Sexually dimorphic expression patterns in human liver cells

Paul Rognon

June 6th, 2020

Contents

1 Introduction

Blekhman et al. (2010) use RNA-sequencing of female and male humans, chimpanzees, and rhesus macaques samples of liver cells to study sex-specific and lineage-specific changes in gene expression patterns. They identify individual genes whose regulation likely evolved under different scenarios of selection in primates: stabilizing selection regardless of the sex, directional selection in the human lineage regardless of sex and conserved sexually dimorphic expression patterns. In a subsequent analysis they study expression patterns at exon level to characterize differential and conserved alternative splicing between sexes and species. In this work, we use the data produced by the authors but retain the human samples annotated at gene level only and will therefore focus on differential expression between sexes.

1.1 Hypothesis

Our working hypothesis is that sexual dimorphism in humans is reflected in a different pattern of genes expression. There are genes in liver tissues that are differentially expressed between sexes in humans. This hypothesis differs from the ones in Blekhman et al. (2010), where the authors work on differential expressions under different natural selection scenarios across species and sexes.

1.2 Objective

We want to understand the differences in genes expression between sexes among humans and determine what biological components, functions and processes they impact. To do so we want to find the differentially expressed (DE) genes and select among them a list of genes which expression pattern characterizes the difference between sexes. Once the list has been established, we want to find which biological components, functions and processes are particularly associated with those genes.

2 Material and methods

2.1 Data acquisition and experimental design

Our data is composed of RNA sequencing of 6 samples of human liver tissues, with 3 biological replicates for each sex, gathered and sequenced by Blekhman et al. (2010). The samples were collected from healthy adults by the National Disease Research Interchange (NDRI). Blekhman et al. (2010) sequenced the samples using Illumina's Solexa technology. Each sample was sequenced using two lanes distributed over multiple flow-cells. The authors processed the data from reads to counts and both raw data (sequence reads) and processed data (counts) can be downloaded from the Gene Expression Omnibus (GEO) database with accession number GSE17274. The original processed data for human samples had two technical replicates per biological replicate. However, we work from transformed count data where the technical replicates have been merged so eventually have a dataset of 6 biological replicates, 3 for each group: HSM1, HSM2, HSM3 are male samples and HSF1, HSF2, HSF3 are female samples. We then have a two-independent group design with a single factor, sex. The transformed count data has already been aligned to the human genome and annotated with ENSEMBL gene IDs.

2.2 Filtering

We apply an independent filtering to our data based on the gene counts scaled by the total size of the library, count-per-million (cpm). We filter out genes with a cpm below our threshold in a minimum number of samples. Indeed, the data contains numerous genes with null or very low counts in most libraries that provide no useful information to identify differential expression between sexes. Including them in our analysis is harmful to our statistical work because the larger the number of genes the smaller the power of the statistical test after adjustment for multiple testing. Moreover, the presence of very lowly expressed genes can make estimation of the mean-variance relationship in the data less reliable. Independent filtering is a critical step of RNA-seq data analysis where important information can be lost, therefore in this work we discuss different thresholds for the cpm values, see Section 4.3.

2.3 Quality control checks

We first check our data for the effect of library size and compositional bias. A common issue in RNA-seq data is large differences in total number of reads between samples that create a technical bias when testing for differential expression. A large difference in counts for a given gene between two samples might be an artifact caused by a much larger sequencing depth in one of the samples and differences in proportions in the libraries. We assess graphically the presence of a compositional bias plotting the total library size across samples and mean-difference plots on the log cpm values. We then check our data for homogeneity of count distribution across samples with boxplots of the samples distributions of the log cpm values and MA plot between samples of the same sex. Finally, we look for potential batch effect with different approaches: multidimensional scaling (MDS), principal component analysis, hierarchical clusterings on the matrix of counts and on the distance matrix between samples. We perform the quality control checks before and after the normalization described in the next paragraph.

2.4 Normalization

RNA-seq data are affected by a range of technical biases, a normalization is needed. We normalize our count data with the trimmed mean of M-values (TMM) method. We choose this method firstly because it is a between-samples normalization method. Indeed, we are looking for differentially expressed genes so we want differences in normalized counts between samples to represent true differences in expression. We are not interested by within samples comparisons so do not correct for effects such as gene length and GC-content targeted by within sample normalizations. Among between-samples normalization, adjusting only for the

total library size as we did with the cpm values is not enough because, even though it tackles the effect of total size, the method does not consider the bias introduced by the changing proportion of mRNA corresponding to a given gene across biological conditions. Adjusting for total library sizes only assumes is too sensitive to few highly expressed genes having a large share of total expression. According to the review by Evans, Hardin, and Stoebel (2018), other families of between-sample normalization methods include normalizations by distribution/testing and normalizations by controls. The latter assumes the existence of control genes in the data, an information that we do not have. Normalization by controls is therefore not an option.

Normalizations by distribution/test are based on the idea that non-DE genes should have, on average, the same normalized counts across conditions and normalization factors can then be computed by equilibrating expression levels for non-DE genes. If technical effects impact non-DE genes and DE genes alike, then we can normalize all genes with the same normalization factor as the non-DE genes. Evans, Hardin, and Stoebel (2018) showed that DESeq and TMM normalizations are among the best performing methods in terms of ability to detect differentially expressed genes and controlling false positives when their assumptions are met. In TMM, a reference sample is first chosen, then, for each of the other samples, the A values (mean log counts across the two samples) and the M values (log fold change in scaled counts), are computed. The more extreme values are trimmed to extract the expression of non-DE genes. The normalization factor is the weighted mean of M values for each genes, where the weights correct the heteroscedasticity of M values. Finally, normalization factors are scaled to multiply one. As we have seen, TMM assumes that technical effects impact non-DE genes and DE genes alike, but another important assumption is symmetry: the numbers of up-regulated and down regulated genes should be roughly equal. Those are two assumptions that we can make in our context of comparison of expression between male and females samples.

2.5 Statistical analysis

2.5.1 Differentially expressed genes and annotation

To estimate the genewise differential expression we use the variance modeling at the observational level (voom) approach implemented in R package `limma`. The principle of voom is to adapt the well-establish `limma` t-test methodology for microarrays to the context of RNA-Seq data. Voom fits, for each gene, a linear model to estimate the genewise log fold change in cpm values and an empirical Bayes method to moderate the standard errors of the estimated log fold change.

We have a two independent groups design with one group composed of 3 human males and one group composed of 3 human females. There is one factor, the sex with two levels: male and female. We then build a 6x2 design matrix, where 6 is the number of samples we have and 2 the number of levels of the factor sex. Each column of the design matrix is a dummy variable that tells us which of the two groups the sample belongs to, that is the level of our factor. In such parametrization of the design, the model has no intercept for it to be identifiable. For each gene, the log-counts per million (log-cpm) value for the gene in sample i is:

$$y_i = \log_2(\frac{c_i + 0.5}{L_i + 1}10^6)$$

where c_i is the normalized count for the gene in sample i, and L_i is the library size in sample i.

The counts are offset away from zero by 0.5 to avoid taking the log of zero, and to reduce the variability of log-cpm for low expression genes. The library size is offset by 1 to ensure that $\exp(y_i)$ is strictly less than 1 as well as strictly greater than zero.

For each gene, the base linear model is then:

$$y_i = \beta_F F_i + \beta_M M_i$$

where F is a dummy variable that takes value 1 if the sample is female, 0 otherwise; M is a dummy variable that takes value 1 if the sample is male, 0 otherwise;

Arrays 1, 2, 3 are female samples whiles arrays 4, 5, 6 are male samples. The model above corresponds to the parametrization with base design matrix D.

$$D = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

In this parametrization, β_F and β_M are coefficients that can be interpreted as the effect on log-cpm of belonging to each group (F female, M male). In this analysis, we are looking to contrast the effect of sex, females against males. We want to test, for each gene, the null hypothesis:

$$H_0 : \quad \beta_F - \beta_M = 0$$

Our contrast matrix C is then a single vector C.

$$C = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

In a subsequent stage of the analysis we change our base design matrix to take into account the presence of batch effect in our data as detailed in Section 4.2.

One of the main hypothesis of linear models is homoscedasticity. However, RNA-Seq data are counts and feature heteroscedasticity. Typically, large counts have much larger standard deviations than small counts and as a consequence log-cpm values cannot be treated as having constant variances. To solve this issue, voom models the mean-variance relationship to obtain precision weights. The weight are input in the linear model fitting and eliminate the effect of the mean-variance relationship in the log-cpm values. In voom, the weights are the results of modelling the mean-variance trend of the log-cpm values at the individual observation level, while the related limma-trend approach models the variance at the gene level.

The weight is the inverse of the predicted variance for the observation such that, observations with larger estimated variance have less weight than observation with smaller estimated variance. They are precision weight, and reflect the fact that observations with high variance are less reliable. The predicted variance is estimated in several steps that involve:

- the fitting of a LOWESS model for the gene-wise mean-variance relationship based on the residual variance of gene-wise linear models on normalized counts.
- the estimation of each observation standard deviation using the gene-wise linear model prediction and the fitted mean-variance model.

From the fitted linear models we obtain for each gene, the estimated log-fold change, the moderated t-statistics and their corresponding raw p-values. We adjust the raw p-values for multiple testing by the Benjamini-Holmberg method that minimizes the false discovery rate (FDR). We then discuss various methods to define a list of differentially expressed genes that use a combination of the estimated log-fold changes, raw and adjusted p-values (see Section 4.3). We also annotate our genes with the Entrez ID, Symbol and gene name using R package `org.Hs.eg.db`. In this annotation process, we filter out tags that contain multiple matches.

2.5.2 Biological significance

Finally, to get a further insight into the biology behind our list of differentially expressed genes, we carry an over-representation analysis on Gene Ontology (GO) terms. In such analysis, the frequencies of GO terms associated to our differentially expressed genes are compared to the frequencies of those terms in the universe of genes (human genes in our case) in the GO database. We identify the over-represented GO terms for

up-regulated and down-regulated genes separately, and for each of the GO domain: biological process, cellular component and molecular function.

3 Software and versions

The versions of R and R packages we use are: R 3.6.3, limma 3.42.2, DESeq2 1.26.0, org.Hs.eg.db 3.10.0, AnnotationDbi 1.48.0, affy 1.64, Biobase 2.46.0, BiocGenerics 0.32.0, edgeR 3.28.1, DEFormats 1.14.0, GenomeInfoDb 1.22.1, GenomicRanges 1.38.0 and BiocParallel 1.20.1.

4 Results and discussion

4.1 Data

```
# data
load("dataset5_eset.RData")
seqdata <- read.table("dataset5_count_table.txt", header=TRUE, sep="\t")

# Pheno data
sampleinfo <- gilad.eset@phenoData@data
features_list <- gilad.eset@featureData@data

rownames(sampleinfo) <- c("HSF1","HSF2","HSF3","HSM1","HSM2","HSM3")

# Remove first two columns from seqdata
countdata <- seqdata[,-1]

# Store ENSEMBL GeneID as rownames
rownames(countdata) <- seqdata[,1]

# Rename samples
colnames(countdata) <- c("HSF1","HSF2","HSF3","HSM1","HSM2","HSM3")
```

We read the transformed dataset of raw annotated counts from the RData file `dataset5_eset`. The counts dataset has 52580 rows, each corresponding to an ENSEMBL gene and 6 columns, each corresponding to a sample.

4.2 Filtering, quality control and normalization

Filtering

```
# Obtain CPMs : Calculate counts-per-million (cpm) (not log scale)
myCPM <- cpm(countdata)

# threshold based on a raw count of 10
thresh_val <- 10*1000000/min(colSums(countdata))
thresh <- myCPM > thresh_val

# average between 1 and threshold based on a raw count of 10
tresh_val2 <- mean(c(1,thresh_val))
thresh2 <- myCPM > tresh_val2
```

```
# we would like to keep genes that have at least 3 TRUES in each row of thresh
keep <- rowSums(thresh2) >= 3
# Subset the rows of countdata to keep the more highly expressed genes
counts.keep <- countdata[keep,]
```

To define the threshold for cpm values, we combine two frequently used approaches to filtering on cpm. One of them is setting the cpm threshold at 1, $t_{cpm,1} = 1$. The other one is to define a threshold that represents a raw count of about 10 for the sample with the smallest library size in the dataset:

$$t_{cpm,2} = \frac{10 \times 10^6}{\min_j \{L_j\}}$$

where L_j is the library size of sample j.

We obtain $t_{cpm,2} = 7.2$. Filtering by a value as high as 7.2 might remove a lot of genes. To minimize the loss of genes, we use the mean of $t_{cpm,1}$ and $t_{cpm,2}$ as our initial threshold value: $t_{cpm} = 4.1$. Moreover, we wish to analyze genes that are at least completely expressed in one of the groups in our design. We have a two-group design with 3 samples in each group, so we keep genes which cpm values larger than our threshold in at least 3 samples. This initial filtering removes 46 353 genes and keeps 6227.

Quality control

```
# DEG object
y <- DGEList(counts.keep)
```

```
################################
# Quality control on raw counts
################################

##### library size #####

### bar plot
barplot(y$samples$lib.size,names=colnames(y),las=2,
        main = "Library sizes before normalization")

### MD plots
# Get log2 counts per million
logcounts <- cpm(y,log=TRUE)
# plots
op <- par()
par(mfrow=c(1,2))
plotMD(logcounts,column = 1, main=colnames(y)[1])
abline(h=0,col="red")
  plotMD(logcounts,column = 2, main=colnames(y)[2])
abline(h=0,col="red")
par(mfrow=c(1,2))
plotMD(logcounts,column = 3, main=colnames(y)[3])
abline(h=0,col="red")
plotMD(logcounts,column = 4, main=colnames(y)[4])
abline(h=0,col="red")
par(mfrow=c(1,2))
plotMD(logcounts,column = 5, main=colnames(y)[5])
abline(h=0,col="red")
plotMD(logcounts,column = 6, main=colnames(y)[6])
abline(h=0,col="red")
```

```
par(op)
```

Before normalization, the differences in library sizes are up to a 39%, between the largest and the smallest, which generates a composition bias as illustrated in the MD plots in Fig. 1. In the MD plots, the log-fold change in cpm values of each sample (y axis) is represented against the average log-expression across the dataset (x axis). The samples HSF1, HSF2 and HSM3 are not centered at 0 but shifted at higher values. The sample HSF3 is not centered at 0 either but shifted at a lower value. In those samples, the counts are biased upward and downward respectively in comparison to the average counts. The MD plots for HSM1 and HSM2 are not reported here because they did not show any bias. We do not reproduce the bar plot of total library sizes for the sake of conciseness.

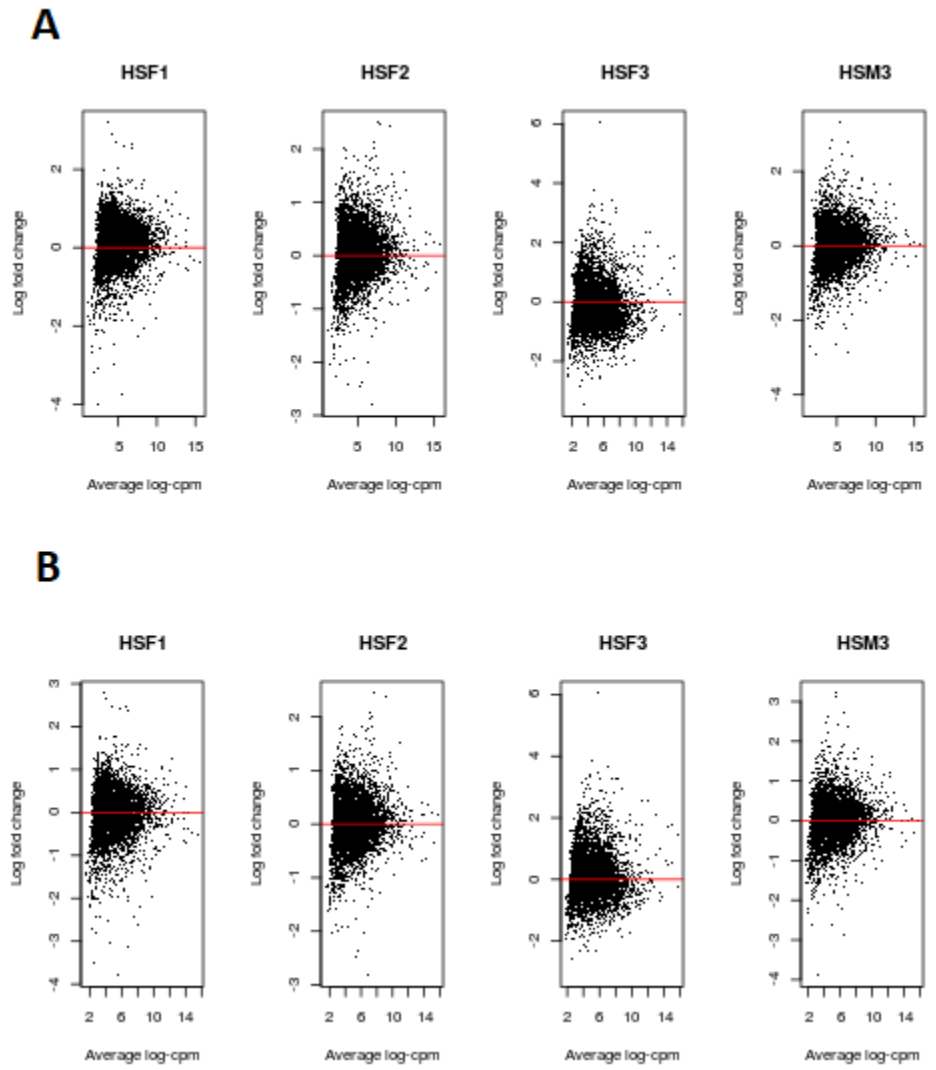

Figure 1: A. MD plots before normalization B. MD plots after normalization

```
#### homogeneity of samples distibrutions ####

### Boxplots
boxplot(logcounts, xlab="", ylab="Log2 counts per million",las=2,
        main = "Boxplots of logCPMs before normalization")
abline(h=median(logcounts),col="blue")
```

```
### MA plots
my_maplot<-function(x,y){
  ## M-values
  M <- x - y
  ## A-values
  A <- (x + y)/2
  df <- data.frame(A, M)
  g <- ggplot(df, aes(x = A, y = M)) + geom_point(size = 1.5, alpha = 1/5) +
    geom_hline(yintercept=0,color = "blue3") + stat_smooth(se = FALSE, method = "loess", color = "red3")
  return(g)
}

MA_females <- list()
comb_fem <- combn(seq(1:3),2)

for(i in 1:ncol(comb_fem)){
  MA_females[[i]] <- my_maplot(logcounts[,comb_fem[1,i]],logcounts[,comb_fem[2,i]]) +
    ggtitle(paste(colnames(logcounts)[comb_fem[1,i]],"vs",
                  colnames(logcounts)[comb_fem[2,i]]),"before normalization")
}

MA_males <- list()
comb_mal <- cbind(c(4,5),c(4,6),c(5,6))

for(i in 1:ncol(comb_mal)){
  MA_males[[i]] <- my_maplot(logcounts[,comb_mal[1,i]],logcounts[,comb_mal[2,i]]) +
    ggtitle(paste(colnames(logcounts)[comb_mal[1,i]],"vs",
                  colnames(logcounts)[comb_mal[2,i]]),"before normalization")
}
```

As for the homogeneity of counts distributions, before normalization the boxplots of the samples distributions did not show major differences between samples and are not reported here. However, the MA plots in Fig.2 show the HSF3 sample is not comparable to the rest of the female samples and needs normalization. Indeed, in MA plots the log-fold change in cpm values between two samples (y axis) is plotted against the log-average of cpm values across the two samples (x axis) and if the distributions of the two samples are comparable, the cloud of points is centered at 0 with no trend along the x axis. The MA plots of HSF3 against other female samples exhibit a trend that is first increasing and then decreasing. The rest of the MA plots did not show any issue and are not reported here.

```
#### batch effect ####

### color code
col.gender <- c("purple","orange")[sampleinfo$gender]

### Multidimensional scaling plots (MDS)
png("MDS_before_norm.png")
plotMDS(y,col=col.gender, labels = colnames(y),
        main = "")
legend("topleft",fill=c("purple","orange"),legend=levels(sampleinfo$gender), horiz = TRUE, x="top")
dev.off()

### Gene clustering
# variance for each row in the logcounts matrix
var_genes <- apply(logcounts, 1, var)
```

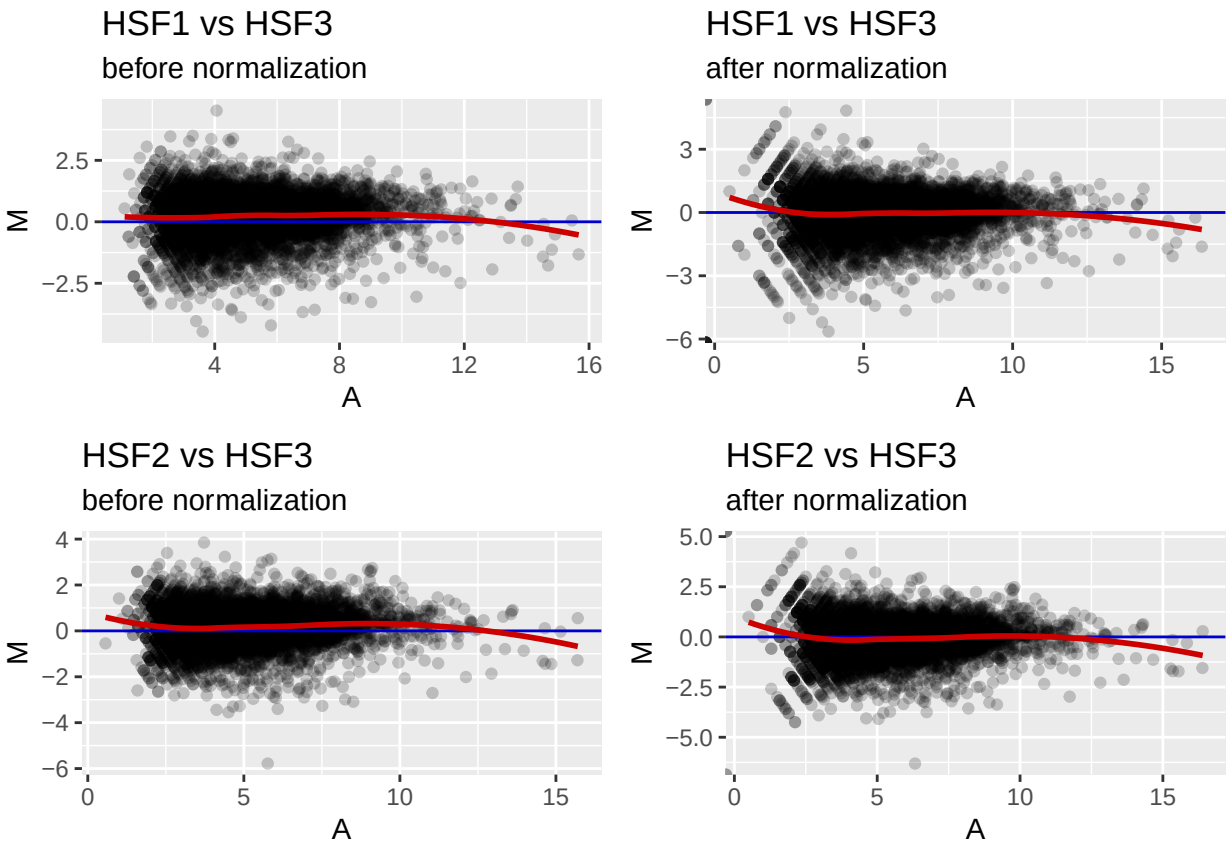

Figure 2: MA plots before and after normalization

```
# Get the gene names for the top 500 most variable genes
select_var <- names(sort(var_genes, decreasing=TRUE))[1:500]

# Subset logcounts matrix
highly_variable_lcpm <- logcounts[select_var,]

# color palette
mypalette <- brewer.pal(11,"RdYlBu")
morecols <- colorRampPalette(mypalette)

# heatmap
heatmap.2(highly_variable_lcpm,col=rev(morecols(50)),trace="none", ColSideColors=col.gender,scale="row"
          main = "Heatmap of over the top 500 most variable genes across samples before normalization")
dev.off()

### Clustering on pairwise distances
# distance matrix between samples
mat.dist <- as.matrix(dist(t(as.matrix(y))))
mat.dist <- mat.dist/max(mat.dist)
hmcol <- colorRampPalette(brewer.pal(9, "GnBu"))(16)

# heatmap
png("distmat_cluster_bn.png")
```

```
heatmap.2(mat.dist,col=hmcol,trace="none", ColSideColors=col.gender,
          RowSideColors=col.gender, dendrogram = c("both"),
          main = "")
dev.off()

### PCA on the the top 500 genes
pr <-data.frame(prcomp(t(highly_variable_lcpm),scale. = TRUE)$x)
ggplot(pr) +
        geom_point(aes(x=PC1,y=PC2,color=sampleinfo$gender)) +
        geom_label(aes(x=PC1,y=PC2,label=colnames(y)),nudge_x=2,nudge_y=2) +
        labs(colour="Group") +
        xlim(c(-35,60)) +
        theme(legend.position = "bottom") +
        ggtitle("Principal Component Analysis before normalization")
```

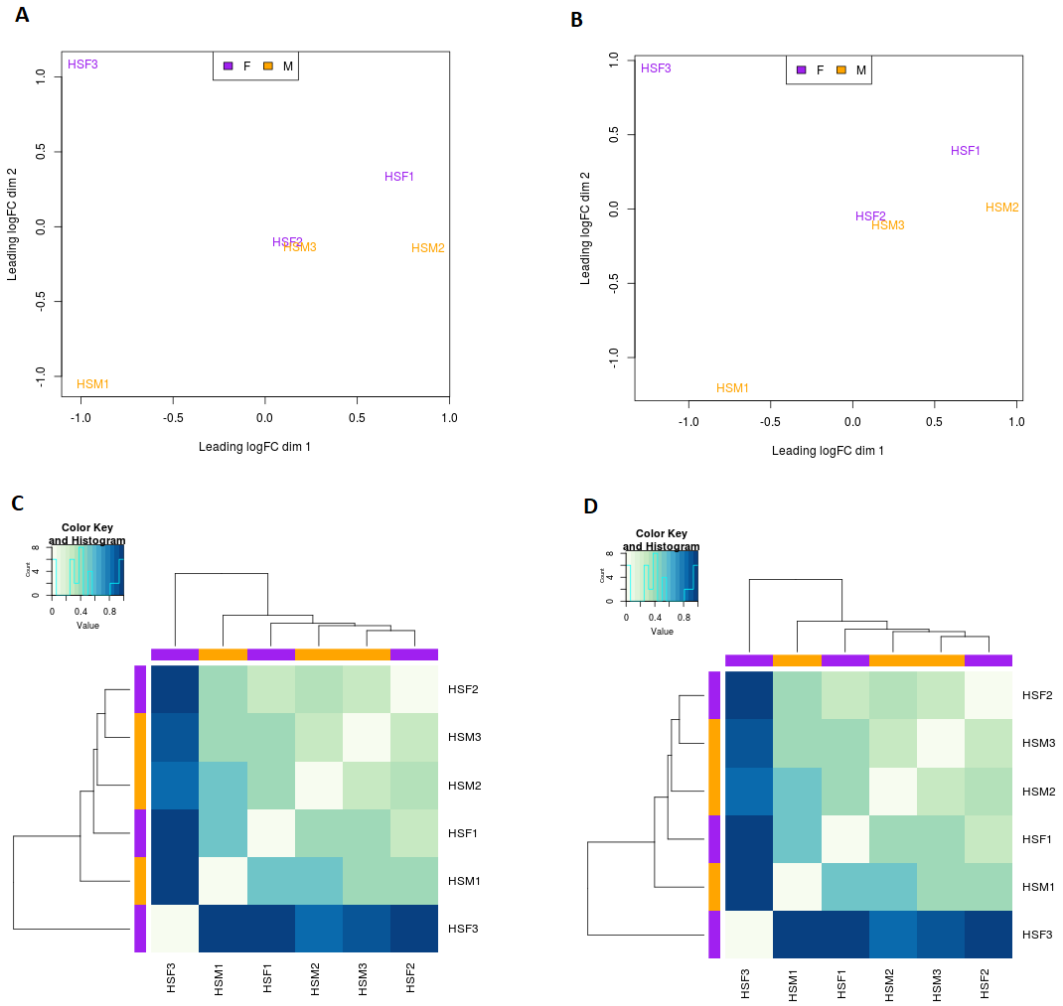

Figure 3: A. MDS plot before normalization B. Heatmap and clustering before normalization C. MDS plot after normalization D. Heatmap and clustering after normalization

Finally, all the assessment regarding batch effects lead to the same conclusions before normalization: the samples of the same sex are not more similar to each other than to samples of the other sex, sex is not the

main factor of variation of counts across samples. This result is illustrated in Fig.3. The MDS plot provides a visual representation of the matrix of pairwise distances between the samples. Samples that are close in the MDS plot are similar is the sense that the distance between the two in the original data is small. The pattern of similarities between samples in our data does not follow our sex groups. Samples such as HSF2 and HSM3 which are of different sexes are very close to each other, then similar, but far from other samples of their respective groups. The samples HSF3 and HSM1 appear far apart from the rest of the samples, suggesting they are not similar to the rest of the samples. In Fig.3 the hierarchical clustering on the euclidean distance matrix of the samples show the samples do no cluster by sex: females samples are not closer to other female samples than to male samples. Again HFS2 and HMS3 are very close while HFS3 and HSM1 are further away from all the samples in the data. The PCA analysis and the clustering on the matrix of gene counts that we did using the top 500 genes with the largest row variances led to the same conclusions and are not reproduced here.

Normalization

```
# TMM normalisation to DGEList object
y_TMM<-calcNormFactors(y, method="TMM")

df<-data.frame(t(y_TMM$samples$norm.factors))
colnames(df) <- colnames(y)
row.names(df)<-"Factors"
```

We normalize our count data by TMM method with the function `calcNormFactors` from `edgeR`. The normalization factors are summarized in Table 1. Samples with factors above one are downscaled and those with factors below one are upscaled to remove the composition bias. The factors are above 1 for samples HFS1, HFS2 and HMS3, which is consistent with the upward bias we observed in the MD plots. The factor is below 1 for HFS3 and very close to one for HMS1 and HMS2, in line with our diagnostic in MD plots.

Table 1: TMM normalization factors

	HSF1	HSF2	HSF3	HSM1	HSM2	HSM3
Factors	1.089	1.023	0.877	1.003	1.005	1.015

In Fig. 1, we see that after the normalization, all the samples are centered at 0 in MD plots. In Fig.2, the MA plots of HSF3 with other females do no present significant trends after normalization. The normalization effectively removed the compositional bias and made the distribution of counts more homogeneous across female samples. However, as shown in Fig.3, the normalization did not make the samples of the same sex more similar to each other and the samples still do not cluster by sex. Sex is not the main factor of similarity between samples of the same sex, and not the main factor of variation between samples of different sex. In our data, factors that are not sex cause a larger variation in counts. In particular, we note the samples HFS3 and HSM1 are constantly flagged as dissimilar from the rest of the samples, suggesting the presence of batch effect for those two samples. As a consequence, we add a batch effect factor in our design matrix without interaction. The design matrix becomes:

```
# Add batch information to the sample information
sampleinfo$batch <- as.factor(c("A","A","B","B","A","A"))

# Group and batch variables
group <- sampleinfo$gender
batch <- sampleinfo$batch

# design withou batch
design_wo_batch <- model.matrix(~ 0 + group)
colnames(design_wo_batch)[1:2] <- levels(group)
# Specify a design matrix without an intercept term
```

```
design <- model.matrix(~ 0 + group + batch)

## Make the column names of the design matrix a bit nicer
colnames(design)[1:2] <- levels(group)

print(data.frame(design))
```

```
##   F M batchB
## 1 1 0      0
## 2 1 0      0
## 3 1 0      1
## 4 0 1      1
## 5 0 1      0
## 6 0 1      0
```

For each gene, the estimated linear model is now:

$$y_i = \beta_F F_i + \beta_M M_i + \beta_B B_i$$

where F and M are the same variables as before and B is a dummy variable that takes value 1 if the sample is sample HFS3 or HSM1, 0 otherwise.

The coefficient β_B can be interpreted as the batch effect. The interpretation of coefficients β_F and β_M does not change, therefore our contrast matrix only changes to adapt to the additional coefficient:

```
# contrast matrix without batch
cont.matrix_wo_batch <- makeContrasts(SexDiph=F-M,levels=design_wo_batch)

# contrast matrix with batch
cont.matrix <- makeContrasts(SexDiph=F-M,levels=design)
print(cont.matrix)
```

```
##         Contrasts
## Levels   SexDiph
##   F            1
##   M           -1
##   batchB       0
```

4.3 Statistical analysis

We run our limma-voom pipeline on our filtered and normalized count data and fit our linear models. In Fig.4, we plot the mean-variance relationship in raw counts, log2 counts and residuals from the linear model. At the top, we see how in filtered raw counts there is a strong heteroscedasticity, with an increasing trend of standard deviation in mean count. In the middle, we see that a log2 transformation of raw counts and normalization does not solve the heteroscedasticity issue : we now observe heteroscedasticity in the form of a decreasing quadratic trend that flattens for larger mean counts. The mean-variance trend estimated by voom in red fits the data well. We also note that there is little to no observations in the lower left corner, that is observations with small counts and small variance. Our independent filtering of lowly expressed genes has been effective. If filtering of lowly-expressed genes is insufficient, a drop in variance levels can be observed at the low end of the expression scale due to very small counts. Finally, at the bottom, we can appreciate how the models residuals do not feature heteroscedasticity anymore. The voom-limma pipeline has been effective in solving the heteroscedasticity issue.

We annotate our genes with the Entrez ID, Symbol and genename.

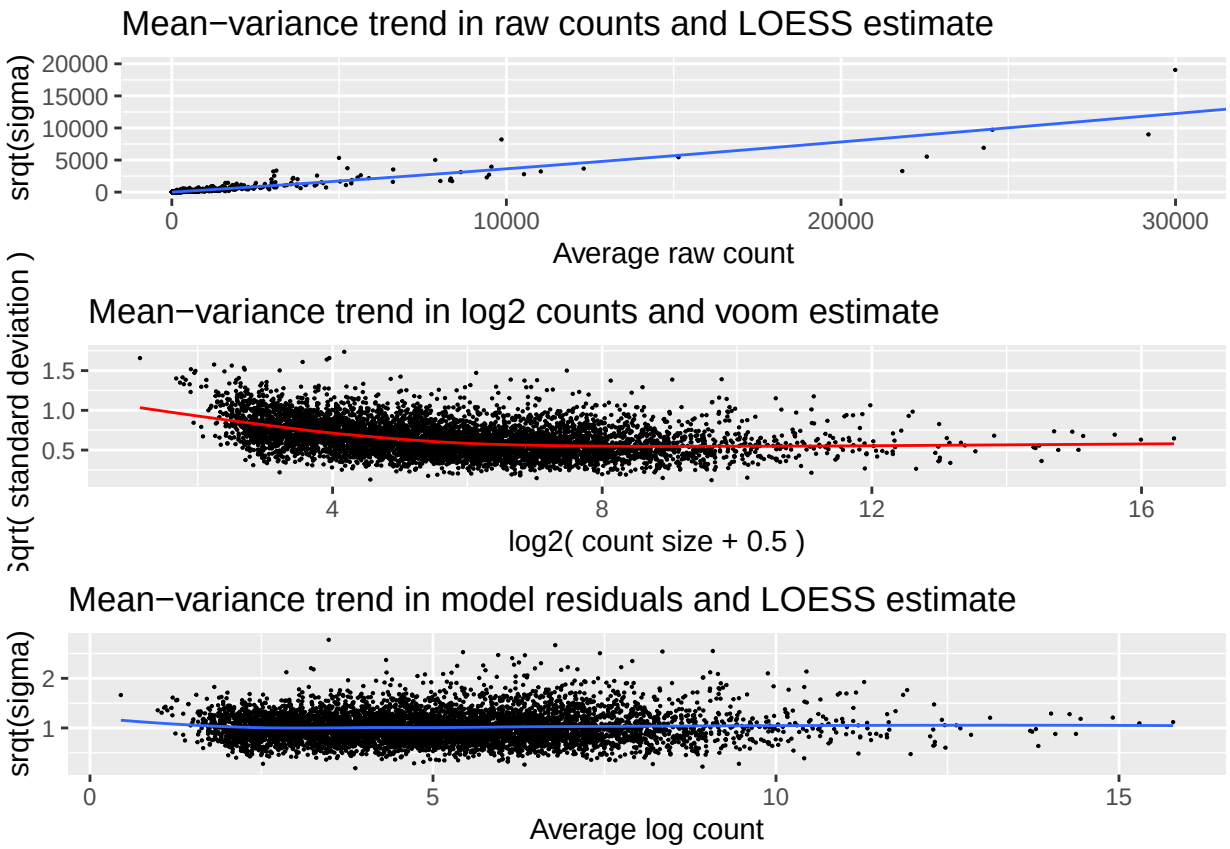

Figure 4: Mean-variance trend

```
ann <- mapIds(org.Hs.eg.db,keys=rownames(fit.cont),keytype = "ENSEMBL", column="ENTREZID", multiVals =
ann <- AnnotationDbi::select(org.Hs.eg.db,keys=ann,columns=c("ENTREZID","SYMBOL","GENENAME"))
fit.cont$genes <- ann
```

In Table 2 we list our top 6 genes sorted by the adjusted p-value. We see that all but two of the top genes have adjusted p-values well above any acceptable threshold. Using a 5% threshold on p-value, only two genes have a statistically significant differential expression between the two sexes. Those two genes are up-regulated in females.

```
top<-topTable(fit.cont,coef="SexDiph",sort.by="p", number=6)
summa.fit <- decideTests(fit.cont,p.value = 0.05)
```

Table 2: Top 6 genes by adjusted p-value

	ENTREZID	logFC	P.Value	adj.P.Val
ENSG00000163220	81788	2.098	0.000	0.048
ENSG00000143546	79169	2.854	0.000	0.048
ENSG00000171051	116844	1.567	0.000	0.475
ENSG00000118972	81932	5.287	0.000	0.622
ENSG00000049239	9563	-1.469	0.000	0.622
ENSG00000122862	NA	1.176	0.001	0.784

To get a better understanding of our results, we analyze the MA plots and volcano plots in Fig.5. In the MA plots, the x axis is the average expression in terms of average log normalized counts (A), the y axis is the log2 fold change between groups (M) which corresponds to the contrast estimated in our models. In volcano plots the y-axis is -log10 the non-adjusted p-value and the x-axis is the log fold change between groups. In the MA plot, we see that various genes have log folds change above 1 and, for a given level of average expression, stand away from the rest of genes but nevertheless they are not marked as significant. In the volcano plot we note that many of the genes with log fold changes superior to 1 also have p-values lower than 0.05. However, how we have seen above, only two genes are flagged as differentially expressed: the variance in our dataset is too large for those log fold changes to be significant after adjustment for multitesting. Our data is composed of biological replicates that cause a large biological noise. Such noise difficults the identification of differentially expressed genes by statistical tests.

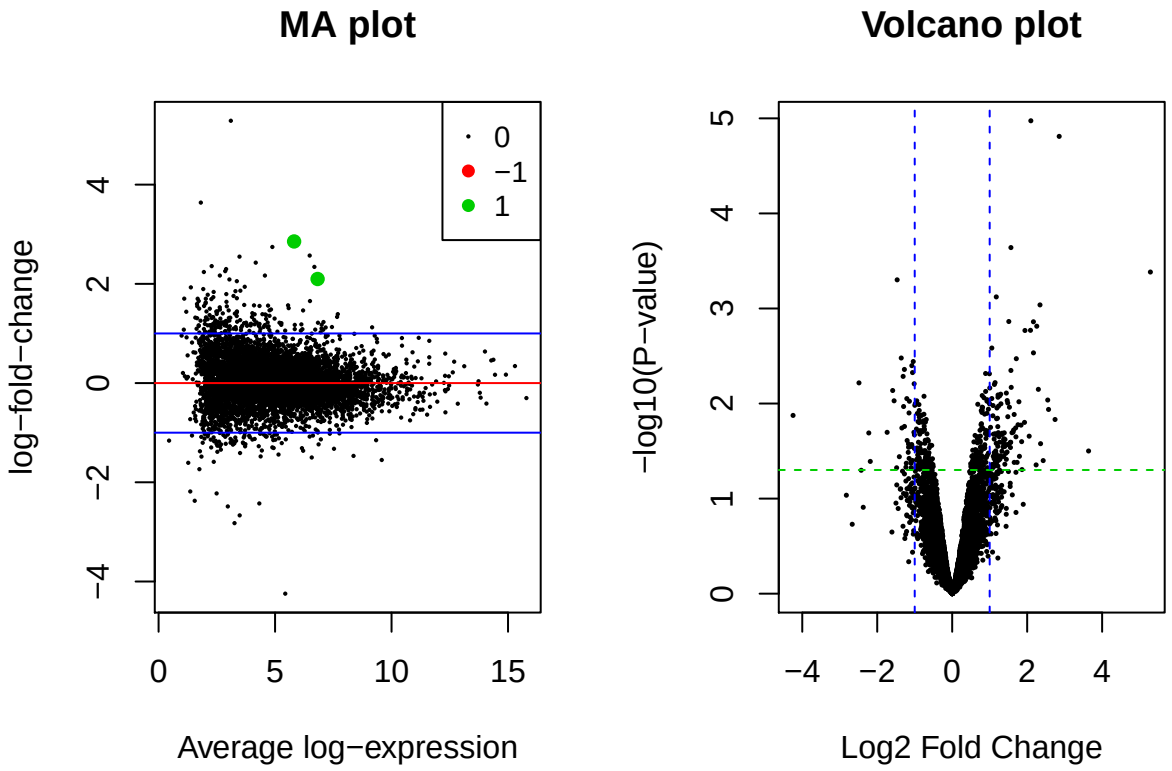

Figure 5: MA plot and volcano plot

To overcome this issue, we adopt a more empirical approach to define our list of differentially expressed genes, based only on the log fold change and the non-adjusted p-value. Considering the large noise we set a high threshold on the p-value, 5% and try different thresholds for the log-fold change. In addition, the initial independent filtering greatly reduced the number of genes and may have caused the loss of differentially expressed genes. We therefore also try different thresholds for the initial filtering on cpm values.

In Fig.6, we plot the number of differentially expressed genes obtained when varying the two threshold values. We see that for a set of pairs of thresholds, we obtain a list of between 100 and 200 DE genes. We pick three pairs of the set:

$$(t_{cpm}, t_{LFC}) \in \{(0,2),(5,1),(25,0.5)\}$$

Where t_{cpm} is the threshold for the cpm values and t_{LFC} is the threshold for the log-fold change.

This set of pairs contemplates scenarios of no filtering and important filtering, as well as high and low log-fold changes thresholds. Considering very low log fold changes makes sense in our context because we

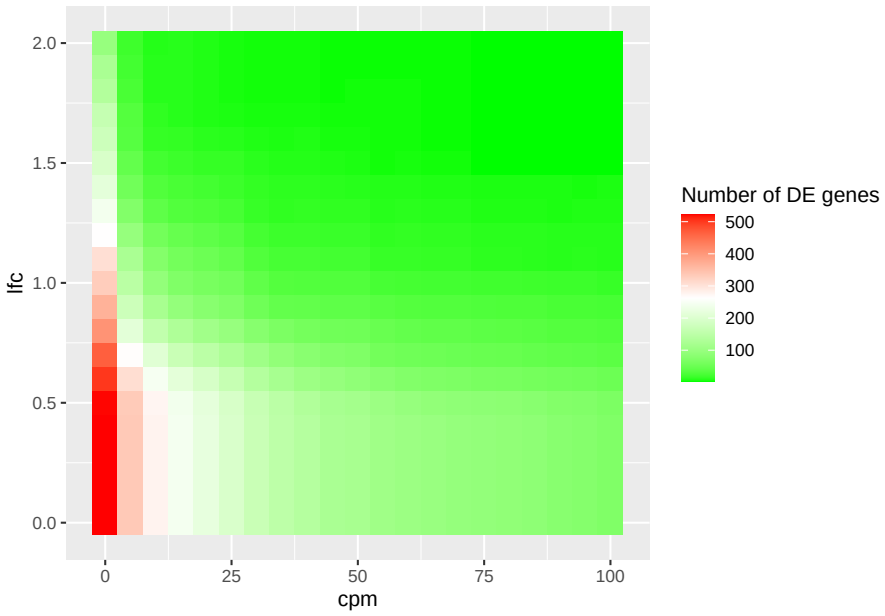

Figure 6: Number of DEG in function of cmp and log-fold chnage thresholds

are comparing samples that differ only by sex and should not expect large changes. For example, in a genome-wide estimation of gender differences in the gene expression of human livers, Delongchamp et al. (2005) found a largest observed fold-change at 1.55. Shen and Wang (2017) identified diffferentially expressed genes between sexes with log fold changes lower than 1.

Before analyzing the lists of DE genes, we check the voom plots produced when fitting the mean-variance trend. On the unfiltered counts ($t_{cpm} = 0$) the plot showed a lot of points in the bottom-left corner, corresponding to lowly expressed genes. However, the trend was still fitted correctly as there was no drop in estimated variance levels at the low end of log counts. The other two plots on filtered counts ($t_{cpm} > 0$) did not have points in the bottom-left corner and no issue in the trend estimate either. The plots are not reproduced here for the sake of conciseness.

We assess the performance of the three lists of DE genes by fitting in each case a hierarchical clustering on the matrix of normalized counts of the selected genes and the plotting the corresponding heatmap. The first two pairs performed poorly as the resulting clustering did not separate the samples by sex and we do not reproduce the plots here. The clustering based the on the third pair, with the lowest threshold on log fold change, correctly separates the samples by sex as can be observed in Fig.7. Those genes are differentially expressed between male and female samples. The pattern of expression of the list also characterizes each sex group. We select that list as our final list of differentially expressed genes and save it in the attached file `differentially_expressed_genes.csv`. The final list has 66 down regulated genes in females, and 125 up regulated genes.

4.4 Biological significance

We use the over-representation analysis as implemented in `goana` but without adjusting for genes length or abundance.

```
up <- sc3$statistics$logFC>0
down <- sc3$statistics$logFC<0

g_up<-goana(sc3$statistics$genes$ENTREZID[up],species="Hs")
g_down<-goana(sc3$statistics$genes$ENTREZID[down],species="Hs")
```

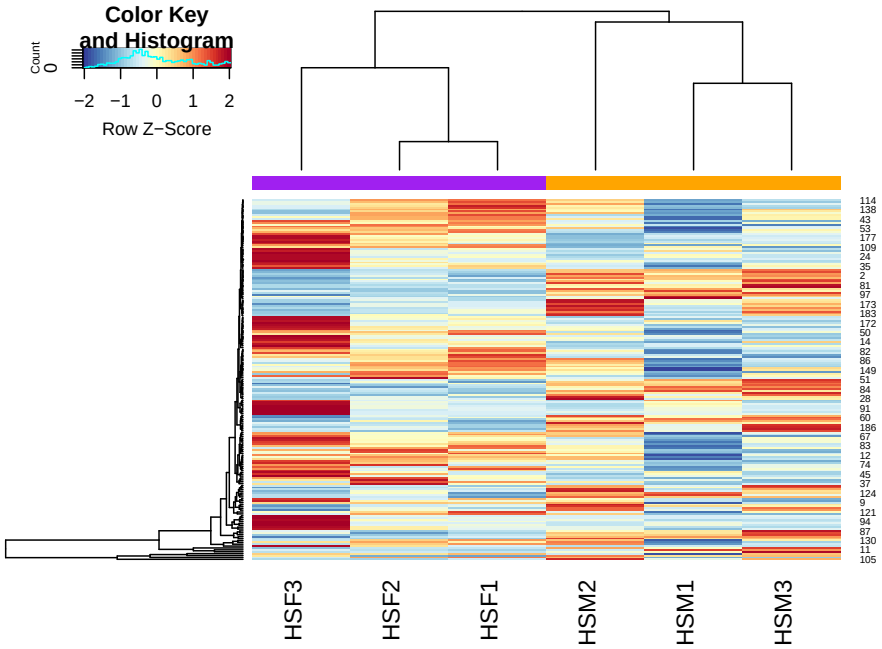

Figure 7: Hierarchical clustering on final list of differentially expressed genes

We obtain the top GO terms from the `goana` output with `topGO` function. We present the top 3 GO terms for up regulated and down regulated genes by domain in Tables 3 to 5. The full lists of GO terms by domain are available in attached files: `topGO_biologicalprocess.csv`, `topGO_molecularfunction.csv`,`topGO_cellularcomponent.csv`.

Table 3: Top 3 GO terms in biological process domain by regulation

	Term	N	DE	regulation
GO:0043312	neutrophil degranulation	485	16	up
GO:0002283	neutrophil activation involved in immune response	488	16	up
GO:0042119	neutrophil activation	498	16	up
GO:0031325	positive regulation of cellular metabolic process	3330	23	down
GO:0048522	positive regulation of cellular process	5490	31	down
GO:0031328	positive regulation of cellular biosynthetic process	1992	16	down

Table 4: Top 3 GO terms in molecular function domain by regulation

	Term	N	DE	regulation
GO:0050786	RAGE receptor binding	11	4	up
GO:0005509	calcium ion binding	705	15	up
GO:0035662	Toll-like receptor 4 binding	4	2	up
GO:0052689	carboxylic ester hydrolase activity	136	4	down
GO:0005543	phospholipid binding	427	6	down
GO:0051219	phosphoprotein binding	83	3	down

Table 5: Top 3 GO terms in cellular component domain by regulation

	Term	N	DE	regulation
GO:0070062	extracellular exosome	2163	43	up
GO:1903561	extracellular vesicle	2186	43	up
GO:0043230	extracellular organelle	2191	43	up
GO:0005811	lipid droplet	81	4	down
GO:0098590	plasma membrane region	1198	12	down
GO:0016324	apical plasma membrane	318	6	down

5 Conclusions

In this work, we showed that there are differences in gene expression between human males and females. We were also able to define a list of differentially expressed genes which expression pattern is characteristic of sexes. Those genes have small log-fold changes which confirms previous results that sexual dimorphism occurs at small levels of differential expression. Finally, we identified a list of biological process, molecular function and cellular component ontologies particularly associated with this defining list of genes. A reader better-versed in biology can now pick up those results to further understand the effects of the evidenced sexually-dimorphic expression patterns of genes.

Our statistical methodology suffered from the presence of batch effects and an important biological noise in our data. In future similar experiments, changes to the experimental design could help mitigate the impact of a large noise. For example, an increase in the number of samples or the inclusion of control genes. However, other statistical methods could be considered to deal with similarly noisy data. Firstly, in this work we used linear models that are based on the normal distribution to model our counts but the Poisson and the negative binomial distributions are also commonly used for this type of data. The negative binomial in particular could be well suited because it allows for overdispersion in the data which is characteristic of sets of biological replicates. Secondly, the inclusion of random effects in different stages of the statistical process could help control the noise. We considered a fixed batch effect for the two problematic samples, HSF3 and HSM1, but batch effects might differ in the two samples and that could be represented with random effects. In general, for all the samples, the biological noise could be modeled with individual random effect per sample. We note this is the approach adopted by Blekhman et al. (2010).

Bibliography

Blekhman, Ran, John C. Marioni, Paul Zumbo, Matthew Stephens, and Yoav Gilad. 2010. "Sex-Specific and Lineage-Specific Alternative Splicing in Primates." *Genome Res* 20(2): 180-189. https://doi.org/10.1101/gr.099226.109.

Delongchamp, Robert R., Dial Velasco Cruz, Stacey, and Angela J. Harris. 2005. "Genome-Wide Estimation of Gender Differences in the Gene Expression of Human Livers: Statistical Design and Analysis." *BMC Bioinformatics* 6, S13. https://doi.org/10.1186/1471-2105-6-S2-S13.

Evans, Ciaran, Johanna Hardin, and Daniel M. Stoebel. 2018. "Selecting Between-Sample Rna-Seq Normalization Methods from the Perspective of Their Assumptions." *Briefings in Bioinformatics* 19(5): 776–92. https://doi.org/10.1093/bib/bbx008.

Shen, Jiangshan J., and Wanling Wang Ting-You & Yang. 2017. "Regulatory and Evolutionary Signatures of Sex-Biased Genes on Both the X Chromosome and the Autosomes." *Biology of Sex Differences* 8, 35. https://doi.org/110.1186/s13293-017-0156-4.