Looking at oligodT enrichment of proteins in HeLa,HEK293 and HuH7

```
#-----------------------------------------------------------------------------
# Author          : Manasa Ramakrishna, mr325@le.ac.uk
# Date started  : 15th September, 2017
# Last modified : 15th September, 2017
# Aim             : Given a set of protein identifiers, map to GO and interpro terms
#                 and return dictionaries for both. Used as input in the 'gene2cat' variable in goseq
#-----------------------------------------------------------------------------

# Invoking libraries
library(mygene)
library(goseq)
library(limma)
library(ggplot2)
library(stringr)
library(data.table)
library(plyr)
library(MSnbase)

#Setting working directories
wd = "/Users/manasa/Documents/Work/TTT/02_Proteomics/03_Leicester-Oligo-dT/"
setwd(wd)
getwd()
```

```
## [1] "/Users/manasa/Documents/Work/TTT/02_Proteomics/03_Leicester-Oligo-dT"
```

```
indir = ("Input/")
outdir = paste("/Users/manasa/Documents/Work/TTT/02_Proteomics/03_Leicester-Oligo-dT/",paste(Sys.Date()

if (exists(outdir)){
  print("Outdir exists")
}else{
  dir.create(outdir)
}
```

```
# ------------------------------------------------------------------------------------------------------
# Step 01: Read background list of proteins for HeLa and HEK293T
# ------------------------------------------------------------------------------------------------------

all.lines <- read.delim("Input/geiger-peptide-11-cell-lines_reannot.txt",sep="\t",header=T,stringsAsFac
head(all.lines,10)
dim(all.lines) # 158292 50

# Filter and subset data
# Filter data - keep peptides with unique master proteins, those which are not "crap" and those that ar
all.filt = all.lines[which(all.lines$unique == 1 & all.lines$master_protein != "" & all.lines$crap_prot
dim(all.filt) # 148593 45

# Loop over dataset and split into multiple cell lines
metadat = all.filt[,grep("Intensity",colnames(all.filt),invert=T)]
```

```
cell.lines = sapply(strsplit(grep("Intensity",colnames(all.filt),value=T),"\\."),"[[",2)

# Split data into each cell line along with metadata
for(j in seq(1,length(cell.lines),by = 3)){
  name = strsplit(cell.lines[j],"_")[[1]][1]
  print(cell.lines[j:(j+2)])
  temp = all.filt[,grep(paste(cell.lines[j:(j+2)],collapse="|"),colnames(all.filt))]
  temp = cbind(metadat,temp)

  agg = aggProt(temp)
  print(dim(agg))

  write.table(temp,paste(outdir,paste(name,"Background-list-of-peptides.txt",sep="_"),sep="/"),sep="\t"
  write.table(agg,paste(outdir,paste(name,"Background-list-of-proteins.txt",sep="_"),sep="/"),sep="\t",
}

# ------------------------------------------
# Function : aggProt
# ------------------------------------------

aggProt <- function(pepdat){

  # Aggregate to peptide groups
  a = aggregate(pepdat[,grep("Intensity",colnames(pepdat))],by=list(sequence=pepdat$Sequence,master_pro

  # Aggregate to proteins
  b = aggregate(a[,grep("Intensity",colnames(a))],by=list(master_protein=a$master_protein,protein_lengt

  b = aggregate(a[,grep("Intensity",colnames(a))],by=list(master_protein=a$master_protein),FUN="median"

  # Remove all 0 rows
  c = b[which(rowSums(b[,grep("Intensity",colnames(b))],na.rm=T)!=0),]
  c$max = apply(c[,grep("Intensity",colnames(c))],1,"max",na.rm=T)

  # Losses at each step
  dim(a)
  dim(b)
  dim(c)

  return(c)
}
```