```
library(stringr)
library(ape)
library(dplyr)
library(readr)
library(tidyr)
library(reshape2)
library(ggplot2)
library(ggrepel)
library(ggbeeswarm)
library(seqinr)
```

Part 0. Ready to start

This is a code, used for analysis of patient-specific mutation effects on atigen presentation for CD4 and CD8 T cells, used in the study "SARS-CoV-2 escape from cytotoxic T cells during long-term COVID-19".

To run this code we used R version 4.0.0 (2020-04-24) and the list of R packages of following version:

```
packages <- data.frame(name = c("stringr", "ape", "dplyr", "readr", "tidyr", "reshape2", "ggplot2",
                                "ggrepel", "ggbeeswarm", "seqinr"))
for (i in packages$name) {packages$version[packages$name == i] <- as.character(packageVersion(i))}
packages
```

```
##          name version
## 1     stringr   1.4.0
## 2         ape   5.4.1
## 3       dplyr   1.0.2
## 4       readr   1.3.1
## 5       tidyr   1.1.2
## 6    reshape2   1.4.4
## 7     ggplot2   3.3.2
## 8     ggrepel   0.9.0
## 9  ggbeeswarm   0.6.0
## 10     seqinr   3.6.1
```

The code assumes, that the working directory contains following objects:

1. **pat__S__mut.rds** - the list of analysed aminoacid changing mutations (all that viral genome accumulated during the study period);

```
head(mutations)
```

```
##                   g_coord pos end    or   mut gene mut_type          name g_pos
## 1             C:21711:T  50  50     S     L    S    subst        S:S50L 21711
## 2       ATACATG:21764:A  69  70   IHV   XXI    S      del    S:del68_70 21764
## 3 TTTTGGGTGTTTA:21981:T 140 144 FLGVY XXXXX    S      del S:del140_144 21981
## 4             G:22381:T 273 273     R     S    S    subst       S:R273S 22381
## 5             T:22882:G 440 440     N     K    S    subst       S:N440K 22882
## 6             T:22917:G 452 452     L     R    S    subst       S:L452R 22917
##            g_or g_mut
## 1             C     T
## 2       ATACATG     A
## 3 TTTTGGGTGTTTA     T
## 4             G     T
## 5             T     G
## 6              T     G
```

- **g_coord** - chane in nucleotide suqence;
- **pos** - amino acid strarting position of the change;
- **end** - amino acid ending position of a change (for substitusions **pos** == **end**);
- **or** - aminoacids before the change (X indicates gap after deletion);
- **mut** - aminoacids after the change (X indicates gap after deletion);
- **gene** - viral gene, where the change happened;
- **mut_type** - type of change (*subst* - substitution, *del* - deletion, *stop* - stop codon producing);
- **name** - name of mutation;
- **g_pos** - starting nucleotide position of the change;
- **g_or** - nucleotides before the change;
- **g_mut** - nucleotides after the change.

2. **pat_S_seq.rds** - the list of analysed aminoacid changing mutations (all that viral genome accumulated during the study period);

- **gene** - gene;
- **start** - nucleotide starting coordinate of the gene;
- **end** - nucleotide ending coordinate of the gene;
- **seq** - aminoaicd gene sequence;

3. **russian.fa** - SARS-CoV2 genome, sampled from the patient;
4. **found_june1.txt** - known epitopes, immunogenic on patient-specific hla alleles from Immune Epitope Database:

```
head(found)
```

```
##        allele     peptide class
## 1 HLA-A*01:01  TTDPSFLGRY  HLAI
## 2 HLA-A*01:01   PTDNYITTY  HLAI
## 3 HLA-A*01:01 HTTDPSFLGRY  HLAI
## 4 HLA-A*01:01    TDNYITTY  HLAI
## 5 HLA-A*01:01 TTDPSFLGRYM  HLAI
## 6 HLA-A*01:01   LFLAFVVFL  HLAI
```

5. **allele_freq.txt** - worldwild frequency of hla alleles analysed.

To run it first define the path of your working directory and load dataframes, described above:

```
path <- "/home/jane/PhD_Skoltech/coronavirus/immunocompromised/SUBMISSION/beauty_code/"
mutations <- readRDS(paste0(path, "pat_S_mut.rds"))
coord <- readRDS(paste0(path, "pat_S_seq.rds"))
ref <- as.character(read.FASTA(paste0(path, "russian.fa"), type = "DNA"))
found <- read.table("/home/jane/PhD_Skoltech/coronavirus/immunocompromised/found_june1.txt")
found <- data.frame(found)
colnames(found) <- c("allele", "peptide", "class")
hla_freq <- read.table(paste0(path, "allele_freq.txt"))
hla_freq <- data.frame(hla_freq)
colnames(hla_freq) <- c("allele", "short_name", "freq")
```

Part 1. Peptides, covering changed sites, and heir binding affinities

This chunk produces two files (mhcI.pept and mhcII.pept), containing the list of HLA class I (8-14 aa long) and HLA class II (12-18 aa long) epitopes, changed due to mutations. For stop-codon creating mutations it will write all epitopes of all suitable lengths, which were lost due to stop codon.

```r
peptides_table <- c()
for (m in 1:nrow(mutations)){
  ## general stuff
  seq_or <- coord$seq[coord$gene == mutations$gene[m]]
  substring(seq_or, mutations$pos[m], mutations$end[m]) <- mutations$or[m]
  seq_mut <- seq_or
  substring(seq_mut, mutations$pos[m], mutations$end[m]) <- mutations$mut[m]

  ## for substitution mutations
  if (mutations$mut_type[m] == "subst") {
    for (i in 8:19){
      mut_start <- mutations$pos[m] - i + 1
      mut_end <- mutations$end[m] + i - 1
      if (mut_start <= 0) {mut_start = 1}
      if (mut_end >= nchar(seq_or)) {mut_end >= nchar(seq_or)}
      mut_region_or <- substring(seq_or, mut_start, mut_end)
      mut_region_mut <- substring(seq_mut, mut_start, mut_end)

      for (j in 1:(nchar(mut_region_or) - i + 1)) {
        pept_or <- substring(mut_region_or, j, j + i - 1)
        pept_mut <- substring(mut_region_mut, j, j + i - 1)
        pept_or <- str_remove_all(pept_or, "X")
        pept_mut <- str_remove_all(pept_mut, "X")
        if (nchar(pept_or) >= 8 & nchar(pept_or) <= 18){
        peptides_table <- rbind(peptides_table,
                          c(mutations$gene[m], mutations$mut_type[m], mutations$name[m], pept_or, pept_
        }
      }
    }

  ## for deletions
  ## by genome coordinates
  } else if (mutations$mut_type[m] == "del"){

    gene_or <- toupper(paste0(ref[[1]][as.numeric(as.character(coord$start[coord$gene == mutations$gene
                          as.numeric(as.character(coord$end[coord$gene == mutations$gene[m]]))], colla
    substring(gene_or, mutations$g_pos[m] - 100 - as.numeric(coord$start[coord$gene == mutations$gene[m
            mutations$g_pos[m] - 100 - as.numeric(coord$start[coord$gene == mutations$gene[m]]) + 1 +
              nchar(mutations$g_or[m]) - 1) <- mutations$g_or[m]

    gene_mut <- gene_or
    substring(gene_mut, mutations$g_pos[m] - 100 - as.numeric(coord$start[coord$gene == mutations$gene[
            mutations$g_pos[m] - 100 - as.numeric(coord$start[coord$gene == mutations$gene[m]]) + 1 +
              nchar(mutations$g_or[m]) - 1) <- paste0(mutations$g_mut[m], rep("-",  nchar(mutations$g_o
                                                    collapse = "")

    pr_or <- paste0(translate(unlist(str_split(gene_or, ""))), collapse = "")
    pr_mut <- paste0(translate(unlist(str_split(gene_mut, ""))), collapse = "")
    del_size <- nchar(mutations$or[m])
    for (i in 8:19){
      mut_start <- mutations$pos[m] - i + 1
      mut_end <- mutations$end[m] + i - 1
      if (mut_start <= 0) {mut_start = 1}
      if (mut_end >= nchar(pr_or)) {mut_end >= nchar(pr_or)}
      mut_region_or <- substring(pr_or, mut_start, mut_end)
      mut_region_mut <- substring(pr_mut, mut_start, mut_end)
      mut_region_mut <- paste0(str_remove_all(mut_region_mut, "X"),
                               substring(pr_mut, mut_end + 1, mut_end + str_count(mut_region_mut, "X"))

      for (j in 1:(nchar(mut_region_or) - i + 1)) {
        pept_or <- substring(mut_region_or, j, j + i - 1)
        pept_mut <- substring(mut_region_mut, j, j + i - 1)
        pept_or <- str_remove_all(pept_or, "X")

        if (nchar(pept_or) >= 8 & nchar(pept_or) <= 18){
        peptides_table <- rbind(peptides_table,
                           c(mutations$gene[m], mutations$mut_type[m], mutations$name[m], pept_or, pept
        }
      }
    }

  ## for stops
  } else if (mutations$mut_type[m] == "stop"){
    for (i in 8:19){
      mut_start <- mutations$pos[m] - i + 1
      mut_end <- nchar(seq_or)
      if (mut_start <= 0) {mut_start = 1}
      mut_region_or <- substring(coord$seq[coord$gene == mutations$gene[m]], mut_start, mut_end)

      for (j in 1:(nchar(mut_region_or) - i + 1)) {
        pept_or <- substring(mut_region_or, j, j + i - 1)
        pept_mut <- "-"
        pept_or <- str_remove_all(pept_or, "X")
        if (nchar(pept_or) >= 8 & nchar(pept_or) <= 18){
          peptides_table <- rbind(peptides_table,
                            c(mutations$gene[m], mutations$mut_type[m], mutations$name[m], pept_or, pep
        }
      }
    }
  }
}
peptides_table <- data.frame(peptides_table)
colnames(peptides_table) <- c("gene", "mut_type", "mut", "pept_or", "pept_mut", "l")
peptides_table[, c(1:6)] <- sapply(peptides_table[, c(1:6)], as.character)
peptides_table[, 6] <- as.numeric(peptides_table[, 6])
peptides_table <-  distinct(peptides_table, .keep_all = FALSE)
saveRDS(peptides_table, file = paste0(path, "peptides_table_8_18.Rds", sep = ""))

peptides_I <- c(peptides_table$pept_or[peptides_table$l >= 8 & peptides_table$l <= 14],
              peptides_table$pept_mut[peptides_table$l >= 8 & peptides_table$l <= 14 & peptides_table$pe
```

```
write.table(peptides_I, quote = FALSE, sep = "\n", col.names = FALSE, row.names = FALSE, file = paste0(

peptides_II <- c(peptides_table$pept_or[peptides_table$l >= 12 & peptides_table$l <= 18],
              peptides_table$pept_mut[peptides_table$l >= 12 & peptides_table$l <= 18 & peptides_table$

write.table(peptides_II, quote = FALSE, sep = "\n", col.names = FALSE, row.names = FALSE, file = paste0
head(peptides_table)
```

```
##   gene mut_type    mut  pept_or pept_mut l
## 1    S    subst S:S50L FRSSVLHS FRSSVLHL 8
## 2    S    subst S:S50L RSSVLHST RSSVLHLT 8
## 3    S    subst S:S50L SSVLHSTQ SSVLHLTQ 8
## 4    S    subst S:S50L SVLHSTQD SVLHLTQD 8
## 5    S    subst S:S50L VLHSTQDL VLHLTQDL 8
## 6    S    subst S:S50L LHSTQDLF LHLTQDLF 8
```

It also produces **peptides_table** data.frame, useful in next partf of analysis analysis. It includes following columns:

- **gene** - gene;
- **mut_type** - type of mutation, affected the peptide;
- **pept_or** - peptide before the mutation;
- **pept_mut** - peptide after the mutation;
- **l** - the length of the peptide;

Next we run netMHCpan and netMHCIIpan to caculate binding affinities of all peptides both before and after mutations. It calculates binding affinities for the most frequent HLA alleles of each family (A, B, C, DP, DR, DQ), covering together 95% of the human population. This set of alleles include alleles of the patinet as well. To run netMHC(II)pan we use next commands:

For HLA II epitopes: > netMHCpan -p mhcI.pept -BA -xls -a HLA-A01:01,HLA-A02:01,HLA-A02:02,HLA-A02:03,HLA-A02:04,HLA-A02:05,HLA-A02:06,HLA-A02:07,HLA-A02:11, HLA-A03:01,HLA-A11:01,HLA-A11:02,HLA-A23:01,HLA-A24:02,HLA-A24:07,HLA-A25:01,HLA-A26:01,HLA-A29:02, HLA-A30:01,HLA-A30:02,HLA-A31:01,HLA-A32:01,HLA-A33:01,HLA-A33:03,HLA-A34:01,HLA-A34:02,HLA-A36:01, HLA-A66:01,HLA-A68:01,HLA-A68:02,HLA-A74:01,HLA-B07:02,HLA-B07:04,HLA-B08:01,HLA-B13:01,HLA-B13:02, HLA-B14:02,HLA-B15:01,HLA-B15:02,HLA-B15:03,HLA-B15:10,HLA-B15:17,HLA-B18:01,HLA-B27:05,HLA-B35:01, HLA-B35:03,HLA-B35:07,HLA-B37:01,HLA-B38:01,HLA-B38:02,HLA-B40:01,HLA-B40:02,HLA-B40:06,HLA-B42:01, HLA-B44:02,HLA-B44:03,HLA-B45:01,HLA-B46:01,HLA-B49:01,HLA-B50:01,HLA-B51:01,HLA-B52:01,HLA-B53:01, HLA-B54:01,HLA-B55:01,HLA-B55:02,HLA-B56:01,HLA-B57:01,HLA-B57:03,HLA-B58:01,HLA-B58:02,HLA-C01:02, HLA-C02:02,HLA-C03:02,HLA-C03:03,HLA-C03:04,HLA-C04:01,HLA-C04:03,HLA-C05:01,HLA-C06:02,HLA-C07:01, HLA-C07:02,HLA-C07:04,HLA-C08:01,HLA-C08:02,HLA-C12:02,HLA-C12:03,HLA-C14:02,HLA-C14:03,HLA-C15:02, HLA-C16:01,HLA-C17:01,HLA-C18:01 -xlsfile mhcI.pept.out.xls > mhcI.pept.out.txt grep 'HLA' mhcI.pept.out.txt | grep -v "Link" | grep -v "Protein" |
grep -v "Distance" | grep -v "#" > mhcI.pept.out.filtered

For HLA II epitopes: > netMHCIIpan -inptype 1 -f mhcII.pept -BA -xls -a HLA-DPA10103-DPB10401,HLA-DPA10103-DPB10201,HLA-DPA10103-DPB10402,HLA-DPA10103-DPB10301, HLA-DPA10103-DPB10101,HLA-DPA10103-DPB11401,HLA-DPA10104-DPB10501,HLA-DPA10105-DPB11701, HLA-DPA10106-DPB10601,HLA-DPA10106-DPB11501,HLA-DPA10107-DPB10901,HLA-DPA10107-DPB11001, HLA-DPA10107-DPB11101,HLA-DPA10108-DPB11601,HLA-DPA10104-DPB10202,HLA-DPA10104-DPB11801, HLA-DPA10103-DPB12801,HLA-DPA10104-DPB12701,DRB1_1501,DRB1_0101,DRB1_0701,DRB1_1301,DRB1
DRB1_0401,DRB1_0801,DRB1_0404,DRB1_1104,DRB1_1601,DRB1_1201,DRB1_1302,DRB1_1303,DRB1_0901,DRB1_1
DRB1_0402,DRB1_1502,DRB1_1001,DRB1_1102,DRB1_0405,DRB1_0803,DRB1_1202,DRB1_1602,DRB1_0403,DRB1_0

DRB1__1402,DRB1__0407,DRB1__0406,DRB1__0102,DRB1__0411,DRB1__1503,DRB1__1405,DRB1__1406,DRB1__0302, HLA-DQA10101-DQB10301,HLA-DQA10101-DQB10501,HLA-DQA10101-DQB10602,HLA-DQA10102-DQB10302, HLA-DQA10501-DQB10201,HLA-DQA10103-DQB10603,HLA-DQA10104-DQB10303,HLA-DQA10104-DQB10402, HLA-DQA10104-DQB10502,HLA-DQA10101-DQB10401,HLA-DQA10104-DQB10503,HLA-DQA10104-DQB10604, HLA-DQA10104-DQB10601,HLA-DQA10105-DQB10304,HLA-DQA10101-DQB10607,HLA-DQA10103-DQB10305, HLA-DQA10101-DQB10608,HLA-DQA10102-DQB10203
-xlsfile mhcII.pept.out.xls > mhcII.pept.out.txt grep 'HLA' mhcII.pept.out.txt | grep -v "Link" | grep -v "Protein" |
grep -v "Distance" | grep -v "#" > mhcII.pept.out.filtered

Result output files mhcI.pept.out.filtered and mhcII.pept.out.filtered are used for calculations of BR and PHBR scores, described in the article.

Part 2. Analysis of mutation effects on atigen presentation on HLA I and HLA II

Loading binding affinities of HLA epitopes, calculated by netMHCpan and netMHCIIpan:

```
## for HLA I
net_pan_files <- list.files(paste0(path, "net_pan/"), pattern = "*filtered")
net_pan_files <- net_pan_files[!str_detect(net_pan_files, "hlaD")]
net_panI <- c()

for (f in net_pan_files){
  net_pan_f <- read_tsv(paste0(path, "net_pan/", f, collapse = ""), col_names = FALSE) %>%  separate(X1
  net_pan_f <- data.frame(net_pan_f[,c(1:17)])
  net_pan_f[, c(1,5:9,12:16)] <- sapply(net_pan_f[, c(1,5:9,12:16)], as.numeric)
  net_pan_f <- unique(net_pan_f[,c(2,3,4,10,12:17)])
  net_panI <- rbind(net_panI, net_pan_f)
}

## for HLA II
net_pan_files <- list.files(paste0(path, "net_pan/"), pattern = "*filtered")
net_pan_files <- net_pan_files[str_detect(net_pan_files, "hlaD")]
net_panII <- c()

for (f in net_pan_files){
  net_pan_f <- read_tsv(paste0(path, "net_pan/", f, collapse = ""), col_names = FALSE) %>%  separate(X1
  net_pan_f <- data.frame(net_pan_f[,c(1:17)])
  net_pan_f[, c(1,5:9,12:16)] <- sapply(net_pan_f[, c(1,5:9,12:16)], as.numeric)
  net_pan_f <- unique(net_pan_f[,c(2,3,4,10,12:17)])
  net_panII <- rbind(net_panII, net_pan_f)
}
```

Calculating BR tor each mutation and each HLA I allele analysed. This chunk outouts **best__rank__alleles__I** and **best__ranke__alleles__II** dataframes for HLA I and HLA II effects rrespectively, which includes following columns:

- **mut** - full name of mutation;
- **state** - BR value before mutation (*or*) or after mutation (*mut*);

… - next columns are called accroding to HLA allele, where BRs before and after mutation were calculated.

```r
allelesI <- unique(net_panI$allele)
# Rank_EL thresholds
thresholds <- c(0, 0.5, 2)
# focusing on HLA I epitopes
peptides_table_I <- peptides_table[nchar(peptides_table$pept_or) <= 11,]
best_rank_alleles_I <- c()
muts <- unique(peptides_table_I$mut)
for (i in muts){
  ## all peptides, covering mutated sites
  peptides_or <- peptides_table_I$pept_or[peptides_table_I$mut == i]
  peptides_mut <- peptides_table_I$pept_mut[peptides_table_I$mut == i]
  rank_or <- c()
  rank_mut <- c()
  for (al in allelesI){
     rank_or <- c(rank_or, min(net_panI$Rank_EL[is.element(net_panI$peptide, peptides_or) & net_panI$all
     if (i == "ORF8:Q18*"){
       rank_mut <- c(rank_mut, 100)
     } else {
      rank_mut <- c(rank_mut, min(net_panI$Rank_EL[is.element(net_panI$peptide, peptides_mut) & net_panI
     }
  }
  best_rank_alleles_I <- rbind(best_rank_alleles_I,
                                c(i, "or", rank_or))
  best_rank_alleles_I <- rbind(best_rank_alleles_I,
                                c(i, "mut", rank_mut))
  #best_rank_alleles <- rbind(best_rank_alleles,
  #                               c(i, "dev", rank_mut/rank_or))
}

best_rank_alleles_I <- data.frame(best_rank_alleles_I)
colnames(best_rank_alleles_I) <- c("mut", "state", allelesI)
best_rank_alleles_I[, 3:97] <- sapply(best_rank_alleles_I[, 3:97], as.numeric)

## for HLA II
allelesII <- unique(net_panII$allele)
# focusing on HLA II epitopes
peptides_table_II <- peptides_table[nchar(peptides_table$pept_or) >= 12,]
best_rank_alleles_II <- c()
for (i in muts){
  peptides_or <- peptides_table_II$pept_or[peptides_table_II$mut == i]
  peptides_mut <- peptides_table_II$pept_mut[peptides_table_II$mut == i]
  rank_or <- c()
  rank_mut <- c()
  for (al in allelesII){
     rank_or <- c(rank_or, min(net_panII$Rank_EL[is.element(net_panII$peptide, peptides_or) &
                                                   net_panII$allele == al]))
     if (i == "ORF8:Q18*"){
        rank_mut <- c(rank_mut, 100)
     } else {
        rank_mut <- c(rank_mut, min(net_panII$Rank_EL[is.element(net_panII$peptide, peptides_mut) &
                                                        net_panII$allele == al]))
     }
  }
  best_rank_alleles_II <- rbind(best_rank_alleles_II,
                                c(i, "or", rank_or))
  best_rank_alleles_II <- rbind(best_rank_alleles_II,
                                c(i, "mut", rank_mut))
}

best_rank_alleles_II <- data.frame(best_rank_alleles_II)
colnames(best_rank_alleles_II) <- c("mut", "state", allelesII)
best_rank_alleles_II[, 3:73] <- sapply(best_rank_alleles_II[, 3:73], as.numeric)

head(best_rank_alleles_I[,1:7])
```

```
##             mut state HLA-A*01:01 HLA-A*02:01 HLA-A*02:02 HLA-A*02:03
## 1        S:S50L    or       1.130       1.679       1.300       2.416
## 2        S:S50L   mut       1.583       1.496       1.950       1.741
## 3    S:del68_70    or       5.871       1.043       2.442       2.135
## 4    S:del68_70   mut       6.954      23.109      20.624      14.532
## 5 S:del140_144    or       0.381       0.542       0.832       0.592
## 6 S:del140_144   mut       4.643      14.414      16.438      24.105
##   HLA-A*02:04
## 1       1.521
## 2       0.950
## 3       0.545
## 4      21.196
## 5       0.683
## 6       8.983
```

Next chunk caclulates PHBR fold change for the patient-specific HLA alleles. It produces *phbrI* and *phbrII* dataframes, consisting of the following columns:

- **mut** - full name of mutation;
- **phbr__or** - patient-specific PHBR before mutation;
- **phbr__mut** - patient specific PHBR after mutation;
- **presented** - indicates whether site of the mutation can be presented at least at one HLA allele and at least before or after mutation (yes or no);
- **rel** - PHBR fold change (PHBR after mutation / PHBR before mutation).

```
## PHBR for HLA I
best_rank_alleles_I %>%  #[best_rank_alleles_I$state != "delta", ] %>%
  melt -> df
s_alleles_I <- c("HLA-A*01:01", "HLA-A*03:01", "HLA-B*07:02", "HLA-B*08:01", "HLA-C*07:01", "HLA-C*07:02
s_best_I <- df[is.element(df$variable, s_alleles_I), ]

phbrI <- c()
for (i in muts){
  values_or <- s_best_I$value[s_best_I$mut == i & s_best_I$state == "or"]
  values_mut <- s_best_I$value[s_best_I$mut == i & s_best_I$state == "mut"]
  if (sum(values_or <= 2) > 0 | sum(values_mut <= 2) > 0) {
    phbrI <- rbind(phbrI, c(i, 6/sum(1/values_or), 6/sum(1/values_mut), "yes"))
  } else {
    phbrI <- rbind(phbrI, c(i, 6/sum(1/values_or), 6/sum(1/values_mut), "no"))
  }
}
```

```
phbrI <- data.frame(phbrI)
colnames(phbrI) <- c("mut", "phbr_or", "phbr_mut", "presented")
phbrI[,2:3] <- sapply(phbrI[,2:3], as.numeric)
phbrI$rel <- phbrI$phbr_mut/phbrI$phbr_or

### to lable only mutations with the highest fold change
phbrI$mut1 <- phbrI$mut
phbrI$mut1[phbrI$rel <= 2] <- NA

best_rank_alleles_II %>%
  melt -> df
s_alleles_II <- c("DRB1_0101",  "DRB1_0301", "HLA-DQA10501-DQB10201", "HLA-DQA10101-DQB10501",
               "HLA-DPA10103-DPB10402", "HLA-DPA10103-DPB10401")
s_best_II <- df[is.element(df$variable, s_alleles_II), ]

## PHBR for HLA II
phbrII <- c()
for (i in muts){
  values_or <- s_best_II$value[s_best_II$mut == i & s_best_II$state == "or"]
  values_mut <- s_best_II$value[s_best_II$mut == i & s_best_II$state == "mut"]
  if (sum(values_or <= 10) > 0 | sum(values_mut <= 10) > 0) {
    phbrII <- rbind(phbrII, c(i, 6/sum(1/values_or), 6/sum(1/values_mut), "yes"))
  } else {
    phbrII <- rbind(phbrII, c(i, 6/sum(1/values_or), 6/sum(1/values_mut), "no"))
  }
  #phbr <- rbind(phbr, c(i, 6/sum(1/values_mut), "mut"))
}
phbrII <- data.frame(phbrII)
colnames(phbrII) <- c("mut", "phbr_or", "phbr_mut", "presented")
phbrII[,2:3] <- sapply(phbrII[,2:3], as.numeric)
phbrII$rel <- phbrII$phbr_mut/phbrII$phbr_or

phbrII$mut1 <- phbrII$mut
phbrII$mut1[phbrII$rel <= 2] <- NA
```

Fig. 2b: Change of PHBR scores caused by mutations for HLA I. Dot color corresponds to PHBR fold change; the mutations that substantially (>3-fold) increase PHBR are signed. Sites that did not bind any of the patient's HLA alleles both in ancestral and derived states are not shown.

```
p_mutI <- ggplot(phbrI[phbrI$presented == "yes" & phbrI$mut != "ORF8:Q18*",], aes(x = phbr_or, y = phbr
  geom_abline(slope = 1, linetype="dashed", color = "grey10") +
  geom_point(aes(fill = rel), size = 6, shape = 21, color = "grey20", alpha = 0.9) +
  scale_x_log10(limits = c(0.005, 10), name = "PHBR in ancestral state") +
  scale_y_log10(limits = c(0.005, 10), name = "PHBR in derived state") +
  #scale_fill_gradientn(colours = pal) +
  scale_fill_distiller(name = "PHBR\nFold\nChange", palette = "RdYlBu", limits = c(0.19, 14)) +
  geom_text_repel(size = 6) +
  #ggtitle("HLA I") +
  theme_bw() +
   theme(legend.position = "none",
        axis.text.y = element_text(size = 18),
        axis.text.x = element_text(size = 18),
        legend.title = element_text (size = 18),
        legend.text = element_text(size = 18),
        axis.title = element_text (size = 18))
p_mutI
```

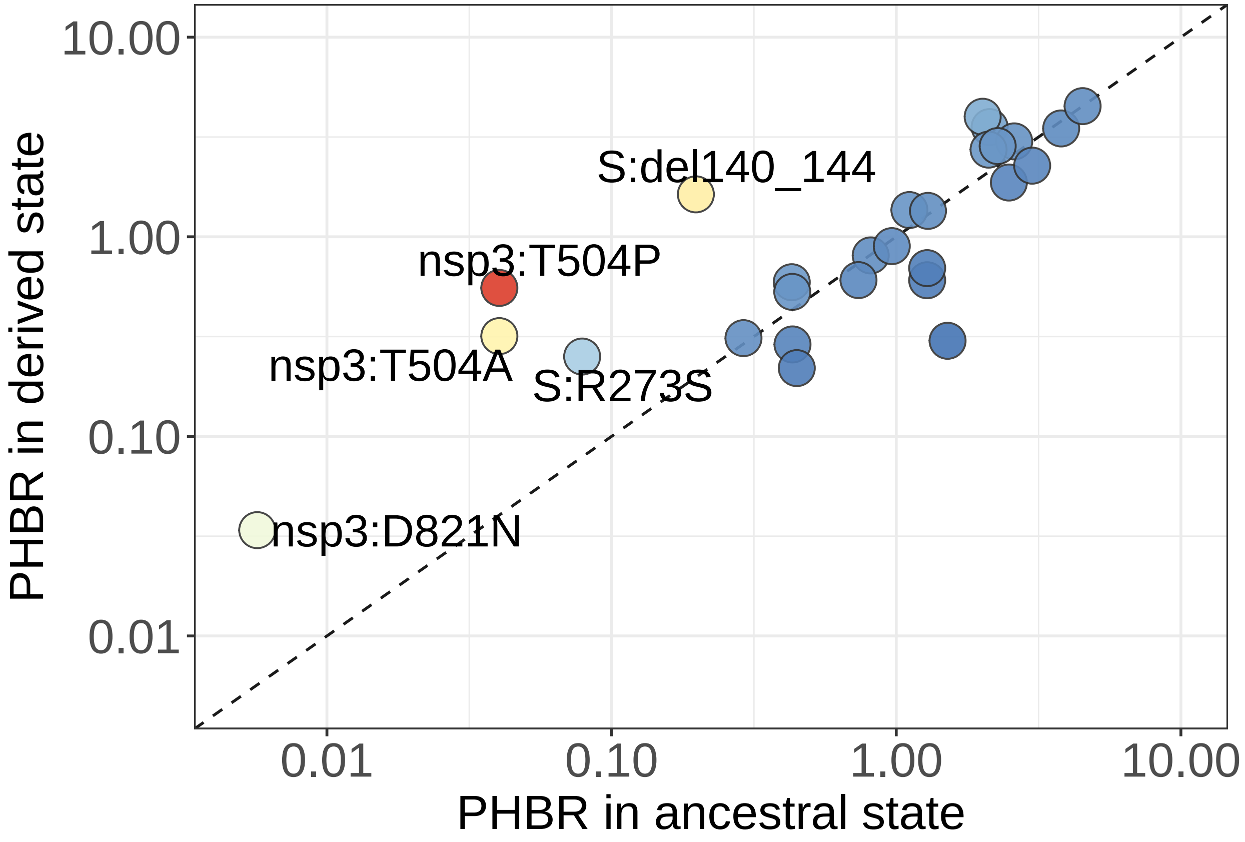
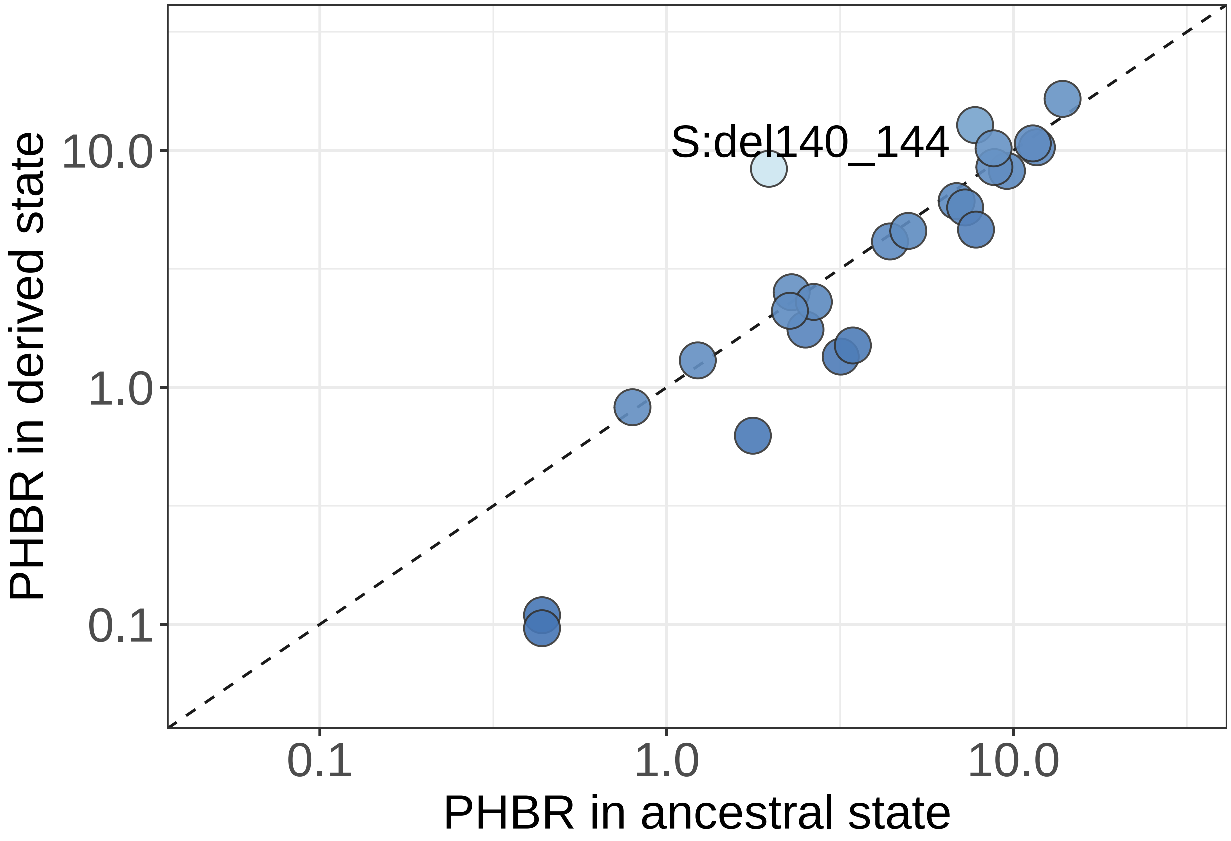


Fig. 2c: Change of PHBR scores caused by mutations for HLA II. Dot color corresponds to PHBR fold change; the mutations that substantially (>3-fold) increase PHBR are signed. Sites that did not bind any of the patient's HLA alleles both in ancestral and derived states are not shown.

```
p_mutII <- ggplot(phbrII[phbrII$presented == "yes" & phbrII$mut != "ORF8:Q18*",], aes(x = phbr_or, y = p
  geom_abline(slope = 1, linetype="dashed", color = "grey10") +
  geom_point(aes(fill = rel), size = 6, shape = 21, color = "grey20", alpha = 0.9) +
  scale_x_log10(limits = c(0.05, 30), name = "PHBR in ancestral state") +
  scale_y_log10(limits = c(0.05, 30), name = "PHBR in derived state") +
  scale_fill_distiller(name = "", palette = "RdYlBu", limits = c(0.19, 14)) +
  #ggtitle("HLA II") +
  geom_text_repel(size = 6) +
  theme_bw() +
   theme(legend.position = "none",
        axis.text.y = element_text(size = 18),
        axis.text.x = element_text(size = 18),
        legend.title = element_text (size = 18),
        legend.text = element_text(size = 18),
        axis.title = element_text (size = 18))
```

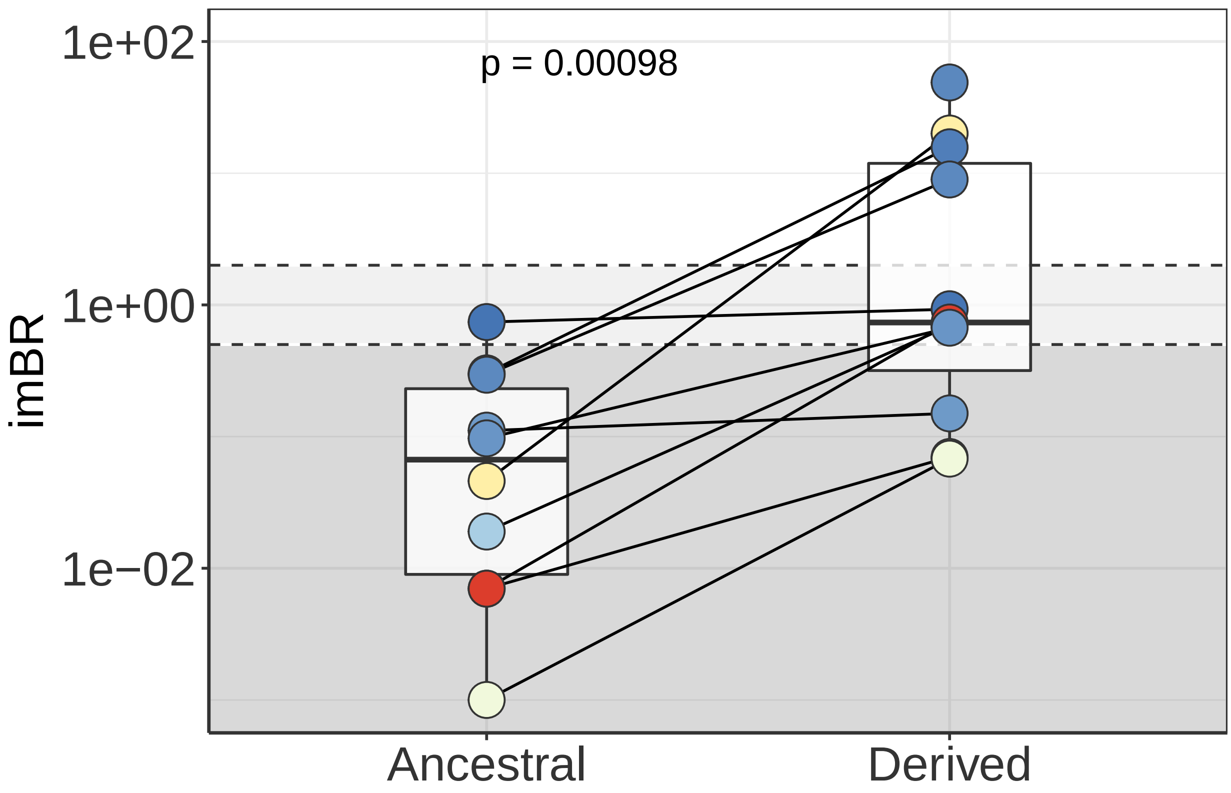
```
p_mutII
```



Next we focus on mutations, affected sites of known immunogenic epitopes.

Fig. 2d: Comparison of imBR scores for the mutated sites in their ancestral and derived states. The level of significance is calculated by the Wilcoxon sign-rank test.

```
foundI <- found[found$class == "HLAI",]
foundII <- found[found$class == "HLAII",]
exact <- c()

for (i in muts[muts != "ORF8:Q18*"]) {
  gg_i <- peptides_table_I[peptides_table_I$mut == i &
                              is.element(peptides_table_I$pept_or, foundI$peptide),]
  best_or <- c()
  best_mut <- c()
  for (j in 1:nrow(gg_i)){
    p_or <- gg_i$pept_or[j]
    p_mut <- gg_i$pept_mut[j]
    best_or <- c(best_or,
                 net_panI$Rank_EL[net_panI$peptide == p_or & net_panI$allele == foundI$allele[foundI$pep
    best_mut <- c(best_mut,
                 net_panI$Rank_EL[net_panI$peptide == p_mut & net_panI$allele == foundI$allele[foundI$pe
  }

  exact <- rbind(exact, c(i, gg_i$pept_or[which(best_or == min(best_or))], min(best_or), "or",
                          foundI$allele[foundI$peptide == gg_i$pept_or[which(best_or == min(best_or))]]
                c(i, gg_i$pept_mut[which(best_mut == min(best_mut))], min(best_mut), "mut",
                          foundI$allele[foundI$peptide == gg_i$pept_or[which(best_mut == min(best_mut))]
}
exact <-data.frame(exact)
colnames(exact) <- c("mut", "pept", "rank", "state", "allele")
exact$rank <- as.numeric(exact$rank)

exact$mut1 <- exact$mut
exact$mut1[exact$state == "or"] <- ""

exact$state <- factor(exact$state, levels = c("or", "mut"))

exact$mut[exact$mut == "S:del141-144"] <- "S:del140_144"

rel <- c()
for (i in 1:nrow(exact)){
  rel <- c(rel, phbrI$rel[phbrI$mut == exact$mut[i]])
}
exact$rel <- rel

p_exact_rank <- ggplot(exact[!is.na(exact$state),], aes(x = state, y = rank, fill = as.numeric(rel))) +
  annotate("rect", ymin=0, ymax=0.5, xmin = -Inf, xmax = Inf,
           alpha=0.3, fill = "grey50", color="white") +
  annotate("rect", ymin=0.5, ymax=2, xmin = -Inf, xmax = Inf,
           alpha=0.1, fill = "grey50", color="white") +
  geom_hline(yintercept = 2, linetype="dashed", color = "grey20") +
  geom_hline(yintercept = 0.5, linetype="dashed", color = "grey20") +
  geom_boxplot(draw_quantiles = c(0.5), width = 0.35,  color = "grey20", alpha = 0.8) +
  geom_line(aes(group = mut)) +
  geom_point(color = "grey20", size = 6, shape = 21, alpha = 1, position = position_dodge()) +
  scale_fill_distiller(name = "", palette = "RdYlBu", limits = c(0.19, 14)) +
  scale_y_continuous(name = "imBR", trans = "log10", limits = c(0.001, 100)) +
  scale_x_discrete(labels=c("or" = "Ancestral", "mut" = "Derived"), name = "") +
  annotate("text", x = 1.2, y =70, label = "p = 0.00098", size = 5) +
  guides(color = FALSE, size = FALSE, alpha = FALSE) +
  theme_bw() +
                   theme(legend.position = "none",
                   legend.background = element_rect(),
                   legend.title = element_text (colour="black", size = 18, face = "plain"),
                   legend.text = element_text(colour="black", size = 18, face = "plain"),
                   plot.title = element_text(colour="black", size = 18, face = "plain"),
                   axis.title = element_text (colour="black", size = 18, face = "plain"),
                   axis.text.y = element_text(colour="grey20", size = 18, face = "plain"),
                   axis.text.x = element_text(colour="grey20", size = 18, face = "plain"),
                   axis.line = element_line(colour="grey20", size = 0.6),
                   panel.background = element_rect(fill = "white", colour = "grey20"),
                   strip.text = element_text(size = 18))
p_exact_rank
```



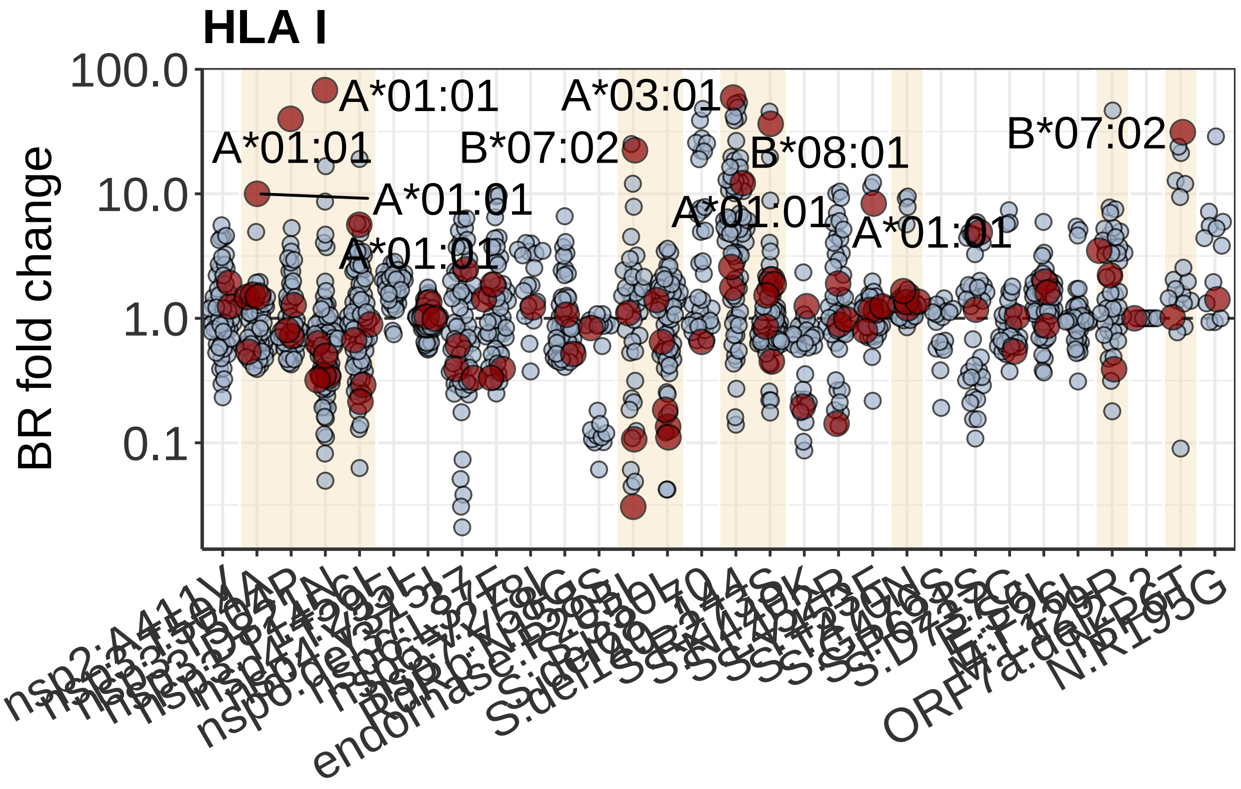
Part 3. Population-level effects of mutations, accumulated in patient S

In next chunks we compare BR fold changes among worldwild and patient-specific alleles, using previously produced **best_rank_alleles_I(II)** dataframes and show that immunogenic position have extreme fold change values on HLA I alleles of patient S.

Fig. 4a: The effect of each of the 30 mutations observed in SARS-CoV-2 of patient S on T cell immune escape, for each of the HLA I alleles carried by patient A (red) and frequent globally (grey). The mutations that change immunogenic peptides (for HLA I) according to IEDB are highlighted. Alleles that do not present the corresponding position in both ancestral and derived state are not shown. For the mutations that correspond to >5-fold increase in BR, the corresponding HLA alleles are signed.

```
best_rank_alleles_I %>%
  melt -> df
dfI <- cbind(df[df$state == "or",], df$value[df$state == "mut"])
colnames(dfI) <- c("mut", "state", "allele", "rank_or", "rank_mut")
dfI$dev <- dfI$rank_mut/dfI$rank_or
dfI <- dfI[dfI$rank_or <= 2 | dfI$rank_mut <= 2, ]

dfI$family <- str_extract(dfI$allele, "HLA-[A-C]*")
dfI$s <- as.numeric(is.element(dfI$allele, s_alleles_I))
```

```
dfI$s <- factor(dfI$s, levels = c("1", "0"))
dfI <- dfI[order(1/(as.numeric(dfI$s) + 1)), ]
dfI <- dfI[dfI$mut != "ORF7b:del2",]

immunogenic <- c("S:S50L", "S:del140_144", "S:R273S", "S:T470N", "nsp3:T504A", "nsp3:T504P", "nsp3:D821

dfI$mut <- factor(dfI$mut, levels = c("nsp2:A411V", "nsp3:T504A", "nsp3:T504P", "nsp3:D821N",
  "nsp3:T1456I", "nsp4:T295I", "nsp4:V315I", "nsp6:del37_37", "nsp6:L37F", "nsp6:V278I", "nsp7:V58G",
  "RdRp:N158S", "endornase:P205L", "S:S50L", "S:del68_70", "S:del140_144", "S:R273S", "S:N440K", "S:L45
  "S:Y453F", "S:T470N", "S:G476S", "S:P621S", "S:D737G", "E:S6L", "E:F26L", "M:L129R", "ORF7a:del2_2",
  "ORF8:Q18*",  "N:P6T", "N:R195G"))

dfI$hla_l <- substring(dfI$allele, 5, 11)
dfI$hla_l[dfI$s != "1" | dfI$dev <= 5] = NA
p_mI <- ggplot(dfI[dfI$mut != "ORF8:Q18*",], aes(x = mut, y = dev, fill = as.factor(s),
                                                  size = as.factor(s), label = hla_l)) +
  annotate("rect", xmin=1.5, xmax=5.5, ymin = 0, ymax = Inf,
           alpha=0.4, fill = "wheat", color="white") +
  annotate("rect", xmin=12.5, xmax=14.5, ymin = 0, ymax = Inf,
           alpha=0.4, fill = "wheat", color="white") +
   annotate("rect", xmin=15.5, xmax=17.5, ymin = 0, ymax = Inf,
           alpha=0.4, fill = "wheat", color="white") +
  annotate("rect", xmin=20.5, xmax=21.5, ymin = 0, ymax = Inf,
           alpha=0.4, fill = "wheat", color="white") +
  annotate("rect", xmin=26.5, xmax=27.5, ymin = 0, ymax = Inf,
           alpha=0.4, fill = "wheat", color="white") +
  annotate("rect", xmin=28.5, xmax=29.5, ymin = 0, ymax = Inf,
           alpha=0.4, fill = "wheat", color="white") +

  geom_hline(yintercept = 1, linetype="dashed", color = "grey20") +
  scale_y_continuous(trans = "log10", name = "BR fold change") +
  scale_x_discrete(name = "") +
  geom_quasirandom(shape = 21, color = "black", alpha = 0.7) +
  geom_text_repel(aes(label = hla_l), vjust = 0.5, size = 6) +
  scale_fill_manual(name = "Allele", values = c("0" = "lightsteelblue3", "1" = "darkred"),
                    labels = c("0" = "Other", "1" = "Patient S")) +
  scale_size_manual(values = c(4, 2.5)) +
  ggtitle("HLA I") +
  guides(size = FALSE,
        fill = guide_legend(override.aes = list(size = 4))) +
 theme_bw() +
                  theme(legend.position = "none",
                  legend.background = element_rect(),
                  legend.title = element_text (colour="black", size = 18, face = "plain"),
                  legend.text = element_text(colour="black", size = 18, face = "plain"),
                  plot.title = element_text(colour="black", size = 18, face = "bold"),
                  axis.title = element_text (colour="black", size = 18, face = "plain"),
                  axis.text.y = element_text(colour="grey20", size = 18, face = "plain"),
                  axis.text.x = element_text(colour="grey20", size = 18, face = "plain", angle = 30, v
                  axis.line = element_line(colour="grey20", size = 0.6),
                  panel.background = element_rect(fill = "white", colour = "grey20"),
                  strip.text = element_text(size = 18))
p_mI
```



We perform permutations (n=100000), randomly chosing a set of patients alleles of each class, to get the probability of how the effect is such extreme by chance.

Fig. 4b: Distribution of mean BPR fold changes among immunogenic positions for HLA I alleles, based on 10^5 random generations of individual allele composition; the red dashed line is the percentile corresponding to the allele composition of patient S.

```
av <- c()
n=100000
for (i in 1:n){
  hla_sample <- c()
  for (f in unique(dfI$family)){
    hla_sample <- c(hla_sample, as.character(sample(unique(dfI$allele[dfI$family == f]), 2)))
  }
  sample_av <- mean(dfI$dev[is.element(dfI$mut, immunogenic) & is.element(dfI$allele, hla_sample)])
  av <- c(av, sample_av)
}
real_av <- mean((dfI$dev[dfI$s == 1 & is.element(dfI$mut, immunogenic)]))
p_valueI <- 1 - sum(real_av >= av)/n

print(paste0("P-value, resulting from 10000 permutations = ", p_valueI))
```

```
## [1] "P-value, resulting from 10000 permutations = 0.00333000000000006"
```

```
av <- data.frame(av)

p_statI <- ggplot(av, aes(x = av)) +
  geom_density(alpha=0.4, fill = "wheat") +
  geom_vline(xintercept = real_av, linetype="dashed", color = "darkred") +
  scale_y_continuous("Density") +
  scale_x_continuous("Mean BR fold change") +
  ggtitle("Permutation test for HLA I") +
  annotate("text", x = 7, y=0.1, label = paste0("p=", sprintf("%s", p_valueI)), size = 6) +
  theme_bw() +
                theme(legend.position = "none",
                 legend.background = element_rect(),
                 legend.title = element_text (colour="black", size = 18, face = "plain"),
                 legend.text = element_text(colour="black", size = 18, face = "plain"),
                 plot.title = element_text(colour="black", size = 18, face = "bold"),
                 axis.title = element_text (colour="black", size = 18, face = "plain"),
                 axis.text.y = element_text(colour="grey20", size = 18, face = "plain"),
                 axis.text.x = element_text(colour="grey20", size = 18, face = "plain", angle = 30, v
                 axis.line = element_line(colour="grey20", size = 0.6),
                 panel.background = element_rect(fill = "white", colour = "grey20"),
                 strip.text = element_text(size = 18))
p_statI
```

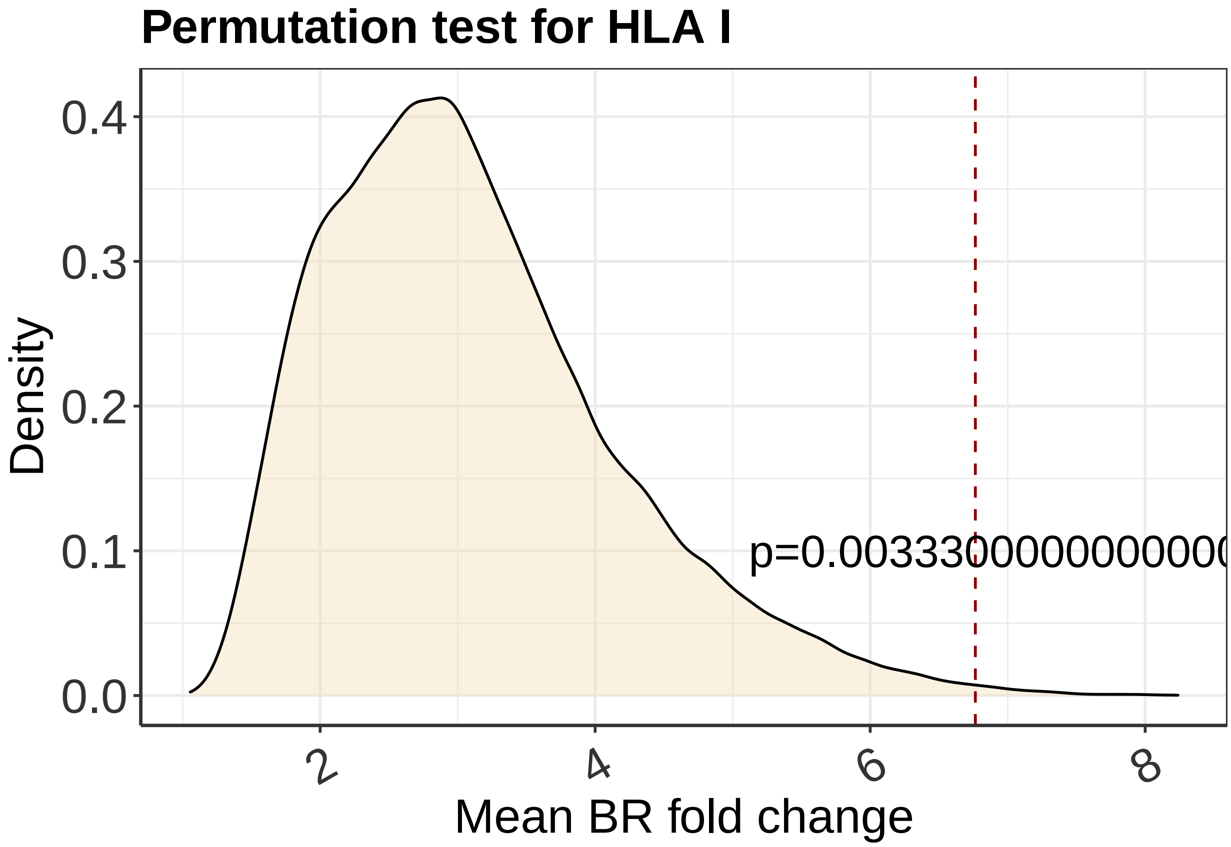


Fig. 4c: The effect of each of the 30 mutations observed in SARS-CoV-2 of patient S on T cell immune escape, for each of the HLA II alleles carried by patient A (red) and frequent globally (grey). The mutations that are adjacent to such peptides (for HLA II) according to IEDB are highlighted. Alleles that do not present the corresponding position in both ancestral and derived state are not shown. For the mutations that correspond to >5-fold increase in BR,

the corresponding HLA alleles are signed.

```
best_rank_alleles_II %>%
  melt -> df
dfII <- cbind(df[df$state == "or",], df$value[df$state == "mut"])
colnames(dfII) <- c("mut", "state", "allele", "rank_or", "rank_mut")
dfII$dev <- dfII$rank_mut/dfII$rank_or
dfII <- dfII[dfII$rank_or <= 10 | dfII$rank_mut <= 10, ]

dfII$family <- str_extract(dfII$allele, "D[A-Z]*")
dfII$s <- as.numeric(is.element(dfII$allele, s_alleles_II))
dfII$s <- factor(dfII$s, levels = c("1", "0"))
dfII <- dfII[order(1/(as.numeric(dfII$s) + 1)), ]
dfII <- dfII[dfII$mut != "ORF7b:del2",]
dfII$mut <- factor(dfII$mut, levels = c("nsp2:A411V", "nsp3:T504A", "nsp3:T504P", "nsp3:D821N",
  "nsp3:T1456I", "nsp4:T295I", "nsp4:V315I", "nsp6:del37_37", "nsp6:L37F", "nsp6:V278I", "nsp7:V58G",
  "RdRp:N158S", "endornase:P205L", "S:S50L", "S:del68_70", "S:del140_144", "S:R273S", "S:N440K", "S:L45
  "S:Y453F", "S:T470N", "S:G476S", "S:P621S", "S:D737G", "E:S6L", "E:F26L", "M:L129R", "ORF7a:del2_2",
  "ORF8:Q18*",  "N:P6T", "N:R195G"))

p_mII <- ggplot(dfII[dfII$mut != "ORF8:Q18*",], aes(x = mut, y = dev, fill = as.factor(s), size = as.fac
  annotate("rect", xmin=13.5, xmax=14.5, ymin = 0, ymax = Inf,
           alpha=0.4, fill = "wheat", color="white") +
  geom_hline(yintercept = 1, linetype="dashed", color = "grey20") +
  scale_y_continuous(trans = "log10", name = "BR fold change") +
  scale_x_discrete(name = "", position = "bottom") +
  geom_quasirandom(shape = 21, color = "black", alpha = 0.7) +
  scale_fill_manual(name = "Allele:", values = c("0" = "lightsteelblue3", "1" = "darkred"),
                    labels = c("0" = "Other", "1" = "Patient S")) +
  scale_size_manual(values = c(4, 2.5)) +
  ggtitle("HLA II") +
  guides(size = FALSE,
         fill = guide_legend(override.aes = list(size = 4))) +
  theme_bw() +
                   theme(legend.position = "none",
                   legend.background = element_rect(),
                   legend.title = element_text (colour="black", size = 18, face = "plain"),
                   legend.text = element_text(colour="black", size = 18, face = "plain"),
                   plot.title = element_text(colour="black", size = 18, face = "bold"),
                   axis.title = element_text (colour="black", size = 18, face = "plain"),
                   axis.text.y = element_text(colour="grey20", size = 18, face = "plain"),
                   axis.text.x = element_text(colour="grey20", size = 18, face = "plain", angle = 30, v
                   axis.line = element_line(colour="grey20", size = 0.6),
                   panel.background = element_rect(fill = "white", colour = "grey20"),
                   strip.text = element_text(size = 18))
p_mII
```

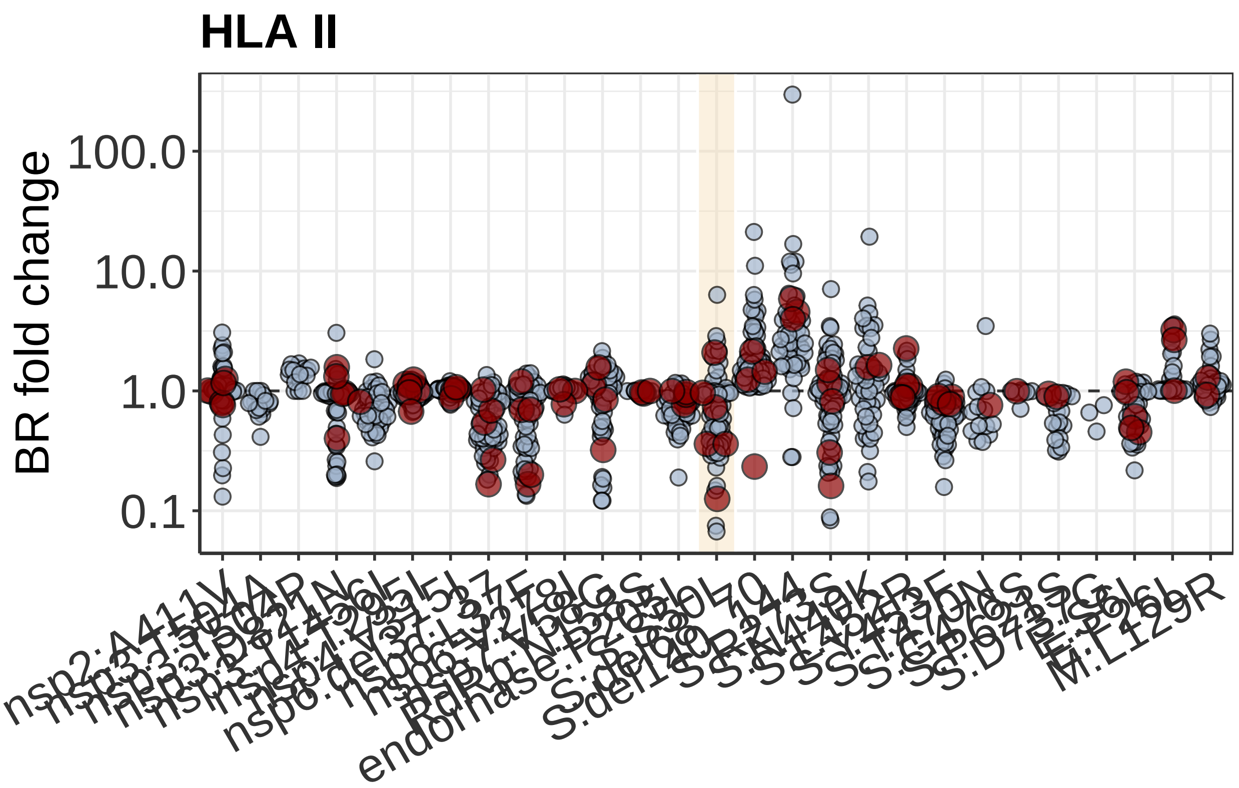


Fig. 4d: Distribution of mean BPR fold changes among immunogenic positions for HLA II alleles, based on 10^5 random generations of individual allele composition; the red dashed line is the percentile corresponding to the allele composition of patient S.

```
immunogenicII <- c("S:S50L")

avII <- c()
n=100000
for (i in 1:n){
  hla_sample <- c()
  for (f in unique(dfII$family)){
    hla_sample <- c(hla_sample, as.character(sample(unique(dfII$allele[dfII$family == f]), 2)))
  }
  sample_av <- mean(dfII$dev[is.element(dfII$mut, immunogenicII) & is.element(dfII$allele, hla_sample)]
  avII <- c(avII, sample_av)
}
real_avII <- mean((dfII$dev[dfII$s == 1 & is.element(dfII$mut, immunogenicII)]))
p_valueII <- 1 - sum(real_avII >= avII)/n

avII <- data.frame(avII)
print(paste0("P-value, resulting from 10000 permutations = ", p_valueII))
```

```
## [1] "P-value, resulting from 10000 permutations = 0.70136"
```

```
p_statII <- ggplot(avII, aes(x = avII)) +
  geom_density(alpha=0.4, fill = "wheat") +
  geom_vline(xintercept = real_avII, linetype="dashed", color = "darkred") +
  scale_y_continuous("Density") +
  scale_x_continuous("Mean BR fold change", limits = c(0, 8)) +
  ggtitle("Permutation test for HLA II") +
  annotate("text", x = 3.5, y=0.5, label = paste0("p=", sprintf("%s", p_valueII)), size = 6) +
  theme_bw() +
                   theme(legend.position = "none",
                   legend.background = element_rect(),
                   legend.title = element_text (colour="black", size = 18, face = "plain"),
                   legend.text = element_text(colour="black", size = 18, face = "plain"),
                   plot.title = element_text(colour="black", size = 18, face = "bold"),
                   axis.title = element_text (colour="black", size = 18, face = "plain"),
                   axis.text.y = element_text(colour="grey20", size = 18, face = "plain"),
                   axis.text.x = element_text(colour="grey20", size = 18, face = "plain", angle = 30, v
                   axis.line = element_line(colour="grey20", size = 0.6),
                   panel.background = element_rect(fill = "white", colour = "grey20"),
                   strip.text = element_text(size = 18))
p_statII
```

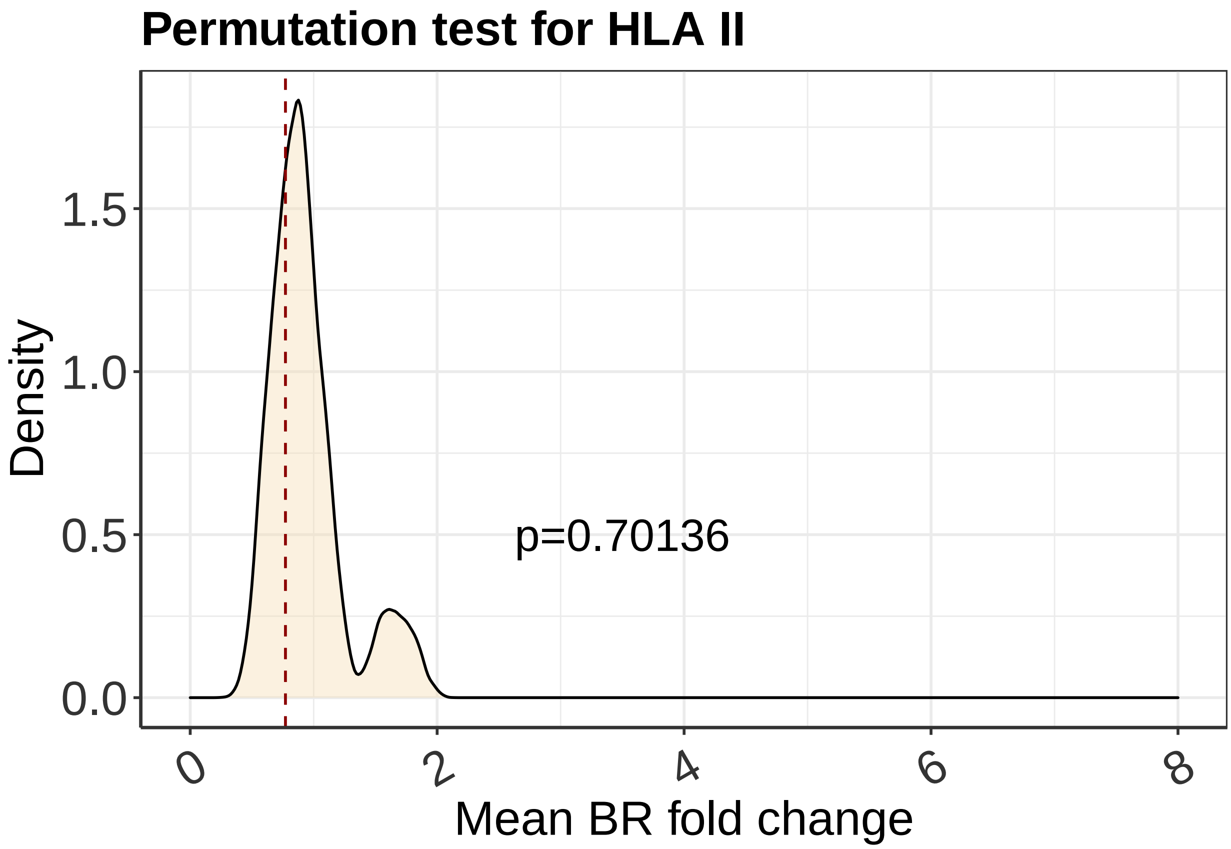


Fig. 4e: The sum effect of the amino acid changing mutations observed in SARS-CoV-2 of patient S on antigen presentation by the globally most frequent HLA class I. Alleles of patient S are in red.

```
df_pop <- c()
for (al in allelesI){
  or <- mean(dfI$rank_or[dfI$allele == al & dfI$mut != "ORF8:Q18*"])
  mut <- mean(dfI$rank_mut[dfI$allele == al & dfI$mut != "ORF8:Q18*"])
  freq <- hla_freq$freq[hla_freq$allele == al]
  if (!is.element(al, hla_freq$allele)){freq = 0.05}
  df_pop <- rbind(df_pop,
  c(al, or, mut, freq))
}

df_pop <- data.frame(df_pop)
colnames(df_pop) <- c("al", "or", "mut", "freq")
df_pop$s <- as.numeric(is.element(df_pop$al, s_alleles_I))
df_pop$s <- factor(df_pop$s, levels = c("1", "0"))
df_pop <- df_pop[order(1/(as.numeric(df_pop$s) + 1)), ]
df_pop[, 2:4] <- sapply(df_pop[,2:4], as.numeric)
df_pop$fam <- substring(df_pop$al, 1, 5)
df_pop$lab <- substring(df_pop$al, 7, 11)

p_dots_I <- ggplot(df_pop, aes(x = or, y = mut, fill = as.factor(s), label = lab)) +
  geom_abline(slope = 1, linetype="dashed", color = "grey10") +
  geom_point(aes(size = freq), shape = 21, color = "black", alpha = 0.7) +
  scale_x_log10(name = "Mean BR of ancestral", limits = c(0.3, 5)) +
  scale_y_log10(name = "Mean BR of derived", limits = c(0.3, 5)) +
  geom_text_repel(size=6, color = "grey10", face = "bold")+
  scale_fill_manual(name = "Allele", values = c("0" = "lightsteelblue4", "1" = "darkred"),
                    labels = c("0" = "Other", "1" = "Patient S")) +
  scale_size_continuous(name = "Population\nfrequency", range = c(2, 8)) +
  guides(size = guide_legend(override.aes = list(fill = "lightsteelblue4")),
         fill = guide_legend(override.aes = list(size = 4))) +
  theme_bw() +
                   theme(legend.position = "none",
                   legend.background = element_rect(),
                   legend.title = element_text (colour="black", size = 18, face = "plain"),
                   legend.text = element_text(colour="black", size = 18, face = "plain"),
                   plot.title = element_text(colour="black", size = 18, face = "bold"),
                   axis.title = element_text (colour="black", size = 18, face = "plain"),
                   axis.text.y = element_text(colour="grey20", size = 18, face = "plain"),
                   axis.text.x = element_text(colour="grey20", size = 18, face = "plain", angle = 0, vju
                   axis.line = element_line(colour="grey20", size = 0.6),
                   panel.background = element_rect(fill = "white", colour = "grey20"),
                   strip.text = element_text(size = 18))  +
  facet_wrap(~fam, ncol = 3)

p_dots_I
```

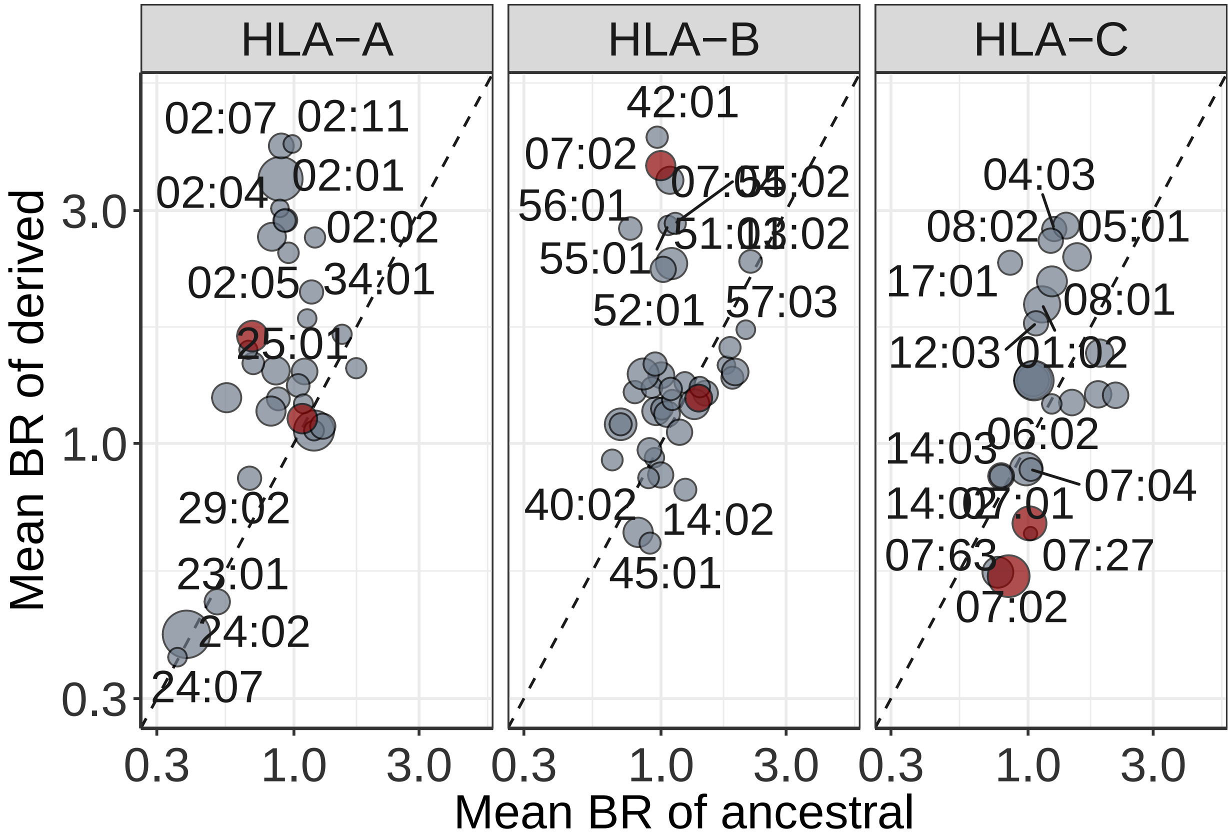
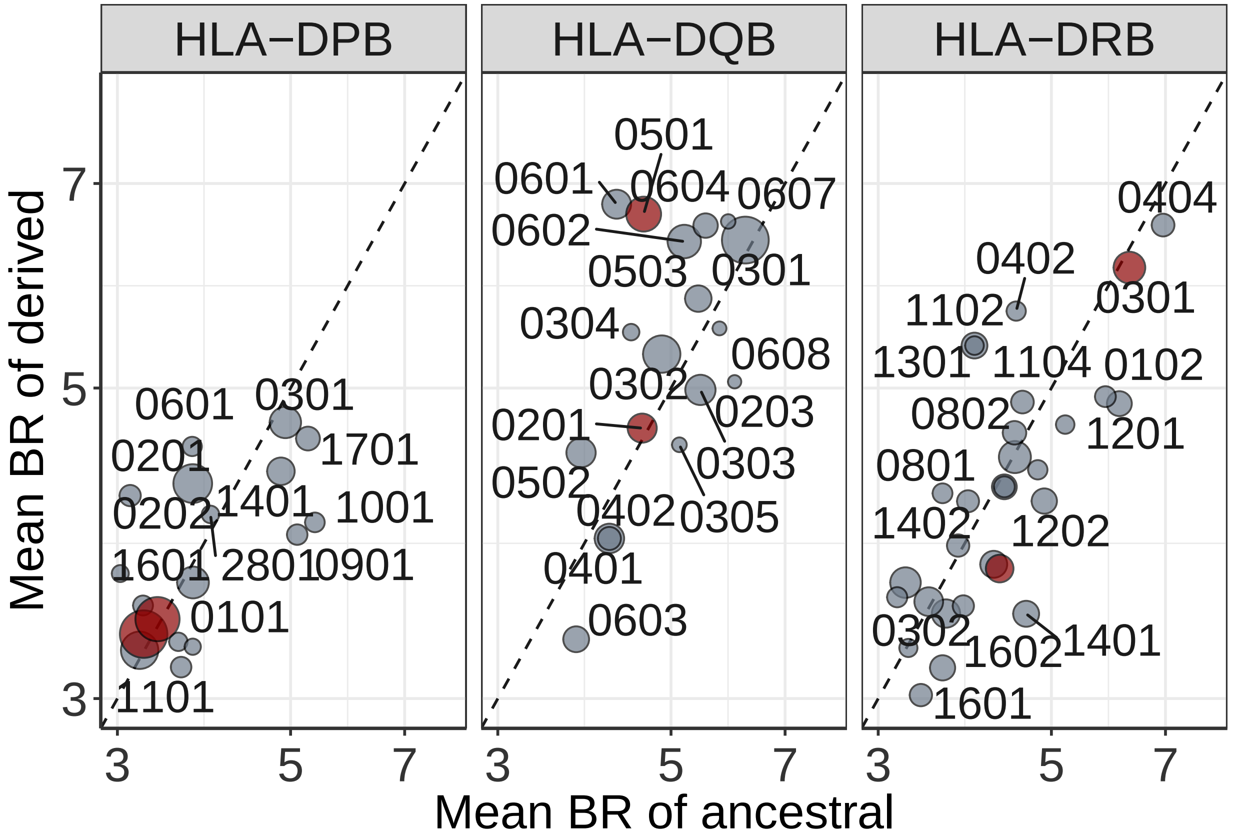


Fig. 4f: The sum effect of the amino acid changing mutations observed in SARS-CoV-2 of patient S on antigen presentation by the globally most frequent HLA class I. Alleles of patient S are in red.

```
df_pop2 <- c()
for (al in allelesII){
  or <- mean(dfII$rank_or[dfII$allele == al & dfII$mut != "ORF8:Q18*"])
  mut <- mean(dfII$rank_mut[dfII$allele == al & dfII$mut != "ORF8:Q18*"])
  freq <- hla_freq$freq[hla_freq$allele == al]
  if (!is.element(al, hla_freq$allele)){
    freq = 0.05
    df_pop2 <- rbind(df_pop2,
    c(al, or, mut, freq, "DQB1*0201", "HLA-DQB"))
    } else{
    df_pop2 <- rbind(df_pop2,
    c(al, or, mut, freq, hla_freq$short_name[hla_freq$allele == al], paste0("HLA-",
                        str_extract(hla_freq$short_name[hla_freq$allele == al], "D[A-Z]*"))))
  }
}

df_pop2 <- data.frame(df_pop2)
colnames(df_pop2) <- c("al", "or", "mut", "freq", "short_name", "fam")
df_pop2$s <- as.numeric(is.element(df_pop2$al, s_alleles_II))
df_pop2$s <- factor(df_pop2$s, levels = c("1", "0"))
df_pop2 <- df_pop2[order(1/(as.numeric(df_pop2$s) + 1)), ]
```

```
df_pop2[, 2:4] <- sapply(df_pop2[,2:4], as.numeric)
df_pop2$lab <- str_extract(df_pop2$short_name, "[0-9][0-9][0-9]*")


p_dots_II <- ggplot(df_pop2, aes(x = or, y = mut, fill = as.factor(s), label = lab)) +
  geom_abline(slope = 1, linetype="dashed", color = "grey10") +
  geom_point(aes(size = freq), shape = 21, color = "black", alpha = 0.7) +
  scale_x_log10(name = "Mean BR of ancestral", limits = c(3, 8)) +
  scale_y_log10(name = "Mean BR of derived", limits = c(3, 8)) +
  geom_text_repel(size=6, color = "grey10", face = "bold")+
  scale_fill_manual(name = "Allele", values = c("0" = "lightsteelblue4", "1" = "darkred"),
                    labels = c("0" = "Other", "1" = "Patient S")) +
  scale_size_continuous(name = "Population\nfrequency", range = c(2, 8)) +
  guides(size = guide_legend(override.aes = list(fill = "lightsteelblue4")),
         fill = FALSE) +
  theme_bw() +
                    theme(legend.position = "none",
                   legend.background = element_rect(),
                   legend.title = element_text (colour="black", size = 18, face = "plain"),
                   legend.text = element_text(colour="black", size = 18, face = "plain"),
                   plot.title = element_text(colour="black", size = 18, face = "bold"),
                   axis.title = element_text (colour="black", size = 18, face = "plain"),
                   axis.text.y = element_text(colour="grey20", size = 18, face = "plain"),
                   axis.text.x = element_text(colour="grey20", size = 18, face = "plain", angle = 0, vju
                   axis.line = element_line(colour="grey20", size = 0.6),
                   panel.background = element_rect(fill = "white", colour = "grey20"),
                   strip.text = element_text(size = 18))  +
  facet_wrap(~fam, ncol = 3)

p_dots_II
```


HLA−DPB
HLA−DQB
HLA−DRB
Mean BR of derived
Mean BR of ancestral
3
5
7
3
5
7
3
5
7
3
5
7
0601
0301
0201
1701
0202
1401
1001
1601
2801
0901
0101
1101
0501
0601
0604
0607
0602
0503
0301
0304
0608
0302
0203
0201
0303
0502
0402
0305
0401
0603
0404
0402
1102
0301
1301
1104
0102
0802
1201
0801
1402
1202
0302
1602
1401
1601