Code to produce figures for manuscript

Liz Ing-Simmons

10 November 2014

Code to produce figures for manuscript from processed data

```
out_dir <- "~/HiC/Paper/Figures/"

generate.plot.matrix = function(norm.cov, regions, size){

    plot.matrix = matrix(0, ncol=size*3, nrow=length(regions))

      for( i in 1:length(regions)){
            #print(i)
              #get region data
                chr = as.character(seqnames(regions[i]))
                reg.start = start(regions[i])
                reg.end = end(regions[i])
                reg.width = width(regions[i])

                #get coverage of region and flanking areas
                plot.cov = norm.cov[[chr]][(reg.start-reg.width):(reg.end+reg.width)]

                #calculate splines
                plot.matrix[i,] = spline(1:length(plot.cov), plot.cov, n = size*3)$y
          }
      return(plot.matrix)
  }

superpose.eb <-
  function (x, y, ebl, ebu = ebl, length = 0.08, ...){
    arrows(x, y + ebu, x, y - ebl, angle = 90, code = 3, length = length, ...)
  }

ggpie <- function (dat, by, totals) {
```

```
    ggplot(dat, aes_string(x=factor(1), y=totals, fill=by)) +
        geom_bar(stat='identity', color='black') +
        coord_polar(theta='y') +
        theme(axis.ticks=element_blank(),
            axis.text=element_blank(),
            axis.title=element_blank(),
            panel.grid=element_blank(),
            panel.border=element_blank(),
            legend.position="none") +
        scale_y_continuous(breaks=cumsum(dat[[totals]]) - dat[[totals]] / 2, labels=dat[[by]
        theme_bw()
}

ensGene <- read.table("~/mm9_data/ensGene.txt")
names(ensGene) <- c("bin", "ensGene", "chr", "strand", "txStart", "txEnd",
                    "cdsStart","cdsEnd", "exonCount", "exonStarts", "exonEnds",
                    "score", "name2", "cdsStartStat", "cdsEndStat","exonFrames")
GTP <- read.table("~/mm9_data/ensGtp.txt", sep="\t") #gene/trasncript/protein conversions
names(GTP) <- c("G", "T", "P")

ensGene_tss.gr <- (function(x){
  plus <- x[x$strand=="+",]
  plus$width <- plus$txEnd - plus$txStart
    plus.gr <- GRanges(seqnames=plus$chr,
                       ranges=IRanges(plus$txStart+1, plus$txStart+1),
                       #start coordinates in database are 0-based
                        ensT=plus$ensGene, genelength=plus$width)
    minus <- x[x$strand=="-",]
    minus$width <- minus$txEnd - minus$txStart
    minus.gr <- GRanges(seqnames=minus$chr,
                        ranges=IRanges(minus$txEnd, minus$txEnd),
                        #end coordinates are already 1-based
                        ensT=minus$ensGene, genelength=minus$width)
    tmp <- c(plus.gr,minus.gr)
    tmp <- tmp[seqnames(tmp) %in% paste0("chr", c(1:19, "X", "Y"))]
    })(ensGene)

ensGene_tss.gr$ensG <- GTP$G[match(ensGene_tss.gr$ensT, GTP$T)]


ensGene_bodies.gr <- GRanges(seqnames=ensGene$chr,
                             ranges=IRanges(ensGene$txStart+1, ensGene$txStart+1),
                             #start coordinates in database are 0-based
                                ensT=ensGene$ensGene)
ensGene_bodies.gr$ensG <- GTP$G[match(ensGene_bodies.gr$ensT, GTP$T)]
```

Super-enhancers were defined using ROSE (Whyte et al., 2013) with a transcription start site exclusion zone size of 4kb (-t 2000) and the default stitching size of 12.5kb. H3K27ac peaks called by MACS were used as input constituent enhancers, and input-subtracted H3K27ac ChIP-seq signal was used for ranking the stitched regions.

We define a consensus set of super-enhancers by taking the intersection of regions between two biological replicates for each cell type, and then taking the union of these regions between control and cohesin-deficient cells. The remaining regions from ROSE output are then filtered to remove regions within 2.5kb of a transcription start site and a consensus set of conventional enhancers is defined in the same way as for super-enhancers.

```
data_dir <- "/mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/ChIPSeq/H3K27ac/Aligned/
WT1_enh <- read.table(paste0(data_dir,"DPT_WT1_H3K27ac_mm9_peaks_AllEnhancers.table.txt"),
                      header=T, stringsAsFactors=F)
KO2_enh <- read.table(paste0(data_dir,"DPT_KO2_H3K27ac_mm9_peaks_AllEnhancers.table.txt"),
                      header=T, stringsAsFactors=F)
WT2_enh <- read.table(paste0(data_dir,"DPT_WT2_H3K27ac_mm9_peaks_AllEnhancers.table.txt"),
                      header=T, stringsAsFactors=F)
KO1_enh <- read.table(paste0(data_dir,"DPT_KO1_H3K27ac_mm9_peaks_AllEnhancers.table.txt"),
                      header=T, stringsAsFactors=F)

#transform into GRanges objects

KO1_enh.gr <- makeGRangesFromDataFrame(KO1_enh, keep.extra.columns=TRUE)
KO2_enh.gr <- makeGRangesFromDataFrame(KO2_enh, keep.extra.columns=TRUE)
WT1_enh.gr <- makeGRangesFromDataFrame(WT1_enh, keep.extra.columns=TRUE)
WT2_enh.gr <- makeGRangesFromDataFrame(WT2_enh, keep.extra.columns=TRUE)

gr.list <- list(WT1_enh.gr, WT2_enh.gr, KO1_enh.gr, KO2_enh.gr)
names(gr.list) <- c("WT1", "WT2", "KO1", "KO2")

SE.list <- lapply(gr.list, function(x){
    return(x[x$isSuper==TRUE])
})

nonSE.list <- lapply(gr.list, function(x){
    return(x[x$isSuper==FALSE])
})

#Find enhancers consistent between replicates, give ID based on length ranking
WT_nonSE_both <- BiocGenerics::intersect(nonSE.list$WT1, nonSE.list$WT2)
KO_nonSE_both <- BiocGenerics::intersect(nonSE.list$KO1, nonSE.list$KO2)
join_nonSE <-  BiocGenerics::union(WT_nonSE_both, KO_nonSE_both)
```

```
#remove promoters
promoters <- resize(ensGene_tss.gr, fix="center", width=5000)
join_nonSE <- join_nonSE[-queryHits(findOverlaps(join_nonSE, promoters))]

join_nonSE <- join_nonSE[order(-width(join_nonSE))]
join_nonSE$ID <- 1:length(join_nonSE)

## SEs
WT_both <- BiocGenerics::intersect(SE.list$WT1, SE.list$WT2) #n=372
KO_both <- BiocGenerics::intersect(SE.list$KO1, SE.list$KO2) #n=434

join_SEs <-  BiocGenerics::union(WT_both, KO_both)
join_SEs <- join_SEs[order(-width(join_SEs))]
join_SEs$ID <- 1:length(join_SEs)
```

Developmentally regulated enhancers are taken from (Zhang et al., 2012)[http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE31235]. Peaks for H3K4me2 were called with MACS, for DN3 and DP thymocyte samples. Stable enhancers were defined as those that overlap between DN3 and DP, new enhancers as those which are only present in DP.

```
data_dir <- "/mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/Zhang_data/"

DN3_1 <- read.table(paste0(data_dir,"DN3_H3K4me2_Rep1_sorted_peaks.bed"),
                    sep="\t", col.names=c("chr", "start", "end", "id", "score"))
DN3_2 <- read.table(paste0(data_dir,"DN3_H3K4me2_Rep2_sorted_peaks.bed"),
                    sep="\t", col.names=c("chr", "start", "end", "id", "score"))
DP_1 <- read.table(paste0(data_dir,"DP_H3K4me2_Rep1_sorted_peaks.bed"),
                   sep="\t", col.names=c("chr", "start", "end", "id", "score"))
DP_2 <- read.table(paste0(data_dir,"DP_H3K4me2_Rep2_sorted_peaks.bed"),
                   sep="\t", col.names=c("chr", "start", "end", "id", "score"))

DN3_1.gr <- makeGRangesFromDataFrame(DN3_1)
DN3_2.gr <- makeGRangesFromDataFrame(DN3_2)
DP_1.gr <- makeGRangesFromDataFrame(DP_1)
DP_2.gr <- makeGRangesFromDataFrame(DP_2)

DN3.gr <- BiocGenerics::intersect(DN3_1.gr, DN3_2.gr)
DP.gr <- BiocGenerics::intersect(DP_1.gr, DP_2.gr)

DN3.gr <- DN3.gr[width(DN3.gr) >= 100]
DP.gr <- DP.gr[width(DP.gr) >= 100]

fO <- findOverlaps(DN3.gr, DP.gr)
```

```
DN3_only.gr <- DN3.gr[-queryHits(fO)]
DP_only.gr <- DP.gr[-subjectHits(fO)]

DP_stable.gr <- DP.gr[unique(subjectHits(fO))]
```

Gene expression analysis was performed by Andre Faure using MMSEQ and DESeq, as described in Seitan et al., 2013.

```
#load gene expression data
gene_expr.df <- read.table("~/HiC/data/gene_expression_dataset_df.tsv",
                           sep="\t", header=T, stringsAsFactors=F)
gene_expr.df$ensG <- rownames(gene_expr.df)
rownames(gene_expr.df) <- NULL
#modify as Andre did
#Exclude genes with zero mean in both conditions
gene_expr.df<-subset(gene_expr.df, gene_expr.df$baseMeanA!=0 | gene_expr.df$baseMeanB!=0)
#Add pseudocount to avoid +/-infinite log2 fold changes
gene_expr.df$log2FoldChange_pseudo<-log2((gene_expr.df$baseMeanB+1)/(gene_expr.df$baseMeanA+
#Calculate mean across conditions
gene_expr.df$baseMean<-apply(cbind(gene_expr.df$baseMeanB, gene_expr.df$baseMeanA), 1, mean)
```

'Nearest neighbor' genes are defined by assigning enhancers or super-enhancers to the expressed transcript whose TSS is the nearest to the center of the enhancer. 'Overlapping genes' are those where any part of the gene body overlaps an enhancer or super-enhancer. Genes with a TSS within 40kb of a super-enhancer are also considered.

```
#associate promoters and SEs

#to find nearest expressed gene (>0 in at least one condition)
promoters <- ensGene_tss.gr[ensGene_tss.gr$ensG %in% gene_expr.df$ensG]
#nearest genes to super-enhancer centres (as in Young lab papers)
centres <- resize(join_SEs, fix="center", width=1)
nearest_promoters <- promoters[nearest(centres,promoters)]
nearest_genes_to_SEs <- gene_expr.df[match(nearest_promoters$ensG, gene_expr.df$ensG),]
rownames(nearest_genes_to_SEs) <- NULL
nearest_genes_to_SEs <- unique(nearest_genes_to_SEs)

#genes overlapping SEs
bodies <- ensGene_bodies.gr[ensGene_bodies.gr$ensG %in% gene_expr.df$ensG]
fO <- findOverlaps(bodies, join_SEs)
overlap_SE_bodies <- bodies[queryHits(fO)]
overlap_SE_genes <- gene_expr.df[match(overlap_SE_bodies$ensG, gene_expr.df$ensG),]
rownames(overlap_SE_genes) <- NULL
overlap_SE_genes <- unique(overlap_SE_genes)
```

```r
#remove transcript IDs and collapse
#SE may overlap start of multiple transcripts but expression is at gene level

#genes within 40kb
f0 <- findOverlaps(promoters, join_SEs, maxgap=40000)
within40kb_promoters <- promoters[queryHits(f0)]
within40kb_genes <- gene_expr.df[match(within40kb_promoters$ensG, gene_expr.df$ensG),]
rownames(within40kb_genes) <- NULL
within40kb_genes <- unique(within40kb_genes)

#get enhancer centres and nearest promoters
centres <- resize(join_nonSE, fix="center", width=1)

nearest_promoters <- promoters[nearest(centres,promoters)]
nearest_genes_to_nonSEs <- gene_expr.df[match(nearest_promoters$ensG, gene_expr.df$ensG),]
rownames(nearest_genes_to_nonSEs) <- NULL
nearest_genes_to_nonSEs <- unique(nearest_genes_to_nonSEs)
#extra step here - remove any that are also associated with SEs
nearest_genes_to_nonSEs <- nearest_genes_to_nonSEs[!(nearest_genes_to_nonSEs$ensG %in% neare

#get genes overlapping typical enhancers in the gene body

f0 <- findOverlaps(bodies, join_nonSE)
overlap_bodies <- bodies[queryHits(f0)]
overlap_bodies <- gene_expr.df[match(overlap_bodies$ensG, gene_expr.df$ensG),]
rownames(overlap_bodies) <- NULL
overlap_bodies <- unique(overlap_bodies)

## collect into data frame for plotting
all.df <- rbind(cbind(nearest_genes_to_nonSEs, Group="nearest_enh"),
  cbind(overlap_bodies, Group="overlap_enh"),
  cbind(nearest_genes_to_SEs, Group="nearest_SE"),
  cbind(overlap_SE_genes, Group="overlap_SE"),
  cbind(gene_expr.df, Group="expressed_genes"),
  cbind(within40kb_genes, Group="within40kb"))

all.df$Group <- factor(all.df$Group,
                       levels = c("expressed_genes", "nearest_enh", "overlap_enh",
                                  "within40kb", "overlap_SE", "nearest_SE"))
```

Figure 1A – Regression model

We used a multinomial logistic regression model to predict gene expression changes in cohesin-deficient thymocytes as previously described (Seitan et al., 2013). In addition to the previously used features, we included the variables

'Next to enhancer' (genes that are nearest neighbors of conventional enhancers), 'Near enhancer cluster' (genes positioned within 40kb of an enhancer cluster) and 'Next to enhancer cluster' (genes that are nearest neighbors of super-enhancers).

```
#colours
col_green<-rgb(0, 191, 50, maxColorValue=255)
col_orange<-rgb(255, 191, 0, maxColorValue=255)

col_up <- "grey60"
col_down <- "grey25"

#same data on gene expr as earlier, but slightly different processing
exp_tab <- read.table("~/HiC/data/gene_expression_dataset_df.tsv",
                      sep="\t", header=T, stringsAsFactors=F)
exp_tab$y<-as.factor(exp_tab$DE_up-exp_tab$DE_down)
#Define "0" as reference level
exp_tab$y<-relevel(exp_tab$y, ref="0")
#Scale gene length to [0,1]
temp<-ecdf(exp_tab$log10GeneLength)
exp_tab$log10GeneLength_scale<-temp(exp_tab$log10GeneLength)

#Scale mean expression to [0,1]
temp<-ecdf(exp_tab$baseMeanA + exp_tab$baseMeanB)
exp_tab$meanExpr_scale <- temp(exp_tab$baseMeanA + exp_tab$baseMeanB)


#add more factor columns
exp_tab$nearest_enh <- 0
exp_tab$nearest_enh[rownames(exp_tab) %in% nearest_genes_to_nonSEs$ensG] <- 1
exp_tab$overlap_enh <- 0
exp_tab$overlap_enh[rownames(exp_tab) %in% overlap_bodies$ensG] <- 1
exp_tab$SE_within_40kb <- 0
exp_tab$SE_within_40kb[rownames(exp_tab) %in% within40kb_genes$ensG] <- 1
exp_tab$nearest_SE <- 0
exp_tab$nearest_SE[rownames(exp_tab) %in% nearest_genes_to_SEs$ensG] <- 1
exp_tab$overlap_SE <- 0
exp_tab$overlap_SE[rownames(exp_tab) %in% overlap_SE_genes$ensG] <- 1

#run model
#Build multivariate model (CNC, RAD21+CTCF)
exp_mlrm<-multinom(y ~ prom_CNC+DI_down+DI_up+DI_rand+prom_CTCF+prom_Med1+prom_Nipbl+prom_Ra
#Get coefficients and coefficient significance
exp_mlrm_summary<-summary(exp_mlrm)
exp_mlrm_coef<-exp_mlrm_summary$coefficients[,2:dim(exp_mlrm_summary$coefficients)[2]]
exp_mlrm_se<-exp_mlrm_summary$standard.errors[,2:dim(exp_mlrm_summary$standard.errors)[2]]
exp_mlrm_sig<-exp_mlrm_se
```

```
for(i in 1:dim(exp_mlrm_sig)[2]){
  exp_mlrm_sig[1,i]<-min(pnorm(0, exp_mlrm_coef[1,i], exp_mlrm_se[1,i],
                               lower.tail=T), pnorm(0, exp_mlrm_coef[1,i],
                                                    exp_mlrm_se[1,i], lower.tail=F))*2
  exp_mlrm_sig[2,i]<-min(pnorm(0, exp_mlrm_coef[2,i], exp_mlrm_se[2,i],
                               lower.tail=T), pnorm(0, exp_mlrm_coef[2,i],
                                                    exp_mlrm_se[2,i], lower.tail=F))*2
}

colnames(exp_mlrm_coef) <- c("Prom. Rad21 not CTCF", "Diff. Hi-C interaction (down)",
                             "DI_up","DI_rand", "Promoter CTCF",
                             "Promoter Med1","Promoter Nipbl","Prom. Rad21+CTCF",
                             "Promoter H3K4me3", "Promoter RNAP2",
                             "Gene length (log10)", "Promoter CpG island",
                             "RNAP2_pausing_index", "Next to enhancer",
                             "overlap_enh", "Near SE", "Next to SE", "overlap_SE")
colnames(exp_mlrm_se) <- colnames(exp_mlrm_coef)
colnames(exp_mlrm_sig) <- colnames(exp_mlrm_coef)

exp_mlrm_coef <- exp_mlrm_coef[, c("Promoter CpG island","Gene length (log10)",
                                   "Next to enhancer","Near SE", "Next to SE",
                                   "Diff. Hi-C interaction (down)", "Promoter H3K4me3",
                                   "Promoter RNAP2","Promoter Nipbl",
                                   "Prom. Rad21+CTCF","Promoter Med1",
                                   "Prom. Rad21 not CTCF", "Promoter CTCF")]
exp_mlrm_se <- exp_mlrm_se[,colnames(exp_mlrm_coef)]
exp_mlrm_sig <- exp_mlrm_sig[,colnames(exp_mlrm_coef)]

#plot
par(mar=c(13, 4, 4, 2), cex=1.5)
b<-barplot(exp_mlrm_coef, las=2,
           ylab='Regression model coefficients', beside=T,
           col=c(col_down, col_up), ylim=c(-2.5, 2.5))

superpose.eb(b, exp_mlrm_coef, 1.96*exp_mlrm_se, lwd=1)

text(b, exp_mlrm_coef + ifelse(exp_mlrm_coef > 0, 1, -1)*exp_mlrm_se*3,
     labels=c('', '*')[(exp_mlrm_sig<0.05 & exp_mlrm_sig>=0.01)+1], srt=90)

text(b, exp_mlrm_coef + ifelse(exp_mlrm_coef > 0, 1, -1)*exp_mlrm_se*3,
     labels=c('', '**')[(exp_mlrm_sig<0.01 & exp_mlrm_sig>=0.001)+1], srt=90)

text(b, exp_mlrm_coef + ifelse(exp_mlrm_coef > 0, 1, -1)*exp_mlrm_se*3,
     labels=c('', '***')[(exp_mlrm_sig<0.001)+1], srt=90)

legend("topright", legend=c('DE (down-regulated)', 'DE (up-regulated)'),
```

```
       fill=c(col_down, col_up), bty="n")
```

Figure1B – Do enhancers explain DE genes?

```
all.df$DE <- "No"
all.df$DE[all.df$DE_up == TRUE] <- "Up"
all.df$DE[all.df$DE_down == TRUE] <- "Down"
all.df$DE <- factor(all.df$DE, levels = c("Down", "Up","No"))

enh_assoc_df <- all.df

## down reg genes
enh_assoc_df %>% filter(DE=="Down") %>%
    filter(Group %in% c("within40kb", "overlap_SE", "nearest_SE")) %>%
    dplyr::select(ensG) %>% unique() %>% unlist() -> down_SE_genes
enh_assoc_df %>% filter(DE=="Down") %>%
    filter(Group %in% c("nearest_enh","overlap_enh")) %>%
    dplyr::select(ensG) %>% unique() %>% unlist() -> down_Enhancer_genes
enh_assoc_df %>% filter(DE=="Down") %>%
    filter(Group=="expressed_genes") %>%
    dplyr::select(ensG) %>% unique() %>% unlist() -> down_All_genes

down_Enhancer_genes <- down_Enhancer_genes[!(down_Enhancer_genes %in% down_SE_genes)]
down_All_genes <- down_All_genes[!((down_All_genes %in% down_SE_genes) | (down_All_genes %in

down_pie <- c(Enhancer=length(down_Enhancer_genes), SE=length(down_SE_genes), Other=length(d

dat <- data.frame(num=down_pie, category=names(down_pie), labels=down_pie)
dat$ymax = cumsum(dat$num)
dat$ymin = c(0, head(dat$ymax, n=-1))
dat$label <- dat$num

ggplot(dat, aes(fill=category, ymax=ymax, ymin=ymin, xmax=4, xmin=0, label=label)) +
    geom_rect(colour="black") +
    geom_text(aes(x=3, y=(ymax+ymin)/2), size=10) +
    coord_polar(theta="y") +
    xlim(c(0, 4))+
    theme_bw() +
    scale_fill_manual(values=c(Enhancer=col_up, SE=col_down, Other="white")) +
    theme(panel.grid=element_blank()) +
    theme(axis.text=element_blank()) +
    theme(axis.ticks=element_blank()) +
    theme(axis.title=element_blank()) +
    theme(panel.border=element_blank()) +
    theme(legend.position="none")
```

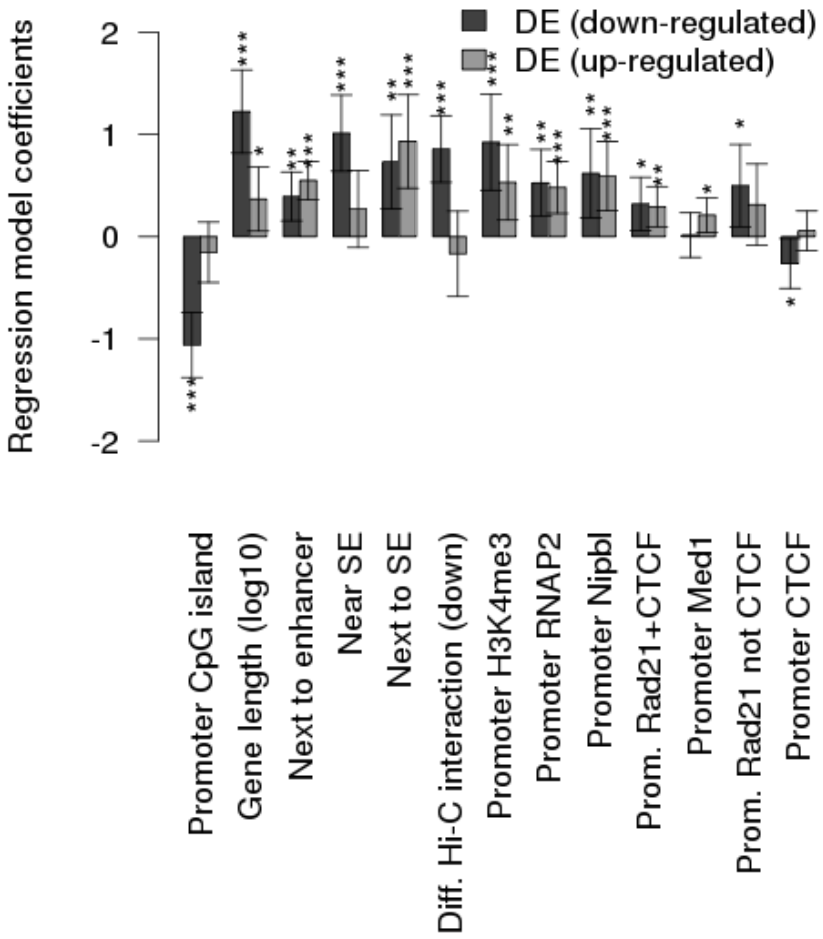

Figure 1: plot of chunk regression_model

Figure 2: plot of chunk unnamed-chunk-1

```
## up reg genes
enh_assoc_df %>% filter(DE=="Up") %>%
    filter(Group %in% c("within40kb", "overlap_SE", "nearest_SE")) %>%
    dplyr::select(ensG) %>% unique() %>% unlist() -> Up_SE_genes
enh_assoc_df %>% filter(DE=="Up") %>%
    filter(Group %in% c("nearest_enh","overlap_enh")) %>%
    dplyr::select(ensG) %>% unique() %>% unlist() -> Up_Enhancer_genes
enh_assoc_df %>% filter(DE=="Up") %>%
    filter(Group=="expressed_genes") %>% dplyr::select(ensG) %>% unique() %>% unlist() -> Up

Up_Enhancer_genes <- Up_Enhancer_genes[!(Up_Enhancer_genes %in% Up_SE_genes)]
Up_All_genes <- Up_All_genes[!((Up_All_genes %in% Up_SE_genes) | (Up_All_genes %in% Up_Enhan

Up_pie <- c(Enhancer=length(Up_Enhancer_genes), SE=length(Up_SE_genes), Other=length(Up_All_
```

```
dat <- data.frame(num=Up_pie, category=names(Up_pie), labels=Up_pie)
dat$ymax = cumsum(dat$num)
dat$ymin = c(0, head(dat$ymax, n=-1))
dat$label <- dat$num

ggplot(dat, aes(fill=category, ymax=ymax, ymin=ymin, xmax=4, xmin=0, label=label)) +
    geom_rect(colour="black") +
    geom_text(aes(x=3, y=(ymax+ymin)/2), size=10) +
    coord_polar(theta="y") +
    xlim(c(0, 4))+
    theme_bw() +
    scale_fill_manual(values=c(Enhancer=col_up, SE=col_down, Other="white")) +
    theme(panel.grid=element_blank()) +
    theme(axis.text=element_blank()) +
    theme(axis.ticks=element_blank()) +
    theme(axis.title=element_blank()) +
    theme(panel.border=element_blank()) +
    theme(legend.position="none")
```

Figure 1C: Genes near enhancers are more likely to be deregulated

```
##Plot % genes nonDE, Up, or Down
all.df %>%
  dplyr::select(ensG, DE_up, DE_down, DE, Group) %>%
  group_by(Group, DE) %>%
  summarise(count=n()) %>%
  ungroup() %>% ##dplyr bug :(
  spread(DE, count) -> DE_mat

DE_mat <- mutate(DE_mat, Percentage=round(100*(Up+Down)/(Up+Down+No)))

cols <- c("No" = NA, "Up" = col_up, "Down" = col_down)
labs <- c("Genome average", "Next to enhancer", "Overlap enhancer", "Near SE", "Overlap SE",
labs<- rev(paste0(labs, ": ", DE_mat$Percentage, "%"))

ggplot(all.df, aes(x=Group, fill=DE)) + geom_bar(position="fill") +
  scale_y_continuous("",labels=percent,limits=c(0,0.35)) +
  scale_x_discrete("", labels=labs, limits=c("nearest_SE", "overlap_SE",
                                             "within40kb", "overlap_enh", "nearest_enh", "ex
  scale_fill_manual(values=cols) +
  theme_bw(base_size=20) + theme(panel.border=element_blank()) + coord_flip()
```

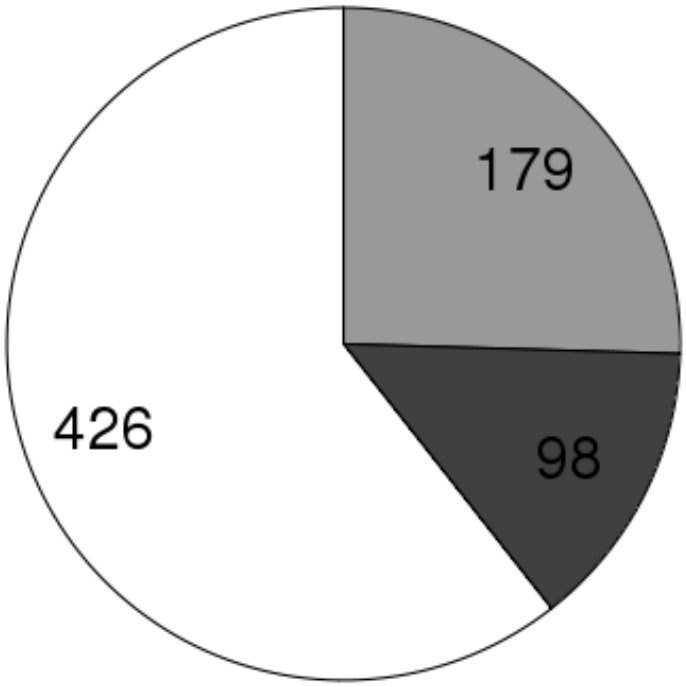

Figure 3: plot of chunk unnamed-chunk-1

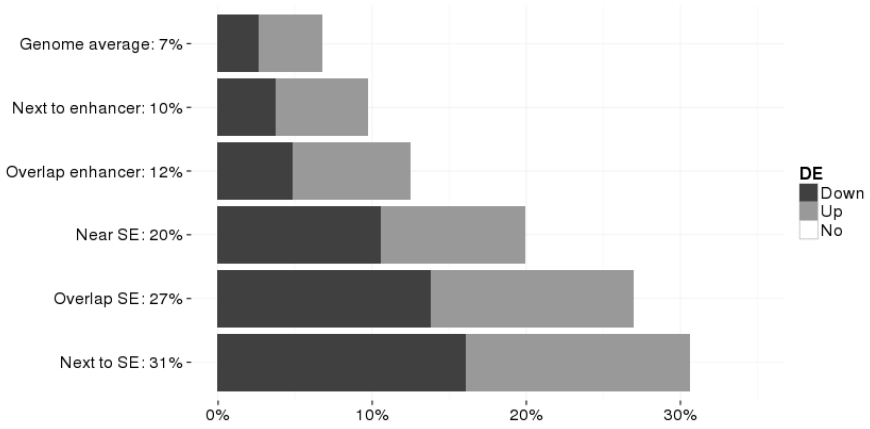

Figure 4: plot of chunk plot_gene_expr

Data for the graph above and chi-squared tests comparing each category to the genome-wide average.

```
kable(DE_mat)
```

Group	Down	Up	No	Percentage
expressed_genes	450	703	15850	7
nearest_enh	115	185	2780	10
overlap_enh	7	11	126	12
within40kb	110	97	829	20
overlap_SE	61	58	322	27
nearest_SE	72	65	310	31

```
DE_mat %>%
  dplyr::select(No,Up,Down) %>%
  as.matrix() %>%
  chisq.test()
```

```
##
##  Pearson's Chi-squared test
##
## data:  DE_mat %>% dplyr::select(No, Up, Down) %>% as.matrix()
## X-squared = 769.1191, df = 10, p-value < 2.2e-16
```

```
groups <- unique(DE_mat$Group[DE_mat$Group!="expressed_genes"])
for (g in groups){
  print(paste0("Comparing all expressed genes and ", g, ":"))

  DE_mat %>%
    filter(Group %in% c("expressed_genes", g)) %>%
    dplyr::select(No,Up,Down) %>%
    (function(x){x[1,] <- x[1,]-x[2,]
                 return(x)}) %>%  print()

  DE_mat %>%
    filter(Group %in% c("expressed_genes", g)) %>%
```

```
## [1] "Comparing all expressed genes and nearest_enh:"
## Source: local data frame [2 x 3]
##
##      No  Up Down
## 1 13070 518  335
## 2  2780 185  115
##
##  Pearson's Chi-squared test
##
## data:  DE_mat %>% filter(Group %in% c("expressed_genes", g)) %>% dplyr::select(No,     Up
## X-squared = 52.2091, df = 2, p-value = 4.602e-12
##
## [1] "Comparing all expressed genes and overlap_enh:"
## Source: local data frame [2 x 3]
##
##      No  Up Down
## 1 15724 692  443
## 2   126  11    7
##
##  Pearson's Chi-squared test
##
## data:  DE_mat %>% filter(Group %in% c("expressed_genes", g)) %>% dplyr::select(No,     Up
## X-squared = 7.5142, df = 2, p-value = 0.02335
##
## [1] "Comparing all expressed genes and within40kb:"
## Source: local data frame [2 x 3]
##
##      No  Up Down
## 1 15021 606  340
## 2   829  97  110
##
##  Pearson's Chi-squared test
##
## data:  DE_mat %>% filter(Group %in% c("expressed_genes", g)) %>% dplyr::select(No,     Up
## X-squared = 358.4209, df = 2, p-value < 2.2e-16
##
## [1] "Comparing all expressed genes and overlap_SE:"
## Source: local data frame [2 x 3]
##
##      No  Up Down
## 1 15528 645  389
## 2   322  58   61
##
##  Pearson's Chi-squared test
##
## data:  DE_mat %>% filter(Group %in% c("expressed_genes", g)) %>% dplyr::select(No,     Up
```

```
## X-squared = 322.8969, df = 2, p-value < 2.2e-16
##
## [1] "Comparing all expressed genes and nearest_SE:"
## Source: local data frame [2 x 3]
##
##      No  Up Down
## 1 15540 638  378
## 2   310  65   72
##
##  Pearson's Chi-squared test
##
## data:  DE_mat %>% filter(Group %in% c("expressed_genes", g)) %>% dplyr::select(No,     Up
## X-squared = 462.5945, df = 2, p-value < 2.2e-16
```

Figure 2C – H3K27ac correlates between WT and KO

```
load("~/HiC/data/H3K27ac_norm_coverage_list.RData")

## Super-enhancers
#create matrices of signal in 1000 bins
windows <- join_SEs <- join_SEs[rev(order(width(join_SEs)))]
targets <- H3K27ac_coverage_list

join_smList<- ScoreMatrixList(windows=windows, targets=targets, bin.num=1000)

WT1_totals <- apply(join_smList$WT1_H3K27ac,1,sum)
WT2_totals <- apply(join_smList$WT2_H3K27ac,1,sum)
KO1_totals <- apply(join_smList$KO1_H3K27ac, 1, sum)
KO2_totals <- apply(join_smList$KO2_H3K27ac, 1, sum)
totals.df <- data.frame(ID =1:length(WT1_totals), WT1_totals,WT2_totals, KO1_totals,KO2_tota
totals.df$WT_mean <- (totals.df$WT1_totals + totals.df$WT2_totals)/2
totals.df$KO_mean <- (totals.df$KO1_totals + totals.df$KO2_totals)/2
totals.df$ratio <- totals.df$KO_mean / totals.df$WT_mean

WTKOcor <- signif(cor(totals.df$WT_mean, totals.df$KO_mean, method="spearman"),4)
WTKOratio <- totals.df$ratio

## PLOT FIGURE 2C
ggplot(totals.df, aes(x=WT_mean, y=KO_mean)) +
    theme_bw(base_size=20) + theme(panel.grid.major=element_blank(),
                                   axis.ticks=element_blank(), axis.text=element_blank()) +
    geom_point() +
  labs(x="WT H3K27ac signal in super-enhancer regions (rpm)",
       y="KO H3K27ac signal in super-enhancer regions (rpm)") +
  xlim(0,5700) + ylim(0,5700) + geom_abline(intercept=0, slope=1,
```

```
                                                    linetype="dashed", colour="darkgrey") +
  annotate("text", x=1000, y=5500, label=paste("Spearman correlation=", WTKOcor))
```

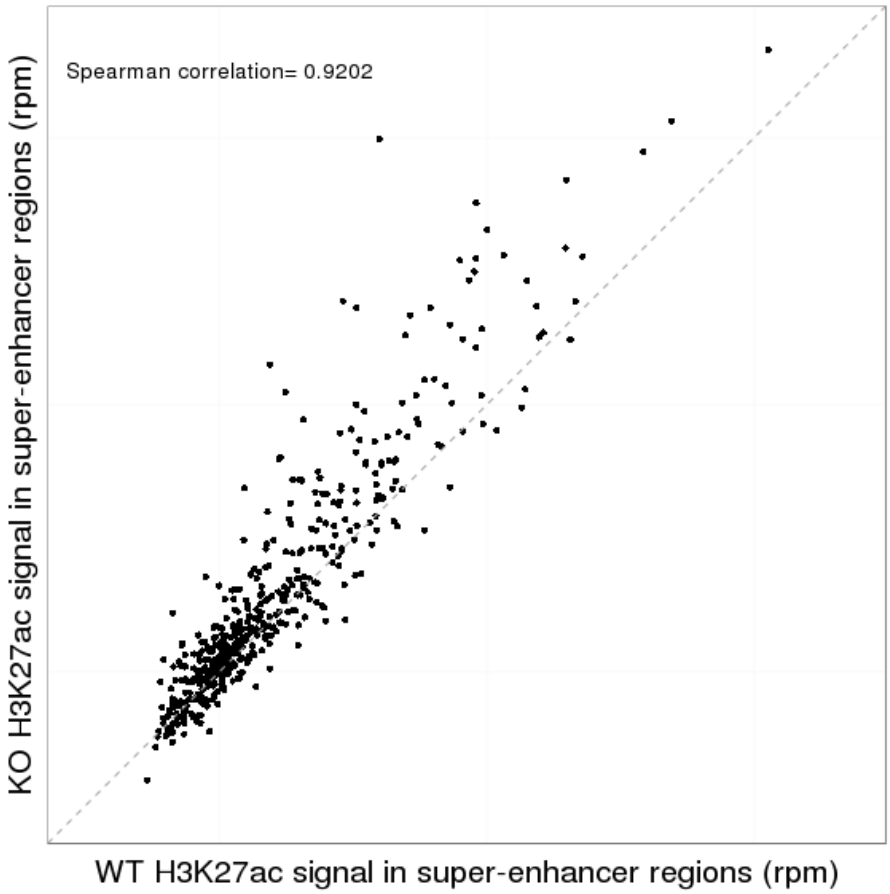

Figure 5: plot of chunk H3K27ac correlation

```
## Shen enhancers
Shen_enhancers <- read.table("~/HiC/data/thymus.enhancers.1kb.bed")
Shen_enhancers.gr <- GRanges(seqnames=Shen_enhancers$V1,
                             ranges=IRanges(Shen_enhancers$V2, Shen_enhancers$V3))

windows2 <- Shen_enhancers.gr

Shen_smlist <- ScoreMatrixList(windows=windows2, targets=targets, bin.num=1000)

WT1_totals <- apply(Shen_smlist$WT1_H3K27ac,1,sum)
```

```
WT2_totals <- apply(Shen_smlist$WT2_H3K27ac,1,sum)
KO1_totals <- apply(Shen_smlist$KO1_H3K27ac, 1, sum)
KO2_totals <- apply(Shen_smlist$KO2_H3K27ac, 1, sum)
totals.df <- data.frame(ID =1:length(WT1_totals), WT1_totals,WT2_totals, KO1_totals,KO2_tota
totals.df$WT_mean <- (totals.df$WT1_totals + totals.df$WT2_totals)/2
totals.df$KO_mean <- (totals.df$KO1_totals + totals.df$KO2_totals)/2
totals.df$ratio <- totals.df$KO_mean / totals.df$WT_mean

Shen_WTKOcor <- signif(cor(totals.df$WT_mean, totals.df$KO_mean, method="spearman"),4)

## PLOT FIGURE 2C
ggplot(totals.df, aes(x=WT_mean, y=KO_mean)) +
 theme_bw(base_size=20) + theme(panel.grid.major=element_blank(),
                                axis.ticks=element_blank(), axis.text=element_blank()) +
    geom_point() +
  labs(x="WT H3K27ac signal in Shen et al. enhancers (rpm)",
       y="KO H3K27ac signal in Shen et al. enhancers (rpm)") +
  xlim(0,5700) + ylim(0,5700) + geom_abline(intercept=0, slope=1,
                                            linetype="dashed", colour="darkgrey") +
  annotate("text", x=1000, y=5500, label=paste("Spearman correlation=", Shen_WTKOcor))


## Zhang enhancers
windows <- DP_stable.gr
targets <- H3K27ac_coverage_list

stable_smlist <- ScoreMatrixList(windows=windows, targets=targets, bin.num=100)

WT1_totals <- apply(stable_smlist$WT1_H3K27ac,1,sum)
WT2_totals <- apply(stable_smlist$WT2_H3K27ac,1,sum)
KO1_totals <- apply(stable_smlist$KO1_H3K27ac, 1, sum)
KO2_totals <- apply(stable_smlist$KO2_H3K27ac, 1, sum)
stable.df <- data.frame(ID =1:length(WT1_totals), WT1_totals,WT2_totals, KO1_totals,KO2_tota
stable.df$WT_mean <- (stable.df$WT1_totals + stable.df$WT2_totals)/2
stable.df$KO_mean <- (stable.df$KO1_totals + stable.df$KO2_totals)/2
stable.df$ratio <- stable.df$KO_mean / stable.df$WT_mean
stable.df$group <- "Stable"

windows <- DP_only.gr
targets <- H3K27ac_coverage_list

new_smlist <- ScoreMatrixList(windows=windows, targets=targets, bin.num=100)

WT1_totals <- apply(new_smlist$WT1_H3K27ac,1,sum)
WT2_totals <- apply(new_smlist$WT2_H3K27ac,1,sum)
KO1_totals <- apply(new_smlist$KO1_H3K27ac, 1, sum)
```

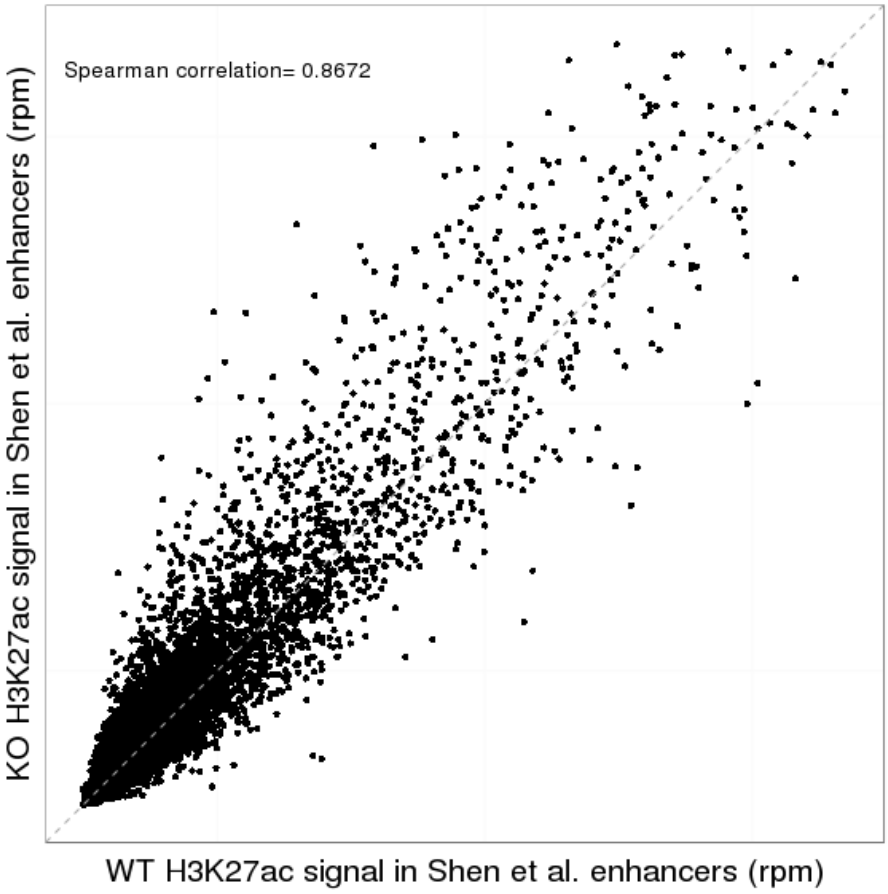

Figure 6: plot of chunk H3K27ac correlation

```
KO2_totals <- apply(new_smlist$KO2_H3K27ac, 1, sum)
new.df <- data.frame(ID =1:length(WT1_totals), WT1_totals,WT2_totals, KO1_totals,KO2_totals)
new.df$WT_mean <- (new.df$WT1_totals + new.df$WT2_totals)/2
new.df$KO_mean <- (new.df$KO1_totals + new.df$KO2_totals)/2
new.df$ratio <- new.df$KO_mean / new.df$WT_mean
new.df$group <- "New"

totals.df <- rbind(stable.df, new.df)

WTKOcor <- signif(cor(totals.df$WT_mean, totals.df$KO_mean, method="spearman"),4)

## PLOT FIGURE 2C
ggplot(totals.df, aes(x=WT_mean, y=KO_mean, colour=group)) +
    theme_bw(base_size=20) +
    theme(panel.grid.major=element_blank(), axis.ticks=element_blank(),
          axis.text=element_blank(), legend.position="none") +
  geom_point(size=2) +
  scale_colour_manual(values=c("firebrick2","darkgrey")) +
  labs(x="WT H3K27ac signal in Zhang et al. enhancers (rpm)",
       y="KO H3K27ac signal in Zhang et al. enhancers (rpm)") +
  geom_abline(intercept=0, slope=1, linetype="dashed", colour="darkgrey") +
  annotate("text", x=250, y=1500, label=paste("Spearman correlation=", WTKOcor))
```

Figure 2D – H3K27ac changes don't explain deregulation

```
#get gene-SE pairs
#genes within 40kb
f0 <- findOverlaps(promoters, join_SEs, maxgap=40000)
within40kb_promoters <- promoters[queryHits(f0)]
within40kb_genes <- gene_expr.df[match(within40kb_promoters$ensG, gene_expr.df$ensG),]
within40kb_genes$ID <- subjectHits(f0)
rownames(within40kb_genes) <- NULL
within40kb_genes <- unique(within40kb_genes)

# note that some genes are repeated
#1097 rows, 1036 unique genes

#get SEs with H3K27ac data
windows <- join_SEs <- join_SEs[rev(order(width(join_SEs)))]
targets <- H3K27ac_coverage_list

join_smList<- ScoreMatrixList(windows=windows, targets=targets, bin.num=1000)

WT1_totals <- apply(join_smList$WT1_H3K27ac,1,sum)
```

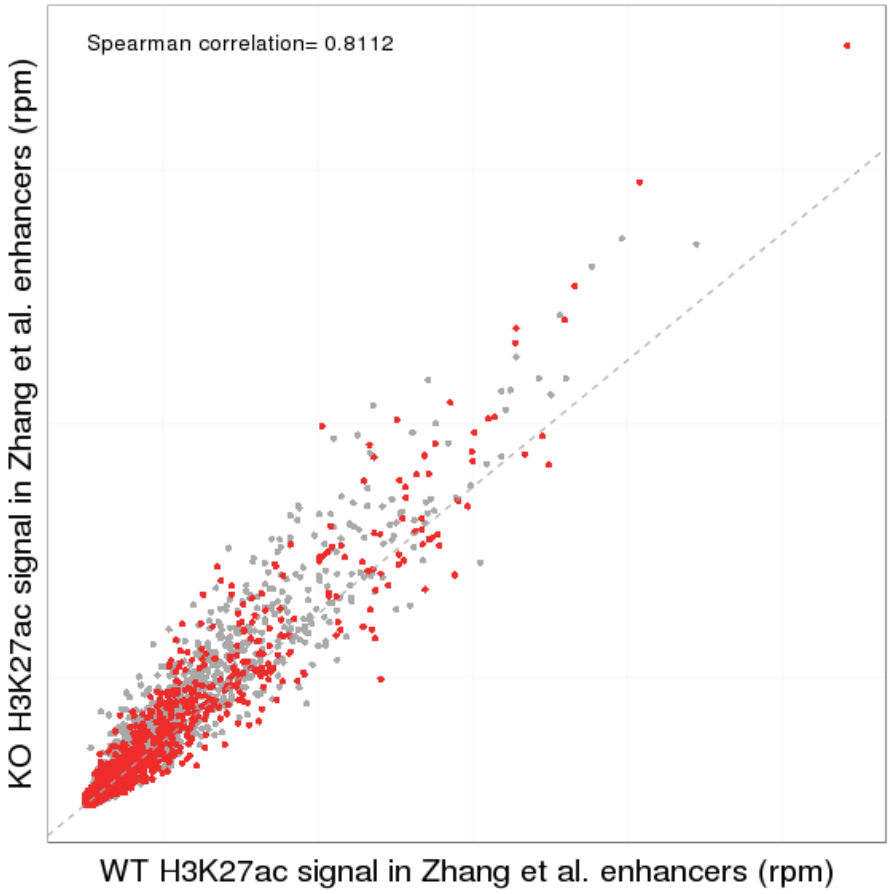

Figure 7: plot of chunk H3K27ac correlation

```
WT2_totals <- apply(join_smList$WT2_H3K27ac,1,sum)
KO1_totals <- apply(join_smList$KO1_H3K27ac, 1, sum)
KO2_totals <- apply(join_smList$KO2_H3K27ac, 1, sum)
SE_totals.df <- data.frame(ID =1:length(WT1_totals), WT1_totals,WT2_totals, KO1_totals,KO2_t
SE_totals.df$WT_mean <- (SE_totals.df$WT1_totals + SE_totals.df$WT2_totals)/2
SE_totals.df$KO_mean <- (SE_totals.df$KO1_totals + SE_totals.df$KO2_totals)/2
SE_totals.df$log2ratio <- log2(SE_totals.df$KO_mean / SE_totals.df$WT_mean)

#combine
genes_SEs_H3K27ac <- merge(within40kb_genes, SE_totals.df,
                           by="ID")[, c("ID", "log2ratio", "WT_mean", "KO_mean",
                                        "ensG", "DE_down", "DE_up", "log2FoldChange_pseudo",
                                        "prom_CTCF", "prom_Rad21", "prom_Rad21_CTCF")]

genes_SEs_H3K27ac$promoter_mark <- "None"
genes_SEs_H3K27ac$promoter_mark[genes_SEs_H3K27ac$prom_CTCF==TRUE] <- "CTCF"
genes_SEs_H3K27ac$promoter_mark[genes_SEs_H3K27ac$prom_Rad21==TRUE] <- "Rad21"
genes_SEs_H3K27ac$promoter_mark[genes_SEs_H3K27ac$prom_Rad21_CTCF==TRUE] <- "Both"

genes_SEs_H3K27ac$DE <- "No"
genes_SEs_H3K27ac$DE[genes_SEs_H3K27ac$DE_up == TRUE] <- "Up"
genes_SEs_H3K27ac$DE[genes_SEs_H3K27ac$DE_down == TRUE] <- "Down"
genes_SEs_H3K27ac$DE <- factor(genes_SEs_H3K27ac$DE, levels = c("No", "Down", "Up"))

cols <- c("No" = "grey50", "Up" = muted("red"), "Down" = muted("blue"))

#get cutoffs:
genes_SEs_H3K27ac$H3K27ac_group <- factor(cut(genes_SEs_H3K27ac$log2ratio,
                                              quantile(genes_SEs_H3K27ac$log2ratio,
                                                       probs = seq(0,1, length.out=4)), incl
                                          labels = c("Lower 1/3", "Middle 1/3", "Top 1/3"))
genes_SEs_H3K27ac$DE <- factor(genes_SEs_H3K27ac$DE, c("Down","Up", "No"))

cols <- c("No" = NA, "Up" = col_up, "Down" = col_up)
#plot
ggplot(genes_SEs_H3K27ac, aes(x=H3K27ac_group, fill=DE)) + geom_bar(position="fill") +
  scale_y_continuous("",labels=percent, limits=c(0,0.25)) +
  scale_x_discrete("") +
  scale_fill_manual(values=cols) +
  theme_bw(base_size=20) + theme(panel.grid.major=element_blank())  + coord_flip()


genes_SEs_H3K27ac %>%
  dplyr::select(ensG, DE_up, DE_down, DE, H3K27ac_group) %>%
  group_by(H3K27ac_group, DE) %>%
  summarise(count=n()) %>%
```

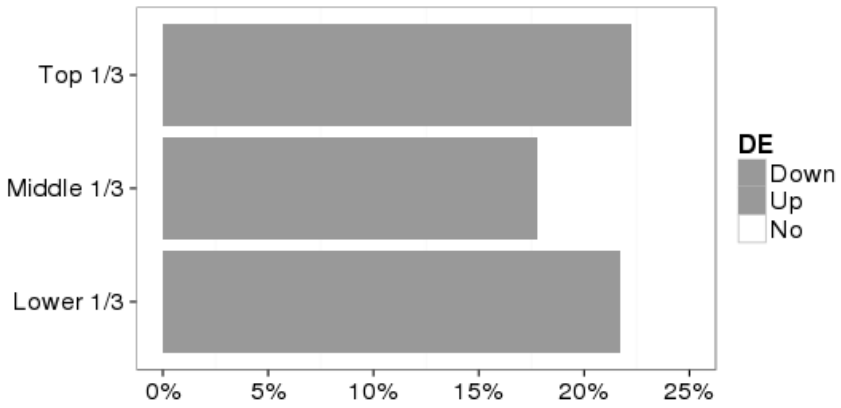

Figure 8: plot of chunk genes_SEs_H3K27ac_plotting

```
ungroup() %>% ##dplyr bug :(
spread(DE, count) %>% kable()
```

H3K27ac_group	Down	Up	No
Lower 1/3	72	8	288
Middle 1/3	35	30	300
Top 1/3	17	64	283

```
#get enhancer centres and nearest promoters
centres <- resize(join_nonSE, fix="center", width=1)

nearest_promoters <- promoters[nearest(centres,promoters)]
nearest_genes_to_nonSEs <- gene_expr.df[match(nearest_promoters$ensG, gene_expr.df$ensG),]

nearest_genes_to_nonSEs$ID <- centres$ID
rownames(nearest_genes_to_nonSEs) <- NULL
nearest_genes_to_nonSEs <- unique(nearest_genes_to_nonSEs)

#extra step here - remove any that are also associated with SEs
nearest_genes_to_nonSEs <- nearest_genes_to_nonSEs[!(nearest_genes_to_nonSEs$ensG %in% neare

# note that some genes are repeated
#5048 rows, 3540 unique genes

#get SEs with H3K27ac data
windows <- join_nonSE <- join_nonSE[rev(order(width(join_nonSE)))]
targets <- H3K27ac_coverage_list

join_smList<- ScoreMatrixList(windows=windows, targets=targets, bin.num=1000)

WT1_totals <- apply(join_smList$WT1_H3K27ac,1,sum)
WT2_totals <- apply(join_smList$WT2_H3K27ac,1,sum)
KO1_totals <- apply(join_smList$KO1_H3K27ac, 1, sum)
KO2_totals <- apply(join_smList$KO2_H3K27ac, 1, sum)
totals.df <- data.frame(ID =1:length(WT1_totals), WT1_totals,WT2_totals, KO1_totals,KO2_tota
totals.df$WT_mean <- (totals.df$WT1_totals + totals.df$WT2_totals)/2
```

```r
genes_enh_H3K27ac$DE[genes_enh_H3K27ac$DE_up == TRUE] <- "Up"
genes_enh_H3K27ac$DE[genes_enh_H3K27ac$DE_down == TRUE] <- "Down"
genes_enh_H3K27ac$DE <- factor(genes_enh_H3K27ac$DE, levels = c("No", "Down", "Up"))

cols <- c("No" = "grey50", "Up" = muted("red"), "Down" = muted("blue"))

#get cutoffs:
genes_enh_H3K27ac$H3K27ac_group <- factor(cut(genes_enh_H3K27ac$log2ratio,
                                              quantile(genes_enh_H3K27ac$log2ratio,
                                                       probs = seq(0,1, length.out=4)), incl
                                          labels = c("Lower 1/3", "Middle 1/3", "Top 1/3"))
genes_enh_H3K27ac$DE <- factor(genes_enh_H3K27ac$DE, c("Down","Up", "No"))
table(genes_enh_H3K27ac$H3K27ac_group)

##
##  Lower 1/3 Middle 1/3    Top 1/3
##       1322       1322       1322

#plot
cols <- c("No" = NA, "Up" = col_up, "Down" = col_up)
#plot
ggplot(genes_enh_H3K27ac, aes(x=H3K27ac_group, fill=DE)) + geom_bar(position="fill") +
  scale_y_continuous("",labels=percent, limits=c(0,0.15)) +
  scale_x_discrete("") +
  scale_fill_manual(values=cols) +
  theme_bw(base_size=20) + theme(panel.grid.major=element_blank())  + coord_flip()
```

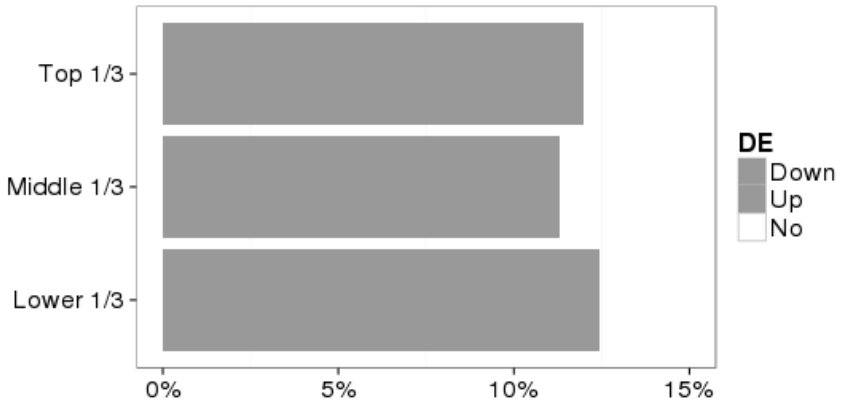

Figure 9: plot of chunk plot_genes_enh_H3K27ac

```
genes_enh_H3K27ac %>%
  dplyr::select(ensG, DE_up, DE_down, DE, H3K27ac_group) %>%
  group_by(H3K27ac_group, DE) %>%
  summarise(count=n()) %>%
  ungroup() %>% ##dplyr bug :(
  spread(DE, count) %>% kable()
```

H3K27ac_group	Down	Up	No
Lower 1/3	95	70	1157
Middle 1/3	56	94	1172
Top 1/3	41	118	1163

Figure 3A and B – cohesin and CTCF signal at enhancers and super-enhancers

```
load("/mnt/biggles/csc_projects/lizis/ES-DPT/DPT/DPT_normalised_coverage_list.RData")
load("~/HiC/data/DKO_H3K27ac_normalised_coverage.RData")

normalised_coverage_list_all <- list(Rad21=normalised_coverage_list$Rad21,
                                     CTCF=normalised_coverage_list$CTCF_Shih,
                                     Smc1a=normalised_coverage_list$Smc1a,
                                     WT1_H3K27ac = H3K27ac_coverage_list$WT1_H3K27ac,
                                     WT2_H3K27ac = H3K27ac_coverage_list$WT2_H3K27ac,
                                     KO1_H3K27ac = H3K27ac_coverage_list$KO1_H3K27ac,
                                     KO2_H3K27ac = H3K27ac_coverage_list$KO2_H3K27ac,
                                     Nipbl = normalised_coverage_list$Nipbl,
                                     Med1 = normalised_coverage_list$Med1,
                                     H3K4me3 = normalised_coverage_list$H3K4me3,
                                     H3K4me1 = normalised_coverage_list$H3K4me1)

size=100
SE.plot.matrix.list <- lapply(normalised_coverage_list_all,
                              function(x){generate.plot.matrix(x, join_SEs, size)})
names(SE.plot.matrix.list) <- names(normalised_coverage_list_all)

nonSE.plot.matrix.list <- lapply(normalised_coverage_list_all,
                                 function(x){generate.plot.matrix(x, join_nonSE, size)})
names(nonSE.plot.matrix.list) <- names(normalised_coverage_list_all)


## PLOT FIGURE 3A
par(cex=2, mar=c(2.1,2.1,1,1))
plot(colMeans(nonSE.plot.matrix.list$CTCF), type="l", ylab= "ChIP-seq signal",
     ylim =c(0,1), xaxt="n", xlab="", col="blue", lwd=2, ann=FALSE)
```

```
axis(side=1, at=c(size,size*2), labels=c("Start", "End"))

lines(colMeans(nonSE.plot.matrix.list$Rad21), col="red", lwd=2)
lines(colMeans(nonSE.plot.matrix.list$Smc1a), col="darkred", lwd=2)

legend("topright", legend = c("CTCF", "Rad21", "Smc1a"),
       col = c("blue", "red", "darkred"), lty=1, lwd=2, bty="n")
legend("topleft", legend="Conventional enhancers", bty="n")
```

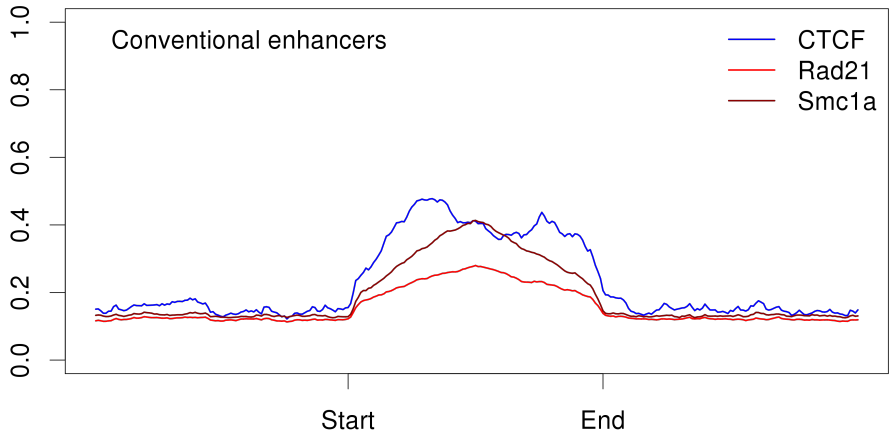

Figure 10: plot of chunk plot_metagenes

```
## PLOT FIGURE 3B
par(cex=2, mar=c(2.1,2.1,1,1))
plot(colMeans(SE.plot.matrix.list$CTCF), type="l", ylab= "ChIP-seq signal",
     ylim =c(0,1), xaxt="n", xlab="", col="blue", lwd=2, ann=FALSE)
axis(side=1, at=c(size,size*2), labels=c("Start", "End"))

lines(colMeans(SE.plot.matrix.list$Rad21), col="red", lwd=2)
lines(colMeans(SE.plot.matrix.list$Smc1a), col="darkred", lwd=2)

legend("topright", legend = c("CTCF", "Rad21", "Smc1a"),
       col = c("blue", "red", "darkred"), lty=1, lwd=2, bty="n")
legend("topleft", legend="Super-enhancers", bty="n")
```

Figure 3C – heatmaps of signal at super-enhancers grouped by CTCF binding

```
load("/mnt/biggles/csc_projects/lizis/ES-DPT/DPT/DPT_GRanges_list.RData")
CTCF_peaks <- DPT_peaks$CTCF
```

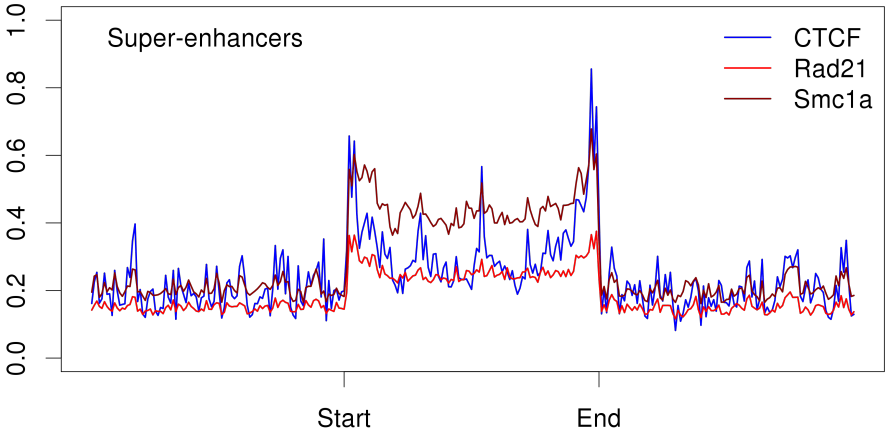

Figure 11: plot of chunk plot_metagenes

```
starts <- resize(join_SEs, fix="start", width=1)
ends <- resize(join_SEs, fix="end", width=1)

start_CTCF_idx <- subjectHits(findOverlaps(CTCF_peaks, starts, maxgap=2000))
end_CTCF_idx <- subjectHits(findOverlaps(CTCF_peaks, ends, maxgap=2000))

both_CTCF <- join_SEs[join_SEs$ID %in% start_CTCF_idx & join_SEs$ID %in% end_CTCF_idx]
start_CTCF <- join_SEs[join_SEs$ID %in% start_CTCF_idx & !(join_SEs$ID %in% end_CTCF_idx)]
end_CTCF <- join_SEs[join_SEs$ID %in% end_CTCF_idx & !(join_SEs$ID %in% start_CTCF_idx)]
non_CTCF <- join_SEs[!(join_SEs$ID %in% end_CTCF_idx) & !(join_SEs$ID %in% start_CTCF_idx)]

both_CTCF_idx <- join_SEs[join_SEs$ID %in% start_CTCF_idx & join_SEs$ID %in% end_CTCF_idx]$I
start2_CTCF_idx <- join_SEs[join_SEs$ID %in% start_CTCF_idx & !(join_SEs$ID %in% end_CTCF_id
end2_CTCF_idx <- join_SEs[join_SEs$ID %in% end_CTCF_idx & !(join_SEs$ID %in% start_CTCF_idx)
non_CTCF_idx <- join_SEs[!(join_SEs$ID %in% end_CTCF_idx) & !(join_SEs$ID %in% start_CTCF_id

windows <- resize(join_SEs, width=100000, fix="center")
targets <- c("WT H3K27ac" = "/mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/ChIPSeq/
  "KO H3K27ac" = "/mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/ChIPSeq/H3K27ac/Ali
  H3K4me1="/mnt/biggles/csc_projects/lizis/ES-DPT/DPT/Aligned/DPT_H3K4me1_sorted.bam",
  H3K4me3="/mnt/biggles/csc_projects/lizis/ES-DPT/DPT/Aligned/DPT_H3K4me3_sorted.bam",
    Med1 = "/mnt/biggles/csc_projects/lizis/ES-DPT/DPT/Aligned/DP_thymocyte_Med1_all_Vlad_so
    Nipbl="/mnt/biggles/csc_projects/lizis/ES-DPT/DPT/Aligned/DP_thymocyte_Nipbl_all_Vlad_m1
    Rad21= "/mnt/biggles/csc_projects/lizis/ES-DPT/DPT/Aligned/DP_thymocyte_Rad21_all_m1_v2_
    Smc1a="/mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/ChIPSeq//Smc1a//Smc1aRep1.
  CTCF="/mnt/biggles/csc_projects/lizis/ES-DPT/DPT/Aligned/DP_thymocyte_CTCF_Shih_sorted.bam
```

```
joinSEs_smList<- ScoreMatrixList(windows=windows, targets=targets,
  bin.num=1000, type="bam")
names(joinSEs_smList) <- names(targets)

joinSEs_smList_outliers_removed <- lapply(joinSEs_smList, function(x){
  up_level <- quantile(x, 0.99)
  x[(x > up_level)] <- up_level
  down_level <- quantile(x,0.01)
  x[(x < down_level)] <- down_level
  return(x)
})
class(joinSEs_smList_outliers_removed) <- "ScoreMatrixList"

joinSEs_smList_scaled <- scaleScoreMatrixList(joinSEs_smList_outliers_removed)

multiHeatMatrix(joinSEs_smList_scaled, xcoords = NULL,
                group = list("Both"=both_CTCF_idx, "One"=c(start2_CTCF_idx,end2_CTCF_idx), '
                common.scale=TRUE, xlab="", cex.main=1.5, col=c("white", "darkblue"))
```

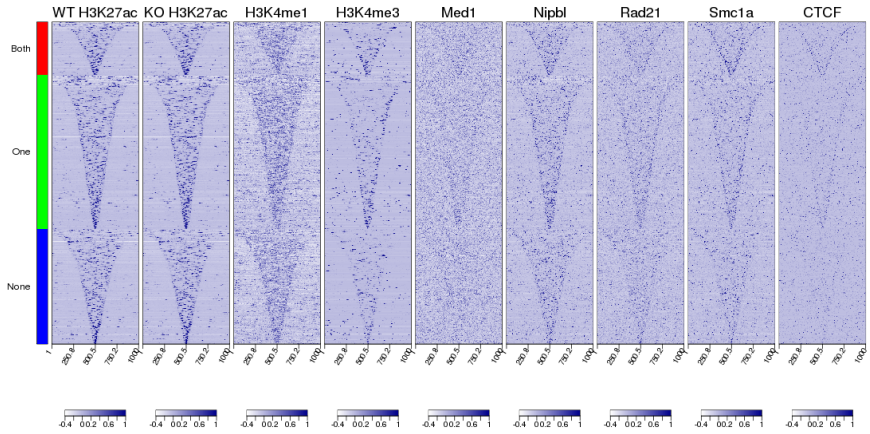

Figure 12: plot of chunk CTCF_flanking

SIMA results

The HOMER Hi-C software analysis pipeline (http://biowhat.ucsd.edu/homer/interactions/) was used to determine significant interactions, differential interactions and to perform Structured Interaction Matrix Analysis (SIMA) (Lin et al., 2012).

To determine genomic features associated with chromatin interactions, we used

SIMA, which pools Hi-C information associated with a given set of genomic regions within a specified set of domains (Lin et al., 2012). We used default resolution ("-res 2500") and optimal Hi-C interaction search space parameters ("-superRes 10000") to consider all reads within a 10kb window around the centre of each feature. Within-compartment associations were assessed independently in control and cohesin-deficient thymocytes for RAD21 peaks, canonical TSSs (excluding pseudogenes; Ensembl version 66), conventional enhancers (Shen et al. 2012) and random regions, as described previously (Seitan et al. 2013). Within-super-enhancer interactions were assessed for all super-enhancers of more than 100kb or 50kb in length. 'Peaks' within these regions were defined by taking the summits of constituent H3K27ac peaks, extending to 1kb and taking the intersection of these regions between all samples. Where super-enhancers that are not active in thymocytes were considered, as these contain no or very few H3K27ac peaks, we chose random "peaks" within them such that the number of peaks in each region was similar to the number of peaks in thymocyte super-enhancers of comparable size. All interactions were normalized using HOMER with a background model that takes sequencing depth and genomic distance between interacting regions into account.

Figure 4D

```
sima_table <- read.table("~/HiC/data/sima_all_homotypic_Enhancer_ranked.csv",
                         header=TRUE, sep=",") #obtained from Andre Faure

sima_gr <- GRanges(seqnames=sima_table$chr.1.,
                   ranges=IRanges(sima_table$start.1., sima_table$end.1.),
                        mcols=data.frame(sima_table$X.DomainName.1., sima_table$PeakEnrichme
                               sima_table$KO_PeakEnrichment, sima_table$KOWT_Ratio))

fO <- findOverlaps(join_SEs, sima_gr)
sima_without_SE <- sima_gr[-subjectHits(fO)]
sima_gr_with_SE <- subsetByOverlaps(sima_gr, join_SEs)

w <- wilcox.test(sima_gr_with_SE$mcols.sima_table.KOWT_Ratio, sima_without_SE$mcols.sima_tab
#can't use wilcoxsign_test from coin as diff numbers of rows

sima.bp <- data.frame(rbind(cbind(sima_gr_with_SE$mcols.sima_table.KOWT_Ratio, "With"),
    cbind(sima_without_SE$mcols.sima_table.KOWT_Ratio, "Without")))

names(sima.bp) <- c("KOWT_Ratio", "Group")

sima.bp$KOWT_Ratio <- as.numeric(as.character(sima.bp$KOWT_Ratio))

#barplot effect sizes
sima_all <- wilcoxsign_test(sima_gr$mcols.sima_table.KO_PeakEnrichment ~ sima_gr$mcols.sima_
```

```
sima_without<- wilcoxsign_test(sima_without_SE$mcols.sima_table.KO_PeakEnrichment ~ sima_wit
sima_with<- wilcoxsign_test(sima_gr_with_SE$mcols.sima_table.KO_PeakEnrichment ~ sima_gr_wit

r_all <- as.numeric(statistic(sima_all, "test"))/sqrt(2*length(sima_gr))
r_without <- as.numeric(statistic(sima_without, "test"))/sqrt(2*length(sima_without_SE))
r_with <- as.numeric(statistic(sima_with, "test"))/sqrt(2*length(sima_gr_with_SE))

effect_size_df <- data.frame(Group=factor(c("with", "without", "all"),
                                          levels=c("with", "without", "all")),
                             Effect=c(r_with, r_without, r_all),
                             P=paste("p =",signif(c(as.numeric(pvalue(sima_with)),
                                                    as.numeric(pvalue(sima_without)),
                                                    as.numeric(pvalue(sima_all))),3)),
                             labs=c("Open compartments with SEs",
                                    "Open compartments without SEs",
                                    "All open compartments"))
labs <- effect_size_df$labs
ggplot(effect_size_df, aes(x=Group, y=Effect, label=P)) + geom_bar(stat="identity") +
    coord_flip() + scale_y_reverse("Effect size") +
    scale_x_discrete(name="",labels=labs) +
    geom_text(y=0, size=10, colour="white", fontface="bold", hjust=-0.05) +
  theme_bw(base_size=20) + theme(panel.border=element_blank(), panel.grid.major=element_blan
```

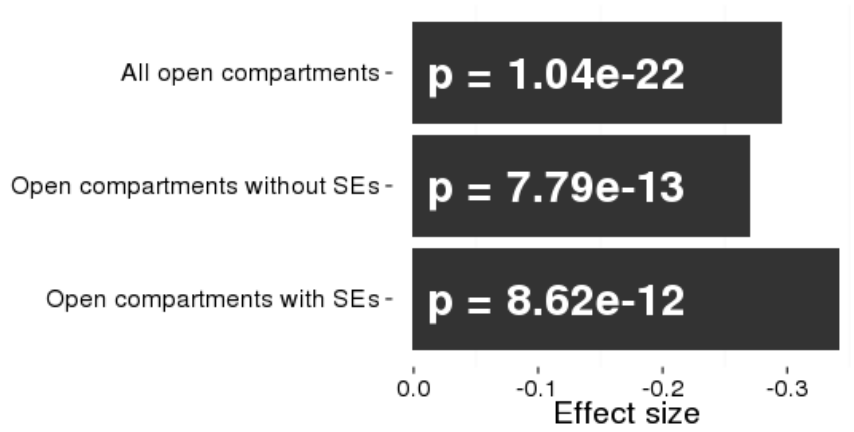

Figure 13: plot of chunk SIMA

```
#note that initial '#' removed from these files to allow header to be read by R...
WT <- read.table("/mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/HiC/SIMA/SIMA_outpu
                 sep="\t", header=T)
KO <- read.table("/mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/HiC/SIMA/SIMA_outpu
```

```
                sep="\t", header=T)

merged <- merge(WT, KO, by=names(WT)[1:12])
names(merged) <- sub("\\.x", "_WT", names(merged))
names(merged) <- sub("\\.y", "_KO", names(merged))

merged$region_size <- merged$end.1. - merged$start.1.
merged$KOWT_Ratio <- merged$Ratio_KO / merged$Ratio_WT

merged <- merged[order(merged$KOWT_Ratio),]
write.table(merged, file="SIMA_KOWT_results.txt", quote=F,
            sep="\t", row.names=F, col.names=T)

w <- wilcoxsign_test(merged$Ratio_KO ~ merged$Ratio_WT)
r <- as.numeric(statistic(w, "test"))/ sqrt(2*length(merged$Ratio_KO))

w1 <- wilcoxsign_test(merged$PeakEnrichment_KO ~ merged$PeakEnrichment_WT)
r1 <- as.numeric(statistic(w1, "test"))/ sqrt(2*length(merged$Ratio_KO))

w2 <- wilcoxsign_test(merged$RandEnrichment_KO ~ merged$RandEnrichment_WT)
r2 <- as.numeric(statistic(w2, "test"))/ sqrt(2*length(merged$Ratio_KO))

#KOWT
df <- data.frame(Ratio=c(merged$Ratio_WT, merged$Ratio_KO),
                 PVal = c(merged$p.value_WT, merged$p.value_KO),
                 Group=c(rep("WT", length(merged$Ratio_WT)),
                         rep("KO", length(merged$Ratio_KO))))
```

To compare interactions within enhancer clusters to interactions within random regions, I ran SIMA within a set of shuffled regions of the same size as the super-enhancer regions and with the same number of constituent peaks randomly placed within them. Their positions were constrained to be in active chromatin compartments.

```
#code used to shuffle regions and run SIMA
print(system("cat ~/HiC/SIMA_randomisation/run_random_SIMA.sh", intern=TRUE))
```

```
##  [1] "#! /bin/bash"
##  [2] ""
##  [3] "#bedtools intersect -a ~/HiC/ROSE_output_analysis/join_SEs_m100kb.bed -b ~/HiC/data
##  [4] ""
##  [5] "#SBATCH -c 3"
##  [6] ""
##  [7] "i=$SLURM_ARRAY_TASK_ID"
##  [8] ""
```

```
##  [9] "echo \"shuffling\" $i"
## [10] "bedtools shuffle -i ~/HiC/ROSE_output_analysis/join_SEs_m100kb.bed -g ~/mm9_data/mm
## [11] "bedtools shuffle -i summits_in_SEs.bed -g ~/mm9_data/mm9.chrom.sizes -incl temp${i}
## [12] ""
## [13] "bed2pos.pl -unique temp${i}_SEs.bed > temp${i}_SEs.homer"
## [14] "bed2pos.pl temp${i}_summits_in_SEs.bed > temp${i}_summits_in_SEs.homer"
## [15] ""
## [16] "SIMA.pl /mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/HiC/WT_HiC_HindIII/
## [17] "SIMA.pl /mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/HiC/KO_HiC_HindIII/
## [18] ""
```

```
columns <- c("DomainName1","chr1","start1","end1","DomainName2","chr2","start2",
             "end2","PeakFile1","PeakFile2","Npx","Npy","pvalue","Ratio",
             "PeakEnrichment","RandEnrichment")

results_df <- data.frame(matrix(0, nrow=21*500, ncol=20))
names(results_df) <- c(columns[1:12], paste0(columns[13:16], "_WT"),
                       paste0(columns[13:16], "_KO"))

for (i in 1:500){
  files <- list.files(path="~/HiC/SIMA_randomisation/",
                      pattern=paste0("SIMA_output_KO", i, ".txt", "|", "SIMA_output_WT", i,
                      full.names=TRUE)
  KO <- read.table(files[1], header=F, stringsAsFactors = FALSE, col.names=columns)
  WT <- read.table(files[2], header=F, stringsAsFactors = FALSE, col.names=columns)
  merged <- merge(WT,KO, by=columns[1:12])
  names(merged) <- sub("\\.x", "_WT", names(merged))
  names(merged) <- sub("\\.y", "_KO", names(merged))

  if (nrow(merged)!=21){
    warning(paste("wrong number of rows in sample", i))

    merged <- rbind(merged, rep(NA, 20))
  }

  results_df[((21*i)-20) : (21*i) ,] <- merged
}

results_df <- results_df[complete.cases(results_df),]


##Get significance results:

w <- wilcoxsign_test(merged$Ratio_KO ~ merged$Ratio_WT)
r <- as.numeric(statistic(w, "test"))/ sqrt(2*length(merged$Ratio_KO))
```

```
w1 <- wilcoxsign_test(merged$PeakEnrichment_KO ~ merged$PeakEnrichment_WT)
r1 <- as.numeric(statistic(w1, "test"))/ sqrt(2*length(merged$Ratio_KO))

w2 <- wilcoxsign_test(merged$RandEnrichment_KO ~ merged$RandEnrichment_WT)
r2 <- as.numeric(statistic(w2, "test"))/ sqrt(2*length(merged$Ratio_KO))


#KOWT
df <- data.frame(Ratio=c(merged$Ratio_WT, merged$Ratio_KO),
                 PVal = c(merged$pvalue_WT, merged$pvalue_KO),
                 Group=c(rep("WT", length(merged$Ratio_WT)),
                         rep("KO", length(merged$Ratio_KO))))
```

To compare to interactions between enhancers and interactions in random regions, I have also carried out SIMA using other features: transcription start sites, Rad21 peaks, CTCF peaks and 'random' peaks (all obtained from Andre). The following graphs represent this analysis within enhancer clusters / random regions of similar size.

```
files <- list.files(path="~/HiC/SIMA_extra/", pattern="SIMA_output.txt", full.names=TRUE)
files <- files[-grep("comp", files)]
names <- unique(sapply(files, function(f){
               x <- strsplit(f, "/")[[1]]
               strsplit(x[length(x)], ".", fixed=TRUE)[[1]][1]
  }))

tables <- lapply(names, function(n){
  fs <- files[grep(n, files)]

  #read real regions
  KO <- read.table(fs[2], header=F, stringsAsFactors = FALSE, col.names=columns)
  WT <- read.table(fs[4], header=F, stringsAsFactors = FALSE, col.names=columns)
  merged <- merge(WT,KO, by=columns[1:12])
  names(merged) <- sub("\\.x", "_WT", names(merged))
  names(merged) <- sub("\\.y", "_KO", names(merged))
  real_regions <- merged

  #read random regions
  KO <- read.table(fs[1], header=F, stringsAsFactors = FALSE, col.names=columns)
  WT <- read.table(fs[3], header=F, stringsAsFactors = FALSE, col.names=columns)
  merged <- merge(WT,KO, by=columns[1:12])
  names(merged) <- sub("\\.x", "_WT", names(merged))
  names(merged) <- sub("\\.y", "_KO", names(merged))

  random_regions <- merged
```

```
  return(list(real_regions, random_regions, n))

})

names(tables) <- names

files <- list.files(path="~/HiC/SIMA_extra/", pattern="SIMA_output.txt", full.names=TRUE)
files <- files[grep("comp", files)]
names <- unique(sapply(files, function(f){
               x <- strsplit(f, "/")[[1]]
               strsplit(x[length(x)], ".", fixed=TRUE)[[1]][1]
  }))

tables <- lapply(names, function(n){
  fs <- files[grep(n, files)]

  #read real regions
  KO <- read.table(fs[1], header=F, stringsAsFactors = FALSE, col.names=columns)
  WT <- read.table(fs[2], header=F, stringsAsFactors = FALSE, col.names=columns)
  merged <- merge(WT,KO, by=columns[1:12])
  names(merged) <- sub("\\.x", "_WT", names(merged))
  names(merged) <- sub("\\.y", "_KO", names(merged))
  real_regions <- merged

  return(list(real_regions, n))

})

names(tables) <- names

counts <- nrow(tables[[1]][[1]])
```

Figure 4C

```
#Collate data to show that the magnitude of enh-enh interaction reduction in enhancer cluste
#to reduction in interaction between cohesin binding sites
EE_in_SE <- read.table("SIMA_KOWT_results.txt", sep="\t", header=TRUE)

EE_in_comp <- read.table("~/HiC/data/sima_all_homotypic_Enhancer_ranked.csv", header=TRUE, s

df <- data.frame(Interaction = c(tables$Rad21_peaks[[1]]$PeakEnrichment_WT,
                                 tables$Rad21_peaks[[1]]$PeakEnrichment_KO,
                                 tables$TSS[[1]]$PeakEnrichment_WT,
                                 tables$TSS[[1]]$PeakEnrichment_KO,
                                tables$Random_peaks[[1]]$PeakEnrichment_WT,
```

```
                                    tables$Random_peaks[[1]]$PeakEnrichment_KO,
                                    EE_in_comp$PeakEnrichment,
                                    EE_in_comp$KO_PeakEnrichment,
                                    EE_in_SE$PeakEnrichment_WT,
                                    EE_in_SE$PeakEnrichment_KO),
                  Group=c(rep("Rad21_WT", counts), rep("Rad21_KO", counts),
                          rep("TSS_WT", counts), rep("TSS_KO", counts),
                          rep("Random_WT", counts), rep("Random_KO", counts),
                          rep("Enhancers_WT", nrow(EE_in_comp)),
                          rep("Enhancers_KO", nrow(EE_in_comp)),
                          rep("SE_WT", 21), rep("SE_KO", 21)))

df$Group <- factor(df$Group, levels=c("SE_WT", "SE_KO", "Enhancers_WT", "Enhancers_KO",
                                      "Rad21_WT", "Rad21_KO", "TSS_WT", "TSS_KO",
                                      "Random_WT", "Random_KO"))

p_EE_SE <- pvalue(wilcoxsign_test(df$Interaction[df$Group=="SE_KO"] ~ df$Interaction[df$Grou
r_EE_SE <- statistic(wilcoxsign_test(df$Interaction[df$Group=="SE_KO"] ~ df$Interaction[df$G
  sqrt(2*sum(df$Group=="SE_KO"))

p_EE <- pvalue(wilcoxsign_test(df$Interaction[df$Group=="Enhancers_KO"] ~ df$Interaction[df$
r_EE <- statistic(wilcoxsign_test(df$Interaction[df$Group=="Enhancers_KO"] ~ df$Interaction[
  sqrt(2*sum(df$Group=="Enhancers_KO"))

p_Rad21 <- pvalue(wilcoxsign_test(df$Interaction[df$Group=="Rad21_KO"] ~ df$Interaction[df$G
r_Rad21 <- statistic(wilcoxsign_test(df$Interaction[df$Group=="Rad21_KO"] ~ df$Interaction[d
  sqrt(2*sum(df$Group=="Rad21_KO"))

p_TSS <- pvalue(wilcoxsign_test(df$Interaction[df$Group=="TSS_KO"] ~ df$Interaction[df$Group
r_TSS <- statistic(wilcoxsign_test(df$Interaction[df$Group=="TSS_KO"] ~ df$Interaction[df$Gr
  sqrt(2*sum(df$Group=="TSS_KO"))

p_rand <- pvalue(wilcoxsign_test(df$Interaction[df$Group=="Random_KO"] ~ df$Interaction[df$G
r_rand <- statistic(wilcoxsign_test(df$Interaction[df$Group=="Random_KO"] ~ df$Interaction[d
  sqrt(2*sum(df$Group=="Random_KO"))

par(cex=2, bty="n", mar=c(7,4,4,2))
boxplot(Interaction~Group, data=df, ylab="Peak interactions above background",
          outline=F, col=c("darkgrey", "firebrick2"), ylim=c(0,7), las=2)
abline(h=1, lty=2, col="red")

labs <- c(paste0("p = ", signif(p_EE_SE,3),"\n", "n = ", sum(df$Group=="SE_KO")),
          paste0("p = ", signif(p_EE,3),"\n", "n = ", sum(df$Group=="Enhancers_KO")),
          paste0("p = ", signif(p_Rad21,3),"\n", "n = ", sum(df$Group=="Rad21_KO")),
          paste0("p = ", signif(p_TSS,3),"\n", "n = ", sum(df$Group=="TSS_KO")),
          paste0("p = ", signif(p_rand,3),"\n", "n = ", sum(df$Group=="Random_KO")))
```

```
axis(side=3, at=c(1.5, 3.5, 5.5, 7.5, 9.5), labels=labs, tick=F, line=-2, cex=1)
```

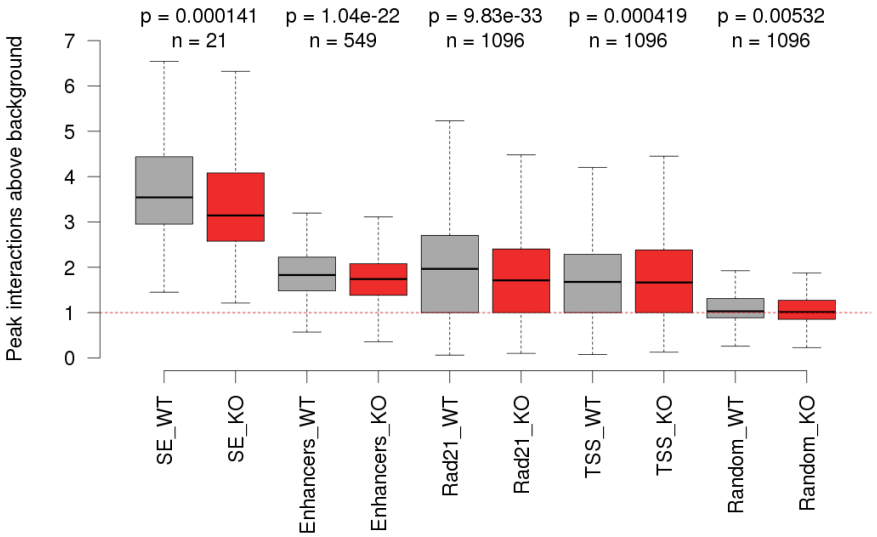

Figure 14: plot of chunk sima_summary_graph

Figure 5A – comparison to CTCF and cohesin signal in super-enhancers from other cell types

```
bed_files <- as.list(list.files("~/HiC/data/", pattern="Whyte_super_enhancers_", full.names=
bed_files <- bed_files[-grep("test|tracklines", bed_files)]

other_SE_list <- lapply(bed_files, function(x){
    tmp <- read.table(x, header = FALSE, stringsAsFactors = FALSE, col.names = c("chr", "sta
    makeGRangesFromDataFrame(tmp)
})

bed_names <- sapply(bed_files, function(x){
  ns <- strsplit(x, "/|\\.|_")
  return(ns[[1]][length(ns[[1]])-1])})
names(other_SE_list) <- bed_names
other_SE_list$all <- unlist(GRangesList(other_SE_list[1:5]))

normalised_coverage_list_short <- list(Rad21=normalised_coverage_list$Rad21,
                                       CTCF=normalised_coverage_list$CTCF_Shih,
                                       Smc1a=normalised_coverage_list$Smc1a)
```

```
size <- 100
other_SE_coverage_list <- lapply(other_SE_list, function(SE){
  tmp <- lapply(normalised_coverage_list_short, function(cov){generate.plot.matrix(cov, SE,
  names(tmp) <- names(normalised_coverage_list_short)
  return(tmp)
})

colours <- brewer.pal(5, "Set1")

par(cex=2, mar=c(2.1,2.1,1,1))
plot(colMeans(SE.plot.matrix.list$CTCF), type="l", lty=1,
     ylab= "Coverage (rpm)", xaxt="n", xlab="", col="black",
     lwd=2, main="CTCF", ylim=c(0,1))
axis(side=1, at=c(size,size*2), labels=c("Start", "End"))

lines(colMeans(other_SE_coverage_list$C2C12$CTCF), lwd=2, col=colours[2])
lines(colMeans(other_SE_coverage_list$ES$CTCF), lwd=2, col=colours[3])
lines(colMeans(other_SE_coverage_list$macrophage$CTCF), lwd=2, col=colours[4])

legend("topright", legend = c("Thymocyte", "Th", "C2C12", "ES", "macrophage", "proB")[c(1,3
       col=c("black", colours)[c(1,3:5)], lwd=2, bty="n")
```

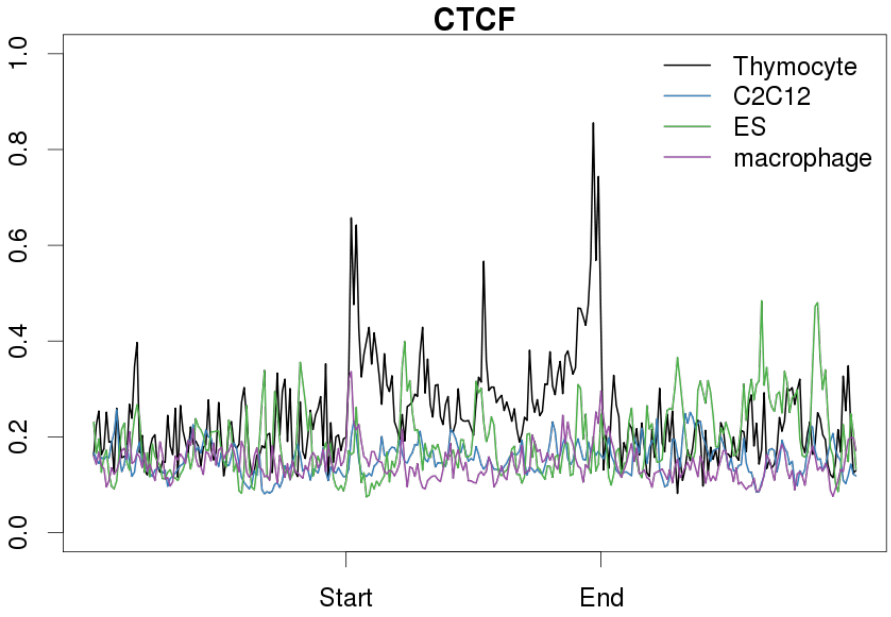

Figure 15: plot of chunk other_SE_metagenes

```
par(cex=2, mar=c(2.1,2.1,1,1))
plot(colMeans(SE.plot.matrix.list$Rad21), type="l", lty=1,
     ylab= "Coverage (rpm)", xaxt="n", xlab="", col="black",
     lwd=2, main="Rad21",ylim=c(0,1))
axis(side=1, at=c(size,size*2), labels=c("Start", "End"))

lines(colMeans(other_SE_coverage_list$C2C12$Rad21), lwd=2, col=colours[2])
lines(colMeans(other_SE_coverage_list$ES$Rad21), lwd=2, col=colours[3])
lines(colMeans(other_SE_coverage_list$macrophage$Rad21), lwd=2, col=colours[4])

legend("topright", legend = c("Thymocyte", "Th", "C2C12", "ES", "macrophage", "proB")[c(1,3
       col=c("black", colours)[c(1,3:5)], lwd=2, bty="n")
```

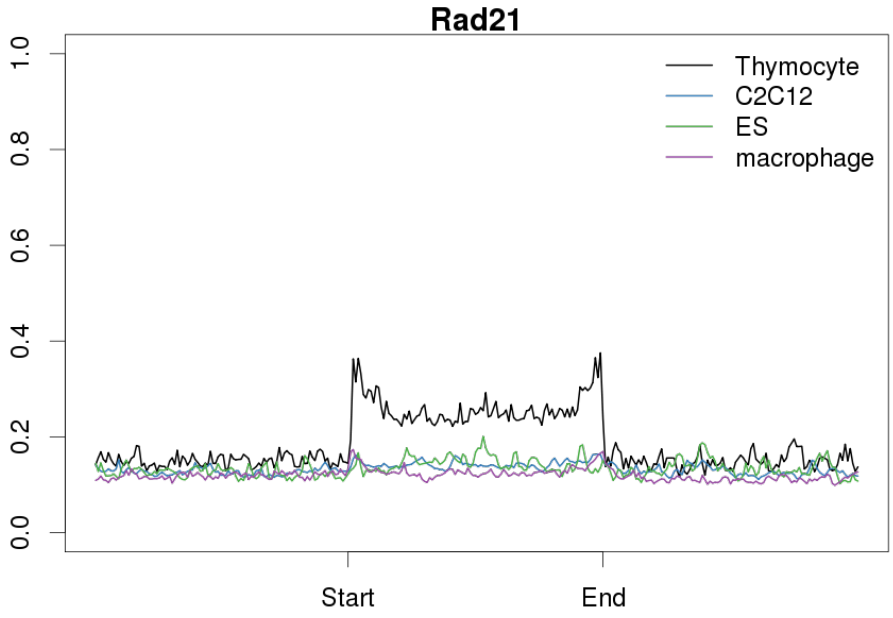

Figure 16: plot of chunk other_SE_metagenes

```
par(cex=2, mar=c(2.1,2.1,1,1))
plot(colMeans(SE.plot.matrix.list$Smc1a), type="l", lty=1,
     ylab= "Coverage (rpm)", xaxt="n", xlab="", col="black",
     lwd=2, main="Smc1a",ylim=c(0,1))
axis(side=1, at=c(size,size*2), labels=c("Start", "End"))

lines(colMeans(other_SE_coverage_list$C2C12$Smc1a), lwd=2, col=colours[2])
lines(colMeans(other_SE_coverage_list$ES$Smc1a), lwd=2, col=colours[3])
lines(colMeans(other_SE_coverage_list$macrophage$Smc1a), lwd=2, col=colours[4])
```

```
legend("topright", legend = c("Thymocyte", "Th", "C2C12", "ES", "macrophage", "proB")[c(1,3:
       col=c("black", colours)[c(1,3:5)], lwd=2, bty="n")
```

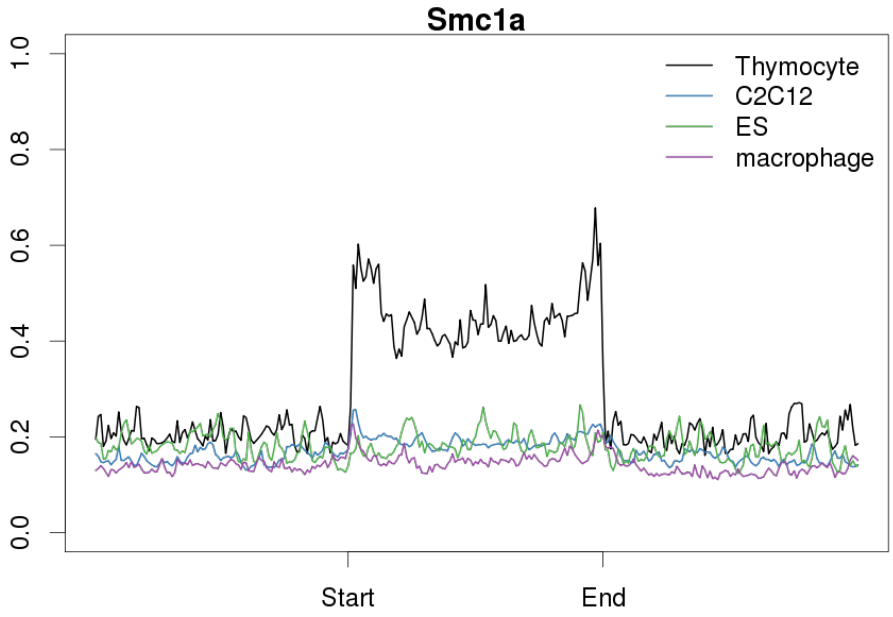

Figure 17: plot of chunk other_SE_metagenes

Figure 5B

The next set of graphs show the SIMA analysis repeated for all super-enhancers more than 50kb in length in order to compare them to super-enhancers from other cell types, all of which are less than 100kb, but 20 of which are more than 50kb. These super-enhancers for other cell types are taken from Whyte et al.

```
WT <- read.table("/mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/HiC/SIMA/SIMA_outpu
                 sep="\t", header=F, stringsAsFactors = FALSE, col.names=columns)
KO <- read.table("/mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/HiC/SIMA/SIMA_outpu
                 sep="\t", header=F, stringsAsFactors = FALSE, col.names=columns)

merged <- merge(WT, KO, by=names(WT)[1:12])
names(merged) <- sub("\\.x", "_WT", names(merged))
names(merged) <- sub("\\.y", "_KO", names(merged))

WT_other <- read.table("/mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/HiC/SIMA/50kb
```

```
                        sep="\t", header=F, stringsAsFactors = FALSE, col.names=columns)
KO_other <- read.table("/mnt/biggles/csc_projects/lizis/HiC/VladThymocytes/mm9/HiC/SIMA/50kb
                        sep="\t", header=F, stringsAsFactors = FALSE, col.names=columns)

merged_other <- merge(WT_other, KO_other, by=names(WT)[1:12])
names(merged_other) <- sub("\\.x", "_WT", names(merged_other))
names(merged_other) <- sub("\\.y", "_KO", names(merged_other))

df <- data.frame(Peaks=c(merged$PeakEnrichment_WT, merged$PeakEnrichment_KO,
                         merged_other$PeakEnrichment_WT, merged_other$PeakEnrichment_KO),
                 Random = c(merged$RandEnrichment_WT, merged$RandEnrichment_KO,
                            merged_other$RandEnrichment_WT, merged_other$RandEnrichment_KO),
                 Ratio = c(merged$Ratio_WT, merged$Ratio_KO,
                           merged_other$Ratio_WT, merged_other$Ratio_KO),

                 Group=c(rep("WT DP", length(merged$Ratio_WT)),
                         rep("KO DP", length(merged$Ratio_KO)),
                         rep("WT other", length(merged_other$Ratio_WT)),
                         rep("KO other", length(merged_other$Ratio_KO))))

df$Group <- factor(df$Group, levels=c("WT DP", "KO DP", "WT other", "KO other"))

p_SE_DP <- pvalue(wilcoxsign_test(df$Peaks[df$Group=="KO DP"] ~ df$Peaks[df$Group=="WT DP"])
r_SE_DP <- statistic(wilcoxsign_test(df$Peaks[df$Group=="KO DP"] ~ df$Peaks[df$Group=="WT DP
  sqrt(2*sum(df$Group=="KO DP"))

p_SE_other <- pvalue(wilcoxsign_test(df$Peaks[df$Group=="KO other"] ~ df$Peaks[df$Group=="WT
r_SE_other <- statistic(wilcoxsign_test(df$Peaks[df$Group=="KO other"] ~ df$Peaks[df$Group==
  sqrt(2*sum(df$Group=="KO other"))


par(cex=2, bty="n", mar=c(7,4,4,2))
boxplot(Peaks~Group, data=df, ylab="Peak interactions above background",
          outline=F, col=c("grey30", "firebrick2", "grey60", "darkorange"),
        ylim=c(0,8), las=2, xaxt="n")
abline(h=1, lty=2, col="red")

labs <- c(paste0("p = ", signif(p_SE_DP,3),"\n"),
          paste0("p = ", signif(p_SE_other,3),"\n"))
axis(side=3, at=c(1.5, 3.5), labels=labs, tick=F, line=-2, cex=1)
axis(side=1, at=1:4, labels=c("WT", "KO", "WT", "KO"), cex=1)
axis(side=1, at=c(1.5, 3.5), labels=c("thymocyte", "other"), line =1, cex=1, tick=FALSE)
```

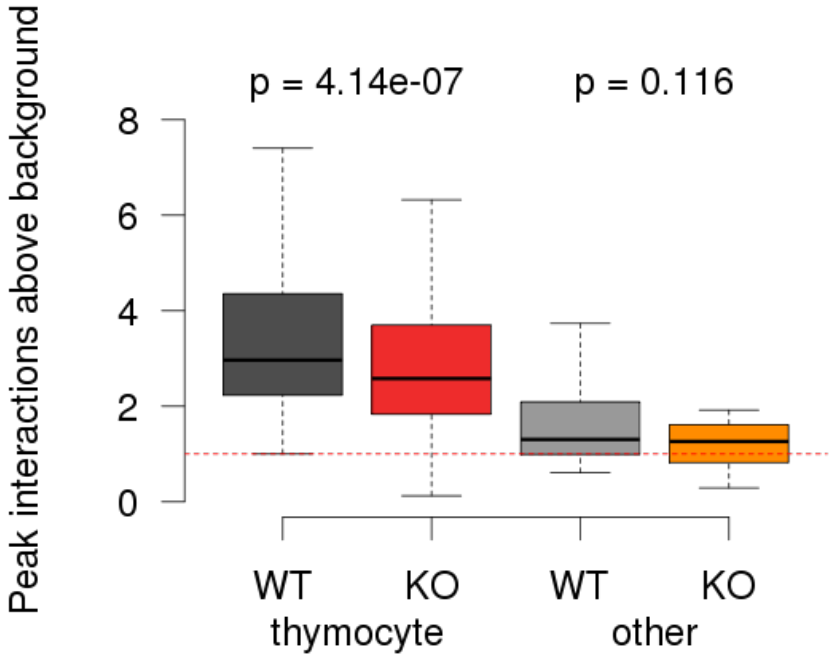

Figure 18: plot of chunk 50kb_plots

Supplementary Figures

Figure S1B

```
par(cex=2, mar=c(2.1,2.1,1,1))
plot(colMeans(SE.plot.matrix.list$Nipbl), type="l", ylab= "Coverage (rpm)",
     ylim =c(0,1), xaxt="n", xlab="", col="blue", lwd=2)
axis(side=1, at=c(size,size*2), labels=c("Start", "End"))

lines(colMeans(SE.plot.matrix.list$Med1), col="black", lwd=2)
legend("topright", legend = c("Nipbl", "Med1"), col = c("blue", "black"),
       lty=1, lwd=2, bty="n")
```

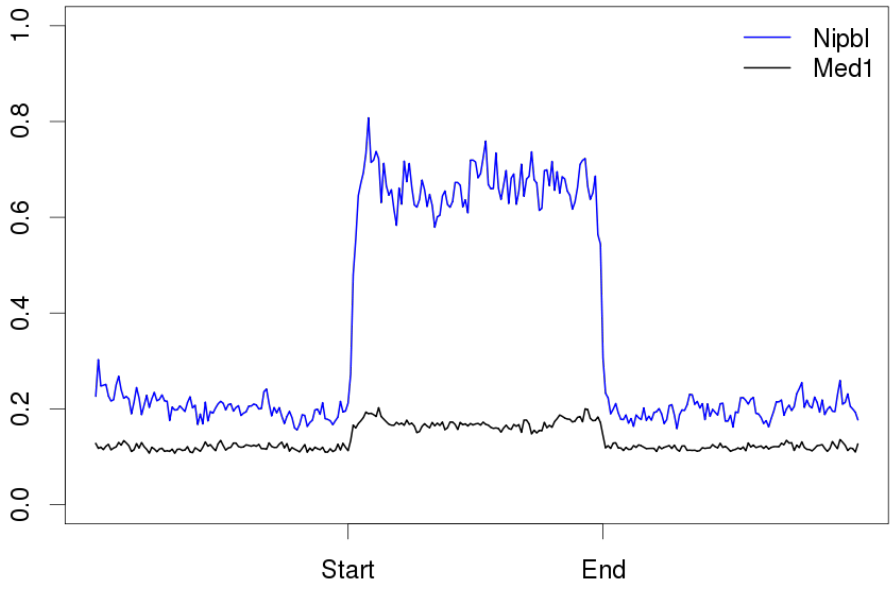

Figure 19: plot of chunk plot_extra_metagenes

Figure S1A

```
par(cex=2, mar=c(2.1,2.1,1,1))
WT <- (colMeans(SE.plot.matrix.list$WT1) + colMeans(SE.plot.matrix.list$WT2))/2
KO <- (colMeans(SE.plot.matrix.list$KO1) + colMeans(SE.plot.matrix.list$KO2))/2

plot(colMeans(SE.plot.matrix.list$H3K4me3), type="l", ylab= "Coverage (rpm)",
     ylim =c(0,3), xaxt="n", xlab="", col="blue", lwd=2)
axis(side=1, at=c(size,size*2), labels=c("Start", "End"))
```

```
lines(colMeans(SE.plot.matrix.list$H3K4me1), col="black", lwd=2)

lines(WT, col="darkred", lwd=2)
lines(KO, col="darkred", lwd=2, lty=2)

legend("topright", legend = c("WT H3K27ac", "KO H3K27ac", "H3K4me3", "H3K4me1"),
       col = c("darkred", "darkred", "blue", "black"), lty=c(1,2,1,1), lwd=2, bty="n")
```

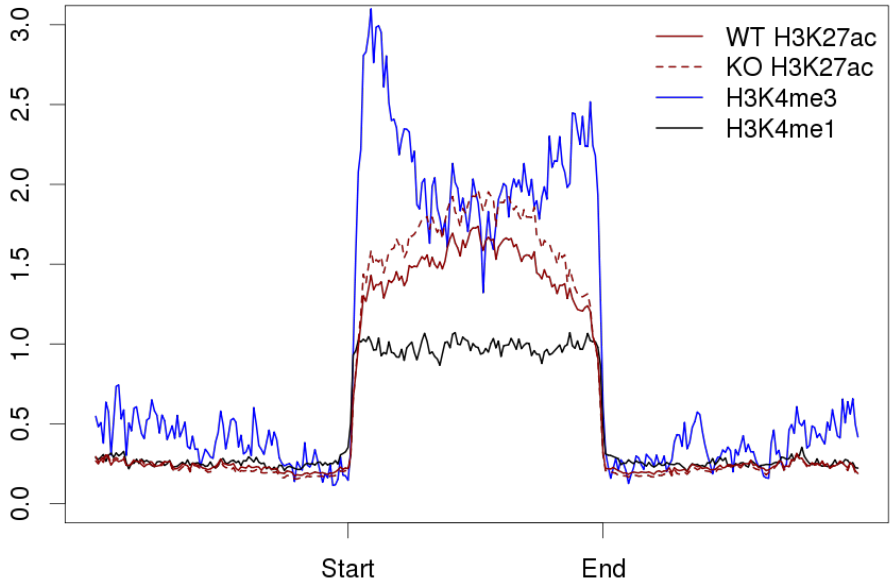

Figure 20: plot of chunk histone_metagenes

Other distance cut-offs for gene association with SEs

```
promoters <- ensGene_tss.gr[ensGene_tss.gr$ensG %in% gene_expr.df$ensG]

cutoffs <- list(0, 10000, 20000, 30000, 40000, 50000, 75000, 100000)

DE_by_distance <- lapply(cutoffs, function(cutoff){
  f0 <- findOverlaps(promoters, join_SEs, maxgap=cutoff)
  nearby_promoters <- promoters[queryHits(f0)]
  nearby_genes <- gene_expr.df[match(nearby_promoters$ensG, gene_expr.df$ensG),]
  rownames(nearby_genes) <- NULL
  nearby_genes <- unique(nearby_genes)

  up <- sum(nearby_genes$DE_up)
```

```
  down <- sum(nearby_genes$DE_down)
  total <- nrow(nearby_genes)

  return(c(up, down, total-up-down, cutoff))
})

#genome avg
up <- sum(gene_expr.df$DE_up)
down <- sum(gene_expr.df$DE_down)
total <- nrow(gene_expr.df)

#make data frame
DE_df <- as.data.frame(do.call(rbind, DE_by_distance))

DE_df <- rbind(DE_df, list(up, down, total-up-down, "Genome average"))

colnames(DE_df) <- c("Up", "Down", "No", "Cutoff")
kable(DE_df)
```

Up	Down	No	Cutoff
67	71	355	0
76	80	465	10000
84	89	583	20000
91	103	703	30000
97	110	829	40000
103	115	953	50000
116	125	1284	75000
127	139	1591	1e+05
703	450	15850	Genome average

```
DE_df <- gather(DE_df, DE, Value, -Cutoff)
DE_df$Cutoff <- factor(DE_df$Cutoff, levels=c(cutoffs, "Genome average"), ordered=TRUE)

cols <- c("No" = NA, "Up" = col_up, "Down" = col_down)

ggplot(DE_df, aes(x=Cutoff, fill=DE, y=Value)) + geom_bar(stat="identity", position="fill")
  scale_x_discrete("Max distance from SE (bp)") +
  scale_y_continuous("% of genes within max distance",labels=percent) +
  scale_fill_manual(values=cols) +
  theme_bw(base_size=20) + theme(panel.border=element_blank()) + coord_flip()


groups <- unique(DE_df$Cutoff[DE_df$Cutoff!="Genome average"])
for (g in groups){
```

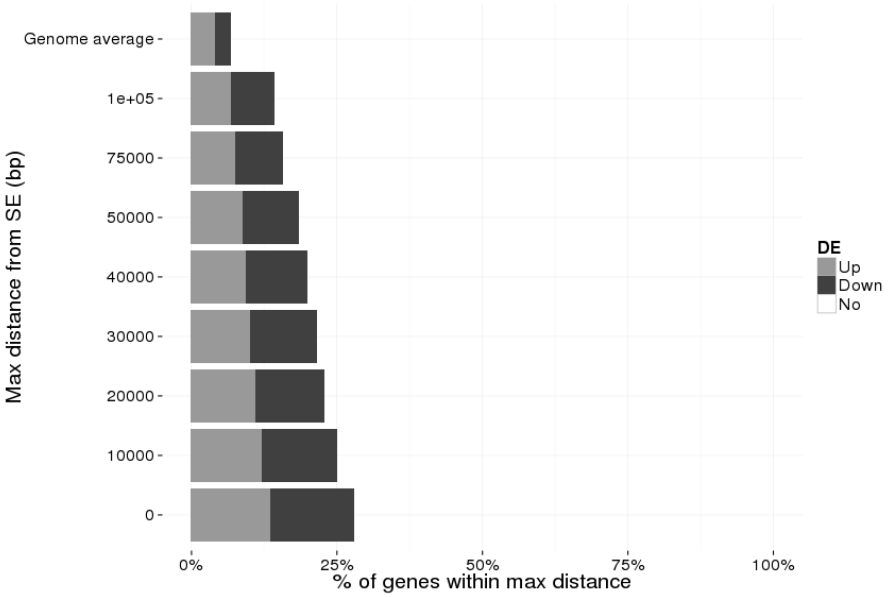

Figure 21: plot of chunk distance_cutoffs

```
 print(paste0("Comparing all expressed genes and ", g, ":"))

 DE_df %>%
   filter(Cutoff %in% c("Genome average", g)) %>%
     spread(DE, Value)%>%
     dplyr::select(No,Up,Down) %>%
   (function(x){x[2,] <- x[2,]-x[1,]
                return(x)}) %>%  print()

DE_df %>%
   filter(Cutoff %in% c("Genome average", g)) %>%
     spread(DE, Value)%>%
     dplyr::select(No,Up,Down) %>%
   (function(x){x[2,] <- x[2,]-x[1,]
                return(x)}) %>%
   as.matrix() %>%
   chisq.test() %>% print()
 }
```

```
## [1] "Comparing all expressed genes and 0:"
##       No  Up Down
## 1   355  67   71
## 2 15495 636  379
```

```
##
##  Pearson's Chi-squared test
##
## data:  DE_df %>% filter(Cutoff %in% c("Genome average", g)) %>% spread(DE,     Value) %>%
## X-squared = 399.3844, df = 2, p-value < 2.2e-16
##
## [1] "Comparing all expressed genes and 10000:"
##      No  Up Down
## 1   465  76   80
## 2 15385 627  370
##
##  Pearson's Chi-squared test
##
## data:  DE_df %>% filter(Cutoff %in% c("Genome average", g)) %>% spread(DE,     Value) %>%
## X-squared = 380.7899, df = 2, p-value < 2.2e-16
##
## [1] "Comparing all expressed genes and 20000:"
##      No  Up Down
## 1   583  84   89
## 2 15267 619  361
##
##  Pearson's Chi-squared test
##
## data:  DE_df %>% filter(Cutoff %in% c("Genome average", g)) %>% spread(DE,     Value) %>%
## X-squared = 364.1092, df = 2, p-value < 2.2e-16
##
## [1] "Comparing all expressed genes and 30000:"
##      No  Up Down
## 1   703  91  103
## 2 15147 612  347
##
##  Pearson's Chi-squared test
##
## data:  DE_df %>% filter(Cutoff %in% c("Genome average", g)) %>% spread(DE,     Value) %>%
## X-squared = 384.4905, df = 2, p-value < 2.2e-16
##
## [1] "Comparing all expressed genes and 40000:"
##      No  Up Down
## 1   829  97  110
## 2 15021 606  340
##
##  Pearson's Chi-squared test
##
## data:  DE_df %>% filter(Cutoff %in% c("Genome average", g)) %>% spread(DE,     Value) %>%
## X-squared = 358.4209, df = 2, p-value < 2.2e-16
##
```

```
## [1] "Comparing all expressed genes and 50000:"
##      No  Up Down
## 1   953 103  115
## 2 14897 600  335
##
##  Pearson's Chi-squared test
##
## data:  DE_df %>% filter(Cutoff %in% c("Genome average", g)) %>% spread(DE,     Value) %>%
## X-squared = 329.5514, df = 2, p-value < 2.2e-16
##
## [1] "Comparing all expressed genes and 75000:"
##      No  Up Down
## 1  1284 116  125
## 2 14566 587  325
##
##  Pearson's Chi-squared test
##
## data:  DE_df %>% filter(Cutoff %in% c("Genome average", g)) %>% spread(DE,     Value) %>%
## X-squared = 258.4564, df = 2, p-value < 2.2e-16
##
## [1] "Comparing all expressed genes and 1e+05:"
##      No  Up Down
## 1  1591 127  139
## 2 14259 576  311
##
##  Pearson's Chi-squared test
##
## data:  DE_df %>% filter(Cutoff %in% c("Genome average", g)) %>% spread(DE,     Value) %>%
## X-squared = 234.0145, df = 2, p-value < 2.2e-16
```

Session details

This report was generated on Wed Dec 17 2014 at 16:07:40.

```
sessionInfo()
```

```
## R version 3.1.1 (2014-07-10)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_GB.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8        LC_COLLATE=en_GB.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_GB.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C
```

```
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
##  [1] splines   grid      stats4    parallel  stats     graphics  grDevices
##  [8] utils     datasets  methods   base
##
## other attached packages:
##  [1] VennDiagram_1.6.9                ppcor_1.0
##  [3] org.Mm.eg.db_3.0.0               RSQLite_1.0.0
##  [5] GenomicAlignments_1.2.1          DBI_0.3.1
##  [7] BSgenome.Mmusculus.UCSC.mm9_1.4.0 BSgenome_1.34.0
##  [9] GenomicFiles_1.2.0               BiocParallel_1.0.0
## [11] rtracklayer_1.26.2               GenomicFeatures_1.18.2
## [13] AnnotationDbi_1.28.1             Biobase_2.26.0
## [15] RColorBrewer_1.1-2               nnet_7.3-8
## [17] coin_1.0-24                      survival_2.37-7
## [19] tidyr_0.1                        dplyr_0.3.0.2
## [21] scales_0.2.4                     ggplot2_1.0.0
## [23] Rsamtools_1.18.2                 Biostrings_2.34.0
## [25] XVector_0.6.0                    genomation_0.99.7
## [27] GenomicRanges_1.18.3             GenomeInfoDb_1.2.3
## [29] IRanges_2.0.0                    S4Vectors_0.4.0
## [31] BiocGenerics_0.12.1              knitr_1.8
##
## loaded via a namespace (and not attached):
##  [1] assertthat_0.1    base64enc_0.1-2   BatchJobs_1.5
##  [4] BBmisc_1.8        biomaRt_2.22.0    bitops_1.0-6
##  [7] brew_1.0-6        checkmate_1.5.0   chron_2.3-45
## [10] codetools_0.2-9   colorspace_1.2-4  data.table_1.9.4
## [13] digest_0.6.4      evaluate_0.5.5    fail_1.2
## [16] foreach_1.4.2     formatR_1.0       gridBase_0.4-7
## [19] gtable_0.1.2      impute_1.40.0     iterators_1.0.7
## [22] labeling_0.3      lazyeval_0.1.9    magrittr_1.5
## [25] MASS_7.3-35       modeltools_0.2-21 munsell_0.4.2
## [28] mvtnorm_1.0-1     plyr_1.8.1        proto_0.3-10
## [31] Rcpp_0.11.3       RCurl_1.95-4.5    reshape2_1.4
## [34] sendmailR_1.2-1   stringr_0.6.2     tools_3.1.1
## [37] XML_3.98-1.1      zlibbioc_1.12.0
```