ANM_Lipid_Mediation_Analysis

Asger Wretlind

2024-05-29

```
#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
```

```r
#Set color palette
color_palette <- c("#11A1B7", "#FF660C", "#0CA61E", "#FE3C1A",
                   "#9966CC", "#4DDF2C", "#FE5387", "#85D0AB",
                   "#18548A", "#FCBB0B", "#FD908F", "#DF56BD", "#F0E4AD")

#Setting seed, since bootstrapping include some level of randomness
set.seed(123)

#Use resdidual of adjusting for Site
use_residuals <- TRUE
```

```r
#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.
```

```r
#Load in linear regression data
reg_data <- vroom(here("data/sup_table_lipid_regression_v1.6.csv"))
```

```
## Rows: 8576 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (3): Model, Lipid, Outcome
## dbl (4): Estimate, StdError, Pval, PvalFDR
##
## 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.
```

```r
#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)
}
```

```r
#Run this chunk if residuals are needed
if(use_residuals){

#temporary data set
tmp_data <- data
```

```r
colnames(tmp_data) <- Clean_names(colnames(tmp_data))

#Loop iterating over each lipid and overwrite with residuals
for (i in colnames(tmp_data)[(which(colnames(tmp_data) %in% "Cerd420")):length(tmp_data)]){

    tmp_formula <- as.formula(paste0(i, "~ Site"))

    tmp_model <- glm(tmp_formula, data = tmp_data)

    tmp_data[, i] <- residuals(tmp_model)

}

#Set names back to original
colnames(tmp_data) <- colnames(data)

#Use residual data
data <- tmp_data

}

rm(use_residuals)
```

```r
library(mediation)
```

```
## Indlæser krævet pakke: MASS

## Warning: pakke 'MASS' blev bygget under R version 4.3.1

##
## Vedhæfter pakke: 'MASS'

## Det følgende objekt er maskeret fra 'package:dplyr':
##
##     select

## Indlæser krævet pakke: Matrix

## Warning: pakke 'Matrix' blev bygget under R version 4.3.1

##
## Vedhæfter pakke: 'Matrix'

## De følgende objekter er maskerede fra 'package:tidyr':
##
##     expand, pack, unpack

## Indlæser krævet pakke: mvtnorm

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

```
## Indlæser krævet pakke: sandwich

## mediation: Causal Mediation Analysis
## Version: 4.5.0
```

```
Mediation_extract <- function(independent, dependent, mediator, data_med){

    #Remove missing variables from data
    data_complete <- data_med %>%
        filter(!is.na(get(dependent))) %>%
        filter(!is.na(get(independent))) %>%
        filter(!is.na(get(mediator)))

    #Clean special characters
    colnames(data_complete) <- Clean_names(colnames(data_complete))

    #Save variables to global environment, as bootstrap loop will forget them otherwise
    assign("independent_clean", Clean_names(independent), envir = globalenv())
    assign("dependent_clean", Clean_names(dependent), envir = globalenv())
    assign("mediator_clean", Clean_names(mediator), envir = globalenv())

    #Model 1 - Total effect (X -> Y)
    model_1 <- glm(
        formula = as.formula(paste0(dependent_clean, " ~ ", independent_clean)),
        data = data_complete)

    #Model 2 - Indirect effect (X -> M)
    model_2 <- glm(
        formula = as.formula(paste0(mediator_clean, " ~ ", independent_clean)),
        data = data_complete)

    #Model 3 - Indirect and direct effect (X + M -> Y)
    model_3 <- glm(
        formula = as.formula(paste0(dependent_clean, " ~ ", independent_clean, " + ", mediator_clean)),
        data = data_complete)

    #Model 4 -  (M -> Y)
    model_4 <- glm(
        formula = as.formula(paste0(dependent_clean, " ~ ", mediator_clean)),
        data = data_complete)


    #Mediation
    mediation_results <- mediate(model.m = model_2, model.y = model_3,
                treat = independent_clean,
                mediator = mediator_clean,
                boot = TRUE, sims = 500)

    #Output table
    table_out <- data.frame("tmp" = c())

    table_out[mediator, "independent"] <- independent
    table_out[mediator, "dependent"] <- dependent
    table_out[mediator, "mediator"] <- mediator
```

```r
    #Model 1
    #Estimate
    table_out[mediator, "model1_estimate"] <-
        summary(model_1)$coefficients[independent_clean,
                                      which(grepl("Estimate",
                    colnames(summary(model_1)$coefficients)))]

    #p-value
    table_out[mediator, "model1_pvalue"] <-
        summary(model_1)$coefficients[independent_clean,
                                      which(grepl("Pr\\(>",
                    colnames(summary(model_1)$coefficients)))]

    #Model 2
    #Estimate
    table_out[mediator, "model2_estimate"] <-
        summary(model_2)$coefficients[independent_clean,
                                      which(grepl("Estimate",                    colnames(summa
    #p-value
    table_out[mediator, "model2_pvalue"] <-
        summary(model_2)$coefficients[independent_clean,
                                      which(grepl("Pr\\(>",                   colnames(summary

    #Model 3
    #Estimate
    table_out[mediator, "model3_estimate"] <-
        summary(model_3)$coefficients[independent_clean,
                                      which(grepl("Estimate",                     colnames(summa

    #p-value
    table_out[mediator, "model3_pvalue"] <-
        summary(model_3)$coefficients[independent_clean,
                                      which(grepl("Pr\\(>",                       colnames(summa

    #Model 4
    #Estimate
    table_out[mediator, "model4_estimate"] <-
        summary(model_4)$coefficients[mediator_clean,
                                      which(grepl("Estimate",                     colnames(summa

    #p-value
    table_out[mediator, "model4_pvalue"] <-
        summary(model_4)$coefficients[mediator_clean,
                                      which(grepl("Pr\\(>",                     colnames(summa


    #Mediate ADE Estimate
    table_out[mediator, "ADE_estimate"] <- mediation_results$z0

    #Mediate ADE p-value
    table_out[mediator, "ADE_pvalue"] <- summary(mediation_results) %>%
        capture.output() %>%
        .[grepl("ADE", .)] %>%
```

```
        .[length(.)] %>%
        str_split(" ") %>%
        unlist() %>%
        .[nzchar(.)] %>%
        .[!grepl("\\*", .)] %>%
        .[!nchar(.) == 1] %>%
        .[length(.)]

    #Mediate Total_effect Estimate
    table_out[mediator, "TotE_estimate"] <- mediation_results$tau.coef

    #Mediate Total_effect p-value
    table_out[mediator, "TotE_pvalue"] <- summary(mediation_results) %>%
        capture.output() %>%
        .[grepl("Total Effect", .)] %>%
        .[length(.)] %>%
        str_split(" ") %>%
        unlist() %>%
        .[nzchar(.)] %>%
        .[!grepl("\\*", .)] %>%
        .[!nchar(.) == 1] %>%
        .[length(.)]

    #Mediate ACME Estimate
    table_out[mediator, "ACME_estimate"] <- mediation_results$d0

    #Mediate ACME p-value
    table_out[mediator, "ACME_pvalue"] <- summary(mediation_results) %>%
        capture.output() %>%
        .[grepl("ACME", .)] %>%
        .[length(.)] %>%
        str_split(" ") %>%
        unlist() %>%
        .[nzchar(.)] %>%
        .[!grepl("\\*", .)] %>%
        .[!nchar(.) == 1] %>%
        .[length(.)]

    #Mediate proportion Estimate
    table_out[mediator, "Prop_estimate"] <- mediation_results$n.avg

    #Mediate proportion p-value
    table_out[mediator, "Prop_pvalue"] <- mediation_results$n.avg.p

    #Clean global environment variables
    rm(independent_clean, dependent_clean, mediator_clean, envir = globalenv())

    return(table_out)
}
```

```
tmp_lipids <- tibble(reg_data) %>%
    filter(Outcome == "AD Female not Adjusted for APOE" |
               Outcome == "AD Male not Adjusted for APOE")  %>%
```

```
    filter(PvalFDR < 0.05) %>%
    pull(Lipid)
```

```
#Vector of independent variables (lipids)
vec_independent <- tmp_lipids

#Vector of mediators (Confounders)
vec_mediators <- c("Total_C",  "Total_TG", "HDL_C", "LDL_C", "ApoB")


#Note that variables with missing values or "factor names" will cause trouble
data_med <- data %>%
    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)) %>%
    filter(Sex == "Female")


#Table to populate
mediate_summary <- data.frame("tmp" = c())

#Outer loop through independent variables
for (j in vec_independent){

    #Inner loop trough mediators
    for (i in vec_mediators){

        #Mediation
        mediate_summary <- mediate_summary %>%
            rbind(., Mediation_extract(independent = j,
                                       dependent = "AD_CTL",
                                       mediator = i,
                                       data_med = data_med))

    }
}
```

```
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
```

```
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
```

```
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
```

```
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
```

```
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
```

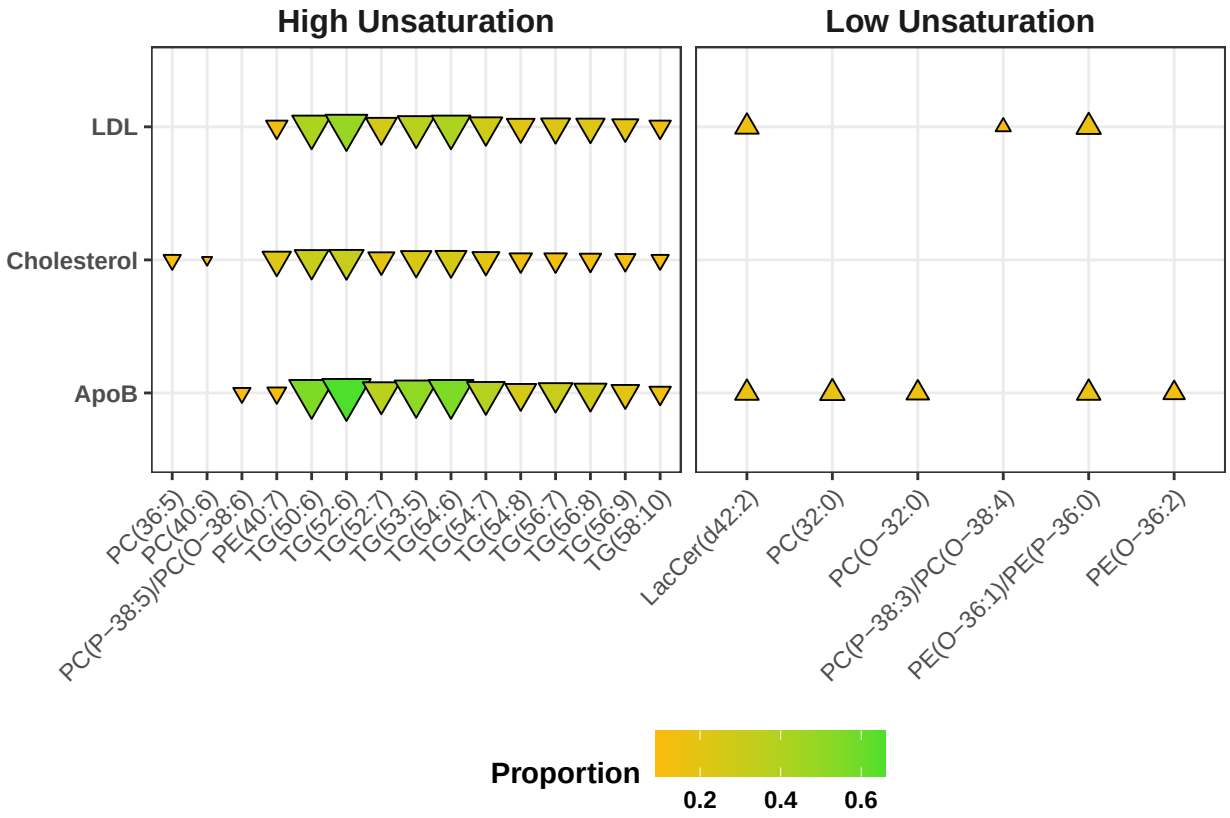
```
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
```

```
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
##
## Running nonparametric bootstrap
```

```r
#Set rownames to numbers
rownames(mediate_summary) <- 1:nrow(mediate_summary)

rm(i, j, vec_mediators, vec_independent)
```

```r
#Get regression direction (AD adjusted up or down regulation)
tmp_direction <- reg_data %>%
    filter(Outcome == "AD Female not Adjusted for APOE") %>%
    mutate(Direction = ifelse(Estimate > 0, "Positive", "Negative")) %>%
    dplyr::select(Lipid, Direction) %>%
    rename(independent = Lipid)

#Data wrangling for plot
data_tmp <- mediate_summary %>%
    filter(ACME_pvalue < 0.05) %>%
    filter(model1_pvalue < 0.05 &
           model2_pvalue < 0.05 &
           model3_pvalue < 0.05 &
           model4_pvalue < 0.05) %>%
```

```r
    #filter(independent != "e4_c") %>%
    mutate(Proportion = round(abs(Prop_estimate), 2)) %>%
    left_join(., tmp_direction, by = "independent") %>%
    mutate(independent = gsub("_A", "", independent)) %>%
    mutate(independent = gsub("_B", "", independent)) %>%
    mutate(mediator = gsub("_C", "", mediator)) %>%
    mutate(mediator = gsub("Total", "Cholesterol", mediator)) %>%
    mutate(Saturation = if_else(as.numeric(str_extract(independent, "(?<=:)[0-9]+(?=\\))")) >= 5, "High

#Correct specific lipid
data_tmp$Saturation[data_tmp$independent == "PC(P-36:4)/PC(O-36:5)"] <- "High Unsaturation"

#Create the plot
Mediation_plot <- ggplot(data_tmp, aes(x = independent,
                                       y = mediator,
                    bg = Proportion,
                    size = Proportion,
                    shape = Direction))+
    geom_point()+
    scale_shape_manual(values = c(25, 24), guide = "none")+
    scale_size_continuous(guide = "none") +
    scale_color_gradient(high = "#4DDF2C", low = "#FCBB0B",
                         aesthetics = "bg",
                         breaks = c(0.2, 0.4, 0.6))+
    facet_wrap(~ Saturation, scales = "free_x", nrow = 1)+
    theme_bw()+
    theme(
        axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title = element_blank(),
        legend.position = "bottom",
        strip.background = element_blank(),
        strip.text = element_text(face = "bold", size = 12),
        axis.text.y = element_text(face = "bold"),
        legend.text = element_text(face = "bold"),
        legend.title = element_text(face =  "bold"))

# #Save figure
# pdf(here("figures/Mediation_plot_v1.4.pdf"), width = 8, height = 3)

Mediation_plot
```




```
# dev.off()


# #Save plot object
# saveRDS(Mediation_plot, here("data/Mediation_plot_v1.4.rds"))

# #Save supplementary table
# vroom_write(mediate_summary, here("data/sup_table_mediation_v1.0.csv"))

rm(data_med, data_tmp, tmp_data, tmp_direction, tmp_model, tmp_formula, tmp_lipids)
```

```
library(ggdag)
```

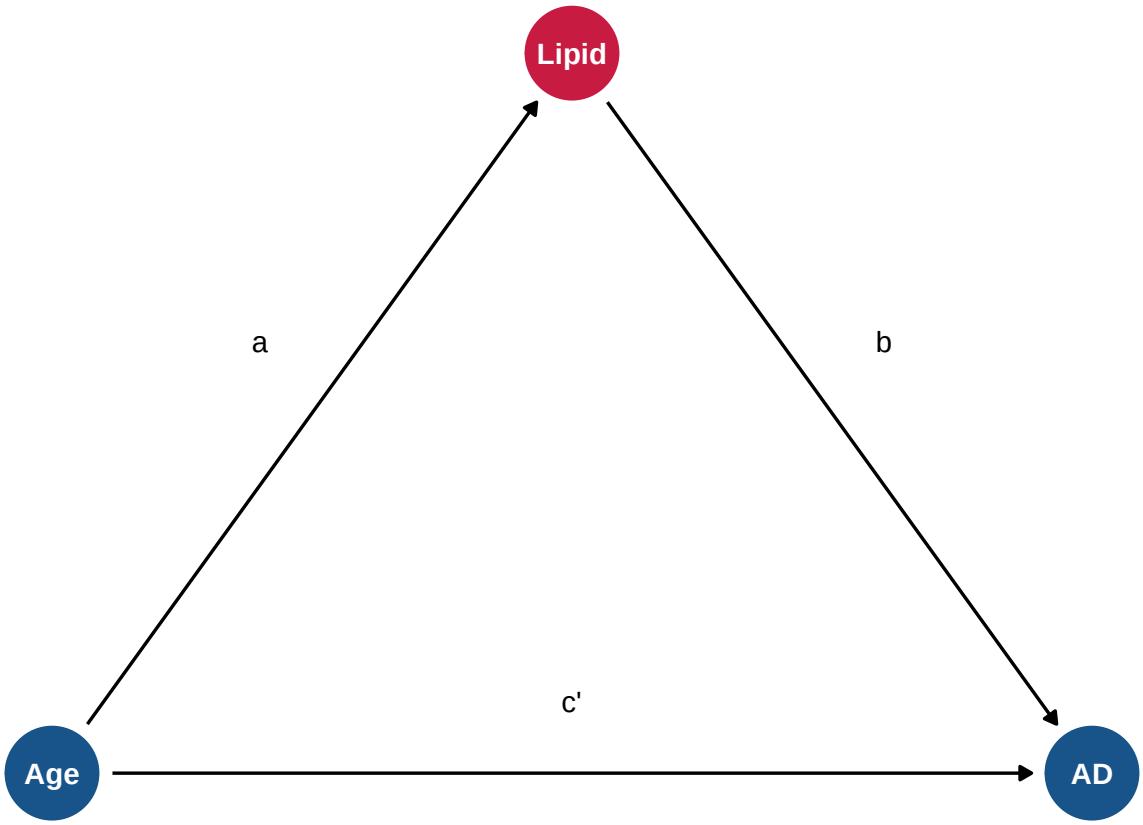
```
## Warning: pakke 'ggdag' blev bygget under R version 4.3.3
```

```
##
## Vedhæfter pakke: 'ggdag'
```

```
## Det følgende objekt er maskeret fra 'package:stats':
##
##     filter
```

```r
#Work in progress

# #Select mediation analysis to plot
# tmp_mediate <- mediate_summary[mediate_summary$mediator == "PC(O-32:0)" &
#                                  mediate_summary$independent == "Sex",]
#
#
# tmp_mediate$mediator <- Clean_names(tmp_mediate$mediator)
#
# #Coordinates
# x_coord <- c(1, 2, 3)
# names(x_coord) <- c(tmp_mediate$independent, tmp_mediate$mediator, tmp_mediate$dependent)
#
# y_coord <- c(0, 1, 0)
# names(y_coord) <- c(tmp_mediate$independent, tmp_mediate$mediator, tmp_mediate$dependent)
#
# coord_dag <- list(x = x_coord, y = y_coord)
#
# #Create graph
# dag <- dagify(formula(paste(tmp_mediate$dependent, " ~ ", tmp_mediate$independent)),
#               formula(paste(tmp_mediate$mediator, " ~ ", tmp_mediate$independent)),
#               formula(paste(tmp_mediate$dependent, " ~ ", tmp_mediate$mediator)),
#               coords = coord_dag) %>%
#     tidy_dagitty() %>%
#     mutate(colour = ifelse(name == "lipid", "farve1", "farve2"))



#### #Working

#Coordinates
coord_dag <- list(
  x = c(Age = 1, Lipid = 2, AD = 3),
  y = c(Age = 0, Lipid = 1, AD = 0))

#Create graph
dag <- dagify(AD ~ Age,
              Lipid ~ Age,
              AD ~ Lipid,
              coords = coord_dag) %>%
    tidy_dagitty() %>%
    mutate(colour = ifelse(name == "Lipid", "farve1", "farve2"))

#Visualize
dag_plot <- dag %>%
    ggplot(aes(x = x, y = y, xend = xend, yend = yend)) +
    geom_dag_point(aes(colour = colour))+
    scale_color_manual(values = c("#C71B42", "#18548A"))+
    geom_dag_edges() +
    geom_dag_text() +
    annotate("text", x = c(1.4, 2.0, 2.6),
                     y = c(0.6, 0.1, 0.6),
             label = c("a", "c'", "b"))+
```

```
    theme_void()+
    theme(legend.position = "none")

# #Save figure
# pdf(here("figures/Directed_acylic_graph_v1.1.pdf"), width = 5, height = 3)

dag_plot
```




```
# dev.off()

# #Save plot object
# saveRDS(dag_plot, here("data/dag_plot_v1.0.rds"))
```

```
#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_AD <- data
colnames(data_AD) <- Clean_names(colnames(data_AD))

#Subset only AD and control samples
data_AD <- data_AD %>%
    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)) %>%
    mutate(Sex_bin = if_else(Sex == "Female", 1, NA)) %>%
    mutate(Sex_bin = if_else(Sex == "Male", 0, Sex_bin)) %>%
    relocate(Sex_bin, .after = Sex) %>%
    data.frame()

#TG5811
#PCO320

#Step 1 - Total effect
#Sex ---> AD
Model_1 <- glm(AD_CTL ~ Sex_bin, data = data_AD, family = "binomial")
summary(Model_1)
```

```
##
## Call:
## glm(formula = AD_CTL ~ Sex_bin, family = "binomial", data = data_AD)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.19845    0.12164  -1.631    0.103
## Sex_bin      0.01431    0.15750   0.091    0.928
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 931.07  on 675  degrees of freedom
## Residual deviance: 931.06  on 674  degrees of freedom
## AIC: 935.06
##
## Number of Fisher Scoring iterations: 3
```

```
#Step 2 - Indirect effect
#Sex ---> Mediator (Lipid)
Model_2 <- glm(PCO340 ~ Sex_bin, data = data_AD)
summary(Model_2)
```

```
##
## Call:
## glm(formula = PCO340 ~ Sex_bin, data = data_AD)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.028281   0.008209  -3.445 0.000607 ***
## Sex_bin      0.055215   0.010633   5.193 2.74e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.0183989)
##
##     Null deviance: 12.897  on 675  degrees of freedom
## Residual deviance: 12.401  on 674  degrees of freedom
## AIC: -778.53
```

```
##
## Number of Fisher Scoring iterations: 2
```

```
#Step 3 - Indirect and direct effect
#    -->Mediator-->
#Sex ------------> AD
Model_3 <- glm(AD_CTL ~ Sex_bin + TG5811, data = data_AD, family = "binomial")
summary(Model_3)
```

```
##
## Call:
## glm(formula = AD_CTL ~ Sex_bin + TG5811, family = "binomial",
##     data = data_AD)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.25385    0.12406  -2.046   0.0407 *
## Sex_bin      0.07833    0.16042   0.488   0.6253
## TG5811      -0.84223    0.19444  -4.332 1.48e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 931.07  on 675  degrees of freedom
## Residual deviance: 911.23  on 673  degrees of freedom
## AIC: 917.23
##
## Number of Fisher Scoring iterations: 4
```

```
mediation_results <- mediate(model.m = Model_2, model.y = Model_3,
               treat = "Sex_bin", mediator = "TG5811",
               boot = TRUE, sims = 500)
```

```
## Running nonparametric bootstrap
```

```
summary(mediation_results)
```

```
##
## Causal Mediation Analysis
##
## Nonparametric Bootstrap Confidence Intervals with the Percentile Method
##
##                          Estimate 95% CI Lower 95% CI Upper p-value
## ACME (control)           -0.01142     -0.01860        -0.01  <2e-16 ***
## ACME (treated)           -0.01151     -0.01902        -0.01  <2e-16 ***
## ADE (control)             0.01936     -0.05511         0.10    0.59
## ADE (treated)             0.01927     -0.05467         0.10    0.59
## Total Effect              0.00785     -0.06790         0.09    0.78
## Prop. Mediated (control) -1.45481     -5.79272         3.42    0.78
## Prop. Mediated (treated) -1.46634     -5.84196         3.43    0.78
## ACME (average)           -0.01146     -0.01884        -0.01  <2e-16 ***
```

```
## ADE (average)             0.01931     -0.05489         0.10    0.59
## Prop. Mediated (average) -1.46057     -5.81734         3.43    0.78
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Sample Size Used: 676
##
##
## Simulations: 500
```

```
#Clean
rm(tmp_lipid_names, tmp_lipid_names_clean, data_AD, Model_1, Model_2, Model_3)
```