# Data Visualization

*Joshua Chang*

*November 2017*

These packages need to be installed.

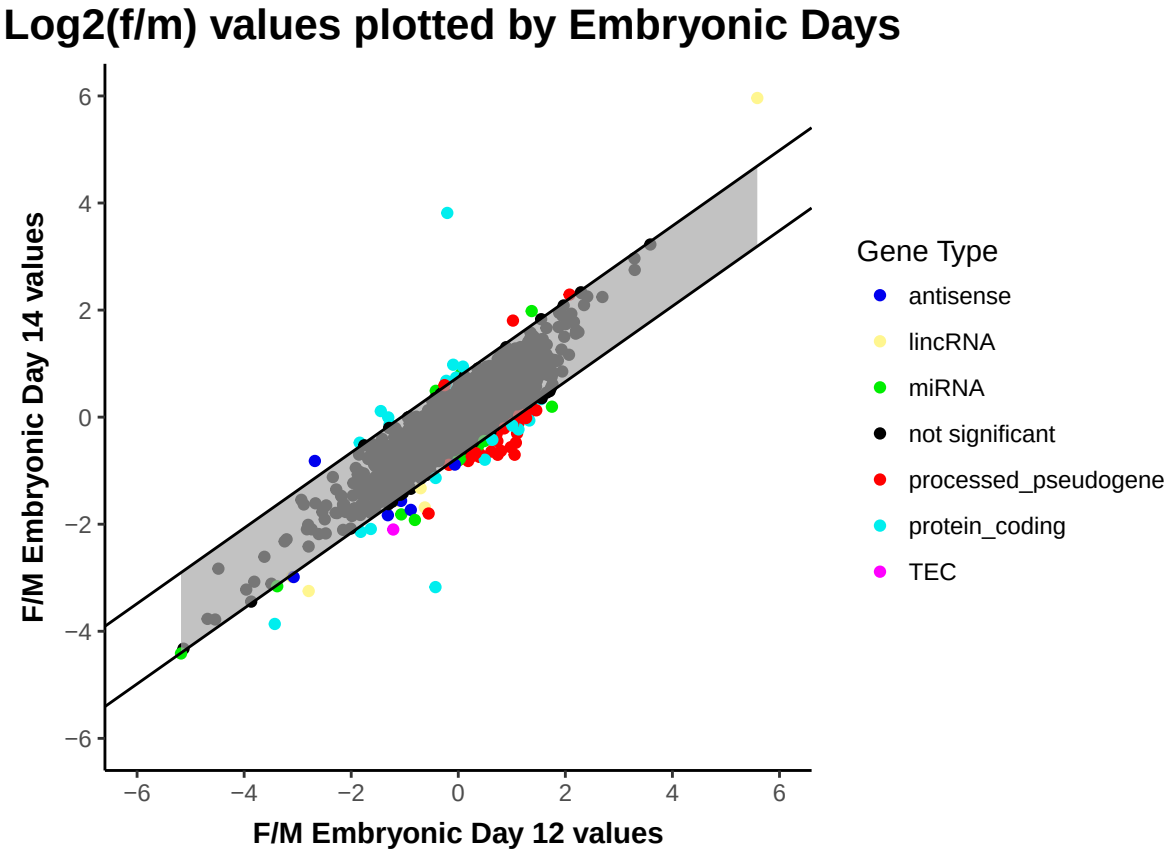
```
#install.packages("ggthemes")
#install.packages("gplots")
#install.packages("viridis")
#install.packages("readxl") #helps to read in xlxs files
#install.packages("ggsubplot")
#install.packages("RColorBrewer") #colors
#install.packages("heatmap.plus")
#install.packages("pheatmap")
#install.packages("Hmisc")
#install.packages("rlang")
#install.packages("ggpubr")
#install.packages("forcats")
#install.packages("colorspace")
#install.packages("ggplot2")
#install.packages("dplyr")
#install.packages("tidyr")
```

```
library(pheatmap) #pretty heat maps
library(gplots) #for heatmap2
library(ggplot2) #create graphs
library(viridis) #color schemes
library(forcats) #fct_inorder
library(readr) #read in files
library(RColorBrewer) # colors
library(readxl) #read in excel
library(tidyr) # tidy up dataframe
library(dplyr) #joining multiple dataframes together
library(tidyverse)
library(ggthemes) #adds plot themes
library(scales)
library(heatmap.plus) #heatmap
library(Hmisc)
library(ggpubr)
library(rlang)
library(colorspace) #new colors
library(dendextend) #dendogram
library(grid)
```

Scatterplots

```
#filter out dataset to only include the 6 genes with highest recorded counts
top6target = c("antisense","lincRNA","miRNA","processed_pseudogene","protein_coding","TEC")
top6type = filter(genders.log, type %in% top6target) #Filters to contain only types specified in top6ta

#we use the mutate function to create new columns calculated from old columns
mfgenders = mutate(top6type, e12mf = e12f - e12m ) #log(f/m) embryonic day 12
```

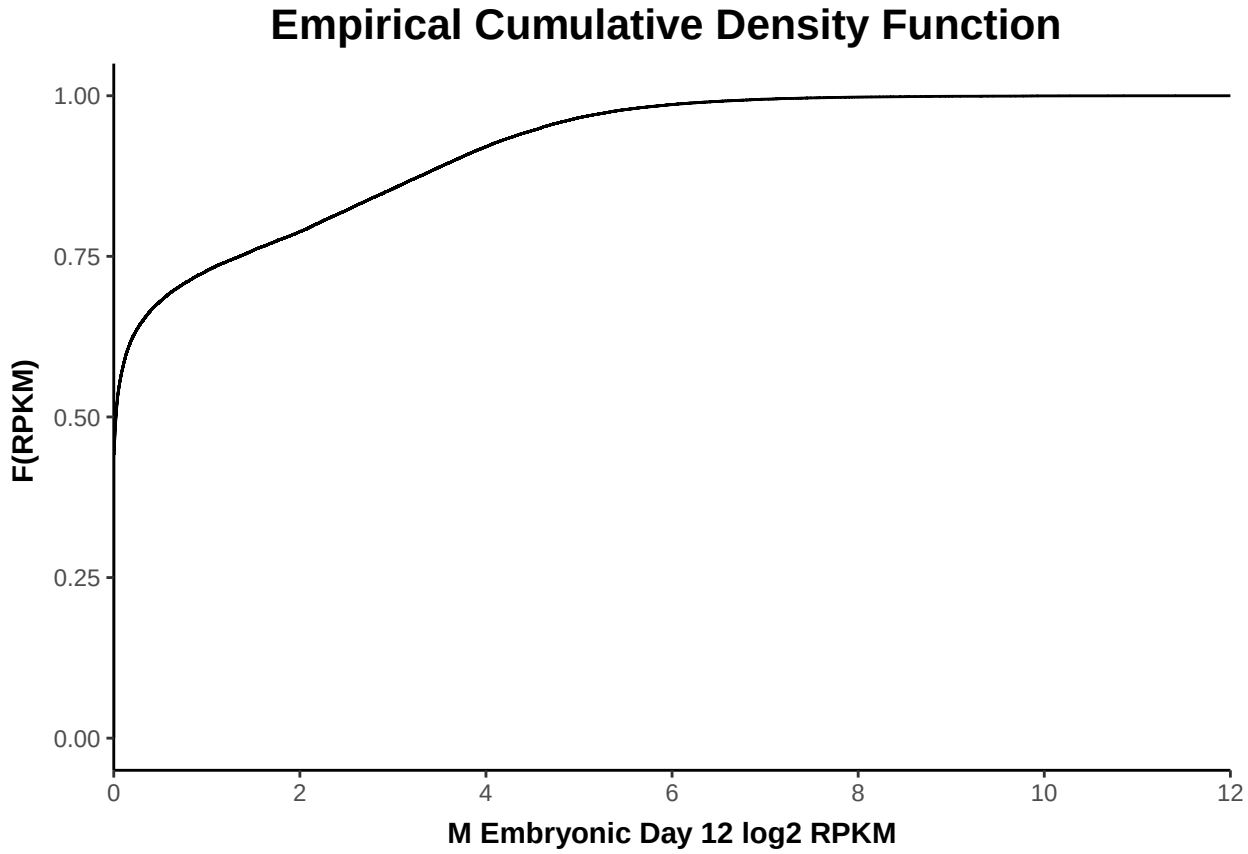
```r
mfgenders = mutate(mfgenders, e14mf = e14f - e14m) #log(f/m) embryonic day 14

#find linear regression line equation.
x = mfgenders$e12mf
model = lm(formula = mfgenders$e14mf~x)
#shaded lines horizontally should be 0.7368584x + 1 and 0.7368584x - 1
test = as.data.frame(summary(model)$coefficients)
m = test$Estimate[2] #access our slope of regression line


#Create base plot and assign non significant colour boundary
col = c("not significant" = "black","antisense" = "blue2","lincRNA" = "khaki1","miRNA" = "green2","proc

g = ggplot(data = mfgenders) +
  geom_point(aes(x = e12mf, y = e14mf, colour = ifelse(e14mf < e12mf*m + .75 & e14mf > e12mf*m - .75, "
  #geom ribbon adds the shaded regions
  geom_ribbon(aes(x = e12mf,ymin = e12mf*m - .75, ymax = e12mf*m + .75), fill = "darkgray", alpha = 0.7
  #adds the legend name and colors
  scale_colour_manual(name = "Gene Type",  values = col)

#Adjust graph and axis scaling  ------------------------------------------------------------------------
  g = g + scale_x_continuous(limits = c(-6,6), breaks = seq(-6,6, by = 2)) + #limits defines boundaries
  scale_y_continuous(limits = c(-6,6), breaks = seq(-6,6, by = 2))  +
  theme_classic() + #Add theme
  theme(aspect.ratio=1) #forces plot to be square instead of rectangle shape

#Axis Titles   ----------------------------------------------------------------------------------------
  g = g +
  labs(title="Log2(f/m) values plotted by Embryonic Days ", y = "F/M Embryonic Day 14 values", x = "F/M
  theme(plot.title=element_text(size=16, #Customize text of the Title
                                face="bold",
                                family= "sans", #arial
                                color="black",
                                hjust=.5,
                                lineheight= 1.4),
            axis.title.x=element_text(vjust= 0, face = "bold", size=11), # X axis title
            axis.title.y=element_text(size=11, face = "bold"),  # Y axis title
            axis.text.x=element_text(size=9, angle = 0 ,vjust=.4),  # X axis text
            axis.text.y=element_text(size=9))

  #draw the lines to illustrate boundary for non significant
  g = g + geom_abline(slope = m, intercept = .75, col = "black", linetype = "solid") +
  geom_abline(slope = m, intercept = -.75, col = "black", linetype = "solid")

  g
```



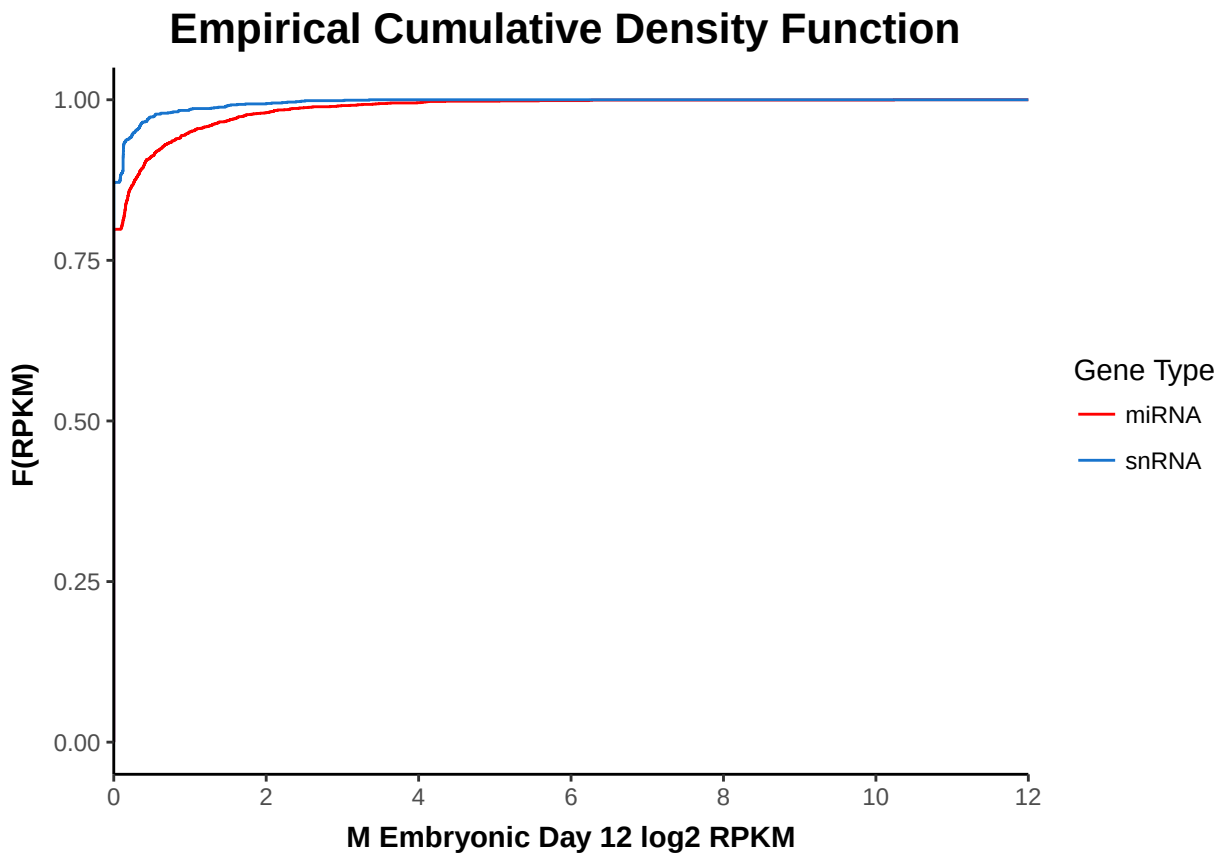

This is a scatterplot that compares

log2(f/m)e14 and log2(f/m)e12

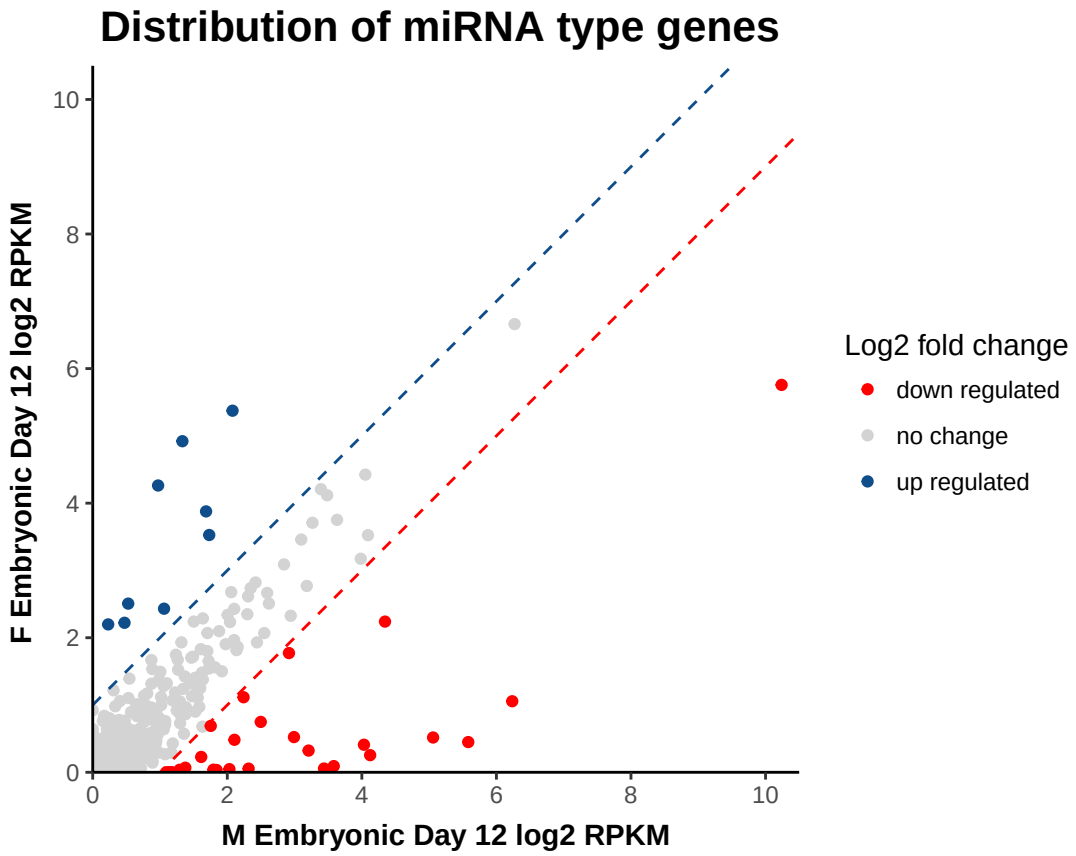
log2f - log2m of e14 on y axis and log2f - log2m of e12 on x axis




Emperical CDF



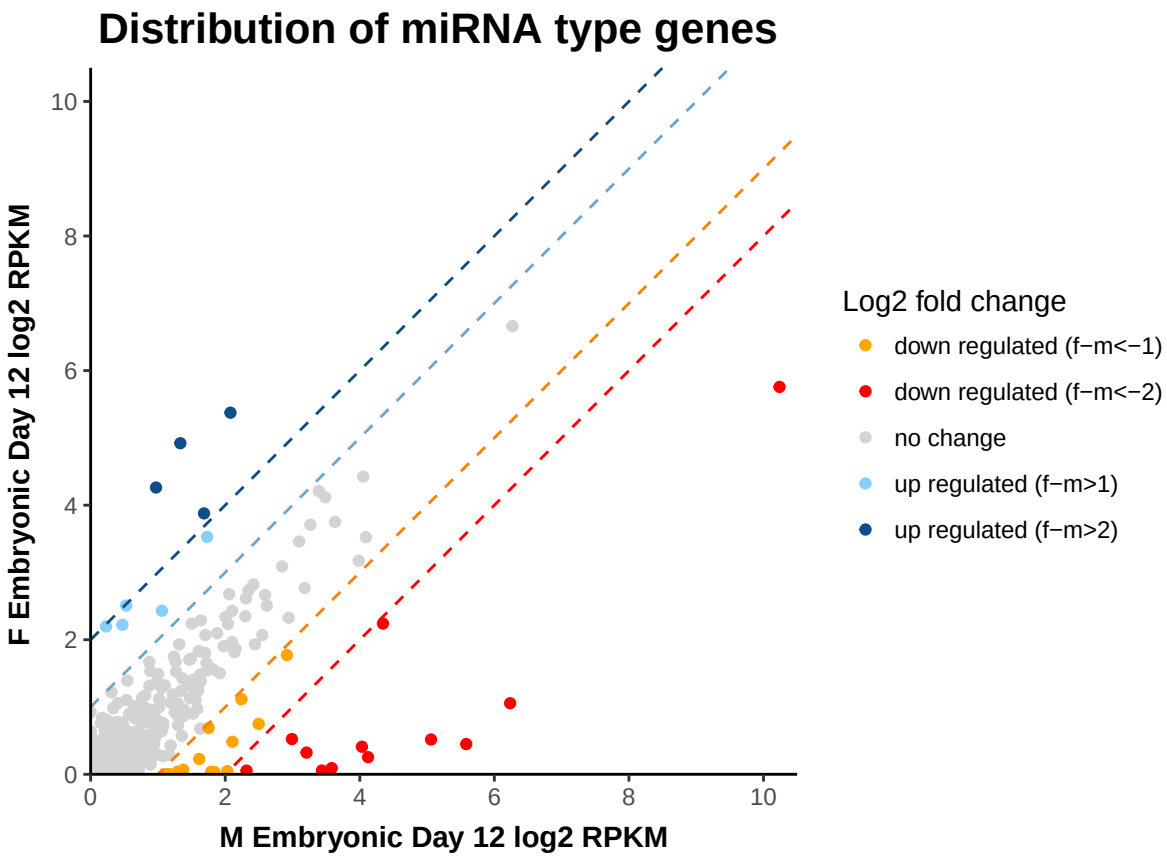

Here is an example of constructing/comparing multiple ECDF on one plot

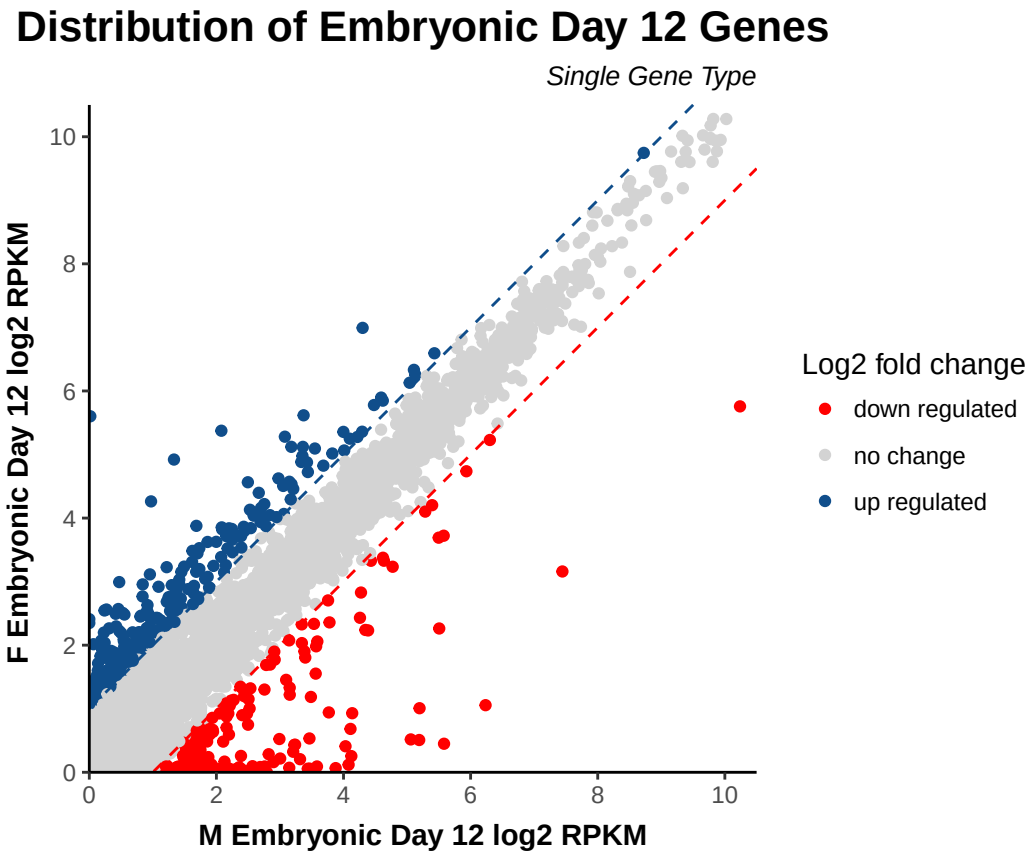



This is a scatterplot that compares the log2 RPKM values of genes between male and female populations. Points are colored if their fold change is significant, i.e. $|\log2(y\_i)-\log2(x\_i)| > 1$

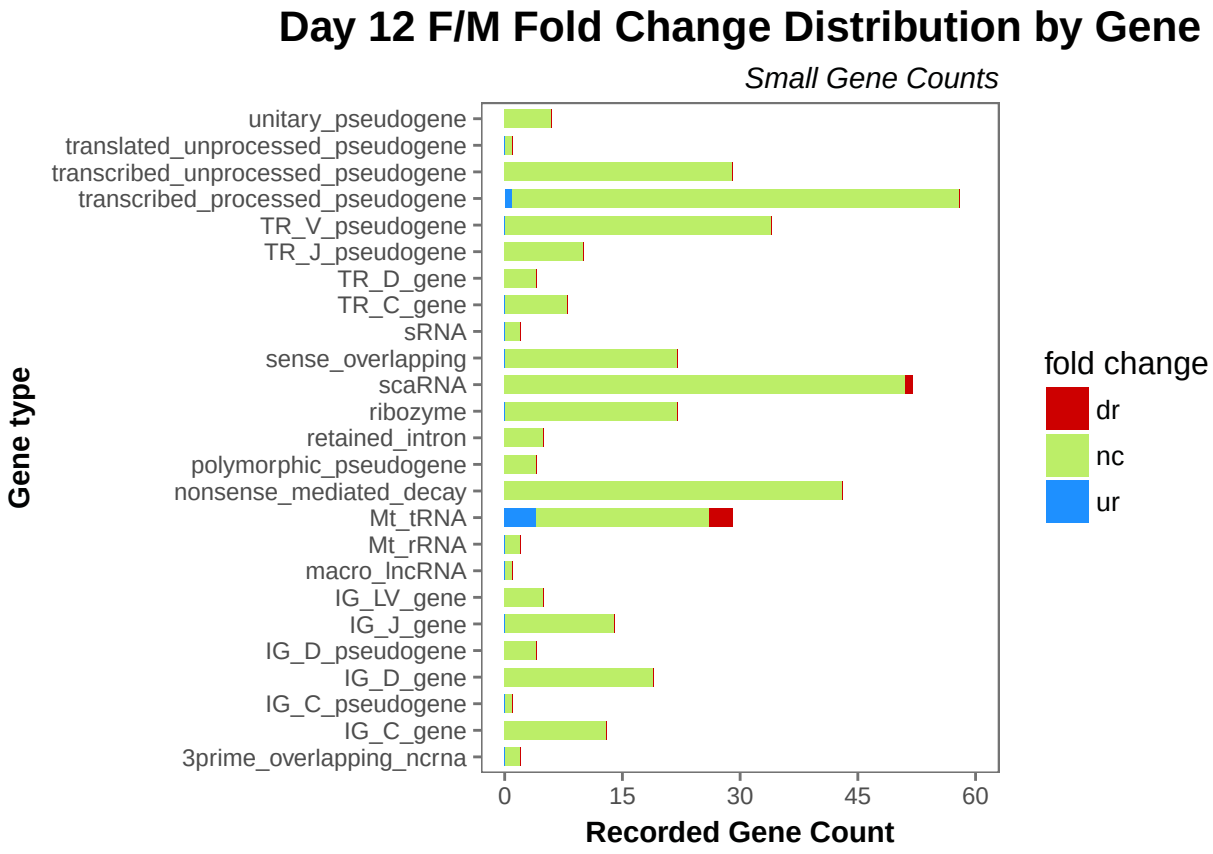
I am claiming that the initial value is the E12M value, final value is E12f value.

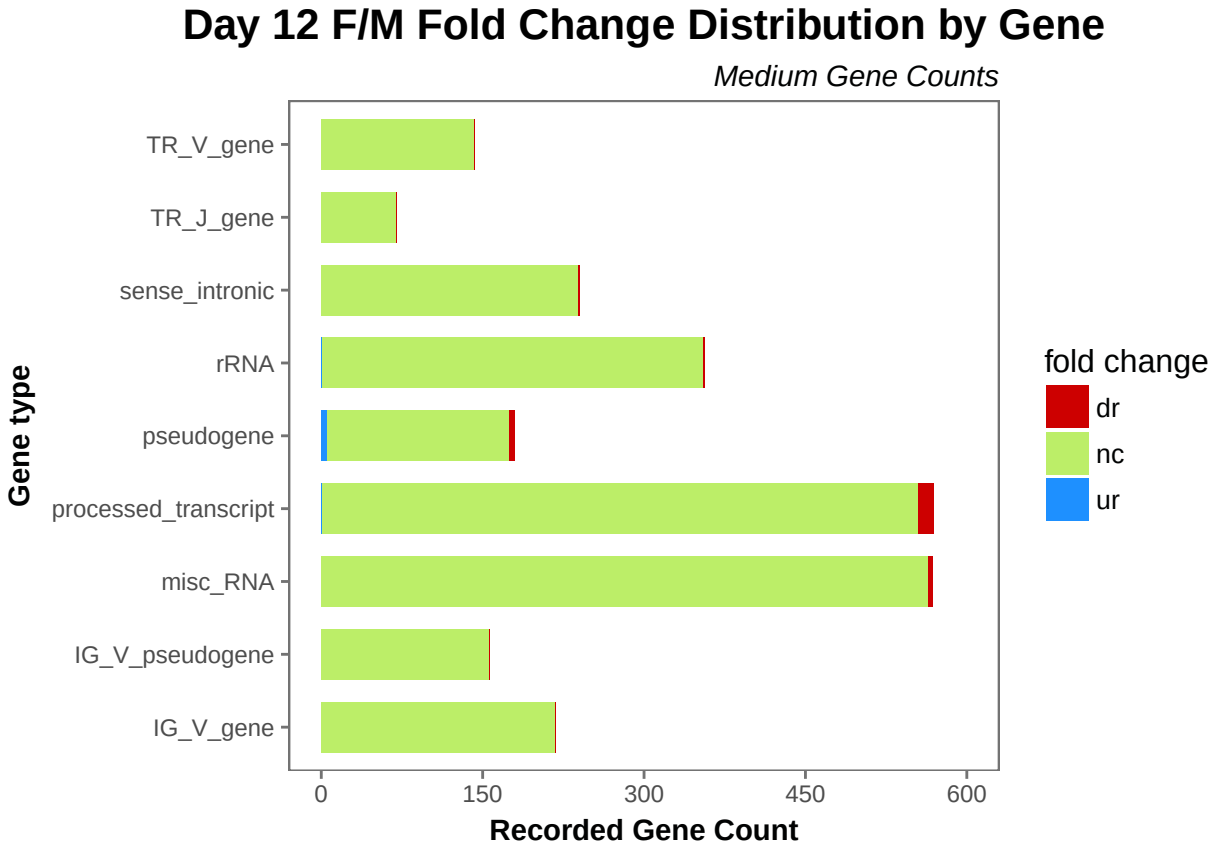
Fold Change is defined as E12F/E12M.

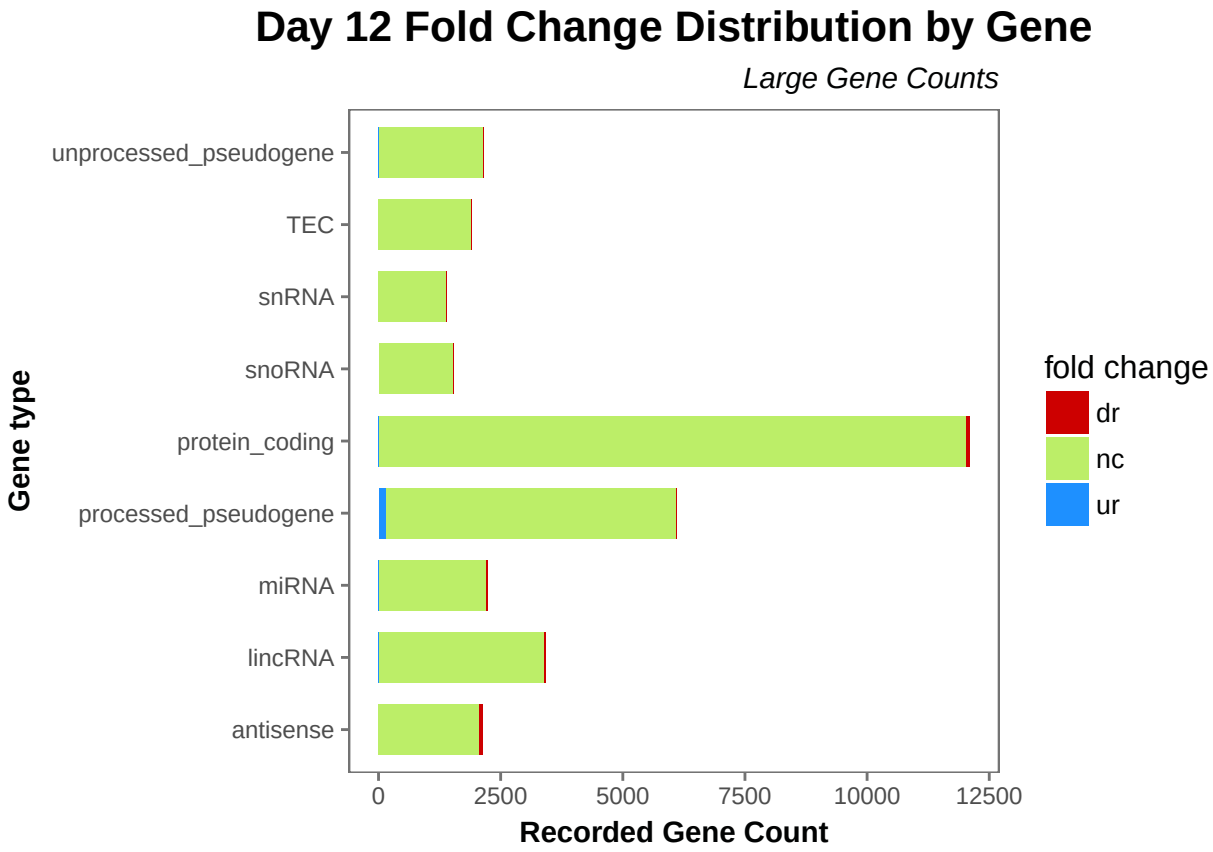

Distribution of miRNA type genes
F Embryonic Day 12 log2 RPKM
M Embryonic Day 12 log2 RPKM
0
2
4
6
8
10
Log2 fold change
down regulated (f−m<−1)
down regulated (f−m<−2)
no change
up regulated (f−m>1)
up regulated (f−m>2)



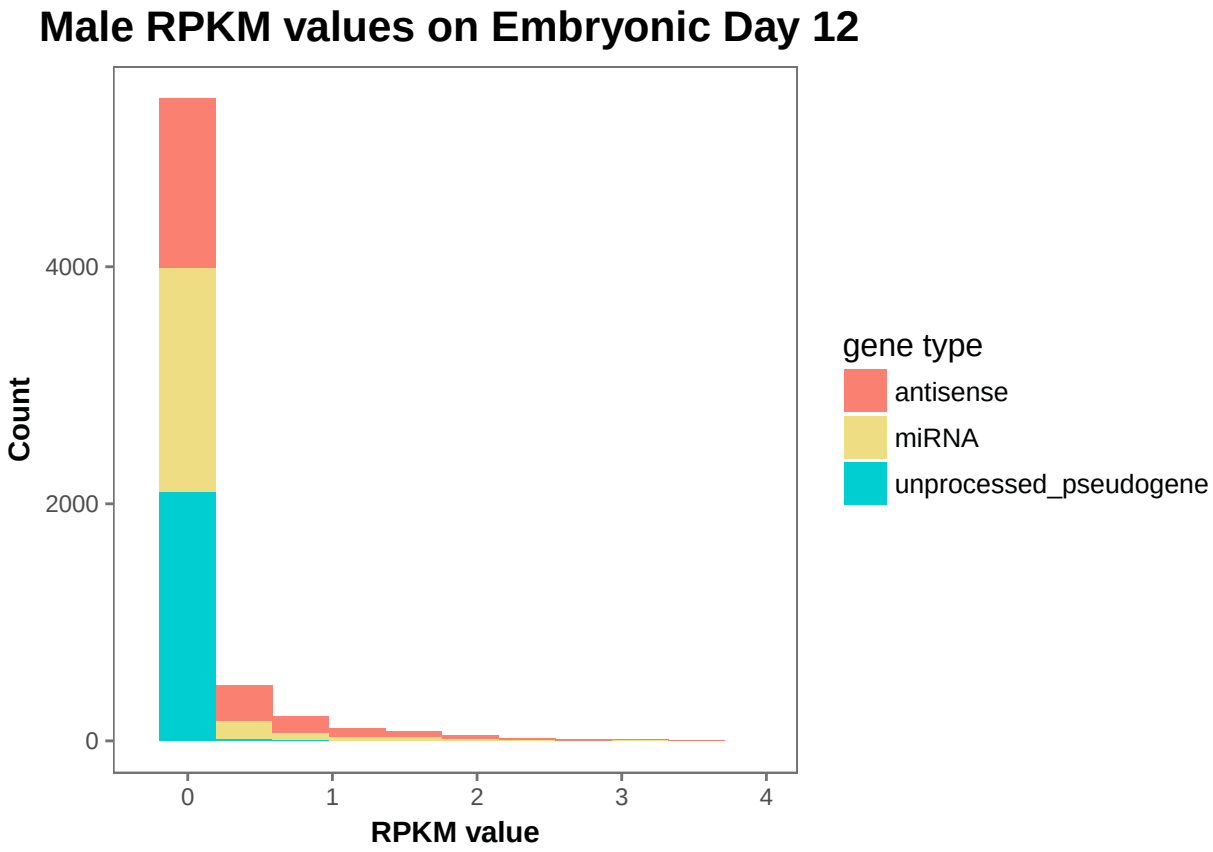

Here is a visual of the data that will be used to generate our stacked bar charts in the next step.


Day 12 F/M Fold Change Distribution by Gene
Small Gene Counts
Gene type
unitary_pseudogene
translated_unprocessed_pseudogene
transcribed_unprocessed_pseudogene
transcribed_processed_pseudogene
TR_V_pseudogene
TR_J_pseudogene
TR_D_gene
TR_C_gene
sRNA
sense_overlapping
scaRNA
ribozyme
retained_intron
polymorphic_pseudogene
nonsense_mediated_decay
Mt_tRNA
Mt_rRNA
macro_lncRNA
IG_LV_gene
IG_J_gene
IG_D_pseudogene
IG_D_gene
IG_C_pseudogene
IG_C_gene
3prime_overlapping_ncrna
0
15
30
45
60
Recorded Gene Count
fold change
dr
nc
ur

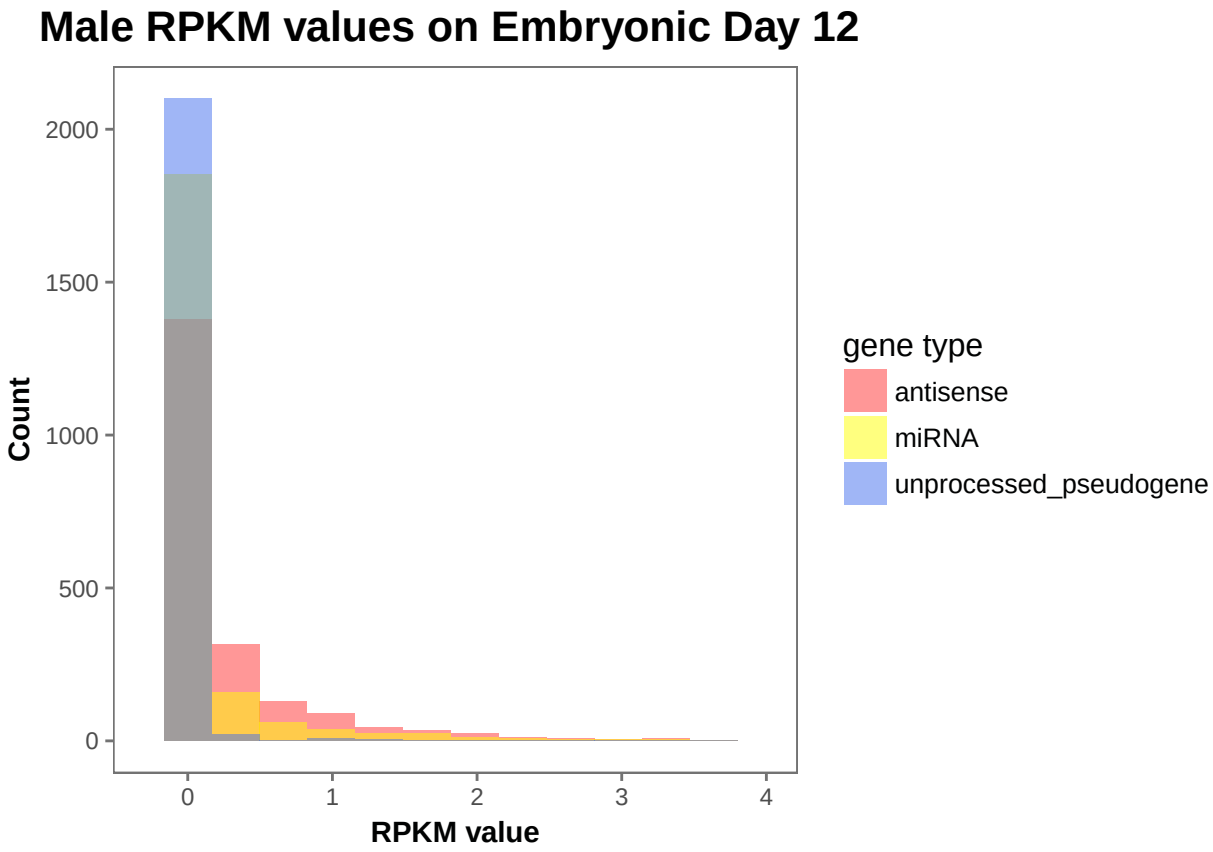

Day 12 F/M Fold Change Distribution by Gene
Medium Gene Counts
Gene type
TR_V_gene
TR_J_gene
sense_intronic
rRNA
pseudogene
processed_transcript
misc_RNA
IG_V_pseudogene
IG_V_gene
0
150
300
450
600
Recorded Gene Count
fold change
dr
nc
ur



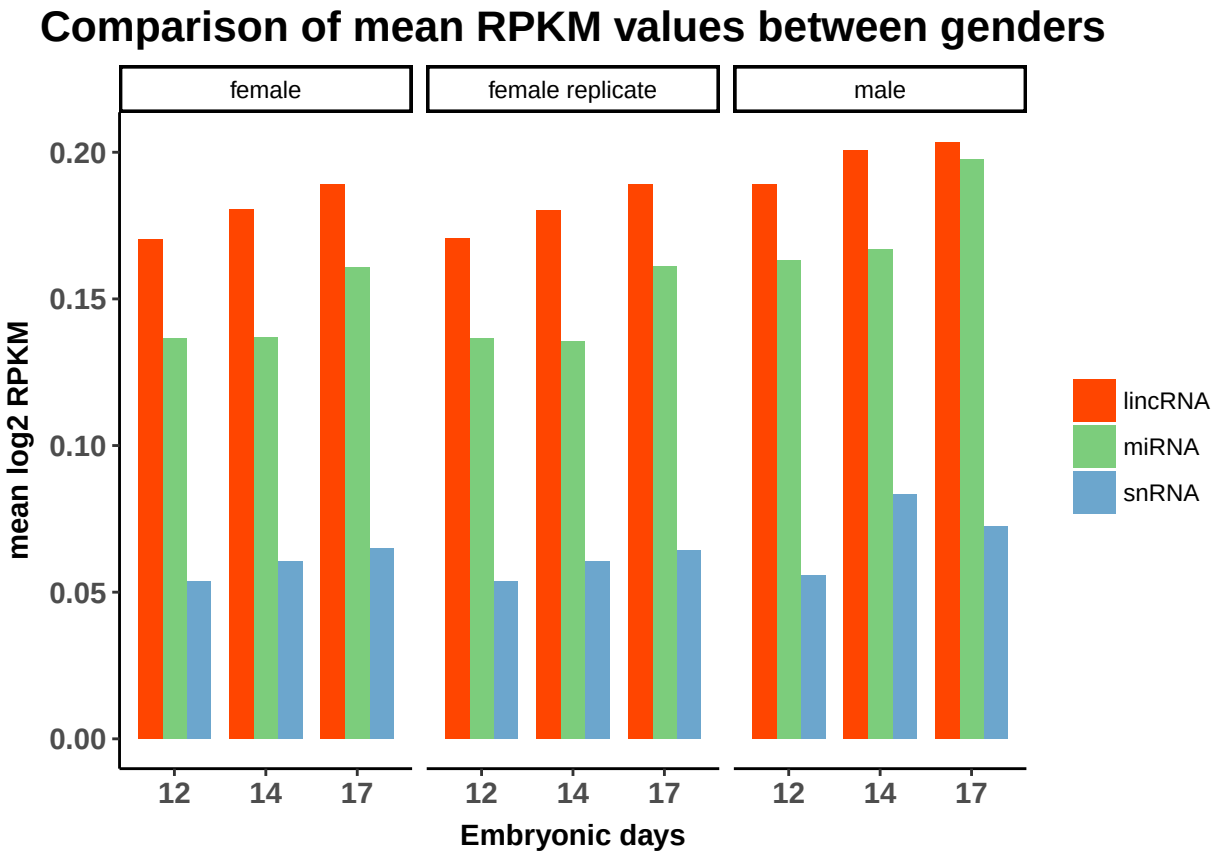

```
a = filter(onetgenders, type %in% c('unprocessed_pseudogene','miRNA','antisense'))
ggplot(a,aes(x=e12m, fill = type)) +
  #default position is stack. set to identity to have overlayed histograms
  geom_histogram(bins = 12, position = "stack") +
#bins sets number of bars in histogram.
 scale_x_continuous(limits = c(-.3,4), breaks = seq(0,4, by = 1))+
#limits defines boundaries of grid, breaks defines boundary of grid labels
 theme_few()+
 #Change color -----------------------------------
 scale_fill_manual(name = "gene type", values = c("salmon","lightgoldenrod","darkturquoise"))  +

labs(title="Male RPKM values on Embryonic Day 12 ", x = "RPKM value", y = "Count") +
  theme(plot.title=element_text(size=16,
                                face="bold",
                                family= "sans", #arial
                                color="black",
                                hjust=.5,
                                lineheight= 1.4),  # title
              plot.subtitle = element_text(family = "sans", hjust = 1, face = "italic"), #subtitle
        axis.title.x=element_text(vjust= 0, face = "bold", size=11), # X axis title
        axis.title.y=element_text(size=11, face = "bold"),  # Y axis title
        axis.text.x=element_text(size=9, angle = 0 ,vjust=.4),  # X axis text
        axis.text.y=element_text(size=9))
```



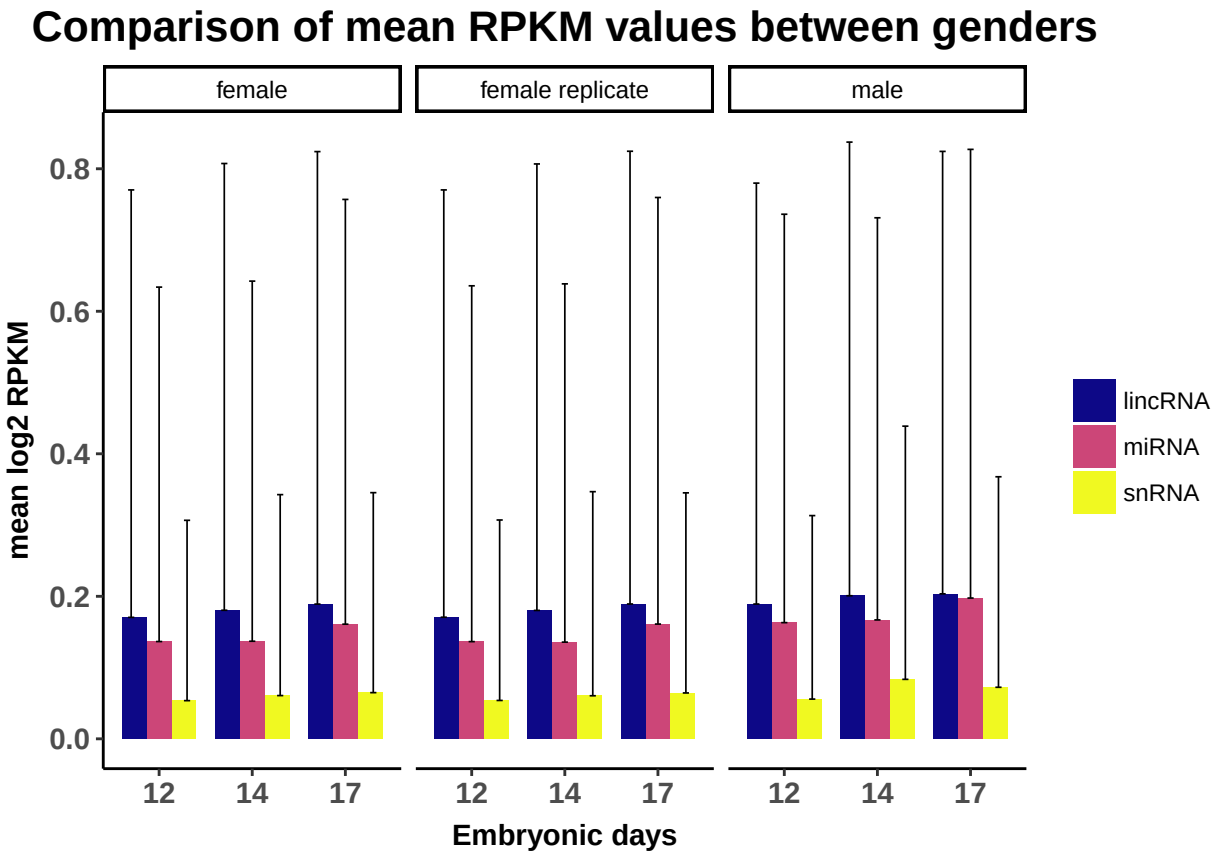

Histogrom (stacked)




Histogram(seperate)

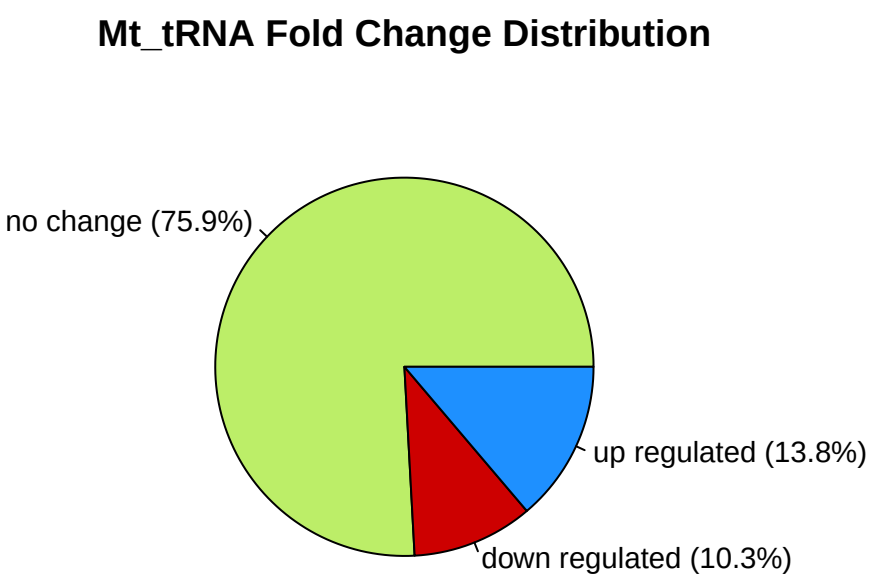



This bargraph compares the mean log2 RPKM values between genes of type miRNA, snRNA, and lincRNA, tested in three different population groups.




Same bargraph as above except with error lines.

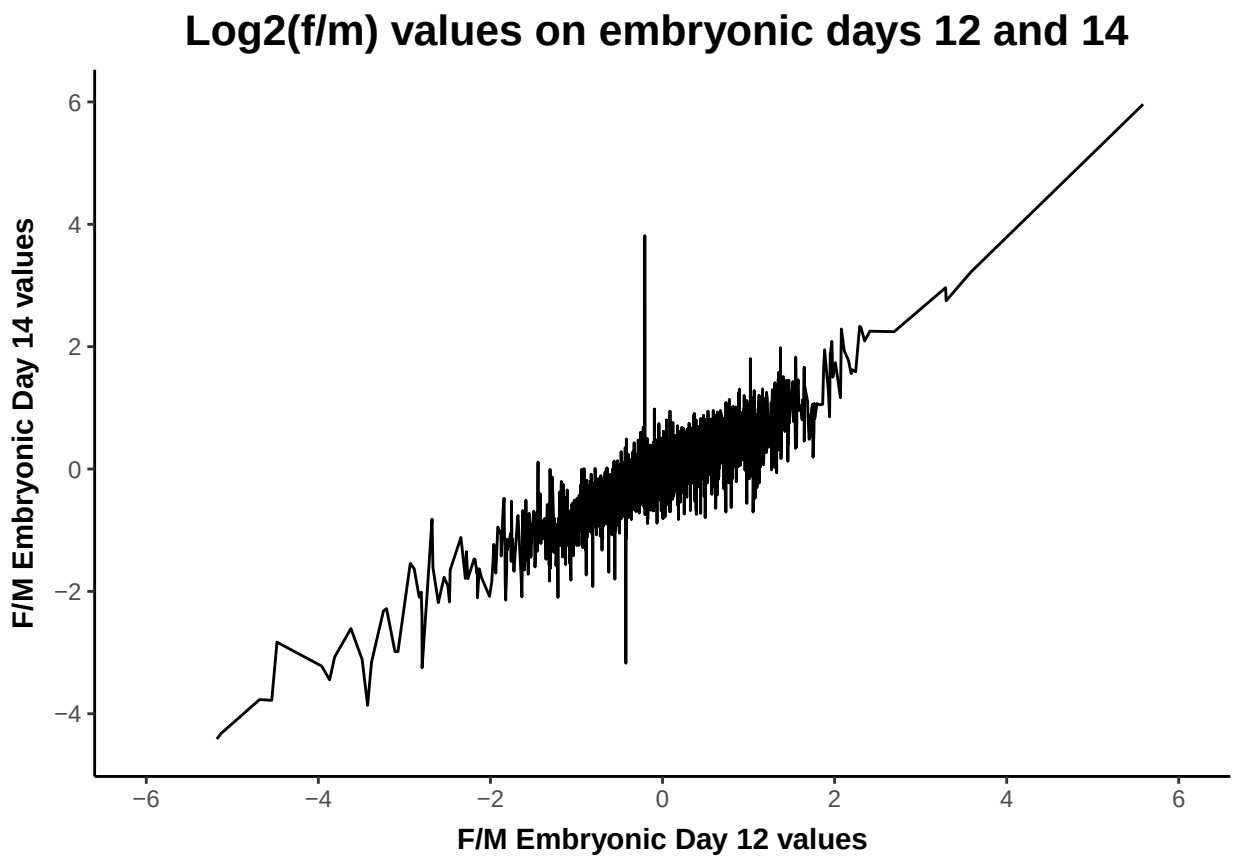
```
pies = filter(smallgathertypes,type == "Mt_tRNA")
#(,1) specifies round to tenth decimal
piepercent = round(pies$counts*100/sum(pies$counts),1)
pielabels = c("no change (", "down regulated (", "up regulated (")
percent = c("%)","%)","%)")

format = paste(pielabels,piepercent, percent, sep = "")

#cex is font size, labels is the text that is shown
pie(pies$counts,labels = format, main = "Mt_tRNA Fold Change Distribution", col = c("darkolivegreen2",
```



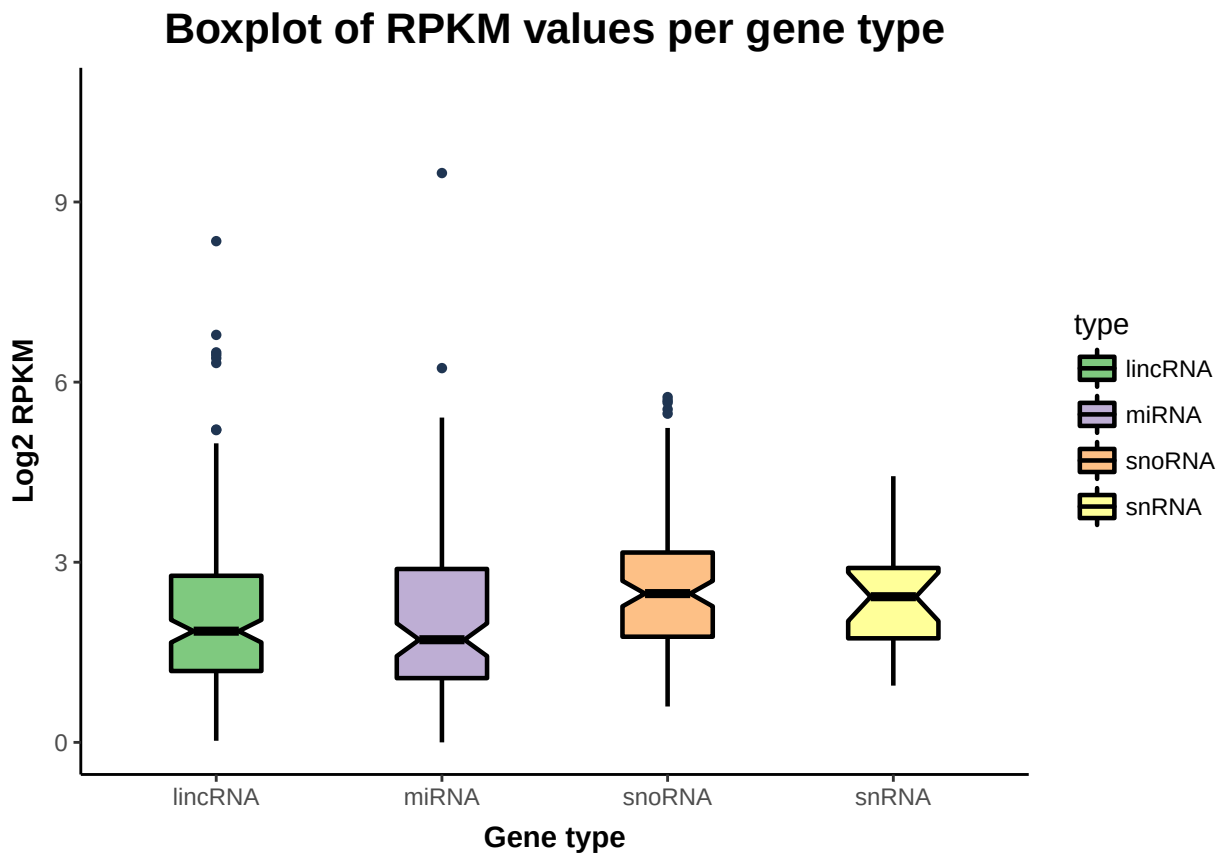

Pie Chart that illustrates the fold change in Mt_tRNA type genes

```
gglinegraph = ggplot(mfgenders, aes(x = e12mf, y = e14mf)) + #base ggplot
  #in geom_line, can adjust colour, linetype, and size.
  geom_line(aes(group = 1)) #adds the line graph
  #can add points to graph, and can customize points
  #geom_point(colour = "color", size = 3, shape = 21, fill = "white")

gglinegraph = gglinegraph +
 scale_x_continuous(limits = c(-6,6), breaks = seq(-6,6, by = 2)) + #limits defines boundaries of grid,
  scale_y_continuous(limits = c(-4.5,6), breaks = seq(-4,6, by = 2))  +
  theme_classic() + #Add theme

#Axis Titles   ------------------------------------------------------------------------------------
  labs(title="Log2(f/m) values on embryonic days 12 and 14 ", y = "F/M Embryonic Day 14 values", x = "F/
  theme(plot.title=element_text(size=16, #Customize text of the Title
                                    face="bold",
                                    family= "sans", #arial
                                    color="black",
                                    hjust=.5,
                                    lineheight= 1.4),
            axis.title.x=element_text(vjust= 0, face = "bold", size=11), # X axis title
            axis.title.y=element_text(size=11, face = "bold"),  # Y axis title
            axis.text.x=element_text(size=9, angle = 0 ,vjust=.4),  # X axis text
            axis.text.y=element_text(size=9))
```

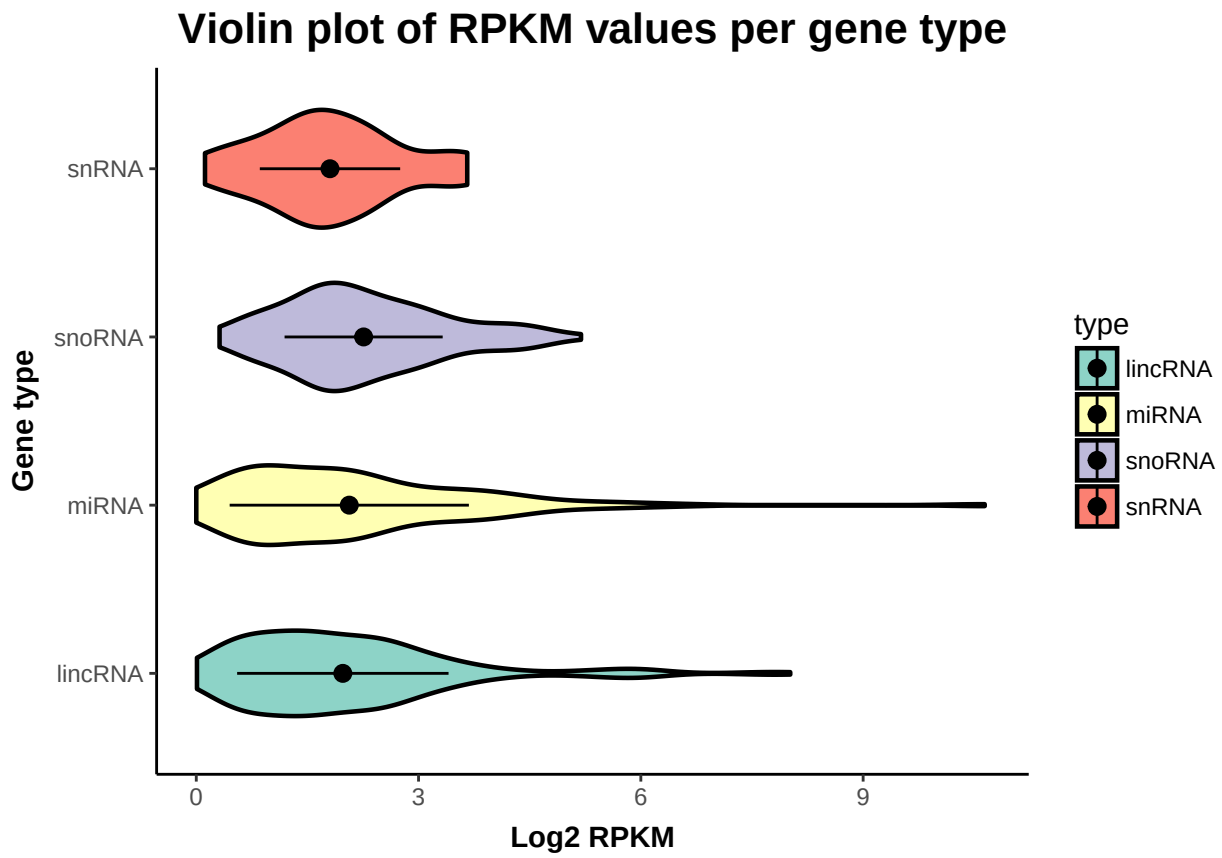
```
gglinegraph
```




Simple line graph depicting e12mf on x axis and e14mf on y axis




Box and Whisker Plot




Example of a violin plot

Dendrogram / Heatmap

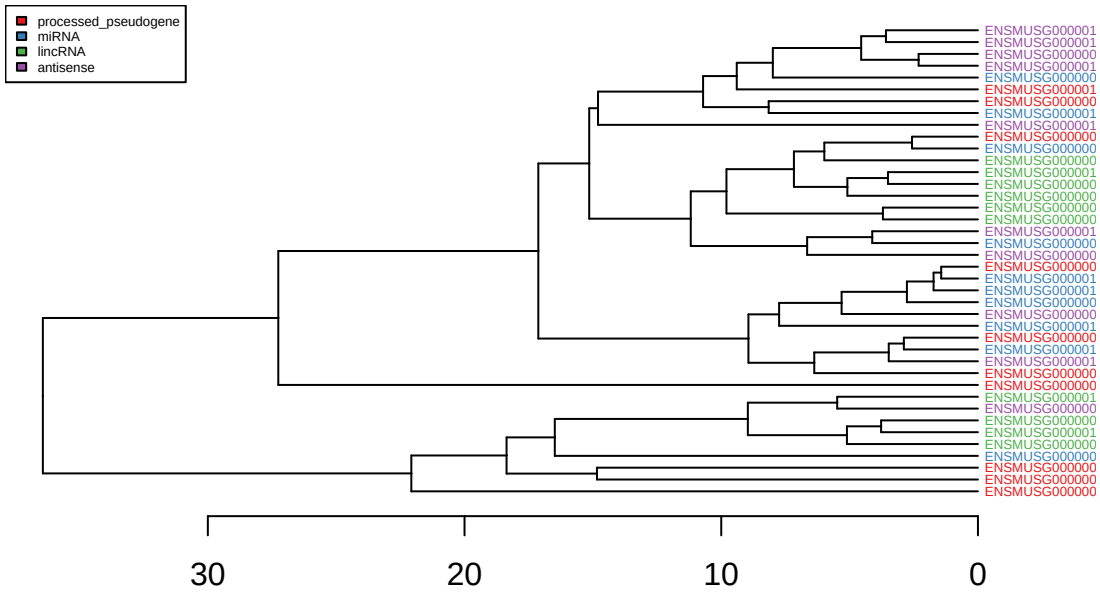
For the purpose of clustering based on type, I removed genes that are classified as multiple types. For simplicity, I only look at the 5 genes with the highest sampled counts.

Calculate Variance of Each Row, and apply the quantile function on that to get rows with highest variance

```
genders.log.typ = genders.log
genders.log.typ = genders.log.typ[- grep(",|transcribed|unprocessed", genders.log.typ$type),]
genders.log.typ = filter(genders.log.typ, grepl("antisense|miRNA|processed_pseudogene|lincRNA",type))
#create row names to allow for matching later
rownames(genders.log.typ) = 1:nrow(genders.log.typ)
#create a dataframe with just numerical values to convert to matrix
numtyp = genders.log.typ[,6:20]
numtyp = as.matrix(numtyp) #Generate a matrix of only the RPKM numerical counts.

#Varlist is a list of the variance of each row.
Varlist = apply(numtyp,MARGIN = 1,var)  #MARGIN = 1 applies the var func over rows, 2 for columns.
Varlist = as.matrix(Varlist)
qt = quantile(Varlist, probs = c( .20,.95)) #only look at genes with top 1% of variance. (first option
#We generate a True/False vector (whether the row's variance is > top 1% of variance) for every row of
rows = apply(numtyp, 1, function(x) any (var(x) > qt[2]))
#we filter our dataframe to only the rows that match with true
genders.log.typ = genders.log.typ[rows,]

genders.log.type = genders.log.typ[FALSE,] #create an empty dataframe using old column names
```

```
list = c("antisense","miRNA","lincRNA","processed_pseudogene")

#This for loop adds random samples of certain gene type to genders.log.type
for (i in list){
temp = subset(genders.log.typ,type == i)
genders.log.type = rbind(genders.log.type,temp[sample(10), ])
}

#These commands are for generating our dendogram.
genders.log.type = as.data.frame(genders.log.type)
genders.log.type$type =as.factor(genders.log.type$type) #need to make into factor type
#add row names to be depicted on dendogram by default.
rownames(genders.log.type) = genders.log.type$gid
```

```
numgenders = genders.log.type[,6:20]

#method = manhattan, euclidian, maximum, ....
d_genders <- dist(numgenders,method = "manhattan")
#method = "average", "single"
hc_genders <- hclust(d_genders, method = "average")
types <- rev(levels(genders.log.type[,4]))

#hang = -1 specifies leaves should be at same length
dendrogram = as.dendrogram(hc_genders,hang = -1)#set hang to other number to have diff length

#this is if you would like to color the branches as well
#dendrogram = color_branches(col = plasma, dendrogram,k=6)

labels_colors(dendrogram) <-
   #select 4 colors in color brewer
   brewer.pal(4,"Set1")[sort_levels_values(
      as.numeric(genders.log.type[,4])[order.dendrogram(dendrogram)]
   )]

#changes the leafnames
#labels(dendrogram) = paste(as.character(genders.log.type[,4])[order.dendrogram(dendrogram)],
#                           "(",labels(dendrogram),")",
#                           sep = "")


dendrogram <- assign_values_to_leaves_nodePar(dendrogram, 0.5, "lab.cex")
dendrogram = set(dendrogram,"labels_cex",0.4)

plot(dendrogram,
     main = "Clustering Gene Expression Data Based on Type",
     horiz = TRUE, nodePar = list(cex = .003))
legend(cex = .4,"topleft",legend = types,fill = brewer.pal(5,"Set1"))
```



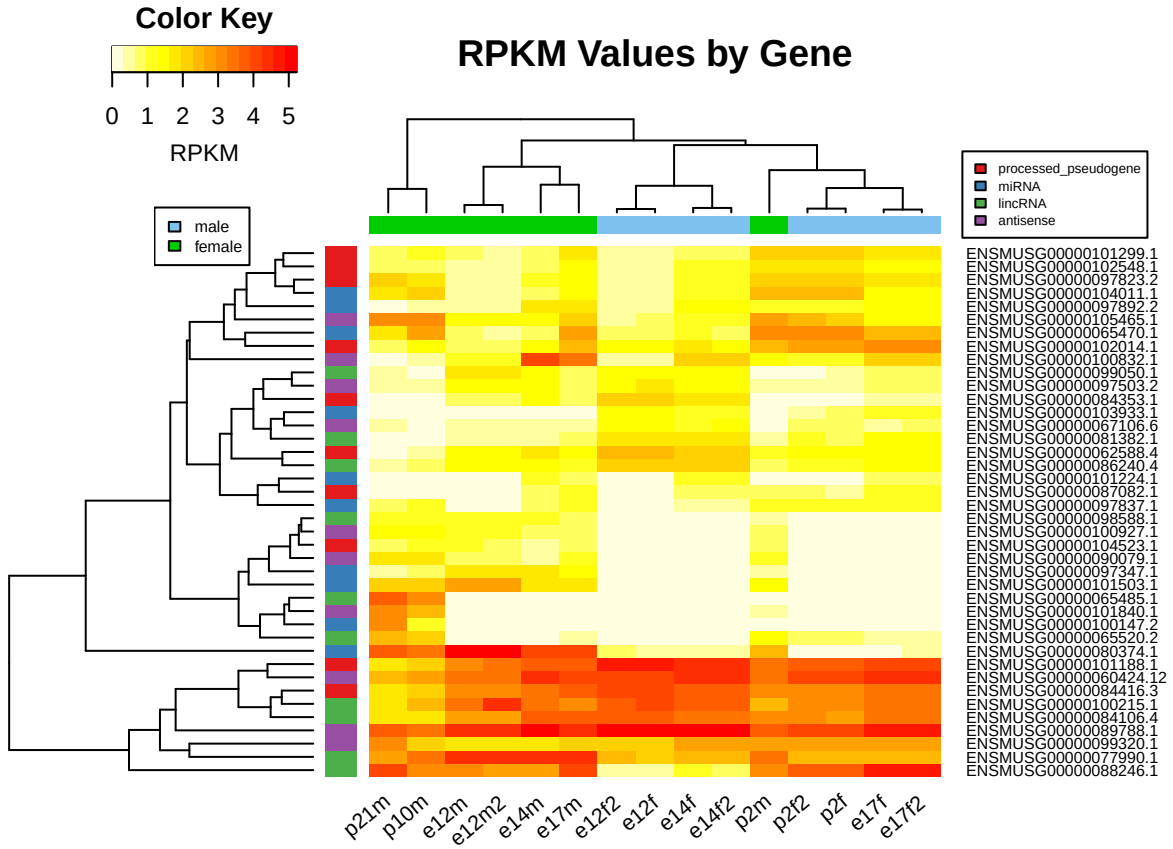

Here is the dendogram

```
#assign group colors based on gender.
colnames(numgenders)
```

```
##  [1] "e12m"  "e14m"  "e17m"  "p2m"   "e12f"  "e14f"  "e17f"  "p2f"
##  [9] "p10m"  "p21m"  "e12f2" "e14f2" "e17f2" "p2f2"  "e12m2"
```

```
color.map = function(gender) { if (grepl('f',gender)) "skyblue2" else "green3" }
groupgenders = unlist(lapply(colnames(numgenders),color.map))


heatmap.2(
 main = "RPKM Values by Gene",
 as.matrix(numgenders),
 srtCol = 40, #column name angle
 Rowv = dendrogram,
 #dendrogram = "row", #hides the column dendogram.
 cexRow = .7, #sets font size for row
 #Colv = NA, #uncomment this if you don't want columns to be sorted.
 ColSideCol = groupgenders,
 scale = "none", #turns off default row normalization
 trace="none", #gets rid of green lines in heatmap
 margins =c(3,12),   #figure margins
 key.xlab = "RPKM",
 denscol = "grey",
 density.info = "none", #gets rid of histogram drawn in scale
```

```
 RowSideColors = rev(labels_colors(dendrogram)), # to add nice colored strips
 col=rev(heat.colors(16)) #16 is the number of different colors in our spectrum
 )

 legend(cex = .5, xpd = TRUE, x = 0, y = .88 ,legend = c("male","female"), c("skyblue2","green3"))
 #need to add legend for column colors
legend(cex = .45, xpd = TRUE, x = .80, y = .99 ,legend = types, brewer.pal(5,"Set1"))
```




Here is the heatmap associated with that dendogram. Also has columns clustered (with group gender coloring)

Additional Sources

For modifying legend font/positioning : http://www.cookbook-r.com/Graphs/Legends_(ggplot2)/

Pretty Heatmap http://wiki.bits.vib.be/index.php/Use_pheatmap_to_draw_heat_maps_in_R

Dendogram https://plot.ly/ggplot2/ggdendro-dendrograms/ https://stackoverflow.com/questions/43794870/plotting-a-clustered-heatmap-with-dendrograms-using-rs-plotly http://slowkow.com/notes/heatmap-tutorial/ https://www2.warwick.ac.uk/fac/sci/moac/people/students/peter_cock/r/heatmap/ http://www.molecularecologist.com/2013/08/making-heatmaps-with-r-for-microbiome-analysis/