HBV gene expression

Purpose:

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

Load required libraries

```r
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
```

```r
library(stringr)
library(gplots)
```

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

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

```r
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 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")'.
```

```r
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_Ctrl_D28"      "BD330_Ctrl_D8"       "BD330_HBV_D28"
##  [4] "BD330_HBV_D8"        "BD330_HBV_HDV_D28_b" "BD330_HBV_HDV_D28"
##  [7] "BD330_HBV_HDV_D8_a"  "BD330_HBV_HDV_D8"    "BD405A_Ctrl_D28"
## [10] "BD405A_Ctrl_D8"      "BD405A_HBV_D28"      "BD405A_HBV_D8"
## [13] "BD405A_HBV_HDV_D28"  "BD405A_HBV_HDV_D8"   "Ctrl_D28_sample_1"
## [16] "Ctrl_D28_sample_2"   "Ctrl_D28_sample_3"   "Ctrl_D8_sample_1"
## [19] "Ctrl_D8_sample_2"    "Ctrl_D8_sample_3"    "HBV_D28_sample_1"
## [22] "HBV_D28_sample_2"    "HBV_D28_sample_3"    "HBV_D8_sample_1"
## [25] "HBV_D8_sample_2"     "HBV_D8_sample_3"     "HU1016_BD_co_D28"
## [28] "HU1016_BD_co_D8"     "HU1016_B_D28"        "HU1016_B_D8"
```

```
##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 =
```

```r
  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_ah|*D8_aa", a), "a",
ifelse(grepl("*sample_2|D28_bh|D28_ba", a),"b", ifelse(grepl("*sample_3", a), "c",""))))
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") out 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(2:31)])
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),])
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
```

```
##
## $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
## HU1007 mock d8 c     3.444013   3.605852   4.832495   5.095296
## HU1007 mock d8 b     3.501088   3.665385   4.987604   5.110558
## HU1007 mock d8 a     3.476548   3.639791   4.867035   5.040963
## HU1020 mock d8       4.080980   4.270122   5.810612   6.038073
## HU1019 mock d8       3.968809   4.143173   5.721576   5.709347
## HU1016 HBV d8        8.340174   8.411815  12.238466  12.429435
## HU1007 HBV d8 c      6.932203   7.457284   9.887312  10.408134
## HU1007 HBV d8 b      6.944184   7.253748   9.934225  10.499538
## HU1007 HBV d8 a      6.779370   7.513262   9.790362  10.347364
## HU1020 HBV d8        8.168949   8.450620  10.980775  11.770962
## HU1019 HBV d8        7.747788   8.055846  11.063447  11.354886
## HU1016 coinf d8      8.615761   8.847841  12.345497  12.636235
## HU1020 coinf d8      8.213058   8.805663  11.985563  12.203588
## HU1019 coinf d8      7.266928   7.929935  11.389913  11.563267
## HU1019 coinf d8 a    7.818483   7.853631  11.397700  11.726857
## HU1007 mock d28 c    3.285104   3.439898   4.627050   5.029624
## HU1007 mock d28 b    3.325075   3.481677   4.666153   5.105776
## HU1007 mock d28 a    3.342786   3.500181   4.726291   5.027424
## HU1020 mock d28      4.267451   4.452481   6.006960   6.117037
## HU1019 mock d28      3.867294   4.240953   5.727538   6.088881
## HU1016 HBV d28       7.847805   8.174608  11.891211  11.990565
## HU1007 HBV d28 c     4.869272   5.662269   9.166843   9.612587
## HU1007 HBV d28 b     4.296709   6.228353   9.113005   9.378421
## HU1007 HBV d28 a     4.985146   5.447593   7.913971   8.838318
## HU1020 HBV d28       7.953003   8.596597  11.154561  11.618686
## HU1019 HBV d28       7.890024   8.466069  11.672067  12.059807
## HU1016 coinf d28     7.494311   7.805947  11.384541  11.661718
## HU1020 coinf d28     8.627017   9.166362  11.561829  12.215301
## HU1019 coinf d28     6.827512   7.140532  10.622735  11.014228
## HU1019 coinf d28 b   6.714177   7.033017  10.350767  10.573936
##
## $rowDendrogram
## 'dendrogram' with 2 branches and 4 members total, at height 18.75746
##
## $colDendrogram
## 'dendrogram' with 2 branches and 30 members total, at height 1.414214
##
## $breaks
##  [1]  3.285104  4.324118  5.363133  6.402147  7.441162  8.480177  9.519191
##  [8] 10.558206 11.597220 12.636235
##
## $col
## [1] "#F7FCFD" "#E0ECF4" "#BFD3E6" "#9EBCDA" "#8C96C6" "#8C6BB1" "#88419D"
## [8] "#810F7C" "#4D004B"
```

```
##
## $colorTable
##         low      high   color
## 1  3.285104  4.324118 #F7FCFD
## 2  4.324118  5.363133 #E0ECF4
## 3  5.363133  6.402147 #BFD3E6
## 4  6.402147  7.441162 #9EBCDA
## 5  7.441162  8.480177 #8C96C6
## 6  8.480177  9.519191 #8C6BB1
## 7  9.519191 10.558206 #88419D
## 8 10.558206 11.597220 #810F7C
## 9 11.597220 12.636235 #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
```

```r
dev.off()
```

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

Session Info

```r
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               data.table_1.10.0
##  [3] genefilter_1.54.2          RColorBrewer_1.1-2
##  [5] tibble_1.3.3               DESeq2_1.12.4
##  [7] SummarizedExperiment_1.2.3 Biobase_2.32.0
##  [9] GenomicRanges_1.24.3       GenomeInfoDb_1.8.7
## [11] IRanges_2.6.1              S4Vectors_0.10.3
## [13] BiocGenerics_0.18.0        openxlsx_4.0.17
## [15] reshape2_1.4.2             gplots_3.0.1
## [17] stringr_1.2.0              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        annotate_1.50.1      ggplot2_2.2.1
## [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          locfit_1.5-9.1
## [37] munsell_0.4.3        cluster_2.0.6        AnnotationDbi_1.34.4
## [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
```