NormalyzerDE analysis of Bile Acid Proteomics Data

```
library(tidyverse)
library(limma)
library(edgeR)
library(readxl)
```

Maxquant Object Import

Read in the MaxQuant proteinGroups file. Only keep columns with iBAQ intensitites, gene info, and contaminant info. Remove iBAQ peptides column from the protein groups file.

```
mq <- read_excel("ravi_pg.xlsx", col_types = c("text",
    "text", "text", "text", "numeric", "numeric",
    "numeric", "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric", "numeric",
    "numeric", "text", "text", "text"))
mq_cleaned <- mq %>%
  filter(is.na(Reverse) , is.na(`Only identified by site`), is.na(`Potential contaminant`)) %>%
  select(`Protein IDs`, `Majority protein IDs`, `Protein names`, `Gene names` ,matches("iBAQ (([01][0-9]
codes <- read_excel(here::here("ravi_proteinGroups.xlsx"), sheet = 2) %>%
  mutate(reps = case_when(
    str_detect(identity, "C.c") ~ "Cc",
    str_detect(identity, "C.m") ~ "Cm",
    str_detect(identity, "L.c") ~ "Lc",
    str_detect(identity, "L.m") ~ "Lm"
  ))
```

Prepare tables for normalyzer analysis.

```
#Prepare objects for normalyzer
norm_meta <- codes %>%
  select("sample" = identity, "group" = reps)
norm_mq <- mq_cleaned %>%
  select(`Majority protein IDs`, contains("iBAQ")) %>%
  rename_with(function(x) norm_meta$sample [1:24], contains("iBAQ")) %>%
  filter(!str_detect(`Majority protein IDs`,"REV_"))
write_tsv(norm_meta [1:24,], file = "norm_meta.tsv")
write_tsv(norm_mq , file = "norm_mq.tsv")
```

NormalyzerDE: normalization

Use normalyzerDE to test all normalization methods.

```
library(NormalyzerDE)
library(SummarizedExperiment)
```

```
normalyzer(jobName="test_run", designPath="norm_meta.tsv", dataPath="norm_mq.tsv",
           outputDir = here::here("normalyzer/"))
```

Differential Abundance Analysis

Based on the QC plots from normalyzer DE, use vsn normalization for differential abundance analysis.

```
normalyzerDE("test_run", designPath = here::here("norm_meta.tsv"),
             dataPath=here::here("normalyzer/test_run/VSN-normalized.txt"),
             outputDir = here::here("normalyzer/"), comparisons = c("Cc-Lc","Cm-Lm"),
             sigThres = 0.05, log2FoldThres = 1)
```

Combine the original intensities table, and the DA analysis output.

```
norm_meta <- read_tsv(here::here("norm_meta.tsv"))
norm_mq_annot <- mq_cleaned %>%
  select(-contains("iBAQ"))
res <- read_tsv(here::here("normalyzer/test_run/test_run_stats.tsv"))
#Annotate the result table
res_annot <- res %>%
  left_join(norm_mq_annot)
write_tsv(res_annot, file = here::here("normalyzer_limma_de_results_annotated.tsv"))
```

Session Info

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Canada.1252  LC_CTYPE=English_Canada.1252
## [3] LC_MONETARY=English_Canada.1252 LC_NUMERIC=C
## [5] LC_TIME=English_Canada.1252
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
```

```
## [8] base
##
## other attached packages:
##  [1] SummarizedExperiment_1.24.0 Biobase_2.54.0
##  [3] GenomicRanges_1.46.1        GenomeInfoDb_1.30.1
##  [5] IRanges_2.28.0             S4Vectors_0.32.4
##  [7] BiocGenerics_0.40.0        MatrixGenerics_1.6.0
##  [9] matrixStats_0.62.0         NormalyzerDE_1.12.0
## [11] readxl_1.4.0               edgeR_3.36.0
## [13] limma_3.50.3              forcats_0.5.1
## [15] stringr_1.4.0             dplyr_1.0.9
## [17] purrr_0.3.4               readr_2.1.2
## [19] tidyr_1.2.0               tibble_3.1.7
## [21] ggplot2_3.3.6             tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] colorspace_2.0-3     deldir_1.0-6         ellipsis_0.3.2
##  [4] class_7.3-19         htmlTable_2.4.1      XVector_0.34.0
##  [7] base64enc_0.1-3      fs_1.5.2             rstudioapi_0.13
## [10] proxy_0.4-27         fansi_1.0.2          lubridate_1.8.0
## [13] xml2_1.3.3           codetools_0.2-18     splines_4.1.2
## [16] knitr_1.39           Formula_1.2-4        jsonlite_1.8.0
## [19] broom_1.0.0          cluster_2.1.2        dbplyr_2.2.1
## [22] png_0.1-7            compiler_4.1.2       httr_1.4.3
## [25] backports_1.4.1      assertthat_0.2.1     Matrix_1.3-4
## [28] fastmap_1.1.0        cli_3.3.0            htmltools_0.5.2
## [31] tools_4.1.2          GenomeInfoDbData_1.2.7 gtable_0.3.0
## [34] glue_1.6.2           Rcpp_1.0.8.3         carData_3.0-5
## [37] cellranger_1.1.0     raster_3.5-21        vctrs_0.4.1
## [40] preprocessCore_1.56.0 xfun_0.29           rvest_1.0.2
## [43] lifecycle_1.0.1      RcmdrMisc_2.7-2      terra_1.5-34
## [46] zlibbioc_1.40.0      MASS_7.3-54          zoo_1.8-10
## [49] scales_1.2.0         hms_1.1.1            sandwich_3.0-2
## [52] RColorBrewer_1.1-3   yaml_2.2.2           gridExtra_2.3
## [55] rpart_4.1-15         latticeExtra_0.6-30  stringi_1.7.6
## [58] nortest_1.0-4        e1071_1.7-11         checkmate_2.1.0
## [61] bitops_1.0-7         rlang_1.0.3          pkgconfig_2.0.3
## [64] evaluate_0.15        lattice_0.20-45      htmlwidgets_1.5.4
## [67] tidyselect_1.1.2     magrittr_2.0.3       R6_2.5.1
## [70] generics_0.1.3       Hmisc_4.7-0          DelayedArray_0.20.0
## [73] DBI_1.1.3            pillar_1.8.0         haven_2.4.3
## [76] foreign_0.8-81       withr_2.5.0          RCurl_1.98-1.7
## [79] survival_3.2-13      abind_1.4-5          sp_1.5-0
## [82] nnet_7.3-16          modelr_0.1.8         crayon_1.5.1
## [85] car_3.1-0            interp_1.1-2         utf8_1.2.2
## [88] tzdb_0.3.0           rmarkdown_2.14       jpeg_0.1-9
## [91] locfit_1.5-9.5       grid_4.1.2           data.table_1.14.2
## [94] reprex_2.0.1         digest_0.6.29        munsell_0.5.0
```