Cell-type specific enrichment analysis

Dexamethasone-Stimulated Human Array Project

```
library(data.table)
library(dplyr)
library(ggplot2)
library(corrplot)
library(factoextra)
library(viridis)
library(ggpubr)
library(UpSetR)
```

```
cbPalette <- c( "#0072B2", "#009E73", "#E69F00", "#F0E442", "#D55E00", "#CC79A7", "#56B4E9", "#999999")
separate <- ""
```

```
treatment   <- "veh"

pval.veh.df <- fread(paste0("~/bio/code/mpip/dex-stim-human-array/output/data/integrative/cell_type_enri

pval.veh.bcc.df <- pval.veh.df[, 1:13]
```

```
fdr.bcc.df <- matrix(p.adjust(as.vector(as.matrix(pval.veh.bcc.df[, 2:13])), method='fdr'),
                     ncol=12) %>%
  data.frame()

fdr.bcc.df <- cbind(pval.veh.bcc.df$CpG_ID, fdr.bcc.df)
colnames(fdr.bcc.df) <- colnames(pval.veh.bcc.df)
```

```
sign.pval.df <- fdr.bcc.df %>% reshape2::melt(measure.vars = colnames(fdr.bcc.df)[2:13]) %>% setDT()
colnames(sign.pval.df) <- c("CpG_ID", "Type", "fdr")

sign.pval.df <- sign.pval.df[fdr <= 0.05]
sign.pval.df <- na.omit(sign.pval.df)
sign.pval.df
```

```
##             CpG_ID Type         fdr
##      1: cg16535257  Bas 0.024366508
##      2: cg13938959  Bas 0.011324577
##      3: cg06931612  Bas 0.031029976
##      4: cg00582671  Bas 0.010838442
##      5: cg11143486  Bas 0.028726078
##     ---
## 734223: cg08425796 Treg 0.003233155
## 734224: cg14496081 Treg 0.035331162
## 734225: cg01370805 Treg 0.003505756
## 734226: cg24849633 Treg 0.035063650
## 734227: cg12502079 Treg 0.005416987
```

```
veh.sign.pval.df <- sign.pval.df
veh.sign.pval.df[["Treatment"]] <- treatment
```

DEX

```
treatment   <- "dex"

pval.dex.df <- fread(paste0("~/bio/code/mpip/dex-stim-human-array/output/data/integrative/cell_type_enr

pval.dex.bcc.df <- pval.dex.df[, 1:13]
```

```
fdr.bcc.df <- matrix(p.adjust(as.vector(as.matrix(pval.dex.bcc.df[, 2:13])), method ='fdr'),
                     ncol=12) %>%
  data.frame()

fdr.bcc.df <- cbind(pval.dex.bcc.df$CpG_ID, fdr.bcc.df)
colnames(fdr.bcc.df) <- colnames(pval.dex.bcc.df)
```

```
sign.pval.df <- fdr.bcc.df %>% reshape2::melt(measure.vars = colnames(fdr.bcc.df)[2:13]) %>% setDT()
colnames(sign.pval.df) <- c("CpG_ID", "Type", "fdr")
sign.pval.df <- sign.pval.df[fdr <= 0.05]

sign.pval.df <- na.omit(sign.pval.df)
sign.pval.df
```

```
##              CpG_ID Type         fdr
##      1: cg21038584  Bas 0.036610119
##      2: cg07211239  Bas 0.038491071
##      3: cg04260676  Bas 0.002423863
##      4: cg02050917  Bas 0.039608066
##      5: cg26785499  Bas 0.024939802
##     ---
## 200548: cg00392007 Treg 0.048659128
## 200549: cg06665773 Treg 0.049944129
## 200550: cg01280327 Treg 0.022621541
## 200551: cg24322531 Treg 0.014209176
## 200552: cg01548456 Treg 0.049194224
```

```
dex.sign.pval.df <- sign.pval.df
dex.sign.pval.df[["Treatment"]] <- treatment
```

Distribution plots for Basleine and Dex together

```
sign.pval.df <- rbind(veh.sign.pval.df, dex.sign.pval.df)
intersect.cpgs <- intersect(dex.sign.pval.df$CpG_ID, veh.sign.pval.df$CpG_ID)

print(paste0("Number of unique dex CpGs: ", length(unique(dex.sign.pval.df$CpG_ID))))
```

```
## [1] "Number of unique dex CpGs: 118284"
```

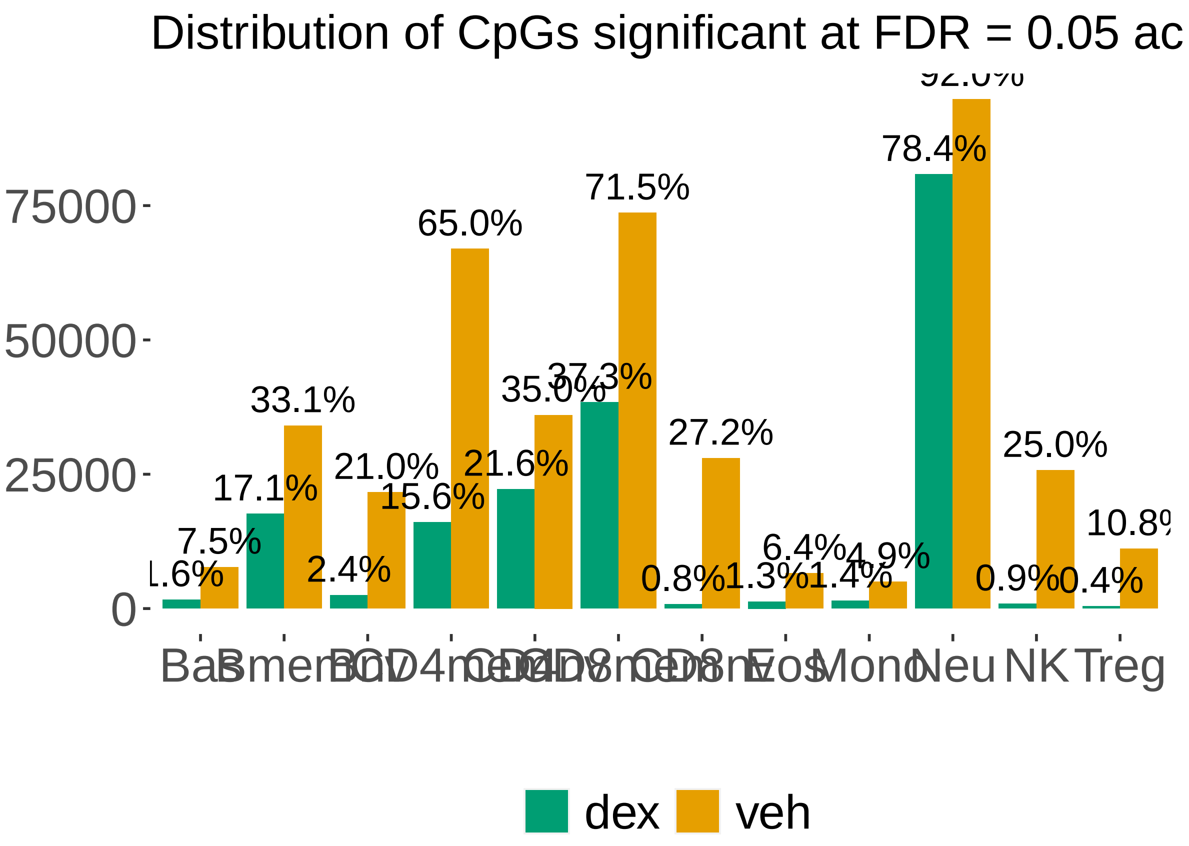
```
print(paste0("Number of unique baseline CpGs: ", length(unique(veh.sign.pval.df$CpG_ID))))
```

```
## [1] "Number of unique baseline CpGs: 289966"
```

```
print(paste0("Number of intesecting CpGs: ", length(unique(intersect.cpgs))))
```

```
## [1] "Number of intesecting CpGs: 103033"
```

```
ggplot(sign.pval.df[CpG_ID %in% intersect.cpgs], aes(x = Type, fill = Treatment)) +
  geom_bar(stat = "count", alpha = 1, position = position_dodge(width = .9)) +
  # geom_text(aes(label =  scales::percent(prop.table(stat(count)))),
  # geom_text(aes(label = scales::percent(stat(count) / length(unique(sign.pval.df$CpG_ID)))),
  geom_text(aes(label = scales::percent(stat(count) / length(intersect.cpgs), accuracy = 0.1)),
            stat = "count", vjust = -0.5, size = 5, position = position_dodge(width = .9)) +
  theme(legend.position = "bottom", # c(0.9, 0.9),
        legend.title = element_blank(),
        legend.text = element_text(size = 18),
        panel.grid.major = element_blank(),
        panel.background = element_blank(),
        plot.title = element_text(size = 18),
        axis.title = element_text(size = 18),
        axis.text = element_text(size = 18)) +
  labs(title = "Distribution of CpGs significant at FDR = 0.05 across 12 blood cell types",
       x = "", y = "") +
  scale_fill_manual(values = cbPalette[2:3])
```



Cell-type specificity on GR-induced (delta)-meQTLs

```
ind.meqtl.delta.df      <- fread("~/bio/code/mpip/dex-stim-human-array/output/data/integrative/matrixEQ
delta.meqtl.cpgs        <- ind.meqtl.delta.df$gene %>% unique()
sign.pval.delta.mqtl.df <- sign.pval.df[CpG_ID %in% delta.meqtl.cpgs, ]

sign.pval.cpgs.dex <- sign.pval.delta.mqtl.df[Treatment == "dex", CpG_ID]
sign.pval.cpgs.veh <- sign.pval.delta.mqtl.df[Treatment == "veh", CpG_ID]

intersect.cpgs          <- intersect(sign.pval.cpgs.dex, sign.pval.cpgs.veh)
sign.pval.delta.mqtl.df <- sign.pval.delta.mqtl.df[CpG_ID %in% intersect.cpgs,]

print(paste0("Number of unique dex CpGs: ", length(unique(sign.pval.cpgs.dex))))
```

```
## [1] "Number of unique dex CpGs: 678"
```

```
print(paste0("Number of unique baseline CpGs: ", length(unique(sign.pval.cpgs.veh))))
```

```
## [1] "Number of unique baseline CpGs: 1663"
```

```
print(paste0("Number of intesecting CpGs: ", length(unique(intersect.cpgs))))
```

```
## [1] "Number of intesecting CpGs: 653"
```

Check if changes are significant

Enrichment GR-meQTL CpGs over all CpGs

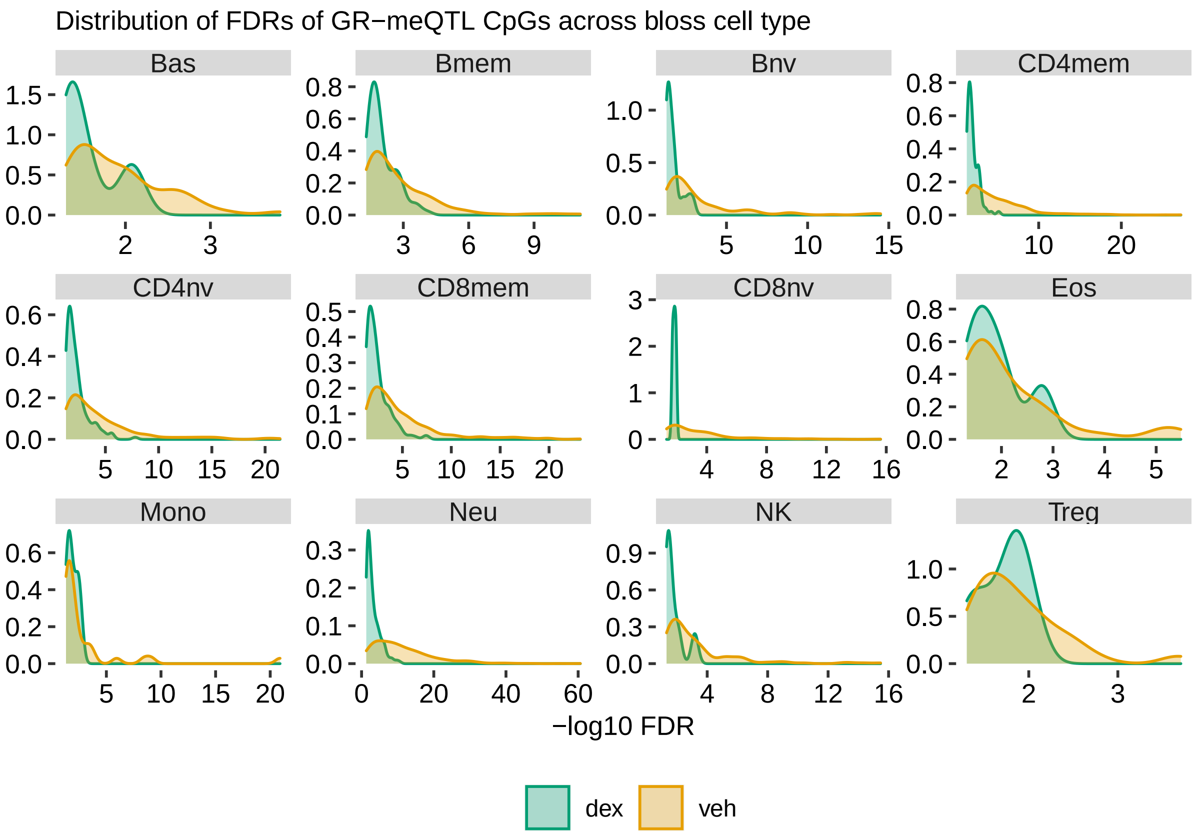
Distribution of P-values for each blood cell-type

Kolmogorov-Smirnov test: Do 2 samples follow the same distribution?

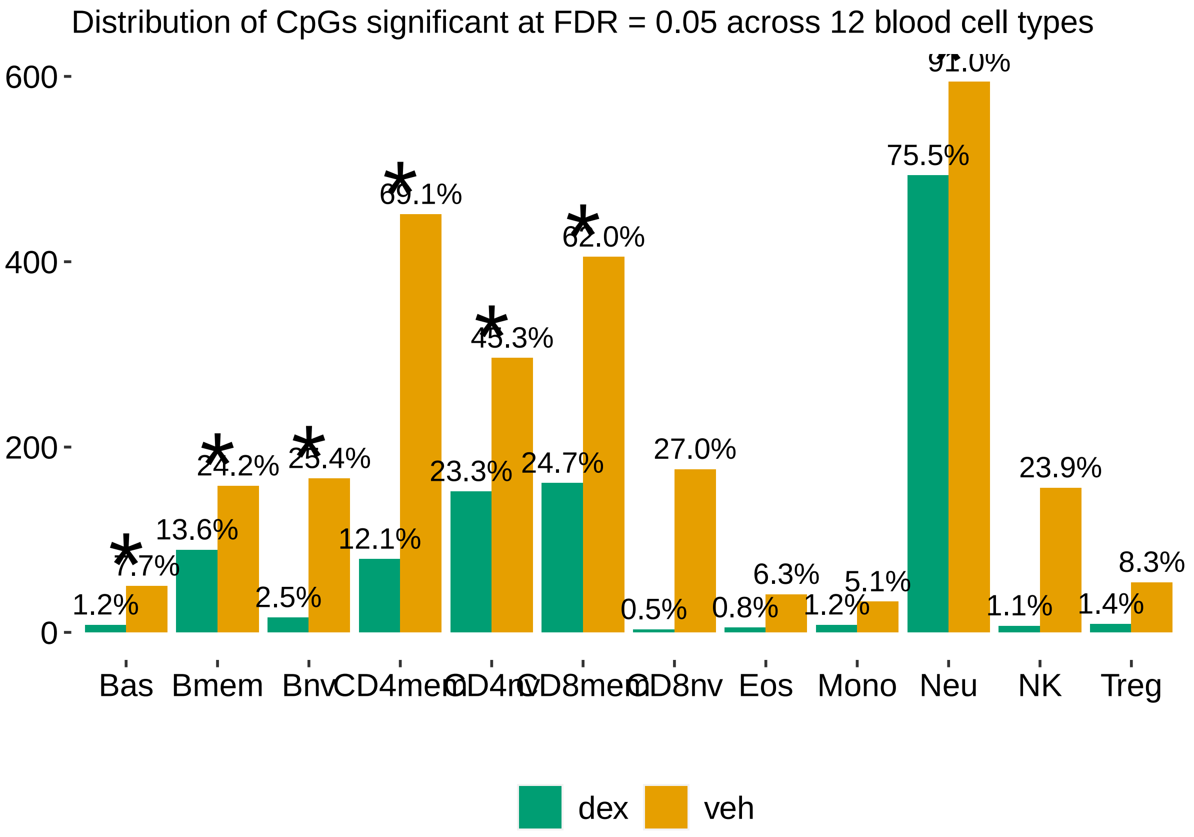
```
blood.cell.types        <- unique(sign.pval.delta.mqtl.df$Type)
# blood.cell.types        <- levels(sign.pval.delta.mqtl.df$Type)

test.res.df <- lapply(blood.cell.types, function(i){  # for each blood cell type
  dex.fdr.lst <- sign.pval.delta.mqtl.df[Type == i][Treatment == "dex", fdr]
  veh.fdr.lst <- sign.pval.delta.mqtl.df[Type == i][Treatment == "veh", fdr]
  test.rslt   <- ks.test(dex.fdr.lst, veh.fdr.lst)

  return(data.frame(Type = i, "p-value" = test.rslt$p.value))
  }) %>%
  bind_rows()

sign.pval.delta.mqtl.df <- left_join(sign.pval.delta.mqtl.df, test.res.df)
```

```
## Joining, by = "Type"
```

```
ggplot(sign.pval.delta.mqtl.df, aes(x = -log10(fdr), fill = Treatment)) +
  geom_density(alpha = 0.3, aes(color = Treatment)) +
  geom_text(data = subset(unique(sign.pval.delta.mqtl.df[, .(Treatment, Type, p.value)])),
            aes(x = -Inf, y = -Inf, label = paste0("KS test = ", signif(p.value, 3))),
            hjust = -0.5, vjust = -15, size = 5, color = "black" ) +
  facet_wrap(~ Type, scales = "free", ncol = 4) +
  theme(legend.position = "bottom", # c(.9,.9),
        legend.title = element_blank(),
        panel.grid.major = element_blank(),
        panel.background = element_blank(),
        plot.title = element_text(size = 10, color = "black"),
        axis.title = element_text(size = 10, color = "black"),
        axis.text.x = element_text(size = 10, color = "black"),
        axis.text.y = element_text(size = 10, color = "black"),
        strip.text.x = element_text(size = 10, margin = margin())) +
  labs(title = "Distribution of FDRs of GR-meQTL CpGs across bloss cell type", x = "-log10 FDR", y = ""
  scale_fill_manual(values = cbPalette[2:3]) +
  scale_color_manual(values = cbPalette[2:3])
```

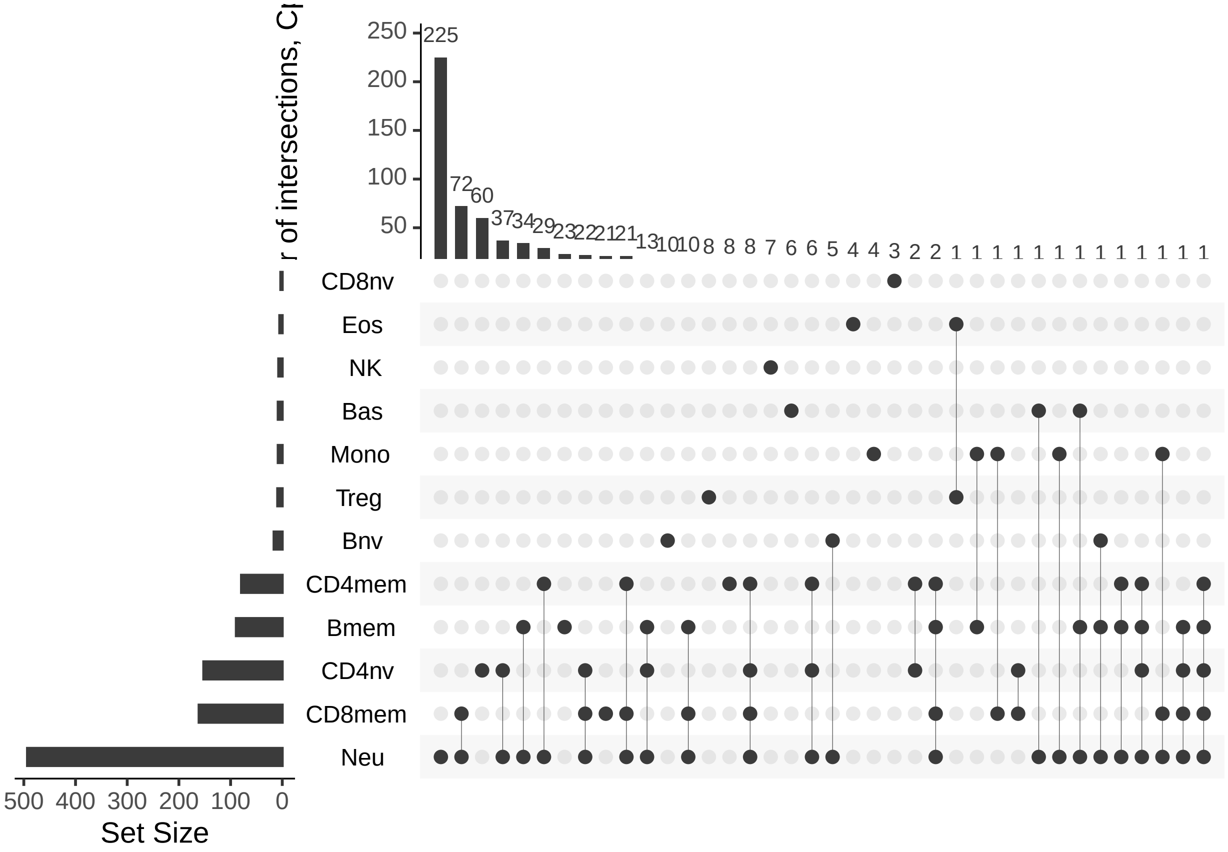


```
plt.size <- 12

ggplot(sign.pval.delta.mqtl.df, aes(x = Type, fill = Treatment)) +
  geom_bar(stat = "count", alpha = 1, position = position_dodge(width = .9)) +
  geom_text(aes(label = scales::percent(stat(count) / length(intersect.cpgs), accuracy = 0.1)),
            stat = "count", vjust = -0.5, size = plt.size / 3, position = position_dodge(width = 0.9))
  geom_text(data = subset(sign.pval.delta.mqtl.df[Treatment == "veh"], p.value <= 0.05),
            aes(label = "*"), stat = "count", vjust = 0.15,  size = plt.size) +
  theme(legend.position = "bottom", # c(0.9, 0.9),
        legend.title = element_blank(),
        legend.text = element_text(size = plt.size),
        panel.grid.major = element_blank(),
        panel.background = element_blank(),
        plot.title = element_text(size = plt.size),
        axis.title = element_text(size = plt.size),
        axis.text = element_text(size = plt.size, colour = "black")) +
  labs(title = "Distribution of CpGs significant at FDR = 0.05 across 12 blood cell types",
       x = "", y = "") +
  scale_fill_manual(values = cbPalette[2:3])
```



Upset plot for DEX CpGs

```
blood.types.df.lst  <- split(sign.pval.delta.mqtl.df[Treatment == "dex"], by = "Type", )
blood.types.cpg.lst <- lapply(blood.types.df.lst, function(df) df$CpG_ID)

upset(fromList(blood.types.cpg.lst),
      nsets = 12,
      nintersects = 50,
      mainbar.y.label = "Number of intersections, CpGs",
      keep.order = T,
      order.by = "freq",
      text.scale = 1.3,
      line.size = 0.1,
      mb.ratio = c(0.3, 0.7))
```



Cell-type specificity on DMPs

```
dmps.sign.anno.fn <- "~/bio/code/mpip/dex-stim-human-array/output/data/methylation/02_dmp/dmps_fdr01_fc
dmps.sign.anno.df <- fread(dmps.sign.anno.fn)
dmps              <- dmps.sign.anno.df$PROBE_ID %>% unique()
sign.pval.dmps.df <- sign.pval.df[CpG_ID %in% dmps, ]

sign.pval.dmps.dex <- sign.pval.dmps.df[Treatment == "dex", CpG_ID]
sign.pval.dmps.veh <- sign.pval.dmps.df[Treatment == "veh", CpG_ID]

intersect.dmps     <- intersect(sign.pval.dmps.dex, sign.pval.dmps.veh)
sign.pval.dmps.df  <- sign.pval.dmps.df[CpG_ID %in% intersect.dmps,]

print(paste0("Number of unique dex CpGs: ", length(unique(sign.pval.dmps.dex))))
```

```
## [1] "Number of unique dex CpGs: 9095"
```

```
print(paste0("Number of unique baseline CpGs: ", length(unique(sign.pval.dmps.veh))))
```

```
## [1] "Number of unique baseline CpGs: 9848"
```

```
print(paste0("Number of intesecting CpGs: ", length(unique(intersect.dmps))))
```

```
## [1] "Number of intesecting CpGs: 9094"
```

Kolmogorov-Smirnov test: Do 2 samples follow the same distribution?

```
blood.cell.types        <- unique(sign.pval.delta.mqtl.df$Type)

test.res.df <- lapply(blood.cell.types, function(i){  # for each blood cell type
  dex.fdr.lst <- sign.pval.dmps.df[Type == i][Treatment == "dex", fdr]
  veh.fdr.lst <- sign.pval.dmps.df[Type == i][Treatment == "veh", fdr]
  test.rslt   <- ks.test(dex.fdr.lst, veh.fdr.lst)

  return(data.frame(Type = i, "p-value" = test.rslt$p.value))
  }) %>%
  bind_rows()
```

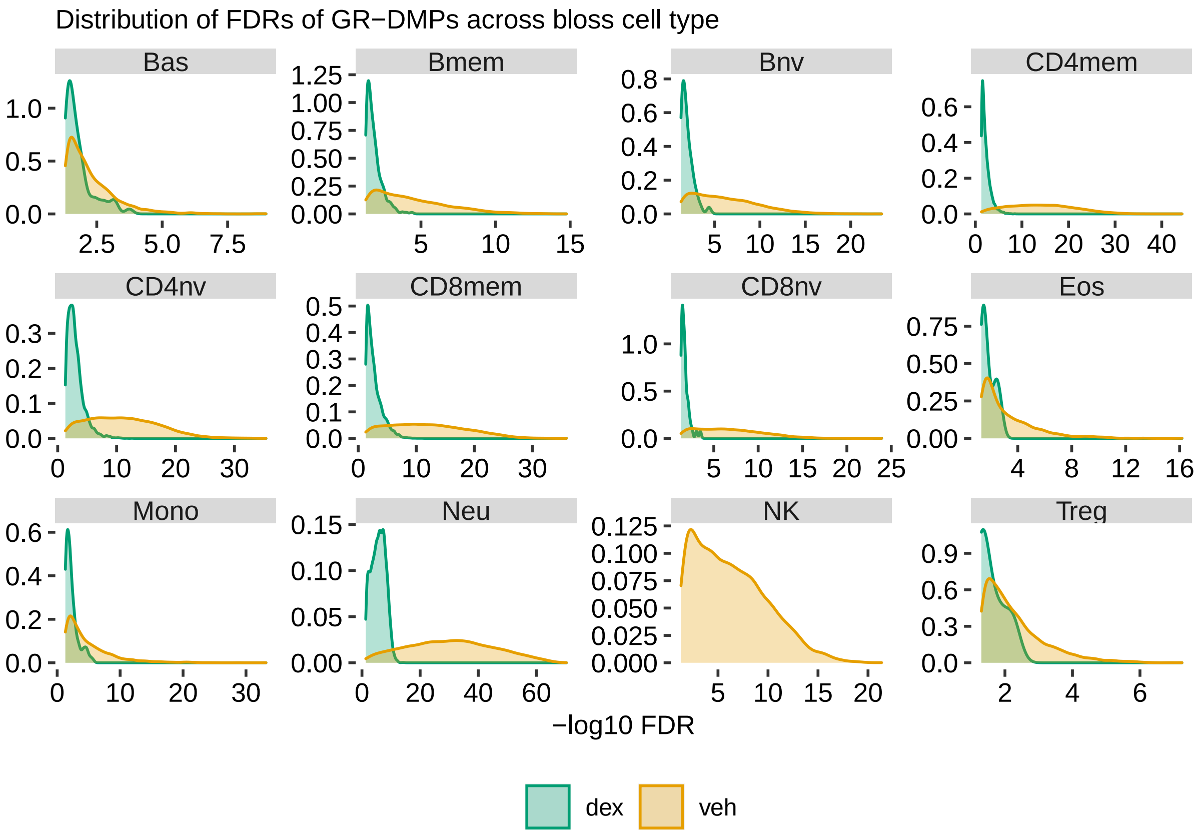
```
## Warning in ks.test(dex.fdr.lst, veh.fdr.lst): p-value will be approximate in the
## presence of ties

## Warning in ks.test(dex.fdr.lst, veh.fdr.lst): p-value will be approximate in the
## presence of ties

## Warning in ks.test(dex.fdr.lst, veh.fdr.lst): p-value will be approximate in the
## presence of ties

## Warning in ks.test(dex.fdr.lst, veh.fdr.lst): p-value will be approximate in the
## presence of ties

## Warning in ks.test(dex.fdr.lst, veh.fdr.lst): p-value will be approximate in the
## presence of ties

## Warning in ks.test(dex.fdr.lst, veh.fdr.lst): p-value will be approximate in the
## presence of ties

## Warning in ks.test(dex.fdr.lst, veh.fdr.lst): p-value will be approximate in the
## presence of ties

## Warning in ks.test(dex.fdr.lst, veh.fdr.lst): p-value will be approximate in the
## presence of ties

## Warning in ks.test(dex.fdr.lst, veh.fdr.lst): cannot compute exact p-value with
## ties

## Warning in ks.test(dex.fdr.lst, veh.fdr.lst): p-value will be approximate in the
## presence of ties
```

```
sign.pval.dmps.df <- left_join(sign.pval.dmps.df, test.res.df)
```

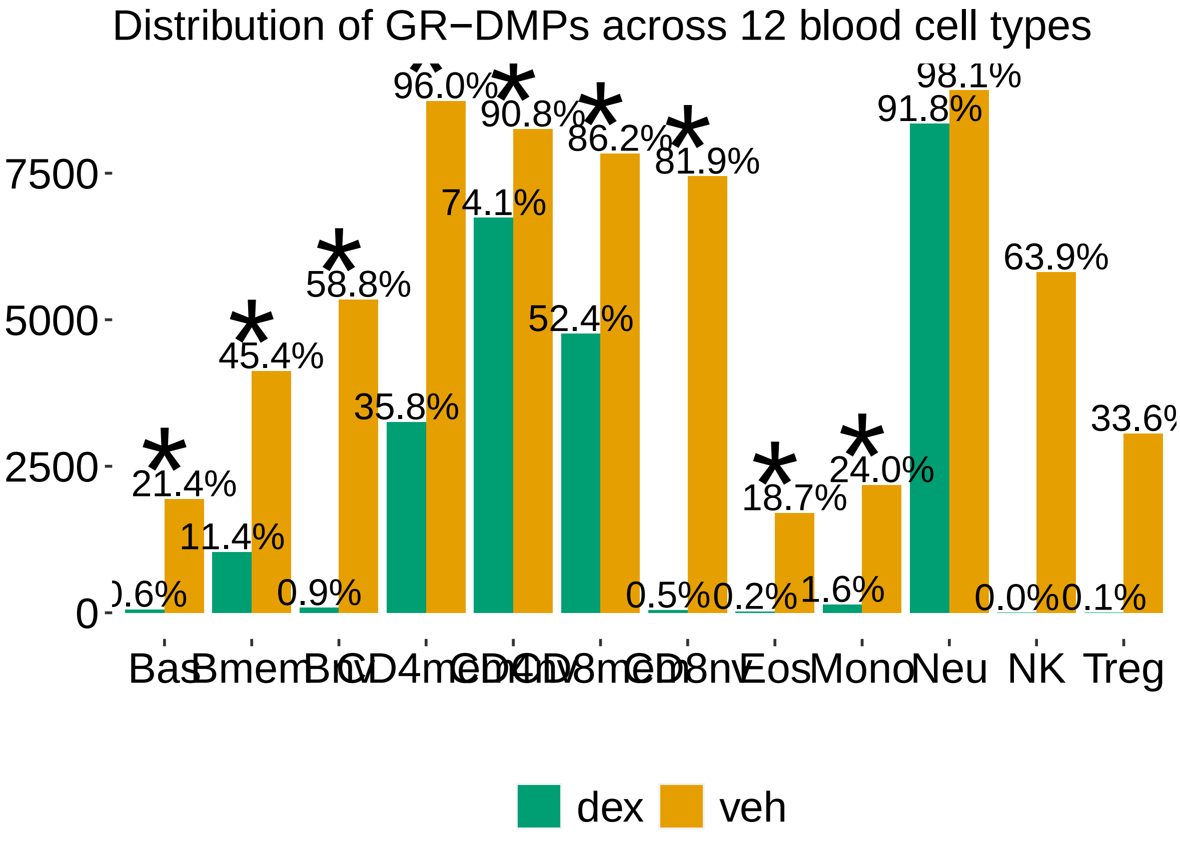
```
## Joining, by = "Type"
```

```
ggplot(sign.pval.dmps.df, aes(x = -log10(fdr), fill = Treatment)) +
  geom_density(alpha = 0.3, aes(color = Treatment)) +
  geom_text(data = subset(unique(sign.pval.dmps.df[, .(Treatment, Type, p.value)])),
            aes(x = -Inf, y = -Inf, label = paste0("KS test = ", signif(p.value, 3))),
            hjust = -0.5, vjust = -15, size = 5, color = "black" ) +
  facet_wrap(~ Type, scales = "free", ncol = 4) +
  theme(legend.position = "bottom", # c(.9,.9),
        legend.title = element_blank(),
        panel.grid.major = element_blank(),
        panel.background = element_blank(),
        plot.title = element_text(size = 10, color = "black"),
        axis.title = element_text(size = 10, color = "black"),
        axis.text.x = element_text(size = 10, color = "black"),
        axis.text.y = element_text(size = 10, color = "black"),
        strip.text.x = element_text(size = 10, margin = margin())) +
  labs(title = "Distribution of FDRs of GR-DMPs across bloss cell type", x = "-log10 FDR", y = "") +
  scale_fill_manual(values = cbPalette[2:3]) +
  scale_color_manual(values = cbPalette[2:3])
```

```
## Warning: Groups with fewer than two data points have been dropped.

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning
## -Inf
```



```r
ggplot(sign.pval.dmps.df, aes(x = Type, fill = Treatment)) +
  geom_bar(stat = "count", alpha = 1, position = position_dodge(width = .9)) +
  geom_text(aes(label = scales::percent(stat(count) / length(intersect.dmps), accuracy = 0.1)),
            stat = "count", vjust = -0.1, size = 5, position = position_dodge(width = .9)) +
  geom_text(data = subset(sign.pval.dmps.df[Treatment == "veh"], p.value <= 0.05),
             aes(label = "*"), stat = "count", vjust = 0.15,size = 16) +
  theme(legend.position = "bottom", # c(0.9, 0.9),
        legend.title = element_blank(),
        legend.text = element_text(size = 16),
        panel.grid.major = element_blank(),
        panel.background = element_blank(),
        plot.title = element_text(size = 16),
        axis.title = element_text(size = 16),
        axis.text = element_text(size = 16, colour = "black")) +
  labs(title = "Distribution of GR-DMPs across 12 blood cell types",
       x = "", y = "") +
  scale_fill_manual(values = cbPalette[2:3])
```


Distribution of GR−DMPs across 12 blood cell types
).6%
*
*
*
*
*
*
*
*
*
Bas
Bmem
Bnv
CD4mem
CD4nv
CD8mem
CD8nv
Eos
Mono
Neu
NK
Treg
🟩 dex
🟧 veh