A naive TCR:pMHC interaction model

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

```
## Warning: package 'data.table' was built under R version 3.2.5
```

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

```
## Warning: package 'reshape2' was built under R version 3.2.5
```

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

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

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.2.5
```

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

```
## 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(ggbeeswarm)
library(RColorBrewer)
library(pROC)
```

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

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

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

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.2.5
```

```r
library(gplots)
```

```
## Warning: package 'gplots' was built under R version 3.2.4
```

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

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

```r
df = fread("../result/structure.txt", header=T, sep="\t")[tcr_region %in% c("CDR1", "CDR2", "CDR3")]

df$tcr_chain = as.factor(substr(as.character(df$tcr_v_allele), 1, 3))
df$contact = df$distance <= 4.5
df$aa_pair = with(df,
  as.factor(ifelse(as.character(aa_tcr) < as.character(aa_antigen), paste(aa_tcr, aa_antigen, sep = "_")

summary(df)
```

```
##      pdb_id             species            mhc_type
##  Length:62077       Length:62077       Length:62077
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  mhc_a_allele       mhc_b_allele       antigen_seq
##  Length:62077       Length:62077       Length:62077
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  tcr_v_allele       tcr_region         tcr_region_seq
##  Length:62077       Length:62077       Length:62077
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     aa_tcr            aa_antigen           len_tcr         len_antigen
##  Length:62077       Length:62077       Min.   : 3.00    Min.   : 8.00
##  Class :character   Class :character   1st Qu.: 6.00    1st Qu.: 9.00
##  Mode  :character   Mode  :character   Median :12.00    Median :10.00
##                                        Mean   :10.33    Mean   :10.94
##                                        3rd Qu.:14.00    3rd Qu.:13.00
##                                        Max.   :18.00    Max.   :20.00
##
##     pos_tcr          pos_antigen         distance         distance_CA
##  Min.   : 0.000    Min.   : 0.000    Min.   : 2.231    Min.   : 3.696
##  1st Qu.: 1.000    1st Qu.: 2.000    1st Qu.: 10.622   1st Qu.: 13.907
```

```
##   Median : 4.000   Median : 5.000   Median : 15.135   Median : 18.415
##   Mean   : 4.663   Mean    : 4.971   Mean    : 17.691   Mean     : 20.927
##   3rd Qu.: 7.000   3rd Qu.: 7.000   3rd Qu.: 20.351   3rd Qu.: 23.483
##   Max.   :17.000   Max.    :19.000   Max.    :126.207   Max.     :129.029
##
##    distance_CB         energy         tcr_chain    contact
##   Min.    :  2.164   Min.    :-76.1000   TRA:30674   Mode :logical
##   1st Qu.: 13.952   1st Qu.:  0.0000   TRB:31403   FALSE:59961
##   Median : 18.757   Median :  0.0000               TRUE :2116
##   Mean    : 21.205   Mean    : -0.2774               NA's :0
##   3rd Qu.: 24.256   3rd Qu.:  0.0000
##   Max.    :132.255   Max.    :774.0000
##                     NA's    :160
##     aa_pair
##   L_S     : 1232
##   G_S     : 1200
##   G_L     : 1146
##   A_G     : 1047
##   A_S     :  902
##   G_Y     :  869
##   (Other):55681
```

Some EDA

Contact distribution

Contacts by MHC, chain and CDR

```
df.contact.sum = df[,.(contacts = sum(contact)),by=.(pdb_id, tcr_chain, tcr_region, mhc_type)]

df.contact.sum.pdb = df.contact.sum[,.(contacts = sum(contacts)), by=.(pdb_id, mhc_type)][contacts>5]

ggplot(df.contact.sum.pdb, aes(contacts, color = mhc_type)) +
  geom_freqpoly() +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```
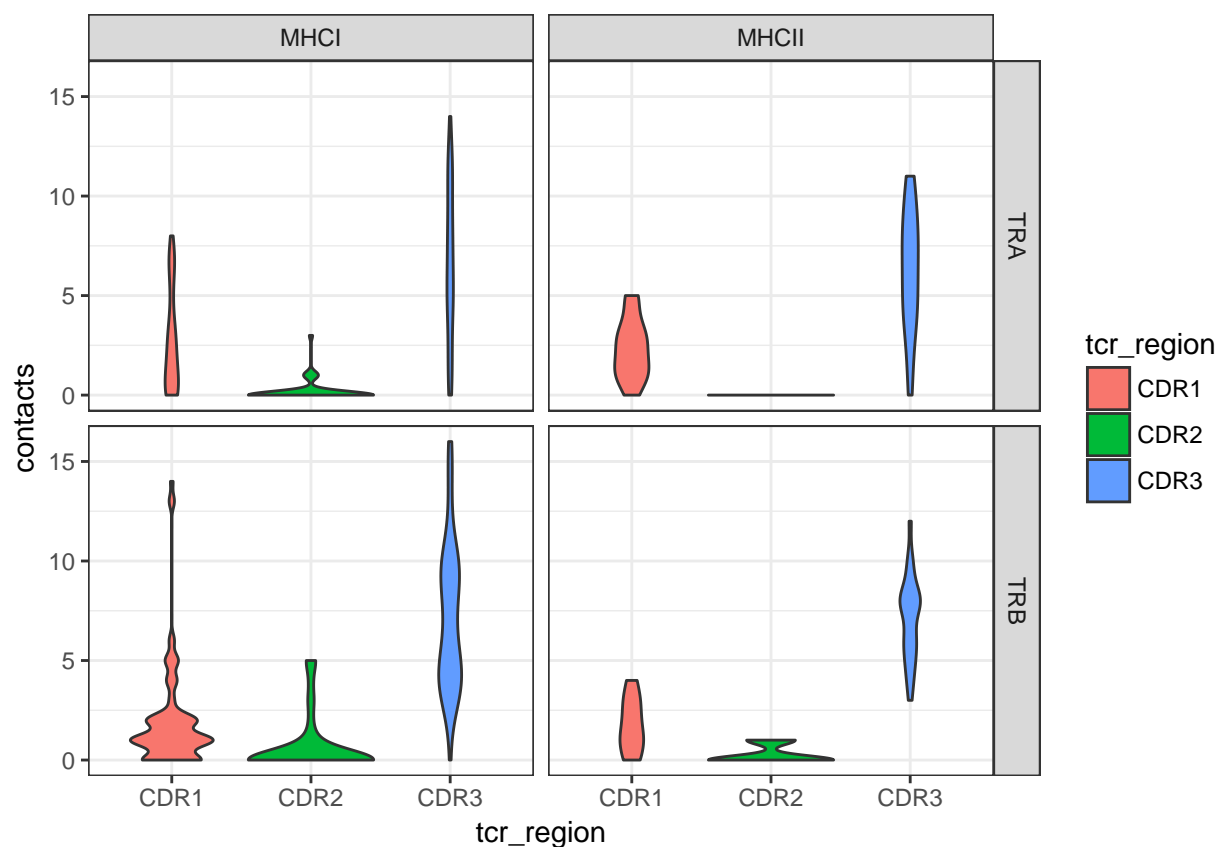
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```
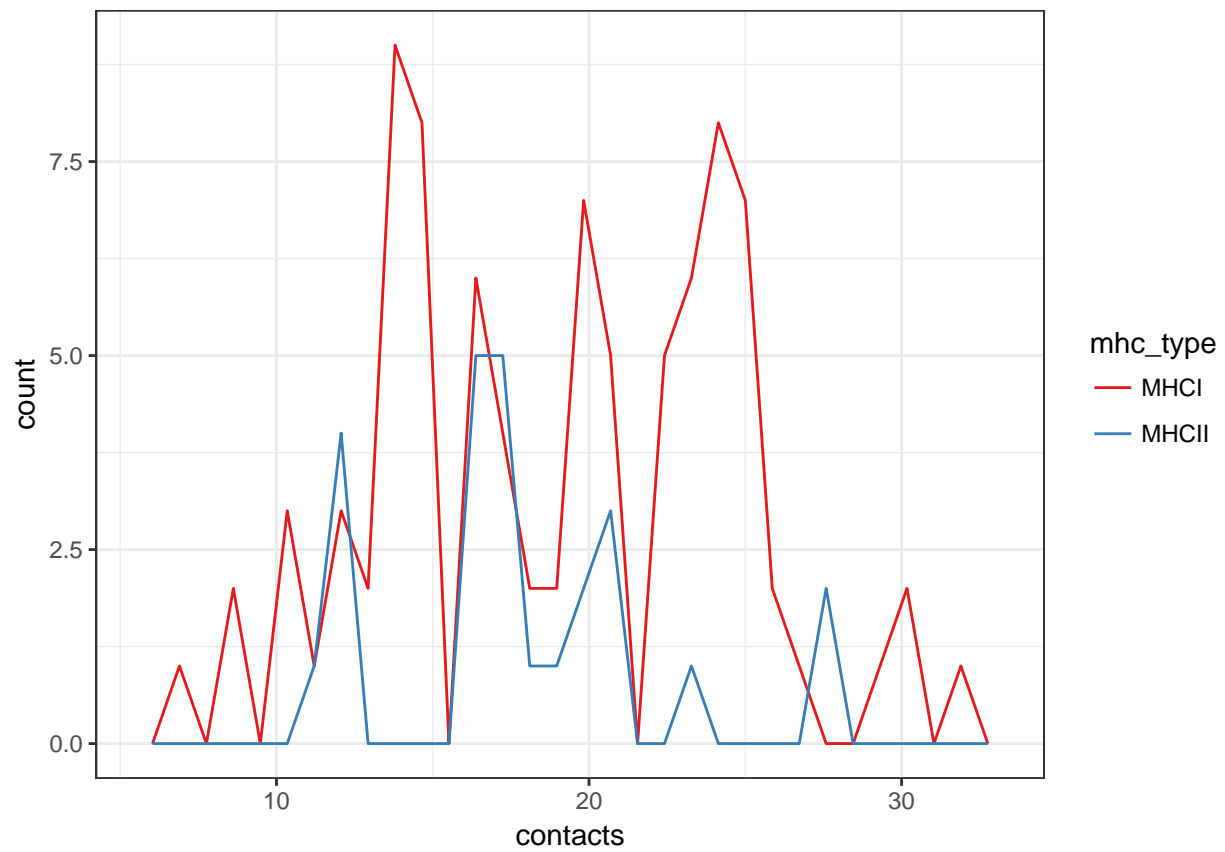
```r
df.contact.sum = df.contact.sum[pdb_id %in% df.contact.sum.pdb$pdb_id]

ggplot(df.contact.sum, aes(x=tcr_region, group = tcr_region, y = contacts, fill = tcr_region)) +
  geom_violin() +
  facet_grid(tcr_chain~mhc_type) +
  theme_bw()
```

```
a = aov(contacts~tcr_chain*tcr_region*mhc_type, df.contact.sum)
anova(a)
```

```
## Analysis of Variance Table
##
## Response: contacts
##                               Df Sum Sq Mean Sq  F value    Pr(>F)
## tcr_chain                      1    2.3    2.35   0.3763  0.539802
## tcr_region                     2 4994.2 2497.09 400.0655 < 2.2e-16 ***
## mhc_type                       1    5.3    5.34   0.8561  0.355160
## tcr_chain:tcr_region           2   61.7   30.85   4.9432  0.007396 **
## tcr_chain:mhc_type             1    0.3    0.26   0.0419  0.837848
## tcr_region:mhc_type            2    0.2    0.10   0.0160  0.984121
## tcr_chain:tcr_region:mhc_type  2    0.8    0.40   0.0642  0.937830
## Residuals                    665 4150.7    6.24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(a, "tcr_region")
```

```
##    Tukey multiple comparisons of means
##      95% family-wise confidence level
##
## Fit: aov(formula = contacts ~ tcr_chain * tcr_region * mhc_type, data = df.contact.sum)
##
## $tcr_region
##                diff       lwr       upr p adj
## CDR2-CDR1 -1.901461 -2.454139 -1.348784     0
```

```
## CDR3-CDR1   4.570796   4.018732   5.122861        0
## CDR3-CDR2   6.472258   5.919580   7.024935        0
```

```r
TukeyHSD(a, "tcr_chain:tcr_region")
```

```
##    Tukey multiple comparisons of means
##      95% family-wise confidence level
##
## Fit: aov(formula = contacts ~ tcr_chain * tcr_region * mhc_type, data = df.contact.sum)
##
## $`tcr_chain:tcr_region`
##                         diff        lwr        upr       p adj
## TRB:CDR1-TRA:CDR1 -0.7078242 -1.6577655  0.2421171 0.2732468
## TRA:CDR2-TRA:CDR1 -2.4499303 -3.4019897 -1.4978710 0.0000000
## TRB:CDR2-TRA:CDR1 -2.0620171 -3.0119584 -1.1120758 0.0000000
## TRA:CDR3-TRA:CDR1  3.8672566  2.9173153  4.8171979 0.0000000
## TRB:CDR3-TRA:CDR1  4.5665121  3.6165708  5.5164534 0.0000000
## TRA:CDR2-TRB:CDR1 -1.7421061 -2.6941655 -0.7900468 0.0000034
## TRB:CDR2-TRB:CDR1 -1.3541930 -2.3041343 -0.4042517 0.0007328
## TRA:CDR3-TRB:CDR1  4.5750808  3.6251395  5.5250221 0.0000000
## TRB:CDR3-TRB:CDR1  5.2743363  4.3243950  6.2242776 0.0000000
## TRB:CDR2-TRA:CDR2  0.3879132 -0.5641462  1.3399725 0.8535634
## TRA:CDR3-TRA:CDR2  6.3171869  5.3651276  7.2692463 0.0000000
## TRB:CDR3-TRA:CDR2  7.0164424  6.0643831  7.9685018 0.0000000
## TRA:CDR3-TRB:CDR2  5.9292738  4.9793325  6.8792151 0.0000000
## TRB:CDR3-TRB:CDR2  6.6285292  5.6785879  7.5784706 0.0000000
## TRB:CDR3-TRA:CDR3  0.6992555 -0.2506858  1.6491968 0.2864734
```

Filter TCRs with no contacts

```r
df = df[pdb_id %in% df.contact.sum.pdb$pdb_id ]
```

Contact distribution on antigen

```r
df.contact.dist.ag = df[,
                    .(contacts = sum(contact), total.pdb = length(unique(pdb_id))),
                    by=.(tcr_chain, tcr_region, mhc_type, pos_antigen, len_antigen)]

ggplot(df.contact.dist.ag, aes(x = pos_antigen - len_antigen / 2, weight = contacts / total.pdb, color =
  geom_freqpoly(bins=10) +
  facet_grid(mhc_type~tcr_region) +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```
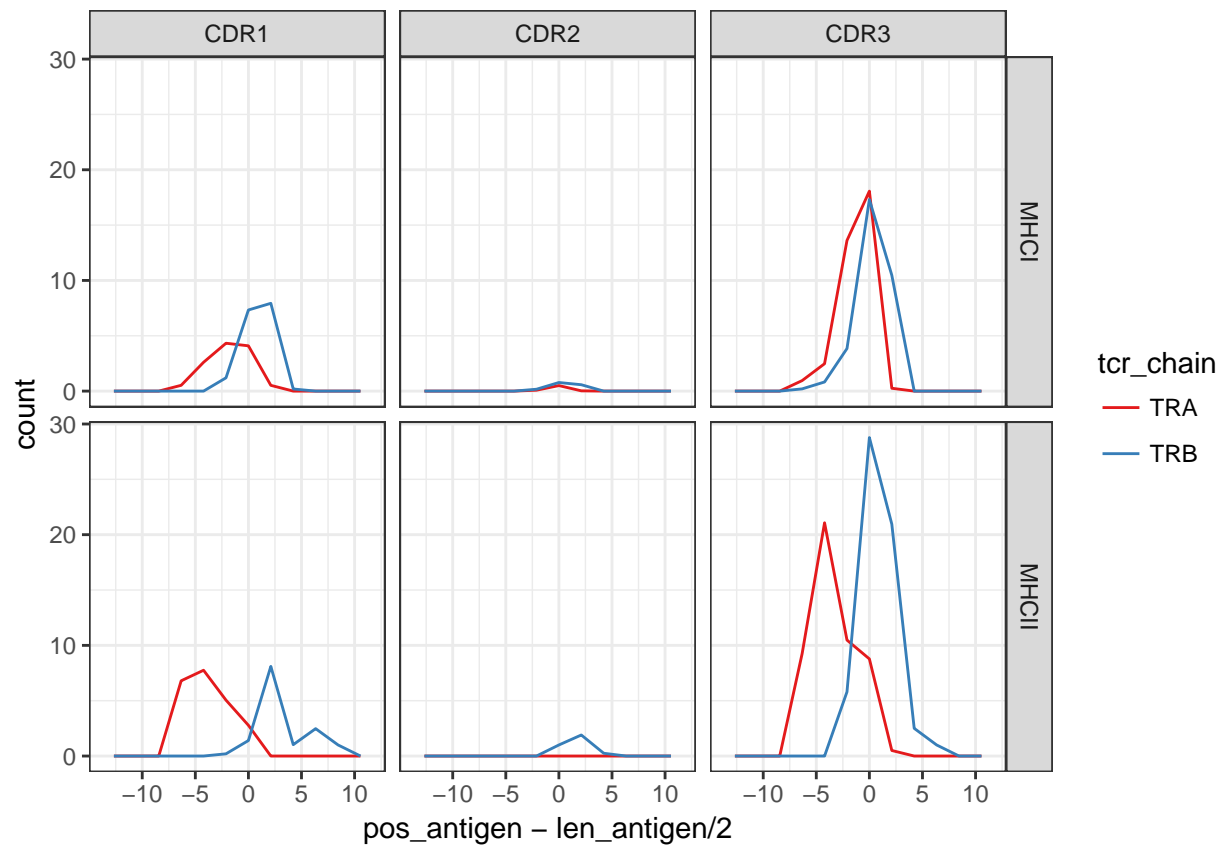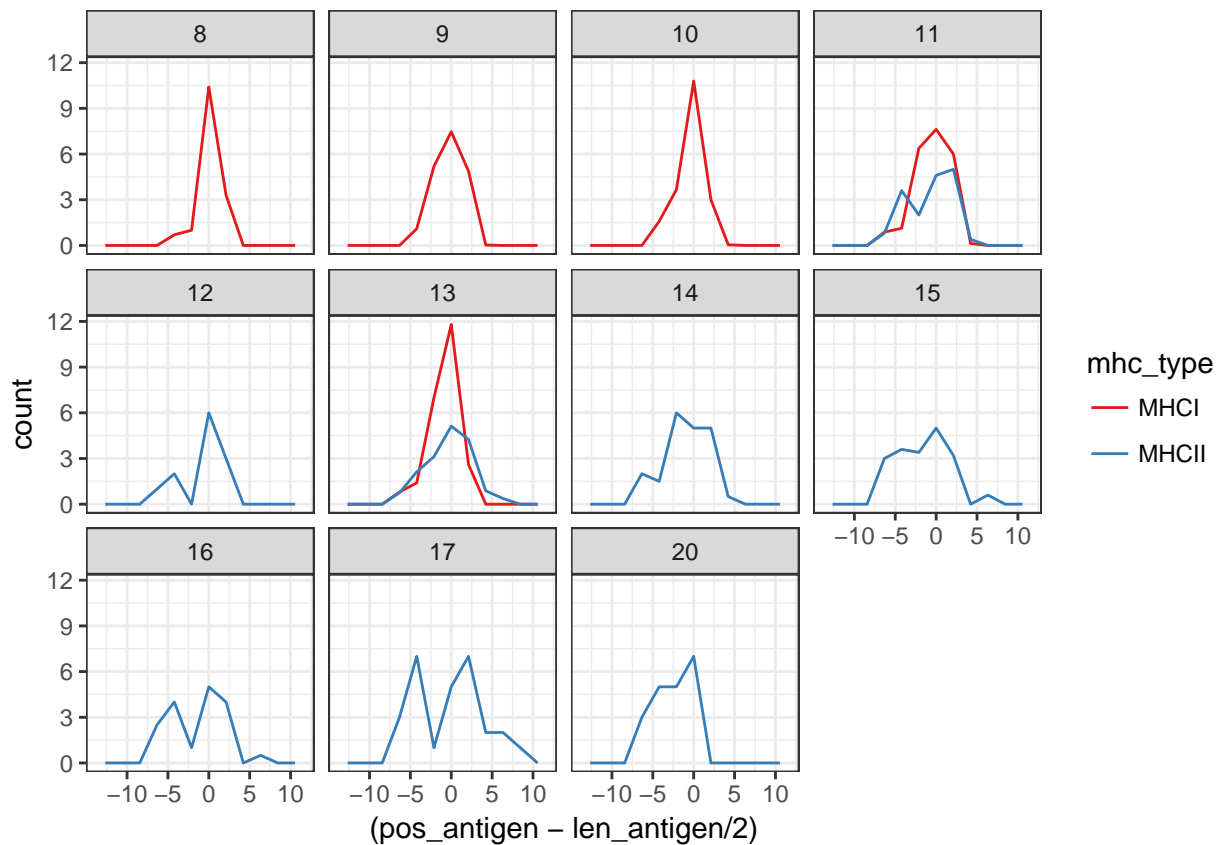
```
ggplot(df.contact.dist.ag, aes(x = (pos_antigen - len_antigen / 2) / len_antigen, weight = contacts / t
  geom_freqpoly(bins=10) +
  facet_grid(mhc_type~tcr_region) +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```
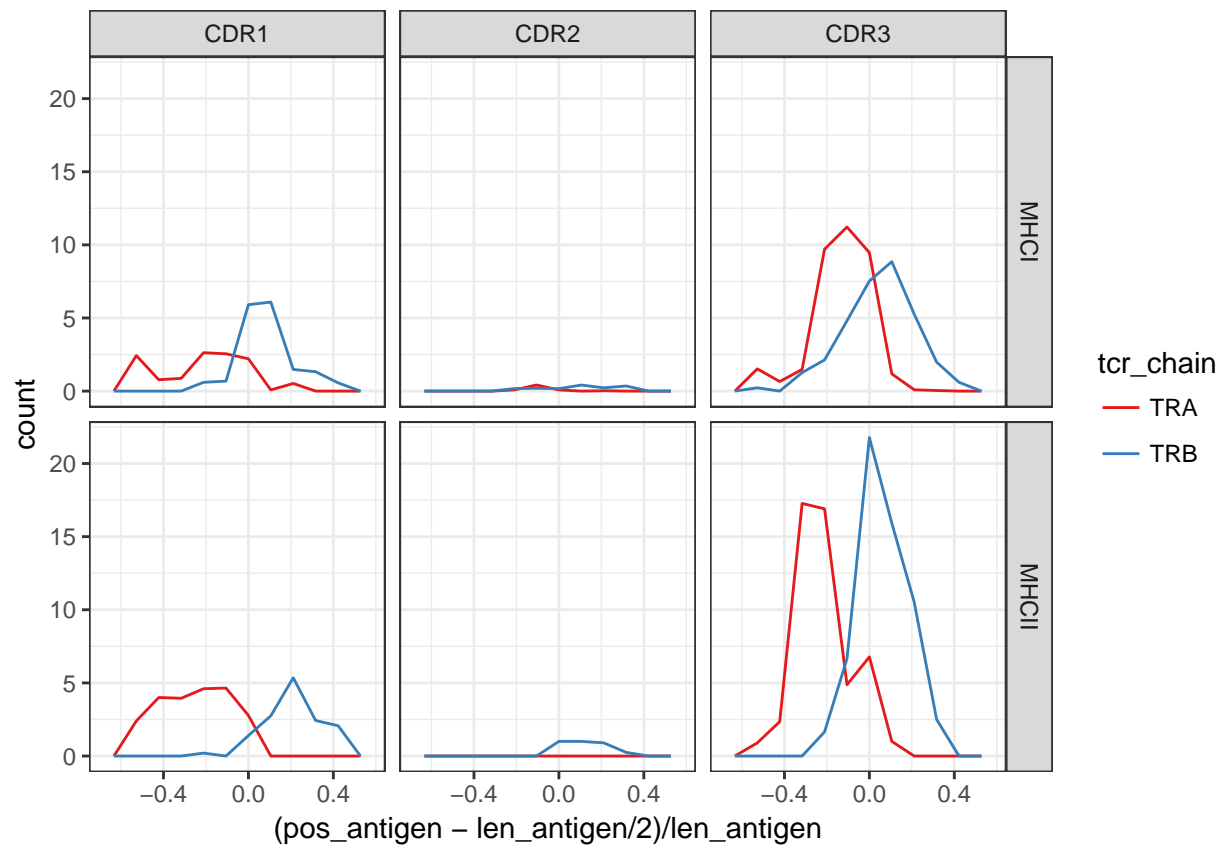
vs antigen length

```
df.contact.dist.ag.len = df[,
                         .(contacts = sum(contact), total.pdb = length(unique(pdb_id))),
                         by=.(pos_antigen, len_antigen, mhc_type)]

ggplot(df.contact.dist.ag.len, aes(x = (pos_antigen - len_antigen / 2), group = paste(len_antigen,mhc_ty
                         weight = contacts / total.pdb, color = mhc_type)) +
  geom_freqpoly(bins=10) +
  facet_wrap(~len_antigen) +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```
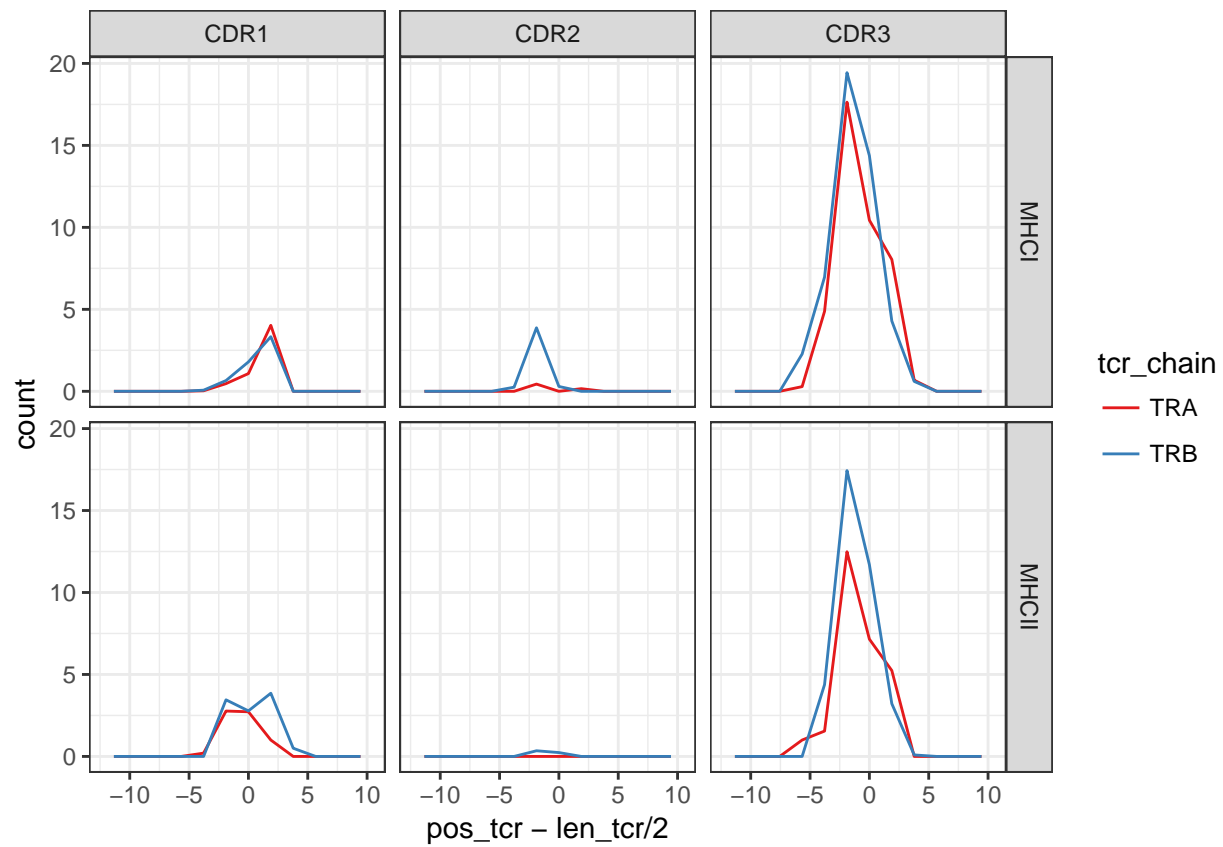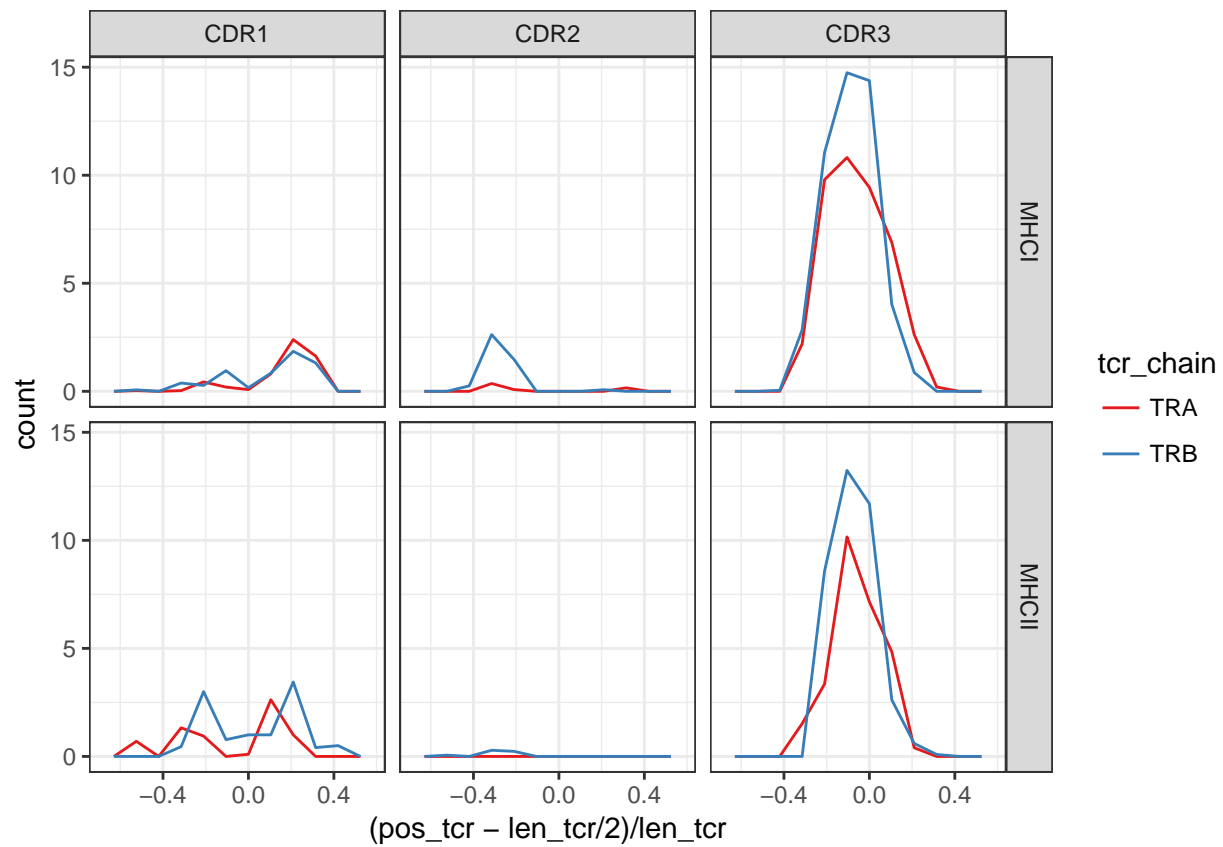
Contact distribution on TCR

```
df.contact.dist.tcr = df[,
                         .(contacts = sum(contact), total.pdb = length(unique(pdb_id))),
                         by=.(tcr_chain, tcr_region, mhc_type, pos_tcr, len_tcr)]

ggplot(df.contact.dist.tcr, aes(x = pos_tcr - len_tcr / 2, weight = contacts / total.pdb, color = tcr_cl
  geom_freqpoly(bins=10) +
  facet_grid(mhc_type~tcr_region) +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```

```
ggplot(df.contact.dist.tcr, aes(x = (pos_tcr - len_tcr / 2) / len_tcr, weight = contacts / total.pdb, c
  geom_freqpoly(bins=10) +
  facet_grid(mhc_type~tcr_region) +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```

vs CDR3 len

```r
df.contact.dist.tcr.len = df[tcr_region == "CDR3",
                            .(contacts = sum(contact), total.pdb = length(unique(pdb_id))),
                            by=.(pos_tcr, len_tcr)]

ggplot(df.contact.dist.tcr.len, aes(x = (pos_tcr - len_tcr / 2), group = len_tcr, weight = contacts / t
  geom_freqpoly(bins=10) +
  facet_wrap(~len_tcr) +
  theme_bw()
```
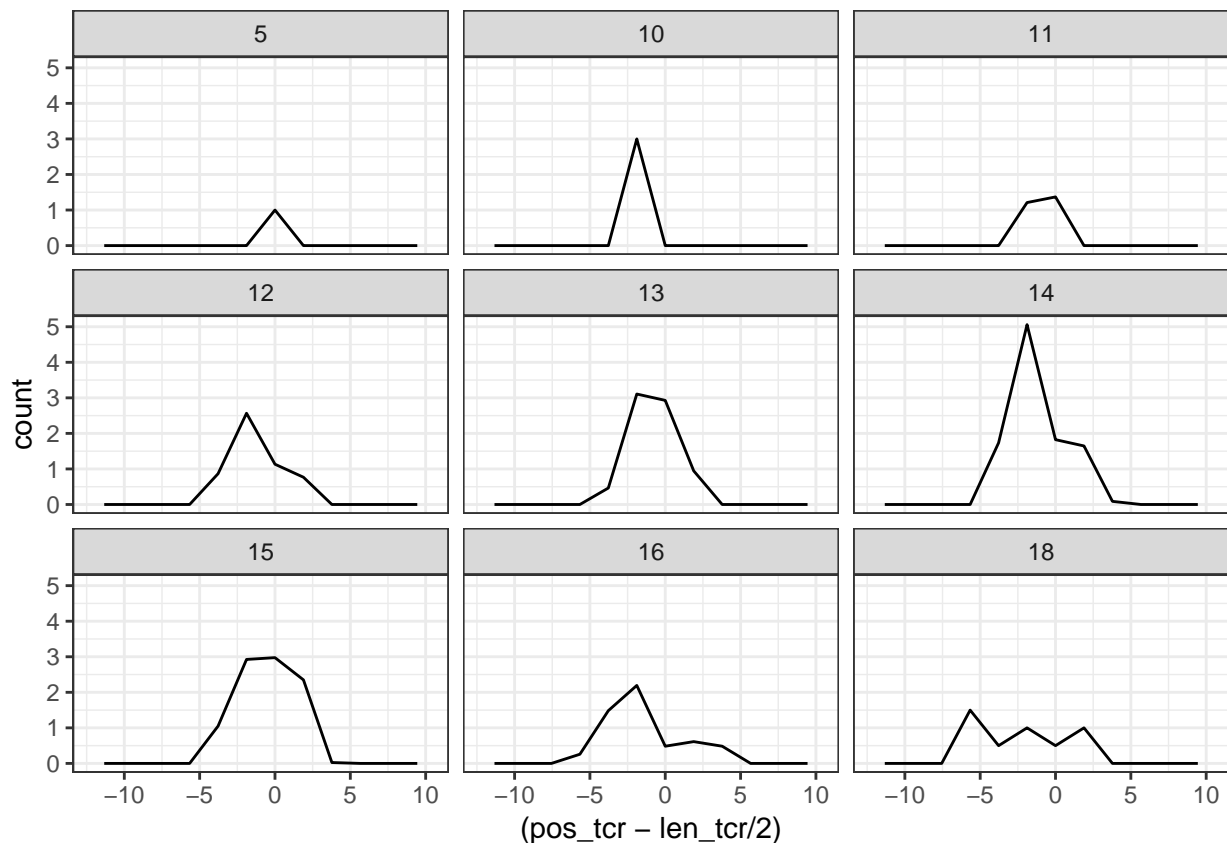
Amino acid pairs in contacts

Modelling

Center coordinates

```
df.pred = df
df.pred$pos_tcr_c = with(df.pred, pos_tcr - round(len_tcr/2))
df.pred$pos_antigen_c = with(df.pred, pos_antigen - round(len_antigen/2))
```

Calpha distance model

Simple mean model

Mean Calpha distances for centered coordinates

```
df.ca.mean = df.pred[,.(distance_CA.m = mean(distance_CA)),
                      by=.(tcr_chain, tcr_region, mhc_type, pos_tcr_c, pos_antigen_c)]

ggplot(df.ca.mean, aes(x=pos_tcr_c, y=pos_antigen_c, fill=distance_CA.m)) +
  geom_tile() +
  geom_vline(xintercept = 0, linetype = "dashed", color = "white") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "white") +
  facet_grid(mhc_type~tcr_chain+tcr_region) +
  scale_x_continuous(limits=c(-10,10)) +
```

```
scale_y_continuous(limits=c(-10,10)) +
scale_fill_gradientn(colors=colorRampPalette(brewer.pal(11, 'Spectral'))(32), limits=c(5, 20)) +
theme_bw() +
theme(panel.background = element_rect(fill = 'grey50'),
      panel.grid.major = element_blank(), panel.grid.minor = element_blank())
```
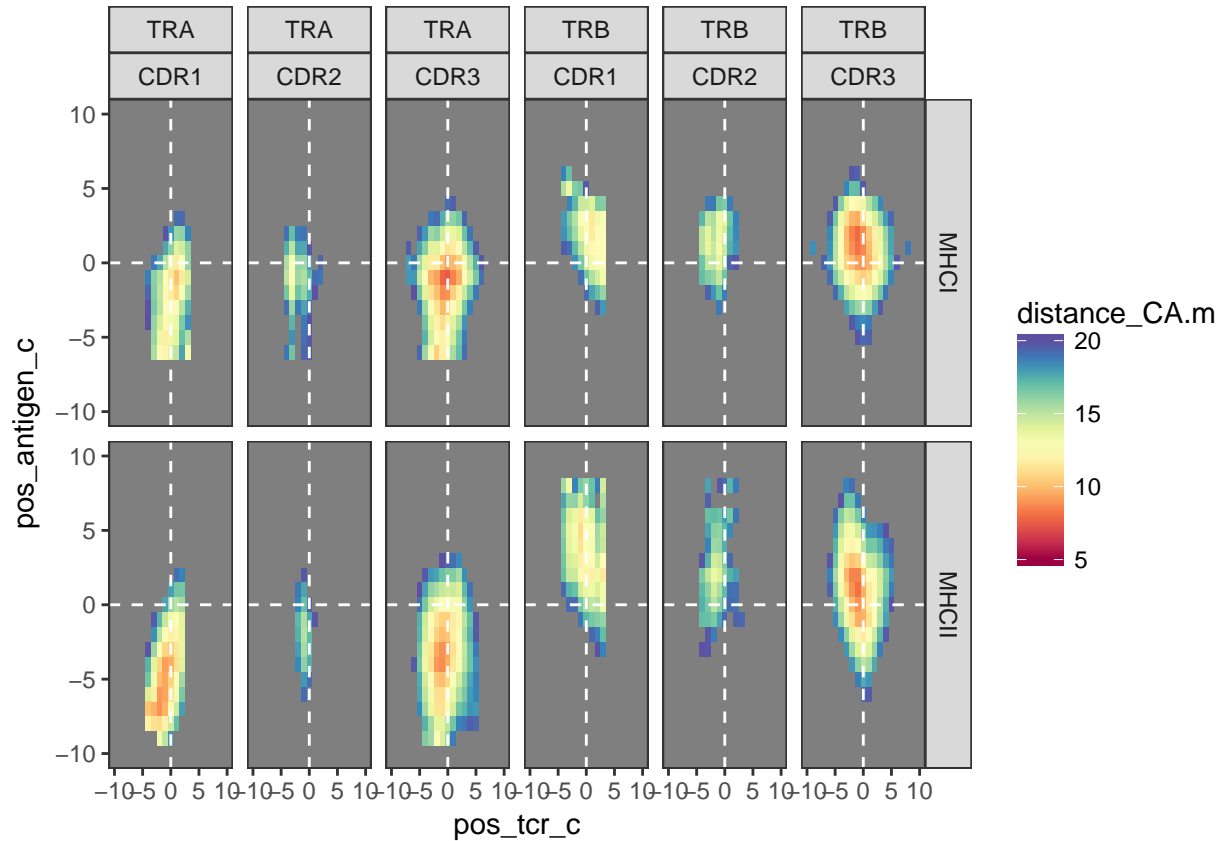

Checking the model

Add mean distance values

```
df.pred = df.pred[as.data.table(df.ca.mean), on = .(tcr_chain, tcr_region, mhc_type, pos_tcr_c, pos_ant
```

Compare to true Calpha distance values

```
ggplot(df.pred, aes(x=round(distance_CA.m), group = round(distance_CA.m), y = distance_CA)) +
  geom_boxplot() +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +
  scale_x_continuous(limits=c(5,20)) +
  scale_y_continuous(limits=c(5,20)) +
  facet_grid(mhc_type~tcr_chain+tcr_region) +
  theme_bw()
```

Warning: Removed 22609 rows containing non-finite values (stat_boxplot).

Warning: Removed 1 rows containing missing values (geom_segment).

Warning: Removed 1 rows containing missing values (geom_segment).

13

```
## Warning: Removed 1 rows containing missing values (geom_segment).

## Warning: Removed 1 rows containing missing values (geom_segment).

## Warning: Removed 1 rows containing missing values (geom_segment).

## Warning: Removed 1 rows containing missing values (geom_segment).

## Warning: Removed 1 rows containing missing values (geom_segment).

## Warning: Removed 1 rows containing missing values (geom_segment).

## Warning: Removed 1 rows containing missing values (geom_segment).

## Warning: Removed 1 rows containing missing values (geom_segment).

## Warning: Removed 1 rows containing missing values (geom_segment).

## Warning: Removed 1 rows containing missing values (geom_segment).
```



```r
summary(lm(distance_CA ~ distance_CA.m, df.pred))
```

```
##
## Call:
## lm(formula = distance_CA ~ distance_CA.m, data = df.pred)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -13.994  -1.898  -0.134   1.658  40.683
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.031e-12  4.828e-02     0.0        1
## distance_CA.m 1.000e+00  2.557e-03   391.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.262 on 55887 degrees of freedom
## Multiple R-squared:  0.7324, Adjusted R-squared:  0.7324
## F-statistic: 1.529e+05 on 1 and 55887 DF,  p-value: < 2.2e-16
```

Plot distance distribution for contacts and non-contacts, for real and estimated distances:

```
ggplot(df.pred, aes(x = distance_CA, fill = contact)) +
  geom_histogram(binwidth = 1) +
  facet_grid(mhc_type~tcr_chain+tcr_region, scales="free_y") +
  scale_x_continuous(limits=c(0,20))+
  scale_fill_brewer(palette = "Set1") +
  theme_bw()
```

```
## Warning: Removed 20140 rows containing non-finite values (stat_bin).
```
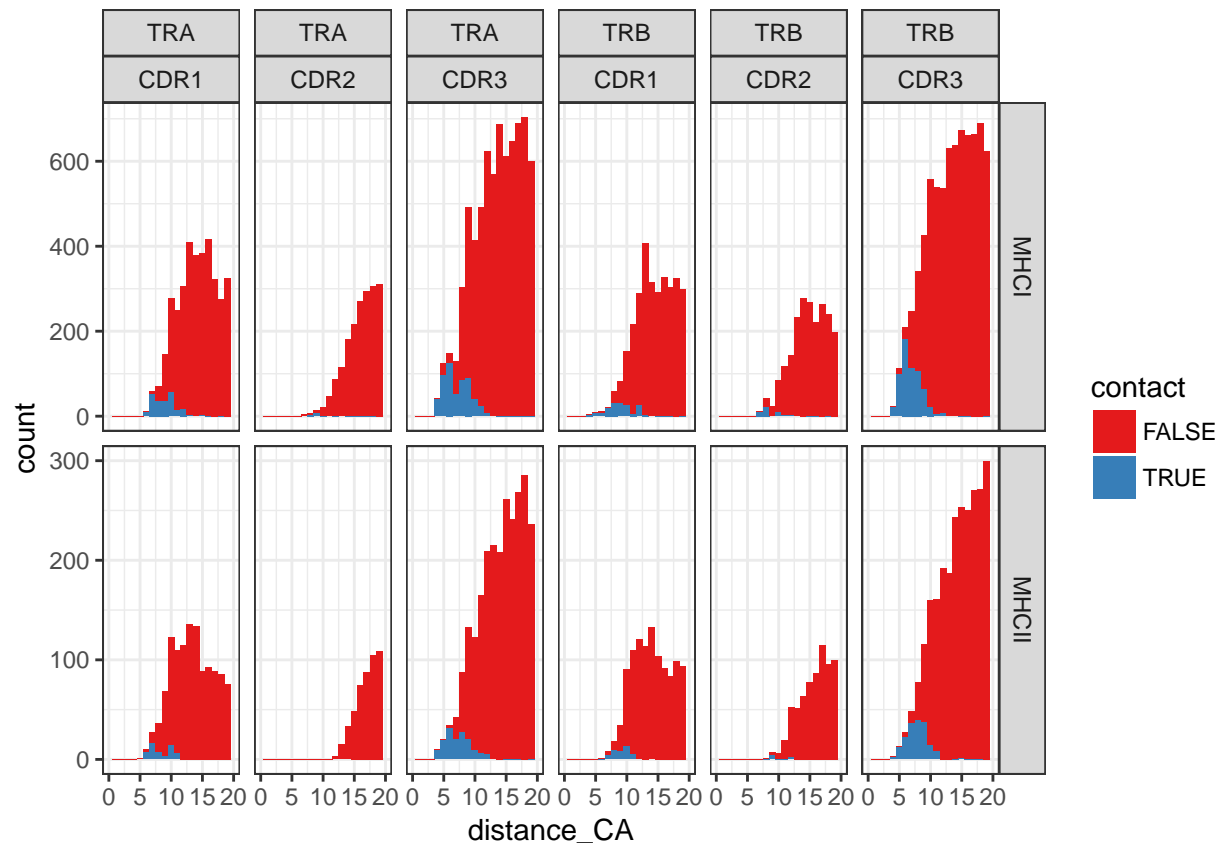


```
ggplot(df.pred, aes(x = distance_CA.m, fill = contact)) +
  geom_histogram(binwidth = 1) +
  facet_grid(mhc_type~tcr_chain+tcr_region, scales="free_y") +
  scale_x_continuous(limits=c(0,20))+
  scale_fill_brewer(palette = "Set1") +
```

```
theme_bw()
```

Warning: Removed 19288 rows containing non-finite values (stat_bin).
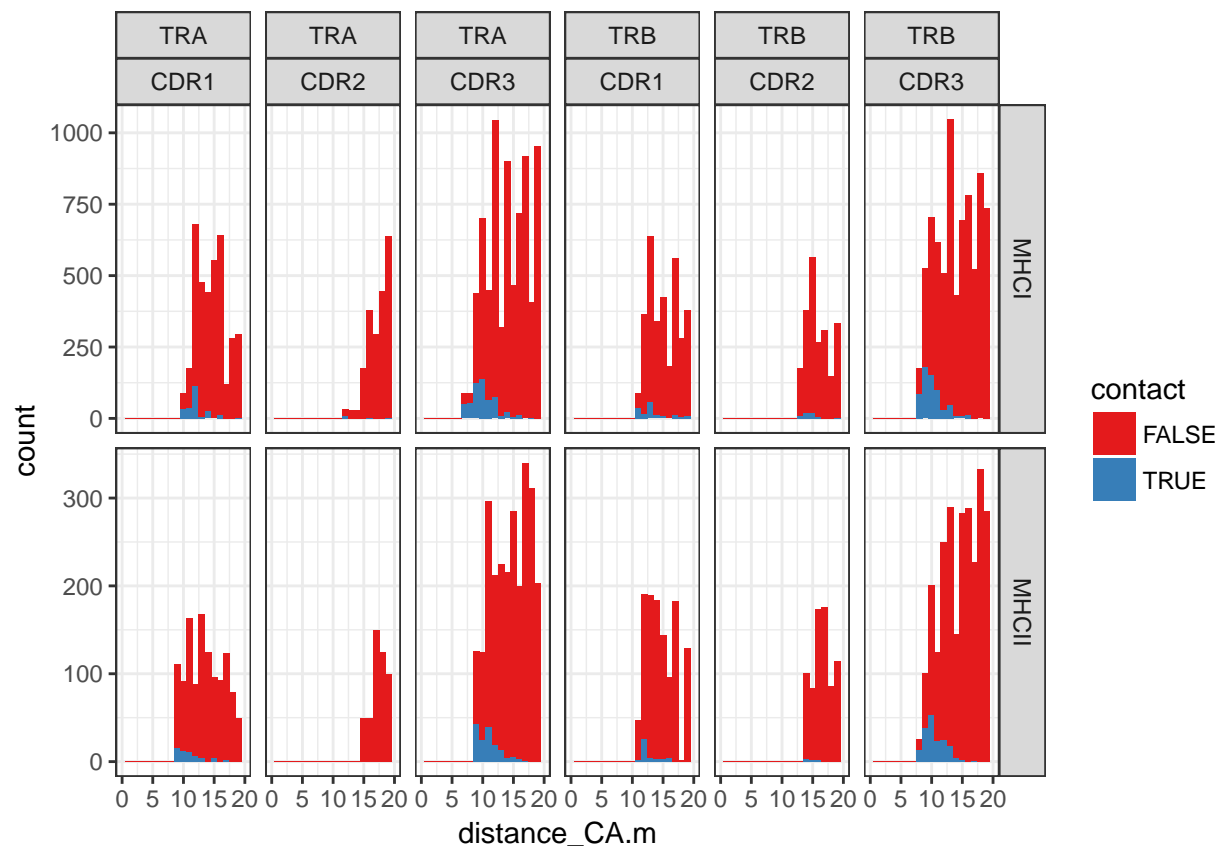
Amino acid preferences and Calpha distance

Using a generalized linear model to fit contacts, operate with amino acid pairs, ignoring which one is in TCR and which one comes from antigen.

```
res = glm(contact ~ distance_CA + aa_pair + 0, family = binomial(), data = df.pred)
```

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

```
summary(res)
```

```
##
## Call:
## glm(formula = contact ~ distance_CA + aa_pair + 0, family = binomial(),
##     data = df.pred)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2988  -0.0306  -0.0024  -0.0001   3.5854
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## distance_CA   -1.1641     0.0239 -48.697  < 2e-16 ***
```

```
## aa_pairA_A      7.4506      0.6490   11.480   < 2e-16 ***
## aa_pairA_C     -5.0823    780.4740   -0.007 0.994804
## aa_pairA_D      6.5046      1.0711    6.073 1.25e-09 ***
## aa_pairA_E      8.0084      0.6800   11.778   < 2e-16 ***
## aa_pairA_F      8.0345      0.5030   15.973   < 2e-16 ***
## aa_pairA_G      6.2345      0.4301   14.495   < 2e-16 ***
## aa_pairA_H     10.6128      0.7373   14.394   < 2e-16 ***
## aa_pairA_I      7.6925      0.7669   10.031   < 2e-16 ***
## aa_pairA_K      9.5920      0.7834   12.244   < 2e-16 ***
## aa_pairA_L      8.8465      0.3937   22.473   < 2e-16 ***
## aa_pairA_M     -5.9568   1130.5277   -0.005 0.995796
## aa_pairA_N      7.3236      0.5978   12.251   < 2e-16 ***
## aa_pairA_P      7.2898      0.6451   11.300   < 2e-16 ***
## aa_pairA_Q      9.8584      0.3929   25.089   < 2e-16 ***
## aa_pairA_R      8.9961      0.4562   19.718   < 2e-16 ***
## aa_pairA_S      6.5389      0.4515   14.482   < 2e-16 ***
## aa_pairA_T      7.2314      0.4943   14.629   < 2e-16 ***
## aa_pairA_V      8.3448      0.5171   16.138   < 2e-16 ***
## aa_pairA_W      8.1996      0.7161   11.451   < 2e-16 ***
## aa_pairA_Y      9.6342      0.4171   23.100   < 2e-16 ***
## aa_pairC_C      5.5391   2779.5674    0.002 0.998410
## aa_pairC_D     -6.8313   1132.6530   -0.006 0.995188
## aa_pairC_E     -2.0493    926.5239   -0.002 0.998235
## aa_pairC_F     -0.7815    797.9707   -0.001 0.999219
## aa_pairC_G      9.2124      1.2538    7.348 2.02e-13 ***
## aa_pairC_H     -4.3174   2039.2983   -0.002 0.998311
## aa_pairC_I     -3.0568    862.9445   -0.004 0.997174
## aa_pairC_K      4.2483   1806.6199    0.002 0.998124
## aa_pairC_L     -3.0819    592.3562   -0.005 0.995849
## aa_pairC_M     -2.1508   1908.9111   -0.001 0.999101
## aa_pairC_N     -6.7657   1332.2014   -0.005 0.995948
## aa_pairC_P     -2.0699    697.2333   -0.003 0.997631
## aa_pairC_Q     -4.5411    848.1169   -0.005 0.995728
## aa_pairC_R     10.4997      2.7515    3.816 0.000136 ***
## aa_pairC_S      8.5894      1.5813    5.432 5.58e-08 ***
## aa_pairC_T     -5.3605    850.3332   -0.006 0.994970
## aa_pairC_V     -2.0320    802.1970   -0.003 0.997979
## aa_pairC_W     14.6097     15.2080    0.961 0.336724
## aa_pairC_Y     11.5405      1.1727    9.841   < 2e-16 ***
## aa_pairD_D     -8.4260   1885.3630   -0.004 0.996434
## aa_pairD_E      8.8254      0.8749   10.088   < 2e-16 ***
## aa_pairD_F      7.6371      0.4809   15.881   < 2e-16 ***
## aa_pairD_G      7.7062      0.4204   18.332   < 2e-16 ***
## aa_pairD_H     10.5940      1.2362    8.570   < 2e-16 ***
## aa_pairD_I      8.3684      1.0899    7.678 1.61e-14 ***
## aa_pairD_K     11.3968      0.6166   18.485   < 2e-16 ***
## aa_pairD_L      8.0172      0.5329   15.044   < 2e-16 ***
## aa_pairD_M     10.5306      0.7343   14.341   < 2e-16 ***
## aa_pairD_N     10.5382      0.5275   19.977   < 2e-16 ***
## aa_pairD_P      8.5002      0.5117   16.611   < 2e-16 ***
## aa_pairD_Q      9.8007      0.5269   18.602   < 2e-16 ***
## aa_pairD_R     10.2193      0.4511   22.654   < 2e-16 ***
## aa_pairD_S      8.5882      0.3648   23.542   < 2e-16 ***
## aa_pairD_T      9.4943      0.5013   18.940   < 2e-16 ***
```

```
## aa_pairD_V      7.6737      0.8521    9.005   < 2e-16 ***
## aa_pairD_W     10.0487      0.5646   17.798   < 2e-16 ***
## aa_pairD_Y     12.5331      0.4058   30.886   < 2e-16 ***
## aa_pairE_E     -4.1498   1179.5758   -0.004  0.997193
## aa_pairE_F      9.3895      0.8108   11.580   < 2e-16 ***
## aa_pairE_G      8.5642      0.3830   22.361   < 2e-16 ***
## aa_pairE_H     -5.4856   1406.6838   -0.004  0.996889
## aa_pairE_I      8.9068      1.0515    8.471   < 2e-16 ***
## aa_pairE_K     12.4610      0.5940   20.980   < 2e-16 ***
## aa_pairE_L      8.6543      0.5354   16.164   < 2e-16 ***
## aa_pairE_M     -4.6918   1639.6501   -0.003  0.997717
## aa_pairE_N      8.1989      0.7943   10.322   < 2e-16 ***
## aa_pairE_P      7.6073      1.1323    6.718  1.84e-11 ***
## aa_pairE_Q     11.8648      0.4099   28.947   < 2e-16 ***
## aa_pairE_R     11.3891      0.4742   24.018   < 2e-16 ***
## aa_pairE_S      8.6229      0.5590   15.425   < 2e-16 ***
## aa_pairE_T      7.7773      0.6697   11.614   < 2e-16 ***
## aa_pairE_V     10.1429      0.6650   15.253   < 2e-16 ***
## aa_pairE_W     10.4254      0.9237   11.287   < 2e-16 ***
## aa_pairE_Y     12.3228      0.4422   27.870   < 2e-16 ***
## aa_pairF_F     -3.2224    771.2796   -0.004  0.996666
## aa_pairF_G      7.9085      0.3799   20.818   < 2e-16 ***
## aa_pairF_H     -5.1197   1246.1192   -0.004  0.996722
## aa_pairF_I      9.2106      0.5143   17.907   < 2e-16 ***
## aa_pairF_K     10.2473      1.0648    9.623   < 2e-16 ***
## aa_pairF_L      8.6929      0.5704   15.239   < 2e-16 ***
## aa_pairF_M      9.6971      0.9663   10.035   < 2e-16 ***
## aa_pairF_N      8.5956      0.5080   16.920   < 2e-16 ***
## aa_pairF_P      9.6564      0.4587   21.052   < 2e-16 ***
## aa_pairF_Q      9.9189      0.4133   23.999   < 2e-16 ***
## aa_pairF_R     10.9736      0.4013   27.347   < 2e-16 ***
## aa_pairF_S      8.6672      0.3586   24.168   < 2e-16 ***
## aa_pairF_T      9.9687      0.4255   23.430   < 2e-16 ***
## aa_pairF_V      7.8259      1.1200    6.987  2.80e-12 ***
## aa_pairF_W      9.7643      0.6815   14.328   < 2e-16 ***
## aa_pairF_Y     10.6024      0.4121   25.727   < 2e-16 ***
## aa_pairG_G      6.4114      0.2999   21.378   < 2e-16 ***
## aa_pairG_H      8.8415      0.6071   14.564   < 2e-16 ***
## aa_pairG_I      7.3621      0.3630   20.282   < 2e-16 ***
## aa_pairG_K      9.3725      0.4510   20.782   < 2e-16 ***
## aa_pairG_L      8.0443      0.2913   27.618   < 2e-16 ***
## aa_pairG_M      8.8593      0.3993   22.187   < 2e-16 ***
## aa_pairG_N      8.2034      0.3522   23.290   < 2e-16 ***
## aa_pairG_P      7.3596      0.3688   19.958   < 2e-16 ***
## aa_pairG_Q      9.2003      0.3126   29.432   < 2e-16 ***
## aa_pairG_R     10.0279      0.3850   26.045   < 2e-16 ***
## aa_pairG_S      6.6783      0.3075   21.721   < 2e-16 ***
## aa_pairG_T      7.5166      0.3074   24.449   < 2e-16 ***
## aa_pairG_V      7.1543      0.3496   20.465   < 2e-16 ***
## aa_pairG_W      8.9571      0.4065   22.033   < 2e-16 ***
## aa_pairG_Y      8.3490      0.3134   26.640   < 2e-16 ***
## aa_pairH_H     -5.9571   4189.8593   -0.001  0.998866
## aa_pairH_I     -5.3697   1577.0481   -0.003  0.997283
## aa_pairH_K     -6.6937   2098.9520   -0.003  0.997455
```

```
## aa_pairH_L    -7.0644   843.9878   -0.008 0.993322
## aa_pairH_M    -5.8359  2692.4847   -0.002 0.998271
## aa_pairH_N    10.2543     0.9601   10.681  < 2e-16 ***
## aa_pairH_P    -7.4070  1291.9322   -0.006 0.995426
## aa_pairH_Q     9.1952     0.7813   11.769  < 2e-16 ***
## aa_pairH_R    -4.2777  1719.6820   -0.002 0.998015
## aa_pairH_S    -6.6240  1059.3532   -0.006 0.995011
## aa_pairH_T     8.9645     0.6992   12.821  < 2e-16 ***
## aa_pairH_V     9.4917     1.1789    8.051 8.21e-16 ***
## aa_pairH_W    11.0727     1.1758    9.417  < 2e-16 ***
## aa_pairH_Y    10.5947     0.6673   15.876  < 2e-16 ***
## aa_pairI_I    -5.1068  1264.5330   -0.004 0.996778
## aa_pairI_K     9.6424     1.1858    8.131 4.25e-16 ***
## aa_pairI_L    10.0496     0.5433   18.498  < 2e-16 ***
## aa_pairI_M     9.1956     0.7545   12.187  < 2e-16 ***
## aa_pairI_N     8.8260     0.4763   18.531  < 2e-16 ***
## aa_pairI_P     8.8625     1.1070    8.006 1.19e-15 ***
## aa_pairI_Q     9.7686     0.4811   20.303  < 2e-16 ***
## aa_pairI_R    10.3780     0.7608   13.641  < 2e-16 ***
## aa_pairI_S     8.9292     0.4018   22.225  < 2e-16 ***
## aa_pairI_T     9.2663     0.4067   22.784  < 2e-16 ***
## aa_pairI_V     9.4079     0.5409   17.392  < 2e-16 ***
## aa_pairI_W     9.4824     0.7782   12.186  < 2e-16 ***
## aa_pairI_Y    10.7984     0.4302   25.102  < 2e-16 ***
## aa_pairK_K    -4.4266  2144.6153   -0.002 0.998353
## aa_pairK_L     8.9371     0.9337    9.571  < 2e-16 ***
## aa_pairK_M    -4.1309  1973.7639   -0.002 0.998330
## aa_pairK_N    10.2295     0.5343   19.147  < 2e-16 ***
## aa_pairK_P     9.0716     0.6459   14.044  < 2e-16 ***
## aa_pairK_Q     9.8796     0.7776   12.706  < 2e-16 ***
## aa_pairK_R     9.6186     0.9726    9.889  < 2e-16 ***
## aa_pairK_S    10.4602     0.4901   21.343  < 2e-16 ***
## aa_pairK_T     9.8153     0.6671   14.712  < 2e-16 ***
## aa_pairK_V     9.0919     0.8868   10.252  < 2e-16 ***
## aa_pairK_W    -6.4313  2300.2325   -0.003 0.997769
## aa_pairK_Y    10.5764     0.7582   13.949  < 2e-16 ***
## aa_pairL_L     8.4956     0.5063   16.779  < 2e-16 ***
## aa_pairL_M    10.1232     0.5793   17.476  < 2e-16 ***
## aa_pairL_N     9.2011     0.3892   23.641  < 2e-16 ***
## aa_pairL_P     9.4667     0.4228   22.392  < 2e-16 ***
## aa_pairL_Q    11.1273     0.3417   32.561  < 2e-16 ***
## aa_pairL_R    11.0384     0.4089   26.996  < 2e-16 ***
## aa_pairL_S     8.1457     0.3843   21.198  < 2e-16 ***
## aa_pairL_T     8.2599     0.3723   22.188  < 2e-16 ***
## aa_pairL_V     8.7641     0.5618   15.599  < 2e-16 ***
## aa_pairL_W    10.1963     0.6578   15.500  < 2e-16 ***
## aa_pairL_Y    10.5892     0.3733   28.365  < 2e-16 ***
## aa_pairM_M    -4.2232  3270.1334   -0.001 0.998970
## aa_pairM_N     9.7992     0.8052   12.170  < 2e-16 ***
## aa_pairM_P    11.2941     0.5793   19.497  < 2e-16 ***
## aa_pairM_Q     9.1321     0.6825   13.380  < 2e-16 ***
## aa_pairM_R    10.2472     1.1014    9.304  < 2e-16 ***
## aa_pairM_S    10.3857     0.6830   15.206  < 2e-16 ***
## aa_pairM_T    10.4871     0.5915   17.729  < 2e-16 ***
```

```
## aa_pairM_V     10.0076     0.7358  13.602  < 2e-16 ***
## aa_pairM_W     -1.3742  2890.8807   0.000 0.999621
## aa_pairM_Y     10.8579     0.5073  21.402  < 2e-16 ***
## aa_pairN_N      9.7526     0.7056  13.822  < 2e-16 ***
## aa_pairN_P      9.0655     0.4420  20.509  < 2e-16 ***
## aa_pairN_Q     10.1333     0.4000  25.332  < 2e-16 ***
## aa_pairN_R     10.1765     0.5927  17.168  < 2e-16 ***
## aa_pairN_S      8.5840     0.5222  16.440  < 2e-16 ***
## aa_pairN_T      9.1378     0.4950  18.461  < 2e-16 ***
## aa_pairN_V      8.3114     0.6490  12.806  < 2e-16 ***
## aa_pairN_W     -8.4428  1476.7351  -0.006 0.995438
## aa_pairN_Y     11.5988     0.4668  24.845  < 2e-16 ***
## aa_pairP_P      8.3694     1.0895   7.682 1.57e-14 ***
## aa_pairP_Q      9.7304     0.5615  17.328  < 2e-16 ***
## aa_pairP_R     11.7045     0.4296  27.242  < 2e-16 ***
## aa_pairP_S      8.7843     0.3882  22.630  < 2e-16 ***
## aa_pairP_T      8.5350     0.4719  18.087  < 2e-16 ***
## aa_pairP_V      8.6306     0.6433  13.417  < 2e-16 ***
## aa_pairP_W     10.9156     0.4580  23.832  < 2e-16 ***
## aa_pairP_Y     10.1015     0.4318  23.395  < 2e-16 ***
## aa_pairQ_Q     -4.6743  1285.0453  -0.004 0.997098
## aa_pairQ_R     12.4911     0.5125  24.373  < 2e-16 ***
## aa_pairQ_S      9.0880     0.4056  22.409  < 2e-16 ***
## aa_pairQ_T      9.2985     0.4610  20.172  < 2e-16 ***
## aa_pairQ_V      9.8137     0.4857  20.206  < 2e-16 ***
## aa_pairQ_W      9.0369     0.9533   9.479  < 2e-16 ***
## aa_pairQ_Y     10.9516     0.4023  27.225  < 2e-16 ***
## aa_pairR_R     13.3130     0.8040  16.559  < 2e-16 ***
## aa_pairR_S     10.2157     0.4588  22.268  < 2e-16 ***
## aa_pairR_T      8.2202     0.5519  14.895  < 2e-16 ***
## aa_pairR_V      7.0930     1.0785   6.577 4.80e-11 ***
## aa_pairR_W     12.5928     0.6466  19.475  < 2e-16 ***
## aa_pairR_Y     11.7568     0.4875  24.117  < 2e-16 ***
## aa_pairS_S      8.0212     0.7304  10.983  < 2e-16 ***
## aa_pairS_T      7.7769     0.6008  12.944  < 2e-16 ***
## aa_pairS_V      8.1070     0.4649  17.438  < 2e-16 ***
## aa_pairS_W      8.8810     0.5756  15.428  < 2e-16 ***
## aa_pairS_Y      9.9924     0.3501  28.540  < 2e-16 ***
## aa_pairT_T      8.7818     0.5402  16.256  < 2e-16 ***
## aa_pairT_V      7.4450     0.5503  13.530  < 2e-16 ***
## aa_pairT_W      9.3202     0.5622  16.579  < 2e-16 ***
## aa_pairT_Y     10.6287     0.4068  26.128  < 2e-16 ***
## aa_pairV_V      9.2680     0.8291  11.179  < 2e-16 ***
## aa_pairV_W     10.3870     0.9506  10.927  < 2e-16 ***
## aa_pairV_Y      9.4449     0.5751  16.422  < 2e-16 ***
## aa_pairW_W     13.8331     1.2502  11.065  < 2e-16 ***
## aa_pairW_Y     10.6918     0.3910  27.347  < 2e-16 ***
## aa_pairY_Y     11.8573     0.4829  24.552  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 77478.6  on 55889  degrees of freedom
```

```
## Residual deviance:  5776.2  on 55678  degrees of freedom
## AIC: 6198.2
##
## Number of Fisher Scoring iterations: 19
```

Explore results

```r
coef = res$coefficients
pvalue = coef(summary(res))[,4]
coef = ifelse(pvalue < 0.05, coef, NA)
names(coef) = str_split_fixed(names(coef), "aa_pair", 2)[,2]

df.aa.coef = data.frame(coef = coef,
                        aa_tcr = str_split_fixed(names(coef), "_", 2)[, 1],
                        aa_antigen = str_split_fixed(names(coef), "_", 2)[ ,2]) %>%
  filter(aa_tcr != "" & aa_antigen != "") %>%
  droplevels

df.aa.coef.diag = df.aa.coef
df.aa.coef.diag$aa_pair = with(df.aa.coef.diag,
  as.factor(ifelse(as.character(aa_tcr) < as.character(aa_antigen), paste(aa_tcr, aa_antigen, sep = "_")

df.aa.coef.diag = df.aa.coef.diag %>% select(aa_pair, coef)

df.aa.coef.rev = df.aa.coef
df.aa.coef.rev$aa_tcr = df.aa.coef$aa_antigen
df.aa.coef.rev$aa_antigen = df.aa.coef$aa_tcr

df.aa.coef = rbind(df.aa.coef, df.aa.coef.rev) %>% unique()

# transform to matrix and plot heatmap.2

aa_pair_mat = dcast(df.aa.coef, aa_tcr ~ aa_antigen, value.var = "coef", fun.aggregate = mean)
rownames(aa_pair_mat) = aa_pair_mat$aa_tcr
aa_pair_mat$aa_tcr = NULL
aa_pair_mat = as.matrix(aa_pair_mat)

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) %>%
  mutate(hydrop.sc = round(100 * (hydrop - min(hydrop)) / (max(hydrop) - min(hydrop))))

df.hydro$color = colorRampPalette(brewer.pal(11, 'PRGn'))(101)[df.hydro$hydrop.sc + 1]

aa_colors = df.hydro$color
names(aa_colors) = df.hydro$aa
```

```
heatmap.2(aa_pair_mat,
          hclustfun = function(x) hclust(x, method = "ward.D2"),
          RowSideColors = aa_colors[rownames(aa_pair_mat)],
          ColSideColors = aa_colors[colnames(aa_pair_mat)],
          trace = "none",
          #breaks = seq(-16, -7, length.out = 101),
          col=colorRampPalette(brewer.pal(9, 'YlGnBu'))(32))
```
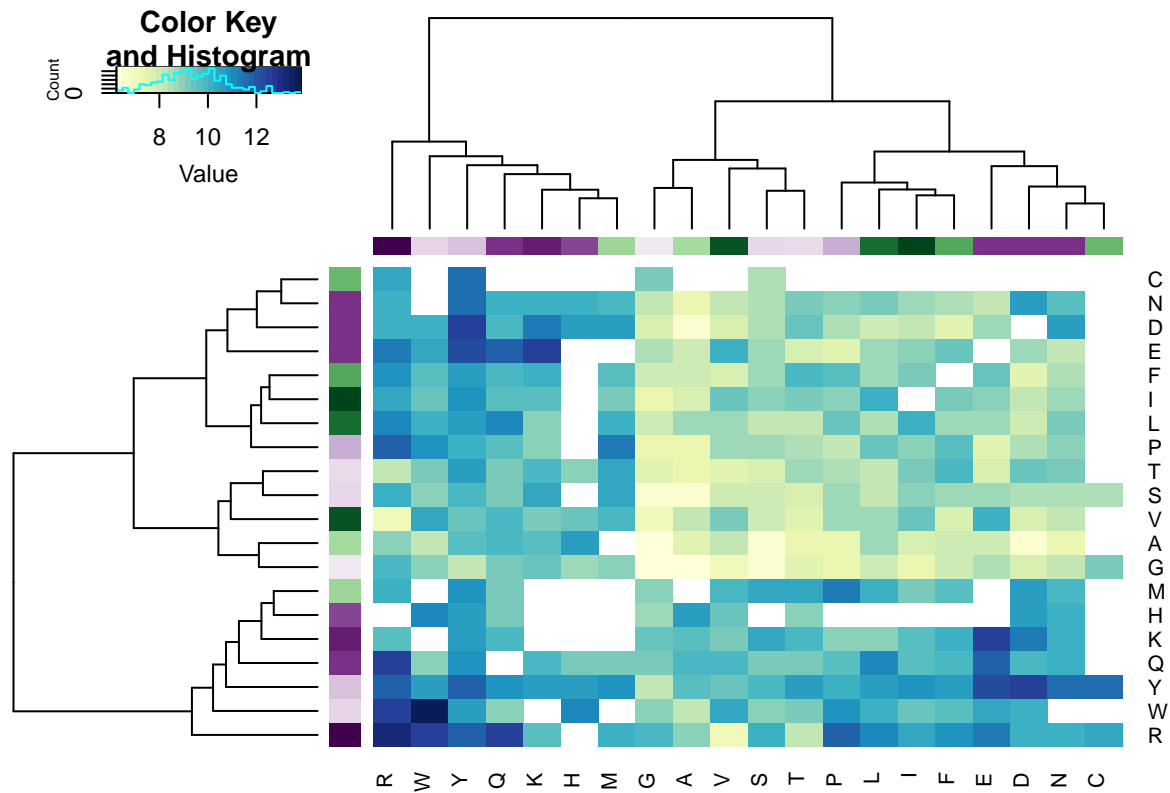

Impute missing values

```
kidera = t(data.frame(lapply(strsplit("A,-1.56,-1.67,-0.97,-0.27,-0.93,-0.78,-0.2,-0.08,0.21,-0.48;R,0.

kidera[,2:ncol(kidera)] = lapply(kidera[,2:ncol(kidera)], function (col) (col - min(col)) / (max(col) -
row.names(kidera) = kidera$amino.acid
kidera = as.matrix(kidera[,-1])


generate_data <- function (.mat, .train_size = .7, .cv = 10, .seed = 42) {
  melted = melt(.mat)[melt(upper.tri(.mat, diag = T))[,3],]
  melted[,1] = as.character(melted[,1])
  melted[,2] = as.character(melted[,2])
  train_data = melted[!is.na(melted[,3]), ]
  test_data = melted[is.na(melted[,3]), ]

  med = median(train_data[,3])

  hi_logic = train_data[,3] >= med
```

```r
    lo_logic = train_data[,3] < med

    train_size = round(.train_size * nrow(train_data))

    train_inds = list()
    val_inds = list()

    set.seed(.seed)
    for (i in 1:.cv) {
      hi_inds = sample(which(hi_logic), train_size / 2, F)
      lo_inds = sample(which(lo_logic), train_size / 2, F)
      train_inds[[i]] = sample(c(hi_inds, lo_inds))

      val_inds[[i]] = sample(c(setdiff(which(hi_logic), hi_inds), setdiff(which(lo_logic), lo_inds)))
    }

    res = matrix(0, nrow(train_data), 10)
    for (i in 1:nrow(res)) {
      res[i,] = (kidera[train_data[i,1], ] + kidera[train_data[i,2], ]) / 2
    }
    row.names(res) = paste0(train_data[,1], train_data[,2])

    res_tst = matrix(0, nrow(test_data), 10)
    for (i in 1:nrow(res_tst)) {
      res_tst[i,] = (kidera[test_data[i,1], ] + kidera[test_data[i,2], ]) / 2
    }
    row.names(res_tst) = paste0(test_data[,1], test_data[,2])

    list(X = res, y = train_data[,3], X_test = res_tst, train = train_inds, val = val_inds)
}


impute_knn_mean <- function (X_train, y_train, X_val, y_val, X_test, .k) {
  res_tr = sapply(1:nrow(X_train), function (row_i) {
    tmp = sapply(1:nrow(X_train), function (row_j) {
      if (row_i != row_j) { sqrt(sum((X_train[row_i, ] - X_train[row_j, ]) ^ 2))
      }
      else { 20 }
    })
    mean(y_train[order(tmp)[1:.k]])
  })

  res_val = sapply(1:nrow(X_val), function (row_i) {
    tmp = sapply(1:nrow(X_train), function (row_j) {
      sqrt(sum((X_val[row_i, ] - X_train[row_j, ]) ^ 2))
    })
    mean(y_train[order(tmp)[1:.k]])
  })

  merged = rbind(X_train, X_val)
  merged_y = c(y_train, y_val)
  imp_tst = sapply(1:nrow(X_test), function (row_i) {
    tmp = sapply(1:nrow(merged), function (row_j) {
```

```
        sqrt(sum((X_test[row_i, ] - merged[row_j, ]) ^ 2))
      })
      mean(merged_y[order(tmp)[1:.k]])
  })

  list(res_tr, res_val, imp_tst)
}


impute_knn_dist <- function (X_train, y_train, X_val, y_val, X_test, .k) {
  res_tr = sapply(1:nrow(X_train), function (row_i) {
    tmp = sapply(1:nrow(X_train), function (row_j) {
      if (row_i != row_j) { sqrt(sum((X_train[row_i, ] - X_train[row_j, ]) ^ 2)) }
      else { 20 }
    })
    mean((min(tmp[order(tmp)[1:.k]]) / tmp[order(tmp)[1:.k]]) ^ 2 * y_train[order(tmp)[1:.k]])
  })

  res_val = sapply(1:nrow(X_val), function (row_i) {
    tmp = sapply(1:nrow(X_train), function (row_j) {
      sqrt(sum((X_val[row_i, ] - X_train[row_j, ]) ^ 2))
    })
    mean((min(tmp[order(tmp)[1:.k]]) / tmp[order(tmp)[1:.k]]) ^ 3 * y_train[order(tmp)[1:.k]])
  })

  merged = rbind(X_train, X_val)
  merged_y = c(y_train, y_val)
  imp_tst = sapply(1:nrow(X_test), function (row_i) {
    tmp = sapply(1:nrow(merged), function (row_j) {
      sqrt(sum((X_test[row_i, ] - merged[row_j, ]) ^ 2))
    })
    mean(merged_y[order(tmp)[1:.k]])
  })

  list(res_tr, res_val, imp_tst)
}


eval_model <- function (.data, .fun, ...) {
  .scorer <- function (ytrue, ypred) {
    sqrt(mean((ytrue - ypred) ^ 2))
  }

  res_tr = c()
  res_val = c()
  for (i in 1:length(.data$train)) {
    tmp = .fun(.data$X[.data$train[[i]], ],
              .data$y[.data$train[[i]]],
              .data$X[.data$val[[i]], ],
              .data$y[.data$val[[i]]],
              .data$X_test, ...)
    res_tr  = c(res_tr,  .scorer(tmp[[1]], .data$y[.data$train[[i]]]))
    res_val = c(res_val, .scorer(tmp[[2]], .data$y[.data$val[[i]]]))
```

```
  }
  list(tr = res_tr, val = res_val)
}


aa_data = generate_data(aa_pair_mat, .cv = 10)
imp_res = list()
imp_res[["mean_1"]] = eval_model(aa_data, impute_knn_mean, .k = 1)
imp_res[["mean_3"]] = eval_model(aa_data, impute_knn_mean, .k = 3)
imp_res[["mean_5"]] = eval_model(aa_data, impute_knn_mean, .k = 5)
imp_res[["dist_1"]] = eval_model(aa_data, impute_knn_dist, .k = 1)
imp_res[["dist_3"]] = eval_model(aa_data, impute_knn_dist, .k = 3)
imp_res[["dist_5"]] = eval_model(aa_data, impute_knn_dist, .k = 5)
imp_res = melt(imp_res)
colnames(imp_res) = c("val", "data", "app")
qplot(x = data, y = val, fill = app, data = imp_res, geom = "boxplot") + theme_bw()
```
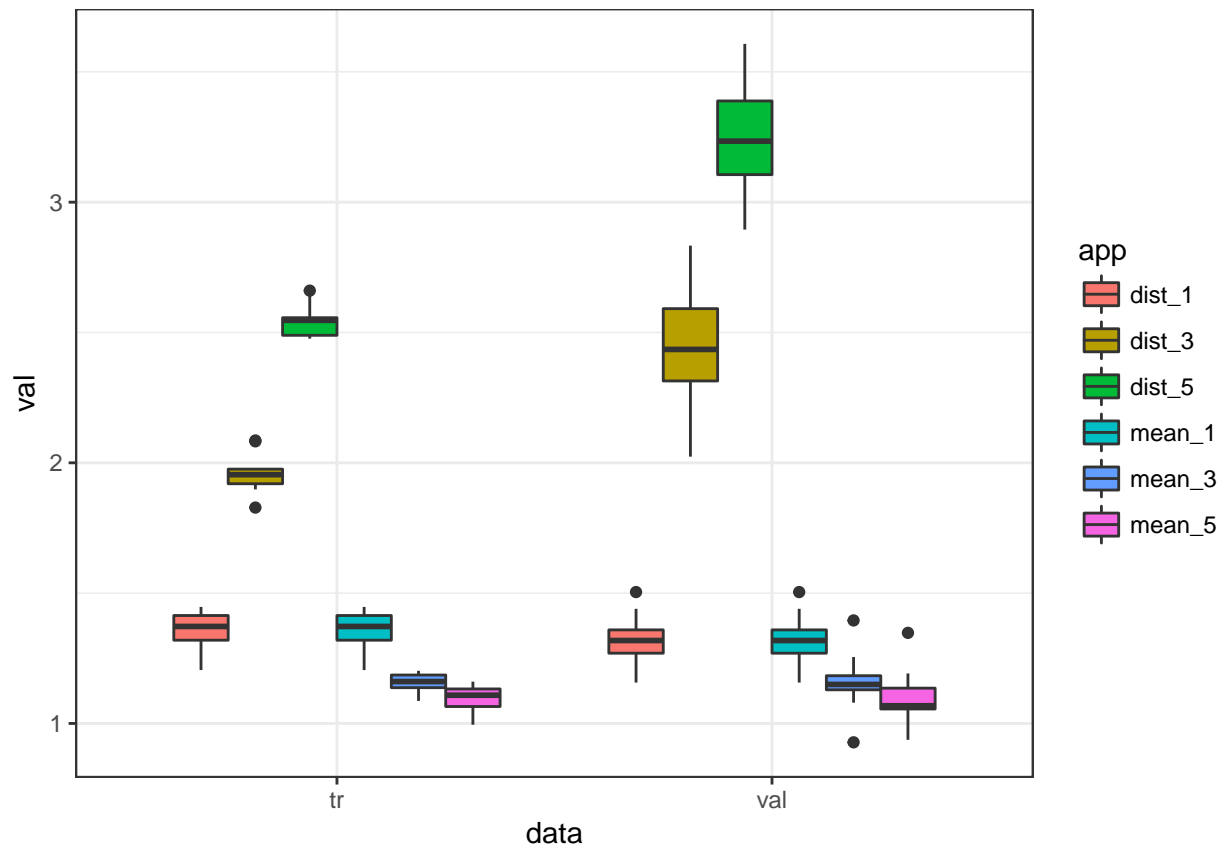


```
melted = melt(aa_pair_mat)[melt(upper.tri(aa_pair_mat, T))[,3],]
melted[,1] = as.character(melted[,1])
melted[,2] = as.character(melted[,2])
test_data = melted[is.na(melted[,3]), ]

imputed = impute_knn_mean(aa_data$X[aa_data$train[[1]], ],
                          aa_data$y[aa_data$train[[1]]],
                          aa_data$X[aa_data$val[[1]], ],
                          aa_data$y[aa_data$val[[1]]],
                          aa_data$X_test, .k = 5)[[3]]
```

```
aa_pair_mat_imp = aa_pair_mat
aa_pair_vec =
for (r in 1:nrow(test_data)) {
  aa_pair_mat_imp[test_data[r,1], test_data[r,2]] = imputed[r]
  aa_pair_mat_imp[test_data[r,2], test_data[r,1]] = imputed[r]

  ind = intersect(which(df.aa.coef[,2] == test_data[r,1]), which(df.aa.coef[,3] == test_data[r,2]))
  if (length(ind) == 0) {
    ind = intersect(which(df.aa.coef[,2] == test_data[r,2]), which(df.aa.coef[,3] == test_data[r,1]))
  }
  df.aa.coef[ind, 1] = imputed[r]
}

# update the df.aa.coef.diag
df.aa.coef.diag = df.aa.coef
df.aa.coef.diag$aa_pair = with(df.aa.coef.diag,
  as.factor(ifelse(as.character(aa_tcr) < as.character(aa_antigen), paste(aa_tcr, aa_antigen, sep = "_")
df.aa.coef.diag = df.aa.coef.diag %>% select(aa_pair, coef)
df.aa.coef.diag = df.aa.coef.diag[!duplicated(df.aa.coef.diag[,1]),]

p2 = heatmap.2(aa_pair_mat_imp,
          hclustfun = function(x) hclust(x, method = "ward.D2"),
          RowSideColors = aa_colors[rownames(aa_pair_mat)],
          ColSideColors = aa_colors[colnames(aa_pair_mat)],
          trace = "none",
          #breaks = seq(-16, -7, length.out = 101),
          col=colorRampPalette(brewer.pal(9, 'YlGnBu'))(32))
```
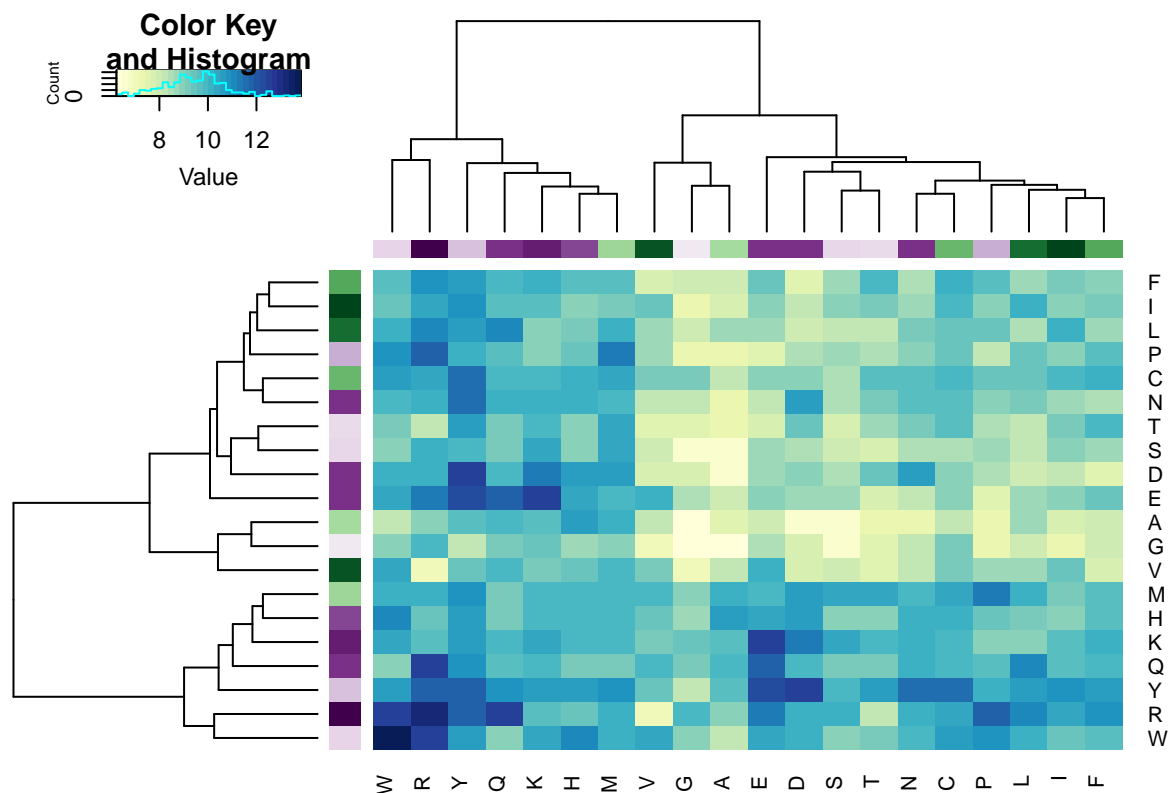
Discard distant amino acids

Append amino acid distance coefficients

```
df.pred = df.pred[df.aa.coef.diag, on = "aa_pair"]
df.pred.trimmed = df.pred[distance_CA <= 15] # discard AAs that are too far away for training
```

Contact energies

Compute mean GROMACS energies

```
df.energies = df[contact == T, .(energy.mean = mean(ifelse(energy > 0 , 0, energy))), by = "aa_pair"]
df.energies$aa_tcr = str_split_fixed(as.character(df.energies$aa_pair), "_", 2)[, 1]
df.energies$aa_antigen = str_split_fixed(as.character(df.energies$aa_pair), "_", 2)[, 2]

df.energies.tmp = df.energies
df.energies.tmp$aa_tcr = df.energies$aa_antigen
df.energies.tmp$aa_antigen = df.energies$aa_tcr

df.energies = rbind(df.energies, df.energies.tmp) %>% unique()

# transform to matrix and plot heatmap.2

aa_pair_energy_mat = dcast(df.energies, aa_tcr ~ aa_antigen, value.var = "energy.mean", fun.aggregate =
rownames(aa_pair_energy_mat) = aa_pair_energy_mat$aa_tcr
aa_pair_energy_mat$aa_tcr = NULL
aa_pair_energy_mat = as.matrix(aa_pair_energy_mat)

heatmap.2(aa_pair_energy_mat,
          hclustfun = function(x) hclust(x, method = "ward.D2"),
          RowSideColors = aa_colors[rownames(aa_pair_mat)],
          ColSideColors = aa_colors[colnames(aa_pair_mat)],
          trace = "none",
          breaks = seq(-40, 2, length.out = 33),
          col=rev(colorRampPalette(brewer.pal(9, 'YlOrRd'))(32)))
```
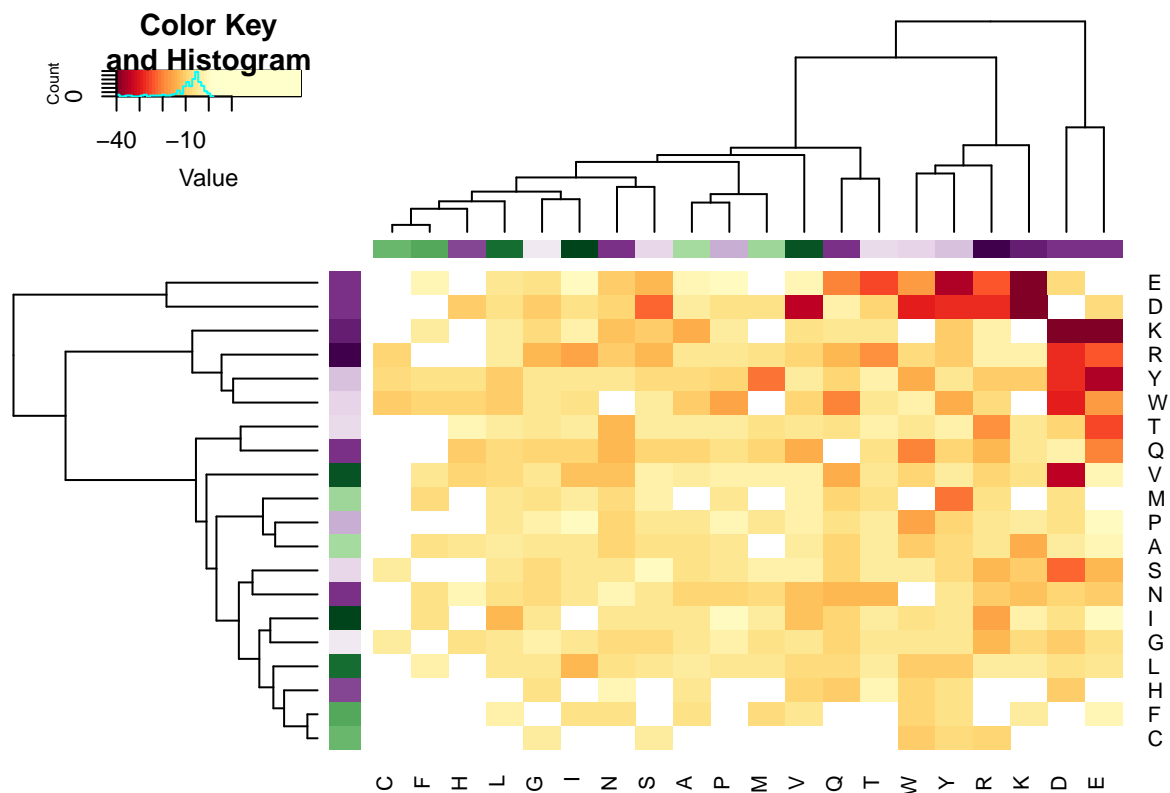
Impute contact energies

```r
aa_data = generate_data(aa_pair_energy_mat, .cv = 10)
imp_res = list()
imp_res[["mean_1"]] = eval_model(aa_data, impute_knn_mean, .k = 1)
imp_res[["mean_3"]] = eval_model(aa_data, impute_knn_mean, .k = 3)
imp_res[["mean_5"]] = eval_model(aa_data, impute_knn_mean, .k = 5)
imp_res[["dist_1"]] = eval_model(aa_data, impute_knn_dist, .k = 1)
imp_res[["dist_3"]] = eval_model(aa_data, impute_knn_dist, .k = 3)
imp_res[["dist_5"]] = eval_model(aa_data, impute_knn_dist, .k = 5)
imp_res = melt(imp_res)
colnames(imp_res) = c("val", "data", "app")
qplot(x = data, y = val, fill = app, data = imp_res, geom = "boxplot") + theme_bw()
```
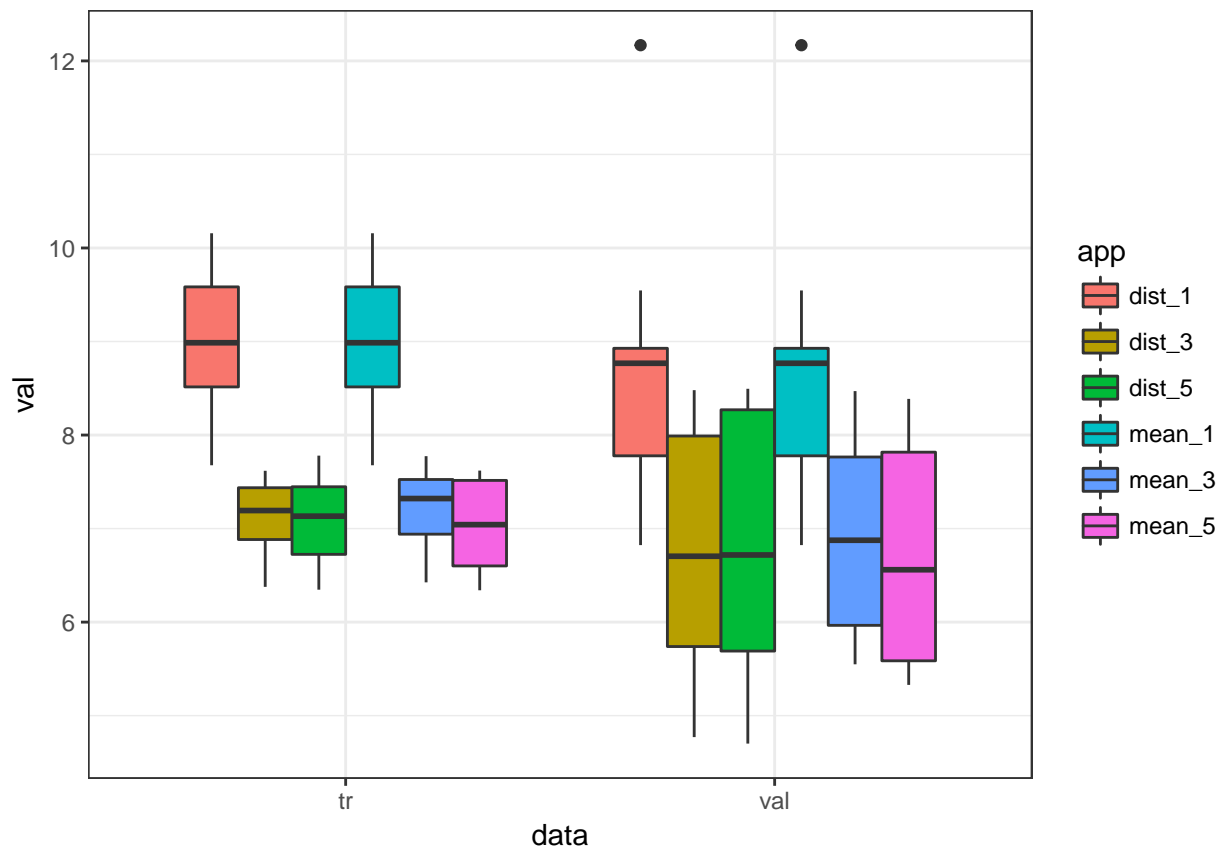
Imputed energies

```r
melted = melt(aa_pair_energy_mat)[melt(upper.tri(aa_pair_energy_mat, T))[,3],]
melted[,1] = as.character(melted[,1])
melted[,2] = as.character(melted[,2])
test_data = melted[is.na(melted[,3]), ]

imputed = impute_knn_mean(aa_data$X[aa_data$train[[1]], ],
                          aa_data$y[aa_data$train[[1]]],
                          aa_data$X[aa_data$val[[1]], ],
                          aa_data$y[aa_data$val[[1]]],
                          aa_data$X_test, .k = 5)[[3]]

test_data[,3] = imputed

aa_pair_energy_mat_imp = aa_pair_energy_mat
for (r in 1:nrow(test_data)) {
  aa_pair_energy_mat_imp[test_data[r,1], test_data[r,2]] = test_data[r,3]
  aa_pair_energy_mat_imp[test_data[r,2], test_data[r,1]] = test_data[r,3]
  ind = intersect(which(df.energies[,3] == test_data[r,1]), which(df.energies[,4] == test_data[r,2]))
  if (length(ind) == 0) {
    ind = intersect(which(df.energies[,3] == test_data[r,2]), which(df.energies[,4] == test_data[r,1]))
  }
  if (length(ind) == 0) {
    df.energies = rbind(df.energies, list(paste0(test_data[r,1], "_", test_data[r,2]), test_data[r,3],
```

```
  } else {
    if (!is.na(df.energies[[ind, 2]])) {
      cat("error!!\n")
      df.energies[[ind, 2]] = imputed[r]
    } else {
      df.energies[[ind, 2]] = imputed[r]
    }
  }
}

heatmap.2(aa_pair_energy_mat_imp,
          hclustfun = function(x) hclust(x, method = "ward.D2"),
          RowSideColors = aa_colors[rownames(aa_pair_mat)],
          ColSideColors = aa_colors[colnames(aa_pair_mat)],
          trace = "none",
          breaks = seq(-40, 2, length.out = 33),
          col=rev(colorRampPalette(brewer.pal(9, 'YlOrRd'))(32)))
```
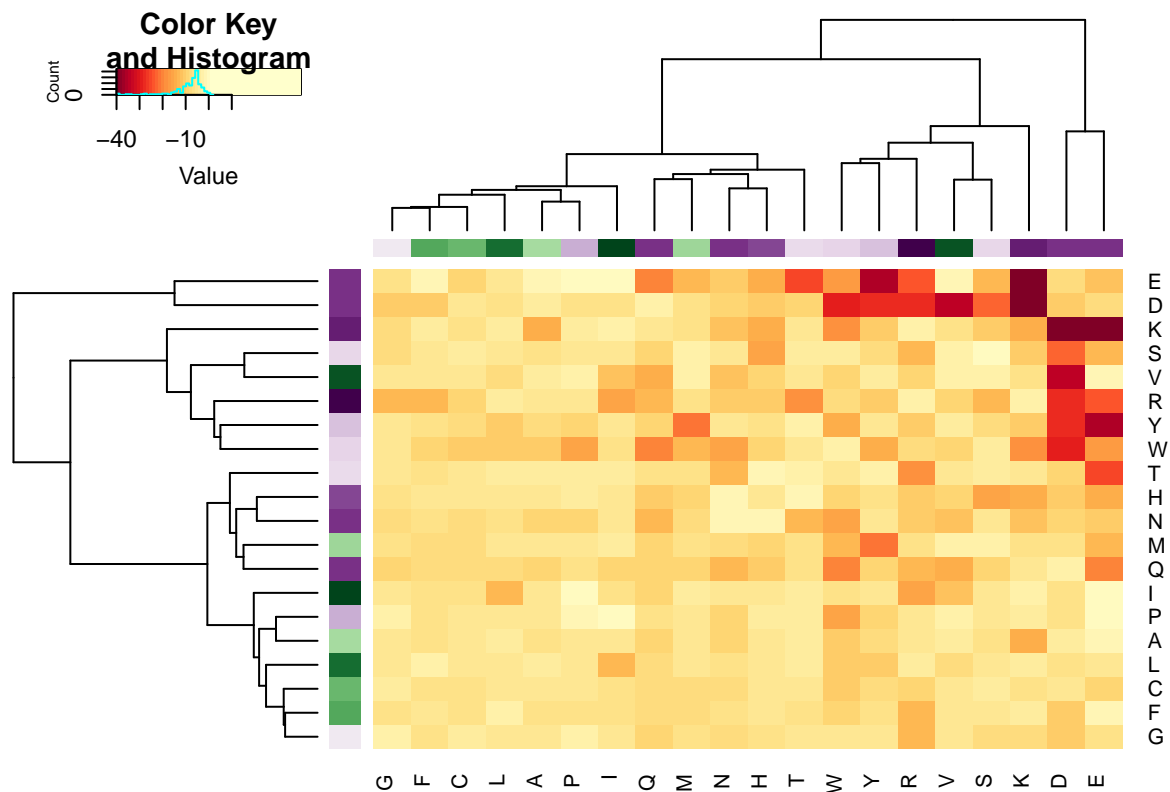

Building and testing a predictor

Final generalized linear model to fit contacts.

```
# Train on a trimmed dataset
contact_glm = glm(contact ~ distance_CA.m + coef, family = binomial(), data = df.pred.trimmed)

summary(contact_glm)
```

```
## 
## Call:
## glm(formula = contact ~ distance_CA.m + coef, family = binomial(),
##     data = df.pred.trimmed)
## 
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1147  -0.4895  -0.3105  -0.1687   3.7756
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.64487    0.19586  -3.293 0.000993 ***
## distance_CA.m -0.47602    0.01237 -38.473  < 2e-16 ***
## coef           0.47024    0.01857  25.320  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 13232  on 18848  degrees of freedom
## Residual deviance: 10798  on 18846  degrees of freedom
## AIC: 10804
## 
## Number of Fisher Scoring iterations: 6
```

```r
df.pred.trimmed$p = predict(contact_glm, df.pred.trimmed, type="response")
df.pred$p = predict(contact_glm, df.pred, type="response")

df.pred = df.pred[df.energies, on = .(aa_tcr, aa_antigen)]
```

Save model for further evaluation:

```r
save(df.ca.mean, df.aa.coef, contact_glm, df.energies, file="eval/model_simple.RData")
write.table(df.ca.mean, "eval/ca_dist_mean.txt", sep="\t", quote=F, row.names = F)
write.table(df.aa.coef, "eval/aa_pairwise_contact_coef.txt", sep="\t", quote=F, row.names = F)
write.table(df.energies, "eval/aa_pairwise_energy.txt", sep="\t", quote=F, row.names = F)
```

Check accuracy

General

ROC curve

```r
rocobj = plot.roc(as.data.frame(df.pred.trimmed)[,"contact"], df.pred.trimmed$p, ci=T)
```
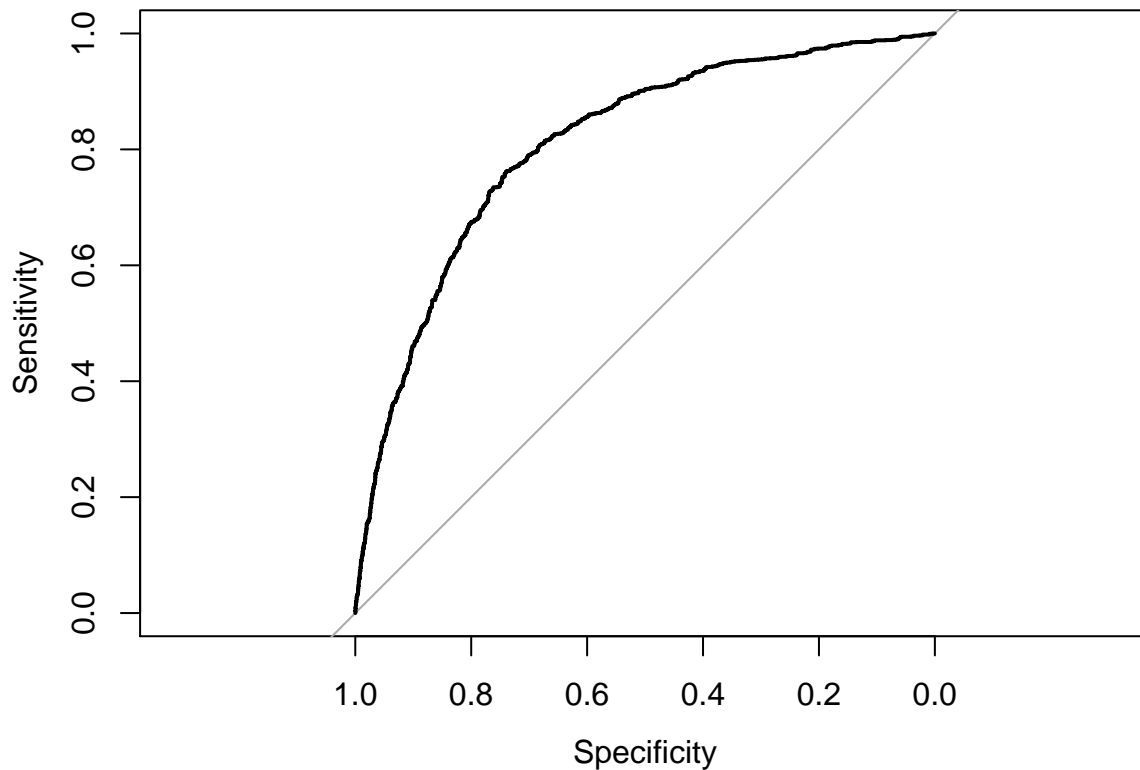
```
rocobj
```

```
##
## Call:
## plot.roc.default(x = as.data.frame(df.pred.trimmed)[, "contact"],     predictor = df.pred.trimmed$p,
##
## Data: df.pred.trimmed$p in 16735 controls (as.data.frame(df.pred.trimmed)[, "contact"] FALSE) < 2114
## Area under the curve: 0.8093
## 95% CI: 0.7997-0.8188 (DeLong)
```

Compute true and estimated total number of contacts

```
df.pred.contsum = df.pred.trimmed[, .(total_contacts = sum(contact), pred_contacts = sum(p, na.rm=T)),

ggplot(df.pred.contsum, aes(x=pred_contacts, y=total_contacts, color = tcr_region)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +

  scale_x_continuous(limits=c(0,15)) +
  scale_y_continuous(limits=c(0,15)) +
  scale_color_brewer(palette = "Set1") +
  facet_wrap(~len_tcr) +
  theme_bw()
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```
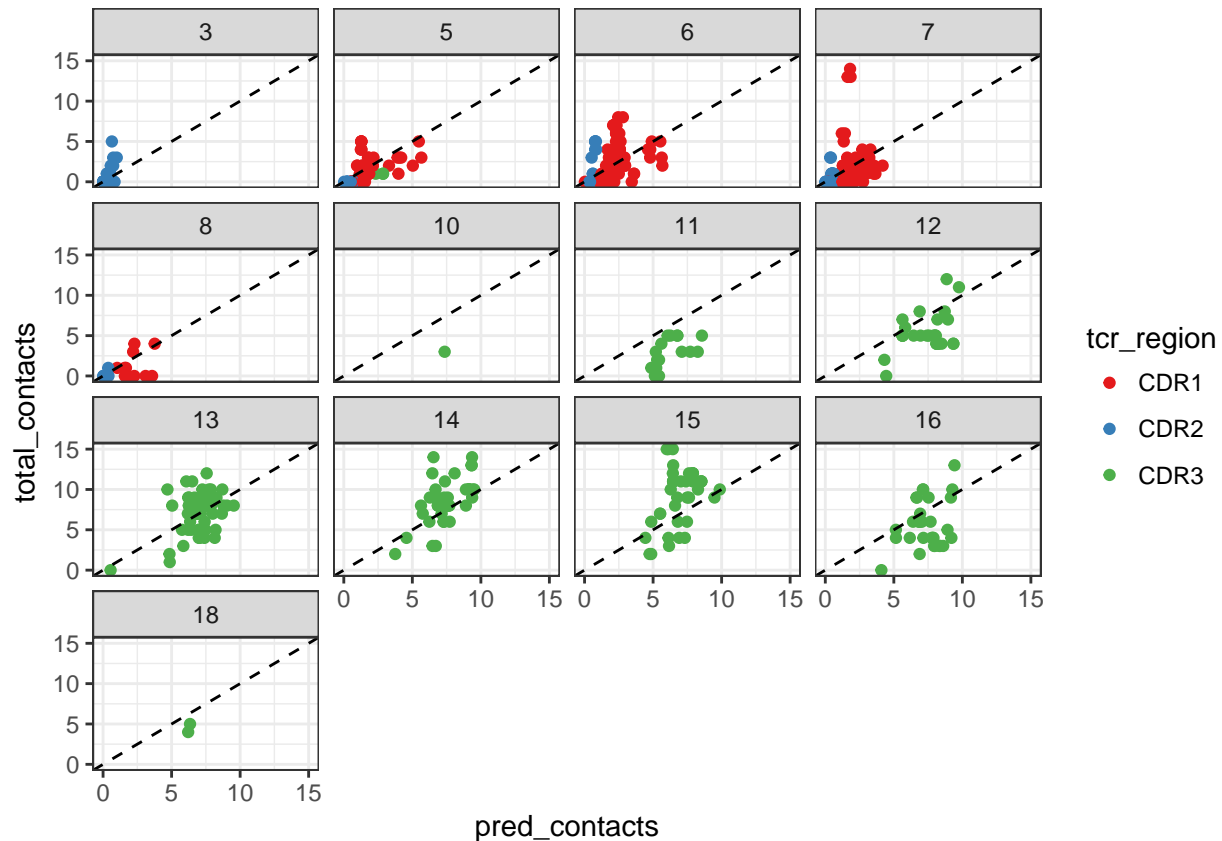
```r
lfit = lm(total_contacts ~ pred_contacts + len_tcr + tcr_region - 1, df.pred.contsum)
summary(lfit)
```

```
##
## Call:
## lm(formula = total_contacts ~ pred_contacts + len_tcr + tcr_region -
##     1, data = df.pred.contsum)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8081 -1.2684 -0.1850  0.7269 11.7638
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## pred_contacts   0.82937    0.08851   9.370  < 2e-16 ***
## len_tcr         0.27863    0.06042   4.612 4.81e-06 ***
## tcr_regionCDR1 -1.21234    0.41211  -2.942 0.003379 **
## tcr_regionCDR2 -1.00515    0.28795  -3.491 0.000514 ***
## tcr_regionCDR3 -2.78366    0.93926  -2.964 0.003151 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.311 on 650 degrees of freedom
## Multiple R-squared:  0.7806, Adjusted R-squared:  0.7789
## F-statistic: 462.5 on 5 and 650 DF,  p-value: < 2.2e-16
```

```r
anova(lfit)
```

```
## Analysis of Variance Table
##
## Response: total_contacts
##                 Df  Sum Sq Mean Sq  F value    Pr(>F)
## pred_contacts    1 12223.5 12223.5 2288.8595 < 2.2e-16 ***
## len_tcr          1    58.3    58.3   10.9194  0.001004 **
## tcr_region       3    68.9    23.0    4.2996  0.005132 **
## Residuals      650  3471.3     5.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Examples from the train data

Check for a couple of antigens, GLCTLVAML

```r
df.pred.glc = df.pred %>%
  filter(antigen_seq == "GLCTLVAML") %>%
  droplevels()

ggplot(df.pred.glc, aes(x=pos_tcr, y=pos_antigen)) +
  geom_tile(fill=NA) +
  geom_label(aes(label=paste(aa_tcr, aa_antigen, sep=":"), fill = p), cex=1.3) +
  geom_point(aes(color=contact)) +
  scale_x_continuous(breaks=0:20) +
  scale_y_continuous(breaks=0:20) +
  #scale_fill_gradient("P",
  #                    low="white", high="#045a8d") +
  scale_color_manual(values = c(NA, "red")) +
  scale_fill_gradientn(colors=colorRampPalette(brewer.pal(9, 'YlGnBu'))(32)) +
  facet_grid(tcr_chain ~ tcr_region, scales="free", space="free") +
  theme_bw()
```

```
## Warning: Removed 371 rows containing missing values (geom_point).
```
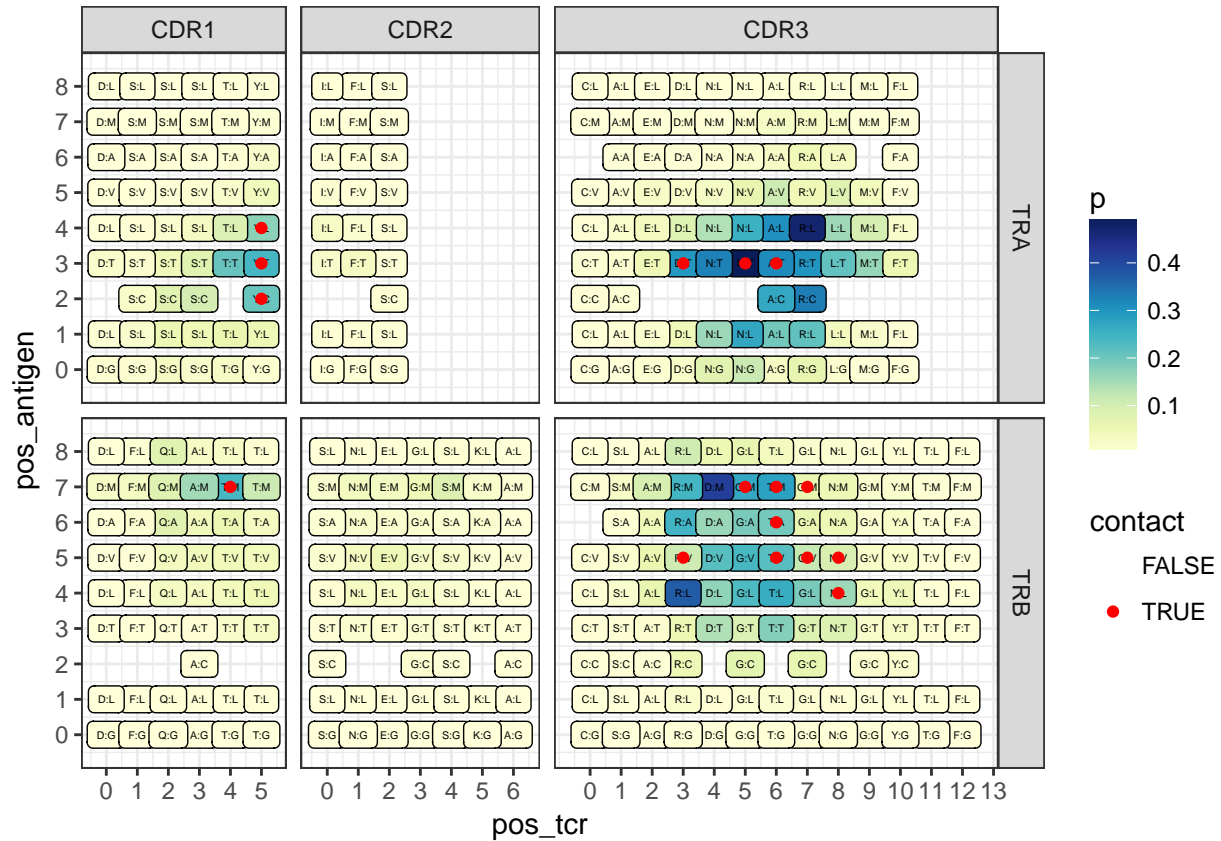
and LLFGYPVAV

```r
df.pred.llf = df.pred %>%
  filter(antigen_seq == "LLFGYPVAV") %>%
  droplevels()

ggplot(df.pred.llf, aes(x=pos_tcr, y=pos_antigen)) +
  geom_tile(fill=NA) +
  geom_label(aes(label=paste(aa_tcr, aa_antigen, sep=":"), fill = p), cex=1.3) +
  geom_point(aes(color=contact)) +
  scale_x_continuous(breaks=0:20) +
  scale_y_continuous(breaks=0:20) +
  #scale_fill_gradient("P",
  #                    low="white", high="#045a8d") +
  scale_color_manual(values = c(NA, "red")) +
  scale_fill_gradientn(colors=colorRampPalette(brewer.pal(9, 'YlGnBu'))(32)) +
  facet_grid(tcr_chain ~ tcr_region, scales="free", space="free") +
  theme_bw()
```

```
## Warning: Removed 444 rows containing missing values (geom_point).
```
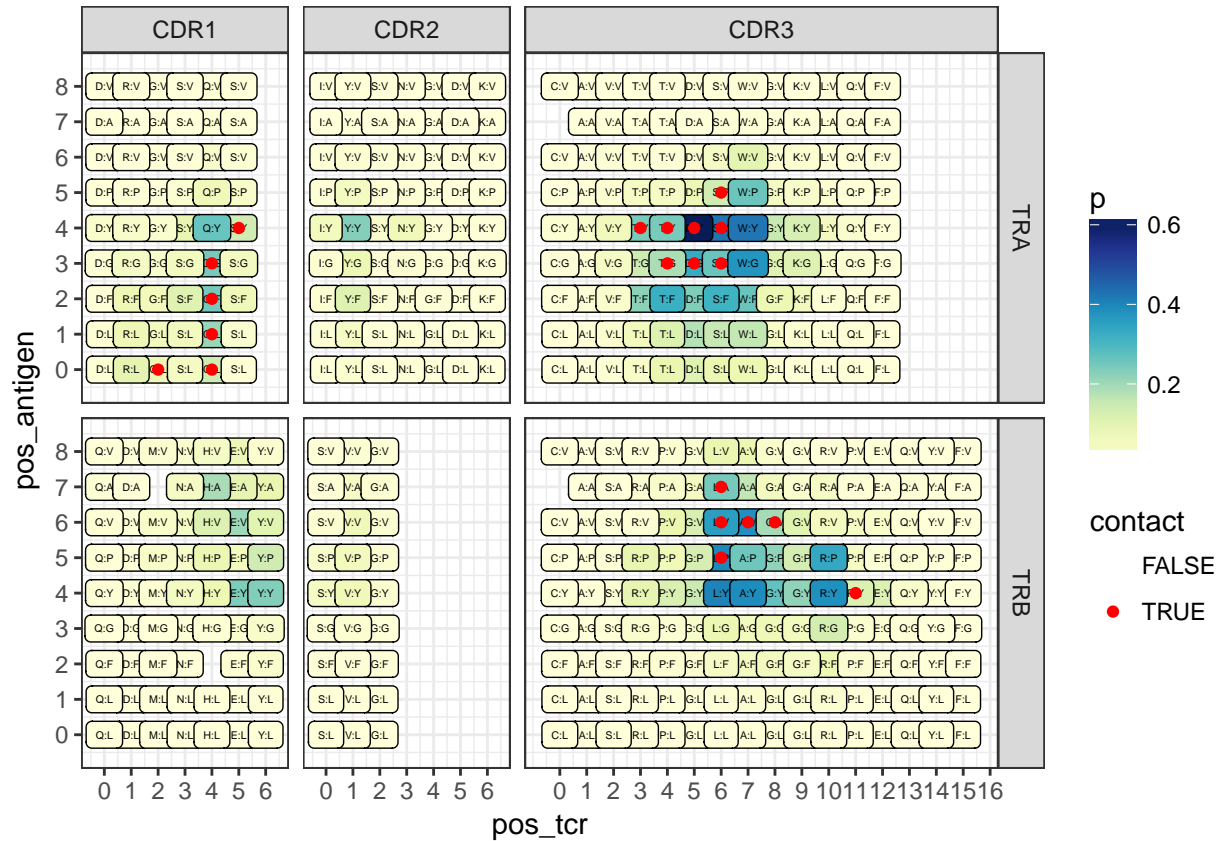
Independent validation

Computing contact map from fitted model for a specified TCR:pMHC setup

```r
compute_contact_map = function(mhc_type, tcr_chain, tcr_region, cdr_seq, ag_seq, id = "tmp") {
  cdr_seq = as.character(cdr_seq)
  ag_seq = as.character(ag_seq)
  df.cdr = data.frame(aa_tcr = strsplit(cdr_seq, "")[[1]],
                      pos_tcr = 1:nchar(cdr_seq) - 1)
  df.ag = data.frame(aa_antigen = strsplit(ag_seq, "")[[1]],
                     pos_antigen = 1:nchar(ag_seq) - 1)

  df.pairs = expand.grid(df.cdr$pos_tcr, df.ag$pos_antigen)
  colnames(df.pairs) = c("pos_tcr", "pos_antigen")

  df.pairs = merge(df.pairs, df.cdr)
  df.pairs = merge(df.pairs, df.ag)

  df.pairs$aa_pair = with(df.pairs,
                    as.factor(ifelse(as.character(aa_tcr) < as.character(aa_antigen),
                                paste(aa_tcr, aa_antigen, sep = "_"), paste(aa_antigen, aa_tcr,

  df.pairs$mhc_type = mhc_type
  df.pairs$tcr_chain = tcr_chain
  df.pairs$tcr_region = tcr_region
```

```r
    df.pairs$len_tcr = nchar(cdr_seq)
    df.pairs$len_antigen = nchar(ag_seq)

    df.pairs$pos_tcr_c = with(df.pairs, pos_tcr - round(len_tcr / 2))
    df.pairs$pos_antigen_c = with(df.pairs, pos_antigen - round(len_antigen / 2))

    df.pairs$id = id # ! id can be anything to group the complex, e.g. clonotype id in sample

    df.res = merge(df.pairs %>% select(id, mhc_type, tcr_chain, tcr_region, pos_tcr_c, pos_antigen_c, aa_
               df.pred %>% select(mhc_type, tcr_chain, tcr_region, pos_tcr_c, pos_antigen_c, aa_pair, p, energy
               unique()

    df.res$p[is.na(df.res$p)] = 0
    df.res$energy.mean[is.na(df.res$energy.mean)] = 0

    df.res
}
```

Testing - example 1 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2681418/ - engeneered peptide

```r
cm.tmp = compute_contact_map("MHCI","TRA","CDR3","CAVDSATSGTYKYIF","ILAKFLHWL","WT")
cm.tmp = rbind(cm.tmp, compute_contact_map("MHCI","TRA","CDR3","CAVDSATSGTYKYIF","ILAAFLHWL","No binding
cm.tmp = rbind(cm.tmp, compute_contact_map("MHCI","TRA","CDR3","CAVDSATSGTYKYIF","GLGGGGGGV","Mock"))
cm.tmp = rbind(cm.tmp, compute_contact_map("MHCI","TRA","CDR3","CAVDSATALPYGYIF","ILAKFLHWL", "EnhancedV

print(cm.tmp %>% group_by(id) %>% summarize(contacts = sum(p), energy = sum(p * energy.mean)))
```

```
## # A tibble: 4 × 3
##          id contacts     energy
##       <chr>    <dbl>      <dbl>
## 1  Enhanced 8.071243  -57.25581
## 2      Mock 3.268209  -17.85175
## 3 No binding 5.029306  -24.22092
## 4        WT 7.931040  -55.63909
```
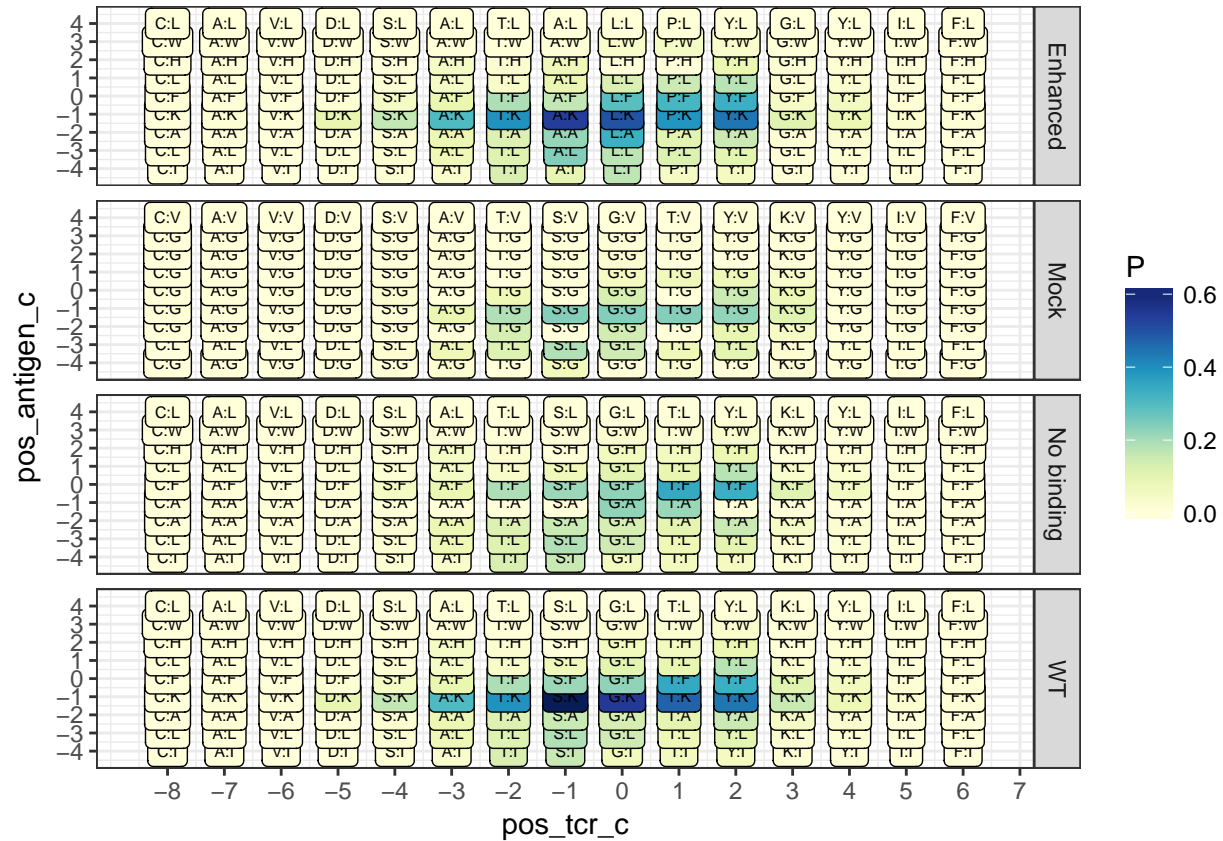
```r
ggplot(cm.tmp, aes(x=pos_tcr_c, y=pos_antigen_c)) +
  geom_tile(fill=NA) +
  geom_label(aes(label=paste(aa_tcr, aa_antigen, sep=":"), fill = p), cex=2) +
  scale_x_continuous(breaks=-8:9) +
  scale_y_continuous(breaks=-5:5) +
  scale_fill_gradientn("P", colors=colorRampPalette(brewer.pal(9, 'YlGnBu'))(32)) +
  facet_grid(id~., scales="free", space="free") +
  theme_bw()
```

```
ggplot(cm.tmp, aes(x=pos_tcr_c, y=pos_antigen_c)) +
  geom_tile(fill=NA) +
  geom_label(aes(label=paste(aa_tcr, aa_antigen, sep=":"), fill = p * energy.mean), cex=2) +
  scale_x_continuous(breaks=-8:9) +
  scale_y_continuous(breaks=-5:5) +
  scale_fill_gradientn("E", colors=rev(colorRampPalette(brewer.pal(9, 'YlOrRd'))(32))) +
  facet_grid(id~., scales="free", space="free") +
  theme_bw()
```
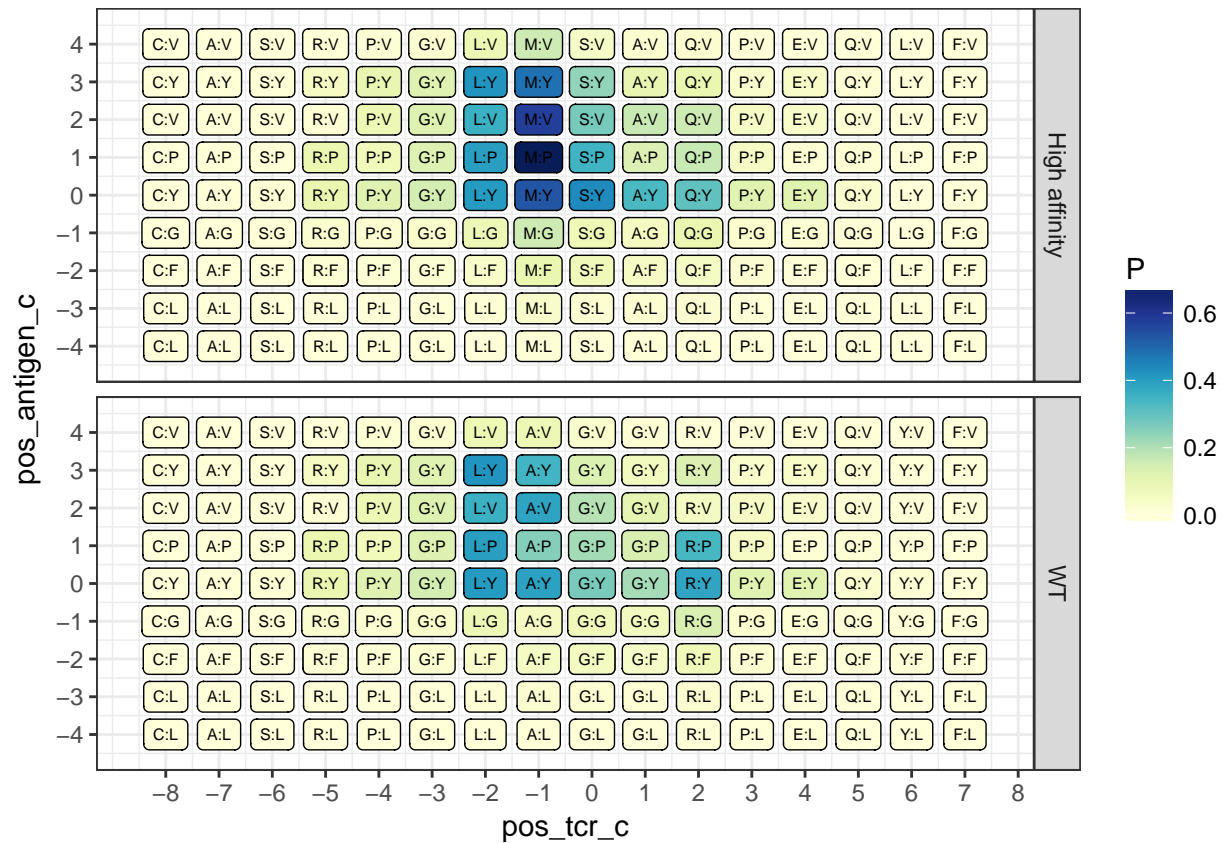
Testing - example 2 http://www.nature.com/articles/ncomms6223, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3049343/ - high affinity Tax mutant

```r
cm.tmp = compute_contact_map("MHCI","TRB","CDR3","CASRPGLAGGRPEQYF","LLFGYPVYV","WT")
cm.tmp = rbind(cm.tmp, compute_contact_map("MHCI","TRB","CDR3","CASRPGLMSAQPEQLF","LLFGYPVYV","High affi

print(cm.tmp %>% group_by(id) %>% summarize(contacts = sum(p), energy = sum(p * energy.mean)))
```
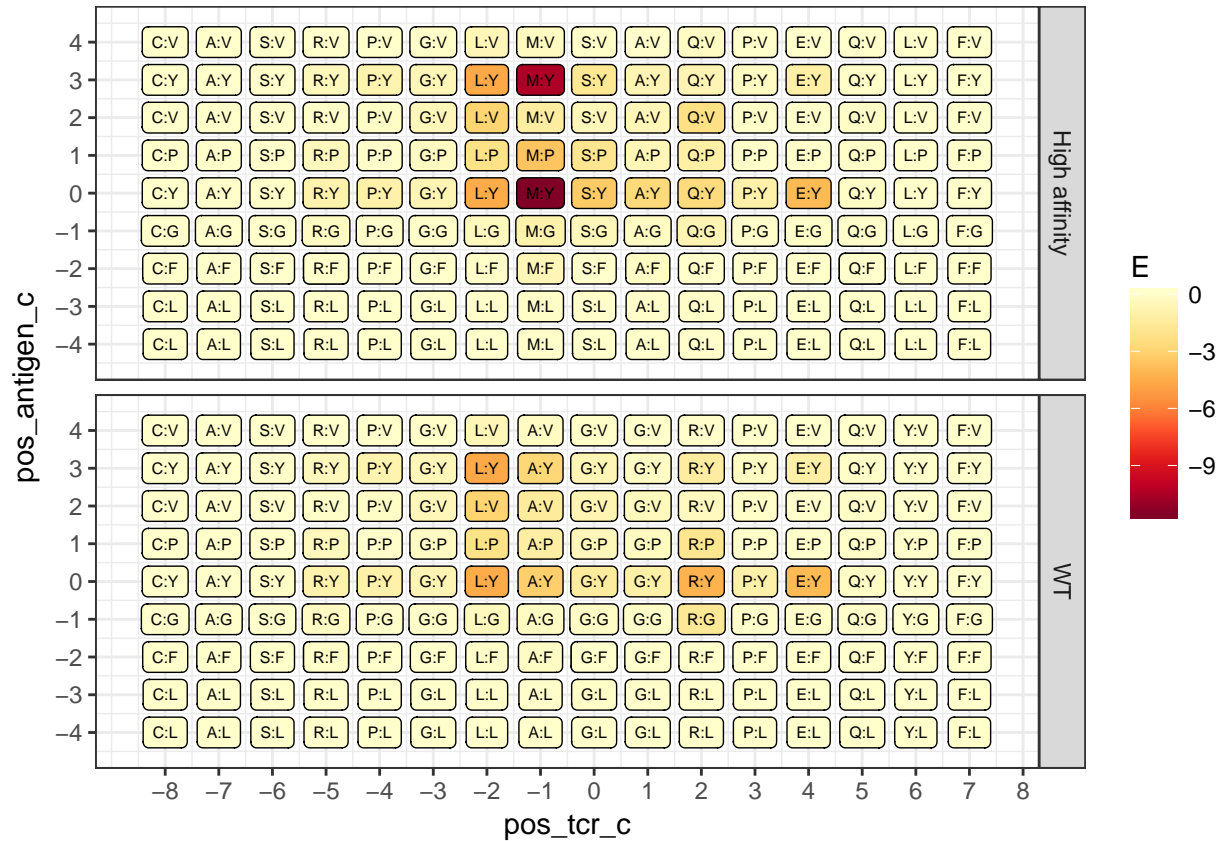
```
## # A tibble: 2 × 3
##                id contacts      energy
##             <chr>    <dbl>       <dbl>
## 1 High affinity 9.617205 -82.79314
## 2            WT 7.991283 -58.06737
```

```r
ggplot(cm.tmp, aes(x=pos_tcr_c, y=pos_antigen_c)) +
  geom_tile(fill=NA) +
  geom_label(aes(label=paste(aa_tcr, aa_antigen, sep=":"), fill = p), cex=2) +
  scale_x_continuous(breaks=-8:9) +
  scale_y_continuous(breaks=-5:5) +
  scale_fill_gradientn("P", colors=colorRampPalette(brewer.pal(9, 'YlGnBu'))(32)) +
  facet_grid(id~., scales="free", space="free") +
  theme_bw()
```

```r
ggplot(cm.tmp, aes(x=pos_tcr_c, y=pos_antigen_c)) +
  geom_tile(fill=NA) +
  geom_label(aes(label=paste(aa_tcr, aa_antigen, sep=":"), fill = p * energy.mean), cex=2) +
  scale_x_continuous(breaks=-8:9) +
  scale_y_continuous(breaks=-5:5) +
  scale_fill_gradientn("E", colors=rev(colorRampPalette(brewer.pal(9, 'YlOrRd'))(32))) +
  facet_grid(id~., scales="free", space="free") +
  theme_bw()
```

Testing - example 3 same as above - comparing Tax specific vs Tax and MART specific vs MART + cross-comparison. We compare CDR3beta of A6 (wild-type Tax-specific variant) and a MART-specific TCR derived from A6 by direct evolution.

```
cm.tmp.1 = compute_contact_map("MHCI","TRB","CDR3","CASRPGLAGGRPEQYF","LLFGYPVYV","Tax-tcr:Tax")
cm.tmp.1 = rbind(cm.tmp.1, compute_contact_map("MHCI","TRB","CDR3","CASRPGLAGGRPEQYF","AAGIGILTV","Tax-t
cm.tmp.1 = rbind(cm.tmp.1, compute_contact_map("MHCI","TRB","CDR3","CASRPGWMAGGVELYF","LLFGYPVYV","MART-
cm.tmp.1 = rbind(cm.tmp.1, compute_contact_map("MHCI","TRB","CDR3","CASRPGWMAGGVELYF","AAGIGILTV","MART-

print(cm.tmp.1 %>% group_by(id) %>% summarize(contacts = sum(p), energy = sum(p * energy.mean)))
```
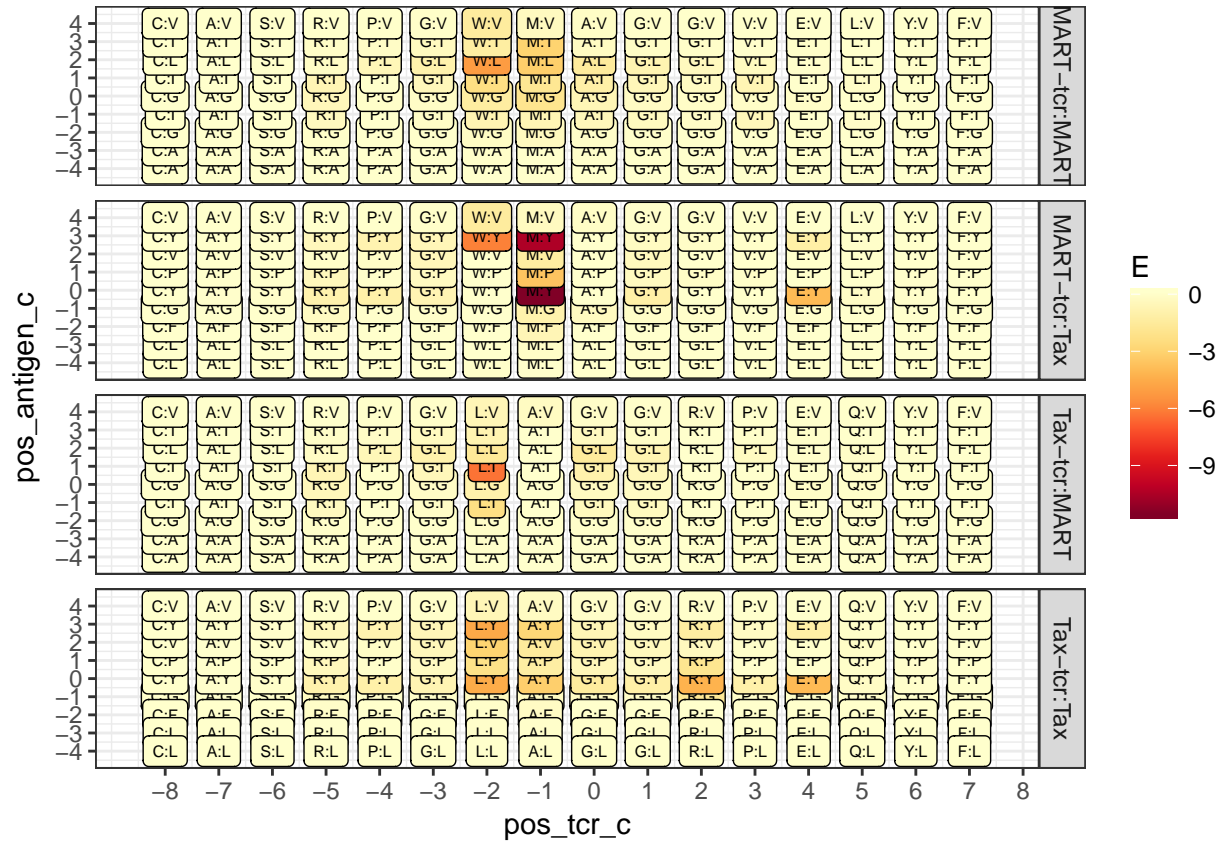
```
## # A tibble: 4 × 3
##               id contacts      energy
##            <chr>    <dbl>       <dbl>
## 1 MART-tcr:MART 7.195740 -41.83845
## 2  MART-tcr:Tax 5.844139 -54.34775
## 3  Tax-tcr:MART 4.369679 -27.50443
## 4   Tax-tcr:Tax 7.991283 -58.06737
```

```
ggplot(cm.tmp.1, aes(x=pos_tcr_c, y=pos_antigen_c)) +
  geom_tile(fill=NA) +
  geom_label(aes(label=paste(aa_tcr, aa_antigen, sep=":"), fill = p), cex=2) +
  scale_x_continuous(breaks=-8:9) +
  scale_y_continuous(breaks=-5:5) +
  scale_fill_gradientn("P", colors=colorRampPalette(brewer.pal(9, 'YlGnBu'))(32)) +
  facet_grid(id~., scales="free", space="free") +
  theme_bw()
```
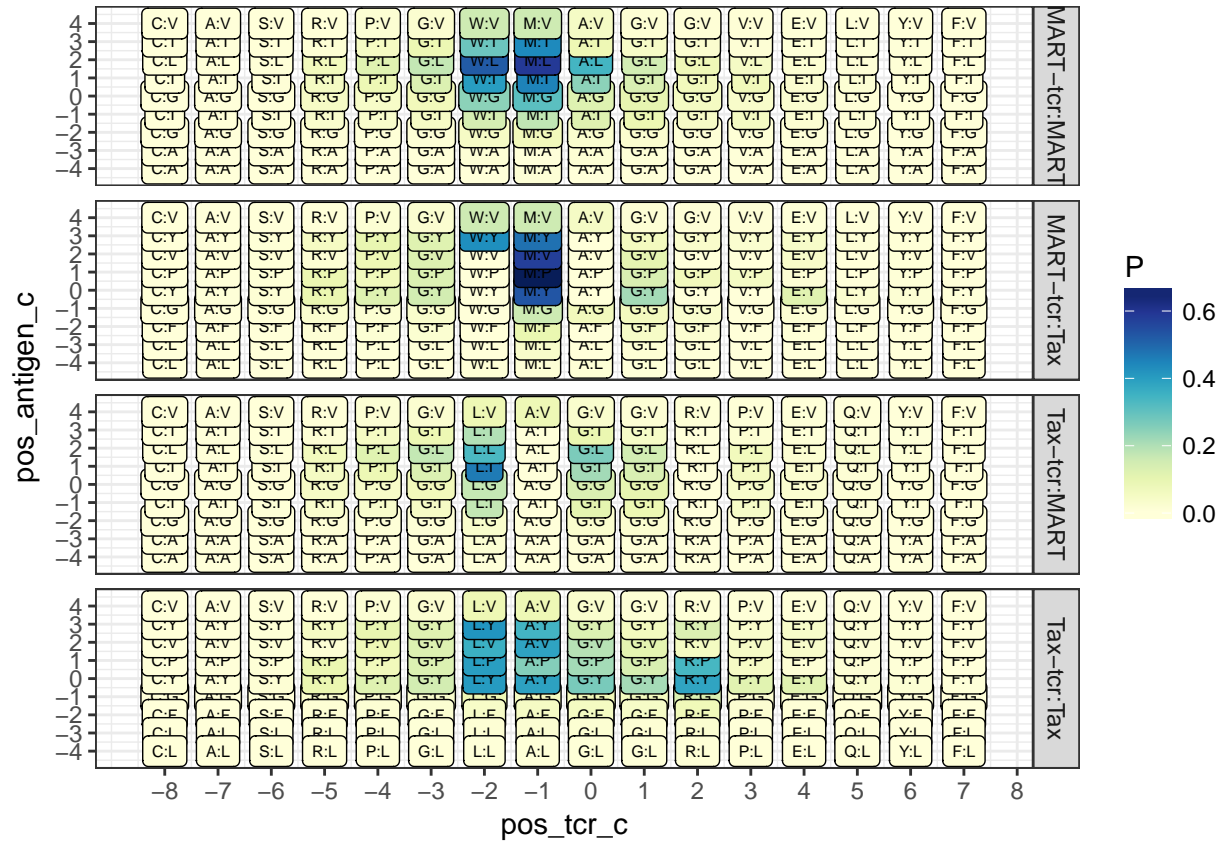
```
ggplot(cm.tmp.1, aes(x=pos_tcr_c, y=pos_antigen_c)) +
  geom_tile(fill=NA) +
  geom_label(aes(label=paste(aa_tcr, aa_antigen, sep=":"), fill = p * energy.mean), cex=2) +
  scale_x_continuous(breaks=-8:9) +
  scale_y_continuous(breaks=-5:5) +
  scale_fill_gradientn("E", colors=rev(colorRampPalette(brewer.pal(9, 'YlOrRd'))(32))) +
  facet_grid(id~., scales="free", space="free") +
  theme_bw()
```

Same as above for alpha chains

```r
cm.tmp.2 = compute_contact_map("MHCI","TRA","CDR3","CAVVVTTDSWGKLQF","LLFGYPVYV","Tax-tcr:Tax")
cm.tmp.2 = rbind(cm.tmp.2, compute_contact_map("MHCI","TRA","CDR3","CAVVVTTDSWGKLQF","AAGIGILTV","Tax-tc
cm.tmp.2 = rbind(cm.tmp.2, compute_contact_map("MHCI","TRA","CDR3","CAVVVTKYSWGKLQF","LLFGYPVYV","MART-t
cm.tmp.2 = rbind(cm.tmp.2, compute_contact_map("MHCI","TRA","CDR3","CAVVVTKYSWGKLQF","AAGIGILTV","MART-t

print(cm.tmp.2 %>% group_by(id) %>% summarize(contacts = sum(p), energy = sum(p * energy.mean)))
```
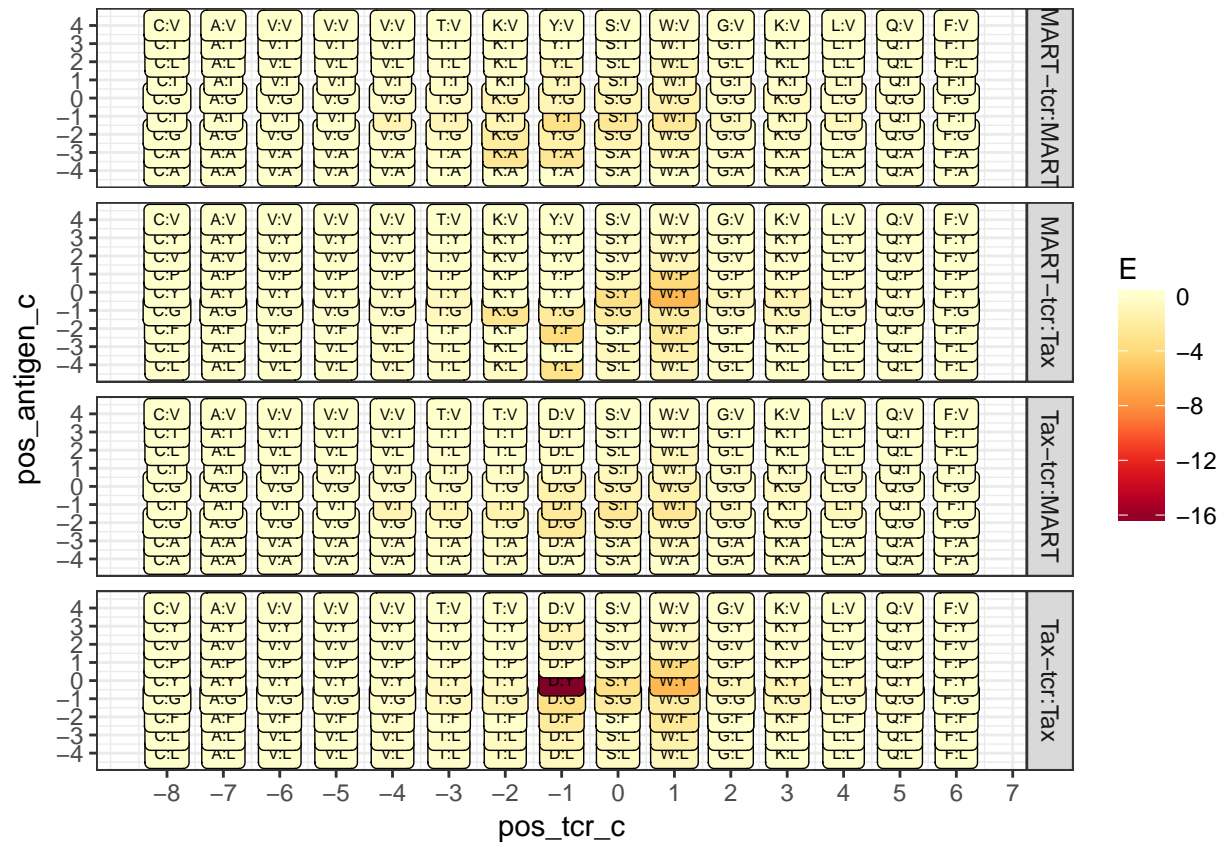
```
## # A tibble: 4 × 3
##               id contacts     energy
##            <chr>    <dbl>      <dbl>
## 1 MART-tcr:MART 6.699041 -41.26890
## 2  MART-tcr:Tax 6.845970 -47.59605
## 3  Tax-tcr:MART 5.593917 -34.06064
## 4   Tax-tcr:Tax 7.738554 -64.15860
```

```r
ggplot(cm.tmp.2, aes(x=pos_tcr_c, y=pos_antigen_c)) +
  geom_tile(fill=NA) +
  geom_label(aes(label=paste(aa_tcr, aa_antigen, sep=":"), fill = p), cex=2) +
  scale_x_continuous(breaks=-8:9) +
  scale_y_continuous(breaks=-5:5) +
  scale_fill_gradientn("P", colors=colorRampPalette(brewer.pal(9, 'YlGnBu'))(32)) +
  facet_grid(id~., scales="free", space="free") +
  theme_bw()
```
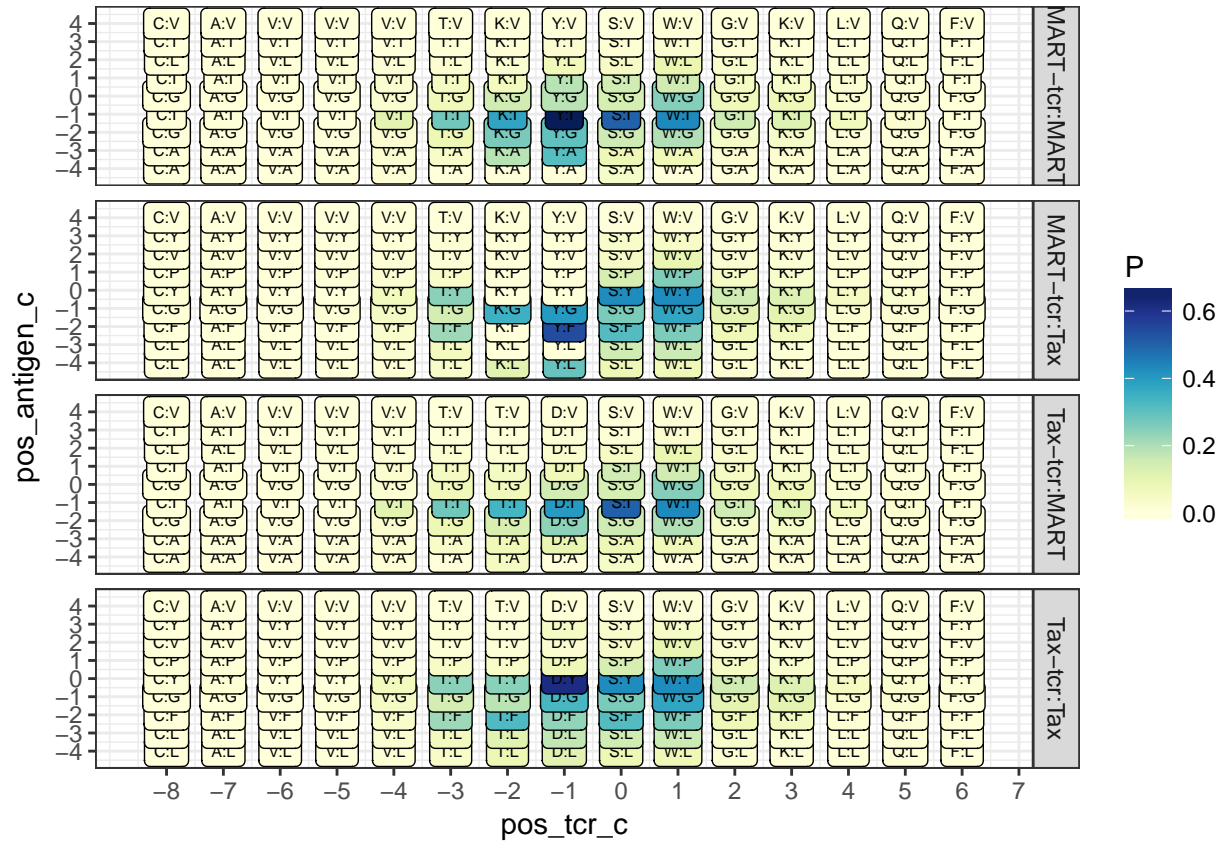
```
ggplot(cm.tmp.2, aes(x=pos_tcr_c, y=pos_antigen_c)) +
  geom_tile(fill=NA) +
  geom_label(aes(label=paste(aa_tcr, aa_antigen, sep=":"), fill = p * energy.mean), cex=2) +
  scale_x_continuous(breaks=-8:9) +
  scale_y_continuous(breaks=-5:5) +
  scale_fill_gradientn("E", colors=rev(colorRampPalette(brewer.pal(9, 'YlOrRd'))(32))) +
  facet_grid(id~., scales="free", space="free") +
  theme_bw()
```

Summarize alpha and beta chains

```
print(rbind(cm.tmp.1, cm.tmp.2) %>% group_by(id) %>% summarize(contacts = sum(p), energy = sum(p * energ
```

```
## # A tibble: 4 × 3
##              id  contacts      energy
##           <chr>     <dbl>       <dbl>
## 1 MART-tcr:MART 13.894781   -83.10735
## 2  MART-tcr:Tax 12.690110  -101.94380
## 3  Tax-tcr:MART  9.963595   -61.56507
## 4   Tax-tcr:Tax 15.729837  -122.22597
```