```r
library(data.table)
library(stringr)
library(dplyr)
library(RGenetics)
library(ape)
library(seqinr)
library(ggtree)
library(ggplot2)
library(ggbiplot)
library(ggbeeswarm)
library(ggpubr)
library(factoextra)
library(directlabels)
library(latex2exp)
```

Part 0. Ready to start

This is a code, used for analysis of B cell clonal lineages and visualisation of results in the study, named "Memory persistence and positively selected transition to antibody-secreting subsets in longitudinal lGH repertoires". To run this code we used R version 4.0.0 (2020-04-24) and the list of R packages of following version:

```r
packages <- data.frame(name = c("data.table", "stringr", "dplyr", "RGenetics", "ape", "seqinr", "ggtree
for (i in packages$name) {packages$version[packages$name == i] <- as.character(packageVersion(i))}
packages
```

```
##            name   version
## 1    data.table    1.13.0
## 2       stringr     1.4.0
## 3         dplyr     1.0.2
## 4     RGenetics       0.1
## 5           ape     5.4.1
## 6        seqinr     3.6.1
## 7        ggtree     2.2.4
## 8       ggplot2     3.3.2
## 9      ggbiplot      0.55
## 10   ggbeeswarm     0.6.0
## 11       ggpubr     0.4.0
## 12   factoextra     1.0.7
## 13 directlabels 2020.6.17
## 14    latex2exp     0.4.0
```

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

whole_reperotires - whole repertoires in the format of MiXCR output;

alignments - alignments of B-cell clonal lineages with the predicted MRCA and germline sequences in FASTA format;

trees - phylogenetic trees of B-cell lineages with the germline sequence as an outgroup in newick format;

G-MRCA - aligments of the germline sequence with MRCA in FASTA format;

timepoint.csv - correspondence of the number of sampling time points to real dates for eacg donor;

```
head(timepoints)
```

```
##   Patient Timepoint       Date
## 1     MRK       p01 2017-04-06
## 2     MRK       p02 2017-05-15
## 3     MRK       p03 2018-03-16
## 4     MRK       p04 2018-05-07
## 5      IM       p01 2017-04-06
## 6      IM       p02 2017-05-15
```

table_of_repeats.rds - the list of clonotypes with the same BCR sequence, but came from different time point / cell subsets / isotypes.

```
head(repeats)
```

```
##   pat   id                    sum_name
## 1  AT  526  489_5_isot_3_L_2017-05-26
## 2  AT  526 1475_5_isot_3_L_2017-04-10
## 3  AT  526 2273_5_isot_1_B_2018-03-23
## 4  AT 3440 5897_5_isot_3_B_2018-03-23
## 5  AT 3440 8397_5_isot_3_B_2017-04-10
## 6  AT 3440  703_5_isot_3_L_2017-04-10
##
## 1                                                                    557_1_isot_3_L_2017-04-10R489_1_isot_3_L_2
## 2                                                                  2697_1_isot_3_P_2017-04-10R1475_1_isot_3_L_2
## 3 1471_1_isot_1_B_2017-04-10R8133_1_isot_1_B_2017-05-26R5577_2_isot_3_B_2017-05-26R2273_1_isot_1_B_2
## 4                                                                  215_1_isot_3_L_2017-04-10R5897_1_isot_3_B_2
## 5                                                                  367_1_isot_3_P_2017-04-10R8397_1_isot_3_B_2
## 6                                                                    39_1_isot_3_P_2017-04-10R703_1_isot_3_L_2
##                   clone
## 1  AT_clone_526_size_32
## 2  AT_clone_526_size_32
## 3  AT_clone_526_size_32
## 4 AT_clone_3440_size_28
## 5 AT_clone_3440_size_28
## 6 AT_clone_3440_size_28
```

path - patient;

id - if of a clonal lineage;

sum_name reflects how BCR sequence of these clonotypesis named in the alignment and phylogenetic tree;

un_names shows indetificators of clonotypes, having this BCR sequnece, listed via R symbol;

clone - full name of a clonal lineage (== filename of corresponding alignment).

The name of clonotypes is coded as the following: **367_1_isot_3_P_2017-04-10**, where

367_ - identificator of the clonotype;

1_ - numer of technical replica (if 1 or 2), or the sign that is it a sum name for several clonotypes (if 5);

isot-3 - isotype (1 for IG M, 2 for Ig G and 3 for Ig A);

P_ - cell subset (B for Bmem, P for plasmablasts and L for plasma cells);

2017-04-10 - date of the sampling.

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

```
path <- "/home/jane/PhD_Skoltech/AF_work/Cell_fractions/draft_optimised/beauty_code/"
timepoints <- fread(paste0(path, "timepoints.csv"), data.table = FALSE)
repeats <- readRDS(paste0(path, "table_of_repeats.rds"))
```

Part 1. Temporal dynamics of clonal groups is associated with cell subset composition

(Results from Figure 3 and Supplementary Figure 4)

Analysis of lineage composition and persistence, PCA clustering of clonal lineages in HBmem and LBmem clusters

```
lineage_composition <- data.frame(file = list.files(paste0(path, "/alignments/"), pattern = "vj.fas"),
                      lineage_id = substring(str_extract(list.files(paste0(path, "/alignments/"),
                                                          pattern = "vj.fas"), "e_[0-9]*"), 3),
                      patient = str_extract(list.files(paste0(path, "/alignments/"), pattern = "vj.fas")

composition <- c()
for (p in unique(lineage_composition$patient)){
  df <- fread(paste0(path, "/whole_repertoires/", p, ".txt"), data.table = FALSE)
  df <- data.frame(df)
  df <- df[df$errorClone == FALSE,]
  ## remove singletons
  df <- df[df$cloneCount > 1,]

  ## clonal fractions in repertoire
  p1 <- nrow(df[df$timepoint == "p01",])
  p2 <- nrow(df[df$timepoint == "p02",])
  p3 <- nrow(df[df$timepoint == "p03",])

  ## pseudocounts
  ps_1 <- 1/p1
  ps_2 <- 1/p2
  ps_3 <- 1/p3

  for (cl in lineage_composition$lineage_id[lineage_composition$patient == p]){
    lineage <- df[df$clonalGroupId == cl & df$donor == p,]
    size <- nrow(lineage)
    size_un <- length(unique(lineage$nSeqVDJRegion))
    fr_time <- c(length(lineage$timepoint[lineage$timepoint == "p01"])/p1 + ps_1,
                 length(lineage$timepoint[lineage$timepoint == "p02"])/p2 + ps_2,
                 length(lineage$timepoint[lineage$timepoint == "p03"])/p3 + ps_3)

    sort_fr <- sort(fr_time)
    persistency <- 1/mean(c(sort_fr[3]/sort_fr[1], sort_fr[3]/sort_fr[2]))

    pbl <- length(lineage$timepoint[lineage$subpop == "PBL"])/size
    pl <- length(lineage$timepoint[lineage$subpop == "PL"])/size
    bmem <- 1 - pbl - pl
    isot_md <- length(lineage$timepoint[lineage$bestCHit == "IGHM*00" | lineage$bestCHit == "IGHD*00"])
```

```
    isot_a <-  length(lineage$timepoint[lineage$bestCHit == "IGHA1*00" | lineage$bestCHit == "IGHA2*00"]
    isot_g <- 1 - isot_md - isot_a
    composition <- rbind(composition, c(persistency, fr_time, bmem, pbl, pl, isot_md, isot_g, isot_a,
                                        size_un))
  }
}

lineage_composition <-cbind(lineage_composition, composition)
colnames(lineage_composition) <- c("file", "lineage_id", "patient", "pers", "freq_T1", "freq_T2", "freq_

lineage_composition[, c(1:14)] <- sapply(lineage_composition[, c(1:14)], as.character)
lineage_composition[, c(4:14)] <- sapply(lineage_composition[, c(4:14)], as.numeric)

# Compute k-means with k = 3
set.seed(123)
res.km <- kmeans(scale(lineage_composition[, 8:13]), 2, nstart = 25)
res.pca <- prcomp(lineage_composition[, 8:13],  scale = TRUE)
lineage_composition$cluster <- res.km$cluster
lineage_composition$cluster[lineage_composition$cluster == 1] <- "HBmem"
lineage_composition$cluster[lineage_composition$cluster == 2] <- "LBmem"
```

Themes of plots used for data visualisation:

```
my_theme <- theme_bw() +
                  theme(legend.position = "none",
                  legend.background = element_rect(),
                  legend.title = element_text (colour="black", size = 12, face = "plain"),
                  legend.text = element_text(colour="black", size = 12, face = "plain"),
                  plot.title = element_text(colour="black", size = 14, face = "bold"),
                  axis.title = element_text (colour="black", size = 12, face = "plain"),
                  axis.text = element_text(colour="grey20", size = 10, face = "plain"),
                  axis.line = element_line(colour="grey20", size = 0.6),
                  panel.background = element_rect(fill = "white", colour = "grey20"),
                  strip.text = element_text(size = 12))

my_colors <- list(scale_color_manual(values = c("HBmem" = "mediumpurple4",
                                                "LBmem" = "tan1")),
                  scale_fill_manual(values = c("HBmem" = "mediumpurple4",
                                               "LBmem" = "tan1")))

my_colors_isot <- list(scale_color_manual(values = c("Ig M" = "#D3DDDC",
                                                     "Ig G" = "#EBCC2A",
                                                     "Ig A" = "#E58601")),
                       scale_fill_manual(values = c("Ig M" = "#D3DDDC",
                                                    "Ig G" = "#EBCC2A",
                                                    "Ig A" = "#E58601")))


my_colors_fractions <- list(scale_color_manual(values = c("Bmem" = "#273046",
                                                          "PBL" = "#F8AFA8",
                                                          "PL" = "#CB2314")),
                            scale_fill_manual(values = c("Bmem" = "#273046",
```

```
                                                    "PBL" = "#F8AFA8",
                                                    "PL" = "#CB2314")))

my_shapes_patients <- list(scale_shape_manual(values = c(22, 21, 24, 23), name = "Patient"))

cluster_comparisons <- list(c("HBmem", "LBmem"))
```

Fig. 3A: Principal component analysis (PCA) of clonal group composition: proportions of B memory cells (Bmem), plasmablasts (PBL) and plasma cells (PL) as well as proportions of isotypes. The arrows represent the projections of the corresponding variables onto the two dimensional PCA plane, with lengths reflecting how well the variable explains the variance of the data. The two principal components (PC1 and PC2) cumulatively explain 90.1% of the variance;

```
p_pca <- ggbiplot(res.pca, groups = lineage_composition$cluster, varname.size = 5, ellipse = TRUE, elli
  geom_point(aes(shape=lineage_composition$patient,  fill = lineage_composition$cluster, alpha = lineag
             size = 3) +
  scale_x_continuous(name = "PC1 (73.4%)", limits = c(-3, 3.5)) +
  scale_y_continuous(name = "PC2 (16.7%)") +
  scale_alpha_manual(values = c("2" = 0.5, "1" = 0.5)) +
  my_shapes_patients +
  my_colors +
  my_theme +
  theme(legend.position = "none")
p_pca
```

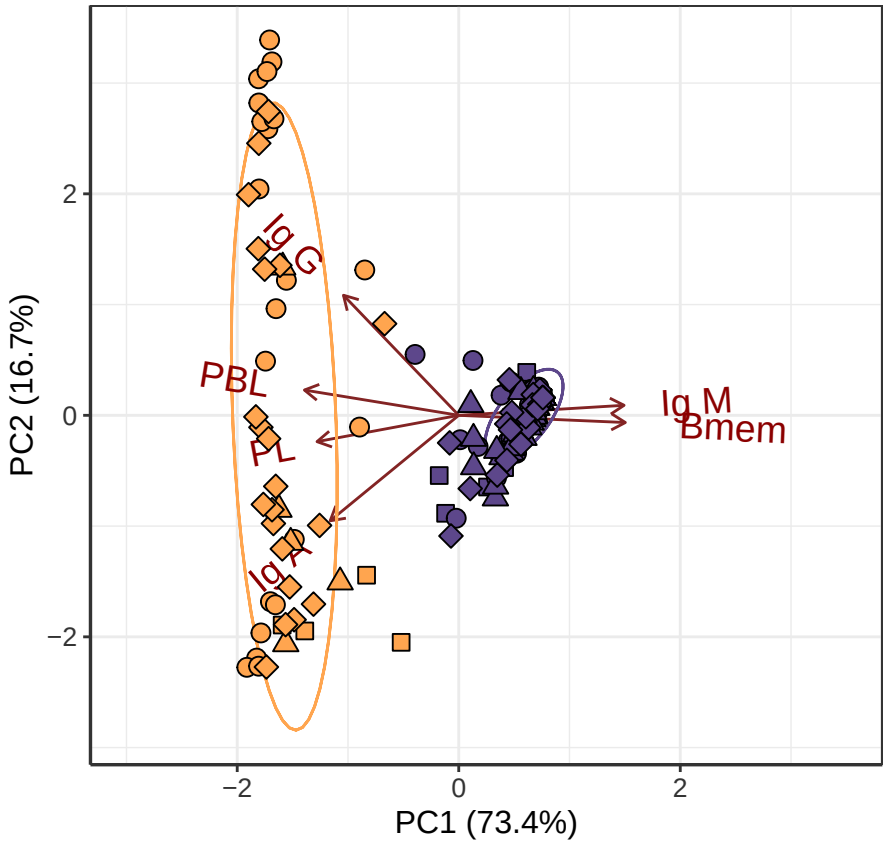


Fig. S4A: Scree plot for principal component analysi s from Fig. 3A of composition of clonal groups, where fractions of memory B cell, plasmablasts, plasma cells and fractions of IGHM, IGHG and IGHA were used as variables;

```
p_scree <- fviz_eig(res.pca, addlabels=TRUE, barfill = "grey70", barcolor = "grey20") +
  ggtitle("") +
  ylim(0,80) + my_theme
p_scree
```

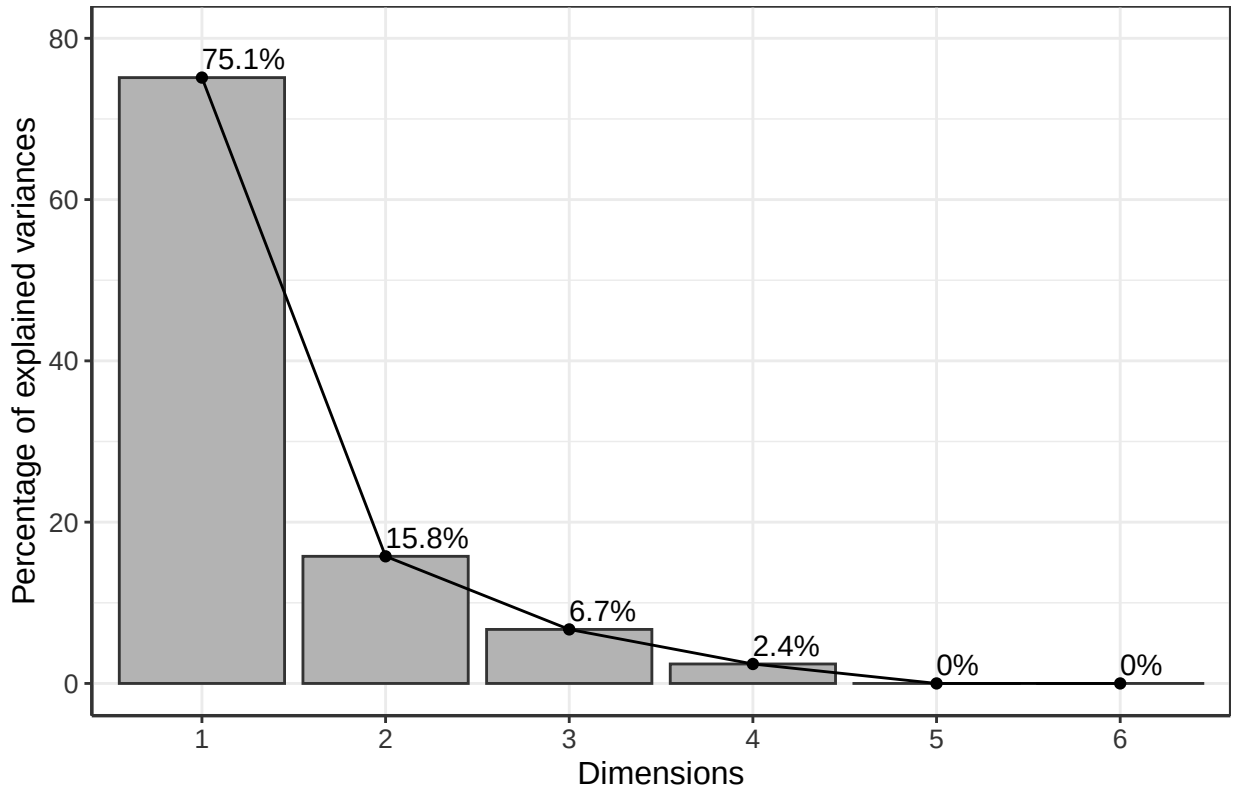


Fig. S4C: The number of clonal groups belonging to HBmem or LBmem clusters in each donor;

```
p_n <- ggplot(plyr::count(lineage_composition[,c(3,15)]), aes(x = patient, y = freq, fill = cluster)) +
  geom_bar(stat="identity", position=position_dodge(), color = "grey10")+
  geom_text(aes(label=freq), vjust=-0.3, size=3.5,  position = position_dodge(0.9))+
  scale_y_continuous(name=("# of clonal groups")) +
  scale_x_discrete(name = "Patient") +
  my_colors +
  my_theme +
  theme(legend.position = "none")
p_n
```

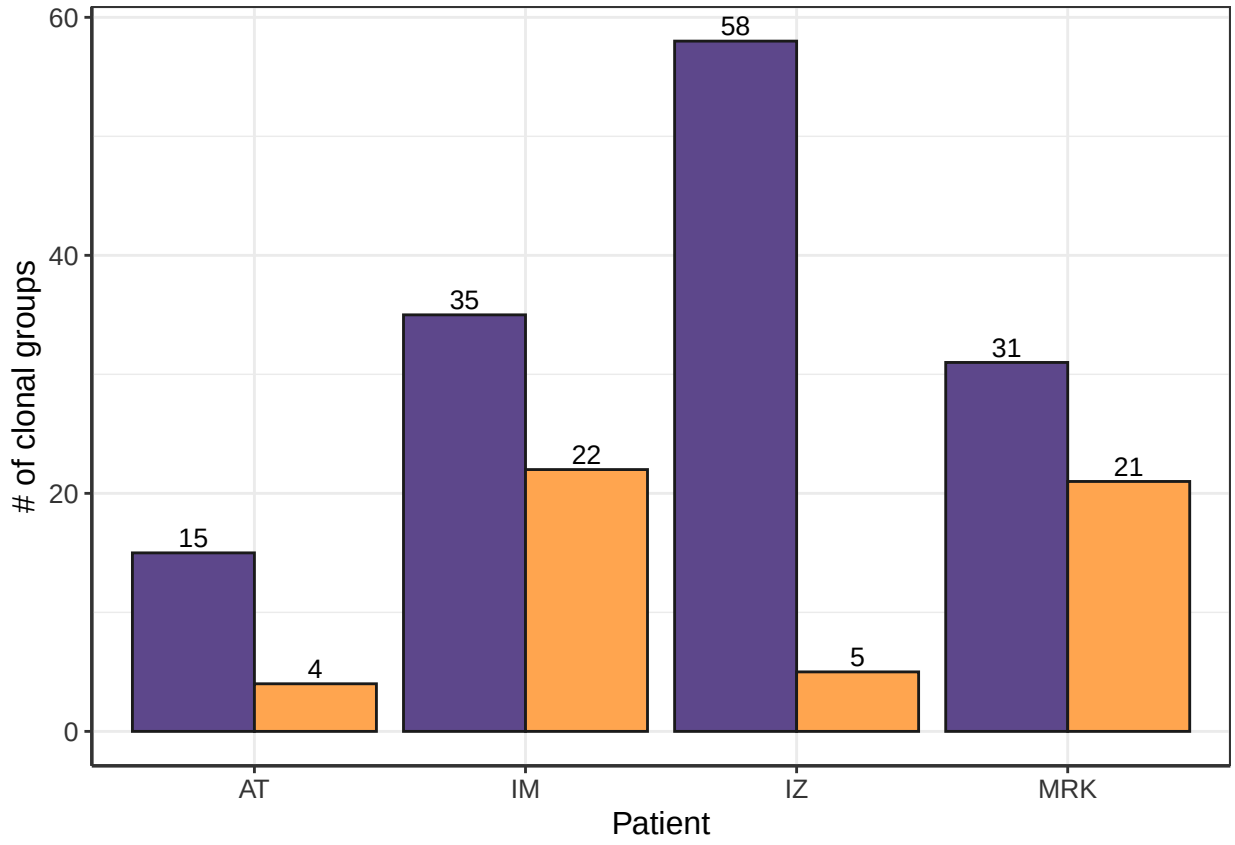


Fig. 3B: Proportion of clonotypes of a certain cell subset or isotype for clonal groups falling into HBmem or LBmem clusters; Statistical significance is the following: * - $p \leq 0.05$, ** - $p \leq 0.01$, * - $p \leq 10^{-3}$, **** - $p \leq 10^{-4}$.**

```
p_comp <- lineage_composition %>% select(Bmem, PBL, PL, 'Ig M', 'Ig G', 'Ig A', cluster, patient) %>%  r
  ggplot(aes(x = cluster, y = value, fill = cluster)) +
  geom_quasirandom(width=0.3, size = 1, color = "grey30", alpha = 0.8) +
  geom_boxplot(alpha = 0.8, outlier.colour = NA) +
  stat_compare_means(comparisons = cluster_comparisons, method = "wilcox.test", size = 4, label = "p.sig
  scale_y_continuous(name=("Fraction of the clonal group"), limits = c(0, 1.15)) +
  scale_x_discrete(name = "Cluster") +
  my_colors +
  my_theme +
  theme(legend.position = "none") +
  facet_wrap(~variable)

p_comp
```

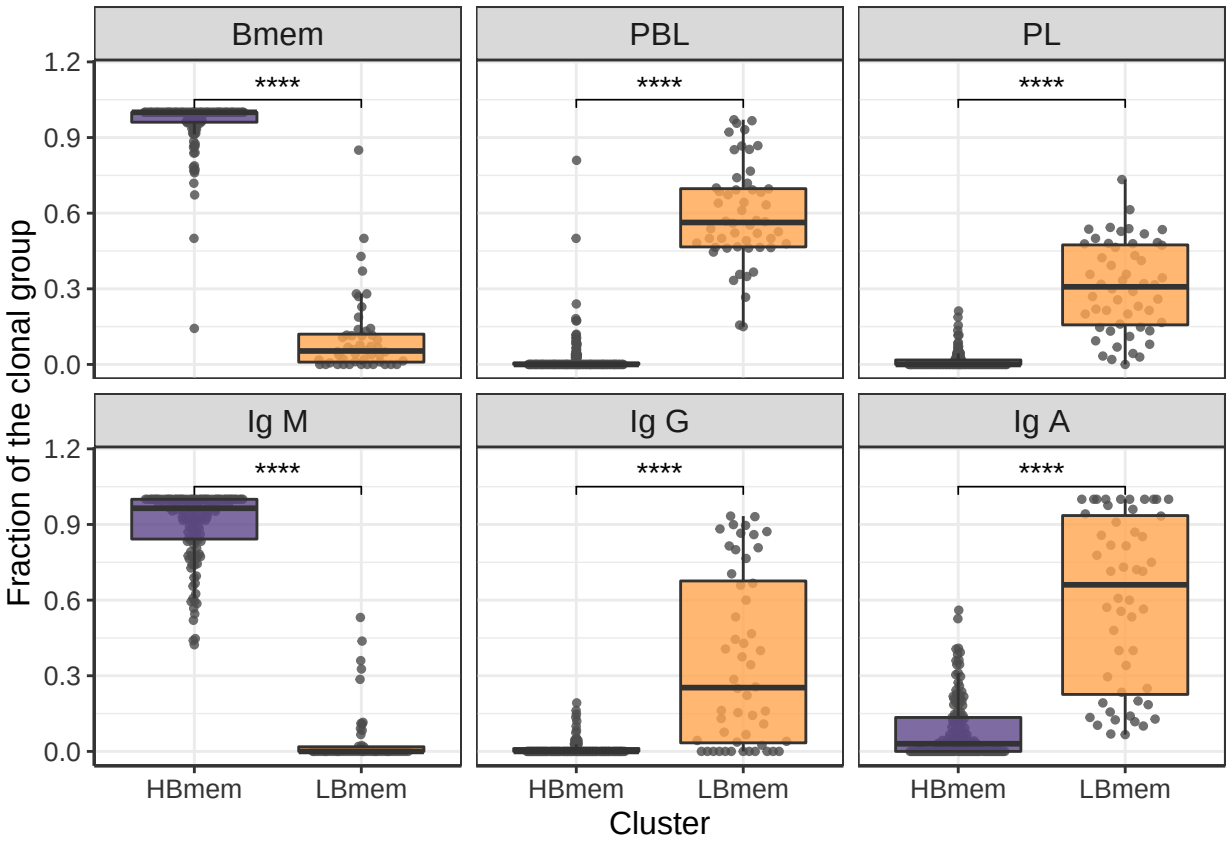


Fig. 3C: Dynamics of clonal group frequency, defined as the number of clonotypes in a group divided by the total number of clonotypes detected at this time point, for HBmem and LBmem clonal groups. Each line connecting the points represents unique clonal group (n=191);

```
lineage_composition %>%
  select(file, freq_T1, freq_T2, freq_T3, cluster) %>%
  melt %>%
  ggplot(aes(x = as.factor(variable), y = value, color = cluster, group = file)) +
  geom_point() +
  geom_line(alpha = 0.6, size = 0.7) +
  scale_x_discrete(name = "Time Point", labels = c("freq_T1" = "T1", "freq_T2" = "T2", "freq_T3" = "T3"
  scale_y_continuous(trans = "log10", name = "Clonal group frequency") +
  my_colors +
  my_theme -> p_traj
p_traj
```

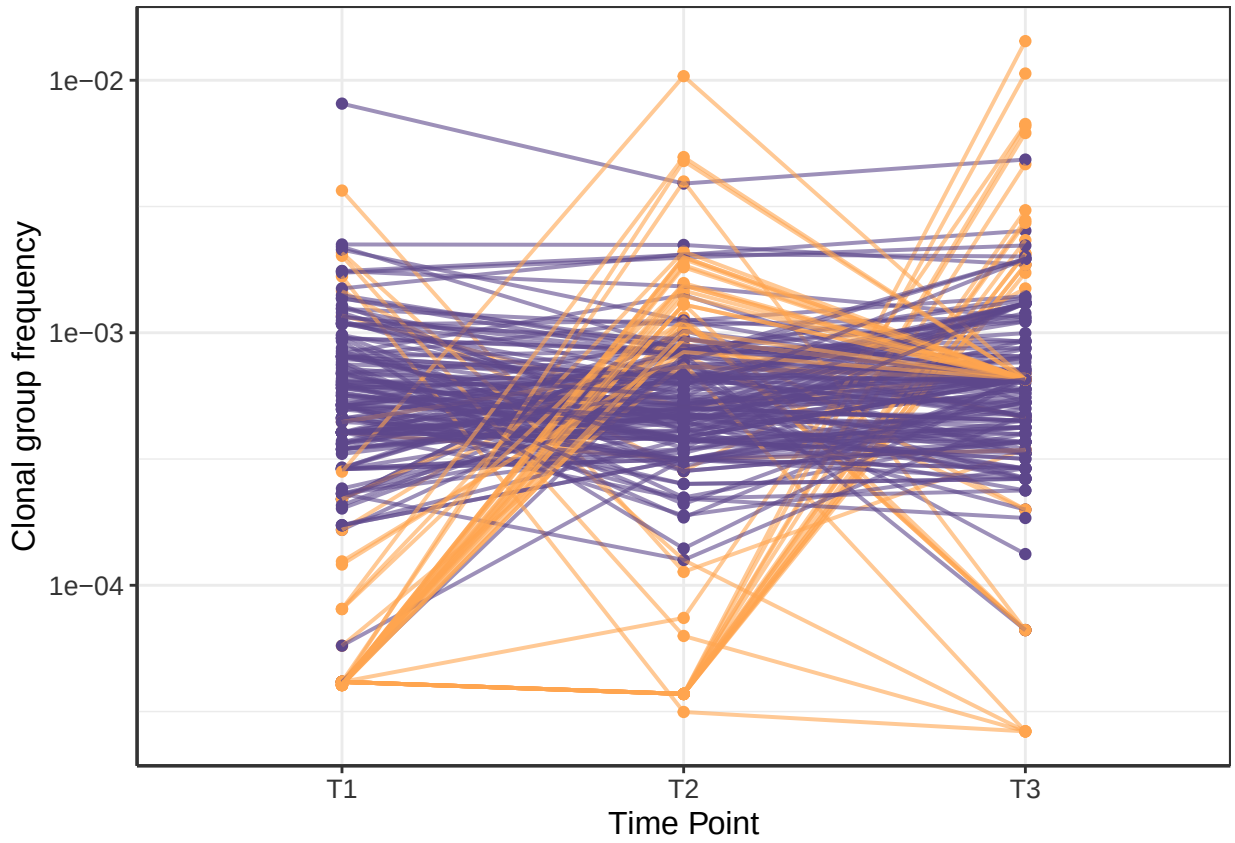


Fig. 3D: A schematic representation of calculation of clonal group persistence. f_{max} is the maximum clonal group frequency among the three time points, and f_{ij} are frequencies at the remaining two timepoints;

```
hb_clone <- lineage_composition$lineage_id[lineage_composition$pers > 0.9][5]
lb_clone <- lineage_composition$lineage_id[lineage_composition$pers < 0.3][3]

df <- rbind(unname(cbind(c("T1", "T2", "T3"), rep("HBmem", 3),
                         as.character(lineage_composition[lineage_composition$lineage_id == hb_clone, 5
                    as.character(round(lineage_composition[lineage_composition$lineage_id == hb_clone, 4
            unname(cbind(c("T1", "T2", "T3"), rep("LBmem", 3),
                         as.character(lineage_composition[lineage_composition$lineage_id == lb_clone, 5
                    as.character(round(lineage_composition[lineage_composition$lineage_id == lb_clone, 4
df <- data.frame(df)
colnames(df) <- c("point", "group", "freq", "pers")
df$group <- as.factor(df$group)
df$pers <- paste0("P = ", df$pers)

fancy_scientific <- function(l) {
     # turn in to character string in scientific notation
     l <- format(l, scientific = TRUE)
     # quote the part before the exponent to keep all the digits
     l <- gsub("^(.*)e", "'\\1'e", l)
     # turn the 'e+' into plotmath format
     l <- gsub("e", "%*%10^", l)
```

```
    # return this as an expression
    parse(text=l)
}

p_eq <- ggplot(df, aes(x = point, y = as.numeric(as.character(freq)), color = group, group = group)) +
  geom_point(size = 5, alpha = 0.9) +
  geom_line(size = 1.5, alpha = 0.9) +
  geom_dl(aes(label = pers), method = list(dl.trans(x = x + 1, y = y + 0.7), "first.points"), size = 12
  geom_text(aes(2.5,0.0007, label=("P*\'=\'~frac(1,frac(1,2)(frac(f[max],f[i])+frac(f[max],f[j])))")),
            parse = TRUE, size = 7, color = "grey10") +
  geom_text(aes(2.3,0.00095, label=("f[max]")), parse = TRUE, size = 6, color = "grey10") +
  geom_text(aes(1.2,0.00015, label=("f[i]")), parse = TRUE, size = 6, color = "grey10") +
  geom_text(aes(3.2,0.0003, label=("f[j]")), parse = TRUE, size = 6, color = "grey10") +
  scale_y_continuous(name=("Clonal group frequency"), labels=fancy_scientific) +
  scale_x_discrete(limits = c("T1", "T2","T3"), expand=c(0.2, 0.4), name = "Time point") +
  my_colors +
  my_theme

p_eq
```

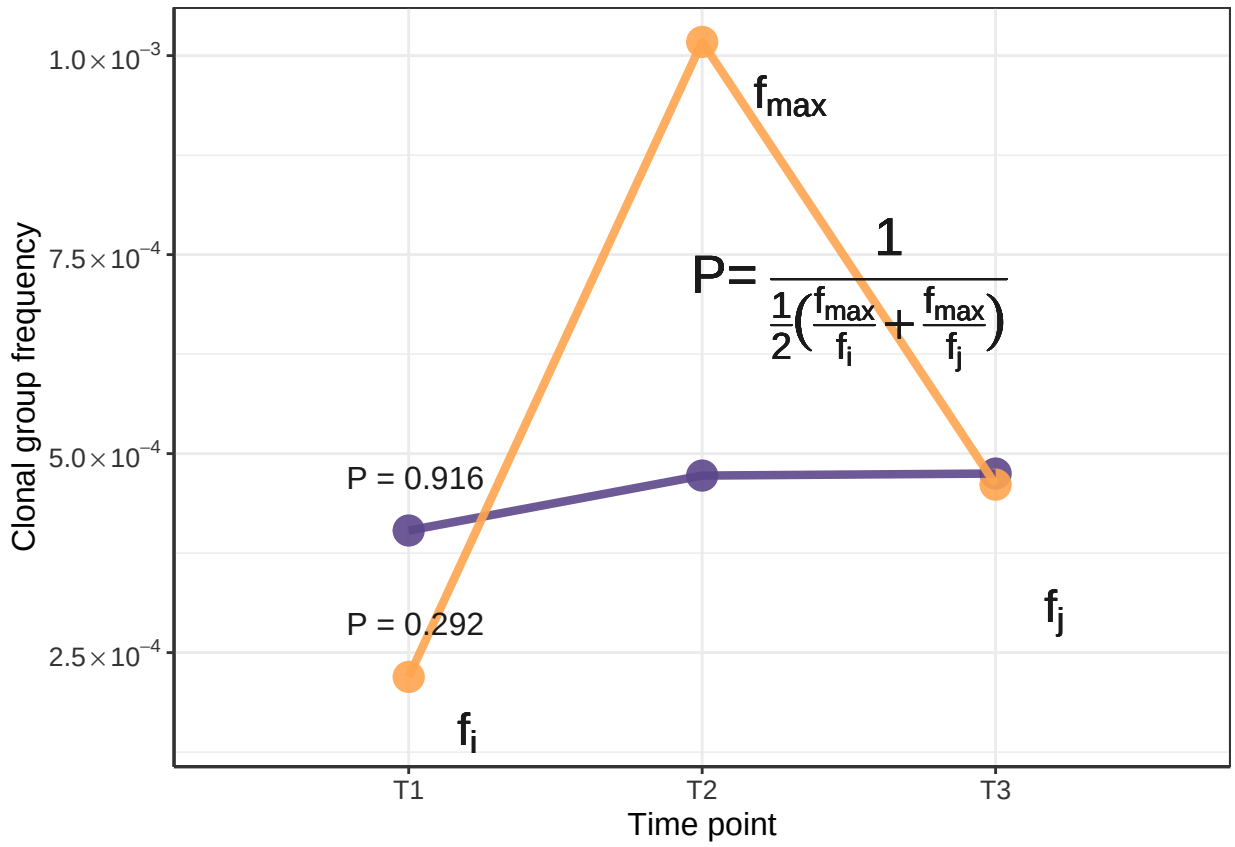


Fig. 3F: Comparison of persistence between HBmem and LBmem clonal groups. Statistical significance is calculated by Mann-Whitney test, notation is the following: * - $p \leq 0.05$, ** - $p \leq 0.01$, * - $p \leq 10^{-3}$, **** - $p \leq 10^{-4}$.**

```r
p_pers <- ggplot(lineage_composition, aes(x = cluster, y = pers, fill = cluster)) +
  geom_quasirandom(width=0.3, size = 1, color = "grey30", alpha = 0.8) +
  geom_boxplot(alpha = 0.8, outlier.colour = NA) +
  stat_compare_means(comparisons = cluster_comparisons, method = "wilcox.test", size = 4, label = "p.sig
  scale_y_log10(name=("Persistence"), limits = c(0.001, 2)) +
  scale_x_discrete(name = "Cluster") +
  my_colors +
  my_theme

p_pers
```

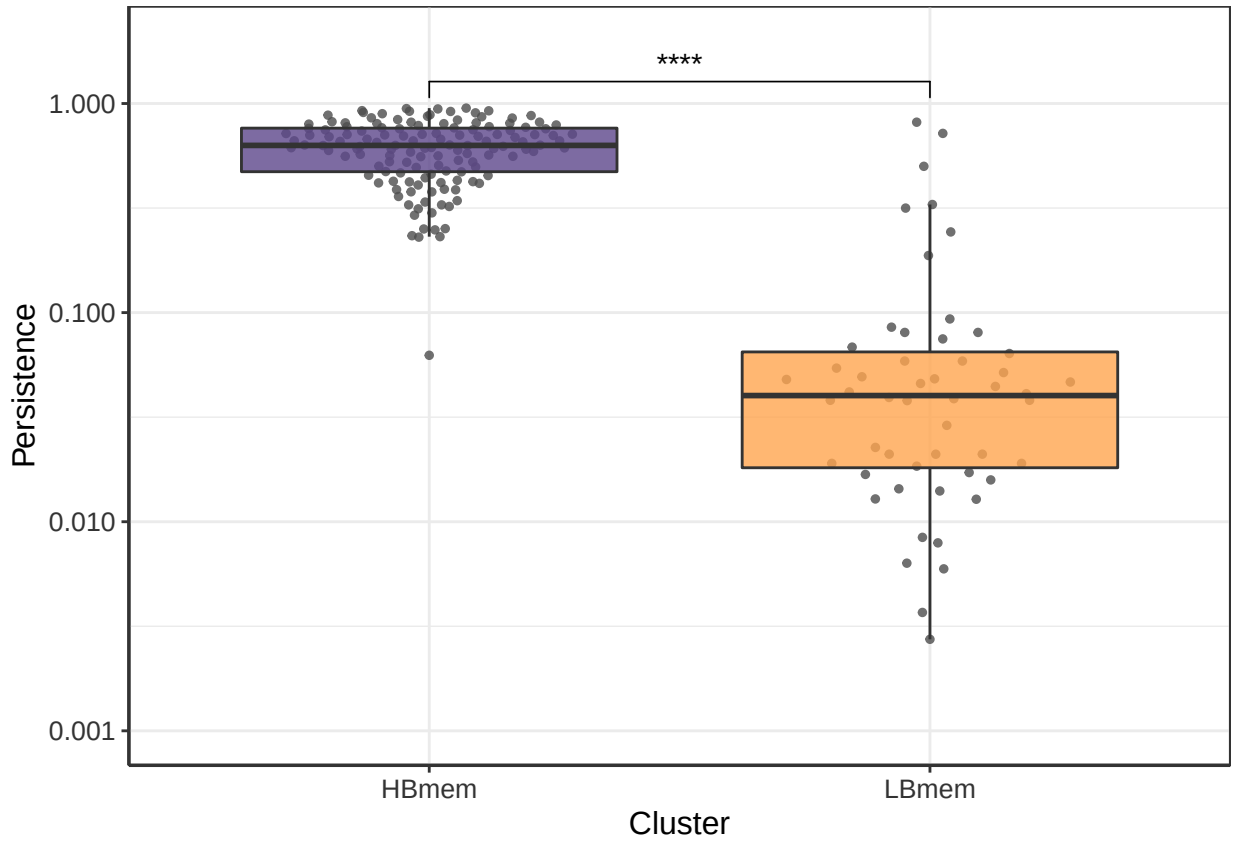


Fig. 3E: Spearman's correlation between persistence of the clonal group and fractions of its clonotypes attributed to the different B cell subpopulations or with a particular isotype; Statistical significance is the following: * - $p \leq 0.05$, ** - $p \leq 0.01$, * - $p \leq 10^{-3}$, **** - $p \leq 10^{-4}$.**

```r
df_x <- lineage_composition %>% select(patient, Bmem, PBL, PL, 'Ig M', 'Ig G', 'Ig A', cluster) %>% mel
df_x$pers <- rep(lineage_composition$pers, 6)

p_corr <- ggplot(df_x, aes(x = value, y = pers)) +
  geom_point(aes(fill = cluster, shape = patient, alpha = cluster), size = 3,  color = "black") +
  geom_smooth(method = 'lm', color = "grey20", fill = "grey40") +
  my_shapes_patients +
  stat_cor(method="spearman", size = 4, label.x.npc = 0.1, label.y.npc = 0.1,
           aes(label = paste(..r.label.., cut(..p.., breaks = c(-Inf, 0.0001, 0.001, 0.01, 0.05, Inf),
```

```
                                        labels = c("'****'", "'***'", "'**'", "'*'", "'ns'")),
                          sep = "~"))) +
  my_colors +
  scale_y_continuous(trans = "log10", name=("Persistence")) +
  scale_x_continuous(name = "Fraction of clonal group") +
  scale_alpha_manual(values = c("2" = 0.7, "1" = 0.5)) +
  my_theme +
  facet_wrap(~variable, ncol = 2)

p_corr
```

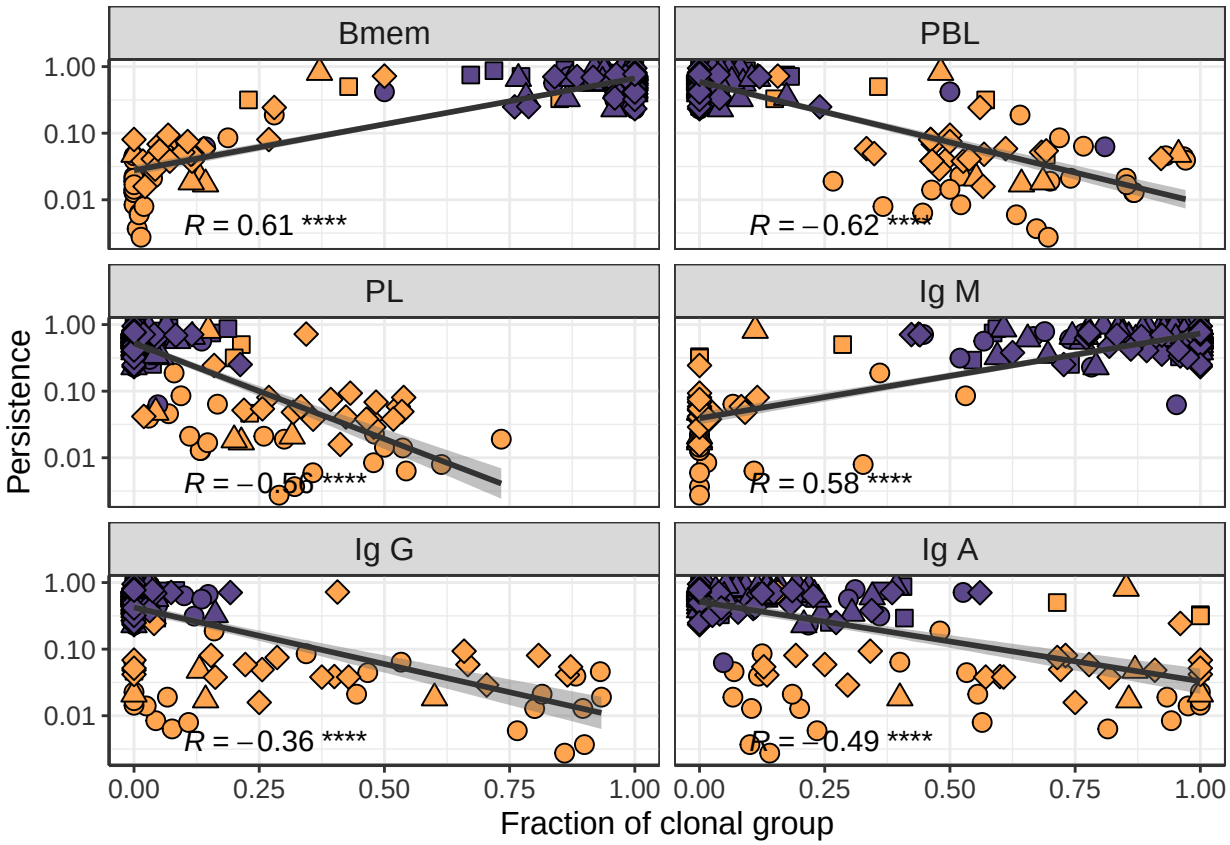


Fig. S4B: Distribution of sizes, i.e. the number of unique clonotypes in a group, for HBmem and LBmem clonal groups; Statistical significance, calculated by Mann-Whitney test, is the following: * - $p \leq 0.05$, ** - $p \leq 0.01$, * - $p \leq 10^{-3}$, **** - $p \leq 10^{-4}$.**

```
p_size <- ggplot(lineage_composition, aes(x = cluster, y = size, fill = cluster)) +
  geom_quasirandom(width=0.3, size = 1, color = "grey30", alpha = 0.8) +
  geom_boxplot(alpha = 0.8, outlier.colour = NA) +
  stat_compare_means(comparisons = cluster_comparisons, method = "wilcox.test", size = 4, label = "p.sig
  scale_y_log10(name=("Size")) +
  scale_x_discrete(labels = c(name = "")) +
  my_colors +
  my_theme

p_size
```

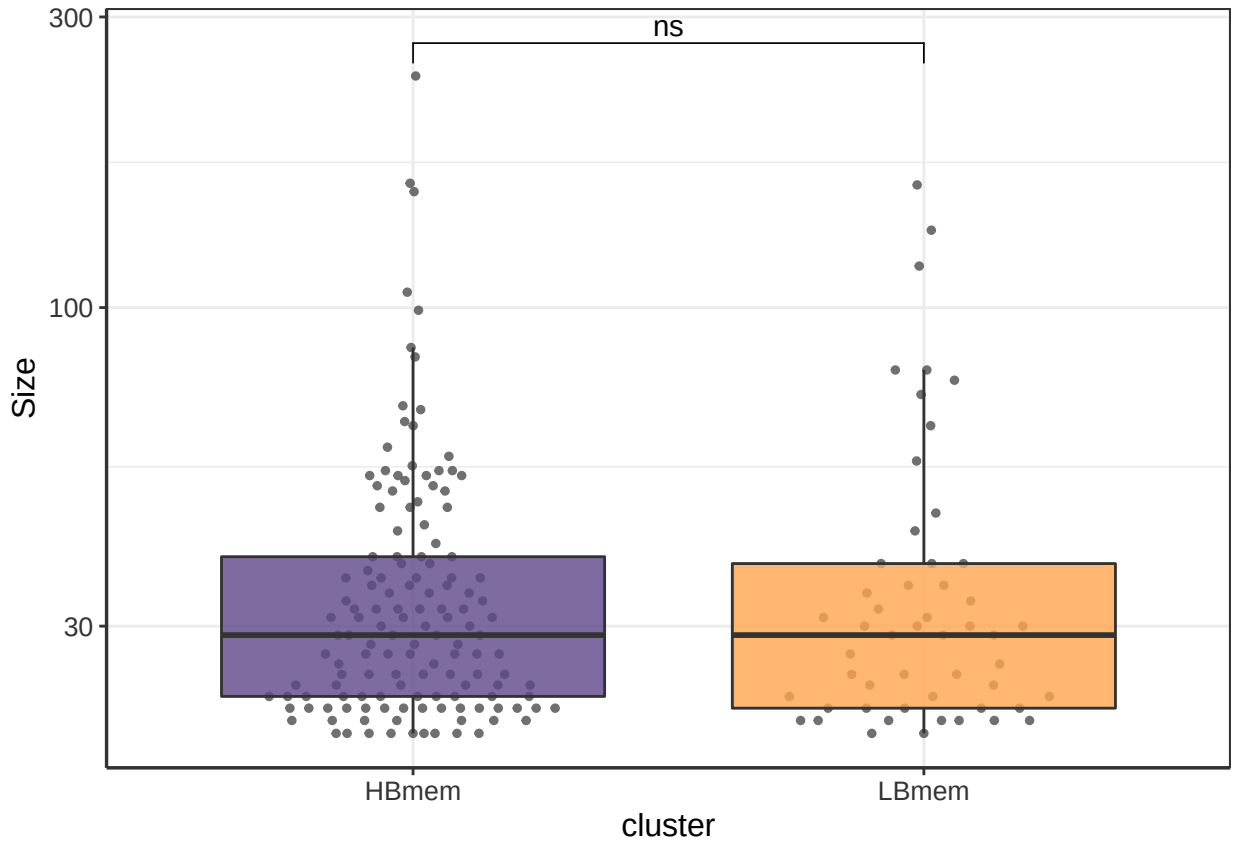
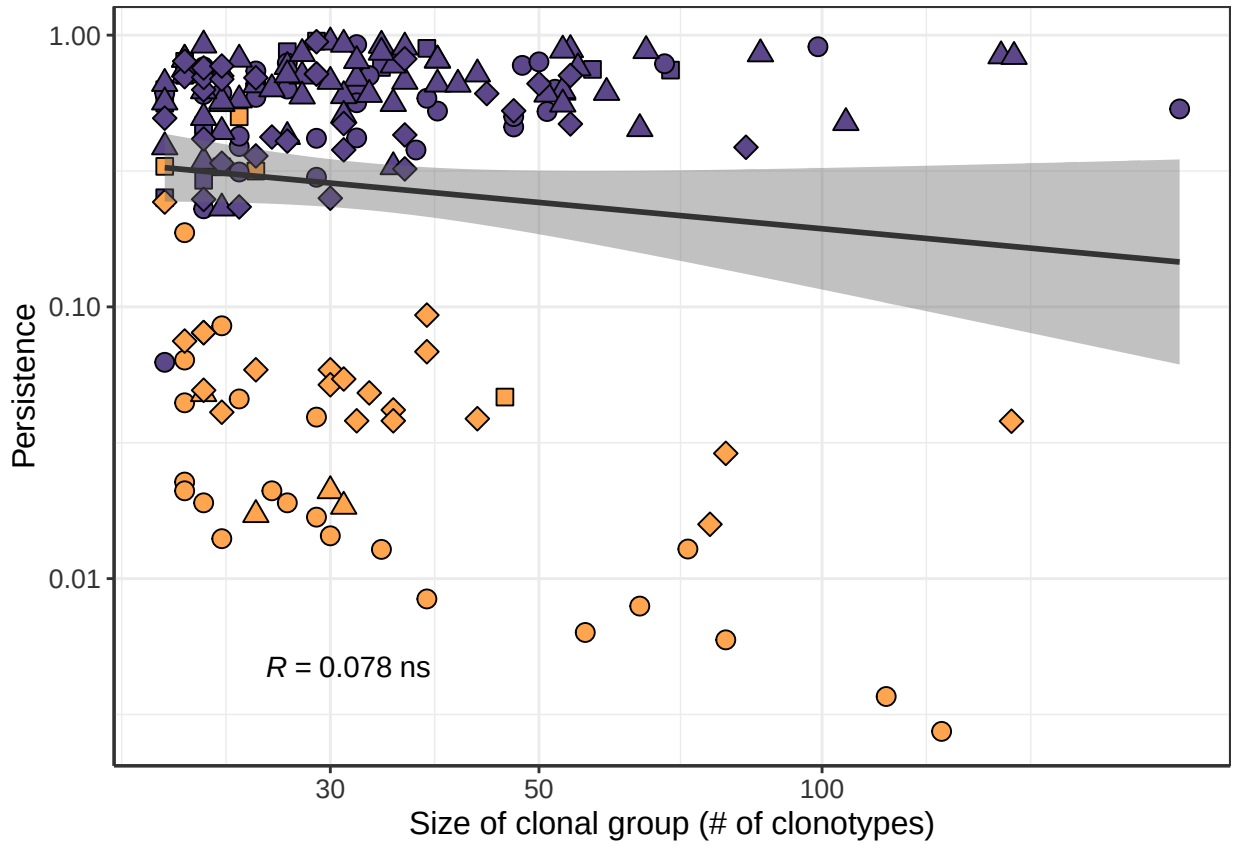


Fig. S4D: Spearman's correlation between the size of the clonal group and its persistence. Statistical significance is the following: * - $p \leq 0.05$, ** - $p \leq 0.01$, * - $p \leq 10^{-3}$, **** - $p \leq 10^{-4}$.**

```
size_pers <- ggplot(lineage_composition, aes(x = size, y = pers)) +
  geom_point(aes(fill = cluster, shape = patient, alpha = cluster), size = 3,  color = "black") +
  geom_smooth(method = 'lm', color = "grey20", fill = "grey40") +
  my_shapes_patients +
  stat_cor(method="spearman", size = 4, label.x.npc = 0.1, label.y.npc = 0.1,
          aes(label = paste(..r.label.., cut(..p.., breaks = c(-Inf, 0.0001, 0.001, 0.01, 0.05, Inf),
                                             labels = c("'****'", "'***'", "'**'", "'*'", "'ns'")),
                            sep = "~"))) +
  my_colors +
  scale_alpha_manual(values = c("2" = 0.7, "1" = 0.5)) +
  scale_y_continuous(trans = "log10", name=("Persistence")) +
  scale_x_continuous(trans = "log10", name = ("Size of clonal group (# of clonotypes)")) +
  my_theme
size_pers
```



Part 2. LBmem clonal groups could arise from HBmem clonal groups.

(Results from Figure 4)

Fig. 4B: Comparison of distances between the germline sequence and the MRCA of a clonal group (G-MRCA distance) for groups of HBmem and LBmem clusters; Level of significance is obtained by the Mann-Whitney test: * - $p \leq 0.05$, ** - $p \leq 0.01$, * - $p \leq 10^{-3}$, **** - $p \leq 10^{-4}$.**

```
for (i in lineage_composition$file){

    ## g_mrca p-distance
    g_mrca_alig <- read.FASTA(paste0(path, "G-MRCA/", i), type = "DNA")
    g_mrca_alig <- as.character(as.matrix(g_mrca_alig))
    seq_length <- length(g_mrca_alig[1, ])
    g_mrca_alig <- g_mrca_alig[, g_mrca_alig[1,] != "-" & g_mrca_alig[2,] != "-"]
    lineage_composition$g_mrca_dist[lineage_composition$file == i] <-
      sum(g_mrca_alig[1,] != g_mrca_alig[2,]) / seq_length

    ## mean phylogenetic distance
    tr <- read.tree(paste0(path, "trees/RAxML_bestTree.", i, ".out"))
    pairs <- cophenetic.phylo(tr)
    pairs <- pairs[,names(pairs[1,]) != "germline"]
```

```
    pairs <- pairs[names(pairs[,1]) != "germline",]
    group_size <- lineage_composition$size[lineage_composition$file == i]
    lineage_composition$phylo_mean_p_dist[lineage_composition$file == i] <-
      sum(pairs)/(group_size*(group_size-1))
}

g_mrca_dist <- ggplot(lineage_composition, aes(x = cluster, y = g_mrca_dist, fill = cluster)) +
  geom_quasirandom(width=0.3, size = 0.4, color = "grey30", alpha = 0.8) +
  geom_boxplot(alpha = 0.8, outlier.colour = NA) +
  stat_compare_means(comparisons = cluster_comparisons, method = "wilcox.test", size = 4, label = "p.sig
  scale_y_continuous(name=("G-MRCA P-distance"), limits = c(0, 0.11)) +
  scale_x_discrete(name = "Cluster") +
  my_colors +
  my_theme
g_mrca_dist
```

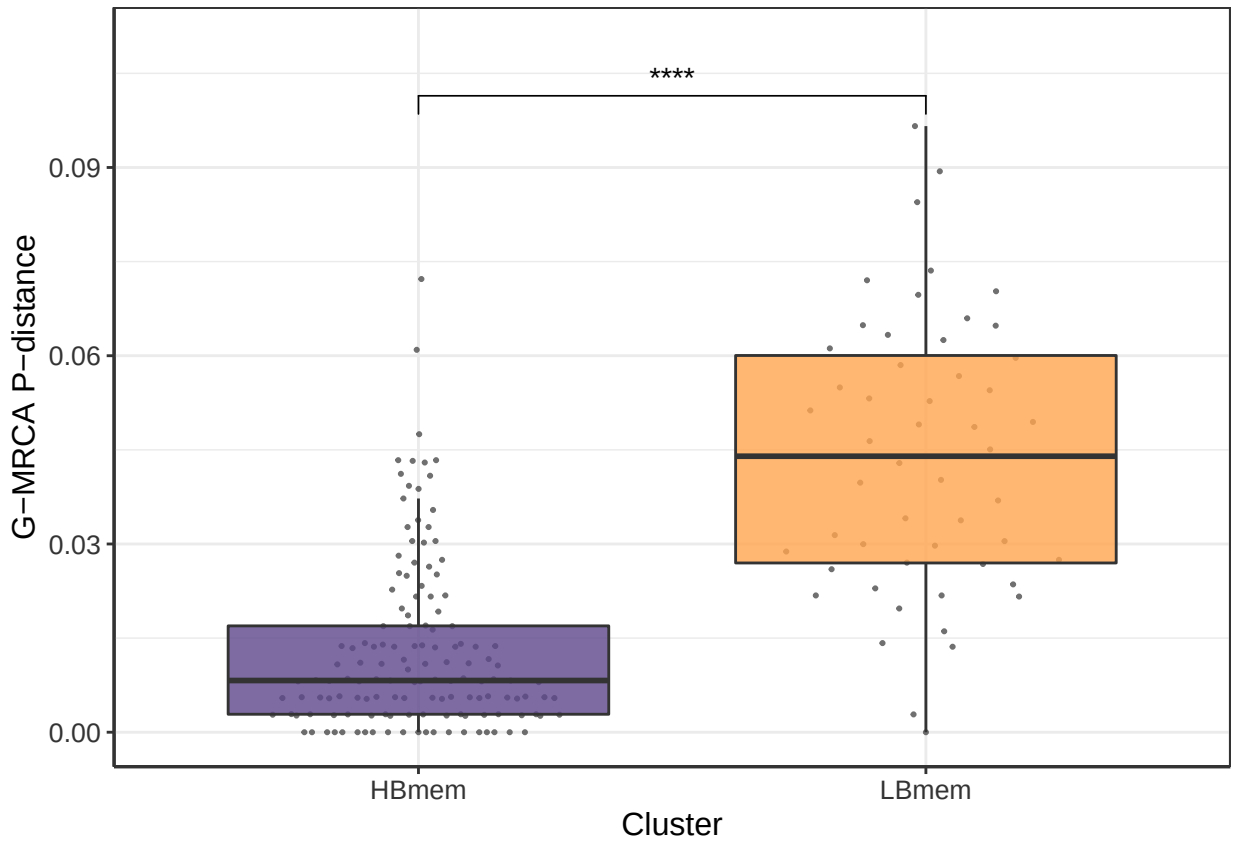


Fig. 4C: Mean pairwise phylogenetic distance, showing genetic divergence of clonotypes of the groups in two clusters; Level of significance is obtained by the Mann-Whitney test: * - $p \leq 0.05$, ** - $p \leq 0.01$, * - $p \leq 10^{-3}$, **** - $p \leq 10^{-4}$.**

```
p_mean_dist <- ggplot(lineage_composition, aes(x = cluster, y = phylo_mean_p_dist, fill = cluster)) +
  geom_quasirandom(width=0.3, size = 0.4, color = "grey30", alpha = 0.8) +
  geom_boxplot(alpha = 0.8, outlier.colour = NA) +
  stat_compare_means(comparisons = cluster_comparisons, method = "wilcox.test", size = 4, label = "p.sig
  scale_y_log10(name=("Mean pairwise distance"), limits = c(0.02, 0.5)) +
```

```
  scale_x_discrete(name = "Cluster") +
  my_colors +
  my_theme
p_mean_dist
```

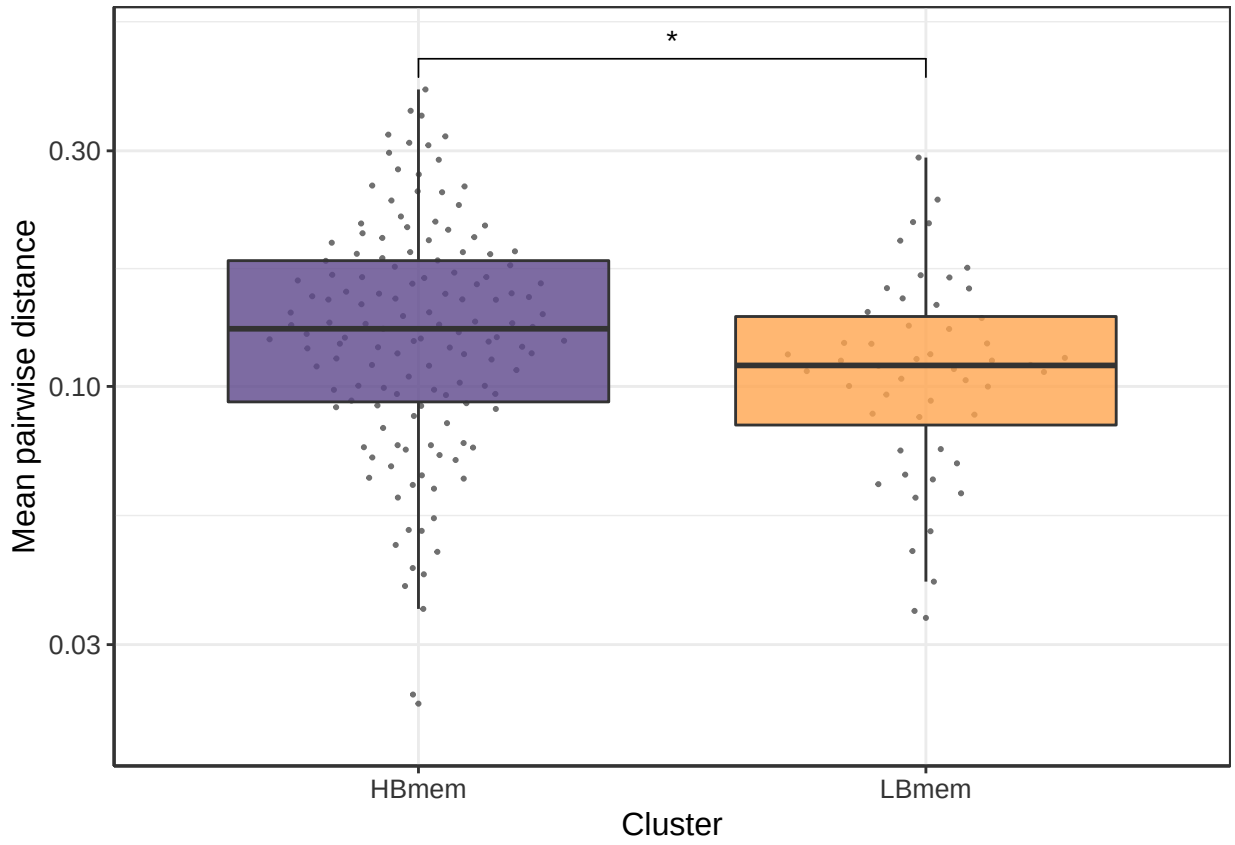


Fig. 4D: Typical representative of phylogenetic trees for clonal groups belonging to HBmem. Circles correspond to individual clonotypes, with the cellular subset indicated by color, and the isotype, by label. The table at the right of each tree indicates the presence or absence of the corresponding clonotype at this time point. The G-MRCA distance is indicated with a thick line.

```
tr_hb <- "IM_clone_39812_size_74_vj.fas"
patient <- "IM"

tree <- unname(unlist(read.table(paste0(path, "trees/RAxML_bestTree.", tr_hb, ".out"))))
tr_repeats <- repeats[repeats$clone == str_remove(tr_hb, "_vj.fas"),]

for (j in 1:nrow(tr_repeats)){
  x <- str_replace_all(tr_repeats$un_names[j], "R", ":0,")
  x <- paste0("(", x, ":0)", sep = "")
  sum_name_j <- str_extract(tr_repeats$sum_name[j], "[0-9]*_5_isot_")
  sum_name_j <- paste0(sum_name_j, "[0-9]*_[A-Z]*_[0-9]*-[0-9]*-[0-9*]")
  tree <- str_replace(tree, sum_name_j, x)
}
```

```
germ_branch <- str_extract(tree, ":[0-9].[0-9]*,germline:[0-9].[0-9]*\\);")
tree <- str_replace(tree, ":[0-9].[0-9]*,germline:[0-9].[0-9]*\\);", paste0("G-MRCA", germ_branch))
tree <- str_replace(tree, ",germline", "\\)")
tree <- str_replace(tree, "\\);", ";")

file_tree <- file(paste0(path, "tr_hb_with_repeats"))
writeLines(tree, file_tree)
close(file_tree)

tree <- read.tree(paste0(path, "tr_hb_with_repeats"))
meta <- cbind(tree$tip.label, str_extract(tree$tip.label, "_[A-Z]*_"),
              str_extract(tree$tip.label, "_isot_[0-9]*_"),
              str_extract(tree$tip.label, "201[0-9]*-[0-9]*-[0-9]*"))
meta <- data.frame(meta)
colnames(meta) <- c("tip", "cell", "isot", "date")
rownames(meta) <- tree$tip.label
meta$cell[meta$cell == "__"] <- "_B_"
meta$isot[meta$isot == "_isot_1_"] <- "M"
meta$isot[meta$isot == "_isot_2_"] <- "G"
meta$isot[meta$isot == "_isot_3_"] <- "A"
meta$cell[meta$cell == "NA"] <- "anc. seq."
meta$T1 <- as.numeric(meta$date == timepoints$Date[timepoints$Patient == patient & timepoints$Timepoint
meta$T2 <- as.numeric(meta$date == timepoints$Date[timepoints$Patient == patient & timepoints$Timepoint
meta$T3 <- as.numeric(meta$date == timepoints$Date[timepoints$Patient == patient & timepoints$Timepoint

meta[nrow(meta), c(5,6,7)] <- 0
tree$node.label[which(tree$node.label != "G-MRCA")] <- ""

p_hb <- ggtree(tree, layout = "rectangular") + geom_rootedge(size=1.5) + geom_rootpoint(color = "springg

p_hb <- p_hb %<+% meta +
     geom_tippoint(aes(color=cell),  shape = 16, size=5, alpha=0.8) +
     geom_label(aes(label=isot, size = isot, alpha = isot), position = position_nudge(x = 0.05), color =
                fill = "#E58601") +
     #geom_label_repel(aes(label=isot, size = isot, alpha = isot), color = "black", fill = "seagreen")
     ggtitle("     HBmem clonal group") +
     scale_size_manual("Cell Type",
                       values = c("M" = 2.5, "MRCA" = 2, "Germline" = 2, "G" = 2.5, "A" = 2.5)) +
     scale_color_manual("Cell Type",
                        values = c("_B_" = "#273046", "_P_" = "#F8AFA8", "_L_" = "#CB2314",
                                   "_Z_" = "slategray3", "_Y_" = "slategray3"),
                        labels = c("_B_" = "Bmem", "_P_" = "PBL", "_L_" = "PL", "_Z_" = "MRCA", "_Y_" =
   scale_alpha_manual("Isotype", values = c("M" = 0, "G" = 0.4, "A" = 1, "MRCA" = 0, "Germline" = 0)) +
   geom_treescale(x = 0.3, fontsize=3,  width = 0.02, linesize=0.3)

p_hb <- gheatmap(p_hb, meta[,c(5,6,7)], offset=0.07, width=0.17, low="white", high="grey40", colnames_p
                                    plot.title = element_text(colour="black", size = 12, face = "bold"))
p_hb
```

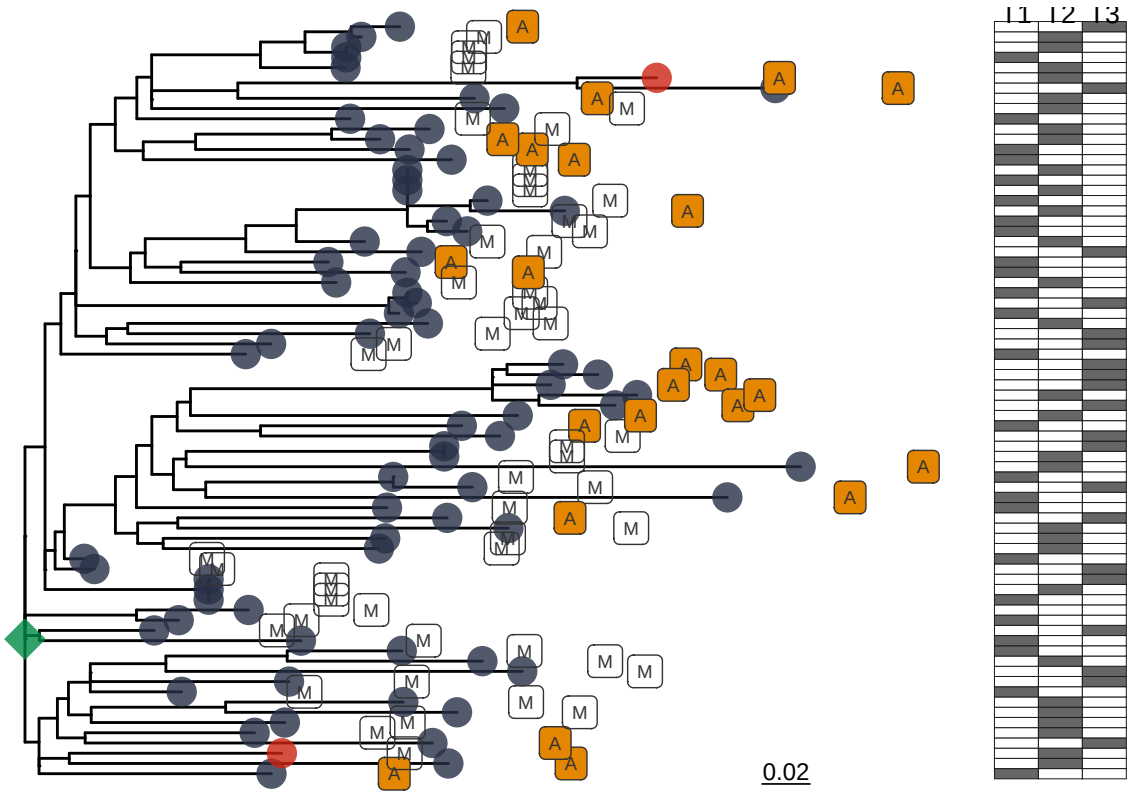


Fig. 4E: Typical representative of phylogenetic trees for clonal groups belonging to LBmem. Circles correspond to individual clonotypes, with the cellular subset indicated by color, and the isotype, by label. The table at the right of each tree indicates the presence or absence of the corresponding clonotype at this time point. The G-MRCA distance is indicated with a thick line.

```
tr_lb <- "MRK_clone_13788_size_55_vj.fas"
patient <- "MRK"

tree <- unname(unlist(read.table(paste0(path, "trees/RAxML_bestTree.", tr_lb, ".out"))))
tr_repeats <- repeats[repeats$clone == str_remove(tr_lb, "_vj.fas"),]

for (j in 1:nrow(tr_repeats)){
  x <- str_replace_all(tr_repeats$un_names[j], "R", ":0,")
  x <- paste0("(", x, ":0)", sep = "")
  sum_name_j <- str_extract(tr_repeats$sum_name[j], "[0-9]*_5_isot_")
  sum_name_j <- paste0(sum_name_j, "[0-9]*_[A-Z]*_[0-9]*-[0-9]*-[0-9*]")
  tree <- str_replace(tree, sum_name_j, x)
}

germ_branch <- str_extract(tree, ":[0-9].[0-9]*,germline:[0-9].[0-9]*\\);")
tree <- str_replace(tree, ":[0-9].[0-9]*,germline:[0-9].[0-9]*\\);", paste0("G-MRCA", germ_branch))
tree <- str_replace(tree, "G-MRCA:[0-9].[0-9]*", "G-MRCA:0")
tree <- str_replace(tree, ",germline", "\\)")
tree <- str_replace(tree, "\\);", ";")
```

```
file_tree <- file(paste0(path, "tr_lb_with_repeats"))
writeLines(tree, file_tree)
close(file_tree)

tree <- read.tree(paste0(path, "tr_lb_with_repeats"))
meta <- cbind(tree$tip.label, str_extract(tree$tip.label, "_[A-Z]*_"),
              str_extract(tree$tip.label, "_isot_[0-9]*_"),
              str_extract(tree$tip.label, "201[0-9]*-[0-9]*-[0-9]*"))
meta <- data.frame(meta)
colnames(meta) <- c("tip", "cell", "isot", "date")
rownames(meta) <- tree$tip.label
meta$cell[meta$cell == "__"] <- "_B_"
meta$isot[meta$isot == "_isot_1_"] <- "M"
meta$isot[meta$isot == "_isot_2_"] <- "G"
meta$isot[meta$isot == "_isot_3_"] <- "A"
meta$cell[meta$cell == "NA"] <- "anc. seq."
meta$T1 <- as.numeric(meta$date == timepoints$Date[timepoints$Patient == patient & timepoints$Timepoint
meta$T2 <- as.numeric(meta$date == timepoints$Date[timepoints$Patient == patient & timepoints$Timepoint
#meta$T2 <- rep(1, nrow(meta))
meta$T3 <- as.numeric(meta$date == timepoints$Date[timepoints$Patient == patient & timepoints$Timepoint
meta$T2[nrow(meta)] <- 1
tree$node.label[which(tree$node.label != "G-MRCA")] <- ""

p_lb <- ggtree(tree, layout = "rectangular") + geom_rootedge(size = 1.5) + geom_rootpoint(color = "spri
   geom_nodelab(nudge_x = -0.03, nudge_y = 1.2, size = 3)

p_lb <- p_lb %<+% meta +
     geom_tippoint(aes(color=cell),  shape = 16, size=5, alpha=0.9) +
     geom_label(aes(label=isot, size = isot, alpha = isot), position = position_nudge(x = 0.05), color =
                fill = "#E58601") +
     #geom_label_repel(aes(label=isot, size = isot, alpha = isot), color = "black", fill = "seagreen")
     ggtitle("     LBmem clonal group") +
     scale_size_manual("Cell Type",
                       values = c("M" = 2.5, "MRCA" = 2, "Germline" = 2, "G" = 2.5, "A" = 2.5)) +
     scale_color_manual("Cell Type",
                       values = c("_B_" = "#273046", "_P_" = "#F8AFA8", "_L_" = "#CB2314",
                                  "_Z_" = "grey40", "_Y_" = "black"),
                       labels = c("_B_" = "Bmem", "_P_" = "PBL", "_L_" = "PL", "_Z_" = "MRCA", "_Y_" =
   scale_alpha_manual("Isotype", values = c("M" = 0, "G" = 0.4, "A" = 1, "MRCA" = 0, "Germline" = 0)) +
   geom_treescale(x = 0.1, width = 0.02, fontsize=3, linesize=0.3)

p_lb1 <- p_lb
p_lb <- gheatmap(p_lb, meta[,c(5,6,7)], offset=0.06, width=0.6, low="white", high="grey40", colnames_pos

p_lb
```

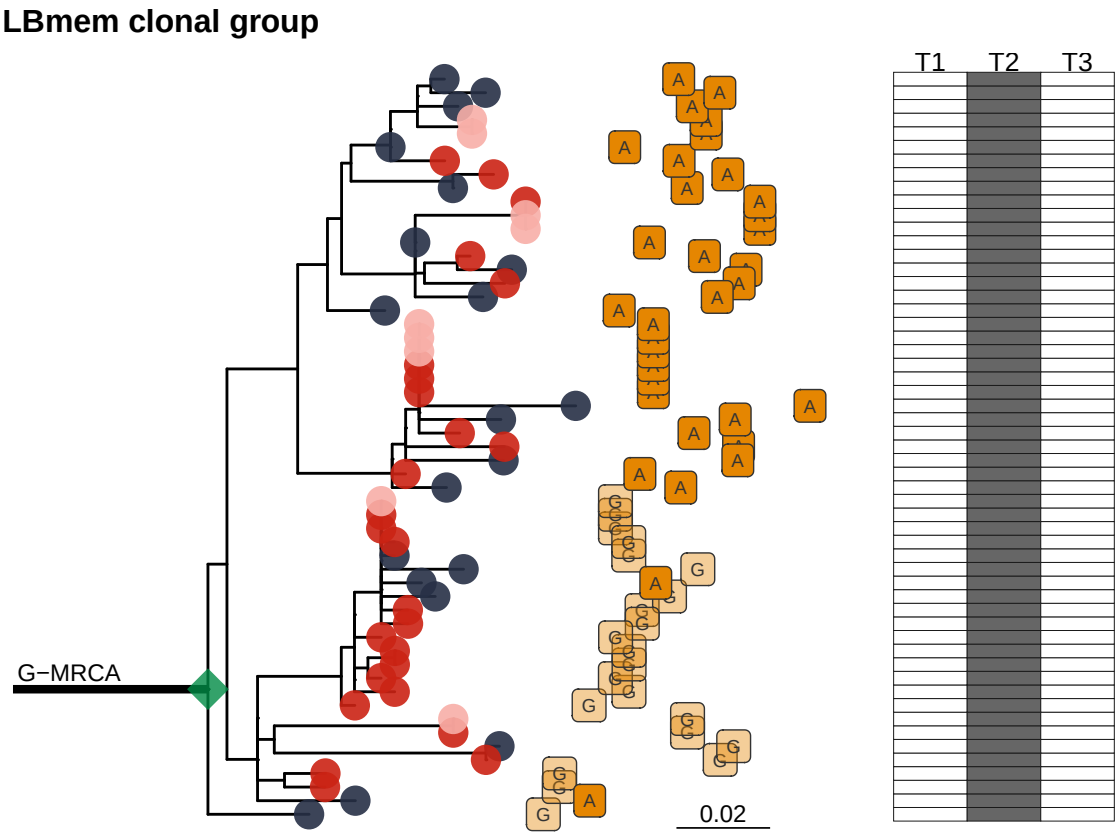


Fig. 4F: The example of the tree, showing HBmem - LBmem transition. Circles correspond to individual clonotypes, with the cellular subset indicated by color, and the isotype, by label. The table at the right of each tree indicates the presence or absence of the corresponding clonotype at this time point. The G-MRCA distance is indicated with a thick line.

```
tr_int <- "MRK_clone_15136_size_32_vj.fas"
patient <- "MRK"

tree <- unname(unlist(read.table(paste0(path, "trees/RAxML_bestTree.", tr_int, ".out"))))
tr_repeats <- repeats[repeats$clone == str_remove(tr_int, "_vj.fas"),]

for (j in 1:nrow(tr_repeats)){
  x <- str_replace_all(tr_repeats$un_names[j], "R", ":0,")
  x <- paste0("(", x, ":0)", sep = "")
  sum_name_j <- str_extract(tr_repeats$sum_name[j], "[0-9]*_5_isot_")
  sum_name_j <- paste0(sum_name_j, "[0-9]*_[A-Z]*_[0-9]*-[0-9]*-[0-9*]")
  tree <- str_replace(tree, sum_name_j, x)
}

germ_branch <- str_extract(tree, ":[0-9].[0-9]*,germline:[0-9].[0-9]*\\);")
tree <- str_replace(tree, ":[0-9].[0-9]*,germline:[0-9].[0-9]*\\);", paste0("G-MRCA", germ_branch))
tree <- str_replace(tree, "G-MRCA:[0-9].[0-9]*", "G-MRCA:0")
tree <- str_replace(tree, ",germline", "\\)")
tree <- str_replace(tree, "\\);", ";")
```

```
file_tree <- file(paste0(path, "tr_int_with_repeats"))
writeLines(tree, file_tree)
close(file_tree)

tree <- read.tree(paste0(path, "tr_int_with_repeats"))
meta <- cbind(tree$tip.label, str_extract(tree$tip.label, "_[A-Z]*_"),
              str_extract(tree$tip.label, "_isot_[0-9]*_"),
              str_extract(tree$tip.label, "201[0-9]*-[0-9]*-[0-9]*"))
meta <- data.frame(meta)
colnames(meta) <- c("tip", "cell", "isot", "date")
rownames(meta) <- tree$tip.label
meta$cell[meta$cell == "__"] <- "_B_"
meta$isot[meta$isot == "_isot_1_"] <- "M"
meta$isot[meta$isot == "_isot_2_"] <- "G"
meta$isot[meta$isot == "_isot_3_"] <- "A"
meta$cell[meta$cell == "NA"] <- "anc. seq."
meta$T1 <- as.numeric(meta$date == timepoints$Date[timepoints$Patient == patient & timepoints$Timepoint
meta$T2 <- as.numeric(meta$date == timepoints$Date[timepoints$Patient == patient & timepoints$Timepoint
meta$T3 <- as.numeric(meta$date == timepoints$Date[timepoints$Patient == patient & timepoints$Timepoint
meta$T3[which(meta$tip == "1878_2_isot_1_B_2018-05-07")] <- 1
meta$T3[which(meta$tip == "790_2_isot_3_B_2018-05-07")] <- 1
meta$T2[which(meta$tip == "5018_1_isot_1_B_2017-05-15")] <- 1

meta[nrow(meta), c(5,6,7)] <- 0
tree$node.label[which(tree$node.label != "G-MRCA")] <- ""

p_react1 <- ggtree(tree, layout = "rectangular")  + geom_rootedge(size = 1.5) + geom_rootpoint(color =

p_react1 <- p_react1 %<+% meta +
     geom_tippoint(aes(color=cell),  shape = 16, size=5, alpha=0.8) +
     geom_label(aes(label=isot, size = isot, alpha = isot), position = position_nudge(x = 0.05), color =
                fill = "#E58601") +
     ggtitle("     HBmem - LBmem transition") +
     scale_size_manual("Cell Type",
                       values = c("M" = 2.5, "MRCA" = 2, "Germline" = 2, "G" = 2.5, "A" = 2.5)) +
     scale_color_manual("Cell Type",
                       values = c("_B_" = "#273046", "_P_" = "#F8AFA8", "_L_" = "#CB2314",
                                  "_Z_" = "grey40", "_Y_" = "black"),
                       labels = c("_B_" = "Bmem", "_P_" = "PBL", "_L_" = "PL", "_Z_" = "MRCA", "_Y_" =
   scale_alpha_manual("Isotype", values = c("M" = 0, "G" = 0.4, "A" = 1, "MRCA" = 0, "Germline" = 0)) +
   geom_treescale(fontsize=3, linesize=0.3, width = 0.02, x = 0.3)

p_react1 <- p_react1
p_react1 <- gheatmap(p_react1, meta[,c(5,6,7)], offset=0.07, width=0.23, low="white", high="grey40", col

p_react1
```

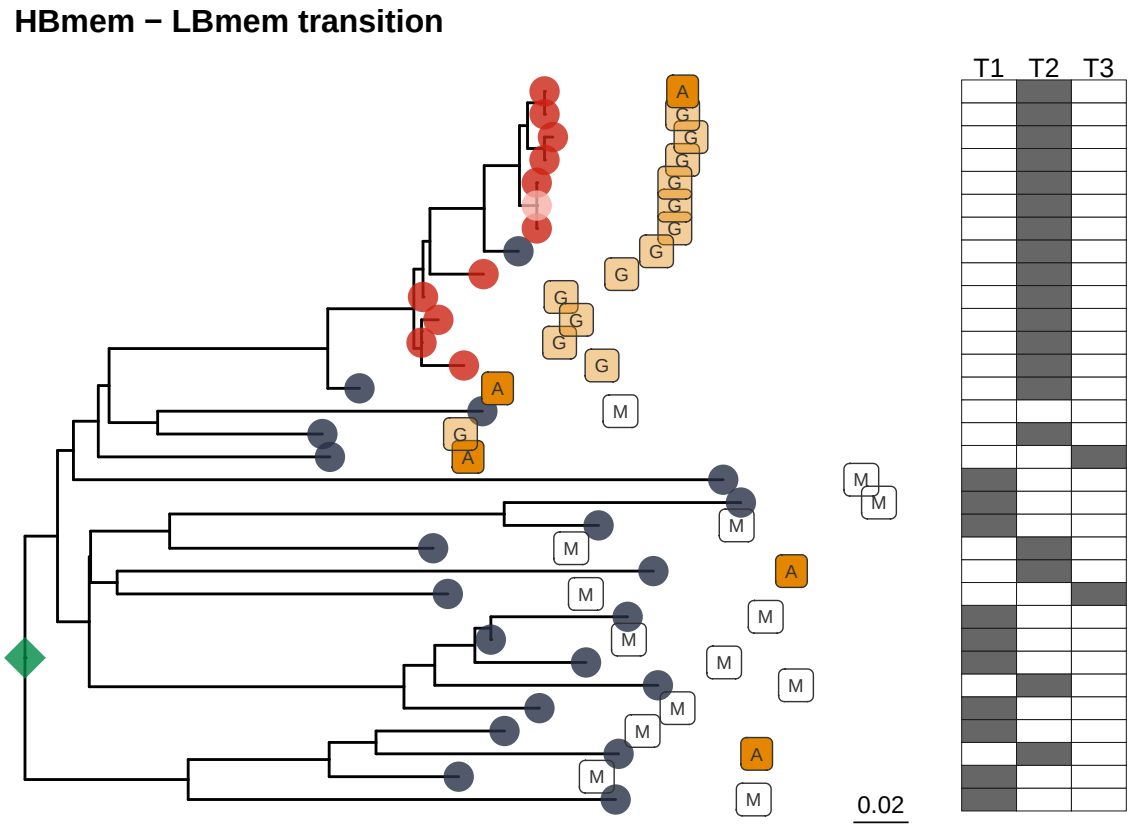


Fig. 4G: Schematic representation of the hypothetical frequency dynamics of HBmem clonal lineage.

```
df <- cbind(1:10, c(0.007, 0.0068, 0.0067, 0.0071, 0.007, 0.0073, 0.007, 0.0067, 0.0068, 0.007),
            c(0.002, 0.0019, 0.002, 0.002, 0.008, 0.005, 0.0025, 0.002, 0.002, 0.002),
            c(0.0058, 0.0061, 0.0057, 0.0058, 0.0076, 0.006, 0.0058, 0.006, 0.0058, 0.0056))
df <- data.frame(df)
colnames(df) <- c("x1", "y1", "y2", "y3")

p_scheme1 <- ggplot(df) +
  annotate("rect", ymin=0, ymax=0.01, xmin = 1, xmax = 10,
           alpha=0.3, fill = "mediumpurple4", color="white") +
  geom_line(aes(x = x1, y = y1), color = "mediumpurple4", size = 2) +
  geom_hline(yintercept = 0.0055, linetype="dashed", color = "grey30", size = 1.5) +
  annotate("text", x = 9.2, y=0.004, label = 'plain("detection \n    limit")', parse = TRUE, size = 4) +
  annotate("point", x = df$x1[2], y = df$y1[2], colour = "coral3", size = 4) +
  annotate("point", x = df$x1[5], y = df$y1[5], colour = "coral3", size = 4) +
  annotate("point", x = df$x1[8], y = df$y1[8], colour = "coral3", size = 4) +
  annotate("text", x = df$x1[2], y = df$y1[2] + 0.001, label = 'bold("sampled \n   at T1")', parse = TRU
  annotate("text", x = df$x1[5], y = df$y1[5] + 0.001, label = 'bold("sampled \n   at T2")', parse = TRU
  annotate("text", x = df$x1[8], y = df$y1[8] + 0.001, label = 'bold("sampled \n   at T3")', parse = TRU
  scale_y_continuous("Repertoire frequency", limits = c(0, 0.01)) +
  scale_x_continuous("Time") +
  #ggtitle("    Sampling of HBmem group") +
  my_theme +
```

```
  theme(axis.ticks.x=element_blank(),
        axis.ticks.y=element_blank(),
        axis.text.x=element_blank(),
        axis.text.y=element_blank())

p_scheme1
```

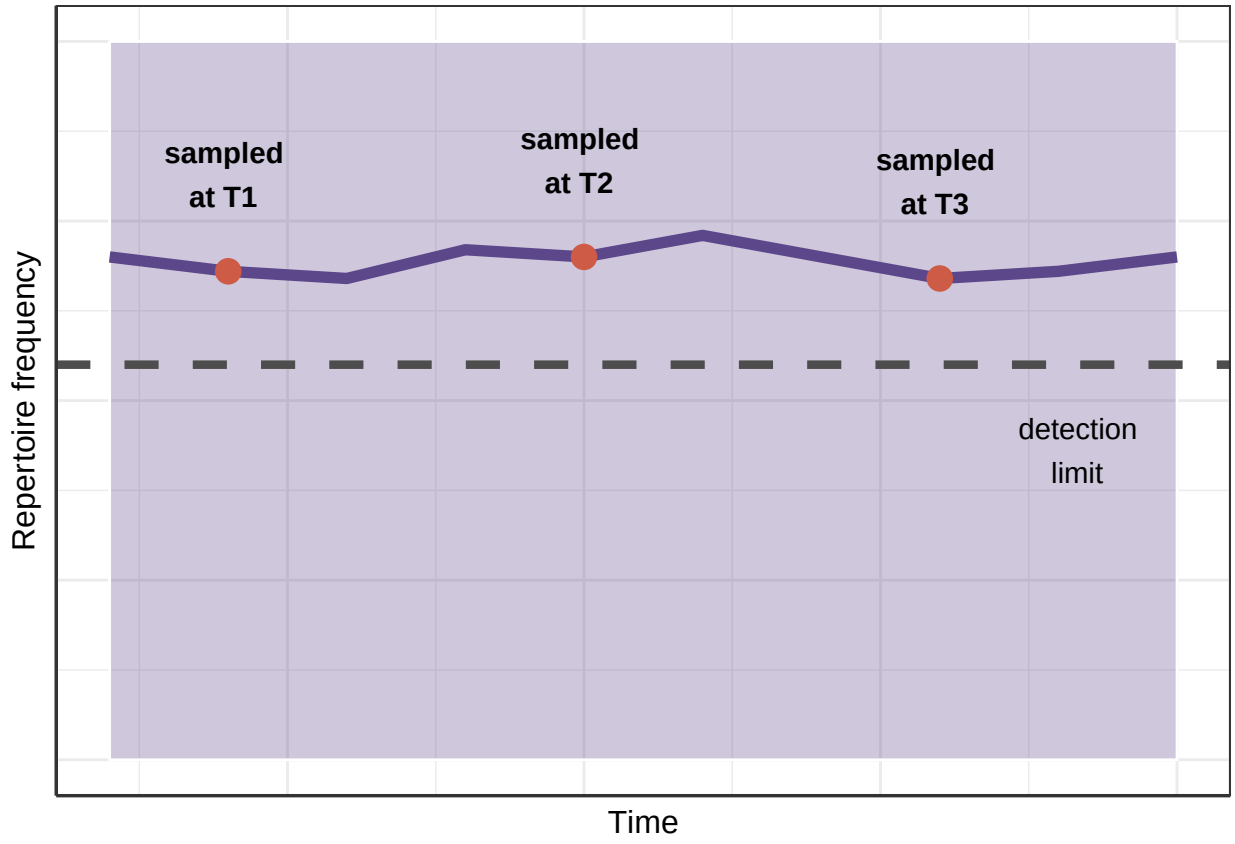


Fig. 4H: Schematic representation of the hypothetical frequency dynamics of LBmem clonal lineage.

```
p_scheme2 <- ggplot(df) +
  annotate("rect", ymin=0, ymax=0.0055, xmin = 1, xmax = df$x1[4],
           alpha=0.3, fill = "mediumpurple4", color="white") +
  annotate("rect", ymin=0, ymax=0.0055, xmin = df$x1[7], xmax = 10,
           alpha=0.3, fill = "mediumpurple4", color="white") +
  annotate("rect", ymin=0, ymax=0.01, xmin = df$x1[4], xmax = df$x1[7],
           alpha=0.3, fill = "tan1", color="white") +
  geom_line(aes(x = x1, y = y2, colour=(x1 > 3 & x1 < 7), group = 1),  size = 2) +
  geom_hline(yintercept = 0.0055, linetype="dashed", color = "grey30", size = 1.5) +
  geom_vline(xintercept = df$x1[4], linetype="dashed", color = "grey30") +
  geom_vline(xintercept = df$x1[7], linetype="dashed", color = "grey30") +
  annotate("text", x = 9.2, y=0.006, label = 'plain("detection \n    limit")', parse = TRUE, size = 4) +
  annotate("point", x = df$x1[2], y = df$y2[2], colour = "coral3", size = 4) +
  annotate("point", x = df$x1[5], y = df$y2[5], colour = "coral3", size = 4) +
  annotate("point", x = df$x1[8], y = df$y2[8], colour = "coral3", size = 4) +
```

```
  annotate("text", x = df$x1[2], y = df$y2[2] + 0.001, label = 'plain("not sampled \n      at T1")', pa
  annotate("text", x = df$x1[5], y = df$y2[5] + 0.001, label = 'bold("sampled \n   at T2")', parse = TR
  annotate("text", x = df$x1[8], y = df$y2[8] + 0.001, label = 'plain("not sampled \n      at T3")', pa
  annotate("text", x = 2.5, y = 0.0004, label = 'plain("persistent \n memory")', parse = TRUE, size = 4
  annotate("text", x = 5.5, y = 0.0004, label = 'bold("   memory \n reactivation")', parse = TRUE, size
  annotate("text", x = 8.5, y = 0.0004, label = 'plain("persistent \n memory")', parse = TRUE, size = 4
  annotate("text", x = 2.7, y = 0.009, label = 'plain("    clonal \n expansion")', parse = TRUE, size =
  annotate("text", x = 8.3, y = 0.009, label = 'plain("    clonal \n contraction")', parse = TRUE, size
  geom_curve(x = 3, y = 0.0087, xend = 4.4, yend = 0.0046, curvature = 0.4, color = "grey30",
             arrow = arrow(length = unit(0.03, "npc"))) +
  geom_curve(x = 8, y = 0.0087, xend = 6.2, yend = 0.0046, curvature = -0.4, color = "grey30",
             arrow = arrow(length = unit(0.03, "npc"))) +
  scale_colour_manual(values = c("mediumpurple4", "tan1")) +
  scale_y_continuous("Repertoire frequency", limits = c(0, 0.01)) +
  scale_x_continuous("Time") +
  #ggtitle("    Sampling of LBmem group") +
  my_theme +
  theme(axis.ticks.x=element_blank(),
        axis.ticks.y=element_blank(),
        axis.text.x=element_blank(),
        axis.text.y=element_blank())

p_scheme2
```

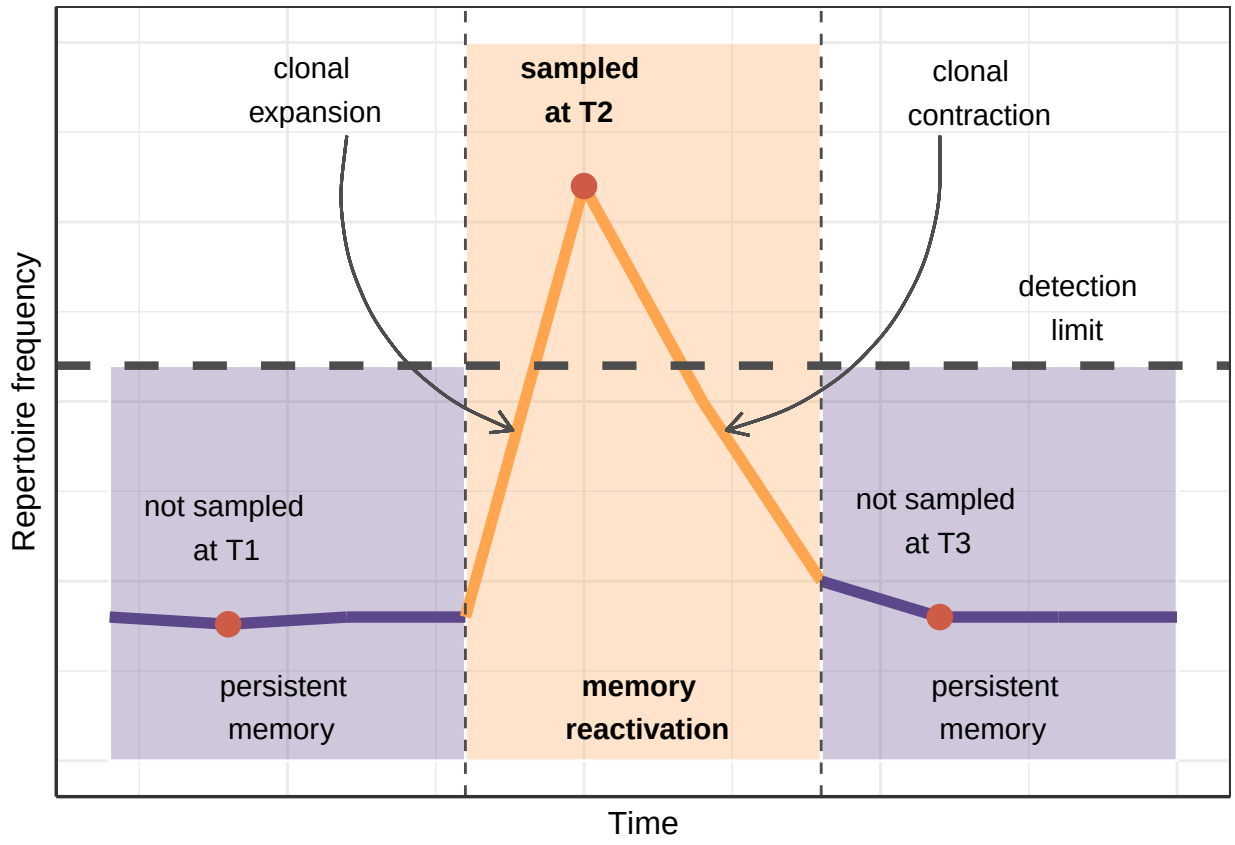
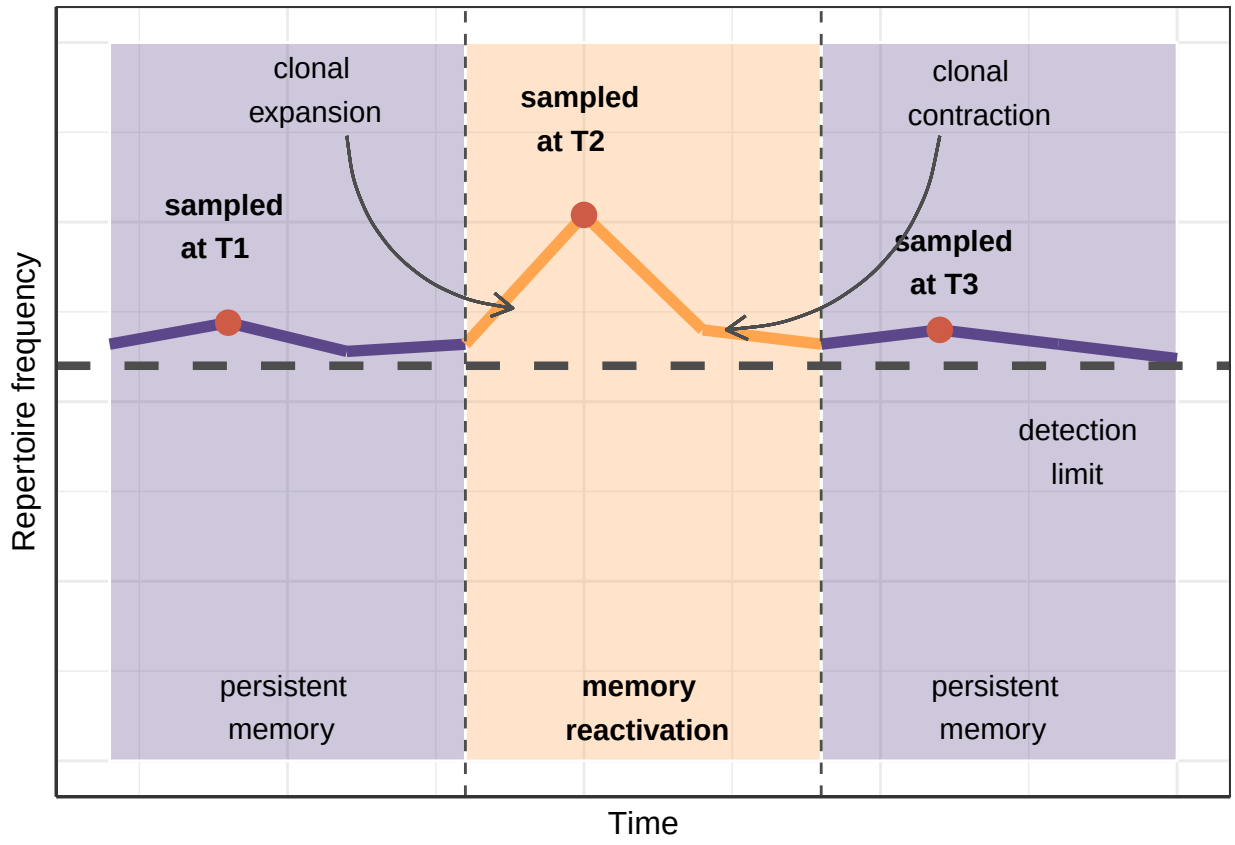


Fig. 4E: Schematic representation of the hypothetical frequency dynamics of clonal lineage with HBmem - LBmem transition, shown of Fig. 4F.

```
p_scheme3 <- ggplot(df) +
  annotate("rect", ymin=0, ymax=0.01, xmin = 1, xmax = df$x1[4],
           alpha=0.3, fill = "mediumpurple4", color="white") +
  annotate("rect", ymin=0, ymax=0.01, xmin = df$x1[7], xmax = 10,
           alpha=0.3, fill = "mediumpurple4", color="white") +
  annotate("rect", ymin=0, ymax=0.01, xmin = df$x1[4], xmax = df$x1[7],
           alpha=0.3, fill = "tan1", color="white") +
  geom_line(aes(x = x1, y = y3, colour=(x1 > 3 & x1 < 7), group = 1),  size = 2) +
  geom_hline(yintercept = 0.0055, linetype="dashed", color = "grey30", size = 1.5) +
  geom_vline(xintercept = df$x1[4], linetype="dashed", color = "grey30") +
  geom_vline(xintercept = df$x1[7], linetype="dashed", color = "grey30") +
  annotate("text", x = 9.2, y=0.004, label = 'plain("detection \n    limit")', parse = TRUE, size = 4) +
  annotate("point", x = df$x1[2], y = df$y3[2], colour = "coral3", size = 4) +
  annotate("point", x = df$x1[5], y = df$y3[5], colour = "coral3", size = 4) +
  annotate("point", x = df$x1[8], y = df$y3[8], colour = "coral3", size = 4) +
  annotate("text", x = df$x1[2], y = df$y3[2] + 0.001, label = 'bold("sampled \n  at T1")', parse = TRU
  annotate("text", x = df$x1[5], y = df$y3[5] + 0.001, label = 'bold("sampled \n  at T2")', parse = TRU
  annotate("text", x = df$x1[8] + 0.15, y = df$y3[8] + 0.0006, label = 'bold("sampled \n  at T3")', pars
  annotate("text", x = 2.5, y = 0.0004, label = 'plain("persistent \n memory")', parse = TRUE, size = 4
  annotate("text", x = 5.5, y = 0.0004, label = 'bold("   memory \n reactivation")', parse = TRUE, size
  annotate("text", x = 8.5, y = 0.0004, label = 'plain("persistent \n memory")', parse = TRUE, size = 4
  annotate("text", x = 2.7, y = 0.009, label = 'plain("    clonal \n expansion")', parse = TRUE, size =
  annotate("text", x = 8.3, y = 0.009, label = 'plain("    clonal \n contraction")', parse = TRUE, size
  geom_curve(x = 3, y = 0.0087, xend = 4.4, yend = 0.0063, curvature = 0.4, color = "grey30",
             arrow = arrow(length = unit(0.03, "npc"))) +
  geom_curve(x = 8, y = 0.0087, xend = 6.2, yend = 0.006, curvature = -0.4, color = "grey30",
             arrow = arrow(length = unit(0.03, "npc"))) +
  scale_colour_manual(values = c("mediumpurple4", "tan1")) +
  scale_y_continuous("Repertoire frequency", limits = c(0, 0.01)) +
  scale_x_continuous("Time") +
  #ggtitle("    Sampling of LBmem group") +
  my_theme +
  theme(axis.ticks.x=element_blank(),
        axis.ticks.y=element_blank(),
        axis.text.x=element_blank(),
        axis.text.y=element_blank())

p_scheme3
```



Part 3. Reactivation of LBmem clonal groups is driven by positive selection.

(Results from Figure 5B)

Loading genetic code and calculating per codon number of nonsynonymous and synonymous sites.

```
gc <- geneticCodeTable(DNA = TRUE)
nucl <- c("A", "T", "G", "C")
Ss <- c()
Ns <- c()
for (i in 1:64) {
  codon <- unlist(str_split(gc$GeneticCode[i], ""))
  aa <- gc$AminoAcids[i]
  s = 0
  ### x - through 1, 2, 3 nucl of codon
  for (x in 1:3){
    subst <- nucl[nucl != codon[x]]
    ### for nucleotides, in which x can be substituted
    for (y in 1:3){
      subst_codon <- codon
      subst_codon[x] <- subst[y]
      subst_aa <- gc$AminoAcids[gc$GeneticCode == paste0(subst_codon, collapse = "")]
      if (aa == subst_aa) {s <- s + 1/3}
```

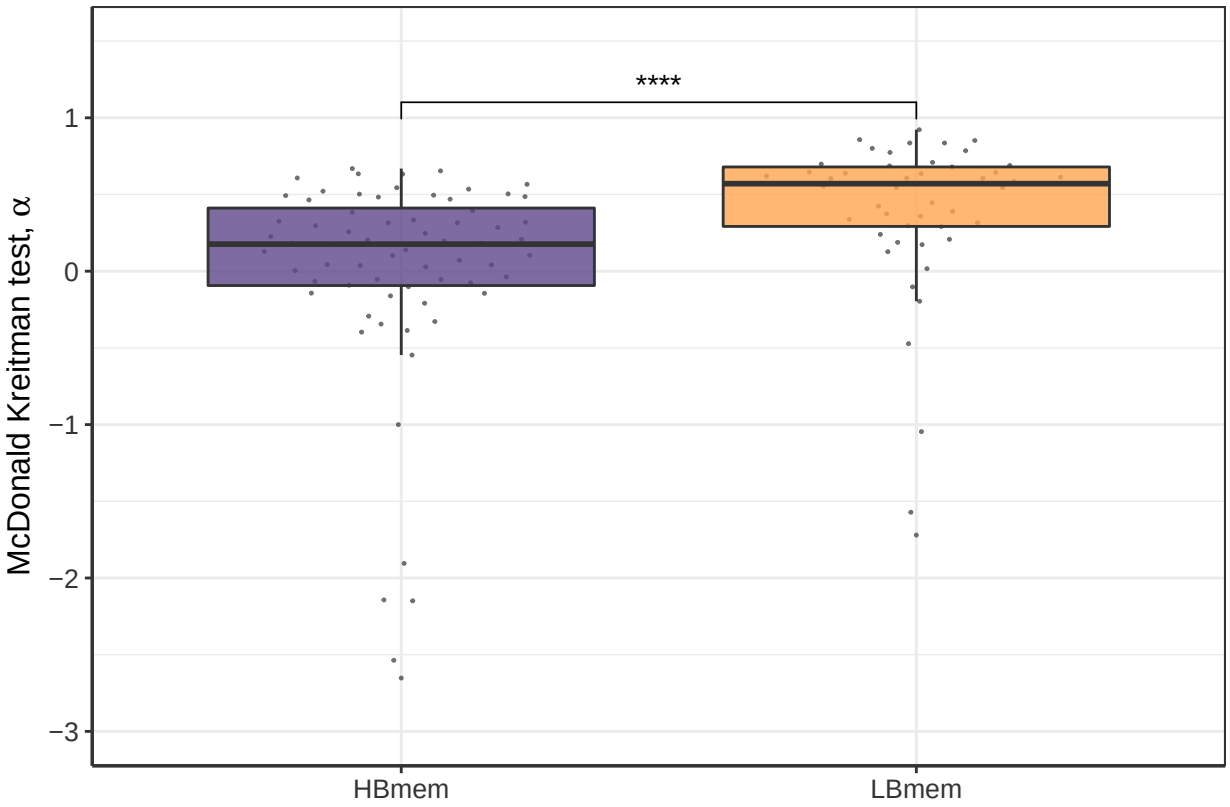
```
    }
  }
  Ss <- c(Ss, s)
  Ns <- c(Ns, 3-s)
}
gc$Ss <- Ss
gc$Ns <- Ns
```

Fig. 5B: McDonald-Kreitman estimate of the fraction of adaptive non-synonymous changes α between germline and MRCA in HBmem and LBmem clonal groups, wich have a nonzero G-MRCA distance. Comparisons were performed by Mann-Whitney test, notation of the level of significance is the following: * - $p \leq 0.05$, ** - $p \leq 0.01$, * - $p \leq 10^{-3}$, **** - $p \leq 10^{-4}$.**

```
thresholds_nice <- c("all", "no singletons", ">5%", ">10%", ">15%", ">20%", ">25%", ">30%")
mcd_kr <- c()

for (lin in lineage_composition$file){
  all <- read.FASTA(paste0(path, "alignments/", lin), type = "DNA")
  al_matrix <- as.character(as.matrix(all))
  al_matrix <- al_matrix[order(row.names(al_matrix)),]

  ## length of the sequence in the alignment
  l_seq <- length(all[[1]])
  l_full <- 3*(l_seq%/%3)
  ## number of conotypes
  n <- nrow(al_matrix)-2
  ## positions in codon
  index <- c(1,2,3)
  mutation_table <- c()

  for (pos in 1:(3*(l_seq%/%3))){

    n_codon <- (2 + pos)%/%3
    codon_pos <-  pos%%3
    if (codon_pos == 0){codon_pos = 3}
    ancestor_codon <- paste0(al_matrix['ancestor', (3*n_codon-2):(3*n_codon)], collapse = "")
    ancestor_nucl <- al_matrix['ancestor', pos]
    germline_codon <- paste0(al_matrix['germline', (3*n_codon-2):(3*n_codon)], collapse = "")
    germline_nucl <- al_matrix['germline', pos]

    variants_nucl <- unique(al_matrix[1:n, pos])
    ## remove - from variants here, so there is no such condition in if
    variants_nucl <- variants_nucl[variants_nucl != "-" & variants_nucl != al_matrix['ancestor', pos]]

    if (length(unique(c(ancestor_nucl, germline_nucl, variants_nucl))) > 1 &
      !str_detect(ancestor_codon, "-") & !str_detect(germline_codon, "-")) {
      # comparing germline with mrca
      if (germline_nucl == ancestor_nucl) {germ_mut = "0"
      } else if (is.element(germline_nucl, variants_nucl)) {
        germ_mut <- "poly"
      } else {
        ancestor_codon_change <- ancestor_codon
        substring(ancestor_codon_change, codon_pos, codon_pos) <- germline_nucl
```

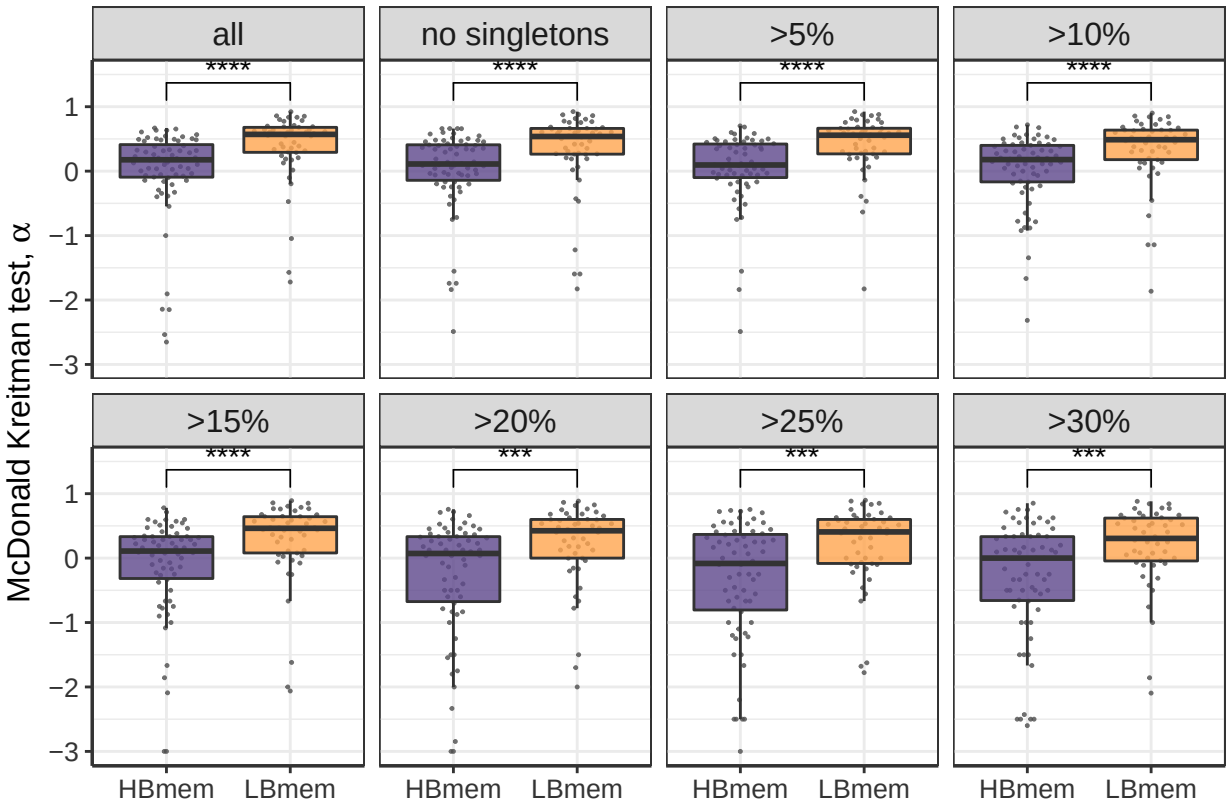
```
      if (gc$AA[gc$GeneticCode == toupper(ancestor_codon)] ==
          gc$AA[gc$GeneticCode == toupper(ancestor_codon_change)]) {germ_mut = "S"
      } else {germ_mut = "N"}
    }
    # comparing variants with mrca
    if (length(variants_nucl) == 0) {
      var_mut = "0"
      var_freqs = "-"
    } else {
      var_mut = ""
      var_freqs = ""
      for (var in variants_nucl) {
        var_codon <- ancestor_codon
        substring(var_codon, codon_pos, codon_pos) <- var
        if (gc$AA[gc$GeneticCode == toupper(ancestor_codon)] ==
            gc$AA[gc$GeneticCode == toupper(var_codon)]) {
          var_mut <- str_glue(var_mut, "S")
          var_freqs <- str_glue(var_freqs, sum(al_matrix[1:n, pos] == var), "_")
        } else {
          var_mut = str_glue(var_mut, "N")
          var_freqs <- str_glue(var_freqs, sum(al_matrix[1:n, pos] == var), "_")
        }
      }
    }
    mutation_table <- rbind(mutation_table,
                            c(pos, germ_mut, var_mut, var_freqs))
  }
}
mutation_table <- data.frame(mutation_table)
colnames(mutation_table) <- c("pos", "germ_mut", "var_mut", "var_freqs")
### counting fixed-polymorphic as polymorphic
Ds <- sum(mutation_table$germ_mut == "S")
Dn <- sum(mutation_table$germ_mut == "N")

mutation_table <-  mutation_table[mutation_table$var_mut != "0",]
zz <- nrow(mutation_table)
for (i in 1:zz){
  if (nchar(mutation_table$var_mut[i]) == 1){
    mutation_table$var_freqs[i] <- str_extract(mutation_table$var_freqs[i], "[0-9]*")
  } else {
    types <- unlist(str_split(mutation_table$var_mut[i], ""))
    freqs <- as.numeric(unlist(str_split(mutation_table$var_freqs[i], "_")))
    freqs <- freqs[!is.na(freqs)]
    ### counting each allele individually
    for (q in 1:length(types)){
      if (q == 1){
        mutation_table$var_mut[i] <- types[1]
        mutation_table$var_freqs[i] <- freqs[1]
      } else{
        mutation_table <- rbind(mutation_table,
                                c(mutation_table$pos[i], "-", types[q], freqs[q]))
      }
    }
```

```
    }
  }
  mutation_table$var_freqs <- round(as.numeric(mutation_table$var_freqs)/n, 5)
  thresholds <- c(0, round(1/n, 5), 0.05, 0.1, 0.15, 0.2, 0.25, 0.3)
  for (thr in thresholds) {
    Pn <- sum(mutation_table$var_mut[mutation_table$var_freqs > thr] == "N")
    Ps <- sum(mutation_table$var_mut[mutation_table$var_freqs > thr] == "S")

    alpha <- 1 - (((Pn+1)/(Ps+1)) / ((Dn+1)/(Ds+1)))
    cont_table <- matrix(c(Ds+1, Dn+1, Ps+1, Pn+1), nrow = 2,
    dimnames = list(c("Syn", "Nonsyn"), c("Fixed", "Poly")))
    f_test_gr <- fisher.test(cont_table, alternative = "greater")$p.value
    f_test_less <- fisher.test(cont_table, alternative = "less")$p.value
    f_test_two <- fisher.test(cont_table, alternative = "two.sided")$p.value

    mcd_kr <- rbind(mcd_kr,
                    c(lin, lineage_composition$cluster[lineage_composition$file == lin], alpha,
                      thresholds_nice[which(thresholds == thr)][1], f_test_gr, f_test_less,
                      f_test_two, Ds, Dn, Ps, Pn))
  }
}
mcd_kr <- data.frame(mcd_kr)
colnames(mcd_kr) <- c("clone", "cluster", "alpha", "thr", "p.val_gr", "p.val_less", "p.val_two", "Ds", "
mcd_kr[, c(3,5:11)] <- sapply(mcd_kr[, c(3,5:11)], as.numeric)
mcd_kr$thr <- factor(mcd_kr$thr, levels = thresholds_nice)

p_mcd <- ggplot(mcd_kr[mcd_kr$thr == "all" & (mcd_kr$Dn + mcd_kr$Ds) != 0,],
                aes(x = cluster, y = alpha, fill = cluster)) +
  geom_quasirandom(width=0.3, size = 0.2, color = "grey30", alpha = 0.8) +
  geom_boxplot(alpha = 0.8, outlier.colour = NA) +
  stat_compare_means(comparisons = cluster_comparisons, method = "wilcox.test", size = 4, paired = FALS
  scale_y_continuous(name=TeX("McDonald Kreitman test, $\\alpha$"), limits = c(-3, 1.5)) +
  scale_x_discrete(name = "") +
  scale_fill_manual(values = c("HBmem" = "mediumpurple4", "LBmem" = "tan1"), name = "") +
  my_theme
p_mcd
```



Robustness of the comparison to the exclusion of low frequent SHMs:

```
p_mcd_thr <- ggplot(mcd_kr[(mcd_kr$Dn + mcd_kr$Ds) != 0,],
                aes(x = cluster, y = alpha, fill = cluster)) +
  geom_quasirandom(width=0.3, size = 0.2, color = "grey30", alpha = 0.8) +
  geom_boxplot(alpha = 0.8, outlier.colour = NA) +
  stat_compare_means(comparisons = cluster_comparisons, method = "wilcox.test", size = 4, paired = FALSE
  scale_y_continuous(name=TeX("McDonald Kreitman test, $\\alpha$"), limits = c(-3, 1.5)) +
  scale_x_discrete(name = "") +
  scale_fill_manual(values = c("HBmem" = "mediumpurple4", "LBmem" = "tan1"), name = "") +
  my_theme +
  facet_wrap(~thr, ncol = 4)
p_mcd_thr
```



Exact Fisher Test on joined variotion in HBmem cluster:

```
H_cont_table <- matrix(apply(mcd_kr[mcd_kr$thr == "all" & mcd_kr$cluster == "HBmem", 8:11], 2, sum),
                       nrow = 2, dimnames = list(c("Syn", "Nonsyn"), c("Fixed", "Poly")))
fisher.test(H_cont_table, alternative = "less")
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  H_cont_table
## p-value = 3.415e-06
## alternative hypothesis: true odds ratio is less than 1
## 95 percent confidence interval:
##  0.0000000 0.6171894
## sample estimates:
## odds ratio
##  0.4485335
```

Exact Fisher Test on joined variotion in LBmem cluster:

```
L_cont_table <- matrix(apply(mcd_kr[mcd_kr$thr == "all" & mcd_kr$cluster == "LBmem", 8:11], 2, sum),
                       nrow = 2, dimnames = list(c("Syn", "Nonsyn"), c("Fixed", "Poly")))
fisher.test(L_cont_table, alternative = "less")
```

```
##
##  Fisher's Exact Test for Count Data
```

```
##
## data:  L_cont_table
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is less than 1
## 95 percent confidence interval:
##  0.0000000 0.4403563
## sample estimates:
## odds ratio
##  0.3513594
```

Part 4. Subsequent evolution of LBmem clonal groups is affected by negative and positive selection.

(Results from Figure 3 and Supplementary Figure 5C:E)

Fig. 5C: Comparison of mean pairwise $\pi N \pi S$ of HBmem and LBmem groups. Comparisons were performed by Mann-Whitney test, notation of the level of significance is the following: * - $p \leq 0.05$, ** - $p \leq 0.01$, * - $p \leq 10^{-3}$, **** - $p \leq 10^{-4}$.**

```
pnps_table <- c()
sfs_all <- c()

bins <- seq(0.05, 1, by=0.05)
bins_0 <- seq(0, 0.95, by=0.05)
bin_l <- 0.05
Bins <- seq(0.2, 1.2, by = 0.2)
Bins_0 <- c(seq(0, 0.8, by = 0.2), 0)

for (lin in lineage_composition$file){
  al <- read.FASTA(paste0(path, "alignments/", lin), type = "DNA")
  al_matrix <- as.character(as.matrix(al))
  al_matrix <- al_matrix[order(row.names(al_matrix)),]

  clone_name <- substring(lin, 1, nchar(lin)-4)
  ## number of sequences in the lineage
  n <- nrow(al_matrix) -2
  ## exclude germline sequence from the alignment
  al_matrix <- al_matrix[1:(n+1),]
    ## length of the sequence in the alignment
  l_seq <- length(al[[1]])

  type_matrix <- c()
    mutation_table <- c()
  l_full <- 3*(l_seq%/%3)
  ## positions in codon
  index <- c(1,2,3)
  Ss_anc <- 0
  Ns_anc <- 0
  sites_table <- c()
  ####################pos
  for (pos in 1:(3*(l_seq%/%3))){
```

```
  n_codon <- (2 + pos)%/%3

  origin <- paste0(al_matrix[(n+1), (3*n_codon-2):(3*n_codon)], collapse = "")
  variants <- unique(al_matrix[,pos])
  variants <- variants[variants != "-" & variants != al_matrix[(n+1), pos]]

  if (!str_detect(origin, "-") & pos%%3 == 1){
    Ss_anc <- Ss_anc + gc$Ss[which(gc$GeneticCode == toupper(origin))]
    Ns_anc <- Ns_anc + gc$Ns[which(gc$GeneticCode == toupper(origin))]
  }

  if (length(variants) > 0 & !str_detect(origin, "-")) {

    for (var in variants) {
      pos_in_codon <- pos %% 3
      if (pos_in_codon == 0) {pos_in_codon = 3}
      var_codon <- unlist(str_split(origin, ""))
      var_codon[pos_in_codon] <- var
      var_codon <- paste0(var_codon, collapse = "")

      ## synonymous SNP
      if (gc$AA[which(gc$GeneticCode == toupper(paste(origin, collapse = '')))] ==
         gc$AA[which(gc$GeneticCode == toupper(paste(var_codon, collapse = '')))]){
         type <- "S"
         S_codon <- 1
         N_codon <- 0
      } else {
         type <- "N"
         S_codon <- 0
         N_codon <- 1
      }

      mutation_table <- rbind(mutation_table,
                              c(n_codon, origin, var_codon, S_codon, N_codon,
                                gc$Ss[gc$GeneticCode == toupper(origin)],
                                gc$Ns[gc$GeneticCode == toupper(origin)],
                                sum(al_matrix[1:n,pos] == var)/n, type))
    }
  }
}
####################pos
mutation_table <- data.frame(mutation_table)
colnames(mutation_table) <- c("codon", "ancestor", "variant","S_l", "N_l", "Ss", "Ns",
                              "freq", "type")
mutation_table[, c(1:9)] <- sapply(mutation_table[, c(1:9)], as.character)
mutation_table[, c(1,4:9)] <- sapply(mutation_table[, c(1,4:9)], as.numeric)

## counting pnps
for (b in Bins) {
  mt <- mutation_table[mutation_table$freq <= b & mutation_table$freq > Bins_0[which(b==Bins)],]
  N_mut <- sum(mt$N_l)
  S_mut <- sum(mt$S_l)
  pnps <- ((N_mut+1)/Ns_anc) / ((S_mut+1)/Ss_anc)
```

```
    mutN_all <- sum(mutation_table$N_1)
    mutS_all <- sum(mutation_table$S_1)

    pnps_norm <- ( ((N_mut+1)/(5+mutN_all)) / Ns_anc) /
                 ( ((S_mut+1)/(5+mutS_all)) / Ss_anc)

    pnps_table <- rbind(pnps_table, c(lin, b, pnps, "all",
                                      lineage_composition$cluster[lineage_composition$file == lin]))
    pnps_table <- rbind(pnps_table, c(lin, b, pnps_norm, "norm_all",
                                       lineage_composition$cluster[lineage_composition$file == lin]))
  }

  ## weighted spectrum
  sfs_scaled <- c()
  for (b in bins){
   sfs_scaled <- rbind(sfs_scaled,
                  c(b, nrow(mutation_table[mutation_table$freq <= b & mutation_table$freq > (b - bin_l)
                                           mutation_table$type == "S",]),
                    nrow(mutation_table[mutation_table$freq <= b & mutation_table$freq > (b - bin_l) &
                                        mutation_table$type == "N",]),
                    nrow(mutation_table[mutation_table$freq <= b & mutation_table$freq > (b - bin_l),]))
  }
  sfs_scaled[,2] <- sfs_scaled[,2]/sum(sfs_scaled[,2])
  sfs_scaled[,3] <- sfs_scaled[,3]/sum(sfs_scaled[,3])
  sfs_scaled[,4] <- sfs_scaled[,4]/sum(sfs_scaled[,4])
  sfs_all <- rbind(sfs_all, cbind(sfs_scaled, rep(lin, 20), rep(n, 20),
                                  rep(lineage_composition$cluster[lineage_composition$file == lin], 20)


}

pnps_table <- data.frame(pnps_table)
colnames(pnps_table) <- c("clone", "scale", "pnps", "calculation", "cluster")
pnps_table[, c(1:4)] <- sapply(pnps_table[, c(1:4)], as.character)
pnps_table[, c(3)] <- sapply(pnps_table[, c(3)], as.numeric)

sfs_all <- data.frame(sfs_all)
colnames(sfs_all) <- c("scale", "syn", "nonsyn", "all", "lineage_id", "size", "cluster")
sfs_all[, c(1:7)] <- sapply(sfs_all[, c(1:7)], as.character)
sfs_all[, c(1:4,6)] <- sapply(sfs_all[, c(1:4,6)], as.numeric)

p_pnps_all <- ggplot(pnps_table[(pnps_table$scale == "1.2" & pnps_table$calculation == "all"),],
                     aes(x = cluster, y = pnps, fill = cluster)) +
  geom_quasirandom(width=0.3, size = 0.2, color = "grey30", alpha = 0.8) +
  geom_boxplot(alpha = 0.8, outlier.colour = NA) +
  stat_compare_means(comparisons = cluster_comparisons, method = "wilcox.test", size = 4, paired = FALSE
  scale_y_log10(name=TeX("$\\pi$N$\\pi$S"), limits = c(0.2, 1.5)) +
  scale_x_discrete(name = "") +
  scale_fill_manual(values = c("HBmem" = "mediumpurple4", "LBmem" = "tan1"), name = "") +
  my_theme
p_pnps_all
```

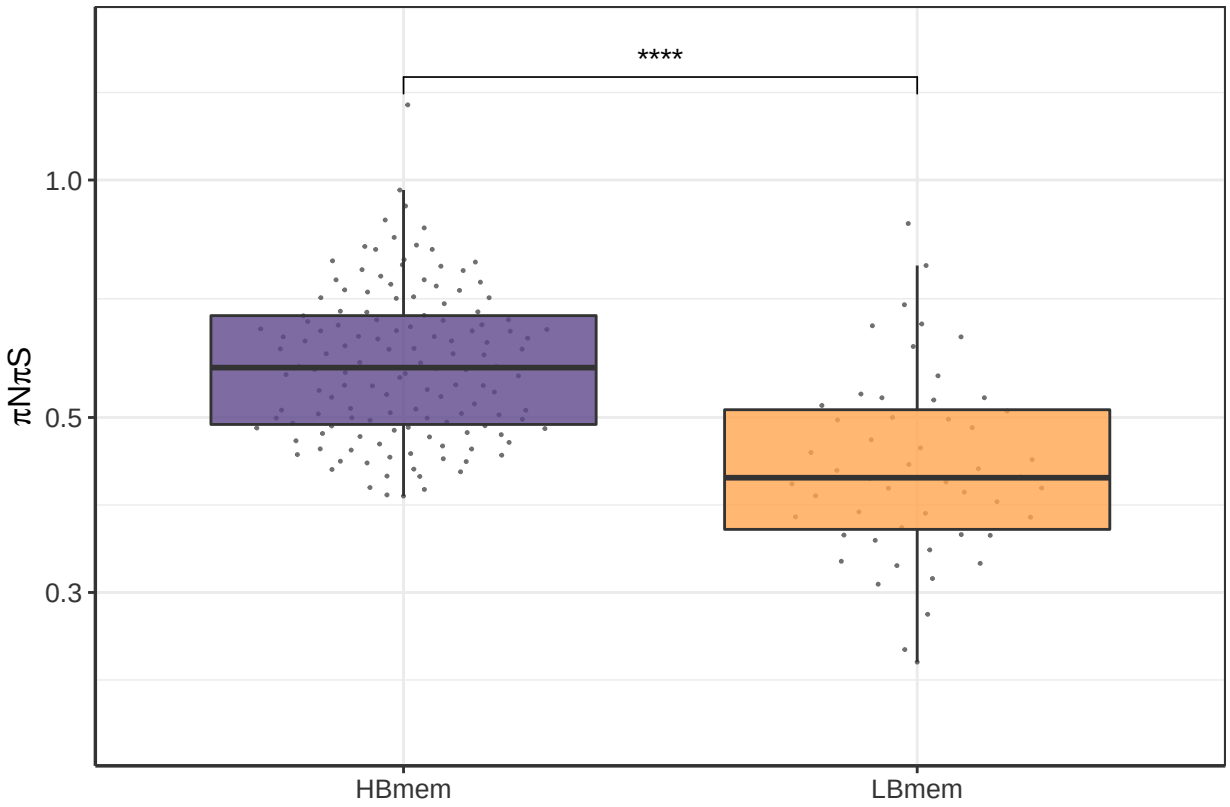


Fig. 5D: Averaged site frequency spectrum for HBmem and LBmem clonal groups;

```
#bins <- seq(0.05, 1, by=0.05)

sfs_all_plot <- c()
#groups_dop <- c("1", "2")
for (bin in bins){
  for (cluster in c("HBmem", "LBmem")){
    sfs_all_plot <- rbind(sfs_all_plot,
                c(bin, mean(sfs_all$all[sfs_all$scale == bin & sfs_all$cluster == cluster]),
      sd(sfs_all$all[sfs_all$scale == bin & sfs_all$cluster == cluster]), cluster))

  }
}
sfs_all_plot <- data.frame(sfs_all_plot)
colnames(sfs_all_plot) <- c("bin", "s", "sd", "cluster")

#sfs_all_plot$site <- factor(sfs_all_plot$site, labels = c("Nonsyn. sites", "Syn. sites"))
sfs_all_plot[, c(1:4)] <- sapply(sfs_all_plot[, c(1:4)], as.character)
sfs_all_plot[, c(1:3)] <- sapply(sfs_all_plot[, c(1:3)], as.numeric)

k <- 1/bins/sum(1/bins)
k2 <- 1/bins^2/sum(1/bins^2)

p_sfs_all <- ggplot(sfs_all_plot, aes(x = bin, y = s, color = cluster)) +
```

```
  geom_point(size = 2, alpha = 0.6) + geom_smooth(se = FALSE, size = 1.5, alpha = 0.4) +
 geom_segment(aes(x = bins[5], y = k[5], xend = bins[8], yend = k[8]),
              linetype="dashed", size=0.6, color = "grey20") +
  geom_text(aes(x=bins[8] + 0.02, y=k[8] + 0.02,  label = "x^{-1}"), parse=TRUE, color = "grey20") +
  geom_segment(aes(x = bins[3], y = k2[3], xend = bins[5], yend = k2[5]),
              linetype="dashed", size=0.6, color = "grey20") +
   geom_text(aes(x=bins[3] + 0.02, y=k2[5] ,  label = "x^{-2}"), parse=TRUE, color = "grey20") +
  scale_x_continuous(trans = 'log10', name=("Derived SHM frequency")) +
   scale_y_continuous(trans = 'log10', name=("Probability density")) +
  my_colors +
  #facet_wrap(~site, nrow = 2) +
  my_theme
p_sfs_all
```

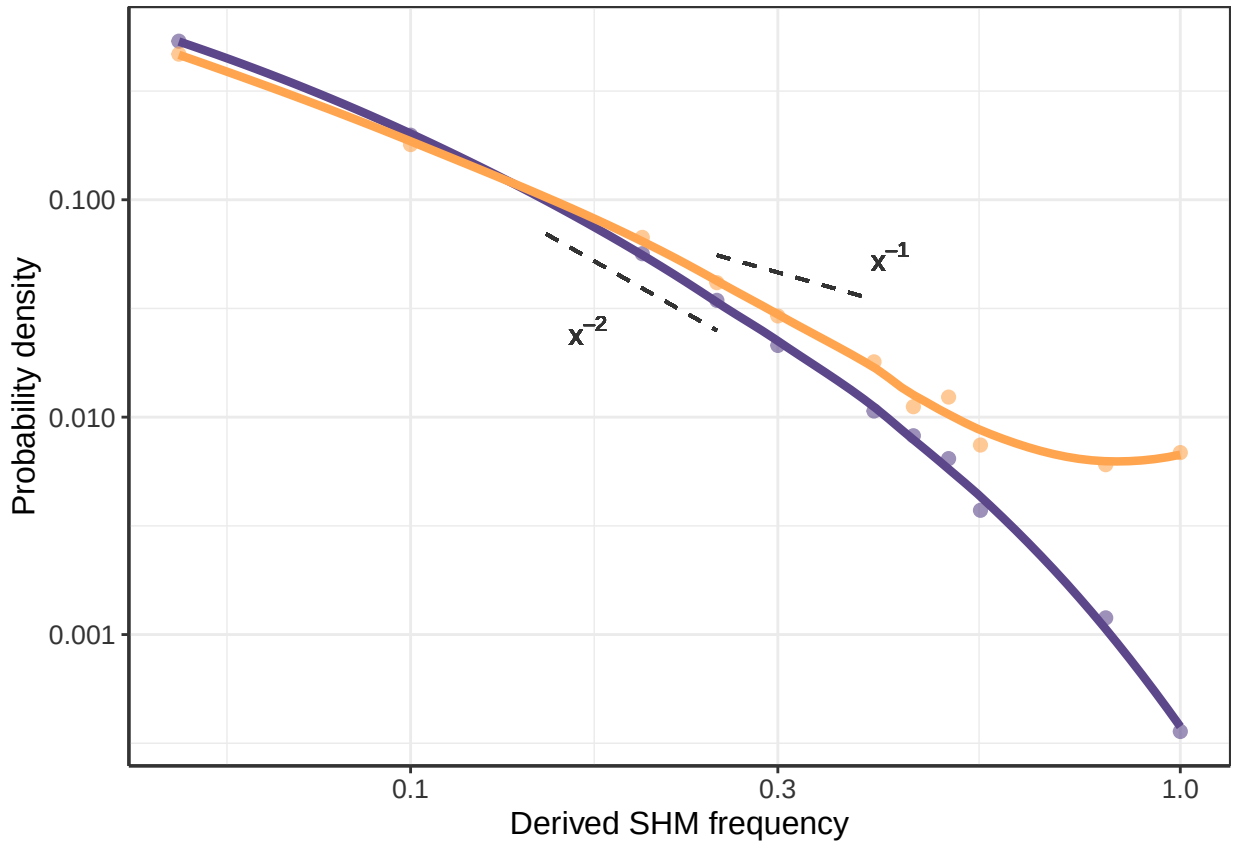
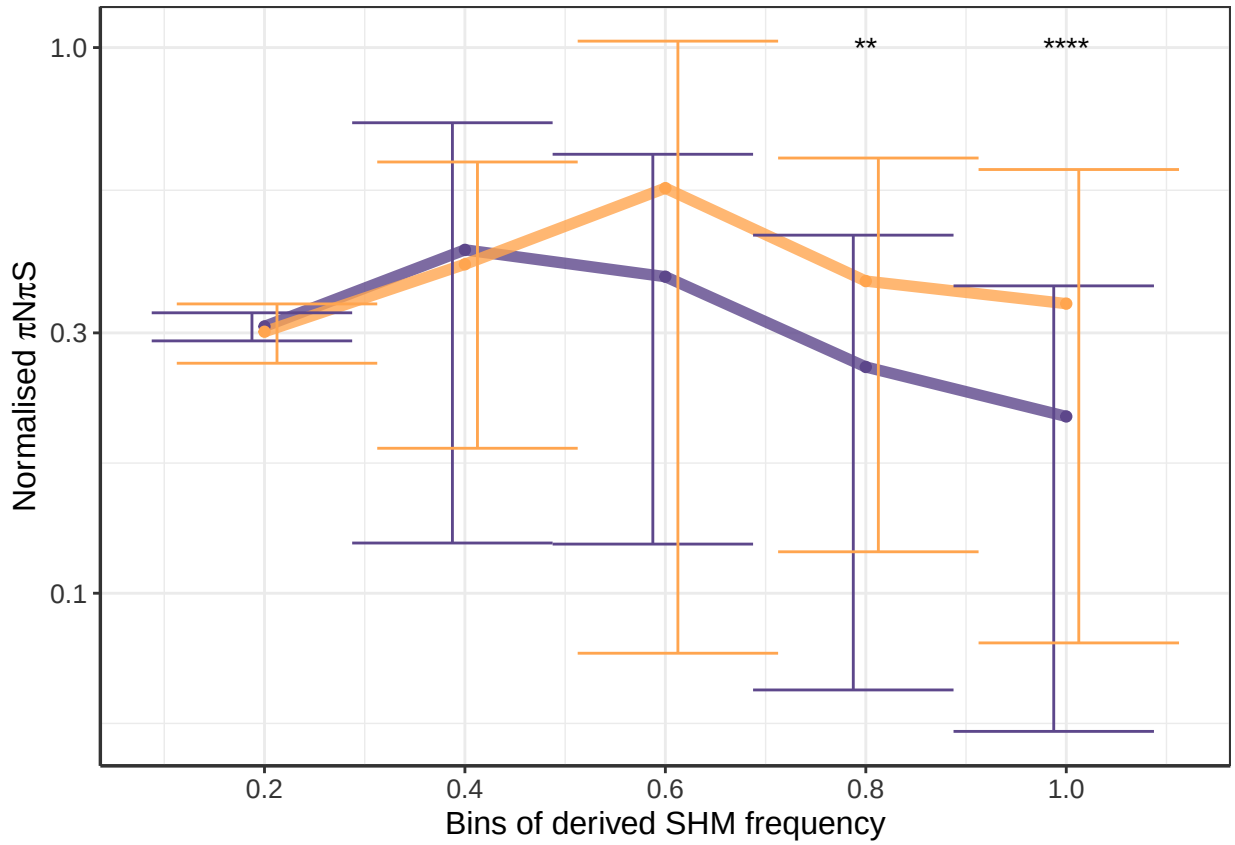


Fig. 5E: Comparison of normalised $\pi N \pi S$ HBmem and LBmem clonal groups in bins of SHM frequencies. The number of polymorphisms in each bin is normalised by the overall number of polymorphisms in a corresponding clonal group; Comparisons were performed by Mann-Whitney test with Bonferroni-Holm multiple testing correction, notation of the level of significance is the following: * - $p \leq 0.05$, ** - $p \leq 0.01$, * - $p \leq 10^{-3}$, **** - $p \leq 10^{-4}$.**

```
data_summary <- function(data, varname, groupnames){
  require(plyr)
  summary_func <- function(x, col){
    c(mean = mean(x[[col]], na.rm=TRUE),
      sd = sd(x[[col]], na.rm=TRUE))
```

```
  }
  data_sum<-ddply(data, groupnames, .fun=summary_func,
                  varname)
  data_sum <- rename(data_sum, c("mean" = varname))
 return(data_sum)
}

df_pnps <- data_summary(pnps_table[(pnps_table$calculation == "norm_all") & pnps_table$scale != "1.2",]
                        varname="pnps", groupnames=c("scale", "calculation", "cluster"))

p_pnps_sfs <- ggplot(df_pnps, aes(x=as.numeric(scale), y=pnps, group=cluster, color=cluster)) +
  geom_line(size = 2, alpha = 0.8) +
  geom_point()+
  geom_errorbar(aes(ymin=pnps-sd, ymax=pnps+sd), width=.4,
                position=position_dodge(0.05)) +
  annotate("text", x = 0.8, y = 1, label= "**", size = 4) +
  annotate("text", x = 1, y = 1, label = "****", size = 4) +
  scale_y_log10(name=TeX("Normalised $\\pi$N$\\pi$S")) +
  scale_x_continuous(name=("Bins of derived SHM frequency"), breaks = Bins[1:5]) +
  my_colors +
  my_theme
p_pnps_sfs
```



Mann-Whitney test with Bonferroni-Holm multiple testing correction for comparison of normalised $\pi N \pi S$ in bins of SHM frequency.

```r
bins <- seq(0.2, 1, 0.2)
for (bin in bins){
   print(paste0("Frequency bin = ", bin))
  x <- wilcox.test(pnps_table$pnps[pnps_table$scale == as.character(bin) & pnps_table$calculation == "no
                                   & pnps_table$cluster == "HBmem"],
                   pnps_table$pnps[pnps_table$scale == as.character(bin) & pnps_table$calculation == "norm_al
                                   & pnps_table$cluster == "LBmem"])
   print(paste0("Correction: p-value = ", x$p.value*5))
}
```

```
## [1] "Frequency bin = 0.2"
## [1] "Correction: p-value = 0.788363921596477"
## [1] "Frequency bin = 0.4"
## [1] "Correction: p-value = 3.5276628315624"
## [1] "Frequency bin = 0.6"
## [1] "Correction: p-value = 0.207109183029783"
## [1] "Frequency bin = 0.8"
## [1] "Correction: p-value = 0.00169783462825742"
## [1] "Frequency bin = 1"
## [1] "Correction: p-value = 1.90855795327008e-05"
```