Code to generate Figure 1 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(dplyr)
library(plyr)
library(ggplot2)
library(knitr)
library(DESeq2)
library(RColorBrewer)
library(pheatmap)
library(gridExtra)
library(wesanderson)

coordinatesPath <- ’../Manuscript_data/’
bigwigPath <- ’../Manuscript_data/PRO-seq_bigwigs/Raw/’

#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 coordinate files for promoters and genes

Taking RefSeq coordinates for genes and parsing out: 1. Promoter coordinates 2. Gene coordinates for each unique TSS and TES (alternative splicing isoforms are ignored) 3. One set of coordinates for each gene, using the most internal TSS and TES, and removing the first and last 1 kb so as to remove sites of polymerase pausing

```
pathToUse='/home/emw97/GR_enhancers/Manuscript_data/'

#Downloaded from UCSC Table Browser
refseqFile=pathToUse+'hg38_refseq.txt'

#BED file that collapses down to unique TSS and TES sites
rsBED=pathToUse+'hg38_refseq.bed'

#BED file that has one of coordinates per gene,
#using most internal TSS and TES, removing 1 kb from 5' and 3' ends
rsBednr=pathToUse+'hg38_refseq_NR_1kb.bed'

promoterFile=pathToUse+'hg38_refseq_promoters.bed'

#amount upstream and downstream of annotated TSS to consider a promoter
lowerBound=1000
upperBound=1000


#creating list of chromosomes for humans
chromList=[]
for i in range(1,23):
    chromList.append('chr'+str(i))
chromList.append('chrX')
chromList.append('chrY')


rsIn=open(refseqFile,'r')
rsIn.readline()
```

```
## '#bin\tname\tchrom\tstrand\ttxStart\ttxEnd\tcdsStart\tcdsEnd\texonCount\texonStarts\texonEnds\tscore\
```

```
rsOut=open(rsBED,'w')
promOut=open(promoterFile,'w')

def inBED(bedList,currBed):
    #finds if current gene coordinates have already been recorded
    if currBed in bedList:
        toReturn='no'
    else:
        toReturn='yes'
        bedList.append(currBed)
    return toReturn,bedList

#Making dictionary of non-redundant promoters and bed file of refSeq information
nonredunProm={}
geneCoord=[]
nrGeneCoord={}
keepProm='no'
keepGene='no'


while 1:
    curr=rsIn.readline()
    if not curr:
        break
    curr=curr.rstrip().split(’\t’)
    if curr[2] in chromList and curr[1][:2]==’NM’:
        currBED=[curr[2],curr[4],curr[5],curr[12],’.’,curr[3]]
        toCheck=[currBED[0],currBED[1],currBED[2],currBED[5]]
        toWrite,geneCoord=inBED(geneCoord,toCheck)
        if toWrite==’yes’:
            rsOut.write(’\t’.join(currBED)+’\n’)

        if currBED[3] in nrGeneCoord.keys():
            if int(currBED[1]) > int(nrGeneCoord[currBED[3]][1]):
                nrGeneCoord[currBED[3]][1] = currBED[1]
            if int(currBED[2]) < int(nrGeneCoord[currBED[3]][2]):
                nrGeneCoord[currBED[3]][2] = currBED[2]
        else:
            nrGeneCoord[currBED[3]]=currBED

        if curr[3]==’+’:
            promStart=int(curr[4])-lowerBound
            promEnd=int(curr[4])+upperBound
        if curr[3]==’-’:
            promStart=int(curr[5])-upperBound
            promEnd=int(curr[5])+lowerBound
        if curr[12] not in nonredunProm:
            nonredunProm[curr[12]]=[[promStart,promEnd]]
            keep=’yes’
        else:
            if [promStart,promEnd] in nonredunProm[curr[12]]:
                keep=’no’
            else:
                nonredunProm[curr[12]].append([promStart,promEnd])
                keep=’yes’
        if keep==’yes’:
            promOutput=[curr[2],promStart,promEnd,curr[12],’.’,curr[3]]
            promOut.write(’\t’.join(map(str,promOutput))+’\n’)
rsOut.close()
rsIn.close()
promOut.close()

rsNRout=open(rsBednr, ’w’)
for i in nrGeneCoord.keys():
    output=nrGeneCoord[i]
    output[1]=str(int(output[1])+lowerBound)
    output[2]=str(int(output[2])+upperBound)
    rsNRout.write(’\t’.join(output) + ’\n’)
rsNRout.close()
```

Finding read counts for each gene from raw bigwigs, using the bigWig package.

```
#read in coordinates
gene_data <- read.table(paste0(coordinatesPath,’hg38_refseq_NR_1kb.bed’),sep=’\t’)
colnames(gene_data) <- c(’chrom’,’start’,’end’,’name’,’’,’strand’)

gene_data <- gene_data[gene_data$end>gene_data$start,]

#creating columns in dataframe for data
for (i in 1:length(samples)){
  gene_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 = ""))
  gene_data[i+6] <- bed6.region.bpQuery.bigWig(bw.plus = pbw, bw.minus = mbw,
      bed6 = gene_data[1:6], op = ’sum’, abs.value = TRUE)
}


gene_data$length <- gene_data$end-gene_data$start+1
gene_data_nr <- gene_data[,c(4,7:19)]
gene_data_nr <- ddply(gene_data_nr,"name",numcolwise(sum))
write.table(gene_data_nr,"GR_gene_counts.txt",quote = FALSE,sep = ’\t’)
```

Running DESeq2 to find differentially expressed genes

```
row.names(gene_data_nr) <- gene_data_nr$name
geneDESeq <- gene_data_nr[2:13]

DESeq_table <- data.frame(row.names = colnames(geneDESeq))
DESeq_table$cellType <- factor(c(rep(’A549’,6),rep(’U2OS’,6)))
DESeq_table$treatment <- factor(rep(c(’1’,’1’,’0’,’0’,’100’,’100’),2))
DESeq_table$group <- factor(paste0(DESeq_table$cellType,DESeq_table$treatment))

dds <- DESeqDataSetFromMatrix (countData= geneDESeq,
                               colData = DESeq_table, design= ~group)
dds <- estimateSizeFactors(dds)
idx <- rowSums( counts(dds, normalized=TRUE) >= 75 ) >= 3

dds <- dds[idx,]
dds <- DESeq(dds)

A549_wt_1vs0dex <- lfcShrink(dds, contrast = c(’group’, ’A5491’, ’A5490’))
A549_wt_100vs0dex <- lfcShrink(dds, contrast = c(’group’, ’A549100’, ’A5490’))
U2OS_wt_1vs0dex <- lfcShrink(dds, contrast = c(’group’, ’U2OS1’, ’U2OS0’))
U2OS_wt_100vs0dex <- lfcShrink(dds, contrast = c(’group’, ’U2OS100’, ’U2OS0’))

vsd <- vst(dds, blind = FALSE)
```

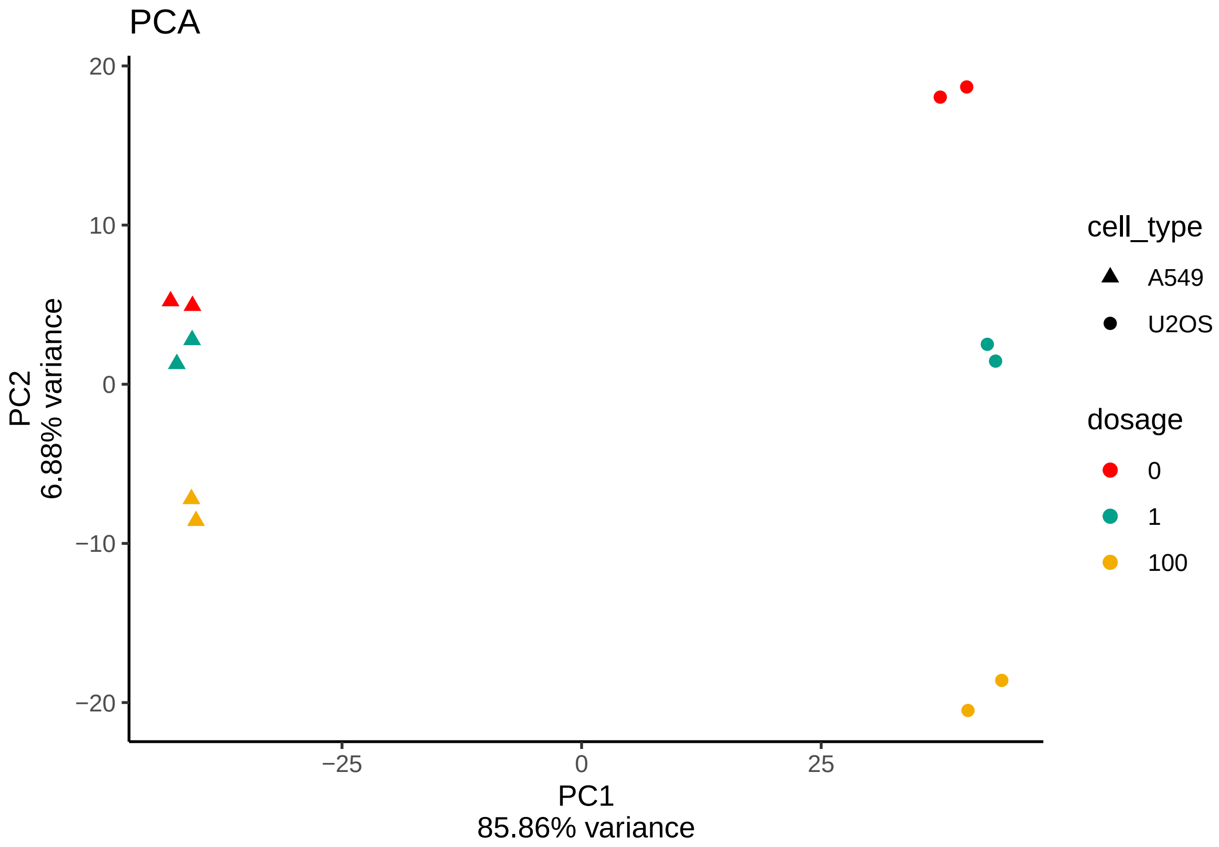
```
gene_DESeq <- as.data.frame(assay(vsd))

gene_DESeq$baseMean <- A549_wt_1vs0dex$baseMean
gene_DESeq$A549_wt_1vs0dex_log2 <- A549_wt_1vs0dex$log2FoldChange
gene_DESeq$A549_wt_1vs0dex_fdr <- A549_wt_1vs0dex$padj

gene_DESeq$A549_wt_100vs0dex_log2 <- A549_wt_100vs0dex$log2FoldChange
gene_DESeq$A549_wt_100vs0dex_fdr <- A549_wt_100vs0dex$padj

gene_DESeq$U2OS_wt_1vs0dex_log2 <- U2OS_wt_1vs0dex$log2FoldChange
gene_DESeq$U2OS_wt_1vs0dex_fdr <- U2OS_wt_1vs0dex$padj

gene_DESeq$U2OS_wt_100vs0dex_log2 <- U2OS_wt_100vs0dex$log2FoldChange
gene_DESeq$U2OS_wt_100vs0dex_fdr <- U2OS_wt_100vs0dex$padj

gene_DESeq$length <- gene_data_nr[idx, 19]

write.table(gene_DESeq, ’gene_DESeq_analysis.txt’, quote = F, sep = ’\t’)
```

PCA in Supp. Fig. 1

```
 #PCA

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

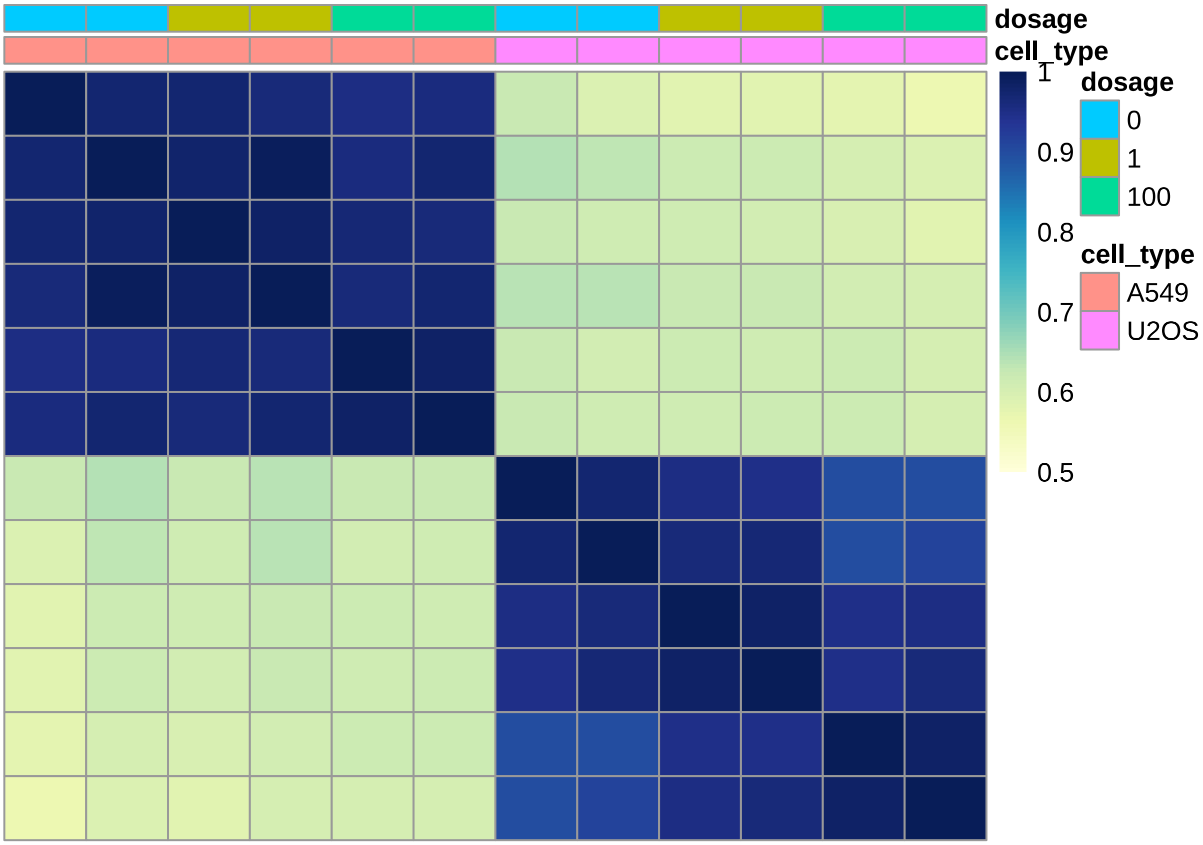
```
## Importance of components:
##                            PC1      PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     43.0869 12.19383 6.88815 5.85922 4.84089 3.95814 3.76157
## Proportion of Variance  0.8586  0.06877 0.02194 0.01588 0.01084 0.00725 0.00654
## Cumulative Proportion   0.8586  0.92734 0.94929 0.96516 0.97600 0.98325 0.98979
##                            PC8     PC9   PC10    PC11      PC12
## Standard deviation     2.72541 2.49785 2.3251 1.73319 7.202e-14
## Proportion of Variance 0.00344 0.00289 0.0025 0.00139 0.000e+00
## Cumulative Proportion  0.99323 0.99611 0.9986 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\n85.86% variance’) + ylab(’PC2\n6.88% variance’) +
  scale_shape_manual(values=c(17,16)) +
  scale_color_manual(values = wes_palette(n=3, name = "Darjeeling1")) + gg_options
```

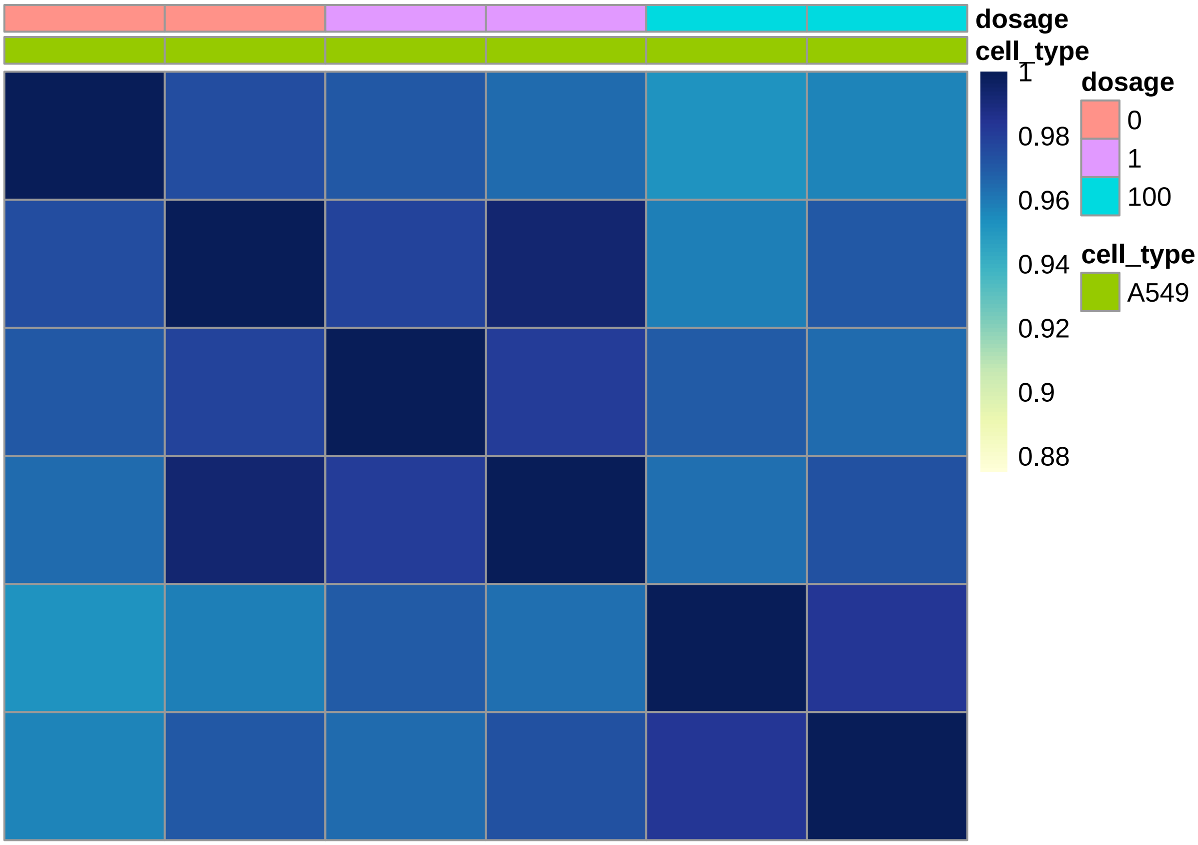


Spearman correlation plots in Supp. Fig. 1

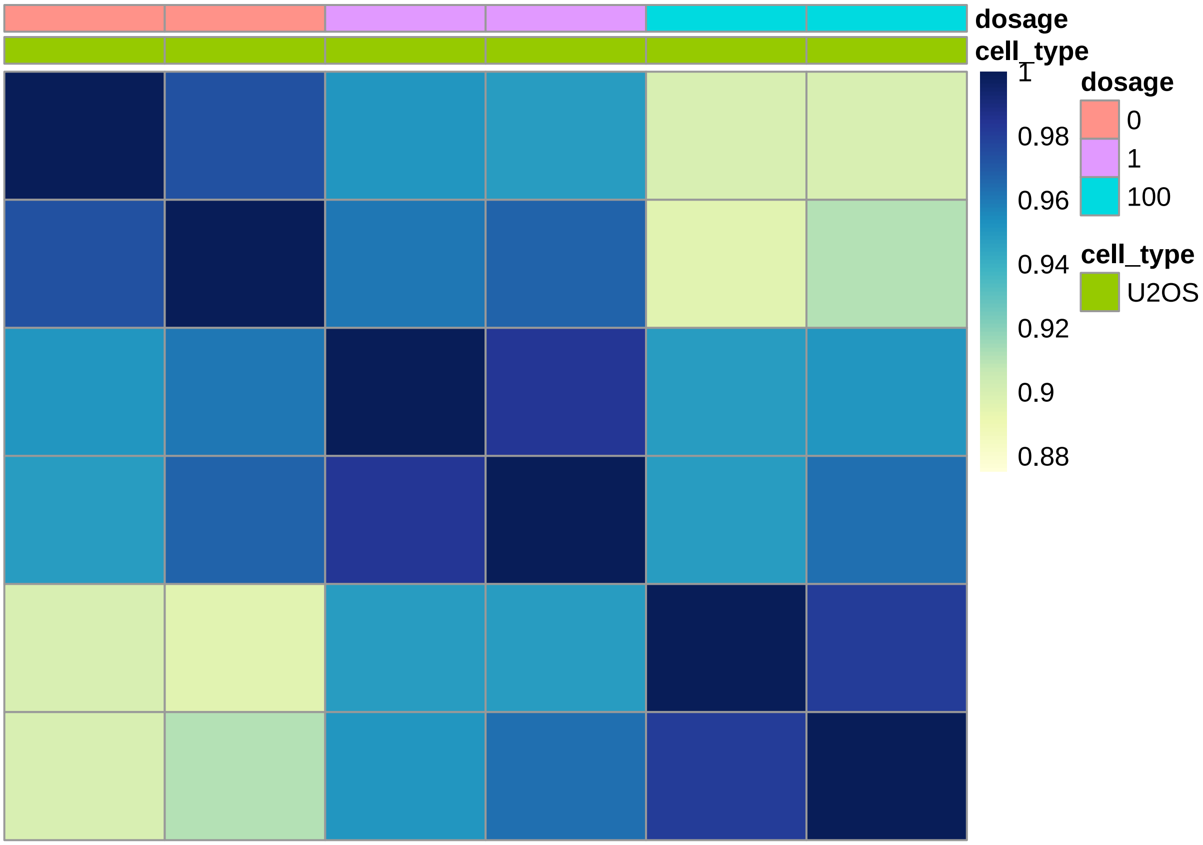
```
# correlation matrix
corr_matrix <- as.data.frame(matrix(nrow = 12, ncol = 12))
rownames(corr_matrix) <- rownames(pca_values)
colnames(corr_matrix) <- rownames(pca_values)


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(gene_DESeq[,toUse[i]], gene_DESeq[,toUse[j]],
                                 method = 'spearman')$estimate
  }
}
pheatmap(corr_matrix,
          color = colorRampPalette(brewer.pal(n = 9, name = "YlGnBu")) (100),
          breaks = seq(0.5, 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(0.875, 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)
```

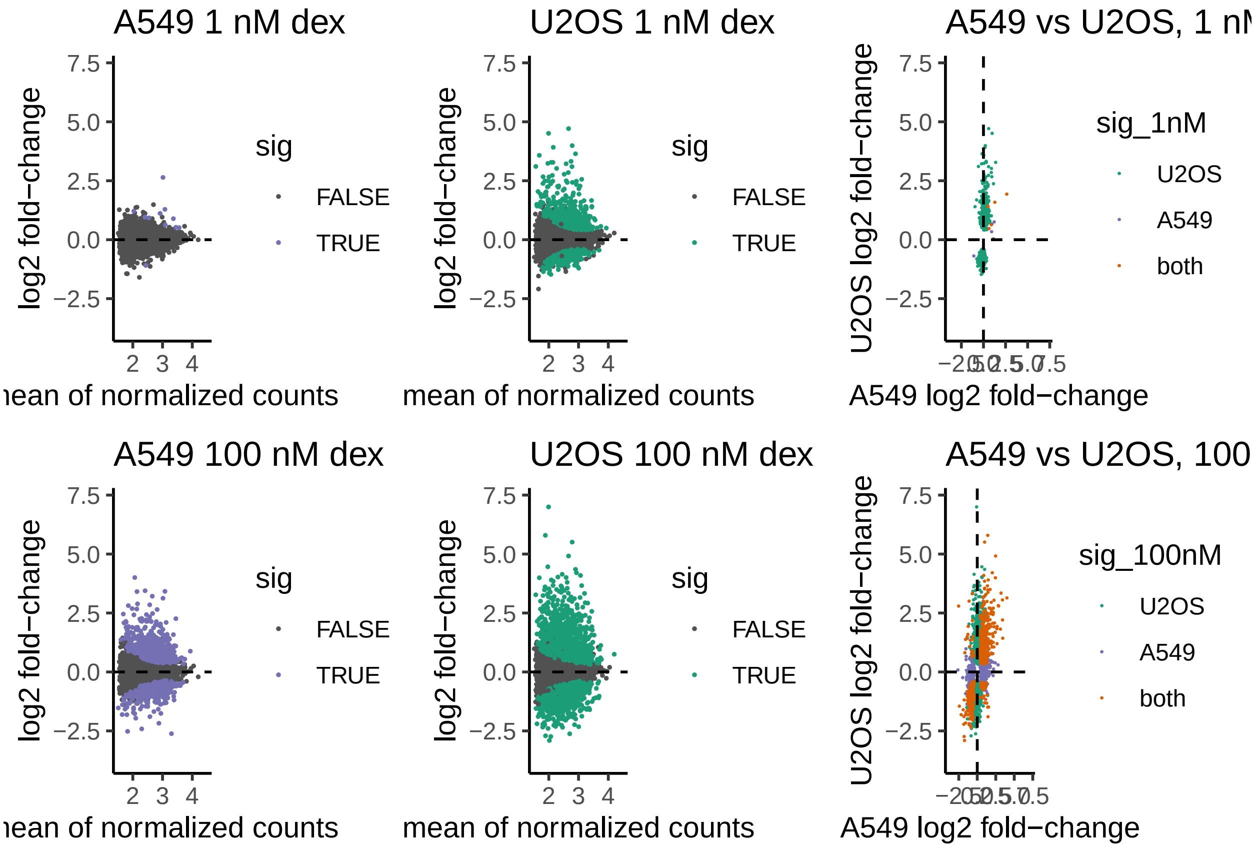


```
pheatmap(corr_matrix[7:12,7:12],
          color = colorRampPalette(brewer.pal(n = 9, name = "YlGnBu")) (100),
         breaks = seq(0.875, 1.0, length.out = 101),
          cluster_rows = F, cluster_cols = F,
          annotation_col = pca_values[7:12,13:14],
          show_rownames = F, show_colnames = F)
```



MA plots and comparison plot for Fig. 1 C-E

```
gene_DESeq[is.na(gene_DESeq)] <- 1
gene_DESeq$sig_1nM <- ’’
gene_DESeq$sig_100nM <- ’’

for (i in 1:nrow(gene_DESeq)){
  if (gene_DESeq[i,15] < 0.05 &  gene_DESeq[i,19] < 0.05){
    gene_DESeq[i,22] <- ’both’
  } else if (gene_DESeq[i,15] < 0.05){
    gene_DESeq[i,22] <- ’A549’
  } else if (gene_DESeq[i,19] < 0.05){
    gene_DESeq[i,22] <- ’U2OS’
  } else {
    gene_DESeq[i,22] <- ’none’
  }

  if (gene_DESeq[i,17] < 0.05 &  gene_DESeq[i,21] < 0.05){
    gene_DESeq[i,23] <- ’both’
  } else if (gene_DESeq[i,17] < 0.05){
    gene_DESeq[i,23] <- ’A549’
  } else if (gene_DESeq[i,21] < 0.05){
    gene_DESeq[i,23] <- ’U2OS’
  } else {
    gene_DESeq[i,23] <- ’none’
  }
}
gene_DESeq$sig_1nM <- factor(gene_DESeq$sig_1nM,
                             levels = c("none", "U2OS", "A549", "both"))
gene_DESeq$sig_100nM <- factor(gene_DESeq$sig_100nM,
                               levels = c("none", "U2OS","A549", "both"))

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

p1 <- ggplot(gene_DESeq, aes(x = log10(baseMean),
  y = A549_wt_1vs0dex_log2,
  color = A549_wt_1vs0dex_fdr < 0.05)) +
  geom_point(size = 0.75, stroke = 0) +
  xlim(1.5,4.5) + ylim(-3.75, 7.25) +
  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(gene_DESeq, aes(x = log10(baseMean),
  y = U2OS_wt_1vs0dex_log2,
  color = U2OS_wt_1vs0dex_fdr < 0.05)) +
  geom_point(size = 0.75, stroke = 0) +
  xlim(1.5,4.5) + ylim(-3.75, 7.25) +
  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

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

p3 <- ggplot(gene_DESeq, aes(x = log10(baseMean),
  y = A549_wt_100vs0dex_log2,
  color = A549_wt_100vs0dex_fdr < 0.05)) +
  geom_point(size = 0.75, stroke = 0) +
  xlim(1.5,4.5) + ylim(-3.75, 7.25) +
  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(gene_DESeq, aes(x = log10(baseMean),
  y = U2OS_wt_100vs0dex_log2,
  color = U2OS_wt_100vs0dex_fdr < 0.05)) +
  geom_point(size = 0.75, stroke = 0) +
  xlim(1.5,4.5) + ylim(-3.75, 7.25) +
  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 <- gene_DESeq[gene_DESeq$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(-3.75, 7.25) + ylim(-3.75, 7.25) +
  geom_vline(xintercept = 0, linetype = 'dashed') +
  geom_hline(yintercept = 0, linetype = 'dashed') + gg_options

sig100 <- gene_DESeq[gene_DESeq$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(-3.75, 7.25) + ylim(-3.75, 7.25) +
  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)
```



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

```
##
##  Spearman's rank correlation rho
##
## data:  gene_DESeq$A549_wt_1vs0dex_log2 and gene_DESeq$U2OS_wt_1vs0dex_log2
## S = 1.582e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.3576217
```

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

```
##
##  Spearman's rank correlation rho
##
## data:  gene_DESeq$A549_wt_100vs0dex_log2 and gene_DESeq$U2OS_wt_100vs0dex_log2
## S = 1.7045e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##      rho
## 0.307904
```

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

```
## [1] "Number of signficantly upregulated genes in A549 after 1 nM dex"
## [2] "10"
```

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

```
## [1] "Number of signficantly downregulated genes in A549 after 1 nM dex"
## [2] "1"
```

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

```
## [1] "Number of signficantly upregulated genes in A549 after 100 nM dex"
## [2] "515"
```

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

```
## [1] "Number of signficantly downregulated genes in A549 after 100 nM dex"
## [2] "312"
```

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

```
## [1] "Number of signficantly upregulated genes in U2OS after 1 nM dex"
## [2] "413"
```

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

```
## [1] "Number of signficantly downregulated genes in U2OS after 1 nM dex"
## [2] "250"
```

```r
print (c(’Number of signficantly upregulated genes in U2OS after 100 nM dex’,
         nrow(gene_DESeq[gene_DESeq$U2OS_wt_100vs0dex_fdr<0.05 &
                           gene_DESeq$U2OS_wt_100vs0dex_log2>0,])))
```

```
## [1] "Number of signficantly upregulated genes in U2OS after 100 nM dex"
## [2] "1193"
```

```r
print (c(’Number of signficantly downregulated genes in U2OS after 100 nM dex’,
         nrow(gene_DESeq[gene_DESeq$U2OS_wt_100vs0dex_fdr<0.05 &
                           gene_DESeq$U2OS_wt_100vs0dex_log2<0,])))
```

```
## [1] "Number of signficantly downregulated genes in U2OS after 100 nM dex"
## [2] "1264"
```

Comparing 1 nM to 100 nM treatment per cell type

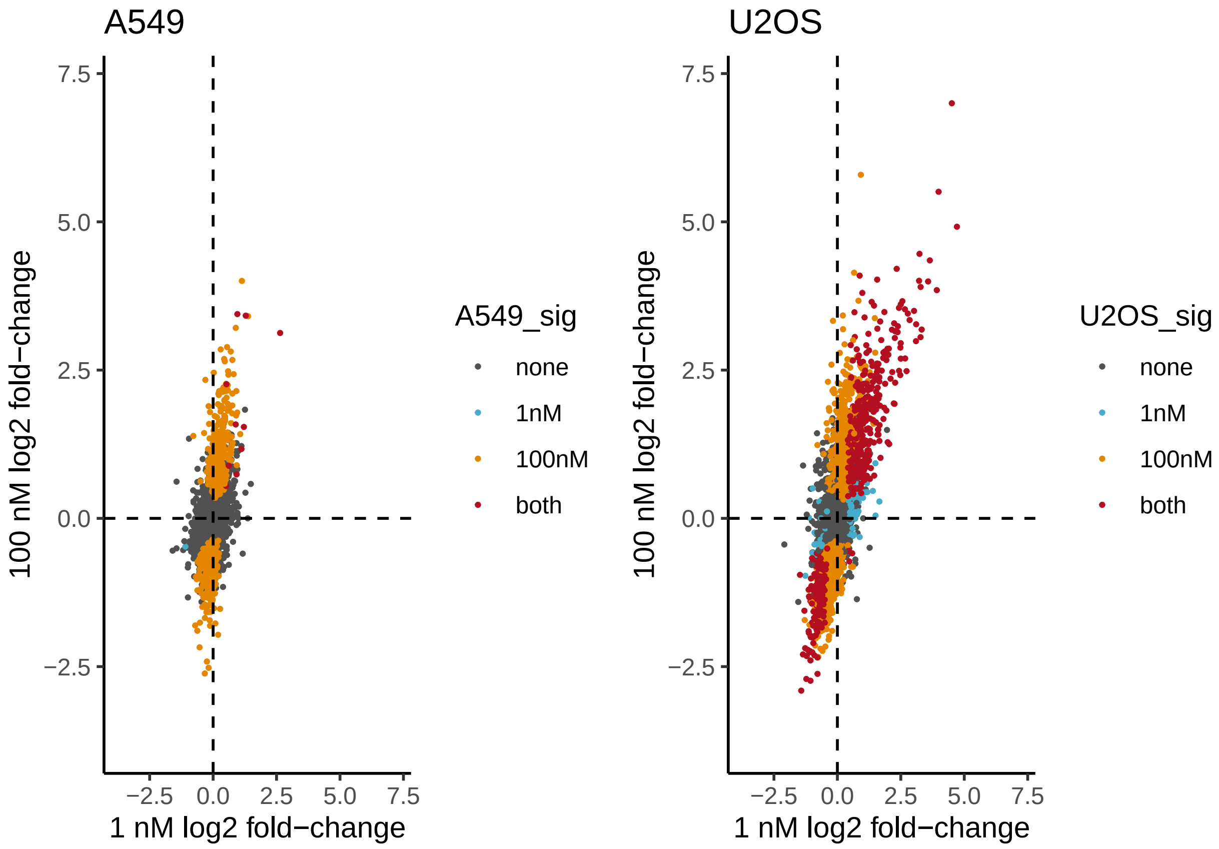
```r
gene_DESeq$A549_sig <- ’’
gene_DESeq$U2OS_sig <- ’’

for (i in 1:nrow(gene_DESeq)){
  if (gene_DESeq[i,15] < 0.05 &  gene_DESeq[i,17] < 0.05){
    gene_DESeq[i,24] <- ’both’
  } else if (gene_DESeq[i, 15] < 0.05){
    gene_DESeq[i,24] <- ’1nM’
  } else if (gene_DESeq[i, 17] < 0.05){
    gene_DESeq[i,24] <- ’100nM’
  } else{
    gene_DESeq[i,24] <- ’none’
  }

  if (gene_DESeq[i,19] < 0.05 &  gene_DESeq[i,21] < 0.05){
    gene_DESeq[i,25] <- ’both’
  } else if (gene_DESeq[i, 19] < 0.05){
    gene_DESeq[i,25] <- ’1nM’
  } else if (gene_DESeq[i, 21] < 0.05){
    gene_DESeq[i,25] <- ’100nM’
  } else{
    gene_DESeq[i,25] <- ’none’
  }
}


gene_DESeq$A549_sig <- factor(gene_DESeq$A549_sig,
                              levels = c("none", "1nM", "100nM", "both"))
gene_DESeq$U2OS_sig <- factor(gene_DESeq$U2OS_sig,
                              levels = c("none", "1nM", "100nM", "both"))



p8 <- ggplot(gene_DESeq, 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', '#46ACC8', '#E58601', '#B40F20'))

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

```
##
##  Spearman's rank correlation rho
##
## data:  gene_DESeq$A549_wt_1vs0dex_log2 and gene_DESeq$A549_wt_100vs0dex_log2
## S = 1.2326e+11, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.4994987
```

```
p9 <- ggplot(gene_DESeq, aes(x = U2OS_wt_1vs0dex_log2,
  y = U2OS_wt_100vs0dex_log2, color = U2OS_sig)) +
  geom_point(size = 1, stroke = 0) +
  xlab('1 nM log2 fold-change') + ylab('100 nM log2 fold-change') +
  ggtitle('U2OS') + 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(gene_DESeq$U2OS_wt_1vs0dex_log2, gene_DESeq$U2OS_wt_100vs0dex_log2, method = 'spearman')
```

```
##
##  Spearman's rank correlation rho
##
## data:  gene_DESeq$U2OS_wt_1vs0dex_log2 and gene_DESeq$U2OS_wt_100vs0dex_log2
## S = 7.262e+10, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.7051261
```

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

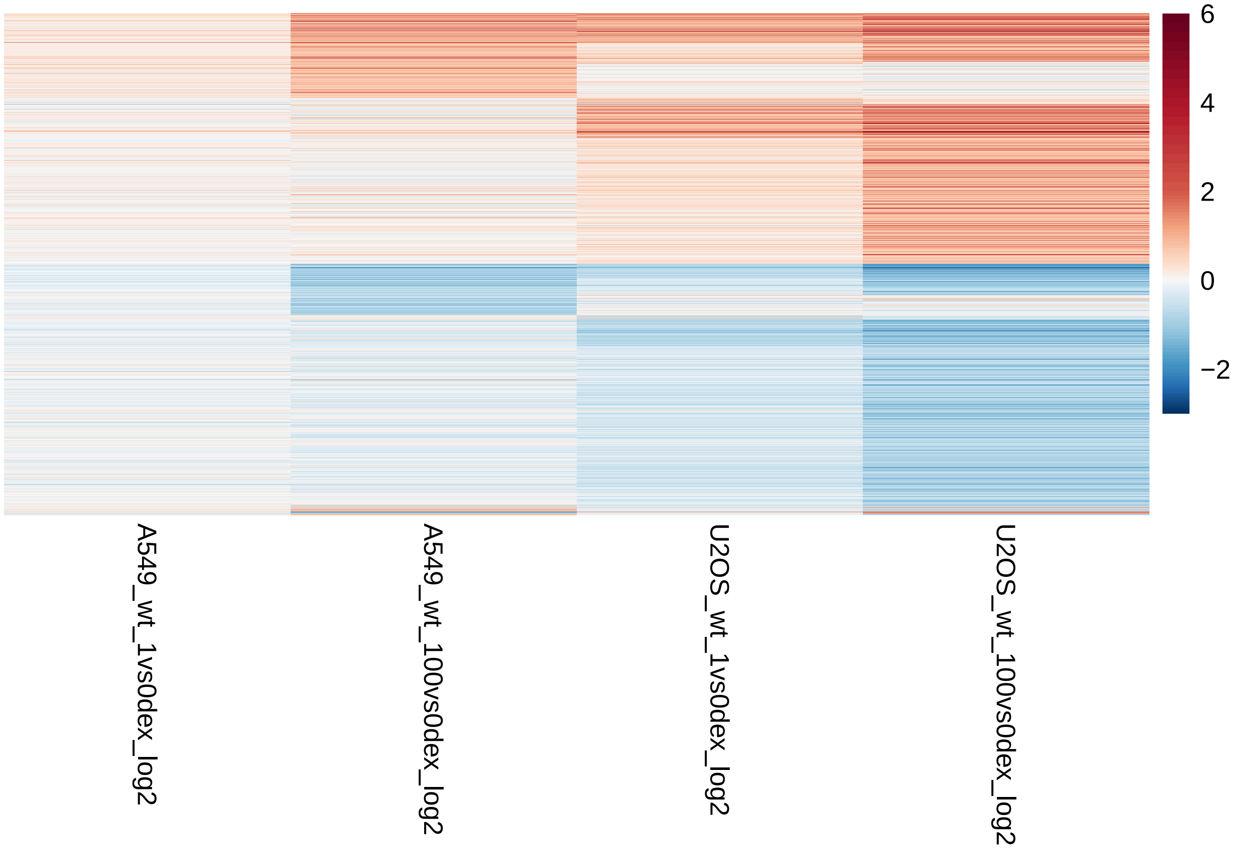


Heatmap of significantly changed genes

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

sig <- sig[,14:21]
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] <- 'U2OS_up'
    }
    else{sig[i,9] <- 'U2OS_down'}
  }
}
sig$category <- factor(sig$category, levels =
                c('both_up' , 'A549_up' , 'U2OS_up' ,
                  'both_down' , 'A549_down' , 'U2OS_down', 'opposite'))
table(sig$category)
```

```
##
##   both_up   A549_up   U2OS_up both_down A549_down U2OS_down  opposite
##       272       209       935       178       113      1076        55
```

```
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(-3.0, -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, 6, length.out = 20)),
      cluster_rows = F, cluster_cols = F)
```


6
4
2
0
−2
A549_wt_1vs0dex_log2
A549_wt_100vs0dex_log2
U2OS_wt_1vs0dex_log2
U2OS_wt_100vs0dex_log2