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

Wanchang Lin

2024-08-01

Contents

0.1 Read data

Select file for signal correction

```
PATH <- here::here("extdata", "data_qcrlsc.xlsx")
```

Load into R

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

Check the data

```
names(xls)
#> [1] "meta" "data"
t(sapply(xls, dim))
#>      [,1] [,2]
#> meta  462    2
#> data  462  656
```

Get meta and data matrix

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

Extract group information of batch and sample types

```
names(meta)
#> [1] "block" "type"
(cls.qc <- factor(meta$type))
#>   [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$block))
#>   [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
#> [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
#> [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
#> [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
#> [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
#> [334] 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
```

```
#> [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 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 4 4
#> [445] 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

Let zero as NA before missing value process

```
data[data == 0] <- NA
```

Check missing value rates

```
tail(sort(mv_perc(data)), 20)
#> V1986  V562 V2098 V1602  V348 V1902  V975 V2017 V2020  V163 V1021 V1676 V1540
#> 0.156 0.158 0.160 0.162 0.165 0.167 0.169 0.169 0.169 0.171 0.171 0.171 0.173
#> V1321 V1935 V1079  V610 V1691 V2077  V926
#> 0.182 0.182 0.190 0.197 0.197 0.197 0.199
```

Filter based on missing values

```
filter_qc <- FALSE    # filter on qc missing 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

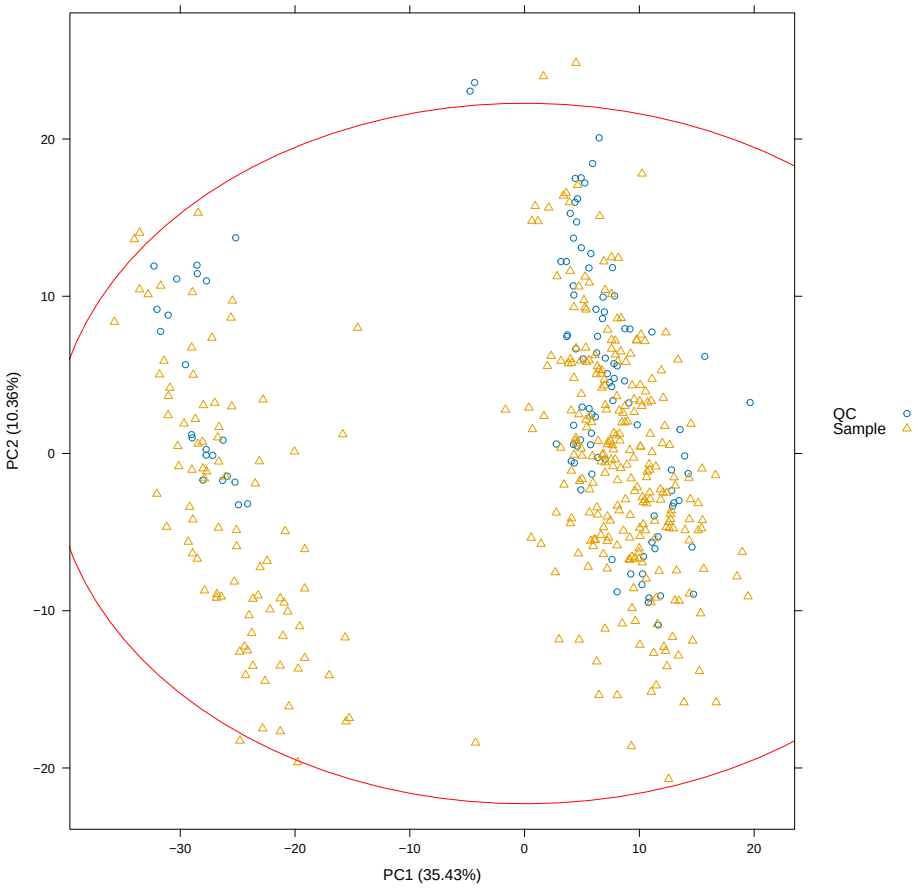
```
dat <- ret$dat
```

Missing values filling for visualisation

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

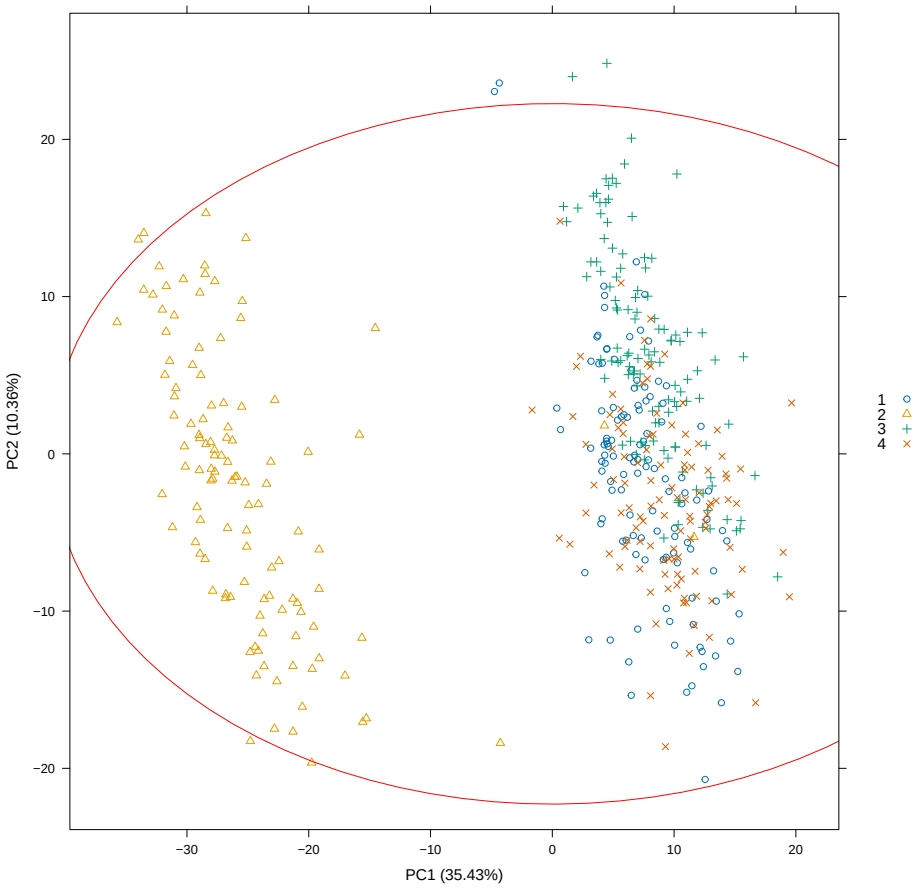
Data screening before signal correction PCA plot for sample types

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



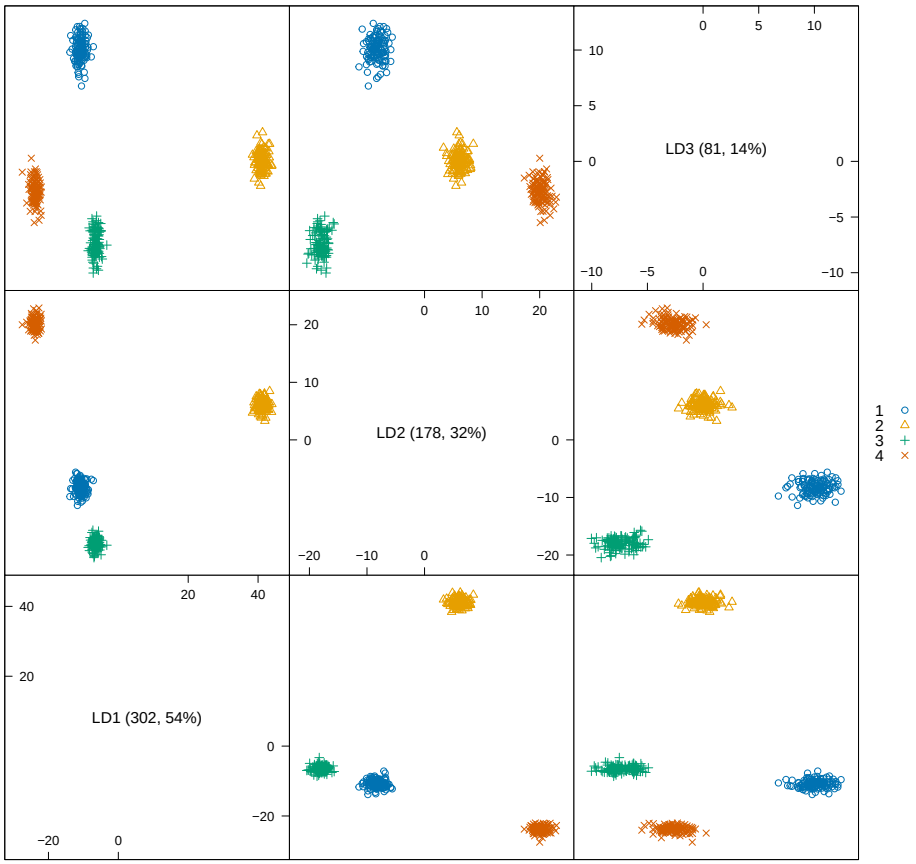
PCA plot for batches

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



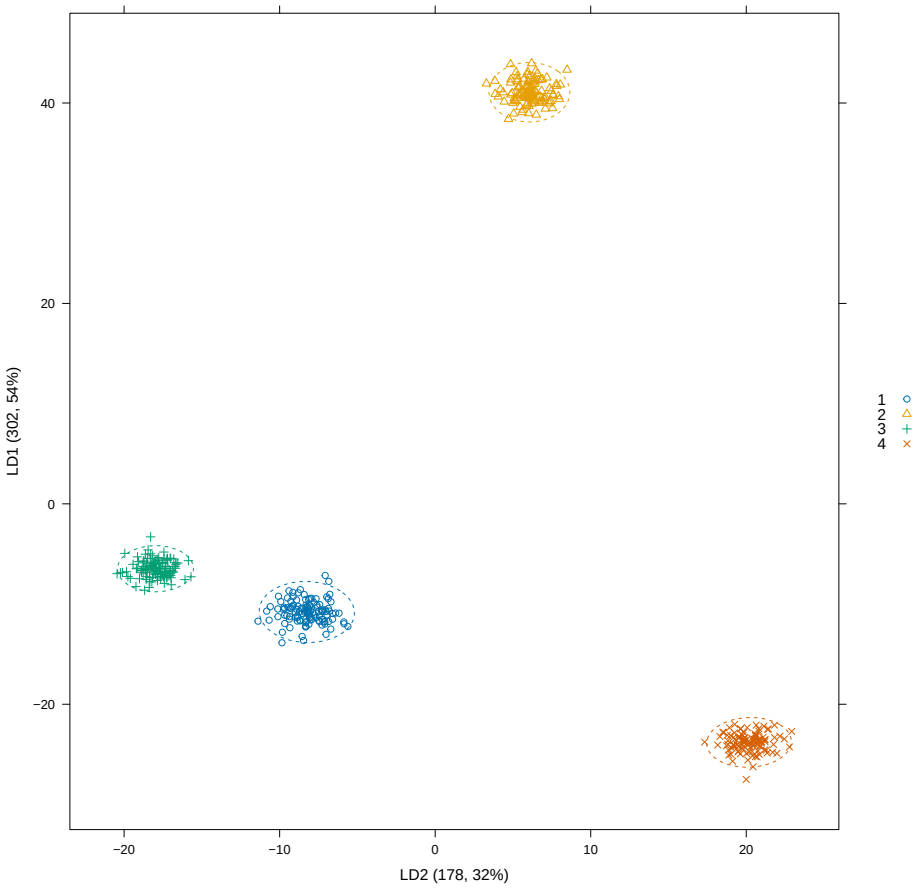
LDA plot for batches

```
plot(pcalda(dat_fill, cls.bl))
```



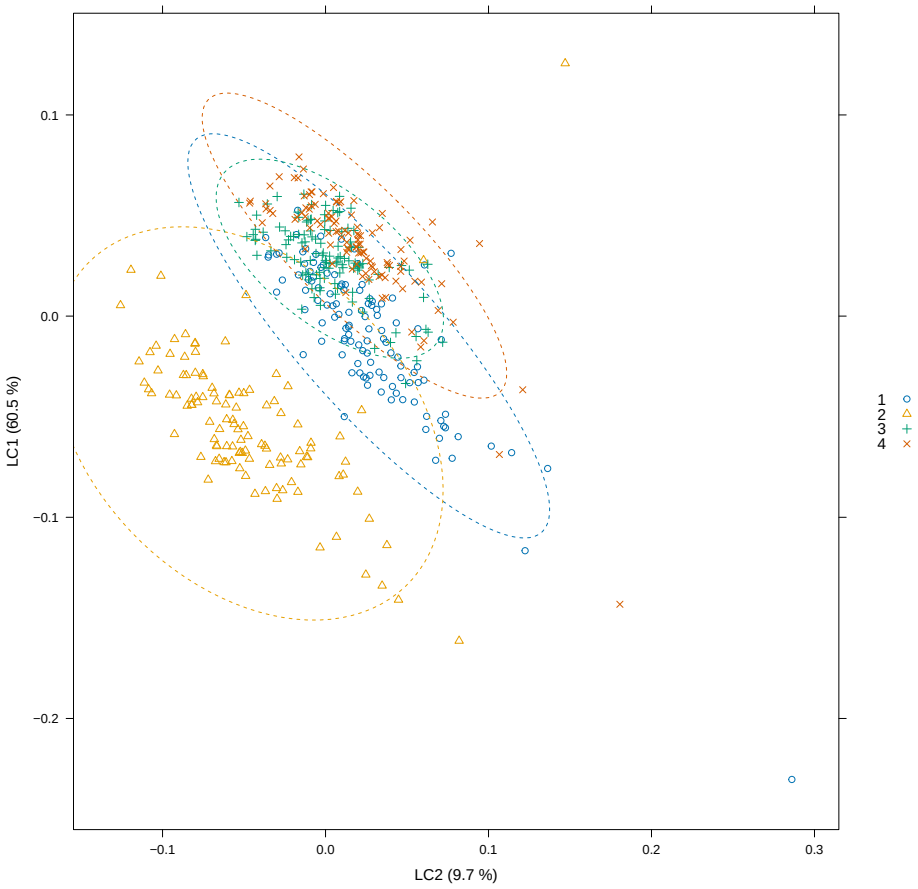
LDA plot of batches: LD1 vs LD2 (only for batch groups larger than 2)

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



PLS plot of batches: LC1 vs LC2

```
plot(plslda(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             # outlier detect in qc samples or not
shift <- T            # batch shift or not
```

0.4 QC outlier detection

log transformation

```
if (log10) {
  dat[dat == 0] <- NA
  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)
      ## assign outlier as qc median
      x[out_ind] <- qc_median
      return(x)
    }) %>% as_tibble()
  }
  dat
  #> # A tibble: 462 x 649
  #>       V3   V13   V17   V18   V19   V20   V22   V23   V25   V26   V31   V33   V34
  #>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
  #>  1  5.94 NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
  #>  2  6.08  4.82  4.38  5.74  8.18  5.66  4.97  7.46  4.74  7.28  4.88  6.59  6.88
  #>  3  5.96  4.82 NA     5.70  8.15  5.62  4.96  7.47 NA     7.28  4.87  6.64  6.90
  #>  4  6.04  4.78  4.58  5.69  8.12  5.60  4.98  7.48  4.75  7.29  4.89  6.61  6.91
  #>  5  5.99  4.57  4.64  5.68  8.14  5.61  4.92  7.45  4.71  7.27  4.72  6.57  6.91
  #>  6  6.04  4.66  4.69  5.67  8.13  5.60  4.95  7.47  4.74  7.28  4.75  6.56  6.92
  #>  7  6.05  4.68  4.67  5.68  8.13  5.60  4.99  7.47  4.75  7.28  4.78  6.56  6.91
  #>  8  5.95  4.58  4.78  5.65  8.09  5.57  4.96  7.46  4.73  7.28  4.71  6.56  6.93
  #>  9  5.97  4.87  4.59  5.63  8.10  5.57  4.95  7.47 NA     7.28  4.93  6.61  6.90
  #> 10  6.02  4.91  4.58  5.61  8.08  5.54  4.98  7.47  4.73  7.29  4.97  6.65  6.91
  #> # i 452 more rows
  #> # i 636 more variables: V38 <dbl>, V39 <dbl>, V45 <dbl>, V46 <dbl>, V47 <dbl>,
  #> #   V48 <dbl>, V50 <dbl>, V51 <dbl>, V66 <dbl>, V68 <dbl>, V71 <dbl>,
  #> #   V72 <dbl>, V73 <dbl>, V74 <dbl>, V104 <dbl>, V106 <dbl>, V112 <dbl>,
  #> #   V115 <dbl>, V116 <dbl>, V121 <dbl>, V122 <dbl>, V123 <dbl>, V124 <dbl>,
  #> #   V125 <dbl>, V126 <dbl>, V128 <dbl>, V134 <dbl>, V138 <dbl>, V140 <dbl>,
  #> #   V142 <dbl>, V147 <dbl>, V149 <dbl>, V157 <dbl>, V158 <dbl>, V159 <dbl>, ...
```

0.5 QC-RLSC

perform qc-rlsc within each batch or not

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

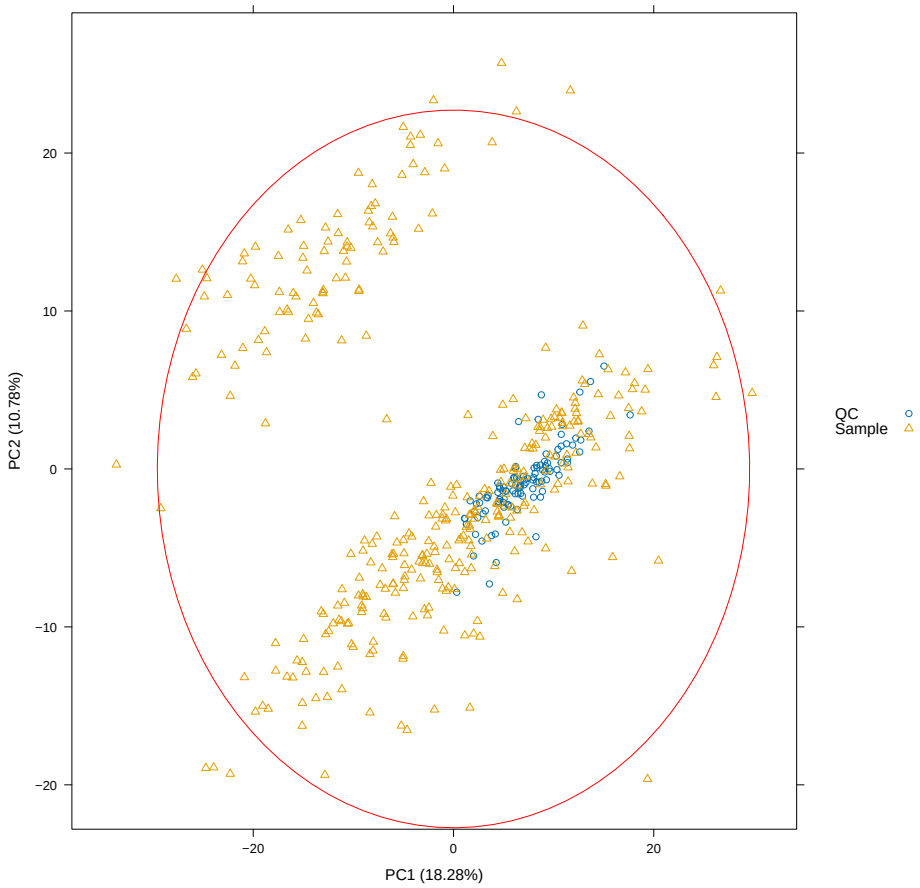
```
  res <- qc_rlsc(dat, cls.qc, method = method, opti = opti)
} else { # do signal correction inside each batch
  res <- lapply(levels(cls.bl), function(x) {
    idx <- cls.bl %in% x
    tmp <- qc_rlsc(dat[idx,], cls.qc[idx], method = method, opti = opti)
  })
  res <- bind_rows(res)
}
toc()
#> 14.918 sec elapsed
```

Data visualisation after signal correction

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

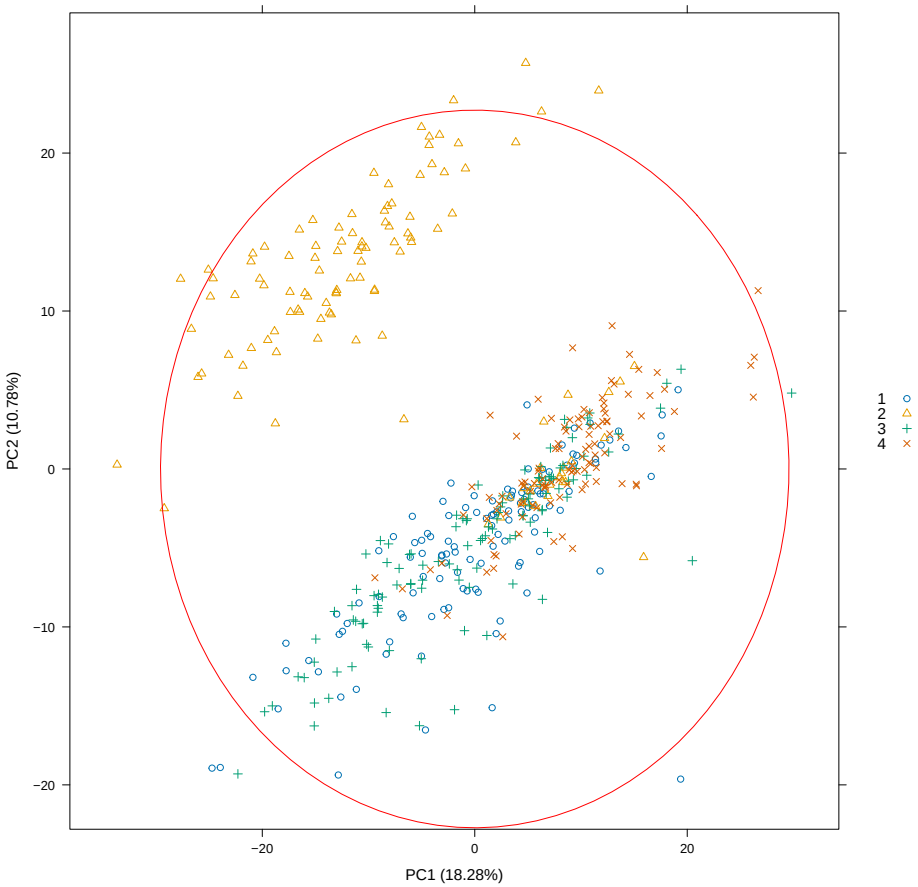
PCA plot for sample types

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



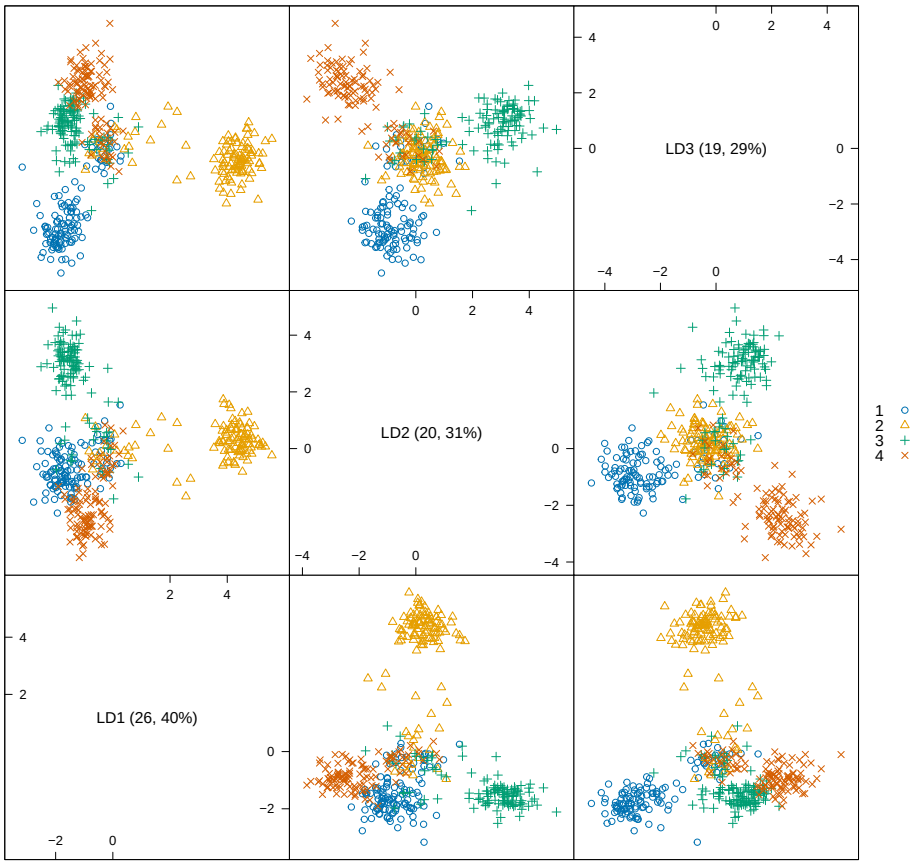
PCA plot for batches

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



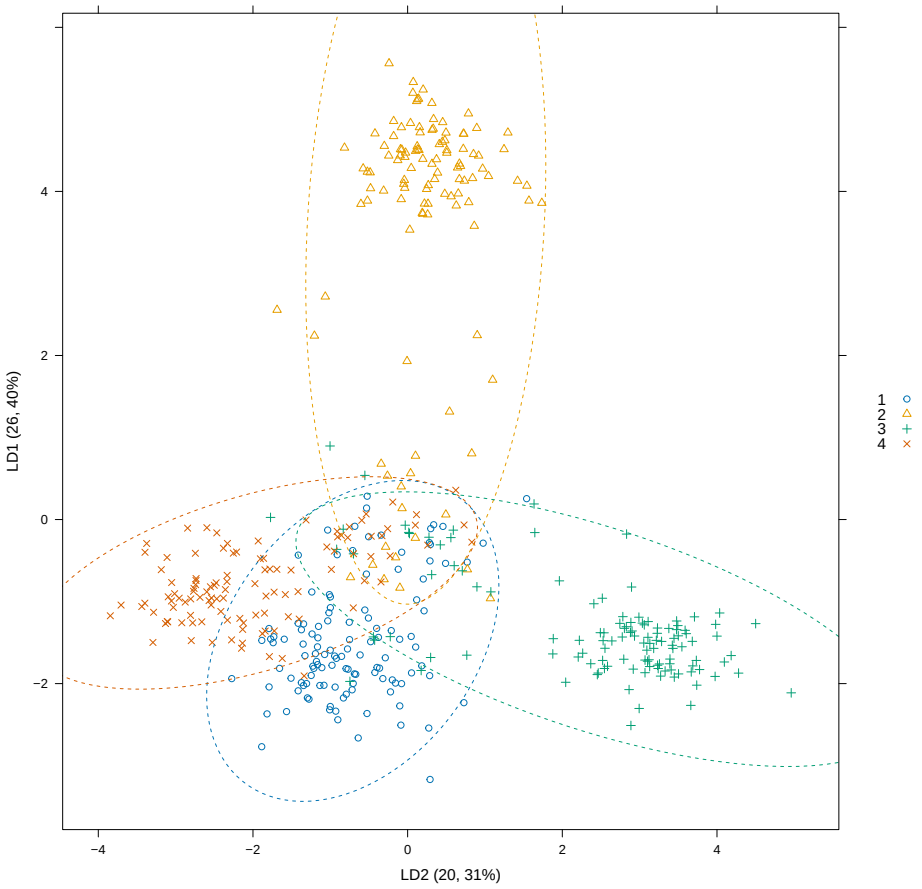
LDA plot for batches

```
plot(pcalda(res_fill, cls.bl))
```



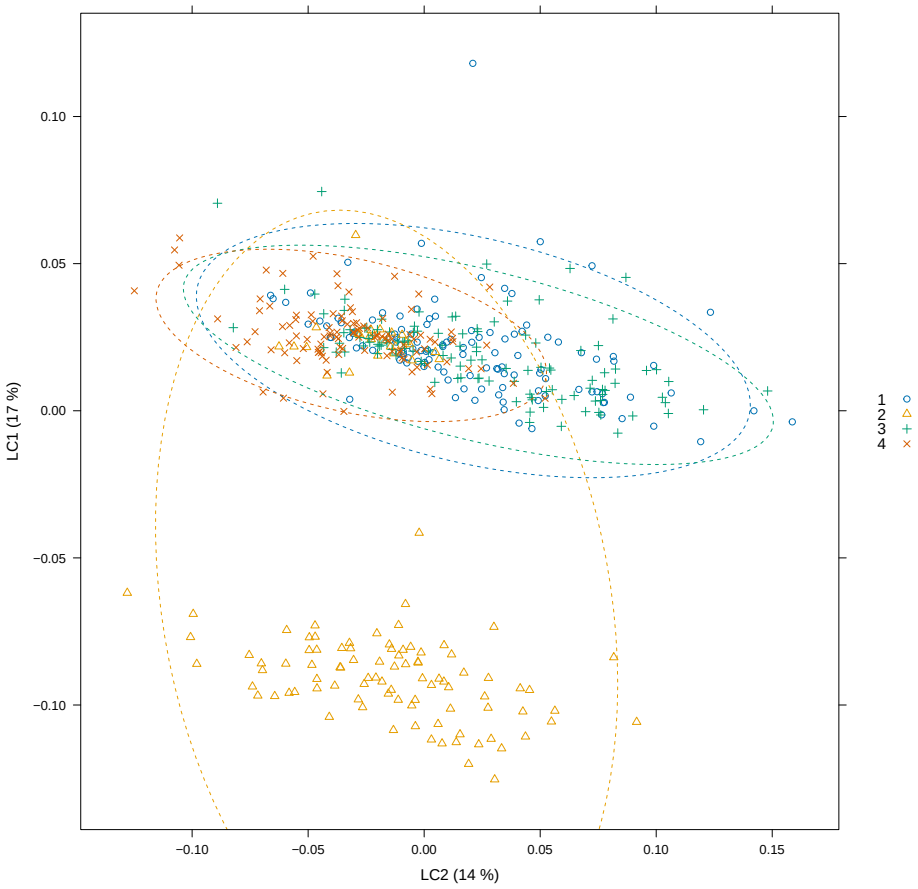
LDA plot of batches: LD1 vs LD2 (only for batch groups larger than 2)

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



PLS plot of batches: LC1 vs LC2

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



0.6 Batch shift

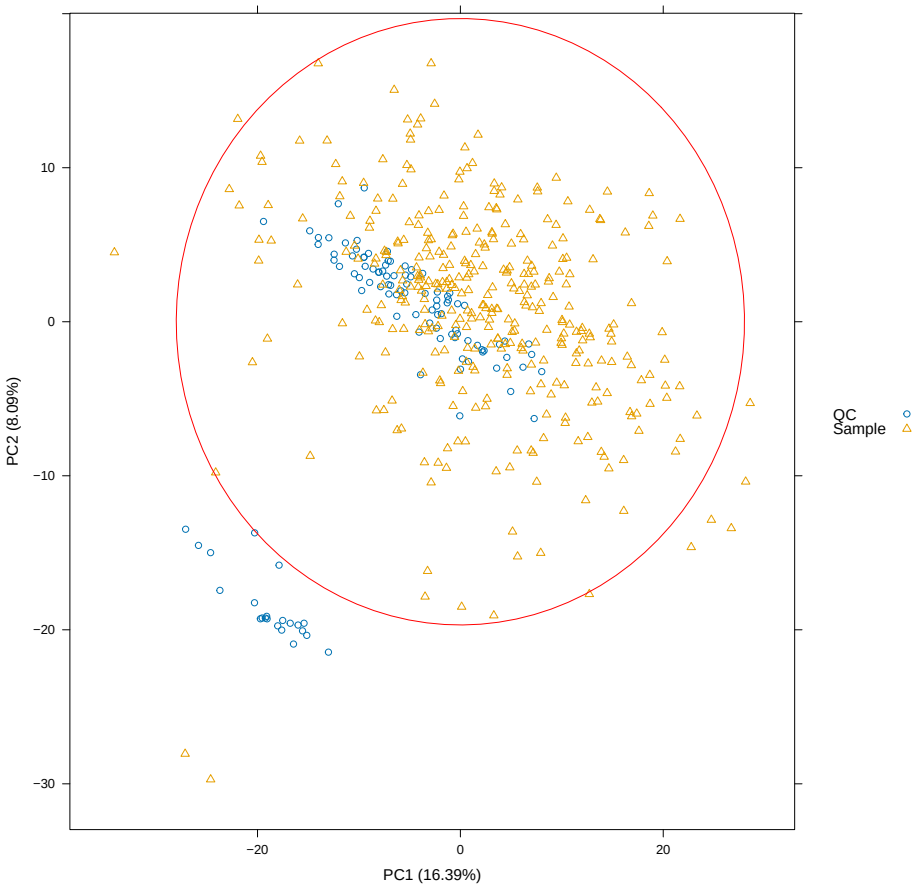
```
if (shift) {
  res <- batch_shift(res, cls.bl, overall_average = T) %>% as_tibble()
}
```

Data visualisation after batch shift

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

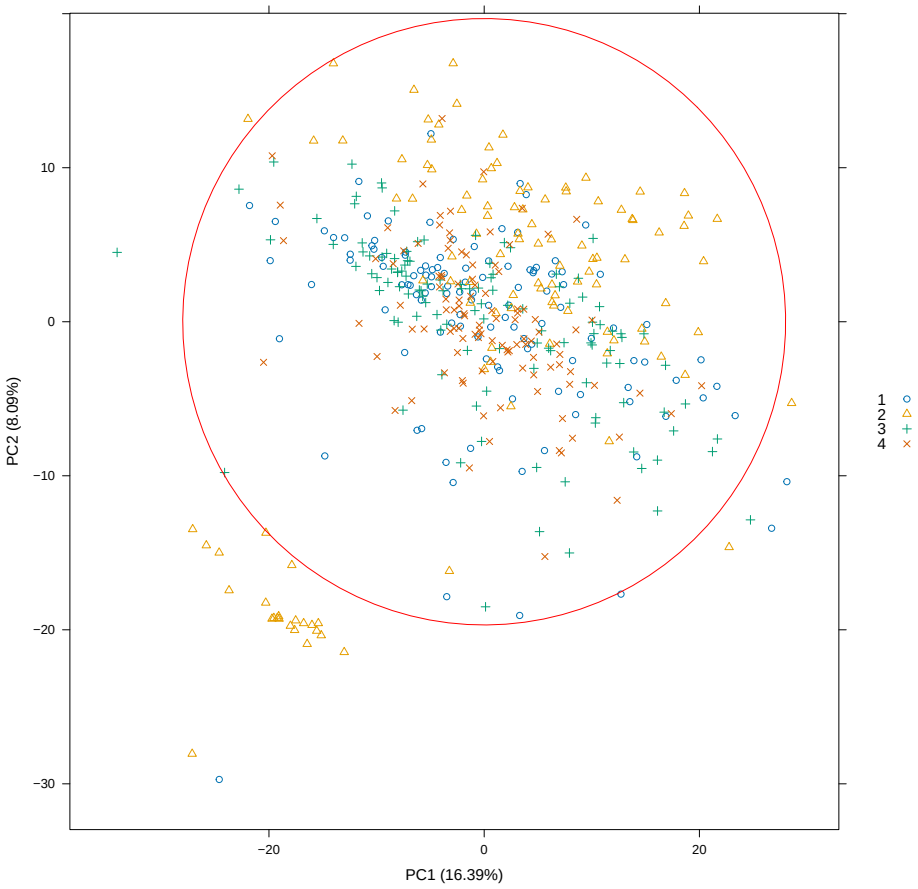
PCA plot for sample types

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



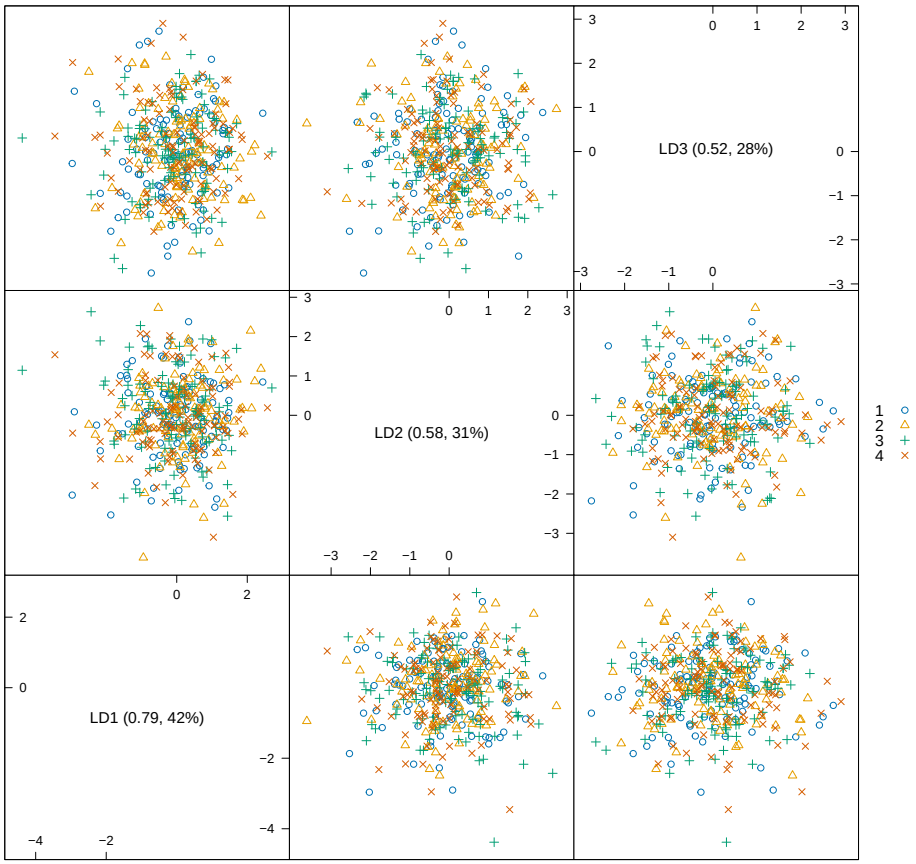
PCA plot for batches

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



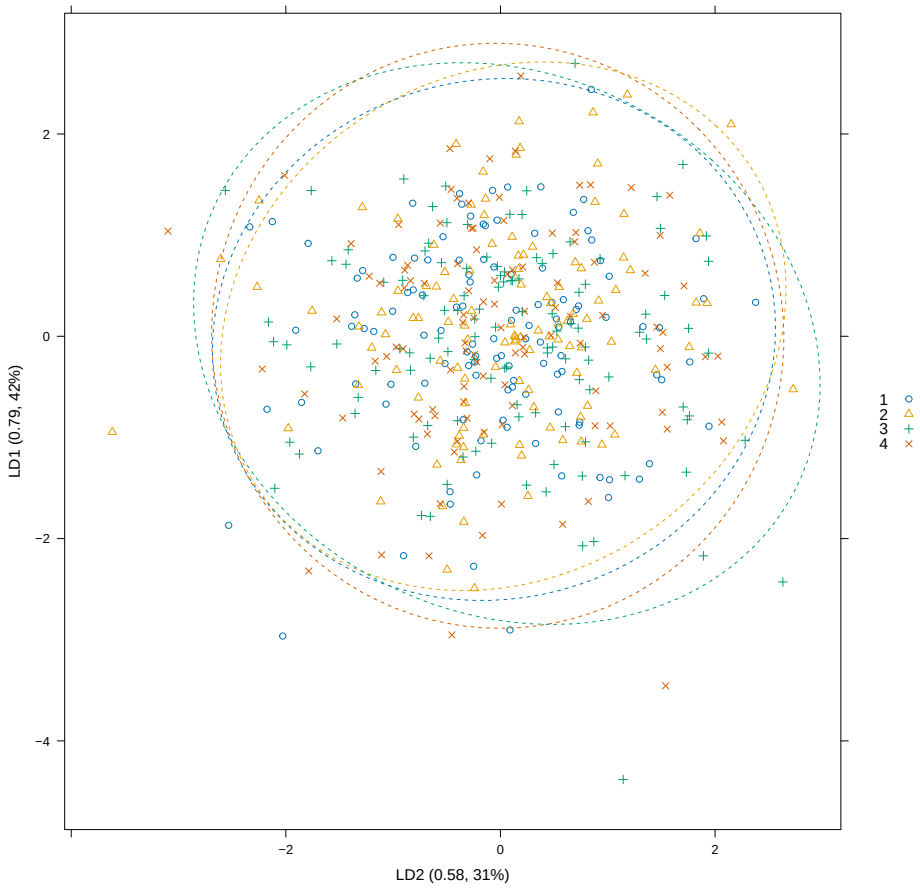
LDA plot for batches

```
plot(pcalda(res_fill, cls.bl))
```



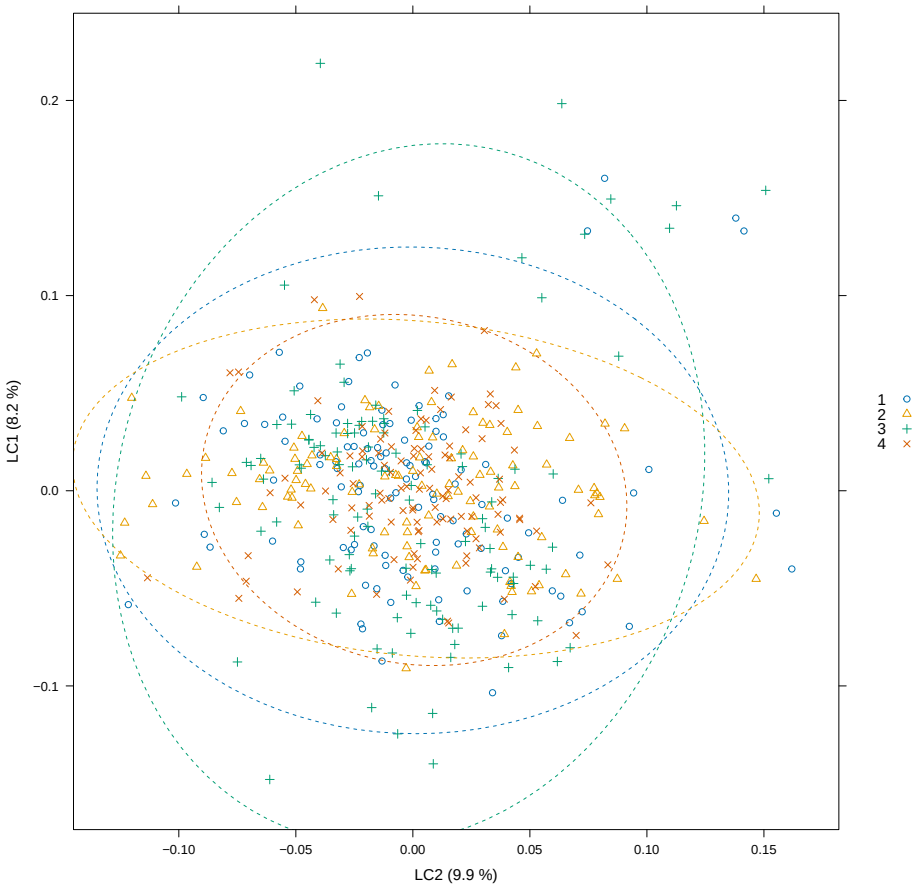
LDA plot of batches: LD1 vs LD2 (only for batch groups larger than 2)

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



PLS plot of batches: LC1 vs LC2

```
plot(plslda(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)

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