Analysis of retinal nerve crush RNA-seq data from Libby lab (updated)

Stanley Yang

1/15/2018

Update Notes on analysis based on our discussion on 1-4-18:

- changed Genotype of sample 42013 from Ddit3_Jun to Ddit3
- included the sex, generation, pen number information into PCA analysis
- increased the dimensional view of the MDS plot and PCA plot (Dim1 vs Dim2, Dim2 vs Dim3, Dim1 vs Dim3)
- increased the comparisons of interaction of genotype and treatment among DJ, D, and J groups (a4, a5, a6)
- generated KEGG and GO analysis based on both glmTreat_1 list (DE genes regardless of how small the fold changes is) and glmTreat_1.2 list (DE gene for at least 1.2 fold change in either direction)

```
## load rna-seq count file and design file
rna.file = "../conc_libby_new/data/Glaucoma_all_gene_counts.txt"
design.file = "./data/design_file.txt"

data.design = read.table(design.file,  sep="\t", head=T, quote="", check.names=F)

data.raw = read.table(rna.file,  sep="\t", head=T, quote="", check.names=F, row.names=1)
colnames(data.raw)= data.design$ID_simple

group=factor(data.design$Group)
```

Summary for all samples

```
# library(dplyr)
data.design %>% group_by(Group) %>% summarise(N=n())
```

```
# A tibble: 8 x 2
  Group              N
  <fct>          <int>
1 Control.CONC       5
2 Control.DNT        5
3 Ddit3_Jun.CONC     4
4 Ddit3_Jun.DNT      4
5 Ddit3.CONC         5
6 Ddit3.DNT          5
7 Jun.CONC           5
8 Jun.DNT            5
```

Have changed Genotype of sample 42013 from Ddit3_Jun to Ddit3

Normalize and Filter the raw RNA-seq data

before normalization

```
library(edgeR)
y <- DGEList(data.raw, group=group, genes=row.names(data.raw)) # must specify
options(digits=3)
y$samples[,-1]
```

```
                      lib.size norm.factors
Ddit3_Jun.CONC.41891L 24118862            1
Ddit3_Jun.DNT.41891R  27797048            1
Ddit3_Jun.CONC.42011L 13623081            1
Ddit3_Jun.DNT.42011R  21684945            1
Ddit3.CONC.42013L     26336538            1
Ddit3.DNT.42013R      29050767            1
Ddit3_Jun.CONC.42014L 21529183            1
Ddit3_Jun.DNT.42014R  28157436            1
Ddit3.CONC.42142L     18971507            1
Ddit3.DNT.42142R      23336377            1
Ddit3.CONC.42229L     24679417            1
Ddit3.DNT.42229R      36803741            1
Ddit3_Jun.CONC.42238L 27703893            1
Ddit3_Jun.DNT.42238R  22776189            1
Control.CONC.42242L   19149517            1
Control.DNT.42242R    23274525            1
Jun.CONC.42244L       16276929            1
Jun.DNT.42244R        25051935            1
Jun.CONC.42288L       25246790            1
Jun.DNT.42288R        23568947            1
Control.CONC.42302L   41994549            1
Control.DNT.42302R    30538919            1
Control.CONC.42366L   45520014            1
Control.DNT.42366R    23455716            1
Jun.CONC.42376L       29882693            1
Jun.DNT.42376R        30244497            1
Jun.CONC.42379L       19800761            1
Jun.DNT.42379R        27301063            1
Control.CONC.42601L   26979460            1
Control.DNT.42601R    18361890            1
Control.CONC.42603L   22093399            1
Control.DNT.42603R    27200990            1
Jun.CONC.42604L       27377613            1
Jun.DNT.42604R        21595708            1
Ddit3.CONC.42645L     22320142            1
Ddit3.DNT.42645R      44909435            1
Ddit3.CONC.42647L     26916559            1
Ddit3.DNT.42647R      23388129            1
```

```
## symbols, message=FALSE
library(org.Mm.eg.db)
y$genes$Symbol <- mapIds(org.Mm.eg.db, rownames(y),
                         keytype="ENSEMBL", column="SYMBOL")  # keytype="ENSEMBL", attach gene Symbol f
y$genes$Entrezid <- mapIds(org.Mm.eg.db, rownames(y),
                           keytype="ENSEMBL", column="ENTREZID")
y$genes$Genename <- mapIds(org.Mm.eg.db, rownames(y),
                           keytype="ENSEMBL", column="GENENAME")
```

before filtering low-count genes

```
before_sum<-data.frame(dim(y$counts))
rownames(before_sum) <- c("total number of genes detected", "total sample number")
before_sum
```

```
                               dim.y.counts.
total number of genes detected         38924
total sample number                       38
```

keep the genes that have more than 1 count per million (cpm) in at least 2 samples

```
## dropNAsymbols
y <- y[!is.na(y$genes$Symbol),]
## keep
keep <- rowSums(cpm(y) > 1) >= 2
# table(keep)
## ----filter------------------------------------------------------------
y <- y[keep, , keep.lib.sizes=FALSE]
after_sum<-data.frame(dim(y$counts))
rownames(after_sum) <- c("total number of genes after filtering", "total sample number")
after_sum
```

```
                                      dim.y.counts.
total number of genes after filtering         14186
total sample number                              38
```

normalize the filtered genes across all samples

```
## ----norm--------------------------------------------------------------
y <- calcNormFactors(y)
# y$samples %>% select(-group)
    ## unknown problem with select function
    ## Error in (function (classes, fdef, mtable) : unable to find an inherited method for function ‘se
y$samples[,-1]
```

```
                      lib.size norm.factors
Ddit3_Jun.CONC.41891L 22418619        0.987
Ddit3_Jun.DNT.41891R  25812270        0.973
Ddit3_Jun.CONC.42011L 13152421        1.022
Ddit3_Jun.DNT.42011R  20024287        1.002
Ddit3.CONC.42013L     24567299        0.992
Ddit3.DNT.42013R      27392590        0.990
Ddit3_Jun.CONC.42014L 20416988        0.990
Ddit3_Jun.DNT.42014R  25872129        0.983
Ddit3.CONC.42142L     18210742        0.992
Ddit3.DNT.42142R      22714707        0.992
Ddit3.CONC.42229L     22883766        0.996
Ddit3.DNT.42229R      34730637        0.995
Ddit3_Jun.CONC.42238L 26871390        0.994
Ddit3_Jun.DNT.42238R  21173382        0.968
Control.CONC.42242L   17774749        1.004
Control.DNT.42242R    21333689        1.002
Jun.CONC.42244L       15581340        1.013
Jun.DNT.42244R        23684446        1.000
```

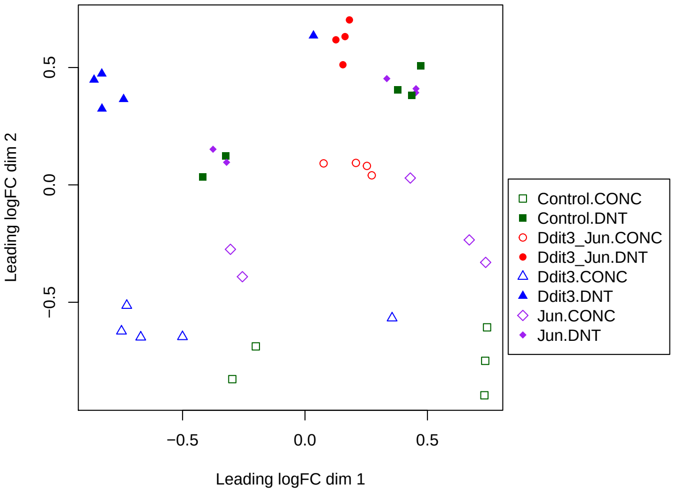

Figure 1: MDS plot

```
Jun.CONC.42288L       23344317        1.015
Jun.DNT.42288R        21749652        1.014
Control.CONC.42302L   38306363        1.025
Control.DNT.42302R    28335367        0.976
Control.CONC.42366L   42956562        1.029
Control.DNT.42366R    21836380        1.001
Jun.CONC.42376L       27608108        1.020
Jun.DNT.42376R        28042325        1.012
Jun.CONC.42379L       18542765        0.997
Jun.DNT.42379R        25189469        0.971
Control.CONC.42601L   24652872        1.023
Control.DNT.42601R    17327605        1.008
Control.CONC.42603L   21258676        1.017
Control.DNT.42603R    25782751        1.009
Jun.CONC.42604L       25286325        1.020
Jun.DNT.42604R        20801628        1.009
Ddit3.CONC.42645L     21232153        0.995
Ddit3.DNT.42645R      42606661        0.995
Ddit3.CONC.42647L     24809101        0.976
Ddit3.DNT.42647R      21621291        0.999
```

Explore the data samples

MDS plot

to check the distance between each sample (Figure 1-3), similar to PCA plots

```
## ----mdsplot # what's difference to plot log.cpm or the whole object y?
par(xpd = T, mar = par()$mar + c(0,0,0,7))  # to make legends outside of the plot
colors <- c("darkgreen","darkgreen","red","red", "blue","blue", "purple", "purple")
pch <- c(0,15, 1, 16, 2, 17, 5, 18)
plotMDS(y, top = 500, cex = 1, pch=pch[group], dim.plot = c(1,2), ndim = 3, gene.selection = "pairwise"
legend(0.83, 0.025,levels(group), pch=pch, col=colors)
```

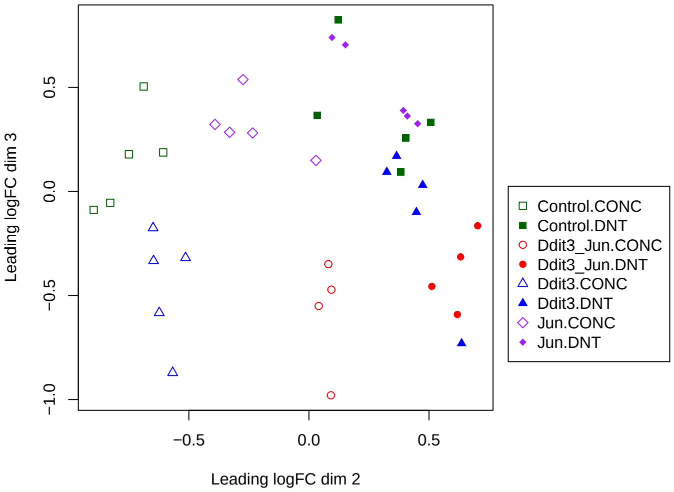

Figure 2: MDS plot

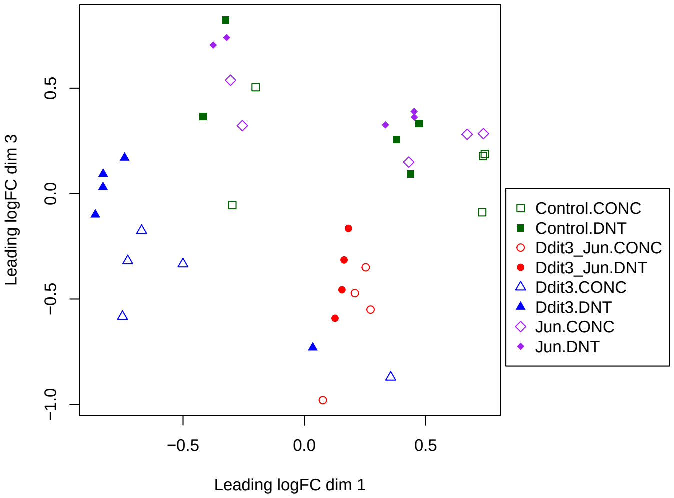

Figure 3: MDS plot

```
par(mar=c(5, 4, 4, 2.5) + 0.1)

par(xpd = T, mar = par()$mar + c(0,0,0,7))  # to make legends outside of the plot
colors <- c("darkgreen","darkgreen","red","red", "blue","blue", "purple", "purple")
pch <- c(0,15, 1, 16, 2, 17, 5, 18)
plotMDS(y, top = 500, cex = 1, pch=pch[group], dim.plot = c(2,3), ndim = 3, gene.selection = "pairwise"
legend(0.83, 0.025,levels(group), pch=pch, col=colors)
```

```
par(mar=c(5, 4, 4, 2.5) + 0.1)


par(xpd = T, mar = par()$mar + c(0,0,0,7))  # to make legends outside of the plot
colors <- c("darkgreen","darkgreen","red","red", "blue","blue", "purple", "purple")
pch <- c(0,15, 1, 16, 2, 17, 5, 18)
plotMDS(y, top = 500, cex = 1, pch=pch[group], dim.plot = c(1,3), ndim = 3, gene.selection = "pairwise"
legend(0.83, 0.025,levels(group), pch=pch, col=colors)
```

```
par(mar=c(5, 4, 4, 2.5) + 0.1)
```

```
## ----design------------------------------------------------------------
design <- model.matrix(~0+group)
colnames(design) <- levels(group)
design

## ----estimateDisp------------------------------------------------------
y <- estimateDisp(y, design, robust=TRUE)

## ----plotBCV, width="3.8in", fig.cap="Scatterplot of the biological coefficient of variation (BCV) aga
# plotBCV(y)

## ----glmQLFit----------------------------------------------------------
fit <- glmQLFit(y, design, robust=TRUE)
head(fit$coefficients)

## ----QLDisp, out.width="3.8in", fig.cap="A plot of the quarter-root QL dispersion against the average
# plotQLDisp(fit)

## ----df.prior----------------------------------------------------------
summary(fit$df.prior)
```

Principal Component Analysis (PCA)

scree plot to show all possible components for variance explained (Figure 4, 5)

```
## ----cpm---------------------------------------------------------------
logCPM <- cpm(y, prior.count=2, log=TRUE)
logCPM.PCA<-logCPM # save it later for PCA plot

## ---Pincicpal component analysis ---
pca_original = prcomp(t(logCPM.PCA),scale=T, center=T)
pca_x <- pca_original$x
pca_table <- data.frame(pca_x, data.design)
x <- pca_original$sdev^2/sum(pca_original$sdev^2) # Proportion of Variance Explained for all components

## Scree plot
plot(x, xlab="Principal Component", ylab="Proportion of Variance Explained", type="b")
```

```
plot(cumsum(x), xlab="Principal Component", ylab="Cumulative Proportion of Variance Explained", type="b
```

- PCA plot to show the first, second and third components (plotted as PC1 vs PC2, PC2 vs PC3, PC1 vs PC3)
- There are no obvious known factors that can explain the first component (Figure 6-8).
- The treatment (conc) can explain the variance of the second component.
- The genotype may partly explain the variance of the third component: most control (circle) and Jun (cross) seems to distribute above 0 on PC3 axis, whereas most Ddit3 (triangle) and Ddit3-Jun (square) seems to distribute below 0 on PC3 axis. (Figure 7)
- I also label the sample based on the sex (Figure 9-11), generation (Figure 12-14), and pen number in PCA plots (Figure 15-17). No obvious pattern is observed, meaning non of these factor drives the variance of gene expression (which is good).

```
## ---Pincicpal component analysis ---
```

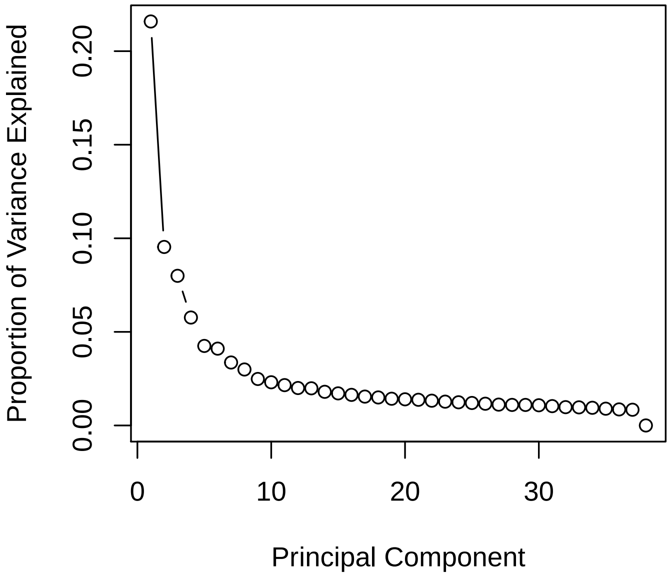

Figure 4: Scree Plot

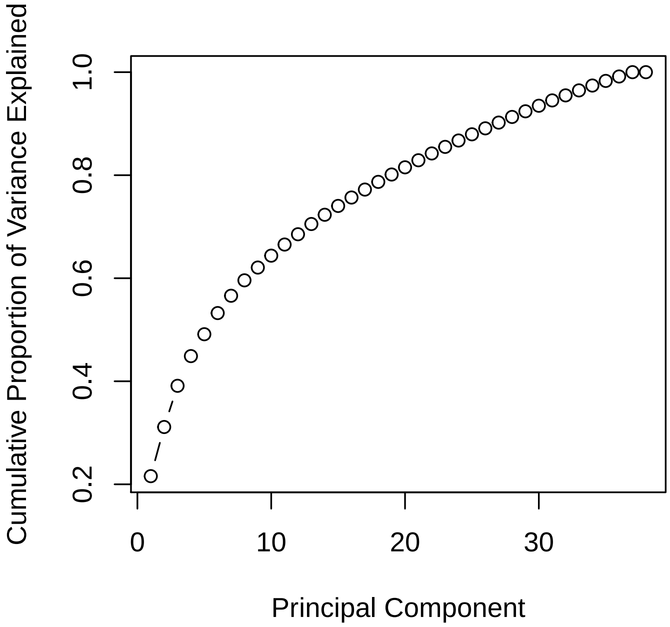

Figure 5: Scree Plot

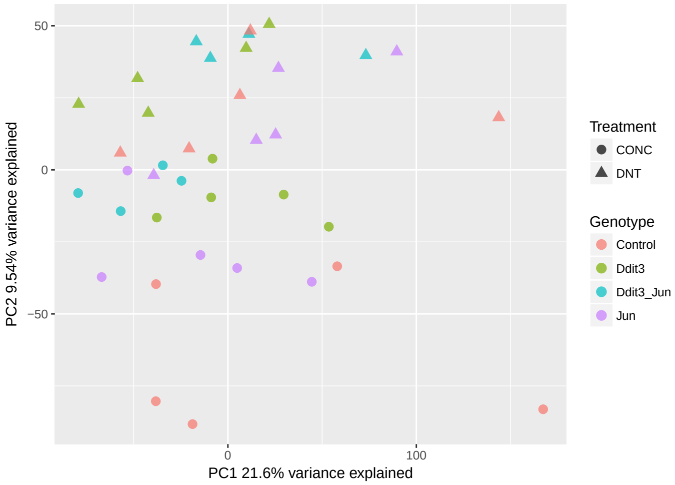

Figure 6: PCA Plot

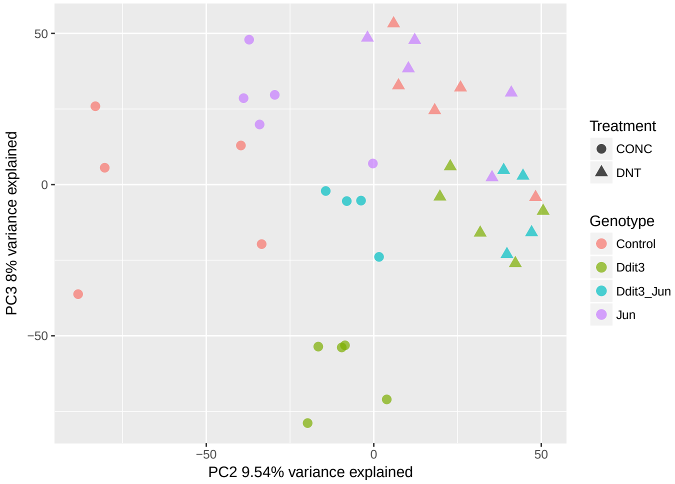

Figure 7: PCA Plot

```
PCA_plot <- function(pca_table, PC_x, PC_y, color, shape){
  #PC_x,PC_y are type of interger
  #color, shape, are type of string
  g <- ggplot(pca_table, aes_string(x=names(pca_table[PC_x]), y=names(pca_table[PC_y]), color=color, sha
  g <- g + geom_point(alpha=0.7, size=3)
  # g <- g + labs(color = "Group", shape="Tissue")
  g + labs(x = paste(names(pca_table[PC_x]), scales::percent(x[PC_x]),"variance explained", sep=" "), y=
  #filename <- paste()
  #ggsave(filename, width=7, height=7, units="in")
}

PCA_plot(pca_table, 1, 2, "Genotype", "Treatment")
```

```
PCA_plot(pca_table, 2, 3, "Genotype", "Treatment")
```

```
PCA_plot(pca_table, 1, 3, "Genotype", "Treatment")
```

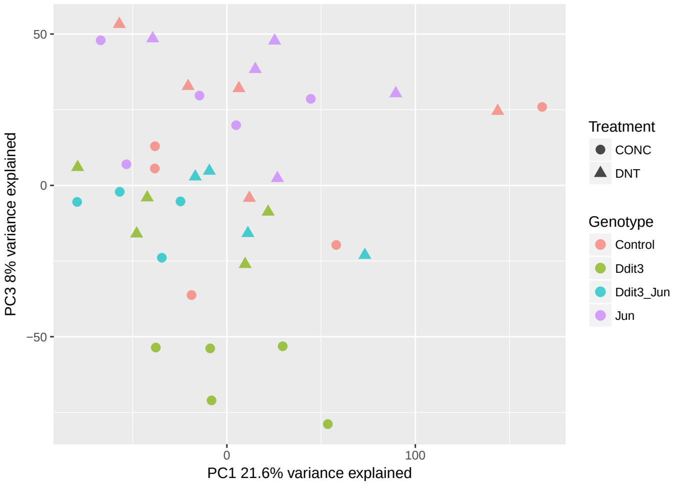

Figure 8: PCA Plot

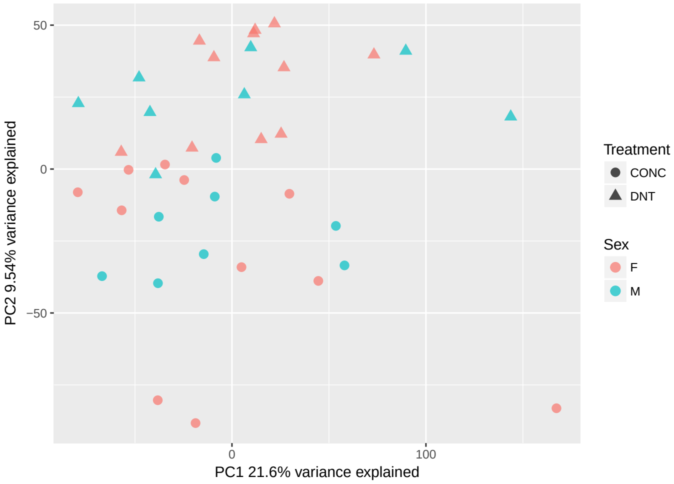

Figure 9: PCA Plot

```
PCA_plot(pca_table, 1, 2, "Sex", "Treatment")
```

```
PCA_plot(pca_table, 2, 3, "Sex", "Treatment")
```

```
PCA_plot(pca_table, 1, 3, "Sex", "Treatment")
```

```
PCA_plot(pca_table, 1, 2, "Gen", "Treatment")
```

```
PCA_plot(pca_table, 2, 3, "Gen", "Treatment")
```

```
PCA_plot(pca_table, 1, 3, "Gen", "Treatment")
```

```
PCA_plot(pca_table, 1, 2, "Pen", "Treatment")
```

```
PCA_plot(pca_table, 2, 3, "Pen", "Treatment")
```

```
PCA_plot(pca_table, 1, 3, "Pen", "Treatment")
```

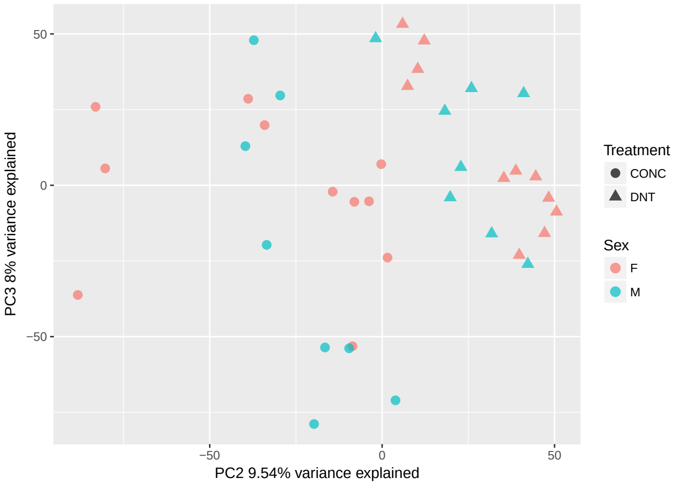

Figure 10: PCA Plot

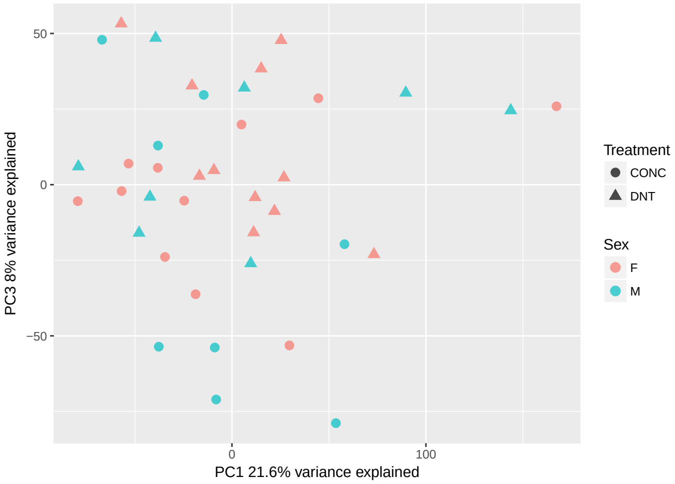

Figure 11: PCA Plot

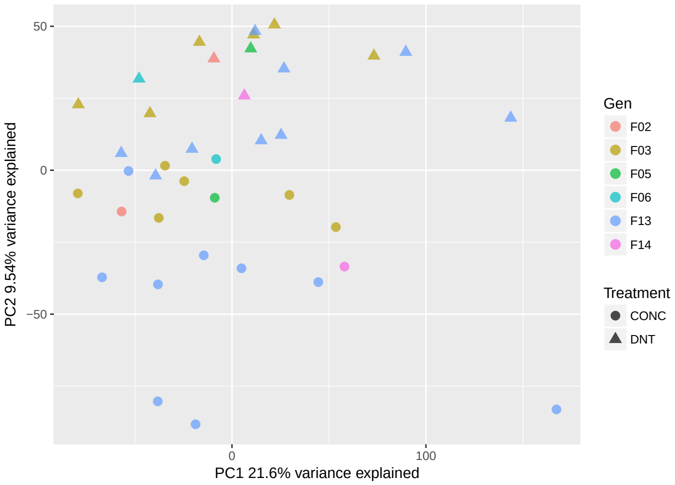

Figure 12: PCA Plot

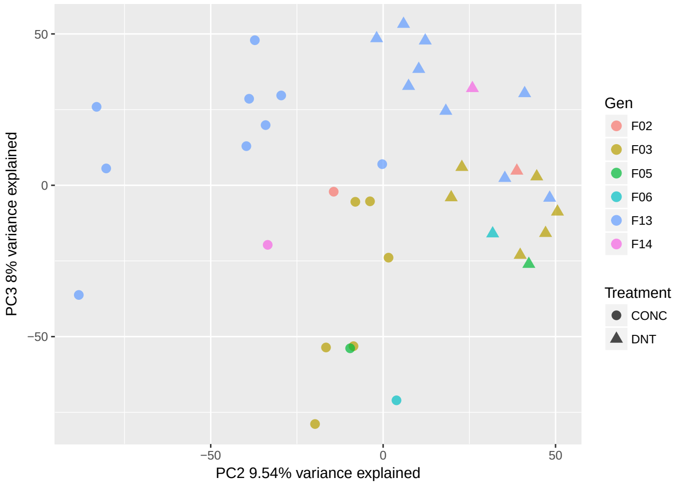

Figure 13: PCA Plot

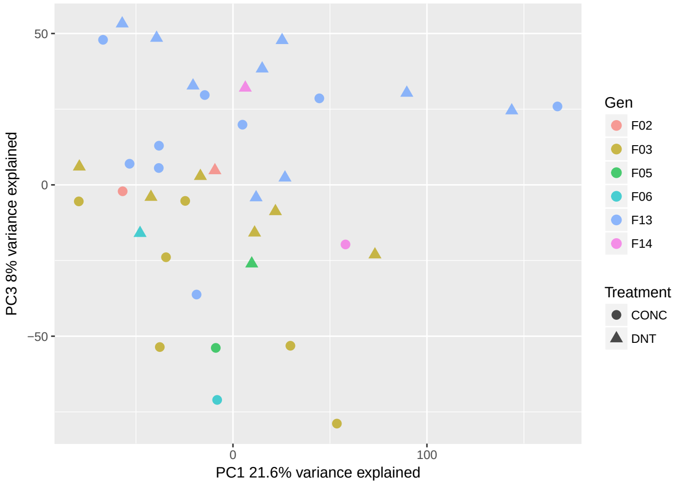

Figure 14: PCA Plot

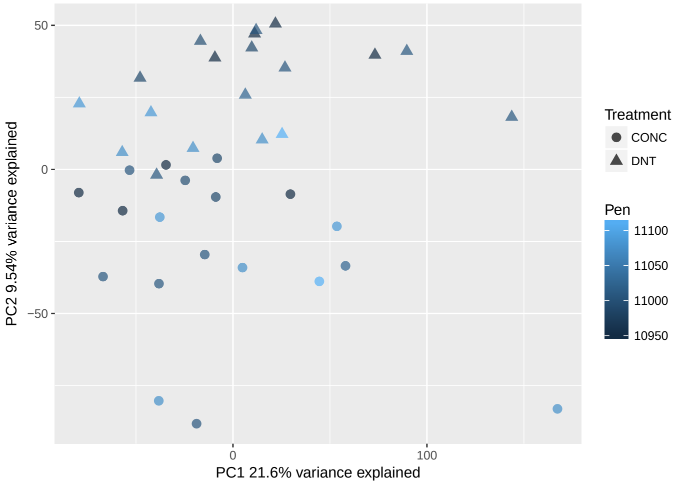

Figure 15: PCA Plot

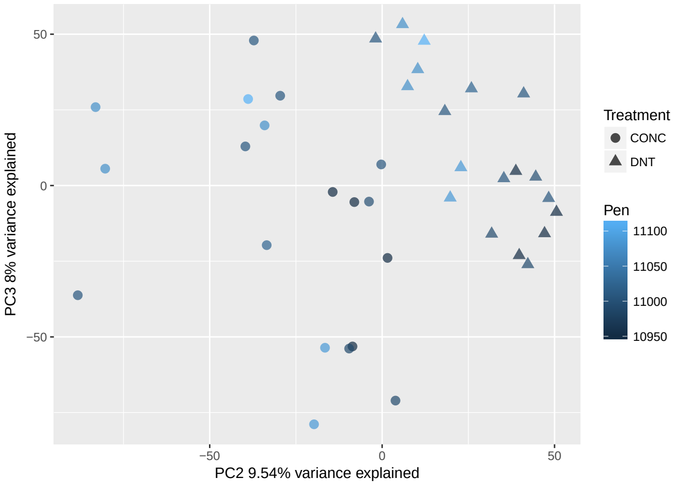

Figure 16: PCA Plot

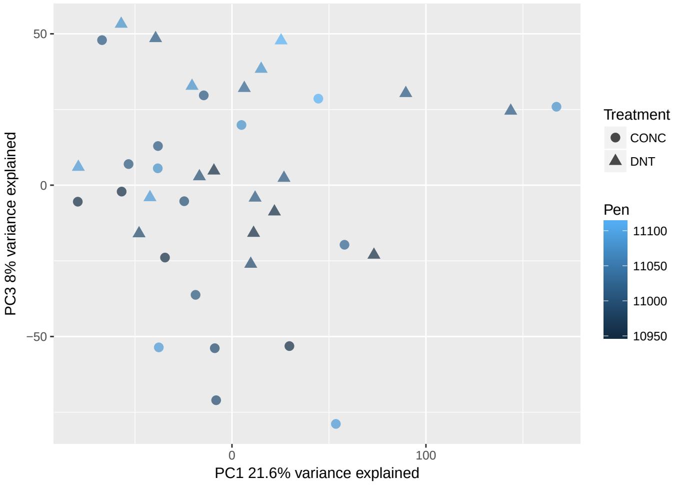

Figure 17: PCA Plot

Differential expression analysis

testing for differential expression

1. Methods for DE gene analysis

- I use generalized linear model based quasi-likelihood (QL) F-tests (glmQLFtest) instead of likelihood ratio test (LRT) for find DE genes as they give stricter error rate control by accounting for the uncertainty in dispersion estimation. (The old DE gene lists were made by LRT methods)
- There are two kinds of QL F-tests used in DE gene analysis, they are marked in the output files
  - glmTreat_1 : identifies differential expression based on statistical significance (*FDR < 0.05* as a cutoff) regardless of how small the difference might be. (1 means the FC = 1)
  - glmTreat_1.2 : identifies the differential expression fold changes are significantly greater than a specified fold change which is 1.2 in this case. (1.2 means FC=1.2, can be changed)

2. list of all the comparisions and their meanings:

- a1_DJvsC = (Ddit3_Jun.CONC - Ddit3_Jun.DNT) - (Control.CONC - Control.DNT)
  - The difference between Control mice and Ddit3-Jun mice in response to CONC crush treatment (interaction effect between genotype and treatment)
  - a1_DJvsC will be attached to the exported file names (as prefix) indicating this comparison. Same rule for the rest of the comparsions.
- a2_JvsC = (Jun.CONC - Jun.DNT) - (Control.CONC - Control.DNT)
  - The difference between Control mice and Jun mice in response to CONC crush treatment (interaction effect between genotype and treatment)
- a3_DvsC = (Ddit3.CONC - Ddit3.DNT) - (Control.CONC - Control.DNT)
  - The difference between Control mice and Ddit3 mice in response to CONC crush treatment (interaction effect between genotype and treatment)
- a4_DJvsJ = (Ddit3_Jun.CONC - Ddit3_Jun.DNT) - (Jun.CONC - Jun.DNT)
  - The difference between Ddit3_Jun mice and Jun mice in response to CONC crush treatment (interaction effect between genotype and treatment)
- a5_DJvsD = (Ddit3_Jun.CONC - Ddit3_Jun.DNT) - (Ddit3.CONC - Ddit3.DNT)
  - The difference between Ddit3_Jun mice and Ddit3 mice in response to CONC crush treatment (interaction effect between genotype and treatment)
- a6_DvsJ = (Ddit3.CONC - Ddit3.DNT) - (Jun.CONC - Jun.DNT)
  - The difference between Ddit3 mice and Jun mice in response to CONC crush treatment (interaction effect between genotype and treatment)
- a7_CONCvsDNT = Ddit3_Jun.CONC + Jun.CONC + Ddit3.CONC + Control.CONC - (Ddit3_Jun.DNT + Jun.DNT + Ddit3.DNT + Control.DNT)
  - The average differences between CONC and DNT treatment in mice of all genotypes
- a8_DJvsC.CONC = Ddit3_Jun.CONC - Control.CONC
  - The difference between Ddit3-Jun and Control mice with CONC treatment
- a9_JvsC.CONC = Jun.CONC - Control.CONC
  - The difference between Jun and Control mice with CONC treatment
- a10_DvsC.CONC = Ddit3.CONC - Control.CONC
  - The difference between Ddit3 and Control mice with CONC treatment
- a11_CONCvsDNT.DJ = Ddit3_Jun.CONC - Ddit3_Jun.DNT
  - The effect of CONC treatment on Ddit3-Jun mice
- a12_CONCvsDNT.J = Jun.CONC - Jun.DNT
  - The effect of CONC treatment on Jun mice
- a13_CONCvsDNT.D = Ddit3.CONC - Ddit3.DNT
  - The effect of CONC treatment on Ddit3 mice
- a14_CONCvsDNT.C = Control.CONC - Control.DNT
  - The effect of CONC treatment on Control mice
- a1_DJvsC-a2_JvsC-a3_DvsC
  - ANOVA-like tests to find any difference between Ddit3-Jun or Ddit3 or Jun mice and Control mice

    in response to CONC crush treatment (ANOVA-like tests on interaction effects between genotype and treatment)

- a8_DJvsC.CONC-a9_JvsC.CONC-a10_DvsC.CONC
    - ANOVA-like tests to find difference between Ddit3-Jun or Ddit3 or Jun mice and Control mice with CONC treatment
- a11_CONCvsDNT.DJ-a12_CONCvsDNT.J-a13_CONCvsDNT.D-a14_CONCvsDNT.C
    - ANOVA-like tests to find the effect of CONC treatment regardless of mouse genotype

3. Pathway analysis

- Gene ontology analysis:
    - all the genes from glmTreat lists with FDR<0.05 were put into gene ontology analysis.
    - The **Up** and **Down** columns indicate the number of genes within the GO term that are sigificantly up- and down-regulated in this differential expression comparison, respectively. The **P.Up** and **P.Down** columns contain the p-values for over-representation of the GO term in the up- and down-regulated genes, respectively.
    - GO terms with p-value less than 10^{-5} were kept.
- KEGG pathway analysis
    - all the genes from glmTreat lists with FDR<0.05 were put into KEGG analysis.
    - same meaning for **Up** and **Down** ,and **P.Up** and **P.Down** columns as in GO terms
    - I kept p-value < 0.05 for KEGG analysis. May need $p < 10^{-5}$ for more stringent threthold.
- For detailed analysis, you are encoraged to put the gene lists of your interested comparison into david kegg pathway analysis tool. https://david.ncifcrf.gov/summary.jsp

4. Decode output files

Take a4_CONCvsDNT comparison as an example: first refer to comparison table to find out this comparison means the average differences between CONC and DNT treatment in mice of all genotypes, it contains four files starting with libby1 followed by the name of this comparison: * libby3_a7_CONCvsDNT_glmTreat_1.txt + glmQLFTest for significant DE genes no matter how small the change is + can further maually put the DE genes into KEGG or GO online tool for detailed analysis * libby3_a7_CONCvsDNT_glmTreat_1.2.txt + glmTreat for significant DE genes that has a fold change greater than 1.2 in either direction * libby3_a7_CONCvsDNT_KEGG_1.txt + KEGG analysis of DE genes (FDR<0.05) from glmTreat_1 file + not all the comparison has an output file of KEGG analysis, only the gene sets meet KEGG analysis p-value criteria will have this output file. Same as gene ontology analysis. * libby3_a7_CONCvsDNT_Ont_1.txt + Gene Ontology analysis of DE genes (FDR<0.05) from glmTreat_1 file * libby3_a7_CONCvsDNT_KEGG_1.2.txt + KEGG analysis of DE genes (FDR<0.05) from glmTreat_1.2 file * libby3_a7_CONCvsDNT_Ont_1.2.txt + Gene Ontology analysis of DE genes (FDR<0.05) from glmTreat_1.2 file

heatmap visualization of sample clustering

- Heatmaps are a popular way to display DE results for publicaiton pruposes. Here I generated a sample heatmap based on top 100 DE genes according to **TREAT** test between CONC and DNT across all genotypes (comparison code: a7_CONCvsDNT), but specified the significant fold change at 1.5 (Figure 18).
- This time with corrected genotype information, the samples were clustered perfectly under CONC treatment based on their genotype.

```
## ----MakeContrasts---------------------------------------------------------------
con <- makeContrasts(
  a1_DJvsC = (Ddit3_Jun.CONC - Ddit3_Jun.DNT) - (Control.CONC - Control.DNT),
  a2_JvsC = (Jun.CONC - Jun.DNT) - (Control.CONC - Control.DNT),
  a3_DvsC = (Ddit3.CONC - Ddit3.DNT) - (Control.CONC - Control.DNT),
  a4_DJvsJ = (Ddit3_Jun.CONC - Ddit3_Jun.DNT) - (Jun.CONC - Jun.DNT),
  a5_DJvsD = (Ddit3_Jun.CONC - Ddit3_Jun.DNT) - (Ddit3.CONC - Ddit3.DNT),
  a6_DvsJ = (Ddit3.CONC - Ddit3.DNT) - (Jun.CONC - Jun.DNT),
  a7_CONCvsDNT = Ddit3_Jun.CONC + Jun.CONC + Ddit3.CONC + Control.CONC - (Ddit3_Jun.DNT + Jun.DNT + Ddi
  a8_DJvsC.CONC =  Ddit3_Jun.CONC - Control.CONC,
  a9_JvsC.CONC = Jun.CONC - Control.CONC,
  a10_DvsC.CONC = Ddit3.CONC - Control.CONC,
  a11_CONCvsDNT.DJ = Ddit3_Jun.CONC - Ddit3_Jun.DNT,
  a12_CONCvsDNT.J = Jun.CONC - Jun.DNT,
  a13_CONCvsDNT.D = Ddit3.CONC - Ddit3.DNT,
  a14_CONCvsDNT.C = Control.CONC - Control.DNT,
  levels=design
)

tr <- glmTreat(fit, contrast=con[,7], lfc=log2(1.5))

logCPM <- cpm(y, prior.count=2, log=TRUE)
logCPM.PCA<-logCPM # save it later for PCA plot
rownames(logCPM) <- y$genes$Symbol
#colnames(logCPM) <- paste(y$samples$group, 1:2, sep="-")
colnames(logCPM) <- data.design$ID_simple # get it into the y project

## ----order---------------------------------------------------------------
o <- order(tr$table$PValue)
logCPM <- logCPM[o[1:100],]

## ----scale---------------------------------------------------------------
logCPM <- t(scale(t(logCPM)))

## ----heatmap, message=FALSE, fig.width=8, fig.height=12, fig.cap="Heat map across all the samples usi
library(gplots)
col.pan <- colorpanel(100, "blue", "white", "red")
heatmap.2(logCPM, col=col.pan, Rowv=TRUE, scale="none",
          trace="none", dendrogram="both", cexRow=0.5, cexCol=0.7, density.info="none",
          margin=c(10,9), lhei=c(2,10), lwid=c(2,6))
```

Ddit3_Jun_CONC_42013L (second sample from the left) is clustered with Ddit3_CONC group. Need to recheck the genotype.
Solution: we can either exclude the samples in the analysis or treat them as Ddit3 genotype instead of Ddit3-Jun double mutant.

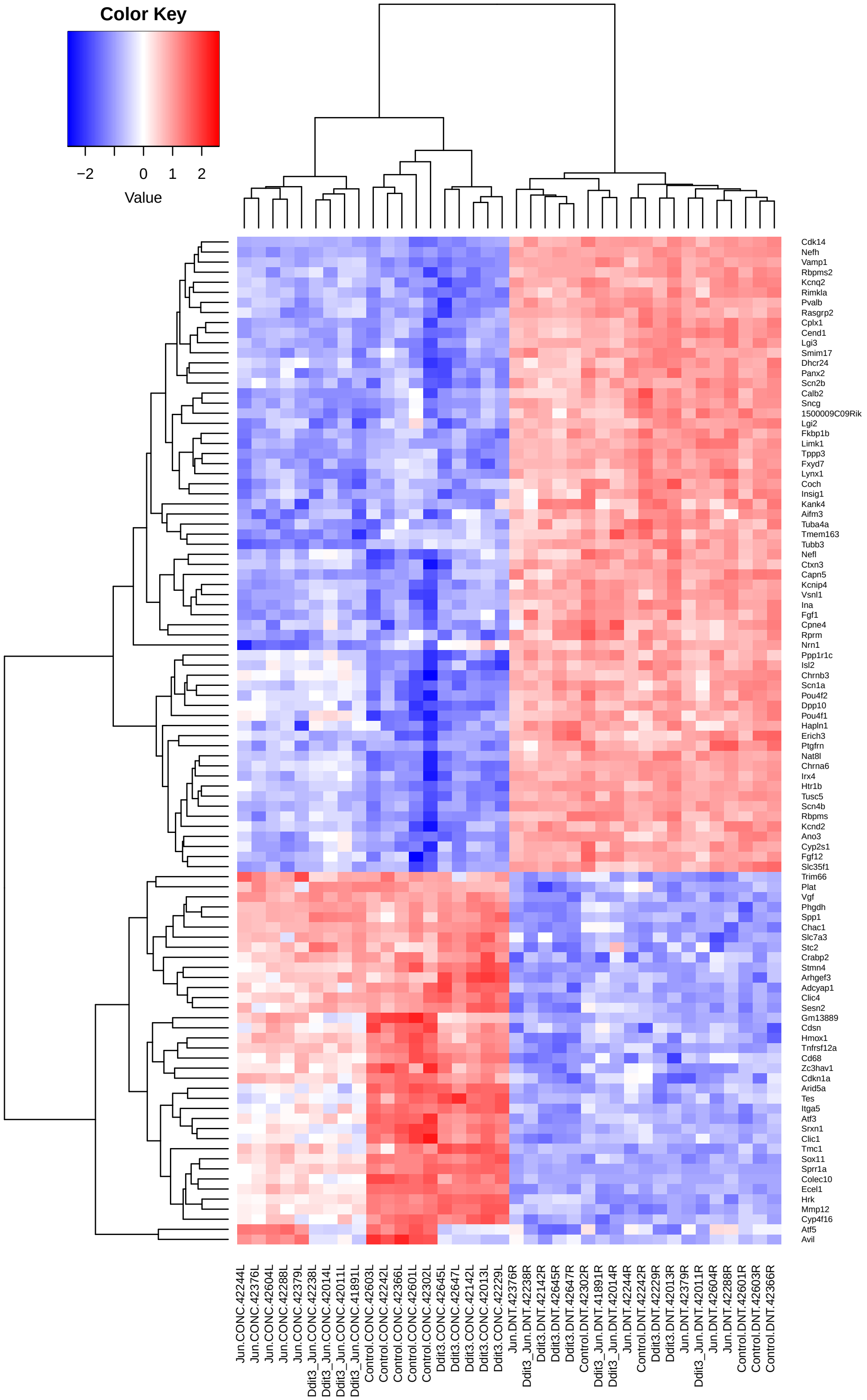

Figure 18: heat map across all the samples using the top 100 most DE genes between CONC and DNT groups of all genotypes