Red Blood Cell Traits-associated genetic variation MPRA data analysis

Dandan Peng Zhi Ye

Biological Setting

Genome-wide association studies (GWAS) have successfully identified over 10,000 common SNPs associated with hundreds of human traits and disease. Each Genome-wide association studies "hit" usually represents, or tags, hundreds of variants that are inherited together across a large (many are up to ~0.5 megabase) genomic region, termed a linkage disequilibrium (LD) block, often containing numerous protein-coding genes. It is estimated that ~80% of the phenotypic heritability in common diseases and traits can be explained by non-coding regulatory variants (85%-90% of GWAS hits tag only non-coding variants), making target-gene identification and subsequent biological inference a considerable challenge. However, these GWAS-nominated variants are significantly enriched at cell-type specific regulatory regions such as DNase I hypersensitivity sites (DHS) and transcription factor (TF) occupancy sites, suggesting the attractive hypothesis that many of these variants may alter the regulation of gene transcription.

Massively parallel reporter assays (MPRA) have emerged as a high-throughput means of measuring the ability of sequences to drive expression. These assays build on the traditional reporter assay framework by coupling each putative regulatory sequence with several short DNA tags, or barcodes, that are incorporated into the RNA output. These tags are counted in the RNA reads and the input DNA, and the resulting counts are used to quantify the activity of a given putative regulatory sequence, typically involving the ratio of RNA counts to DNA counts.

Here we find a paper related to MPRA data analysis, "Systematic Functional Dissection of Common Genetic Variation Affecting Red Blood Cell Traits". Then we reproduce the whole MPRA data analysis procedure and do a data analysis from MPRA raw data which provided in the paper. Finally, we identify 32 functional variants based on GWAS hits which are similar to the results in the paper. Some analysis code are modified based on the provided R code.

Experiment and Data description

The data we worked on is from a study of allelic effects in GWAS to understand red blood cell (RBC) traits. They modified a recently designed MPRA to simultaneously screen regulatory effects in 2756 variants in high LD with 75 GWAS hits from a comprehensive study of RBC traits. A total of six replicates were performed for the MPRA screen in K562 cells and four replicates in K562+GATA1 cells. Each construct was assigned 14 unique, designed barcodes. GATA1 is a transcription factor critical for normal megakaryocytopoiesis, and several different causative mutations have been identified within the region encoding its N-terminal zinc finger domain.

Read the MPRA raw data, and here are the first 6 lines of raw data, each line represents one barcode:

```r
dir = "./data/"
MPRA <- data.frame(read_delim(file = paste0(dir, "Raw/", "RBC_MPRA_minP_raw.txt"),
    delim = "\t", col_names = T, col_types = cols(chr = "c")))
head(MPRA)
```

```
##   chr     pos ref alt type bot top clean     oligo     construct
## 1   1 3684954   C   A  Ref 2/3 1/3   var 1 3684954 1 3684954 2/3
## 2   1 3684954   C   A  Ref 2/3 1/3   var 1 3684954 1 3684954 2/3
## 3   1 3684954   C   A  Ref 2/3 1/3   var 1 3684954 1 3684954 2/3
## 4   1 3684954   C   A  Ref 2/3 1/3   var 1 3684954 1 3684954 2/3
## 5   1 3684954   C   A  Ref 2/3 1/3   var 1 3684954 1 3684954 2/3
## 6   1 3684954   C   A  Ref 2/3 1/3   var 1 3684954 1 3684954 2/3
##            byallele K562_minP_DNA1 K562_minP_DNA2 K562_CTRL_minP_RNA1
## 1 1 3684954 2/3 Ref             33             28                   2
## 2 1 3684954 2/3 Ref             58             40                   6
## 3 1 3684954 2/3 Ref             57             35                   8
## 4 1 3684954 2/3 Ref            114             74                  61
## 5 1 3684954 2/3 Ref             36             23                  34
## 6 1 3684954 2/3 Ref             34             20                   5
##   K562_CTRL_minP_RNA2 K562_CTRL_minP_RNA3 K562_CTRL_minP_RNA4
## 1                   7                  39                   2
## 2                   6                  11                 145
## 3                  17                  68                  57
## 4                  15                 106                  13
## 5                  12                   1                   1
## 6                   1                   7                  33
##   K562_CTRL_minP_RNA5 K562_CTRL_minP_RNA6 K562_GATA1_minP_RNA1
## 1                  32                   7                   31
## 2                  38                  10                   78
## 3                  19                  46                   12
## 4                 163                 244                    4
## 5                  20                  91                    0
## 6                  34                  99                    2
##   K562_GATA1_minP_RNA2 K562_GATA1_minP_RNA3 K562_GATA1_minP_RNA4
## 1                    0                   23                   10
## 2                   53                   35                   40
## 3                    7                    4                  123
## 4                   38                   77                   73
## 5                    1                   39                  158
## 6                   31                   33                   10
```

Counts data preprocessing

Modified constructs were removed due to restriction enzyme cut sites. Plasmid counts were combined from both replicates. Barcodes with fewer than 8 transformed counts were removed from each replicate:

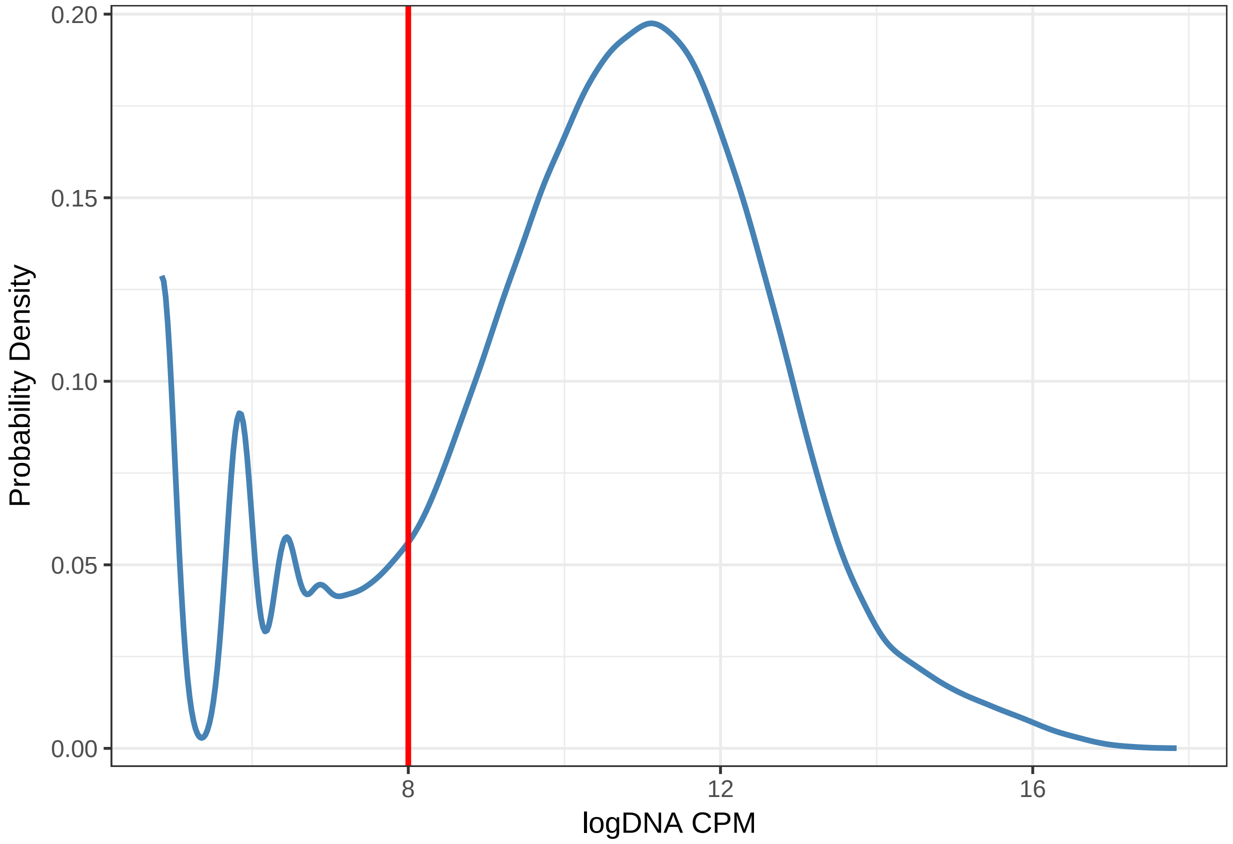
```
MPRA <- subset(MPRA, clean == "var")
MPRA <- as.data.frame(append(MPRA, list(K562_minP_DNA = MPRA$K562_minP_DNA1 +
    MPRA$K562_minP_DNA2), after = 13))
```

A pseudocount of 1 was added to DNA and RNA barcode counts that were subsequently normalized to counts per million (CPM):

```
for (i in 12:length(MPRA)) {
    MPRA[, i] <- (MPRA[, i] + 1)/1e+06 * sum(MPRA[, i])
}
```

Log normalize the raw data and check plasmid barcode abundance, we can observe that most barcodes are abundant (with more than 8 transformed counts) in the pooled library (>80%):

```
MPRA[(length(MPRA) + 1):(length(MPRA) + length(MPRA[, 12:length(MPRA)]))] <- log(MPRA[,
12:(length(MPRA))], 2)

ggplot(MPRA, aes(x = K562_minP_DNA.1), title = "Density of DNA Counts") +
  stat_density(geom="line", size = 1.0, colour = "steel blue") +
  geom_vline(aes(xintercept=8), colour = "red", size = 1.0) +
  scale_y_continuous(expand = c(0, 0.005)) +
  theme(axis.title.x = element_blank()) +
  theme(plot.background = element_blank(),panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), panel.border = element_blank()) +
  theme_bw() + labs(x = "logDNA CPM", y = "Probability Density")
```



Barcodes with fewer than 8 transformed counts were removed from each replicate:

```
MPRA_minP <- MPRA[MPRA$K562_minP_DNA.1 >= 8, ]
print(c("Percent of barcodes remaining: ", (dim(MPRA_minP)/dim(MPRA))[1]))
```

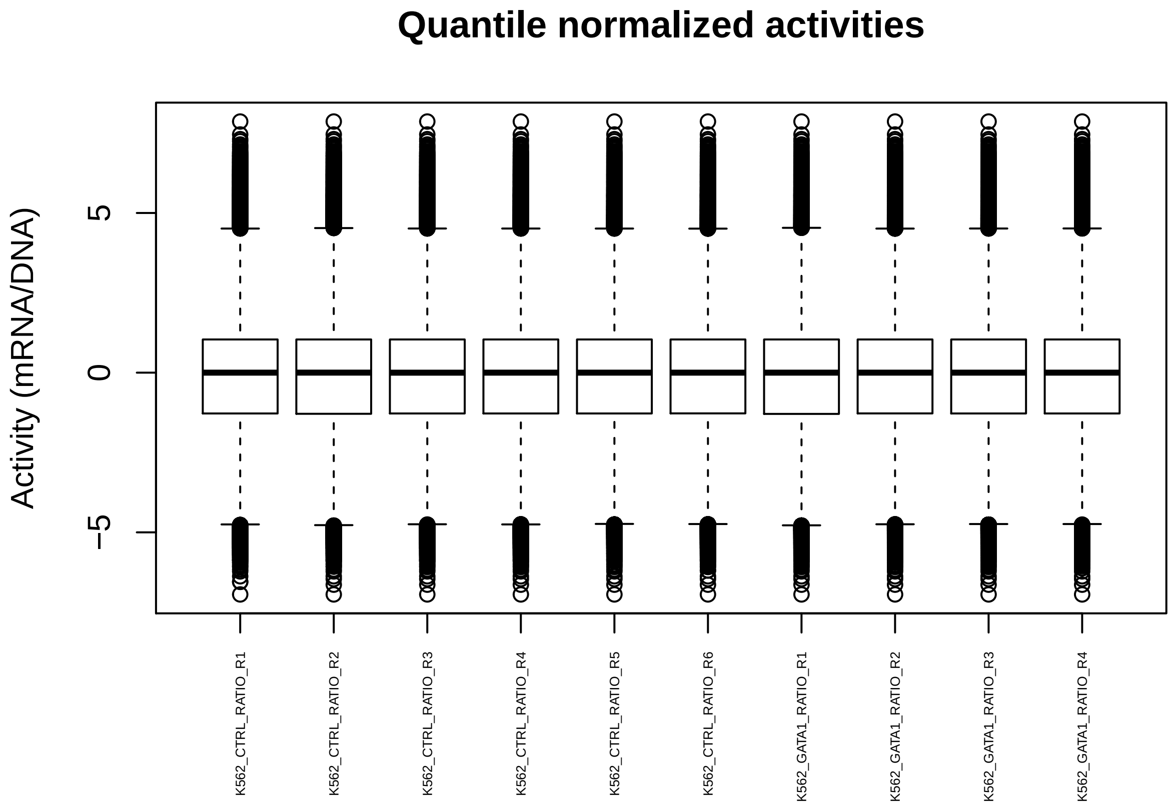
```
## [1] "Percent of barcodes remaining: " "0.82615502963935"
```

Define activity [log2(mRNA/DNA) = log2(mRNA) - log2(DNA)]:

```
attach(MPRA_minP)
MPRA_minP$K562_CTRL_RATIO_R1 <- K562_CTRL_minP_RNA1.1 - K562_minP_DNA.1
MPRA_minP$K562_CTRL_RATIO_R2 <- K562_CTRL_minP_RNA2.1 - K562_minP_DNA.1
MPRA_minP$K562_CTRL_RATIO_R3 <- K562_CTRL_minP_RNA3.1 - K562_minP_DNA.1
MPRA_minP$K562_CTRL_RATIO_R4 <- K562_CTRL_minP_RNA4.1 - K562_minP_DNA.1
MPRA_minP$K562_CTRL_RATIO_R5 <- K562_CTRL_minP_RNA5.1 - K562_minP_DNA.1
MPRA_minP$K562_CTRL_RATIO_R6 <- K562_CTRL_minP_RNA6.1 - K562_minP_DNA.1
MPRA_minP$K562_GATA1_RATIO_R1 <- K562_GATA1_minP_RNA1.1 - K562_minP_DNA.1
MPRA_minP$K562_GATA1_RATIO_R2 <- K562_GATA1_minP_RNA2.1 - K562_minP_DNA.1
MPRA_minP$K562_GATA1_RATIO_R3 <- K562_GATA1_minP_RNA3.1 - K562_minP_DNA.1
MPRA_minP$K562_GATA1_RATIO_R4 <- K562_GATA1_minP_RNA4.1 - K562_minP_DNA.1
detach(MPRA_minP)
```

Replicates were quantile normalized and combined as independent observations:

```r
temp <- normalize.quantiles(as.matrix(MPRA_minP[, 38:47]))
temp <- temp - median(temp)
MPRA_minP.melt <- melt(MPRA_minP[c("chr", "pos", "ref", "alt", "type", "bot",
    "top", "clean", "oligo", "construct", "byallele", "K562_CTRL_RATIO_R1",
    "K562_CTRL_RATIO_R2", "K562_CTRL_RATIO_R3", "K562_CTRL_RATIO_R4", "K562_CTRL_RATIO_R5",
    "K562_CTRL_RATIO_R6", "K562_GATA1_RATIO_R1", "K562_GATA1_RATIO_R2", "K562_GATA1_RATIO_R3",
    "K562_GATA1_RATIO_R4")], id = c("chr", "pos", "ref", "alt", "type", "bot",
    "top", "clean", "oligo", "construct", "byallele"))
MPRA_minP.melt$value <- melt(temp[, 1:10])$value
boxplot(value ~ variable, MPRA_minP.melt, las = 2, cex.axis = 0.4, yaxt = "n",
    main = "Quantile normalized activities", ylab = "Activity (mRNA/DNA)")
axis(2, cex = 1.5)
```



Activity was calculated by collapsing barcodes and taking the median:

```r
CTRL.temp <- MPRA_minP.melt[grep("CTRL", MPRA_minP.melt$variable), ]
CTRL.value <- tapply(CTRL.temp$value, factor(CTRL.temp$byallele), median)
GATA1.temp <- MPRA_minP.melt[grep("GATA1", MPRA_minP.melt$variable), ]
GATA1.value <- tapply(GATA1.temp$value, factor(GATA1.temp$byallele), median)
RATIO.temp <- MPRA_minP.melt[!duplicated(MPRA_minP.melt$byallele), !(names(MPRA_minP.melt) %in%
    c("value", "variable"))]
MPRA_minP.ratio <- merge(RATIO.temp, data.frame(byallele = names(CTRL.value),
    CTRL.median = CTRL.value), by = "byallele")
MPRA_minP.ratio <- merge(MPRA_minP.ratio, data.frame(byallele = names(GATA1.value),
    GATA1.median = GATA1.value), by = "byallele")
```

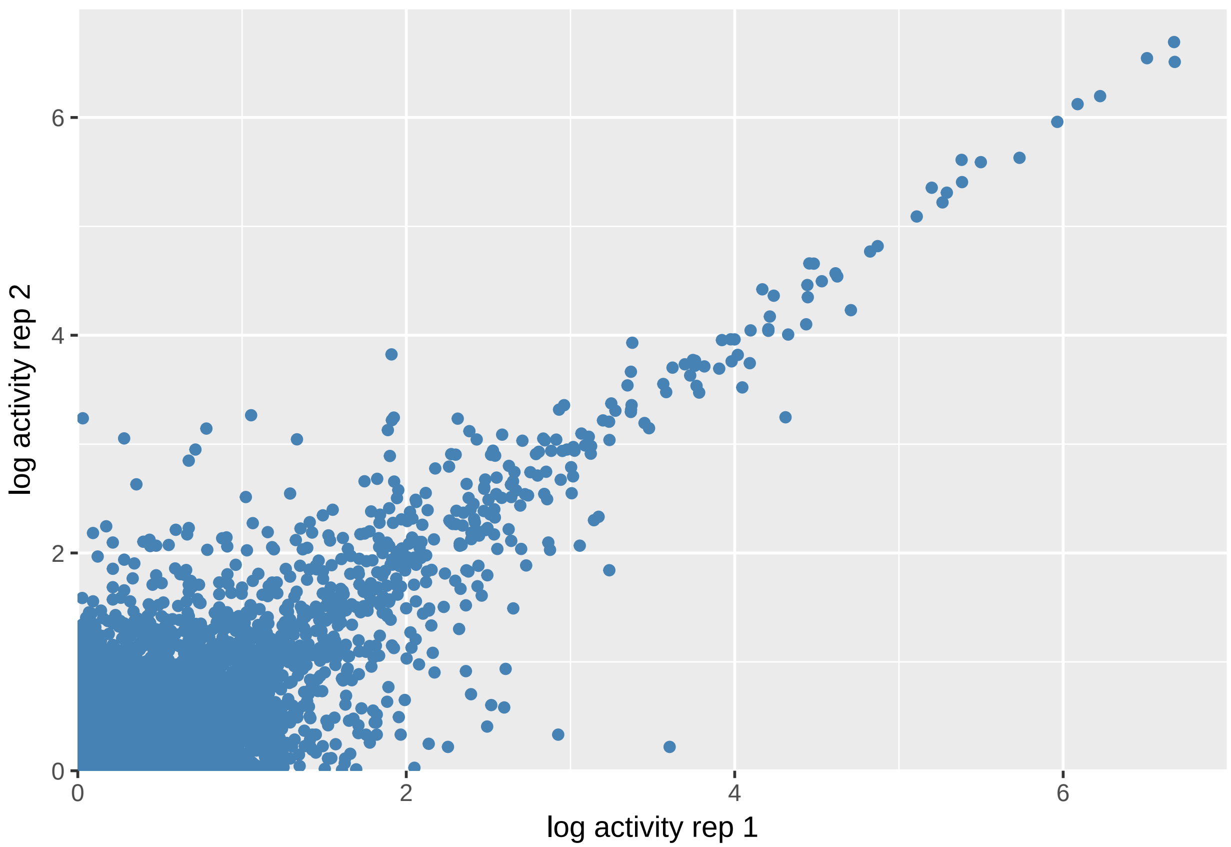
Identify endogenous regulatory elements

Calculate activity correlations, a quick way to estimate active constructs:

```
CTRL_activity_reps <- tapply(CTRL.temp$value, list(factor(CTRL.temp$byallele),
    factor(CTRL.temp$variable)), median)
max00 <- as.vector(quantile(as.matrix(CTRL_activity_reps), c(0)))
max50 <- as.vector(quantile(as.matrix(CTRL_activity_reps), c(0.5)))
CTRL_activity_reps.cor00 <- cor(CTRL_activity_reps[apply(CTRL_activity_reps,
    MARGIN = 1, function(x) all(x > max00)), ])
CTRL_activity_reps.cor50 <- cor(CTRL_activity_reps[apply(CTRL_activity_reps,
    MARGIN = 1, function(x) all(x > max50)), ])
GATA1_activity_reps <- tapply(GATA1.temp$value, list(factor(GATA1.temp$byallele),
    factor(GATA1.temp$variable)), median)
max00 <- as.vector(quantile(as.matrix(GATA1_activity_reps), c(0)))
max50 <- as.vector(quantile(as.matrix(GATA1_activity_reps), c(0.5)))
GATA1_activity_reps.cor00 <- cor(GATA1_activity_reps[apply(GATA1_activity_reps,
    MARGIN = 1, function(x) all(x > max00)), ])
GATA1_activity_reps.cor50 <- cor(GATA1_activity_reps[apply(GATA1_activity_reps,
    MARGIN = 1, function(x) all(x > max50)), ])
```

Correlation between the top 50% activity constructs from two replicates was plotted. We can see that highly active regulatory elements show good reproducibility and lowly active regulatory elements show poor reproducibility. This can be partly attibuted to the larger log2 variance at low count barcodes:

```
ggplot(as.data.frame(CTRL_activity_reps), aes(CTRL_activity_reps[, 1], CTRL_activity_reps[,
    2])) + geom_point(color = "steel blue") + scale_x_continuous(expand = c(0,
    0), limits = c(0, 7)) + scale_y_continuous(expand = c(0, 0), limits = c(0,
    7)) + labs(x = "log activity rep 1", y = "log activity rep 2")
```



Previous study has verified the function of three mutant elements in human erythroid disorders, which disrupt a canonical binding motif for the hematopoietic transcription factor GATA1. Deletions of two to four nucleotides in these elements resulted in a decrease in target gene expression. Here we set them as control elements and first investigated the activity of each control element to confirm that we could identify active regulatory elements. From the activity boxplots of the five positive control variants in K562 cell, we can see that constructs with intact GATA1-binding sites (Ref) show increased activity when compared with broken binding sites (Mut).

```
controls <- c("1 155271258", "X 55054634","X 55054635", "X 55054636","10 127505272")
MPRA_minP.ratio <-as.data.frame(append(MPRA_minP.ratio,
    list(controls = ifelse(MPRA_minP.ratio$oligo %in% controls, 1, 0)), after = 11))
MPRA_minP.ratio <- MPRA_minP.ratio[!(MPRA_minP.ratio$construct %in% "1 159174683"),]
```

After confirmation, now we can examine all active constructs (ACs) in the assay. ACs are defined as elements with activity significantly greater than the activity distribution formed from all investigated constructs (FDR<1%) and representing fewer than 4% of the tested constructs, show a similar activity distribution to non-muted controls.Here we used 1-sided Mann-Whitney-U test versus all others. Since the level of factor byallele amounts to 15752, so here we just use the first 100 levels in the for loop to calculate p-values for overall enhancer activity.

```
MPRA_minP.control.final <- MPRA_minP.melt[grep("CTRL", MPRA_minP.melt$variable), ]
MPRA_minP.control.final.pvalues <- data.frame(matrix(ncol = 3))
colnames(MPRA_minP.control.final.pvalues) <- c("byallele", "oligo", "Control_P")
for (j in levels(factor(MPRA_minP.control.final$byallele))[1:100]) {
if (length(MPRA_minP.control.final[MPRA_minP.control.final$byallele == j, ]$value) >=7) {
MPRA_minP.control.final.pvalues <- rbind(MPRA_minP.control.final.pvalues,
data.frame(byallele = j,
oligo =
MPRA_minP.control.final[MPRA_minP.control.final$byallele == j, ]$oligo[1],
Control_P = wilcox.test(
MPRA_minP.control.final[MPRA_minP.control.final$byallele == j, ]$value,
MPRA_minP.control.final[MPRA_minP.control.final$byallele != j, ]$value,
alternative = c("greater")
)$p.value))
}
}

saveRDS(MPRA_minP.control.final.pvalues, file = "MPRA_minP.control.final.pvalues.rds")
```

Calculate p-values for differential activity between alleles to do 2-sided Mann-Whitney-U test between alleles to determine alleles with difference in activity:

```
MPRA_minP.control.final <- MPRA_minP.melt[grep("CTRL", MPRA_minP.melt$variable),]
MPRA_minP.control.final.mut.pvalues <- data.frame(matrix(ncol = 3))
colnames(MPRA_minP.control.final.mut.pvalues) <-
c("construct", "oligo", "Control_P")
for (j in levels(factor(MPRA_minP.control.final$construct))[1:100]) {
if (nlevels(factor(MPRA_minP.control.final[MPRA_minP.control.final$construct ==
j, ]$type)) == 2) {
MPRA_minP.control.final.mut.pvalues <-
rbind(
MPRA_minP.control.final.mut.pvalues,
data.frame(
construct = j,
oligo = MPRA_minP.control.final[MPRA_minP.control.final$byallele == j, ]$oligo[1],
Control_P = wilcox.test(value ~ factor(type), data = MPRA_minP.control.final[MPRA_minP.control.final$co
j, ])$p.value))
}
}

saveRDS(MPRA_minP.control.final.mut.pvalues, file = "MPRA_minP.control.final.mut.pvalues.rds")
```

Read in already calculated p-values for activity and differential activity by allele:

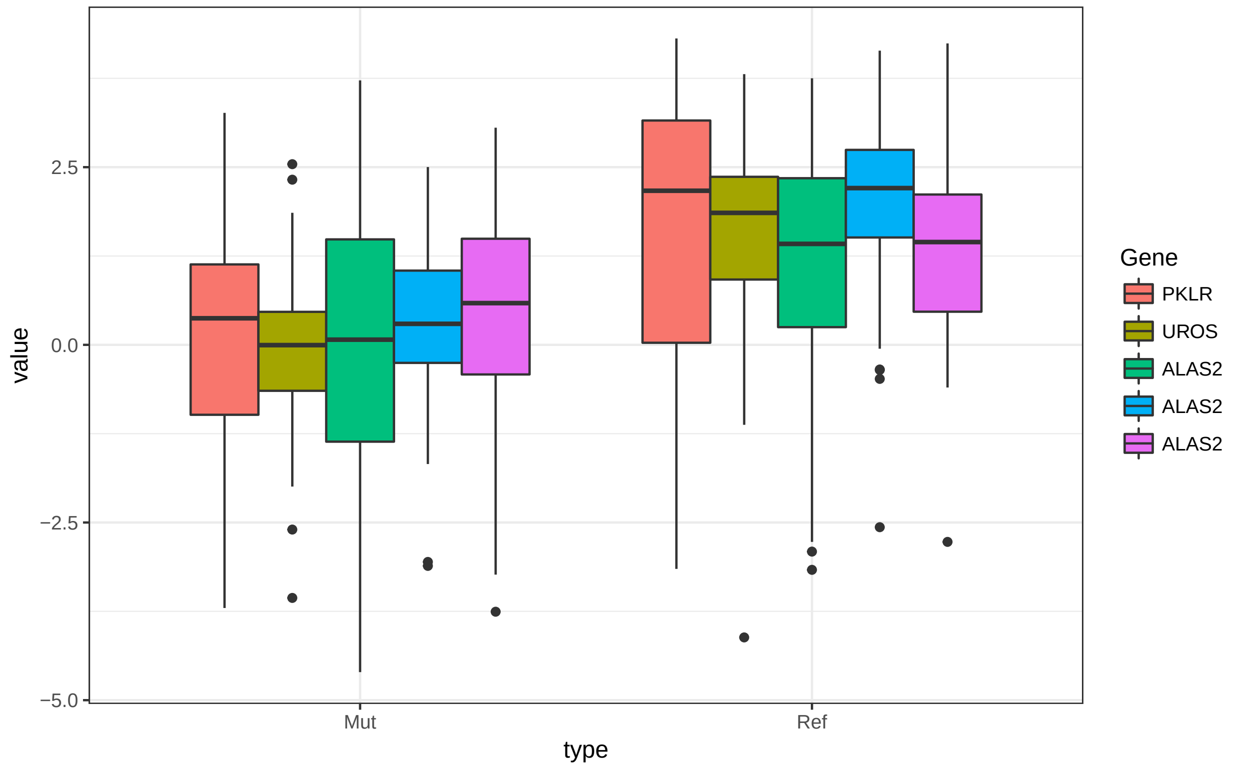
```
MPRA_minP.CTRL.pvalues <- readRDS(
  paste0(dir, "Precomputed/","MPRA_minP.control.final.pvalues.rds"))
MPRA_minP.GATA1.pvalues <- readRDS(
  paste0(dir, "Precomputed/","MPRA_minP.GATA1.final.pvalues.rds"))
MPRA_minP.CTRL.mut.pvalues <- readRDS(
  paste0(dir, "Precomputed/","MPRA_minP.control.final.mut.pvalues.rds"))
MPRA_minP.GATA1.mut.pvalues <- readRDS(
  paste0(dir, "Precomputed/","MPRA_minP.GATA1.final.mut.pvalues.rds"))
```

Merge test results and calculate FDR:

```
MPRA_minP.ratio <- merge(MPRA_minP.ratio, MPRA_minP.CTRL.pvalues, by = c("byallele",
    "oligo"))
MPRA_minP.ratio <- merge(MPRA_minP.ratio, MPRA_minP.GATA1.pvalues, by = c("byallele",
    "oligo"))
colnames(MPRA_minP.ratio)[15:16] <- c("CTRL.p", "GATA1.p")
MPRA_minP.ratio$CTRL.padj <- qvalue(MPRA_minP.ratio$CTRL.p)[3]$qvalues
MPRA_minP.ratio$GATA1.padj <- qvalue(MPRA_minP.ratio$GATA1.p)[3]$qvalues
MPRA_minP.ratio <- merge(MPRA_minP.ratio, MPRA_minP.CTRL.mut.pvalues[, c(1,
    3)], by = c("construct"))
MPRA_minP.ratio <- merge(MPRA_minP.ratio, MPRA_minP.GATA1.mut.pvalues[, c(1,
    3)], by = c("construct"))
colnames(MPRA_minP.ratio)[19:20] <- c("CTRL.mut.p", "GATA1.mut.p")
MPRA_minP.ratio$CTRL.mut.padj <- qvalue(MPRA_minP.ratio$CTRL.mut.p)[3]$qvalues
MPRA_minP.ratio$GATA1.mut.padj <- qvalue(MPRA_minP.ratio$GATA1.mut.p)[3]$qvalues
```

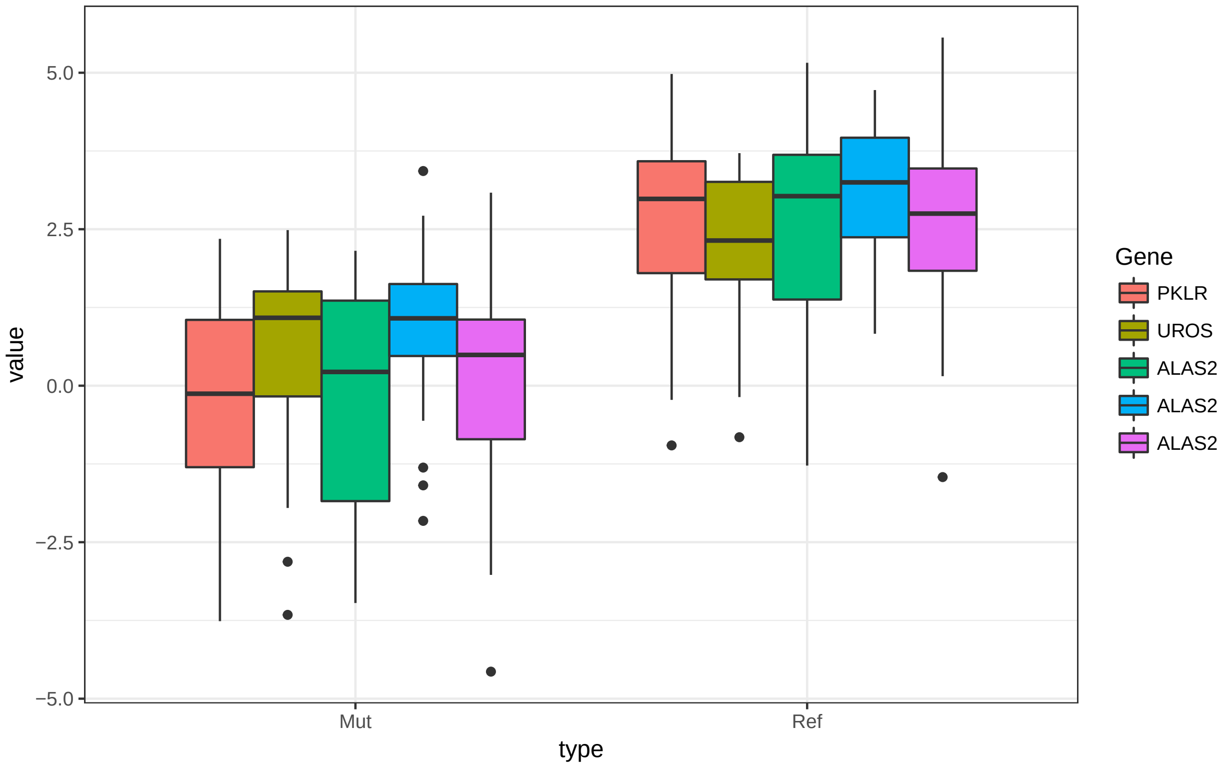
Boxplots of Mendelian positive control variants for K562:

```
ggplot(data = CTRL.temp[CTRL.temp$oligo == controls, ], aes(x = type, y = value)) +
    geom_boxplot(aes(fill = oligo)) + theme(plot.background = element_blank()) +
    theme_bw() + scale_fill_discrete(name = "Gene", labels = c("PKLR", "UROS",
    "ALAS2", "ALAS2", "ALAS2"))
```



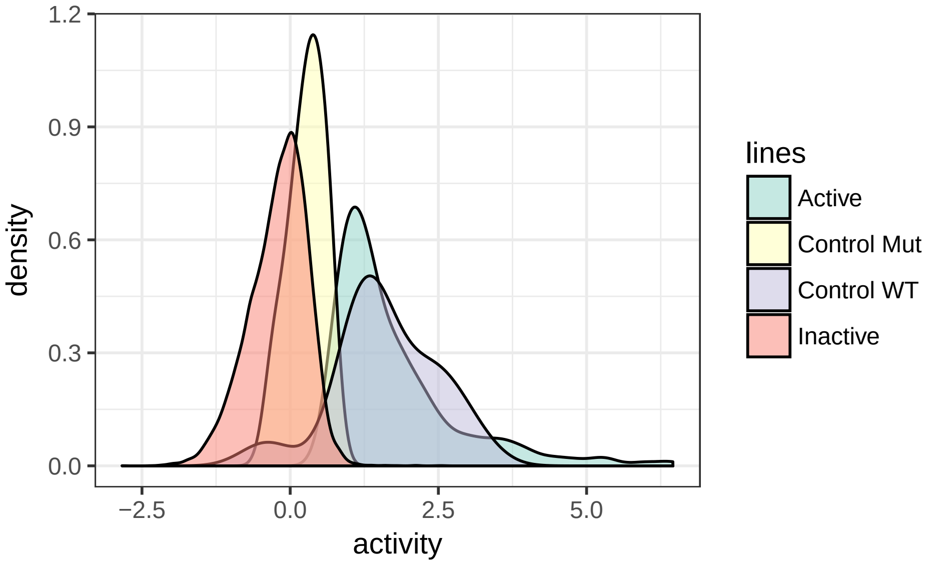
Boxplots of Mendelian positive control variants for K562+GATA1:

```
ggplot(data = GATA1.temp[GATA1.temp$oligo == controls, ], aes(x = type, y = value)) +
    geom_boxplot(aes(fill = oligo)) + theme(plot.background = element_blank()) +
    theme_bw() + scale_fill_discrete(name = "Gene", labels = c("PKLR", "UROS",
    "ALAS2", "ALAS2", "ALAS2"))
```



Density plots of active/inactive constructs and Mendelian controls (ref/mut) in K562:

```
dat <- data.frame(dens = c(
  MPRA_minP.ratio[MPRA_minP.ratio$CTRL.padj <= 0.01,]$CTRL.median,
  MPRA_minP.ratio[MPRA_minP.ratio$CTRL.padj > 0.01,]$CTRL.median,
  MPRA_minP.ratio[MPRA_minP.ratio$type == "Mut" &
  MPRA_minP.ratio$controls == 1,]$CTRL.median,
  MPRA_minP.ratio[MPRA_minP.ratio$type == "Ref" &
  MPRA_minP.ratio$controls == 1,]$CTRL.median),
  lines = c(rep("Active",
  length(MPRA_minP.ratio[MPRA_minP.ratio$CTRL.padj <= 0.01,]$CTRL.median)),
  rep("Inactive", length(MPRA_minP.ratio[MPRA_minP.ratio$CTRL.padj > 0.01,]$CTRL.median)),
  rep("Control Mut", length(MPRA_minP.ratio[MPRA_minP.ratio$type =="Mut" &
  MPRA_minP.ratio$controls == 1,]$CTRL.median)),rep("Control WT",
  length(MPRA_minP.ratio[MPRA_minP.ratio$type =="Ref" & MPRA_minP.ratio$controls == 1,]$
         CTRL.median))))
ggplot(dat, aes(x = dens, fill = lines)) + geom_density(alpha = 0.5) +
  theme(axis.title.x = element_blank()) + theme(plot.background = element_blank(),
  panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
  panel.border = element_blank()) + theme_bw() + labs(x = "activity") +
  scale_fill_brewer(palette = "Set3")
```

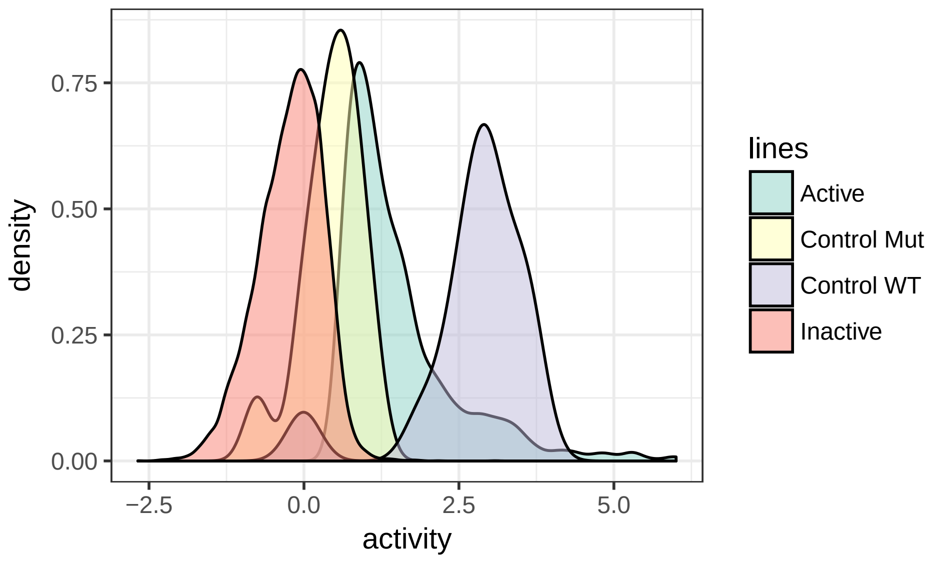


```
print(c('Percentage of ACs of MPRA library in K562 is:',length(
  MPRA_minP.ratio[MPRA_minP.ratio$CTRL.padj <= 0.01,]$CTRL.median)/dim(dat)[1]))
```

```
## [1] "Percentage of ACs of MPRA library in K562 is:"
## [2] "0.0353886373780527"
```

Density plots of active/inactive constructs and Mendlian controls:

```
dat <- data.frame( dens =
  c(MPRA_minP.ratio[MPRA_minP.ratio$GATA1.padj <= 0.01,]$GATA1.median,
  MPRA_minP.ratio[MPRA_minP.ratio$GATA1.padj > 0.01,]$GATA1.median,
  MPRA_minP.ratio[MPRA_minP.ratio$type == "Mut" &
  MPRA_minP.ratio$controls == 1,]$GATA1.median,
  MPRA_minP.ratio[MPRA_minP.ratio$type == "Ref" &
  MPRA_minP.ratio$controls == 1,]$GATA1.median),
  lines = c(rep("Active", length(
  MPRA_minP.ratio[MPRA_minP.ratio$GATA1.padj <= 0.01,]$GATA1.median)),
  rep("Inactive", length(MPRA_minP.ratio[MPRA_minP.ratio$GATA1.padj > 0.01,]$GATA1.median)),
  rep("Control Mut", length(MPRA_minP.ratio[MPRA_minP.ratio$type =="Mut" &
  MPRA_minP.ratio$controls == 1,]$GATA1.median)), rep("Control WT",
  length(MPRA_minP.ratio[MPRA_minP.ratio$type == "Ref" &
  MPRA_minP.ratio$controls == 1,]$GATA1.median))))
ggplot(dat, aes(x = dens, fill = lines)) + geom_density(alpha = 0.5) +
theme(axis.title.x = element_blank()) + theme(plot.background = element_blank(),
panel.grid.major = element_blank(),panel.grid.minor = element_blank(),
panel.border = element_blank()) + labs(x = "activity") + theme_bw() +
scale_fill_brewer(palette = "Set3")
```



```r
print(c('Percentage of ACs of MPRA library in K562+GATA1 is:',length(
      MPRA_minP.ratio[MPRA_minP.ratio$GATA1.padj <= 0.01,]$GATA1.median)/dim(dat)[1]))
```

```
## [1] "Percentage of ACs of MPRA library in K562+GATA1 is:"
## [2] "0.058662245743799"
```

Calculate fold change:

```r
REF <- MPRA_minP.ratio[MPRA_minP.ratio$type == "Ref", c("construct", "CTRL.median",
    "GATA1.median")]
MUT <- MPRA_minP.ratio[MPRA_minP.ratio$type == "Mut", c("construct", "CTRL.median",
    "GATA1.median")]
temp <- merge(MUT, REF, by = "construct")
temp$CTRL.fc <- temp$CTRL.median.x - temp$CTRL.median.y
temp$GATA1.fc <- temp$GATA1.median.x - temp$GATA1.median.y
temp <- temp[, c("construct", "CTRL.fc", "GATA1.fc")]
MPRA_minP.ratio <- merge(MPRA_minP.ratio, temp, by = "construct")
```

Identify MPRA Functional Variants

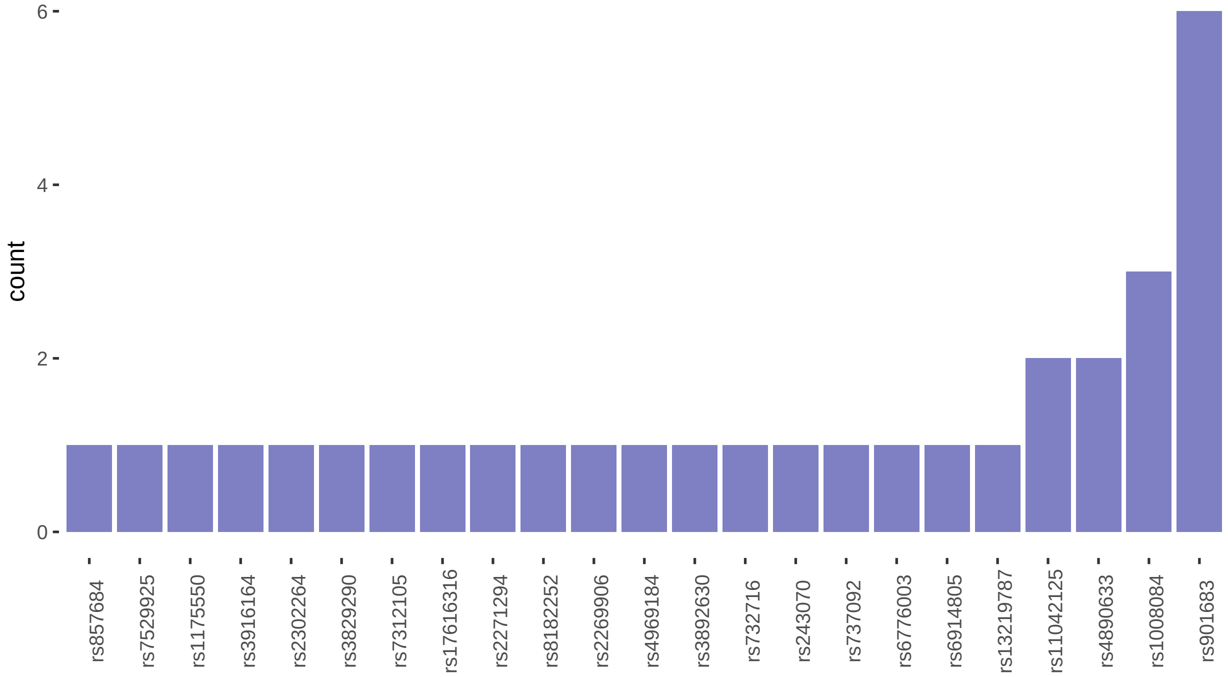
Two-sided Mann-Whitney-U test was used to identify constructs that show significant allelic variation in activity as mentioned above. Across 23 sentinel variants, 32 MPRA functional variants (MFVs) were identified. Out of all constructs, 43 of them containing 32 variants with differential allelic activity (FDR<1%).

Read in tag variants:

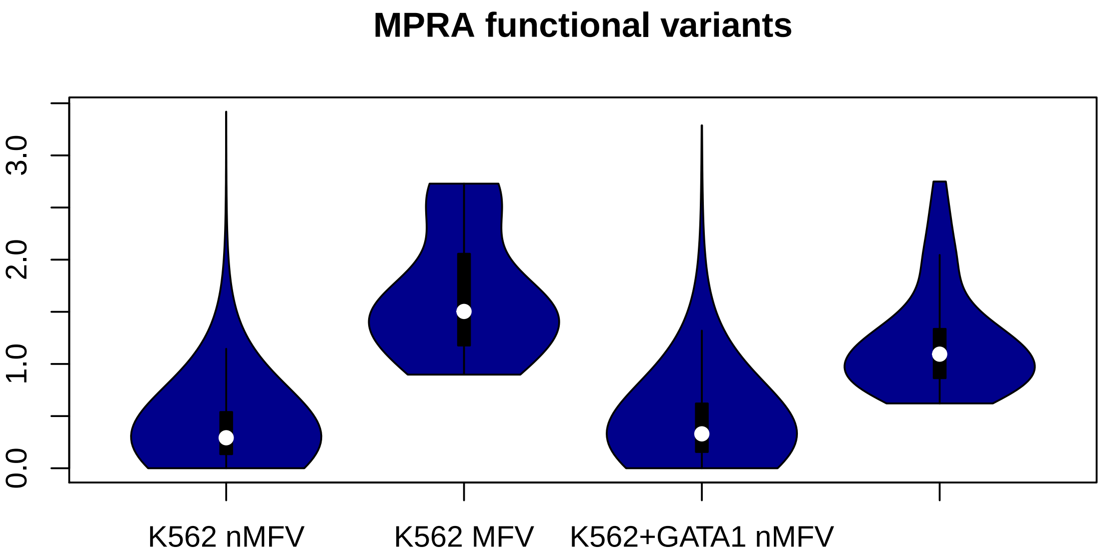
```r
tags <- read.table(paste0(dir, "Annotations/", "1000Genomes_Pilot3_LDpt8.tags"),
    sep = " ")
tags$tagSNP <- do.call(paste, as.data.frame(cbind(tags$V1, tags$V2)))
tags$oligo <- do.call(paste, as.data.frame(cbind(tags$V5, tags$V6)))
colnames(tags) <- c("tag_chr", "tag_pos", "tag_ref", "tag_alt", "chr", "pos",
    "ref", "alt", "tagSNP", "oligo")
```

Plot MFVs and their effect sizes. On average, the effect sizes of the MFVs were moderate:

```r
MPRA_minP.sig <-
MPRA_minP.ratio[MPRA_minP.ratio$CTRL.mut.padj<0.01|MPRA_minP.ratio$GATA1.mut.padj < 0.01,]
MPRA_minP.sig <- merge(MPRA_minP.sig, tags, by = "oligo")
MPRA_minP.hist <- data.frame(tapply(MPRA_minP.sig$oligo, MPRA_minP.sig$tagSNP, function(x) nlevels(facto
colnames(MPRA_minP.hist) <- c("count")
MPRA_minP.hist$oligo <- row.names(MPRA_minP.hist)
rsnumber <- read.csv(paste0(dir, "Annotations/", "rsSNP_annotation.csv"))
MPRA_minP.hist <- merge(MPRA_minP.hist, rsnumber, by = "oligo")
MPRA_minP.hist <- MPRA_minP.hist[order(MPRA_minP.hist$count), ]
row.names(MPRA_minP.hist) <- seq(1, dim(MPRA_minP.hist)[1], 1)
ggplot(data = MPRA_minP.hist, aes(x = factor(dbSNP, levels = dbSNP), y = count)) +
geom_bar(stat = "identity",alpha = .5, fill = "dark blue") +
theme_bw(base_size = 10, base_family = "Helvetica") +
theme(text = element_text(size =10), axis.text.x = element_text(angle = 90, vjust = 1)) +
theme(axis.title.x = element_blank()) +
theme(plot.background = element_blank(), panel.grid.major = element_blank(),
            panel.grid.minor = element_blank(), panel.border = element_blank())
```

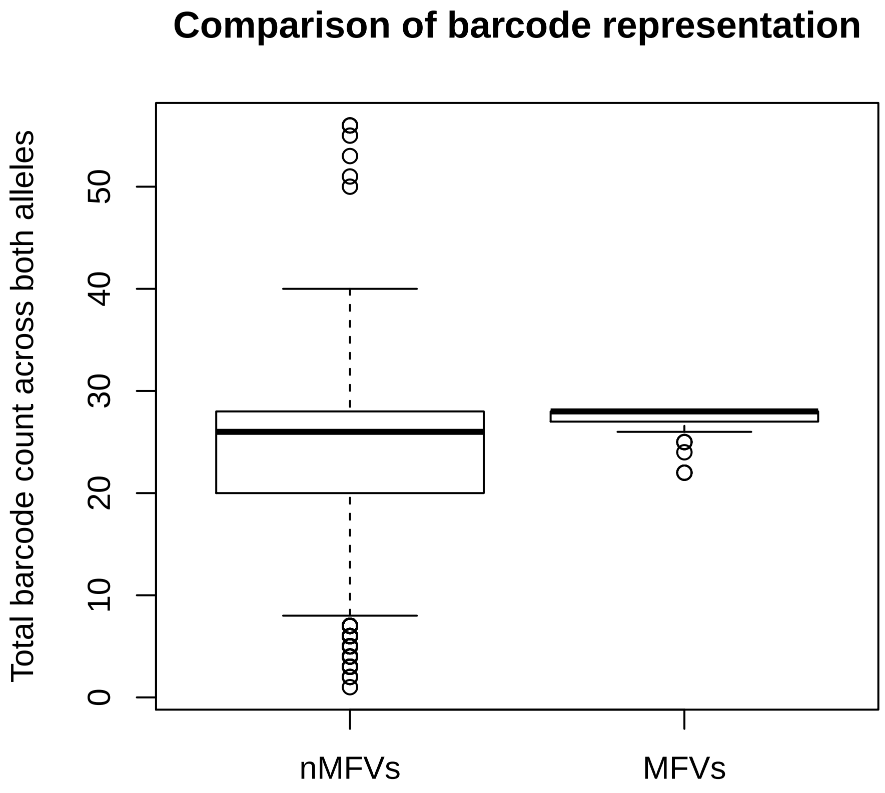


```
CTRL.construct <-unique(MPRA_minP.ratio[MPRA_minP.ratio$CTRL.mut.padj < 0.01 &
MPRA_minP.ratio$controls == 0, ]$construct)
GATA1.construct <- unique(MPRA_minP.ratio[MPRA_minP.ratio$GATA1.mut.padj < 0.01 &
MPRA_minP.ratio$controls == 0, ]$construct)

nMFV_ES_CTRL <- abs(as.vector(
  MPRA_minP.ratio[!(MPRA_minP.ratio$construct %in% CTRL.construct), ]$CTRL.fc))
MFV_ES_CTRL <- abs(as.vector(
  MPRA_minP.ratio[MPRA_minP.ratio$construct %in% CTRL.construct, ]$CTRL.fc))
nMFV_ES_GATA <- abs(as.vector(
  MPRA_minP.ratio[!(MPRA_minP.ratio$construct %in% GATA1.construct), ]$GATA1.fc))
MFV_ES_GATA <- abs(as.vector(
  MPRA_minP.ratio[MPRA_minP.ratio$construct %in% GATA1.construct, ]$GATA1.fc))

vioplot(nMFV_ES_CTRL,MFV_ES_CTRL,nMFV_ES_GATA,MFV_ES_GATA,names = c(
  "K562 nMFV", "K562 MFV", "K562+GATA1 nMFV", "K562+GATA1 MFV"), col = "dark blue")
title("MPRA functional variants")
```



Compare number of barcodes between MFV and nMFVs. We can see from the plot that variants identified as MFVs had better barcode representation than non-MFVs (nMFVs):

```
nMFV_barcodes <- as.vector(table(as.vector(MPRA_minP[!(MPRA_minP$construct %in%
c(MPRA_minP.sig[MPRA_minP.sig$controls == 0, ]$construct,
MPRA_minP.ratio[MPRA_minP.ratio$controls == 1, ]$construct)), ]$construct)))
MFV_barcodes <- as.vector(table(as.vector(MPRA_minP[MPRA_minP$construct %in%
MPRA_minP.sig[MPRA_minP.sig$controls == 0, ]$construct, ]$construct)))
boxplot(nMFV_barcodes, MFV_barcodes,
ylab = "Total barcode count across both alleles", names = c("nMFVs", "MFVs"),
main = "Comparison of barcode representation")
```



```
t.test(nMFV_barcodes, MFV_barcodes)
```

```
##
##  Welch Two Sample t-test
##
## data:  nMFV_barcodes and MFV_barcodes
## t = -14.487, df = 47.86, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.545258 -3.437269
## sample estimates:
## mean of x mean of y
##  23.12502  27.11628
```

Summary

This project is about a massively parallel reporter assay to simultaneously screen 2756 variants in strong linkage disequilibrium with 75 sentinel variants associated with red blood cell traits. This assay identified elements with endogenous eryehroid regulatory activity. Across 23 sentinel variants, 32 MPRA functional variants were identified. Active constructs were identified by a one-sided Mann-Whitney-U test. MPRA functional variants were identified by a two-sided Mann-Whitney-U test for K562 and K562+GATA1. After the analysis, the results are similar to the results in the paper although there are some slight difference owning to the modified data processing methods.

Reference

[1] Ulirsch, J. C., Nandakumar, S. K., Wang, L., Giani, F. C., Zhang, X., Rogov, P., . . . & Sankaran, V. G. (2016). Systematic functional dissection of common genetic variation affecting red blood cell traits. Cell, 165(6), 1530-1545.

[2] Welter, D., MacArthur, J., Morales, J., Burdett, T., Hall, P., Junkins, H., . . . & Parkinson, H. (2013). The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. Nucleic acids research, 42(D1), D1001-D1006.

[3] Raychaudhuri, S. (2011). Mapping rare and common causal alleles for complex human diseases. Cell, 147(1), 57-69.

[4] Campagna, D. R., Bie, C. I., Schmitz-Abe, K., Sweeney, M., Sendamarai, A. K., Schmidt, P. J., . . . & Niemeyer, C. M. (2014). X-linked sideroblastic anemia due to ALAS2 intron 1 enhancer element GATA-binding site mutations. American journal of hematology, 89(3), 315-319.

[5] Wakabayashi, A., Ulirsch, J. C., Ludwig, L. S., Fiorini, C., Yasuda, M., Choudhuri, A., . . . & Sankaran, V. G. (2016). Insight into GATA1 transcriptional activity through interrogation of cis elements disrupted in human erythroid disorders. Proceedings of the National Academy of Sciences, 113(16), 4434-4439.