

```

---
title: "R Notebook for Chd4 and Brg1 project"
output: html_notebook
editor_options:
  chunk_output_type: console
---

```{r global options, include = FALSE}
knitr::opts_chunk$set(echo=T, include = T, warning=FALSE, message=FALSE, tidy = TRUE, tidy.opts = list(comment = FALSE))
```

# ```{r cachecontrol, cache = TRUE, cache.extra = tools::md5sum(expr_data_norm)}
# knitr::opts_chunk$set(cache.rebuild = TRUE)
# ```

#####
#### LOAD LIBRARIES
#####
```{r setup}
library(DESeq2)
library(preprocessCore)
library(reshape)
library(ggplot2)
library(doBy)
library(ggpubr)
library(gridExtra)
library(Hmisc)
library(plyr)
library(pheatmap)
library(Gviz)
library(rlist)
library(GenomicRanges)
library(GenomicFeatures)
library(TxDb.Mmusculus.UCSC.mm10.knownGene)
library(rtracklayer)
library(trackViewer)
library(plyr)
library(ComplexHeatmap)
library("GeneOverlap")
library(Vennerable)
theme_set(theme_classic2())
source("~/software/myScripts/R-scripts/FUNCTIONS.R")
```

#####
#### LOAD OR CREATE FILES
#####

```

```

```{r}
##-----##
Load gene promoter file (esc)
##-----##

df_esc <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/analysis/promoters_esc/GENES_PROMOTER.BED.annotation.tpm",
header=T, comment.char = "")
dim(df_esc) # N=21826

df_expr <- read.table("~/bric_share/bp_nextgen/09_ALL_PUBLIC/rna-seq/mouse/mef_esc_reprogramming/mmc2.txt", header=T, comment.char =
"")
df_expr <- df_expr[,c(1,2,3,4,6)]
colnames(df_expr) <- c("gene", "expr_mef", "expr_mef48", "expr_ipsc", "expr_esc")
df_esc <- merge(df_esc, df_expr, by.x="name", by.y="gene")

df_hk <- read.csv("~/genomes/annotations/GENES/Housekeeping_GenesMouse.csv", sep = ";")
df_esc$house_keeping <- ifelse(df_esc$name %in% df_hk$Gene, "yes", "no")
dim(df_esc) # N=18394

##-----##
Load file containing gene expression in esc (wt), and upon KD of:
brg1, chd1, chd2, chd4, chd6, chd8, chd9, ep400, ezh2, ring1, mll34, mll2 and h2az
##-----##

df_expr <- read.table("~/bric_share/bp_nextgen/09_ALL_PUBLIC/rna-seq/mouse/mef_esc_reprogramming/dieuleveult_2016/GENE_EXPR.MATRIX")
colnames(df_expr) <- c("gene", "brd4_fc", "brd4_pVal", "brd4_qVal", "brd4_class",
"brg1_fc", "brg1_pVal", "brg1_qVal", "brg1_class",
"chd1_fc", "chd1_pVal", "chd1_qVal", "chd1_class",
"chd2_fc", "chd2_pVal", "chd2_qVal", "chd2_class",
"chd4_fc", "chd4_pVal", "chd4_qVal", "chd4_class",
"chd6_fc", "chd6_pVal", "chd6_qVal", "chd6_class",
"chd8_fc", "chd8_pVal", "chd8_qVal", "chd8_class",
"chd9_fc", "chd9_pVal", "chd9_qVal", "chd9_class",
"ep400_fc", "ep400_pVal", "ep400_qVal", "ep400_class",
"ezh2_fc", "ezh2_pVal", "ezh2_qVal", "ezh2_class",
"ring1_fc", "ring1_pVal", "ring1_qVal", "ring1_class",
"suz12_fc", "suz12_pVal", "suz12_qVal", "suz12_class",
"mll34_fc", "mll34_pVal", "mll34_qVal", "mll34_class",
"mll2_fc", "mll2_pVal", "mll2_qVal", "mll2_class",
"h2az_fc", "h2az_pVal", "h2az_qVal", "h2az_class")

df_esc <- merge(df_esc, df_expr, by.x="name", by.y="gene")
dim(df_esc) # N=13520

df_esc <- df_esc[,c(2:4,1,5:ncol(df_esc))]
write.table(df_esc, "~/project/chip-seq-
analysis/analysis_chd4_brg1/analysis/promoters_esc/GENES_PROMOTER.BED.annotation.tpm.selected", quote = F, col.names=T, row.names =
F, sep="\t")

```

```

df_esc <- df_esc[which(df_esc$promoter_activity!="1_inactive" & df_esc$promoter_activity!="3_repressed"),]
dim(df_esc) # N=9345

##-----##
Define ep400_cluster_merged class
##-----##

df_esc <- dplyr::arrange(df_esc, promoter_activity, promoter_class, desc(pol2_esc_wt_Rep1))
df_esc$ep400_cluster <- "low"
df_esc[which(df_esc$pol2_esc_wt_Rep1 > summary(df_esc$pol2_esc_wt_Rep1)[2] &
 df_esc$pol2_esc_wt_Rep1 <= summary(df_esc$pol2_esc_wt_Rep1)[5]),]$ep400_cluster <- "average"
df_esc[which(df_esc$pol2_esc_wt_Rep1 > summary(df_esc$pol2_esc_wt_Rep1)[5]),]$ep400_cluster <- "high"
hist(log(df_esc$pol2_esc_wt_Rep1)); abline(v = c(log(summary(df_esc$pol2_esc_wt_Rep1)[2]), log(summary(df_esc$pol2_esc_wt_Rep1)
[5])), col="red", lwd=3, lty=2)
table(df_esc$ep400_cluster)

df_esc$ep400_cluster_merged <- "bivalent"
df_esc[which(df_esc$promoter_activity=="5_active"),]$ep400_cluster_merged <-
as.character(df_esc[which(df_esc$promoter_activity=="5_active"),]$ep400_cluster)
df_esc$ep400_cluster_merged <- factor(df_esc$ep400_cluster_merged, levels=c("bivalent", "high", "average", "low"))
df_esc$ep400_cluster_merged <- factor(df_esc$ep400_cluster_merged, levels=c("bivalent", "low", "average", "high"),
 ordered=T)
table(df_esc$ep400_cluster_merged)

df_esc$promoter_class_merged <- "0_bivalent"
df_esc[which(df_esc$promoter_activity=="5_active"),]$promoter_class_merged <-
as.character(df_esc[which(df_esc$promoter_activity=="5_active"),]$promoter_class)
df_esc$promoter_class_merged <- as.factor(df_esc$promoter_class_merged)
df_esc$promoter_class_merged <- factor(df_esc$promoter_class_merged, levels=c("0_bivalent", "3_broad", "2_medium", "1_narrow"),
 order=T)
table(df_esc$promoter_class_merged)

##-----##
Define chd4 and brg1 ratio classes
##-----##

df_esc$ratio_chd4_brg1 <- ((df_esc$chd4_esc_Rep1+1)/(df_esc$brg1_esc_Rep1+1))
df_esc$ratio_chd4_brg1_class <- "brg1"
df_esc[which(df_esc$ratio_chd4_brg1 > summary(df_esc$ratio_chd4_brg1)[2] & df_esc$ratio_chd4_brg1 <= summary(df_esc$ratio_chd4_brg1)
[5]),]$ratio_chd4_brg1_class <- "equal"
df_esc[which(df_esc$ratio_chd4_brg1 > summary(df_esc$ratio_chd4_brg1)[5]),]$ratio_chd4_brg1_class <- "chd4"
table(df_esc$ratio_chd4_brg1_class)

##-----##
pol2 pausing data
##-----##

```

```

df <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/analysis/promoters_esc/allGenes_pol2pausing.bed", header = T)
df$gene_coor <- sprintf("%s:%d-%d", dfchr, dfstart, df$end)
df$gene_length <- df$end - df$start
colnames(df) <- sprintf("pausing_%s", colnames(df))
df_esc <- merge(df_esc, df[,c(4,8:ncol(df))], by.x="name", by.y="pausing_name", all.x=T)
df_esc <- df_esc[,c(2:4,1,5:ncol(df_esc))]

pol2 pausing data (counts)
df <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/analysis/promoters_esc/allGenes_pol2Signal.bed", header=T)

REPLICATE1
mat <- t(apply(df[,c(11:ncol(df))], 1, function(x) as.numeric(gsub("^.*/", "", x))))
row.names(mat) <- df$name
colnames(mat) <- colnames(df[,c(11:ncol(df))])
res.brg1 <- matrix2deseq2(mat[,c(1:2,3:4)], c("ko,ko,wt,wt"))
colnames(res.brg1)[grep("log2FoldChange", colnames(res.brg1))] <- "pol2_fc_gb_Brg1_rep1"
df <- merge(df, res.brg1[,c("gene", "pol2_fc_gb_Brg1_rep1")], by.x="name", by.y="gene")
res.chd4 <- matrix2deseq2(mat[,c(6:7,9:10)], c("ko,ko,wt,wt"))
colnames(res.chd4)[grep("log2FoldChange", colnames(res.chd4))] <- "pol2_fc_gb_Ch4_rep1"
df <- merge(df, res.chd4[,c("gene", "pol2_fc_gb_Ch4_rep1")], by.x="name", by.y="gene")

mat <- t(apply(df[,c(11:ncol(df))], 1, function(x) as.numeric(gsub("/.*$", "", x))))
row.names(mat) <- df$name
colnames(mat) <- colnames(df[,c(11:ncol(df))])
res.brg1 <- matrix2deseq2(mat[,c(1:2,3:4)], c("ko,ko,wt,wt"))
colnames(res.brg1)[grep("log2FoldChange", colnames(res.brg1))] <- "pol2_fc_tssr_Brg1_rep1"
df <- merge(df, res.brg1[,c("gene", "pol2_fc_tssr_Brg1_rep1")], by.x="name", by.y="gene")
res.chd4 <- matrix2deseq2(mat[,c(6:7,9:10)], c("ko,ko,wt,wt"))
colnames(res.chd4)[grep("log2FoldChange", colnames(res.chd4))] <- "pol2_fc_tssr_Ch4_rep1"
df <- merge(df, res.chd4[,c("gene", "pol2_fc_tssr_Ch4_rep1")], by.x="name", by.y="gene")

REPLICATE2
mat <- t(apply(df[,c(11:ncol(df))], 1, function(x) as.numeric(gsub("^.*/", "", x))))
row.names(mat) <- df$name
colnames(mat) <- colnames(df[,c(11:ncol(df))])
res.brg1 <- matrix2deseq2(mat[,c(11:12,17,18)], c("ko,ko,wt,wt"))
colnames(res.brg1)[grep("log2FoldChange", colnames(res.brg1))] <- "pol2_fc_gb_Brg1_rep2"
df <- merge(df, res.brg1[,c("gene", "pol2_fc_gb_Brg1_rep2")], by.x="name", by.y="gene")
res.chd4 <- matrix2deseq2(mat[,c(13:14,17:18)], c("ko,ko,wt,wt"))
colnames(res.chd4)[grep("log2FoldChange", colnames(res.chd4))] <- "pol2_fc_gb_Ch4_rep2"
df <- merge(df, res.chd4[,c("gene", "pol2_fc_gb_Ch4_rep2")], by.x="name", by.y="gene")

mat <- t(apply(df[,c(11:ncol(df))], 1, function(x) as.numeric(gsub("/.*$", "", x))))
row.names(mat) <- df$name
colnames(mat) <- colnames(df[,c(11:ncol(df))])
res.brg1 <- matrix2deseq2(mat[,c(11:12,17,18)], c("ko,ko,wt,wt"))
colnames(res.brg1)[grep("log2FoldChange", colnames(res.brg1))] <- "pol2_fc_tssr_Brg1_rep2"
df <- merge(df, res.brg1[,c("gene", "pol2_fc_tssr_Brg1_rep2")], by.x="name", by.y="gene")

```

```

res.chd4 <- matrix2deseq2(mat[,c(13:14,17:18)], c("ko,ko,wt,wt"))
colnames(res.chd4)[grep("log2FoldChange", colnames(res.chd4))] <- "pol2_fc_tssr_Ch4_rep2"
df <- merge(df, res.chd4[,c("gene", "pol2_fc_tssr_Ch4_rep2")], by.x="name", by.y="gene")

df <- df[,c(2:4,1,5:ncol(df))]
df_esc <- merge(df_esc, df[,c(4,32:39)], by.x="name", by.y="name", all.x=T)
df_esc <- df_esc[,c(2:4,1,5:ncol(df_esc))]

GRO-seq data (directionality)
df <- read.table(pipe("cat /home/pundhir/project/chip-seq-
analysis/analysis_mel_me3/promoters/esc/GENES_PROMOTER.BED.RC.CLUSTERS.GENE.groseqDirectionality | cut -f 1-12"))
colnames(df) <- c("chr", "start", "end", "gene", "score", "strand", "dist_tss_down", "cluster",
"groseq_wt_down_rep1", "groseq_wt_up_rep1", "groseq_wt_down_rep2", "groseq_wt_up_rep2")
NORM_METHOD="qu"
df$groseq_directionality_rep1 <- apply(scale_df(df[,c(9:10)], NORM_METHOD), 1, function(x) (x[1]-x[2])/(x[1]+x[2]))
df$groseq_directionality_rep2 <- apply(scale_df(df[,c(11:12)], NORM_METHOD), 1, function(x) (x[1]-x[2])/(x[1]+x[2]))
gene_enzyme_enhancer <- merge(gene_enzyme_enhancer[,c(1:ncol(gene_enzyme_enhancer))], df[,c(4,13,14)], by.x="gene", by.y="gene")
gene_enzyme_enhancer <- gene_enzyme_enhancer[,c(2:4,1,5:ncol(gene_enzyme_enhancer))]
gene_enzyme_enhancer$groseq_class <- "0"
gene_enzyme_enhancer[which(gene_enzyme_enhancer$groseq_directionality_rep2 < -0.35),]$groseq_class <- "-1"
gene_enzyme_enhancer[which(gene_enzyme_enhancer$groseq_directionality_rep2 > 0.35),]$groseq_class <- "1"
gene_enzyme_enhancer$groseq_class <- factor(gene_enzyme_enhancer$groseq_class, levels=c("-1", "0", "1"), ordered=T)
table(gene_enzyme_enhancer$groseq_class)
#
write.table(gene_enzyme_enhancer, "/home/pundhir/project/chip-seq-analysis/analysis_mel_me3/analysis/gene_enzyme_enhancer.bed",
quote = F, col.names=T, row.names = F, sep="\t")

##-----##
Load gene promoter file (mef)
##-----##

df_mef <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/analysis/promoters_mef/GENES_PROMOTER.BED.annotation.tpm",
header=T, comment.char = "")
dim(df_mef) # N=21826

df_expr <- read.table("~/bric_share/bp_nextgen/09_ALL_PUBLIC/rna-seq/mouse/mef_esc_reprogramming/mmc2.txt", header=T, comment.char =
"")
df_expr <- df_expr[,c(1,2,3,4,6)]
colnames(df_expr) <- c("gene", "expr_mef", "expr_mef48", "expr_ipsc", "expr_esc")
df_mef <- merge(df_mef, df_expr, by.x="name", by.y="gene")

df_hk <- read.csv("~/genomes/annotations/GENES/Housekeeping_GenesMouse.csv", sep = ";")
df_mef$house_keeping <- ifelse(df_mef$name %in% df_hk$Gene, "yes", "no")
dim(df_mef) # N=18394

df_mef <- df_mef[which(df_mef$promoter_activity!="1_inactive" & df_mef$promoter_activity!="3_repressed"),]
dim(df_mef) # N=10351

```

```

##-----##
Define ep400_cluster_merged class
##-----##

df_mef <- dplyr::arrange(df_mef, promoter_activity, promoter_class, desc(pol2_mef_wt_Rep1))
df_mef$ep400_cluster <- "low"
df_mef[which(df_mef$pol2_mef_wt_Rep1 > summary(df_mef$pol2_mef_wt_Rep1)[2] &
 df_mef$pol2_mef_wt_Rep1 <= summary(df_mef$pol2_mef_wt_Rep1)[5]),]$ep400_cluster <- "average"
df_mef[which(df_mef$pol2_mef_wt_Rep1 > summary(df_mef$pol2_mef_wt_Rep1)[5]),]$ep400_cluster <- "high"
hist(log(df_mef$pol2_mef_wt_Rep1)); abline(v = c(log(summary(df_mef$pol2_mef_wt_Rep1)[2]), log(summary(df_mef$pol2_mef_wt_Rep1)
[5])), col="red", lwd=3, lty=2)
table(df_mef$ep400_cluster)

df_mef$ep400_cluster_merged <- "bivalent"
df_mef[which(df_mef$promoter_activity=="5_active"),]$ep400_cluster_merged <-
as.character(df_mef[which(df_mef$promoter_activity=="5_active"),]$ep400_cluster)
df_mef$ep400_cluster_merged <- factor(df_mef$ep400_cluster_merged, levels=c("bivalent", "high", "average", "low"))
table(df_mef$ep400_cluster_merged)

df_mef$promoter_class_merged <- "0_bivalent"
df_mef[which(df_mef$promoter_activity=="5_active"),]$promoter_class_merged <-
as.character(df_mef[which(df_mef$promoter_activity=="5_active"),]$promoter_class)
df_mef$promoter_class_merged <- as.factor(df_mef$promoter_class_merged)
df_mef$promoter_class_merged <- factor(df_mef$promoter_class_merged, levels=c("0_bivalent", "3_broad", "2_medium", "1_narrow"),
order=T)
table(df_mef$promoter_class_merged)

##-----##
Define chd4 and brg1 ratio classes
##-----##

df_mef$ratio_chd4_brg1 <- ((df_mef$chd4_mef_Rep1+1)/(df_mef$brg1_mef_Rep1+1))
df_mef$ratio_chd4_brg1_class <- "brg1"
df_mef[which(df_mef$ratio_chd4_brg1 > summary(df_mef$ratio_chd4_brg1)[2] & df_mef$ratio_chd4_brg1 <= summary(df_mef$ratio_chd4_brg1)
[5]),]$ratio_chd4_brg1_class <- "equal"
df_mef[which(df_mef$ratio_chd4_brg1 > summary(df_mef$ratio_chd4_brg1)[5]),]$ratio_chd4_brg1_class <- "chd4"
table(df_mef$ratio_chd4_brg1_class)
```

#####
#### WRITE MATRIX TO FILE
#### THESE (promoters_esc.bed and promoters_mef.bed) ARE THE MAIN DATA FILES USED FOR THE ANALYSIS
#### USERS CAN LOAD THESE FILES TO REPRODUCE THE ANALYSIS
#####
```{r}
write.table(df_esc[order(df_esc$X.chr, df_esc$start, df_esc$end),],
 "~/project/chip-seq-analysis/analysis_chd4_brg1/analysis/promoters_esc.bed", quote = F, sep="\t",
 row.names=F, col.names = T)

```

```
write.table(df_mef[order(df_mef$X.chr, df_mef$start, df_mef$end),],
 "~/project/chip-seq-analysis/analysis_chd4_brg1/analysis/promoters_mef.bed", quote = F, sep="\t",
 row.names=F, col.names = T)
...
```

```
#####
ANALYSIS SHOWING THE ROLE OF POL2 AND PRC2 IN EXPLAINING THE ANATAGONISTIC FUNCTION OF CHD4 AND BRG1
CODE TO REPRODUCE FIGURES: Supplementary Figure 1A, 1D, Figure 1D and Supplementary Figure 3F
#####
```

```
```{r}
##-----##
## volcano plot analysis
##-----##
```

```
p1 <- ggplot(df_esc, aes(x=chd4_fc, y=-log10(chd4_pVal))) +
  geom_point(aes(colour = chd4_fc, fill = chd4_fc), shape = 21, size = 3, stroke = 1) +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(9, "PuOr"), bias=0.7)(256))) +
  scale_color_gradientn(colours = rev(colorRampPalette(brewer.pal(9, "PuOr"), bias=0.7)(256))) +
  theme(legend.position="none") + theme_classic() + scale_x_reverse() +
  geom_vline(xintercept=c(-0.41, 0.41), linetype = 2) +
  scale_y_continuous(position = "right")
```

```
p2 <- ggplot(df_esc, aes(x=brg1_fc, y=-log10(brg1_pVal))) +
  geom_point(aes(colour = brg1_fc, fill = brg1_fc), shape = 21, size = 3, stroke = 1) +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(9, "PuOr"), bias=1.1)(256))) +
  scale_color_gradientn(colours = rev(colorRampPalette(brewer.pal(9, "PuOr"), bias=1.1)(256))) +
  theme(legend.position="none") + theme_classic() + scale_x_reverse() +
  geom_vline(xintercept=c(-0.41, 0.41), linetype = 2) +
  scale_y_continuous(position = "right")
```

```
test <- df_esc
test$chd4_class <- as.character(cut2(test$chd4_fc, m=400))
test[which(df_esc$chd4_class=="neutral"),]$chd4_class <- "0"
test$chd4_class <- as.factor(as.numeric(gsub("]", "", gsub("[)]+", "", gsub("^.*", "", test$chd4_class))))
table(test$chd4_class)
test.melt <- melt(summaryBy(pol2_esc_wt_Repl ~ chd4_class, data = test, FUN = list(median)))
p3 <- ggplot(test.melt, aes(chd4_class, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=1.1)(56))) +
  theme(axis.text.x = element_text(size=10, angle=45)) + theme_bw()
```

```
test.melt <- melt(summaryBy(ezh2_esc_Repl ~ chd4_class, data = test, FUN = list(median)))
p4 <- ggplot(test.melt, aes(chd4_class, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=0.7)(56))) +
  theme(axis.text.x = element_text(size=10, angle=45)) + theme_bw()
```

```
test$brg1_class <- as.character(cut2(test$brg1_fc, m=500))
test[which(df_esc$brg1_class=="neutral"),]$brg1_class <- "0"
```

```

test$brg1_class <- as.factor(as.numeric(gsub("]", "", gsub("[)]+", "", gsub("^.*", "", test$brg1_class))))
table(test$brg1_class)
test.melt <- melt(summaryBy(pol2_esc_wt_Rep1 ~ brg1_class, data = test, FUN = list(median)))
p5 <- ggplot(test.melt, aes(brg1_class, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=1.1)(56))) +
  theme(axis.text.x = element_text(size=10, angle=45)) + theme_bw()

test.melt <- melt(summaryBy(ezh2_esc_Rep1 ~ brg1_class, data = test, FUN = list(median)))
p6 <- ggplot(test.melt, aes(brg1_class, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=0.7)(56))) +
  theme(axis.text.x = element_text(size=10, angle=45)) + theme_bw()

test.melt <- melt(summaryBy(chd4_fc ~ chd4_class, data = test, FUN = list(median)))
p7 <- ggplot(test.melt, aes(chd4_class, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "PuOr"), bias=0.7)(56))) +
  theme(axis.text.x = element_text(size=10, angle=45)) + theme_bw()

test.melt <- melt(summaryBy(brg1_fc ~ brg1_class, data = test, FUN = list(median)))
p8 <- ggplot(test.melt, aes(brg1_class, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "PuOr"), bias=1.2)(56))) +
  theme(axis.text.x = element_text(size=10, angle=45)) + theme_bw()

freqPlot(melt(table(gene_enzyme_enhancer[,c(165,78)])), "density") + theme_bw()
freqPlot(melt(table(gene_enzyme_enhancer[,c(165,90)])), "density") + theme_bw()

g <- ggarrange(p1, p2, p3, p5, p4, p6, p7, p8, nrow=4, ncol=2)
g
ggsave(g, dpi=300, height=15, width=18, filename="~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/gene_expression_volcano1.pdf",
useDingbats=FALSE)

##-----##
## heatmap showing that mutually exclusive ep400 and prc2 binding explain anatagonistic function of Chd4 and Brg1
##-----##

## All (promoter_class)
test <- df_esc
test$diff_fc <- log((test$expr_esc+1)/(test$expr_mef+1))
test$ep400_cluster <- cut2(test$pol2_esc_wt_Rep1, m=25)
test$gene_length <- log(test$gene_length)

test.melt <- melt(summaryBy(chd4_fc ~ ep400_cluster + promoter_class_merged, data = test, FUN = list(median)))
p1 <- ggplot(test.melt, aes(ep400_cluster, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=0.55)(56))) +
  facet_grid(~ promoter_class_merged) +
  theme(axis.text.x = element_text(size=10, angle=45))

test.melt <- melt(summaryBy(brg1_fc ~ ep400_cluster + promoter_class_merged, data = test, FUN = list(median)))
#barplot(test.melt[order(test.melt$promoter_class_merged, test.melt$ep400_cluster),]$value)

```

```

p2 <- ggplot(test.melt, aes(ep400_cluster, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=1)(56))) +
  facet_grid(.~promoter_class_merged) +
  theme(axis.text.x = element_text(size=10, angle=45))

test.melt <- melt(summaryBy(ring1_fc ~ ep400_cluster + promoter_class_merged, data = test, FUN = list(median)))
p3 <- ggplot(test.melt, aes(ep400_cluster, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=1.7)(56))) +
  facet_grid(.~promoter_class_merged) +
  theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(CpG_island_length ~ ep400_cluster + promoter_class_merged, data = test, FUN = list(median)))
p4 <- ggplot(test.melt, aes(ep400_cluster, variable, fill=log(value+1))) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(9, "RdBu"), bias=0.8)(256))) +
  facet_grid(.~promoter_class_merged) +
  theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(expr_esc + expr_ipsc + expr_mef48 + expr_mef ~ ep400_cluster + promoter_class_merged, data = test, FUN =
list(median)))
p5 <- ggplot(test.melt, aes(ep400_cluster, variable, fill=log(value))) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(9, "RdBu"), bias=1)(10))) +
  facet_grid(.~promoter_class_merged) +
  theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(diff_fc ~ ep400_cluster + promoter_class_merged, data = test, FUN = list(median)))
p6 <- ggplot(test.melt, aes(ep400_cluster, variable, fill=(value))) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(9, "RdBu"), bias=1)(10))) +
  facet_grid(.~promoter_class_merged) +
  theme(axis.text.x = element_blank())

# comparison between gene expression changes upon chd4 and mbd3 kd
mbd3 <- read.table(pipe("grep -v Gene.symbol ~/bric_share/bp_nextgen/09_ALL_PUBLIC/rna-
seq/mouse/mef_esc_reprogramming/dieuleveult_2016/other/mbd3.fc.uniqueGeneId"))
colnames(mbd3) <- c("gene", "mbd3_fc", "mbd3_pVal", "mbd3_qVal", "mbd3_class")
test <- merge(test, mbd3, by.x="name", by.y="gene")

test.melt <- melt(summaryBy(mbd3_fc ~ ep400_cluster + promoter_class_merged, data = test, FUN = list(median)))
p7 <- ggplot(test.melt, aes(ep400_cluster, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=0.7)(56))) +
  facet_grid(.~promoter_class_merged) +
  theme(axis.text.x = element_text(size=10, angle=45))

test.melt <- melt(summaryBy(gene_length ~ ep400_cluster + promoter_class_merged, data = test, FUN = list(median)))
p8 <- ggplot(test.melt, aes(ep400_cluster, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=1.1)(256))) +
  facet_grid(.~promoter_class_merged) +
  theme(axis.text.x = element_text(size=10, angle=45))

```

```

g <- arrangeGrob(p1, p2, p3, p4, p5, p6, p7, p8, nrow = 8, ncol=1)
ggsave(g, dpi=300, height=25, width=20, filename=~ /project/chip-seq-analysis/analysis_chd4_brg1/pdf/cluster_4_analysis_ALL.pdf",
useDingbats=FALSE)

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/cor_chd4_mbd3_expr_fc.pdf")
test.melt <- summaryBy(chd4_fc + mbd3_fc ~ ep400_cluster, data = test, FUN = list(median))
smoothScatter(test.melt$chd4_fc.median, test.melt$mbd3_fc.median, colramp=colorRampPalette(my.cols), pch=19, cex=.8, useRaster=T,
xlab = "chd4 (logFC)", ylab = "mbd3 (logFC)")
lines(lowess(test.melt$chd4_fc.median, test.melt$mbd3_fc.median),col='red', lty="dashed")
cor.test(test.melt$chd4_fc.median, test.melt$mbd3_fc.median, method="spearman")
dev.off()
```

#####
Validation of gene expression changes with independent datasets
CODE TO REPRODUCE FIGURES: Supplementary Figure 1H
#####
```{r}
##-----##
## validation with Dirk Schubler's data
##-----##
dirk_data <- read.csv("~/bric_share/bp_nextgen/09_ALL_PUBLIC/rna-
seq/mouse/mef_esc_reprogramming/schubeler_2021//GSE158343_fmigschub_RNA_gene_counts.csv.gz")
dirk_data$mean_expr_wt <- rowMeans(dirk_data[,c(5,6)])
dirk_data <- dirk_data[order(dirk_data$SYMBOL, -dirk_data$mean_expr_wt),]
dim(dirk_data)
dirk_data <- dirk_data[match(unique(dirk_data$SYMBOL), dirk_data$SYMBOL),]
dirk_data <- dirk_data[!is.na(dirk_data$SYMBOL),]
dim(dirk_data)
mat <- dirk_data[,c(5:6,11:12)]
row.names(mat) <- dirk_data$SYMBOL
mat <- mat[which(rowMin(as.matrix(mat)) > 2),]
res_dirk_data <- matrix2deseq2(mat, "wt,wt,ko,ko")

## sanity check
# ggplot(res_dirk_data, aes(log2FoldChange, -log10(padj), color=class)) + geom_point() + theme_bw()
# dirk_data_analyzed <- merge(dirk_data, res_dirk_data[,c("gene", "log2FoldChange", "class")], by.x="SYMBOL", by.y="gene")
# df.melt <- melt(dirk_data_analyzed[,c(5:16,23:30,45)])
# df.melt$variable <- gsub("RNA_BRM014.|RNA_DMSO_", "", df.melt$variable)
# df.melt$variable <- factor(df.melt$variable, unique(df.melt$variable), ordered=T)
# df.melt$variable <- gsub("_[0-9]+$", "", df.melt$variable)
# df.melt$variable <- factor(df.melt$variable, unique(df.melt$variable), ordered=T)
# df.melt$value <- log(df.melt$value+1)
# ggline(df.melt, x="variable", y="value", add="mean_se", facet.by = "class") +
#   theme(axis.text.x = element_text(size=10, angle=45)) + xlab("")

test <- merge(res_dirk_data, df_esc[,c("name", "pol2_esc_wt_Rep1", "brg1_fc", "chd4_fc", "ep400_cluster_merged")], by.x="gene",
by.y="name")

```

```

test <- merge(test, dirk_data[,c("SYMBOL", "mean_expr_wt")], by.x="gene", by.y="SYMBOL")
test$expr_class <- cut2(test$mean_expr_wt, g=3)
test$ep400_cluster_merged <- factor(test$ep400_cluster_merged, levels=c("bivalent", "low", "average", "high"), ordered=T)
test$class <- factor(test$class, levels=c("neutral", "down", "up"), ordered=T)

# freqPlot(melt(table(test[,c("ep400_cluster_merged", "class")])), "density")
p1 <- ggboxplot(test[which(test$padj < 0.05)], x = "ep400_cluster_merged", y = "log2FoldChange", fill = "gray", notch=0, notchwidth
= 0.6,
  yscale="none", legend="none", title="inhibitor - BRM014", outlier.shape=NA) + xlab("Gene classes defined in our study") +
  ylab("log2FC in expr. (3hr vs DMSO)") +
  stat_compare_means(comparisons = list(c("average", "low"), c("high", "bivalent")), method.args = list(alternative = "g")) +
  geom_hline(yintercept = 0) +
  theme(axis.text.x = element_text(size=10, angle=45))

p1.1 <- ggboxplot(test, x = "ep400_cluster_merged", y = "log2FoldChange", fill = "gray", notch=0, notchwidth = 0.6,
  yscale="none", legend="none", title="inhibitor - BRM014", outlier.shape=NA) + xlab("Gene classes defined in our study") +
  ylab("log2FC in expr. (3hr vs DMSO)") +
  stat_compare_means(comparisons = list(c("average", "low"), c("high", "bivalent")), method.args = list(alternative = "g"),
  label.y=c(0.7,1)) +
  geom_hline(yintercept = 0) +
  theme(axis.text.x = element_text(size=10, angle=45)) + ylim(c(-1,1.5))

p1.2 <- freqPlot(melt(table(test[which(test$class!="neutral"),c("ep400_cluster_merged", "class")])), "density")

##-----##
## validation with Mbd4 KD data
##-----##
df <- read.table("/localhome/bric/xfd783/bric_share/bp_nextgen/09_ALL_PUBLIC/gro-
seq/mouse/mef_esc_reprogramming/mbd3_kd/GENE_EXPRESSION.MATRIX", header=T)
mat <- df[,c(9:14)]
row.names(mat) <- df$name
mat <- mat[which(row.names(mat) %in% dirk_data$SYMBOL),]
res.chd4 <- matrix2deseq2(mat[,c(2:3,5:6)], "ko,ko,wt,wt", cutoff = 0.05)

test <- merge(res.chd4, df_esc[,c("name", "pol2_esc_wt_Rep1", "brg1_fc", "chd4_fc", "ep400_cluster_merged")], by.x="gene",
by.y="name")
test$ep400_cluster_merged <- factor(test$ep400_cluster_merged, levels=c("bivalent", "low", "average", "high"), ordered=T)

p2 <- ggboxplot(test[which(test$pvalue<0.05)], x = "ep400_cluster_merged", y = "log2FoldChange", fill = "gray", notch=0, notchwidth
= 0.6,
  yscale="none", legend="none", title="Mbd3 0h vs 24h", outlier.shape=NA) + xlab("Gene classes defined in our study") +
  ylab("log2FC in expr. (Mbd3)") +
  stat_compare_means(comparisons = list(c("average", "low"), c("high", "bivalent")), method.args = list(alternative = "l"),
  label.y=c(2,2.4)) +
  geom_hline(yintercept = 0) +
  theme(axis.text.x = element_text(size=10, angle=45))

p2.1 <- ggboxplot(test, x = "ep400_cluster_merged", y = "log2FoldChange", fill = "gray", notch=1, notchwidth = 0.6,

```

```

        yscale="none", legend="none", title="Mbd3 0h vs 24h", outlier.shape=NA) + xlab("Gene classes defined in our study") +
        ylab("log2FC in expr. (Mbd3)") +
        stat_compare_means(comparisons = list(c("average", "low"), c("high", "bivalent")), method.args = list(alternative = "l"),
label.y=c(1.2,1.4)) +
        geom_hline(yintercept = 0) +
        theme(axis.text.x = element_text(size=10, angle=45)) + ylim(c(-2,2))

tmp <- test
tmp[which(tmp$log2FoldChange > 0 & tmp$pvalue < 0.05),]$class <- "up"
tmp[which(tmp$log2FoldChange < 0 & tmp$pvalue < 0.05),]$class <- "down"
table(tmp$class)
p2.2 <- freqPlot(melt(table(tmp[which(tmp$class!="neutral"),c("ep400_cluster_merged", "class")])), "density")

g <- ggarrange(p1, p1.1, p1.2, p2, p2.1, p2.2, nrow=2, ncol=3, labels=c("A", "B", "C", "D", "E", "F"))
ggsave(g, dpi=300, filename="~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/external_validation.pdf", useDingbats=FALSE)

df <- merge(res_dirk_data[,c("gene", "log2FoldChange")], res.chd4[,c("gene", "log2FoldChange")], by.x="gene", by.y="gene")
colnames(df) <- c("gene", "log2FoldChange_brg1", "log2FoldChange_mbd3")
df <- merge(df, df_esc[,c("name", "pol2_esc_wt_Rep1")], by.x="gene", by.y="name")
df$class <- cut2(df$pol2_esc_wt_Rep1, m=1)
test.melt <- summaryBy(log2FoldChange_brg1 + log2FoldChange_mbd3 ~ class, data = df, FUN = list(median))
ggscatter(test.melt, x = "log2FoldChange_brg1.median", y = "log2FoldChange_mbd3.median", color="#636363", alpha=0.6,
        add = "loess", conf.int = TRUE) + stat_cor(label.x = 0, method="spearman") +
        geom_hline(yintercept = 0, lty=2) +
        geom_vline(xintercept = 0, lty=2)
...

#####
#### CHD4 AND BRG1 BINDING FOLD CHANGE AND POL2 PAUSING ANALYSIS (ESC)
#### CODE TO REPRODUCE FIGURES: Figure 3C and 3F
#####
```{r}
test <- df_esc
test$pausing_brg1_wt <- rowMeans(test[,c("pausing_pol2_brg1_wt_Rep1", "pausing_pol2_brg1_wt_Rep2")])
test$pausing_brg1_ko <- rowMeans(test[,c("pausing_pol2_brg1_ko_Rep1", "pausing_pol2_brg1_ko_Rep2")])
test$pausing_mbd3_wt <- rowMeans(test[,c("pausing_pol2S5P_esc_mbd3_24h_Rep2", "pausing_pol2S5P_esc_mbd3_24h_Rep3")])
test$pausing_mbd3_ko <- rowMeans(test[,c("pausing_pol2S5P_esc_mbd3_0h_Rep2", "pausing_pol2S5P_esc_mbd3_0h_Rep3")])

test$pausing_wt <- rowMeans(test[,c("pausing_pol2_esc_wt_Rep1", "pausing_pol2_esc_wt_Rep2")])
test$pausing_brg1_kd <- rowMeans(test[,c("pausing_pol2_esc_brg1_kd_Rep1", "pausing_pol2_esc_brg1_kd_Rep2")])
test$pausing_chd4_kd <- rowMeans(test[,c("pausing_pol2_esc_chd4_kd_Rep1", "pausing_pol2_esc_chd4_kd_Rep2")])

test$ratio_chd4_brg1 <- cut2((test$chd4_esc_Rep1+1)/(test$brg1_esc_Rep1+1), m=25)
test$ep400_cluster_merged_col <- "red"
test[which(test$ep400_cluster_merged=="low"),]$ep400_cluster_merged_col <- "green"
test[which(test$ep400_cluster_merged=="average"),]$ep400_cluster_merged_col <- "blue"
test[which(test$ep400_cluster_merged=="high"),]$ep400_cluster_merged_col <- "yellow"

```

```

test$ratio_ser2p_ser5p <-
as.numeric(normalize.quantiles(as.matrix(test$pol2_esc_ser2p_Rep1))/normalize.quantiles(as.matrix(test$pol2_esc_ser5p_Rep1)))

test <- test[order(test$ratio_chd4_brg1),]

##-----##
plot line plot containing results of pol2 pausing analysis
##-----##

test.ep400 <- test[which(test$ep400_cluster_merged=="high" | test$ep400_cluster_merged=="average"),]
test.ep400$ratio_chd4_brg1 <- log2((test.ep400$chd4_esc_Rep1+1)/(test.ep400$brg1_esc_Rep1+1))

test.ep400$pause_class <- cut2(test.ep400$pol2_fc_gb_Brg1_rep1, m=1500)
table(test.ep400$pause_class)

brg1 <- test.ep400[which(test.ep400$pause_class=="[0.4466, 2.4844]"),]
test.melt <- melt(test.ep400[,c("pause_class", "pausing_brg1_wt", "pausing_brg1_ko")])
ggboxplot(test.melt, x = "variable", y = "value", add = "mean_se") +
 stat_compare_means(comparisons = list(c("pausing_brg1_ko", "pausing_brg1_wt"))) + facet_wrap(~pause_class)

test.melt <- summarySE(test.melt, measurevar = "value", groupvars = c("pause_class", "variable"))
p1 <- ggplot(test.melt, aes(pause_class, value, color=variable)) +
 geom_errorbar(aes(ymin=value-se, ymax=value+se), width=.1, alpha=1) +
 geom_line(aes(group=factor(variable), color=factor(variable)), alpha=1) +
 geom_point(shape = 21, size = 2, fill="white") +
 scale_color_manual(values=c("#a50026", "#313695")) +
 theme(legend.position="none") + theme_bw()

test.melt <- melt(test.ep400[,c("pause_class", "brg1_fc")])
test.melt <- summarySE(test.melt, measurevar = "value", groupvars = c("pause_class", "variable"))
p2 <- ggplot(test.melt, aes(pause_class, value, color=variable)) +
 geom_errorbar(aes(ymin=value-se, ymax=value+se), width=.1, alpha=1) +
 geom_line(aes(group=factor(variable), color=factor(variable)), alpha=1) +
 geom_point(shape = 21, size = 2, fill="white") +
 scale_color_manual(values=c("#33a02c")) +
 theme(legend.position="none") + theme_bw()

test.ep400 <- test[which(test$ep400_cluster_merged=="high" | test$ep400_cluster_merged=="average"),]
test.ep400$ratio_chd4_brg1 <- log2((test.ep400$chd4_esc_Rep1+1)/(test.ep400$brg1_esc_Rep1+1))
test.ep400$pause_class <- cut2(test.ep400$pol2_fc_gb_Ch4_rep1, m=1500)
table(test.ep400$pause_class)

chd4 <- test.ep400[which(test.ep400$pause_class=="[-0.57685,-0.04763]"),]
test.melt <- melt(test.ep400[,c("pause_class", "pausing_mbd3_wt", "pausing_mbd3_ko")])
ggboxplot(test.melt, x = "variable", y = "value", add = "mean_se") +
 stat_compare_means(comparisons = list(c("pausing_mbd3_ko", "pausing_mbd3_wt"))) + facet_wrap(~pause_class)

test.melt <- summarySE(test.melt, measurevar = "value", groupvars = c("pause_class", "variable"))

```

```

p3 <- ggplot(test.melt, aes(pause_class, value, color=variable)) +
 geom_errorbar(aes(ymin=value-se, ymax=value+se), width=.1, alpha=1) +
 geom_line(aes(group=factor(variable), color=factor(variable)), alpha=1) +
 geom_point(shape = 21, size = 2, fill="white") +
 scale_color_manual(values=c("#a50026", "#313695")) +
 theme(legend.position="none") + theme_bw()

test.melt <- melt(test.ep400[,c("pause_class", "chd4_fc")])
test.melt <- summarySE(test.melt, measurevar = "value", groupvars = c("pause_class", "variable"))
p4 <- ggplot(test.melt, aes(pause_class, value, color=variable)) +
 geom_errorbar(aes(ymin=value-se, ymax=value+se), width=.1, alpha=1) +
 geom_line(aes(group=factor(variable), color=factor(variable)), alpha=1) +
 geom_point(shape = 21, size = 2, fill="white") +
 scale_color_manual(values=c("#33a02c")) +
 theme(legend.position="none") + theme_bw()

g1 <- ggarrange(p1, p2, p3, p4, nrow=2, ncol=2, labels=c("A", "B", "C", "D"))

df.melt <- melt(test.ep400[,c("pausing_brg1_wt", "pausing_brg1_ko")])
p5 <- ggplot(df.melt, aes(value, color=variable)) + stat_ecdf() + xlim(c(0.6,1)) + theme_bw()
wilcox.test(test.ep400$pausing_brg1_ko, test.ep400$pausing_brg1_wt, alternative = "l")$p.value

df.melt <- melt(test.ep400[,c("pausing_mbd3_wt", "pausing_mbd3_ko")])
p6 <- ggplot(df.melt, aes(value, color=variable)) + stat_ecdf() + xlim(c(0.6,1)) + theme_bw()
wilcox.test(test.ep400$pausing_mbd3_ko, test.ep400$pausing_mbd3_wt, alternative = "g")$p.value

g2 <- ggarrange(p5, p6, nrow = 2, ncol=2)

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/pausing.pdf")
g1
g2
dev.off()
```

#####
#### Relationship between H2A.Z, H3.3 and Pol2 pausing (ESC)
#### CODE TO REPRODUCE FIGURES: Figure 6B and Supplementary Figure 7A
#####
```{r}
##-----##
correlation plots between H2A.Z, H3.3 and Pol2 pausing in ESC (ep400 bound)
##-----##

groseq_esc <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/analysis/promoters_esc/allGenes_transcriptSignal.bed",
header=T)
transcript_width <- (groseq_esc$end - groseq_esc$start)
groseq_esc <- groseq_esc[which(groseq_esc$dist_closest_tss > 3000),]
tssr <- t(apply(groseq_esc[,c("groseq_esc_Rep1", "groseq_esc_Rep2")], 1, function(x) as.numeric(gsub("/.*$", "", x))))

```

```

colnames(tssr) <- c("groseq_tssr_rep1", "groseq_tssr_rep2")
gb <- t(apply(groseq_esc[,c("groseq_esc_Rep1", "groseq_esc_Rep2")], 1, function(x) as.numeric(gsub("^.*/", "", x))))
colnames(gb) <- c("groseq_gb_rep1", "groseq_gb_rep2")
groseq_esc <- cbind(groseq_esc[,c(4),drop=F], tssr, gb)

groseq_esc$groseq_tssr_rep1_norm <- groseq_esc$groseq_tssr_rep1/(groseq_esc$groseq_tssr_rep1+groseq_esc$groseq_gb_rep1)
groseq_esc$groseq_tssr_rep2_norm <- groseq_esc$groseq_tssr_rep2/(groseq_esc$groseq_tssr_rep2+groseq_esc$groseq_gb_rep2)
groseq_esc$groseq_gb_rep1_norm <- groseq_esc$groseq_gb_rep1/(groseq_esc$groseq_tssr_rep1+groseq_esc$groseq_gb_rep1)
groseq_esc$groseq_gb_rep2_norm <- groseq_esc$groseq_gb_rep2/(groseq_esc$groseq_tssr_rep2+groseq_esc$groseq_gb_rep2)

test <- merge(df_esc, groseq_esc, by.x="name", by.y="name")
test$ratio_chd4_brg1 <- cut2((test$chd4_esc_Rep1+1)/(test$brg1_esc_Rep1+1), m=25)
test$ep400_cluster <- cut2(test$groseq_tssr_rep2, m=25)

test.ep400 <- test[which(test$ep400_cluster_merged=="high" | test$ep400_cluster_merged=="average"),]

Pol2 pausing is not possible below a certain level of gene expression
Pol2 pausing formula can thus give a false sense of high pausing for many low expressed genes

test.melt <- summaryBy(h2az_esc_Rep1 + h3v3_esc_Rep1 + groseq_tssr_rep1 + groseq_tssr_rep2 + pol2_esc_ser5p_Rep1 +
pol2_esc_ser2p_Rep1 +
 groseq_tssr_rep1_norm + groseq_tssr_rep2_norm + groseq_gb_rep1_norm + groseq_gb_rep2_norm +
 chd4_esc_Rep1 + brg1_esc_Rep1 + atac_esc_wt ~ ep400_cluster, data = test.ep400, FUN = list(median))

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/cor_groseq_h2az_h3v3_esc.pdf")

p1 <- ggscatter(test.melt, x = "groseq_tssr_rep2_norm.median", y = "h2az_esc_Rep1.median", color="#784c97", alpha=0.6,
add = "loess", conf.int = TRUE) + stat_cor(label.x = 0, method="spearman")

p2 <- ggscatter(test.melt, x = "groseq_tssr_rep2_norm.median", y = "h3v3_esc_Rep1.median", color="#f49d58", alpha=0.6,
add = "loess", conf.int = TRUE) + stat_cor(label.x = 0, method="spearman") +
scale_y_continuous(position = "right")

p3 <- ggscatter(test.melt, x = "atac_esc_wt.median", y = "h2az_esc_Rep1.median", color="#784c97", alpha=0.6,
add = "loess", conf.int = TRUE) + stat_cor(method="spearman")

p4 <- ggscatter(test.melt, x = "atac_esc_wt.median", y = "h3v3_esc_Rep1.median", color="#f49d58", alpha=0.6,
add = "loess", conf.int = TRUE) + stat_cor(method="spearman") +
scale_y_continuous(position = "right")

ggarrange(p1, p2, p3, p4, ncol=2, nrow=2)

dev.off()
```


```

#####
Relationship between H2A.Z, H3.3 and Pol2 pausing (MEF)
CODE TO REPRODUCE FIGURES: Figure 6C and Supplementary Figure 7B

```


```

```
#####
```{r}
##-----##
correlation plots between H2A.Z, H3.3 and Pol2 pausing in MEF (ep400 bound)
##-----##

groseq_mef <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/analysis/promoters_mef/allGenes_transcriptSignal.bed",
header=T)
transcript_width <- (groseq_mef$end - groseq_mef$start)
groseq_mef <- groseq_mef[which(groseq_mef$dist_closest_tss > 3000),]
tssr <- t(apply(groseq_mef[,c("groseq_mef_Rep1", "groseq_mef_Rep2", "groseq_mef_Rep3")], 1, function(x) as.numeric(gsub("/.*$", "",
x))))
colnames(tssr) <- c("groseq_tssr_rep1", "groseq_tssr_rep2", "groseq_tssr_rep3")
gb <- t(apply(groseq_mef[,c("groseq_mef_Rep1", "groseq_mef_Rep2", "groseq_mef_Rep3")], 1, function(x) as.numeric(gsub("^.*/", "",
x))))
colnames(gb) <- c("groseq_gb_rep1", "groseq_gb_rep2", "groseq_gb_rep3")
groseq_mef <- cbind(groseq_mef[,c(4),drop=F], tssr, gb)

groseq_mef$groseq_tssr_rep1_norm <- groseq_mef$groseq_tssr_rep1/(groseq_mef$groseq_tssr_rep1+groseq_mef$groseq_gb_rep1)
groseq_mef$groseq_tssr_rep2_norm <- groseq_mef$groseq_tssr_rep2/(groseq_mef$groseq_tssr_rep2+groseq_mef$groseq_gb_rep2)
groseq_mef$groseq_tssr_rep3_norm <- groseq_mef$groseq_tssr_rep3/(groseq_mef$groseq_tssr_rep3+groseq_mef$groseq_gb_rep3)
groseq_mef$groseq_gb_rep1_norm <- groseq_mef$groseq_gb_rep1/(groseq_mef$groseq_tssr_rep1+groseq_mef$groseq_gb_rep1)
groseq_mef$groseq_gb_rep2_norm <- groseq_mef$groseq_gb_rep2/(groseq_mef$groseq_tssr_rep2+groseq_mef$groseq_gb_rep2)
groseq_mef$groseq_gb_rep3_norm <- groseq_mef$groseq_gb_rep3/(groseq_mef$groseq_tssr_rep3+groseq_mef$groseq_gb_rep3)

test <- merge(df_mef, groseq_mef, by.x="name", by.y="name")
test$ratio_chd4_brg1 <- cut2((test$chd4_mef_Rep1+1)/(test$brg1_mef_Rep1+1), m=25)
test$ep400_cluster <- cut2(test$groseq_tssr_rep2, m=25)

test.ep400 <- test[which(test$ep400_cluster_merged=="high" | test$ep400_cluster_merged=="average"),]

Pol2 pausing is not possible below a certain level of gene expression
Pol2 pausing formula can thus give a false sense of high pausing for many low expressed genes
test.melt <- summaryBy(h2az_mef_Rep1 + h3v3_mef_Rep1 + groseq_tssr_rep1 + groseq_tssr_rep2 + groseq_tssr_rep3 +
 pol2_mef_ser5p_Rep1 + pol2_mef_ser2p_Rep1 +
 groseq_tssr_rep1_norm + groseq_tssr_rep2_norm + groseq_tssr_rep3_norm +
 groseq_gb_rep1_norm + groseq_gb_rep2_norm + groseq_gb_rep3_norm +
 chd4_mef_Rep1 + brg1_mef_Rep1 + atac_mef_Rep1 ~ ep400_cluster, data = test.ep400, FUN = list(median))

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/cor_groseq_h2az_h3v3_mef.pdf")

p1 <- ggscatter(test.melt, x = "groseq_tssr_rep2_norm.median", y = "h2az_mef_Rep1.median", color="#784c97", alpha=0.6,
 add = "loess", conf.int = TRUE) + stat_cor(label.x = 0, method="spearman")

p2 <- ggscatter(test.melt, x = "groseq_tssr_rep2_norm.median", y = "h3v3_mef_Rep1.median", color="#f49d58", alpha=0.6,
 add = "loess", conf.int = TRUE) + stat_cor(label.x = 0, method="spearman") +
 scale_y_continuous(position = "right")

```

```

p3 <- ggscatter(test.melt, x = "atac_mef_Repl.median", y = "h2az_mef_Repl.median", color="#784c97", alpha=0.6,
 add = "loess", conf.int = TRUE) + stat_cor(method="spearman")

p4 <- ggscatter(test.melt, x = "atac_mef_Repl.median", y = "h3v3_mef_Repl.median", color="#f49d58", alpha=0.6,
 add = "loess", conf.int = TRUE) + stat_cor(method="spearman") +
 scale_y_continuous(position = "right")

ggarrange(p1, p2, p3, p4, ncol=2, nrow=2)

dev.off()
```

#####
#### CHD4 AND BRG1 BINDING RATIO ANALYSIS (ESC)
#### CODE TO REPRODUCE FIGURES: Figure 3G, Supplementary Figure 4A, Figure 6A
#####
```{r}
test <- df_esc
test[,c("chd4_esc_Repl1", "brg1_esc_Repl1")] <- normalize.quantiles(as.matrix(test[,c("chd4_esc_Repl1", "brg1_esc_Repl1")]))
test[,c("chd4_esc_Repl2", "brg1_esc_Repl2")] <- normalize.quantiles(as.matrix(test[,c("chd4_esc_Repl2", "brg1_esc_Repl2")]))
test$ratio_chd4_brg1 <- cut2((test$chd4_esc_Repl1+1)/(test$brg1_esc_Repl1+1), m=25)
test <- test[order(test$ratio_chd4_brg1),]
test$ratio_ser2p_ser5p <-
as.numeric(normalize.quantiles(as.matrix(test$pol2_esc_ser2p_Repl1))/normalize.quantiles(as.matrix(test$pol2_esc_ser5p_Repl1)))

at Cdk1 promoter, Chd4 and Brg1 binding is at comparable levels
test[grep("^Cdk1$", test$name),c("chd4_esc_Repl1", "brg1_esc_Repl1")]
summary(test[,c("chd4_esc_Repl1", "brg1_esc_Repl1")])

##-----##
plot heatmap for low or bivalent genes (prc2 bound)
##-----##

test.prc2 <- test[which(test$ep400_cluster_merged=="bivalent" | test$ep400_cluster_merged=="low"),]

test.melt <- melt(summaryBy(chd4_esc_Repl1 + brg1_esc_Repl1 ~ ratio_chd4_brg1, data = test.prc2, FUN = list(median)))
p1 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.3)(256))) +
 theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(ezh2_esc_Repl1 ~ ratio_chd4_brg1, data = test.prc2, FUN = list(median)))
p2 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.8)(256))) +
 theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(h3k27me3_esc_Repl1 ~ ratio_chd4_brg1, data = test.prc2, FUN = list(median)))
p3 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=2)(256))) +

```

```

theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(expr_esc ~ ratio_chd4_brg1, data = test.prc2, FUN = list(median)))
p4 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=1)(256))) +
 theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(brg1_fc ~ ratio_chd4_brg1, data = test.prc2, FUN = list(median)))
p5 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=1.4)(256))) +
 theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(chd4_fc ~ ratio_chd4_brg1, data = test.prc2, FUN = list(median)))
p6 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.5)(256))) +
 theme(axis.text.x = element_blank())

g <- arrangeGrob(p1, p2, p3, p4, p5, p6, nrow = 6, ncol=1)
ggsave(g, dpi=300, height=25, width=20, filename=~ /project/chip-seq-analysis/analysis_chd4_brg1/pdf/chd4_brg1_prc2_esc.pdf",
useDingbats=FALSE)

##-----##
plot heatmap for low or bivalent genes (Pol II bound)
##-----##

test.ep400 <- test[which(test$ep400_cluster_merged=="high" | test$ep400_cluster_merged=="average"),]

test.melt <- melt(summaryBy(chd4_esc_Repl + brg1_esc_Repl ~ ratio_chd4_brg1, data = test.prc2, FUN = list(median)))
p1 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.7)(256))) +
 theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(pol2_esc_wt_Repl ~ ratio_chd4_brg1, data = test.ep400, FUN = list(median)))
p2 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.7)(256))) +
 theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(pol2_esc_ser5p_Repl ~ ratio_chd4_brg1, data = test.ep400, FUN = list(median)))
p3 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=1.2)(256))) +
 theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(pol2_esc_ser2p_Repl ~ ratio_chd4_brg1, data = test.ep400, FUN = list(median)))
p4 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.4)(256))) +
 theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(ratio_ser2p_ser5p ~ ratio_chd4_brg1, data = test.ep400, FUN = list(median)))

```

```

p5 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.4)(256))) +
 theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(h3v3_esc_Rep1 ~ ratio_chd4_brg1, data = test.ep400, FUN = list(median)))
p6 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=1.5)(256))) +
 theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(h2az_esc_Rep1 ~ ratio_chd4_brg1, data = test.ep400, FUN = list(median)))
p7 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
 scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=1.5)(256))) +
 theme(axis.text.x = element_blank())

g <- arrangeGrob(p1, p2, p3, p4, p5, p6, p7, nrow = 7, ncol=1)
ggsave(g, dpi=300, height=25, width=20, filename=~ /project/chip-seq-analysis/analysis_chd4_brg1/pdf/chd4_brg1_pol2_esc.pdf",
useDingbats=FALSE)

...

#####
CHD4 AND BRG1 BINDING RATIO ANALYSIS (MEF)
CODE TO REPRODUCE FIGURES: Supplementary Figure 4B, Supplementary Figure 6A, Supplementary Figure 5E
#####
```{r}
test <- df_mef
test[,c("chd4_mef_Rep1", "brg1_mef_Rep1")] <- normalize.quantiles(as.matrix(test[,c("chd4_mef_Rep1", "brg1_mef_Rep1")]))
test$ratio_chd4_brg1 <- cut2((test$chd4_mef_Rep1+1)/(test$brg1_mef_Rep1+1), m=25)
test$ratio_ser2p_ser5p <-
as.numeric(normalize.quantiles(as.matrix(test$pol2_mef_ser2p_Rep1))/normalize.quantiles(as.matrix(test$pol2_mef_ser5p)))
test <- test[order(test$ratio_chd4_brg1),]

##-----##
## plot heatmap for low or bivalent genes (prc2 bound)
##-----##

test.prc2 <- test[which(test$ep400_cluster_merged=="bivalent" | test$ep400_cluster_merged=="low"),]

test.melt <- melt(summaryBy(chd4_mef_Rep1 + brg1_mef_Rep1 ~ ratio_chd4_brg1, data = test.prc2, FUN = list(median)))
p1 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=(value))) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.8)(256))) +
  theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(ezh2_mef_Rep1 ~ ratio_chd4_brg1, data = test.prc2, FUN = list(median)))
p2 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.8)(256))) +
  theme(axis.text.x = element_blank())

```

```

test.melt <- melt(summaryBy(h3k27me3_mef_Repl ~ ratio_chd4_brg1, data = test.prc2, FUN = list(median)))
p3 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.9)(256))) +
  theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(expr_mef ~ ratio_chd4_brg1, data = test.prc2, FUN = list(median)))
p4 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=1)(256))) +
  theme(axis.text.x = element_blank())

g <- arrangeGrob(p1, p2, p3, p4, nrow = 4, ncol=1)
ggsave(g, dpi=300, height=25, width=20, filename="~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/chd4_brg1_prc2_mef.pdf",
useDingbats=FALSE)

##-----##
## plot heatmap for low or bivalent genes (Pol II bound)
##-----##

test.ep400 <- test[which(test$ep400_cluster_merged=="high" | test$ep400_cluster_merged=="average"),]

test.melt <- melt(summaryBy(chd4_mef_Repl + brg1_mef_Repl ~ ratio_chd4_brg1, data = test.ep400, FUN = list(median)))
p1 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=(value))) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.5)(256))) +
  theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(pol2_mef_wt_Repl ~ ratio_chd4_brg1, data = test.ep400, FUN = list(median)))
p2 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.9)(256))) +
  theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(pol2_mef_ser5p_Repl ~ ratio_chd4_brg1, data = test.ep400, FUN = list(median)))
p3 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=0.8)(256))) +
  theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(h3v3_mef_Repl ~ ratio_chd4_brg1, data = test.ep400, FUN = list(median)))
p4 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=1)(256))) +
  theme(axis.text.x = element_blank())

test.melt <- melt(summaryBy(h2az_mef_Repl ~ ratio_chd4_brg1, data = test.ep400, FUN = list(median)))
p5 <- ggplot(test.melt, aes(ratio_chd4_brg1, variable, fill=log(value))) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(11, "RdBu"), bias=1)(256))) +
  theme(axis.text.x = element_blank())

g <- arrangeGrob(p1, p2, p3, p4, p5, nrow = 5, ncol=1)
ggsave(g, dpi=300, height=25, width=20, filename="~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/chd4_brg1_pol2_mef.pdf",
useDingbats=FALSE)

```

```

```

CHD4 AND BRG1 BINDING FOLD CHANGE AND POL2 PAUSING ANALYSIS (ESC vs MEF)
CODE TO REPRODUCE FIGURES: Supplementary Figure 4C, Supplementary Figure 5G

```{r}  
test <- merge(df_esc[,c("name", "chd4_esc_Rep1", "brg1_esc_Rep1", "ezh2_esc_Rep1", "h3k27me3_esc_Rep1", "expr_esc",  
                        "pol2_esc_ser5p_Rep1", "pol2_esc_ser2p_Rep1", "pol2_esc_wt_Rep1", "h2az_esc_Rep1", "h3v3_esc_Rep1")],  
             df_mef[,c("name", "chd4_mef_Rep1", "brg1_mef_Rep1", "ezh2_mef_Rep1", "h3k27me3_mef_Rep1", "expr_mef",  
                       "pol2_mef_ser5p_Rep1", "pol2_mef_ser2p_Rep1", "pol2_mef_wt_Rep1", "h2az_mef_Rep1", "h3v3_mef_Rep1")],  
             by.x="name", by.y="name")  
test$expr_fc <- log2((test$expr_mef+1)/(test$expr_esc+1))  
test <- test[which(abs(test$expr_fc)>1),]  
  
test[,c("chd4_esc_Rep1", "brg1_esc_Rep1", "chd4_mef_Rep1", "brg1_mef_Rep1")] <-  
normalize.quantiles(as.matrix(test[,c("chd4_esc_Rep1", "brg1_esc_Rep1", "chd4_mef_Rep1", "brg1_mef_Rep1")]))  
test[,c("ezh2_esc_Rep1", "ezh2_mef_Rep1")] <- normalize.quantiles(as.matrix(test[,c("ezh2_esc_Rep1", "ezh2_mef_Rep1")]))  
test[,c("h3k27me3_esc_Rep1", "h3k27me3_mef_Rep1")] <- normalize.quantiles(as.matrix(test[,c("h3k27me3_esc_Rep1",  
"h3k27me3_mef_Rep1")]))  
test[,c("expr_esc", "expr_mef")] <- normalize.quantiles(as.matrix(test[,c("expr_esc", "expr_mef")]))  
test[,c("pol2_esc_ser5p_Rep1", "pol2_mef_ser5p_Rep1")] <- normalize.quantiles(as.matrix(test[,c("pol2_esc_ser5p_Rep1",  
"pol2_mef_ser5p_Rep1")]))  
test[,c("pol2_esc_ser2p_Rep1", "pol2_mef_ser2p_Rep1")] <- normalize.quantiles(as.matrix(test[,c("pol2_esc_ser2p_Rep1",  
"pol2_mef_ser2p_Rep1")]))  
test[,c("pol2_esc_wt_Rep1", "pol2_mef_wt_Rep1")] <- normalize.quantiles(as.matrix(test[,c("pol2_esc_wt_Rep1", "pol2_mef_wt_Rep1")]))  
test[,c("h2az_esc_Rep1", "h2az_mef_Rep1")] <- normalize.quantiles(as.matrix(test[,c("h2az_esc_Rep1", "h2az_mef_Rep1")]))  
test[,c("h3v3_esc_Rep1", "h3v3_mef_Rep1")] <- normalize.quantiles(as.matrix(test[,c("h3v3_esc_Rep1", "h3v3_mef_Rep1")]))  
  
test$ratio_chd4_brg1_mef <- (test$chd4_mef_Rep1/test$brg1_mef_Rep1)  
test$ratio_chd4_brg1_esc <- (test$chd4_esc_Rep1/test$brg1_esc_Rep1)  
  
test$ratio_class <- cut2(log2((test$ratio_chd4_brg1_mef)/(test$ratio_chd4_brg1_esc)), m = 100)  
test$ratio_class <- cut2(log2((test$ratio_chd4_brg1_mef)/(test$ratio_chd4_brg1_esc)), c(-0.7343,0.7362))  
test <- test[order(test$ratio_class),]  
table(test$ratio_class)  
  
## ESC and MEF specific genes ##  
gene_esc <- read.table("~/bric_share/bp_nextgen/09_ALL_PUBLIC/rna-seq/mouse/mef_esc_reprogramming/genes_esc_specific.txt")  
gene_mef <- read.table("~/bric_share/bp_nextgen/09_ALL_PUBLIC/rna-seq/mouse/mef_esc_reprogramming/genes_mef_specific.txt")  
  
test$gene_specificity <- "none"  
test[test$name %in% gene_esc$V1,]$gene_specificity <- "esc"  
test[test$name %in% gene_mef$V1,]$gene_specificity <- "mef"  
table(test$gene_specificity)  
  
test.melt <- melt(test[,c("chd4_mef_Rep1", "chd4_esc_Rep1", "ratio_class")])
```

```

p1 <- ggboxplot(test.melt, x = "variable", y = "value",
  fill = "ratio_class", notch=1, notchwidth = 0.6, yscale="log2",
  facet.by = "ratio_class", legend="none", nrow=3,
  #add = "boxplot", add.params = list(fill = "white"),
  palette = c("#784b97", "#ebebcb", "#ffa054")) +
  xlab("") +
  ylab("CHD4 signal (log2)") +
  stat_compare_means(comparisons = list(c("chd4_mef_Rep1", "chd4_esc_Rep1"))))
test.melt <- melt(test[,c("brg1_mef_Rep1", "brg1_esc_Rep1", "ratio_class")])
p2 <- ggboxplot(test.melt, x = "variable", y = "value",
  fill = "ratio_class", notch=1, notchwidth = 0.6, yscale="log2",
  facet.by = "ratio_class", legend="none", nrow=3,
  #add = "boxplot", add.params = list(fill = "white"),
  palette = c("#784b97", "#ebebcb", "#ffa054")) +
  xlab("") +
  ylab("BRG1 signal (log2)") +
  stat_compare_means(comparisons = list(c("brg1_mef_Rep1", "brg1_esc_Rep1"))))

test.melt <- melt(test[,c("ezh2_mef_Rep1", "ezh2_esc_Rep1", "ratio_class")])
p3 <- ggboxplot(test.melt, x = "variable", y = "value", nrow=3,
  fill = "ratio_class", notch=1, notchwidth = 0.6, yscale="log2",
  facet.by = "ratio_class", legend="none",
  #add = "boxplot", add.params = list(fill = "white"),
  palette = c("#784b97", "#ebebcb", "#ffa054")) +
  xlab("") +
  ylab("EZH2 signal (log2)") +
  stat_compare_means(comparisons = list(c("ezh2_mef_Rep1", "ezh2_esc_Rep1"))))

test.melt <- melt(test[,c("h3k27me3_mef_Rep1", "h3k27me3_esc_Rep1", "ratio_class")])
p4 <- ggboxplot(test.melt, x = "variable", y = "value", nrow=3,
  fill = "ratio_class", notch=1, notchwidth = 0.6, yscale="log2",
  facet.by = "ratio_class", legend="none",
  #add = "boxplot", add.params = list(fill = "white"),
  palette = c("#784b97", "#ebebcb", "#ffa054")) +
  xlab("") +
  ylab("H3K27me3 signal (log2)") +
  stat_compare_means(comparisons = list(c("h3k27me3_mef_Rep1", "h3k27me3_esc_Rep1"))))

test.melt <- melt(test[,c("expr_mef", "expr_esc", "ratio_class")])
p5 <- ggboxplot(test.melt, x = "variable", y = "value",
  fill = "ratio_class", notch=1, notchwidth = 0.6, yscale="log2",
  facet.by = "ratio_class", legend="none", nrow=3,
  #add = "boxplot", add.params = list(fill = "white"),
  palette = c("#784b97", "#ebebcb", "#ffa054")) +
  xlab("") +
  ylab("Expr. signal (log2)") +
  stat_compare_means(comparisons = list(c("expr_mef", "expr_esc"))))

```

```

test.melt <- melt(test[,c("pol2_mef_ser5p_Rep1", "pol2_esc_ser5p_Rep1", "ratio_class")])
p6 <- ggboxplot(test.melt, x = "variable", y = "value",
  fill = "ratio_class", notch=1, notchwidth = 0.6, yscale="log2",
  facet.by = "ratio_class", legend="none", nrow=3,
  #add = "boxplot", add.params = list(fill = "white"),
  palette = c("#784b97", "#ebebeb", "#ffa054")) +
  xlab("") +
  ylab("POL2-Ser5p signal (log2)") +
  stat_compare_means(comparisons = list(c("pol2_mef_ser5p_Rep1", "pol2_esc_ser5p_Rep1"))))

test$pol2_esc_ser2p_Rep1_ratio <- test$pol2_esc_ser2p_Rep1/test$pol2_esc_wt_Rep1
test$pol2_mef_ser2p_Rep1_ratio <- test$pol2_mef_ser2p_Rep1/test$pol2_mef_wt_Rep1
test.melt <- melt(test[,c("pol2_mef_ser2p_Rep1_ratio", "pol2_esc_ser2p_Rep1_ratio", "ratio_class")])
p7 <- ggboxplot(test.melt, x = "variable", y = "value", nrow=3,
  fill = "ratio_class", notch=1, notchwidth = 0.6, yscale="log2",
  facet.by = "ratio_class", legend="none",
  #add = "boxplot", add.params = list(fill = "white"),
  palette = c("#784b97", "#ebebeb", "#ffa054")) +
  xlab("") +
  ylab("POL2-Ser2p/POL2 (log2)") +
  stat_compare_means(comparisons = list(c("pol2_mef_ser2p_Rep1_ratio", "pol2_esc_ser2p_Rep1_ratio"))))

test.melt <- melt(test[,c("h2az_mef_Rep1", "h2az_esc_Rep1", "ratio_class")])
p8 <- ggboxplot(test.melt, x = "variable", y = "value",
  fill = "ratio_class", notch=1, notchwidth = 0.6, yscale="log2",
  facet.by = "ratio_class", legend="none", nrow=3,
  #add = "boxplot", add.params = list(fill = "white"),
  palette = c("#784b97", "#ebebeb", "#ffa054")) +
  xlab("") +
  ylab("H2A.Z signal (log2)") +
  stat_compare_means(comparisons = list(c("h2az_mef_Rep1", "h2az_esc_Rep1"))))

test.melt <- melt(test[,c("h3v3_mef_Rep1", "h3v3_esc_Rep1", "ratio_class")])
p9 <- ggboxplot(test.melt, x = "variable", y = "value",
  fill = "ratio_class", notch=1, notchwidth = 0.6, yscale="log2",
  facet.by = "ratio_class", legend="none", nrow=3,
  #add = "boxplot", add.params = list(fill = "white"),
  palette = c("#784b97", "#ebebeb", "#ffa054")) +
  xlab("") +
  ylab("H3.3 signal (log2)") +
  stat_compare_means(comparisons = list(c("h3v3_mef_Rep1", "h3v3_esc_Rep1"))))

g <- ggarrange(p1, p2, p3, p4, p5, p6, p7, p8, p9, nrow=1, ncol=9)

ggsave(g, dpi=300, filename="~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/analysis_mef_esc_boxplot.pdf", useDingbats=FALSE,
width=15, height=7)

## ESC and MEF specific genes ##

```

```

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/mef_esc_specific_genes.pdf")
par(mfrow=c(1,2))
barplot(table(test[which(test$gene_specificity=="mef"),]$ratio_class)/table(test$ratio_class), ylim=c(0,0.06), main="MEF-specific",
las=2)
barplot(table(test[which(test$gene_specificity=="esc"),]$ratio_class)/table(test$ratio_class), ylim=c(0,0.06), main="ESC-specific",
las=2)
dev.off()
```

#####
Gehre et al. 2020 (H3.3 knock out) & Subramanian et al. 2013 (H2A.Z knock out)
CODE TO REPRODUCE FIGURES: Figure 6G
#####
```{r}
gehre <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/external_data/gehre_2020/gene_de.matrix", header=T)
gehre$gene_class <- "neutral"
gehre[which(gehre$log2FoldChange>0 & gehre$padj<0.05),]$gene_class <- "up"
gehre[which(gehre$log2FoldChange<0 & gehre$padj<0.05),]$gene_class <- "down"
df <- merge(gehre, df_esc[,c("name", "ep400_cluster_merged", "pol2_esc_wt_Repl", "promoter_class_merged", "chd4_fc", "brg1_fc",
"chd2_fc", "chd6_fc", "chd8_fc", "chd9_fc", "brd4_fc")], by.x="external_gene_name", by.y="name")

subramanian <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/external_data/subramanian_2013/pgen.1003725.s007.txt",
header=T, sep="\t")
subramanian <- subramanian[order(subramanian$GeneID, -subramanian$H2A.ZWT_RPKM),]
subramanian <- subramanian[match(unique(subramanian$GeneID), subramanian$GeneID),]
colnames(subramanian)[c(10,11)] <- c("log2FC_H2AZ", "log2FC_H2AZ_AP3")
dim(subramanian)
df <- merge(df, subramanian[,c("GeneID", "log2FC_H2AZ", "log2FC_H2AZ_AP3")], by.x="external_gene_name", by.y="GeneID")

dim(df)
df$ep400_cluster <- cut2(df$pol2_esc_wt_Repl, m=25)

df.summary <- summaryBy(log2FoldChange + chd4_fc + brg1_fc + log2FC_H2AZ + chd2_fc + chd8_fc + brd4_fc + chd9_fc ~ ep400_cluster,
data = df, FUN = list(median))
df.summary <- df.summary[!is.na(df.summary$log2FoldChange.median),]

smoothScatter(df.summary$log2FoldChange.median, df.summary$brg1_fc.median, colramp=colorRampPalette(my.cols),
pch=19, cex=.8, useRaster=T, xlab = "Gene expr. FC (H3.3)", ylab = "Gene expr. FC (Brg1)")
lines(lowess(df.summary$log2FoldChange.median, df.summary$brg1_fc.median), col='red', lty="dashed")
cor(df.summary$log2FoldChange.median, df.summary$brg1_fc.median, use = "complete.obs", method="spearman")
cor.test(df.summary$log2FoldChange.median, df.summary$brg1_fc.median)$p.value

smoothScatter(df.summary$log2FC_H2AZ.median, df.summary$chd4_fc.median, colramp=colorRampPalette(my.cols),
pch=19, cex=.8, useRaster=T, xlab = "Gene expr. FC (H3.3)", ylab = "Gene expr. FC (Chd4)")
lines(lowess(df.summary$log2FoldChange.median, df.summary$chd4_fc.median), col='red', lty="dashed")
cor(df.summary$log2FC_H2AZ.median, df.summary$chd4_fc.median, use = "complete.obs", method = "spearman")
cor.test(df.summary$log2FC_H2AZ.median, df.summary$chd4_fc.median)$p.value

```

```

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf//chd4_brg1_h2az_h3v3_expr.pdf")
pheatmap(cor(df.summary[,c(2:5)], use="complete.obs", method = "spearman"), display_numbers = T, color =
rev(colorBrewer2palette("RdYlBu", bias=0.7)))
df.melt <- melt(df.summary[,c(1:5)])
df.melt$variable <- factor(df.melt$variable, levels=c("brg1_fc.median", "log2FoldChange.median",
                                                    "chd4_fc.median", "log2FC_H2AZ.median"), ordered = T)

ggplot(df.melt, aes(ep400_cluster, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=1.2)(56))) +
  theme(axis.text.x = element_text(size=10, angle=45)) + theme_bw() +
  coord_flip()
dev.off()
```

#####
revision (Pol II ChIP-seq data analysis - brg1)
CODE TO REPRODUCE FIGURES: Figure 4A, Supplementary Figure 6E, 6F
#####
```{r}
## Pol II ser5p
file <- "~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/brg1/pol2_ser5p.deepToolsMatrix.gz"
df <- read.table(pipe(sprintf("zgrep -v '@' %s", file)))

mat <- (df[, -c(1:6)])
library("jsonlite")
h <- scan(file, n=1, sep="\n", what=character())
params <- fromJSON(txt=gsub("@", "", h))

len <- params$sample_boundaries[2]-params$sample_boundaries[1]
mat.lst <- lapply(seq(1,length(params$sample_boundaries)-1), function(x) {
  mat[,c((params$sample_boundaries[x]+1):(params$sample_boundaries[x]+len))]
}))

samples <- unlist(strsplit("BRM014,BRM014,BRM014,DMSO,DMSO,DMSO", ","))
index <- lapply(unique(samples), function(x) grep(x, samples))

mat1 <- mat.lst[[1]]
colnames(mat1) <- seq(1,ncol(mat1))
mat2 <- mat.lst[[2]]
colnames(mat2) <- seq(1,ncol(mat2))

reformatMat <- as.data.frame(mat1-mat2)

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/brg1/pol2_ser5p_heatmap.pdf")
Heatmap(reformatMat, col = circlize::colorRamp2(c(-500, -200, 0, 200, 500), c("#2166ac", "#4393c3", "white", "#d6604d", "#b2182b")),
  cluster_rows=F, cluster_columns=F)
dev.off()

## Pol II ser2p

```

```

file <- "~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/brg1/pol2_ser2p.deepToolsMatrix.gz"
df <- read.table(pipe(sprintf("zgrep -v '@' %s", file)))

mat <- (df[,-c(1:6)])
library("jsonlite")
h <- scan(file, n=1, sep="\n", what=character())
params <- fromJSON(txt=gsub("@", "", h))

len <- params$sample_boundaries[2]-params$sample_boundaries[1]
mat.lst <- lapply(seq(1,length(params$sample_boundaries)-1), function(x) {
  mat[,c((params$sample_boundaries[x]+1):(params$sample_boundaries[x]+len))]
}))

samples <- unlist(strsplit("BRM014,BRM014,BRM014,DMSO,DMSO,DMSO", ","))
index <- lapply(unique(samples), function(x) grep(x, samples))

mat1 <- mat.lst[[1]]
colnames(mat1) <- seq(1,ncol(mat1))
mat2 <- mat.lst[[2]]
colnames(mat2) <- seq(1,ncol(mat2))

reformatMat <- as.data.frame(mat1-mat2)

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/brg1/pol2_ser2p_heatmap.pdf")
Heatmap(reformatMat, col = circlize::colorRamp2(c(-20, -10, 0, 10, 20), c("#2166ac", "#4393c3", "white", "#d6604d", "#b2182b")),
  cluster_rows=F, cluster_columns=F)
dev.off()

## PCA plots
pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/pca_brg1.pdf")
df <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/brg1/pol2_ser2p_replicates_multiBigwigSummary.bedSignal",
header=T)
plotCustomPCA(df[,c(8:13)], n = 6)
ggarrange(matrix2Heatmap(cor(df[,c(8:13)]), method="spearman"), col = "Reds", displayN = T, clusterRows = T, clusterCols = T))

df <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/brg1/pol2_ser5p_replicates_multiBigwigSummary.bedSignal",
header=T)
plotCustomPCA(df[,c(8:13)], n = 6)
ggarrange(matrix2Heatmap(cor(df[,c(8:13)]), method="spearman"), col = "Reds", displayN = T, clusterRows = T, clusterCols = T))
dev.off()
```



```

#####
#### revision (Pol II ChIP-seq data analysis - chd4)
#### CODE TO REPRODUCE FIGURES: Figure 5A, Supplementary Figure 6G, 6H
#####
```{r}
Pol II ser5p

```


```

```

file <- "~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/chd4/pol2_ser5p.deepToolsMatrix.gz"
df <- read.table(pipe(sprintf("zgrep -v '@' %s", file)))

mat <- (df[, -c(1:6)])
library("jsonlite")
h <- scan(file, n=1, sep="\n", what=character())
params <- fromJSON(txt=gsub("@", "", h))

len <- params$sample_boundaries[2]-params$sample_boundaries[1]
mat.lst <- lapply(seq(1,length(params$sample_boundaries)-1), function(x) {
  mat[,c((params$sample_boundaries[x]+1):(params$sample_boundaries[x]+len))]
}))

samples <- unlist(strsplit("1_shRNA,1_shRNA,1_shRNA,2_SCR,2_SCR,2_SCR", ","))
index <- lapply(unique(samples), function(x) grep(x, samples))

mat1 <- mat.lst[[1]]
colnames(mat1) <- seq(1,ncol(mat1))
mat2 <- mat.lst[[2]]
colnames(mat2) <- seq(1,ncol(mat2))

reformatMat <- as.data.frame(mat1-mat2)

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/chd4/pol2_ser5p_heatmap.pdf")
Heatmap(reformatMat, col = circlize::colorRamp2(c(-2000, -1000, 0, 1000, 2000), c("#2166ac", "#4393c3", "white", "#d6604d",
"#b2182b")),
  cluster_rows=F, cluster_columns=F)
dev.off()

## Pol II ser2p
file <- "~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/chd4/pol2_Ser2p_genebody.deepToolsMatrix.gz"
df <- read.table(pipe(sprintf("zgrep -v '@' %s", file)))

mat <- (df[, -c(1:6)])
library("jsonlite")
h <- scan(file, n=1, sep="\n", what=character())
params <- fromJSON(txt=gsub("@", "", h))

len <- params$sample_boundaries[2]-params$sample_boundaries[1]
mat.lst <- lapply(seq(1,length(params$sample_boundaries)-1), function(x) {
  mat[,c((params$sample_boundaries[x]+1):(params$sample_boundaries[x]+len))]
}))

samples <- unlist(strsplit("1_shRNA,1_shRNA,1_shRNA,2_SCR,2_SCR,2_SCR", ","))
index <- lapply(unique(samples), function(x) grep(x, samples))

mat1 <- mat.lst[[1]]
colnames(mat1) <- seq(1,ncol(mat1))

```

```

mat2 <- mat.lst[[2]]
colnames(mat2) <- seq(1,ncol(mat2))

reformatMat <- as.data.frame(mat1-mat2)

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/chd4/pol2_ser2p_heatmap.pdf")
Heatmap(reformatMat, col = circlize::colorRamp2(c(-20, -10, 0, 300, 400), c("#2166ac", "white", "white", "white", "#b2182b")),
        cluster_rows=F, cluster_columns=F)
dev.off()

## PCA plots
pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/pca_chd4.pdf")
df <- read.table("~/project/chip-seq-
analysis/analysis_chd4_brg1/deepPlot/chd4/pol2_Ser2p_genebody_replicates_multiBigwigSummary.bedSignal", header=T)
plotCustomPCA(df[,c(8:13)], n = 6)
ggarrange(matrix2Heatmap(cor(df[,c(8:13)]), method="spearman"), col = "Reds", displayN = T, clusterRows = T, clusterCols = T))

df <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/chd4/pol2_ser5p_replicates_multiBigwigSummary.bedSignal",
header=T)
plotCustomPCA(df[,c(8:13)], n = 6)
ggarrange(matrix2Heatmap(cor(df[,c(8:13)]), method="spearman"), col = "Reds", displayN = T, clusterRows = T, clusterCols = T))
dev.off()
```

#####
histone variants edgeR analysis
CODE TO REPRODUCE FIGURES: Figure 6E, 6F, Supplementary Figure 7F, 7G
#####
```{r}
## Chd4
chd4_h2az <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/megadepth/chd4_h2az.signal")
colnames(chd4_h2az) <- c("chr", "start", "end", "shRNA1", "shRNA3", "SCR1", "SCR3")
df <- read.table(pipe("less ~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/REGIONS_INTEREST.BED | cut -f 1-3,4"), header=F)
colnames(df) <- c("chr", "start", "end", "gene")
chd4_h2az$gene <- df$gene

chd4_h2az <- merge(chd4_h2az, df_esc[,c("name", "chd4_esc_Rep1")], by.x="gene", by.y="name")

chd4_h2az$class_chd4 <- cut(chd4_h2az$chd4_esc_Rep1, breaks = c(0, 0.45, 0.65, 0.75, 0.85, 1.2, 100))
table(chd4_h2az$class_chd4)
chd4_h2az$shRNA <- rowMeans(chd4_h2az[,c("shRNA1", "shRNA3")])
chd4_h2az$SCR <- rowMeans(chd4_h2az[,c("SCR1", "SCR3")])
chd4_h2az$value <- chd4_h2az$shRNA - chd4_h2az$SCR

df.melt <- melt((chd4_h2az[,c("shRNA", "SCR", "class_chd4")]))
df.melt$variable <- factor(df.melt$variable, levels=c("SCR", "shRNA"), ordered=T)
p1 <- ggline(df.melt, x = "class_chd4", y = "value", color="variable", add = "mean_se", error.plot = "errorbar") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1)) +

```

```

scale_color_manual(values=c("#b2182b", "#2166ac"))
p2 <- ggerrorplot(df.melt, x = "variable", y = "value", color="variable", facet.by = "class_chd4") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1)) +
  stat_compare_means(comparisons = list( c("SCR", "shRNA")), method.args = list(c(alternative = "g")),
    label.y=c(-70, -80, -90, -100)) +
  scale_color_manual(values=c("#b2182b", "#2166ac"))

## Brg1
brg1_h3v3 <- read.table("~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/megadepth/brg1_h3v3.signal")
colnames(brg1_h3v3) <- c("chr", "start", "end", "BRM014_1", "BRM014_2", "DMSO2", "DMSO3")
df <- read.table(pipe("less ~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/REGIONS_INTEREST.BED | cut -f 1-3,4"), header=F)
colnames(df) <- c("chr", "start", "end", "gene")
brg1_h3v3$gene <- df$gene

brg1_h3v3 <- merge(brg1_h3v3, df_esc[,c("name", "brg1_esc_Rep1")], by.x="gene", by.y="name")

brg1_h3v3$class_brg1 <- cut(brg1_h3v3$brg1_esc_Rep1, breaks = c(0, 0.4, 0.5, 0.7, 0.8, 1, 100))
table(brg1_h3v3$class_brg1)
brg1_h3v3$BRM014 <- rowMeans(brg1_h3v3[,c("BRM014_1", "BRM014_2")])
brg1_h3v3$DMSO <- rowMeans(brg1_h3v3[,c("DMSO2", "DMSO3")])
brg1_h3v3$value <- brg1_h3v3$BRM014 - brg1_h3v3$DMSO

df.melt <- melt((brg1_h3v3[,c("BRM014", "DMSO", "class_brg1")]))
df.melt$variable <- factor(df.melt$variable, levels=c("DMSO", "BRM014"), ordered=T)
p3 <- ggline(df.melt, x = "class_brg1", y = "value", color="variable", add = "mean_se", error.plot = "errorbar") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1)) +co
  scale_color_manual(values=c("#b2182b", "#2166ac"))
p4 <- ggerrorplot(df.melt, x = "variable", y = "value", color="variable", facet.by = "class_brg1") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=1)) +
  stat_compare_means(comparisons = list( c("DMSO", "BRM014") ), , method.args = list(c(alternative = "g")),
    label.y=c(-70, -80, -90, -100)) +
  scale_color_manual(values=c("#b2182b", "#2166ac"))

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/deepPlot/histone_variants/histone_variants_diff_per_remodeler_signal.pdf")
ggarrange(p1, p2, p3, p4, nrow=2, ncol=2, labels = c("A) H2A.Z (CHD4)", "B) H2A.Z (CHD4)", "C) H3.3 (BRG1)", "D) H3.3 (BRG1)")
dev.off()

## PCA plots
pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/pca_histone_variants.pdf")
df <- read.table("~/project/chip-seq-
analysis/analysis_chd4_brg1/deepPlot/histone_variants/brg1_h3v3_replicates_multiBigwigSummary.bedSignal", header=T)
plotCustomPCA(df[,c(8:11)], n = 4)
ggarrange(matrix2Heatmap(cor(df[,c(8:11)]), method="spearman"), col = "Reds", displayN = T, clusterRows = T, clusterCols = T))

df <- read.table("~/project/chip-seq-
analysis/analysis_chd4_brg1/deepPlot/histone_variants/brg1_h2az_replicates_multiBigwigSummary.bedSignal", header=T)
plotCustomPCA(df[,c(8:11)], n = 4)
ggarrange(matrix2Heatmap(cor(df[,c(8:11)]), method="spearman"), col = "Reds", displayN = T, clusterRows = T, clusterCols = T))

```

```

df <- read.table("~/project/chip-seq-
analysis/analysis_chd4_brg1/deepPlot/histone_variants/chd4_h3v3_replicates_multiBigwigSummary.bedSignal", header=T)
plotCustomPCA(df[,c(8:11)], n = 4)
ggarrange(matrix2Heatmap(cor(df[,c(8:11)]), method="spearman"), col = "Reds", displayN = T, clusterRows = T, clusterCols = T))

df <- read.table("~/project/chip-seq-
analysis/analysis_chd4_brg1/deepPlot/histone_variants/chd4_h2az_replicates_multiBigwigSummary.bedSignal", header=T)
plotCustomPCA(df[,c(8:11)], n = 4)
ggarrange(matrix2Heatmap(cor(df[,c(8:11)]), method="spearman"), col = "Reds", displayN = T, clusterRows = T, clusterCols = T))
dev.off()
```

#####
Nascent RNA-seq data analysis (load data)
#####
```{r}
## load raw read counts mapping to mouse genes (mm10)
df_nascent <- read.table("~/bric_share/bp_nextgen/01_ALL_RNA-SEQ_RAW/CHD4_BRG1/map_trim_reads/GENE_EXPRESSION.MATRIX.wo_coor",
header=T)
mat_nascent <- df_nascent[,c(-1)]
row.names(mat_nascent) <- df_nascent$name
# ggarrange(matrix2Heatmap(cor(mat_nascent, method="spearman"), displayN = T, clusterRows = T, clusterCols = T))

## load raw read counts mapping to ERCC controls
df_nascent_ercc <- read.table("~/bric_share/bp_nextgen/01_ALL_RNA-
SEQ_RAW/CHD4_BRG1/map_trim_reads/GENE_EXPRESSION.MATRIX.ercc.wo_coor", header=T)
df_nascent_ercc <- df_nascent_ercc[which(df_nascent_ercc$name %in% c("ERCC-00043", "ERCC-00170", "ERCC-00136", "ERCC-00145", "ERCC-
00092", "ERCC-00002")),]
mat_nascent_ercc <- df_nascent_ercc[,c(-1)]
row.names(mat_nascent_ercc) <- df_nascent_ercc$name
colnames(mat_nascent_ercc) <- colnames(mat_nascent)
```

#####
Nascent RNA-seq data analysis (spike-in - Brg1 - Volcano plots)
CODE TO REPRODUCE FIGURES: Supplementary Figure 1B, 1C, 1G, Figure 4D (barplot)
#####
```{r}
library("EDASeq")
library("RUVSeq")
library("edgeR")

# organize matrix containing raw read counts mapping to mouse genes and ERCC controls
# filter genes with less than 5 reads in at least 2 samples
mat <- rbind(mat_nascent, mat_nascent_ercc[which(row.names(mat_nascent_ercc) %in% c("ERCC-00043", "ERCC-00136", "ERCC-00092")),])
mat <- mat[which(apply(mat[,c(1:2,5:6)], 1, function(x) length(x[x>=5])>=2)),c(1:2,5:6)]
genes <- rownames(mat)[grep("^ERCC-", invert=TRUE, rownames(mat))]

```

```

spikes <- rownames(mat)[grep("^ERCC-", rownames(mat))]

# create EDASeq Object
x <- as.factor(c(rep("KO", 2), rep("Ctrl", 2)))
set <- newSeqExpressionSet(as.matrix(mat), phenoData = data.frame(x, row.names=colnames(mat)))

# boxplots of relative log expression and PCE before between-sample normalization
library(RColorBrewer)
colors <- brewer.pal(3, "Set2")
plotRLE(set, outline=FALSE, ylim=c(-4,4), col=colors[x])
plotPCA(set, col=colors[x], cex=1.2)

# between lane normalization using EDASeq
set <- betweenLaneNormalization(set, which = "upper")
plotRLE(set, outline=FALSE, ylim=c(-4,4), col=colors[x])
plotPCA(set, col=colors[x], cex=1.2)

# spike-in normalization (estimating factors of unwanted variation (technical variation) using ERCC spike-ins)
set1 <- RUVg(set, spikes, k=1)
pData(set1)
plotRLE(set1, outline=FALSE, ylim=c(-4,4), col=colors[x])
plotPCA(set1, col=colors[x], cex=1.2, lty=1)

# use weight factors (W_1) computed using ERCC spike-in to normalize raw read counts
design <- model.matrix(~x + W_1, data = pData(set1))
y <- DGEList(counts=counts(set1), group=x)

# identify differentially expressed genes using edgeR
y <- calcNormFactors(y, method="upperquartile")
y <- estimateGLMCommonDisp(y, design)
y <- estimateGLMTagwiseDisp(y, design)
fit <- glmFit(y, design)
lrt <- glmLRT(fit, coef=2)
plotMD(lrt, ylab = "log2 Fold Change", main = "4sU-seq: Ctrl vs BRM014", adjust.method = "BH", p.value = 0.05, cex = 0.5,
hl.col=c("green3","red"))
res_brg1 <- as.data.frame(topTags(lrt, n=Inf, sort.by="logFC", p.value = 1))
res_brg1$class <- "neutral"
if(nrow(res_brg1[which(res_brg1$FDR<0.05 & res_brg1$logFC>0),])>0) {
  res_brg1[which(res_brg1$FDR<0.05 & res_brg1$logFC>0),]$class <- "up"
}
if(nrow(res_brg1[which(res_brg1$FDR<0.05 & res_brg1$logFC<0),])>0) {
  res_brg1[which(res_brg1$FDR<0.05 & res_brg1$logFC<0),]$class <- "down"
}
summary(decideTests(lrt, adjust.method = "BH", p.value = 0.05))
table(res_brg1$class)

normcounts_brg1 <- cpm(y, normalized.lib.sizes = T)
normcounts_brg1 <- merge(normcounts_brg1, res_brg1, by.x="row.names", by.y="row.names")

```

```

write.table(res_brg1,
            "~/bric_share/bp_nextgen/01_ALL_RNA-SEQ_RAW/CHD4_BRG1/map_trim_reads/GENE_EXPRESSION.MATRIX.DE_BRG1", quote = F,
            sep="\t",
            row.names=T, col.names = T)

## plot barplot showing changes in Cdk1 expression
normcounts_brg1[grep("^Cdk1$", normcounts_brg1$Row.names),]
dirk_data_analyzed[grep("^Cdk1$", dirk_data_analyzed$SYMBOL),c("SYMBOL", "log2FoldChange", "class")]

df.melt <- melt(normcounts_brg1[grep("^Cdk1$", normcounts_brg1$Row.names),][1,c(2:5)])
df.melt$variable <- gsub("_r.*", "", as.character(df.melt$variable))
df.melt$variable <- factor(df.melt$variable, levels=c("DMSO", "BRM014"), ordered=T)
pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/cdk1_nascentRNA_brg1.pdf")
ggbarplot(df.melt, x = "variable", y = "value", add = "mean_se", label = TRUE) + xlab("") + ylab("Gene expr. (Cdk1)")
dev.off()

##-----##
## volcano plot analysis (Supplementary Figure 1B, 1C and 1G)
##-----##

test_brg1 <- merge(res_brg1, df_esc, by.x="row.names", by.y="name")
test_brg1$sep400_cluster_merged <- factor(test_brg1$sep400_cluster_merged, levels=c("bivalent", "low", "average", "high"),
                                         ordered=T)
test_brg1$brg1_class_spikeIn <- "0"
test_brg1[which(test_brg1$logFC > 0.41),]$brg1_class_spikeIn <- as.character(cut2(test_brg1[which(test_brg1$logFC > 0.41),]$logFC,
g=3))
test_brg1[which(test_brg1$logFC < -0.41),]$brg1_class_spikeIn <- as.character(cut2(test_brg1[which(test_brg1$logFC < -0.41),]$logFC,
g=3))
test_brg1$brg1_class_spikeIn <- as.factor(as.numeric(gsub("]", "", gsub("[)]+", "", gsub("^.*", "", 
test_brg1$brg1_class_spikeIn))))
table(test_brg1$brg1_class_spikeIn)

p1 <- ggplot(test_brg1, aes(x=logFC, y=-log10(FDR))) +
  geom_point(aes(colour = logFC, fill = logFC), shape = 21, size = 3, stroke = 1) +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(9, "PuOr"), bias=1)(256))) +
  scale_color_gradientn(colours = rev(colorRampPalette(brewer.pal(9, "PuOr"), bias=1)(256))) +
  theme(legend.position="none") + theme_classic() +
  geom_vline(xintercept=c(-0.41, 0.41), linetype = 2) +
  coord_flip() + theme(legend.position="top")

test_brg1.melt <- melt(summaryBy(pol2_esc_wt_Repl ~ brg1_class_spikeIn, data = test_brg1, FUN = list(median)))
p2 <- ggplot(test_brg1.melt, aes(brg1_class_spikeIn, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=1.4)(56))) +
  theme(axis.text.x = element_text(size=10, angle=45)) + theme_bw() +
  coord_flip() + ylab("") + xlab("") + theme(legend.position="top")

test_brg1.melt <- melt(summaryBy(ezh2_esc_Repl ~ brg1_class_spikeIn, data = test_brg1, FUN = list(median)))
p3 <- ggplot(test_brg1.melt, aes(brg1_class_spikeIn, variable, fill=value)) + geom_tile() +

```

```

scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=0.5)(56))) +
theme(axis.text.x = element_text(size=10, angle=45)) + theme_bw() +
coord_flip() + ylab("") + xlab("") + theme(legend.position="top")

g1 <- ggarrange(p1, ggarrange(p2, p3, nrow=1, ncol=2), nrow=1, ncol=2)

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/nascentRNA_brg1.pdf")
g1
test <- test_brg1
test$ep400_cluster <- cut2(test$pol2_esc_wt_Repl, m=25)
test.melt <- summaryBy(brg1_fc + logFC ~ ep400_cluster, data = test, FUN = list(median))
smoothScatter(test.melt$brg1_fc.median, test.melt$logFC.median, colramp=colorRampPalette(my.cols), pch=19, cex=.8, useRaster=T, xlab
= "Brg1 KD (logFC)", ylab = "Brg1 BRM014 (logFC; spike-in)")
lines(lowess(test.melt$brg1_fc.median, test.melt$logFC.median),col='red', lty="dashed")
cor.test(test.melt$brg1_fc.median, test.melt$logFC.median, method="spearman")
ggscatter(test.melt, x = "brg1_fc.median", y = "logFC.median", color="#784c97", alpha=0.6,
  add = "reg.line", conf.int = TRUE) + stat_cor(label.x = 0, method="spearman") +
  xlab("Brg1 KD (logFC)") + ylab("Brg1 BRM014 (logFC; spike-in)")

test <- merge(res_brg1, dirk_data_analyzed[,c("SYMBOL", "log2FoldChange")], by.x="row.names", by.y="SYMBOL")
test$ep400_cluster <- cut2(test$logCPM, m=25)
test.melt <- summaryBy(log2FoldChange + logFC ~ ep400_cluster, data = test, FUN = list(median))
smoothScatter(test.melt$log2FoldChange.median, test.melt$logFC.median, colramp=colorRampPalette(my.cols), pch=19, cex=.8,
useRaster=T, xlab = "Brg1 BRM014 (log2FC)", ylab = "Brg1 BRM014 (logFC, spike-in)")
lines(lowess(test.melt$log2FoldChange.median, test.melt$logFC.median),col='red', lty="dashed")
cor.test(test.melt$log2FoldChange.median, test.melt$logFC.median, method="spearman")
ggscatter(test.melt, x = "log2FoldChange.median", y = "logFC.median", color="#784c97", alpha=0.6,
  add = "reg.line", conf.int = TRUE) + stat_cor(label.x = 0, method="spearman") +
  xlab("Brg1 BRM014 (log2FC)") + ylab("Brg1 BRM014 (logFC, spike-in)") +
  geom_hline(yintercept = 0, lty=2) + geom_vline(xintercept = 0, lty=2)
plotPCA(set1, col=colors[x], cex=1.2, lty=1)
ggarrange(matrix2Heatmap(cor(counts(set1), method="spearman"), clusterRows = T, clusterCols = T, col = "Reds", displayN = T))
dev.off()
```

#####
Nascent RNA-seq data analysis (spike-in - Chd4 - Volcano plots)
CODE TO REPRODUCE FIGURES: Supplementary Figure 1E, 1F and 1G, Figure 5D (barplot)
#####
```{r}
library("EDASeq")
library("RUVSeq")
library("edgeR")

# organize matrix containing raw read counts mapping to mouse genes and ERCC controls
# filter genes with less than 5 reads in at least 2 samples
mat <- rbind(mat_nascent, mat_nascent_ercc[which(row.names(mat_nascent_ercc) %in% c("ERCC-00043", "ERCC-00136", "ERCC-00092")),])
mat <- mat[which(apply(mat[,c(3,4,7,8)], 1, function(x) length(x[x>=5])>=2)),c(3,4,7,8)]

```

```

genes <- rownames(mat)[grep("^ERCC-", invert=TRUE, rownames(mat))]
spikes <- rownames(mat)[grep("^ERCC-", rownames(mat))]

# create EDASeq Object
x <- as.factor(c(rep("KO", 2), rep("Ctrl", 2)))
set <- newSeqExpressionSet(as.matrix(mat), phenoData = data.frame(x, row.names=colnames(mat)))

# boxplots of relative log expression and PCE before between-sample normalization
library(RColorBrewer)
colors <- brewer.pal(3, "Set2")
plotRLE(set, outline=FALSE, ylim=c(-4,4), col=colors[x])
plotPCA(set, col=colors[x], cex=1.2)

# between lane normalization using EDASeq
set <- betweenLaneNormalization(set, which = "upper")
plotRLE(set, outline=FALSE, ylim=c(-4,4), col=colors[x])
plotPCA(set, col=colors[x], cex=1.2)

# spike-in normalization (estimating factors of unwanted variation (technical variation) using ERCC spike-ins)
set1 <- RUVg(set, spikes, k=1)
pData(set1)
plotRLE(set1, outline=FALSE, ylim=c(-4,4), col=colors[x])
plotPCA(set1, col=colors[x], cex=1.2, lty=1)
ggarrange(matrix2Heatmap(cor(counts(set1)), clusterRows = T, clusterCols = T))

# use weight factors (W_1) computed using ERCC spike-in to normalize raw read counts
design <- model.matrix(~x + W_1, data = pData(set1))
y <- DGEList(counts=counts(set1), group=x)

# identify differentially expressed genes using edgeR
y <- calcNormFactors(y, method="upperquartile")
y <- estimateGLMCommonDisp(y, design)
y <- estimateGLMTagwiseDisp(y, design)
fit <- glmFit(y, design)
lrt <- glmLRT(fit, coef=2)
plotMD(lrt, ylab = "log2 Fold Change", main = "4sU-seq: Ctrl vs BRM014", adjust.method = "BH", p.value = 0.05, cex = 0.5,
hl.col=c("green3","red"))
res_chd4 <- as.data.frame(topTags(lrt, n=Inf, sort.by="logFC", p.value = 1))
res_chd4$class <- "neutral"
if(nrow(res_chd4[which(res_chd4$PValue<0.01 & res_chd4$logFC>0),])>0) {
  res_chd4[which(res_chd4$PValue<0.01 & res_chd4$logFC>0),]$class <- "up"
}
if(nrow(res_chd4[which(res_chd4$PValue<0.01 & res_chd4$logFC<0),])>0) {
  res_chd4[which(res_chd4$PValue<0.01 & res_chd4$logFC<0),]$class <- "down"
}
summary(decideTests(lrt, adjust.method = "BH", p.value = 0.05))
table(res_chd4$class)

```

```

normcounts_chd4 <- cpm(y, normalized.lib.sizes = T)
normcounts_chd4 <- merge(normcounts_chd4, res_chd4, by.x="row.names", by.y="row.names")
write.table(res_chd4,
            "~/bric_share/bp_nextgen/01_ALL_RNA-SEQ_RAW/CHD4_BRG1/map_trim_reads/GENE_EXPRESSION.MATRIX.DE_CHD4.1", quote = F,
            sep="\t",
            row.names=T, col.names = T)

## plot barplot showing changes in Cdk1 expression
normcounts_chd4[grep("^Cdk1$", normcounts_chd4$Row.names),]
df.melt <- melt(normcounts_chd4[grep("^Cdk1$", normcounts_chd4$Row.names),][1,c(2:5)])
df.melt$variable <- gsub("_r.*", "", as.character(df.melt$variable))
df.melt$variable <- factor(df.melt$variable, levels=c("ScrambleShRNA", "Chd4ShRNA"), ordered=T)
pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/cdk1_nascentRNA_chd4.pdf")
ggbarplot(df.melt, x = "variable", y = "value",
          add = "mean_se", label = TRUE) + xlab("") + ylab("Gene expr. (Cdk1)")
dev.off()

##-----##
## volcano plot analysis (Supplementary Figure 1E, 1F and 1G)
##-----##

test_chd4 <- merge(res_chd4, df_esc, by.x="row.names", by.y="name")
test_chd4$ep400_cluster_merged <- factor(test_chd4$ep400_cluster_merged, levels=c("bivalent", "low", "average", "high"),
                                         ordered=T)

test_chd4$chd4_class_spikeIn <- "0"
test_chd4[which(test_chd4$logFC > 1.2),]$chd4_class_spikeIn <- as.character(cut2(test_chd4[which(test_chd4$logFC > 1.2),]$logFC,
g=3))
test_chd4[which(test_chd4$logFC < -1.2),]$chd4_class_spikeIn <- as.character(cut2(test_chd4[which(test_chd4$logFC < -1.2),]$logFC,
g=3))
test_chd4$chd4_class_spikeIn <- as.factor(as.numeric(gsub("]", "", gsub("[)]+", "", gsub("^.*", "", 
test_chd4$chd4_class_spikeIn))))
table(test_chd4$chd4_class_spikeIn)

p1 <- ggplot(test_chd4, aes(x=logFC, y=-log10(FDR))) +
  geom_point(aes(colour = logFC, fill = logFC), shape = 21, size = 3, stroke = 1) +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(9, "PuOr"), bias=1)(256))) +
  scale_color_gradientn(colours = rev(colorRampPalette(brewer.pal(9, "PuOr"), bias=1)(256))) +
  theme(legend.position="none") + theme_classic() +
  geom_vline(xintercept=c(-1.2, 1.2), linetype = 2) +
  coord_flip() + theme(legend.position="top")

test_chd4.melt <- melt(summaryBy(pol2_esc_wt_Rep1 ~ chd4_class_spikeIn, data = test_chd4, FUN = list(median)))
p2 <- ggplot(test_chd4.melt, aes(chd4_class_spikeIn, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=1.3)(56))) +
  theme(axis.text.x = element_text(size=10, angle=45)) + theme_bw() +
  coord_flip() + theme(legend.position="top")

test_chd4.melt <- melt(summaryBy(ezh2_esc_Rep1 ~ chd4_class_spikeIn, data = test_chd4, FUN = list(median)))

```

```

p3 <- ggplot(test_chd4.melt, aes(chd4_class_spikeIn, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=0.3)(56))) +
  theme(axis.text.x = element_text(size=10, angle=45)) + theme_bw() +
  coord_flip() + theme(legend.position="top")

g2 <- ggarrange(p1, ggarrange(p2, p3, nrow=1, ncol=2), nrow=1, ncol=2)

pdf("~/project/chip-seq-analysis/analysis_chd4_brg1/pdf/nascentRNA_chd4.pdf", useDingbats=FALSE)
g2
test <- test_chd4
test$ep400_cluster <- cut2(test$pol2_esc_wt_Repl, m=25)
test.melt <- summaryBy(chd4_fc + logFC ~ ep400_cluster, data = test, FUN = list(median))
smoothScatter(test.melt$chd4_fc.median, test.melt$logFC.median, colramp=colorRampPalette(my.cols), pch=19, cex=.8, useRaster=T, xlab
= "Chd4 KD (logFC)", ylab = "Chd4 KD (logFC; spike-in)")
lines(lowess(test.melt$chd4_fc.median, test.melt$logFC.median), col='red', lty="dashed")
cor.test(test.melt$chd4_fc.median, test.melt$logFC.median, method="spearman")
ggscatter(test.melt, x = "chd4_fc.median", y = "logFC.median", color="#784c97", alpha=0.6,
  add = "reg.line", conf.int = TRUE) + stat_cor(label.x = 0, method="spearman") +
  xlab("Chd4 KD (logFC)") + ylab("Chd4 KD (logFC; spike-in)") +
  geom_hline(yintercept = 0, lty=2) + geom_vline(xintercept = 0, lty=2)
plotPCA(set, col=colors[x], cex=1.2, lty=1)
ggarrange(matrix2Heatmap(cor(counts(set1), method="spearman"), clusterRows = T, clusterCols = T, col = "Reds", displayN = T))
dev.off()

## correlation between changes in nascent RNA levels upon Chd4 and Brg1 perurbation
test <- merge(test_brg1[,c("Row.names", "logFC")], test_chd4[,c("Row.names", "logFC", "pol2_esc_wt_Repl")], by="Row.names")
test$ep400_cluster <- cut2(test$pol2_esc_wt_Repl, m=25)
test.melt <- summaryBy(logFC.x + logFC.y ~ ep400_cluster, data = test, FUN = list(median))
smoothScatter(test.melt$logFC.x.median, test.melt$logFC.y.median, colramp=colorRampPalette(my.cols), pch=19, cex=.8, useRaster=T,
xlab = "Brg1 BRM014 (logFC; spike-in)", ylab = "Chd4 shRNA (logFC; spike-in)")
lines(lowess(test.melt$logFC.x.median, test.melt$logFC.y.median), col='red', lty="dashed")
cor.test(test.melt$logFC.x.median, test.melt$logFC.y.median, method="spearman")
```

#####
Nascent RNA-seq data analysis (spike-in - Brg1 and Chd4 - heatmaps)
CODE TO REPRODUCE FIGURES: Figure 1D, Figure 4A, Figure 5A
#####
```{r}
## Figure 1 (Changes in nascent RNA-seq levels)
test <- test_brg1
test$ep400_cluster <- cut2(test$logCPM, m=25)

test.melt <- melt(summaryBy(logFC ~ ep400_cluster + promoter_class_merged, data = test, FUN = list(mean)))
p1 <-ggplot(test.melt, aes(ep400_cluster, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=0.64)(256))) +
  facet_grid(~ promoter_class_merged ) +
  theme(axis.text.x = element_text(size=10, angle=45)) + xlab("BRG1 inhibition (nascent RNA changes)")

```

```

test <- test_chd4
test$ep400_cluster <- cut2(test$pol2_esc_wt_Rep1, m=25)

test.melt <- melt(summaryBy(logFC ~ ep400_cluster + promoter_class_merged, data = test, FUN = list(median)))
p3 <- ggplot(test.melt, aes(ep400_cluster, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=0.9)(56))) +
  facet_grid(.~ promoter_class_merged) +
  theme(axis.text.x = element_text(size=10, angle=45)) + xlab("Chd4 knock-down (nascent RNA changes)")

g <- ggarrange(p1, p3, nrow=2, ncol=1)
g
ggsave(g, dpi=300, height=15, width=18, filename="~/project/chip-seq-
analysis/analysis_chd4_brg1/pdf/nascentRNA_chd4_brg1_heatmap.pdf", useDingbats=FALSE)

## Figure 4A and 5A (side panels - changes in nascent RNA-seq levels)
test <- test_brg1[which(test_brg1$ep400_cluster_merged=="High" | test_brg1$ep400_cluster_merged=="average"),]
test$ep400_cluster <- cut2(test$pol2_esc_wt_Rep1, m=25)
test.melt <- melt(summaryBy(logFC ~ ep400_cluster, data = test, FUN = list(median)))
p1 <- ggplot(test.melt, aes(ep400_cluster, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=1)(56))) +
  theme(axis.text.x = element_text(size=10, angle=45))

test <- test_chd4[which(test_chd4$ep400_cluster_merged=="High" | test_chd4$ep400_cluster_merged=="average"),]
test$ep400_cluster <- cut2(test$pol2_esc_wt_Rep1, m=25)
test.melt <- melt(summaryBy(logFC ~ ep400_cluster, data = test, FUN = list(median)))
#barplot(test.melt[order(test.melt$promoter_class_merged, test.melt$ep400_cluster),]$value)
p2 <- ggplot(test.melt, aes(ep400_cluster, variable, fill=value)) + geom_tile() +
  scale_fill_gradientn(colours = rev(colorRampPalette(brewer.pal(5, "RdBu"), bias=2.5)(56))) +
  theme(axis.text.x = element_text(size=10, angle=45))
g <- ggarrange(p1, p2, nrow=2, ncol=1)
g
ggsave(g, dpi=300, height=15, width=18, filename="~/project/chip-seq-
analysis/analysis_chd4_brg1/pdf/nascentRNA_chd4_brg1_heatmap1.pdf", useDingbats=FALSE)
```

```