BNLEARN

Load data and filter

```
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
```

```
library(RColorBrewer)
library(gplots)
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

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


df <- read.table("structure.txt", header = T, sep="\t") %>%
  filter(mhc_type == "MHCI") %>%
  mutate(tcr_chain = as.factor(substr(as.character(tcr_v_allele), 1, 3)),
         pos_tcr = as.numeric(pos_tcr),
         len_tcr = as.numeric(len_tcr),
         pos_antigen = as.numeric(pos_antigen),
         len_antigen = as.numeric(len_antigen)) %>%
  select(tcr_region, tcr_chain, pos_tcr, len_tcr, aa_tcr, pos_antigen, len_antigen, aa_antigen, energy)
  mutate(contact = as.factor(energy < 0),
         pos_rel_tcr = cut(pos_tcr / (len_tcr - 1), 10),
         pos_rel_antigen = cut(pos_antigen / (len_antigen - 1), 10)) %>%
  select(tcr_region, tcr_chain, pos_rel_tcr, aa_tcr, pos_rel_antigen, aa_antigen, contact)

df$contact[is.na(df$contact)] <- "FALSE"

head(df)
```

```
##   tcr_region tcr_chain  pos_rel_tcr aa_tcr pos_rel_antigen aa_antigen
## 1       CDR1       TRA (-0.001,0.1]      D    (-0.001,0.1]          L
## 2       CDR1       TRA (-0.001,0.1]      D       (0.1,0.2]          L
## 3       CDR1       TRA (-0.001,0.1]      D       (0.2,0.3]          F
## 4       CDR1       TRA (-0.001,0.1]      D       (0.3,0.4]          G
## 5       CDR1       TRA (-0.001,0.1]      D       (0.4,0.5]          Y
## 6       CDR1       TRA (-0.001,0.1]      D       (0.6,0.7]          P
##   contact
## 1    TRUE
## 2   FALSE
## 3   FALSE
## 4   FALSE
## 5   FALSE
## 6   FALSE
```

```
summary(df)
```

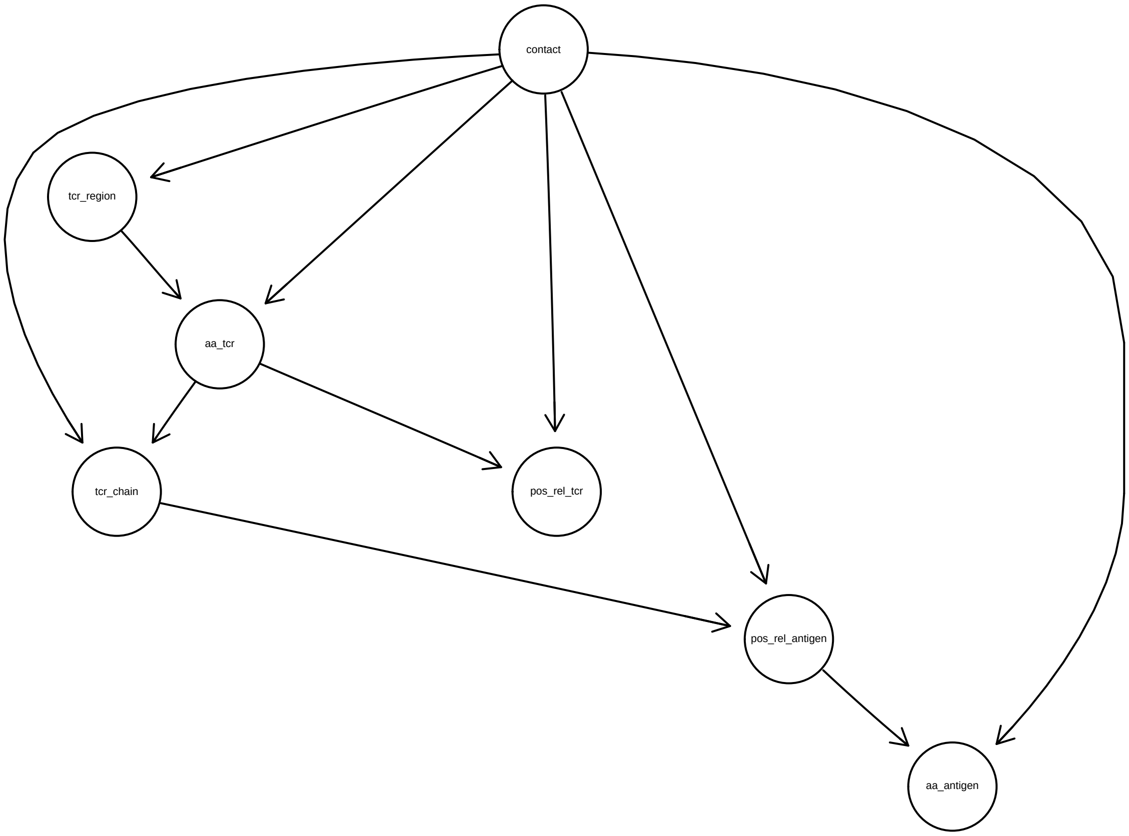
```
##  tcr_region   tcr_chain         pos_rel_tcr       aa_tcr
##  CDR1: 9111   TRA:17301   (-0.001,0.1]:5769   S      : 4663
##  CDR2: 6040   TRB:17293   (0.9,1]     :5593   G      : 3981
##  CDR3:19443               (0.4,0.5]   :4303   A      : 2443
##                           (0.3,0.4]   :3259   F      : 2367
##                           (0.1,0.2]   :3192   Y      : 2309
##                           (0.7,0.8]   :3059   T      : 2103
##                           (Other)     :9419   (Other):16728
##      pos_rel_antigen    aa_antigen     contact
##  (-0.001,0.1]: 4220   L      : 5499   FALSE:29819
##  (0.4,0.5]   : 3862   G      : 3468   TRUE : 4775
##  (0.9,1]     : 3862   V      : 2893
##  (0.1,0.2]   : 3627   Y      : 2830
##  (0.2,0.3]   : 3627   F      : 2420
##  (0.7,0.8]   : 3627   A      : 2381
##  (Other)     :11769   (Other):15103
```

Inferred model

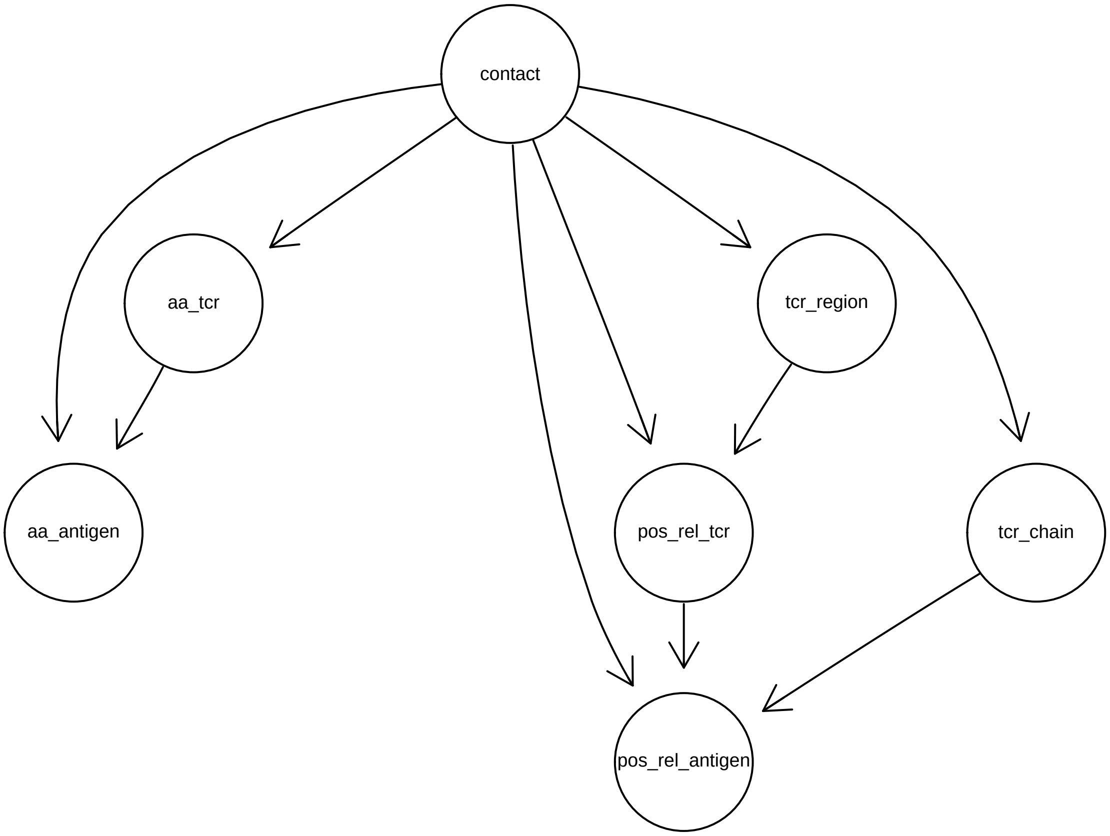
```
tb <- tree.bayes(df, training = "contact")

graphviz.plot(tb)
```

```
## Loading required namespace: Rgraphviz
```

```
## Note: the specification for S3 class "AsIs" in package 'BiocGenerics' seems equivalent to one from pa
```



```
emp_net <- model2network(paste(
  "[contact]",
  "[tcr_chain|contact]",
  "[tcr_region|contact]",
  "[pos_rel_antigen|pos_rel_tcr:tcr_chain:contact]",
  "[aa_antigen|aa_tcr:contact]",
  "[pos_rel_tcr|tcr_region:contact]",
  "[aa_tcr|contact]",
  sep =""))

graphviz.plot(emp_net)
```



```r
fit <- bn.fit(emp_net, df, method="bayes")

BIC(fit, df)
```

```
## [1] -419457.2
```

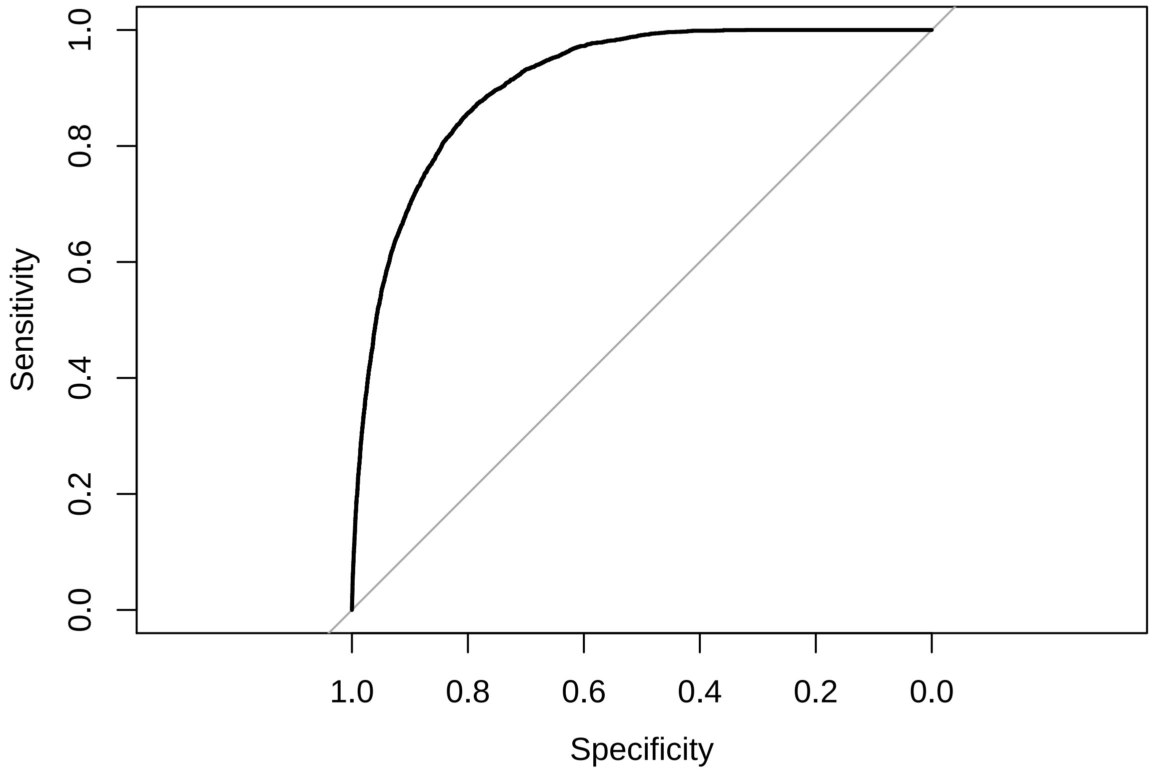
```r
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
res <- predict(fit, node="contact", method="bayes-lw", data=df, prob=T)

p <- attributes(res)$prob

rocobj <- plot.roc(df[,"contact"], p[1,], ci=T)
```



```
rocobj
```

```
##
## Call:
## plot.roc.default(x = df[, "contact"], predictor = p[1, ], ci = T)
##
## Data: p[1, ] in 29819 controls (df[, "contact"] FALSE) > 4775 cases (df[, "contact"] TRUE).
## Area under the curve: 0.9105
## 95% CI: 0.9068-0.9142 (DeLong)
```

```
# df.cplx <- df %>% select(pdb_id, tcr_chain, contact)
# df.cplx$p <- p[1,]
#
# df.cplx <- df.cplx %>%
#   group_by(pdb_id, tcr_chain) %>%
#   summarise(contacts = sum(as.logical(contact)),
#             contacts.pred = sum(p))
#
# ggplot(df.cplx, aes(contacts, contacts.pred)) +
#   geom_point(shape=21) +
#   geom_abline(slope = 1, intercept = 0) +
#   scale_x_continuous(limits=c(0,200)) +
#   scale_y_continuous(limits=c(0,200)) +
#   theme_bw()
```

```
get_prob <- function(var_name) {
  .df <- as.data.frame(fit[[var_name]]$prob)
  colnames(.df) <- gsub("Var1", "contact", colnames(.df))
  colnames(.df) <- gsub("Freq", paste("Freq", var_name, sep="."), colnames(.df))
  .df
```

```r
}

prob.tmp <- get_prob("contact")

for (var in colnames(df)[!(colnames(df) %in% c("contact", "pdb_id"))]) {
  prob.tmp <- merge(prob.tmp, get_prob(var))
}

prob.tmp$contact <- as.logical(prob.tmp$contact)

prob.tmp$P <- apply(prob.tmp[,which(grepl("Freq",colnames(prob.tmp)))], 1,
                function(x) prod(x))
```

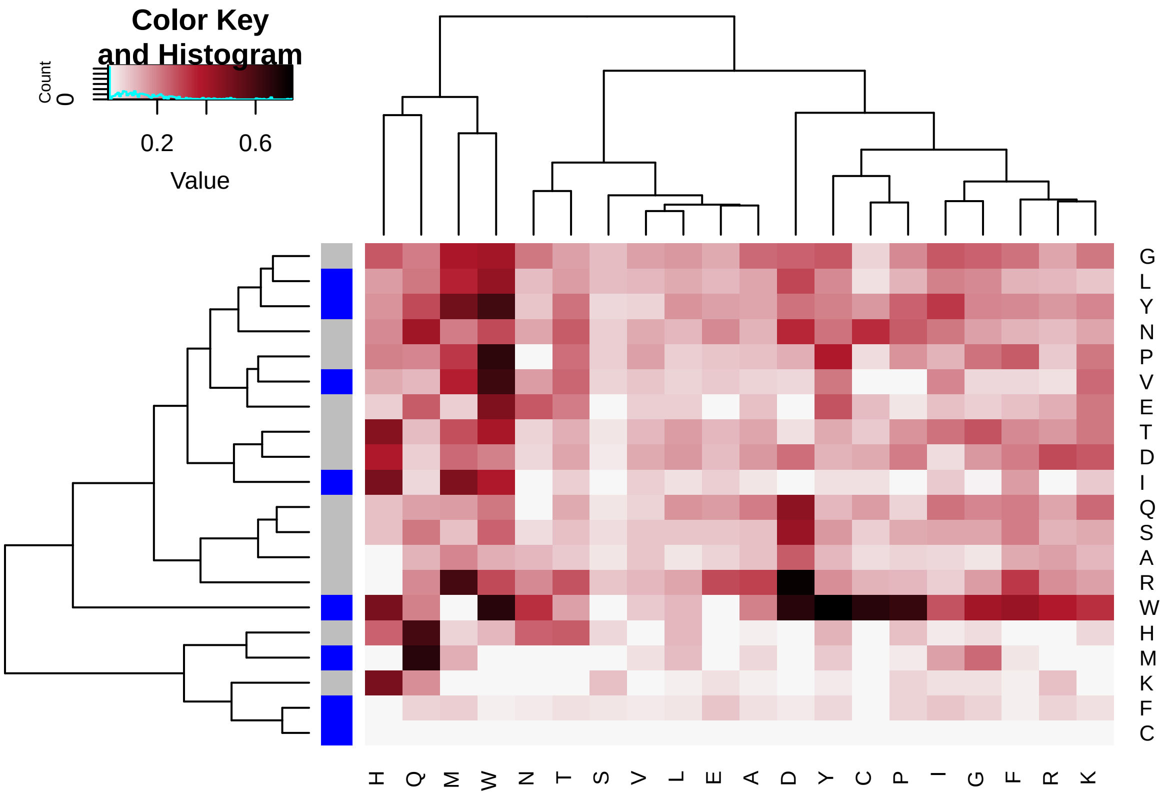
```r
prob.aTaAC <- prob.tmp %>%
  group_by(aa_tcr, aa_antigen, contact) %>%
  summarise(P = sum(P)) %>%
  group_by(aa_tcr, aa_antigen) %>%
  summarise(P = P[which(contact)] / sum(P))
```

```r
aa_pair_mat <- dcast(prob.aTaAC, aa_tcr ~ aa_antigen)
```

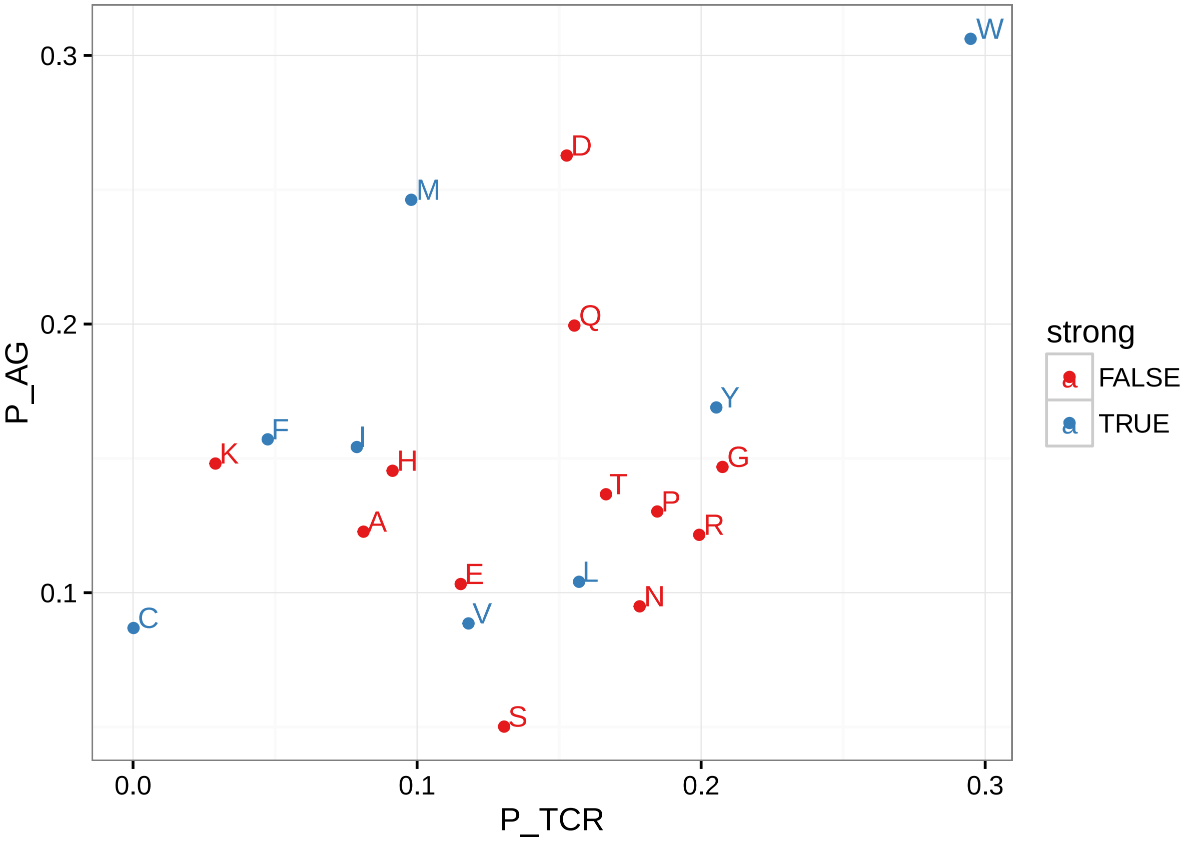
```
## Using P as value column: use value.var to override.
```

```r
rownames(aa_pair_mat) <- aa_pair_mat$aa_tcr
aa_pair_mat$aa_tcr <- NULL
aa_pair_mat <- as.matrix(aa_pair_mat)
aa_pair_mat[is.na(aa_pair_mat)] <- 0

strong_aa <- c("F", "I", "L", "M", "V", "W", "Y", "C")
strong_col <- ifelse(rownames(aa_pair_mat) %in% strong_aa, "blue", "grey")
names(strong_col) <- rownames(aa_pair_mat)

js_calc <- function(p, q) {
  p<-p/sum(p)
  q<-q/sum(q)
  m <- 0.5 * (p + q)
  0.5 * (sum(p * log(p / m)) + sum(q * log(q / m)))
}

js_dist <- function(x) {
    mat <- x
    for(i in 1:nrow(mat)) {
        for(j in 1:nrow(mat)) {
            mat[i, j] <- js_calc(x[i, ], x[j, ])
    }}
    return(as.dist(mat))
}

heatmap.2(aa_pair_mat,
          hclustfun = function(x) hclust(x, method = "ward"),
          #distfun = function(x) js_dist(x),
          RowSideColors = strong_col,
          #ColSideColors = strong_col,
          trace = "none",
          #breaks = seq(-4, 0, length.out = 101),
          col=colorpanel(100, "#f7f7f7", "#b2182b", "black"))
```

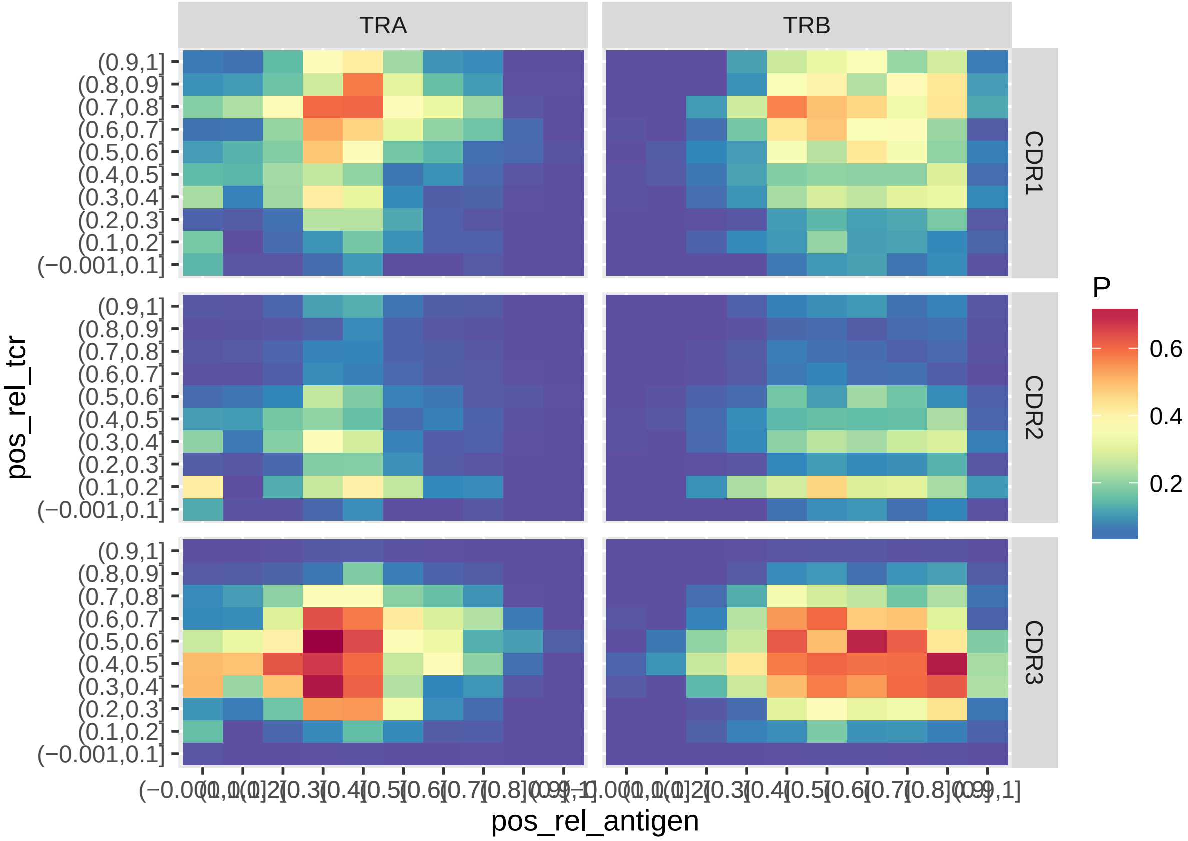
```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```



```
df.1 <- prob.tmp %>%
  group_by(aa_tcr, contact) %>%
  summarise(P = sum(P)) %>%
  group_by(aa_tcr) %>%
  summarise(P = P[which(contact)] / sum(P))

df.2 <- prob.tmp %>%
  group_by(aa_antigen, contact) %>%
  summarise(P = sum(P)) %>%
  group_by(aa_antigen) %>%
  summarise(P = P[which(contact)] / sum(P))

colnames(df.1) <- c("aa", "P_TCR")
colnames(df.2) <- c("aa", "P_AG")

df.1 <- merge(df.1, df.2)

df.1$strong <- ifelse(df.1$aa %in% strong_aa, T, F)

ggplot(df.1, aes(x=P_TCR, y=P_AG, color=strong)) +
  geom_point() +
  geom_text(aes(label=aa), vjust=0, hjust=-0.2) +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```



```
rf <- colorRampPalette(rev(brewer.pal(11, 'Spectral')))
r <- rf(32)

df.1 <- prob.tmp %>%
  group_by(pos_rel_antigen, pos_rel_tcr, tcr_chain, tcr_region, contact) %>%
  summarise(P = sum(P)) %>%
  group_by(pos_rel_antigen, pos_rel_tcr, tcr_chain, tcr_region) %>%
  summarise(P = P[which(contact)] / sum(P))

ggplot(df.1, aes(x=pos_rel_antigen, y = pos_rel_tcr, fill=P)) +
  geom_tile() +
  scale_fill_gradientn(colors=r) +
  facet_grid(tcr_region~tcr_chain)
```


TRA
TRB
CDR1
CDR2
CDR3
pos_rel_tcr
(0.9,1]
(0.8,0.9]
(0.7,0.8]
(0.6,0.7]
(0.5,0.6]
(0.4,0.5]
(0.3,0.4]
(0.2,0.3]
(0.1,0.2]
(−0.001,0.1]
pos_rel_antigen
(−0.001,0.1]
(0.1,0.2]
(0.2,0.3]
(0.3,0.4]
(0.4,0.5]
(0.5,0.6]
(0.6,0.7]
(0.7,0.8]
(0.8,0.9]
(0.9,1]
P
0.6
0.4
0.2