Astrocyte reactivity: RNA-seq data analysis

[illegible]

[illegible]

Orignal paper title:

Modulation of astrocyte reactivity improves functional deficits in mouse models of Alzheimer's disease https://doi.org/10.1186/s40478-018-0606-1

Expring GEO (Gene Expression Omnibus)

GEO - Gene Expression Omnibus
GDS - GEO DataSet
GSE - GEO Series
GPL - GEO Platform
https://www.ncbi.nlm.nih.gov/geo/info/faq.html

Download GEO data and create the GEOquery object

```
gse <- getGEO('GSE108520')
```

Obtaining samples matadata:

```
class(gse)
```

```
## [1] "list"
```

```
length(gse)
```

```
## [1] 1
```

```
gse <- gse[[1]]
class(gse)
```

```
## [1] "ExpressionSet"
## attr(,"package")
## [1] "Biobase"
```

```
pheno     <- pData(gse) ## print the sample information

glimpse(pheno, width=80)
```

```
## Rows: 19
## Columns: 42
## $ title                      <chr> "Astro-WT-GFP-1", "Astro-WT-GFP-2", "Astro-W~
## $ geo_accession              <chr> "GSM2902723", "GSM2902724", "GSM2902725", "G~
## $ status                     <chr> "Public on Sep 28 2018", "Public on Sep 28 2~
## $ submission_date           <chr> "Dec 26 2017", "Dec 26 2017", "Dec 26 2017",~
## $ last_update_date          <chr> "Sep 28 2018", "Sep 28 2018", "Sep 28 2018",~
## $ type                      <chr> "SRA", "SRA", "SRA", "SRA", "SRA", "SRA", "S~
## $ channel_count             <chr> "1", "1", "1", "1", "1", "1", "1", "1", "1",~
## $ source_name_ch1           <chr> "astrocyte", "astrocyte", "astrocyte", "astr~
## $ organism_ch1              <chr> "Mus musculus", "Mus musculus", "Mus musculu~
## $ characteristics_ch1       <chr> "strain: C57bl6", "strain: C57bl6", "strain:~
## $ characteristics_ch1.1     <chr> "Sex: male", "Sex: male", "Sex: male", "Sex:~
## $ characteristics_ch1.2     <chr> "age: 9 month-old", "age: 9 month-old", "age~
## $ characteristics_ch1.3     <chr> "tissue: brain", "tissue: brain", "tissue: b~
## $ molecule_ch1              <chr> "total RNA", "total RNA", "total RNA", "tota~
## $ extract_protocol_ch1      <chr> "RNA was extracted with Trizol reagent, foll~
## $ extract_protocol_ch1.1    <chr> "Full length double strand cDNA librairies w~
## $ taxid_ch1                 <chr> "10090", "10090", "10090", "10090", "10090",~
## $ description               <chr> "replicate 1-astrocyte-WT-GFP-group", "repli~
## $ data_processing           <chr> "Sequencing, data quality, reads repartition~
## $ data_processing.1         <chr> "Reads were mapped using STAR_2.4.0", "Reads~
## $ data_processing.2         <chr> "Genome_build: mm10", "Genome_build: mm10", ~
## $ data_processing.3         <chr> "Supplementary_files_format_and_content: Tab~
## $ platform_id               <chr> "GPL13112", "GPL13112", "GPL13112", "GPL1311~
## $ contact_name              <chr> "Noémie,,Robil", "Noémie,,Robil", "Noémie,,R~
## $ contact_email             <chr> "noemie.robil@genosplice.com", "noemie.robil~
## $ contact_institute         <chr> "GenoSplice technology", "GenoSplice technol~
## $ contact_address           <chr> "iPEPS-ICM-Hopital de la pitié Salpétrière -~
## $ contact_city              <chr> "Paris", "Paris", "Paris", "Paris", "Paris",~
## $ `contact_zip/postal_code` <chr> "75013", "75013", "75013", "75013", "75013",~
## $ contact_country           <chr> "France", "France", "France", "France", "Fra~
## $ data_row_count            <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0",~
## $ instrument_model          <chr> "Illumina HiSeq 2000", "Illumina HiSeq 2000"~
## $ library_selection         <chr> "cDNA", "cDNA", "cDNA", "cDNA", "cDNA", "cDN~
## $ library_source            <chr> "transcriptomic", "transcriptomic", "transcr~
## $ library_strategy          <chr> "RNA-Seq", "RNA-Seq", "RNA-Seq", "RNA-Seq", ~
## $ relation                  <chr> "BioSample: https://www.ncbi.nlm.nih.gov/bio~
## $ relation.1                <chr> "SRA: https://www.ncbi.nlm.nih.gov/sra?term=~
## $ supplementary_file_1      <chr> "NONE", "NONE", "NONE", "NONE", "NONE", "NON~
## $ `age:ch1`                 <chr> "9 month-old", "9 month-old", "9 month-old",~
## $ `Sex:ch1`                 <chr> "male", "male", "male", "male", "male", "mal~
## $ `strain:ch1`              <chr> "C57bl6", "C57bl6", "C57bl6", "C57bl6", "C57~
## $ `tissue:ch1`              <chr> "brain", "brain", "brain", "brain", "brain",~
```

Now we take GSE object:

```
geo_dat <- getGEO('GSE108520', destdir=".", GSEMatrix=F, AnnotGPL=T)
```

```
mode(geo_dat)
```

```
## [1] "S4"
```

```
class(geo_dat)
```

```
## [1] "GSE"
## attr(,"package")
## [1] "GEOquery"
```

We are sure **geo_dat** is **GSE** Class

Exploring papaer information

```r
meta <- Meta(geo_dat)
attributes(meta)
```

```
## $names
##  [1] "contact_address"         "contact_city"
##  [3] "contact_country"         "contact_email"
##  [5] "contact_institute"       "contact_name"
##  [7] "contact_zip/postal_code" "contributor"
##  [9] "email"                   "geo_accession"
## [11] "institute"               "last_update_date"
## [13] "name"                    "overall_design"
## [15] "platform_id"             "platform_taxid"
## [17] "pubmed_id"               "relation"
## [19] "sample_id"               "sample_taxid"
## [21] "status"                  "submission_date"
## [23] "summary"                 "supplementary_file"
## [25] "title"                   "type"
## [27] "web_link"
```

Summary:

```r
meta$summary
```

We analyzed the transcriptional profile of astrocytes from: 1) WT mice infected with AAV-GFP 2) reactive astrocytes from 9-month old APP/PSdE9 mice infected with AAV-GFP 3) de-activated astrocytes from 9-month old APP/PSdE9 mice infected with AAV-SOCS3 "We show SOCS3 normalizes the inflammatory profile of APP astrocytes

Experiment type:

```r
meta$type
```

Expression profiling by high throughput sequencing

```r
meta$overall_design
```

Total RNA was extracted from GFP+ astrocytes isolated by FACS from WT and APP/PS1dE9 mice injected with an AAV targeting astrocytes and encoding GFP alone (controls, N=7 WT-GFP, N=4 APP-GFP) or SOCS3 and GFP (N=5 APP-SOCS3, same total viral load). Non GFP+ cells (including microglia, neurons, non infected astrocytes, called OTHER) were analyzed as well, in 3 samples of the control WT-GFP group.

Data Analysis

Getting data:

```
data_file <- meta$supplementary_file
dat <- read_delim(data_file, delim = "\t")
#dat <- read_delim("GSE108520_Deseq2_normalized_gene_expression_with_annotations.txt.gz", delim="\t")
```

```
glimpse(dat, width=80)
```

```
## Rows: 60,567
## Columns: 22
## $ `FastDB Stale ID` <chr> "GSMG0000003", "GSMG0000004", "GSMG0000005", "GSMG00~
## $ coordinates       <chr> "chr1:4496551-4499378", "chr1:4785776-4786630", "chr~
## $ symbol            <chr> "NULL", "NULL", "Lypla1", "Tcea1", "NULL", "Gm16041"~
## $ Astro_APP_GFP_2   <dbl> 0.000000, 0.000000, 206.822394, 728.622012, 0.000000~
## $ Astro_APP_GFP_4   <dbl> 48.820746, 66.256727, 442.873913, 652.977483, 25.282~
## $ Astro_WT_GFP_6    <dbl> 0.000000, 103.661382, 129.576727, 541.112413, 67.379~
## $ Astro_APP_GFP_3   <dbl> 0.000000, 51.441614, 294.800018, 479.791976, 48.4738~
## $ Other_WT_GFP_1    <dbl> 57.715874, 15.245703, 821.089980, 989.881686, 22.868~
## $ Astro_APP_SOCS_5  <dbl> 0.00000, 54.48867, 428.87854, 592.34454, 69.42911, 1~
## $ Astro_APP_SOCS_2  <dbl> 0.0000000, 64.1786841, 554.3550103, 703.1751473, 62.~
## $ Other_WT_GFP_2    <dbl> 88.474141, 12.099028, 843.907195, 851.469087, 3.0247~
## $ Astro_APP_SOCS_3  <dbl> 30.003791, 46.005813, 221.027927, 450.056865, 39.004~
## $ Astro_WT_GFP_4    <dbl> 1.967208, 55.081833, 554.752750, 609.834584, 26.5573~
## $ Astro_WT_GFP_1    <dbl> 0.000000, 127.548617, 542.081620, 681.777724, 0.0000~
## $ Astro_APP_SOCS_4  <dbl> 0.000000, 54.889713, 290.131341, 465.582388, 52.9293~
## $ Astro_WT_GFP_3    <dbl> 0.00000, 0.00000, 718.01402, 766.27398, 49.43703, 87~
## $ Astro_WT_GFP_7    <dbl> 11.279031, 78.953217, 503.044781, 584.253804, 115.04~
## $ Astro_WT_GFP_5    <dbl> 0.000000, 38.062370, 485.295213, 721.454915, 36.3322~
## $ Astro_WT_GFP_2    <dbl> 0.000000, 97.657908, 111.284593, 387.224961, 0.00000~
## $ Other_WT_GFP_3    <dbl> 314.655409, 0.000000, 815.806018, 700.903762, 7.9547~
## $ Astro_APP_SOCS_1  <dbl> 0.000000, 44.160860, 375.367311, 572.987159, 132.482~
## $ Astro_APP_GFP_1   <dbl> 1.810387, 47.070063, 599.238108, 568.461529, 75.1310~
```

Format data:

```
names(dat)
```

```
##  [1] "FastDB Stale ID"  "coordinates"      "symbol"           "Astro_APP_GFP_2"
##  [5] "Astro_APP_GFP_4"  "Astro_WT_GFP_6"   "Astro_APP_GFP_3"  "Other_WT_GFP_1"
##  [9] "Astro_APP_SOCS_5" "Astro_APP_SOCS_2" "Other_WT_GFP_2"   "Astro_APP_SOCS_3"
## [13] "Astro_WT_GFP_4"   "Astro_WT_GFP_1"   "Astro_APP_SOCS_4" "Astro_WT_GFP_3"
## [17] "Astro_WT_GFP_7"   "Astro_WT_GFP_5"   "Astro_WT_GFP_2"   "Other_WT_GFP_3"
## [21] "Astro_APP_SOCS_1" "Astro_APP_GFP_1"
```

```
edat_raw <- dat %>% select(-coordinates, -symbol)
edat_raw <- edat_raw %>% column_to_rownames(var='FastDB Stale ID')
edat_raw <- edat_raw[,sort(names(edat_raw))]

dim(edat_raw)
```

```
## [1] 60567    19
## gene names in rows
## samples in columns
edat_raw[1:5,1:4]
```

```
##             Astro_APP_GFP_1 Astro_APP_GFP_2 Astro_APP_GFP_3 Astro_APP_GFP_4
## GSMG0000003        1.810387          0.0000         0.00000        48.82075
## GSMG0000004       47.070063          0.0000        51.44161        66.25673
## GSMG0000005      599.238108        206.8224       294.80002       442.87391
## GSMG0000006      568.461529        728.6220       479.79198       652.97748
## GSMG0000007       75.131062          0.0000        48.47383        25.28217
```

```
summary(edat_raw[,1:4])
```

```
##  Astro_APP_GFP_1    Astro_APP_GFP_2     Astro_APP_GFP_3     Astro_APP_GFP_4
##  Min.   :     0.0   Min.   :      0.0   Min.   :      0.0   Min.   :     0.0
##  1st Qu.:     0.0   1st Qu.:      0.0   1st Qu.:      0.0   1st Qu.:     0.0
##  Median :     0.0   Median :      0.0   Median :      0.0   Median :     0.0
##  Mean   :   385.1   Mean   :    402.6   Mean   :    406.5   Mean   :   384.7
##  3rd Qu.:    41.6   3rd Qu.:      6.6   3rd Qu.:     45.5   3rd Qu.:    47.9
##  Max.   :843242.1   Max.   :1280773.3   Max.   :1384105.9   Max.   :941960.6
```

Preparing datastes, metadata with four groups:

```
## we colud parse samples metadata just from samples name, but let's do it form GEO metadata
pheno %>% select(title, description)
```

```
##                       title                         description
## GSM2902723   Astro-WT-GFP-1    replicate 1-astrocyte-WT-GFP-group
## GSM2902724   Astro-WT-GFP-2    replicate 2-astrocyte-WT-GFP-group
## GSM2902725   Astro-WT-GFP-3    replicate 3-astrocyte-WT-GFP-group
## GSM2902726   Astro-WT-GFP-4    replicate 4-astrocyte-WT-GFP-group
## GSM2902727   Astro-WT-GFP-5    replicate 5-astrocyte-WT-GFP-group
## GSM2902728   Astro-WT-GFP-6    replicate 6-astrocyte-WT-GFP-group
## GSM2902729   Astro-WT-GFP-7    replicate 7-astrocyte-WT-GFP-group
## GSM2902730  Astro-APP-GFP-1   replicate 1-astrocyte-APP-GFP-group
## GSM2902731  Astro-APP-GFP-2   replicate 2-astrocyte-APP-GFP-group
## GSM2902732  Astro-APP-GFP-3   replicate 3-astrocyte-APP-GFP-group
## GSM2902733  Astro-APP-GFP-4   replicate 4-astrocyte-APP-GFP-group
## GSM2902734 Astro-APP-SOCS-1 replicate 1-astrocyte-APP-SOCS3-group
## GSM2902735 Astro-APP-SOCS-2 replicate 2-astrocyte-APP-SOCS3-group
## GSM2902736 Astro-APP-SOCS-3 replicate 3-astrocyte-APP-SOCS3-group
## GSM2902737 Astro-APP-SOCS-4 replicate 4-astrocyte-APP-SOCS3-group
## GSM2902738 Astro-APP-SOCS-5 replicate 5-astrocyte-APP-SOCS3-group
## GSM2902739   Other-WT-GFP-1        replicate 1-other-WT-GFP-group
## GSM2902740   Other-WT-GFP-2        replicate 2-other-WT-GFP-group
## GSM2902741   Other-WT-GFP-3        replicate 3-other-WT-GFP-group
```

```
pdf4       <- pheno %>% select(title)
pdf4$title <- pdf4$title %>%  str_replace_all("-","_")

pdf4$group <- str_split(pdf4$title, "_\\d", simplify=T)[,1] %>%
  str_replace_all("Astro_","a") %>%
  str_replace_all("Other_","o")

pdf4$group <- as.factor(pdf4$group)
```

```
names(pdf4) <- c("sname", "group")
rownames(pdf4) <- pdf4$sname
pdf4 <- arrange(pdf4, sname)

pdf4 %>% dplyr::count(group)
```

```
##        group n
## 1  aAPP_GFP 4
## 2 aAPP_SOCS 5
## 3   aWT_GFP 7
## 4   oWT_GFP 3
```

```
pdf4
```

```
##                             sname     group
## Astro_APP_GFP_1   Astro_APP_GFP_1  aAPP_GFP
## Astro_APP_GFP_2   Astro_APP_GFP_2  aAPP_GFP
## Astro_APP_GFP_3   Astro_APP_GFP_3  aAPP_GFP
## Astro_APP_GFP_4   Astro_APP_GFP_4  aAPP_GFP
## Astro_APP_SOCS_1 Astro_APP_SOCS_1 aAPP_SOCS
## Astro_APP_SOCS_2 Astro_APP_SOCS_2 aAPP_SOCS
## Astro_APP_SOCS_3 Astro_APP_SOCS_3 aAPP_SOCS
## Astro_APP_SOCS_4 Astro_APP_SOCS_4 aAPP_SOCS
## Astro_APP_SOCS_5 Astro_APP_SOCS_5 aAPP_SOCS
## Astro_WT_GFP_1     Astro_WT_GFP_1   aWT_GFP
## Astro_WT_GFP_2     Astro_WT_GFP_2   aWT_GFP
## Astro_WT_GFP_3     Astro_WT_GFP_3   aWT_GFP
## Astro_WT_GFP_4     Astro_WT_GFP_4   aWT_GFP
## Astro_WT_GFP_5     Astro_WT_GFP_5   aWT_GFP
## Astro_WT_GFP_6     Astro_WT_GFP_6   aWT_GFP
## Astro_WT_GFP_7     Astro_WT_GFP_7   aWT_GFP
## Other_WT_GFP_1     Other_WT_GFP_1   oWT_GFP
## Other_WT_GFP_2     Other_WT_GFP_2   oWT_GFP
## Other_WT_GFP_3     Other_WT_GFP_3   oWT_GFP
```

Preparing metadata with three groups; we remove “other” types as they are not astrocytes, so thier experssion obvioulsy will be different:

```
pdf3<- pdf4 %>% filter(!group=='oWT_GFP')
pdf3 <- droplevels(pdf3)
pdf3 %>% dplyr::count(group)
```

```
##        group n
## 1  aAPP_GFP 4
## 2 aAPP_SOCS 5
## 3   aWT_GFP 7
```

Filtering expression counts and log transformation

```
## remove low expressed data
edat10  <- edat_raw[rowMeans(edat_raw) > 10, ] %>% arrange(rownames(.))
## wa want fold changes, but we can't do log2(0) so we add 1
edatlog4 <- log2(as.matrix(edat10) + 1) %>% as.data.frame()
```

```
## let's create only data set for three groups
edatraw3 <- edat_raw %>%  select(-Other_WT_GFP_1, -Other_WT_GFP_2, -Other_WT_GFP_3) %>% arrange(rownames
edatraw3 <- edatraw3[rowMeans(edatraw3) > 10, ]
edatlog3 <- log2(as.matrix(edatraw3) + 1) %>% as.data.frame()

summary(edatlog3[,1:4])
```

```
##  Astro_APP_GFP_1  Astro_APP_GFP_2  Astro_APP_GFP_3  Astro_APP_GFP_4
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 4.990   1st Qu.: 0.000   1st Qu.: 5.072   1st Qu.: 5.199
##  Median : 7.057   Median : 7.074   Median : 7.083   Median : 7.246
##  Mean   : 6.972   Mean   : 5.832   Mean   : 6.989   Mean   : 7.088
##  3rd Qu.: 9.270   3rd Qu.: 9.371   3rd Qu.: 9.256   3rd Qu.: 9.334
##  Max.   :19.686   Max.   :20.289   Max.   :20.401   Max.   :19.845
```

Let's make sure the names are aligned

```
all.equal(colnames(edat_raw), pdf4$sname)
```

```
## [1] TRUE
```

```
all.equal(colnames(edatlog4), pdf4$sname)
```

```
## [1] TRUE
```

```
all.equal(colnames(edatraw3), pdf3$sname)
```

```
## [1] TRUE
```

```
all.equal(colnames(edatlog3), pdf3$sname)
```

```
## [1] TRUE
```

```
dim(edat_raw)
```

```
## [1] 60567    19
```

```
dim(edatlog4)
```

```
## [1] 23044    19
```

```
dim(edatraw3)
```

```
## [1] 21543    16
```

```
dim(edatlog3)
```

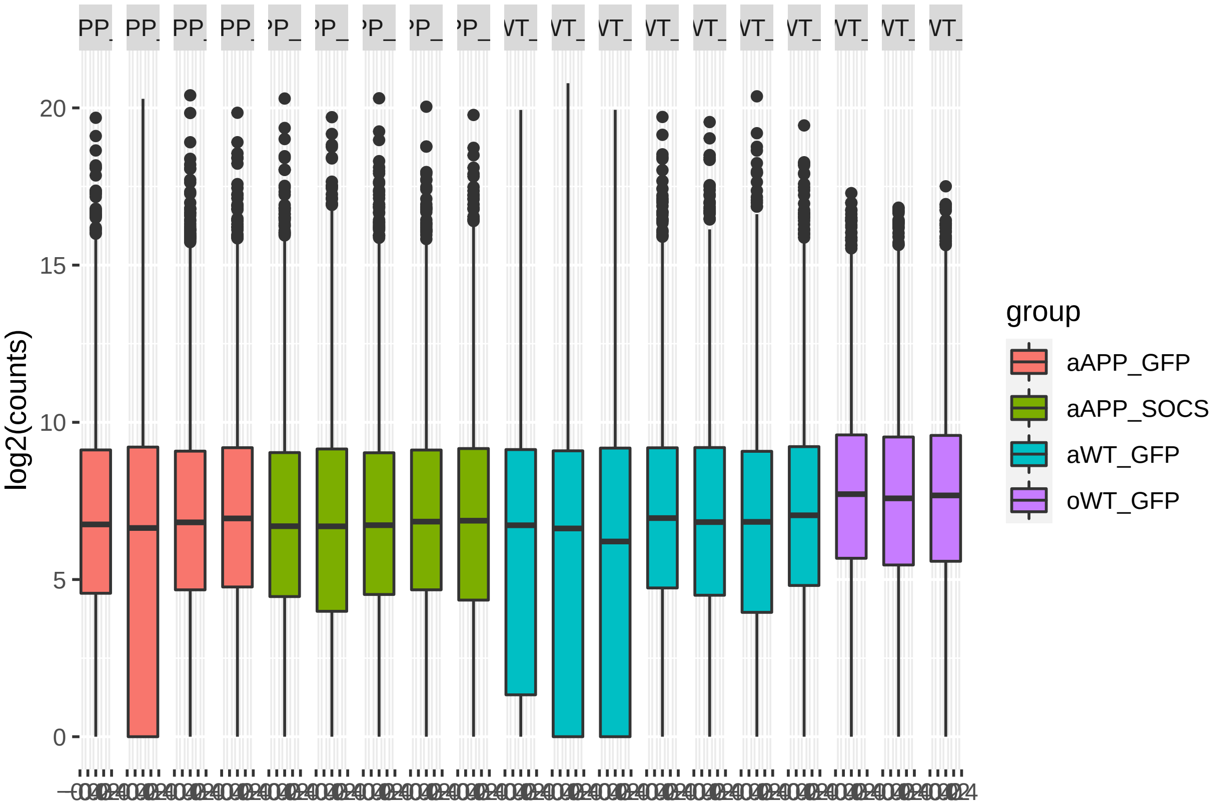
```
## [1] 21543    16
```

Let's make some data in tidy form:

```
etidy <- gather(edatlog4, key="sname", value="expr") %>% arrange(sname)
etidy <- left_join(etidy, pdf4)
```

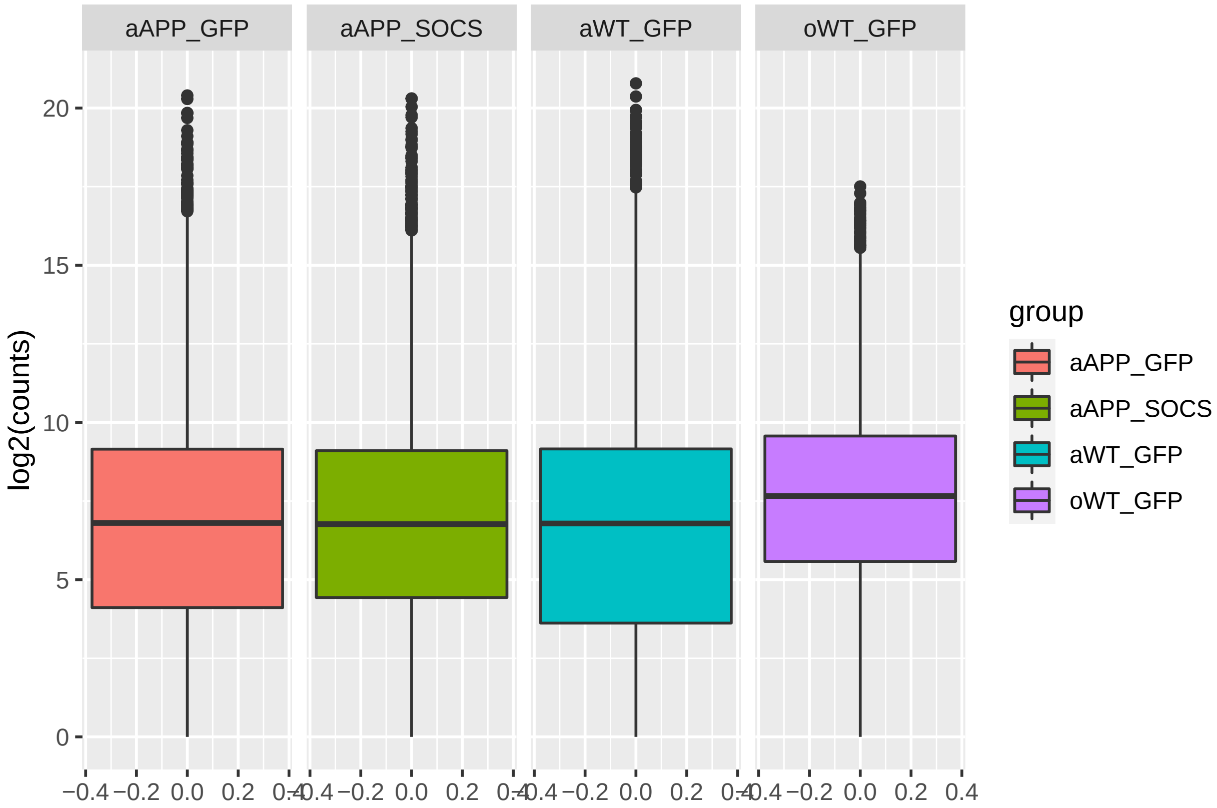
Let's see data summary:

```
ggplot(etidy, aes(x=0,y=expr, fill=group)) +
  geom_boxplot()  +
  facet_grid(~sname) +
  ylab("log2(counts)") +
  xlab("")
```



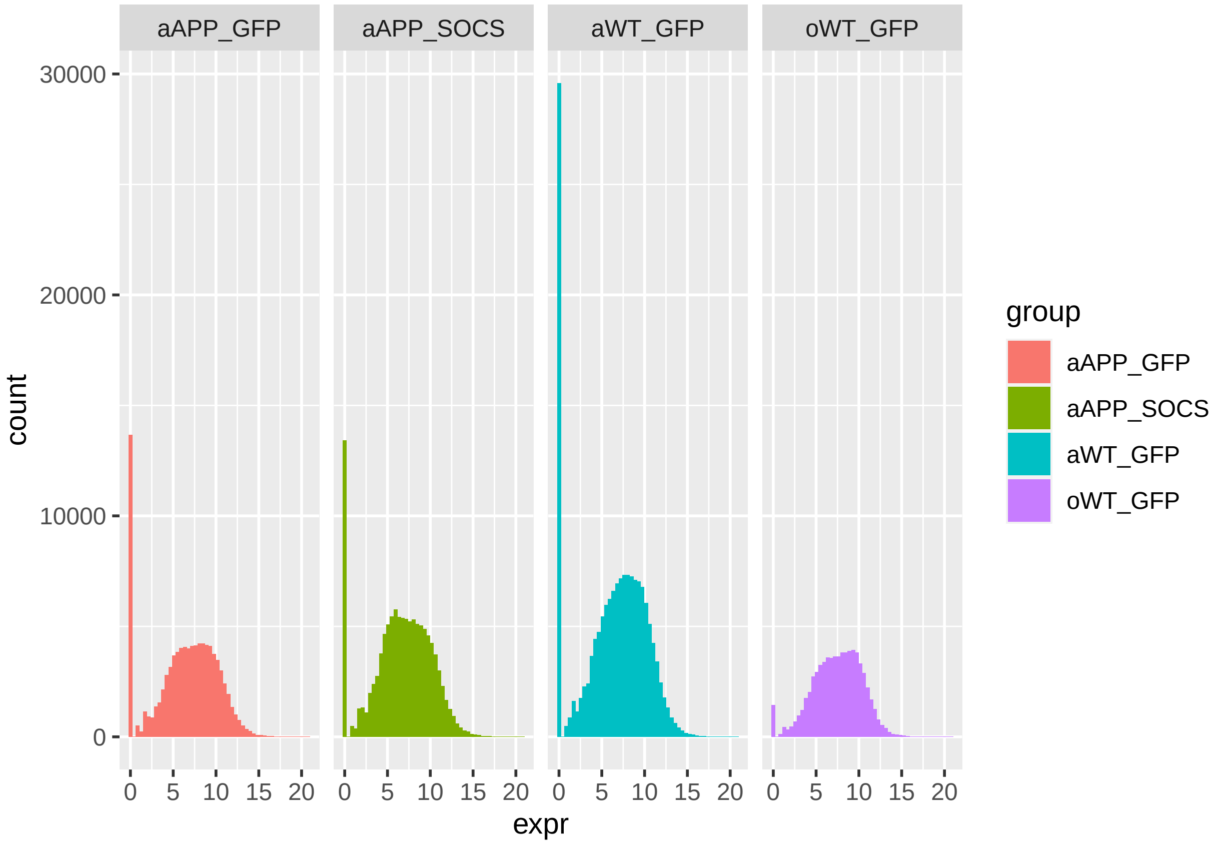
Summary per group:

```
ggplot(etidy, aes(x=0,y=expr, fill=group)) +
  geom_boxplot()  +
  facet_grid(~group) +
  ylab("log2(counts)") +
  xlab("")
```



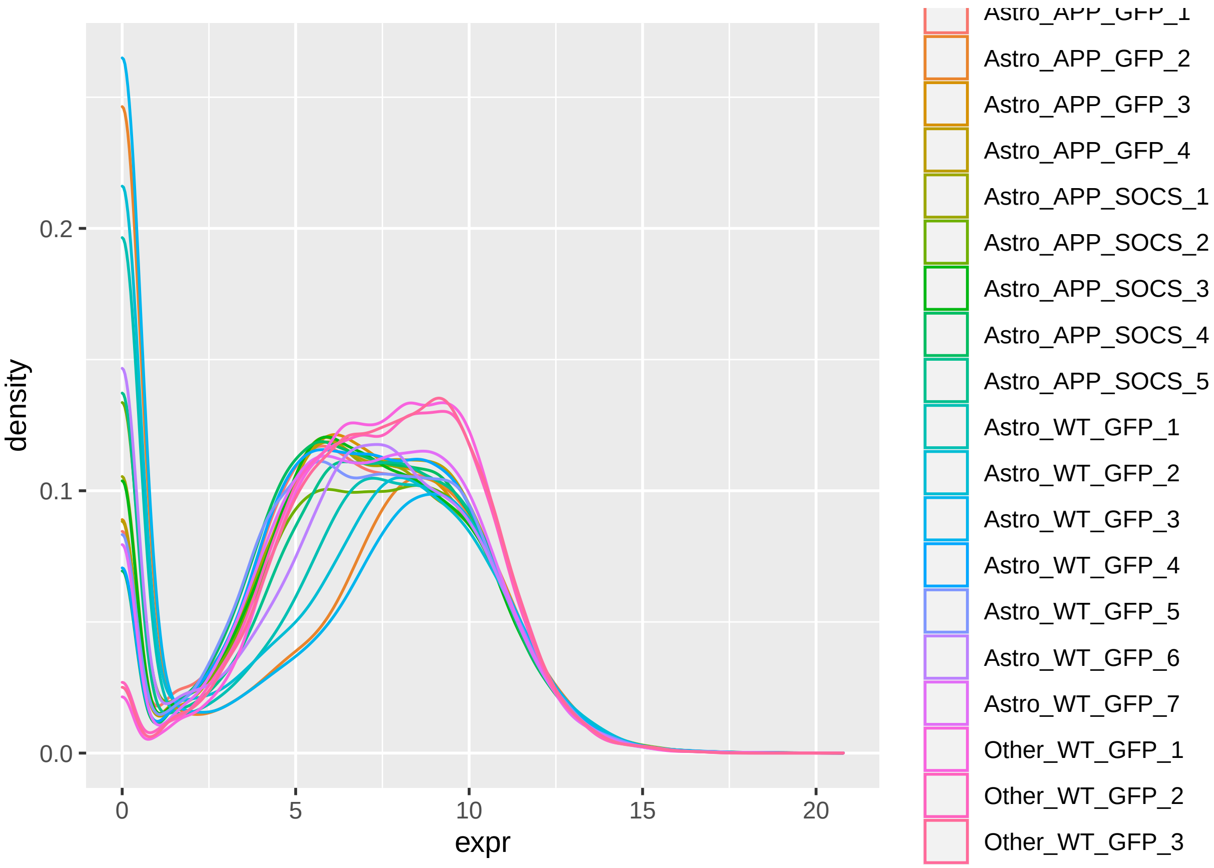
Histograms per group:

```
ggplot(etidy, aes(x=expr, fill=group)) +
  geom_histogram(bins="50")  +
  facet_grid(~group)
```

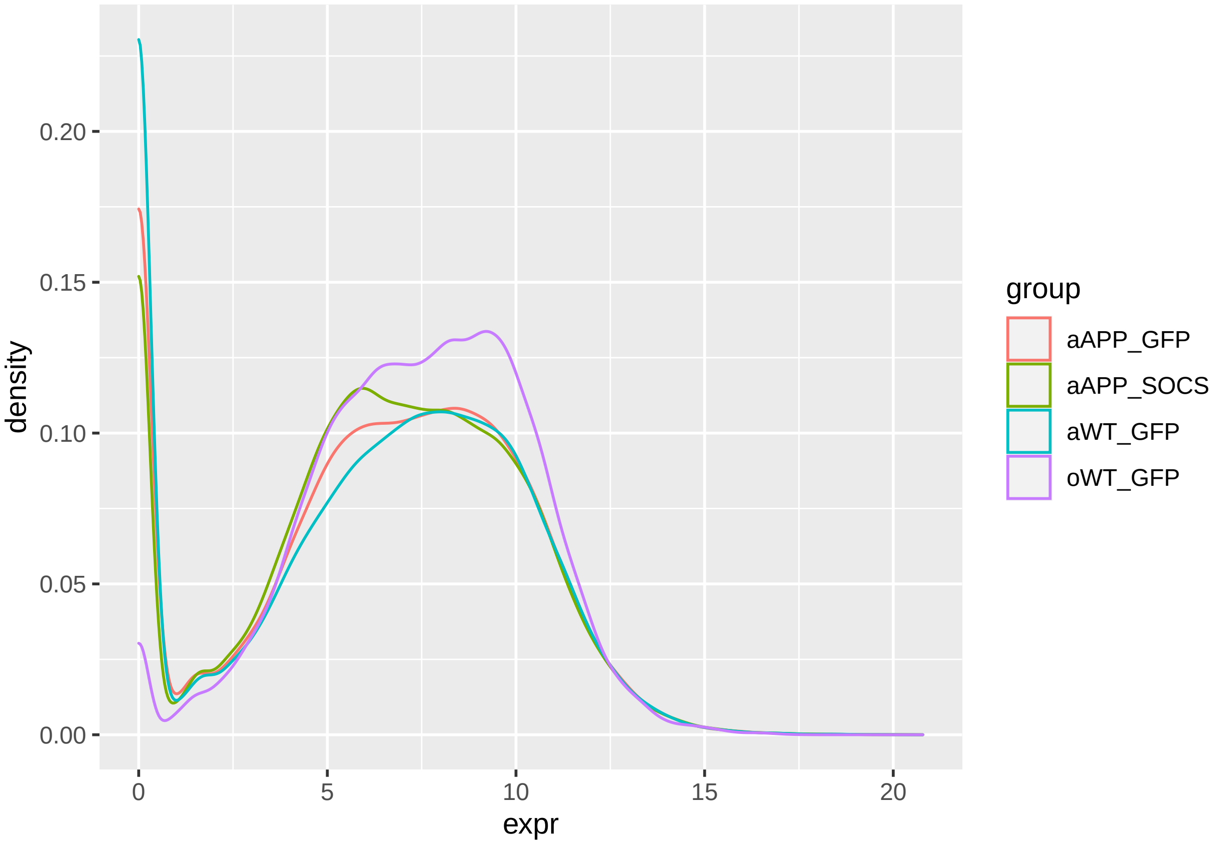


Density plots:

```
ggplot(etidy, aes(x=expr, colour=sname)) +
  geom_density()
```

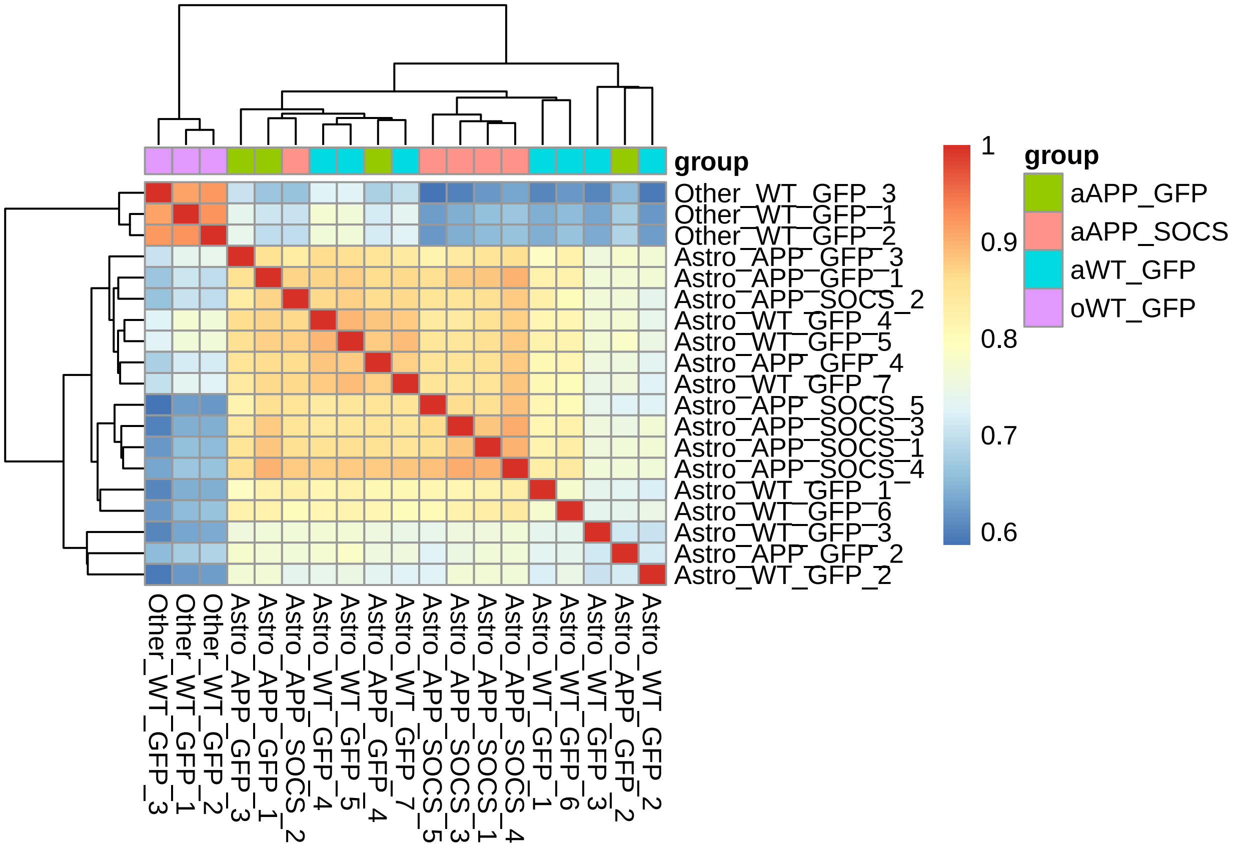


```
ggplot(etidy, aes(x=expr, colour=group)) +
  geom_density()
```



Heatmap:

```
library(pheatmap)

corMatrix <- cor(edatlog4)
pheatmap(corMatrix, annotation_col = select(pdf4, -sname))
```

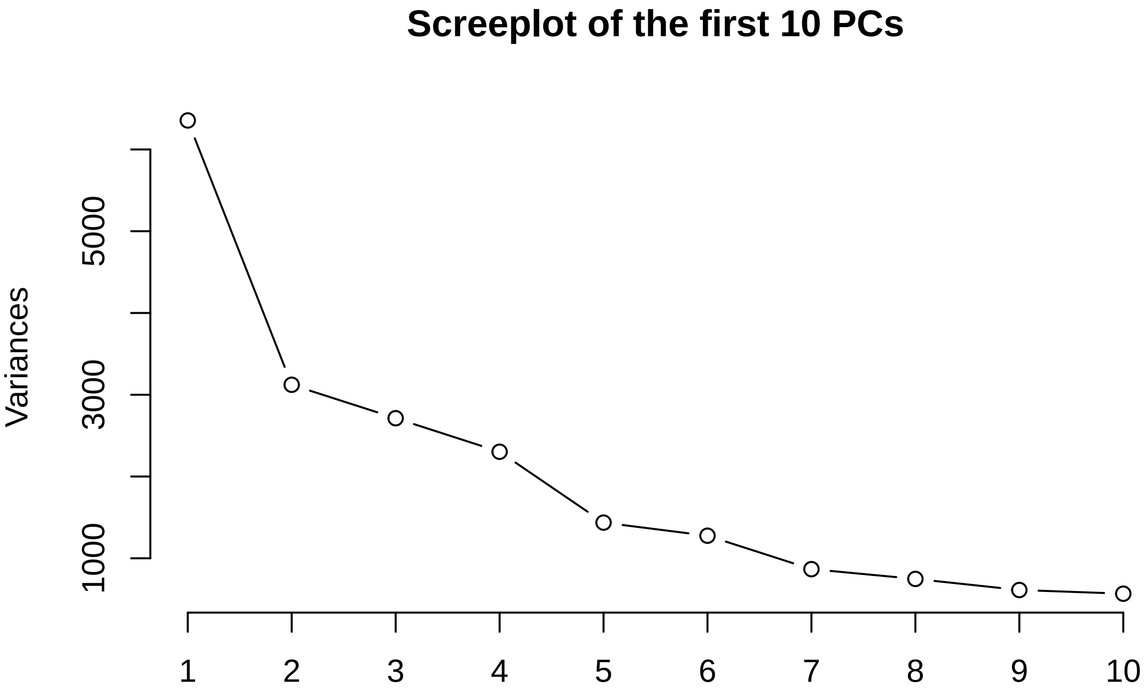


PCA:

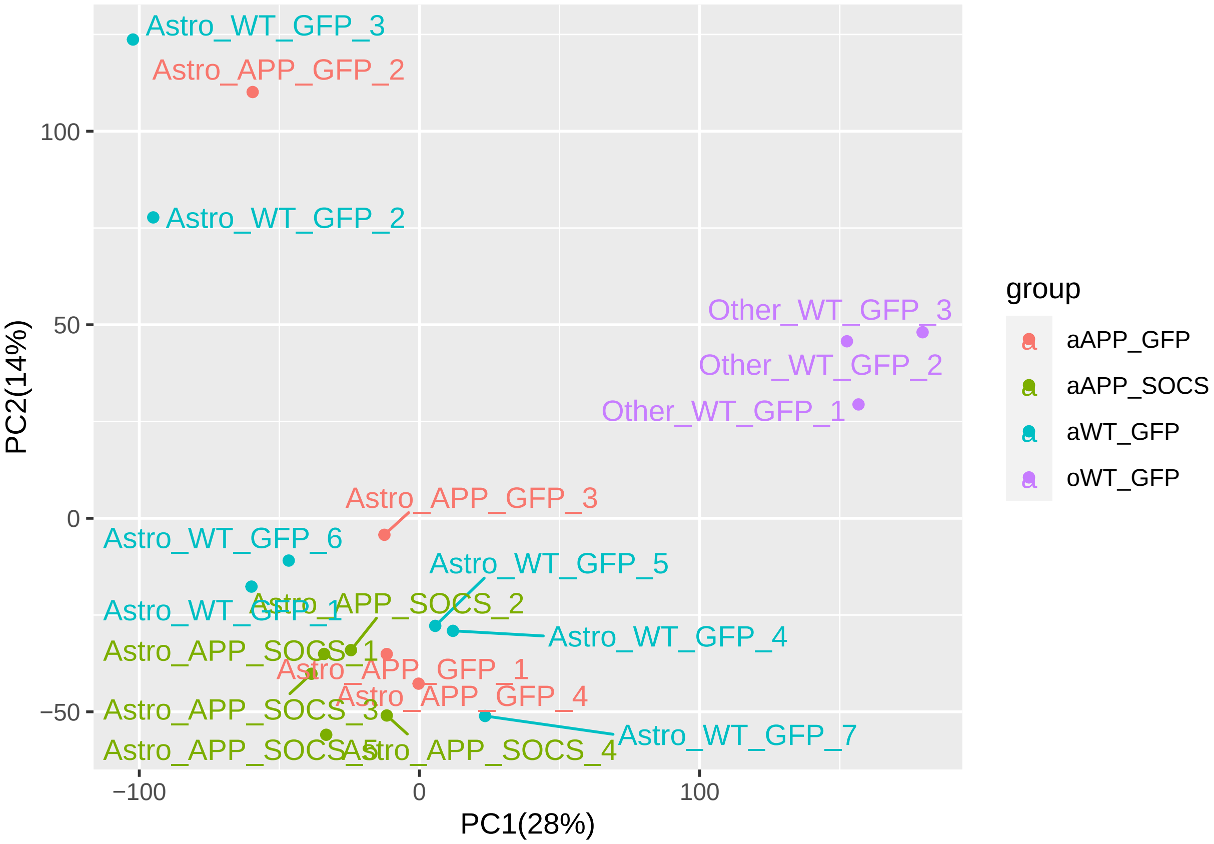
```r
library(ggplot2)
library(ggrepel)
library(stats)

expr_pca<-prcomp(t(edatlog4), center = TRUE, scale. = TRUE)
summary(expr_pca)
```

```
## Importance of components:
##                            PC1     PC2     PC3      PC4      PC5      PC6
## Standard deviation     79.7153 55.8740 52.0867 47.98782 37.90073 35.72330
## Proportion of Variance  0.2758  0.1355  0.1177  0.09993  0.06234  0.05538
## Cumulative Proportion   0.2758  0.4112  0.5290  0.62890  0.69123  0.74661
##                             PC7      PC8      PC9     PC10     PC11     PC12
## Standard deviation     29.44954 27.34419 24.73827 23.81193 22.13448 21.77910
## Proportion of Variance  0.03764  0.03245  0.02656  0.02461  0.02126  0.02058
## Cumulative Proportion   0.78425  0.81669  0.84325  0.86786  0.88912  0.90970
##                            PC13     PC14     PC15     PC16     PC17     PC18
## Standard deviation     20.90576 20.50394 19.17287 17.95464 16.91160 15.72980
## Proportion of Variance  0.01897  0.01824  0.01595  0.01399  0.01241  0.01074
## Cumulative Proportion   0.92867  0.94691  0.96286  0.97685  0.98926  1.00000
##                             PC19
## Standard deviation     2.061e-13
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

```
screeplot(expr_pca, type = "l", npcs = 10, main = "Screeplot of the first 10 PCs")
```



```
imp <- summary(expr_pca)$importance
pc1 <- round(imp["Proportion of Variance","PC1"] *100, digits=0)
pc2 <- round(imp["Proportion of Variance","PC2"] *100, digits=0)

cbind(pdf4, expr_pca$x) %>%
 ggplot(aes(x = PC1, y=PC2, col=group, label=rownames(pdf4) ) ) +
 geom_point() +
 geom_text_repel() +
 ylab(paste0("PC2(",pc2,"%)")) +
 xlab(paste0("PC1(",pc1,"%)"))
```



We see, that other group has a bit different density profiles and cluster together. It is expected, as these are different cell types. We exclude them from differential expression analysis

Differential Expression for many groups

Many sources recommend to use linear models to find relations in RNA-seq count data, however in such scenario the data should be normally distributed (or at least the LM's residuals should). In the paper they use ANOVA or Kruskal-Wallis tests, depending on assumptions fulfillment.

Here I try to use Generalized Linear Model, as RNA-Seq use to be not normal. Firstly let's test a normality of some random sample (if just one sample is not normally distributed, we can't use parametric testes or linear models)

```
shapiro.test(sample(edatraw3$Astro_APP_GFP_2, 5000))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(edatraw3$Astro_APP_GFP_2, 5000)
## W = 0.098952, p-value < 2.2e-16
```

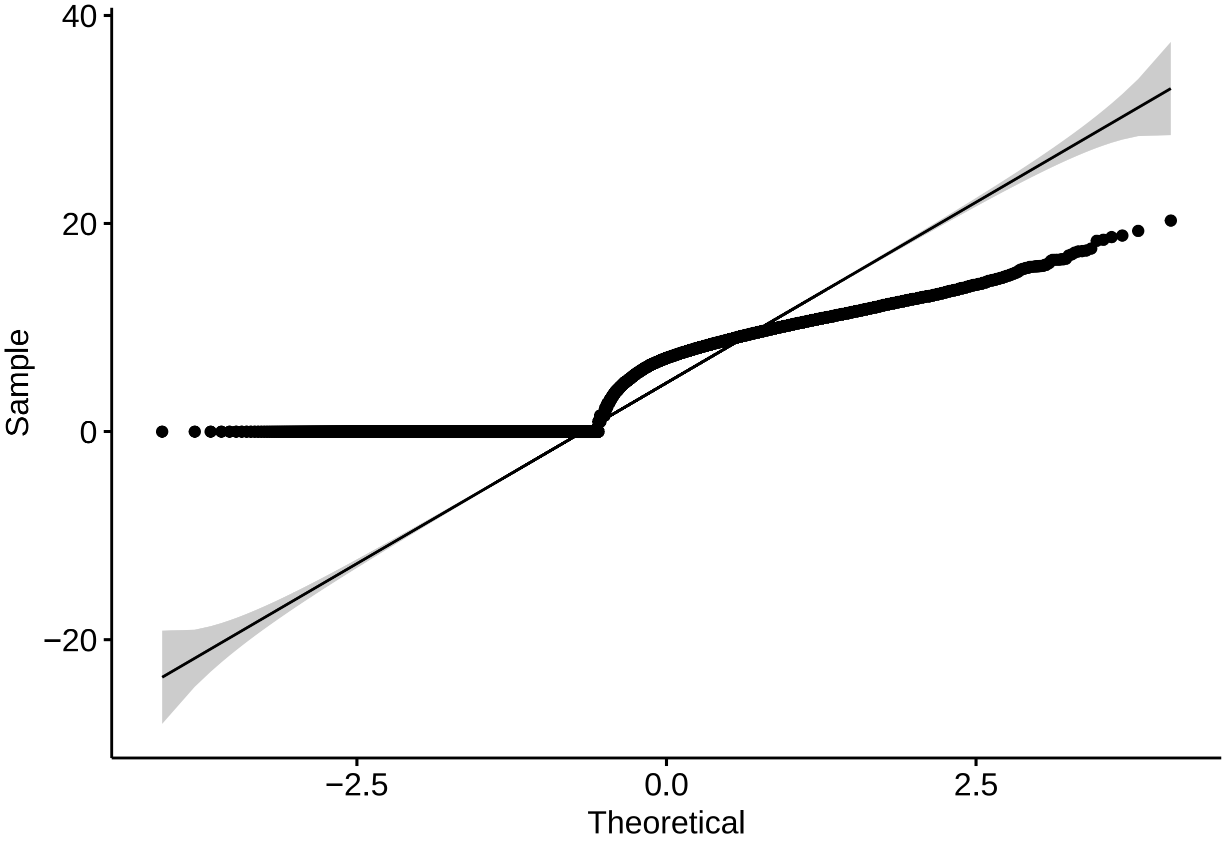
```
shapiro.test(sample(edatlog3$Astro_APP_GFP_2, 5000))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(edatlog3$Astro_APP_GFP_2, 5000)
## W = 0.87924, p-value < 2.2e-16
```

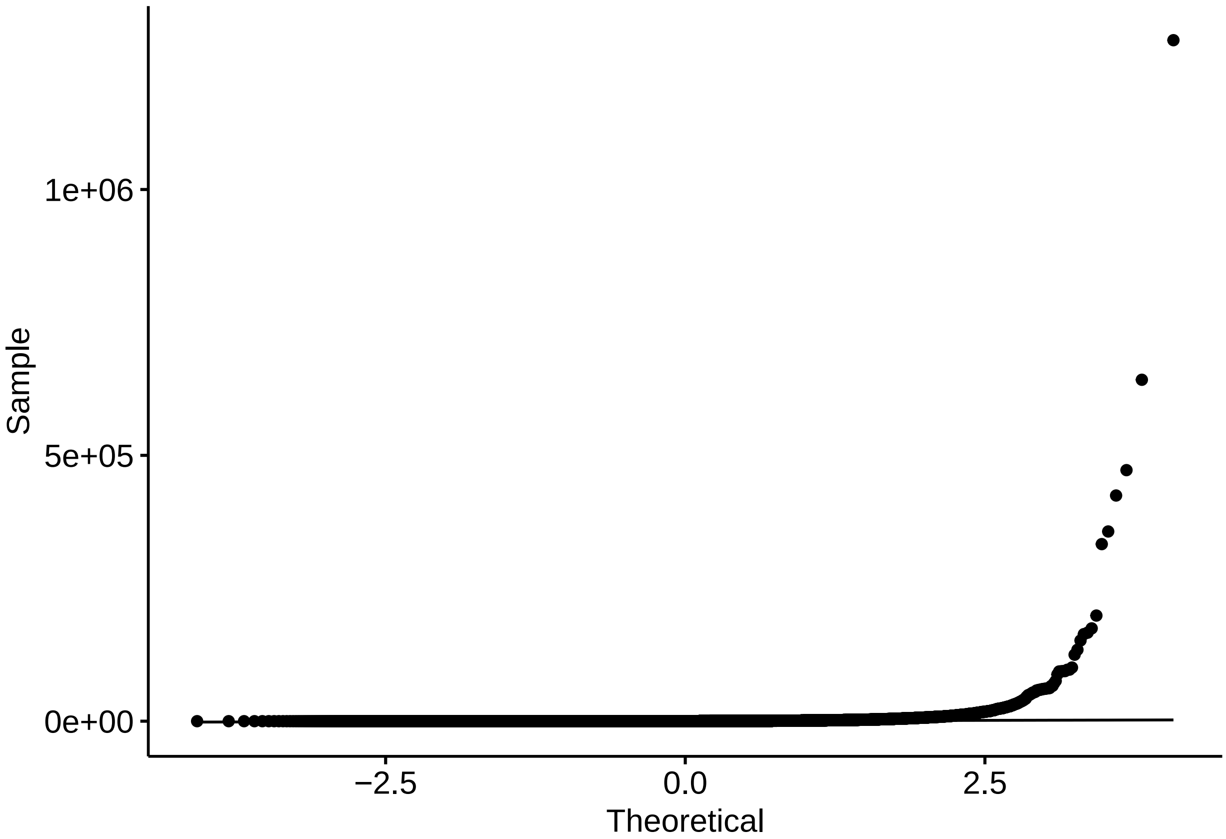
Shapiro-Wilk's p-value is less than 0.01 in both datasets (raw counts and log ratios), so should use non-parametric approaches

Let's check a qq plot:

```
library(ggpubr)

ggqqplot(data=edatlog3, x="Astro_APP_GFP_2")
```



```
ggqqplot(data=edatraw3, x="Astro_APP_GFP_2")
```



Both plots are concordant with the results of SW test.

We can do the same test for some random genes (across all samples)

```
test1 <- as.numeric(edatlog3['GSMG0032532',])
test2 <- as.numeric(edatlog3['GSMG0026079',])
shapiro.test(test1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  test1
## W = 0.46987, p-value = 1.206e-06
```

```
shapiro.test(test2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  test2
## W = 0.78258, p-value = 0.001614
```

GLM: Crate a description and contrast matrix: we are interested in differences in any group pairs

```
suppressPackageStartupMessages({library(edgeR);library(limma)})

des_mat <- model.matrix(~ group + 0, data = pdf3)
colnames(des_mat) <- stringr::str_remove(colnames(des_mat), "group")

print(des_mat)
```

```
##                  aAPP_GFP aAPP_SOCS aWT_GFP
## Astro_APP_GFP_1         1         0       0
## Astro_APP_GFP_2         1         0       0
## Astro_APP_GFP_3         1         0       0
## Astro_APP_GFP_4         1         0       0
## Astro_APP_SOCS_1        0         1       0
## Astro_APP_SOCS_2        0         1       0
## Astro_APP_SOCS_3        0         1       0
## Astro_APP_SOCS_4        0         1       0
## Astro_APP_SOCS_5        0         1       0
## Astro_WT_GFP_1          0         0       1
## Astro_WT_GFP_2          0         0       1
## Astro_WT_GFP_3          0         0       1
## Astro_WT_GFP_4          0         0       1
## Astro_WT_GFP_5          0         0       1
## Astro_WT_GFP_6          0         0       1
## Astro_WT_GFP_7          0         0       1
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
```

```
contrast_matrix <- makeContrasts(
  "aAPP_GFPvs" = aAPP_GFP  - aWT_GFP ,
  "aAPP_SOCSvs"= aAPP_SOCS - aWT_GFP ,
  "SOCS_GFAP"  = aAPP_SOCS - aAPP_GFP ,
  levels = des_mat
)

print(contrast_matrix)
```

```
##            Contrasts
## Levels      aAPP_GFPvs aAPP_SOCSvs SOCS_GFAP
##   aAPP_GFP           1           0        -1
##   aAPP_SOCS          0           1         1
##   aWT_GFP           -1          -1         0
```

Let's be sure again that the names in data and matadata are aligned

```
all.equal(colnames(edatraw3), pdf3$sname)
```

```
## [1] TRUE
```

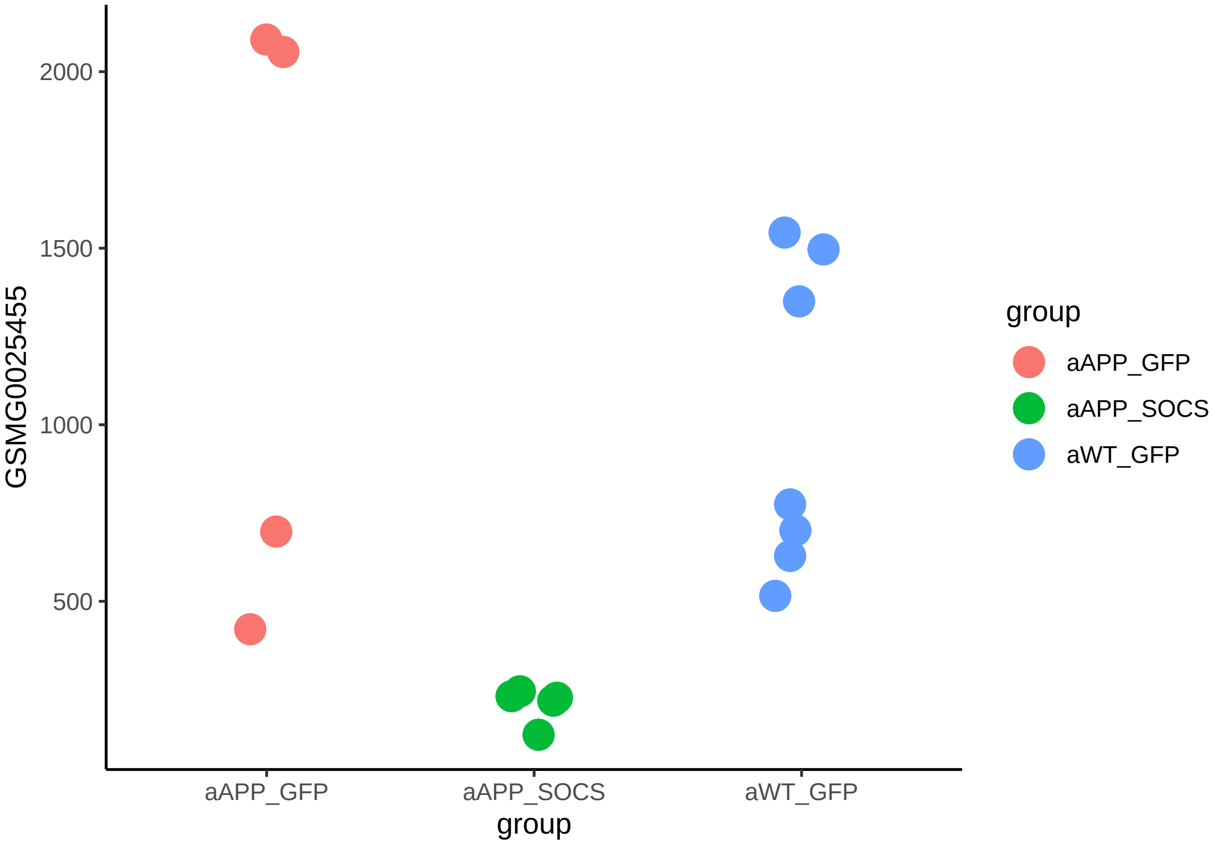
For GLM we use raw counts data:

```r
# https://bioinformatics-core-shared-training.github.io/RNAseq-R/rna-seq-de.nb.html
# https://rpubs.com/bman/79395

# dge <- DGEList( counts=edatlog3, group=pdf3$group, lib.size=colSums( edatlog3 ) )
dge <- DGEList( counts=edatraw3, group=pdf3$group, lib.size=colSums( edatraw3 ) )
dge <- calcNormFactors( dge )

dge <- estimateGLMCommonDisp( dge, des_mat )
dge <- estimateGLMTrendedDisp( dge, des_mat )
dge <- estimateGLMTagwiseDisp( dge, des_mat )

fit <- glmFit( dge, des_mat )

glms <- glmLRT( fit, contrast=contrast_matrix )

diffs <-topTags( glms, n = nrow(glms)) %>% as.data.frame() %>% filter(FDR<0.01) %>% rownames_to_column(


gene_map <- dat %>% select(`FastDB Stale ID`, symbol)
colnames(gene_map) <- c("genes", "symbol")

diffs <- select(diffs, genes, FDR)

left_join(diffs,gene_map) %>% select(genes, symbol, FDR)
```

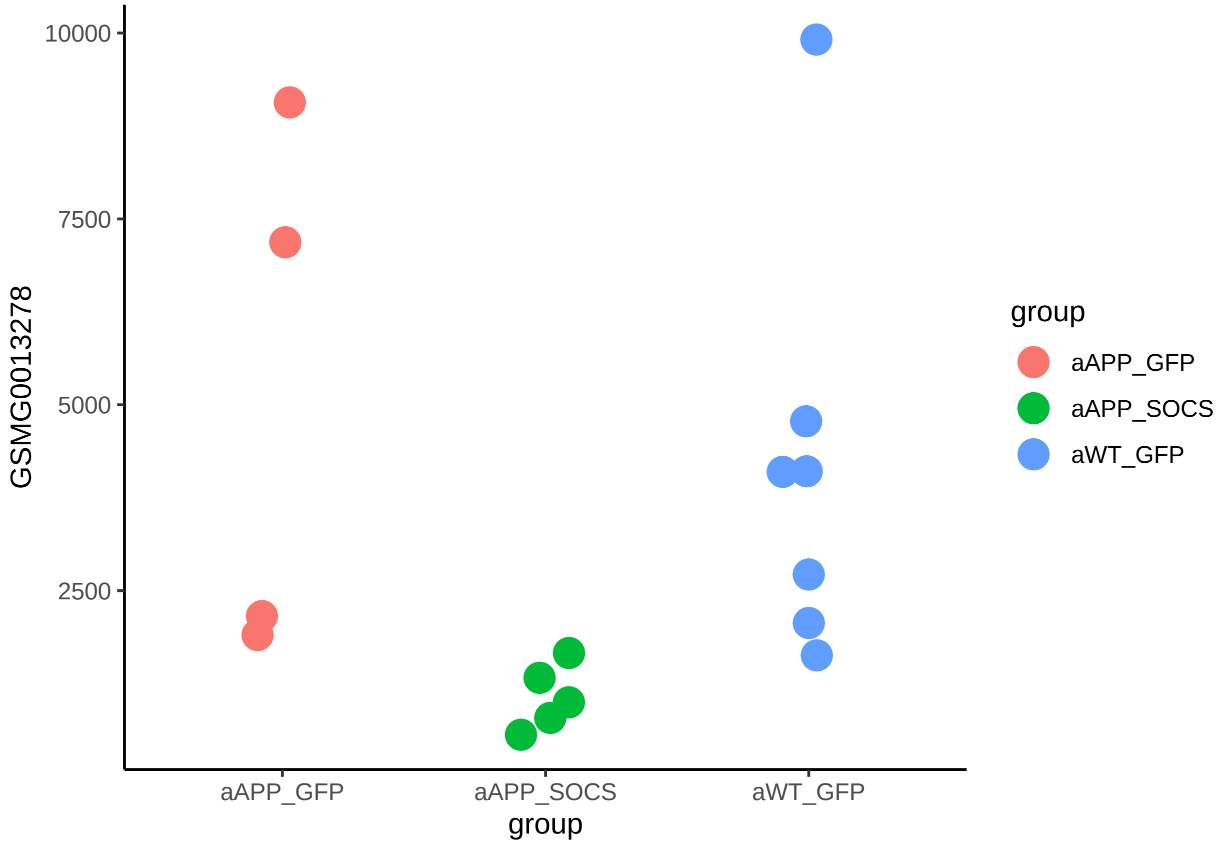
```
##          genes            symbol          FDR
## 1  GSMG0007690             Socs3 4.501947e-52
## 2  GSMG0021942              Cst7 2.783989e-18
## 3  GSMG0016568              NULL 5.861383e-06
## 4  GSMG0017434               C4b 2.183675e-05
## 5  GSMG0031379              Flt1 4.586550e-05
## 6  GSMG0025455            P2ry13 1.267427e-04
## 7  GSMG0017445 Hspa1a // Hspa1b 2.066839e-04
## 8  GSMG0024658              Ctss 2.358180e-04
## 9  GSMG0009764             S1pr3 4.224505e-04
## 10 GSMG0005598               Vtn 8.096517e-04
## 11 GSMG0016569              NULL 1.002883e-03
## 12 GSMG0018220              Egr1 1.315221e-03
## 13 GSMG0054270              NULL 1.796204e-03
## 14 GSMG0015646              Apod 1.903556e-03
## 15 GSMG0011527     Ang // Rnase4 1.903556e-03
## 16 GSMG0005688              Car4 2.275087e-03
## 17 GSMG0005310              Grap 2.291143e-03
## 18 GSMG0033846            Slco1a4 2.501076e-03
## 19 GSMG0042796             Itm2a 2.716639e-03
## 20 GSMG0020929               Eng 2.716639e-03
## 21 GSMG0013660            Acvrl1 3.372366e-03
## 22 GSMG0009699              Cd83 3.770716e-03
## 23 GSMG0013663             Nr4a1 3.770716e-03
## 24 GSMG0028527              C1qc 3.924539e-03
## 25 GSMG0035329            Cyp2e1 4.071091e-03
## 26 GSMG0034882              Ucp2 4.157088e-03
## 27 GSMG0025433            Tm4sf1 4.157088e-03
## 28 GSMG0013278              Ly6e 4.166640e-03
## 29 GSMG0017120             Mas1 4.223254e-03
## 30 GSMG0016770            Trem2 4.478486e-03
## 31 GSMG0043213           Plac9a 4.491891e-03
## 32 GSMG0042792          Cysltr1 5.101854e-03
## 33 GSMG0030942           Selplg 5.186481e-03
## 34 GSMG0014090             NULL 5.186481e-03
## 35 GSMG0002111             Btg2 5.818395e-03
## 36 GSMG0007122             Ccl6 5.833124e-03
## 37 GSMG0030665           Igfbp7 6.788921e-03
## 38 GSMG0001070           Tagln2 7.431580e-03
## 39 GSMG0028528             C1qa 7.971180e-03
## 40 GSMG0019362          Slc22a8 8.244338e-03
## 41 GSMG0028526             C1qb 8.477187e-03
```

We discovered 41 significant results. The next step could be GO and KEGG anntations, functional analysis etc. but this is out of the scope of this project. At the end let's visualize some randomly selected results (gene expressions)

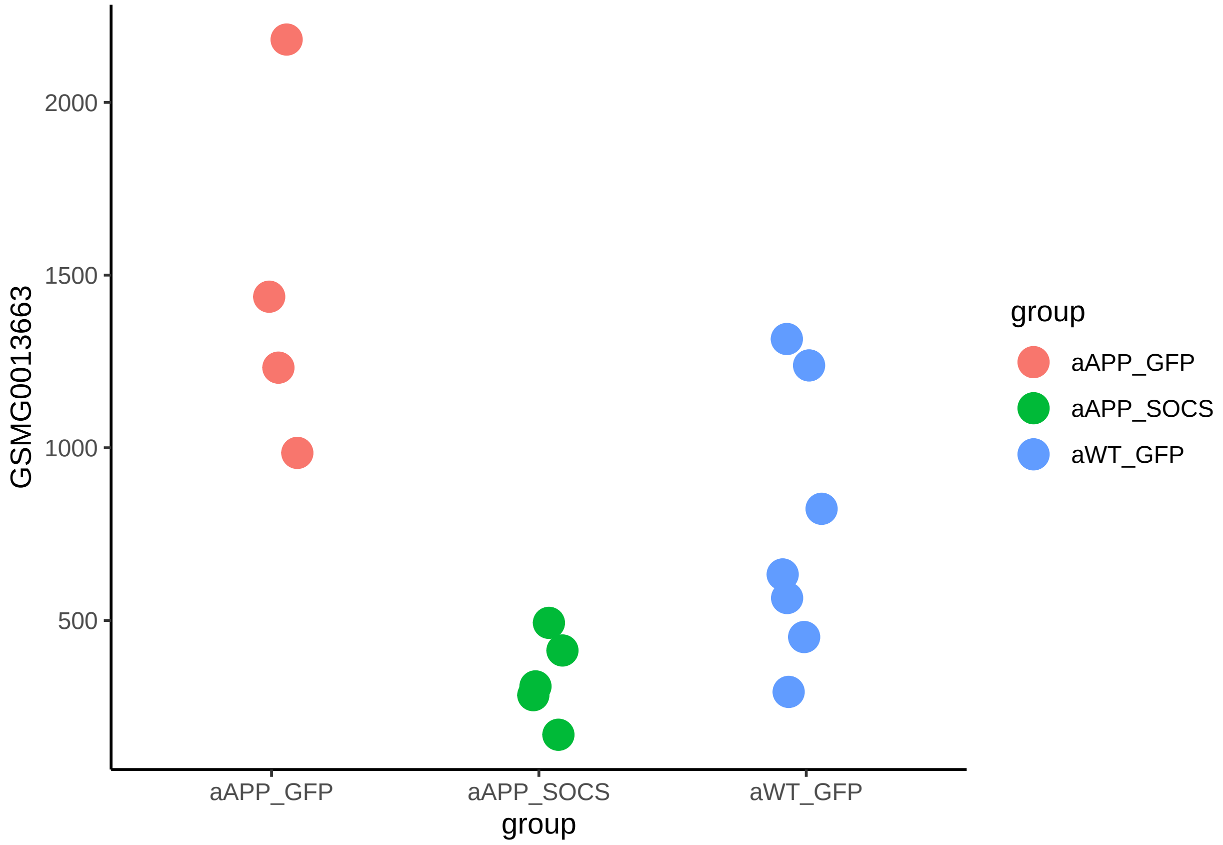
```r
plot_gene_expr <- function(gene_id) {

  top_gene_df <- edatraw3 %>%
    # Extract this gene from `expression_df`
    dplyr::filter(rownames(.) == gene_id) %>% as.matrix() %>%
    # Transpose so the gene is a column
    t() %>%
    # Transpose made this a matrix, let's make it back into a data.frame like before
    data.frame() %>%
    # Store the sample ids as their own column instead of being row names
    tibble::rownames_to_column("sname") %>%
    # # Join on the selected columns from metadata
    dplyr::inner_join(dplyr::select(
      pdf3,
      sname,
      group
    ))

  ggplot(top_gene_df, aes_string(x = "group", y = gene_id, color = "group")) +
    geom_jitter(width = 0.1, height = 0, size=5) +
    theme_classic()
}


plot_gene_expr("GSMG0025455")
```



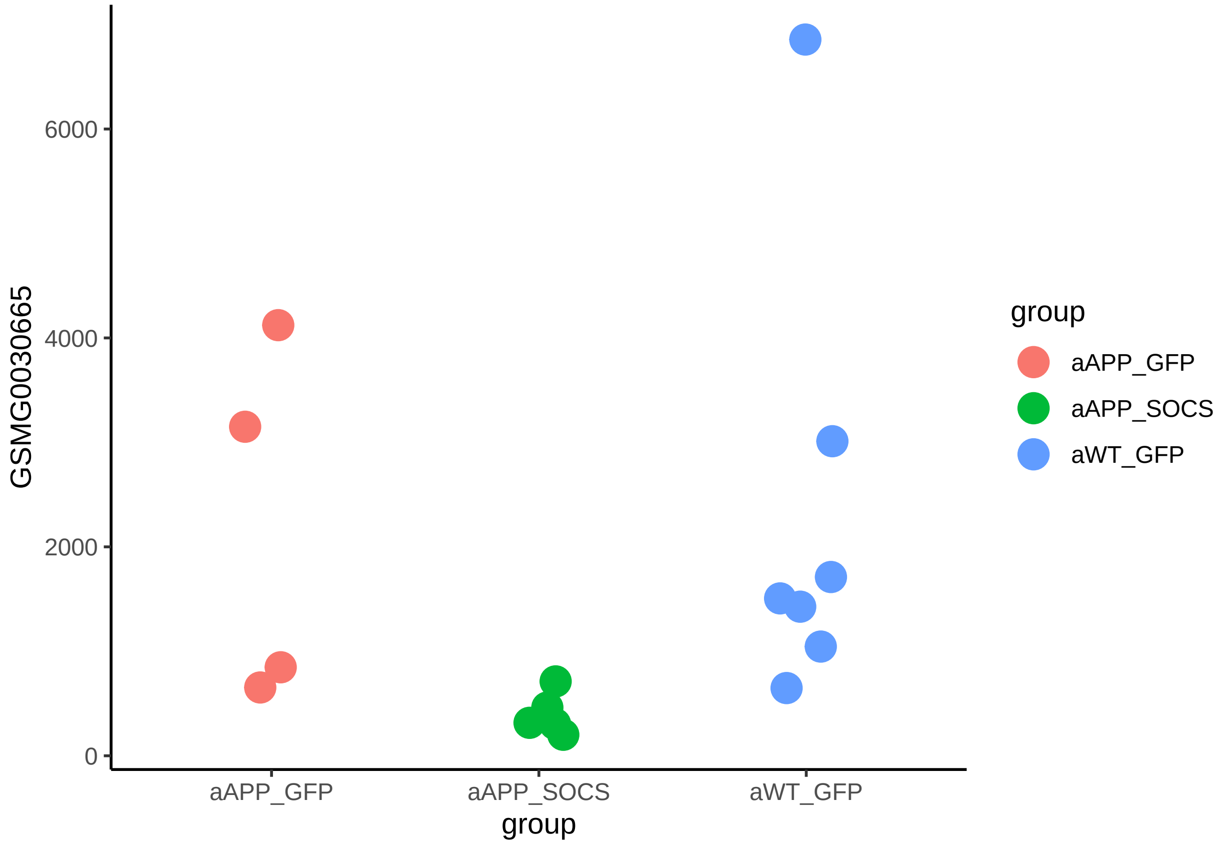
```
plot_gene_expr("GSMG0013278")
```



```
plot_gene_expr("GSMG0013663")
```



```
plot_gene_expr("GSMG0030665")
```


GSMG0030665
6000
4000
2000
0
aAPP_GFP
aAPP_SOCS
aWT_GFP
group
group
● aAPP_GFP
● aAPP_SOCS
● aWT_GFP