

MDS_geneExpression_sexCheck.R

kimberlyolney

2020-10-08

```
library(limma)
library(edgeR)

## Warning: package 'edgeR' was built under R version 3.6.1
library(RColorBrewer)
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.6.2
library(tidyverse)

## -- Attaching packages -----
## v tibble  3.0.3      v dplyr   1.0.1
## v tidyr   1.1.1      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.5.0
## v purrr   0.3.4
## Warning: package 'tibble' was built under R version 3.6.2
## Warning: package 'tidyr' was built under R version 3.6.2
## Warning: package 'purrr' was built under R version 3.6.2
## Warning: package 'dplyr' was built under R version 3.6.2
## -- Conflicts -----
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
library(matrixStats)

##
## Attaching package: 'matrixStats'
## The following object is masked from 'package:dplyr':
##
##     count
library(reshape)

##
## Attaching package: 'reshape'
## The following object is masked from 'package:dplyr':
##
##     rename
## The following objects are masked from 'package:tidyr':
```

```

##
##      expand, smiths
library(ggpubr)

## Warning: package 'ggpubr' was built under R version 3.6.2
library("gridExtra")

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##      combine
# set working directory
#setwd("~/Dropbox (ASU)/Placenta/BATCH2_PLACENTA_DECIDUA_ANALYSIS/HISAT_FeatureCounts/batch1_and_batch2")

# Read in count, genes, and phenotype data

counts_g <- read.delim("counts_pheno/placenta_batch1and2_geneCounts.tsv", header=TRUE, sep="\t")
colnames(counts_g) <- str_replace_all(colnames(counts_g), pattern="\\.","-") # replace . with - in samp
genes <- read.csv("counts_pheno/genesID.csv", header=TRUE, sep = ",")

counts_t <- read.delim("counts_pheno/placenta_batch1and2_transcriptCounts.tsv", header=TRUE, sep="\t")
colnames(counts_t) <- str_replace_all(colnames(counts_t), pattern="\\.","-") # replace . with - in samp
transcripts <- read.csv("counts_pheno/transcriptsID.csv", header=TRUE, sep = ",")

#pheno <- read.csv("counts_pheno/placenta_pheno.txt", header=TRUE, sep="\t")
pheno <- read.csv("counts_pheno/200508_placentas_pheno.csv", header=TRUE, sep=",")

placenta_batch1_sampleIDs <- c("OBG0044-1", "OBG0044-2", "OBG0053-1", "OBG0053-2", "OBG0068-1",
                              "OBG0068-2", "OBG0111-1", "OBG0111-2", "OBG0112-1", "OBG0112-2",
                              "OBG0115-1", "OBG0115-2", "OBG0116-1", "OBG0116-2", "OBG0117-1",
                              "OBG0117-2", "OBG0118-1", "OBG0118-2", "OBG0120-1", "OBG0120-2",
                              "OBG0122-1", "OBG0122-2", "OBG0123-1", "OBG0123-2", "OBG0126-1",
                              "OBG0126-2", "OBG0130-1", "OBG0130-2", "OBG0132-1", "OBG0132-2",
                              "OBG0133-1", "OBG0133-2", "OBG0156-1", "OBG0156-2", "OBG0158-1",
                              "OBG0158-2", "OBG0166-1", "OBG0166-2", "OBG0170-1", "OBG0170-2",
                              "OBG0174-1", "OBG0174-2", "OBG0175-1", "OBG0175-2", "OBG0178-1",
                              "OBG0178-2", "YPOPS0006-1", "YPOPS0006-2")

# batch 2 samples
placenta_batch2_sampleIDs <- c("OBG0014-1", "OBG0014-2", "OBG0015-1", "OBG0015-2", "OBG0019-1",
                              "OBG0019-2", "OBG0021-1", "OBG0021-2", "OBG0022-1", "OBG0022-2",
                              "OBG0024-1", "OBG0024-2", "OBG0026-1", "OBG0026-2", "OBG0027-1",
                              "OBG0027-2", "OBG0028-1", "OBG0028-2", "OBG0029-1", "OBG0029-2",
                              "OBG0030-1", "OBG0030-2", "OBG0031-1", "OBG0031-2", "OBG0032-1",
                              "OBG0032-2", "OBG0039-1", "OBG0039-2", "OBG0047-1", "OBG0047-2",
                              "OBG0050-1", "OBG0050-2", "OBG0051-1", "OBG0051-2", "OBG0053B2-1",
                              "OBG0053B2-2", "OBG0065-1", "OBG0065-2", "OBG0066-1", "OBG0066-2",
                              "OBG0085-1", "OBG0085-2", "OBG0090-1", "OBG0090-2", "OBG0107-1",
                              "OBG0107-2", "OBG0121-1", "OBG0121-2", "OBG0138-1", "OBG0138-2",
                              "OBG0149-1", "OBG0149-2", "OBG0180-1", "OBG0180-2", "OBG0188-1",
                              "OBG0188-2", "OBG0191-1", "OBG0191-2", "OBG0201-1", "OBG0201-2",

```

```

"OBG0205-1", "OBG0205-2", "OBG0289-1", "OBG0289-2", "OBG0338-1",
"OBG0338-2", "OBG0342-1", "OBG0342-2", "YPOPS0007M-1", "YPOPS0007M-2",
"YPOPS0123M-1", "YPOPS0123M-2")

# samples to remove due to failed QC and/or outlier in MDS plot
placenta_batch1_removals <- c("OBG0174-1", "OBG0174-2", "OBG0175-1", "OBG0175-2")
placenta_batch2_removals <- c("OBG0015-1", "OBG0015-2", "OBG0065-1", "OBG0065-2",
"OBG0188-1", "OBG0188-2", "OBG0014-1", "OBG0014-2",
"OBG0026-1", "OBG0026-2", "YPOPS0007M-1", "YPOPS0007M-2",
"OBG0019-1", "OBG0019-2", "OBG0021-1", "OBG0021-2")

placenta_removals <- c(placenta_batch2_sampleIDs)
all_removals <- c(placenta_removals)

samplesToRemove <- c(all_removals) # update depending on comparison being made
SAMPLE_LENGTH <- as.numeric(length(samplesToRemove)) # to call later
half_sample_length <- SAMPLE_LENGTH/2 # half the sample length

removals_g <- (names(counts_g) %in% samplesToRemove[1:SAMPLE_LENGTH]) # for matching names create a val
counts_ExRemovals_g <- counts_g[!removals_g] # create a new counts file that excludes (Ex) the removals

removals_t <- (names(counts_t) %in% samplesToRemove[1:SAMPLE_LENGTH]) # for matching names create a val
counts_ExRemovals_t <- counts_t[!removals_t] # create a new counts file that excludes (Ex) the removals
pheno_ExRemovals <- pheno[! pheno$sample %in% samplesToRemove[1:SAMPLE_LENGTH],] # update 1:16 dependin

# create a DGElist

dge_g <- DGEList(counts=counts_ExRemovals_g, genes=genes)
dge_t <- DGEList(counts=counts_ExRemovals_t, genes=transcripts)
dim(dge_g)

## [1] 57133 48

dim(dge_t)

## [1] 206694 48

# organize sample information

samplenames <- (pheno_ExRemovals$sample) # sample names are not unique because a sample may belong to m
as.data.frame(samplenames)

##      samplenames
## 1      OBG0044-1
## 2      OBG0044-2
## 3      OBG0053-1
## 4      OBG0053-2
## 5      OBG0068-1
## 6      OBG0068-2
## 7      OBG0111-1
## 8      OBG0111-2
## 9      OBG0112-1
## 10     OBG0112-2
## 11     OBG0115-1
## 12     OBG0115-2
## 13     OBG0116-1

```

```

## 14 OBG0116-2
## 15 OBG0117-1
## 16 OBG0117-2
## 17 OBG0118-1
## 18 OBG0118-2
## 19 OBG0120-1
## 20 OBG0120-2
## 21 OBG0122-1
## 22 OBG0122-2
## 23 OBG0123-1
## 24 OBG0123-2
## 25 OBG0126-1
## 26 OBG0126-2
## 27 OBG0130-1
## 28 OBG0130-2
## 29 OBG0132-1
## 30 OBG0132-2
## 31 OBG0133-1
## 32 OBG0133-2
## 33 OBG0156-1
## 34 OBG0156-2
## 35 OBG0158-1
## 36 OBG0158-2
## 37 OBG0166-1
## 38 OBG0166-2
## 39 OBG0170-1
## 40 OBG0170-2
## 41 OBG0174-1
## 42 OBG0174-2
## 43 OBG0175-1
## 44 OBG0175-2
## 45 OBG0178-1
## 46 OBG0178-2
## 47 YPOPS0006-1
## 48 YPOPS0006-2

```

```

colnames(dge_g) <- samplenames
colnames(dge_t) <- samplenames

```

```

# create groups for the samples
sex <- factor(pheno_ExRemovals$sex, levels=c("female", "male"))
batch<- factor(pheno_ExRemovals$batch, levels=c("1", "2"))

```

```

dge_g$samples$sex <- sex
dge_g$samples$batch <- batch

```

```

dge_t$samples$sex <- sex
dge_t$samples$batch <- batch

```

```

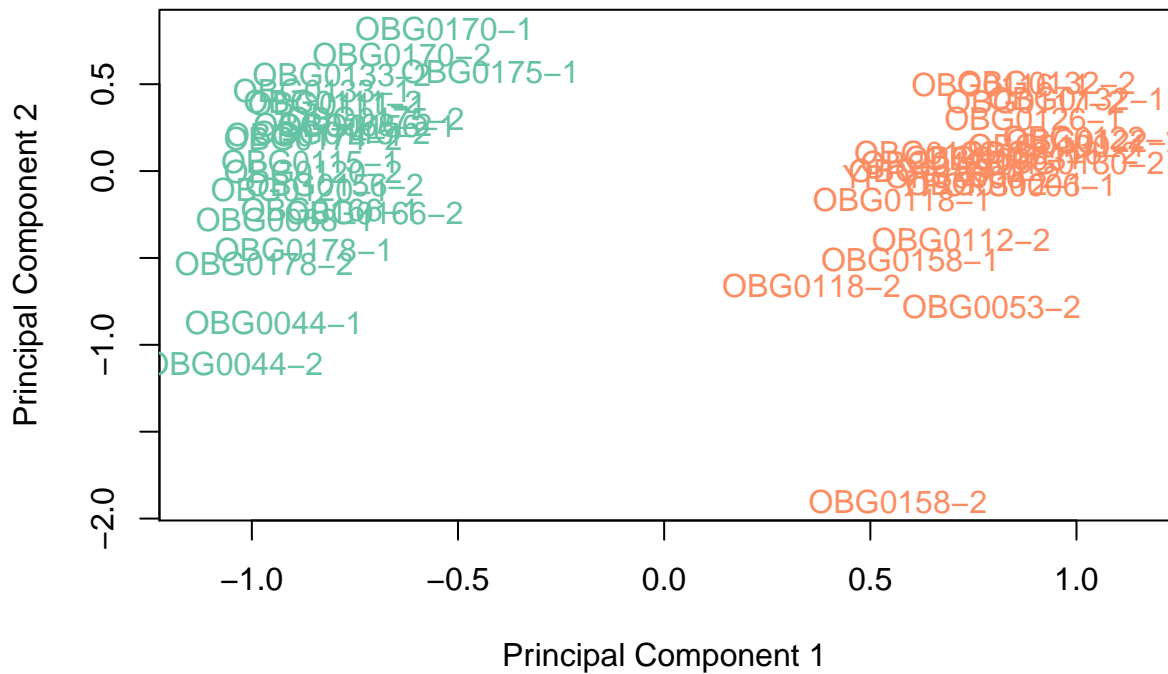
# data pre-processing

```

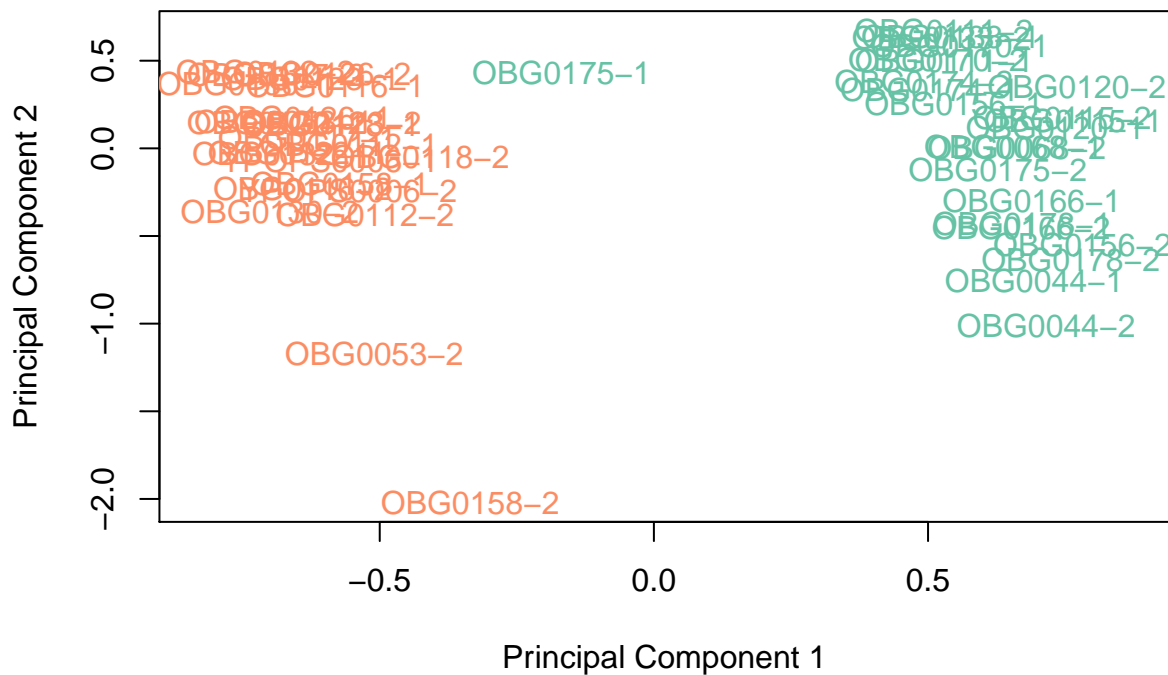
```

#dge_g <-sumTechReps(dge_g, dge_g$samples$rep) # comment out to not sum replicates
#dge_t <-sumTechReps(dge_t, dge_t$samples$rep) # comment out to not sum replicates

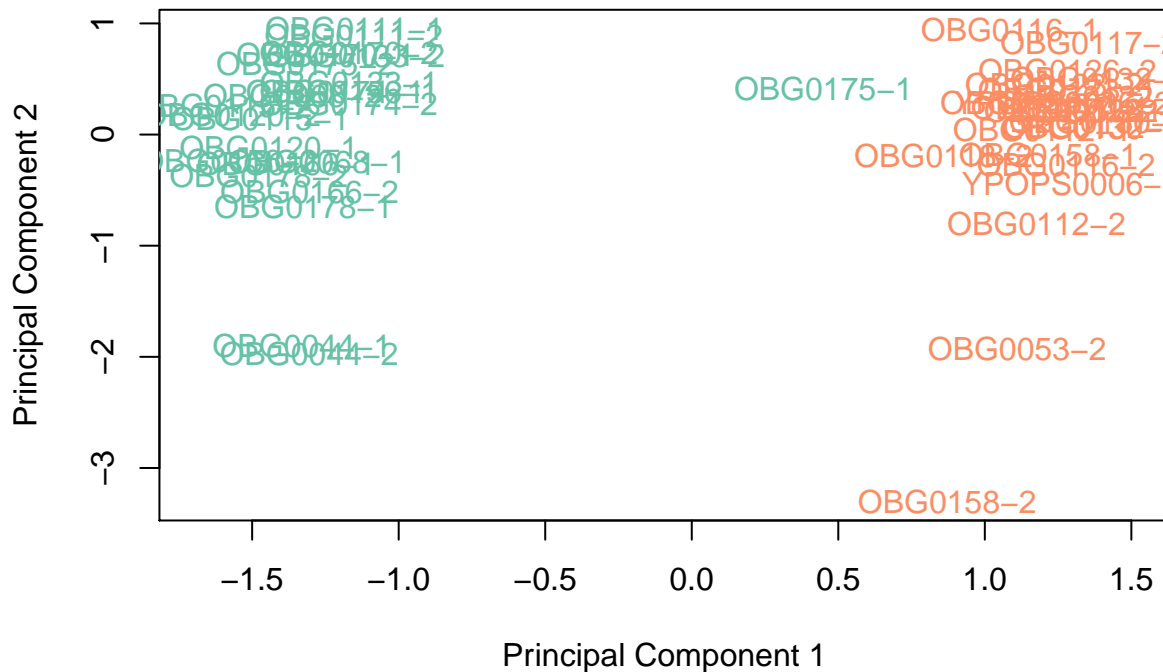
```

```
# all transcripts
plotMDS(lcpm_tran, col=col.sex,
        gene.selection = "common", dim.plot = c(1,2))
```



```
# top 100 transcripts
plotMDS(lcpm_tran, col=col.sex,
        top = 100,
        gene.selection = "common", dim.plot = c(1,2))
```



```
# XIST and Y-linked gene expression
```

```
may <- cbind(genes, cpm_gene)
may$length <- NULL
may2 <- melt(may, id=c("Geneid", "chr"))
df_merged <- merge(may2, pheno_ExRemovals, by.x="variable", by.y="sample", all.x=TRUE)
df <- df_merged[ -c(5,7:21) ]
head(df)
```

```
##   variable      Geneid chr    value  sex
## 1 OBG0044-1    DDX11L1 chr1 0.16879918 female
## 2 OBG0044-1    WASH7P  chr1 5.85170497 female
## 3 OBG0044-1    MIR6859-1 chr1 0.22506558 female
## 4 OBG0044-1  MIR1302-2HG chr1 0.05626639 female
## 5 OBG0044-1    MIR1302-2 chr1 0.00000000 female
## 6 OBG0044-1    FAM138A chr1 0.00000000 female
```

```
XIST_Ygenes <- read.delim("XIST_Ygenes.txt", header = TRUE)
sexGenes <- c("EIF1AY", "KDM5D", "UTY", "DDX3Y", "USP9Y", "PCDH11Y", "ZFY", "RPS4Y1", "XIST")
# subset the placental CPM data to only include XIST and Y-linked protein-coding genes
#CPM_geneExpression <- subset(df, Geneid %in% XIST_Ygenes$Geneid)
#CPM_geneExpression$Geneid <- droplevels(CPM_geneExpression$Geneid)
```

```
CPM_geneExpression <- subset(df, Geneid %in% sexGenes)
CPM_geneExpression$Geneid <- droplevels(CPM_geneExpression$Geneid)
```

```
violin_Func <- function(a) {
  geneDF <- subset(CPM_geneExpression, Geneid == a)
  means <- aggregate(value ~sex, geneDF, mean)
  p <- ggplot(geneDF, aes(x = sex, y = value, color = sex)) +
    geom_violin() + scale_color_manual(values = c("black", "#66C2A5", "#FC8D62")) +
    # facet_wrap(~geneComb) + theme(strip.text.x = element_text(size = 12)) +
    geom_boxplot(width = 0.1, outlier.shape = NA) +
    geom_jitter(aes(shape = factor(sex)),
```

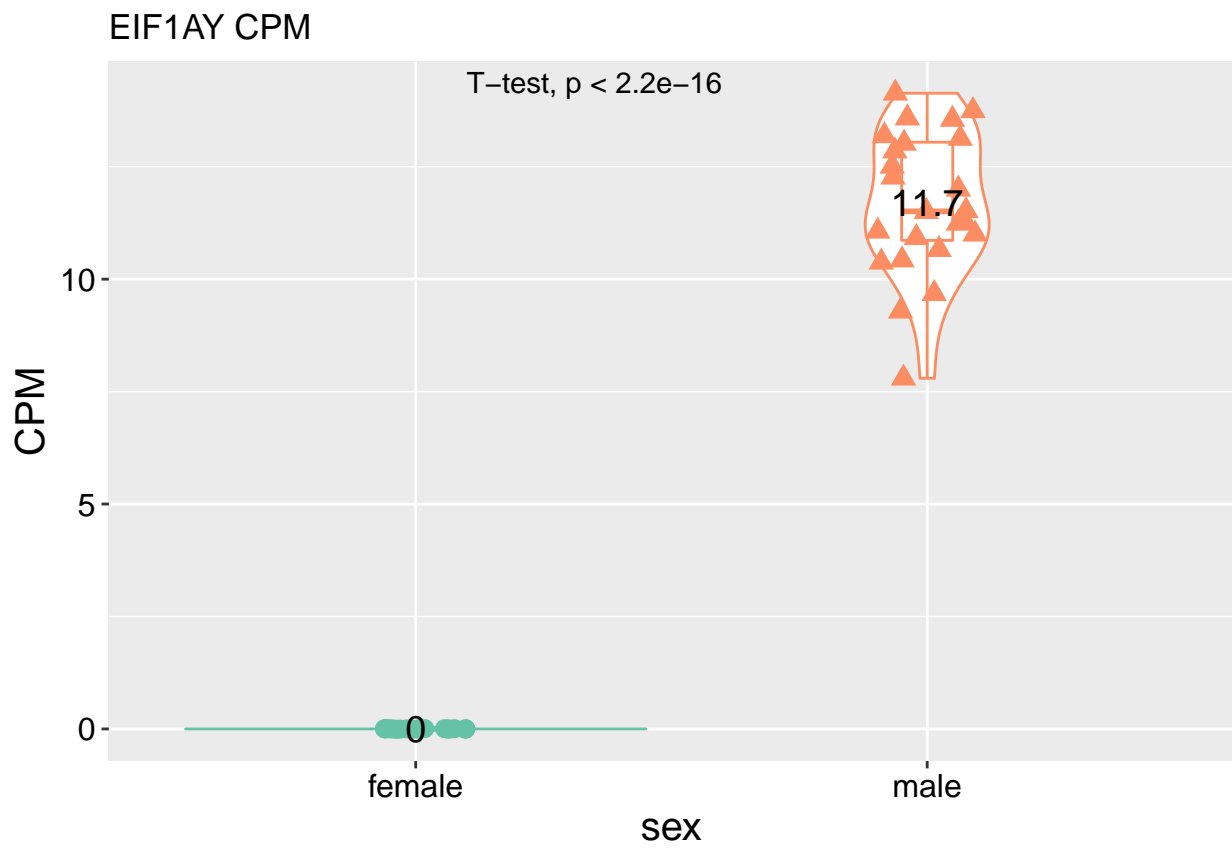
```

        size = 3,
        position = position_jitter(0.1)) +
theme(legend.position = "none") + ggtitle(paste0(a, " CPM")) +
theme(axis.title.x=element_text(size=15),
      axis.text.x=element_text(size=12)) +
theme(axis.title.y=element_text(size=15),
      axis.text.y=element_text(size=12)) +
theme(axis.title=element_text(size=15)) +
theme(legend.text=element_text(size=12)) +
theme(legend.title=element_text(size=15)) +
geom_text(
  data = means,
  aes(
    label = round(value, digits = 2),
    y = value,
    color = "black"
  ),
  position = position_dodge(width = 0.9),
  size = 5
) +
theme(axis.text = element_text(size = 5, colour="black")) +
stat_compare_means(method = "t.test",
                  label.x = 1.2,
                  label.y.npc = 1) +
labs(y = "CPM")
}
#violinPlots <- Map(violoin_Func, a = XIST_Ygenes$Geneid)
violinPlots <- Map(violoin_Func, a = sexGenes)

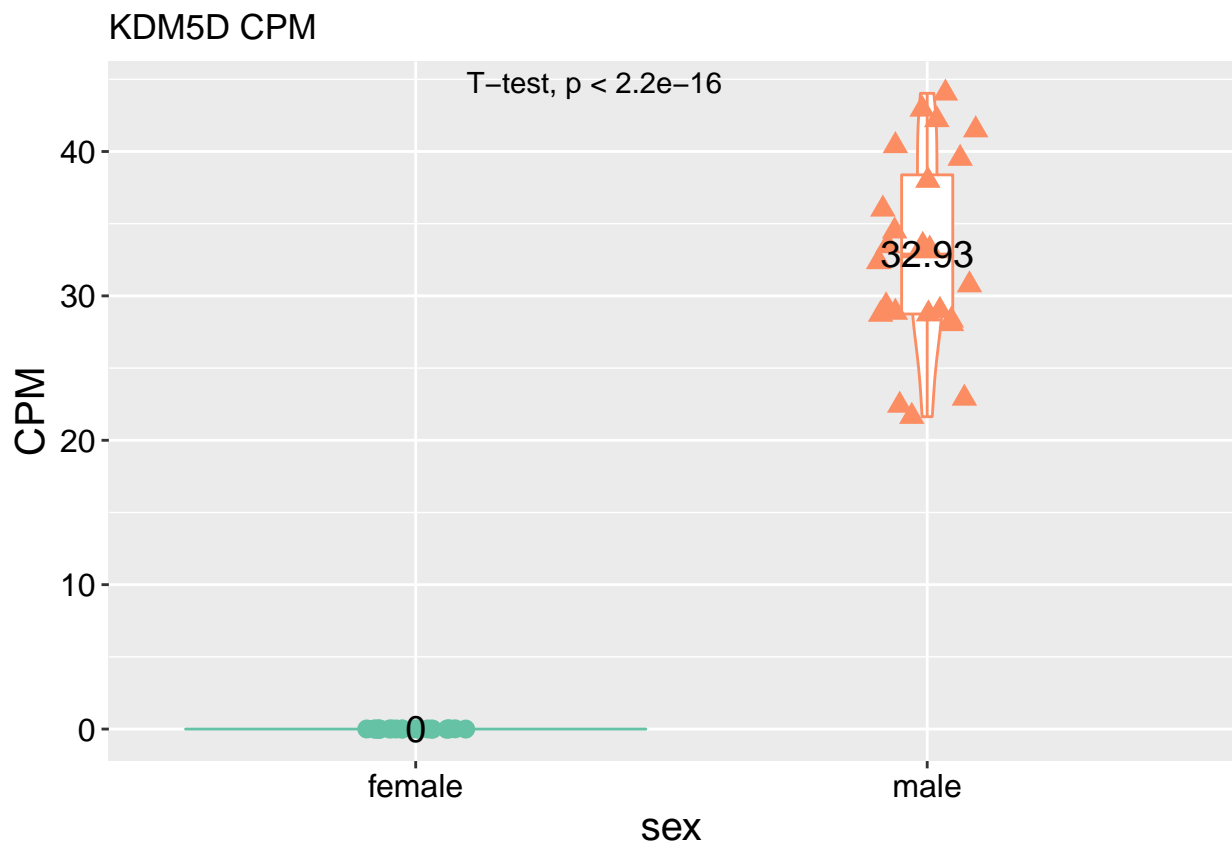
violinPlots

## $EIF1AY

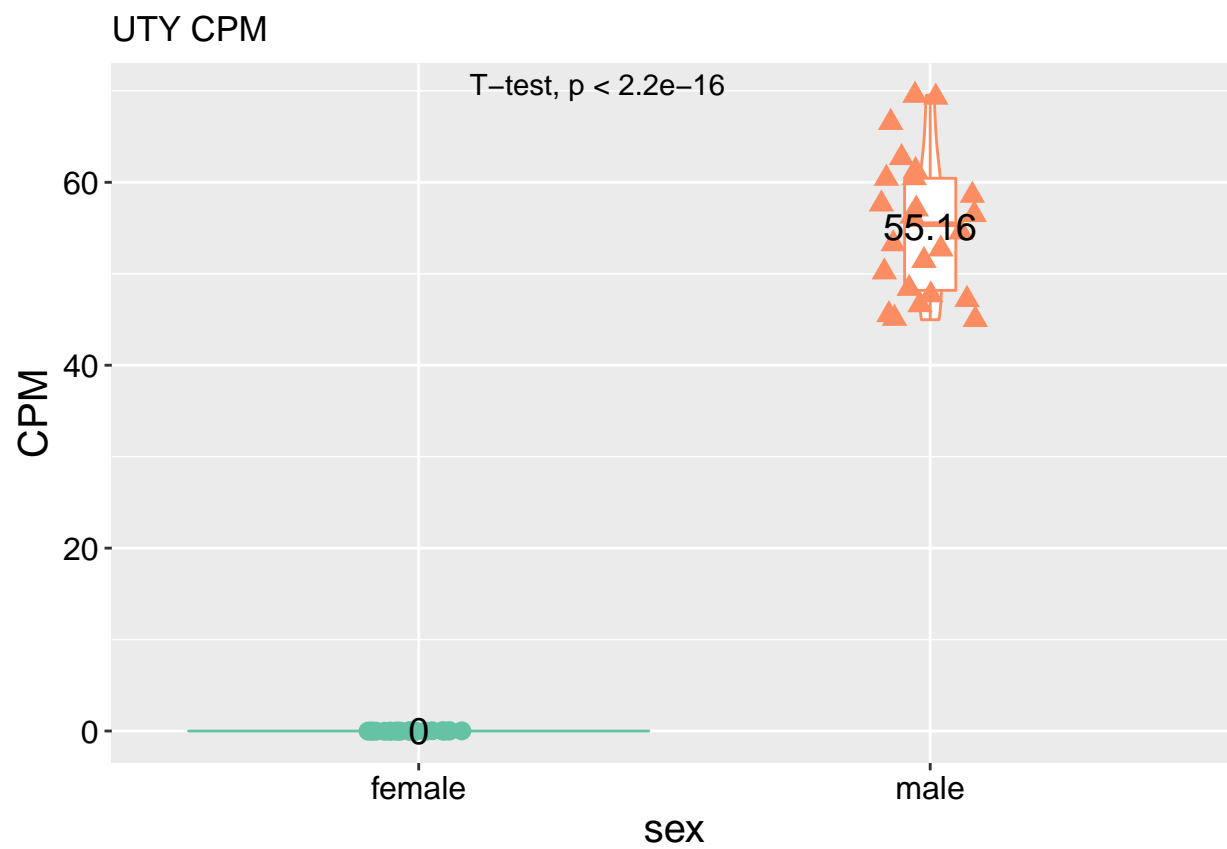
```

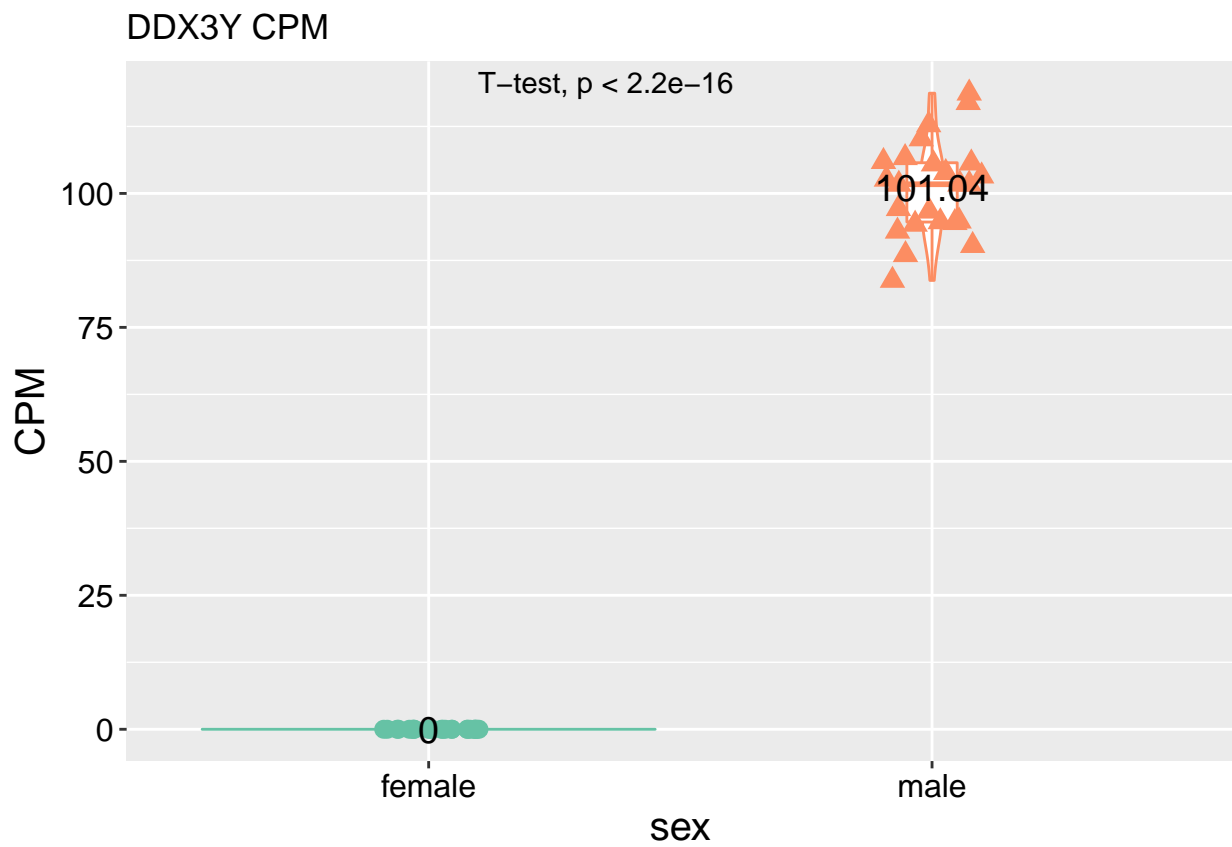



\$KDM5D

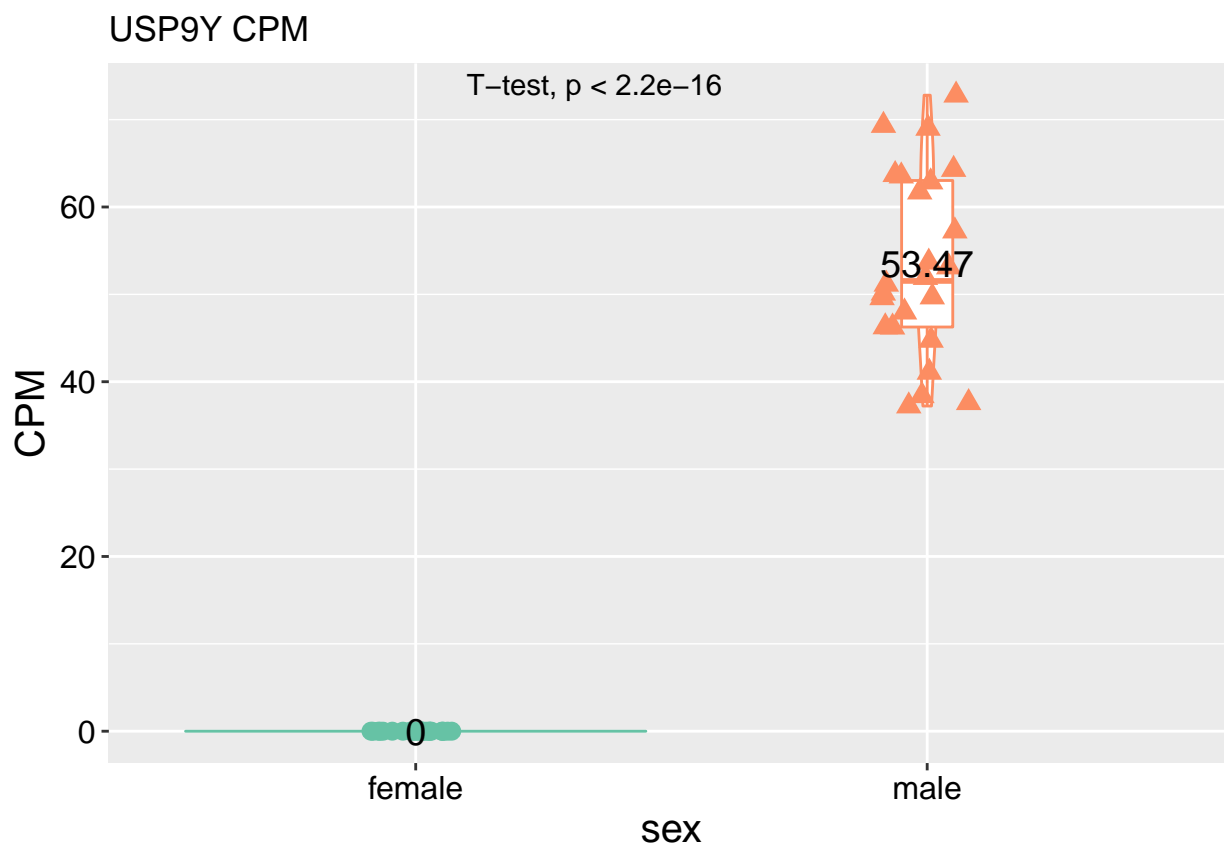


\$UTY

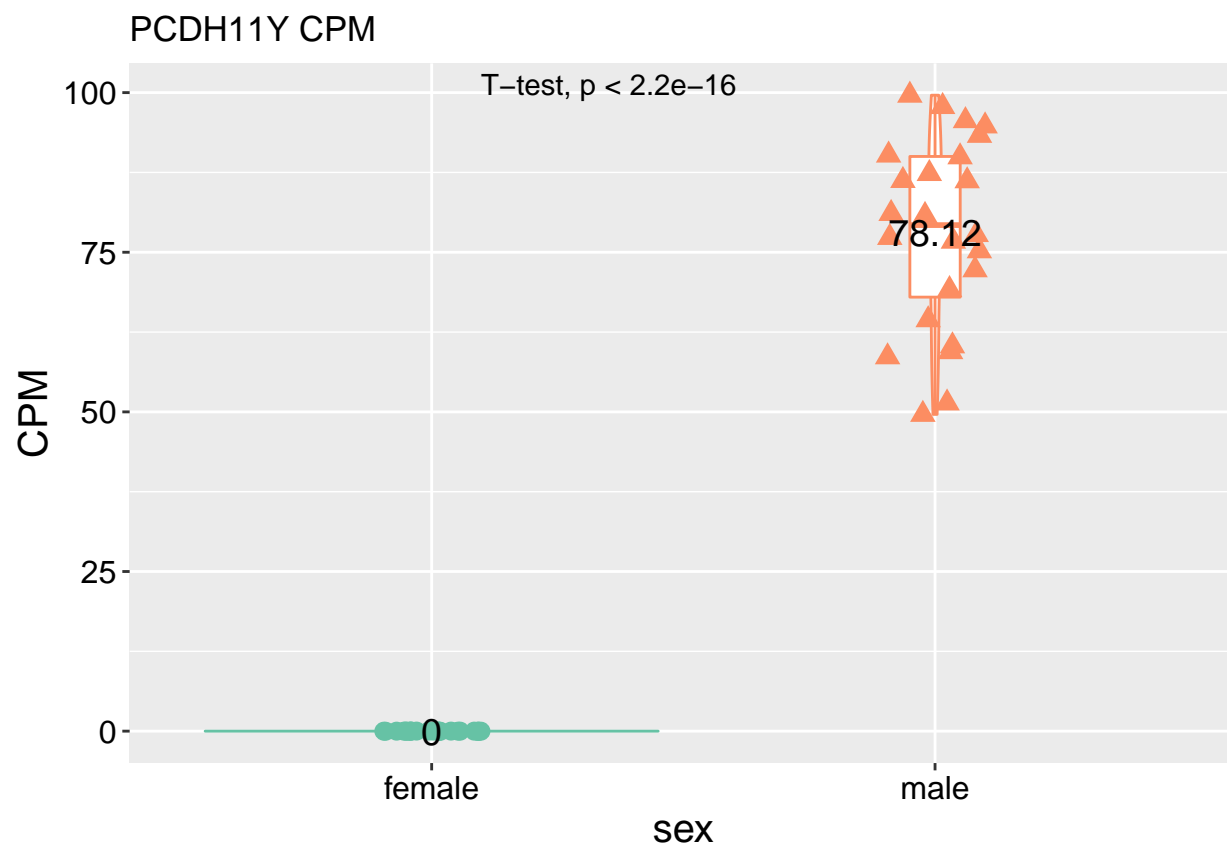


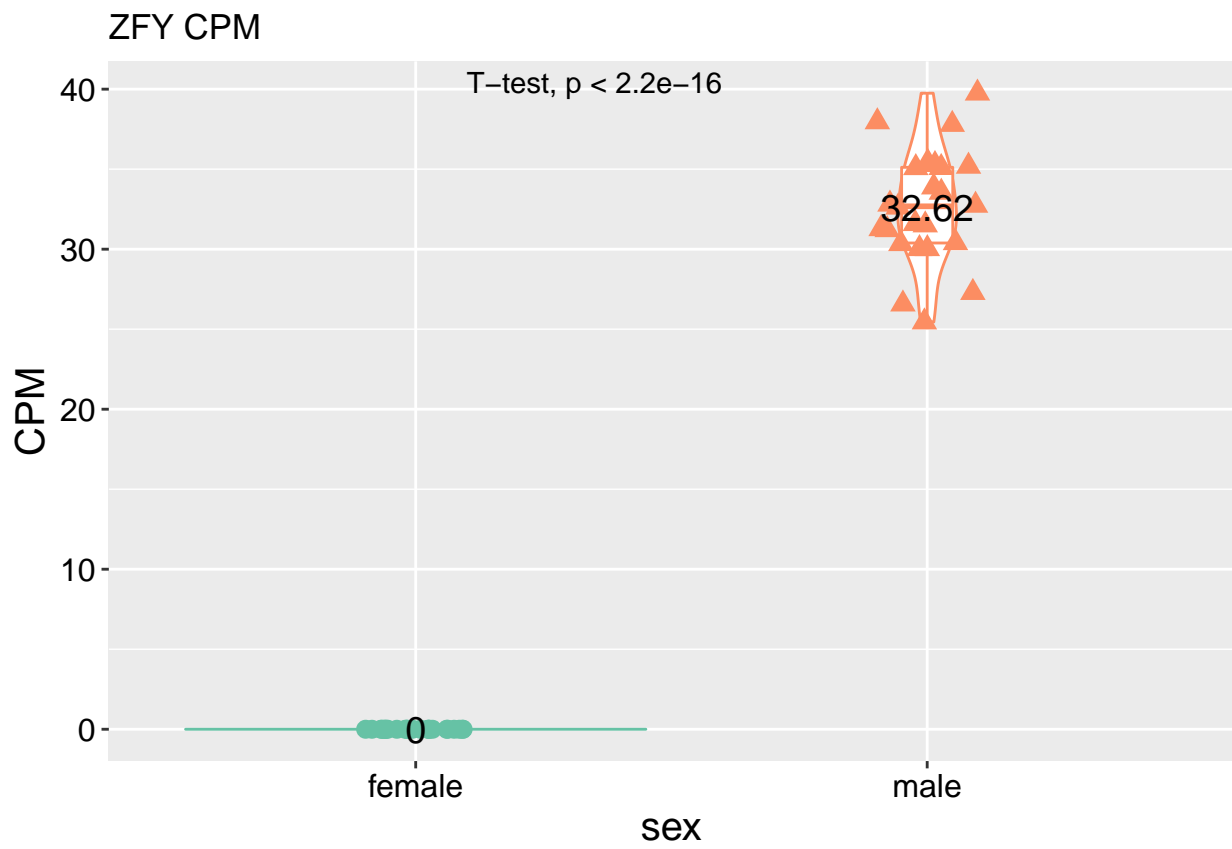


\$USP9Y

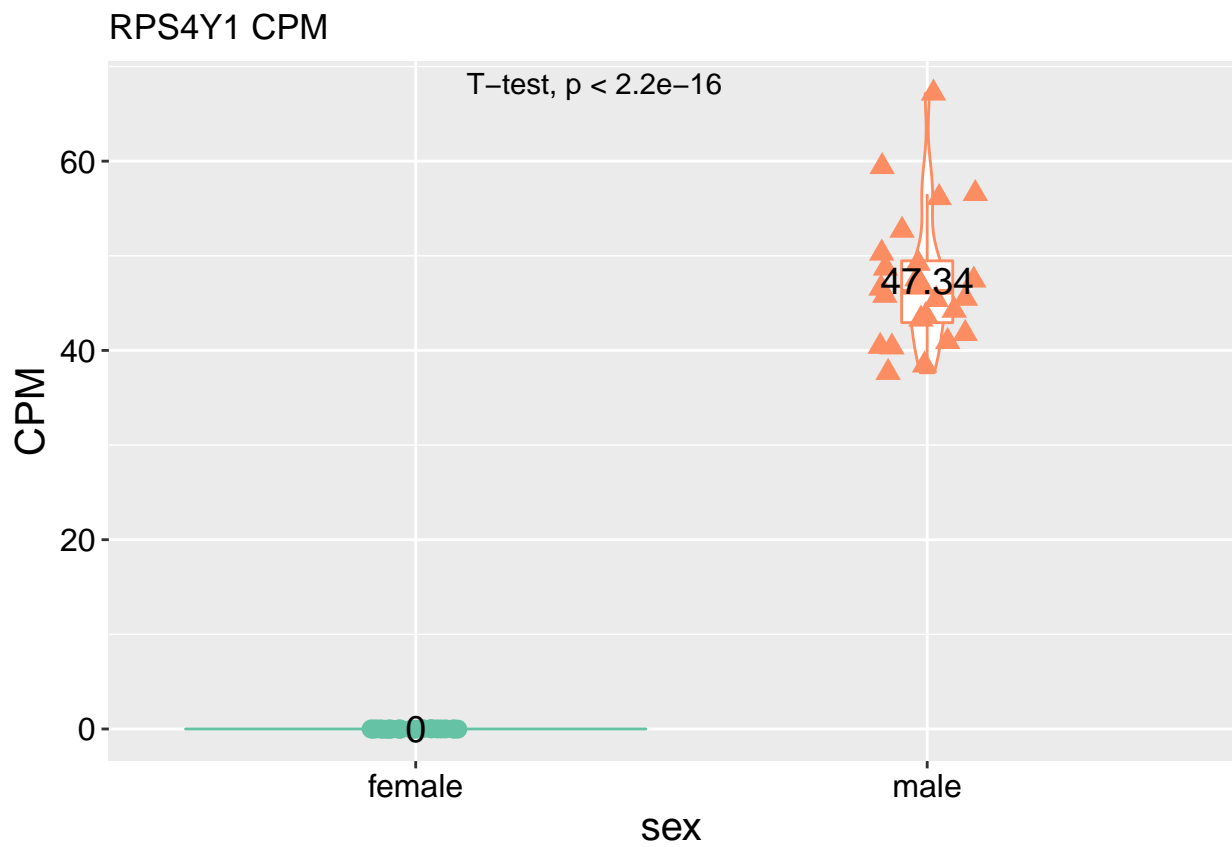


\$PCDH11Y

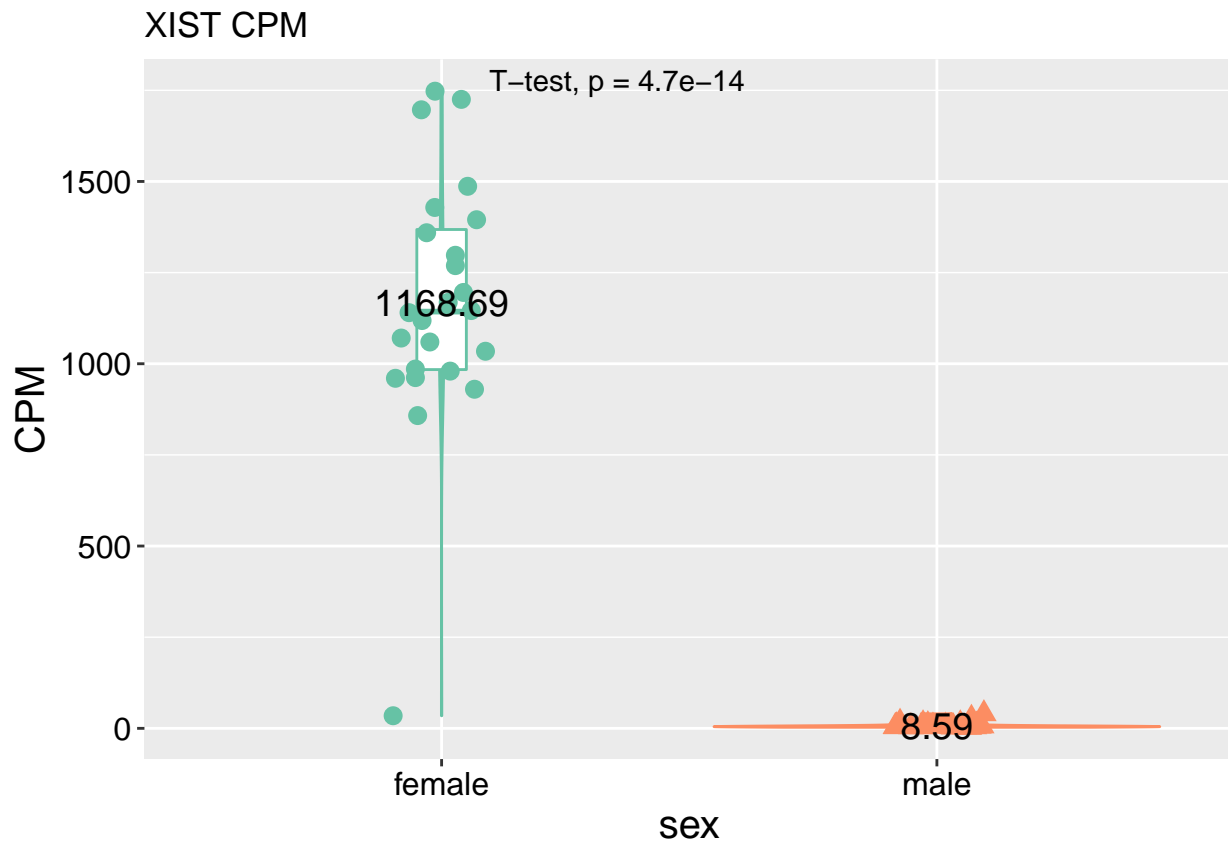




\$RPS4Y1



\$XIST



```
# Clear cluster of expression differences between male and female placenta samples
# EIF1AY
# KDM5D
# UTY
# DDX3Y
# USP9Y
# PCDH11Y
# ZFY
# RPS4Y1
# XIST
```

```
# Table of the mean and median expression for XIST and Y-linked genes in the placenta
```