RNA- and ChIP-seq study: to which degree differential H3K36me3 associate with DGE?

Pedro L. Baldoni

03 March, 2021

Contents

Here, I will assess to which degree differential enrichment for H3K36me3 associate with differential gene expression (DGE) levels. The main question I want to answer is: can we use differential H3K36me3 enrichment to infer DGE, and vice-versa?

To answer this question, I will use RNA-seq and ChIP-seq data from cell lines Helas3 and Hepg2 (two of the cell lines reported in our manuscripts). There are two isogenic replicates from each assay. RNA-seq data were

previously quantified with Salmon. PCR duplicates and low-quality reads were removed from ChIP-seq data.

1 RNA-seq data preprocessing

The analysis of RNA-seq data will follow the steps from the DESeq2 vignette.

1.1 Preparing the data

```
# Loading some libraries
library(EnsDb.Hsapiens.v75)
library(tximport)
library(pheatmap)
library(DESeq2)
library(vsn)
library(data.table)
library(ggplot2)
library(apeglm)
```

```
# Reading in Salmon's qf files

files <- list.files('../../Public/Salmon','^quant.sf$',recursive = T,full.names = T)

for(i in files){
  cellname = strsplit(i,'/')[[1]][length(strsplit(i,'/')[[1]])-2]
  repname = strsplit(i,'/')[[1]][length(strsplit(i,'/')[[1]])-1]
  repname = substr(repname,nchar(repname),nchar(repname))
  names(files)[which(files==i)] = paste0(cellname,'.Rep',repname)
}

files
```

```
                                H1hesc.Rep1
"../../Public/Salmon/H1hesc/Output1/quant.sf"
                                H1hesc.Rep2
"../../Public/Salmon/H1hesc/Output2/quant.sf"
```

```
                                  Helas3.Rep1
"../../Public/Salmon/Helas3/Output1/quant.sf"
                                  Helas3.Rep2
"../../Public/Salmon/Helas3/Output2/quant.sf"
                                   Hepg2.Rep1
 "../../Public/Salmon/Hepg2/Output1/quant.sf"
                                   Hepg2.Rep2
 "../../Public/Salmon/Hepg2/Output2/quant.sf"
                                   Huvec.Rep1
 "../../Public/Salmon/Huvec/Output1/quant.sf"
                                   Huvec.Rep2
 "../../Public/Salmon/Huvec/Output2/quant.sf"
```

```
# Creating annotation table & calling tximport
Tx <- transcripts(EnsDb.Hsapiens.v75,return.type='data.frame')
tx2gene = subset(Tx,select=c('tx_id','gene_id'))


txi.salmon <- tximport(files, type = "salmon", tx2gene = tx2gene)
```

```
reading in files with read_tsv
1 2 3 4 5 6 7 8
summarizing abundance
summarizing counts
summarizing length
```

```
head(txi.salmon$counts)
```

```
                H1hesc.Rep1 H1hesc.Rep2 Helas3.Rep1 Helas3.Rep2 Hepg2.Rep1
ENSG00000000003    8738.250    4656.098    9022.340    6107.521   9479.250
ENSG00000000005      73.000      36.000       0.000       0.000      0.000
ENSG00000000419    2050.091     898.301    9384.999    6086.000   4416.513
ENSG00000000457     908.429     766.651    1835.313    3028.256    798.903
ENSG00000000460    5592.126    3086.932    9610.124   10472.699   1496.057
ENSG00000000938     145.003     115.024       2.000       5.000      0.000
                Hepg2.Rep2 Huvec.Rep1 Huvec.Rep2
```

```
ENSG00000000003   6012.293   5066.381   7245.946
ENSG00000000005      0.000      0.000      0.000
ENSG00000000419   3620.364    674.270   1280.048
ENSG00000000457    772.587    652.533    857.948
ENSG00000000460   1135.364   1193.470   1725.631
ENSG00000000938      3.000     19.000     29.000
```

```
# Creating DESeq2 object
sampleTable <-
  data.frame(condition = factor(unlist(lapply(
    strsplit(colnames(txi.salmon$counts), "\\."),
    FUN = function(x) {
      x[1]
    }
  ))))
rownames(sampleTable) <- colnames(txi.salmon$counts)
sampleTable
```

```
            condition
H1hesc.Rep1    H1hesc
H1hesc.Rep2    H1hesc
Helas3.Rep1    Helas3
Helas3.Rep2    Helas3
Hepg2.Rep1      Hepg2
Hepg2.Rep2      Hepg2
Huvec.Rep1      Huvec
Huvec.Rep2      Huvec
```

```
dds <-
  DESeqDataSetFromTximport(txi.salmon, sampleTable, design =  ~ condition)
```

```
using counts and average transcript lengths from tximport
```

```
rowRanges(dds) <- genes(EnsDb.Hsapiens.v75)[rownames(dds)]
seqlevelsStyle(dds) <- 'UCSC'
head(assay(dds))
```

```
                H1hesc.Rep1 H1hesc.Rep2 Helas3.Rep1 Helas3.Rep2 Hepg2.Rep1
ENSG00000000003        8738        4656        9022        6108        9479
ENSG00000000005          73          36           0           0           0
ENSG00000000419        2050         898        9385        6086        4417
ENSG00000000457         908         767        1835        3028         799
ENSG00000000460        5592        3087        9610       10473        1496
ENSG00000000938         145         115           2           5           0
                Hepg2.Rep2 Huvec.Rep1 Huvec.Rep2
ENSG00000000003       6012       5066       7246
ENSG00000000005          0          0          0
ENSG00000000419       3620        674       1280
ENSG00000000457        773        653        858
ENSG00000000460       1135       1193       1726
ENSG00000000938          3         19         29
```

1.2 Pre-filtering

```
# Selecting only the relevant cell lines
dds <- dds[, colData(dds)$condition %in% c('Helas3','Hepg2')]
dds$condition <- droplevels(dds$condition)
dds$condition <- relevel(dds$condition, ref = "Helas3")
head(assay(dds))
```

```
                Helas3.Rep1 Helas3.Rep2 Hepg2.Rep1 Hepg2.Rep2
ENSG00000000003        9022        6108       9479       6012
ENSG00000000005           0           0          0          0
ENSG00000000419        9385        6086       4417       3620
ENSG00000000457        1835        3028        799        773
ENSG00000000460        9610       10473       1496       1135
ENSG00000000938           2           5          0          3
```

```
# Filtering lowly expressed genes
keep.cts <- rowSums(counts(dds)) >= 25
dds <- dds[keep.cts,]
```

```
head(assay(dds))
```

```
                Helas3.Rep1 Helas3.Rep2 Hepg2.Rep1 Hepg2.Rep2
ENSG00000000003        9022        6108       9479       6012
ENSG00000000419        9385        6086       4417       3620
ENSG00000000457        1835        3028        799        773
ENSG00000000460        9610       10473       1496       1135
ENSG00000000971       75125       47441          3          3
ENSG00000001036       10888        6812      11376       8168
```

```
# Keeping only protein coding genes
keep.coding <- rowRanges(dds)$gene_biotype == 'protein_coding'
dds <- dds[keep.coding,]
dim(dds)
```

```
[1] 16252     4
```

```
# Selecting only relevant chromosomes (chrY is not present here)
keep.chr <- seqnames(dds) %in% paste0('chr',c(1:22,'X'))
dds <- dds[keep.chr,]
dim(dds)
```

```
[1] 15476     4
```

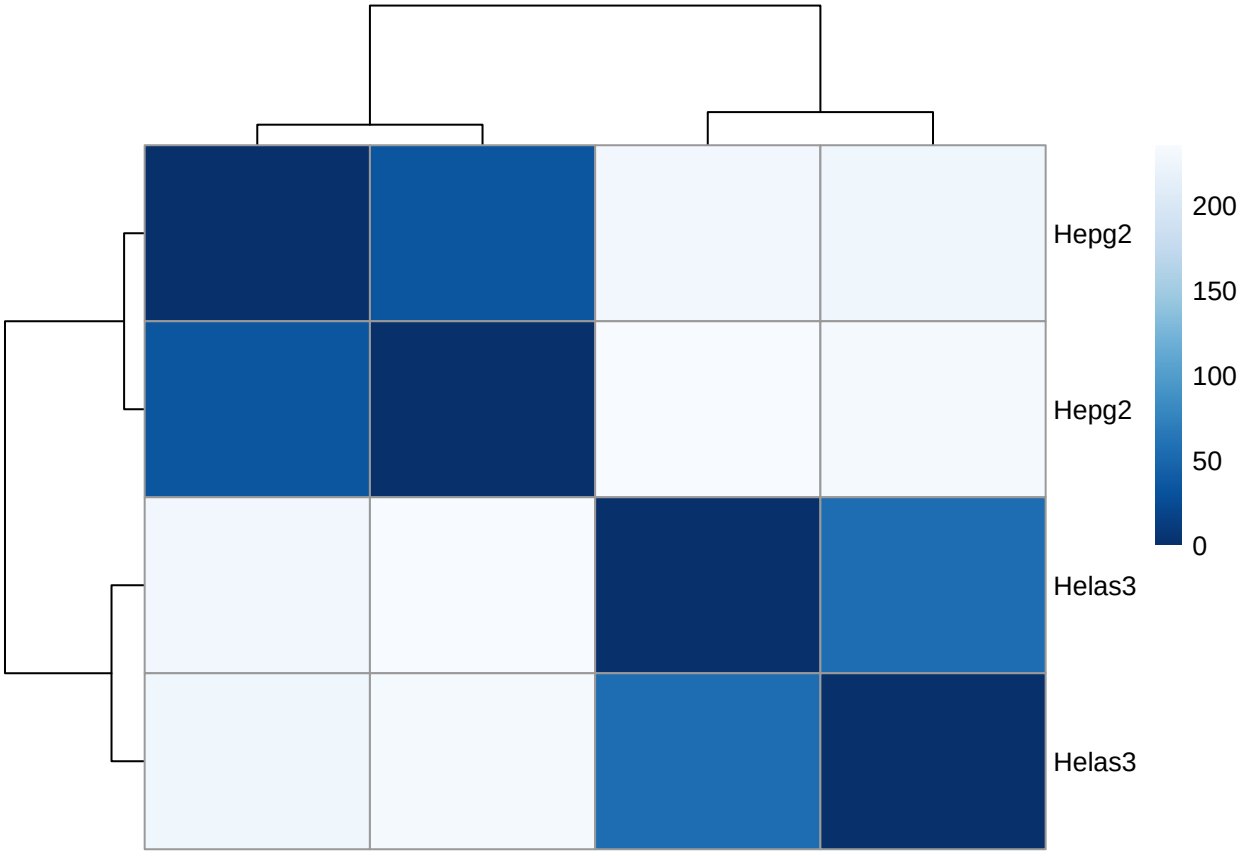
1.3 Visualization

```
# Sample-to-sample distances
vsd <- vst(dds, blind=FALSE)
```

```
using 'avgTxLength' from assays(dds), correcting for library size
```

```
sampleDists <- dist(t(assay(vsd)))
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsd$condition, sep="-")
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(RColorBrewer::brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
```

```
         clustering_distance_rows=sampleDists,
         clustering_distance_cols=sampleDists,
         col=colors)
```



1.4 DGE

Now, I will run DESeq2 to find DEG between the two cell lines. I expect to see lots of DGE genes since we are comparing two very distinct cell lines.

```
# Calling DESeq2
dds <- DESeq(dds)
```

```
estimating size factors
using 'avgTxLength' from assays(dds), correcting for library size
estimating dispersions
gene-wise dispersion estimates
mean-dispersion relationship
```

```
final dispersion estimates

fitting model and testing
```

```
res <- results(dds)
res
```

```
log2 fold change (MLE): condition Hepg2 vs Helas3
Wald test p-value: condition Hepg2 vs Helas3
DataFrame with 15476 rows and 6 columns
                 baseMean log2FoldChange     lfcSE      stat       pvalue
                <numeric>      <numeric> <numeric> <numeric>    <numeric>
ENSG00000000003   7743.03       0.550607  0.317169   1.73601  8.25627e-02
ENSG00000000419   5597.28      -0.378313  0.289196  -1.30815  1.90821e-01
ENSG00000000457   1616.65      -1.653539  0.336689  -4.91118  9.05301e-07
ENSG00000000460   4927.28      -2.346662  0.292420  -8.02498  1.01539e-15
ENSG00000000971  28178.46     -14.095082  0.658630 -21.40062 1.31857e-101
...                   ...            ...       ...       ...          ...
ENSG00000273173  5143.820      15.857670  1.465342 10.821821  2.71332e-27
ENSG00000273259   884.158      13.330312  1.473836  9.044640  1.50159e-19
ENSG00000273274   761.458       0.678806  0.313572  2.164756  3.04063e-02
ENSG00000273294   506.355       2.040432  0.377608  5.403572  6.53269e-08
ENSG00000273439   178.013       0.254591  0.413813  0.615231  5.38402e-01
                      padj
                 <numeric>
ENSG00000000003 1.20247e-01
ENSG00000000419 2.53011e-01
ENSG00000000457 2.90733e-06
ENSG00000000460 6.89220e-15
ENSG00000000971 1.12741e-99
...                     ...
ENSG00000273173 3.37822e-26
ENSG00000273259 1.28248e-18
ENSG00000273274 4.90022e-02
ENSG00000273294 2.36768e-07
```

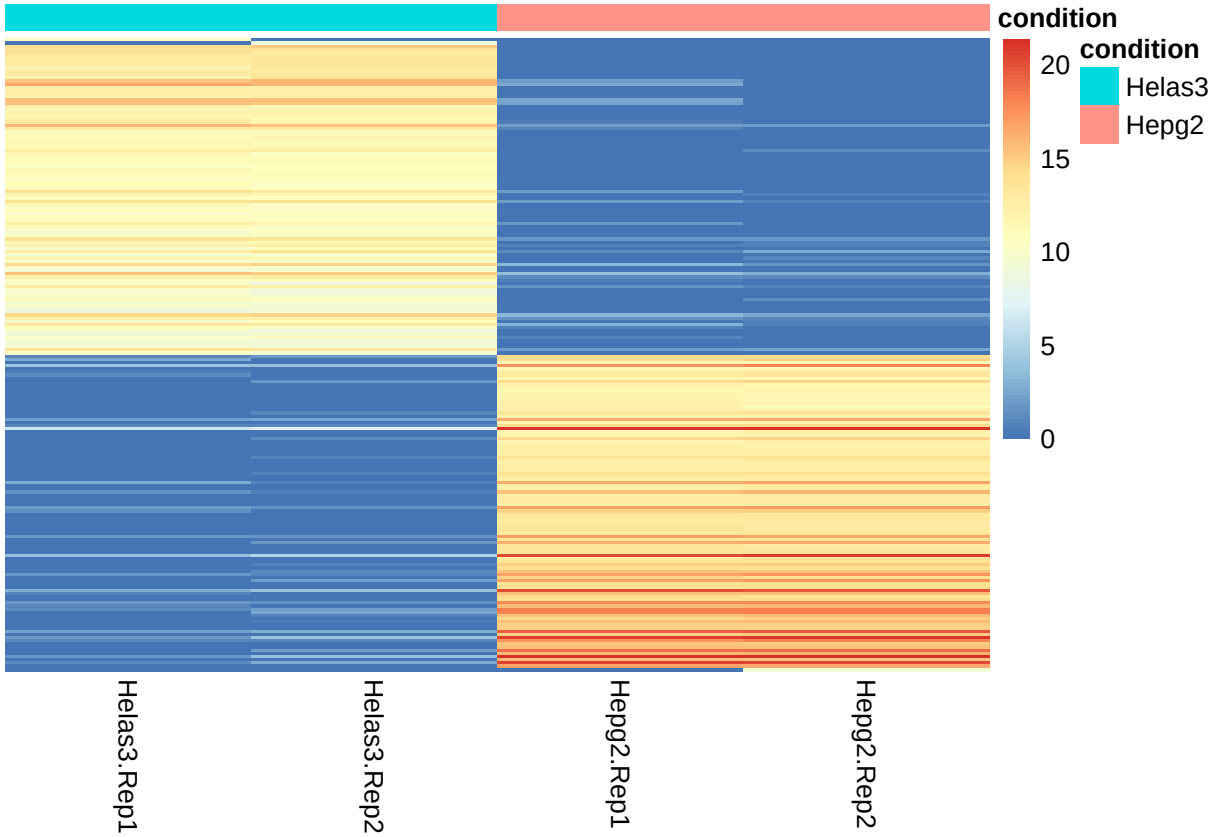
```
ENSG00000273439 6.14115e-01
```

```
# How many DGE are there? Let's set a p-value cutoff of 0.01
cutoff <- 0.01
sum(res$padj<cutoff)
```

```
[1] 8261
```

Let's take a look at a heatmap of some of these genes.

```
# Normalizing cts
ntd <- normTransform(dds)

# Selecting some genes with largest absolute LFC differences for visualization
sig.res <- res[res$padj<cutoff,]
sig.res <- sig.res[order(sig.res$log2FoldChange),]
select <- c(rownames(head(sig.res,100)),rownames(tail(sig.res,100)))

# Set up annotation
df <- as.data.frame(colData(dds)[,c("condition")])
colnames(df) <- 'condition'
rownames(df) <- colnames(dds)

# Plotting
pheatmap(assay(ntd)[select,],cluster_rows=FALSE, show_rownames=FALSE,
         cluster_cols=FALSE, annotation_col=df)
```



2 ChIP-seq data preprocessing

I will follow the same analysis steps for RNA-seq data here. H3K36me3 is expected to be enriched only on (expressed?) gene bodies, so it is reasonable to use gene bodies as an annotation reference. To this end, I will compute ChIP-seq read counts mapping onto the gene bodies as suggested in the DESeq2 vignette here

2.1 Preparing the data

```
# Reading in ChIP-seq bam files

files <- list.files('../../Data/','^wgEncode*.*.bam$',recursive = T,full.names = T)
files <- files[grep('H3K36me3',files)]
files <- files[grep('helas3|hepg2',files)]

names(files) <- c(paste0('Helas3.Rep',1:2),paste0('Hepg2.Rep',1:2))
```

```
files
```

```
                                                                                                Helas3.Rep
"../../Data//Encode_helas3/H3K36me3/wgEncodeBroadHistoneHelas3H3k36me3StdAlnRep1.markdup.q10.sorted.bam
                                                                                                Helas3.Rep
"../../Data//Encode_helas3/H3K36me3/wgEncodeBroadHistoneHelas3H3k36me3StdAlnRep2.markdup.q10.sorted.bam
                                                                                                 Hepg2.Rep
  "../../Data//Encode_hepg2/H3K36me3/wgEncodeBroadHistoneHepg2H3k36me3StdAlnRep1.markdup.q10.sorted.bam
                                                                                                 Hepg2.Rep
  "../../Data//Encode_hepg2/H3K36me3/wgEncodeBroadHistoneHepg2H3k36me3StdAlnRep2.markdup.q10.sorted.bam
```

```
# Counting reads with bamsignals
chip.cts <- list()
for (i in seq_len(length(files))) {
  chip.cts[[i]] <- bamsignals::bamCount(files[i],rowRanges(dds))
}
```

```
Processing ../../Data//Encode_helas3/H3K36me3/wgEncodeBroadHistoneHelas3H3k36me3StdAlnRep1.markdup.q10.s
Processing ../../Data//Encode_helas3/H3K36me3/wgEncodeBroadHistoneHelas3H3k36me3StdAlnRep2.markdup.q10.s
Processing ../../Data//Encode_hepg2/H3K36me3/wgEncodeBroadHistoneHepg2H3k36me3StdAlnRep1.markdup.q10.so
Processing ../../Data//Encode_hepg2/H3K36me3/wgEncodeBroadHistoneHepg2H3k36me3StdAlnRep2.markdup.q10.so
```

```
chip.cts <- do.call(cbind,chip.cts)
colnames(chip.cts) <- names(files)
rownames(chip.cts) <- rownames(dds)

head(chip.cts)
```

```
                Helas3.Rep1 Helas3.Rep2 Hepg2.Rep1 Hepg2.Rep2
ENSG00000000003         162         183          32          82
ENSG00000000419        1109        1148         600         774
ENSG00000000457        1504        1693         398         570
ENSG00000000460        2642        3216         462        1117
ENSG00000000971        1561        1894          14         188
ENSG00000001036         425         467         223         415
```

```
# Creating DESeq2 object
sampleTable <-
  data.frame(condition = factor(unlist(lapply(
    strsplit(colnames(chip.cts), "\\."),
    FUN = function(x) {
      x[1]
    }
  ))))
rownames(sampleTable) <- colnames(chip.cts)
sampleTable
```

```
            condition
Helas3.Rep1    Helas3
Helas3.Rep2    Helas3
Hepg2.Rep1      Hepg2
Hepg2.Rep2      Hepg2
```

```
chip.dds <-
  DESeqDataSetFromMatrix(chip.cts, sampleTable, design =  ~ condition)
head(assay(chip.dds))
```

```
                Helas3.Rep1 Helas3.Rep2 Hepg2.Rep1 Hepg2.Rep2
ENSG00000000003         162         183         32         82
ENSG00000000419        1109        1148        600        774
ENSG00000000457        1504        1693        398        570
ENSG00000000460        2642        3216        462       1117
ENSG00000000971        1561        1894         14        188
ENSG00000001036         425         467        223        415
```

At this point, `chip.dds` already has all necessary information we need. Now, note that there are a few genes without any mapped ChIP-seq counts.

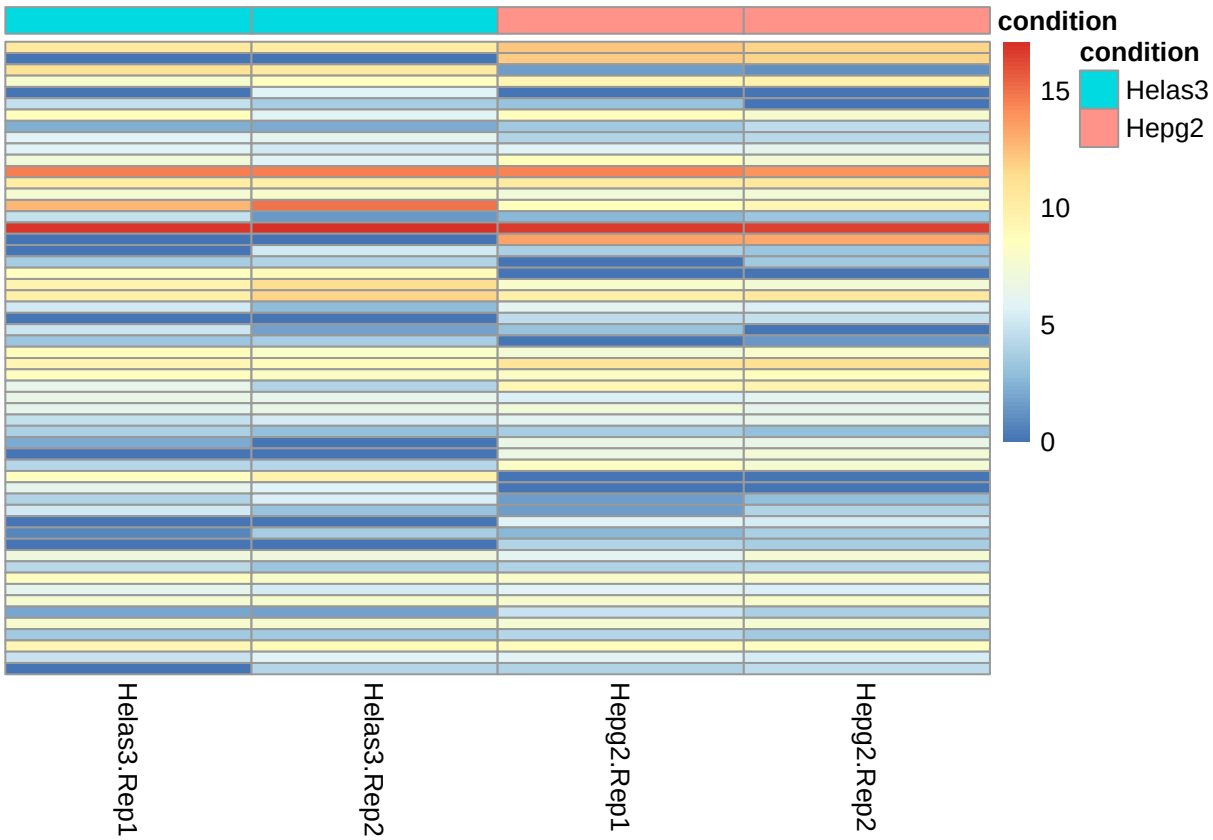
```
# Genes without H3K36me3 counts
nocts <- (rowSums(assay(chip.dds)) == 0)
```

```
# How many are there?
nrow(chip.dds[nocts,])
```

```
[1] 56
```

See below that the expression levels of these genes can be, sometimes, non negligible.

```
# Plotting
pheatmap(assay(ntd[row.names(chip.dds[nocts,]),]),cluster_rows=FALSE, show_rownames=FALSE,
         cluster_cols=FALSE, annotation_col=df)
```

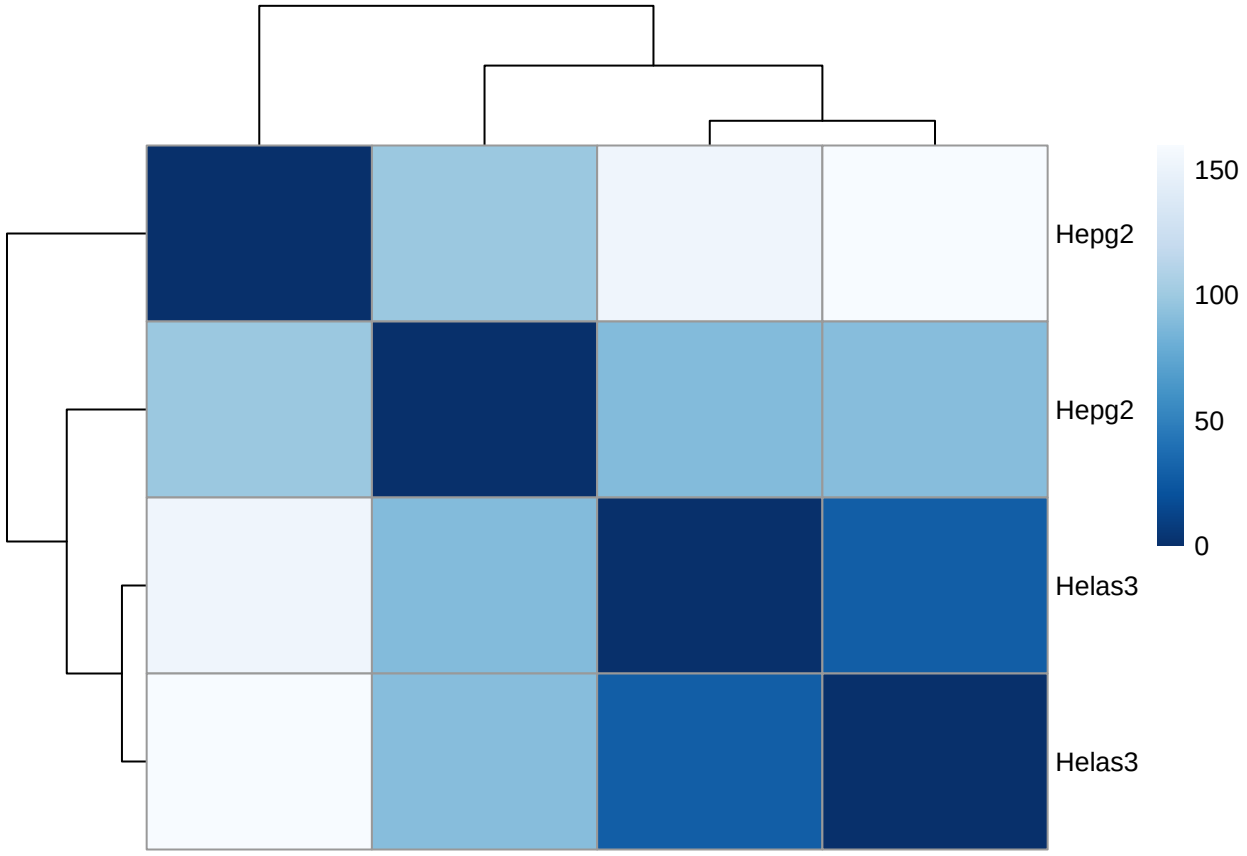


In any case, I will keep these genes in for now. When computing fold changes, they will not be considered, though.

2.2 Visualization

```
# Sample-to-sample distances
vsd <- vst(chip.dds, blind=FALSE)
sampleDists <- dist(t(assay(vsd)))
```

```
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsd$condition, sep="-")
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(RColorBrewer::brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows=sampleDists,
         clustering_distance_cols=sampleDists,
         col=colors)
```



At this point, we already start seeing some problems. Samples from Helas3 tend to be more similar between each other than with samples from Hepg2. However, note that one of the Hepg2 samples is as similar to its isogenic replicate as to replicates from Helas3. Maybe it is a poor quality ChIP-seq sample?

There is definitely a big difference in depth between cell lines, but I don't think it can justify alone the problematic case. The sample Hepg2.Rep2 also has very low depth but it is quite different from the Helas3 replicates from the distance heatmap above.

```
# Quick check of sequencing depth
colSums(assay(chip.dds))/1e6
```

```
Helas3.Rep1 Helas3.Rep2  Hepg2.Rep1  Hepg2.Rep2
  15.489462   16.261022    4.473797    7.438636
```

2.3 Differential enrichment

I am looking for a simple metric to compare DEG and differential enrichment for H3K36me3. I will use LFC to this end. If differential enrichment for H3K36me3 is necessary and sufficient to observe DGE, then the LFC of RNA-seq and ChIP-seq should be in agreement.

Now, I will run DESeq2 to get the LFC estimates from ChIP-seq data.

```
# Calling DESeq2
chip.dds <- DESeq(chip.dds)
```

```
estimating size factors
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

```
chip.res <- results(chip.dds)
chip.res
```

```
log2 fold change (MLE): condition Hepg2 vs Helas3
Wald test p-value: condition Hepg2 vs Helas3
DataFrame with 15476 rows and 6 columns
                 baseMean log2FoldChange     lfcSE      stat      pvalue
                <numeric>      <numeric> <numeric> <numeric>   <numeric>
ENSG00000000003   92.9396      -0.224842  0.481067 -0.467381 0.640227013
ENSG00000000419  895.4778       0.783871  0.371696  2.108903 0.034952931
ENSG00000000457  859.0835      -0.246441  0.355587 -0.693054 0.488275682
```

```
ENSG00000000460 1449.4861      -0.518785  0.374204 -1.386371 0.165633688
ENSG00000000971  568.7955      -2.938287  0.812002 -3.618573 0.000296232
...                   ...            ...       ...       ...         ...
ENSG00000273173   200.929       2.214234  0.411970   5.37475 7.66887e-08
ENSG00000273259   339.060       2.621708  0.365334   7.17620 7.16780e-13
ENSG00000273274    73.671      -1.323596  0.743680  -1.77979 7.51098e-02
ENSG00000273294  1299.351      -0.565076  0.320947  -1.76065 7.82970e-02
ENSG00000273439   364.216       1.095569  0.480925   2.27805 2.27238e-02
                      padj
                 <numeric>
ENSG00000000003  0.79396292
ENSG00000000419  0.12459652
ENSG00000000457  0.67872289
ENSG00000000460  0.34254535
ENSG00000000971  0.00348196
...                     ...
ENSG00000273173 3.29472e-06
ENSG00000273259 1.19841e-10
ENSG00000273274 2.06507e-01
ENSG00000273294 2.12585e-01
ENSG00000273439 9.30163e-02
```

```
# How many differentially enriched genes are there? Using a p-value cutoff of 0.01
## I am removing the 56 genes where there was no enrichment signal for H3K36me3
sum(chip.res$padj<cutoff,na.rm = TRUE)
```

```
[1] 1622
```

The analysis above is telling us that there are about 1500 genes with differential enrichment for H3K36me3. This number is about 5 times smaller than the number of DEG genes.

3 Results

3.1 Do log fold changes between ChIP-seq and RNA-seq agree?

I will compute shrunken LFC for better visualization and ranking of genes as detailed here.

```
shk.chip.res <- lfcShrink(chip.dds,type="apeglm",coef = 'condition_Hepg2_vs_Helas3')
```

```
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
```

```
shk.res <- lfcShrink(dds,type="apeglm",coef = 'condition_Hepg2_vs_Helas3')
```

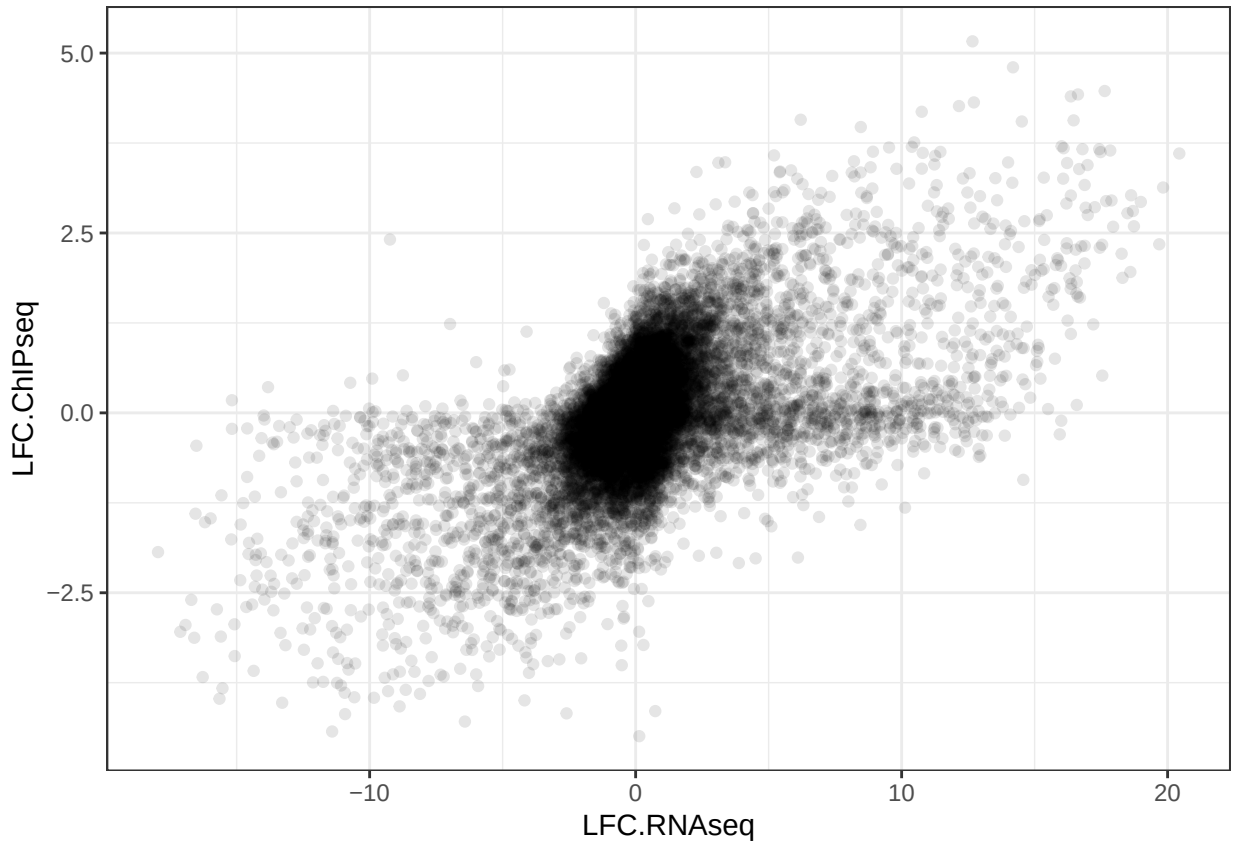
```
using 'apeglm' for LFC shrinkage. If used in published research, please cite:
    Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
    sequence count data: removing the noise and preserving large differences.
    Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895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

```
# First, make sure we are comparing the same genes
all.equal(rownames(shk.chip.res),rownames(shk.res))
```

```
[1] TRUE
```

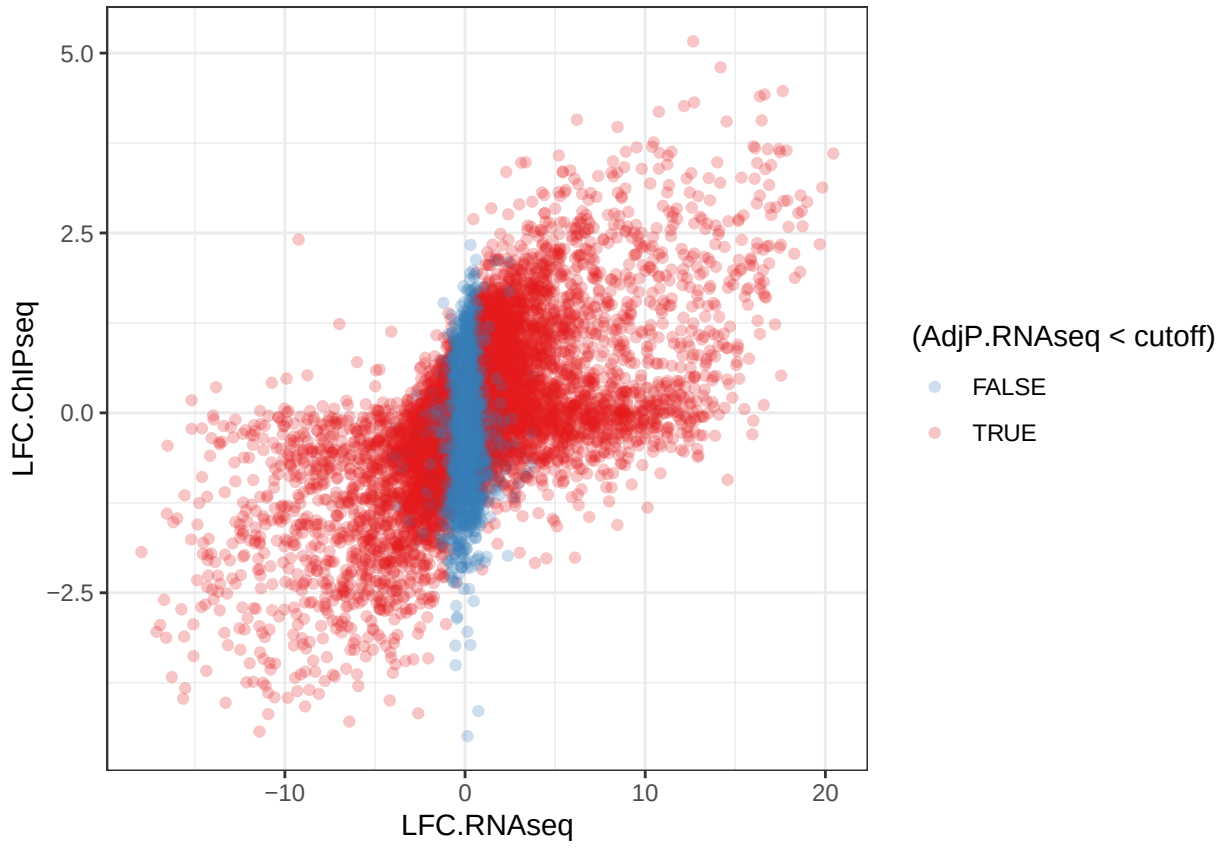
```
dt.lfc <- data.table(Gene = rownames(shk.res),
                     LFC.RNAseq = shk.res$log2FoldChange,
                     LFC.ChIPseq = shk.chip.res$log2FoldChange,
                     AdjP.RNAseq = shk.res$padj,
                     AdjP.ChIPseq = shk.chip.res$padj)


ggplot(dt.lfc) +
  geom_point(aes(x = LFC.RNAseq,y = LFC.ChIPseq),alpha = 0.1) +
  theme_bw()
```

```
Warning: Removed 56 rows containing missing values (geom_point).
```



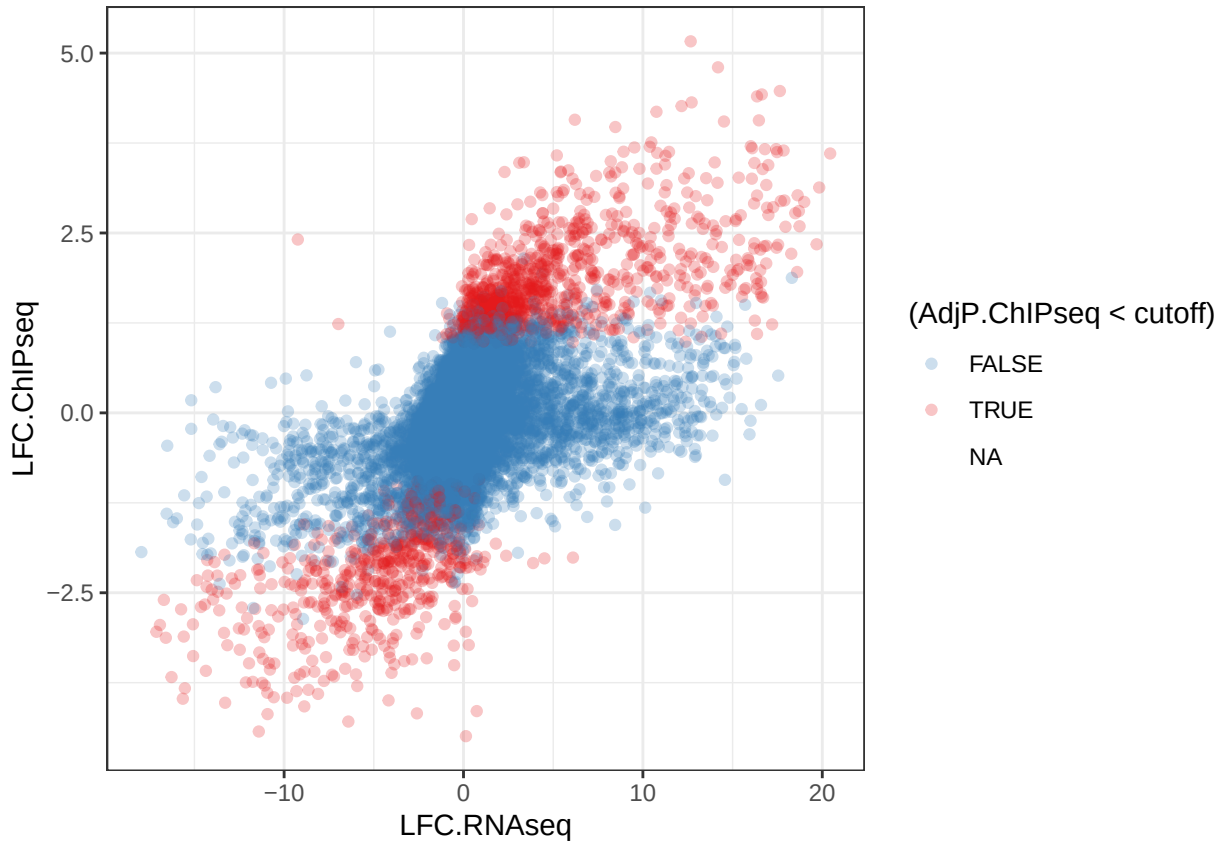

```
ggplot(dt.lfc) +
  geom_point(aes(x = LFC.RNAseq,y = LFC.ChIPseq,color = (AdjP.RNAseq<cutoff)),alpha = 0.25) +
  scale_color_brewer(palette = 'Set1',direction = -1)+
  theme_bw()
```

```
Warning: Removed 56 rows containing missing values (geom_point).
```



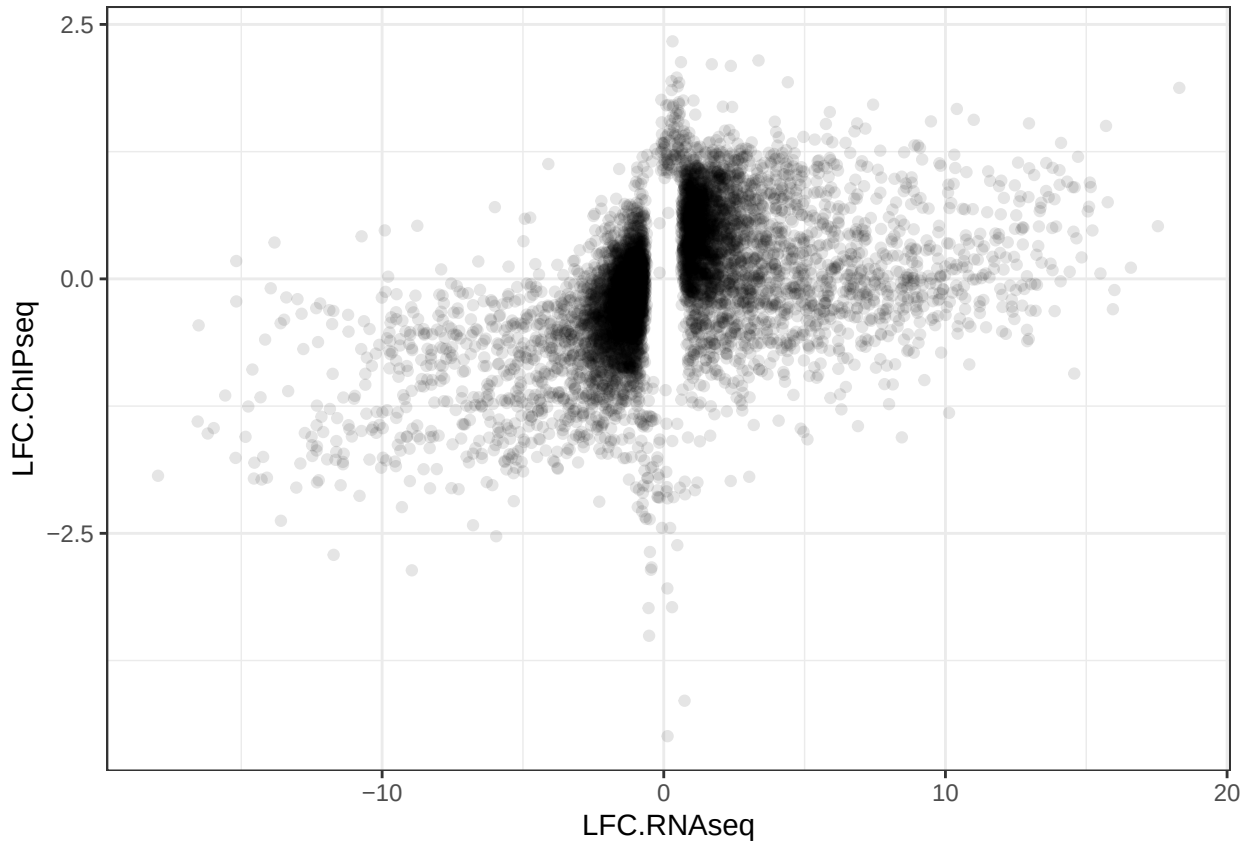

```
ggplot(dt.lfc) +
  geom_point(aes(x = LFC.RNAseq,y = LFC.ChIPseq,color = (AdjP.ChIPseq<cutoff)),alpha = 0.25) +
  scale_color_brewer(palette = 'Set1',direction = -1)+
  theme_bw()
```

```
Warning: Removed 654 rows containing missing values (geom_point).
```

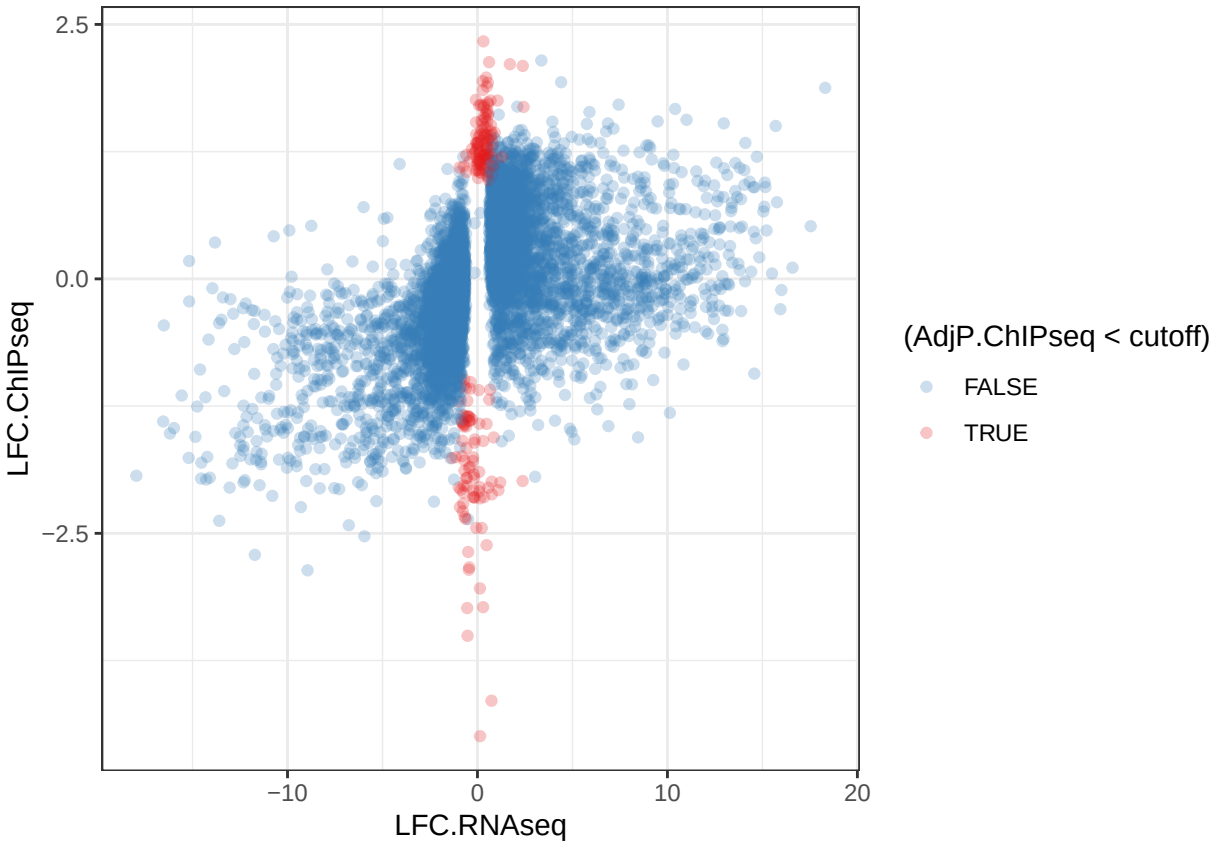


We can see that there is a positive association between LFC of RNA- and ChIP-seq counts. However, several genes exhibit no association between RNA-seq and ChIP-seq counts.

```
# Selecting problematic genes
dt.lfc.issues <- dt.lfc[((AdjP.RNAseq<cutoff) & (AdjP.ChIPseq>cutoff)) |
                          ((AdjP.RNAseq>cutoff) & (AdjP.ChIPseq<cutoff)),]


ggplot(dt.lfc.issues) +
  geom_point(aes(x = LFC.RNAseq,y = LFC.ChIPseq),alpha = 0.1) +
  theme_bw()
```




```
ggplot(dt.lfc.issues) +
  geom_point(aes(x = LFC.RNAseq,y = LFC.ChIPseq,color = (AdjP.ChIPseq<cutoff)),alpha = 0.25) +
  scale_color_brewer(palette = 'Set1',direction = -1)+
  theme_bw()
```



3.2 UCSC Genome Browser view os problematic cases

Below, I present a few snap shots from the UCSC Genome Browser of problematic cases from the `dt.lfc.issues` table above.

3.2.1 Example 1: Gene ENSG00000113389

Here we have an example of a DE gene without differential enrichment of H3K36me3. Note that, in Helas3, enrichment on the gene body is no different from enrichment on the intergenic (background) region. Therefore, observed differences in ChIP-seq are simply a sequencing depth artifact and would have to be accounted by the peak caller with proper normalization methods. If used as a benchmarking mark, differential peaks covering this genes would be incorrectly considered as a true positive calls.

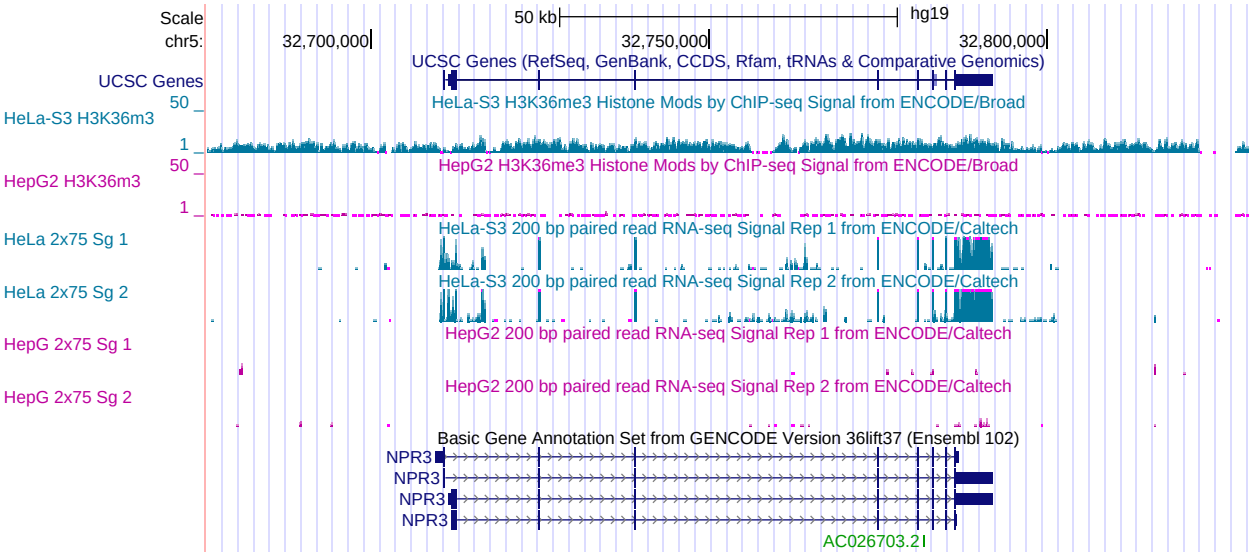


Figure 1: Example 1 (Gene ENSG00000113389)

3.2.2 Example 2 (Gene ENSG00000137285)

Another problematic example here. We have a gene that is expressed in both cell lines but the gene expression level is different.

There are two problems with the example. First, DEG level does not lead to differential ChIP-seq enrichment. The enrichment for H3K36me3 is similar in both cell lines. Second, it appears that the enrichment of H3K36me3 reads on the gene body is no different than intergenic (background) region.

If considered as a benchmarking mark, this gene would be considered as a true positive. One would expect differential peak callers to not call differential peaks on this region. If so, this gene would lead to an increased false negative rate in methods that correctly call this region as consensus background.

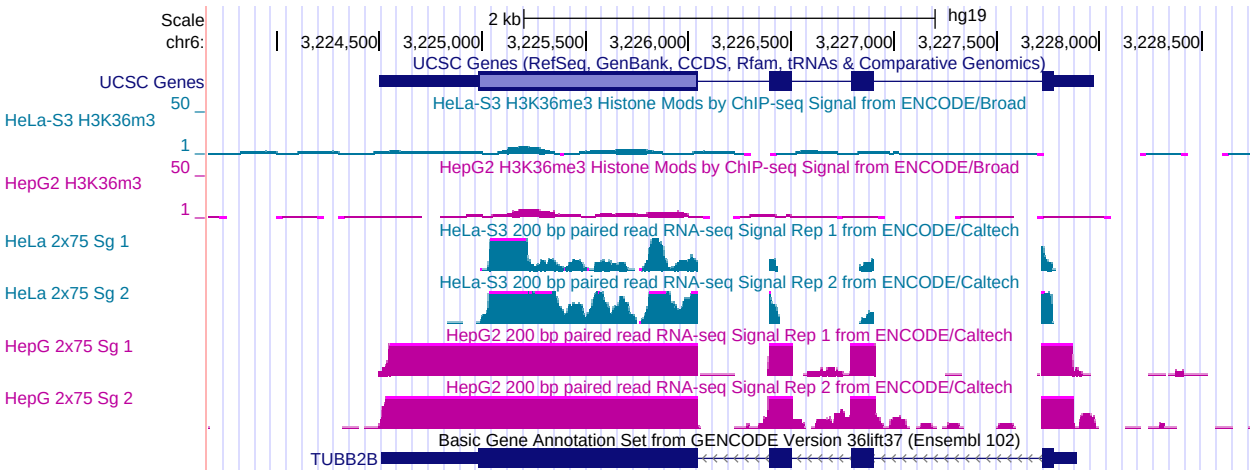


Figure 2: Example 2 (Gene ENSG00000137285)

3.2.3 Example 3 (Gene ENSG00000139998)

This is another similar example of DEG not leading to differential H3K36me3.

However, there is one additional point to make here. If the gene below is DEG, and one assumes that there is a local ChIP-seq signal difference in between cell lines on the exonic regions, how do we justify the similar ChIP-seq coverage on the remaining regions on the gene body (right most part of the figure) between the two cell lines. Should differential peaks covering those regions be true positive calls (because they cover a DE gene)? Or should they be false positive calls (because there is no difference in ChIP-seq signal)? I believe the latter is a more reasonable choice in this example.

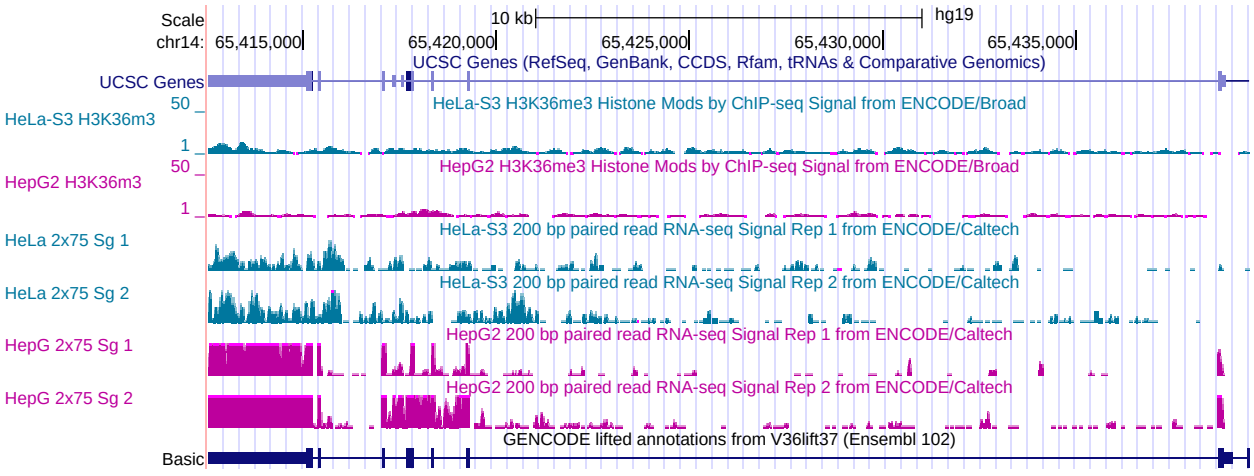


Figure 3: Example 3 (Gene ENSG00000139998)

3.2.4 Example 4 (Gene ENSG00000144852)

Another example of DE gene not implying differential enrichment of H3K36me3.

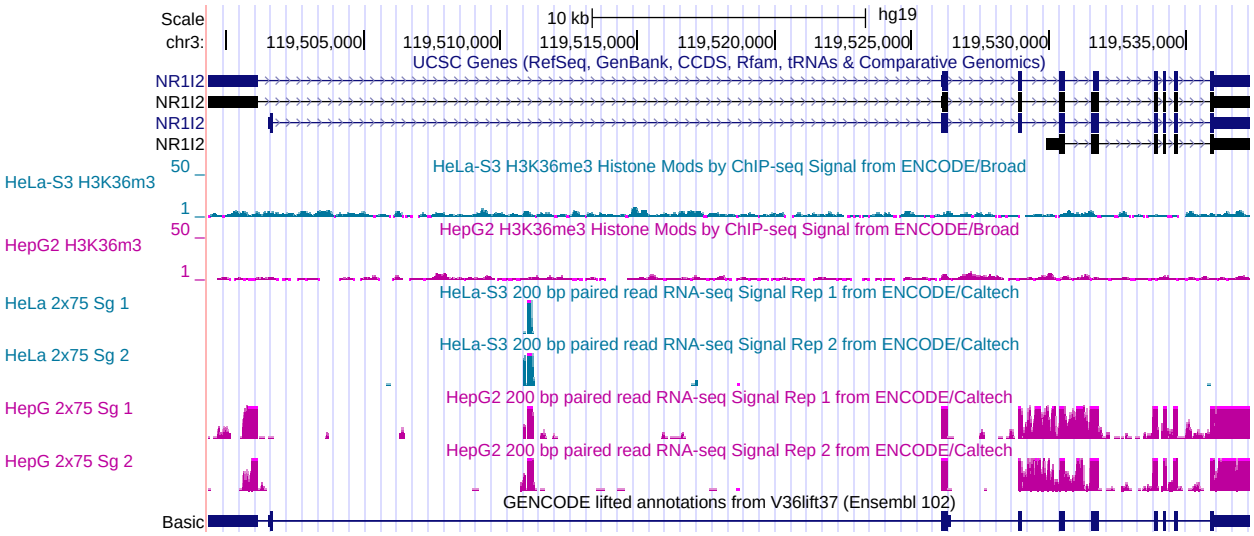


Figure 4: Example 4 (Gene ENSG00000144852)

3.2.5 Example 5 (Gene ENSG00000160202)

Another example of DE gene not implying differential enrichment of H3K36me3.

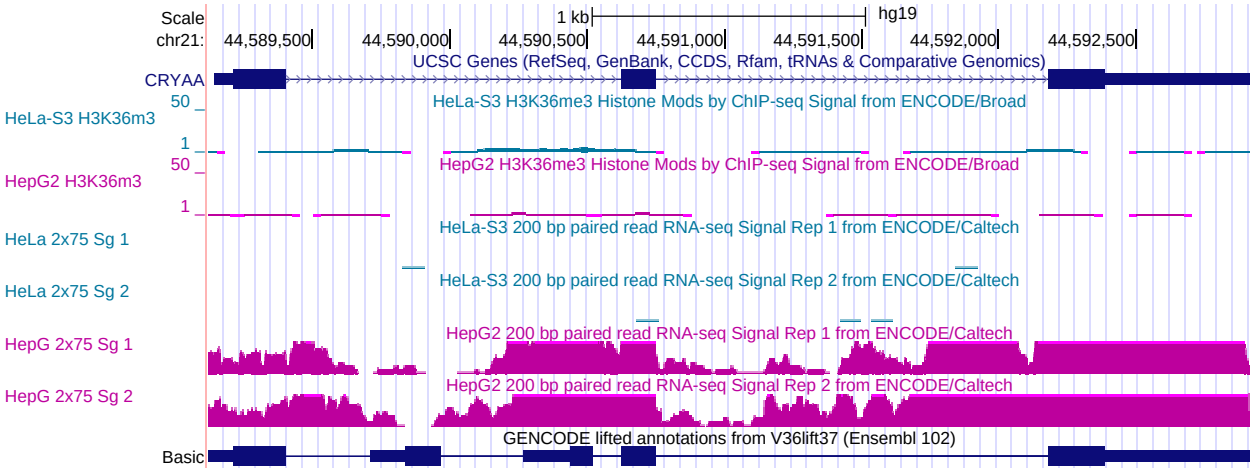


Figure 5: Example 5 (Gene ENSG00000160202)

3.2.6 Example 6 (Gene ENSG00000178498)

Another example of DE gene not implying differential enrichment of H3K36me3.

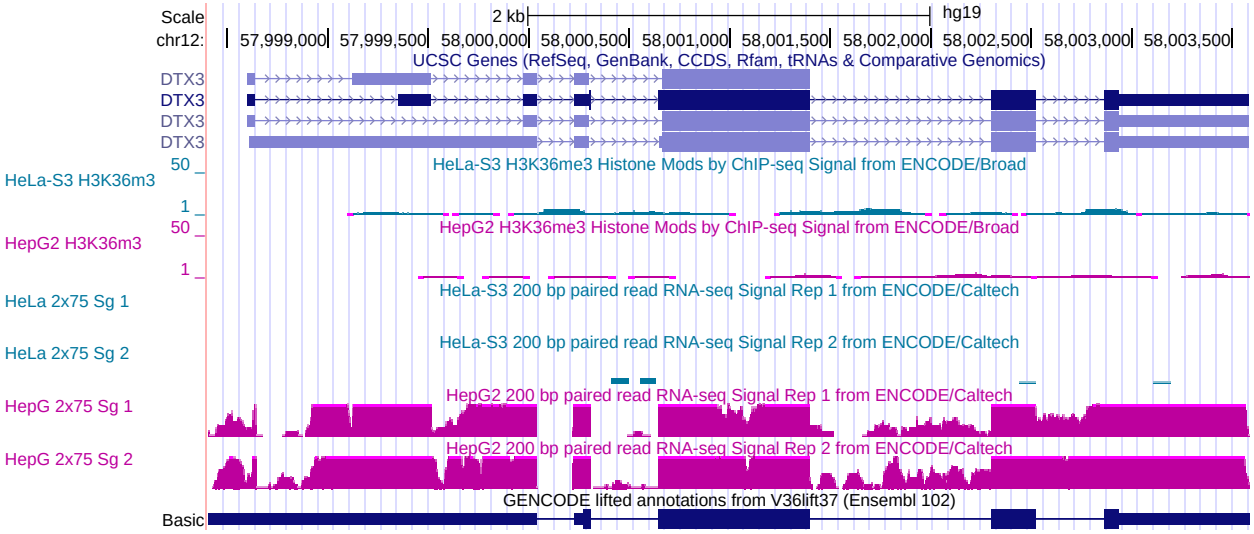


Figure 6: Example 6 (Gene ENSG00000178498)

3.2.7 Example 7 (Gene ENSG00000198610)

Another example of DE gene not implying differential enrichment of H3K36me3.

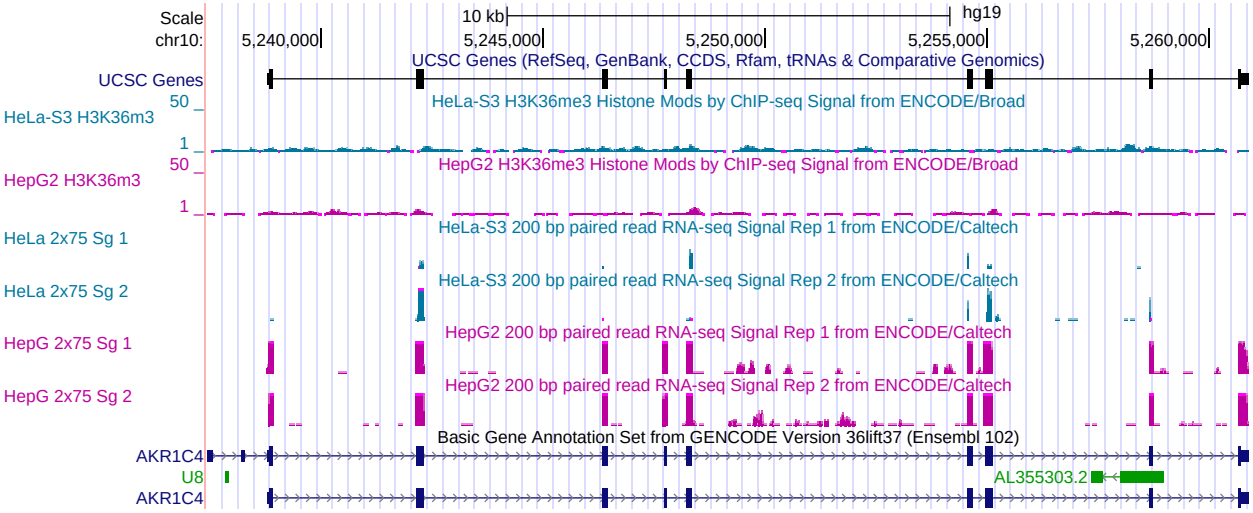


Figure 7: Example 7 (Gene ENSG00000198610)

4 Summary

Based on the results above, there seems to be a positive association between DGE and differential enrichment for H3K36me3 when comparing Helas3 and Hepg2 cell lines. However, it is straightforward to find examples where DGE does not lead to differential ChIP-seq enrichment. When comparing methods regarding sensitivity/specificity, including these problematic genes in the set of gold-standard marks would lead to an invalid benchmarking study. Therefore, utilizing DE genes as a metric to compare methods regarding the genome-wide sensitivity/specificity of differential peak calls is not ideal. To pursue with this approach, one would need to manually curate the gene list while utilizing ChIP-seq data to determine the gold-standard set of true positive and true negative genes.

As detailed in our paper, we defined true and positive genes as those with differential enrichment for H3K36me3 ChIP-seq read counts. While this approach is not ideal either, it does not introduce problematic examples as those shown above. If one is concerned about the cutoff choice to define the gold-standard set of genes based on ChIP-seq counts, we show in our Supplementary Materials that our method, epigraHMM, outperforms all current differential peak callers in terms of sensitivity/specificity regardless the chosen LFC cutoff.

Lastly, it is important to note here that, as H3K36me3 is a broad epigenomic mark, the calculation of sensitivity/specificity of differential peak callers should not be based on 'hits on genes'. In other words, a differential peak hitting a DE gene does not imply that the differential peak is properly covering the entire differential region of ChIP-seq read enrichment. While the 'hits on genes' could be a valid approach to compute sensitivity/specificity from *short* marks (which is, by the way, the strategy used by many other papers; see csaw's paper from Lun & Smyth (2015) for an example), it is not ideal for *broad* marks for which the enrichment of counts can be quite complex and expand through large genomic domains. Therefore, the calculation of sensitivity/specificity for marks such as H3K36me3 should be based on either the average proportion of DE gene coverage by differential peaks or on the standard window-based calculation of sensitivity/specificity. In our paper, we utilize the latter strategy.