

ANM_Lipid_Regression

Asger Wretlind

2023-11-09

```
#Load libraries  
library(tidyverse)
```

```
## Warning: pakke 'ggplot2' blev bygget under R version 4.3.1
```

```
## Warning: pakke 'purrr' blev bygget under R version 4.3.1
```

```
## Warning: pakke 'dplyr' blev bygget under R version 4.3.1
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
## v dplyr      1.1.3      v readr      2.1.4
```

```
## v forcats   1.0.0      v stringr   1.5.0
```

```
## v ggplot2    3.4.3      v tibble    3.2.1
```

```
## v lubridate  1.9.2      v tidyr     1.3.0
```

```
## v purrr      1.0.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()     masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(here)
```

```
## here() starts at H:/Desktop/ANM_PRS/ANM_Data_Analysis
```

```
library(vroom)
```

```
## Warning: pakke 'vroom' blev bygget under R version 4.3.1
```

```
##
```

```
## Vedhæfter pakke: 'vroom'
```

```
##
```

```
## De følgende objekter er maskerede fra 'package:readr':
```

```
##
```

```
## as.col_spec, col_character, col_date, col_datetime, col_double,
```

```
## col_factor, col_guess, col_integer, col_logical, col_number,
```

```
## col_skip, col_time, cols, cols_condense, cols_only, date_names,
```

```
## date_names_lang, date_names_langs, default_locale, fwf_cols,
```

```
## fwf_empty, fwf_positions, fwf_widths, locale, output_column,
```

```
## problems, spec
```

```

#Load data
data <- vroom(here("data/ANM_Lipid_Preprocessed_v4.csv"))

## Rows: 841 Columns: 293
## -- Column specification -----
## Delimiter: "\t"
## chr (8): ID, Site, Date, Status, Sex, DOB, Accommodation, Marital_Status
## dbl (285): Visit, Order, Label, Age, Fulltime_Education_Years, apoe, e4_p, e...
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

```

```

#Load list of selected lipids
selected_lipids <- vroom(here("data/ANM_module_selected_lipids_res_v1.1.csv"))

```

```

## Rows: 47 Columns: 3
## -- Column specification -----
## Delimiter: "\t"
## chr (3): Module, Lipid, Color
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.

```

```

#Function to clear special characters that may create confusion
Clean_names <- function(lipid_names){
  tmp_lipid_names <- gsub("\\(", "", lipid_names)
  tmp_lipid_names <- gsub("\\:", "", tmp_lipid_names)
  tmp_lipid_names <- gsub("\\)", "", tmp_lipid_names)
  tmp_lipid_names <- gsub("\\-", "", tmp_lipid_names)
  tmp_lipid_names <- gsub("\\/", "", tmp_lipid_names)

  return(tmp_lipid_names)
}

```

```

#Clean lipid names of all lipids
tmp_lipid_names <- colnames(data[which(colnames(data) == "Cer(d42:0)":length(data))])
tmp_lipid_names_clean <- Clean_names(tmp_lipid_names)

```

```

#Create a new data subset and clean the lipid names
data_regression <- data
colnames(data_regression) <- Clean_names(colnames(data_regression))

```

```

#Subset only AD and control samples
data_regression_AD <- data_regression %>%
  mutate(AD_CTL = if_else(Status == "ADC", 1, NA)) %>%
  mutate(AD_CTL = if_else(Status == "CTL", 0, AD_CTL)) %>%
  relocate(AD_CTL, .after = Status) %>%
  filter(!is.na(AD_CTL)) %>%
  data.frame()

```

```

#Subset only MCI and control samples
data_regression_MCI <- data_regression %>%

```

```

mutate(MCI_CTL = if_else(Status == "MCI", 1, NA)) %>%
mutate(MCI_CTL = if_else(Status == "CTL", 0, MCI_CTL)) %>%
relocate(MCI_CTL, .after = Status) %>%
filter(!is.na(MCI_CTL)) %>%
data.frame()

#Subset for no missing MMSE
data_regression_MMSE <- data_regression %>%
  filter(!is.na(MMSE_Total)) %>%
  rename(MMSE = MMSE_Total) %>%
  data.frame()

#Subset for no missing APOE4
data_regression_APOE4 <- data_regression %>%
  filter(!is.na(e4_c)) %>%
  data.frame()

#Subset by Sex
data_regression_sex <- data_regression %>%
  mutate(Sex_binary = if_else(Sex == "Female", 1, NA)) %>%
  mutate(Sex_binary = if_else(Sex == "Male", 0, Sex_binary)) %>%
  relocate(Sex_binary, .after = Status) %>%
  filter(!is.na(Sex_binary)) %>%
  data.frame()

#Subset only AD and control samples in Women
data_regression_AD_women <- data_regression %>%
  filter(!is.na(Sex)) %>%
  filter(Sex == "Female") %>%
  mutate(AD_CTL = if_else(Status == "ADC", 1, NA)) %>%
  mutate(AD_CTL = if_else(Status == "CTL", 0, AD_CTL)) %>%
  relocate(AD_CTL, .after = Status) %>%
  filter(!is.na(AD_CTL)) %>%
  data.frame()

#Subset only AD and control samples in Men
data_regression_AD_men <- data_regression %>%
  filter(!is.na(Sex)) %>%
  filter(Sex == "Male") %>%
  mutate(AD_CTL = if_else(Status == "ADC", 1, NA)) %>%
  mutate(AD_CTL = if_else(Status == "CTL", 0, AD_CTL)) %>%
  relocate(AD_CTL, .after = Status) %>%
  filter(!is.na(AD_CTL)) %>%
  data.frame()

#Subset only MCI and control samples in Women
data_regression_MCI_women <- data_regression %>%
  filter(!is.na(Sex)) %>%
  filter(Sex == "Female") %>%
  mutate(MCI_CTL = if_else(Status == "MCI", 1, NA)) %>%
  mutate(MCI_CTL = if_else(Status == "CTL", 0, MCI_CTL)) %>%
  relocate(MCI_CTL, .after = Status) %>%
  filter(!is.na(MCI_CTL)) %>%

```

```

data.frame()

#Subset only MCI and control samples in Men
data_regression_MCI_men <- data_regression %>%
  filter(!is.na(Sex)) %>%
  filter(Sex == "Male") %>%
  mutate(MCI_CTL = if_else(Status == "MCI", 1, NA)) %>%
  mutate(MCI_CTL = if_else(Status == "CTL", 0, MCI_CTL)) %>%
  relocate(MCI_CTL, .after = Status) %>%
  filter(!is.na(MCI_CTL)) %>%
  data.frame()

#Subset for no missing MMSE in Women
data_regression_MMSE_women <- data_regression %>%
  filter(!is.na(Sex)) %>%
  filter(Sex == "Female") %>%
  filter(!is.na(MMSE_Total)) %>%
  rename(MMSE = MMSE_Total) %>%
  data.frame()

#Subset for no missing MMSE in Men
data_regression_MMSE_men <- data_regression %>%
  filter(!is.na(Sex)) %>%
  filter(Sex == "Male") %>%
  filter(!is.na(MMSE_Total)) %>%
  rename(MMSE = MMSE_Total) %>%
  data.frame()

#Subset for no missing APOE4
data_regression_APOE4_women <- data_regression %>%
  filter(!is.na(Sex)) %>%
  filter(Sex == "Female") %>%
  filter(!is.na(e4_c)) %>%
  data.frame()

#Subset for no missing APOE4
data_regression_APOE4_men <- data_regression %>%
  filter(!is.na(Sex)) %>%
  filter(Sex == "Male") %>%
  filter(!is.na(e4_c)) %>%
  data.frame()

#Create a list containing all info needed for each model
regression_models <- list(
  Model_1 = c(name = "ADraw",
    formula_prefix = "AD_CTL ~ ",
    formula_sufix = "",
    data = "data_regression_AD",
    family = "binomial"),
  Model_2 = c(name = "ADadj",
    formula_prefix = "AD_CTL ~ ",
    formula_sufix = " + Sex + e4_c + Site + Age",
    data = "data_regression_AD",

```

```

        family = "binomial"),
Model_3 = c(name = "MCIraw",
            formula_prefix = "MCI_CTL ~ ",
            formula_sufix = "",
            data = "data_regression_MCI",
            family = "binomial"),
Model_4 = c(name = "MCIadj",
            formula_prefix = "MCI_CTL ~ ",
            formula_sufix = " + Sex + e4_c + Site + Age",
            data = "data_regression_MCI",
            family = "binomial"),
Model_5 = c(name = "MMSEraw",
            formula_prefix = "MMSE ~ ",
            formula_sufix = "",
            data = "data_regression_MMSE",
            family = "gaussian"),
Model_6 = c(name = "MMSEadj",
            formula_prefix = "MMSE ~ ",
            formula_sufix = " + Status + e4_c + Sex + Site + Age",
            data = "data_regression_MMSE",
            family = "gaussian"),
Model_7 = c(name = "APOEraw",
            formula_prefix = "e4_c ~ ",
            formula_sufix = "",
            data = "data_regression_APOE4",
            family = "gaussian"),
Model_8 = c(name = "APOEadj",
            formula_prefix = "e4_c ~ ",
            formula_sufix = " + Status + Sex + Site + Age",
            data = "data_regression_APOE4",
            family = "gaussian"),
Model_9 = c(name = "Sexraw",
            formula_prefix = "Sex_binary ~ ",
            formula_sufix = "",
            data = "data_regression_sex",
            family = "binomial"),
Model_10 = c(name = "Sexadj",
            formula_prefix = "Sex_binary ~ ",
            formula_sufix = " + Status + e4_c + Site + Age",
            data = "data_regression_sex",
            family = "binomial"),

Model_11 = c(name = "ADFemaleraw",
            formula_prefix = "AD_CTL ~ ",
            formula_sufix = "",
            data = "data_regression_AD_women",
            family = "binomial"),
Model_12 = c(name = "ADFemaleadj",
            formula_prefix = "AD_CTL ~ ",
            formula_sufix = " + e4_c + Site + Age",
            data = "data_regression_AD_women",
            family = "binomial"),
Model_13 = c(name = "ADMenraw",

```

```

        formula_prefix = "AD_CTL ~ ",
        formula_sufix = "",
        data = "data_regression_AD_men",
        family = "binomial"),
Model_14 = c(name = "ADMenadj",
        formula_prefix = "AD_CTL ~ ",
        formula_sufix = " + e4_c + Site + Age" ,
        data = "data_regression_AD_men",
        family = "binomial"),

Model_15 = c(name = "MCIFemaleraw",
        formula_prefix = "MCI_CTL ~ ",
        formula_sufix = "",
        data = "data_regression_MCI_women",
        family = "binomial"),
Model_16 = c(name = "MCIFemaleadj",
        formula_prefix = "MCI_CTL ~ ",
        formula_sufix = " + e4_c + Site + Age",
        data = "data_regression_MCI_women",
        family = "binomial"),
Model_17 = c(name = "MCIMenraw",
        formula_prefix = "MCI_CTL ~ ",
        formula_sufix = "",
        data = "data_regression_MCI_men",
        family = "binomial"),
Model_18 = c(name = "MCIMenadj",
        formula_prefix = "MCI_CTL ~ ",
        formula_sufix = " + e4_c + Site + Age" ,
        data = "data_regression_MCI_men",
        family = "binomial"),

Model_19 = c(name = "MMSEFemaleraw",
        formula_prefix = "MMSE ~ ",
        formula_sufix = "",
        data = "data_regression_MMSE_women",
        family = "gaussian"),
Model_20 = c(name = "MMSEFemaleadj",
        formula_prefix = "MMSE ~ ",
        formula_sufix = " + Status + e4_c + Site + Age",
        data = "data_regression_MMSE_women",
        family = "gaussian"),
Model_21 = c(name = "MMSEMenraw",
        formula_prefix = "MMSE ~ ",
        formula_sufix = "",
        data = "data_regression_MMSE_men",
        family = "gaussian"),
Model_22 = c(name = "MMSEMenadj",
        formula_prefix = "MMSE ~ ",
        formula_sufix = " + Status + e4_c + Site + Age" ,
        data = "data_regression_MMSE_men",
        family = "gaussian"),

Model_23 = c(name = "APOEFemaleraw",

```

```

        formula_prefix = "e4_c ~ ",
        formula_sufix = "",
        data = "data_regression_APOE4_women",
        family = "gaussian"),
Model_24 = c(name = "APOEFemaleleadj",
             formula_prefix = "e4_c ~ ",
             formula_sufix = " + Status + Site + Age",
             data = "data_regression_APOE4_women",
             family = "gaussian"),
Model_25 = c(name = "APOEMenraw",
             formula_prefix = "e4_c ~ ",
             formula_sufix = "",
             data = "data_regression_APOE4_men",
             family = "gaussian"),
Model_26 = c(name = "APOEMenadj",
             formula_prefix = "e4_c ~ ",
             formula_sufix = " + Status + Site + Age" ,
             data = "data_regression_APOE4_men",
             family = "gaussian"),

Model_27 = c(name = "ADFemaleNotadjAPOE",
             formula_prefix = "AD_CTL ~ ",
             formula_sufix = " + Site + Age",
             data = "data_regression_AD_women",
             family = "binomial"),
Model_28 = c(name = "ADMaleNotadjAPOE",
             formula_prefix = "AD_CTL ~ ",
             formula_sufix = " + Site + Age" ,
             data = "data_regression_AD_men",
             family = "binomial"),
Model_29 = c(name = "ADFemaleNotadjAge",
             formula_prefix = "AD_CTL ~ ",
             formula_sufix = "+ e4_c + Site",
             data = "data_regression_AD_women",
             family = "binomial"),
Model_30 = c(name = "ADMaleNotadjAge",
             formula_prefix = "AD_CTL ~ ",
             formula_sufix = "+ e4_c + Site" ,
             data = "data_regression_AD_men",
             family = "binomial"),
Model_31 = c(name = "ADFemaleOnlyadjSite",
             formula_prefix = "AD_CTL ~ ",
             formula_sufix = "+ Site",
             data = "data_regression_AD_women",
             family = "binomial"),
Model_32 = c(name = "ADMaleOnlyadjSite",
             formula_prefix = "AD_CTL ~ ",
             formula_sufix = "+ Site" ,
             data = "data_regression_AD_men",
             family = "binomial"))

```

```

#Data frame to populate
regression_summary <- data.frame("tmp" = c())

```

```

#Loop over each regression model
for (j in names(regression_models)){

  #Loop over each lipid
  for (i in tmp_lipid_names_clean){

    #Formula
    tmp_formula <- as.formula( paste0(regression_models[[j]][["formula_prefix"]], i,
    regression_models[[j]][["formula_sufix"]]))

    #Data
    tmp_data <- eval(as.symbol(regression_models[[j]][["data"]]))

    #Model type
    tmp_family <- regression_models[[j]][["family"]]

    #Fit logistic regression model
    model_fit <- glm(formula = tmp_formula,
                     data = tmp_data,
                     family = tmp_family)

    #Model suffix
    tmp_model_suffix <- regression_models[[j]][["name"]]

    #Extract summary statistics
    regression_summary[i, paste0("Estimate_", tmp_model_suffix)] <- summary(model_fit)$coefficients[i, "Est"
    regression_summary[i, paste0("StdError_", tmp_model_suffix)] <- summary(model_fit)$coefficients[i, "Std"
    regression_summary[i, paste0("Pval_", tmp_model_suffix)] <- summary(model_fit)$coefficients[i, which(gr
    regression_summary[i, paste0("Model_", tmp_model_suffix)] <- format(tmp_formula)

  }

  #Adjust for multiple testing
  regression_summary[, paste0("PvalFDR_", tmp_model_suffix)] <- p.adjust(regression_summary[, paste0("Pval"
    method = "fdr")

}

# #Example
# tmp <- glm(formula = MMSE ~ LacCerd321 + Status + e4_c + Site + Age + Total_TG,
#           data = data_regression_MMSE_women,
#           family = "gaussian")
#
# summary(tmp)

#clean
rm(i, j, tmp_formula, tmp_model_suffix, model_fit, tmp_data, data_regression, data_regression_MCI, data,
  data_regression_AD_women, data_regression_AD_men, tmp_family,
  data_regression_APOE4_women, data_regression_APOE4_men,
  data_regression_MCI_women, data_regression_MCI_men,

```



```
data_regression_MMSE_women, data_regression_MMSE_men)
```

```
#Remove lipids not selected by the WGCNA
regression_summary_selected <- regression_summary %>%
  mutate(Lipid = tmp_lipid_names) %>%
  relocate(Lipid, .before = 1 ) %>%
  filter(Lipid %in% selected_lipids$Lipid)

#Loop over each regression model
for (j in names(regression_models)){

  #Model suffix
  tmp_model_suffix <- regression_models[[j]][["name"]]

  #Adjust for multiple testing
  regression_summary_selected[, paste0("PvalFDR_", tmp_model_suffix)] <-
  p.adjust(regression_summary_selected[, paste0("Pval_", tmp_model_suffix)], method = "fdr")

}

rm(j, tmp_model_suffix)

library(gh4x)
```

```
## Warning: pakke 'gh4x' blev bygget under R version 4.3.1
```

```
#change the rownames into an actual column
regression_summary_selected$Lipid_clean <- rownames(regression_summary_selected)

#Empty tibble to populate
regression_results <- c()
tmp_order <- c()

#Stack summary results for heatmap
for (j in names(regression_models)){
  regression_results <- regression_results %>%
    rbind(.,
    tibble(regression_summary_selected) %>%
    select(contains(regression_models[[j]][["name"]]), Lipid) %>%
    rename_with(~gsub(paste0("_", regression_models[[j]][["name"]]), "", .)) %>%
    mutate(Outcome = regression_models[[j]][["name"]]))
  tmp_order <- append(tmp_order, regression_models[[j]][["name"]])
}

tmp_order <- gsub("adj", "", tmp_order)
tmp_order <- gsub("Men", "\nMale", tmp_order)
tmp_order <- gsub("Female", "\nFemale", tmp_order)

#Set facet strip color to the module colors
tmp_strip <- strip_themed(background_y = elem_list_rect(fill = unique(selected_lipids$Color)))

#Data wrangling
```

```

regression_results <- regression_results %>%
  mutate(Pstar = if_else(PvalFDR < 0.05, "*", "")) %>%
  mutate(Pstar = if_else(PvalFDR < 0.01, "**", Pstar)) %>%
  mutate(Pstar = if_else(PvalFDR < 0.001, "**", Pstar)) %>%
  mutate(Lipid = factor(Lipid, levels = rev(tmp_lipid_names))) %>%
  mutate(Outcome = gsub("Men", "\nMale", Outcome)) %>%
  mutate(Outcome = gsub("Female", "\nFemale", Outcome)) %>%
  mutate(Outcome = gsub("raw", "", Outcome)) %>%
  left_join(., selected_lipids) %>%
  mutate(Module = factor(Module, levels = unique(selected_lipids$Module))) %>%
  filter(grepl("adj", Outcome)) %>%
  mutate(Outcome = gsub("adj", "", Outcome)) %>%
  mutate(Outcome = factor(Outcome, levels = tmp_order)) %>%
  relocate(Model, .after = Outcome)

```

```
## Joining with 'by = join_by(Lipid)'
```

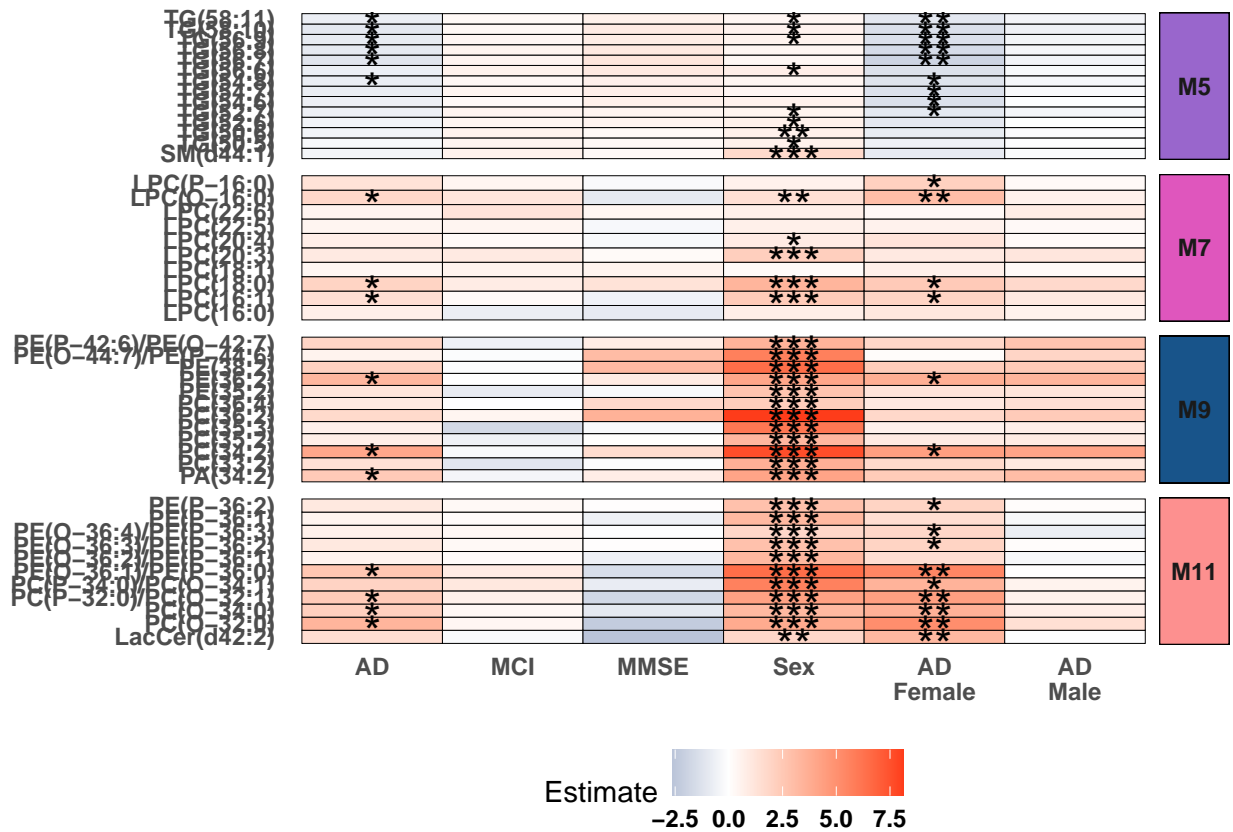
```

#plot heatmap
fig2_heatmap <- regression_results %>%
  filter(!grepl("MCI\n", Outcome), !grepl("MMSE\n", Outcome), !grepl("APOE", Outcome),
    !grepl("Age", Outcome), !grepl("Site", Outcome)) %>%
  mutate(Lipid = gsub("_A", "", Lipid)) %>%
  mutate(Lipid = gsub("_B", "", Lipid)) %>%
  ggplot(aes(x = Outcome, y = Lipid, fill = Estimate))+
  geom_tile(color = "black") +
  scale_fill_gradient2(low = "#18548A",
    mid = "#FFFFFF",
    high = "#FE3C1A")+
  geom_text(aes(label = Pstar,
    color = "black", size = 6, vjust = 0.75 )+
  facet_grid2(Module ~ ., scales = "free", strip = tmp_strip) +
  theme_minimal()+
  theme(axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.major = element_blank(),
    axis.text = element_text(face = "bold"),
    legend.text = element_text(face = "bold"),
    legend.position = "bottom",
    strip.text.y = element_text(angle = 0, face = "bold"))

# #Save figure
# pdf(here("figures/Heatmap_lipid_regression_v2.6.pdf"), width = 5, height = 7)

fig2_heatmap

```



```
# dev.off()

#Select significant lipids to AD
tmp_significant_lipids <- regression_results %>%
  filter(Outcome == "AD") %>%
  filter(PvalFDR < 0.05) %>%
  pull(Lipid)

#Create new variable in selected lipid to note significant AD association
selected_lipids$AD_associated <- "No"

selected_lipids$AD_associated[selected_lipids$Lipid %in% tmp_significant_lipids] <- "Yes"

rm(j, tmp_order, tmp_significant_lipids, tmp_strip, fig2_heatmap, regression_summary_selected)

#change the rownames into an actual column
regression_summary$Lipid_clean <- rownames(regression_summary)
regression_summary$Lipid <- tmp_lipid_names

#Empty tibble to populate
regression_results <- c()
tmp_order <- c()

#Stack summary results for heatmap
for (j in names(regression_models)){
  regression_results <- regression_results %>%
```

```

    rbind(.,
    tibble(regression_summary) %>%
    select(contains(regression_models[[j]][["name"]]), Lipid) %>%
    rename_with(~gsub(paste0("_", regression_models[[j]][["name"]]), "", .)) %>%
    mutate(Outcome = regression_models[[j]][["name"]]))
  tmp_order <- append(tmp_order, regression_models[[j]][["name"]])
}

#Data wrangling
regression_results <- regression_results %>%
  mutate(Outcome = gsub("Men", " Male", Outcome)) %>%
  mutate(Outcome = gsub("Female", " Female", Outcome)) %>%
  mutate(Outcome = gsub("raw", "", Outcome)) %>%
  mutate(Outcome = gsub("adj", " Adjusted", Outcome)) %>%
  mutate(Outcome = gsub("ADMaleNot", "AD MaleNot", Outcome)) %>%
  mutate(Outcome = gsub("aleNot AdjustedAPOE", "ale not Adjusted for APOE", Outcome)) %>%
  mutate(Outcome = gsub("aleNot AdjustedAge", "ale not Adjusted for Age", Outcome)) %>%
  mutate(Outcome = gsub("ADMaleOnly", "AD MaleOnly", Outcome)) %>%
  mutate(Outcome = gsub("aleOnly AdjustedSite", "ale Adjusted only for Site", Outcome))

# #Export supplementary table
# vroom_write(regression_results, here("data/sup_table_lipid_regression_v1.6.csv"))

#Clean
rm(regression_models, regression_results, j, tmp_order, regression_summary)

#Create a list containing all info needed for each model
regression_models <- list(
  Model_1 = c(name = "ADraw",
    formula_prefix = "AD_CTL ~ ",
    formula_sufix = "",
    data = "data_regression_AD",
    family = "binomial"),
  Model_2 = c(name = "ADSex",
    formula_prefix = "AD_CTL ~ ",
    formula_sufix = " + Sex",
    data = "data_regression_AD",
    family = "binomial"),
  Model_3 = c(name = "ADAPOE",
    formula_prefix = "AD_CTL ~ ",
    formula_sufix = " + e4_c",
    data = "data_regression_AD",
    family = "binomial"),
  Model_4 = c(name = "ADSite",
    formula_prefix = "AD_CTL ~ ",
    formula_sufix = " + Site",
    data = "data_regression_AD",
    family = "binomial"),
  Model_5 = c(name = "ADAge",
    formula_prefix = "AD_CTL ~ ",
    formula_sufix = " + Age",
    data = "data_regression_AD",
    family = "binomial"),

```

```

Model_6 = c(name = "ADall",
            formula_prefix = "AD_CTL ~ ",
            formula_sufix = " + Sex + e4_c + Site + Age ",
            data = "data_regression_AD",
            family = "binomial"))

#Data frame to populate
regression_summary <- data.frame("tmp" = c())

#Loop over each regression model
for (j in names(regression_models)){

  #Loop over each lipid
  for (i in tmp_lipid_names_clean){

    #Formula
    tmp_formula <- as.formula( paste0(regression_models[[j]][["formula_prefix"]], i,
    regression_models[[j]][["formula_sufix"]]))

    #Data
    tmp_data <- eval(as.symbol(regression_models[[j]][["data"]]))

    #Model type
    tmp_family <- regression_models[[j]][["family"]]

    #Fit logistic regression model
    model_fit <- glm(formula = tmp_formula,
                     data = tmp_data,
                     family = tmp_family)

    #Model suffix
    tmp_model_suffix <- regression_models[[j]][["name"]]

    #Extract summary statistics
    regression_summary[i, paste0("Estimate_", tmp_model_suffix)] <- summary(model_fit)$coefficients[i, "Est."
    regression_summary[i, paste0("StdError_", tmp_model_suffix)] <- summary(model_fit)$coefficients[i, "Std."
    regression_summary[i, paste0("Pval_", tmp_model_suffix)] <- summary(model_fit)$coefficients[i, which(gr
    regression_summary[i, paste0("Model_", tmp_model_suffix)] <- format(tmp_formula)

  }

  #Adjust for multiple testing
  regression_summary[, paste0("PvalFDR_", tmp_model_suffix)] <- p.adjust(regression_summary[, paste0("Pval
    method = "fdr")

}

#change the rownames into an actual column
regression_summary$Lipid_clean <- rownames(regression_summary)

```

```

regression_summary$Lipid <- tmp_lipid_names

#Empty tibble to populate
regression_results <- c()
tmp_order <- c()

#Stack summary results for heatmap
for (j in names(regression_models)){
  regression_results <- regression_results %>%
    rbind(.,
    tibble(regression_summary) %>%
    select(contains(regression_models[[j]][["name"]]), Lipid) %>%
    rename_with(~gsub(paste0("_", regression_models[[j]][["name"]]), "", .)) %>%
    mutate(Outcome = regression_models[[j]][["name"]]))
  tmp_order <- append(tmp_order, regression_models[[j]][["name"]])
}

#Data wrangling
regression_results <- regression_results %>%
  mutate(Lipid = factor(Lipid, levels = rev(tmp_lipid_names))) %>%
  mutate(Outcome = gsub("AD", "", Outcome)) %>%
  relocate(Model, .after = Outcome)

# #Export supplementary table
# vroom_write(regression_results, here("data/sup_table_lipid_sensitivity_v1.4.csv"))

#Clean
rm(data_regression_AD, regression_models, regression_summary,
  tmp_data, i, j, tmp_family, tmp_model_suffix, tmp_order, tmp_lipid_names,
  tmp_lipid_names_clean, model_fit)

```