

CUT&RUN LPM WT vs KO

Domien Vanneste

2025-09-05 21:10:34 +0200

Contents

Introduction	2
Load Packages	3
nf-core/cutandrun	3
Homer	3
Create “Tag Directory” with makeTagDirectory	3
QC: Tag quantification and distribution	4
Peak selection (WT vs KO)	27
Annotate Peaks	31
Overlap with BMDM peaks	32
MafB target genes	35

Introduction

To test whether MafB also regulates mac identity in primary RTM, we performed CUT&RUN for MafB on LPMs from Mafbfl/fl or Lyz2CreMafbfl/fl mice. 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
```

Homer

Create “Tag Directory” with makeTagDirectory

```
#IgG  
makeTagDirectory IgG_WT_R1/ IgG_WT_R1.target.markdup.sorted.bam  
makeTagDirectory IgG_KO_R1/ IgG_KO_R1.target.markdup.sorted.bam  
  
#MafB1  
makeTagDirectory MafB1_WT_R1/ MafB1_WT_R1.target.markdup.sorted.bam  
makeTagDirectory MafB1_WT_R2/ MafB1_WT_R2.target.markdup.sorted.bam  
makeTagDirectory MafB1_KO_R1/ MafB1_KO_R1.target.markdup.sorted.bam  
makeTagDirectory MafB1_KO_R2/ MafB1_KO_R2.target.markdup.sorted.bam  
  
#MafB2  
makeTagDirectory MafB2_WT_R1/ MafB2_WT_R1.target.markdup.sorted.bam  
makeTagDirectory MafB2_WT_R2/ MafB2_WT_R2.target.markdup.sorted.bam  
makeTagDirectory MafB2_KO_R1/ MafB2_KO_R1.target.markdup.sorted.bam  
makeTagDirectory MafB2_KO_R2/ MafB2_KO_R2.target.markdup.sorted.bam  
  
#MafB3  
makeTagDirectory MafB3_WT_R1/ MafB3_WT_R1.target.markdup.sorted.bam  
makeTagDirectory MafB3_WT_R2/ MafB3_WT_R2.target.markdup.sorted.bam  
makeTagDirectory MafB3_KO_R1/ MafB3_KO_R1.target.markdup.sorted.bam  
makeTagDirectory MafB3_KO_R2/ MafB3_KO_R2.target.markdup.sorted.bam
```

QC: Tag quantification and distribution

MafB1

```
MafB1_peaks_rep1 <- read.table("MafB1_WT_R1.seacr.peaks.stringent.bed",
  header = FALSE, sep = "\t")

MafB1_peaks_rep2 <- read.table("MafB1_WT_R2.seacr.peaks.stringent.bed",
  header = FALSE, sep = "\t")

MafB1_peaks_consensus <- read.table("MafB1_WT.seacr.consensus.peak_counts.bed",
  header = FALSE, sep = "\t")

# peaks rep1
nrow(MafB1_peaks_rep1)

## [1] 14412

# peaks rep2
nrow(MafB1_peaks_rep2)

## [1] 14273

# consensus peaks
nrow(MafB1_peaks_consensus)

## [1] 19528

# shared peaks
length(which(MafB1_peaks_consensus[, 10] == 2))

## [1] 8509

MafB1_peaks_merged <- MafB1_peaks_consensus[MafB1_peaks_consensus[, 10] == 2, ]

split_scores <- strsplit(as.character(MafB1_peaks_merged$V6),
  ", ")

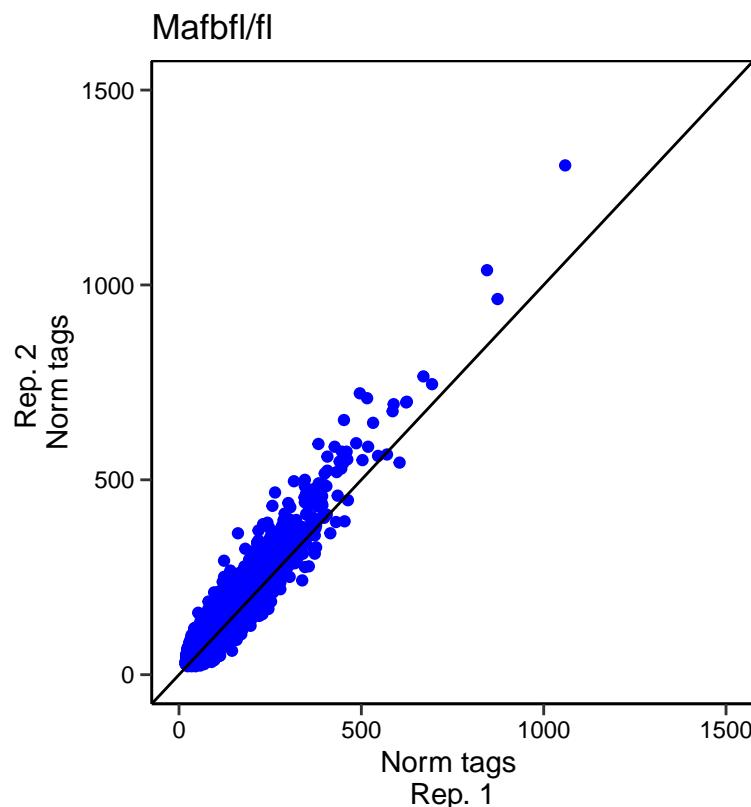
means <- sapply(split_scores, function(x) {
  if (all(is.na(x)))
    return(NA_real_)
  nums <- as.numeric(x)
  mean(nums, na.rm = TRUE)
})

MafB1_peaks_merged$V4 <- means

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 MafB1_WT_R1/ MafB1_WT_R2/ MafB1_KO_R1/ MafB1_KO_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"] <- "PeakID"  
colnames(MafB1_counts)[colnames(MafB1_counts) == "MafB1_WT_R1..Tag.Count.in.given.bp..5505710.0.Total..0"] <- "WT_R1_Total"  
colnames(MafB1_counts)[colnames(MafB1_counts) == "MafB1_WT_R2..Tag.Count.in.given.bp..5540315.0.Total..0"] <- "WT_R2_Total"  
colnames(MafB1_counts)[colnames(MafB1_counts) == "MafB1_KO_R1..Tag.Count.in.given.bp..6630990.0.Total..0"] <- "KO_R1_Total"  
colnames(MafB1_counts)[colnames(MafB1_counts) == "MafB1_KO_R2..Tag.Count.in.given.bp..5953272.0.Total..0"] <- "KO_R2_Total"  
  
ggplot(data = MafB1_counts, aes(x = MafB1_WT_R1, y = MafB1_WT_R2)) +  
  geom_point(show.legend = FALSE, colour = "blue") + geom_abline(slope = 1) +  
  ggtitle("Mafbfl/fl") + xlab("Norm tags \nRep. 1") + ylab("Rep. 2 \nNorm tags") +  
  xlim(0, 1500) + ylim(0, 1500) + 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"))
```

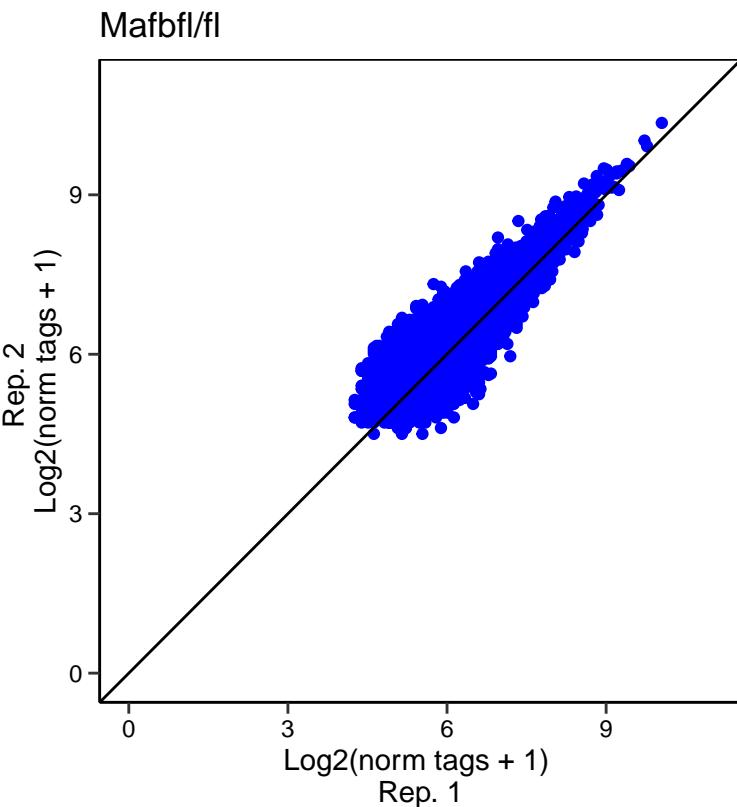


```
ggplot(data = MafB1_counts, aes(x = log2(MafB1_WT_R1 + 1), y = log2(MafB1_WT_R2 +  
1))) + geom_point(show.legend = FALSE, colour = "blue") +
```

```

geom_abline(slope = 1) + ggtitle("Mafbfl/f1") + xlab("Log2(norm tags + 1) \nRep. 1") +
ylab("Rep. 2 \nLog2(norm tags + 1)") + xlim(0, 11) + ylim(0,
11) + 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"))

```

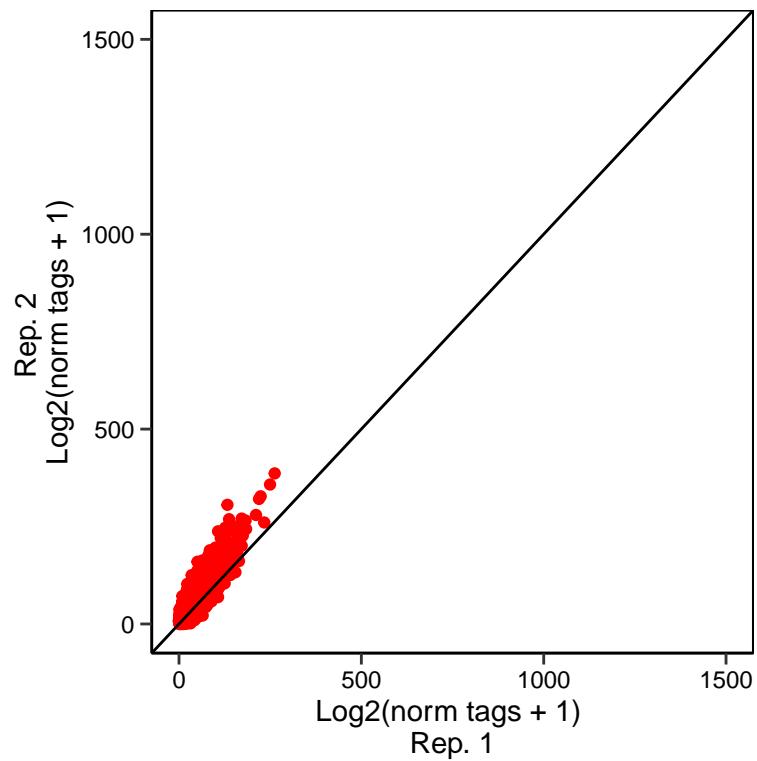


```

ggplot(data = MafB1_counts, aes(x = MafB1_KO_R1, y = MafB1_KO_R2)) +
  geom_point(show.legend = FALSE, colour = "red") + geom_abline(slope = 1) +
  ggtitle("Lyz2CreMafbfl/f1") + xlab("Log2(norm tags + 1) \nRep. 1") +
  ylab("Rep. 2 \nLog2(norm tags + 1)") + xlim(0, 1500) + ylim(0,
  1500) + 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"))

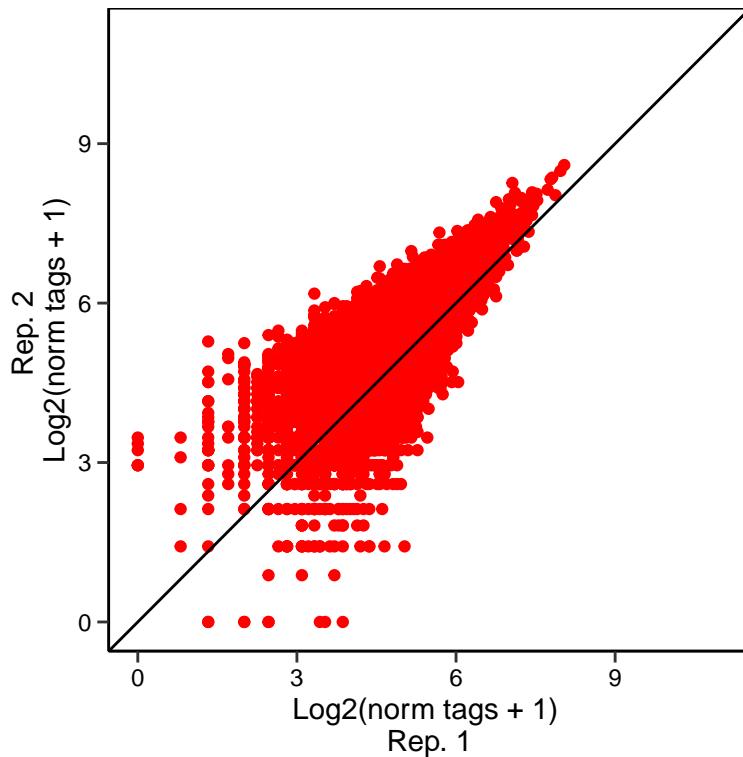
```

Lyz2CreMafbfl/fl



```
ggplot(data = MafB1_counts, aes(x = log2(MafB1_KO_R1 + 1), y = log2(MafB1_KO_R2 + 1))) + geom_point(show.legend = FALSE, colour = "red") +  
  geom_abline(slope = 1) + ggtitle("Lyz2CreMafbfl/fl") + xlab("Log2(norm tags + 1) \nRep. 1") +  
  ylab("Rep. 2 \nLog2(norm tags + 1)") + xlim(0, 11) + ylim(0,  
  11) + 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"))
```

Lyz2CreMafbfl/fl



```
annotatePeaks.pl MafB1_peaks_merged.bed mm10 -size 4000 -hist 10 -d MafB1_WT_R1/ MafB1_WT_R2/ MafB1_KO_R1/ MafB1_KO_R2/
```

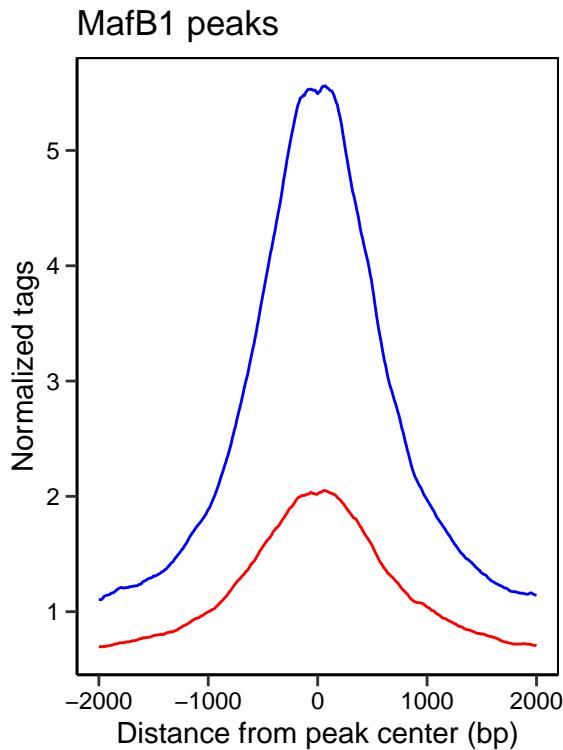
```
MafB1_hist <- read.table("MafB1_hist.txt", header = TRUE, sep = "\t")
```

```
colnames(MafB1_hist)[colnames(MafB1_hist) == "Distance.from.Center..cmd.annotatePeaks.pl.MafB1_peaks_merged"] <- c("Distance.from.Center")
```

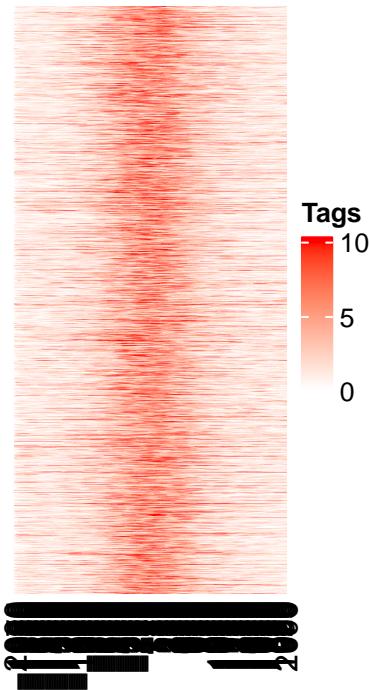
```
MafB1_hist$MafB1_WT <- rowMeans(MafB1_hist[, c("MafB1_WT_R1..Coverage", "MafB1_WT_R2..Coverage")])
```

```
MafB1_hist$MafB1_KO <- rowMeans(MafB1_hist[, c("MafB1_KO_R1..Coverage", "MafB1_KO_R2..Coverage")])
```

```
ggplot(data = MafB1_hist, aes(x = Distance.from.Center, y = MafB1_WT)) +
  geom_line(show.legend = FALSE, colour = "blue") + geom_line(show.legend = FALSE,
  colour = "red", aes(x = Distance.from.Center, y = MafB1_KO)) +
  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"))
```



Mafbfl/fl



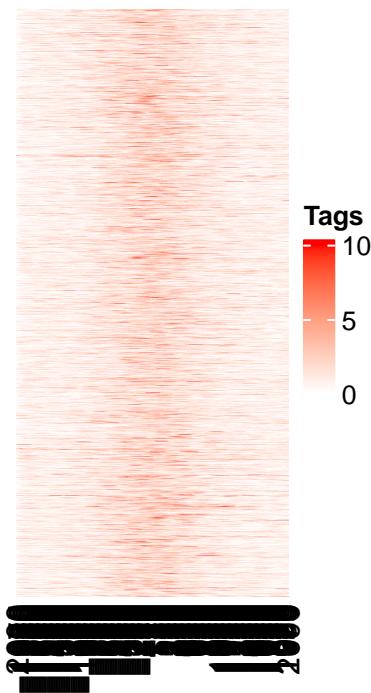
```
col_fun = colorRamp2(c(0, 10), c("white", "red"))

Heatmap(MafB1_K0_heatmap, col_fun, cluster_columns = F, cluster_rows = F,
        heatmap_legend_param = list(title = "Tags"), use_raster = TRUE,
        raster_quality = 10, , column_title = "LyzCreMafbfl/fl")
```

```
## 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.
```

LyzCreMafbfl/fl



MafB2

```
MafB2_peaks_rep1 <- read.table("MafB2_WT_R1.seacr.peaks.stringent.bed",
  header = FALSE, sep = "\t")

MafB2_peaks_rep2 <- read.table("MafB2_WT_R2.seacr.peaks.stringent.bed",
  header = FALSE, sep = "\t")

MafB2_peaks_consensus <- read.table("MafB2_WT.seacr.consensus.peak_counts.bed",
  header = FALSE, sep = "\t")

# peaks rep1
nrow(MafB2_peaks_rep1)

## [1] 24776

# peaks rep2
nrow(MafB2_peaks_rep2)

## [1] 15978

# consensus peaks
nrow(MafB2_peaks_consensus)

## [1] 28717
```

```

# shared peaks
length(which(MafB2_peaks_consensus[, 10] == 2))

## [1] 11000

MafB2_peaks_merged <- MafB2_peaks_consensus[MafB2_peaks_consensus[,
  10] == 2, ]

split_scores <- strsplit(as.character(MafB2_peaks_merged$V6),
  ", ")

means <- sapply(split_scores, function(x) {
  if (all(is.na(x)))
    return(NA_real_)
  nums <- as.numeric(x)
  mean(nums, na.rm = TRUE)
})

MafB2_peaks_merged$V4 <- means

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 MafB2_WT_R1/ MafB2_WT_R2/ MafB2_KO_R1/ MafB2_KO_R2/ > MafB2_peaks_consensus

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..cmd.annotatePeaks.pl.MafB2_peaks_merged.bed.mm10"

colnames(MafB2_counts)[colnames(MafB2_counts) == "MafB2_WT_R1..Tag.Count.in.given.bp..5198073.0.Total..0"] <- "MafB2_WT_R1..Tag.Count.in.given.bp..5198073.0.Total..0"

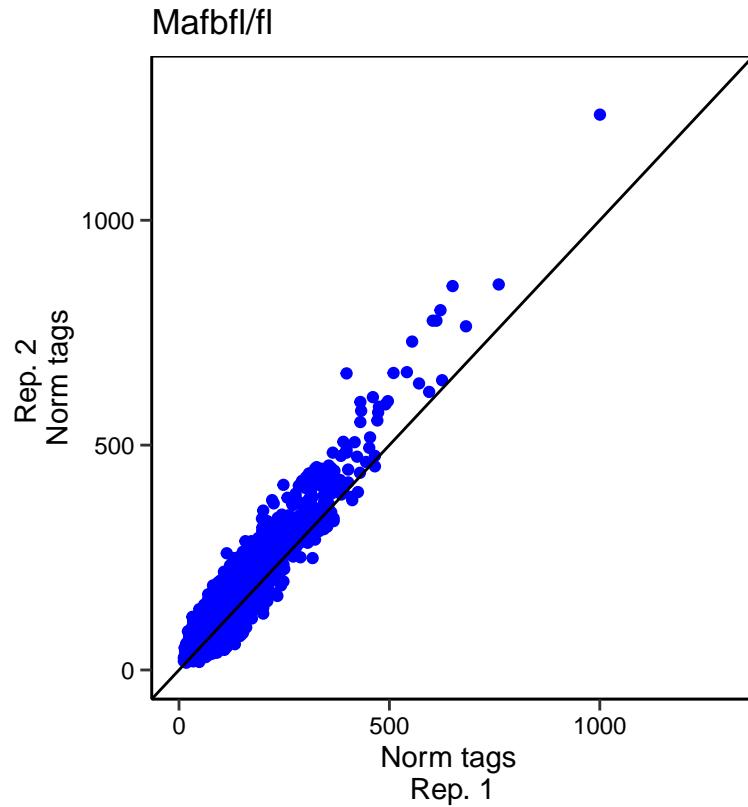
colnames(MafB2_counts)[colnames(MafB2_counts) == "MafB2_WT_R2..Tag.Count.in.given.bp..5587453.0.Total..0"] <- "MafB2_WT_R2..Tag.Count.in.given.bp..5587453.0.Total..0"

colnames(MafB2_counts)[colnames(MafB2_counts) == "MafB2_KO_R1..Tag.Count.in.given.bp..6219029.0.Total..0"] <- "MafB2_KO_R1..Tag.Count.in.given.bp..6219029.0.Total..0"

colnames(MafB2_counts)[colnames(MafB2_counts) == "MafB2_KO_R2..Tag.Count.in.given.bp..6260374.0.Total..0"] <- "MafB2_KO_R2..Tag.Count.in.given.bp..6260374.0.Total..0"

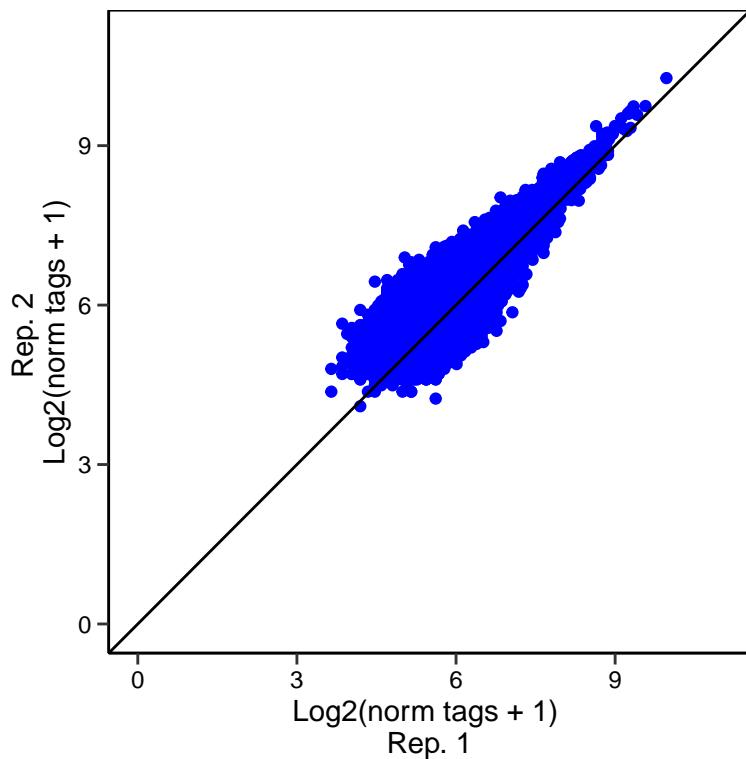
ggplot(data = MafB2_counts, aes(x = MafB2_WT_R1, y = MafB2_WT_R2)) +
  geom_point(show.legend = FALSE, colour = "blue") + geom_abline(slope = 1) +
  ggtitle("Mafbfl1/f1") + xlab("Norm tags \nRep. 1") + ylab("Rep. 2 \nNorm tags") +
  xlim(0, 1300) + ylim(0, 1300) + theme_classic() + theme(axis.text.x = 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"))

```



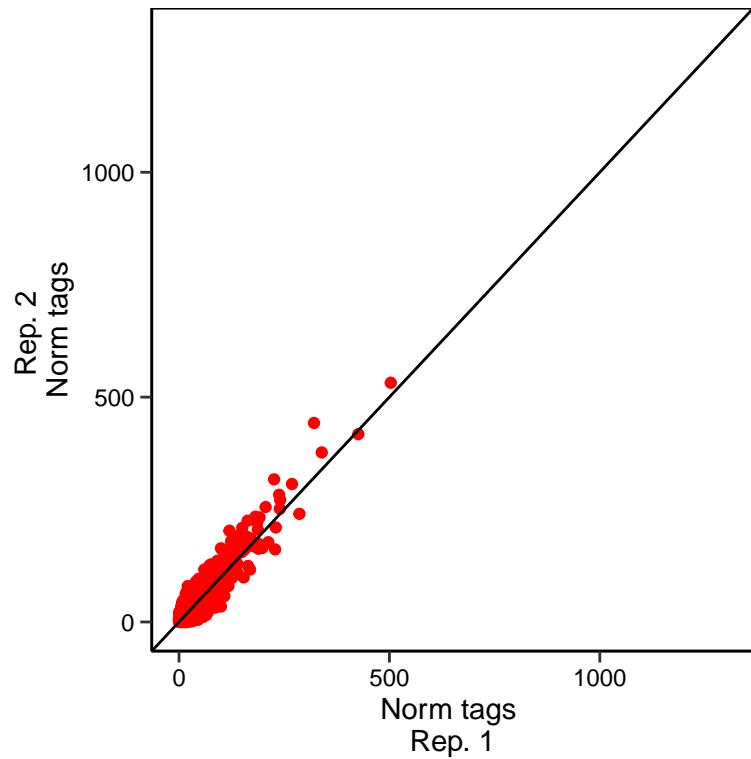
```
ggplot(data = MafB2_counts, aes(x = log2(MafB2_WT_R1 + 1), y = log2(MafB2_WT_R2 + 1))) + geom_point(show.legend = FALSE, colour = "blue") +
  geom_abline(slope = 1) + ggtitle("Mafbfl/fl") + xlab("Log2(norm tags + 1) \nRep. 1") +
  ylab("Rep. 2 \nLog2(norm tags + 1)") + xlim(0, 11) + ylim(0, 11) +
  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"))
```

Mafbfl/fl

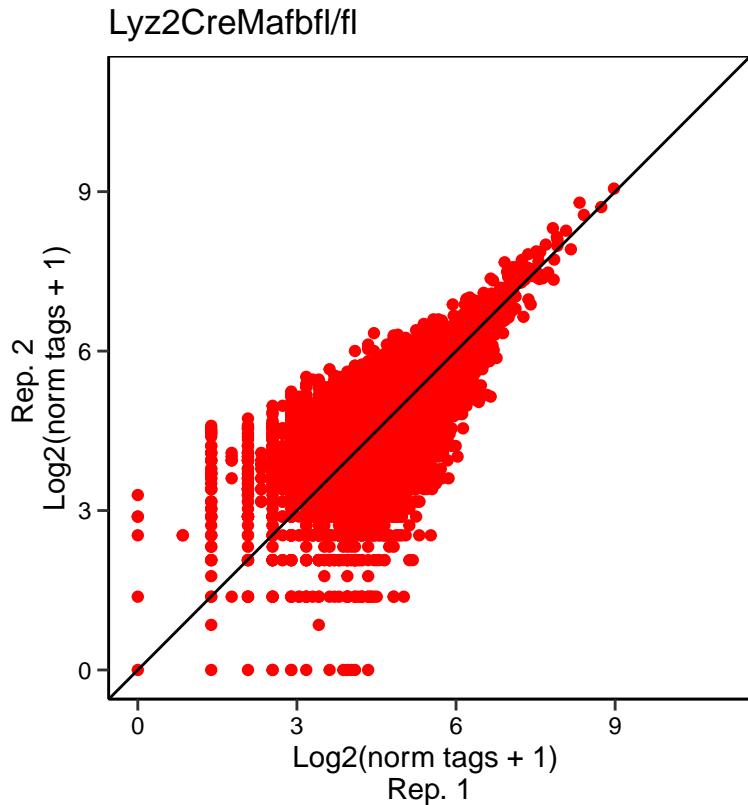


```
ggplot(data = MafB2_counts, aes(x = MafB2_KO_R1, y = MafB2_KO_R2)) +
  geom_point(show.legend = FALSE, colour = "red") + geom_abline(slope = 1) +
  ggtitle("Lyz2CreMafbfl/f1") + xlab("Norm tags \nRep. 1") +
  ylab("Rep. 2 \nNorm tags") + xlim(0, 1300) + ylim(0, 1300) +
  theme_classic() + theme(axis.text.x = 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"))
```

Lyz2CreMafbfl/fl



```
ggplot(data = MafB2_counts, aes(x = log2(MafB2_KO_R1 + 1), y = log2(MafB2_KO_R2 + 1))) + geom_point(show.legend = FALSE, colour = "red") +  
  geom_abline(slope = 1) + ggtitle("Lyz2CreMafbfl/fl") + xlab("Log2(norm tags + 1) \nRep. 1") +  
  ylab("Rep. 2 \nLog2(norm tags + 1)") + xlim(0, 11) + ylim(0,  
  11) + 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 MafB2_WT_R1/ MafB2_WT_R2/ MafB2_KO_1

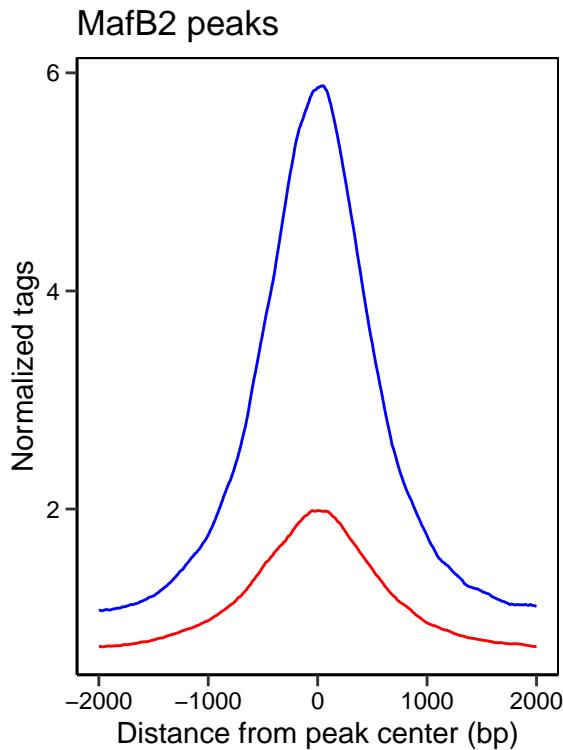
MafB2_hist <- read.table("MafB2_hist.txt", header = TRUE, sep = "\t")

colnames(MafB2_hist)[colnames(MafB2_hist) == "Distance.from.Center..cmd.annotatePeaks.pl.MafB2_peaks_me"]

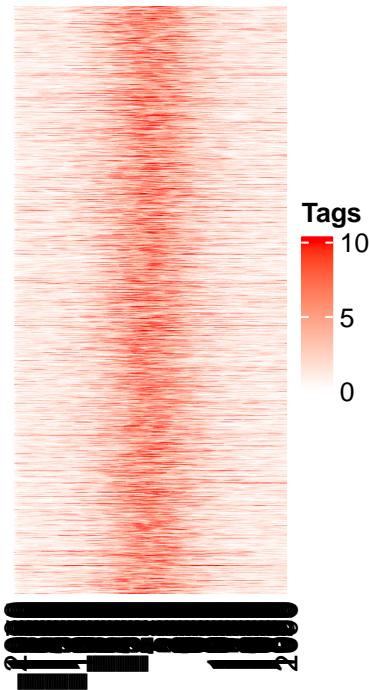
MafB2_hist$MafB2_WT <- rowMeans(MafB2_hist[, c("MafB2_WT_R1..Coverage",
  "MafB2_WT_R2..Coverage")])

MafB2_hist$MafB2_KO <- rowMeans(MafB2_hist[, c("MafB2_KO_R1..Coverage",
  "MafB2_KO_R2..Coverage")])

ggplot(data = MafB2_hist, aes(x = Distance.from.Center, y = MafB2_WT)) +
  geom_line(show.legend = FALSE, colour = "blue") + geom_line(show.legend = FALSE,
  colour = "red", aes(x = Distance.from.Center, y = MafB2_KO)) +
  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"))
```



Mafbfl/fl



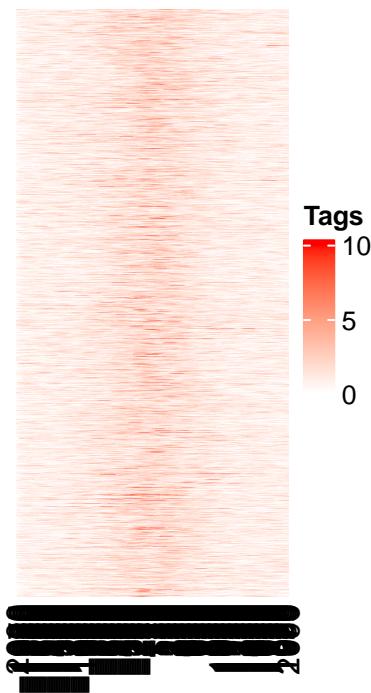
```
col_fun = colorRamp2(c(0, 10), c("white", "red"))

Heatmap(MafB2_K0_heatmap, col_fun, cluster_columns = F, cluster_rows = F,
        heatmap_legend_param = list(title = "Tags"), use_raster = TRUE,
        raster_quality = 10, , column_title = "LyzCreMafbfl/fl")
```

```
## 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.
```

LyzCreMafbfl/fl



MafB3

```
MafB3_peaks_rep1 <- read.table("MafB3_WT_R1.seacr.peaks.stringent.bed",
  header = FALSE, sep = "\t")

MafB3_peaks_rep2 <- read.table("MafB3_WT_R2.seacr.peaks.stringent.bed",
  header = FALSE, sep = "\t")

MafB3_peaks_consensus <- read.table("MafB3_WT.seacr.consensus.peak_counts.bed",
  header = FALSE, sep = "\t")

# peaks rep1
nrow(MafB3_peaks_rep1)

## [1] 31914

# peaks rep2
nrow(MafB3_peaks_rep2)

## [1] 50969

# consensus peaks
nrow(MafB3_peaks_consensus)

## [1] 55159
```

```

# shared peaks
length(which(MafB3_peaks_consensus[, 10] == 2))

## [1] 23975

MafB3_peaks_merged <- MafB3_peaks_consensus[MafB3_peaks_consensus[,
  10] == 2, ]

split_scores <- strsplit(as.character(MafB3_peaks_merged$V6),
  ", ")

means <- sapply(split_scores, function(x) {
  if (all(is.na(x)))
    return(NA_real_)
  nums <- as.numeric(x)
  mean(nums, na.rm = TRUE)
})

MafB3_peaks_merged$V4 <- means

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 MafB3_WT_R1/ MafB3_WT_R2/ MafB3_KO_R1/ MafB3_KO_R2/ > MafB3_peaks_merged.bed

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..cmd.annotatePeaks.pl.MafB3_peaks_merged.bed.mm10"

colnames(MafB3_counts)[colnames(MafB3_counts) == "MafB3_WT_R1..Tag.Count.in.given.bp..6212688.0.Total..0"] <- "MafB3_WT_R1..Tag.Count.in.given.bp..6212688.0.Total..0"

colnames(MafB3_counts)[colnames(MafB3_counts) == "MafB3_WT_R2..Tag.Count.in.given.bp..5747494.0.Total..0"] <- "MafB3_WT_R2..Tag.Count.in.given.bp..5747494.0.Total..0"

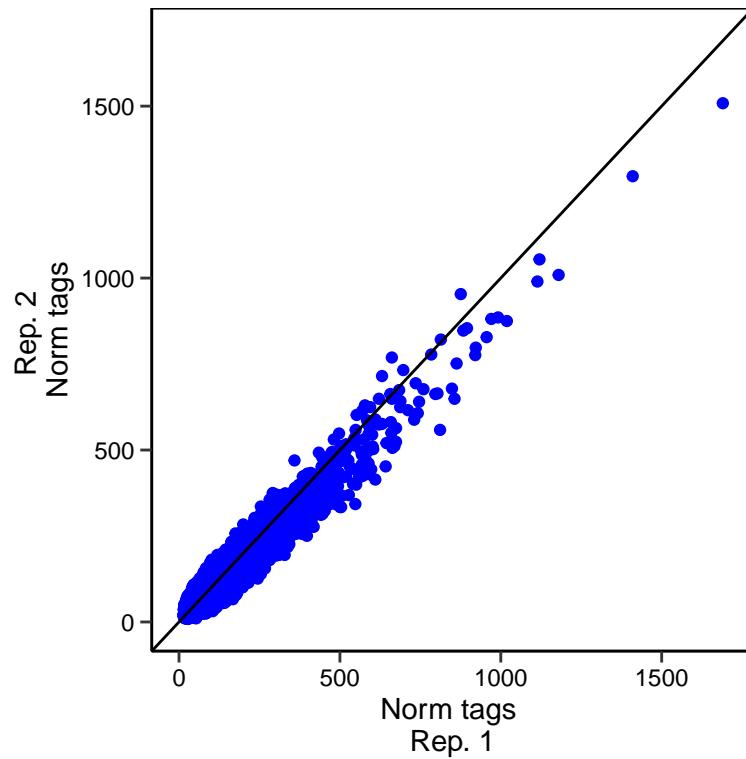
colnames(MafB3_counts)[colnames(MafB3_counts) == "MafB3_KO_R1..Tag.Count.in.given.bp..6221759.0.Total..0"] <- "MafB3_KO_R1..Tag.Count.in.given.bp..6221759.0.Total..0"

colnames(MafB3_counts)[colnames(MafB3_counts) == "MafB3_KO_R2..Tag.Count.in.given.bp..5779352.0.Total..0"] <- "MafB3_KO_R2..Tag.Count.in.given.bp..5779352.0.Total..0"

ggplot(data = MafB3_counts, aes(x = MafB3_WT_R1, y = MafB3_WT_R2)) +
  geom_point(show.legend = FALSE, colour = "blue") + geom_abline(slope = 1) +
  ggtitle("Mafbfl/f1") + xlab("Norm tags \nRep. 1") + ylab("Rep. 2 \nNorm tags") +
  xlim(0, 1700) + ylim(0, 1700) + 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"))

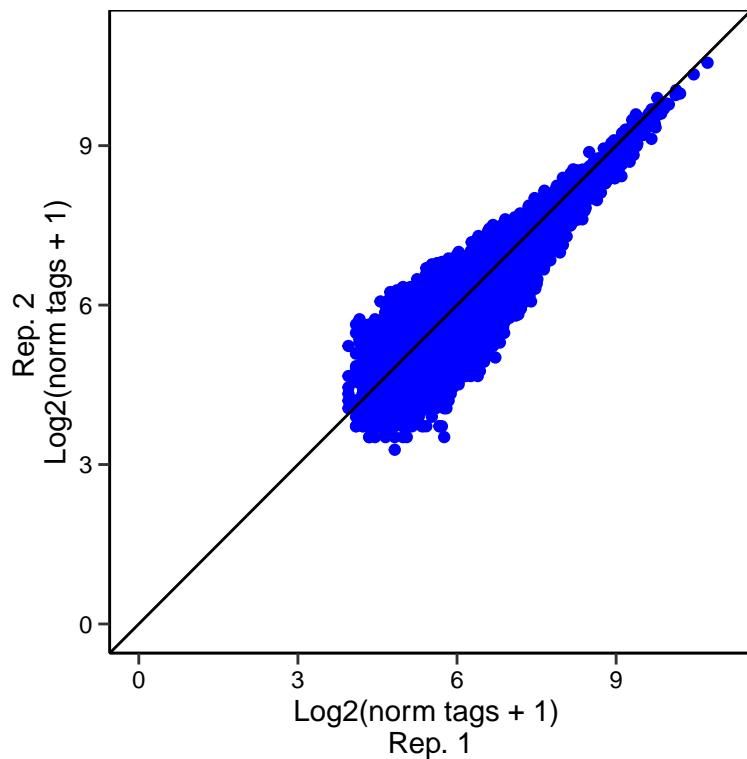
```

Mafbfl/fl



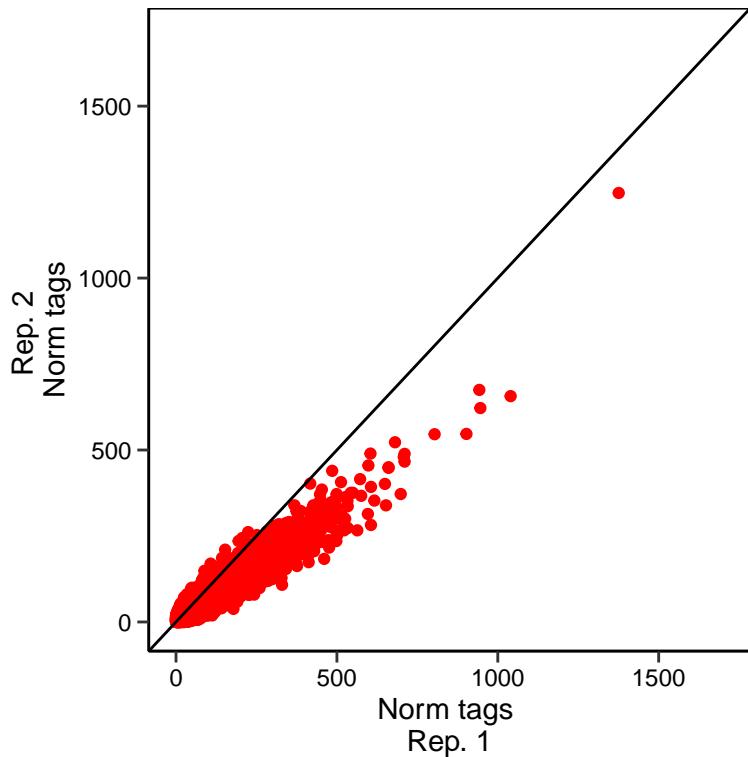
```
ggplot(data = MafB3_counts, aes(x = log2(MafB3_WT_R1 + 1), y = log2(MafB3_WT_R2 + 1))) + geom_point(show.legend = FALSE, colour = "blue") +  
  geom_abline(slope = 1) + ggtitle("Mafbfl/fl") + xlab("Log2(norm tags + 1) \nRep. 1") +  
  ylab("Rep. 2 \nLog2(norm tags + 1)") + xlim(0, 11) + ylim(0,  
  11) + 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"))
```

Mafbfl/fl



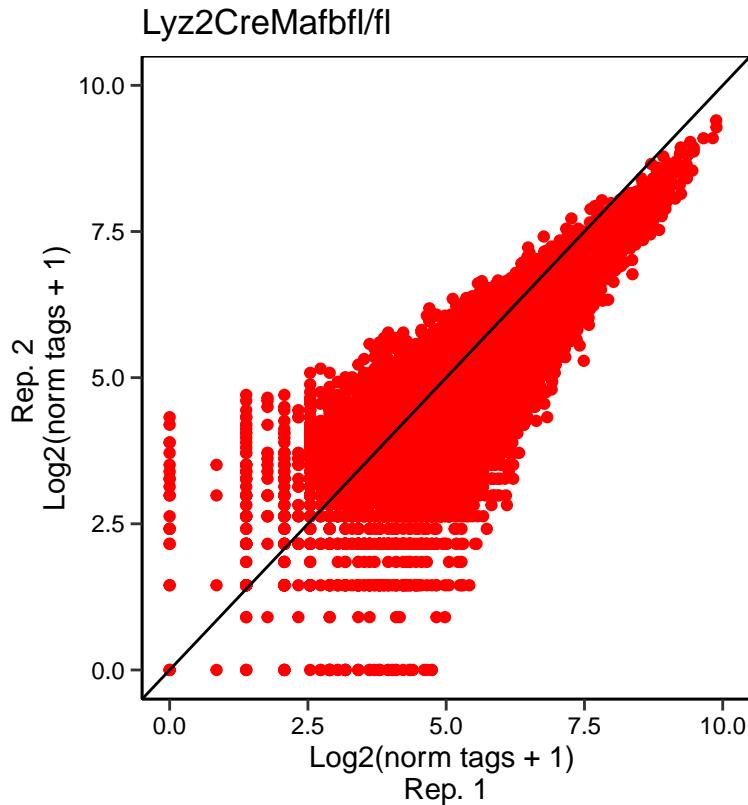
```
ggplot(data = MafB3_counts, aes(x = MafB3_KO_R1, y = MafB3_KO_R2)) +
  geom_point(show.legend = FALSE, colour = "red") + geom_abline(slope = 1) +
  ggtitle("Lyz2CreMafbfl/f1") + xlab("Norm tags \nRep. 1") +
  ylab("Rep. 2 \nNorm tags") + xlim(0, 1700) + ylim(0, 1700) +
  theme_classic() + theme(axis.text.x = 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"))
```

Lyz2CreMafbfl/fl



```
ggplot(data = MafB3_counts, aes(x = log2(MafB3_KO_R1 + 1), y = log2(MafB3_KO_R2 + 1))) + geom_point(show.legend = FALSE, colour = "red") +  
  geom_abline(slope = 1) + ggtitle("Lyz2CreMafbfl/fl") + xlab("Log2(norm tags + 1) \nRep. 1") +  
  ylab("Rep. 2 \nLog2(norm tags + 1)") + xlim(0, 10) + ylim(0,  
  10) + 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"))
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range  
## ('geom_point()').
```



```
annotatePeaks.pl MafB3_peaks_merged.bed mm10 -size 4000 -hist 10 -d MafB3_WT_R1/ MafB3_WT_R2/ MafB3_KO_1

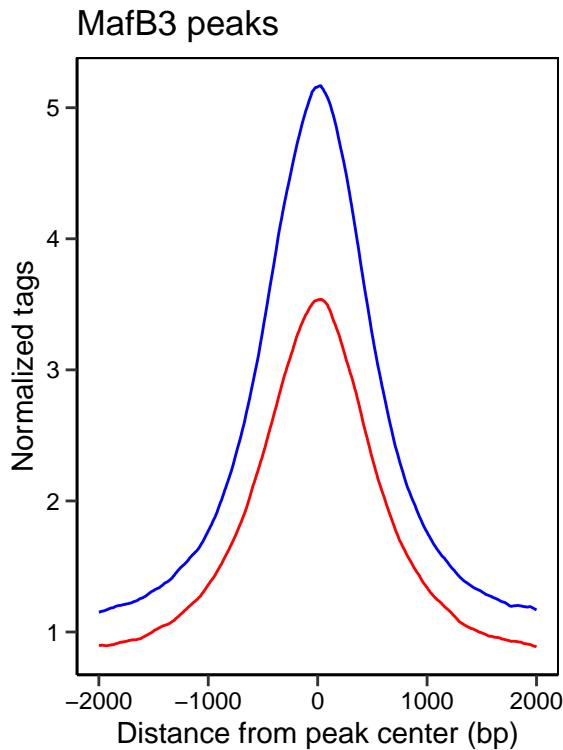
MafB3_hist <- read.table("MafB3_hist.txt", header = TRUE, sep = "\t")

colnames(MafB3_hist)[colnames(MafB3_hist) == "Distance.from.Center..cmd.annotatePeaks.pl.MafB3_peaks_me"]

MafB3_hist$MafB3_WT <- rowMeans(MafB3_hist[, c("MafB3_WT_R1..Coverage",
  "MafB3_WT_R2..Coverage")])

MafB3_hist$MafB3_KO <- rowMeans(MafB3_hist[, c("MafB3_KO_R1..Coverage",
  "MafB3_KO_R2..Coverage")])

ggplot(data = MafB3_hist, aes(x = Distance.from.Center, y = MafB3_WT)) +
  geom_line(show.legend = FALSE, colour = "blue") + geom_line(show.legend = FALSE,
  colour = "red", aes(x = Distance.from.Center, y = MafB3_KO)) +
  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 MafB3_WT_R1/ MafB3_WT_R2/ Ma

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

MafB3_WT_R1_heatmap <- MafB3_heatmap[, 2:162]
MafB3_WT_R2_heatmap <- MafB3_heatmap[, 163:323]
MafB3_WT_heatmap <- (MafB3_WT_R1_heatmap + MafB3_WT_R2_heatmap)/2

MafB3_KO_R1_heatmap <- MafB3_heatmap[, 324:484]
MafB3_KO_R2_heatmap <- MafB3_heatmap[, 485:645]
MafB3_KO_heatmap <- (MafB3_KO_R1_heatmap + MafB3_KO_R2_heatmap)/2

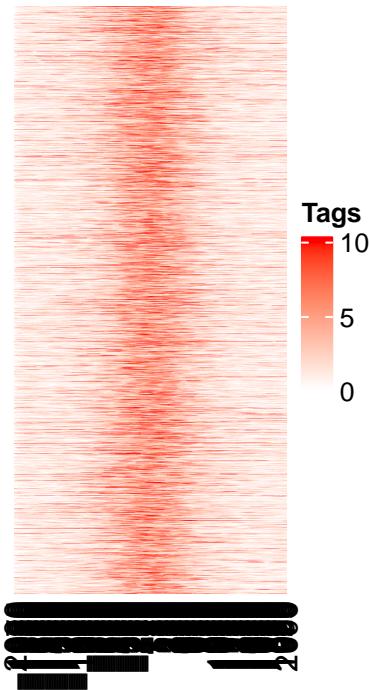
col_fun = colorRamp2(c(0, 10), c("white", "red"))

Heatmap(MafB3_WT_heatmap, col_fun, cluster_columns = F, cluster_rows = F,
  heatmap_legend_param = list(title = "Tags"), use_raster = TRUE,
  raster_quality = 10, column_title = "Mafbfl/fl")

## 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.
```

Mafbfl/fl



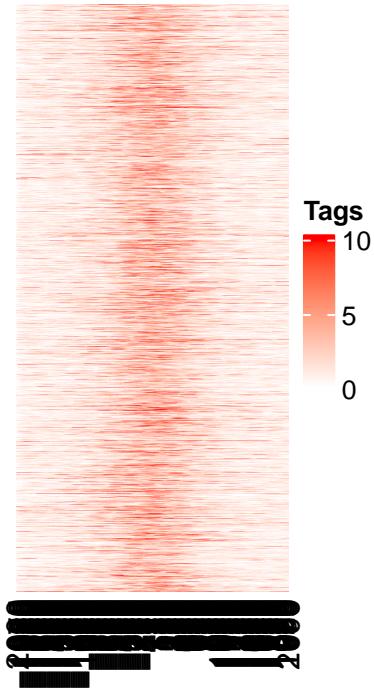
```
col_fun = colorRamp2(c(0, 10), c("white", "red"))

Heatmap(MafB3_K0_heatmap, col_fun, cluster_columns = F, cluster_rows = F,
        heatmap_legend_param = list(title = "Tags"), use_raster = TRUE,
        raster_quality = 10, , column_title = "LyzCreMafbfl/fl")
```

```
## 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.
```

LyzCreMafbfl/fl



Peak selection (WT vs KO)

MafB1

```

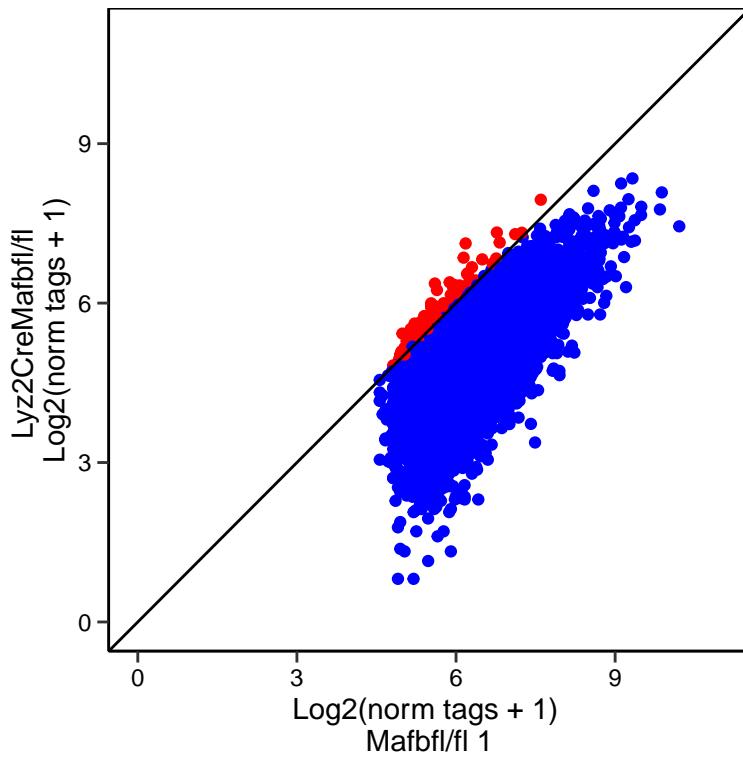
MafB1_counts$MafB1_WT <- rowMeans(MafB1_counts[, c("MafB1_WT_R1",
  "MafB1_WT_R2")])
MafB1_counts$MafB1_KO <- rowMeans(MafB1_counts[, c("MafB1_KO_R1",
  "MafB1_KO_R2")])

MafB1_counts$log2_MafB1_WT <- log2(MafB1_counts$MafB1_WT + 1)
MafB1_counts$log2_MafB1_KO <- log2(MafB1_counts$MafB1_KO + 1)
MafB1_counts$below_diag <- MafB1_counts$log2_MafB1_KO < MafB1_counts$log2_MafB1_WT

ggplot(data = MafB1_counts, aes(x = log2_MafB1_WT, y = log2_MafB1_KO)) +
  geom_point(aes(color = below_diag), show.legend = FALSE) +
  scale_color_manual(values = c(`FALSE` = "red", `TRUE` = "blue")) +
  geom_abline(slope = 1) + ggtitle("MafB1 Peaks") + xlab("Log2(norm tags + 1) \nMafbfl/fl 1") +
  ylab("Lyz2CreMafbfl/fl \nLog2(norm tags + 1)") + xlim(0,
  11) + ylim(0, 11) + 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$MafB2_WT <- rowMeans(MafB2_counts[, c("MafB2_WT_R1",
  "MafB2_WT_R2")])
MafB2_counts$MafB2_KO <- rowMeans(MafB2_counts[, c("MafB2_KO_R1",
  "MafB2_KO_R2")])

MafB2_counts$log2_MafB2_WT <- log2(MafB2_counts$MafB2_WT + 1)
MafB2_counts$log2_MafB2_KO <- log2(MafB2_counts$MafB2_KO + 1)
MafB2_counts$below_diag <- MafB2_counts$log2_MafB2_KO < MafB2_counts$log2_MafB2_WT

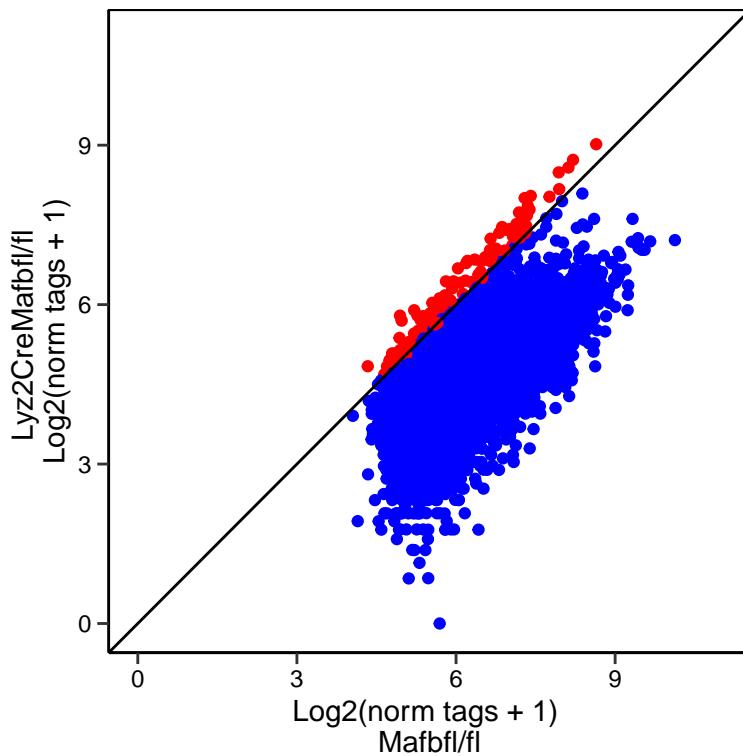
ggplot(data = MafB2_counts, aes(x = log2_MafB2_WT, y = log2_MafB2_KO)) +
  geom_point(aes(color = below_diag), show.legend = FALSE) +
  scale_color_manual(values = c(`FALSE` = "red", `TRUE` = "blue")) +
  geom_abline(slope = 1) + ggtitle("MafB2 Peaks") + xlab("Log2(norm tags + 1) \nMafbfl/fl 1") +
  ylab("Lyz2CreMafbfl/fl \nLog2(norm tags + 1)") + xlim(0,
  11) + ylim(0, 11) + 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$MafB3_WT <- rowMeans(MafB3_counts[, c("MafB3_WT_R1",
  "MafB3_WT_R2")])
MafB3_counts$MafB3_KO <- rowMeans(MafB3_counts[, c("MafB3_KO_R1",
  "MafB3_KO_R2")])

MafB3_counts$log2_MafB3_WT <- log2(MafB3_counts$MafB3_WT + 1)
MafB3_counts$log2_MafB3_KO <- log2(MafB3_counts$MafB3_KO + 1)
MafB3_counts$below_diag <- MafB3_counts$log2_MafB3_KO < MafB3_counts$log2_MafB3_WT

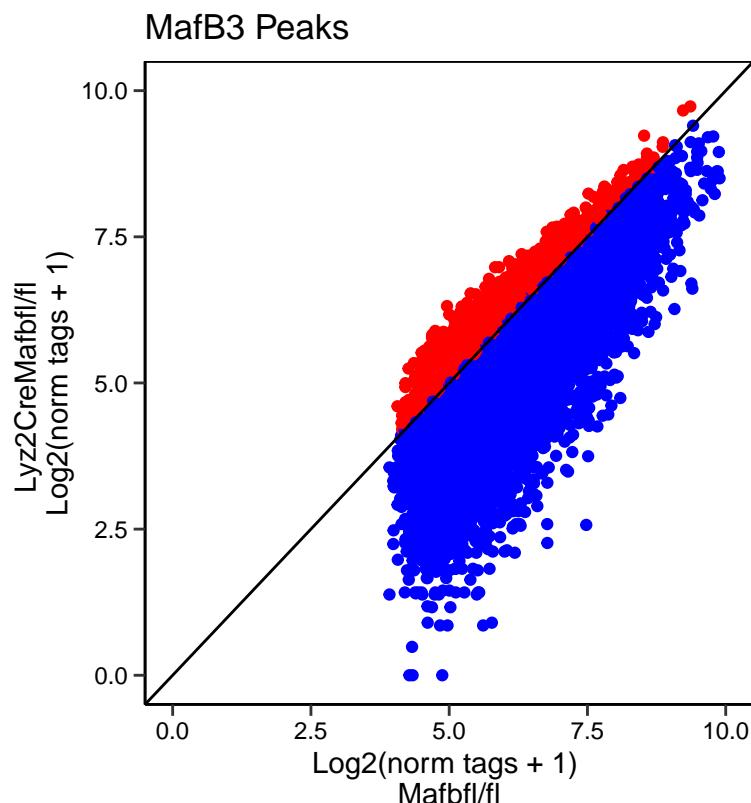
```

```

ggplot(data = MafB3_counts, aes(x = log2_MafB3_WT, y = log2_MafB3_KO)) +
  geom_point(aes(color = below_diag), show.legend = FALSE) +
  scale_color_manual(values = c(`FALSE` = "red", `TRUE` = "blue")) +
  geom_abline(slope = 1) + ggtitle("MafB3 Peaks") + xlab("Log2(norm tags + 1) \nMafbfl/f1") +
  ylab("Lyz2CreMafbfl/f1 \nLog2(norm tags + 1)") + xlim(0,
  10) + ylim(0, 10) + 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"))

```

Warning: Removed 6 rows containing missing values or values outside the scale range
('geom_point()'').



```

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
##          400        2517       3091       2209       205
```

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
##          565        3165       3891       2986       251
```

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
##          1113        6036       7625       5069       589
```

Overlap with BMDM peaks

MafB1

```
MafB1_BMDM_peaks <- read.table("MafB1_BMDM_peaks.bed", header = FALSE,
  sep = "\t")
MafB1_peaks <- read.table("MafB1_peaks.bed", header = FALSE,
  sep = "\t")

nrow(MafB1_BMDM_peaks)

## [1] 7852

nrow(MafB1_peaks)

## [1] 8423

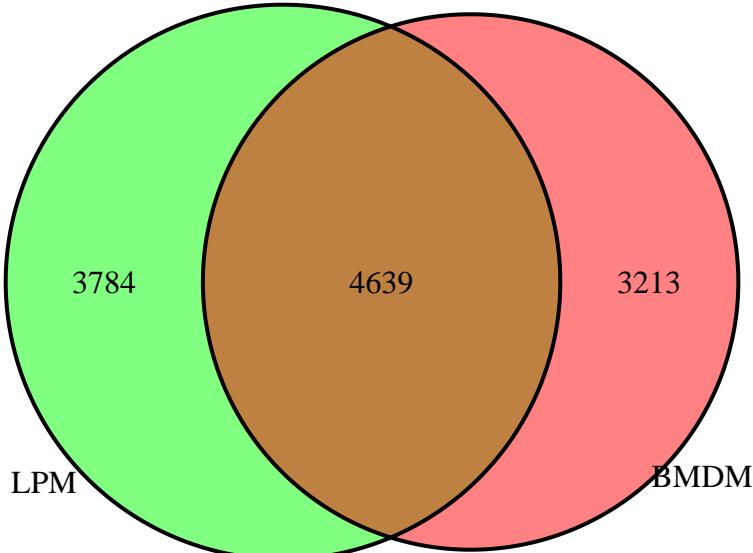
bedtools window -a MafB1_peaks.bed -b MafB1_BMDM_peaks.bed -w 100 > MafB1_overlap.bed

MafB1_overlap <- read.table("MafB1_overlap.bed", header = FALSE,
  sep = "\t")

nrow(MafB1_overlap)

## [1] 4639

draw.pairwise.venn(area1 = 7852, area2 = 8423, cross.area = 4639,
  category = c("BMDM", "LPM"), fill = c("red", "green"))
```



```
## (polygon[GRID.polygon.2009], polygon[GRID.polygon.2010], polygon[GRID.polygon.2011], polygon[GRID.polygon.2012])  
grid.newpage()
```

MafB2

```
MafB2_BMDM_peaks <- read.table("MafB2_BMDM_peaks.bed", header = FALSE,  
sep = "\t")  
MafB2_peaks <- read.table("MafB2_peaks.bed", header = FALSE,  
sep = "\t")  
nrow(MafB2_BMDM_peaks)
```

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

```
nrow(MafB2_peaks)
```

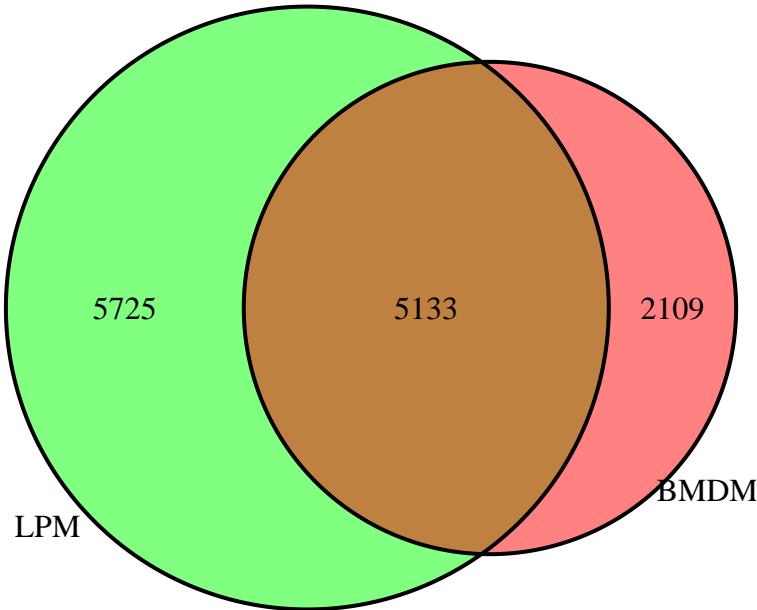
```
## [1] 10858
```

```
bedtools window -a MafB2_peaks.bed -b MafB2_BMDM_peaks.bed -w 100 > MafB2_overlap.bed
```

```
MafB2_overlap <- read.table("MafB2_overlap.bed", header = FALSE,  
sep = "\t")  
nrow(MafB2_overlap)
```

```
## [1] 5133
```

```
draw.pairwise.venn(area1 = 7242, area2 = 10858, cross.area = 5133,  
category = c("BMDM", "LPM"), fill = c("red", "green"))
```



```
## (polygon[GRID.polygon.2018], polygon[GRID.polygon.2019], polygon[GRID.polygon.2020], polygon[GRID.po
grid.newpage()
```

MafB3

```
MafB3_BMDM_peaks <- read.table("MafB3_BMDM_peaks.bed", header = FALSE,
sep = "\t")
MafB3_peaks <- read.table("MafB3_peaks.bed", header = FALSE,
sep = "\t")

nrow(MafB3_BMDM_peaks)
```

```
## [1] 2338
```

```
nrow(MafB3_peaks)
```

```
## [1] 20432
```

```
bedtools window -a MafB3_peaks.bed -b MafB3_BMDM_peaks.bed -w 100 > MafB3_overlap.bed
```

```
MafB3_overlap <- read.table("MafB3_overlap.bed", header = FALSE,
sep = "\t")
```

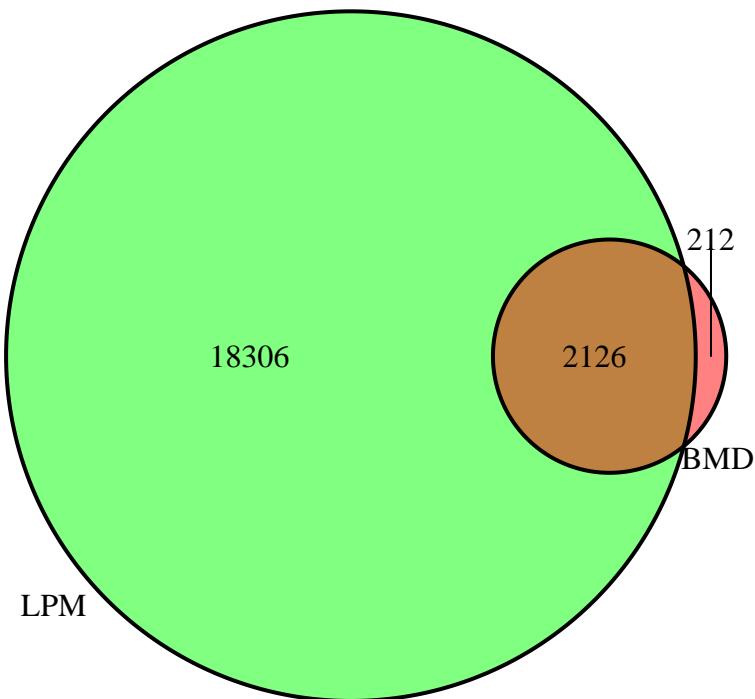
```
nrow(MafB3_overlap)
```

```

## [1] 2126

draw.pairwise.venn(area1 = 2338, area2 = 20432, cross.area = 2126,
category = c("BMDM", "LPM"), fill = c("red", "green"))

```



```

## (polygon[GRID.polygon.2027], polygon[GRID.polygon.2028], polygon[GRID.polygon.2029], polygon[GRID.po
grid.newpage()

```

MafB target genes

MafB1

```

MafB1_LPM_genes <- MafB1_peaks_annot[MafB1_peaks_annot[, "Distance.to.TSS"] >=
-2000 & MafB1_peaks_annot[, "Distance.to.TSS"] <= 2000, ]

MafB1_LPM_genes <- unique(MafB1_LPM_genes$Gene.Name)
MafB1_LPM_genes <- na.omit(MafB1_LPM_genes)

```

```

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

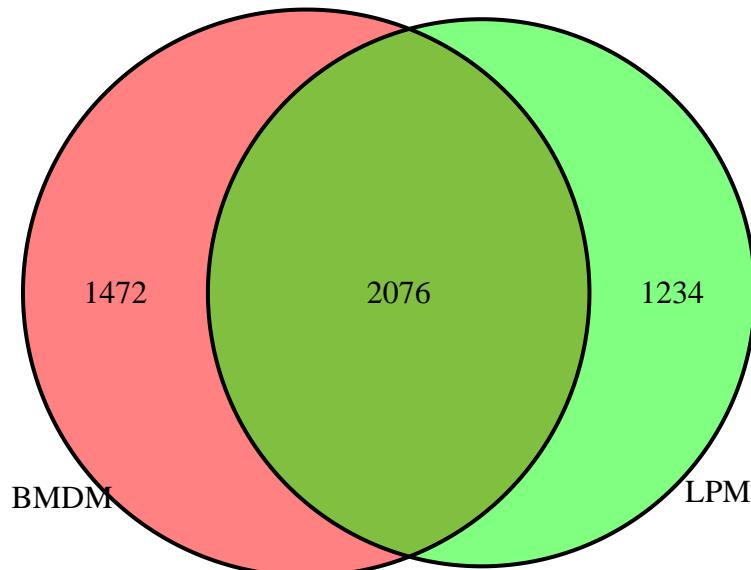
MafB1_BMDM_genes <- MafB1_BMDM_peaks_annot[MafB1_BMDM_peaks_annot[, "Distance.to.TSS"] >= -2000 & MafB1_BMDM_peaks_annot[, "Distance.to.TSS"] <= 2000, ]

```

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

```
# 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(LPM = MafB1_LPM_genes, BMDM = MafB1_BMDM_genes)
# pdf(file =
# 'C:/Users/domie/Documents/CUTandRUN/Venn_MafB1_Mac.pdf',
# width = 4, height = 4)
display_venn(x, fill = c("green", "red"))
```



```
# dev.off()
```

```
g <- readGFF("C:/Users/domie/Documents/CUTandRUN/genes.gtf")
pc <- g %>%
  dplyr::filter(type == "gene")
rm(g)
dim(pc)
# genes 22597
```

```

dat <- matrix(c(1307, 2003, 1305, 17982), nrow = 2, dimnames = list(Mac_sign = c("YES",
  "NO"), MafB1_target = c("YES", "NO")))
dat

##           MafB1_target
## Mac_sign   YES     NO
##       YES 1307  1305
##       NO   2003 17982

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:
##  8.213355 9.841977
## sample estimates:
## odds ratio
##    8.989829

```

MafB2

```

MafB2_LPM_genes <- MafB2_peaks_annot[MafB2_peaks_annot[, "Distance.to.TSS"] >=
  -2000 & MafB2_peaks_annot[, "Distance.to.TSS"] <= 2000, ]

MafB2_LPM_genes <- unique(MafB2_LPM_genes$Gene.Name)
MafB2_LPM_genes <- na.omit(MafB2_LPM_genes)

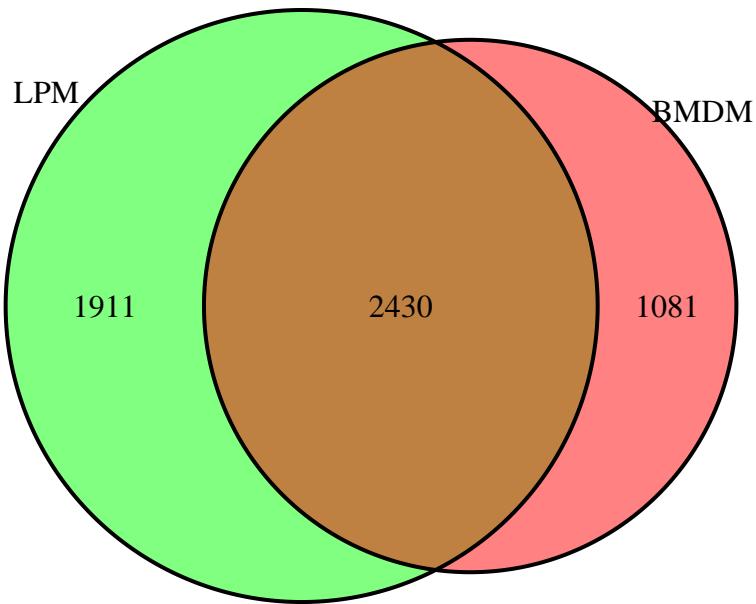
MafB2_BMDM_peaks_annot <- read.table("MafB2_BMDM_peaks_annot.txt",
  header = TRUE, sep = "\t")

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)

x <- list(LPM = MafB2_LPM_genes, BMDM = MafB2_BMDM_genes)
# pdf(file =
# 'C:/Users/domie/Documents/CUTandRUN/Venn_MafB1_Mac.pdf',
# width = 4, height = 4)
display_venn(x, fill = c("green", "red"))

```



```
# dev.off()

dat <- matrix(c(1528, 2813, 969, 17287), nrow = 2, dimnames = list(Mac_sign = c("YES",
  "NO"), MafB2_target = c("YES", "NO")))
dat

##           MafB2_target
## Mac_sign   YES      NO
##       YES 1528    969
##       NO   2813 17287

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:
##  8.847346 10.612952
## sample estimates:
## odds ratio
##  9.686429
```

MafB3

```
MafB3_LPM_genes <- MafB3_peaks_annot[MafB3_peaks_annot[, "Distance.to.TSS"] >=
-2000 & MafB3_peaks_annot[, "Distance.to.TSS"] <= 2000, ]
```

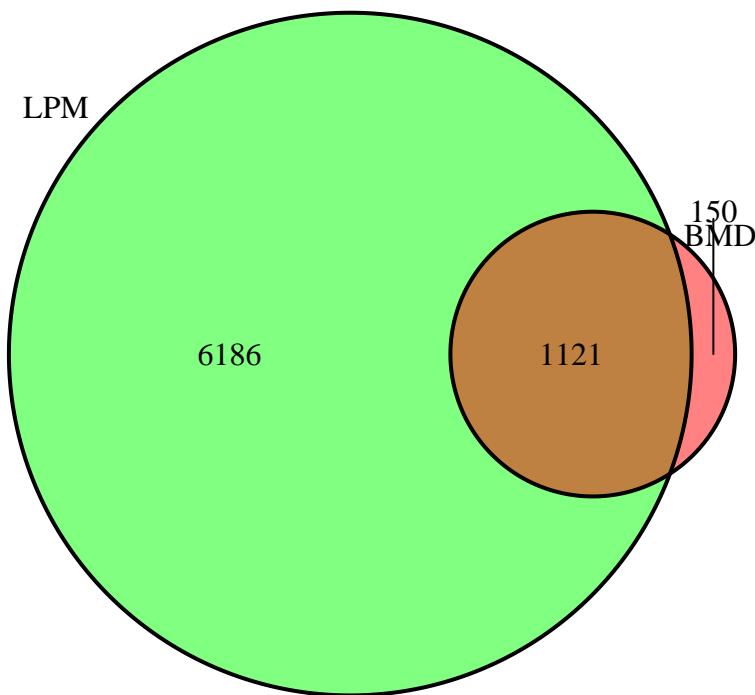
```
MafB3_LPM_genes <- unique(MafB3_LPM_genes$Gene.Name)
MafB3_LPM_genes <- na.omit(MafB3_LPM_genes)
```

```
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)
```

```
x <- list(LPM = MafB3_LPM_genes, BMDM = MafB3_BMDM_genes)
# pdf(file =
# 'C:/Users/domie/Documents/CUTandRUN/Venn_MafB1_Mac.pdf',
# width = 4, height = 4)
display_venn(x, fill = c("green", "red"))
```



```
# dev.off()
```

```

dat <- matrix(c(721, 6586, 167, 15123), nrow = 2, dimnames = list(Mac_sign = c("YES",
  "NO"), MafB1_target = c("YES", "NO")))
dat

##           MafB1_target
## Mac_sign   YES      NO
##       YES  721    167
##       NO   6586  15123

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:
##  8.343927 11.831702
## sample estimates:
## odds ratio
##  9.912454

```