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") %>%
  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(pdb_id, tcr_region, tcr_chain, pos_tcr, len_tcr, aa_tcr, pos_antigen, len_antigen, aa_antigen,
  mutate(contact = as.factor(distance <= 6),
         pos_rel_tcr = cut(pos_tcr / (len_tcr - 1), 5),
         pos_rel_antigen = cut(pos_antigen / (len_antigen - 1), 5))

# Filter no contact complexes

df.noc <- df %>%
  group_by(pdb_id, tcr_chain) %>%
  summarise(contacts = sum(as.logical(contact))) %>%
  filter(contacts < 3)

df <- df %>%
  filter(!(pdb_id %in% df.noc$pdb_id))

pdb_id <- df$pdb_id

df <- df %>% select(tcr_region, tcr_chain, pos_rel_tcr, aa_tcr, pos_rel_antigen, aa_antigen, mhc_type,
```

```r
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 (-0.001,0.2]      D    (-0.001,0.2]          L
## 2       CDR1       TRA (-0.001,0.2]      D    (-0.001,0.2]          L
## 3       CDR1       TRA (-0.001,0.2]      D       (0.2,0.4]          F
## 4       CDR1       TRA (-0.001,0.2]      D       (0.2,0.4]          G
## 5       CDR1       TRA (-0.001,0.2]      D       (0.4,0.6]          Y
## 6       CDR1       TRA (-0.001,0.2]      D       (0.6,0.8]          P
##   mhc_type contact
## 1     MHCI   FALSE
## 2     MHCI   FALSE
## 3     MHCI   FALSE
## 4     MHCI   FALSE
## 5     MHCI   FALSE
## 6     MHCI   FALSE
```

```r
summary(df)
```

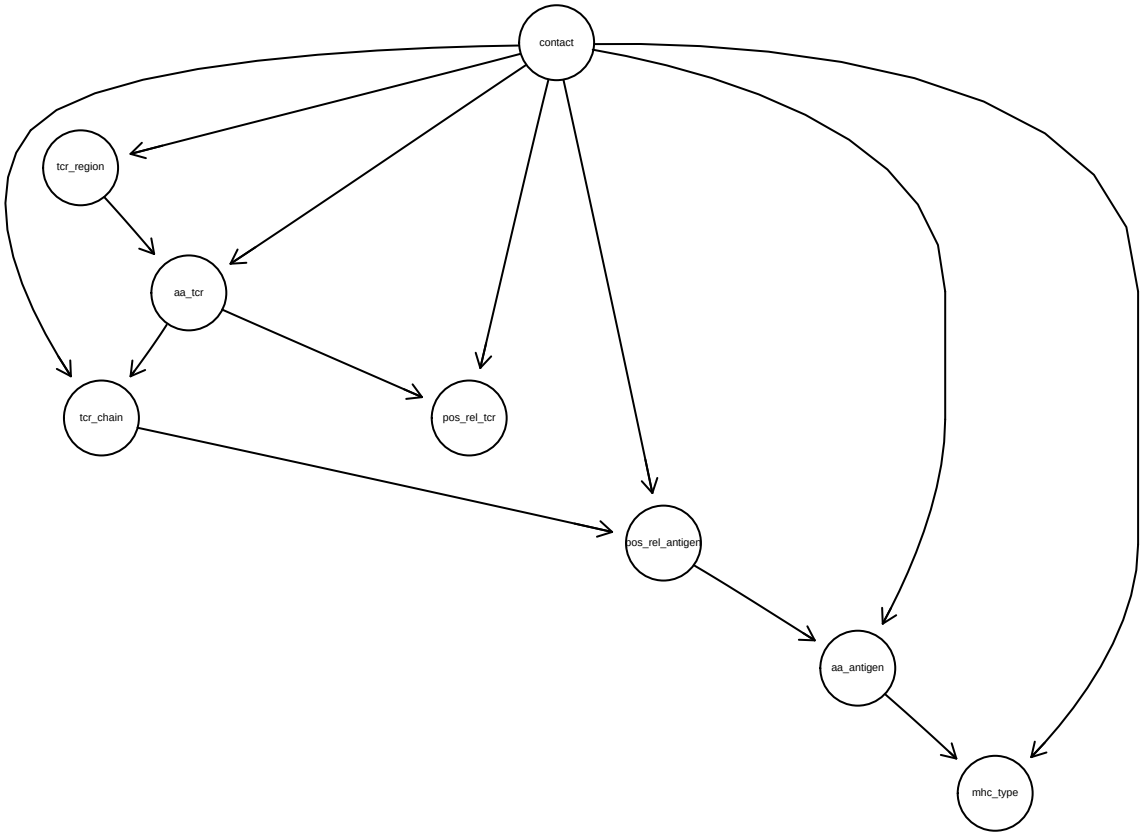
```
##  tcr_region   tcr_chain         pos_rel_tcr        aa_tcr
##  CDR1:11904   TRA:22432   (-0.001,0.2]:11585   S      : 6000
##  CDR2: 7451   TRB:22793   (0.2,0.4]   : 7187   G      : 5065
##  CDR3:25870               (0.4,0.6]   : 8839   A      : 3613
##                           (0.6,0.8]   : 7187   F      : 3033
##                           (0.8,1]     :10427   T      : 2987
##                                                N      : 2953
##                                                (Other):21574
##      pos_rel_antigen    aa_antigen     mhc_type      contact
##  (-0.001,0.2]:10257   L      : 6008   MHCI :31685   FALSE:42074
##  (0.2,0.4]   : 8648   G      : 4972   MHCII:13540   TRUE : 3151
##  (0.4,0.6]   : 7972   P      : 3600
##  (0.6,0.8]   : 8648   A      : 3374
##  (0.8,1]     : 9700   Y      : 3265
##                       V      : 2951
##                       (Other):21055
```

Inferred model

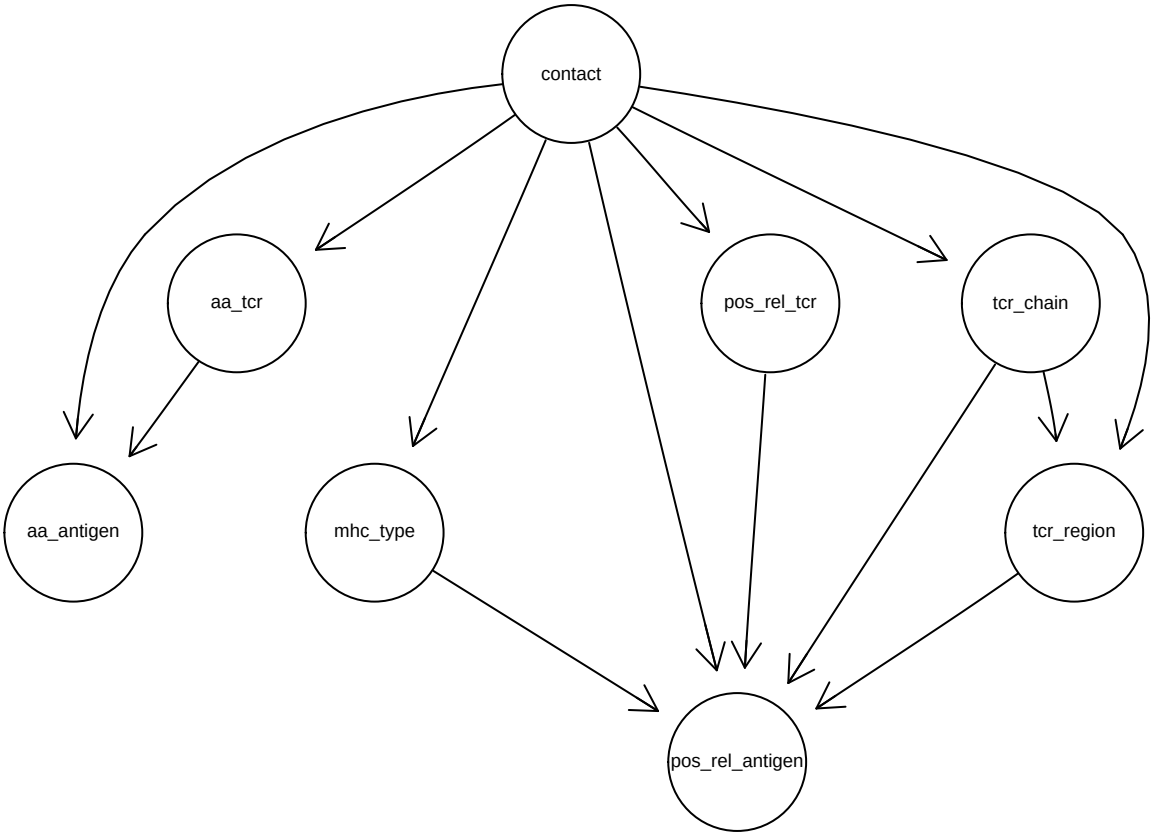
```r
tb <- tree.bayes(df, training = "contact")

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]",
  "[mhc_type|contact]",
  "[tcr_region|tcr_chain:contact]",
  "[pos_rel_antigen|pos_rel_tcr:tcr_region:mhc_type:tcr_chain:contact]",
  "[aa_antigen|aa_tcr:contact]",
  "[pos_rel_tcr|contact]",
  "[aa_tcr|contact]",
  sep =""))

graphviz.plot(emp_net)
```




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

BIC(fit, df)
```

```
## [1] -514320.1
```

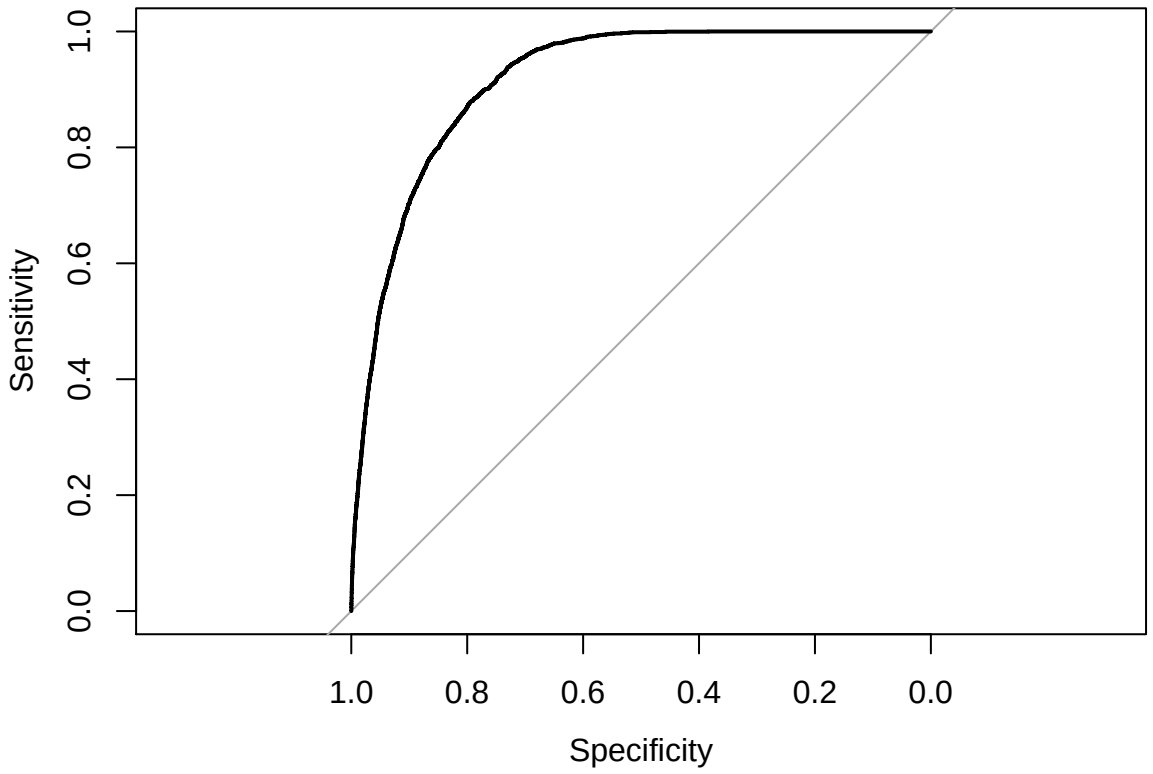
```
library(pROC)
```

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

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

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

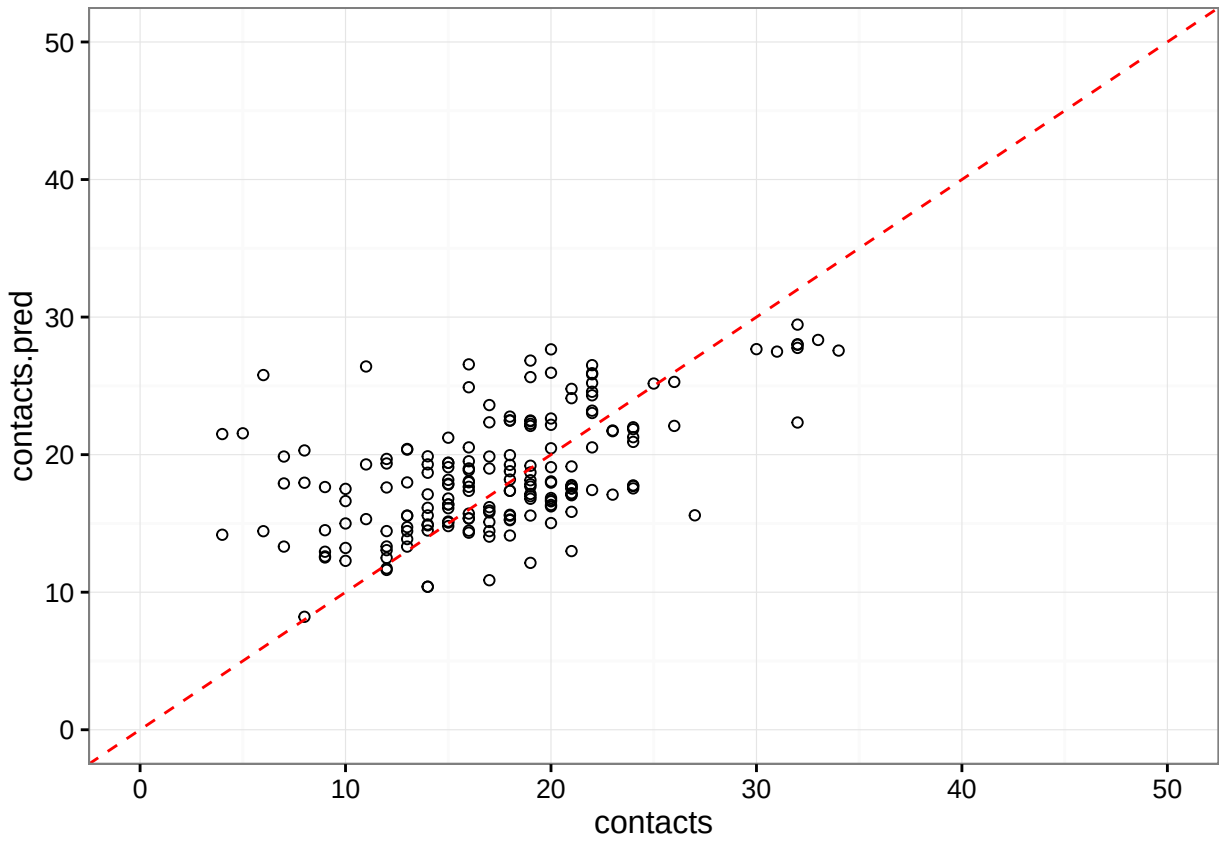
```
res <- predict(fit, node="contact", method="bayes-lw", data=df, prob=T)

p <- attributes(res)$prob

rocobj <- plot.roc(df[,"contact"], p[2,], ci=T)
```




```
rocobj
```

```
##
## Call:
## plot.roc.default(x = df[, "contact"], predictor = p[2, ], ci = T)
##
## Data: p[2, ] in 42074 controls (df[, "contact"] FALSE) < 3151 cases (df[, "contact"] TRUE).
## Area under the curve: 0.9156
## 95% CI: 0.9119-0.9193 (DeLong)
```

```
df.cplx <- data.frame(pdb_id = pdb_id)
df.cplx$tcr_chain <- df$tcr_chain
df.cplx$contact <- as.logical(df$contact)
df.cplx$p <- p[2,]

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, color = "red", linetype="dashed") +
   scale_x_continuous(limits=c(0, 50)) +
   scale_y_continuous(limits=c(0, 50)) +
   theme_bw()
```

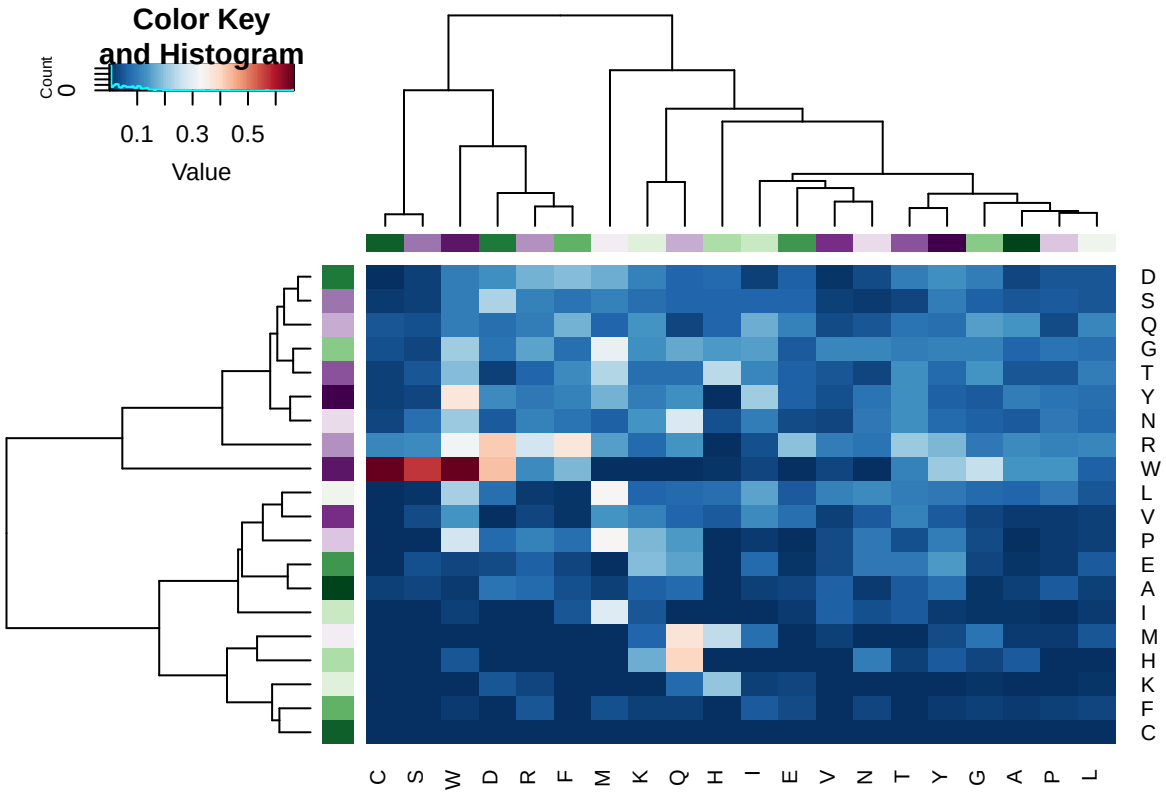



```
get_prob <- function(var_name) {
  .df <- as.data.frame(fit[[var_name]]$prob)
  colnames(.df) <- gsub("Var1", "contact", colnames(.df))
  colnames(.df) <- gsub("Freq", paste("Freq", var_name, sep="."), colnames(.df))
  .df
}

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))
```

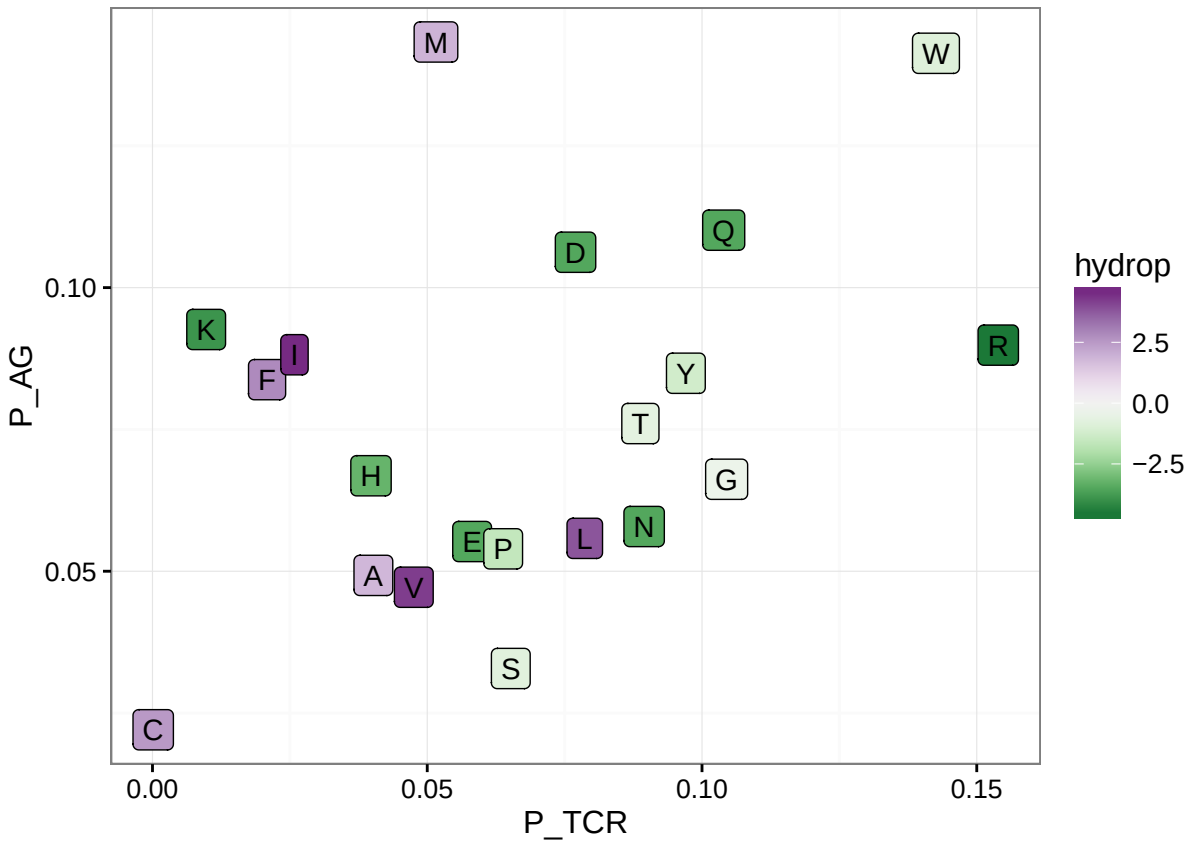
```
prob.aTaAC <- prob.tmp %>%
  group_by(aa_tcr, aa_antigen, contact) %>%
  summarise(P = sum(P)) %>%
  group_by(aa_tcr, aa_antigen) %>%
  summarise(P = P[which(contact)] / sum(P))
```

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

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

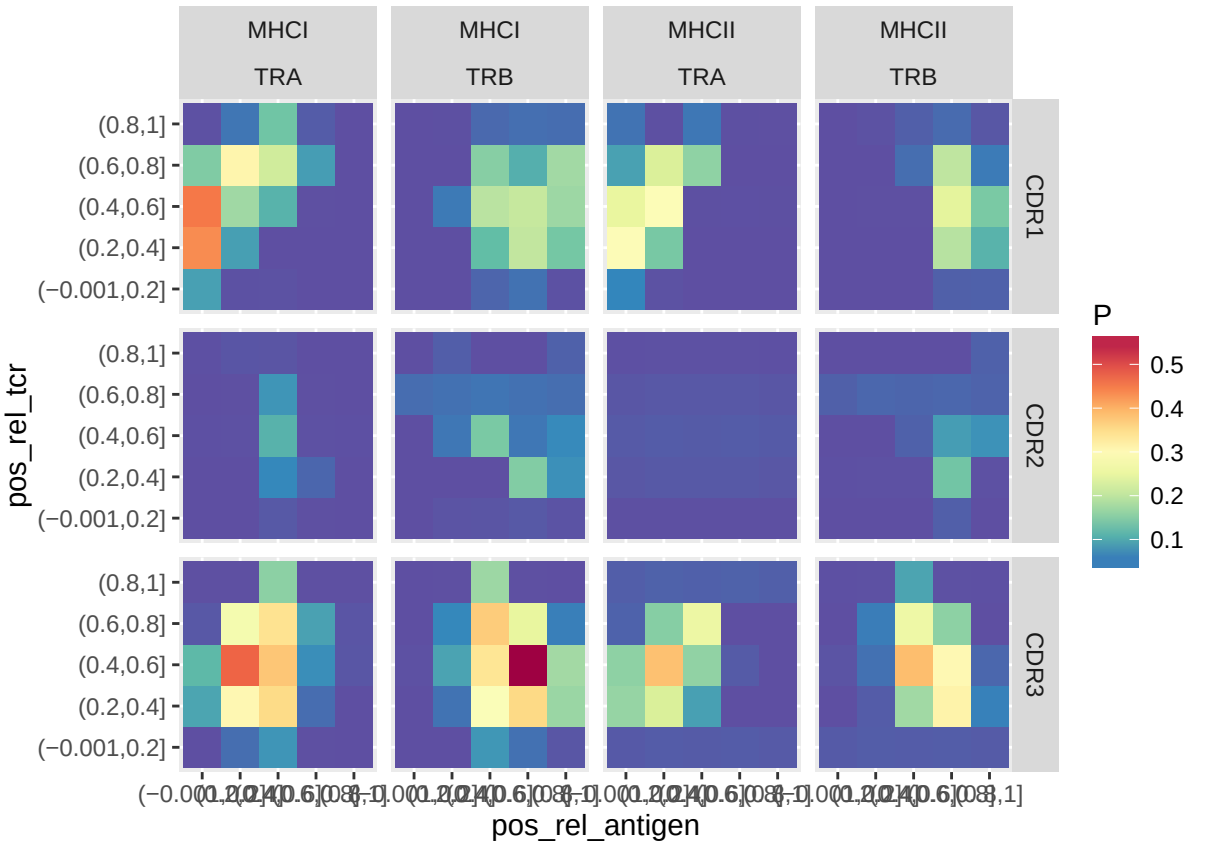
```
rownames(aa_pair_mat) <- aa_pair_mat$aa_tcr
aa_pair_mat$aa_tcr <- NULL
aa_pair_mat <- as.matrix(aa_pair_mat)
aa_pair_mat[is.na(aa_pair_mat)] <- 0

df.hydro <- data.frame(
  aa = strsplit("I V L F C M A W G T S Y P H N D Q E K R", " ")[[1]],
  hydrop = strsplit("4.5 4.2 3.8 2.8 2.5 1.9 1.8 -0.9 -0.4 -0.7 -0.8 -1.3 -1.6 -3.2 -3.5 -3.5 -3.5 -3.5
)

df.hydro <- df.hydro %>%
  mutate(hydrop = as.numeric(as.character(hydrop))) %>%
  arrange(hydrop)

df.hydro$color <- colorRampPalette(rev(brewer.pal(11, 'PRGn')))(20)

aa_colors <- df.hydro$color
names(aa_colors) <- df.hydro$aa

js_calc <- function(p, 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 = aa_colors,
          ColSideColors = aa_colors,
          trace = "none",
          #breaks = seq(0, 0.2, length.out = 101),
          col=colorRampPalette(rev(brewer.pal(11, 'RdBu')))(100))
```



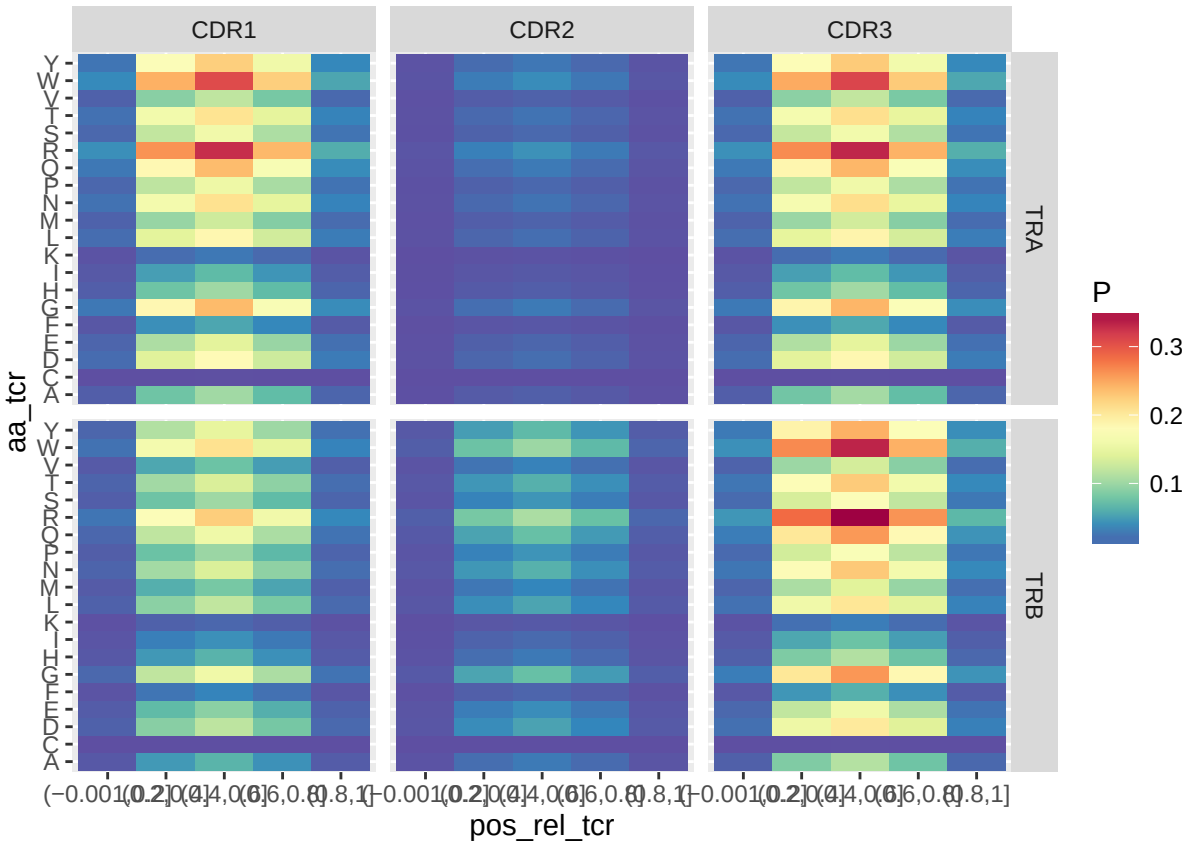

```
df.1 <- prob.tmp %>%
  group_by(aa_tcr, contact) %>%
  summarise(P = sum(P)) %>%
  group_by(aa_tcr) %>%
  summarise(P = P[which(contact)] / sum(P))

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

ggplot(df.1, aes(x=P_TCR, y=P_AG, fill=hydrop)) +
  geom_label(aes(label=aa)) +
  scale_fill_gradientn(colors = colorRampPalette(rev(brewer.pal(9, 'PRGn')))(20)) +
  #scale_color_gradientn(colors = df.1$color) +
  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, mhc_type, contact) %>%
  summarise(P = sum(P)) %>%
  group_by(pos_rel_antigen, pos_rel_tcr, tcr_chain, tcr_region, mhc_type) %>%
  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~mhc_type+tcr_chain)
```




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

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


CDR1
CDR2
CDR3
TRA
TRB
aa_tcr
pos_rel_tcr
P
0.3
0.2
0.1