BIC metabolomics data analysis

In this document we present the joint analysis of the different metabolomics datasets submitted to BIC (currently, as of June 2019). We rely on the following resources for metabolomics-specific issues (in addition to the MOP): Gorrochategui et al. (https://www.sciencedirect.com/science/article/pii/S0165993616300425), section 3.2.5 on data intensity normalization.

Load the parsed meta-data (from the cloud), required for all analyses presented here.

```
system(paste("~/google-cloud-sdk/bin/gsutil",
             "cp gs://bic_data_analysis/pass1a/pheno_dmaqc/merged_dmaqc_data.RData",
             "."))
load("merged_dmaqc_data.RData")
system("rm merged_dmaqc_data.RData")

# load required libraries
library(ggplot2)
library(reshape2)
library(gridExtra)

# load our helper functions
source("https://raw.githubusercontent.com/david-dd-amar/motrpac/master/tools/preprocessing_helper_functi
```

1 Untrageted data from Broad

Unfortunately, as of June 2019, we do not have batch or qc metrics info with this submission. Load the data:

```
broad_dir = "/Users/David/Desktop/MoTrPAC/data/pass_1a/metabolomics/broad_untargeted/"
data_matrix_file = "broad_pass1a_combined_wide.txt"
raw_data_broad = read.delim(paste(broad_dir,data_matrix_file,sep=""),check.names = F,
                            stringsAsFactors = F)
sample_info_file = "broad_pass1a_sampleType.txt"
sample_info = read.delim(paste(broad_dir,sample_info_file,sep=""),check.names = F,
                            stringsAsFactors = F)
# get the samples data using the vial ids:
broad_meta = merged_dmaqc_data[is.element(
  set=colnames(raw_data_broad),merged_dmaqc_data$viallabel),]
rownames(broad_meta) = broad_meta$viallabel
print("Broad untargeted data loaded, the represented samples are:")
```

```
## [1] "Broad untargeted data loaded, the represented samples are:"
```

```
print(table(broad_meta$sampletypedescription))
```

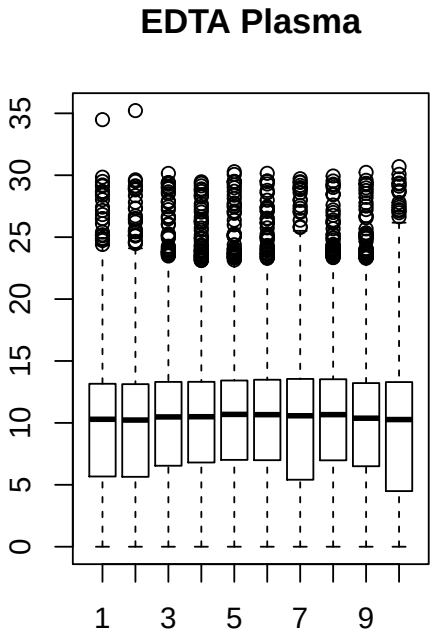
```
##
##   EDTA Plasma Gastrocnemius         Liver White Adipose
##            78            78            78            78
```

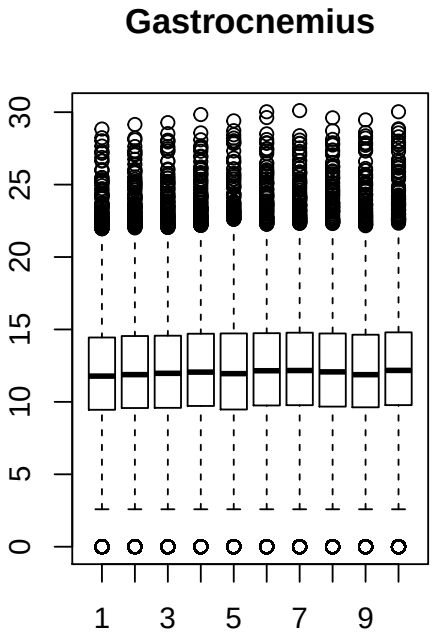
1.1 Sanity check: abundance data distribution

```
par(mfrow=c(1,2))
tissue2filtered_data = list()
# bxplots = list()
for (tissue in unique(broad_meta$sampletypedescription)){
  curr_vialids = as.character(
    broad_meta$viallabel[broad_meta$sampletypedescription==tissue])
  tissue_data = raw_data_broad[,curr_vialids]
  print(table(is.na(tissue_data)))
  tissue_data[is.na(tissue_data)] = 0
  tissue_data = log(tissue_data+1,base=2)
  print(table(apply(tissue_data==0,1,all)))
  tissue_data = tissue_data[!apply(tissue_data==0,1,all),]

  boxplot(tissue_data[,1:10],main=tissue,names=NULL,labels=NULL)
  tissue_data = run_quantile_normalization(tissue_data)
  tissue2filtered_data[[tissue]] = tissue_data

  # Comment out: ggplot is too much for this simple plot...
  # bxplots[[tissue]]=ggplot(data = melt(tissue_data[,1:10]), aes(x=variable, y=value)) +
  #   geom_boxplot() +
  #   theme(axis.text.x = element_text(angle = 45,size=10))+
  #   ggtitle(tissue)+
  #   theme(plot.title = element_text(hjust = 0.5,size=14))
}
```

```
##
##  FALSE   TRUE
## 517579 452273
##
## FALSE  TRUE
##  8295  4139
```

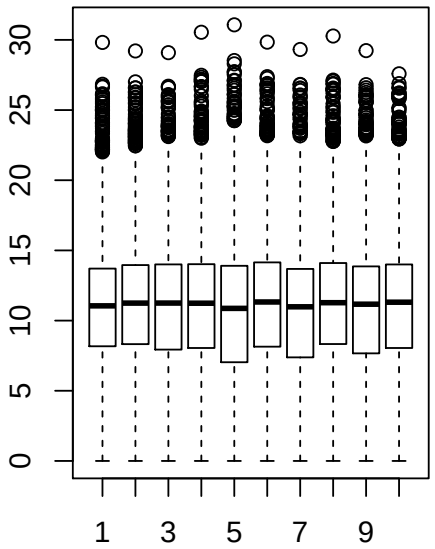
```
##
##  FALSE   TRUE
## 410669 559183
##
## FALSE  TRUE
##  5911  6523
```

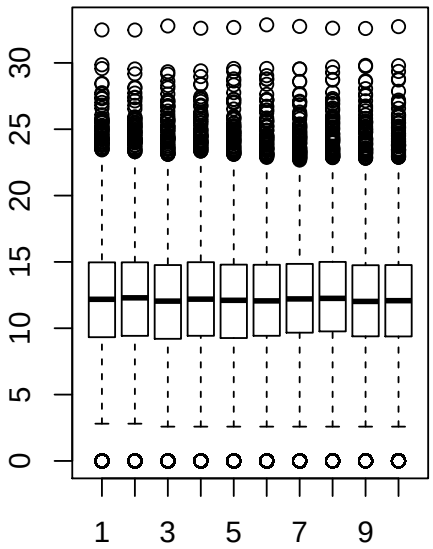





```
##
##  FALSE   TRUE
## 499172 470680
##
## FALSE  TRUE
##  7635  4799
```

```
##
##  FALSE   TRUE
## 576739 393113
##
## FALSE  TRUE
##  8523  3911
```





1.2 PCA plots

1.3 Correlations with meta/pheno data

1.4 Differential analysis