

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
```

```
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
```

```
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

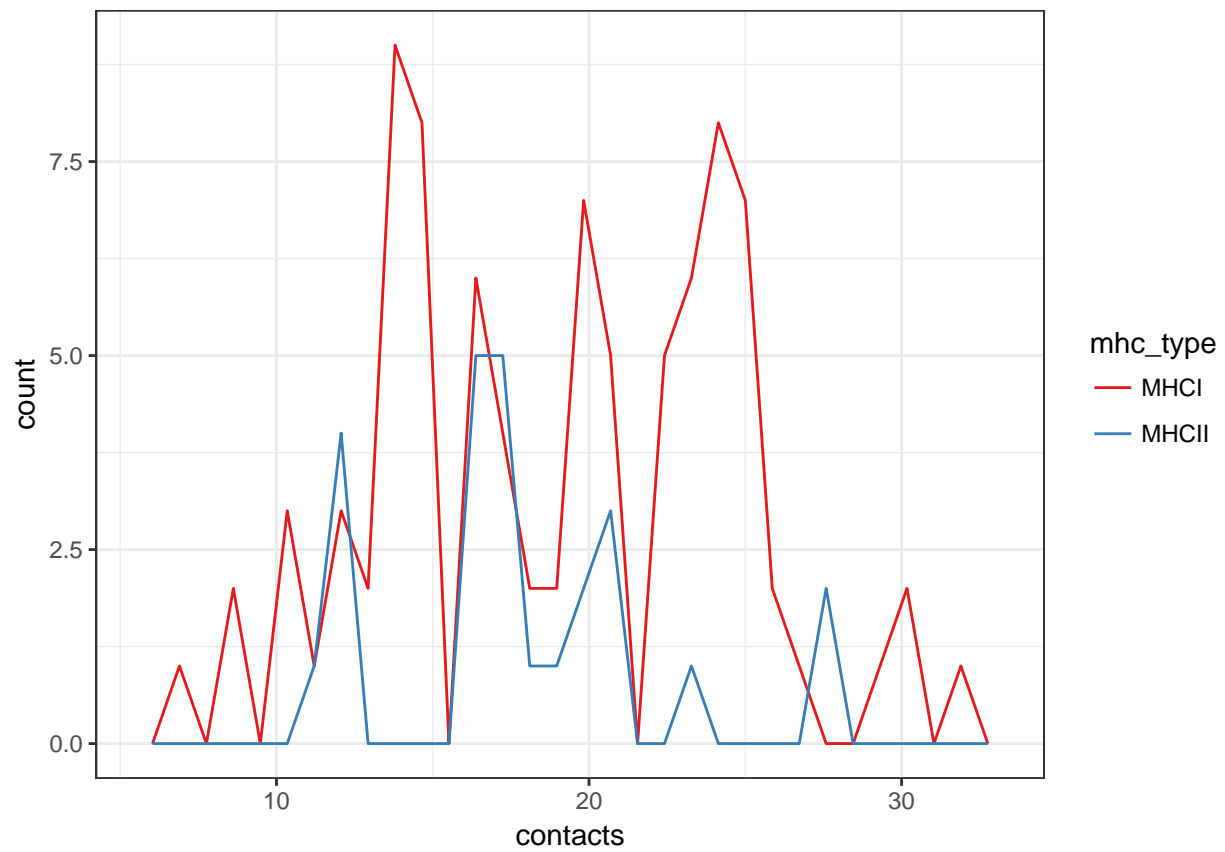
Contacts by MHC, chain and CDR

```
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]

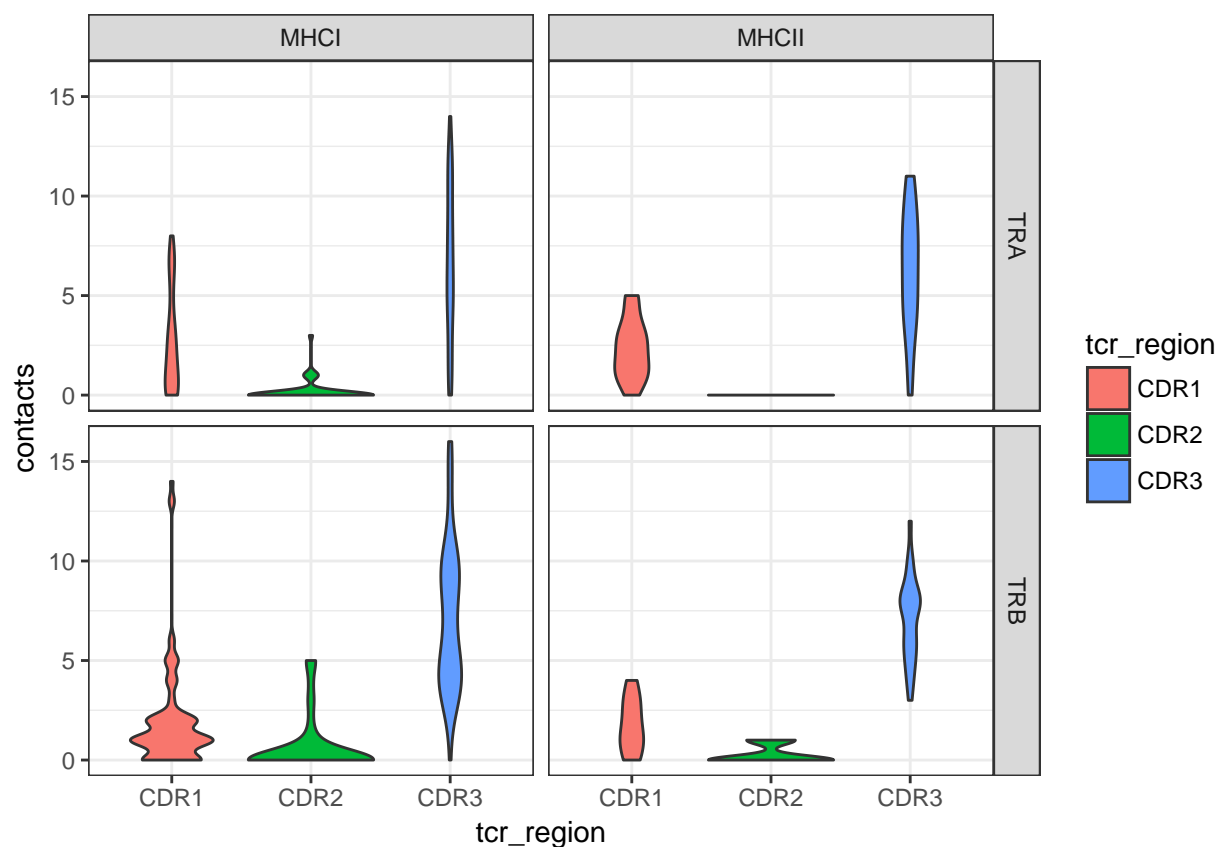
ggplot(df.contact.sum.pdb, aes(contacts, color = mhc_type)) +
  geom_freqpoly() +
  scale_color_brewer(palette = "Set1") +
  theme_bw()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
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
```

```
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
```

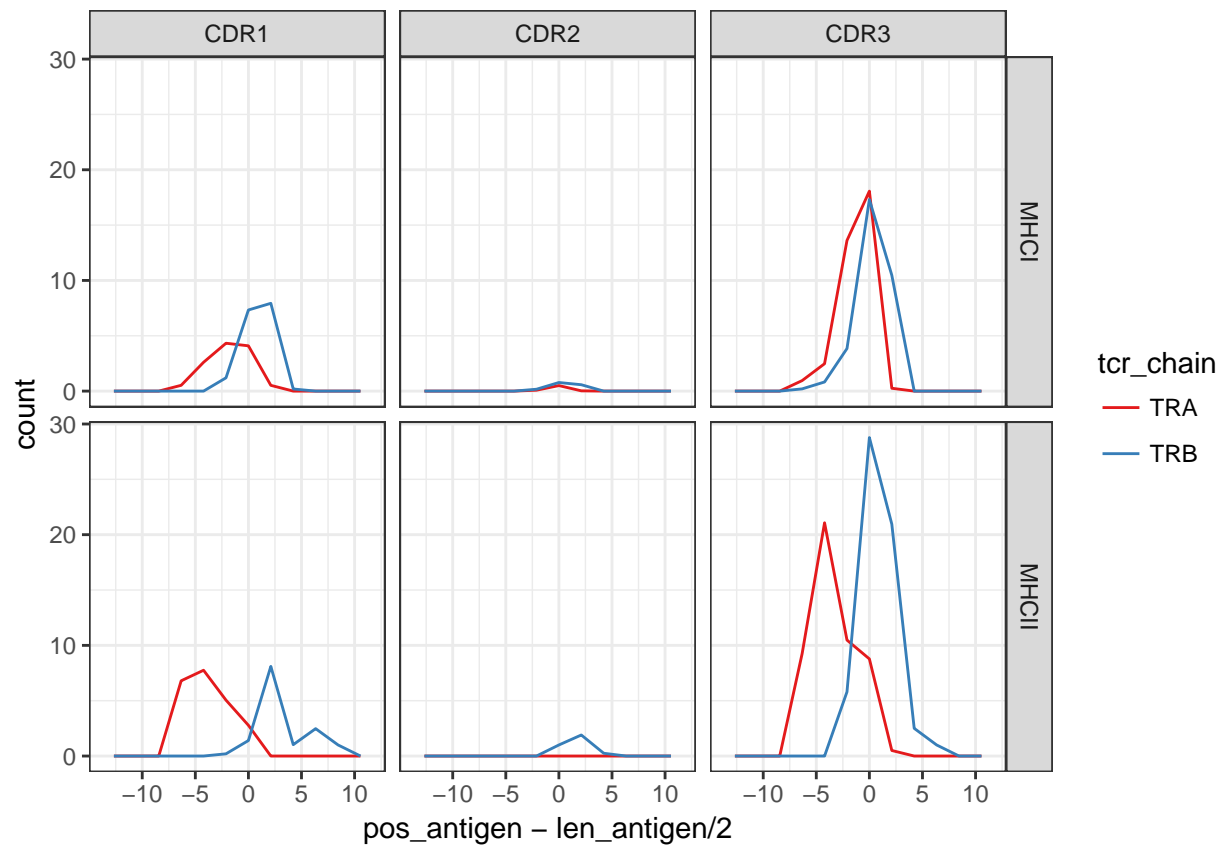
Filter TCRs with no contacts

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

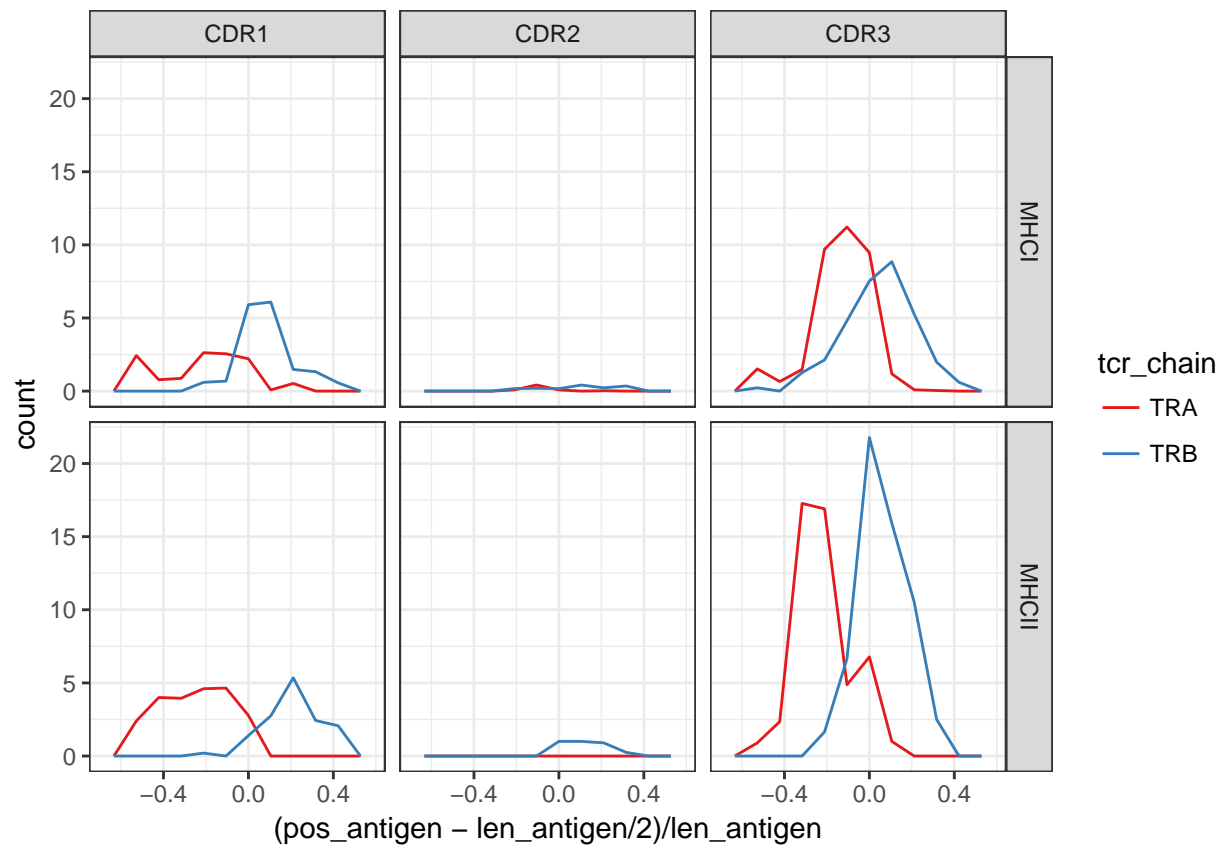
Contact distribution on antigen

```
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 = tcr_chain)) +
  geom_freqpoly(bins=10) +
  facet_grid(mhc_type~tcr_region) +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```



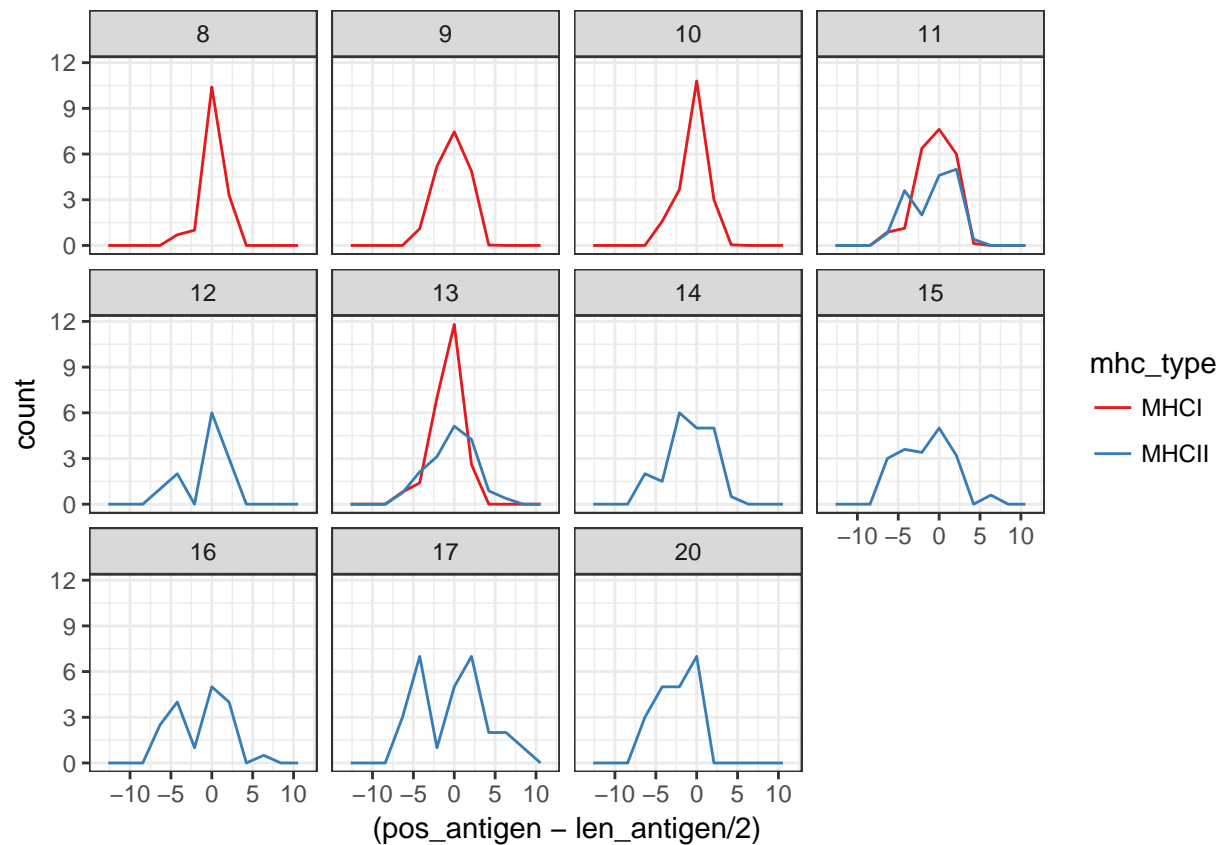
```
ggplot(df.contact.dist.ag, aes(x = (pos_antigen - len_antigen / 2) / len_antigen, weight = contacts / tcr_chain)) +
  geom_freqpoly(bins=10) +
  facet_grid(mhc_type~tcr_region) +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```



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)]

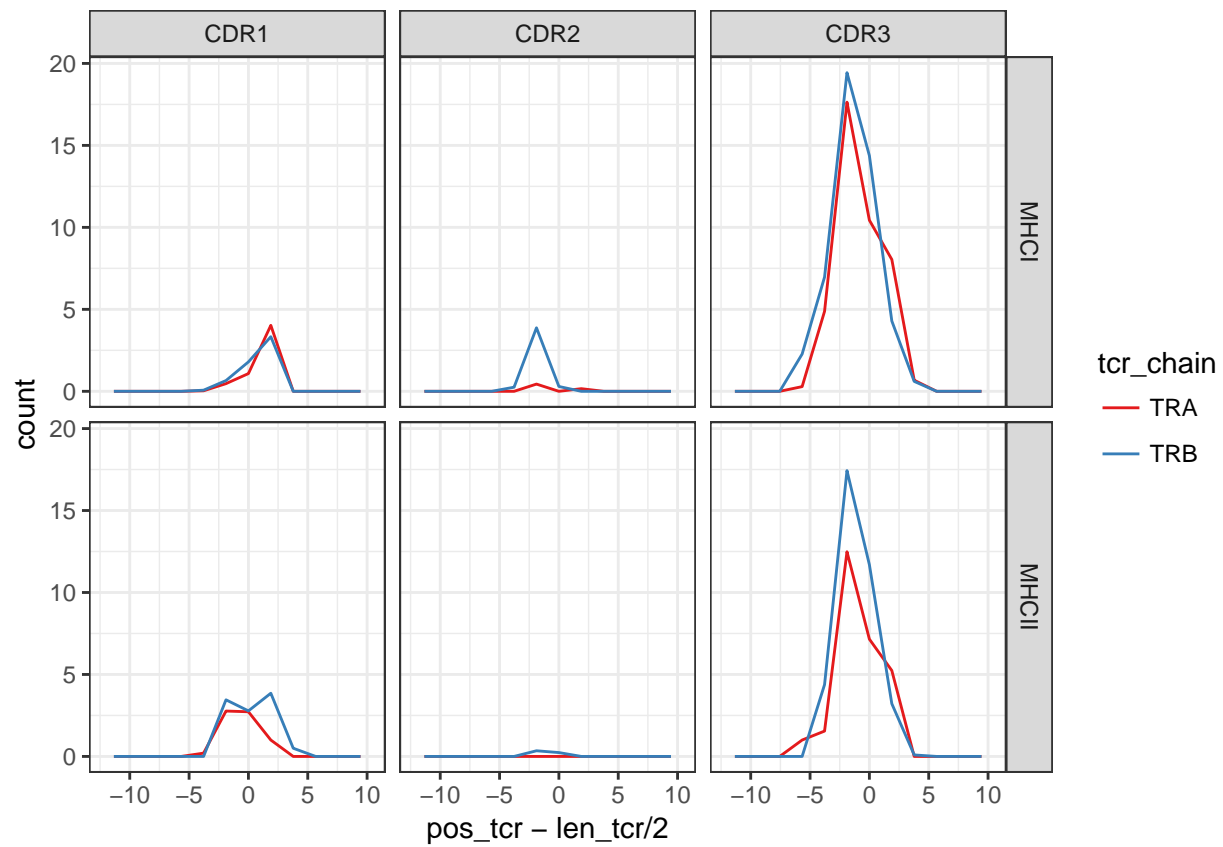
ggplot(df.contact.dist.ag.len, aes(x = (pos_antigen - len_antigen / 2), group = paste(len_antigen, mhc_type),
  weight = contacts / total.pdb, color = mhc_type)) +
  geom_freqpoly(bins=10) +
  facet_wrap(~len_antigen) +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```

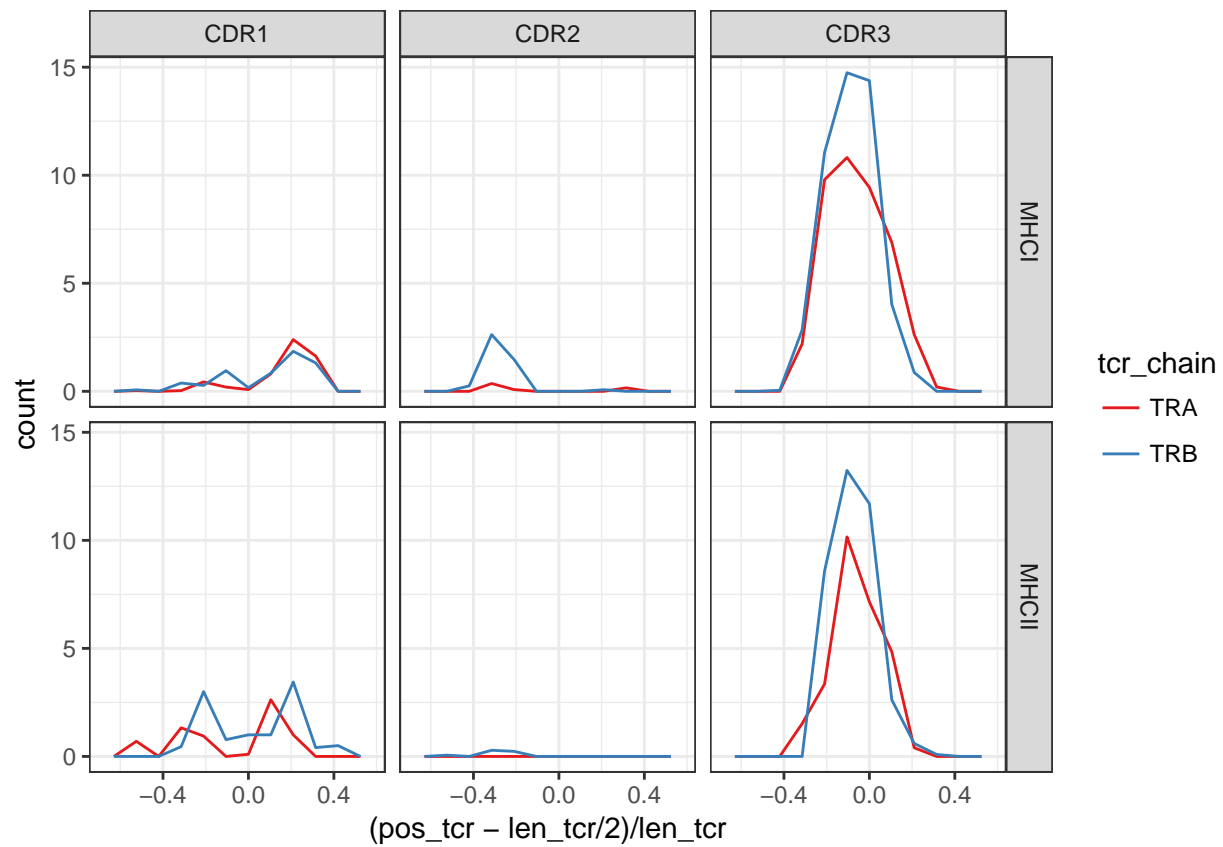
Contact distribution on TCR

```
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_ch)) +
  geom_freqpoly(bins=10) +
  facet_grid(mhc_type~tcr_region) +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```



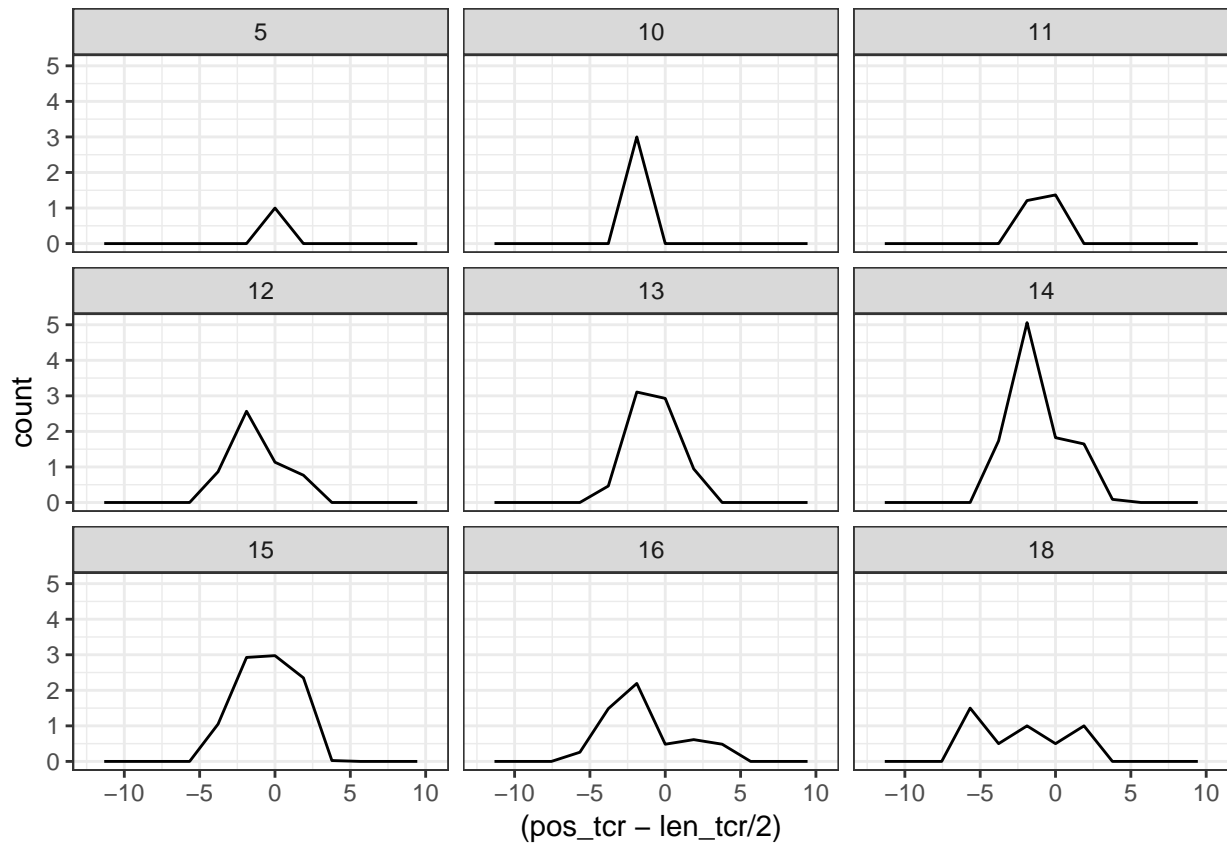
```
ggplot(df.contact.dist.tcr, aes(x = (pos_tcr - len_tcr / 2) / len_tcr, weight = contacts / total.pdb, color = tcr_chain)) +
  geom_freqpoly(bins=10) +
  facet_grid(mhc_type~tcr_region) +
  scale_color_brewer(palette = "Set1") +
  theme_bw()
```



vs CDR3 len

```
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 / total.pdb)) +
  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

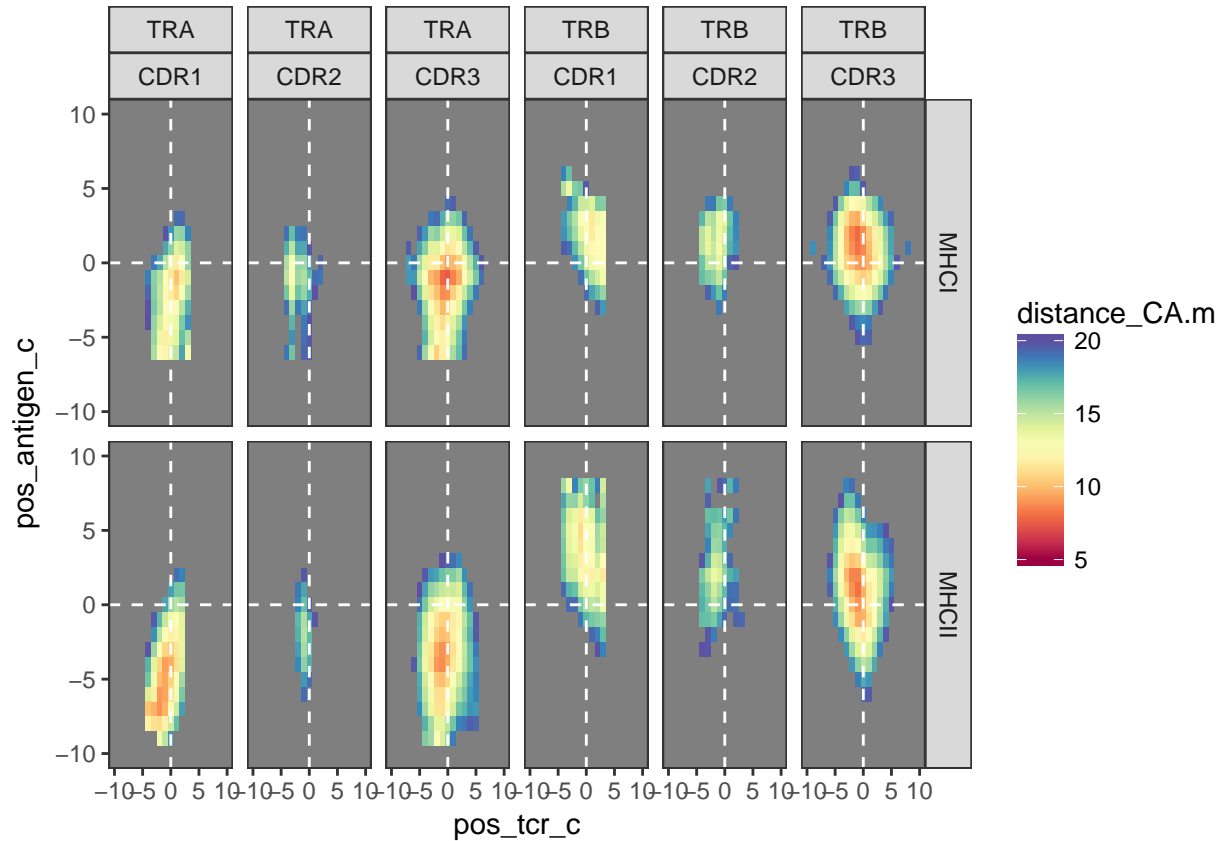
Simple mean model

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_antigen_c)]
```

Compare to true Calpha distance values

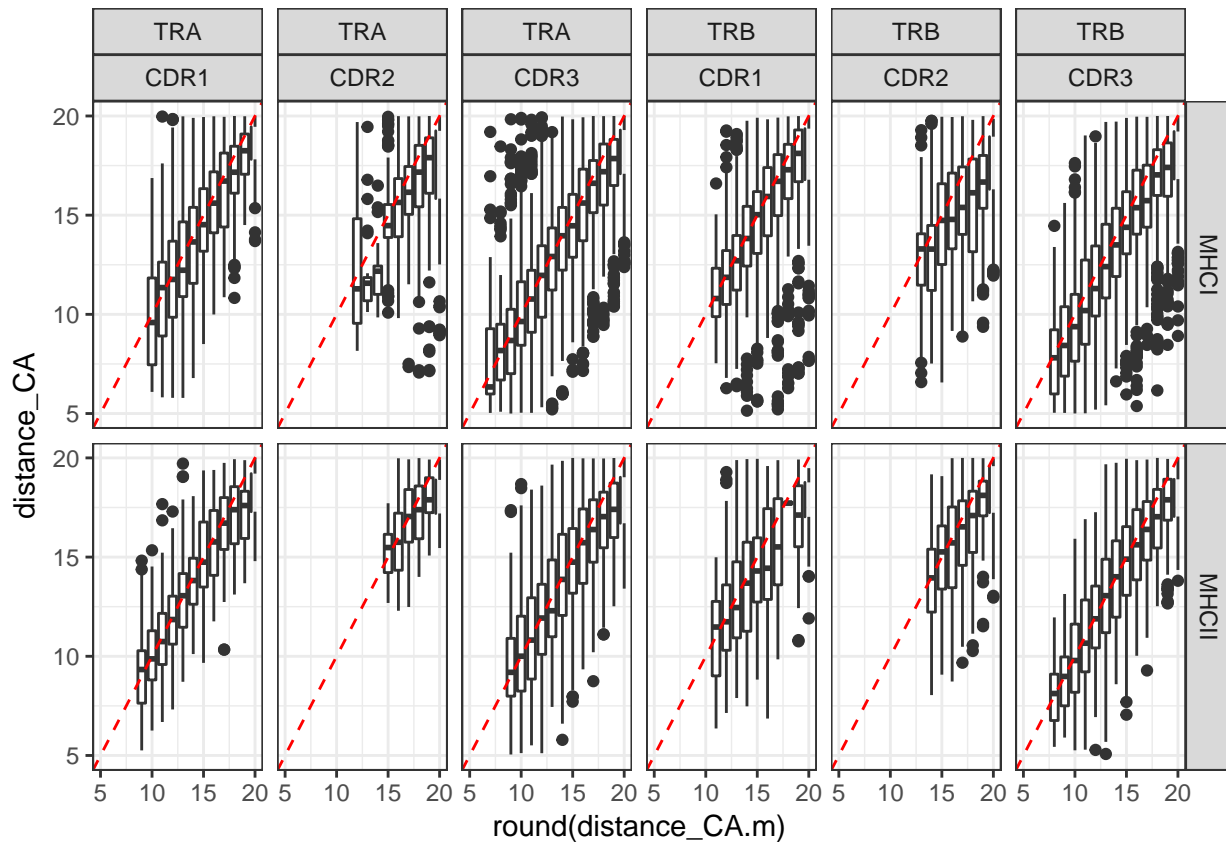
```
ggplot(df.pred, aes(x=round(distance_CA.m), group = round(distance_CA.m), y = distance_CA)) +
  geom_boxplot() +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +
  scale_x_continuous(limits=c(5,20)) +
  scale_y_continuous(limits=c(5,20)) +
  facet_grid(mhc_type~tcr_chain+tcr_region)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

```
## 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).
```



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

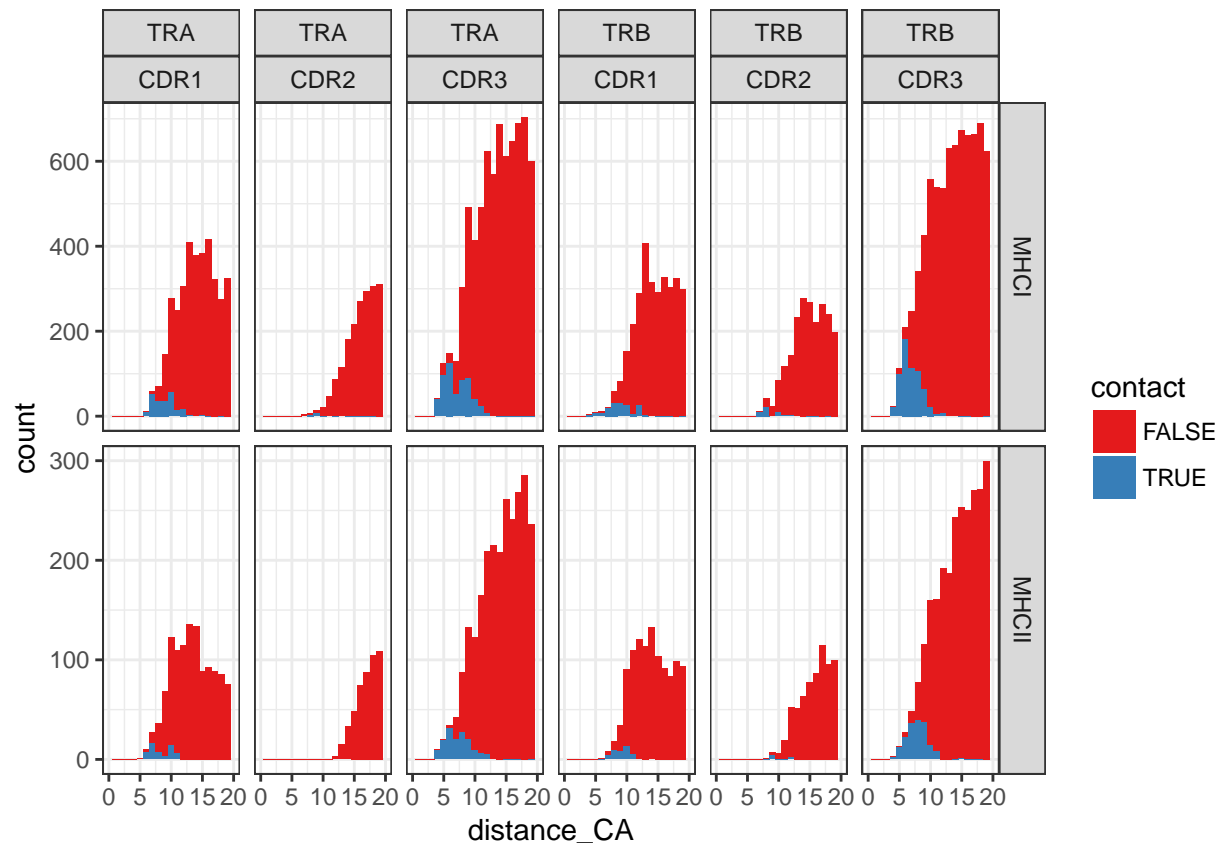
```
##
## Call:
## lm(formula = distance_CA ~ distance_CA.m, data = df.pred)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -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()
```

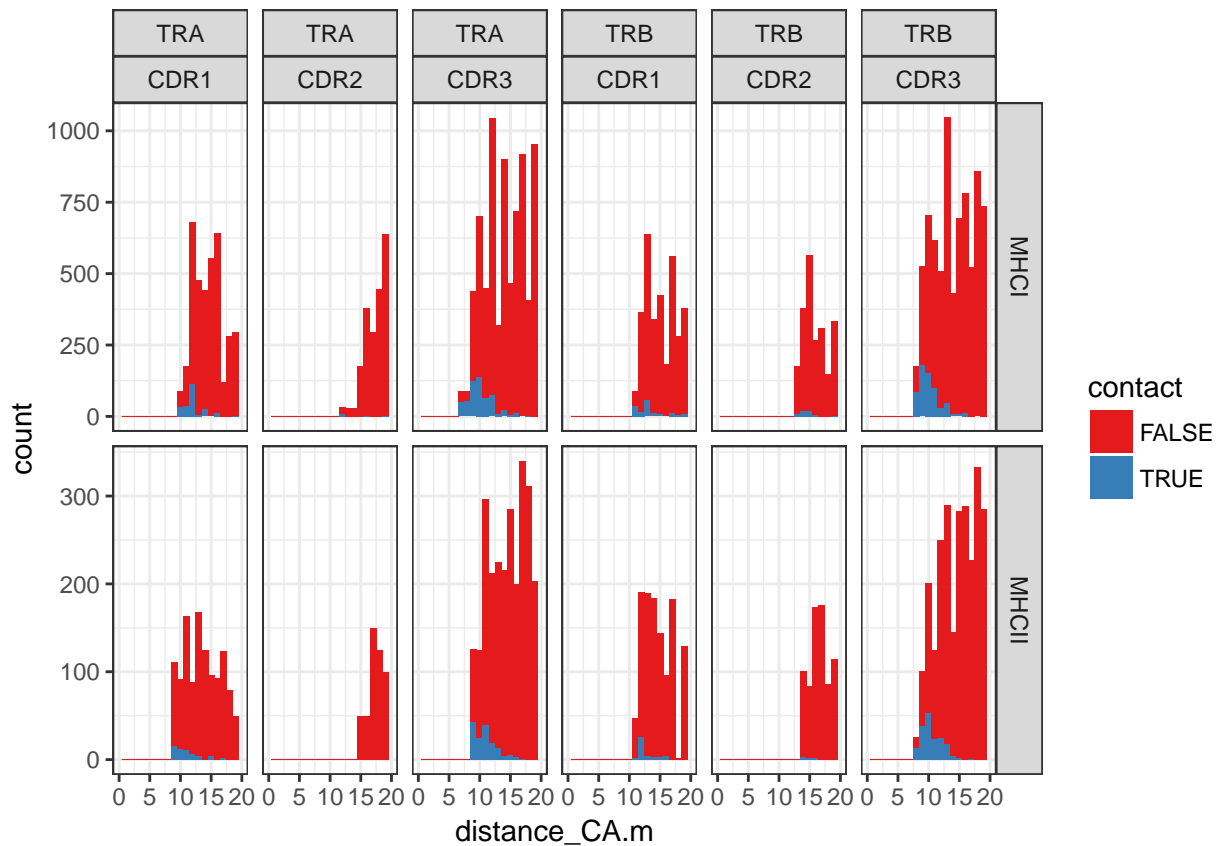
Warning: Removed 20140 rows containing non-finite values (stat_bin).



```
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

```
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)

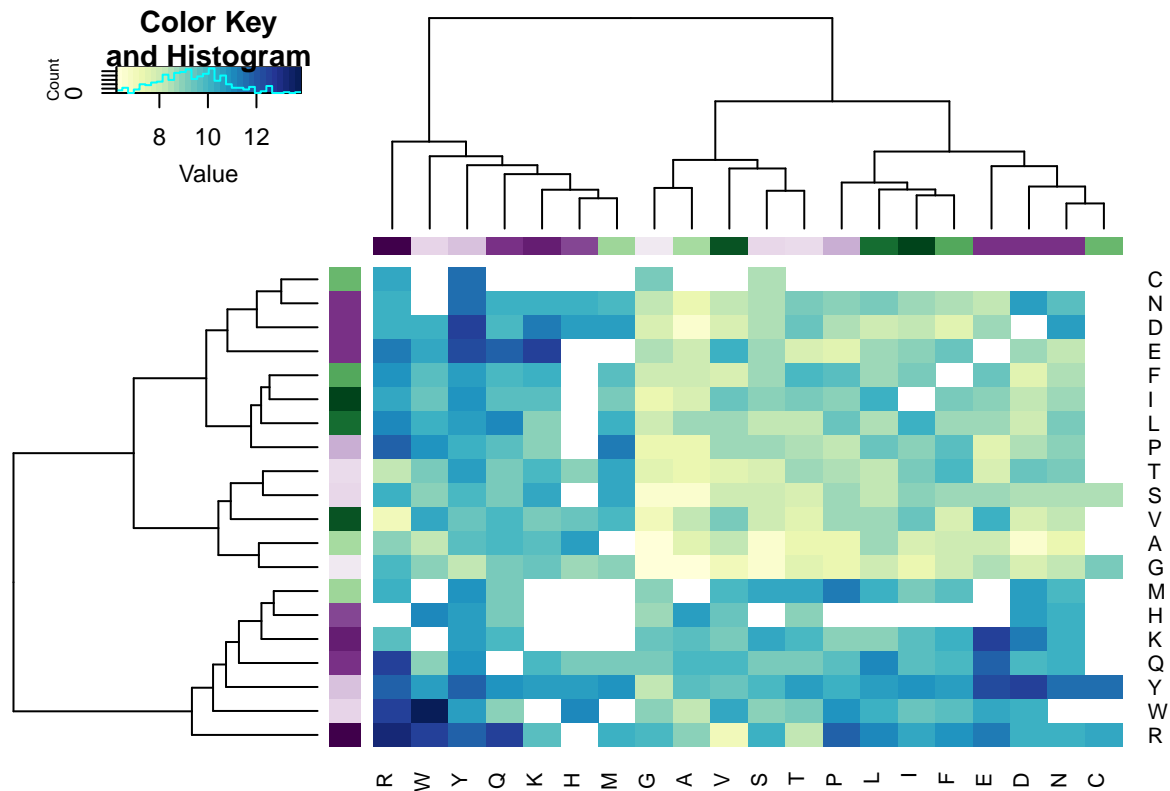
df.hydro <- data.frame(
  aa = strsplit("I V L F C M A W G T S Y P H N D Q E K R", " ")[[1]],
  hydrop = strsplit("4.5 4.2 3.8 2.8 2.5 1.9 1.8 -0.9 -0.4 -0.7 -0.8 -1.3 -1.6 -3.2 -3.5 -3.5 -3.5 -3.5
)

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))
```



Generate data for the imputation and EDA

```
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,0.08,0.12,0.15,0.18,0.21,0.24,0.27,0.3,0.33,0.36,0.39,0.42,0.45,0.48,0.51,0.54,0.57,0.6,0.63,0.66,0.69,0.72,0.75,0.78,0.81,0.84,0.87,0.9,0.93,0.96,0.99,1.02,1.05,1.08,1.11,1.14,1.17,1.2,1.23,1.26,1.29,1.32,1.35,1.38,1.41,1.44,1.47,1.5,1.53,1.56,1.59,1.62,1.65,1.68,1.71,1.74,1.77,1.8,1.83,1.86,1.89,1.92,1.95,1.98,2.01,2.04,2.07,2.1,2.13,2.16,2.19,2.22,2.25,2.28,2.31,2.34,2.37,2.4,2.43,2.46,2.49,2.52,2.55,2.58,2.61,2.64,2.67,2.7,2.73,2.76,2.79,2.82,2.85,2.88,2.91,2.94,2.97,3.0,3.03,3.06,3.09,3.12,3.15,3.18,3.21,3.24,3.27,3.3,3.33,3.36,3.39,3.42,3.45,3.48,3.51,3.54,3.57,3.6,3.63,3.66,3.69,3.72,3.75,3.78,3.81,3.84,3.87,3.9,3.93,3.96,3.99,4.02,4.05,4.08,4.11,4.14,4.17,4.2,4.23,4.26,4.29,4.32,4.35,4.38,4.41,4.44,4.47,4.5,4.53,4.56,4.59,4.62,4.65,4.68,4.71,4.74,4.77,4.8,4.83,4.86,4.89,4.92,4.95,4.98,5.01,5.04,5.07,5.1,5.13,5.16,5.19,5.22,5.25,5.28,5.31,5.34,5.37,5.4,5.43,5.46,5.49,5.52,5.55,5.58,5.61,5.64,5.67,5.7,5.73,5.76,5.79,5.82,5.85,5.88,5.91,5.94,5.97,6.0,6.03,6.06,6.09,6.12,6.15,6.18,6.21,6.24,6.27,6.3,6.33,6.36,6.39,6.42,6.45,6.48,6.51,6.54,6.57,6.6,6.63,6.66,6.69,6.72,6.75,6.78,6.81,6.84,6.87,6.9,6.93,6.96,6.99,7.02,7.05,7.08,7.11,7.14,7.17,7.2,7.23,7.26,7.29,7.32,7.35,7.38,7.41,7.44,7.47,7.5,7.53,7.56,7.59,7.62,7.65,7.68,7.71,7.74,7.77,7.8,7.83,7.86,7.89,7.92,7.95,7.98,8.01,8.04,8.07,8.1,8.13,8.16,8.19,8.22,8.25,8.28,8.31,8.34,8.37,8.4,8.43,8.46,8.49,8.52,8.55,8.58,8.61,8.64,8.67,8.7,8.73,8.76,8.79,8.82,8.85,8.88,8.91,8.94,8.97,9.0,9.03,9.06,9.09,9.12,9.15,9.18,9.21,9.24,9.27,9.3,9.33,9.36,9.39,9.42,9.45,9.48,9.51,9.54,9.57,9.6,9.63,9.66,9.69,9.72,9.75,9.78,9.81,9.84,9.87,9.9,9.93,9.96,9.99,10.02,10.05,10.08,10.11,10.14,10.17,10.2,10.23,10.26,10.29,10.32,10.35,10.38,10.41,10.44,10.47,10.5,10.53,10.56,10.59,10.62,10.65,10.68,10.71,10.74,10.77,10.8,10.83,10.86,10.89,10.92,10.95,10.98,11.01,11.04,11.07,11.1,11.13,11.16,11.19,11.22,11.25,11.28,11.31,11.34,11.37,11.4,11.43,11.46,11.49,11.52,11.55,11.58,11.61,11.64,11.67,11.7,11.73,11.76,11.79,11.82,11.85,11.88,11.91,11.94,11.97,12.0,12.03,12.06,12.09,12.12,12.15,12.18,12.21,12.24,12.27,12.3,12.33,12.36,12.39,12.42,12.45,12.48,12.51,12.54,12.57,12.6,12.63,12.66,12.69,12.72,12.75,12.78,12.81,12.84,12.87,12.9,12.93,12.96,12.99,13.02,13.05,13.08,13.11,13.14,13.17,13.2,13.23,13.26,13.29,13.32,13.35,13.38,13.41,13.44,13.47,13.5,13.53,13.56,13.59,13.62,13.65,13.68,13.71,13.74,13.77,13.8,13.83,13.86,13.89,13.92,13.95,13.98,14.01,14.04,14.07,14.1,14.13,14.16,14.19,14.22,14.25,14.28,14.31,14.34,14.37,14.4,14.43,14.46,14.49,14.52,14.55,14.58,14.61,14.64,14.67,14.7,14.73,14.76,14.79,14.82,14.85,14.88,14.91,14.94,14.97,15.0,15.03,15.06,15.09,15.12,15.15,15.18,15.21,15.24,15.27,15.3,15.33,15.36,15.39,15.42,15.45,15.48,15.51,15.54,15.57,15.6,15.63,15.66,15.69,15.72,15.75,15.78,15.81,15.84,15.87,15.9,15.93,15.96,15.99,16.02,16.05,16.08,16.11,16.14,16.17,16.2,16.23,16.26,16.29,16.32,16.35,16.38,16.41,16.44,16.47,16.5,16.53,16.56,16.59,16.62,16.65,16.68,16.71,16.74,16.77,16.8,16.83,16.86,16.89,16.92,16.95,16.98,17.0,17.03,17.06,17.09,17.12,17.15,17.18,17.21,17.24,17.27,17.3,17.33,17.36,17.39,17.42,17.45,17.48,17.51,17.54,17.57,17.6,17.63,17.66,17.69,17.72,17.75,17.78,17.81,17.84,17.87,17.9,17.93,17.96,17.99,18.0,18.03,18.06,18.09,18.12,18.15,18.18,18.21,18.24,18.27,18.3,18.33,18.36,18.39,18.42,18.45,18.48,18.51,18.54,18.57,18.6,18.63,18.66,18.69,18.72,18.75,18.78,18.81,18.84,18.87,18.9,18.93,18.96,18.99,19.0,19.03,19.06,19.09,19.12,19.15,19.18,19.21,19.24,19.27,19.3,19.33,19.36,19.39,19.42,19.45,19.48,19.51,19.54,19.57,19.6,19.63,19.66,19.69,19.72,19.75,19.78,19.81,19.84,19.87,19.9,19.93,19.96,19.99,20.0,20.03,20.06,20.09,20.12,20.15,20.18,20.21,20.24,20.27,20.3,20.33,20.36,20.39,20.42,20.45,20.48,20.51,20.54,20.57,20.6,20.63,20.66,20.69,20.72,20.75,20.78,20.81,20.84,20.87,20.9,20.93,20.96,20.99,21.0,21.03,21.06,21.09,21.12,21.15,21.18,21.21,21.24,21.27,21.3,21.33,21.36,21.39,21.42,21.45,21.48,21.51,21.54,21.57,21.6,21.63,21.66,21.69,21.72,21.75,21.78,21.81,21.84,21.87,21.9,21.93,21.96,21.99,22.0,22.03,22.06,22.09,22.12,22.15,22.18,22.21,22.24,22.27,22.3,22.33,22.36,22.39,22.42,22.45,22.48,22.51,22.54,22.57,22.6,22.63,22.66,22.69,22.72,22.75,22.78,22.81,22.84,22.87,22.9,22.93,22.96,22.99,23.0,23.03,23.06,23.09,23.12,23.15,23.18,23.21,23.24,23.27,23.3,23.33,23.36,23.39,23.42,23.45,23.48,23.51,23.54,23.57,23.6,23.63,23.66,23.69,23.72,23.75,23.78,23.81,23.84,23.87,23.9,23.93,23.96,23.99,24.0,24.03,24.06,24.09,24.12,24.15,24.18,24.21,24.24,24.27,24.3,24.33,24.36,24.39,24.42,24.45,24.48,24.51,24.54,24.57,24.6,24.63,24.66,24.69,24.72,24.75,24.78,24.81,24.84,24.87,24.9,24.93,24.96,24.99,25.0,25.03,25.06,25.09,25.12,25.15,25.18,25.21,25.24,25.27,25.3,25.33,25.36,25.39,25.42,25.45,25.48,25.51,25.54,25.57,25.6,25.63,25.66,25.69,25.72,25.75,25.78,25.81,25.84,25.87,25.9,25.93,25.96,25.99,26.0,26.03,26.06,26.09,26.12,26.15,26.18,26.21,26.24,26.27,26.3,26.33,26.36,26.39,26.42,26.45,26.48,26.51,26.54,26.57,26.6,26.63,26.66,26.69,26.72,26.75,26.78,26.81,26.84,26.87,26.9,26.93,26.96,26.99,27.0,27.03,27.06,27.09,27.12,27.15,27.18,27.21,27.24,27.27,27.3,27.33,27.36,27.39,27.42,27.45,27.48,27.51,27.54,27.57,27.6,27.63,27.66,27.69,27.72,27.75,27.78,27.81,27.84,27.87,27.9,27.93,27.96,27.99,28.0,28.03,28.06,28.09,28.12,28.15,28.18,28.21,28.24,28.27,28.3,28.33,28.36,28.39,28.42,28.45,28.48,28.51,28.54,28.57,28.6,28.63,28.66,28.69,28.72,28.75,28.78,28.81,28.84,28.87,28.9,28.93,28.96,28.99,29.0,29.03,29.06,29.09,29.12,29.15,29.18,29.21,29.24,29.27,29.3,29.33,29.36,29.39,29.42,29.45,29.48,29.51,29.54,29.57,29.6,29.63,29.66,29.69,29.72,29.75,29.78,29.81,29.84,29.87,29.9,29.93,29.96,29.99,30.0,30.03,30.06,30.09,30.12,30.15,30.18,30.21,30.24,30.27,30.3,30.33,30.36,30.39,30.42,30.45,30.48,30.51,30.54,30.57,30.6,30.63,30.66,30.69,30.72,30.75,30.78,30.81,30.84,30.87,30.9,30.93,30.96,30.99,31.0,31.03,31.06,31.09,31.12,31.15,31.18,31.21,31.24,31.27,31.3,31.33,31.36,31.39,31.42,31.45,31.48,31.51,31.54,31.57,31.6,31.63,31.66,31.69,31.72,31.75,31.78,31.81,31.84,31.87,31.9,31.93,31.96,31.99,32.0,32.03,32.06,32.09,32.12,32.15,32.18,32.21,32.24,32.27,32.3,32.33,32.36,32.39,32.42,32.45,32.48,32.51,32.54,32.57,32.6,32.63,32.66,32.69,32.72,32.75,32.78,32.81,32.84,32.87,32.9,32.93,32.96,32.99,33.0,33.03,33.06,33.09,33.12,33.15,33.18,33.21,33.24,33.27,33.3,33.33,33.36,33.39,33.42,33.45,33.48,33.51,33.54,33.57,33.6,33.63,33.66,33.69,33.72,33.75,33.78,33.81,33.84,33.87,33.9,33.93,33.96,33.99,34.0,34.03,34.06,34.09,34.12,34.15,34.18,34.21,34.24,34.27,34.3,34.33,34.36,34.39,34.42,34.45,34.48,34.51,34.54,34.57,34.6,34.63,34.66,34.69,34.72,34.75,34.78,34.81,34.84,34.87,34.9,34.93,34.96,34.99,35.0,35.03,35.06,35.09,35.12,35.15,35.18,35.21,35.24,35.27,35.3,35.33,35.36,35.39,35.42,35.45,35.48,35.51,35.54,35.57,35.6,35.63,35.66,35.69,35.72,35.75,35.78,35.81,35.84,35.87,35.9,35.93,35.96,35.99,36.0,36.03,36.06,36.09,36.12,36.15,36.18,36.21,36.24,36.27,36.3,36.33,36.36,36.39,36.42,36.45,36.48,36.51,36.54,36.57,36.6,36.63,36.66,36.69,36.72,36.75,36.78,36.81,36.84,36.87,36.9,36.93,36.96,36.99,37.0,37.03,37.06,37.09,37.12,37.15,37.18,37.21,37.24,37.27,37.3,37.33,37.36,37.39,37.42,37.45,37.48,37.51,37.54,37.57,37.6,37.63,37.66,37.69,37.72,37.75,37.78,37.81,37.84,37.87,37.9,37.93,37.96,37.99,38.0,38.03,38.06,38.09,38.12,38.15,38.18,38.21,38.24,38.27,38.3,38.33,38.36,38.39,38.42,38.45,38.48,38.51,38.54,38.57,38.6,38.63,38.66,38.69,38.72,38.75,38.78,38.81,38.84,38.87,38.9,38.93,38.96,38.99,39.0,39.03,39.06,39.09,39.12,39.15,39.18,39.21,39.24,39.27,39.3,39.33,39.36,39.39,39.42,39.45,39.48,39.51,39.54,39.57,39.6,39.63,39.66,39.69,39.72,39.75,39.78,39.81,39.84,39.87,39.9,39.93,39.96,39.99,40.0,40.03,40.06,40.09,40.12,40.15,40.18,40.21,40.24,40.27,40.3,40.33,40.36,40.39,40.42,40.45,40.48,40.51,40.54,40.57,40.6,40.63,40.66,40.69,40.72,40.75,40.78,40.81,40.84,40.87,40.9,40.93,40.96,40.99,41.0,41.03,41.06,41.09,41.12,41.15,41.18,41.21,41.24,41.27,41.3,41.33,41.36,41.39,41.42,41.45,41.48,41.51,41.54,41.57,41.6,41.63,41.66,41.69,41.72,41.75,41.78,41.81,41.84,41.87,41.9,41.93,41.96,41.99,42.0,42.03,42.06,42.09,42.12,42.15,42.18,42.21,42.24,42.27,42.3,42.33,42.36,42.39,42.42,42.45,42.48,42.51,42.54,42.57,42.6,42.63,42.66,42.69,42.72,42.75,42.78,42.81,42.84,42.87,42.9,42.93,42.96,42.99,43.0,43.03,43.06,43.09,43.12,43.15,43.18,43.21,43.24,43.27,43.3,43.33,43.36,43.39,43.42,43.45,43.48,43.51,43.54,43.57,43.6,43.63,43.66,43.69,43.72,43.75,43.78,43.81,43.84,43.87,43.9,43.93,43.96,43.99,44.0,44.03,44.06,44.09,44.12,44.15,44.18,44.21,44.24,44.27,44.3,44.33,44.36,44.39,44.42,44.45,44.48,44.51,44.54,44.57,44.6,44.63,44.66,44.69,44.72,44.75,44.78,44.81,44.84,44.87,44.9,44.93,44.96,44.99,45.0,45.03,45.06,45.09,45.12,45.15,45.18,45.21,45.24,45.27,45.3,45.33,45.36,45.39,45.42,45.45,45.48,45.51,45.54,45.57,45.6,45.63,45.66,45.69,45.72,45.75,45.78,45.81,45.84,45.87,45.9,45.93,45.96,45.99,46.0,46.03,46.06,46.09,46.12,46.15,46.18,46.21,46.24,46.27,46.3,46.33,46.36,46.39,46.42,46.45,46.48,46.51,46.54,46.57,46.6,46.63,46.66,46.69,46.72,46.75,46.78,46.81,46.84,46.87,46.9,46.93,46.96,46.99,47.0,47.03,47.06,47.09,47.12,47.15,47.18,47.21,47.24,47.27,47.3,47.33,47.36,47.39,47.42,47.45,47.48,47.51,47.54,47.57,47.6,47.63,47.66,47.69,47.72,47.75,47.78,47.81,47.84,47.87,47.9,47.93,47.96,47.99,48.0,48.03,48.06,48.09,48.12,48.15,48.18,48.21,48.24,48.27,48.3,48.33,48.36,48.39,48.42,48.45,48.48,48.51,48.54,48.57,48.6,48.63,48.66,48.69,48.72,48.75,48.78,48.81,48.84,48.87,48.9,48.93,48.96,48.99,49.0,49.03,49.06,49.09,49.12,49.15,49.18,49.21,49.24,49.27,49.3,49.33,49.36,49.39,49.42,49.45,49.48,49.51,49.54,49.57,49.6,49.63,49.66,49.69,49.72,49.75,49.78,49.81,49.84,49.87,49.9,49.93,49.96,49.99,50.0,50.03,50.06,50.09,50.12,50.15,50.18,50.21,50.24,50.27,50.3,50.33,50.36,50.39,50.42,50.45,50.48,50.51,50.54,50.57,50.6,50.63,50.66,50.69,50.72,50.75,50.78,50.81,50.84,50.87,50.9,50.93,50.96,50.99,51.0,51.03,51.06,51.09,51.12,51.15,51.18,51.21,51.24,51.27,51.3,51.33,51.36,51.39,51.42,51.45,51.48,51.51,51.54,51.57,51.6,51.63,51.66,51.69,51.72,51.75,51.78,51.81,51.84,51.87,51.9,51.93,51.96,51.99,52.0,52.03,52.06,52.09,52.12,52.15,52.18,52.21,52.24,52.27,52.3,52.33,52.36,52.39,52.42,52.45,52.48,52.51,52.54,52.57,52.6,52.63,52.66,52.69,52.72,52.75,52.78,52.81,52.84,52.87,52.9,52.93,52.96,52.99,53.0,53.03,53.06,53.09,53.12,53.15,53.18,53.21,53.24,53.27,53.3,53.33,53.36,53.39,53.42,53.45,53.48,53.51,53.54,53.57,53.6,53.63,53.66,53.69,53.72,53.75,53.78,53.81,53.84,53.87,53.9,53.93,53.96,53.99,54.0,54.03,54.06,54.09,54.12,54.15,54.18,54.21,54.24,54.27,54.3,54.33,54.36,54.39,54.42,54.45,54.48,54.51,54.54,54.57,54.6,54.63,54.66,54.69,54.72,54.75,54.78,54.81,54.84,54.87,54.9,54.93,54.96,54.99,55.0,55.03,55.06,55.09,55.12,55.15,55.18,55.21,55.24,55.27,55.3,55.33,55.36,55.39,55.42,55.45,55.48,55.51,55.54,55.57,55.6,55.63,55.66,55.69,55.72,55.75,55.78,55.81,55.84,55.87,55.9,55.93,55.96,55.99,56.0,56.03,56.06,56.09,56.12,56.15,56.18,56.21,56.24,56.27,56.3,56.33,56.36,56.39,56.42,56.45,56.48,56.51,56.54,56.57,56.6,56.63,56.66,56.69,56.72,56.75,56.78,56.81,56.84,56.87,56.9,56.93,56.96,56.99,57.0,57.03,57.06,57.09,57.12,57.15,57.18,57.21,57.24,57.27,57.3,57.33,57.36,57.39,57.42,57.45,57.48,57.51,57.54,57.57,57.6,57.63,57.66,57.69,57.72,57.75,57.78,57.81,57.84,57.87,57.9,57.93,57.96,57.99,58.0,58.03,58.06,58.09,58.12,58.15,58.18,58.21,58.24,58.27,58.3,58.33,58.36,58.39,58.42,58.45,58.48,58.51,58.54,58.57,58.6,58.63,58.66,58.69,58.72,58.75,58.78,58.81,58.84,58.87,58.9,58.93,58.96,58.99,59.0,59.03,59.06,59.09,59.12,59.15,59.18,59.21,59.24,59.27,59.3,59.33,59.36,59.39,59.42,59.45,59.48,59.51,59.54,59.57,59.6,59.63,59.66,59.69,59.72,59.75,59.78,59.81,59.84,59.87,59.9,59.93,59.96,59.99,60.0,60.03,60.06,60.09,60.12,60.15,60.18,60.21,60.24,60.27,60.3,60.33,60.36,60.39,60.42,60.45,60.48,60.51,60.54,60.57,60.6,60.63,60.66,60.69,60.72,60.75,60.78,60.81,60.84,60.87,60.9,60.93,60.96,60.99,61.0,61.03,61.06,61.09,61.12,61.15,61.18,61
```

```

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)
}

aa_data = generate_data(aa_pair_mat, .cv = 10)

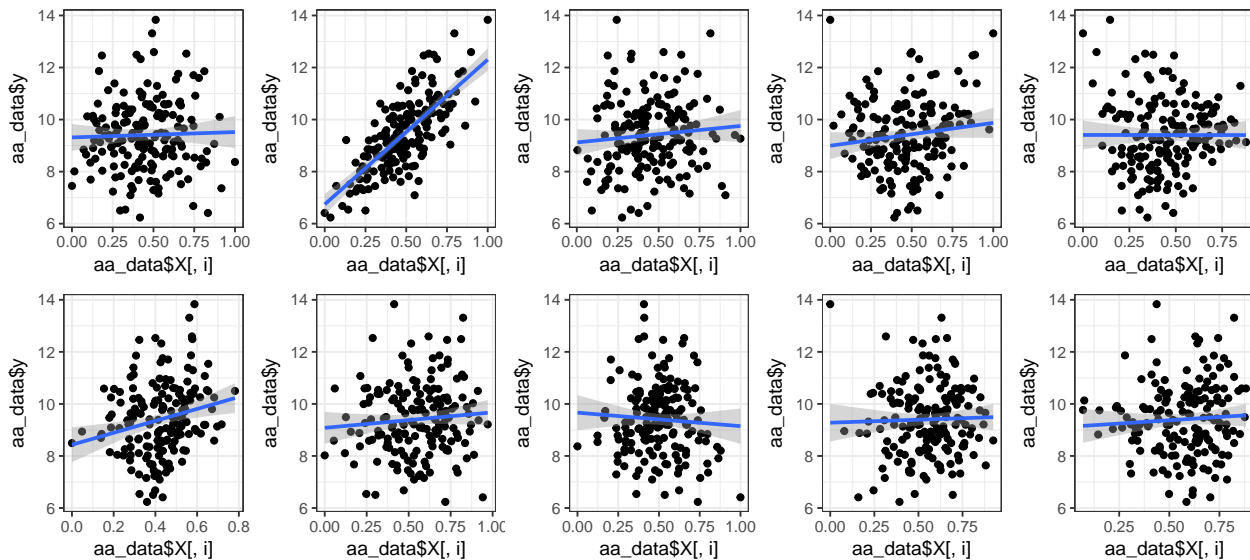
suppressWarnings( {
  ps = lapply(1:10, function (i) { qplot(aa_data$X[i], aa_data$y, geom = c("point", "smooth"), method=
  })
summary(lm(aa_data$y ~ aa_data$X))

##
## Call:
## lm(formula = aa_data$y ~ aa_data$X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.06671 -0.51364 -0.02839  0.52748  2.40687
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.90095     0.61912   7.916 3.85e-13 ***
## aa_data$X1  -0.06842     0.35199  -0.194  0.84613
## aa_data$X2   5.51113     0.39215  14.054 < 2e-16 ***
## aa_data$X3   0.24973     0.35140   0.711  0.47832
## aa_data$X4   0.89005     0.35531   2.505  0.01325 *

```

```
## aa_data$X5    0.68336    0.39071    1.749    0.08221 .
## aa_data$X6    1.57619    0.56217    2.804    0.00568 **
## aa_data$X7    0.32237    0.35857    0.899    0.36997
## aa_data$X8   -0.32045    0.43291   -0.740    0.46024
## aa_data$X9   -0.04105    0.43003   -0.095    0.92406
## aa_data$X10   0.69980    0.42492    1.647    0.10154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9105 on 160 degrees of freedom
## Multiple R-squared:  0.5986, Adjusted R-squared:  0.5736
## F-statistic: 23.86 on 10 and 160 DF,  p-value: < 2.2e-16
```

```
do.call(gridExtra::grid.arrange, c(ps, list(ncol = 5)))
```



Impute the missing values

```
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)
}

impute_reg <- function (X_train, y_train, X_val, y_val, X_test) {
  model = lm(aa_data$y ~ ., data = data.frame(aa_data$X))

  X_df = data.frame(X_train)
  res_tr = sapply(1:nrow(X_train), function (row_i) {
    predict(model, X_df[row_i, ])
  })

  X_df = data.frame(X_val)
  res_val = sapply(1:nrow(X_val), function (row_i) {
    predict(model, X_df[row_i, ])
  })
}

```

```

merged = rbind(X_train, X_val)
merged_y = c(y_train, y_val)
X_df = data.frame(merged)
model = lm(merged_y ~ ., data = X_df)
imp_tst = sapply(1:nrow(X_test), function (row_i) {
  predict(model, X_df[row_i, ])
})

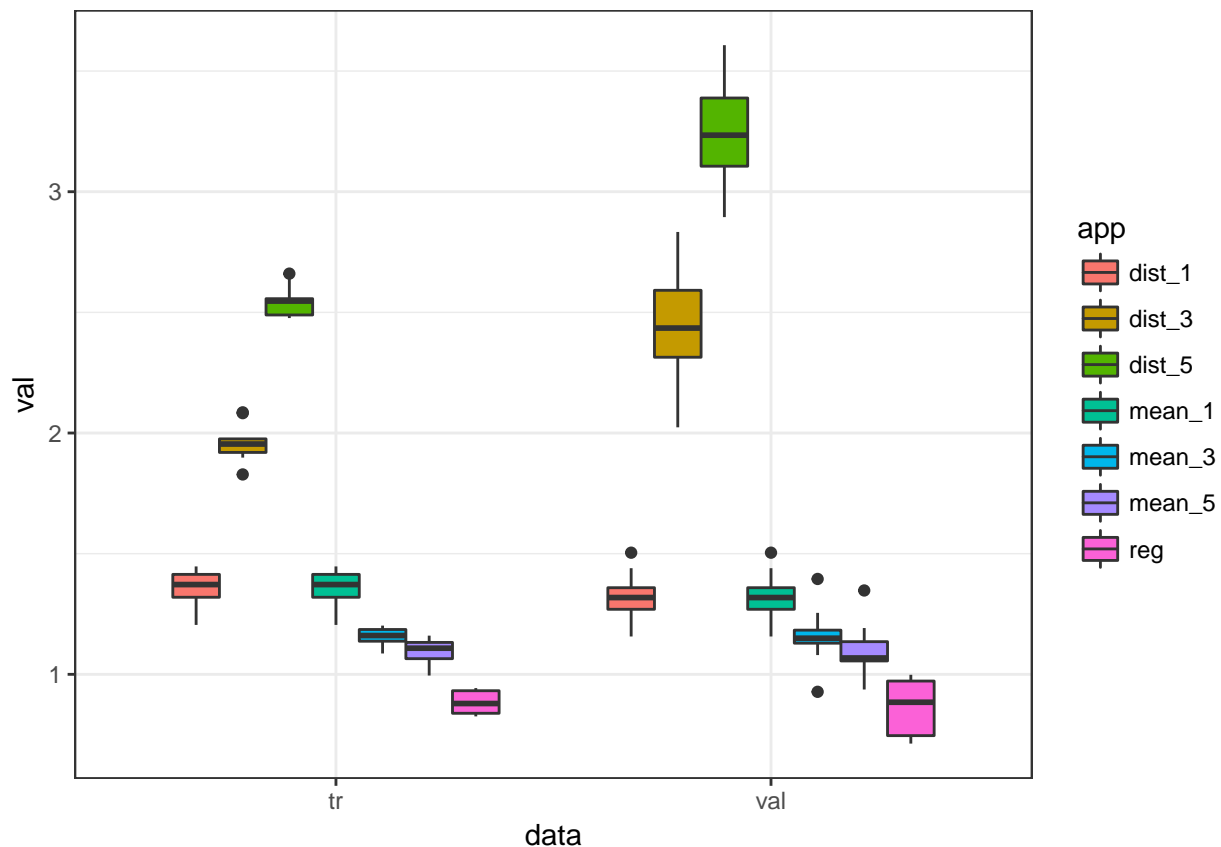
list(res_tr, res_val, imp_tst)
}

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)
}

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[["reg"]] = eval_model(aa_data, impute_reg)
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_reg(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)[[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,

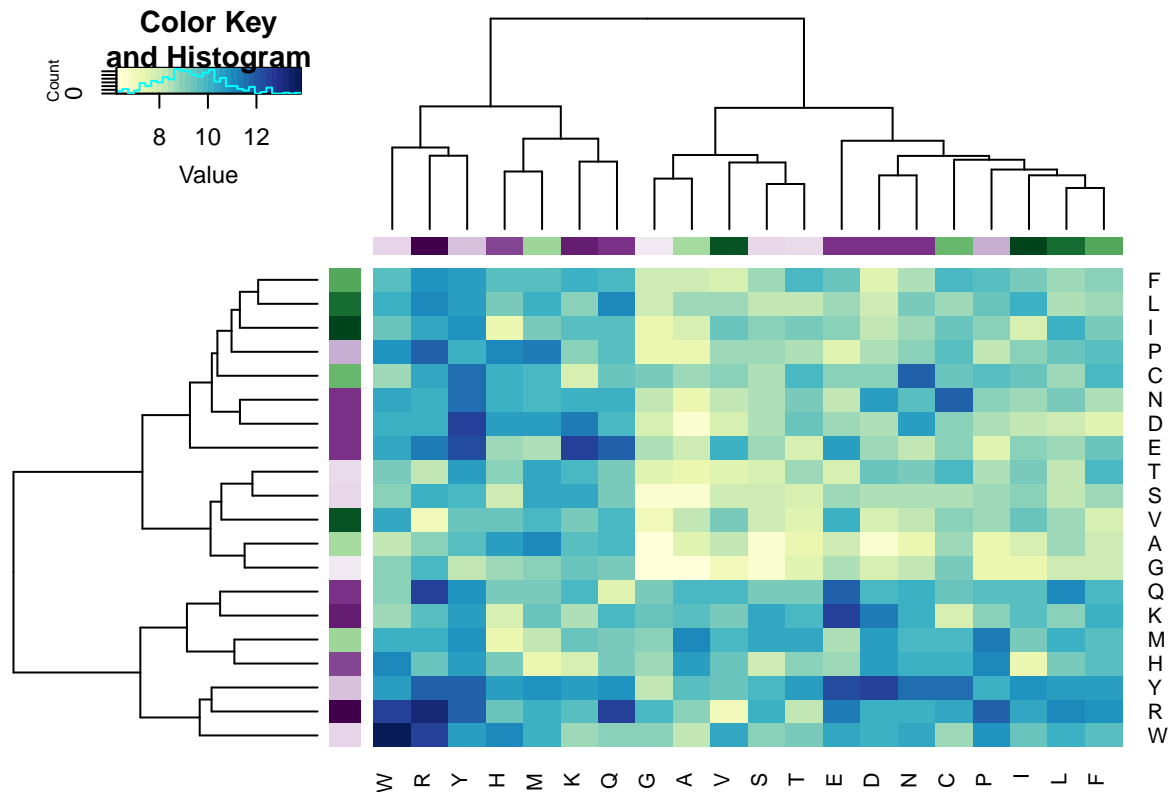
```

```

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

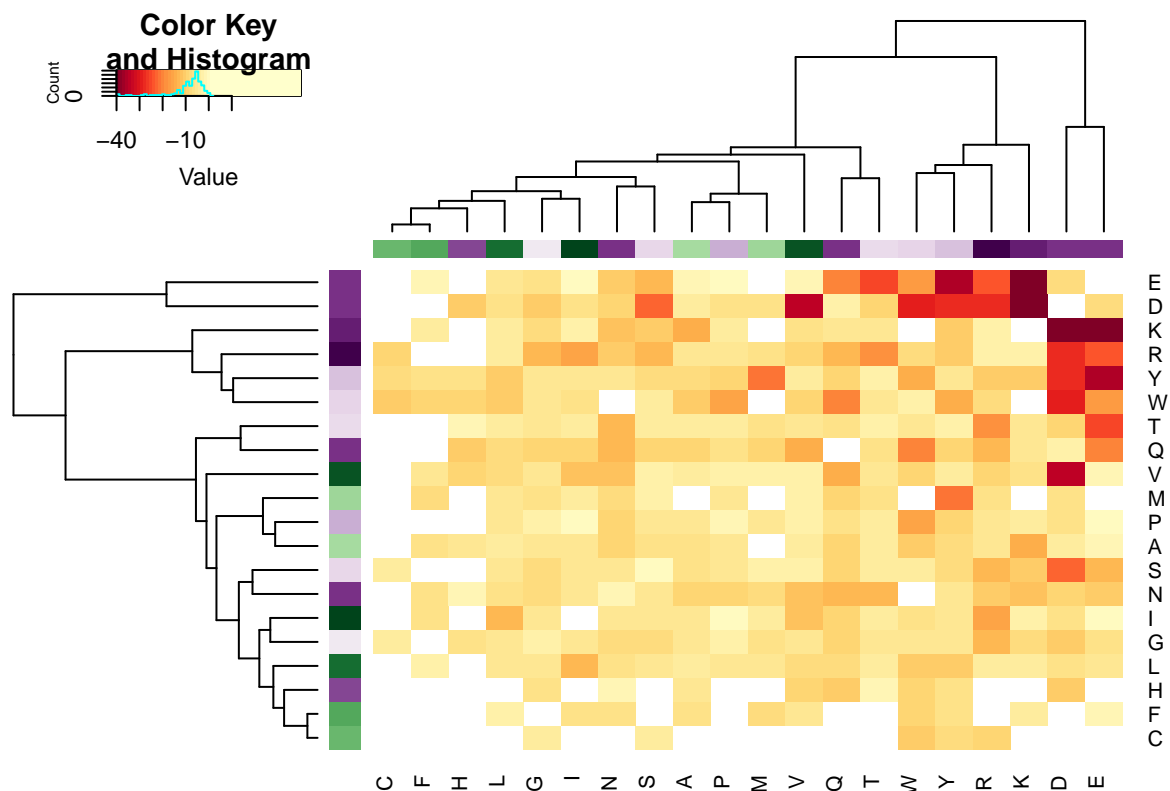
df.energies = rbind(df.energies, df.energies.tmp) %>% unique()

# 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

```

aa_data = generate_data(aa_pair_energy_mat, .cv = 10)

suppressWarnings( {
  ps = lapply(1:10, function (i) { qplot(aa_data$X[,i], aa_data$y, geom = c("point", "smooth"), method=

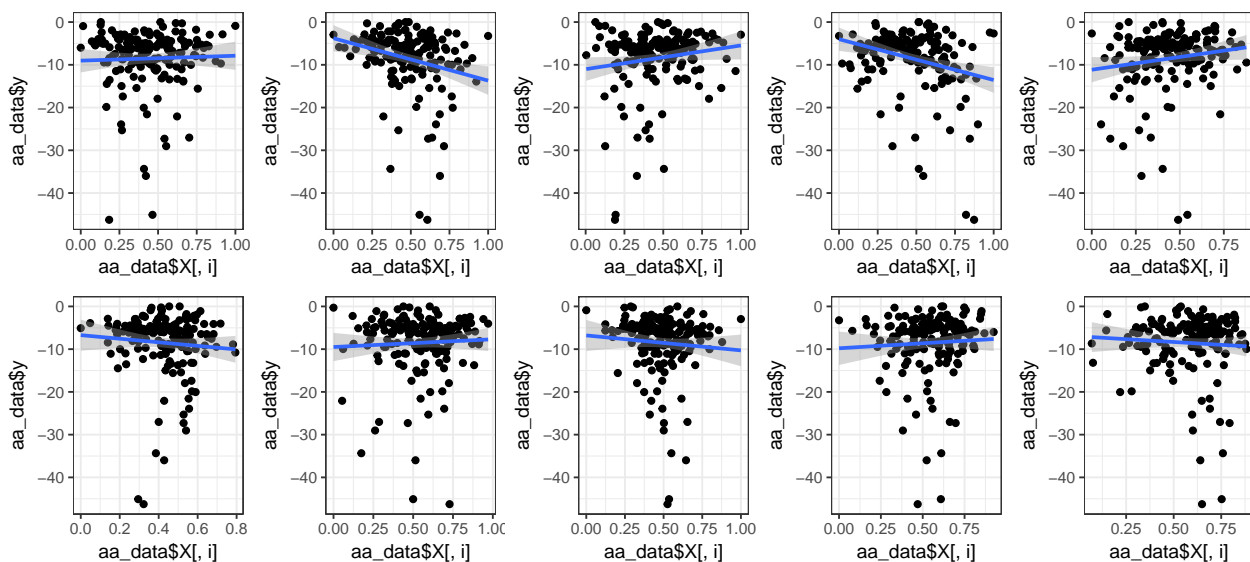
```

```

})
summary(lm(aa_data$y ~ aa_data$X))

##
## Call:
## lm(formula = aa_data$y ~ aa_data$X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.5685  -1.5786   0.4366   3.6539  12.8935
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.976     4.631  -1.074  0.284298
## aa_data$X1     1.659     2.622   0.633  0.527820
## aa_data$X2    -11.517     2.922  -3.941  0.000123 ***
## aa_data$X3     7.072     2.625   2.694  0.007842 **
## aa_data$X4    -11.438     2.618  -4.368  2.29e-05 ***
## aa_data$X5     3.276     2.893   1.132  0.259215
## aa_data$X6     1.408     4.031   0.349  0.727331
## aa_data$X7     4.195     2.690   1.559  0.120974
## aa_data$X8    -5.092     3.247  -1.568  0.118890
## aa_data$X9     4.970     3.305   1.504  0.134711
## aa_data$X10   -1.958     3.164  -0.619  0.537000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.731 on 154 degrees of freedom
## Multiple R-squared:  0.2227, Adjusted R-squared:  0.1722
## F-statistic: 4.412 on 10 and 154 DF,  p-value: 1.853e-05
do.call(gridExtra::grid.arrange, c(ps, list(ncol = 5)))

```



```

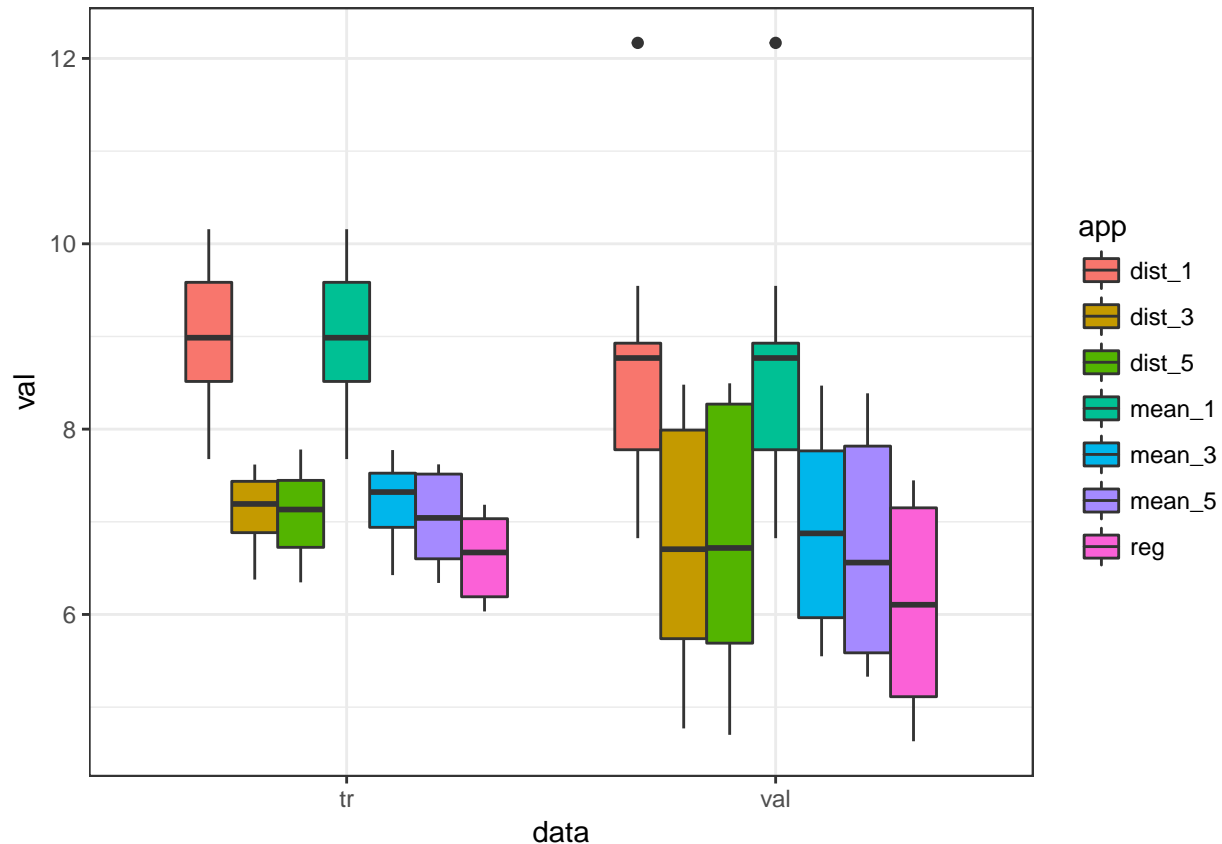
imp_res = list()
imp_res[["mean_1"]] = eval_model(aa_data, impute_knn_mean, .k = 1)
imp_res[["mean_3"]] = eval_model(aa_data, impute_knn_mean, .k = 3)

```

```

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[["reg"]] = eval_model(aa_data, impute_reg)
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

```

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_reg(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)[[3]]

test_data[,3] = imputed

aa_pair_energy_mat_imp = aa_pair_energy_mat

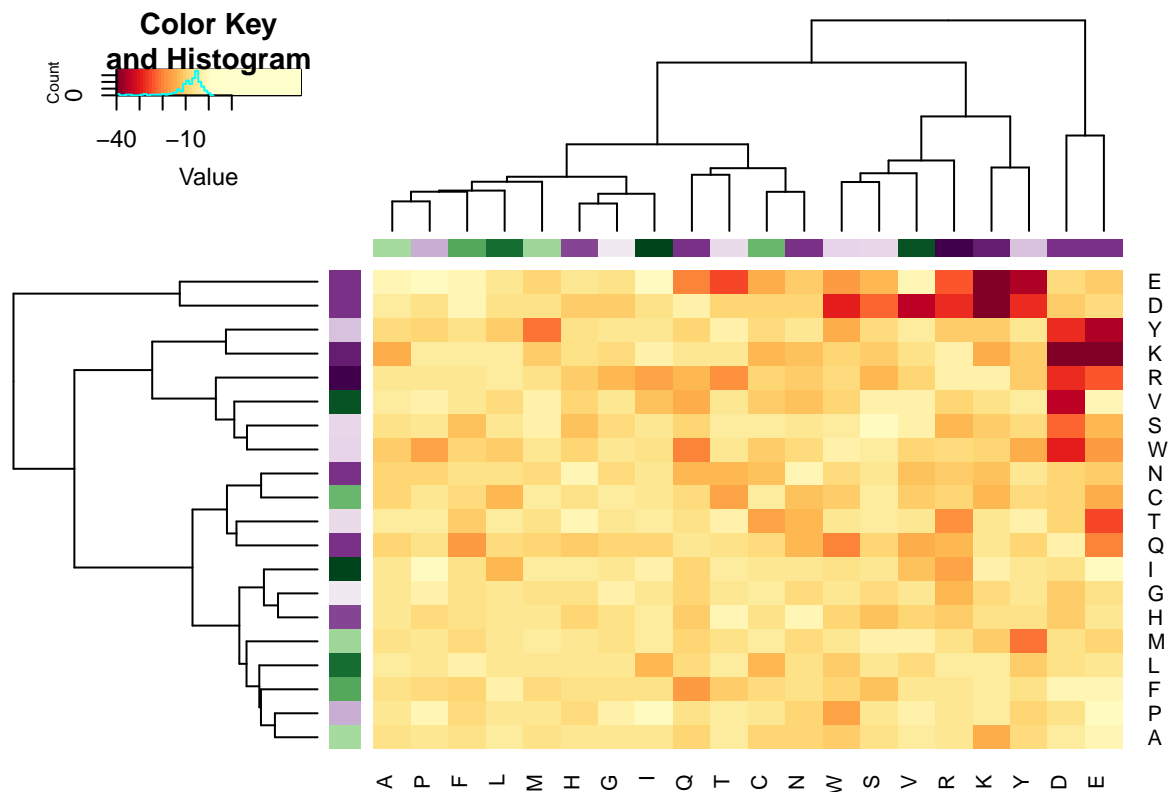
```

```

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 {
    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
## -2.1033  -0.4916  -0.3102  -0.1683   3.7768
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.59224    0.19437  -3.047  0.00231 **
## distance_CA.m -0.47580    0.01238 -38.441 < 2e-16 ***
## coef          0.46437    0.01837  25.278 < 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: 10802  on 18846  degrees of freedom
## AIC: 10808
##
## Number of Fisher Scoring iterations: 6

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:

```

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)

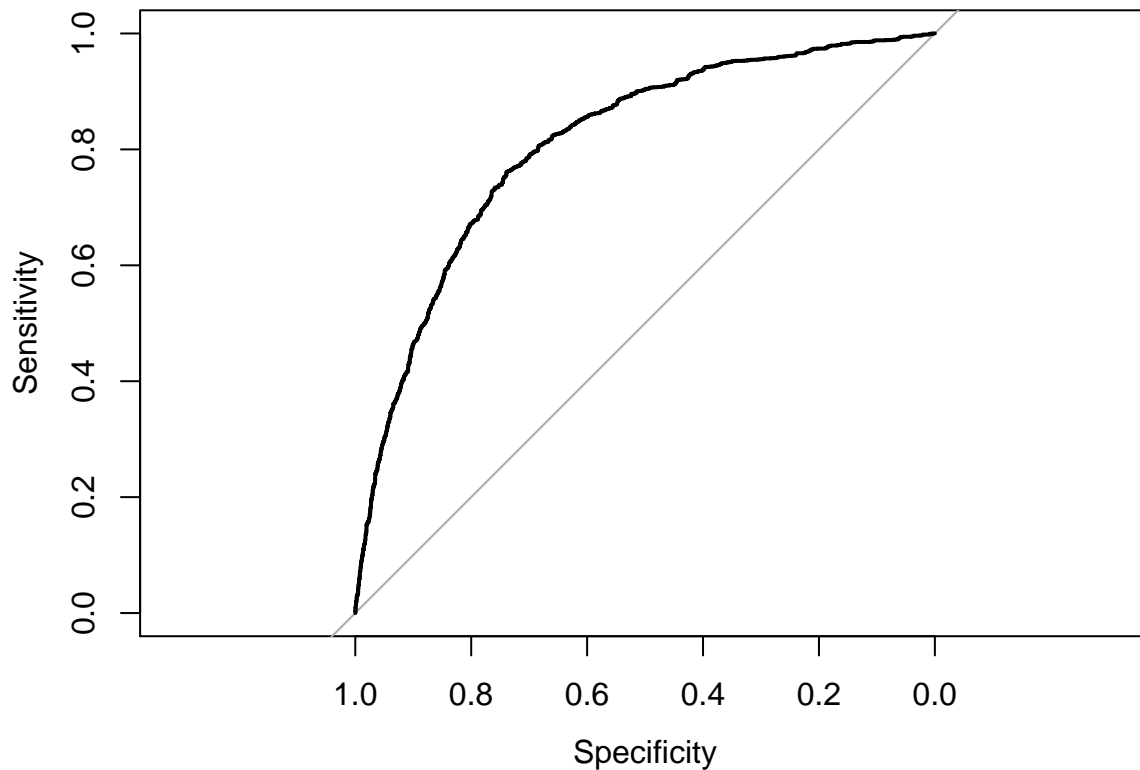
```

Check accuracy

General

ROC curve

```
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: 0.8091
## 95% CI: 0.7996-0.8187 (DeLong)
```

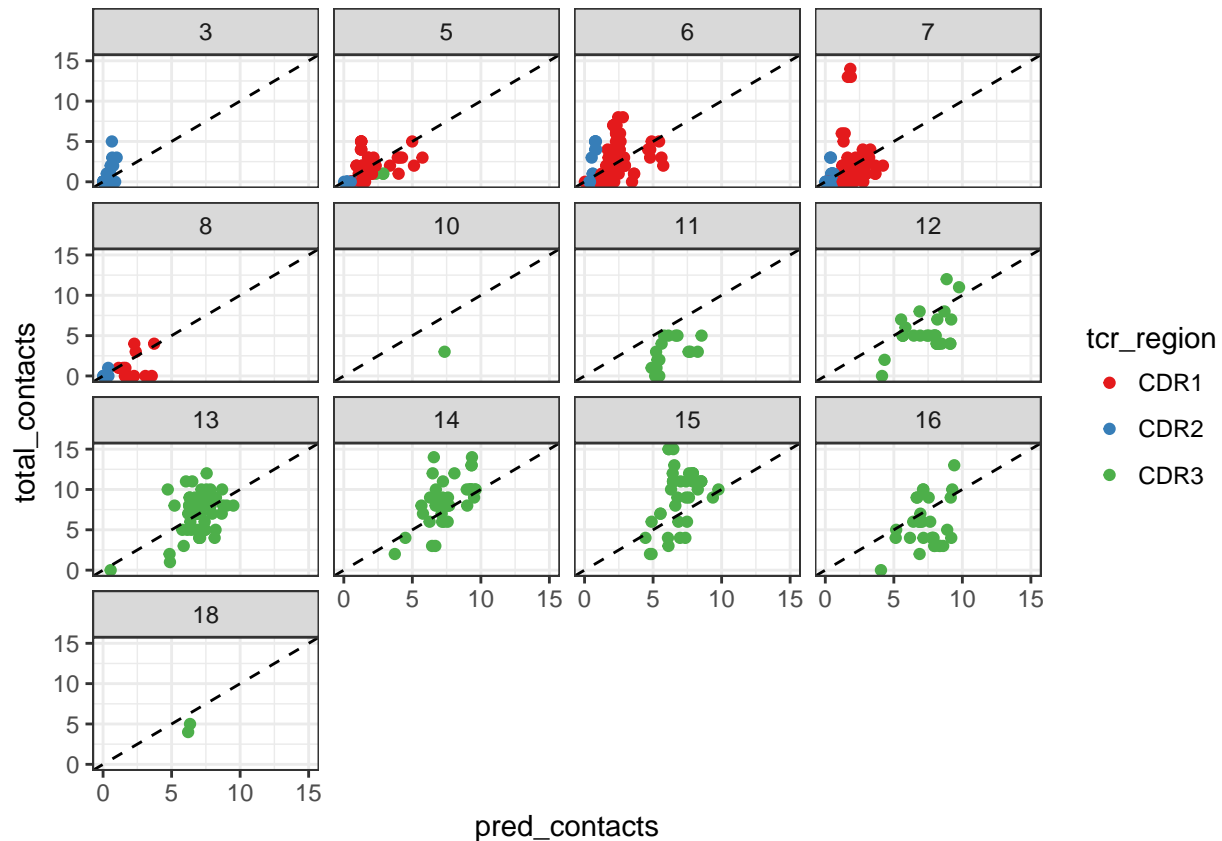
Compute true and estimated total number of contacts

```
df.pred.contsum = df.pred.trimmed[, .(total_contacts = sum(contact), pred_contacts = sum(p, na.rm=T)), 1]
```

```
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).
```



```
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
## -5.8153 -1.2573 -0.1692  0.7362 11.7520
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## pred_contacts    0.83186    0.08847   9.402 < 2e-16 ***
## len_tcr          0.27850    0.06039   4.612 4.81e-06 ***
## tcr_regionCDR1  -1.22029    0.41207  -2.961 0.003175 **
## tcr_regionCDR2  -1.00606    0.28782  -3.495 0.000505 ***
## tcr_regionCDR3  -2.79530    0.93879  -2.978 0.003014 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.31 on 650 degrees of freedom
## Multiple R-squared:  0.7808, Adjusted R-squared:  0.7791
## F-statistic:  463 on 5 and 650 DF, p-value: < 2.2e-16
```

```
anova(lfit)
```

```
## Analysis of Variance Table
##
## Response: total_contacts
##           Df Sum Sq Mean Sq  F value    Pr(>F)
## pred_contacts  1 12227.1 12227.1 2291.4261 < 2.2e-16 ***
## len_tcr        1   57.3    57.3   10.7403 0.001104 **
## tcr_region     3   69.1    23.0    4.3177 0.005006 **
## Residuals     650 3468.4     5.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

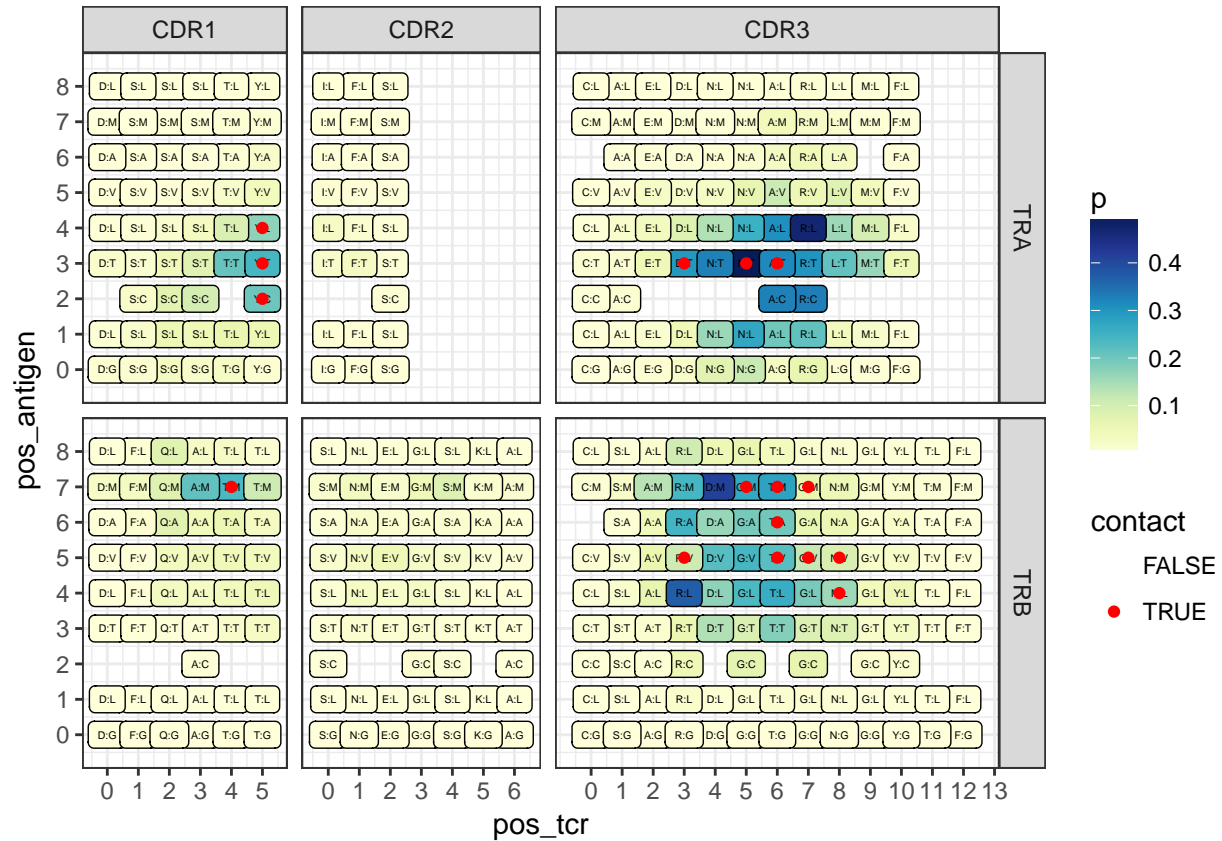
Examples from the train data

Check for a couple of antigens, GLCTLVAML

```
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).
```

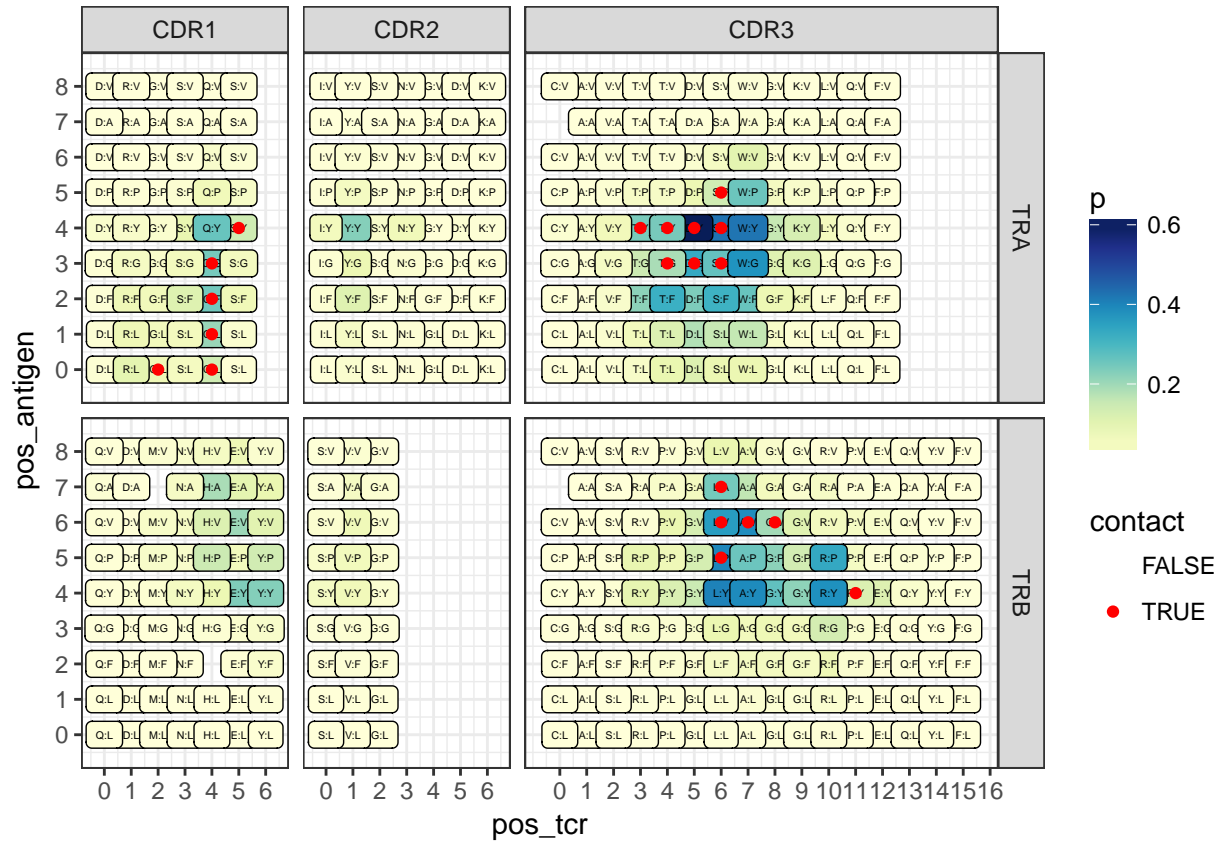


and LLFGYPVAV

```
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

```
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
}
```

```

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/> - engineered peptide

```

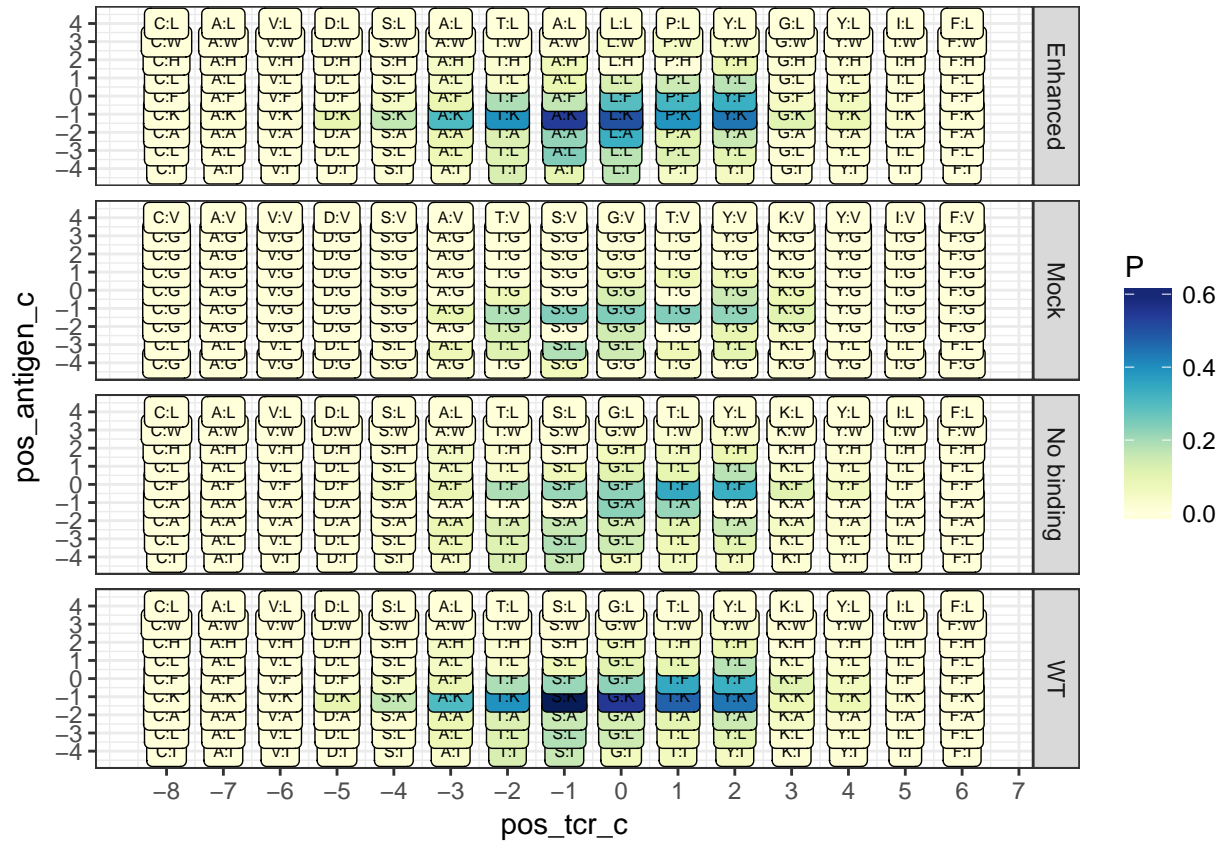
cm.tmp = compute_contact_map("MHCI", "TRA", "CDR3", "CAVDSATSGTYKYIF", "ILAKFLHWL", "WT")
cm.tmp = rbind(cm.tmp, compute_contact_map("MHCI", "TRA", "CDR3", "CAVDSATSGTYKYIF", "ILAAFLHWL", "No binding
cm.tmp = rbind(cm.tmp, compute_contact_map("MHCI", "TRA", "CDR3", "CAVDSATSGTYKYIF", "GLGGGGGGV", "Mock"))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

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 8.070395 -57.10482
## 2 Mock     3.294718 -17.97989
## 3 No binding 5.040805 -24.11150
## 4 WT       7.846778 -54.21853

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()
```



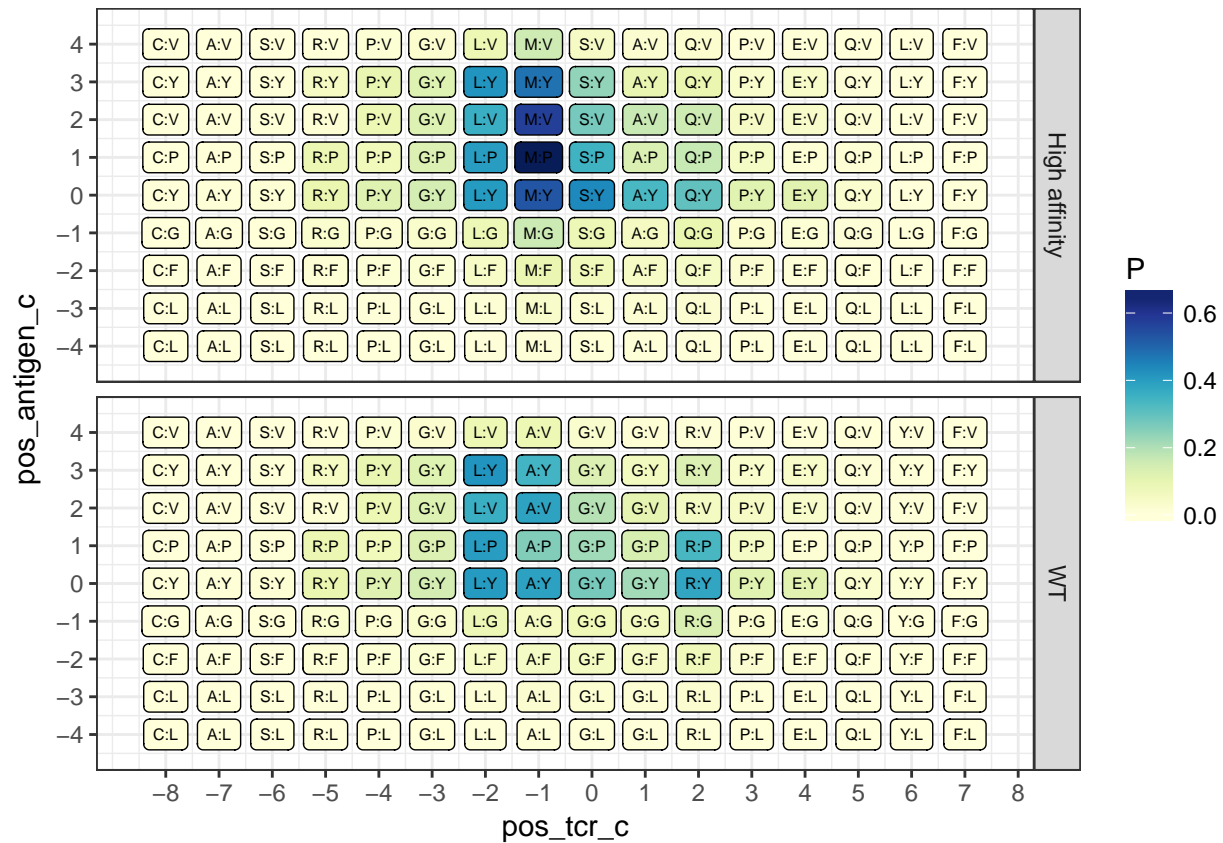

Testing - example 2 <http://www.nature.com/articles/ncomms6223>, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3049343/> - high affinity Tax mutant

```
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 aff"))
```

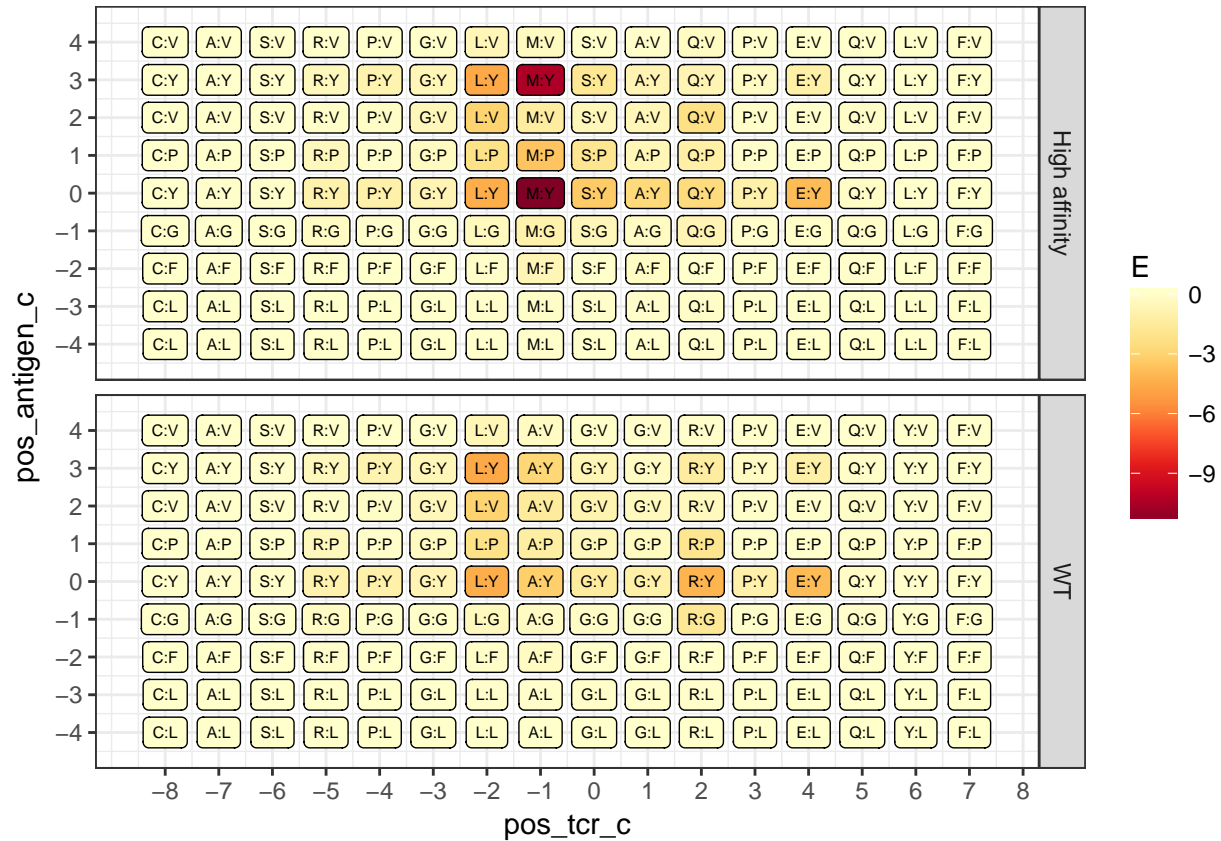
```
print(cm.tmp %>% group_by(id) %>% summarize(contacts = sum(p), energy = sum(p * energy.mean)))
```

```
## # A tibble: 2 × 3
##       id contacts    energy
##   <chr>   <dbl>   <dbl>
## 1 High affinity 9.608607 -82.49189
## 2           WT 7.998641 -57.89135
```

```
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()
```



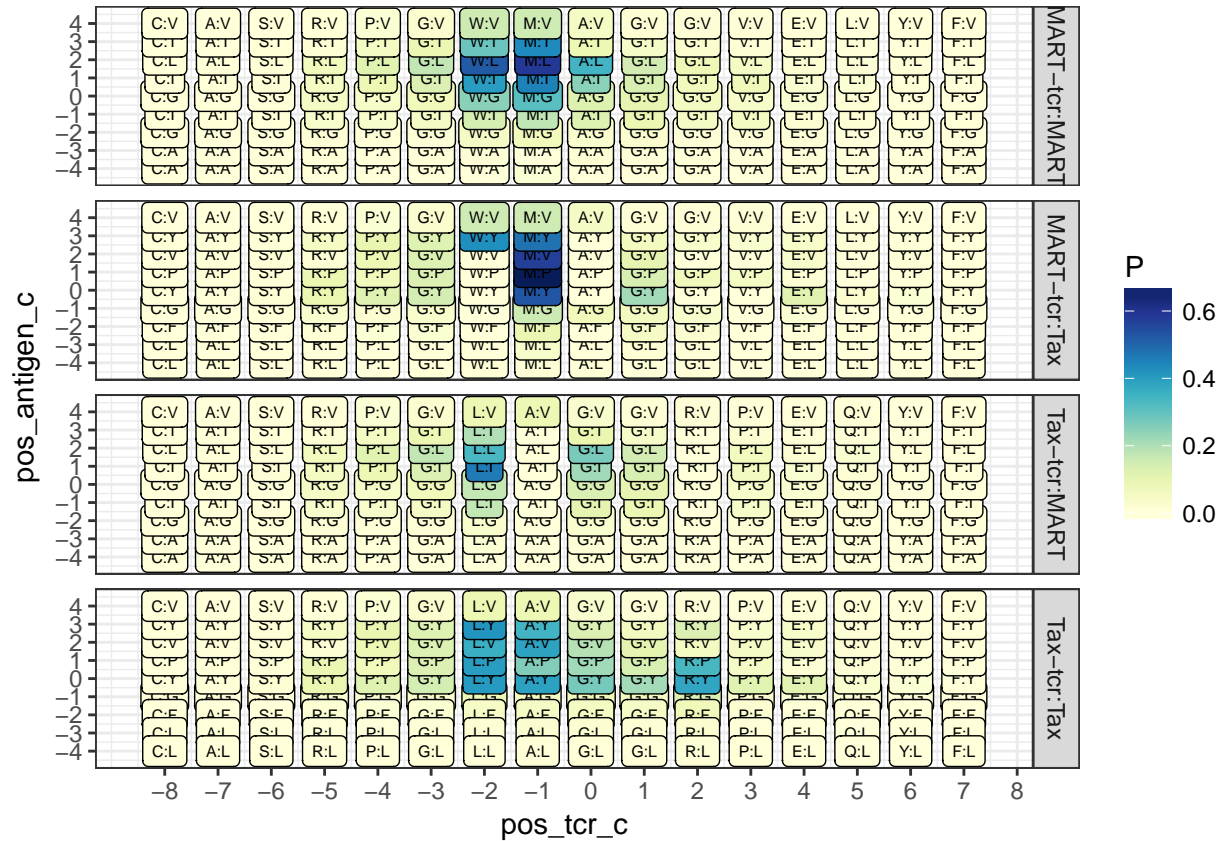
Testing - example 3 same as above - comparing Tax specific vs Tax and MART specific vs MART + cross-comparison. We compare CDR3beta of A6 (wild-type Tax-specific variant) and a MART-specific TCR derived from A6 by direct evolution.

```
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-
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-

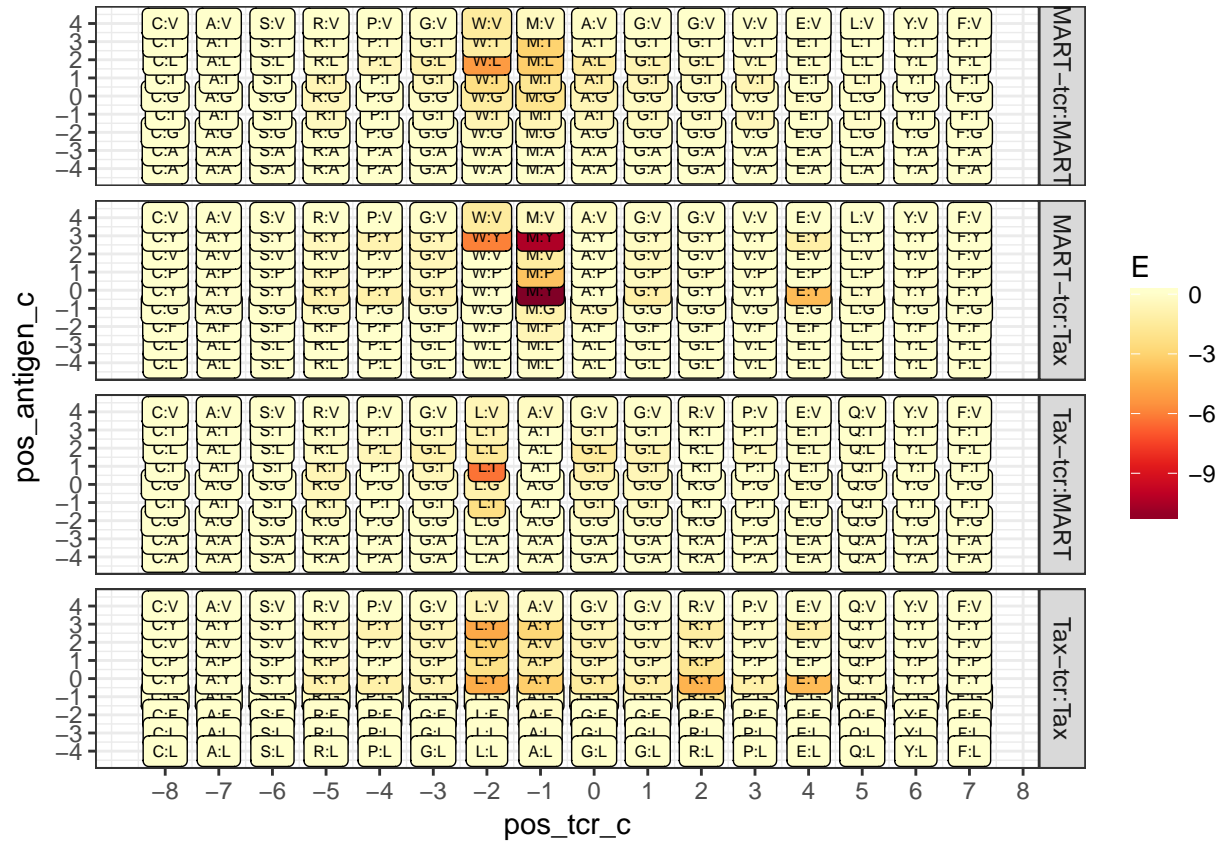
print(cm.tmp.1 %>% group_by(id) %>% summarize(contacts = sum(p), energy = sum(p * energy.mean)))

## # A tibble: 4 × 3
##       id contacts    energy
##   <chr>    <dbl>    <dbl>
## 1 MART-tcr:MART 7.216964 -41.94367
## 2 MART-tcr:Tax 5.841405 -54.07126
## 3 Tax-tcr:MART 4.391924 -27.59468
## 4 Tax-tcr:Tax 7.998641 -57.89135

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()
```



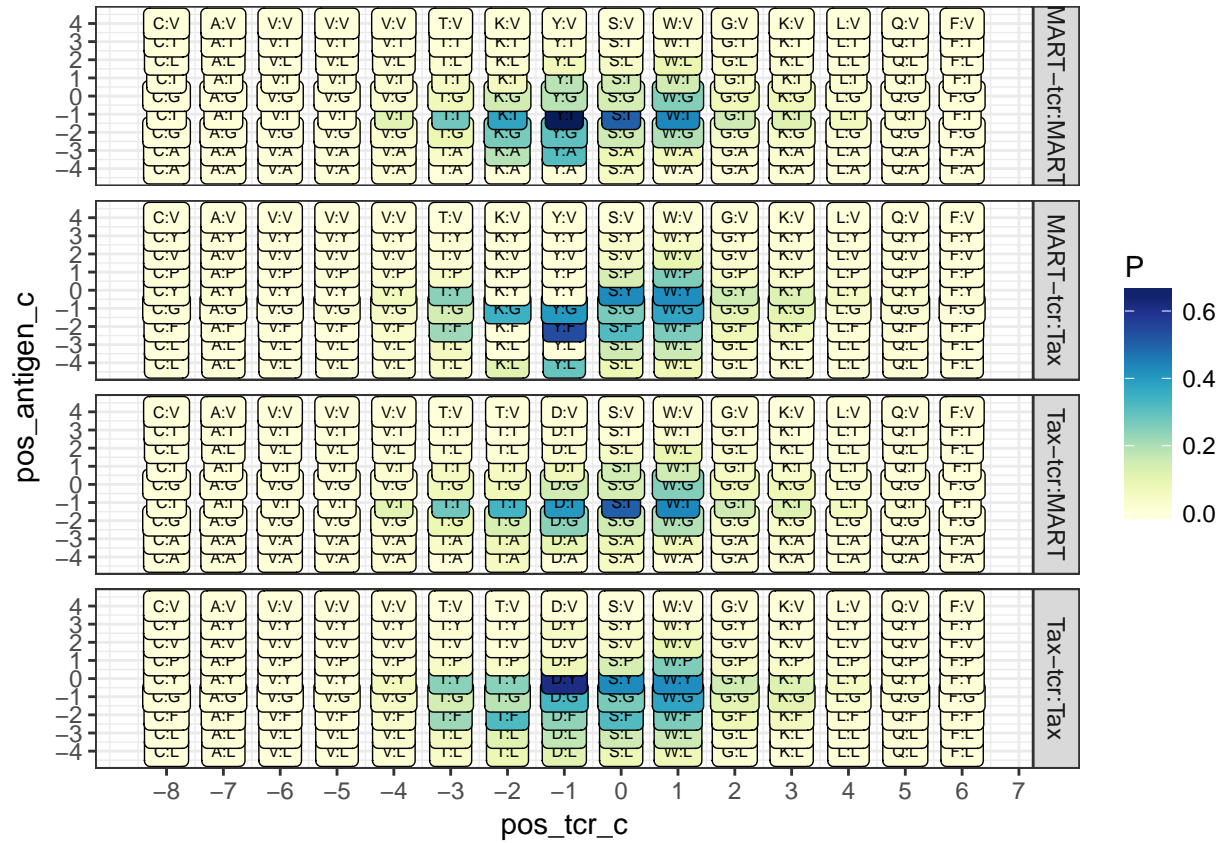
Same as above for alpha chains

```
cm.tmp.2 = compute_contact_map("MHCI","TRA","CDR3","CAVVVTDSWGKQLQF","LLFGYPVYV","Tax-tcr:Tax")
cm.tmp.2 = rbind(cm.tmp.2, compute_contact_map("MHCI","TRA","CDR3","CAVVVTDSWGKQLQF","AAGIGILTV","Tax-tcr:Tax"))
cm.tmp.2 = rbind(cm.tmp.2, compute_contact_map("MHCI","TRA","CDR3","CAVVVTKYSWGKQLQF","LLFGYPVYV","MART-tcr:MART"))
cm.tmp.2 = rbind(cm.tmp.2, compute_contact_map("MHCI","TRA","CDR3","CAVVVTKYSWGKQLQF","AAGIGILTV","MART-tcr:Tax"))

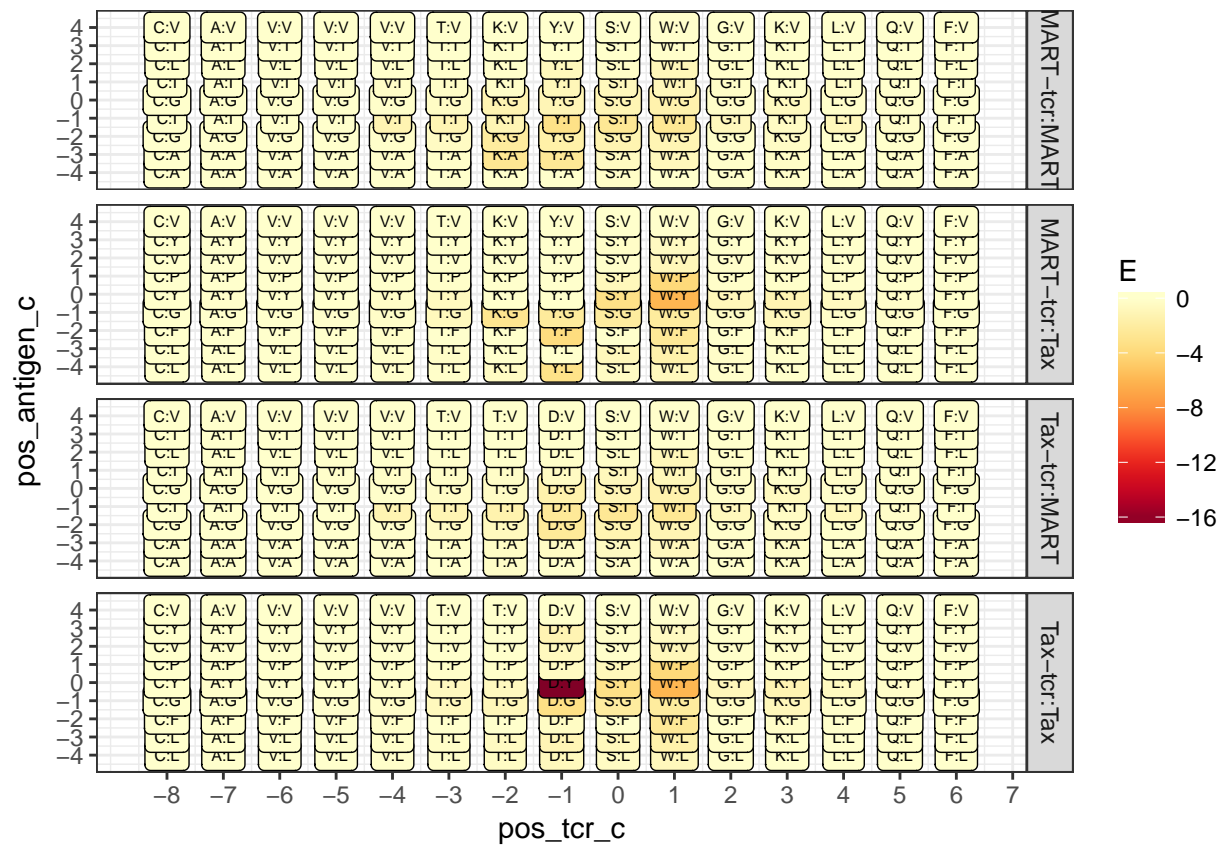
print(cm.tmp.2 %>% group_by(id) %>% summarize(contacts = sum(p), energy = sum(p * energy.mean)))

## # A tibble: 4 × 3
##       id contacts    energy
##   <chr>    <dbl>    <dbl>
## 1 MART-tcr:MART 6.710023 -41.32907
## 2 MART-tcr:Tax 6.848372 -47.55246
## 3 Tax-tcr:MART 5.617401 -34.19848
## 4 Tax-tcr:Tax 7.744718 -62.10414

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 * energy)))
```

```
## # A tibble: 4 × 3
##       id contacts    energy
##   <chr>    <dbl>    <dbl>
## 1 MART-tcr:MART 13.92699 -83.27275
## 2 MART-tcr:Tax 12.68978 -101.62373
## 3 Tax-tcr:MART 10.00933 -61.79316
## 4 Tax-tcr:Tax 15.74336 -119.99550
```