# RNAseq Data Differential Expression

*Manfred Schmid*

03 July, 2020

```
suppressWarnings(library("tidyverse"))
suppressWarnings(library("knitr"))
suppressWarnings(library("magrittr"))
suppressWarnings(library("broom"))
suppressWarnings(library("DESeq2"))
suppressWarnings(library("limma"))
suppressWarnings(library("RColorBrewer"))
```

## Mapping

RNA extraction and RNA-seq library preparation upon GFP and NCBP3 depletion were performed as described (Iasillo et al., 2017). Total RNAseq of siNCBP3 depletion samples are first reported here, but were collected as part of same experiment described in (Iasillo et al., 2017 and Winczura et al., 2018) and deposited at GEO:GSE99059. Reads from siNCBP3,control siEGFP and relevant depletions from GEO:GSE99059 were processed in parallel as described in (Silla et al., under revision). In brief, raw reads were quality filtered and trimmed as described (Meola et al., 2016), using Trimmomatic (v 0.32) and settings (PE ILLUMINACLIP:/com/extra/Trimmomatic/0.32/adapters/TruSeq3-PE-2.fa:2:30:10 HEADCROP:12 LEADING:22 SLIDINGWINDOW:4:22 MINLEN:25). Cleaned reads were then mapped to GRCh38 with HISAT2 (v 2.1.0) (Kim et al., 2015) using default settings and the genome index 'H. sapiens, UCSC hg38 and Refseq gene annotations' provided at the HISAT2 download page (ftp://ftp.ccb.jhu.edu/pub/infphilo/hisat2/data/hg38_tran.tar.gz). Only proper pairs with both reads mapping to the genome were used for further analysis.

### Trimming

```
#!/bin/sh
#call:
## cd faststorage/ClaudiaI/scripts/RNAseq_hg38/
## for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_HK_may2016_fastq/clean_reads/Ars2*/*
## for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_HK_may2016_fastq/clean_reads/eGFP*/*
## for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_HK_may2016_fastq/clean_reads/NCBP3*/
## for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_HK_may2016_fastq/clean_reads/Cbp80*/
## for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_HK_may2016_fastq/clean_reads/Z18*/*_
## for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_HK_may2016_fastq/clean_reads/Cbp20*/
## for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_BrU_HK_fastq/clean_reads/*/*_1.fq.gz


source /com/extra/Trimmomatic/0.32/load.sh

fastq1=$1
echo $fastq1

fastq2=${fastq1/_1.fq.gz/_2.fq.gz}
fastq_out1=${fastq1/_1.fq.gz/_1P.fq.gz}
fastq_out2=${fastq1/_1.fq.gz/_1U.fq.gz}
```

```
fastq_out3=${fastq1/_1.fq.gz/_2P.fq.gz}
fastq_out4=${fastq1/_1.fq.gz/_2U.fq.gz}

java -jar /com/extra/Trimmomatic/0.32/trimmomatic-0.32.jar PE ${fastq1} ${fastq2} ${fastq_out1} ${fastq

echo "done"
```

### Mapping

```
#!/bin/sh
##hisat2 to hg38 mapping!

# cd faststorage/ClaudiaI/scripts/RNAseq_hg38/
# for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_HK_may2016_fastq/clean_reads/eGFP*/*_
# for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_HK_may2016_fastq/clean_reads/NCBP3*/*
# for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_HK_may2016_fastq/clean_reads/Ars2*/*_
# for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_HK_may2016_fastq/clean_reads/Cbp80*/*
# for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_HK_may2016_fastq/clean_reads/Z18*/*_1
# for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_HK_may2016_fastq/clean_reads/Cbp20*/*
# for f in /home/schmidm/faststorage/ClaudiaI/RNAseq_fastq/RNASeq_BrU_HK_fastq/clean_reads/*/*_1P.fq.gz


. /home/schmidm/miniconda2/etc/profile.d/conda.sh
conda activate hisat2

set -x
fastq1=$1
fastq2=${fastq1//_1P.fq.gz/_2P.fq.gz}
name=$(echo $fastq1 | sed s/.*clean_reads\\///g | sed s/-/_/g | sed s/_rep.*//g | sed s/tot/tot_/g)

index="/home/schmidm/annotations/hg38/HISAT2_index/hg38_tran/genome_tran"


sam="/home/schmidm/faststorage/ClaudiaI/RNAseq_bams_hg38/${name}.sam"

echo "for ${name}"
echo " mapping mate1 $fastq1"
echo " mapping mate2 $fastq2"
echo " into $sam"
hisat2 -p 8 -x $index -1 $fastq1 -2 $fastq2 -S $sam

bam=${sam/.sam/_unique_properlypaired.bam}

source /com/extra/samtools/1.6.0/load.sh

echo "sorting and filtering $sam into $bam"
## -f 2 selest only reads in proper pairs
## -F 780 deselect reads unmapped (0x4), read with mate unmapped (0x8), not primary alignment (0x100) a
## -u uncompressed output
## -S input is sam file
samtools view -S -u -f 2 -F 780 ${sam} | samtools sort -m 14G -T "/scratch/${SLURM_JOBID}/tmp" -o ${bam

echo " indexing"
```

```
samtools index ${bam}

echo "DONE"
```

### Gene counts from bam files

```
#!/bin/sh
##featureCounts to hg38 mapping!

libs="tot"
/home/schmidm/ms_tools/subread-2.0.0-Linux-x86_64/bin/featureCounts -p -C -s 2 -F SAF -a /home/schmidm/r
echo "DONE"
```

## load count data

### load exon counts to R

```
read_count_file <- function(fname) {
  df <- read_tsv(fname, comment = '#') %>%
    dplyr::select(Geneid, contains('.bam')) %>%
    mutate(Geneid = as.character(Geneid))

  colnames(df) <- sub('.*\\/', '', colnames(df)) %>% sub('_unique.*', '', .)

  df %>%
    gather(lib, cnt, -Geneid)
}
```

```
(exon_counts <- read_count_file('/Volumes/GenomeDK/faststorage/ClaudiaI/RNAseq_bams_hg38/featureCounts_
```

```
## # A tibble: 511,110 x 3
##    Geneid    lib          cnt
##    <chr>     <chr>      <int>
##  1 100287102 Ars2_tot_1     0
##  2 653635    Ars2_tot_1    91
##  3 102466751 Ars2_tot_1     2
##  4 100302278 Ars2_tot_1     0
##  5 645520    Ars2_tot_1     0
##  6 79501     Ars2_tot_1     0
##  7 729737    Ars2_tot_1     0
##  8 102725121 Ars2_tot_1     0
##  9 102723897 Ars2_tot_1    28
## 10 102465909 Ars2_tot_1     0
## # ... with 511,100 more rows
```

## Prepare counts and sample info for DESeq2

```
cnt_mat <- exon_counts %>%
  spread(lib, cnt) %>%
  data.frame %>%
  column_to_rownames('Geneid')

head(cnt_mat)
```

```
##           Ars2_tot_1 Ars2_tot_2 Ars2_tot_3 Cbp20_tot_1 Cbp20_tot_2
## 1                  2          2          3           6           2
## 10                 0          0          0           0           0
## 100              548        414        526         875         653
## 1000               0          1          3           0           0
## 10000              0          0          0           0           0
## 100008587          0          0          0           0           0
##           Cbp20_tot_3 Cbp80_tot_1 Cbp80_tot_2 Cbp80_tot_3 eGFP_tot_1
## 1                   1           1           0           2          0
## 10                  0           0           0           0          0
## 100               508         551         556         660        874
## 1000                0           0           0           4          0
## 10000               0           0           1           0          0
## 100008587           0           0           0           0          0
##           eGFP_tot_2 eGFP_tot_3 NCBP3_tot_1 NCBP3_tot_2 NCBP3_tot_3
## 1                   1          0           1           0           0
## 10                  0          0           0           0           0
## 100              1214        774         796         578         945
## 1000                2          2           1           0           2
## 10000               0          0           0           0           0
## 100008587           0          0           0           0           0
##           Z18_tot_1 Z18_tot_2 Z18_tot_3
## 1                 6         1         1
## 10                0         0         0
## 100            1143       473       797
## 1000              0         0         0
## 10000             0         0         0
## 100008587         0         0         0
```

```
coldata <- data.frame(condition = colnames(cnt_mat)) %>%
  tidyr::separate(condition, c('siRNA', 'fraction', 'replicate'), by='_', remove=FALSE) %>%
  column_to_rownames('condition')

kable(coldata)
```

| | siRNA | fraction | replicate |
|---|---|---|---|
| Ars2_tot_1 | Ars2 | tot | 1 |
| Ars2_tot_2 | Ars2 | tot | 2 |
| Ars2_tot_3 | Ars2 | tot | 3 |
| Cbp20_tot_1 | Cbp20 | tot | 1 |
| Cbp20_tot_2 | Cbp20 | tot | 2 |
| Cbp20_tot_3 | Cbp20 | tot | 3 |
| Cbp80_tot_1 | Cbp80 | tot | 1 |
| Cbp80_tot_2 | Cbp80 | tot | 2 |
| Cbp80_tot_3 | Cbp80 | tot | 3 |
| eGFP_tot_1 | eGFP | tot | 1 |
| eGFP_tot_2 | eGFP | tot | 2 |

| | siRNA | fraction | replicate |
|---|---|---|---|
| eGFP_tot_3 | eGFP | tot | 3 |
| NCBP3_tot_1 | NCBP3 | tot | 1 |
| NCBP3_tot_2 | NCBP3 | tot | 2 |
| NCBP3_tot_3 | NCBP3 | tot | 3 |
| Z18_tot_1 | Z18 | tot | 1 |
| Z18_tot_2 | Z18 | tot | 2 |
| Z18_tot_3 | Z18 | tot | 3 |

–> several of these libraries are not used actively here, but they are all samples from this very same batch of experiment. All are included at this step assuming that the more samples yield more accurate dispersion estimation.

```
ddsFullCountTable <- DESeqDataSetFromMatrix(countData = cnt_mat,
                                            colData = coldata,
                                            design = ~ siRNA)

dds <- DESeq(ddsFullCountTable)
```

```
save(dds, file = '../data/DESeq2_hg38.RData')
```

alternative starting point:

```
load('../data/DESeq2_hg38.RData', verbose=T)
```

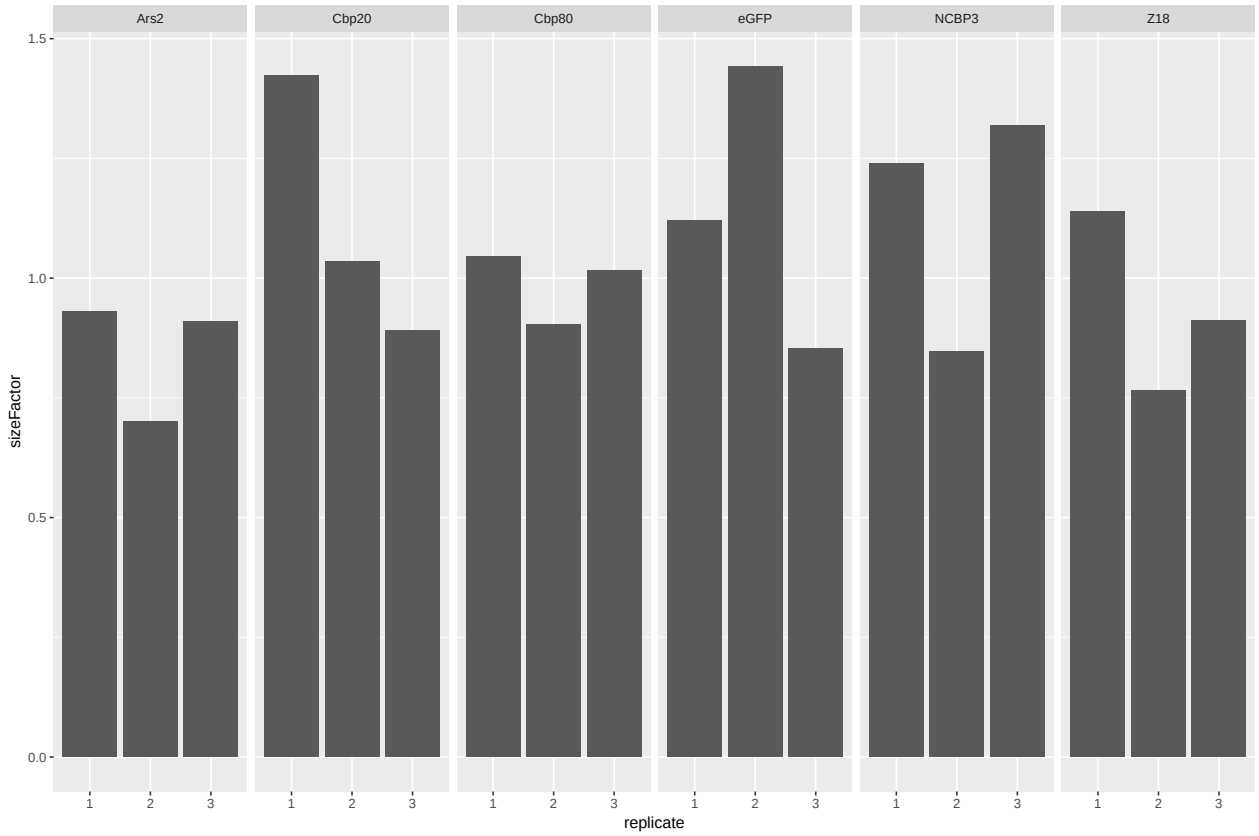
```
## Loading objects:
##   dds
```

```
dds
```

```
## class: DESeqDataSet
## dim: 28395 18
## metadata(1): version
## assays(4): counts mu H cooks
## rownames(28395): 1 10 ... 9994 9997
## rowData names(38): baseMean baseVar ... deviance maxCooks
## colnames(18): Ars2_tot_1 Ars2_tot_2 ... Z18_tot_2 Z18_tot_3
## colData names(4): siRNA fraction replicate sizeFactor
```

**size factors exons**

```
ggplot(data.frame(colData(dds)), aes(x=replicate,y=sizeFactor)) + geom_bar(stat='identity') + facet_gri
```




Some variation but no extreme outliers.

```
sfs <- sizeFactors(dds)
save(sfs, file = '../data/DESeq2_hg38_sizeFactors.RData')
```

# PCA of count data

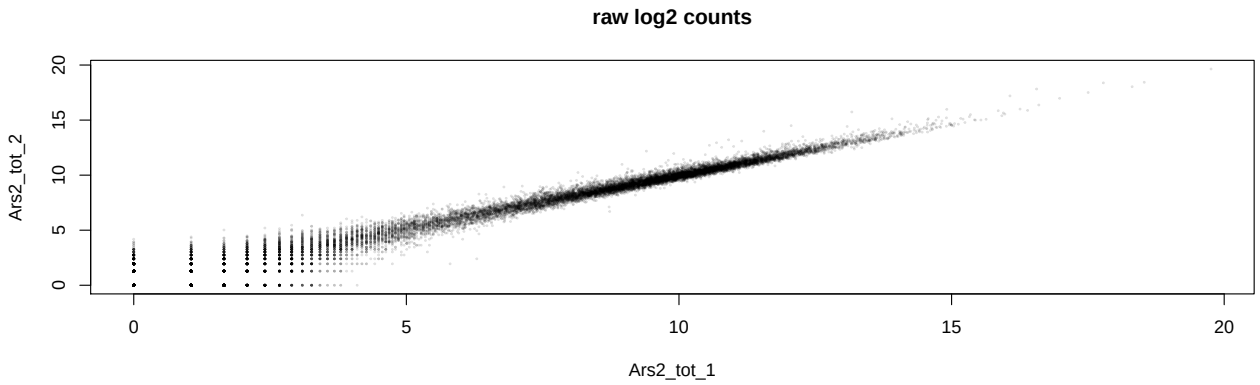
Follow standard DESeq2 vignette-based pipeline. Using vst for variance-stabilization before clustering.

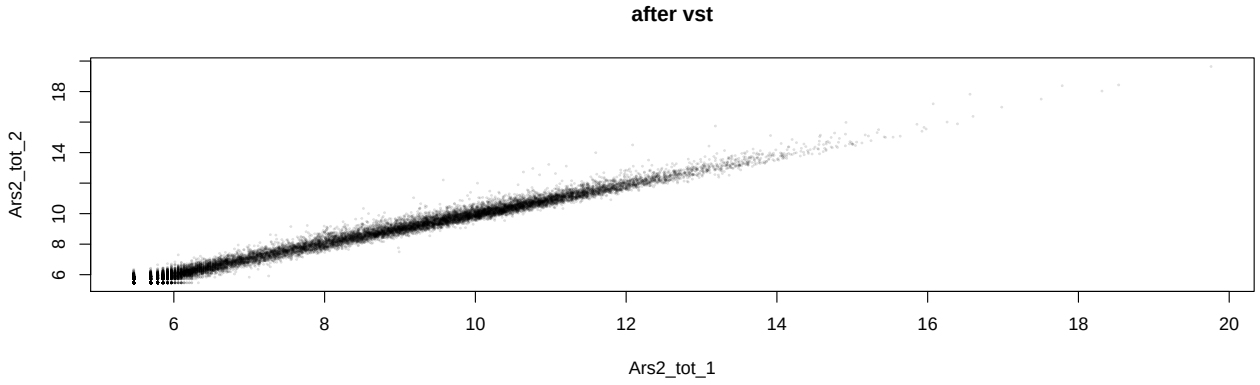
```
vst <- vst( dds )
head( assay(vst) )
```

```
##           Ars2_tot_1 Ars2_tot_2 Ars2_tot_3 Cbp20_tot_1 Cbp20_tot_2
## 1           5.785422   5.833698   5.860856    5.911489    5.768919
## 10          5.468083   5.468083   5.468083    5.468083    5.468083
## 100         9.398349   9.403572   9.375885    9.451865    9.484809
## 1000        5.468083   5.726957   5.860856    5.468083    5.468083
## 10000       5.468083   5.468083   5.468083    5.468083    5.468083
## 100008587   5.468083   5.468083   5.468083    5.468083    5.468083
##           Cbp20_tot_3 Cbp80_tot_1 Cbp80_tot_2 Cbp80_tot_3 eGFP_tot_1
## 1            5.697658    5.679975    5.468083    5.771684   5.468083
## 10           5.468083    5.468083    5.468083    5.468083   5.468083
## 100          9.357876    9.258328    9.455156    9.521877   9.758642
## 1000         5.468083    5.468083    5.468083    5.896655   5.468083
## 10000        5.468083    5.468083    5.696123    5.468083   5.468083
## 100008587    5.468083    5.468083    5.468083    5.468083   5.468083
##           eGFP_tot_2 eGFP_tot_3 NCBP3_tot_1 NCBP3_tot_2 NCBP3_tot_3
```

```
## 1           5.648528   5.468083    5.662726    5.468083    5.468083
## 10          5.468083   5.468083    5.468083    5.468083    5.468083
## 100         9.858235   9.955394    9.508127    9.585135    9.647631
## 1000        5.723105   5.799189    5.662726    5.468083    5.734638
## 10000       5.468083   5.468083    5.468083    5.468083    5.468083
## 100008587   5.468083   5.468083    5.468083    5.468083    5.468083
##           Z18_tot_1 Z18_tot_2 Z18_tot_3
## 1          5.963442  5.715633  5.694940
## 10         5.468083  5.468083  5.468083
## 100       10.090993  9.458822  9.907784
## 1000       5.468083  5.468083  5.468083
## 10000      5.468083  5.468083  5.468083
## 100008587  5.468083  5.468083  5.468083
```

```
par(mfrow=c(2,1))
plot( log2( 1+counts(dds, normalized=TRUE)[, 1:2] ), col="#00000020", pch=20, cex=0.3, main='raw log2 c
plot( assay(vst)[, 1:2], col="#00000020", pch=20, cex=0.3, main='after vst')
```

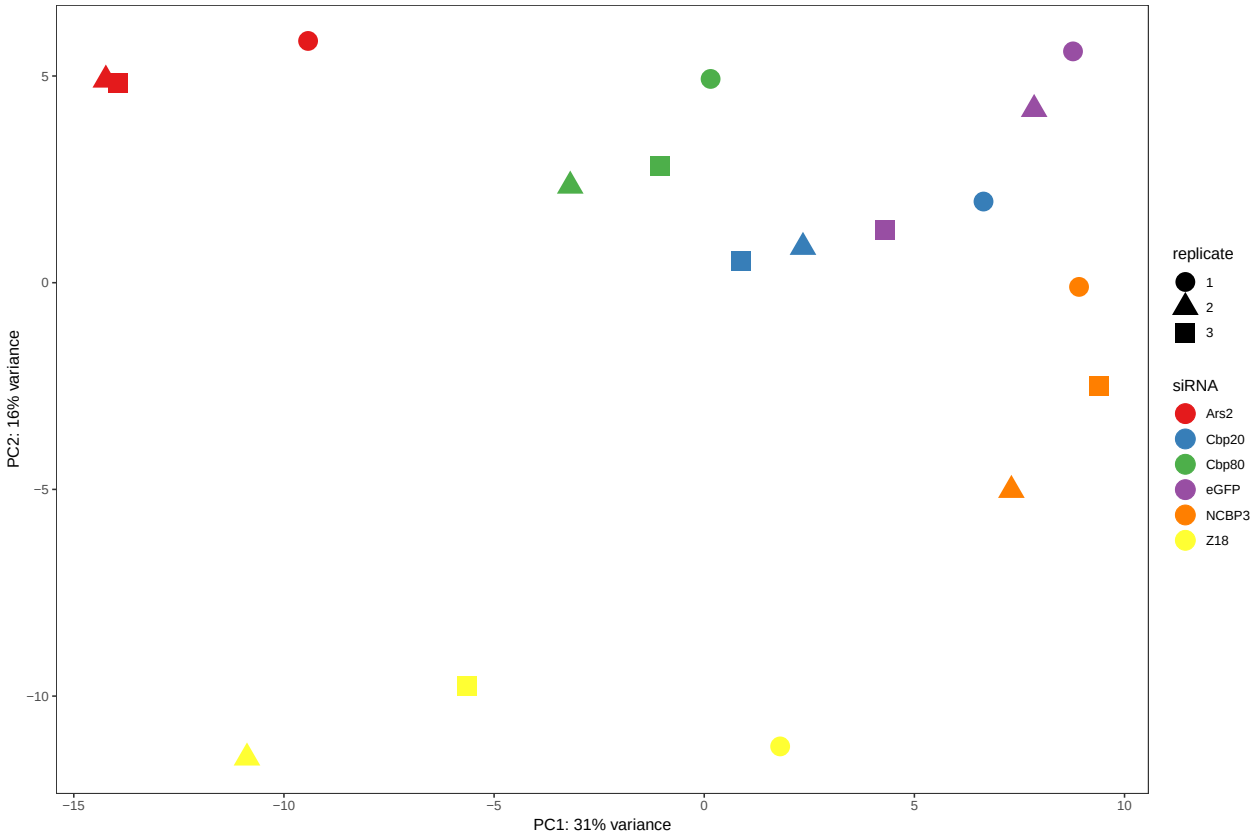



```
par(mfrow=c(1,1))
```

```
pca <- plotPCA(vst, intgroup = c('siRNA', 'replicate'), returnData = TRUE)

ggplot(pca, (aes(x=PC1, y=PC2, color=siRNA, shape=replicate))) +
    geom_point(size=6) +
    xlab(paste0("PC1: ", round(attr(pca, 'percentVar')[1] *
        100), "% variance")) +
    ylab(paste0("PC2: ", round(attr(pca, 'percentVar')[2] *
        100), "% variance")) +
    scale_color_brewer(palette = 'Set1') +
```

```
    theme_bw() +
  theme(panel.grid=element_blank())
```




–> Cbp20 has minor phenotype, possibly due to bad RNAi efficiency but this was not followed up. Cbp20 data is not used for any conclusions. The other KDs show distinct phenotypes as expected.

# Differential expression

contrasts to use (all relative to egfp of course)

```
(sirnas <- unique(colData(dds)$siRNA) %>%
  keep(!grepl('eGFP', .)) %>%
  as.character)
```

```
## [1] "Ars2"  "Cbp20" "Cbp80" "NCBP3" "Z18"
```

```
contrasts <- lapply(sirnas, function(sirna) c('siRNA', sirna, 'eGFP'))
names(contrasts) <- sirnas
```

```
RNAseq_DESeq2_results <- lapply(contrasts, function(contr) results(dds,
                                      contrast=contr,
                                      tidy = TRUE))
```

```
(deseq_res <- lapply(seq_along(RNAseq_DESeq2_results), function(i) data.frame(RNAseq_DESeq2_results[[i]]
                     as_tibble %>%
                     mutate(comparison = names(contrasts)[i])) %>%
  bind_rows %>%
```

```
  mutate(sig = case_when(.$padj < .1 & .$log2FoldChange > 0 ~ 'sig up',
                         .$padj < .1 & .$log2FoldChange < 0 ~ 'sig dn',
                         TRUE ~ 'not sig')))
```

```
## # A tibble: 141,975 x 9
##    row       baseMean log2FoldChange  lfcSE     stat    pvalue     padj
##    <chr>        <dbl>          <dbl>  <dbl>    <dbl>     <dbl>    <dbl>
##  1 1           1.54            3.10   1.74    1.78    0.0750   NA
##  2 10          0              NA     NA      NA      NA        NA
##  3 100       691.             -0.522  0.153  -3.42    0.000637  0.00724
##  4 1000        0.818           0.362  2.49    0.146   0.884    NA
##  5 10000       0.0616          0.408  5.27    0.0774  0.938    NA
##  6 100008587   0              NA     NA      NA      NA        NA
##  7 100008588   0              NA     NA      NA      NA        NA
##  8 100008589   0              NA     NA      NA      NA        NA
##  9 100009601   0              NA     NA      NA      NA        NA
## 10 100009602   0.188           1.37   5.27    0.260   0.795    NA
## # ... with 141,965 more rows, and 2 more variables: comparison <chr>,
## #   sig <chr>
```

```
save(deseq_res, file='../data/DESeq2_hg38_res_all_Rel_EGFP.RData')
```

```
load('../data/DESeq2_hg38_res_all_Rel_EGFP.RData', verbose=T)
```

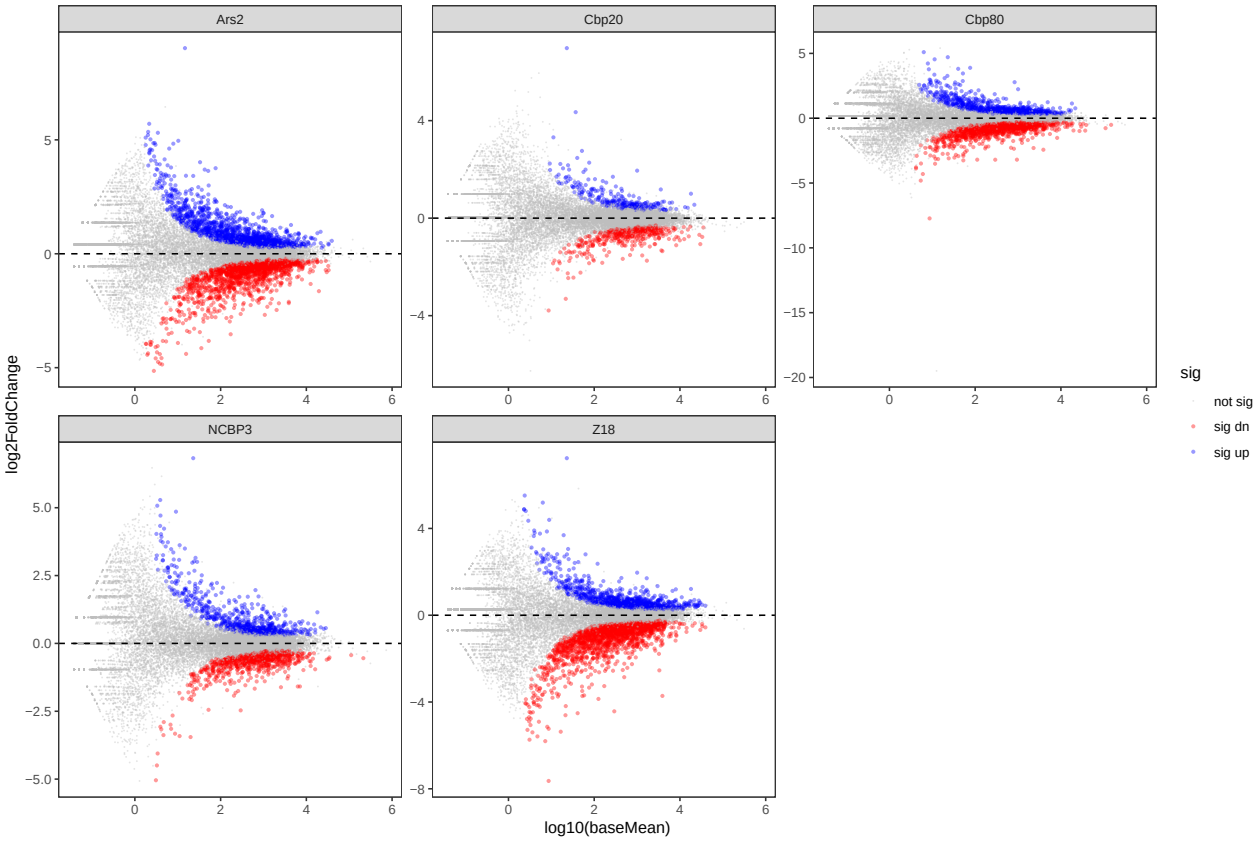
```
## Loading objects:
##   deseq_res
```

```
head(deseq_res)
```

```
## # A tibble: 6 x 9
##   row       baseMean log2FoldChange  lfcSE     stat    pvalue     padj
##   <chr>        <dbl>          <dbl>  <dbl>    <dbl>     <dbl>    <dbl>
## 1 1           1.54            3.10   1.74    1.78    0.0750   NA
## 2 10          0              NA     NA      NA      NA        NA
## 3 100       691.             -0.522  0.153  -3.42    0.000637  0.00724
## 4 1000        0.818           0.362  2.49    0.146   0.884    NA
## 5 10000       0.0616          0.408  5.27    0.0774  0.938    NA
## 6 100008587   0              NA     NA      NA      NA        NA
## # ... with 2 more variables: comparison <chr>, sig <chr>
```

### MA plots

```
ggplot(deseq_res, aes(x=log10(baseMean), y=log2FoldChange, color=sig, size=sig)) +
  geom_point(alpha=.4, shape=16) +
  scale_color_manual(values=c('gray', 'red', 'blue'), na.value='gray') +
  scale_size_manual(values=c(.25, 1, 1), na.value=.25) +
  geom_hline(yintercept = 0, linetype=2) +
  facet_wrap(~comparison, scales='free') +
  theme_bw() +
  theme(panel.grid=element_blank())
```




–> the subplot for NCBP3 is included in the paper.

**total sig up and down numbers per KD**

```
deseq_res %>%
  group_by(sig, comparison) %>%
  summarize(cnt = n()) %>%
  spread(sig, cnt)
```

```
## # A tibble: 5 x 4
##   comparison `not sig` `sig dn` `sig up`
##   <chr>          <int>    <int>    <int>
## 1 Ars2           25267     1650     1478
## 2 Cbp20          27736      398      261
## 3 Cbp80          25966     1416     1013
## 4 NCBP3          26853      818      724
## 5 Z18            25483     1763     1149
```

# Add gene annotations

```
library("AnnotationDbi")
library("org.Hs.eg.db")
columns(org.Hs.eg.db)
```

```
##  [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"
```

```
##  [5] "ENSEMBLTRANS" "ENTREZID"     "ENZYME"       "EVIDENCE"
##  [9] "EVIDENCEALL"  "GENENAME"     "GO"           "GOALL"
## [13] "IPI"          "MAP"          "OMIM"         "ONTOLOGY"
## [17] "ONTOLOGYALL"  "PATH"         "PFAM"         "PMID"
## [21] "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"
## [25] "UNIGENE"      "UNIPROT"
```

```
entrezid_map <- select(org.Hs.eg.db,
       key=unique(deseq_res$row), columns=c('ENSEMBL', 'SYMBOL'),
       keytype="ENTREZID")

head(entrezid_map)
```

```
##   ENTREZID         ENSEMBL SYMBOL
## 1        1 ENSG00000121410   A1BG
## 2       10 ENSG00000156006   NAT2
## 3      100 ENSG00000196839    ADA
## 4     1000 ENSG00000170558   CDH2
## 5    10000 ENSG00000117020   AKT3
## 6    10000 ENSG00000275199   AKT3
```

### add detailed annotation info

this is based on ENSEMBL IDs, which can be used to obtain detailed info for each gene

```
suppressWarnings(library('AnnotationHub'))

hub <- AnnotationHub()
hub <- subset(hub, hub$genome=='GRCh38')
hub <- subset(hub, hub$title=='Homo_sapiens.GRCh38.92.gtf')

hub_name <- names(hub)
gr <- hub[[hub_name[1]]]
head(gr)
```

```
## GRanges object with 6 ranges and 22 metadata columns:
##       seqnames      ranges strand |   source       type     score
##          <Rle>   <IRanges>  <Rle> | <factor>   <factor> <numeric>
##   [1]        1 11869-14409      + |   havana       gene      <NA>
##   [2]        1 11869-14409      + |   havana transcript      <NA>
##   [3]        1 11869-12227      + |   havana       exon      <NA>
##   [4]        1 12613-12721      + |   havana       exon      <NA>
##   [5]        1 13221-14409      + |   havana       exon      <NA>
##   [6]        1 12010-13670      + |   havana transcript      <NA>
##           phase         gene_id gene_version   gene_name gene_source
##       <integer>     <character>  <character> <character> <character>
##   [1]      <NA> ENSG00000223972            5     DDX11L1      havana
##   [2]      <NA> ENSG00000223972            5     DDX11L1      havana
##   [3]      <NA> ENSG00000223972            5     DDX11L1      havana
##   [4]      <NA> ENSG00000223972            5     DDX11L1      havana
##   [5]      <NA> ENSG00000223972            5     DDX11L1      havana
##   [6]      <NA> ENSG00000223972            5     DDX11L1      havana
##                             gene_biotype   transcript_id
##                              <character>     <character>
##   [1] transcribed_unprocessed_pseudogene            <NA>
```

```
##   [2] transcribed_unprocessed_pseudogene ENST00000456328
##   [3] transcribed_unprocessed_pseudogene ENST00000456328
##   [4] transcribed_unprocessed_pseudogene ENST00000456328
##   [5] transcribed_unprocessed_pseudogene ENST00000456328
##   [6] transcribed_unprocessed_pseudogene ENST00000450305
##       transcript_version transcript_name transcript_source
##              <character>     <character>       <character>
##   [1]               <NA>            <NA>              <NA>
##   [2]                  2     DDX11L1-202            havana
##   [3]                  2     DDX11L1-202            havana
##   [4]                  2     DDX11L1-202            havana
##   [5]                  2     DDX11L1-202            havana
##   [6]                  2     DDX11L1-201            havana
##                       transcript_biotype         tag
##                                  <character> <character>
##   [1]                                <NA>        <NA>
##   [2]               processed_transcript       basic
##   [3]               processed_transcript       basic
##   [4]               processed_transcript       basic
##   [5]               processed_transcript       basic
##   [6] transcribed_unprocessed_pseudogene       basic
##       transcript_support_level exon_number         exon_id exon_version
##                    <character> <character>     <character>  <character>
##   [1]                     <NA>        <NA>            <NA>         <NA>
##   [2]                        1        <NA>            <NA>         <NA>
##   [3]                        1           1 ENSE00002234944            1
##   [4]                        1           2 ENSE00003582793            1
##   [5]                        1           3 ENSE00002312635            1
##   [6]                       NA        <NA>            <NA>         <NA>
##        protein_id protein_version     ccds_id
##       <character>     <character> <character>
##   [1]        <NA>            <NA>        <NA>
##   [2]        <NA>            <NA>        <NA>
##   [3]        <NA>            <NA>        <NA>
##   [4]        <NA>            <NA>        <NA>
##   [5]        <NA>            <NA>        <NA>
##   [6]        <NA>            <NA>        <NA>
##   -------
##   seqinfo: 47 sequences (1 circular) from GRCh38 genome; no seqlengths
```

```
(gene_info <- data.frame(gr[gr$type == 'gene']) %>%
   tbl_df %>%
   dplyr::select(gene_id, gene_biotype, gene_name))
```

```
## # A tibble: 58,395 x 3
##    gene_id         gene_biotype                        gene_name
##    <chr>           <chr>                               <chr>
##  1 ENSG00000223972 transcribed_unprocessed_pseudogene DDX11L1
##  2 ENSG00000227232 unprocessed_pseudogene              WASH7P
##  3 ENSG00000278267 miRNA                               MIR6859-1
##  4 ENSG00000243485 lincRNA                             MIR1302-2HG
##  5 ENSG00000284332 miRNA                               MIR1302-2
##  6 ENSG00000237613 lincRNA                             FAM138A
##  7 ENSG00000268020 unprocessed_pseudogene              OR4G4P
##  8 ENSG00000240361 transcribed_unprocessed_pseudogene OR4G11P
```

```
##  9 ENSG00000186092 protein_coding                    OR4F5
## 10 ENSG00000238009 lincRNA                           AL627309.1
## # ... with 58,385 more rows
```

combine all the info to the DESeq2 results

```
(deseq_res %<>%
  dplyr::rename(ENTREZID = row) %>%
  left_join(., entrezid_map) %>%
  dplyr::rename(gene_id = ENSEMBL) %>%
  left_join(., gene_info))
```

```
## # A tibble: 158,780 x 13
##    ENTREZID  baseMean log2FoldChange  lfcSE     stat    pvalue     padj
##    <chr>        <dbl>          <dbl>  <dbl>    <dbl>     <dbl>    <dbl>
##  1 1           1.54            3.10   1.74    1.78    0.0750   NA
##  2 10          0              NA     NA      NA      NA        NA
##  3 100       691.             -0.522  0.153  -3.42    0.000637  0.00724
##  4 1000        0.818           0.362  2.49    0.146   0.884    NA
##  5 10000       0.0616          0.408  5.27    0.0774  0.938    NA
##  6 10000       0.0616          0.408  5.27    0.0774  0.938    NA
##  7 100008587   0              NA     NA      NA      NA        NA
##  8 100008588   0              NA     NA      NA      NA        NA
##  9 100008589   0              NA     NA      NA      NA        NA
## 10 100009601   0              NA     NA      NA      NA        NA
## # ... with 158,770 more rows, and 6 more variables: comparison <chr>,
## #   sig <chr>, gene_id <chr>, SYMBOL <chr>, gene_biotype <chr>,
## #   gene_name <chr>
```

### Inspect a few examples of diff expressed genes

downregulated in siNCBP3:

```
filter(deseq_res, padj < .1, baseMean > 100) %>%
  filter(comparison == 'NCBP3') %>%
  arrange(log2FoldChange) %>%
  dplyr::select(comparison, gene_id, gene_name, baseMean, log2FoldChange, padj)
```

```
## # A tibble: 1,278 x 6
##    comparison gene_id         gene_name baseMean log2FoldChange     padj
##    <chr>      <chr>           <chr>        <dbl>          <dbl>    <dbl>
##  1 NCBP3      ENSG00000122711 SPINK4        294.          -2.47 1.66e- 6
##  2 NCBP3      ENSG00000116690 PRG4          205.          -1.93 8.92e- 2
##  3 NCBP3      ENSG00000205403 CFI           223.          -1.82 2.66e- 6
##  4 NCBP3      ENSG00000260804 LINC01963     257.          -1.75 1.40e-11
##  5 NCBP3      ENSG00000185499 MUC1          142.          -1.72 2.67e- 2
##  6 NCBP3      ENSG00000152749 GPR180        624.          -1.67 4.02e-19
##  7 NCBP3      ENSG00000125730 C3           2554.          -1.61 2.22e- 4
##  8 NCBP3      ENSG00000203668 CHML         6310.          -1.59 4.70e- 7
##  9 NCBP3      ENSG00000105854 PON2         7081.          -1.59 5.77e-26
## 10 NCBP3      ENSG00000260896 LINC02170     378.          -1.54 4.55e- 6
## # ... with 1,268 more rows
```

```
  head
```

```
## standardGeneric for "head" defined from package "utils"
##
```

```
## function (x, ...)
## standardGeneric("head")
## <environment: 0x7fa30344cd40>
## Methods may be defined for arguments: x
## Use  showMethods("head")  for currently available ones.
```

upregulated in siNCBP3:

```
filter(deseq_res, padj < .1, baseMean > 100) %>%
  filter(comparison == 'NCBP3') %>%
  arrange(-log2FoldChange) %>%
  dplyr::select(comparison, gene_id, gene_name, baseMean, log2FoldChange, padj)
```

```
## # A tibble: 1,278 x 6
##    comparison gene_id         gene_name baseMean log2FoldChange     padj
##    <chr>      <chr>           <chr>        <dbl>          <dbl>    <dbl>
##  1 NCBP3      ENSG00000169247 SH3TC2        114.           2.05 1.52e- 6
##  2 NCBP3      ENSG00000119922 IFIT2         650.           1.96 1.92e- 4
##  3 NCBP3      ENSG00000139318 DUSP6         104.           1.87 1.33e- 9
##  4 NCBP3      ENSG00000135114 OASL          766.           1.73 4.26e- 8
##  5 NCBP3      ENSG00000237836 PHKA2-AS1     430.           1.72 1.74e-10
##  6 NCBP3      ENSG00000173702 MUC13         273.           1.71 1.92e- 2
##  7 NCBP3      ENSG00000139289 PHLDA1        210.           1.60 2.51e- 4
##  8 NCBP3      ENSG00000154654 NCAM2         157.           1.58 3.09e- 4
##  9 NCBP3      ENSG00000136235 GPNMB         229.           1.53 1.64e- 5
## 10 NCBP3      ENSG00000140287 HDC           158.           1.49 6.54e- 6
## # ... with 1,268 more rows
```

```
  head
```

```
## standardGeneric for "head" defined from package "utils"
##
## function (x, ...)
## standardGeneric("head")
## <environment: 0x7fa30344cd40>
## Methods may be defined for arguments: x
## Use  showMethods("head")  for currently available ones.
```

### total number of sig genes per gene biotype

```
deseq_res %>%
  group_by(comparison, sig, gene_biotype) %>%
  summarize(cnt=n()) %>%
  spread(comparison, cnt) %>%
  filter(sig == 'sig up') %>%
  arrange(-NCBP3) %>%
  kable
```

| sig | gene_biotype | Ars2 | Cbp20 | Cbp80 | NCBP3 | Z18 |
|---|---|---|---|---|---|---|
| sig up | protein_coding | 1186 | 236 | 856 | 675 | 1049 |
| sig up | NA | 149 | 12 | 85 | 38 | 66 |
| sig up | antisense | 76 | 11 | 41 | 15 | 16 |
| sig up | lincRNA | 59 | 4 | 31 | 11 | 14 |
| sig up | processed_transcript | 18 | 4 | 13 | 6 | 9 |
| sig up | transcribed_unprocessed_pseudogene | 12 | 2 | 14 | 3 | 10 |
| sig up | bidirectional_promoter_lncRNA | 3 | NA | 1 | 1 | NA |

| sig | gene_biotype | Ars2 | Cbp20 | Cbp80 | NCBP3 | Z18 |
|---|---|---|---|---|---|---|
| sig up | processed_pseudogene | 2 | NA | NA | 1 | NA |
| sig up | sense_intronic | 1 | NA | 1 | 1 | NA |
| sig up | sense_overlapping | 1 | NA | NA | 1 | NA |
| sig up | transcribed_processed_pseudogene | 3 | NA | 2 | 1 | 6 |
| sig up | miRNA | 11 | NA | 3 | NA | 2 |
| sig up | non_coding | 1 | NA | NA | NA | NA |
| sig up | snoRNA | 6 | NA | 4 | NA | 12 |
| sig up | snRNA | 12 | NA | 7 | NA | 9 |
| sig up | TEC | 1 | NA | NA | NA | 1 |
| sig up | transcribed_unitary_pseudogene | 3 | NA | 2 | NA | NA |
| sig up | unprocessed_pseudogene | 2 | 1 | 2 | NA | 2 |

```
deseq_res %>%
  group_by(comparison, sig, gene_biotype) %>%
  summarize(cnt=n()) %>%
  spread(comparison, cnt) %>%
  filter(sig == 'sig dn') %>%
  arrange(-NCBP3) %>%
  kable
```

| sig | gene_biotype | Ars2 | Cbp20 | Cbp80 | NCBP3 | Z18 |
|---|---|---|---|---|---|---|
| sig dn | protein_coding | 1538 | 358 | 1263 | 751 | 1509 |
| sig dn | NA | 58 | 16 | 64 | 32 | 108 |
| sig dn | antisense | 30 | 10 | 26 | 19 | 84 |
| sig dn | lincRNA | 25 | 16 | 31 | 19 | 77 |
| sig dn | processed_transcript | 12 | 2 | 12 | 4 | 22 |
| sig dn | transcribed_unprocessed_pseudogene | 6 | NA | 2 | 3 | 6 |
| sig dn | scaRNA | NA | NA | NA | 2 | NA |
| sig dn | unprocessed_pseudogene | NA | 1 | 1 | 2 | NA |
| sig dn | bidirectional_promoter_lncRNA | NA | NA | NA | 1 | 3 |
| sig dn | miRNA | 1 | NA | 2 | 1 | 2 |
| sig dn | sense_intronic | 3 | NA | 1 | 1 | 3 |
| sig dn | sense_overlapping | 1 | NA | 1 | 1 | 2 |
| sig dn | snoRNA | 18 | 5 | 52 | 1 | 3 |
| sig dn | transcribed_unitary_pseudogene | 2 | NA | NA | 1 | 2 |
| sig dn | misc_RNA | NA | NA | 1 | NA | NA |
| sig dn | polymorphic_pseudogene | NA | NA | NA | NA | 1 |
| sig dn | processed_pseudogene | NA | NA | NA | NA | 1 |
| sig dn | TR_C_gene | 1 | NA | NA | NA | NA |
| sig dn | transcribed_processed_pseudogene | 1 | NA | 2 | NA | 4 |
| sig dn | unitary_pseudogene | 1 | NA | NA | NA | NA |

**save final table**

```
save(deseq_res, file='../data/DESeq2_hg38_all_annotated_results.RData')
```

# Exon nr analysis

### Add exon count information

ie from table used by featureCounts

```
(exons <- read_tsv('/Volumes/GenomeDK/ms_tools/subread-2.0.0-Linux-x86_64/annotation/hg38_RefSeq_exon.t
```

```
## # A tibble: 261,752 x 5
##       GeneID Chr   Start   End Strand
##        <int> <chr> <int> <int> <chr>
##  1 100287102 chr1  11874 12227 +
##  2 100287102 chr1  12613 12721 +
##  3 100287102 chr1  13221 14409 +
##  4    653635 chr1  14362 14829 -
##  5    653635 chr1  14970 15038 -
##  6    653635 chr1  15796 15947 -
##  7    653635 chr1  16607 16765 -
##  8    653635 chr1  16858 17055 -
##  9    653635 chr1  17233 17368 -
## 10    653635 chr1  17606 17742 -
## # ... with 261,742 more rows
```

```
(exons %<>%
  dplyr::mutate(ENTREZID = as.character(GeneID)) %>%
  mutate(width = End-Start) %>%
  group_by(ENTREZID) %>%
  summarize(exon_cnt = n(),
            exons_width = sum(width)))
```

```
## # A tibble: 28,395 x 3
##    ENTREZID  exon_cnt exons_width
##    <chr>        <int>       <int>
##  1 1                8        1758
##  2 10               3        1415
##  3 100             13        1851
##  4 1000            20        4874
##  5 10000           23        8411
##  6 100008587        1         155
##  7 100008588        1        1868
##  8 100008589        1        5069
##  9 100009601        2          71
## 10 100009602        2          71
## # ... with 28,385 more rows
```

```
deseq_res %<>%
  left_join(., exons)
```

### exon cnt in sig up vs sig dn

```
deseq_res %>%
  filter(sig == 'sig up' | sig == 'sig dn') %>%
  group_by(comparison) %>%
  do(tidy(wilcox.test(.$exon_cnt~.$sig)))
```

```
## # A tibble: 5 x 5
```

```
## # Groups:   comparison [5]
##   comparison statistic  p.value method                          alternative
##   <chr>          <dbl>    <dbl> <fct>                           <fct>
## 1 Ars2        1705500. 1.49e-49 Wilcoxon rank sum test with c~ two.sided
## 2 Cbp20         65310  4.11e- 5 Wilcoxon rank sum test with c~ two.sided
## 3 Cbp80        808851  5.45e- 2 Wilcoxon rank sum test with c~ two.sided
## 4 NCBP3        361655  4.44e- 7 Wilcoxon rank sum test with c~ two.sided
## 5 Z18         1066060. 2.59e- 1 Wilcoxon rank sum test with c~ two.sided
```

mostly significant, but what direction

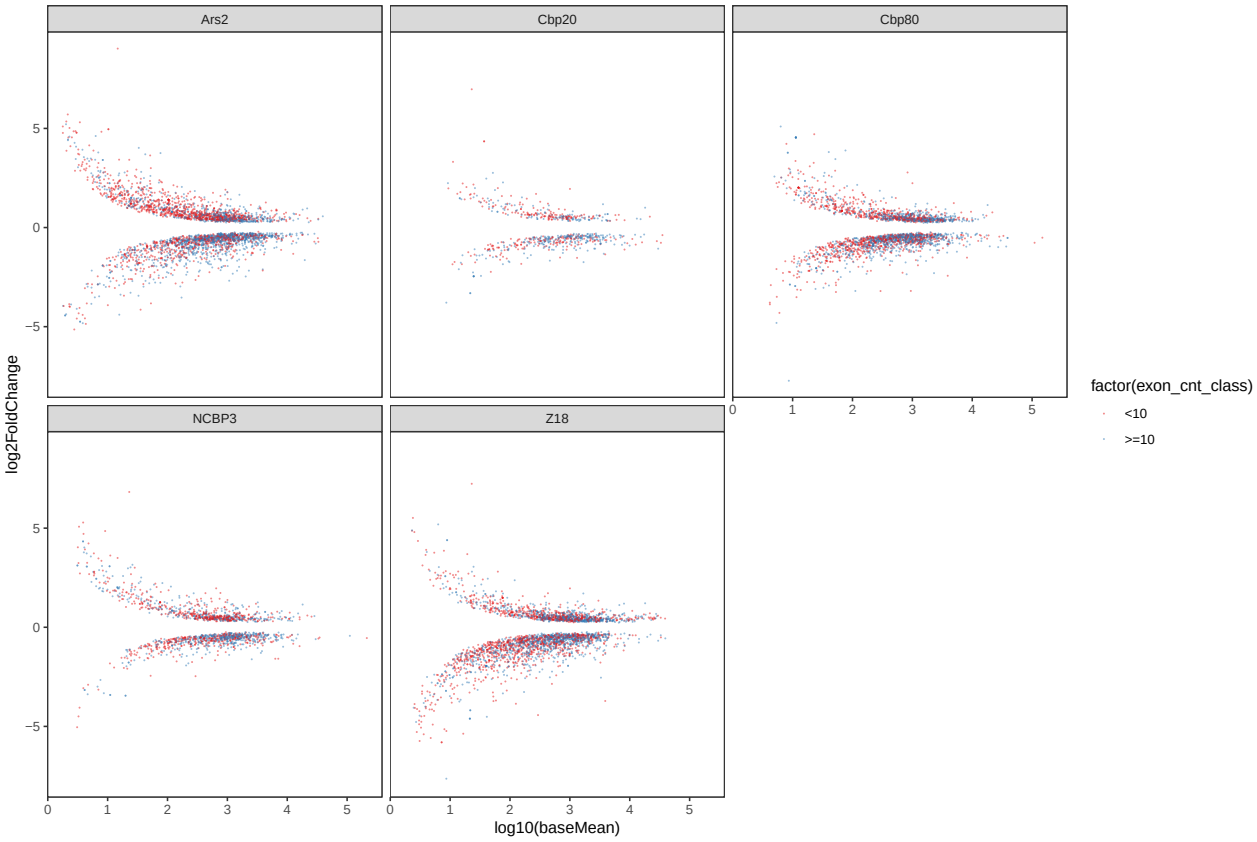
```
deseq_res %>%
  filter(sig == 'sig up' | sig == 'sig dn') %>%
  group_by(comparison) %>%
  do(tidy(wilcox.test(.$exon_cnt~.$sig, alternative = 'greater')))
```

```
## # A tibble: 5 x 5
## # Groups:   comparison [5]
##   comparison statistic  p.value method                          alternative
##   <chr>          <dbl>    <dbl> <fct>                           <fct>
## 1 Ars2        1705500. 7.43e-50 Wilcoxon rank sum test with c~ greater
## 2 Cbp20         65310  2.05e- 5 Wilcoxon rank sum test with c~ greater
## 3 Cbp80        808851  2.72e- 2 Wilcoxon rank sum test with c~ greater
## 4 NCBP3        361655  2.22e- 7 Wilcoxon rank sum test with c~ greater
## 5 Z18         1066060. 8.71e- 1 Wilcoxon rank sum test with c~ greater
```

–> all except Z18 go in the same direction

### MA plot vs exon cnt

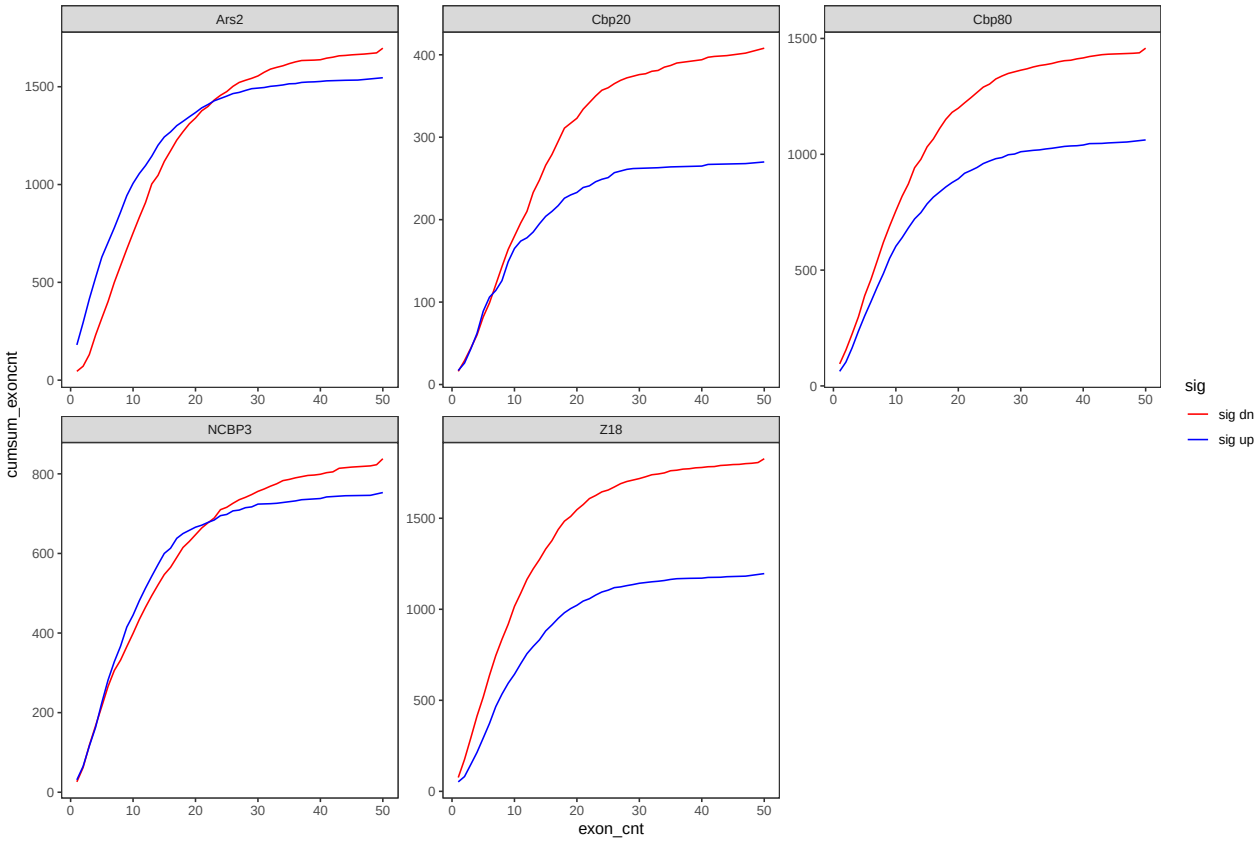
```
deseq_res %>%
  filter(sig != 'not sig') %>%
    mutate(exon_cnt_class = case_when(.$exon_cnt < 10 ~ '<10',
                                      TRUE ~ '>=10')) %>%
 ggplot(., aes(x=log10(baseMean), y=log2FoldChange, color=factor(exon_cnt_class))) +
  geom_point(alpha=.5, size=.3, shape=16) +
  scale_color_brewer(palette='Set1') +
  facet_wrap(~comparison) +
  theme_bw() +
  theme(panel.grid=element_blank())
```




this is OK, but not super-illustrative.

### Cumulative sum plot

```
deseq_res %>%
  ungroup %>%
  filter(sig != 'not sig') %>%
  mutate(exon_cnt = ifelse(exon_cnt > 50, 50, exon_cnt)) %>%
  group_by(comparison, sig, exon_cnt) %>%
  summarize(cnt=n()) %>%
  group_by(comparison, sig) %>%
  mutate(cumsum_exoncnt = cumsum(cnt),
         cumsum_freq = cumsum_exoncnt/max(cumsum_exoncnt)) %>%
  ggplot(., aes(x=exon_cnt, y=cumsum_exoncnt, color=sig)) +
  geom_line() +
  facet_wrap(~comparison, scales='free') +
  scale_color_manual(values=c('red', 'blue')) +
  theme_bw() +
  theme(panel.grid=element_blank())
```

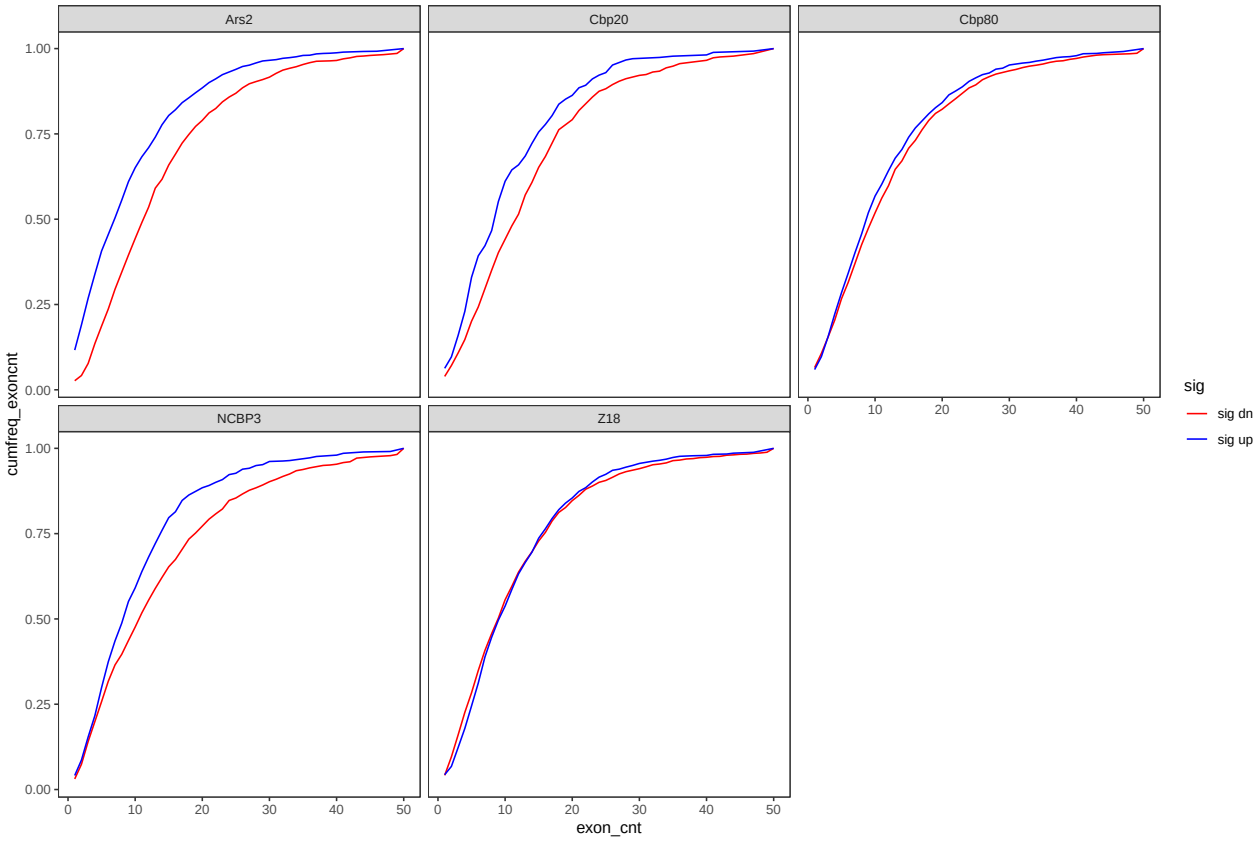



in siNCBP3 lower exon cnt sig dn depleted but generally more, bit confusing. some of the other have strong bias towards either being mostly downregulated in this analysis.

### Cumulative frequency plot

Better representation of exon cnt influence as frequencies and not raw counts

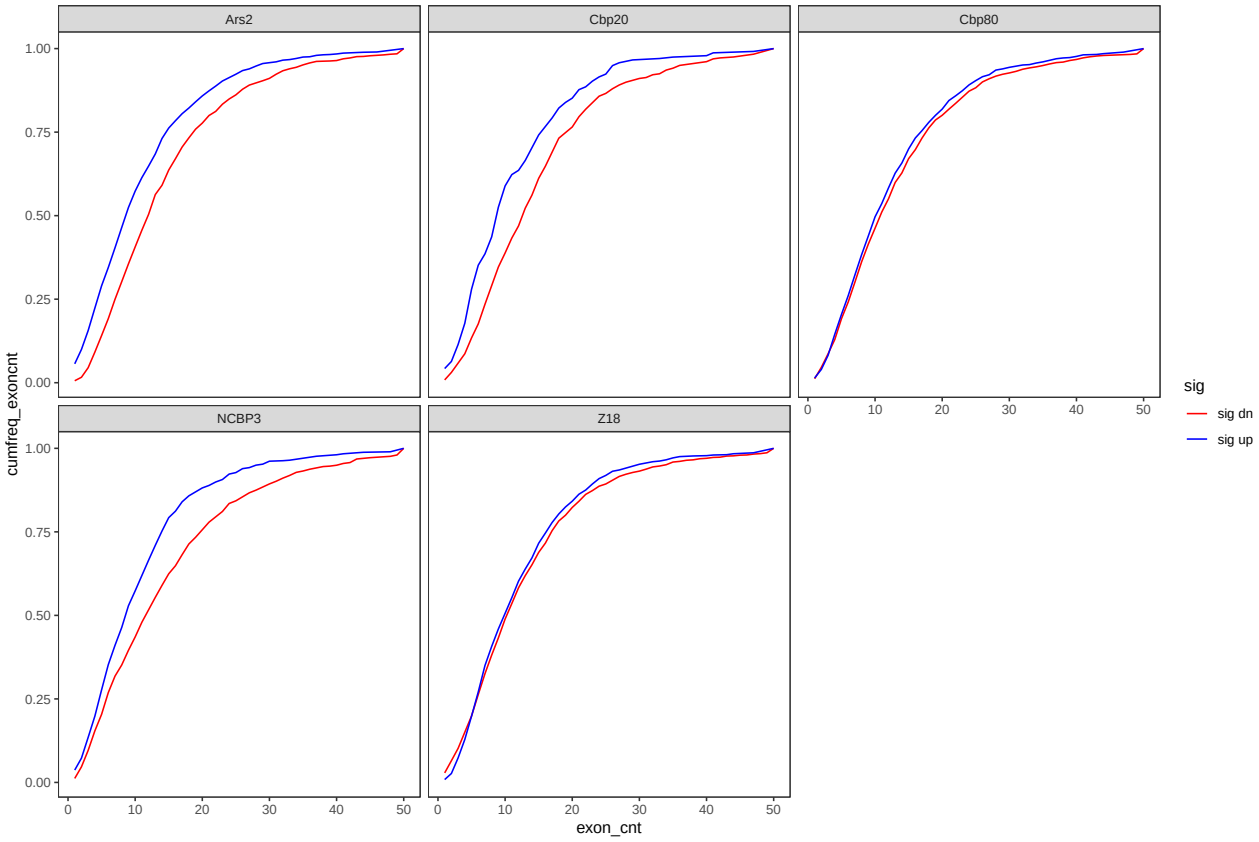
```
deseq_res %>%
  ungroup %>%
  filter(sig != 'not sig') %>%
  mutate(exon_cnt = ifelse(exon_cnt > 50, 50, exon_cnt)) %>%
  group_by(comparison, sig, exon_cnt) %>%
  summarize(cnt=n()) %>%
  group_by(comparison, sig) %>%
  mutate(cumsum_exoncnt = cumsum(cnt),
         cumfreq_exoncnt = cumsum_exoncnt/max(cumsum_exoncnt)) %>%
  ggplot(., aes(x=exon_cnt, y=cumfreq_exoncnt, color=sig)) +
  geom_line() +
  facet_wrap(~comparison) +
  scale_color_manual(values=c('red', 'blue')) +
  theme_bw() +
  theme(panel.grid=element_blank())
```




**protein-coding subset**

Protein coding genes are most relevant in this analysis, focus on those:

```
deseq_res %>%
  ungroup %>%
  filter(gene_biotype == 'protein_coding',
         sig != 'not sig') %>%
  mutate(exon_cnt = ifelse(exon_cnt > 50, 50, exon_cnt)) %>%
  group_by(comparison, sig, exon_cnt) %>%
  summarize(cnt=n()) %>%
  group_by(comparison, sig) %>%
  mutate(cumsum_exoncnt = cumsum(cnt),
         cumfreq_exoncnt = cumsum_exoncnt/max(cumsum_exoncnt)) %>%
  ggplot(., aes(x=exon_cnt, y=cumfreq_exoncnt, color=sig)) +
  geom_line() +
  facet_wrap(~comparison) +
  scale_color_manual(values=c('red', 'blue')) +
  theme_bw() +
  theme(panel.grid=element_blank())
```




statistics for protein-coding genes

```
deseq_res %>%
  ungroup %>%
  filter(gene_biotype == 'protein_coding',
         sig != 'not sig') %>%
  group_by(comparison) %>%
  do(tidy(wilcox.test(.$exon_cnt~.$sig)))
```

```
## # A tibble: 5 x 5
## # Groups:   comparison [5]
##   comparison statistic  p.value method                          alternative
##   <chr>          <dbl>    <dbl> <fct>                           <fct>
## 1 Ars2        1123665  2.29e-25 Wilcoxon rank sum test with c~ two.sided
## 2 Cbp20         52014. 1.78e- 6 Wilcoxon rank sum test with c~ two.sided
## 3 Cbp80        561236. 1.34e- 1 Wilcoxon rank sum test with c~ two.sided
## 4 NCBP3        302208. 3.29e-10 Wilcoxon rank sum test with c~ two.sided
## 5 Z18          808965  3.41e- 1 Wilcoxon rank sum test with c~ two.sided
```

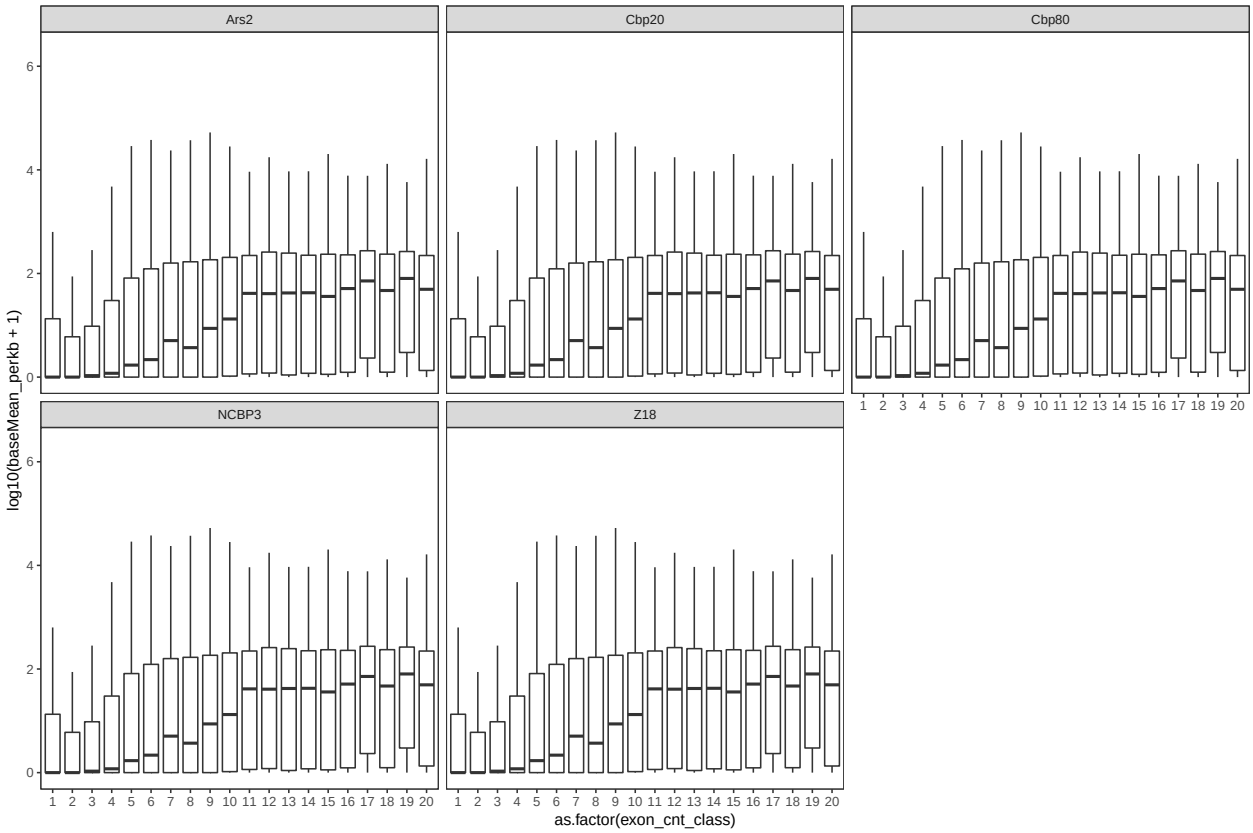
### expression bias

estimate expression as baseMean per kb of exon. baseMean is a (moderated?) average read pair count in KD and control.

```
deseq_res %<>%
  mutate(baseMean_perkb = 1000*baseMean/exons_width)
```

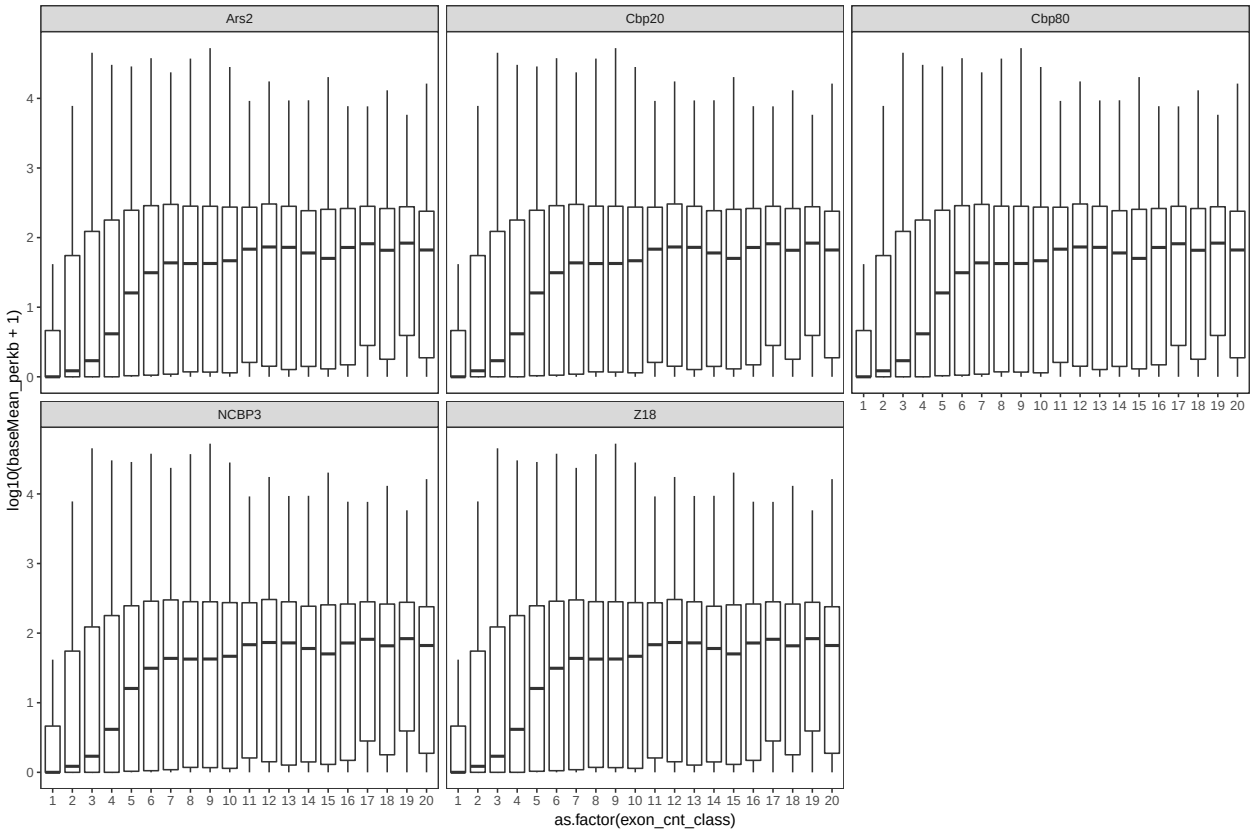
very few genes with a specific number of exons for exon cnt > 15ish, use a cutoff of combining all genes with >=20 exons.

```
deseq_res %>%
  mutate(exon_cnt_class = case_when(.$exon_cnt > 20 ~ 20,
                                    #.$exon_cnt > 15 ~ 15,
                                    TRUE ~ as.numeric(.$exon_cnt))) %>%
  ggplot(., aes(x=as.factor(exon_cnt_class), y=log10(baseMean_perkb+1))) +
  geom_boxplot(outlier.shape=NA) +
  facet_wrap(~comparison) +
  theme_bw() +
  theme(panel.grid=element_blank())
```




same for protein-coding genes only

```
deseq_res %>%
  filter(gene_biotype == 'protein_coding') %>%
  mutate(exon_cnt_class = case_when(.$exon_cnt > 20 ~ 20,
                                    #.$exon_cnt > 15 ~ 15,
                                    TRUE ~ as.numeric(.$exon_cnt))) %>%
  ggplot(., aes(x=as.factor(exon_cnt_class), y=log10(baseMean_perkb+1))) +
  geom_boxplot(outlier.shape=NA) +
  facet_wrap(~comparison) +
  theme_bw() +
  theme(panel.grid=element_blank())
```




# Expr-matched exon cnt analysis

Seems like bias in expression could explain some of the results above. Use expression-matched subset of differentially expressed genes and repeat the relevant parts.

custom function for expression-matching. Rationale is based on splitting the KD with the smallest number of genes into n (here 20) expression quantiles and for the other KDs of genes select for each quantile the same number of genes fitting this expression quantile. Finally combine the quantiles for each KD.

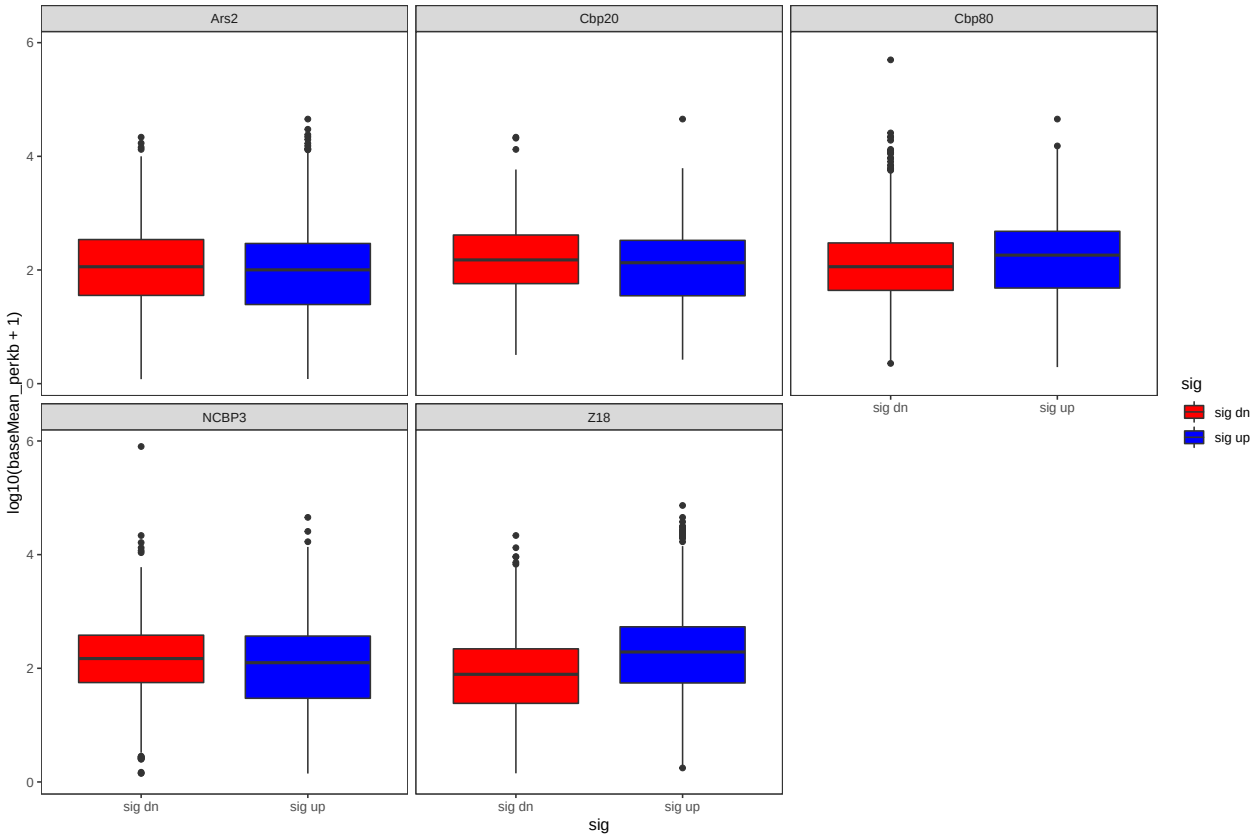
```
match_quantiles <- function(list_of_values, n_quantiles=20) {
  set.seed(0)

  set_lengths <- sapply(list_of_values, length)
  min_set <- which(set_lengths == min(set_lengths))[1]
  qs <- quantile(list_of_values[[min_set]], probs = seq(0,1,1/n_quantiles))

  matched <- lapply(1:(length(qs)-1), function(x) {
    q_sets <- lapply(list_of_values, function(values) which(values > qs[x] & values <= qs[x+1]))
    min_qsize <- min(sapply(q_sets, length))
    q_sets <- lapply(q_sets, function(q_set) sample(q_set, min_qsize))
    q_sets
  })

  res <- lapply(seq_along(list_of_values), function(i) unlist(sapply(matched, function(x) x[[i]])))

  return( res )
```

```
}
```

```
sigup_NCBP3 <- deseq_res %>%
  filter(comparison == 'NCBP3', sig == 'sig up')
sigdn_NCBP3 <- deseq_res %>%
  filter(comparison == 'NCBP3', sig == 'sig dn')


sigs_value_list <- list(sigup_NCBP3$baseMean_perkb, sigdn_NCBP3$baseMean_perkb)

sig_matched <- match_quantiles(sigs_value_list)

exprmatched_res_tbl <- bind_rows(sigup_NCBP3[sig_matched[[1]],], sigdn_NCBP3[sig_matched[[2]],])
```

### Expression sig up vs down genes before expression-matching

```
deseq_res %>%
  filter(sig != 'not sig') %>%
  ggplot(., aes(x=sig, y=log10(baseMean_perkb+1), fill=sig)) +
  geom_boxplot() +
  facet_wrap(~comparison) +
  scale_fill_manual(values = c('red', 'blue')) +
  theme_bw() +
  theme(panel.grid=element_blank())
```



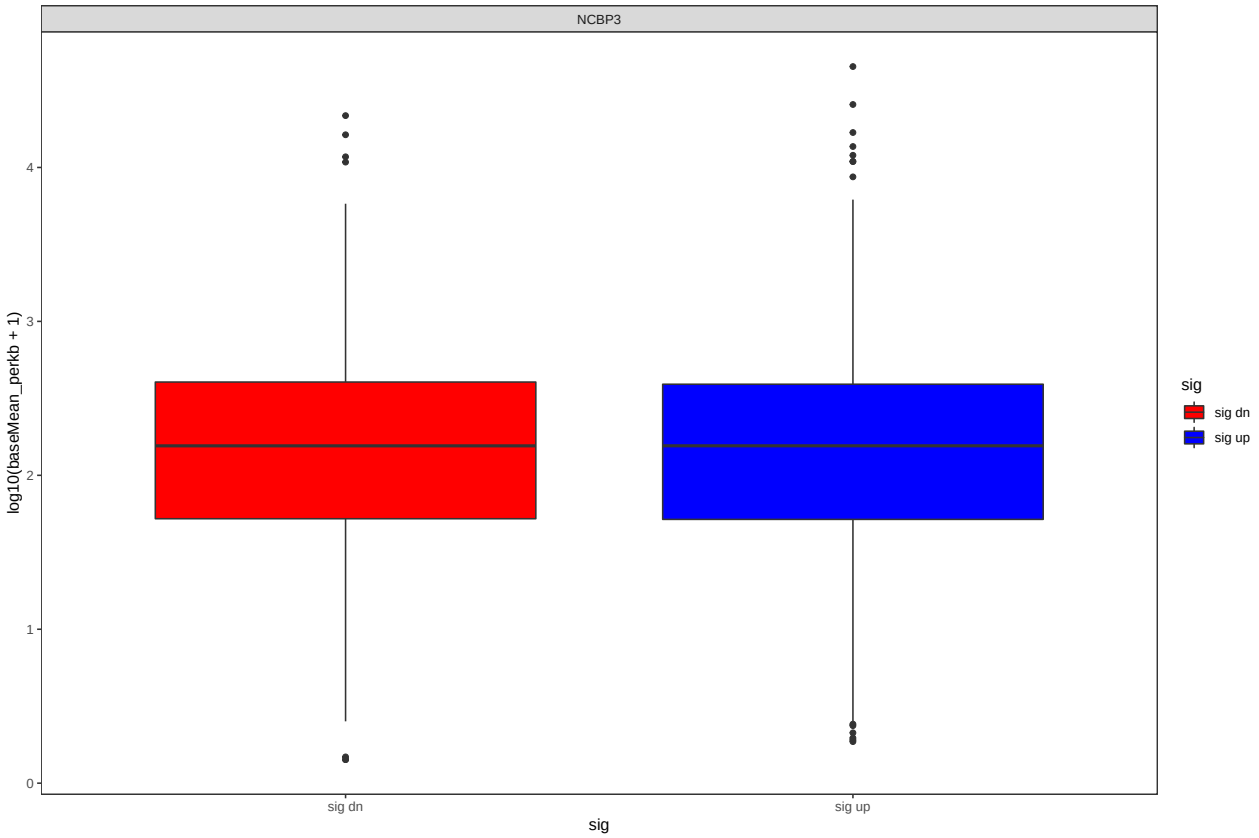

are the differences significant?

```
deseq_res %>%
  filter(sig != 'not sig') %>%
  group_by(comparison) %>%
  do(tidy(wilcox.test(.$baseMean_perkb~.$sig)))
```

```
## # A tibble: 5 x 5
## # Groups:   comparison [5]
##   comparison statistic  p.value method                          alternative
##   <chr>          <dbl>    <dbl> <fct>                           <fct>
## 1 Ars2         1412588 1.54e- 4 Wilcoxon rank sum test with c~ two.sided
## 2 Cbp20          60667 2.52e- 2 Wilcoxon rank sum test with c~ two.sided
## 3 Cbp80         693060 6.84e- 6 Wilcoxon rank sum test with c~ two.sided
## 4 NCBP3         344278 1.66e- 3 Wilcoxon rank sum test with c~ two.sided
## 5 Z18           789459 3.65e-38 Wilcoxon rank sum test with c~ two.sided
```

### Expression sig up vs down genes after expression-matching

```
exprmatched_res_tbl %>%
  ungroup %>%
  mutate(exon_cnt_class = case_when(.$exon_cnt > 20 ~ 20,
                                    .$exon_cnt > 15 ~ 15,
                                    TRUE ~ as.numeric(.$exon_cnt))) %>%
  ggplot(., aes(x=sig, y=log10(baseMean_perkb+1), fill=sig)) +
  geom_boxplot() +
  facet_wrap(~comparison) +
  scale_fill_manual(values = c('red', 'blue')) +
  theme_bw() +
  theme(panel.grid=element_blank())
```




should have no differences significant

```
exprmatched_res_tbl %>%
  filter(sig != 'not sig') %>%
  group_by(comparison) %>%
  do(tidy(wilcox.test(.$baseMean_perkb~.$sig)))
```

```
## # A tibble: 1 x 5
## # Groups:   comparison [1]
##   comparison statistic p.value method                          alternative
##   <chr>          <dbl>   <dbl> <fct>                           <fct>
## 1 NCBP3         227780   0.996 Wilcoxon rank sum test with co~ two.sided
```

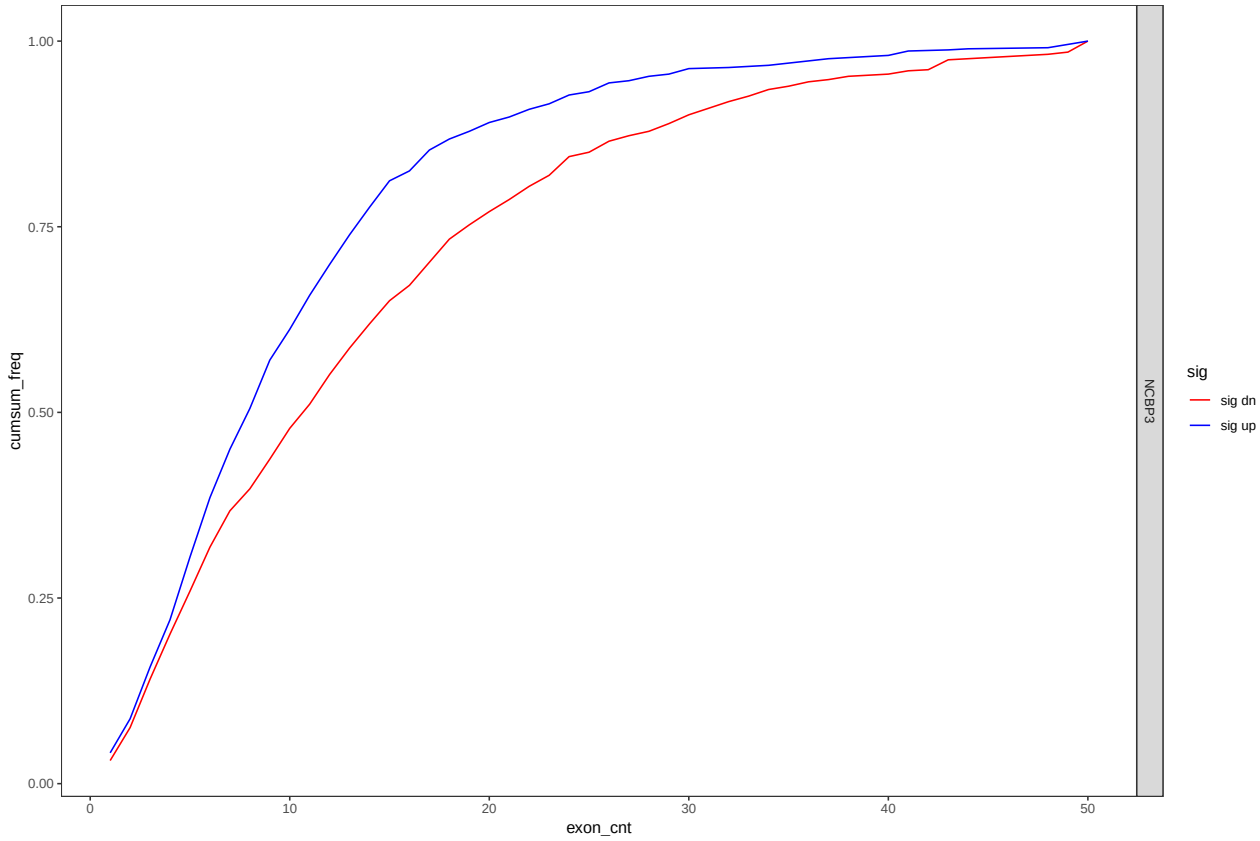
how many genes are matched up?

```
exprmatched_res_tbl %>%
  group_by(comparison, sig) %>%
  summarize(cnt=n())
```

```
## # A tibble: 2 x 3
## # Groups:   comparison [?]
##   comparison sig      cnt
##   <chr>      <chr>  <int>
## 1 NCBP3      sig dn   675
## 2 NCBP3      sig up   675
```

```
exprmatched_res_tbl %>%
  ungroup %>%
  mutate(exon_cnt = ifelse(exon_cnt > 50, 50, exon_cnt)) %>%
  group_by(comparison, sig, exon_cnt) %>%
```

```
  summarize(cnt=n()) %>%
  group_by(comparison, sig) %>%
  mutate(cumsum_exoncnt = cumsum(cnt),
         cumsum_freq = cumsum_exoncnt/max(cumsum_exoncnt)) %>%
  ggplot(., aes(x=exon_cnt, y=cumsum_freq, color=sig)) +
  geom_line() +
  facet_grid(comparison~., scales='free') +
  scale_color_manual(values = c('red', 'blue')) +
  theme_bw() +
  theme(panel.grid=element_blank())
```




wilcox test for the difference

```
exprmatched_res_tbl %>%
  group_by(comparison) %>%
  do(tidy(wilcox.test(.$exon_cnt[.$sig == 'sig up'], .$exon_cnt[.$sig == 'sig dn'])))
```

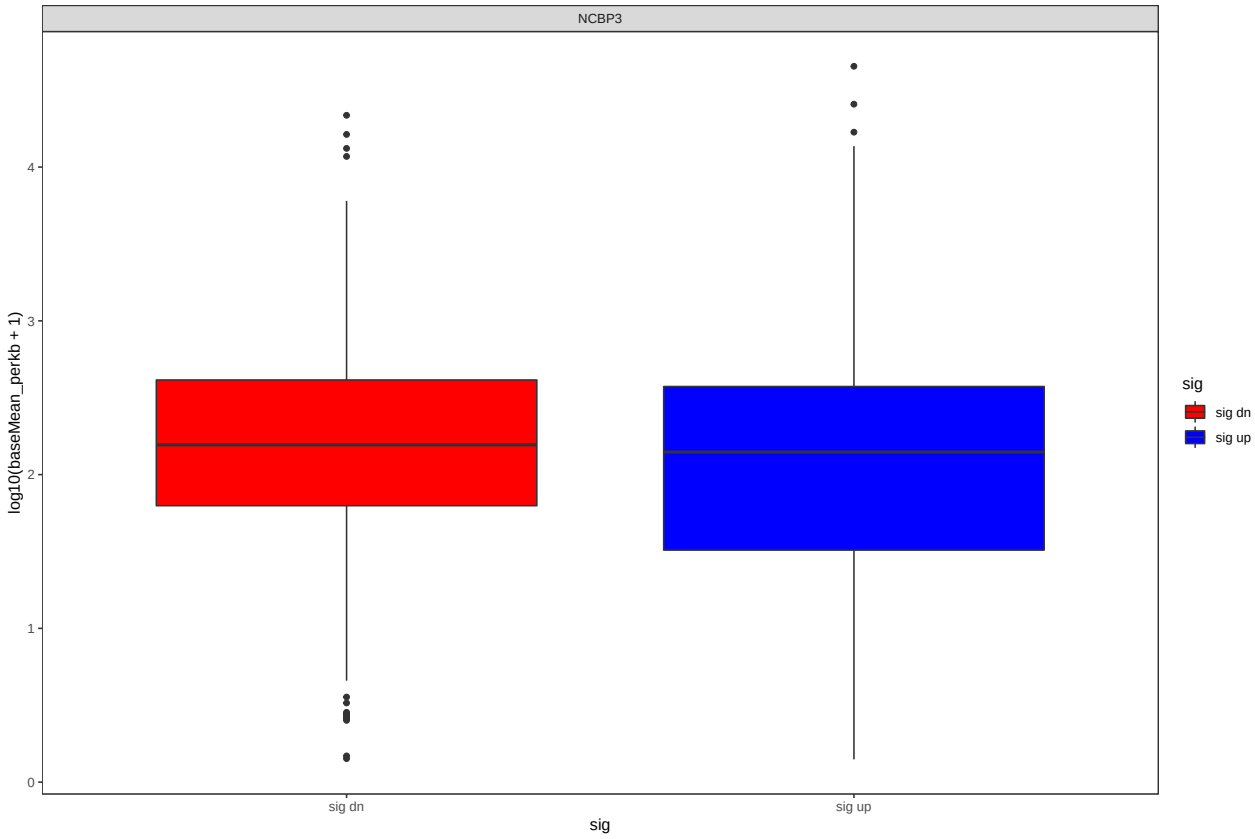
```
## # A tibble: 1 x 5
## # Groups:   comparison [1]
##   comparison statistic     p.value method                      alternative
##   <chr>          <dbl>       <dbl> <fct>                       <fct>
## 1 NCBP3         190400 0.000000170 Wilcoxon rank sum test wit~ two.sided
```

Note that the exact p-value varies for each expression-matched depends on the random generator seed.

**expr matched protein-coding genes**

```
sigup_NCBP3 <- deseq_res %>%
  filter(gene_biotype == 'protein_coding', comparison == 'NCBP3', sig == 'sig up')
sigdn_NCBP3 <- deseq_res %>%
  filter(gene_biotype == 'protein_coding', comparison == 'NCBP3', sig == 'sig dn')


sigs_value_list <- list(sigup_NCBP3$baseMean_perkb, sigdn_NCBP3$baseMean_perkb)

sig_matched <- match_quantiles(sigs_value_list)

exprmatched_res_tbl <- bind_rows(sigup_NCBP3[sig_matched[[1]],], sigdn_NCBP3[sig_matched[[2]],])
```

```
deseq_res %>%
  filter(gene_biotype == 'protein_coding', comparison == 'NCBP3', sig != 'not sig') %>%
  ungroup %>%
  mutate(exon_cnt_class = case_when(.$exon_cnt > 20 ~ 20,
                                    .$exon_cnt > 15 ~ 15,
                                    TRUE ~ as.numeric(.$exon_cnt))) %>%
  ggplot(., aes(x=sig, y=log10(baseMean_perkb+1), fill=sig)) +
  geom_boxplot() +
  facet_wrap(~comparison) +
  scale_fill_manual(values = c('red', 'blue')) +
  theme_bw() +
  theme(panel.grid=element_blank())
```




is the difference in expression of protein-coding genes significant?

```
deseq_res %>%
  filter(gene_biotype == 'protein_coding', sig != 'not sig') %>%
  group_by(comparison) %>%
  do(tidy(wilcox.test(.$baseMean_perkb~.$sig)))
```

```
## # A tibble: 5 x 5
## # Groups:   comparison [5]
##   comparison statistic  p.value method                          alternative
##   <chr>          <dbl>    <dbl> <fct>                           <fct>
## 1 Ars2          936369 2.32e- 1 Wilcoxon rank sum test with c~ two.sided
## 2 Cbp20          45964 6.92e- 2 Wilcoxon rank sum test with c~ two.sided
## 3 Cbp80         413917 5.01e-20 Wilcoxon rank sum test with c~ two.sided
## 4 NCBP3         277500 1.96e- 3 Wilcoxon rank sum test with c~ two.sided
## 5 Z18           600725 3.00e-25 Wilcoxon rank sum test with c~ two.sided
```

did expression-matching work? ie should have no differences significant after matching

```
exprmatched_res_tbl %>%
  filter(sig != 'not sig') %>%
  group_by(comparison) %>%
  do(tidy(wilcox.test(.$baseMean_perkb~.$sig)))
```

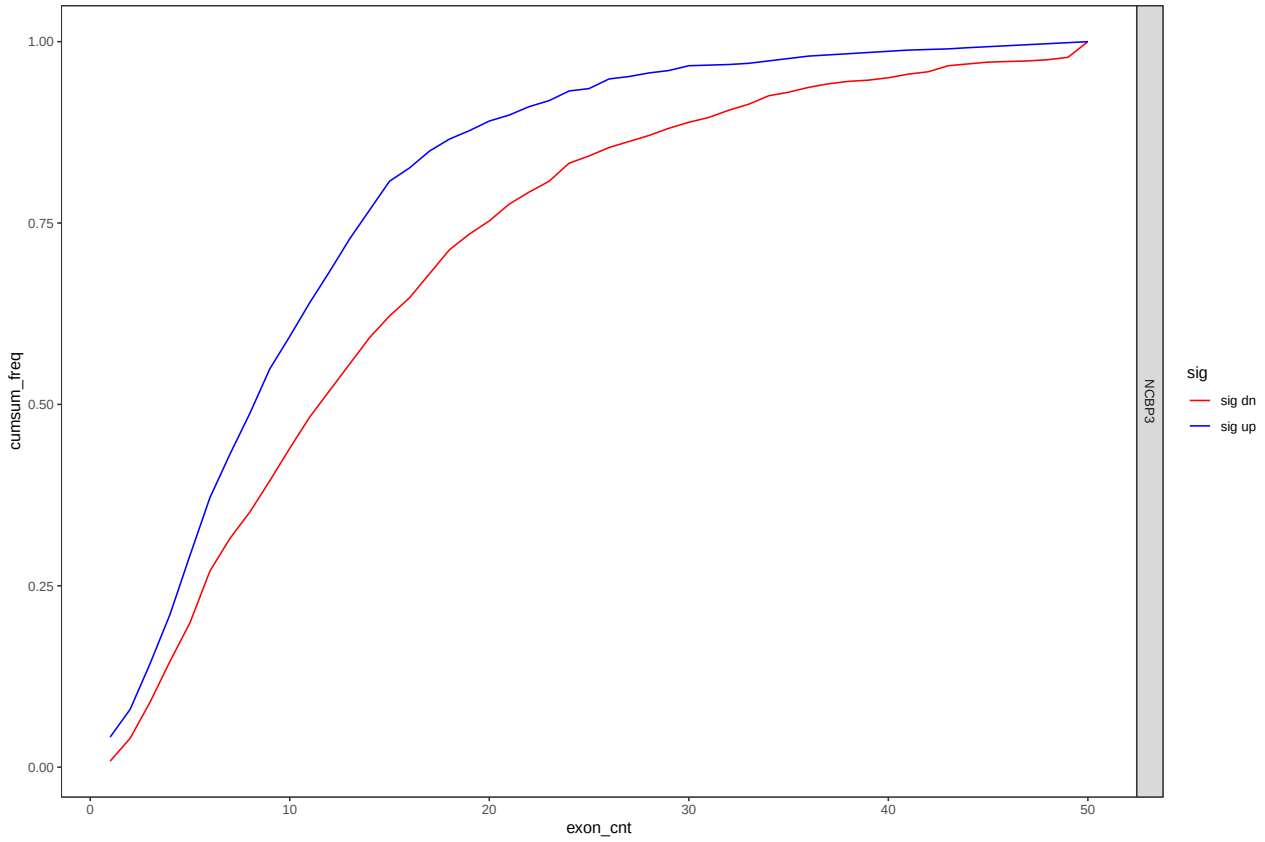
```
## # A tibble: 1 x 5
## # Groups:   comparison [1]
##   comparison statistic p.value method                          alternative
##   <chr>          <dbl>   <dbl> <fct>                           <fct>
## 1 NCBP3         182051   0.968 Wilcoxon rank sum test with co~ two.sided
```

how many genes are matched up?

```
exprmatched_res_tbl %>%
  group_by(comparison, sig) %>%
  summarize(cnt=n())
```

```
## # A tibble: 2 x 3
## # Groups:   comparison [?]
##   comparison sig      cnt
##   <chr>      <chr>  <int>
## 1 NCBP3      sig dn   603
## 2 NCBP3      sig up   603
```

```
exprmatched_res_tbl %>%
  ungroup %>%
  mutate(exon_cnt = ifelse(exon_cnt > 50, 50, exon_cnt)) %>%
  group_by(comparison, sig, exon_cnt) %>%
  summarize(cnt=n()) %>%
  group_by(comparison, sig) %>%
  mutate(cumsum_exoncnt = cumsum(cnt),
         cumsum_freq = cumsum_exoncnt/max(cumsum_exoncnt)) %>%
  ggplot(., aes(x=exon_cnt, y=cumsum_freq, color=sig)) +
  geom_line() +
  facet_grid(comparison~., scales='free') +
  scale_color_manual(values = c('red', 'blue')) +
  theme_bw() +
  theme(panel.grid=element_blank())
```




wilcox test for the difference

```
exprmatched_res_tbl %>%
  group_by(comparison) %>%
  do(tidy(wilcox.test(.$exon_cnt[.$sig == 'sig up'], .$exon_cnt[.$sig == 'sig dn'])))
```

```
## # A tibble: 1 x 5
## # Groups:   comparison [1]
##   comparison statistic  p.value method                          alternative
##   <chr>          <dbl>    <dbl> <fct>                           <fct>
## 1 NCBP3        141444. 2.38e-11 Wilcoxon rank sum test with c~ two.sided
```

Note that the exact p-value varies for each expression-matched depends on the random generator seed.

# sessionInfo

```
sessionInfo()
```

```
## R version 3.5.0 (2018-04-23)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS  10.14.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
```

```
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] rtracklayer_1.40.3          AnnotationHub_2.12.1
##  [3] org.Hs.eg.db_3.7.0          AnnotationDbi_1.44.0
##  [5] bindrcpp_0.2.2              RColorBrewer_1.1-2
##  [7] limma_3.36.1                DESeq2_1.20.0
##  [9] SummarizedExperiment_1.10.1 DelayedArray_0.6.0
## [11] BiocParallel_1.14.1         matrixStats_0.53.1
## [13] Biobase_2.40.0              GenomicRanges_1.32.3
## [15] GenomeInfoDb_1.16.0         IRanges_2.14.10
## [17] S4Vectors_0.18.3            BiocGenerics_0.26.0
## [19] broom_0.4.4                 magrittr_1.5
## [21] knitr_1.20                  forcats_0.3.0
## [23] stringr_1.3.1               dplyr_0.7.5
## [25] purrr_0.2.5                 readr_1.1.1
## [27] tidyr_0.8.1                 tibble_1.4.2
## [29] ggplot2_3.1.0               tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] colorspace_1.3-2              rprojroot_1.3-2
##  [3] htmlTable_1.12                XVector_0.20.0
##  [5] base64enc_0.1-3               rstudioapi_0.7
##  [7] bit64_0.9-7                   interactiveDisplayBase_1.18.0
##  [9] lubridate_1.7.4               xml2_1.2.0
## [11] splines_3.5.0                 mnormt_1.5-5
## [13] geneplotter_1.58.0            Formula_1.2-3
## [15] jsonlite_1.5                  Rsamtools_1.32.0
## [17] annotate_1.58.0               cluster_2.0.7-1
## [19] shiny_1.1.0                   compiler_3.5.0
## [21] httr_1.3.1                    backports_1.1.2
## [23] assertthat_0.2.0              Matrix_1.2-14
## [25] lazyeval_0.2.1                cli_1.0.0
## [27] later_0.7.3                   acepack_1.4.1
## [29] htmltools_0.3.6               tools_3.5.0
## [31] gtable_0.2.0                  glue_1.2.0
## [33] GenomeInfoDbData_1.1.0        reshape2_1.4.3
## [35] Rcpp_0.12.17                  cellranger_1.1.0
## [37] Biostrings_2.48.0             nlme_3.1-137
## [39] psych_1.8.4                   rvest_0.3.2
## [41] mime_0.5                      XML_3.98-1.11
## [43] zlibbioc_1.26.0               scales_0.5.0
## [45] BSgenome_1.48.0               BiocInstaller_1.30.0
## [47] promises_1.0.1                hms_0.4.2
## [49] curl_3.2                      yaml_2.1.19
## [51] memoise_1.1.0                 gridExtra_2.3
## [53] rpart_4.1-13                  latticeExtra_0.6-28
## [55] stringi_1.2.3                 RSQLite_2.1.1
## [57] highr_0.7                     genefilter_1.62.0
## [59] checkmate_1.8.5               rlang_0.2.1
```

```
## [61] pkgconfig_2.0.1              bitops_1.0-6
## [63] evaluate_0.10.1              lattice_0.20-35
## [65] bindr_0.1.1                  GenomicAlignments_1.16.0
## [67] htmlwidgets_1.2              labeling_0.3
## [69] bit_1.1-14                   tidyselect_0.2.4
## [71] plyr_1.8.4                   R6_2.2.2
## [73] Hmisc_4.1-1                  DBI_1.0.0
## [75] pillar_1.2.3                 haven_1.1.1
## [77] foreign_0.8-70               withr_2.1.2
## [79] survival_2.42-3              RCurl_1.95-4.10
## [81] nnet_7.3-12                  modelr_0.1.2
## [83] crayon_1.3.4                 utf8_1.1.4
## [85] rmarkdown_1.10               locfit_1.5-9.1
## [87] grid_3.5.0                   readxl_1.1.0
## [89] data.table_1.11.4            blob_1.1.1
## [91] digest_0.6.15                xtable_1.8-2
## [93] httpuv_1.4.5                 munsell_0.5.0
```