Venn diagrams

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"
```

```r
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) %>%
    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
##                                                   61
##  human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28
##                                                    0
##   human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d8
##                                                   67
## human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28
##                                                    4
##  human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d8
##                                                   53
```

```r
write.table(subset_up, file = "Venn diagrams/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
##                                                    1
##    human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d8
##                                                  102
##  human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28
##                                                   10
##   human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d8
##                                                   58
## human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28
##                                                   18
##  human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d8
##                                                     95
```

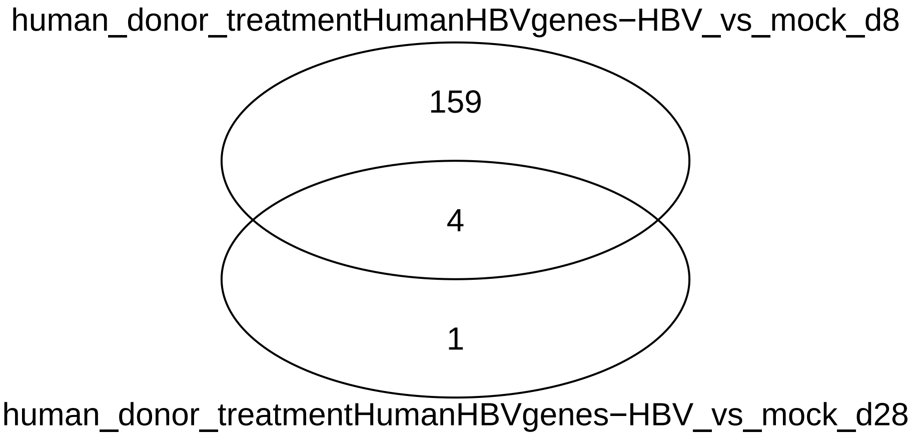
```
write.table(subset_down, file = "Venn diagrams/Venn genes downregulated.txt",
            col.names = FALSE, row.names=TRUE,sep="\t",quote=FALSE)

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
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(data, universe = universe, intersections = intersections): List element

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

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

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

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

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

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

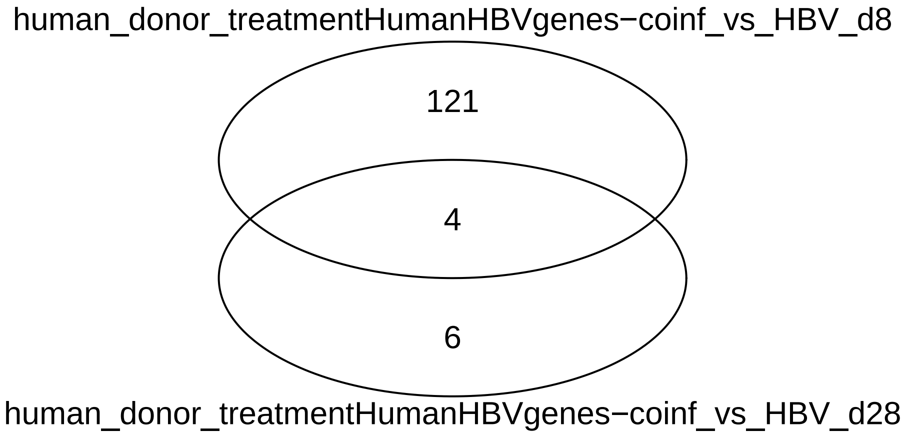


```
attr(venn1, "intersections")
```

```
## $`human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d28:human_donor_treatmentHumanHBVgenes-HBV_vs_mock_
## [1] "AAB59969.1" "AAB59972.1" "AAB59971.1" "AAB59970.1"
##
## $`human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d8`
##   [1] "ENSG00000130508" "ENSG00000179776" "ENSG00000187498"
##   [4] "ENSG00000261371" "ENSG00000125810" "ENSG00000136872"
##   [7] "ENSG00000138207" "ENSG00000087303" "ENSG00000091879"
##  [10] "ENSG00000110245" "ENSG00000087245" "ENSG00000134871"
##  [13] "ENSG00000118523" "ENSG00000158104" "ENSG00000134013"
##  [16] "ENSG00000255974" "ENSG00000142798" "ENSG00000100652"
##  [19] "ENSG00000163975" "ENSG00000150938" "ENSG00000143819"
##  [22] "ENSG00000011028" "ENSG00000196616" "ENSG00000163453"
##  [25] "ENSG00000154133" "ENSG00000169744" "ENSG00000164692"
##  [28] "ENSG00000124762" "ENSG00000165092" "ENSG00000118137"
##  [31] "ENSG00000198848" "ENSG00000106991" "ENSG00000196569"
##  [34] "ENSG00000172831" "ENSG00000166592" "ENSG00000248144"
##  [37] "ENSG00000204301" "ENSG00000151790" "ENSG00000108821"
##  [40] "ENSG00000244067" "ENSG00000196502" "ENSG00000151365"
##  [43] "ENSG00000161638" "ENSG00000116962" "ENSG00000145623"
##  [46] "ENSG00000118257" "ENSG00000175003" "ENSG00000166086"
##  [49] "ENSG00000140945" "ENSG00000100031" "ENSG00000175899"
##  [52] "ENSG00000244474" "ENSG00000116661" "ENSG00000091136"
##  [55] "ENSG00000010327" "ENSG00000186642" "ENSG00000137801"
##  [58] "ENSG00000196628" "ENSG00000130300" "ENSG00000161940"
##  [61] "ENSG00000160282" "ENSG00000164125" "ENSG00000171724"
##  [64] "ENSG00000175274" "ENSG00000113140" "ENSG00000184557"
##  [67] "ENSG00000256612" "ENSG00000214548" "ENSG00000171747"
##  [70] "ENSG00000257335" "ENSG00000164161" "ENSG00000021826"
##  [73] "ENSG00000004399" "ENSG00000143772" "ENSG00000017427"
##  [76] "ENSG00000134954" "ENSG00000164283" "ENSG00000143845"
##  [79] "ENSG00000166341" "ENSG00000188338" "ENSG00000064989"
##  [82] "ENSG00000115380" "ENSG00000100889" "ENSG00000124875"
##  [85] "ENSG00000149591" "ENSG00000162460" "ENSG00000131747"
##  [88] "ENSG00000156515" "ENSG00000145321" "ENSG00000162482"
##  [91] "ENSG00000196600" "ENSG00000270504" "ENSG00000153936"
##  [94] "ENSG00000169738" "ENSG00000143627" "ENSG00000157510"
##  [97] "ENSG00000112902" "ENSG00000147862" "ENSG00000142910"
## [100] "ENSG00000182704" "ENSG00000196083" "ENSG00000141756"
## [103] "ENSG00000187193" "ENSG00000145555" "ENSG00000151388"
## [106] "ENSG00000125144" "ENSG00000143416" "ENSG00000066056"
## [109] "ENSG00000100003" "ENSG00000153822" "ENSG00000106804"
## [112] "ENSG00000071246" "ENSG00000171388" "ENSG00000170439"
## [115] "ENSG00000087116" "ENSG00000134817" "ENSG00000118785"
## [118] "ENSG00000128052" "ENSG00000158125" "ENSG00000198077"
## [121] "ENSG00000164266" "ENSG00000072080" "ENSG00000132693"
## [124] "ENSG00000026025" "ENSG00000177556" "ENSG00000189334"
## [127] "ENSG00000169242" "ENSG00000135424" "ENSG00000148803"
## [130] "ENSG00000129422" "ENSG00000133048" "ENSG00000140505"
## [133] "ENSG00000186910" "ENSG00000144908" "ENSG00000162618"
## [136] "ENSG00000142173" "ENSG00000168306" "ENSG00000135114"
## [139] "ENSG00000147113" "ENSG00000198099" "ENSG00000129657"
## [142] "ENSG00000035862" "ENSG00000166391" "ENSG00000111348"
## [145] "ENSG00000105825" "ENSG00000170921" "ENSG00000136856"
## [148] "ENSG00000239887" "ENSG00000113600" "ENSG00000120899"
## [151] "ENSG00000123243" "ENSG00000112419" "ENSG00000260314"
## [154] "ENSG00000171603" "ENSG00000186908" "ENSG00000131495"
## [157] "ENSG00000110799" "ENSG00000129538" "ENSG00000107957"
##
## $`human_donor_treatmentHumanHBVgenes-HBV_vs_mock_d28`
## [1] "ENSG00000126838"
```

```
##Four in common (all HBV); otherwise d8 has 159; d28 has 1.

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(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−coinf_vs_HBV_d8
121
4
6
human_donor_treatmentHumanHBVgenes−coinf_vs_HBV_d28

```
attr(venn2, "intersections")
```

```
## $`human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28:human_donor_treatmentHumanHBVgenes-coinf_vs_HB
## [1] "ENSG00000130300" "ENSG00000128052" "ENSG00000261371" "ENSG00000154133"
##
## $`human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d8`
##   [1] "ENSG00000011465" "ENSG00000168542" "ENSG00000118785"
##   [4] "ENSG00000142748" "ENSG00000091879" "ENSG00000125810"
##   [7] "ENSG00000102755" "ENSG00000196569" "ENSG00000145321"
##  [10] "ENSG00000010327" "ENSG00000054654" "ENSG00000039537"
##  [13] "ENSG00000108821" "ENSG00000164692" "ENSG00000162618"
##  [16] "ENSG00000166391" "ENSG00000152377" "ENSG00000151388"
##  [19] "ENSG00000118946" "ENSG00000164188" "ENSG00000142156"
##  [22] "ENSG00000188257" "ENSG00000125144" "ENSG00000164283"
##  [25] "ENSG00000165092" "ENSG00000146072" "ENSG00000164850"
##  [28] "ENSG00000129538" "ENSG00000132693" "ENSG00000163687"
##  [31] "ENSG00000125966" "ENSG00000177575" "ENSG00000138315"
##  [34] "ENSG00000058085" "ENSG00000164120" "ENSG00000124839"
##  [37] "ENSG00000179776" "ENSG00000090238" "ENSG00000184374"
##  [40] "ENSG00000095303" "ENSG00000136235" "ENSG00000038427"
##  [43] "ENSG00000116690" "ENSG00000166592" "ENSG00000140450"
##  [46] "ENSG00000106351" "ENSG00000147027" "ENSG00000134363"
##  [49] "ENSG00000087245" "ENSG00000119946" "ENSG00000134853"
##  [52] "ENSG00000119922" "ENSG00000168631" "ENSG00000163581"
##  [55] "ENSG00000186340" "ENSG00000184557" "ENSG00000124145"
##  [58] "ENSG00000011028" "ENSG00000115232" "ENSG00000110925"
##  [61] "ENSG00000161011" "ENSG00000111799" "ENSG00000142173"
##  [64] "ENSG00000253368" "ENSG00000140416" "ENSG00000228278"
##  [67] "ENSG00000187955" "ENSG00000087116" "ENSG00000136881"
##  [70] "ENSG00000104635" "ENSG00000198840" "ENSG00000276980"
##  [73] "ENSG00000239887" "ENSG00000144481" "ENSG00000156968"
##  [76] "ENSG00000125845" "ENSG00000205060" "ENSG00000244414"
##  [79] "ENSG00000163050" "ENSG00000160285" "ENSG00000102760"
##  [82] "ENSG00000106327" "ENSG00000164266" "ENSG00000137054"
##  [85] "ENSG00000025423" "ENSG00000115380" "ENSG00000244067"
##  [88] "ENSG00000111181" "ENSG00000198121" "ENSG00000152779"
##  [91] "ENSG00000032444" "ENSG00000135069" "ENSG00000165169"
##  [94] "ENSG00000110693" "ENSG00000139194" "ENSG00000188338"
##  [97] "ENSG00000162460" "ENSG00000254166" "ENSG00000165475"
## [100] "ENSG00000136720" "ENSG00000175538" "ENSG00000066739"
## [103] "ENSG00000254607" "ENSG00000198610" "ENSG00000188488"
## [106] "ENSG00000196975" "ENSG00000076351" "ENSG00000123384"
## [109] "ENSG00000188707" "ENSG00000127954" "ENSG00000115884"
## [112] "ENSG00000170458" "ENSG00000144852" "ENSG00000214548"
## [115] "ENSG00000110799" "ENSG00000091436" "ENSG00000135744"
## [118] "ENSG00000204262" "ENSG00000196878" "ENSG00000134871"
## [121] "ENSG00000160282"
##
## $`human_donor_treatmentHumanHBVgenes-coinf_vs_HBV_d28`
## [1] "ENSG00000113555" "ENSG00000107719" "ENSG00000164161" "ENSG00000176435"
## [5] "ENSG00000163762" "ENSG00000133574"
```

```
##Four in common; d8 has 121, d28 has 6

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

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

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

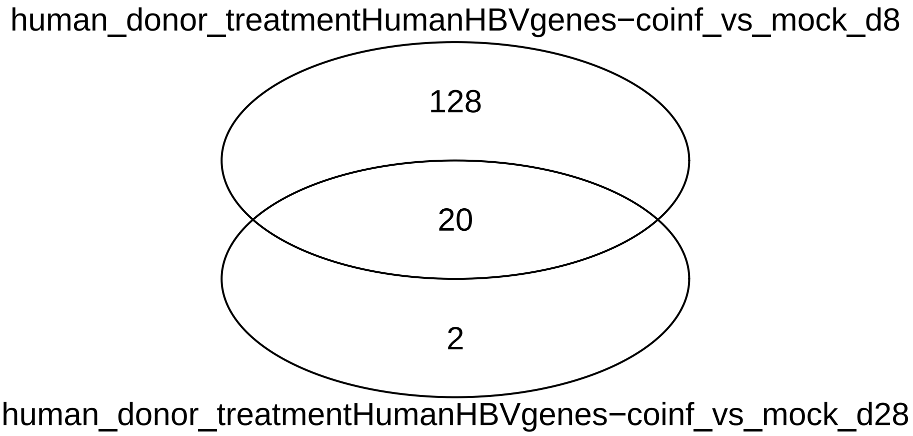
```
## Warning in getVennCounts.list(data, universe = universe, intersections = intersections): List element
```

```
## 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(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−coinf_vs_mock_d8
128
20
2
human_donor_treatmentHumanHBVgenes−coinf_vs_mock_d28

```r
attr(venn3, "intersections")
```

```
## $`human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28:human_donor_treatmentHumanHBVgenes-coinf_vs_mo
##  [1] "AAB59970.1"      "AAB59969.1"      "AAB59971.1"
##  [4] "AAB59972.1"      "ENSG00000136235" "ENSG00000261371"
##  [7] "ENSG00000118785" "ENSG00000091879" "ENSG00000069122"
## [10] "ENSG00000010327" "ENSG00000163687" "ENSG00000204301"
## [13] "ENSG00000260314" "ENSG00000176435" "ENSG00000124126"
## [16] "ENSG00000133574" "ENSG00000118946" "ENSG00000135218"
## [19] "ENSG00000164161" "ENSG00000169744"
##
## $`human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d8`
##   [1] "ENSG00000125810" "ENSG00000179776" "ENSG00000154133"
##   [4] "ENSG00000110799" "ENSG00000142748" "ENSG00000162618"
##   [7] "ENSG00000177575" "ENSG00000102755" "ENSG00000129538"
##  [10] "ENSG00000164283" "ENSG00000128052" "ENSG00000146038"
##  [13] "ENSG00000130300" "ENSG00000064989" "ENSG00000127399"
##  [16] "ENSG00000127329" "ENSG00000122254" "ENSG00000118271"
##  [19] "ENSG00000188488" "ENSG00000161267" "ENSG00000091583"
##  [22] "ENSG00000102760" "ENSG00000106351" "ENSG00000175538"
##  [25] "ENSG00000123700" "ENSG00000119922" "ENSG00000105825"
##  [28] "ENSG00000038945" "ENSG00000155659" "ENSG00000139567"
##  [31] "ENSG00000113555" "ENSG00000091436" "ENSG00000196628"
##  [34] "ENSG00000152779" "ENSG00000198408" "ENSG00000111181"
##  [37] "ENSG00000108582" "ENSG00000170525" "ENSG00000089220"
##  [40] "ENSG00000086548" "ENSG00000054654" "ENSG00000066056"
##  [43] "ENSG00000163050" "ENSG00000147113" "ENSG00000214530"
##  [46] "ENSG00000001617" "ENSG00000118257" "ENSG00000157554"
##  [49] "ENSG00000138722" "ENSG00000128917" "ENSG00000160862"
##  [52] "ENSG00000173269" "ENSG00000265972" "ENSG00000123243"
##  [55] "ENSG00000110169" "ENSG00000124839" "ENSG00000167191"
##  [58] "ENSG00000135744" "ENSG00000120885" "ENSG00000162511"
##  [61] "ENSG00000197629" "ENSG00000058085" "ENSG00000138207"
##  [64] "ENSG00000151790" "ENSG00000171345" "ENSG00000137801"
##  [67] "ENSG00000116690" "ENSG00000145321" "ENSG00000101445"
##  [70] "ENSG00000165124" "ENSG00000196975" "ENSG00000171115"
##  [73] "ENSG00000161940" "ENSG00000153162" "ENSG00000166173"
##  [76] "ENSG00000117601" "ENSG00000125734" "ENSG00000176046"
##  [79] "ENSG00000138079" "ENSG00000121858" "ENSG00000170315"
##  [82] "ENSG00000144481" "ENSG00000124882" "ENSG00000231574"
##  [85] "ENSG00000125968" "ENSG00000158352" "ENSG00000106927"
##  [88] "ENSG00000188707" "ENSG00000184374" "ENSG00000171793"
##  [91] "ENSG00000142168" "ENSG00000145012" "ENSG00000138315"
##  [94] "ENSG00000071246" "ENSG00000073849" "ENSG00000233276"
##  [97] "ENSG00000185950" "ENSG00000143418" "ENSG00000182551"
## [100] "ENSG00000105939" "ENSG00000156096" "ENSG00000132821"
## [103] "ENSG00000213694" "ENSG00000104635" "ENSG00000107719"
## [106] "ENSG00000163735" "ENSG00000170458" "ENSG00000106541"
## [109] "ENSG00000105835" "ENSG00000124145" "ENSG00000125703"
## [112] "ENSG00000164543" "ENSG00000198712" "ENSG00000037280"
## [115] "ENSG00000196611" "ENSG00000173372" "ENSG00000168631"
## [118] "ENSG00000104938" "ENSG00000106991" "ENSG00000153048"
## [121] "ENSG00000211459" "ENSG00000116661" "ENSG00000169903"
## [124] "ENSG00000135678" "ENSG00000131386" "ENSG00000151726"
## [127] "ENSG00000155368" "ENSG00000165092"
##
## $`human_donor_treatmentHumanHBVgenes-coinf_vs_mock_d28`
## [1] "ENSG00000126838" "ENSG00000144668"
```

```r
venn3_attr <- attr(venn3, "intersections")
## In common 20 total including 4 HBV genes; otherwise d8 has 128; d28 has 2.

##As the only comparison with overlapping genes besides the HBV genes (which we
##know all are > 1), I want to see how the expression of these common genes compare
##(i.e. upregulated versus downregulated at each timepoint).
venn3_intersect <- as.data.frame(venn3_attr[1], stringsAsFactors = FALSE)
colnames(venn3_intersect) <- c("X")

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
```

```r
write.csv(venn3_intersect_l2FC, file = file.path("Venn diagrams",
  filename =  "coinfvmock_d8_d28_intersection.csv"))
```

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] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] bindrcpp_0.2        extrafont_0.17      VennDiagram_1.6.17
## [4] futile.logger_1.4.3 gplots_3.0.1        stringr_1.2.0
## [7] tibble_1.3.3        dplyr_0.7.3
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.10         Rttf2pt1_1.3.5       knitr_1.16
##  [4] bindr_0.1            magrittr_1.5         R6_2.2.0
##  [7] rlang_0.1.2          caTools_1.17.1       tools_3.3.3
## [10] KernSmooth_2.23-15   extrafontdb_1.0      lambda.r_1.1.9
## [13] htmltools_0.3.5      gtools_3.5.0         yaml_2.1.14
## [16] assertthat_0.2.0     rprojroot_1.2        digest_0.6.12
## [19] futile.options_1.0.0 bitops_1.0-6         glue_1.1.1
## [22] evaluate_0.10        rmarkdown_1.4        gdata_2.17.0
## [25] stringi_1.1.5        backports_1.0.5      pkgconfig_2.0.1
```