RNA-Seq BMDM WT vs KO

Domien Vanneste

2024-12-09 10:46:07 +0100

Contents

Introduction

Two hundred fifty thousand bone marrow monocytes from Mafbfl/fl or Lyz2CreMafbfl/fl mice were isolated, seeded in tissue culture-treated 6-well plates (Greiner, 657160) and differentiated into BMDM, as described above. After 4 days of differentiation, non-adherent cells were washed off with ice-cold PBS and total RNA was isolated with the standard TRIzol (Invitrogen, 15596018) RNA extraction protocol. Total RNA was further purified and concentrated using a RNA Clean & Concentrator Kit (Zymo Research, R1013) according to manufacturer's instructions. RNA quality and quantity were evaluated using a 5200 Fragment Analyzer (Agilent). Samples with a RIN $>$ 9.9 were selected for sequencing. One hundred nanograms of RNA was used to generate the libraries using the TruSeq Stranded mRNA kit (Illumina, 20020594). These libraries were sequenced on an Illumina NovaSeq sequencer on an SP flow cell. Preprocessing, alignment to the mouse genome (GRCm38/mm10), sequence counting, and quality control of the bulk RNA-Seq data were carried out with the nf-core/rnaseq pipeline. RNA-Seq data were further analyzed using R Bioconductor (4.2.3) and the DESeq2 package (1.38.3).

```
suppressMessages({
    library(DESeq2)
    library(ggplot2)
    library(pheatmap)
    library(RColorBrewer)
    library(apeglm)
    library(readxl)
    library(ggrepel)
    library(dplyr)
    library(ComplexHeatmap)
})
```

Counting from fastq data using nf-core/rnaseq pipeline

The following codes were used to do the mapping and counting.

```
nextflow run nf-core/rnaseq --input samplesheet.csv --fasta GRCm38/fasta/genome.fa --gtf GRCm38/genes/g
```

`samplesheet.csv` is text file with 4 columns: sample, fastq_1, fastq_2 and strandedness. Prepared following to the software's instructions.

Counts data processing

```
COUNTS <- read.table("salmon.merged.gene_counts.tsv", sep = "\t",
    header = T, row.names = NULL)

dim(COUNTS)
```

```
## [1] 22597    10
```

```
Genes <- COUNTS$gene_name

rownames(COUNTS) = make.names(Genes, unique = TRUE)
```

```
COUNTS <- COUNTS[, -c(1, 2)]

COUNTS <- round(COUNTS, digits = 0)

head(COUNTS, 3)
```

```
##       KO.F_1 KO.F_2 KO.M_1 KO.M_2 WT.F_1 WT.F_2 WT.M_1 WT.M_2
## Gnai3   2949   3498   2859   3495   3813   3370   3163   3546
## Pbsn       0      0      0      0      0      0      0      0
## Cdc45    657    811    585    821    942    798    642    678
```

```
COUNTS <- COUNTS[, c(5, 6, 7, 8, 1, 2, 3, 4)]

head(COUNTS, 3)
```

```
##       WT.F_1 WT.F_2 WT.M_1 WT.M_2 KO.F_1 KO.F_2 KO.M_1 KO.M_2
## Gnai3   3813   3370   3163   3546   2949   3498   2859   3495
## Pbsn       0      0      0      0      0      0      0      0
## Cdc45    942    798    642    678    657    811    585    821
```

```
write.csv(COUNTS, file = "COUNTS.csv")
```

Make metadata for bulkRNAseq samples

```
SampleSheet <- data.frame(genotype = c(rep("WT", 4), rep("KO",
    4)), gender = c(rep(rep(c("female", "male"), each = 2), 2)),
    replicate = c(rep(rep(c("1", "2")), 4)))

rownames(SampleSheet) <- colnames(COUNTS)

SampleSheet
```

```
##        genotype gender replicate
## WT.F_1       WT female         1
## WT.F_2       WT female         2
## WT.M_1       WT   male         1
## WT.M_2       WT   male         2
## KO.F_1       KO female         1
## KO.F_2       KO female         2
## KO.M_1       KO   male         1
## KO.M_2       KO   male         2
```

DESeq2 analysis

```
dds <- DESeqDataSetFromMatrix(countData = COUNTS, colData = SampleSheet,
    design = ~genotype + gender)
```

```
## converting counts to integer mode
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
dds
```

```
## class: DESeqDataSet
## dim: 22597 8
## metadata(1): version
## assays(1): counts
## rownames(22597): Gnai3 Pbsn ... AC112683.2 BX571804.1
## rowData names(0):
## colnames(8): WT.F_1 WT.F_2 ... KO.M_1 KO.M_2
## colData names(3): genotype gender replicate
```

Perform rlog transformation for distances and PCA

```
# keep only genes with more than a single read
dds <- dds[rowSums(counts(dds)) > 1, ]
# perform rlog transformation for distances (for
# clustering) and PCA
rld <- rlog(dds)
```

```
dds <- dds[rowSums(counts(dds)) > 1, ]
nrow(dds)
```

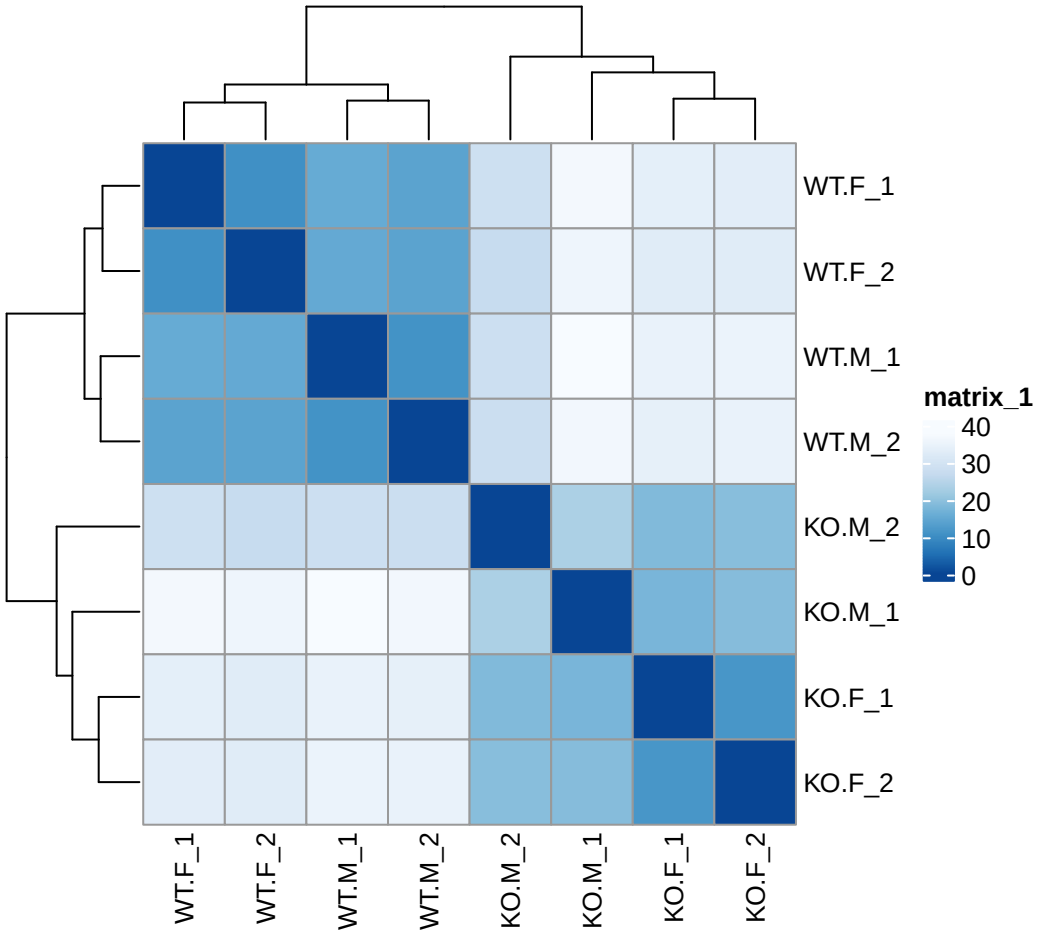
```
## [1] 13899
```

Calculate sample-to-sammple distances

```
sampleDists <- dist(t(assay(rld)))
sampleDistMatrix <- as.matrix(sampleDists)
```

Heatmap

```
colors <- colorRampPalette(rev(brewer.pal(ncol(COUNTS), "Blues")))(255)
heatmap <- pheatmap(sampleDistMatrix, clustering_distance_rows = sampleDists,
    clustering_distance_cols = sampleDists, col = colors)
heatmap
```

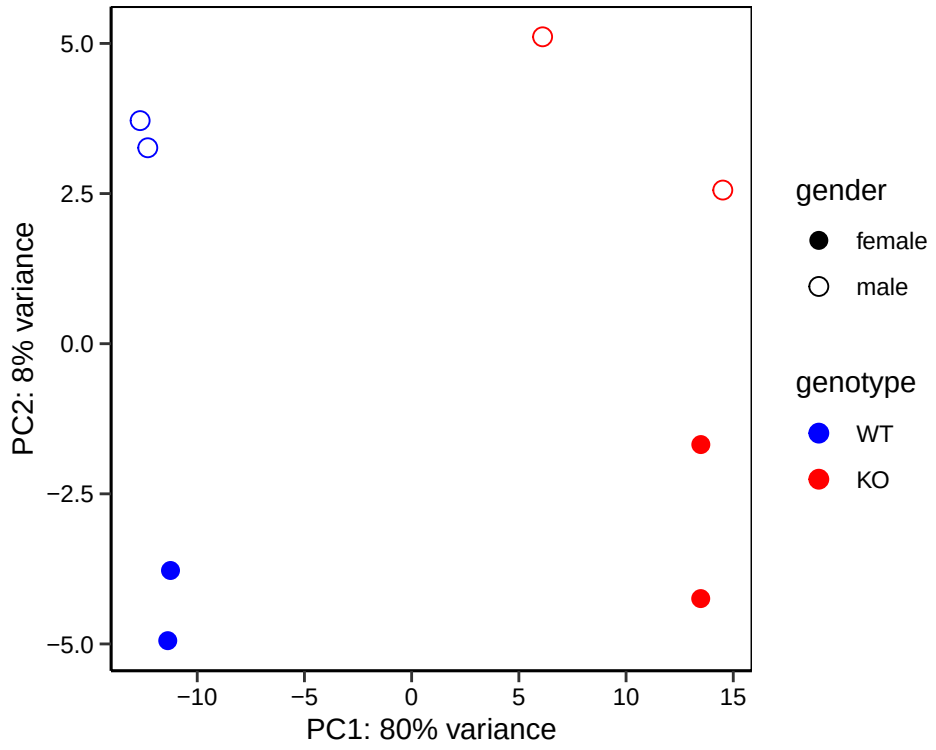


PCA analysis

Calculate PCs:

```
PCA <- plotPCA(rld, intgroup = c("genotype", "gender"), returnData = TRUE)
```

```
percentVar <- round(100 * attr(PCA, "percentVar"))

ggplot(PCA, aes(PC1, PC2)) + geom_point(size = 3, aes(color = genotype,
    shape = gender)) + scale_color_manual(breaks = c("WT", "KO"),
    values = c("blue", "red")) + scale_shape_manual(breaks = c("female",
    "male"), values = c(16, 1)) + xlab(paste0("PC1: ", percentVar[1],
    "% variance")) + ylab(paste0("PC2: ", percentVar[2], "% variance")) +
    theme_classic() + theme(axis.text.x = element_text(color = "black"),
    axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
        "cm"), panel.border = element_rect(fill = NA, color = "black",
        linetype = "solid"))
```



DEG analysis

```
dds <- DESeq(dds)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
resultsNames(dds)
```

```
## [1] "Intercept"             "genotype_WT_vs_KO"     "gender_male_vs_female"
```

```
dds$genotype <- relevel(dds$genotype, ref = "WT")
resultsNames(dds)
```

```
## [1] "Intercept"             "genotype_WT_vs_KO"     "gender_male_vs_female"
```

```
dds <- DESeq(dds)
```

```
## using pre-existing size factors

## estimating dispersions

## found already estimated dispersions, replacing these

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

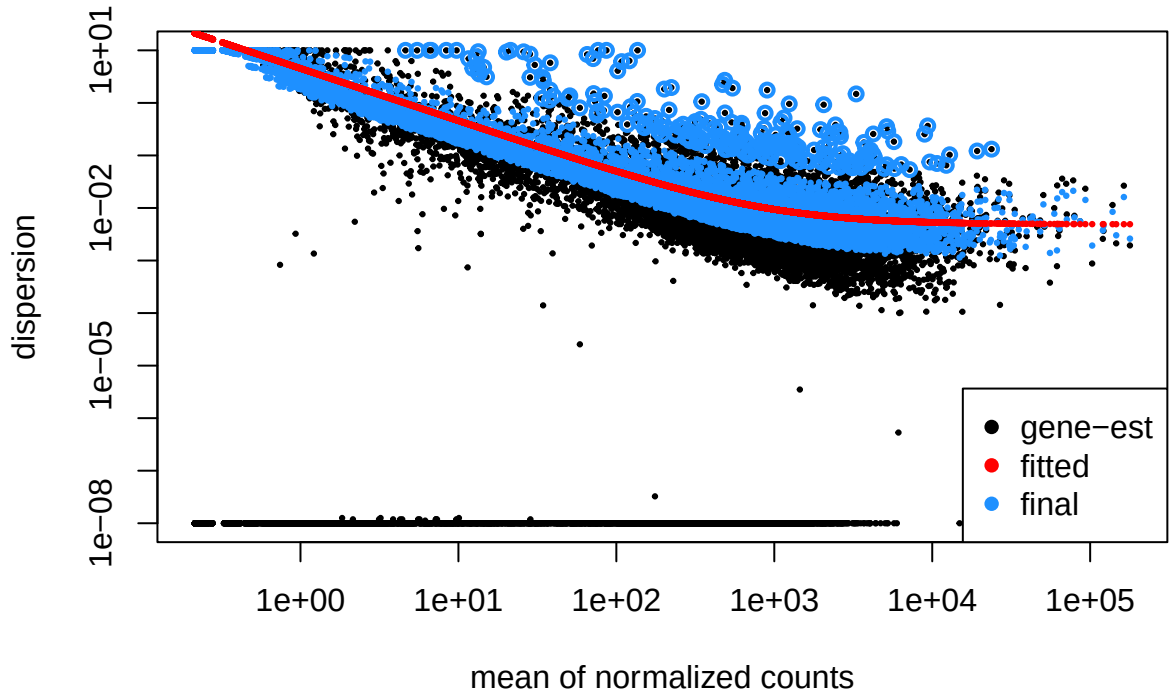
```
resultsNames(dds)
```

```
## [1] "Intercept"             "genotype_KO_vs_WT"     "gender_male_vs_female"
```

```
dds$genotype <- relevel(dds$genotype, ref = "KO")
resultsNames(dds)
```

```
## [1] "Intercept"             "genotype_KO_vs_WT"     "gender_male_vs_female"
```

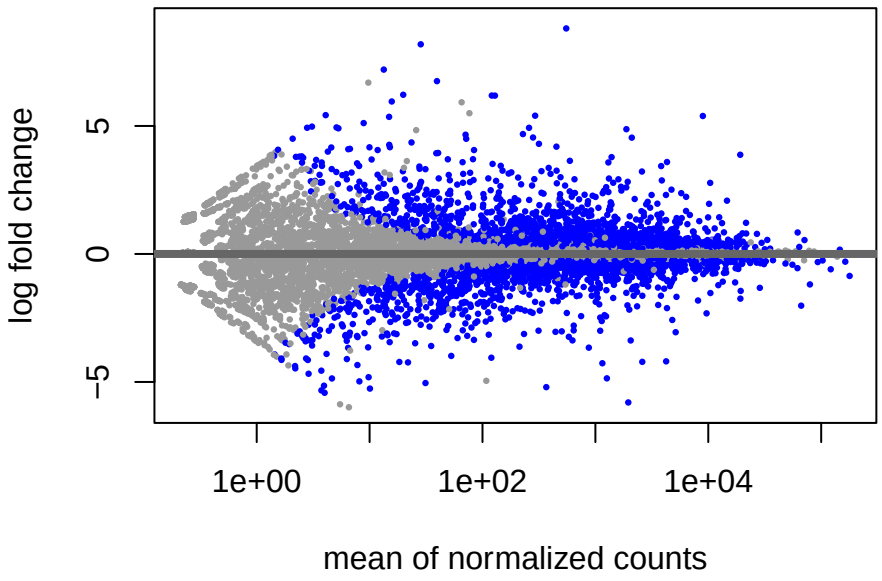
```
plotDispEsts(dds)
```



```
results <- results(dds, name = "genotype_KO_vs_WT")
summary(results)
```

```
##
## out of 13899 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 2633, 19%
## LFC < 0 (down)     : 2512, 18%
## outliers [1]       : 0, 0%
## low counts [2]     : 1348, 9.7%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
plotMA(results, ylim = c(-6, 9))
```



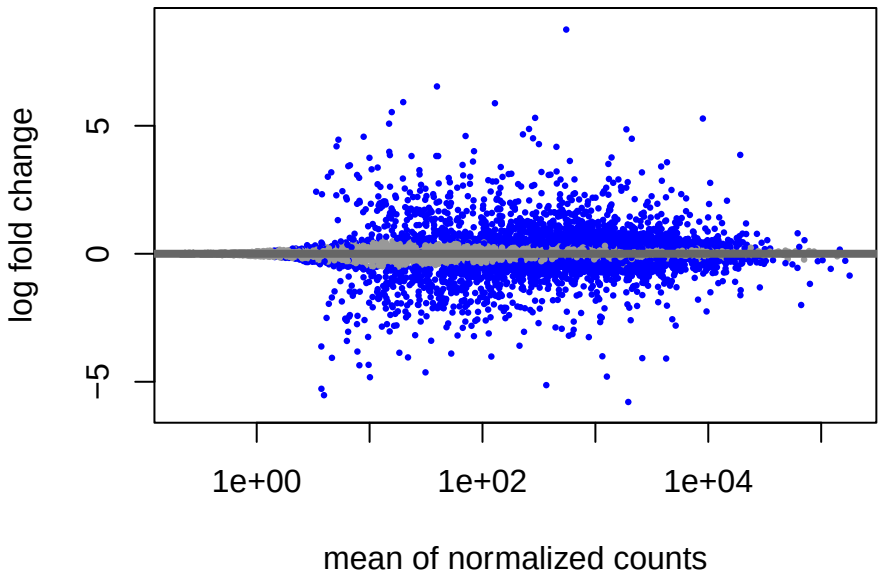

```
results_shrunk <- lfcShrink(dds, coef = "genotype_KO_vs_WT",
    type = "apeglm")
```

```
## using 'apeglm' for LFC shrinkage. If used in published research, please cite:
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
##     sequence count data: removing the noise and preserving large differences.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895
```

```
results_shrunk$log2FoldChange <- results_shrunk$log2FoldChange *
    -1
```

```
deg <- merge(x = as.data.frame(results), y = as.data.frame(results_shrunk),
    by = c(0, 1))
```

```
plotMA(results_shrunk, ylim = c(-6, 9))
```




```
Genes2 <- deg$Row.names

rownames(deg) = make.names(Genes2, unique = TRUE)

deg <- deg[, -1]
```

Filter

```
deg <- deg[!is.na(deg$padj.y), ]

deg_WT <- deg[deg$log2FoldChange.y < -1 & deg$padj.y < 0.05,
    ]
deg_KO <- deg[deg$log2FoldChange.y > 1 & deg$padj.y < 0.05, ]

deg_significant <- rbind(deg_WT, deg_KO)
```

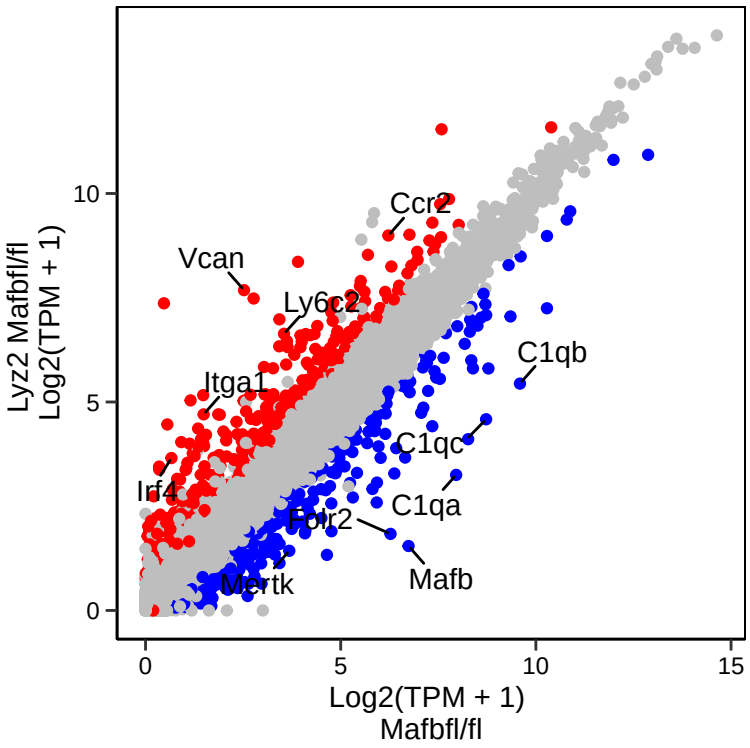
```
deg_ordered <- deg_significant[order(deg_significant$log2FoldChange.y),
    ]
```

```
write.csv(deg_ordered, file = "DEG.csv")
```

Expression plot

```
COUNTS_LogTPM <- read_excel("COUNTS_LogTPM.xlsx")
COUNTS_LogTPM <- as.data.frame(COUNTS_LogTPM)


rownames(COUNTS_LogTPM) <- make.names(COUNTS_LogTPM$gene_name,
    unique = TRUE)

COUNTS_LogTPM$diffexpressed <- "Not significant"

KO <- rownames(deg_significant[deg_significant$log2FoldChange.y >
    1, ])
WT <- rownames(deg_significant[deg_significant$log2FoldChange.y <
    -1, ])

COUNTS_LogTPM$diffexpressed[COUNTS_LogTPM$gene_name %in% WT] <- "WT"
COUNTS_LogTPM$diffexpressed[COUNTS_LogTPM$gene_name %in% KO] <- "KO"


COUNTS_LogTPM$label <- NA

Mo_signature_genes <- c("Ccr2", "Vcan", "Ly6c2", "Irf4", "Itga1")

Mac_signature_genes <- c("Mafb", "C1qa", "C1qb", "C1qc", "Mertk",
    "Folr2")

COUNTS_LogTPM$label[COUNTS_LogTPM$gene_name %in% Mo_signature_genes] <- COUNTS_LogTPM$gene_name[COUNTS_
    Mo_signature_genes]

COUNTS_LogTPM$label[COUNTS_LogTPM$gene_name %in% Mac_signature_genes] <- COUNTS_LogTPM$gene_name[COUNTS
    Mac_signature_genes]

ggplot(data = COUNTS_LogTPM, aes(x = WT_Log2_mean, y = KO_Log2_mean,
    col = diffexpressed, label = label)) + geom_point(show.legend = FALSE) +
    geom_text_repel(min.segment.length = 0, box.padding = 0.5,
        size = 4, col = "black") + scale_color_manual(values = c("red",
    "grey", "blue")) + xlab("Log2(TPM + 1) \nMafbfl/fl") + ylab("Lyz2 Mafbfl/fl \nLog2(TPM + 1)") +
    theme_classic() + theme(axis.text.x = element_text(color = "black"),
    axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
        "cm"), panel.border = element_rect(fill = NA, color = "black",
        linetype = "solid"))
```

```
## Warning: Removed 22586 rows containing missing values (‘geom_text_repel()‘).
```



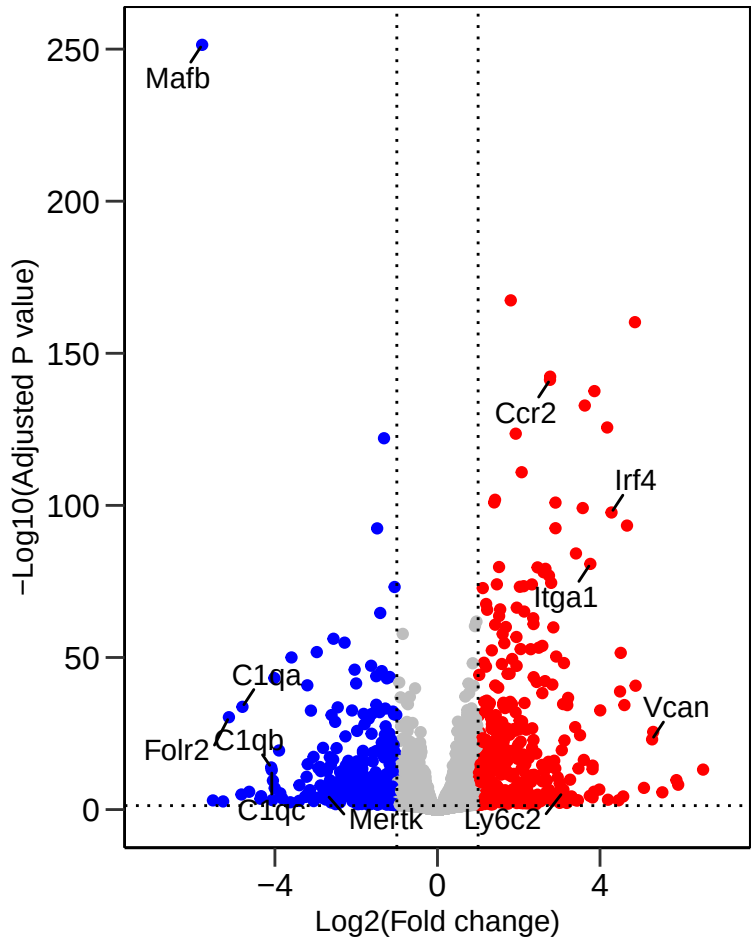
Volcano plot

```
mat <- deg

mat$genes <- rownames(mat)

mat$diffexpressed <- "Not significant"

mat$diffexpressed[mat$log2FoldChange.y > 1 & mat$padj.y < 0.05] <- "KO"

mat$diffexpressed[mat$log2FoldChange.y < -1 & mat$padj.y < 0.05] <- "WT"



mat$label <- NA

Mo_signature_genes <- c("Ccr2", "Vcan", "Ly6c2", "Irf4", "Itga1")

Mac_signature_genes <- c("Mafb", "C1qa", "C1qb", "C1qc", "Mertk",
    "Folr2")

mat$label[mat$genes %in% Mo_signature_genes] <- mat$genes[mat$genes %in%
    Mo_signature_genes]

mat$label[mat$genes %in% Mac_signature_genes] <- mat$genes[mat$genes %in%
    Mac_signature_genes]
```

```
ggplot(data = mat, aes(x = log2FoldChange.y, y = -log10(padj.y),
    col = diffexpressed, label = label)) + geom_point(show.legend = FALSE) +
    geom_text_repel(min.segment.length = 0, box.padding = 0.5,
        size = 4, col = "black") + geom_vline(xintercept = c(-1,
    1), col = "black", linetype = "dotted") + geom_hline(yintercept = -log10(0.05),
    col = "black", linetype = "dotted") + scale_color_manual(values = c("red",
    "grey", "blue")) + xlab("Log2(Fold change)") + ylab("-Log10(Adjusted P value)") +
    xlim(-7, 7) + theme_classic() + theme(axis.text.x = element_text(color = "black",
    size = 12), axis.text.y = element_text(color = "black", size = 12),
    axis.ticks.length = unit(0.25, "cm"), panel.border = element_rect(fill = NA,
        color = "black", linetype = "solid"))
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```

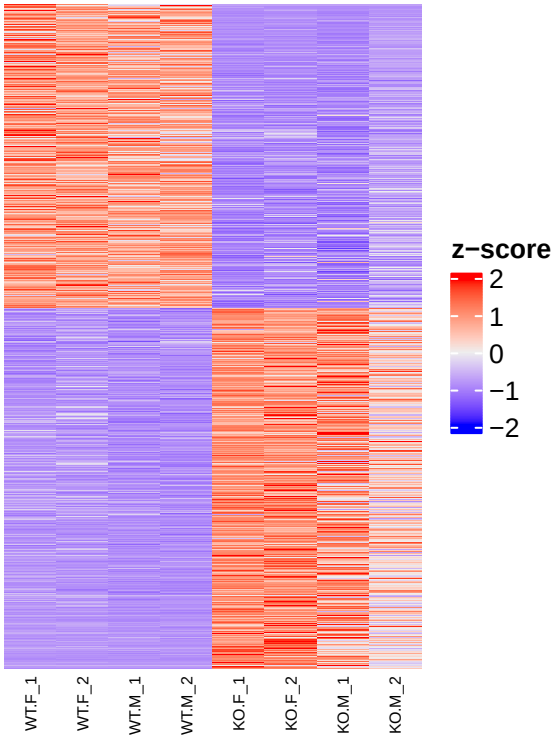
```
## Warning: Removed 12540 rows containing missing values (`geom_text_repel()`).
```




```
ggsave(filename = "Volcanoplot_DEG.pdf", path = "C:/Users/domie/Documents/PhD_Domien_Vanneste_ULiege/Ph
    height = 5, width = 4, device = "pdf")
```

Heatmap

```
counts_norm <- counts(dds, normalized = TRUE)[rownames(deg_significant),
    ]

counts_Z_score <- t(apply(counts_norm, 1, scale))

colnames(counts_Z_score) <- colnames(counts_norm)

counts_Z_score_ordered <- counts_Z_score[rownames(deg_ordered),
    ]

Heatmap(counts_Z_score_ordered, cluster_columns = FALSE, cluster_rows = FALSE,
    row_names_gp = gpar(fontsize = 0), column_names_gp = gpar(fontsize = 6),
    heatmap_legend_param = list(title = "z-score"))
```




```
sessionInfo()
```

```
## R version 4.2.3 (2023-03-15 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Dutch_Netherlands.utf8  LC_CTYPE=Dutch_Netherlands.utf8
## [3] LC_MONETARY=Dutch_Netherlands.utf8 LC_NUMERIC=C
```

```
## [5] LC_TIME=Dutch_Netherlands.utf8
##
## attached base packages:
## [1] grid      stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] ComplexHeatmap_2.14.0       dplyr_1.1.4
##  [3] ggrepel_0.9.4               readxl_1.4.3
##  [5] apeglm_1.20.0               RColorBrewer_1.1-3
##  [7] pheatmap_1.0.12             ggplot2_3.4.4
##  [9] DESeq2_1.38.3               SummarizedExperiment_1.28.0
## [11] Biobase_2.58.0              MatrixGenerics_1.10.0
## [13] matrixStats_1.2.0           GenomicRanges_1.50.2
## [15] GenomeInfoDb_1.34.9         IRanges_2.32.0
## [17] S4Vectors_0.36.2            BiocGenerics_0.44.0
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-7           bit64_4.0.5            doParallel_1.0.17
##  [4] httr_1.4.7             numDeriv_2016.8-1.1    tools_4.2.3
##  [7] utf8_1.2.4             R6_2.5.1               DBI_1.2.0
## [10] colorspace_2.1-0       GetoptLong_1.0.5       withr_3.0.1
## [13] tidyselect_1.2.1       bit_4.0.5              compiler_4.2.3
## [16] cli_3.6.2              formatR_1.14           DelayedArray_0.24.0
## [19] labeling_0.4.3         scales_1.3.0           mvtnorm_1.2-4
## [22] digest_0.6.31          rmarkdown_2.25         XVector_0.38.0
## [25] pkgconfig_2.0.3        htmltools_0.5.7        highr_0.10
## [28] fastmap_1.1.1          bbmle_1.0.25.1         rlang_1.1.2
## [31] GlobalOptions_0.1.2    rstudioapi_0.15.0      RSQLite_2.3.4
## [34] farver_2.1.1           shape_1.4.6            generics_0.1.3
## [37] BiocParallel_1.32.6    RCurl_1.98-1.13        magrittr_2.0.3
## [40] GenomeInfoDbData_1.2.9 Matrix_1.6-4           Rcpp_1.0.11
## [43] munsell_0.5.1          fansi_1.0.6            lifecycle_1.0.4
## [46] yaml_2.3.7             MASS_7.3-58.2          zlibbioc_1.44.0
## [49] plyr_1.8.9             blob_1.2.4             parallel_4.2.3
## [52] bdsmatrix_1.3-6        crayon_1.5.2           lattice_0.20-45
## [55] Biostrings_2.66.0      annotate_1.76.0        circlize_0.4.15
## [58] KEGGREST_1.38.0        locfit_1.5-9.8         knitr_1.45
## [61] pillar_1.9.0           rjson_0.2.21           geneplotter_1.76.0
## [64] codetools_0.2-19       XML_3.99-0.16          glue_1.6.2
## [67] evaluate_0.23          png_0.1-8              vctrs_0.6.5
## [70] foreach_1.5.2          cellranger_1.1.0       gtable_0.3.5
## [73] clue_0.3-65            cachem_1.0.8           emdbook_1.3.13
## [76] xfun_0.39              xtable_1.8-4           coda_0.19-4
## [79] tibble_3.2.1           iterators_1.0.14       AnnotationDbi_1.60.2
## [82] memoise_2.0.1          cluster_2.1.4
```