Analysing Mass Spec Data

This is an R Markdown document to show how I have analysed my mass spectroscopy data. Author: Catherine Truman Date: 25/10/2019

First, we: (1) import the libraries we need (2) set up our directory (replace this text with the directories and file paths of your files) (3) read in our data and databases we will reference later on

Note, we change the names of columns - make sure the right names are called for your database

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------------

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ---------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(dplyr)
library(readr)
library(ggalluvial)
library(ggrepel)
#BiocManager::install("hpar")
library(hpar)
```

```
## This is hpar version 1.26.0,
## based on the Human Protein Atlas
##   Version: 18.1
##   Release data: 2018.11.15
##   Ensembl build: 88.38
## See '?hpar' or 'vignette('hpar')' for details.
```

```
require(knitr)
```

```
## Loading required package: knitr
```

```
WD <- paste("C:/Users/Greye/Dropbox/DPHIL PHD UPDATED/DATA/MASS SPECTRONOMY/REPLICATES/",
            "USING REP 1 AND 2", sep="")
knitr::opts_knit$set(root.dir = normalizePath(WD)) # change this to show your own working
#directory

NCBI_DB <- read.csv(file.path(WD, "NCBI_HIV_INT_DB.csv"), stringsAsFactors = F)
# NCBI HIV database
MS_Data <- read.csv(file.path(WD, "RawData.csv"), stringsAsFactors = F, header = T)
# MS database
MS_Data <- MS_Data %>% rename("AdjPValue" = "adj.P.Val", "Significance" = "sig",
                              "LogFC" = "logFC", "GeneNames" = "Row.names")
# Give common column names
```

Let's count how many hits we have of significance and summarise in a table

```
SigHits <- (MS_Data$GeneNames[MS_Data$Significance  == "+"])
NoSigHits <- length(SigHits)

DoubleSigHits <- (MS_Data$GeneNames[MS_Data$Significance == "++"])
NoDoubleSigHits <- length(DoubleSigHits)

OtherHits <- (MS_Data$GeneNames[MS_Data$Significance == ""| MS_Data$Significance == "-" ])
NoOtherHits <- length(OtherHits)

Total = NoSigHits + NoDoubleSigHits + NoOtherHits

Summary <- cbind(NoOtherHits, NoSigHits, NoDoubleSigHits, Total)
colnames(Summary)[1:4] <- c('Other', '+','++', 'Total')
Summary
```

```
##      Other   +  ++ Total
## [1,]  1096 225 100  1421
```

Now let's make a volcano plot. Let's apply a log10 p-value to suit the scale. We also need to feed the function our variables. Replace these with your viral genes, IP protein and conditions of interest. (!) Make sure your protein names match what is in the MS table exactly

```
MS_Data$Difference <- -log10(MS_Data$AdjPValue)

Viral_Proteins <- c('MATRIX-p17', 'P6-GAG', 'CA-p24', 'P2', 'NC-p17', 'RT-p51',
                    'Rnase-p15', 'Integrase-P31', 'VIF', 'VPR', 'TAT', 'mCherry-T2A')
Bait_Protein <- c('REV-FLAG-3xMyc') # Replace these with your protein names
Positive <- 'REV-FLAG-3xMyc'
Negative <- 'mCherry-Nef'
```

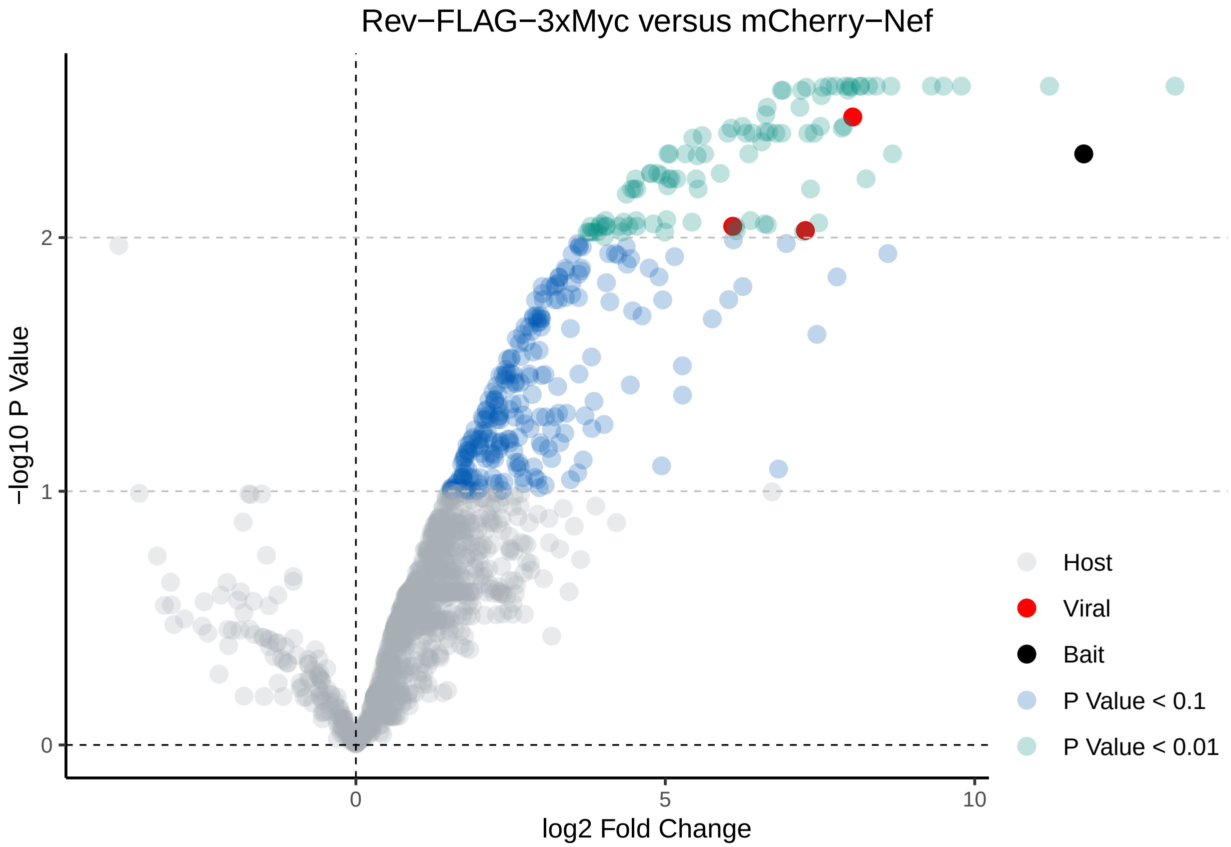
Next, we make a function to produce a volcano plot. Study it carefully. You give it your MS_Data, list of proteins and conditions and it colours points based on sigificance. You can comment out or in (using hash) the things you want labelled

```
MakeVPTable <- function(Database, Viral_Proteins, Bait_Proteins, Positive, Negative){

  ### COLOURING
  Database$"Colouring" <- "" # In this column, we'll add labels to points we want coloured
  Database$Colouring[Database$GeneNames %in% DoubleSigHits] <- 'P Value < 0.01'
  Database$Colouring[Database$GeneNames %in% SigHits] <- 'P Value < 0.1'
  Database$Colouring[Database$GeneNames %in% OtherHits] <- 'Host'
  Database$Colouring[Database$GeneNames %in% Viral_Proteins] <- 'Viral'
  Database$Colouring[Database$GeneNames %in% Bait_Protein] <- 'Bait'
  Point_Colours <- c('#A7AFB540',"#FF0000", "#000000", "#005AB540", "#008E8040")
  Database$Colouring <- factor(Database$Colouring, levels=c('Host',
                                                            'Viral', 'Bait',
                                                            'P Value < 0.1', 'P Value < 0.01'))

  ### LABELLING
  Database$"Labelling" <- "" # In this column, we add labels to points we want labelled

  # Label P < 0.01, viral and IP proteins
 # Database$Labelling[Database$GeneNames %in% DoubleSigHits] <-
  #Database$GeneNames[Database$GeneNames %in% DoubleSigHits]
#  Database$Labelling[Database$GeneNames %in% Viral_Proteins] <-
```

```
  #Database$GeneNames[Database$GeneNames   %in% Viral_Proteins]
#  Database$Labelling[Database$GeneNames %in% Bait_Protein] <-
  #Database$GeneNames[Database$GeneNames     %in% Bait_Protein]

  # Label P < 0.1
# Database$Labelling[Database$GeneNames %in% SigHits] <-
  #Database$GeneNames[Database$GeneNames %in%     SigHits]

  # Remove RPS/RPL proteins
 # Database$Labelling[grep("RPL|RPS", Database$Labelling)] <- ""

  # Remove viral proteins
  Matches <<- paste(unique(grep(paste(Viral_Proteins,collapse="\\b|\\b"),
                                 Database$Labelling, value=TRUE, ignore.case=FALSE)),
                    collapse="|")
#  Database$Labelling <- ifelse(grepl(Matches, Database$Labelling), '', Database$Labelling)

  # Label only viral proteins
 Database$Labelling <- ifelse(grepl(Matches, Database$Labelling), Database$Labelling, '')

  # Label only RPS/RPL proteins
 #Database$Labelling <- ifelse(grepl("RPL|RPS", Database$Labelling), Database$Labelling, '')

  Database <<- Database

  VP = ggplot(Database, aes(LogFC, Difference, color = Database$Colouring)) +
    geom_point(shape = 16, size = 3, show.legend = TRUE) + theme_classic() +
    theme(text = element_text(size = 10)) + scale_colour_manual(values = Point_Colours) +
    ggrepel::geom_text_repel(label=Database$Labelling, size = 3,
                              box.padding = unit(0.1, "lines"),
    point.padding = unit(0.1, "lines"), segment.size = 0.5,
    segment.color="black", colour="black") +
    geom_hline( yintercept=0, linetype="dashed", size=0.3, colour="black") +
    geom_vline( xintercept=0, linetype="dashed", size=0.3, colour="black") +
    ggtitle(paste0(Positive, " versus ", Negative)) +
    theme(plot.title = element_text(hjust=0.5)) +
    xlab("log2 Fold Change") + ylab("-log10 P Value") +
    geom_hline( yintercept=1, linetype="dashed", size=0.3, colour="grey") +
    geom_hline( yintercept=2, linetype="dashed", size=0.3, colour="grey") +
    theme(legend.position = c(0.9, 0.18)) +    theme(legend.title = element_blank()) +
    theme(legend.text=element_text(size=9))


return(VP)
}
```

Finally, let's run the function with our parameters. We should get back a plot.

```
MakeVPTable(MS_Data, Viral_Proteins, Bait_Proteins, "Rev-FLAG-3xMyc", "mCherry-Nef")
```



What the code originally made was a little.. overcrowded. You might notice some lines of code which are modifiable. These can be commented out using a hashtag, meaning they will not be processed. This can allow you to label more specifically.

Finally, let's save.

```
ggsave('Volcano_P001Cellular_NoLabel.pdf', plot=last_plot(), path = WD, dpi=700)
```

```
## Saving 6.5 x 4.5 in image
```

Now we have a lot of hits, but we need a candidate list. How do we select? We have three parameters: -P Value -Fold Change -Rev intensity

Let's remind ourselves what our data looks like at a glance.

```
Summary
```

```
##      Other   +  ++ Total
## [1,]  1096 225 100  1421
```

Looks like we have 100 proteins identified at a P Value of <0.01. In my case, lots of these are ribosomal. We don't care about those. Let's segment our list by P Value < 0.01 and remove ribosomal hits. Then let's calculate the average Rev intensity. if you have a different number of replicates, change the code! Finally, we select only the columns we're interested in and sort by intensity.

```
MS_DataP001 <- MS_Data[MS_Data$Significance == '++', ]
MS_DataP001 <- MS_DataP001[!grepl("RPL|RPS", MS_DataP001$GeneNames), ]
MS_DataP001 <- MS_DataP001[!grepl(Positive, MS_DataP001$GeneNames), ]
MS_DataSig <- MS_Data[MS_Data$Significance == '+' | MS_Data$Significance == '++' , ]
#MS_DataSig <- MS_DataSig[!grepl("RPL|RPS", MS_DataSig$GeneNames), ]
```

```
AveRevInt <- data.frame(Means=rowMeans(MS_DataP001[,5:6]))
MS_DataP001 <- cbind(MS_DataP001, AveRevInt$Means)
colnames(MS_DataP001)[13] <- 'Means'

MS_DataP001 <- MS_DataP001 %>% select('GeneNames', 'LogFC',
                                      'AdjPValue', 'Significance', 'Means')
Hits <- MS_DataP001 %>% arrange(desc(Means)) %>% top_n(40) %>% filter(LogFC > 4)
```

```
## Selecting by Means
```

```
Hits
```

```
##      GeneNames     LogFC   AdjPValue Significance    Means
## 1        ELMO1 13.237358 0.002531002           ++ 33.98500
## 2       ERGIC3 11.207816 0.002531002           ++ 31.95546
## 3          NCL  5.040164 0.004679567           ++ 31.20564
## 4    HNRNPA2B1  4.410919 0.009027623           ++ 31.08732
## 5      HNRNPA1  4.496173 0.006436298           ++ 30.97058
## 6         NPM1  4.300902 0.009510799           ++ 30.57319
## 7        SRP72  9.784672 0.002531002           ++ 30.53232
## 8       PABPC1  4.249331 0.009027623           ++ 30.04502
## 9   MATRIX-p17  7.263803 0.009383212           ++ 29.17770
## 10       SRP68  7.883068 0.003646752           ++ 29.15195
## 11     SYNCRIP  6.349835 0.004679567           ++ 28.93985
## 12      CA-p24  8.028737 0.003347656           ++ 28.77638
## 13        DHX9  6.302005 0.003879966           ++ 28.76236
## 14     HNRNPA3  7.908034 0.002531002           ++ 28.65568
## 15        RBMX  4.538991 0.009027623           ++ 28.49899
## 16      P6-GAG  6.089904 0.009027623           ++ 27.61885
## 17         SSB  4.520697 0.005871785           ++ 27.60243
## 18       SRP19  6.787682 0.003879966           ++ 27.53533
## 19      HNRNPR  6.407519 0.003879966           ++ 27.45876
## 20       MITD1  6.646146 0.003064925           ++ 27.39379
## 21       MTCL1  6.615494 0.003835257           ++ 27.36314
## 22      CHCHD1  6.064054 0.003706303           ++ 26.81170
## 23    HIST1H1C  6.006282 0.003879966           ++ 26.75393
## 24       LARP7  5.530760 0.006436298           ++ 26.68516
## 25      HNRNPL  5.595874 0.003964471           ++ 26.68088
## 26        YBX3  5.887285 0.005587352           ++ 26.63493
## 27        UPF1  5.023016 0.008503033           ++ 26.62454
## 28        H1FX  5.635593 0.004679567           ++ 26.38324
## 29     ZC3HAV1  5.499019 0.005871785           ++ 26.24666
## 30        UTS2  5.444846 0.004054848           ++ 26.19249
## 31         FBL  5.431179 0.008695255           ++ 26.17882
## 32       NOP56  5.183590 0.005871785           ++ 25.93123
## 33        TOP1  5.069536 0.004679567           ++ 25.81718
## 34      PABPC4  5.060279 0.005871785           ++ 25.80792
## 35     FAM120A  5.036649 0.006250957           ++ 25.78429
## 36       TMED1  4.989796 0.009510799           ++ 25.73744
## 37       DDX21  4.806712 0.008834485           ++ 25.55436
## 38       SRRM1  4.763614 0.005587352           ++ 25.51126
## 39       MOV10  4.759745 0.005587352           ++ 25.50739
## 40    HIST1H1D  4.536912 0.006436298           ++ 25.28456
```

You'll see we've included our IP protein, which is a good reference for intensity. These are our top hits! But maybe some of these are already top interactors. Let's look at the data at a bigger glance. We'll take all significant interactors and get some data about them. First, we extract all annotaiton types from the NCBI HIV database attributed to Rev and get a list of lists

```
MS_DataSig <- MS_Data[MS_Data$Significance == '++' | MS_Data$Significance == '+', ]
'%!in%' <- function(x,y)!('%in%'(x,y))

Interaction <- c("binds", "interacts with")
RevBinders <- unique(NCBI_DB %>% select(HIV.1_Prot_Name, Keyword, Human_GeneSymbol) %>%
                       filter(HIV.1_Prot_Name == "Rev", Keyword %in% Interaction) %>%
                       select(Human_GeneSymbol))

RevAffiliated <- unique(NCBI_DB %>% select(HIV.1_Prot_Name, Keyword, Human_GeneSymbol) %>%
                          filter(HIV.1_Prot_Name == "Rev") %>% select(Human_GeneSymbol))
RevAffiliated <- RevAffiliated$Human_GeneSymbol[!RevAffiliated$Human_GeneSymbol %in%
                                                 RevBinders$Human_GeneSymbol]

MS_DataSig$RevBinder <- ""
MS_DataSig$RevAffiliated <- ""
MS_DataSig$HIVAffiliated <- ""
MS_DataSig$RevBinder[MS_DataSig$GeneNames %in% RevBinders$Human_GeneSymbol] <- '+'
NoRevBinders <- length(MS_DataSig$RevBinder[MS_DataSig$GeneNames %in% RevBinders$Human_GeneSymbol])
MS_DataSig$RevAffiliated[MS_DataSig$GeneNames %in% RevAffiliated] <- '+'
NoRevAff <- length(MS_DataSig$RevAffiliated[MS_DataSig$GeneNames %in% RevAffiliated])

HIVAffiliated <- NCBI_DB %>% filter(!HIV.1_Prot_Name == "Rev") %>%
  select(HIV.1_Prot_Name, Keyword, Human_GeneSymbol)
HIVAffiliated <- unique(HIVAffiliated[c("Human_GeneSymbol", "Keyword", "HIV.1_Prot_Name")])
HIVAffiliated <- HIVAffiliated$Human_GeneSymbol[!HIVAffiliated$Human_GeneSymbol %in%
                                                 RevBinders$Human_GeneSymbol]
HIVAffiliated <- HIVAffiliated[!HIVAffiliated %in% RevAffiliated]
MS_DataSig$HIVAffiliated[MS_DataSig$GeneNames %in% HIVAffiliated] <- '+'
NoHIVAff <- length(MS_DataSig$HIVAffiliated[MS_DataSig$GeneNames %in% HIVAffiliated])

print(paste0("Of ",length(MS_DataSig$GeneNames),
             " significantly enriched proteins in my ", Positive,
             " IP, there are ",NoHIVAff," proteins previously linked to HIV-1 and ",
             NoRevAff," which have been linked to Rev, with a final ",NoRevBinders,
             " having a direct interaction with ",Positive,"."))
```

```
## [1] "Of 325 significantly enriched proteins in my REV-FLAG-3xMyc IP, there are 144 proteins previousl
```

Let's make this into a Sankey diagram

```
SankeyRevAffiliated <- NCBI_DB %>% select(HIV.1_Prot_Name, Keyword, Human_GeneSymbol) %>%
  filter(HIV.1_Prot_Name == "Rev") %>% select(Human_GeneSymbol, Keyword)
er <- SankeyRevAffiliated[SankeyRevAffiliated$Human_GeneSymbol %in% MS_DataSig$GeneNames,]
results <- as.data.frame(summarise(group_by(er,Human_GeneSymbol,Keyword),count =n()))
results$KeywordFactor <- as.factor(results$Keyword)
results$KeywordFactor <- relevel(results$KeywordFactor, c("binds"))
results$KeywordFactor <- relevel(results$KeywordFactor, c("interacts with"))
levels(results$KeywordFactor)
```

```
## [1] "interacts with"    "binds"             "activates"
## [4] "associates with"   "co-localizes with" "cooperates with"
## [7] "enhanced by"       "inhibited by"      "inhibits"
## [10] "modulated by"      "phosphorylated by" "regulated by"
## [13] "requires"          "stimulated by"     "stimulates"
## [16] "upregulated by"
```

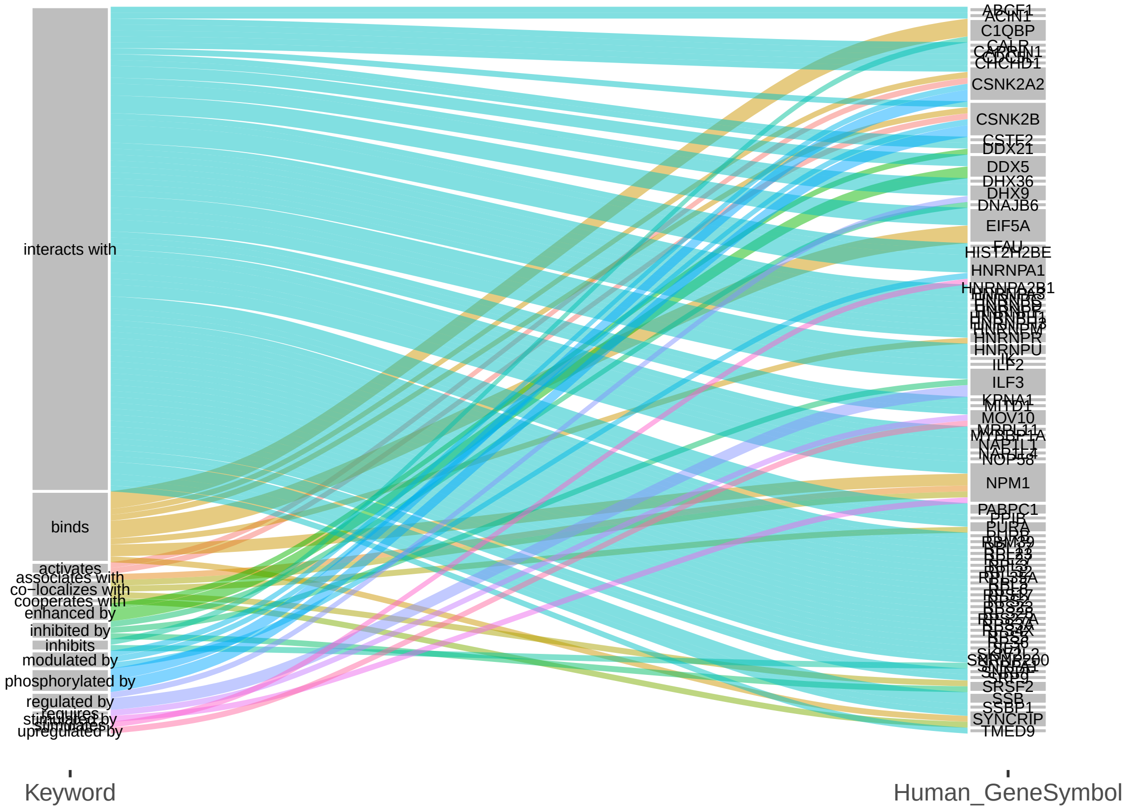
```
results[results$GeneLevels <- factor(results$Human_GeneSymbol),]
```

```
##      Human_GeneSymbol           Keyword count     KeywordFactor
## 1               ABCF1    interacts with     1    interacts with
## 2               ACIN1    interacts with     1    interacts with
## 3               C1QBP             binds     3             binds
## 3.1             C1QBP             binds     3             binds
## 4               C1QBP          inhibits     1          inhibits
## 5                CALR    interacts with     1    interacts with
## 6             CAPRIN1    interacts with     1    interacts with
## 7               CDC5L    interacts with     1    interacts with
## 8              CHCHD1    interacts with     1    interacts with
## 8.1            CHCHD1    interacts with     1    interacts with
## 8.2            CHCHD1    interacts with     1    interacts with
## 8.3            CHCHD1    interacts with     1    interacts with
## 8.4            CHCHD1    interacts with     1    interacts with
## 9             CSNK2A2         activates     1         activates
## 9.1           CSNK2A2         activates     1         activates
## 9.2           CSNK2A2         activates     1         activates
## 9.3           CSNK2A2         activates     1         activates
## 9.4           CSNK2A2         activates     1         activates
## 10            CSNK2A2             binds     1             binds
## 11            CSNK2A2    interacts with     1    interacts with
## 11.1          CSNK2A2    interacts with     1    interacts with
## 12            CSNK2A2      modulated by     1      modulated by
## 12.1          CSNK2A2      modulated by     1      modulated by
## 13            CSNK2A2 phosphorylated by     2 phosphorylated by
## 14             CSNK2B         activates     1         activates
## 14.1           CSNK2B         activates     1         activates
## 15             CSNK2B             binds     1             binds
## 16             CSNK2B    interacts with     1    interacts with
## 16.1           CSNK2B    interacts with     1    interacts with
## 17             CSNK2B      modulated by     1      modulated by
## 18             CSNK2B phosphorylated by     2 phosphorylated by
## 19              CSTF2    interacts with     1    interacts with
## 19.1            CSTF2    interacts with     1    interacts with
## 19.2            CSTF2    interacts with     1    interacts with
## 20              DDX21       enhanced by     1       enhanced by
## 21              DDX21    interacts with     1    interacts with
## 22               DDX5       enhanced by     2       enhanced by
## 23               DDX5    interacts with     2    interacts with
## 24              DHX36    interacts with     1    interacts with
## 25               DHX9    interacts with     2    interacts with
## 26               DHX9      regulated by     1      regulated by
## 27             DNAJB6      inhibited by     1      inhibited by
## 28              EIF5A             binds     3             binds
## 28.1            EIF5A             binds     3             binds
## 29              EIF5A    interacts with     3    interacts with
## 30                FAU    interacts with     1    interacts with
## 31          HIST2H2BE    interacts with     1    interacts with
## 32            HNRNPA1    interacts with     3    interacts with
## 32.1          HNRNPA1    interacts with     3    interacts with
## 32.2          HNRNPA1    interacts with     3    interacts with
## 33            HNRNPA1      modulated by     1      modulated by
## 34            HNRNPA1        stimulates     1        stimulates
## 35          HNRNPA2B1    interacts with     1    interacts with
## 35.1        HNRNPA2B1    interacts with     1    interacts with
## 35.2        HNRNPA2B1    interacts with     1    interacts with
## 36            HNRNPA3    interacts with     1    interacts with
## 37             HNRNPC    interacts with     1    interacts with
## 38             HNRNPD    interacts with     1    interacts with
## 39             HNRNPF    interacts with     1    interacts with
## 40            HNRNPH1    interacts with     1    interacts with
## 41            HNRNPH3    interacts with     1    interacts with
## 41.1          HNRNPH3    interacts with     1    interacts with
## 41.2          HNRNPH3    interacts with     1    interacts with
## 41.3          HNRNPH3    interacts with     1    interacts with
## 41.4          HNRNPH3    interacts with     1    interacts with
## 42             HNRNPM    interacts with     1    interacts with
## 43             HNRNPR             binds     1             binds
## 44             HNRNPR    interacts with     1    interacts with
## 44.1           HNRNPR    interacts with     1    interacts with
## 45             HNRNPU    interacts with     2    interacts with
## 46                 IK    interacts with     1    interacts with
## 47               ILF2    interacts with     1    interacts with
## 48               ILF3      inhibited by     1      inhibited by
## 49               ILF3    interacts with     2    interacts with
## 50               ILF3      regulated by     2      regulated by
## 51              KPNA1    interacts with     1    interacts with
## 52              MITD1    interacts with     1    interacts with
## 53              MOV10    interacts with     1    interacts with
## 54              MOV10          requires     1          requires
## 55              MOV10    upregulated by     1    upregulated by
## 56             MRPL11    interacts with     1    interacts with
## 57            MYBBP1A    interacts with     1    interacts with
## 58             NAP1L1    interacts with     2    interacts with
## 59             NAP1L4    interacts with     1    interacts with
## 60              NOP58    interacts with     1    interacts with
## 61               NPM1   associates with     1   associates with
## 62               NPM1             binds     2             binds
## 63               NPM1 co-localizes with     1 co-localizes with
## 64               NPM1    interacts with     2    interacts with
## 65               NPM1     stimulated by     1     stimulated by
## 66             PABPC1    interacts with     2    interacts with
## 67               PPIB    interacts with     1    interacts with
## 68               PURA co-localizes with     1 co-localizes with
## 69               PURA    interacts with     1    interacts with
## 70               PURB    interacts with     1    interacts with
## 70.1             PURB    interacts with     1    interacts with
## 71              RBM39    interacts with     1    interacts with
## 72              RPL17    interacts with     1    interacts with
## 73              RPL23    interacts with     1    interacts with
## 73.1            RPL23    interacts with     1    interacts with
## 73.2            RPL23    interacts with     1    interacts with
## 74              RPL27    interacts with     1    interacts with
```

```
class(results$GeneLevels)
```

```
## [1] "factor"
```

```
plot.GO.MF <- ggplot(results) + aes(y = count, axis1 = KeywordFactor, axis2 = GeneLevels) +
  geom_alluvium(aes(fill=Keyword), width = 1/12) +
  geom_stratum(width=1/12, fill="grey", color="white") +
  scale_x_discrete(limits = c("Keyword", "Human_GeneSymbol"), expand = c(.05, .05)) +
  theme_bw() + #scale_fill_manual(values=cls) +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
                                                                                      axis.title.y=

plot.GO.MF + theme(legend.position = "none") +
  geom_text(label.strata=T, stat = "stratum",  size= 2)#, direction="y", nudge_x=.5)
```



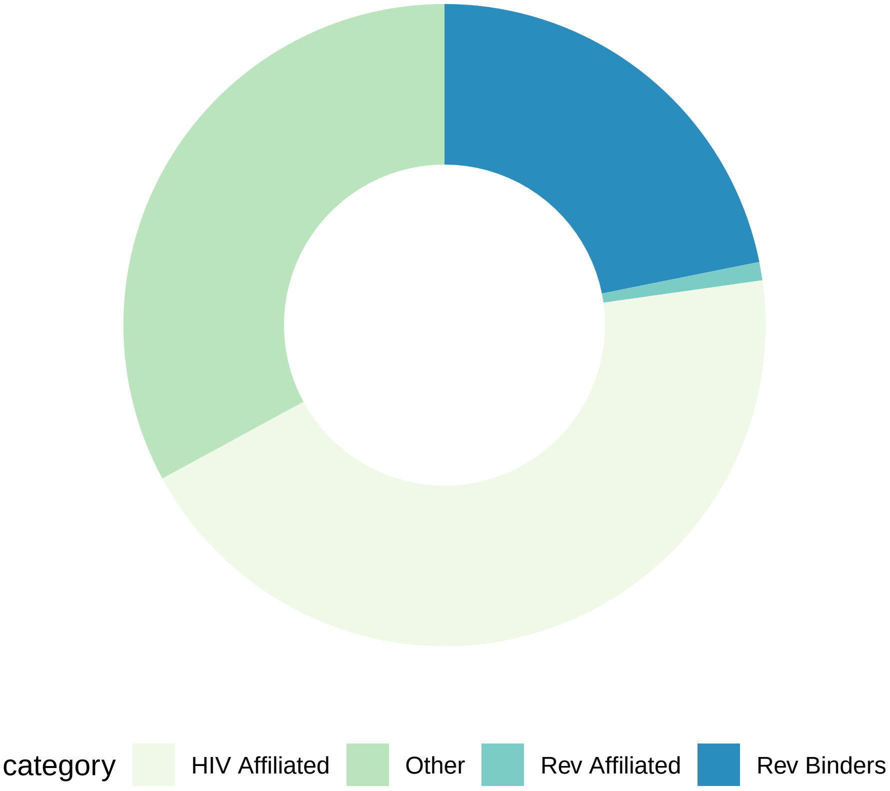
```
ggsave('Sankey2.pdf', plot=last_plot(), path = WD, dpi=1700)
```

```
## Saving 6.5 x 4.5 in image
```

Let's make a doughnut

```
data <- data.frame(
  category=c("Rev Binders", "Rev Affiliated", "HIV Affiliated", "Other"),
  count=c(NoRevBinders, NoRevAff, NoHIVAff, 107)
)
data$fraction = data$count / sum(data$count)
data$ymax = cumsum(data$fraction)
data$ymin = c(0, head(data$ymax, n=-1))
```

```
data$labelPosition <- (data$ymax + data$ymin) / 2
#data$label <- paste0(data$category, "\n value: ", data$count)

doughnut <- ggplot(data, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3, fill=category)) +
  geom_rect() +
  scale_fill_brewer(palette=4) +
  coord_polar(theta="y") +
 # geom_label( x=3.5, aes(y=labelPosition, label=label), size=3) +
  xlim(c(2, 4)) +
  theme_void() + theme(legend.position="bottom")

doughnut
```



Let's save

```
ggsave('DoughnutNoLabel2.pdf', plot=last_plot(), path = WD, dpi=1700)
```

```
## Saving 6.5 x 4.5 in image
```

```
MS_DataSig <- MS_Data[MS_Data$Significance == '+' | MS_Data$Significance == '++' , ]

data("hpaSubcellularLoc")
Loc <- as.data.frame(hpaSubcellularLoc)
Loc$IDs <- as.vector(droplevels(Loc$Gene))
colnames(Loc)[12] <- "IDs"
Loc <- Loc[Loc$Reliability == "Approved" | Loc$Reliability == "Supported" |
             Loc$Reliability == "Enhanced"  ,]
```

```r
MS_DataSig$ID <- ""
MS_DataSig$Localisation <- ""
for(i in 1:length(MS_DataSig$GeneNames)){
  if(MS_DataSig$GeneNames[i] %in% Loc$Gene.name){
    x <- match(MS_DataSig$GeneNames[i], Loc$Gene.name)
    MS_DataSig$ID[i] <- Loc$IDs[x]
    MS_DataSig$Localisation[i] <- as.vector(str_remove_all(droplevels(Loc$GO.id[x]),
                                                           "\\WGO:\\d\\d\\d\\d\\d\\d\\d\\W"))
  }
}



Nuclear_Membrane <- 0 #
Nuclear_Speckles <- 0 #
Nuclear_Bodies <- 0 #
Lipid_Droplets <- 0 #
Peroxisomes <- 0 #
Nucleoli <- 0 #
Nucleoplasm <- 0 #
Actin_filaments <- 0 #
Centrosome <- 0 #
Cytosol <- 0 #
Mitochondria <- 0 #
Microtubules <- 0 #
Golgi_apparatus <- 0 #
Endoplasmic_reticulum <- 0 #
Vesicles <- 0 #
Aggresome <- 0 #
Cytoplasmic_Bodies = 0 #

for(i in 1:length(MS_DataSig$Localisation)){
  ifelse(grepl("Mitochondria", MS_DataSig$Localisation[i]),
         Mitochondria <- Mitochondria + 1, "Mitochondria" == "Mitochondria")

  ifelse(grepl("Nucleoplasm", MS_DataSig$Localisation[i]),
         Nucleoplasm <- Nucleoplasm + 1, "Nucleoplasm" == "Nucleoplasm")

  ifelse(grepl("Cytosol", MS_DataSig$Localisation[i]),
         Cytosol <- Cytosol + 1, "Cytosol" == "Cytosol")

  ifelse(grepl("Nucleoli", MS_DataSig$Localisation[i]),
         Nucleoli <- Nucleoli + 1, "Nucleoli" == "Nucleoli")

  ifelse(grepl("Centrosome", MS_DataSig$Localisation[i]),
         Centrosome <- Centrosome + 1, "Centrosome" == "Centrosome")

  ifelse(grepl("Vesicles", MS_DataSig$Localisation[i]),
         Vesicles <- Vesicles + 1, "Vesicles" == "Vesicles")

  ifelse(grepl("Endoplasmic", MS_DataSig$Localisation[i]),
         Endoplasmic_reticulum <- Endoplasmic_reticulum + 1,
         "Endoplasmic_reticulum" == "Endoplasmic_reticulum")
    ifelse(grepl("Actin", MS_DataSig$Localisation[i]),
           Actin_filaments <- Actin_filaments + 1, "Actin_filaments" == "Actin_filaments")

    ifelse(grepl("Golgi", MS_DataSig$Localisation[i]),
           Golgi_apparatus <- Golgi_apparatus + 1, "Golgi_apparatus" == "Golgi_apparatus")

    ifelse(grepl("Microtubule", MS_DataSig$Localisation[i]),
           Microtubules <- Microtubules + 1, "Microtubules" == "Microtubules")

    ifelse(grepl("Nuclear membrane", MS_DataSig$Localisation[i]),
           Nuclear_Membrane <- Nuclear_Membrane + 1, "Nuclear_Membrane" == "Nuclear_Membrane")

    ifelse(grepl("Nuclear speckles", MS_DataSig$Localisation[i]),
Nuclear_Speckles <- Nuclear_Speckles + 1, "Nuclear_Speckles" == "Nuclear_Speckles")

    ifelse(grepl("Nuclear bodies", MS_DataSig$Localisation[i]),
           Nuclear_Bodies <- Nuclear_Bodies + 1, "Nuclear_Bodies" == "Nuclear_Bodies")

    ifelse(grepl("Aggresome", MS_DataSig$Localisation[i]),
           Aggresome <- Aggresome + 1, "Aggresome" == "Aggresome")

   ifelse(grepl("Cytoplasmic bodies", MS_DataSig$Localisation[i]),
          Cytoplasmic_Bodies <- Cytoplasmic_Bodies + 1, "Cytoplasmic_Bodies" == "Cytoplasmic_Bodies")

   ifelse(grepl("Lipid droplets", MS_DataSig$Localisation[i]),
          Lipid_Droplets <- Lipid_Droplets + 1, "Lipid_Droplets" == "Lipid_Droplets")

   ifelse(grepl("Peroxisomes", MS_DataSig$Localisation[i]),
          Peroxisomes <- Peroxisomes + 1, "Peroxisomes" == "Peroxisomes")
}

Total_Localisation <- sum(!is.na(MS_DataSig$Localisation))

Localisation_Summary <- cbind(Centrosome, Vesicles, Nuclear_Membrane,
                              Golgi_apparatus, Microtubules, Nuclear_Speckles,
                              Nuclear_Bodies, Nucleoplasm, Cytosol, Nucleoli,
                              Lipid_Droplets, Peroxisomes, Endoplasmic_reticulum,
                              Cytoplasmic_Bodies, Aggresome, Mitochondria,
                              Actin_filaments, Total_Localisation)

colnames(Localisation_Summary)[1:18] <- c('Centrosome', 'Vesicles',
                                          'Nuclear Membrane', 'Golgi Apparatus',
                                          'Mirotubules', 'Nuclear Speckles',
                                          'Nuclear Bodies', 'Nucleoplasm', 'Cytosol',
                                          'Nucleoli', 'Lipid_Droplets', 'Peroxisomes',
                                          'Endoplasmic Reticulum', 'Cytoplasmic Bodies',
                                          'Aggresome','Mitochondria','Actin Filaments','Total')
Localisation_Summary
```

```
##      Centrosome Vesicles Nuclear Membrane Golgi Apparatus Mirotubules
## [1,]          2       14                4               7           4
##      Nuclear Speckles Nuclear Bodies Nucleoplasm Cytosol Nucleoli
## [1,]               11              5          79     116       42
##      Lipid_Droplets Peroxisomes Endoplasmic Reticulum Cytoplasmic Bodies
## [1,]              1           1                    58                  3
##      Aggresome Mitochondria Actin Filaments Total
## [1,]         1           40               1   325
```

Comparing to Manuel's dataset…

```
capsid <- read.delim('C:/Users/Greye/Dropbox/DPHIL PHD UPDATED/DATA/MASS SPECTRONOMY/REPLICATES/USING RE
capsid$Genes <- sapply(strsplit(as.character(capsid$Genes), ";"), "[", 1)
MS_DataSig$Capsid = ""
MS_DataSig$Capsid[MS_DataSig$GeneNames %in% capsid$Genes] <- '+'
Manuel <- unique(MS_DataSig$GeneNames[MS_DataSig$Capsid == '+'])
```