Coronavirus Liver and Blood Capillary Samples

Janis Corona

2/8/2020

These samples are the headers added from three Gene Expression Omnibus studies at

- ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE89166
- ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE89160
- ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100509

The first two studies are part of the same study that used human liver tumor samples in vitro to compare the effects of the coronavirus over time. The third study used human microvascular blood capillaries in vitro to study the effects of the coronavirus over time.

In the first two studies that used the liver tumor samples to examine the effects of the coronavirus in vitro, there were four groups inoculated or treated with the active coronavirus and four groups not inoculated with the active coranavirus, and two samples that were treated with heat inactivated coronavirus, and two samples that were treated with active coronavirus and IL-1alpha to see the gene expression changes over one hour's time.

In the the third study that used blood capillaries, there were five samples followed over a 0,12,24,36, and 48 hour time intervals in groups A,B,C,D, and E that compared the time interval values of screening for changes in microarray analysis with a mock group of the same.

This following data is the data of all genes in common between these three studies, cleaned to remove missing values and with the attached gene symbols from the GEO platform for the probe IDs.

The libraries used for this script are tidyr, dplyr, and ggplot2.

```
library(ggplot2)
library(dplyr)
library(tidyr)

both <- read.csv('both_clean_liver_capillary_CoV.csv', sep=',', header=TRUE,
                 na.strings=c('',' '))

dim(both)

## [1] 21754    63

colnames(both)

##  [1] "GENE_SYMBOL"
##  [2] "LiverTumorSamples.GSM2359851_CoV1"
##  [3] "LiverTumorSamples.GSM2359853_CoV2"
##  [4] "LiverTumorSamples.GSM2359910_CoV3"
##  [5] "LiverTumorSamples.GSM2359913_CoV4"
##  [6] "LiverTumorSamples.GSM2359850_ctrl1"
##  [7] "LiverTumorSamples.GSM2359852_ctrl2"
##  [8] "LiverTumorSamples.GSM2359911_ctrl3"
##  [9] "LiverTumorSamples.GSM2359914_ctrl4"
## [10] "LiverTumorSamples.GSM2359912_Il1"
## [11] "LiverTumorSamples.GSM2359917_IL2"
## [12] "LiverTumorSamples.GSM2359915_inactiveHeatCoV1"
## [13] "LiverTumorSamples.GSM2359916_inactiveHeatCoV2"
## [14] "capillarySamples.GSM2685693_MERS_CoV_0hr_A"
## [15] "capillarySamples.GSM2685694_MERS_CoV_0hr_B"
## [16] "capillarySamples.GSM2685695_MERS_CoV_0hr_C"
## [17] "capillarySamples.GSM2685696_MERS_CoV_0hr_D"
## [18] "capillarySamples.GSM2685697_MERS_CoV_0hr_E"
## [19] "capillarySamples.GSM2685698_ctrl_0hr_A"
## [20] "capillarySamples.GSM2685699_ctrl_0hr_B"
## [21] "capillarySamples.GSM2685700_ctrl_0hr_C"
## [22] "capillarySamples.GSM2685701_ctrl_0hr_D"
## [23] "capillarySamples.GSM2685702_ctrl_0hr_E"
## [24] "capillarySamples.GSM2685703_MERS_CoV_12hr_A"
## [25] "capillarySamples.GSM2685704_MERS_CoV_12hr_B"
## [26] "capillarySamples.GSM2685705_MERS_CoV_12hr_C"
## [27] "capillarySamples.GSM2685706_MERS_CoV_12hr_D"
## [28] "capillarySamples.GSM2685707_MERS_CoV_12hr_E"
## [29] "capillarySamples.GSM2685708_ctrl_12hr_A"
## [30] "capillarySamples.GSM2685709_ctrl_12hr_B"
## [31] "capillarySamples.GSM2685710_ctrl_12hr_C"
## [32] "capillarySamples.GSM2685711_ctrl_12hr_D"
## [33] "capillarySamples.GSM2685712_ctrl_12hr_E"
## [34] "capillarySamples.GSM2685713_MERS_CoV_24hr_A"
## [35] "capillarySamples.GSM2685714_MERS_CoV_24hr_B"
## [36] "capillarySamples.GSM2685715_MERS_CoV_24hr_C"
## [37] "capillarySamples.GSM2685716_MERS_CoV_24hr_D"
## [38] "capillarySamples.GSM2685717_MERS_CoV_24hr_E"
## [39] "capillarySamples.GSM2685718_ctrl_24hr_A"
## [40] "capillarySamples.GSM2685719_ctrl_24hr_B"
## [41] "capillarySamples.GSM2685720_ctrl_24hr_C"
## [42] "capillarySamples.GSM2685721_ctrl_24hr_D"
## [43] "capillarySamples.GSM2685722_ctrl_24hr_E"
## [44] "capillarySamples.GSM2685723_MERS_CoV_36hr_A"
## [45] "capillarySamples.GSM2685724_MERS_CoV_36hr_B"
## [46] "capillarySamples.GSM2685725_MERS_CoV_36hr_C"
## [47] "capillarySamples.GSM2685726_MERS_CoV_36hr_D"
## [48] "capillarySamples.GSM2685727_MERS_CoV_36hr_E"
## [49] "capillarySamples.GSM2685728_ctrl_36hr_A"
## [50] "capillarySamples.GSM2685729_ctrl_36hr_B"
## [51] "capillarySamples.GSM2685730_ctrl_36hr_C"
## [52] "capillarySamples.GSM2685731_ctrl_36hr_D"
## [53] "capillarySamples.GSM2685732_ctrl_36hr_E"
## [54] "capillarySamples.GSM2685733_MERS_CoV_48hr_A"
## [55] "capillarySamples.GSM2685734_MERS_CoV_48hr_B"
## [56] "capillarySamples.GSM2685735_MERS_CoV_48hr_C"
## [57] "capillarySamples.GSM2685736_MERS_CoV_48hr_D"
## [58] "capillarySamples.GSM2685737_MERS_CoV_48hr_E"
## [59] "capillarySamples.GSM2685738_ctrl_48hr_A"
## [60] "capillarySamples.GSM2685739_ctrl_48hr_B"
## [61] "capillarySamples.GSM2685740_ctrl_48hr_C"
## [62] "capillarySamples.GSM2685741_ctrl_48hr_D"
## [63] "capillarySamples.GSM2685742_ctrl_48hr_E"
```

Lets group the samples that are our columns with descriptive and GEO ID names into their respective groups, get the fold change between the controls from those groups, attach to the original data table, both, as a different names, then order by the genes that have the most fold change then the least fold change. Take the first 100 genes from both lists, combine into one table of 200 genes and the samples with their fold change values ordered, make into a transposed data frame so that the samples are the rows, the stats removed, and the 200 genes are the header columns to save as a machine learning ready file.

Liver tumor study control and CoV treated. Also, the IL-alpha treated and the inactive CoV treated tables are in this code block.

```
names <- both$GENE_SYMBOL

liverCtrl <- both[,c(6:9)]
row.names(liverCtrl) <- names

liverCoV <- both[,c(2:5)]
row.names(liverCoV) <- names

liverIL <- both[,10:11]
row.names(liverIL) <- names

liverIACoV <- both[,12:13]
row.names(liverIACoV) <- names
```

Get the row means of those liver samples groups each.

```
liverCtrl$CtrlMeanLvr <- rowMeans(liverCtrl)
liverCoV$CoVMeanLvr <- rowMeans(liverCoV)
```

```
liverIL$ILMeanLvr <- rowMeans(liverIL)
liverIACoV$IACoVMeanLvr <- rowMeans(liverIACoV)
```

Get the fold change values of those states as a ratio to the control group values.

```
fold1 <- as.data.frame(cbind(liverCtrl$CtrlMeanLvr,liverCoV$CoVMeanLvr,liverI
L$ILMeanLvr,
               liverIACoV$IACoVMeanLvr))
row.names(fold1) <- names
colnames(fold1) <- c('CtrlMeanLvr','CoVMeanLvr','ILMeanLvr','IACoVMeanLvr')

fold1$FC_CoV <- fold1$CoVMeanLvr/fold1$CtrlMeanLvr
fold1$FC_IL <- fold1$ILMeanLvr/fold1$CtrlMeanLvr
fold1$FC_IACov <- fold1$IACoVMeanLvr/fold1$CtrlMeanLvr
```

Most expressed in liver samples by fold change of the Coronavirus, inactive CoronaVirus, and the IL-alpha treated Coronavirus as tables.

```
mostCoV <- fold1[order(fold1$FC_CoV, decreasing = TRUE)[0:100],]
mostIL <- fold1[order(fold1$FC_IL, decreasing = TRUE)[0:100],]
mostIACoV <- fold1[order(fold1$FC_IACov, decreasing = TRUE)[0:100],]
```

Least expressed in liver samples by fold change of the Coronavirus, inactive CoronaVirus, and the IL-alpha treated Coronavirus as tables.

```
leastCoV <- fold1[order(fold1$FC_CoV, decreasing = FALSE)[0:100],]
leastIL <- fold1[order(fold1$FC_IL, decreasing = FALSE)[0:100],]
leastIACoV <- fold1[order(fold1$FC_IACov, decreasing = FALSE)[0:100],]
```

Gene Expressions with most changes in the liver samples.

```
changes <- rbind(mostCoV,mostIL,mostIACoV,leastCoV,leastIL,leastIACoV)
Changes <- changes[!duplicated(row.names(changes)),]
length(unique(row.names(Changes)))

## [1] 600
```

Get the magnitude of the fold change genes’ row means.

```
Changes$MagnitudeFCs <- abs(rowMeans(Changes[,5:7]))
```

Combine this to the samples data for the liver tumor group.

```
Changes$Gene <- row.names(Changes)
combined1 <- merge(both, Changes, by.x='GENE_SYMBOL', by.y='Gene')

combined2 <- combined1[order(combined1$MagnitudeFCs, decreasing=TRUE),]

CombinedLiver <- combined2[c(0:100,354:453),]
```

Machine Learning data for liver samples with 200 genes in the group of most gene expression changes.

```r
names1 <- CombinedLiver$GENE_SYMBOL
names2 <- colnames(CombinedLiver)
row.names(CombinedLiver) <- names1

Combo_lvr_ML <- as.data.frame(t(CombinedLiver))

colnames(Combo_lvr_ML) <- gsub('-','_',colnames(Combo_lvr_ML))
Combo1 <- Combo_lvr_ML[c(2:63),] #remove stats of fold change values and gene symbol row
```

Lets add a class field called Class_Type to use machine learning on predicting class with these 200 genes and 62 mixed samples of capillary and liver tumor both inoculated with Coronavirus.

```r
a <- rep('liver_CoV', 4)
b <- rep('liver_Ctrl',4)
c <- rep('liver_CoV_IL',2)
d <- rep('liver_IA_CoV',2)
e <- rep('capillary_CoV_0hr',5)
f <- rep('capillary_Ctrl_0hr',5)
g <- rep('capillary_Cov_12hr',5)
h <- rep('capillary_Ctrl_12hr',5)
i <- rep('capillary_Cov_24hr',5)
j <- rep('capillary_Ctrl_24hr',5)
k <- rep('capillary_Cov_36hr',5)
l <- rep('capillary_Ctrl_36hr',5)
m <- rep('capillary_Cov_48hr',5)
n <- rep('capillary_Ctrl_48hr',5)

type <- as.data.frame(c(a,b,c,d,e,f,g,h,i,j,k,l,m,n))
colnames(type) <- 'Class_Type'
row.names(type) <- row.names(Combo1)

Combo2 <- cbind(type,Combo1)
```

Write this ML ready file to csv.

```r
write.csv(Combo2, 'ML_ready_CoV_14_classes.csv', row.names=TRUE)
```

Make a separate ML ready file with a smaller set of classes to classify by liver or capillary and control or CoronaVirus

```r
a <- rep('liver', 4)
b <- rep('liver',4)
c <- rep('liver',2)
d <- rep('liver',2)
e <- rep('capillary',5)
f <- rep('capillary',5)
g <- rep('capillary',5)
h <- rep('capillary',5)
i <- rep('capillary',5)
```

```r
j <- rep('capillary',5)
k <- rep('capillary',5)
l <- rep('capillary',5)
m <- rep('capillary',5)
n <- rep('capillary',5)

type <- as.data.frame(c(a,b,c,d,e,f,g,h,i,j,k,l,m,n))
colnames(type) <- 'Class_Type'
row.names(type) <- row.names(Combo1)

Combo3 <- cbind(type,Combo1)


write.csv(Combo3, 'ML_ready_CoV_2_classes.csv', row.names=TRUE)

a <- rep('CoV', 4)
b <- rep('Ctrl',4)
c <- rep('CoV_IL',2)
d <- rep('IA_CoV',2)
e <- rep('CoV',5)
f <- rep('Ctrl',5)
g <- rep('Cov',5)
h <- rep('Ctrl',5)
i <- rep('Cov',5)
j <- rep('Ctrl',5)
k <- rep('Cov',5)
l <- rep('Ctrl',5)
m <- rep('Cov',5)
n <- rep('Ctrl',5)

type <- as.data.frame(c(a,b,c,d,e,f,g,h,i,j,k,l,m,n))
colnames(type) <- 'Class_Type'
row.names(type) <- row.names(Combo1)

Combo4 <- cbind(type,Combo1)


write.csv(Combo4, 'ML_ready_CoV_4_classes.csv', row.names=TRUE)
```

We didn't do any fold change or stat measures on the capillary samples, but we can plot them by using ggplot2 and group the sets by timed intervals for each group A through E and picking a handful of genes to compare over the 0,12,24,36, and 48 hour time intervals for the control group and the Coronavirus inoculated groups.

When the values are a ratio like this, it is easier to see the larger changes as in 9 compared to a low change like 0.0005, but this just means that compared to the control samples the inoculated Coronavirus had 9 times the gene expression values or had downregulated or suppressed gene expression values to 1/5000th the amount of the normal range of gene expression values respectively. ***

It makes sense to use some genes we already know have a higher magnitude of change, and we have a column for that in the CombinedLiver table called MagnitudeFCs that was already sorted from largest to smallest when made. We'll just select the first five of those genes to compare in these capillary samples over time.

```
mostChanged <- CombinedLiver[1:5,c(1,71)]
mostSuppressed <- CombinedLiver[196:200,c(1,71)]
row.names(mostChanged)

## [1] "NEURL3" "DUSP1"  "ATF3"   "PCLO"   "LHB"

row.names(mostSuppressed)

## [1] "RASSF7"       "LOC100335030" "C2orf78"      "DEFB1"        "ZNF610"

capillary <- merge(mostChanged, CombinedLiver, by.x='GENE_SYMBOL', by.y='GENE
_SYMBOL')
capillary1 <- merge(mostSuppressed, CombinedLiver, by.x='GENE_SYMBOL', by.y='
GENE_SYMBOL')
capillaries <- rbind(capillary,capillary1)
Capillaries <- capillaries[,c(1,15:64)]
row.names(Capillaries) <- Capillaries$GENE_SYMBOL

Capillaries2 <- as.data.frame(t(Capillaries))
Capillaries2 <- Capillaries2[-1,]
row.names(Capillaries2) <- gsub('capillarySamples.','',row.names(Capillaries2
))
row.names(Capillaries2) <- gsub('GSM[0-9][0-9][0-9][0-9][0-9][0-9][0-9]_','',
row.names(Capillaries2))
row.names(Capillaries2) <- gsub('MERS_','', row.names(Capillaries2))

CoV <- grep('CoV', row.names(Capillaries2))
ctrl <- grep('ctrl', row.names(Capillaries2))

Capillaries2$Class <- 'CoV or ctrl'

Capillaries2[CoV,11] <- 'Coronavirus'
```

```
Capillaries2[ctrl,11] <- 'control'

A <- grep('_A', row.names(Capillaries2))
B <- grep('_B', row.names(Capillaries2))
C <- grep('_C', row.names(Capillaries2))
D <- grep('_D', row.names(Capillaries2))
E <- grep('_E', row.names(Capillaries2))

Capillaries2$Group <- 'group'

Capillaries2[A,12] <- 'A'
Capillaries2[B,12] <- 'B'
Capillaries2[C,12] <- 'C'
Capillaries2[D,12] <- 'D'
Capillaries2[E,12] <- 'E'

hr0 <- grep('0hr', row.names(Capillaries2))
hr12 <- grep('12hr', row.names(Capillaries2))
hr24 <- grep('24hr', row.names(Capillaries2))
hr36 <- grep('36hr', row.names(Capillaries2))
hr48 <- grep('48hr', row.names(Capillaries2))

Capillaries2$TimeInterval <- 'time'

Capillaries2[hr0,13] <- '0 hr'
Capillaries2[hr12,13] <- '12 hr'
Capillaries2[hr24,13] <- '24 hr'
Capillaries2[hr36,13] <- '36 hr'
Capillaries2[hr48,13] <- '48 hr'


write.csv(Capillaries2,'FC_10_capillaries_CoV.csv', row.names=TRUE)
```

The above table has 10 genes as the columns with the added Class (Coronavirus or control), Group (A,B,C,D,E), and TimeInterval (0,12,24,36,48 hours) fields to filter by and plot.

Lets make these group tables for the corona virus and see how they compare over time.

```
A_group <- filter(Capillaries2, Group=='A' & Class == 'Coronavirus')
B_group <- filter(Capillaries2, Group=='B' & Class == 'Coronavirus')
C_group <- filter(Capillaries2, Group=='C' & Class == 'Coronavirus')
D_group <- filter(Capillaries2, Group=='D' & Class == 'Coronavirus')
E_group <- filter(Capillaries2, Group=='E' & Class == 'Coronavirus')
```

We will do this for the A_group table and ignore the Group and Class fields, because we made it only the A group of the Coronavirus class.

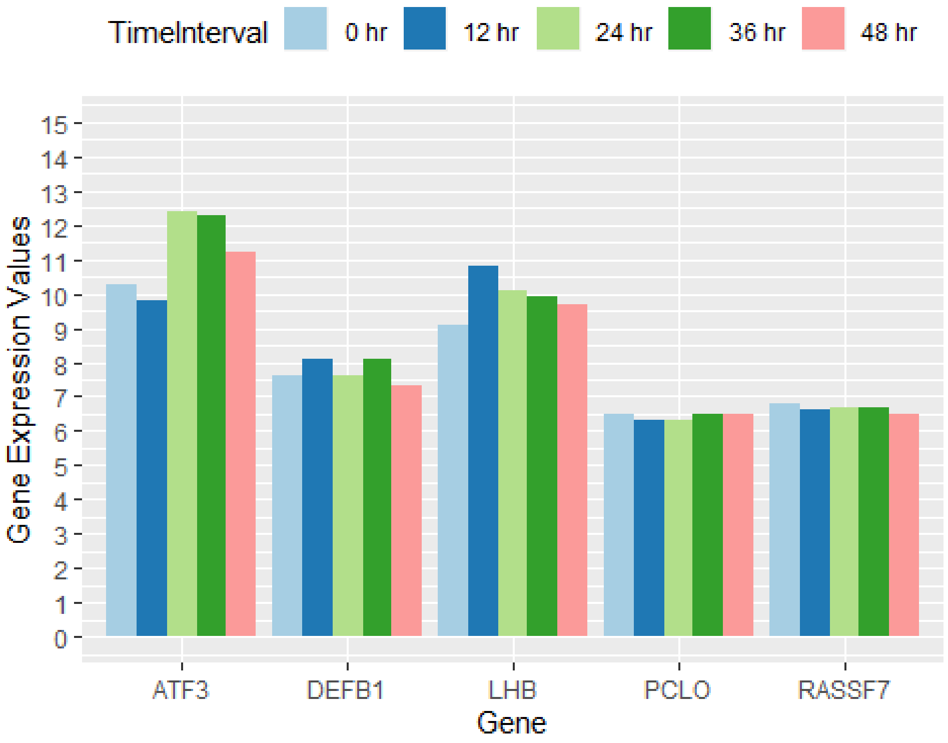
```
A_group2 <- A_group[,c(1,3,5,7,9,11:13)]
A_tidy <- gather(A_group2, 'Gene','GeneExpression',1:5)
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
A_tidy$GeneExpression <- round(as.numeric(A_tidy$GeneExpression),1)
A_tidy$TimeInterval <- as.factor(A_tidy$TimeInterval)
A_tidy$Gene <- as.factor(A_tidy$Gene)
A_tidy
```

```
##          Class Group TimeInterval   Gene GeneExpression
## 1  Coronavirus     A         0 hr   ATF3           10.3
## 2  Coronavirus     A        12 hr   ATF3            9.8
## 3  Coronavirus     A        24 hr   ATF3           12.4
## 4  Coronavirus     A        36 hr   ATF3           12.3
## 5  Coronavirus     A        48 hr   ATF3           11.2
## 6  Coronavirus     A         0 hr    LHB            9.1
## 7  Coronavirus     A        12 hr    LHB           10.8
## 8  Coronavirus     A        24 hr    LHB           10.1
## 9  Coronavirus     A        36 hr    LHB            9.9
## 10 Coronavirus     A        48 hr    LHB            9.7
## 11 Coronavirus     A         0 hr   PCLO            6.5
## 12 Coronavirus     A        12 hr   PCLO            6.3
## 13 Coronavirus     A        24 hr   PCLO            6.3
## 14 Coronavirus     A        36 hr   PCLO            6.5
## 15 Coronavirus     A        48 hr   PCLO            6.5
## 16 Coronavirus     A         0 hr  DEFB1            7.6
## 17 Coronavirus     A        12 hr  DEFB1            8.1
## 18 Coronavirus     A        24 hr  DEFB1            7.6
## 19 Coronavirus     A        36 hr  DEFB1            8.1
## 20 Coronavirus     A        48 hr  DEFB1            7.3
## 21 Coronavirus     A         0 hr RASSF7            6.8
## 22 Coronavirus     A        12 hr RASSF7            6.6
## 23 Coronavirus     A        24 hr RASSF7            6.7
## 24 Coronavirus     A        36 hr RASSF7            6.7
## 25 Coronavirus     A        48 hr RASSF7            6.5
```

```
ggplot(data = A_tidy, aes(x=Gene, y=GeneExpression, fill=TimeInterval)) +

  geom_bar(stat='identity', position=position_dodge())+
  scale_y_continuous(breaks = seq(0, 15, by=1), limits=c(0,15))+
  scale_fill_brewer(palette='Paired') +
  theme(legend.position="top")+
  #ggtitle('Group A with Coronavirus for Selected Genes Part 1')+
  xlab('Gene')+
  ylab('Gene Expression Values')
```

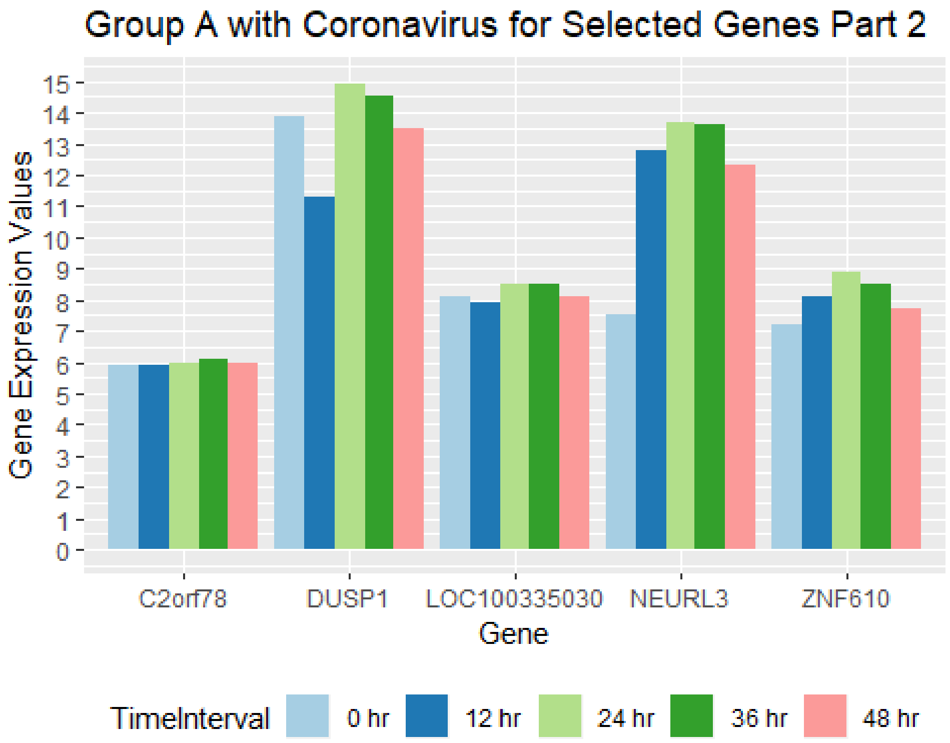


The genes above for Part 1 of the group A samples of coronavirus in blood capillaries show some variation in gene expression values for some of these genes that had the most change in the liver tumor samples. Starting at the initial hour up to 48 hours after being inoculated in vitro, there is an increase then decrease for ATF3 and LHB genes, while a decrease then increase close to initial value with PCLO and slightly with RASSF7. For DEFB1, it has a cyclical increase, decrease, increase, then decrease to stabilize closer to the initial gene expressio value.

Now lets find the other five genes in the group A set of ten genes found to have the most change in the liver tumor samples, and examined here in the blood capillary samples.

```
A_group3 <- A_group[,c(2,4,6,8,10,11:13)]
A_tidy1 <- gather(A_group3, 'Gene','GeneExpression',1:5)

## Warning: attributes are not identical across measure variables;
## they will be dropped

A_tidy1$GeneExpression <- round(as.numeric(A_tidy1$GeneExpression),1)
A_tidy1$TimeInterval <- as.factor(A_tidy1$TimeInterval)
A_tidy1$Gene <- as.factor(A_tidy1$Gene)
```

```
ggplot(data = A_tidy1, aes(x=Gene, y=GeneExpression, fill=TimeInterval)) +
  geom_bar(stat='identity', position=position_dodge())+
  scale_y_continuous(breaks = seq(0, 15, by=1), limits=c(0,15))+
  scale_fill_brewer(palette='Paired') +
  theme(legend.position="bottom")+
  ggtitle('Group A with Coronavirus for Selected Genes Part 2')+
  xlab('Gene')+
  ylab('Gene Expression Values')
```



The above genes in part 2 of the group A Coronavirus samples over 48 hours, shows that the gene expression values increase up to 24 hours then decrease to 48 hours for most of the genes above.

Lets look back at the platforms and the features removed. The sequence field is an interesting field because it can show copy number variants of genes by the genes that are duplicates at other probes from the samples.

```
Platform13497 <- read.csv('GPL13497-9755-forSequenceFeature-GSE100509.csv', s
ep=',',
                          na.strings=c('',' '), header=TRUE)
Platform16699 <- read.csv('GPL16699-forSequenceFeatureGSE89166_GSE89160.csv',
sep=',',
                                    na.strings=c('',' '), header=TRUE)
```

The features in Platform13497 and the first five listed ID values for this platform:

```
colnames(Platform13497)

##  [1] "ID"                    "SPOT_ID"               "CONTROL_TYPE"
##  [4] "REFSEQ"                "GB_ACC"                "GENE"
##  [7] "GENE_SYMBOL"           "GENE_NAME"             "UNIGENE_ID"
## [10] "ENSEMBL_ID"            "TIGR_ID"               "ACCESSION_STRING"
## [13] "CHROMOSOMAL_LOCATION"  "CYTOBAND"              "DESCRIPTION"
## [16] "GO_ID"                 "SEQUENCE"
```

The features in Platform16699 and the first five listed ID values of that platform:

```
colnames(Platform16699)

##  [1] "ID"                    "COL"                   "ROW"
##  [4] "NAME"                  "SPOT_ID"               "CONTROL_TYPE"
##  [7] "REFSEQ"                "GB_ACC"                "LOCUSLINK_ID"
## [10] "GENE_SYMBOL"           "GENE_NAME"             "UNIGENE_ID"
## [13] "ENSEMBL_ID"            "ACCESSION_STRING"      "CHROMOSOMAL_LOCATION"
## [16] "CYTOBAND"              "DESCRIPTION"           "GO_ID"
## [19] "SEQUENCE"
```

Lets keep only the ID (Platform16699) or SPOT_ID (Platform13497), GENE_SYMBOL, DESCRIPTION, and SEQUENCE features of both platforms.

```
P16699 <- Platform16699[,c(1,10,17,19)]
P13497 <- Platform13497[,c(2,7,15,17)]
```

Lets also remove the incomplete cases in both platforms.

```
work16699 <- P16699[complete.cases(P16699),]
work13497 <- P13497[complete.cases(P13497),]
```

Now merge these data sets to their corresponding samples by SPOT_ID.First read in the samples data for each platform and series.

```
GSE89166_89160 <- read.csv('GSE89166_GSE89160.csv',sep=',', na.strings=c('','
'),
                                     header=TRUE)
```

```
GSE100509 <- read.csv('GSE100509.csv', sep=',', header=TRUE,
                      na.strings=c('',' '))
```

Now merge the series data sets to their respective platforms of gene informational meta features.

```
Series100509 <- merge(work13497,GSE100509,by.x='SPOT_ID', by.y='ID_REF')
Series89166_89160 <- merge(work16699,GSE89166_89160,by.x='ID', by.y='ID_REF')
```

Rename the columns of the Series100509 to the 5 groups for each of CoV and Ctrl over 0,12,24,36, and 48 hours.

```
colnames(Series100509)

##  [1] "SPOT_ID"                    "GENE_SYMBOL"
##  [3] "DESCRIPTION"                "SEQUENCE"
##  [5] "GSM2685693_MERS_CoV_0hr"    "GSM2685694_MERS_CoV_0hr"
##  [7] "GSM2685695_MERS_CoV_0hr"    "GSM2685696_MERS_CoV_0hr"
##  [9] "GSM2685697_MERS_CoV_0hr"    "GSM2685698_ctrl_0hr"
## [11] "GSM2685699_ctrl_0hr"        "GSM2685700_ctrl_0hr"
## [13] "GSM2685701_ctrl_0hr"        "GSM2685702_ctrl_0hr"
## [15] "GSM2685703_MERS_CoV_12hr"   "GSM2685704_MERS_CoV_12hr"
## [17] "GSM2685705_MERS_CoV_12hr"   "GSM2685706_MERS_CoV_12hr"
## [19] "GSM2685707_MERS_CoV_12hr"   "GSM2685708_ctrl_12hr"
## [21] "GSM2685709_ctrl_12hr"       "GSM2685710_ctrl_12hr"
## [23] "GSM2685711_ctrl_12hr"       "GSM2685712_ctrl_12hr"
## [25] "GSM2685713_MERS_CoV_24hr"   "GSM2685714_MERS_CoV_24hr"
## [27] "GSM2685715_MERS_CoV_24hr"   "GSM2685716_MERS_CoV_24hr"
## [29] "GSM2685717_MERS_CoV_24hr"   "GSM2685718_ctrl_24hr"
## [31] "GSM2685719_ctrl_24hr"       "GSM2685720_ctrl_24hr"
## [33] "GSM2685721_ctrl_24hr"       "GSM2685722_ctrl_24hr"
## [35] "GSM2685723_MERS_CoV_36hr"   "GSM2685724_MERS_CoV_36hr"
## [37] "GSM2685725_MERS_CoV_36hr"   "GSM2685726_MERS_CoV_36hr"
## [39] "GSM2685727_MERS_CoV_36hr"   "GSM2685728_ctrl_36hr"
## [41] "GSM2685729_ctrl_36hr"       "GSM2685730_ctrl_36hr"
## [43] "GSM2685731_ctrl_36hr"       "GSM2685732_ctrl_36hr"
## [45] "GSM2685733_MERS_CoV_48hr"   "GSM2685734_MERS_CoV_48hr"
## [47] "GSM2685735_MERS_CoV_48hr"   "GSM2685736_MERS_CoV_48hr"
## [49] "GSM2685737_MERS_CoV_48hr"   "GSM2685738_ctrl_48hr"
## [51] "GSM2685739_ctrl_48hr"       "GSM2685740_ctrl_48hr"
## [53] "GSM2685741_ctrl_48hr"       "GSM2685742_ctrl_48hr"

group <- rep(1:5,10)
Group <- gsub('1','Group_A',group)
Group <- gsub('2','Group_B',Group)
Group <- gsub('3','Group_C',Group)
Group <- gsub('4','Group_D',Group)
Group <- gsub('5','Group_E',Group)

names <- colnames(Series100509)[5:54]
```

```
Names <- paste(names,Group,sep='_')

newNames <- gsub('_MERS','', Names)
newNames

##  [1] "GSM2685693_CoV_0hr_Group_A"   "GSM2685694_CoV_0hr_Group_B"
##  [3] "GSM2685695_CoV_0hr_Group_C"   "GSM2685696_CoV_0hr_Group_D"
##  [5] "GSM2685697_CoV_0hr_Group_E"   "GSM2685698_ctrl_0hr_Group_A"
##  [7] "GSM2685699_ctrl_0hr_Group_B"  "GSM2685700_ctrl_0hr_Group_C"
##  [9] "GSM2685701_ctrl_0hr_Group_D"  "GSM2685702_ctrl_0hr_Group_E"
## [11] "GSM2685703_CoV_12hr_Group_A"  "GSM2685704_CoV_12hr_Group_B"
## [13] "GSM2685705_CoV_12hr_Group_C"  "GSM2685706_CoV_12hr_Group_D"
## [15] "GSM2685707_CoV_12hr_Group_E"  "GSM2685708_ctrl_12hr_Group_A"
## [17] "GSM2685709_ctrl_12hr_Group_B" "GSM2685710_ctrl_12hr_Group_C"
## [19] "GSM2685711_ctrl_12hr_Group_D" "GSM2685712_ctrl_12hr_Group_E"
## [21] "GSM2685713_CoV_24hr_Group_A"  "GSM2685714_CoV_24hr_Group_B"
## [23] "GSM2685715_CoV_24hr_Group_C"  "GSM2685716_CoV_24hr_Group_D"
## [25] "GSM2685717_CoV_24hr_Group_E"  "GSM2685718_ctrl_24hr_Group_A"
## [27] "GSM2685719_ctrl_24hr_Group_B" "GSM2685720_ctrl_24hr_Group_C"
## [29] "GSM2685721_ctrl_24hr_Group_D" "GSM2685722_ctrl_24hr_Group_E"
## [31] "GSM2685723_CoV_36hr_Group_A"  "GSM2685724_CoV_36hr_Group_B"
## [33] "GSM2685725_CoV_36hr_Group_C"  "GSM2685726_CoV_36hr_Group_D"
## [35] "GSM2685727_CoV_36hr_Group_E"  "GSM2685728_ctrl_36hr_Group_A"
## [37] "GSM2685729_ctrl_36hr_Group_B" "GSM2685730_ctrl_36hr_Group_C"
## [39] "GSM2685731_ctrl_36hr_Group_D" "GSM2685732_ctrl_36hr_Group_E"
## [41] "GSM2685733_CoV_48hr_Group_A"  "GSM2685734_CoV_48hr_Group_B"
## [43] "GSM2685735_CoV_48hr_Group_C"  "GSM2685736_CoV_48hr_Group_D"
## [45] "GSM2685737_CoV_48hr_Group_E"  "GSM2685738_ctrl_48hr_Group_A"
## [47] "GSM2685739_ctrl_48hr_Group_B" "GSM2685740_ctrl_48hr_Group_C"
## [49] "GSM2685741_ctrl_48hr_Group_D" "GSM2685742_ctrl_48hr_Group_E"
```

Change the column names in Series100509 to the new column names identifying which group the samples is from in A:E.

```
colnames(Series100509)[5:54] <- newNames
colnames(Series100509)

##  [1] "SPOT_ID"                      "GENE_SYMBOL"
##  [3] "DESCRIPTION"                  "SEQUENCE"
##  [5] "GSM2685693_CoV_0hr_Group_A"   "GSM2685694_CoV_0hr_Group_B"
##  [7] "GSM2685695_CoV_0hr_Group_C"   "GSM2685696_CoV_0hr_Group_D"
##  [9] "GSM2685697_CoV_0hr_Group_E"   "GSM2685698_ctrl_0hr_Group_A"
## [11] "GSM2685699_ctrl_0hr_Group_B"  "GSM2685700_ctrl_0hr_Group_C"
## [13] "GSM2685701_ctrl_0hr_Group_D"  "GSM2685702_ctrl_0hr_Group_E"
## [15] "GSM2685703_CoV_12hr_Group_A"  "GSM2685704_CoV_12hr_Group_B"
## [17] "GSM2685705_CoV_12hr_Group_C"  "GSM2685706_CoV_12hr_Group_D"
## [19] "GSM2685707_CoV_12hr_Group_E"  "GSM2685708_ctrl_12hr_Group_A"
## [21] "GSM2685709_ctrl_12hr_Group_B" "GSM2685710_ctrl_12hr_Group_C"
## [23] "GSM2685711_ctrl_12hr_Group_D" "GSM2685712_ctrl_12hr_Group_E"
## [25] "GSM2685713_CoV_24hr_Group_A"  "GSM2685714_CoV_24hr_Group_B"
## [27] "GSM2685715_CoV_24hr_Group_C"  "GSM2685716_CoV_24hr_Group_D"
## [29] "GSM2685717_CoV_24hr_Group_E"  "GSM2685718_ctrl_24hr_Group_A"
## [31] "GSM2685719_ctrl_24hr_Group_B" "GSM2685720_ctrl_24hr_Group_C"
## [33] "GSM2685721_ctrl_24hr_Group_D" "GSM2685722_ctrl_24hr_Group_E"
## [35] "GSM2685723_CoV_36hr_Group_A"  "GSM2685724_CoV_36hr_Group_B"
## [37] "GSM2685725_CoV_36hr_Group_C"  "GSM2685726_CoV_36hr_Group_D"
## [39] "GSM2685727_CoV_36hr_Group_E"  "GSM2685728_ctrl_36hr_Group_A"
## [41] "GSM2685729_ctrl_36hr_Group_B" "GSM2685730_ctrl_36hr_Group_C"
## [43] "GSM2685731_ctrl_36hr_Group_D" "GSM2685732_ctrl_36hr_Group_E"
## [45] "GSM2685733_CoV_48hr_Group_A"  "GSM2685734_CoV_48hr_Group_B"
## [47] "GSM2685735_CoV_48hr_Group_C"  "GSM2685736_CoV_48hr_Group_D"
## [49] "GSM2685737_CoV_48hr_Group_E"  "GSM2685738_ctrl_48hr_Group_A"
## [51] "GSM2685739_ctrl_48hr_Group_B" "GSM2685740_ctrl_48hr_Group_C"
## [53] "GSM2685741_ctrl_48hr_Group_D" "GSM2685742_ctrl_48hr_Group_E"
```

Remove the ID and SPOT_ID fields of the probe labels that won't be needed for the analysis.

```
Series89 <- Series89166_89160[,-1]
Series100 <- Series100509[,-c(1,3,4)]
```

Now combine the series together by genes in common.

```
ComboLiverCapillarySequences <- merge(Series89,Series100, by.x='GENE_SYMBOL',
                                      by.y='GENE_SYMBOL')
```

There should be different genotypes or copy number variations in the SEQUENCE feature column, which will identify which nucleotide jumps, is deleted, rearranged in the gene expressions. This time around lets group by SEQUENCE to see how many genotypes there are in all the genes. This could give more information in the analysis of any genotypes of genes could be more susceptible to pathogenesis of CoV or immunity to it. ***

This next portion of this analysis shows the top five genes in the data, the number of genotypes or copy number variations of the nucleotide sequences are in each gene, and the gene name. This could be useful to determine how well the genotypes within these five genes that were expressed more than the other 65k genes were when analyzing the effects of CoV, inactive CoV, CoV treated with an interleukin, and the control samples.

```
SeqGroup <- ComboLiverCapillarySequences %>% group_by(GENE_SYMBOL) %>% count(
n=n())
SeqGroup <- SeqGroup[order(SeqGroup$n, decreasing=TRUE),-3]

SeqGroup5 <- SeqGroup[1:5,]

genes5sequences <- merge(SeqGroup5,ComboLiverCapillarySequences,
                         by.x='GENE_SYMBOL', by.y='GENE_SYMBOL')

genotypes5 <- genes5sequences %>% group_by(SEQUENCE) %>% count(n=n)
```

```
genotypes5$n <- as.factor(genotypes5$n)
SeqGroup5$n <- as.factor(SeqGroup5$n)

genotypes5_1 <- merge(genotypes5, SeqGroup5, by.x='n', by.y='n')
colnames(genotypes5_1)[c(1,3)] <- c('geneCount','genotypeCount')
head(genotypes5_1)
```

```
##   geneCount                                                     SEQUENCE
## 1       255 AAACTTACTCCAGAGCTCCTTGTGCATCTGACCAGCACCATCGACAGAATAAACACAGAA
## 2       255 AACAGAGTCCTCAGGGAAGAAAATCGAAGACTTCAGGCTCAACTGAGTCATGTTTCCAGA
## 3       255 GCACCTGTGTTCTTTGAGTTCACATCATGAATGTGGTGATTTCCCAGATACCATCTCAGG
## 4       255 ATGGGGTGCTCTGGGGAAATATTGGAGGGTCATCCATTCCACATTAAAAGAGCAAGTTGT
## 5       255 AAGTGCTTGGAAATACTTGGGTGAATGTTACCAGACTCCTTCTCTCTCAGCTTACAGCCT
## 6       255 AATCTACGAGGCACTTTATGGCAATTCCAAGAAGGGGCTGAAAGGTATGTGTTCTTCTCC
##   genotypeCount GENE_SYMBOL
## 1            15     PDE4DIP
## 2            15     PDE4DIP
## 3            15     PDE4DIP
## 4            15     PDE4DIP
## 5            15     PDE4DIP
## 6            15     PDE4DIP
```

Write the files above out to csv.

```
write.csv(ComboLiverCapillarySequences, 'SequencesBothCleaned.csv',row.names=
FALSE)
write.csv(genotypes5_1, 'Genotypes_5_1.csv', row.names=FALSE)

ComboCNV <- ComboLiverCapillarySequences[,-c(1,2)]
copynumbers <- merge(genotypes5_1,ComboCNV,
                     by.x='SEQUENCE', by.y='SEQUENCE')
CNV <- copynumbers[!duplicated(copynumbers$SEQUENCE),]
write.csv(CNV, 'copyNumbers.csv', row.names=FALSE)

colnames(CNV)
```

```
##  [1] "SEQUENCE"                     "geneCount"                   
##  [3] "genotypeCount"                "GENE_SYMBOL"                 
##  [5] "GSM2359851_CoV1"              "GSM2359853_CoV2"             
##  [7] "GSM2359910_CoV3"              "GSM2359913_CoV4"             
##  [9] "GSM2359850_ctrl1"             "GSM2359852_ctrl2"            
## [11] "GSM2359911_ctrl3"             "GSM2359914_ctrl4"            
## [13] "GSM2359912_Il1"               "GSM2359917_IL2"              
## [15] "GSM2359915_inactiveHeatCoV1"  "GSM2359916_inactiveHeatCoV2" 
## [17] "GSM2685693_CoV_0hr_Group_A"   "GSM2685694_CoV_0hr_Group_B"  
## [19] "GSM2685695_CoV_0hr_Group_C"   "GSM2685696_CoV_0hr_Group_D"  
## [21] "GSM2685697_CoV_0hr_Group_E"   "GSM2685698_ctrl_0hr_Group_A" 
## [23] "GSM2685699_ctrl_0hr_Group_B"  "GSM2685700_ctrl_0hr_Group_C" 
## [25] "GSM2685701_ctrl_0hr_Group_D"  "GSM2685702_ctrl_0hr_Group_E" 
## [27] "GSM2685703_CoV_12hr_Group_A"  "GSM2685704_CoV_12hr_Group_B"
## [29] "GSM2685705_CoV_12hr_Group_C"  "GSM2685706_CoV_12hr_Group_D"
## [31] "GSM2685707_CoV_12hr_Group_E"  "GSM2685708_ctrl_12hr_Group_A"
## [33] "GSM2685709_ctrl_12hr_Group_B" "GSM2685710_ctrl_12hr_Group_C"
## [35] "GSM2685711_ctrl_12hr_Group_D" "GSM2685712_ctrl_12hr_Group_E"
## [37] "GSM2685713_CoV_24hr_Group_A"  "GSM2685714_CoV_24hr_Group_B"
## [39] "GSM2685715_CoV_24hr_Group_C"  "GSM2685716_CoV_24hr_Group_D"
## [41] "GSM2685717_CoV_24hr_Group_E"  "GSM2685718_ctrl_24hr_Group_A"
## [43] "GSM2685719_ctrl_24hr_Group_B" "GSM2685720_ctrl_24hr_Group_C"
## [45] "GSM2685721_ctrl_24hr_Group_D" "GSM2685722_ctrl_24hr_Group_E"
## [47] "GSM2685723_CoV_36hr_Group_A"  "GSM2685724_CoV_36hr_Group_B"
## [49] "GSM2685725_CoV_36hr_Group_C"  "GSM2685726_CoV_36hr_Group_D"
## [51] "GSM2685727_CoV_36hr_Group_E"  "GSM2685728_ctrl_36hr_Group_A"
## [53] "GSM2685729_ctrl_36hr_Group_B" "GSM2685730_ctrl_36hr_Group_C"
## [55] "GSM2685731_ctrl_36hr_Group_D" "GSM2685732_ctrl_36hr_Group_E"
## [57] "GSM2685733_CoV_48hr_Group_A"  "GSM2685734_CoV_48hr_Group_B"
## [59] "GSM2685735_CoV_48hr_Group_C"  "GSM2685736_CoV_48hr_Group_D"
## [61] "GSM2685737_CoV_48hr_Group_E"  "GSM2685738_ctrl_48hr_Group_A"
## [63] "GSM2685739_ctrl_48hr_Group_B" "GSM2685740_ctrl_48hr_Group_C"
## [65] "GSM2685741_ctrl_48hr_Group_D" "GSM2685742_ctrl_48hr_Group_E"
```

This feature, SEQUENCE, provides the single nucleotide polymorphisms (SNP)s or copy number variants of each gene's DNA, many are duplicated. Here are the first few SNPs.

```
head(CNV$SEQUENCE)
```

```
## [1] AAACTTACTCCAGAGCTCCTTGTGCATCTGACCAGCACCATCGACAGAATAAACACAGAA
## [2] AACAGAGTCCTCAGGGAAGAAAATCGAAGACTTCAGGCTCAACTGAGTCATGTTTCCAGA
## [3] AAGGATTTGCTTATAAGGGTTCCTGCTTTCACAAAATTATTCCAGGTTTTATGTGTCAGG
## [4] AAGGATTTGGTTGTAAGGGCTCCCGCTTTCACAGAATTATTCCAGGGTTTATGTGTCAGG
## [5] AAGTGCTTGGAAATACTTGGGTGAATGTTACCAGACTCCTTCTCTCTCAGCTTACAGCCT
## [6] AATCTACGAGGCACTTTATGGCAATTCCAAGAAGGGGCTGAAAGGTATGTGTTCTTCTCC
## 50571 Levels: AAACAAAAAACAGGTTAAGAAAATTACTTGGGTGGGCAGACTTAGGAACGCTCTACTCGG
...
```

It would be interesting in the fold change values between genotypes of the genes expressed in comparing the capillary samples all within 1 hour after being inoculated with CoV, inactive CoV, CoV and an interleukin alpha, or control group. Then compare the liver tumor samples of each group A through E that was monitored after being inoculated with CoV at 0,12,24,36, and 48 hours side by side with the control groups of groups A through E.

We could also detect patterns to analyze those sequences of copy number variations within the top 5 genes expressed the most number of times in this data. Comparing networks of genes associated with processes in the body like immune response, pathogenesis of disease onset, networks of human processes in the body associated with cancer or subsequent diseases like autoimmune, celiac disease, hemochromatosis, anemia, etc. To see how well any of those genotypes fair.

We have the five genes that had the highest copy number variations within them and merged it with our liver tumor and capillary CoV and control samples in the CNV data table.

I joinged genecards.org and found that this site has some useful information on gene networks for analyzing genes that play a role in a network of genes that function at some level on the human anatomy. Such as anemia, cannabidiol,celiac disease, diabetes type 1 and 2, hemochromatosis, immunity, pain, and tumorigenesis. I downloaded the csv files for each of these networks just mentioned. There is a ranking of each gene as importance the higher the ranked order in the network functional role on the human body.

We can import each of those now.

```
anemiaNetwork <- read.csv('GeneCards-SearchResults-anemiaNetwork.csv', sep=',',
                          header=TRUE, na.strings=c('',' '))

cannabidiolNetwork <- read.csv('GeneCards-SearchResults-cannabidiolNetwork.csv', sep=',',
                          header=TRUE, na.strings=c('',' '))
celiacDiseaseNetwork <- read.csv('GeneCards-SearchResults-celiacDiseaseNetwork.csv',
                                 sep=',', header=TRUE, na.strings=c('',' '))
diabetesType1Network <- read.csv('GeneCards-SearchResults-diabetesType1Network.csv',
                                 sep=',', header=TRUE, na.strings=c('',' '))
diabetesType2Network <- read.csv('GeneCards-SearchResults-diabetesType2Network.csv',
                                 sep=',', header=TRUE, na.strings=c('',' '))
hemochromatosisNetwork <- read.csv('GeneCards-SearchResults-hemochromatosisNetwork.csv',
                          sep=',',header=TRUE, na.strings=c('',' '))
immunityNetwork <- read.csv('GeneCards-SearchResults-immunityNetwork.csv',
                          sep=',',header=TRUE, na.strings=c('',' '))
painNetwork <- read.csv('GeneCards-SearchResults-painNetwork.csv',
                          sep=',', header=TRUE, na.strings=c('',' '))
tumorigenesisNetwork <- read.csv('GeneCards-SearchResults-tumorigenesisNetwork.csv', sep=',',
                          header=TRUE, na.strings=c('',' '))
```

We could then take our combined data from the liver and capillary datasets to look at the copy number variants of genotypes within these networks separately to understand how or if these genes have a high number of copy number in control versus diseased and diseased/treated states.

Lets take the top five genes from each network, add a column feature to label the gene network, then combine those tables. They are already ordered each by Relevance.score.

```
amenia5 <- anemiaNetwork[1:5,]
cannabidiol5 <- cannabidiolNetwork[1:5,]
diabetesOne5 <- diabetesType1Network[1:5,]
diabetesTwo5 <- diabetesType2Network[1:5,]
hemochromatosis5 <- hemochromatosisNetwork[1:5,]
immunity5 <- immunityNetwork[1:5,]
pain5 <- painNetwork[1:5,]
tumorigenesis5 <- tumorigenesisNetwork[1:5,]
```

Lets merge each network disease pathway with the ComboLiverCapillarySequences data table.

```
anemia5_path <- merge(amenia5, ComboLiverCapillarySequences, by.x='Gene.Symbo
l',
                      by.y='GENE_SYMBOL')
cannabidiol5_path <- merge(cannabidiol5, ComboLiverCapillarySequences, by.x='
Gene.Symbol',
                      by.y='GENE_SYMBOL')
diabetesOne5_path <- merge(diabetesOne5, ComboLiverCapillarySequences, by.x='
Gene.Symbol',
                      by.y='GENE_SYMBOL')
diabetesTwo5_path <- merge(diabetesTwo5, ComboLiverCapillarySequences, by.x='
Gene.Symbol',
                      by.y='GENE_SYMBOL')
hemochromatosis5_path <- merge(hemochromatosis5, ComboLiverCapillarySequences
, by.x='Gene.Symbol',
                      by.y='GENE_SYMBOL')
immunity5_path <- merge(immunity5, ComboLiverCapillarySequences, by.x='Gene.S
ymbol',
                      by.y='GENE_SYMBOL')
pain5_path <- merge(pain5, ComboLiverCapillarySequences, by.x='Gene.Symbol',
                      by.y='GENE_SYMBOL')
tumorigenesis5_path <- merge(tumorigenesis5, ComboLiverCapillarySequences, by
.x='Gene.Symbol',
                      by.y='GENE_SYMBOL')
```

Combine the above data tables together.

```
Network <- as.data.frame(c(rep('anemia',length(anemia5_path$Gene.Symbol)),
                         rep('cannabidiol',length(cannabidiol5_path$Gene.Symb
ol)),
                         rep('diabetes 1',length(diabetesOne5_path$Gene.Symbo
l)),
                         rep('diabetes 2', length(diabetesTwo5_path$Gene.Symb
ol)),
                         rep('hemochromatosis', length(hemochromatosis5_path$
Gene.Symbol)),
                         rep('immunity', length(immunity5_path$Gene.Symbol)),
                         rep('pain', length(pain5_path$Gene.Symbol)),
                         rep('tumorigenesis',length(tumorigenesis5_path$Gene.
Symbol))))
colnames(Network) <- 'geneNetwork'

networkPathwayGenes <- rbind(anemia5_path,cannabidiol5_path,diabetesOne5_path
,                             diabetesTwo5_path,hemochromatosis5_path,immunit
y5_path,
                             pain5_path,tumorigenesis5_path)
PathwayGenes <- cbind(Network,networkPathwayGenes)
```

Now group by gene for the count and then by sequence for the copy number variant count within the gene.

```
pathGenesCount <- PathwayGenes %>% group_by(Gene.Symbol) %>% count(n())
pathGenotypeCount <- PathwayGenes %>% group_by(SEQUENCE) %>% count(n())

pathGenes <- pathGenesCount[,-2]
colnames(pathGenes)[2] <- 'GeneCount'
pathGenos <- pathGenotypeCount[,-2]
colnames(pathGenos)[2] <- 'GenotypeCount'

pG <- merge(pathGenos, PathwayGenes, by.x='SEQUENCE', by.y='SEQUENCE')
pG1 <- merge(pathGenes, pG, by.x='Gene.Symbol', by.y='Gene.Symbol')
colnames(pG1)[c(13,14,17,18)] <- paste('Study1',colnames(pG1)[c(13,14,17,18)]
,
                                       sep='_')
write.csv(pG1, 'genes-genotypes-networks.csv', row.names=FALSE)
```

I want to remove and hold it separate for one of the liver tumor samples studies that uses CoV1, CoV2, ctrl1, and ctrl2, because the values are not scaled the same and these numbers will ruin the analysis included as is. They were already log scaled but seem to be much larger than the other values after scaling.The previous block of code added ‘Study1’ to those larger scaled samples to extract for separate analysis.

```
scaleSet <- pG1[,c(1:14,17,18)]
otherSet <- pG1[,c(1:12,15,16,19:74)]
```

Lets add in that the scaleSet columns are from study1 and leave the study2 column names in the otherSet as is. Combine and make a separate data set. I have previously examined cannabidiol in sebum/acne and brain tumor samples to briefly determine the CBD effects on hormone and immune response genes. I added this cannabidiol network to get the top five CBD genes by relative score in the CBD network. We can look at this scale set of CoV and ctrl samples from liver tumors to see if there are any obvious effects to this network that has shown to treat chronic pain in some folks.

Lets get the data set of cannabidiol, CBD, to compare CoV to mock or control liver tumor samples from the larger scaled data set.

```
CBD <- subset(scaleSet, scaleSet$geneNetwork == 'cannabidiol')
orderCBD <- CBD[order(CBD$Gene.Symbol, decreasing=TRUE),]
```

We can see how many geno types are unique to a gene by adding up the genotypes to equal the genes in count. For instance, the PPARG gene has 22 counts of that gene in the data with one genotype having 22 counts, and the other genotype for that gene having only 2 counts. So that there are two unique genotypes for the gene PPARG. Lets go ahead and use dplyr to group by SEQUENCE then take the mean of that sequence but for the CoV and ctrl groups separately.

The two separate CoV and Ctrl datasets for the five CBD genes with the highest count of occurence in the data.

```
CBD_CoV <- CBD[,-c(15,16)]

CBD_ctrl <- CBD[,-c(13,14)]

samples <- as.vector(colnames(CBD_CoV)[13:14])
samples1 <- as.vector(colnames(CBD_ctrl)[13:14])

CBD_CovMeans <- CBD_CoV %>% group_by(SEQUENCE) %>% summarise_at(vars(samples)
,
                                                               mean)
CBD_CtrlMeans <- CBD_ctrl %>% group_by(SEQUENCE) %>% summarise_at(vars(sample
s1),
                                                               mean)
```

It seems a small data set, but with this network, CBD network, there was only one gene with more than one genotype. This is why there are only six genotypes when we selected five genes.

```
unique(CBD$SEQUENCE)

## [1] TGTACTAGGCCTACTGGGGATCAGAGTTCCCAAGAAAGGAAACCTTTTCTTGTATCTGGA
## [2] AGCATCTGCTCCCTCTACCAGCTGGAGAACTACTGCAACTAGACGCAGCCCGCAGGCAGC
## [3] CCAAGGCTTCATGACAAGGGAGTTTCTAAAGAGCCTGCGAAAGCCTTTTGGTGACTTTAT
## [4] ACAATCAGATTGAAGCTTATCTATGACAGATGTGATCTTAACTGTCGGATCCACAAAAAA
## [5] TGAGATATTTAAGGTTGAATGTTTGTCCTTAGGATAGGCCTATGTGCTAGCCCACAAAGA
## [6] GGGGTATCCTGGGGGACCCAATGTAGGAGCTGCCTTGGCTCAGACATGTTTTCCGTGAAA
## 50571 Levels: AAACAAAAAACAGGTTAAGAAAATTACTTGGGTGGGCAGACTTAGGAACGCTCTACTCGG
...
```

Lets combine the data means for each group by genotype or unique sequence with the genes they belong to from two features of the CBD data set, Gene.Symbol and SEQUENCE.

```
geneSeq <- CBD[,c(1:5)]
geneSeq1 <- merge(geneSeq, CBD_CovMeans, by.x='SEQUENCE', by.y='SEQUENCE')
geneSeq2 <- merge(geneSeq1, CBD_CtrlMeans, by.x='SEQUENCE', by.y='SEQUENCE')
```

```
geneSeq3 <- geneSeq2[!duplicated(geneSeq2),]
geneSeq3
```

```
##                                                         SEQUENCE Gene.Symbo
l
## 1  ACAATCAGATTGAAGCTTATCTATGACAGATGTGATCTTAACTGTCGGATCCACAAAAAA       PPAR
G
## 3  AGCATCTGCTCCCTCTACCAGCTGGAGAACTACTGCAACTAGACGCAGCCCGCAGGCAGC         IN
S
## 4  CCAAGGCTTCATGACAAGGGAGTTTCTAAAGAGCCTGCGAAAGCCTTTTGGTGACTTTAT       PPAR
G
## 24 GGGGTATCCTGGGGGACCCAATGTAGGAGCTGCCTTGGCTCAGACATGTTTTCCGTGAAA         TN
F
## 25 TGAGATATTTAAGGTTGAATGTTTGTCCTTAGGATAGGCCTATGTGCTAGCCCACAAAGA       PTGS
2
## 35 TGTACTAGGCCTACTGGGGATCAGAGTTCCCAAGAAAGGAAACCTTTTCTTGTATCTGGA        CNR
1
##    GeneCount GenotypeCount geneNetwork Study1_GSM2359851_CoV1
## 1         22             2 cannabidiol              302.82500
## 3          3             3 cannabidiol                8.99750
## 4         22            20 cannabidiol             1076.81550
## 24         4             4 cannabidiol               11.87500
## 25        10            10 cannabidiol               12.35425
## 35         1             1 cannabidiol               65.61000
##    Study1_GSM2359853_CoV2 Study1_GSM2359850_ctrl1 Study1_GSM2359852_ctrl2
## 1               246.56250               393.17250               380.69500
## 3                14.20750                15.63000                20.21500
## 4               856.73738              1321.94000              1184.44325
## 24               15.31250                15.35750                 8.09750
## 25               10.88063                11.85675                10.14488
## 35               84.29750                44.82750                68.61000
```

The above table shows the mean value of each genotype in the four samples, where two samples are liver tumor inoculated with CoV, and the other two are the liver tumor samples without CoV with both having been screened after 1 hour.

Since we want the fold change values within each geno type of the CBD network, lets get the means of each type of either CoV or ctrl. This is now the collective sample means of the genotype means per unique sample of either CoV or ctrl.

```
geneSeq3$CoV_Genotype_Mean <- rowMeans(geneSeq3[,6:7])
geneSeq3$Ctrl_Geneotype_Mean <- rowMeans(geneSeq3[,8:9])
geneSeq3[,c(1,2,10,11)]

##                                                         SEQUENCE Gene.Symbo
l
## 1  ACAATCAGATTGAAGCTTATCTATGACAGATGTGATCTTAACTGTCGGATCCACAAAAAA       PPAR
G
## 3  AGCATCTGCTCCCTCTACCAGCTGGAGAACTACTGCAACTAGACGCAGCCCGCAGGCAGC         IN
S
## 4  CCAAGGCTTCATGACAAGGGAGTTTCTAAAGAGCCTGCGAAAGCCTTTTGGTGACTTTAT       PPAR
G
## 24 GGGGTATCCTGGGGGACCCAATGTAGGAGCTGCCTTGGCTCAGACATGTTTTCCGTGAAA         TN
F
## 25 TGAGATATTTAAGGTTGAATGTTTGTCCTTAGGATAGGCCTATGTGCTAGCCCACAAAGA       PTGS
2
## 35 TGTACTAGGCCTACTGGGGATCAGAGTTCCCAAGAAAGGAAACCTTTTCTTGTATCTGGA        CNR
1
##    CoV_Genotype_Mean Ctrl_Geneotype_Mean
## 1          274.69375           386.93375
## 3           11.60250            17.92250
## 4          966.77644          1253.19162
## 24          13.59375            11.72750
## 25          11.61744            11.00081
## 35          74.95375            56.71875
```

Looking at the above, both genotypes of PPARG decrease in CoV treated samples as well as insulin, INS. But tumor necrosis factor or TNF, prostaglandin-Endoperoxide Synthase 2 or PTGS2 and Cannabinoid Receptor 1 or CNR1 all increase in CoV treated samples. This could mean that in liver tumor samples that are inoculated with Coronavirus, more tumor suppressant or destroyer called TNF is activated, more pain reliever activation in the increased CNR1, and less insulin is produced. Insulin is a diabetic factor that plays a role in glucose processing for energy. If less insulin is produced after one hour, this could mean the body doesn't want to feed a contaminant it is recognizing in the body by lessening the need for the body to consume more glucose. An aside on the insulin and glucose relationship in regulating the body: Insulin and glucose are supposed to be balanced, where the more glucose is burned by the body for energy will lead to more insulin produced to compensate energy needs. The two types of diabetes controlled by insulin is either an overproduction of insulin that leads to an irregular need to ead more than normal amounts of food to get more glucose in the blood, or an underproduction of insulin that leads to an inadequate amount of glucose the body needs, loss of appetite and irregularly small amounts of food consumed.

Immediately, some more analysis to answer questions of these assertions on the results above can be further confirmed by:

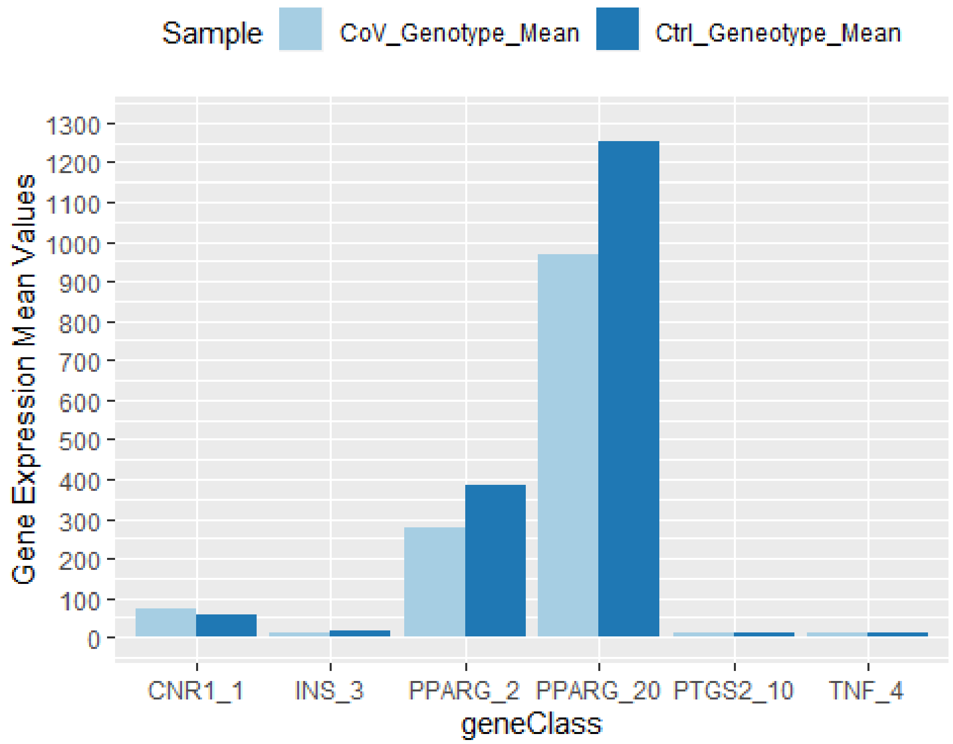
1.) Analyzing the **time sequence of the blood capillary samples** inoculated with CoV and their respective control groups. Examining how these CBD genes are effected over time.

2.) Confirming that insulin is effected by CoV by looking at the **diabetic gene networks** for type 1 and type 2 diabetes.

3.) Also, looking at the **tumorigenesis gene network** to see how the genes that are involved in tumor growth in the body react to CoV in the liver tumor samples and the blood capillary samples.

Lets put an interesting bar chart into this script to show what we have so far in the CBD genes network for liver tumor samples inoculated over one hour's time. Lets use the mutate function of dplyr to create a new feature that will name each gene with more than one genotype or copy number variant in this data. Then use tidyr to gather the sample means per genotype. And finally, plot the CoV and ctrl means using ggplot2.

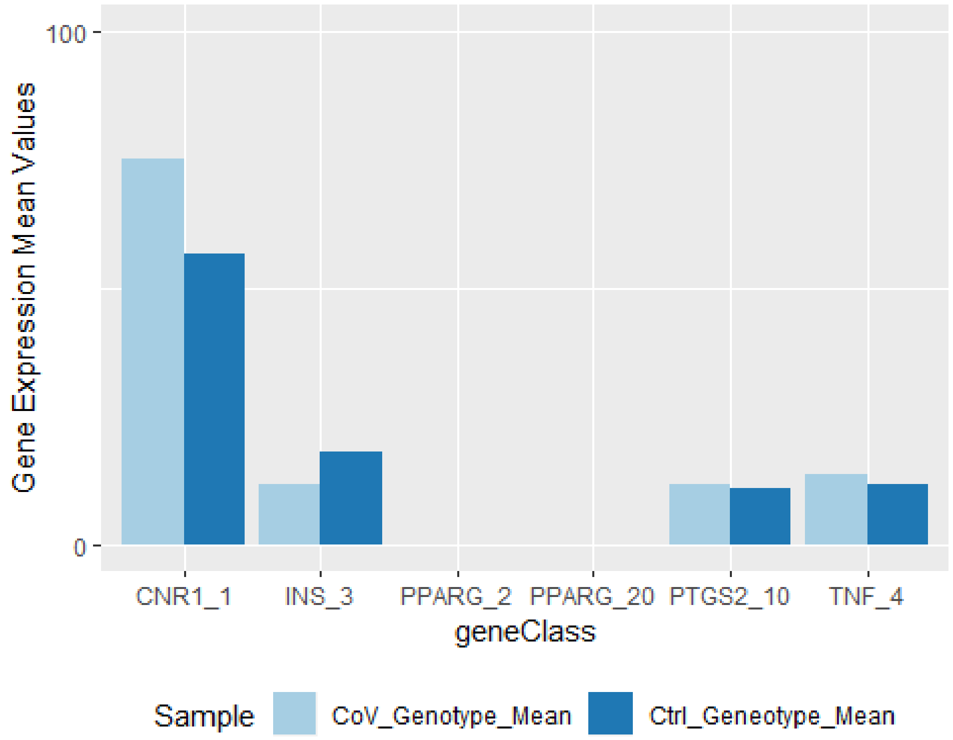
```r
CBD_plot <- geneSeq3 %>% mutate(geneClass = paste(Gene.Symbol,GenotypeCount,
                                                   sep='_'))

CBD_plot2 <- gather(CBD_plot, 'Sample','MeanValue',10:11)

ggplot(data = CBD_plot2, aes(x=geneClass, y=MeanValue, fill=Sample)) +
  geom_bar(stat='identity', position=position_dodge())+
  scale_y_continuous(breaks = seq(0, 1300, by=100), limits=c(0,1300))+
  scale_fill_brewer(palette='Paired') +
  #ggtitle('CBD Network Gene Expression Values Treated with Coronavirus')+
  theme(legend.position="top")+
  #xlab('Gene and Genotype or Single Nucleotide Polymorphism or Copy Number V
ariant')+
  ylab('Gene Expression Mean Values')
```



The lower values aren't as easy to distinguish as the scale of change in Mean values is significantly greater for some genes. We could fix this by making the plot fold change and add that field, or adding a log2 field of the means to put on the same scale for visualization purposes. Or cut off the scale for those less than 100 in Mean Values and those greater than 100.

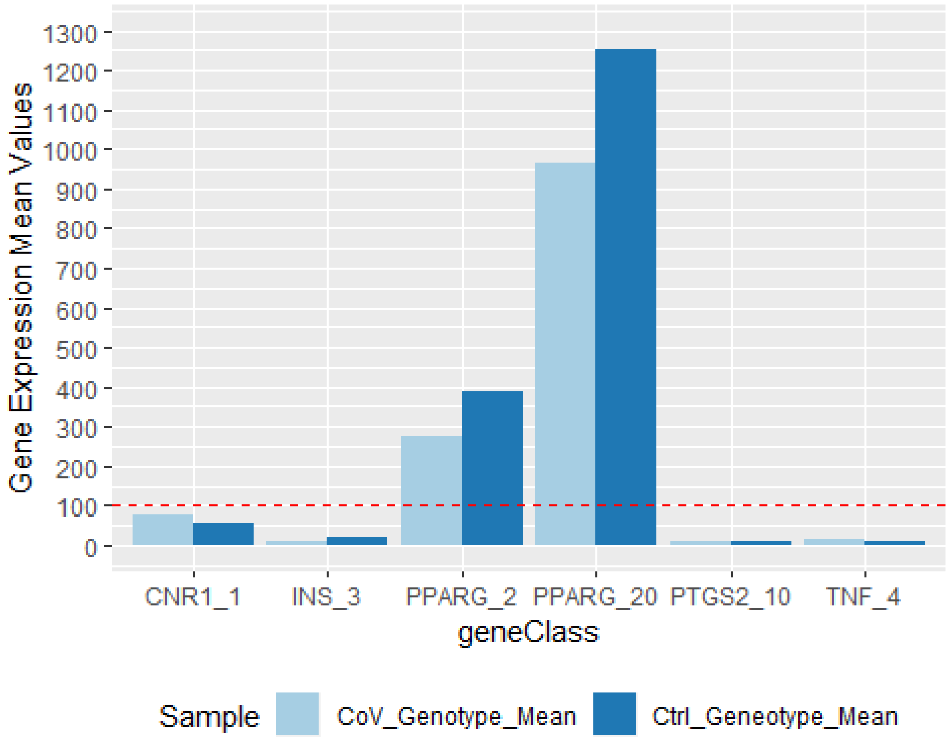
The following is the genes less than 100 for Mean genotype expression values.

```r
ggplot(data = CBD_plot2, aes(x=geneClass, y=MeanValue, fill=Sample)) +
  geom_bar(stat='identity', position=position_dodge())+
  scale_y_continuous(breaks = seq(0, 100, by=100), limits=c(0,100))+
  scale_fill_brewer(palette='Paired') +
  theme(legend.position="bottom")+
  #ggtitle('CBD Network Gene Expression Values Less than 100 Treated with CoV
')+
  ylab('Gene Expression Mean Values')
```



The following is the genes greater than 100 for Mean genotype expression values:

```
ggplot(data = CBD_plot2, aes(x=geneClass, y=MeanValue, fill=Sample)) +
  geom_bar(stat='identity', position=position_dodge())+
  scale_y_continuous(breaks = seq(0, 1300, by=100), limits=c(0,1300))+
  scale_fill_brewer(palette='Paired') +
  geom_hline(yintercept=100, linetype="dashed", color = "red")+
  theme(legend.position="bottom")+
  #ggtitle('CBD Network Gene levels > 100 Treated with CoV')+
  ylab('Gene Expression Mean Values')
```

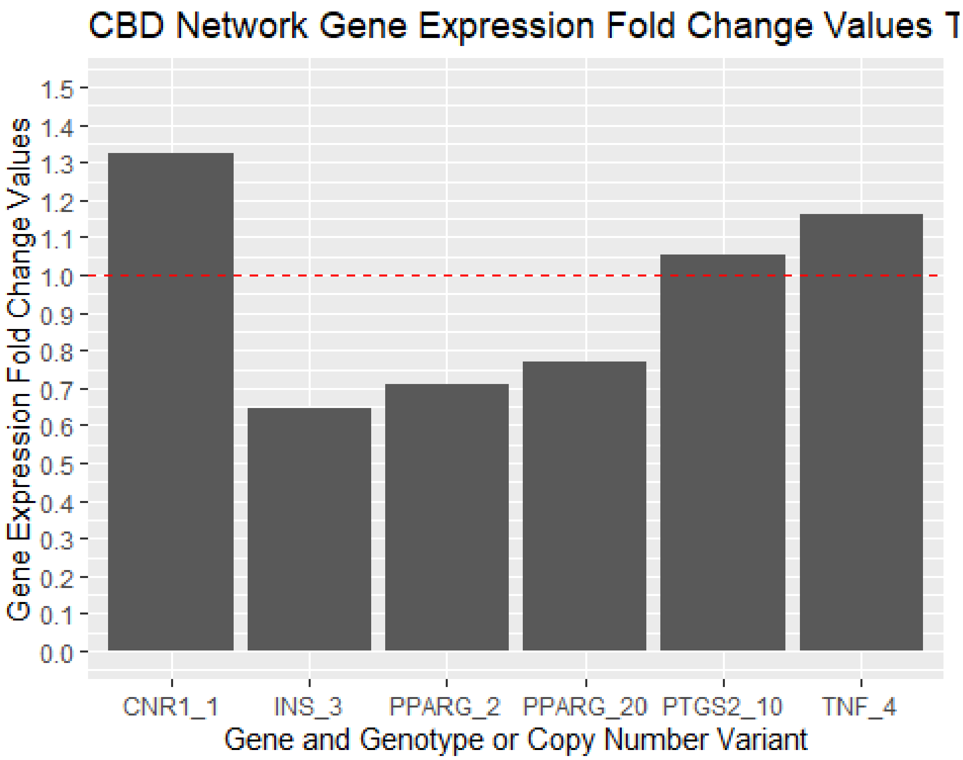


The following is the fold change chart.

```
CBD_plot$FoldChange <- CBD_plot$CoV_Genotype_Mean/CBD_plot$Ctrl_Geneotype_Mea
n
CBD_plot3 <- CBD_plot[,c(1:9,12,10,11,13)] #fold change plot data
CBD_plot2$logMean <- log2(CBD_plot2$MeanValue) # log2 scaled plot data
```

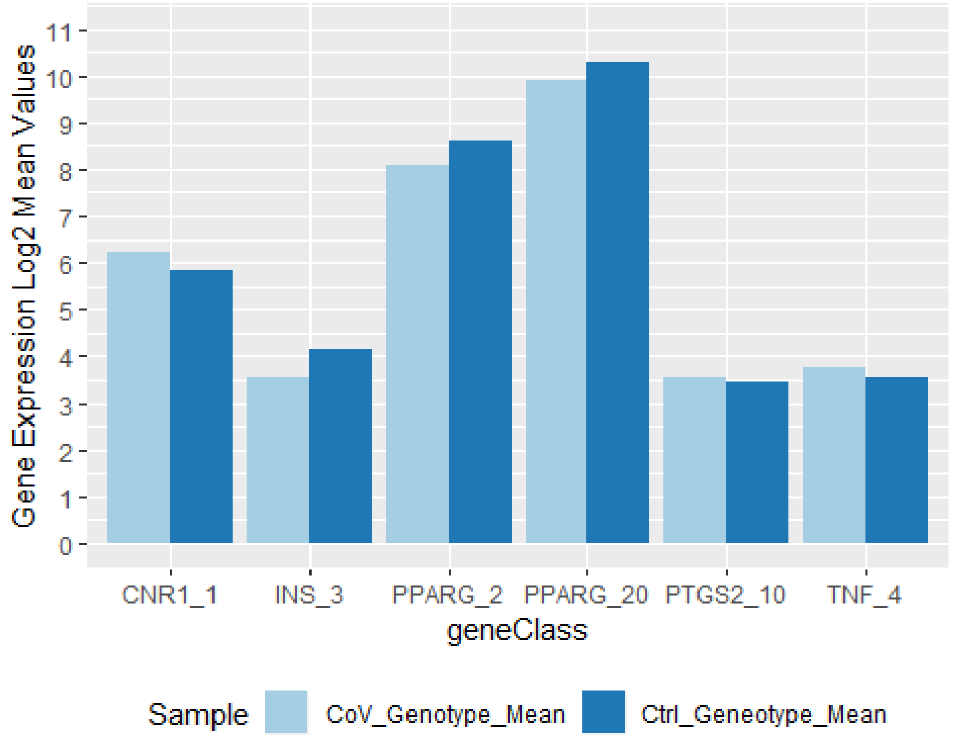
The chart of fold change values of the genotype or single nucleotide polymorphism (SNP) or copy number variant of the CBD gene network top five listed genes by Relevance.score.with an added dashed red line to show that genes below the line decreased and those above increased in percentage of gene expression of CoV/ctrl.

```
ggplot(data = CBD_plot3, aes(x=geneClass, y=FoldChange)) +
  geom_bar(stat='identity', position=position_dodge())+
  scale_y_continuous(breaks = seq(0, 1.5, by=.1), limits=c(0,1.5))+
  scale_fill_brewer(palette='Paired') +
  geom_hline(yintercept=1, linetype="dashed", color = "red")+
  ggtitle('CBD Network Gene Expression Fold Change Values Treated with CoV')+
  xlab('Gene and Genotype or Copy Number Variant')+
  ylab('Gene Expression Fold Change Values')
```



The next chart is the log2 scaled Mean values for the CBD genotypes.

```
ggplot(data = CBD_plot2, aes(x=geneClass, y=logMean, fill=Sample)) +
  geom_bar(stat='identity', position=position_dodge())+
  scale_y_continuous(breaks = seq(0, 11, by=1), limits=c(0,11))+
  scale_fill_brewer(palette='Paired') +
  theme(legend.position="bottom") +
  #ggtitle('CBD Network Gene Expression Log2 Mean Values Treated with Coronav
irus')+
  ylab('Gene Expression Log2 Mean Values')
```



Those visuals help establish that the PPARG has two copy number variants of the gene that are expressed mush more in both the control and CoV inoculated samples by mean genotype expression values, fold change values, and log2 scaled values. We can see that CoV infected samples after 1 hour in liver tumor samples increases genotype expressions of CNR_1, PTGS2, and TNF. And that INS and PPARG are decreased in the first hour of inoculation with CoV.

Now, lets start working on the data that confirms or better asserts what is going on with these genes. Recall that we wanted to examine three possibilities to understand why we made the observation above on the what the data provided for the cannabidiol genes network.

1.) Analyzing the **time sequence of the blood capillary samples** inoculated with CoV and their respective control groups. Examining how these CBD genes are effected over time.

2.) Confirming that insulin is effected by CoV by looking at the **diabetic gene networks** for type 1 and type 2 diabetes.

3.) Also, looking at the **tumorigenesis gene network** to see how the genes that are involved in tumor growth in the body react to CoV in the liver tumor samples and the blood capillary samples.