Code to generate Figure 2 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(DESeq2)

genesFile <- 'gene_DESeq_analysis.txt'

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()
```

Code used in 2A to make heatmaps of ChIP-seq binding data, using Deeptools

Making heatmaps with Deeptools of the overlapping and distinct binding sites of GR in A549 and U2OS cells.

```
#Taking GR ChIP-seq summits and creating files of +- 25 bp windows surrounding the summits
csPath='../Manuscript_data/'
coordPath='../Manuscript_data/'

cat ${csPath}GR_summits_A549.bed | awk 'BEGIN{OFS="\t"}{print $1,$2 -25,$3 +24}' >
GR_summits_A549_50bp.bed

cat ${csPath}GR_summits_U2OS-hGR.bed | awk 'BEGIN{OFS="\t"}{print $1,$2 -25,$3 +24}' >
GR_summits_U2OS-hGR_50bp.bed

#Writing files of GR ChIP-seq peaks that are shared or cell type-specific
intersectBed -wa -a GR_summits_A549_50bp.bed \
-b GR_summits_U2OS-hGR_50bp.bed > A549_U2OS_GR_overlap.bed

intersectBed -wa -v -a GR_summits_A549_50bp.bed \
-b GR_summits_U2OS-hGR_50bp.bed > A549_GR_unique.bed

intersectBed -wa -v -a GR_summits_U2OS-hGR_50bp.bed \
-b GR_summits_A549_50bp.bed > U2OS_GR_unique.bed

#With Deeptools, plotting signal of GR ChIP-seq and ATAC-seq data,
#+- 500 bp from GR peaks, in A549 and U2OS cells,
#at GR peaks that are shared or cell type-specific.
#Bigwigs for GR came from GSE163398.
#ATAC-seq data was re-aligned using the ENCODE ATAC-seq pipeline
#(https://github.com/ENCODE-DCC/atac-seq-pipeline)
#using fastqs from ENCSR220ASC (A549) and GSE109589 (U2OS)

computeMatrix reference-point
  -R A549_U2OS_GR_overlap.bed A549_GR_unique.bed U2OS_GR_unique.bed
  -S GR_ChIPseq_A549_0nMdex.bw GR_ChIPseq_A549_100nMdex.bw A549_ATAC_EtOH.pval.bw
  GR_ChIPseq_U2OS-hGR_0nMdex.bw GR_ChIPseq_U2OS-hGR_100nMdex.bw U2OS_ATAC_EtOH.pval.bw
  -out A549_U2OS_GR_ATAC_3categories_1kb.computeMatrix.gz
  --referencePoint center --upstream 500 --downstream 500 --missingDataAsZero

plotHeatmap --matrixFile A549_U2OS_GR_ATAC_3categories_1kb.computeMatrix.gz
  --outFileName A549_U2OS_GR_ATAC_3categories_1kb.heatmap.pdf
  --xAxisLabel 'GORs' --refPointLabel 'center'
  --colorMap Reds Reds Blues Reds Reds Blues
  --zMin 0 --zMax 100 100 50 100 100 50 --plotType std
```

Code used in 2B to run HOMER for finding motif enrichment at shared and cell type-specific GORs

```
# Creating fasta files to use in HOMER that are +-150 bp from the GR summit.
#The hg38.fa file is too large for Github but came from
#https://hgdownload.cse.ucsc.edu/goldenpath/hg38/bigZips/hg38.fa.gz

# All GORs in A549 cells
cat ${csPath}GR_summits_A549.bed |
  awk 'BEGIN{OFS="\t"}{print $1,$2 -150,$3 +149}' |
  grep -v chrM | grep -v chrEBV | sed '/_/d' |
  fastaFromBed -fi hg38.fa -bed stdin -fo A549_GORs.fa

# All GORs in U2OS cells
cat ${csPath}GR_summits_U2OS-hGR.bed |
  awk 'BEGIN{OFS="\t"}{print $1,$2 -150,$3 +149}' |
  grep -v chrM | grep -v chrEBV | sed '/_/d' |
  fastaFromBed -fi hg38.fa -bed stdin -fo U2OS_GORs.fa

# GORs shared in A549 and U2OS cells
cat A549_U2OS_GR_overlap.bed |
  awk 'BEGIN{OFS="\t"}{print $1,$2 -125,$3 +124}' |
  grep -v chrM | grep -v chrEBV | sed '/_/d' |
  fastaFromBed -fi hg38.fa -bed stdin -fo A549_U2OS_GR_overlap.fa

# GORs only in A549 cells
cat A549_GR_unique.bed |
  awk 'BEGIN{OFS="\t"}{print $1,$2 -125,$3 +124}' |
  grep -v chrM | grep -v chrEBV | sed '/_/d' |
  fastaFromBed -fi hg38.fa -bed stdin -fo A549_GR_unique.fa

# GORs only in U2OS cells
cat U2OS_GR_unique.bed |
  awk 'BEGIN{OFS="\t"}{print $1,$2 -125,$3 +124}' |
  grep -v chrM | grep -v chrEBV | sed '/_/d' |
  fastaFromBed -fi hg38.fa -bed stdin -fo U2OS_GR_unique.fa

# GORs only in A549 or U2OS cells, but not in both cell types
cat A549_GR_unique.bed U2OS_GR_unique.bed |
  awk 'BEGIN{OFS="\t"}{print $1,$2 -125,$3 +124}' |
  grep -v chrM | grep -v chrEBV | sed '/_/d' |
  fastaFromBed -fi hg38.fa -bed stdin -fo A549_U2OS_distinct_overlap.fa

# Running HOMER

hoco='../Manuscript_data/HOCOMOCOv11_core_HUMAN_mono_homer_format_0.001.motif'

# Finding motifs specific to A549 GORs
findMotifs.pl A549_GORs.fa fasta A549_GORs_vs_U2OS/
  -len 8,10,12 -fastaBg U2OS_GORs.fa
  -mcheck $hoco -bits -nogo -mknown $hoco

# Finding motifs specific to U2OS GORs
findMotifs.pl U2OS_GORs.fa fasta U2OS_GORs_vs_A549/
  -len 8,10,12 -fastaBg A549_GORs.fa
  -mcheck $hoco -bits -nogo -mknown $hoco

# Finding motifs specific to shared GORs
findMotifs.pl A549_U2OS_GR_overlap.fa fasta
  overlapping_vs_unique_GORS/
  -len 8,10,12
  -fastaBg A549_U2OS_distinct_overlap.fa
  -mcheck $hoco -bits -nogo -mknown $hoco
```

Finding expression levels of pioneer factors

```
EtOH_genes <- read.table('GR_gene_counts.txt', sep = '\t')[,c(1,4,5,10,11)]
row.names(EtOH_genes) <- EtOH_genes$name
EtOH_genes <- EtOH_genes[2:5]

DESeq_table <- data.frame(row.names = colnames(EtOH_genes))
DESeq_table$cellType <- factor(c('A549', 'A549', 'U2OS', 'U2OS'))

dds <- DESeqDataSetFromMatrix (countData= EtOH_genes,
                               colData = DESeq_table, design= ~cellType)
dds <- estimateSizeFactors(dds)
idx <- rowSums( counts(dds, normalized=TRUE) >= 100 ) >= 2

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

EtOH_diff <- lfcShrink(dds, coef = 'cellType_U2OS_vs_A549')

EtOH_diff_DEseq <- as.data.frame(assay(vst(dds, blind = FALSE)))
EtOH_diff_DEseq$log2 <- EtOH_diff$log2FoldChange
EtOH_diff_DEseq$padj <- EtOH_diff$padj

genes_of_interest <- c(
  'FOXA1','FOXA2','CEBPA','CEBPB','RUNX1','RUNX2',
  'NR3C1','FOS','FOSL1','FOSL2','JUN','JUNB')
EtOH_diff_DEseq[genes_of_interest, 5:6]
```

```
##              log2         padj
## FOXA1 -4.6484210 1.054854e-37
## FOXA2 -3.0651540 1.670134e-23
## CEBPA -2.9754384 6.339162e-28
## CEBPB -1.1932576 2.737302e-06
## RUNX1  0.8562571 1.198145e-07
## RUNX2  2.4861709 3.896162e-32
## NR3C1 -1.1105145 4.151186e-10
## FOS    0.7493683 5.931906e-02
## FOSL1  0.9151102 8.624001e-04
## FOSL2 -0.9578456 1.658272e-09
## JUN    0.5587163 3.311179e-01
## JUNB  -0.6966183 9.194504e-03
```

Code used in 2C to run Deeptools for making metaplots of TF occupancy at shared and cell type-specific GORs.

ChIP-seq data was re-aligned using the ENCODE ChIP-seq pipeline (https://github.com/ENCODE-DCC/chip-seq-pipeline) using fastqs from GSE90454 (FOXA2), ENCSR701TCU (CEBPB), ENCSR656VWZ (JUNB), ENCSR593DGU (FOSL2), and ENCSR192PBJ (JUN). Bigwigs for GR came from GSE163398. ATAC-seq data was re-aligned using the ENCODE ATAC-seq pipeline (https://github.com/ENCODE-DCC/atac-seq-pipeline) using fastqs from ENCSR220ASC (A549).

```
computeMatrix reference-point
  -R A549_U2OS_GR_overlap.bed A549_GR_unique.bed U2OS_GR_unique.bed
  -S GR_ChIPseq_A549_0nMdex.bw GR_ChIPseq_A549_100nMdex.bw A549_ATAC_EtOH.pval.bw
  CEBPB.pval.bw FOXA2.pval.bw FOSL2.pval.bw  JUN.pval.bw JUNB.pval.bw
  GR_ChIPseq_U2OS-hGR_0nMdex.bw GR_ChIPseq_U2OS-hGR_100nMdex.bw U2OS_ATAC_EtOH.pval.bw
  -out A549_U2OS_GR_ATAC_3categories_ChIPdata_1kb.computeMatrix.gz
  --referencePoint center --upstream 500 --downstream 500 --missingDataAsZero

plotProfile
  --matrixFile A549_U2OS_GR_ATAC_3categories_ChIPdata_1kb.computeMatrix.gz
  --outFileName A549_U2OS_GR_ATAC_3categories_ChIPdata_1kb.metaplot.pdf
  --refPointLabel 'center'
  --yMin 0 --yMax 100 100 75 50 150 25 125 125 150 150 75
  --plotType std --colors Blue Green Magenta
```

Code used in 2D to find log odds scores of GBS's

Using Homer to find all GBS scores

```
GCR='../Manuscript_data/GCR_HUMAN.H11MO.0.A.motif'

findMotifs.pl A549_U2OS_GR_overlap.fa fasta test/ -find $GCR > overlapping_GORS_GCR_scores.txt

findMotifs.pl A549_GR_unique.fa fasta test/ -find $GCR > overlapping_GORS_GCR_scores_A549.txt

findMotifs.pl U2OS_GR_unique.fa fasta test/ -find $GCR > overlapping_GORS_GCR_scores_U2OS.txt
```

Finding the GBS with the highest score in each GOR and number of GBSs in each GOR

```
def BestMatch(sample):
  dataIn=open(sample+'.txt').readlines()[1:]
  dataOut=open(sample+'_highestScore.txt', 'w')
  bestMatch={}
  for i in dataIn:
    curr=i.split('\t')
    name=curr[0]
    if name not in bestMatch.keys():
      bestMatch[name]=[float(curr[5]), float(curr[5]), 1]
    else:
      bestMatch[name][1]=float(curr[5]) + bestMatch[name][1]
      if float(curr[5]) > bestMatch[name]:
        bestMatch[name][0]=float(curr[5])
      bestMatch[name][2]=bestMatch[name][2]+1

  coords=sorted(bestMatch.keys())

  for i in coords:
    dataOut.write(i + '\t' + str(bestMatch[i][0]) + '\t' +
    str(bestMatch[i][1]) + '\t' + str(bestMatch[i][2]) + '\n')
  dataOut.close()

samples=['overlapping_GORS_GCR_scores', 'overlapping_GORS_GCR_scores_A549', \
'overlapping_GORS_GCR_scores_U2OS']

for j in samples:
  BestMatch(j)
```

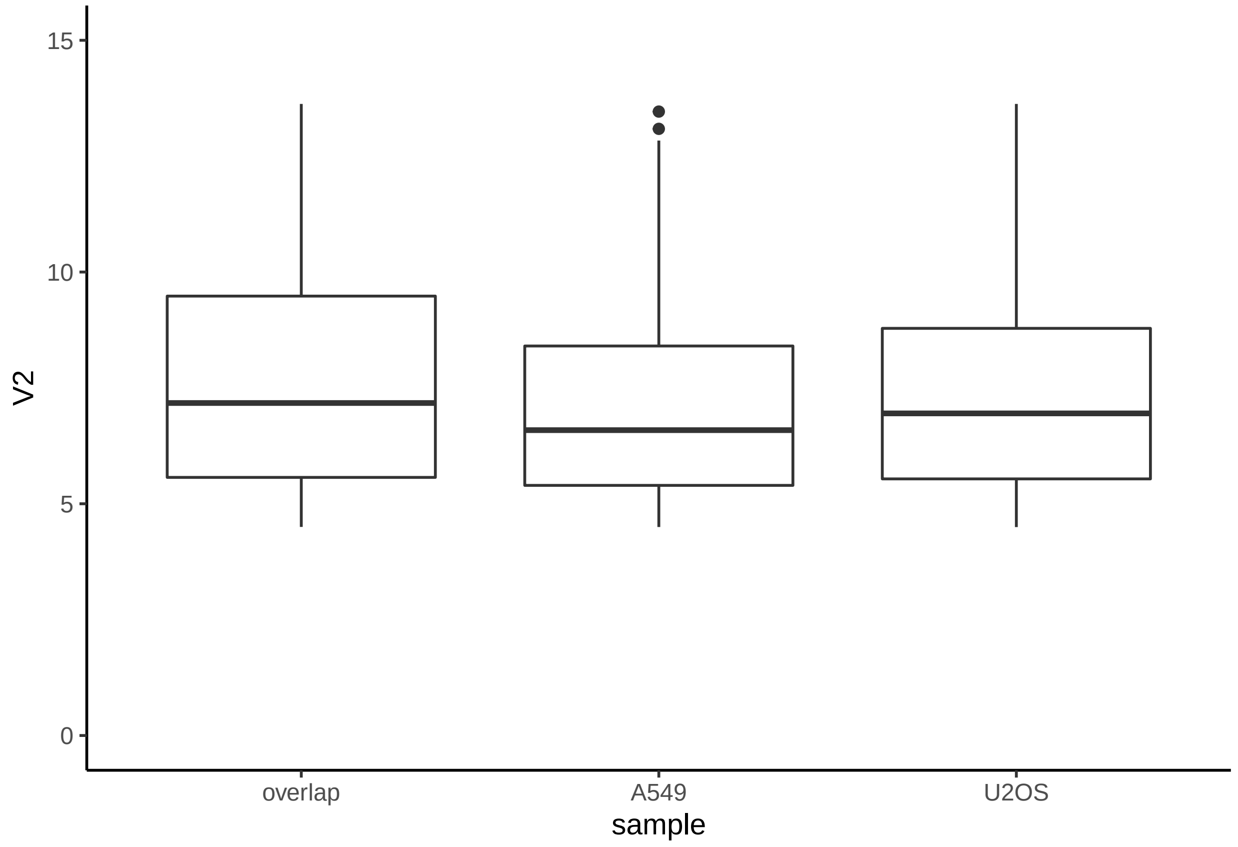
Plotting GBS's with the highest scores and number of GBS per GOR

```
overlap <- read.table('overlapping_GORS_GCR_scores_highestScore.txt', sep = '\t')
overlap$sample <- 'overlap'
A549 <- read.table('overlapping_GORS_GCR_scores_A549_highestScore.txt', sep = '\t')
A549$sample <- 'A549'
U2OS <- read.table('overlapping_GORS_GCR_scores_U2OS_highestScore.txt', sep = '\t')
U2OS$sample <- 'U2OS'

toPlot <- rbind.data.frame(overlap, A549, U2OS)
toPlot$sample <- factor(toPlot$sample,
```

```r
                        levels = c('overlap', 'A549', 'U2OS'))

ggplot(data = toPlot, mapping = aes(x = sample, y = V2)) +
  geom_boxplot() + gg_options + ylim(c(0,15))
```



```r
wilcox.test(overlap[,2], A549[,2])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  overlap[, 2] and A549[, 2]
## W = 6089866, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

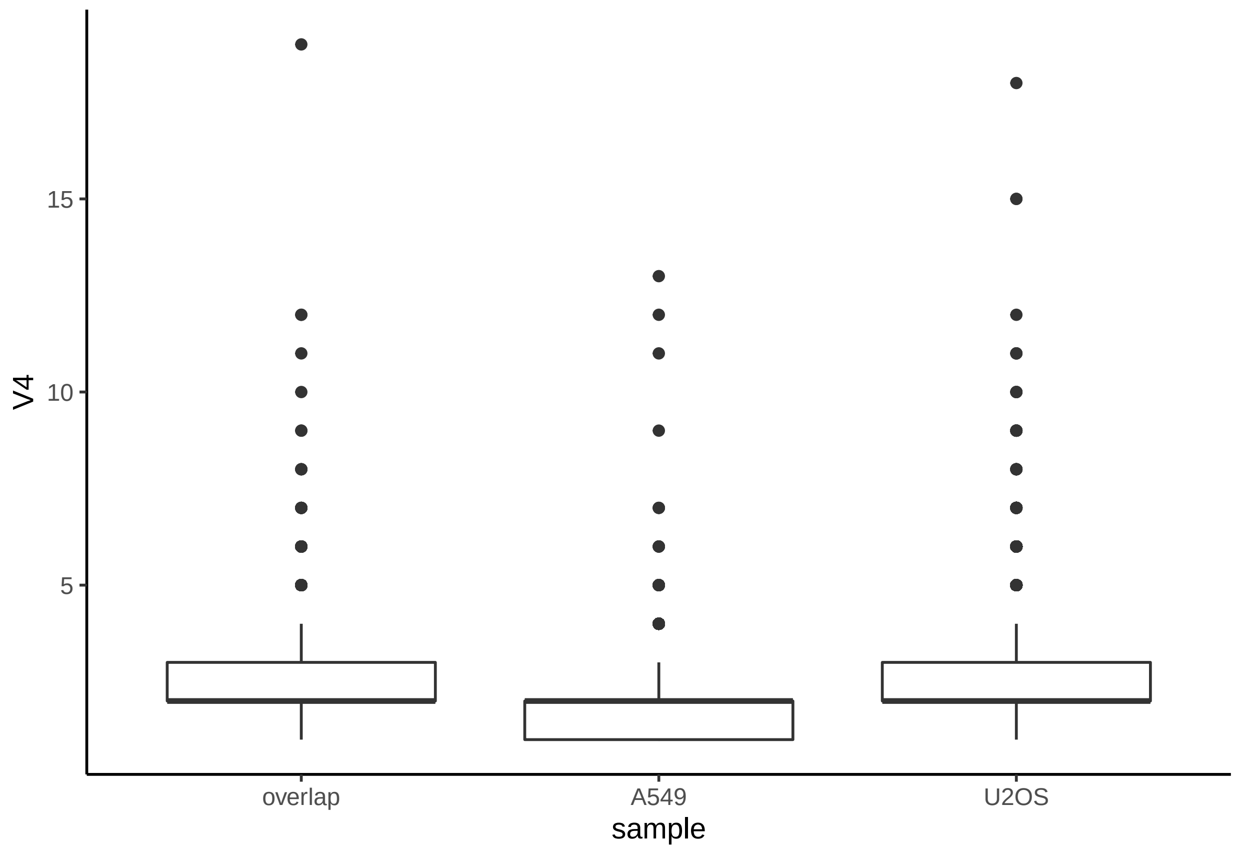
```r
wilcox.test(overlap[,2], U2OS[,2])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  overlap[, 2] and U2OS[, 2]
## W = 47105723, p-value = 1.543e-13
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(A549[,2], U2OS[,2])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549[, 2] and U2OS[, 2]
## W = 21273775, p-value = 1.162e-09
## alternative hypothesis: true location shift is not equal to 0
```

```
ggplot(data = toPlot, mapping = aes(x = sample, y = V4)) +
  geom_boxplot() + gg_options
```



```
wilcox.test(overlap[,4], A549[,4])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  overlap[, 4] and A549[, 4]
## W = 6811491, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(overlap[,4], U2OS[,4])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  overlap[, 4] and U2OS[, 4]
## W = 43761084, p-value = 0.5426
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(A549[,4], U2OS[,4])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  A549[, 4] and U2OS[, 4]
## W = 16552480, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

Finding enrichment of GR at induced and repressed gene promoters

Finding GORs that overlap promoters

```
intersectBed -wb -a A549_U2OS_GR_overlap.bed -b ${coordPath}hg38_refseq_promoters.bed > A549_U2OS_GR_ove

intersectBed -wb -a A549_GR_unique.bed -b ${coordPath}hg38_refseq_promoters.bed > A549_GR_unique_promot

intersectBed -wb -a U2OS_GR_unique.bed -b ${coordPath}hg38_refseq_promoters.bed > U2OS_GR_unique_promot
```

Finding statistics for enrichment of GR bound promoters being induced or repressed

```
both_bound <- read.table(
  'A549_U2OS_GR_overlap_promoters_geneNames.bed',
  stringsAsFactors = F, sep = '\t')[,7]
A549_bound <- read.table(
  'A549_GR_unique_promoters_geneNames.bed',
  stringsAsFactors = F, sep = '\t')[,7]
U2OS_bound <- read.table(
  'U2OS_GR_unique_promoters_geneNames.bed',
  stringsAsFactors = F, sep = '\t')[,7]

genesData <- read.table(genesFile, sep = '\t', stringsAsFactors = F)[,14:21]

genesData[is.na(genesData)] <- 1

genesData$A549_sig <- 'no'
genesData$U2OS_sig <- 'no'
genesData$A549_bound <- 'no'
genesData$U2OS_bound <- 'no'

genesData$name <- row.names(genesData)
for (i in 1:nrow(genesData)){
  if (genesData[i,4] < 0.05 & genesData[i,3] > 0){
    genesData[i,9] <- 'up'
  }
  if (genesData[i,4] < 0.05 & genesData[i,3] < 0){
    genesData[i,9] <- 'down'
  }
  if (genesData[i,8] < 0.05 & genesData[i,7] > 0){
    genesData[i,10] <- 'up'
  }
  if (genesData[i,8] < 0.05 & genesData[i,7] < 0){
    genesData[i,10] <- 'down'
  }
  if (genesData[i,13] %in% c(A549_bound, both_bound)){
    genesData[i,11] <- 'yes'
  }
   if (genesData[i,13] %in% c(U2OS_bound, both_bound)){
    genesData[i,12] <- 'yes'
  }
}

bound_stats <- as.data.frame(matrix(nrow = 4, ncol = 6))
colnames(bound_stats) <- c('cell_type', 'gene_exp', 'observed',
                           'expected', 'enrichment', 'fisher_p')
bound_stats$cell_type <- c('A549', 'A549', 'U2OS', 'U2OS')
bound_stats$gene_exp <- c('induced', 'repressed', 'induced', 'repressed')

A549_up_bound <- matrix(c(length(genesData[genesData$A549_bound == 'yes' &
                                     genesData$A549_sig=='up',1]),
                          length(genesData[genesData$A549_bound == 'no' &
                                     genesData$A549_sig=='up',1]),
                          length(genesData[genesData$A549_bound == 'yes' &
                                     genesData$A549_sig=='no',1]),
                          length(genesData[genesData$A549_bound == 'no' &
                                     genesData$A549_sig=='no',1])), nrow = 2)

bound_stats[1,3] <- length(genesData[genesData$A549_bound == 'yes' &
                                       genesData$A549_sig=='up',1])
bound_stats[1,4] <- length(genesData[genesData$A549_bound == 'yes',1])/
  sum(A549_up_bound)*length(genesData[genesData$A549_sig=='up',1])
bound_stats[1,5] <- bound_stats[1,3]/bound_stats[1,4]
bound_stats[1,6] <- fisher.test(A549_up_bound)$p.value


A549_down_bound <- matrix(c(length(genesData[genesData$A549_bound == 'yes' &
                                       genesData$A549_sig=='down',1]),
                            length(genesData[genesData$A549_bound == 'no' &
                                       genesData$A549_sig=='down',1]),
                            length(genesData[genesData$A549_bound == 'yes' &
                                       genesData$A549_sig=='no',1]),
                            length(genesData[genesData$A549_bound == 'no' &
                                       genesData$A549_sig=='no',1])), nrow = 2)

bound_stats[2,3] <- length(genesData[genesData$A549_bound == 'yes' &
                                       genesData$A549_sig=='down',1])
bound_stats[2,4] <- length(genesData[genesData$A549_bound == 'yes',1])/
```

```
  sum(A549_up_bound)*length(genesData[genesData$A549_sig=='down',1])
bound_stats[2,5] <- bound_stats[2,3]/bound_stats[2,4]
bound_stats[2,6] <- fisher.test(A549_down_bound)$p.value


U2OS_up_bound <- matrix(c(length(genesData[genesData$U2OS_bound == 'yes'
                                      & genesData$U2OS_sig=='up',1]),
                          length(genesData[genesData$U2OS_bound == 'no' &
                                            genesData$U2OS_sig=='up',1]),
                          length(genesData[genesData$U2OS_bound == 'yes' &
                                            genesData$U2OS_sig=='no',1]),
                          length(genesData[genesData$U2OS_bound == 'no' &
                                            genesData$U2OS_sig=='no',1])), nrow = 2)

bound_stats[3,3] <- length(genesData[genesData$U2OS_bound == 'yes'
                                  & genesData$U2OS_sig=='up',1])
bound_stats[3,4] <- length(genesData[genesData$U2OS_bound == 'yes',1])/
  sum(U2OS_up_bound)*length(genesData[genesData$U2OS_sig=='up',1])
bound_stats[3,5] <- bound_stats[3,3]/bound_stats[3,4]
bound_stats[3,6] <- fisher.test(U2OS_up_bound)$p.value


U2OS_down_bound <- matrix(c(length(genesData[genesData$U2OS_bound == 'yes' &
                                              genesData$U2OS_sig=='down',1]),
                            length(genesData[genesData$U2OS_bound == 'no' &
                                              genesData$U2OS_sig=='down',1]),
                            length(genesData[genesData$U2OS_bound == 'yes' &
                                              genesData$U2OS_sig=='no',1]),
                            length(genesData[genesData$U2OS_bound == 'no' &
                                              genesData$U2OS_sig=='no',1])), nrow = 2)

bound_stats[4,3] <- length(genesData[genesData$U2OS_bound == 'yes' &
                                    genesData$U2OS_sig=='down',1])
bound_stats[4,4] <- length(genesData[genesData$U2OS_bound == 'yes',1])/
  sum(U2OS_up_bound)*length(genesData[genesData$U2OS_sig=='down',1])
bound_stats[4,5] <- bound_stats[4,3]/bound_stats[4,4]
bound_stats[4,6] <- fisher.test(U2OS_down_bound)$p.value


print(bound_stats)
```

```
##   cell_type  gene_exp observed  expected enrichment     fisher_p
## 1      A549   induced       76 16.085033  4.7248892 1.452719e-32
## 2      A549 repressed       13  9.744719  1.3340559 6.238666e-02
## 3      U2OS   induced      234 86.240964  2.7133277 2.721348e-60
## 4      U2OS repressed       61 91.373494  0.6675897 1.000000e+00
```

Finding genomic distribution of GR binding

```
coordPath='../Manuscript_data/'
#Promoters:
```

```
intersectBed -u -a A549_U2OS_GR_overlap.bed \
-b ${coordPath}hg38_refseq_promoters.bed | wc -l

intersectBed -u -a A549_GR_unique.bed \
-b ${coordPath}hg38_refseq_promoters.bed | wc -l

intersectBed -u -a U2OS_GR_unique.bed \
-b ${coordPath}hg38_refseq_promoters.bed | wc -l

#Intragenic:
intersectBed -u -a A549_U2OS_GR_overlap.bed -b ${coordPath}hg38_refseq.bed |
intersectBed -v -a stdin -b ${coordPath}hg38_refseq_promoters.bed | wc -l

intersectBed -u -a A549_GR_unique.bed -b ${coordPath}hg38_refseq.bed |
intersectBed -v -a stdin -b ${coordPath}hg38_refseq_promoters.bed | wc -l

intersectBed -u -a U2OS_GR_unique.bed -b ${coordPath}hg38_refseq.bed |
intersectBed -v -a stdin -b ${coordPath}hg38_refseq_promoters.bed | wc -l

#Intergenic:
intersectBed -v -a A549_U2OS_GR_overlap.bed -b ${coordPath}hg38_refseq.bed |
intersectBed -v -a stdin -b ${coordPath}hg38_refseq_promoters.bed | wc -l

intersectBed -v -a A549_GR_unique.bed -b ${coordPath}hg38_refseq.bed |
intersectBed -v -a stdin -b ${coordPath}hg38_refseq_promoters.bed | wc -l

intersectBed -v -a U2OS_GR_unique.bed -b ${coordPath}hg38_refseq.bed |
intersectBed -v -a stdin -b ${coordPath}hg38_refseq_promoters.bed | wc -l
```

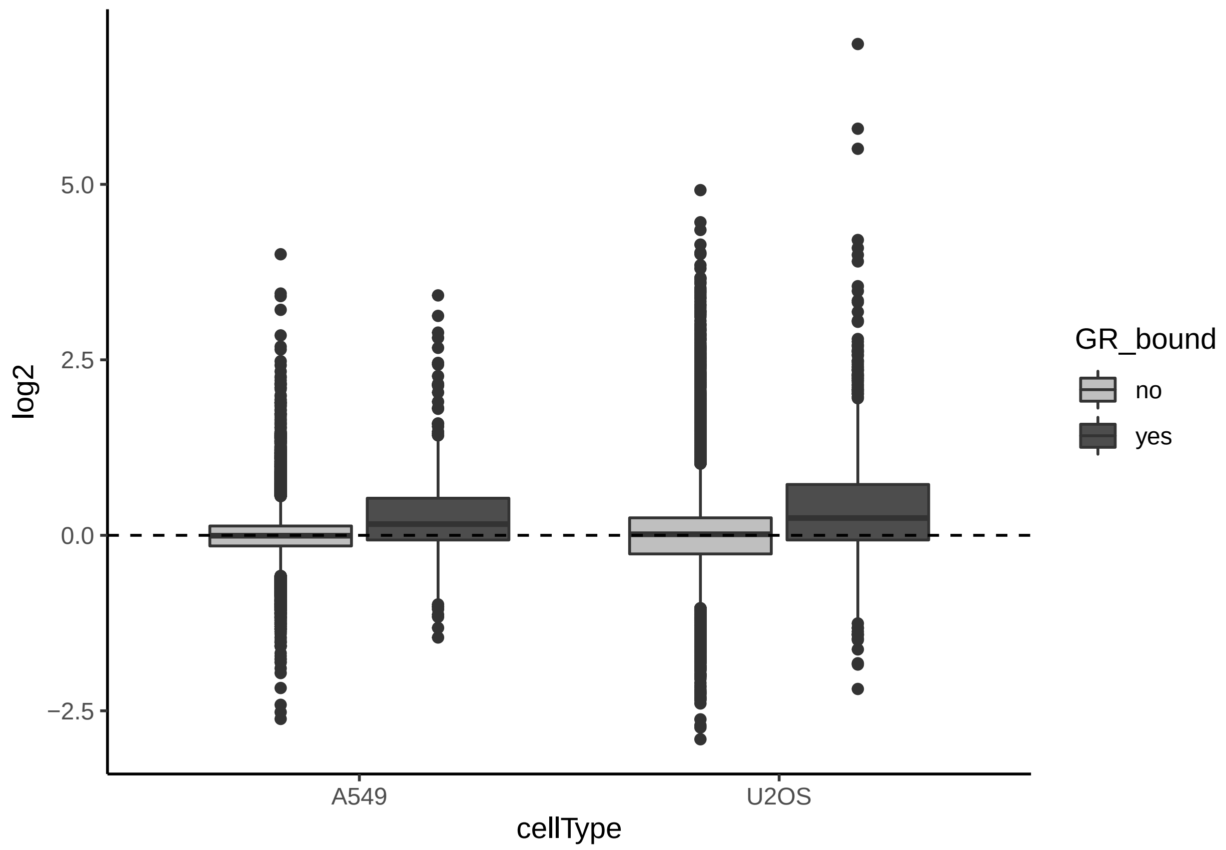
```
## 264
## 159
## 702
## 1922
## 1245
## 8346
## 2410
## 1313
## 11247
```

```
toPlot <- as.data.frame(matrix(nrow = 2*nrow(genesData), ncol = 4))
colnames(toPlot) <- c('gene', 'cellType', 'GR_bound', 'log2')

toPlot$gene <- rep(rownames(genesData), 2)
toPlot$cellType <- c(rep('A549', nrow(genesData)),
                     rep('U2OS', nrow(genesData)))
toPlot$GR_bound <- c(genesData$A549_bound, genesData$U2OS_bound)
toPlot$log2 <- c(genesData$A549_wt_100vs0dex_log2,
                 genesData$U2OS_wt_100vs0dex_log2)

ggplot(
  data = toPlot,
  aes(x=cellType, fill=GR_bound, y=log2)) +
  geom_boxplot() + gg_options +
```

```r
  geom_hline(yintercept = 0, linetype='dashed') +
  scale_fill_manual(values = c('grey75', 'grey30'))
```



```r
wilcox.test(toPlot[toPlot$cellType=='A549' & toPlot$GR_bound=='yes', 4],
            toPlot[toPlot$cellType=='A549' & toPlot$GR_bound=='no', 4])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  toPlot[toPlot$cellType == "A549" & toPlot$GR_bound == "yes",  and toPlot[toPlot$cellType == "
## W = 2496574, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(toPlot[toPlot$cellType=='U2OS' & toPlot$GR_bound=='yes', 4],
            toPlot[toPlot$cellType=='U2OS' & toPlot$GR_bound=='no', 4])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  toPlot[toPlot$cellType == "U2OS" & toPlot$GR_bound == "yes",  and toPlot[toPlot$cellType == "
## W = 5071885, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```