Epigenomics for Social Scientists

03 Single Site, Gene Ontology Association Analysis

Kelly Bakulski, Shan Andrews, John Dou, Jonah Fisher, Erin Ware

Last compiled on July 29, 2021

Setup

Load relevant packages

```
library(minfi)
library(limma)
library(matrixStats)
library(MASS)
library(sva)
library(Hmisc)
#library(missMethyl)
library(tidyverse)
library(qqman)
library(here)
library(ggsci)
library(ggpubr)
```

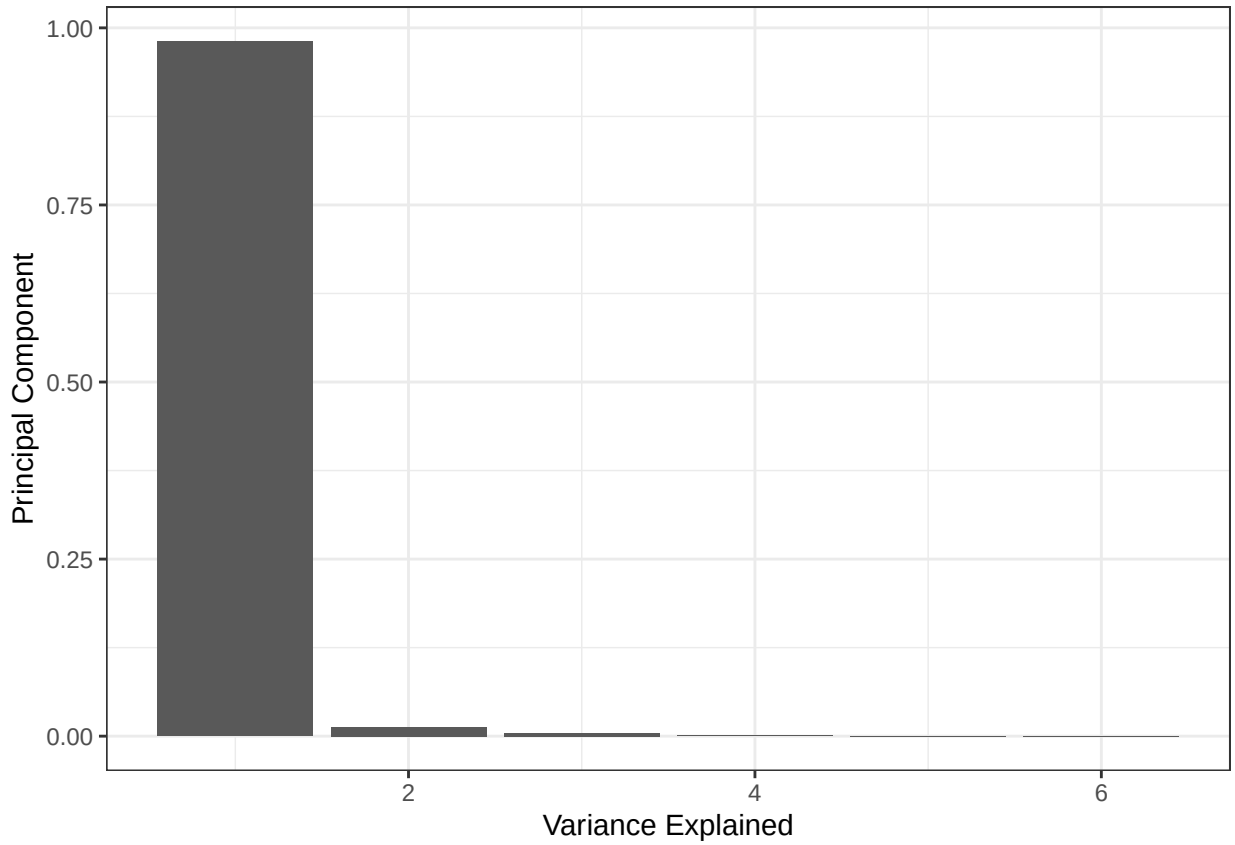
Read in the data

```
# Need noob, combat.beta, pd files
load(here("Data", "Premade_Intermediate_Files","combat-beta.rda"))
load(here("Data", "pdqc.rda"))

# Make sure this outputs to TRUE
all.equal.character(pd$Basename, colnames(combat.beta))
```

```
## [1] TRUE
```

Principal components on cell proportions

```
celltypes <- c("GR","NK","B","CD4","CD8","MO")
cellpcs <- prcomp(t(pd[celltypes]))
pcvar <- data.frame(pc = seq(1, length(cellpcs$sdev)), var = cellpcs$sdev^2 / sum(cellpcs$sdev^2))

ggplot(pcvar, aes(x = pc, y = var)) +
  geom_bar(stat = "identity") +
  labs(x = "Variance Explained", y = "Principal Component") +
  theme_bw()
```




```
summary(pd$GR)
```

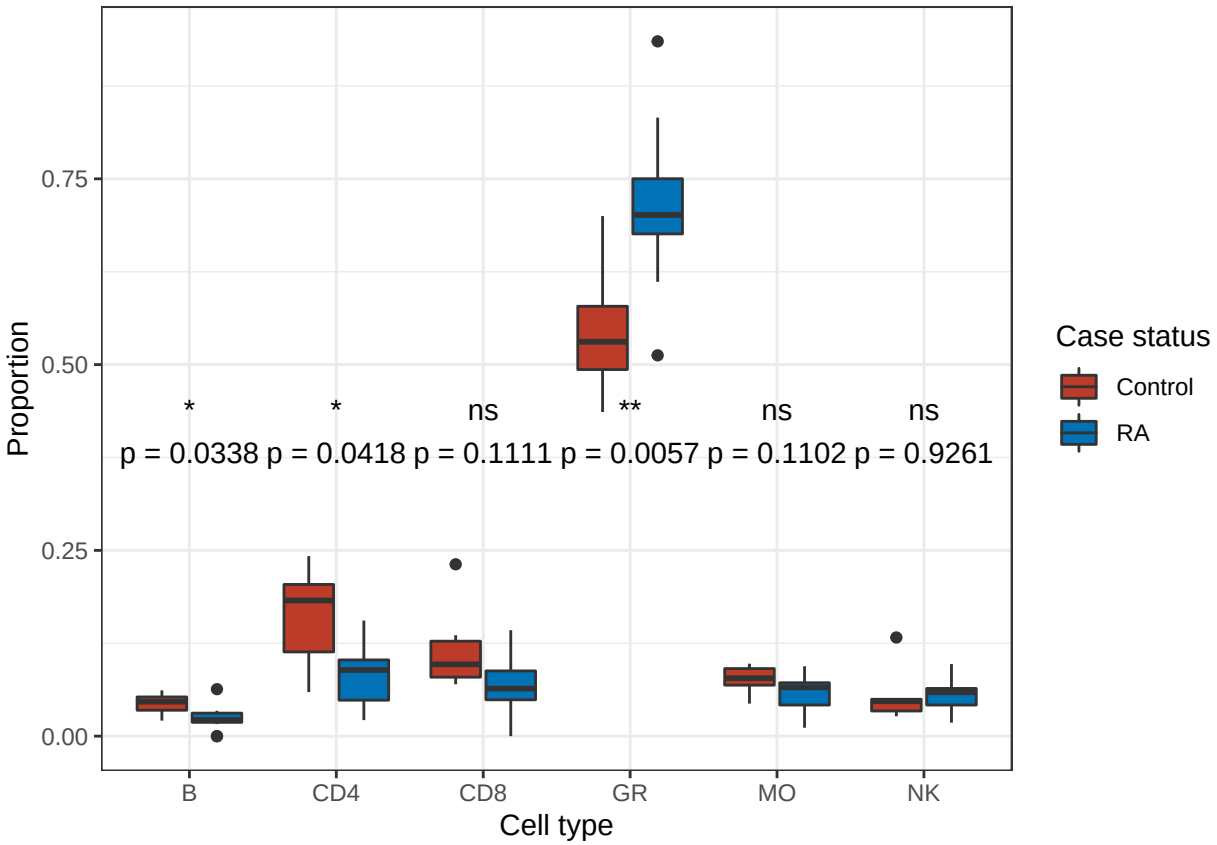
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4362  0.5352  0.6997  0.6543  0.7050  0.9350
```

Box Plots and t-statistics for each cell type

```
# Box Plots and t-statistics for each cell type
celldf <- pivot_longer(pd[c(celltypes, "casestatus")], cols = celltypes)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(celltypes)` instead of `celltypes` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
ggplot(celldf, aes(x = name, y = value, fill = casestatus)) +
  geom_boxplot() +
  labs(x = "Cell type", y = "Proportion", fill = "Case status") +
  theme_bw() +
  scale_fill_nejm() +
  stat_compare_means(label = c("p.format"), method = "t.test", vjust = 20) +
  stat_compare_means(label = c("p.signif"), method = "t.test", vjust = 18)
```



Single site association testing

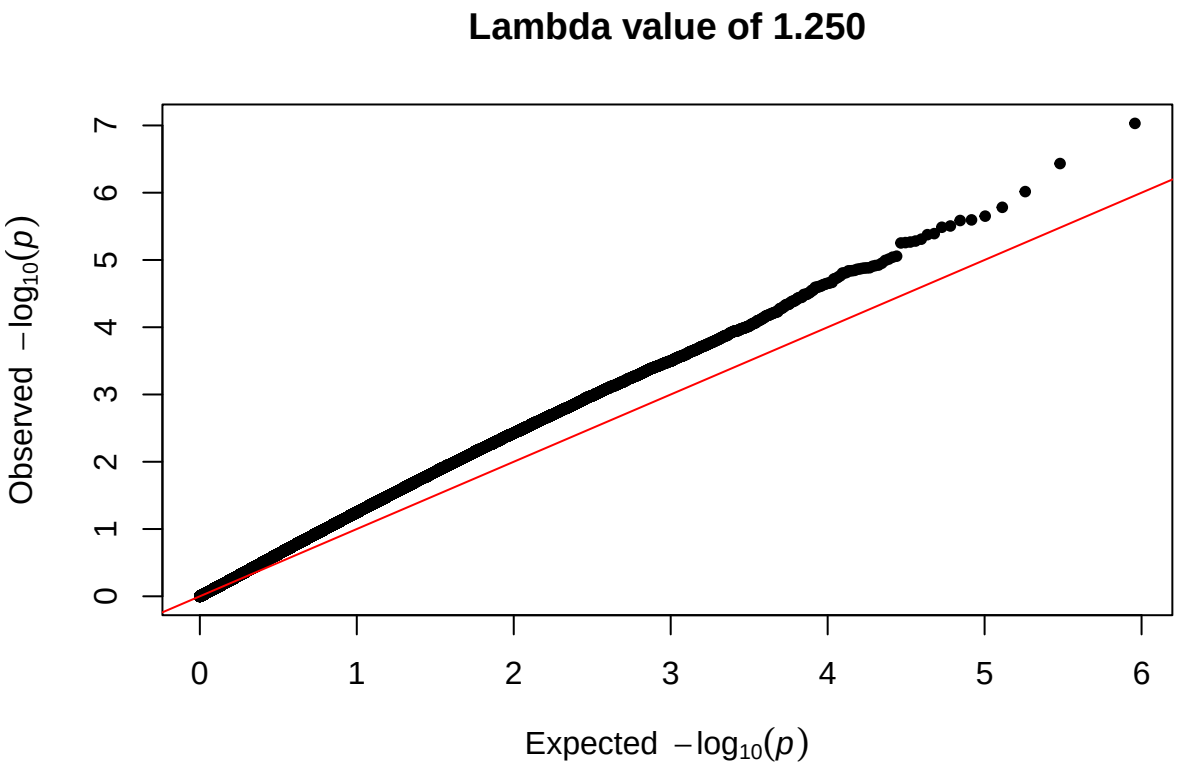
```
# Construct the model matrix
mod <- model.matrix(~ factor(pd$casestatus) + pd$age + factor(pd$sex) + factor(pd$smoking) + pd$GR)

# Run the single site association model
out <- lmFit(combat.beta, mod)
out <- eBayes(out)
ss.hits <- topTable(out, coef = 2, number = nrow(combat.beta))

head(ss.hits, n = 10)
```

```
##                  logFC   AveExpr         t      P.Value  adj.P.Val        B
## cg02330683  0.08771029 0.2052667 12.395356 9.324720e-08 0.04231576 8.339951
## cg05633597  0.14670774 0.7220319 10.821716 3.685954e-07 0.08363466 7.082819
## cg04156650 -0.07104587 0.8788333 -9.825937 9.626811e-07 0.14562220 6.178775
## cg19436320  0.13781428 0.1805937  9.301688 1.648447e-06 0.15848902 5.664274
## cg24805886  0.13123110 0.8468549  9.017146 2.229936e-06 0.15848902 5.372995
## cg09761230  0.05568162 0.1266593  8.897542 2.537587e-06 0.15848902 5.247932
## cg12623328 -0.06703143 0.2604477 -8.878366 2.591044e-06 0.15848902 5.227733
## cg14873818 -0.08612308 0.2720616 -8.707518 3.124757e-06 0.15848902 5.045951
## cg03285617  0.06573615 0.4689368  8.668116 3.264002e-06 0.15848902 5.003558
## cg01109337  0.15242356 0.3299824  8.472768 4.060770e-06 0.15848902 4.790739
```

```
rm(mod, out)
```

QQ plot with lambda statistic

```
observed <- -log10(sort(ss.hits$P.Value, decreasing = F))
expected <- -log10(ppoints(length(ss.hits$P.Value)))
lambda <- median(observed) / median(expected)

# Make a qq plot of our data
qq(ss.hits$P.Value, main = sprintf("Lambda value of %.3f", lambda))
```



Make a manhattan plot of our data

Manhattan plots place negative log10 transformed P values on the y axis. After the transformation, a higher value indicates greater significance. The X axis is ordered by chromosome number and position within the genome.

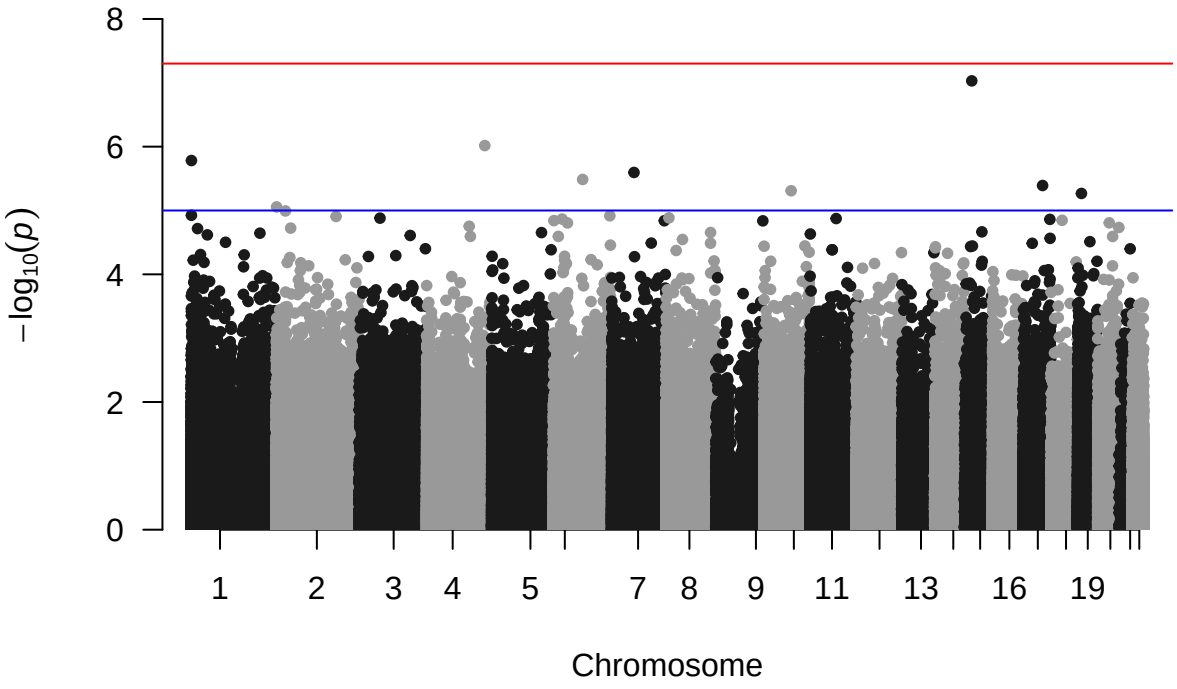
Manhattan plots are not quite as helpful with Epigenetics as they are with Genetics. This is partly due to how significant locations differ between DNA and DNAm. While genetic hits often cluster within GWAS, epigenetic hits from EWAS tend to not cluser together very much.

Nonetheless, it may still be useful to take a look at one.

```
anno <- IlluminaHumanMethylation450kanno.ilmn12.hg19::Locations
anno$chr <- str_split(anno$chr, "chr",n=2,simplify=T)[,2]
anno <- anno[anno$chr %in% 1:22,]
anno$chr <- as.numeric(anno$chr)
```

```
forman <- merge(anno[c("chr", "pos")], ss.hits[c("P.Value")], by = "row.names")
forman <- forman[complete.cases(forman),]
forman$SNP <- ""
```

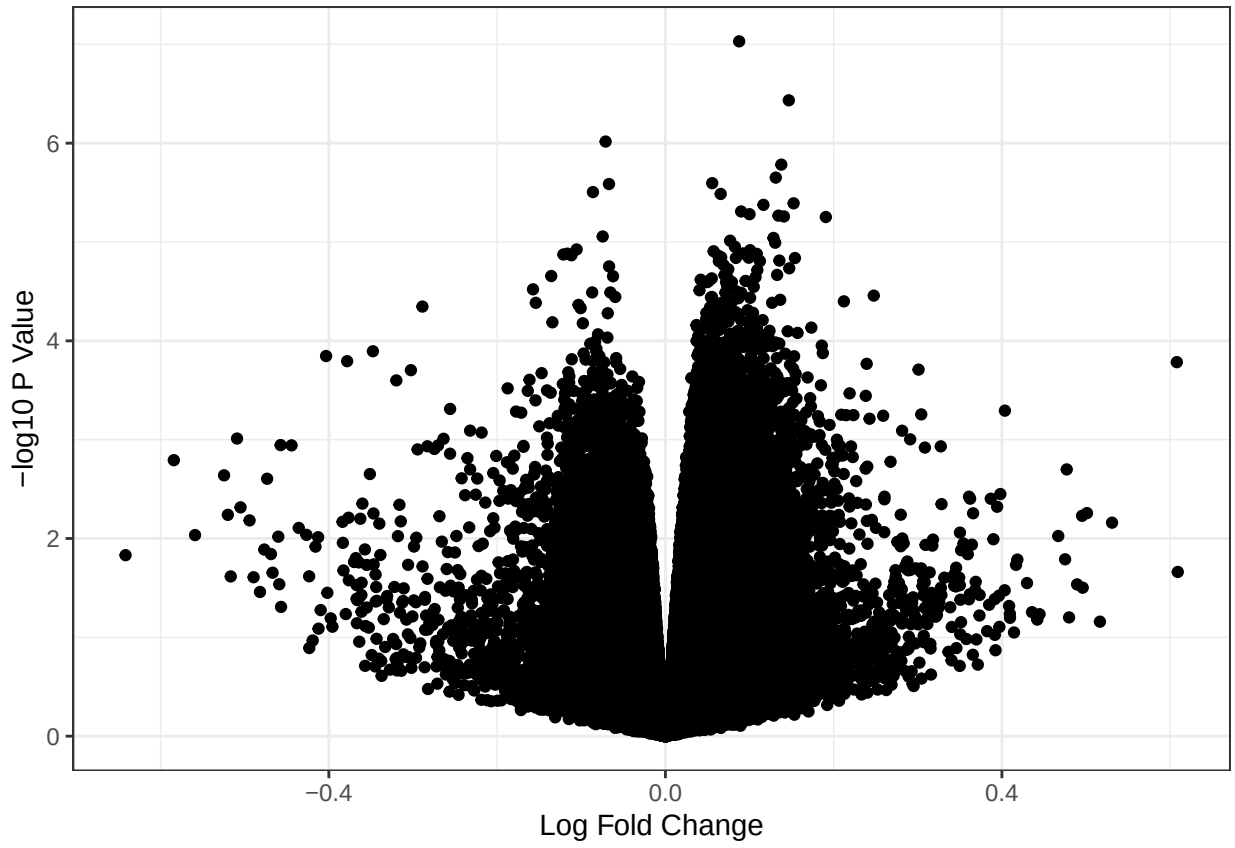
```
# Call function
qqman::manhattan(forman, chr = "chr", bp = "pos", p = "P.Value")
```



Volcano Plot

Volcano plots show effect size estimates (positive and negative) on the x axis against log transformed P values on the y axis. Typically the higher the absolute value of the effect size, the more significant the p value is. This results in a distribution of points that resembles a volcano.

```
ggplot(ss.hits, aes(x = logFC, y = -log10(P.Value))) +
  geom_point() +
  theme_bw() +
  labs(x = "Log Fold Change", y = "-log10 P Value")
```



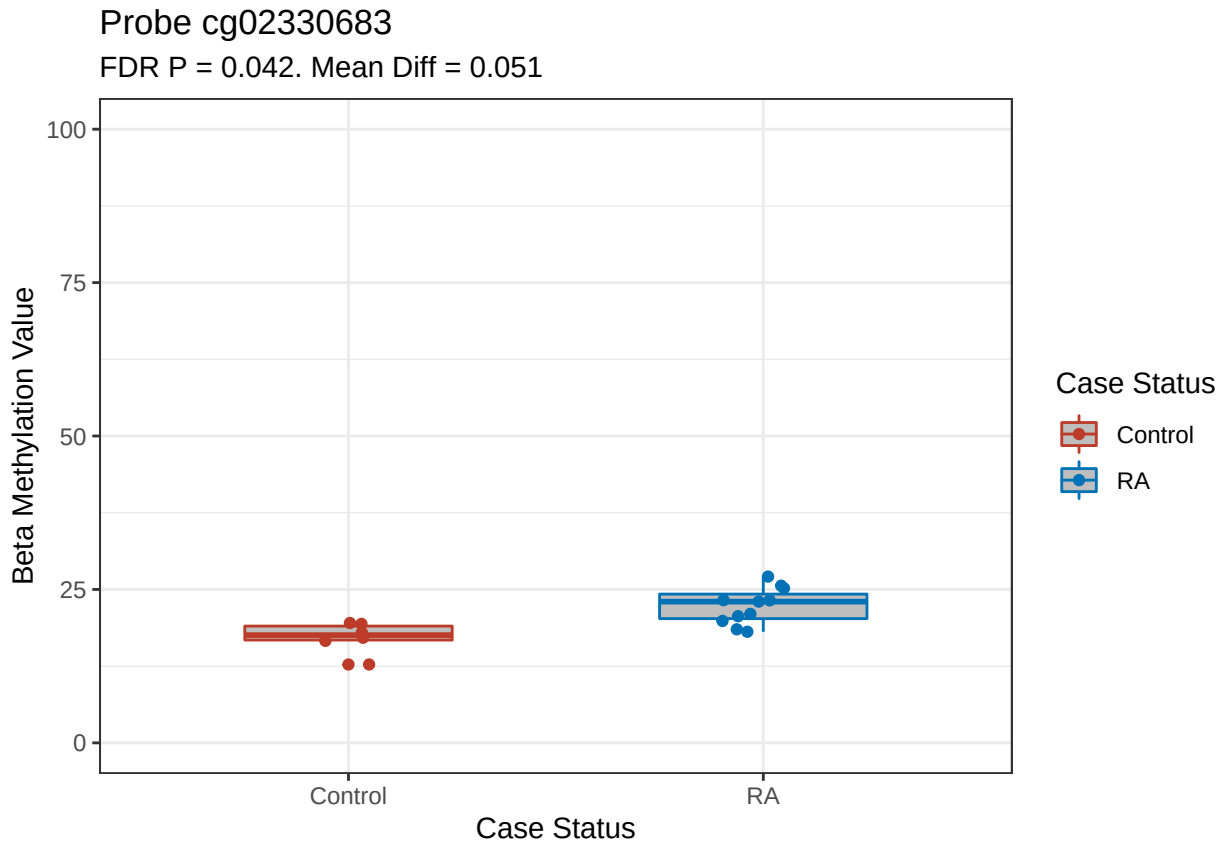
Plot the top 6 hits and examine case vs control beta differences

```
# Create a mini result list of the first 6 hits
tophits <- ss.hits[1:6, ]

# Function to make plot of one CpG
plotop <- function(hit){
  dat <- data.frame(vals = combat.beta[hit,], case = pd$casestatus)
  mdiff <- mean(dat$vals[dat$case == "RA"]) - mean(dat$vals[dat$case == "Control"])
  ggplot(dat, aes(x = case, y = vals, color = case)) +
    geom_boxplot(outliers = F, fill = "grey", width = 0.5) +
    geom_jitter(width = 0.1) +
    labs(x = "Case Status", y = "Beta Methylation Value", title = sprintf("Probe %s", hit),
         subtitle = sprintf("FDR P = %.3f. Mean Diff = %.3f", tophits$adj.P.Val[rownames(tophits) == hi
         color = "Case Status") +
    scale_color_nejm() +
    scale_y_continuous(limits = c(0, 1), breaks = seq(0, 1, by = 0.25), labels = c("0", "25", "50", "75
    theme_bw()
}

# Version without y-axis set to 0-100% methylation
# plotop <- function(hit){
#   dat <- data.frame(vals = combat.beta[hit,], case = pd$casestatus)
#   mdiff <- mean(dat$vals[dat$case == "RA"]) - mean(dat$vals[dat$case == "Control"])
#   ggplot(dat, aes(x = case, y = vals, color = case)) +
```

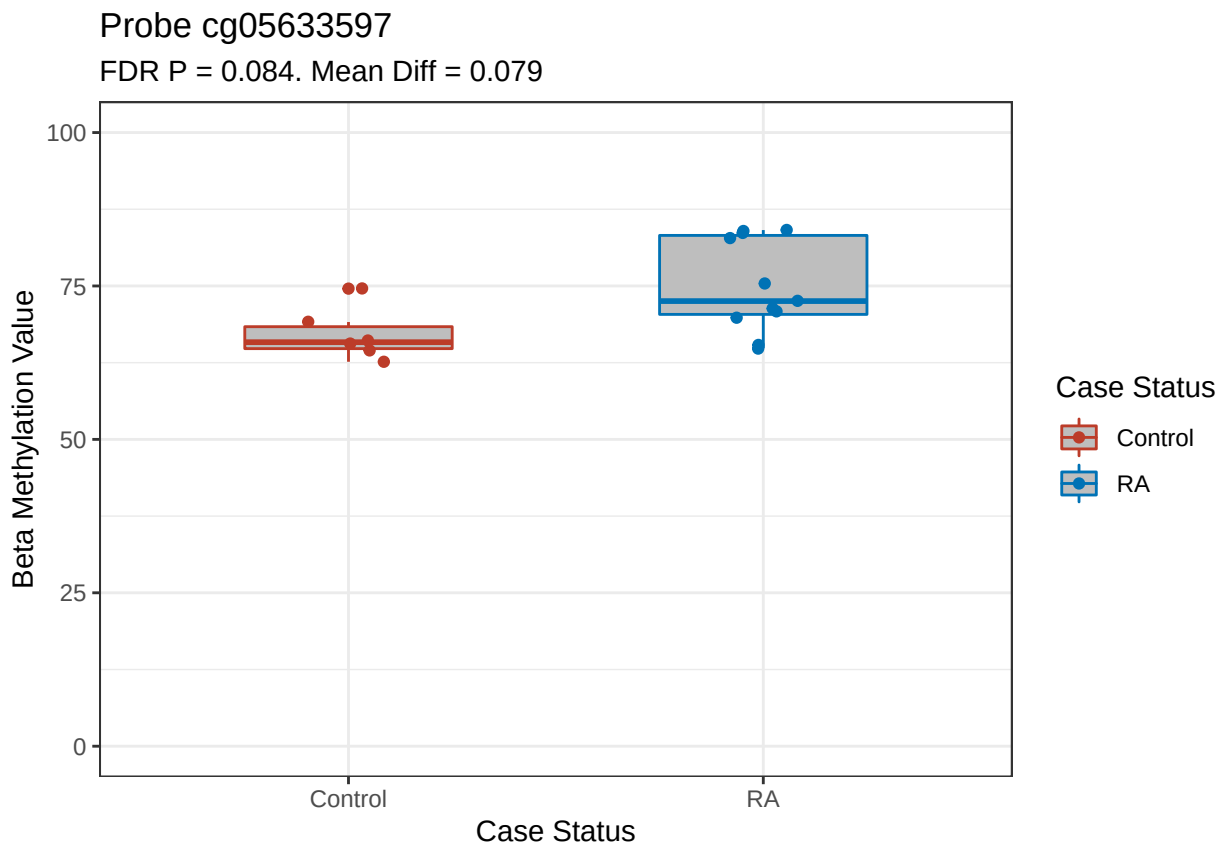
```
#     geom_boxplot(outliers = F, fill = "grey", width = 0.5) +
#     geom_jitter(width = 0.1) +
#     labs(x = "Case Status", y = "Beta Methylation Value", title = sprintf("Probe %s", hit),
#          subtitle = sprintf("FDR P = %.3f. Mean Diff = %.3f", tophits$adj.P.Val[rownames(tophits) == h
#          color = "Case Status") +
#     scale_color_nejm() +
#     theme_bw()
# }


# Loop through top six CpGs and use plot function
for(cpg in rownames(tophits)){
  tophit_plot <- plotop(cpg)
  print(tophit_plot)
}
```

```
## Warning: Ignoring unknown parameters: outliers
```

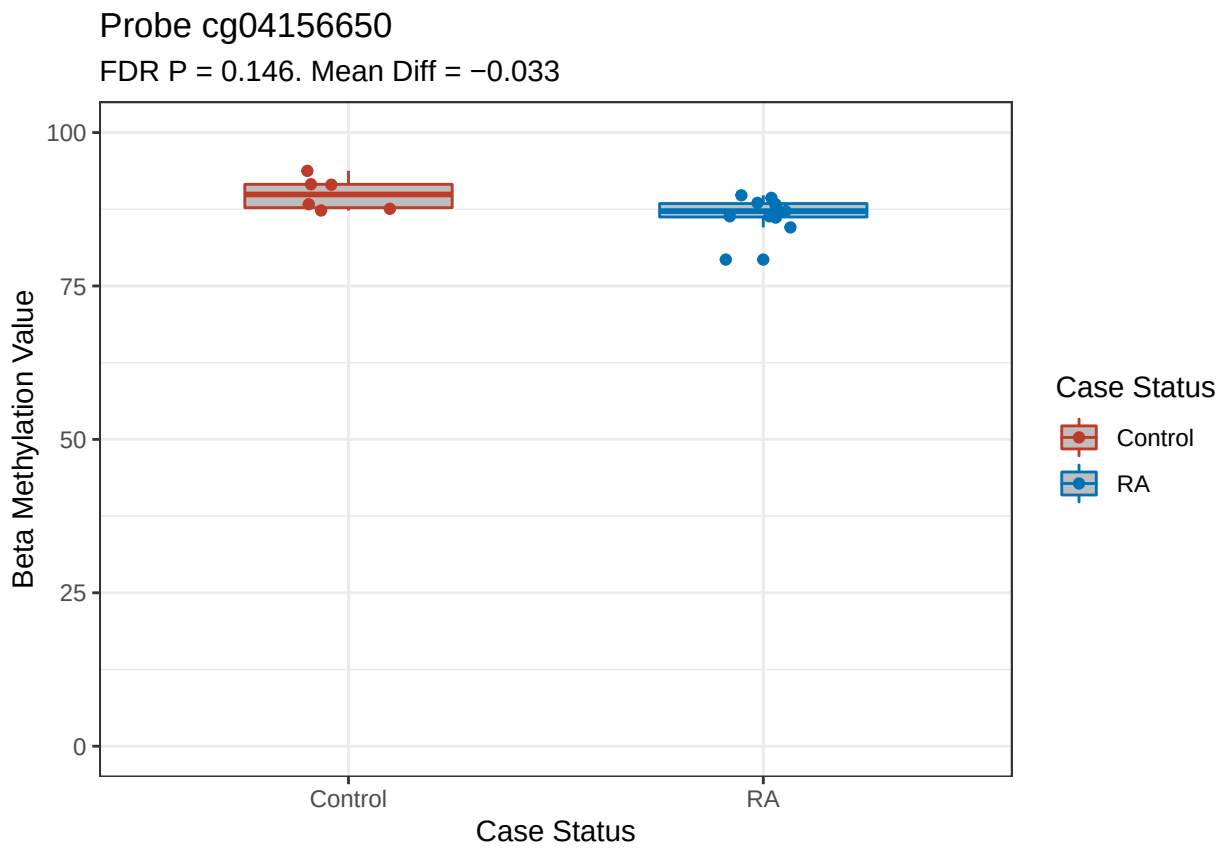
```
## Warning: Ignoring unknown parameters: outliers
```



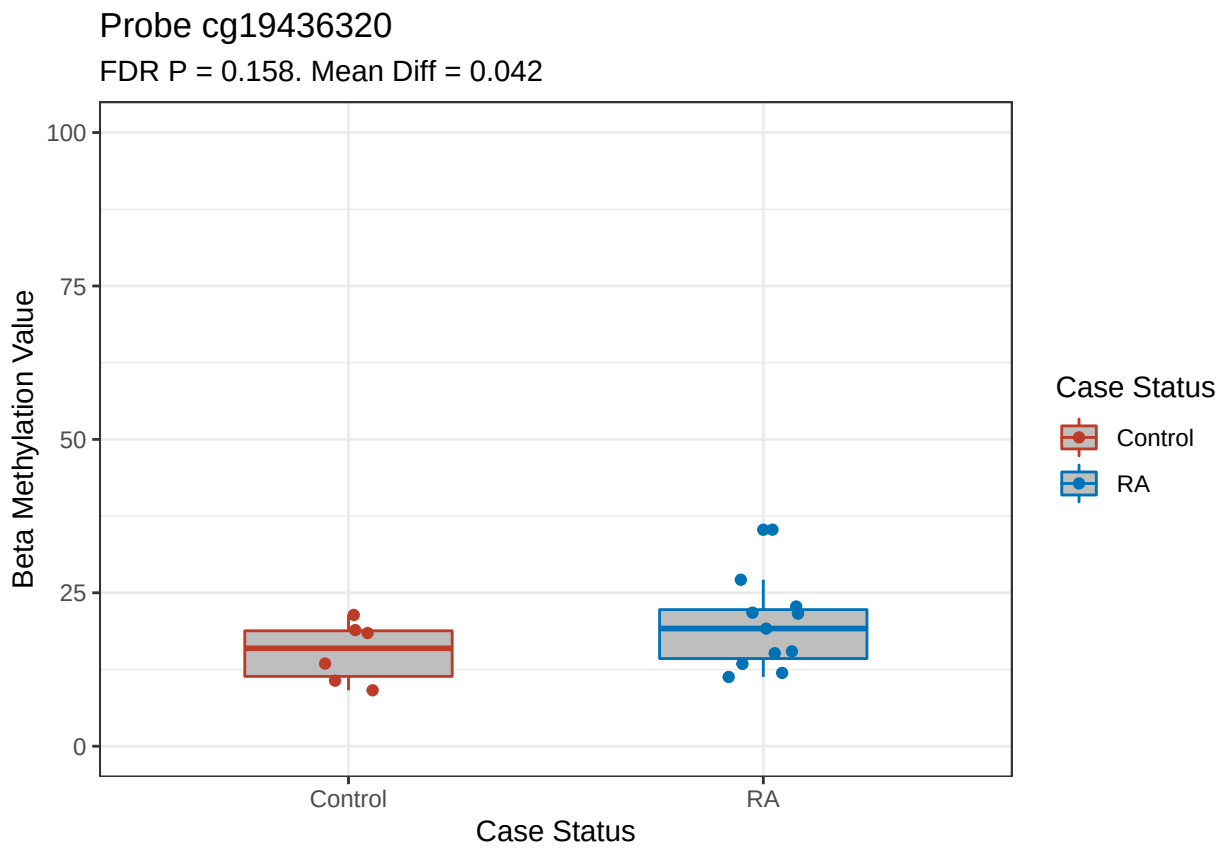

```
## Warning: Ignoring unknown parameters: outliers
```



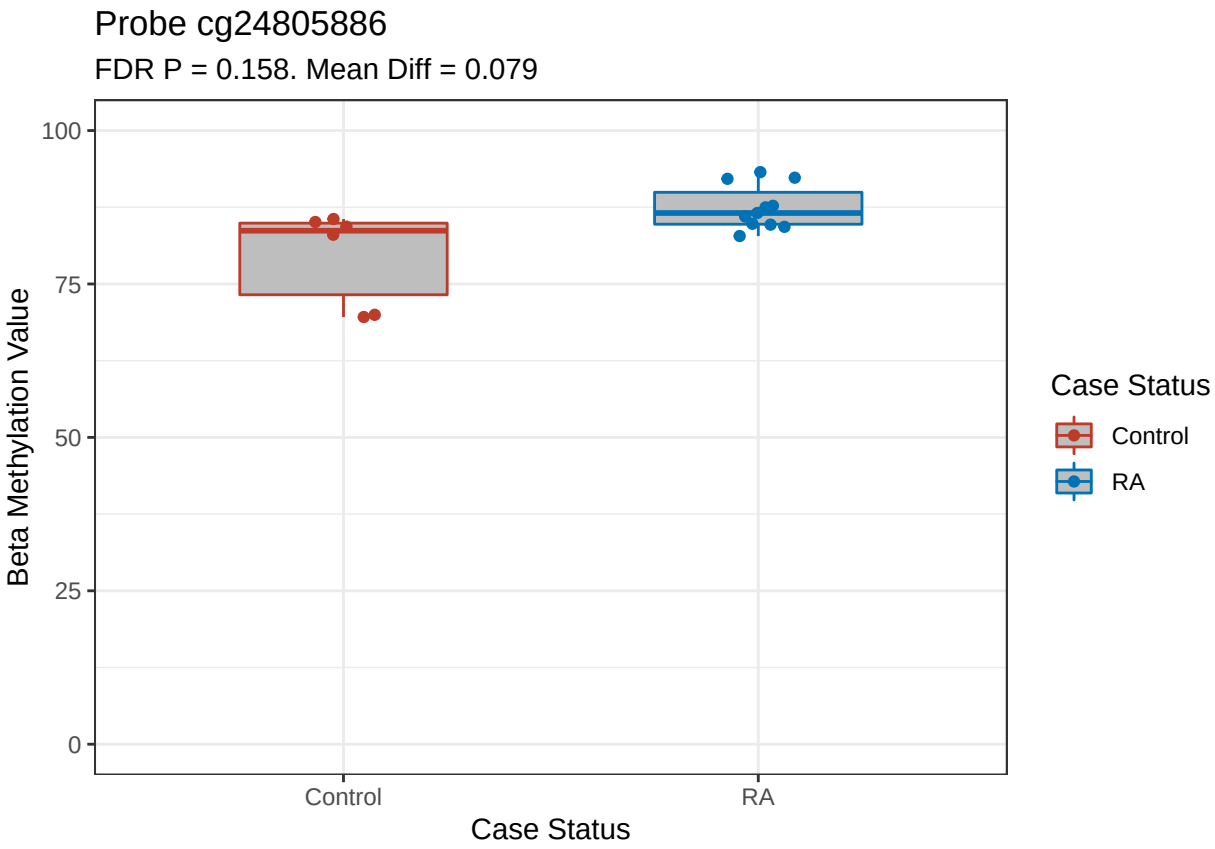

```
## Warning: Ignoring unknown parameters: outliers
```

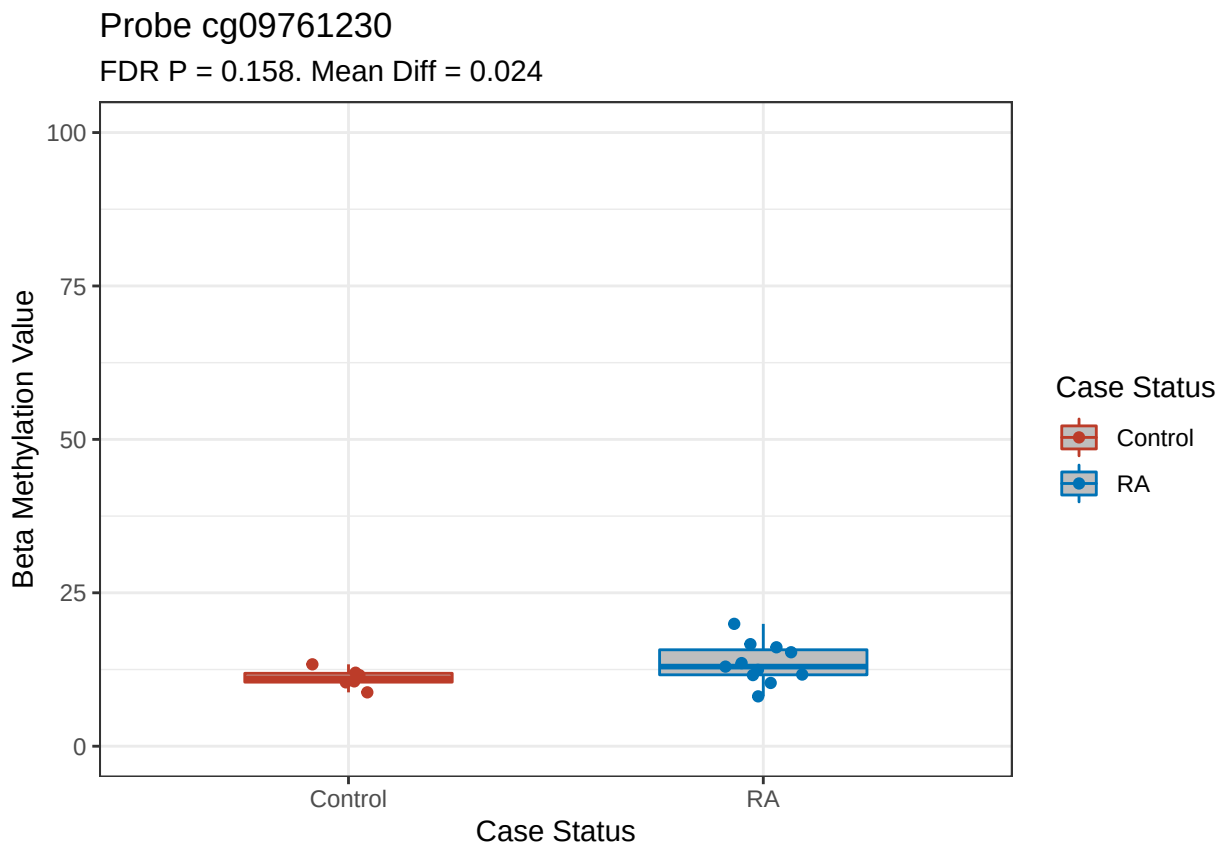



```
## Warning: Ignoring unknown parameters: outliers
```




```
## Warning: Ignoring unknown parameters: outliers
```


Probe cg24805886
FDR P = 0.158. Mean Diff = 0.079
Beta Methylation Value
100
75
50
25
0
Control
RA
Case Status
Case Status
Control
RA




```
rm(formanhattan)
```

```
## Warning in rm(formanhattan): object 'formanhattan' not found
```

Gene ontology analysis

```
# Need to pick a significance cutoff for inclusion. Here is 1x10-5, but may need to be flexible with da
#gene.ontology <- missMethyl::gometh(as.character(rownames(ss.hits)[ss.hits$P.Value < 1e-5]), all.cpg =
#save(gene.ontology, file = here("Data", "Premade_Intermediate_Files", "gene-ontology.rda"))
# memory requirements of missMethyl package may be too much for RStudio cloud, load the premade gene.on

load(here("Data", "Premade_Intermediate_Files","gene-ontology.rda"))

dim(gene.ontology)
```

```
## [1] 22716     6
```

```
summary(gene.ontology$P.DE)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0004966 1.0000000 1.0000000 0.9754508 1.0000000 1.0000000
```

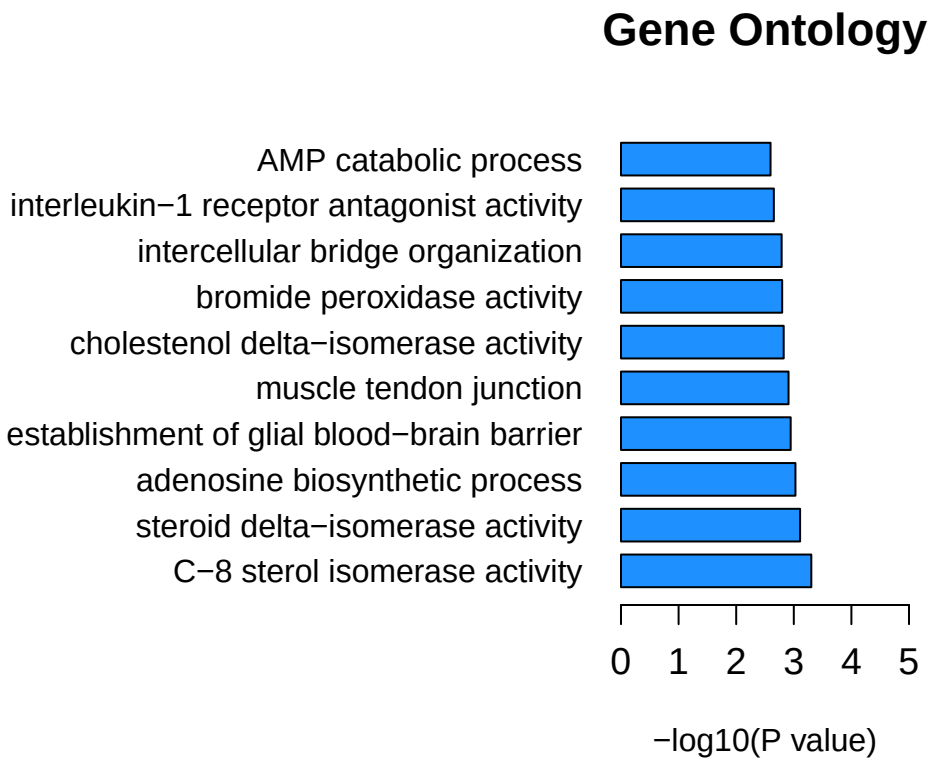
```
summary(gene.ontology$FDR)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
```

```r
gene.ontology <- gene.ontology[order(gene.ontology$P.DE), ]
head(gene.ontology)
```

```
##            ONTOLOGY                                        TERM N DE
## GO:0000247       MF              C-8 sterol isomerase activity 1  1
## GO:0004769       MF           steroid delta-isomerase activity 4  1
## GO:0046086       BP             adenosine biosynthetic process 1  1
## GO:0060857       BP establishment of glial blood-brain barrier 1  1
## GO:0005927       CC                      muscle tendon junction 2  1
## GO:0047750       MF       cholestenol delta-isomerase activity 2  1
##                      P.DE FDR
## GO:0000247 0.0004966248   1
## GO:0004769 0.0007773255   1
## GO:0046086 0.0009363475   1
## GO:0060857 0.0011323011   1
## GO:0005927 0.0012338191   1
## GO:0047750 0.0014980539   1
```

```r
# Some functions that will be useful in making the plot
wrap.it <- function(x, len) {
  sapply(x, function(y) {
    paste(strwrap(y, len),
      collapse = "\n"
    )
  },
  USE.NAMES = FALSE
  )
}
# Call this function with a list or vector
wrap.labels <- function(x, len) {
  if (is.list(x)) {
    lapply(x, wrap.it, len)
  } else {
    wrap.it(x, len)
  }
}

par(mai = c(1, 4, 1, 1))
barplot(abs(log(as.numeric(gene.ontology$P.DE[1:10]), base = 10)),
  main = "Gene Ontology", horiz = TRUE, names.arg = wrap.labels(gene.ontology$TERM[1:10], 50),
  xlab = "-log10(P value)", col = "dodgerblue", las = 1, cex.axis = 1.2, cex.main = 1.4, cex.lab = 1, c
)
```


Gene Ontology
AMP catabolic process
interleukin−1 receptor antagonist activity
intercellular bridge organization
bromide peroxidase activity
cholestenol delta−isomerase activity
muscle tendon junction
establishment of glial blood−brain barrier
adenosine biosynthetic process
steroid delta−isomerase activity
C−8 sterol isomerase activity
0
1
2
3
4
5
−log10(P value)