Rearrangement and selection of epitope-specific T-cell clonotypes

```
library(data.table)
library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
## data.table + dplyr code now lives in dtplyr.
## Please library(dtplyr)!
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
library(scales)
library(stringr)
```

Rearrangement model suppl

Pre-processing and selecting epitopes

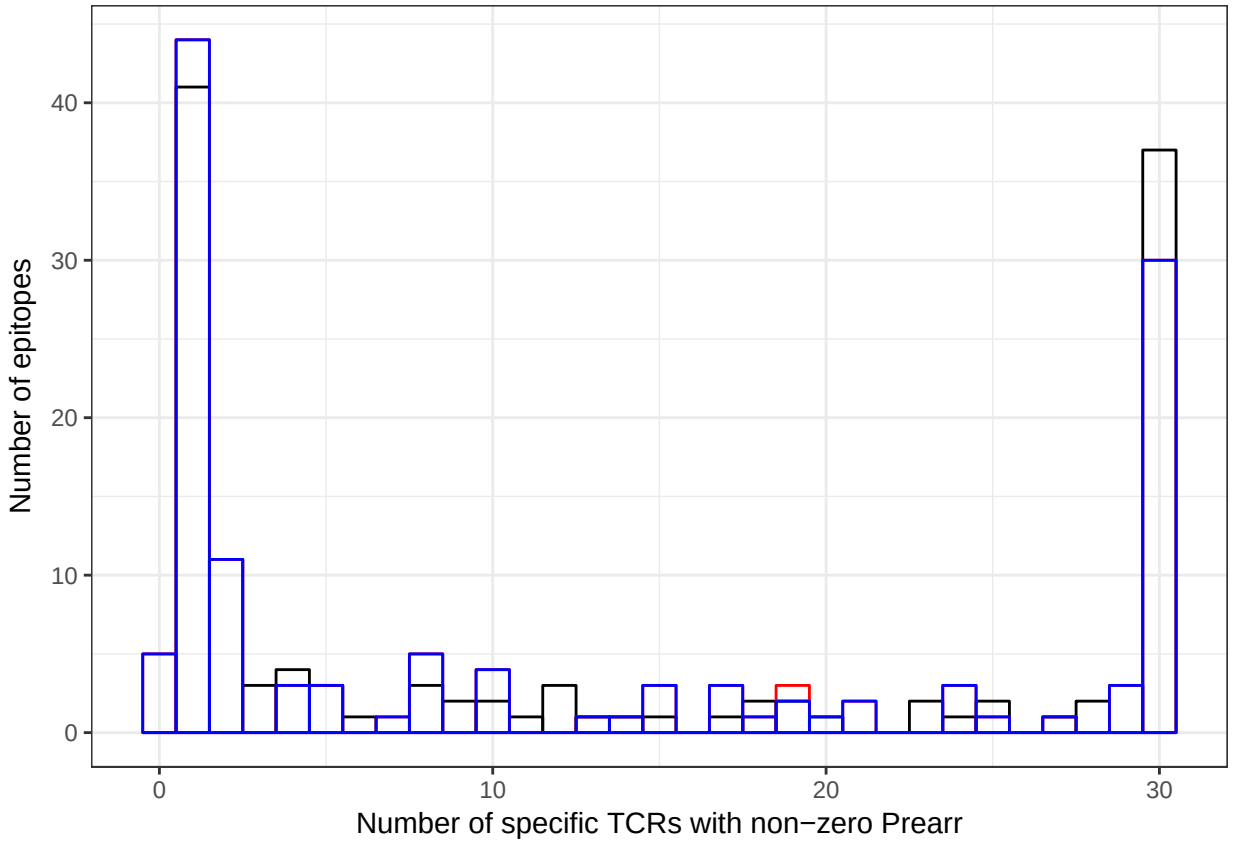
```
dt.vdjdb = fread("rearr_model/VDJDB_fullP_rob_ageing.txt")
```

Fraction of non-zero Prearr TCRs

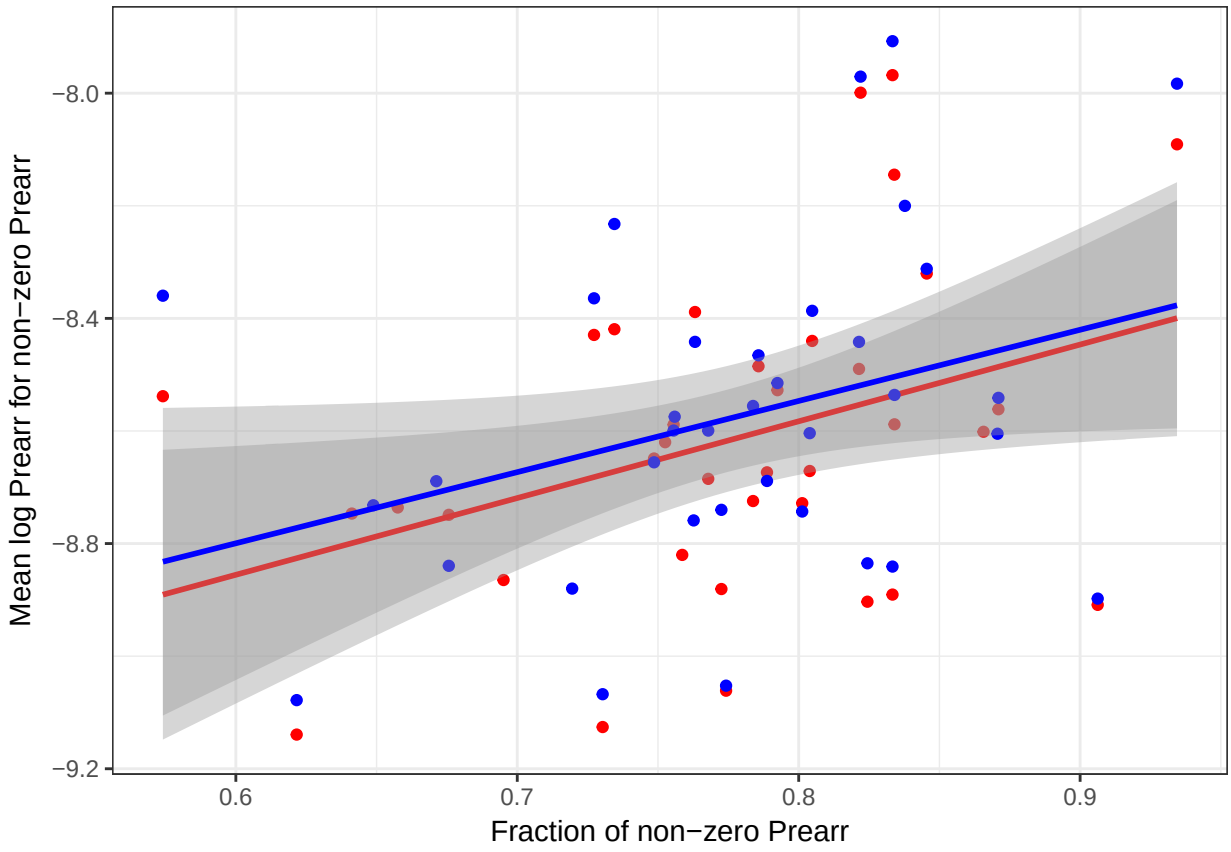
```
dt.epicount1 = dt.vdjdb %>%
  group_by(antigen.epitope) %>%
  summarise(total = n(),
            total_r_a = sum(genP_0mism_ageing > 0),
            frac_r_a = sum(genP_0mism_ageing > 0) / n(),
            log_mean_p_a = sum(ifelse(genP_0mism_ageing > 0, log10(genP_0mism_ageing), 0)) / sum(genP_0
            total_r_h = sum(genP_0mism_rob > 0),
```

```
            frac_r_h = sum(genP_0mism_rob > 0) / n(),
            log_mean_p_h = sum(ifelse(genP_0mism_rob > 0, log10(genP_0mism_rob), 0)) / sum(genP_0mism_r
```

```
ggplot(dt.epicount1) +
  geom_histogram(aes(pmin(30,total)), binwidth = 1, color= "black", fill=NA) +
  geom_histogram(aes(pmin(30,total_r_a)), binwidth = 1, color = "red", fill=NA) +
  geom_histogram(aes(pmin(30,total_r_h)), binwidth = 1, color = "blue", fill=NA) +
  ylab("Number of epitopes") + xlab("Number of specific TCRs with non-zero Prearr") +
  theme_bw()
```




```
ggplot(dt.epicount1 %>% filter(total > 30)) +
  geom_point(aes(x = frac_r_a, y = log_mean_p_a), color = "red") +
  geom_smooth(aes(x = frac_r_a, y = log_mean_p_a), method = "lm", color = "red") +
  geom_point(aes(x = frac_r_h, y = log_mean_p_h), color = "blue") +
  geom_smooth(aes(x = frac_r_h, y = log_mean_p_h), method = "lm", color = "blue") +
  ylab("Mean log Prearr for non-zero Prearr") + xlab("Fraction of non-zero Prearr") +
  theme_bw()
```




```
print(with(dt.epicount1 %>% filter(total > 30), cor.test(frac_r_a, log_mean_p_a)))
```

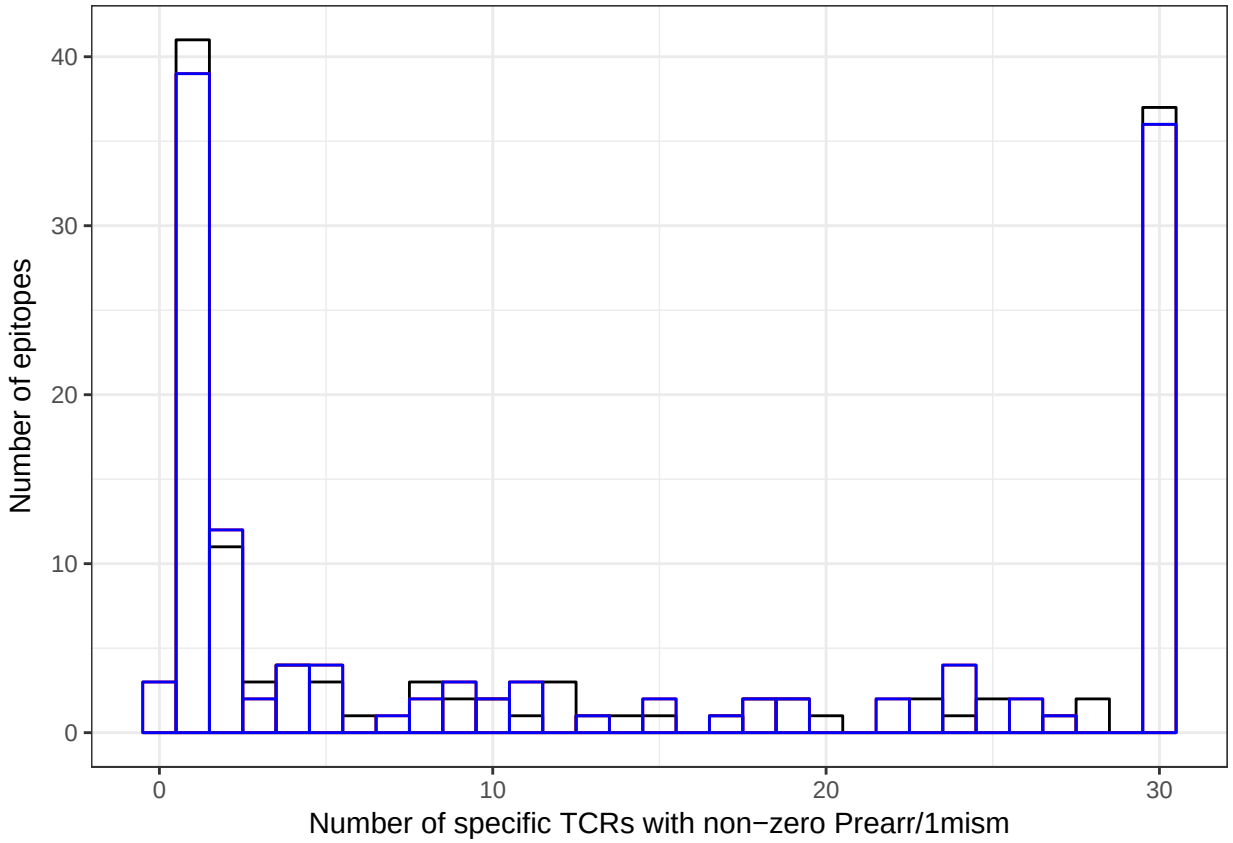
```
##
##  Pearson's product-moment correlation
##
## data:  frac_r_a and log_mean_p_a
## t = 2.3229, df = 34, p-value = 0.0263
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.04730466 0.62288904
## sample estimates:
##       cor
## 0.3700887
```

```
print(with(dt.epicount1 %>% filter(total > 30), cor.test(frac_r_h, log_mean_p_h)))
```

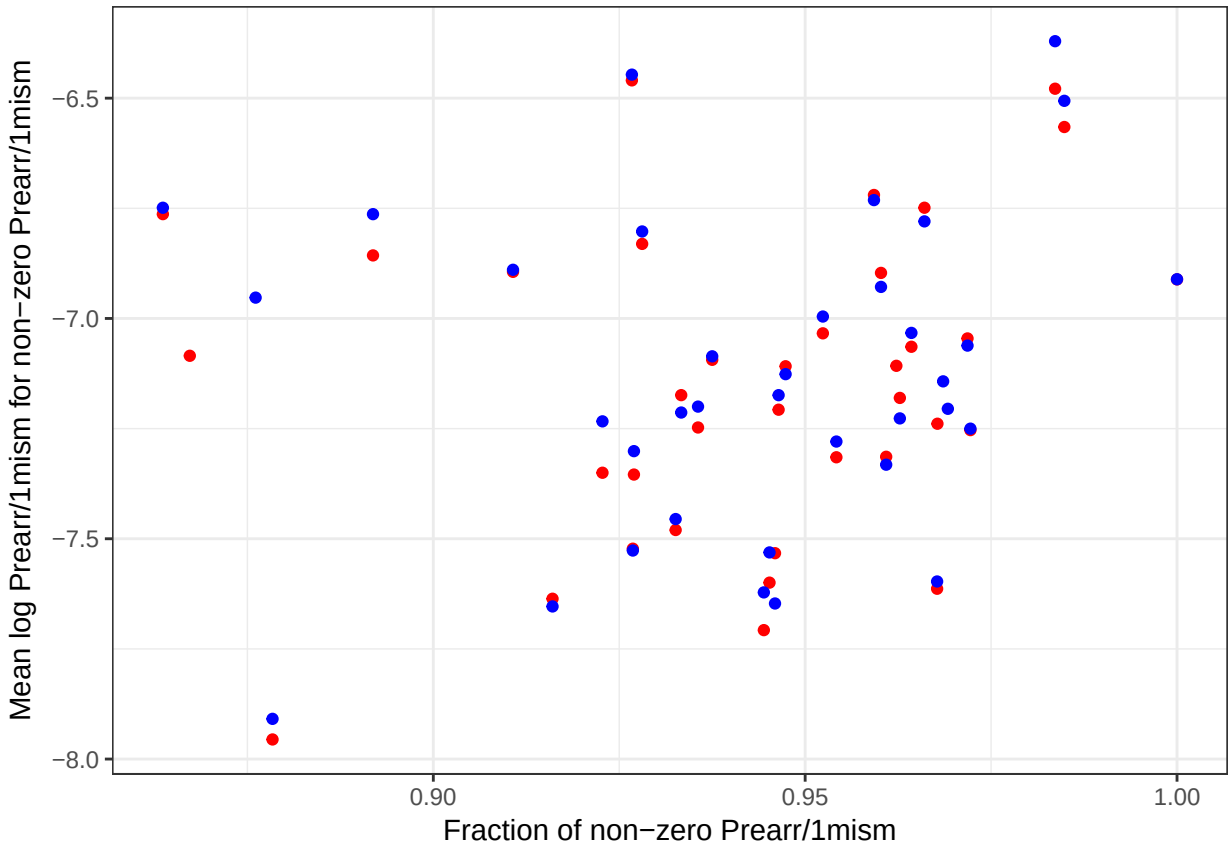
```
##
##  Pearson's product-moment correlation
##
## data:  frac_r_h and log_mean_p_h
## t = 2.0397, df = 34, p-value = 0.04921
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.00185248 0.59425881
## sample estimates:
##       cor
## 0.3301873
```

Fraction of non-zero Prearr TCRs

```
dt.epicount2 = dt.vdjdb %>%
  group_by(antigen.epitope) %>%
  summarise(total = n(),
            total_r_a = sum(genP_1mism_ageing > 0),
            frac_r_a = sum(genP_1mism_ageing > 0) / n(),
            log_mean_p_a = sum(ifelse(genP_1mism_ageing > 0, log10(genP_1mism_ageing), 0)) / sum(genP_1
            total_r_h = sum(genP_1mism_rob > 0),
            frac_r_h = sum(genP_1mism_rob > 0) / n(),
            log_mean_p_h = sum(ifelse(genP_1mism_rob > 0, log10(genP_1mism_rob), 0)) / sum(genP_1mism_r

ggplot(dt.epicount2) +
  geom_histogram(aes(pmin(30,total)), binwidth = 1, color= "black", fill=NA) +
  geom_histogram(aes(pmin(30,total_r_a)), binwidth = 1, color = "red", fill=NA) +
  geom_histogram(aes(pmin(30,total_r_h)), binwidth = 1, color = "blue", fill=NA) +
  ylab("Number of epitopes") + xlab("Number of specific TCRs with non-zero Prearr/1mism") +
  theme_bw()
```




```
ggplot(dt.epicount2 %>% filter(total > 30)) +
  geom_point(aes(x = frac_r_a, y = log_mean_p_a), color = "red") +
  geom_point(aes(x = frac_r_h, y = log_mean_p_h), color = "blue") +
  ylab("Mean log Prearr/1mism for non-zero Prearr/1mism") + xlab("Fraction of non-zero Prearr/1mism") +
  theme_bw()
```




```
print(with(dt.epicount2 %>% filter(total > 30), cor.test(frac_r_a, log_mean_p_a)))
```

```
##
##  Pearson's product-moment correlation
##
## data:  frac_r_a and log_mean_p_a
## t = 1.1525, df = 34, p-value = 0.2572
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1438013  0.4911440
## sample estimates:
##       cor
## 0.1938948
```

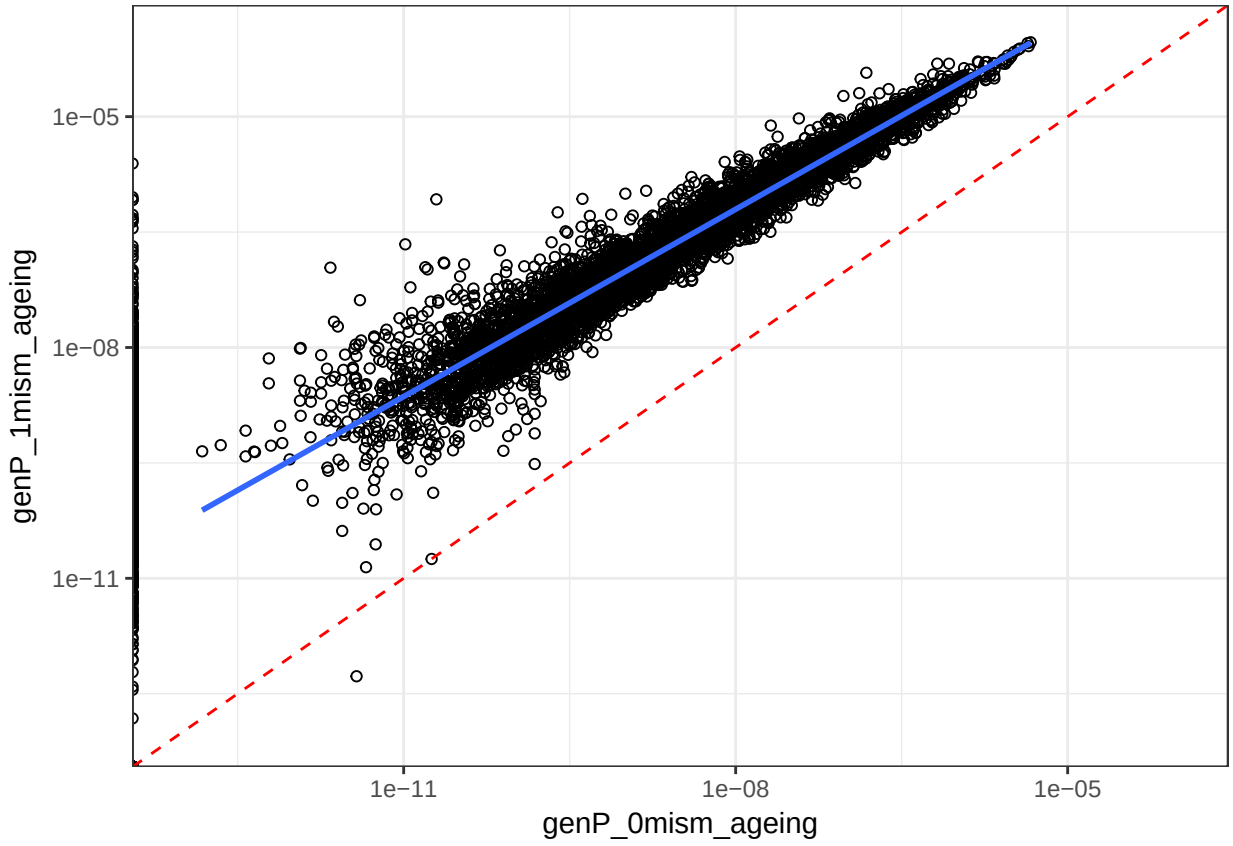
```
print(with(dt.epicount2 %>% filter(total > 30), cor.test(frac_r_h, log_mean_p_h)))
```

```
##
##  Pearson's product-moment correlation
##
## data:  frac_r_h and log_mean_p_h
## t = 0.91588, df = 34, p-value = 0.3662
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1826781  0.4602434
## sample estimates:
##       cor
## 0.1551704
```

Correlation between rearrangement probs from two models, with and without mismatches

```
ggplot(dt.vdjdb, aes(x=genP_0mism_ageing,y=genP_1mism_ageing)) +
  geom_point(shape=21) +
  geom_smooth(method = "lm") +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  scale_x_log10(limits=c(1e-13,1e-4)) +
  scale_y_log10(limits=c(1e-13,1e-4)) +
  theme_bw()
```

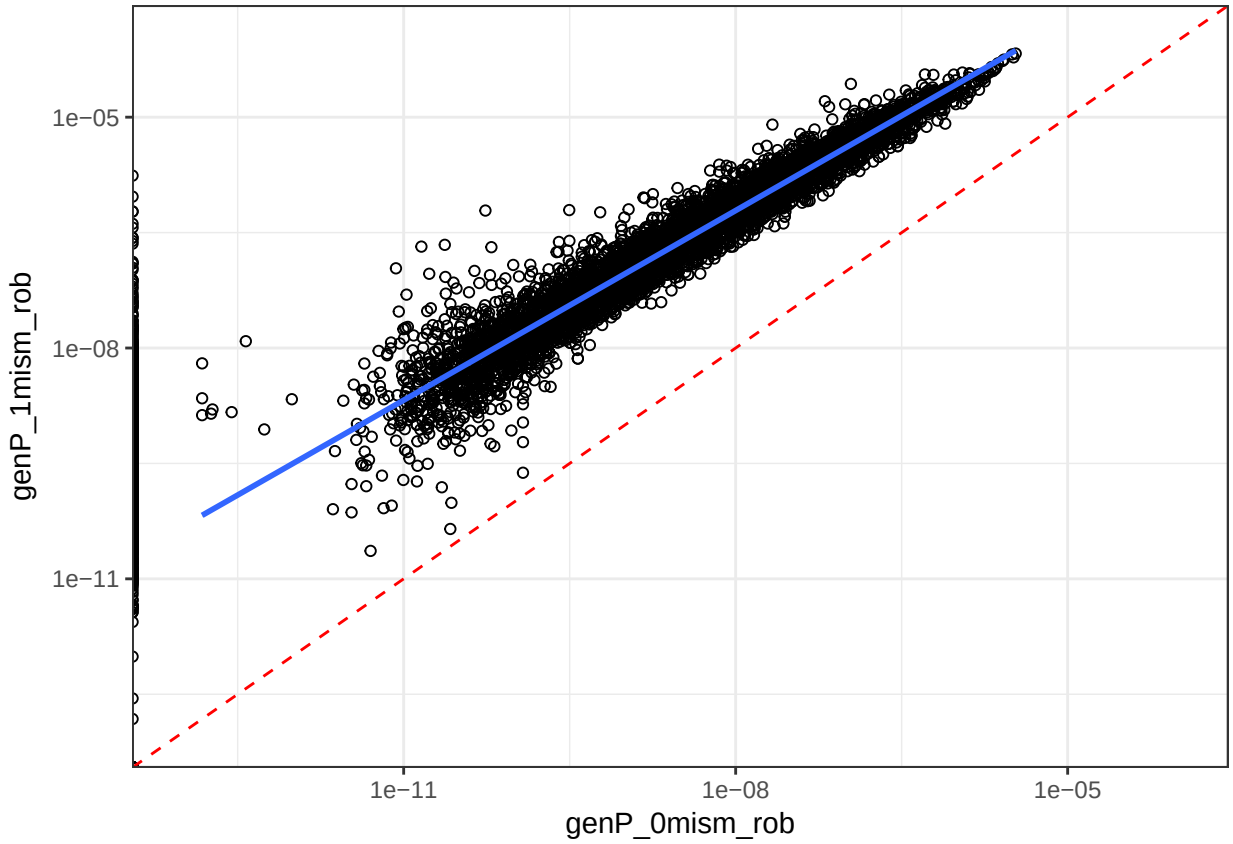
```
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 1447 rows containing non-finite values (stat_smooth).
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
ggplot(dt.vdjdb, aes(x=genP_0mism_rob,y=genP_1mism_rob)) +
  geom_point(shape=21) +
  geom_smooth(method = "lm") +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  scale_x_log10(limits=c(1e-13,1e-4)) +
  scale_y_log10(limits=c(1e-13,1e-4)) +
  theme_bw()
```

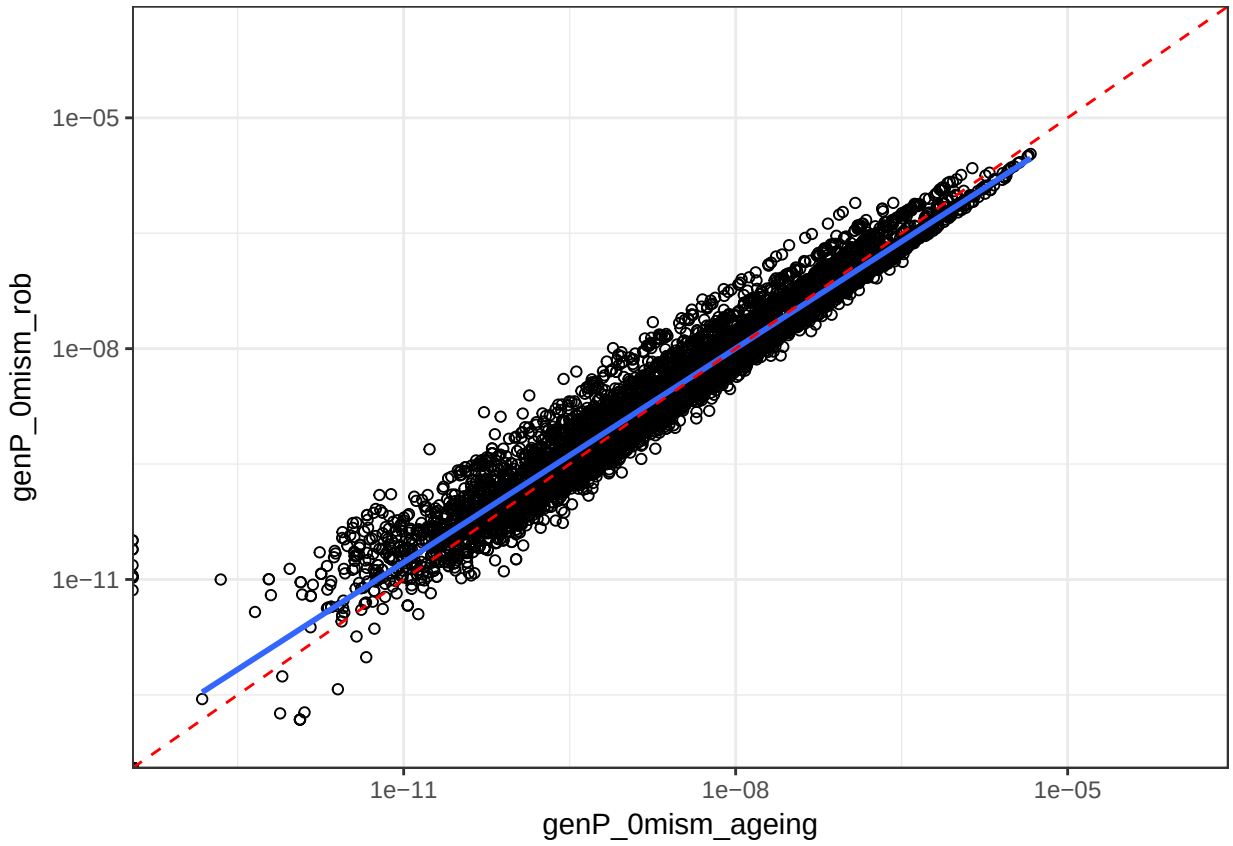
```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 1441 rows containing non-finite values (stat_smooth).
## Warning: Removed 6 rows containing missing values (geom_point).
```



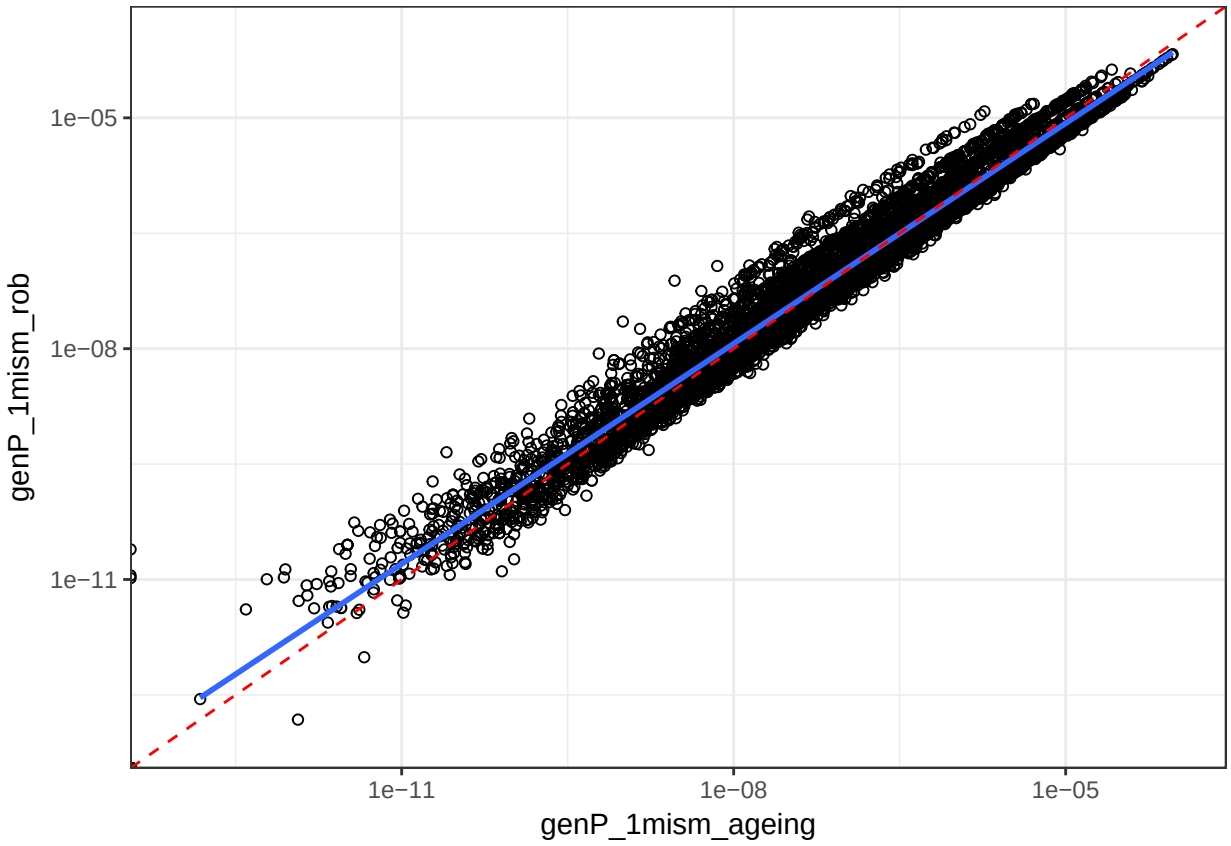

```
ggplot(dt.vdjdb, aes(x=genP_0mism_ageing,y=genP_0mism_rob)) +
  geom_point(shape=21) +
  geom_smooth(method = "lm") +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  scale_x_log10(limits=c(1e-13,1e-4)) +
  scale_y_log10(limits=c(1e-13,1e-4)) +
  theme_bw()
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 1450 rows containing non-finite values (stat_smooth).
## Warning: Removed 3 rows containing missing values (geom_point).
```




```
ggplot(dt.vdjdb, aes(x=genP_1mism_ageing,y=genP_1mism_rob)) +
  geom_point(shape=21) +
  geom_smooth(method = "lm") +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  scale_x_log10(limits=c(1e-13,1e-4)) +
  scale_y_log10(limits=c(1e-13,1e-4)) +
  theme_bw()
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 375 rows containing non-finite values (stat_smooth).
## Warning: Removed 3 rows containing missing values (geom_point).
```



Annotated data suppl

```
df.hip.raw = fread("zcat annotations/hip_annot_1.txt.gz")
```

```
##
Read 67.3% of 311817 rows
Read 311817 rows and 32 (of 32) columns from 0.077 GB file in 00:00:03
```

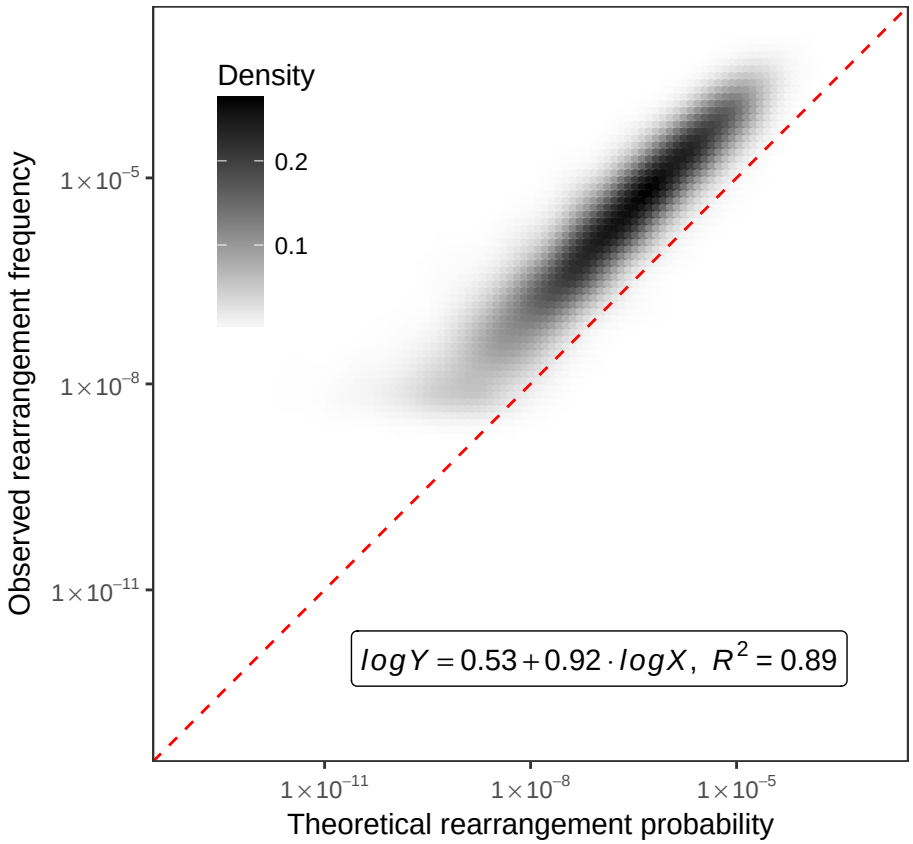
```
HIP_R = sum(fread("annotations/hip_stats.txt")$diversity)
HIP_S = sum(nrow(fread("annotations/hip_stats.txt")))

df.hip = df.hip.raw %>%
  group_by(cdr3, antigen.epitope, antigen.species, mhc.a, mhc.b, mhc.class) %>%
  summarise(count = sum(occurrences),
            freq = sum(occurrences) / HIP_R,
            incidence = sum(incidence) / HIP_S / n(),
            convergence = sum(convergence) / sum(occurrences)) %>%
  merge(dt.vdjdb %>% select(cdr3, antigen.epitope, antigen.species, mhc.a, mhc.b, mhc.class,
                            genP_1mism_rob),
        by = c("cdr3", "antigen.epitope", 'antigen.species', "mhc.a", 'mhc.b', 'mhc.class'), all.y = T) %>%
  mutate(count = ifelse(is.na(count), 0, count),
         freq = ifelse(is.na(freq), 0, freq),
         incidence = ifelse(is.na(incidence), 0, incidence),
         convergence = ifelse(is.na(convergence), 0, convergence))
```

```
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)
}

lm_eqn = function(ll){
    eq = substitute(italic(logY) == a + b %.% italic(logX)*","~~italic(R)^2~"="~r2,
         list(a = format(coef(ll)[1], digits = 2),
              b = format(coef(ll)[2], digits = 2),
             r2 = format(summary(ll)$r.squared, digits = 2)))
    as.character(as.expression(eq))
}

rr = with(df.hip %>% filter(genP_1mism_rob > 0, freq > 0), cor.test(log10(genP_1mism_rob), log10(freq)))
ll = with(df.hip %>% filter(genP_1mism_rob > 0, freq > 0), lm(log10(freq) ~ log10(genP_1mism_rob)))

p1=ggplot(df.hip, aes(x=genP_1mism_rob,y=freq)) +
  stat_density_2d(geom = "hex", aes(fill=..density..), contour = F) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  annotate(geom = "label", x = 1e-7, y = 1e-12, label = lm_eqn(ll), parse = T) +
  scale_x_log10("Theoretical rearrangement probability", limits=c(1e-13,1e-3), labels = fancy_scientific
  scale_y_log10("Observed rearrangement frequency", limits=c(1e-13,1e-3), labels = fancy_scientific) +
  scale_fill_gradient("Density", low = "white", high = "black") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        aspect.ratio = 1, legend.position = c(0.15, 0.75))
p1
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 920 rows containing non-finite values (stat_density2d).
```



Expected and observed difference between occurrences

```
obs_ratio = as.numeric(10^(-coef(ll)[1]))

mean_stop_prob = (df.hip.raw %>%
  filter(DStart > 0) %>%
  select(cdr3nt, VEnd, DStart, DEnd, JStart) %>%
  unique %>%
  summarise(mean_stop_prob = mean((61/64)^round((DStart - VEnd + JStart - DEnd + 2)/3))))$mean_stop_prob

coding_prob = (1-2/3) * mean_stop_prob

obs_ratio
```

```
## [1] 0.2939306
```

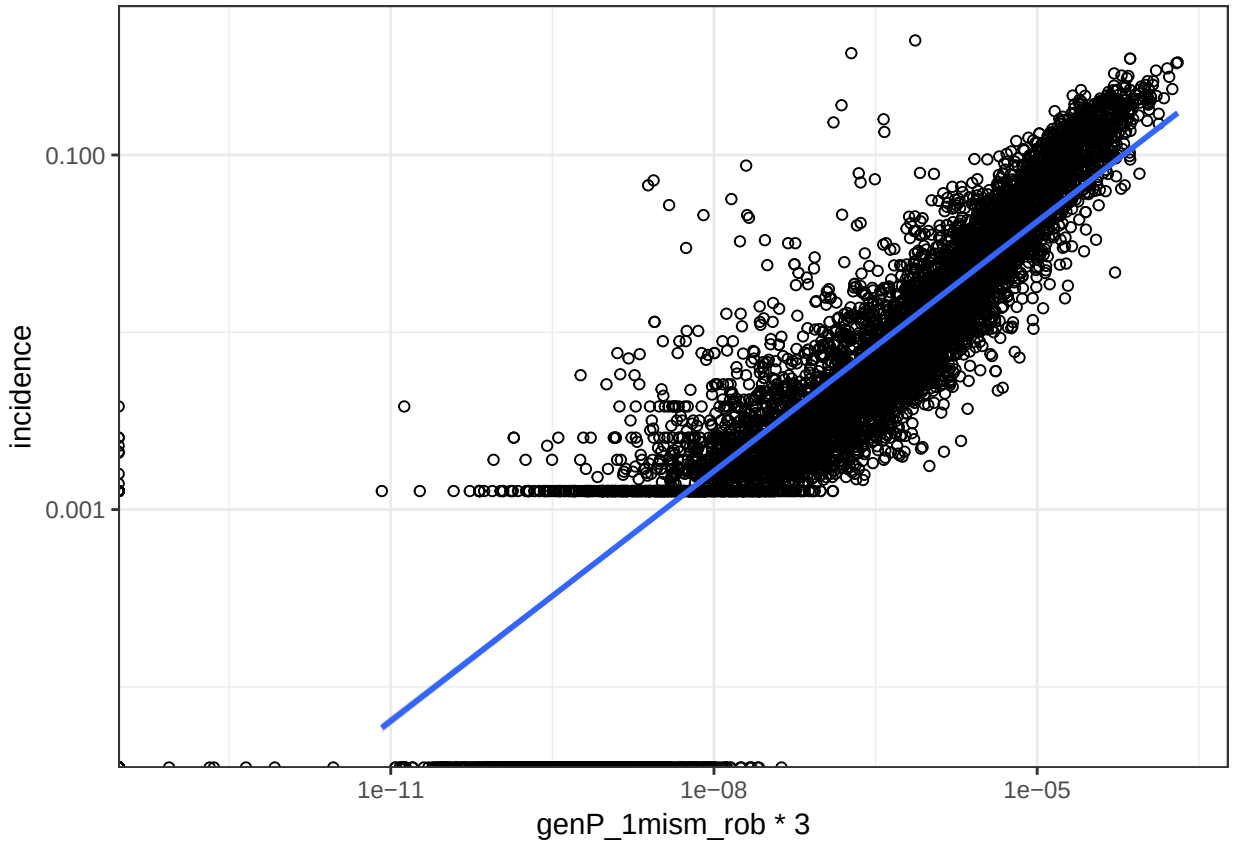
```
coding_prob
```

```
## [1] 0.298463
```

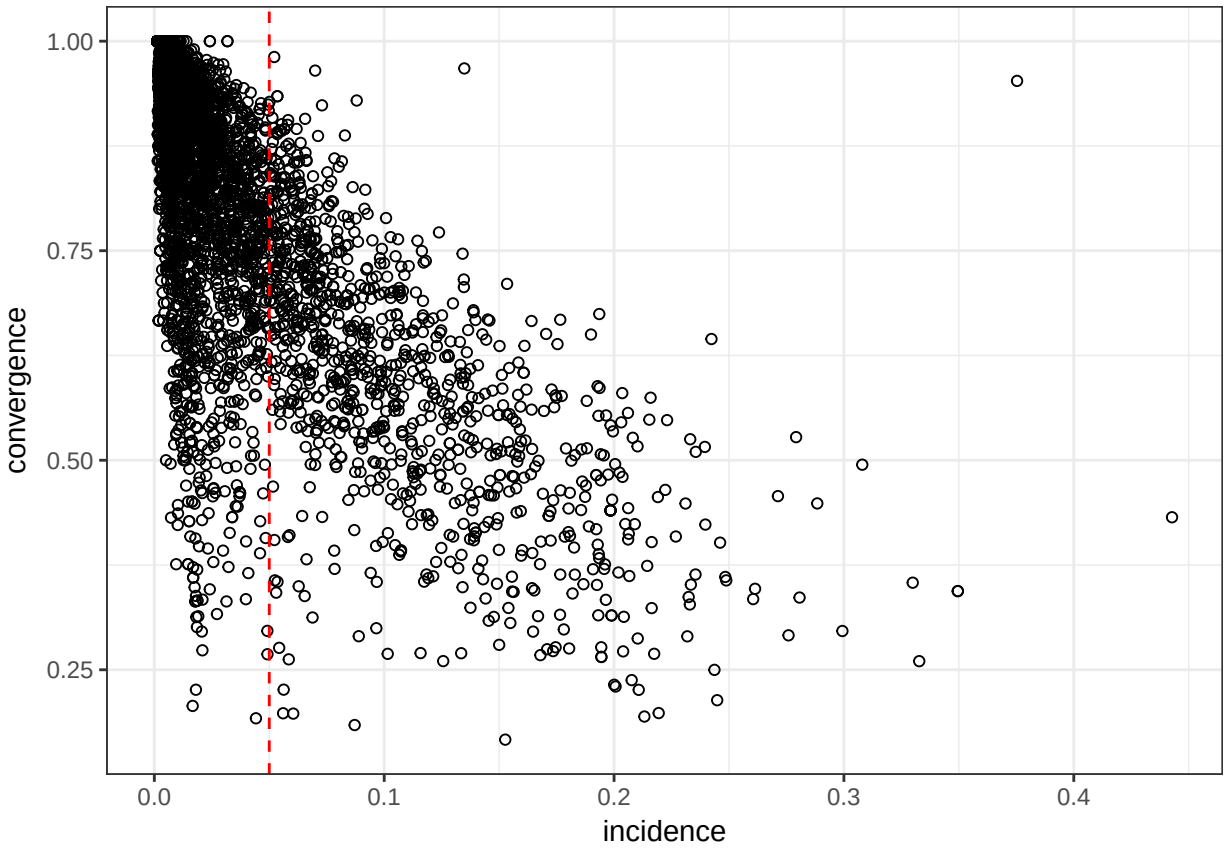
```
ggplot(df.hip, aes(x=genP_1mism_rob*3,y=incidence)) +
  geom_point(shape=21) +
  geom_smooth(method = "lm") +
  scale_x_log10() +
  scale_y_log10() +
  scale_color_discrete(guide=F) +
  theme_bw()
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

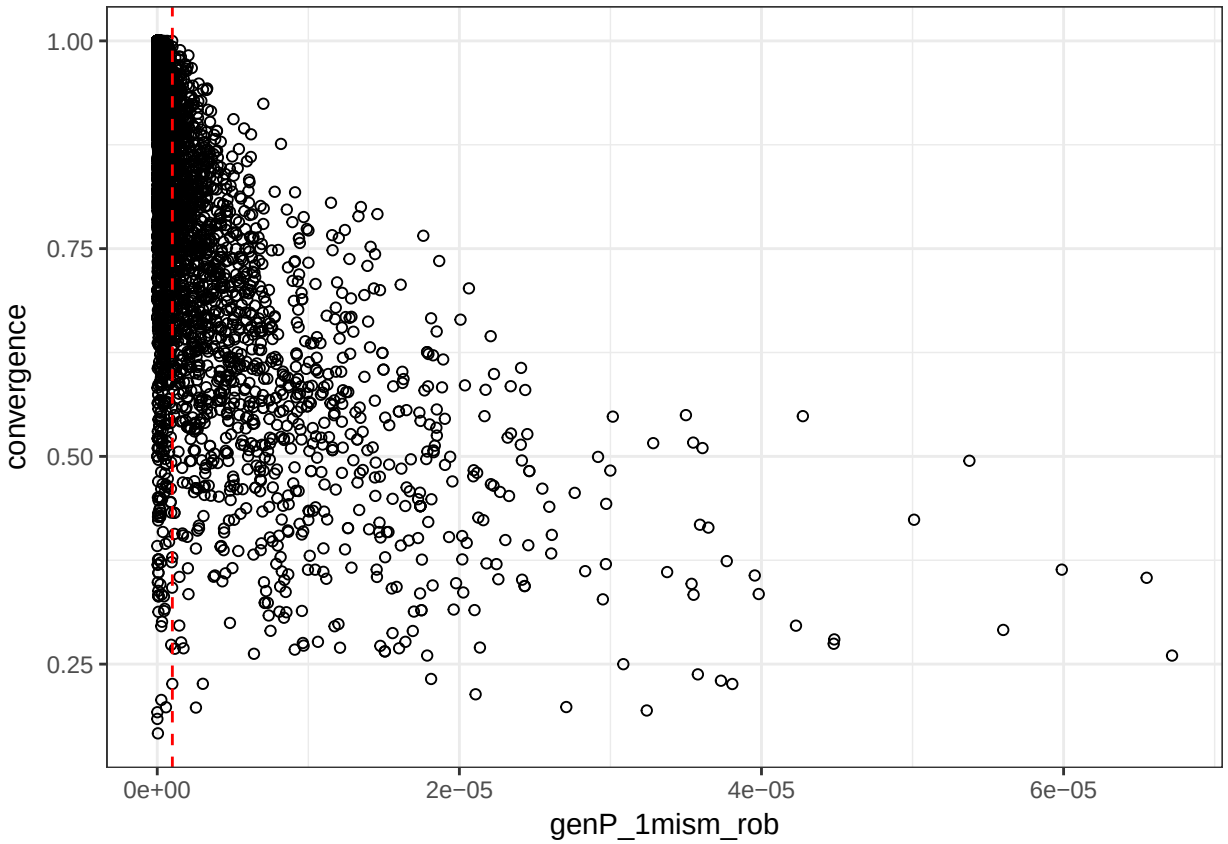
```
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Transformation introduced infinite values in continuous x-axis
## Warning: Transformation introduced infinite values in continuous y-axis
## Warning: Removed 920 rows containing non-finite values (stat_smooth).
```




```
ggplot(df.hip %>% filter(incidence > 0), aes(x=incidence, y=convergence)) +
  geom_point(shape=21) +
 # geom_smooth() +
  geom_vline(xintercept = 0.05, color = "red", linetype = "dashed") +
  scale_color_discrete(guide=F) +
  theme_bw()
```




```
ggplot(df.hip %>% filter(incidence > 0), aes(x=genP_1mism_rob, y=convergence)) +
  geom_point(shape=21) +
 # geom_smooth() +
  geom_vline(xintercept = 1e-6, color = "red", linetype = "dashed") +
  scale_color_discrete(guide=F) +
  theme_bw()
```




```
p2=ggplot(df.hip %>% filter(convergence > 0), aes(x=genP_1mism_rob,y=convergence)) +
  stat_density_2d(geom = "tile", aes(fill=..density..), contour = F) +
  geom_smooth(color = "red") +
  scale_x_log10("Theoretical rearrangement probability", limits=c(1e-13,1e-3), labels = fancy_scientifi
  scale_y_continuous("Unique variants per rearrangement (convergence)", expand = c(0,0), limits = c(-0.
  scale_fill_gradient("Density", low = "white", high = "black") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        aspect.ratio = 1, legend.position = c(0.15, 0.25))
p2
```

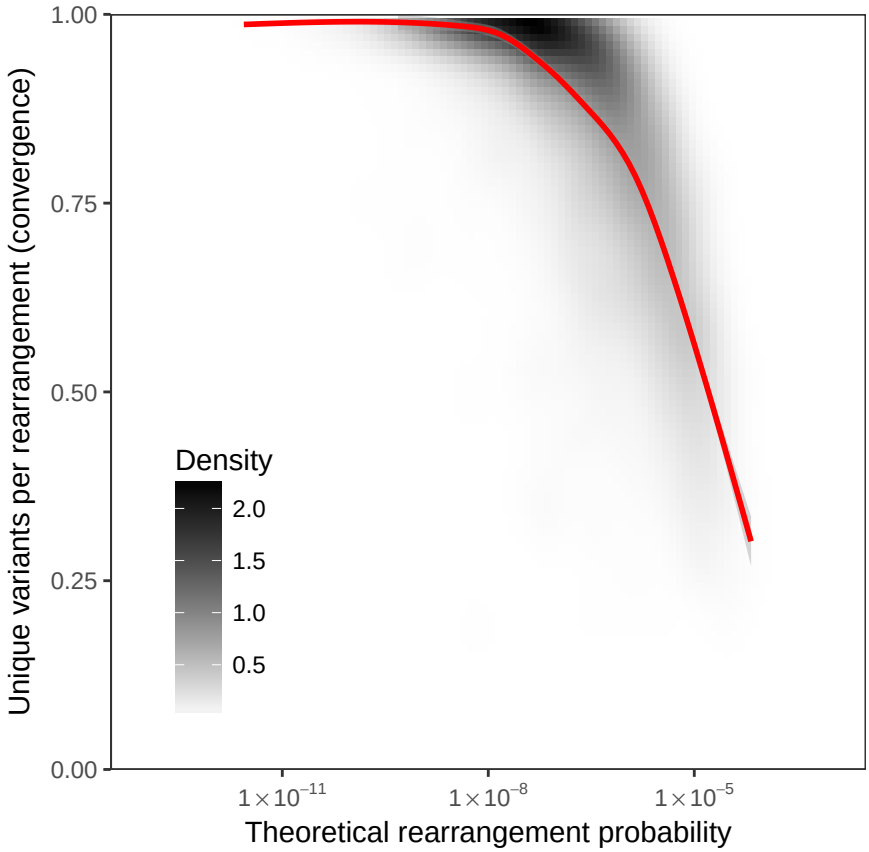
```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

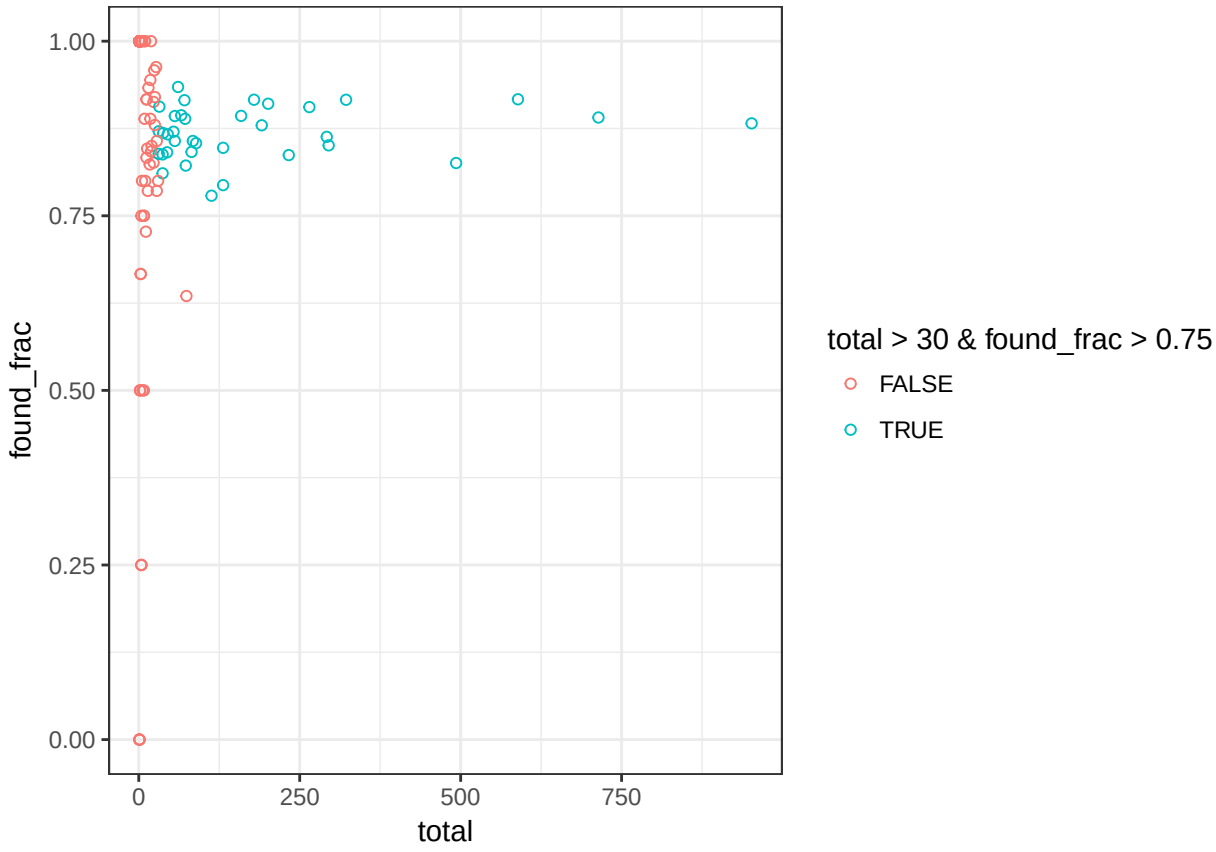
```
## Warning: Removed 22 rows containing non-finite values (stat_density2d).
```

```
## `geom_smooth()` using method = 'gam'
```

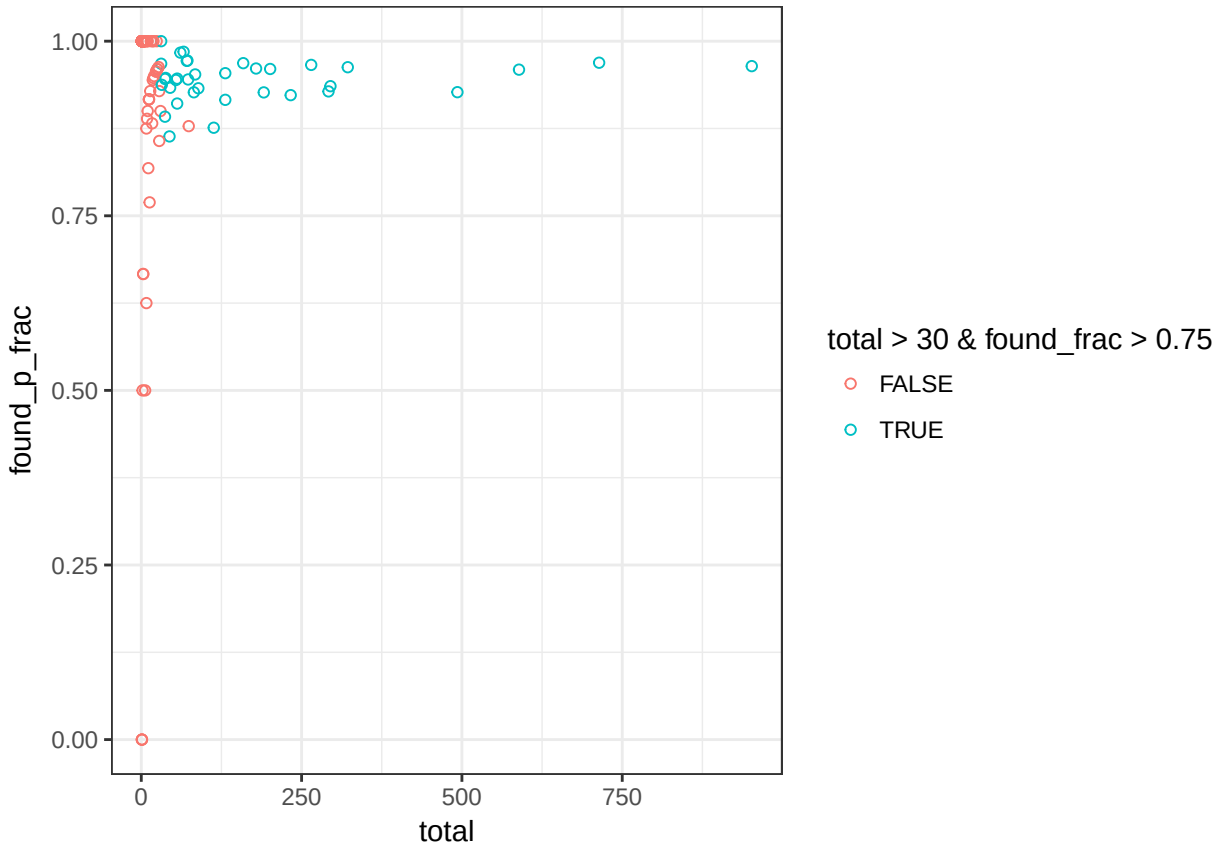
```
## Warning: Removed 22 rows containing non-finite values (stat_smooth).
```




```
dt.epicount3 = df.hip %>%
  group_by(antigen.epitope) %>%
  summarise(total = n(),
            found_frac = sum(count > 0) / total,
            found_p_frac = sum(genP_1mism_rob > 0) / total)

ggplot(dt.epicount3, aes(x = total, y = found_frac)) +
  geom_point(shape = 21, aes(color = total > 30 & found_frac > 0.75)) +
  theme_bw()
```




```
ggplot(dt.epicount3, aes(x = total, y = found_p_frac)) +
  geom_point(shape = 21, aes(color = total > 30 & found_frac > 0.75)) +
  theme_bw()
```

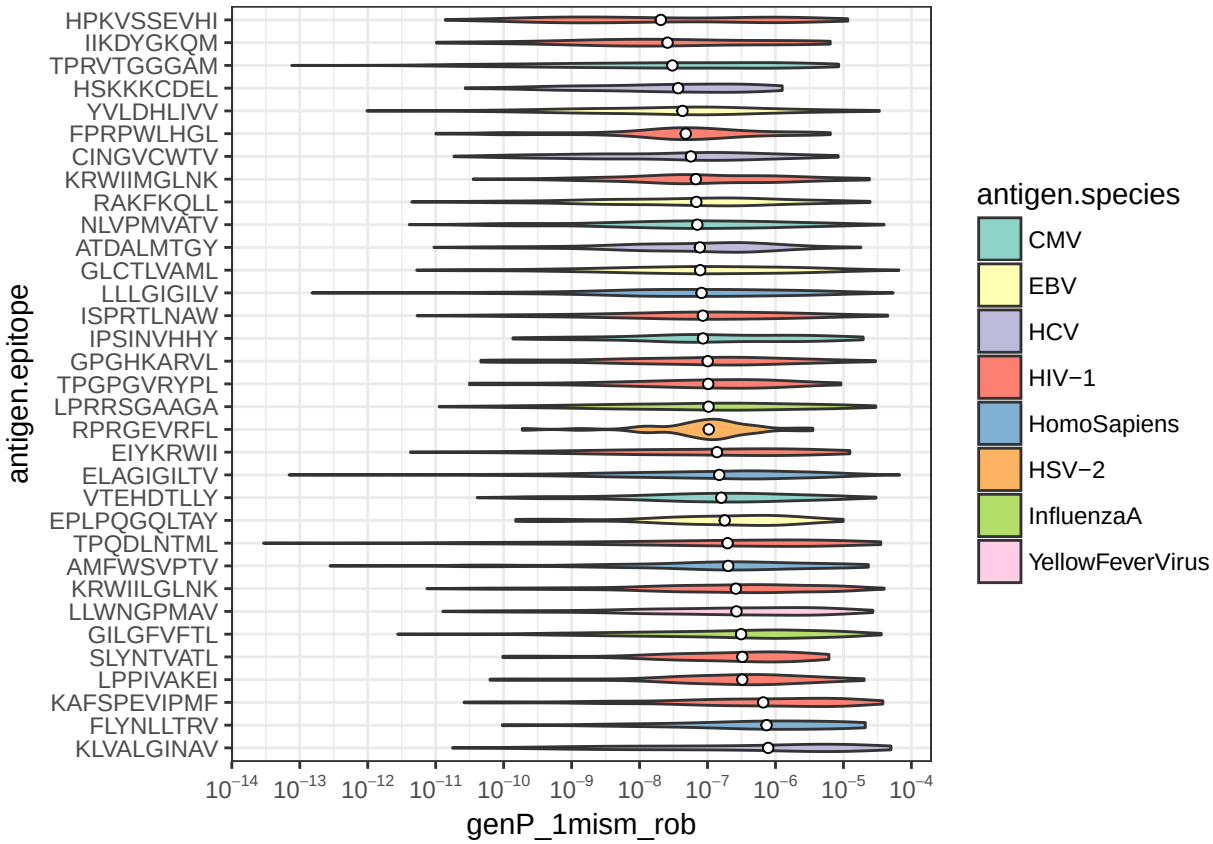



```
good_epi = (dt.epicount3 %>% filter(total > 30 & found_frac > 0.75))$antigen.epitope

df.hip.all = df.hip
df.hip = df.hip %>% filter(antigen.epitope %in% good_epi, mhc.class == "MHCI")
```

Exploratory analysis

By epitope

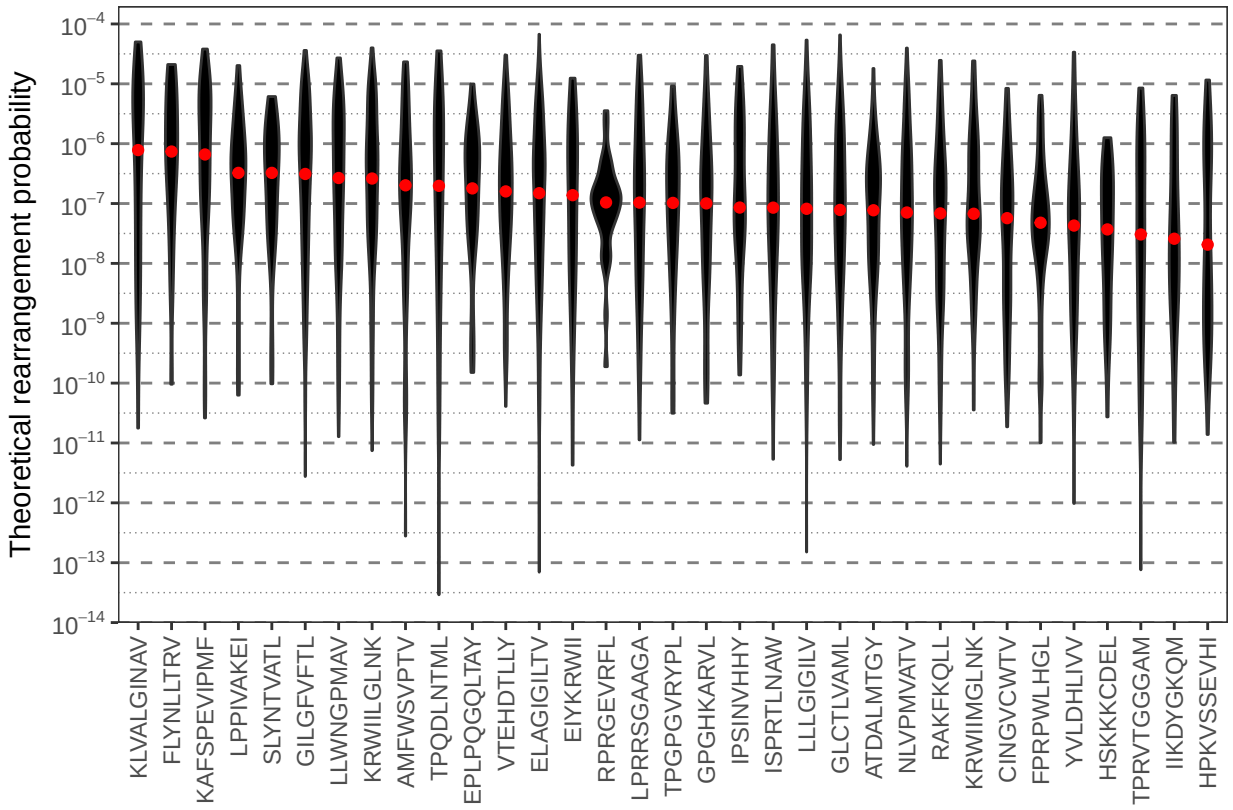
```
df.hip.gp = df.hip %>%
  filter(genP_1mism_rob > 0)
df.hip.gp.s = df.hip.gp %>%
  group_by(antigen.epitope) %>%
  summarise(genP_med = median(genP_1mism_rob)) %>%
  arrange(-genP_med)
df.hip.gp$antigen.epitope = factor(df.hip.gp$antigen.epitope,
                                   levels = df.hip.gp.s$antigen.epitope)

ggplot(df.hip.gp, aes(x=antigen.epitope, group = antigen.epitope, y=genP_1mism_rob, fill = antigen.spec
  geom_violin() + stat_summary(fun.y=median, geom="point", shape=21, fill = "white", color="black") +
  scale_y_log10(breaks = 10^(-15:-1),
                label= function(x) {ifelse(x==0, "0", parse(text=gsub("[+]", "", gsub(".+e", "10^", sci
  coord_flip() +
  scale_fill_brewer(palette = "Set3") +
  theme_bw()
```



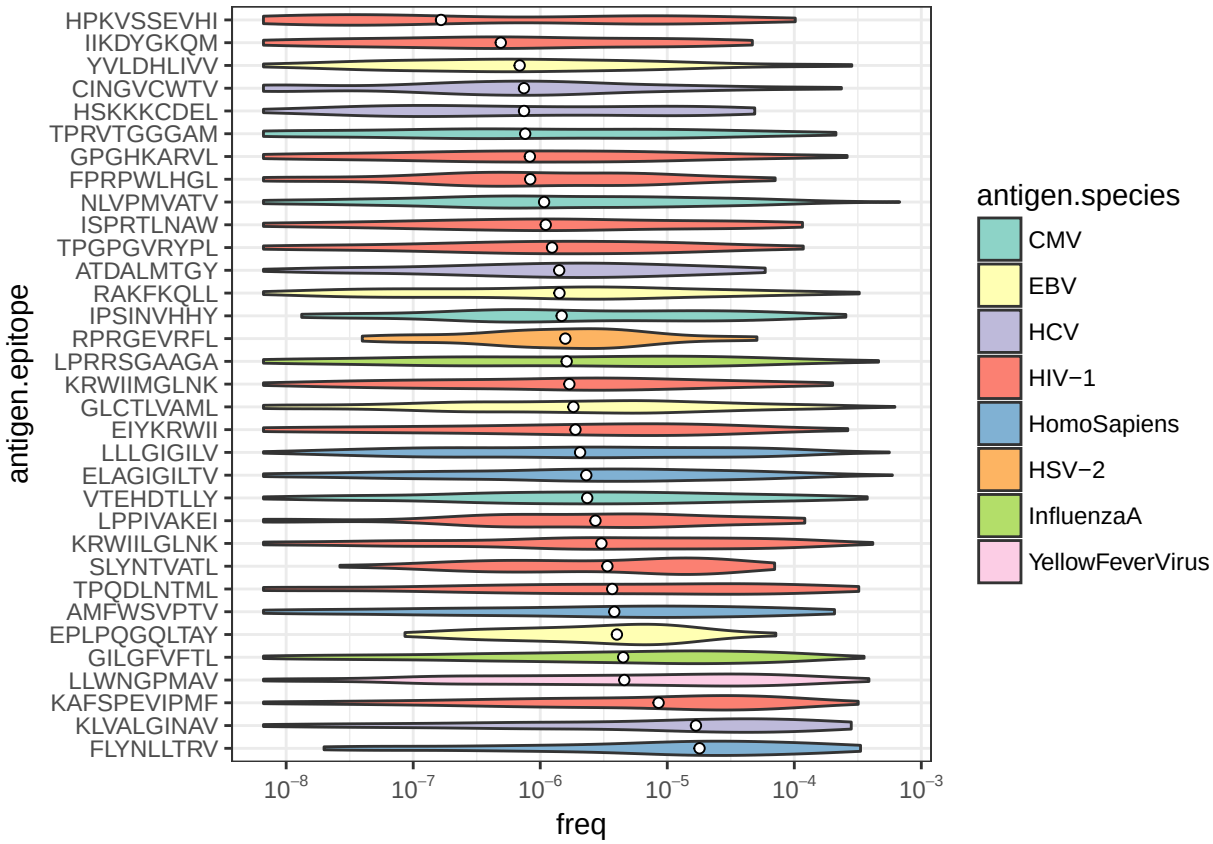

```
a = kruskal.test(freq ~ antigen.epitope, data = df.hip.gp)
print(a)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  freq by antigen.epitope
## Kruskal-Wallis chi-squared = 195.85, df = 32, p-value < 2.2e-16
```

```
p3=ggplot(df.hip.gp, aes(x=antigen.epitope, group = antigen.epitope, y=genP_1mism_rob)) +
  geom_violin(fill = "black") +
  stat_summary(fun.y=median, geom="point", color="red") +
  scale_y_log10("Theoretical rearrangement probability",
                breaks = 10^(-15:-1),
                label= function(x) {ifelse(x==0, "0", parse(text=gsub("[+]", "", gsub(".+e", "10^", sci
  xlab("") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
        panel.grid.major.y = element_line(colour="grey50", linetype="dashed"),
        panel.grid.minor.y = element_line(colour="grey50", linetype="dotted"),
        panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
p3
```



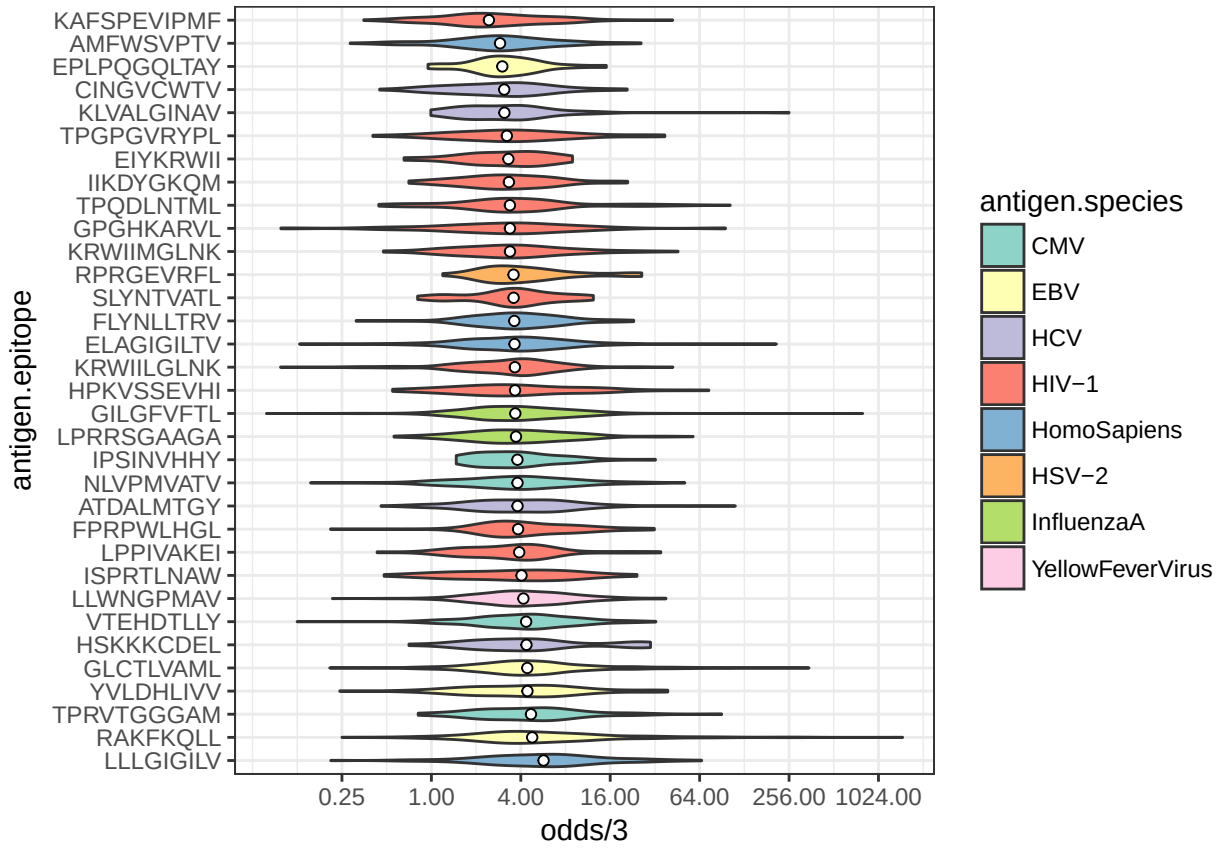

```
df.hip.obs = df.hip %>%
  filter(freq > 0)
df.hip.obs.s = df.hip.obs %>%
  group_by(antigen.epitope) %>%
  summarise(freq_med = median(freq)) %>%
  arrange(-freq_med)
df.hip.obs$antigen.epitope = factor(df.hip.obs$antigen.epitope,
                                    levels = df.hip.obs.s$antigen.epitope)

ggplot(df.hip.obs, aes(x=antigen.epitope, group = antigen.epitope, y=freq, fill = antigen.species)) +
  geom_violin() + stat_summary(fun.y=median, geom="point", shape=21, fill = "white", color="black") +
  scale_y_log10(breaks = 10^(-15:-1),
                label= function(x) {ifelse(x==0, "0", parse(text=gsub("[+]", "", gsub(".+e", "10^", sci
  coord_flip() +
  scale_fill_brewer(palette = "Set3") +
  theme_bw()
```




```
a = kruskal.test(freq ~ antigen.epitope, data = df.hip.obs)
print(a)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  freq by antigen.epitope
## Kruskal-Wallis chi-squared = 187.05, df = 32, p-value < 2.2e-16
```

```
df.hip.odds = df.hip %>%
  filter(freq > 0, genP_1mism_rob > 0) %>%
  mutate(odds = freq/genP_1mism_rob,
         log.odds = log(odds))
df.hip.odds.s = df.hip.odds %>%
  group_by(antigen.epitope) %>%
  summarise(odds_med = median(odds)) %>%
  arrange(-odds_med)
df.hip.odds$antigen.epitope = factor(df.hip.odds$antigen.epitope,
                                     levels = df.hip.odds.s$antigen.epitope)

ggplot(df.hip.odds, aes(x=antigen.epitope, group = antigen.epitope, y=odds/3, fill = antigen.species)) +
  geom_violin() + stat_summary(fun.y=median, geom="point", shape=21, fill = "white", color="black") +
  scale_y_log10(breaks = 2^(seq(-2,20,by=2))) +
  coord_flip() +
  scale_fill_brewer(palette = "Set3") +
  theme_bw()
```

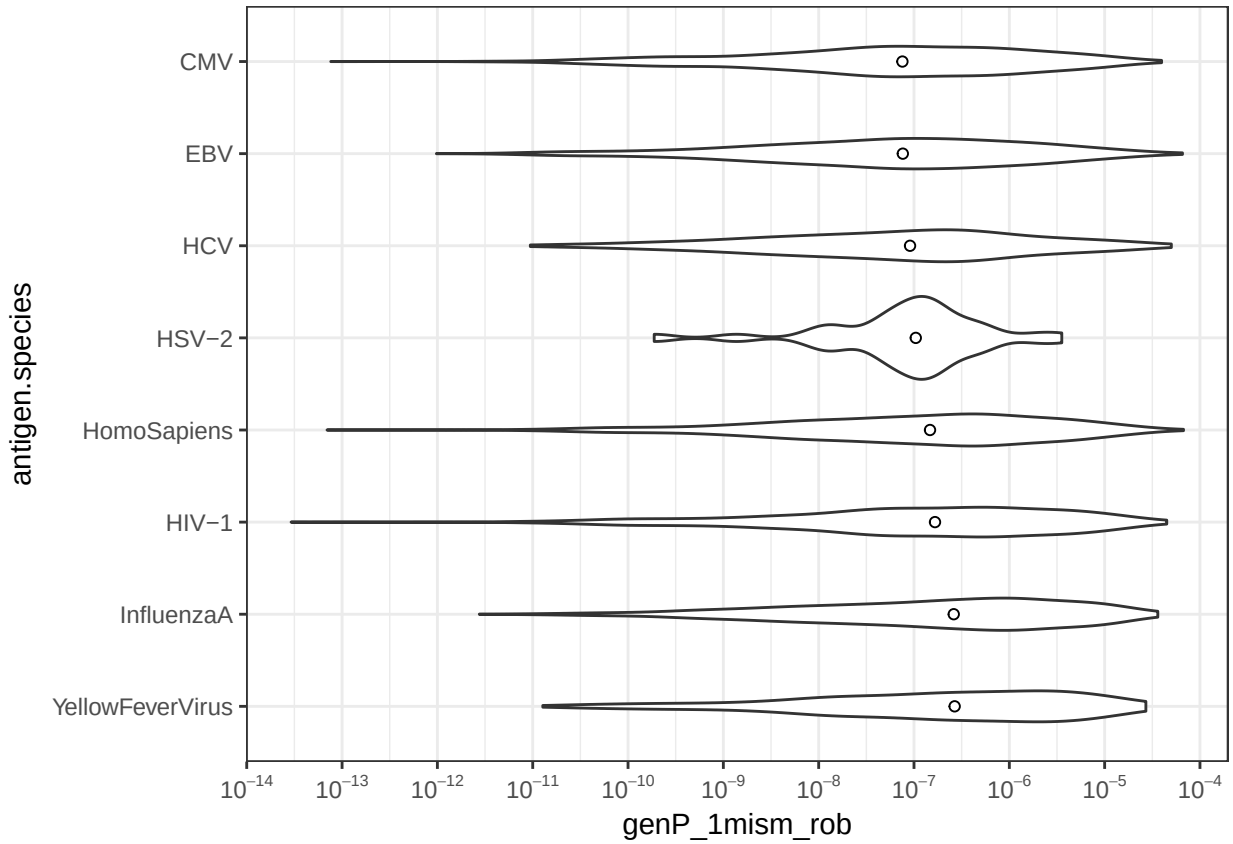



```
a = kruskal.test(odds ~ antigen.epitope, data = df.hip.odds)
print(a)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  odds by antigen.epitope
## Kruskal-Wallis chi-squared = 176.72, df = 32, p-value < 2.2e-16
```

By species

```
df.hip.gp = df.hip %>%
  filter(genP_1mism_rob > 0)
df.hip.gp.s = df.hip.gp %>%
  group_by(antigen.species) %>%
  summarise(genP_med = median(genP_1mism_rob)) %>%
  arrange(-genP_med)
df.hip.gp$antigen.species = factor(df.hip.gp$antigen.species,
                                   levels = df.hip.gp.s$antigen.species)

ggplot(df.hip.gp, aes(x=antigen.species, group = antigen.species, y=genP_1mism_rob)) +
  geom_violin() + stat_summary(fun.y=median, geom="point", shape=21, fill = "white", color="black") +
  scale_y_log10(breaks = 10^(-15:-1),
                label= function(x) {ifelse(x==0, "0", parse(text=gsub("[+]", "", gsub(".+e", "10^", sci
  coord_flip() +
  theme_bw()
```




```
kk = kruskal.test(freq ~ antigen.species, data = df.hip.gp)
print(kk)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  freq by antigen.species
## Kruskal-Wallis chi-squared = 61.42, df = 7, p-value = 7.854e-11
```

```
aa = aov(log(genP_1mism_rob) ~ antigen.species, data = df.hip.gp)
summary(aa)
```

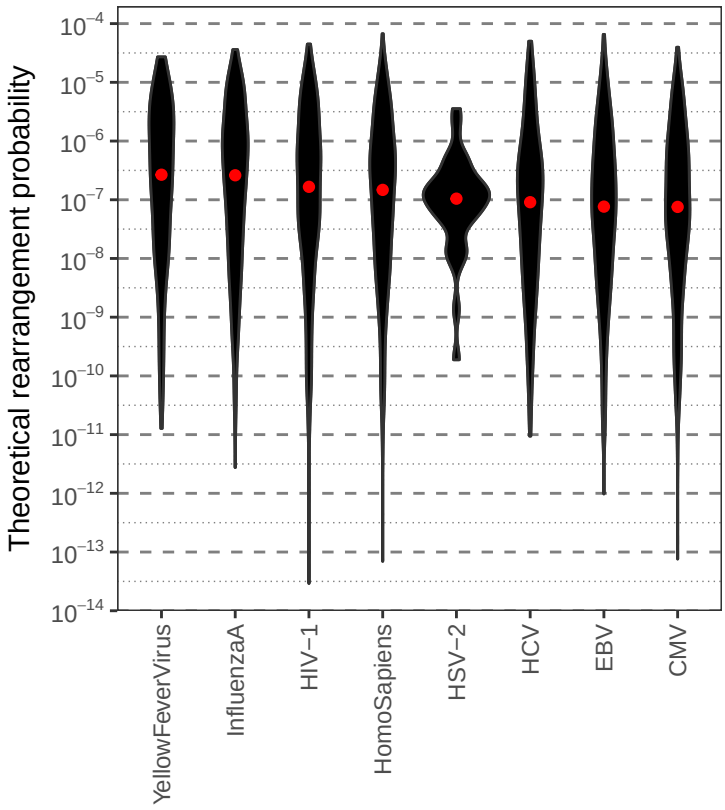
```
##                   Df Sum Sq Mean Sq F value   Pr(>F)
## antigen.species    7    725  103.57    9.57 7.13e-12 ***
## Residuals       5408  58524   10.82
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(aa, "antigen.species")
```

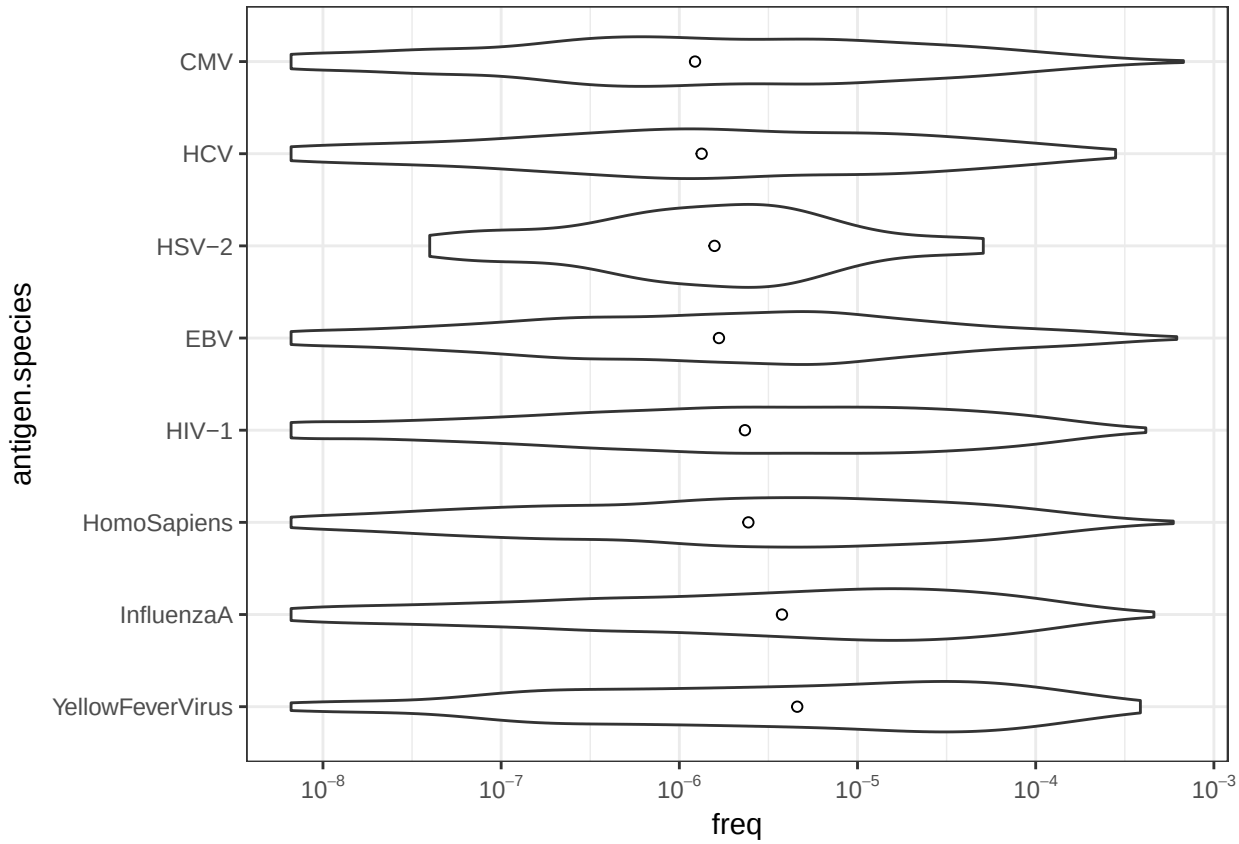
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = log(genP_1mism_rob) ~ antigen.species, data = df.hip.gp)
##
## $antigen.species
##                                    diff        lwr         upr     p adj
## InfluenzaA-YellowFeverVirus  -0.091651700 -0.8176027  0.63429927 0.9999442
## HIV-1-YellowFeverVirus       -0.426998123 -1.1218053  0.26780905 0.5763570
```

```
## HomoSapiens-YellowFeverVirus -0.583615565 -1.2668330  0.09960191 0.1596112
## HSV-2-YellowFeverVirus       -0.705184118 -2.6300116  1.21964338 0.9546710
## HCV-YellowFeverVirus         -0.964323630 -1.8204123 -0.10823495 0.0148207
## EBV-YellowFeverVirus         -1.067506582 -1.7682872 -0.36672595 0.0001070
## CMV-YellowFeverVirus         -1.058699289 -1.7727116 -0.34468698 0.0001904
## HIV-1-InfluenzaA             -0.335346423 -0.8175262  0.14683331 0.4092257
## HomoSapiens-InfluenzaA       -0.491963865 -0.9572878 -0.02663989 0.0295015
## HSV-2-InfluenzaA             -0.613532418 -2.4722151  1.24515023 0.9744335
## HCV-InfluenzaA               -0.872671930 -1.5673863 -0.17795752 0.0035334
## EBV-InfluenzaA               -0.975854882 -1.4666031 -0.48510670 0.0000000
## CMV-InfluenzaA               -0.967047589 -1.4765119 -0.45758330 0.0000003
## HomoSapiens-HIV-1            -0.156617442 -0.5716820  0.25844714 0.9470009
## HSV-2-HIV-1                  -0.278185996 -2.1249273  1.56855529 0.9998163
## HCV-HIV-1                    -0.537325507 -1.1994285  0.12477744 0.2127480
## EBV-HIV-1                    -0.640508459 -1.0838886 -0.19712828 0.0003226
## CMV-HIV-1                    -0.631701166 -1.0957120 -0.16769032 0.0009699
## HSV-2-HomoSapiens            -0.121568553 -1.9639807  1.72084358 0.9999994
## HCV-HomoSapiens              -0.380708065 -1.0306384  0.26922226 0.6364588
## EBV-HomoSapiens              -0.483891017 -0.9088794 -0.05890268 0.0130571
## CMV-HomoSapiens              -0.475083724 -0.9215534 -0.02861405 0.0276280
## HCV-HSV-2                    -0.259139512 -2.1724048  1.65412581 0.9999101
## EBV-HSV-2                    -0.362322464 -2.2113195  1.48667452 0.9989556
## CMV-HSV-2                    -0.353515171 -2.2075675  1.50053713 0.9991258
## EBV-HCV                      -0.103182952 -0.7715517  0.56518581 0.9997834
## CMV-HCV                      -0.094375659 -0.7766050  0.58785369 0.9998964
## CMV-EBV                       0.008807293 -0.4641013  0.48171590 1.0000000
```

```
p4=ggplot(df.hip.gp, aes(x=antigen.species, y=genP_1mism_rob)) +
  geom_violin(aes(group = antigen.species)) +
  geom_violin(fill = "black") +
  stat_summary(fun.y=median, geom="point", color="red") +
  scale_y_log10("Theoretical rearrangement probability",
                breaks = 10^(-15:-1),
                label= function(x) {ifelse(x==0, "0", parse(text=gsub("[+]", "", gsub(".+e", "10^", sci
  xlab("") +
  theme_bw() +
  theme(aspect.ratio = 1, axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
        panel.grid.major.y = element_line(colour="grey50", linetype="dashed"),
        panel.grid.minor.y = element_line(colour="grey50", linetype="dotted"),
        panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
p4
```




```
df.hip.obs = df.hip %>%
  filter(freq > 0)
df.hip.obs.s = df.hip.obs %>%
  group_by(antigen.species) %>%
  summarise(freq_med = median(freq)) %>%
  arrange(-freq_med)
df.hip.obs$antigen.species = factor(df.hip.obs$antigen.species,
                                    levels = df.hip.obs.s$antigen.species)

ggplot(df.hip.obs, aes(x=antigen.species, group = antigen.species, y=freq)) +
  geom_violin() + stat_summary(fun.y=median, geom="point", shape=21, fill = "white", color="black") +
  scale_y_log10(breaks = 10^(-15:-1),
                label= function(x) {ifelse(x==0, "0", parse(text=gsub("[+]", "", gsub(".+e", "10^", sci
  coord_flip() +
  theme_bw()
```




```
kk = kruskal.test(freq ~ antigen.species, data = df.hip.obs)
print(kk)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  freq by antigen.species
## Kruskal-Wallis chi-squared = 49.856, df = 7, p-value = 1.542e-08
```

```
aa = aov(log(freq) ~ antigen.species, data = df.hip.obs)
summary(aa)
```

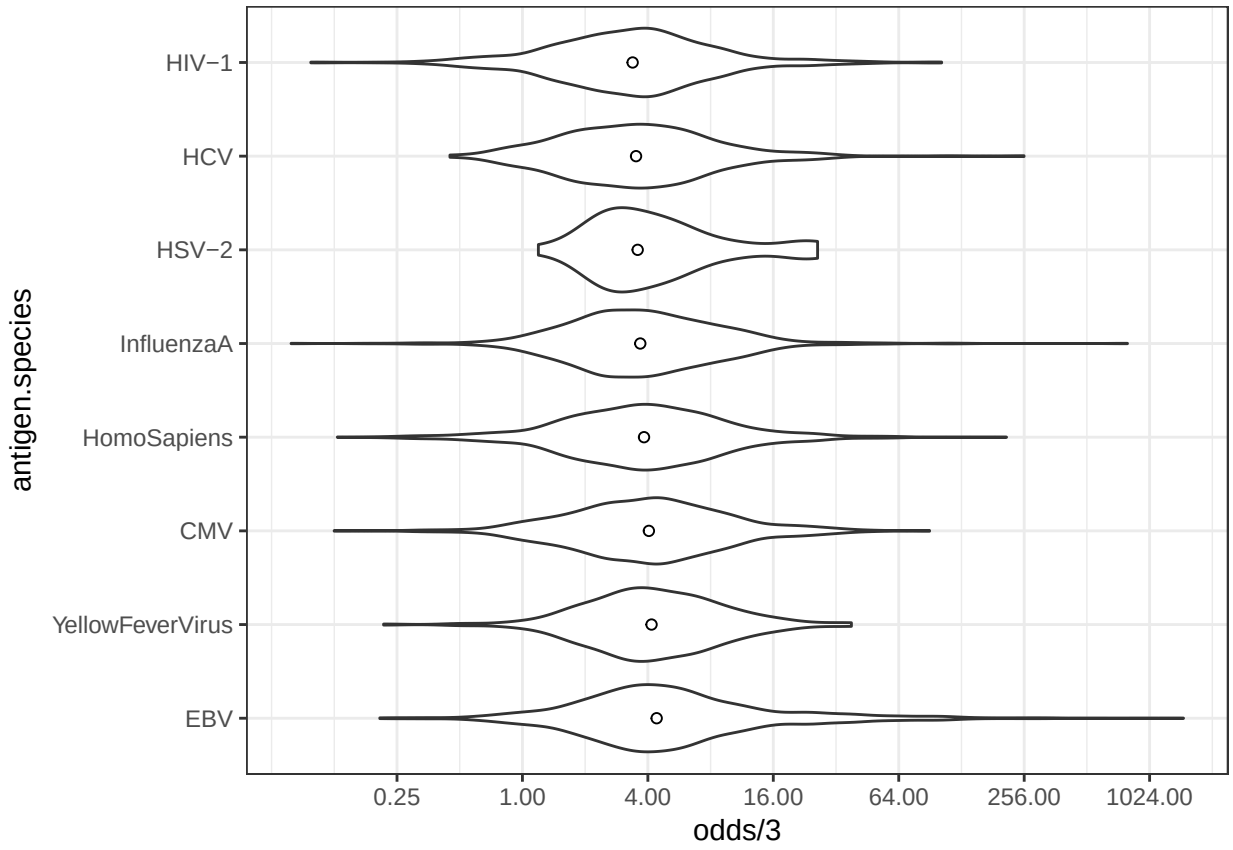
```
##                   Df Sum Sq Mean Sq F value   Pr(>F)
## antigen.species    7    326   46.59   6.379 1.73e-07 ***
## Residuals       4981  36381    7.30
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(aa, "antigen.species")
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = log(freq) ~ antigen.species, data = df.hip.obs)
##
## $antigen.species
##                                     diff        lwr          upr     p adj
## InfluenzaA-YellowFeverVirus  -0.34383690 -0.9588698  0.271195972 0.6905298
## HomoSapiens-YellowFeverVirus -0.58861986 -1.1695610 -0.007678736 0.0444203
```

```
## HIV-1-YellowFeverVirus       -0.63756363 -1.2277537 -0.047373508 0.0236070
## EBV-YellowFeverVirus         -0.88280379 -1.4784114 -0.287196210 0.0001921
## HSV-2-YellowFeverVirus       -0.96089289 -2.5719176  0.650131802 0.6142660
## HCV-YellowFeverVirus         -0.95055158 -1.6868365 -0.214266648 0.0023328
## CMV-YellowFeverVirus         -0.98568497 -1.5942106 -0.377159376 0.0000256
## HomoSapiens-InfluenzaA       -0.24478296 -0.6399474  0.150381474 0.5660248
## HIV-1-InfluenzaA             -0.29372672 -0.7023668  0.114913356 0.3643354
## EBV-InfluenzaA               -0.53896689 -0.9553930 -0.122540750 0.0022425
## HSV-2-InfluenzaA             -0.61705599 -2.1707815  0.936669501 0.9309366
## HCV-InfluenzaA               -0.60671468 -1.2073625 -0.006066828 0.0456860
## CMV-InfluenzaA               -0.64184807 -1.0765500 -0.207146174 0.0002079
## HIV-1-HomoSapiens            -0.04894377 -0.4042097  0.306322146 0.9998992
## EBV-HomoSapiens              -0.29418393 -0.6583788  0.070010911 0.2179532
## HSV-2-HomoSapiens            -0.37227303 -1.9128217  1.168275590 0.9960223
## HCV-HomoSapiens              -0.36193172 -0.9276216  0.203758118 0.5231726
## CMV-HomoSapiens              -0.39706511 -0.7820234 -0.012106813 0.0376016
## EBV-HIV-1                    -0.24524017 -0.6240141  0.133533739 0.5071380
## HSV-2-HIV-1                  -0.32332926 -1.8673894  1.220730916 0.9983957
## HCV-HIV-1                    -0.31298795 -0.8881721  0.262196173 0.7195952
## CMV-HIV-1                    -0.34812134 -0.7469003  0.050657640 0.1395864
## HSV-2-EBV                    -0.07808910 -1.6242281  1.468049916 0.9999999
## HCV-EBV                      -0.06774779 -0.6484894  0.512993803 0.9999674
## CMV-EBV                      -0.10288118 -0.5096350  0.303872670 0.9947193
## HCV-HSV-2                     0.01034131 -1.5952467  1.615929355 1.0000000
## CMV-HSV-2                    -0.02479208 -1.5759532  1.526369056 1.0000000
## CMV-HCV                      -0.03513339 -0.6291164  0.558849606 0.9999997
```

```
df.hip.odds = df.hip %>%
  filter(freq > 0, genP_1mism_rob > 0) %>%
  mutate(odds = freq/genP_1mism_rob,
         log.odds = log(odds))
df.hip.odds.s = df.hip.odds %>%
  group_by(antigen.species) %>%
  summarise(odds_med = median(odds)) %>%
  arrange(-odds_med)
df.hip.odds$antigen.species = factor(df.hip.odds$antigen.species,
                                     levels = df.hip.odds.s$antigen.species)

ggplot(df.hip.odds, aes(x=antigen.species, group = antigen.species, y=odds/3)) +
  geom_violin() + stat_summary(fun.y=median, geom="point", shape=21, fill = "white", color="black") +
  scale_y_log10(breaks = 2^(seq(-2,20,by=2))) +
  coord_flip() +
  theme_bw()
```




```
kk = kruskal.test(odds ~ antigen.species, data = df.hip.odds)
print(kk)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  odds by antigen.species
## Kruskal-Wallis chi-squared = 83.127, df = 7, p-value = 3.168e-15
```

```
aa = aov(log(odds) ~ antigen.species, data = df.hip.odds)
summary(aa)
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## antigen.species    7     92  13.210   15.62 <2e-16 ***
## Residuals       4963   4198   0.846
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(aa, "antigen.species")
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = log(odds) ~ antigen.species, data = df.hip.odds)
##
## $antigen.species
##                                   diff        lwr          upr
## YellowFeverVirus-EBV        -0.159361915 -0.3620938  0.043370002
## CMV-EBV                     -0.235908342 -0.3746524 -0.097164263
```

```
## HomoSapiens-EBV              -0.284382768 -0.4084422 -0.160323358
## InfluenzaA-EBV               -0.288859385 -0.4306959 -0.147022905
## HSV-2-EBV                    -0.134144645 -0.6603110  0.392021708
## HCV-EBV                      -0.351246328 -0.5495247 -0.152967973
## HIV-1-EBV                    -0.422802617 -0.5519633 -0.293641980
## CMV-YellowFeverVirus         -0.076546427 -0.2837980  0.130705175
## HomoSapiens-YellowFeverVirus -0.125020853 -0.3227430  0.072701287
## InfluenzaA-YellowFeverVirus  -0.129497470 -0.3388319  0.079836938
## HSV-2-YellowFeverVirus        0.025217270 -0.5230111  0.573445633
## HCV-YellowFeverVirus         -0.191884414 -0.4429175  0.059148679
## HIV-1-YellowFeverVirus       -0.263440702 -0.4644028 -0.062478573
## HomoSapiens-CMV              -0.048474426 -0.1797898  0.082840972
## InfluenzaA-CMV               -0.052951043 -0.2011758  0.095273731
## HSV-2-CMV                     0.101763697 -0.4261606  0.629687960
## HCV-CMV                      -0.115337986 -0.3182353  0.087559284
## HIV-1-CMV                    -0.186894275 -0.3230393 -0.050749254
## InfluenzaA-HomoSapiens       -0.004476617 -0.1390552  0.130101990
## HSV-2-HomoSapiens             0.150238123 -0.3740183  0.674494592
## HCV-HomoSapiens              -0.066863560 -0.2600167  0.126289541
## HIV-1-HomoSapiens            -0.138419849 -0.2595656 -0.017274133
## HSV-2-InfluenzaA              0.154714740 -0.3740307  0.683460137
## HCV-InfluenzaA               -0.062386943 -0.2674113  0.142637373
## HIV-1-InfluenzaA             -0.133943232 -0.2732384  0.005351902
## HCV-HSV-2                    -0.217101683 -0.7636988  0.329495439
## HIV-1-HSV-2                  -0.288657972 -0.8141450  0.236829019
## HIV-1-HCV                    -0.071556289 -0.2680247  0.124912165
##                                  p adj
## YellowFeverVirus-EBV         0.2495678
## CMV-EBV                      0.0000073
## HomoSapiens-EBV              0.0000000
## InfluenzaA-EBV               0.0000000
## HSV-2-EBV                    0.9944544
## HCV-EBV                      0.0000023
## HIV-1-EBV                    0.0000000
## CMV-YellowFeverVirus         0.9526597
## HomoSapiens-YellowFeverVirus 0.5389385
## InfluenzaA-YellowFeverVirus  0.5677719
## HSV-2-YellowFeverVirus       0.9999999
## HCV-YellowFeverVirus         0.2839124
## HIV-1-YellowFeverVirus       0.0018301
## HomoSapiens-CMV              0.9527929
## InfluenzaA-CMV               0.9604156
## HSV-2-CMV                    0.9990611
## HCV-CMV                      0.6717423
## HIV-1-CMV                    0.0008384
## InfluenzaA-HomoSapiens       1.0000000
## HSV-2-HomoSapiens            0.9887347
## HCV-HomoSapiens              0.9666678
## HIV-1-HomoSapiens            0.0125232
## HSV-2-InfluenzaA             0.9872529
## HCV-InfluenzaA               0.9839603
## HIV-1-InfluenzaA             0.0696794
## HCV-HSV-2                    0.9308990
## HIV-1-HSV-2                  0.7096322
```

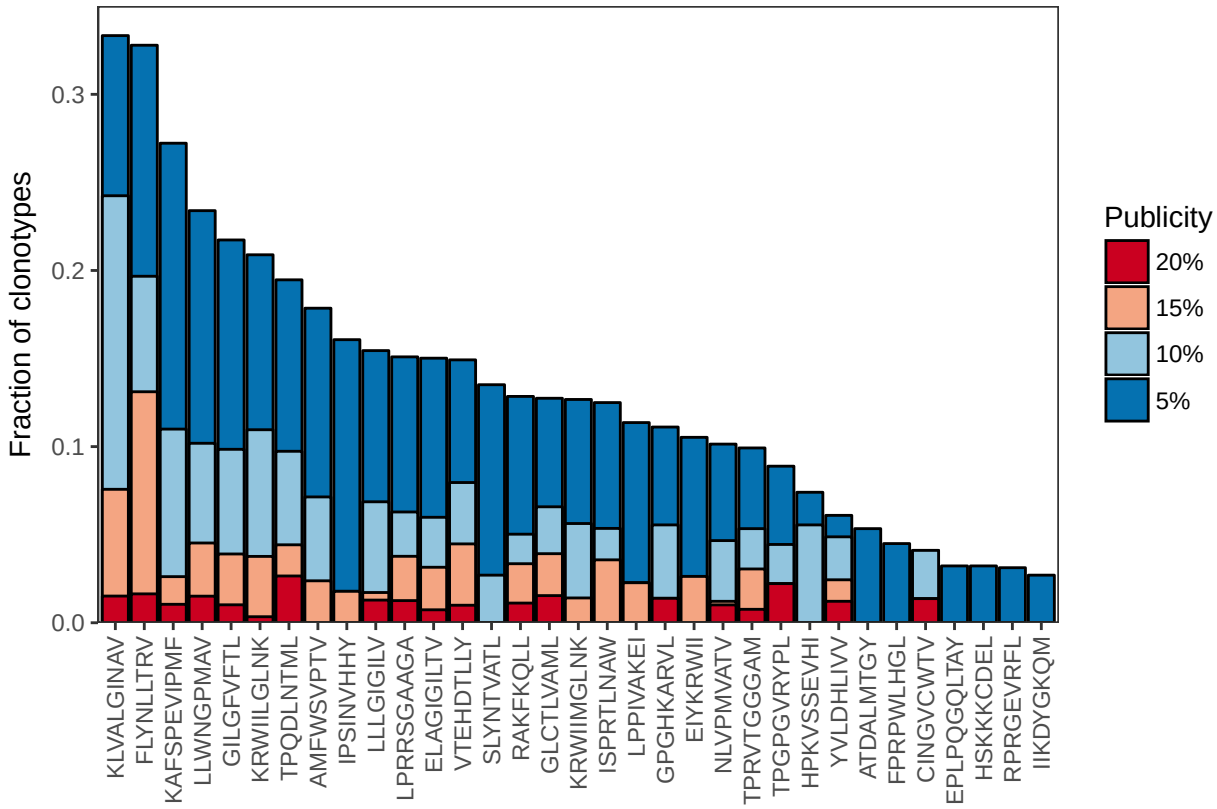
```
## HIV-1-HCV                    0.9560655
```

Number of publics

```
df.hip.frac = df.hip %>%
  group_by(antigen.epitope) %>%
  summarise(publ_5 = sum(incidence > 0.05) / n(),
            publ_10 = sum(incidence > 0.1) / n(),
            publ_15 = sum(incidence > 0.15) / n(),
            publ_20 = sum(incidence > 0.20) / n())

df.hip.frac$antigen.epitope = with(df.hip.frac,
                                   factor(antigen.epitope, levels = antigen.epitope[order(-publ_5)]))

df.hip.frac = df.hip.frac %>%
  melt
```

```
## Using antigen.epitope as id variables
```

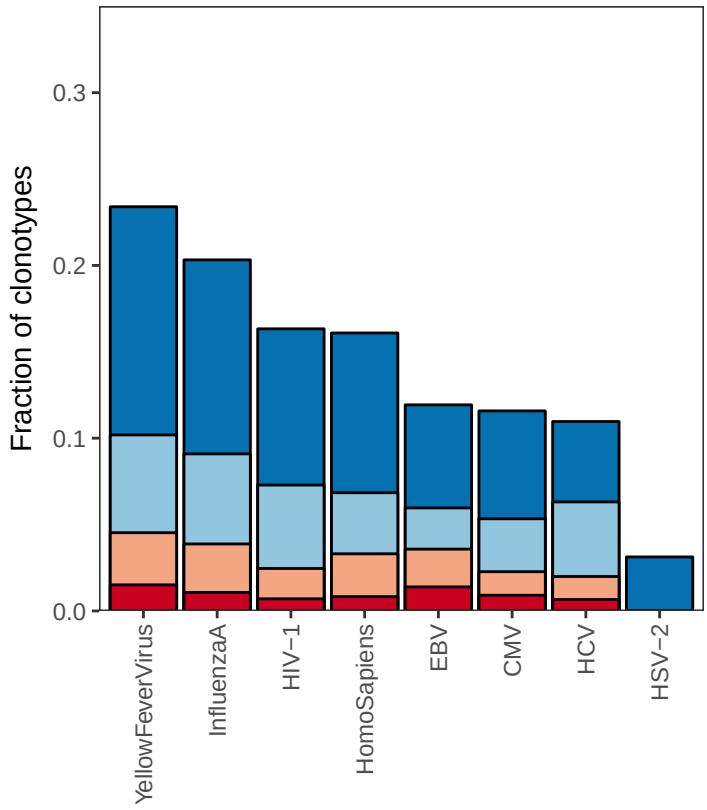
```
df.hip.frac$publ = paste0(str_split_fixed(df.hip.frac$variable, "_", n = 2)[,2],"%")
tmp = unique(df.hip.frac$publ)
df.hip.frac$publ = factor(df.hip.frac$publ, levels = tmp[order(sapply(tmp, function(x) -as.numeric(gsub

p5=ggplot(df.hip.frac,
       aes(x=antigen.epitope, y = value, fill = publ)) +
  geom_bar(stat="identity", color = "black", position = "identity") +
  scale_fill_brewer("Publicity", palette = "RdBu") +
  scale_x_discrete("") +
  scale_y_continuous("Fraction of clonotypes", expand = c(0,0), limits = c(0,0.35)) +
  theme_bw() +
    theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
p5
```




```
df.hip.frac = df.hip %>%
  group_by(antigen.species) %>%
  summarise(publ_5 = sum(incidence > 0.05) / n(),
            publ_10 = sum(incidence > 0.1) / n(),
            publ_15 = sum(incidence > 0.15) / n(),
            publ_20 = sum(incidence > 0.20) / n())

df.hip.frac$antigen.species = with(df.hip.frac,
                                   factor(antigen.species, levels = antigen.species[order(-publ_5)]))

df.hip.frac = df.hip.frac %>%
  melt
```

```
## Using antigen.species as id variables
```

```
df.hip.frac$publ = paste0(str_split_fixed(df.hip.frac$variable, "_", n = 2)[,2],"%")
tmp = unique(df.hip.frac$publ)
df.hip.frac$publ = factor(df.hip.frac$publ, levels = tmp[order(sapply(tmp, function(x) -as.numeric(gsub

p6=ggplot(df.hip.frac,
       aes(x=antigen.species, y = value, fill = publ)) +
  geom_bar(stat="identity", color = "black", position = "identity") +
  scale_fill_brewer("Publicity", palette = "RdBu", guide = F) +
  scale_x_discrete("") +
  scale_y_continuous("Fraction of clonotypes", expand = c(0,0), limits = c(0,0.35)) +
  theme_bw() +
    theme(aspect.ratio = 1, axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
p6
```



Epitope properties and rearrangement prob/selection

```
df.hip.all.1 = df.hip.all %>% filter(mhc.class == "MHCI", nchar(antigen.epitope) %in% c(8:11))
```

```
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     combine, src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

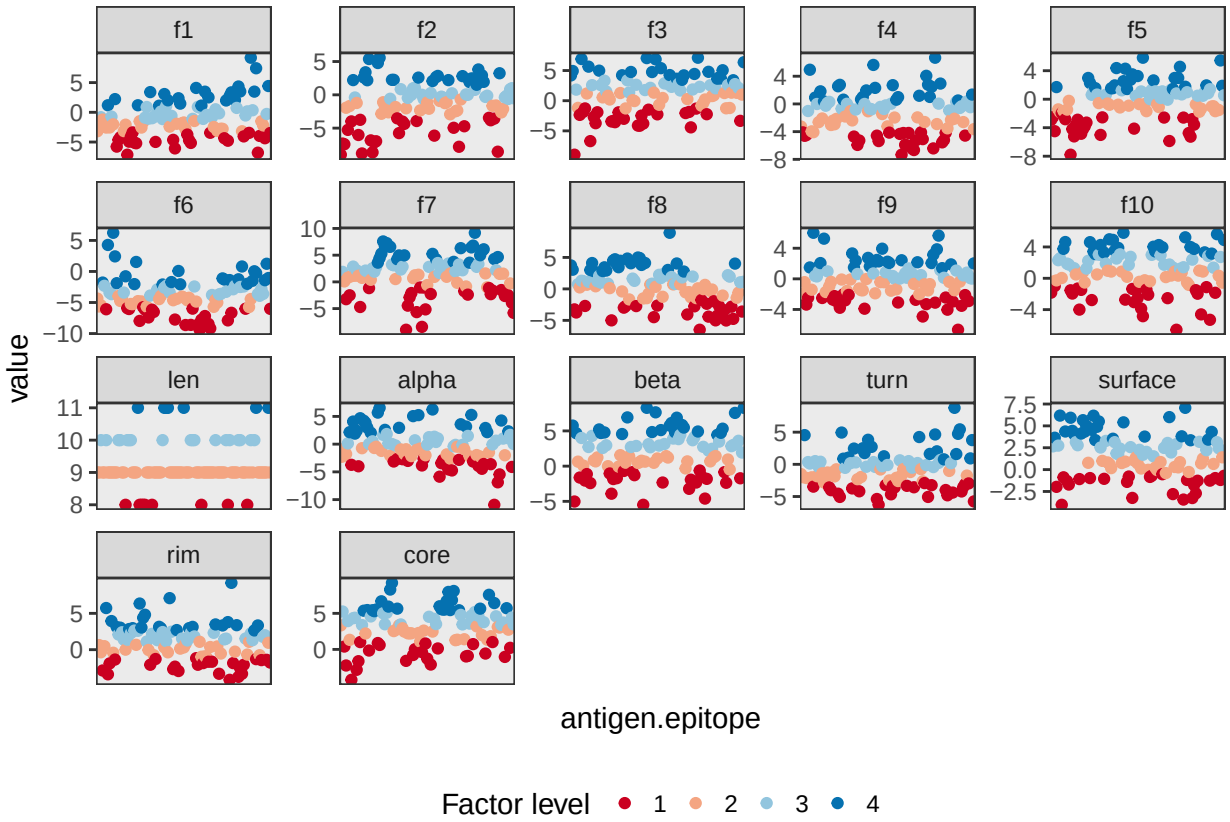
```
dt.epi.prop = rbindlist(lapply(strsplit(unique(df.hip.all.1$antigen.epitope), split = ""),
                               function(x) data.table(aa = x,
                                                      antigen.epitope = paste0(x, collapse = ""))))

dt.epi.prop = rbind(dt.epi.prop %>% merge(fread("kidera.txt") %>% mutate(len = 1) %>% melt, by = "aa", a
                    dt.epi.prop %>% merge(fread("martin-lavery.txt") %>% melt, by = "aa", allow.cartesia
  group_by(antigen.epitope, variable) %>%
```

```
  summarise(value = sum(value)) %>%
  group_by(variable) %>%
  mutate(value_c = as.factor(ifelse(variable == "len", as.integer(as.factor(value)), as.integer(cut2(va
```

```
## Using aa as id variables
```

```
## Warning in melt.data.table(.): To be consistent with reshape2's melt,
## id.vars and measure.vars are internally guessed when both are 'NULL'. All
## non-numeric/integer/logical type columns are conisdered id.vars, which
## in this case are columns [aa]. Consider providing at least one of 'id' or
## 'measure' vars in future.
```

```
ggplot(dt.epi.prop, aes(x = antigen.epitope, y = value, color = value_c)) +
  geom_point() +
  facet_wrap(~variable, scales = "free") +
  scale_color_brewer("Factor level", palette = "RdBu") +
  theme_bw() +
  theme(legend.position = "bottom",
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```



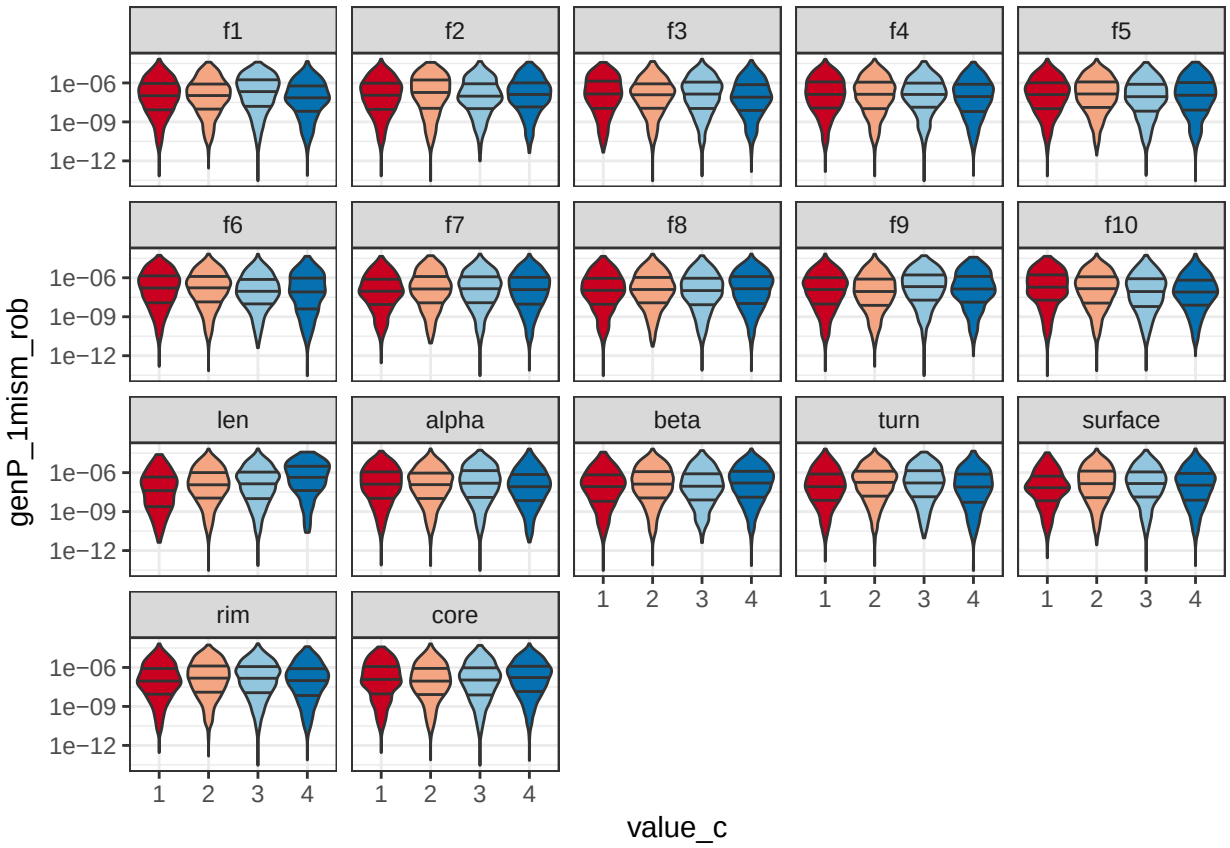

```
dt.hip.prop = dt.epi.prop %>% merge(df.hip.all.1, by = "antigen.epitope", allow.cartesian = T)
```

```
ggplot(dt.hip.prop, aes(x = value_c, y = genP_1mism_rob)) +
  geom_violin(aes(group = value_c, fill = value_c), draw_quantiles = c(0.25,0.5,0.75)) +
  #stat_summary(fun.y=mean, geom="point", shape=21, fill = "white", color="black") +
  scale_y_log10() +
  scale_fill_brewer(guide = F, palette = "RdBu") +
  facet_wrap(~variable) +
```

```
  theme_bw()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 5729 rows containing non-finite values (stat_ydensity).
```



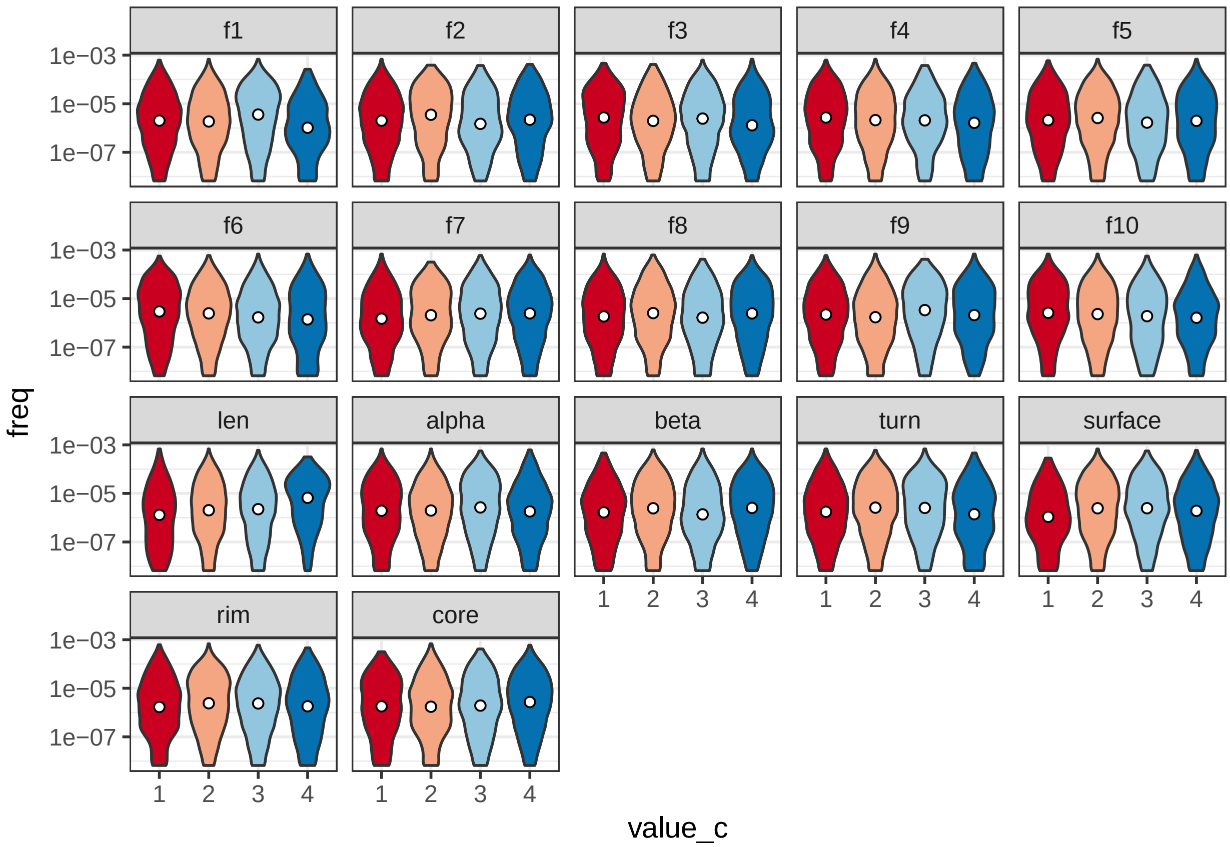

```
ggplot(dt.hip.prop, aes(x = value_c, group = value_c, y = freq, fill = value_c)) +
  geom_violin() +
  stat_summary(fun.y=median, geom="point", shape=21, fill = "white", color="black") +
  scale_y_log10() +
  scale_fill_brewer(guide = F, palette = "RdBu") +
  facet_wrap(~variable) +
  theme_bw()
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 13957 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 13957 rows containing non-finite values (stat_summary).
```




```
dt.hip.prop.cor = dt.hip.prop %>%
  group_by(variable) %>%
  summarise(r_freq = cor.test(freq, value, method = "spearman")$estimate,
            p_freq = cor.test(freq, value, method = "spearman")$p.value,
            r_P = cor.test(genP_1mism_rob, value, method = "spearman")$estimate,
            p_P = cor.test(genP_1mism_rob, value, method = "spearman")$p.value)
```

```
## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties
```

```
dt.hip.prop.aov1 = dt.hip.prop %>%
  filter(freq > 0) %>%
  group_by(variable) %>%
  summarise(F_freq = summary(aov(log(freq1)~value_c1, data.table(freq1 = freq, value_c1 = as.integer(va
            p_freq = summary(aov(log(freq1)~value_c1, data.table(freq1 = freq, value_c1 = as.integer(va
  mutate(p_freq_adj = p.adjust(p_freq))

dt.hip.prop.aov2 = dt.hip.prop %>%
  filter(genP_1mism_rob > 0) %>%
  group_by(variable) %>%
  summarise(F_P = summary(aov(log(P1)~value_c1, data.table(P1 = genP_1mism_rob, value_c1 = as.integer(v
            pa_P = summary(aov(log(P1)~value_c1, data.table(P1 = genP_1mism_rob, value_c1 = as.integer(
  mutate(pa_P_adj = p.adjust(pa_P))
```

```
dt.hip.prop.cor1 = dt.hip.prop %>%
  filter(freq > 0) %>%
  group_by(variable) %>%
  summarise(r_freq = cor.test(freq, value, method = "spearman")$estimate,
            p_freq = cor.test(freq, value, method = "spearman")$p.value) %>%
  mutate(p_freq_adj = p.adjust(p_freq))
```

```
## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(4.67816830819278e-05, 7.87971732023978e-07, :
## Cannot compute exact p-value with ties
```

```
dt.hip.prop.cor2 = dt.hip.prop %>%
  filter(genP_1mism_rob > 0) %>%
  group_by(variable) %>%
  summarise(r_P = cor.test(genP_1mism_rob, value, method = "spearman")$estimate,
            pr_P = cor.test(genP_1mism_rob, value, method = "spearman")$p.value) %>%
  mutate(pr_P_adj = p.adjust(pr_P))
```

```
## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties
```

```
## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties

## Warning in cor.test.default(c(1.40884137647308e-06, 1.00923389556377e-08, :
## Cannot compute exact p-value with ties
```

```r
dt.hip.prop.1 = merge(dt.hip.prop.aov2, dt.hip.prop.cor2)
print(dt.hip.prop.1)
```

```
##    variable          F_P         pa_P     pa_P_adj          r_P
## 1     alpha 2.209862e+00 1.371836e-01 8.231017e-01 -0.007805924
## 2      beta 1.372536e+01 2.134871e-04 2.775332e-03  0.040227142
## 3      core 1.269636e+01 3.692239e-04 4.430687e-03  0.040167713
## 4        f1 2.507524e+00 1.133567e-01 7.934970e-01  0.028626056
## 5       f10 4.496325e+01 2.191933e-11 3.726287e-10 -0.077066927
## 6        f2 2.769807e+00 9.611015e-02 7.688812e-01  0.039770177
## 7        f3 6.405035e+00 1.140515e-02 1.140515e-01 -0.009287120
## 8        f4 1.232098e+01 4.511956e-04 4.963152e-03 -0.040435637
## 9        f5 4.042170e+00 4.442154e-02 3.997938e-01 -0.020115647
## 10       f6 3.330935e+01 8.254793e-09 1.320767e-07 -0.069395926
## 11       f7 1.419590e+00 2.335182e-01 9.340728e-01 -0.004495350
## 12       f8 2.172639e+00 1.405376e-01 8.231017e-01  0.040132494
## 13       f9 1.738971e+01 3.087756e-05 4.322858e-04  0.043110431
## 14      len 3.263438e+01 1.165884e-08 1.748826e-07  0.066296284
## 15      rim 1.601153e-01 6.890647e-01 1.000000e+00 -0.011359450
## 16  surface 1.313674e+00 2.517763e-01 9.340728e-01  0.003072155
## 17     turn 1.949782e-05 9.964770e-01 1.000000e+00  0.013760077
##            pr_P     pr_P_adj
## 1  5.460965e-01 1.000000e+00
## 2  1.858793e-03 2.287575e-02
## 3  1.887971e-03 2.287575e-02
## 4  2.682691e-02 2.146153e-01
## 5  2.398130e-09 4.076821e-08
## 6  2.094220e-03 2.287575e-02
## 7  4.726569e-01 1.000000e+00
## 8  1.759673e-03 2.287575e-02
## 9  1.197926e-01 8.385484e-01
## 10 7.757830e-08 1.241253e-06
## 11 7.281295e-01 1.000000e+00
## 12 1.905461e-03 2.287575e-02
## 13 8.525018e-04 1.193502e-02
## 14 2.864445e-07 4.296667e-06
## 15 3.797142e-01 1.000000e+00
## 16 8.122210e-01 1.000000e+00
## 17 2.872929e-01 1.000000e+00
```

```
print(dt.hip.prop.1 %>% filter(pr_P_adj < 0.01 & pa_P_adj < 0.01))
```

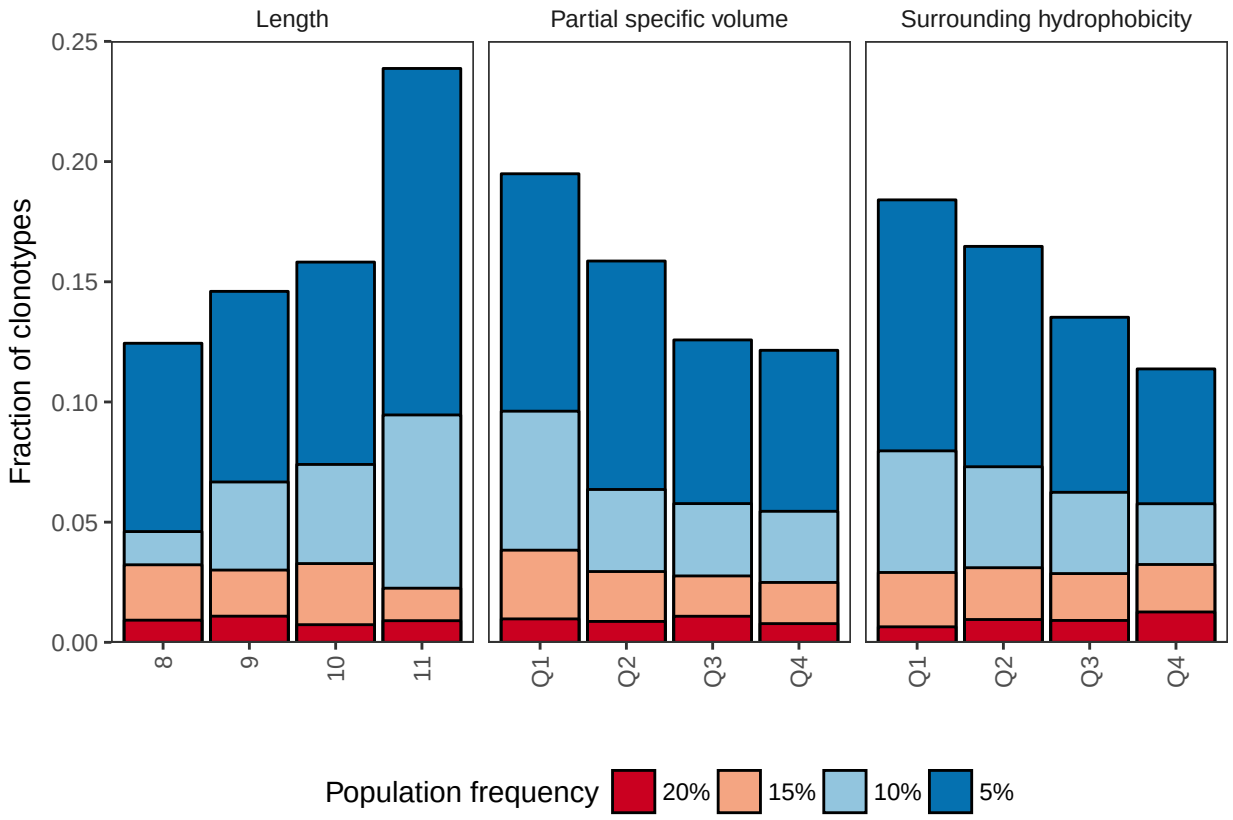
```
##   variable      F_P         pa_P     pa_P_adj         r_P         pr_P
## 1      f10 44.96325 2.191933e-11 3.726287e-10 -0.07706693 2.398130e-09
## 2       f6 33.30935 8.254793e-09 1.320767e-07 -0.06939593 7.757830e-08
## 3      len 32.63438 1.165884e-08 1.748826e-07  0.06629628 2.864445e-07
##       pr_P_adj
## 1 4.076821e-08
## 2 1.241253e-06
## 3 4.296667e-06
```

```
df.hip.frac.prop = df.hip %>%
  merge(dt.hip.prop %>%
          select(antigen.epitope, variable, value_c) %>%
          unique %>%
          merge(dt.hip.prop.1 %>% filter(pr_P_adj < 0.01 & pa_P_adj < 0.01) %>% select(variable) %>% uni
        by = "antigen.epitope", allow.cartesian = T) %>%
  group_by(variable, value_c) %>%
```

```
  summarise(publ_5 = sum(incidence > 0.05) / n(),
            publ_10 = sum(incidence > 0.1) / n(),
            publ_15 = sum(incidence > 0.15) / n(),
            publ_20 = sum(incidence > 0.20) / n())


df.hip.frac.prop$property = df.hip.frac.prop$variable
df.hip.frac.prop$variable = NULL
df.hip.frac.prop$property_level = df.hip.frac.prop$value_c
df.hip.frac.prop$value_c = NULL

df.hip.frac.prop = df.hip.frac.prop %>%
  melt
```

```
## Using property, property_level as id variables
```

```
df.hip.frac.prop$variable = with(df.hip.frac.prop,
                                 factor(paste0(str_split_fixed(variable, "_", 2)[,2], "%"), levels = rev

df.hip.frac.prop$property_level2 = factor(with(df.hip.frac.prop,
                                       ifelse(property == "len", as.character(as.integer(property_leve
                                       levels = c("Q1","Q2","Q3","Q4", "8", "9", "10", "11"))
df.hip.frac.prop$property2 = with(df.hip.frac.prop,
                                  ifelse(property=="len","Length", ifelse(property=="f6", "Partial speci
# 6 - Partial specific volume,
# 10 - Surrounding hydrophobicity
p7=ggplot(df.hip.frac.prop,
       aes(x=property_level2, y = value,
           fill = variable)) +
  geom_bar(stat="identity", color = "black", position = "identity") +
  scale_fill_brewer("Population frequency", palette = "RdBu") +
  scale_x_discrete("") +
  scale_y_continuous("Fraction of clonotypes", expand = c(0,0), limits = c(0,0.25)) +
  facet_wrap(~property2, scales = "free_x") +
  theme_bw() +
    theme(legend.position = "bottom", axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        strip.background = element_blank())
p7
```



Figures

```
ggsave("figures/p1.pdf", p1, width = 4, height = 4)
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 920 rows containing non-finite values (stat_density2d).
```

```
ggsave("figures/p2.pdf", p2, width = 4, height = 4)
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Removed 22 rows containing non-finite values (stat_density2d).
```

```
## `geom_smooth()` using method = 'gam'
```

```
## Warning: Removed 22 rows containing non-finite values (stat_smooth).
```

```
ggsave("figures/p3.pdf", p3, width = 4*2, height = 4)
ggsave("figures/p4.pdf", p4, width = 4, height = 4)
ggsave("figures/p5.pdf", p5, width = 4*2, height = 4)
ggsave("figures/p6.pdf", p6, width = 4, height = 4)
ggsave("figures/p7.pdf", p7, width = 4*2, height = 4)
```