

# hpgltools examples using the fission dataset

atb [abelew@gmail.com](mailto:abelew@gmail.com)

2016-02-06

## Example hpgltool usage with a real data set (fission)

This document aims to provide further examples in how to use the hpgltools.

Note to self, the header has rmarkdown::pdf\_document instead of html\_document or html\_vignette because it gets some bullcrap error ‘margins too large’...

### Setting up

Here are the commands I invoke to get ready to play with new data, including everything required to install hpgltools, the software it uses, and the fission data.

```
## These first 4 lines are not needed once hpgltools is installed.  
## source("http://bioconductor.org/biocLite.R")  
## biocLite("devtools")  
## library(devtools)  
## install_github("elsayed-lab/hpgltools")  
library(hpgltools)  
require.auto("fission")  
  
## Creating a generic function for 'nchar' from package 'base' in package 'S4Vectors'  
  
library(fission)  
data(fission)  
knitr::opts_knit$set(progress=TRUE, verbose=TRUE, error=TRUE, fig.width=7, fig.height=7)
```

### Data import

All the work I do in Dr. El-Sayed’s lab makes some pretty hard assumptions about how data is stored. As a result, to use the fission data set I will do a little bit of shenanigans to match it to the expected format. Now that I have played a little with fission, I think its format is quite nice and am likely to have my experiment class instead be a SummarizedExperiment.

```
## Extract the meta data from the fission dataset  
meta <- as.data.frame(fission@colData)  
## Make conditions and batches  
meta$condition <- paste(meta$strain, meta$minute, sep=".")  
meta$batch <- meta$replicate  
meta$sample.id <- rownames(meta)  
## Grab the count data  
fission_data <- fission@assays$data$counts  
## This will make an experiment superclass called 'expt' and it contains  
## an ExpressionSet along with any arbitrary additional information one might want to include.  
## Along the way it writes a Rdata file which is by default called 'expt.Rdata'  
fission_expt <- create_expt(meta_dataframe=meta, count_dataframe=fission_data)
```

```

## create_expt(): This needs columns with conditions and batches in the sample sheet.

## create_experiment(): This function assumes some columns in the sample sheet:

## Sample.ID, Stage, Type, condition, batch

```

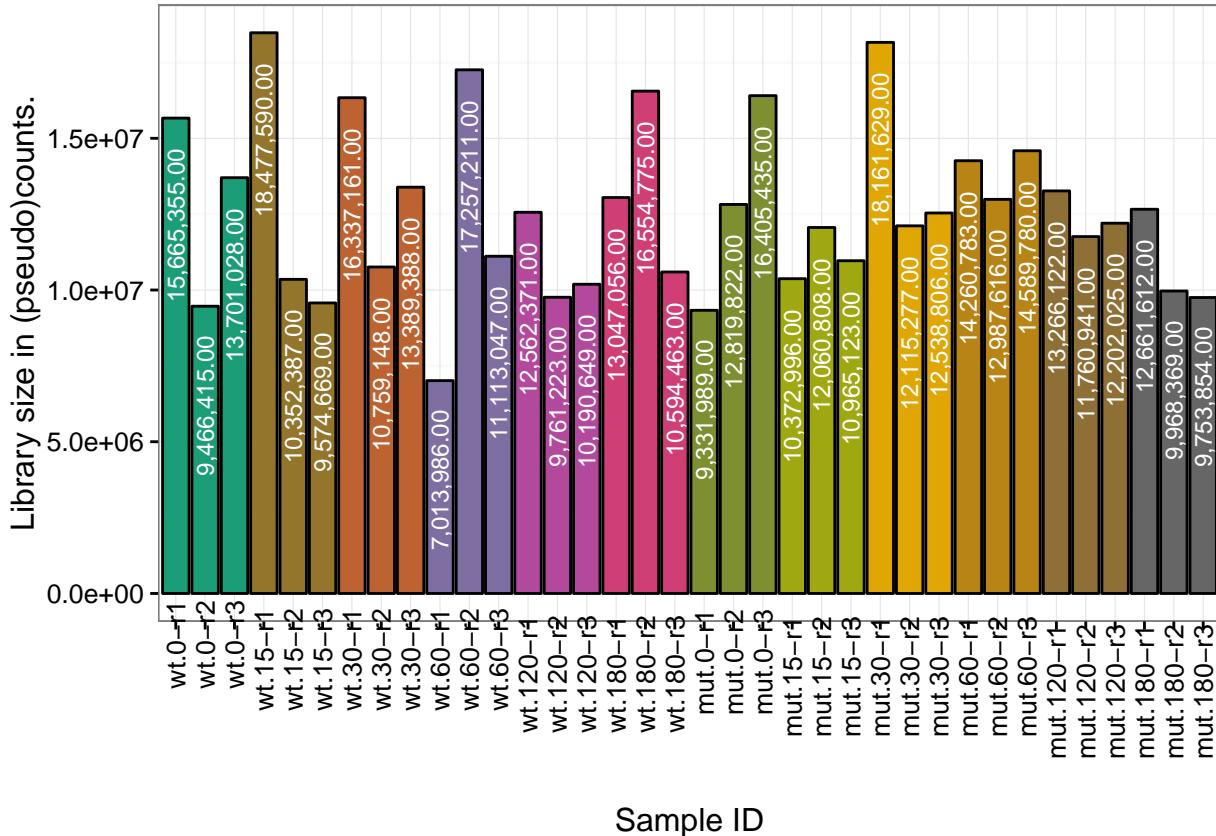
## Normalizing and exploring data

There are lots of toys we have learned to use to play with raw data and explore stuff like batch effects or non-canonical distributions or skewed counts. hpgltools provides some functionality to make this process easier. The graphs shown below and many more are generated with the wrapper ‘graph\_metrics()’ but that takes away the chance to explain the graphs as I generate them.

```

## First make a bar plot of the library sizes in the experiment.
## Notice that the colors were auto-chosen by create_expt() and they should
## be maintained throughout this process
fis_libsize <- hpgl_libsize(fission_expt)
fis_libsize

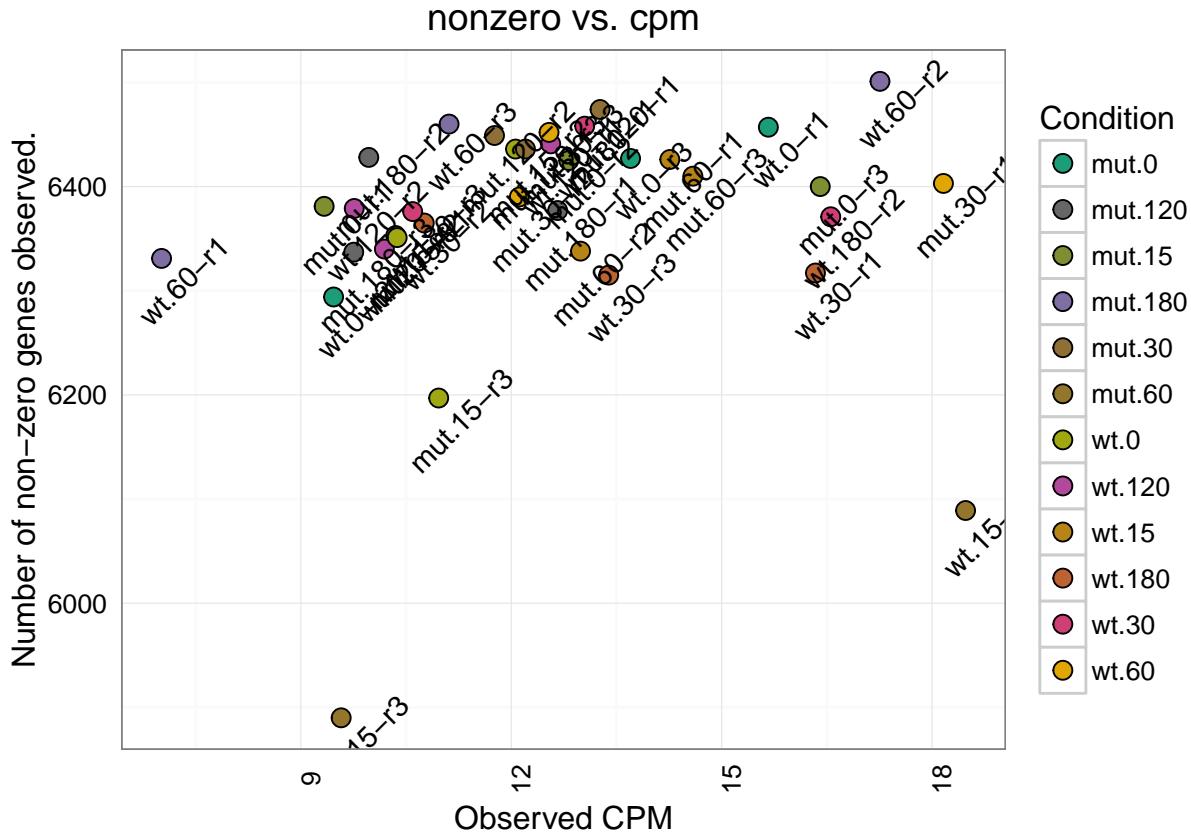
```



```

## Here we see that the wild type replicate 3 sample for 15 minutes has fewer non-zero genes than all i
fis_nonzero <- hpgl_nonzero(fission_expt, labels="boring", title="nonzero vs. cpm")
fis_nonzero

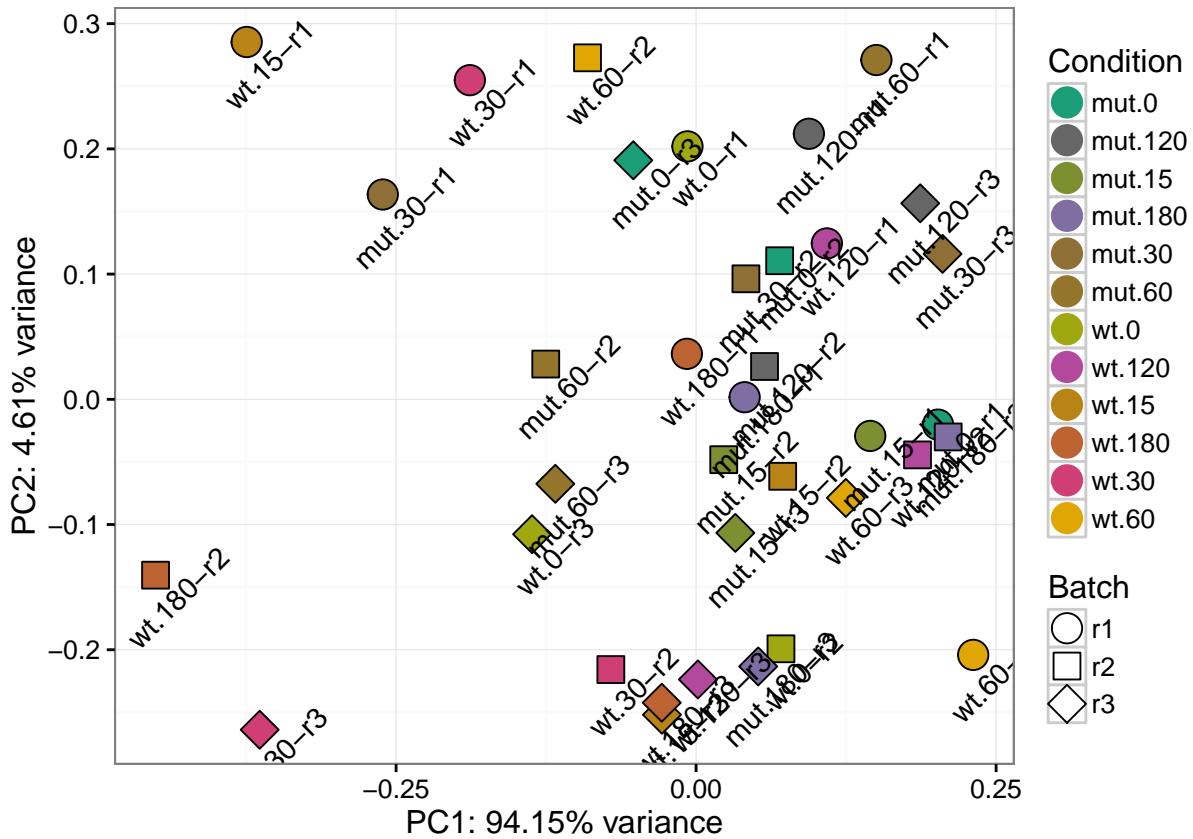
```



### An initial pca plot

In most cases, raw data does not cluster very well, lets see if that is also true for the fission experiment. Assuming it doesn't, lets normalize the data using the defaults (cpm, quantile, log2) and try again.

```
## Unsurprisingly, the raw data doesn't cluster well at all...
fis_rawpca <- hpgl_pca(fission_expt, expt_labels=fission_expt$condition)
fis_rawpca$plot
```



```

## So, normalize the data
norm_expt <- normalize_expt(fission_expt, transform="log2", norm="quant", convert="cpm")

## This function will replace the expt$expressionset slot with:
## log2(quant(cpm(data)))

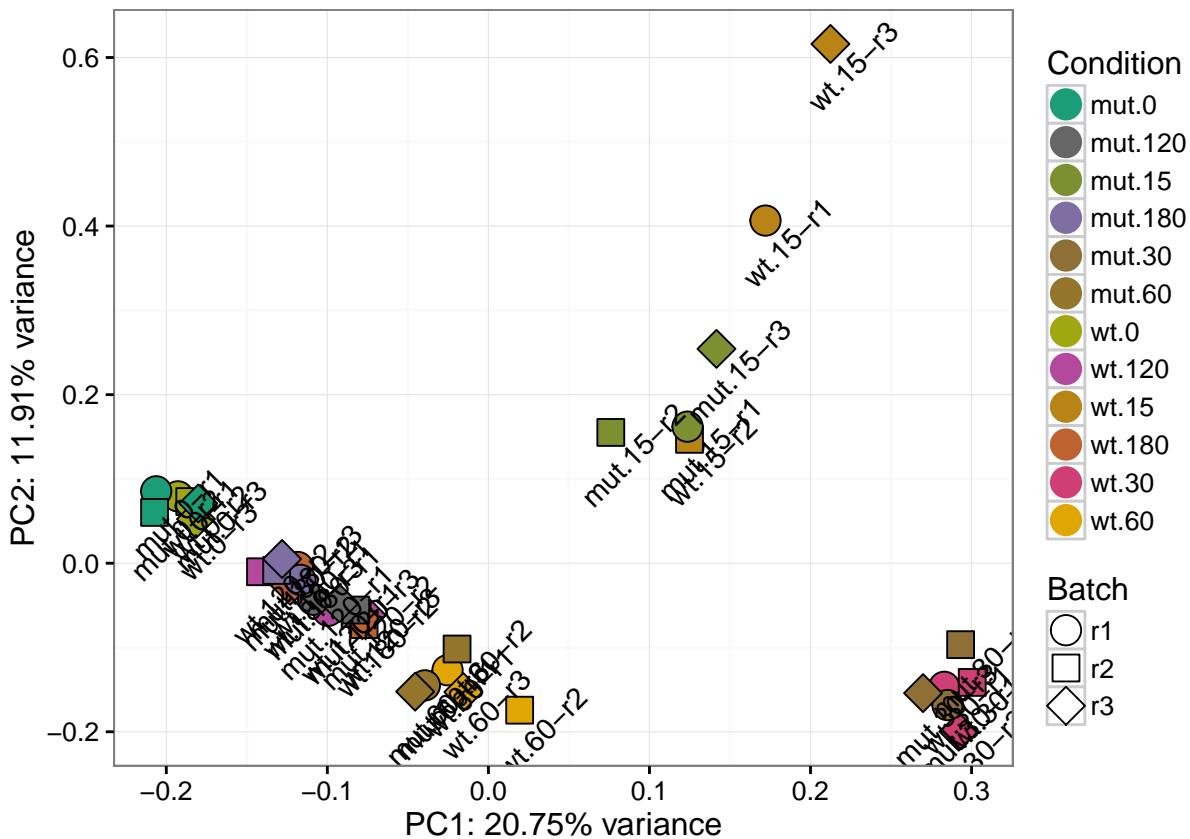
## It saves the current data into a slot named:
##   expt$backup_expressionset. It will also save copies of each step along the way
##   in expt$normalized with the corresponding libsizes. Keep the libsizes in mind
##   when invoking limma. The appropriate libsize is the non-log(cpm(normalized)).
##   This is most likely kept in the slot called:
##   'new_expt$normalized$normalized_counts$libsize' which is copied into
##   new_expt$best_libsize

## Filter low is false, this should likely be set to something, good
##   choices include ccbc, kofa, pofa (anything but FALSE). If you want this to
##   stay FALSE, keep in mind that if other normalizations are performed, then the
##   resulting libsizes are likely to be strange (potentially negative!)

## Not correcting the count-data for batch effects. If batch is
##   included in EdgerR/limma's model, then this is probably wise; but in extreme
##   batch effects this is a good parameter to play with.

```

```
## And try the pca again
fis_normPCA <- hpgl_pca(norm_expt, plot_labels="boring", title="normalized pca")
fis_normPCA$plot
```



```
normbatch_expt <- normalize_expt(fission_expt, transform="log2", norm="quant", convert="cpm", batch="sv")

## This function will replace the expt$expressionset slot with:
## log2(quant(cpm(batch-correct(data)))) 

## It saves the current data into a slot named:
## expt$backup_expressionset. It will also save copies of each step along the way
## in expt$normalized with the corresponding libsizes. Keep the libsizes in mind
## when invoking limma. The appropriate libsize is the non-log(cpm(normalized)).
## This is most likely kept in the slot called:
## 'new_expt$normalized$normalized_counts$libsize' which is copied into
## new_expt$best_libsize

## Filter low is false, this should likely be set to something, good
## choices include ccbc, kofa, pofa (anything but FALSE). If you want this to
## stay FALSE, keep in mind that if other normalizations are performed, then the
## resulting libsizes are likely to be strange (potentially negative!)

## batch_counts: Before batch correction, 47195 entries 0<x<1.
```

```

## batch_counts: Using sva::fsva for batch correction.

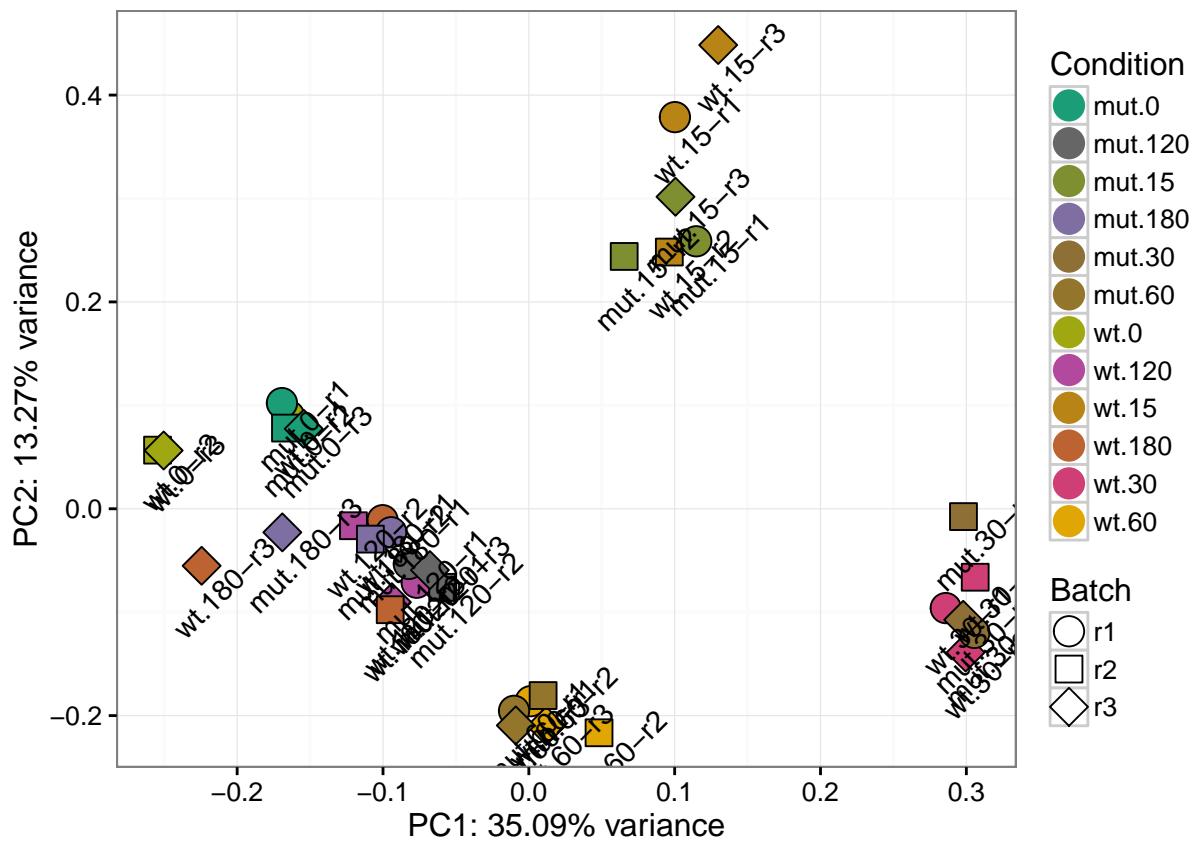
## Number of significant surrogate variables is: 1
## Iteration (out of 5 ):1 2 3 4 5

## The number of elements which are < 0 after batch correction is: 1383

## transform_counts: Found 1383 values equal to 0, adding 0.5
## to the matrix.

fis_normbatchpca <- hpgl_pca(normbatch_expt, title="Normalized PCA with batch effect correction.")
fis_normbatchpca$plot

```



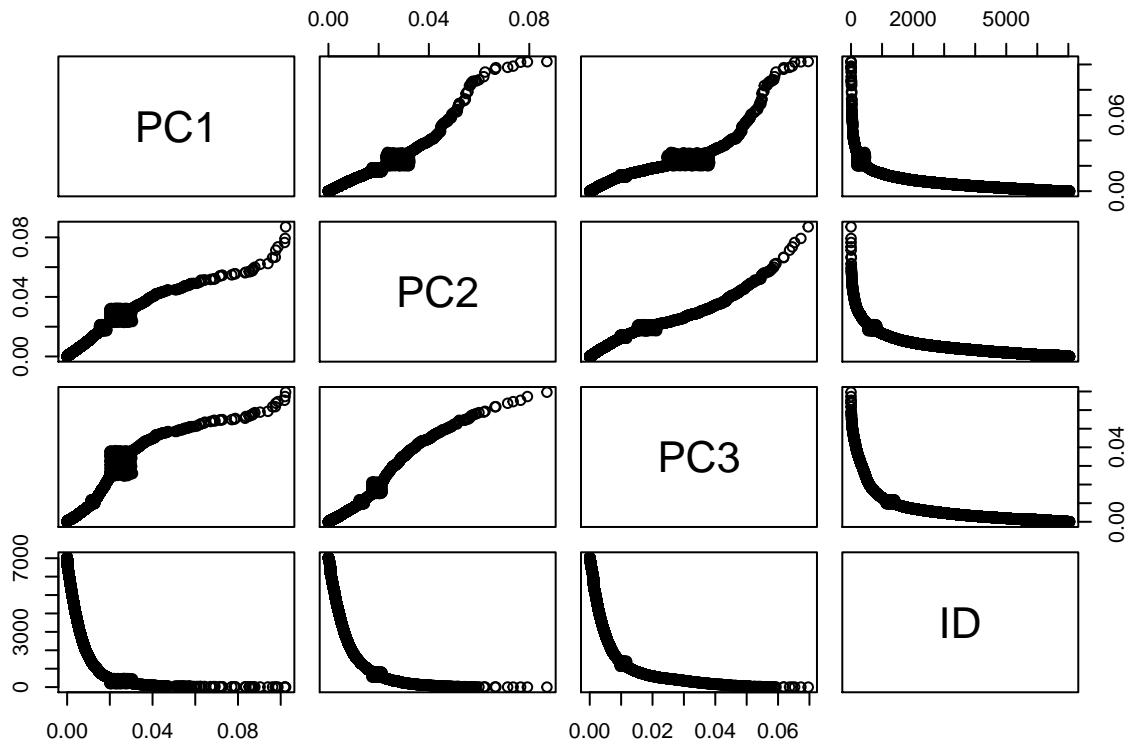
```

## ok, that caused the 0, 60, 15, and 30 minute samples to cluster nicely
## the 120 and 180 minute samples are still a bit tight

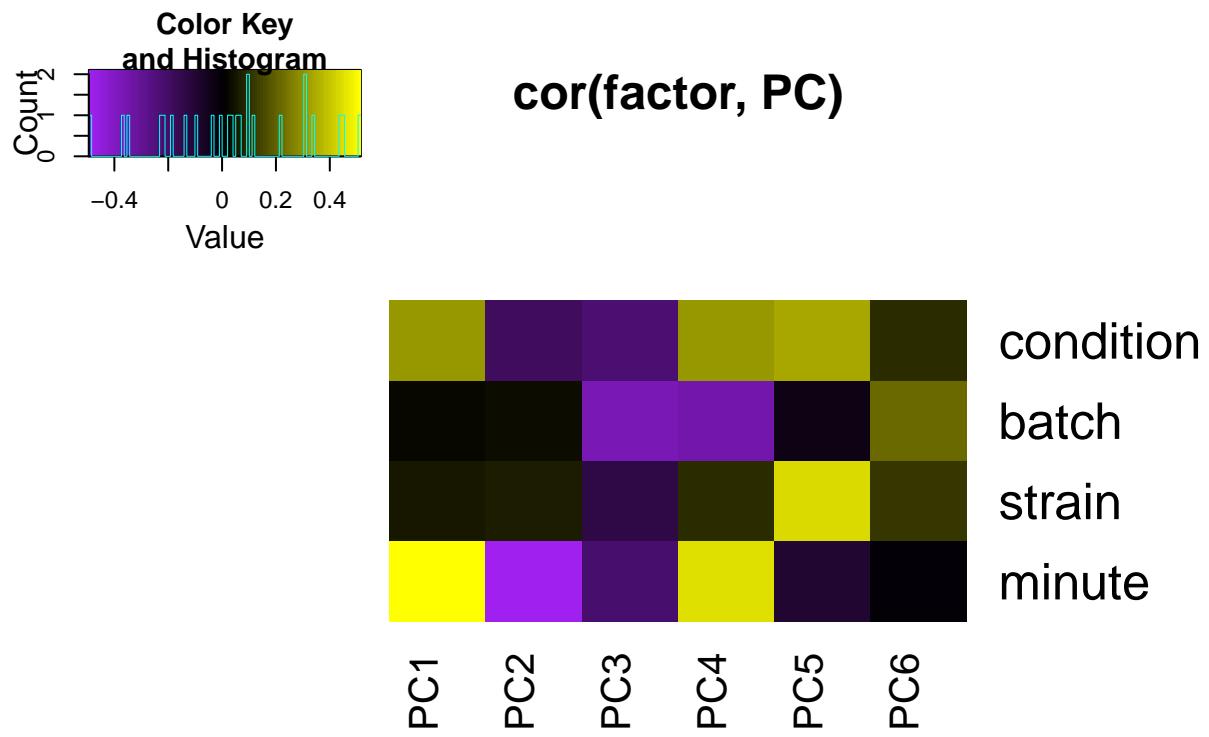
## pca_information provides some more information about the call to
## fast.svd that went into making the pca plot
fis_info <- pca_information(norm_expt, expt_factors=c("condition","batch","strain","minute"), num_compono

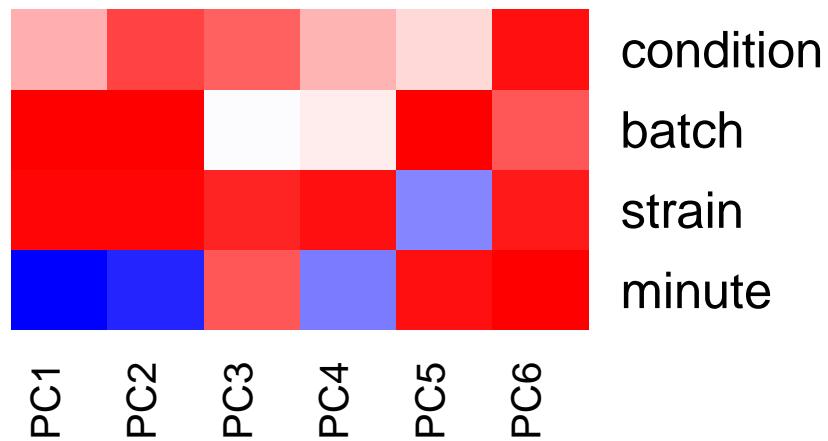
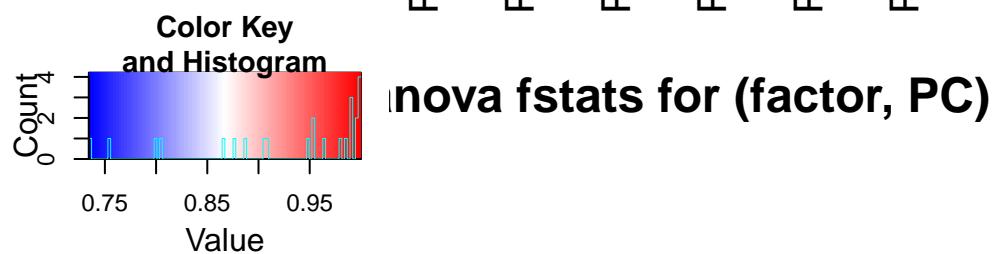
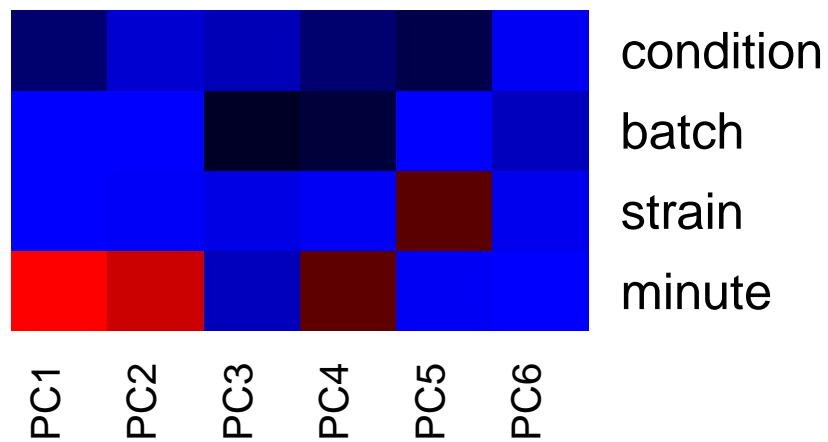
```

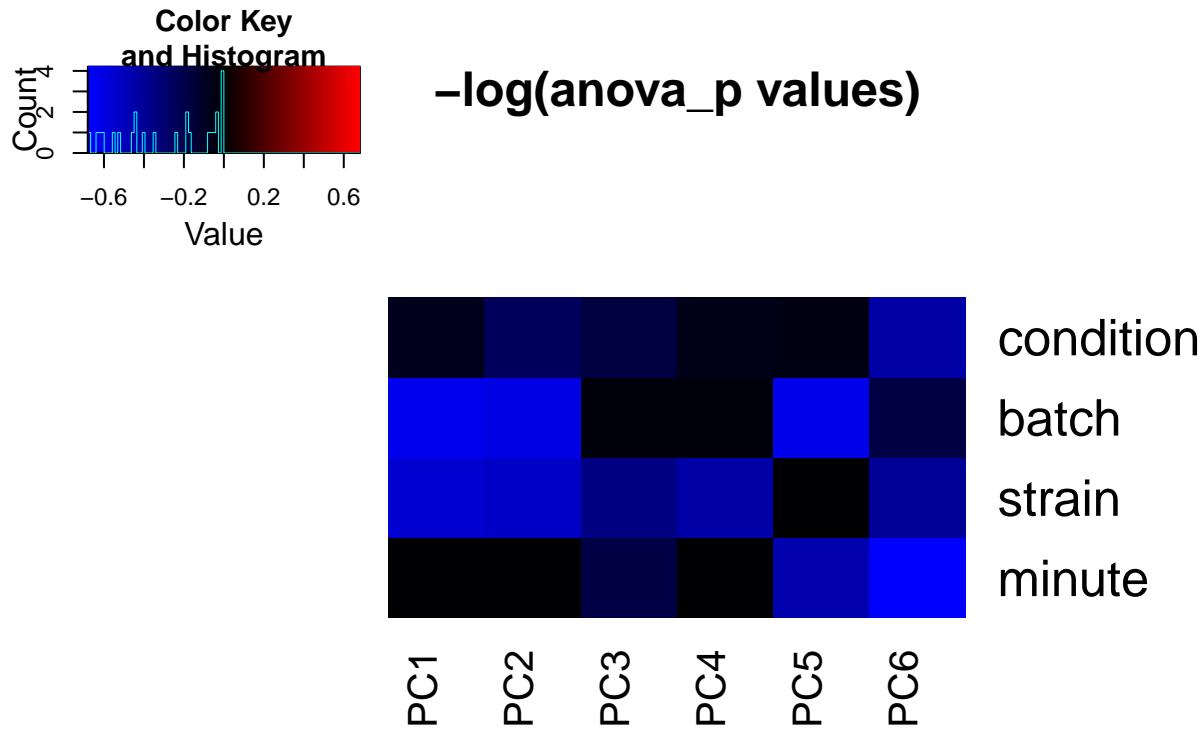
## The more shallow the curves in these plots, the more genes responsible for this principle component.



```
## [1] "PC1: 20.75% variance"
## [1] "PC2: 11.91% variance"
```







```
## The r^2 table shows that quite a lot of the variance in the data is explained by condition
head(fis_info$rsquared_table)
```

```
##   prop_var cumulative_prop_var condition  batch strain minute
## 1    20.752           20.752    98.743  0.069  0.315 98.053
## 2    11.911           32.663    87.067  0.772  0.443 80.859
## 3     7.702           40.365   15.586 13.626  1.889 11.256
## 4     6.138           46.503   76.204 12.331  0.997 65.848
## 5     4.863           51.366   70.220  0.917 19.633 37.284
## 6     3.891           55.257   74.218  4.921  1.369 67.245
```

```
## We can look at the correlation between the principle components and the factors in the experiment
## in this case looking at condition/batch vs the first 4 components.
fis_info$pca_cor
```

```
##          PC1        PC2        PC3        PC4        PC5
## condition 0.30380317 -0.18690226 -0.2253680  0.30765103  0.33650975
## batch      0.02397345  0.03691367 -0.3645445 -0.35037843 -0.03137802
## strain     0.05616874  0.06653382 -0.1374555  0.09987154  0.44308965
## minute     0.51541477 -0.49466299 -0.2140827  0.44642328 -0.09814799
##          PC6
## condition  0.099040592
## batch       0.215088641
## strain      0.117025631
## minute     -0.005377346
```

```
## And p-values to lend some credence(or not to those assertions)
fis_info$anova_p
```

```

##          PC1        PC2        PC3        PC4        PC5
## condition 0.071650310 0.275057211 0.18631370 0.067953327 0.044775771
## batch      0.889620757 0.830751187 0.02882176 0.036170535 0.855842993
## strain     0.744896750 0.699835860 0.42403995 0.562229292 0.006801267
## minute     0.001295445 0.002163481 0.20992903 0.006348466 0.569028441
##          PC6
## condition 0.5655026
## batch      0.2077438
## strain     0.4966850
## minute     0.9751694

## Try again with batch removed data
batchnorm_expt <- normalize_expt(fission_expt, batch="limma", norm="quant", transform="log2", convert=""

## This function will replace the expt$expressionset slot with:

## log2(quant(cpm(batch-correct(data)))) 

## It saves the current data into a slot named:
##   expt$backup_expressionset. It will also save copies of each step along the way
##   in expt$normalized with the corresponding libsizes. Keep the libsizes in mind
##   when invoking limma. The appropriate libsize is the non-log(cpm(normalized)).
##   This is most likely kept in the slot called:
##   'new_expt$normalized$normalized_counts$libsize' which is copied into
##   new_expt$best_libsize

## Filter low is false, this should likely be set to something, good
##   choices include ccbc, kofa, pofa (anything but FALSE). If you want this to
##   stay FALSE, keep in mind that if other normalizations are performed, then the
##   resulting libsizes are likely to be strange (potentially negative!)

## batch_counts: Before batch correction, 47195 entries 0<x<1.

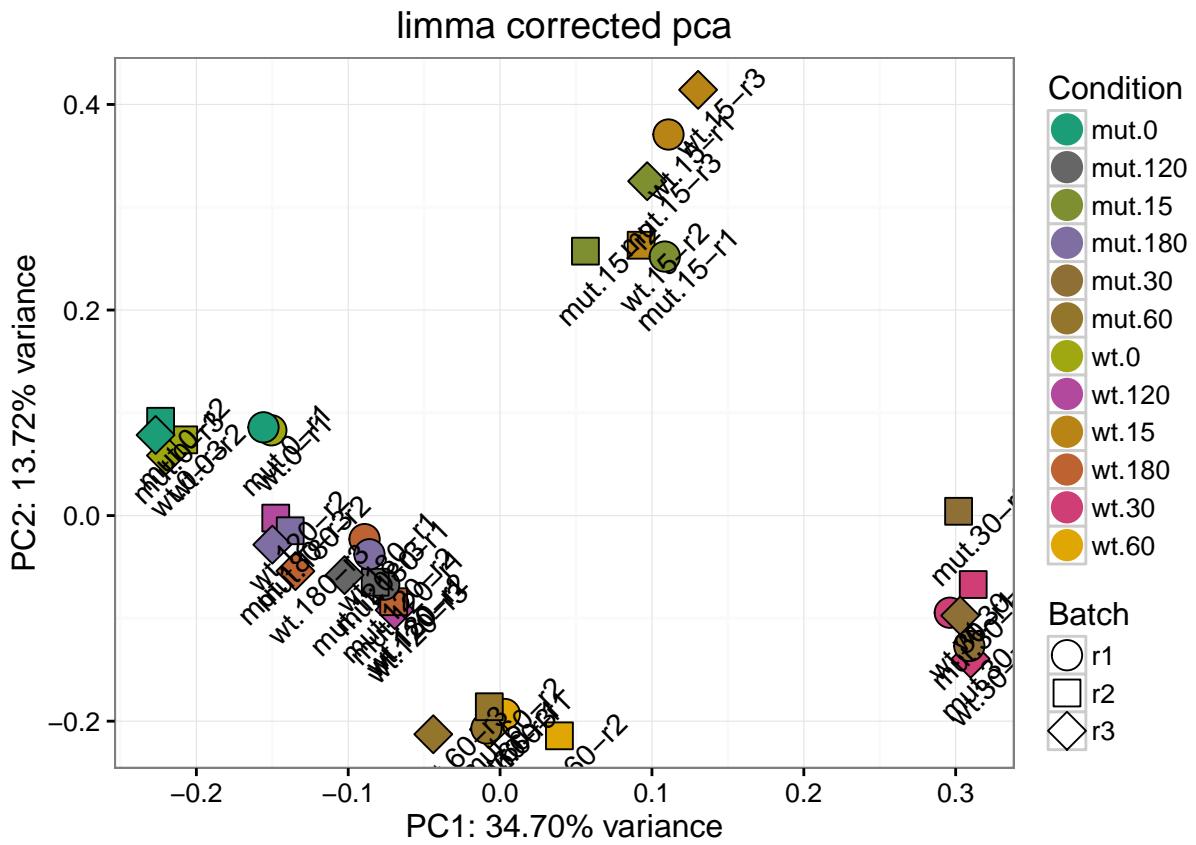
## batch_counts: Using limma's removeBatchEffect to remove batch effect.

## The number of elements which are < 0 after batch correction is: 1689

## transform_counts: Found 1689 values equal to 0, adding 0.5
##   to the matrix.

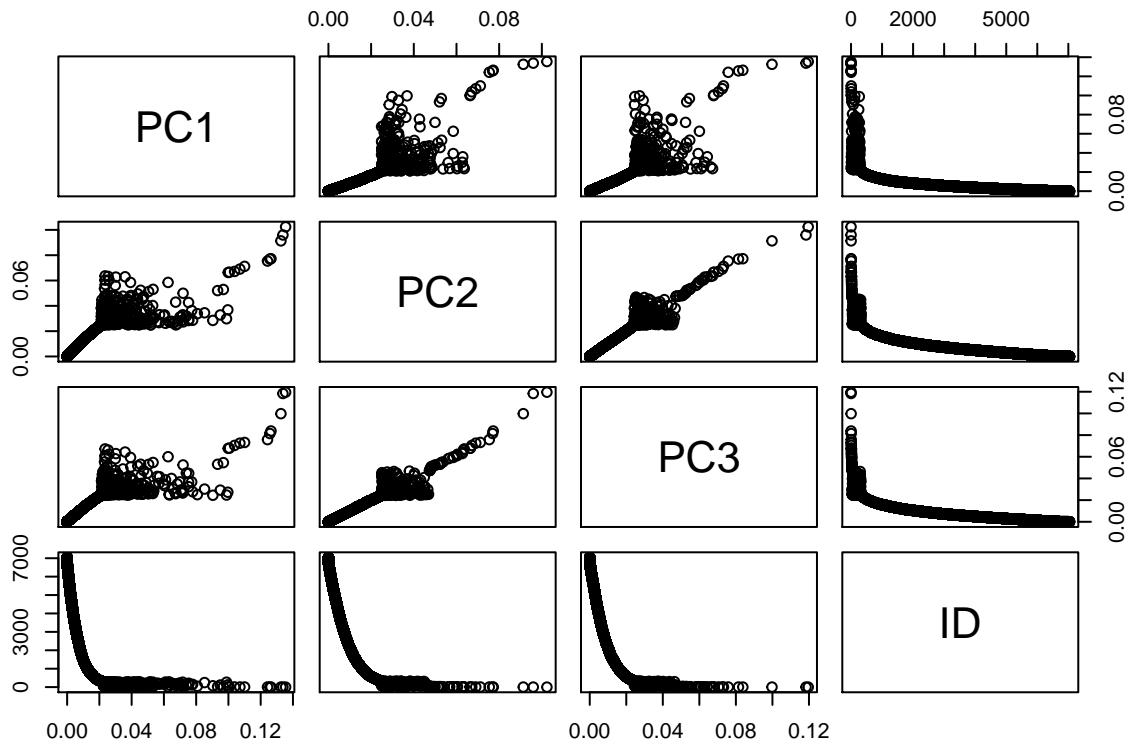
fis_batchnormpca <- hpgl_pca(batchnorm_expt, plot_title="limma corrected pca")
fis_batchnormpca$plot

```

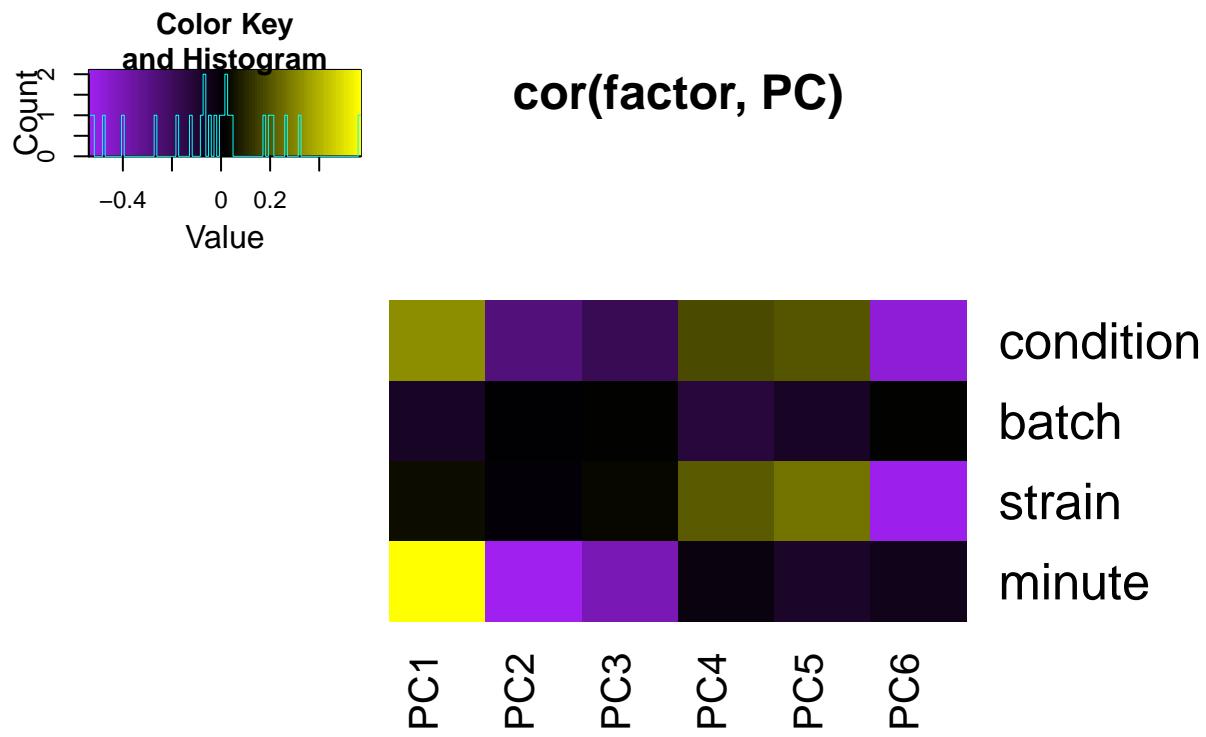


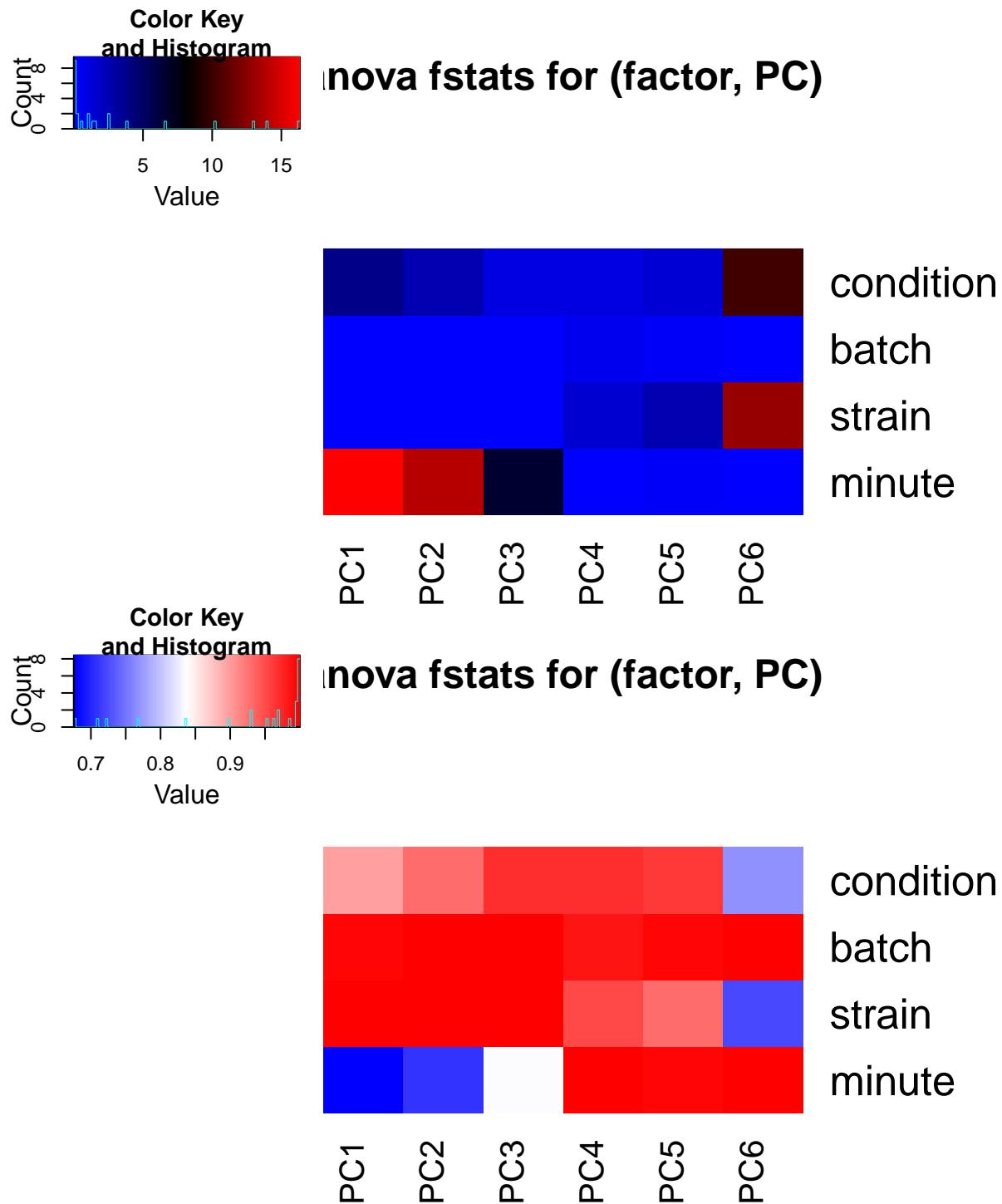
```
test_pca <- pca_information(batchnorm_expt, expt_factors=c("condition", "batch", "strain", "minute"), num_
```

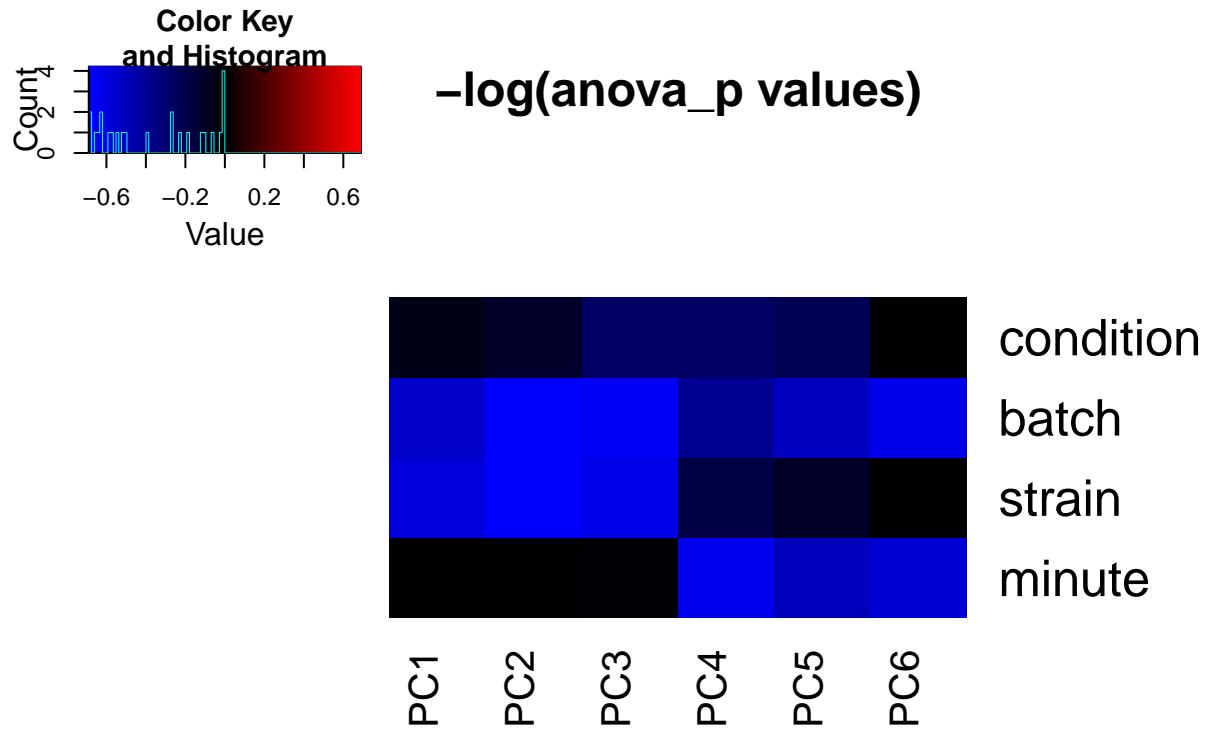
```
## The more shallow the curves in these plots, the more genes responsible for this principle component.
```



```
## [1] "PC1: 34.70% variance"
## [1] "PC2: 13.72% variance"
```







Interesting, the batch normalized pca plot looks much the same as the normalized. The variances are in fact pretty much the exact same...

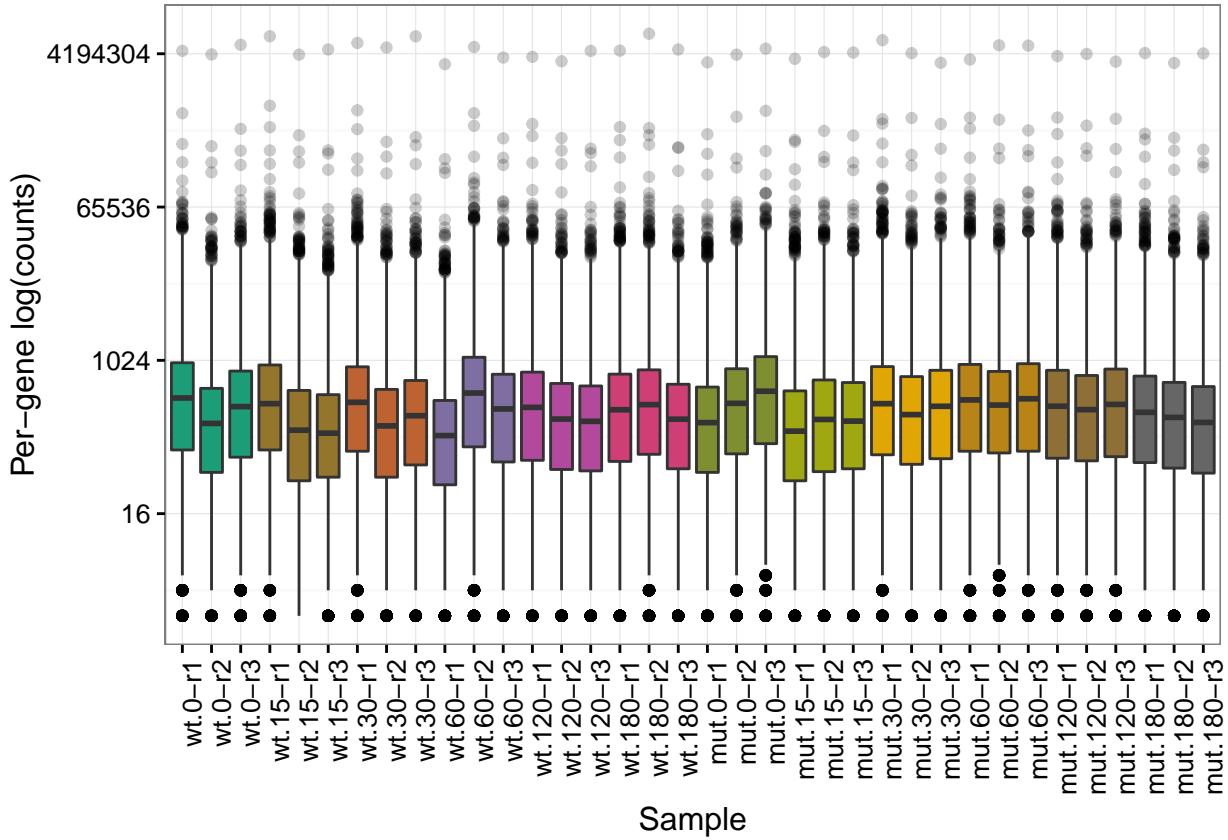
## Look at the data distributions

We have some tools which provide visualizations of the distribution of the data:

```
hpgl_boxplot(fission_expt)
```

```
## I am reasonably sure this should be log scaled and am setting it.
## If this is incorrect, set scale='raw'

## Warning: Removed 24130 rows containing non-finite values (stat_boxplot).
```



```

sf_expt <- normalize_expt(fission_expt, norm="sf")

## This function will replace the expt$expressionset slot with:
## sf(data)

## It saves the current data into a slot named:
##   expt$backup_expressionset. It will also save copies of each step along the way
##   in expt$normalized with the corresponding libsizes. Keep the libsizes in mind
##   when invoking limma. The appropriate libsize is the non-log(cpm(normalized)).
##   This is most likely kept in the slot called:
##   'new_expt$normalized$normalized_counts$libsize' which is copied into
##   new_expt$best_libsize

## Filter low is false, this should likely be set to something, good
##   choices include ccbc, kofa, pofa (anything but FALSE). If you want this to
##   stay FALSE, keep in mind that if other normalizations are performed, then the
##   resulting libsizes are likely to be strange (potentially negative!)

## Leaving the data in its current base format, keep in mind that
##   some metrics are easier to see when the data is log2 transformed, but
##   EdgeR/DESeq don't like transformed data.

## Leaving the data unconverted. It is often advisable to cpm/rpk
##   the data to normalize for sampling differences, keep in mind though that rpk
##   has some annoying biases, and voom() by default does a cpm (though hpgl_voom()
##   will try to detect this).

```

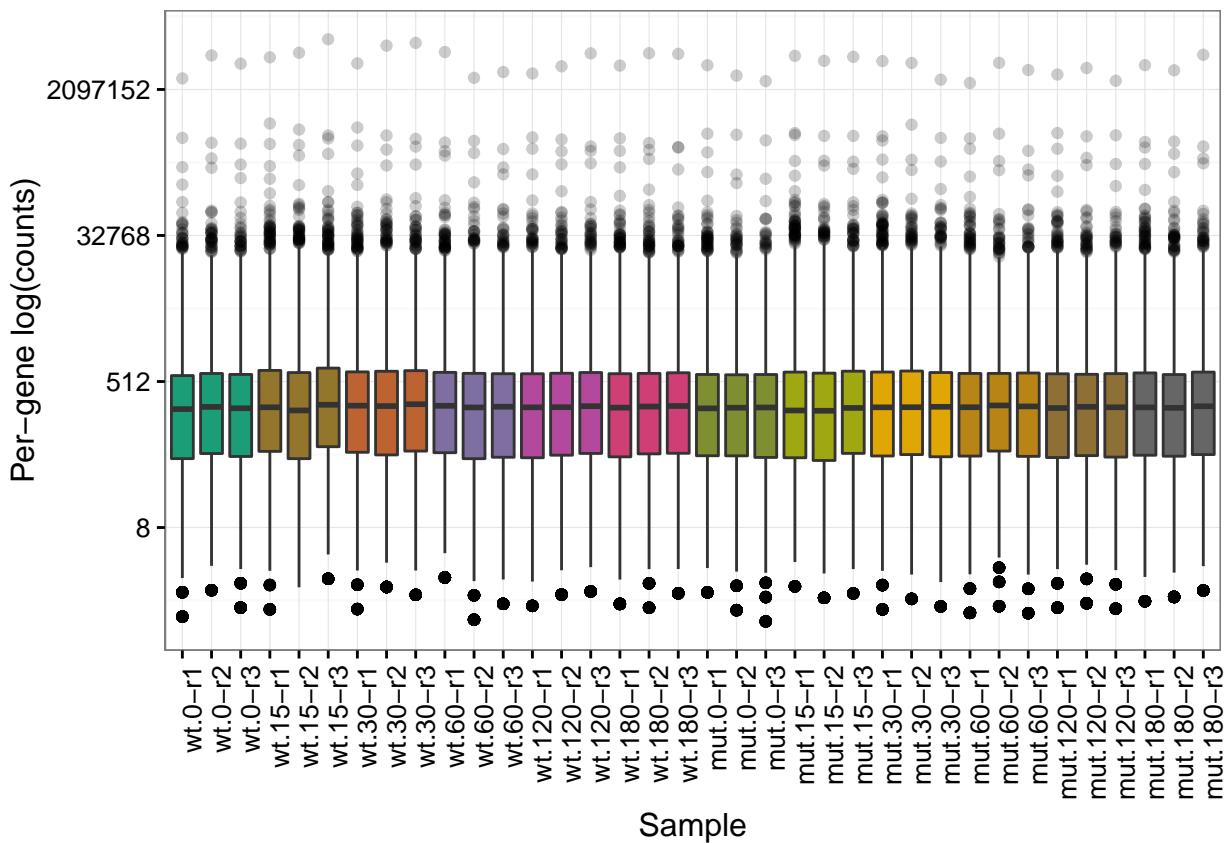
```
## Not correcting the count-data for batch effects. If batch is
## included in EdgerR/limma's model, then this is probably wise; but in extreme
## batch effects this is a good parameter to play with.
```

```
## Warning: replacing previous import by 'ggplot2::Position' when loading
## 'DESeq2'
```

```
hpgl_boxplot(sf_expt)
```

```
## I am reasonably sure this should be log scaled and am setting it.
## If this is incorrect, set scale='raw'
```

```
## Warning: Removed 24130 rows containing non-finite values (stat_boxplot).
```



```
tm_expt <- normalize_expt(fission_expt, norm="tmm")
```

```
## This function will replace the expt$expressionset slot with:
```

```
## tmm(data)
```

```
## It saves the current data into a slot named:
## expt$backup_expressionset. It will also save copies of each step along the way
## in expt$normalized with the corresponding libsizes. Keep the libsizes in mind
## when invoking limma. The appropriate libsize is the non-log(cpm(normalized)).
```

```

## This is most likely kept in the slot called:
## 'new_expt$normalized$normalized_counts$libsize' which is copied into
## new_expt$best_libsize

## Filter low is false, this should likely be set to something, good
## choices include ccbc, kofa, pofa (anything but FALSE). If you want this to
## stay FALSE, keep in mind that if other normalizations are performed, then the
## resulting libsizes are likely to be strange (potentially negative!)

## Leaving the data in its current base format, keep in mind that
## some metrics are easier to see when the data is log2 transformed, but
## EdgeR/DESeq don't like transformed data.

## Leaving the data unconverted. It is often advisable to cpm/rpkm
## the data to normalize for sampling differences, keep in mind though that rpkm
## has some annoying biases, and voom() by default does a cpm (though hpgl_voom()
## will try to detect this).

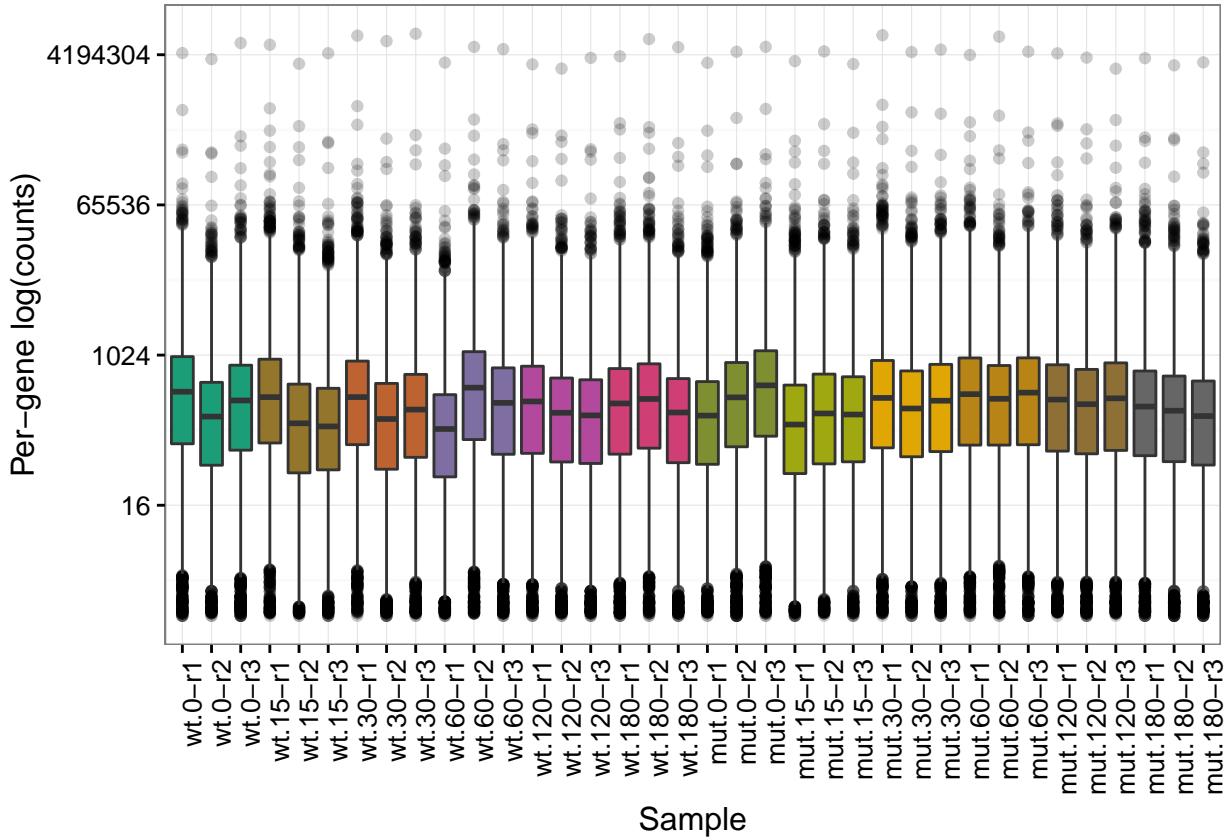
## Not correcting the count-data for batch effects. If batch is
## included in EdgerR/limma's model, then this is probably wise; but in extreme
## batch effects this is a good parameter to play with.

hpgl_boxplot(tm_expt)

## I am reasonably sure this should be log scaled and am setting it.
## If this is incorrect, set scale='raw'

## Warning: Removed 24130 rows containing non-finite values (stat_boxplot).

```



```
rle_expt <- normalize_expt(fission_expt, norm="rle")

## This function will replace the expt$expressionset slot with:

## rle(data)

## It saves the current data into a slot named:
##   expt$backup_expressionset. It will also save copies of each step along the way
##   in expt$normalized with the corresponding libsizes. Keep the libsizes in mind
##   when invoking limma. The appropriate libsize is the non-log(cpm(normalized)).
##   This is most likely kept in the slot called:
##   'new_expt$normalized$normalized_counts$libsize' which is copied into
##   new_expt$best_libsize

## Filter low is false, this should likely be set to something, good
##   choices include ccbc, kofa, pofa (anything but FALSE). If you want this to
##   stay FALSE, keep in mind that if other normalizations are performed, then the
##   resulting libsizes are likely to be strange (potentially negative!)

## Leaving the data in its current base format, keep in mind that
##   some metrics are easier to see when the data is log2 transformed, but
##   EdgeR/DESeq don't like transformed data.

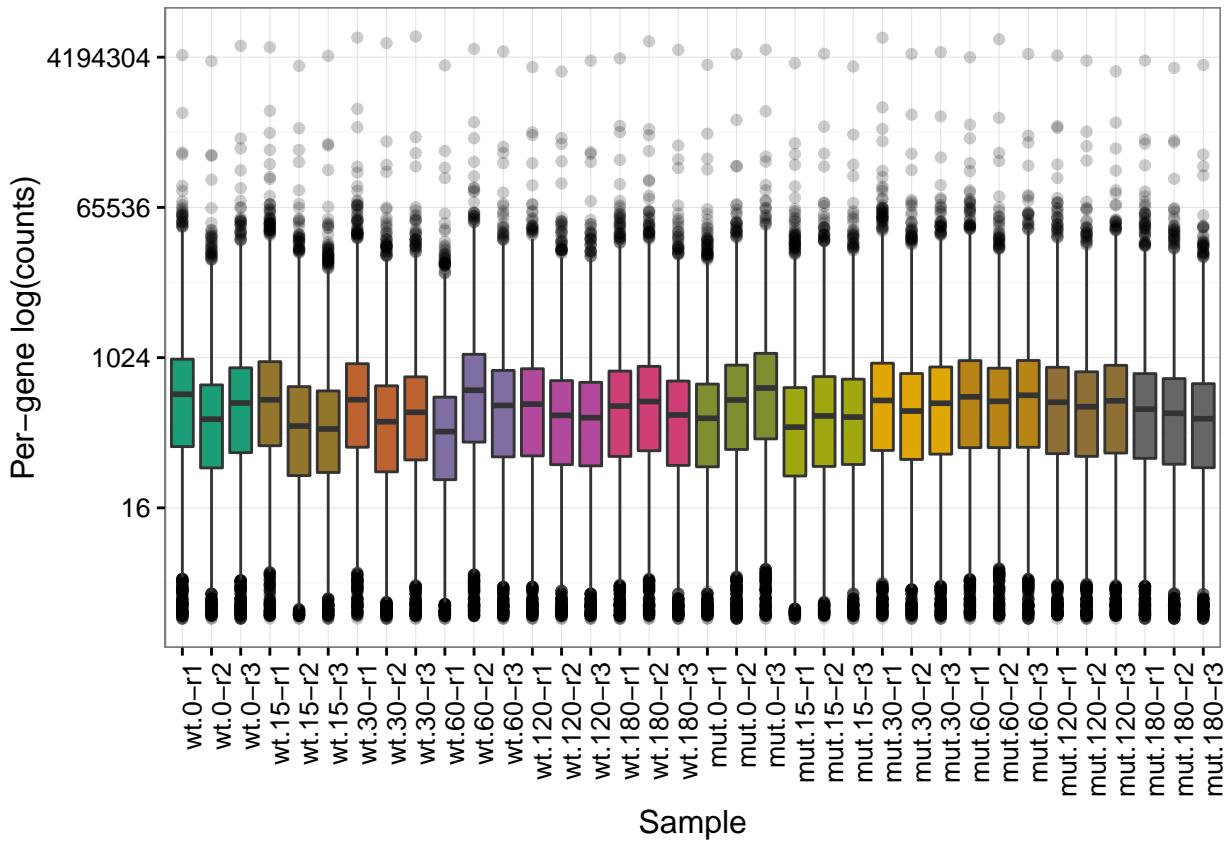
## Leaving the data unconverted. It is often advisable to cpm/rpk
##   the data to normalize for sampling differences, keep in mind though that rpk
##   has some annoying biases, and voom() by default does a cpm (though hpgl_voom()
##   will try to detect this).
```

```
## Not correcting the count-data for batch effects. If batch is
## included in EdgerR/limma's model, then this is probably wise; but in extreme
## batch effects this is a good parameter to play with.
```

```
hpgl_boxplot(rle_expt)
```

```
## I am reasonably sure this should be log scaled and am setting it.
## If this is incorrect, set scale='raw'
```

```
## Warning: Removed 24130 rows containing non-finite values (stat_boxplot).
```



```
up_expt <- normalize_expt(fission_expt, norm="upperquartile")
```

```
## This function will replace the expt$expressionset slot with:
```

```
## upperquartile(data)
```

```
## It saves the current data into a slot named:
## expt$backup_expressionset. It will also save copies of each step along the way
## in expt$normalized with the corresponding libsizes. Keep the libsizes in mind
## when invoking limma. The appropriate libsize is the non-log(cpm(normalized)).
## This is most likely kept in the slot called:
## 'new_expt$normalized$normalized_counts$libsize' which is copied into
## new_expt$best_libsize
```

```

## Filter low is false, this should likely be set to something, good
## choices include ccbc, kofa, pofa (anything but FALSE). If you want this to
## stay FALSE, keep in mind that if other normalizations are performed, then the
## resulting libsizes are likely to be strange (potentially negative!)

## Leaving the data in its current base format, keep in mind that
## some metrics are easier to see when the data is log2 transformed, but
## EdgeR/DESeq don't like transformed data.

## Leaving the data unconverted. It is often advisable to cpm/rpk
## the data to normalize for sampling differences, keep in mind though that rpk
## has some annoying biases, and voom() by default does a cpm (though hpgl_voom()
## will try to detect this).

## Not correcting the count-data for batch effects. If batch is
## included in EdgerR/limma's model, then this is probably wise; but in extreme
## batch effects this is a good parameter to play with.

```

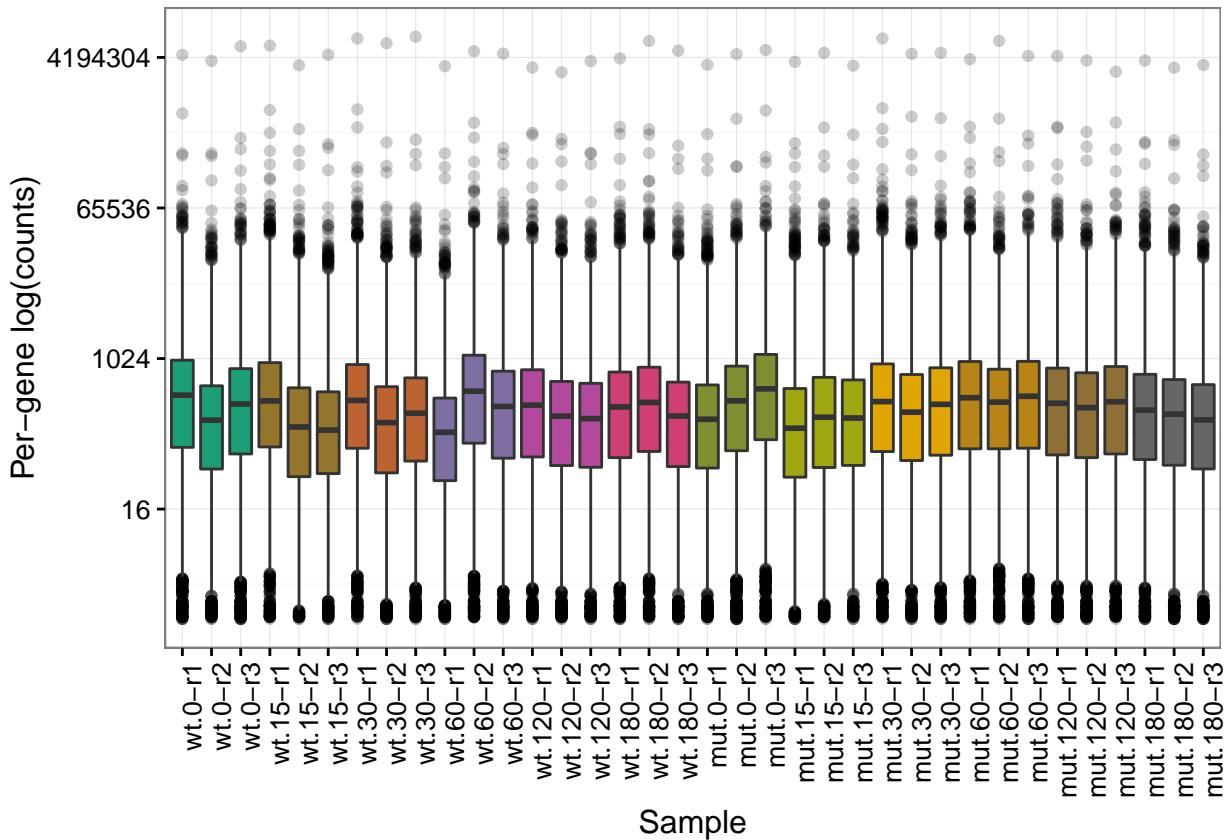
```
hpgl_boxplot(up_expt)
```

```

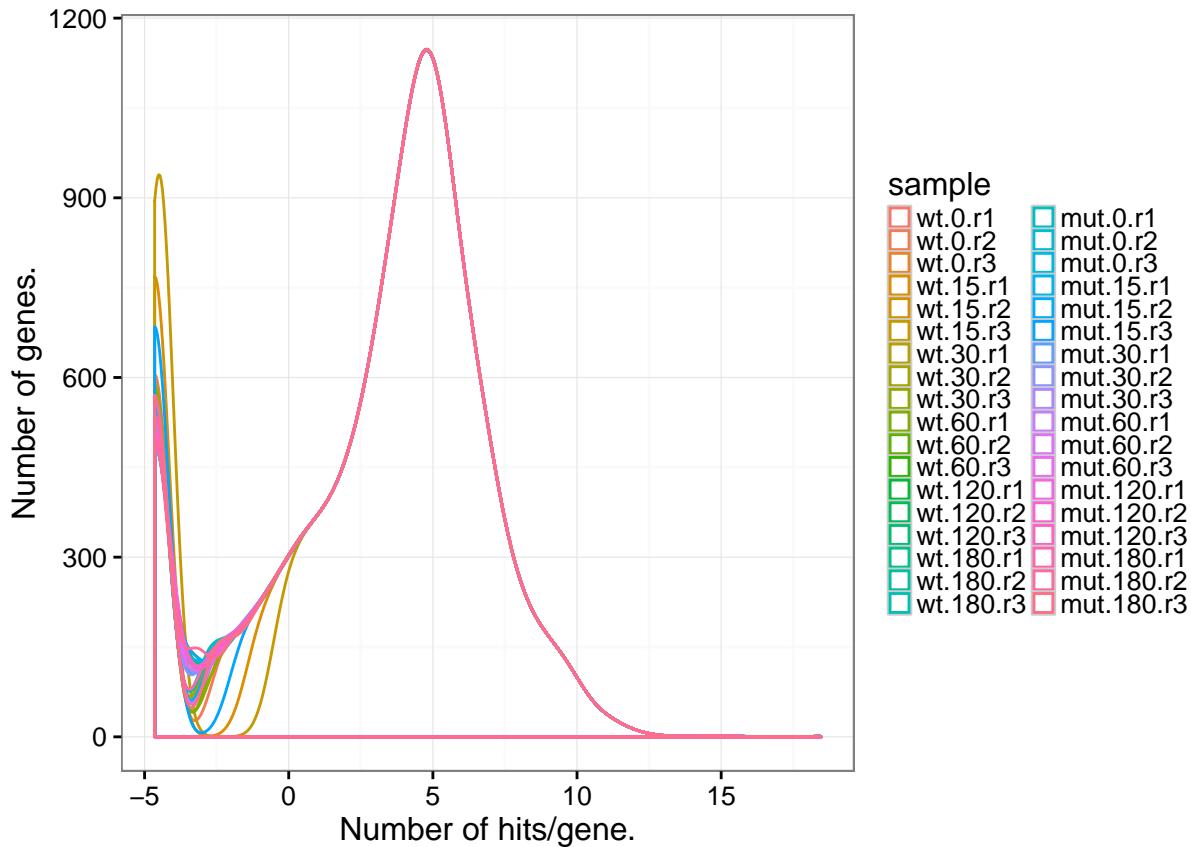
## I am reasonably sure this should be log scaled and am setting it.
## If this is incorrect, set scale='raw'

## Warning: Removed 24130 rows containing non-finite values (stat_boxplot).

```

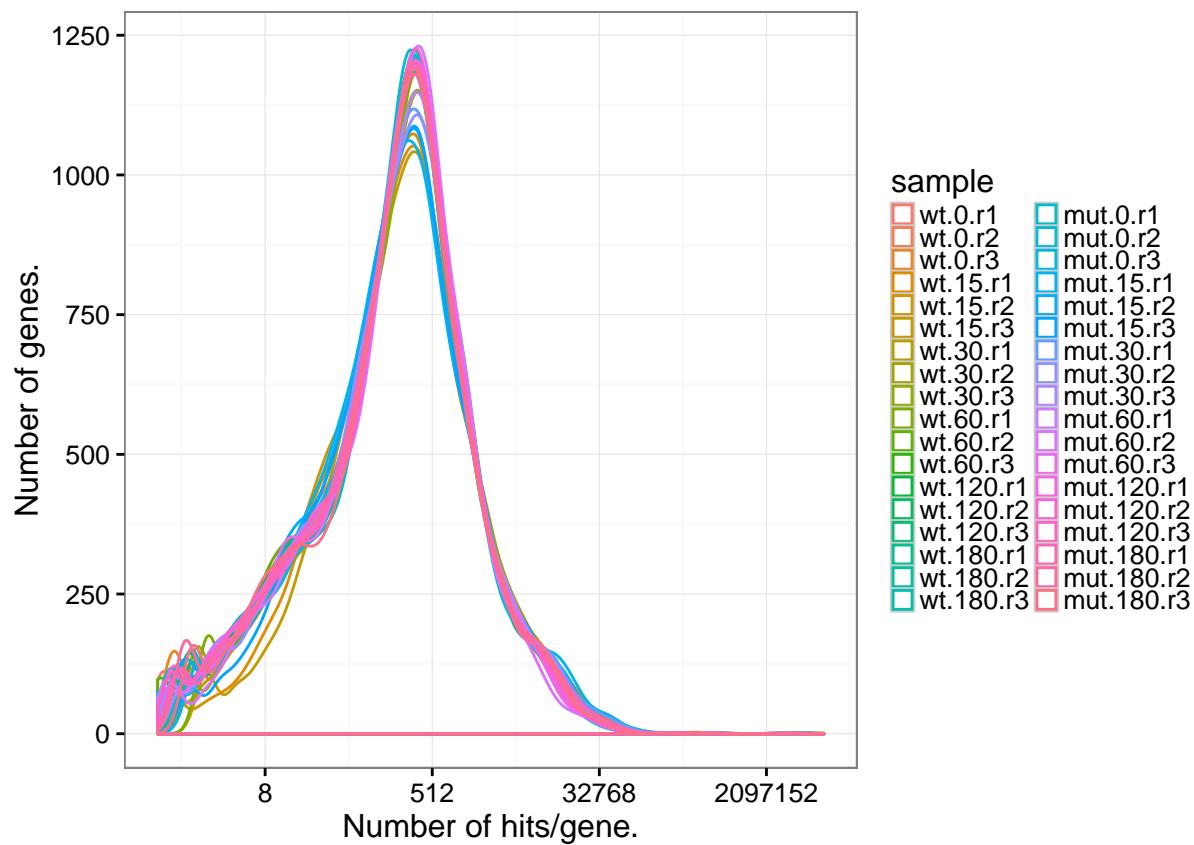


```
hpgl_density(norm_expt)
```



```
hpgl_density(sf_expt)
```

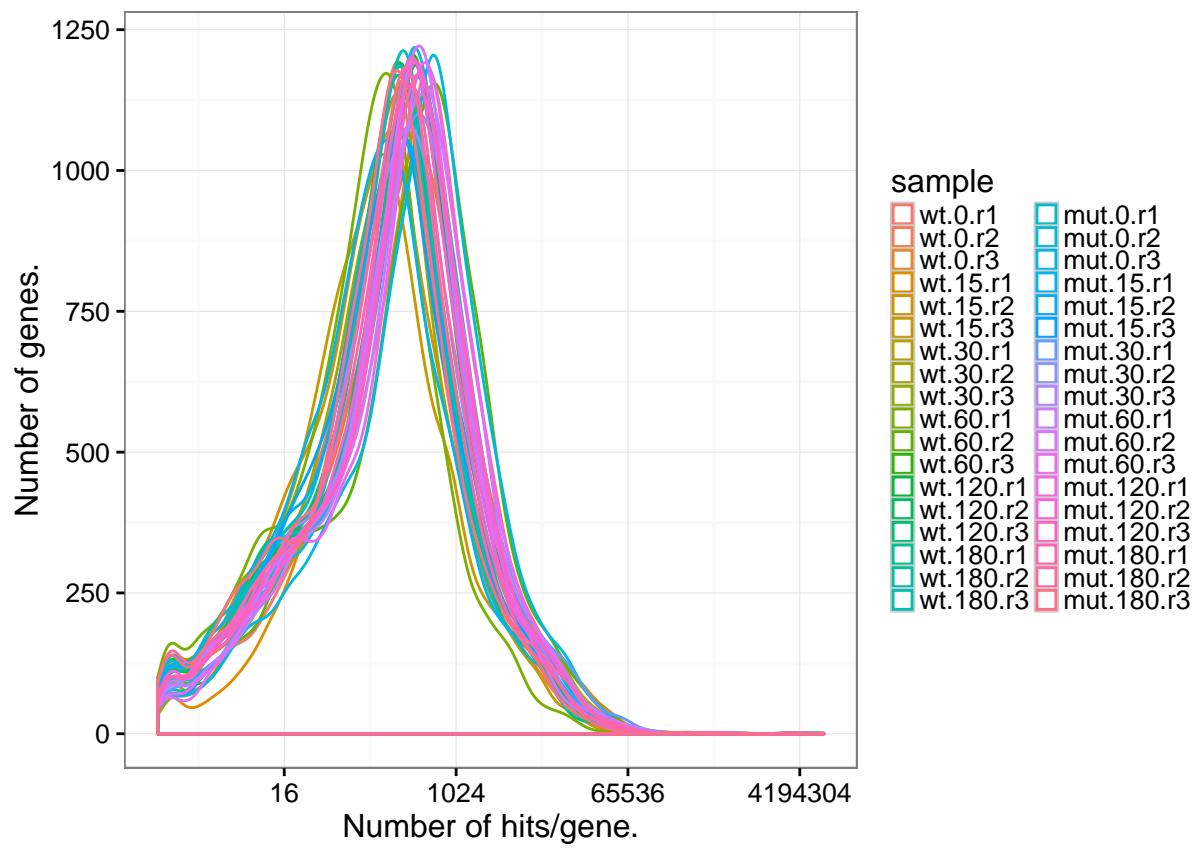
```
## This data will benefit from being displayed on the log scale.  
## If this is not desired, set scale='raw'  
## Warning: Removed 24130 rows containing non-finite values (stat_density).
```



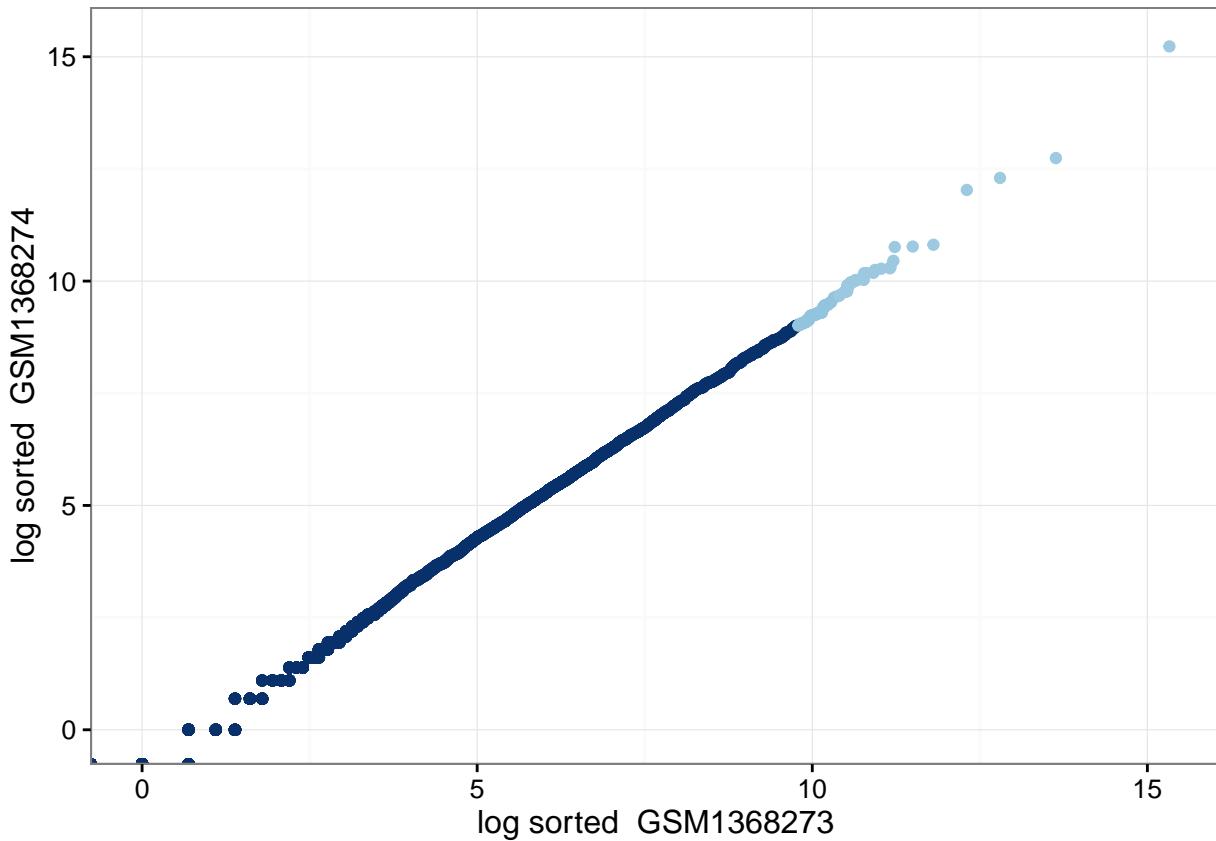
```
hpgl_density(tm_expt)
```

```
## This data will benefit from being displayed on the log scale.
## If this is not desired, set scale='raw'

## Warning: Removed 24130 rows containing non-finite values (stat_density).
```



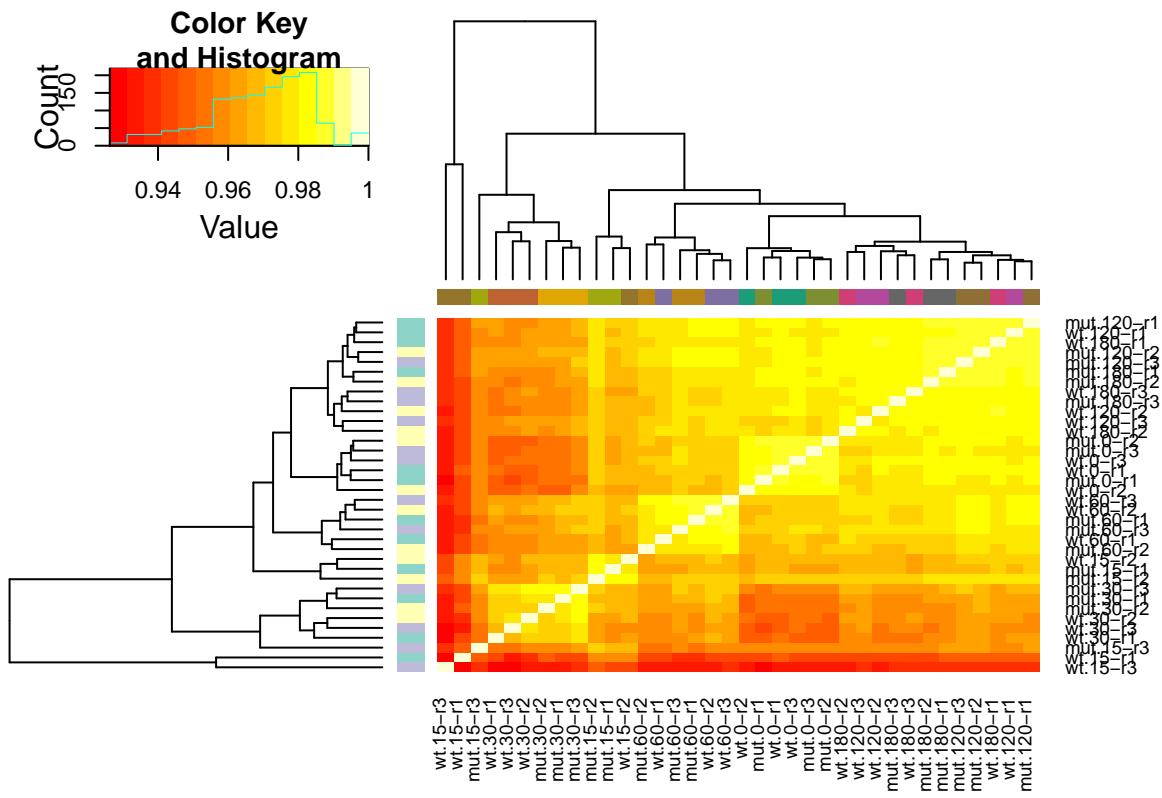
```
compare_12 <- hpgl_qq_plot(fission_expt, x=1, y=2)
compare_12$log
```



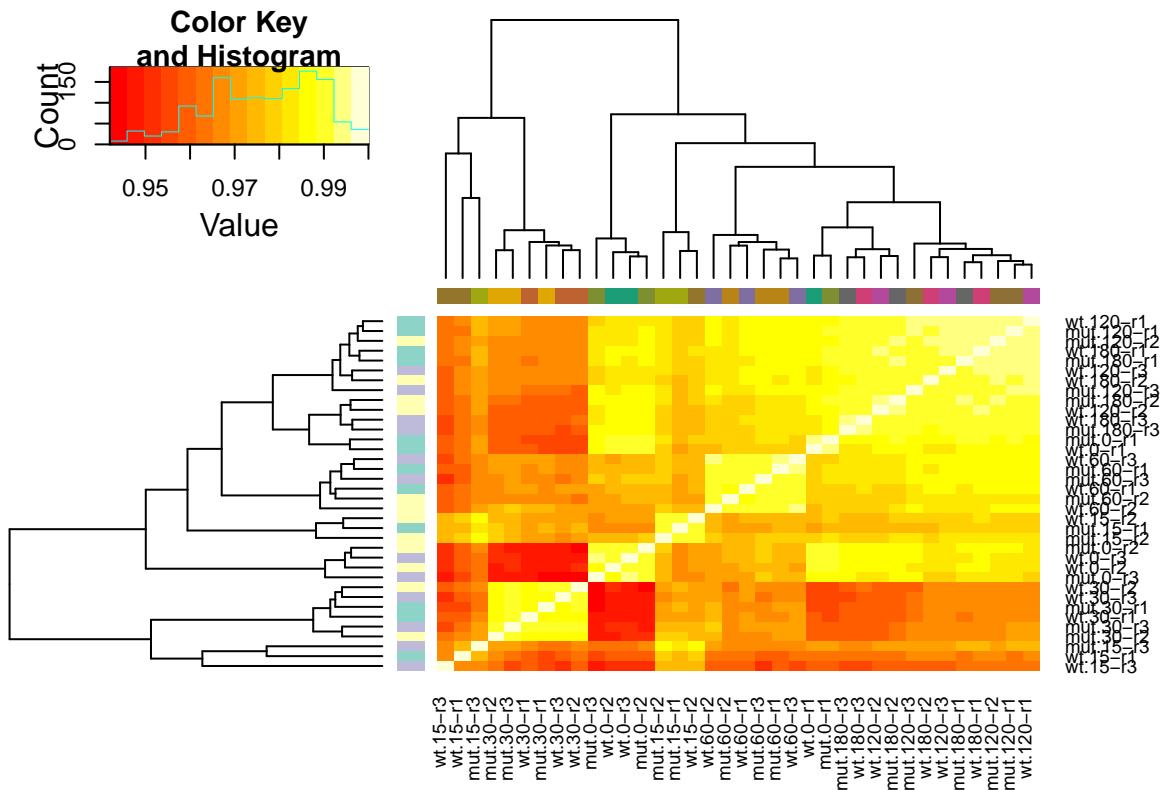
See how they cluster

Ok, so we can further check out how the data cluster with respect to one another...

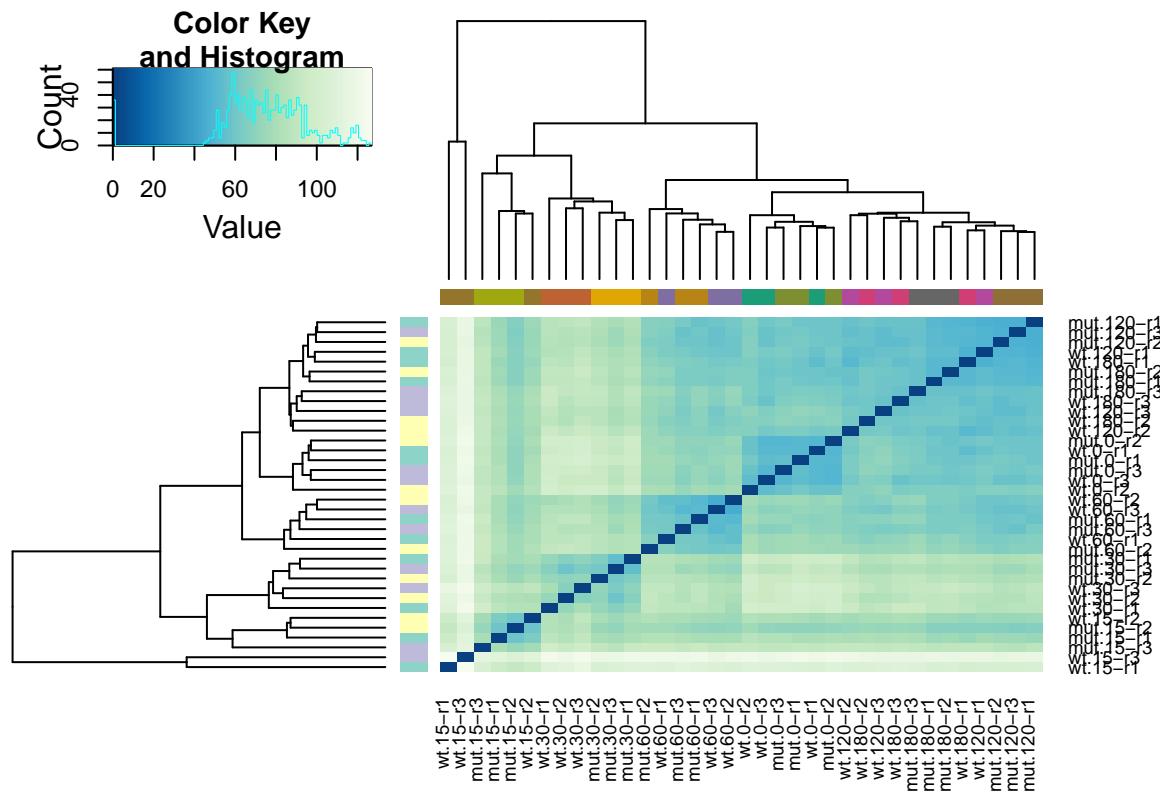
```
hpgl_corheat(norm_expt)
```



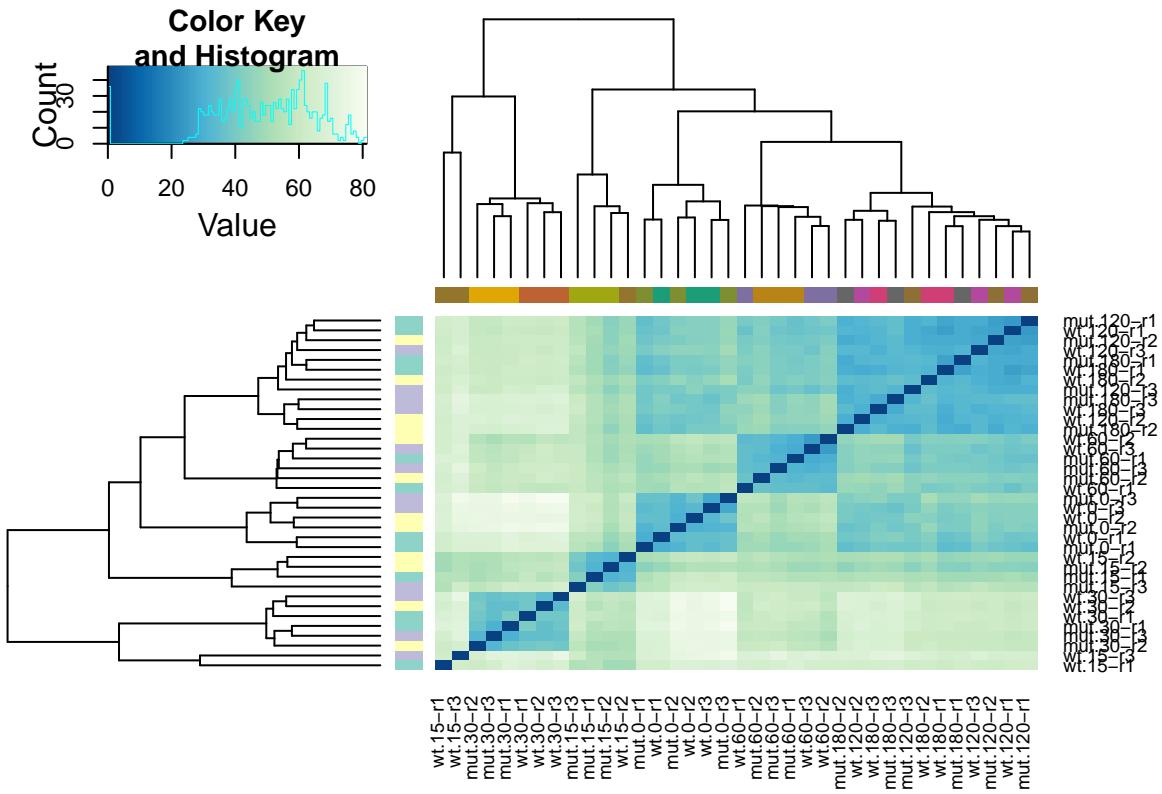
`hpgl_corheat(batchnorm_expt)`



```
hpgl_disheat(norm_expt)
```



```
hpgl_disheat(batchnorm_expt)
```



# Some simple differential expression analyses

```
limma_comparison <- limma_pairwise(normbatch_expt)

## Starting limma pairwise comparison.

## libsize was not specified, this parameter has profound effects on limma's result.

## Using the libsize from expt$best_libsize.

## Limma step 1/6: choosing model.

## Limma step 2/6: running voom

## limma step 3/6: running lmFit

## Limma step 4/6: making and fitting contrasts.

## As a reference, the identity is: mut.0 = mut.0,

## As a reference, the identity is: mut.120 = mut.120,

## As a reference, the identity is: mut.15 = mut.15,
```

```

## As a reference, the identity is: mut.180 = mut.180,
## As a reference, the identity is: mut.30 = mut.30,
## As a reference, the identity is: mut.60 = mut.60,
## As a reference, the identity is: wt.0 = wt.0,
## As a reference, the identity is: wt.120 = wt.120,
## As a reference, the identity is: wt.15 = wt.15,
## As a reference, the identity is: wt.180 = wt.180,
## As a reference, the identity is: wt.30 = wt.30,
## As a reference, the identity is: wt.60 = wt.60,
## Limma step 5/6: Running eBayes and topTable.

## Limma step 6/6: Writing limma outputs.

## limma step 6/6: 1/78: Printing table: mut.0.

## Warning: replacing previous import by 'grid::arrow' when loading 'qvalue'

## Warning: replacing previous import by 'grid::unit' when loading 'qvalue'

## limma step 6/6: 2/78: Printing table: mut.120.

## limma step 6/6: 3/78: Printing table: mut.15.

## limma step 6/6: 4/78: Printing table: mut.180.

## limma step 6/6: 5/78: Printing table: mut.30.

## limma step 6/6: 6/78: Printing table: mut.60.

## limma step 6/6: 7/78: Printing table: wt.0.

## limma step 6/6: 8/78: Printing table: wt.120.

## limma step 6/6: 9/78: Printing table: wt.15.

## limma step 6/6: 10/78: Printing table: wt.180.

## limma step 6/6: 11/78: Printing table: wt.30.

```

```
## limma step 6/6: 12/78: Printing table: wt.60.

## limma step 6/6: 13/78: Printing table: mut.120_vs_mut.0.

## limma step 6/6: 14/78: Printing table: mut.15_vs_mut.0.

## limma step 6/6: 15/78: Printing table: mut.180_vs_mut.0.

## limma step 6/6: 16/78: Printing table: mut.30_vs_mut.0.

## limma step 6/6: 17/78: Printing table: mut.60_vs_mut.0.

## limma step 6/6: 18/78: Printing table: wt.0_vs_mut.0.

## limma step 6/6: 19/78: Printing table: wt.120_vs_mut.0.

## limma step 6/6: 20/78: Printing table: wt.15_vs_mut.0.

## limma step 6/6: 21/78: Printing table: wt.180_vs_mut.0.

## limma step 6/6: 22/78: Printing table: wt.30_vs_mut.0.

## limma step 6/6: 23/78: Printing table: wt.60_vs_mut.0.

## limma step 6/6: 24/78: Printing table: mut.15_vs_mut.120.

## limma step 6/6: 25/78: Printing table: mut.180_vs_mut.120.

## limma step 6/6: 26/78: Printing table: mut.30_vs_mut.120.

## limma step 6/6: 27/78: Printing table: mut.60_vs_mut.120.

## limma step 6/6: 28/78: Printing table: wt.0_vs_mut.120.

## limma step 6/6: 29/78: Printing table: wt.120_vs_mut.120.

## limma step 6/6: 30/78: Printing table: wt.15_vs_mut.120.

## limma step 6/6: 31/78: Printing table: wt.180_vs_mut.120.

## limma step 6/6: 32/78: Printing table: wt.30_vs_mut.120.

## limma step 6/6: 33/78: Printing table: wt.60_vs_mut.120.

## limma step 6/6: 34/78: Printing table: mut.180_vs_mut.15.

## limma step 6/6: 35/78: Printing table: mut.30_vs_mut.15.
```

```
## limma step 6/6: 36/78: Printing table: mut.60_vs_mut.15.

## limma step 6/6: 37/78: Printing table: wt.0_vs_mut.15.

## limma step 6/6: 38/78: Printing table: wt.120_vs_mut.15.

## limma step 6/6: 39/78: Printing table: wt.15_vs_mut.15.

## limma step 6/6: 40/78: Printing table: wt.180_vs_mut.15.

## limma step 6/6: 41/78: Printing table: wt.30_vs_mut.15.

## limma step 6/6: 42/78: Printing table: wt.60_vs_mut.15.

## limma step 6/6: 43/78: Printing table: mut.30_vs_mut.180.

## limma step 6/6: 44/78: Printing table: mut.60_vs_mut.180.

## limma step 6/6: 45/78: Printing table: wt.0_vs_mut.180.

## limma step 6/6: 46/78: Printing table: wt.120_vs_mut.180.

## limma step 6/6: 47/78: Printing table: wt.15_vs_mut.180.

## limma step 6/6: 48/78: Printing table: wt.180_vs_mut.180.

## limma step 6/6: 49/78: Printing table: wt.30_vs_mut.180.

## limma step 6/6: 50/78: Printing table: wt.60_vs_mut.180.

## limma step 6/6: 51/78: Printing table: mut.60_vs_mut.30.

## limma step 6/6: 52/78: Printing table: wt.0_vs_mut.30.

## limma step 6/6: 53/78: Printing table: wt.120_vs_mut.30.

## limma step 6/6: 54/78: Printing table: wt.15_vs_mut.30.

## limma step 6/6: 55/78: Printing table: wt.180_vs_mut.30.

## limma step 6/6: 56/78: Printing table: wt.30_vs_mut.30.

## limma step 6/6: 57/78: Printing table: wt.60_vs_mut.30.

## limma step 6/6: 58/78: Printing table: wt.0_vs_mut.60.

## limma step 6/6: 59/78: Printing table: wt.120_vs_mut.60.
```

```

## limma step 6/6: 60/78: Printing table: wt.15_vs_mut.60.

## limma step 6/6: 61/78: Printing table: wt.180_vs_mut.60.

## limma step 6/6: 62/78: Printing table: wt.30_vs_mut.60.

## limma step 6/6: 63/78: Printing table: wt.60_vs_mut.60.

## limma step 6/6: 64/78: Printing table: wt.120_vs_wt.0.

## limma step 6/6: 65/78: Printing table: wt.15_vs_wt.0.

## limma step 6/6: 66/78: Printing table: wt.180_vs_wt.0.

## limma step 6/6: 67/78: Printing table: wt.30_vs_wt.0.

## limma step 6/6: 68/78: Printing table: wt.60_vs_wt.0.

## limma step 6/6: 69/78: Printing table: wt.15_vs_wt.120.

## limma step 6/6: 70/78: Printing table: wt.180_vs_wt.120.

## limma step 6/6: 71/78: Printing table: wt.30_vs_wt.120.

## limma step 6/6: 72/78: Printing table: wt.60_vs_wt.120.

## limma step 6/6: 73/78: Printing table: wt.180_vs_wt.15.

## limma step 6/6: 74/78: Printing table: wt.30_vs_wt.15.

## limma step 6/6: 75/78: Printing table: wt.60_vs_wt.15.

## limma step 6/6: 76/78: Printing table: wt.30_vs_wt.180.

## limma step 6/6: 77/78: Printing table: wt.60_vs_wt.180.

## limma step 6/6: 78/78: Printing table: wt.60_vs_wt.30.

names(limma_comparison$all_tables)

```

```

## [1] "mut.0"           "mut.120"          "mut.15"
## [4] "mut.180"          "mut.30"            "mut.60"
## [7] "wt.0"             "wt.120"           "wt.15"
## [10] "wt.180"           "wt.30"            "wt.60"
## [13] "mut.120_vs_mut.0" "mut.15_vs_mut.0"  "mut.180_vs_mut.0"
## [16] "mut.30_vs_mut.0"  "mut.60_vs_mut.0"  "wt.0_vs_mut.0"
## [19] "wt.120_vs_mut.0"  "wt.15_vs_mut.0"  "wt.180_vs_mut.0"
## [22] "wt.30_vs_mut.0"   "wt.60_vs_mut.0"  "mut.15_vs_mut.120"
## [25] "mut.180_vs_mut.120" "mut.30_vs_mut.120" "mut.60_vs_mut.120"

```

```

## [28] "wt.0_vs_mut.120"      "wt.120_vs_mut.120"    "wt.15_vs_mut.120"
## [31] "wt.180_vs_mut.120"    "wt.30_vs_mut.120"    "wt.60_vs_mut.120"
## [34] "mut.180_vs_mut.15"    "mut.30_vs_mut.15"    "mut.60_vs_mut.15"
## [37] "wt.0_vs_mut.15"       "wt.120_vs_mut.15"    "wt.15_vs_mut.15"
## [40] "wt.180_vs_mut.15"    "wt.30_vs_mut.15"    "wt.60_vs_mut.15"
## [43] "mut.30_vs_mut.180"    "mut.60_vs_mut.180"   "wt.0_vs_mut.180"
## [46] "wt.120_vs_mut.180"    "wt.15_vs_mut.180"    "wt.180_vs_mut.180"
## [49] "wt.30_vs_mut.180"    "wt.60_vs_mut.180"    "mut.60_vs_mut.30"
## [52] "wt.0_vs_mut.30"       "wt.120_vs_mut.30"    "wt.15_vs_mut.30"
## [55] "wt.180_vs_mut.30"    "wt.30_vs_mut.30"    "wt.60_vs_mut.30"
## [58] "wt.0_vs_mut.60"       "wt.120_vs_mut.60"    "wt.15_vs_mut.60"
## [61] "wt.180_vs_mut.60"    "wt.30_vs_mut.60"    "wt.60_vs_mut.60"
## [64] "wt.120_vs_wt.0"       "wt.15_vs_wt.0"      "wt.180_vs_wt.0"
## [67] "wt.30_vs_wt.0"       "wt.60_vs_wt.0"      "wt.15_vs_wt.120"
## [70] "wt.180_vs_wt.120"    "wt.30_vs_wt.120"    "wt.60_vs_wt.120"
## [73] "wt.180_vs_wt.15"    "wt.30_vs_wt.15"    "wt.60_vs_wt.15"
## [76] "wt.30_vs_wt.180"    "wt.60_vs_wt.180"    "wt.60_vs_wt.30"

```

```
summary(limma_comparison$all_tables$wt.120_vs_mut.120)
```

```

##      logFC          AveExpr          t
##  Min. :-1.9160000  Min. :-0.8891  Min. :-5.19700
##  1st Qu.:-0.0934400 1st Qu.: 1.5575  1st Qu.:-0.57450
##  Median : -0.0004676 Median : 4.1260  Median :-0.01783
##  Mean   : -0.0006456 Mean   : 3.7550  Mean   :-0.03103
##  3rd Qu.: 0.0880050 3rd Qu.: 5.5870  3rd Qu.: 0.53625
##  Max.   : 2.5740000 Max.   :18.4600  Max.   : 5.75100
##      P.Value        adj.P.Val         B      qvalue
##  Length:7039        Length:7039  Min. :-5.879  Length:7039
##  Class :character   Class :character  1st Qu.:-5.499  Class :character
##  Mode  :character   Mode  :character  Median :-5.102  Mode  :character
##                           Mean   :-5.108
##                           3rd Qu.:-4.927
##                           Max.   : 1.755

```

```

wt_120 <- limma_comparison$all_tables$wt.120
mut_120 <- limma_comparison$all_tables$mut.120
scatter_wt_mut <- limma_coefficient_scatter(limma_comparison, x="wt.120", y="mut.120", gvis_filename=NU
```

```
## This can do comparisons among the following columns in the limma result:
```

```
## mut.0mut.120mut.15mut.180mut.30mut.60wt.0wt.120wt.15wt.180wt.30wt.60mut.120_vs_mut.0mut.15_vs_mut.0m
```

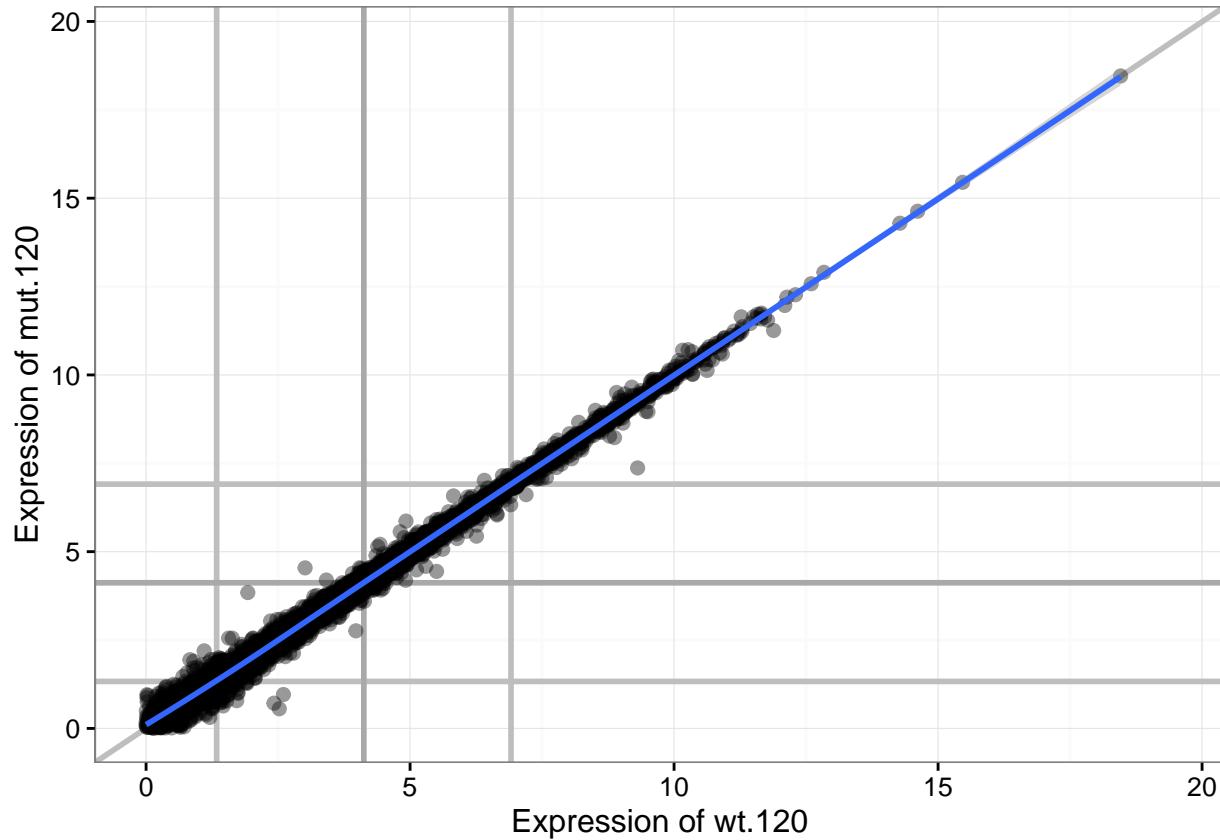
```
## Actually comparing wt.120 and mut.120.
```

```
## Setting binwidth to 0.0389150079764497 in order to have 500 bins.
```

```
scatter_wt_mut$scatter
```

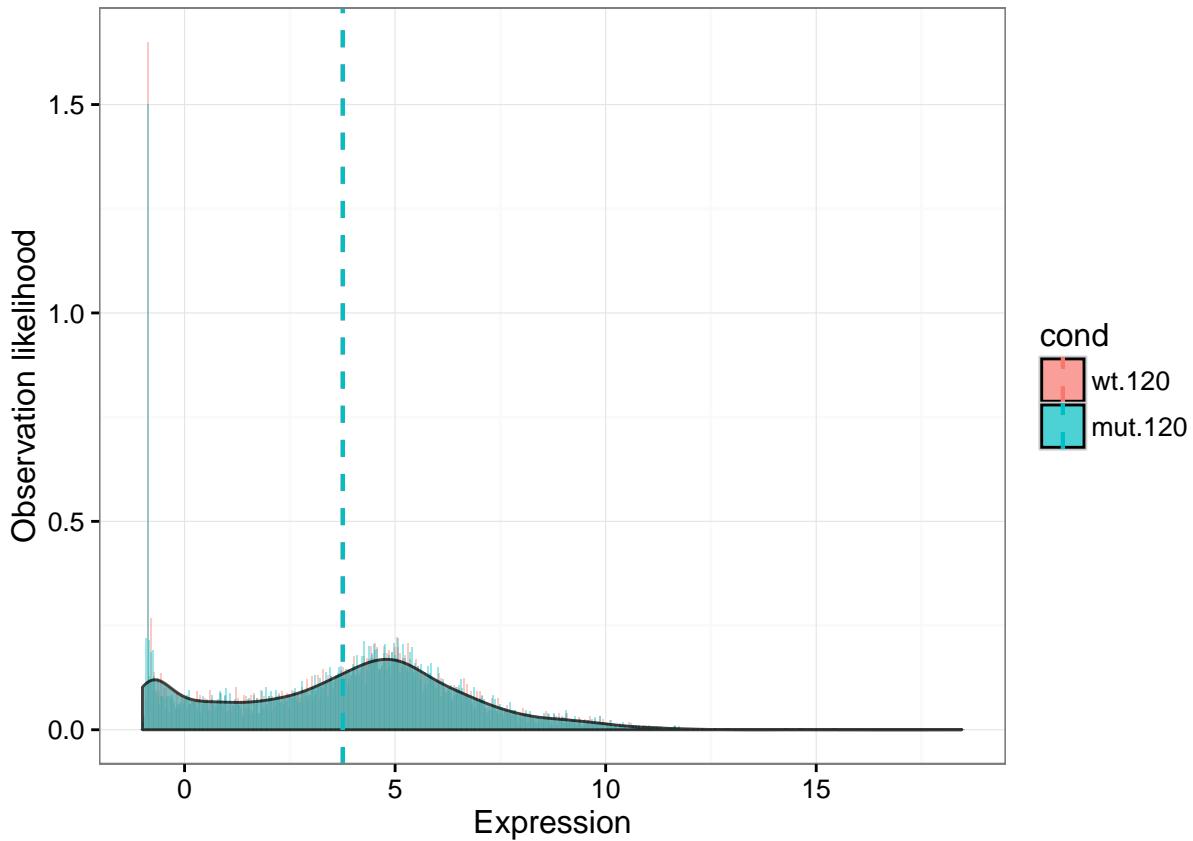
```
## Warning: Removed 1108 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1108 rows containing missing values (geom_point).
```



```
scatter_wt_mut$both_histogram
```

```
## $plot
```



```

## 
## $data_summary
##   wt.120      mut.120
##   Min.   :-0.9011   Min.   :-1.000
##   1st Qu.: 1.5503  1st Qu.: 1.552
##   Median : 4.1239  Median : 4.120
##   Mean    : 3.7575  Mean    : 3.758
##   3rd Qu.: 5.6070  3rd Qu.: 5.615
##   Max.    :18.4575  Max.    :18.458
##
## 
## $uncor_t
##
##  Pairwise comparisons using t tests with pooled SD
##
## data: play_all$expression and play_all$cond
##
##       wt.120
## mut.120 0.99
##
## P value adjustment method: none
##
## $bon_t
##
##  Pairwise comparisons using t tests with pooled SD
##
## data: play_all$expression and play_all$cond
##

```

```

##          wt.120
## mut.120 0.99
##
## P value adjustment method: bonferroni

deseq_comparison <- deseq2_pairwise(fission_expt, model_batch=TRUE)

## Starting DESeq2 pairwise comparisons.

## DESeq2 step 1/5: Including batch and condition in the deseq model.

## DESeq2 step 2/5

## DESeq2 step 3/5

## DESeq2 step 4/5

## DESeq2 step 5/5: 1/66: Printing table: mut.120_vs_mut.0

## DESeq2 step 5/5: 2/66: Printing table: mut.15_vs_mut.0

## DESeq2 step 5/5: 3/66: Printing table: mut.180_vs_mut.0

## DESeq2 step 5/5: 4/66: Printing table: mut.30_vs_mut.0

## DESeq2 step 5/5: 5/66: Printing table: mut.60_vs_mut.0

## DESeq2 step 5/5: 6/66: Printing table: wt.0_vs_mut.0

## DESeq2 step 5/5: 7/66: Printing table: wt.120_vs_mut.0

## DESeq2 step 5/5: 8/66: Printing table: wt.15_vs_mut.0

## DESeq2 step 5/5: 9/66: Printing table: wt.180_vs_mut.0

## DESeq2 step 5/5: 10/66: Printing table: wt.30_vs_mut.0

## DESeq2 step 5/5: 11/66: Printing table: wt.60_vs_mut.0

## DESeq2 step 5/5: 12/66: Printing table: mut.15_vs_mut.120

## DESeq2 step 5/5: 13/66: Printing table: mut.180_vs_mut.120

## DESeq2 step 5/5: 14/66: Printing table: mut.30_vs_mut.120

## DESeq2 step 5/5: 15/66: Printing table: mut.60_vs_mut.120

## DESeq2 step 5/5: 16/66: Printing table: wt.0_vs_mut.120

```

```
## DESeq2 step 5/5: 17/66: Printing table: wt.120_vs_mut.120

## DESeq2 step 5/5: 18/66: Printing table: wt.15_vs_mut.120

## DESeq2 step 5/5: 19/66: Printing table: wt.180_vs_mut.120

## DESeq2 step 5/5: 20/66: Printing table: wt.30_vs_mut.120

## DESeq2 step 5/5: 21/66: Printing table: wt.60_vs_mut.120

## DESeq2 step 5/5: 22/66: Printing table: mut.180_vs_mut.15

## DESeq2 step 5/5: 23/66: Printing table: mut.30_vs_mut.15

## DESeq2 step 5/5: 24/66: Printing table: mut.60_vs_mut.15

## DESeq2 step 5/5: 25/66: Printing table: wt.0_vs_mut.15

## DESeq2 step 5/5: 26/66: Printing table: wt.120_vs_mut.15

## DESeq2 step 5/5: 27/66: Printing table: wt.15_vs_mut.15

## DESeq2 step 5/5: 28/66: Printing table: wt.180_vs_mut.15

## DESeq2 step 5/5: 29/66: Printing table: wt.30_vs_mut.15

## DESeq2 step 5/5: 30/66: Printing table: wt.60_vs_mut.15

## DESeq2 step 5/5: 31/66: Printing table: mut.30_vs_mut.180

## DESeq2 step 5/5: 32/66: Printing table: mut.60_vs_mut.180

## DESeq2 step 5/5: 33/66: Printing table: wt.0_vs_mut.180

## DESeq2 step 5/5: 34/66: Printing table: wt.120_vs_mut.180

## DESeq2 step 5/5: 35/66: Printing table: wt.15_vs_mut.180

## DESeq2 step 5/5: 36/66: Printing table: wt.180_vs_mut.180

## DESeq2 step 5/5: 37/66: Printing table: wt.30_vs_mut.180

## DESeq2 step 5/5: 38/66: Printing table: wt.60_vs_mut.180

## DESeq2 step 5/5: 39/66: Printing table: mut.60_vs_mut.30

## DESeq2 step 5/5: 40/66: Printing table: wt.0_vs_mut.30
```

```
## DESeq2 step 5/5: 41/66: Printing table: wt.120_vs_mut.30  
## DESeq2 step 5/5: 42/66: Printing table: wt.15_vs_mut.30  
## DESeq2 step 5/5: 43/66: Printing table: wt.180_vs_mut.30  
## DESeq2 step 5/5: 44/66: Printing table: wt.30_vs_mut.30  
## DESeq2 step 5/5: 45/66: Printing table: wt.60_vs_mut.30  
## DESeq2 step 5/5: 46/66: Printing table: wt.0_vs_mut.60  
## DESeq2 step 5/5: 47/66: Printing table: wt.120_vs_mut.60  
## DESeq2 step 5/5: 48/66: Printing table: wt.15_vs_mut.60  
## DESeq2 step 5/5: 49/66: Printing table: wt.180_vs_mut.60  
## DESeq2 step 5/5: 50/66: Printing table: wt.30_vs_mut.60  
## DESeq2 step 5/5: 51/66: Printing table: wt.60_vs_mut.60  
## DESeq2 step 5/5: 52/66: Printing table: wt.120_vs_wt.0  
## DESeq2 step 5/5: 53/66: Printing table: wt.15_vs_wt.0  
## DESeq2 step 5/5: 54/66: Printing table: wt.180_vs_wt.0  
## DESeq2 step 5/5: 55/66: Printing table: wt.30_vs_wt.0  
## DESeq2 step 5/5: 56/66: Printing table: wt.60_vs_wt.0  
## DESeq2 step 5/5: 57/66: Printing table: wt.15_vs_wt.120  
## DESeq2 step 5/5: 58/66: Printing table: wt.180_vs_wt.120  
## DESeq2 step 5/5: 59/66: Printing table: wt.30_vs_wt.120  
## DESeq2 step 5/5: 60/66: Printing table: wt.60_vs_wt.120  
## DESeq2 step 5/5: 61/66: Printing table: wt.180_vs_wt.15  
## DESeq2 step 5/5: 62/66: Printing table: wt.30_vs_wt.15  
## DESeq2 step 5/5: 63/66: Printing table: wt.60_vs_wt.15  
## DESeq2 step 5/5: 64/66: Printing table: wt.30_vs_wt.180
```

```

## DESeq2 step 5/5: 65/66: Printing table: wt.60_vs_wt.180

## DESeq2 step 5/5: 66/66: Printing table: wt.60_vs_wt.30

## Collected coefficients for: mut.0

## Collected coefficients for: mut.120

## Collected coefficients for: mut.15

## Collected coefficients for: mut.180

## Collected coefficients for: mut.30

## Collected coefficients for: mut.60

## Collected coefficients for: wt.0

## Collected coefficients for: wt.120

## Collected coefficients for: wt.15

## Collected coefficients for: wt.180

## Collected coefficients for: wt.30

## Collected coefficients for: wt.60

```

```
summary(deseq_comparison$all_tables$wt.120_vs_mut.120)
```

```

##      baseMean        logFC        lfcSE         stat
##  Min.   :    0   Min.  :-1.11100   Min.  :0.05733   Min.  :-4.14700
##  1st Qu.:   31   1st Qu.:-0.10090   1st Qu.:0.12760   1st Qu.:-0.55910
##  Median :  208   Median :-0.00094   Median :0.17565   Median :-0.00721
##  Mean   : 1744   Mean   :-0.00503   Mean   :0.20351   Mean   :-0.00765
##  3rd Qu.:  585   3rd Qu.: 0.09756   3rd Qu.:0.27132   3rd Qu.: 0.54302
##  Max.   :4586000  Max.   : 1.71500   Max.   :0.39700   Max.   : 5.13700
##                  NA's   :279       NA's   :279       NA's   :279
##      P.Value        adj.P.Val       qvalue
##  Length:7039       Length:7039       Length:7039
##  Class :character  Class :character  Class :character
##  Mode  :character  Mode  :character  Mode  :character
##
##
```

```
edger_comparison <- edger_pairwise(fission_expt, model_batch=TRUE)

## Starting edgeR pairwise comparisons.

## EdgeR step 1/9: normalizing data.

## EdgeR step 2/9: Estimating the common dispersion.

## EdgeR step 3/9: Estimating dispersion across genes.

## EdgeR step 4/9: Estimating GLM Common dispersion.

## EdgeR step 5/9: Estimating GLM Trended dispersion.

## EdgeR step 6/9: Estimating GLM Tagged dispersion.

## EdgeR step 7/9: Running glmFit.

## EdgeR step 8/9: Making pairwise contrasts.

## As a reference, the identity is: mut.0 = mut.0,

## As a reference, the identity is: mut.120 = mut.120,

## As a reference, the identity is: mut.15 = mut.15,

## As a reference, the identity is: mut.180 = mut.180,

## As a reference, the identity is: mut.30 = mut.30,

## As a reference, the identity is: mut.60 = mut.60,

## As a reference, the identity is: wt.0 = wt.0,

## As a reference, the identity is: wt.120 = wt.120,

## As a reference, the identity is: wt.15 = wt.15,

## As a reference, the identity is: wt.180 = wt.180,

## As a reference, the identity is: wt.30 = wt.30,

## As a reference, the identity is: wt.60 = wt.60,

## EdgeR step 9/9: 1/66: Printing table: mut.120_vs_mut.0.

## EdgeR step 9/9: 2/66: Printing table: mut.15_vs_mut.0.
```

```
## EdgeR step 9/9: 3/66: Printing table: mut.180_vs_mut.0.

## EdgeR step 9/9: 4/66: Printing table: mut.30_vs_mut.0.

## EdgeR step 9/9: 5/66: Printing table: mut.60_vs_mut.0.

## EdgeR step 9/9: 6/66: Printing table: wt.0_vs_mut.0.

## EdgeR step 9/9: 7/66: Printing table: wt.120_vs_mut.0.

## EdgeR step 9/9: 8/66: Printing table: wt.15_vs_mut.0.

## EdgeR step 9/9: 9/66: Printing table: wt.180_vs_mut.0.

## EdgeR step 9/9: 10/66: Printing table: wt.30_vs_mut.0.

## EdgeR step 9/9: 11/66: Printing table: wt.60_vs_mut.0.

## EdgeR step 9/9: 12/66: Printing table: mut.15_vs_mut.120.

## EdgeR step 9/9: 13/66: Printing table: mut.180_vs_mut.120.

## EdgeR step 9/9: 14/66: Printing table: mut.30_vs_mut.120.

## EdgeR step 9/9: 15/66: Printing table: mut.60_vs_mut.120.

## EdgeR step 9/9: 16/66: Printing table: wt.0_vs_mut.120.

## EdgeR step 9/9: 17/66: Printing table: wt.120_vs_mut.120.

## EdgeR step 9/9: 18/66: Printing table: wt.15_vs_mut.120.

## EdgeR step 9/9: 19/66: Printing table: wt.180_vs_mut.120.

## EdgeR step 9/9: 20/66: Printing table: wt.30_vs_mut.120.

## EdgeR step 9/9: 21/66: Printing table: wt.60_vs_mut.120.

## EdgeR step 9/9: 22/66: Printing table: mut.180_vs_mut.15.

## EdgeR step 9/9: 23/66: Printing table: mut.30_vs_mut.15.

## EdgeR step 9/9: 24/66: Printing table: mut.60_vs_mut.15.

## EdgeR step 9/9: 25/66: Printing table: wt.0_vs_mut.15.

## EdgeR step 9/9: 26/66: Printing table: wt.120_vs_mut.15.
```

```
## EdgeR step 9/9: 27/66: Printing table: wt.15_vs_mut.15.

## EdgeR step 9/9: 28/66: Printing table: wt.180_vs_mut.15.

## EdgeR step 9/9: 29/66: Printing table: wt.30_vs_mut.15.

## EdgeR step 9/9: 30/66: Printing table: wt.60_vs_mut.15.

## EdgeR step 9/9: 31/66: Printing table: mut.30_vs_mut.180.

## EdgeR step 9/9: 32/66: Printing table: mut.60_vs_mut.180.

## EdgeR step 9/9: 33/66: Printing table: wt.0_vs_mut.180.

## EdgeR step 9/9: 34/66: Printing table: wt.120_vs_mut.180.

## EdgeR step 9/9: 35/66: Printing table: wt.15_vs_mut.180.

## EdgeR step 9/9: 36/66: Printing table: wt.180_vs_mut.180.

## EdgeR step 9/9: 37/66: Printing table: wt.30_vs_mut.180.

## EdgeR step 9/9: 38/66: Printing table: wt.60_vs_mut.180.

## EdgeR step 9/9: 39/66: Printing table: mut.60_vs_mut.30.

## EdgeR step 9/9: 40/66: Printing table: wt.0_vs_mut.30.

## EdgeR step 9/9: 41/66: Printing table: wt.120_vs_mut.30.

## EdgeR step 9/9: 42/66: Printing table: wt.15_vs_mut.30.

## EdgeR step 9/9: 43/66: Printing table: wt.180_vs_mut.30.

## EdgeR step 9/9: 44/66: Printing table: wt.30_vs_mut.30.

## EdgeR step 9/9: 45/66: Printing table: wt.60_vs_mut.30.

## EdgeR step 9/9: 46/66: Printing table: wt.0_vs_mut.60.

## EdgeR step 9/9: 47/66: Printing table: wt.120_vs_mut.60.

## EdgeR step 9/9: 48/66: Printing table: wt.15_vs_mut.60.

## EdgeR step 9/9: 49/66: Printing table: wt.180_vs_mut.60.

## EdgeR step 9/9: 50/66: Printing table: wt.30_vs_mut.60.
```

```

## EdgeR step 9/9: 51/66: Printing table: wt.60_vs_mut.60.

## EdgeR step 9/9: 52/66: Printing table: wt.120_vs_wt.0.

## EdgeR step 9/9: 53/66: Printing table: wt.15_vs_wt.0.

## EdgeR step 9/9: 54/66: Printing table: wt.180_vs_wt.0.

## EdgeR step 9/9: 55/66: Printing table: wt.30_vs_wt.0.

## EdgeR step 9/9: 56/66: Printing table: wt.60_vs_wt.0.

## EdgeR step 9/9: 57/66: Printing table: wt.15_vs_wt.120.

## EdgeR step 9/9: 58/66: Printing table: wt.180_vs_wt.120.

## EdgeR step 9/9: 59/66: Printing table: wt.30_vs_wt.120.

## EdgeR step 9/9: 60/66: Printing table: wt.60_vs_wt.120.

## EdgeR step 9/9: 61/66: Printing table: wt.180_vs_wt.15.

## EdgeR step 9/9: 62/66: Printing table: wt.30_vs_wt.15.

## EdgeR step 9/9: 63/66: Printing table: wt.60_vs_wt.15.

## EdgeR step 9/9: 64/66: Printing table: wt.30_vs_wt.180.

## EdgeR step 9/9: 65/66: Printing table: wt.60_vs_wt.180.

## EdgeR step 9/9: 66/66: Printing table: wt.60_vs_wt.30.

summary(edger_comparison$all_tables$wt.120_vs_mut.120)

```

```

##      logFC            logCPM           LR
##  Min.   :-4.72600   Min.   :-2.682   Min.   :-0.000002
##  1st Qu.:-0.11565   1st Qu.: 1.462   1st Qu.: 0.046550
##  Median : 0.00000   Median : 4.120   Median : 0.281700
##  Mean   :-0.02486   Mean    : 3.526   Mean    : 0.739098
##  3rd Qu.: 0.10640   3rd Qu.: 5.598   3rd Qu.: 0.947000
##  Max.   : 4.98900   Max.   :18.530   Max.   :25.880000
##      PValue          FDR            qvalue
##  Length:7039        Length:7039        Length:7039
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
```

```

basic_comparison <- basic_pairwise(fission_expt)

## Starting basic pairwise comparison.

## Basic step 1/3: Creating median and variance tables.

## Basic step 2/3: Performing comparisons.

## Basic step 2/3: 1/66: Performing log2 subtraction: mut.120_vs_mut.0

## Basic step 2/3: 2/66: Performing log2 subtraction: mut.15_vs_mut.0

## Basic step 2/3: 3/66: Performing log2 subtraction: mut.180_vs_mut.0

## Basic step 2/3: 4/66: Performing log2 subtraction: mut.30_vs_mut.0

## Basic step 2/3: 5/66: Performing log2 subtraction: mut.60_vs_mut.0

## Basic step 2/3: 6/66: Performing log2 subtraction: wt.0_vs_mut.0

## Basic step 2/3: 7/66: Performing log2 subtraction: wt.120_vs_mut.0

## Basic step 2/3: 8/66: Performing log2 subtraction: wt.15_vs_mut.0

## Basic step 2/3: 9/66: Performing log2 subtraction: wt.180_vs_mut.0

## Basic step 2/3: 10/66: Performing log2 subtraction: wt.30_vs_mut.0

## Basic step 2/3: 11/66: Performing log2 subtraction: wt.60_vs_mut.0

## Basic step 2/3: 12/66: Performing log2 subtraction: mut.15_vs_mut.120

## Basic step 2/3: 13/66: Performing log2 subtraction: mut.180_vs_mut.120

## Basic step 2/3: 14/66: Performing log2 subtraction: mut.30_vs_mut.120

## Basic step 2/3: 15/66: Performing log2 subtraction: mut.60_vs_mut.120

## Basic step 2/3: 16/66: Performing log2 subtraction: wt.0_vs_mut.120

## Basic step 2/3: 17/66: Performing log2 subtraction: wt.120_vs_mut.120

## Basic step 2/3: 18/66: Performing log2 subtraction: wt.15_vs_mut.120

## Basic step 2/3: 19/66: Performing log2 subtraction: wt.180_vs_mut.120

## Basic step 2/3: 20/66: Performing log2 subtraction: wt.30_vs_mut.120

```

```
## Basic step 2/3: 21/66: Performing log2 subtraction: wt.60_vs_mut.120

## Basic step 2/3: 22/66: Performing log2 subtraction: mut.180_vs_mut.15

## Basic step 2/3: 23/66: Performing log2 subtraction: mut.30_vs_mut.15

## Basic step 2/3: 24/66: Performing log2 subtraction: mut.60_vs_mut.15

## Basic step 2/3: 25/66: Performing log2 subtraction: wt.0_vs_mut.15

## Basic step 2/3: 26/66: Performing log2 subtraction: wt.120_vs_mut.15

## Basic step 2/3: 27/66: Performing log2 subtraction: wt.15_vs_mut.15

## Basic step 2/3: 28/66: Performing log2 subtraction: wt.180_vs_mut.15

## Basic step 2/3: 29/66: Performing log2 subtraction: wt.30_vs_mut.15

## Basic step 2/3: 30/66: Performing log2 subtraction: wt.60_vs_mut.15

## Basic step 2/3: 31/66: Performing log2 subtraction: mut.30_vs_mut.180

## Basic step 2/3: 32/66: Performing log2 subtraction: mut.60_vs_mut.180

## Basic step 2/3: 33/66: Performing log2 subtraction: wt.0_vs_mut.180

## Basic step 2/3: 34/66: Performing log2 subtraction: wt.120_vs_mut.180

## Basic step 2/3: 35/66: Performing log2 subtraction: wt.15_vs_mut.180

## Basic step 2/3: 36/66: Performing log2 subtraction: wt.180_vs_mut.180

## Basic step 2/3: 37/66: Performing log2 subtraction: wt.30_vs_mut.180

## Basic step 2/3: 38/66: Performing log2 subtraction: wt.60_vs_mut.180

## Basic step 2/3: 39/66: Performing log2 subtraction: mut.60_vs_mut.30

## Basic step 2/3: 40/66: Performing log2 subtraction: wt.0_vs_mut.30

## Basic step 2/3: 41/66: Performing log2 subtraction: wt.120_vs_mut.30

## Basic step 2/3: 42/66: Performing log2 subtraction: wt.15_vs_mut.30

## Basic step 2/3: 43/66: Performing log2 subtraction: wt.180_vs_mut.30

## Basic step 2/3: 44/66: Performing log2 subtraction: wt.30_vs_mut.30
```

```

## Basic step 2/3: 45/66: Performing log2 subtraction: wt.60_vs_mut.30

## Basic step 2/3: 46/66: Performing log2 subtraction: wt.0_vs_mut.60

## Basic step 2/3: 47/66: Performing log2 subtraction: wt.120_vs_mut.60

## Basic step 2/3: 48/66: Performing log2 subtraction: wt.15_vs_mut.60

## Basic step 2/3: 49/66: Performing log2 subtraction: wt.180_vs_mut.60

## Basic step 2/3: 50/66: Performing log2 subtraction: wt.30_vs_mut.60

## Basic step 2/3: 51/66: Performing log2 subtraction: wt.60_vs_mut.60

## Basic step 2/3: 52/66: Performing log2 subtraction: wt.120_vs_wt.0

## Basic step 2/3: 53/66: Performing log2 subtraction: wt.15_vs_wt.0

## Basic step 2/3: 54/66: Performing log2 subtraction: wt.180_vs_wt.0

## Basic step 2/3: 55/66: Performing log2 subtraction: wt.30_vs_wt.0

## Basic step 2/3: 56/66: Performing log2 subtraction: wt.60_vs_wt.0

## Basic step 2/3: 57/66: Performing log2 subtraction: wt.15_vs_wt.120

## Basic step 2/3: 58/66: Performing log2 subtraction: wt.180_vs_wt.120

## Basic step 2/3: 59/66: Performing log2 subtraction: wt.30_vs_wt.120

## Basic step 2/3: 60/66: Performing log2 subtraction: wt.60_vs_wt.120

## Basic step 2/3: 61/66: Performing log2 subtraction: wt.180_vs_wt.15

## Basic step 2/3: 62/66: Performing log2 subtraction: wt.30_vs_wt.15

## Basic step 2/3: 63/66: Performing log2 subtraction: wt.60_vs_wt.15

## Basic step 2/3: 64/66: Performing log2 subtraction: wt.30_vs_wt.180

## Basic step 2/3: 65/66: Performing log2 subtraction: wt.60_vs_wt.180

## Basic step 2/3: 66/66: Performing log2 subtraction: wt.60_vs_wt.30

## Basic step 3/3: Creating faux DE Tables.

## Basic: Returning tables.

```

```

summary(basic_comparison$all_tables$wt.120_vs_mut.120)

##   numerator_median   denominator_median   numerator_var
##   Min.      :    0.0   Min.      :    0.04   Length:7039
##   1st Qu.:    2.4   1st Qu.:    2.85   Class :character
##   Median  :   17.6   Median  :   19.43   Mode  :character
##   Mean    :  141.5   Mean    :  137.59
##   3rd Qu.:   49.2   3rd Qu.:   53.69
##   Max.    :350300.0   Max.    :296700.00
##   denominator_var          t          p
##   Length:7039           Min.    :-7.7150   Length:7039
##   Class :character       1st Qu.: 0.1175   Class :character
##   Mode  :character       Median : 0.8897   Mode  :character
##                           Mean    : 0.8374
##                           3rd Qu.: 1.6050
##                           Max.    :14.6000
##   logFC
##   Min.    :-3.63900
##   1st Qu.:-0.28705
##   Median :-0.12100
##   Mean   :-0.13283
##   3rd Qu.: 0.05047
##   Max.    : 4.51700

```

```
all_comparisons <- all_pairwise(fission_expt, model_batch=TRUE)
```

```

## Starting limma pairwise comparison.

## libsize was not specified, this parameter has profound effects on limma's result.

## Using the libsize from expt$normalized$normalized_counts.

## Limma step 1/6: choosing model.

## Limma step 2/6: running voom

## The voom input was not cpm, converting now.

## The voom input was not log2, transforming now.

## limma step 3/6: running lmFit

## Limma step 4/6: making and fitting contrasts.

## As a reference, the identity is: mut.0 = mut.0,
## As a reference, the identity is: mut.120 = mut.120,
## As a reference, the identity is: mut.15 = mut.15,

```

```
## As a reference, the identity is: mut.180 = mut.180,  
  
## As a reference, the identity is: mut.30 = mut.30,  
  
## As a reference, the identity is: mut.60 = mut.60,  
  
## As a reference, the identity is: wt.0 = wt.0,  
  
## As a reference, the identity is: wt.120 = wt.120,  
  
## As a reference, the identity is: wt.15 = wt.15,  
  
## As a reference, the identity is: wt.180 = wt.180,  
  
## As a reference, the identity is: wt.30 = wt.30,  
  
## As a reference, the identity is: wt.60 = wt.60,  
  
## Limma step 5/6: Running eBayes and topTable.  
  
## Limma step 6/6: Writing limma outputs.  
  
## limma step 6/6: 1/78: Printing table: mut.0.  
  
## limma step 6/6: 2/78: Printing table: mut.120.  
  
## limma step 6/6: 3/78: Printing table: mut.15.  
  
## limma step 6/6: 4/78: Printing table: mut.180.  
  
## limma step 6/6: 5/78: Printing table: mut.30.  
  
## limma step 6/6: 6/78: Printing table: mut.60.  
  
## limma step 6/6: 7/78: Printing table: wt.0.  
  
## limma step 6/6: 8/78: Printing table: wt.120.  
  
## limma step 6/6: 9/78: Printing table: wt.15.  
  
## limma step 6/6: 10/78: Printing table: wt.180.  
  
## limma step 6/6: 11/78: Printing table: wt.30.  
  
## limma step 6/6: 12/78: Printing table: wt.60.  
  
## limma step 6/6: 13/78: Printing table: mut.120_vs_mut.0.
```

```
## limma step 6/6: 14/78: Printing table: mut.15_vs_mut.0.

## limma step 6/6: 15/78: Printing table: mut.180_vs_mut.0.

## limma step 6/6: 16/78: Printing table: mut.30_vs_mut.0.

## limma step 6/6: 17/78: Printing table: mut.60_vs_mut.0.

## limma step 6/6: 18/78: Printing table: wt.0_vs_mut.0.

## limma step 6/6: 19/78: Printing table: wt.120_vs_mut.0.

## limma step 6/6: 20/78: Printing table: wt.15_vs_mut.0.

## limma step 6/6: 21/78: Printing table: wt.180_vs_mut.0.

## limma step 6/6: 22/78: Printing table: wt.30_vs_mut.0.

## limma step 6/6: 23/78: Printing table: wt.60_vs_mut.0.

## limma step 6/6: 24/78: Printing table: mut.15_vs_mut.120.

## limma step 6/6: 25/78: Printing table: mut.180_vs_mut.120.

## limma step 6/6: 26/78: Printing table: mut.30_vs_mut.120.

## limma step 6/6: 27/78: Printing table: mut.60_vs_mut.120.

## limma step 6/6: 28/78: Printing table: wt.0_vs_mut.120.

## limma step 6/6: 29/78: Printing table: wt.120_vs_mut.120.

## limma step 6/6: 30/78: Printing table: wt.15_vs_mut.120.

## limma step 6/6: 31/78: Printing table: wt.180_vs_mut.120.

## limma step 6/6: 32/78: Printing table: wt.30_vs_mut.120.

## limma step 6/6: 33/78: Printing table: wt.60_vs_mut.120.

## limma step 6/6: 34/78: Printing table: mut.180_vs_mut.15.

## limma step 6/6: 35/78: Printing table: mut.30_vs_mut.15.

## limma step 6/6: 36/78: Printing table: mut.60_vs_mut.15.

## limma step 6/6: 37/78: Printing table: wt.0_vs_mut.15.
```

```
## limma step 6/6: 38/78: Printing table: wt.120_vs_mut.15.

## limma step 6/6: 39/78: Printing table: wt.15_vs_mut.15.

## limma step 6/6: 40/78: Printing table: wt.180_vs_mut.15.

## limma step 6/6: 41/78: Printing table: wt.30_vs_mut.15.

## limma step 6/6: 42/78: Printing table: wt.60_vs_mut.15.

## limma step 6/6: 43/78: Printing table: mut.30_vs_mut.180.

## limma step 6/6: 44/78: Printing table: mut.60_vs_mut.180.

## limma step 6/6: 45/78: Printing table: wt.0_vs_mut.180.

## limma step 6/6: 46/78: Printing table: wt.120_vs_mut.180.

## limma step 6/6: 47/78: Printing table: wt.15_vs_mut.180.

## limma step 6/6: 48/78: Printing table: wt.180_vs_mut.180.

## limma step 6/6: 49/78: Printing table: wt.30_vs_mut.180.

## limma step 6/6: 50/78: Printing table: wt.60_vs_mut.180.

## limma step 6/6: 51/78: Printing table: mut.60_vs_mut.30.

## limma step 6/6: 52/78: Printing table: wt.0_vs_mut.30.

## limma step 6/6: 53/78: Printing table: wt.120_vs_mut.30.

## limma step 6/6: 54/78: Printing table: wt.15_vs_mut.30.

## limma step 6/6: 55/78: Printing table: wt.180_vs_mut.30.

## limma step 6/6: 56/78: Printing table: wt.30_vs_mut.30.

## limma step 6/6: 57/78: Printing table: wt.60_vs_mut.30.

## limma step 6/6: 58/78: Printing table: wt.0_vs_mut.60.

## limma step 6/6: 59/78: Printing table: wt.120_vs_mut.60.

## limma step 6/6: 60/78: Printing table: wt.15_vs_mut.60.

## limma step 6/6: 61/78: Printing table: wt.180_vs_mut.60.
```

```

## limma step 6/6: 62/78: Printing table: wt.30_vs_mut.60.

## limma step 6/6: 63/78: Printing table: wt.60_vs_mut.60.

## limma step 6/6: 64/78: Printing table: wt.120_vs_wt.0.

## limma step 6/6: 65/78: Printing table: wt.15_vs_wt.0.

## limma step 6/6: 66/78: Printing table: wt.180_vs_wt.0.

## limma step 6/6: 67/78: Printing table: wt.30_vs_wt.0.

## limma step 6/6: 68/78: Printing table: wt.60_vs_wt.0.

## limma step 6/6: 69/78: Printing table: wt.15_vs_wt.120.

## limma step 6/6: 70/78: Printing table: wt.180_vs_wt.120.

## limma step 6/6: 71/78: Printing table: wt.30_vs_wt.120.

## limma step 6/6: 72/78: Printing table: wt.60_vs_wt.120.

## limma step 6/6: 73/78: Printing table: wt.180_vs_wt.15.

## limma step 6/6: 74/78: Printing table: wt.30_vs_wt.15.

## limma step 6/6: 75/78: Printing table: wt.60_vs_wt.15.

## limma step 6/6: 76/78: Printing table: wt.30_vs_wt.180.

## limma step 6/6: 77/78: Printing table: wt.60_vs_wt.180.

## limma step 6/6: 78/78: Printing table: wt.60_vs_wt.30.

## Starting DESeq2 pairwise comparisons.

## DESeq2 step 1/5: Including batch and condition in the deseq model.

## DESeq2 step 2/5

## DESeq2 step 3/5

## DESeq2 step 4/5

## DESeq2 step 5/5: 1/66: Printing table: mut.120_vs_mut.0

## DESeq2 step 5/5: 2/66: Printing table: mut.15_vs_mut.0

```

```
## DESeq2 step 5/5: 3/66: Printing table: mut.180_vs_mut.0  
## DESeq2 step 5/5: 4/66: Printing table: mut.30_vs_mut.0  
## DESeq2 step 5/5: 5/66: Printing table: mut.60_vs_mut.0  
## DESeq2 step 5/5: 6/66: Printing table: wt.0_vs_mut.0  
## DESeq2 step 5/5: 7/66: Printing table: wt.120_vs_mut.0  
## DESeq2 step 5/5: 8/66: Printing table: wt.15_vs_mut.0  
## DESeq2 step 5/5: 9/66: Printing table: wt.180_vs_mut.0  
## DESeq2 step 5/5: 10/66: Printing table: wt.30_vs_mut.0  
## DESeq2 step 5/5: 11/66: Printing table: wt.60_vs_mut.0  
## DESeq2 step 5/5: 12/66: Printing table: mut.15_vs_mut.120  
## DESeq2 step 5/5: 13/66: Printing table: mut.180_vs_mut.120  
## DESeq2 step 5/5: 14/66: Printing table: mut.30_vs_mut.120  
## DESeq2 step 5/5: 15/66: Printing table: mut.60_vs_mut.120  
## DESeq2 step 5/5: 16/66: Printing table: wt.0_vs_mut.120  
## DESeq2 step 5/5: 17/66: Printing table: wt.120_vs_mut.120  
## DESeq2 step 5/5: 18/66: Printing table: wt.15_vs_mut.120  
## DESeq2 step 5/5: 19/66: Printing table: wt.180_vs_mut.120  
## DESeq2 step 5/5: 20/66: Printing table: wt.30_vs_mut.120  
## DESeq2 step 5/5: 21/66: Printing table: wt.60_vs_mut.120  
## DESeq2 step 5/5: 22/66: Printing table: mut.180_vs_mut.15  
## DESeq2 step 5/5: 23/66: Printing table: mut.30_vs_mut.15  
## DESeq2 step 5/5: 24/66: Printing table: mut.60_vs_mut.15  
## DESeq2 step 5/5: 25/66: Printing table: wt.0_vs_mut.15  
## DESeq2 step 5/5: 26/66: Printing table: wt.120_vs_mut.15
```

```
## DESeq2 step 5/5: 27/66: Printing table: wt.15_vs_mut.15

## DESeq2 step 5/5: 28/66: Printing table: wt.180_vs_mut.15

## DESeq2 step 5/5: 29/66: Printing table: wt.30_vs_mut.15

## DESeq2 step 5/5: 30/66: Printing table: wt.60_vs_mut.15

## DESeq2 step 5/5: 31/66: Printing table: mut.30_vs_mut.180

## DESeq2 step 5/5: 32/66: Printing table: mut.60_vs_mut.180

## DESeq2 step 5/5: 33/66: Printing table: wt.0_vs_mut.180

## DESeq2 step 5/5: 34/66: Printing table: wt.120_vs_mut.180

## DESeq2 step 5/5: 35/66: Printing table: wt.15_vs_mut.180

## DESeq2 step 5/5: 36/66: Printing table: wt.180_vs_mut.180

## DESeq2 step 5/5: 37/66: Printing table: wt.30_vs_mut.180

## DESeq2 step 5/5: 38/66: Printing table: wt.60_vs_mut.180

## DESeq2 step 5/5: 39/66: Printing table: mut.60_vs_mut.30

## DESeq2 step 5/5: 40/66: Printing table: wt.0_vs_mut.30

## DESeq2 step 5/5: 41/66: Printing table: wt.120_vs_mut.30

## DESeq2 step 5/5: 42/66: Printing table: wt.15_vs_mut.30

## DESeq2 step 5/5: 43/66: Printing table: wt.180_vs_mut.30

## DESeq2 step 5/5: 44/66: Printing table: wt.30_vs_mut.30

## DESeq2 step 5/5: 45/66: Printing table: wt.60_vs_mut.30

## DESeq2 step 5/5: 46/66: Printing table: wt.0_vs_mut.60

## DESeq2 step 5/5: 47/66: Printing table: wt.120_vs_mut.60

## DESeq2 step 5/5: 48/66: Printing table: wt.15_vs_mut.60

## DESeq2 step 5/5: 49/66: Printing table: wt.180_vs_mut.60

## DESeq2 step 5/5: 50/66: Printing table: wt.30_vs_mut.60
```

```
## DESeq2 step 5/5: 51/66: Printing table: wt.60_vs_mut.60

## DESeq2 step 5/5: 52/66: Printing table: wt.120_vs_wt.0

## DESeq2 step 5/5: 53/66: Printing table: wt.15_vs_wt.0

## DESeq2 step 5/5: 54/66: Printing table: wt.180_vs_wt.0

## DESeq2 step 5/5: 55/66: Printing table: wt.30_vs_wt.0

## DESeq2 step 5/5: 56/66: Printing table: wt.60_vs_wt.0

## DESeq2 step 5/5: 57/66: Printing table: wt.15_vs_wt.120

## DESeq2 step 5/5: 58/66: Printing table: wt.180_vs_wt.120

## DESeq2 step 5/5: 59/66: Printing table: wt.30_vs_wt.120

## DESeq2 step 5/5: 60/66: Printing table: wt.60_vs_wt.120

## DESeq2 step 5/5: 61/66: Printing table: wt.180_vs_wt.15

## DESeq2 step 5/5: 62/66: Printing table: wt.30_vs_wt.15

## DESeq2 step 5/5: 63/66: Printing table: wt.60_vs_wt.15

## DESeq2 step 5/5: 64/66: Printing table: wt.30_vs_wt.180

## DESeq2 step 5/5: 65/66: Printing table: wt.60_vs_wt.180

## DESeq2 step 5/5: 66/66: Printing table: wt.60_vs_wt.30

## Collected coefficients for: mut.0

## Collected coefficients for: mut.120

## Collected coefficients for: mut.15

## Collected coefficients for: mut.180

## Collected coefficients for: mut.30

## Collected coefficients for: mut.60

## Collected coefficients for: wt.0

## Collected coefficients for: wt.120
```

```
## Collected coefficients for: wt.15

## Collected coefficients for: wt.180

## Collected coefficients for: wt.30

## Collected coefficients for: wt.60

## Starting edgeR pairwise comparisons.

## EdgeR step 1/9: normalizing data.

## EdgeR step 2/9: Estimating the common dispersion.

## EdgeR step 3/9: Estimating dispersion across genes.

## EdgeR step 4/9: Estimating GLM Common dispersion.

## EdgeR step 5/9: Estimating GLM Trended dispersion.

## EdgeR step 6/9: Estimating GLM Tagged dispersion.

## EdgeR step 7/9: Running glmFit.

## EdgeR step 8/9: Making pairwise contrasts.

## As a reference, the identity is: mut.0 = mut.0,

## As a reference, the identity is: mut.120 = mut.120,

## As a reference, the identity is: mut.15 = mut.15,

## As a reference, the identity is: mut.180 = mut.180,

## As a reference, the identity is: mut.30 = mut.30,

## As a reference, the identity is: mut.60 = mut.60,

## As a reference, the identity is: wt.0 = wt.0,

## As a reference, the identity is: wt.120 = wt.120,

## As a reference, the identity is: wt.15 = wt.15,

## As a reference, the identity is: wt.180 = wt.180,

## As a reference, the identity is: wt.30 = wt.30,
```

```
## As a reference, the identity is: wt.60 = wt.60,  
  
## EdgeR step 9/9: 1/66: Printing table: mut.120_vs_mut.0.  
  
## EdgeR step 9/9: 2/66: Printing table: mut.15_vs_mut.0.  
  
## EdgeR step 9/9: 3/66: Printing table: mut.180_vs_mut.0.  
  
## EdgeR step 9/9: 4/66: Printing table: mut.30_vs_mut.0.  
  
## EdgeR step 9/9: 5/66: Printing table: mut.60_vs_mut.0.  
  
## EdgeR step 9/9: 6/66: Printing table: wt.0_vs_mut.0.  
  
## EdgeR step 9/9: 7/66: Printing table: wt.120_vs_mut.0.  
  
## EdgeR step 9/9: 8/66: Printing table: wt.15_vs_mut.0.  
  
## EdgeR step 9/9: 9/66: Printing table: wt.180_vs_mut.0.  
  
## EdgeR step 9/9: 10/66: Printing table: wt.30_vs_mut.0.  
  
## EdgeR step 9/9: 11/66: Printing table: wt.60_vs_mut.0.  
  
## EdgeR step 9/9: 12/66: Printing table: mut.15_vs_mut.120.  
  
## EdgeR step 9/9: 13/66: Printing table: mut.180_vs_mut.120.  
  
## EdgeR step 9/9: 14/66: Printing table: mut.30_vs_mut.120.  
  
## EdgeR step 9/9: 15/66: Printing table: mut.60_vs_mut.120.  
  
## EdgeR step 9/9: 16/66: Printing table: wt.0_vs_mut.120.  
  
## EdgeR step 9/9: 17/66: Printing table: wt.120_vs_mut.120.  
  
## EdgeR step 9/9: 18/66: Printing table: wt.15_vs_mut.120.  
  
## EdgeR step 9/9: 19/66: Printing table: wt.180_vs_mut.120.  
  
## EdgeR step 9/9: 20/66: Printing table: wt.30_vs_mut.120.  
  
## EdgeR step 9/9: 21/66: Printing table: wt.60_vs_mut.120.  
  
## EdgeR step 9/9: 22/66: Printing table: mut.180_vs_mut.15.  
  
## EdgeR step 9/9: 23/66: Printing table: mut.30_vs_mut.15.
```

```
## EdgeR step 9/9: 24/66: Printing table: mut.60_vs_mut.15.

## EdgeR step 9/9: 25/66: Printing table: wt.0_vs_mut.15.

## EdgeR step 9/9: 26/66: Printing table: wt.120_vs_mut.15.

## EdgeR step 9/9: 27/66: Printing table: wt.15_vs_mut.15.

## EdgeR step 9/9: 28/66: Printing table: wt.180_vs_mut.15.

## EdgeR step 9/9: 29/66: Printing table: wt.30_vs_mut.15.

## EdgeR step 9/9: 30/66: Printing table: wt.60_vs_mut.15.

## EdgeR step 9/9: 31/66: Printing table: mut.30_vs_mut.180.

## EdgeR step 9/9: 32/66: Printing table: mut.60_vs_mut.180.

## EdgeR step 9/9: 33/66: Printing table: wt.0_vs_mut.180.

## EdgeR step 9/9: 34/66: Printing table: wt.120_vs_mut.180.

## EdgeR step 9/9: 35/66: Printing table: wt.15_vs_mut.180.

## EdgeR step 9/9: 36/66: Printing table: wt.180_vs_mut.180.

## EdgeR step 9/9: 37/66: Printing table: wt.30_vs_mut.180.

## EdgeR step 9/9: 38/66: Printing table: wt.60_vs_mut.180.

## EdgeR step 9/9: 39/66: Printing table: mut.60_vs_mut.30.

## EdgeR step 9/9: 40/66: Printing table: wt.0_vs_mut.30.

## EdgeR step 9/9: 41/66: Printing table: wt.120_vs_mut.30.

## EdgeR step 9/9: 42/66: Printing table: wt.15_vs_mut.30.

## EdgeR step 9/9: 43/66: Printing table: wt.180_vs_mut.30.

## EdgeR step 9/9: 44/66: Printing table: wt.30_vs_mut.30.

## EdgeR step 9/9: 45/66: Printing table: wt.60_vs_mut.30.

## EdgeR step 9/9: 46/66: Printing table: wt.0_vs_mut.60.

## EdgeR step 9/9: 47/66: Printing table: wt.120_vs_mut.60.
```

```

## EdgeR step 9/9: 48/66: Printing table: wt.15_vs_mut.60.

## EdgeR step 9/9: 49/66: Printing table: wt.180_vs_mut.60.

## EdgeR step 9/9: 50/66: Printing table: wt.30_vs_mut.60.

## EdgeR step 9/9: 51/66: Printing table: wt.60_vs_mut.60.

## EdgeR step 9/9: 52/66: Printing table: wt.120_vs_wt.0.

## EdgeR step 9/9: 53/66: Printing table: wt.15_vs_wt.0.

## EdgeR step 9/9: 54/66: Printing table: wt.180_vs_wt.0.

## EdgeR step 9/9: 55/66: Printing table: wt.30_vs_wt.0.

## EdgeR step 9/9: 56/66: Printing table: wt.60_vs_wt.0.

## EdgeR step 9/9: 57/66: Printing table: wt.15_vs_wt.120.

## EdgeR step 9/9: 58/66: Printing table: wt.180_vs_wt.120.

## EdgeR step 9/9: 59/66: Printing table: wt.30_vs_wt.120.

## EdgeR step 9/9: 60/66: Printing table: wt.60_vs_wt.120.

## EdgeR step 9/9: 61/66: Printing table: wt.180_vs_wt.15.

## EdgeR step 9/9: 62/66: Printing table: wt.30_vs_wt.15.

## EdgeR step 9/9: 63/66: Printing table: wt.60_vs_wt.15.

## EdgeR step 9/9: 64/66: Printing table: wt.30_vs_wt.180.

## EdgeR step 9/9: 65/66: Printing table: wt.60_vs_wt.180.

## EdgeR step 9/9: 66/66: Printing table: wt.60_vs_wt.30.

## Starting basic pairwise comparison.

## Basic step 1/3: Creating median and variance tables.

## Basic step 2/3: Performing comparisons.

## Basic step 2/3: 1/66: Performing log2 subtraction: mut.120_vs_mut.0

## Basic step 2/3: 2/66: Performing log2 subtraction: mut.15_vs_mut.0

```

```
## Basic step 2/3: 3/66: Performing log2 subtraction: mut.180_vs_mut.0

## Basic step 2/3: 4/66: Performing log2 subtraction: mut.30_vs_mut.0

## Basic step 2/3: 5/66: Performing log2 subtraction: mut.60_vs_mut.0

## Basic step 2/3: 6/66: Performing log2 subtraction: wt.0_vs_mut.0

## Basic step 2/3: 7/66: Performing log2 subtraction: wt.120_vs_mut.0

## Basic step 2/3: 8/66: Performing log2 subtraction: wt.15_vs_mut.0

## Basic step 2/3: 9/66: Performing log2 subtraction: wt.180_vs_mut.0

## Basic step 2/3: 10/66: Performing log2 subtraction: wt.30_vs_mut.0

## Basic step 2/3: 11/66: Performing log2 subtraction: wt.60_vs_mut.0

## Basic step 2/3: 12/66: Performing log2 subtraction: mut.15_vs_mut.120

## Basic step 2/3: 13/66: Performing log2 subtraction: mut.180_vs_mut.120

## Basic step 2/3: 14/66: Performing log2 subtraction: mut.30_vs_mut.120

## Basic step 2/3: 15/66: Performing log2 subtraction: mut.60_vs_mut.120

## Basic step 2/3: 16/66: Performing log2 subtraction: wt.0_vs_mut.120

## Basic step 2/3: 17/66: Performing log2 subtraction: wt.120_vs_mut.120

## Basic step 2/3: 18/66: Performing log2 subtraction: wt.15_vs_mut.120

## Basic step 2/3: 19/66: Performing log2 subtraction: wt.180_vs_mut.120

## Basic step 2/3: 20/66: Performing log2 subtraction: wt.30_vs_mut.120

## Basic step 2/3: 21/66: Performing log2 subtraction: wt.60_vs_mut.120

## Basic step 2/3: 22/66: Performing log2 subtraction: mut.180_vs_mut.15

## Basic step 2/3: 23/66: Performing log2 subtraction: mut.30_vs_mut.15

## Basic step 2/3: 24/66: Performing log2 subtraction: mut.60_vs_mut.15

## Basic step 2/3: 25/66: Performing log2 subtraction: wt.0_vs_mut.15

## Basic step 2/3: 26/66: Performing log2 subtraction: wt.120_vs_mut.15
```

```
## Basic step 2/3: 27/66: Performing log2 subtraction: wt.15_vs_mut.15

## Basic step 2/3: 28/66: Performing log2 subtraction: wt.180_vs_mut.15

## Basic step 2/3: 29/66: Performing log2 subtraction: wt.30_vs_mut.15

## Basic step 2/3: 30/66: Performing log2 subtraction: wt.60_vs_mut.15

## Basic step 2/3: 31/66: Performing log2 subtraction: mut.30_vs_mut.180

## Basic step 2/3: 32/66: Performing log2 subtraction: mut.60_vs_mut.180

## Basic step 2/3: 33/66: Performing log2 subtraction: wt.0_vs_mut.180

## Basic step 2/3: 34/66: Performing log2 subtraction: wt.120_vs_mut.180

## Basic step 2/3: 35/66: Performing log2 subtraction: wt.15_vs_mut.180

## Basic step 2/3: 36/66: Performing log2 subtraction: wt.180_vs_mut.180

## Basic step 2/3: 37/66: Performing log2 subtraction: wt.30_vs_mut.180

## Basic step 2/3: 38/66: Performing log2 subtraction: wt.60_vs_mut.180

## Basic step 2/3: 39/66: Performing log2 subtraction: mut.60_vs_mut.30

## Basic step 2/3: 40/66: Performing log2 subtraction: wt.0_vs_mut.30

## Basic step 2/3: 41/66: Performing log2 subtraction: wt.120_vs_mut.30

## Basic step 2/3: 42/66: Performing log2 subtraction: wt.15_vs_mut.30

## Basic step 2/3: 43/66: Performing log2 subtraction: wt.180_vs_mut.30

## Basic step 2/3: 44/66: Performing log2 subtraction: wt.30_vs_mut.30

## Basic step 2/3: 45/66: Performing log2 subtraction: wt.60_vs_mut.30

## Basic step 2/3: 46/66: Performing log2 subtraction: wt.0_vs_mut.60

## Basic step 2/3: 47/66: Performing log2 subtraction: wt.120_vs_mut.60

## Basic step 2/3: 48/66: Performing log2 subtraction: wt.15_vs_mut.60

## Basic step 2/3: 49/66: Performing log2 subtraction: wt.180_vs_mut.60

## Basic step 2/3: 50/66: Performing log2 subtraction: wt.30_vs_mut.60
```

```

## Basic step 2/3: 51/66: Performing log2 subtraction: wt.60_vs_mut.60

## Basic step 2/3: 52/66: Performing log2 subtraction: wt.120_vs_wt.0

## Basic step 2/3: 53/66: Performing log2 subtraction: wt.15_vs_wt.0

## Basic step 2/3: 54/66: Performing log2 subtraction: wt.180_vs_wt.0

## Basic step 2/3: 55/66: Performing log2 subtraction: wt.30_vs_wt.0

## Basic step 2/3: 56/66: Performing log2 subtraction: wt.60_vs_wt.0

## Basic step 2/3: 57/66: Performing log2 subtraction: wt.15_vs_wt.120

## Basic step 2/3: 58/66: Performing log2 subtraction: wt.180_vs_wt.120

## Basic step 2/3: 59/66: Performing log2 subtraction: wt.30_vs_wt.120

## Basic step 2/3: 60/66: Performing log2 subtraction: wt.60_vs_wt.120

## Basic step 2/3: 61/66: Performing log2 subtraction: wt.180_vs_wt.15

## Basic step 2/3: 62/66: Performing log2 subtraction: wt.30_vs_wt.15

## Basic step 2/3: 63/66: Performing log2 subtraction: wt.60_vs_wt.15

## Basic step 2/3: 64/66: Performing log2 subtraction: wt.30_vs_wt.180

## Basic step 2/3: 65/66: Performing log2 subtraction: wt.60_vs_wt.180

## Basic step 2/3: 66/66: Performing log2 subtraction: wt.60_vs_wt.30

## Basic step 3/3: Creating faux DE Tables.

## Basic: Returning tables.

## 1/66: Comparing analyses: mut.120_vs_mut.0

## 2/66: Comparing analyses: mut.15_vs_mut.0

## 3/66: Comparing analyses: mut.180_vs_mut.0

## 4/66: Comparing analyses: mut.30_vs_mut.0

## 5/66: Comparing analyses: mut.60_vs_mut.0

## 6/66: Comparing analyses: wt.0_vs_mut.0

```

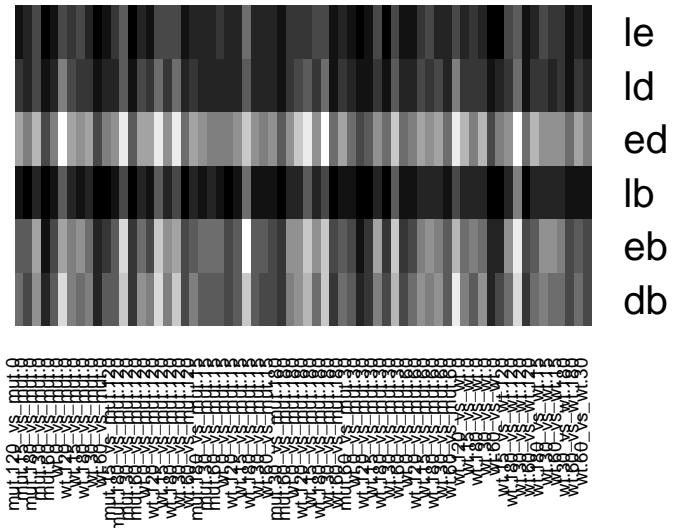
```
## 7/66: Comparing analyses: wt.120_vs_mut.0  
## 8/66: Comparing analyses: wt.15_vs_mut.0  
## 9/66: Comparing analyses: wt.180_vs_mut.0  
## 10/66: Comparing analyses: wt.30_vs_mut.0  
## 11/66: Comparing analyses: wt.60_vs_mut.0  
## 12/66: Comparing analyses: mut.15_vs_mut.120  
## 13/66: Comparing analyses: mut.180_vs_mut.120  
## 14/66: Comparing analyses: mut.30_vs_mut.120  
## 15/66: Comparing analyses: mut.60_vs_mut.120  
## 16/66: Comparing analyses: wt.0_vs_mut.120  
## 17/66: Comparing analyses: wt.120_vs_mut.120  
## 18/66: Comparing analyses: wt.15_vs_mut.120  
## 19/66: Comparing analyses: wt.180_vs_mut.120  
## 20/66: Comparing analyses: wt.30_vs_mut.120  
## 21/66: Comparing analyses: wt.60_vs_mut.120  
## 22/66: Comparing analyses: mut.180_vs_mut.15  
## 23/66: Comparing analyses: mut.30_vs_mut.15  
## 24/66: Comparing analyses: mut.60_vs_mut.15  
## 25/66: Comparing analyses: wt.0_vs_mut.15  
## 26/66: Comparing analyses: wt.120_vs_mut.15  
## 27/66: Comparing analyses: wt.15_vs_mut.15  
## 28/66: Comparing analyses: wt.180_vs_mut.15  
## 29/66: Comparing analyses: wt.30_vs_mut.15  
## 30/66: Comparing analyses: wt.60_vs_mut.15
```

```
## 31/66: Comparing analyses: mut.30_vs_mut.180
## 32/66: Comparing analyses: mut.60_vs_mut.180
## 33/66: Comparing analyses: wt.0_vs_mut.180
## 34/66: Comparing analyses: wt.120_vs_mut.180
## 35/66: Comparing analyses: wt.15_vs_mut.180
## 36/66: Comparing analyses: wt.180_vs_mut.180
## 37/66: Comparing analyses: wt.30_vs_mut.180
## 38/66: Comparing analyses: wt.60_vs_mut.180
## 39/66: Comparing analyses: mut.60_vs_mut.30
## 40/66: Comparing analyses: wt.0_vs_mut.30
## 41/66: Comparing analyses: wt.120_vs_mut.30
## 42/66: Comparing analyses: wt.15_vs_mut.30
## 43/66: Comparing analyses: wt.180_vs_mut.30
## 44/66: Comparing analyses: wt.30_vs_mut.30
## 45/66: Comparing analyses: wt.60_vs_mut.30
## 46/66: Comparing analyses: wt.0_vs_mut.60
## 47/66: Comparing analyses: wt.120_vs_mut.60
## 48/66: Comparing analyses: wt.15_vs_mut.60
## 49/66: Comparing analyses: wt.180_vs_mut.60
## 50/66: Comparing analyses: wt.30_vs_mut.60
## 51/66: Comparing analyses: wt.60_vs_mut.60
## 52/66: Comparing analyses: wt.120_vs_wt.0
## 53/66: Comparing analyses: wt.15_vs_wt.0
## 54/66: Comparing analyses: wt.180_vs_wt.0
```

```

## 55/66: Comparing analyses: wt.30_vs_wt.0
## 56/66: Comparing analyses: wt.60_vs_wt.0
## 57/66: Comparing analyses: wt.15_vs_wt.120
## 58/66: Comparing analyses: wt.180_vs_wt.120
## 59/66: Comparing analyses: wt.30_vs_wt.120
## 60/66: Comparing analyses: wt.60_vs_wt.120
## 61/66: Comparing analyses: wt.180_vs_wt.15
## 62/66: Comparing analyses: wt.30_vs_wt.15
## 63/66: Comparing analyses: wt.60_vs_wt.15
## 64/66: Comparing analyses: wt.30_vs_wt.180
## 65/66: Comparing analyses: wt.60_vs_wt.180
## 66/66: Comparing analyses: wt.60_vs_wt.30

```



```
all_combined <- combine_de_tables(all_comparisons)

## Working on table 1/66: mut.120_vs_mut.0

## Working on table 2/66: mut.15_vs_mut.0

## Working on table 3/66: mut.180_vs_mut.0

## Working on table 4/66: mut.30_vs_mut.0

## Working on table 5/66: mut.60_vs_mut.0

## Working on table 6/66: wt.0_vs_mut.0

## Working on table 7/66: wt.120_vs_mut.0

## Working on table 8/66: wt.15_vs_mut.0

## Working on table 9/66: wt.180_vs_mut.0

## Working on table 10/66: wt.30_vs_mut.0

## Working on table 11/66: wt.60_vs_mut.0

## Working on table 12/66: mut.15_vs_mut.120

## Working on table 13/66: mut.180_vs_mut.120

## Working on table 14/66: mut.30_vs_mut.120

## Working on table 15/66: mut.60_vs_mut.120

## Working on table 16/66: wt.0_vs_mut.120

## Working on table 17/66: wt.120_vs_mut.120

## Working on table 18/66: wt.15_vs_mut.120

## Working on table 19/66: wt.180_vs_mut.120

## Working on table 20/66: wt.30_vs_mut.120

## Working on table 21/66: wt.60_vs_mut.120

## Working on table 22/66: mut.180_vs_mut.15

## Working on table 23/66: mut.30_vs_mut.15
```

```
## Working on table 24/66: mut.60_vs_mut.15

## Working on table 25/66: wt.0_vs_mut.15

## Working on table 26/66: wt.120_vs_mut.15

## Working on table 27/66: wt.15_vs_mut.15

## Working on table 28/66: wt.180_vs_mut.15

## Working on table 29/66: wt.30_vs_mut.15

## Working on table 30/66: wt.60_vs_mut.15

## Working on table 31/66: mut.30_vs_mut.180

## Working on table 32/66: mut.60_vs_mut.180

## Working on table 33/66: wt.0_vs_mut.180

## Working on table 34/66: wt.120_vs_mut.180

## Working on table 35/66: wt.15_vs_mut.180

## Working on table 36/66: wt.180_vs_mut.180

## Working on table 37/66: wt.30_vs_mut.180

## Working on table 38/66: wt.60_vs_mut.180

## Working on table 39/66: mut.60_vs_mut.30

## Working on table 40/66: wt.0_vs_mut.30

## Working on table 41/66: wt.120_vs_mut.30

## Working on table 42/66: wt.15_vs_mut.30

## Working on table 43/66: wt.180_vs_mut.30

## Working on table 44/66: wt.30_vs_mut.30

## Working on table 45/66: wt.60_vs_mut.30

## Working on table 46/66: wt.0_vs_mut.60

## Working on table 47/66: wt.120_vs_mut.60
```

```

## Working on table 48/66: wt.15_vs_mut.60

## Working on table 49/66: wt.180_vs_mut.60

## Working on table 50/66: wt.30_vs_mut.60

## Working on table 51/66: wt.60_vs_mut.60

## Working on table 52/66: wt.120_vs_wt.0

## Working on table 53/66: wt.15_vs_wt.0

## Working on table 54/66: wt.180_vs_wt.0

## Working on table 55/66: wt.30_vs_wt.0

## Working on table 56/66: wt.60_vs_wt.0

## Working on table 57/66: wt.15_vs_wt.120

## Working on table 58/66: wt.180_vs_wt.120

## Working on table 59/66: wt.30_vs_wt.120

## Working on table 60/66: wt.60_vs_wt.120

## Working on table 61/66: wt.180_vs_wt.15

## Working on table 62/66: wt.30_vs_wt.15

## Working on table 63/66: wt.60_vs_wt.15

## Working on table 64/66: wt.30_vs_wt.180

## Working on table 65/66: wt.60_vs_wt.180

## Working on table 66/66: wt.60_vs_wt.30

sig_genes <- extract_significant_genes(all_combined, sig_table=NULL)

## Writing excel data sheet 1/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 117 genes.

## After (adj)p filter, the down genes table has 99 genes.

```

```

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 82 genes.

## After fold change filter, the down genes table has 44 genes.

## Writing excel data sheet 2/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 186 genes.

## After (adj)p filter, the down genes table has 2272 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 143 genes.

## After fold change filter, the down genes table has 818 genes.

## Writing excel data sheet 3/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 47 genes.

## After (adj)p filter, the down genes table has 164 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 27 genes.

## After fold change filter, the down genes table has 62 genes.

## Writing excel data sheet 4/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 665 genes.

## After (adj)p filter, the down genes table has 2008 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 367 genes.

## After fold change filter, the down genes table has 706 genes.

```

```

## Writing excel data sheet 5/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 302 genes.

## After (adj)p filter, the down genes table has 397 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 163 genes.

## After fold change filter, the down genes table has 110 genes.

## Writing excel data sheet 6/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 0 genes.

## After (adj)p filter, the down genes table has 1 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 0 genes.

## After fold change filter, the down genes table has 1 genes.

## Writing excel data sheet 7/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 83 genes.

## After (adj)p filter, the down genes table has 205 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 55 genes.

## After fold change filter, the down genes table has 82 genes.

## Writing excel data sheet 8/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 197 genes.

```

```
## After (adj)p filter, the down genes table has 3677 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 141 genes.

## After fold change filter, the down genes table has 1696 genes.

## Writing excel data sheet 9/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 95 genes.

## After (adj)p filter, the down genes table has 711 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 55 genes.

## After fold change filter, the down genes table has 135 genes.

## Writing excel data sheet 10/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 498 genes.

## After (adj)p filter, the down genes table has 3029 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 287 genes.

## After fold change filter, the down genes table has 1249 genes.

## Writing excel data sheet 11/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 304 genes.

## After (adj)p filter, the down genes table has 524 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 196 genes.
```

```

## After fold change filter, the down genes table has 141 genes.

## Writing excel data sheet 12/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 169 genes.

## After (adj)p filter, the down genes table has 2030 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 99 genes.

## After fold change filter, the down genes table has 746 genes.

## Writing excel data sheet 13/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 0 genes.

## After (adj)p filter, the down genes table has 0 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 0 genes.

## After fold change filter, the down genes table has 0 genes.

## Writing excel data sheet 14/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 550 genes.

## After (adj)p filter, the down genes table has 1555 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 291 genes.

## After fold change filter, the down genes table has 521 genes.

## Writing excel data sheet 15/66

## Assuming the fold changes are on the log scale and so taking >< 0

```

```

## After (adj)p filter, the up genes table has 128 genes.

## After (adj)p filter, the down genes table has 100 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 65 genes.

## After fold change filter, the down genes table has 47 genes.

## Writing excel data sheet 16/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 58 genes.

## After (adj)p filter, the down genes table has 204 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 29 genes.

## After fold change filter, the down genes table has 126 genes.

## Writing excel data sheet 17/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 0 genes.

## After (adj)p filter, the down genes table has 0 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 0 genes.

## After fold change filter, the down genes table has 0 genes.

## Writing excel data sheet 18/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 148 genes.

## After (adj)p filter, the down genes table has 3420 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

```

```

## After fold change filter, the up genes table has 92 genes.

## After fold change filter, the down genes table has 1558 genes.

## Writing excel data sheet 19/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 0 genes.

## After (adj)p filter, the down genes table has 0 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 0 genes.

## After fold change filter, the down genes table has 0 genes.

## Writing excel data sheet 20/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 415 genes.

## After (adj)p filter, the down genes table has 2708 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 224 genes.

## After fold change filter, the down genes table has 1012 genes.

## Writing excel data sheet 21/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 140 genes.

## After (adj)p filter, the down genes table has 116 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 84 genes.

## After fold change filter, the down genes table has 31 genes.

## Writing excel data sheet 22/66

```

```

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 843 genes.

## After (adj)p filter, the down genes table has 189 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 385 genes.

## After fold change filter, the down genes table has 120 genes.

## Writing excel data sheet 23/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 1295 genes.

## After (adj)p filter, the down genes table has 367 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 587 genes.

## After fold change filter, the down genes table has 135 genes.

## Writing excel data sheet 24/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 2363 genes.

## After (adj)p filter, the down genes table has 330 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 921 genes.

## After fold change filter, the down genes table has 114 genes.

## Writing excel data sheet 25/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 1368 genes.

## After (adj)p filter, the down genes table has 264 genes.

```

```
## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 516 genes.

## After fold change filter, the down genes table has 173 genes.

## Writing excel data sheet 26/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 1117 genes.

## After (adj)p filter, the down genes table has 218 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 471 genes.

## After fold change filter, the down genes table has 127 genes.

## Writing excel data sheet 27/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 2 genes.

## After (adj)p filter, the down genes table has 48 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 1 genes.

## After fold change filter, the down genes table has 41 genes.

## Writing excel data sheet 28/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 815 genes.

## After (adj)p filter, the down genes table has 284 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 377 genes.

## After fold change filter, the down genes table has 145 genes.
```

```

## Writing excel data sheet 29/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 885 genes.

## After (adj)p filter, the down genes table has 860 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 416 genes.

## After fold change filter, the down genes table has 271 genes.

## Writing excel data sheet 30/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 1917 genes.

## After (adj)p filter, the down genes table has 307 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 755 genes.

## After fold change filter, the down genes table has 96 genes.

## Writing excel data sheet 31/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 801 genes.

## After (adj)p filter, the down genes table has 831 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 437 genes.

## After fold change filter, the down genes table has 285 genes.

## Writing excel data sheet 32/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 501 genes.

```

```

## After (adj)p filter, the down genes table has 119 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 174 genes.

## After fold change filter, the down genes table has 33 genes.

## Writing excel data sheet 33/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 47 genes.

## After (adj)p filter, the down genes table has 63 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 24 genes.

## After fold change filter, the down genes table has 46 genes.

## Writing excel data sheet 34/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 0 genes.

## After (adj)p filter, the down genes table has 0 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 0 genes.

## After fold change filter, the down genes table has 0 genes.

## Writing excel data sheet 35/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 183 genes.

## After (adj)p filter, the down genes table has 1988 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 126 genes.

```

```
## After fold change filter, the down genes table has 950 genes.

## Writing excel data sheet 36/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 0 genes.

## After (adj)p filter, the down genes table has 0 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 0 genes.

## After fold change filter, the down genes table has 0 genes.

## Writing excel data sheet 37/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 562 genes.

## After (adj)p filter, the down genes table has 1617 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 314 genes.

## After fold change filter, the down genes table has 557 genes.

## Writing excel data sheet 38/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 361 genes.

## After (adj)p filter, the down genes table has 122 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 191 genes.

## After fold change filter, the down genes table has 30 genes.

## Writing excel data sheet 39/66

## Assuming the fold changes are on the log scale and so taking >< 0
```

```
## After (adj)p filter, the up genes table has 1639 genes.

## After (adj)p filter, the down genes table has 517 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 497 genes.

## After fold change filter, the down genes table has 291 genes.

## Writing excel data sheet 40/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 1430 genes.

## After (adj)p filter, the down genes table has 880 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 451 genes.

## After fold change filter, the down genes table has 497 genes.

## Writing excel data sheet 41/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 985 genes.

## After (adj)p filter, the down genes table has 769 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 328 genes.

## After fold change filter, the down genes table has 396 genes.

## Writing excel data sheet 42/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 298 genes.

## After (adj)p filter, the down genes table has 2263 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc
```

```

## After fold change filter, the up genes table has 115 genes.

## After fold change filter, the down genes table has 1140 genes.

## Writing excel data sheet 43/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 778 genes.

## After (adj)p filter, the down genes table has 992 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 263 genes.

## After fold change filter, the down genes table has 472 genes.

## Writing excel data sheet 44/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 0 genes.

## After (adj)p filter, the down genes table has 3 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 0 genes.

## After fold change filter, the down genes table has 3 genes.

## Writing excel data sheet 45/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 1169 genes.

## After (adj)p filter, the down genes table has 512 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 335 genes.

## After fold change filter, the down genes table has 255 genes.

## Writing excel data sheet 46/66

```

```

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 267 genes.

## After (adj)p filter, the down genes table has 512 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 84 genes.

## After fold change filter, the down genes table has 254 genes.

## Writing excel data sheet 47/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 95 genes.

## After (adj)p filter, the down genes table has 315 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 37 genes.

## After fold change filter, the down genes table has 146 genes.

## Writing excel data sheet 48/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 250 genes.

## After (adj)p filter, the down genes table has 3459 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 128 genes.

## After fold change filter, the down genes table has 1726 genes.

## Writing excel data sheet 49/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 107 genes.

## After (adj)p filter, the down genes table has 772 genes.

```

```
## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 35 genes.

## After fold change filter, the down genes table has 202 genes.

## Writing excel data sheet 50/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 388 genes.

## After (adj)p filter, the down genes table has 2711 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 227 genes.

## After fold change filter, the down genes table has 1014 genes.

## Writing excel data sheet 51/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 2 genes.

## After (adj)p filter, the down genes table has 4 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 2 genes.

## After fold change filter, the down genes table has 4 genes.

## Writing excel data sheet 52/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 86 genes.

## After (adj)p filter, the down genes table has 77 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 63 genes.

## After fold change filter, the down genes table has 41 genes.
```

```

## Writing excel data sheet 53/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 203 genes.

## After (adj)p filter, the down genes table has 2696 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 133 genes.

## After fold change filter, the down genes table has 1214 genes.

## Writing excel data sheet 54/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 57 genes.

## After (adj)p filter, the down genes table has 112 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 42 genes.

## After fold change filter, the down genes table has 45 genes.

## Writing excel data sheet 55/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 561 genes.

## After (adj)p filter, the down genes table has 2314 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 319 genes.

## After fold change filter, the down genes table has 843 genes.

## Writing excel data sheet 56/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 429 genes.

```

```
## After (adj)p filter, the down genes table has 315 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 246 genes.

## After fold change filter, the down genes table has 86 genes.

## Writing excel data sheet 57/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 186 genes.

## After (adj)p filter, the down genes table has 2248 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 120 genes.

## After fold change filter, the down genes table has 1069 genes.

## Writing excel data sheet 58/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 0 genes.

## After (adj)p filter, the down genes table has 1 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 0 genes.

## After fold change filter, the down genes table has 1 genes.

## Writing excel data sheet 59/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 496 genes.

## After (adj)p filter, the down genes table has 1698 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 276 genes.
```

```

## After fold change filter, the down genes table has 603 genes.

## Writing excel data sheet 60/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 258 genes.

## After (adj)p filter, the down genes table has 106 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 142 genes.

## After fold change filter, the down genes table has 24 genes.

## Writing excel data sheet 61/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 1743 genes.

## After (adj)p filter, the down genes table has 232 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 870 genes.

## After fold change filter, the down genes table has 122 genes.

## Writing excel data sheet 62/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 1462 genes.

## After (adj)p filter, the down genes table has 543 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 773 genes.

## After fold change filter, the down genes table has 211 genes.

## Writing excel data sheet 63/66

## Assuming the fold changes are on the log scale and so taking >< 0

```

```

## After (adj)p filter, the up genes table has 3052 genes.

## After (adj)p filter, the down genes table has 212 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 1480 genes.

## After fold change filter, the down genes table has 85 genes.

## Writing excel data sheet 64/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 595 genes.

## After (adj)p filter, the down genes table has 1272 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 323 genes.

## After fold change filter, the down genes table has 442 genes.

## Writing excel data sheet 65/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 491 genes.

## After (adj)p filter, the down genes table has 116 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 184 genes.

## After fold change filter, the down genes table has 24 genes.

## Writing excel data sheet 66/66

## Assuming the fold changes are on the log scale and so taking >< 0

## After (adj)p filter, the up genes table has 2196 genes.

## After (adj)p filter, the down genes table has 336 genes.

## Assuming the fold changes are on the log scale and so taking -1 * fc

## After fold change filter, the up genes table has 781 genes.

## After fold change filter, the down genes table has 180 genes.

## Not printing excel sheets for the significant genes.

```

## Ontology searches

```
limma_results <- limma_comparison$all_tables
## The set of comparisons performed
names(limma_results)

## [1] "mut.0"           "mut.120"          "mut.15"
## [4] "mut.180"          "mut.30"            "mut.60"
## [7] "wt.0"             "wt.120"           "wt.15"
## [10] "wt.180"           "wt.30"            "wt.60"
## [13] "mut.120_vs_mut.0" "mut.15_vs_mut.0"  "mut.180_vs_mut.0"
## [16] "mut.30_vs_mut.0"  "mut.60_vs_mut.0"  "wt.0_vs_mut.0"
## [19] "wt.120_vs_mut.0"  "wt.15_vs_mut.0"  "wt.180_vs_mut.0"
## [22] "wt.30_vs_mut.0"   "wt.60_vs_mut.0"  "mut.15_vs_mut.120"
## [25] "mut.180_vs_mut.120" "mut.30_vs_mut.120" "mut.60_vs_mut.120"
## [28] "wt.0_vs_mut.120"  "wt.120_vs_mut.120" "wt.15_vs_mut.120"
## [31] "wt.180_vs_mut.120" "wt.30_vs_mut.120"  "wt.60_vs_mut.120"
## [34] "mut.180_vs_mut.15"  "mut.30_vs_mut.15"  "mut.60_vs_mut.15"
## [37] "wt.0_vs_mut.15"    "wt.120_vs_mut.15"  "wt.15_vs_mut.15"
## [40] "wt.180_vs_mut.15"  "wt.30_vs_mut.15"  "wt.60_vs_mut.15"
## [43] "mut.30_vs_mut.180"  "mut.60_vs_mut.180" "wt.0_vs_mut.180"
## [46] "wt.120_vs_mut.180"  "wt.15_vs_mut.180"  "wt.180_vs_mut.180"
## [49] "wt.30_vs_mut.180"  "wt.60_vs_mut.180" "mut.60_vs_mut.30"
## [52] "wt.0_vs_mut.30"    "wt.120_vs_mut.30"  "wt.15_vs_mut.30"
## [55] "wt.180_vs_mut.30"  "wt.30_vs_mut.30"   "wt.60_vs_mut.30"
## [58] "wt.0_vs_mut.60"    "wt.120_vs_mut.60"  "wt.15_vs_mut.60"
## [61] "wt.180_vs_mut.60"  "wt.30_vs_mut.60"   "wt.60_vs_mut.60"
## [64] "wt.120_vs_wt.0"   "wt.15_vs_wt.0"   "wt.180_vs_wt.0"
## [67] "wt.30_vs_wt.0"    "wt.60_vs_wt.0"   "wt.15_vs_wt.120"
## [70] "wt.180_vs_wt.120"  "wt.30_vs_wt.120"  "wt.60_vs_wt.120"
## [73] "wt.180_vs_wt.15"  "wt.30_vs_wt.15"  "wt.60_vs_wt.15"
## [76] "wt.30_vs_wt.180"  "wt.60_vs_wt.180" "wt.60_vs_wt.30"

table <- limma_results$wt.180_vs_wt.0
dim(table)

## [1] 7039      7

gene_names <- rownames(table)
updown_genes <- get_sig_genes(table)

## No n, z, nor fc provided, setting z to 1.

## Getting the genes >= 1 z scores away from the median of all.

## After z filter, the up genes table has 1231 genes.

## After z filter, the down genes table has 1290 genes.
```

```

##orthologs <- read.table("ftp://ftp.ebi.ac.uk/pub/databases/pombase/pombe/orthologs/cerevisiae-ortholo")
##colnames(orthologs) <- c("pombe","cerevisiae")

##head(updown_genes$up_genes)
##updown_genes$up_genes = merge(updown_genes$up_genes, orthologs, by.x="row.names", by.y="pombe")
##rownames(updown_genes$up_genes) = make.names(updown_genes$up_genes$cerevisiae, unique=TRUE)
##updown_genes$down_genes = merge(updown_genes$down_genes, orthologs, by.x="row.names", by.y="pombe")
##rownames(updown_genes$down_genes) = make.names(updown_genes$down_genes$cerevisiae, unique=TRUE)

require.auto("GenomicFeatures")
require.auto("biomaRt")
ensembl_pombe = biomaRt::useMart("fungal_mart", dataset="spombe_eg_gene", host="fungi.ensembl.org")
pombe_filters = biomaRt::listFilters(ensembl_pombe)
head(pombe_filters, n=20) ## 11 looks to be my guy

```

```

##
##          name
## 1      chromosome_name
## 2              start
## 3              end
## 4              strand
## 5      chromosomal_region
## 6      with_chembl
## 7      with_embl
## 8      with_protein_id
## 9      with_entrezgene
## 10             with_fypo
## 11             with_ontology_go
## 12             with_go
## 13      with_ox_goslim_goa
## 14      with_kegg_enzyme
## 15             with_merops
## 16             with_metacyc
## 17             with_mod
## 18             with_pdb
## 19      with_pombase_gene_name
## 20 with_pombase_interaction_genetic
##
##          description
## 1      Chromosome name
## 2              Start
## 3              End
## 4              Strand
## 5 e.g. 1:100:10000:-1, 1:100000:2000000:1
## 6             with ChEMBL ID(s)
## 7             with ENA/GenBank ID(s)
## 8      with ENA/GenBank protein ID(s)
## 9             with EntrezGene ID(s)
## 10            with FYPO term accession(s)
## 11             with GO ID(s)
## 12            with GO term accession(s)
## 13             with GOSlim GOA ID(s)
## 14             with KEGG enzyme ID(s)
## 15             with MEROPS ID(s)
## 16             with Metacyc ID(s)

```

```

## 17      with MOD term accession(s)
## 18          with PDB ID(s)
## 19          with PomBase gene name(s)
## 20  with PomBase genetic interaction ID(s)

## getBM(attributes=c('hgnc_symbol', 'chromosome_name', 'start_position', 'end_position'), filters='with'
pombe_goids = getBM(attributes=c('pombase_gene_name', 'go_accession'), values=gene_names, mart=ensembl_
colnames(pombe_goids) = c("ID", "GO")
pombe = makeTxDbFromBiomart(biomart ="fungal_mart", dataset = "spombe_eg_gene", host="fungi.ensembl.org

## Download and preprocess the 'transcripts' data frame ...

## OK

## Download and preprocess the 'spliceings' data frame ...

## OK

## Download and preprocess the 'genes' data frame ...

## OK

## Prepare the 'metadata' data frame ...

## OK

## Make the TxDb object ...

## Warning in .normarg_makeTxDb_chrominfo(chrominfo, transcripts$tx_chrom, :
## chromosome lengths and circularity flags are not available for this TxDb
## object

## OK

pombe_transcripts = as.data.frame(transcriptsBy(pombe))
lengths = pombe_transcripts[,c("group_name", "width")]
colnames(lengths) = c("ID", "width")
## Something useful I didn't notice before:
## makeTranscriptDbFromGFF() ## From GenomicFeatures, much like my own gff2df()

goseq_search = simple_goseq(de_genes=updown_genes$up_genes, lengths=lengths, goids=pombe_goids)

## simple_goseq() makes some pretty hard assumptions about the data it is fed:

## It requires 2 tables, one of GOids which must have columns (gene)ID and GO(category)

## The other table is of gene lengths with columns (gene)ID and (gene)width.

## Other columns are fine, but ignored.

```

```

## simple_goseq(): Using the explicit lengths df for gene lengths.

## simple_goseq(): Using the length data to fill in the de vector.

## Loading required package: DBI

## 

## Using manually entered categories.

## Calculating the p-values...

## simple_goseq(): Calculating q-values

## simple_goseq(): Filling godata with terms, this is slow.

## Testing that go categories are defined.

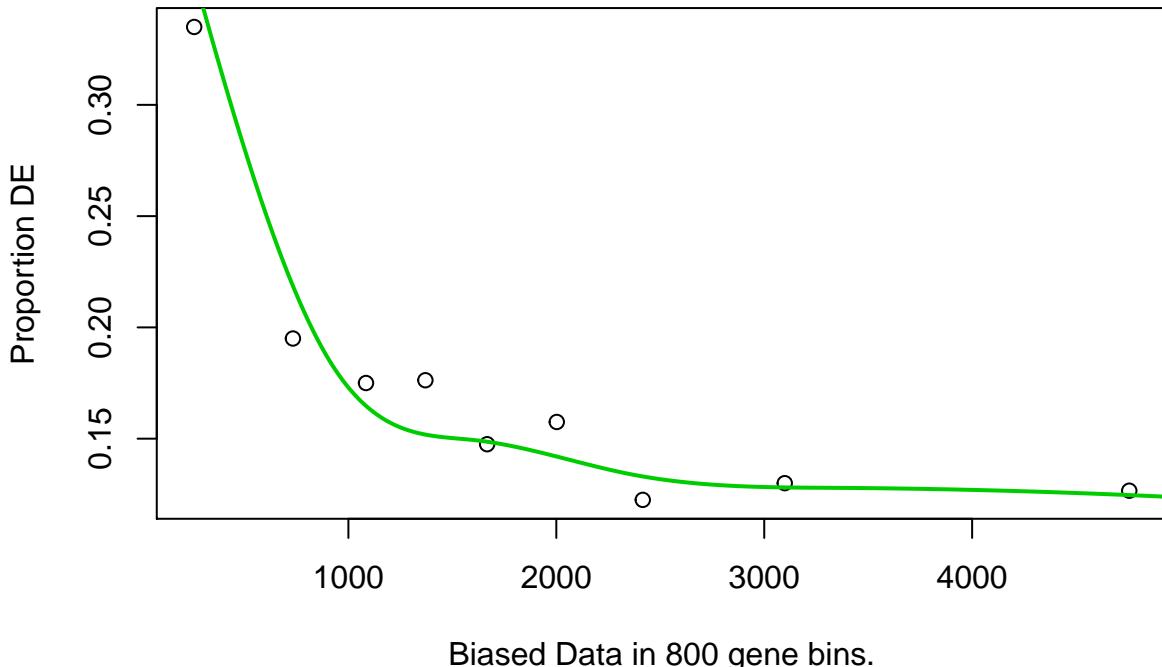
## Removing undefined categories.

## Gathering synonyms.

## Gathering category definitions.

## simple_goseq(): Making pvalue plots for the ontologies.

```



```
##spombe_gff = download.file(url="http://www.broadinstitute.org/ftp/pub/annotation/fungi/schizosaccharomyces_pombe/pombe.gff")
##cluster_search = simple_clusterprofiler(de_genes=updown_genes$up_genes, goids=pombe_goids, gff="pombe.gff")
##topgo_search = simple_topgo(de_genes=updown_genes$up_genes, goids_df=pombe_goids)
##gostats_search = simple_gostats(de_genes=updown_genes$up_genes, gff="pombe.gff", goids=pombe_goids, upregulated_only=TRUE)
```