LirAlbu_4cer_preprocessing

Asger_Wretlind

30/3/2022

```
#import raw ms peaks from MZmine step 12 export
data_raw_peaks <-
    vroom::vroom(here::here("data-raw/0097_LirAlbu_non-annotated_export_31032022.csv"))
```

```
## New names:
## Rows: 3161 Columns: 166
## -- Column specification
## -------------------------------------------------------- Delimiter: ";" dbl
## (164): row ID, row m/z, row retention time, 0097_LirAlbu_P1_11_Cal7,1_1.... lgl
## (2): row identity, ...166
## 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.
## * `` -> `...166`
```

```
# #View data to check if it is correctly loaded
# View(data_raw_peaks)
```

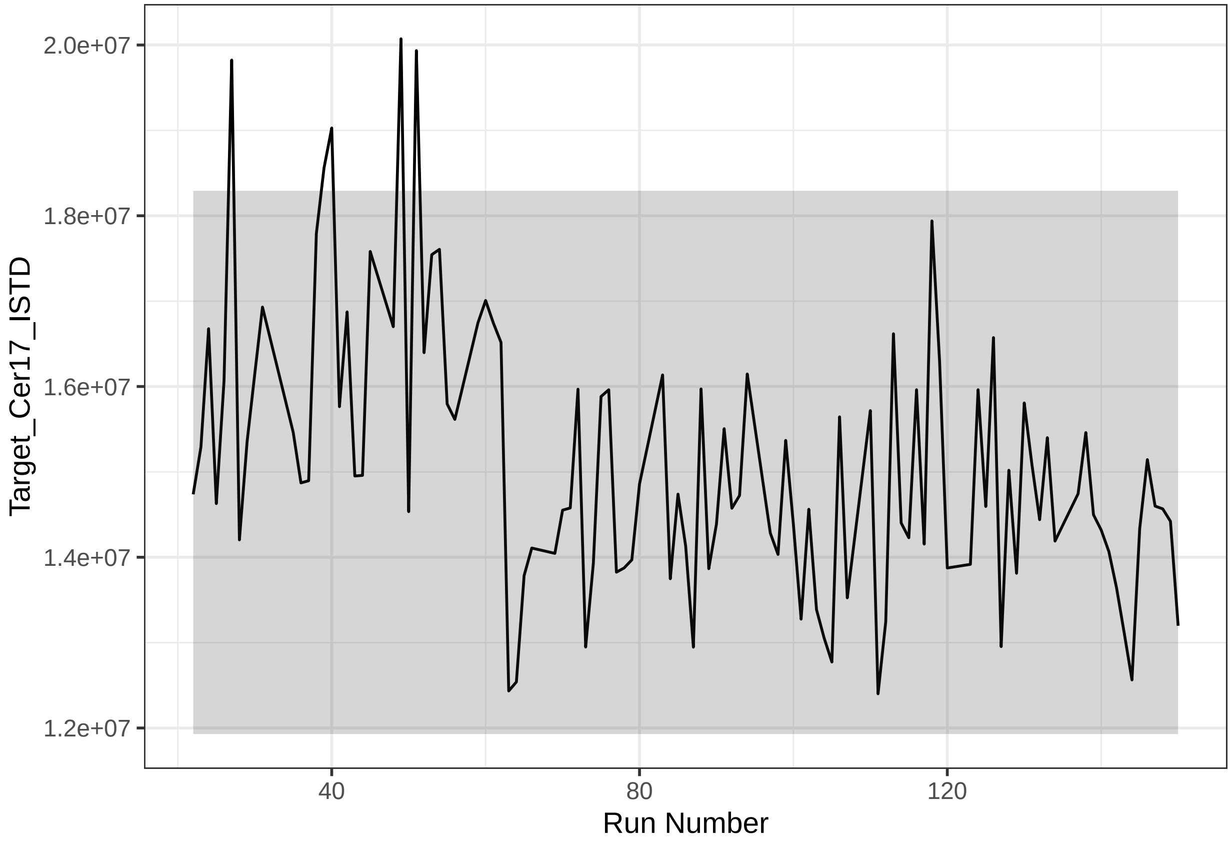
```
#Select peaks based on specific IDs
#input peak ID based on viewed chromatograms
ID_list <- c("Target_Cer17_H_ISTD" = 10,
                 "Target_Cer17_H20_ISTD" = 7,
                 #"Target_Cer16_H" = NA,
                 "Target_Cer16_H20" = 61,
                 #"Target_Cer18_H" = NA,
                 "Target_Cer18_H20" = 64,
                 "Target_Cer20_H" = 2885,
                 "Target_Cer20_H20" = 2727,
                 "Target_Cer22_H" = 622,
                 "Target_Cer22_H20" = 618,
                 "Target_Cer24:0_H" = 63,
                 "Target_Cer24:0_H20" = 59,
                 "Target_Cer24:1_H" = 2146,
                 "Target_Cer24:1_H20" = 2147)

ID_list <- ID_list[order(ID_list)]

data <- data_raw_peaks %>%
    filter(data_raw_peaks$`row ID` %in% ID_list) %>%
    arrange(`row ID`) %>%
    mutate(`row identity` = names(ID_list)) %>%
    arrange(`row identity`)
```

```r
rm(ID_list, data_raw_peaks)
```

```r
#NOTE This is project specific cleaning

#Remove weird artifact at the final data column
data <- data[,-length(data)]

#Keep only H2o adducts, remove RT, mz, row ID a and pivot table
data <- data %>%
    filter(grepl("H2O", `row identity`)) %>%
    mutate(ID = gsub("_H2O", "", `row identity`)) %>%
    select(-c("row m/z", "row retention time", "row identity", "row ID")) %>%
    pivot_longer(cols = -ID, names_to = "Steno ID") %>%
    pivot_wider(names_from = ID, values_from = value)

#Separate Steno ID into a new columns
data <- data %>%
        separate(`Steno ID`,
        c(  "Project_nr",
            "Project",
            "Plate_nr",
            "Run_nr",
            "Sample_ID",
            "Time_point",
            "Repeat"))
```

```
## Warning: Expected 7 pieces. Additional pieces discarded in 161 rows [1, 2, 3, 4, 5, 6,
## 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```r
#Run_nr_plot function
Run_nr_plot <- function(Data, Target, Filter_out = FALSE) {
    Target_pos <- which(colnames(Data) %in% Target)
    Target <- sym(Target)

    tmp_bounds <- data.frame("Mean" = NA, "SD" = NA, "lower" = NA, "upper" = NA)

    Data %>%
        filter(!grepl(Filter_out, `Sample_ID`)) %>%
        pull(Target) %>%
        mean() -> tmp_bounds$Mean

    Data %>%
        filter(!grepl(Filter_out, `Sample_ID`)) %>%
        pull(Target) %>%
        sd() -> tmp_bounds$SD

    tmp_bounds$lower <- tmp_bounds$Mean - 2*tmp_bounds$SD
    tmp_bounds$upper <- tmp_bounds$Mean + 2*tmp_bounds$SD

    p <- Data %>%
        filter(!grepl(Filter_out, `Sample_ID`)) %>%
        ggplot(aes(x = as.numeric(Run_nr), y = !!Target)) +
        geom_line() +
        geom_ribbon(aes(ymin = tmp_bounds$lower, ymax = tmp_bounds$upper),
                    alpha = 0.2) +
        xlab(label = "Run Number")+
        theme_bw()

    return(p)
}

#ISTD Run nr plot w/o QC samples
Run_nr_plot(Data = data, Target = "Target_Cer17_ISTD", Filter_out = "^[A-Za-z]+")
```



```
# data %>%
#     filter(Sample_ID == "PO") %>%
#     ggplot(aes(x = as.numeric(Run_nr), y = Target_Cer17_ISTD)) +
#                geom_line()
#
# #cer 16 Run nr plot, w/o QC samples
# Run_nr_plot(Data = data, Target = "Target_Cer16", Filter_out = "^[A-Za-z]+")
#
# data %>%
#     filter(Sample_ID == "PO") %>%
#     summarise("Cer16_median" = median(Target_Cer16),
#               "Cer17_median" = median(Target_Cer17_ISTD),
#               "Cer16_norm" = median(Target_Cer16/Target_Cer17_ISTD))
```

```
#Relative Standard deviation (RSD) of Pooled samples before normalization
data %>%
    filter(Sample_ID == "PO") %>%
    summarise(across(starts_with("Target_"), ~ sd(.)/mean(.)*100, .names = "RSD_{.col}"))
```

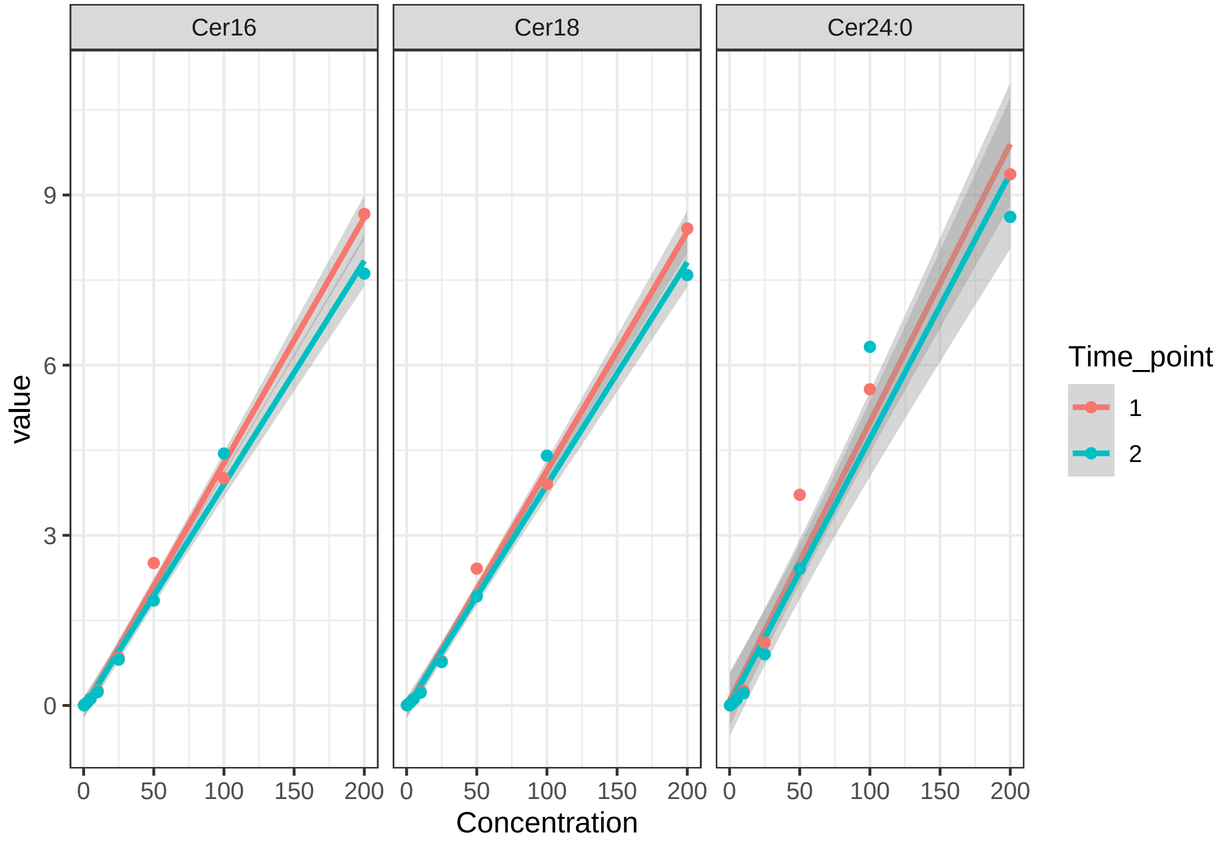
```
## # A tibble: 1 x 7
##   RSD_Target_Cer16 RSD_Target_Cer17_ISTD RSD_T~1 RSD_T~2 RSD_T~3 RSD_T~4 RSD_T~5
##              <dbl>                 <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1             10.2                  15.7    13.7    14.2    13.9    16.2    12.5
## # ... with abbreviated variable names 1: RSD_Target_Cer18, 2: RSD_Target_Cer20,
## #   3: RSD_Target_Cer22, 4: `RSD_Target_Cer24:0`, 5: `RSD_Target_Cer24:1`
```

```
#Normalize to ISTD
data <- data %>%
    mutate(across(starts_with("Target") & !contains("ISTD"),
                  ~ ./Target_Cer17_ISTD))

#Relative Standard deviation (RSD) of Pooled samples after normalization
data %>%
    filter(Sample_ID == "PO") %>%
    summarise(across(starts_with("Target_"), ~ sd(.)/mean(.)*100, .names = "RSD_{.col}"))
```

```
## # A tibble: 1 x 7
##   RSD_Target_Cer16 RSD_Target_Cer17_ISTD RSD_T~1 RSD_T~2 RSD_T~3 RSD_T~4 RSD_T~5
##              <dbl>                 <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1             8.72                  15.7    6.30    9.13    9.70    14.0    17.7
## # ... with abbreviated variable names 1: RSD_Target_Cer18, 2: RSD_Target_Cer20,
## #   3: RSD_Target_Cer22, 4: `RSD_Target_Cer24:0`, 5: `RSD_Target_Cer24:1`
```

```
#Concentration for each calibration levels in ug/mL
Cal_con <-
    tibble( Sample_ID =
      c("Cal1", "Cal2", "Cal3", "Cal4", "Cal5", "Cal6", "Cal7", "Cal8", "Cal9", "Cal10"),
      Concentration = c(200, 100,  50,  25,  10,   5, 2.5,   1, 0.5, 0.25))

#Plot calibration curves
data %>%
    filter(grepl("Cal", Sample_ID)) %>%
    left_join(x = .,y = Cal_con, by = "Sample_ID") %>%
    pivot_longer(cols = starts_with("Target"), names_to = "Ceramide") %>%
    filter(Ceramide == "Target_Cer16" |
           Ceramide == "Target_Cer18" |
           Ceramide == "Target_Cer24:0") %>%
    mutate(Ceramide = gsub("Target_", "", Ceramide)) %>%
    ggplot(aes(x = Concentration, y = value, color = Time_point)) +
    geom_smooth(method = "lm") +
    geom_point() +
    facet_wrap(. ~Ceramide) +
    theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# #Extract the mean measured value from both calibration curves of the three standards
# #and add them to Cal_con
# Cal_con <- data %>%
#     filter(grepl("Cal", Sample_ID)) %>%
#     pivot_wider(names_from = Sample_ID,
#                 values_from = c(Target_Cer16, Target_Cer18, `Target_Cer24:0`)) %>%
#     summarise(across(contains("Cal"), ~ mean(., na.rm = TRUE))) %>%
#     pivot_longer(everything(), values_to = "mean_con") %>%
#     mutate(name = gsub("24:0", "24", name)) %>%
#     separate(name, c("Prefix", "Cer", "Sample_ID")) %>%
#     unite(Cer, c(Prefix, Cer)) %>%
#     pivot_wider(names_from = Cer, values_from = mean_con) %>%
#     left_join(Cal_con, ., by = "Sample_ID")

#Use all individual points rather than mean
Cal_con <- data %>%
    filter(grepl("Cal", Sample_ID)) %>%
    select(Sample_ID, Time_point, Target_Cer16, Target_Cer18, `Target_Cer24:0`) %>%
    left_join(Cal_con, .) %>%
    select(-Time_point) %>%
    rename(Target_Cer24 = `Target_Cer24:0`)
```

```
## Joining with `by = join_by(Sample_ID)`
```

```
## Warning in left_join(Cal_con, .): Each row in `x` is expected to match at most 1 row in `y`.
## i Row 1 of `x` matches multiple rows.
## i If multiple matches are expected, set `multiple = "all"` to silence this
##   warning.
```

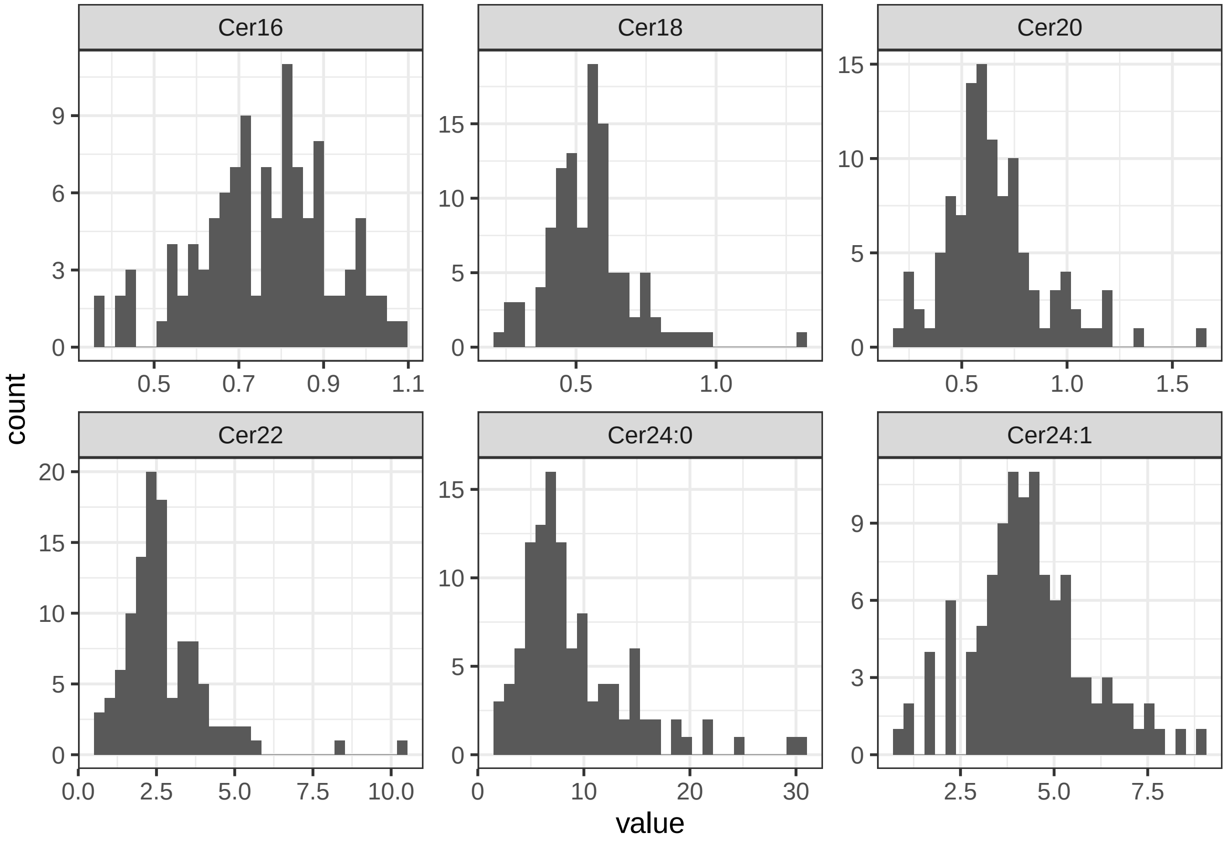
```
#Add a point zero
Cal_con[nrow(Cal_con)+1, "Sample_ID"]  <-
    c(paste0("Cal", length(unique(Cal_con$Concentration))+1))
Cal_con[nrow(Cal_con),2:ncol(Cal_con)] <- 0

#Create linear models from each standard and extract line parameters
CalCurve_Par <- Cal_con %>%
    summarise(across(starts_with("Target_"),
        list(slope = ~ lm(.x~ Concentration)$coefficients[[2]],
             intercept = ~ lm(.x~ Concentration)$coefficients[[1]],
             rsquared = ~ summary(lm(.x~ Concentration))["adj.r.squared"][[1]]))) %>%
    pivot_longer(everything()) %>%
    separate(name, c("Prefix", "Cer", "Parameter")) %>%
    unite(Cer, c(Prefix, Cer)) %>%
    pivot_wider(names_from = Cer)

#Create a helping tibble to match metabolites with appropriate calibration curve parameter
Cal_fit <- data %>%
    select( starts_with("Target_") & !ends_with("ISTD")) %>%
    colnames() %>%
    tibble("Metabolite" = ., "Calibration_fit" = c(
        "Target_Cer16",
        rep("Target_Cer18", 2),
        rep("Target_Cer24", 3)))

#Apply calibration curve parameters to convert observed data into ug/mL concentration
#This functions takes the area "." and divides by the slope from "CalCurve_Par" of chosen calibration o
data_tmp <- data %>%
    mutate(across(starts_with("Target_") & ! ends_with("ISTD"),
                  .names = "{.col}_ugmL",
                  ~ (.  / CalCurve_Par[CalCurve_Par$Parameter == "slope",
                  Cal_fit$Calibration_fit[Cal_fit$Metabolite == cur_column()]][[1]])*14))


#Evaluate calibration curve fit
data_eval <- data_tmp %>%
    filter(grepl("Cal", Sample_ID)) %>%
    left_join(x = .,y = Cal_con[,c("Sample_ID", "Concentration")], by = "Sample_ID") %>%
    select(Sample_ID, Time_point, Concentration,
           Target_Cer16_ugmL, Target_Cer18_ugmL, `Target_Cer24:0_ugmL`) %>%
    mutate(across(contains("ugmL"),
                  list(CalAcc = ~ ./Concentration*100,
                       CalAccDif = ~ abs((./Concentration*100)-100)))) %>%
    select(Sample_ID, Time_point, Concentration,
           Target_Cer16_ugmL, Target_Cer18_ugmL, `Target_Cer24:0_ugmL`,
           contains("CalAccDif")) %>%
    filter(!duplicated(Sample_ID))
```

```
## Warning in left_join(x = ., y = Cal_con[, c("Sample_ID", "Concentration")], : Each row in `x` is expe
## i Row 1 of `x` matches multiple rows.
## i If multiple matches are expected, set `multiple = "all"` to silence this
##   warning.
```

```
#Overwrite area data with calculated concentrations in ug/mL
data <- data_tmp %>%
    select(!starts_with("Target"), contains("ugmL")) %>%
    rename_with(~ gsub("_ugmL", "", .))

#remove temporary variables
rm(Cal_con, Cal_fit, CalCurve_Par, data_tmp, data_eval)
```

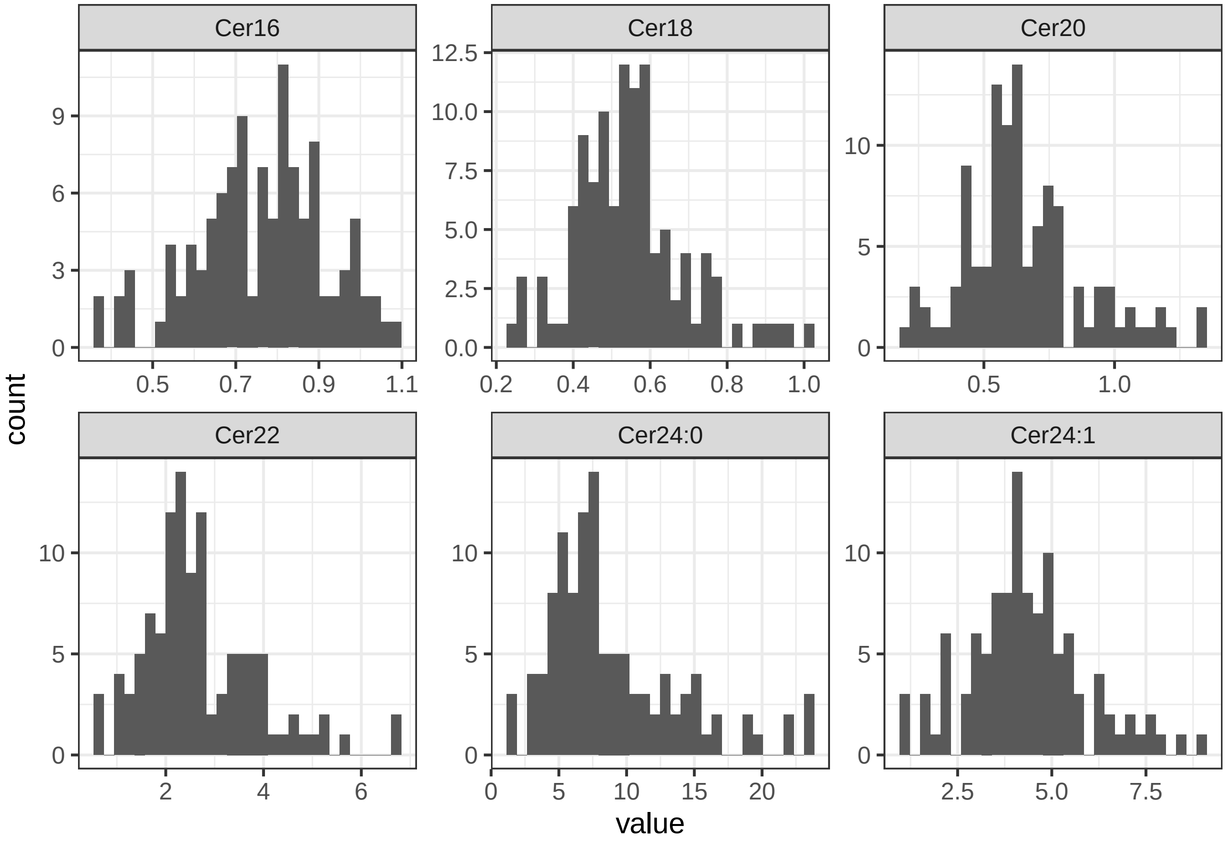
```
#Remove QC samples and unnecessary features
data <- data %>%
    filter(!grepl("^[A-Za-z]+", `Sample_ID`)) %>%
    select(-c(Project_nr, Project, Plate_nr, Run_nr, Repeat))
```

```
#plot distribution
data %>%
    pivot_longer(cols = starts_with("Target")) %>%
    separate(name, sep = "_", c("Prefix", "Ceramide")) %>%
    #mutate(value = log10(value)) %>%
    ggplot(aes(x = value)) +
    geom_histogram(bins = 30) +
    theme_bw()+
    facet_wrap(~ Ceramide, scales = "free")
```



)

```
#Truncate outliers (outside median +- 3*sd) to median +- 3*sd
data <- data %>%
  mutate(across(starts_with("Target"),
  ~ ifelse(. > median(.)+3*sd(.), median(.)+3*sd(.), .))) %>%
  mutate(across(starts_with("Target"),
  ~ ifelse(. < median(.)-3*sd(.), median(.)-3*sd(.), .)))

#plot distribution
data %>%
    pivot_longer(cols = starts_with("Target")) %>%
    separate(name, sep = "_", c("Prefix", "Ceramide")) %>%
    #mutate(value = log10(value)) %>%
    ggplot(aes(x = value)) +
    geom_histogram(bins = 30) +
    theme_bw()+
    facet_wrap(~ Ceramide, scales = "free")
```



```
#Import clinical measurements
data_raw_clinical <-
    readxl::read_xlsx(path = here::here("data-raw/Liralbu_clinical_data.xlsx"))

#Remove 38 from the patient ID to be able to match Steno ID and fix inconsistencies
data_clinical <- data_raw_clinical %>%
    mutate(Patient_Number = str_sub(Patient_Number, 4, 5)) %>%
    rename_with(~gsub("UAERv", "UAER_v", .), everything()) %>%
    rename_with(~gsub("Lira_v1", "v1_Lira", .), everything()) %>%
    rename_with(~gsub("Lira_v2", "v2_Lira", .), everything()) %>%
    rename_with(~gsub("Placebo_v1", "v1_Placebo", .), everything()) %>%
    rename_with(~gsub("Placebo_v2", "v2_Placebo", .), everything()) %>%
    rename_with(~gsub("Office_SBP", "OfficeSBP", .), everything()) %>%
    rename(Sequence = Sequence_Lira_Placebo_ABBA)

##Sort out visits from clinical data at stack them atop each other (like pivot longer)
#Block 1 (Placebo visit 1)
Pv1 <- data_clinical %>%
  select(c(Patient_Number, ends_with("v1_Placebo"))) %>%
  rename_with(~gsub("_v1_Placebo", "", .), everything()) %>%
  mutate(Treatment = "Placebo") %>%
  mutate(Visit = "1")

#Block 2 (Placebo visit 2)
Pv2 <- data_clinical %>%
  select(c(Patient_Number, ends_with("v2_Placebo"))) %>%
  rename_with(~gsub("_v2_Placebo", "", .), everything()) %>%
  mutate(Treatment = "Placebo") %>%
  mutate(Visit = "2")

#Block 3 (Lira visit 1)
Lv1 <- data_clinical %>%
  select(c(Patient_Number, ends_with("v1_Lira"))) %>%
  rename_with(~gsub("_v1_Lira", "", .), everything()) %>%
  mutate(Treatment = "Liraglutide") %>%
  mutate(Visit = "1")

#Block 4 (Lira visit 2)
Lv2 <- data_clinical %>%
  select(c(Patient_Number, ends_with("v2_Lira"))) %>%
  rename_with(~gsub("_v2_Lira", "", .), everything()) %>%
  mutate(Treatment = "Liraglutide") %>%
  mutate(Visit = "2")

#Bind block 1 and 2 together (placebo)
Pla <- bind_rows(Pv1, Pv2) %>%
  left_join(.,
          data_clinical[,c("Patient_Number", "CrEDTA_GFR_Placebo", "CrEDTA_ECV_Placebo")]) %>%
  rename_with(~gsub("_Placebo", "", .), everything())
```

```
## Joining with `by = join_by(Patient_Number)`
```

```
#Bind block 3 and 4 together (Liraglutide)
Lir <- bind_rows(Lv1, Lv2) %>%
  left_join(.,
          data_clinical[,c("Patient_Number", "CrEDTA_GFR_Lira", "CrEDTA_ECV_Lira")]) %>%
  rename_with(~gsub("_Lira", "", .), everything())
```

```
## Joining with `by = join_by(Patient_Number)`
```

```r
#Bind everything together and remove unneeded variables
data_clinical <- bind_rows(Pla, Lir) %>%
  left_join(data_clinical[,!grepl("Placebo", colnames(data_clinical)) &
                          !grepl("Lira", colnames(data_clinical))], .) %>%
  select(-Randomisation_Number, -Patient_Initials)
```

```
## Joining with `by = join_by(Patient_Number)`
```

```
## Warning in left_join(data_clinical[, !grepl("Placebo", colnames(data_clinical)) & : Each row in `x`
## i Row 1 of `x` matches multiple rows.
## i If multiple matches are expected, set `multiple = "all"` to silence this
##   warning.
```

```r
#Create Time_point variable for merging
data_clinical <- data_clinical %>%
  mutate(Time_point = if_else(Sequence == 0 &
                              Treatment == "Placebo" & Visit == 2, "2", "1")) %>%
  mutate(Time_point = if_else(Sequence == 0 &
                              Treatment == "Liraglutide" & Visit == 1, "3", Time_point)) %>%
  mutate(Time_point = if_else(Sequence == 0 &
                              Treatment == "Liraglutide" & Visit == 2, "4", Time_point)) %>%
  mutate(Time_point = if_else(Sequence == 1 &
                              Treatment == "Liraglutide" & Visit == 2, "2", Time_point)) %>%
  mutate(Time_point = if_else(Sequence == 1 &
                              Treatment == "Placebo" & Visit == 1, "3", Time_point)) %>%
  mutate(Time_point = if_else(Sequence == 1 &
                              Treatment == "Placebo" & Visit == 2, "4", Time_point))

#Create prepost time point
data_clinical$PrePost <-
  paste0(as.character(data_clinical$Treatment), as.character(data_clinical$Visit))

data_clinical <- data_clinical %>%
    mutate(PrePost = str_replace(PrePost, "glutide", "")) %>%
    mutate(PrePost = str_replace(PrePost, "1", "Pre")) %>%
    mutate(PrePost = str_replace(PrePost, "2", "Post"))

rm(data_raw_clinical, Lir, Lv1, Lv2, Pla, Pv1, Pv2)
```

```r
#Merge and remove observations with no treatment info and arrange variables
data <- data %>%
  left_join(., data_clinical,
            by = c("Sample_ID" = "Patient_Number", "Time_point")) %>%
  filter(!is.na(Sequence)) %>%
  relocate(, c(Treatment, Visit, PrePost), .after = 1)

rm(data_clinical)
```

```r
#Remove "Target_" from metabolite names
data <- data %>%
    rename_with(~gsub("Target_", "", .))
```

```r
#Export data
vroom::vroom_write(data, here::here("data/0097_liralbu_4cer_data_preprocessed.csv"))
```