Mining CDR-like loops

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(gglogo)
```

```
##
## Attaching package: 'gglogo'
```

```
## The following object is masked from 'package:ggplot2':
##
##     fortify
```

```
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

 if (numPlots==1) {
    print(plots[[1]])

 } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))
```

```
      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

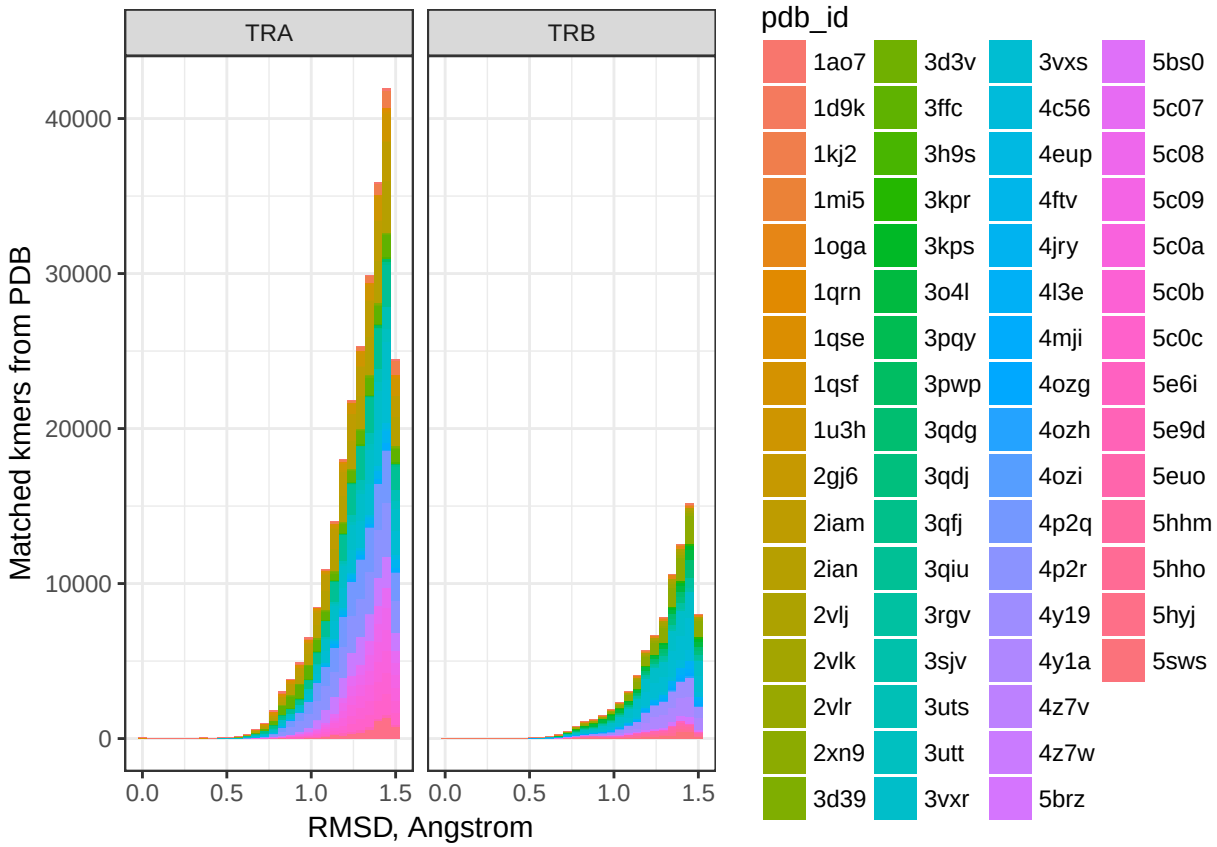
Summary

The analysis is limited to CDR-like loops with length 13, RMSD threshold is 1.5A

```
df.rmsd = read.table("loops_13.rmsd_stat.txt.gz", header = T, sep = "\t")
```

```
ggplot(df.rmsd, aes(x=rmsd, fill = pdb_id)) +
  geom_histogram() + facet_wrap(~tcr_chain) +
  xlab("RMSD, Angstrom") + ylab("Matched kmers from PDB") +
  theme_bw()
```

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



Fetched data

Load

```
df.cdr.pdb = read.table("loops_13.putative_cdr.txt.gz", header = T, sep = "\t")
df.cdr.real = read.table("loops_13.real_cdr.txt.gz", header = T, sep = "\t")
```

Check for presence of canonical CDR-like structures, i.e. starting with Cys and ending with Phe/Trp

```
df.cdr.canon = df.cdr.pdb %>%
  filter((aa_kmer == "C" & pos_tcr == 0) | (aa_kmer %in% c("F", "W") & pos_tcr == len_tcr - 1)) %>%
  select(pdb_id_kmer, chain_id_kmer, start_kmer, pos_tcr, len_tcr, aa_kmer) %>%
  unique() %>%
  group_by(pdb_id_kmer, chain_id_kmer, start_kmer, len_tcr) %>%
  summarize(canon = n()) %>%
  filter(canon == 2) %>%
  select(pdb_id_kmer, chain_id_kmer, start_kmer, len_tcr) %>%
  mutate(in_tcr_db = pdb_id_kmer %in% df.cdr.real$pdb_id)

print(summary(df.cdr.canon))
```

```
##   pdb_id_kmer  chain_id_kmer   start_kmer       len_tcr   in_tcr_db
##  3pnw   :  8   A      : 73   Min.   :  9.0   Min.   :13   Mode :logical
##  4xwo   :  8   D      : 55   1st Qu.: 86.0   1st Qu.:13   FALSE:281
##  3j2t   :  7   E      : 43   Median : 87.0   Median :13   TRUE :114
##  3d5o   :  5   B      : 41   Mean   :120.9   Mean   :13   NA's :0
##  1epf   :  4   L      : 36   3rd Qu.: 89.0   3rd Qu.:13
##  1nfd   :  4   C      : 35   Max.   :861.0   Max.   :13
##  (Other):359   (Other):112
```

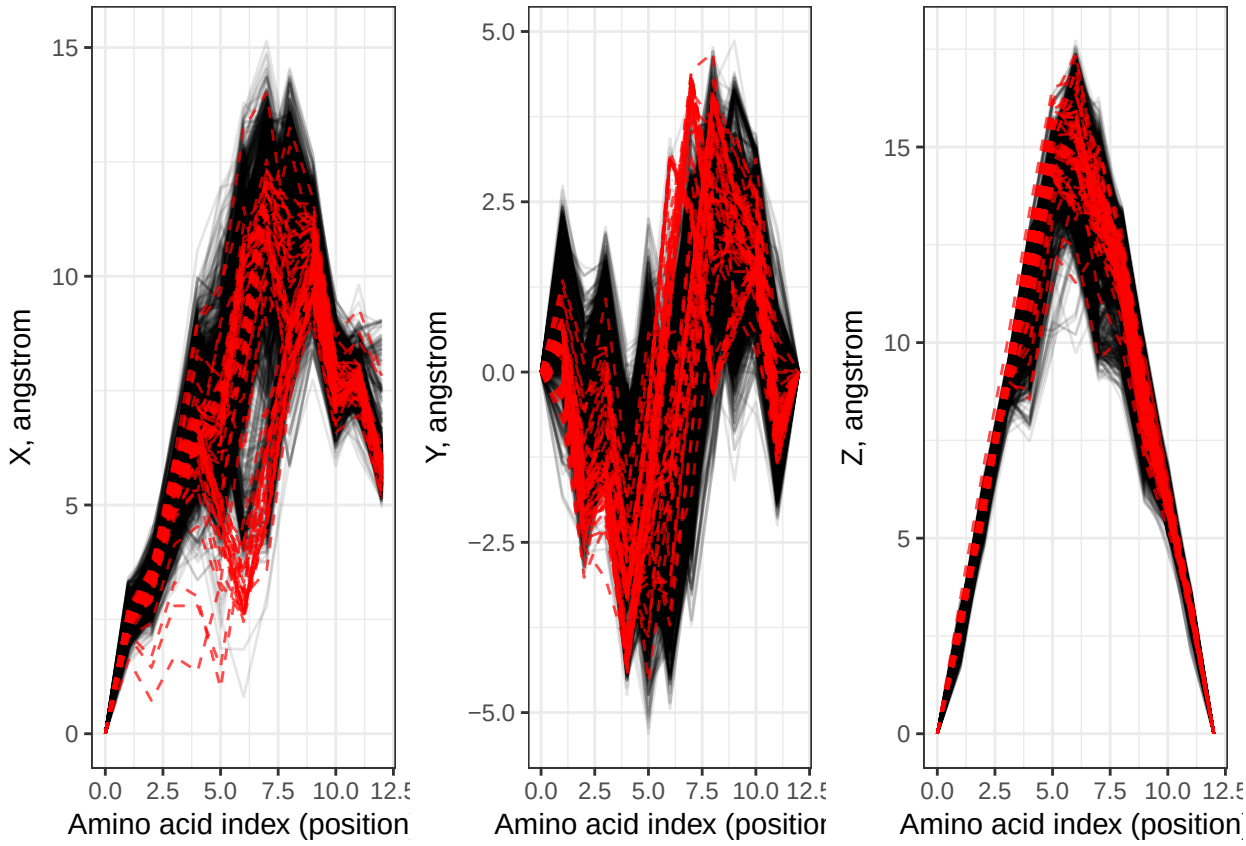
Coordinates

Take first 1000 PDB matches

```
sampled_pdbs = unique(df.cdr.pdb$pdb_id_kmer)[1:1000]
```

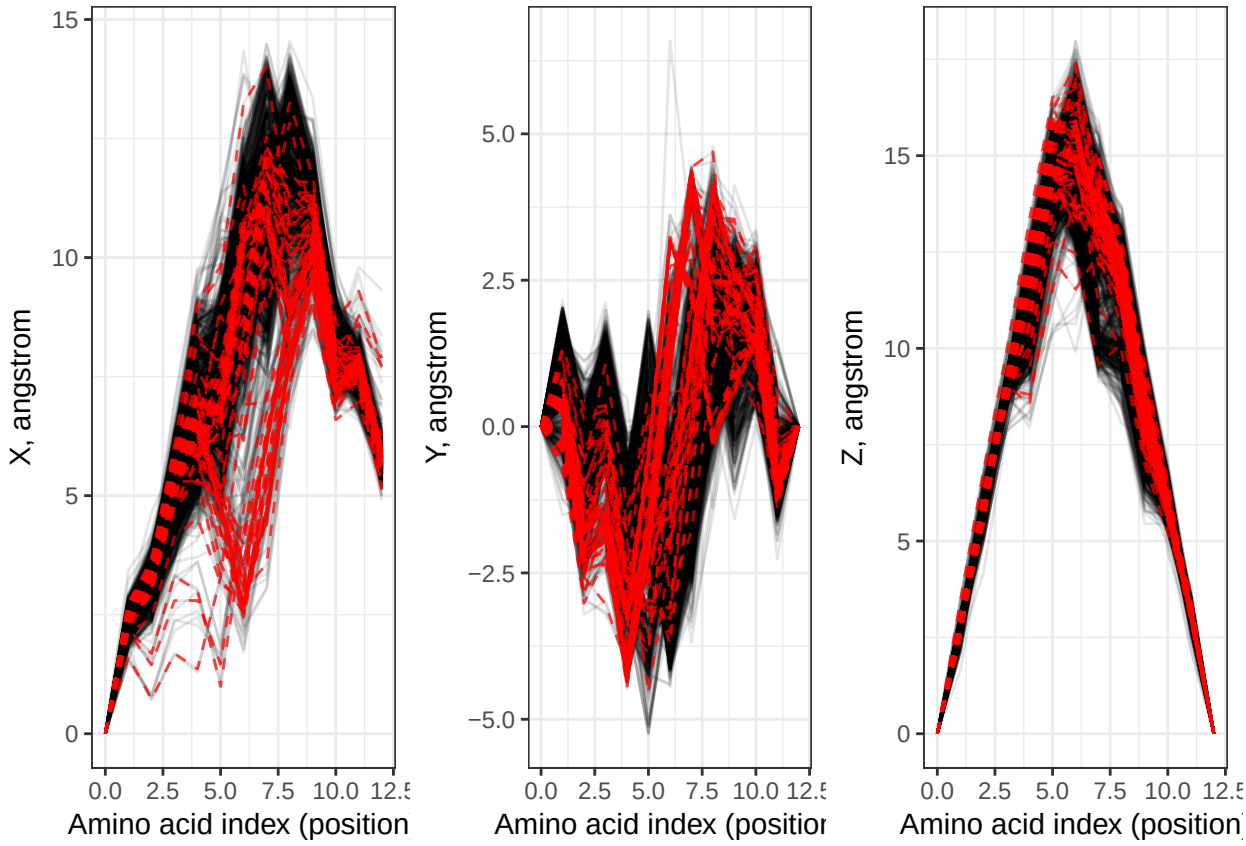
Plot known CDR (red) and matched region (black) in x, y and z coordinates

```
p1 = ggplot() +
  geom_line(data=df.cdr.pdb %>% filter(pdb_id_kmer %in% sampled_pdbs), aes(x=pos_tcr, y=x_kmer,
                                                                           group = paste(pdb_id_kmer, chain_
  geom_line(data=df.cdr.real, aes(x=pos_tcr, y=x, group = interaction(pdb_id, tcr_chain)), color = "red"
  xlab("Amino acid index (position)") + ylab("X, angstrom") +
  theme_bw()

p2 = ggplot() +
  geom_line(data=df.cdr.pdb %>% filter(pdb_id_kmer %in% sampled_pdbs), aes(x=pos_tcr, y=y_kmer,
                                                                           group = paste(pdb_id_kmer, chain_
  geom_line(data=df.cdr.real, aes(x=pos_tcr, y=y, group = interaction(pdb_id, tcr_chain)), color = "red"
  xlab("Amino acid index (position)") + ylab("Y, angstrom") +
  theme_bw()

p3 = ggplot() +
  geom_line(data=df.cdr.pdb %>% filter(pdb_id_kmer %in% sampled_pdbs), aes(x=pos_tcr, y=z_kmer,
                                                                           group = paste(pdb_id_kmer, chain_
  geom_line(data=df.cdr.real, aes(x=pos_tcr, y=z, group = interaction(pdb_id, tcr_chain)), color = "red"
  xlab("Amino acid index (position)") + ylab("Z, angstrom") +
  theme_bw()

multiplot(p1, p2, p3, cols=3)
```



Same for canonical k-mers

```
p1 = ggplot() +
  geom_line(data=df.cdr.pdb %>% filter(pdb_id_kmer %in% df.cdr.canon$pdb_id_kmer), aes(x=pos_tcr, y=x_km
                                                                                       group = paste(pdb_id_kmer, chain_
  geom_line(data=df.cdr.real, aes(x=pos_tcr, y=x, group = interaction(pdb_id, tcr_chain)), color = "red
  xlab("Amino acid index (position)") + ylab("X, angstrom") +
  theme_bw()

p2 = ggplot() +
  geom_line(data=df.cdr.pdb %>% filter(pdb_id_kmer %in% df.cdr.canon$pdb_id_kmer), aes(x=pos_tcr, y=y_km
                                                                                       group = paste(pdb_id_kmer, chain_
  geom_line(data=df.cdr.real, aes(x=pos_tcr, y=y, group = interaction(pdb_id, tcr_chain)), color = "red
  xlab("Amino acid index (position)") + ylab("Y, angstrom") +
  theme_bw()

p3 = ggplot() +
  geom_line(data=df.cdr.pdb %>% filter(pdb_id_kmer %in% df.cdr.canon$pdb_id_kmer), aes(x=pos_tcr, y=z_km
                                                                                       group = paste(pdb_id_kmer, chain_
  geom_line(data=df.cdr.real, aes(x=pos_tcr, y=z, group = interaction(pdb_id, tcr_chain)), color = "red
  xlab("Amino acid index (position)") + ylab("Z, angstrom") +
  theme_bw()

multiplot(p1, p2, p3, cols=3)
```



Amino acid composition

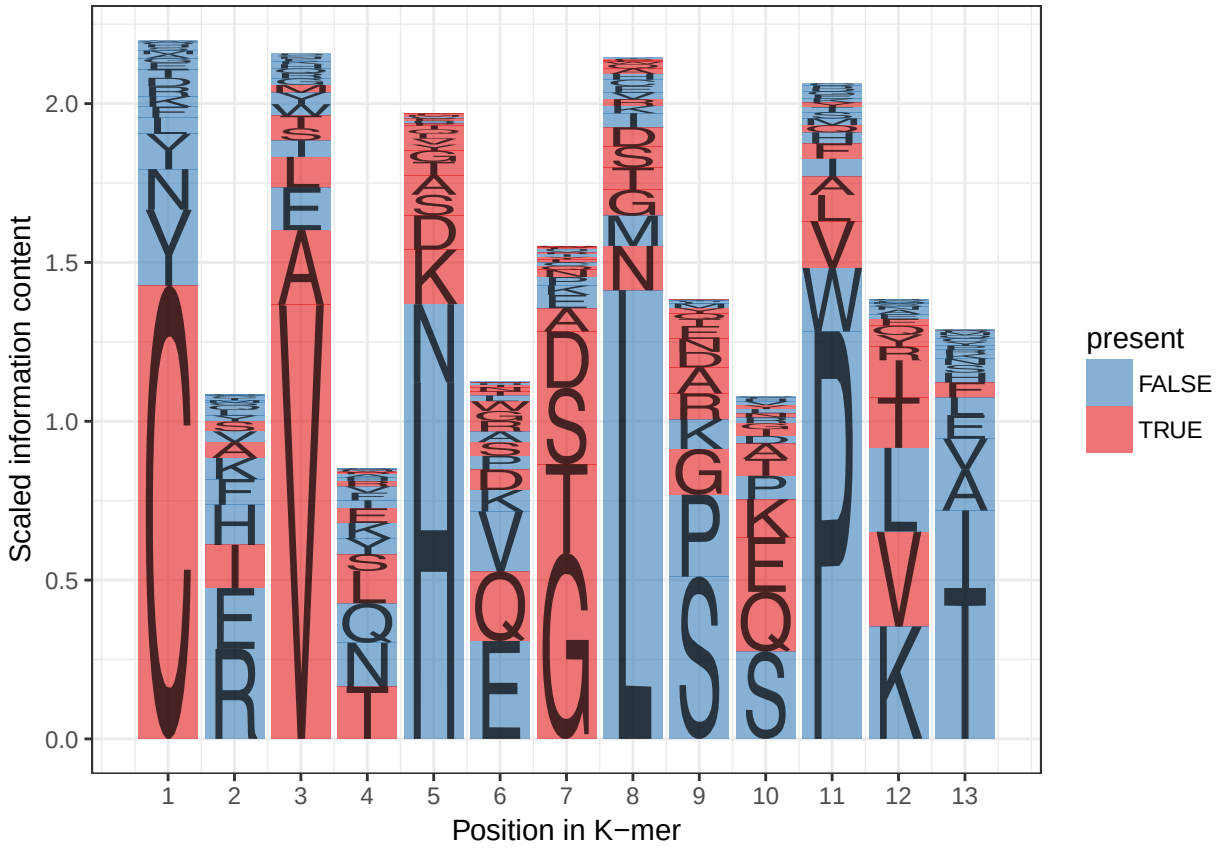
Bulk amino acid composition

```
get_aa_freqs = function(rmsd_threshold = 1, only_canonical = F) {
  tmp = df.cdr.pdb
  if (only_canonical) {
    tmp = merge(tmp, df.cdr.canon %>% filter(!in_tcr_db))
  }

  tmp = tmp %>%
      filter(rmsd < rmsd_threshold) %>%
      select(pdb_id_kmer, chain_id_kmer, start_kmer, pos_tcr, len_tcr, aa_kmer) %>%
      unique() %>%
      group_by(pos_tcr, len_tcr, aa_kmer) %>%
      summarize(count = n()) %>%
      group_by(pos_tcr, len_tcr) %>%
      mutate(freq = count / sum(count), I = freq * (log2(20) - sum(-freq * log2(freq)) ))


  tmp = merge(tmp, df.cdr.real %>%
                       select(pos_tcr, len_tcr, aa_tcr) %>%
                       unique() %>%
                       mutate(aa_kmer = aa_tcr, present = T), all.x = T)

  tmp$present[is.na(tmp$present)] = F
```

```
  as.data.frame(tmp)
}
```

Sequence logo (http://genome.cshlp.org/content/14/6/1188.full) representation of amino acid frequencies vs position in K-mer. Amino acids found in real CDR3s in corresponding position are shown with red. Note there is likely some 1-2 AA spaced motif.

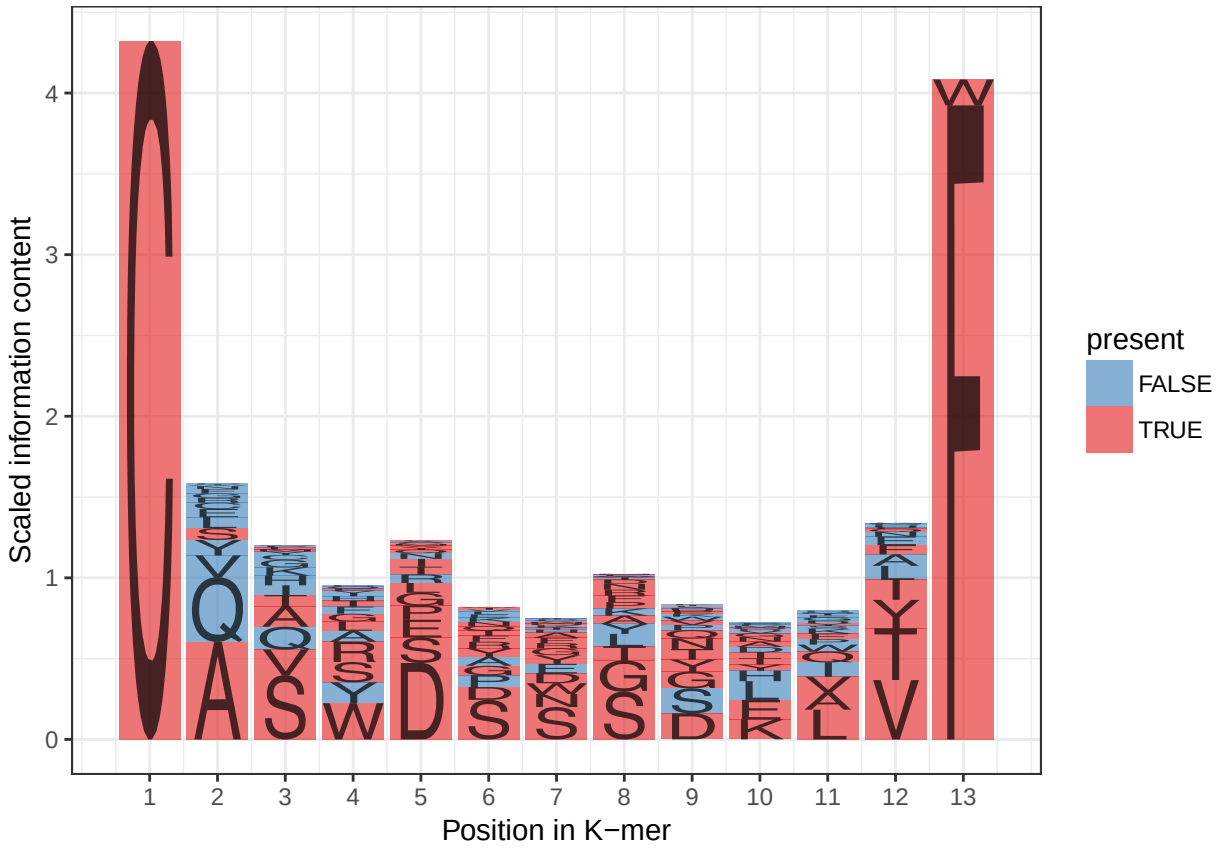
```
df.cdr.aapor = get_aa_freqs()

ggplot(df.cdr.aapor) +
  geom_logo(aes(x=pos_tcr+1, y=I, group=aa_kmer,
     label=aa_kmer, fill = present), position="classic", color=NA)  +
  ylab("Scaled information content") +
  scale_x_continuous("Position in K-mer", breaks = 1:13, limits=c(0.5,13.5)) +
  scale_fill_brewer(palette = "Set1", direction = -1) +
  theme_bw()
```



Same for canonical kmers, not much motif here:

```
df.cdr.aapor.c = get_aa_freqs(1.5, T)

ggplot(df.cdr.aapor.c) +
  geom_logo(aes(x=pos_tcr + 1, y=I, group=aa_kmer,
     label=aa_kmer, fill = present), position="classic", color=NA)  +
  ylab("Scaled information content") +
  scale_x_continuous("Position in K-mer", breaks = 1:13, limits=c(0.5,13.5)) +
  scale_fill_brewer(palette = "Set1", direction = -1) +
```

```
theme_bw()
```


Scaled information content
0
1
2
3
4
1
2
3
4
5
6
7
8
9
10
11
12
13
Position in K−mer
present
FALSE
TRUE