HBV gene expression_including July 2018 samples

Purpose:

To make a heatmap of the normalized counts for the HBV genes across samples.

Load required libraries

```
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(gplots)
```

```
##
## Attaching package: 'gplots'
```

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

```
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, which, which.max,
##     which.min

##
## Attaching package: 'S4Vectors'

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

## 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(dplyr)
library(tibble)
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(RColorBrewer)
data("egSymb")
```

```
## Warning in data("egSymb"): data set 'egSymb' not found
```

Read in the appropriate count files

```
humanHBVcounts <- "All human HBV genes"
humanHBV_sampleCounts <- basename(Sys.glob(file.path(humanHBVcounts, "*.txt")))

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

##Read in all of the count files
humanHBVcounts_readin <- lapply(file.path(humanHBVcounts, humanHBV_sampleCounts),
                            exptcounts)
names(humanHBVcounts_readin) <- sub('humanHBVgenes.txt', '', humanHBV_sampleCounts)
names(humanHBVcounts_readin)
```

```
##  [1] "BD330 HBV_HDV Day 28 sample 1" "BD330 HBV_HDV Day 28 sample 2"
##  [3] "BD330 HBV_HDV Day 28 sample 3" "BD330 HBV_HDV Day 8 sample 1"
##  [5] "BD330 HBV_HDV Day 8 sample 2"  "BD330 HBV_HDV Day 8 sample 3"
##  [7] "BD330_Ctrl_D28"                "BD330_Ctrl_D8"
##  [9] "BD330_HBV_D28"                 "BD330_HBV_D8"
## [11] "BD330_HBV_HDV_D28_b"           "BD330_HBV_HDV_D28"
## [13] "BD330_HBV_HDV_D8_a"            "BD330_HBV_HDV_D8"
## [15] "BD405A HBV_HDV D28 sample 1"   "BD405A HBV_HDV D28 sample 2"
## [17] "BD405A HBV_HDV D28 sample 3"   "BD405A HBV_HDV D8 sample 1"
## [19] "BD405A HBV_HDV D8 sample 2"    "BD405A HBV_HDV D8 sample 3"
## [21] "BD405A_Ctrl_D28"               "BD405A_Ctrl_D8"
## [23] "BD405A_HBV_D28"                "BD405A_HBV_D8"
## [25] "BD405A_HBV_HDV_D28"            "BD405A_HBV_HDV_D8"
## [27] "Ctrl_D28_sample_1"             "Ctrl_D28_sample_2"
## [29] "Ctrl_D28_sample_3"             "Ctrl_D8_sample_1"
```

```
## [31] "Ctrl_D8_sample_2"              "Ctrl_D8_sample_3"
## [33] "HBV_D28_sample_1"              "HBV_D28_sample_2"
## [35] "HBV_D28_sample_3"              "HBV_D8_sample_1"
## [37] "HBV_D8_sample_2"               "HBV_D8_sample_3"
## [39] "HU1016 Ctrl D28"               "HU1016 Ctrl D8"
## [41] "HU1016_BD_co_D28"              "HU1016_BD_co_D8"
## [43] "HU1016_B_D28"                  "HU1016_B_D8"
```

```r
##Function to perform regularized log transformation on all counts for each sample.
rld_generation <- function(sampledirectory, sampleset) {
a <- basename(Sys.glob(file.path(sampledirectory, "*.txt")))
sampleTable <- data.frame(sampleName = names(sampleset), 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|Day 8", a), "d8", "d28"),
  replicate = ifelse(grepl("*sample_1h|*D8_ah|*D8_aa|*D8_am|*sample_1m", a), "a",
        ifelse(grepl("*sample_2h|D28_bh|D28_ba|D28_bm|*sample_2m", a), "b",
         ifelse(grepl("*sample_3h| * sample 1h|* sample 1m", a), "c",
          ifelse(grepl("* sample 2h|* sample 2m", a), "d",
           ifelse(grepl("* sample 3h|* sample 3m", a), "e", ""))))))
sampleTable$sampleName <- with(sampleTable, paste(donor, treatment, time, replicate))
dds <- DESeqDataSetFromHTSeqCount(sampleTable = sampleTable, directory = sampledirectory,
  design = ~ donor + treatment)
  dds@colData
  rld <- rlog(dds, blind = TRUE)
}


##Execute function on the human and HBV gene counts.
rld_humanHBV <- rld_generation(humanHBVcounts, humanHBVcounts_readin)

##Pulling just the normalized gene counts for HBV genes (i.e. begin with "AAB") and
##making into a data frame.
mat_humanHBV <- assay(rld_humanHBV)
humanHBV_df <- as.data.frame(mat_humanHBV) %>%
  rownames_to_column(var = "ENSEMBL") %>%
  dplyr::filter(grepl('^AAB', ENSEMBL))

##Now convert back to matrix for heatmap
IDed_m <- as.matrix(humanHBV_df[,c(2:45)])
rownames(IDed_m) <- humanHBV_df[,1]
```

Now making a heat map of the normalized counts of these hepatic genes for each of our samples.

```r
##Reorganizing the row order of samples to what we want for visualization.
sampleTable <- data.frame(sampleName = colnames(IDed_m),
    treatment = ifelse(grepl("mock", colnames(IDed_m)),
                "mock", ifelse(grepl("coinf", colnames(IDed_m)), "coinf", "HBV")),
    donor = ifelse(grepl("HU1019", colnames(IDed_m)), "HU1019",
                ifelse(grepl("HU1007", colnames(IDed_m)), "HU1007",
                    ifelse(grepl("HU1016", colnames(IDed_m)), "HU1016",
                        "HU1020"))),
    time = ifelse(grepl("d8", colnames(IDed_m)), "d8", "d28"))
sampleTable <- with(sampleTable, sampleTable[order(time, treatment),])
```

```r
col.order <- as.character(rev(sampleTable$sampleName))
mat_hep_subset <- IDed_m[, col.order]

##Now plotting the heatmap
png(file = file.path(paste(Sys.Date(), "HBVgenes_heatmap.png")), units = 'in',
    height = 15, width = 30, res = 300)
distance_heatmap <- heatmap.2(mat_hep_subset, trace="none", keysize = 0.7,
   col = brewer.pal(9, "BuPu"), dendrogram = "none", colsep = 1:ncol(mat_hep_subset),
   rowsep = 1:nrow(mat_hep_subset), sepwidth = c(0.05, 0.005), sepcolor = "grey",
   density.info = "none", margin = c(15,20), srtCol = 90, cexRow = 3,
   cexCol = 1.5, scale = "none", Colv = FALSE)
print(distance_heatmap)
```

```
## $rowInd
## [1] 3 4 2 1
##
## $colInd
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## [24] 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
##
## $call
## heatmap.2(x = mat_hep_subset, Colv = FALSE, dendrogram = "none",
##     scale = "none", col = brewer.pal(9, "BuPu"), colsep = 1:ncol(mat_hep_subset),
##     rowsep = 1:nrow(mat_hep_subset), sepcolor = "grey", sepwidth = c(0.05,
##         0.005), trace = "none", margins = c(15, 20), cexRow = 3,
##     cexCol = 1.5, srtCol = 90, keysize = 0.7, density.info = "none")
##
## $carpet
##                     AAB59971.1 AAB59972.1 AAB59970.1 AAB59969.1
## HU1016 mock d8        3.630190   3.783998   4.885246   5.509554
## HU1007 mock d8 c      3.267794   3.428501   4.568846   5.100305
## HU1007 mock d8 b      3.317227   3.476655   4.724557   5.090084
## HU1007 mock d8 a      3.306872   3.466558   4.601262   5.034339
## HU1020 mock d8        3.911853   4.063429   5.501350   5.921221
## HU1019 mock d8        3.837359   3.983105   5.482180   5.630856
## HU1016 HBV d8         8.475574   8.496906  12.120100  12.361799
## HU1007 HBV d8 c       6.996394   7.478894   9.718431  10.301972
## HU1007 HBV d8 b       7.011077   7.277350   9.765541  10.392556
## HU1007 HBV d8 a       6.851971   7.544688   9.632403  10.252400
## HU1020 HBV d8         8.247018   8.478527  10.807106  11.649159
## HU1019 HBV d8         7.864316   8.124552  10.933984  11.282138
## HU1016 coinf d8       8.735713   8.916770  12.211840  12.552092
## HU1020 coinf d8       8.323564   8.865716  11.842802  12.112841
## HU1020 coinf d8 e     9.505782   9.419355   9.820651  11.196851
## HU1020 coinf d8 d     9.436350   9.795485  10.455974  11.768262
## HU1020 coinf d8 c     9.320319   9.430845  10.215615  11.561357
## HU1019 coinf d8       7.366675   7.984110  11.244449  11.473773
## HU1019 coinf d8 a     7.916874   7.904567  11.251276  11.634816
## HU1019 coinf d8 e     8.280361   8.299215   9.780535  10.722551
## HU1019 coinf d8 d     8.428286   7.916992   9.851419  10.247276
## HU1019 coinf d8 c     8.179790   8.225005   9.240743  10.641315
## HU1016 mock d28       4.207128   4.465356   5.242477   5.710713
## HU1007 mock d28 c     3.133385   3.298255   4.413140   5.101753
## HU1007 mock d28 b     3.170758   3.334362   4.439377   5.165417
```

```
## HU1007 mock d28 a    3.183979   3.347156    4.500878    5.076801
## HU1020 mock d28       4.147899   4.290937    5.724800    5.996791
## HU1019 mock d28       3.701071   4.109572    5.489472    6.046333
## HU1016 HBV d28        7.971981   8.249678   11.762396   11.915234
## HU1007 HBV d28 c      4.937127   5.726741    9.064694    9.580853
## HU1007 HBV d28 b      4.294773   6.270959    8.976432    9.316020
## HU1007 HBV d28 a      5.027807   5.472087    7.778488    8.780749
## HU1020 HBV d28        8.064752   8.655708   11.008096   11.526476
## HU1019 HBV d28        7.991893   8.520673   11.528811   11.968814
## HU1016 coinf d28      7.617950   7.882450   11.260839   11.592800
## HU1020 coinf d28      8.724302   9.210499   11.401974   12.105613
## HU1020 coinf d28 e    9.414041   9.480318    9.589468   11.170269
## HU1020 coinf d28 d    8.595996   9.236371    9.388516   10.729347
## HU1020 coinf d28 c    9.203872   9.557055    9.756820   11.169182
## HU1019 coinf d28      6.925495   7.192206   10.469611   10.919761
## HU1019 coinf d28 b    6.821276   7.096755   10.218874   10.504768
## HU1019 coinf d28 e    7.427781   7.637465    8.521400    9.444559
## HU1019 coinf d28 d    7.754708   7.934717    8.551557    9.596797
## HU1019 coinf d28 c    8.114541   8.079414    8.878690    9.944666
##
## $rowDendrogram
## 'dendrogram' with 2 branches and 4 members total, at height 19.66494
##
## $colDendrogram
## 'dendrogram' with 2 branches and 44 members total, at height 1.414214
##
## $breaks
##  [1]  3.133385  4.179908  5.226431  6.272954  7.319477  8.366000  9.412523
##  [8] 10.459046 11.505569 12.552092
##
## $col
## [1] "#F7FCFD" "#E0ECF4" "#BFD3E6" "#9EBCDA" "#8C96C6" "#8C6BB1" "#88419D"
## [8] "#810F7C" "#4D004B"
##
## $colorTable
##         low      high   color
## 1  3.133385  4.179908 #F7FCFD
## 2  4.179908  5.226431 #E0ECF4
## 3  5.226431  6.272954 #BFD3E6
## 4  6.272954  7.319477 #9EBCDA
## 5  7.319477  8.366000 #8C96C6
## 6  8.366000  9.412523 #8C6BB1
## 7  9.412523 10.459046 #88419D
## 8 10.459046 11.505569 #810F7C
## 9 11.505569 12.552092 #4D004B
##
## $layout
## $layout$lmat
##      [,1] [,2]
## [1,]    4    3
## [2,]    2    1
##
## $layout$lhei
## [1] 0.7 4.0
```

```
##
## $layout$lwid
## [1] 0.7 4.0
```

```
dev.off()
```

```
## pdf
##   2
```

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] bindrcpp_0.2.2          data.table_1.11.4
##  [3] genefilter_1.56.0       RColorBrewer_1.1-2
##  [5] tibble_1.4.2            DESeq2_1.14.1
##  [7] SummarizedExperiment_1.4.0 Biobase_2.34.0
##  [9] GenomicRanges_1.26.4    GenomeInfoDb_1.10.3
## [11] IRanges_2.8.2           S4Vectors_0.12.2
## [13] BiocGenerics_0.20.0     openxlsx_4.1.0
## [15] reshape2_1.4.3          gplots_3.0.1
## [17] stringr_1.3.1           dplyr_0.7.6
##
## loaded via a namespace (and not attached):
##  [1] bit64_0.9-7         splines_3.3.3       gtools_3.8.1
##  [4] Formula_1.2-3       assertthat_0.2.0    latticeExtra_0.6-28
##  [7] blob_1.1.1          yaml_2.2.0          RSQLite_2.1.1
## [10] pillar_1.3.0        backports_1.1.2     lattice_0.20-35
## [13] glue_1.3.0          digest_0.6.15       XVector_0.14.1
## [16] checkmate_1.8.5     colorspace_1.3-2    htmltools_0.3.6
## [19] Matrix_1.2-8        plyr_1.8.4          XML_3.98-1.12
## [22] pkgconfig_2.0.1     zlibbioc_1.20.0     xtable_1.8-2
## [25] purrr_0.2.5         scales_0.5.0        gdata_2.18.0
## [28] BiocParallel_1.8.2  htmlTable_1.12      annotate_1.52.1
## [31] ggplot2_3.0.0       nnet_7.3-12         lazyeval_0.2.1
## [34] survival_2.42-6     magrittr_1.5        crayon_1.3.4
## [37] memoise_1.1.0       evaluate_0.11       foreign_0.8-71
## [40] tools_3.3.3         locfit_1.5-9.1      munsell_0.5.0
## [43] cluster_2.0.5       zip_1.0.0          AnnotationDbi_1.36.2
## [46] caTools_1.17.1.1    rlang_0.2.1         grid_3.3.3
## [49] RCurl_1.95-4.11     rstudioapi_0.7      htmlwidgets_1.2
## [52] bitops_1.0-6        base64enc_0.1-3     rmarkdown_1.10
## [55] gtable_0.2.0        DBI_1.0.0           R6_2.2.2
## [58] gridExtra_2.3       knitr_1.20          bit_1.1-14
```

```
## [61] bindr_0.1.1          Hmisc_4.1-1        rprojroot_1.3-2
## [64] KernSmooth_2.23-15   stringi_1.2.4      Rcpp_0.12.18
## [67] geneplotter_1.52.0   rpart_4.1-13       acepack_1.4.1
## [70] tidyselect_0.2.4
```