1+].

You will need to specify your own file pathways to export the results. These files were created in and exported in Section 2.1.

```
# Creating a list of the PFAS we are evaluating
PFAS <- c("PFOA", "PFOS", "PFHxS", "PFNA", "MEFOSAA")

# Creating an empty list to store the results for each PFAS
results_list <- list()

# Creating a loop to loop through each PFAS and output the results
for (PFAS in PFAS) {
  # Creating an object for storing data for mummichog
  mSet <- InitDataObjects("mass_all", "mummichog", FALSE)
  # Ranking the peaks by their p-value
  mSet <- SetPeakFormat(mSet, "rmp")
  # Specifying instrument parameters
  mSet <- UpdateInstrumentParameters(mSet, 5.0, "mixed", "no")
  # Reading in the specific peak list
  file_path <- paste("Final Datasets/Metaboanalyst/Individual PFAS/", PFAS,
"_METABOANALYST.csv", sep = "")
  mSet <- Read.PeakListData(mSet, file_path)
  # Performing a sanity check
  mSet <- SanityCheckMummichogData(mSet)
  # Adding in the adduct data
  mSet <- Setup.AdductData(mSet, add.vec)
  # Specifying both positive and negative adducts
  mSet <- PerformAdductMapping(mSet, "mixed")
  # Running the mummichog algorithm using selected adducts and version 2 of
the mummichog algorithm
  mSet <- SetPeakEnrichMethod(mSet, "mum", "v2")
  mSet <- SetMummichogPval(mSet, 0.05) # Specifying a p-value of 0.05
  # Selecting the human MFN network, selecting the current human MFN library,
restricting to metabolite datasets with at least 3 entries, and running
10,000 permutations
  mSet <- PerformPSEA(mSet, "hsa_mfn", "current", 3, 10000)
  # Storing the results as a dataframe
  results_list[[PFAS]] <- as.data.frame(mSet$mummi.resmat)
  # Restricting the results to those with a p(gamma) < 0.05
  results_list[[PFAS]] <- results_list[[PFAS]][results_list[[PFAS]]$Gamma <
```

```
0.05, ]
  # Storing the results as a .csv
  write.csv(results_list[[PFAS]], paste("mummi_", PFAS, "_res.csv", sep =
""))
, eval=FALSE}
# Removing unneeded datasets and values
rm(all.mzsn, mdata.all, mdata.siggenes, metaboanalyst_env, mSet, anal.type,
api.base, file_path, meta.selected, module.count, msg.vec, PFAS,
primary.user, smpdbpw.count, url.pre, mummi_results_4pfas)
```

3.4. Sensitivity Analysis: PEA for the 5-PFAS Mixture

Using the results from the 5-PFAS Mixture MWAS using quantile-based q-computation (Section 2.3), we conducted a *mummichog* PEA to identify enriched pathways using MetaboAnalystR version 3.2. The specifics of this analysis are described in detail in our paper (**paper under review**). More information on *mummichog* can be found here (Li et al.).

In brief, the analysis was run in mixed mode (for C18 and HILIC simultaneously) and restricted to adducts that could potentially form based on our mobile phases and internal standards (the specific aduts are detailed below). This PEA was conducted using the human MFN network, a mass tolerance of 5ppm, 10,000 permutations and a p-value cutoff <0.05 to delineate between significantly enriched and non-significantly enriched features. The analysis was restricted to metabolite data sets containing at least 3 entries. A p(gamma) < 0.05 was considered statistically significant.

Adducts Restricted to in Our Anaysis

a. Negative Ion Mode M+FA-H [1-], M-H [1-], 2M-H [1-], M-H2O-H [1-], M-H+O [1-], M(C13)-H [1-], 2M+FA-H [1-], M-3H [3-], M-2H [2-], M+ACN-H [1-], M+HCOO [1-], and M+CH3COO [1-].

b. Positive Ion Mode M [1+], M+H [1+], M+2H [2+], M+3H [3+], M+H2O+H [1+], M-H2O+H [1+], M(C13)+H [1+], M(C13)+2H [2+], M(C13)+3H [3+], M-NH3+H [1+], M+ACN+H [1+], M+ACN+2H [2+], M+2ACN+2H [2+], M+3ACN+2H [2+], M+NH4 [1+], M+H+NH4 [2+], 2M+H [1+], and 2M+ACN+H [1+].

You will need to specify your own file pathways to export the results. These files were created in and exported in Section 2.1.

```
# Creating an object for storing data for mummichog
mSet5 <- InitDataObjects("mass_all", "mummichog", FALSE)

# Ranking the peaks by thei p-value
mSet5<- SetPeakFormat(mSet5, "rmp")
```

```r
# Specifying
    #a. a mass tolerance of 5.0
    #b. mixed mode
    #c. not enforcing primary ions.
mSet5<-UpdateInstrumentParameters(mSet5, 5.0, "mixed", "no");

# Reading in the specific peak list
mSet5<-Read.PeakListData(mSet5, "Final
Datasets/Metaboanalyst/QG_comp_full_METABOLOMICS_4PFAS_and_ME_PFOSA.csv"); #
INSERT YOUR PATHWAY HERE

# Performing a sanity check to ensure the data is in a suitable format for
further analysis
mSet5<-SanityCheckMummichogData(mSet5)

# Adding in the adduct data
mSet5<-Setup.AdductData(mSet5, add.vec);

# Specifying both positive and negative adducts
mSet5<-PerformAdductMapping(mSet5, "mixed")

# Running the mummichog algorithm using selected adducts and version 2 of the
mummichog algorithm
mSet5<-SetPeakEnrichMethod(mSet5, "mum", "v2")
mSet5<-SetMummichogPval(mSet5, .05) #Specifying a p-value of 0.05

# Selecting the human MFN network, selecting the current human MFN library,
restricting to metabolite datasets with at least 3 entries, and running
10,000 permutations.
mSet5<-PerformPSEA(mSet5, "hsa_mfn", "current", 3 , 10000)

# Storing the results as a dataframe
mummi_results_5pfas<- as.data.frame(mSet5$mummi.resmat)

# Restricting the results to those with a p(gamma) <0.05
mummi_results_5pfas <- mummi_results_5pfas[mummi_results_5pfas$Gamma< 0.05, ]

# Storing the results as a .csv
write.csv(mummi_results_4pfas, "/Users/amber/Desktop/mummi_4pfas_res.csv")

#Removing datasets and values that are not needed
rm(all.mzsn, mdata.all, mdata.siggenes, metaboanalyst_env, mSet5, anal.type,
api.base, meta.selected, module.count, msg.vec, primary.user, smpdbpw.count,
url.pre, results_list)
```