

# hpgltools examples using the fission dataset

*atb abelew@gmail.com*

*2016-03-13*

## 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")  
library(fission)  
  
## Loading required package: GenomicRanges  
  
## Loading required package: methods  
  
## Loading required package: BiocGenerics  
  
## Loading required package: parallel  
  
##  
## Attaching package: 'BiocGenerics'  
  
## The following objects are masked from 'package:parallel':  
##  
##   clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,  
##   clusterExport, clusterMap, parApply, parCapply, parLapply,  
##   parLapplyLB, parRapply, parSapply, parSapplyLB  
  
## The following object is masked from 'package:stats':  
##  
##   xtabs
```

```

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, as.vector, cbind,
##     colnames, do.call, duplicated, eval, evalq, Filter, Find, get,
##     intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rep.int, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unlist, unsplit

## Loading required package: S4Vectors

## Loading required package: stats4

## Creating a generic function for 'nchar' from package 'base' in package 'S4Vectors'

## Loading required package: IRanges

## Loading required package: GenomeInfoDb

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

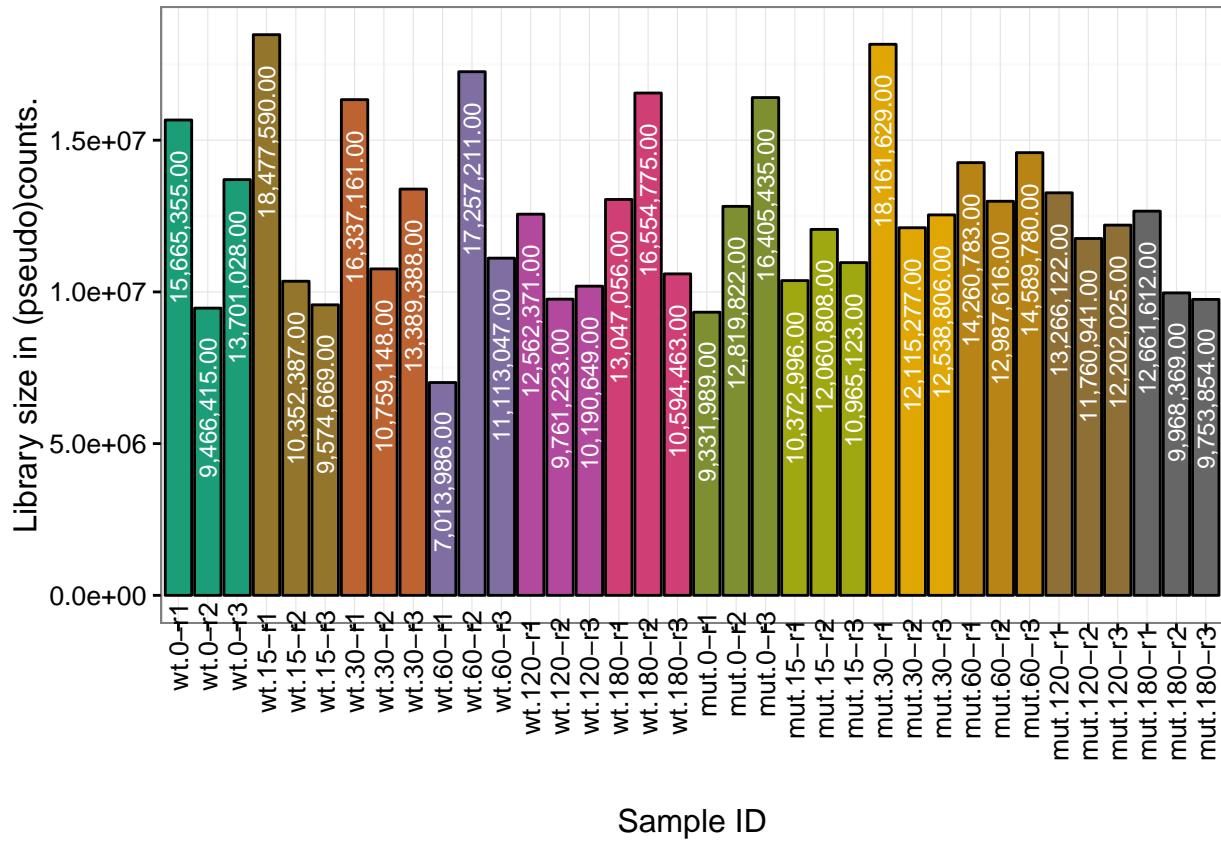
## Loading required namespace: Biobase

```

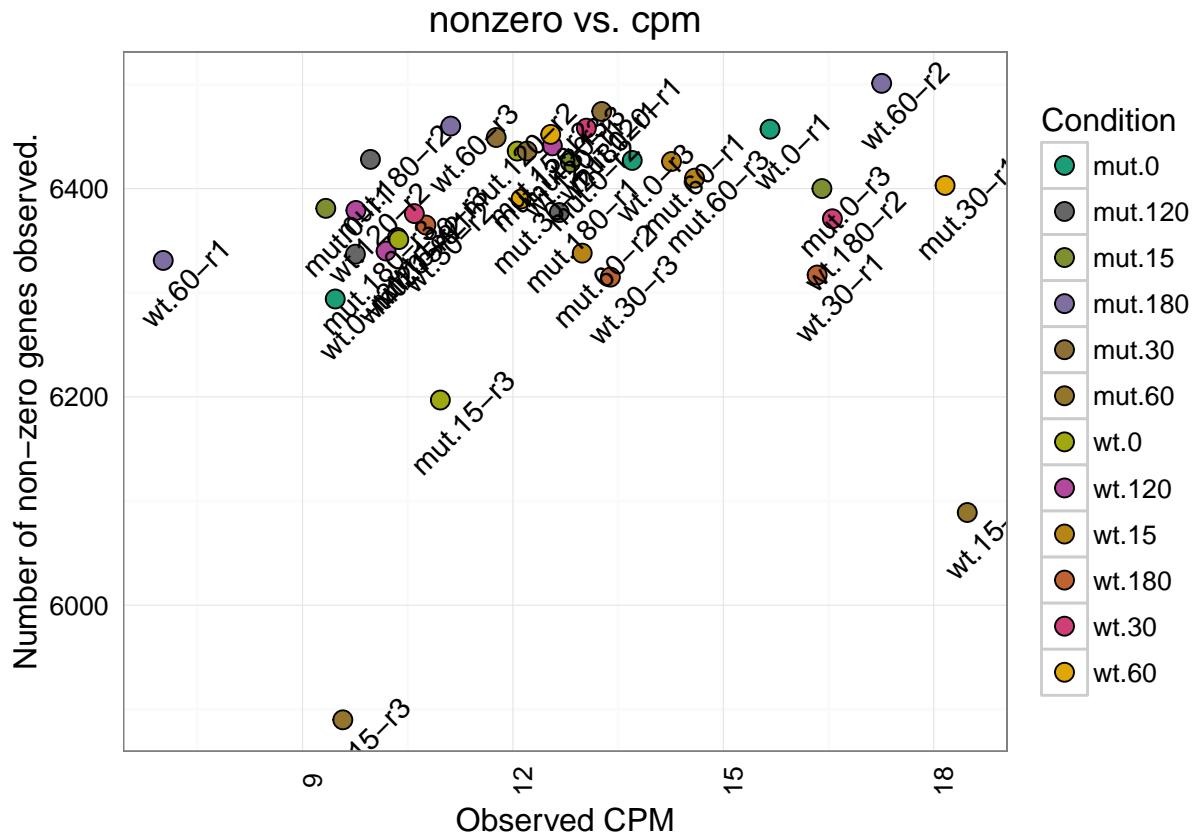
## Normalizing and exploring data

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
```



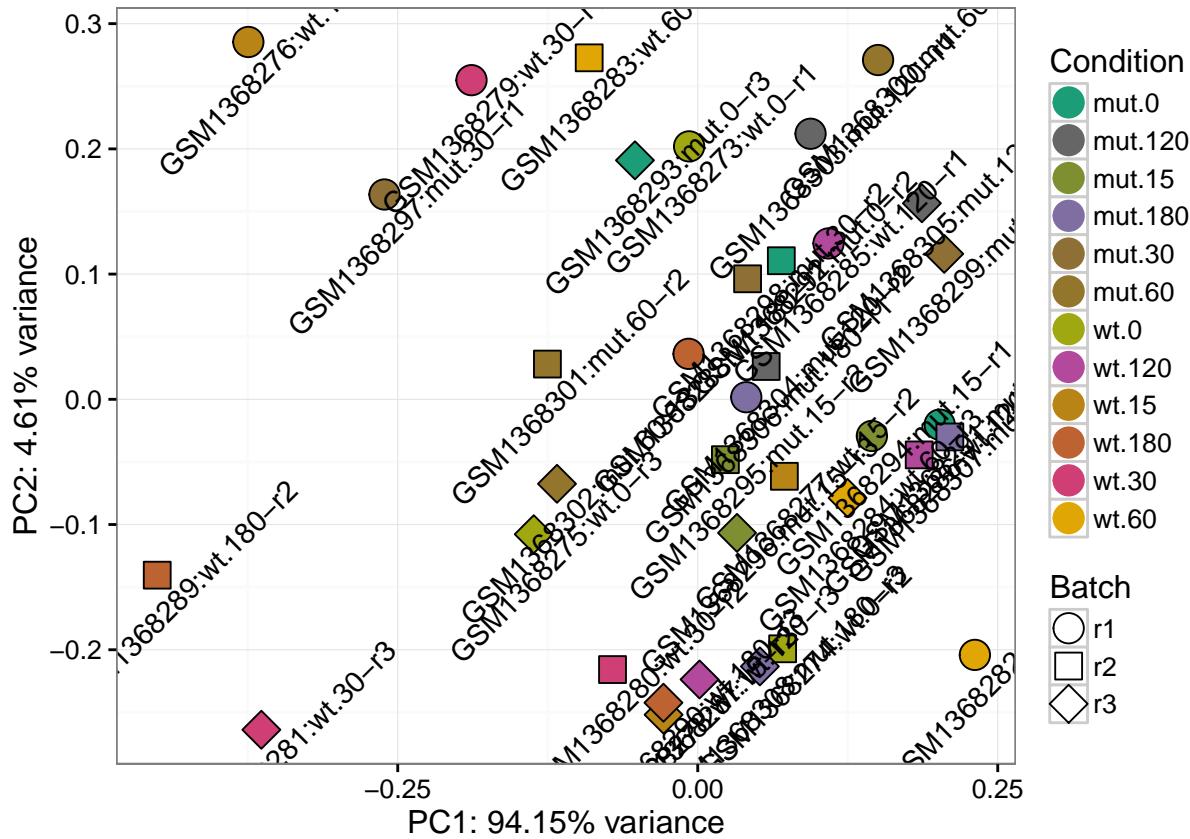
```
## Here we see that the wild type replicate 3 sample for 15 minutes has fewer non-zero genes than all i
fis_nonzero <- hpgl_nonzero(fission_expt, labels="boring", title="nonzero vs. cpm")
fis_nonzero
```



### 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.

```
## Something in this is causing a build loop on travis...
## I am no longer certain that code is maintained, what remains from it I might pull in to my own.
## require.auto("kokrah/cbcSEQ") ## Install Kwame's cbcSEQ
## 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)))

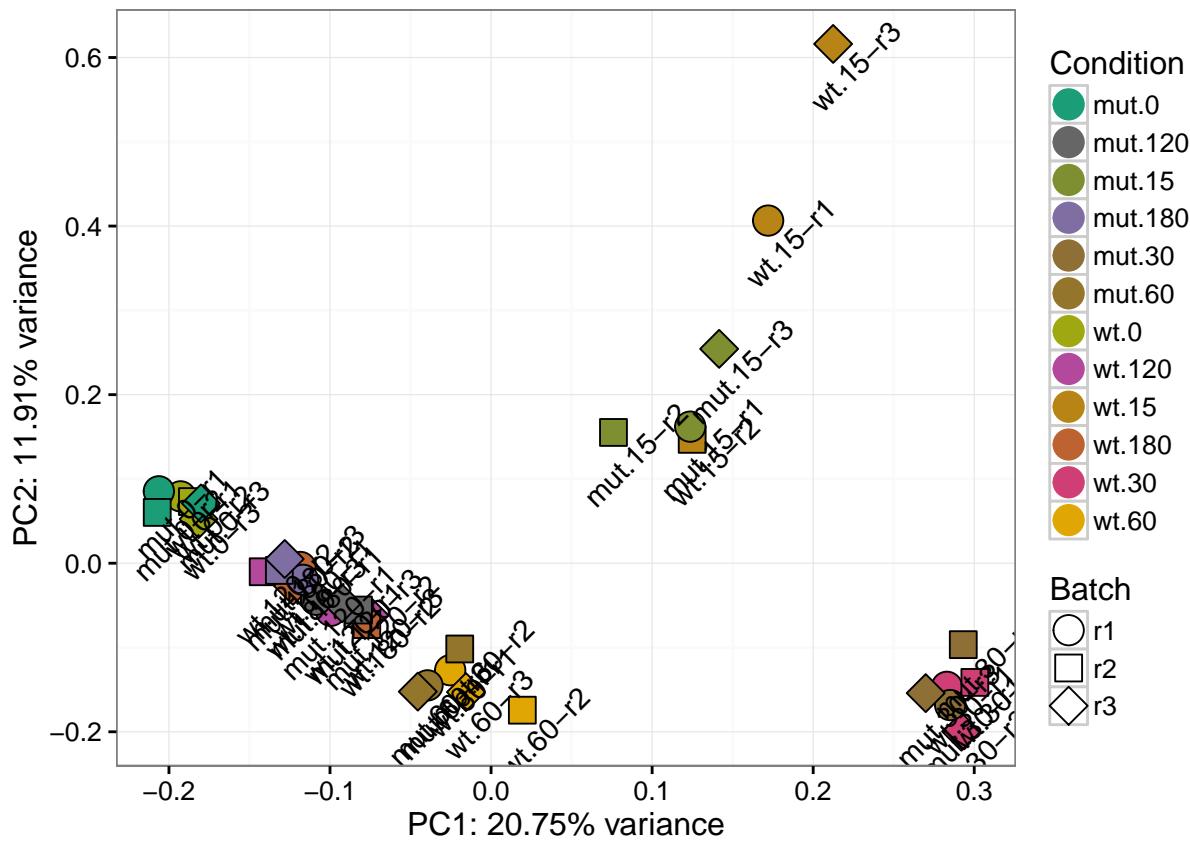
## It saves the current data into a slot named:
##   expt$backup_expressionset. It will also save copies of each step along the way
##   in expt$normalized with the corresponding libsizes. Keep the libsizes in mind
##   when invoking limma. The appropriate libsize is the non-log(cpm(normalized)).
##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

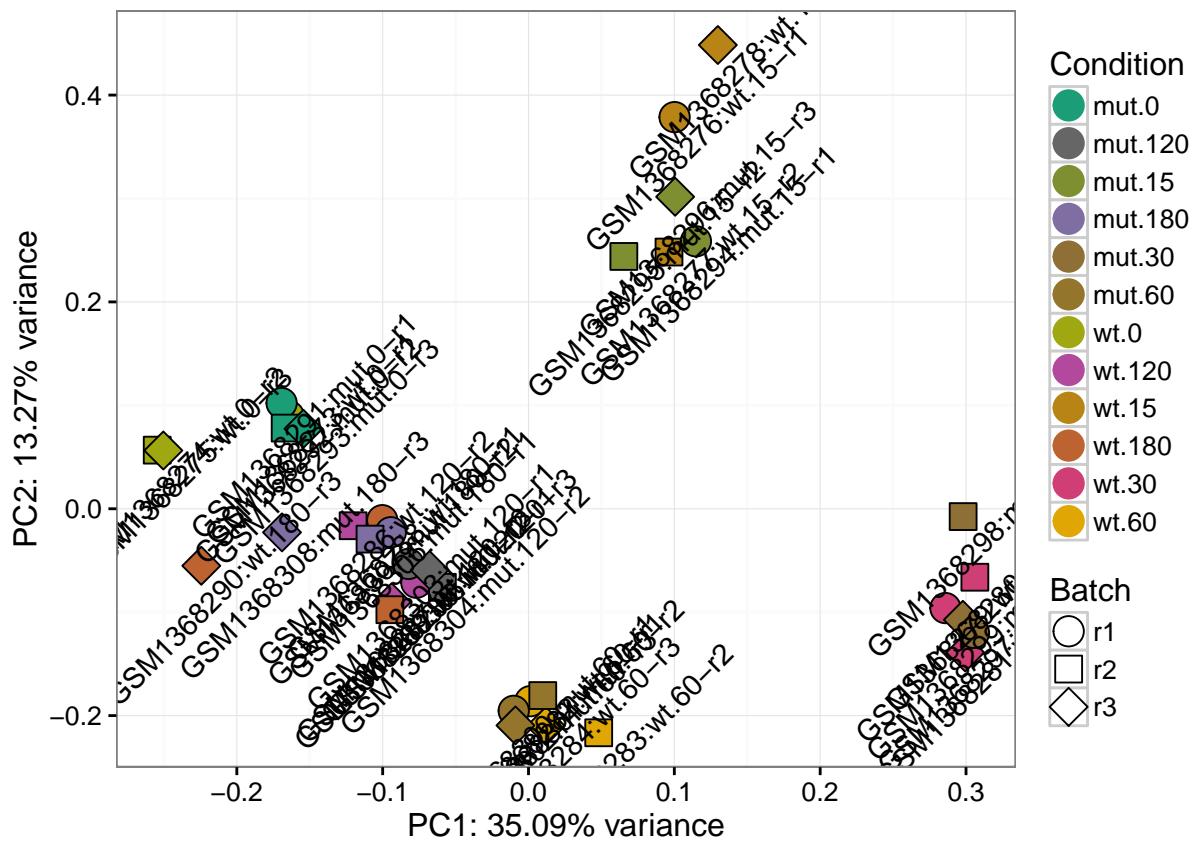
## Number of significant surrogate variables is: 1
## Iteration (out of 5 ):1 2 3 4 5

## 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

```



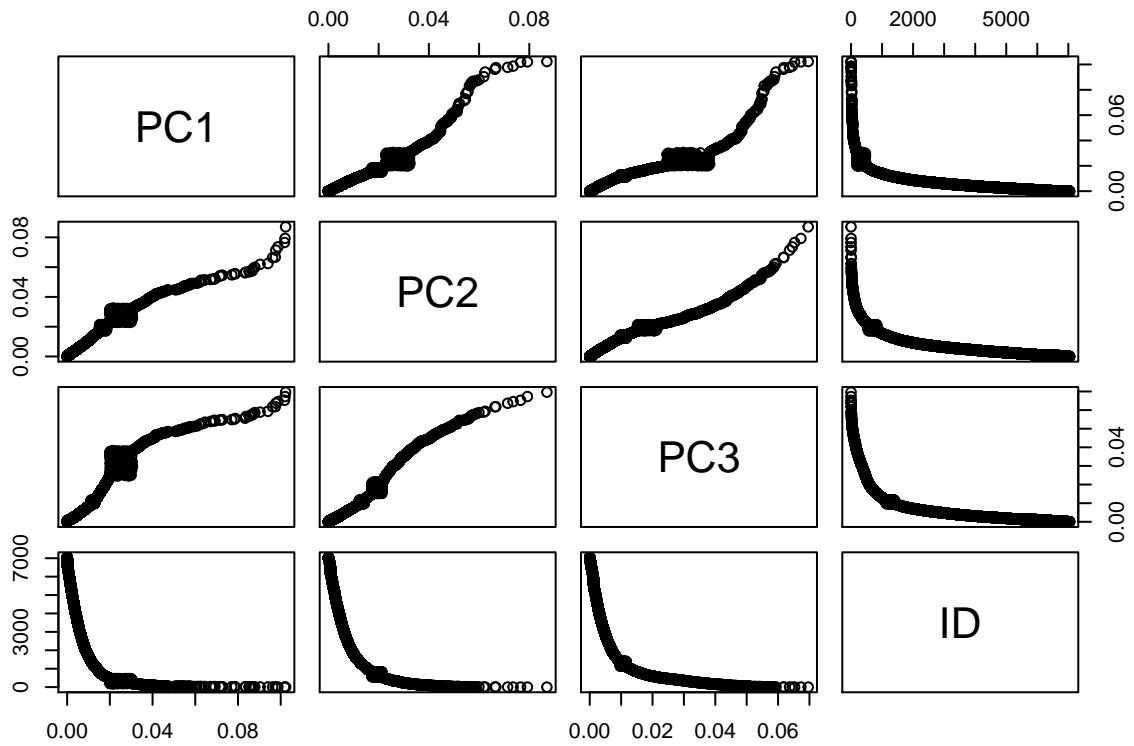
```

## 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

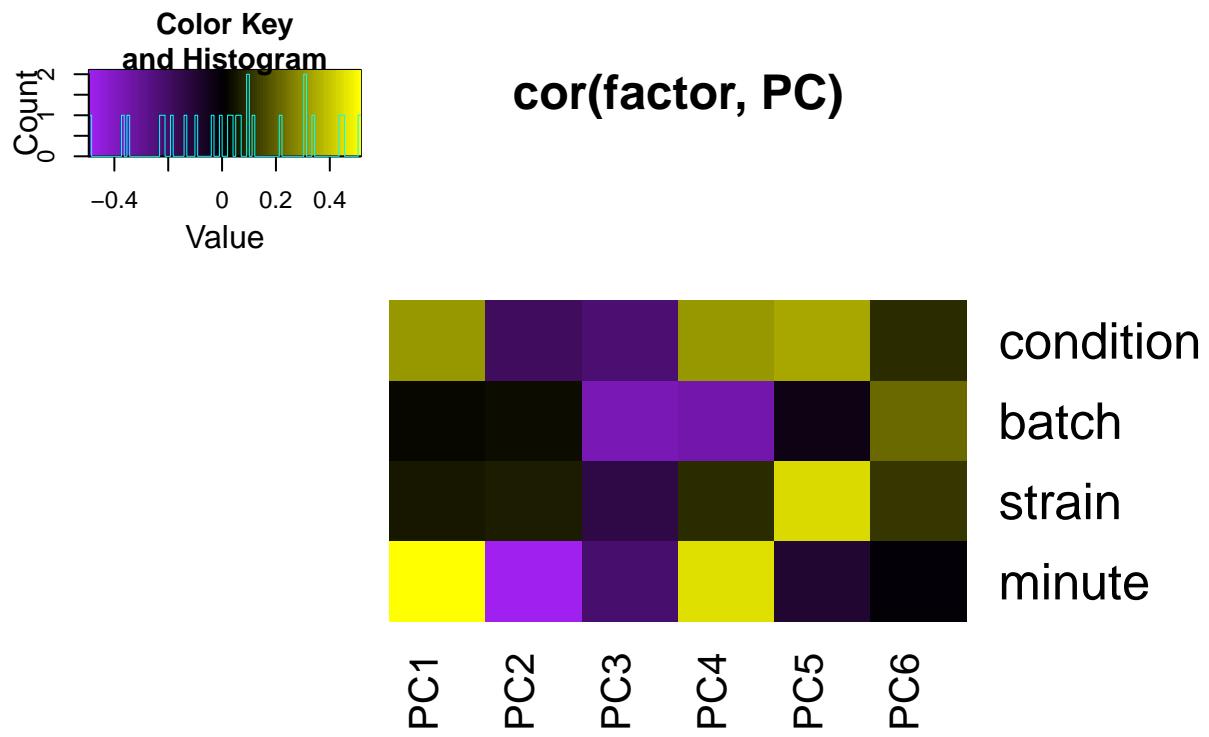
## pca_information provides some more information about the call to
## fast.svd that went into making the pca plot
fis_info <- pca_information(norm_expt, expt_factors=c("condition","batch","strain","minute"), num_compono

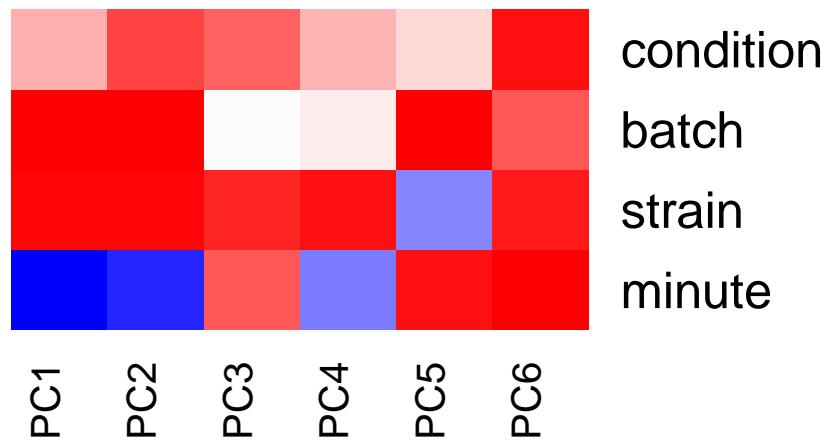
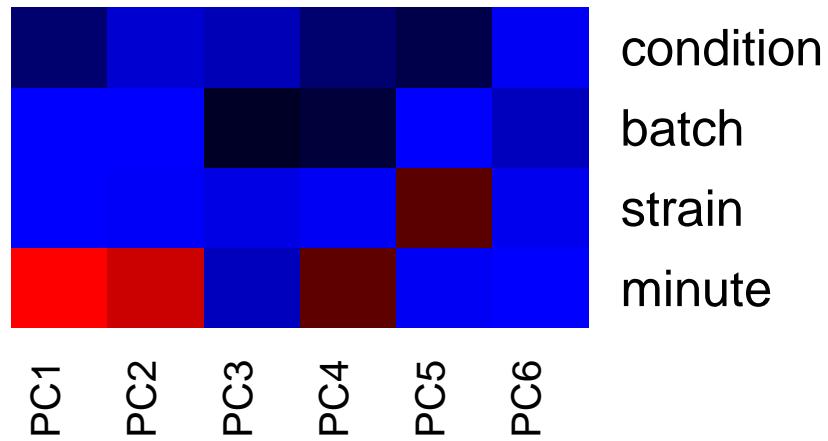
```

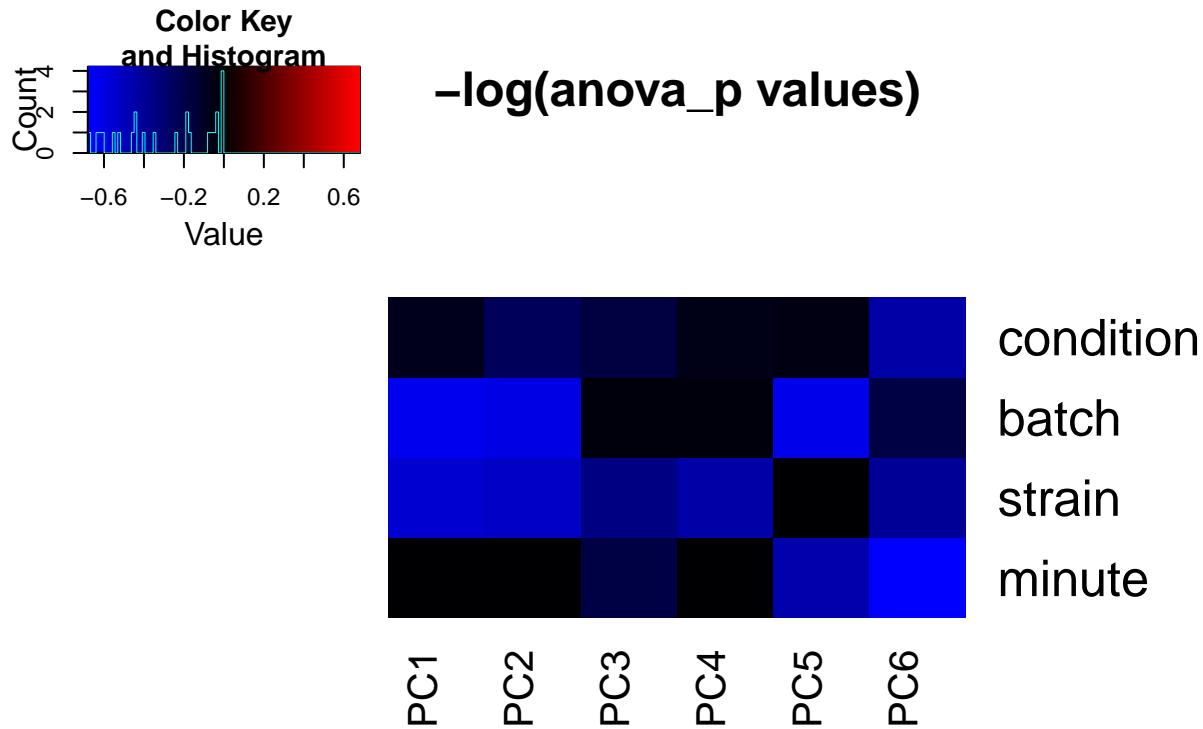
## The more shallow the curves in these plots, the more genes responsible for this principle component.



```
## [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.

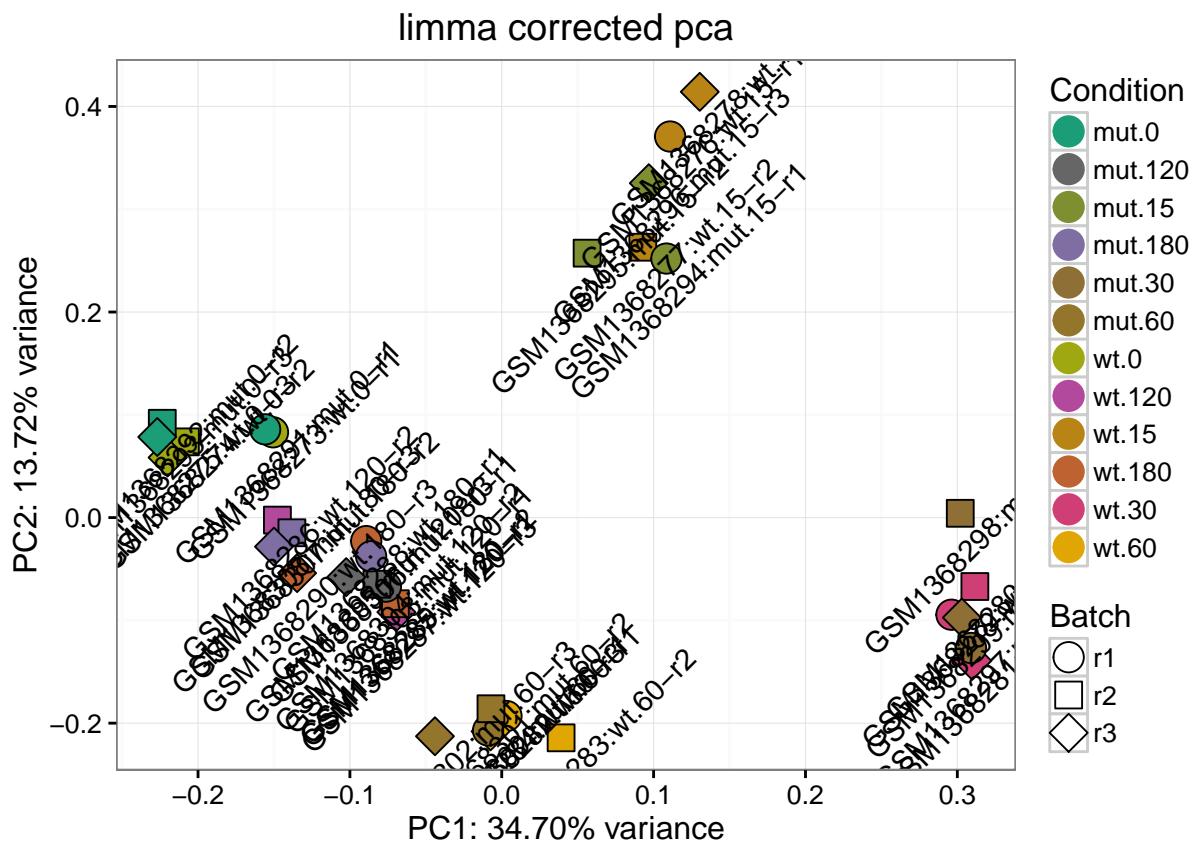
## batch_counts: Using limma's removeBatchEffect to remove batch effect.

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

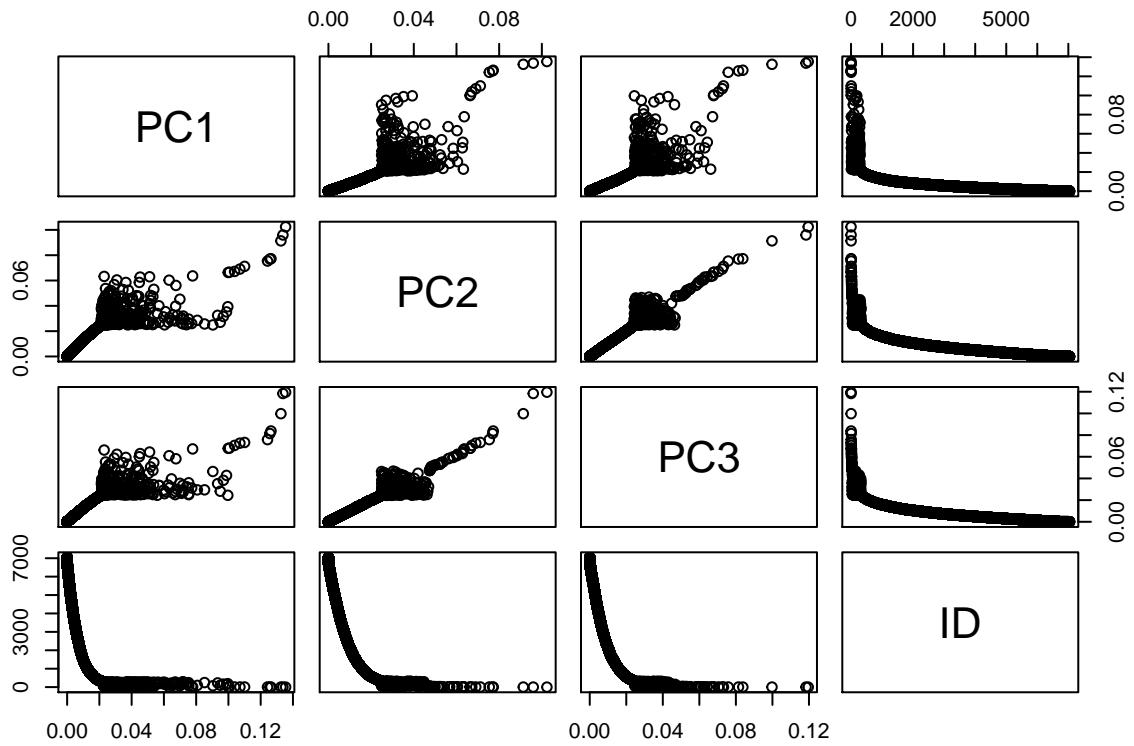
## transform_counts: Found 1689 values equal to 0, adding 0.5
## to the matrix.

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

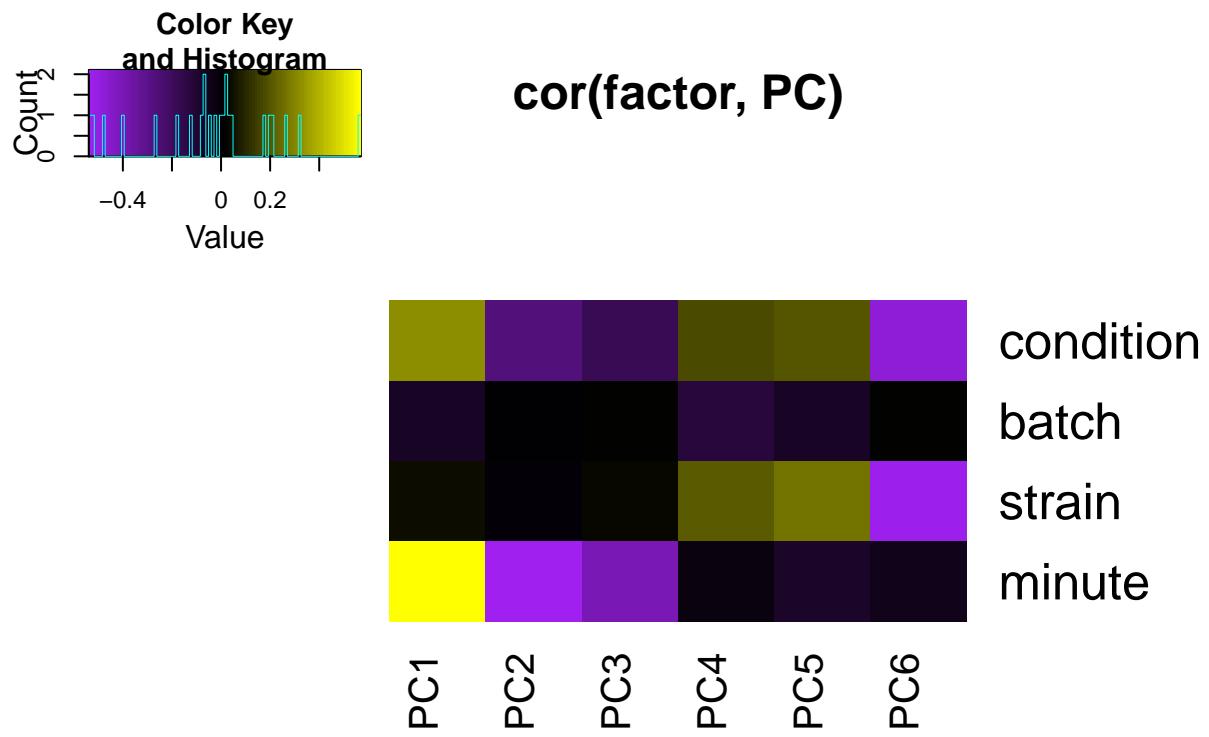
```

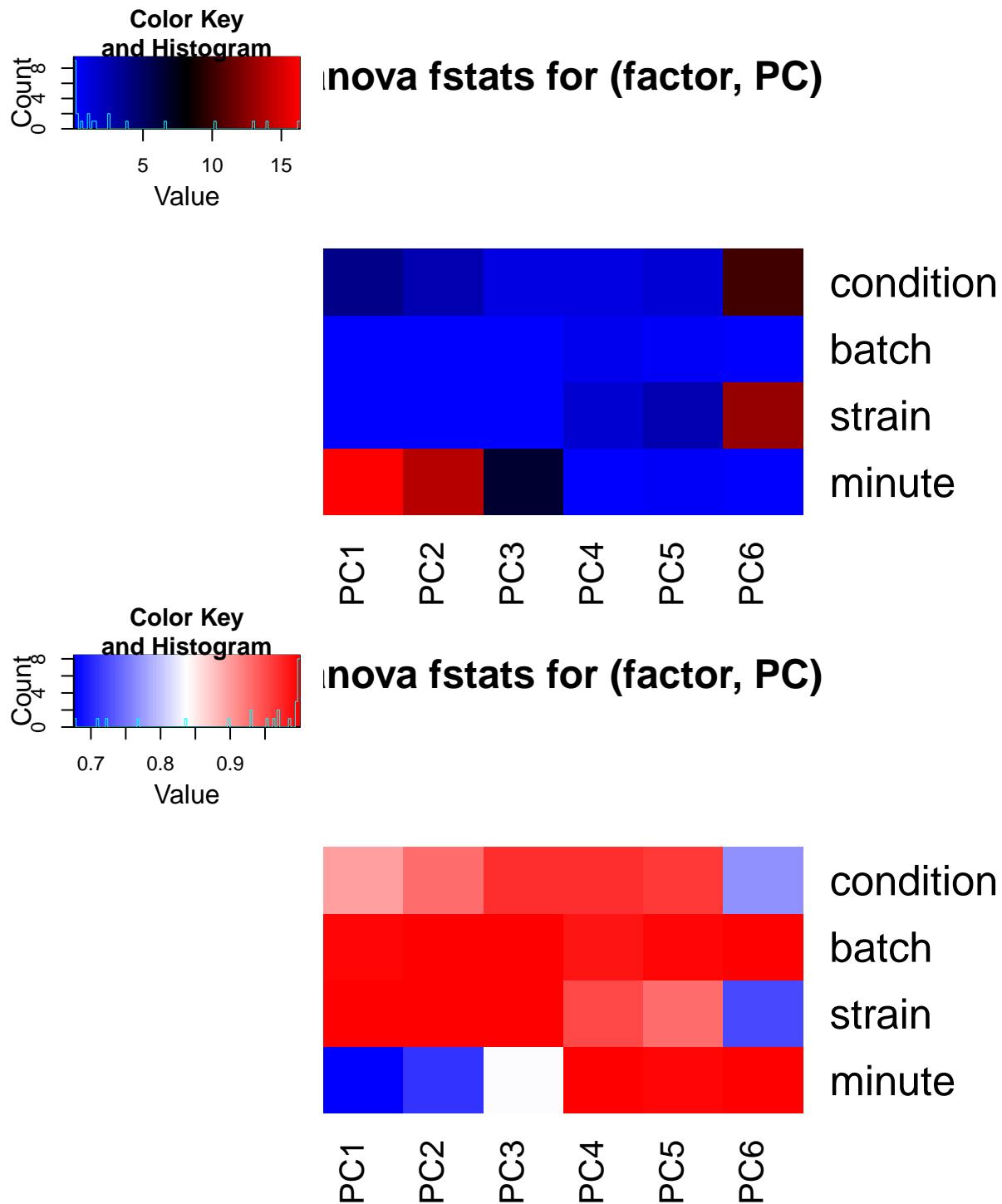


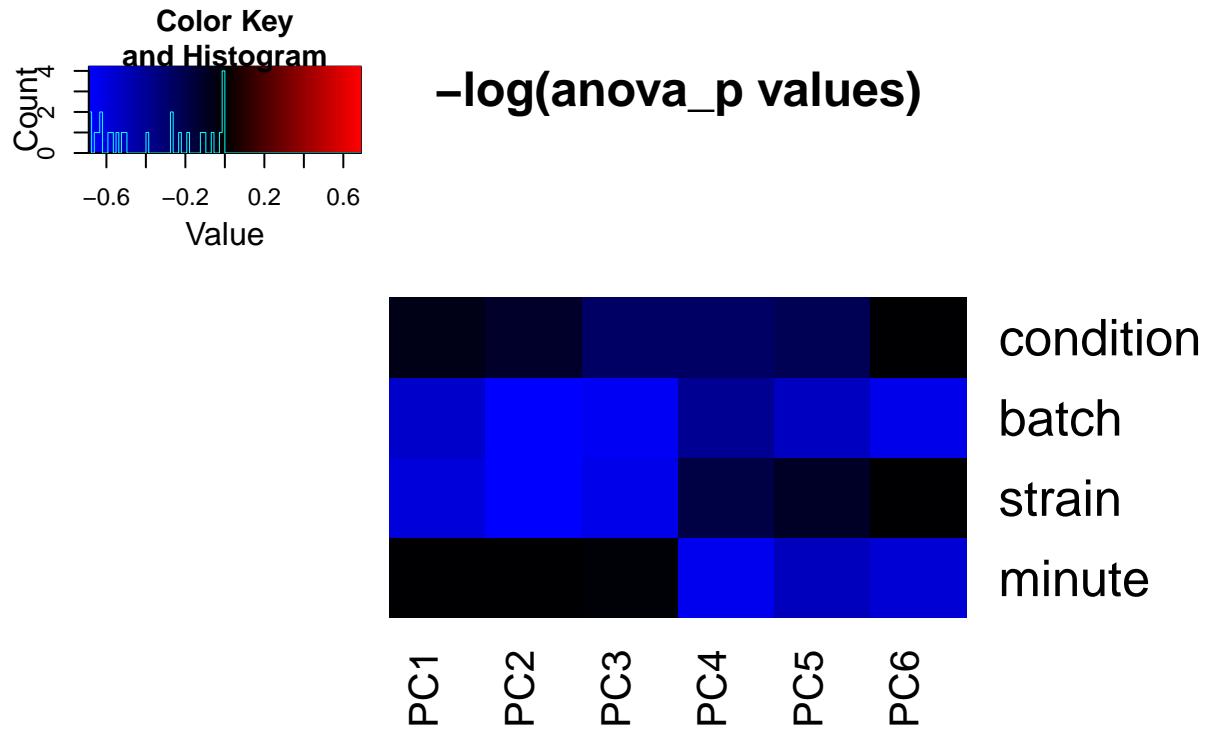
```
## The more shallow the curves in these plots, the more genes responsible for this principle component.
```



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







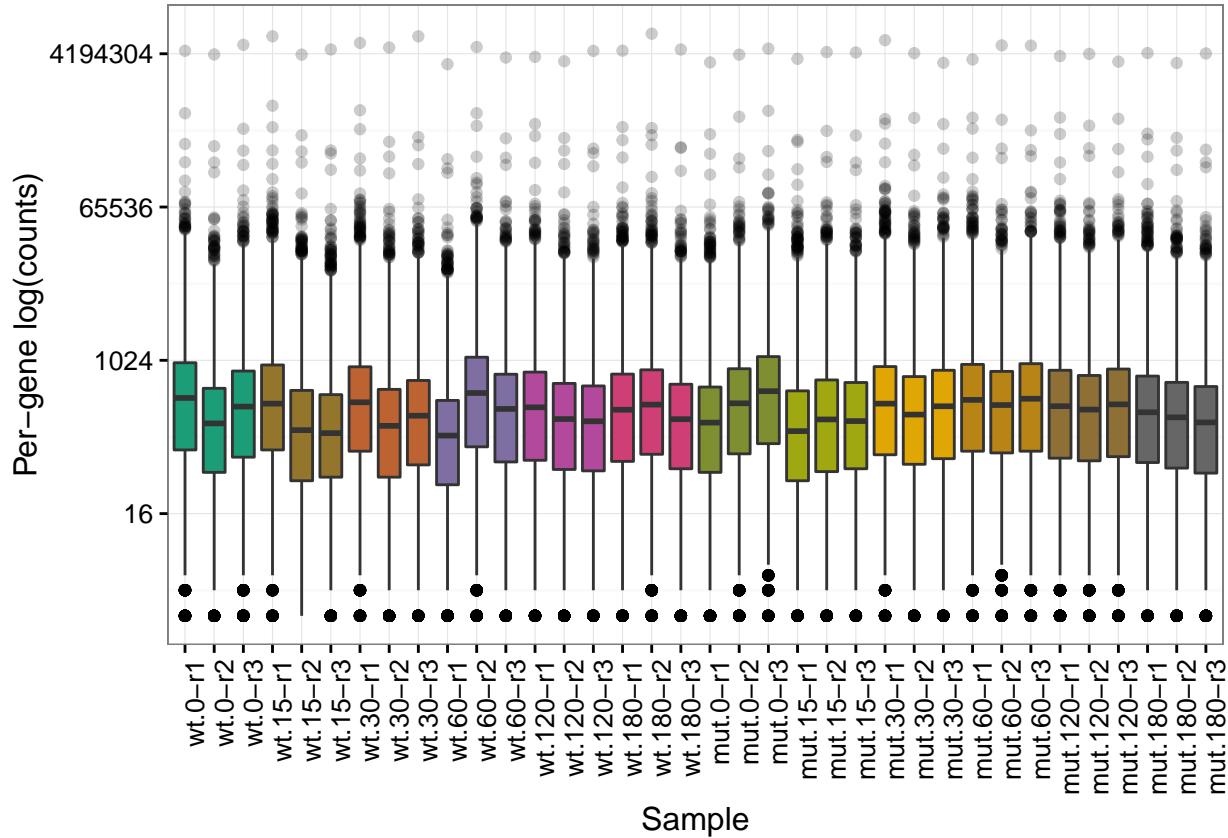
Interesting, the batch normalized pca plot looks much the same as the normalized. The variances are in fact pretty much the exact same...

## 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).

```

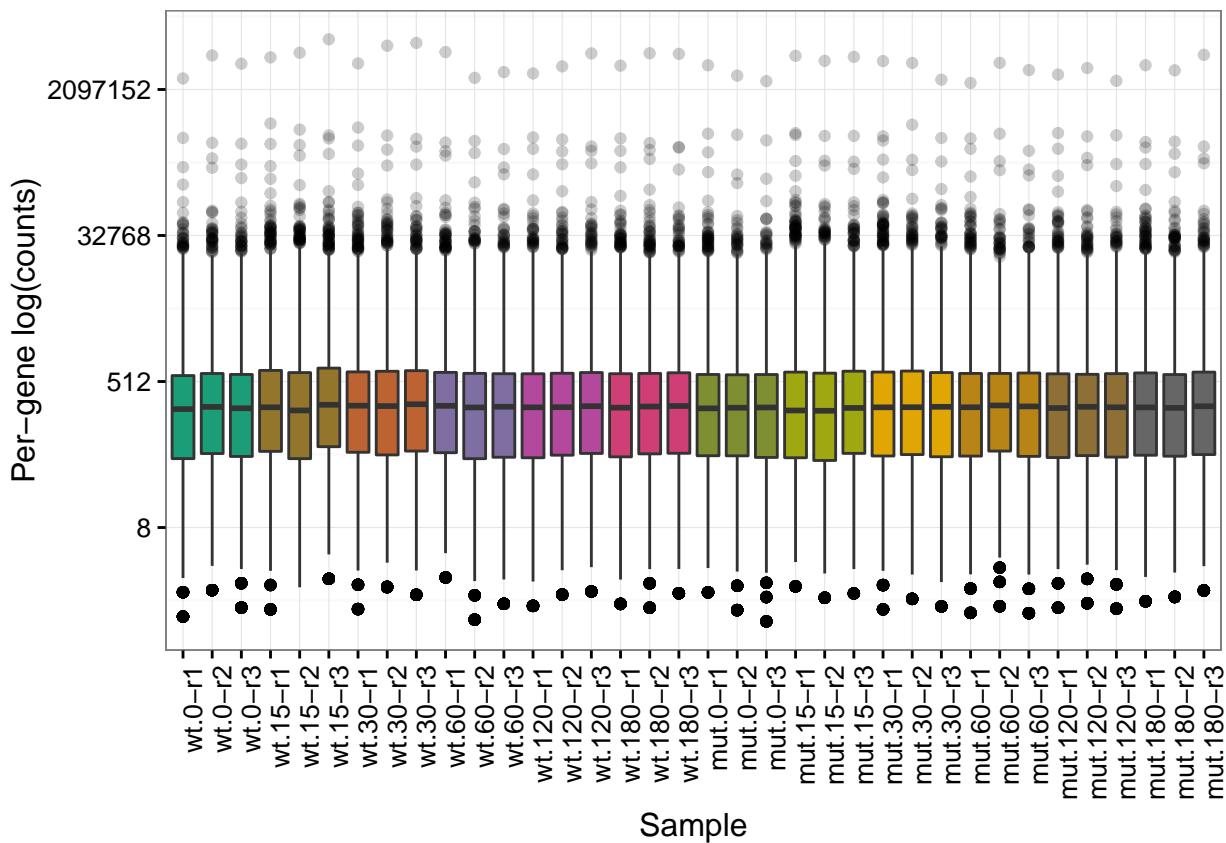
```
## Not correcting the count-data for batch effects. If batch is
## included in EdgerR/limma's model, then this is probably wise; but in extreme
## batch effects this is a good parameter to play with.
```

```
## Warning: replacing previous import 'S4Vectors::Position' 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).

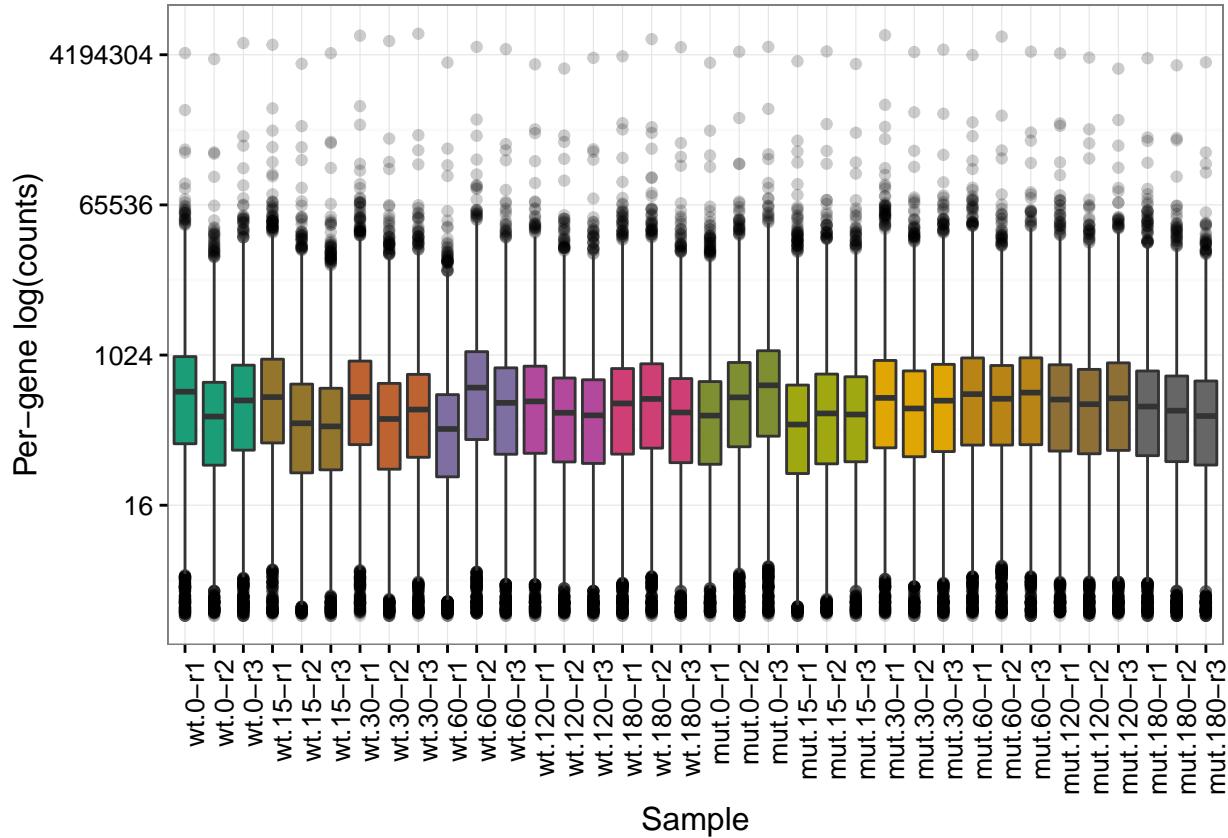
## Not correcting the count-data for batch effects. If batch is
## included in EdgerR/limma's model, then this is probably wise; but in extreme
## batch effects this is a good parameter to play with.

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.

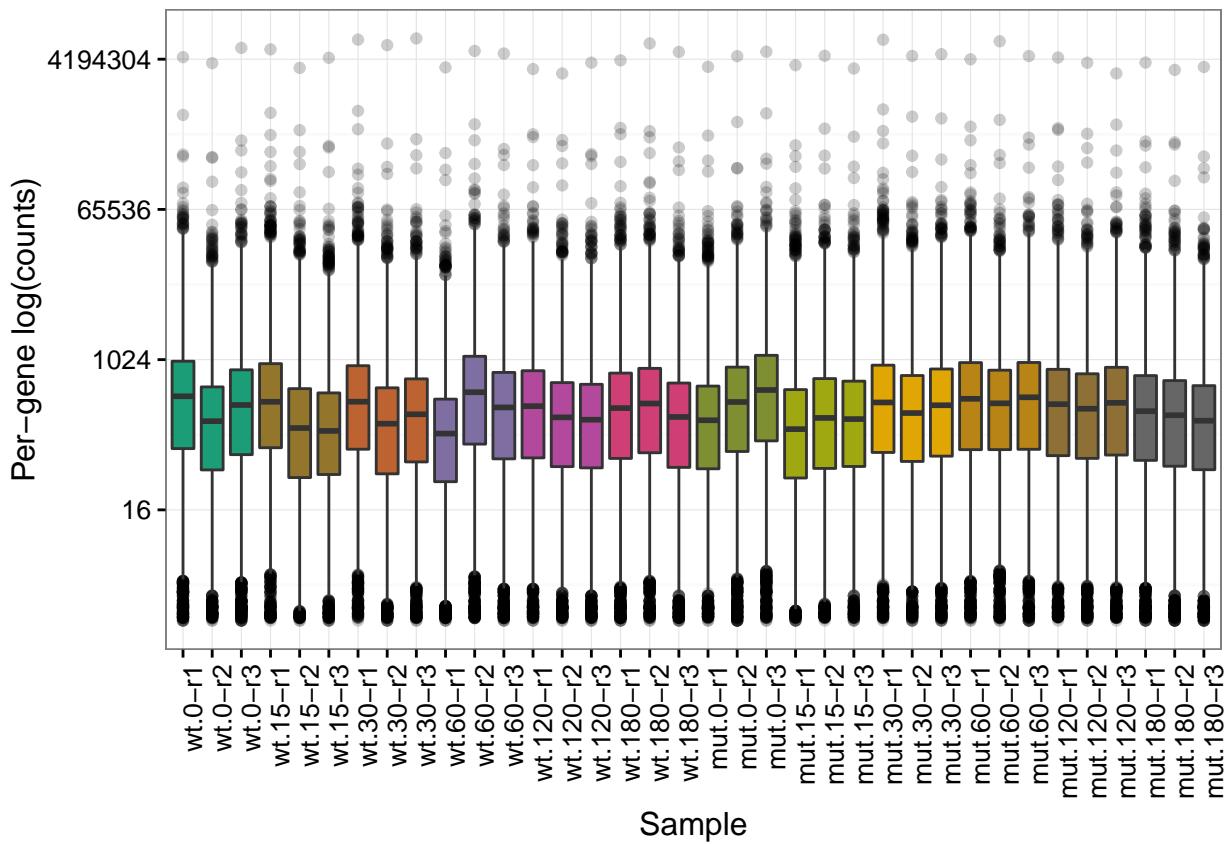
## Leaving the data unconverted. It is often advisable to cpm/rpk
##   the data to normalize for sampling differences, keep in mind though that rpk
##   has some annoying biases, and voom() by default does a cpm (though hpgl_voom()
##   will try to detect this).
```

```
## 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.

```

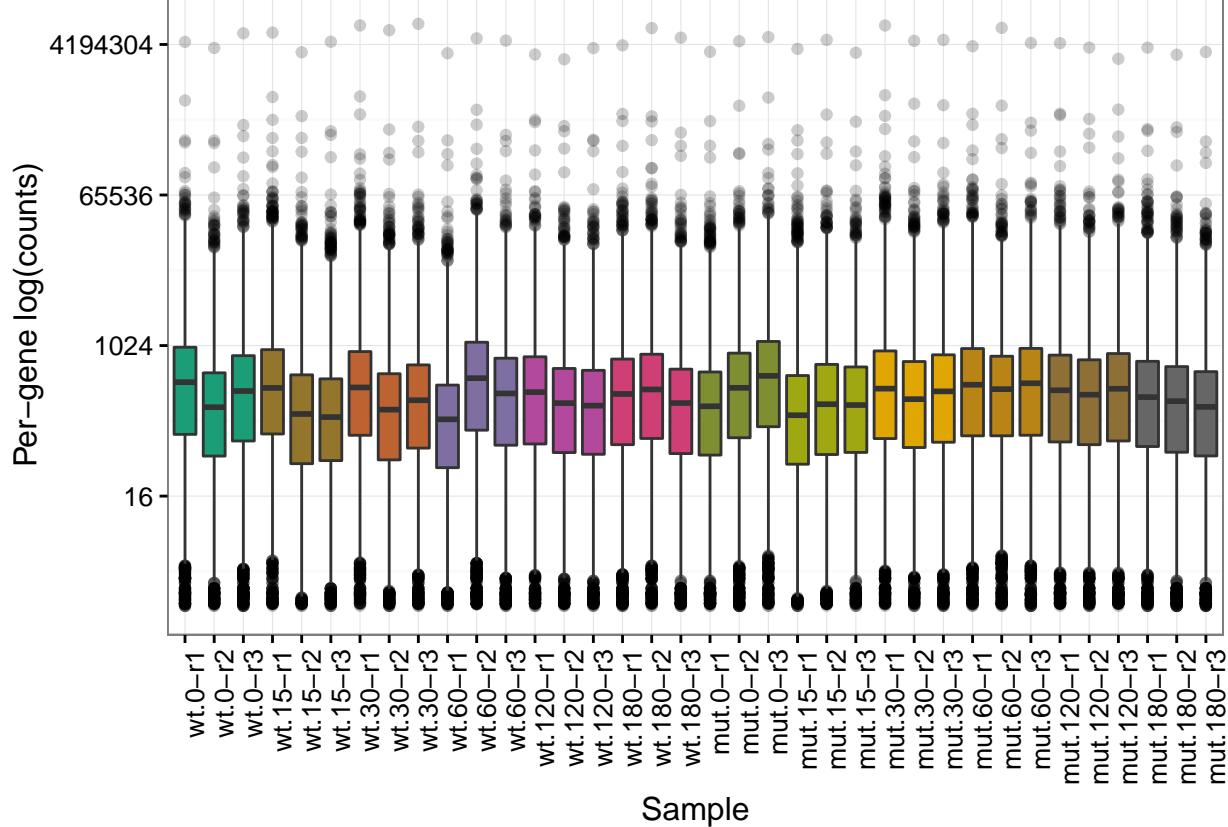
```
hpgl_boxplot(up_expt)
```

```

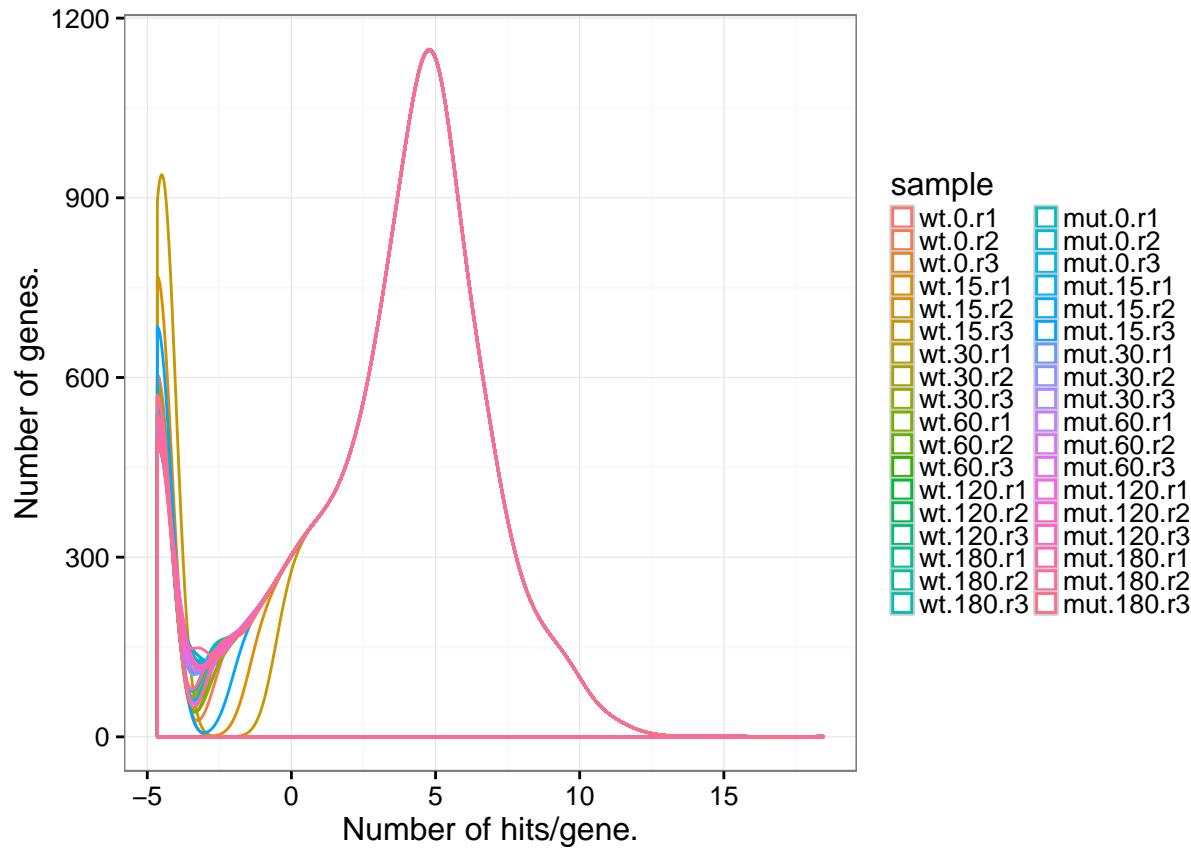
## I am reasonably sure this should be log scaled and am setting it.
## If this is incorrect, set scale='raw'

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

```

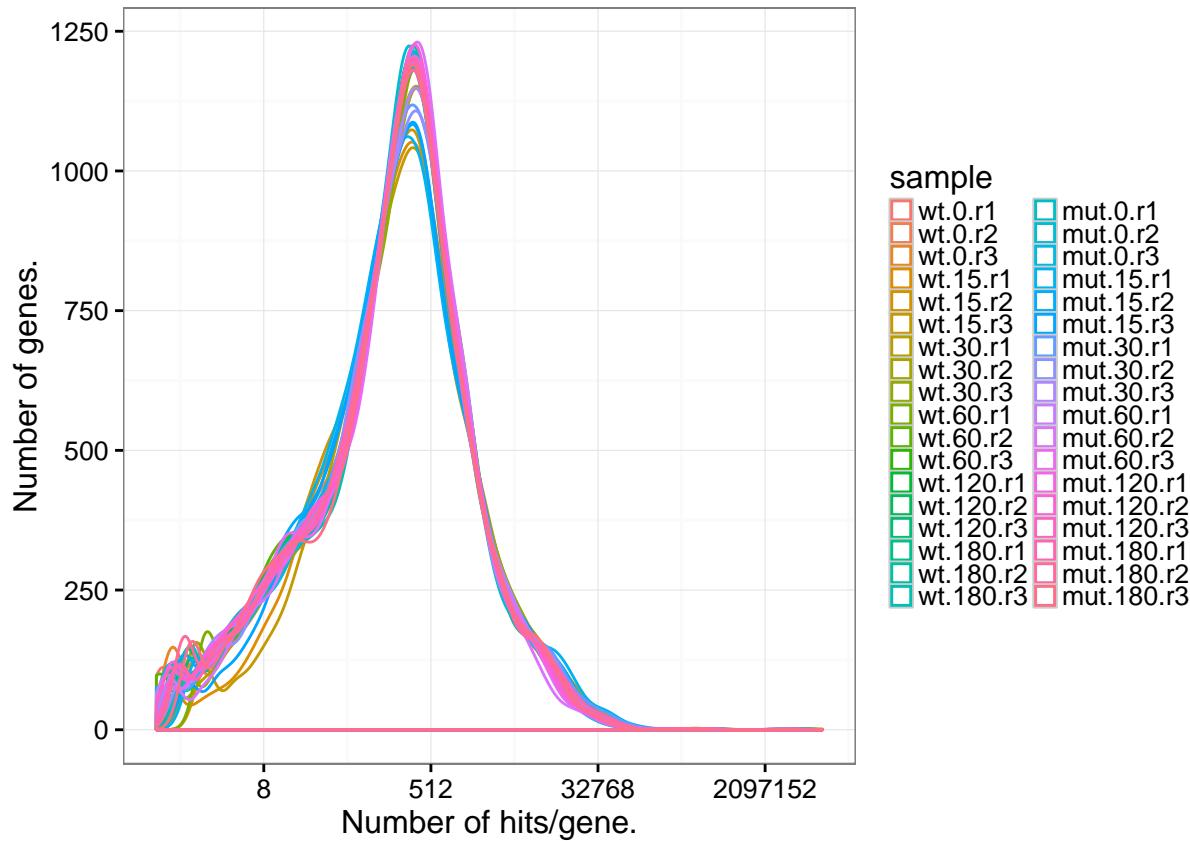


```
hpgl_density(norm_expt)
```



```
hpgl_density(sf_expt)
```

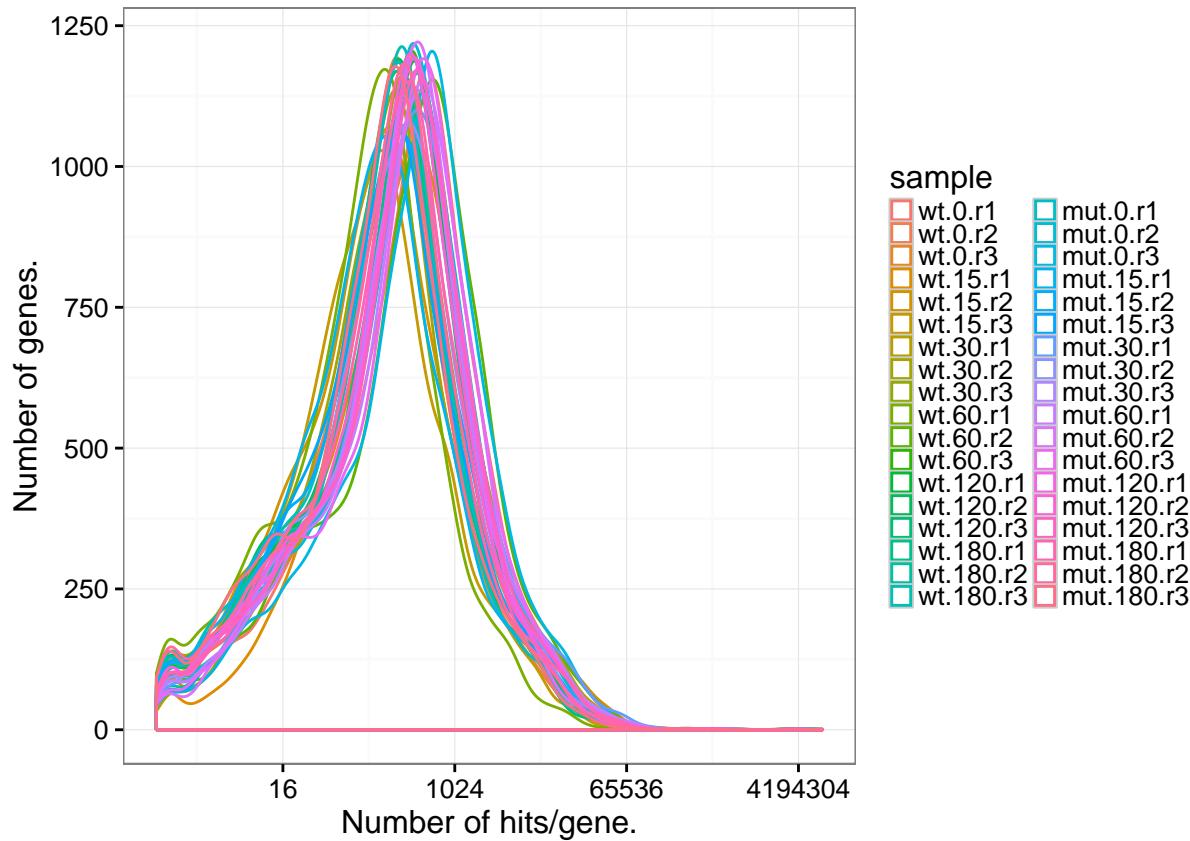
```
## This data will benefit from being displayed on the log scale.  
## If this is not desired, set scale='raw'  
## Warning: Removed 24130 rows containing non-finite values (stat_density).
```



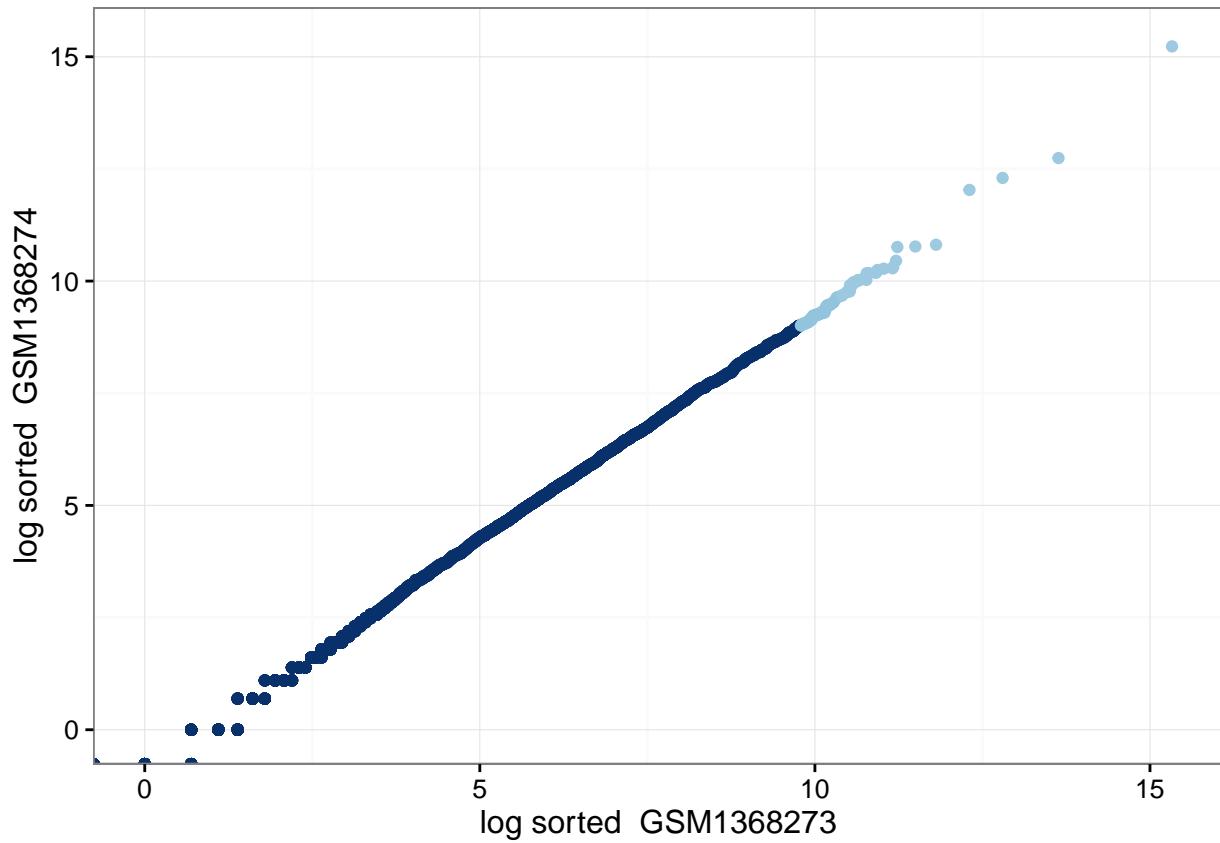
```
hpgl_density(tm_expt)
```

```
## 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).
```



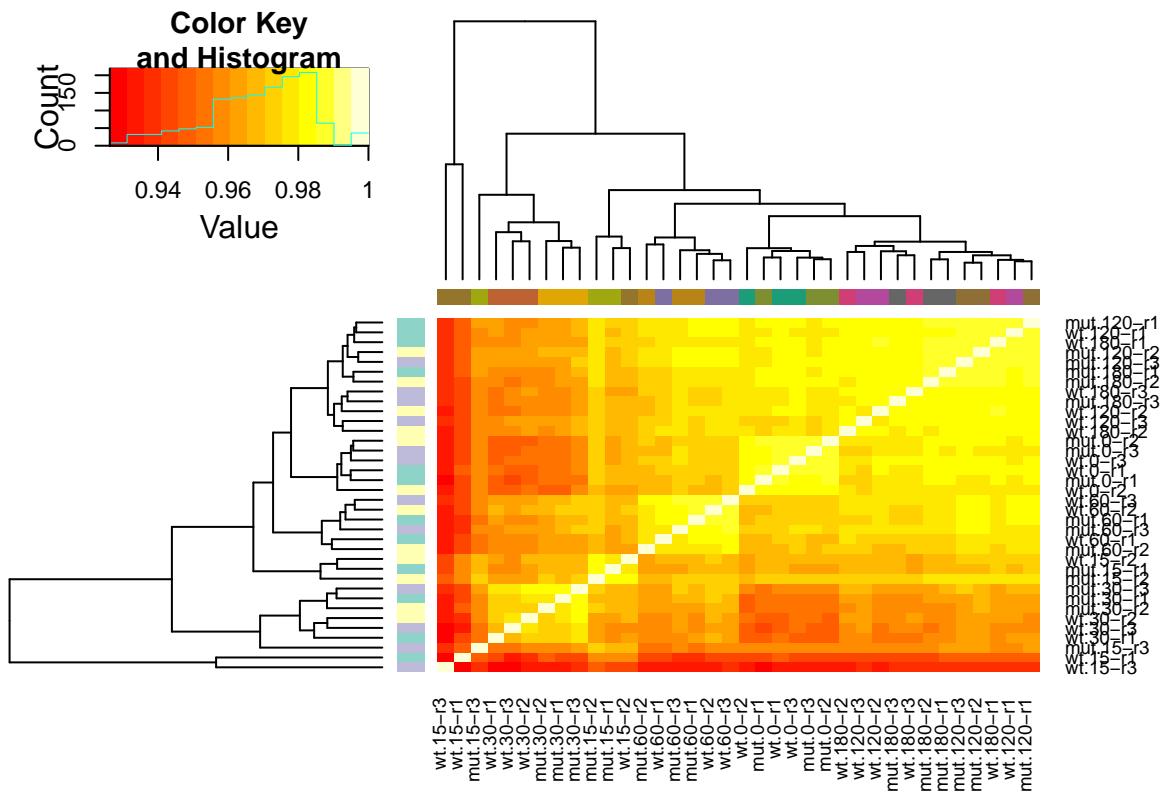
```
compare_12 <- hpgl_qq_plot(fission_expt, x=1, y=2)
compare_12$log
```



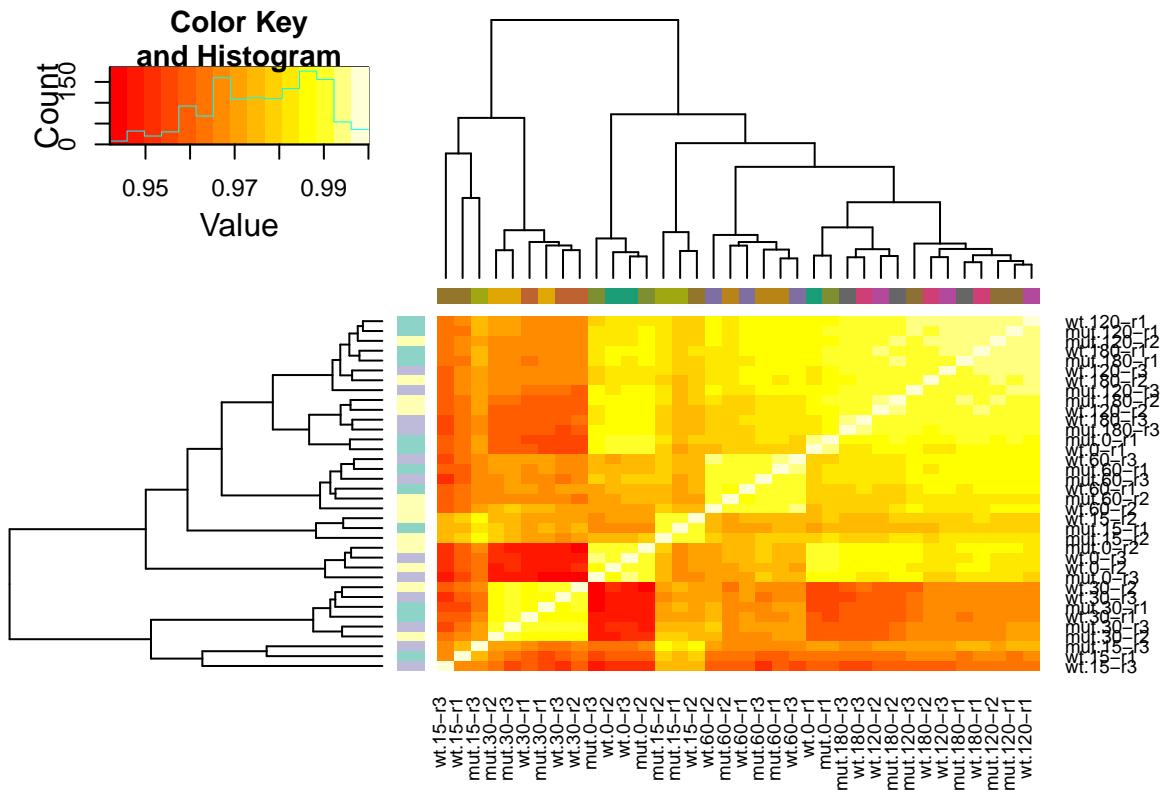
See how they cluster

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

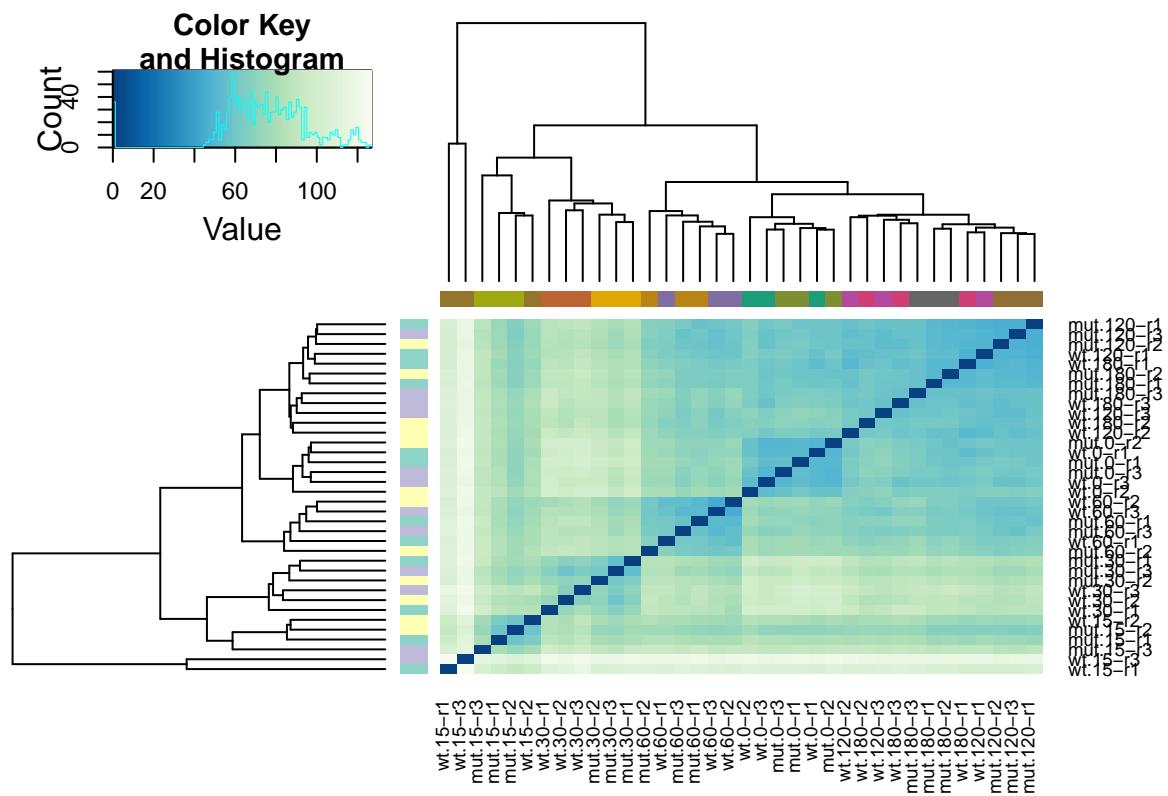
```
hpgl_corheat(norm_expt)
```



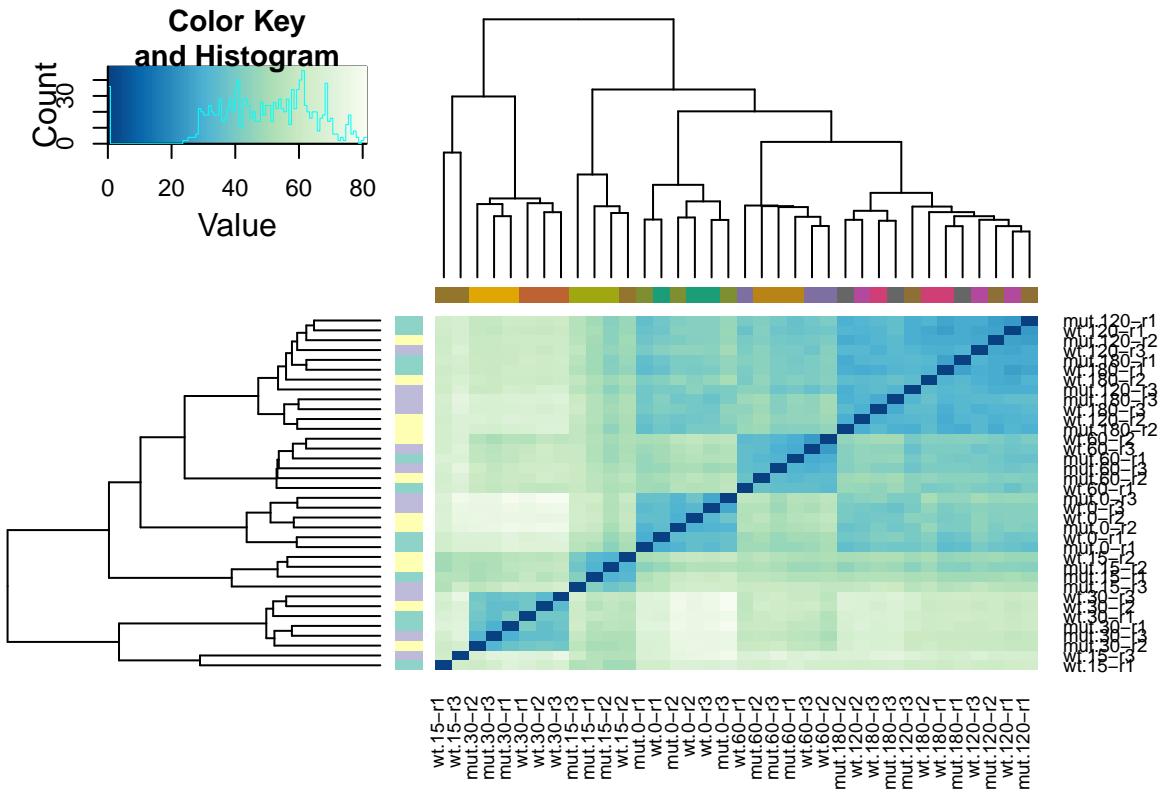
`hpgl_corheat(batchnorm_expt)`



**hpgl\_disheat**(norm\_expt)



`hpgl_disheat(batchnorm_expt)`



## Some simple differential expression analyses

Travis wisely imposes a limit on the amount of time for building vignettes. My tools by default will attempt all possible pairwise comparisons, which takes a long time. Therefore I am going to take a subset of the data and limit these comparisons to that.

```
fun_data <- expt_subset(fission_expt, subset="condition=='wt.120' | condition=='mut.120'")
fun_norm <- normalize_expt(fun_data, batch="limma", norm="quant", transform="log2", convert="cpm")

## 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, 7600 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: 175

## transform_counts: Found 175 values equal to 0, adding 0.5
## to the matrix.

```

## Try using limma first

```

limma_comparison <- limma_pairwise(fun_norm)

## 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.120 = mut.120,

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

## Limma step 5/6: Running eBayes and topTable.

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

## limma step 6/6: 1/3: Printing table: mut.120.

## limma step 6/6: 2/3: Printing table: wt.120.

## limma step 6/6: 3/3: Printing table: wt.120_vs_mut.120.

names(limma_comparison$all_tables)

## [1] "mut.120"           "wt.120"            "wt.120_vs_mut.120"

```

```

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

##      logFC          AveExpr          t
##  Min. :-1.5910000  Min. :-0.8809  Min. :-31.41000
##  1st Qu.:-0.0958500 1st Qu.: 1.6650  1st Qu.: -1.33750
##  Median : 0.0000002 Median : 4.2660  Median :  0.00002
##  Mean   : 0.0052609 Mean   : 3.8557  Mean   : -0.15042
##  3rd Qu.: 0.0976950 3rd Qu.: 5.7225  3rd Qu.:  1.14750
##  Max.   : 2.1010000 Max.   :18.3500  Max.   : 36.59000
##      P.Value        adj.P.Val         B       qvalue
##  Length:7039      Length:7039      Min. :-7.571  Length:7039
##  Class :character Class :character  1st Qu.:-7.067  Class :character
##  Mode  :character Mode  :character  Median :-6.421  Mode  :character
##                                         Mean  :-5.701
##                                         3rd Qu.:-5.029
##                                         Max.   : 6.493

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=NULL)

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

## mut.120wt.120wt.120_vs_mut.120

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

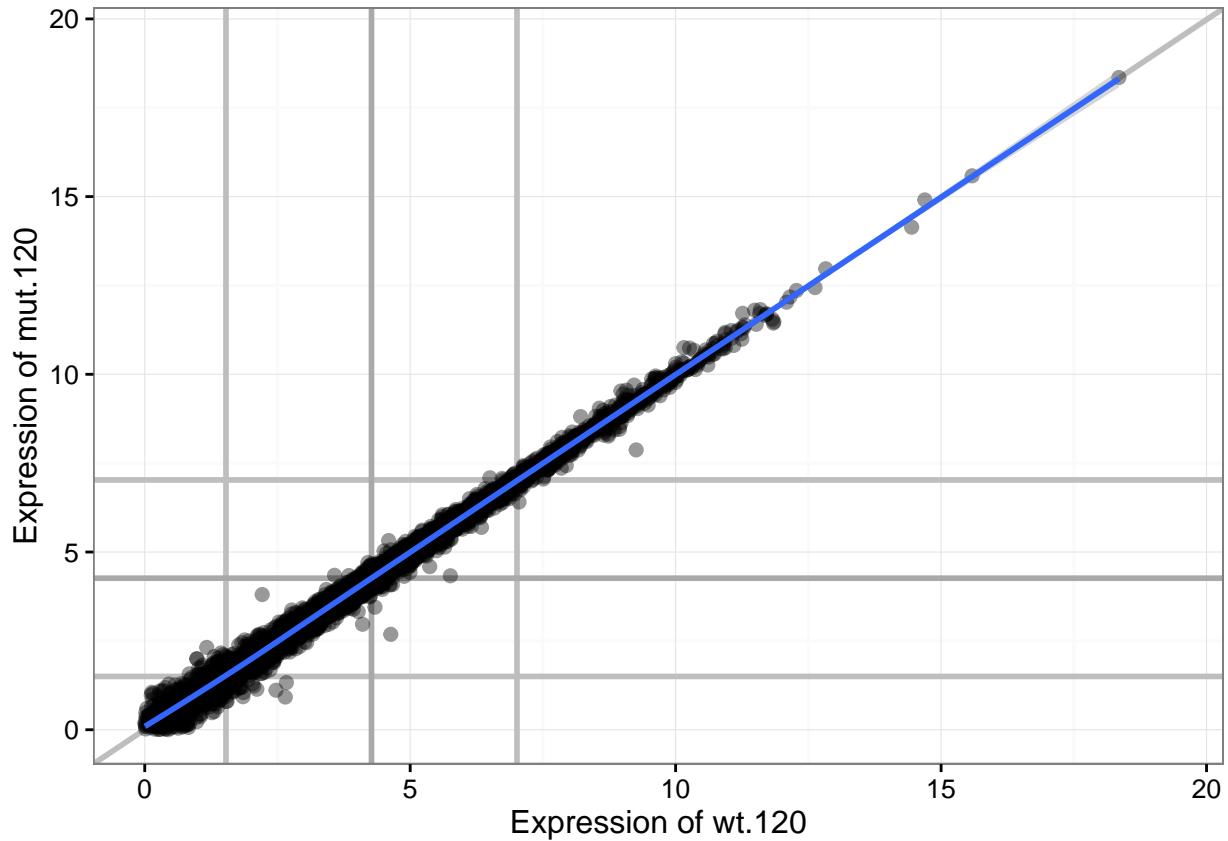
## Setting binwidth to 0.0384765861670427 in order to have 500 bins.

scatter_wt_mut$scatter

## Warning: Removed 1069 rows containing non-finite values (stat_smooth).

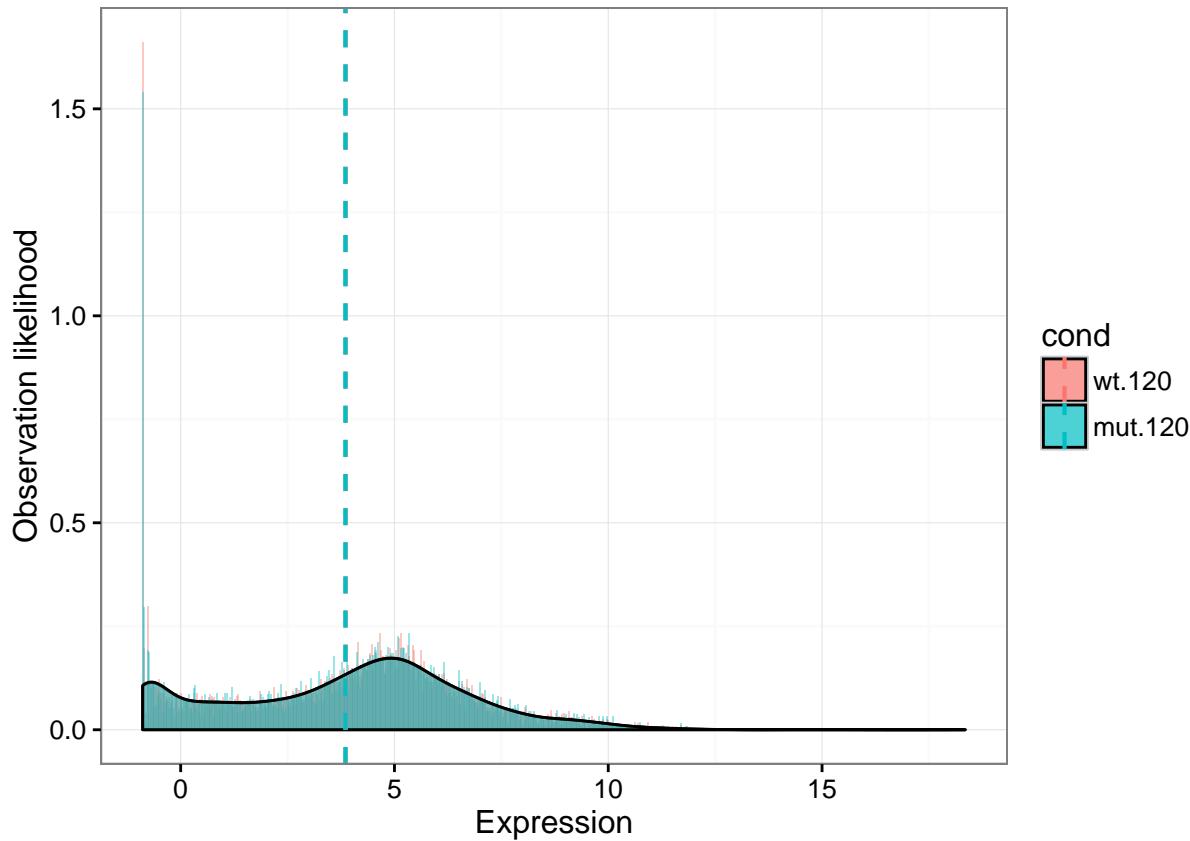
## Warning: Removed 1069 rows containing missing values (geom_point).

```



```
scatter_wt_mut$both_histogram
```

```
## $plot
```



```

## 
## $data_summary
##      wt.120          mut.120
##  Min. :-0.8833  Min. :-0.8883
##  1st Qu.: 1.6569  1st Qu.: 1.6557
##  Median : 4.2711  Median : 4.2644
##  Mean   : 3.8583  Mean   : 3.8531
##  3rd Qu.: 5.7084  3rd Qu.: 5.7203
##  Max.   :18.3500  Max.   :18.3500
## 
## $uncor_t
## 
##  Pairwise comparisons using t tests with pooled SD
## 
## data: play_all$expression and play_all$cond
## 
##      wt.120
## mut.120 0.91
## 
## 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 0.91
##
## P value adjustment method: bonferroni

```

## Then DESeq2

```

deseq_comparison <- deseq2_pairwise(fun_data, model_batch=TRUE)

## Starting DESeq2 pairwise comparisons.

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

## factor levels were dropped which had no samples

## DESeq2 step 2/5: Estimate size factors.

## DESeq2 step 3/5: estimate Dispersions.

## DESeq2 step 4/5: nbinomWaldTest.

## DESeq2 step 5/5: 1/1: Printing table: wt.120_vs_mut.120

## Collected coefficients for: mut.120

## Collected coefficients for: wt.120

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

##      baseMean        logFC        lfcSE        stat
## Min. :     0  Min. :-0.4275  Min. :0.0115  Min. :-4.2710
## 1st Qu.:   30  1st Qu.:-0.0436  1st Qu.:0.0982  1st Qu.:-0.4824
## Median :  215  Median :-0.0001  Median :0.1089  Median :-0.0026
## Mean   : 1645  Mean   : 0.0036  Mean   :0.0985  Mean   : 0.0299
## 3rd Qu.:  597  3rd Qu.: 0.0480  3rd Qu.:0.1154  3rd Qu.: 0.5072
## Max.   :3976000  Max.   : 0.5620  Max.   :0.1182  Max.   : 4.7570
##                  NA's   :387    NA's   :387    NA's   :387
##      P.Value       adj.P.Val       qvalue
## Min. :0.000002  Min. :0.006838  Min. :0.01383
## 1st Qu.:0.406350 1st Qu.:1.000000 1st Qu.:1.000000
## Median :0.646500  Median :1.000000  Median :1.000000
## Mean   :0.615159  Mean   :0.982937  Mean   :0.99856
## 3rd Qu.:0.854750 3rd Qu.:1.000000 3rd Qu.:1.000000
## Max.   :1.000000  Max.   :1.000000  Max.   :1.000000
##

```

## And EdgeR

```

edger_comparison <- edger_pairwise(fun_data, 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.120 = mut.120,

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

## EdgeR step 9/9: 1/1: Printing table: wt.120_vs_mut.120.

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

```

```

##      logFC          logCPM          LR        PValue
##  Min. :-4.72200   Min. :-2.554   Min. : 0.00000  Length:7039
##  1st Qu.:-0.12195  1st Qu.: 1.494   1st Qu.: 0.06314  Class :character
##  Median : 0.00000   Median : 4.234   Median : 0.36910  Mode  :character
##  Mean   :-0.03085   Mean   : 3.599   Mean   : 1.01156
##  3rd Qu.: 0.09977   3rd Qu.: 5.700   3rd Qu.: 1.26400
##  Max.   : 5.20200   Max.   :18.400   Max.   :48.90000
##      FDR          qvalue
##  Length:7039       Length:7039
##  Class :character  Class :character
##  Mode  :character  Mode  :character
##
## 
## 
## 
```

## My stupid basic comparison

```

basic_comparison <- basic_pairwise(fun_data)

## Starting basic pairwise comparison.

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

## Basic step 2/3: Performing comparisons.

## Basic step 2/3: 1/1: Performing log2 subtraction: wt.120_vs_mut.120

## 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

```

## Combine them all

```

all_comparisons <- all_pairwise(fun_data, 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.120 = mut.120,

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

## Limma step 5/6: Running eBayes and topTable.

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

## limma step 6/6: 1/3: Printing table: mut.120.

## limma step 6/6: 2/3: Printing table: wt.120.

## limma step 6/6: 3/3: Printing table: wt.120_vs_mut.120.

## Starting DESeq2 pairwise comparisons.

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

## factor levels were dropped which had no samples

## DESeq2 step 2/5: Estimate size factors.

## DESeq2 step 3/5: estimate Dispersions.

## DESeq2 step 4/5: nbinomWaldTest.

## DESeq2 step 5/5: 1/1: Printing table: wt.120_vs_mut.120

## Collected coefficients for: mut.120

## Collected coefficients for: wt.120

## 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.120 = mut.120,

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

## EdgeR step 9/9: 1/1: Printing table: wt.120_vs_mut.120.

## Starting basic pairwise comparison.

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

## Basic step 2/3: Performing comparisons.

## Basic step 2/3: 1/1: Performing log2 subtraction: wt.120_vs_mut.120

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

## Basic: Returning tables.

## 1/1: Comparing analyses: wt.120_vs_mut.120

all_combined <- combine_de_tables(all_comparisons)

## Working on table 1/1: wt.120_vs_mut.120

## The table is: wt.120_vs_mut.120

sig_genes <- extract_significant_genes(all_combined, excel=NULL)

## Writing excel data sheet 1/1

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

## After (adj)p filter, the up genes table has 7 genes.

## After (adj)p filter, the down genes table has 8 genes.

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

## After fold change filter, the up genes table has 7 genes.

## After fold change filter, the down genes table has 8 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.120"           "wt.120"           "wt.120_vs_mut.120"

table <- limma_results$wt.120_vs_mut.120
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 1288 genes.

## After z filter, the down genes table has 1214 genes.

##orthologs <- read.table("ftp://ftp.ebi.ac.uk/pub/databases/pombase/orthologs/cerevisiae-orthologs")
##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 <- biomaRt::getBM(attributes=c('pombase_gene_name', 'go_accession'), values=gene_names, mart=biomaRt, columnNames=pombe_goids) <- c("ID", "GO")
pombe <- GenomicFeatures::makeTxDbFromBiomart(biomart ="fungal_mart", dataset = "spombe_eg_gene", host=""

## 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(GenomicFeatures::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

## 

## Warning in pcls(G): initial point very close to some inequality constraints

## Using manually entered categories.

## Calculating the p-values...

## simple_goseq(): Calculating q-values

## Loading required package: GO.db

## Loading required package: AnnotationDbi

## Loading required package: Biobase

```

```

## Welcome to Bioconductor
##
##      Vignettes contain introductory material; view with
##      'browseVignettes()'. To cite Bioconductor, see
##      'citation("Biobase")', and for packages 'citation("pkgname")'.

## simple_goseq(): Filling godata with terms, this is slow.

## Testing that go categories are defined.

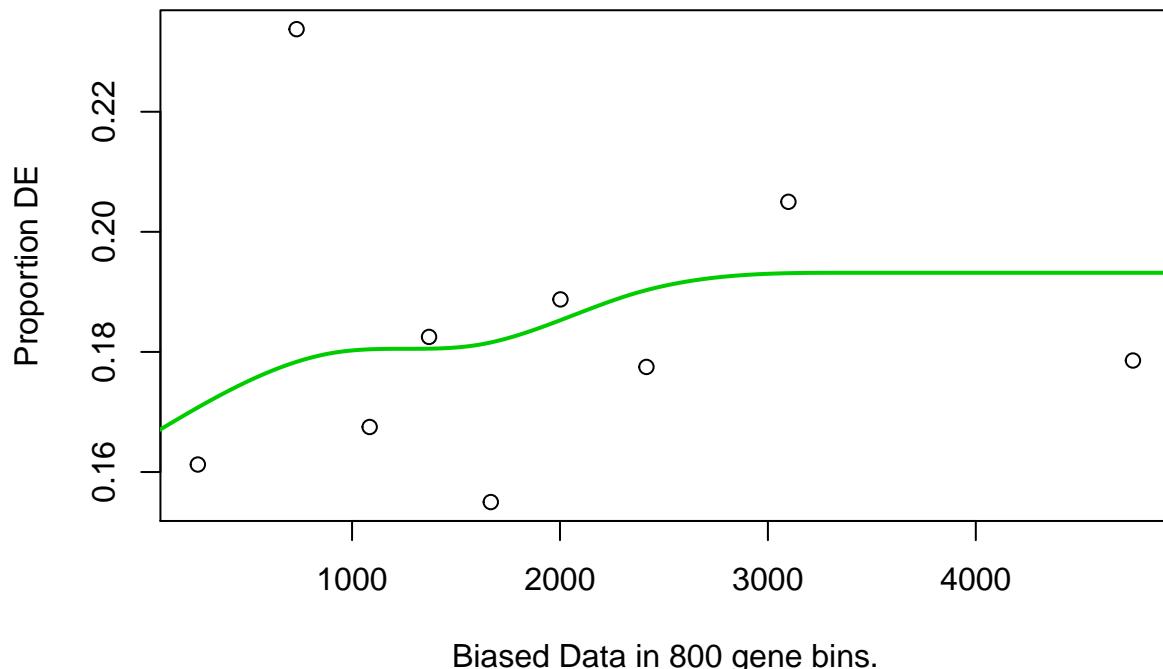
## Removing undefined categories.

## Gathering synonyms.

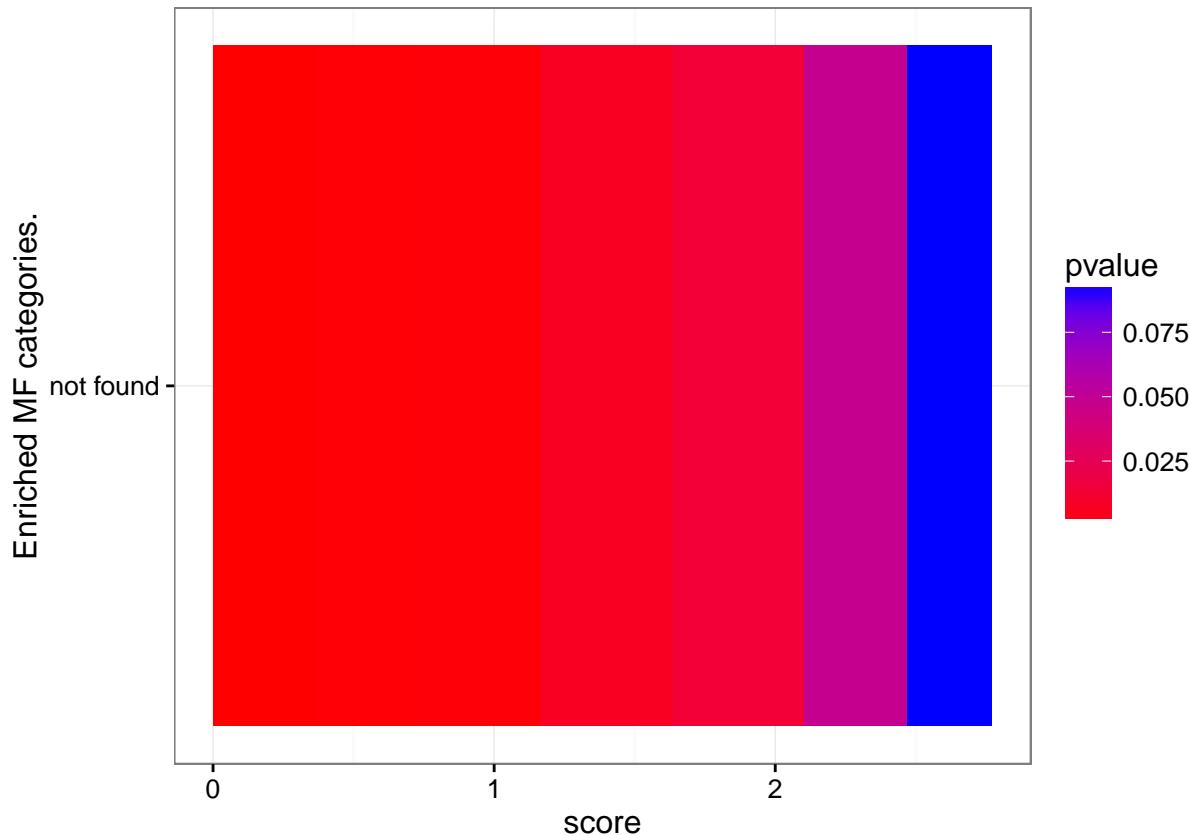
## Gathering category definitions.

## simple_goseq(): Making pvalue plots for the ontologies.

```



```
goseq_search$mfp_plot
```



```

##cluster_search = simple_clusterprofiler(de_genes=updown_genes$up_genes, goids=pombe_goids, gff="pombe
topgo_search = simple_topgo(de_genes=updown_genes$up_genes, goids_df=pombe_goids)

## Loading required namespace: topGO

## Loading required package: graph

## Loading required package: SparseM

##
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
##
##      backsolve

##
## groupGOTerms:      GOBPTerm, GOMFTerm, GOCCTerm environments built.

##
## Attaching package: 'topGO'

## The following object is masked from 'package:IRanges':
##
##      members

```

## Attempting to generate a id2go file in the format expected by topGO.

```

## simple_topgo(): Found ID->GO map: c("SPAC1002.01", "SPAC1002.02", "SPAC1002.03c", "SPAC1002.04c", "SPAC1002.05c", "SPAC1039.05c", "SPAC1039.06", "SPAC1039.07c", "SPAC1039.08", "SPAC1039.09", "SPAC1039.10", "SPAC1039.11", "SPAC10F6.04", "SPAC10F6.05c", "SPAC10F6.06", "SPAC10F6.07c", "SPAC10F6.08c", "SPAC10F6.09c", "SPAC10F6.10c", "SPAC11D3.06", "SPAC11D3.07c", "SPAC11D3.08c", "SPAC11D3.09", "SPAC11D3.10", "SPAC11D3.11c", "SPAC11D3.12c", "SPAC11G7.06c", "SPAC11H11.01", "SPAC11H11.02c", "SPAC11H11.03c", "SPAC11H11.04", "SPAC11H11.05c", "SPAC11H11.06c", "SPAC12B10.14c", "SPAC12B10.15c", "SPAC12B10.16c", "SPAC12G12.01c", "SPAC12G12.02", "SPAC12G12.03", "SPAC12G12.04c", "SPAC1399.06", "SPAC13A11.01c", "SPAC13A11.02c", "SPAC13A11.03", "SPAC13A11.04c", "SPAC13A11.05", "SPAC13A11.06c", "SPAC13G6.06c", "SPAC13G6.07c", "SPAC13G6.08", "SPAC13G6.09", "SPAC13G6.10c", "SPAC13G6.11c", "SPAC13G6.12c", "SPAC144.03", "SPAC144.04c", "SPAC144.05", "SPAC144.06", "SPAC144.07c", "SPAC144.08", "SPAC144.09c", "SPAC144.10c", "SPAC14C4.08", "SPAC14C4.09", "SPAC14C4.10c", "SPAC14C4.11", "SPAC14C4.12c", "SPAC14C4.13", "SPAC14C4.14", "SPAC15A10.06", "SPAC15A10.07", "SPAC15A10.08", "SPAC15A10.09c", "SPAC15A10.10", "SPAC15A10.11", "SPAC15A10.12c", "SPAC1639.01c", "SPAC1639.02c", "SPAC167.01", "SPAC167.02", "SPAC167.03c", "SPAC167.04", "SPAC167.05c", "SPAC167.06c", "SPAC1687.23c", "SPAC16A10.01", "SPAC16A10.02", "SPAC16A10.03c", "SPAC16A10.04", "SPAC16A10.05c", "SPAC16A10.06c", "SPAC16E8.17c", "SPAC16E8.18", "SPAC16E8.18c", "SPAC1705.02", "SPAC1705.03c", "SPAC1751.01c", "SPAC1751.02c", "SPAC17A2.01", "SPAC17A2.02c", "SPAC17A2.03c", "SPAC17A2.04c", "SPAC17A2.05", "SPAC17A2.06c", "SPAC17A2.07c", "SPAC17C9.02c", "SPAC17C9.03", "SPAC17C9.05c", "SPAC17C9.06", "SPAC17C9.07", "SPAC17C9.08", "SPAC17C9.09c", "SPAC17G6.17", "SPAC17G8.01c", "SPAC17G8.02", "SPAC17G8.03c", "SPAC17G8.04c", "SPAC17G8.05", "SPAC17G8.06c", "SPAC17H9.19c", "SPAC17H9.20", "SPAC1805.01c", "SPAC1805.02c", "SPAC1805.03c", "SPAC1805.04", "SPAC1805.05c", "SPAC1851.03", "SPAC1851.04c", "SPAC186.01", "SPAC186.02c", "SPAC186.03", "SPAC186.04c", "SPAC186.05c", "SPAC186.06c", "SPAC18G6.14c", "SPAC18G6.15", "SPAC1952.01", "SPAC1952.02", "SPAC1952.03", "SPAC1952.04c", "SPAC1952.05c", "SPAC1952.06c", "SPAC19A8.15", "SPAC19A8.16", "SPAC19B12.01", "SPAC19B12.02c", "SPAC19B12.03", "SPAC19B12.04", "SPAC19B12.05c", "SPAC19G12.05", "SPAC19G12.06c", "SPAC19G12.07c", "SPAC19G12.08", "SPAC19G12.09", "SPAC19G12.10c", "SPAC19G12.11c", "SPAC1B3.01c", "SPAC1B3.02c", "SPAC1B3.03c", "SPAC1B3.04c", "SPAC1B3.05", "SPAC1B3.06c", "SPAC1B3.07c", "SPAC1D4.14", "SPAC1F12.02c", "SPAC1F12.04c", "SPAC1F12.05", "SPAC1F12.06c", "SPAC1F12.07", "SPAC1F12.08c", "SPAC1F7.06", "SPAC1F7.07c", "SPAC1F7.08", "SPAC1F7.09c", "SPAC1F7.10", "SPAC1F7.11c", "SPAC1F7.12", "SPAC1F7.13c", "SPAC20G8.09c", "SPAC20G8.10c", "SPAC20H4.01", "SPAC20H4.02", "SPAC20H4.03c", "SPAC20H4.04", "SPAC20H4.05c", "SPAC222.05c", "SPAC222.06", "SPAC222.07c", "SPAC222.08c", "SPAC222.09", "SPAC222.10c", "SPAC222.11c", "SPAC222.12c", "SPAC22A12.02c", "SPAC22A12.03c", "SPAC22A12.04c", "SPAC22A12.05", "SPAC22A12.06c", "SPAC22A12.07c", "SPAC22A12.08c", "SPAC22E12.18", "SPAC22E12.19", "SPAC22F3.02", "SPAC22F3.03c", "SPAC22F3.04", "SPAC22F3.05c", "SPAC22F3.06c", "SPAC22G7.08", "SPAC22G7.09c", "SPAC22G7.10", "SPAC22G7.11c", "SPAC22H10.02", "SPAC22H10.03c", "SPAC22H10.04c", "SPAC22H10.05c", "SPAC23A1.14c", "SPAC23A1.15c", "SPAC23A1.16c", "SPAC23A1.17", "SPAC23A1.18c", "SPAC23A1.19c", "SPAC23A1.20c", "SPAC23C4.10", "SPAC23C4.11", "SPAC23C4.12", "SPAC23C4.13", "SPAC23C4.14", "SPAC23C4.15", "SPAC23C4.16c", "SPAC23G3.05c", "SPAC23G3.06", "SPAC23G3.07c", "SPAC23G3.08c", "SPAC23G3.09", "SPAC23G3.10c", "SPAC23G3.11c", "SPAC23H4.12", "SPAC23H4.13c", "SPAC23H4.14", "SPAC23H4.15", "SPAC23H4.16c", "SPAC23H4.17c", "SPAC23H4.18c", "SPAC24C9.16c", "SPAC24H6.01c", "SPAC24H6.02c", "SPAC24H6.03", "SPAC24H6.04", "SPAC24H6.05", "SPAC24H6.06c", "SPAC25B8.17", "SPAC25B8.18", "SPAC25B8.19c", "SPAC25B8.20", "SPAC25G10.01", "SPAC25G10.02", "SPAC25G10.03c", "SPAC26A3.12c", "SPAC26A3.13c", "SPAC26A3.14c", "SPAC26A3.15c", "SPAC26A3.16", "SPAC26A3.17c", "SPAC26A3.18c", "SPAC27D7.03c", "SPAC27D7.04", "SPAC27D7.05c", "SPAC27D7.06", "SPAC27D7.07c", "SPAC27D7.08c", "SPAC27D7.09c", "SPAC27F1.10", "SPAC29A4.02c", "SPAC29A4.03c", "SPAC29A4.04c", "SPAC29A4.05", "SPAC29A4.06c", "SPAC29A4.07c", "SPAC29B12.13", "SPAC29B12.14c", "SPAC29E6.01", "SPAC29E6.02", "SPAC29E6.03c", "SPAC29E6.04", "SPAC29E6.05c", "SPAC29E6.06c", "SPAC2E1P3.04", "SPAC2E1P3.05c", "SPAC2E1P5.01c", "SPAC2E1P5.02c", "SPAC2E1P5.03", "SPAC2E1P5.04c", "SPAC2E1P5.05c", "SPAC2F7.11", "SPAC2F7.13c", "SPAC2F7.14c", "SPAC2F7.15", "SPAC2F7.16c", "SPAC2F7.17", "SPAC2G11.02c", "SPAC2G11.03c", "SPAC2G11.04c", "SPAC2G11.05c", "SPAC2G11.06c", "SPAC2G11.07c", "SPAC2G11.08c", "SPAC30D11.01c", "SPAC30D11.02c", "SPAC30D11.03", "SPAC30D11.04c", "SPAC30D11.05", "SPAC30D11.06c", "SPAC30D11.07c", "SPAC31G5.03", "SPAC31G5.04", "SPAC31G5.05c", "SPAC31G5.06", "SPAC31G5.07", "SPAC31G5.08", "SPAC31G5.09c", "SPAC328.09", "SPAC328.10c", "SPAC32A11.01", "SPAC32A11.02c", "SPAC32A11.03c", "SPAC32A11.04c", "SPAC32A11.05c", "SPAC32A11.06c", "SPAC32A11.07c", "SPAC33A11.09", "SPAC3A11.10c", "SPAC3A11.11c", "SPAC3A11.12c", "SPAC3A11.13", "SPAC3A11.14c", "SPAC3A11.15c", "SPAC3C7.10", "SPAC3C7.11c", "SPAC3C7.12", "SPAC3C7.13c", "SPAC3C7.14c", "SPAC3F10.02c", "SPAC3F10.03c", "SPAC3G9.01", "SPAC3G9.02", "SPAC3G9.03", "SPAC3G9.04", "SPAC3G9.05", "SPAC3G9.06", "SPAC3G9.07c", "SPAC3G9.08c", "SPAC3H5.08c", "SPAC3H5.09c", "SPAC3H5.10", "SPAC3H5.11", "SPAC3H5.12c", "SPAC3H5.13", "SPAC3H8.02", "SPAC3H8.03c", "SPAC4A8.16c", "SPAC4C5.01", "SPAC4C5.02c", "SPAC4C5.03", "SPAC4C5.04", "SPAC4D7.01c", "SPAC4D7.02c", "SPAC4D7.03c", "SPAC4F10.16c", "SPAC4F10.17", "SPAC4F10.18", "SPAC4F10.19c", "SPAC4F10.20", "SPAC4F10.22", "SPAC4F8.03c", "SPAC4G9.05", "SPAC4G9.06c", "SPAC4G9.07", "SPAC4G9.08c", "SPAC4G9.09c", "SPAC4G9.10", "SPAC4G9.11c", "SPAC4G9.12c"

```

## "SPAC513.04", "SPAC513.05", "SPAC513.06c", "SPAC513.07", "SPAC521.02", "SPAC521.03", "SPAC521.04c",  
## "SPAC57A10.06", "SPAC57A10.07", "SPAC57A10.08c", "SPAC57A10.09c", "SPAC57A10.10c", "SPAC57A10.11c",  
## "SPAC5D6.05", "SPAC5D6.06c", "SPAC5D6.07c", "SPAC5D6.08c", "SPAC5D6.09c", "SPAC5D6.10c", "SPAC5D6.12c",  
## "SPAC630.07c", "SPAC630.08c", "SPAC630.09c", "SPAC630.10", "SPAC630.11", "SPAC630.12", "SPAC630.13c",  
## "SPAC644.17c", "SPAC644.18c", "SPAC652.01", "SPAC664.01c", "SPAC664.02c", "SPAC664.03", "SPAC664.04c",  
## "SPAC694.04c", "SPAC694.05c", "SPAC694.06c", "SPAC6B12.02c", "SPAC6B12.03c", "SPAC6B12.04c", "SPAC6B12.05c",  
## "SPAC6F12.08c", "SPAC6F12.09", "SPAC6F12.10c", "SPAC6F12.11c", "SPAC6F12.12", "SPAC6F12.13c", "SPAC6F12.14c",  
## "SPAC6G10.09", "SPAC6G10.10c", "SPAC6G10.11c", "SPAC6G10.12c", "SPAC6G9.01c", "SPAC6G9.02c", "SPAC6G9.03c",  
## "SPAC7D4.05", "SPAC7D4.06c", "SPAC7D4.07c", "SPAC7D4.08", "SPAC7D4.09c", "SPAC7D4.10", "SPAC7D4.11c",  
## "SPAC823.07", "SPAC823.08c", "SPAC823.09c", "SPAC823.10c", "SPAC823.11", "SPAC823.12", "SPAC823.13c",  
## "SPAC890.08", "SPAC8C9.03", "SPAC8C9.04", "SPAC8C9.05", "SPAC8C9.06c", "SPAC8C9.07", "SPAC8C9.08", "SPAC8C9.09c",  
## "SPAC8F11.10c", "SPAC9.02c", "SPAC9.03c", "SPAC9.04", "SPAC9.05", "SPAC9.06c", "SPAC9.07c", "SPAC9.08c",  
## "SPAC959.10", "SPAC959.11", "SPAC977.01", "SPAC977.02", "SPAC977.03", "SPAC977.04", "SPAC977.05c", "SPAC977.06c",  
## "SPAC9E9.17c", "SPAC9G1.02", "SPAC9G1.03c", "SPAC9G1.04", "SPAC9G1.05", "SPAC9G1.06c", "SPAC9G1.07", "SPAC9G1.08c",  
## "SPAP14E8.04", "SPAP14E8.05c", "SPAP19A11.05c", "SPAP27G11.02", "SPAP27G11.03", "SPAP27G11.04c", "SPAP27G11.05c",  
## "SPAP8A3.08", "SPAP8A3.09c", "SPAP8A3.10", "SPAP8A3.11c", "SPAP8A3.12c", "SPAP8A3.13c", "SPAP8A3.14c",  
## "SPAPB1A10.05", "SPAPB1A10.06c", "SPAPB1A10.07c", "SPAPB1A10.08", "SPAPB1A10.09", "SPAPB1A10.10c", "SPAPB1A10.11c",  
## "SPAPB24D3.02c", "SPAPB24D3.03", "SPAPB24D3.04c", "SPAPB24D3.05c", "SPAPB24D3.06c", "SPAPB24D3.07c",  
## "SPAPJ698.02c", "SPAPJ698.03c", "SPAPJ760.02c", "SPAPJ760.03c", "SPAPYUG7.02c", "SPAPYUG7.03c", "SPAPYUG7.04c",  
## "SPATRNAILE.02", "SPATRNAILE.03", "SPATRNAILE.04", "SPATRNALEU.01", "SPATRNALEU.02", "SPATRNALEU.03",  
## "SPATRNATYR.01", "SPATRNAVAL.01", "SPATRNAVAL.02", "SPATRNAVAL.03", "SPATRNAVAL.04", "SPBC106.01", "SPBC106.02",  
## "SPBC1105.10", "SPBC1105.11c", "SPBC1105.12", "SPBC1105.13c", "SPBC1105.14", "SPBC1105.15c", "SPBC1105.16c",  
## "SPBC1198.04c", "SPBC1198.05", "SPBC1198.06c", "SPBC1198.07c", "SPBC1198.08", "SPBC1198.09", "SPBC1198.10c",  
## "SPBC11G11.01", "SPBC11G11.02c", "SPBC11G11.03", "SPBC11G11.04", "SPBC11G11.05", "SPBC11G11.06c", "SPBC11G11.07c",  
## "SPBC1289.09", "SPBC1289.10c", "SPBC1289.11", "SPBC1289.12", "SPBC1289.13c", "SPBC1289.14", "SPBC1289.15c",  
## "SPBC1306.01c", "SPBC1306.02", "SPBC1347.01c", "SPBC1347.02", "SPBC1347.03", "SPBC1347.04", "SPBC1347.05c",  
## "SPBC13A2.04c", "SPBC13E7.01", "SPBC13E7.02", "SPBC13E7.03c", "SPBC13E7.04", "SPBC13E7.05", "SPBC13E7.06c",  
## "SPBC146.06c", "SPBC146.07", "SPBC146.08c", "SPBC146.09c", "SPBC146.10", "SPBC146.11c", "SPBC146.12",  
## "SPBC14F5.08", "SPBC14F5.09c", "SPBC14F5.10c", "SPBC14F5.11c", "SPBC14F5.12c", "SPBC14F5.13c", "SPBC14F5.14c",  
## "SPBC15D4.12c", "SPBC15D4.13c", "SPBC15D4.14", "SPBC15D4.15", "SPBC1604.01", "SPBC1604.02c", "SPBC1604.03c",  
## "SPBC1683.04", "SPBC1683.05", "SPBC1683.06c", "SPBC1683.07", "SPBC1683.08", "SPBC1683.09c", "SPBC1683.10c",  
## "SPBC16A3.07c", "SPBC16A3.08c", "SPBC16A3.09c", "SPBC16A3.10", "SPBC16A3.11", "SPBC16A3.12c", "SPBC16A3.13c",  
## "SPBC16D10.07c", "SPBC16D10.08c", "SPBC16D10.09", "SPBC16D10.10", "SPBC16D10.11c", "SPBC16E9.01c", "SPBC16E9.02c",  
## "SPBC16G5.09", "SPBC16G5.10", "SPBC16G5.11c", "SPBC16G5.12c", "SPBC16G5.13", "SPBC16G5.14c", "SPBC16G5.15c",  
## "SPBC1703.09", "SPBC1703.10", "SPBC1703.11", "SPBC1703.12", "SPBC1703.13c", "SPBC1703.14c", "SPBC1703.15c",  
## "SPBC1711.04", "SPBC1711.05", "SPBC1711.06", "SPBC1711.07", "SPBC1711.08", "SPBC1711.09c", "SPBC1711.10c",  
## "SPBC1734.12c", "SPBC1734.13", "SPBC1734.14c", "SPBC1734.15", "SPBC1734.16c", "SPBC1773.01", "SPBC1773.02c",  
## "SPBC17A3.02", "SPBC17A3.03c", "SPBC17A3.04c", "SPBC17A3.05c", "SPBC17A3.06", "SPBC17A3.07", "SPBC17A3.08c",  
## "SPBC17G9.07", "SPBC17G9.08c", "SPBC17G9.09", "SPBC17G9.10", "SPBC17G9.11c", "SPBC17G9.12c", "SPBC17G9.13c",  
## "SPBC18E5.14c", "SPBC18E5.15", "SPBC18H10.02", "SPBC18H10.03", "SPBC18H10.04c", "SPBC18H10.05", "SPBC18H10.06c",  
## "SPBC19C2.05", "SPBC19C2.06c", "SPBC19C2.07", "SPBC19C2.08", "SPBC19C2.09", "SPBC19C2.10", "SPBC19C2.11c",  
## "SPBC19F8.06c", "SPBC19F8.07", "SPBC19F8.08", "SPBC19G7.01c", "SPBC19G7.02", "SPBC19G7.03c", "SPBC19G7.04c",  
## "SPBC1D7.04", "SPBC1D7.05", "SPBC1E8.02", "SPBC1E8.03c", "SPBC1E8.04", "SPBC1E8.05", "SPBC20F10.01",  
## "SPBC215.04", "SPBC215.05", "SPBC215.06c", "SPBC215.07c", "SPBC215.08c", "SPBC215.09c", "SPBC215.10c",  
## "SPBC21C3.01c", "SPBC21C3.02c", "SPBC21C3.03", "SPBC21C3.04c", "SPBC21C3.05", "SPBC21C3.06", "SPBC21C3.07c",  
## "SPBC21H7.06c", "SPBC21H7.07c", "SPBC23E6.01c", "SPBC23E6.02", "SPBC23E6.03c", "SPBC23E6.04c", "SPBC23E6.05c",  
## "SPBC24C6.07", "SPBC24C6.08c", "SPBC24C6.09c", "SPBC24C6.10c", "SPBC24C6.11", "SPBC24C6.13", "SPBC24C6.14c",  
## "SPBC25H2.13c", "SPBC25H2.14", "SPBC25H2.15", "SPBC25H2.16c", "SPBC25H2.18", "SPBC26H8.01", "SPBC26H8.02c",  
## "SPBC27B12.07", "SPBC27B12.08", "SPBC27B12.09c", "SPBC27B12.10c", "SPBC27B12.11c", "SPBC27B12.12c",  
## "SPBC29A10.08", "SPBC29A10.09c", "SPBC29A10.10c", "SPBC29A10.11c", "SPBC29A10.12", "SPBC29A10.13", "SPBC29A10.14c",  
## "SPBC29B5.04c", "SPBC2A9.02", "SPBC2A9.03", "SPBC2A9.04c", "SPBC2A9.05c", "SPBC2A9.06c", "SPBC2A9.07c",  
## "SPBC2F12.04", "SPBC2F12.05c", "SPBC2F12.06c", "SPBC2F12.07c", "SPBC2F12.08c", "SPBC2F12.09c", "SPBC2F12.10", "SPBC2F12.11c",  
## "SPBC2G5.06c", "SPBC2G5.07c", "SPBC30B4.01c", "SPBC30B4.02c", "SPBC30B4.03c", "SPBC30B4.04c", "SPBC30B4.05c",  
## "SPBC31E1.01c", "SPBC31E1.02c", "SPBC31E1.03", "SPBC31E1.04", "SPBC31E1.05", "SPBC31E1.06", "SPBC31E1.07c",  
## "SPBC32A10.03", "SPBC32A10.04c", "SPBC32A10.05", "SPBC32A10.06", "SPBC32A10.07", "SPBC32A10.08c", "SPBC32A10.09c",  
## "SPBC33B5.02", "SPBC33B5.03", "SPBC33B5.04c", "SPBC33B5.05", "SPBC33B5.06", "SPBC33B5.07c", "SPBC33B5.08c",  
## "SPBC34C6.01", "SPBC34C6.02", "SPBC34C6.03c", "SPBC34C6.04", "SPBC34C6.05", "SPBC34C6.06c", "SPBC34C6.07c",  
## "SPBC35H2.05", "SPBC35H2.06c", "SPBC35H2.07", "SPBC35H2.08", "SPBC35H2.09", "SPBC35H2.10c", "SPBC35H2.11c",  
## "SPBC36B5.03", "SPBC36B5.04c", "SPBC36B5.05", "SPBC36B5.06", "SPBC36B5.07c", "SPBC36B5.08c", "SPBC36B5.09c",  
## "SPBC37C6.01", "SPBC37C6.02", "SPBC37C6.03c", "SPBC37C6.04", "SPBC37C6.05", "SPBC37C6.06c", "SPBC37C6.07c",  
## "SPBC38H2.04", "SPBC38H2.05c", "SPBC38H2.06", "SPBC38H2.07", "SPBC38H2.08", "SPBC38H2.09c", "SPBC38H2.10c",  
## "SPBC39B5.02", "SPBC39B5.03c", "SPBC39B5.04", "SPBC39B5.05", "SPBC39B5.06c", "SPBC39B5.07c", "SPBC39B5.08c",  
## "SPBC40C6.01", "SPBC40C6.02", "SPBC40C6.03c", "SPBC40C6.04", "SPBC40C6.05", "SPBC40C6.06c", "SPBC40C6.07c",  
## "SPBC41H2.03", "SPBC41H2.04c", "SPBC41H2.05", "SPBC41H2.06", "SPBC41H2.07c", "SPBC41H2.08c", "SPBC41H2.09c",  
## "SPBC42B5.01", "SPBC42B5.02c", "SPBC42B5.03", "SPBC42B5.04", "SPBC42B5.05c", "SPBC42B5.06c", "SPBC42B5.07c",  
## "SPBC43C6.00", "SPBC43C6.01c", "SPBC43C6.02", "SPBC43C6.03c", "SPBC43C6.04", "SPBC43C6.05c", "SPBC43C6.06c",  
## "SPBC44H2.02", "SPBC44H2.03c", "SPBC44H2.04", "SPBC44H2.05", "SPBC44H2.06c", "SPBC44H2.07c", "SPBC44H2.08c",  
## "SPBC45B5.01", "SPBC45B5.02c", "SPBC45B5.03", "SPBC45B5.04", "SPBC45B5.05c", "SPBC45B5.06c", "SPBC45B5.07c",  
## "SPBC46C6.00", "SPBC46C6.01c", "SPBC46C6.02", "SPBC46C6.03c", "SPBC46C6.04", "SPBC46C6.05c", "SPBC46C6.06c",  
## "SPBC47H2.01", "SPBC47H2.02c", "SPBC47H2.03", "SPBC47H2.04", "SPBC47H2.05c", "SPBC47H2.06c", "SPBC47H2.07c",  
## "SPBC48B5.00", "SPBC48B5.01c", "SPBC48B5.02", "SPBC48B5.03c", "SPBC48B5.04", "SPBC48B5.05c", "SPBC48B5.06c",  
## "SPBC49C6.00", "SPBC49C6.01c", "SPBC49C6.02", "SPBC49C6.03c", "SPBC49C6.04", "SPBC49C6.05c", "SPBC49C6.06c",  
## "SPBC50H2.00", "SPBC50H2.01c", "SPBC50H2.02", "SPBC50H2.03c", "SPBC50H2.04", "SPBC50H2.05c", "SPBC50H2.06c",  
## "SPBC51B5.00", "SPBC51B5.01c", "SPBC51B5.02", "SPBC51B5.03c", "SPBC51B5.04", "SPBC51B5.05c", "SPBC51B5.06c",  
## "SPBC52C6.00", "SPBC52C6.01c", "SPBC52C6.02", "SPBC52C6.03c", "SPBC52C6.04", "SPBC52C6.05c", "SPBC52C6.06c",  
## "SPBC53H2.00", "SPBC53H2.01c", "SPBC53H2.02", "SPBC53H2.03c", "SPBC53H2.04", "SPBC53H2.05c", "SPBC53H2.06c",  
## "SPBC54B5.00", "SPBC54B5.01c", "SPBC54B5.02", "SPBC54B5.03c", "SPBC54B5.04", "SPBC54B5.05c", "SPBC54B5.06c",  
## "SPBC55C6.00", "SPBC55C6.01c", "SPBC55C6.02", "SPBC55C6.03c", "SPBC55C6.04", "SPBC55C6.05c", "SPBC55C6.06c",  
## "SPBC56H2.00", "SPBC56H2.01c", "SPBC56H2.02", "SPBC56H2.03c", "SPBC56H2.04", "SPBC56H2.05c", "SPBC56H2.06c",  
## "SPBC57B5.00", "SPBC57B5.01c", "SPBC57B5.02", "SPBC57B5.03c", "SPBC57B5.04", "SPBC57B5.05c", "SPBC57B5.06c",  
## "SPBC58C6.00", "SPBC58C6.01c", "SPBC58C6.02", "SPBC58C6.03c", "SPBC58C6.04", "SPBC58C6.05c", "SPBC58C6.06c",  
## "SPBC59H2.00", "SPBC59H2.01c", "SPBC59H2.02", "SPBC59H2.03c", "SPBC59H2.04", "SPBC59H2.05c", "SPBC59H2.06c",  
## "SPBC60B5.00", "SPBC60B5.01c", "SPBC60B5.02", "SPBC60B5.03c", "SPBC60B5.04", "SPBC60B5.05c", "SPBC60B5.06c",  
## "SPBC61C6.00", "SPBC61C6.01c", "SPBC61C6.02", "SPBC61C6.03c", "SPBC61C6.04", "SPBC61C6.05c", "SPBC61C6.06c",  
## "SPBC62H2.00", "SPBC62H2.01c", "SPBC62H2.02", "SPBC62H2.03c", "SPBC62H2.04", "SPBC62H2.05c", "SPBC62H2.06c",  
## "SPBC63B5.00", "SPBC63B5.01c", "SPBC63B5.02", "SPBC63B5.03c", "SPBC63B5.04", "SPBC63B5.05c", "SPBC63B5.06c",  
## "SPBC64C6.00", "SPBC64C6.01c", "SPBC64C6.02", "SPBC64C6.03c", "SPBC64C6.04", "SPBC64C6.05c", "SPBC64C6.06c",  
## "SPBC65H2.00", "SPBC65H2.01c", "SPBC65H2.02", "SPBC65H2.03c", "SPBC65H2.04", "SPBC65H2.05c", "SPBC65H2.06c",  
## "SPBC66B5.00", "SPBC66B5.01c", "SPBC66B5.02", "SPBC66B5.03c", "SPBC66B5.04", "SPBC66B5.05c", "SPBC66B5.06c",  
## "SPBC67C6.00", "SPBC67C6.01c", "SPBC67C6.02", "SPBC67C6.03c", "SPBC67C6.04", "SPBC67C6.05c", "SPBC67C6.06c",  
## "SPBC68H2.00", "SPBC68H2.01c", "SPBC68H2.02", "SPBC68H2.03c", "SPBC68H2.04", "SPBC68H2.05c", "SPBC68H2.06c",  
## "SPBC69B5.00", "SPBC69B5.01c", "SPBC69B5.02", "SPBC69B5.03c", "SPBC69B5.04", "SPBC69B5.05c", "SPBC69B5.06c",  
## "SPBC70C6.00", "SPBC70C6.01c", "SPBC70C6.02", "SPBC70C6.03c", "SPBC70C6.04", "SPBC70C6.05c", "SPBC70C6.06c",  
## "SPBC71H2.00", "SPBC71H2.01c", "SPBC71H2.02", "SPBC71H2.03c", "SPBC71H2.04", "SPBC71H2.05c", "SPBC71H2.06c",  
## "SPBC72B5.00", "SPBC72B5.01c", "SPBC72B5.02", "SPBC72B5.03c", "SPBC72B5.04", "SPBC72B5.05c", "SPBC72B5.06c",  
## "SPBC73C6.00", "SPBC73C6.01c", "SPBC73C6.02", "SPBC73C6.03c", "SPBC73C6.04", "SPBC73C6.05c", "SPBC73C6.06c",  
## "SPBC74H2.00", "SPBC74H2.01c", "SPBC74H2.02", "SPBC74H2.03c", "SPBC74H2.04", "SPBC74H2.05c", "SPBC74H2.06c",  
## "SPBC75B5.00", "SPBC75B5.01c", "SPBC75B5.02", "SPBC75B5.03c", "SPBC75B5.04", "SPBC75B5.05c", "SPBC75B5.06c",  
## "SPBC76C6.00", "SPBC76C6.01c", "SPBC76C6.02", "SPBC76C6.03c", "SPBC76C6.04", "SPBC76C6.05c", "SPBC76C6.06c",  
## "SPBC77H2.00", "SPBC77H2.01c", "SPBC77H2.02", "SPBC77H2.03c", "SPBC77H2.04", "SPBC77H2.05c", "SPBC77H2.06c",  
## "SPBC78B5.00", "SPBC78B5.01c", "SPBC78B5.02", "SPBC78B5.03c", "SPBC78B5.04", "SPBC78B5.05c", "SPBC78B5.06c",  
## "SPBC79C6.00", "SPBC79C6.01c", "SPBC79C6.02", "SPBC79C6.03c", "SPBC79C6.04", "SPBC79C6.05c", "SPBC79C6.06c",  
## "SPBC80H2.00", "SPBC80H2.01c", "SPBC80H2.02", "SPBC80H2.03c", "SPBC80H2.04", "SPBC80H2.05c", "SPBC80H2.06c",  
## "SPBC81B5.00", "SPBC81B5.01c", "SPBC81B5.02", "SPBC81B5.03c", "SPBC81B5.04", "SPBC81B5.05c", "SPBC81B5.06c",  
## "SPBC82C6.00", "SPBC82C6.01c", "SPBC82C6.02", "SPBC82C6.03c", "SPBC82C6.04", "SPBC82C6.05c", "SPBC82C6.06c",  
## "SPBC83H2.00", "SPBC83H2.01c", "SPBC83H2.02", "SPBC83H2.03c", "SPBC83H2.04", "SPBC83H2.05c", "SPBC83H2.06c",  
## "SPBC84B5.00", "SPBC84B5.01c", "SPBC84B5.02", "SPBC84B5.03c", "SPBC84B5.04", "SPBC84B5.05c", "SPBC84B5.06c",  
## "SPBC85C6.00", "SPBC85C6.01c", "SPBC85C6.02", "SPBC85C6.03c", "SPBC85C6.04", "SPBC85C6.05c", "SPBC85C6.06c",  
## "SPBC86H2.00", "SPBC86H2.01c", "SPBC86H2.02", "SPBC86H2.03c", "SPBC86H2.04", "SPBC86H2.05c", "SPBC86H2.06c",  
## "SPBC87B5.00", "SPBC87B5.01c", "SPBC87B5.02", "SPBC87B5.03c", "SPBC87B5.04", "SPBC87B5.05c", "SPBC87B5.06c",  
## "SPBC88C6.00", "SPBC88C6.01c", "SPBC88C6.02", "SPBC88C6.03c", "SPBC88C6.04", "SPBC88C6.05c", "SPBC88C6.06c",  
## "SPBC89H2.00", "SPBC89H2.01c", "SPBC89H2.02", "SPBC89H2.03c", "SPBC89H2.04", "SPBC89H2.05c", "SPBC89H2.06c",  
## "SPBC90B5.00", "SPBC90B5.01c", "SPBC90B5.02", "SPBC90B5.03c", "SPBC90B5.04", "SPBC90B5.05c", "SPBC90B5.06c",  
## "SPBC91C6.00", "SPBC91C6.01c", "SPBC91C6.02", "SPBC91C6.03c", "SPBC91C6.04", "SPBC91C6.05c", "SPBC91C6.06c",  
## "SPBC92H2.00", "SPBC92H2.01c", "SPBC92H2.02", "SPBC92H2.03c", "SPBC92H2.04", "SPBC92H2.05c", "SPBC92H2.06c",  
## "SPBC93B5.00", "SPBC93B5.01c", "SPBC93B5.02", "SPBC93B5.03c", "SPBC93B5.04", "SPBC93B5.05c", "SPBC93B5.06c",  
## "SPBC94C6.00", "SPBC94C6.01c", "SPBC94C6.02", "SPBC94C6.03c", "SPBC94C6.04", "SPBC94C6.05c", "SPBC94C6.06c",  
## "SPBC95H2.00", "SPBC95H2.01c", "SPBC95H2.02", "SPBC95H2.03c", "SPBC95H2.04", "SPBC95H2.05c", "SPBC95H2.06c",  
## "SPBC96B5.00", "SPBC96B5.01c", "SPBC96B5.02", "SPBC96B5.03c", "SPBC96B5.04", "SPBC96B5.05c", "SPBC96B5.06c",  
## "SPBC97C6.00", "SPBC97C6.01c", "SPBC97C6.02", "SPBC97C6.03c", "SPBC97C6.04", "SPBC97C6.05c", "SPBC97C6.06c",  
## "SPBC98H2.00", "SPBC98H2.01c", "SPBC98H2.02", "SPBC98H2.03c", "SPBC98H2.04", "SPBC98H2.05c", "SPBC98H2.06c",  
## "SPBC99B5.00", "SPBC99B5.01c", "SPBC99B5.02", "SPBC99B5.03c", "SPBC99B5.04", "SPBC99B5.05c", "SPBC99B5.06c",  
## "SPBC100C6.00", "SPBC100C6.01c", "SPBC100C6.02", "SPBC100C6.03c", "SPBC100C6.04", "SPBC100C6.05c", "SPBC100C6.06c",  
## "SPBC101H2.00", "SPBC101H2.01c", "SPBC101H2.02", "SPBC101H2.03c", "SPBC101H2.04", "SPBC101H2.05c", "SPBC101H2.06c",  
## "SPBC102B5.00", "SPBC102B5.01c", "SPBC102B5.02", "SPBC102B5.03c", "SPBC102B5.04", "SPBC102B5.05c", "SPBC102B5.06c",  
## "SPBC103C6.00", "SPBC103C6.01c", "SPBC103C6.02", "SPBC103C6.03c", "SPBC103C6.04", "SPBC103C6.05c", "SPBC103C6.06c",  
## "SPBC104H2.00", "SPBC104H2.01c", "SPBC104H2.02", "SPBC104H2.03c", "SPBC104H2.04", "SPBC104H2.05c", "SPBC104H2.06c",  
## "SPBC105B5.00", "SPBC105B5.01c", "SPBC105B5.02", "SPBC105B5.03c", "SPBC105B5.04", "SPBC105B5.05c", "SPBC105B5.06c",  
## "SPBC106C6.00", "SPBC106C6.01c", "SPBC106C6.02", "SPBC106C6.03c", "SPBC106C6.04", "SPBC106C6.05c", "SPBC106C6.06c",  
## "SPBC107H2.00", "SPBC107H2.01c", "SPBC107H2.02", "SPBC107H2.03c", "SPBC107H2.04", "SPBC107H2.05c", "SPBC107H2.06c",  
## "SPBC108B5.00", "SPBC108B5.01c", "SPBC108B5.02", "SPBC108B5.03c", "SPBC108B5.04", "SPBC108B5.05c", "SPBC108B5.06c",  
## "SPBC109C6.00", "SPBC109C6.01c", "SPBC109C6.02", "SPBC109C6.03c", "SPBC109C6.04", "SPBC109C6.05c", "SPBC109C6.06c",  
## "SPBC110H2.00", "SPBC110H2.01c", "SPBC110H2.02", "SPBC110H2.03c", "SPBC110H2.04", "SPBC110H2.05c", "SPBC110H2.06c",  
## "SPBC111B5.00", "SPBC111B5.01c", "SPBC111B5.02", "SPBC111B5.03c", "SPBC111B5.04", "SPBC111B5.05c", "SPBC111B5.06c",  
## "SPBC112C6.00", "SPBC112C6.01c", "SPBC112C6.02", "SPBC112C6.03c", "SPBC112C6.04", "SPBC112C6.05c", "SPBC112C6.06c",  
## "SPBC113H2.00", "SPBC113H2.01c", "SPBC113H2.02", "SPBC113H2.03c", "SPBC113H2.04", "SPBC113H2.05c", "SPBC113H2.06c",  
## "SPBC114B5.00", "SPBC114B5.01c", "SPBC114B5.02", "SPBC114B5.03c", "SPBC114B5.04", "SPBC114B5.05c", "SPBC114B5.06c",  
## "SPBC115C6.00", "SPBC115C6.01c", "SPBC115C6.02", "SPBC115C6.03c", "SPBC115C6.04", "SPBC115C6.05c", "SPBC115C6.06c",  
## "SPBC116H2.00", "SPBC116H2.01c", "SPBC116H2.02", "SPBC116H2.03c", "SPBC116H2.04", "SPBC116H2.05c", "SPBC116H2.06c",  
## "SPBC117B5.00", "SPBC117B5.01c", "SPBC117B5.02", "SPBC117B5.03c", "SPBC117B5.04", "SPBC117B5.05c", "SPBC117B5.06c",  
## "SPBC118C6.00", "SPBC118C6.01c", "SPBC118C6.02", "SPBC118C6.03c", "SPBC118C6.04", "SPBC118C6.05c", "SPBC118C6.06c",  
## "SPBC119H2.00", "SPBC119H2.01c", "SPBC119H2.02", "SPBC119H2.03c", "SPBC119H2.04", "SPBC119H2.05c", "SPBC119H2.06c",  
## "SPBC120B5.00", "SPBC120B5.01c", "SPBC120B5.02", "SPBC120B5.03c", "SPBC120B5.04", "SPBC120B5.05c", "SPBC120B5.06c",  
## "SPBC121C6.00", "SPBC121C6.01c", "SPBC121C6.02", "SPBC121C6.03c", "SPBC121C6.04", "SPBC121C6.05c", "SPBC121C6.06c",  
## "SPBC122H2.00", "SPBC122H2.01c", "SPBC122H2.02", "SPBC122H2.





```

## "GO:0060237,GO:0045944,GO:0000122,GO:0005730,GO:0000790,GO:0072686,GO:0005634,GO:0001077,GO:0000978,GO:0000979,GO:0000980,GO:0005635", "GO:0006357,GO:0000324,GO:0005634,GO:0000978,GO:0000982,GO:0008270", "GO:0008033,GO:0006378,GO:0006379", "GO:0002182,GO:0022625,GO:0005829,GO:0003735", "GO:0006457,GO:0005730,GO:0005634,GO:0030544", "GO:0001680,GO:0006400,GO:0004810,GO:0052928,GO:0052927,GO:0052929,GO:0003723,GO:0005524", "GO:0006371,GO:0007017,GO:0006400,GO:0005829,GO:0005634,GO:0005737", "GO:0030433,GO:0016567,GO:0005789,GO:0016021,GO:0000749,GO:0071432,GO:0034613,GO:0018229,GO:0018125,GO:0005789,GO:0016021,GO:0005783,GO:0004671", "GO:0032951,GO:0046909,GO:0033157,GO:0032488,GO:2000769,GO:0032955,GO:0032005,GO:0017157,GO:0030950,GO:0000054,GO:0042254,GO:0030690,GO:0030691,GO:0005730", "GO:0098705,GO:0005773,GO:0005887,GO:0005888", "GO:0055085,GO:0000329,GO:0005794,GO:0016021,GO:0015297", "GO:0006357,GO:0005634,GO:0000978,GO:0000979,GO:0016021", "GO:0043486,GO:0006338,GO:0000812,GO:0005634,GO:0004003,GO:0003677,GO:0005524", "GO:0015991,GO:0007035,GO:0000221,GO:0000329,GO:0046961,GO:0005524", "GO:0006310,GO:0048478,GO:0006971,GO:0016021", "GO:0060360,GO:0034762,GO:0006511,GO:0034067,GO:0016567,GO:0051453,GO:0071944,GO:0005730,GO:0000750,GO:0031139,GO:1900482,GO:1901099,GO:0035624,GO:0005887,GO:0031520,GO:0000324,GO:0051286,GO:0006307,GO:0005829,GO:0005634,GO:0032132,GO:0003684", "GO:0006364,GO:0002181,GO:0022625,GO:0005828", "GO:0016226,GO:0005739,GO:0016491,GO:0005506,GO:0010181,GO:0003958,GO:0050661,GO:0050660", "GO:0030835,GO:0051016,GO:0034613,GO:0030036,GO:0008360,GO:0000281,GO:0008290,GO:0005884,GO:0030479,GO:0045721,GO:0043161,GO:0034657,GO:0005829,GO:0005634,GO:0005737", "GO:0007165,GO:0051286,GO:0005888", "GO:0006364,GO:0005730,GO:0005783,GO:0004521", "GO:0005829", "GO:0032543,GO:0005762,GO:0005739,GO:0005738,GO:0005634,GO:0016021", "GO:0006281,GO:0005737,GO:0004519,GO:0003677,GO:0046872", "GO:0006538,GO:0009450,GO:0005829,GO:0005737,GO:0009013", "GO:0006342,GO:0006302,GO:0006348,GO:0016571,GO:0006696,GO:0005789,GO:0016021,GO:0005783,GO:0008398,GO:0046872,GO:0005515", "GO:0007131,GO:0000742,GO:0031301,GO:0031965,GO:0005794,GO:0005737,GO:0005783", "GO:0005829,GO:0005634", "GO:0042787,GO:0034599,GO:0032443,GO:0019005,GO:0042175,GO:0005829,GO:0005634,GO:0005789,GO:0005515", "GO:0006071,GO:0019564,GO:0005739,GO:0008270,GO:0008888", "GO:0032511,GO:0016192,GO:0005771,GO:0005730,GO:0006506,GO:0030176,GO:0005783,GO:0051377", "GO:0045039,GO:0042719,GO:0005758,GO:0005829,GO:0008561,GO:0015959,GO:0071543,GO:1901909,GO:1901911,GO:1901906,GO:1901907,GO:0005829,GO:0005634,GO:0052096,GO:0006696,GO:0030176,GO:0005783,GO:0034737", "GO:0000096,GO:0070814,GO:0019379,GO:0019344,GO:0009013,GO:0031138,GO:0005829,GO:0003723", "GO:0006121,GO:0006105,GO:0006099,GO:0005749,GO:0005737,GO:0051537,GO:0033617,GO:0032543,GO:0031304,GO:0005759,GO:0005763,GO:0005739,GO:0016021,GO:0008168,GO:0005507", "GO:0006896,GO:0006886,GO:0030123,GO:0030665,GO:0005794,GO:0005634,GO:0008565", "GO:0034087,GO:0005829,GO:0005737", "GO:1904541,GO:0046379,GO:0030995,GO:0005576,GO:0005618,GO:1990819,GO:0043311,GO:1990758,GO:0007052,GO:0042729", "GO:0030979,GO:0071852,GO:0046379,GO:0031233,GO:0005783,GO:0043107,GO:0009972,GO:0006216,GO:0006206,GO:0005829,GO:0005634,GO:0005737,GO:0004126,GO:0008270", "GO:0006307,GO:0034613,GO:0031028,GO:0010973,GO:0044732,GO:0005525,GO:0003924,GO:0005515", "GO:0051444,GO:2000447,GO:0002181,GO:0005829,GO:0005634,GO:0005737,GO:0008270", "GO:0046496,GO:0005739,GO:0005829,GO:0005634", "GO:0006888,GO:0005794,GO:0005829,GO:0005634", "GO:0006409,GO:0005794,GO:0005524", "GO:0072356,GO:0005829,GO:0005730,GO:1990813,GO:0051177,GO:0045132,GO:0034613,GO:0051455,GO:0051321,GO:0034991,GO:0000941,GO:0005634,GO:0006606,GO:0031965,GO:0005635,GO:0005829,GO:0005634,GO:0008536", "GO:0032949,GO:0071963,GO:0030995,GO:1990141,GO:0045292,GO:0005681,GO:0005829,GO:0005634,GO:0005737,GO:0005515", "GO:0032543,GO:0005730,GO:0070317,GO:0018344,GO:0072661,GO:0006888,GO:0005968,GO:0005829,GO:0005634,GO:0004663,GO:0017137,GO:0007323,GO:0071586,GO:0030176,GO:0005783,GO:0004222", "GO:0070914,GO:0006271,GO:0006310,GO:1903411,GO:0048015,GO:0006914,GO:0048017,GO:0032258,GO:0000407,GO:0051286,GO:0005730,GO:0044732,GO:0072686,GO:0051984,GO:0018108,GO:0038083,GO:0033047,GO:0018105,GO:0007165,GO:0005829,GO:0005634,GO:0004712,GO:0061621,GO:0005829,GO:0005634,GO:0016787", "GO:0000288,GO:0005741,GO:0005829,GO:0003729", "GO:0016925,GO:0000724,GO:0030915,GO:0005829,GO:0005634,GO:0061665,GO:0046872,GO:0005515", "GO:0031841,GO:0006368,GO:0016575,GO:0006338,GO:0032221,GO:0005829,GO:0005634,GO:0008270", "GO:2000815,GO:0070416,GO:0009094,GO:0046417,GO:0000162,GO:0006571,GO:0005829,GO:0005634,GO:0005737,GO:0004106", "GO:0051321,GO:0032543,GO:0005763,GO:0005739,GO:0003723,GO:0003735", "GO:0000717,GO:0006367,GO:0000112,GO:0000447,GO:0005829,GO:0005634,GO:0016021", "", "GO:0005730,GO:0005829,GO:0005634", "GO:0031505,GO:0035556,GO:0005737,GO:0072543,GO:0003755", "GO:0000001,GO:0007007,GO:0006457,GO:0005743,GO:0005737", "GO:0031118,GO:0031120,GO:0031429,GO:0005732,GO:0005634,GO:0019843,GO:0034513,GO:0030515"

```

```

## "GO:0048096,GO:0060303,GO:0006369,GO:1990141,GO:0030466,GO:0030702,GO:0045944,GO:0000122,GO:0007076,GO:0005739", "GO:0002181,GO:0005730,GO:0022625,GO:0005634,GO:0003735", "GO:0006457,GO:0043234,GO:0005829,GO:0005634,GO:0031072", "GO:0006099,GO:0006106,GO:0005739,GO:0005829", "GO:0034613,GO:0007095,GO:0044773,GO:0051286,GO:0005829,GO:0005634,GO:0032153,GO:0004860,GO:0005515", "GO:0016926,GO:0005634,GO:0016929", "GO:0006888,GO:0006886,GO:0030134,GO:0005794,GO:0016021,GO:0005739", "GO:0006424,GO:0002181,GO:0005829,GO:0005634,GO:0004818,GO:0005524,GO:0005515", "GO:0070647,GO:0034613,GO:0006490,GO:0006486,GO:0005789,GO:0016021,GO:0005783,GO:0042283", "GO:0000737,GO:0006401,GO:0006369", "GO:0007031,GO:0006625,GO:0005778,GO:0005829,GO:0033328", "GO:0005829,GO:0005634", "GO:1903710,GO:0005829", "GO:0005737,GO:0016787,GO:0046872", "GO:0018343,GO:0072661,GO:0005965,GO:0005829,GO:0005634", "GO:0070880,GO:0030148,GO:0006078,GO:0016021,GO:0005783,GO:0015926", "GO:0031146,GO:0042787,GO:19034613,GO:0046135,GO:0019358,GO:0008655,GO:0034356,GO:0005634,GO:0005737,GO:0070635,GO:0045437,GO:0070636,GO:0005794,GO:0016021,GO:0005783", "GO:0051568,GO:0070869,GO:0048188,GO:0000790,GO:0005829,GO:0005634", "GO:0090334,GO:0035556,GO:0005829,GO:0005634,GO:0032153,GO:0004697,GO:0046872,GO:0005524,GO:0005515", "GO:0005634", "GO:0006614,GO:0045900,GO:0005786,GO:0005730,GO:0005829,GO:0005634", "GO:1990559,GO:0005829", "GO:0034316,GO:0000147,GO:0030479,GO:0005829,GO:0005634,GO:0071933", "GO:0018063,GO:0007005,GO:0031365,GO:0007165,GO:0005829,GO:0005634,GO:0005737,GO:0004674,GO:0005524", "GO:0022904,GO:0005739,GO:0005829", "GO:1902423,GO:0000742,GO:0030989,GO:0044816,GO:0030286,GO:0032117,GO:0071687,GO:0030981,GO:0035974,GO:0006152,GO:0034356,GO:0005829,GO:0005634,GO:0004731", "GO:0006611,GO:0006605,GO:0000055,GO:0034551,GO:0006488,GO:0006487,GO:0030176,GO:0005783,GO:0052926,GO:0052918,GO:0000030", "GO:0071574,GO:0072634,GO:0042144,GO:0006888,GO:0006886,GO:0005737,GO:0016887,GO:0005524,GO:0005515", "", "GO:0046474,GO:0005739,GO:0005829,GO:0005634", "GO:0006089,GO:0006364,GO:0005732,GO:0005730,GO:0005634,GO:0003723", "GO:0031497,GO:0071894,GO:1901487,GO:0071487,GO:0005739,GO:0005829,GO:0005634,GO:0016872", "GO:0031508,GO:0071459,GO:1990141,GO:0031048,GO:0034613,GO:0005739,GO:0005829", "GO:1900247,GO:0043234,GO:0005829", "GO:0030490,GO:0005730,GO:0032040,GO:0072686,GO:0005634,GO:0030515,GO:0010570,GO:0033566,GO:0000742,GO:0090307,GO:0043148,GO:0000743,GO:0007018,GO:0030989,GO:0034613,GO:0045292,GO:0005681,GO:0005634", "GO:0031573,GO:0033314,GO:0007095,GO:0006281,GO:0030896,GO:0035824,GO:0032543,GO:0005762,GO:0005739,GO:0003735", "GO:0042138,GO:0007131,GO:0005829,GO:0005634", "GO:00031965,GO:0005737,GO:0005783,GO:0016726", "GO:0006364,GO:0005730,GO:0032040,GO:0005634,GO:0003739,GO:0000395,GO:0045292,GO:0000243,GO:0005685,GO:0071004,GO:0005829,GO:0005634,GO:0030619,GO:0003729", "GO:0006508,GO:0030137,GO:0005794,GO:0016021,GO:0004252", "GO:0005829,GO:0005634,GO:0046872", "GO:0005739,GO:0032873,GO:0035335,GO:0010972,GO:0005739,GO:0005829,GO:0005634,GO:0005737,GO:0038082,GO:0019740,GO:0009450,GO:0006536,GO:0005739,GO:0005829,GO:0003867,GO:0030170", "GO:0015074,GO:0006364,GO:0045842,GO:0005680,GO:0044732,GO:0005829,GO:0005634,GO:0032153,GO:0005515", "GO:0043570,GO:0006281,GO:0045292,GO:0005681,GO:0005634,GO:0003723", "GO:0006675,GO:0006633,GO:0005739,GO:0005829,GO:0005634,GO:0005737,GO:0038082,GO:0046488,GO:0000935,GO:0005886,GO:0005524,GO:0016308", "GO:0005991,GO:0005992,GO:0005946,GO:0004861,GO:0005794,GO:0005737", "GO:0036286,GO:0051286,GO:0005886", "GO:0005829,GO:0005634", "GO:0046513,GO:1990626,GO:0008053,GO:0000002,GO:0005741,GO:0005739,GO:0016021", "GO:0019368,GO:0030148,GO:0030170,GO:0070131,GO:0005759,GO:0043022,GO:0005525,GO:0003924", "GO:0000289,GO:0006402,GO:0006364,GO:0030490,GO:0005730,GO:0034455,GO:0032040,GO:0030515", "GO:0015991,GO:0000220,GO:0016021,GO:0005739,GO:1902660,GO:0045744,GO:0031138,GO:0035556,GO:0005829,GO:0004674,GO:0005524", "GO:0000027,GO:0005739,GO:0007126,GO:0007165,GO:0005829,GO:0005634,GO:0019912,GO:0004672,GO:0030295,GO:0016538,GO:0004674,GO:1903695,GO:0000751,GO:0071507,GO:0000750,GO:0032005,GO:0006468,GO:0030437,GO:0000747,GO:0051286,GO:0016255,GO:0042765,GO:0005783,GO:0003923", "GO:0042168,GO:0005829,GO:0005634,GO:0020037,GO:0003923,GO:0030490,GO:0005634,GO:0005737,GO:0046872", "GO:0006896,GO:0006886,GO:0030131,GO:0005802,GO:0005739,GO:0006904,GO:0051286,GO:0005829,GO:0005634,GO:0032153", "GO:0031204,GO:0061077,GO:0034620,GO:0005739,GO:0006364,GO:0005730,GO:0005634,GO:0004004,GO:0003723,GO:0005524", "GO:0033157,GO:0070509,GO:0071487,GO:0046704,GO:0006240,GO:0006235,GO:0009263,GO:0006165,GO:0005971,GO:0005829,GO:0005737,GO:0061731,GO:0005829,GO:0005634,GO:0016614", "GO:0002181,GO:0022625,GO:0005829,GO:0019843,GO:0003735", "GO:1904600,GO:0051017,GO:0045010,GO:0030036,GO:0032220,GO:0005937,GO:1990819,GO:0043332,GO:0005829,GO:0031087,GO:0033962,GO:0010494,GO:0000932,GO:0005737,GO:0032947,GO:0005515", "", "GO:0006266,GO:0000288,GO:0006354,GO:0000289,GO:0006402,GO:0030015,GO:0005829,GO:0005634,GO:0005737", "GO:0006636,GO:0006487,GO:0034067,GO:0016567,GO:0016485,GO:0005794,GO:0016021,GO:0005783", "GO:0006368,GO:0006366,GO:0007165,GO:0005829,GO:0019211", "GO:0045292,GO:0005681,GO:0005829,GO:0005634,GO:0004004,GO:0031225,GO:0005886", "GO:1990376,GO:1900391,GO:0010847,GO:0035065,GO:0010848,GO:0043949,GO:0045829"

```

## "GO:0030148, GO:0017059, GO:0016021, GO:0005783, GO:0004758", "GO:0045039, GO:0042719, GO:0005758, GO:000583  
## "GO:0031030, GO:0046578, GO:0044732, GO:0005886, GO:0005829, GO:0005096, GO:0005515", "GO:0006782, GO:000678  
## "GO:0010388, GO:0008180, GO:0005628, GO:0005829, GO:0005634, GO:0005737", "", "GO:0045292, GO:0005634, GO:00  
## "GO:0006378, GO:0005847, GO:0005829, GO:0005634, GO:0046872", "GO:0006536, GO:0009396, GO:0046901, GO:00057  
## "GO:0071165, GO:1902977, GO:1902975, GO:0000811, GO:0031261, GO:0043596, GO:0000790, GO:0005634, GO:0005737"  
## "GO:0035269, GO:0000032, GO:0044845, GO:0012505, GO:0005789, GO:0016021, GO:0004169", "GO:0007006, GO:00320  
## "GO:0006506, GO:0030176, GO:0005783, GO:0017176", "GO:0005829, GO:0005737", "GO:0030433, GO:0034620, GO:00  
## "GO:0006890, GO:0006621, GO:0006888, GO:0006886, GO:0030137, GO:0000139, GO:0030134, GO:0005794, GO:0016021"  
## "GO:0022900, GO:0006784, GO:0031305, GO:0005743, GO:0005739, GO:0016627, GO:0051537, GO:0009055, GO:0046872"  
## "GO:0005829, GO:0005737", "GO:0070869, GO:0016575, GO:0006338, GO:0034967, GO:0070210, GO:0005634, GO:00036  
## "GO:0051445, GO:0045944, GO:0006355, GO:0071931, GO:0006357, GO:0030907, GO:0000790, GO:0005829, GO:0005634,  
## "GO:0051072, GO:0009272, GO:0005794, GO:0016021", "GO:0005829, GO:0005634", "GO:1990536, GO:0035436, GO:00  
## "GO:0019220, GO:0044242, GO:0034644, GO:0048017, GO:0007165, GO:0005829, GO:0005634, GO:0005737, GO:0004871,  
## "GO:0080009, GO:0005829, GO:0001734", "GO:0007165, GO:0005829, GO:0005634, GO:0004674, GO:0005524", "GO:00  
## "", "GO:0000902, GO:0032488, GO:2000769, GO:0032005, GO:0031139, GO:0000747, GO:0071521, GO:0051286, GO:0005  
## "GO:0006631, GO:0005739, GO:0008474", "GO:0042254, GO:0002181, GO:0022627, GO:0005829, GO:0003735", "GO:00  
## "GO:0016567, GO:0005737, GO:0061630, GO:0008270", "GO:0002181, GO:0022625, GO:0005829, GO:0003735", "GO:00  
## "GO:0070940, GO:0005829, GO:0005634, GO:0005737, GO:0008420, GO:0043175, GO:0046872", "GO:0030036, GO:00304  
## "GO:0005829, GO:0005634, GO:0005737, GO:0004427, GO:0046872", "GO:0005737, GO:0016021", "GO:0005739, GO:00  
## "GO:0007127, GO:0045842, GO:0005680, GO:0005829, GO:0005634, GO:0005515", "GO:0032264, GO:0032263, GO:00056  
## "GO:0051204, GO:0005743, GO:0005739, GO:0016021", "GO:0051017, GO:0030041, GO:0030036, GO:0035838, GO:00301  
## "GO:0006887, GO:0006886, GO:0051286, GO:0032153, GO:0005085", "GO:0042776, GO:0000276, GO:0005739, GO:00160  
## "GO:0034496, GO:0044242, GO:0006914, GO:0016236, GO:0005775, GO:0032585, GO:0005794, GO:0016021, GO:0005783,  
## "GO:0043087, GO:0016192, GO:0007165, GO:0005794, GO:0005096", "GO:0006114, GO:0046168, GO:0006116, GO:00093  
## "GO:0016998, GO:1904541, GO:0031505, GO:0072324, GO:1990819, GO:0016021, GO:0052862, GO:0052861, GO:0072317"  
## "GO:0051445, GO:0033169, GO:0006338, GO:0033193, GO:0000790, GO:0005829, GO:0005634, GO:0031491, GO:0034649,  
## "GO:0006357, GO:0006338, GO:0031011, GO:0005829, GO:0005634, GO:0003677", "GO:0030010, GO:0005886, GO:00160  
## "GO:0006508, GO:0005634, GO:0004252", "GO:0045292, GO:0005686, GO:0005681, GO:0005829, GO:0005634", "GO:00  
## "GO:0006357, GO:0006338, GO:0016514, GO:0016586, GO:0000790, GO:0005737, GO:0003677, GO:0008270", "GO:00060  
## "GO:0007165, GO:0044732, GO:0005829, GO:0005634, GO:0005737, GO:0004674, GO:0005524", "GO:0006888, GO:00068  
## "GO:0000070, GO:0031511, GO:0000776, GO:0005829, GO:0005634", "GO:0016573, GO:0071963, GO:0016575, GO:00063  
## "GO:0031146, GO:0042787, GO:0016567, GO:0043224, GO:0005829, GO:0005634, GO:0005737, GO:0061630, GO:0008270,  
## "GO:0051286, GO:0032153, GO:0005737, GO:0008677, GO:0050661", "GO:0006888, GO:0006886, GO:0000139, GO:00301  
## "GO:0018160, GO:0006782, GO:0006783, GO:0005634, GO:0005737, GO:0004418", "", "GO:0018063, GO:0007005, GO:00  
## "GO:0006435, GO:0032543, GO:0005739, GO:0004829, GO:0005524", "GO:0032543, GO:0005763, GO:0005739, GO:00000  
## "GO:0030150, GO:0005744, GO:0005759, GO:0005739, GO:0051082, GO:0030544, GO:0008270", "GO:0042787, GO:00190  
## "GO:0042254, GO:0002181, GO:0022627, GO:0005829, GO:0019843, GO:0003735", "GO:0005829, GO:0005634", "GO:200  
## "GO:0006893, GO:0005794, GO:0005737, GO:0016021", "GO:0032017, GO:0006338, GO:0000790, GO:0005634, GO:00080  
## "GO:0070131, GO:0005739, GO:0005783", "GO:0031119, GO:0005829, GO:0005634, GO:0005737, GO:0009982, GO:00037  
## "GO:0034508, GO:0000070, GO:0031511, GO:0044732, GO:0005829, GO:0005634", "GO:0031591, GO:0030488, GO:00058  
## "GO:0006357, GO:0005634, GO:0000978, GO:0000982, GO:0005515", "GO:0042138, GO:0007131, GO:0034613, GO:00309  
## "GO:0016577, GO:0006338, GO:0000790, GO:0032452", "GO:0007127, GO:0016236, GO:0000407, GO:0034045, GO:00058  
## "GO:1902800, GO:0006284, GO:0006298, GO:0006285, GO:0045007, GO:0034644, GO:0006281, GO:0006974, GO:0005634,  
## "GO:0000395, GO:0045292, GO:0005686, GO:0005685, GO:0046540, GO:0005681, GO:0005829, GO:0005634", "GO:00435  
## "GO:0000054, GO:0044613, GO:0044615, GO:0005643, GO:0034399, GO:0005635, GO:0017056", "GO:0042177, GO:19030  
## "GO:0042843, GO:0019568, GO:0005829, GO:0005634, GO:0005737, GO:0047935", "GO:0016021, GO:0005783", "GO:00  
## "GO:0000731, GO:1903461, GO:0005634, GO:0043142, GO:0003677, GO:0046872, GO:0005524", "GO:1990426, GO:00063  
## "GO:0070879, GO:0009277, GO:0004338", "GO:0006740, GO:0005634, GO:0016491", "GO:0006452, GO:0002182, GO:00  
## "GO:0030437, GO:0006729, GO:0031160, GO:0042764, GO:0005829, GO:0005634, GO:0008124", "GO:0031145, GO:00458  
## "GO:0016570, GO:0006368, GO:0006355, GO:0016593, GO:0044732, GO:0005634", "GO:0046379, GO:0009986, GO:00058  
## "GO:0051072, GO:0009272, GO:0046379, GO:0051286, GO:0009986, GO:0005794, GO:0005829, GO:0005634, GO:0016021,  
## "GO:0016021", "", "", "GO:0044396, GO:1903475, GO:0032220, GO:0000281, GO:0005826, GO:0070938, GO:0032432,  
## "GO:0030026, GO:0071421, GO:0030176, GO:0005783, GO:0005384, GO:0046872", "GO:0000245, GO:0005686, GO:00710  
## "GO:0045292, GO:0030529, GO:0005829, GO:0005634, GO:0003729", "GO:0006357, GO:0016592, GO:0005829, GO:00056

```

## "GO:0019627,GO:0005730,GO:0005737,GO:0016151", "GO:0045046,GO:0006625,GO:0005779,GO:0005778,GO:0005829", "GO:0002098,GO:0033588,GO:0005829,GO:0005737,GO:0051536,GO:0008080", "", "GO:0006357,GO:0006281,GO:0006914,GO:0016192,GO:0006886,GO:0006901,GO:0048208,GO:0030127,GO:0005789,GO:0005737", "GO:0070822,GO:0045292,GO:0046540,GO:0005681,GO:0044732,GO:0072686,GO:0005634", "GO:0048280,GO:0035493,GO:0006832,GO:0005829,GO:0005634", "GO:0005829,GO:0005737", "GO:0000395,GO:0045292,GO:0005686,GO:0005685,GO:0006364,GO:0005730,GO:0005829,GO:0005634,GO:0003723", "GO:0006388,GO:0005634,GO:0000215", "GO:0003333,GO:0051286,GO:0005794,GO:0032153,GO:0016021,GO:0015174", "", "GO:0005739,GO:0008106,GO:0016021,GO:0017137", "GO:0006457,GO:0016021,GO:0005783,GO:0030544", "GO:0009250,GO:0018344,GO:0006605,GO:0006913,GO:0005643,GO:0034399,GO:0005635,GO:0043234,GO:0005829,GO:0005634,GO:0008536,GO:0006400,GO:0008616,GO:0005737,GO:0046872,GO:0008479", "GO:0045292,GO:0005684,GO:0005634", "GO:0006357,GO:0016592,GO:0005829,GO:0005634,GO:0001104", "GO:0002183,GO:0042256,GO:0022627,GO:0005829,GO:0018345,GO:0090002,GO:0016192,GO:0031901,GO:0000329,GO:0043234,GO:0005794,GO:0016021,GO:0019706,GO:0070126,GO:0005743,GO:0005739,GO:0005737,GO:0003747,GO:0016149", "GO:0042254,GO:0005730,GO:0005829,GO:0000387,GO:0005681,GO:0005634", "GO:0070588,GO:0000329,GO:0016021,GO:0015085", "GO:0018192,GO:0032974,GO:0006914,GO:1902475,GO:0071627,GO:0000329,GO:0015179,GO:0034639", "GO:0006367,GO:0005634,GO:0005829,GO:0005634,GO:0035939,GO:1990605,GO:0043047,GO:0003723,GO:0005515", "GO:0071996,GO:0071247,GO:0005829,GO:0005634", "GO:0044247,GO:0009313,GO:0046379,GO:0009986,GO:0090599,GO:0030246", "GO:00033169,GO:0006357,GO:0033193,GO:0005634,GO:0031491,GO:0008270", "GO:0045292,GO:0005684,GO:0005634,GO:0032543,GO:0005762,GO:0005739,GO:0003735", "GO:0045842,GO:0019774,GO:0005635,GO:0005829,GO:0005634,GO:0005525,GO:0061608", "GO:0007264,GO:0005829,GO:0005634,GO:0005737,GO:0030695,GO:0005525", "GO:1990561,GO:0006879,GO:0006832,GO:0035023,GO:1903617,GO:1902406,GO:0031029,GO:0071341,GO:0005826,GO:0070938,GO:0044732,GO:0005829,GO:0009052,GO:0005829,GO:0005634,GO:0004750,GO:0046872", "GO:0000002,GO:0005739,GO:0005829,GO:0005634,GO:0045842,GO:0043161,GO:0016579,GO:0008541,GO:0034399,GO:0005634,GO:0005737,GO:0004843", "GO:0019402,GO:0005730,GO:0005634", "GO:0019674,GO:0006741,GO:0005759,GO:0005739,GO:0042736,GO:0003951,GO:0005525,GO:0006611,GO:0005643,GO:0005635,GO:0005829,GO:0005634,GO:0005049,GO:0005525,GO:0061608", "GO:0007123,GO:0000902,GO:0007021,GO:1990727,GO:0005096,GO:0005515", "GO:0006839,GO:1990550,GO:1990551,GO:0045842,GO:0005680,GO:0061630,GO:0008270,GO:0005515", "GO:0045721,GO:0043161,GO:0034657,GO:0044732,GO:0005783", "GO:0070681,GO:0032543,GO:0030956,GO:0005739,GO:0050567,GO:0005524", "GO:0002183,GO:0005829,GO:0005634", "GO:0016021", "GO:0005829,GO:0046872", "GO:0002181,GO:0005737,GO:0008757", "GO:0036376,GO:0030007,GO:0097623,GO:0005886,GO:0016021,GO:0015079,GO:0015385", "GO:0006508,GO:0005737,GO:0006091,GO:0005739,GO:0004427,GO:0000287", "GO:0016567,GO:0016021,GO:0061630,GO:0008270", "GO:0000387,GO:0045292,GO:0000974,GO:0005681,GO:0005634,GO:0036002,GO:0017070,GO:0046872,GO:0005515", "GO:0006535,GO:0005739,GO:0004124", "GO:0006006,GO:0051156,GO:0009051,GO:0006091,GO:0005829,GO:0005737,GO:0051692,GO:0009272,GO:0046379,GO:0031362,GO:0005829,GO:0005634,GO:0008496", "GO:0007165,GO:0030427,GO:0030010,GO:0010570,GO:0051519,GO:0000226,GO:0034613,GO:0071963,GO:0030437,GO:0015630,GO:0051286,GO:0046938,GO:0006750,GO:0090423,GO:0071585,GO:0071276,GO:0036087,GO:0005829,GO:0043295,GO:0004363,GO:0071996,GO:0071276,GO:0071995,GO:0000329,GO:0005794,GO:0016021,GO:0071997,GO:0071992,GO:0005524", "GO:0045842,GO:0043161,GO:0008541,GO:0005643,GO:0005829,GO:0005634,GO:0005737,GO:0005515", "GO:0006623,GO:0042147,GO:0000938,GO:0005829,GO:0005634,GO:0005768", "GO:0036297,GO:0035861,GO:0000792,GO:0006432,GO:0002181,GO:0009328,GO:0005829,GO:0005737,GO:0004826,GO:0000049,GO:0005524", "GO:0006090,GO:0006091,GO:0005829,GO:0005634,GO:0004737,GO:0030976,GO:0000287", "GO:0007026,GO:0032947,GO:0018070,GO:0006633,GO:0008897,GO:0046872", "GO:1902975,GO:0005656,GO:0031261,GO:0005664,GO:0043582,GO:0046938,GO:0071585,GO:0071276,GO:0010273,GO:0071722,GO:0005829,GO:0005634,GO:0005737,GO:0046870,GO:0007534,GO:1902981,GO:0006279,GO:1902975,GO:0000734,GO:0005658,GO:0000784,GO:0043596,GO:0035861,GO:0015914,GO:0071852,GO:0005737,GO:0008526", "GO:0032543,GO:0005762,GO:0005739,GO:0003735", "GO:0016071,GO:0030529,GO:0042175,GO:0005829", "GO:0007004,GO:0001934,GO:0031929,GO:0006338,GO:0072447,GO:0006367,GO:0005673,GO:0005634,GO:0000991,GO:0043565,GO:0046872,GO:0005515", "GO:0005829,GO:0005634,GO:0046520,GO:0019722,GO:0031902,GO:0005886,GO:0005794,GO:0005783,GO:0004143,GO:0017050,GO:0005524", "GO:0045842,GO:0019774,GO:0005635,GO:0005829,GO:0005634,GO:0004298,GO:0004175", "GO:0006015,GO:0009127,GO:0016021", "GO:0016925,GO:0031510,GO:0005829,GO:0005634,GO:0019948", "GO:0032012,GO:0006888,GO:0000022,GO:0044732,GO:0016021,GO:0005783", "GO:0046083,GO:0009113,GO:0006189,GO:0005737,GO:0004086,GO:0005886", "GO:0006487,GO:0070317,GO:0000136,GO:0000137,GO:0005794,GO:0005783", "GO:0034613,GO:0000281,GO:0005922,GO:0005739", "GO:0006913,GO:0005643,GO:0043234,GO:0005829,GO:0005634,GO:0005515", "GO:0000492,GO:0000493"

```

```

## "GO:0032543,GO:0005762,GO:0005829,GO:0005634,GO:0003735", "GO:0032543,GO:0005763,GO:0005739,GO:0003735",
## "GO:1903479,GO:1903478,GO:0071574,GO:1903477,GO:1902405,GO:0043087,GO:1903475,GO:0000281,GO:0007264,
## "GO:0071444,GO:0007165,GO:0004674,GO:0005524", "GO:0031591,GO:0030488,GO:0005829,GO:0005634,GO:0005735",
## "GO:0006506,GO:0030176,GO:0005783,GO:0000026", "GO:0033173,GO:0019722,GO:0006357,GO:0006874,GO:0005829,
## "GO:0000329,GO:0005794", "GO:0006383,GO:0005666,GO:0005829,GO:0003677,GO:0032549,GO:0003899,GO:0046872",
## "GO:0030497,GO:0005789,GO:0016021,GO:0005783,GO:0045703", "GO:0002181,GO:0022625,GO:0005829,GO:0005634,
## "GO:0005829,GO:0005634", "GO:0005739,GO:0004792", "GO:0006635,GO:0006631,GO:0005782,GO:0005829,GO:0005829,
## "", "", "GO:0002182,GO:0005829,GO:0003746,GO:0005525", "GO:0005829,GO:0005634,GO:0016787", "GO:0045829,
## "GO:0042254,GO:0002181,GO:0002182,GO:0022627,GO:0005829,GO:0003735", "GO:0006488,GO:0005829,GO:0005634,
## "GO:0006744,GO:0008299,GO:0031305,GO:0005743,GO:0005739,GO:0047293,GO:0002083,GO:0008412", "GO:0005735,
## "", "GO:0005739,GO:0005634", "GO:0016021,GO:0005783", "GO:0031142,GO:0045944,GO:0010738,GO:0031141,GO:0005735,
## "GO:0016021,GO:0005783,GO:0016787", "GO:0006359,GO:0006334,GO:0045944,GO:0006338,GO:0000790,GO:0005829,
## "GO:0006555,GO:0006534,GO:0005634,GO:0005737,GO:0008898", "GO:0030007,GO:0043462,GO:0009992,GO:0006834,
## "", "GO:1900387,GO:0045944,GO:0000122,GO:0006357,GO:0016592,GO:0005829,GO:0005634,GO:0001104", "GO:0005735,
## "GO:0016567,GO:0042254,GO:0002181,GO:0022627,GO:0005730,GO:0005737,GO:0046872,GO:0003735", "GO:0070125,
## "GO:1902103,GO:1904191,GO:0005829,GO:0005634,GO:0005515,GO:0090645", "GO:0006847,GO:0000329,GO:0005829,
## "GO:0043458,GO:0005739,GO:0004024,GO:0004022,GO:0046872", "GO:0005829,GO:0005634,GO:0005737", "GO:0005735,
## "GO:0046854,GO:0005829,GO:0005634,GO:0035004,GO:0008440", "GO:0045842,GO:0008541,GO:0005634,GO:0005524,
## "GO:0000324,GO:0016021", "GO:0043087,GO:0016192,GO:0007165,GO:0005829,GO:0005634,GO:0005096", "GO:0005735,
## "GO:0071473,GO:0043156,GO:0045014,GO:0006342,GO:0000122,GO:0043234,GO:0000790,GO:0005829,GO:0005634,
## "GO:0042125,GO:0009272,GO:0000139,GO:0016021,GO:0005783,GO:0031278", "GO:0001732,GO:0071540,GO:0016285,
## "GO:0070941,GO:0048015,GO:0030950,GO:0030036,GO:0032126,GO:0051285,GO:0035091", "GO:0098507,GO:0009458,
## "GO:0019509,GO:0019284,GO:0005829,GO:0005634,GO:0005737,GO:0000287,GO:0043716,GO:0043715,GO:0043874,
## "GO:0002182,GO:0022625,GO:0005829,GO:0003735", "GO:0033621,GO:0006378,GO:0006379,GO:0005849,GO:0005634,
## "GO:0034728,GO:0006338,GO:0031011,GO:0005829,GO:0005634,GO:0005524", "GO:0016570,GO:0006368,GO:0016567,
## "GO:0030150,GO:0006457,GO:0001405,GO:0005739,GO:0051082,GO:0016887,GO:0005524", "GO:0016226,GO:0034582,
## "GO:0034065,GO:0043007,GO:1902969,GO:0000727,GO:0033557,GO:0030874,GO:0030875,GO:0005634,GO:0017108,
## "GO:0016021,GO:0005783", "GO:0005829,GO:0005634,GO:0016888", "GO:0018026,GO:1990497,GO:0005634,GO:0005735,
## "GO:0034276,GO:0005739,GO:0005829,GO:0005634,GO:0005737,GO:0016212,GO:0047536", "GO:0006355,GO:0006354,
## "GO:0030150,GO:0005742,GO:0005739,GO:0016021,GO:0008565", "GO:0006470,GO:0007165,GO:0005829,GO:0005634,
## "", "GO:0045842,GO:0005838,GO:0005829,GO:0005634", "GO:0001682,GO:0005655,GO:0005829,GO:0005634,GO:0005735,
## "GO:0070096,GO:0030150,GO:0005742,GO:0005739,GO:0016021,GO:0015450", "GO:0006887,GO:0006886,GO:0000146,
## "GO:0033386,GO:0033384,GO:0045337,GO:0006696,GO:0008299,GO:0070317,GO:0005829,GO:0005634,GO:0004161,
## "GO:0070647,GO:0043161,GO:0000151,GO:0005739,GO:0030674,GO:0005515", "GO:0000055,GO:0005730,GO:0044725,
## "GO:0034314,GO:0000147,GO:0006897,GO:0005885,GO:0030479,GO:0051286,GO:0005829,GO:0005634,GO:0032153,
## "GO:0000723,GO:0000783,GO:0005724,GO:0005730,GO:0005634,GO:0005515", "", "GO:1902405,GO:0061160,GO:0005735,
## "GO:0006102,GO:0006740,GO:0006099,GO:0005739,GO:0005737,GO:0004450,GO:0000287,GO:0051287", "GO:0006472,
## "GO:0009132,GO:0005777,GO:0005739,GO:0008413,GO:0030145,GO:0000287", "GO:0007020,GO:0000070,GO:0071685,
## "GO:0016192,GO:0006897,GO:0006886,GO:0030659,GO:0051286,GO:0051285,GO:0005794,GO:0032153,GO:0005737,
## "GO:0006006,GO:0006000,GO:0006003,GO:0005829,GO:0004331,GO:0003873,GO:0005524", "", "GO:0055085,GO:0005735,
## "GO:0016021", "GO:0034497,GO:0030242,GO:0031142,GO:0006995,GO:0016236,GO:0032258,GO:0019898,GO:0000466,
## "GO:0006890,GO:0006886,GO:0031965,GO:0005829,GO:0005634,GO:0005789,GO:0005783", "GO:0016021,GO:0005735,
## "GO:0006183,GO:0061508,GO:0006228,GO:0006241,GO:0005758,GO:0005829,GO:0005634,GO:0004550,GO:0046872,
## "GO:0006357,GO:0031139,GO:0005634,GO:0000978,GO:0000982,GO:0008270,GO:0005515", "GO:0007092,GO:0034672,
## "GO:0051518,GO:0061246,GO:2000100,GO:0030950,GO:0007165,GO:0071470,GO:0097248,GO:0071940,GO:0033109,
## "GO:0030026,GO:0006506,GO:0000329,GO:0005794,GO:0005789,GO:0016021", "GO:0006364,GO:0005730,GO:0005829,
## "GO:0006470,GO:2000765,GO:0007165,GO:0005634,GO:0004721,GO:0046872", "GO:0006623,GO:0006914,GO:0016228,
## "GO:0030150,GO:0001405,GO:0005739,GO:0016021,GO:0001671,GO:0051082,GO:0030544,GO:0015450", "GO:000672,
## "GO:0005829,GO:0005634,GO:0004328", "GO:0008272,GO:0055085,GO:0005886,GO:0005794,GO:0016021,GO:0008272,
## "GO:0007165,GO:0051286,GO:0005829,GO:0032153,GO:0004674,GO:0005524", "GO:0000463,GO:0000466,GO:004225,
## "GO:0051321,GO:0005886", "GO:0006399,GO:0019478,GO:0002181,GO:0005829,GO:0005634,GO:0005737,GO:0016785,
## "GO:0005829,GO:0005634", "GO:1990758,GO:0042729,GO:0000779,GO:0000778,GO:0044732,GO:0072686,GO:0005829,
## "GO:0002084,GO:0005829,GO:0005634,GO:0004620", "GO:0005829,GO:0005634", "GO:0005829,GO:0005634", "GO:0005829,
## "GO:1903469,GO:0043596,GO:0035861,GO:0005829,GO:0005634,GO:0032153,GO:0005515", "GO:0016021,GO:0005735

```

```

## "GO:0019509,GO:0005737,GO:0046570,GO:0046872", "GO:0007264,GO:0005829,GO:0005634,GO:0005737,GO:00055
## "GO:0005829,GO:0005737", "GO:0055085,GO:0005737,GO:0016021,GO:0022857", "GO:0009107,GO:0006633,GO:0005737
## "GO:0000329,GO:0043234,GO:0005829,GO:0005634", "GO:0000743,GO:0007017,GO:0030029,GO:0007018,GO:00309
## "GO:0035268,GO:0000032,GO:0000139,GO:0005829,GO:0005634,GO:0005737,GO:0000026", "GO:0006621,GO:00064
## "GO:0009897,GO:0016021", "GO:0007155,GO:0010339", "GO:0005829,GO:0005634", "GO:0046475,GO:0005576,GO:0005737
## "GO:0055085,GO:0016021,GO:0022857", "", "GO:0016021", "", "GO:0009098,GO:0005829,GO:0003861,GO:00515
## "GO:0016570,GO:0030702,GO:0000070,GO:0000780,GO:0044732,GO:0000790,GO:0072686,GO:0005634,GO:0019237,
## "GO:0006364,GO:0002181,GO:0022625,GO:0005829,GO:0003735", "GO:0051205,GO:0045039,GO:0033617,GO:0005737
## "GO:0034613,GO:0000281,GO:0005940,GO:0031105,GO:0005829,GO:0032153,GO:0005525,GO:0005515", "GO:00304
## "GO:0097394,GO:0006396,GO:0006367,GO:0016591,GO:0005829,GO:0005634,GO:0003899,GO:0003723,GO:0005515"
## "GO:1990023,GO:0035861,GO:0005634,GO:0005515", "GO:0009187,GO:0030136,GO:0005829,GO:0005634,GO:00041
## "GO:0006515,GO:0005743,GO:0031942,GO:0005739,GO:0051787", "GO:0001732,GO:1901990,GO:0005829,GO:00056
## "GO:0005829,GO:0005634", "GO:0009231,GO:0005829,GO:0005634,GO:0005737,GO:0003935,GO:0005525", "GO:000
## "GO:0034975,GO:0072546", "GO:0006177,GO:0005829,GO:0016462,GO:0003922,GO:0005524", "GO:0097036,GO:000
## "GO:0010972,GO:0005829,GO:0005634,GO:0005737", "GO:0070481,GO:0070478,GO:0006402,GO:0000178,GO:00058
## "GO:0006627,GO:0031314,GO:0005758,GO:0005739,GO:0005829,GO:0005634", "GO:0032543,GO:0019898,GO:00313
## "GO:0010388,GO:0090052,GO:0090054,GO:0006974,GO:0008180,GO:0005829,GO:0005634", "GO:0002181,GO:00226
## "GO:0002181,GO:0022625,GO:0005829,GO:0003735", "GO:0006110,GO:0006000,GO:0006003,GO:0005737,GO:00038
## "GO:0005634", "GO:0030490,GO:0005730,GO:0032040,GO:0005634,GO:0003724,GO:0004004,GO:0005524", "GO:000
## "GO:0051286,GO:0032153,GO:0005737", "GO:0032446,GO:0000151,GO:0030674", "GO:0032366,GO:0006665,GO:000
## "GO:0006383,GO:0005666,GO:0044732,GO:0005634,GO:0003677,GO:0003899", "GO:0034232,GO:0000324,GO:00092
## "GO:0035269,GO:0012505,GO:0005789,GO:0016021,GO:0005783,GO:0004169", "GO:0006384,GO:0005666,GO:00058
## "GO:0006285,GO:0045007,GO:0006307,GO:0000725,GO:0032993,GO:0005634,GO:0052820,GO:0052822,GO:0008725,GO:000
## "GO:0072450,GO:0031659,GO:1900087,GO:0031138,GO:0007089,GO:0000307,GO:0044732,GO:0005634,GO:0061575,GO:000
## "GO:0061024,GO:0032366,GO:0000328", "GO:0045836,GO:0010514,GO:0031139,GO:0000747,GO:0005576,GO:00058
## "GO:0006897,GO:0007015,GO:0030479", "GO:0000028,GO:0042274,GO:0002181,GO:0022627,GO:0005829,GO:00037
## "GO:0055129,GO:0006561,GO:0005829,GO:0005737,GO:0004735", "GO:0005829,GO:0016787", "GO:0006906,GO:000
## "GO:0002182,GO:0005737,GO:0033458", "GO:0002182,GO:0005737,GO:0033458", "GO:0002182,GO:0005737,GO:000
## "GO:0002182,GO:0005737,GO:0033433", "GO:0002182,GO:0005737,GO:0033420", "GO:0002182,GO:0005737,GO:000
## "GO:0002182,GO:0005737,GO:0033402", "GO:0002182,GO:0005737,GO:0033423", "GO:0002182,GO:0005737,GO:000
## "GO:0002182,GO:0005737,GO:0033410", "GO:0002182,GO:0005737,GO:0033452", "GO:0002182,GO:0005737,GO:000
## "GO:0042776,GO:0000276,GO:0046933,GO:0046961,GO:0005198", "GO:0007021,GO:0006457,GO:0005832,GO:00058
## "GO:0045721,GO:0043161,GO:0034657,GO:0005829,GO:0005634,GO:0005737", "GO:0000055,GO:0042273,GO:00057
## "GO:0030490,GO:0043234,GO:0005634", "GO:0019878,GO:0009085,GO:0005739,GO:0005829,GO:0005634,GO:00044
## "GO:0030466,GO:0016567,GO:0005634,GO:0061631,GO:0005524", "GO:0070072,GO:0071294,GO:0007035,GO:00161
## "GO:1990758,GO:0061641,GO:0043505,GO:0000775,GO:0034506,GO:0000780,GO:0000776,GO:0005634,GO:0019237,GO:000
## "GO:0006584,GO:0005829,GO:0005634,GO:0016206,GO:0046872", "GO:0031953,GO:0071901,GO:0051446,GO:00302
## "GO:0070981,GO:0006529,GO:0005829,GO:0005634,GO:0005737,GO:0005524,GO:0004066", "GO:0034475,GO:00004
## "GO:0005794,GO:0005737,GO:0016021,GO:0005783", "GO:0006627,GO:0051603,GO:0006508,GO:0005758,GO:00057
## "GO:0051692,GO:0009272,GO:0046379,GO:0031362,GO:0016021,GO:0005783,GO:0008496", "GO:0006751,GO:00065
## "GO:0019563,GO:0006094,GO:0042149,GO:0005829,GO:0005634,GO:0042132,GO:0046872", "GO:0006490,GO:00064
## "GO:0007166,GO:0010570,GO:0005770,GO:0005886,GO:0016021,GO:0005783", "GO:0005829,GO:0005634", "GO:200
## "GO:0045292,GO:0005686,GO:0005829,GO:0005634,GO:0003723", "GO:1903475,GO:0005826,GO:0005829,GO:00057
## "GO:0045292,GO:0005634,GO:0005737,GO:0003723,GO:0005515", "GO:0042254,GO:0002181,GO:0022625,GO:00058
## "GO:0006360,GO:0005736,GO:0005634,GO:0003899", "GO:0000395,GO:0045292,GO:0005686,GO:0005685,GO:00465
## "GO:0050983,GO:0008612,GO:0005829,GO:0005634,GO:0005737,GO:0034038", "GO:0005829,GO:0005634,GO:00082
## "GO:0070124,GO:0005739,GO:0003743,GO:0003723,GO:0005525,GO:0003924", "GO:0000917,GO:0090093,GO:00346
## "GO:0006360,GO:0006383,GO:0005736,GO:0005666,GO:0005730,GO:0005829,GO:0005634,GO:0001056,GO:0001054,
## "GO:0007155,GO:0005576,GO:0010339,GO:0005737", "GO:0005829,GO:0048038,GO:0052596,GO:0005507,GO:00525
## "GO:0002184,GO:0006406,GO:0005635,GO:0005829,GO:0005737,GO:0004004,GO:0005524", "GO:0008295,GO:00159
## "GO:0006749,GO:0019243,GO:0098754,GO:0071470,GO:0005829,GO:0005634,GO:0005737,GO:0008270,GO:0004462"
## "GO:0050732,GO:0032949,GO:2000769,GO:0090334,GO:0006468,GO:0009272,GO:0035556,GO:0000935,GO:0051286,
## "GO:0032456,GO:0043234,GO:0005829,GO:0005634,GO:0005768", "GO:0043504,GO:0006281,GO:0035861,GO:00057
## "GO:0006364,GO:0005739,GO:0005829,GO:0005634,GO:0008408", "GO:1903469,GO:0032299,GO:0005829,GO:00056

```

```

## "GO:0016021", "GO:0005829, GO:0005634, GO:0008168", "GO:0055085, GO:0016021, GO:0022857", "GO:0005829", "GO:0016024, GO:0006658, GO:0006655, GO:0016021, GO:0005783, GO:0004605", "GO:0035443, GO:0035442, GO:0005829, "GO:0005737, GO:0016021", "GO:0016570, GO:0006368, GO:0016593, GO:0005634, GO:0003723", "GO:0051127, GO:0005737, GO:0006307, GO:0005829, GO:0005634, GO:0003906, GO:0016706, GO:0046872", "GO:0034975, GO:0005794, GO:0005737, GO:0043162, GO:0006900, GO:0006623, GO:0032511, GO:0000815, GO:0005770", "GO:0000717, GO:0006367, GO:0000112, GO:0045292, GO:0005684, GO:0005737", "GO:0036297, GO:0000724, GO:0005634, GO:0008409, GO:0004528, GO:0046872, "GO:0006744, GO:0005743, GO:0005739, GO:0016709, GO:0050660", "GO:0044855, GO:0051127, GO:0045010, GO:0030150, GO:0001405, GO:0005759, GO:0015450, GO:0005524", "GO:0035551, GO:0006508, GO:0005829, GO:0005737, GO:0035861", "GO:0032543, GO:0005762, GO:0005737, GO:0003735", "", "GO:0006360, GO:0006383, GO:0006366, GO:0005730", "GO:0005829, GO:0005634", "GO:0006891, GO:0030008, GO:0005829, GO:0005634", "GO:0006605, GO:0006357, GO:0016592, GO:0005634, GO:0001104", "GO:0006189, GO:0006106, GO:0044208, GO:0005829, GO:0005634, "GO:0071789, GO:0071786, GO:0071790, GO:0034613, GO:1902408, GO:0006998, GO:0071782, GO:0032541, GO:0031965, GO:0000162, GO:0005950, GO:0005829, GO:0004049, GO:0033984, GO:0004425, GO:0004640", "GO:0042273, GO:0030666, GO:0000022, GO:0051012, GO:0098783, GO:0005875, GO:0051233, GO:0005654, GO:0005634, GO:0003777, GO:0008017, GO:0000045, GO:0006914, GO:0016236, GO:0032258, GO:0000407, GO:0034045, GO:0000329, GO:0005794, GO:0005737, GO:0052707, GO:1903255, GO:0052699, GO:1903257, GO:1903253, GO:0005788, GO:0005829, GO:0005634, GO:0044876, GO:0006364, GO:0005730, GO:0005634, GO:0008408", "GO:0006357, GO:0006366, GO:0016592, GO:0005829, GO:0005634, "GO:0044732, GO:0072686, GO:0005829, GO:0005634, GO:0032153, GO:0003723", "GO:0006338, GO:0070209, GO:0005829, GO:0033617, GO:0005739", "GO:0000432, GO:0000790, GO:0005829, GO:0005634, GO:0043565", "GO:0003333, GO:0046135, GO:0019358, GO:0008655, GO:0034356, GO:0005829, GO:0005634, GO:0005737, GO:0070635, GO:0045437, GO:0035353, GO:0005887, GO:0005783, GO:0015663", "GO:0006808, GO:0000122, GO:0005634, GO:0000978, GO:0000980, GO:0090514, GO:0090515, GO:0090516, GO:0090513, GO:0090518, GO:0032975, GO:0090452, GO:0090517, GO:1901481, "", "GO:0090002, GO:0070941, GO:0007009, GO:0035838, GO:0032126, GO:0005886, GO:0005794, GO:0005737, GO:0016021, GO:0005794, GO:0005739, GO:0005783, GO:0004024, GO:0008270", "GO:0048255, GO:0007005, GO:0005737, GO:0034182, GO:0018003, GO:2000720, GO:0019985, GO:0034087, GO:0006473, GO:0043596, GO:0035861, GO:0000790, GO:0055085, GO:0016021, GO:0022857", "GO:2000815, GO:0034599, GO:0005829, GO:0005737, GO:0003723", "GO:0006122, GO:0005750, GO:0005739, GO:0008121", "GO:0035269, GO:0000032, GO:0012505, GO:0005789, GO:0016021, GO:0043234, GO:0005730, GO:0005634", "GO:0030488, GO:0043528, GO:0005829, GO:0005634, GO:0004809, GO:0000006, GO:0044749, GO:0016021, GO:0005783, GO:0000006", "GO:0031453, GO:1990678, GO:1990679, GO:0006342, GO:19901481, GO:0070914, GO:0006271, GO:0006301, GO:1903460, GO:1900264, GO:0043596, GO:0005829, GO:0005634, GO:0043626, GO:0005829, GO:0005634", "GO:0005739, GO:0016021", "GO:0007015, GO:0051321, GO:0005628, GO:0005737", "GO:0005829, GO:0005634", "GO:0042759, GO:0005524", "GO:0051445, GO:0007131, GO:0006279, GO:0000079, GO:00051685, GO:0008654, GO:0032541, GO:0030176, GO:0005783", "GO:0016021, GO:0005783", "GO:0007006, GO:0016573, GO:0036410, GO:0005829, GO:0005634", "GO:0002181, GO:0022627, GO:0005829, GO:0003723, GO:0003723, GO:0016925, GO:0031510, GO:0005634, GO:0019948, GO:0046872, GO:0005524", "GO:2000185, GO:0016020, GO:0005737, GO:0061246, GO:2000100, GO:0034613, GO:0071470, GO:1903359, GO:1903360, GO:0035246, GO:0071521, GO:0005938, GO:0006298, GO:0032300, GO:0044732, GO:0005829, GO:0005634, GO:0030983, GO:0003677, GO:0005524", "GO:0030488, GO:0005737", "GO:0034613, GO:0006897, GO:0016579, GO:0051286, GO:0005829, GO:0005634, GO:0032153, GO:0005737, "", "GO:0008053, GO:0007005, GO:0005741, GO:0016021, GO:0005525, GO:0003924", "GO:0006031, GO:0000935, GO:0032790, GO:0070126, GO:0005739, GO:0008079", "GO:0015680, GO:0006878, GO:0034759, GO:0005829, GO:0005634, "GO:0006378, GO:0006379, GO:0005849, GO:0005847, GO:0005634", "GO:0006725, GO:0005829, GO:0005634, GO:0016725, GO:0006357, GO:0000747, GO:0044732, GO:0072686, GO:0005634, GO:0000978, GO:0000982", "GO:0034975, GO:0072547, GO:0034727, GO:0034498, GO:0006623, GO:0032258, GO:0005776, GO:0000407, GO:0005829, GO:0005634, GO:0032153, GO:0007131, GO:0030998, GO:0003677, GO:0008270", "GO:0006360, GO:0005736, GO:0005730, GO:0003899, GO:0005525, GO:0006271, GO:1904161, GO:1903460, GO:0043625, GO:0043596, GO:0005829, GO:0005634, GO:0008408, GO:0031321, GO:0006900, GO:0045324, GO:0043328, GO:0033565, GO:0005794, GO:0005829, GO:0005634, GO:0005737, GO:0045944, GO:0000122, GO:0006338, GO:0016586, GO:0000790, GO:0005829, GO:0005634", "GO:0006342, GO:0006301, GO:0031291, GO:0043087, GO:0006606, GO:0006405, GO:0005829, GO:0005096, GO:0005525, GO:0005515", "GO:0035246, GO:0042450, GO:0006526, GO:0000050, GO:0005829, GO:0004056", "GO:0055085, GO:0051286, GO:0005794, GO:0032153, GO:0006734, GO:0005829, GO:0005634, GO:0005737, GO:0046872, GO:0000210", "GO:0001932, GO:0007135, GO:0007135, GO:0043087, GO:0016192, GO:0007165, GO:0005829, GO:0005634, GO:0005096", "GO:0006468, GO:0007165, GO:0000210, GO:0006749, GO:0098754, GO:0045454, GO:0036245, GO:0034599, GO:0005739, GO:0005829, GO:0004362, GO:0050661, GO:0006360, GO:0000500, GO:0005730, GO:0044732, GO:0005829", "GO:0005829", "GO:0000027, GO:0006364, GO:0006364, GO:1902981, GO:1990077, GO:0005658, GO:0043596, GO:0005829, GO:0005634, GO:0005737, GO:0003896, GO:0051539, GO:0005737

```

```

## "GO:0016570,GO:0006368,GO:0016593,GO:0005829,GO:0005634", "GO:0002181,GO:0006430,GO:0005737,GO:00055
## "GO:0001731,GO:0002183,GO:0005850,GO:0005829,GO:0005737,GO:0003743,GO:0005525,GO:0003924", "GO:00422
## "GO:0043596,GO:0044732,GO:0005634,GO:0003688,GO:0005515", "GO:0017196,GO:0031417,GO:0005829,GO:00056
## "GO:0007166,GO:0005887,GO:0005783,GO:0004872", "GO:0005829,GO:0005634", "GO:0030148,GO:0017059,GO:00
## "GO:0006627,GO:0017087,GO:0005739,GO:0004222,GO:0046872", "GO:0070124,GO:0005739,GO:0003743", "GO:00
## "GO:0016579,GO:0032153,GO:0005737,GO:0030133,GO:0005768,GO:0005515,GO:0004843", "GO:0008270", "GO:00
## "GO:0005737,GO:0016021", "GO:0010508,GO:0045324,GO:0005942,GO:0000329,GO:0005794,GO:0005829,GO:00057
## "GO:0016573,GO:0006357,GO:0006366,GO:0006338,GO:0000124,GO:0005829,GO:0005634", "GO:0000389,GO:00002
## "GO:0016021", "GO:0061621,GO:0005829,GO:0005634,GO:0005737,GO:0004332,GO:0008270", "GO:0045292,GO:00
## "GO:1990141,GO:0000395,GO:0070317,GO:0045292,GO:0005686,GO:0000243,GO:0005685,GO:0046540,GO:0071004,
## "GO:0006433,GO:0002181,GO:0005829,GO:0043907,GO:0004827,GO:0005524", "GO:0000379,GO:0005741,GO:00002
## "GO:0031657,GO:0007096,GO:2000045,GO:0005829,GO:0005634,GO:0016538", "GO:0030490,GO:0034455,GO:00320
## "GO:0005739,GO:0016021,GO:0004519,GO:0046872", "GO:0005739", "GO:0003333,GO:0016021,GO:0015171", "GO
## "GO:0006570,GO:0006568,GO:0006558,GO:0008153,GO:0046656,GO:0046654,GO:0005829,GO:0005634,GO:0005737,
## "GO:0000917,GO:0043547,GO:0000935,GO:0005829,GO:0032153,GO:0005737,GO:0005509,GO:0005515", "GO:00169
## "GO:0006616,GO:0005784,GO:0008565", "GO:0005739,GO:0005829,GO:0005634", "GO:0030150,GO:0005742,GO:00
## "GO:0007021,GO:0006457,GO:0005832,GO:0005856,GO:0005829,GO:0005634,GO:0005737,GO:0051082,GO:0005524"
## "GO:0031142,GO:1903695,GO:0000751,GO:0071507,GO:0000750,GO:0032005,GO:0010514,GO:0000935,GO:0051286,
## "GO:1990492,GO:0045841,GO:0007094,GO:0000777,GO:1990498,GO:0033597,GO:0000778,GO:0034399,GO:0000790,
## "GO:0005634", "GO:0034613,GO:0006357,GO:0006366,GO:0016592,GO:0005829,GO:0005634,GO:0001104,GO:00055
## "GO:0032012,GO:0043001,GO:0030036,GO:0006888,GO:0006886,GO:0005794,GO:0005829,GO:0005737,GO:0016021,
## "GO:0034661,GO:0071038,GO:0071042,GO:0070481,GO:0071035,GO:0000467,GO:0070478,GO:0070651,GO:0000178,
## "GO:0006114,GO:0046168,GO:0006116,GO:0009992,GO:0071470,GO:0009331,GO:0005829,GO:0005634,GO:0005737,
## "GO:0031362,GO:0005783", "GO:0006900,GO:0045324,GO:0043328,GO:0005773,GO:0000815,GO:0005938", "GO:00
## "GO:0043111,GO:0007534,GO:0048478,GO:0033314,GO:0031297,GO:0045950,GO:0031298,GO:0044732,GO:0000790,
## "GO:0045033,GO:0030048,GO:0005779,GO:0051286,GO:0032153,GO:0017022,GO:0043495", "GO:0071966,GO:00092
## "GO:0032543,GO:0005763", "GO:0016021", "GO:0045053,GO:0016192,GO:0045324,GO:0006886,GO:0005794,GO:00
## "GO:0009231,GO:0005829,GO:0005634,GO:0008703", "GO:0006511,GO:0005829,GO:0005634,GO:0005737", "GO:00
## "GO:0043950,GO:0000747,GO:0044732,GO:0005829,GO:0005634,GO:0032153,GO:0004871", "GO:0051519,GO:00068
## "GO:0006348,GO:0071963,GO:0030702,GO:0006974,GO:0031097,GO:0051285,GO:0051020,GO:0005515", "GO:00063
## "GO:1990505,GO:0000724,GO:0016567,GO:0006281,GO:0006338,GO:0000790,GO:0005634,GO:0004003,GO:0061630,
## "GO:0042147,GO:0032045,GO:0005794,GO:0005829,GO:0005634", "GO:0016584,GO:0006342,GO:0000122,GO:00432
## "GO:0005829,GO:0005634,GO:0003959,GO:0018548,GO:0010181,GO:0052690", "GO:0006285,GO:0006307,GO:00056
## "GO:0030490,GO:0005732,GO:0005730,GO:0044732,GO:0032040,GO:0005829,GO:0005634,GO:0032153,GO:0030515"
## "GO:0034613,GO:0010971,GO:0031028,GO:0044732,GO:0005515", "GO:0043001,GO:0030120,GO:0005829,GO:00056
## "GO:0000329,GO:0005794,GO:0005783,GO:0046872", "GO:0032543,GO:0005743,GO:0005739,GO:0005525,GO:00039
## "GO:1902975,GO:0042555,GO:0097373,GO:0005656,GO:0031261,GO:0043596,GO:0000790,GO:0005634,GO:0017116,
## "GO:0045039,GO:0042721,GO:0005739,GO:0016021,GO:0008565", "GO:0051083,GO:0005854,GO:0042788,GO:00058
## "GO:0007021,GO:0006457,GO:0005832,GO:0005856,GO:0005829,GO:0005634,GO:0005737,GO:0051082,GO:0005524"
## "GO:0031321,GO:0006906,GO:0006893,GO:0030437,GO:0006886,GO:0005628,GO:0005484", "GO:0006656,GO:00057
## "GO:0000054,GO:0006364,GO:0040010,GO:0005730,GO:0005634,GO:0003723,GO:0005525,GO:0003924", "GO:00063
## "GO:0005829,GO:0005634", "GO:0015680,GO:0006878,GO:0033617,GO:0007005,GO:0005758,GO:0005829,GO:00056
## "GO:0045040,GO:0000002,GO:0032865,GO:0030176,GO:0005737,GO:0005783", "GO:0071459,GO:0000070,GO:00345
## "GO:1990548,GO:0032592,GO:0005739,GO:0015230", "GO:0045039,GO:0030150,GO:0045040,GO:0005742,GO:00160
## "GO:0070650,GO:0060305,GO:0032231,GO:2000784,GO:0060583,GO:0032956,GO:0034613,GO:0043087,GO:0007165,
## "GO:0042843,GO:0019568,GO:0005829,GO:0005634,GO:0032866,GO:0032867,GO:0004033,GO:0004032", "GO:00450
## "GO:0006366,GO:0005665,GO:0005829,GO:0005634,GO:0003677,GO:0003899,GO:0046872,GO:0005515", "GO:00066
## "GO:0051292,GO:0005643,GO:0005829,GO:0005634", "GO:0006913,GO:0005643,GO:0005635,GO:0016021", "GO:00
## "GO:1902969,GO:0033314,GO:0005656,GO:0031261,GO:0005664,GO:0043596,GO:0000790,GO:0005634,GO:0003688
## "GO:0043486,GO:0006338,GO:0000812,GO:0005829,GO:0005634,GO:0046872", "GO:0006364,GO:0005730,GO:00320
## "GO:0007004,GO:0000723,GO:0000333,GO:0000784,GO:0005634,GO:0003721,GO:0003677,GO:0003723,GO:0046872"
## "GO:1901583,GO:0005886,GO:0005794,GO:0016021,GO:0005783,GO:1901584", "GO:0002181,GO:0005730,GO:00226
## "GO:0045944,GO:0006457,GO:0005737", "GO:1990273,GO:1990276", "GO:0005829,GO:0005634", "GO:1902985,GO
## "GO:0006627,GO:0042720,GO:0005739,GO:0016021,GO:0008236,GO:0004222", "GO:0032543,GO:0005762,GO:00057

```

```

## "GO:0070649,GO:0061572,GO:1903475,GO:1903480,GO:1904601,GO:0090337,GO:0016459,GO:0005826,GO:1990819,0
## "GO:0016567,GO:0005829,GO:0005634,GO:0061631,GO:0005524,GO:0005515", "GO:0032543,GO:0005762,GO:000573
## "GO:0000083,GO:0051091,GO:0071931,GO:0005634,GO:0001190,GO:0046872,GO:0005515", "GO:0045292,GO:000573
## "GO:0044755,GO:0055085,GO:0005886,GO:0005794,GO:0016021,GO:0015233", "GO:0046856,GO:0048017,GO:000683
## "GO:0032543,GO:0005763,GO:0005739,GO:0003735", "GO:0009257,GO:0009113,GO:0046656,GO:0006164,GO:00359
## "GO:0031167,GO:0005739,GO:0008650", "GO:0000032,GO:0006486,GO:0009298,GO:0005829,GO:0005634,GO:00082
## "GO:0046938,GO:0006750,GO:0071585,GO:0071276,GO:0070219,GO:0042762,GO:0005739,GO:0048038,GO:0071949,
## "GO:0002098,GO:0032543,GO:0005739,GO:0050660", "GO:0006367,GO:0006289,GO:0006355,GO:0000439,GO:00056
## "GO:0006729,GO:0005829,GO:0005634,GO:0004757", "GO:0006402,GO:0045292,GO:0006364,GO:0005688,GO:00465
## "GO:0006506,GO:0000506,GO:0016021,GO:0005783,GO:0017176", "GO:0032543,GO:0005763,GO:0005739,GO:00037
## "GO:0045944,GO:0005635,GO:0005829,GO:0005634,GO:0000978,GO:0000982", "GO:1990809,GO:0034613,GO:00325
## "GO:0016973,GO:0005643,GO:0034399,GO:0005635,GO:0005829,GO:0005634,GO:0005737,GO:0031369", "GO:00422
## "GO:0010780,GO:0005829,GO:0005634", "GO:0006357,GO:0016592,GO:0005635,GO:0005829,GO:0005634,GO:00011
## "GO:0006893,GO:0030136,GO:0030140,GO:0005829,GO:0005634,GO:0032153,GO:0005737", "GO:0072686,GO:00056
## "GO:0042744,GO:0006790,GO:0045454,GO:0019430,GO:0070301,GO:0071470,GO:0005739,GO:0005829,GO:0005634,
## "GO:1990758,GO:0007052,GO:0042729", "GO:0001933,GO:0031568,GO:1904030,GO:0071901,GO:0045736,GO:00309
## "GO:0016021", "GO:0016192,GO:0005829,GO:0005634", "GO:00000338,GO:0006508,GO:0005634,GO:0005737,GO:00
## "GO:1903655,GO:1990443,GO:0038083,GO:0036289,GO:0018105,GO:0018107,GO:0008024,GO:0000307,GO:0005634,
## "GO:0016021", "GO:0000018,GO:0060542,GO:0032508,GO:0031146,GO:0000724,GO:0006310,GO:0016567,GO:00312
## "GO:0006401,GO:0005730,GO:0000790,GO:0005634,GO:0004523,GO:0000287", "GO:0006384,GO:0000127,GO:00056
## "GO:0006627,GO:0042720,GO:0005739,GO:0016021,GO:0008236,GO:0004222", "GO:0000289,GO:0007165,GO:00312
## "GO:0006508,GO:0009272,GO:0005783,GO:0004181,GO:0008270", "GO:0016567,GO:0051321,GO:0031386,GO:00055
## "GO:0006656,GO:0016021,GO:0005783,GO:0000773,GO:0004608,GO:0080101", "GO:0006490,GO:0006486,GO:00003
## "GO:0051286,GO:0005525", "GO:0006620,GO:0006616,GO:0005784,GO:0005794,GO:0016021,GO:0005783,GO:00430
## "GO:0016021", "GO:0061621,GO:0005829,GO:0051287,GO:0050661,GO:0004365", "GO:0043087,GO:0007165,GO:00
## "GO:0015629,GO:0005829,GO:0005634,GO:0016829", "GO:1903711,GO:1903710,GO:0005886,GO:0016021,GO:00156
## "GO:0033386,GO:0016117,GO:0033384,GO:0045337,GO:0006696,GO:0008299,GO:0005829,GO:0005634,GO:0004311,
## "GO:0016021", "GO:0015914,GO:0071852,GO:0051286,GO:0005829,GO:0005634,GO:0032153,GO:0005737,GO:00085
## "GO:0030433,GO:0005789,GO:0016021,GO:0005783", "GO:1902969,GO:0005730,GO:0005634,GO:0005737,GO:00468
## "GO:0005739,GO:0016021", "GO:0019358,GO:0005777,GO:0005829,GO:0005634,GO:0005737,GO:0008936,GO:00468
## "GO:0035066,GO:0006334,GO:0000790,GO:0005634,GO:0042393", "GO:0071263,GO:1990497,GO:0031571,GO:00718
## "GO:0046512,GO:0030148,GO:0016021,GO:0005783,GO:0042284", "GO:0005829,GO:0005634", "GO:0030490,GO:00
## "GO:0005739,GO:0016021,GO:0008757", "GO:0045041,GO:0005758,GO:0008270", "GO:0006501,GO:0000045,GO:00
## "GO:0006888,GO:0006886,GO:0030008,GO:0005801,GO:0005794,GO:0005829,GO:0005634,GO:0005783", "GO:00021
## "GO:0030150,GO:0006457,GO:0001405,GO:0051087,GO:0051082,GO:0000774", "GO:0034428,GO:0000290,GO:00310
## "GO:0006506,GO:0000506,GO:0005783,GO:0008194,GO:0017176", "GO:0000288,GO:0006396,GO:0000290,GO:00058
## "GO:0034975,GO:0005794,GO:0016021,GO:0005783,GO:0003756", "GO:0002181,GO:0022627,GO:0005829,GO:00037
## "GO:0006289,GO:0006295,GO:1901255,GO:0005634,GO:0004519,GO:0003697,GO:0046872", "GO:0034975,GO:00160
## "GO:0006470,GO:0007165,GO:0043234,GO:0005634,GO:0005737,GO:0004722,GO:0046872", "GO:0006696,GO:00057
## "GO:1902358,GO:0005886,GO:0005829,GO:0005737,GO:0016021,GO:0008271,GO:0015116", "GO:0006103,GO:00060
## "GO:0002098,GO:0033588,GO:0005829,GO:0005634", "GO:0030036,GO:0005884,GO:0005829,GO:0005634,GO:00510
## "GO:0005634", "GO:0032511,GO:0006888,GO:0006886,GO:0030127,GO:0005829,GO:0005783,GO:0008270", "GO:00
## "GO:0006501,GO:0016236,GO:0032258,GO:0051321,GO:0000407,GO:0034045,GO:0005829,GO:0005737", "GO:00064
## "GO:0060542,GO:0031146,GO:0007346,GO:0030163,GO:0000920,GO:0042787,GO:0045841,GO:0006998,GO:0017117,
## "GO:0000070,GO:0000941,GO:0000779,GO:0031617,GO:0005634,GO:0000444,GO:0005515", "GO:0009152,GO:00460
## "GO:0005829,GO:0005634", "GO:0006644,GO:0019932,GO:0000329,GO:0016021,GO:0005783,GO:0008195,GO:00008
## "GO:0009099,GO:0009098,GO:0009097,GO:0005759,GO:0005829,GO:0005634,GO:0005737,GO:0052656,GO:0052655,
## "GO:0005739", "GO:0019344,GO:0006555,GO:0006535,GO:0071269,GO:0071266,GO:0071268,GO:0005829,GO:00056
## "GO:2000719,GO:0034183,GO:0005694,GO:0005634", "GO:1902985,GO:1903466,GO:0005656,GO:0000790,GO:00057
## "GO:0055085,GO:0016021,GO:0005783,GO:0022857", "GO:0015937,GO:0005829,GO:0005634,GO:0005737,GO:00045
## "GO:1990539,GO:0098708,GO:0005887,GO:0005886,GO:0005794,GO:0055054,GO:0005358", "GO:0000395,GO:00452
## "GO:0006487,GO:0005794,GO:0005737,GO:0005783,GO:0008375", "GO:0006487,GO:0005794,GO:0005829,GO:00056
## "GO:0042254,GO:0005730,GO:0005634,GO:0004004,GO:0005524", "GO:0032543,GO:0005762,GO:0005739,GO:00037
## "GO:0034975,GO:0005789,GO:0005783,GO:0016671,GO:0003756,GO:0015035", "GO:0050821,GO:0042026,GO:00433

```

```

## "GO:1901522,GO:0005829,GO:0005634,GO:0000978,GO:0000982,GO:0008270", "GO:0007005,GO:0005737", "GO:00048026,GO:0046777,GO:1900182,GO:0006468,GO:0035556,GO:0046822,GO:1900181,GO:0044732,GO:0005829,GO:0035445,GO:0000324,GO:0005886,GO:0016021,GO:0005452,GO:0046715", "GO:0006364,GO:0005730,GO:0005632,GO:0005829,GO:0005634", "GO:0032888,GO:0007096,GO:0010972,GO:0031291,GO:1901673,GO:0046827,GO:000726,GO:0005829,GO:0005634", "GO:0042273,GO:0005829,GO:0016887,GO:0005524", "GO:0000288,GO:0005737,GO:0007131,GO:0005829,GO:0005634", "GO:0006646,GO:0046854,GO:0048015,GO:0048017,GO:0030036,GO:000588,GO:0046488,GO:0046856,GO:0048017,GO:0006897,GO:0005737,GO:0004439,GO:0004445", "GO:0006591,GO:001092,GO:0032118,GO:0007129,GO:0030989,GO:0034613,GO:0032117,GO:0035974,GO:0044732,GO:0005634,GO:0042802,GO:00070647,GO:0080008,GO:0005829", "GO:1903710,GO:0005886,GO:0005794,GO:0016021,GO:0000297", "GO:0006696,GO:0008299,GO:0042175,GO:0005789,GO:0016021,GO:0005783,GO:0051996,GO:0004310,GO:0005515", "GO:0007265,GO:0046580,GO:0032005,GO:0043087,GO:0005829,GO:0005634,GO:0005096", "GO:0035308,GO:20012,GO:0000715,GO:0070914,GO:0006289,GO:1901255,GO:0000109,GO:0003684,GO:0046872", "GO:0000747,GO:005126,GO:", "GO:0006383,GO:0005666,GO:0005737,GO:0003677,GO:0032549,GO:0003899,GO:0008270", "GO:0016570,GO:0032790,GO:0070125,GO:0005739,GO:0003746,GO:0005525,GO:0003924", "GO:0000723,GO:0000784,GO:0005829,GO:0009051,GO:0006091,GO:0019521,GO:0005739,GO:0005829,GO:0005737,GO:0050661,GO:0004616", "GO:001602,GO:0005829,GO:0005634", "GO:1902975,GO:0005656,GO:0031261,GO:0005664,GO:0043596,GO:0044732,GO:000079,GO:0051286,GO:0005829,GO:0005634,GO:0032153,GO:0005737", "GO:0030472,GO:0045144,GO:0051315,GO:003352,GO:0033127,GO:0042138,GO:0006302,GO:0031573,GO:0044773,GO:0000723,GO:0006281,GO:0030870,GO:0000784,GO:0071897,GO:0051103,GO:0005634,GO:0003910,GO:0003677,GO:0005524", "GO:0006797,GO:0000329,GO:0005737,GO:0000160,GO:0010972,GO:0007165,GO:0034599,GO:0032874,GO:0007234,GO:0005829,GO:0005634,GO:0005515,GO:1990542,GO:0005739,GO:0016021", "GO:0033217,GO:0006109,GO:0016602,GO:0005634,GO:0000979,GO:000107,GO:0006360,GO:0000120,GO:0070860,GO:0044732,GO:0072686,GO:0005829,GO:0005634,GO:0032153,GO:0001169,GO:0016020,GO:0005794,GO:0016021,GO:0005783,GO:0015144", "GO:0030010,GO:0051666,GO:0005938,GO:0005829,GO:0000778,GO:0000799,GO:0005730,GO:0005634,GO:0005737,GO:0016887", "GO:0019432,GO:004633,GO:0005829,GO:0005634,GO:0005737,GO:0001135,GO:0001191", "GO:0016584,GO:0035391,GO:0006348,GO:0030472,GO:0005737,GO:0000127,GO:0005829,GO:0005634", "GO:0009052,GO:0030466,GO:0031048,GO:0034613,GO:0030958,GO:0031618,GO:0031934,GO:1990707,GO:0044732,GO:0016021,GO:0005783", "GO:0089722,GO:0005794,GO:0016021,GO:0005783,GO:1902406,GO:0000281,GO:0005826,GO:0051286,GO:0005829,GO:0005634,GO:0032153,GO:0005737,GO:0005543,GO:0071852,GO:0006506,GO:0000329,GO:0005794,GO:0005789,GO:0016021,GO:0051377", "GO:0070124,GO:0005737,GO:0009257,GO:0009113,GO:0046656,GO:0006164,GO:0035999,GO:0006730,GO:0005829,GO:0004329,GO:0004488,GO:0034499,GO:0006886,GO:0031902,GO:0005794,GO:0005829,GO:0005634,GO:0005768,GO:0032266", "GO:003254,GO:0006633,GO:0005739,GO:0005634,GO:0004315", "GO:1902983,GO:0044806,GO:0032508,GO:1903469,GO:0006348,GO:0034203,GO:0005789,GO:0016021,GO:0005783,GO:0005319", "GO:0006465,GO:0045047,GO:0005787,GO:001602,GO:0006383,GO:0000127,GO:0005829,GO:0005634", "GO:0045292,GO:0005686,GO:0071004,GO:0005681,GO:0005829,GO:0005737,GO:0000127,GO:0005829,GO:0005634", "GO:0042125,GO:0009272,GO:0000139,GO:0016021,GO:0005783,GO:0031278", "GO:0016259,GO:0006555,GO:0006737,GO:0045132,GO:0051455,GO:0031390,GO:0043596,GO:0005829,GO:0005634,GO:0003689,GO:0005524", "GO:0001578,GO:0051013,GO:0051286,GO:0005634,GO:0032153,GO:0005524,GO:0008568", "GO:0016197,GO:0016192,GO:0031505,GO:0006886,GO:0030130,GO:0030132,GO:0030125,GO:0031410,GO:0005794,GO:0005737,GO:0006367,GO:0006355,GO:0010971,GO:0000079,GO:0005675,GO:0070985,GO:0005634,GO:0016538,GO:0005634", "GO:0006457,GO:0005739,GO:0005789,GO:0004499,GO:0050661,GO:0050660", "GO:0046168,GO:0007006,GO:003204,GO:0051567,GO:0034630,GO:0051572,GO:0030466,GO:0030702,GO:0031507,GO:0000070,GO:0006338,GO:0031618,GO:0045292,GO:0000974,GO:0005681,GO:0005634,GO:0005515", "GO:0000054,GO:0002181,GO:0022627,GO:0005829,GO:0006360,GO:0005736,GO:0003677,GO:0032549,GO:0003899,GO:0046872", "GO:0008608,GO:1990758,GO:0016573,GO:0032968,GO:0006368,GO:0045944,GO:0032783,GO:0005829,GO:0005634,GO:0005515", "GO:0006627,GO:001702,GO:0005730,GO:0005829,GO:0005634", "GO:0030100,GO:0008360,GO:0007165,GO:0000329,GO:0005737,GO:00046872,GO:0016584,GO:0030466,GO:0030702,GO:0000183,GO:0000122,GO:0006338,GO:0070824,GO:0033553,GO:0031618,GO:0006511,GO:0005829,GO:0005634", "GO:0051205,GO:0045039,GO:0033617,GO:0005743,GO:0005739,GO:001602,GO:0000122,GO:0042149,GO:0072686,GO:0005634,GO:0001078,GO:0000978,GO:0046872", "GO:0006508,GO:0005737,GO:0006482,GO:0005829,GO:0005634,GO:0051722", "GO:0005739", "GO:0030433,GO:0006457,GO:0006886,GO:0002182,GO:0022625,GO:0005829,GO:0003735", "GO:0016570,GO:0000790,GO:0005829,GO:0005634,GO:001802,GO:1902975,GO:0008622,GO:0031261,GO:0000790,GO:0005829,GO:0005634,GO:0003887,GO:0003677", "GO:0006348

```

```

## "GO:0031167,GO:0006364,GO:0005730,GO:0005634,GO:0008173,GO:0008757,GO:0003723", "GO:0060628,GO:00165
## "GO:0051286,GO:0032153,GO:0005737", "GO:0016567,GO:0051286,GO:0044732,GO:0005634,GO:0032153,GO:00616
## "GO:0006790,GO:0005829,GO:0005634,GO:0046872,GO:0004065", "", "GO:0055085,GO:0005886,GO:0016021,GO:00
## "GO:0005737,GO:0016021,GO:0005783,GO:0016787", "GO:0006541,GO:0005829,GO:0005634,GO:0005737,GO:00167
## "GO:0046835,GO:0006012,GO:0033499,GO:0005829,GO:0005737,GO:0005524,GO:0004335", "GO:0009897,GO:00160
## "GO:0016021,GO:0005783", "GO:0006744,GO:0008299,GO:0032476,GO:0005743,GO:0005739,GO:0008834,GO:00468
## "GO:0002182,GO:0005737,GO:0033446", "GO:0002182,GO:0005737,GO:0033446", "GO:0002182,GO:0005737,GO:00
## "GO:0002182,GO:0005737,GO:0033404", "GO:0002182,GO:0005737,GO:0033420", "GO:0002182,GO:0005737,GO:00
## "GO:0002182,GO:0005737,GO:0033452", "GO:0002182,GO:0005737,GO:0033452", "GO:0002182,GO:0005737,GO:00
## "GO:0002182,GO:0005737,GO:0033417", "GO:0002182,GO:0005737,GO:0033417", "GO:0002181,GO:0005737,GO:00
## "GO:0002182,GO:0005737,GO:0033445", "GO:0002182,GO:0005737,GO:0033423", "GO:0002182,GO:0005737,GO:00
## "GO:0048250,GO:0055085,GO:0006880,GO:0005739,GO:0016021,GO:0005381", "GO:0006354,GO:0006360,GO:00063
## "GO:0034975,GO:0072546,GO:0005783", "GO:0045892,GO:0006353,GO:0000790,GO:0005634", "GO:0044242,GO:00
## "GO:0018283,GO:0072592,GO:1901967,GO:0006879,GO:0005759,GO:0005739,GO:0004322,GO:0008199,GO:0005506"
## "GO:0016021", "GO:0006884,GO:0006874,GO:0070588,GO:0097038,GO:0016021,GO:0005783,GO:0015275,GO:00055
## "GO:0002098,GO:0033588,GO:0005829,GO:0005634,GO:0005737", "GO:0002183,GO:0005851,GO:0005829,GO:00056
## "GO:0018022,GO:0042254,GO:0005730,GO:0005829,GO:0005634,GO:0016279", "GO:0002181,GO:0022625,GO:00058
## "GO:0032968,GO:0006368,GO:0045944,GO:0032783,GO:0005515", "GO:0000173,GO:0006470,GO:0030968,GO:00714
## "GO:0006747,GO:0046444,GO:0005829,GO:0005634,GO:0005737,GO:0005524,GO:0003919", "GO:0070869,GO:00007
## "GO:1902361,GO:0031305,GO:0005739,GO:0050833", "GO:0030437,GO:0005829,GO:0005634,GO:0005737", "GO:190
## "GO:0006508,GO:0030176,GO:0008233,GO:0046872", "GO:0006360,GO:0006353,GO:0005736,GO:0005730,GO:00058
## "GO:0043087,GO:0016192,GO:0007165,GO:0005829,GO:0005634,GO:0005096", "GO:0007039,GO:0045721,GO:00071
## "GO:0031119,GO:0005634,GO:0009982,GO:0003723,GO:0046872", "GO:0010570,GO:0016568,GO:0016573,GO:00063
## "GO:0005739,GO:0005829,GO:0005634,GO:0005737", "GO:0016575,GO:0000790,GO:0005829,GO:0005634,GO:00001
## "GO:0006636,GO:0006633,GO:0031227,GO:0005789,GO:0016021,GO:0005783,GO:0004768,GO:0046872", "GO:00345
## "", "GO:0000965,GO:0005759,GO:0005739,GO:0008859,GO:0003723", "GO:0031292,GO:0000736,GO:0000729,GO:00
## "GO:0016021,GO:0005783", "GO:0045931,GO:0051445,GO:0046777,GO:0032092,GO:0010972,GO:0031098,GO:00057
## "GO:0007034,GO:0042144,GO:0016237,GO:0000329,GO:0016021,GO:0005783", "GO:0002181,GO:0005730,GO:00226
## "GO:0016021", "GO:0006357,GO:0005829,GO:0005634,GO:0043565,GO:0000978,GO:0000982,GO:0046872", "GO:00
## "GO:0000750,GO:0000747,GO:0007264,GO:0005829,GO:0005085", "GO:0000778,GO:0005829,GO:0005634", "GO:190
## "GO:0005739", "GO:0006659,GO:0010512,GO:0006646,GO:0070317,GO:0071852,GO:0030176,GO:0005829,GO:00057
## "GO:0006364,GO:0044732,GO:0005829,GO:0005634", "GO:0002182,GO:0005853,GO:0005829,GO:0005085,GO:00037
## "GO:0007165,GO:0045842,GO:0005829,GO:0004674,GO:0005524", "GO:0005739,GO:0005829,GO:0005634,GO:00350
## "GO:0016579,GO:0005829,GO:0005634,GO:0005737,GO:0004843", "GO:0006364,GO:0005730,GO:0005829,GO:00040
## "GO:0005829,GO:0005634,GO:0005737", "GO:0055085,GO:0016021,GO:0005783,GO:0022857", "GO:0006402,GO:00
## "GO:0006104,GO:0006099,GO:0042709,GO:0005739,GO:0004776,GO:0005525,GO:0005524", "GO:00329
## "GO:0045292,GO:0046540,GO:0005634,GO:0003723", "GO:0005829,GO:0005634", "GO:0005829,GO:0016491", "GO
## "GO:0006468,GO:0007165,GO:0004674,GO:0005524", "GO:0044211,GO:0044206,GO:0006206,GO:0008655,GO:00447
## "GO:0006437,GO:0002181,GO:0005829,GO:0005737,GO:0004831,GO:0005524", "GO:0032958,GO:0005829,GO:00056
## "GO:0055085,GO:0016021,GO:0005783,GO:0042626,GO:0046872,GO:0005524", "GO:0045048,GO:0016192,GO:00058
## "GO:0016021", "GO:0006367,GO:0006289,GO:0006281,GO:0000439,GO:0005675,GO:0005829,GO:0005634,GO:00009
## "GO:0016021,GO:0005783,GO:0035091", "GO:0045842,GO:0008540,GO:0005829,GO:0005634,GO:0004175,GO:00168
## "GO:0006696,GO:0005739,GO:0005789,GO:0016021,GO:0009055,GO:0020037,GO:0046872", "GO:0032543,GO:00057
## "GO:0005739,GO:0016021", "GO:0044732,GO:0005634,GO:0032153", "GO:0034243,GO:0006369,GO:0000122,GO:00
## "GO:0009051,GO:0005829,GO:0005634,GO:0005737,GO:0017057", "GO:0000083,GO:0007089,GO:0007165,GO:00427
## "GO:0006379,GO:0000172,GO:0005829,GO:0005634", "GO:0005829,GO:0005634", "", "GO:1902969,GO:0008622,GO
## "GO:0005829,GO:0005634,GO:0016491", "GO:0009060,GO:0005751,GO:0005739,GO:0004129", "GO:0045921,GO:00
## "GO:0006406,GO:0005643,GO:0034399,GO:0005635,GO:0005829,GO:0005634", "GO:0000324,GO:0016021", "GO:00
## "GO:0072480,GO:0045841,GO:0007094,GO:0033597,GO:0000778,GO:0005829,GO:0005634,GO:0010997,GO:0061637
## "GO:0032543,GO:0005763,GO:0005739,GO:0005829,GO:0005634,GO:0019843,GO:0003735", "GO:0007127,GO:00165
## "GO:0016567,GO:0005794,GO:0005829,GO:0061630,GO:0008270", "GO:0000920,GO:0006887,GO:0048278,GO:00068
## "GO:0009443,GO:0005829,GO:0005634,GO:0008478,GO:0046872,GO:0005524", "GO:0051568,GO:0070869,GO:00481
## "GO:0006099,GO:0006106,GO:0045239,GO:0005759,GO:0005829,GO:0004333", "GO:0006487,GO:0008250,GO:00160
## "GO:0070647,GO:0006511,GO:0000151,GO:0005829,GO:0005737,GO:0030674,GO:0005515", "GO:0016973,GO:00066

```

```

## "GO:0006006,GO:0005829,GO:0005634,GO:0005737,GO:0008973,GO:0004614,GO:0000287", "GO:0042776,GO:00002
## "GO:0015833,GO:0055085,GO:0005887,GO:0005829,GO:0005634,GO:0022857", "GO:0031030,GO:0031029,GO:00058
## "GO:0006897,GO:0016579,GO:0005794,GO:0005829,GO:0005634,GO:0030133,GO:0005515,GO:0004843", "GO:00071
## "", "", "GO:0098716,GO:0005887,GO:0005783,GO:0044750", "GO:0055085,GO:0046618,GO:0005886,GO:0016021,
## "GO:0006913,GO:0005643,GO:0034399,GO:0005634", "GO:0006428,GO:0032543,GO:0005739,GO:0005829,GO:00048
## "GO:0005829,GO:0005634,GO:0043130", "GO:0005794,GO:0005737,GO:0016021", "GO:0016021", "GO:0005829,GO
## "GO:0005739,GO:0005829,GO:0005634", "GO:0034484,GO:0005987,GO:0005576,GO:0000324,GO:0004564,GO:00045
## "GO:0032543,GO:0005762,GO:0005739,GO:0003735", "GO:0042273,GO:0042256,GO:0005730,GO:0005829,GO:00056
## "GO:0006359,GO:0006383,GO:0005634,GO:0000126,GO:0003677,GO:0001026", "GO:0031497,GO:0006336,GO:00718
## "GO:0019674,GO:0006741,GO:0005829,GO:0005634,GO:0003951", "GO:0005829,GO:0005634,GO:0005737", "GO:000
## "GO:0045041,GO:0005741,GO:0005739,GO:0016887,GO:0005524", "GO:0006368,GO:0006406,GO:0000347,GO:00058
## "GO:0005829,GO:0005634,GO:0016491", "GO:0006098,GO:0006094,GO:0061621,GO:0005829,GO:0005634,GO:00048
## "GO:0005737,GO:0016021", "GO:0016485,GO:0043161,GO:0005783", "GO:0006402,GO:0045292,GO:0030490,GO:000
## "GO:0006383,GO:0005666,GO:0005829,GO:0005634,GO:0003677,GO:0003899", "GO:0006913,GO:0005643,GO:00343
## "GO:0007165,GO:0005829,GO:0005634,GO:0019992", "GO:0005634", "GO:0035435,GO:0005887,GO:0005794,GO:000
## "GO:0006078,GO:0009272,GO:0000329,GO:0030176,GO:0005783", "GO:0042273,GO:0005730,GO:0005634", "GO:000
## "GO:0030433,GO:0006888,GO:0006886,GO:0030127,GO:0005794,GO:0005783,GO:0005096,GO:0008270", "GO:00002
## "GO:0006783,GO:0005743,GO:0005829,GO:0005634,GO:0004325,GO:0051537,GO:0046872", "GO:0006614,GO:00057
## "GO:0006290,GO:0006289,GO:0016567,GO:0000109,GO:0005634,GO:0004003,GO:0003677,GO:0061630,GO:0008270,
## "GO:0006364,GO:0042274,GO:0030686,GO:0005730,GO:0005829,GO:0005634", "GO:0006370,GO:0001510,GO:00165
## "GO:0061077,GO:0005739", "GO:1902391,GO:0017196,GO:0031415,GO:0005737,GO:0010698,GO:0005515", "GO:000
## "GO:0018279,GO:0006487,GO:0008250,GO:0016021,GO:0005783,GO:0004579", "GO:0042787,GO:0000723,GO:19034
## "GO:0006336,GO:0000790,GO:0005829,GO:0042393,GO:0005515", "GO:0006564,GO:0005829,GO:0016597,GO:00512
## "GO:0002182,GO:0022626,GO:0005829,GO:0003746,GO:0016887,GO:0005524", "GO:0006357,GO:0005634,GO:00160
## "GO:0034638,GO:0005789,GO:0005737,GO:0016021,GO:0004622,GO:0052689", "GO:0006782,GO:0006783,GO:00058
## "GO:0055085,GO:0000329,GO:0016021,GO:0015297", "GO:0045292,GO:0005681,GO:0005634", "GO:0034613,GO:19
## "GO:0070262,GO:0035970,GO:0000173,GO:0032873,GO:0005829,GO:0005634,GO:1990439,GO:0004724,GO:0046872"
## "GO:0032543,GO:0005762,GO:0005739,GO:0003735", "GO:0033169,GO:0033193,GO:0005634,GO:0031491,GO:00082
## "GO:0009231,GO:0031119,GO:0005739,GO:0005829,GO:0005634,GO:0005737,GO:0008835,GO:0009982,GO:0008270
## "GO:0015758,GO:0098655,GO:0035428,GO:1990539,GO:0098708,GO:0015992,GO:0005887,GO:0031520,GO:0005886,
## "GO:0005829,GO:0005634,GO:0005737,GO:0004040,GO:0016884", "GO:0000329,GO:0005794,GO:0005783,GO:00080
## "GO:0008298,GO:0006342,GO:0042175,GO:0005829,GO:0005737,GO:0005783,GO:0003677,GO:0003723", "GO:00422
## "GO:0042254,GO:0005829,GO:0005737,GO:0005525,GO:0003924", "GO:1902981,GO:1902975,GO:0005658,GO:00007
## "GO:0046084,GO:0006189,GO:0005634,GO:0004644,GO:0008168", "", "GO:0006151,GO:0005829,GO:0005634,GO:00
## "GO:0006890,GO:0006886,GO:0030126,GO:0005829,GO:0005634", "GO:0002181,GO:0022627,GO:0032040,GO:00058
## "GO:2000765,GO:0017183,GO:0005829,GO:0005634,GO:0005737,GO:0004164", "GO:0007165,GO:0071852,GO:00057
## "GO:0002184,GO:0018444,GO:0005829,GO:0003747,GO:0005525,GO:0003924", "GO:0006906,GO:0006904,GO:00512
## "GO:0045141,GO:0005635,GO:0016021,GO:0005783,GO:0005515", "GO:0006367,GO:0016573,GO:0006355,GO:00063
## "GO:0016573,GO:0006357,GO:0006338,GO:0000124,GO:0000790,GO:0005829,GO:0005634,GO:0003712", "GO:00058
## "GO:0002181,GO:0005730,GO:0022625,GO:0005829,GO:0005634,GO:0019843,GO:0003735", "GO:0016074,GO:00056
## "GO:0016021", "GO:0016021,GO:0005783", "GO:0005737,GO:0016021", "GO:0016021,GO:0005783", "GO:0016021
## "GO:0016192,GO:0005737,GO:0005096,GO:0046872", "GO:0005634,GO:0005737", "GO:0090052,GO:0033696,GO:000
## "GO:0009272,GO:0030476,GO:0070591,GO:0005618,GO:0031362,GO:0046658,GO:0005886,GO:0005783,GO:0004556,0
## "GO:0006487,GO:0043048,GO:0030176,GO:0005789,GO:0005783,GO:0004168", "GO:0000354,GO:0005681,GO:00056
## "GO:0035023,GO:0030994,GO:0010590,GO:0051666,GO:0043087,GO:0090334,GO:0043547,GO:0060624,GO:0005826,
## "GO:0005634", "GO:0034244,GO:0005634,GO:0008270", "GO:0006457,GO:0043234,GO:0005829,GO:0005634,GO:000
## "GO:0005829,GO:0005634,GO:0016491", "GO:0005829,GO:0005634,GO:0016491", "GO:0030488,GO:0005829,GO:000
## "GO:0032780,GO:0005753,GO:0005634,GO:0042030", "GO:0006562,GO:0097054,GO:0005759,GO:0004657", "GO:000
## "GO:0006491,GO:0042125,GO:0006493,GO:0006487,GO:0009272,GO:0000139,GO:0005794,GO:0016021,GO:0031278
## "GO:0051447,GO:0033621,GO:0033620,GO:1990251,GO:0005634,GO:0003729,GO:0003723,GO:0005515", "GO:00164
## "GO:1903459,GO:0005658,GO:0043596,GO:0005829,GO:0005634,GO:0033203,GO:0043141,GO:0005524", "GO:000000
## "GO:0003333,GO:0005737,GO:0016021,GO:0015171", "GO:0002181,GO:0022625,GO:0005829,GO:0003735", "GO:000
## "GO:0034661,GO:0071038,GO:0071042,GO:0070481,GO:0071035,GO:0000467,GO:0070478,GO:0070651,GO:0000178,
## "GO:0006357,GO:0005829,GO:0005634,GO:0000978,GO:0000982,GO:0008270", "GO:0005829,GO:0005634,GO:00084

```



```

## Build GO DAG topology ..... ( 874 GO terms and 1842 relations. )
##
## Annotating nodes ..... ( 5190 genes annotated to the GO terms. )
##
## Building most specific GOs .... ( 1601 GO terms found. )
##
## Build GO DAG topology ..... ( 2164 GO terms and 2776 relations. )
##
## Annotating nodes ..... ( 3948 genes annotated to the GO terms. )
##
## Building most specific GOs .... ( 2171 GO terms found. )
##
## Build GO DAG topology ..... ( 4631 GO terms and 10807 relations. )
##
## Annotating nodes ..... ( 4517 genes annotated to the GO terms. )
##
## Building most specific GOs .... ( 629 GO terms found. )
##
## Build GO DAG topology ..... ( 874 GO terms and 1842 relations. )
##
## Annotating nodes ..... ( 5190 genes annotated to the GO terms. )
##
##          -- Classic Algorithm --
##
##          the algorithm is scoring 822 nontrivial nodes
##          parameters:
##                  test statistic: Fisher test
##
##          -- Classic Algorithm --
##
##          the algorithm is scoring 2504 nontrivial nodes
##          parameters:
##                  test statistic: Fisher test
##
##          -- Classic Algorithm --
##
##          the algorithm is scoring 502 nontrivial nodes
##          parameters:
##                  test statistic: Fisher test
##
##          -- Classic Algorithm --
##
##          the algorithm is scoring 2164 nontrivial nodes
##          parameters:
##                  test statistic: KS tests
##                  score order: increasing
##
##          -- Classic Algorithm --
##
##          the algorithm is scoring 4631 nontrivial nodes
##          parameters:
##                  test statistic: KS tests
##                  score order: increasing
##

```

```

##          -- Classic Algorithm --
##
##      the algorithm is scoring 874 nontrivial nodes
##      parameters:
##          test statistic:  KS tests
##          score order:  increasing
##
##          -- Elim Algorithm --
##
##      the algorithm is scoring 2164 nontrivial nodes
##      parameters:
##          test statistic:  Fisher test
##          cutOff:  0.01
##          score order:  increasing
##
##      Level 15:  3 nodes to be scored      (0 eliminated genes)
##
##      Level 14:  6 nodes to be scored      (0 eliminated genes)
##
##      Level 13:  6 nodes to be scored      (0 eliminated genes)
##
##      Level 12:  19 nodes to be scored     (31 eliminated genes)
##
##      Level 11:  41 nodes to be scored     (31 eliminated genes)
##
##      Level 10:  86 nodes to be scored     (31 eliminated genes)
##
##      Level 9:   195 nodes to be scored    (31 eliminated genes)
##
##      Level 8:   273 nodes to be scored    (576 eliminated genes)
##
##      Level 7:   401 nodes to be scored    (576 eliminated genes)
##
##      Level 6:   591 nodes to be scored    (576 eliminated genes)
##
##      Level 5:   315 nodes to be scored    (728 eliminated genes)
##
##      Level 4:   163 nodes to be scored    (741 eliminated genes)
##
##      Level 3:   51 nodes to be scored     (887 eliminated genes)
##
##      Level 2:   13 nodes to be scored     (2181 eliminated genes)
##
##      Level 1:   1 nodes to be scored      (2408 eliminated genes)
##
##          -- Elim Algorithm --
##
##      the algorithm is scoring 4631 nontrivial nodes
##      parameters:
##          test statistic:  Fisher test
##          cutOff:  0.01
##          score order:  increasing
##
##      Level 18:  3 nodes to be scored      (0 eliminated genes)

```

```

##          Level 17: 8 nodes to be scored      (0 eliminated genes)
##
##          Level 16: 28 nodes to be scored     (0 eliminated genes)
##
##          Level 15: 90 nodes to be scored     (0 eliminated genes)
##
##          Level 14: 156 nodes to be scored    (0 eliminated genes)
##
##          Level 13: 277 nodes to be scored    (0 eliminated genes)
##
##          Level 12: 426 nodes to be scored    (29 eliminated genes)
##
##          Level 11: 534 nodes to be scored    (46 eliminated genes)
##
##          Level 10: 613 nodes to be scored    (188 eliminated genes)
##
##          Level 9:  627 nodes to be scored    (299 eliminated genes)
##
##          Level 8:  545 nodes to be scored    (373 eliminated genes)
##
##          Level 7:  475 nodes to be scored    (574 eliminated genes)
##
##          Level 6:  405 nodes to be scored    (714 eliminated genes)
##
##          Level 5:  255 nodes to be scored    (815 eliminated genes)
##
##          Level 4:  132 nodes to be scored    (1302 eliminated genes)
##
##          Level 3:  40 nodes to be scored     (1355 eliminated genes)
##
##          Level 2:  16 nodes to be scored     (3041 eliminated genes)
##
##          Level 1:  1 nodes to be scored      (3041 eliminated genes)
##
##          -- Elim Algorithm --
##
##          the algorithm is scoring 874 nontrivial nodes
##          parameters:
##              test statistic: Fisher test
##              cutOff: 0.01
##              score order: increasing
##
##          Level 17: 1 nodes to be scored      (0 eliminated genes)
##
##          Level 16: 5 nodes to be scored      (0 eliminated genes)
##
##          Level 15: 32 nodes to be scored     (0 eliminated genes)
##
##          Level 14: 63 nodes to be scored     (0 eliminated genes)
##
##          Level 13: 77 nodes to be scored     (0 eliminated genes)
##
##          Level 12: 82 nodes to be scored     (0 eliminated genes)

```

```

##      Level 11: 123 nodes to be scored (12 eliminated genes)
##
##      Level 10: 124 nodes to be scored (26 eliminated genes)
##
##      Level 9:   59 nodes to be scored (289 eliminated genes)
##
##      Level 8:   90 nodes to be scored (289 eliminated genes)
##
##      Level 7:   53 nodes to be scored (801 eliminated genes)
##
##      Level 6:   56 nodes to be scored (4696 eliminated genes)
##
##      Level 5:   35 nodes to be scored (4696 eliminated genes)
##
##      Level 4:   55 nodes to be scored (4696 eliminated genes)
##
##      Level 3:   11 nodes to be scored (4696 eliminated genes)
##
##      Level 2:    7 nodes to be scored (4714 eliminated genes)
##
##      Level 1:    1 nodes to be scored (4714 eliminated genes)
##
##          -- Weight Algorithm --
##
##      The algorithm is scoring 822 nontrivial nodes
##      parameters:
##          test statistic: Fisher test : ratio
##
##      Level 15: 1 nodes to be scored.
##
##      Level 14: 3 nodes to be scored.
##
##      Level 13: 2 nodes to be scored.
##
##      Level 12: 6 nodes to be scored.
##
##      Level 11: 10 nodes to be scored.
##
##      Level 10: 30 nodes to be scored.
##
##      Level 9:  70 nodes to be scored.
##
##      Level 8:  85 nodes to be scored.
##
##      Level 7: 132 nodes to be scored.
##
##      Level 6: 184 nodes to be scored.
##
##      Level 5: 143 nodes to be scored.
##
##      Level 4: 106 nodes to be scored.
##
##      Level 3: 38 nodes to be scored.

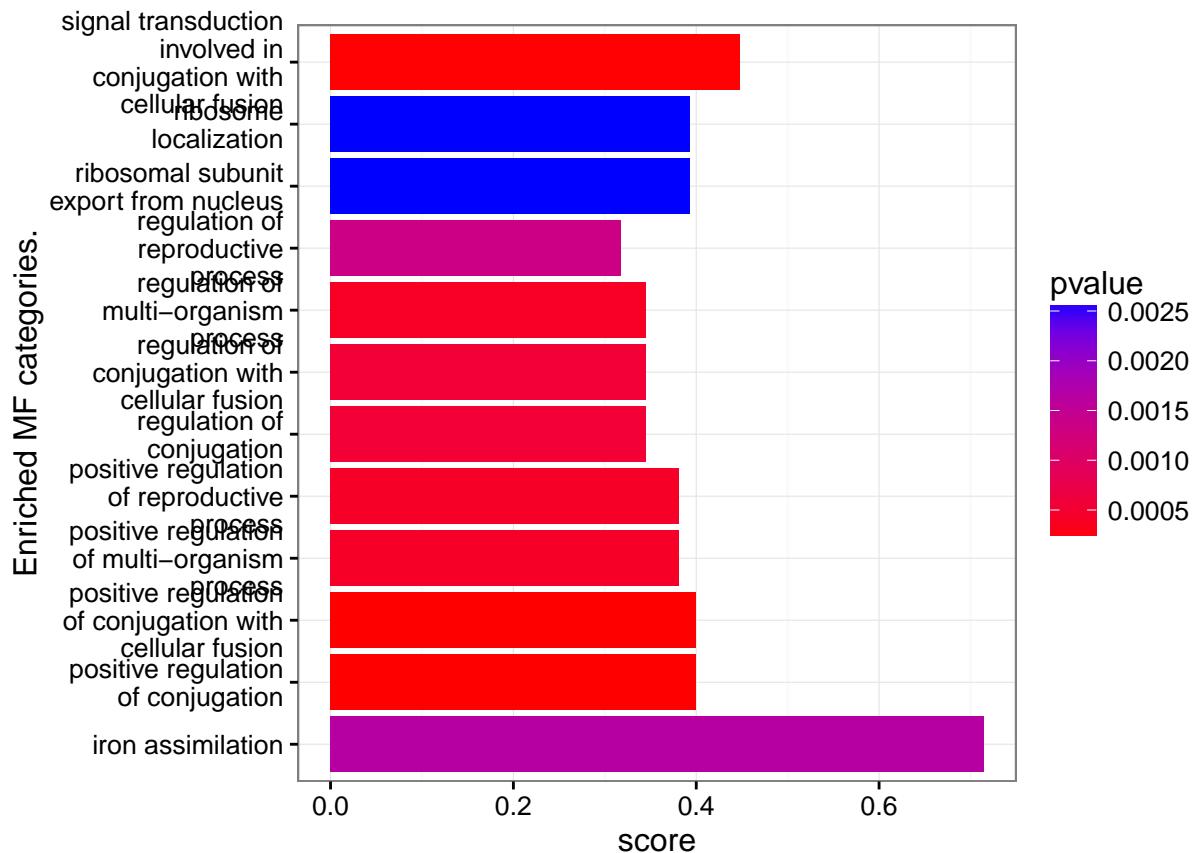
```

```

## 
##   Level 2: 11 nodes to be scored.
## 
##   Level 1: 1 nodes to be scored.
## 
##           -- Weight Algorithm --
## 
##       The algorithm is scoring 2504 nontrivial nodes
##       parameters:
##           test statistic: Fisher test : ratio
## 
##   Level 18: 1 nodes to be scored.
## 
##   Level 17: 3 nodes to be scored.
## 
##   Level 16: 6 nodes to be scored.
## 
##   Level 15: 24 nodes to be scored.
## 
##   Level 14: 51 nodes to be scored.
## 
##   Level 13: 102 nodes to be scored.
## 
##   Level 12: 165 nodes to be scored.
## 
##   Level 11: 234 nodes to be scored.
## 
##   Level 10: 291 nodes to be scored.
## 
##   Level 9: 355 nodes to be scored.
## 
##   Level 8: 326 nodes to be scored.
## 
##   Level 7: 299 nodes to be scored.
## 
##   Level 6: 275 nodes to be scored.
## 
##   Level 5: 204 nodes to be scored.
## 
##   Level 4: 114 nodes to be scored.
## 
##   Level 3: 38 nodes to be scored.
## 
##   Level 2: 15 nodes to be scored.
## 
##   Level 1: 1 nodes to be scored.
## 
##           -- Weight Algorithm --
## 
##       The algorithm is scoring 502 nontrivial nodes
##       parameters:
##           test statistic: Fisher test : ratio
## 
##   Level 16: 2 nodes to be scored.

```

```
##      Level 15: 17 nodes to be scored.  
##  
##      Level 14: 36 nodes to be scored.  
##  
##      Level 13: 43 nodes to be scored.  
##  
##      Level 12: 43 nodes to be scored.  
##  
##      Level 11: 68 nodes to be scored.  
##  
##      Level 10: 68 nodes to be scored.  
##  
##      Level 9: 37 nodes to be scored.  
##  
##      Level 8: 53 nodes to be scored.  
##  
##      Level 7: 31 nodes to be scored.  
##  
##      Level 6: 34 nodes to be scored.  
##  
##      Level 5: 20 nodes to be scored.  
##  
##      Level 4: 33 nodes to be scored.  
##  
##      Level 3: 10 nodes to be scored.  
##  
##      Level 2: 6 nodes to be scored.  
##  
##      Level 1: 1 nodes to be scored.  
  
## simple_topgo(): Set densities=TRUE for ontology densitya plots.  
  
topgo_search$pvalue_plots$BP
```



```

gff_from_txdb <- GenomicFeatures::asGFF(pombe)
## why is GeneID: getting prefixed to the IDs!?
gff_from_txdb$ID <- gsub(x=gff_from_txdb$ID, pattern="GeneID:", replacement="")
written_gff <- rtracklayer::export.gff3(gff_from_txdb, con="pombe.gff")
gostats_search = simple_gostats(updown_genes$up_genes, "pombe.gff", pombe_goids, gff_type="gene")

## Loading required namespace: GOstats

## The namespaces/environments uses by GOstats are entirely too complex.

## If I try to call functions with Category:: or GOstats:: then they collide

## And things fail without error, but if I try library() then R CMD check

## gets pissed, well I tried both ways and I am calling library().

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following object is masked from 'package:IRanges':
##      expand

```

```

## 
## Attaching package: 'Category'

## The following objects are masked from 'package:topGO':
## 
##     ontology<-, testName

## Loading required package: annotate

## Loading required package: XML

## 
## Attaching package: 'XML'

## The following object is masked from 'package:graph':
## 
##    .addNode

## 
## Attaching package: 'GSEABase'

## The following objects are masked from 'package:topGO':
## 
##     ontology, phenotype

## 
## Attaching package: 'GOstats'

## The following object is masked from 'package:AnnotationDbi':
## 
##     makeGOTGraph

## simple_gostats(): gff_type is: gene. Change that if there are bad merges.

## simple_gostats(): type CDS has 0 annotations.

## simple_gostats(): type gene has 7015 annotations.

## simple_gostats(): type exon has 0 annotations.

## simple_gostats(): the current annotations has: 7015 rows and 12 columns.

## simple_gostats(): Creating the gene set collection.

## simple_gostats(): Performing MF GSEA.

## Found 90 over MF categories.

## simple_gostats(): Performing BP GSEA.

```

```

## Found 216 over BP categories.

## simple_gostats(): Performing CC GSEA.

## Found 62 over CC categories.

## simple_gostats(): Performing under MF GSEA.

## Found 56 under MF categories.

## simple_gostats(): Performing under BP GSEA.

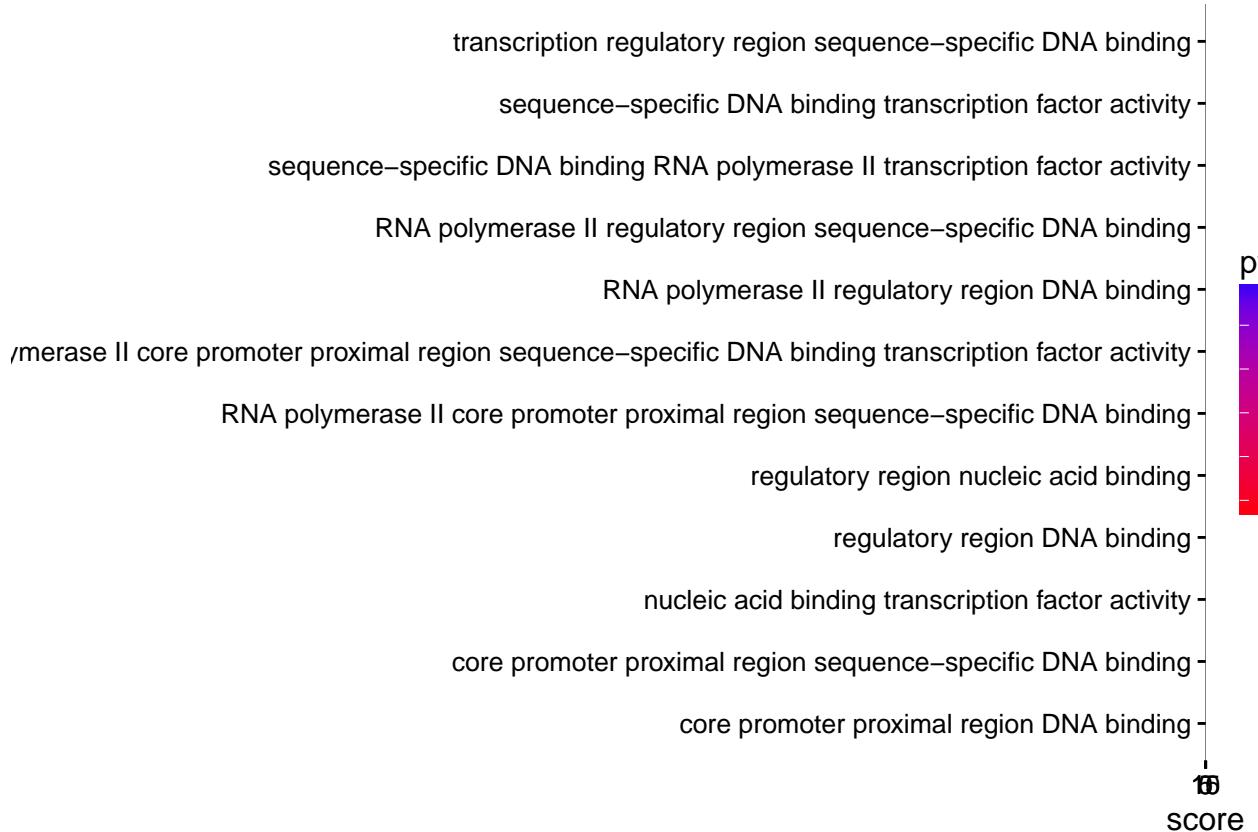
## Found 211 under BP categories.

## simple_gostats(): Performing under CC GSEA.

## Found 69 under CC categories.

## Oops I forgot to add my 25 character wrapper for these plots, whatever
## that is weird I thought I did!
gostats_search$pvalue_plots$mfp_plot_over

```



```
head(gostats_search$mf_over_all)
```

```
##      GOMFID      Pvalue OddsRatio ExpCount Count Size
## 1 GO:0000981 2.655249e-06 2.995569 14.86398   33   93
## 2 GO:0001067 3.483858e-06 2.945556 15.02381   33   94
## 3 GO:0000987 3.588363e-06 3.001785 14.38450   32   90
## 4 GO:0000978 3.588363e-06 3.001785 14.38450   32   90
## 5 GO:0001159 3.588363e-06 3.001785 14.38450   32   90
## 6 GO:0000982 4.712668e-06 2.950001 14.54433   32   91
##
## 1 <a href="http://www.godatabase.org/cgi-bin/amigo/go.cgi?view=details&search_constraint=terms&depth=1"
## 2 <a href="http://www.godatabase.org/cgi-bin/amigo/go.cgi?view=details&search_constraint=terms&depth=1"
## 3 <a href="http://www.godatabase.org/cgi-bin/amigo/go.cgi?view=details&search_constraint=terms&depth=1"
## 4 <a href="http://www.godatabase.org/cgi-bin/amigo/go.cgi?view=details&search_constraint=terms&depth=1"
## 5 <a href="http://www.godatabase.org/cgi-bin/amigo/go.cgi?view=details&search_constraint=terms&depth=1"
## 6 <a href="http://www.godatabase.org/cgi-bin/amigo/go.cgi?view=details&search_constraint=terms&depth=1"
##      qvalue
## 1 0.00050447
## 2 0.00050447
## 3 0.00050447
## 4 0.00050447
## 5 0.00050447
## 6 0.00050447
```