

CUT&RUN MDM

Domien Vanneste

2025-09-05 22:01:37 +0200

Contents

Introduction	2
Load Packages	3
nf-core/cutandrun	3
SEACR	3
Homer	3
Create “Tag Directory” with makeTagDirectory	3
QC: Tag quantification and distribution	4
Peak selection (WT vs KO)	21
Annotate Peaks	24
MafB conservation	25
MafB target genes	30

Introduction

To assess whether MafB also regulates macrophage identity in humans, we performed CUT&RUN for MafB on human monocyte-derived mac (MDMs). CUT&RUN was performed with a CUTANA ChIC/CUT&RUN Kit (EpiCypher, 141048) according to manufacturer's instructions, with modifications. BMDMs from Mafbfl/fl or Lyz2CreMafbfl/fl mice were washed with ice-cold PBS and fixed with 0.1 % formaldehyde in PBS for 2 min at room temperature. Fixation was quenched by adding glycine (Merck, 104691000) to 0.125 M. For each CUT&RUN sample, 1 x 10⁶ fixed cells and 0.5 µg antibodies were added:

IgG (EpiCypher, 13-0042)

MafB1 (Sigma, HPA005653) MafB2 (Cell Signaling Technology, 41019) MafB3 (Proteintech, 20189-1-AP)

CUT&RUN libraries were prepared with a CUTANA™ CUT&RUN Library Prep Kit (EpiCypher, 141001) according to manufacturer's instructions. These libraries were sequenced on an NovaSeq 6000 (Illumina) sequencer on an S4 flow cell at 10 million reads per sample.

Load Packages

```
suppressMessages({  
  library(ggplot2)  
  library(colorRamp2)  
  library(ComplexHeatmap)  
  library(readxl)  
  library(futile.logger)  
  library(grid)  
  library(rtracklayer)  
  library(dplyr)  
  library(VennDiagram)  
})
```

```
## Warning: package 'colorRamp2' was built under R version 4.4.3
```

nf-core/cutandrun

The command used to launch the workflow was as follows:

```
nextflow run nf-core/cutandrun --input sample_list_all.csv --gtf genes.gtf --fasta genome.fa --peakcall
```

SEACR

```
#MafB1  
. /SEACR_1.3.sh MafB1_R1.sorted.bedgraph IgG_R1.sorted.bedgraph norm stringent MafB1_R1.seacr.peaks  
. /SEACR_1.3.sh MafB1_R2.sorted.bedgraph IgG_R1.sorted.bedgraph norm stringent MafB1_R2.seacr.peaks  
bedtools window -a MafB1_R1.seacr.peaks.stringent.bed -b MafB1_R2.seacr.peaks.stringent.bed -w 100 > Ma  
  
#MafB2  
. /SEACR_1.3.sh MafB2_R1.sorted.bedgraph IgG_R1.sorted.bedgraph norm stringent MafB2_R1.seacr.peaks  
. /SEACR_1.3.sh MafB2_R2.sorted.bedgraph IgG_R1.sorted.bedgraph norm stringent MafB2_R2.seacr.peaks  
bedtools window -a MafB2_R1.seacr.peaks.stringent.bed -b MafB2_R2.seacr.peaks.stringent.bed -w 100 > Ma  
  
#MafB3  
. /SEACR_1.3.sh MafB3_R1.sorted.bedgraph IgG_R1.sorted.bedgraph norm stringent MafB3_R1.seacr.peaks  
. /SEACR_1.3.sh MafB3_R2.sorted.bedgraph IgG_R1.sorted.bedgraph norm stringent MafB3_R2.seacr.peaks  
bedtools window -a MafB3_R1.seacr.peaks.stringent.bed -b MafB3_R2.seacr.peaks.stringent.bed -w 100 > Ma
```

Homer

Create “Tag Directory” with makeTagDirectory

```

#IgG
makeTagDirectory IgG_R1/ IgG_R1.target.markdup.sorted.bam

#MafB1
makeTagDirectory MafB1_R1/ MafB1_R1.target.markdup.sorted.bam
makeTagDirectory MafB1_R2/ MafB1_R2.target.markdup.sorted.bam

#MafB2
makeTagDirectory MafB2_R1/ MafB2_R1.target.markdup.sorted.bam
makeTagDirectory MafB2_R2/ MafB2_R2.target.markdup.sorted.bam

#MafB3
makeTagDirectory MafB3_R1/ MafB3_R1.target.markdup.sorted.bam
makeTagDirectory MafB3_R2/ MafB3_R2.target.markdup.sorted.bam

```

QC: Tag quantification and distribution

MafB1

```

MafB1_peaks_rep1 <- read.table("MafB1_R1.seacr.peaks.stringent.bed",
  header = FALSE, sep = "\t")

MafB1_peaks_rep2 <- read.table("MafB1_R2.seacr.peaks.stringent.bed",
  header = FALSE, sep = "\t")

MafB1_peaks_consensus <- read.table("MafB1.seacr.peaks.consensus.bed",
  header = FALSE, sep = "\t")

# peaks rep1
nrow(MafB1_peaks_rep1)

## [1] 25580

# peaks rep2
nrow(MafB1_peaks_rep2)

## [1] 28143

# shared peaks
nrow(MafB1_peaks_consensus)

## [1] 23271

score <- rowMeans(MafB1_peaks_consensus[, c("V4", "V10")])

MafB1_peaks_consensus$V4 <- score

MafB1_peaks_merged <- MafB1_peaks_consensus[, 1:5]

```

```
write.table(MafB1_peaks_merged, "MafB1_peaks_merged.bed", sep = "\t",
            quote = FALSE, row.names = FALSE, col.names = FALSE)
```

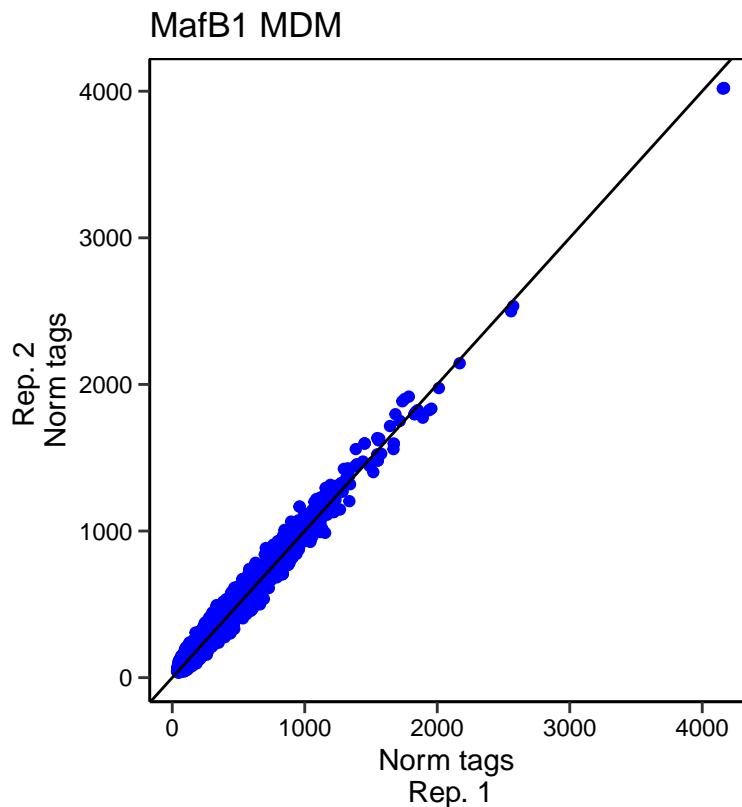
```
annotatePeaks.pl MafB1_peaks_merged.bed mm10 -d IgG_R1/ MafB1_R1/ MafB1_R2/ > MafB1_counts.txt
```

```
MafB1_counts <- read.table("MafB1_counts.txt", header = TRUE,
                           sep = "\t")

colnames(MafB1_counts)[colnames(MafB1_counts) == "PeakID..cmd.annotatePeaks.pl.MafB1_peaks_merged.bed.mm10"] <- "Norm tags"
colnames(MafB1_counts)[colnames(MafB1_counts) == "IgG_R1..Tag.Count.in.given.bp..4149983.0.Total..normed"] <- "Rep. 1"
colnames(MafB1_counts)[colnames(MafB1_counts) == "MafB1_R1..Tag.Count.in.given.bp..7504020.0.Total..normed"] <- "Rep. 2"
colnames(MafB1_counts)[colnames(MafB1_counts) == "MafB1_R2..Tag.Count.in.given.bp..7543614.0.Total..normed"] <- "Norm tags"

ggplot(data = MafB1_counts, aes(x = MafB1_R1, y = MafB1_R2)) +
  geom_point(show.legend = FALSE, colour = "blue") + geom_abline(slope = 1) +
  ggtitle("MafB1 MDM") + xlab("Norm tags \nRep. 1") + ylab("Rep. 2 \nNorm tags") +
  # xlim(0,850)+ ylim(0,850)+
```

theme_classic() + theme(axis.text.x = element_text(color = "black"),
 axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
 "cm"), panel.border = element_rect(fill = NA, color = "black",
 linetype = "solid"))

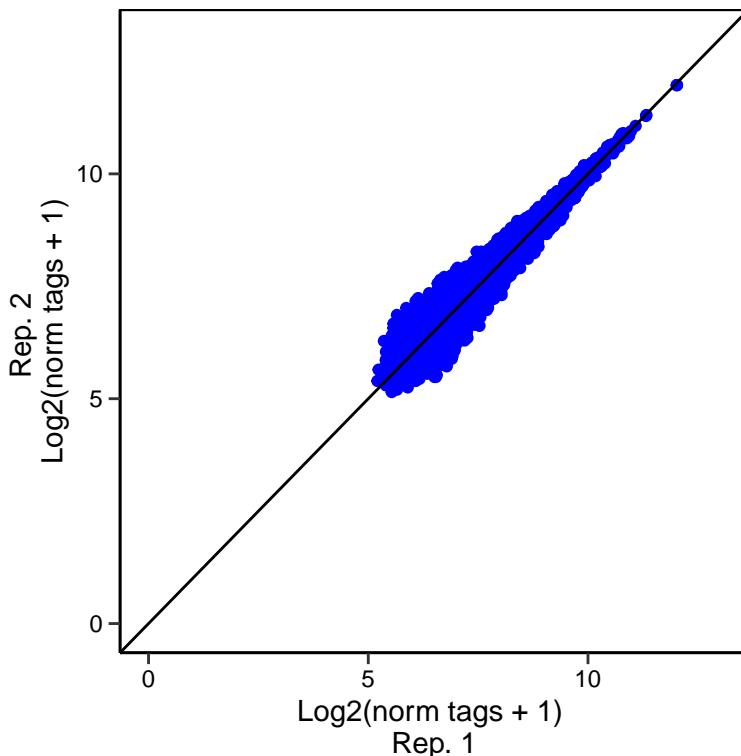


```

ggplot(data = MafB1_counts, aes(x = log2(MafB1_R1 + 1), y = log2(MafB1_R2 +
  1))) + geom_point(show.legend = FALSE, colour = "blue") +
  geom_abline(slope = 1) + ggtitle("MafB1 MDM") + xlab("Log2(norm tags + 1) \nRep. 1") +
  ylab("Rep. 2 \nLog2(norm tags + 1)") + xlim(0, 13) + ylim(0,
  13) + theme_classic() + theme(axis.text.x = element_text(color = "black"),
  axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
  "cm"), panel.border = element_rect(fill = NA, color = "black",
  linetype = "solid"))

```

MafB1 MDM



```
annotatePeaks.pl MafB1_peaks_merged.bed mm10 -size 4000 -hist 10 -d IgG_R1/ MafB1_R1/ MafB1_R2/ > MafB1
```

```
MafB1_hist <- read.table("MafB1_hist.txt", header = TRUE, sep = "\t")
```

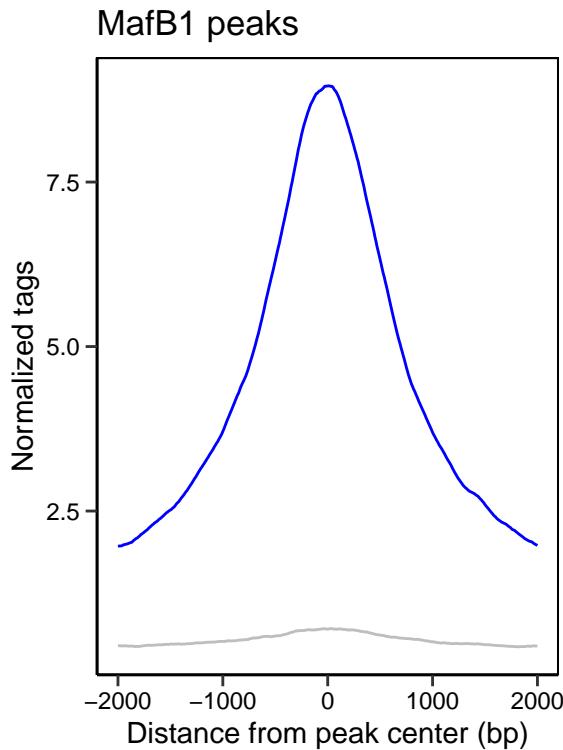
```
colnames(MafB1_hist)[colnames(MafB1_hist) == "Distance.from.Center..cmd.annotatePeaks.pl.MafB1_peaks_me"]
```

```
MafB1_hist$MafB1 <- rowMeans(MafB1_hist[, c("MafB1_R1..Coverage",
  "MafB1_R2..Coverage")])
```

```

ggplot(data = MafB1_hist, aes(x = Distance.from.Center, y = MafB1)) +
  geom_line(show.legend = FALSE, colour = "blue") + geom_line(show.legend = FALSE,
  colour = "grey", aes(x = Distance.from.Center, y = IgG_R1..Coverage)) +
  ggtitle("MafB1 peaks") + xlab("Distance from peak center (bp)") +
  ylab("Normalized tags") + theme_classic() + theme(axis.text.x = element_text(color = "black"),
  axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
  "cm"), panel.border = element_rect(fill = NA, color = "black",
  linetype = "solid"))

```



```
annotatePeaks.pl MafB1_peaks_merged.bed mm10 -size 4000 -hist 25 -ghist -d IgG_R1/ MafB1_R1/ MafB1_R2/ 

MafB1_heatmap <- read.table("MafB1_heatmap.txt", header = TRUE,
  check.names = FALSE, sep = "\t")

IgG_R1_heatmap <- MafB1_heatmap[, 2:162]

MafB1_R1_heatmap <- MafB1_heatmap[, 163:323]
MafB1_R2_heatmap <- MafB1_heatmap[, 324:484]
MafB1_heatmap <- (MafB1_R1_heatmap + MafB1_R2_heatmap)/2

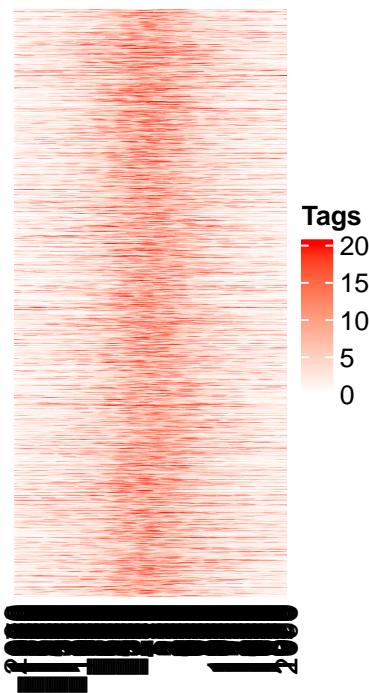
col_fun = colorRamp2(c(0, 20), c("white", "red"))

Heatmap(MafB1_heatmap, col_fun, cluster_columns = F, cluster_rows = F,
  heatmap_legend_param = list(title = "Tags"), use_raster = TRUE,
  raster_quality = 10, column_title = "MafB1")

## Warning: The input is a data frame-like object, convert it to a matrix.

## 'magick' package is suggested to install to give better rasterization.
##
## Set 'ht_opt$message = FALSE' to turn off this message.
```

MafB1



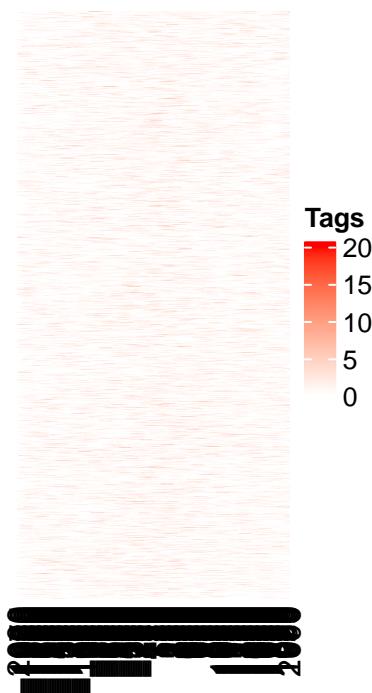
```
col_fun = colorRamp2(c(0, 20), c("white", "red"))

Heatmap(IgG_R1_heatmap, col_fun, cluster_columns = F, cluster_rows = F,
        heatmap_legend_param = list(title = "Tags"), use_raster = TRUE,
        raster_quality = 10, , column_title = "IgG")

## Warning: The input is a data frame-like object, convert it to a matrix.

## 'magick' package is suggested to install to give better rasterization.
##
## Set 'ht_opt$message = FALSE' to turn off this message.
```

IgG



MafB2

```
MafB2_peaks_rep1 <- read.table("MafB2_R1.seacr.peaks.stringent.bed",
  header = FALSE, sep = "\t")

MafB2_peaks_rep2 <- read.table("MafB2_R2.seacr.peaks.stringent.bed",
  header = FALSE, sep = "\t")

MafB2_peaks_consensus <- read.table("MafB2.seacr.peaks.consensus.bed",
  header = FALSE, sep = "\t")

# peaks rep1
nrow(MafB2_peaks_rep1)

## [1] 26327

# peaks rep2
nrow(MafB2_peaks_rep2)

## [1] 30718

# shared peaks
nrow(MafB2_peaks_consensus)

## [1] 24755
```

```

score <- rowMeans(MafB2_peaks_consensus[, c("V4", "V10")])

MafB2_peaks_consensus$V4 <- score

MafB2_peaks_merged <- MafB2_peaks_consensus[, 1:5]

write.table(MafB2_peaks_merged, "MafB2_peaks_merged.bed", sep = "\t",
            quote = FALSE, row.names = FALSE, col.names = FALSE)

annotatePeaks.pl MafB2_peaks_merged.bed mm10 -d IgG_R1/ MafB2_R1/ MafB2_R2/ > MafB2_counts.txt

MafB2_counts <- read.table("MafB2_counts.txt", header = TRUE,
                           sep = "\t")

colnames(MafB2_counts)[colnames(MafB2_counts) == "PeakID..cmd.annotatePeaks.pl.MafB2_peaks_merged.bed.mm10"] <- "PeakID"

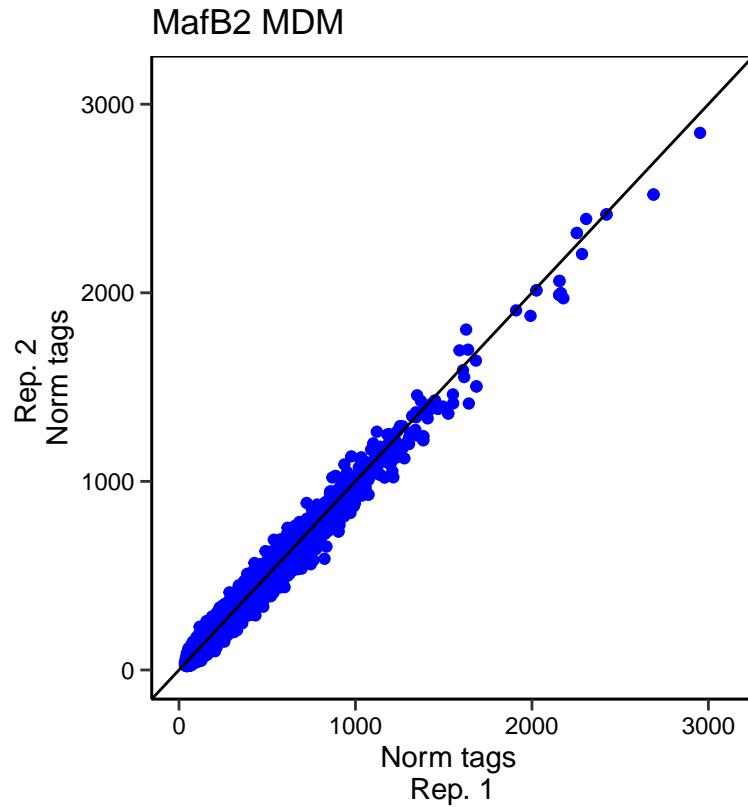
colnames(MafB2_counts)[colnames(MafB2_counts) == "IgG_R1..Tag.Count.in.given.bp..4149983.0.Total..normalization"] <- "IgG_R1..Tag.Count.in.given.bp..4149983.0.Total..norm"

colnames(MafB2_counts)[colnames(MafB2_counts) == "MafB2_R1..Tag.Count.in.given.bp..7202841.0.Total..normalization"] <- "MafB2_R1..Tag.Count.in.given.bp..7202841.0.Total..norm"

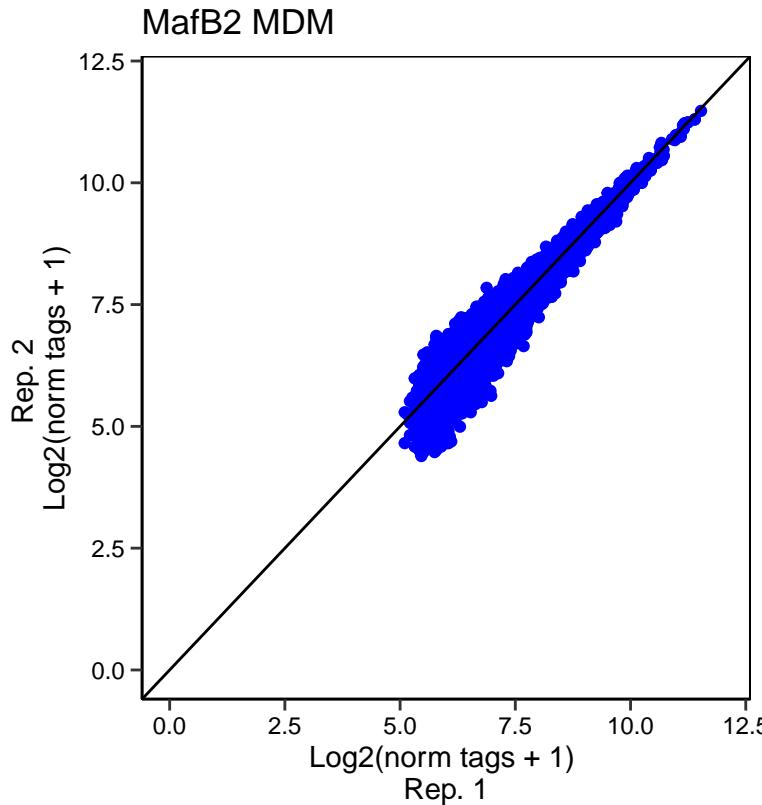
colnames(MafB2_counts)[colnames(MafB2_counts) == "MafB2_R2..Tag.Count.in.given.bp..8260258.0.Total..normalization"] <- "MafB2_R2..Tag.Count.in.given.bp..8260258.0.Total..norm"

ggplot(data = MafB2_counts, aes(x = MafB2_R1, y = MafB2_R2)) +
  geom_point(show.legend = FALSE, colour = "blue") + geom_abline(slope = 1) +
  ggtitle("MafB2 MDM") + xlab("Norm tags \nRep. 1") + ylab("Rep. 2 \nNorm tags") +
  xlim(0, 3100) + ylim(0, 3100) + theme_classic() + theme(axis.text.x = element_text(color = "black"),
  axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
  "cm"), panel.border = element_rect(fill = NA, color = "black",
  linetype = "solid"))

```



```
ggplot(data = MafB2_counts, aes(x = log2(MafB2_R1 + 1), y = log2(MafB2_R2 + 1))) + geom_point(show.legend = FALSE, colour = "blue") +
  geom_abline(slope = 1) + ggtitle("MafB2 MDM") + xlab("Log2(norm tags + 1) \nRep. 1") +
  ylab("Rep. 2 \nLog2(norm tags + 1)") + xlim(0, 12) + ylim(0, 12) +
  theme_classic() + theme(axis.text.x = element_text(color = "black"),
  axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
  "cm"), panel.border = element_rect(fill = NA, color = "black",
  linetype = "solid"))
```



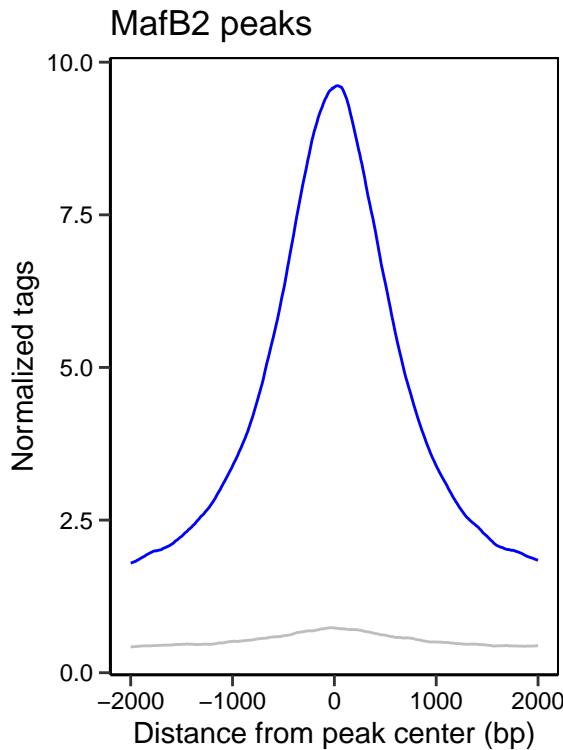
```
annotatePeaks.pl MafB2_peaks_merged.bed mm10 -size 4000 -hist 10 -d IgG_R1/ MafB2_R1/ MafB2_R2/ > MafB2_hist.txt

MafB2_hist <- read.table("MafB2_hist.txt", header = TRUE, sep = "\t")

colnames(MafB2_hist)[colnames(MafB2_hist) == "Distance.from.Center..cmd.annotatePeaks.pl.MafB2_peaks_merged"] <- c("MafB2")

MafB2_hist$MafB2 <- rowMeans(MafB2_hist[, c("MafB2_R1..Coverage",
                                             "MafB2_R2..Coverage")])

ggplot(data = MafB2_hist, aes(x = Distance.from.Center, y = MafB2)) +
  geom_line(show.legend = FALSE, colour = "blue") + geom_line(show.legend = FALSE,
  colour = "grey", aes(x = Distance.from.Center, y = IgG_R1..Coverage)) +
  ggtitle("MafB2 peaks") + xlab("Distance from peak center (bp)") +
  ylab("Normalized tags") + theme_classic() + theme(axis.text.x = element_text(color = "black"),
  axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
  "cm"), panel.border = element_rect(fill = NA, color = "black",
  linetype = "solid"))
```



```
annotatePeaks.pl MafB2_peaks_merged.bed mm10 -size 4000 -hist 25 -ghist -d IgG_R1/ MafB2_R1/ MafB2_R2/ 

MafB2_heatmap <- read.table("MafB2_heatmap.txt", header = TRUE,
  check.names = FALSE, sep = "\t")

IgG_R1_heatmap <- MafB2_heatmap[, 2:162]

MafB2_R1_heatmap <- MafB2_heatmap[, 163:323]
MafB2_R2_heatmap <- MafB2_heatmap[, 324:484]
MafB2_heatmap <- (MafB2_R1_heatmap + MafB2_R2_heatmap)/2

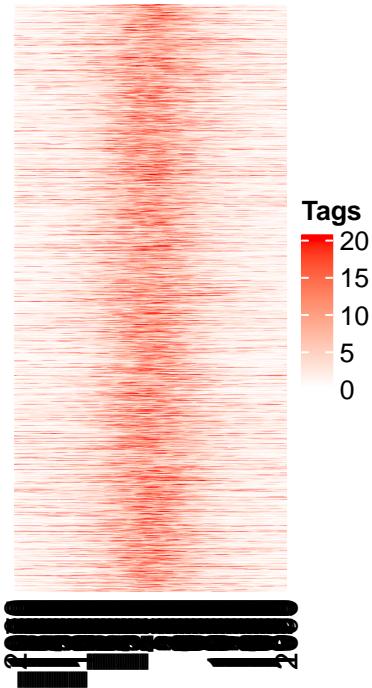
col_fun = colorRamp2(c(0, 20), c("white", "red"))

Heatmap(MafB2_heatmap, col_fun, cluster_columns = F, cluster_rows = F,
  heatmap_legend_param = list(title = "Tags"), use_raster = TRUE,
  raster_quality = 10, column_title = "MafB2")

## Warning: The input is a data frame-like object, convert it to a matrix.

## 'magick' package is suggested to install to give better rasterization.
##
## Set 'ht_opt$message = FALSE' to turn off this message.
```

MafB2



```
col_fun = colorRamp2(c(0, 20), c("white", "red"))

Heatmap(IgG_R1_heatmap, col_fun, cluster_columns = F, cluster_rows = F,
        heatmap_legend_param = list(title = "Tags"), use_raster = TRUE,
        raster_quality = 10, , column_title = "IgG")

## Warning: The input is a data frame-like object, convert it to a matrix.

## 'magick' package is suggested to install to give better rasterization.
##
## Set 'ht_opt$message = FALSE' to turn off this message.
```

IgG



MafB3

```
MafB3_peaks_rep1 <- read.table("MafB3_R1.seacr.peaks.stringent.bed",
  header = FALSE, sep = "\t")

MafB3_peaks_rep2 <- read.table("MafB3_R2.seacr.peaks.stringent.bed",
  header = FALSE, sep = "\t")

MafB3_peaks_consensus <- read.table("MafB3.seacr.peaks.consensus.bed",
  header = FALSE, sep = "\t")

# peaks rep1
nrow(MafB3_peaks_rep1)

## [1] 28539

# peaks rep2
nrow(MafB3_peaks_rep2)

## [1] 24807

# shared peaks
nrow(MafB3_peaks_consensus)

## [1] 22908
```

```

score <- rowMeans(MafB3_peaks_consensus[, c("V4", "V10")])

MafB3_peaks_consensus$V4 <- score

MafB3_peaks_merged <- MafB3_peaks_consensus[, 1:5]

write.table(MafB3_peaks_merged, "MafB3_peaks_merged.bed", sep = "\t",
            quote = FALSE, row.names = FALSE, col.names = FALSE)

annotatePeaks.pl MafB3_peaks_merged.bed mm10 -d IgG_R1/ MafB3_R1/ MafB3_R2/ > MafB3_counts.txt

MafB3_counts <- read.table("MafB3_counts.txt", header = TRUE,
                           sep = "\t")

colnames(MafB3_counts)[colnames(MafB3_counts) == "PeakID..cmd.annotatePeaks.pl.MafB3_peaks_merged.bed.mm10"] <- "PeakID"

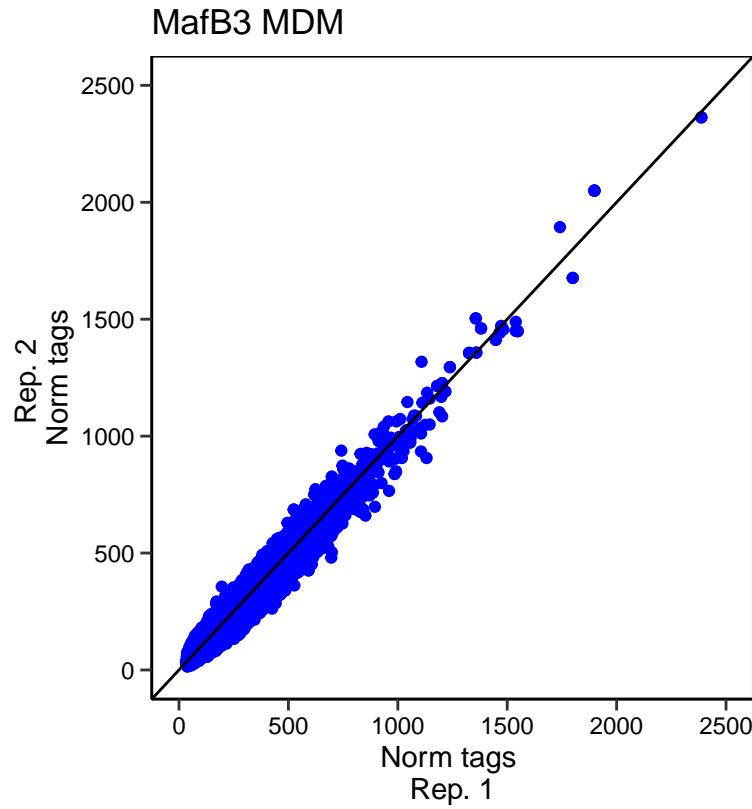
colnames(MafB3_counts)[colnames(MafB3_counts) == "IgG_R1..Tag.Count.in.given.bp..4149983.0.Total..normalized"] <- "Norm tags \nRep. 1"

colnames(MafB3_counts)[colnames(MafB3_counts) == "MafB3_R1..Tag.Count.in.given.bp..7874750.0.Total..normalized"] <- "Rep. 2 \nNorm tags"

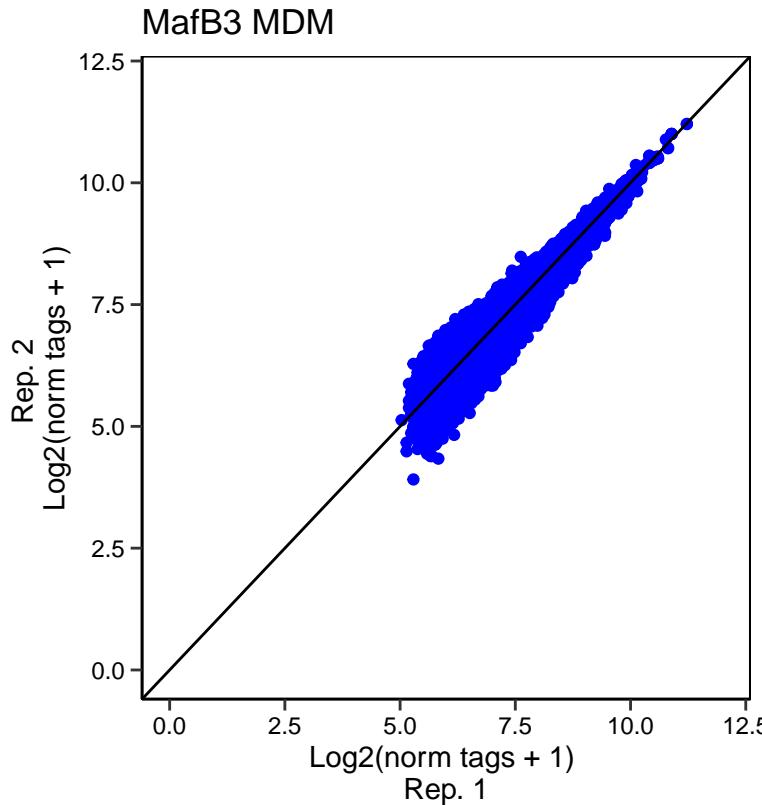
colnames(MafB3_counts)[colnames(MafB3_counts) == "MafB3_R2..Tag.Count.in.given.bp..6764209.0.Total..normalized"] <- "Rep. 3 \nNorm tags"

ggplot(data = MafB3_counts, aes(x = MafB3_R1, y = MafB3_R2)) +
  geom_point(show.legend = FALSE, colour = "blue") + geom_abline(slope = 1) +
  ggtitle("MafB3 MDM") + xlab("Norm tags \nRep. 1") + ylab("Rep. 2 \nNorm tags") +
  xlim(0, 2500) + ylim(0, 2500) + theme_classic() + theme(axis.text.x = element_text(color = "black"),
  axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
  "cm"), panel.border = element_rect(fill = NA, color = "black",
  linetype = "solid"))

```



```
ggplot(data = MafB3_counts, aes(x = log2(MafB3_R1 + 1), y = log2(MafB3_R2 + 1))) + geom_point(show.legend = FALSE, colour = "blue") +
  geom_abline(slope = 1) + ggtitle("MafB3 MDM") + xlab("Log2(norm tags + 1) \nRep. 1") +
  ylab("Rep. 2 \nLog2(norm tags + 1)") + xlim(0, 12) + ylim(0, 12) +
  theme_classic() + theme(axis.text.x = element_text(color = "black"),
  axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
  "cm"), panel.border = element_rect(fill = NA, color = "black",
  linetype = "solid"))
```



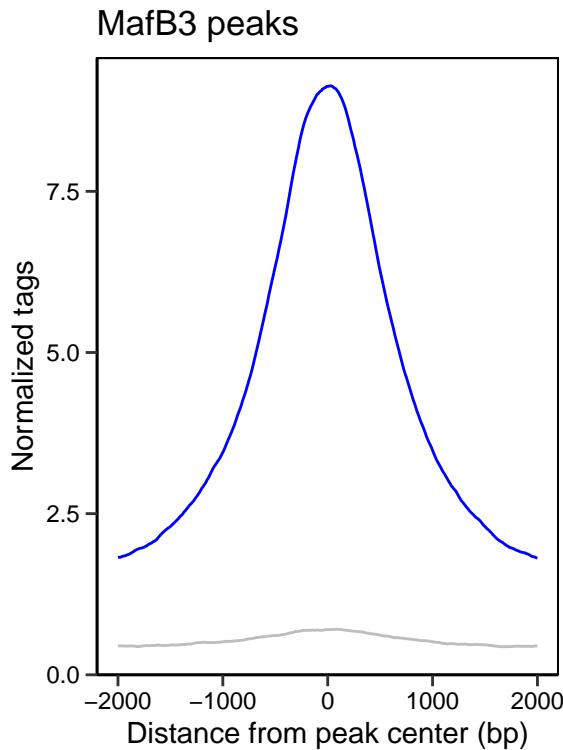
```
annotatePeaks.pl MafB3_peaks_merged.bed mm10 -size 4000 -hist 10 -d IgG_R1/ MafB3_R1/ MafB3_R2/ > MafB3_hist.txt

MafB3_hist <- read.table("MafB3_hist.txt", header = TRUE, sep = "\t")

colnames(MafB3_hist)[colnames(MafB3_hist) == "Distance.from.Center..cmd.annotatePeaks.pl.MafB3_peaks_merged"] <- c("MafB3")

MafB3_hist$MafB3 <- rowMeans(MafB3_hist[, c("MafB3_R1..Coverage", "MafB3_R2..Coverage")])

ggplot(data = MafB3_hist, aes(x = Distance.from.Center, y = MafB3)) +
  geom_line(show.legend = FALSE, colour = "blue") + geom_line(show.legend = FALSE, colour = "grey", aes(x = Distance.from.Center, y = IgG_R1..Coverage)) +
  ggtitle("MafB3 peaks") + xlab("Distance from peak center (bp)") +
  ylab("Normalized tags") + theme_classic() + theme(axis.text.x = element_text(color = "black"),
  axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
  "cm"), panel.border = element_rect(fill = NA, color = "black",
  linetype = "solid"))
```



```
annotatePeaks.pl MafB3_peaks_merged.bed mm10 -size 4000 -hist 25 -ghist -d IgG_R1/ MafB3_R1/ MafB3_R2/ 

MafB3_heatmap <- read.table("MafB3_heatmap.txt", header = TRUE,
  check.names = FALSE, sep = "\t")

IgG_R1_heatmap <- MafB3_heatmap[, 2:162]

MafB3_R1_heatmap <- MafB3_heatmap[, 163:323]
MafB3_R2_heatmap <- MafB3_heatmap[, 324:484]
MafB3_heatmap <- (MafB3_R1_heatmap + MafB3_R2_heatmap)/2

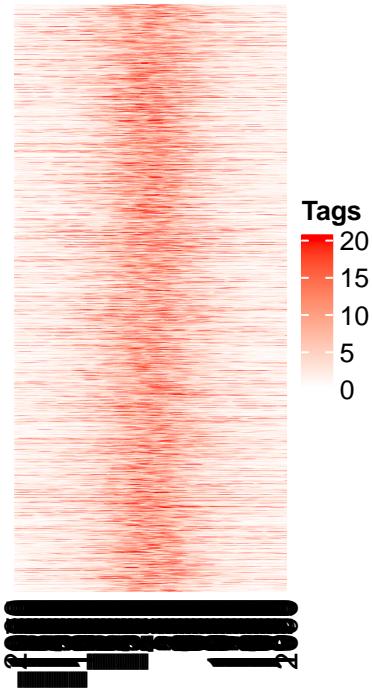
col_fun = colorRamp2(c(0, 20), c("white", "red"))

Heatmap(MafB3_heatmap, col_fun, cluster_columns = F, cluster_rows = F,
  heatmap_legend_param = list(title = "Tags"), use_raster = TRUE,
  raster_quality = 10, column_title = "MafB3")

## Warning: The input is a data frame-like object, convert it to a matrix.

## 'magick' package is suggested to install to give better rasterization.
##
## Set 'ht_opt$message = FALSE' to turn off this message.
```

MafB3



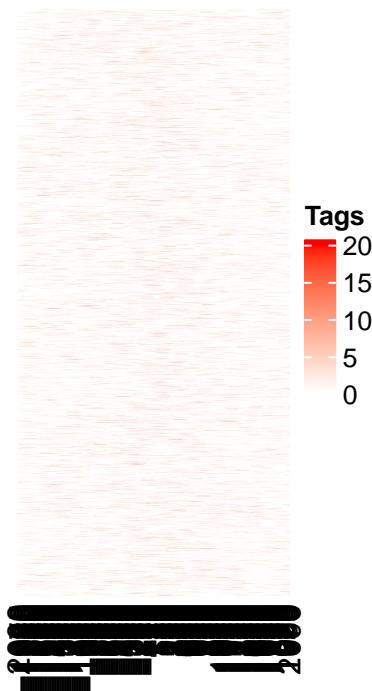
```
col_fun = colorRamp2(c(0, 20), c("white", "red"))

Heatmap(IgG_R1_heatmap, col_fun, cluster_columns = F, cluster_rows = F,
        heatmap_legend_param = list(title = "Tags"), use_raster = TRUE,
        raster_quality = 10, , column_title = "IgG")

## Warning: The input is a data frame-like object, convert it to a matrix.

## 'magick' package is suggested to install to give better rasterization.
##
## Set 'ht_opt$message = FALSE' to turn off this message.
```

IgG

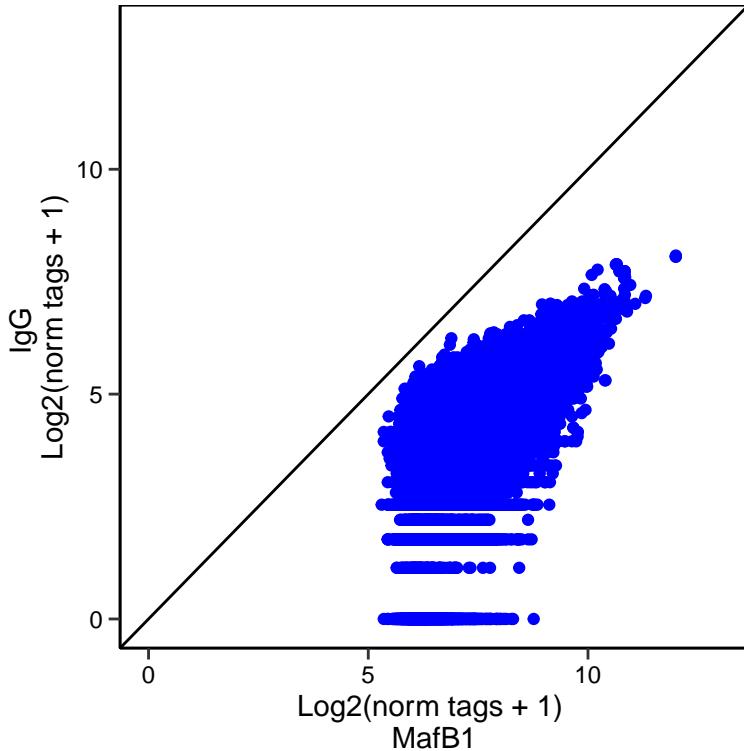


Peak selection (WT vs KO)

MafB1

```
MafB1_counts$MafB1 <- rowMeans(MafB1_counts[, c("MafB1_R1", "MafB1_R2")])  
  
MafB1_counts$log2_IgG <- log2(MafB1_counts$IgG_R1 + 1)  
MafB1_counts$log2_MafB1 <- log2(MafB1_counts$MafB1 + 1)  
MafB1_counts$below_diag <- MafB1_counts$log2_IgG < MafB1_counts$log2_MafB1  
  
ggplot(data = MafB1_counts, aes(x = log2_MafB1, y = log2_IgG)) +  
  geom_point(aes(color = below_diag), show.legend = FALSE) +  
  scale_color_manual(values = c(`FALSE` = "grey", `TRUE` = "blue")) +  
  geom_abline(slope = 1) + ggtitle("MafB1 Peaks") + xlab("Log2(norm tags + 1) \nMafB1") +  
  ylab("IgG \nLog2(norm tags + 1)") + xlim(0, 13) + ylim(0,  
  13) + theme_classic() + theme(axis.text.x = element_text(color = "black"),  
  axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,  
  "cm"), panel.border = element_rect(fill = NA, color = "black",  
  linetype = "solid"))
```

MafB1 Peaks



```

MafB1_PeakID <- c(MafB1_counts[MafB1_counts[, "below_diag"] ==
  "TRUE", ]$PeakID)
MafB1_peaks <- MafB1_peaks_merged[MafB1_peaks_merged[, 4] %in%
  MafB1_PeakID, ]

write.table(MafB1_peaks, "MafB1_peaks.bed", sep = "\t", quote = FALSE,
  row.names = FALSE, col.names = FALSE)
  
```

MafB2

```

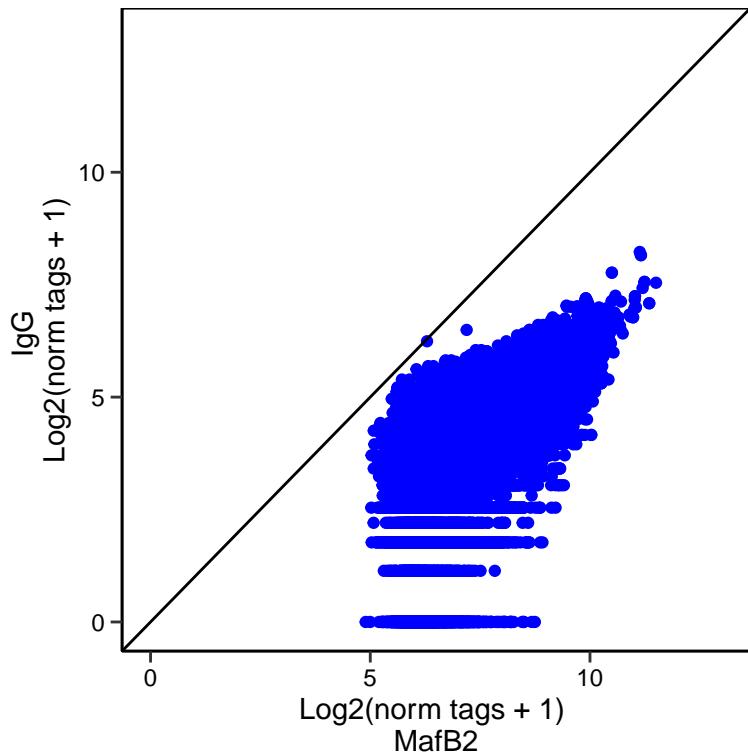
MafB2_counts$MafB2 <- rowMeans(MafB2_counts[, c("MafB2_R1", "MafB2_R2")])

MafB2_counts$log2_IgG <- log2(MafB2_counts$IgG_R1 + 1)
MafB2_counts$log2_MafB2 <- log2(MafB2_counts$MafB2 + 1)
MafB2_counts$below_diag <- MafB2_counts$log2_IgG < MafB2_counts$log2_MafB2
  
```

```

ggplot(data = MafB2_counts, aes(x = log2_MafB2, y = log2_IgG)) +
  geom_point(aes(color = below_diag), show.legend = FALSE) +
  scale_color_manual(values = c(`FALSE` = "grey", `TRUE` = "blue")) +
  geom_abline(slope = 1) + ggtitle("MafB2 Peaks") + xlab("Log2(norm tags + 1) \nMafB2") +
  ylab("IgG \nLog2(norm tags + 1)") + xlim(0, 13) + ylim(0,
  13) + theme_classic() + theme(axis.text.x = element_text(color = "black"),
  axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
  "cm"), panel.border = element_rect(fill = NA, color = "black",
  linetype = "solid"))
  
```

MafB2 Peaks



```

MafB2_PeakID <- c(MafB2_counts[MafB2_counts[, "below_diag"] ==
  "TRUE", ]$PeakID)
MafB2_peaks <- MafB2_peaks_merged[MafB2_peaks_merged[, 4] %in%
  MafB2_PeakID, ]

write.table(MafB2_peaks, "MafB2_peaks.bed", sep = "\t", quote = FALSE,
  row.names = FALSE, col.names = FALSE)
  
```

MafB3

```

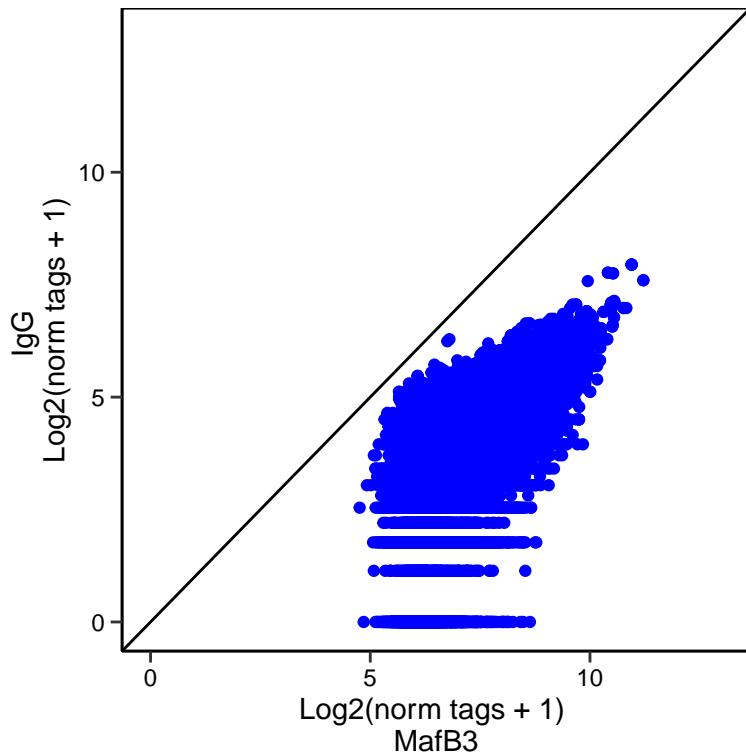
MafB3_counts$MafB3 <- rowMeans(MafB3_counts[, c("MafB3_R1", "MafB3_R2")])

MafB3_counts$log2_IgG <- log2(MafB3_counts$IgG_R1 + 1)
MafB3_counts$log2_MafB3 <- log2(MafB3_counts$MafB3 + 1)
MafB3_counts$below_diag <- MafB3_counts$log2_IgG < MafB3_counts$log2_MafB3
  
```

```

ggplot(data = MafB3_counts, aes(x = log2_MafB3, y = log2_IgG)) +
  geom_point(aes(color = below_diag), show.legend = FALSE) +
  scale_color_manual(values = c(`FALSE` = "grey", `TRUE` = "blue")) +
  geom_abline(slope = 1) + ggtitle("MafB3 Peaks") + xlab("Log2(norm tags + 1) \nMafB3") +
  ylab("IgG \nLog2(norm tags + 1)") + xlim(0, 13) + ylim(0,
  13) + theme_classic() + theme(axis.text.x = element_text(color = "black"),
  axis.text.y = element_text(color = "black"), axis.ticks.length = unit(0.15,
  "cm"), panel.border = element_rect(fill = NA, color = "black",
  linetype = "solid"))
  
```

MafB3 Peaks



```

MafB3_PeakID <- c(MafB3_counts[MafB3_counts[, "below_diag"] ==
  "TRUE", ]$PeakID)
MafB3_peaks <- MafB3_peaks_merged[MafB3_peaks_merged[, 4] %in%
  MafB3_PeakID, ]

write.table(MafB3_peaks, "MafB3_peaks.bed", sep = "\t", quote = FALSE,
  row.names = FALSE, col.names = FALSE)
  
```

Annotate Peaks

MafB1

```
annotatePeaks.pl MafB1_peaks.bed genome.fa -gtf genes.gtf > MafB1_peaks_annot.txt
```

```
MafB1_peaks_annot <- read.table("MafB1_peaks_annot.txt", header = TRUE,
  sep = "\t")
```

```
MafB1_peaks_annot$Annotation.not.detailed <- sub(" .*", "", MafB1_peaks_annot$Annotation)

table(MafB1_peaks_annot$Annotation.not.detailed)
```

```
##
##      exon    Intergenic      intron promoter-TSS        TTS
##      1149       4542       10177       6337       1066
```

MafB2

```
annotatePeaks.pl MafB2_peaks.bed genome.fa -gtf genes.gtf > MafB2_peaks_annot.txt

MafB2_peaks_annot <- read.table("MafB2_peaks_annot.txt", header = TRUE,
  sep = "\t")

MafB2_peaks_annot$Annotation.not.detailed <- sub(" .*", "", MafB2_peaks_annot$Annotation)

table(MafB2_peaks_annot$Annotation.not.detailed)

##  
## exon Intergenic intron promoter-TSS TTS  
## 1076 5173 11663 5606 1237
```

MafB3

```
annotatePeaks.pl MafB3_peaks.bed genome.fa -gtf genes.gtf > MafB3_peaks_annot.txt

MafB3_peaks_annot <- read.table("MafB3_peaks_annot.txt", header = TRUE,
  sep = "\t")

MafB3_peaks_annot$Annotation.not.detailed <- sub(" .*", "", MafB3_peaks_annot$Annotation)

table(MafB3_peaks_annot$Annotation.not.detailed)

##  
## exon Intergenic intron promoter-TSS TTS  
## 1224 4377 9540 6711 1056
```

MafB conservation

MafB1

```
MafB1_BMDM_peaks <- read.table("MafB1_BMDM_peaks.bed", header = FALSE,
  sep = "\t")

MafB1_BMDM_peaks$V1 <- paste0("chr", MafB1_BMDM_peaks$V1)

MafB1_BMDM_peaks <- MafB1_BMDM_peaks[, 1:4]

write.table(MafB1_BMDM_peaks, "MafB1_BMDM_peaks_for_liftover.bed",
  sep = "\t", quote = FALSE, row.names = FALSE, col.names = FALSE)
```

Convert mouse MafB1 BMDM peaks to human

```
MafB1_BMDM_peaks_human <- read.table("MafB1_BMDM_peaks_human.bed",
  header = FALSE, sep = "\t")

MafB1_BMDM_peaks_human$V1 <- sub("^chr", "", MafB1_BMDM_peaks_human$V1)

write.table(MafB1_BMDM_peaks_human, "MafB1_BMDM_peaks_human.bed",
  sep = "\t", quote = FALSE, row.names = FALSE, col.names = FALSE)
```

check overlap

```
bedtools window -a MafB1_peaks.bed -b MafB1_BMDM_peaks_human.bed -w 100 > MafB1_peaks_conserved.bed
```

```
MafB1_peaks_conserved <- read.table("MafB1_peaks_conserved.bed",
  header = FALSE, sep = "\t")

nrow(MafB1_peaks)
```

```
## [1] 113
```

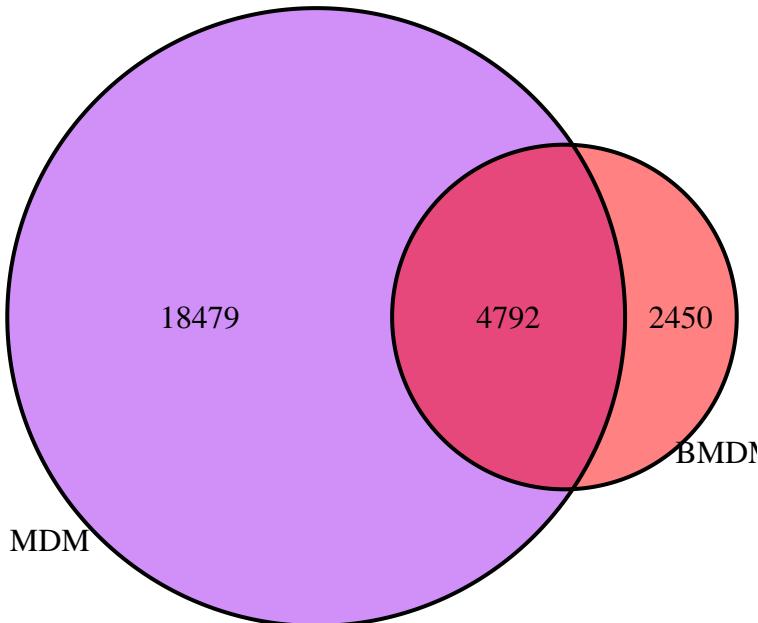
```
nrow(MafB1_BMDM_peaks_human)
```

```
## [1] 7242
```

```
nrow(MafB1_peaks_conserved)
```

```
## [1] 22
```

```
draw.pairwise.venn(area1 = 7242, area2 = 23271, cross.area = 4792,
  category = c("BMDM", "MDM"), fill = c("red", "purple"))
```



```
## (polygon[GRID.polygon.1853], polygon[GRID.polygon.1854], polygon[GRID.polygon.1855], polygon[GRID.po
grid.newpage()
```

MafB2

```
MafB2_BMDM_peaks <- read.table("MafB2_BMDM_peaks.bed", header = FALSE,
sep = "\t")

MafB2_BMDM_peaks$V1 <- paste0("chr", MafB2_BMDM_peaks$V1)

MafB2_BMDM_peaks <- MafB2_BMDM_peaks[, 1:4]

write.table(MafB2_BMDM_peaks, "MafB2_BMDM_peaks_for_liftover.bed",
sep = "\t", quote = FALSE, row.names = FALSE, col.names = FALSE)
```

Convert mouse MafB2 BMDM peaks to human

```
MafB2_BMDM_peaks_human <- read.table("MafB2_BMDM_peaks_human.bed",
header = FALSE, sep = "\t")

MafB2_BMDM_peaks_human$V1 <- sub("^chr", "", MafB2_BMDM_peaks_human$V1)

write.table(MafB2_BMDM_peaks_human, "MafB2_BMDM_peaks_human.bed",
sep = "\t", quote = FALSE, row.names = FALSE, col.names = FALSE)
```

check overlap

```
bedtools window -a MafB2_peaks.bed -b MafB2_BMDM_peaks_human.bed -w 100 > MafB2_peaks_conserved.bed

MafB2_peaks_conserved <- read.table("MafB2_peaks_conserved.bed",
  header = FALSE, sep = "\t")

nrow(MafB2_peaks)

## [1] 24755

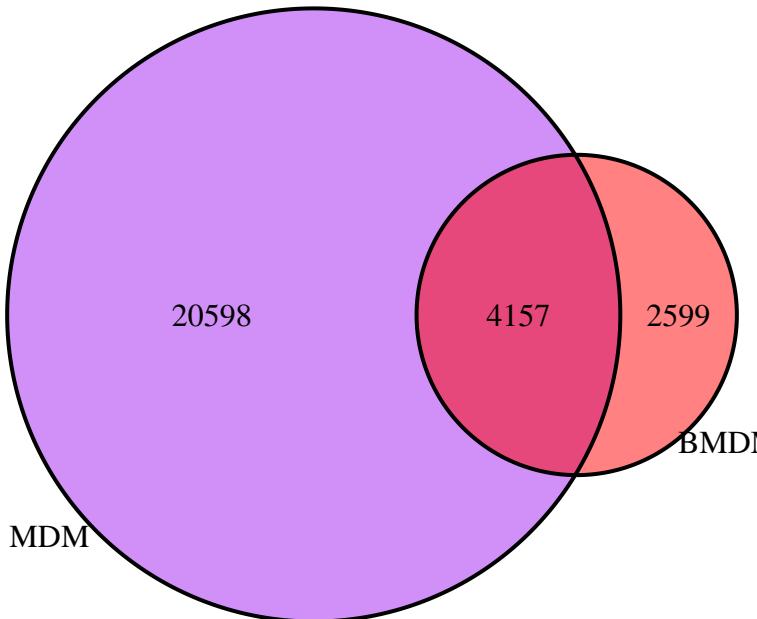
nrow(MafB2_BMDM_peaks_human)

## [1] 6756

nrow(MafB2_peaks_conserved)

## [1] 4157

draw.pairwise.venn(area1 = 6756, area2 = 24755, cross.area = 4157,
  category = c("BMDM", "MDM"), fill = c("red", "purple"))
```



```
## (polygon[GRID.polygon.1862], polygon[GRID.polygon.1863], polygon[GRID.polygon.1864], polygon[GRID.po
```

```
grid.newpage()
```

MafB3

```
MafB3_BMDM_peaks <- read.table("MafB3_BMDM_peaks.bed", header = FALSE,
  sep = "\t")

MafB3_BMDM_peaks$V1 <- paste0("chr", MafB3_BMDM_peaks$V1)

MafB3_BMDM_peaks <- MafB3_BMDM_peaks[, 1:4]

write.table(MafB3_BMDM_peaks, "MafB3_BMDM_peaks_for_liftover.bed",
  sep = "\t", quote = FALSE, row.names = FALSE, col.names = FALSE)
```

Convert mouse MafB3 BMDM peaks to human

```
MafB3_BMDM_peaks_human <- read.table("MafB3_BMDM_peaks_human.bed",
  header = FALSE, sep = "\t")

MafB3_BMDM_peaks_human$V1 <- sub("^chr", "", MafB3_BMDM_peaks_human$V1)

write.table(MafB3_BMDM_peaks_human, "MafB3_BMDM_peaks_human.bed",
  sep = "\t", quote = FALSE, row.names = FALSE, col.names = FALSE)
```

check overlap

```
bedtools window -a MafB3_peaks.bed -b MafB3_BMDM_peaks_human.bed -w 100 > MafB3_peaks_conserved.bed

MafB3_peaks_conserved <- read.table("MafB3_peaks_conserved.bed",
  header = FALSE, sep = "\t")

nrow(MafB3_peaks)

## [1] 22908

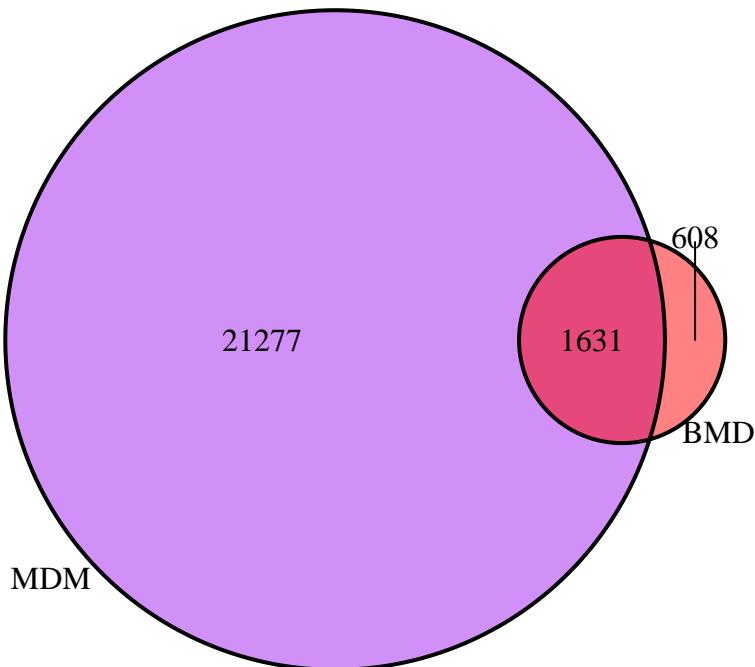
nrow(MafB3_BMDM_peaks_human)

## [1] 2239

nrow(MafB3_peaks_conserved)

## [1] 1631

draw.pairwise.venn(area1 = 2239, area2 = 22908, cross.area = 1631,
  category = c("BMDM", "MDM"), fill = c("red", "purple"))
```



```
## (polygon[GRID.polygon.1871], polygon[GRID.polygon.1872], polygon[GRID.polygon.1873], polygon[GRID.po
grid.newpage()
```

MafB target genes

MafB1

```
MafB1_genes <- MafB1_peaks_annot[MafB1_peaks_annot[, "Distance.to.TSS"] >=
-2000 & MafB1_peaks_annot[, "Distance.to.TSS"] <= 2000, ]

MafB1_genes <- unique(MafB1_genes$Gene.Name)
MafB1_genes <- na.omit(MafB1_genes)

MafB1_BMDM_peaks_annot <- read.table("MafB1_BMDM_peaks_annot.txt",
header = TRUE, sep = "\t")

MafB1_BMDM_genes <- MafB1_BMDM_peaks_annot[MafB1_BMDM_peaks_annot[, "Distance.to.TSS"] >= -2000 & MafB1_BMDM_peaks_annot[, "Distance.to.TSS"] <= 2000, ]

MafB1_BMDM_genes <- unique(MafB1_BMDM_genes$Gene.Name)
MafB1_BMDM_genes <- na.omit(MafB1_BMDM_genes)

write.table(MafB1_BMDM_genes, "MafB1_BMDM_genes.txt", sep = "\t",
quote = FALSE, row.names = FALSE, col.names = FALSE)
```

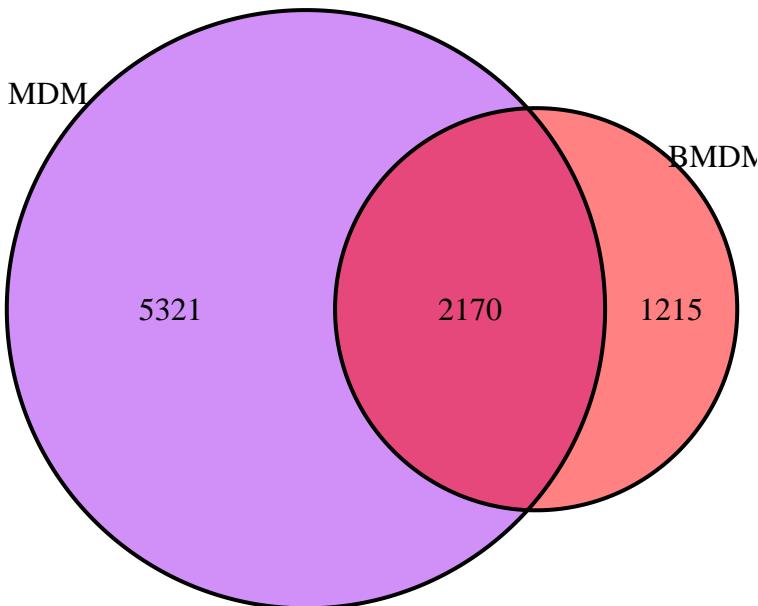
```

MafB1_BMDM_genes_human <- read.csv("MafB1_BMDM_genes_human.csv")
MafB1_BMDM_genes_human <- na.omit(unique(MafB1_BMDM_genes_human$ortholog_name))

# Helper function to display Venn diagram
display_venn <- function(x, ...) {
  grid.newpage()
  venn_object <- venn.diagram(x, filename = NULL, ...)
  grid.draw(venn_object)
}

x <- list(MDM = MafB1_genes, BMDM = MafB1_BMDM_genes_human)
# pdf(file =
# 'C:/Users/domie/Documents/CUTandRUN/Venn_MafB1_Mac.pdf',
# width = 4, height = 4)
display_venn(x, fill = c("purple", "red"))

```



```

# dev.off()

g <- readGFF("C:/Users/domie/Documents/CUTandRUN_MDM/genes.gtf")
pc <- g %>%
  dplyr::filter(type == "gene")
rm(g)
dim(pc)
# genes 63140

```

```

dat <- matrix(c(1609, 5882, 1016, 54633), nrow = 2, dimnames = list(Mac_sign = c("YES",
    "NO"), MafB1_target = c("YES", "NO")))
dat

##           MafB1_target
## Mac_sign   YES     NO
##       YES 1609 1016
##       NO  5882 54633

fisher.test(dat)

##
## Fisher's Exact Test for Count Data
##
## data: dat
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 13.52885 16.00381
## sample estimates:
## odds ratio
##      14.7081

```

MafB2

```

MafB2_genes <- MafB2_peaks_annot[MafB2_peaks_annot[, "Distance.to.TSS"] >=
    -2000 & MafB2_peaks_annot[, "Distance.to.TSS"] <= 2000, ]

MafB2_genes <- unique(MafB2_genes$Gene.Name)
MafB2_genes <- na.omit(MafB2_genes)

MafB2_BMDM_peaks_annot <- read.table("MafB2_BMDM_peaks_annot.txt",
    header = TRUE, sep = "\t")

MafB2_BMDM_genes <- MafB2_BMDM_peaks_annot[MafB2_BMDM_peaks_annot[, "Distance.to.TSS"] >= -2000 & MafB2_BMDM_peaks_annot[, "Distance.to.TSS"] <= 2000, ]

MafB2_BMDM_genes <- unique(MafB2_BMDM_genes$Gene.Name)
MafB2_BMDM_genes <- na.omit(MafB2_BMDM_genes)

write.table(MafB2_BMDM_genes, "MafB2_BMDM_genes.txt", sep = "\t",
    quote = FALSE, row.names = FALSE, col.names = FALSE)

MafB2_BMDM_genes_human <- read.csv("MafB2_BMDM_genes_human.csv")

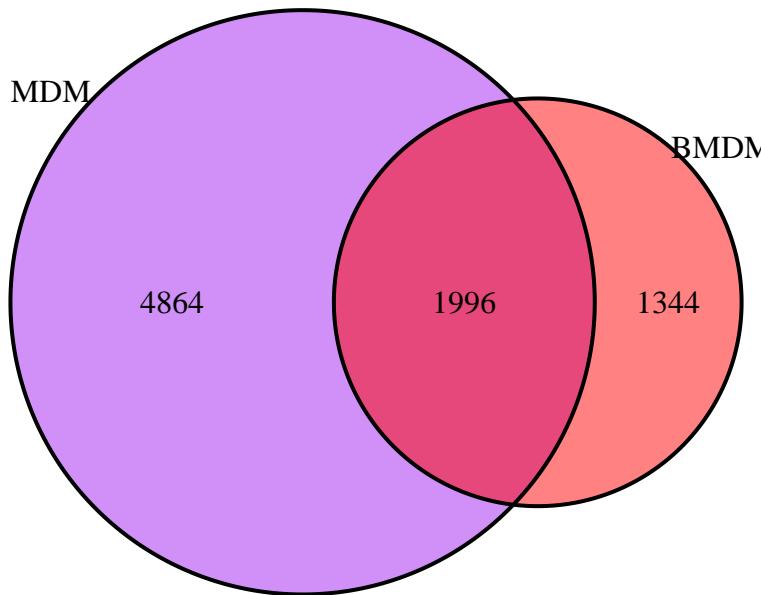
MafB2_BMDM_genes_human <- na.omit(unique(MafB2_BMDM_genes_human$ortholog_name))

```

```

x <- list(MDM = MafB2_genes, BMDM = MafB2_BMDM_genes_human)
# pdf(file =
#   'C:/Users/domie/Documents/CUTandRUN/Venn_MafB2_Mac.pdf',
#   width = 4, height = 4)
display_venn(x, fill = c("purple", "red"))

```



```

# dev.off()

Conserved_MafB_genes <- intersect(MafB2_genes, MafB2_BMDM_genes_human)

write.table(Conserved_MafB_genes, "Conserved_MafB_genes.txt",
            sep = "\t", quote = FALSE, row.names = FALSE, col.names = FALSE)

dat <- matrix(c(1206, 5654, 863, 55417), nrow = 2, dimnames = list(Mac_sign = c("YES",
"NO"), MafB2_target = c("YES", "NO")))
dat

##          MafB2_target
## Mac_sign  YES     NO
##      YES 1206    863
##      NO   5654 55417

fisher.test(dat)

##
## Fisher's Exact Test for Count Data

```

```

## 
## data: dat
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 12.49011 15.02716
## sample estimates:
## odds ratio
## 13.69581

```

MafB3

```

MafB3_genes <- MafB3_peaks_annot[MafB3_peaks_annot[, "Distance.to.TSS"] >=
-2000 & MafB3_peaks_annot[, "Distance.to.TSS"] <= 2000, ]

MafB3_genes <- unique(MafB3_genes$Gene.Name)
MafB3_genes <- na.omit(MafB3_genes)

MafB3_BMDM_peaks_annot <- read.table("MafB3_BMDM_peaks_annot.txt",
header = TRUE, sep = "\t")

MafB3_BMDM_genes <- MafB3_BMDM_peaks_annot[MafB3_BMDM_peaks_annot[, "Distance.to.TSS"] >= -2000 & MafB3_BMDM_peaks_annot[, "Distance.to.TSS"] <= 2000, ]

MafB3_BMDM_genes <- unique(MafB3_BMDM_genes$Gene.Name)
MafB3_BMDM_genes <- na.omit(MafB3_BMDM_genes)

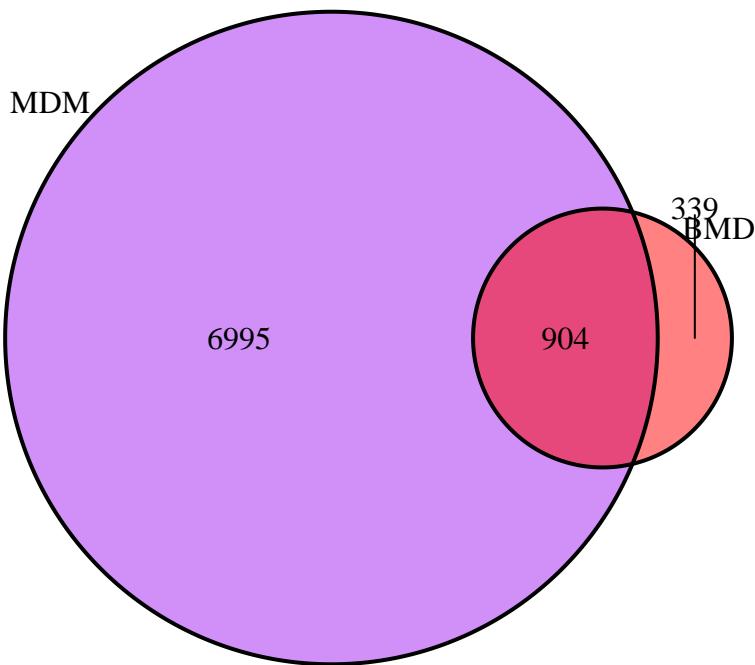
write.table(MafB3_BMDM_genes, "MafB3_BMDM_genes.txt", sep = "\t",
quote = FALSE, row.names = FALSE, col.names = FALSE)

MafB3_BMDM_genes_human <- read.csv("MafB3_BMDM_genes_human.csv")

MafB3_BMDM_genes_human <- na.omit(unique(MafB3_BMDM_genes_human$ortholog_name))

x <- list(MDM = MafB3_genes, BMDM = MafB3_BMDM_genes_human)
# pdf(file =
# 'C:/Users/domie/Documents/CUTandRUN/Venn_MafB3_Mac.pdf',
# width = 4, height = 4)
display_venn(x, fill = c("purple", "red"))

```



```
# dev.off()

dat <- matrix(c(765, 7134, 347, 54894), nrow = 2, dimnames = list(Mac_sign = c("YES",
  "NO"), MafB3_target = c("YES", "NO")))
dat

##           MafB3_target
## Mac_sign   YES      NO
##       YES  765    347
##       NO   7134 54894

fisher.test(dat)

##
## Fisher's Exact Test for Count Data
##
## data: dat
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
## 14.88569 19.35857
## sample estimates:
## odds ratio
## 16.96239
```