Untitled

Load SHM table and pre-process it

Change the file name below to load the `*.shm.txt` table generated by MIGMAP/Analyze.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```

```
prefix <- "raji_R12.shm.txt"
df <- read.table(prefix, header=T, sep="\t")
```

(Misc) Order regions and define mutation types.

```
df$region = factor(df$region, c("FR1","CDR1","FR2","CDR2","FR3","CDR3","FR4"))
df$replacement <- ifelse(df$mutation.type != "Substitution", "Indel",
                  ifelse(as.character(df$from.aa) != as.character(df$to.aa), "Replacement", "Silent"))
```

Summarize the by-clonotype table to Variable-Joining segment pair and individual mutation level: - `df` contains mutations for each clonotype, there could several mutations with the same *signature* (segment, position, nucleotides) that are present in different clonotypes - `df.s1` mutations with identical signature are combined within the same V-J pair - `df.s2` mutations with identical signature are combined regardless of parent clonotype/V-J pair

```
library(plyr)

df.s1 <- ddply(df, c("clonotype.v","clonotype.j","segment","segment.name","region",
                     "mutation.type","pos.nt","from.nt","to.nt","pos.aa","from.aa","to.aa",
                     "replacement", "total.clonotypes", "total.count", "total.freq"),
               summarize,
               count.clonotypes = sum(count.clonotypes),
               count.reads = sum(count.reads),
               count.freq = sum(count.freq))

s2.cols <- c("segment","segment.name","region",
                     "mutation.type","pos.nt","from.nt","to.nt","pos.aa","from.aa","to.aa",
                     "replacement", "total.clonotypes", "total.count", "total.freq")

df.s2 <- ddply(df.s1, s2.cols,
               summarize,
               count.clonotypes = sum(count.clonotypes),
               count.reads = sum(count.reads),
               count.freq = sum(count.freq))
```

Define mutations that are in fact allelic variants. Here we require that there are at least `5` clonotypes with a given mutation, the fraction of clonotypes with this mutation among clonotypes with the corresponding V/J allele should be more than `50%` and the total number of reads with this mutation among all reads with corresponding V/J allele should also be more than `50%`.

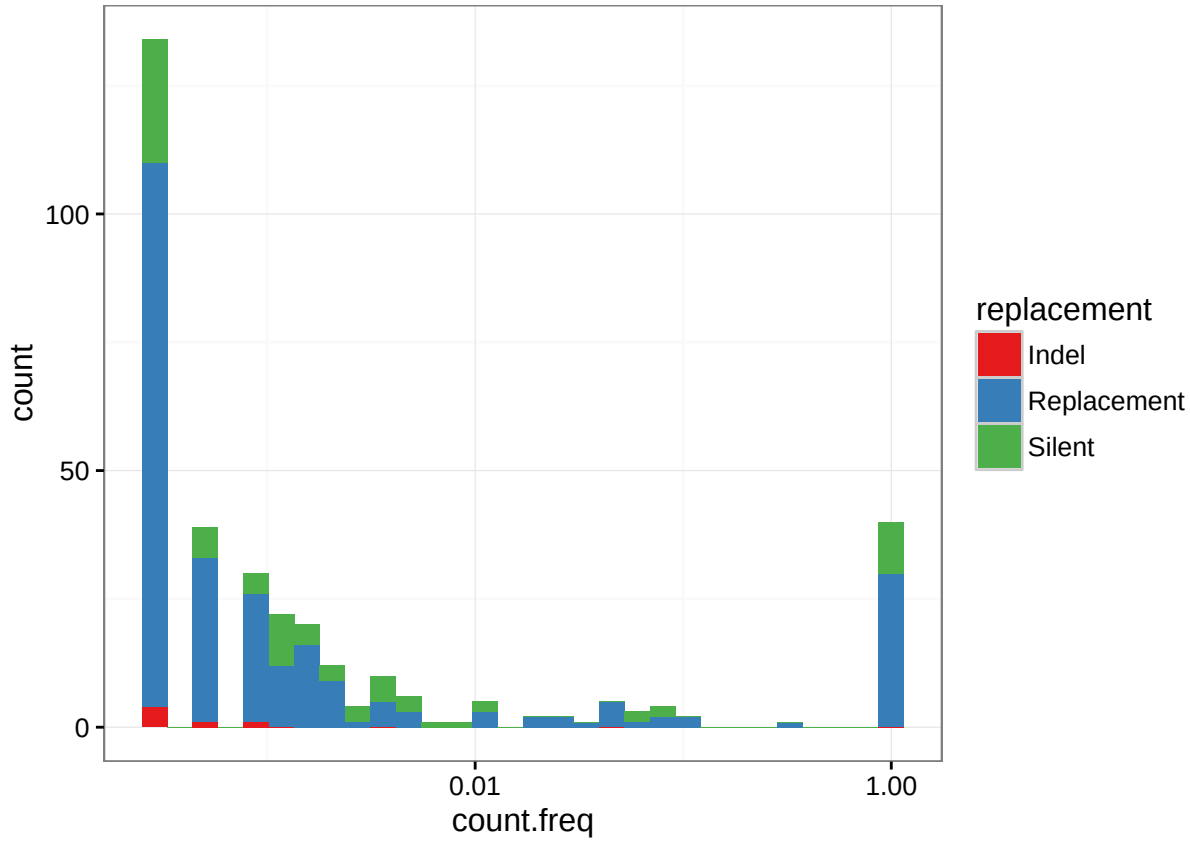
```
df.s2 <- ddply(df.s2, .(segment.name), transform,
               is.allele = count.clonotypes >= 5 &
               count.clonotypes / total.clonotypes >= 0.5 &
               count.freq / total.freq >= 0.5)

df <- merge(df, df.s2, by.x = s2.cols, by.y = s2.cols,
            all.x = T, suffixes = c("", "y"))
df.s1 <- merge(df.s1, df.s2, by.x = s2.cols, by.y = s2.cols,
               all.x = T, suffixes = c("", "y"))
```

Silent and replacement mutation rates

Show the distribution of mutation frequencies (fraction of reads). After this stage we only work with somatic hypermutations and exclude alleles.

```
ggplot(df.s2, aes(x=count.freq)) +
  geom_histogram(aes(fill=replacement)) +
  scale_x_log10() + scale_fill_brewer(palette = "Set1") + theme_bw()
```

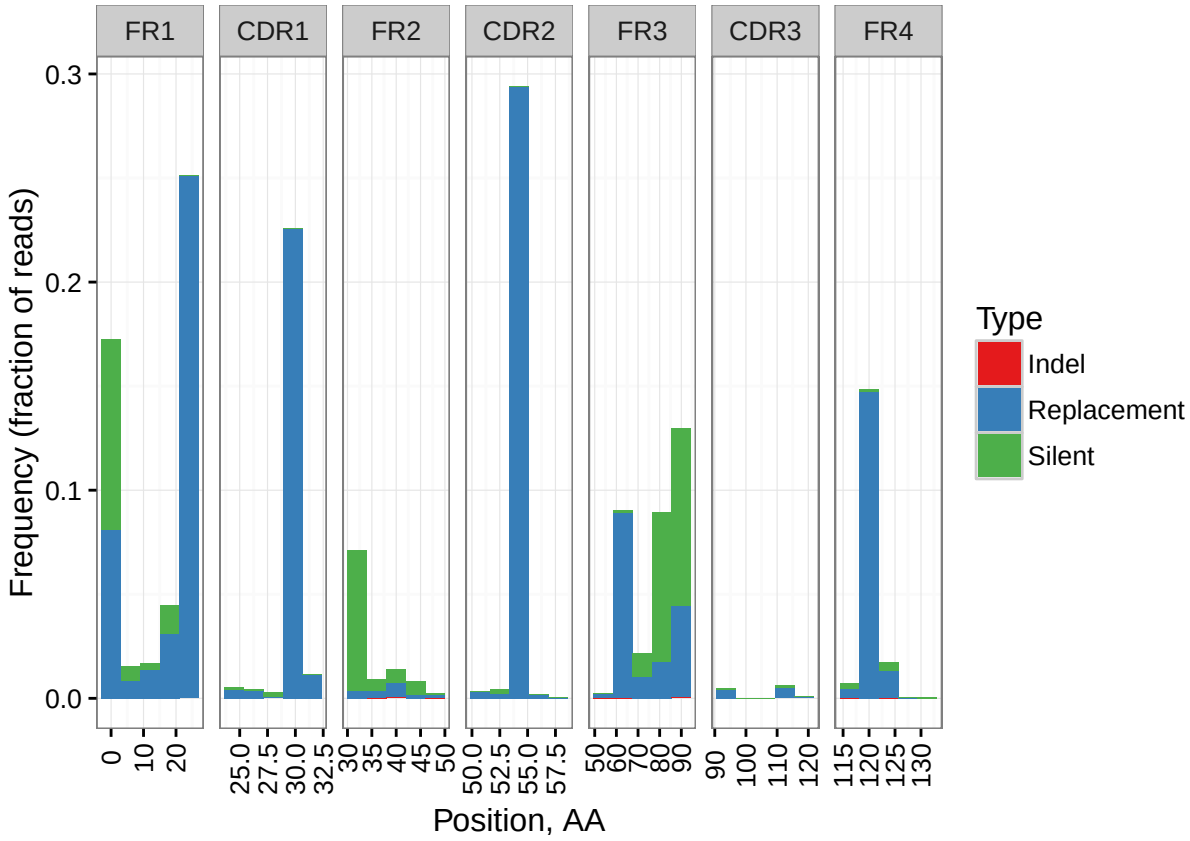
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
df <- subset(df, !is.allele)
df.s1 <- subset(df.s1, !is.allele)
df.s2 <- subset(df.s2, !is.allele)
```

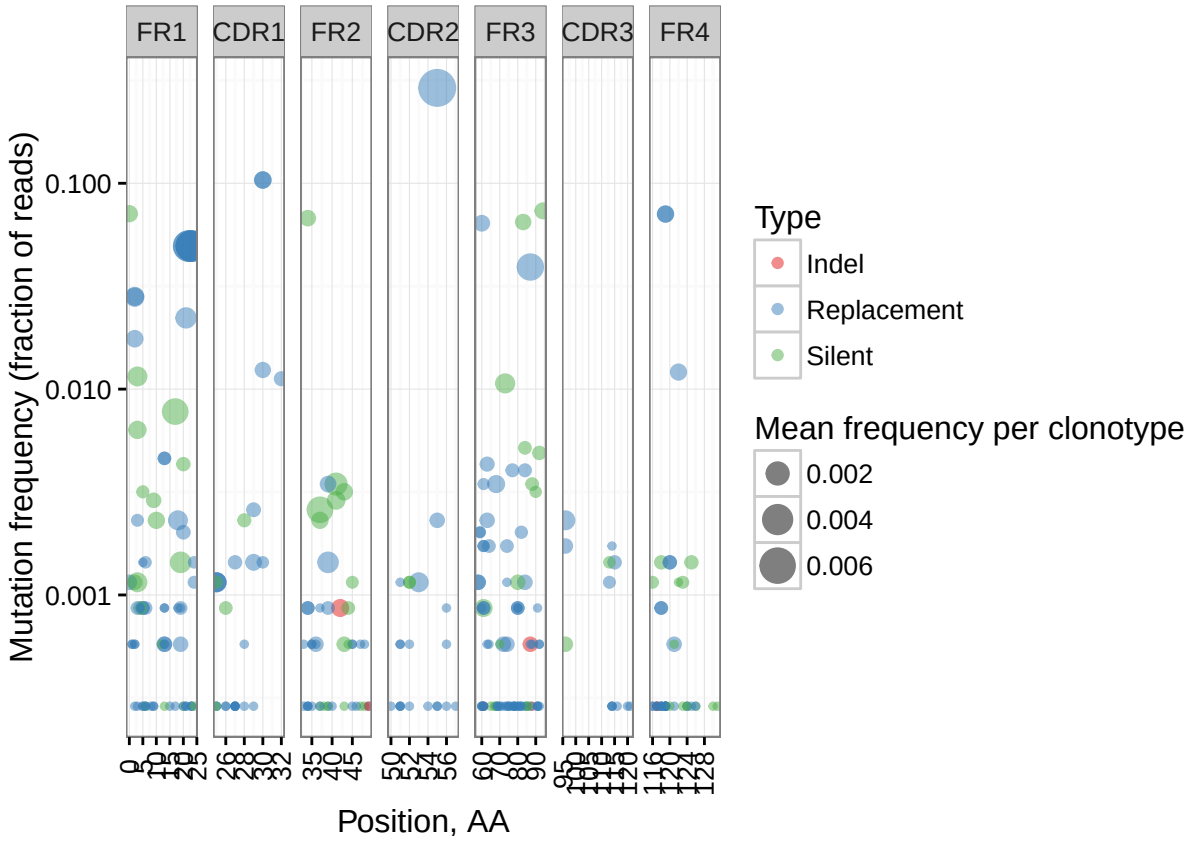
The plot below shows mutation frequency distribution across various IG regions.

```
ggplot(df.s2) +
  geom_histogram(aes(weight=count.freq, fill=replacement, x=pos.aa), bins=5) +
  scale_fill_brewer(name="Type", palette = "Set1") +
  facet_grid(~region, scales="free_x") +
  ylab("Frequency (fraction of reads)") +
  xlab("Position, AA") +
  theme_bw() + theme(axis.text.x = element_text(angle = 90, vjust=0.5))
```



The plot below shows distribution of individual mutations across IG regions: y axis shows the frequency (fraction of reads) for a given mutation, while the point size shows mean mutation frequency per clonotype (fraction of reads divided by total number of clonotypes).

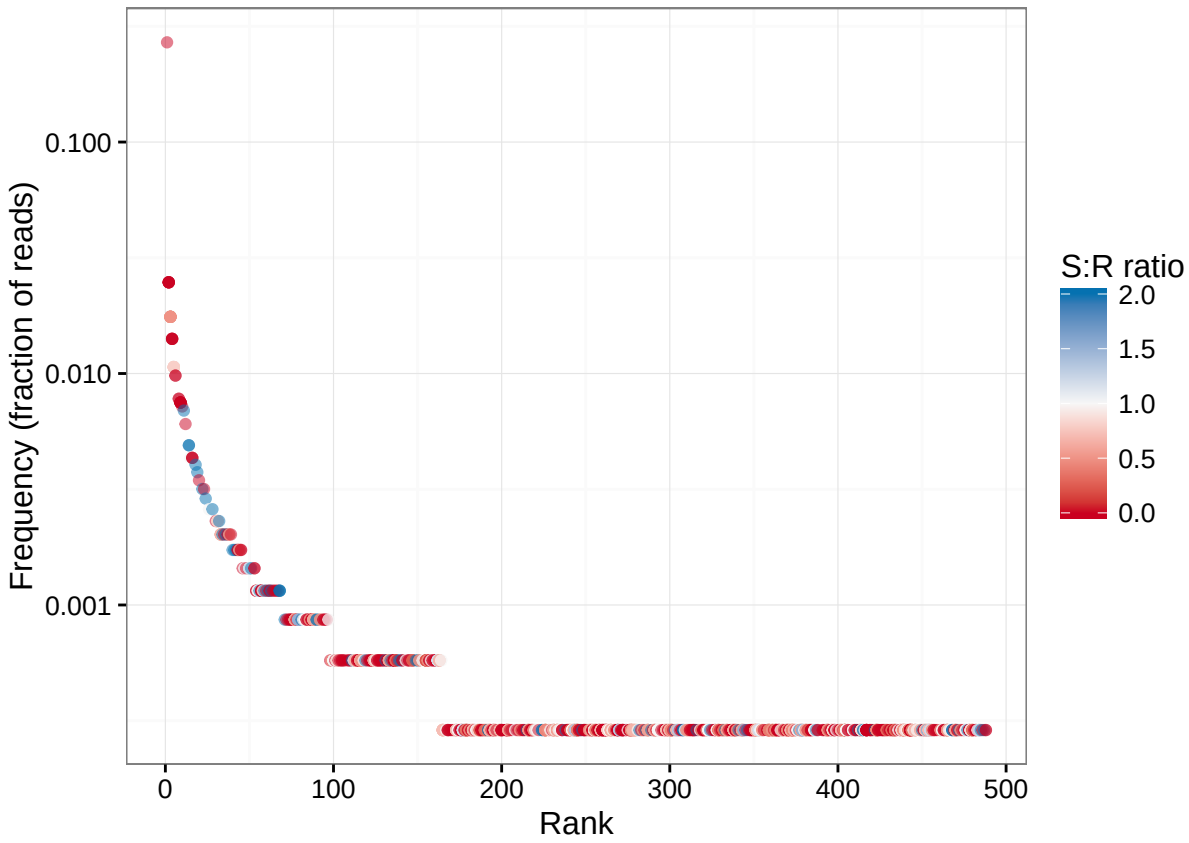
```
ggplot(df.s2) +
  #geom_boxplot(aes(x=region, group=interaction(region, replacement), y = count.freq, fill = replacemen
  geom_point(aes(y=count.freq, color=replacement, x=pos.aa, size=count.freq/count.clonotypes), alpha=0.5
  scale_color_brewer(name = "Type", palette = "Set1") +
  facet_grid(~region, scales="free_x") +
  scale_y_log10(name = "Mutation frequency (fraction of reads)") +
  scale_size(name = "Mean frequency per clonotype") +
  xlab("Position, AA") +
  theme_bw() + theme(axis.text.x = element_text(angle = 90, vjust=0.5))
```



Clonotype-level statistics

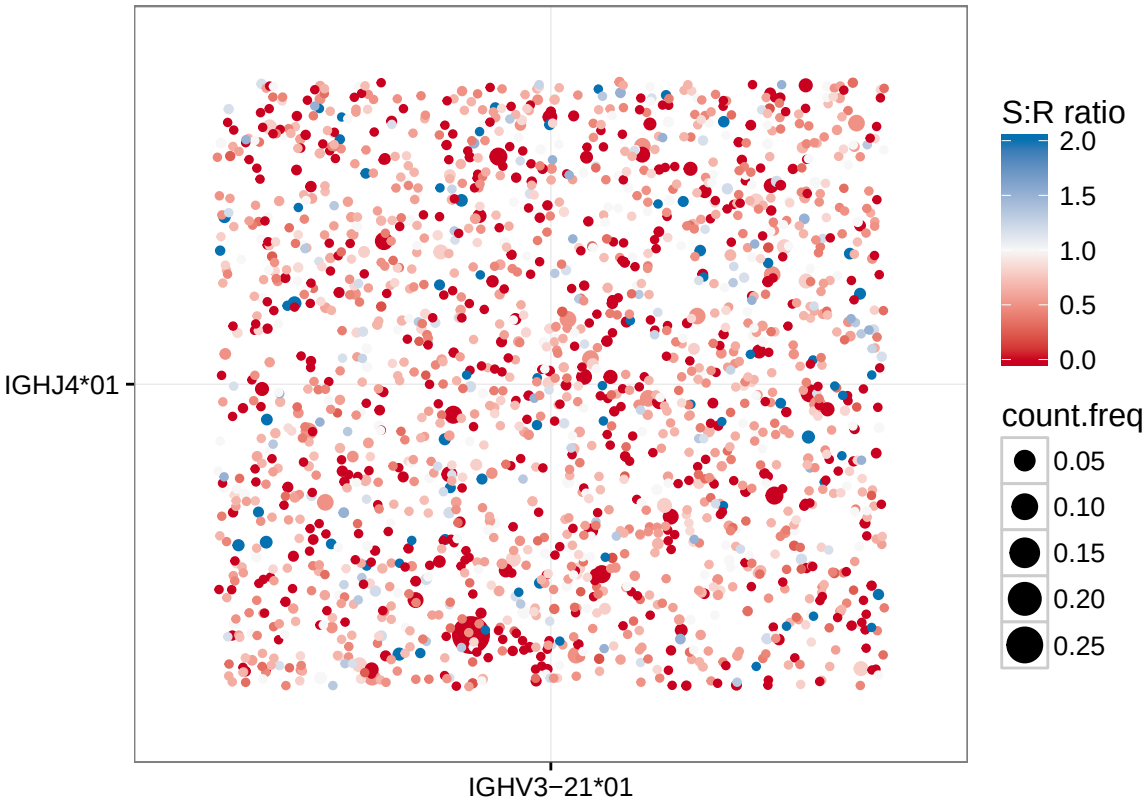
The following plot shows clonotype frequency (frequency-rank) distribution, individual clonotypes are represented by points colored by the silent:replacement (S:R) ratio of their hypermutations.

```
df <- ddply(df, .(clonotype.id), transform,
              sr = sum(count.freq[which(replacement == "Silent")]) / sum(count.freq[which(replacement ==

ggplot(df, aes(x=clonotype.id, y=count.freq, color=sr)) +
  geom_point(alpha=0.5) + scale_y_log10(name = "Frequency (fraction of reads)") + xlab("Rank") +
  scale_color_gradient2(name = "S:R ratio",
                        low = "#ca0020", mid="#f7f7f7", high = "#0571b0", midpoint = 1, na.value = "#057
  theme_bw()
```



The plot below shows scatter plots of clonotypes colored by their S:R ratio and grouped by V-J pair (in this example only one V-J pair is present), a-la Vidjil.

```
df.s1 <- ddply(df.s1, .(clonotype.v, clonotype.j), transform,
               sr = sum(count.freq[which(replacement == "Silent")]) / sum(count.freq[which(replacement =

ggplot(df, aes(x=clonotype.v, y=clonotype.j, color=sr, size=count.freq)) +
  geom_jitter() + xlab("") + ylab("") +
  scale_color_gradient2(name = "S:R ratio",
                        low = "#ca0020", mid="#f7f7f7", high = "#0571b0", midpoint = 1, na.value = "#05
  theme_bw()
```


IGHJ4*01
IGHV3−21*01
S:R ratio
2.0
1.5
1.0
0.5
0.0
count.freq
0.05
0.10
0.15
0.20
0.25