Quality control–based robust LOESS signal correction (QC-RLSC)

Wanchang Lin

2024-07-24

Contents

0.1 Read data

Select file for signal correction

```
## FILE <- "data_qcrlsc_b4"
FILE <- "data_qcmxp_b4_tidy"
PATH <- here::here("data", paste0(FILE, ".xlsx"))
```

Load into R

```
xls  <- PATH %>%
  excel_sheets() %>%
  set_names() %>%
  map(read_excel, path = PATH)
```

Check the data

```
names(xls)
#> [1] "data" "meta" "peak"
t(sapply(xls, dim))
#>      [,1] [,2]
#> data  462 2106
#> meta  462    4
#> peak 2106    2
```

Get meta and data matrix

```
meta <- xls$meta
data <- xls$data %>%
  mutate_if(is.character, as.numeric)
peak <- xls$peak
```

Extract group information of batch and sample types

```
names(meta)
#> [1] "SampleID"   "SampleType" "Order"      "Batch"
## (cls.bl <- factor(meta$batch))
## (cls.bl <- factor(meta$batch))
(cls.qc <- factor(meta$SampleType))
#>   [1] QC     QC     QC     QC     QC     QC     QC     QC     QC     QC
#>  [11] Sample Sample Sample Sample Sample QC     Sample Sample Sample Sample
#>  [21] Sample QC     Sample Sample Sample Sample Sample QC     Sample Sample
#>  [31] Sample Sample Sample QC     Sample Sample Sample Sample Sample QC
#>  [41] Sample Sample Sample Sample Sample QC     Sample Sample Sample Sample
#>  [51] Sample QC     Sample Sample Sample Sample Sample QC     Sample Sample
#>  [61] Sample Sample Sample QC     Sample Sample Sample Sample Sample QC
#>  [71] Sample Sample Sample Sample Sample QC     Sample Sample Sample Sample
```

```
#>  [81] Sample QC     Sample Sample Sample Sample Sample QC     Sample Sample
#>  [91] Sample Sample Sample QC     Sample Sample Sample Sample Sample QC
#> [101] Sample Sample Sample Sample Sample QC     Sample Sample Sample Sample
#> [111] Sample QC     Sample Sample Sample Sample Sample QC     QC     QC
#> [121] QC     QC     QC     QC     Sample Sample Sample Sample Sample QC
#> [131] Sample Sample Sample Sample Sample QC     Sample Sample Sample Sample
#> [141] Sample QC     Sample Sample Sample Sample Sample QC     Sample Sample
#> [151] Sample Sample Sample QC     Sample Sample Sample Sample Sample QC
#> [161] Sample Sample Sample Sample Sample QC     Sample Sample Sample Sample
#> [171] Sample QC     Sample Sample Sample Sample Sample QC     Sample Sample
#> [181] Sample Sample Sample QC     Sample Sample Sample Sample Sample QC
#> [191] Sample Sample Sample Sample Sample QC     Sample Sample Sample Sample
#> [201] Sample QC     Sample Sample Sample Sample Sample QC     Sample Sample
#> [211] Sample Sample Sample QC     Sample Sample Sample Sample Sample QC
#> [221] Sample Sample Sample Sample Sample QC     Sample Sample Sample Sample
#> [231] Sample QC     QC     QC     QC     QC     QC     QC     QC     QC
#> [241] QC     QC     QC     Sample Sample Sample Sample Sample QC     Sample
#> [251] Sample Sample Sample Sample QC     Sample Sample Sample Sample Sample
#> [261] QC     Sample Sample Sample Sample Sample QC     Sample Sample Sample
#> [271] Sample Sample QC     Sample Sample Sample Sample Sample QC     Sample
#> [281] Sample Sample Sample Sample QC     Sample Sample Sample Sample Sample
#> [291] QC     Sample Sample Sample Sample Sample QC     Sample Sample Sample
#> [301] Sample Sample QC     Sample Sample Sample Sample Sample QC     Sample
#> [311] Sample Sample Sample Sample QC     Sample Sample Sample Sample Sample
#> [321] QC     Sample Sample Sample Sample Sample QC     Sample Sample Sample
#> [331] Sample Sample QC     Sample Sample Sample Sample Sample QC     Sample
#> [341] Sample Sample Sample Sample QC     Sample Sample Sample Sample Sample
#> [351] QC     QC     QC     QC     QC     QC     QC     QC     QC     QC
#> [361] QC     QC     Sample Sample Sample Sample Sample QC     Sample Sample
#> [371] Sample Sample Sample QC     Sample Sample Sample Sample Sample QC
#> [381] Sample Sample Sample Sample Sample QC     Sample Sample Sample Sample
#> [391] Sample QC     Sample Sample Sample Sample Sample QC     Sample Sample
#> [401] Sample Sample Sample QC     Sample Sample Sample Sample Sample QC
#> [411] Sample Sample Sample Sample Sample QC     Sample Sample Sample Sample
#> [421] Sample QC     Sample Sample Sample Sample Sample QC     Sample Sample
#> [431] Sample Sample Sample QC     Sample Sample Sample Sample Sample QC
#> [441] Sample Sample Sample Sample Sample QC     Sample Sample Sample Sample
#> [451] Sample QC     Sample Sample Sample Sample Sample QC     Sample Sample
#> [461] QC     QC
#> Levels: QC Sample
(cls.bl <- factor(meta$Batch))
#>   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
#>  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
#>  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
#> [112] 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

```
#> [149] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
#> [186] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
#> [223] 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
#> [260] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
#> [297] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
#> [334] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
#> [371] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
#> [408] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
#> [445] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
#> Levels: 1 2 3 4
```

0.2 Missing value filter and fill

Check missing value rates

```
tail(sort(mv_perc(data)), 20)
#> V1575 V1100  V198  V791  V925 V1412  V881 V1229  V284   V62 V1287 V1303  V937
#> 0.569 0.574 0.578 0.584 0.584 0.584 0.587 0.591 0.593 0.595 0.595 0.600 0.643
#> V1785  V905  V892 V1106 V1624  V676  V978
#> 0.654 0.662 0.669 0.669 0.669 0.677 0.703
```

Filter based on missing values

```
filter_qc <- FALSE    # filter on qc mising values or all missing values
thres <- 0.2           # threshold for filtering

if (filter_qc) {      # filter using all missing values
  ret <- mv_filter(data, thres = thres)
} else {              # filter using qc missing values
  ret <- mv_filter_qc(data, cls.qc, thres = thres)
}
```

Update data matrix and peak

```
dat <- ret$dat
pek <- peak[ret$idx,]
```

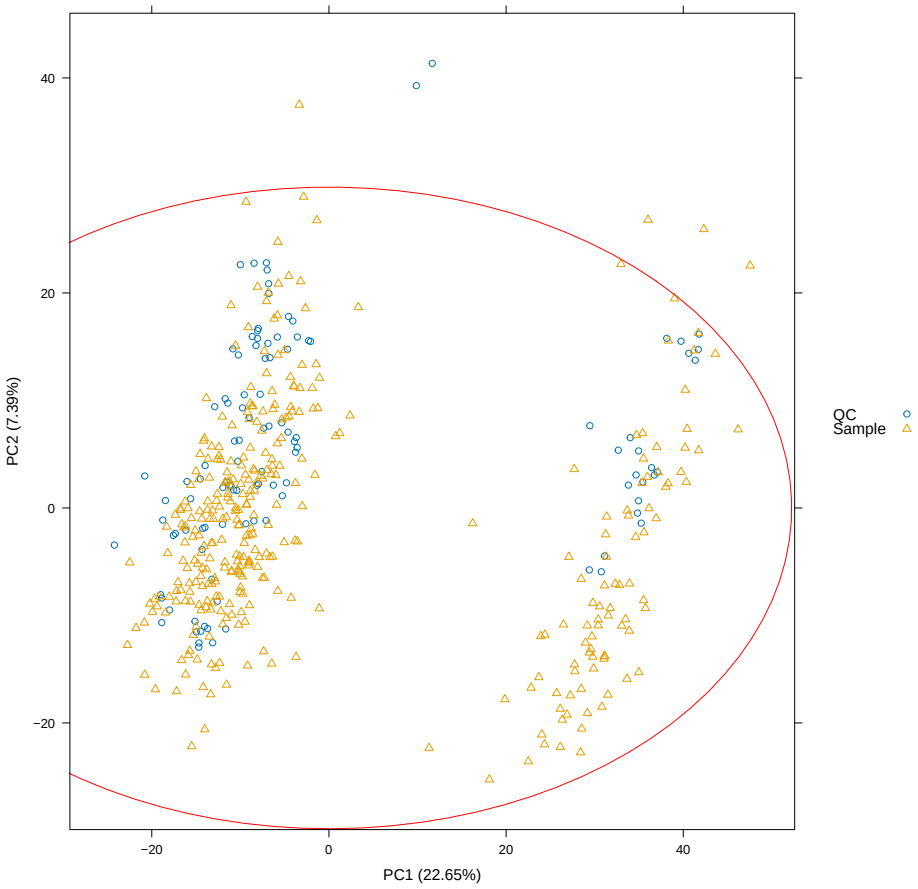
Missing values filling for visulisation

```
dat_fill  <- dat %>% mv.fill(method = "median", ze_ne = T) %>% as_tibble()
```

Data screening before signal correction PCA plot for sample types

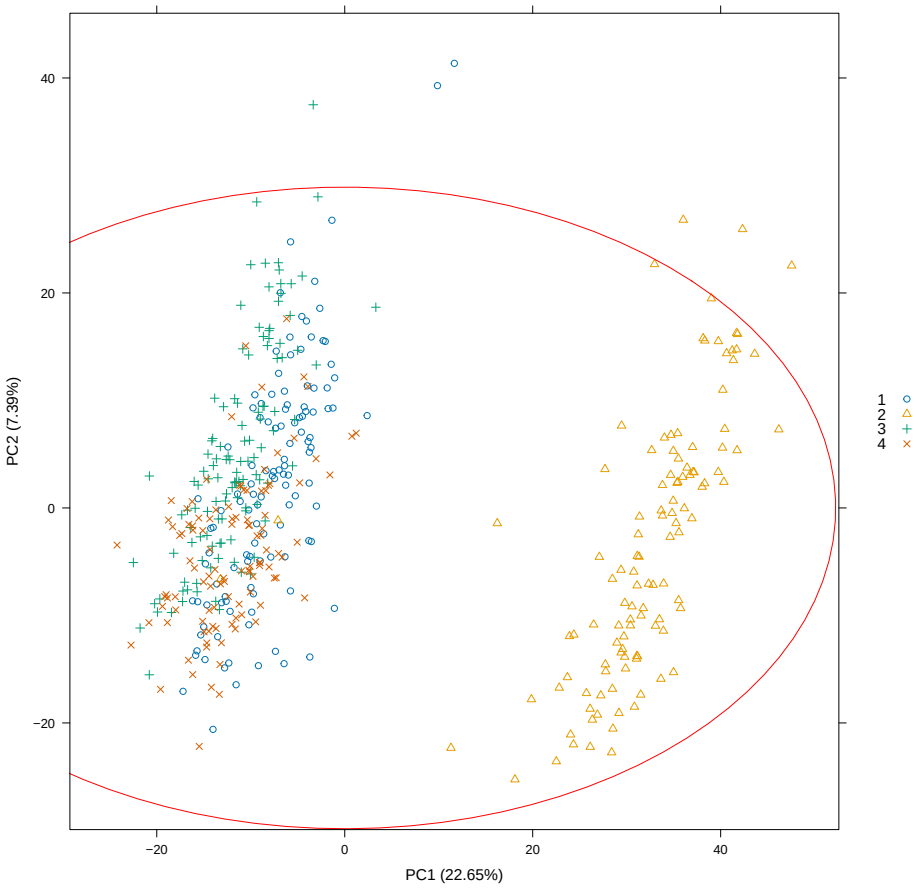
```
pcaplot(dat_fill, cls.qc, pcs = c(2, 1), ep = 1)
```

Quality control–based robust LOESS signal correction (QC-RLSC)



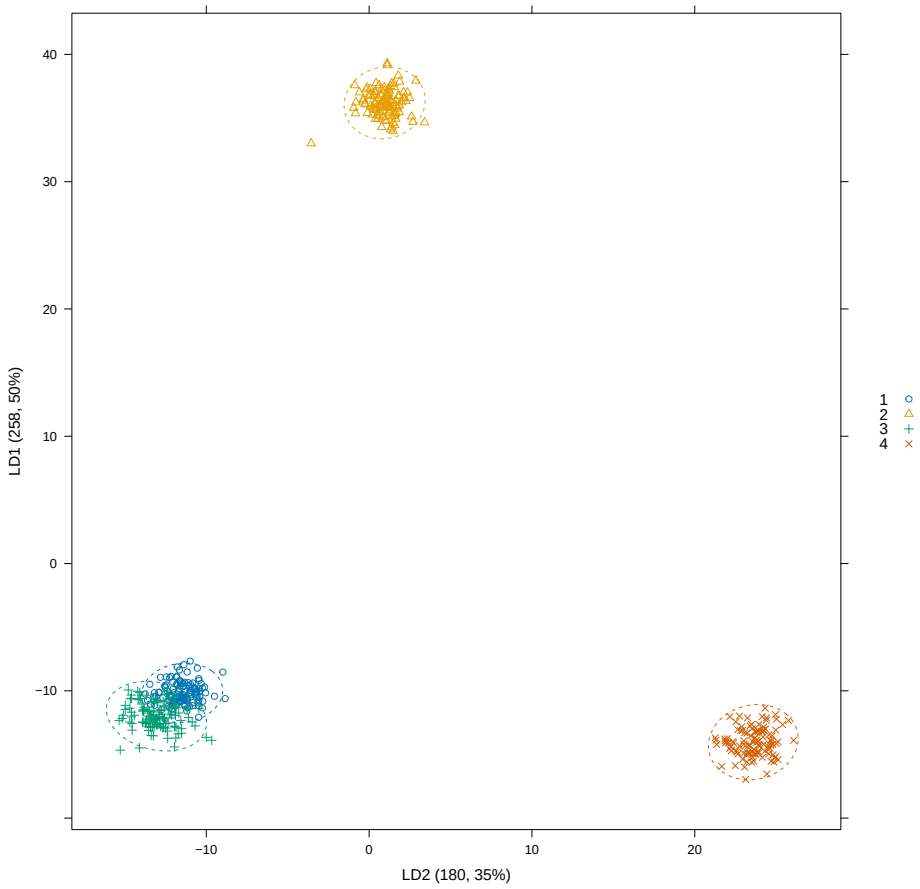
PCA plot for batches

```
pcaplot(dat_fill, cls.bl, pcs = c(2, 1), ep = 1)
```



LDA plot for batches

```
plot(pcalda(dat_fill, cls.bl), dimen = c(1:2), ep = 2)
```



0.3 Set parameters for QC-RLSC

```
method <- "subtract"  # two methods: "subtract", "divide"
intra <- F            # signal correction within batch or not
opti <- T             # optimise smooth parameter  or not
log10 <- T            # log 10 transform data or not
outl <- T             # outliter detect in qc samples or not
shift <- T            # batch shift or not
```

0.4 QC outlier detectio

log transformation

```
if (log10) {
  dat <- log10(dat)
}
```

outlier detection based on QC

```
  if (outl) {
    dat <- sapply(dat, function(x){ #' x <- dat[, 6, drop = T]
      qc_ind <- grepl("qc", cls.qc, ignore.case =  TRUE, perl = TRUE)
      ## get median of qc data
      qc_dat <- x[qc_ind]
      qc_median <- median(qc_dat, na.rm = TRUE)
      ## assign other data as NA for QC outlier detection
      tmp <- x
      tmp[!qc_ind] <- NA
      ## QC outlier detection
      out_ind <- outl_det_u(tmp)
      ## asisgn outlier as qc median
      x[out_ind] <- qc_median
      return(x)
    }) %>% as_tibble()
  }
  dat
  #> # A tibble: 462 x 1,633
  #>       V2    V3    V4    V5    V6    V7    V8    V9   V10   V11   V13   V17   V18
  #>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
  #>  1 NA     5.94 NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
  #>  2  4.86  6.08  5.47  4.61  4.89  7.34  5.23  4.60  7.15  4.91  4.82  4.38  5.74
  #>  3  4.90  5.96  5.46  4.71  4.89  7.31  5.22  4.61  7.14  4.87  4.82 NA     5.70
  #>  4  4.92  6.04  5.44  4.66  4.87  7.33  5.18 NA     7.15  4.95  4.78  4.58  5.69
  #>  5  4.95  5.99  5.47  4.60  4.86  7.30  5.19  4.48  7.13  4.90  4.57  4.64  5.68
  #>  6  5.00  6.04  5.43  4.70  4.86  7.31  5.20  4.59  7.15  4.85  4.66  4.69  5.67
  #>  7  4.98  6.05  5.46  4.72  4.92  7.33  5.26  4.64  7.17  4.91  4.68  4.67  5.68
  #>  8  4.98  5.95  5.47  4.70  4.85  7.30  5.18  4.54  7.14  4.88  4.58  4.78  5.65
  #>  9  4.99  5.97  5.45  4.66  4.87  7.29  5.13  4.57  7.14  4.86  4.87  4.59  5.63
  #> 10  4.98  6.02  5.47  4.66  4.87  7.31  5.21  4.59  7.15  4.89  4.91  4.58  5.61
  #> # i 452 more rows
  #> # i 1,620 more variables: V19 <dbl>, V20 <dbl>, V21 <dbl>, V22 <dbl>,
  #> #   V23 <dbl>, V24 <dbl>, V25 <dbl>, V26 <dbl>, V28 <dbl>, V29 <dbl>,
  #> #   V31 <dbl>, V33 <dbl>, V34 <dbl>, V35 <dbl>, V37 <dbl>, V38 <dbl>,
  #> #   V39 <dbl>, V40 <dbl>, V42 <dbl>, V43 <dbl>, V44 <dbl>, V45 <dbl>,
  #> #   V46 <dbl>, V47 <dbl>, V48 <dbl>, V50 <dbl>, V51 <dbl>, V52 <dbl>,
  #> #   V53 <dbl>, V54 <dbl>, V55 <dbl>, V56 <dbl>, V57 <dbl>, V58 <dbl>, ...
```

0.5 QC-RLSC

perform qc-rlsc within each batch or not

```
  tic()
  if (!intra) {
```

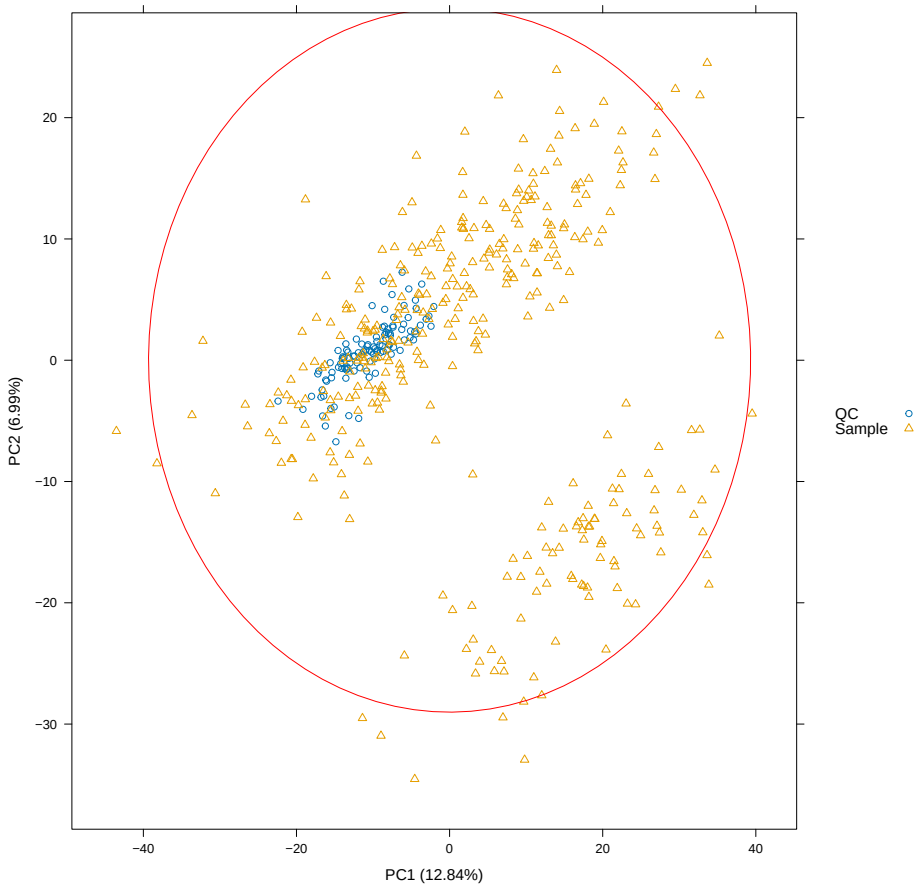
```
  res <- qc_rlsc(dat, cls.qc, method = method, opti = opti)
} else { # do signal correction inside each batch
  res <- lapply(levels(cls.bl), function(x){
    idx <- cls.bl %in% x
    tmp <- qc_rlsc(dat[idx,], cls.qc[idx], method = method, opti = opti)
  })
  res <- bind_rows(res)
}
toc()
#> 30.505 sec elapsed
```

Data visualisation after signal correction

```
res_fill  <- res %>% mv.fill(method = "median", ze_ne = T) %>% as_tibble()
```

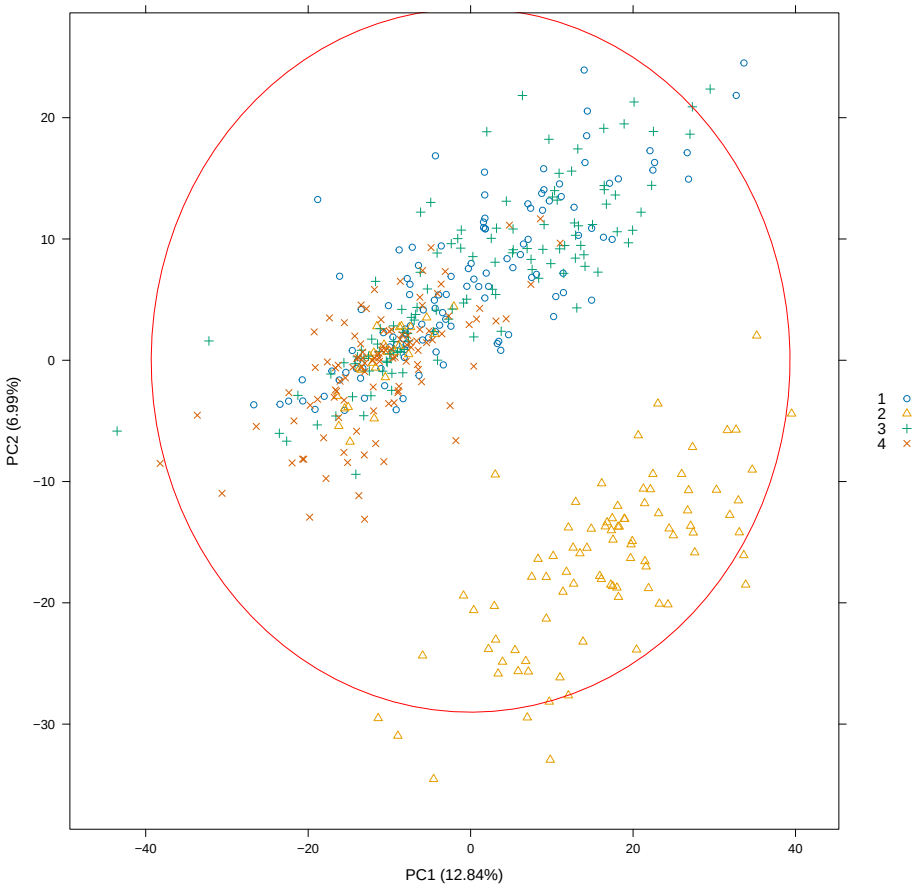
PCA plot for sample types

```
pcaplot(res_fill, cls.qc, pcs = c(2, 1), ep = 1)
```



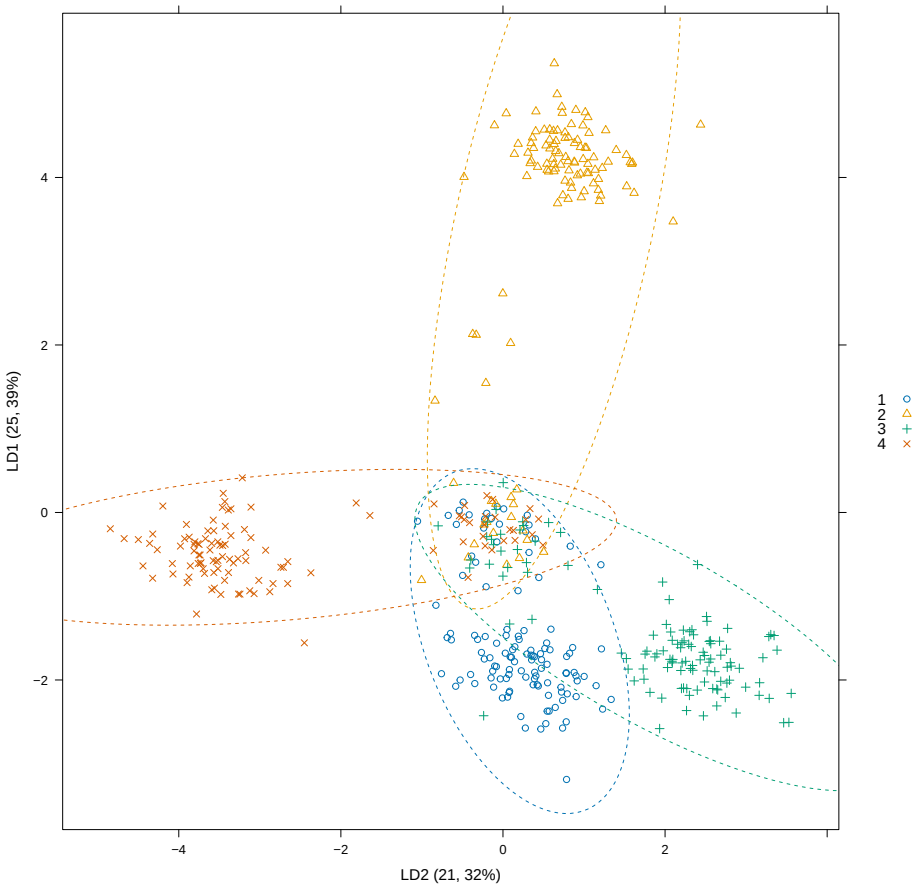
PCA plot for batches

```
pcaplot(res_fill, cls.bl, pcs = c(2, 1), ep = 1)
```



LDA plot for batches

```
plot(pcalda(res_fill, cls.bl), dimen = c(1:2), ep = 2)
```



0.6 Batch shift

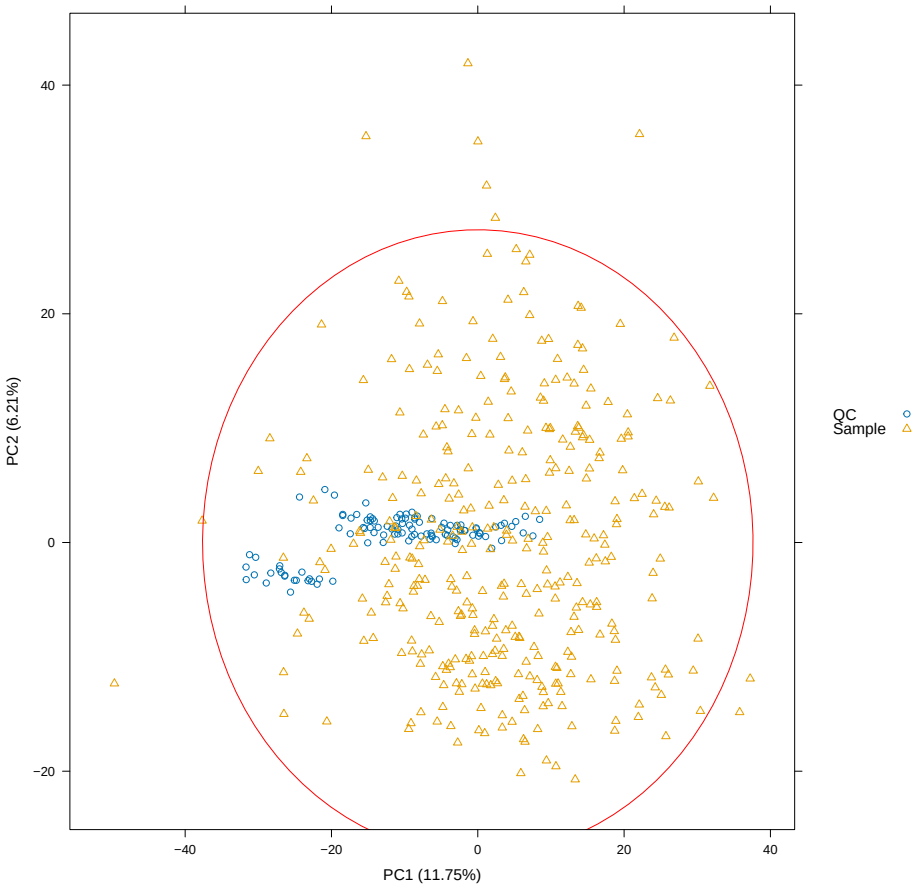
```
if (shift) {
  res <- batch_shift(res, cls.bl, overall_average = T) %>% as_tibble()
}
```

Data visualisation after batch shift

```
res_fill  <- res %>% mv.fill(method = "median", ze_ne = T) %>% as_tibble()
```

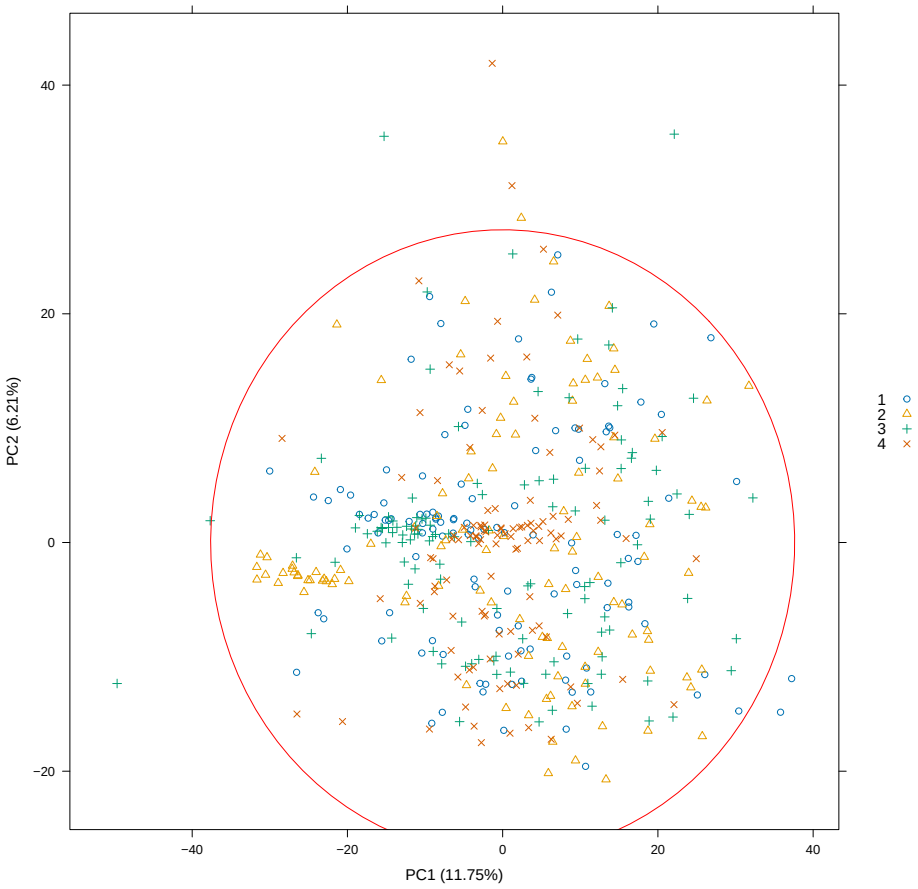
PCA plot for sample types

```
pcaplot(res_fill, cls.qc, pcs = c(2, 1), ep = 1)
```



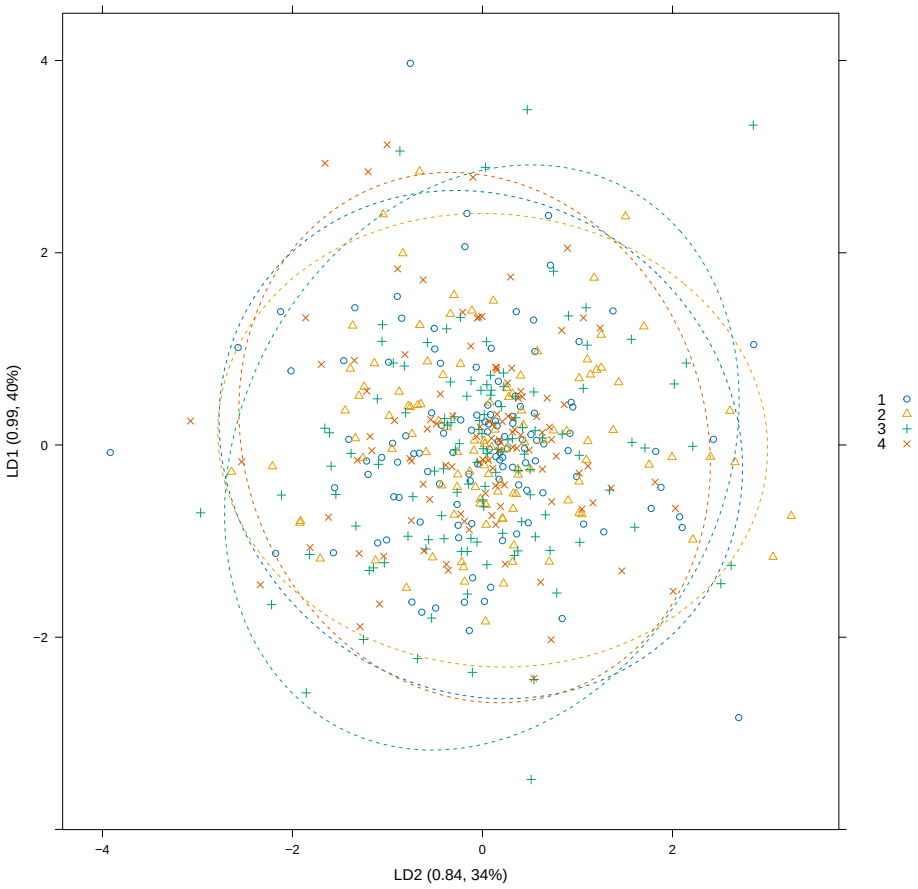
PCA plot for batches

```
pcaplot(res_fill, cls.bl, pcs = c(2, 1), ep = 1)
```



LDA plot for batches

```
plot(pcalda(res_fill, cls.bl), dimen = c(1:2), ep = 2)
```



0.7 Save results

inverse log10 transformation

```
res <- 10^res %>% as_tibble()

## tmp <- list(data =  res, meta = meta)
tmp <- list(data =  res, meta = meta, peak = pek)

## write.xlsx(tmp, file = here::here("data", paste0(FILE, "_res.xlsx")),
##            asTable = F, overwrite = T, rowNames = F, colNames = T)
```