SLING Data Analysis Workshop

Part 2: Post-processing of integrated peak data

Bo Burla

## 1 Importing MRMkit results with integrated peak areas

We import the peak areas obtained from peak integration using MRMkit in Part 1 of this workshop. This MRMkit result file contains the peak areas of integrated peaks (features) in all processed raw data file. In addition, the MRMkit result file reports peak retention times and widths, and metadata from the mzML files, i.e. the acquisition time-stamp and transition m/z values. We will import these metadata as well by setting use\_metadata = TRUE.

data\_path <- "./data/sPerfect\_MRMkit-20240911.tsv"  
metadata\_path <- "./data/sPerfect\_Metadata\_MidarTemplate-20241001\_All.xlsm"  
  
myexp <- midar::MidarExperiment(title = "sPerfect")  
myexp <- midar::data\_import\_mrmkit(myexp, path = data\_path, use\_metadata = TRUE)

✔ Imported 499 analyses with 503 features

ℹ `feature\_area` selected as default raw feature intensity. Use `data\_set\_intensity\_var()` to modify.

✔ Analysis metadata associated with 499 analyses.

✔ Feature metadata associated with 503 features.

## 2 A glimpse on the imported MRMkit results

Let’s have a look at the imported data be running code below or by running View(myexp@dataset\_orig) in the console. As we can see, the data is in the long format, allowing use to view multiple parameters for each analysis-feature pair.

Explore the different columns and use filter function in the RStudio table viewer.

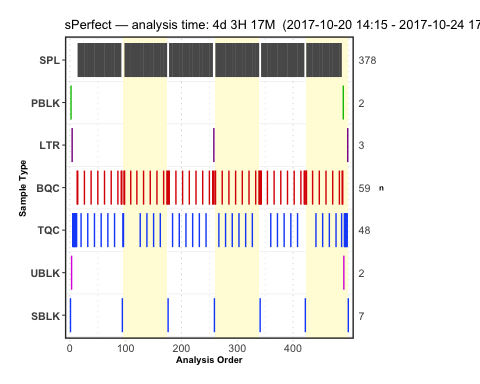
print(myexp@dataset\_orig) # TODO: Replace by actual getter

# A tibble: 250,997 × 19  
 analysis\_id data\_source raw\_data\_filename acquisition\_time\_stamp sample\_type  
 <chr> <chr> <chr> <dttm> <chr>   
 1 Longit\_BLAN… ./data/sPe… Longit\_BLANK-01 … 2017-10-20 14:15:36 SBLK   
 2 Longit\_B-IS… ./data/sPe… Longit\_B-ISTD 01… 2017-10-20 14:27:06 PBLK   
 3 Longit\_Un-I… ./data/sPe… Longit\_Un-ISTD 0… 2017-10-20 14:38:26 UBLK   
 4 Longit\_LTR … ./data/sPe… Longit\_LTR 01 2017-10-20 14:49:48 LTR   
 5 Longit\_TQC-… ./data/sPe… Longit\_TQC-10% 2017-10-20 15:12:31 TQC   
 6 Longit\_TQC-… ./data/sPe… Longit\_TQC-20% 2017-10-20 15:23:51 TQC   
 7 Longit\_TQC-… ./data/sPe… Longit\_TQC-40% 2017-10-20 15:35:11 TQC   
 8 Longit\_TQC-… ./data/sPe… Longit\_TQC-60% 2017-10-20 15:46:31 TQC   
 9 Longit\_TQC-… ./data/sPe… Longit\_TQC-80% 2017-10-20 15:57:51 TQC   
10 Longit\_TQC-… ./data/sPe… Longit\_TQC-100% 2017-10-20 16:09:11 TQC   
# ℹ 250,987 more rows  
# ℹ 14 more variables: batch\_id <chr>, feature\_id <chr>,  
# norm\_istd\_feature\_id <chr>, integration\_qualifier <lgl>, feature\_rt <dbl>,  
# feature\_area <dbl>, feature\_height <dbl>, feature\_fwhm <dbl>,  
# feature\_int\_start <dbl>, feature\_int\_end <dbl>, method\_polarity <chr>,  
# method\_precursor\_mz <dbl>, method\_product\_mz <dbl>,  
# method\_collision\_energy <dbl>

## 3 Analytical design and timeline

Having an overview of the analysis design and timelines can be helpful for the next processing steps. The plot below shows the batch structure, the QC samples included with their positions, and some information on date, duration and run time of the analysis in the title. Use show\_timestamp = TRUE to show the analysis timestamps in the x axis. This can be helpful to see if there were interruptions during the analysis and when the different batches were analysed.

qc\_plot\_runsequence(  
 myexp,   
 qc\_types = NA,   
 show\_batches = TRUE,   
 batches\_as\_shades = TRUE,   
 batch\_shading\_color = "#fffbdb",   
 segment\_thickness = 0.5,  
 show\_timestamp = FALSE)

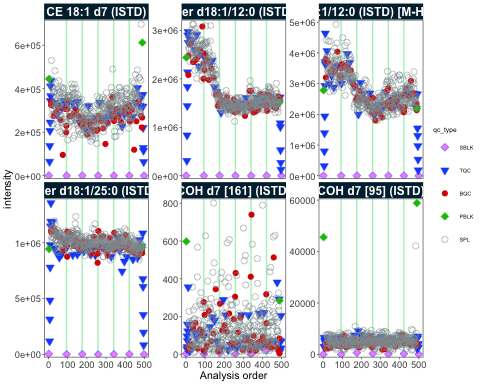


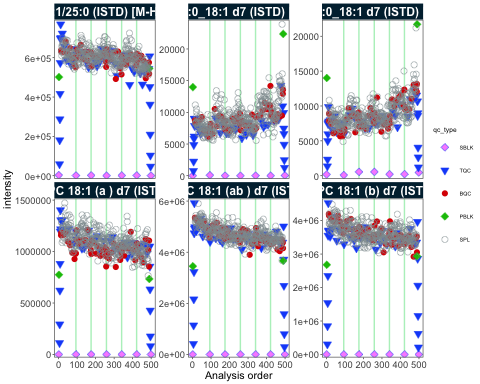
## 4 Signal trends of Internal Standards

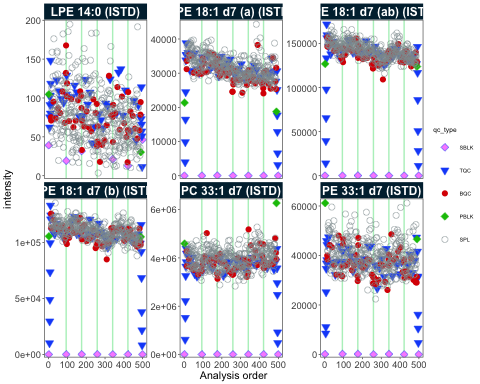
To have a first idea on how the analyses went, we can look at the Internal Standards in all samples across all 6 batches. Since the same ISTD amount was spiked into each sample (except SBLK) we should expect same intensities in all samples and sample types. What do you observe? You can set save\_pdf to TRUE to have a pdf from the plots (which will be in the output folder).

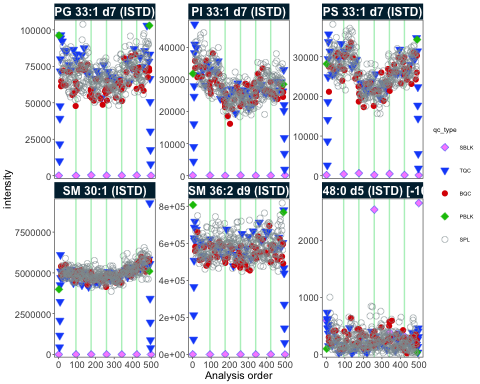
qc\_plot\_runscatter(  
 data = myexp,  
 variable = "intensity",  
 qc\_types = c("BQC", "TQC", "SPL", "PBLK", "SBLK"),  
 analysis\_no\_range = NA, #get\_batch\_boundaries(myexp, c(1,6)),   
 filt\_include\_features = "ISTD",   
 filt\_exclude\_features = "Hex|282",  
 cap\_outliers = FALSE,  
 log\_scale = FALSE,   
 show\_batches = TRUE,base\_font\_size = 5,  
 save\_pdf = FALSE,  
 path = "./output/runscatter\_istd.pdf",  
 cols\_page = 3, rows\_page = 2  
)

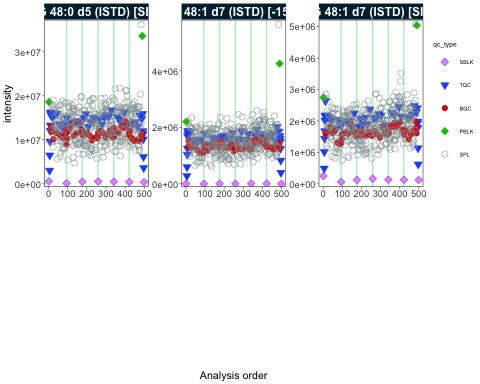
Generating plots (5 pages)...











- done!

## 5 Adding detailed metadata

For further processing we need additional metadata describing the samples and features. The MiDAR EXCEL template is an option to collect, organize and pre-validate analysis metadata. Open the XLSM file in the data folder to explore the metadata structure.

Using below function we import metadata from this template. In case of errors in the metadata (e.g. duplicate or missing ID), the import will fail with an error message. A summary of identified or potential issues in the metadata will be shown in a table. Work through these warnings to check your metadata, or ignore the warnings and proceed using ignore\_warnings = TRUE.

myexp <- midar::metadata\_import\_midarxlm(myexp, path = metadata\_path, ignore\_warnings = TRUE)

! Metadata has following warnings and notifications:

---------------------------------------------------------------------------------  
 Type Table Issue Column Count  
1 W Analyses Analyses not in analysis data NA 15  
2 W Features Feature(s) not in analysis data NA 3  
3 W ISTDs Internal standard(s) not used quant\_istd\_feature\_id 1  
---------------------------------------------------------------------------------  
E = Error, W = Warning, N = Note  
---------------------------------------------------------------------------------

✔ Analysis metadata associated with 499 analyses.

✔ Feature metadata associated with 503 features.

✔ Internal Standard metadata associated with 17 ISTDs.

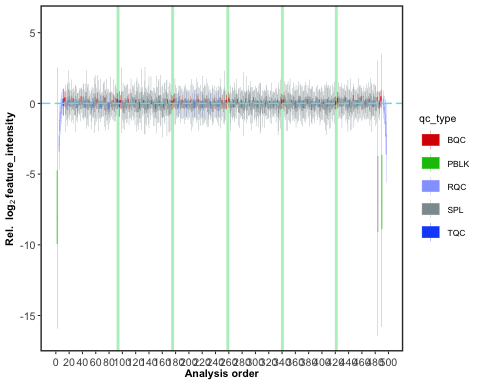
✔ Response Curve metadata associated with 12 analyses.

## 6 Overall trends and possible technical outlier

To examine overall technical trends and issues affecting the majority of analytes (features), the RLA (Relative Log Abundance) plot can useful (De Livera et al, Analytical Chemistry, 2015). In this plot all feature is normalized (by the median across or within-batch) and the plotted as a boxplot per sample. This plot can to identify potential pipetting errors, sample spillage, injection volume changes or overall instrument sensitivity changes.

First run the code below as it is. What do you observe? Then have a look at batch 6 only uncommenting the line #analysis\_no\_range.. What does this tell us? Identify the potential … by setting x\_axis\_variable = "analysis\_id". Set the y axis limits manually y\_lim = c(-2,2) and show all analyses to see if there are other trends or fluctuations.

midar::qc\_plot\_rla\_boxplot(  
 data = myexp,  
 rla\_type\_batch = c("within"),  
 variable = "intensity",  
 qc\_types = c("BQC", "SPL", "RQC", "TQC", "PBLK"),   
 filter\_data = FALSE,   
 #analysis\_no\_range = get\_batch\_boundaries(myexp, batch\_ids = c(6,6)),   
 #y\_lim = c(-3,3),  
 x\_axis\_variable = "run\_seq\_num",  
 ignore\_outliers = FALSE, x\_gridlines = FALSE,  
 batches\_as\_shades = FALSE,  
 linewidth = 0.1  
)

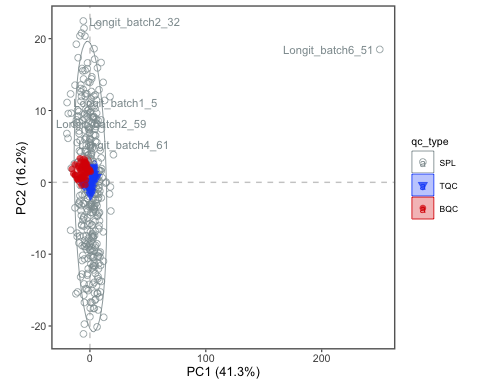


## 7 PCA plot of all QC types

This is plot is an other way to obtain an overview of the study and QC samples and to identify potential issues in the data.

Explore other PCA dimension and try adding blanks (i.e. PBLK) to the plot, by adding "PBLK" to qc\_types below.

qc\_plot\_pca(  
 data = myexp,   
 variable = "feature\_intensity",   
 filter\_data = FALSE,  
 pca\_dim = c(1,2),  
 label\_k\_mad = 3,   
 qc\_types = c("SPL", "BQC", "NIST", "TQC"),  
 log\_transform = TRUE,   
 point\_size = 2, point\_alpha = 0.7, font\_base\_size = 8, ellipse\_alpha = 0.3,   
 remove\_istds = TRUE)



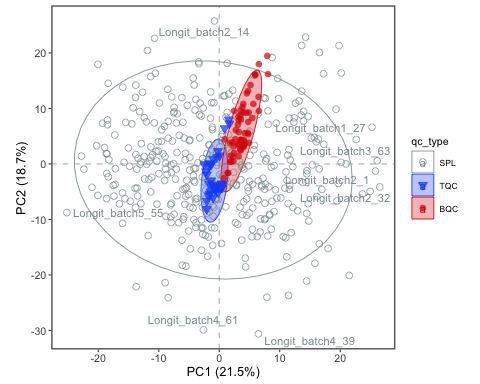
## 8 Exclude technical outliers

We classified a technical outlier and decide to remove it from all downstream processing, using data\_exclude\_analyses(). What do we now observe in the new PCA plot?

# Exclude the sample from the processing  
myexp <- data\_exclude\_analyses(myexp, analyses\_exlude = c("Longit\_batch6\_51"), overwrite = TRUE)

ℹ 1 analyses were excluded for downstream processing. Please reprocess data.

# Replot the PCA  
qc\_plot\_pca(  
 data = myexp,   
 variable = "feature\_intensity",   
 filter\_data = FALSE,  
 pca\_dim = c(1,2),  
 label\_k\_mad = 3,   
 qc\_types = c("SPL", "BQC", "NIST", "TQC"),  
 log\_transform = TRUE,   
 point\_size = 2, point\_alpha = 0.7, font\_base\_size = 8, ellipse\_alpha = 0.3,   
 remove\_istds = TRUE,   
 hide\_label\_text = NA)



## 9 Response curves

# Exclude very low abundant features  
myexp <- midar::qc\_apply\_feature\_filter(myexp, intensity.median.spl.min = 200)

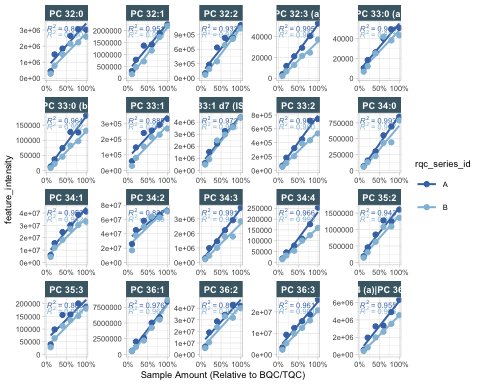
Calculating feature QC metrics - please wait...

✔   
New QC filter criteria were defined: 416 of 424 quantifier features meet QC criteria (excluding the 25 quantifier ISTD features).

qc\_plot\_responsecurves(  
 data = myexp,   
 variable = "intensity",   
 filter\_data = TRUE,   
 feature\_incl\_filt = "PC 3[2-9]",   
 save\_pdf = FALSE, cols\_page = 5,   
)

Generating plots (2 pages):

|   
 | | 0%



|   
 |========================= | 50%



|   
 |==================================================| 100% - done!

## 10

## 11 Isotope interference correction

As shown in the course presentation, we have a few cases where peaks were co-integrated with the interfering isotope peaks of other lipid species. We can subtract these intereferences from the raw intensities (areas) with below function, which uses information in the metadata.

myexp <- midar::correct\_interferences(myexp)

✔ Interference-correction has been applied to 11 of the 503 features.

## 12 Normalization and quantification based on ISTDs

myexp <- midar::calc\_normalize\_by\_istd(myexp)

✔ 461 features normalized with 17 ISTDs in 498 analyses.

myexp <- midar::calc\_quant\_by\_istd(myexp)

✔ 461 feature concentrations calculated based on 17 ISTDs and sample amounts of 498 analyses.

ℹ Concentrations are given in μmol/L.

## 13 Examine the effects of class-wide ISTD normalization

Class-specific ISTDs are common in lipidomics. However, normalization with internal standards can lead to artefacts and increase the data variability rather than reduce technical variability.

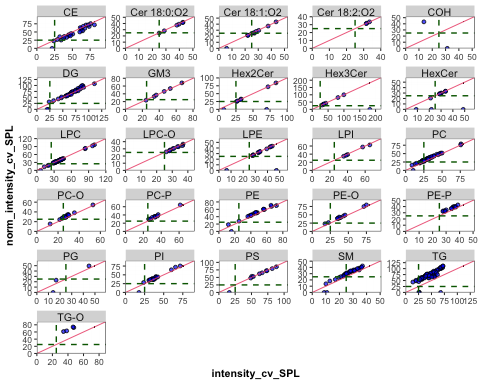
What do we observe below? What could be the reasons?

myexp <- midar::qc\_apply\_feature\_filter(myexp, intensity.median.spl.min = 1000, overwrite = FALSE)

Calculating feature QC metrics - please wait...

✔   
New QC filter criteria were defined: 394 of 424 quantifier features meet QC criteria (excluding the 25 quantifier ISTD features).

qc\_plot\_normalization\_cv(  
 data = myexp,   
 filter\_data = FALSE,   
 qc\_type = "SPL",   
 var\_before = "intensity",   
 var\_after = "norm\_intensity")



## 14 Drift correction

We will be using a Gaussian kernel smoothing based on the study sample to correct for within-batch drifts in the concentration data. The summary provided by this function is not meant to reflect real diagostics of the fit, but rather to understand if the fit caused any artefacts. Beside smoothing the trend, there is also an option to scale along the fit by setting scale\_smooth = TRUE.

myexp <- midar::correct\_drift\_gaussiankernel(  
 data = myexp,  
 qc\_types = c("SPL"),  
 batch\_wise = TRUE,  
 kernel\_size = 10,  
 outlier\_filter = TRUE,  
 outlier\_ksd = 5,  
 location\_smooth = TRUE,  
 scale\_smooth = TRUE,   
 show\_progress = TRUE # set to FALSE when rendering  
)

Applying drift correction...

|   
 | | 0%  
 |   
 |= | 2%  
 |   
 |== | 4%  
 |   
 |=== | 6%  
 |   
 |==== | 8%  
 |   
 |==== | 10%  
 |   
 |===== | 12%  
 |   
 |====== | 14%  
 |   
 |======= | 16%  
 |   
 |======== | 18%  
 |   
 |========= | 20%  
 |   
 |========== | 22%  
 |   
 |=========== | 24%  
 |   
 |============ | 26%  
 |   
 |============ | 28%  
 |   
 |============= | 30%  
 |   
 |============== | 32%  
 |   
 |=============== | 34%  
 |   
 |================ | 36%  
 |   
 |================= | 38%  
 |   
 |================== | 40%  
 |   
 |=================== | 43%  
 |   
 |==================== | 45%  
 |   
 |==================== | 47%  
 |   
 |===================== | 49%  
 |   
 |====================== | 51%  
 |   
 |======================= | 53%  
 |   
 |======================== | 55%  
 |   
 |========================= | 57%  
 |   
 |========================== | 59%  
 |   
 |=========================== | 61%  
 |   
 |============================ | 63%  
 |   
 |============================= | 65%  
 |   
 |============================= | 67%  
 |   
 |============================== | 69%  
 |   
 |=============================== | 71%  
 |   
 |================================ | 73%  
 |   
 |================================= | 75%  
 |   
 |================================== | 77%  
 |   
 |=================================== | 79%  
 |   
 |==================================== | 81%  
 |   
 |===================================== | 83%  
 |   
 |===================================== | 85%  
 |   
 |====================================== | 87%  
 |   
 |======================================= | 89%  
 |   
 |======================================== | 91%  
 |   
 |========================================= | 93%  
 |   
 |========================================== | 95%  
 |   
 |=========================================== | 97%  
 |   
 |============================================| 99%  
 |   
 |============================================| 100% - trend smoothing done!  
  
 |   
 | | 0%  
 |   
 |= | 2%  
 |   
 |== | 4%  
 |   
 |=== | 6%  
 |   
 |==== | 8%  
 |   
 |==== | 10%  
 |   
 |===== | 12%  
 |   
 |====== | 14%  
 |   
 |======= | 16%  
 |   
 |======== | 18%  
 |   
 |========= | 20%  
 |   
 |========== | 22%  
 |   
 |=========== | 24%  
 |   
 |============ | 26%  
 |   
 |============ | 28%  
 |   
 |============= | 30%  
 |   
 |============== | 32%  
 |   
 |=============== | 34%  
 |   
 |================ | 36%  
 |   
 |================= | 38%  
 |   
 |================== | 40%  
 |   
 |=================== | 43%  
 |   
 |==================== | 45%  
 |   
 |==================== | 47%  
 |   
 |===================== | 49%  
 |   
 |====================== | 51%  
 |   
 |======================= | 53%  
 |   
 |======================== | 55%  
 |   
 |========================= | 57%  
 |   
 |========================== | 59%  
 |   
 |=========================== | 61%  
 |   
 |============================ | 63%  
 |   
 |============================= | 65%  
 |   
 |============================= | 67%  
 |   
 |============================== | 69%  
 |   
 |=============================== | 71%  
 |   
 |================================ | 73%  
 |   
 |================================= | 75%  
 |   
 |================================== | 77%  
 |   
 |=================================== | 79%  
 |   
 |==================================== | 81%  
 |   
 |===================================== | 83%  
 |   
 |===================================== | 85%  
 |   
 |====================================== | 87%  
 |   
 |======================================= | 89%  
 |   
 |======================================== | 91%  
 |   
 |========================================= | 93%  
 |   
 |========================================== | 95%  
 |   
 |=========================================== | 97%  
 |   
 |============================================| 99%  
 |   
 |============================================| 100% - trend recalc done!

✔ Drift correction (batch-wise) was applied to raw concentrations of 461 of 461 features.

ℹ The median CV of all features in study samples (batche medians) decreased by -1.7% (-3.7 to 0.3%) to 42.2%.

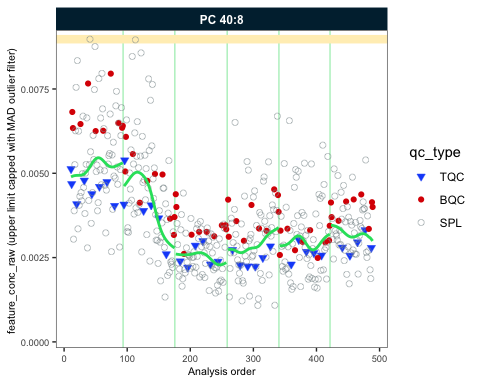
To illustrate the correction we will plot an example (PC 40:8) before after the drift and batch correction. Since we use the same plot several tomes, we first create a function wrapping the plot functionaly more easily.

# Define a wrapper function  
  
my\_trend\_plot <- function(variable, feature){  
 qc\_plot\_runscatter(  
 data = myexp,  
 variable = variable,  
 qc\_types = c("BQC", "TQC", "SPL"),  
 filt\_include\_features = feature,  
 filt\_exclude\_features = "ISTD",  
 cap\_outliers = TRUE,  
 log\_scale = FALSE,  
 show\_trend = TRUE,  
 save\_pdf = FALSE,  
 path = "./output/runscatter\_PC408\_beforecorr.pdf",  
 cols\_page = 1, rows\_page = 1,   
 )  
}

Let’s use this before defined function to plot the trends of one selected example before and after within-batch smoothing. What do we observe?

my\_trend\_plot("conc\_raw", "PC 40:8")

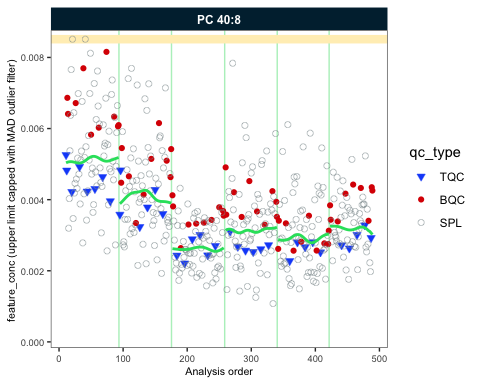
Generating plots (1 page)...



- done!

my\_trend\_plot("conc", "PC 40:8")

Generating plots (1 page)...



- done!

## 15 Batch-effect correction

As we observed the trend lines of the different batches are not aligned. We will use correct\_batcheffects() to correct for median center (location) and scale differences between the batches. The define that the correction should be based on the study samples medians. An optinal scale correction can be performed by setting correct\_scale = FALSE. After the correction we directly plot our example lipid species again. What do you observe now?

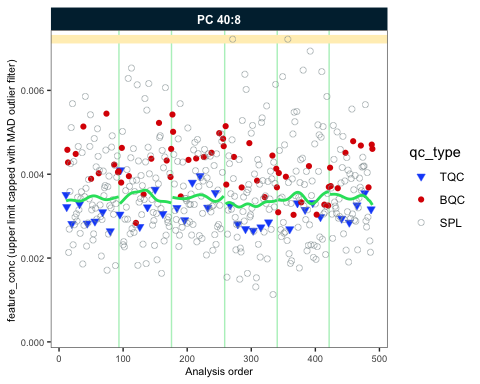
myexp <- midar::correct\_batcheffects(  
 myexp,   
 qc\_types = "SPL", overwrite = T,  
 correct\_location = TRUE,   
 correct\_scale = FALSE)

✔ Batch median-centering of 6 batches was applied to drift-corrected concentrations of all 461 features.

ℹ The median CV of features in the study samples across batches decreased by -0.6% (-2.2 to 1.0%) to 44.7%.

my\_trend\_plot("conc", "PC 40:8")

Generating plots (1 page)...



- done!

myexp <- midar::correct\_batcheffects(  
 myexp,   
 qc\_types = "SPL", overwrite = T,  
 correct\_location = TRUE,   
 correct\_scale = TRUE)

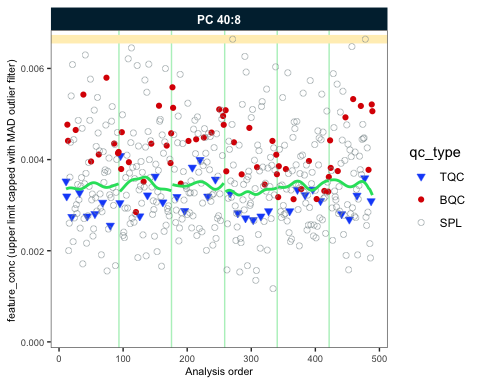
! Previous batch correction is being overwritten!

✔ Batch median-centering of 6 batches was applied to drift-corrected concentrations of all 461 features.

ℹ The median CV of features in the study samples across batches decreased by -0.5% (-2.7 to 1.7%) to 45.2%.

my\_trend\_plot("conc", "PC 40:8")

Generating plots (1 page)...



- done!

## 16 Saving *runscatter* plots of all features as PDF

For further inspection and documentation we can save plots for species of interest. We do not plot Blanks are they can easily have random concentrations when signals of feature and ISTDs are near or below LOD. The plots are saved in the output folder. Have a look and feel free to adjust how many plots are saved per page by setting cols\_page and rows\_page. Use ?runscatter or press F2 on the function name to see all available options.

qc\_plot\_runscatter(  
 data = myexp,  
 variable = "conc",  
 qc\_types = c("BQC", "TQC", "SPL"),  
 filt\_include\_features = NA,  
 filt\_exclude\_features = "ISTD",  
 cap\_outliers = TRUE,  
 log\_scale = FALSE,  
 show\_trend = TRUE,  
 save\_pdf = TRUE,  
 path = "./output/runscatter\_after-drift-batch-correction.pdf",  
 cols\_page = 2,   
 rows\_page = 2,  
 show\_progress = TRUE  
)

## 17 QC-based feature filtering

As the last step in the processing we now apply a set of filters to exclude features that do not meet the specific QC criteria. The function qc\_apply\_feature\_filter() allows to set the filters based on the data and metadata, press TAB after the open bracket after the function name of use F1 or ?qc\_apply\_feature\_filter to see available filter criteria. The function can be run multiple times to explore the effect of different filters. > Experiment with different filter set and filter criteria.

myexp <- qc\_apply\_feature\_filter(  
 data = myexp,  
 overwrite = TRUE,  
 batch\_medians = TRUE,  
 qualifier.include = FALSE,  
 istd.include = FALSE,  
 response.curve.id = 1,  
 response.rsquare.min = 0.8,  
 response.yintersect.rel.max = 0.6,  
 signalblank.median.pblk.min = 10,  
 intensity.median.spl.min = 100,  
 #dratio.conc.tqc.mad.max = 0.5,  
 cv.conc.bqc.max = 25,  
 features\_to\_keep = c("CE 20:4", "CE 22:5", "CE 22:6", "CE 16:0", "CE 18:0")  
)

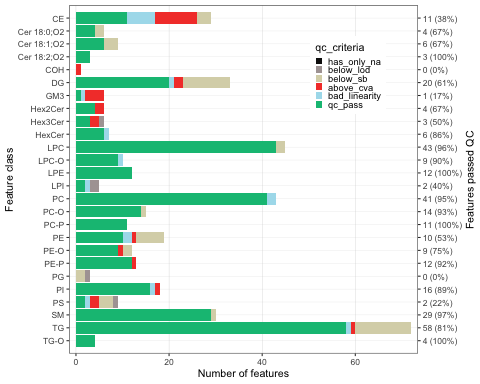
Calculating feature QC metrics - please wait...

✔   
New QC filter criteria were defined: 330 of 424 quantifier features meet QC criteria (excluding the 25 quantifier ISTD features).

## 18 Summary of the QC filtering

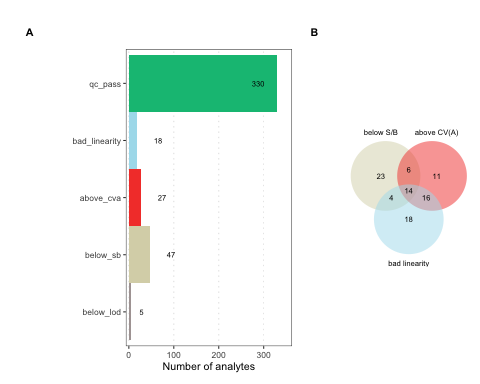
The plot belows provides an overview of the data quality and the feature filtering. The segments in green indicate the number of species that passed all previously defined qc filtering criteria. The rest are number of species that failed in the different filtering criteria. Note that criteria are hierarchically organized, i.e. feature is only classified as failing a criteria (e.g., CVA), when it has passed the hierarchically lower filters (e.g., S/B and LOD).

midar::qc\_plot\_summary\_classes(myexp, include\_qualifier = FALSE)



The following plot summarizes the feature filtering even more, depicting the total number of features that passed the filtering. Also here the classification is based on hierarchical application of filters, as mentioned before. The Venn diagram on the right shows the number of features that failed a particular filtering criteria.

midar::qc\_plot\_summary(myexp, include\_qualifier = FALSE)



## 19 Saving a report with data, metadata and processing details

With the function below we can save an EXCEL workbook with different sheets comprising raw and processed datasets, metadata and feature QC metrics. The report can be found in the output folder.

midar::report\_write\_xlsx(myexp, path = "./output/myexp-midar\_report.xlsx")

Saving report to disk - please wait...

✔   
The data processing report of analysis 'sPerfect' has been saved.

Additionally, data can be saved as flat wide CSV file. Check the function arguments to select and subset the data.

midar::report\_write\_csv(  
 data = myexp,   
 path = "./output/sperfect\_filt\_uM.csv",  
 variable = "conc",   
 qc\_types = "SPL",   
 include\_qualifier = FALSE,  
 filter\_data = TRUE)

✔ Concentration values of 376 analyses and 330 features have been exported.

## 20 Sharing the MidarExperiment dataset

The myexp object can be saved as an RDS file and shared. RDS files serialized R variable/objects that can be opened in R even if midar is not installed. You use the imported MidarExperiment object for re-processing, plotting or inspection using the midar package. > Below we save our experiment to the disk and re-open it again under a different name and check the status. **TODO: use midar helper function to load the RDS file, asserting that the class is a MidarExperiment object.**

saveRDS(myexp, file = "./output/myexp-midar.rds", compress = TRUE)  
my\_saved\_exp <- readRDS(file = "./output/myexp-midar.rds")  
print(myexp)

── MidarExperiment ─────────────────────────────────────────────────────────────

Title: sPerfect

Processing status: Batch- and drift-corrected concentrations

── Annotated Raw Data ──

• Samples: 498

• Features: 503

── Metadata ──

• Sample annotation: ✔

• Feature annotation: ✔

• Internal standard annotation: ✔

• Response curve annotation: ✔

• Study samples annotation: ✖

── Processing Status ──

• Isotope corrected: ✔

• ISTD normalized: ✔

• ISTD quantitated: ✔

• Drift corrected: ✔

• Batch corrected: ✔

• Feature QC-filtered: ✔

── Exclusion of Analyses and Features ──

• One or more analyses manually excluded: ✔

## 21 More QC

Code for additional QC analyses will be added here before the start of the workshop.