Meat comparison report

Lisa Schneider

April 10, 2020

Installation and namespaces

The files with the self-written functions for this workflow are loaded with source(). The first source-file "install_packages_lipidome_comparison.R" contains a function that installs all required packages. The installed packages are loaded into the namespace with library().

```r
source("install_packages_lipidome_comparison.R")
# install_packages_lipidome_comparison() # installs all reqired packages for this workflow

library(dplyr) # select part of data
library(stringr) # count separators
library(data.table) # transpose data frame
library(imputeLCMD) # various imputation procedures, including left censored imputation
library(impute) # dependecy for imputeLCMD
library(ggplot2) # plots
library(tibble) # data frame manipulation
library(viridis) # colorblind save color schemes
library(GGally) # paralell plot
library(fmsb) # spider chart
library(scales) # scale opacity of filling (alpha)
library(ggrepel) # avoids overlapping labels in ggplot graphs
library(factoextra) # graphs for PCA
library(ggfortify) # biplot with ggplot
library(heatmaply) # interactive heatmap
library(gplots) # heatmap
library(plotly) # interactive ggplots

source("lipidome_comparison_dataTransformaions.R")
source("lipidome_comparison_EDA.R")
source("lipidome_comparison_pca.R")
source("lipidome_comparison_clustering.R")
source("lipidome_comparison_hypothesis_testing.R")
```

Set directorys

This is set for my system. The syntax for windows is slightly different. Making this work on all systems, maybe from command line is planned for later.

```r
working_directory <- "/home/lisa/FH/Masterarbeit/LipidomeComparison"
setwd(working_directory)

test_path <- "/home/lisa/FH/Masterarbeit/LipidomeComparison/data/Probe-Datensatz_lisa.csv"
```

```
meat_data_path <- "/home/lisa/FH/Masterarbeit/LipidomeComparison/data/meat_fish_final_raw.csv"

plot_path <- paste(working_directory, "/plots", sep = "")
plot_name <- paste(plot_path, "/meat_data", sep = "")
```

A lot of data preprocessing

The meat data was used without preprocessing, except exproting the excel sheet to csv. The data was read from csv to the script and stripped of meta data. The string "N/F" for not identified lipids was replaced with NA and the data was separated in target data and internal standard data. The target data was then re-formatted, with the compounds as the columns and the sample IDs as the rows. The used function flip_df() is from the "lipidome_comparison_dataTransformaions.R" file. Metadata was extracted from the sample IDs and the data set was separated by extraction method (AS and N).

```
meat_data <- read.csv(meat_data_path, sep = ",", dec = ".", header = TRUE)
meat_data <- subset(meat_data, select = c(Compound, Type, Filename, Status, Group, Area))
meat_data <- subset(meat_data, Status == "Processed")
meat_data[meat_data==''] <- NA
meat_data[meat_data=='N/F'] <- NA
meat_data$Area <- as.numeric(meat_data$Area)
meat_target <- subset(meat_data, Type == "Target Compound")
meat_standard <- subset(meat_data, Type == "Internal Standard")

meat_target <- flip_df(meat_target)

meat_target <- subset(meat_target, !is.na(Group))
meat_target$SID <- sub(".*probe","sample", meat_target$SID)
meat_target$SID <- sub("\\.*pos2","2", meat_target$SID)
meat_target$SID <- sub("\\.*pos","1", meat_target$SID)

meat_AS <- meat_target$SID[str_detect(meat_target$SID, "AS") == TRUE]
meat_target$SID[str_detect(meat_target$SID, "AS") == TRUE] <- sub(".*sample","AS_sample", meat_AS)
meat_N <- meat_target$SID[str_detect(meat_target$SID, "AS") == FALSE]
meat_target$SID[str_detect(meat_target$SID, "AS") == FALSE] <- sub(".*sample","N_sample", meat_N)
meat_target$SID <- str_remove(meat_target$SID, "_AS")

meta_info <- read.table(text = meat_target$SID, sep = "_")
colnames(meta_info) <- c("Treatment", "Sample_nr", "Biol_rep", "Tech_rep")
meta_info$Biol_rep <- paste(meta_info$Sample_nr, meta_info$Biol_rep, sep = "_")
meta_info$Tech_rep <- paste(meta_info$Biol_rep, meta_info$Tech_rep, sep = "_")

meat_target <- cbind(meat_target$SID, meta_info, meat_target[, -1])
meat_target <- droplevels(meat_target)
levels(meat_target$Group)[levels(meat_target$Group) == "fleisch"] <- "meat"
levels(meat_target$Group)[levels(meat_target$Group) == "wild"] <- "game"
levels(meat_target$Group)[levels(meat_target$Group) == "FISCH"] <- "fish"
colnames(meat_target) <- c("SID", colnames(meat_target[-1]))
head(meat_target[1:7], 3)
```

```
##              SID Treatment Sample_nr  Biol_rep    Tech_rep Group LPC (16:0)_1
## 4 N_sample1_1_1         N   sample1 sample1_1 sample1_1_1  meat         1234
## 5 N_sample1_2_1         N   sample1 sample1_2 sample1_2_1  meat         3481
## 6 N_sample1_3_1         N   sample1 sample1_3 sample1_3_1  meat          242
```

```
meat_N <- subset(meat_target, Treatment == "N")
meat_AS <- subset(meat_target, Treatment == "AS")
```

Data imputation

It can be assumed that most of the missing values are due to the concentration being below the detection limit. Therefore the missing values are considdered non-random and left-censored. Imputation only works up to a certain percentage of missing values, without changing the results. I found 20% missing values in the literature (Gurke, 2019, doi: 10.3389/fpsyt.2019.00041), therefore all compounds with more than 20% missing values were filtered. The remaining missing values where imputed using the QRILC (Quantile Regression Imputation of Left-Censored data) procedure. This procedure has the disadvantage of producing negative values (because it performs random draws from a distribution), which is a problem for the calculation of the log2 foldchange (I'm working on that).

```
# remove columns where > 20% of the values are missing
impute_meat <- meat_N[, which(colMeans(!is.na(meat_N)) > 0.8)]
impute_meat <- as.matrix(select_if(impute_meat, is.numeric))

# perform missing data imputation
meat_QRILC <- impute.QRILC(impute_meat, tune.sigma = 1) #todo constraints agains negative values
meat_imputed <- as.data.frame(meat_QRILC[[1]])
meat_imputed <- cbind(meat_N[, 1:6], meat_imputed)
meat_imputed <- droplevels(meat_imputed) # remove unused levels from factors
head(meat_imputed[1:7], 3)
```

```
##             SID Treatment Sample_nr  Biol_rep    Tech_rep Group LPC (16:0)_1
## 4 N_sample1_1_1         N   sample1 sample1_1 sample1_1_1  meat         1234
## 5 N_sample1_2_1         N   sample1 sample1_2 sample1_2_1  meat         3481
## 6 N_sample1_3_1         N   sample1 sample1_3 sample1_3_1  meat          242
```

Reducing the replicates

To not artificially produce more samples than we hat and to reduce variation in preprocessing and measurement the means for each sample or biological replicate are calculated. The idea is, to continue working with these values, but in most cases this reduces the sample number below a point where most procedures work (n[fish] = 1). So in the for now I continued working with the non reduced data.
The self-wirtten functions in this part are in "lipidome_comparison_dataTransformaions.R" and "lipidome_comparison_EDA.R".

```
meat_groups <- generate_categorical_table(meat_imputed$Group)
meat_treatment <- generate_categorical_table(meat_imputed$Treatment)

meat_numeric <- meat_imputed
meat_numeric$Group <- as.numeric(meat_numeric$Group)
meat_biol <- calc_by_replicate(meat_numeric, meat_numeric$Sample_nr, mean)
meat_tech <- calc_by_replicate(meat_numeric, meat_numeric$Biol_rep, mean)

new_meat_biol <- paste_catecorical_variable(meat_biol, 2, meat_groups)
head(new_meat_biol[1:7], 3)
```

```
##   Group.1 Group LPC (16:0)_1 LPC (16:0)_2 LPC (16:1)_1 LPC (18:0) LPC (18:1)_1
## 6 sample6  meat      1397.60     3254.600       1102.8   2067.600     3179.000
## 1 sample1  meat      1327.75     1950.500       1023.5   1136.000     1889.000
## 2 sample2  meat      1659.00     1126.667       2616.0   2099.333     3313.333
```

```
new_meat_biol <- paste_catecorical_variable(meat_tech, 2, meat_groups)
head(new_meat_biol[1:7], 3)
```

```
##      Group.1 Group LPC (16:0)_1 LPC (16:0)_2 LPC (16:1)_1 LPC (18:0)
## 12 sample6_1  fish       1534.0       3492.0         1468       3366
## 13 sample6_2  fish       1273.5       3110.0          852       1695
## 14 sample6_3  fish       1453.5       3280.5         1171       1791
##    LPC (18:1)_1
## 12       3250.0
## 13       3038.0
## 14       3284.5
```

Exploratory data analysis

Exploratory data analysis was performed graphical and using the shapiro-wilk test. The graphical data analysis (using qqplot, histogram and boxplot) turned out not to be verry practical due to the numer of compounds, but it gives a good overview over the distributions of the variables. It showed, that some of the variables had normal distribution, while others where multi modal. Therefor the results of the following tests for normal distribution and correlation have to be handelled carefully. All the graphs are in a separate zip-folder due to thier size and runtime.

```
### test for normality
meat_normality <- shapiro_by_factor(meat_imputed, meat_imputed$Group)
```

```
## [1] "p < 0.05 ... no normal distribution; p > 0.05 ... normal distribution"
```

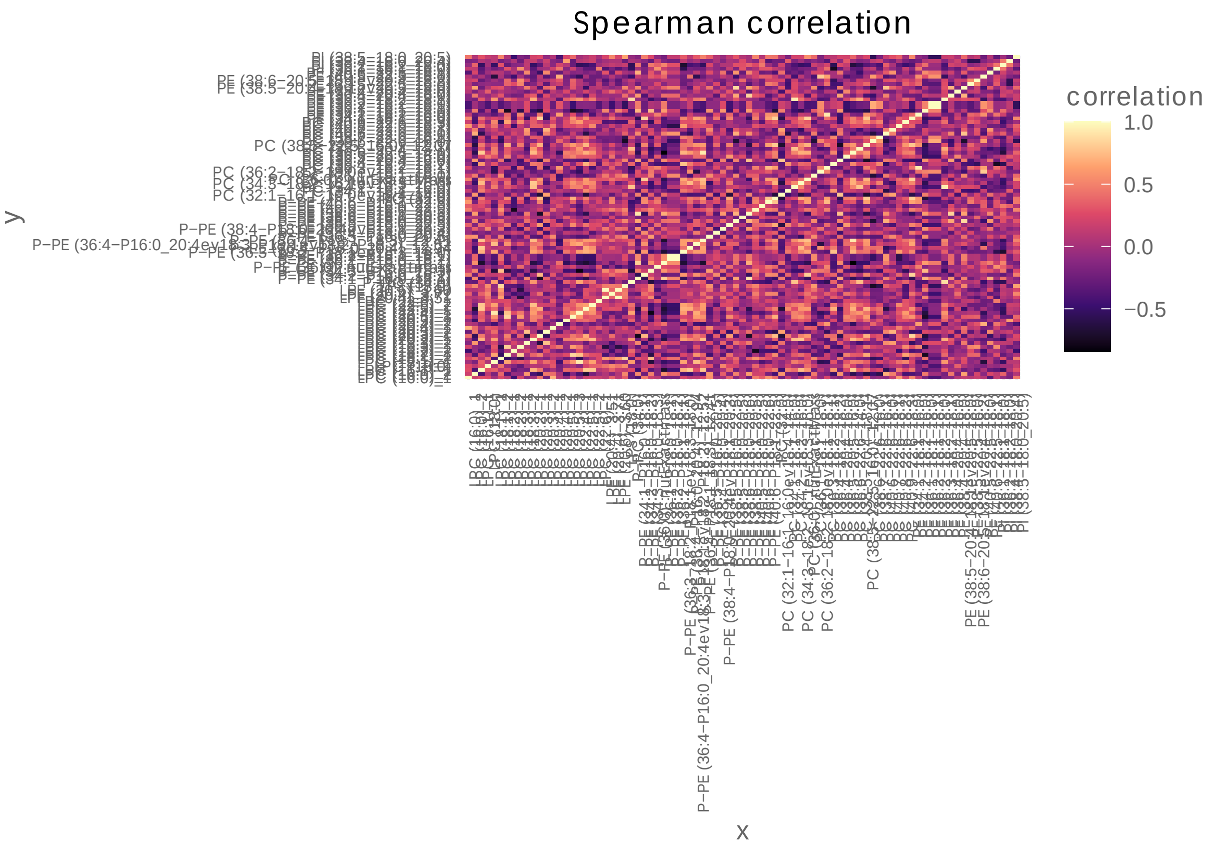
```
head(meat_normality[1:7], 3)
```

```
##   Group.1 LPC (16:0)_1 LPC (16:0)_2 LPC (16:1)_1 LPC (18:0) LPC (18:1)_1
## 1    fish   0.71426005   0.83493706  0.814781645 0.01310474   0.25602914
## 2    meat   0.09764837   0.01816648  0.008933696 0.21283253   0.00543952
## 3    game   0.41622511   0.68388787  0.167223408 0.92867852   0.93190887
##   LPC (18:1)_2
## 1   0.86421833
## 2   0.04870863
## 3   0.48844782
```

```
### test for correlation
meat_correlation <- cor(select_if(meat_imputed, is.numeric), method = "spearman")
head(meat_correlation[1:7], 3)
```

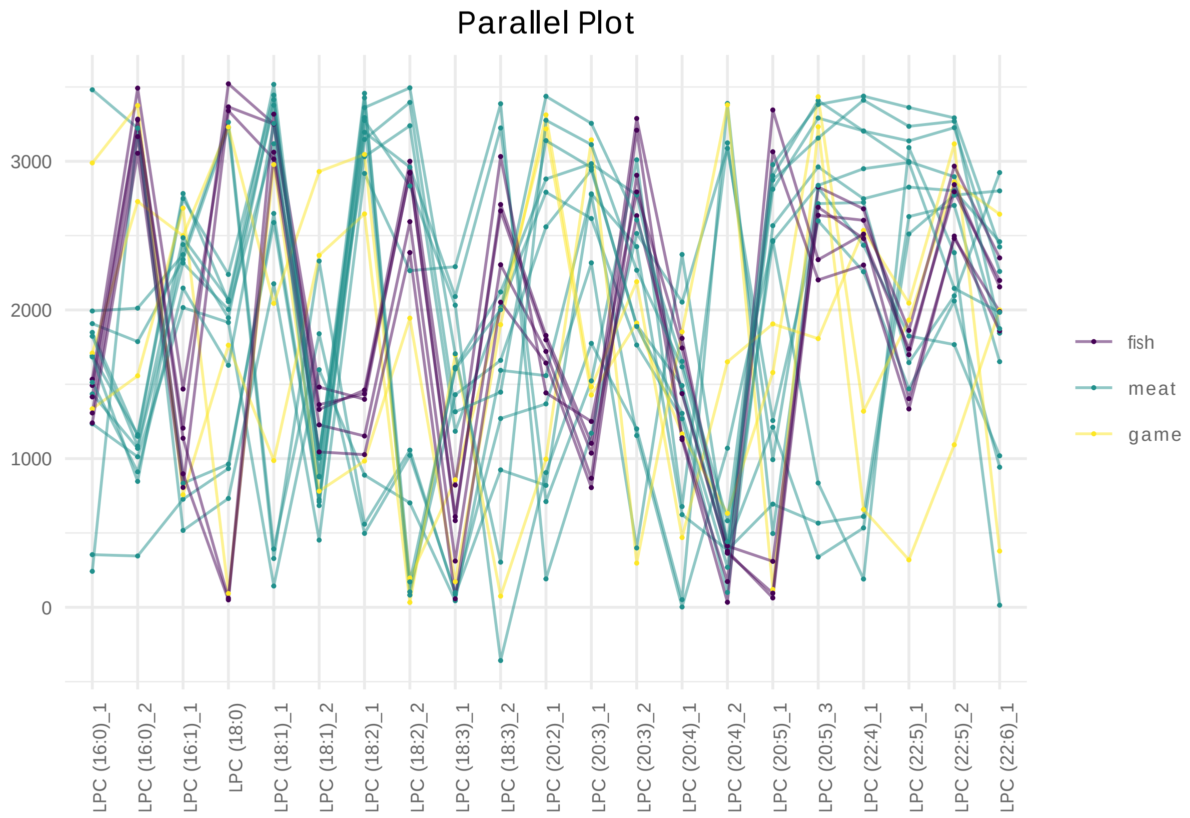
```
## [1] 1.0000000 0.1774436 0.2766917
```

```
correlation_heatmap(meat_imputed, interactive = FALSE)
```

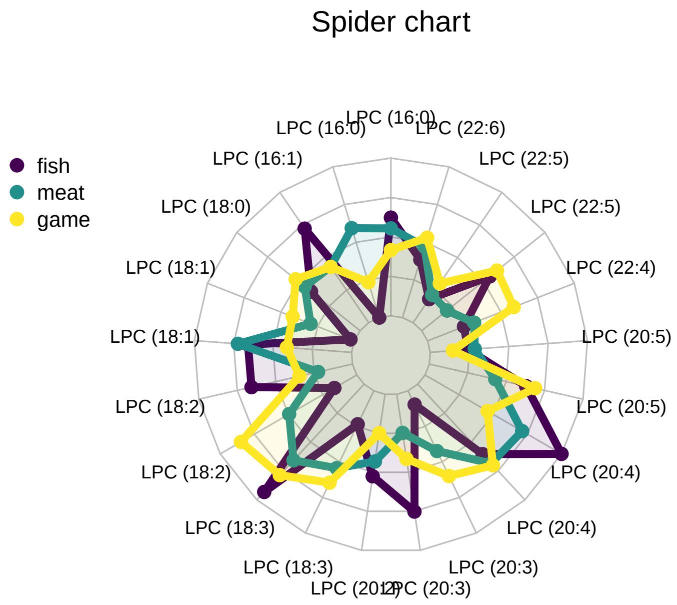


Some additional graphs for EDA: These graphs only make sense when comparing a specific part of the data set. Therefor I slices where only LPC is compared. In the paralell plot we can see, that there is a small variation between the fish samples (which makes sense, because they are only technical replicates) and more variation in the meat and game samples. A separation by group is not visible in any of the variables (including those not shown in the plot).

```
parallel_plot(meat_imputed[1:27], meat_imputed$Group)
```



```
meat_spider <- calc_by_replicate(meat_imputed, meat_imputed$Group, mean)
spider_chart(meat_spider[,1:22], legend_lab = meat_spider$Group.1)
```


Spider chart
fish
meat
game
LPC (16:0)
LPC (16:0)
LPC (22:6)
LPC (16:1)
LPC (22:5)
LPC (18:0)
LPC (22:5)
LPC (18:1)
LPC (22:4)
LPC (18:1)
LPC (20:5)
LPC (18:2)
LPC (20:5)
LPC (18:2)
LPC (20:4)
LPC (18:3)
LPC (20:4)
LPC (18:3)
LPC (20:3)
LPC (20:2)
LPC (20:3)