

Code to generate Figure 3 of ‘Glucocorticoid receptor collaborates with pioneer factors and AP-1 to execute genome-wide regulation’

Erin M. Wissink

## Loading packages and setting paths

```
library(bigWig)
library(ggplot2)
library(knitr)
library(DESeq2)
library(RColorBrewer)
library(pheatmap)
library(gridExtra)
library(wesanderson)
library(UpSetR)
library(bedtoolsr)

coordinatesPath <- '../Manuscript_data/'
bigwigPath <- '../Manuscript_data/PRO-seq_bigwigs/Raw/'
dREGpath <- '../Manuscript_data/dREG_files/'

#getting lists of bigwig files
posbw <- list.files(path = bigwigPath, pattern = '*_plus.bw')
minbw <- list.files(path = bigwigPath, pattern = '*_minus.bw')

samples <- substr(posbw,4,regexpr('_plus.bw',posbw)-1)

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()
```

## Creating unified set of CCREs from dREG

I found coordinates of TREs for my 6 samples. I am now merging those TREs to have one unified set, and then will filter out any promoter-matching TREs to create a list of CCREs.

```
dREG_files <- list.files(path = dREGpath, pattern = "\\.bed$")
dREG_calls <- read.table(paste0(dREGpath, dREG_files[1]), sep = '\t')
```

```

for (f in dREG_files[-1]) {
  dREG_calls <- rbind(dREG_calls, read.table(paste0(dREGpath, f), sep = '\t'))
}

dREG_calls <- bedtoolsr::bt.sort(dREG_calls)
dREG_calls <- bedtoolsr::bt.merge(dREG_calls)

CCRE_data <- bedtoolsr::bt.intersect(v = T, a = dREG_calls,
                                      b = paste0(coordinatesPath, 'hg38_refseq_promoters.bed'))

```

Determining overlap between dREG calls that are U2OS-specific, A549-specific, or shared with promoters, intergenic, and intragenic sequences

```

dREG_calls$coord <- paste0(dREG_calls$V1, ':', dREG_calls$V2, '(', dREG_calls$V3)

toUpset <- data.frame(row.names = dREG_calls$coord)
toUpset$A549 <- 0
toUpset$U2OS <- 0
toUpset$promoter <- 0
toUpset$intergenic <- 0
toUpset$intragenic <- 0

#Finding overlap with dREG calls found in A549 OnM cells
curr <- bedtoolsr::bt.intersect(wa = T, a = dREG_calls,
                                 b = paste0(dREGpath, 'A549_0.dREG.peak.full.bed'))
curr <- paste0(curr$V1, ':', curr$V2, ')', curr$V3)
toUpset[curr,1] <- 1

#Finding overlap with dREG calls found in A549 1nM cells
curr <- bedtoolsr::bt.intersect(wa = T, a = dREG_calls,
                                 b = paste0(dREGpath, 'A549_1.dREG.peak.full.bed'))
curr <- paste0(curr$V1, ':', curr$V2, ')', curr$V3)
toUpset[curr,1] <- 1

#Finding overlap with dREG calls found in A549 100nM cells
curr <- bedtoolsr::bt.intersect(wa = T, a = dREG_calls,
                                 b = paste0(dREGpath, 'A549_100.dREG.peak.full.bed'))
curr <- paste0(curr$V1, ':', curr$V2, ')', curr$V3)
toUpset[curr,1] <- 1

#Finding overlap with dREG calls found in U2OS OnM cells
curr <- bedtoolsr::bt.intersect(wa = T, a = dREG_calls,
                                 b = paste0(dREGpath, 'U2OS_0.dREG.peak.full.bed'))
curr <- paste0(curr$V1, ':', curr$V2, ')', curr$V3)
toUpset[curr,2] <- 1

#Finding overlap with dREG calls found in U2OS 1nM cells
curr <- bedtoolsr::bt.intersect(wa = T, a = dREG_calls,
                                 b = paste0(dREGpath, 'U2OS_1.dREG.peak.full.bed'))
curr <- paste0(curr$V1, ':', curr$V2, ')', curr$V3)

```

```

toUpset[curr,2] <- 1

#Finding overlap with dREG calls found in U2OS 100nM cells
curr <- bedtoolsr::bt.intersect(wa = T, a = dREG_calls,
                                b = paste0(dREGpath, 'U2OS_100.dREG.peak.full.bed'))
curr <- paste0(curr$V1, ':', curr$V2, '-', curr$V3)
toUpset[curr,2] <- 1

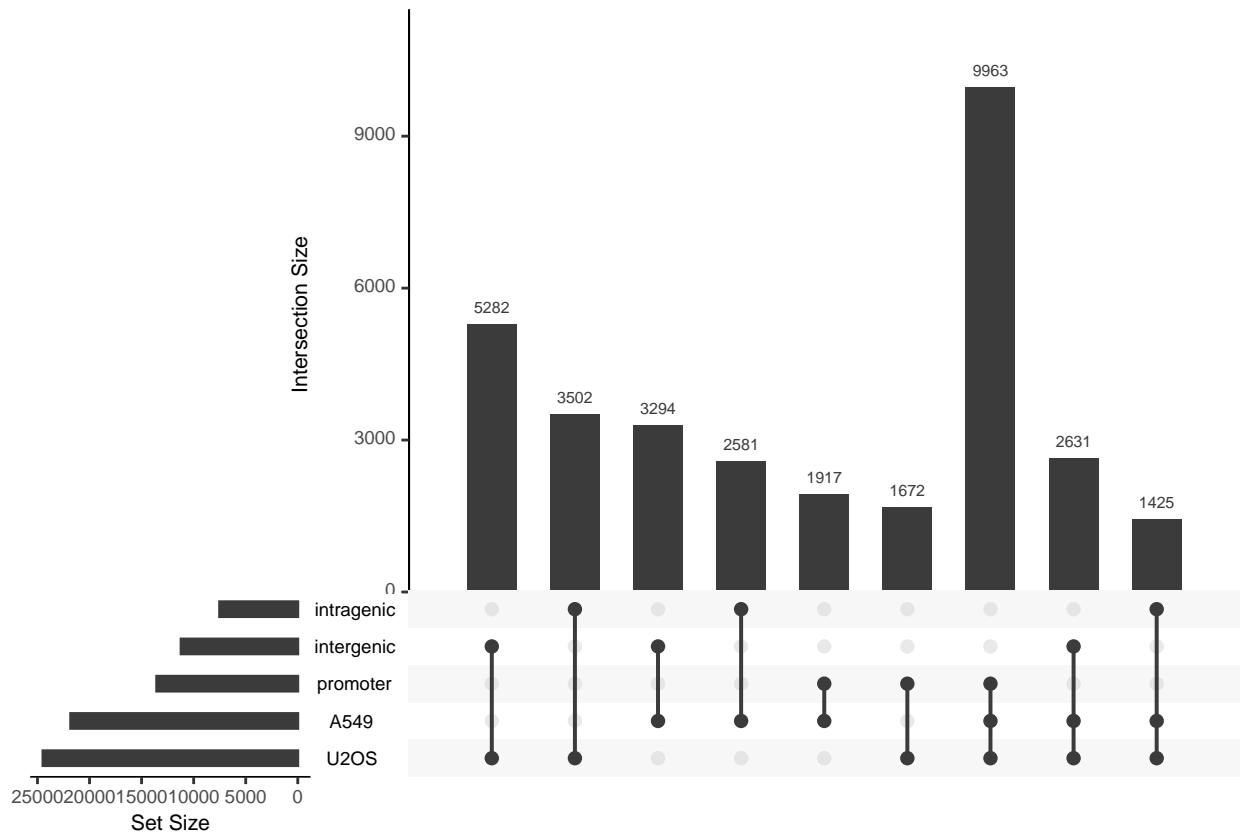
#Finding overlap with promoters
curr <- bedtoolsr::bt.intersect(u = T, a = dREG_calls,
                                b = paste0(coordinatesPath, 'hg38_refseq_promoters.bed'))
curr <- paste0(curr$V1, ':', curr$V2, '-', curr$V3)
toUpset[curr,3] <- 1

#Finding CCREs with overlap with intergenic sites
curr <- bedtoolsr::bt.intersect(v = T, a = CCRE_data,
                                b = paste0(coordinatesPath, 'hg38_refseq.bed'))
curr <- paste0(curr$V1, ':', curr$V2, '-', curr$V3)
toUpset[curr,4] <- 1

#Finding CCREs overlap with intragenic sites
curr <- bedtoolsr::bt.intersect(u = T, a = CCRE_data,
                                b = paste0(coordinatesPath, 'hg38_refseq.bed'))
curr <- paste0(curr$V1, ':', curr$V2, '-', curr$V3)
toUpset[curr,5] <- 1

upset(toUpset)

```



## Finding read counts for each CCRE with the bigWig package

```
#creating columns in dataframe for data
colnames(CCRE_data) <- c('chrom', 'start', 'end')
for (i in 1:length(samples)){
  CCRE_data[,samples[i]] <- NA
}

for (i in 1:length(samples)){
  pbw <- load.bigWig(paste(bigwigPath, posbw[i], sep = ""))
  mbw <- load.bigWig(paste(bigwigPath, minbw[i], sep = ""))
  CCRE_data[i+3] <- bed.region.bpQuery.bigWig(
    bw = pbw, bed = CCRE_data[1:3], op = 'sum', abs.value = TRUE
  ) + bed.region.bpQuery.bigWig(
    bw = mbw, bed = CCRE_data[1:3], op = 'sum', abs.value = TRUE
  )
}

write.table(CCRE_data, "GR_CCRE_counts.txt", quote = FALSE, sep = '\t')
```

Running DESeq2 to find differentially expressed CCREs. Because I expect large cell-type differences, I'm running DESeq separately on U2OS and A549 cells.

```
row.names(CCRE_data) <- paste0(CCRE_data$chrom, ':', CCRE_data$start, ' ', CCRE_data$end)

CCRE_cpm <- CCRE_data[,4:15]
for (i in 1:12){
  CCRE_cpm[,i] <- CCRE_cpm[,i]/sum(CCRE_cpm[,i])*1000000
}

idx <- rowSums(CCRE_cpm) >= 10) >= 3

for_A549_DESeq <- CCRE_data[idx, 4:9]
DESeq_table_A549 <- data.frame(row.names = colnames(for_A549_DESeq))
DESeq_table_A549$treatment <- factor(c('1','1','0','0','100','100'))

dds_A549 <- DESeqDataSetFromMatrix (
  countData= for_A549_DESeq, colData = DESeq_table_A549, design= ~treatment
)
dds_A549 <- estimateSizeFactors(dds_A549)
dds_A549 <- DESeq(dds_A549)

A549_wt_1vs0dex <- lfcShrink(dds_A549, coef = 'treatment_1_vs_0')
A549_wt_100vs0dex <- lfcShrink(dds_A549, coef = 'treatment_100_vs_0')

vsd_A549 <- as.data.frame(assay(vst(dds_A549, blind = FALSE)))

for_U2OS_DESeq <- CCRE_data[idx, 10:15]
DESeq_table_U2OS <- data.frame(row.names = colnames(for_U2OS_DESeq))
DESeq_table_U2OS$treatment <- factor(c('1','1','0','0','100','100'))

dds_U2OS <- DESeqDataSetFromMatrix (
  countData= for_U2OS_DESeq, colData = DESeq_table_U2OS, design= ~treatment
)
dds_U2OS <- estimateSizeFactors(dds_U2OS)
dds_U2OS <- DESeq(dds_U2OS)

U2OS_wt_1vs0dex <- lfcShrink(dds_U2OS, coef = 'treatment_1_vs_0')
U2OS_wt_100vs0dex <- lfcShrink(dds_U2OS, coef = 'treatment_100_vs_0')

vsd_U2OS <- as.data.frame(assay(vst(dds_U2OS, blind = FALSE)))

CCRE_DESeq_results <- cbind(vsd_A549, vsd_U2OS)

CCRE_DESeq_results$A549_baseMean <- A549_wt_1vs0dex$baseMean
CCRE_DESeq_results$A549_wt_1vs0dex_log2 <- A549_wt_1vs0dex$log2FoldChange
CCRE_DESeq_results$A549_wt_1vs0dex_fdr <- A549_wt_1vs0dex$padj
CCRE_DESeq_results$A549_wt_100vs0dex_log2 <- A549_wt_100vs0dex$log2FoldChange
CCRE_DESeq_results$A549_wt_100vs0dex_fdr <- A549_wt_100vs0dex$padj

CCRE_DESeq_results$U2OS_baseMean <- U2OS_wt_1vs0dex$baseMean
CCRE_DESeq_results$U2OS_wt_1vs0dex_log2 <- U2OS_wt_1vs0dex$log2FoldChange
```

```

CCRE_DESeq_results$U2OS_wt_1vs0dex_fdr <- U2OS_wt_1vs0dex$padj
CCRE_DESeq_results$U2OS_wt_100vs0dex_log2 <- U2OS_wt_100vs0dex$log2FoldChange
CCRE_DESeq_results$U2OS_wt_100vs0dex_fdr <- U2OS_wt_100vs0dex$padj

write.table(CCRE_DESeq_results, 'CCRE_DESeq_analysis.txt', quote = F, sep = '\t')

```

## PCA in Supp. Fig. 2

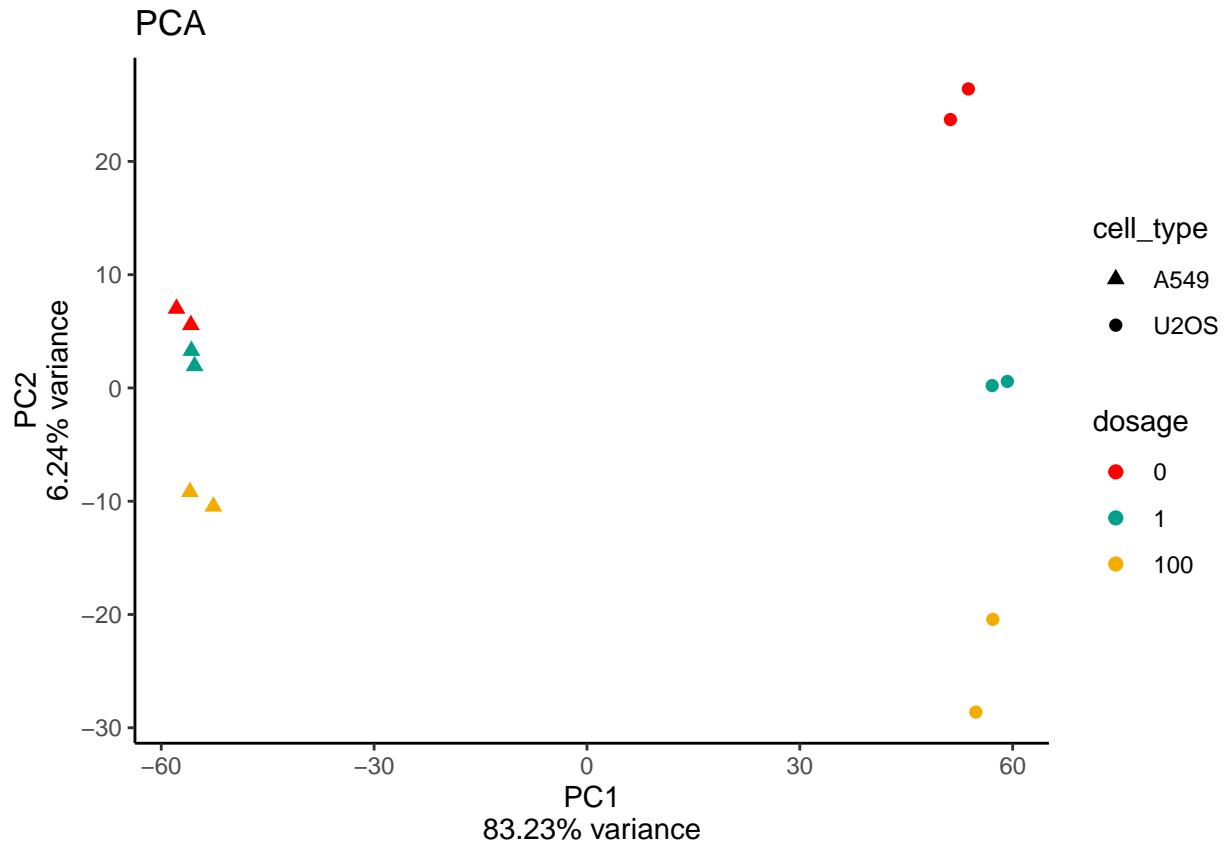
```

forPCA <- t(CCRE_DESeq_results[,c(3,4,1,2,5,6,9,10,7,8,11,12)])
pca <- prcomp(forPCA)
summary(pca)

## Importance of components:
##                PC1       PC2       PC3       PC4       PC5       PC6       PC7
## Standard deviation   58.0717 15.90645 9.34966 8.82452 8.13890 6.86862 6.53022
## Proportion of Variance  0.8323  0.06244 0.02157 0.01922 0.01635 0.01164 0.01052
## Cumulative Proportion  0.8323  0.89474 0.91632 0.93553 0.95188 0.96353 0.97405
##                  PC8       PC9       PC10      PC11      PC12
## Standard deviation   6.00210 5.36653 4.88179 4.06002 4.651e-14
## Proportion of Variance  0.00889 0.00711 0.00588 0.00407 0.000e+00
## Cumulative Proportion  0.98294 0.99005 0.99593 1.00000 1.000e+00

pca_values <- as.data.frame(pca$x)
pca_values$cell_type <- c(rep(c('A549','A549'),3), rep(c('U2OS','U2OS'),3))
pca_values$dosage <- rep(c(rep('0',2), rep('1',2), rep('100',2)),2)
ggplot(pca_values, aes(x = PC1, y = PC2, color = dosage, shape = cell_type)) +
  geom_point(size=2) + ggtitle("PCA") +
  xlab('PC1\n83.23% variance') + ylab('PC2\n6.24% variance') +
  scale_shape_manual(values=c(17,16)) +
  scale_color_manual(values = wes_palette(n=3, name = "Darjeeling1")) +
  gg_options

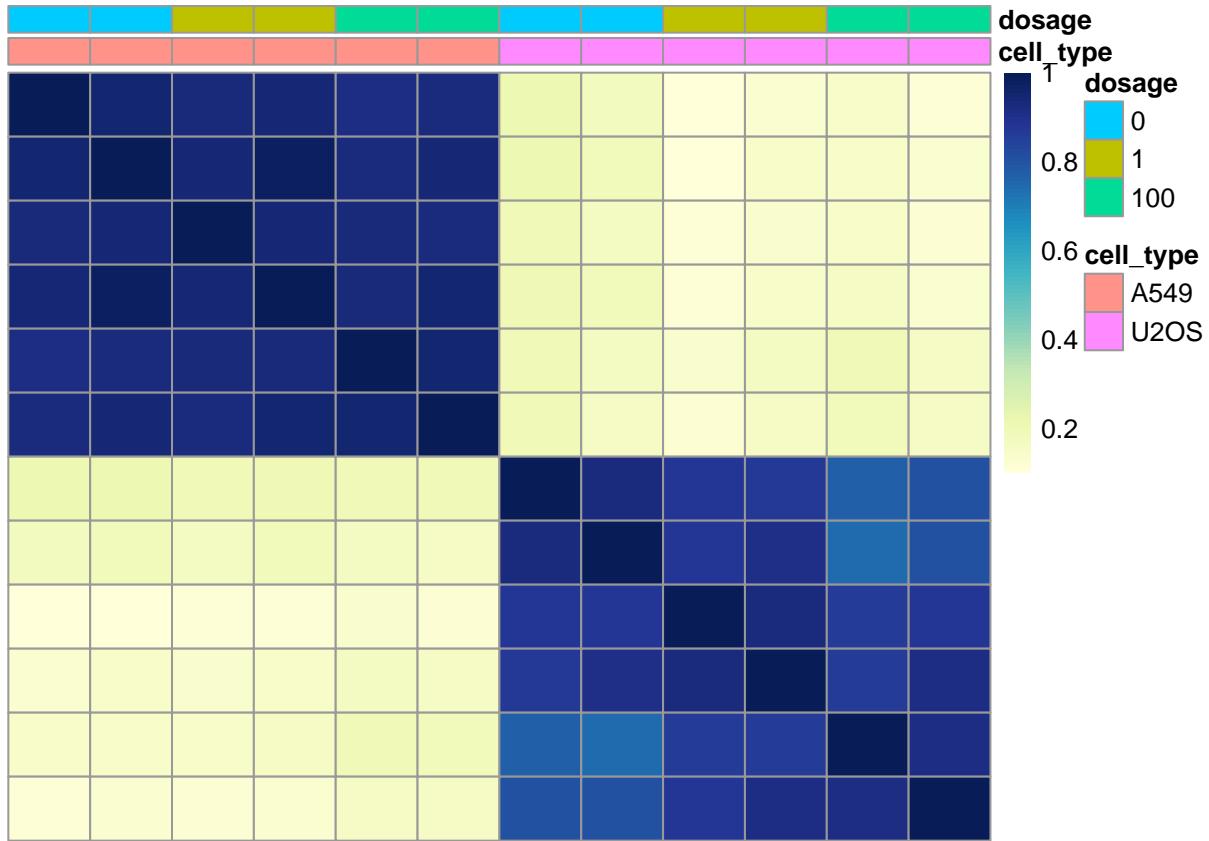
```



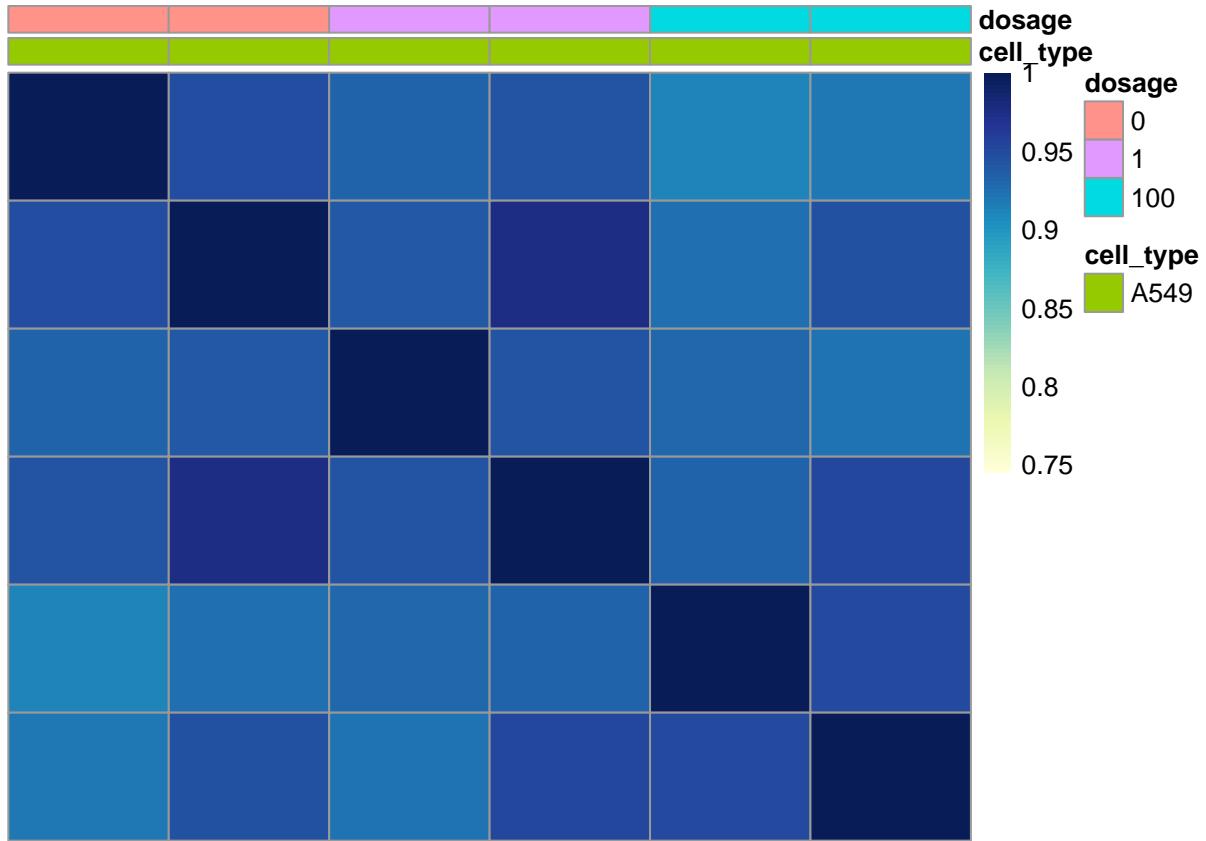
Spearman correlation plots in Supp. Fig. 2

```
# correlation matrix
corr_matrix <- as.data.frame(matrix(nrow = 12, ncol = 12))
rownames(corr_matrix) <- rownames(pca_values)
colnames(corr_matrix) <- rownames(pca_values)

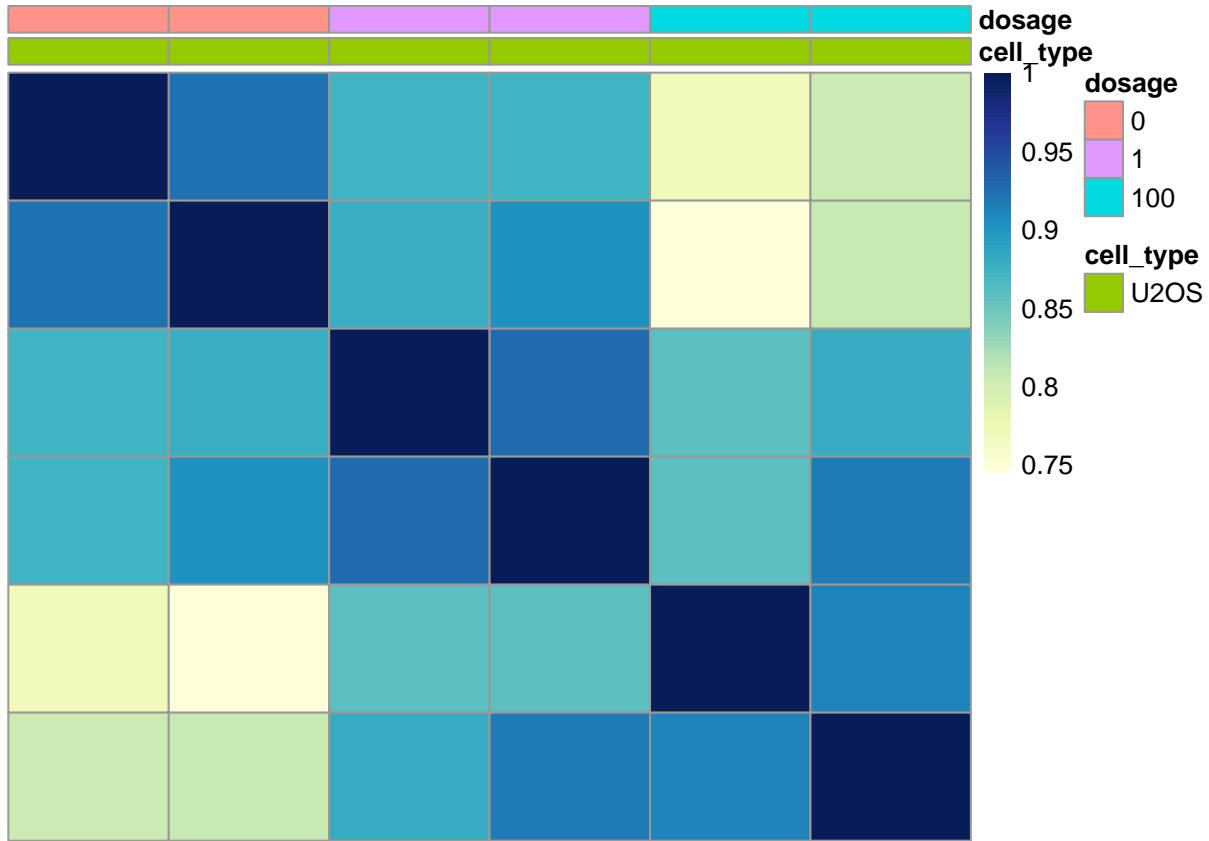
toUse <- c(3,4,1,2,5,6,9,10,7,8,11,12)
for (i in 1:length(toUse)){
  for (j in 1:length(toUse)){
    corr_matrix[i,j] <- cor.test(
      CCRE_DESeq_results[,toUse[i]],
      CCRE_DESeq_results[,toUse[j]],
      method = 'spearman')$estimate
  }
}
pheatmap(corr_matrix,
          color = colorRampPalette(brewer.pal(n = 9, name = "YlGnBu"))(100),
          breaks = seq(0.1, 1.0, length.out = 101),
          cluster_rows = F, cluster_cols = F,
          annotation_col = pca_values[13:14],
          show_rownames = F, show_colnames = F)
```



```
pheatmap(corr_matrix[1:6,1:6],  
         color = colorRampPalette(brewer.pal(n = 9, name = "YlGnBu"))(100),  
         breaks = seq(0.745, 1.0, length.out = 101),  
         cluster_rows = F, cluster_cols = F,  
         annotation_col = pca_values[1:6,13:14],  
         show_rownames = F, show_colnames = F)
```



```
phewmap(corr_matrix[7:12,7:12],  
        color = colorRampPalette(brewer.pal(n = 9, name = "YlGnBu"))(100),  
        breaks = seq(0.745, 1.0, length.out = 101),  
        cluster_rows = F, cluster_cols = F,  
        annotation_col = pca_values[7:12,13:14],  
        show_rownames = F, show_colnames = F)
```



MA plots and comparison plot for Fig. 3 B-D

```

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

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

CCRE_DESeq_results$sig_1nM <- ''
CCRE_DESeq_results$sig_100nM <- ''

for (i in 1:nrow(CCRE_DESeq_results)){
  if (CCRE_DESeq_results[i,15] < 0.05 & CCRE_DESeq_results[i,20] < 0.05){
    CCRE_DESeq_results[i,23] <- 'both'
  } else if (CCRE_DESeq_results[i,15] < 0.05){
    CCRE_DESeq_results[i,23] <- 'A549'
  } else if (CCRE_DESeq_results[i,20] < 0.05){
    CCRE_DESeq_results[i,23] <- 'U2OS'
  } else {
    CCRE_DESeq_results[i,23] <- 'none'
  }
}

```

```

if (CCRE_DESeq_results[i,17] < 0.05 & CCRE_DESeq_results[i,22] < 0.05){
  CCRE_DESeq_results[i,24] <- 'both'
} else if (CCRE_DESeq_results[i,17] < 0.05){
  CCRE_DESeq_results[i,24] <- 'A549'
} else if (CCRE_DESeq_results[i,22] < 0.05){
  CCRE_DESeq_results[i,24] <- 'U2OS'
} else {
  CCRE_DESeq_results[i,24] <- 'none'
}
}

CCRE_DESeq_results$sig_1nM <- factor(CCRE_DESeq_results$sig_1nM,
                                         levels = c("none", "U2OS", "A549", "both"))
CCRE_DESeq_results$sig_100nM <- factor(CCRE_DESeq_results$sig_100nM,
                                         levels = c("none", "U2OS", "A549", "both"))

CCRE_DESeq_results <- CCRE_DESeq_results[order(CCRE_DESeq_results$sig_1nM),]

p1 <- ggplot(CCRE_DESeq_results, aes(x = log10(1+A549_baseMean),
y = A549_wt_1vs0dex_log2,
color = A549_wt_1vs0dex_fdr < 0.05)) +
  geom_point(size = 0.75, stroke = 0) +
  xlim(0, 3.25) + ylim(-2.75, 4.25) +
  scale_color_manual(values = c('#515151', '#7570b3')) +
  xlab('mean of normalized counts') +
  ylab('log2 fold-change') + ggtitle('A549 1 nM dex') +
  geom_hline(yintercept = 0, linetype = 'dashed') +
  labs(colour = 'sig') + gg_options

p2 <- ggplot(CCRE_DESeq_results, aes(x = log10(1+U2OS_baseMean),
y = U2OS_wt_1vs0dex_log2,
color = U2OS_wt_1vs0dex_fdr < 0.05)) +
  geom_point(size = 0.75, stroke = 0) +
  xlim(0, 3.25) + ylim(-2.75, 4.25) +
  scale_color_manual(values = c('#515151', '#1b9e77')) +
  xlab('mean of normalized counts') +
  ylab('log2 fold-change') + ggtitle('U2OS 1 nM dex') +
  geom_hline(yintercept = 0, linetype = 'dashed') +
  labs(colour = "sig") + gg_options

CCRE_DESeq_results <- CCRE_DESeq_results[order(CCRE_DESeq_results$sig_100nM),]

p3 <- ggplot(CCRE_DESeq_results, aes(x = log10(1+A549_baseMean),
y = A549_wt_100vs0dex_log2,
color = A549_wt_100vs0dex_fdr < 0.05)) +
  geom_point(size = 0.75, stroke = 0) +
  xlim(0, 3.25) + ylim(-2.75, 4.25) +
  scale_color_manual(values = c('#515151', '#7570b3')) +
  xlab('mean of normalized counts') +
  ylab('log2 fold-change') + ggtitle('A549 100 nM dex') +
  geom_hline(yintercept = 0, linetype = 'dashed') +
  labs(colour = 'sig') + gg_options

p4 <- ggplot(CCRE_DESeq_results, aes(x = log10(1+U2OS_baseMean),

```

```

y = U2OS_wt_100vs0dex_log2,
color = U2OS_wt_100vs0dex_fdr < 0.05) +
geom_point(size = 0.75, stroke = 0) +
xlim(0, 3.25) + ylim(-2.75, 4.25) +
scale_color_manual(values = c('#515151', '#1b9e77')) +
xlab('mean of normalized counts') +
ylab('log2 fold-change') + ggtitle('U2OS 100 nM dex') +
geom_hline(yintercept = 0, linetype = 'dashed') +
labs(colour = "sig") + gg_options

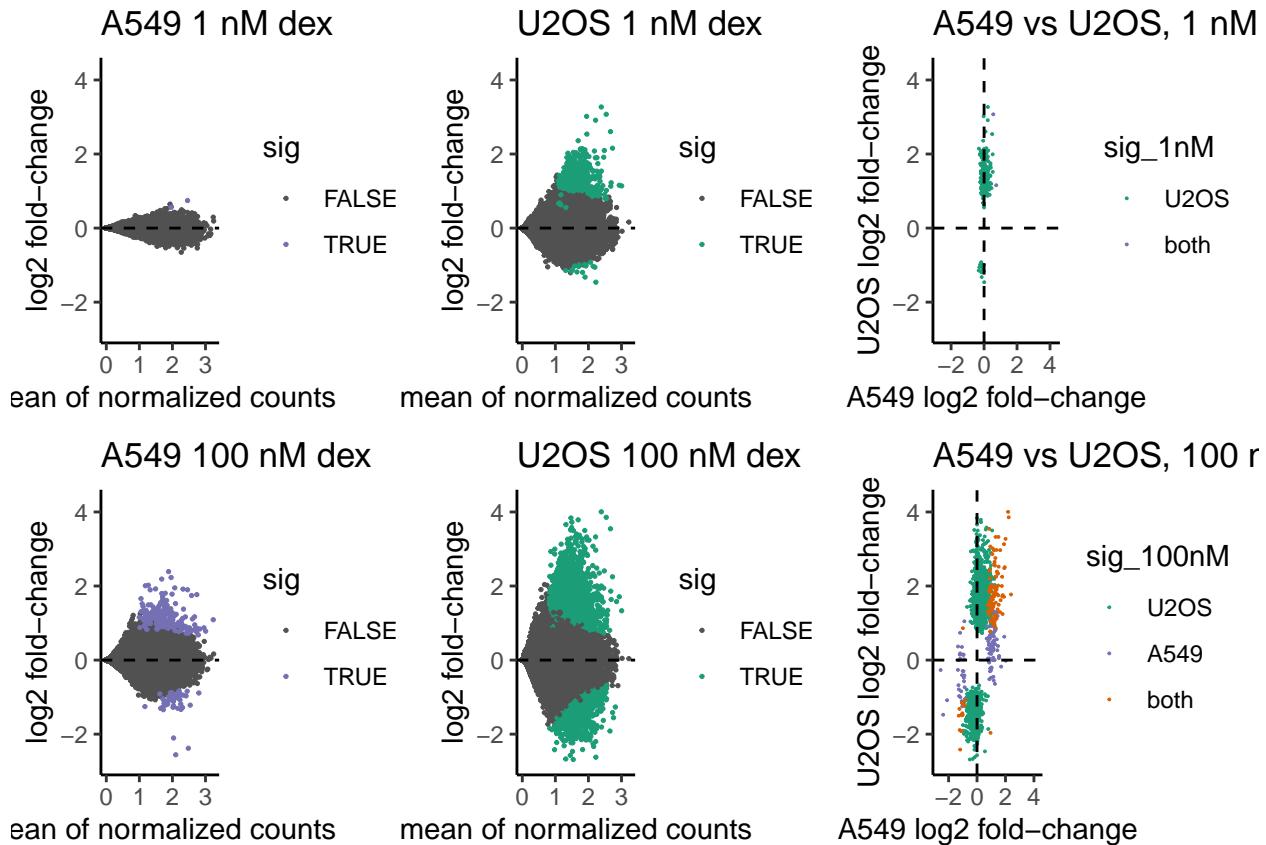
sig1 <- CCRE_DESeq_results[CCRE_DESeq_results$sig_1nM != 'none',]

p5 <- ggplot(sig1, aes(x = A549_wt_1vs0dex_log2,
y = U2OS_wt_1vs0dex_log2, color = sig_1nM)) +
geom_point(size = 0.5, stroke = 0) +
scale_color_manual(values = c("#1b9e77", "#7570b3", "#d95f02")) +
xlab('A549 log2 fold-change') +
ylab('U2OS log2 fold-change') + ggtitle('A549 vs U2OS, 1 nM dex') +
xlim(-2.75, 4.25) + ylim(-2.75, 4.25) +
geom_vline(xintercept = 0, linetype = 'dashed') +
geom_hline(yintercept = 0, linetype = 'dashed') + gg_options

sig100 <- CCRE_DESeq_results[CCRE_DESeq_results$sig_100nM != 'none',]
p6 <- ggplot(sig100, aes(x = A549_wt_100vs0dex_log2,
y = U2OS_wt_100vs0dex_log2, color = sig_100nM)) +
geom_point(size = 0.5, stroke = 0) +
scale_color_manual(values = c("#1b9e77", "#7570b3", "#d95f02")) +
xlab('A549 log2 fold-change') +
ylab('U2OS log2 fold-change') + ggtitle('A549 vs U2OS, 100 nM dex') +
xlim(-2.75, 4.25) + ylim(-2.75, 4.25) +
geom_vline(xintercept = 0, linetype = 'dashed') +
geom_hline(yintercept = 0, linetype = 'dashed') + gg_options

p7 <- grid.arrange(p1, p2, p5, p3, p4, p6, nrow = 2)

```



```
cor.test(CCRE_DESeq_results$A549_wt_1vs0dex_log2,
         CCRE_DESeq_results$U2OS_wt_1vs0dex_log2,
         method = 'spearman')
```

```
## Warning in cor.test.default(CCRE_DESeq_results$A549_wt_1vs0dex_log2,
## CCRE_DESeq_results$U2OS_wt_1vs0dex_log2, : Cannot compute exact p-value with
## ties

##
## Spearman's rank correlation rho
##
## data: CCRE_DESeq_results$A549_wt_1vs0dex_log2 and CCRE_DESeq_results$U2OS_wt_1vs0dex_log2
## S = 6.4937e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.1277438
```

```
cor.test(CCRE_DESeq_results$A549_wt_100vs0dex_log2,
         CCRE_DESeq_results$U2OS_wt_100vs0dex_log2,
         method = 'spearman')
```

```
## Warning in cor.test.default(CCRE_DESeq_results$A549_wt_100vs0dex_log2,
## CCRE_DESeq_results$U2OS_wt_100vs0dex_log2, : Cannot compute exact p-value with
## ties
```

```

##  

## Spearman's rank correlation rho  

##  

## data: CCRE_DESeq_results$A549_wt_100vs0dex_log2 and CCRE_DESeq_results$U2OS_wt_100vs0dex_log2  

## S = 5.3538e+11, p-value < 2.2e-16  

## alternative hypothesis: true rho is not equal to 0  

## sample estimates:  

## rho  

## 0.2808562

print (c('Number of significantly upregulated genes in A549 after 1 nM dex',
       nrow(CCRE_DESeq_results[CCRE_DESeq_results$A549_wt_1vs0dex_fdr<0.05 &
                               CCRE_DESeq_results$A549_wt_1vs0dex_log2>0,
                               ])))

## [1] "Number of significantly upregulated genes in A549 after 1 nM dex"
## [2] "2"

print (c('Number of significantly downregulated genes in A549 after 1 nM dex',
       nrow(CCRE_DESeq_results[CCRE_DESeq_results$A549_wt_1vs0dex_fdr<0.05 &
                               CCRE_DESeq_results$A549_wt_1vs0dex_log2<0,
                               ])))

## [1] "Number of significantly downregulated genes in A549 after 1 nM dex"
## [2] "0"

print (c('Number of significantly upregulated genes in A549 after 100 nM dex',
       nrow(CCRE_DESeq_results[CCRE_DESeq_results$A549_wt_100vs0dex_fdr<0.05 &
                               CCRE_DESeq_results$A549_wt_100vs0dex_log2>0,
                               ])))

## [1] "Number of significantly upregulated genes in A549 after 100 nM dex"
## [2] "177"

print (c('Number of significantly downregulated genes in A549 after 100 nM dex',
       nrow(CCRE_DESeq_results[CCRE_DESeq_results$A549_wt_100vs0dex_fdr<0.05 &
                               CCRE_DESeq_results$A549_wt_100vs0dex_log2<0,
                               ])))

## [1] "Number of significantly downregulated genes in A549 after 100 nM dex"
## [2] "47"

print (c('Number of significantly upregulated genes in U2OS after 1 nM dex',
       nrow(CCRE_DESeq_results[CCRE_DESeq_results$U2OS_wt_1vs0dex_fdr<0.05 &
                               CCRE_DESeq_results$U2OS_wt_1vs0dex_log2>0,
                               ])))

## [1] "Number of significantly upregulated genes in U2OS after 1 nM dex"
## [2] "395"

```

```

print (c('Number of significantly downregulated genes in U2OS after 1 nM dex',
       nrow(CCRE_DESeq_results[CCRE_DESeq_results$U2OS_wt_1vs0dex_fdr<0.05 &
                               CCRE_DESeq_results$U2OS_wt_1vs0dex_log2<0,
                               ])))

## [1] "Number of significantly downregulated genes in U2OS after 1 nM dex"
## [2] "16"

print (c('Number of significantly upregulated genes in U2OS after 100 nM dex',
       nrow(CCRE_DESeq_results[CCRE_DESeq_results$U2OS_wt_100vs0dex_fdr<0.05 &
                               CCRE_DESeq_results$U2OS_wt_100vs0dex_log2>0,
                               ])))

## [1] "Number of significantly upregulated genes in U2OS after 100 nM dex"
## [2] "1104"

print (c('Number of significantly downregulated genes in U2OS after 100 nM dex',
       nrow(CCRE_DESeq_results[CCRE_DESeq_results$U2OS_wt_100vs0dex_fdr<0.05 &
                               CCRE_DESeq_results$U2OS_wt_100vs0dex_log2<0,
                               ])))

## [1] "Number of significantly downregulated genes in U2OS after 100 nM dex"
## [2] "509"

```

## Comparing 1 nM to 100 nM treatment per cell type

```

CCRE_DESeq_results$A549_sig <- ''
CCRE_DESeq_results$U2OS_sig <- ''

for (i in 1:nrow(CCRE_DESeq_results)){
  if (CCRE_DESeq_results[i,15] < 0.05 & CCRE_DESeq_results[i,17] < 0.05){
    CCRE_DESeq_results[i,25] <- 'both'
  } else if (CCRE_DESeq_results[i, 15] < 0.05){
    CCRE_DESeq_results[i,25] <- '1nM'
  } else if (CCRE_DESeq_results[i, 17] < 0.05){
    CCRE_DESeq_results[i,25] <- '100nM'
  } else{
    CCRE_DESeq_results[i,25] <- 'none'
  }

  if (CCRE_DESeq_results[i,20] < 0.05 & CCRE_DESeq_results[i,22] < 0.05){
    CCRE_DESeq_results[i,26] <- 'both'
  } else if (CCRE_DESeq_results[i, 20] < 0.05){
    CCRE_DESeq_results[i,26] <- '1nM'
  } else if (CCRE_DESeq_results[i, 22] < 0.05){
    CCRE_DESeq_results[i,26] <- '100nM'
  } else{
    CCRE_DESeq_results[i,26] <- 'none'
  }
}

```

```

CCRE_DESeq_results$A549_sig <- factor(CCRE_DESeq_results$A549_sig,
                                         levels = c("none", "1nM", "100nM", "both"))
CCRE_DESeq_results$U20S_sig <- factor(CCRE_DESeq_results$U20S_sig,
                                         levels = c("none", "1nM", "100nM", "both"))

p8 <- ggplot(CCRE_DESeq_results, aes(x = A549_wt_1vs0dex_log2,
                                       y = A549_wt_100vs0dex_log2, color = A549_sig)) +
  geom_point(size = 1, stroke = 0) + xlab('1 nM log2 fold-change') +
  ylab('100 nM log2 fold-change') + ggtitle('A549') + xlim(-3.75, 7.25) +
  ylim(-3.75, 7.25) + geom_vline(xintercept = 0, linetype = 'dashed') +
  geom_hline(yintercept = 0, linetype = 'dashed') + gg_options +
  scale_color_manual(values = c('#515151', '#E58601', '#B40F20'))

cor.test(CCRE_DESeq_results$A549_wt_1vs0dex_log2,
         CCRE_DESeq_results$A549_wt_100vs0dex_log2,
         method = 'spearman')

## Spearman's rank correlation rho
## data: CCRE_DESeq_results$A549_wt_1vs0dex_log2 and CCRE_DESeq_results$A549_wt_100vs0dex_log2
## S = 7.8544e+11, p-value = 1.591e-12
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## -0.0550251

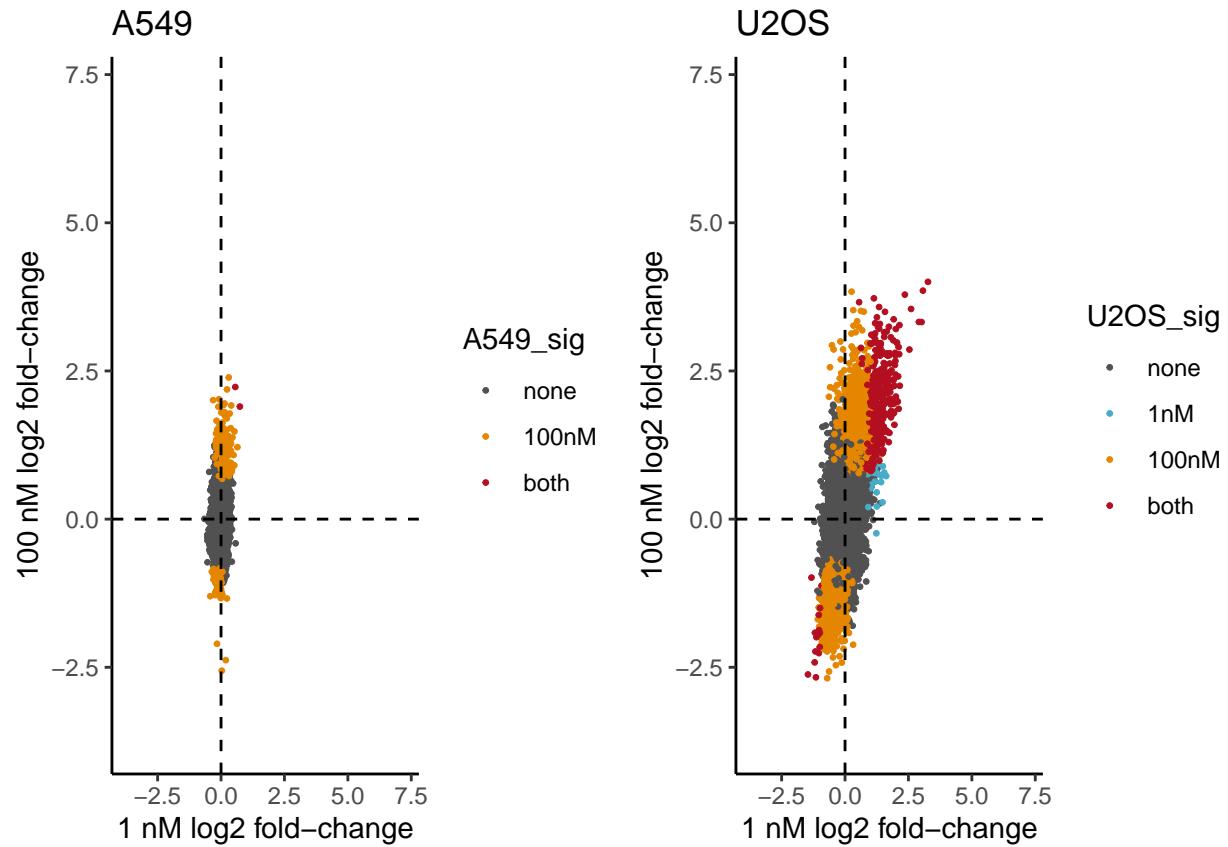
p9 <- ggplot(CCRE_DESeq_results, aes(x = U20S_wt_1vs0dex_log2,
                                       y = U20S_wt_100vs0dex_log2, color = U20S_sig)) +
  geom_point(size = 1, stroke = 0) + xlab('1 nM log2 fold-change') +
  ylab('100 nM log2 fold-change') + ggtitle('U20S') + xlim(-3.75, 7.25) +
  ylim(-3.75, 7.25) + geom_vline(xintercept = 0, linetype = 'dashed') +
  geom_hline(yintercept = 0, linetype = 'dashed') + gg_options +
  scale_color_manual(values = c('#515151', '#46ACC8', '#E58601', '#B40F20'))

cor.test(CCRE_DESeq_results$U20S_wt_1vs0dex_log2,
         CCRE_DESeq_results$U20S_wt_100vs0dex_log2,
         method = 'spearman')

## Spearman's rank correlation rho
## data: CCRE_DESeq_results$U20S_wt_1vs0dex_log2 and CCRE_DESeq_results$U20S_wt_100vs0dex_log2
## S = 4.1688e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## 0.4400293

```

```
p10 <- grid.arrange(p8, p9, nrow = 1)
```



### Heatmap of significantly changed CCREs

```
sig <- unique(rbind(sig1, sig100))

sig <- sig[,c(14:17, 19:22)]
sig$category <- ''

for (i in 1:nrow(sig)){
  if (sig[i,4] < 0.05 & sig[i,8]<0.05){
    if (sig[i,3] > 0 & sig[i,7] > 0){
      sig[i,9] <- 'both_up'
    }
    else if (sig[i,3] < 0 & sig[i,7] < 0){
      sig[i,9] <- 'both_down'
    }
    else{
      sig[i,9] <- 'opposite'
    }
  }
  else if (sig[i,4] <0.05){
    if (sig[i,3] > 0) {
```

```

        sig[i,9] <- 'A549_up'
    }
    else {sig[i,9] <- 'A549_down'}
}
else{
  if (sig[i,7] > 0){
    sig[i,9] <- 'U20S_up'
  }
  else{sig[i,9] <- 'U20S_down'}
}
}
sig$category <- factor(sig$category,
                      levels = c('both_up' , 'A549_up' , 'U20S_up' ,
                                'both_down' , 'A549_down' , 'U20S_down' , 'opposite'))
table(sig$category)

##          both_up   A549_up   U20S_up both_down A549_down U20S_down opposite
##             94       82     1036      14       32      495        2

sig <- sig[order(sig$category),]

pheatmap(sig[,c(1,3,5,7)],
  show_rownames = F,
  color = rev(colorRampPalette(brewer.pal(n = 11, name = "RdBu"))(99)),
  breaks = c(seq(-2.7, -2, length.out = 20), seq(-1.9,-.5, length.out = 20),
            seq(-0.4,0.4,length.out = 20), seq(0.5, 1.9, length.out = 20),
            seq(2, 4.1, length.out = 20)),
  cluster_rows = F, cluster_cols = F)

```

