

# Venn diagrams\_overlap

Required packages

```
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(tibble)
library(grid)
library(stringr)
library(gplots)

##
## Attaching package: 'gplots'
## The following object is masked from 'package:stats':
##
##   lowess
library(VennDiagram)

## Loading required package: futile.logger
library(extrafont)

## Registering fonts with R
```

## Purpose:

To determine the overlap in differentially expressed genes between timepoints d8 and d28 post-infection for the various infection conditions.

Upload files

```
data_dir <- "Human DGEs_donortreatment"
sampleFiles <- basename(Sys.glob(file.path(data_dir, "*.txt")))
sampleNames <- str_replace(sampleFiles, "[0-9]*-[0-9]*-[0-9]*", "") %>%
  str_replace("_analysis_results.txt", "")
sampleNames

## [1] "human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d28"
## [2] "human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d8"
## [3] "human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28"
## [4] "human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d8"
## [5] "human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28"
## [6] "human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d8"
```

```

output_dir <- "Venn diagrams"

read_in <- function(files) {
  d <- read.delim(files, header = TRUE)
  dd <- dplyr::select(d, log2FoldChange, padj, X) %>%
    na.omit() %>%
    dplyr::filter(padj <= 0.05) %>%
    dplyr::filter(abs(log2FoldChange) >= 0.5) %>%
    dplyr::select(X, log2FoldChange)
  dd
}

subset <- lapply(file.path(data_dir, sampleFiles), read_in)
names(subset) <- sampleNames

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

subset_up <- lapply(subset, function(x) dplyr::filter(x, log2FoldChange > 0)) %>%
  sapply(nrow)
subset_up

##   human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d28
##                                                    4
##   human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d8
##                                                    44
##   human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28
##                                                    0
##   human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d8
##                                                    162
##   human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28
##                                                    4
##   human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d8
##                                                    11

write.table(subset_up, file = file.path(output_dir, paste(Sys.Date(),
  "Venn genes upregulated.txt")), col.names = FALSE,
  row.names=TRUE, sep="\t", quote=FALSE)

subset_down <- lapply(subset, function(x) dplyr::filter(x, log2FoldChange < 0)) %>%
  sapply(nrow)
subset_down

##   human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d28
##                                                    0
##   human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d8
##                                                    50
##   human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28
##                                                    20
##   human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d8

```

```

##                               101
## human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28
##                               7
## human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d8
##                               50

write.table(subset_down, file = file.path(output_dir, paste(Sys.Date(),
  "Venn genes downregulated.txt")), col.names = FALSE,
  row.names=TRUE, sep="\t", quote=FALSE)

genes_only <- lapply(subset, function(x) dplyr::select(x, X))

##To keep track of the names of the different elements within the list
names(genes_only)

## [1] "human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d28"
## [2] "human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d8"
## [3] "human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28"
## [4] "human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d8"
## [5] "human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28"
## [6] "human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d8"

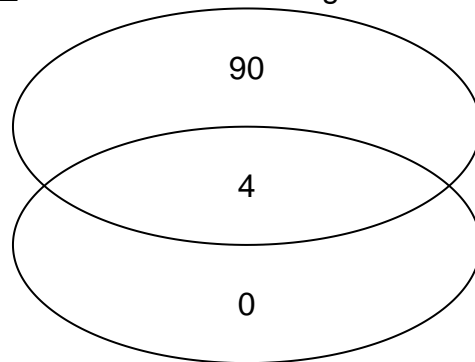
##For determining overlap areas, use gplots venn function

##HBV v mock
venn1 <- venn(genes_only[1:2])

## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List element
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List element
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List element
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List element
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List element
## Warning in getVennCounts.list(l, universe, intersections = TRUE, ...): List element [[2]] has dimens
## Warning in getVennCounts.list(l, universe, intersections = TRUE, ...): List element [[1]] has dimens
## Warning in getVennCounts.list(l, universe, intersections = TRUE, ...): List element [[1]] has dimens
## Warning in getVennCounts.list(l, universe, intersections = TRUE, ...): List element [[2]] has dimens
## Warning in getVennCounts.list(l, universe, intersections = TRUE, ...): List element [[1]] has dimens
## Warning in getVennCounts.list(l, universe, intersections = TRUE, ...): List element [[2]] has dimens

```

human\_donor\_treatmentHumanHBVgenes-HBV\_vs\_mock\_d8



human\_donor\_treatmentHumanHBVgenes-HBV\_vs\_mock\_d28

```
venn1_attr <- attr(venn1, "intersections")
##To see how many genes fall in each section of the Venn diagram
str(venn1_attr)

## List of 2
## $ human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d28:human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d28
## $ human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d8
venn1_attr

## $`human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d28:human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d28`
## [1] "AAB59969.1" "AAB59972.1" "AAB59971.1" "AAB59970.1"
##
## $`human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d8`
## [1] "ENSG000000187498" "ENSG000000137801" "ENSG000000087245"
## [4] "ENSG000000136872" "ENSG000000255974" "ENSG000000125810"
## [7] "ENSG000000011028" "ENSG000000166592" "ENSG000000261371"
## [10] "ENSG000000163975" "ENSG000000150938" "ENSG000000182704"
## [13] "ENSG000000198077" "ENSG000000100652" "ENSG000000169744"
## [16] "ENSG000000130508" "ENSG000000118523" "ENSG000000091879"
## [19] "ENSG000000163453" "ENSG000000110245" "ENSG000000100031"
## [22] "ENSG000000248144" "ENSG000000091136" "ENSG000000113140"
## [25] "ENSG000000105825" "ENSG000000145692" "ENSG000000198848"
## [28] "ENSG000000196569" "ENSG000000158104" "ENSG000000172831"
## [31] "ENSG000000116962" "ENSG000000162482" "ENSG000000143819"
## [34] "ENSG000000196628" "ENSG000000171724" "ENSG000000115884"
## [37] "ENSG000000256612" "ENSG000000160282" "ENSG000000166086"
## [40] "ENSG000000143845" "ENSG000000153936" "ENSG000000164125"
## [43] "ENSG000000072080" "ENSG000000243955" "ENSG000000135114"
## [46] "ENSG000000186908" "ENSG000000124875" "ENSG000000017427"
## [49] "ENSG000000118137" "ENSG000000188338" "ENSG000000175003"
## [52] "ENSG000000196616" "ENSG000000161940" "ENSG000000170927"
## [55] "ENSG000000164692" "ENSG000000175274" "ENSG000000196083"
## [58] "ENSG000000171747" "ENSG000000112902" "ENSG000000153822"
## [61] "ENSG000000184557" "ENSG000000087303" "ENSG000000138207"
## [64] "ENSG000000142798" "ENSG000000100889" "ENSG000000134871"
## [67] "ENSG000000131747" "ENSG000000257335" "ENSG000000134954"
## [70] "ENSG000000170921" "ENSG000000140945" "ENSG000000100003"
## [73] "ENSG000000064989" "ENSG000000170439" "ENSG000000154133"
## [76] "ENSG000000162460" "ENSG000000198899" "ENSG000000136856"
## [79] "ENSG000000135424" "ENSG000000143627" "ENSG000000118785"
```

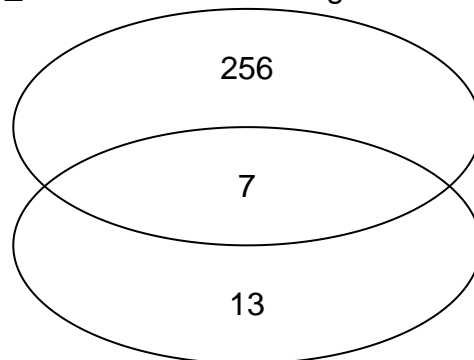
```
## [82] "ENSG00000129422" "ENSG00000143570" "ENSG00000143416"
## [85] "ENSG00000164161" "ENSG00000130300" "ENSG00000198763"
## [88] "ENSG00000151388" "ENSG00000196502" "ENSG00000171388"
```

```
d8_HBV <- venn1_attr$`human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d8`
```

```
##Coinf vs HBV
venn2 <- venn(genes_only[3:4])
```

```
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List element
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List element
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List element
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List element
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List element
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List element
## Warning in getVennCounts.list(1, universe, intersections = TRUE, ...): List element [[2]] has dimens
## Warning in getVennCounts.list(1, universe, intersections = TRUE, ...): List element [[1]] has dimens
## Warning in getVennCounts.list(1, universe, intersections = TRUE, ...): List element [[1]] has dimens
## Warning in getVennCounts.list(1, universe, intersections = TRUE, ...): List element [[2]] has dimens
## Warning in getVennCounts.list(1, universe, intersections = TRUE, ...): List element [[1]] has dimens
## Warning in getVennCounts.list(1, universe, intersections = TRUE, ...): List element [[2]] has dimens
```

human\_donor\_treatmentHumanHBVgenes-coinf\_vs\_HBV\_d8



human\_donor\_treatmentHumanHBVgenes-coinf\_vs\_HBV\_d28

```
venn2_attr <- attr(venn2, "intersections")
##To see how many genes fall in each section of the Venn diagram
str(venn2_attr)
```

```
## List of 3
## $ human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28:human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28
## $ human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d8
## $ human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28
```

```
venn2_attr
```

```
## $`human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28:human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28`
## [1] "ENSG00000261371" "ENSG00000128052" "ENSG00000154133" "ENSG00000163687"
## [5] "ENSG00000142748" "ENSG00000179776" "ENSG00000125810"
```

```

##
## $`human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d8`
## [1] "ENSG00000118785" "ENSG00000091879" "ENSG00000116785"
## [4] "ENSG00000164283" "ENSG00000145692" "ENSG00000038427"
## [7] "ENSG00000164188" "ENSG00000136235" "ENSG000000243955"
## [10] "ENSG00000011465" "ENSG00000039537" "ENSG00000177575"
## [13] "ENSG00000152377" "ENSG00000164176" "ENSG00000182704"
## [16] "ENSG00000010327" "ENSG00000111799" "ENSG00000134363"
## [19] "ENSG00000054654" "ENSG00000144820" "ENSG00000116690"
## [22] "ENSG00000108821" "ENSG00000102755" "ENSG00000139194"
## [25] "ENSG00000171747" "ENSG00000115461" "ENSG00000213694"
## [28] "ENSG00000205060" "ENSG00000066739" "ENSG00000144852"
## [31] "ENSG00000119946" "ENSG00000164692" "ENSG00000167588"
## [34] "ENSG00000109181" "ENSG00000165185" "ENSG00000168542"
## [37] "ENSG00000196878" "ENSG00000147650" "ENSG00000145321"
## [40] "ENSG00000158104" "ENSG00000166592" "ENSG00000119922"
## [43] "ENSG00000135378" "ENSG00000239887" "ENSG00000253368"
## [46] "ENSG00000130300" "ENSG00000205358" "ENSG00000058085"
## [49] "ENSG00000161011" "ENSG00000147027" "ENSG00000131018"
## [52] "ENSG00000096384" "ENSG00000110925" "ENSG00000101236"
## [55] "ENSG00000163825" "ENSG00000275764" "ENSG00000198743"
## [58] "ENSG00000184374" "ENSG00000129538" "ENSG00000006468"
## [61] "ENSG00000276980" "ENSG00000171345" "ENSG00000152455"
## [64] "ENSG00000134365" "ENSG00000141639" "ENSG00000100652"
## [67] "ENSG00000187955" "ENSG00000164300" "ENSG00000137504"
## [70] "ENSG00000244067" "ENSG00000106541" "ENSG00000137463"
## [73] "ENSG00000164120" "ENSG00000188707" "ENSG00000137561"
## [76] "ENSG00000139329" "ENSG00000271254" "ENSG00000169851"
## [79] "ENSG00000091831" "ENSG00000137054" "ENSG00000169903"
## [82] "ENSG00000110693" "ENSG00000152620" "ENSG00000206538"
## [85] "ENSG00000128294" "ENSG00000198650" "ENSG00000151388"
## [88] "ENSG00000100003" "ENSG00000183287" "ENSG00000160282"
## [91] "ENSG00000134853" "ENSG00000104635" "ENSG00000162460"
## [94] "ENSG00000198121" "ENSG00000174807" "ENSG00000182253"
## [97] "ENSG00000170153" "ENSG00000241119" "ENSG00000110422"
## [100] "ENSG00000069667" "ENSG00000213626" "ENSG00000112159"
## [103] "ENSG00000125966" "ENSG00000031003" "ENSG00000135111"
## [106] "ENSG00000165475" "ENSG00000196975" "ENSG00000187193"
## [109] "ENSG00000128607" "ENSG00000151790" "ENSG00000124145"
## [112] "ENSG00000253729" "ENSG00000006576" "ENSG00000255823"
## [115] "ENSG00000164038" "ENSG00000092010" "ENSG00000147324"
## [118] "ENSG00000152779" "ENSG00000169762" "ENSG00000111885"
## [121] "ENSG00000198740" "ENSG00000196323" "ENSG00000163171"
## [124] "ENSG00000025708" "ENSG00000149476" "ENSG00000173230"
## [127] "ENSG00000150782" "ENSG00000147526" "ENSG00000100197"
## [130] "ENSG00000159166" "ENSG00000261701" "ENSG00000166783"
## [133] "ENSG00000006459" "ENSG00000103449" "ENSG00000143416"
## [136] "ENSG00000134109" "ENSG00000135424" "ENSG00000092439"
## [139] "ENSG00000136872" "ENSG00000118946" "ENSG00000023330"
## [142] "ENSG00000144810" "ENSG00000118762" "ENSG00000169432"
## [145] "ENSG00000204262" "ENSG00000162366" "ENSG00000106819"
## [148] "ENSG00000147162" "ENSG00000196517" "ENSG00000213553"
## [151] "ENSG00000138315" "ENSG00000096070" "ENSG00000181555"
## [154] "ENSG00000066827" "ENSG00000135913" "ENSG00000101928"

```

```
## [157] "ENSG00000141441" "ENSG00000115884" "ENSG00000162896"
## [160] "ENSG00000157087" "ENSG00000244879" "ENSG00000144320"
## [163] "ENSG00000005812" "ENSG00000107890" "ENSG00000143845"
## [166] "ENSG00000275066" "ENSG00000129116" "ENSG00000069812"
## [169] "ENSG00000179761" "ENSG00000106605" "ENSG00000164825"
## [172] "ENSG00000158552" "ENSG00000135114" "ENSG00000178685"
## [175] "ENSG00000254416" "ENSG00000172831" "ENSG00000112977"
## [178] "ENSG00000165092" "ENSG00000142634" "ENSG00000166479"
## [181] "ENSG00000115464" "ENSG00000243137" "ENSG00000164850"
## [184] "ENSG00000064393" "ENSG00000165959" "ENSG00000119686"
## [187] "ENSG00000090273" "ENSG00000146072" "ENSG00000125144"
## [190] "ENSG00000144935" "ENSG00000127241" "ENSG00000135049"
## [193] "ENSG00000135048" "ENSG00000135069" "ENSG00000198734"
## [196] "ENSG00000178038" "ENSG00000186340" "ENSG00000050555"
## [199] "ENSG00000166391" "ENSG00000119917" "ENSG00000124214"
## [202] "ENSG00000134294" "ENSG00000171606" "ENSG00000127914"
## [205] "ENSG00000083642" "ENSG00000106804" "ENSG00000057252"
## [208] "ENSG00000078269" "ENSG00000173905" "ENSG00000187676"
## [211] "ENSG00000111371" "ENSG00000075151" "ENSG00000144199"
## [214] "ENSG00000100031" "ENSG00000137767" "ENSG00000177463"
## [217] "ENSG00000092964" "ENSG00000230590" "ENSG00000162409"
## [220] "ENSG00000085224" "ENSG00000204310" "ENSG00000241399"
## [223] "ENSG00000086619" "ENSG00000116882" "ENSG00000147421"
## [226] "ENSG00000188002" "ENSG00000177542" "ENSG00000144481"
## [229] "ENSG00000169905" "ENSG00000144357" "ENSG00000166197"
## [232] "ENSG00000021645" "ENSG00000105676" "ENSG00000102908"
## [235] "ENSG00000064651" "ENSG00000129003" "ENSG00000082438"
## [238] "ENSG00000196569" "ENSG00000185215" "ENSG00000175893"
## [241] "ENSG00000054118" "ENSG00000112186" "ENSG00000011405"
## [244] "ENSG00000138079" "ENSG00000198157" "ENSG00000165434"
## [247] "ENSG00000143627" "ENSG00000065413" "ENSG00000080200"
## [250] "ENSG00000126561" "ENSG00000197006" "ENSG00000103512"
## [253] "ENSG00000115484" "ENSG00000061987" "ENSG00000138246"
## [256] "ENSG00000166147"
##
## $`human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28`
## [1] "ENSG00000113555" "ENSG00000107719" "ENSG00000176435"
## [4] "ENSG00000164161" "ENSG00000110799" "ENSG00000133800"
## [7] "ENSG00000163762" "ENSG00000144668" "ENSG00000147113"
## [10] "ENSG00000171115" "ENSG00000169744" "ENSG00000120156"
## [13] "ENSG00000128510"
```

```
##Coinf vs mock
```

```
venn3 <- venn(genes_only[5:6])
```

```
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List elemen
```

```
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List elemen
```

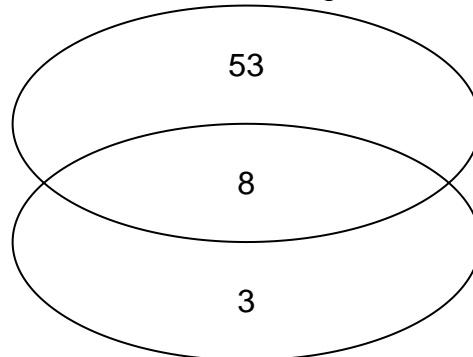
```
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List elemen
```

```
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List elemen
```

```
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List elemen
```

```
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List elemen
```

```
## Warning in getVennCounts.list(1, universe, intersections = TRUE, ...): List element [[2]] has dimens
## Warning in getVennCounts.list(1, universe, intersections = TRUE, ...): List element [[1]] has dimens
## Warning in getVennCounts.list(1, universe, intersections = TRUE, ...): List element [[1]] has dimens
## Warning in getVennCounts.list(1, universe, intersections = TRUE, ...): List element [[2]] has dimens
## Warning in getVennCounts.list(1, universe, intersections = TRUE, ...): List element [[1]] has dimens
## Warning in getVennCounts.list(1, universe, intersections = TRUE, ...): List element [[2]] has dimens
human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d8
```



human\_donor\_treatmentHumanHBVgenes-coinf\_vs\_mock\_d28

```
venn3_attr <- attr(venn3, "intersections")
##To see how many genes fall in each section of the Venn diagram
str(venn3_attr)
```

```
## List of 3
## $ human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28:human_donor_treatmentHumanHBVgenes-coinf_vs_m
## $ human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d8
## $ human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28
```

```
venn3_attr
```

```
## $`human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28:human_donor_treatmentHumanHBVgenes-coinf_vs_m
## [1] "AAB59969.1"      "AAB59972.1"      "AAB59971.1"      "AAB59970.1"
## [5] "ENSG00000261371" "ENSG00000163687" "ENSG00000179776" "ENSG00000107719"
##
## $`human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d8`
## [1] "ENSG00000091879" "ENSG00000118785" "ENSG00000136235"
## [4] "ENSG00000164283" "ENSG00000142748" "ENSG00000177575"
## [7] "ENSG00000125810" "ENSG00000129538" "ENSG00000064989"
## [10] "ENSG00000102755" "ENSG00000128052" "ENSG00000130300"
## [13] "ENSG00000155659" "ENSG00000158352" "ENSG00000154133"
## [16] "ENSG00000122254" "ENSG00000102760" "ENSG00000118946"
## [19] "ENSG00000175538" "ENSG00000260314" "ENSG00000152779"
## [22] "ENSG00000113555" "ENSG00000204301" "ENSG00000161940"
## [25] "ENSG00000086548" "ENSG00000196611" "ENSG00000134121"
## [28] "ENSG00000165072" "ENSG00000162618" "ENSG00000129467"
## [31] "ENSG00000023171" "ENSG00000123700" "ENSG00000105825"
## [34] "ENSG00000139567" "ENSG00000134363" "ENSG00000127329"
## [37] "ENSG00000125703" "ENSG00000171115" "ENSG00000104938"
## [40] "ENSG00000182795" "ENSG00000088836" "ENSG00000204385"
## [43] "ENSG00000146038" "ENSG00000198650" "ENSG00000213626"
```



```

## [46] "ENSG00000188707" "ENSG00000157014" "ENSG00000203883"
## [49] "ENSG00000114279" "ENSG00000104967" "ENSG00000239887"
## [52] "ENSG00000163935" "ENSG00000090382"
##
## $`human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28`
## [1] "ENSG0000010327" "ENSG00000176435" "ENSG00000159189"

d8_coinf <- venn3_attr$`human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28`

##Look for overlap between d8 coinf and d8 mono-infected genes.
Reduce(intersect, list(d8_HBV, d8_coinf))

## [1] "ENSG00000125810" "ENSG00000091879" "ENSG00000105825" "ENSG00000161940"
## [5] "ENSG00000064989" "ENSG00000154133" "ENSG00000118785" "ENSG00000130300"

##The comparison between coinf v HBV d8:d28 and the comparison between coinf v mock
##d8:d28 both had overlapping genes. Besides the HBV genes, which we know have expression
##all > 1, I want to compare the expression of these common genes compare
##(i.e. upregulated versus downregulated at each timepoint).

venn2_intersect <- as.data.frame(venn2_attr[1], stringsAsFactors = FALSE)
colnames(venn2_intersect) <- c("X")

venn3_intersect <- as.data.frame(venn3_attr[1], stringsAsFactors = FALSE)
colnames(venn3_intersect) <- c("X")

coinfvHBVd28_df <- as.data.frame(subset[3]) %>%
  droplevels()
colnames(coinfvHBVd28_df) <- c("X", "d28log2FoldChange")
coinfvHBVd8_df <- as.data.frame(subset[4]) %>%
  droplevels()
colnames(coinfvHBVd8_df) <- c("X", "d8log2FoldChange")
venn2_intersect_l2FC <- inner_join(venn2_intersect, coinfvHBVd8_df, by = "X") %>%
  inner_join(coinfvHBVd28_df, by = "X")

## Warning: Column `X` joining character vector and factor, coercing into
## character vector

## Warning: Column `X` joining character vector and factor, coercing into
## character vector

write.csv(venn2_intersect_l2FC, file = file.path(output_dir, paste(Sys.Date(),
  filename = "coinfvHBV_d8_d28_intersection.csv")))

coinfvmockd28_df <- as.data.frame(subset[5]) %>%
  droplevels()
colnames(coinfvmockd28_df) <- c("X", "d28log2FoldChange")
coinfvmockd8_df <- as.data.frame(subset[6]) %>%
  droplevels()
colnames(coinfvmockd8_df) <- c("X", "d8log2FoldChange")
venn3_intersect_l2FC <- inner_join(venn3_intersect, coinfvmockd8_df, by = "X") %>%
  inner_join(coinfvmockd28_df, by = "X")

## Warning: Column `X` joining character vector and factor, coercing into
## character vector

## Warning: Column `X` joining character vector and factor, coercing into

```

```
## character vector
```

```
write.csv(venn3_intersect_l2FC, file = file.path(output_dir, paste(Sys.Date(),  
  filename = "coinfvmock_d8_d28_intersection.csv")))
```

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] grid      stats      graphics  grDevices  utils      datasets  methods
## [8] base
##
## other attached packages:
## [1] bindrcpp_0.2.2      extrafont_0.17      VennDiagram_1.6.20
## [4] futile.logger_1.4.3 gplots_3.0.1        stringr_1.3.1
## [7] tibble_1.4.2        dplyr_0.7.6
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.18        pillar_1.3.0        formatR_1.5
## [4] bindr_0.1.1         bitops_1.0-6        futile.options_1.0.1
## [7] tools_3.3.3         digest_0.6.15       evaluate_0.11
## [10] pkgconfig_2.0.1     rlang_0.2.1         rstudioapi_0.7
## [13] yaml_2.2.0          Rttf2pt1_1.3.7      knitr_1.20
## [16] gtools_3.8.1        caTools_1.17.1.1    rprojroot_1.3-2
## [19] tidyselect_0.2.4    glue_1.3.0          R6_2.2.2
## [22] rmarkdown_1.10      gdata_2.18.0        extrafontdb_1.0
## [25] purrr_0.2.5         lambda.r_1.2.3       magrittr_1.5
## [28] backports_1.1.2     htmltools_0.3.6     assertthat_0.2.0
## [31] KernSmooth_2.23-15  stringi_1.2.4       crayon_1.3.4
```