Analysis of DNA methylation's mediation of differential expression

Sam Bogan

10/11/2022

Load required packages

```r
# Load required packages
library(edgeR)
library(tidyverse)
library(ape)
library(vegan)
library(data.table)
library(plyr)
library(mediation)
library(brms)
```

Read in data, add metadata

```r
# Read in csv of read counts per gene
gene_counts <- read.csv("Input_data/gene_read_counts.csv")
exon_perc_meth <- read.csv("Input_data/meth_exon_perc_meth.csv")
int_perc_meth <- read.csv("Input_data/meth_intron_perc_meth.csv")
pr_perc_meth <- read.csv("Input_data/meth_promoter_perc_meth.csv")

# Remove duplicated transcripts
n_occur_gc <- data.frame(table(gene_counts$Geneid))
n_occur_gc <- n_occur_gc[n_occur_gc$Freq > 1, ]
n_occur_gc <- n_occur_gc$Var1

gene_counts <- gene_counts[!gene_counts$Geneid %in% n_occur_gc,]

#Make gene id matrix rowname
row.names(gene_counts) <- gene_counts$Geneid

gene_counts <- subset(gene_counts,
                      select = -c(Geneid,
                                  Chr,
                                  Start,
                                  End,
                                  Strand,
                                  Length))
```

```r
# Replace sample IDs with simple names
colnames( gene_counts ) <- c( "NN1","NN2","NN3","NU1","NU2","NU3",
                     "UN1","UN2","UN3","UU1","UU2","UU3" )

# Create treatment group df
Mat = c( "N","N","N","N","N","N",
         "U","U","U","U","U","U" )

Dev = c( "N","N","N","U","U","U",
         "N","N","N","U","U","U" )

targets_gc <- data.frame( Mat = c( "N","N","N","N","N","N",
                                   "U","U","U","U","U","U" ),
                          Dev = c( "N","N","N","U","U","U",
                                   "N","N","N","U","U","U" ) )

ex_meth <- t(exon_perc_meth[-c(1,2)])

targets_gc$grouping <- paste( targets_gc$Mat,
                              targets_gc$Dev,
                              sep="_" )

# Round counts (if necessary() for use in edgeR
data_input_gc <- round( gene_counts )
```

Normalize RNAseq read counts and plot PCOA

```r
# Make a DGEList
DGEList <- DGEList( counts = data_input_gc,
                    group = targets_gc$grouping,
                    remove.zeros = T )
```

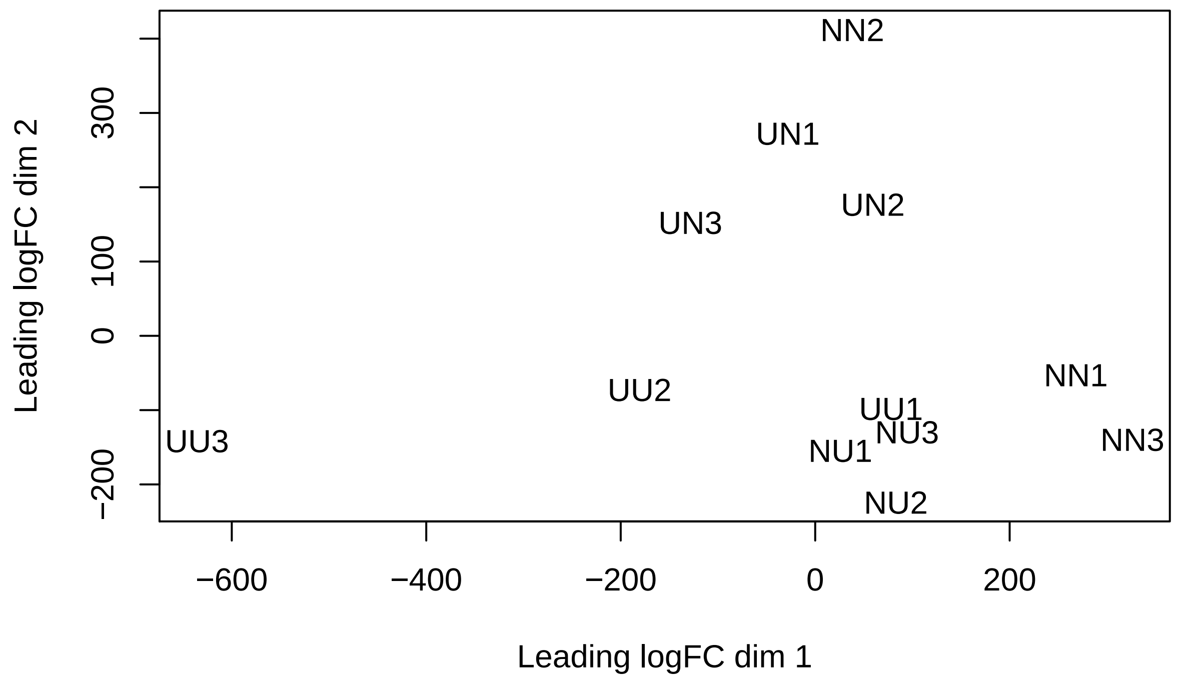
```
## Removing 4952 rows with all zero counts
```

```r
# Let's remove genes with less then 0.5 cpm (this is ~10 counts in the count file) in no fewer then 9 s
DGEList_keep <- rowSums( cpm( DGEList ) > 0.5 ) >= 9

# How many genes are removed by read count filter?
table( DGEList_keep )
```

```
## DGEList_keep
## FALSE  TRUE
##  9029 16303
```

```r
# Filter and set keep.lib.sizes = F to have R recalculate library sizes after filtering
DGEList <- DGEList[ DGEList_keep,
                    keep.lib.sizes = FALSE ]
```

```r
# Create library size normalization factors
DGEList <- calcNormFactors( DGEList )


# CPM conversion and log^2 transformation of read counts
DGEList_log <- cpm( DGEList,
                    log = FALSE,
                    prior.count = 2 )

# MDS of normalized gene read counts
MDS <- plotMDS( DGEList_log )
```



```r
# Print MDS plot
MDS
```

```
## An object of class MDS
## $dim.plot
## [1] 1 2
##
## $distance.matrix
##          NN1      NN2      NN3      NU1      NU2      NU3      UN1      UN2
## NN1   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
## NN2 584.8778   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
## NN3 164.2569 690.8519   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
## NU1 343.8254 586.6116 399.9130   0.0000   0.0000   0.0000   0.0000   0.0000
## NU2 339.1218 653.9205 383.7000 129.9180   0.0000   0.0000   0.0000   0.0000
## NU3 381.6145 563.4832 437.3479 207.2959 181.5580   0.0000   0.0000   0.0000
## UN1 476.2352 250.7085 587.2041 444.6359 521.9478 469.7774   0.0000   0.0000
## UN2 343.9600 324.4443 452.7283 367.8115 430.3751 388.9985 173.9963   0.0000
## UN3 467.5155 442.8330 576.2921 414.7813 486.8882 495.5666 252.0502 234.3086
## UU1 348.1560 535.2235 412.0827 182.3233 184.9990 142.1535 422.4316 333.7291
## UU2 507.0227 554.5517 583.8177 264.0301 327.8696 325.5127 406.5706 374.9803
## UU3 913.7975 896.9307 967.5739 687.2252 750.0288 782.2148 750.4284 775.8103
##          UN3      UU1      UU2 UU3
## NN1   0.0000   0.0000   0.0000   0
## NN2   0.0000   0.0000   0.0000   0
## NN3   0.0000   0.0000   0.0000   0
## NU1   0.0000   0.0000   0.0000   0
## NU2   0.0000   0.0000   0.0000   0
## NU3   0.0000   0.0000   0.0000   0
## UN1   0.0000   0.0000   0.0000   0
## UN2   0.0000   0.0000   0.0000   0
## UN3   0.0000   0.0000   0.0000   0
## UU1 429.8790   0.0000   0.0000   0
## UU2 328.5532 275.8539   0.0000   0
## UU3 616.8622 757.3709 506.5889   0
##
## $cmdscale.out
##           [,1]       [,2]
## NN1  268.07014  -52.44675
## NN2   38.14360  412.20797
## NN3  326.24056 -139.76526
## NU1   25.89297 -154.62485
## NU2   82.98686 -224.32252
## NU3   94.45667 -129.66317
## UN1  -28.16867  272.23348
## UN2   59.28169  176.65190
## UN3 -128.38485  152.25925
## UU1   77.94444  -98.14957
## UU2 -180.64416  -72.74681
## UU3 -635.81925 -141.63368
##
## $top
## [1] 500
##
## $gene.selection
## [1] "pairwise"
##
## $x
##        NN1        NN2        NN3        NU1        NU2        NU3        UN1
##  268.07014   38.14360  326.24056   25.89297   82.98686   94.45667  -28.16867
##        UN2        UN3        UU1        UU2        UU3
##   59.28169 -128.38485   77.94444 -180.64416 -635.81925
##
## $y
##        NN1        NN2        NN3        NU1        NU2        NU3        UN1
##  -52.44675  412.20797 -139.76526 -154.62485 -224.32252 -129.66317  272.23348
##        UN2        UN3        UU1        UU2        UU3
##  176.65190  152.25925  -98.14957  -72.74681 -141.63368
##
## $axislabel
## [1] "Leading logFC dim"
```

```
# Run pcoa on gene read counts
pcoa_gc <- pcoa( vegdist( t( DGEList_log <- cpm ( DGEList,
                                                  log = TRUE,
                                                  prior.count = 2 ) ),
                          method = "euclidean" ) / 1000 )

# Print sample scores across vectors
head( pcoa_gc$vectors )
```

```
##           Axis.1       Axis.2        Axis.3        Axis.4        Axis.5
## NN1 -0.026568073 -0.008656190  0.0070963478  0.0074712453  0.0047101833
## NN2 -0.024468794 -0.003034360  0.0141527713  0.0068743690  0.0081443602
## NN3 -0.032824963 -0.009636454  0.0060014416  0.0171324908 -0.0010127111
## NU1  0.009074835 -0.027294437  0.0009254015 -0.0154988856 -0.0088824152
## NU2  0.012371072 -0.029667844  0.0042784348 -0.0154089285 -0.0002445781
## NU3  0.015286171 -0.028287756 -0.0001437379  0.0005923961  0.0040426530
##            Axis.6        Axis.7       Axis.8       Axis.9      Axis.10
## NN1 -0.0007604117 -0.0009807580  0.002225455  0.008623741 -0.006098279
## NN2 -0.0011895572  0.0076329778 -0.002907772  0.011244749  0.025510481
## NN3  0.0053392889 -0.0116135967  0.002412909 -0.003173279 -0.018145058
## NU1 -0.0114768357  0.0123880511 -0.017211751  0.012510254 -0.011534077
## NU2 -0.0012441025  0.0007414662  0.029566590 -0.004650000  0.002464760
## NU3  0.0122129146 -0.0072038874 -0.017021373 -0.021985404  0.008377531
##          Axis.11
## NN1  0.029261362
## NN2 -0.009893666
## NN3 -0.017051700
## NU1 -0.005029752
## NU2 -0.001978285
## NU3  0.005028531
```

Wrangle data

Melt matrices of read counts and methylation counts into one tabular, merged data frame. This is required for fitting lm() or glm.nb() models.

```
# Write custom code for modelling gene expression as a function of environments, methylation, and envir
logCPM_df <- as.data.frame(DGEList_log)

# Create tabularized df containing all replicates using 'melt' function in reshape2
logCPM_df$geneid <- row.names(logCPM_df)

tab_exp_df <- melt(logCPM_df,
                   id = c("geneid"))

tab_exp_df$geneid <- gsub("transcript:", "",
                          gsub("-tr", "", tab_exp_df$geneid))

tab_exp_df$Mat_treat <- substr(tab_exp_df$variable,1,1)
tab_exp_df$Dev_treat <- substr(tab_exp_df$variable,2,2)

## Melt and merge exon perc meth data with tab_exp_df
# For each transcript, average exon CpG methylation per sample
```

```
exon_perc_meth$geneid <- gsub("-tr.*", "", exon_perc_meth$V41)
int_perc_meth$geneid <- gsub("transcript:", "",
                         gsub("-tr.*", "", int_perc_meth$V41))

all_gw_p_meth <- rbind(exon_perc_meth,
                       int_perc_meth)

all_gw_p_meth <- aggregate(all_gw_p_meth[c(2:14)], list(all_gw_p_meth$geneid), FUN = mean)


all_gw_meth <- melt(all_gw_p_meth[-c(2)],
                    id = c("Group.1"))

all_gw_meth$Group.1 <- gsub("transcript:", "", all_gw_meth$Group.1)

names(all_gw_meth)[names(all_gw_meth) == 'value'] <- 'all_gw_meth'

all_gw_meth$sample_gene <- paste(all_gw_meth$variable,
                                  all_gw_meth$Group.1,
                                  sep = "_")
tab_exp_df$sample_gene <- paste(tab_exp_df$variable,
                                  tab_exp_df$geneid,
                                  sep = "_")

# Merge logCPM and exon methylation datasets

tab_exp_df3 <- merge(tab_exp_df,
                     all_gw_meth[-c(2)],
                    by = "sample_gene")
```

Fit structural equation models to transcripts

```
# Fit SEM models
lm1s <- dlply(tab_exp_df3, c("geneid"), function(df)
lm(all_gw_meth ~ Mat_treat + Dev_treat, data = df))

lm2s <- dlply(tab_exp_df3, c("geneid"), function(df)
lm(value ~ Mat_treat + Dev_treat + all_gw_meth, data = df))

## If dev treat contains NA coefficient, remove from lm1 and lm2
# First, convert lm's to coefficient df's
lm1_coefs <- list()

for (i in 1:length(lm1s)) {
 lm1_coefs[[i]] <- as.data.frame(lm1s[[i]]$effects)[2,1]
}
names(lm1_coefs) <- names(lm1s)

lm2_coefs <- list()

for (i in 1:length(lm2s)) {
```

```r
 lm2_coefs[[i]] <- as.data.frame(lm2s[[i]]$effects)
}
names(lm2_coefs) <- names(lm2s)

lm1_pvals <- list()

# Report significant DM exons

for (i in 1:length(lm1s)) {
 lm1_pvals[[i]] <- as.data.frame(anova(lm1s[[i]]))[1,5]
}
```

```
## Warning in anova.lm(lm1s[[i]]): ANOVA F-tests on an essentially perfect fit are
## unreliable

## Warning in anova.lm(lm1s[[i]]): ANOVA F-tests on an essentially perfect fit are
## unreliable
```

```r
names(lm1_pvals) <- names(lm1s)
```

Mediation analysis

```r
# Mediation of maternal DE effect by DM
mat_mediations <- list() # Create list to add vcov results to

for (i in 1:length(lm1s)) {
 tryCatch({mat_mediations[[i]] <- summary(mediate(lm1s[[i]], lm2s[[i]], sims = 1000, treat = "Mat_treat
 }, error=function(e){})
}
names(mat_mediations) <- names(lm1s)

save(mat_mediations, file = "mat_mediations.RData")
```

Report significant mediated effects

```r
# Extract p-values for indirect effects: E -> M -> GE
ind_p_m <- list()

for (i in 1:length(mat_mediations)) {
 ind_p_m[[i]] <- mat_mediations[[i]]$d0.p
}
names(ind_p_m) <- names(mat_mediations)

# Extract indirect effect confidence intervals
ind_ci_m <- list()

for (i in 1:length(mat_mediations)) {
 ind_ci_m[[i]] <- data.frame(t(as.data.frame(mat_mediations[[i]]$d0.ci)))
}
names(ind_ci_m) <- names(mat_mediations)

ind_p_vm <- list()

# Create df with geneid and indirect effect pvals
for (i in 1:length(ind_p_m)) {
 new_value_m <- as.numeric(ind_p_m[[i]])
 ind_p_vm <- c(ind_p_vm, new_value_m)
}

ind_ci_m_df <- bind_rows(ind_ci_m, .id = "column_label")

ind_ci_vm <- list()

# Create df with geneid and indirect effect pvals
ind_p_dfm <- as.data.frame(t(data.frame(ind_p_vm)))

ind_p_dfm$fdr <- p.adjust(ind_p_dfm$V1, method = "fdr")

nrow(filter(ind_p_dfm, fdr < 0.05)) # 21 genes
```

```
## [1] 21
```

```r
# Filter to include only transcripts with significant mediation parameter estimate
sig_dev_meth_dfm <- filter(ind_ci_m_df, X2.5. > 0 & X97.5. > 0 | X2.5. < 0 & X97.5. < 0) # 66 mat indir
nrow(sig_dev_meth_dfm)
```

```
## [1] 66
```

Export parameters from mediation analysis

```r
# Extract mediation effect
prop_m <- list()

for (i in 1:length(mat_mediations)) {
 prop_m[[i]] <- mat_mediations[[i]]$n1
}
names(prop_m) <- names(mat_mediations)

prop_m_df <- as.data.frame(t(bind_rows(prop_m, .id = "column_label")))
prop_m_df$geneid <- rownames(prop_m_df)

d0_m <- list()

for (i in 1:length(mat_mediations)) {
 d0_m[[i]] <- mat_mediations[[i]]$d0
}

names(d0_m) <- names(mat_mediations)
```

```r
d0_m_df <- as.data.frame(t(bind_rows(d0_m, .id = "column_label")))
d0_m_df$geneid <- rownames(d0_m_df)

int_params_df <- merge(d0_m_df, prop_m_df, by = "geneid")

names(ind_ci_m_df)[names(ind_ci_m_df) == "column_label"] <- "geneid"

int_params_df <- merge(int_params_df,
                       ind_ci_m_df,
                       by = "geneid")

# Extract proportion mediated and ci's and merge with parameter df
pr_ci_m <- list()

for (i in 1:length(mat_mediations)) {
 pr_ci_m[[i]] <- data.frame(t(as.data.frame(mat_mediations[[i]]$n1.ci)))
}
names(pr_ci_m) <- names(mat_mediations)

pr_ci_m_df <- bind_rows(pr_ci_m, .id = "column_label")
names(pr_ci_m_df)[names(pr_ci_m_df) == "column_label"] <- "geneid"

int_params_df <- merge(int_params_df,
                       pr_ci_m_df,
                       by = "geneid")
```

Filter mediation genes according to likelihood of causal direction

```r
# Run line below to prevent RStan from producing a message asking you to install RTools
options(buildtools.check = function(action) TRUE )

## Filter genes based on likelihood of causal direction using looped brms models and bayes_factor tests
tab_exp_df3_filt <- filter(tab_exp_df3, geneid %in% sig_dev_meth_dfm$column_label)

# Fit 'triangle' models
triangle_brms <- dlply(tab_exp_df3_filt, c("geneid"), function(df)
brm(bf(value ~ Mat_treat + Dev_treat + all_gw_meth) +
                      bf(all_gw_meth ~ Mat_treat + Dev_treat) +
                      set_rescor(FALSE),
         data = df,
         family = gaussian(),
         iter = 20000,
         save_mevars = TRUE,
         save_pars = save_pars(all = TRUE)))

save(triangle_brms, file = "triangle_brms.RData")

# Fit 'straight line' models
straight_brms <- dlply(tab_exp_df3_filt, c("geneid"), function(df)
  brm(bf(value ~ Mat_treat + Dev_treat) +
                    bf(all_gw_meth ~ value) +
                    set_rescor(FALSE),
         data = df,
         family = gaussian(),
         iter = 20000,
         save_mevars = TRUE,
         save_pars = save_pars(all = TRUE)))

save(straight_brms, file = "straight_brms.RData")
```

```
# Apply Bayes factor tests with for loop
bf_list <- list()

for (i in 1:length(triangle_brms)) {
 bf_list[[i]] <- bayes_factor(triangle_brms[[i]], straight_brms[[i]])
}
names(bf_list) <- names(triangle_brms)

save(bf_list, file = "bf_list.RData")
```

```
load("bf_list.RData")

# Extract bf values
bf_vals <- list()

for (i in 1:length(bf_list)) {
 bf_vals[[i]] <- as.data.frame(bf_list[[i]]$bf)
}

names(bf_vals) <- names(bf_list)

# Convert list of bf's to df
bf_vals_df <- bind_rows(bf_vals, .id = "column_label")

# Count and summarize genes with support for
mean(bf_vals_df$`bf_list[[i]]$bf`) # 186.7349
```

```
## [1] 186.5799
```

```
sd(bf_vals_df$`bf_list[[i]]$bf`) # 512.1488
```

```
## [1] 510.8494
```

```
filt_bf <- filter(bf_vals_df, `bf_list[[i]]$bf` > 1)
nrow(filt_bf) # 51 or 77.27%
```

```
## [1] 51
```

Plot and export figures

```r
## Volcano plot of DM across gene bodies predicted by structural model
lm1_coefs_df <- as.data.frame(t(bind_rows(lm1_coefs, .id = "column_label")))
lm1_coefs_df$geneid <- rownames(lm1_coefs_df)
lm1_pvals_df <- as.data.frame(t(bind_rows(lm1_pvals, .id = "column_label")))
lm1_pvals_df$geneid <- rownames(lm1_pvals_df)

lm1_p_coef_df <- merge(lm1_coefs_df,
                       lm1_pvals_df,
                       by = "geneid")

lm1_p_coef_df$fdr <- p.adjust(lm1_p_coef_df$V1.y, method = "fdr")

# Plot volcano plot of maternal sig DM gene bodies
lm1_p_coef_df$sig <- ifelse(lm1_p_coef_df$fdr < 0.05, "Yes", "No")
count(lm1_p_coef_df$sig)
```

```
##      x freq
## 1   No 5834
## 2  Yes    7
## 3 <NA>    8
```

```r
mat_meth_volcano <- ggplot(data = lm1_p_coef_df,
                           aes(x = V1.x, y = -log(V1.y),
                               color = sig, size = sig)) +
  geom_point() +
  theme_classic(base_size = 20) +
  theme(legend.position = "none") +
  scale_color_manual(values = c("Black", "Red")) +
  scale_size_manual(values = c(1.5, 3)) +
  scale_y_continuous(limits = c(0 , 17)) +
  scale_x_continuous(limits = c(-55, 55)) +
  labs(x = "Differential methylation (slope)", y = "-log(p-value)")

# Export volcano plot
png( "med_meth_volcano.png", units = "in", width = 7,
      height = 7,
      res = 600 )

mat_meth_volcano
```

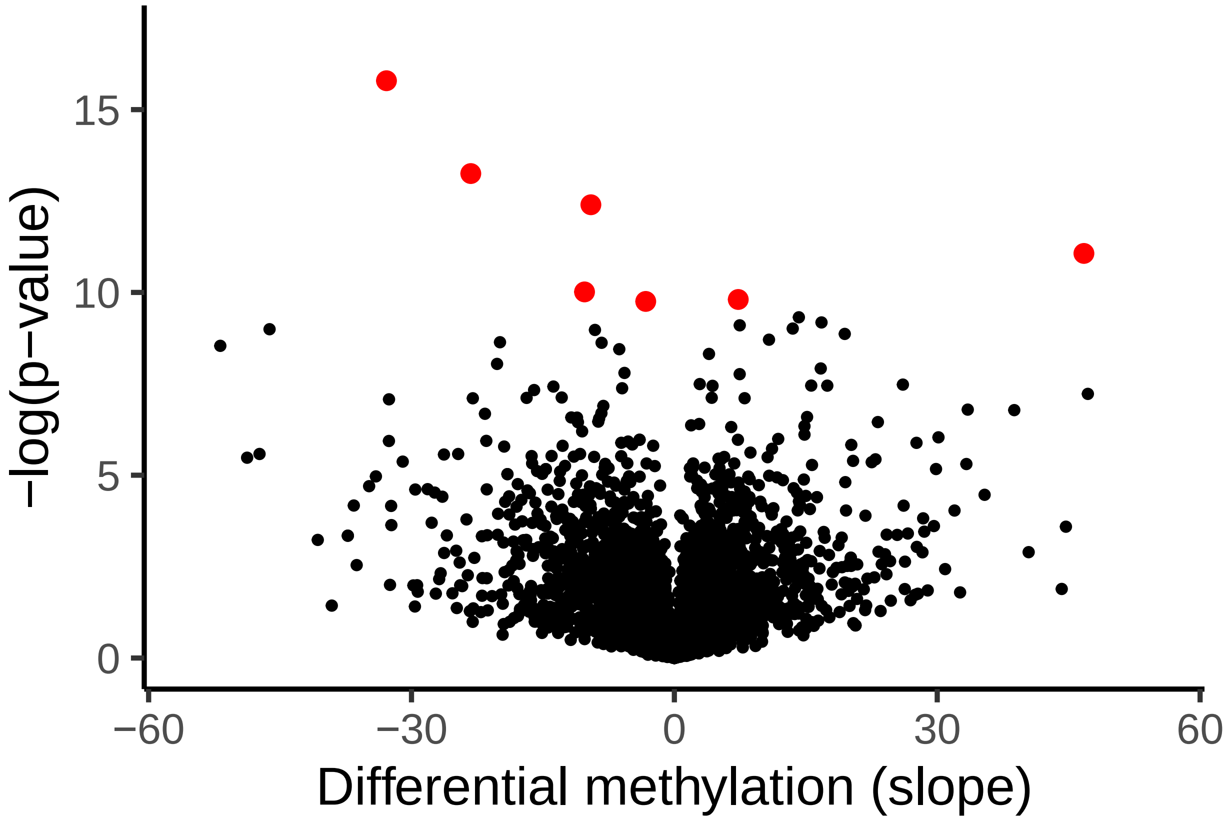
```
## Warning: Removed 8 rows containing missing values (geom_point).
```

```r
dev.off()
```

```
## pdf
##   2
```

```
mat_meth_volcano
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```



```
# Plot volcano of dev sig DM
lm1_coef_d <- list()

for (i in 1:length(lm1s)) {
 lm1_coef_d[[i]] <- as.data.frame(lm1s[[i]]$effects)[3,1]
}
names(lm1_coef_d) <- names(lm1s)

lm1_pval_d <- list()

# Report significant DM gene bodies

for (i in 1:length(lm1s)) {
 lm1_pval_d[[i]] <- as.data.frame(anova(lm1s[[i]]))[2,5]
}
```

```
## Warning in anova.lm(lm1s[[i]]): ANOVA F-tests on an essentially perfect fit are
## unreliable
```

```
## Warning in anova.lm(lm1s[[i]]): ANOVA F-tests on an essentially perfect fit are
## unreliable
```

```r
names(lm1_pval_d) <- names(lm1s)

as.data.frame(anova(lm2s[[1]]))[2,5]
```

```
## [1] 0.03407576
```

```r
lm1_coef_d_df <- as.data.frame(t(bind_rows(lm1_coef_d, .id = "column_label")))
lm1_coef_d_df$geneid <- rownames(lm1_coef_d_df)
lm1_pval_d_df <- as.data.frame(t(bind_rows(lm1_pval_d, .id = "column_label")))
lm1_pval_d_df$geneid <- rownames(lm1_pval_d_df)

lm1_p_coef_d_df <- merge(lm1_coef_d_df,
                       lm1_pval_d_df,
                       by = "geneid")

lm1_p_coef_d_df$fdr <- p.adjust(lm1_p_coef_d_df$V1.y, method = "fdr")

# Plot volcano plot of dev sig DM gene bodies
lm1_p_coef_d_df$sig <- ifelse(lm1_p_coef_d_df$fdr < 0.05, "Yes", "No")
count(lm1_p_coef_d_df$sig)
```

```
##      x freq
## 1   No 5841
## 2 <NA>    8
```

```r
dev_meth_volcano <- ggplot(data = lm1_p_coef_d_df,
                           aes(x = V1.x, y = -log(V1.y),
                               color = sig, size = sig)) +
  geom_point() +
  theme_classic(base_size = 20) +
  theme(legend.position = "none") +
  scale_color_manual(values = c("Black", "Red")) +
  scale_size_manual(values = c(1.5, 3)) +
  scale_y_continuous(limits = c(0 , 17)) +
  scale_x_continuous(limits = c(-55, 55)) +
  labs(x = "Differential methylation (slope)", y = "-log(p-value)")

# Export bf_dens plot
png( "dev_meth_volcano.png", units = "in", width = 7,
      height = 7,
      res = 600 )

dev_meth_volcano
```

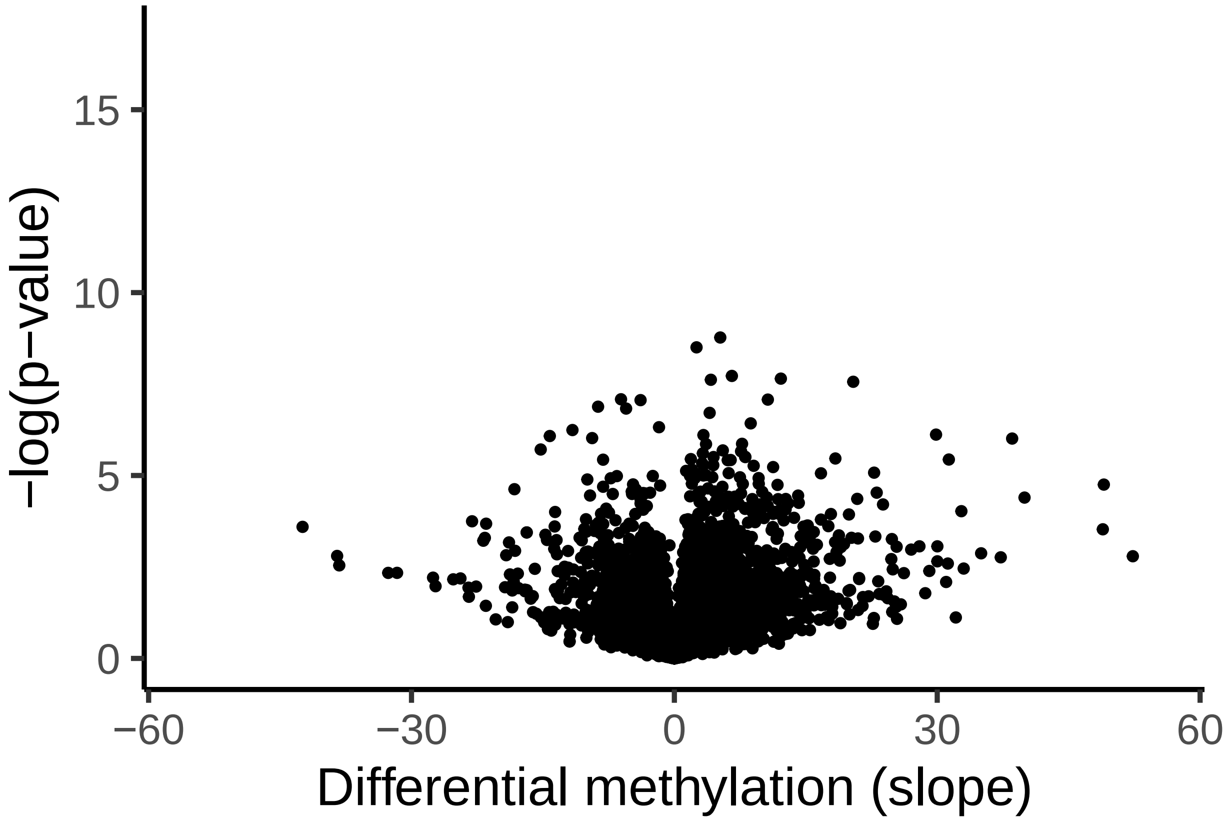
```
## Warning: Removed 9 rows containing missing values (geom_point).
```

```r
dev.off()
```

```
## pdf
##   2
```

```
dev_meth_volcano
```

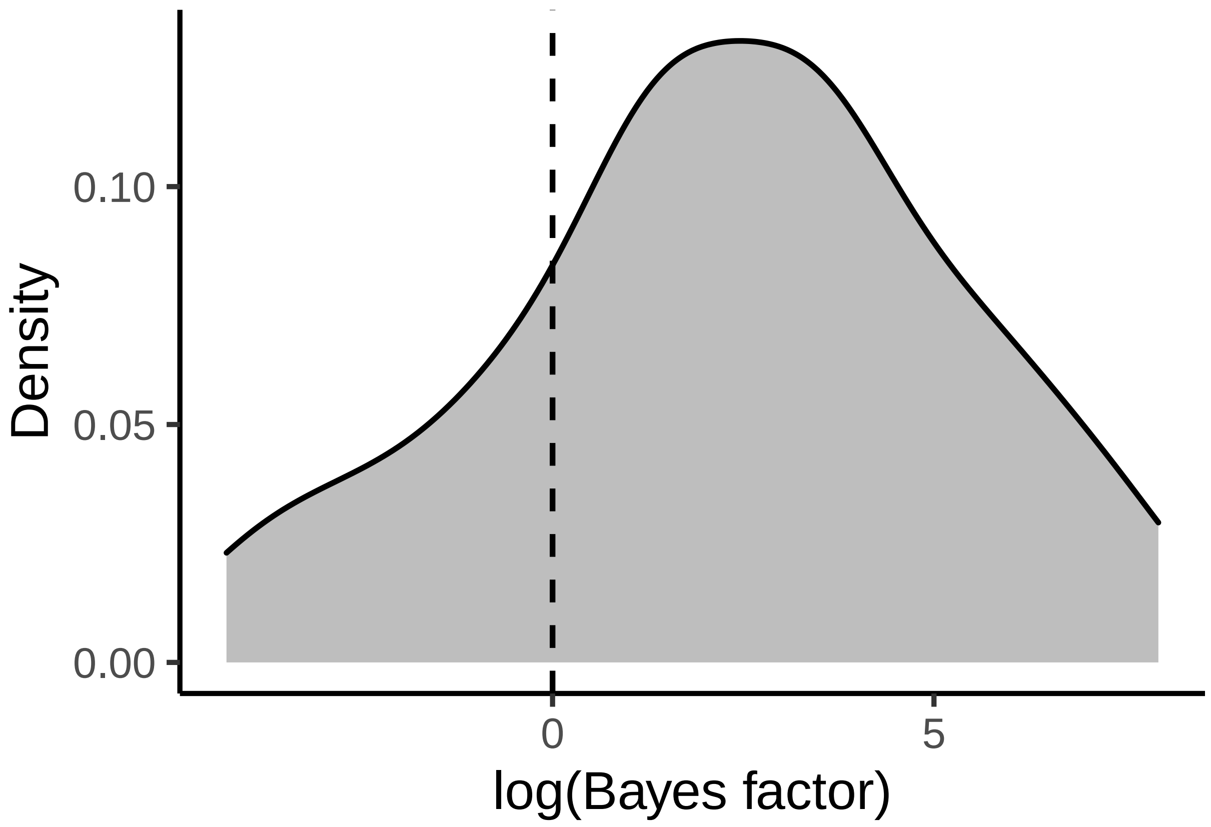
```
## Warning: Removed 9 rows containing missing values (geom_point).
```



```
# Plot distribution of bf's
bf_dens <- ggplot(data = bf_vals_df,
       aes(x = log(`bf_list[[i]]$bf`))) +
  geom_density(size = 1, fill = "grey") +
  theme_classic(base_size = 20) +
  geom_vline(xintercept = log(1), lty = 2, size = 1, color = "black") +
  labs( x = "log(Bayes factor)", y = "Density")

# Export bf_dens plot
png( "bf_dens_plot.png", units = "in", width = 7,
      height = 7,
      res = 600 )

bf_dens

dev.off()
```

```
## pdf
##   2
```

```
bf_dens
```



```
# 62% of mediated effects are in genes with positive correlations between expression and methylation
med_genes_plot <- ggplot(data = filter(tab_exp_df3, Group.1 %in% sig_dev_meth_dfm$column_label &
                          Group.1 %in% filt_bf$column_label),
       aes(y = value, x = all_gw_meth)) +
  geom_point(aes(color = Mat_treat)) +
  geom_smooth(method = "lm", se = TRUE) +
  theme_classic(base_size = 20) +
  facet_wrap(~Group.1, scale = "free", nrow = 5) +
  theme(strip.text = element_blank(),
        strip.background = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        legend.position = "none") +
  labs(y = "CPM", x = "% GBM", color = "Maternal treatment")

# Export bf_dens plot
png( "med_genes_plot.png", units = "in", width = 12,
      height = 7,
      res = 600 )

med_genes_plot
```

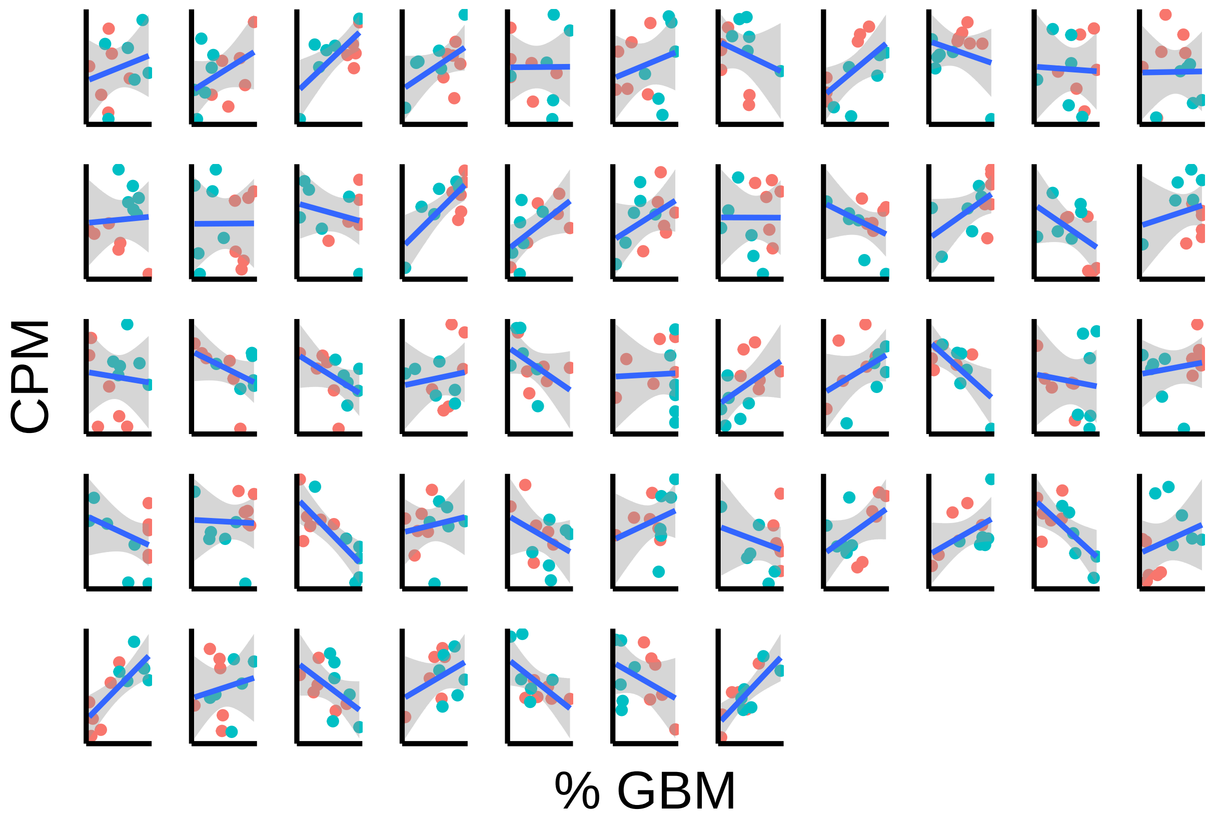
```
## `geom_smooth()` using formula 'y ~ x'
```

```
dev.off()
```

```
## pdf
##   2
```

```
med_genes_plot
```

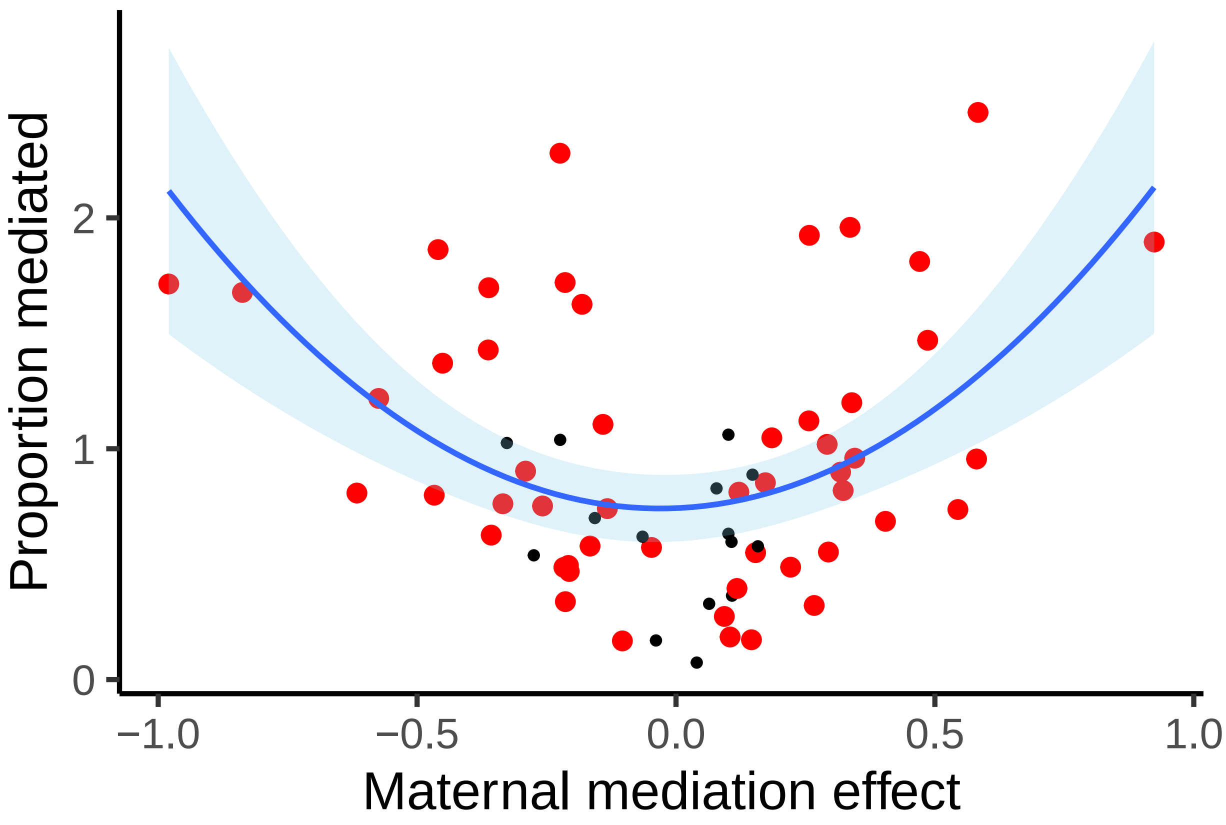
```
## `geom_smooth()` using formula 'y ~ x'
```



```
# Plot mediated effects against their proportion of total mediated
int_params_df$color <- ifelse(int_params_df$geneid %in% filt_bf$column_label, "Red", "Black")

med_effect_plot <-
  ggplot(data = filter(int_params_df, geneid %in% sig_dev_meth_dfm$column),
       aes(y = abs(V1.y), x = V1.x)) +
  #geom_errorbar(aes(xmin = X2.5..x, xmax = X97.5..x), width = 0, color = "lightgrey") +
  geom_point(aes(color = color, size = color)) +
  geom_smooth(method = "lm", se = TRUE, formula = y~poly(x,2), fill = "skyblue", alpha = 0.25) +
  theme_classic(base_size = 20) +
  theme(legend.position = "none") +
  scale_color_manual(values = c("Black", "Red")) +
  scale_size_manual(values = c(1.5, 3)) +
  labs(y = "Proportion mediated", x = "Maternal mediation effect")

# Export med_effect_plot
png( "med_effect_plot.png", units = "in", width = 7,
      height = 7,
      res = 600 )

med_effect_plot

dev.off()
```

```
## pdf
##   2
```

```
med_effect_plot
```


Proportion mediated
Maternal mediation effect