Define quality threshold

```
library(DBI)
library(RMySQL)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

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

```
library(rlist)
con <- DBI::dbConnect(RMySQL::MySQL(),
                      host = "localhost",
                      user = "root",
                      dbname = "cagablea",
                      password = "")
```

A list of all the requests calling samples:

The function below generates a list with the requests to download the samples Treshholds is a variable, where the cutoff on coverage, p.val, alt_freq and others can be added:

```
samples <- function(treshholds){
  samples <- list()
  req <- "SELECT * FROM `ww_ivar`,`ww_barcode_decode` WHERE ww_barcode_decode.barcode = ww_ivar.filename
  samples$oct <- "AND ww_ivar.filename LIKE BINARY '%210224%'"
  samples$nov <- "AND ww_ivar.filename LIKE BINARY '%20201120%'"
  samples$dec <- "AND ww_ivar.filename LIKE BINARY '%20201223%'"
  samples$jan <- "AND ww_ivar.filename LIKE BINARY '%210127%'"
  samples$fev <- "AND ww_ivar.filename LIKE BINARY '%20210302%'"
  samples$mar <-  "AND ww_ivar.filename LIKE BINARY '%210325%'"
  samples <- lapply(samples, function(x) paste(req, treshholds, x, " "))
  return(samples)
}
treshholds <- "AND ww_ivar.total_dp > 10 AND ww_ivar.pval < 0.05 AND ww_ivar.alt_freq > 0.03"
samples1 <- samples(treshholds)
```

Count mutations not associated to any lineage and the mutations described for at least one lineage: TRUE correspons to the mutations that are not described for any lineages

```
no_lineage <- function(d_request){
  feb <- dbGetQuery(con, d_request)
  feb <- feb[,-1]
  feb <- distinct(feb, pos, alt, .keep_all = F)
  #how many mutations are not associated with any lineage?
  pango <- dbGetQuery(con, "SELECT * FROM `pango_lineages`")
  all.mut.id <-  feb %>% left_join(pango, by = c("pos"="pos", "alt"="alt"), keep = T )
  all.mut.id <- all.mut.id %>% select(pos.x, ref, alt.x, lineage )%>% distinct(pos.x, alt.x, ref, .keep
  #all.mut.id
  all.mut.id %>% count(is.na(lineage))

}
#no_lineage(samples1$mar)#test
#Positions with nonsynonymous mutations in spike region not associated with lineages (top 2000 from cov
  #lin_na %>% filter (ref_aa != alt_aa) %>% filter(pos.x >= 21563) %>% filter(pos.x <= 25384)
n_mut <- function(samples){
  n_id_mutations <- lapply(samples, no_lineage)
  #n_id_mutations #the number of id and unidentified mut, min number of reads 10
  n_mut <- cbind(n_id_mutations$oct, n_id_mutations$nov$n, n_id_mutations$dec$n, n_id_mutations$jan$n,
  names(n_mut) <- c("lin_not_id", "oct", "nov", "dec", "jan", "fev", "mar")
  n_mut
}
```

total_dp > 10 AND ww_ivar.pval < 0.05 AND ww_ivar.alt_freq > 0.03

```
treshholds <- "AND ww_ivar.total_dp > 10 AND ww_ivar.pval < 0.05 AND ww_ivar.alt_freq > 0.03"
samples1 <- samples(treshholds)
n_mut(samples1)
```

```
##   lin_not_id  oct   nov   dec  jan  fev  mar
## 1      FALSE  152   192   229  280  229  235
## 2       TRUE 5947 15187 20229 8132 6822 7576
```

Minimum number of reads 100, alt_freq 0.1, pval<0.01:

```
treshholds <- "AND ww_ivar.total_dp > 100 AND ww_ivar.pval < 0.01 AND ww_ivar.alt_freq > 0.1"
samples1 <- samples(treshholds)
n_mut(samples1)
```

```
##   lin_not_id oct  nov  dec jan fev  mar
## 1      FALSE  88   84   98 160 115  137
## 2       TRUE 756 1277 1925 981 856 1145
```