Code to generate Figure 4 of 'Glucocorticoid receptor collaborates with pioneer factors and AP-1 to execute genome-wide regulation'

Erin M. Wissink

Loading packages and setting paths

```
library(ggplot2)
library(bigWig)
library(gridExtra)
library(bedtoolsr)


ssBigwigPath <- '../Manuscript_data/PRO-seq_bigwigs/Merged_subsampled/'
msDataPath <- '../Manuscript_data/'
hg38='../Manuscript_data/hg38.fa'

geneData <- read.table('gene_DESeq_analysis.txt', sep = '\t')
promoters <- read.table(paste0(msDataPath, 'hg38_refseq_promoters.bed')
                        , sep = '\t')

gg_options <- theme(axis.text.x = element_text(size = 8),
                    axis.text.y = element_text(size = 8),
                    axis.title.x = element_text(size = 8, colour='black'),
                    axis.title.y = element_text(size = 8, colour='black'),
                    legend.text=element_text(size=8, colour='black'),
                    legend.title = element_text(size = 8, colour='black'),
                    axis.line = element_line(colour = 'black', size = 0.5),
                    axis.ticks = element_line(colour = "black", size = 0.5)) +
                    theme_classic()

A549_genes_up <- rownames(geneData[geneData$A549_wt_100vs0dex_fdr < 0.05 &
                                     geneData$A549_wt_100vs0dex_log2 > 0,])
A549_genes_down <- rownames(geneData[geneData$A549_wt_100vs0dex_fdr < 0.05 &
                                       geneData$A549_wt_100vs0dex_log2 < 0,])
U2OS_genes_up <- rownames(geneData[geneData$U2OS_wt_100vs0dex_fdr < 0.05 &
                                     geneData$U2OS_wt_100vs0dex_log2 > 0,])
U2OS_genes_down <- rownames(geneData[geneData$U2OS_wt_100vs0dex_fdr < 0.05 &
                                       geneData$U2OS_wt_100vs0dex_log2 < 0,])
```

Finding motifs that are enriched in induced CCREs using HOMER, looking at the central 300 bp

Separating CCREs by induction or repression category

```
CCREs <- read.table('CCRE_DESeq_analysis.txt')[,c(14:17, 19:22)]

#for CCREs that were not tested because of low read counts, set padj to 1
CCREs$A549_wt_1vs0dex_fdr[is.na(CCREs$A549_wt_1vs0dex_fdr)] <- 1
CCREs$A549_wt_100vs0dex_fdr[is.na(CCREs$A549_wt_100vs0dex_fdr)] <- 1
CCREs$U2OS_wt_1vs0dex_fdr[is.na(CCREs$U2OS_wt_1vs0dex_fdr)] <- 1
CCREs$U2OS_wt_100vs0dex_fdr[is.na(CCREs$U2OS_wt_100vs0dex_fdr)] <- 1

#for CCREs that were not tested because of low read counts, set log2fc to 0
CCREs[is.na(CCREs)] <- 0
CCREs$category <- ''

for (i in 1:nrow(CCREs)){
  if (CCREs[i,4] < 0.05 & CCREs[i,8]<0.05){
    if (CCREs[i,3] > 0 & CCREs[i,7] > 0){
      CCREs[i,9] <- 'both_up'
    }
    else if (CCREs[i,3] < 0 & CCREs[i,7] < 0){
      CCREs[i,9] <- 'both_down'
    }
    else{
      CCREs[i,9] <- 'opposite'
    }
  }
  else if (CCREs[i,4] <0.05){
    if (CCREs[i,3] > 0) {
      CCREs[i,9] <- 'A549_up'
    }
    else {CCREs[i,9] <- 'A549_down'}
  }
  else if (CCREs[i,8] < 0.05){
    if (CCREs[i,7] > 0){
      CCREs[i,9] <- 'U2OS_up'
    }
    else{CCREs[i,9] <- 'U2OS_down'}
  }
  else{CCREs[i,9] <- 'bkgd'}
}

CCREs$cood <- rownames(CCREs)
CCREs$chr <- ''
CCREs$start <- ''
CCREs$end <- ''

for (i in 1:nrow(CCREs)){
  CCREs[i,11] <- substr(CCREs[i,10], 1,
                        which(strsplit(CCREs[i,10], "")[[1]]==":")-1)
  CCREs[i,12] <- substr(CCREs[i,10],
```

```
                        which(strsplit(CCREs[i,10], "")[[1]]==":")+1,
                        which(strsplit(CCREs[i,10], "")[[1]]=="-")-1)
  CCREs[i,13] <- substr(CCREs[i,10],
                        which(strsplit(CCREs[i,10], '')[[1]]=='-')+1,
                        stop = nchar(CCREs[i,10]))
}

CCREs$start300 <- (as.numeric(CCREs$start) + as.numeric(CCREs$end))/2 -150
CCREs$end300 <- CCREs$start300 + 300

both_up_bed <- CCREs[CCREs$category=='both_up', c(11, 14, 15)]
A549_up_all_bed <- CCREs[CCREs$category=='both_up' |
                           CCREs$category=='A549_up', c(11, 14, 15)]
U2OS_up_all_bed <- CCREs[CCREs$category=='both_up' |
                           CCREs$category=='U2OS_up', c(11, 14, 15)]
A549_up_only_bed <- CCREs[CCREs$category=='A549_up', c(11, 14, 15)]
U2OS_up_only_bed <- CCREs[CCREs$category=='U2OS_up', c(11, 14, 15)]


both_down_bed <- CCREs[CCREs$category=='both_down', c(11, 14, 15)]
A549_down_all_bed <- CCREs[CCREs$category=='both_down' |
                           CCREs$category=='A549_down', c(11, 14, 15)]
U2OS_down_all_bed <- CCREs[CCREs$category=='both_down' |
                           CCREs$category=='U2OS_down', c(11, 14, 15)]
A549_down_only_bed <- CCREs[CCREs$category=='A549_down', c(11, 14, 15)]
U2OS_down_only_bed <- CCREs[CCREs$category=='U2OS_down', c(11, 14, 15)]

bkgd_bed <- CCREs[CCREs$category=='bkgd', c(11, 14, 15)]
```

Creating fa files for HOMER

```
bedtoolsr::bt.getfasta(fi = '../Manuscript_data/hg38.fa',
                       bed = A549_up_all_bed, fo = 'A549_all_induced_CCREs_300bp.fa')
```

```
## data frame with 0 columns and 0 rows
```

```
bedtoolsr::bt.getfasta(fi = '../Manuscript_data/hg38.fa',
                       bed = A549_down_all_bed, fo = 'A549_all_repressed_CCREs_300bp.fa')
```

```
## data frame with 0 columns and 0 rows
```

```
bedtoolsr::bt.getfasta(fi = '../Manuscript_data/hg38.fa',
                       bed = U2OS_up_all_bed, fo = 'U2OS_all_induced_CCREs_300bp.fa')
```

```
## data frame with 0 columns and 0 rows
```

```
bedtoolsr::bt.getfasta(fi = '../Manuscript_data/hg38.fa',
                       bed = U2OS_down_all_bed, fo = 'U2OS_all_repressed_CCREs_300bp.fa')
```

```
## data frame with 0 columns and 0 rows
```

```
bedtoolsr::bt.getfasta(fi = '../Manuscript_data/hg38.fa',
                       bed = bkgd_bed, fo = 'bkgd_CCREs_300bp.fa')
```

```
## data frame with 0 columns and 0 rows
```

Running HOMER

export PATH=$PATH:/home/emw97/homer/bin

```
findMotifs.pl A549_all_induced_CCREs_300bp.fa \
fasta A549_induced_CCREs_vs_bkgd_CCREs/ \
-len 8,10,12 -fastaBg bkgd_CCREs_300bp.fa \
-mcheck HOCOMOCOv11_core_HUMAN_mono_homer_format_0.001.motif \
-bits -nogo  -p 8 \
-mknown HOCOMOCOv11_core_HUMAN_mono_homer_format_0.001.motif

findMotifs.pl U2OS_all_induced_CCREs_300bp.fa \
fasta U2OS_induced_CCREs_vs_bkgd_CCREs/ \
-len 8,10,12 -fastaBg bkgd_CCREs_300bp.fa \
-mcheck HOCOMOCOv11_core_HUMAN_mono_homer_format_0.001.motif \
-bits -nogo  -p 8 \
-mknown HOCOMOCOv11_core_HUMAN_mono_homer_format_0.001.motif


findMotifs.pl A549_all_repressed_CCREs_300bp.fa \
fasta A549_repressed_CCREs_vs_bkgd_CCREs/ \
-len 8,10,12 -fastaBg bkgd_CCREs_300bp.fa \
-mcheck HOCOMOCOv11_core_HUMAN_mono_homer_format_0.001.motif \
-bits -nogo -p 8 \
-mknown HOCOMOCOv11_core_HUMAN_mono_homer_format_0.001.motif

findMotifs.pl U2OS_all_repressed_CCREs_300bp.fa \
fasta U2OS_repressed_CCREs_vs_bkgd_CCREs/ \
-len 8,10,12 -fastaBg bkgd_CCREs_300bp.fa \
-mcheck HOCOMOCOv11_core_HUMAN_mono_homer_format_0.001.motif \
-bits -nogo -p 8 \
-mknown HOCOMOCOv11_core_HUMAN_mono_homer_format_0.001.motif
```

Finding nearest expressed gene to induced and repressed CCREs and finding their expression changes

In this next section, I will take CCREs that are induced or repressed, and I will find the nearest expressed gene and determine its expression change. The steps to do so are: 1. Identify the promoters for expressed genes 2. Determine which promoter is the most used isoform for each cell type in all conditions, using subsampled PRO-seq data 3. Find which promoter is closest to each CCRE 4. Compare gene expression with CCRE expression

```
colnames(promoters) <- c('chr', 'start', 'end', 'gene', '.', 'strand')
expPromoters <- promoters[promoters$gene %in% rownames(geneData),]

expPromoters$A549 <- ''
expPromoters$U2OS <- ''

A549_0_pbw <- load.bigWig(paste0(ssBigwigPath,
                                 'A549_WT_0dex_merged_subsampled_fwd.bw'))
A549_1_pbw <- load.bigWig(paste0(ssBigwigPath,
                                 'A549_WT_01dex_merged_subsampled_fwd.bw'))
A549_100_pbw <- load.bigWig(paste0(ssBigwigPath,
                                   'A549_WT_100dex_merged_subsampled_fwd.bw'))
A549_0_mbw <- load.bigWig(paste0(ssBigwigPath,
                                 'A549_WT_0dex_merged_subsampled_rev.bw'))
A549_1_mbw <- load.bigWig(paste0(ssBigwigPath,
                                 'A549_WT_01dex_merged_subsampled_rev.bw'))
A549_100_mbw <- load.bigWig(paste0(ssBigwigPath,
                                   'A549_WT_100dex_merged_subsampled_rev.bw'))
U2OS_0_pbw <- load.bigWig(paste0(ssBigwigPath,
                                 'U2OS_WT_0dex_merged_subsampled_fwd.bw'))
U2OS_1_pbw <- load.bigWig(paste0(ssBigwigPath,
                                 'U2OS_WT_01dex_merged_subsampled_fwd.bw'))
U2OS_100_pbw <- load.bigWig(paste0(ssBigwigPath,
                                   'U2OS_WT_100dex_merged_subsampled_fwd.bw'))
U2OS_0_mbw <- load.bigWig(paste0(ssBigwigPath,
                                 'U2OS_WT_0dex_merged_subsampled_rev.bw'))
U2OS_1_mbw <- load.bigWig(paste0(ssBigwigPath,
                                 'U2OS_WT_01dex_merged_subsampled_rev.bw'))
U2OS_100_mbw <- load.bigWig(paste0(ssBigwigPath,
                                   'U2OS_WT_100dex_merged_subsampled_rev.bw'))

expPromoters$A549 <-
  bed6.region.bpQuery.bigWig(bw.plus = A549_0_pbw,
                             bw.minus = A549_0_mbw,
                             bed6 = expPromoters[1:6],
                             op = 'sum', abs.value = T) +
  bed6.region.bpQuery.bigWig(bw.plus = A549_1_pbw,
                             bw.minus = A549_1_mbw,
                             bed6 = expPromoters[1:6],
                             op = 'sum', abs.value = T) +
  bed6.region.bpQuery.bigWig(bw.plus = A549_100_pbw,
                             bw.minus = A549_100_mbw,
                             bed6 = expPromoters[1:6],
                             op = 'sum', abs.value = T)

expPromoters$U2OS <-
  bed6.region.bpQuery.bigWig(bw.plus = U2OS_0_pbw,
                             bw.minus = U2OS_0_mbw,
                             bed6 = expPromoters[1:6],
                             op = 'sum', abs.value = T) +
  bed6.region.bpQuery.bigWig(bw.plus = U2OS_1_pbw,
                             bw.minus = U2OS_1_mbw,
                             bed6 = expPromoters[1:6],
```

```
                              op = 'sum', abs.value = T) +
  bed6.region.bpQuery.bigWig(bw.plus = U2OS_100_pbw,
                              bw.minus = U2OS_100_mbw,
                              bed6 = expPromoters[1:6],
                              op = 'sum', abs.value = T)

write.table(expPromoters, 'Read_counts_for_promoters_of_exp_genes.txt',
            sep = '\t', quote = F, row.names = F, col.names = F)
```

For genes with multiple promoters, determining which has the highest read count.

```
dataIn = open('Read_counts_for_promoters_of_exp_genes.txt', 'r')
A549out = open('Promoter_highest_read_count_A549.bed', 'w')
U2OSout = open('Promoter_highest_read_count_U2OS.bed', 'w')

A549 = {}
U2OS = {}

while 1:
  curr=dataIn.readline()
  if not curr:
    break
  curr=curr.rstrip().split('\t')
  if curr[3] not in A549.keys():
    A549[curr[3]] = [curr[0:6], int(curr[6])]
  else:
    if int(curr[6]) > A549[curr[3]][1]:
      A549[curr[3]] = [curr[0:6], int(curr[6])]
  if curr[3] not in U2OS.keys():
    U2OS[curr[3]] = [curr[0:6], int(curr[7])]
  else:
    if int(curr[7]) > U2OS[curr[3]][1]:
      U2OS[curr[3]] = [curr[0:6], int(curr[7])]
dataIn.close()

for i in A549.keys():
    A549out.write('\t'.join(A549[i][0]) + '\n')
for i in U2OS.keys():
    U2OSout.write('\t'.join(U2OS[i][0]) + '\n')
A549out.close()
U2OSout.close()
```

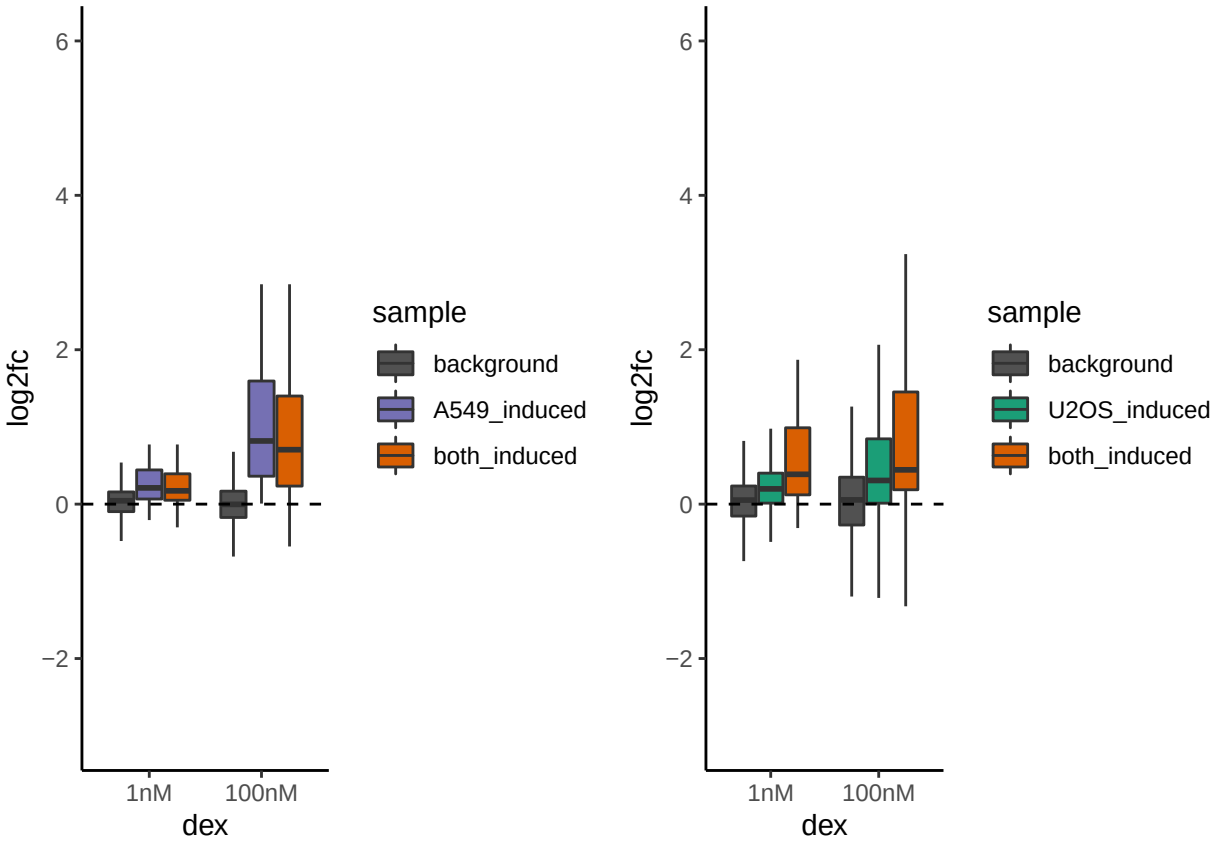
Filtering out CCREs that are within genes, identifying which promoter is closest to each induced CCRE and background, and plotting expression difference.

```
promA549 <-
  bedtoolsr::bt.sort(i = read.table('Promoter_highest_read_count_A549.bed', sep = '\t'))
promU2OS <-
  bedtoolsr::bt.sort(i = read.table('Promoter_highest_read_count_U2OS.bed', sep = '\t'))

#sorting out intragenic CCREs
A549_both_up <- bedtoolsr::bt.intersect(
 v = T, a = both_up_bed, b = paste0(msDataPath, 'hg38_refseq.bed'))
```

```
A549_up <- bedtoolsr::bt.intersect(
  v = T, a = A549_up_only_bed, b = paste0(msDataPath, 'hg38_refseq.bed'))
A549_both_down <- bedtoolsr::bt.intersect(
 v = T, a = both_down_bed, b = paste0(msDataPath, 'hg38_refseq.bed'))
A549_down <- bedtoolsr::bt.intersect(
  v = T, a = A549_down_only_bed, b = paste0(msDataPath, 'hg38_refseq.bed'))
A549_bkgd <- bedtoolsr::bt.intersect(
  v = T, a = bkgd_bed, b = paste0(msDataPath, 'hg38_refseq.bed'))


U2OS_both_up <- bedtoolsr::bt.intersect(
 v = T, a = both_up_bed, b = paste0(msDataPath, 'hg38_refseq.bed'))
U2OS_up <- bedtoolsr::bt.intersect(
  v = T, a = U2OS_up_only_bed, b = paste0(msDataPath, 'hg38_refseq.bed'))
U2OS_both_down <- bedtoolsr::bt.intersect(
 v = T, a = both_down_bed, b = paste0(msDataPath, 'hg38_refseq.bed'))
U2OS_down <- bedtoolsr::bt.intersect(
  v = T, a = U2OS_down_only_bed, b = paste0(msDataPath, 'hg38_refseq.bed'))
U2OS_bkgd <- bedtoolsr::bt.intersect(
  v = T, a = bkgd_bed, b = paste0(msDataPath, 'hg38_refseq.bed'))

# Finding closest promoter and creating unique list
A549_both_up <- unique(as.character(bedtoolsr::bt.closest(
 a = A549_both_up, b = promA549)[,7] ))
A549_up <- unique(as.character(bedtoolsr::bt.closest(
  a = A549_up, b = promA549)[,7] ))
A549_both_down <- unique(as.character(bedtoolsr::bt.closest(
 a = A549_both_down, b = promA549)  [,7] ))
A549_down <- unique(as.character(bedtoolsr::bt.closest(
  a = A549_down, b = promA549)[,7] ))
A549_bkgd <- unique(as.character(bedtoolsr::bt.closest(
  a = A549_bkgd, b = promA549)[,7] ))

U2OS_both_up <- unique(as.character(bedtoolsr::bt.closest(
 a = U2OS_both_up, b = promU2OS)  [,7] ))
U2OS_up <- unique(as.character(bedtoolsr::bt.closest(
  a = U2OS_up, b = promU2OS)[,7] ))
U2OS_both_down <- unique(as.character(bedtoolsr::bt.closest(
 a = U2OS_both_down, b = promU2OS)  [,7] ))
U2OS_down <- unique(as.character(bedtoolsr::bt.closest(
  a = U2OS_down, b = promU2OS)[,7] ))
U2OS_bkgd <- unique(as.character(bedtoolsr::bt.closest(
  a = U2OS_bkgd, b = promU2OS)[,7] ))

#Plotting and finding statistical significance for
#gene expression of genes near induced enhancers

both_up_A549_closest <- geneData[A549_both_up,c(14,16)]
A549_up_closest <- geneData[A549_up,c(14,16)]
both_down_A549_closest <- geneData[A549_both_down,c(14,16)]
A549_down_closest <- geneData[A549_down,c(14,16)]
A549_bkgd_closest <- geneData[A549_bkgd,c(14,16)]
both_up_U2OS_closest <- geneData[U2OS_both_up,c(18,20)]
```

```
U2OS_up_closest <- geneData[U2OS_up,c(18,20)]
both_down_U2OS_closest <- geneData[U2OS_both_down,c(18,20)]
U2OS_down_closest <- geneData[U2OS_down,c(18,20)]
U2OS_bkgd_closest <- geneData[U2OS_bkgd,c(18,20)]

# Plotting induced CCREs
A549_to_plot <- as.data.frame(
  matrix(nrow = 2*length(A549_up) + 2*length(A549_bkgd) + 2*length(A549_both_up),
         ncol = 3))
colnames(A549_to_plot) <- c('dex', 'sample', 'log2fc')
A549_to_plot$dex <- factor(
  c(rep('1nM', length(A549_up) + length(A549_bkgd) + length(A549_both_up)),
    rep('100nM', length(A549_up) + length(A549_bkgd) + length(A549_both_up))),
  levels = c('1nM', '100nM'))
A549_to_plot$sample <- factor(
  rep(c(rep('both_induced', length(A549_both_up)),
        rep('A549_induced', length(A549_up)),
        rep('background', length(A549_bkgd))),2),
  levels = c('background', 'A549_induced', 'both_induced'))
A549_to_plot$log2fc <- c(
  both_up_A549_closest$A549_wt_1vs0dex_log2,
  A549_up_closest$A549_wt_1vs0dex_log2,
  A549_bkgd_closest$A549_wt_1vs0dex_log2,
  both_up_A549_closest$A549_wt_100vs0dex_log2,
  A549_up_closest$A549_wt_100vs0dex_log2,
  A549_bkgd_closest$A549_wt_100vs0dex_log2)

U2OS_to_plot <- as.data.frame(matrix(
  nrow = 2*length(U2OS_up) + 2*length(U2OS_bkgd) + 2*length(U2OS_both_up),
  ncol = 3))
colnames(U2OS_to_plot) <- c('dex', 'sample', 'log2fc')
U2OS_to_plot$dex <- factor(
  c(rep('1nM', length(U2OS_up) + length(U2OS_bkgd) + length(U2OS_both_up)),
    rep('100nM', length(U2OS_up) + length(U2OS_bkgd) + length(U2OS_both_up))),
  levels = c('1nM', '100nM'))
U2OS_to_plot$sample <- factor(
  rep(c(rep('both_induced', length(U2OS_both_up)),
        rep('U2OS_induced', length(U2OS_up)),
        rep('background', length(U2OS_bkgd))),2),
  levels = c('background', 'U2OS_induced', 'both_induced'))
U2OS_to_plot$log2fc <- c(
  both_up_U2OS_closest$U2OS_wt_1vs0dex_log2,
  U2OS_up_closest$U2OS_wt_1vs0dex_log2,
  U2OS_bkgd_closest$U2OS_wt_1vs0dex_log2,
  both_up_U2OS_closest$U2OS_wt_100vs0dex_log2,
  U2OS_up_closest$U2OS_wt_100vs0dex_log2,
  U2OS_bkgd_closest$U2OS_wt_100vs0dex_log2)


p1 <- ggplot(
  data=subset(A549_to_plot, !is.na(log2fc)),
  aes(x=dex, y=log2fc, fill=sample)) +
  geom_boxplot(outlier.shape = NA) + gg_options +
```

```
  geom_hline(yintercept = 0, linetype='dashed') +
  scale_fill_manual(values = c('#515151', '#7570b3', '#d95f02')) +
  ylim(-3,6)
p2 <- ggplot(
  data=subset(U2OS_to_plot, !is.na(log2fc)),
  aes(x=dex, y=log2fc, fill=sample)) +
  geom_boxplot(outlier.shape = NA) + gg_options +
  geom_hline(yintercept = 0, linetype='dashed') +
  scale_fill_manual(values = c('#515151', '#1b9e77', '#d95f02'))+
  ylim(-3,6)
p3 <- grid.arrange(p1, p2, nrow = 1)
```




```
# Stats for induced CCREs
wilcox.test(
  A549_to_plot[A549_to_plot$dex=='1nM' & A549_to_plot$sample == 'background', 3],
  A549_to_plot[A549_to_plot$dex=='1nM' & A549_to_plot$sample=='both_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_to_plot[A549_to_plot$dex == "1nM" & A549_to_plot$sample ==  and A549_to_plot[A549_to_plo
## W = 52238, p-value = 1.802e-06
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(
  A549_to_plot[A549_to_plot$dex=='1nM' & A549_to_plot$sample == 'background', 3],
  A549_to_plot[A549_to_plot$dex=='1nM' & A549_to_plot$sample=='A549_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_to_plot[A549_to_plot$dex == "1nM" & A549_to_plot$sample ==  and A549_to_plot[A549_to_plo
## W = 33404, p-value = 1.177e-05
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(
  A549_to_plot[A549_to_plot$dex=='100nM' & A549_to_plot$sample == 'background', 3],
  A549_to_plot[A549_to_plot$dex=='100nM' & A549_to_plot$sample=='both_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_to_plot[A549_to_plot$dex == "100nM" & A549_to_plot$sample ==  and A549_to_plot[A549_to_pl
## W = 28579, p-value = 1.141e-15
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(
  A549_to_plot[A549_to_plot$dex=='100nM' & A549_to_plot$sample == 'background', 3],
  A549_to_plot[A549_to_plot$dex=='100nM' & A549_to_plot$sample=='A549_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_to_plot[A549_to_plot$dex == "100nM" & A549_to_plot$sample ==  and A549_to_plot[A549_to_pl
## W = 11687, p-value = 1.566e-15
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(
  U2OS_to_plot[U2OS_to_plot$dex=='1nM' & U2OS_to_plot$sample == 'background', 3],
  U2OS_to_plot[U2OS_to_plot$dex=='1nM' & U2OS_to_plot$sample=='both_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_to_plot[U2OS_to_plot$dex == "1nM" & U2OS_to_plot$sample ==  and U2OS_to_plot[U2OS_to_plo
## W = 42161, p-value = 7.765e-10
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(
  U2OS_to_plot[U2OS_to_plot$dex=='1nM' & U2OS_to_plot$sample == 'background', 3],
  U2OS_to_plot[U2OS_to_plot$dex=='1nM' & U2OS_to_plot$sample=='U2OS_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
```

```
##
## data:  U2OS_to_plot[U2OS_to_plot$dex == "1nM" & U2OS_to_plot$sample ==  and U2OS_to_plot[U2OS_to_plo
## W = 504053, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  U2OS_to_plot[U2OS_to_plot$dex=='100nM' & U2OS_to_plot$sample == 'background', 3],
  U2OS_to_plot[U2OS_to_plot$dex=='100nM' & U2OS_to_plot$sample=='both_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_to_plot[U2OS_to_plot$dex == "100nM" & U2OS_to_plot$sample ==  and U2OS_to_plot[U2OS_to_p
## W = 47137, p-value = 4.535e-08
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  U2OS_to_plot[U2OS_to_plot$dex=='100nM' & U2OS_to_plot$sample == 'background', 3],
  U2OS_to_plot[U2OS_to_plot$dex=='100nM' & U2OS_to_plot$sample=='U2OS_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_to_plot[U2OS_to_plot$dex == "100nM" & U2OS_to_plot$sample ==  and U2OS_to_plot[U2OS_to_p
## W = 483641, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```
table(rownames(A549_up_closest) %in% A549_genes_up)
```

```
##
## FALSE  TRUE
##    14    19
```

```
table(rownames(A549_bkgd_closest) %in% A549_genes_up)
```

```
##
## FALSE  TRUE
##  3389   242
```

```
table(rownames(both_up_A549_closest) %in% A549_genes_up)
```

```
##
## FALSE  TRUE
##    22    26
```

```
table(rownames(U2OS_up_closest) %in% U2OS_genes_up)
```

```
##
## FALSE  TRUE
##   256   129
```

```r
table(rownames(U2OS_bkgd_closest) %in% U2OS_genes_up)
```

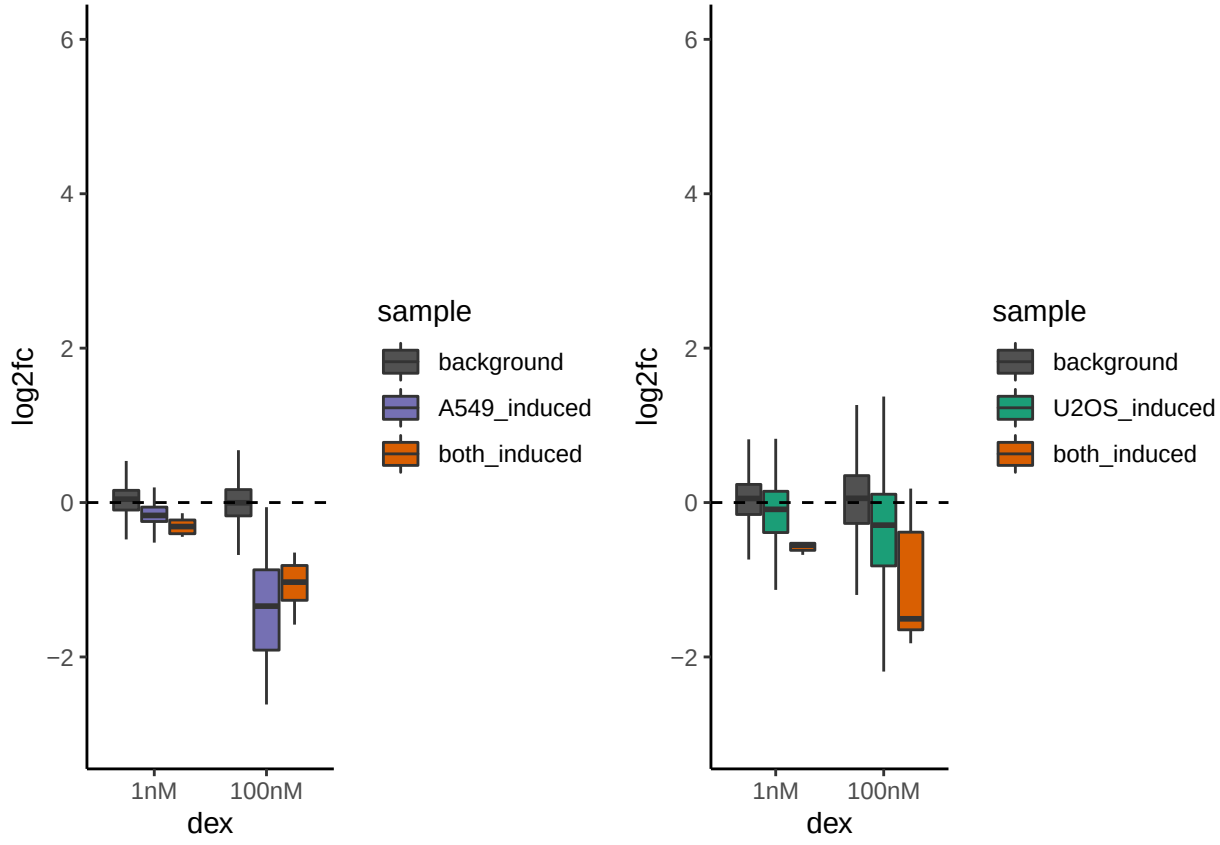
```
##
## FALSE  TRUE
##  3088   541
```

```r
table(rownames(both_up_U2OS_closest) %in% U2OS_genes_up)
```

```
##
## FALSE  TRUE
##    26    22
```

```r
# Plotting repressed CCREs
A549_to_plot <- as.data.frame(
  matrix(nrow = 2*length(A549_down) + 2*length(A549_bkgd) + 2*length(A549_both_down),
         ncol = 3))
colnames(A549_to_plot) <- c('dex', 'sample', 'log2fc')
A549_to_plot$dex <- factor(
  c(rep('1nM', length(A549_down) + length(A549_bkgd) + length(A549_both_down)),
    rep('100nM', length(A549_down) + length(A549_bkgd) + length(A549_both_down))),
  levels = c('1nM', '100nM'))
A549_to_plot$sample <- factor(
  rep(c(rep('both_induced', length(A549_both_down)),
        rep('A549_induced', length(A549_down)),
        rep('background', length(A549_bkgd))),2),
  levels = c('background', 'A549_induced', 'both_induced'))
A549_to_plot$log2fc <- c(
  both_down_A549_closest$A549_wt_1vs0dex_log2,
  A549_down_closest$A549_wt_1vs0dex_log2,
  A549_bkgd_closest$A549_wt_1vs0dex_log2,
  both_down_A549_closest$A549_wt_100vs0dex_log2,
  A549_down_closest$A549_wt_100vs0dex_log2,
  A549_bkgd_closest$A549_wt_100vs0dex_log2)

U2OS_to_plot <- as.data.frame(matrix(
  nrow = 2*length(U2OS_down) + 2*length(U2OS_bkgd) + 2*length(U2OS_both_down),
  ncol = 3))
colnames(U2OS_to_plot) <- c('dex', 'sample', 'log2fc')
U2OS_to_plot$dex <- factor(
  c(rep('1nM', length(U2OS_down) + length(U2OS_bkgd) + length(U2OS_both_down)),
    rep('100nM', length(U2OS_down) + length(U2OS_bkgd) + length(U2OS_both_down))),
  levels = c('1nM', '100nM'))
U2OS_to_plot$sample <- factor(
  rep(c(rep('both_induced', length(U2OS_both_down)),
        rep('U2OS_induced', length(U2OS_down)),
        rep('background', length(U2OS_bkgd))),2),
  levels = c('background', 'U2OS_induced', 'both_induced'))
U2OS_to_plot$log2fc <- c(
  both_down_U2OS_closest$U2OS_wt_1vs0dex_log2,
  U2OS_down_closest$U2OS_wt_1vs0dex_log2,
  U2OS_bkgd_closest$U2OS_wt_1vs0dex_log2,
  both_down_U2OS_closest$U2OS_wt_100vs0dex_log2,
```

```
  U2OS_down_closest$U2OS_wt_100vs0dex_log2,
  U2OS_bkgd_closest$U2OS_wt_100vs0dex_log2)


p4 <- ggplot(
  data=subset(A549_to_plot, !is.na(log2fc)),
  aes(x=dex, y=log2fc, fill=sample)) +
  geom_boxplot(outlier.shape = NA) + gg_options +
  geom_hline(yintercept = 0, linetype='dashed') +
  scale_fill_manual(values = c('#515151', '#7570b3', '#d95f02')) +
  ylim(-3,6)
p5 <- ggplot(data=subset(U2OS_to_plot, !is.na(log2fc)),
  aes(x=dex, y=log2fc, fill=sample)) +
  geom_boxplot(outlier.shape = NA) + gg_options +
  geom_hline(yintercept = 0, linetype='dashed') +
  scale_fill_manual(values = c('#515151', '#1b9e77', '#d95f02'))+
  ylim(-3,6)
p6 <- grid.arrange(p4, p5, nrow = 1)
```




```
# Stats for induced CCREs
wilcox.test(
  A549_to_plot[A549_to_plot$dex=='1nM' & A549_to_plot$sample == 'background', 3],
  A549_to_plot[A549_to_plot$dex=='1nM' & A549_to_plot$sample=='both_induced',3])
```

```
##
```

```
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_to_plot[A549_to_plot$dex == "1nM" & A549_to_plot$sample ==  and A549_to_plot[A549_to_plo
## W = 19982, p-value = 0.0004056
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  A549_to_plot[A549_to_plot$dex=='1nM' & A549_to_plot$sample == 'background', 3],
  A549_to_plot[A549_to_plot$dex=='1nM' & A549_to_plot$sample=='A549_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_to_plot[A549_to_plot$dex == "1nM" & A549_to_plot$sample ==  and A549_to_plot[A549_to_plo
## W = 27604, p-value = 0.004426
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  A549_to_plot[A549_to_plot$dex=='100nM' & A549_to_plot$sample == 'background', 3],
  A549_to_plot[A549_to_plot$dex=='100nM' & A549_to_plot$sample=='both_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_to_plot[A549_to_plot$dex == "100nM" & A549_to_plot$sample ==  and A549_to_plot[A549_to_pl
## W = 21319, p-value = 4.978e-05
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  A549_to_plot[A549_to_plot$dex=='100nM' & A549_to_plot$sample == 'background', 3],
  A549_to_plot[A549_to_plot$dex=='100nM' & A549_to_plot$sample=='A549_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_to_plot[A549_to_plot$dex == "100nM" & A549_to_plot$sample ==  and A549_to_plot[A549_to_pl
## W = 34300, p-value = 1.155e-06
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(U2OS_to_plot[U2OS_to_plot$dex=='1nM' & U2OS_to_plot$sample == 'background', 3],
            U2OS_to_plot[U2OS_to_plot$dex=='1nM' & U2OS_to_plot$sample=='both_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_to_plot[U2OS_to_plot$dex == "1nM" & U2OS_to_plot$sample ==  and U2OS_to_plot[U2OS_to_plo
## W = 19605, p-value = 0.000689
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(U2OS_to_plot[U2OS_to_plot$dex=='1nM' & U2OS_to_plot$sample == 'background', 3],
            U2OS_to_plot[U2OS_to_plot$dex=='1nM' & U2OS_to_plot$sample=='U2OS_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_to_plot[U2OS_to_plot$dex == "1nM" & U2OS_to_plot$sample ==  and U2OS_to_plot[U2OS_to_plo
## W = 488784, p-value = 9.936e-10
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  U2OS_to_plot[U2OS_to_plot$dex=='100nM' & U2OS_to_plot$sample == 'background', 3],
  U2OS_to_plot[U2OS_to_plot$dex=='100nM' & U2OS_to_plot$sample=='both_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_to_plot[U2OS_to_plot$dex == "100nM" & U2OS_to_plot$sample ==  and U2OS_to_plot[U2OS_to_p
## W = 17843, p-value = 0.00677
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  U2OS_to_plot[U2OS_to_plot$dex=='100nM' & U2OS_to_plot$sample == 'background', 3],
  U2OS_to_plot[U2OS_to_plot$dex=='100nM' & U2OS_to_plot$sample=='U2OS_induced',3])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_to_plot[U2OS_to_plot$dex == "100nM" & U2OS_to_plot$sample ==  and U2OS_to_plot[U2OS_to_p
## W = 524104, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```
table(rownames(A549_down_closest) %in% A549_genes_down)
```

```
##
## FALSE  TRUE
##     2     8
```

```
table(rownames(A549_bkgd_closest) %in% A549_genes_down)
```

```
##
## FALSE  TRUE
##  3459   172
```

```
table(rownames(both_down_A549_closest) %in% A549_genes_down)
```

```
##
## FALSE  TRUE
##     1     5
```

```r
table(rownames(U2OS_down_closest) %in% U2OS_genes_down)
```

```
##
## FALSE  TRUE
##   136    80
```

```r
table(rownames(U2OS_bkgd_closest) %in% U2OS_genes_down)
```

```
##
## FALSE  TRUE
##  3142   487
```

```r
table(rownames(both_down_U2OS_closest) %in% U2OS_genes_down)
```

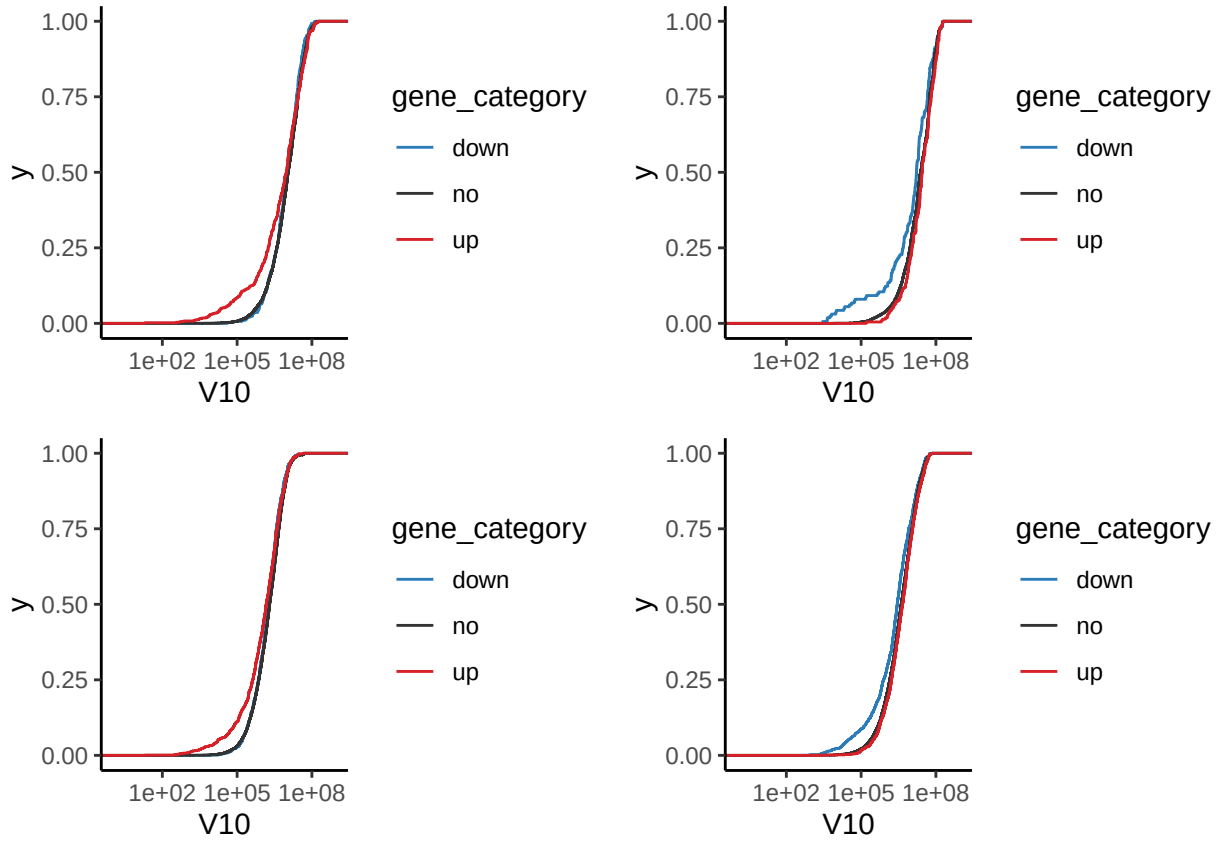
```
##
## FALSE  TRUE
##     2     4
```

Finding distance from dex-responsive genes to closest dex-responsive CCREs

```r
#Removing intragenic sites from all up- or downregulated CCREs
A549_intergenic_CCREs_up <- bedtoolsr::bt.intersect(
  a = A549_up_all_bed, b = paste0(msDataPath, 'hg38_refseq.bed'), v = T)
A549_intergenic_CCREs_down <- bedtoolsr::bt.intersect(
  a = A549_down_all_bed, b = paste0(msDataPath, 'hg38_refseq.bed'), v = T)
U2OS_intergenic_CCREs_up <- bedtoolsr::bt.intersect(
  a = U2OS_up_all_bed, b = paste0(msDataPath, 'hg38_refseq.bed'), v = T)
U2OS_intergenic_CCREs_down <- bedtoolsr::bt.intersect(
  a = U2OS_down_all_bed, b = paste0(msDataPath, 'hg38_refseq.bed'), v = T)



A549_distance_CCREs_up <- bedtoolsr::bt.closest(
  a = promA549, b = A549_intergenic_CCREs_up, d = TRUE)
A549_distance_CCREs_up$V4 <- as.character(A549_distance_CCREs_up$V4)
A549_distance_CCREs_up$gene_category <- 'no'
A549_distance_CCREs_up[
  A549_distance_CCREs_up$V4 %in% A549_genes_up, 11] <- 'up'
A549_distance_CCREs_up[
  A549_distance_CCREs_up$V4 %in% A549_genes_down, 11] <- 'down'
A549_distance_CCREs_up$CCRE_category <- 'up'
A549_distance_CCREs_up <- A549_distance_CCREs_up[
  A549_distance_CCREs_up$V10 != -1,]

A549_distance_CCREs_down <- bedtoolsr::bt.closest(
  a = promA549, b = A549_intergenic_CCREs_down, d = TRUE)
A549_distance_CCREs_down$V4 <- as.character(A549_distance_CCREs_down$V4)
A549_distance_CCREs_down$gene_category <- 'no'
A549_distance_CCREs_down[
```

```
  A549_distance_CCREs_down$V4 %in% A549_genes_up, 11] <- 'up'
A549_distance_CCREs_down[
  A549_distance_CCREs_down$V4 %in% A549_genes_down, 11] <- 'down'
A549_distance_CCREs_down$CCRE_category <- 'down'
A549_distance_CCREs_down <- A549_distance_CCREs_down[
  A549_distance_CCREs_down$V10 != -1,]

A549_distance_CCREs <- rbind(
  A549_distance_CCREs_up, A549_distance_CCREs_down)
A549_distance_CCREs$category <- paste(
  A549_distance_CCREs$gene_category,
  A549_distance_CCREs$CCRE_category, sep = '_')

p7 = ggplot(
  data = A549_distance_CCREs_up) +
  stat_ecdf(mapping = aes(x=V10, color = gene_category)) +
  scale_x_log10(limits = c(1, 1e9)) + gg_options +
  scale_color_manual(values = c('#2c7cb7', 'gray20','#d72027'))

p8 = ggplot(
  data = A549_distance_CCREs_down) +
  stat_ecdf(mapping = aes(x=V10, color = gene_category)) +
  scale_x_log10(limits = c(1, 1e9)) + gg_options+
  scale_color_manual(values = c('#2c7cb7', 'gray20','#d72027'))

U2OS_distance_CCREs_up <- bedtoolsr::bt.closest(
  a = promU2OS, b = U2OS_intergenic_CCREs_up, d = TRUE)
U2OS_distance_CCREs_up$V4 <- as.character(U2OS_distance_CCREs_up$V4)
U2OS_distance_CCREs_up$gene_category <- 'no'
U2OS_distance_CCREs_up[
  U2OS_distance_CCREs_up$V4 %in% U2OS_genes_up, 11] <- 'up'
U2OS_distance_CCREs_up[
  U2OS_distance_CCREs_up$V4 %in% U2OS_genes_down, 11] <- 'down'
U2OS_distance_CCREs_up$CCRE_category <- 'up'
U2OS_distance_CCREs_up <- U2OS_distance_CCREs_up[
  U2OS_distance_CCREs_up$V10 != -1,]

U2OS_distance_CCREs_down <- bedtoolsr::bt.closest(
  a = promU2OS, b = U2OS_intergenic_CCREs_down, d = TRUE)
U2OS_distance_CCREs_down$V4 <- as.character(
  U2OS_distance_CCREs_down$V4)
U2OS_distance_CCREs_down$gene_category <- 'no'
U2OS_distance_CCREs_down[
  U2OS_distance_CCREs_down$V4 %in% U2OS_genes_up, 11] <- 'up'
U2OS_distance_CCREs_down[
  U2OS_distance_CCREs_down$V4 %in% U2OS_genes_down, 11] <- 'down'
U2OS_distance_CCREs_down$CCRE_category <- 'down'
U2OS_distance_CCREs_down <- U2OS_distance_CCREs_down[
  U2OS_distance_CCREs_down$V10 != -1,]

U2OS_distance_CCREs <- rbind(
  U2OS_distance_CCREs_up, U2OS_distance_CCREs_down)
U2OS_distance_CCREs <- U2OS_distance_CCREs[
```

```
  U2OS_distance_CCREs$V10 != -1,]
U2OS_distance_CCREs$category <- paste(
  U2OS_distance_CCREs$gene_category,
  U2OS_distance_CCREs$CCRE_category, sep = '_')

p9 = ggplot(data = U2OS_distance_CCREs_up) +
  stat_ecdf(mapping = aes(x=V10, color = gene_category)) +
  gg_options + scale_x_log10(limits = c(1, 1e9)) +
  scale_color_manual(values = c('#2c7cb7', 'gray20','#d72027'))

p10 = ggplot(data = U2OS_distance_CCREs_down) +
  stat_ecdf(mapping = aes(x=V10, color = gene_category)) +
  scale_x_log10(limits = c(1, 1e9)) + gg_options +
  scale_color_manual(values = c('#2c7cb7', 'gray20','#d72027'))

p11 = grid.arrange(p7, p8, p9, p10, nrow =2)
```




```
wilcox.test(
  A549_distance_CCREs_up[
    A549_distance_CCREs_up$gene_category=='up',10],
  A549_distance_CCREs_up[
    A549_distance_CCREs_up$gene_category=='no',10])
```

```
##
##  Wilcoxon rank sum test with continuity correction
```

```
##
## data:  A549_distance_CCREs_up[A549_distance_CCREs_up$gene_category ==  and A549_distance_CCREs_up[A54
## W = 2310506, p-value = 8.501e-05
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  A549_distance_CCREs_up[
    A549_distance_CCREs_up$gene_category=='down',10],
  A549_distance_CCREs_up[
    A549_distance_CCREs_up$gene_category=='no',10])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_distance_CCREs_up[A549_distance_CCREs_up$gene_category ==  and A549_distance_CCREs_up[A54
## W = 1478150, p-value = 0.1787
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  A549_distance_CCREs_up[
    A549_distance_CCREs_up$gene_category=='up',10],
  A549_distance_CCREs_up[
    A549_distance_CCREs_up$gene_category=='down',10])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_distance_CCREs_up[A549_distance_CCREs_up$gene_category ==  and A549_distance_CCREs_up[A54
## W = 71729, p-value = 0.108
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  U2OS_distance_CCREs_up[
    U2OS_distance_CCREs_up$gene_category=='up',10],
            U2OS_distance_CCREs_up[
              U2OS_distance_CCREs_up$gene_category=='no',10])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_distance_CCREs_up[U2OS_distance_CCREs_up$gene_category ==  and U2OS_distance_CCREs_up[U2O
## W = 4643192, p-value = 5.823e-13
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  U2OS_distance_CCREs_up[
    U2OS_distance_CCREs_up$gene_category=='down',10],
            U2OS_distance_CCREs_up[
              U2OS_distance_CCREs_up$gene_category=='no',10])
```

```
##
```

```
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_distance_CCREs_up[U2OS_distance_CCREs_up$gene_category ==  and U2OS_distance_CCREs_up[U2
## W = 5299462, p-value = 0.0004485
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  U2OS_distance_CCREs_up[
    U2OS_distance_CCREs_up$gene_category=='up',10],
            U2OS_distance_CCREs_up[
              U2OS_distance_CCREs_up$gene_category=='down',10])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_distance_CCREs_up[U2OS_distance_CCREs_up$gene_category ==  and U2OS_distance_CCREs_up[U2
## W = 696586, p-value = 0.001093
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  A549_distance_CCREs_down[
    A549_distance_CCREs_down$gene_category=='up',10],
  A549_distance_CCREs_down[
    A549_distance_CCREs_down$gene_category=='no',10])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_distance_CCREs_down[A549_distance_CCREs_down$gene_category ==  and A549_distance_CCREs_d
## W = 623978, p-value = 0.06692
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  A549_distance_CCREs_down[
    A549_distance_CCREs_down$gene_category=='down',10],
  A549_distance_CCREs_down[
    A549_distance_CCREs_down$gene_category=='no',10])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_distance_CCREs_down[A549_distance_CCREs_down$gene_category ==  and A549_distance_CCREs_d
## W = 359100, p-value = 0.0002067
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  A549_distance_CCREs_down[
    A549_distance_CCREs_down$gene_category=='up',10],
  A549_distance_CCREs_down[
    A549_distance_CCREs_down$gene_category=='down',10])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549_distance_CCREs_down[A549_distance_CCREs_down$gene_category ==  and A549_distance_CCREs_d
## W = 22138, p-value = 5.864e-05
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  U2OS_distance_CCREs_down[
    U2OS_distance_CCREs_down$gene_category=='up',10],
            U2OS_distance_CCREs_down[
              U2OS_distance_CCREs_down$gene_category=='no',10])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_distance_CCREs_down[U2OS_distance_CCREs_down$gene_category ==  and U2OS_distance_CCREs_d
## W = 5555111, p-value = 0.01572
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  U2OS_distance_CCREs_down[
    U2OS_distance_CCREs_down$gene_category=='down',10],
  U2OS_distance_CCREs_down[
    U2OS_distance_CCREs_down$gene_category=='no',10])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_distance_CCREs_down[U2OS_distance_CCREs_down$gene_category ==  and U2OS_distance_CCREs_d
## W = 4888000, p-value = 1.239e-14
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(
  U2OS_distance_CCREs_down[
    U2OS_distance_CCREs_down$gene_category=='up',10],
  U2OS_distance_CCREs_down[
    U2OS_distance_CCREs_down$gene_category=='down',10])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  U2OS_distance_CCREs_down[U2OS_distance_CCREs_down$gene_category ==  and U2OS_distance_CCREs_d
## W = 885435, p-value = 7.439e-14
## alternative hypothesis: true location shift is not equal to 0
```

Finding correlation between degree of change in transcription for dex-responsive CCREs and nearest dex-responsive gene

First finding genes and CCREs that are differentially expressed after 100 nM dex, then finding ones closest to each other

```
# Names of differentially expressed genes
genes_A549_diff <- row.names(
  geneData[geneData$A549_wt_100vs0dex_fdr < 0.05 , ])
genes_U2OS_diff <- row.names(
  geneData[geneData$U2OS_wt_100vs0dex_fdr < 0.05 , ])

A549_diff_promoters <- promA549[
  promA549$V4 %in% genes_A549_diff,]
U2OS_diff_promoters <- promU2OS[
  promU2OS$V4 %in% genes_U2OS_diff,]

A549_intergenic_CCREs <- bedtoolsr::bt.sort(
  rbind(A549_intergenic_CCREs_up, A549_intergenic_CCREs_down)
)
U2OS_intergenic_CCREs <- bedtoolsr::bt.sort(
  rbind(U2OS_intergenic_CCREs_up, U2OS_intergenic_CCREs_down)
)

bedtoolsr::bt.closest(
  a = A549_intergenic_CCREs, b = A549_diff_promoters,
  d = T, output = 'Closest_diffExp_CCRE_promoter_A549.txt'
  )

bedtoolsr::bt.closest(
  a = U2OS_intergenic_CCREs, b = U2OS_diff_promoters,
  d = T, output = 'Closest_diffExp_CCRE_promoter_U2OS.txt'
  )

write.table(CCREs[,c(11,14,15,3,7)],
            'CCREs_300bp_100nM_response.txt', sep = '\t',
            row.names = F, col.names = F, quote = F)
```

Next collating data and adding up signal from each CCRE that matches one gene.

```
distance=100000

genesIn = open('gene_DESeq_analysis.txt', 'r').readlines()[1:]
genes = {}
for i in genesIn:
  i=i.rstrip().split('\t')
  #for each gene, save log2fc after 100 nM dex for A549 and U2OS
  genes[i[0]]=[i[16], i[20]]

CCREsIn=open('CCREs_300bp_100nM_response.txt', 'r').readlines()
CCREs = {}
for i in CCREsIn:
  i=i.rstrip().split('\t')
  name = i[0] + ':' + i[1] + '-' + i[2]
  CCREs[name] = [i[3], i[4]] #for each CCRE, save log2fc after 100 nM dex for A549 and U2OS


A549_closest=open('Closest_diffExp_CCRE_promoter_A549.txt', 'r').readlines()
A549_closest_toUse={} # gene log2fc, sum of CCRE log2fc
```

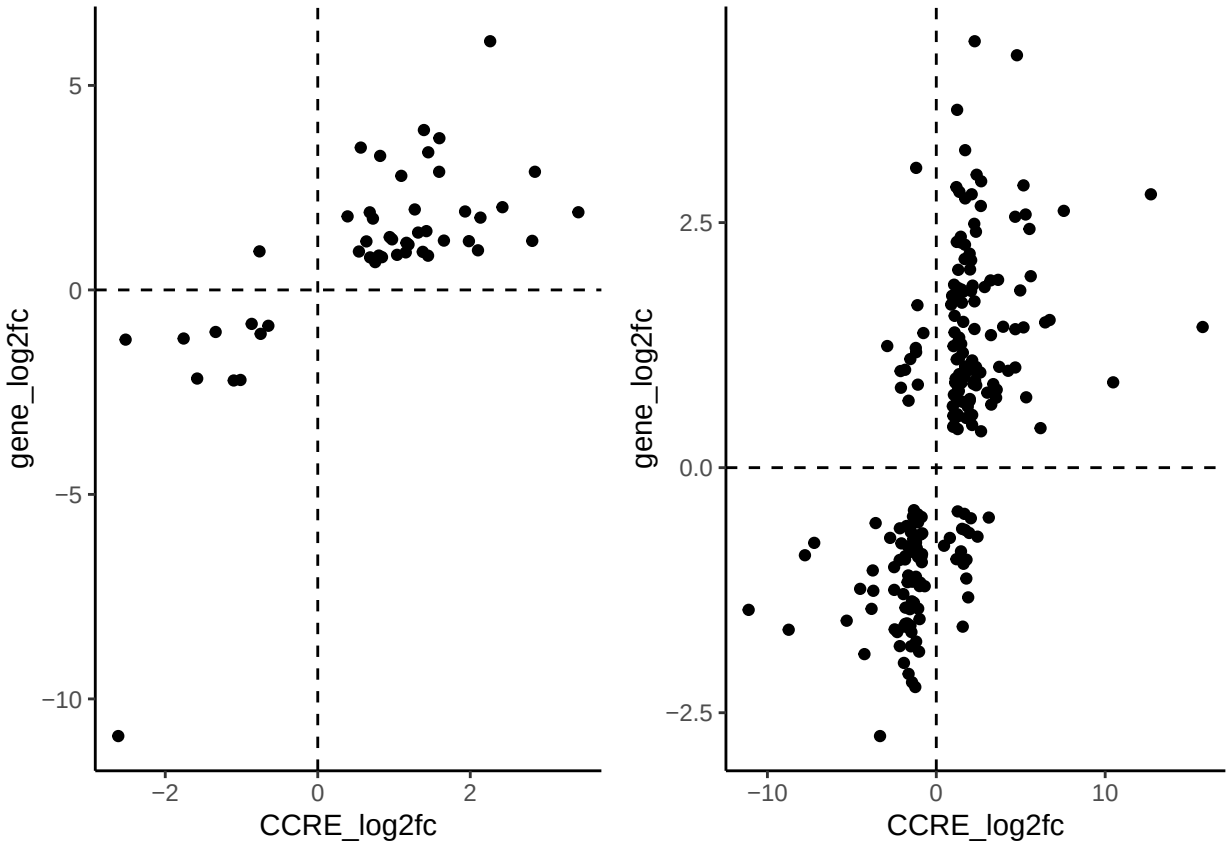
```
A549_count=0
for i in A549_closest:
  i=i.rstrip().split('\t')
  if int(i[9]) <= distance:
    A549_count=A549_count+1
    gene = i[6]
    CCRE = i[0] + ':' + i[1] + '-' + i[2]
    if gene not in A549_closest_toUse.keys():
      A549_closest_toUse[gene] = [genes[gene][0], CCREs[CCRE][0]]
    else:
      A549_closest_toUse[gene][1] = \
      str(float(A549_closest_toUse[gene][1]) + float(CCREs[CCRE][0]))

header=['gene', 'gene_log2fc', 'CCRE_log2fc']
A549out = open('Closest_diffExp_CCRE_promoter_A549_values.txt', 'w')
A549out.write('\t'.join(header) + '\n')
for i in A549_closest_toUse.keys():
  output=[i] + A549_closest_toUse[i]
  A549out.write('\t'.join(output) + '\n')
A549out.close()

U2OS_closest = open('Closest_diffExp_CCRE_promoter_U2OS.txt', 'r').readlines()
U2OS_closest_toUse = {} # gene log2fc, sum of CCRE log2fc
U2OS_count=0
for i in U2OS_closest:
  i=i.rstrip().split('\t')
  if int(i[9]) <= distance:
    U2OS_count=U2OS_count+1
    gene = i[6]
    CCRE = i[0] + ':' + i[1] + '-' + i[2]
    if gene not in U2OS_closest_toUse.keys():
      U2OS_closest_toUse[gene] = [genes[gene][1], CCREs[CCRE][1]]
    else:
      U2OS_closest_toUse[gene][1] = \
      str(float(U2OS_closest_toUse[gene][1]) + float(CCREs[CCRE][1]))

header=['gene', 'gene_log2fc', 'CCRE_log2fc']
U2OSout = open('Closest_diffExp_CCRE_promoter_U2OS_values.txt', 'w')
U2OSout.write('\t'.join(header) + '\n')
for i in U2OS_closest_toUse.keys():
  output=[i] + U2OS_closest_toUse[i]
  U2OSout.write('\t'.join(output) + '\n')
U2OSout.close()

print(A549_count,U2OS_count)
```

```
## (75, 305)
```

Now plotting the log2 fc differences for closest diff exp CCREs and genes

```
A549_closest <-
  read.table('Closest_diffExp_CCRE_promoter_A549_values.txt',
             sep = '\t', stringsAsFactors = F, skip = 1)
```

```
colnames(A549_closest) <- c('gene', 'CCRE_log2fc', 'gene_log2fc')
a549_slope <-
  lm(A549_closest$gene_log2fc ~ A549_closest$CCRE_log2fc -1)

p12 <-
  ggplot(A549_closest, aes(x=CCRE_log2fc, y=gene_log2fc)) +
  geom_point()+ gg_options +
  geom_hline(yintercept = 0, linetype = 'dashed') +
  geom_vline(xintercept = 0, linetype = 'dashed')


U2OS_closest <-
  read.table('Closest_diffExp_CCRE_promoter_U2OS_values.txt',
             sep = '\t', stringsAsFactors = F, skip = 1)
colnames(U2OS_closest) <- c('gene', 'gene_log2fc', 'CCRE_log2fc')
u2os_slope <-
  lm(U2OS_closest$gene_log2fc ~ U2OS_closest$CCRE_log2fc -1)

p13 <-
  ggplot(U2OS_closest, aes(x=CCRE_log2fc, y=gene_log2fc)) +
  geom_point() + gg_options +
  geom_hline(yintercept = 0, linetype = 'dashed') +
  geom_vline(xintercept = 0, linetype = 'dashed')

p14 <- grid.arrange(p12, p13, nrow = 1)
```



```r
print(a549_slope)
```

```
##
## Call:
## lm(formula = A549_closest$gene_log2fc ~ A549_closest$CCRE_log2fc -
##     1)
##
## Coefficients:
## A549_closest$CCRE_log2fc
##                    1.273
```

```r
print(u2os_slope)
```

```
##
## Call:
## lm(formula = U2OS_closest$gene_log2fc ~ U2OS_closest$CCRE_log2fc -
##     1)
##
## Coefficients:
## U2OS_closest$CCRE_log2fc
##                   0.3068
```

```r
cor.test(A549_closest$CCRE_log2fc, A549_closest$gene_log2fc)
```

```
##
##  Pearson's product-moment correlation
##
## data:  A549_closest$CCRE_log2fc and A549_closest$gene_log2fc
## t = 7.1137, df = 46, p-value = 6.155e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5534388 0.8359802
## sample estimates:
##       cor
## 0.7237612
```

```r
cor.test(U2OS_closest$CCRE_log2fc, U2OS_closest$gene_log2fc)
```

```
##
##  Pearson's product-moment correlation
##
## data:  U2OS_closest$CCRE_log2fc and U2OS_closest$gene_log2fc
## t = 11.06, df = 212, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5123093 0.6835327
## sample estimates:
##      cor
## 0.604866
```