

Somatic hypermutations signatures

Anna Obratsova

4/20/2017

```
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(reshape2)
library(stringr)
library(tidyr)

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

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))

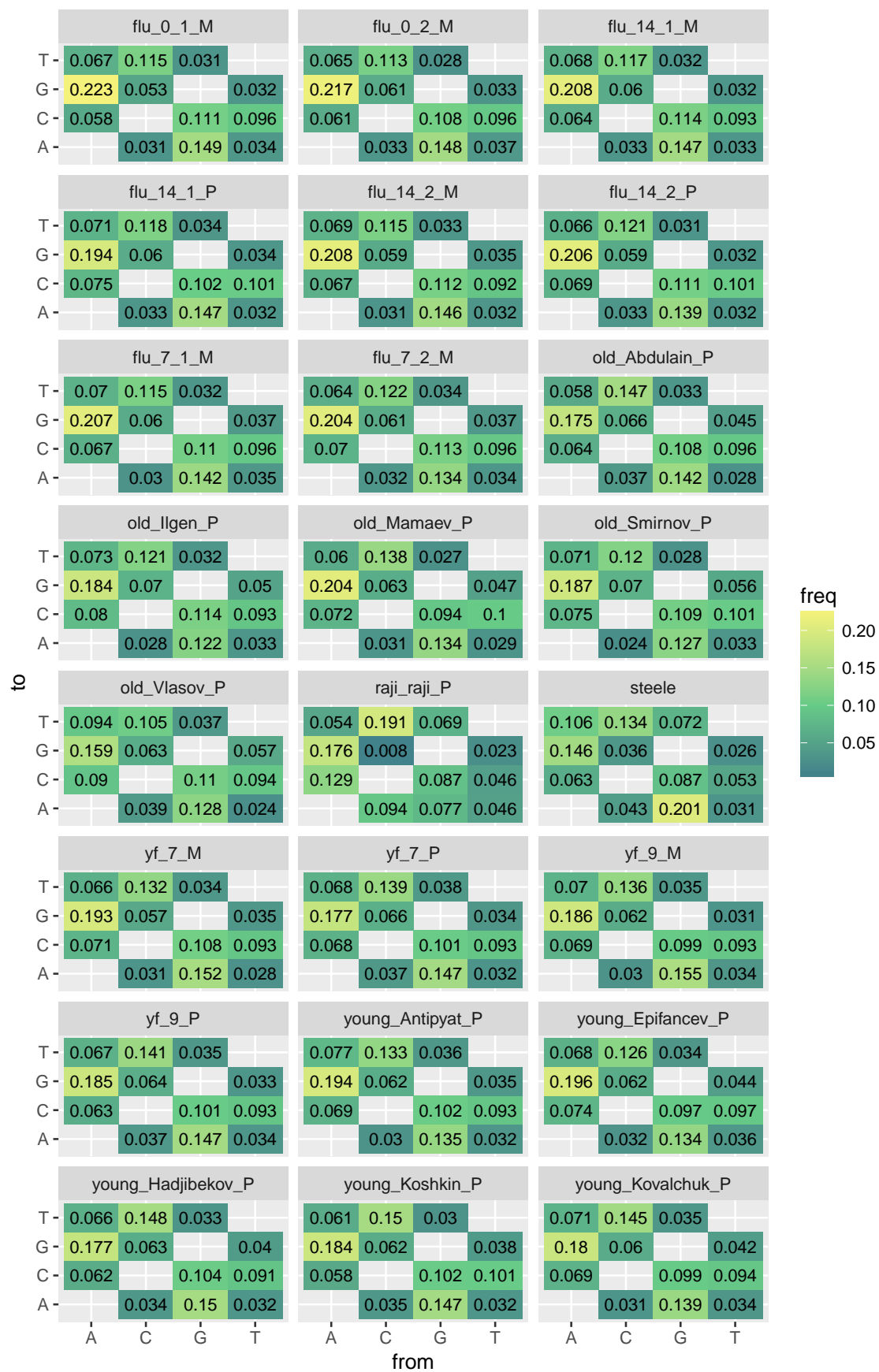
f <- f %>% merge(f1) %>%
  mutate(freq = weight / weight.total) %>%
  dplyr::select(proj, sample, cells, from, to, freq, name)

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

f <- rbind(f, st)

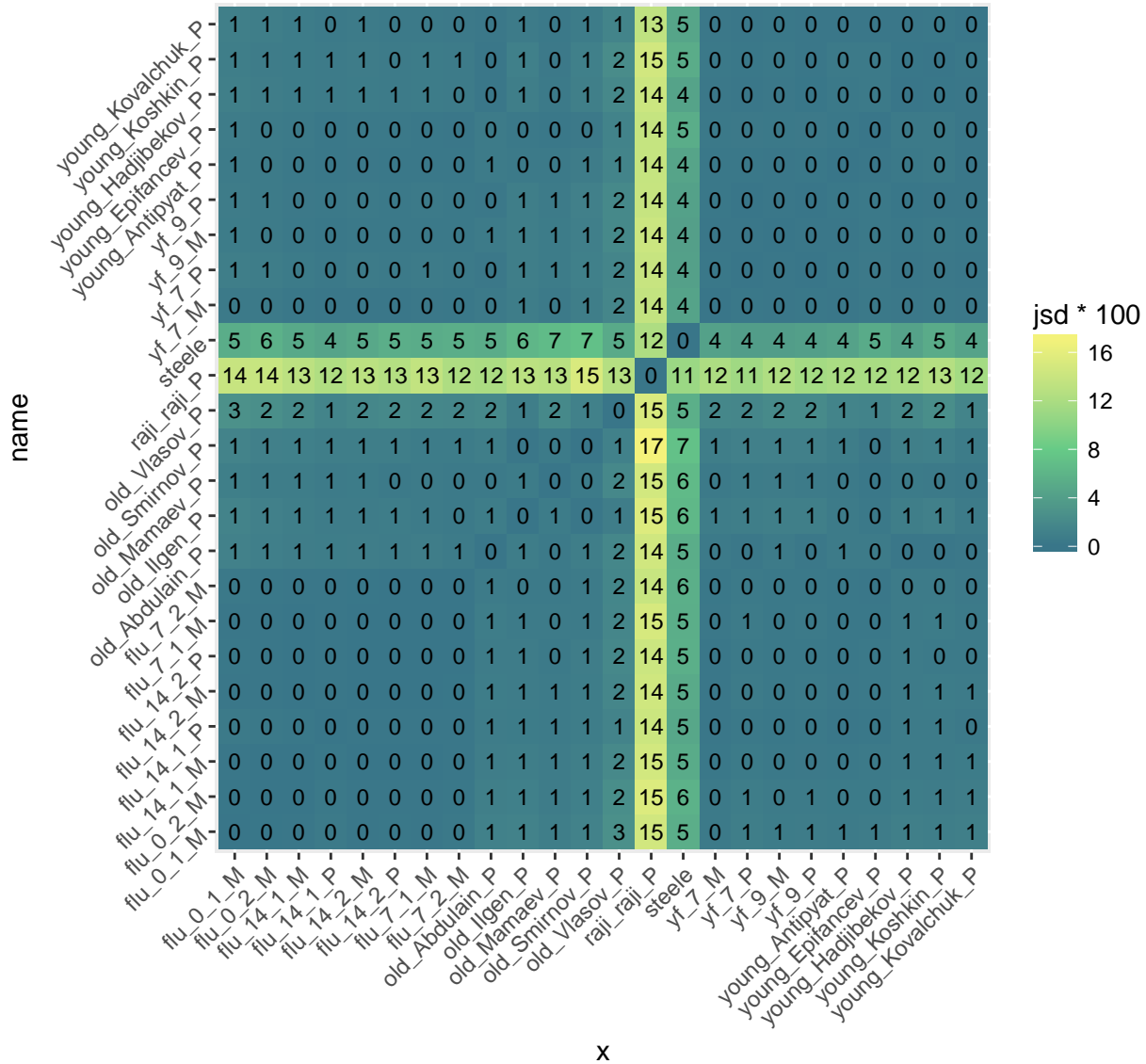
ggplot(f, aes(x = from, y = to)) +
  geom_tile(aes(fill = freq)) +
  geom_text(aes(label = round(freq, 3)), cex=3) +
  facet_wrap(~name, ncol=3) +
  scale_fill_gradient2(low = "#2F6B89", mid = "#67CB87", high = "#F4F27B", midpoint = 0.1, na.value = "

```



Calculate Jensen-Shannon divergence

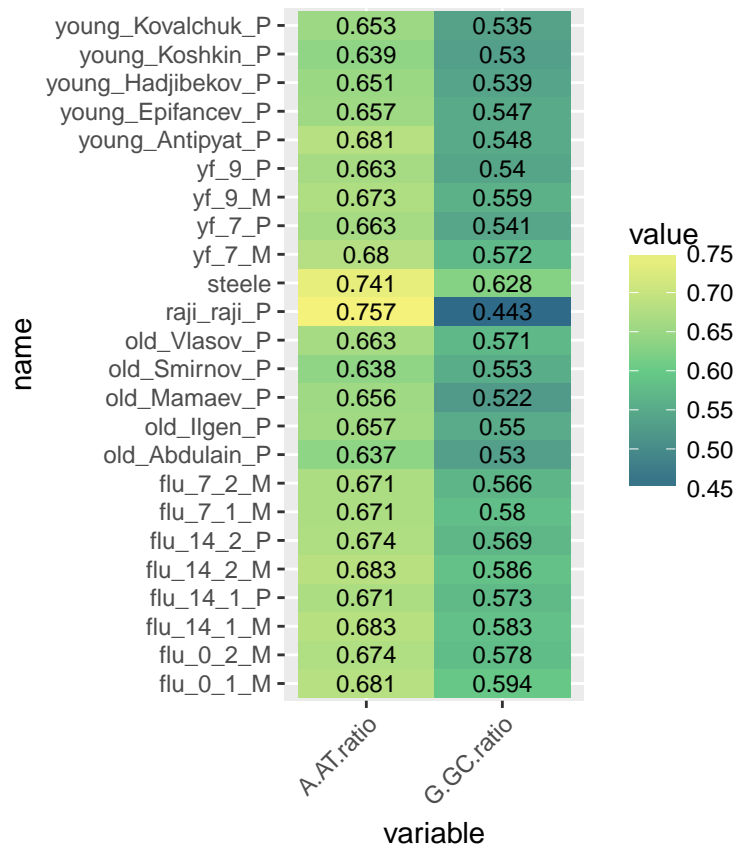
```
jsd <- function(v1, v2){  
  m <- 0.5 * (v1 + v2)  
  0.5 * (sum(v1 * log2(v1 / m)) + sum(v2 * log2(v2 / m)))  
}  
  
f <- f[ order(f$from, f$to), ]  
  
jsd.df <- data.frame()  
  
for (i in unique(f$name)){  
  x <- filter(f, name == i)$freq  
  
  .jsd.df <- f %>% dplyr::group_by(name) %>%  
    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))
```



```
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") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
for (i in unique(shm$name)){
  set <- filter(shm, name == i, nchar(context) == 7)$context
}
```