Mechanical competition drives immune cell death to fuel intestinal tumorigenesis

Erica Valentini

2021-02-19

Abstract

Tissue homeostasis is achieved by balancing stem cell maintenance, cell proliferation and differentiation, as well as the purging damaged cells. Competition for space induces elimination of unfit cells in tissue development. However, the underlying mechanisms driving competitive growth when homeostasis fails, for example during tumorigenesis, remain largely unresolved. Here, using a Drosophila intestinal model, we find that tumor cells outcompete nearby enterocytes (ECs) by exerting junctional tensile forces. This process relies on activating the immune-responsive Relish/NF-kB pathway to induce EC delamination and requires a JNK-dependent transcriptional upregulation of the peptidoglycan recognition protein PGRP-LA. Consequently, in organisms with impaired PGRP-LA function, tumor growth is delayed, and lifespan extended. Our study identifies a non-cell autonomous role for a JNK/PGRP-LA/Relish signaling axis in mediating death of neighboring normal cells to facilitate tumor growth. We propose that intestinal tumors 'hijack' innate immune signaling to eliminate enterocytes in order to support their own growth.

Contents

1 About

This document contains computer code to reproduce the heatmaps and differential gene expression analyses presented in the manuscript.

For this analysis two different datasets have been analysed:

The first dataset consists in the following samples:

1. WT day7 (three replicates)
2. Shn RNAi day7 (two replicates)
3. Shn RNAi PucE69/+ (two replicates)

The comparison performed in section Differential gene expression is: **WT day7 vs. Shn RNAi PucE69/+**

The second dataset is described in the last section Second dataset and contains the following experiments:

1. WT_PMG (two replicates)
2. RelKO_PMG (two replicates)
3. Shn_PMG (three replicates)
4. ShnRelKO_PMG (two replicates)

2 Dependencies

We load a number of packages whose functions are needed throughout the analysis

```
library("tximport")
library("readr")
library("tximportData")
library("tidyverse")
library("tibble")
library("tximeta")
library("org.Dm.eg.db")
library("DESeq2")
library("vsn")
library("pheatmap")
library("RColorBrewer")
library("readxl")
library("apeglm")
library("fgsea")
```

3 Transcript mapping and quantification

Salmon (Patro et al. 2017) was used to map and quantify the transcripts.

The last version of Salmon (salmon-1.0.0_linux_x86_64.tar.gz) was downloaded from github

The reference genome and transcriptome were downloaded from: ensembl.

In order to run it one needs first to create an index:

```
grep "^>" <(gunzip -c Drosophila_melanogaster.BDGP6.22.dna_rm.toplevel.fa.gz) |
cut -d " " -f 1 > decoys.txt

sed -i.bak -e 's/>//g' decoys.txt

cat Drosophila_melanogaster.BDGP6.22.cdna.all.fa.gz
Drosophila_melanogaster.BDGP6.22.dna_rm.toplevel.fa.gz > gentrome.fa.gz

./salmon-latest_linux_x86_64/bin/salmon index -t gentrome.fa.gz -d decoys.txt
-p 12 -i salmon_index
```

And then to quantify the transcripts using a bash script like the following:

```
#!/bin/bash

for fn in directory_with_fastq_files/*;
    do
        samp=`basename ${fn}`
        echo "Processing sample ${samp}"
        echo "located in $fn/fastq/${samp}_R1.fastq.gz"
        ./salmon-latest_linux_x86_64/bin/salmon  quant -i salmon_index -l A \
            -r $fn/fastq/${samp}_R1.fastq.gz \
            -p 8 --validateMappings -o quants/${samp}_quant
done
```

3.1 Import metadata and quantification files

The quantification files are then imported together with a file containing all the metadata regarding the project.

```
# load metadata
samples <- read_csv("samples.csv")
samples %>% dplyr::select(id, condition, SAMPLE_ID, sample_name, FASTQ_FILE)
## # A tibble: 7 x 5
##      id condition       SAMPLE_ID        sample_name      FASTQ_FILE
##   <dbl> <chr>           <chr>            <chr>            <chr>
## 1     1 WT day7         1.WT day7        w1118_1          AS-96216-LR-13830_R1.~
## 2     2 WT day7         2. WT day7       w1118_2          AS-97402-LR-13624_R1.~
## 3     3 WT day7         3. WT day7       w1118_3          AS-97404-LR-13625_R1.~
## 4     4 Shn RNAi day7   4. Shn RNAi day7 Shn-RNAi_1       AS-96218-LR-13830_R1.~
## 5     5 Shn RNAi day7   5. Shn RNAi day7 Shn-RNAi_2       AS-97406-LR-13624_R1.~
## 6     6 Shn RNAi PucE6~ 6. Shn RNAi Puc~ Shn-RNAi,PucE69~ AS-97409-LR-13625_R1.~
## 7     7 Shn RNAi PucE6~ 7. Shn RNAi Puc~ Shn-RNAi,PucE69~ AS-96221-LR-13830_R1.~
# load files with quantification
files <- file.path(samples$directory, "quant.sf")
names(files) <- samples$sample_name
```

3.2 Transform transcript to gene

Tximeta (M. I. Love et al. 2020) was used to import the quantification files and map them to the Drosophila genome.

```
coldata <- data.frame(files, names = samples$sample_name)
se <- tximeta(coldata)
gse <- summarizeToGene(se)
```

Gene symbol and names were added to the FlyBase identifiers:

```
gse <- addIds(gse, "REFSEQ", gene=TRUE)
mcols(gse)
## DataFrame with 14020 rows and 10 columns
##                 gene_id   gene_name   gene_biotype seq_coord_system description
##             <character> <character>    <character>      <character> <character>
## FBgn0000008 FBgn0000008           a protein_coding       chromosome        NULL
## FBgn0000014 FBgn0000014       abd-A protein_coding       chromosome        NULL
## FBgn0000015 FBgn0000015       Abd-B protein_coding       chromosome        NULL
## FBgn0000017 FBgn0000017         Abl protein_coding       chromosome        NULL
## FBgn0000018 FBgn0000018         abo protein_coding       chromosome        NULL
## ...                 ...         ...            ...              ...         ...
## FBgn0286199 FBgn0286199        shps protein_coding       chromosome        NULL
## FBgn0286203 FBgn0286203         stw protein_coding       chromosome        NULL
## FBgn0286204 FBgn0286204         ich protein_coding       chromosome        NULL
## FBgn0286213 FBgn0286213       RpS12 protein_coding       chromosome        NULL
## FBgn0286222 FBgn0286222        Fum1 protein_coding       chromosome        NULL
##             gene_id_version      symbol entrezid
##                 <character> <character>   <list>
## FBgn0000008     FBgn0000008           a    43852
## FBgn0000014     FBgn0000014       abd-A    42037
## FBgn0000015     FBgn0000015       Abd-B    47763
## FBgn0000017     FBgn0000017         Abl    45821
## FBgn0000018     FBgn0000018         abo    44793
## ...                     ...         ...      ...
## FBgn0286199     FBgn0286199        shps    42892
## FBgn0286203     FBgn0286203         stw    35494
## FBgn0286204     FBgn0286204         ich    41069
## FBgn0286213     FBgn0286213       RpS12    39480
## FBgn0286222     FBgn0286222        Fum1    31605
##                                           tx_ids       REFSEQ
##                                     <CharacterList>  <character>
## FBgn0000008 FBtr0071763,FBtr0071764,FBtr0100521,... NM_001014543
## FBgn0000014 FBtr0083387,FBtr0083388,FBtr0300485,... NM_001170161
## FBgn0000015 FBtr0083381,FBtr0083382,FBtr0083383,... NM_001275719
## FBgn0000017 FBtr0075357,FBtr0112790,FBtr0330130,... NM_001104153
## FBgn0000018                           FBtr0080168    NM_080045
## ...                                           ...          ...
## FBgn0286199                           FBtr0084600    NM_142982
## FBgn0286203 FBtr0299918,FBtr0299920,FBtr0299921,... NM_001144134
## FBgn0286204                 FBtr0082014,FBtr0334329 NM_001275464
## FBgn0286213                           FBtr0075878    NM_168534
```

```
## FBgn0286222                FBtr0070953,FBtr0070954    NM_132111
```

4 Gene expression analysis

The `SummarizedExperiment` object produced by tximeta is loaded into DESeq2 (Love, Huber, and Anders 2014) to perform the gene expression analysis. The different conditions **__**WT day7**_**, **Shrn RNAi day7** and **Shn RNAi PucE69/+** were used for the design with WT as reference level.

```
colData(gse)$condition <- as_factor(samples$condition)
dds <- DESeqDataSet(gse, design = ~ condition)
dds$condition <- relevel(dds$condition, ref = "WT day7")
```

4.1 Filter

Only genes with at least 5 counts among all samples are kept.

```
before <- nrow(dds)
#remove low counts
dds <- dds[rowSums(counts(dds)) >= 5, ]
after <- nrow(dds)
#rename genes
rownames(dds) <- mcols(dds)$symbol
```

After filtering the number of genes went from 14020 to 10902.

4.2 Normalization

VST or Variance Stabilizing Transformation (Huber et al. 2002) was the chosen transformation. VST produces transformed data on the log2 scale which has been normalized with respect to library size and other normalization factors to remove the dependence of the variance on the mean, particularly the high variance of the logarithm of count data when the mean is low.

```
# apply VST normalization
vsd <- vst(dds, blind = FALSE)
```

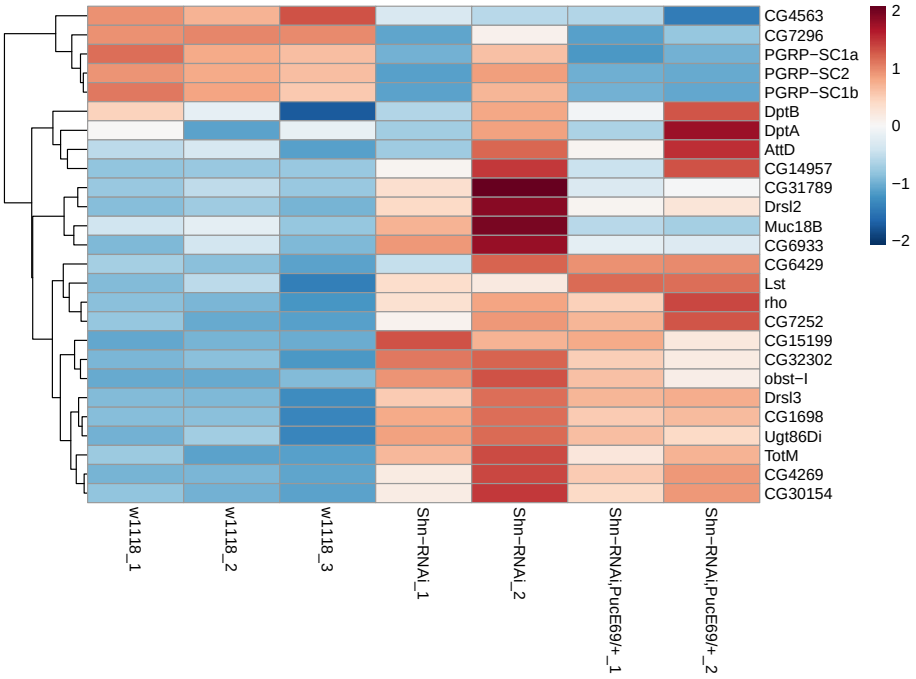
5 Heatmaps

The following heatmaps have been produced using the normalized data and centering and scaling the genes.

5.1 Figure 1D

The heatmap in figure 1D shows the normalized expression of a subset of the genes involved in innate immunity.

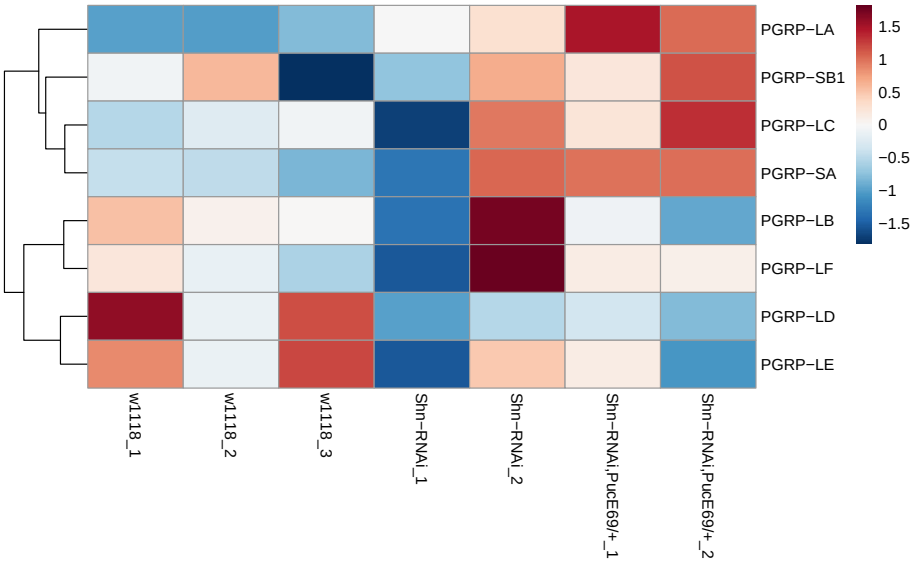
```
# load genes to show in heatmap
NFkB_signature_subset <- read.csv("NFkB_signature.csv",
  row.names=1, stringsAsFactors = FALSE)
colnames(NFkB_signature_subset) <- "gene"
# make heatmap
pheatmap(assay(vsd)[NFkB_signature_subset$gene,],
         scale= 'row',
         color=colorRampPalette(rev(brewer.pal(n=11, name = "RdBu")))(200),
         cluster_cols = F)
```



5.2 Figure 5A

The heatmap in figure 5A shows a list of genes involved in the activation of Imd/Relish pathway.

```
#load genes to show in heatmap
upregulated_genes_immunity <- read.csv("upregulated_genes_immunity.csv",
  row.names=1, stringsAsFactors = FALSE)
colnames(upregulated_genes_immunity) <- "gene"
#make heatmap
pheatmap(assay(vsd)[upregulated_genes_immunity$gene,],
         scale= 'row',
         color=colorRampPalette(rev(brewer.pal(n=11, name = "RdBu")))(200),
         cluster_cols = F)
```



6 Differential gene expression

We calculated the log2 fold changes between **Shn RNAi PucE69/+** and control **WT day7**. The log2 fold changes have then been schrinked using apeglm (Zhu, Ibrahim, and Love 2018) to reduce high variability in low expressing genes.

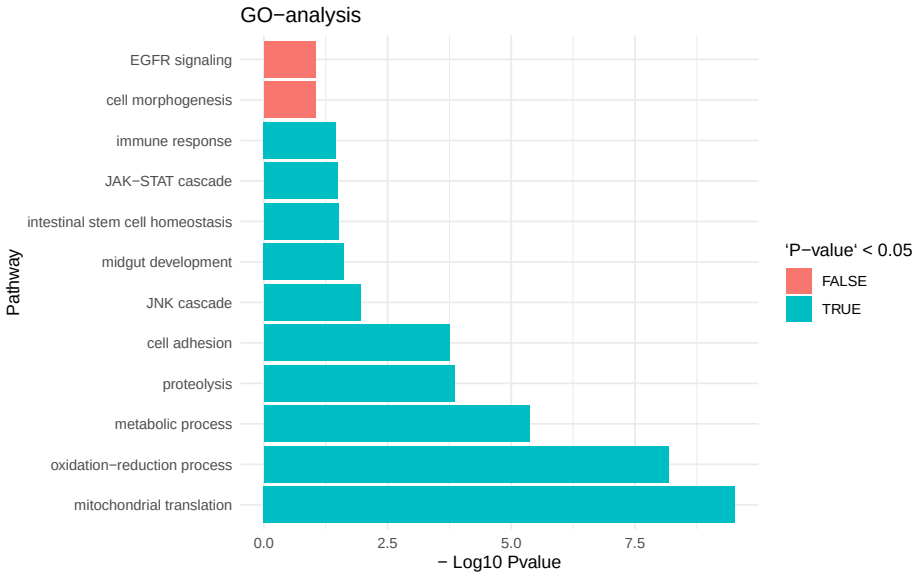
```
dds <- DESeq(dds)
res_wt_puc <- results(dds, contrast = c("condition", "Shn RNAi PucE69/+", "WT day7"))
res <- lfcShrink(dds, coef="condition_Shn.RNAi.PucE69..._vs_WT.day7", type="apeglm")
```

6.1 GO analysis: Figure 1B

A total of 1673 genes are differentially expressed with a p-value below 0.01. Those have been used as input for the gene ontology analysis of biological processes using the online tool DAVID (Huang, Sherman, and Lempicki 2009).

The output of the analysis has been plot using the following:

```
# load go analysis output
res_wt_vs_puc_schrinked <- read_csv("res_wt_vs_puc_schrinked.csv")
# plot results
ggplot(res_wt_vs_puc_schrinked,
       aes(reorder(`Go term_BP enrichment`, `LOG10 P`), -`LOG10 P`)) +
  geom_col(aes(fill=`P-value`<0.05)) +
  coord_flip() +
  labs(x="Pathway", y="- Log10 Pvalue",
       title="GO-analysis") +
  theme_minimal()
```



6.2 GSEA analysis: Figure 1C

We perform gene set enrichment analysis using the Broad Institute's GSEA (Subramanian et al. 2005). An R version of the algorithm is implemented in the `fgsea` algorithm (Sergushichev 2016), which we use for this analysis.

First we load the two gene signatures:

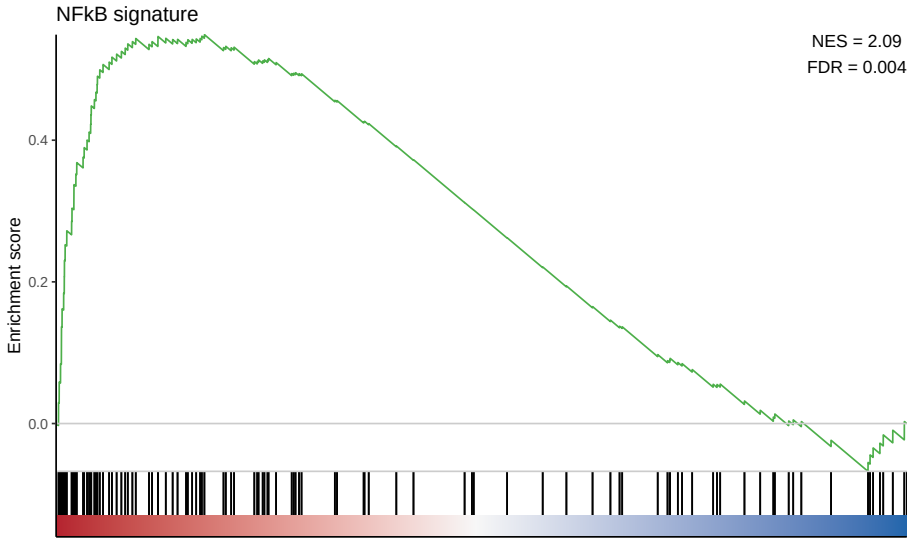
```
NFkB_dependent_gene_list <- read.csv("NFkB_dependent_gene_list.csv",
  row.names=1, stringsAsFactors = FALSE)
colnames(NFkB_dependent_gene_list) <- "gene"

Hippo_signaling_gene_list <- read.csv("Hippo signaling.csv",
  row.names=1, stringsAsFactors = FALSE)
colnames(Hippo_signaling_gene_list) <- "gene"
```

Then we run `fgsea`:

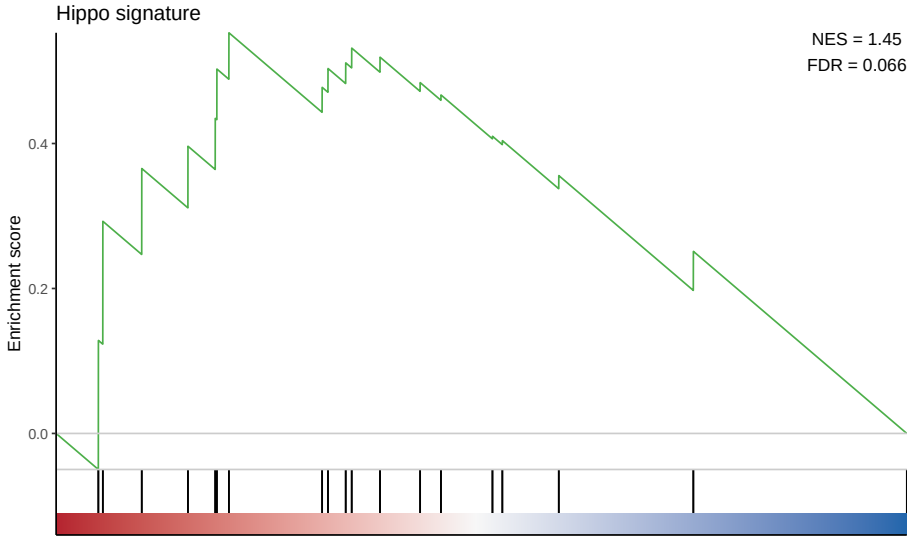
```
pathway <- list()
pathway$NFkB <- NFkB_dependent_gene_list$gene
pathway$Hippo <- Hippo_signaling_gene_list$gene

res_gsea <- as.data.frame(res_wt_puc) %>%
  rownames_to_column(var="SYMBOL") %>%
  dplyr::select(SYMBOL, stat)

ranks <- deframe(res_gsea)

fgseaRes <- fgsea(pathways=pathway, stats=ranks, nperm=1000)
```

We want to visualize the results as a barcode plot. `fgsea` already implements a nice barcode plot, which we customize a bit to adapt it according to our expectations using the following script inspired from (Rauscher, n.d.).

```
custom_barcode_plot <- function(stat_vector, sig){
## genes in signature
sig_genes <- pathway[[sig]]
## generate barcode plot
bc_plot <- plotEnrichment(sig_genes, stat_vector)
## remove unwanted layers
bc_plot$layers <- list()
## add barcode at the bottom
lowest_pos <- min(bc_plot$data[,2])
dash_length <- abs(purrr::reduce(range(bc_plot$data[,2]), `-`)*0.1)
middle <- which.min(abs(sort(stat_vector, decreasing=T)))
bc_plot_custom <- bc_plot + geom_segment(aes(x=x, xend=x), y=lowest_pos,
yend=lowest_pos-dash_length) +
geom_line(colour='#4daf4a') +
geom_hline(yintercept=lowest_pos, colour='#cccccc') +
geom_hline(yintercept=0, colour='#cccccc') + xlab('') +
theme_classic() +
geom_tile(data=tibble(rank=1:length(stat_vector),
y=lowest_pos-(1.25*dash_length)),
aes(x=rank, y=y, fill=rank),
width=1,
height=0.5*dash_length) +
scale_fill_gradient2(low ='#b2182b', high='#2166ac',
mid='#f7f7f7', midpoint = middle) +
scale_x_continuous(expand = c(0, 0)) +
scale_y_continuous(expand = c(0, 0)) +
theme(panel.grid=element_blank(),
axis.text.x=element_blank(),
axis.ticks.x = element_blank(),
legend.position = 'none') +
ggtitle(paste(str_replace_all(sig, "_", " "), "signature", sep = " ")) +
ylab('Enrichment score')
return(bc_plot_custom)
}

bc_plots <- map(1:length(pathway), function(j){
bcp <- custom_barcode_plot(ranks, names(pathway[j])) +
annotate('text', x=Inf , y=Inf, hjust=1, vjust=1,
label=paste('NES =', round(fgseaRes$NES[j], 2),
'\nFDR =', round(fgseaRes$padj[j], 3)))
return(bcp)
})

print(bc_plots)
## [[1]]
```




```
##
## [[2]]
```



7 Second dataset

For the analysis of the second dataset `Salmon` was also used and imported with tximeta as explained in the previous sections.

7.1 Import metadata and quantification files

The quantification files are then imported together with a file containing all the metadata regarding the project.

```
# load metadata
samples <- read_csv("samples_second_dataset.csv")
```

```
samples %>% dplyr::select(id, condition, SAMPLE_ID, sample_name, FASTQ_FILE)
## # A tibble: 9 x 5
##      id condition    SAMPLE_ID        sample_name     FASTQ_FILE
##   <dbl> <chr>        <chr>            <chr>           <chr>
## 1     1 WT_PMG       1-WT_PMG-1       w1118_1         AS-182531-LR-28104_R1.fas~
## 2     3 WT_PMG       3-WT_PMG-3       w1118_2         AS-182532-LR-28105_R1.fas~
## 3     5 RelKO_PMG    5-RelKO_PMG-1    RelE20_1        AS-182534-LR-28104_R1.fas~
## 4     6 RelKO_PMG    6-RelKO_PMG-2    RelE20_2        AS-182535-LR-28105_R1.fas~
## 5     8 Shn_PMG      8-Shn_PMG-1      Shn-RNAi_1      AS-182537-LR-28104_R1.fas~
## 6     9 Shn_PMG      9-Shn_PMG-2      Shn-RNAi_2      AS-182538-LR-28105_R1.fas~
## 7    10 Shn_PMG      10-Shn_PMG-3     Shn-RNAi_3      AS-182539-LR-28104_R1.fas~
## 8    12 ShnRelKO_PMG 12-ShnRelKO_PMG~ Shn-RNAi,RelE2~ AS-182541-LR-28104_R1.fas~
## 9    13 ShnRelKO_PMG 13-ShnRelKO_PMG~ Shn-RNAi,RelE2~ AS-182542-LR-28105_R1.fas~
# load files with quantification
files <- file.path(samples$directory, "quant.sf")
names(files) <- samples$sample_name

# Transform transcript to gene
coldata <- data.frame(files, names = samples$sample_name)
se <- tximeta(coldata)
gse <- summarizeToGene(se)

# Gene symbol and names were added to the FlyBase identifiers:
gse <- addIds(gse, "REFSEQ", gene=TRUE)
```

7.2 Gene expression analysis

The different conditions **WT_PMG**, **RelKO_PMG**, **Shn_PMG** and **ShnRelKO_PMG** were used for the design with **WT_PMG** as reference level.

```
colData(gse)$condition <- as_factor(samples$condition)
dds <- DESeqDataSet(gse, design = ~ condition)
dds$condition <- relevel(dds$condition, ref = "WT_PMG")
```

7.3 Filter and normalization

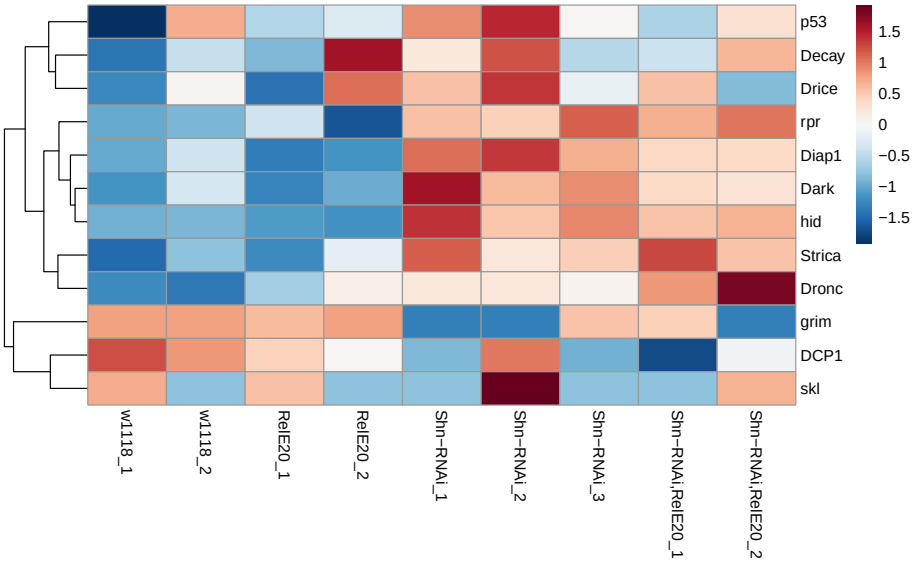
Only genes with at least 5 counts among all samples are kept.

```
# number of genes before:
nrow(dds)
## [1] 14020
#remove low counts
dds <- dds[rowSums(counts(dds)) >= 5, ]
#number of genes after:
nrow(dds)
## [1] 10925
#rename genes
rownames(dds) <- mcols(dds)$symbol
# apply VST normalization
vsd <- vst(dds, blind = FALSE)
```

7.4 Heatmap Figure S3C

The heatmap in figure S3C shows the normalized expression of upregulated genes involved in pro-apoptotic processes.

```
# load genes to show in heatmap
apoptotic_genes <- read.csv("apoptotic_genes.csv",
  row.names=1, stringsAsFactors = FALSE)
colnames(apoptotic_genes) <- "gene"
# make heatmap
pheatmap(assay(vsd)[apoptotic_genes$gene,],
         scale= 'row',
         color=colorRampPalette(rev(brewer.pal(n=11, name = "RdBu")))(200),
         cluster_cols = F)
```



8 Session info

```
sessionInfo()
## R version 4.0.2 (2020-06-22)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Gentoo/Linux
##
## Matrix products: default
## BLAS:   /usr/lib64/libblas.so.3.9.0
## LAPACK: /usr/lib64/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=C.UTF8       LC_NUMERIC=C          LC_TIME=C.UTF8
##  [4] LC_COLLATE=C.UTF8     LC_MONETARY=C.UTF8    LC_MESSAGES=C.UTF8
##  [7] LC_PAPER=C.UTF8       LC_NAME=C             LC_ADDRESS=C
## [10] LC_TELEPHONE=C        LC_MEASUREMENT=C.UTF8 LC_IDENTIFICATION=C
##
```

```
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] fgsea_1.16.0               apeglm_1.12.0
##  [3] readxl_1.3.1               RColorBrewer_1.1-2
##  [5] pheatmap_1.0.12            vsn_3.58.0
##  [7] DESeq2_1.30.0              SummarizedExperiment_1.20.0
##  [9] MatrixGenerics_1.2.1       matrixStats_0.58.0
## [11] GenomicRanges_1.42.0       GenomeInfoDb_1.26.2
## [13] org.Dm.eg.db_3.12.0        AnnotationDbi_1.52.0
## [15] IRanges_2.24.1             S4Vectors_0.28.1
## [17] Biobase_2.50.0             BiocGenerics_0.36.0
## [19] tximeta_1.8.4              forcats_0.5.1
## [21] stringr_1.4.0              dplyr_1.0.4
## [23] purrr_0.3.4                tidyr_1.1.2
## [25] tibble_3.0.6               ggplot2_3.3.3
## [27] tidyverse_1.3.0            tximportData_1.18.0
## [29] readr_1.4.0                tximport_1.18.0
## [31] BiocStyle_2.18.1
##
## loaded via a namespace (and not attached):
##   [1] backports_1.2.1              fastmatch_1.1-0
##   [3] AnnotationHub_2.22.0         BiocFileCache_1.14.0
##   [5] plyr_1.8.6                   lazyeval_0.2.2
##   [7] splines_4.0.2                BiocParallel_1.24.1
##   [9] digest_0.6.27                ensembldb_2.14.0
##  [11] htmltools_0.5.1.1            fansi_0.4.2
##  [13] magrittr_2.0.1               memoise_2.0.0
##  [15] limma_3.46.0                 Biostrings_2.58.0
##  [17] annotate_1.68.0              modelr_0.1.8
##  [19] bdsmatrix_1.3-4              askpass_1.1
##  [21] prettyunits_1.1.1            colorspace_2.0-0
##  [23] blob_1.2.1                   rvest_0.3.6
##  [25] rappdirs_0.3.3               haven_2.3.1
##  [27] xfun_0.21                    crayon_1.4.1
##  [29] RCurl_1.98-1.2               jsonlite_1.7.2
##  [31] genefilter_1.72.1            survival_3.1-12
##  [33] glue_1.4.2                   gtable_0.3.0
##  [35] zlibbioc_1.36.0              XVector_0.30.0
##  [37] DelayedArray_0.16.1          scales_1.1.1
##  [39] mvtnorm_1.1-1                DBI_1.1.1
##  [41] Rcpp_1.0.6                   xtable_1.8-4
##  [43] progress_1.2.2               emdbook_1.3.12
##  [45] bit_4.0.4                    preprocessCore_1.52.1
##  [47] httr_1.4.2                   ellipsis_0.3.1
##  [49] farver_2.0.3                 pkgconfig_2.0.3
##  [51] XML_3.99-0.5                 dbplyr_2.1.0
##  [53] utf8_1.1.4                   locfit_1.5-9.4
##  [55] labeling_0.4.2               tidyselect_1.1.0
```

```
##  [57] rlang_0.4.10                  later_1.1.0.1
##  [59] munsell_0.5.0                 BiocVersion_3.12.0
##  [61] cellranger_1.1.0              tools_4.0.2
##  [63] cachem_1.0.4                  cli_2.3.0
##  [65] generics_0.1.0                RSQLite_2.2.3
##  [67] broom_0.7.4                   evaluate_0.14
##  [69] fastmap_1.1.0                 yaml_2.2.1
##  [71] knitr_1.31                    bit64_4.0.5
##  [73] fs_1.5.0                      AnnotationFilter_1.14.0
##  [75] mime_0.10                     xml2_1.3.2
##  [77] biomaRt_2.46.3                compiler_4.0.2
##  [79] rstudioapi_0.13               curl_4.3
##  [81] interactiveDisplayBase_1.28.0 affyio_1.60.0
##  [83] reprex_1.0.0                  geneplotter_1.68.0
##  [85] stringi_1.5.3                 GenomicFeatures_1.42.1
##  [87] lattice_0.20-41               ProtGenerics_1.22.0
##  [89] Matrix_1.2-18                 vctrs_0.3.6
##  [91] pillar_1.4.7                  lifecycle_1.0.0
##  [93] BiocManager_1.30.10           data.table_1.13.6
##  [95] bitops_1.0-6                  httpuv_1.5.5
##  [97] rtracklayer_1.50.0            R6_2.5.0
##  [99] affy_1.68.0                   bookdown_0.21
## [101] promises_1.2.0.1              gridExtra_2.3
## [103] MASS_7.3-51.6                 assertthat_0.2.1
## [105] openssl_1.4.3                 withr_2.4.1
## [107] GenomicAlignments_1.26.0      Rsamtools_2.6.0
## [109] GenomeInfoDbData_1.2.4        hms_1.0.0
## [111] grid_4.0.2                    coda_0.19-4
## [113] rmarkdown_2.6                 bbmle_1.0.23.1
## [115] numDeriv_2016.8-1.1           shiny_1.6.0
## [117] lubridate_1.7.9.2
```

Bibliography

Huang, Da Wei, Brad T Sherman, and Richard A Lempicki. 2009. "Systematic and Integrative Analysis of Large Gene Lists Using David Bioinformatics Resources." *Nature Protocols* 4 (1): 44–57. doi:10.1038/nprot.2008.211.

Huber, Wolfgang, Anja von Heydebreck, Holger Sueltmann, Annemarie Poustka, and Martin Vingron. 2002. "Variance Stabilization Applied to Microarray Data Calibration and to the Quantification of Differential Expression." *Bioinformatics* 18 Suppl. 1: S96–S104.

Love, Michael I, Wolfgang Huber, and Simon Anders. 2014. "Moderated Estimation of Fold Change and Dispersion for Rna-Seq Data with Deseq2." *Genome Biology* 15 (12). BioMed Central: 550. doi:10.1186/s13059-014-0550-8.

Love, Michael I., Charlotte Soneson, Peter F. Hickey, Lisa K. Johnson, N. Tessa Pierce, Lori Shepherd, Martin Morgan, and Rob Patro. 2020. "Tximeta: Reference Sequence Checksums for Provenance Identification in Rna-Seq." Edited by MihaelaEditor Pertea. *PLOS Computational Biology* 16 (2): e1007664. doi:10.1371/journal.pcbi.1007664.

Patro, Rob, Geet Duggal, Michael I Love, Rafael A Irizarry, and Carl Kingsford. 2017. "Salmon Provides Fast and Bias-Aware Quantification of Transcript Expression." *Nature Methods* 14 (4): 417–19. doi:10.1038/nmeth.4197.

Rauscher, Benedikt. n.d. *GitHub*. https://github.com/boutroslab/Supplemental-Material/blob/master/Zhan&Ambrosi&Wandmacher_2018/MEKWnt2018.pdf.

Sergushichev, Alexey. 2016. "An Algorithm for Fast Preranked Gene Set Enrichment Analysis Using Cumulative Statistic Calculation." *bioRxiv*. Cold Spring Harbor Labs Journals. doi:10.1101/060012.

Subramanian, A., P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, et al. 2005. "Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles." *Proceedings of the National Academy of Sciences* 102 (43): 15545–50. doi:10.1073/pnas.0506580102.

Zhu, Anqi, Joseph G Ibrahim, and Michael I Love. 2018. "Heavy-Tailed Prior Distributions for Sequence Count Data: Removing the Noise and Preserving Large Differences." Edited by OliverEditor Stegle. *Bioinformatics* 35 (12): 2084–92. doi:10.1093/bioinformatics/bty895.