

Human DGEs of SACC-PHHs mono-infected with HBV or coinfecte with HBV/HDV factors donor time

Purpose:

To determine the DGE profiles (for human genes), relative to uninfected controls, of self-assembling co-cultures of primary human hepatocytes (SACC-PHHs) (co-cultured with 3T3J mouse non-parenchymal cells) mono-infected with HBV or co-infected with HBV/HDV at 8 and 28 days post-infection. Here, donor and time are factors in the design.

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(stringr)
library(ggplot2)
library(reshape2)
library(openxlsx)
library(DESeq2)

## Loading required package: S4Vectors
## Loading required package: stats4
## 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 objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, cbind, colnames,
```

```

##      do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##      grepl, intersect, is.unsorted, lapply, lengths, Map, mapply,
##      match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##      Position, rank, rbind, Reduce, rownames, sapply, setdiff,
##      sort, table, tapply, union, unique, unsplit

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##      first, rename

## The following objects are masked from 'package:base':
##      colMeans, colSums, expand.grid, rowMeans, rowSums

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##      collapse, desc, slice

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## 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")'.

library(gplots)

##
## Attaching package: 'gplots'

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

## The following object is masked from 'package:S4Vectors':
##      space

## The following object is masked from 'package:stats':
##      lowess

library(dplyr)
library(RColorBrewer)
library(stringr)
library(genefilter)
library(data.table)

```

```

## 
## Attaching package: 'data.table'
## The following object is masked from 'package:SummarizedExperiment':
##   shift
## The following object is masked from 'package:GenomicRanges':
##   shift
## The following object is masked from 'package:IRanges':
##   shift
## The following objects are masked from 'package:S4Vectors':
##   first, second
## The following objects are masked from 'package:reshape2':
##   dcast, melt
## The following objects are masked from 'package:dplyr':
##   between, first, last
library(genefilter)
library(ggrepel)
library(viridis)

## Loading required package: viridisLite
source("http://bioconductor.org/biocLite.R")

## Bioconductor version 3.3 (BiocInstaller 1.22.3), ?biocLite for help
## A newer version of Bioconductor is available for this version of R,
##   ?BiocUpgrade for help
biocLite("org.Hs.eg.db", suppressUpdates = TRUE)

## BioC_mirror: https://bioconductor.org
## Using Bioconductor 3.3 (BiocInstaller 1.22.3), R 3.3.3 (2017-03-06).
## Installing package(s) 'org.Hs.eg.db'
## installing the source package 'org.Hs.eg.db'
require(org.Hs.eg.db)

## Loading required package: org.Hs.eg.db
## Loading required package: AnnotationDbi
##
## Attaching package: 'AnnotationDbi'
## The following object is masked from 'package:dplyr':
##   select
##

```

Pulling in the human counts files

```
##Pulling in the counts of only human genes as determined previously in my
##DGE_sorting analysis.
humancounts <- "All human HBV genes"
human_sampleCounts <- basename(Sys.glob(file.path(humancounts, "*.txt")))

##Function to read in the feature counts
exptcounts <- function(files) {
  d <- read.table(files)
  d
}

##Read in all of the count files
humancounts_readin <- lapply(file.path(humancounts, human_sampleCounts), exptcounts)
names(humancounts_readin) <- sub('humanHBVgenes', ' ', human_sampleCounts)
names(humancounts_readin)

## [1] "BD330_Ctrl_D28.txt"      "BD330_Ctrl_D8.txt"
## [3] "BD330_HBV_D28.txt"      "BD330_HBV_D8.txt"
## [5] "BD330_HBV_HDV_D28_b.txt" "BD330_HBV_HDV_D28.txt"
## [7] "BD330_HBV_HDV_D8_a.txt"  "BD330_HBV_HDV_D8.txt"
## [9] "BD405A_Ctrl_D28.txt"     "BD405A_Ctrl_D8.txt"
## [11] "BD405A_HBV_D28.txt"     "BD405A_HBV_D8.txt"
## [13] "BD405A_HBV_HDV_D28.txt" "BD405A_HBV_HDV_D8.txt"
## [15] "Ctrl_D28_sample_1.txt"   "Ctrl_D28_sample_2.txt"
## [17] "Ctrl_D28_sample_3.txt"   "Ctrl_D8_sample_1.txt"
## [19] "Ctrl_D8_sample_2.txt"    "Ctrl_D8_sample_3.txt"
## [21] "HBV_D28_sample_1.txt"   "HBV_D28_sample_2.txt"
## [23] "HBV_D28_sample_3.txt"   "HBV_D8_sample_1.txt"
## [25] "HBV_D8_sample_2.txt"    "HBV_D8_sample_3.txt"
## [27] "HU1016_BD_co_D28.txt"   "HU1016_BD_co_D8.txt"
## [29] "HU1016_B_D28.txt"       "HU1016_B_D8.txt"
```

Now subset feature counts by “treatment” for DGE analysis.

```
ctrl <- names(humancounts_readin)[grep("*Ctrl", names(humancounts_readin))]
ctrl_counts <- humancounts_readin[match(ctrl, names(humancounts_readin))]
names(ctrl_counts)

## [1] "BD330_Ctrl_D28.txt"      "BD330_Ctrl_D8.txt"
## [3] "BD405A_Ctrl_D28.txt"     "BD405A_Ctrl_D8.txt"
## [5] "Ctrl_D28_sample_1.txt"   "Ctrl_D28_sample_2.txt"
## [7] "Ctrl_D28_sample_3.txt"   "Ctrl_D8_sample_1.txt"
## [9] "Ctrl_D8_sample_2.txt"    "Ctrl_D8_sample_3.txt"

HBV <- names(humancounts_readin)[grep("*HBV_D|_B_", names(humancounts_readin))]
HBV_counts <- humancounts_readin[match(HBV, names(humancounts_readin))]
names(HBV_counts)

## [1] "BD330_HBV_D28.txt"      "BD330_HBV_D8.txt"      "BD405A_HBV_D28.txt"
## [4] "BD405A_HBV_D8.txt"      "HBV_D28_sample_1.txt"  "HBV_D28_sample_2.txt"
## [7] "HBV_D28_sample_3.txt"   "HBV_D8_sample_1.txt"   "HBV_D8_sample_2.txt"
## [10] "HBV_D8_sample_3.txt"    "HU1016_B_D28.txt"     "HU1016_B_D8.txt"

coinf <- names(humancounts_readin)[grep("*HBV_HDV_|_co_", names(humancounts_readin))]
coinf_counts <- humancounts_readin[match(coinf, names(humancounts_readin))]
```

```

names(coinf_counts)

## [1] "BD330_HBV_HDV_D28_b.txt" "BD330_HBV_HDV_D28.txt"
## [3] "BD330_HBV_HDV_D8_a.txt"   "BD330_HBV_HDV_D8.txt"
## [5] "BD405A_HBV_HDV_D28.txt"   "BD405A_HBV_HDV_D8.txt"
## [7] "HU1016_BD_co_D28.txt"     "HU1016_BD_co_D8.txt"

Make files of these separated feature counts

for(i in names(ctrl_counts)) {
  filename <- paste(i, sep = "")
  write.table(ctrl_counts[i], file = file.path("HumanHBV_d8d28_ctrl", filename),
              col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(HBV_counts)) {
  filename <- paste(i, sep = "")
  write.table(HBV_counts[i], file = file.path("HumanHBV_d8d28_HBV", filename),
              col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(coinf_counts)) {
  filename <- paste(i, sep = "")
  write.table(coinf_counts[i], file = file.path("HumanHBV_d8d28_cointf", filename),
              col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

```

Function to perform DGE analysis with both donor and time set as factors influencing the counts. Since we already sorted out counts into folders containing the ENSEMBL IDs for human genes under different infection conditions, we will pull the files from these folders to perform the DGE analysis.

```

DGE_analysis <- function(sampledirectory) {
  a <- basename(Sys.glob(file.path(sampledirectory, "*.txt")))
  sample_names <- sub('.txt', '', a)
  ##Here the donors are renamed based off the Hurel names (i.e. HU___) - RNASeq reads were all named
  ##using a different ID system.
  sampleTable <- data.frame(sampleName = sample_names, sampleFile = a, treatment =
    ifelse(grepl("Ctrl", a), "mock", ifelse(grepl("*co|*HDV", a), "coinf", "HBV")), donor =
      ifelse(grepl("BD330*", a), "HU1019", ifelse(grepl("BD405*", a), "HU1020",
      ifelse(grepl("HU1016*", a), "HU1016", "HU1007"))), time = ifelse(grepl("*D8", a), "d8",
      "d28"), replicate = ifelse(grepl("*sample_1|*D8_a", a), "a",
      ifelse(grepl("*sample_2|D28_b", a), "b",
      ifelse(grepl("*sample_3", a), "c", ""))))
  dds <- DESeqDataSetFromHTSeqCount(sampleTable = sampleTable, directory = sampledirectory, design =
    ~donor + time)
  dds
  dds@colData
  contrast <- c("time", levels(sampleTable$time))
  output_basename <- sprintf("%s-%s_vs_%s_%s_analysis", "humangenes", contrast[2], contrast[3],
    levels(sampleTable$treatment))
  dds <- estimateSizeFactors(dds)
  dds@colData
  dds <- estimateDispersions(dds)

  plotDispEsts(dds, main=sprintf("%s Dispersion Estimates", output_basename))

```

```

dds <- nbinomWaldTest(dds)
res <- results(dds, contrast=contrast)
res <- res[order(res$padj, -abs(res$log2FoldChange)),]
mcols(res, use.names=TRUE)
##Log-intensity ratios = M values, log-intensity averages = A values
##Red points indicate padj < 0.1.
plotMA(res, alpha=0.1, main=sprintf(output_basename))
attr(res, "filterThreshold")

metadata(res)$alpha
metadata(res)$filterThreshold
plot(metadata(res)$filterNumRej,
  type="b", ylab="number of rejections",
  xlab="quantiles of filter")
lines(metadata(res)$lo.fit, col="red")
abline(v=metadata(res)$filterTheta)

key = "ENSEMBL"
cols = c("ENTREZID", "SYMBOL", "GENENAME", "ALIAS", "REFSEQ", "ACCCNUM")
for (col in cols) {
  # Get annotation data for column
  annotation_data <- AnnotationDbi::select(org.Hs.eg.db, rownames(res), col, keytype=key)
  # Collapse one-to-many relationships
  tmp <- aggregate(annotation_data[col], by=annotation_data[key],
    # to a list
    FUN=function(x)list(x))
  # Match on key and append to results
  idx <- match(rownames(res), tmp[[key]])
  res[[col]] <- tmp[idx,col]
}

output_data <- as.data.frame(res)
LIST_COLS <- sapply(output_data, is.list)
for (COL in colnames(output_data)[LIST_COLS]) {
  output_data[COL] <-
    sapply(output_data[COL],
      function(x)sapply(x, function(y) paste(unlist(y),
        collapse=", ")))
}

# Save data frame above as tab-separated file
write.table(output_data,
  file=file.path("Human_DGEs_donortime", paste(Sys.Date(),
    "human_donor_time",
    output_basename, "_results.txt", sep='')), quote=FALSE, sep="\t",
  row.names=TRUE, col.names=NA)
return(list(dds@colData, head(res)))
}

##For each infection group, determine the DGE profile when comparing
##the different times to one another (i.e. d8 versus d28).

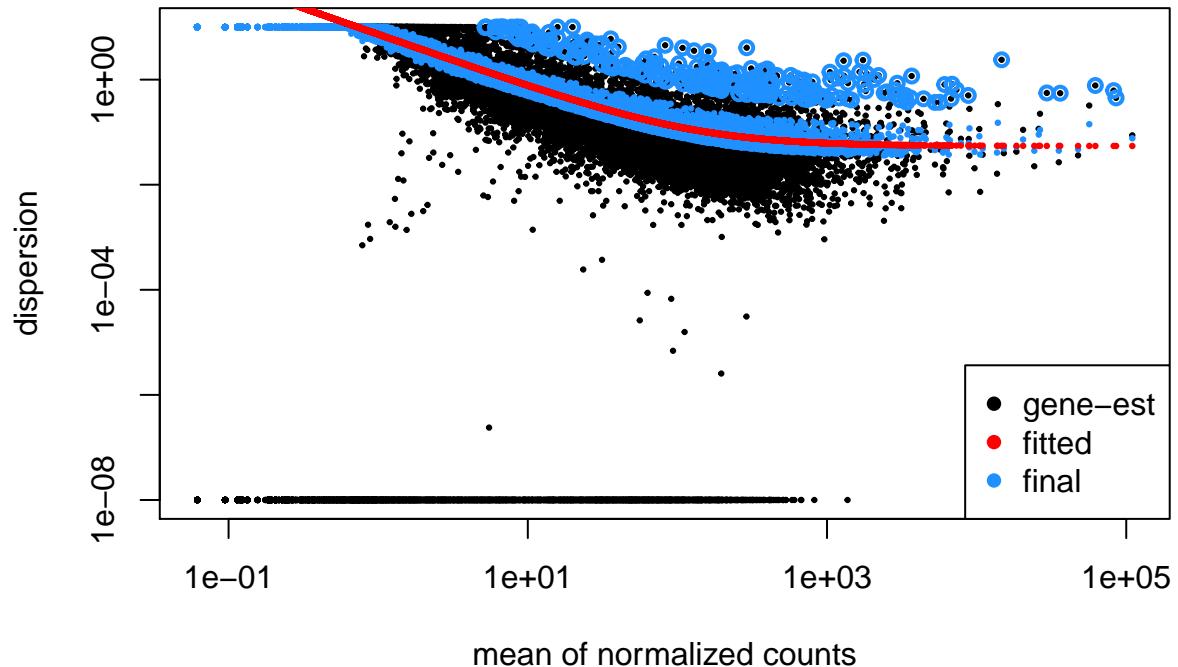
##coinfected

```

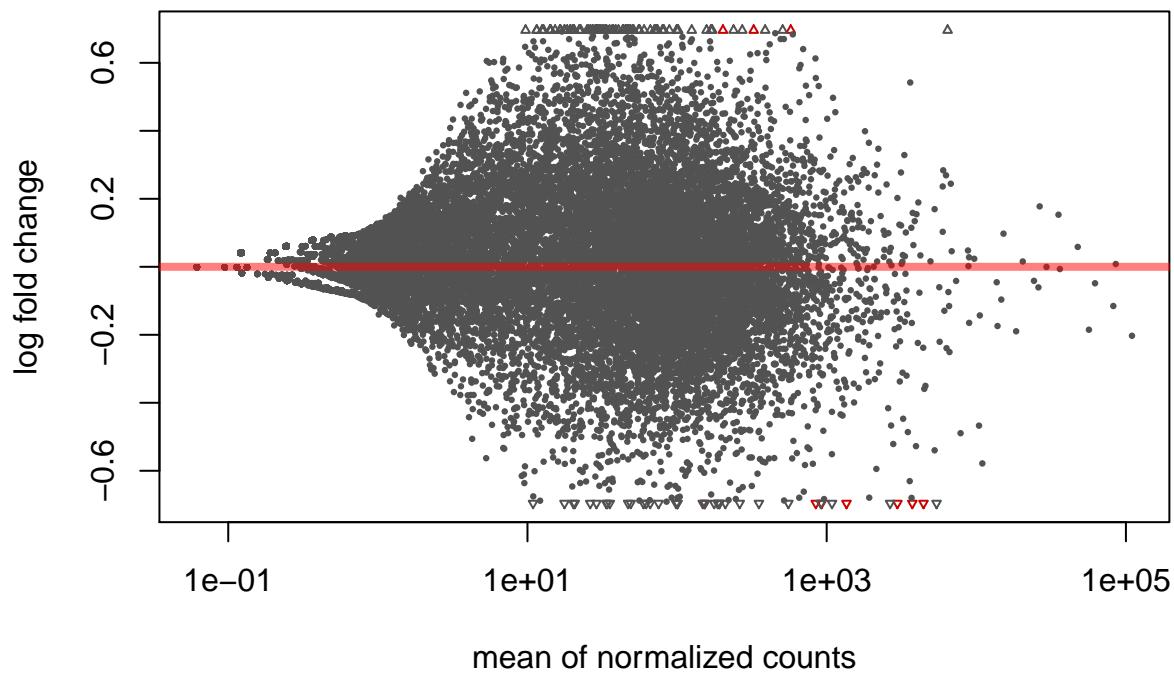
```
DGE_analysis("HumanHBV_d8d28_coinf")
```

```
## gene-wise dispersion estimates  
## mean-dispersion relationship  
## final dispersion estimates
```

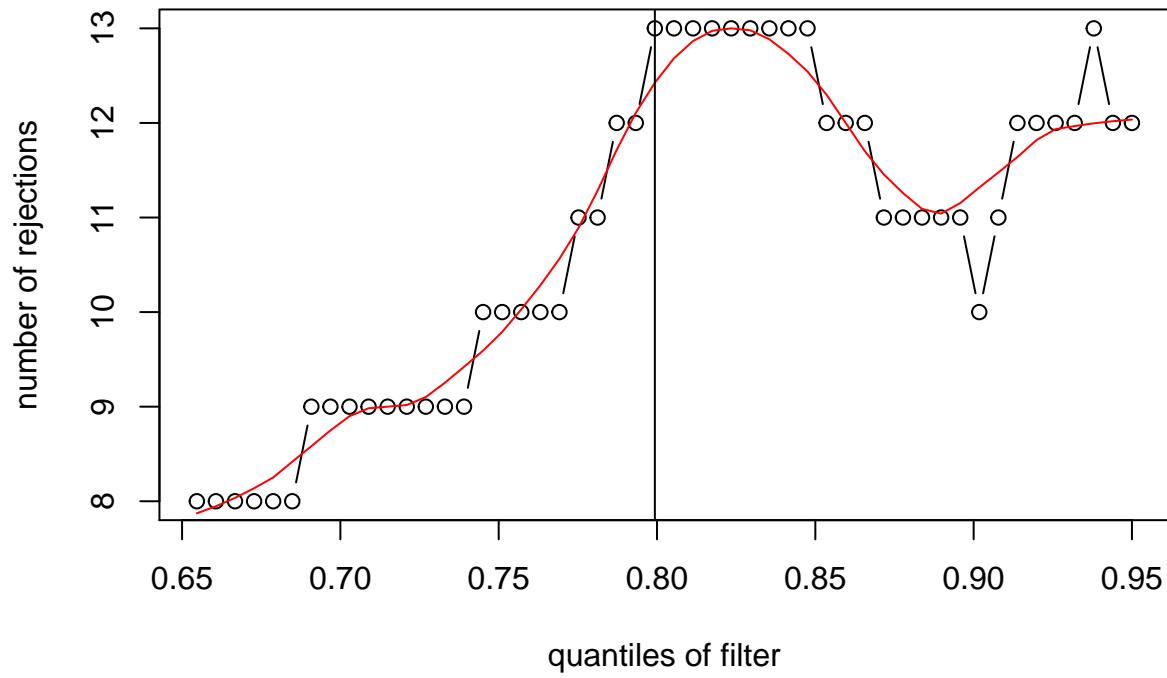
humangenes-d28_vs_d8_coinf_analysis Dispersion Estimates



humangenes-d28_vs_d8_coinf_analysis



```
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
```



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

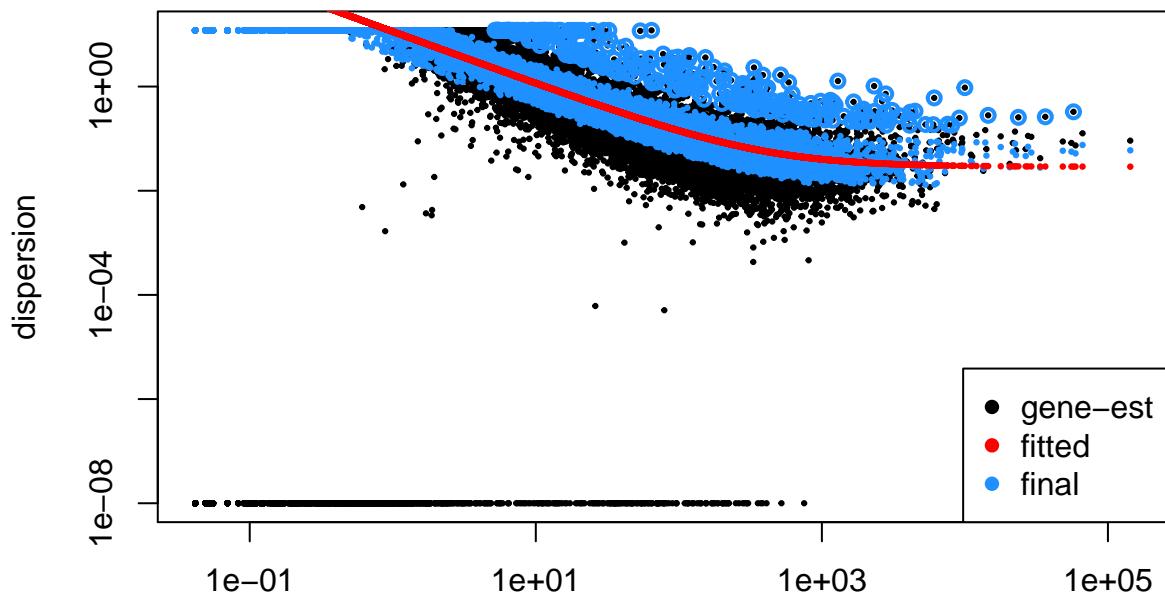
```

## DataFrame with 8 rows and 5 columns
##           treatment   donor    time replicate sizeFactor
##           <factor> <factor> <factor> <factor> <numeric>
## BD330_HBV_HDV_D28     coinf  HU1019    d28      0.6561771
## BD330_HBV_HDV_D28_b    coinf  HU1019    d28      b 1.0388943
## BD330_HBV_HDV_D8      coinf  HU1019    d8       1.3189720
## BD330_HBV_HDV_D8_a    coinf  HU1019    d8      a 2.0263762
## BD405A_HBV_HDV_D28    coinf  HU1020    d28      0.6083868
## BD405A_HBV_HDV_D8      coinf  HU1020    d8       0.9363511
## HU1016_BD_co_D28     coinf  HU1016    d28      1.0162846
## HU1016_BD_co_D8      coinf  HU1016    d8       1.0935164
##
## [[2]]
## log2 fold change (MAP): time d28 vs d8
## Wald test p-value: time d28 vs d8
## DataFrame with 6 rows and 12 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric> <numeric> <numeric> <numeric> <numeric>
## ENSG00000117983 28.96142      1.791582 0.3035690 5.901729 3.597125e-09
## ENSG00000115602 72.84319      1.642225 0.2982479 5.506244 3.665712e-08
## ENSG00000165124 575.26843      1.188672 0.2281984 5.208940 1.899223e-07
## ENSG00000160862 923.93567      -1.027754 0.1981009 -5.188031 2.125287e-07
## ENSG00000168906 325.23649      1.135355 0.2497809 4.545402 5.483043e-06
## ENSG00000145192 4436.91763      -1.088028 0.2410504 -4.513693 6.370830e-06
##           padj ENTREZID SYMBOL GENENAME ALIAS REFSEQ
##           <numeric> <list> <list> <list> <list> <list>
## ENSG00000117983 4.208996e-05 ##### ##### ##### #####
## ENSG00000115602 2.144625e-04 ##### ##### ##### #####
## ENSG00000165124 6.216995e-04 ##### ##### ##### #####
## ENSG00000160862 6.216995e-04 ##### ##### ##### #####
## ENSG00000168906 1.238827e-02 ##### ##### ##### #####
## ENSG00000145192 1.238827e-02 ##### ##### ##### #####
##           ACCNUM
##           <list>
## ENSG00000117983 #####
## ENSG00000115602 #####
## ENSG00000165124 #####
## ENSG00000160862 #####
## ENSG00000168906 #####
## ENSG00000145192 #####
##monoinfected with HBV
DGE_analysis("HumanHBV_d8d28_HBV")

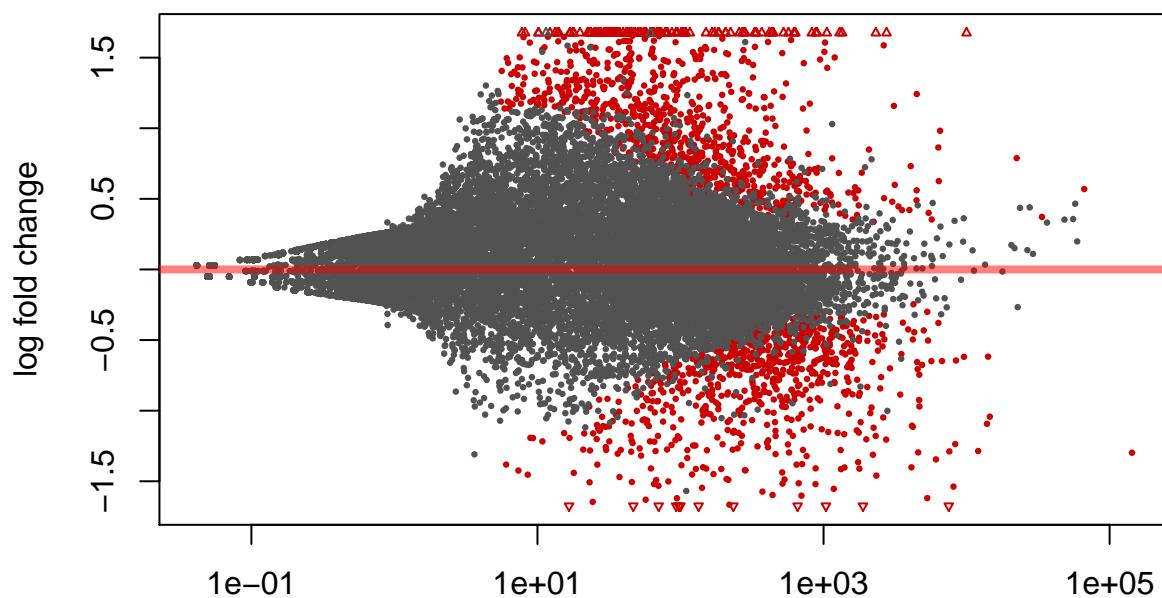
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates

```

humangenes-d28_vs_d8_HBV_analysis Dispersion Estimates



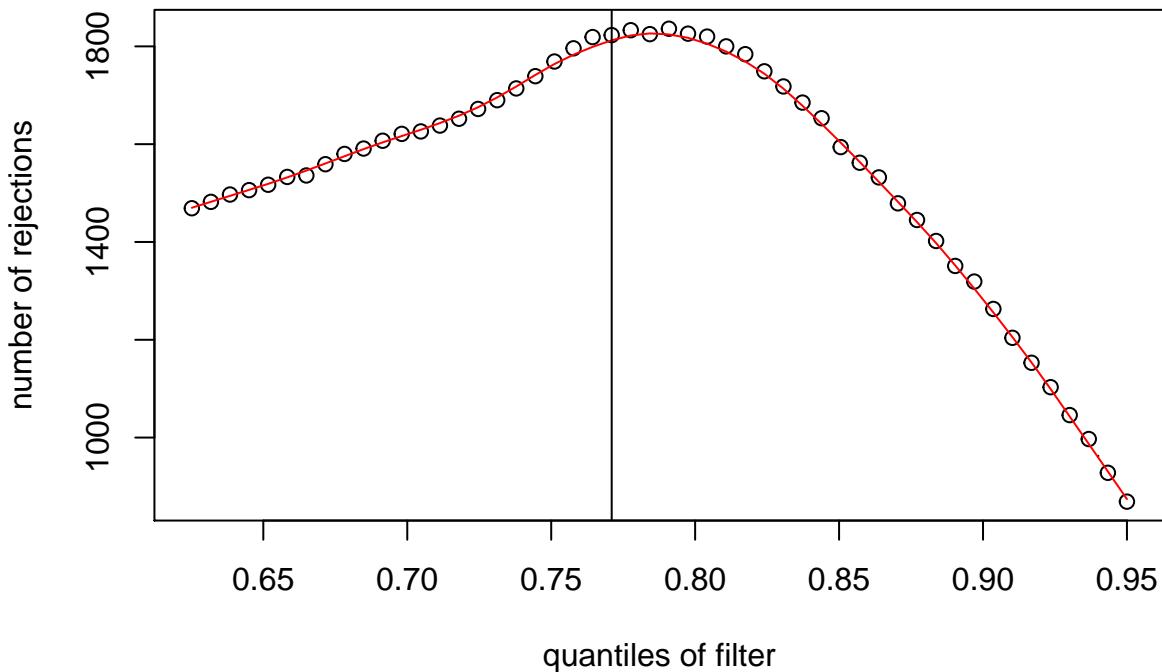
mean of normalized counts humangenes-d28_vs_d8_HBV_analysis



mean of normalized counts

```
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
```

```
## 'select()' returned 1:many mapping between keys and columns
```



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

```
## DataFrame with 12 rows and 5 columns
##   treatment donor time replicate sizeFactor
##   <factor> <factor> <factor> <factor> <numeric>
## BD330_HBV_D28      HBV  HU1019    d28     1.6059297
## BD330_HBV_D8       HBV  HU1019    d8      0.7914293
## BD405A_HBV_D28     HBV  HU1020    d28     0.3450091
## BD405A_HBV_D8       HBV  HU1020    d8      0.5533128
## HBV_D28_sample_1    HBV  HU1007    d28     a      1.5051295
## ...
##   ...     ...     ...     ...     ...
## HBV_D8_sample_1     HBV  HU1007    d8      a      1.6927444
## HBV_D8_sample_2     HBV  HU1007    d8      b      1.1916495
## HBV_D8_sample_3     HBV  HU1007    d8      c      1.6385834
## HU1016_B_D28       HBV  HU1016    d28     0.5486196
## HU1016_B_D8        HBV  HU1016    d8      0.5782392
##
## [[2]]
## log2 fold change (MAP): time d28 vs d8
## Wald test p-value: time d28 vs d8
## DataFrame with 6 rows and 12 columns
##   baseMean log2FoldChange    lfcSE      stat     pvalue
##   <numeric> <numeric> <numeric> <numeric> <numeric>
## ENSG00000087245  55.16884   3.981201  0.4028361  9.882930  4.936774e-23
## ENSG00000134871  900.68453  2.270930  0.2392537  9.491720  2.272525e-21
## ENSG00000145192  7504.92966 -1.736861  0.1981389 -8.765876  1.853299e-18
## ENSG00000171234  1178.37675 -1.553767  0.1773259 -8.762211  1.914573e-18
## ENSG00000142798  1349.82328  2.187439  0.2579493  8.480114  2.249726e-17
## ENSG00000109181  658.62636 -2.237485  0.2710106 -8.256078  1.505373e-16
##   padj ENTREZID SYMBOL GENENAME ALIAS REFSEQ
##   <numeric> <list> <list> <list> <list> <list>
## ENSG00000087245 6.569365e-19 ##### ##### ##### ##### ##### #####
```

```

## ENSG00000134871 1.512024e-17 #####
## ENSG00000145192 6.369305e-15 #####
## ENSG00000171234 6.369305e-15 #####
## ENSG00000142798 5.987422e-14 #####
## ENSG00000109181 3.338667e-13 #####
##                                     ACCNUM
##                                     <list>
## ENSG00000087245 #####
## ENSG00000134871 #####
## ENSG00000145192 #####
## ENSG00000171234 #####
## ENSG00000142798 #####
## ENSG00000109181 #####
##ctrl
DGE_analysis("HumanHBV_d8d28_ctrl")

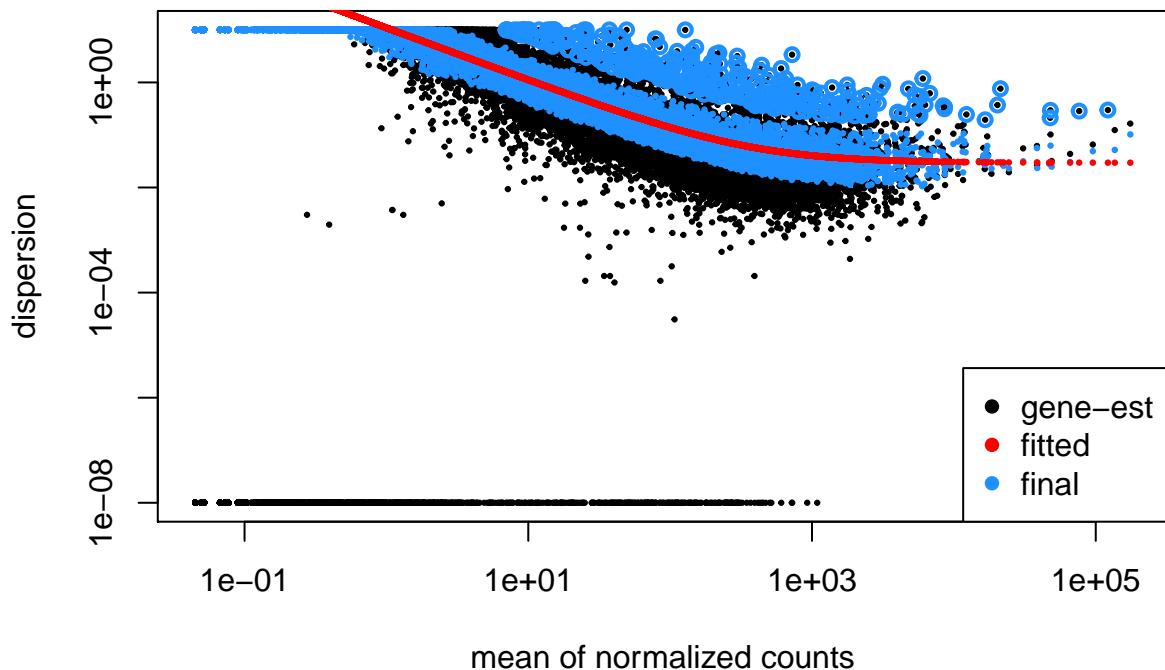
```

```

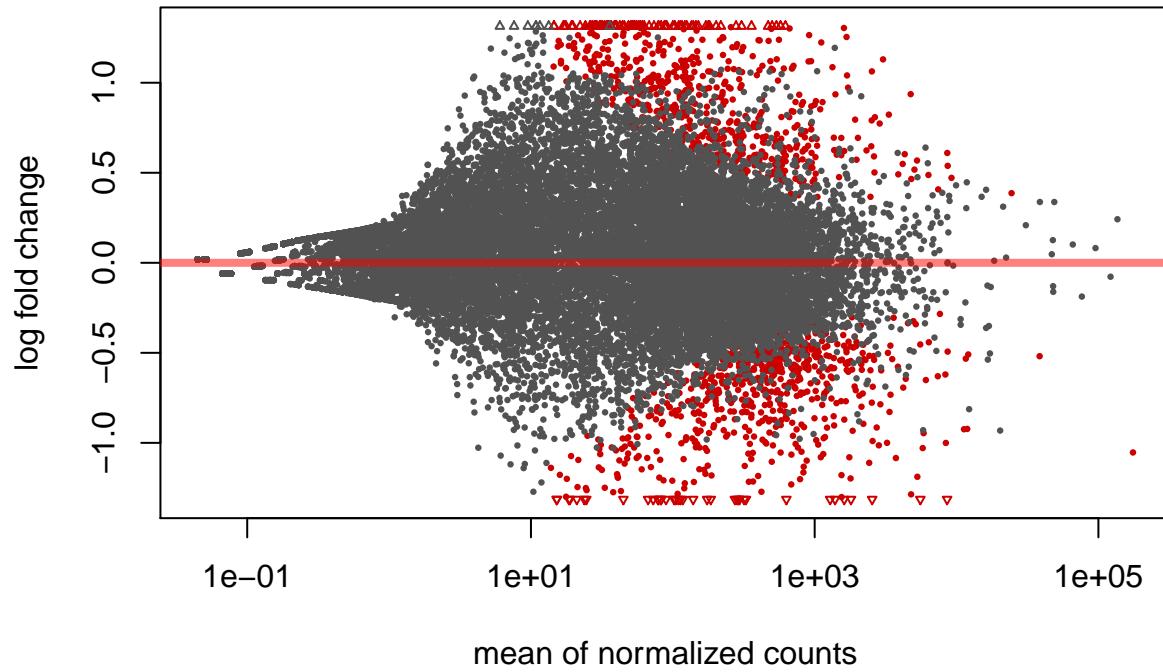
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates

```

humangenes-d28_vs_d8_mock_analysis Dispersion Estimates

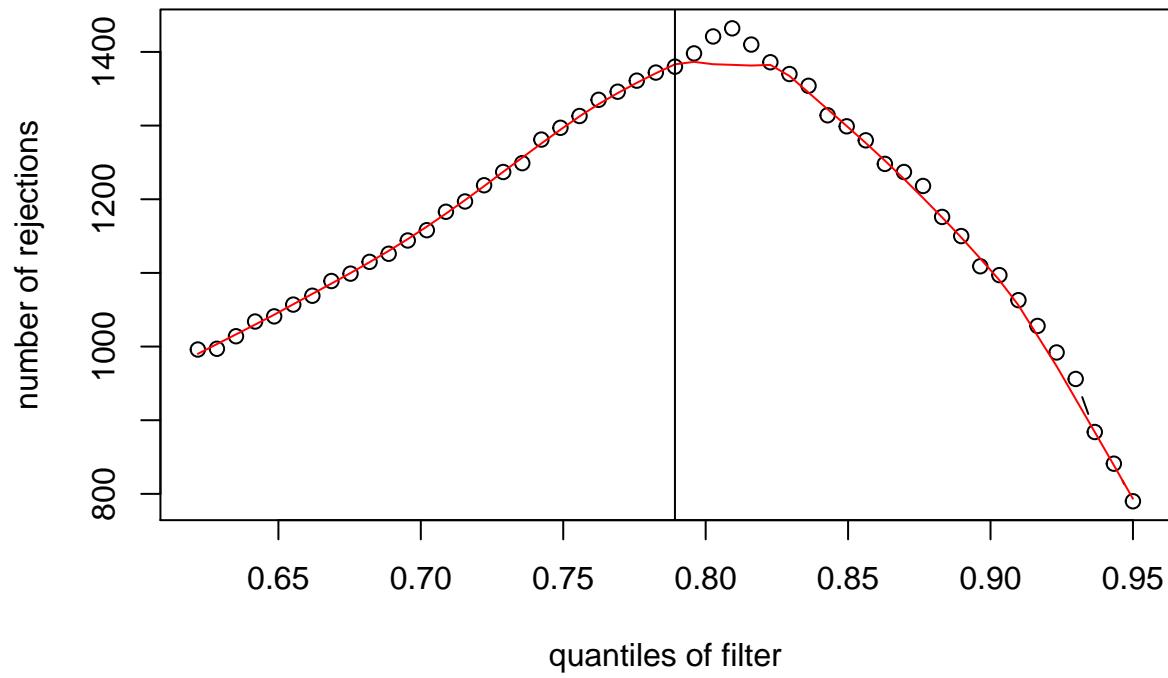


humangenes-d28_vs_d8_mock_analysis



mean of normalized counts

```
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns
```



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

```

## DataFrame with 10 rows and 5 columns
##           treatment   donor    time replicate sizeFactor
##           <factor> <factor> <factor> <factor> <numeric>
## BD330_Ctrl_D28      mock  HU1019    d28     0.6029363
## BD330_Ctrl_D8       mock  HU1019     d8      0.6097982
## BD405A_Ctrl_D28     mock  HU1020    d28     0.3536643
## BD405A_Ctrl_D8     mock  HU1020     d8      0.3953867
## Ctrl_D28_sample_1   mock  HU1007    d28      a  1.9310556
## Ctrl_D28_sample_2   mock  HU1007    d28      b  2.0160145
## Ctrl_D28_sample_3   mock  HU1007    d28      c  2.2288123
## Ctrl_D8_sample_1    mock  HU1007     d8      a  1.3770683
## Ctrl_D8_sample_2    mock  HU1007     d8      b  1.2997594
## Ctrl_D8_sample_3    mock  HU1007     d8      c  1.4893190
##
## [[2]]
## log2 fold change (MAP): time d28 vs d8
## Wald test p-value: time d28 vs d8
## DataFrame with 6 rows and 12 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>     <numeric> <numeric> <numeric>     <numeric>
## ENSG00000115602  177.8098      1.743402  0.2489768  7.002267 2.518538e-12
## ENSG00000115380  500.9996      2.147284  0.3196812  6.716955 1.855608e-11
## ENSG00000166592  116.1884      1.652413  0.2522396  6.550963 5.716718e-11
## ENSG00000160862 1285.5345     -1.543401  0.2483018 -6.215826 5.105529e-10
## ENSG00000168906  559.9614      1.238758  0.2009034  6.165937 7.006710e-10
## ENSG00000249948  139.0729     -2.086698  0.3437171 -6.070976 1.271349e-09
##           padj ENTREZID      SYMBOL GENENAME      ALIAS      REFSEQ
##           <numeric>     <list>   <list>   <list>   <list>   <list>
## ENSG00000115602 3.091757e-08 ##### ##### ##### ##### #####
## ENSG00000115380 1.138972e-07 ##### ##### ##### ##### #####
## ENSG00000166592 2.339281e-07 ##### ##### ##### ##### #####
## ENSG00000160862 1.566887e-06 ##### ##### ##### ##### #####
## ENSG00000168906 1.720287e-06 ##### ##### ##### ##### #####
## ENSG00000249948 2.229582e-06 ##### ##### ##### ##### #####
##           ACCNUM
##           <list>
## ENSG00000115602 #####
## ENSG00000115380 #####
## ENSG00000166592 #####
## ENSG00000160862 #####
## ENSG00000168906 #####
## ENSG00000249948 #####

```

Session Info

```

sessionInfo()

## R version 3.3.3 (2017-03-06)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:

```

```

## [1] parallel stats4 stats graphics grDevices utils datasets
## [8] methods base
##
## other attached packages:
## [1] org.Hs.eg.db_3.3.0 AnnotationDbi_1.34.4
## [3] BiocInstaller_1.22.3 viridis_0.4.0
## [5] viridisLite_0.2.0 ggrepel_0.6.5
## [7] data.table_1.10.0 genefilter_1.54.2
## [9] RColorBrewer_1.1-2 gplots_3.0.1
## [11] DESeq2_1.12.4 SummarizedExperiment_1.2.3
## [13] Biobase_2.32.0 GenomicRanges_1.24.3
## [15] GenomeInfoDb_1.8.7 IRanges_2.6.1
## [17] S4Vectors_0.10.3 BiocGenerics_0.18.0
## [19] openxlsx_4.0.17 reshape2_1.4.2
## [21] ggplot2_2.2.1 stringr_1.2.0
## [23] dplyr_0.7.3
##
## loaded via a namespace (and not attached):
## [1] splines_3.3.3 gtools_3.5.0 Formula_1.2-1
## [4] assertthat_0.2.0 latticeExtra_0.6-28 yaml_2.1.14
## [7] RSQLite_1.1-2 backports_1.0.5 lattice_0.20-35
## [10] glue_1.1.1 digest_0.6.12 XVector_0.12.1
## [13] checkmate_1.8.2 colorspace_1.3-2 htmltools_0.3.5
## [16] Matrix_1.2-8 plyr_1.8.4 XML_3.98-1.9
## [19] pkgconfig_2.0.1 zlibbioc_1.18.0 xtable_1.8-2
## [22] scales_0.4.1 gdata_2.17.0 BiocParallel_1.6.6
## [25] htmlTable_1.9 tibble_1.3.3 annotate_1.50.1
## [28] nnet_7.3-12 lazyeval_0.2.0 survival_2.41-3
## [31] magrittr_1.5 memoise_1.0.0 evaluate_0.10
## [34] foreign_0.8-67 tools_3.3.3 munsell_0.4.3
## [37] locfit_1.5-9.1 cluster_2.0.6 bindrcpp_0.2
## [40] caTools_1.17.1 rlang_0.1.2 grid_3.3.3
## [43] RCurl_1.95-4.8 htmlwidgets_0.9 bitops_1.0-6
## [46] base64enc_0.1-3 rmarkdown_1.4 gtable_0.2.0
## [49] DBI_0.6-1 R6_2.2.0 gridExtra_2.2.1
## [52] knitr_1.16 bindr_0.1 Hmisc_4.0-2
## [55] rprojroot_1.2 KernSmooth_2.23-15 stringi_1.1.5
## [58] Rcpp_0.12.10 geneplotter_1.50.0 rpart_4.1-10
## [61] acepack_1.4.1

```