CDM5104

Experiment in R

```r
# Run this first to install packages
# install.packages(c("tidyverse", "here", "broom", "purrr"))
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.1.0
## v tidyr   1.3.0      v stringr 1.5.0
## v readr   2.1.4      v forcats 1.0.0
## -- Conflicts ---------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(here)
```

```
## here() starts at /Users/lsibjb/Code/CDMS4104_Lipidomics
```

```r
library(broom)
library(purrr)
```

Analysing lipid sum compositions

Import and prepare data

Read dataset, convert to long format and retrieve lipid class name from lipid name. Note: lipid class name is assumed to be anything before the first space. For ether and plasmalogen PCs (e.g. PE O- and PE P-), we change to the lipid name so that also for those the class name is before the first space.

Data from: Boretti et al., Metabolites, 2019 https://doi.org/10.1007/s11306-019-1621-3

```r
d_orig <- readr::read_csv(here("data/DogDietStudy_PlasmaLipidomics_Diets-12_CONCENTRATIONS.csv"))
```

```
## Rows: 16 Columns: 316
## -- Column specification ----------------------------------------------------------
## Delimiter: ","
## chr   (1): Diet
## dbl (315): DogID, CE 16:1, CE 18:1, CE 18:2, CE 18:3, CE 20:2, CE 20:3, CE 2...
##
## 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.
```

```
# Prepare data: convert to long, get LipidClass names
d_long <- d_orig |>
  pivot_longer(-DogID:-Diet, names_to = "Lipid" , values_to = "Conc") |>
  mutate(Lipid = str_replace(Lipid, fixed(" P-"), "-P "),
         Lipid = str_replace(Lipid, fixed(" O-"), "-O ")) |>
  separate(col = Lipid, into = c("LipidClass", "Chain"),remove = FALSE, sep = " ")
```

Calculate sum per class per sample

```
# Calculate sums of all species per class
d_sum <- d_long |>
  group_by(DogID, Diet, LipidClass) |>
  summarise(Conc = sum(Conc))
```

```
## 'summarise()' has grouped output by 'DogID', 'Diet'. You can override using the
## '.groups' argument.
```

```
# Save data as CSV file
write_csv(x = d_sum, file = here("output/DogDietStudy_SumCompositions.csv"))
head(d_sum)
```

```
## # A tibble: 6 x 4
## # Groups:   DogID, Diet [1]
##   DogID Diet   LipidClass    Conc
##   <dbl> <chr>  <chr>        <dbl>
## 1     1 Diet-1 CE        8814.
## 2     1 Diet-1 Cer          5.57
## 3     1 Diet-1 DG         129.
## 4     1 Diet-1 GM3          1.28
## 5     1 Diet-1 Hex1Cer      0.547
## 6     1 Diet-1 Hex2Cer      1.32
```

Perform a Welch's t test between diet 1 and diet 2

We could also consider to log transform the data

```
# get t test results (p, FDR, and fold-change)
d_stat <- d_sum |>
  group_by(LipidClass) |>
  nest() |>
  mutate(res = purrr::map(data, \(x) broom::tidy(
    t.test(Conc ~ Diet, var.equal = FALSE, data = x))
    )) |>
  unnest(res) |>
  ungroup() |>
  mutate(FDR = p.adjust(p.value, method = "BH"),
         log2FC = log2(estimate2/estimate1)) |>
  select(LipidClass, log2FC, p.value, FDR)

head(d_stat)
```

```
## # A tibble: 6 x 4
##   LipidClass  log2FC p.value    FDR
##   <chr>        <dbl>   <dbl>  <dbl>
## 1 CE         -0.148   0.491  0.495
## 2 Cer        -0.350   0.0871 0.168
## 3 DG         -1.13    0.0234 0.0643
## 4 GM3        -0.0590  0.495  0.495
## 5 Hex1Cer    -0.111   0.467  0.495
## 6 Hex2Cer    -0.718   0.0127 0.0399
```

Simple Dot plot

Note: y axis is in log scale

```
ggplot(d_sum, aes(x=LipidClass, y = Conc, group = Diet, color = Diet)) +
  geom_point(position = position_dodge(width = 0.5)) +
  scale_y_log10() +
  theme_bw() +
  theme(axis.text.x = element_text(size = 7,angle = 45, hjust = 1))
```
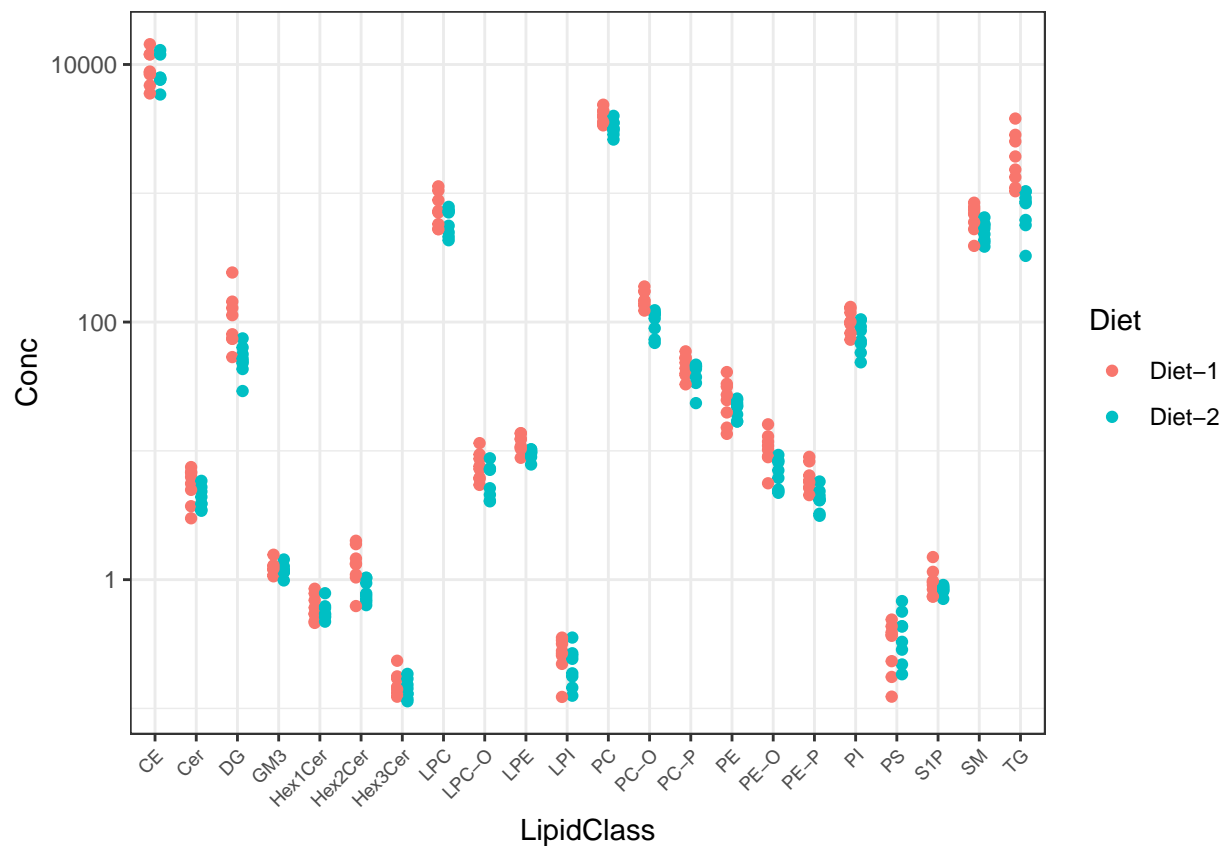

Simple volcano plot

```
ggplot(d_stat, aes(x=log2FC, y = -log10(p.value), label=LipidClass)) +
  geom_point() +
  geom_text(nudge_y = -0.1,nudge_x = -0.1, size=3) +
  scale_x_continuous(limits = c(-2,2)) +
  geom_vline(xintercept = 0, color = "grey50", linewidth =.4)+
  geom_hline(yintercept = -log10(0.05), color = "green")+
  theme_bw()+
  theme(aspect.ratio=1)
```