Somatic hypermutations signatures

Anna Obraztsova

4/20/2017

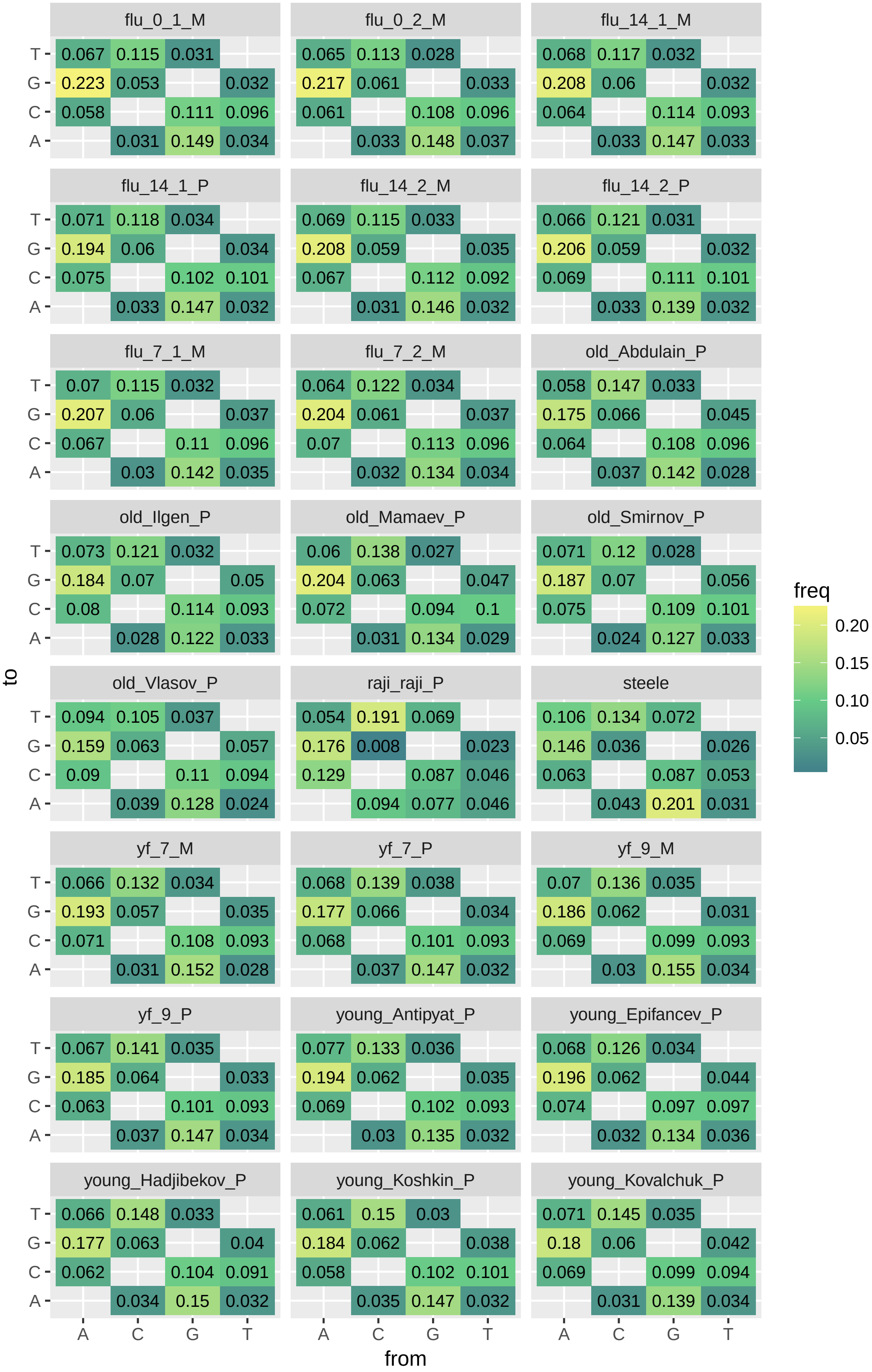
```r
library(dplyr)
```

```
##
## 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
```

```r
library(ggplot2)
library(reshape2)
library(stringr)
library(tidyr)
```

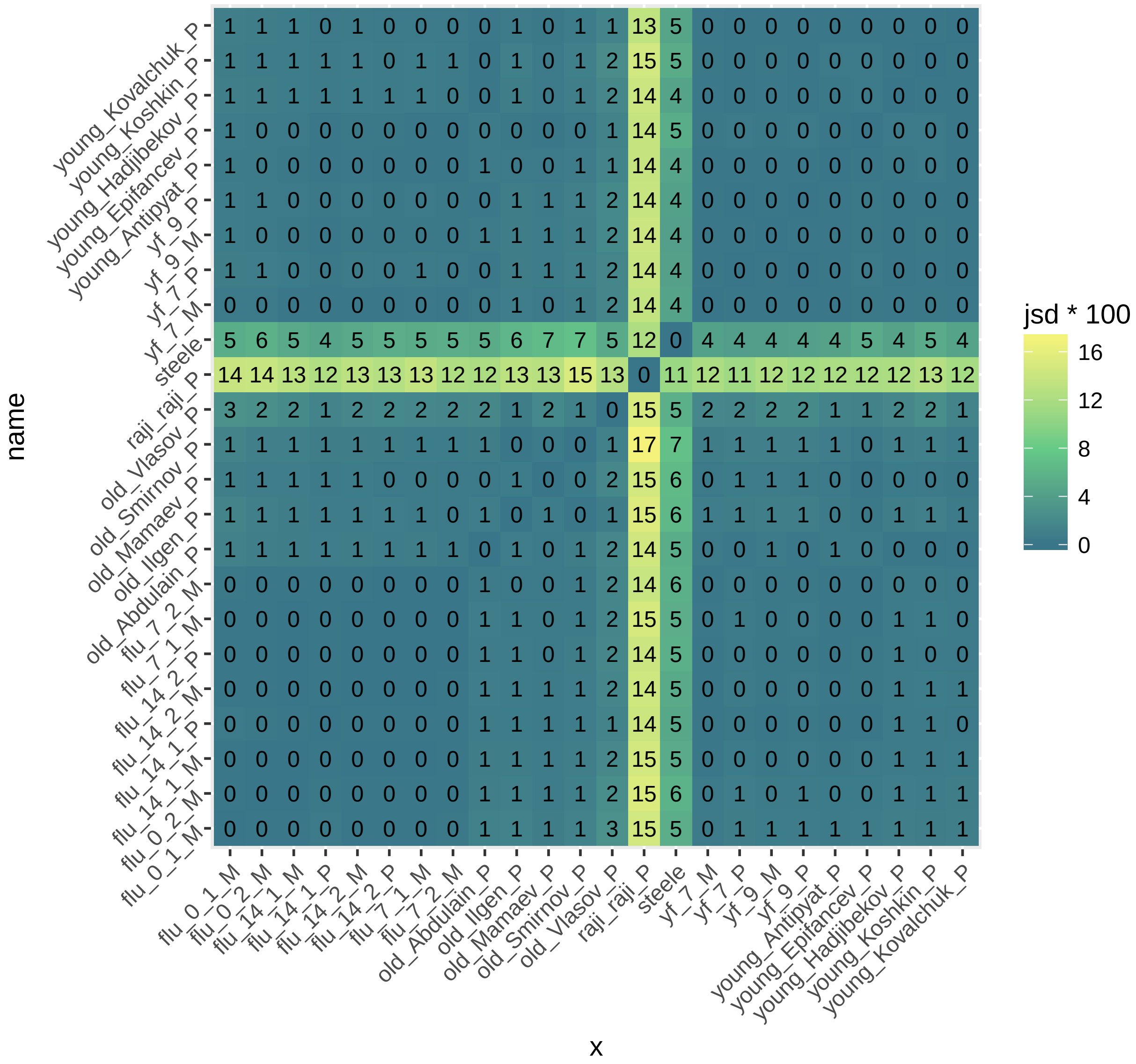
```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
##
##     smiths
```

```r
load('shm.Rda')
shm$region <- factor(shm$region, c("FR1","CDR1","FR2","CDR2","FR3","CDR3"))
shm$clonotypes <- 1
shm$name <- paste(shm$proj, shm$sample, shm$cells, sep='_')

contig.melted <- shm %>% dplyr::select(proj, sample, cells, name, contignt) %>%
  unique() %>%
  mutate(nt = str_split(contignt, '')) %>%
  unnest(nt = nt)

contig.total <- contig.melted %>% dplyr::group_by(proj, sample, cells, name) %>%
  dplyr::summarise(total.nt = n())

contig.freq <- contig.melted %>% dplyr::group_by(proj, sample, cells, nt, name) %>%
  dplyr::summarise(count = n()) %>%
  merge(contig.total) %>%
  mutate(nt.freq = count/total.nt)

total.mutations <- shm %>% dplyr::group_by(proj, sample, cells, name) %>%
  dplyr::summarise(total.mutations = n())

f <- shm %>% dplyr::group_by(proj, sample, cells, name, from, to) %>%
  dplyr::summarise(total.clonotypes = sum(clonotypes)) %>%
  merge(dplyr::select(contig.freq, from=nt, proj, sample, cells, nt.freq, name)) %>%
  merge(total.mutations) %>%
  mutate(weight = total.clonotypes/(total.mutations*nt.freq))

f1 <- f %>% dplyr::group_by(proj, sample, cells, name) %>%
  dplyr::summarise(weight.total = sum(weight))

f <- f %>% merge(f1) %>%
  mutate(freq = weight / weight.total) %>%
  dplyr::select(proj, sample, cells, from, to, freq, name)

st <- read.table('freq_steele.txt', header=T) %>%
  mutate(proj = 'steele', sample = 'steele', cells='N', name = 'steele',
         freq = freq/100)

f <- rbind(f, st)

ggplot(f, aes(x = from, y = to)) +
  geom_tile(aes(fill = freq)) +
  geom_text(aes(label = round(freq, 3)), cex=3) +
  facet_wrap(~name, ncol=3) +
  scale_fill_gradient2(low = "#2F6B89", mid = "#67CB87", high = "#F4F27B", midpoint = 0.1, na.value = "
```


flu_0_1_M
T 0.067 0.115 0.031
G 0.223 0.053 0.032
C 0.058 0.111 0.096
A 0.031 0.149 0.034
flu_0_2_M
T 0.065 0.113 0.028
G 0.217 0.061 0.033
C 0.061 0.108 0.096
A 0.033 0.148 0.037
flu_14_1_M
T 0.068 0.117 0.032
G 0.208 0.06 0.032
C 0.064 0.114 0.093
A 0.033 0.147 0.033
flu_14_1_P
T 0.071 0.118 0.034
G 0.194 0.06 0.034
C 0.075 0.102 0.101
A 0.033 0.147 0.032
flu_14_2_M
T 0.069 0.115 0.033
G 0.208 0.059 0.035
C 0.067 0.112 0.092
A 0.031 0.146 0.032
flu_14_2_P
T 0.066 0.121 0.031
G 0.206 0.059 0.032
C 0.069 0.111 0.101
A 0.033 0.139 0.032
flu_7_1_M
T 0.07 0.115 0.032
G 0.207 0.06 0.037
C 0.067 0.11 0.096
A 0.03 0.142 0.035
flu_7_2_M
T 0.064 0.122 0.034
G 0.204 0.061 0.037
C 0.07 0.113 0.096
A 0.032 0.134 0.034
old_Abdulain_P
T 0.058 0.147 0.033
G 0.175 0.066 0.045
C 0.064 0.108 0.096
A 0.037 0.142 0.028
old_Ilgen_P
T 0.073 0.121 0.032
G 0.184 0.07 0.05
C 0.08 0.114 0.093
A 0.028 0.122 0.033
old_Mamaev_P
T 0.06 0.138 0.027
G 0.204 0.063 0.047
C 0.072 0.094 0.1
A 0.031 0.134 0.029
old_Smirnov_P
T 0.071 0.12 0.028
G 0.187 0.07 0.056
C 0.075 0.109 0.101
A 0.024 0.127 0.033
old_Vlasov_P
T 0.094 0.105 0.037
G 0.159 0.063 0.057
C 0.09 0.11 0.094
A 0.039 0.128 0.024
raji_raji_P
T 0.054 0.191 0.069
G 0.176 0.008 0.023
C 0.129 0.087 0.046
A 0.094 0.077 0.046
steele
T 0.106 0.134 0.072
G 0.146 0.036 0.026
C 0.063 0.087 0.053
A 0.043 0.201 0.031
yf_7_M
T 0.066 0.132 0.034
G 0.193 0.057 0.035
C 0.071 0.108 0.093
A 0.031 0.152 0.028
yf_7_P
T 0.068 0.139 0.038
G 0.177 0.066 0.034
C 0.068 0.101 0.093
A 0.037 0.147 0.032
yf_9_M
T 0.07 0.136 0.035
G 0.186 0.062 0.031
C 0.069 0.099 0.093
A 0.03 0.155 0.034
yf_9_P
T 0.067 0.141 0.035
G 0.185 0.064 0.033
C 0.063 0.101 0.093
A 0.037 0.147 0.034
young_Antipyat_P
T 0.077 0.133 0.036
G 0.194 0.062 0.035
C 0.069 0.102 0.093
A 0.03 0.135 0.032
young_Epifancev_P
T 0.068 0.126 0.034
G 0.196 0.062 0.044
C 0.074 0.097 0.097
A 0.032 0.134 0.036
young_Hadjibekov_P
T 0.066 0.148 0.033
G 0.177 0.063 0.04
C 0.062 0.104 0.091
A 0.034 0.15 0.032
young_Koshkin_P
T 0.061 0.15 0.03
G 0.184 0.062 0.038
C 0.058 0.102 0.101
A 0.035 0.147 0.032
young_Kovalchuk_P
T 0.071 0.145 0.035
G 0.18 0.06 0.042
C 0.069 0.099 0.094
A 0.031 0.139 0.034
A C G T
from
to
freq
0.20
0.15
0.10
0.05

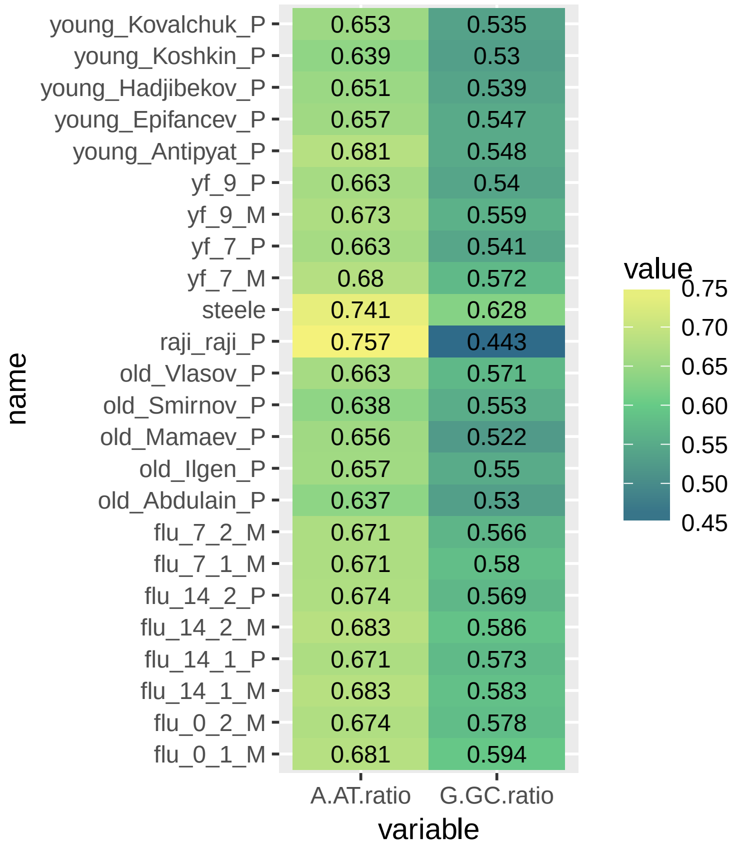
Calculate Jensen-Shannon divergence

```r
jsd <- function(v1, v2){
  m <- 0.5 * (v1 + v1)
  0.5 * (sum(v1 * log2(v1 / m)) + sum(v2 * log2(v2 / m)))
}

f <- f[ order(f$from, f$to), ]

jsd.df <- data.frame()

for (i in unique(f$name)){
  x <- filter(f, name == i)$freq

  .jsd.df <- f %>% dplyr::group_by(name) %>%
  dplyr::summarise(jsd = jsd(x, freq))

  .jsd.df$x <- i

  jsd.df <- rbind(jsd.df, .jsd.df)
}

ggplot(jsd.df, aes(x, name)) +
  geom_tile(aes(fill = jsd*100)) +
  geom_text(aes(label = round(jsd*100)), cex=3) +
  scale_fill_gradient2(low = "#2F6B89", mid = "#67CB87", high = "#F4F27B", midpoint = 8,
                       na.value = "white") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1),
          axis.text.y = element_text(angle = 45, hjust = 1))
```



	flu_0_1_M	flu_0_2_M	flu_14_1_M	flu_14_1_P	flu_14_2_M	flu_14_2_P	flu_7_1_M	flu_7_2_M	old_Abdulain_P	old_Ilgen_P	old_Mamaev_P	old_Smirnov_P	old_Vlasov_P	raji_raji_P	steele	yf_7_M	yf_7_P	yf_9_M	yf_9_P	young_Antipyat_P	young_Epifancev_P	young_Hadjibekov_P	young_Koshkin_P	young_Kovalchuk_P
young_Kovalchuk_P	1	1	1	0	1	0	0	0	0	1	0	1	1	13	5	0	0	0	0	0	0	0	0	0
young_Koshkin_P	1	1	1	1	1	0	1	1	0	1	0	1	2	15	5	0	0	0	0	0	0	0	0	0
young_Hadjibekov_P	1	1	1	1	1	1	1	0	0	1	0	1	2	14	4	0	0	0	0	0	0	0	0	0
young_Epifancev_P	1	0	0	0	0	0	0	0	0	0	0	0	1	14	5	0	0	0	0	0	0	0	0	0
young_Antipyat_P	1	0	0	0	0	0	0	0	1	0	0	1	1	14	4	0	0	0	0	0	0	0	0	0
yf_9_P	1	1	0	0	0	0	0	0	0	1	1	1	2	14	4	0	0	0	0	0	0	0	0	0
yf_9_M	1	0	0	0	0	0	0	0	1	1	1	1	2	14	4	0	0	0	0	0	0	0	0	0
yf_7_P	1	1	0	0	0	0	1	0	0	1	1	1	2	14	4	0	0	0	0	0	0	0	0	0
yf_7_M	0	0	0	0	0	0	0	0	0	1	0	1	2	14	4	0	0	0	0	0	0	0	0	0
steele	5	6	5	4	5	5	5	5	5	6	7	7	5	12	0	4	4	4	4	4	5	4	5	4
raji_raji_P	14	14	13	12	13	13	13	12	12	13	13	15	13	0	11	12	11	12	12	12	12	12	13	12
old_Vlasov_P	3	2	2	1	2	2	2	2	2	1	2	1	0	15	5	2	2	2	2	1	1	2	2	1
old_Smirnov_P	1	1	1	1	1	1	1	1	1	0	0	0	1	17	7	1	1	1	1	1	0	1	1	1
old_Mamaev_P	1	1	1	1	1	0	0	0	0	1	0	0	2	15	6	0	1	1	1	0	0	0	0	0
old_Ilgen_P	1	1	1	1	1	1	1	0	1	0	1	0	1	15	6	1	1	1	1	0	0	1	1	1
old_Abdulain_P	1	1	1	1	1	1	1	1	0	1	0	1	2	14	5	0	0	1	0	1	0	0	0	0
flu_7_2_M	0	0	0	0	0	0	0	0	1	0	0	1	2	14	6	0	0	0	0	0	0	0	0	0
flu_7_1_M	0	0	0	0	0	0	0	0	1	1	0	1	2	15	5	0	1	0	0	0	0	1	1	0
flu_14_2_P	0	0	0	0	0	0	0	0	1	1	0	1	2	14	5	0	0	0	0	0	0	1	0	0
flu_14_2_M	0	0	0	0	0	0	0	0	1	1	1	1	2	14	5	0	0	0	0	0	0	1	1	1
flu_14_1_P	0	0	0	0	0	0	0	0	1	1	1	1	1	14	5	0	0	0	0	0	0	1	1	0
flu_14_1_M	0	0	0	0	0	0	0	0	1	1	1	1	2	15	5	0	0	0	0	0	0	1	1	1
flu_0_2_M	0	0	0	0	0	0	0	0	1	1	1	1	2	15	6	0	1	0	1	0	0	1	1	1
flu_0_1_M	0	0	0	0	0	0	0	0	1	1	1	1	3	15	5	0	1	1	1	1	1	1	1	1

```
f2 <- f %>% dplyr::group_by(proj, sample, cells, name, from) %>%
  dplyr::summarise(freq = sum(freq)) %>%
  dcast(proj + sample + cells +name ~ from, value.var='freq')

f2$A.AT.ratio = f2$A/(f2$A+f2[['T']])
f2$G.GC.ratio = f2$G/(f2$G+f2$C)
f2 <- f2 %>%
  dplyr::select(name, A.AT.ratio, G.GC.ratio) %>%
  melt(id.vars=c('name'))

ggplot(f2, aes(variable, name)) +
  geom_tile(aes(fill = value)) +
  geom_text(aes(label = round(value, 3)), cex=3) +
  scale_fill_gradient2(low = "#2F6B89", mid = "#67CB87", high = "#F4F27B", midpoint = 0.6,
                       na.value = "white")
```

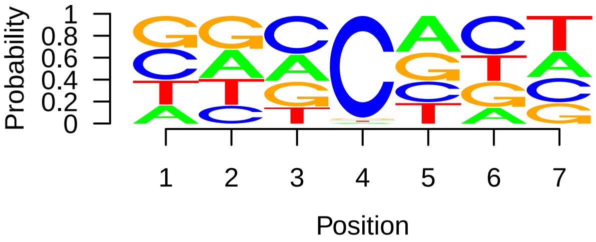


```
library(seqLogo)
```

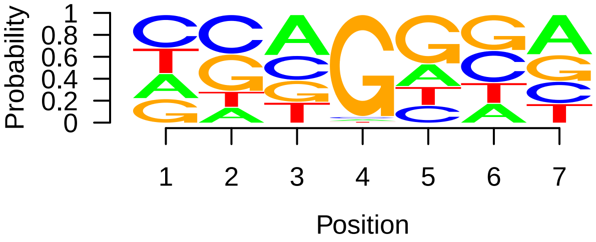
```
## Loading required package: grid
```

```
shm$type <- paste(shm$from, shm$to, sep='>')

for (i in unique(shm$type)){
  print(i)
  set <- filter(shm, type == i, nchar(context) == 7, str_detect(context, 'N') == F) %>%
    separate(context, c('n.1', 'n.2', 'n.3', 'n.4', 'n.5', 'n.6', 'n.7'), c(1,2,3,4,5,6)) %>%
    dplyr::select(starts_with('n.'))

  prob <- data.frame()

  for (j in c('A','C','G','T')){
    .prob <- apply(set, 2, function(x) length(x[x==j])/length(x))
    prob <- rbind(prob, .prob)
  }
  rownames(prob) <- c('A','C','G','T')
  colnames(prob) <- paste0('p', 1:7)
  seqLogo(makePWM(prob), ic.scale=F,  xfontsize=10, yfontsize=10)
}
```

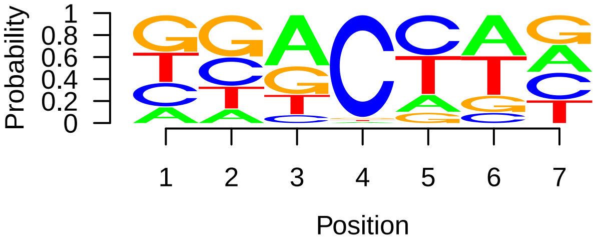
```
## [1] "T>C"
```



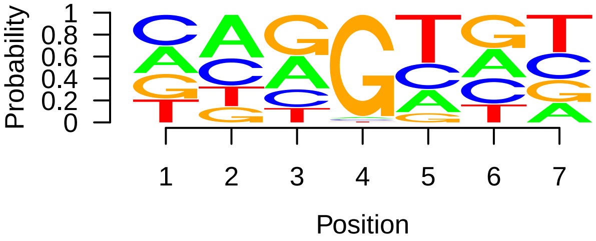
```
## [1] "A>G"
```



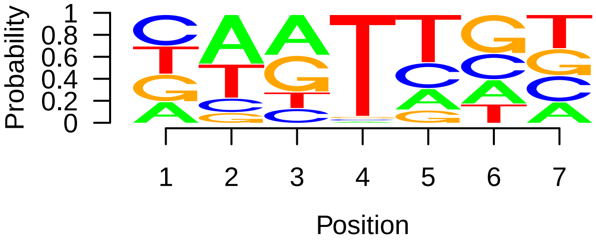
```
## [1] "G>C"
```



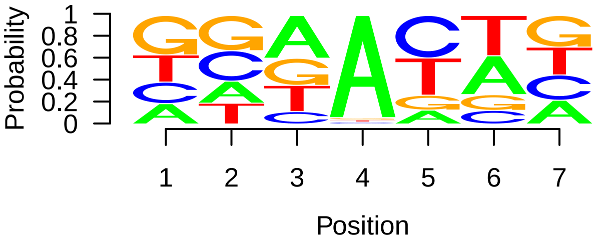
```
## [1] "C>G"
```



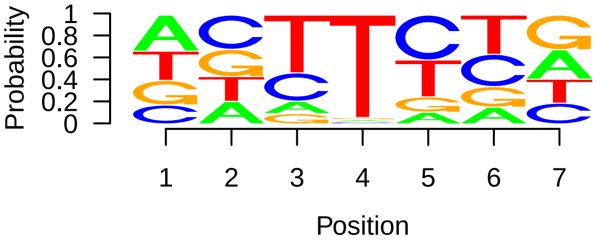
```
## [1] "C>T"
```



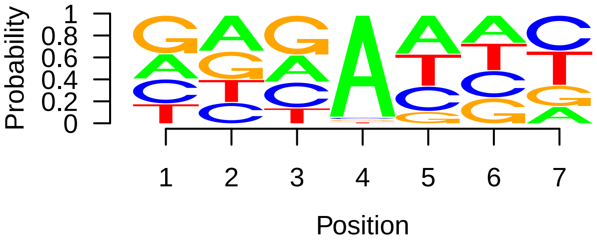
```
## [1] "G>A"
```



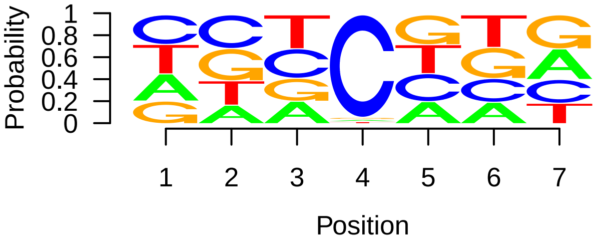
```
## [1] "A>T"
```



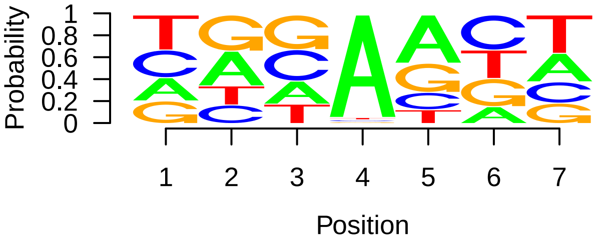
```
## [1] "C>A"
```



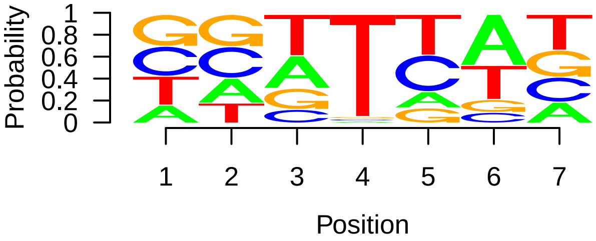
```
## [1] "A>C"
```



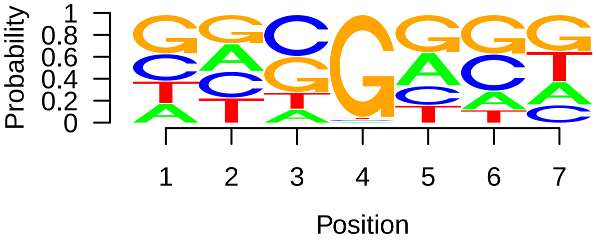
```
## [1] "T>A"
```



```
## [1] "G>T"
```


Probability
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
Position

```
## [1] "T>G"
```


Probability
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
Position