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 [perfluorooctanesulfonic acid (PFOS), perfluorooctanoic acid (PFOA), perfluorohexanesulfonic acid (PFHxS), and perfluorononanoic acid (PFNA)] measured at 16 weeks' gestation and cord serum metabolites. The metabolites were identified using an untargeted analysis approach. Additionally, we examined each PFAS individually and included *N*-methylperfluorooctane sulfonamido acetic acid (MeFOSAA) in the mixture analysis to gain insight into their potential influence on the mixture in secondary and sensitivity analyses, respectively.

The code provided here is the corresponding code for our analyses. Functions performed using this code can be seen in the box below.

Functions Performed Using this Code

1. Data Import and Cleaning

Cleans and structures data for analyses.

2. Metabolome Wide Association Study (MWAS)

Section 2.1 Conducts an MWAS using quantile-based g-computation (Keil et al.). This evaluates associations between a 4-PFAS mixture (PFOS, PFOA, PFNA, PFHxS) with each cord serum metabolite.

Section 2.2 Conducts an MWAS using linear regression to identify individual associations between each PFAS and each metabolite. The purpose of this was to observe the potential impact of individual PFAS on the mixture.

Section 2.3 Conducts an MWAS using quantile-based g-computation to evaluate associations between the 5-PFAS mixture (4 PFAS mixture plus MeFOSAA) and each metabolite. This was a sensitivity analysis to determine the impact of MeFOSAA on our final results.

3. Pathway Enrichment Analysis (PEA)

Section 3.2 Performs a PEA using *mummichog* via Metaboanalyst 3.2 to identify the association between our 4-PFAS mixture (PFOS, PFOA, PFNA, PFHxS) and specific metabolic pathways (Li et al). The dataset used for this analysis was created during the MWAS of the 4-PFAS mixture (Section 2.1).

Section 3.3 Performs a PEA using *mummichog* via Metaboanalyst 3.2 to identify the association between each individual PFAS and specific metabolic pathways. This dataset was created during the single PFAS

MWAS using linear regression (Section 2.2). The purpose of this was to observe the potential impact of individual PFAS on the mixture.

Section 3.4 Performs a PEA using *mummichog* via Metaboanalyst 3.2 to identify the association between a 5-PFAS mixture and specific metabolic pathways. This dataset was created during the MWAS of the 5-PFAS mixture (Section 2.3). This was a sensitivity analysis to determine the impact of MeFOSAA on our final results.

The data for this study is from the Health Outcomes and Measures of the Environment (HOME) Study (Braun et al.). This data is not publically available. However, the HOME Study principal investigators actively engaged in collaborative data-sharing projects and welcome new collaborations. Interested investigators can request data access here (HOME Study Data Request). The HOME Study Data Sharing Committee meets regularly to review proposed research projects, ensure that they do not overlap with extant projects, and are an efficient use of scarce resources (e.g. cord blood).

1. Data Import and Data Cleaning

1.1. Needed Libraries

I first secured all needed packages if I did not have these installed already; this step is required before you can run this code. An example of code to install a package in R is as follows:

```
install.packages("tidyverse")
```

where "tidyverse" can be replaced with any of the packages in the code block below except for MetaboanalystR. MetaboanalystR installation instructions can be found here (MetaboAnalystR).
After all packages are appropriately installed, load all necessary packages using the code below.

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

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

#Removing libraries
rm(libraries)
```

1.2. Importing Needed Data

After loading all the needed packages, I then imported all of my data. I started with my metabolomics data. This is the focus of the code chunk below.

For this code chunk I have 2 sets of files:

The main C18 and HILIC files (named *C18* and *HILIC*)

The map C18 and HILIC files (named *C18_map* and *HILIC_map*)

The main files contain feature data (mass to charge ratios (*mz*), retention time, and ion abundances) for each feature for each participant. The map files contain the participant IDs and the metabolomic-specific IDs for each participant. 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 detailed in the box below.

Main File Headers

Dataframe Names: C18 **and** HILIC

column 1. *mz*: the mass to charge ratio.

column 2. *time*: retention time.

column 3. *mode*: labeled 1 for C18 and 2 for HILIC files.

column 4+. a column for each participant containing the the feature data. There are likely several hundred to several thousand features

that have been identified. The header name for each row is the metabolomic-specific ID (which differs from the participant.

ID). An example of a metabolomics specific ID is *C18_105*.

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

For the map files, the headers are detailed in the box below.

Map File Headers

Dataframe Names: C18_map **and** HILIC_map

column 1. *participant_ID*: The assigned participant ID for the data in this study.

column 2. *HILIC_ID* for HILIC data and *C18_ID* for C18 data. IDs that are specific only to the metabolomics data.

IMPORTANT

For our data, both HILIC and C18 datasets had already:

been batch corrected using WaveICA 2.0 (Deng et al.)

had values removed if CV >30% (within triplicate)

had values removed if non-detect intensities >20%

had missing values imputed using:
$$\frac{\text{the minimum value per feature}}{\sqrt{2}}$$

Next, I imported the feature data and map files using the code chunk below.

```
# 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 HILIC 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
```

I then imported the PFAS data. This dataset was named *pfas_only*.

The headers for *pfas_only* are in the box below.

PFAS Dataset Headers

Dataframe Name: pfas_only

column 1. *participant_ID*: the assigned participant ID for the data in this study. This was used to merge with the metabolomics data.

'participant ID later.

column 2. *pfoa_log2*: log-2 transformed serum PFOA concentrations (ng/mL).

column 3. *pfos_log2*: log-2 transformed serum PFOS concentrations (ng/mL).

column 4. *pfna_log2*: log-2 transformed serum PFNA concentrations (ng/mL).

column 5. *pfhxs_log2*: log-2 transformed serum PFHxS concentrations (ng/mL).

column 6. *me_pfosa_acoh_log2*: log-2 transformed serum MeFOSAA concentrations (ng/mL).

```
# 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
```

After importing the feature and PFAS data, I imported the covariate data. This dataset was named *cov*.

For *cov*, the headers are as follows:

Covariate Dataset Headers

Dataframe Name: cov

column 1. *participant_ID*: The assigned participant ID for the data in this study.

column 2. *momagedeliv*: maternal age at delivery (years).

column 3. *ct_16w*: log10 transformed serum cotinine concentrations measured at 16 weeks' gestation (ng/mL).

column 4. *mom_race*: maternal race (coded as "White, not-Hispanic" and "Other race").

column 5. *midincome*: family income in USD (continuous).

column 6. *parity*: parity (coded as "Nulliparous" and "Parous").

```
# Importing covariate data
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 was to create a master dataset named *pfas*. I did this by merging the following datasets into a single dataset:

1. *pfas_only*: PFAS data
2. *cov*: covariate data
3. *C18_map*: contains metabolomic-specific IDs and participant IDs for the C18 data
4. *HILIC_map*: contains metabolomic-specific IDs and participant IDs for the HILIC data

This is seen below.

```r
# 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 Next, I created a new dataframe named *feature_table*. *feature_table* contains only participants with both PFAS and covariate data (i.e. is a subset of the full dataframe; this is for a complete case analysis). This feature table also contains both C18 and HILIC data stacked on top of one another. Columns for the new feature table can be seen below.

Feature Table Headers

Dataframe Name: feature_table

column 1. *mz*: the mass to charge ratio.

column 2. *time*: retention time.

column 3. labeled 1 for C18 and 2 for HILIC files.

column 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 is equal to the number of participants + 3.

Below is the code I used to create this feature table.

```r
# 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 C
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
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 a
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)
```

```r
#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
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 was to create a list that 'splits' each individual feature into their own tibble. Each individual tibble contains the mass-to-charge ratio (*mz*) and retention time (*time*) for that metabolite as well as the pfas and covariate data for each participant. The purpose of doing this is so that we can run each individual metabolite independently through our MWAS.

```r
# Adding a new feature column named 'feature_id'. This columns contains numbers 1,2,3,...N. This column
feature_table <- cbind(feature_id = 1:nrow(feature_table), feature_table)

# Pivoting the dataset to a 'long file'. This contains 6 values (feature_ID, mz, time, mode, name [meta
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
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)

About Quantile-Based g-Computation

The purpose of quantile-based g-computation (QGComp) is to examine the joint effects of a mixture PFAS with relation to an outcome of interest. QGComp accomplishes this by calculating the parameters of a marginal structural model. For the QGComp models in our study, these parameters characterize the difference in feature intensity resulting from a simultaneous, one-quantile increase of all PFAS in the mixture. Resources on quantile-based g-computation are in the box below.

Resources for Quantile-Based g-Computation:

The original paper (Keil et al.).

Comprehensive R Archive Network (CRAN) (Cran website).

Github page (QGComp Github)

2.1. MWAS for the 4-PFAS Mixture (Quantile-Based g-Computation)

This chunk of code is the primary Metabolome Wide Association Study (MWAS) analysis for our study **(study under review)**. We evaluated whether the joint effect of our 4 PFAS of interest (PFOS, PFOA, PFHxS, PFNA) were associated with the ion abundance for each feature, adjusting for parity, family income, maternal age, maternal race, and cotinine concentrations.

The results of these models are outputted to 3 files, detailed in the box below.

Files Outputted from the 4-PFAS MWAS Code

File 1:* a full or overall file that contains all the results

File 2: a significant file that contains significant values (i.e. those with FDR values <0.20)

File 3:* a metaboanalyst file that contains only the variables needed to import into metaboanalyst (*mz*, *time*, *p-value*, *mode*).
*Files 1 and 3 contain 9 columns. These columns are:

m.z: the mass to charge ratio

r.t: retention time

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

SE: standard error

ci_lower: lower 95% confidence interval

ci_upper: upper 95% confidence interval

p.value: unadjusted p-value (before FDR correction)

mode: negative (C18) or positive (HILIC)

FDR: false discovery rate (FDR) corrected p-value

For the code below, 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 code below. *value* in this code refers to the ion abundance of each feature.

```
# 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,
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', r

# 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. MWAS for Individual PFAS (Linear Regression)

The code chunk below evaluates whether each individual PFAS (PFOS, PFOA, PFHxS, PFNA, or MeFOSAA) is associated with the ion abundance for each feature, adjusting for parity, family income, maternal age, maternal race, and cotinine concentrations. The purpose of this was to observe the potential impact of the individual metabolites on the 4-PFAS mixture (detailed in Section 2.1).

The results of these models are outputted to 3 files, detailed in the box below.

Files Outputted from the Single Pollutant PFAS MWAS Code

File 1:* a full or overall file that contains all the results

File 2: a significant file that contains significant values (i.e. those with FDR values <0.20)

File 3:* a metaboanalyst file that contains only the variables needed to import into metaboanalyst (*mz*, *time*, *p-value*, *mode*).
*Files 1 and 3 contain 9 columns. These columns are:

m.z: the mass to charge ratio

r.t: retention time

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

SE: standard error

ci_lower: lower 95% confidence interval

ci_upper: upper 95% confidence interval

p.value: unadjusted p-value (before FDR correction)

mode: negative (C18) or positive (HILIC)

FDR: false discovery rate (FDR) corrected p-value

For the code below, 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. As a reminder, all PFAS have already been log2 transformed. *value* in this code refers to the ion abundance of each feature.

```
# Creating 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]])
    # Fitting 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]])
    # Filling 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)$coefficients[PFAS, "Pr(>|t|)"] #p-value
    res_mat[i, 8] <- as.numeric(list[[i]][1, 4]) #mode (i.e. direction)
  , eval=FALSE}
  # Converting matrix to data frame and add FDR column
  res_df <- as.data.frame(res_mat)
  colnames(res_df) <- colnames(res_mat)
  res_df$FDR <- p.adjust(res_df$p.value, method = 'BH')
  # Recoding mode column
  res_df$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',

  # 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', r
  # 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', r
# 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 = TRU
  write.csv(PFHxS_METABOANALYST, 'Final Datasets/Metaboanalyst/Individual PFAS/PFHxS_METABOANALYST.csv'

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

# 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*: MWAS for the 5-PFAS Mixture

This is a sensitivity analysis to evaluate the impact of MeFOSAA on the original 4-PFAS mixture. In this code chunk, we evaluated whether the joint effect of a 5-PFAS mixture (4-PFAS mixture *detailed in section 2.1* plus MeFOSAA) were associated with the ion abundance for each feature, adjusting for parity, family income, maternal age, maternal race, and cotinine concentrations. This was accomplished using a quantile-based g-computation model. The purpose of this sensitivity analysis was to determine the impact of MeFOSAA on our final results.

The results of these models are outputted to 3 files:

Files Outputted from the 5-PFAS MWAS Code

File 1:* a full or overall file that contains all the results

File 2: a significant file that contains significant values (i.e. those with FDR values <0.20)

File 3:* a metaboanalyst file that contains only the variables needed to import into metaboanalyst (*mz*, *time*, *p-value*, *mode*).
*Files 1 and 3 contain 9 columns. These columns are:

m.z: the mass to charge ratio

r.t: retention time

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

SE: standard error

ci_lower: lower 95% confidence interval

ci_upper: upper 95% confidence interval

p.value: unadjusted p-value (before FDR correction)

mode: negative (C18) or positive (HILIC)

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 code below. *value* in the code below refers to the ion abundance of each feature.

```
# 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,
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$p.value, method = 'BH')
  # Recoding mode to 'negative' and 'positive'
  res_df$mode[res_df$mode == 1] <- "negative"
```

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

# Creating files
qg_comp_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.name
write.csv(qg_comp_5_METABOANALYST, 'Final Datasets/Metaboanalyst/QG_comp_full_METABOLOMICS_4PFAS_and_ME

# 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. MetaboAnalystR performs a pathway enrichment analysis (PEA) using *mummichog* to identify associations between an exposure of interest, such as PFAS, and metabolic pathways. More information on *mummichog* can be found here (Li et al.). Information on Metaboanalyst can be found at (MetaboAnalyst). Additionally, information on MetaboAnalystR can be found at (MetaboanalystR).

3.1. Specifying Specific Adducts

For our PEA, we first specify the adducts permitted for our 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. As such, this may differ from study to study.

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

```
# Creating a vector that contains the customized adducts for our study.
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 (Section 2.1), we conducted a *mummichog* PEA to identify enriched pathways using MetaboAnalystR.

Details regarding our untargeted analysis are described in detail in our paper (**paper under review**). From this information, we restricted to adducts that could potentially form based on our mobile phases

and internal standards (the specific adducts used in our analyses are detailed in the box 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. Furthermore, this analysis was restricted to metabolite data sets containing at least 3 entries. A p(gamma) < 0.05 was considered statistically significant.

Adducts We Restricted to in Our Analyses

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-]

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. The files imported for this analysis were created and exported out 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");

# 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 d
mSet4 <- PerformPSEA(mSet4, "hsa_mfn", "current", 3 , 10000)

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

```r
# 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.ba
```

3.3. PEA for Individual PFAS

Using the results from the single pollutant MWAS models (Section 2.2), we conducted a *mummichog* PEA to identify enriched pathways using MetaboAnalystR. This was part of a secondary analysis that evaluated the impact of individual PFAS on the PEA of the 4-PFAS mixture (this is discussed in Section 3.2).

Details regarding our untargeted analysis are described in detail in our paper (**paper under review**). From this, we restricted to adducts that could potentially form based on our mobile phases and internal standards (the specific adducts used in our analyses are detailed in the box 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. Furthermore, this analysis was restricted to metabolite data sets containing at least 3 entries. A p(gamma) < 0.05 was considered statistically significant.

Adducts We Restricted to in Our Analyses

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-]

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. The files imported for this analysis were created and exported out in Section 2.2.

```r
# 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
  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.s
```

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

Using the results from the 5-PFAS Mixture MWAS (Section 2.3), we conducted a *mummichog* PEA to identify enriched pathways using MetaboAnalystR. This was a sensitivity analysis to determine the impact of MeFOSAA on the PEA of our 4-PFAS mixture (this is discussed in Section 3.2).

Details regarding our untargeted analysis are described in detail in our paper (**paper under review**). From this information, we restricted to adducts that could potentially form based on our mobile phases and internal standards (the specific adducts used in our analyses are detailed in the box 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. Furthermore, this analysis was restricted to metabolite data sets containing at least 3 entries. A p(gamma) < 0.05 was considered statistically significant.

Adducts We Restricted to in Our Analyses

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-]

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. The files imported for this analysis were created and exported out in Section 2.3.

```
# 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")
```

```
# 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_PFU

# 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 d
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, m
```