Comparing the most differentially expressed genes when using human or species-specific genome

Load required libraries

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

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

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

```
library(tibble)
library(stringr)
library(biomaRt)
library(ggrepel)
```

```
## Loading required package: ggplot2
```

```
library(gplots)
```

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

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

```
library(ggplot2)
library(RColorBrewer)
library(stringr)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(devtools)
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(purrr)
```

```
##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:data.table':
##
##     transpose
```

```
## The following object is masked from 'package:plyr':
##
##     compact
```

```
library(gtools)
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, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind,
##     colMeans, colnames, colSums, dirname, 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, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, which.max,
##     which.min
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following objects are masked from 'package:data.table':
##
##     first, second
```

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

```
##     space
## The following objects are masked from 'package:dplyr':
##
##     first, rename
## The following object is masked from 'package:plyr':
##
##     rename
## The following object is masked from 'package:base':
##
##     expand.grid
## Loading required package: IRanges
##
## Attaching package: 'IRanges'
## The following object is masked from 'package:purrr':
##
##     reduce
## The following object is masked from 'package:data.table':
##
##     shift
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
## The following object is masked from 'package:plyr':
##
##     desc
## 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")'.
## Loading required package: DelayedArray
## Loading required package: matrixStats
##
## Attaching package: 'matrixStats'
## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians
## The following object is masked from 'package:dplyr':
##
##     count
```

```
## The following object is masked from 'package:plyr':
##
##     count
```

```
## Loading required package: BiocParallel
```

```
##
## Attaching package: 'DelayedArray'
```

```
## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
```

```
## The following object is masked from 'package:purrr':
##
##     simplify
```

```
## The following objects are masked from 'package:base':
##
##     aperm, apply
```

```
library(directlabels)
```

Purpose

Comparing the top 500 most significantly differentially expressed genes between the two mapping methods

Loading in the DGE profiles

```
HumanMapped_DGE <- "Expanded_Design_Factor_SpeciesHomologs_Outputs_HumanMapped"
sampleFiles_HumanMapped <- list.files(basename(Sys.glob(file.path(HumanMapped_DGE))),
                  pattern = "treated dds1 HumanMapped_DGE_results.txt|*related.*.txt")

sampleNames_HumanMapped <- sub('_treated_v_mock|.treatmenttreated.*', '', sampleFiles_HumanMapped) %>%
  sub('\\d+-\\d+-\\d+\\s', '', .) %>%
  sub("HumanMapped.*", "", .) %>%
  sub('species', '', .)
length(sampleFiles_HumanMapped)
```

```
## [1] 16
```

```
SpeciesMapped_DGE <- "Expanded_Design_Factor_SpeciesHomologs_Outputs_SpeciesMapped"

sampleFiles_SpeciesMapped <- list.files(basename(Sys.glob(file.path(SpeciesMapped_DGE))),
                  pattern = "*treated dds1 DGE_results.txt|*related.*.txt")

sampleNames_SpeciesMapped <- sub('_treated_v_mock|.treatmenttreated.*', '', sampleFiles_SpeciesMapped) %
  sub('\\d+-\\d+-\\d+\\s', '', .) %>%
  sub("SpeciesMapped.*", "", .) %>%
  sub('species', '', .)
length(sampleFiles_SpeciesMapped)
```

```
## [1] 16
```

```
exptcounts <- function(files) {
  d <- read.delim(files)
  d
}
```

```
human_DGEs_read <-llply(file.path(HumanMapped_DGE, sampleFiles_HumanMapped), exptcounts)
names(human_DGEs_read) <- sampleNames_HumanMapped

species_DGEs_read <-llply(file.path(SpeciesMapped_DGE, sampleFiles_SpeciesMapped), exptcounts)
names(species_DGEs_read) <- sampleNames_SpeciesMapped
```

Getting the top 500 DGEs for each mapping method

```
output_dir <- "ReferenceGenomes_TopHit_Comparison_Output"
##First focus on the NHP DGEs for both mapping approaches. We don't need to include the
##"human-related" samples for the time being.
human_mapped <- human_DGEs_read[!grepl("^human_related", names(human_DGEs_read))] %>%
  llply(., function(x) {
    colnames(x) <- c("X", paste(colnames(x[-1]), "human_ref", sep = "_"))
    x}) %>%
  llply(., function(x) {
    b <- dplyr::select(x,  matches('X|log2FoldChange|padj|SYMBOL'))
    b
  })

species_mapped <- species_DGEs_read[!grepl("^human_related", names(species_DGEs_read))] %>%
  llply(., function(x) {
    colnames(x) <- c("X", paste(colnames(x[-1]), "species_ref", sep = "_"))
    x}) %>%
  llply(., function(x) {
    b <- dplyr::select(x,  matches('X|log2FoldChange|padj|SYMBOL'))
    b
  })

##Pulling out top 500 most significant genes
top_sig <- function(input) {
  e <- arrange_at(input, vars(contains('padj')))
  e[1:500,]
}

human_mapped_sig <- llply(human_mapped, top_sig)
species_mapped_sig <- llply(species_mapped, top_sig)

##What is in common between the two results for each species
human_species_mapped_common <- mapply(function(x, y) inner_join(x, y, by = "X"), x = human_mapped_sig,
            y = species_mapped_sig, SIMPLIFY = FALSE)

##Function to mark genes as "different" or not between the two mapping methods if the ratio of the
##log2FoldChange is > 1.5 or < 0.67 and then also making a condition that the log2FoldChange needs to ha
##an absolute value of greater than or equal to two. Additional column made to mark genes as significant
##or not, as well.
pinpointer <- function(x) {
x$differs <- ifelse(
  (((x$log2FoldChange_species_ref/x$log2FoldChange_human_ref) < 0.67) |
  ((x$log2FoldChange_species_ref/x$log2FoldChange_human_ref) > 1.5) &
  (abs(x$log2FoldChange_species_ref) >= 2 | abs(x$log2FoldChange_human_ref >= 2)) |
  is.na(x$log2FoldChange_species_ref) | is.na(x$log2FoldChange_human_ref)), "different", "not")

x$significance <-
```

```
  ifelse((x$padj_species_ref > 0.05 | x$padj_human_ref > 0.05 | is.na(x$padj_human_ref) | is.na(x$padj_s
         "nonsig", "sig")
x$SYMBOL <-
  ifelse(is.na(x$SYMBOL_species_ref) | is.na(x$SYMBOL_human_ref), as.character(x$X), as.character(x$SYM
x
}

##For the top 500 significant genes after mapping to human, now looking at the log2FC and padj values in
##mapping to species-specific genomes.
human_sig_in_species <- mapply(function(x, y) inner_join(x, y, by = "X"), x = human_mapped_sig,
            y = species_mapped, SIMPLIFY = FALSE) %>%
  llply(., function(x) dplyr::select(x, matches('X|log2FoldChange|padj|SYMBOL'))) %>%
  llply(., pinpointer)

species_sig_in_human <- mapply(function(x, y) inner_join(x, y, by = "X"), x = species_mapped_sig,
            y = human_mapped, SIMPLIFY = FALSE) %>%
  llply(., function(x) dplyr::select(x, matches('X|log2FoldChange|padj|SYMBOL'))) %>%
  llply(., pinpointer)

##Making tables of the data
for (i in 1:8) {
a <- human_sig_in_species[[i]]
c <- names(human_sig_in_species[i])

write.csv(a, file.path(output_dir, paste(Sys.Date(), c,
            "HumanTop500Sig_ExpressionInSpecies.csv")))

}
for (i in 1:8) {
a <- species_sig_in_human[[i]]
c <- names(species_sig_in_human[i])

write.csv(a, file.path(output_dir, paste(Sys.Date(), c,
            "SpeciesTop500Sig_ExpressionInHuman.csv")))

}

##Now plotting the data and labeling points as stipulated if the gene's differential expression is consi
##different between the two mapping methods by the stipulations we made above.
for(i in 1:8) {
  a <- human_sig_in_species[[i]]
  a$X <- factor(a$X,
     levels = a$X[order(a$log2FoldChange_human_ref)])
  a_subset <- dplyr::select(a, X, log2FoldChange_human_ref, SYMBOL, differs, significance, log2FoldChang
  a.m <- melt(a_subset)
  c <- names(human_sig_in_species[i])
  ggplot(a.m, aes(x=X, y=value, color=variable, label = SYMBOL, size = significance)) +
    geom_point(alpha = 0.5) +
    scale_size_manual(values = c(5, 3)) +
  geom_text_repel(data = filter(a.m, differs == "different" | significance == "nonsig") %>%
                filter(., variable == "log2FoldChange_human_ref"),
            aes(label = as.character(SYMBOL)), show.legend = FALSE,
            force = 1, min.segment.length = 0,
```

```
            vjust = 1, direction ='y', nudge_y=1.5, segment.size = 0.25, color = "gray42") +
 geom_segment(data = filter(a.m, differs == "different" | significance == "nonsig") %>%
                 filter(., variable == "log2FoldChange_human_ref"),
                aes(x=X, xend=X, y=-5, yend=value), size = 0.25, colour = "gray42") +
    labs(x = "Transcript", y = "log2FoldChange", title = c) +
 scale_color_manual(name = "Reference Genome",
                    labels = c("Human", "Species-Specific"), values = c("#7fbf7b", "#af8dc3")) +
  theme(axis.text.x = element_blank(), axis.title = element_text(size=20),
        plot.title = element_text(size = 40, hjust = 0.5),
        panel.background = element_rect(fill = "white"),
        axis.text.y = element_text(size = 20),
        legend.text = element_text(size = 14),
        legend.background = element_rect(fill = "white"),
        axis.line= element_line(size = 1, colour = "black"),
        panel.grid.major.y = element_line(colour = "black"),
        panel.grid.minor.y = element_line(colour = "black")) +
      coord_cartesian(ylim = c(-5, 11), clip = "off")
  ggsave(filename = file.path(output_dir, paste(Sys.Date(),
        c, "plot_500_human_sig_versus_species.pdf")), device = "pdf", width = 15, height = 5)
}
```

```
## Using X, SYMBOL, differs, significance as id variables
## Using X, SYMBOL, differs, significance as id variables
## Using X, SYMBOL, differs, significance as id variables
## Using X, SYMBOL, differs, significance as id variables
## Using X, SYMBOL, differs, significance as id variables
## Using X, SYMBOL, differs, significance as id variables
## Using X, SYMBOL, differs, significance as id variables
## Using X, SYMBOL, differs, significance as id variables
```

```
for(i in 1:8) {
  a <- species_sig_in_human[[i]]
  a$X <- factor(a$X,
     levels = a$X[order(a$log2FoldChange_species_ref)])
  a_subset <- dplyr::select(a, X, log2FoldChange_human_ref, SYMBOL, differs, significance, log2FoldChang
  a.m <- melt(a_subset)
  c <- names(species_sig_in_human[i])
  ggplot(a.m, aes(x=X, y=value, color=variable, label = SYMBOL, size = significance)) +
    geom_point(alpha = 0.5) +
    scale_size_manual(values = c(5, 3)) +
  geom_text_repel(data = filter(a.m, differs == "different" | significance == "nonsig") %>%
                 filter(., variable == "log2FoldChange_species_ref"),
            aes(label = as.character(SYMBOL)), show.legend = FALSE,
            force = 1, min.segment.length = 0,
            vjust = 1, direction ='y', nudge_y=1.5, segment.size = 0.25, color = "gray42") +
 geom_segment(data = filter(a.m, differs == "different" | significance == "nonsig") %>%
                 filter(., variable == "log2FoldChange_species_ref"),
                aes(x=X, xend=X, y=-5, yend=value), size = 0.25, colour = "gray42") +
    labs(x = "Transcript", y = "log2FoldChange", title = c) +
 scale_color_manual(name = "Reference Genome",
                    labels = c("Human", "Species-Specific"), values = c("#7fbf7b", "#af8dc3")) +
  theme(axis.text.x = element_blank(), axis.title = element_text(size=20),
        plot.title = element_text(size = 40, hjust = 0.5),
        panel.background = element_rect(fill = "white"),
```

```
        axis.text.y = element_text(size = 20),
        legend.text = element_text(size = 14),
        legend.background = element_rect(fill = "white"),
        axis.line= element_line(size = 1, colour = "black"),
        panel.grid.major.y = element_line(colour = "black"),
        panel.grid.minor.y = element_line(colour = "black")) +
      coord_cartesian(ylim = c(-5, 11), clip = "off")
  ggsave(filename = file.path(output_dir, paste(Sys.Date(),
        c, "plot_500_species_sig_versus_human.pdf")), device = "pdf", width = 15, height = 5)
}
```

```
## Using X, SYMBOL, differs, significance as id variables
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Using X, SYMBOL, differs, significance as id variables
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

```
## Using X, SYMBOL, differs, significance as id variables
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Using X, SYMBOL, differs, significance as id variables
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Using X, SYMBOL, differs, significance as id variables
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

```
## Using X, SYMBOL, differs, significance as id variables
## Using X, SYMBOL, differs, significance as id variables
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

```
## Using X, SYMBOL, differs, significance as id variables
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
sapply(species_sig_in_human,
       function(x) nrow(dplyr::filter(x, is.na(log2FoldChange_species_ref) | is.na(log2FoldChange_human_
```

```
##            bonobo        chimpanzee           gorilla      olive_baboon
##                 1                 3                 1                 1
##         orangutan pigtailed_macaque    rhesus_macaque   squirrel_monkey
##                 2                 0                 5                 1
```

```
sapply(human_sig_in_species,
       function(x) nrow(dplyr::filter(x, is.na(log2FoldChange_species_ref) | is.na(log2FoldChange_human_
```

```
##            bonobo        chimpanzee           gorilla      olive_baboon
##                 0                 0                 0                 0
##         orangutan pigtailed_macaque    rhesus_macaque   squirrel_monkey
##                 0                 0                 0                 0
```

```
sapply(species_sig_in_human, function(x) nrow(dplyr::filter(x, significance == "nonsig" & differs == "di
```

```
##            bonobo        chimpanzee           gorilla      olive_baboon
##                 3                 6                 4                 2
##         orangutan pigtailed_macaque    rhesus_macaque   squirrel_monkey
##                 6                 4                 7                 4
```

```
sapply(human_sig_in_species, function(x) nrow(dplyr::filter(x, significance == "nonsig" & differs == "di
```

```
##            bonobo        chimpanzee           gorilla      olive_baboon
##                 1                 2                 1                 4
##         orangutan pigtailed_macaque    rhesus_macaque   squirrel_monkey
##                 3                 5                 8                 4
```

```
##Comparing genes with significantly different expression from mapping to species-specific genome versus
##corresponding values when using human genome -- the Spearman coefficients
Spearman_coeff_species_sig_in_human <- list()
for(i in 1:8) {
  a <- dplyr::select(species_sig_in_human[[i]], log2FoldChange_species_ref, log2FoldChange_human_ref) %>
               na.omit(.)
  correlation <- cor(a$log2FoldChange_species_ref, a$log2FoldChange_human_ref, method = "spearman") %>%
    round(., digits = 3)
  species <- names(species_sig_in_human[i])
  Spearman_coeff_species_sig_in_human[[i]] <- cbind(species, correlation)
}
write.csv(do.call(rbind, Spearman_coeff_species_sig_in_human),
        file = file.path(output_dir, paste(Sys.Date(), "Spearman_coeff_species_sig_in_human.csv")))

##Comparing genes with significantly different expression from mapping to human genome versus the
##corresponding values when using species-specific genomes -- the Spearman coefficients
Spearman_coeff_human_sig_in_species <- list()
for(i in 1:8) {
  a <- dplyr::select(human_sig_in_species[[i]], log2FoldChange_species_ref, log2FoldChange_human_ref) %>
               na.omit(.)
  correlation <- cor(a$log2FoldChange_species_ref, a$log2FoldChange_human_ref, method = "spearman") %>%
    round(., digits = 3)
  species <- names(human_sig_in_species[i])
  Spearman_coeff_human_sig_in_species[[i]] <- cbind(species, correlation)
}
write.csv(do.call(rbind, Spearman_coeff_human_sig_in_species),
        file = file.path(output_dir, paste(Sys.Date(), "Spearman_coeff_human_sig_in_species.csv")))
```

Session Info

```
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## 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             directlabels_2018.05.22
##  [3] DESeq2_1.22.2              SummarizedExperiment_1.12.0
##  [5] DelayedArray_0.8.0         BiocParallel_1.16.5
##  [7] matrixStats_0.54.0         Biobase_2.42.0
##  [9] GenomicRanges_1.34.0       GenomeInfoDb_1.18.1
## [11] IRanges_2.16.0             S4Vectors_0.20.1
## [13] BiocGenerics_0.28.0        gtools_3.8.1
## [15] purrr_0.2.5                data.table_1.12.0
## [17] usethis_1.4.0              devtools_2.0.1
## [19] viridis_0.5.1              viridisLite_0.3.0
## [21] RColorBrewer_1.1-2         gplots_3.0.1
## [23] ggrepel_0.8.0              ggplot2_3.1.0
## [25] biomaRt_2.38.0             stringr_1.3.1
## [27] tibble_2.0.1               dplyr_0.7.8
## [29] plyr_1.8.4
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-6           fs_1.2.6               bit64_0.9-7
##  [4] progress_1.2.0         httr_1.4.0             rprojroot_1.3-2
##  [7] tools_3.5.2            backports_1.1.3        R6_2.3.0
## [10] rpart_4.1-13           KernSmooth_2.23-15     Hmisc_4.1-1
## [13] DBI_1.0.0              lazyeval_0.2.1         colorspace_1.4-0
## [16] nnet_7.3-12            withr_2.1.2            tidyselect_0.2.5
## [19] gridExtra_2.3          prettyunits_1.0.2      processx_3.2.1
## [22] bit_1.1-14             compiler_3.5.2         cli_1.0.1
## [25] htmlTable_1.13.1       desc_1.2.0             labeling_0.3
## [28] checkmate_1.9.1        caTools_1.17.1.1       scales_1.0.0
## [31] quadprog_1.5-7         genefilter_1.64.0      callr_3.1.1
## [34] digest_0.6.18          foreign_0.8-71         rmarkdown_1.11
## [37] XVector_0.22.0         base64enc_0.1-3        pkgconfig_2.0.2
## [40] htmltools_0.3.6        sessioninfo_1.1.1      htmlwidgets_1.3
## [43] rlang_0.3.1            rstudioapi_0.9.0       RSQLite_2.1.1
## [46] bindr_0.1.1            acepack_1.4.1          RCurl_1.95-4.11
## [49] magrittr_1.5           Formula_1.2-3          GenomeInfoDbData_1.2.0
## [52] Matrix_1.2-15          Rcpp_1.0.0             munsell_0.5.0
## [55] stringi_1.2.4          yaml_2.2.0             zlibbioc_1.28.0
## [58] pkgbuild_1.0.2         grid_3.5.2             blob_1.1.1
## [61] gdata_2.18.0           crayon_1.3.4           lattice_0.20-38
## [64] splines_3.5.2          annotate_1.60.0        hms_0.4.2
## [67] locfit_1.5-9.1         knitr_1.21             ps_1.3.0
## [70] pillar_1.3.1           reshape2_1.4.3         geneplotter_1.60.0
## [73] pkgload_1.0.2          XML_3.98-1.16          glue_1.3.0
## [76] evaluate_0.12          latticeExtra_0.6-28    remotes_2.0.2
## [79] gtable_0.2.0           assertthat_0.2.0       xfun_0.4
## [82] xtable_1.8-3           survival_2.43-3        AnnotationDbi_1.44.0
## [85] memoise_1.1.0          cluster_2.0.7-1
```