Exploratory data analysis-1

```
library(ggplot2)
library(dplyr)
```

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

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

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

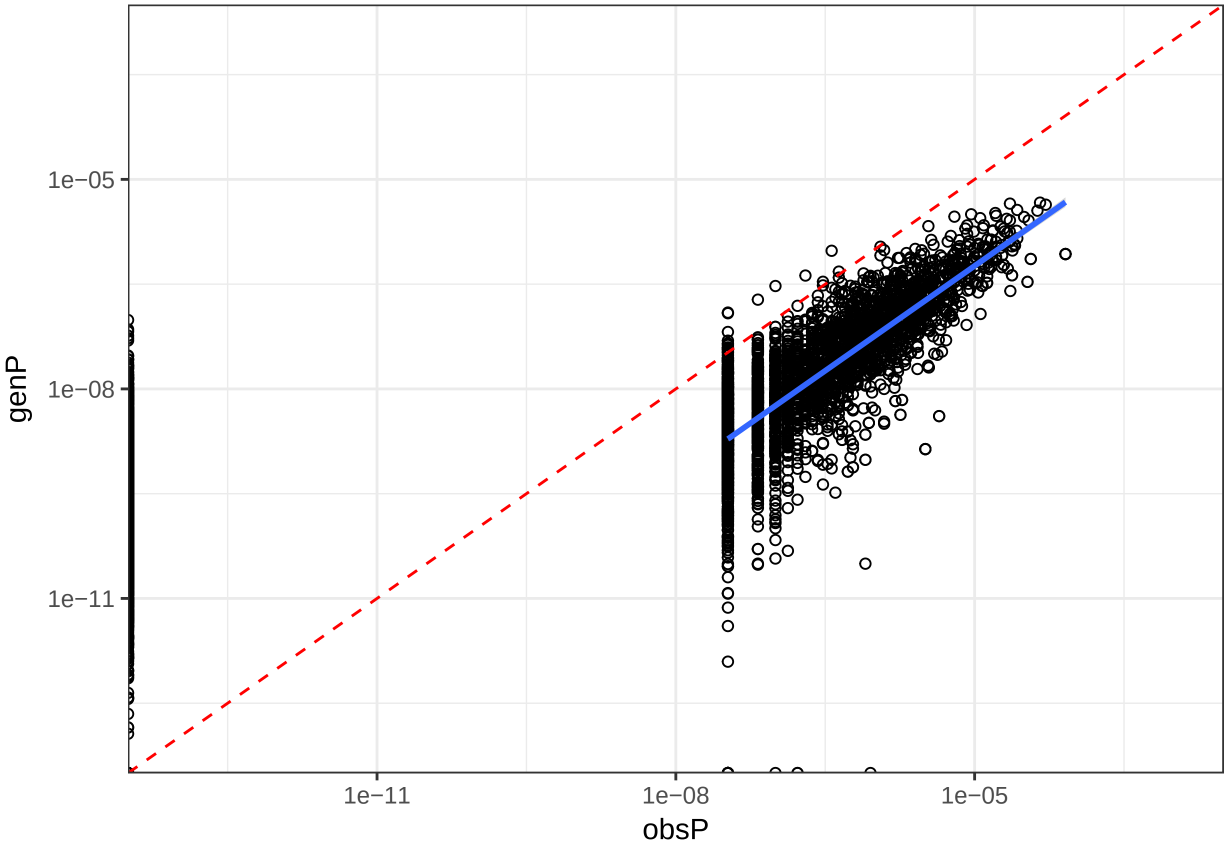
Added columns: `genP` full generation probability weighted by VJ usage for our data (aging); `ageing_occur` - number of occurrences in aging dataset, should be normalized by `29,989,055` - total number of rearrangements in aging data.

```
TOTAL_REARRANGEMENTS_AGING = 29989055

df = read.table("VDJDB_fullP.txt", header=T, sep="\t")

# Fix issue with SLYNTVATL epitope labelled as CMV in 1555537 -> should put an issue in vdjdb-db!
# which is actually HIV
df$antigen.species = as.factor(ifelse(df$antigen.epitope == "SLYNTVATL", "HIV-1", as.character(df$antig

df$obsP = df$ageing_occur / TOTAL_REARRANGEMENTS_AGING
```

Only intercept (constant) bias is present (from plot):

```
ggplot(df, aes(x=obsP, y=genP)) +
  geom_point(shape=21)+
  geom_smooth(method="lm") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  scale_x_log10(limits = c(1e-13, 1e-3)) +
  scale_y_log10(limits = c(1e-13, 1e-3)) +
  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 3632 rows containing non-finite values (stat_smooth).
```



```
# Not quite obvious from ANOVA..

lmP = lm(log10(obsP) ~ log10(genP), subset(df, obsP > 0 & genP > 0))
summary(lmP)
```

```
##
## Call:
## lm(formula = log10(obsP) ~ log10(genP), data = subset(df, obsP >
##     0 & genP > 0))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.50718 -0.30004  0.00176  0.28453  2.19940
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.500653   0.069658  -21.54   <2e-16 ***
## log10(genP)  0.647065   0.008907   72.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4461 on 2873 degrees of freedom
## Multiple R-squared:  0.6475, Adjusted R-squared:  0.6474
## F-statistic:  5278 on 1 and 2873 DF,  p-value: < 2.2e-16
```

```
anova(lmP)
```

```
## Analysis of Variance Table
```

```
##
## Response: log10(obsP)
##                Df  Sum Sq Mean Sq F value    Pr(>F)
## log10(genP)     1 1050.23  1050.2  5277.5 < 2.2e-16 ***
## Residuals    2873  571.73     0.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Lets try GLM

glmP = glm(obsP * TOTAL_REARRANGEMENTS_AGING ~ genP, data = df, family = poisson)
summary(glmP)
```

```
##
## Call:
## glm(formula = obsP * TOTAL_REARRANGEMENTS_AGING ~ genP, family = poisson,
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -73.017   -5.567   -5.567   -3.180  121.239
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.741e+00  3.083e-03   888.9   <2e-16 ***
## genP        1.268e+06  1.795e+03   706.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 622590  on 6506  degrees of freedom
## Residual deviance: 430074  on 6505  degrees of freedom
## AIC: 441862
##
## Number of Fisher Scoring iterations: 7
```

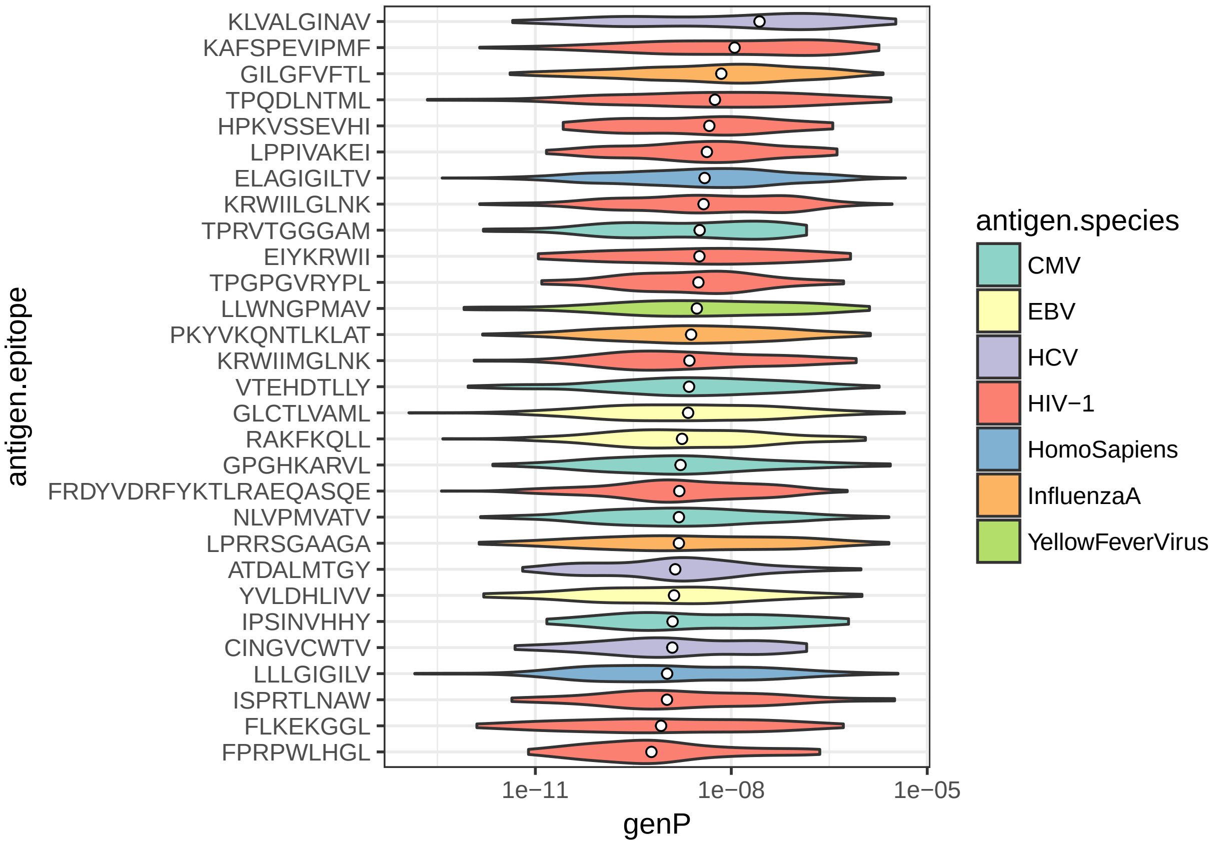
Comparing genP across epitopes

Filter epitopes with few representative TCRs, remove everything with 0 generation prob

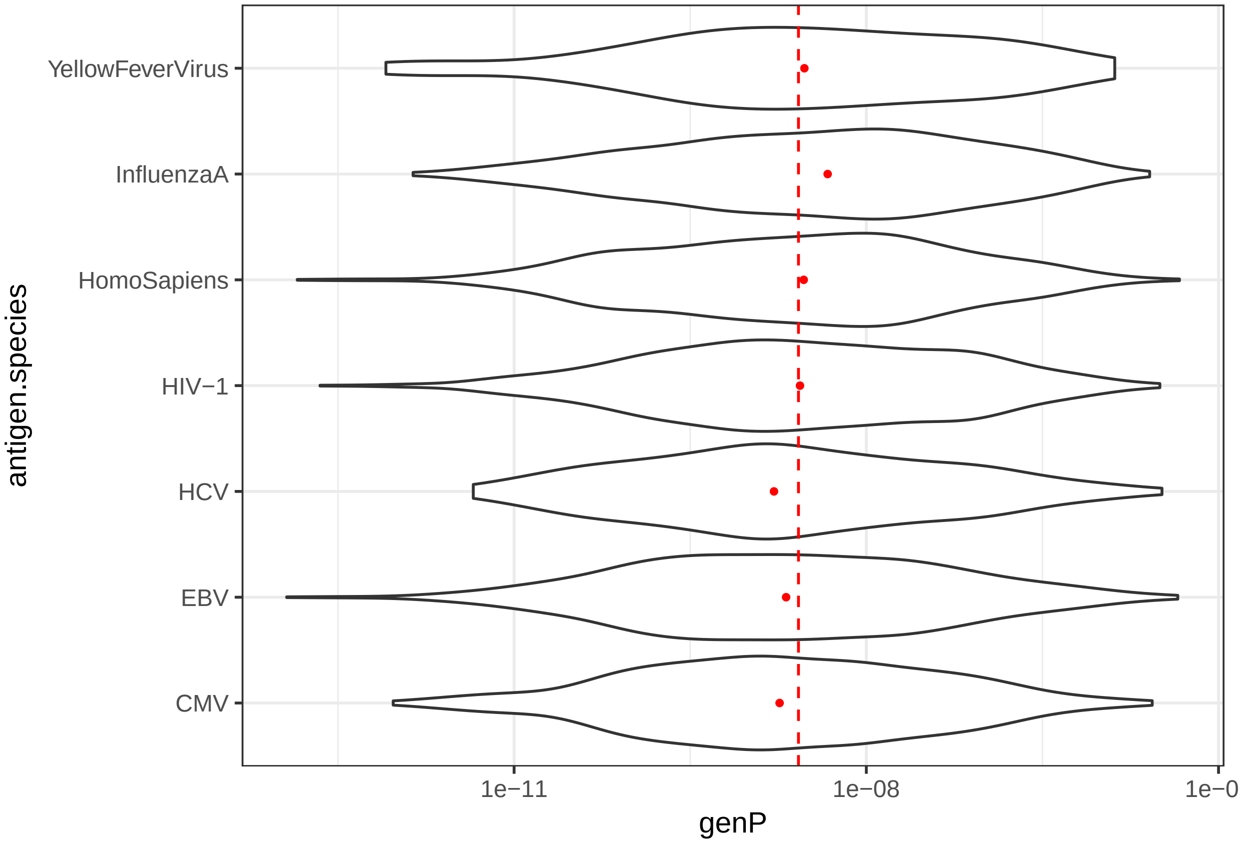
```
df.tcr.per.epitope = df %>%
  filter(genP > 0) %>%
  group_by(antigen.epitope) %>%
  dplyr::summarise(count = n(), genP_med = median(genP)) %>%
  filter(count >= 30)
```

Compare rearrangement prob across epitopes and their parent species

```
df.1 = subset(df, antigen.epitope %in% df.tcr.per.epitope$antigen.epitope & genP > 0)
df.1$antigen.epitope = factor(df.1$antigen.epitope, levels = df.tcr.per.epitope$antigen.epitope[order(d

ggplot(df.1, aes(x=antigen.epitope, group = antigen.epitope, y=genP, fill = antigen.species)) +
  geom_violin() + stat_summary(fun.y=median, geom="point", shape=21, fill = "white", color="black") +
  scale_y_log10() +
  coord_flip() +
  scale_fill_brewer(palette = "Set3") +
  theme_bw()
```



```
ggplot(df.1, aes(x=antigen.species, group = antigen.species, y=genP)) +
  geom_violin() + stat_summary(fun.y=median, geom="point", shape=21, fill = "red", color="white") +
  geom_hline(yintercept = median(df.1$genP), linetype = "dashed", color = "red") +
  scale_y_log10() +
  coord_flip() +
  theme_bw()
```



```r
a1 = aov(log10(genP) ~ antigen.epitope, df.1)
summary(a1)
```

```
##                   Df Sum Sq Mean Sq F value  Pr(>F)
## antigen.epitope   28    230   8.197   4.959 2.4e-16 ***
## Residuals       4564   7544   1.653
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
a2 = aov(log10(genP) ~ antigen.species, df.1)
summary(a2)
```

```
##                   Df Sum Sq Mean Sq F value   Pr(>F)
## antigen.species    6     40   6.692   3.968 0.000578 ***
## Residuals       4586   7734   1.686
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
TukeyHSD(a2, "antigen.species")
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = log10(genP) ~ antigen.species, data = df.1)
##
## $antigen.species
##                                         diff         lwr       upr     p adj
## EBV-CMV                          0.043890929 -0.15589589 0.2436777 0.9951538
## HCV-CMV                          0.049775136 -0.24522150 0.3447718 0.9988914
## HIV-1-CMV                     0.172112230 -0.01339119 0.3576157 0.0895409
## HomoSapiens-CMV               0.120723624 -0.07063342 0.3120807 0.5066893
## InfluenzaA-CMV                0.278978450  0.08441335 0.4735436 0.0004767
## YellowFeverVirus-CMV          0.186990017 -0.30935016 0.6833302 0.9249008
## HCV-EBV                       0.005884207 -0.28680542 0.2985738 1.0000000
## HIV-1-EBV                     0.128221301 -0.05359102 0.3100336 0.3646205
## HomoSapiens-EBV               0.076832695 -0.11094834 0.2646137 0.8917625
## InfluenzaA-EBV                0.235087522  0.04403837 0.4261367 0.0053305
## YellowFeverVirus-EBV          0.143099088 -0.35187341 0.6380716 0.9791885
## HIV-1-HCV                     0.122337094 -0.16079532 0.4054695 0.8637450
## HomoSapiens-HCV               0.070948488 -0.21605319 0.3579502 0.9908239
## InfluenzaA-HCV                0.229203315 -0.05994720 0.5183538 0.2260047
## YellowFeverVirus-HCV          0.137214881 -0.40325345 0.6776832 0.9894256
## HomoSapiens-HIV-1            -0.051388606 -0.22389503 0.1211178 0.9757661
## InfluenzaA-HIV-1              0.106866221 -0.06919208 0.2829245 0.5546221
## YellowFeverVirus-HIV-1        0.014877787 -0.47450399 0.5042596 1.0000000
## InfluenzaA-HomoSapiens        0.158254827 -0.02396077 0.3404704 0.1381275
## YellowFeverVirus-HomoSapiens  0.066266393 -0.42536408 0.5578969 0.9996950
## YellowFeverVirus-InfluenzaA  -0.091988434 -0.58487643 0.4008996 0.9980396
```

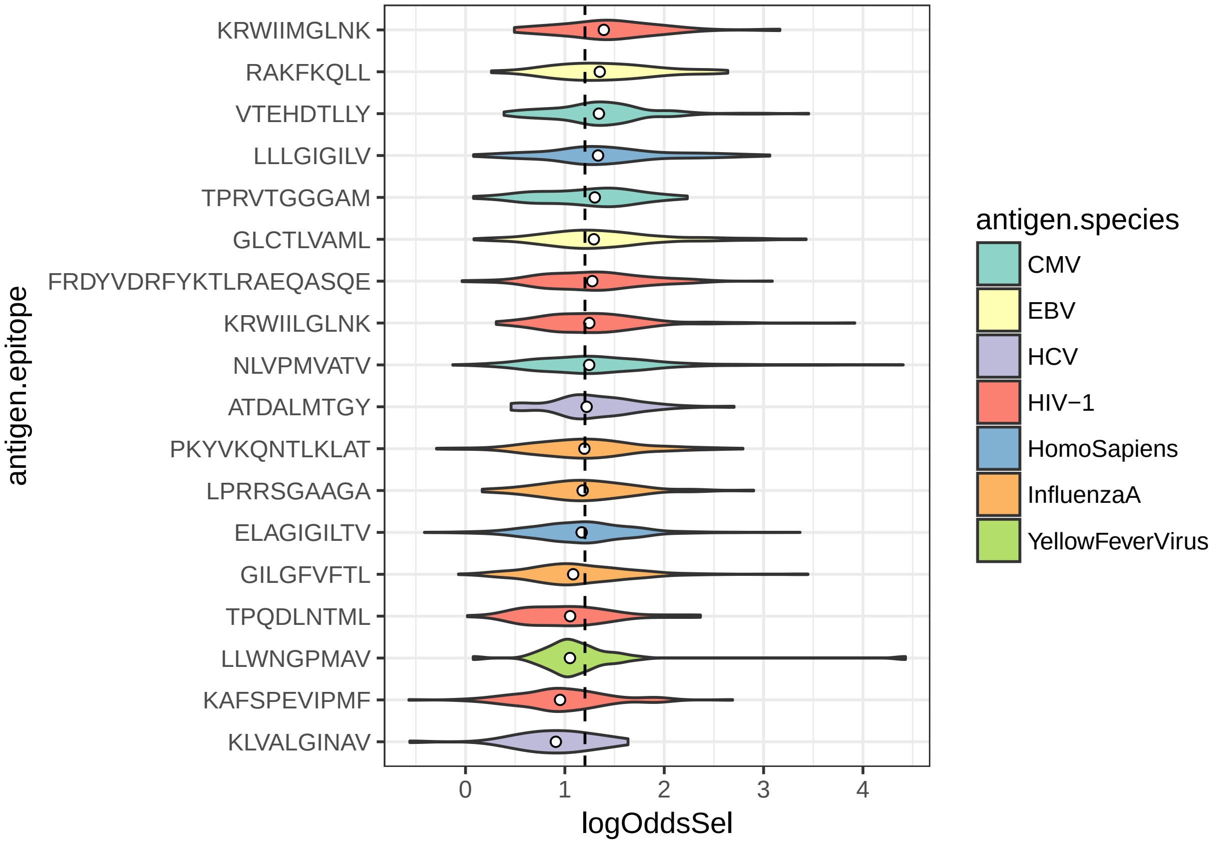
Comparing selection prob

Pre-filter

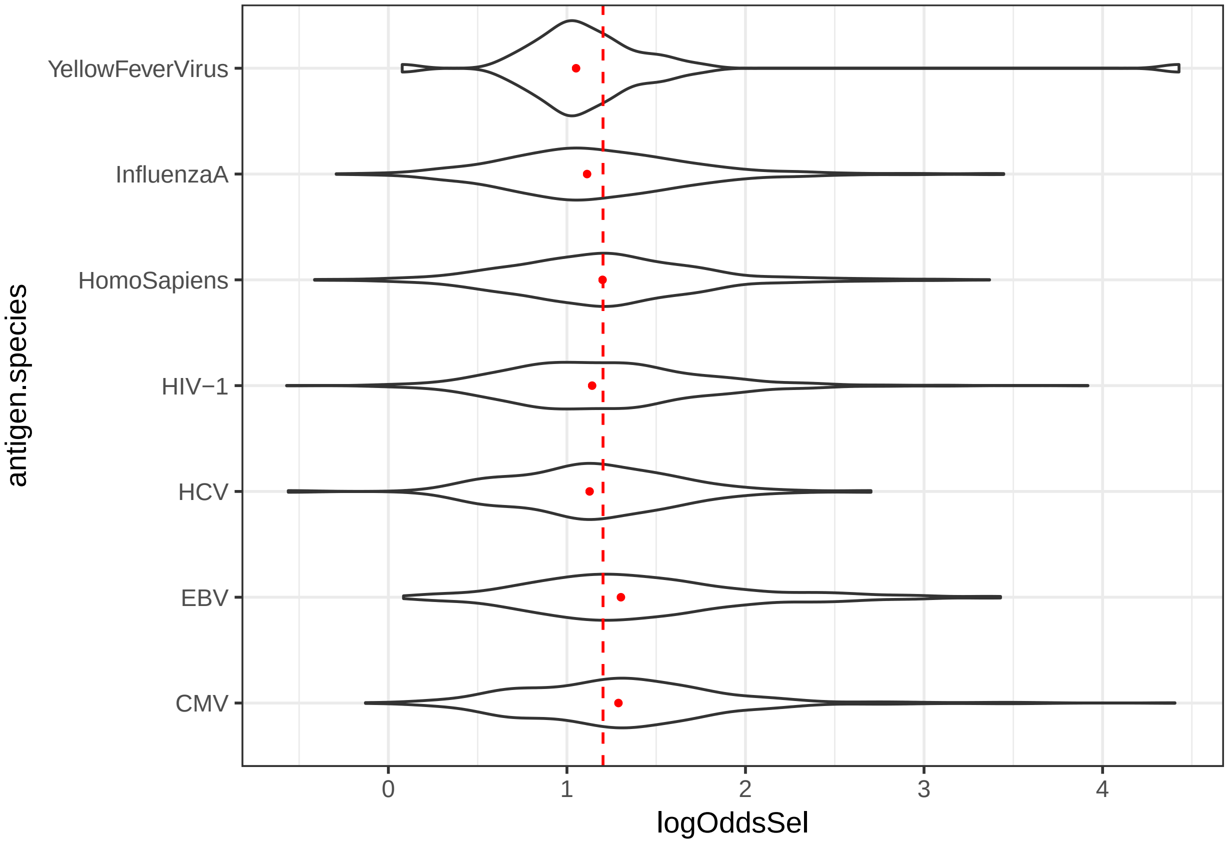
```
df.tcr.per.epitope.2 = df %>%
  filter(genP > 0 & obsP > 0) %>%
  group_by(antigen.epitope) %>%
  dplyr::summarise(count = n(), logOddsSel_med = median(log10(obsP) - log10(genP))) %>%
  filter(count >= 30)

df.2 = subset(df, antigen.epitope %in% df.tcr.per.epitope.2$antigen.epitope & genP > 0 & obsP > 0)
df.2$logOddsSel = with(df.2, log10(obsP) - log10(genP))
df.2$antigen.epitope = factor(df.2$antigen.epitope, levels = df.tcr.per.epitope.2$antigen.epitope[order

med_log_odds = median(df.2$logOddsSel)
```

Compare selection factors

```
ggplot(df.2, aes(x=antigen.epitope, group = antigen.epitope, y=logOddsSel, fill = antigen.species)) +
  geom_violin() + stat_summary(fun.y=median, geom="point", shape=21, fill = "white", color="black") +
  geom_hline(yintercept = med_log_odds, linetype = "dashed", color = "black") +
  coord_flip() +
  scale_fill_brewer(palette = "Set3") +
  theme_bw()
```



```
ggplot(df.2, aes(x=antigen.species, group = antigen.species, y=logOddsSel)) +
  geom_violin() + stat_summary(fun.y=median, geom="point", shape=21, fill = "red", color="white") +
  geom_hline(yintercept = med_log_odds, linetype = "dashed", color = "red") +
  coord_flip() +
  theme_bw()
```



```r
a1 = aov(logOddsSel ~ antigen.epitope, df.2)
summary(a1)
```

```
##                   Df Sum Sq Mean Sq F value   Pr(>F)    
## antigen.epitope   17     34  2.0029   6.628 1.02e-15 ***
## Residuals       2333    705  0.3022                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
a2 = aov(logOddsSel ~ antigen.species, df.2)
summary(a2)
```

```
##                   Df Sum Sq Mean Sq F value   Pr(>F)    
## antigen.species    6   15.9  2.6429   8.566 3.13e-09 ***
## Residuals       2344  723.2  0.3085                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
TukeyHSD(a2, "antigen.species")
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
## 
## Fit: aov(formula = logOddsSel ~ antigen.species, data = df.2)
## 
## $antigen.species
##                                      diff          lwr           upr
## EBV-CMV                        0.07615086 -0.04581397  0.1981156819
## HCV-CMV                       -0.17590525 -0.36754090  0.0157303872
## HIV-1-CMV                    -0.11772431 -0.23591821  0.0004695923
## HomoSapiens-CMV              -0.10193062 -0.21729323  0.0134319894
## InfluenzaA-CMV               -0.14566815 -0.26133593 -0.0300003714
## YellowFeverVirus-CMV         -0.14197351 -0.43720897  0.1532619512
## HCV-EBV                      -0.25205611 -0.43998231 -0.0641299075
## HIV-1-EBV                    -0.19387516 -0.30595476 -0.0817955644
## HomoSapiens-EBV              -0.17808148 -0.28717121 -0.0689917412
## InfluenzaA-EBV               -0.22181900 -0.33123140 -0.1124066047
## YellowFeverVirus-EBV         -0.21812436 -0.51096564  0.0747169155
## HIV-1-HCV                     0.05818095 -0.12732010  0.2436819901
## HomoSapiens-HCV               0.07397463 -0.10973539  0.2576846521
## InfluenzaA-HCV                0.03023711 -0.15366470  0.2141389106
## YellowFeverVirus-HCV          0.03393175 -0.29409998  0.3619634758
## HomoSapiens-HIV-1             0.01579369 -0.08906312  0.1206504922
## InfluenzaA-HIV-1             -0.02794384 -0.13313629  0.0772486135
## YellowFeverVirus-HIV-1       -0.02424920 -0.31554011  0.2670417095
## InfluenzaA-HomoSapiens       -0.04373753 -0.14573844  0.0582633861
## YellowFeverVirus-HomoSapiens -0.04004289 -0.33019652  0.2501107415
## YellowFeverVirus-InfluenzaA   0.00369464 -0.28658046  0.2939697353
##                                  p adj
## EBV-CMV                      0.5192687
## HCV-CMV                      0.0964773
## HIV-1-CMV                    0.0517068
## HomoSapiens-CMV              0.1241213
## InfluenzaA-CMV               0.0038930
## YellowFeverVirus-CMV         0.7916860
## HCV-EBV                      0.0015125
## HIV-1-EBV                    0.0000074
## HomoSapiens-EBV              0.0000319
## InfluenzaA-EBV               0.0000001
## YellowFeverVirus-EBV         0.2968490
## HIV-1-HCV                    0.9685425
## HomoSapiens-HCV              0.8987796
## InfluenzaA-HCV               0.9990400
## YellowFeverVirus-HCV         0.9999348
## HomoSapiens-HIV-1            0.9994188
## InfluenzaA-HIV-1             0.9865313
## YellowFeverVirus-HIV-1       0.9999819
## InfluenzaA-HomoSapiens       0.8677225
## YellowFeverVirus-HomoSapiens 0.9996491
## YellowFeverVirus-InfluenzaA  1.0000000
```

Summary so far

- There is a difference in both rearrangement prob and selection prob across epitopes
- There is large difference in selection prob across species, no such difference for rearrangement prob
- Difference in selection prob shows that EBV and CMV are favored compared to other species. This can be due to clonal expansions, but: 1) no such difference for Flu 2) we don't account for clonal size, only counting unique rearrangements 3) A02-NLVP is not the top favoured epitope

Features

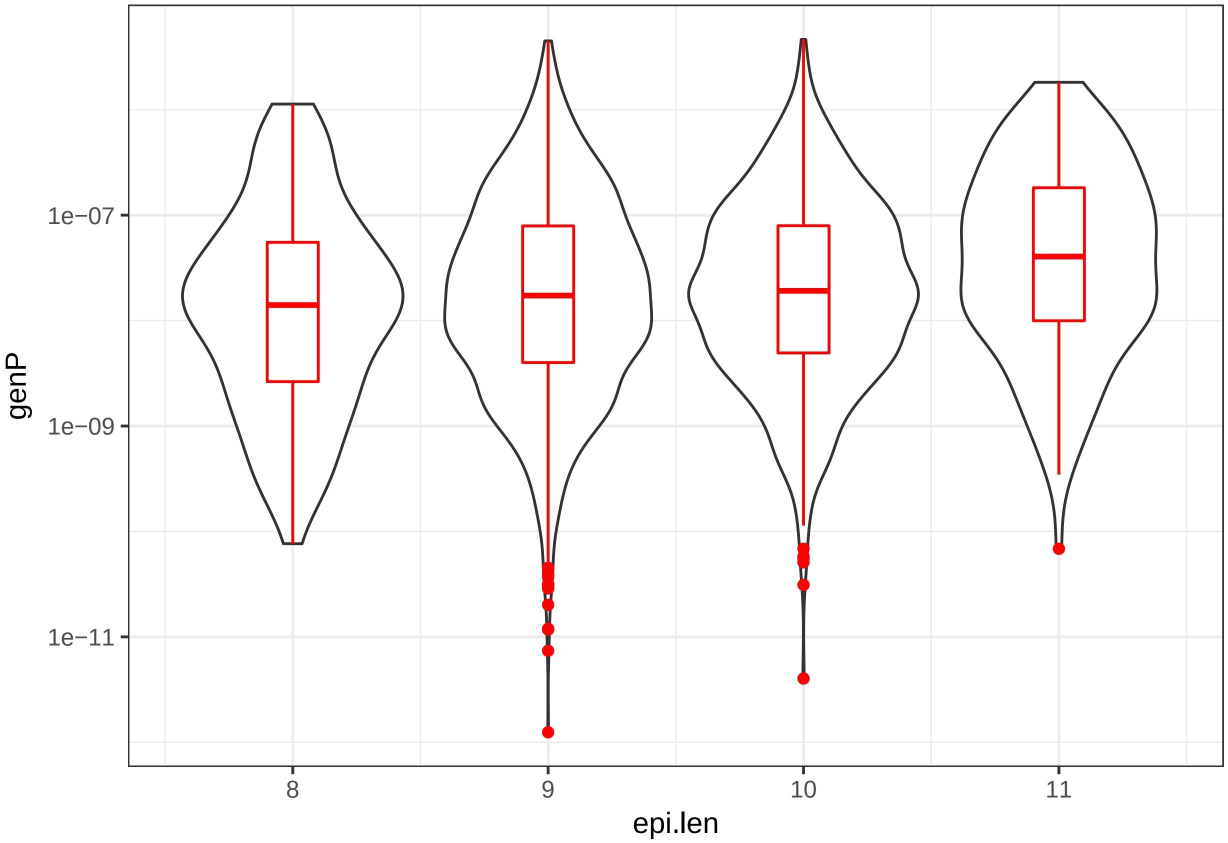
Epitope len for MHCI

Its more likely to generate TCR recognizing longer epitope, however the observed occurrence frequency is independent of the length of cognate epitope => differences in selection.

```
df.epi = df %>% filter(mhc.class=="MHCI" & genP > 0 & obsP > 0)
df.epi$epi.len = nchar(as.character(df.epi$antigen.epitope))
df.epi.s = df.epi %>%
  group_by(epi.len) %>%
  dplyr::summarise(count = n()) %>%
  arrange(-count)
print(df.epi.s)
```

```
## # A tibble: 7 × 2
##   epi.len count
##     <int> <int>
## 1       9  1533
## 2      10   805
## 3      11   142
## 4       8   125
## 5      12     2
## 6      13     2
## 7      15     1
```

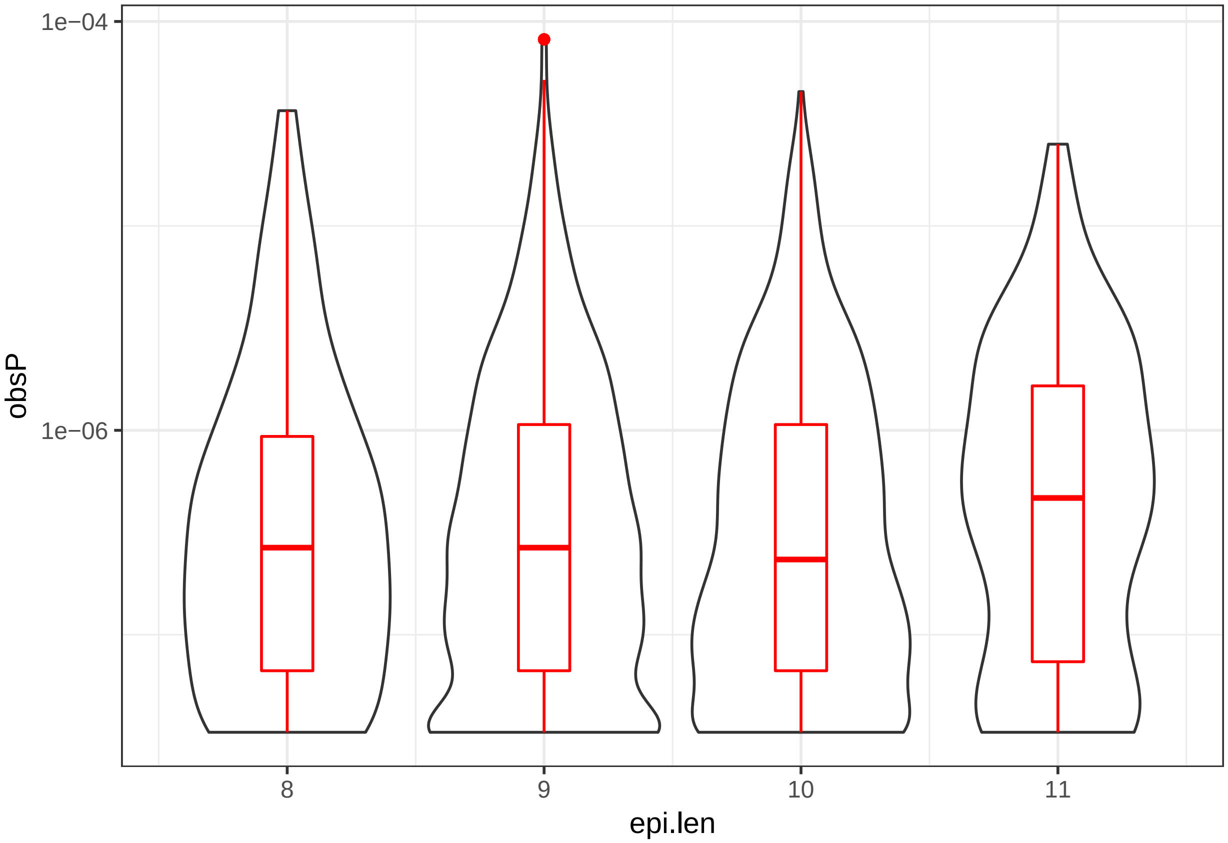
```
df.epi = df.epi %>% filter(epi.len < 12)

ggplot(df.epi, aes(x=epi.len, group=epi.len, y=genP)) +
  geom_violin() + geom_boxplot(color="red", width=0.2) +
  scale_y_log10() +
  theme_bw()
```



```
summary(lm(log10(genP) ~ epi.len, df.epi))
```

```
##
## Call:
## lm(formula = log10(genP) ~ epi.len, data = df.epi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1111 -0.6086  0.0077  0.6617  2.4467
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8.8034     0.2611 -33.713  < 2e-16 ***
## epi.len       0.1122     0.0278   4.035 5.61e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9391 on 2603 degrees of freedom
## Multiple R-squared:  0.006217,   Adjusted R-squared:  0.005835
## F-statistic: 16.28 on 1 and 2603 DF,  p-value: 5.61e-05
```

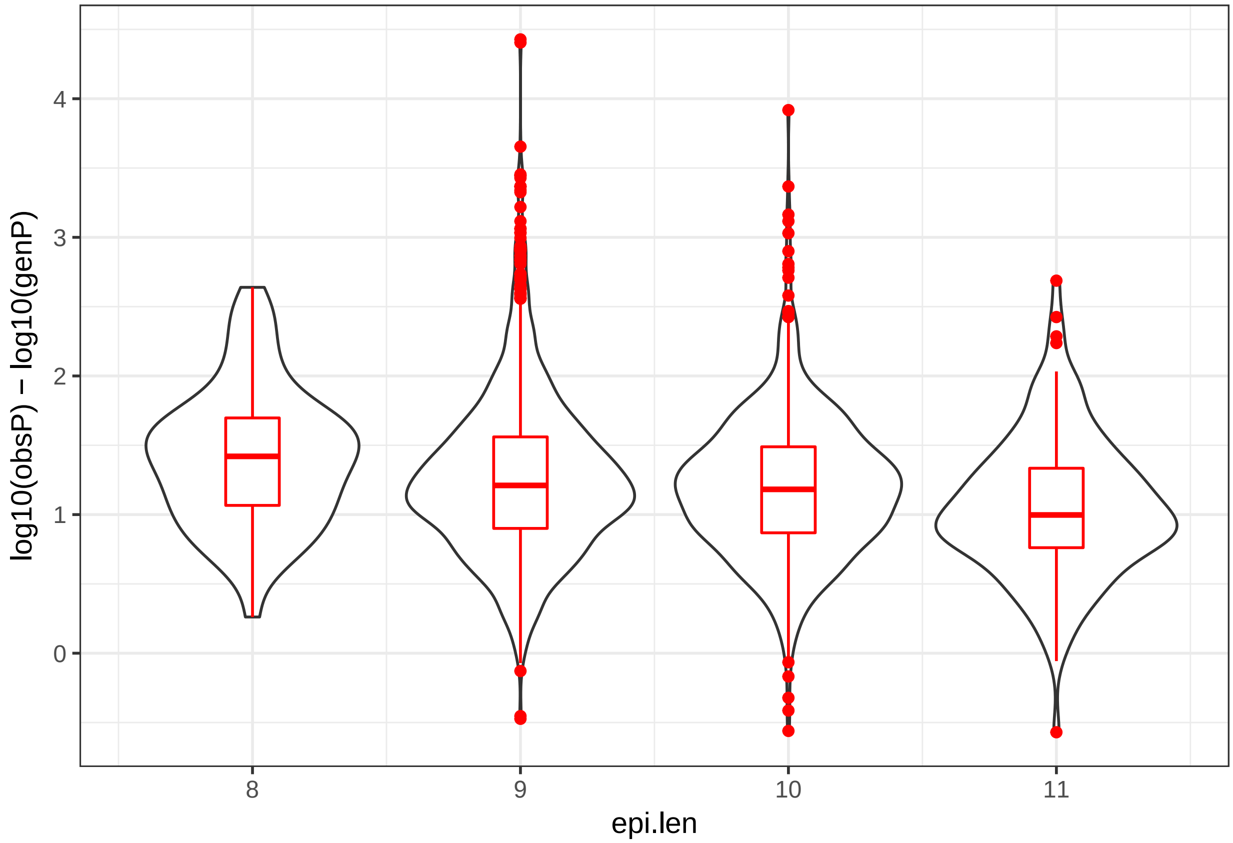
```
ggplot(df.epi, aes(x=epi.len, group=epi.len, y=obsP)) +
  geom_violin() + geom_boxplot(color="red", width=0.2) +
  scale_y_log10() +
  theme_bw()
```



```
summary(lm(log10(obsP) ~ epi.len, df.epi))
```

```
##
## Call:
## lm(formula = log10(obsP) ~ epi.len, data = df.epi)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98982 -0.65729 -0.07098  0.54683  2.43085
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.66039    0.20994 -31.725   <2e-16 ***
## epi.len      0.01575    0.02235   0.705    0.481
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.755 on 2603 degrees of freedom
## Multiple R-squared:  0.0001907,  Adjusted R-squared:  -0.0001934
## F-statistic: 0.4966 on 1 and 2603 DF,  p-value: 0.4811
```

```
ggplot(df.epi, aes(x=epi.len, group=epi.len, y=log10(obsP)-log10(genP))) +
  geom_violin() + geom_boxplot(color="red", width=0.2) +
  theme_bw()
```



```
summary(lm(log10(obsP)-log10(genP) ~ epi.len, df.epi))
```

```
##
## Call:
## lm(formula = log10(obsP) - log10(genP) ~ epi.len, data = df.epi)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.74844 -0.34320 -0.04468  0.29191  3.15282
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.14298    0.15415  13.902  < 2e-16 ***
## epi.len     -0.09643    0.01641  -5.876 4.73e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5544 on 2603 degrees of freedom
## Multiple R-squared:  0.01309,    Adjusted R-squared:  0.01271
## F-statistic: 34.53 on 1 and 2603 DF,  p-value: 4.732e-09
```

CDR3 features

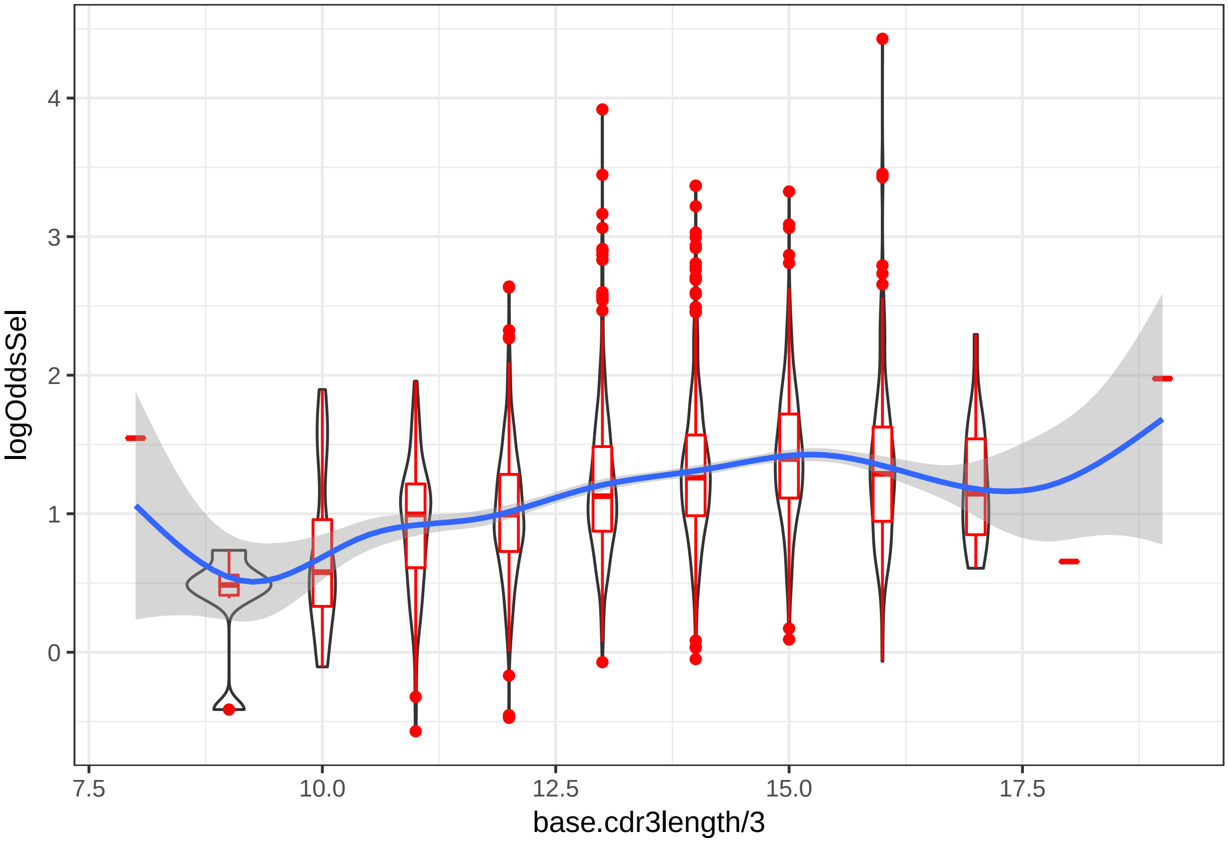
Load annotations produced by VDJdb/Annotate

```
# some dummy stuff
df.ann = read.table("ann.aging_annot_0.txt", header = T, sep = "\t") %>%
  merge(df) %>%
  filter(obsP > 0 & genP > 0) %>%
  mutate(logOddsSel = log10(obsP) - log10(genP))
```

The only effect comes from length..

```
ggplot(df.ann, aes(x=base.cdr3length / 3, y=logOddsSel)) +
  geom_violin(aes(group = base.cdr3length / 3)) +
  geom_boxplot(aes(group = base.cdr3length / 3), color="red", width=0.2) +
  geom_smooth() +
  theme_bw()
```

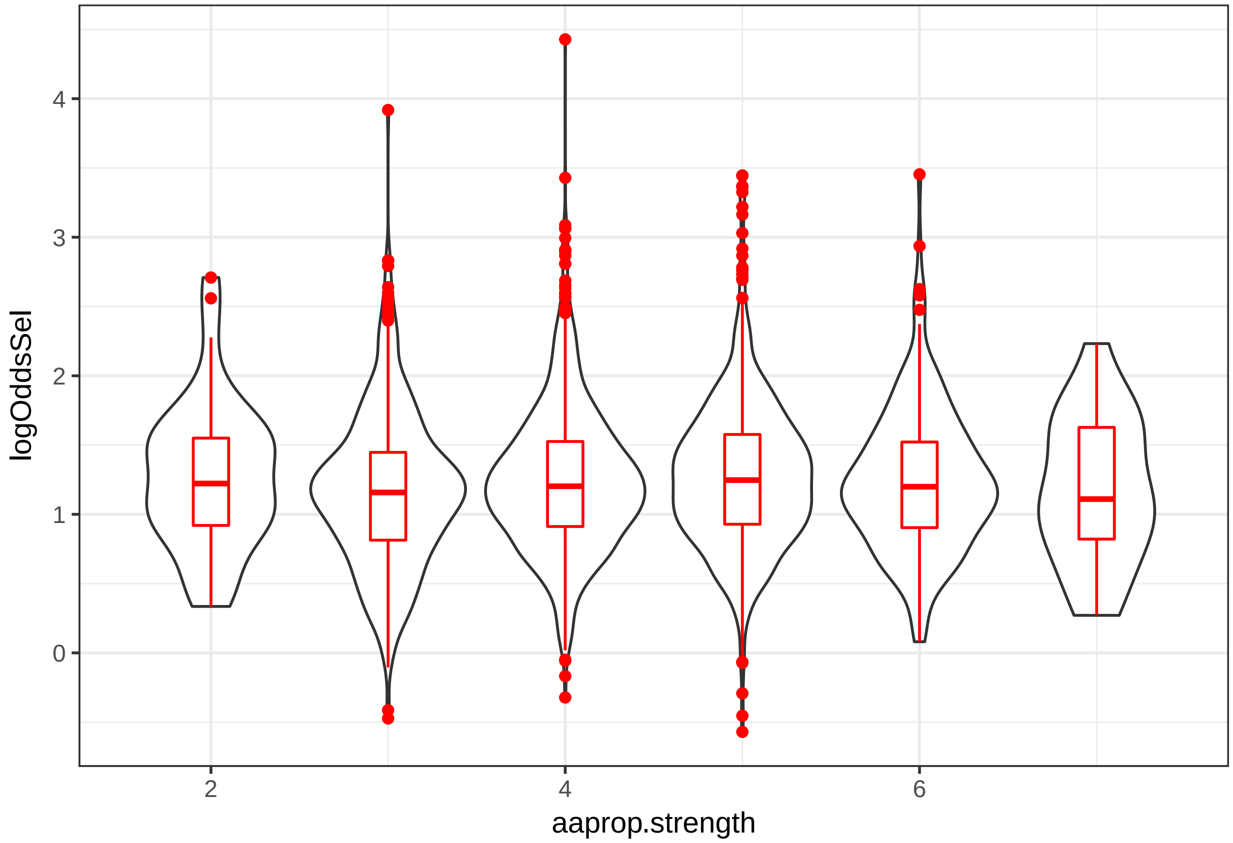
```
## `geom_smooth()` using method = 'gam'
```



```
ggplot(df.ann, aes(x=aaprop.strength, y=logOddsSel)) +
  geom_violin(aes(group = aaprop.strength)) +
  geom_boxplot(aes(group = aaprop.strength), color="red", width=0.2) +
  geom_smooth() +
  theme_bw()
```

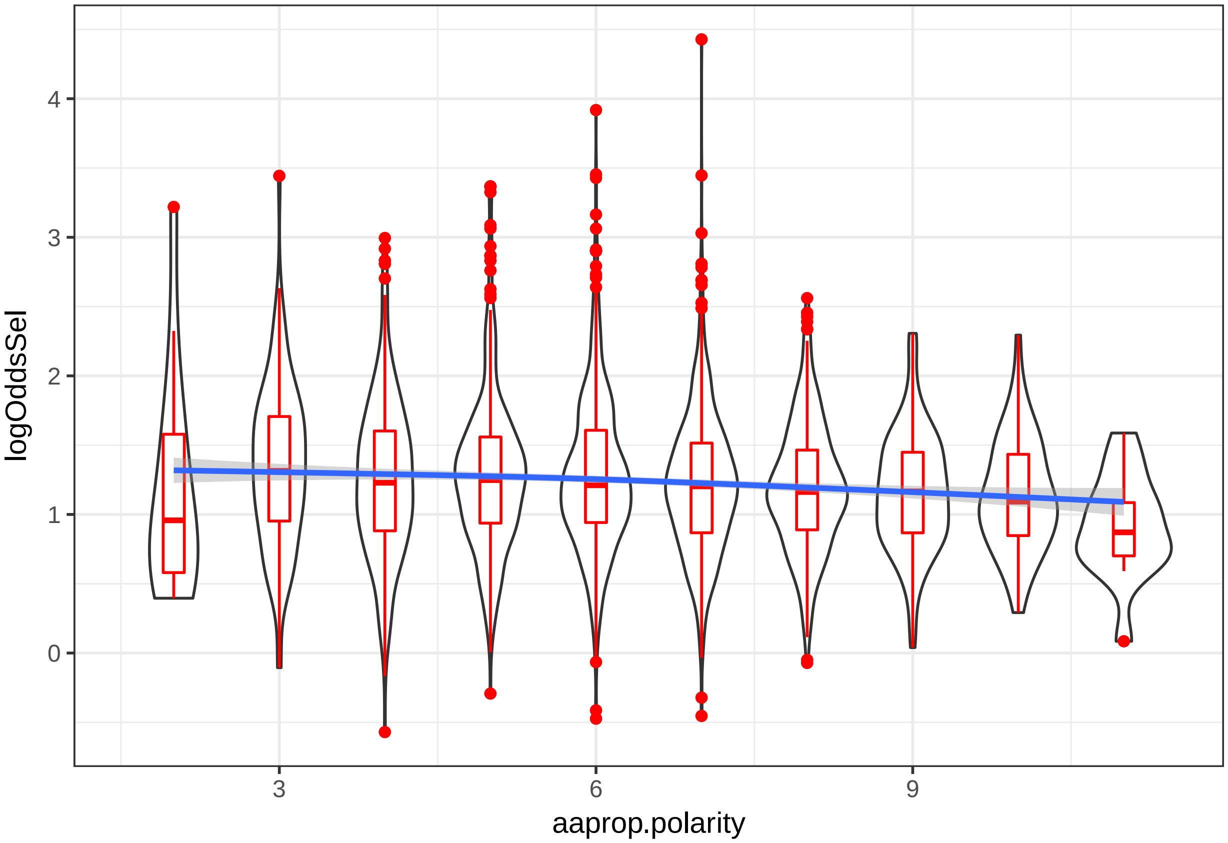
```
## `geom_smooth()` using method = 'gam'
```

```
## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.
```



```
ggplot(df.ann, aes(x=aaprop.polarity, y=logOddsSel)) +
  geom_violin(aes(group = aaprop.polarity)) +
  geom_boxplot(aes(group = aaprop.polarity), color="red", width=0.2) +
  geom_smooth() +
  theme_bw()
```

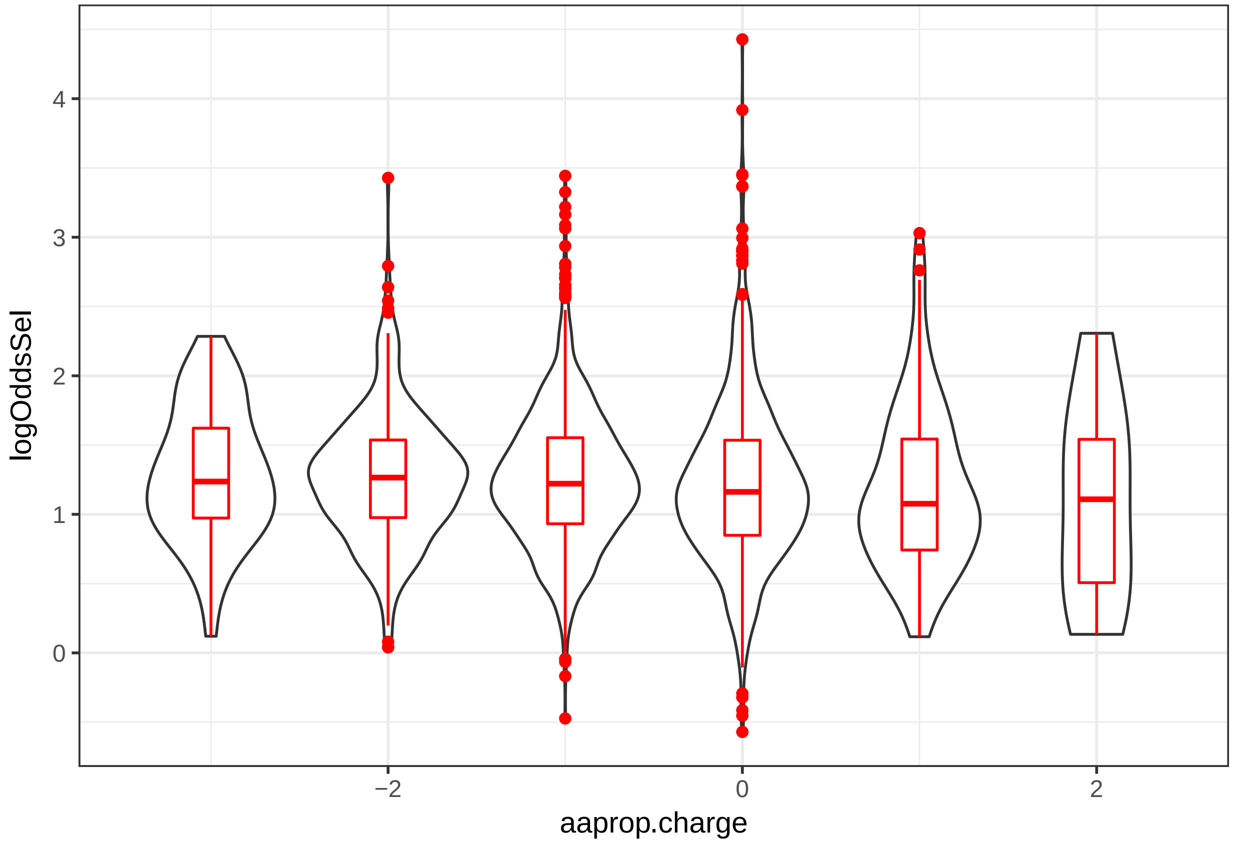
```
## `geom_smooth()` using method = 'gam'
```



```
ggplot(df.ann, aes(x=aaprop.charge, y=logOddsSel)) +
  geom_violin(aes(group = aaprop.charge)) +
  geom_boxplot(aes(group = aaprop.charge), color="red", width=0.2) +
  geom_smooth() +
  theme_bw()
```

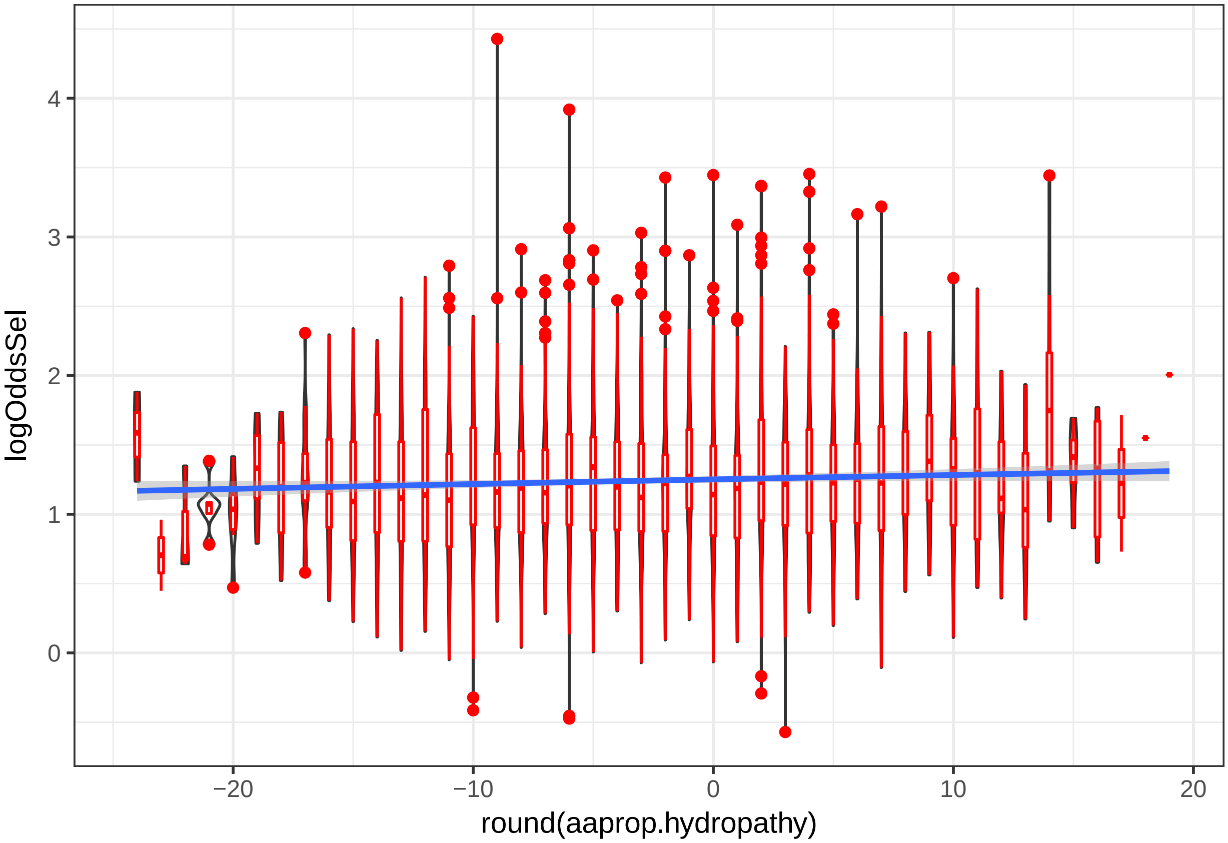
```
## `geom_smooth()` using method = 'gam'
```

```
## Warning: Computation failed in `stat_smooth()`:
## x has insufficient unique values to support 10 knots: reduce k.
```



```
ggplot(df.ann, aes(x=round(aaprop.hydropathy), y=logOddsSel)) +
  geom_violin(aes(group = round(aaprop.hydropathy))) +
  geom_boxplot(aes(group = round(aaprop.hydropathy)), color="red", width=0.2) +
  geom_smooth() +
  theme_bw()
```

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



Further work

Need to annotate Robins data. Check HLA-mediated effect. Can we consistently rule out clonal expansions.. well we can show effect both in CMV+ and CMV- patients for specific clonotypes