

DGE sorting

Purpose:

To sort the feature counts from Galaxy output into those from mouse versus those from human and HBV combined. Prepare files of each of these separated groups. This data includes sequenced samples up until July 2018.

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

##
## 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
```

```
Read in the feature counts downloaded from Galaxy
```

```
##All counts stored in the folder "Featurecounts"
counts <- "Featurecounts"
sampleCounts <- basename(Sys.glob(file.path(counts, "*.tabular")))
```

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

```
##Read in all of the count files
```

```
all_counts <- lapply(file.path(counts, sampleCounts), exptcounts)
names(all_counts) <- sub('.tabular', '', sampleCounts)
names(all_counts)
```

```
## [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"            "BD330_HBV_HDV_D28_b"
## [13] "BD330_HBV_HDV_D8"             "BD330_HBV_HDV_D8_a"
## [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"
```

Now include only the feature counts arising from human, HBV, or HDV genes.

```
##Function looking for only the human ENSEMBL gene ids (i.e. beginning with "ENSG") or HBV
##symbols (which begin with AAB for HBV or AAA for HDV)
humanHBV <- function(dataframes) {
  d <- dataframes[grep("^AA|^ENSG", dataframes[,1]),] %>%
    droplevels()
}

##Apply function to all_counts
all_humanHBV <- lapply(all_counts, humanHBV)
colnames <- c("ID", "count")
all_humanHBV <- lapply(all_humanHBV, setNames, colnames)
```

Feature counts arising from mouse genes

```
##Function for only the mouse ENSEMBL gene IDs (i.e. beginning with "ENSM")
mouse <- function(dataframes) {
  d <- dataframes[grep("^ENSM", dataframes[,1]),] %>%
    droplevels()
}

##Apply function to all_counts
all_mouse <- lapply(all_counts, mouse)
colnames <- c("ID", "count")
all_mouse <- lapply(all_mouse, setNames, colnames)
```

Now assign feature counts to different subsets for performing DGE analysis. Human

```
d8 <- names(all_humanHBV)[grep("*D8| Day 8", names(all_humanHBV))]

d8_counts <- all_humanHBV[match(d8, names(all_humanHBV))]
d8_ctrlvHBV <- d8_counts[grep("*HBV_D|_B_|Ctrl", names(d8_counts))]
d8_ctrlvcoinf <- d8_counts[grep("*HBV_HDV|_co_|Ctrl|HBV_HDV", names(d8_counts))]
d8_coinfvHBV <- d8_counts[grep("*HBV_D|_B_|HBV_HDV|_co_|HBV_HDV", names(d8_counts))]

d28 <- names(all_humanHBV)[grep("*D28|Day 28", names(all_humanHBV))]

d28_counts <- all_humanHBV[match(d28, names(all_humanHBV))]
d28_ctrlvHBV <- d28_counts[grep("*HBV_D|_B_|Ctrl", names(d28_counts))]
d28_ctrlvcoinf <- d28_counts[grep("*HBV_HDV|_co_|Ctrl|HBV_HDV", names(d28_counts))]
d28_coinfvHBV <- d28_counts[grep("*HBV_D|_B_|HBV_HDV|_co_|HBV_HDV", names(d28_counts))]
```

Mouse

```

##All human and HBV genes
d8_mouse <- names(all_mouse)[grep("*D8| Day 8", names(all_mouse))]

d8_counts_mouse <- all_mouse[match(d8_mouse, names(all_mouse))]
d8_ctrlvHBV_mouse <- d8_counts_mouse[grep("*HBV_D|_B_|Ctrl", names(d8_counts_mouse))]
d8_ctrlvcoinf_mouse <- d8_counts_mouse[grep("*HBV_HDV|_co_|Ctrl|HBV_HDV",
names(d8_counts_mouse))]
d8_coinfvHBV_mouse <- d8_counts_mouse[grep("*HBV_D|_B_|HBV_HDV|_co_|HBV_HDV",
names(d8_counts_mouse))]

d28_mouse <- names(all_mouse)[grep("*D28|Day 28", names(all_mouse))]

d28_counts_mouse <- all_mouse[match(d28_mouse, names(all_mouse))]
d28_ctrlvHBV_mouse <- d28_counts_mouse[grep("*HBV_D|_B_|Ctrl", names(d28_counts_mouse))]
d28_ctrlvcoinf_mouse <- d28_counts_mouse[grep("*HBV_HDV|_co_|Ctrl|HBV_HDV",
names(d28_counts_mouse))]
d28_coinfvHBV_mouse <- d28_counts_mouse[grep("*HBV_D|_B_|HBV_HDV|_co_|HBV_HDV",
names(d28_counts_mouse))]

```

Make files of these separated feature counts Human

```

for(i in names(d8_ctrlvHBV)) {
  filename <- paste(i, "humanHBVgenes.txt", sep = "")
  write.table(d8_ctrlvHBV[i], file = file.path("Human d8_ctrlvHBV", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(d8_ctrlvcoinf)) {
  filename <- paste(i, "humanHBVgenes.txt", sep = "")
  write.table(d8_ctrlvcoinf[i], file = file.path("Human d8_ctrlvcoinf", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(d8_coinfvHBV)) {
  filename <- paste(i, "humanHBVgenes.txt", sep = "")
  write.table(d8_coinfvHBV[i], file = file.path("Human d8_coinfvHBV", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(d28_ctrlvHBV)) {
  filename <- paste(i, "humanHBVgenes.txt", sep = "")
  write.table(d28_ctrlvHBV[i], file = file.path("Human d28_ctrlvHBV", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(d28_ctrlvcoinf)) {
  filename <- paste(i, "humanHBVgenes.txt", sep = "")
  write.table(d28_ctrlvcoinf[i], file = file.path("Human d28_ctrlvcoinf", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(d28_coinfvHBV)) {
  filename <- paste(i, "humanHBVgenes.txt", sep = "")
  write.table(d28_coinfvHBV[i], file = file.path("Human d28_coinfvHBV", filename),

```

```

        col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(all_humanHBV)) {
  filename <- paste(i, "humanHBVgenes.txt", sep = "")
  write.table(all_humanHBV[i], file = file.path("All human HBV genes", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

```

Mouse

```

for(i in names(d8_ctrlvHBV_mouse)) {
  filename <- paste(i, "mousegenes.txt", sep = "")
  write.table(d8_ctrlvHBV_mouse[i], file = file.path("Mouse d8_ctrlvHBV", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(d8_ctrlvcoinf_mouse)) {
  filename <- paste(i, "mousegenes.txt", sep = "")
  write.table(d8_ctrlvcoinf_mouse[i], file = file.path("Mouse d8_ctrlvcoinf", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(d8_coinfvHBV_mouse)) {
  filename <- paste(i, "mousegenes.txt", sep = "")
  write.table(d8_coinfvHBV_mouse[i], file = file.path("Mouse d8_coinfvHBV", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(d28_ctrlvHBV_mouse)) {
  filename <- paste(i, "mousegenes.txt", sep = "")
  write.table(d28_ctrlvHBV_mouse[i], file = file.path("Mouse d28_ctrlvHBV", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(d28_ctrlvcoinf_mouse)) {
  filename <- paste(i, "mousegenes.txt", sep = "")
  write.table(d28_ctrlvcoinf_mouse[i], file = file.path("Mouse d28_ctrlvcoinf", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(d28_coinfvHBV_mouse)) {
  filename <- paste(i, "mousegenes.txt", sep = "")
  write.table(d28_coinfvHBV_mouse[i], file = file.path("Mouse d28_coinfvHBV", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

for(i in names(all_mouse)) {
  filename <- paste(i, "mousegenes.txt", sep = "")
  write.table(all_mouse[i], file = file.path("All mouse genes", filename),
    col.names = FALSE, row.names=FALSE,sep="\t",quote=FALSE)
}

```

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] viridis_0.5.1          viridisLite_0.3.0
## [3] ggrepel_0.8.0          data.table_1.11.4
## [5] genefilter_1.56.0      RColorBrewer_1.1-2
## [7] gplots_3.0.1           DESeq2_1.14.1
## [9] SummarizedExperiment_1.4.0 Biobase_2.34.0
## [11] GenomicRanges_1.26.4   GenomeInfoDb_1.10.3
## [13] IRanges_2.8.2          S4Vectors_0.12.2
## [15] BiocGenerics_0.20.0    openxlsx_4.1.0
## [17] reshape2_1.4.3         ggplot2_3.0.0
## [19] 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            pillar_1.3.0
## [10] RSQLite_2.1.1          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        purrr_0.2.5
## [25] xtable_1.8-2           scales_0.5.0           gdata_2.18.0
## [28] BiocParallel_1.8.2     tibble_1.4.2           htmlTable_1.12
## [31] annotate_1.52.1        withr_2.1.2            nnet_7.3-12
## [34] lazyeval_0.2.1         survival_2.42-6        magrittr_1.5
## [37] crayon_1.3.4           memoise_1.1.0          evaluate_0.11
## [40] foreign_0.8-71         tools_3.3.3            locfit_1.5-9.1
## [43] munsell_0.5.0          cluster_2.0.5          zip_1.0.0
## [46] AnnotationDbi_1.36.2   bindrcpp_0.2.2         caTools_1.17.1.1
## [49] rlang_0.2.1            grid_3.3.3            RCurl_1.95-4.11
## [52] rstudioapi_0.7         htmlwidgets_1.2        bitops_1.0-6
## [55] base64enc_0.1-3        rmarkdown_1.10         gtable_0.2.0
## [58] DBI_1.0.0              R6_2.2.2              gridExtra_2.3
## [61] knitr_1.20             bit_1.1-14            bindr_0.1.1
## [64] Hmisc_4.1-1            rprojroot_1.3-2        KernSmooth_2.23-15
## [67] stringi_1.2.4          Rcpp_0.12.18           geneplotter_1.52.0
## [70] rpart_4.1-13           acepack_1.4.1          tidyselect_0.2.4
```