immunogenicity.Rmd

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

```
##
## 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
```

```r
library(stringdist)
library(reshape2)
```

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

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

```r
library(parallel)
library(EMCluster)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

## Loading required package: Matrix

##
## Attaching package: 'EMCluster'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
library(ggplot2)
select = dplyr::select
```

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

```r
dt.immv = merge(dt.imm %>% select(antigen.epitope, immunogenicity) %>% unique,
                dt.vdjdb %>%
                  filter(mhc.class == "MHCI", species == "HomoSapiens") %>%
                  group_by(antigen.epitope) %>% mutate(epi.count = n()) %>%
                  filter(epi.count > 30) %>%
                  summarise(pGen = median(genP_1mism_rob)),
                all.x=T, all.y=T)
```

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

dt.epi.prop = dt.epi.prop %>%
  merge(fread("kidera.txt") %>% mutate(len = 1) %>%
          melt, by = "aa", allow.cartesian = T) %>%
  group_by(antigen.epitope, variable) %>%
  summarise(value = sum(value))
```

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

```
dt.imm.prop = dt.immv %>%
  merge(dt.epi.prop) %>%
  dcast(antigen.epitope + immunogenicity + pGen ~ variable,
        value.var = "value")
```

PCA analysis

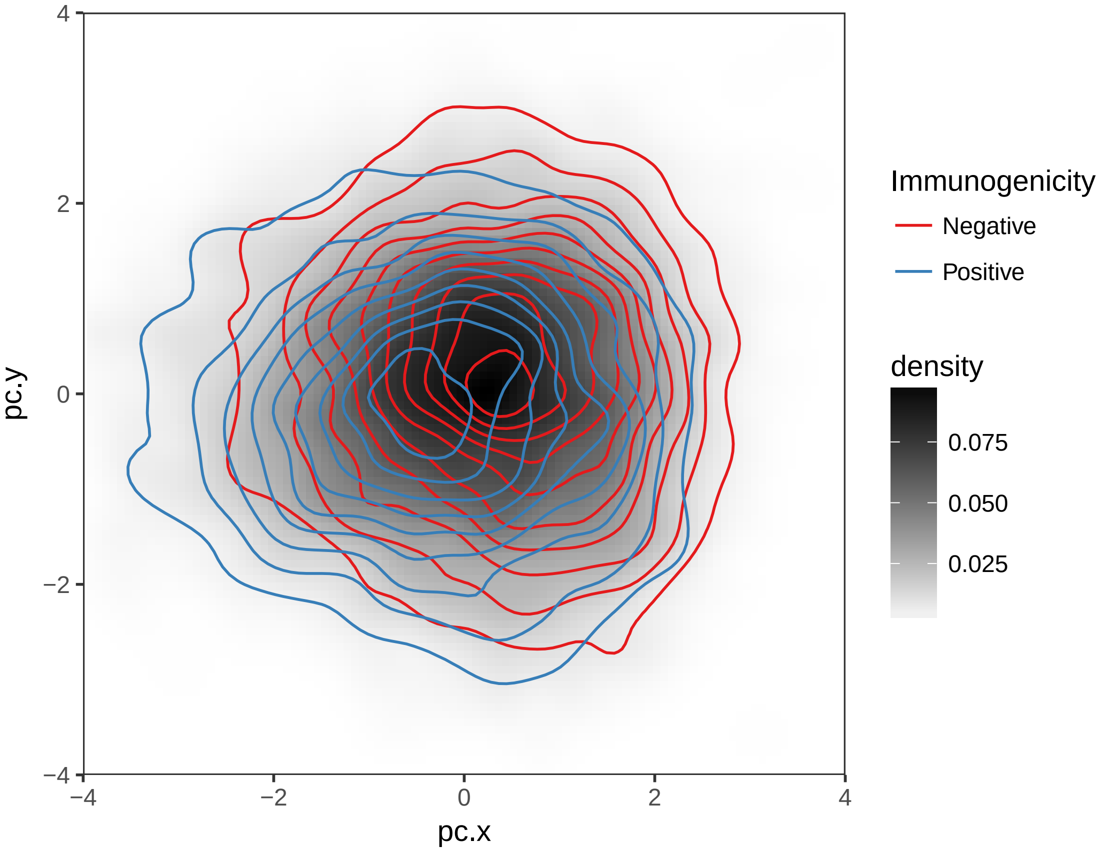
```
pc = prcomp(as.matrix(dt.imm.prop[,4:13]),
            scale = T, rank = 2)
```

```
dt.imm.prop$pc.x = pc$x[,1]
dt.imm.prop$pc.y = pc$x[,2]
```

```
p20=ggplot(dt.imm.prop %>% filter(!is.na(immunogenicity)), aes(x = pc.x, y = pc.y)) +
  stat_density_2d(data = dt.imm.prop %>% select(pc.x, pc.y), geom = "raster",
                 aes(fill = ..density..), contour = F) +
  geom_density2d(aes(color = immunogenicity)) +
  scale_color_brewer("Immunogenicity", palette = "Set1") +
  scale_fill_gradient(low = "white", high="black") +
  scale_x_continuous(expand=c(0,0), limits = c(-4,4))+
  scale_y_continuous(expand=c(0,0), limits = c(-4,4))+
  theme_bw() +
  theme(aspect = 1,
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
p20
```

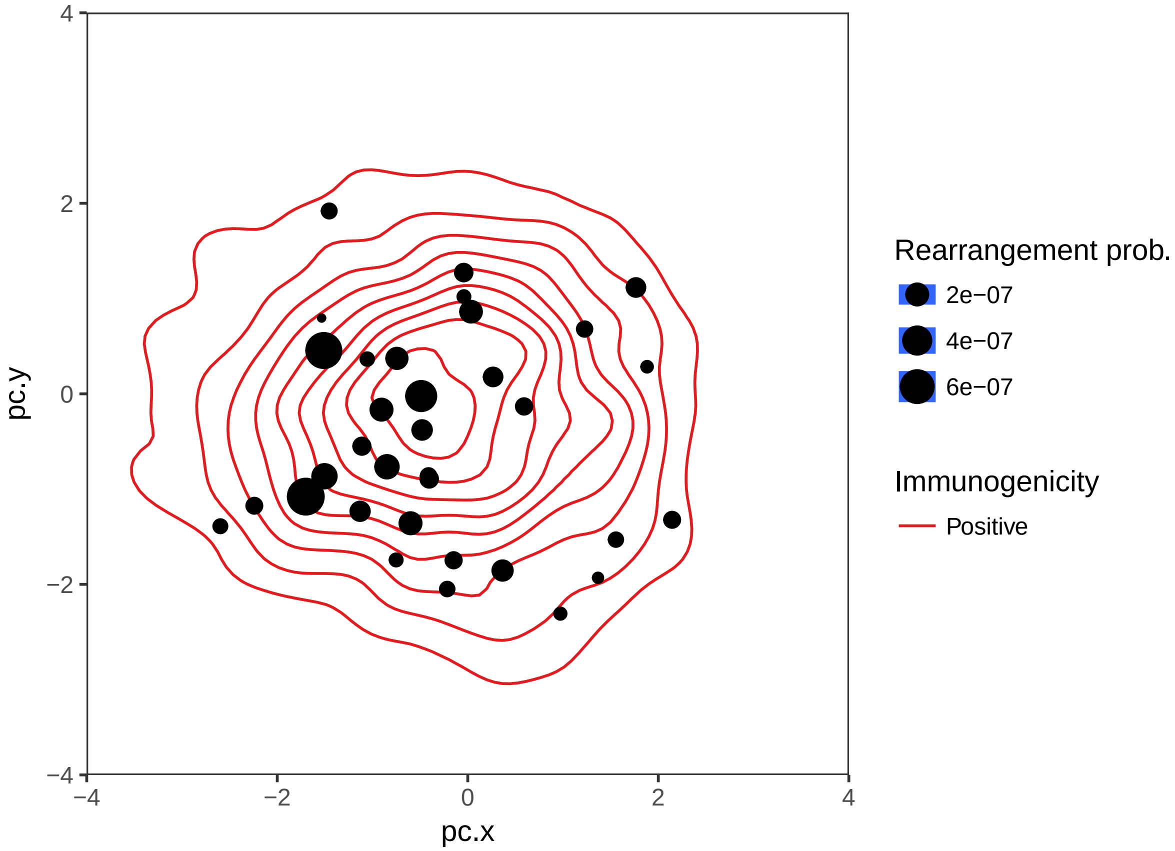
```
## Warning: Removed 56 rows containing non-finite values (stat_density2d).

## Warning: Removed 56 rows containing non-finite values (stat_density2d).
```



```
p21 = ggplot(dt.imm.prop %>% filter(!is.na(immunogenicity) & immunogenicity == "Positive"),
       aes(x = pc.x, y = pc.y)) +
  #stat_density_2d(data = dt.imm.prop %>% select(pc.x, pc.y), geom = "raster",
  #                aes(fill = ..density..), contour = F) +
  geom_density2d(aes(color = immunogenicity)) +
  geom_point(data=dt.imm.prop %>% filter(!is.na(pGen)), aes(size=pGen)) +
  scale_color_brewer("Immunogenicity", palette = "Set1") +
  scale_fill_gradient(low = "white", high="black") +
  scale_size_continuous("Rearrangement prob.") +
  scale_x_continuous(expand=c(0,0), limits = c(-4,4))+
  scale_y_continuous(expand=c(0,0), limits = c(-4,4))+
  theme_bw() +
  theme(aspect = 1,
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
p21
```

```
## Warning: Removed 50 rows containing non-finite values (stat_density2d).
```



Pure distances

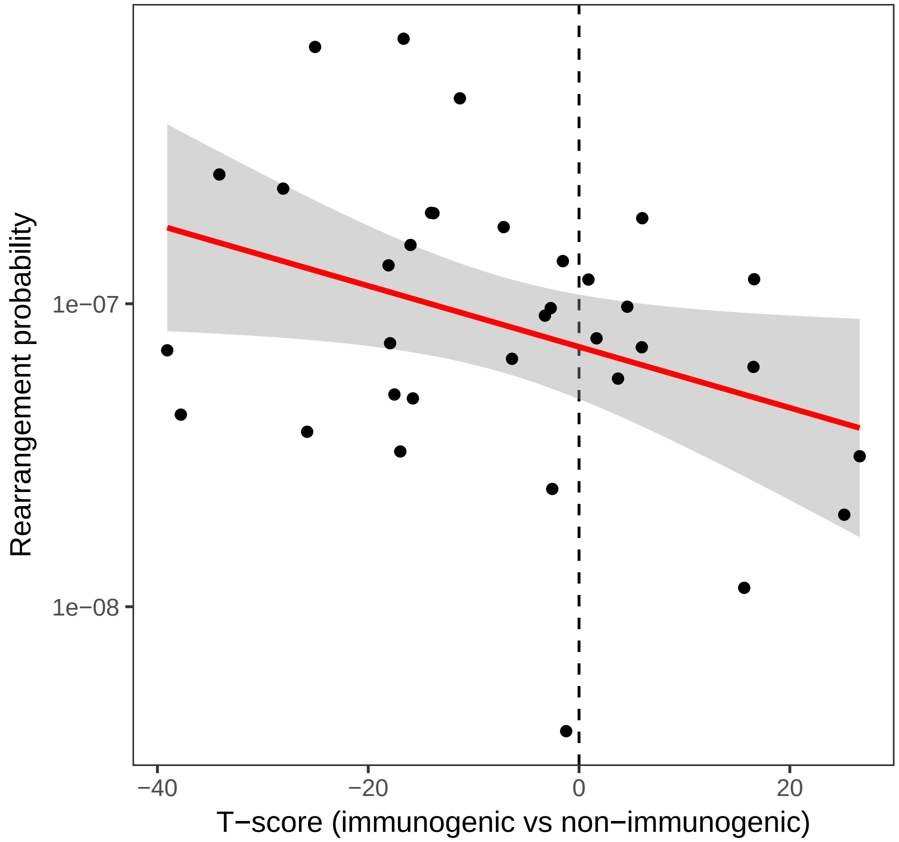
```
mm = as.matrix(dt.imm.prop[,4:13])
rownames(mm) = dt.imm.prop$antigen.epitope
vdjdb_epis = unique((dt.immv %>% filter(!is.na(pGen)))$antigen.epitope)
dd = dist(mm) %>% as.matrix %>% melt %>%
  as.data.table %>%
  filter(Var1 %in% vdjdb_epis | Var2 %in% vdjdb_epis)
```

```
dd2 = dd
tmp = dd$Var1
dd2$Var1 = dd$Var2
dd2$Var2 = tmp
dd = rbind(dd, dd2) %>%
  filter(Var1 %in% vdjdb_epis & !(Var2 %in% vdjdb_epis))
```

```
dt.imm.ann = as.data.table(dd) %>%
  merge(dt.immv %>% mutate(Var1 = antigen.epitope) %>% select(Var1, pGen), by = "Var1") %>%
  merge(dt.immv %>% mutate(Var2 = antigen.epitope) %>% select(Var2, immunogenicity), by = "Var2")

dt.imm.ann = dt.imm.ann %>%
  group_by(Var1, pGen) %>%
  summarise(tscore = t.test(value[which(immunogenicity == "Positive")],
                            value[which(immunogenicity == "Negative")], alternative = "less")$statistic)
```

```r
p22=ggplot(dt.imm.ann,
       aes(x = tscore, y = pGen)) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  geom_smooth(method = "lm", color = "red") +
  geom_point() +
  scale_y_log10("Rearrangement probability") +
  scale_x_continuous("T-score (immunogenic vs non-immunogenic)") +
  theme_bw() +
  theme(aspect.ratio = 1,
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
p22
```



```r
summary(lm(log(pGen) ~ tscore, dt.imm.ann))
```

```
##
## Call:
## lm(formula = log(pGen) ~ tscore, data = dt.imm.ann)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9492 -0.7412  0.1813  0.5999  1.9560
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.44543    0.19491 -84.376   <2e-16 ***
## tscore       -0.02317    0.01079  -2.148   0.0394 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.039 on 32 degrees of freedom
## Multiple R-squared:  0.126,  Adjusted R-squared:  0.09873
## F-statistic: 4.615 on 1 and 32 DF,  p-value: 0.03937
```

SVM-based P-values

```
library(e1071)
#Perform grid-based search to estimate optimal SVM parameters

#svm_params = expand.grid(C = 2^seq(-1,8,by=1),
#                          gamma = 2^seq(1,-6,by=-1))
svm_params = expand.grid(C = 2^seq(-1,8,by=1),
                         gamma = 0.25 * 2^seq(1,-6,by=-1))

svm_train_data = dt.imm.prop %>% filter(!is.na(immunogenicity)) %>%
                select(immunogenicity,f1,f2,f3,f4,f5,f6,f7,f8,f9,f10)
svm_train_data$immunogenicity = as.factor(svm_train_data$immunogenicity)

pred_svm = function(params) {
  svm_mdl = svm(immunogenicity ~ .,
              data= svm_train_data,
              cross = 3,# probability = T,
              cost = params$C, gamma = params$gamma, cachesize = 2000)
  #list(mdl = svm_mdl)
  list(C = params$C,
       gamma = params$gamma,
       acc = svm_mdl$tot.accuracy)
}

#grid_search_res = mclapply(apply(svm_params, 1, as.list),
#                            pred_svm, mc.cores = nrow(svm_params))

dt.grid_search_res = as.data.table(t(matrix(unlist(grid_search_res), nrow = 3)))
colnames(dt.grid_search_res) = c("C", "gamma", "acc")

ggplot(dt.grid_search_res, aes(x = C, y = gamma)) +
  geom_contour(aes(z = acc,colour = ..level..)) +
  scale_x_log10() +
  scale_y_log10() +
  theme_bw()

svm_mdl = svm(as.factor(immunogenicity) ~ .,
              #kernel = "linear",
              cost = 1, gamma = 0.25,
              data=dt.imm.prop %>% filter(!is.na(immunogenicity)) %>%
                select(immunogenicity,f1,f2,f3,f4,f5,f6,f7,f8,f9,f10),
              cross = 5, probability = T)

print(svm_mdl)
summary(svm_mdl)
```

```
#svm_mdl$tot.accuracy

dt.imm.ann.2 = dt.imm.prop %>% filter(!is.na(pGen))
dt.imm.ann.2$immunogenicity = NULL

svm_pred = predict(svm_mdl,
                   newdata = dt.imm.ann.2,
                   probability = T)

dt.imm.ann.2$pImm = attr(svm_pred, "probabilities")[,2]

ggplot(dt.imm.ann.2, aes(x = pImm, y = pGen)) +
  geom_smooth(method = "lm", color = "red") +
  geom_point() +
  scale_y_log10("Rearrangement probability") +
  scale_x_continuous("T-score (immunogenic vs non-immunogenic).") +
  theme_bw() +
  theme(aspect.ratio = 1,
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())

summary(lm(log(pGen) ~ pImm, dt.imm.ann.2))
```

EM-based classifier

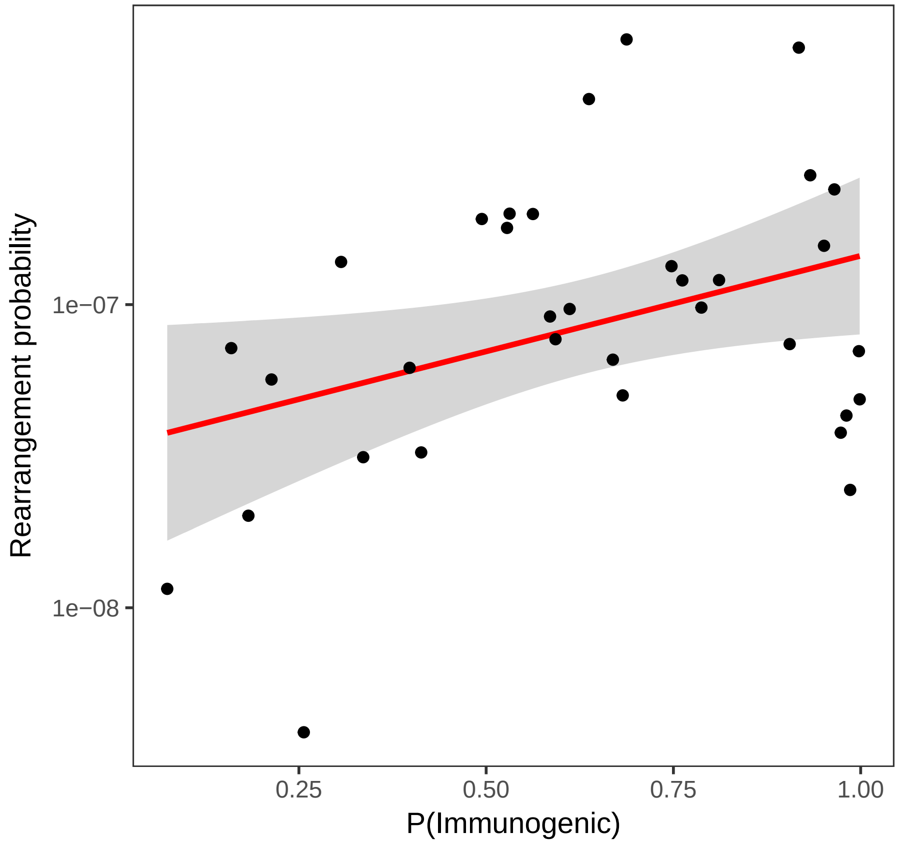
```
x.epi.prop = dt.imm.prop[,4:13]
lab.epi.prob = with(dt.imm.prop, ifelse(immunogenicity == "Positive", 1, 2))
lab.epi.prob = with(dt.imm.prop, ifelse(!is.na(pGen), 0, lab.epi.prob))
res_em = init.EM(as.matrix(x.epi.prop), nclass = 2, lab = lab.epi.prob)

res_probs = e.step(as.matrix(x.epi.prop), res_em, norm = F)
res_probs = exp(as.data.table(res_probs))
colnames(res_probs) = c("Gamma.unnorm.V1", "Gamma.unnorm.V2")

res_probs2 = e.step(as.matrix(x.epi.prop), res_em)
res_probs2 = as.data.table(res_probs2)
```

```
dt.imm.ann.3 = cbind(dt.imm.prop, res_probs, res_probs2) %>%
  filter(!is.na(pGen))
```

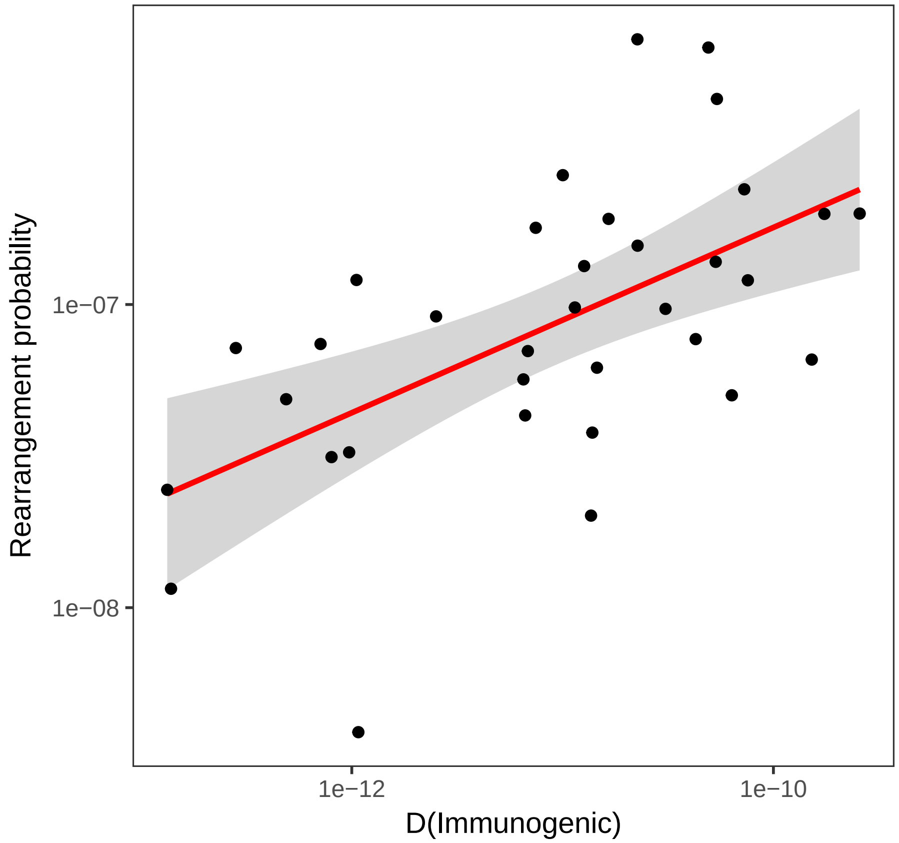
```
p23=ggplot(dt.imm.ann.3, aes(x = Gamma.V1, y = pGen)) +
  #geom_vline(xintercept = 0.5, linetype = "dashed") +
  geom_smooth(method = "lm", color = "red") +
  geom_point() +
  scale_y_log10("Rearrangement probability") +
  scale_x_continuous("P(Immunogenic)") +
  theme_bw() +
  theme(aspect.ratio = 1,
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
p23
```



```
summary(lm(log(pGen) ~ Gamma.V1, dt.imm.ann.3))
```

```
##
## Call:
## lm(formula = log(pGen) ~ Gamma.V1, data = dt.imm.ann.3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5391 -0.6307  0.1159  0.6710  2.0975
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.1999     0.4453 -38.623   <2e-16 ***
## Gamma.V1      1.4524     0.6422   2.262   0.0306 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.032 on 32 degrees of freedom
## Multiple R-squared:  0.1378, Adjusted R-squared:  0.1109
## F-statistic: 5.115 on 1 and 32 DF,  p-value: 0.03065
```

```
p24=ggplot(dt.imm.ann.3, aes(x = Gamma.unnorm.V1, y = pGen)) +
  geom_smooth(method = "lm", color = "red") +
  geom_point() +
  scale_y_log10("Rearrangement probability") +
  scale_x_log10("D(Immunogenic)") +
  theme_bw() +
  theme(aspect.ratio = 1,
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
p24
```



```
summary(lm(log(pGen) ~ log(Gamma.unnorm.V1), dt.imm.ann.3))
```

```
##
## Call:
## lm(formula = log(pGen) ~ log(Gamma.unnorm.V1), data = dt.imm.ann.3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.44795 -0.43689 -0.06781  0.57427  1.88314
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -8.49523    1.95461  -4.346 0.000131 ***
## log(Gamma.unnorm.V1)  0.30565    0.07654   3.993 0.000357 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9078 on 32 degrees of freedom
## Multiple R-squared:  0.3326, Adjusted R-squared:  0.3117
## F-statistic: 15.95 on 1 and 32 DF,  p-value: 0.000357
```

Predicting precursor frequency

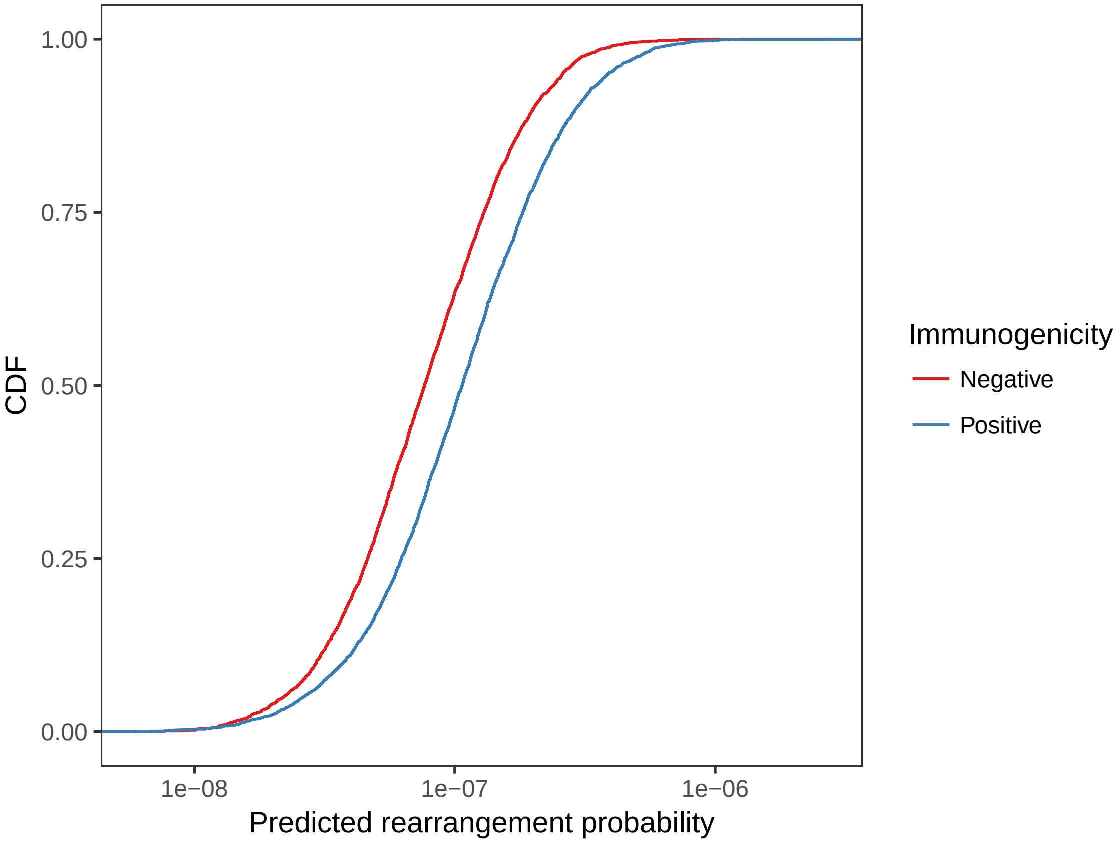
```
mdl_p = lm(log(pGen) ~ f1 + f2 + f3 + f4 + f5 + f6 + f7 + f8 + f9 + f10, #len + f6 + f10,
           dt.imm.prop %>% filter(!is.na(pGen)))
summary(mdl_p)
```

```
##
## Call:
## lm(formula = log(pGen) ~ f1 + f2 + f3 + f4 + f5 + f6 + f7 + f8 +
##     f9 + f10, data = dt.imm.prop %>% filter(!is.na(pGen)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1112 -0.5156  0.0132  0.6243  1.5005
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.59826    0.40271 -41.216   <2e-16 ***
## f1           -0.01460    0.07365  -0.198   0.8446
## f2           -0.04196    0.06720  -0.624   0.5385
## f3            0.03184    0.08573   0.371   0.7138
## f4           -0.06440    0.07255  -0.888   0.3839
## f5           -0.10783    0.07655  -1.409   0.1723
## f6           -0.05863    0.08653  -0.678   0.5048
## f7            0.02425    0.06734   0.360   0.7221
## f8           -0.07880    0.07946  -0.992   0.3316
## f9            0.14229    0.11640   1.222   0.2339
## f10          -0.13685    0.06720  -2.036   0.0534 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.009 on 23 degrees of freedom
## Multiple R-squared:  0.4078, Adjusted R-squared:  0.1504
## F-statistic: 1.584 on 10 and 23 DF,  p-value: 0.174
```

```
pred_p = predict(mdl_p, dt.imm.prop)
```

```
dt.pred_p = dt.imm.prop
dt.pred_p$pGenPred = pred_p
```

```
p25=ggplot(dt.pred_p %>% filter(!is.na(immunogenicity)), aes(x = exp(pGenPred),
                                                              color = immunogenicity)) +
  stat_ecdf() +
  ylab("CDF") +
  scale_x_log10("Predicted rearrangement probability") +
  scale_color_brewer("Immunogenicity", palette = "Set1") +
  theme_bw() +
  theme(aspect = 1,
        panel.grid.major = element_blank(), panel.grid.minor = element_blank())
p25
```



```
ks.test((dt.pred_p %>% filter(!is.na(immunogenicity) & immunogenicity == "Positive"))$pGenPred,
        (dt.pred_p %>% filter(!is.na(immunogenicity) & immunogenicity == "Negative"))$pGenPred)
```

```
## Warning in ks.test((dt.pred_p %>% filter(!is.na(immunogenicity) &
## immunogenicity == : p-value will be approximate in the presence of ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  (dt.pred_p %>% filter(!is.na(immunogenicity) & immunogenicity ==  and (dt.pred_p %>% filter(!
## D = 0.16864, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
ggsave("p20.pdf", p20)
```

```
## Saving 6.5 x 4.5 in image

## Warning: Removed 56 rows containing non-finite values (stat_density2d).

## Warning: Removed 56 rows containing non-finite values (stat_density2d).
```

```
ggsave("p21.pdf", p21)
```

```
## Saving 6.5 x 4.5 in image

## Warning: Removed 50 rows containing non-finite values (stat_density2d).
```

```
ggsave("p22.pdf", p22)
```

```
## Saving 6.5 x 4.5 in image
```

```r
ggsave("p23.pdf", p23)
```

```
## Saving 6.5 x 4.5 in image
```

```r
ggsave("p24.pdf", p24)
```

```
## Saving 6.5 x 4.5 in image
```

```r
ggsave("p25.pdf", p25)
```

```
## Saving 6.5 x 4.5 in image
```