Data Preprocessing and QC

Part 1: iSLS11 Clinical Lipidomics Data Analysis Workshop

Bo Burla

Setup

We first load packages used in this part of the workshop. We will use several packages from the tidyverse which can be loaded using `library(tidyverse)`. The package `here`provides the function `here()` that returns the root of the project. `broom` provides functions to convert outputs of R functions such as `t.test` and `lm` into tidy tables (dataframes). `ggpmisc` extends `ggplot2`.

```
library(tidyverse)
library(here)
library(broom)
library(ggpmisc)
here::i_am("Part_1/Part1.qmd")
here::here()
```

```
## [1] "/Users/lsibjb/Documents/Code/iSLS11"
```

Background

Importing raw data

We start with loading the table with peak areas. It is always good to check the if the data were imported correctly, i.e. by inspecting column types. Text values within columns also be an issue.

```
d_orig <- readr::read_csv(file = here("Part_1/data/SPERFECT_SLINGpanel_MRMkit_RawAreas_clean.csv"),col_
```

```
## Rows: 519 Columns: 407
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (3): FILENAME, BATCH, QC_TYPE
## dbl (404): CE 14:0, CE 15:0, CE 16:0, CE 16:1, CE 16:2, CE 17:0, CE 17:1, CE...
##
## 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.
```

```
d_orig
```

```
## # A tibble: 519 x 407
##    FILEN~1 BATCH QC_TYPE CE 14~2 CE 15~3 CE 16~4 CE 16~5 CE 16~6 CE 17~7 CE 17~8
##    <chr>   <chr> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
```

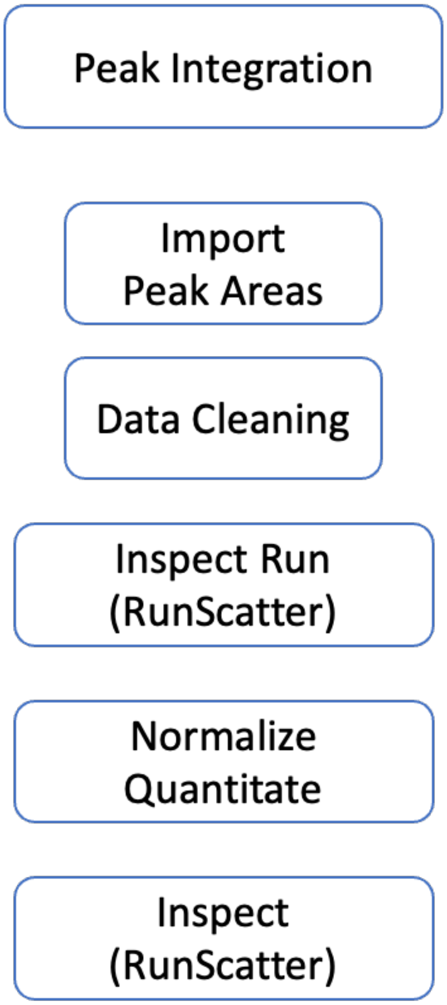


Figure 1: Hello

```
##  1 SBLK.m~ B_1   SBLK      181.    209.    1806.    584.  56.4     169.    276.
##  2 PBLK.m~ B_1   PBLK       47.3    54.6    440.    143.   0.568    19.4    96.9
##  3 UBLK.m~ B_1   UBLK       63.1    99.0    354.    122.  38.6      37.9    28.3
##  4 RQC-1-~ B_1   RQC        87.3   262.   23404.   3271. 248.      390.    518.
##  5 RQC-1-~ B_1   RQC       210.    530.   37327.   4811. 226.     1212.   1451.
##  6 RQC-1-~ B_1   RQC       335.    186.   52478.   4923. 307.     1133.    948.
##  7 RQC-1-~ B_1   RQC       189.    239.   66109.   5774. 417.     1459.   1123.
##  8 RQC-1-~ B_1   RQC       592.    230.   75214.   6100. 256.     1852.    803.
##  9 RQC-1-~ B_1   RQC       302.    173.   49464.   6516. 253.     1502.   1483.
## 10 B1_TQC~ B_1   TQC       168.    370.   46518.   7232. 237.     1380.   1980.
## # ... with 509 more rows, 397 more variables: ‘CE 18:0‘ <dbl>, ‘CE 18:1‘ <dbl>,
## #   ‘CE 18:1 d7 (ISTD)‘ <dbl>, ‘CE 18:2‘ <dbl>, ‘CE 18:3‘ <dbl>,
## #   ‘CE 20:1‘ <dbl>, ‘CE 20:2‘ <dbl>, ‘CE 20:3‘ <dbl>, ‘CE 20:4‘ <dbl>,
## #   ‘CE 20:5‘ <dbl>, ‘CE 22:0‘ <dbl>, ‘CE 22:1‘ <dbl>, ‘CE 22:4‘ <dbl>,
## #   ‘CE 22:5‘ <dbl>, ‘CE 22:6‘ <dbl>, ‘CE 24:0‘ <dbl>, ‘CE 24:1‘ <dbl>,
## #   ‘CE 24:4‘ <dbl>, ‘CE 24:5‘ <dbl>, ‘CE 24:6‘ <dbl>, ‘Cer d18:0/16:0‘ <dbl>,
## #   ‘Cer d18:0/18:0‘ <dbl>, ‘Cer d18:0/20:0‘ <dbl>, ‘Cer d18:0/22:0‘ <dbl>, ...
```

Prepare and convert to a long format table

First we clean the sample names, by removing `.mzML`, and we add the runorder number `RUN_ID` as first column. Then, we convert the data into the *long format*. In the long format every observation is a row, i.e. every lipid/sample pair is a row and peak areas are in a single column,

```
d_orig <- d_orig |>
  mutate(FILENAME = stringr::str_replace(FILENAME, ".mzML", "")) |>
  mutate(RUN_ID = row_number(), .before = 1)

d_orig
```

```
## # A tibble: 519 x 408
##    RUN_ID FILENAME BATCH QC_TYPE CE 14~1 CE 15~2 CE 16~3 CE 16~4 CE 16~5 CE 17~6
##     <int> <chr>    <chr> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1      1 SBLK     B_1   SBLK      181.    209.    1806.    584.  56.4     169.
##  2      2 PBLK     B_1   PBLK       47.3    54.6    440.    143.   0.568    19.4
##  3      3 UBLK     B_1   UBLK       63.1    99.0    354.    122.  38.6      37.9
##  4      4 RQC-1-10 B_1   RQC        87.3   262.   23404.   3271. 248.      390.
##  5      5 RQC-1-20 B_1   RQC       210.    530.   37327.   4811. 226.     1212.
##  6      6 RQC-1-40 B_1   RQC       335.    186.   52478.   4923. 307.     1133.
##  7      7 RQC-1-60 B_1   RQC       189.    239.   66109.   5774. 417.     1459.
##  8      8 RQC-1-80 B_1   RQC       592.    230.   75214.   6100. 256.     1852.
##  9      9 RQC-1-1~ B_1   RQC       302.    173.   49464.   6516. 253.     1502.
## 10     10 B1_TQC01 B_1   TQC       168.    370.   46518.   7232. 237.     1380.
## # ... with 509 more rows, 398 more variables: ‘CE 17:1‘ <dbl>, ‘CE 18:0‘ <dbl>,
## #   ‘CE 18:1‘ <dbl>, ‘CE 18:1 d7 (ISTD)‘ <dbl>, ‘CE 18:2‘ <dbl>,
## #   ‘CE 18:3‘ <dbl>, ‘CE 20:1‘ <dbl>, ‘CE 20:2‘ <dbl>, ‘CE 20:3‘ <dbl>,
## #   ‘CE 20:4‘ <dbl>, ‘CE 20:5‘ <dbl>, ‘CE 22:0‘ <dbl>, ‘CE 22:1‘ <dbl>,
## #   ‘CE 22:4‘ <dbl>, ‘CE 22:5‘ <dbl>, ‘CE 22:6‘ <dbl>, ‘CE 24:0‘ <dbl>,
## #   ‘CE 24:1‘ <dbl>, ‘CE 24:4‘ <dbl>, ‘CE 24:5‘ <dbl>, ‘CE 24:6‘ <dbl>,
## #   ‘Cer d18:0/16:0‘ <dbl>, ‘Cer d18:0/18:0‘ <dbl>, ‘Cer d18:0/20:0‘ <dbl>, ...
```

```r
d_long <- d_orig |>
  pivot_longer(names_to = "LIPID", values_to = "AREA", cols = -RUN_ID:-QC_TYPE) %>%
  arrange(LIPID)

d_long
```

```
## # A tibble: 209,676 x 6
##    RUN_ID FILENAME  BATCH QC_TYPE LIPID    AREA
##     <int> <chr>     <chr> <chr>   <chr>   <dbl>
##  1      1 SBLK      B_1   SBLK    CE 14:0 181.
##  2      2 PBLK      B_1   PBLK    CE 14:0  47.3
##  3      3 UBLK      B_1   UBLK    CE 14:0  63.1
##  4      4 RQC-1-10  B_1   RQC     CE 14:0  87.3
##  5      5 RQC-1-20  B_1   RQC     CE 14:0 210.
##  6      6 RQC-1-40  B_1   RQC     CE 14:0 335.
##  7      7 RQC-1-60  B_1   RQC     CE 14:0 189.
##  8      8 RQC-1-80  B_1   RQC     CE 14:0 592.
##  9      9 RQC-1-100 B_1   RQC     CE 14:0 302.
## 10     10 B1_TQC01  B_1   TQC     CE 14:0 168.
## # ... with 209,666 more rows
```

```r
#View(d_long) # or ALT-click
```

First look at the data: plotting responses *vs* run order

To have a first idea how the analysis went, we first look the peak areas internal standards (ISTDs) over the analysis sequence. In this analysis we included different QC samples (see (Broadhurst et al. 2018)):

- BQC: Batch QC
- TQC: Technical/Instrument QC
- NIST: NIST SRM1950 plasma
- PBLK: Process/extraction blank
- SBLK: Solvent blank
- RQC: Response QCs

We observe that some ISTDs shows drifts during the analysis.

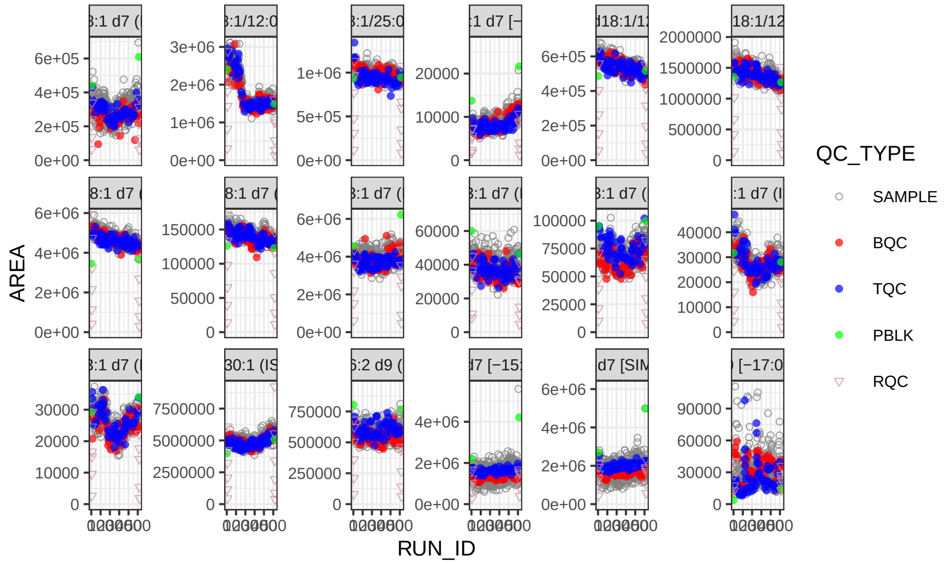
```r
# Filter for ISTDs only
d_istd <- d_long %>% filter(str_detect(LIPID, "ISTD"))
#d_plot <- d_long %>% filter(str_detect(LIPID, "ISTD") & str_detect(LIPID, "Cer"))

# Convert QC_TYPE to a factor and sort, to ensure correct layering in plot
d_istd$QC_TYPE <- factor(d_istd$QC_TYPE, c("SAMPLE", "BQC", "TQC", "PBLK", "RQC"))
d_istd <- d_istd |> arrange(QC_TYPE)

# Define colors and shapes for each QC_TYPE
qc_colors <- c(SAMPLE = "grey50", BQC = "red", TQC = "blue",
               PBLK = "green", SBLK = "darkgreen", RQC = "pink3")

qc_shapes <- c(SAMPLE = 1, BQC = 21, TQC = 21,
               PBLK = 21, SBLK = 23, RQC = 6)
```

```
# Plot
 p <- ggplot(d_istd, aes(x=RUN_ID, y=AREA)) +
       geom_point(aes(colour = QC_TYPE, fill = QC_TYPE, shape  = QC_TYPE),
                  size = 1, alpha =0.7, stroke = 0.3) +
       facet_wrap(vars(LIPID), ncol = 6, nrow = 4, scales="free_y") +
       scale_shape_manual(na.value = NA, values = qc_shapes) +
       scale_fill_manual(values = qc_colors, na.value = NA) +
       scale_colour_manual(values = qc_colors, na.value = NA) +
       scale_x_continuous(breaks = seq(0, max(d_istd$RUN_ID), by = 100 )) +
       scale_y_continuous(limits = c(0, NA)) +
       theme_bw(base_size = 8)
p
```



```
ggsave(plot = p, filename = here("Part_1/output/runscatter_ISTD.pdf"),
       width = 280, height = 180, units = "mm")
```

Checking Linear Response

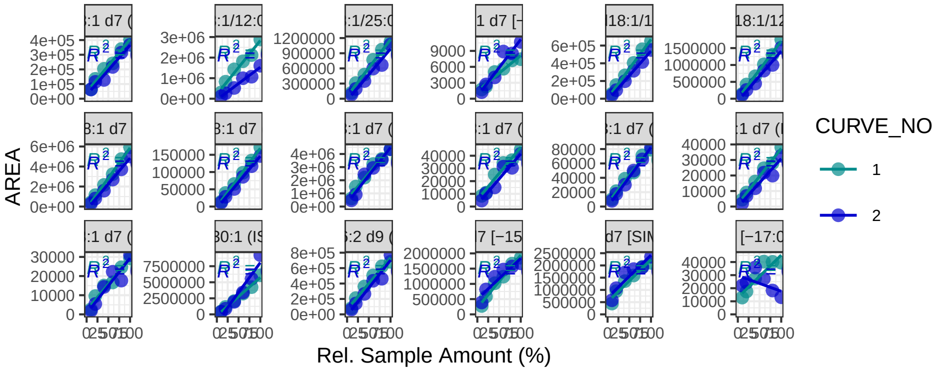
Injected sample amount need to be carefully chose when measuring analytes covering a large AREAentration range. It is a trade-off between sensitivity and not exceeding the linear range of the measurement, as well as other factors. While protocols define an optimal injected sample amount (volume), the linear range of the system can change, even within an run. We therefore always check as QC the linear response using dilution or injection volume series of a pooled QC extract.

Let's plot the response curves from ISTDs measured at the beginning and end of this run. For this we extract the curve number and relative AREAentration from the sample name.

```
d_rqc <- d_long |>
  filter(QC_TYPE == "RQC") |>
  separate(col = FILENAME,
           into = c("TYPE","CURVE_NO","AMOUNT"),
           sep = "-",
           remove = FALSE, convert = TRUE)
```

```r
d_rqc$CURVE_NO <- factor(d_rqc$CURVE_NO)
d_rqc$AMOUNT <- as.numeric(d_rqc$AMOUNT)

p <- ggplot(d_rqc |> filter(str_detect(LIPID, "ISTD")),
            aes(x=AMOUNT, y=AREA, color = CURVE_NO, group = CURVE_NO)) +
        geom_point(size = 2, alpha =0.7, stroke = 0.3) +
        facet_wrap(vars(LIPID), ncol = 6, nrow = 4, scales="free_y") +
        ggpmisc::stat_poly_line(linewidth = 0.5, se = FALSE) +
        ggpmisc::stat_poly_eq(aes(label = after_stat(rr.label)),
                     size = 2.4,
                     lineheight = 1, ) +
        scale_colour_manual(values = c("1" = "cyan4", "2" ="blue3")) +
        scale_x_continuous(limits = c(0, NA)) +
        scale_y_continuous(limits = c(0, NA)) +
        labs(x = "Rel. Sample Amount (%)") +
        theme_bw(base_size = 8)
plot(p)
```



```r
ggsave(plot = p, filename = here("Part_1/output/reponse_curves.pdf"),
       width = 280, height = 120, units = "mm")
```

Normalization and quantification

```r
d_istd_map <- readr::read_csv(file = here("Part_1/data/ISTD_mapping.csv"),
                          col_names = TRUE, trim_ws = TRUE, col_types = "c")

d_istd_conc <- readr::read_csv(file = here("Part_1/data/ISTD_conc.csv"),
                          col_names = TRUE, trim_ws = TRUE, col_types = "c")

d_processed <- d_long |>
  left_join(d_istd_map, by = c("LIPID")) |>
  left_join(d_istd_conc, by = c("ISTD")) |>
  mutate(isISTD = (LIPID == ISTD)) |>
  group_by(ISTD, FILENAME) |>
  mutate(normAREA = AREA/AREA[isISTD],
         CONC = normAREA * RF * ISTD_conc_nM/1000 * 4.5/ 10) |>
  ungroup()
```

Inspect normalized data

Normalization with the class-specific ISTD often helps to remove systematic drifts and batch effects, but may also introduce additional noise and artefacts. Let's have a look on the how the data looks after normalization.

Before we plotted the ISTD runscatter in one page, however if we would like to look at all spececies we could distribute the plots over several pages. There are different ways to archive this. One possibility is using `facet_wrap_paginate()` from the `ggforce` package, but this can be slow when having large datasets. We here are using another, manual, approach, by slicing the long table into pages that will then be plotted.

```
plot_page <- function(data, nrows, ncols){
 ggplot(data, aes(x=RUN_ID, y=CONC)) +
        geom_point(aes(colour = QC_TYPE, fill = QC_TYPE, shape  = QC_TYPE),
                   size = 1, alpha =0.7, stroke = 0.3) +
        facet_wrap(vars(LIPID), ncol = ncols, nrow = nrows, scales="free_y") +
        scale_shape_manual(na.value = NA, values = qc_shapes) +
        scale_fill_manual(values = qc_colors, na.value = NA) +
        scale_colour_manual(values = qc_colors, na.value = NA) +
        scale_x_continuous(breaks = seq(0, max(d_istd$RUN_ID), by = 100 )) +
        scale_y_continuous(limits = c(0, NA)) +
        theme_bw(base_size = 8)
 }

rows_page = 5
columns_page = 5

#get a table with page numbers for each lipid species
d_pages <- d_processed |>
  select(LIPID) |>
  distinct() |>
  mutate(page_no = ceiling(row_number() / (rows_page * columns_page)))
#plot each page from a nested table
d_plots <- d_processed %>%
  filter(!str_detect(QC_TYPE, "BLK|RQC"), !str_detect(LIPID, "ISTD")) |>
  left_join(d_pages) %>%
  nest(.by = page_no) %>%
  mutate(plt = map(data, ~ plot_page(., rows_page, columns_page)))

# Save pages to a PDF. The i
pdf(file = here("Part_1/output/run_scatter_CONC_all.pdf"),onefile = TRUE,
       width = 280/25.4, height = 180/25.4)
#d_plots$plt
invisible(walk(d_plots$plt, print)) # use this to prevent printing of index
dev.off()
```

```
## pdf
##   2
```

Calculate quality-control (QC) values for each lipid species

To evaluate the quality of the analysis and to filter the date we calculate different QC values for each lipid species. This included the analytical coefficient of variation (CV) based on the BQCs, the signal-to-blank ratio, and the r squared of the response curves.

```
rsd <- function(x) sd(x, na.rm = TRUE)/mean(x, na.rm = TRUE)

d_qc_1 <- d_processed |>
  group_by(LIPID) |>
  summarise(
    Area_SPL = median(AREA[QC_TYPE == "SAMPLE"], rm.na = TRUE),
    SB_ratio = Area_SPL/median(AREA[QC_TYPE == "PBLK"], rm.na = TRUE),
    Conc_SPL = median(CONC[QC_TYPE == "SAMPLE"], rm.na = TRUE),
    CV_TQC = rsd(CONC[QC_TYPE == "TQC"]) * 100,
    CV_BQC = rsd(CONC[QC_TYPE == "BQC"]) * 100,
    CV_SPL = rsd(CONC[QC_TYPE == "SAMPLE"]) * 100,
    D_ratio = sd(CONC[QC_TYPE == "BQC"])/sd(CONC[QC_TYPE == "SAMPLE"])
  ) |> ungroup()

f <- function(x) broom::glance(lm(AREA ~ AMOUNT, data = x))

d_qc_LM <- d_rqc |>
  nest(.by = c(LIPID, CURVE_NO)) |>
  mutate(res = purrr::map(data, f)) |>
  unnest(res)

d_qc_LM2 <- d_qc_LM |>
  select(LIPID, CURVE_NO, r.squared, p.value) |>
  pivot_wider(names_from = CURVE_NO, values_from = c(r.squared, p.value))

d_qc <- d_qc_1 |>
  left_join(d_qc_LM2)
```

```
## Joining with ‘by = join_by(LIPID)‘
```

```
d_qc <- d_qc |>
  mutate(LIPID_tmp = str_replace(LIPID, " O\\-", "-O "),
         LIPID_tmp = str_replace(LIPID, " P\\-", "-P "), .after = LIPID) |>
  separate(LIPID_tmp, into = c("CLASS", "CHAINS", "OTHER"), sep = " ", remove = TRUE, extra = "drop")


write_csv(x = d_qc, file = here("Part_1/output/QC-summary.csv"))
```

QC filter and save dataset

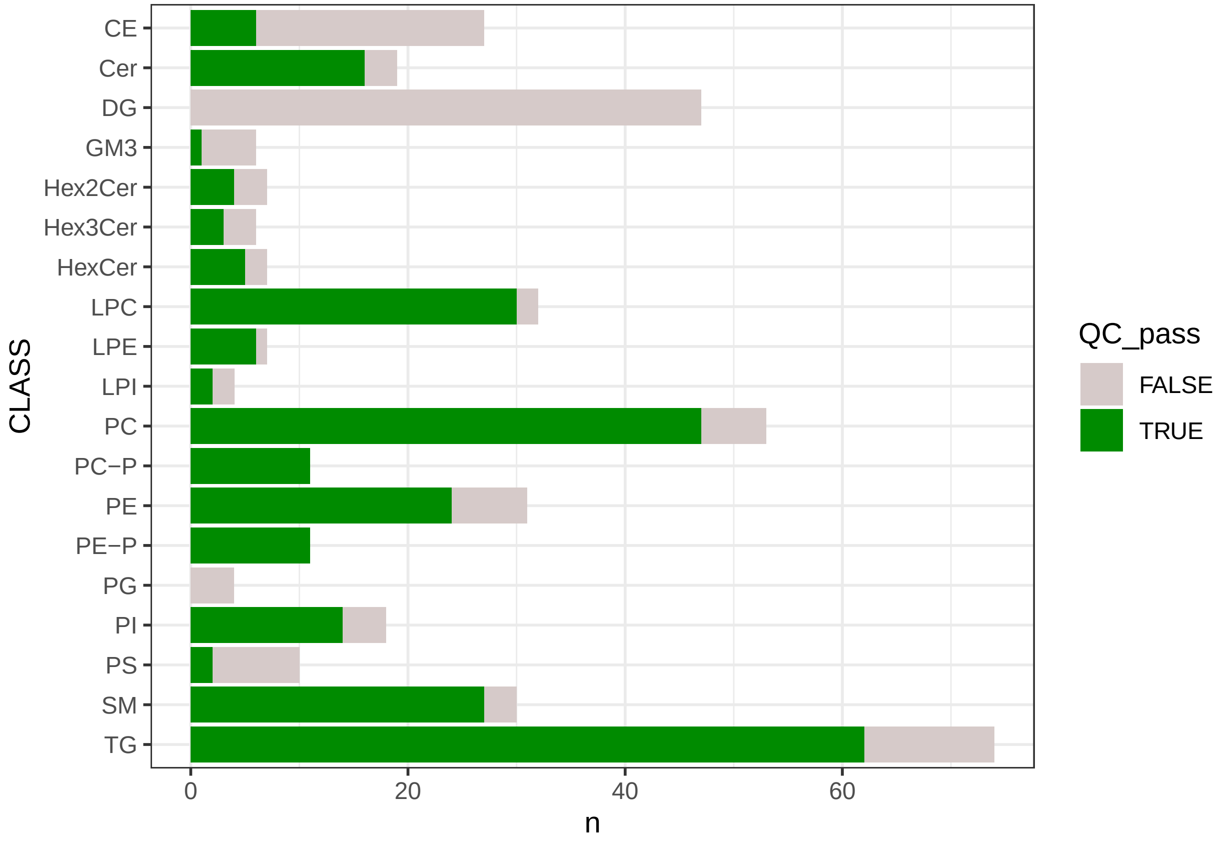
```
d_qc <- d_qc |>
  mutate(
    QC_pass =
    CV_BQC < 25 &
    SB_ratio > 3 &
    r.squared_1 > 0.8)
```

Inspect QC results

We now can (and should) have a look at how many species passed the QC criteria and if there are any pattern specific to lipid classes.

```
d_qc_summary <- d_qc |> dplyr::count(CLASS, QC_pass)

ggplot(d_qc_summary,
       aes(x = CLASS, y = n, fill = QC_pass, group = QC_pass)) +
  geom_bar(position="stack", stat="identity") +
  scale_fill_manual(values = c("FALSE" = "#d6cac9", "TRUE" = "green4")) +
  scale_x_discrete(limits=rev)+
  coord_flip() + theme_bw()
```



Parse lipid names and save final dataset

```
# QC filter data
d_final <- d_processed |>
  filter(QC_TYPE == "SAMPLE", !str_detect(LIPID, "ISTD")) |>
  right_join(d_qc[d_qc$QC_pass,"LIPID"])
```

```
## Joining with ‘by = join_by(LIPID)‘
```

```
d_final_wide <- d_final |>
  pivot_wider(id_cols = c(FILENAME, QC_TYPE), names_from = "LIPID", values_from = "CONC")

write_csv(d_final_wide, here("Part_1/output/qc_filtered_results.csv"))
```

References

Broadhurst, David, Royston Goodacre, Stacey N. Reinke, Julia Kuligowski, Ian D. Wilson, Matthew R. Lewis, and Warwick B. Dunn. 2018. “Guidelines and Considerations for the Use of System Suitability and Quality Control Samples in Mass Spectrometry Assays Applied in Untargeted Clinical Metabolomic Studies.” *Metabolomics* 14 (6): 72. https://doi.org/10.1007/s11306-018-1367-3.