An example of data analysis

Libraries

```
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(RColorBrewer)
library(scales)
```

Auxiliary functions

```
get_segment_id = function(s) {
  unlist(lapply(strsplit(s, "*", fixed=T), function(x) x[1]))
}

read_mixcr = function(file_name) {
  .df = read.table(file_name, header=T, stringsAsFactors = F, sep="\t", fill = T) %>%
    select(Clone.ID, Clone.count, All.V.hits, All.D.hits, All.J.hits, N..Seq..CDR3, AA..Seq..CDR3)
  colnames(.df) = c("clone.id", "count", "v", "d", "j", "cdr3nt", "cdr3aa")
  .df$freq = .df$count / sum(.df$count)

  .df %>% mutate(v=get_segment_id(v),d=get_segment_id(d),j=get_segment_id(j))
}


head(read_mixcr("chudakovlab/A1_8_alpha.txt.gz"))
```

```
##   clone.id count           v    d      j
## 1        0    27   TRAV23DV6 <NA> TRAJ49
## 2        1    21      TRAV20 <NA> TRAJ27
## 3        2     5    TRAV12-3 <NA> TRAJ31
## 4        3     5      TRAV21 <NA>  TRAJ7
## 5        4     4    TRAV13-2 <NA> TRAJ13
## 6        5     4 TRAV38-2DV8 <NA> TRAJ49
##                                       cdr3nt         cdr3aa        freq
## 1 TGTGCAGCAAGCAACGGGGACACCGGTAACCAGTTCTATTTT CAASNGDTGNQFYF 0.015615963
## 2    TGTGCTGTGCTGGGCACCAATGCAGGCAAATCAACCTTT  CAVLGTNAGKSTF 0.012145749
## 3    TGTGCAATGAGCCTCAATAACAATGCCAGACTCATGTTT  CAMSLNNNARLMF 0.002891845
## 4       TGTGCTGTGAGAAAGGGGAACAACAGACTCGCTTTT   CAVRKGNNRLAF 0.002891845
## 5 TGTGCAGAGTTCCGCGCTGGGGGTTACCAGAAAGTTACCTTT CAEFRAGGYQKVTF 0.002313476
## 6     TGTGCTTATAGGAGACCTACCGGTAACCAGTTCTATTT  CAYRRPTGNQFYF 0.002313476
```

Read samples

```r
df = data.frame()
for (chain in c("alpha", "beta")) {
for (replica in c("A1", "A2", "A3")) {
  file_name = paste0("chudakovlab/", replica, "_", 8, "_", chain, ".txt.gz")
  .df = read_mixcr(file_name)
  .df$replica = replica
  .df$chain = chain
  .df$amount = "8ng"
  df = rbind (df, .df)
}
for (replica in c("B", "C", "D")) {
  file_name = paste0("chudakovlab/", replica, "_", 65, "_", chain, ".txt.gz")
  .df = read_mixcr(file_name)
  .df$replica = replica
  .df$chain = chain
  .df$amount = "65ng"
  df = rbind (df, .df)
}
}

summary(df)
```

```
##     clone.id         count              v                  d
##  Min.   :    0   Min.   :  1.000   Length:154156      Length:154156
##  1st Qu.: 3757   1st Qu.:  1.000   Class :character   Class :character
##  Median :10169   Median :  1.000   Mode  :character   Mode  :character
##  Mean   :12432   Mean   :  1.291
##  3rd Qu.:20204   3rd Qu.:  1.000
##  Max.   :33614   Max.   :679.000
##       j                cdr3nt             cdr3aa
##  Length:154156      Length:154156      Length:154156
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##       freq             replica             chain
##  Min.   :2.221e-05   Length:154156      Length:154156
##  1st Qu.:2.231e-05   Class :character   Class :character
##  Median :2.327e-05   Mode  :character   Mode  :character
##  Mean   :7.784e-05
##  3rd Qu.:6.118e-05
##  Max.   :2.174e-02
##     amount
##  Length:154156
##  Class :character
##  Mode  :character
```

```
##
##
##
```

Basic repertoire properties

Segment usage

Summarize data

```
df.segm = df %>%
  group_by(replica, chain, amount, v, j) %>%
  summarize(freq = sum(freq), uniq = n()) %>%
  group_by(replica) %>%
  mutate(freq.rank = rank(-freq))

df.segm.v = df.segm %>%
  group_by(replica, chain, amount, v) %>%
  summarize(freq = sum(freq), uniq = sum(uniq)) %>%
  group_by(v) %>%
  mutate(freq.tot = mean(freq))

df.segm.v$v <- factor(df.segm.v$v, levels=df.segm.v$v[order(df.segm.v$freq.tot)])
```

```
## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated
```

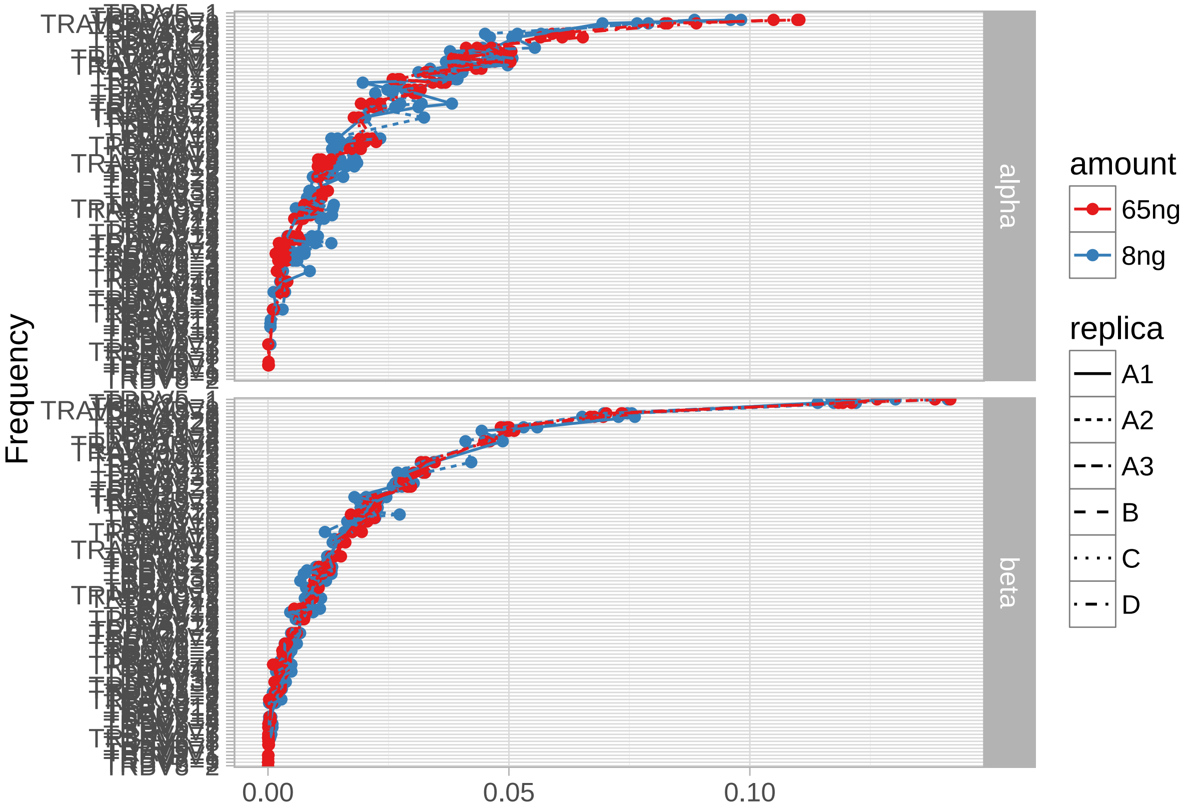
```
df.segm.j = df.segm %>%
  group_by(replica, chain, amount, j) %>%
  summarize(freq = sum(freq), uniq = sum(uniq))  %>%
  group_by(j) %>%
  mutate(freq.tot = mean(freq))

df.segm.j$j <- factor(df.segm.j$j, levels=df.segm.j$j[order(df.segm.j$freq.tot)])
```

```
## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated
```

Variable segment usage

```
ggplot(df.segm.v, aes(x=v, y=freq, color=amount, linetype=replica)) +
  geom_point() +
  geom_line(aes(group=replica)) +
  coord_flip() +
  facet_grid(chain~., space = "free") +
  xlab("Frequency") + ylab("") +
  scale_color_brewer(palette = "Set1") +
  theme_light()
```

```
## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated
## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated

## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated
```



```r
a <- aov(freq ~ v + amount : v, df.segm.v)
```

```
## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated

## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated
```

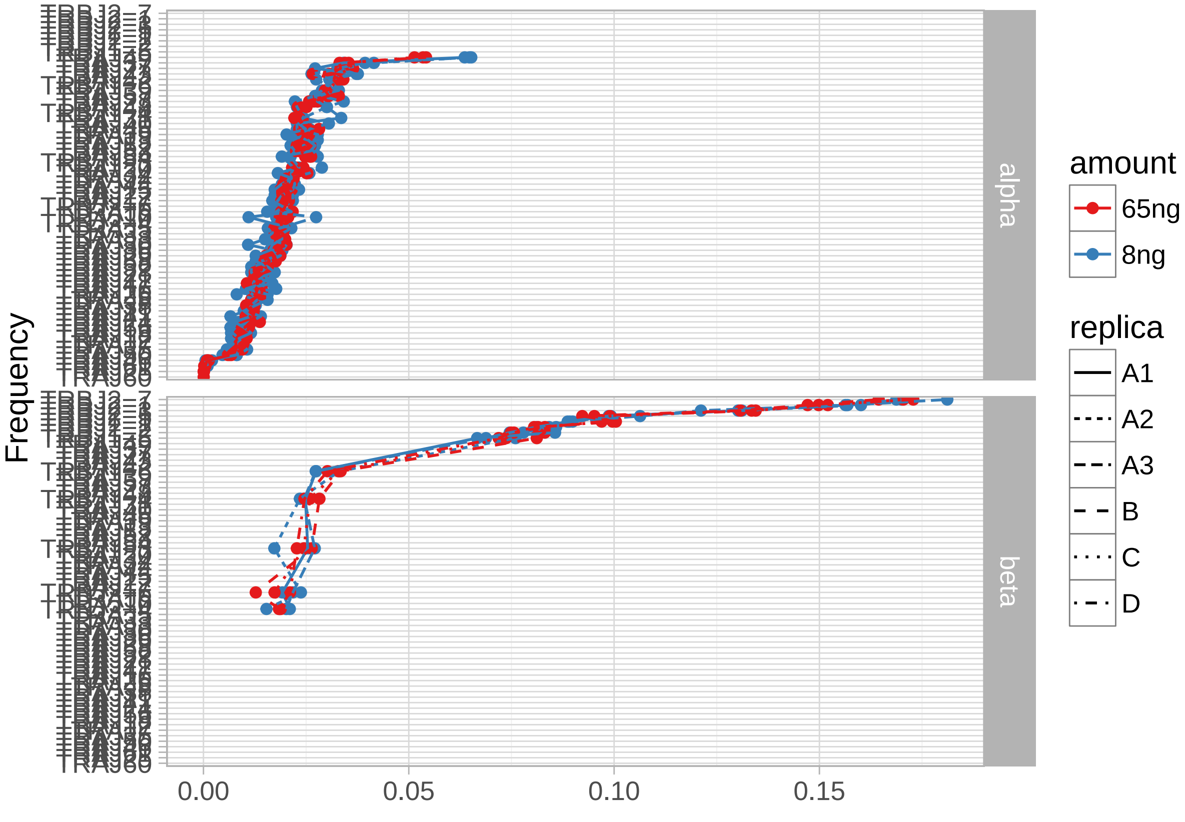
```r
summary(a)
```

```
##               Df Sum Sq  Mean Sq F value Pr(>F)    
## v            105 0.3583 0.003413 569.139 <2e-16 ***
## v:amount      95 0.0025 0.000026   4.386 <2e-16 ***
## Residuals    382 0.0023 0.000006                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Joining segment usage

```
ggplot(df.segm.j, aes(x=j, y=freq, color=amount, linetype=replica)) +
  geom_point() +
  geom_line(aes(group=replica)) +
  coord_flip() +
  xlab("Frequency") + ylab("") +
  facet_grid(chain~., space = "free") +
  scale_color_brewer(palette = "Set1") +
  theme_light()
```

```
## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated

## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated

## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated
```



```
a <- aov(freq ~ j + amount : j, df.segm.j)
```

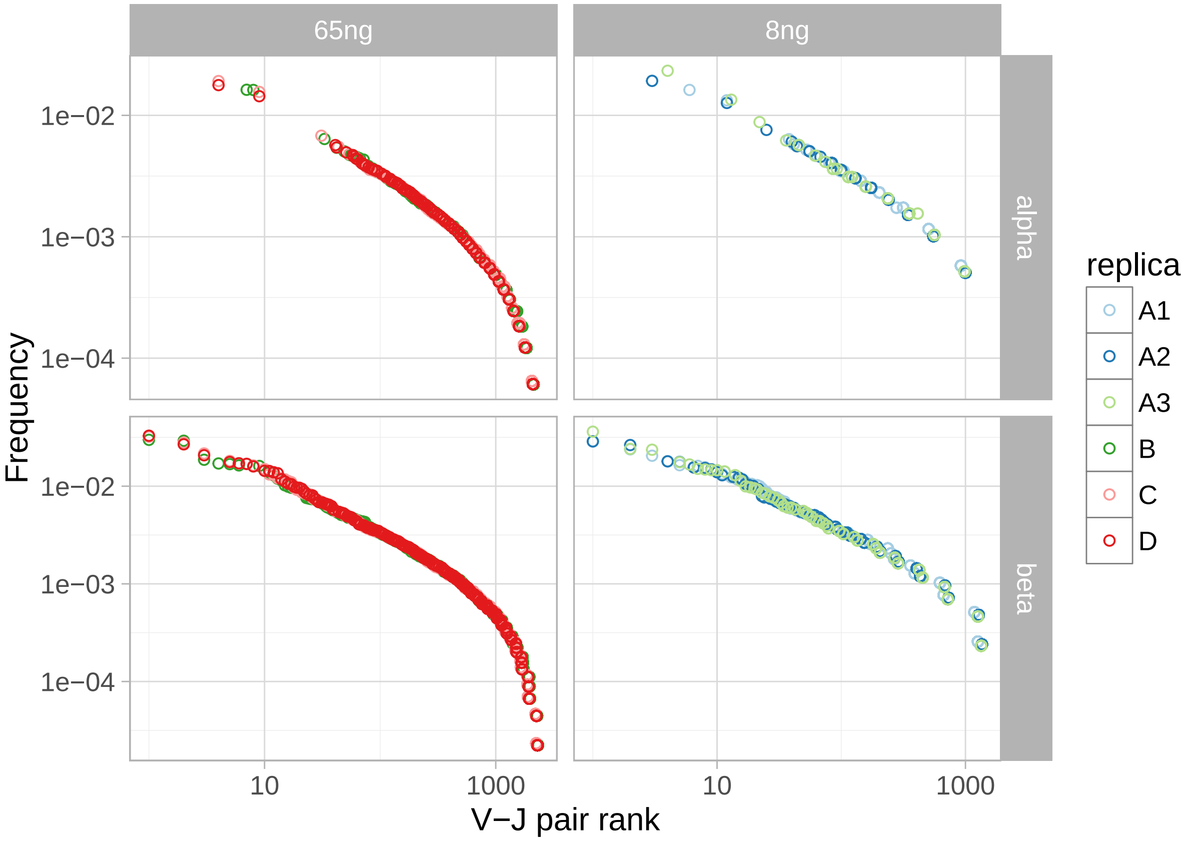
```
## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated

## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated
summary(a)
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)    
## j              66 0.4356 6.6e-03 925.840  < 2e-16 ***
## j:amount       65 0.0010 1.5e-05   2.087 2.61e-05 ***
## Residuals     260 0.0019 7.0e-06                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that there is some difference is segment usage related to the starting amount of RNA

V-J segment usage

```
ggplot(df.segm, aes(x=freq.rank, y=freq, color=replica)) +
  geom_point(shape=21) +
  scale_x_log10("V-J pair rank") + scale_y_log10("Frequency") +
  facet_grid(chain~amount, scales="free") +
  scale_color_brewer(palette = "Paired") +
  theme_light()
```



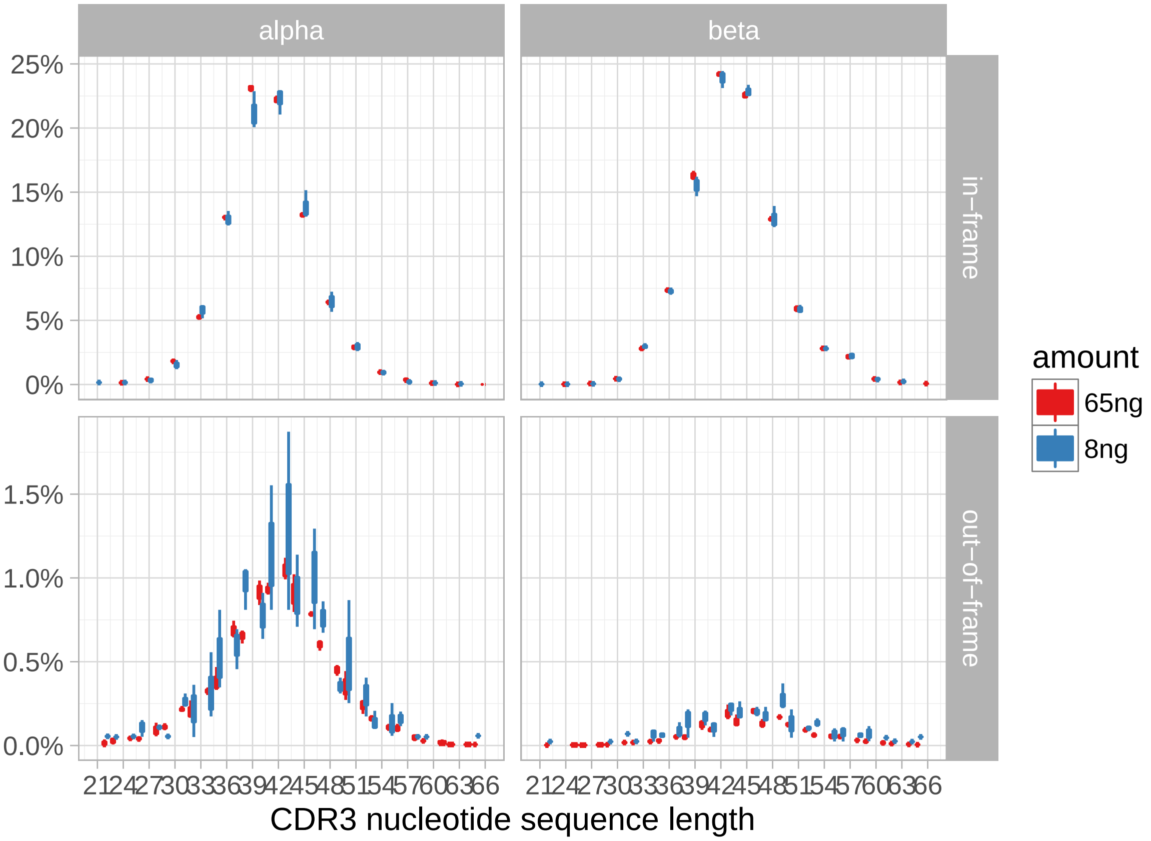
Spectratype

```
df.sp = df %>%
  mutate(cdr3.len = nchar(cdr3nt)) %>%
  group_by(replica, chain, amount, cdr3.len) %>%
  summarize(freq = sum(freq), uniq=n())
```

CDR3 length distribution and out-of-frame sequences

```
df.sp$in.frame = ifelse(df.sp$cdr3.len %% 3 == 0, "in-frame", "out-of-frame")
ggplot(df.sp, aes(x=cdr3.len, y=freq, fill=amount, color=amount)) +
  geom_boxplot(aes(group=interaction(cdr3.len,amount))) +
  scale_x_continuous("CDR3 nucleotide sequence length", limits=c(21,66), breaks = seq(21,66,by=3)) +
  scale_y_continuous("", labels=percent) +
  facet_grid(in.frame~chain, scales="free") +
  scale_fill_brewer(palette = "Set1") +
  scale_color_brewer(palette = "Set1") +
  theme_light()
```

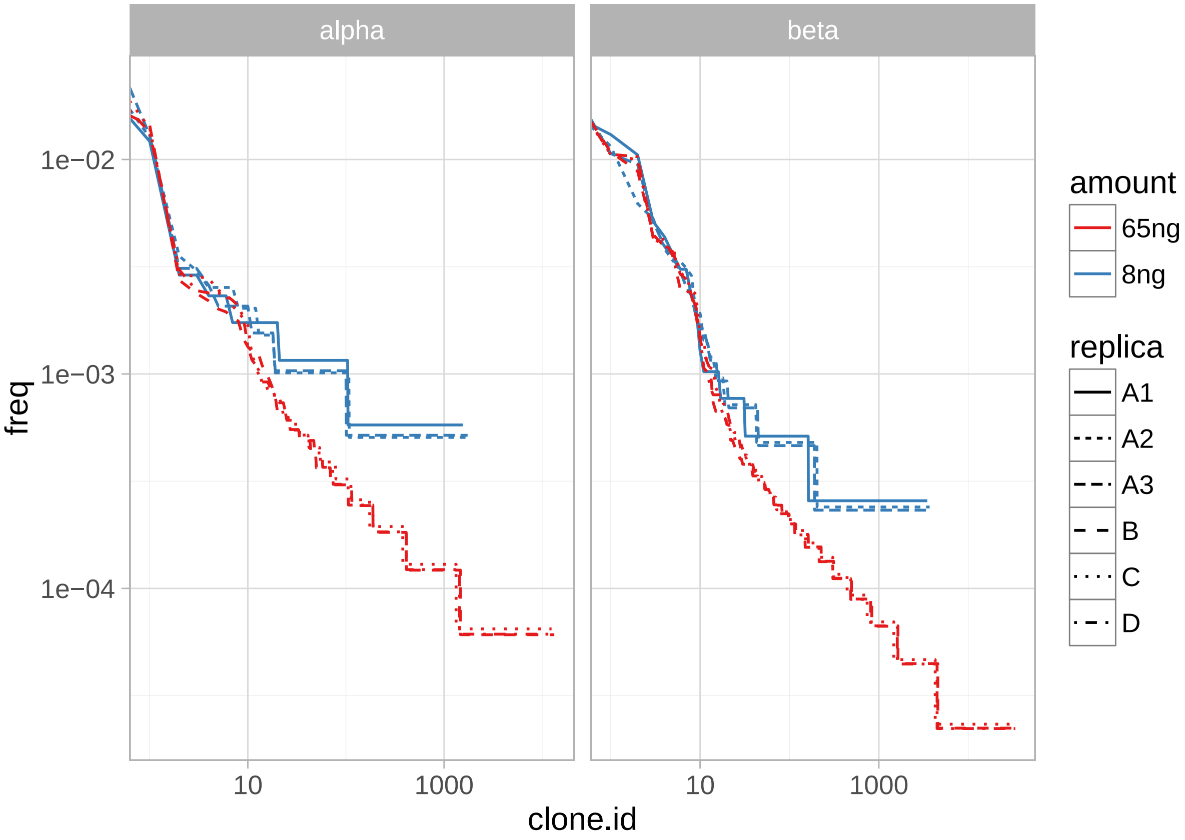
```
## Warning: Removed 180 rows containing non-finite values (stat_boxplot).

## Warning: Removed 3 rows containing missing values (geom_boxplot).

## Warning: Removed 1 rows containing missing values (geom_segment).
```



Clonotype abundance quantification

Clonotype frequency and rank

```
ggplot(df, aes(x=clone.id, y=freq, linetype=replica, group=replica, color=amount)) +
  geom_line() +
  scale_y_log10() + scale_x_log10() +
  facet_grid(~chain, scales="free") +
  scale_color_brewer(palette = "Set1") +
  theme_light()
```



Do some preprocessing

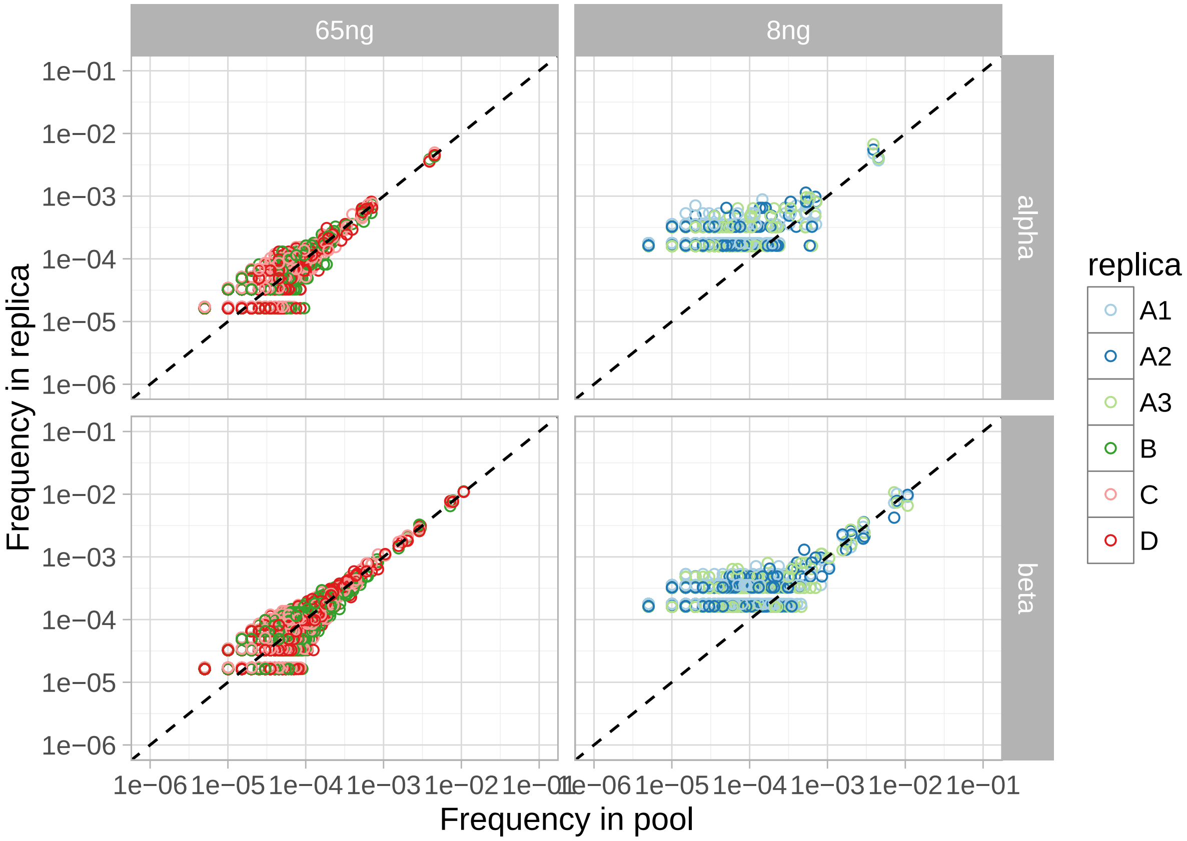
```
df.1 = df %>%
  select(cdr3nt, replica, amount, count, chain)

# fill missing clonotypes with 0
dummy = expand.grid(cdr3nt = unique(df.1$cdr3nt),
                    replica = unique(df.1$replica),
                    chain = unique(df.1$chain),
                    amount = unique(df.1$amount))

real_ccdr = interaction(df.1$chain, df.1$cdr3nt)
real_replamount = interaction(df.1$replica, df.1$amount)
dummy = subset(dummy, interaction(chain, cdr3nt) %in% real_ccdr)
dummy = subset(dummy, interaction(replica, amount) %in% real_replamount)
dummy$count = 0
```

```
df.1 = rbind(df.1, dummy) %>%
  group_by(cdr3nt, chain, replica, amount) %>%
  summarize(count = sum(count))

df.1$count.grand = sum(df.1$count)

df.1 = df.1 %>%
  group_by(cdr3nt) %>%
  mutate(count.total = sum(count)) %>%
  group_by(replica) %>%
  mutate(count.replica.total = sum(count))
```

Log frequency variance

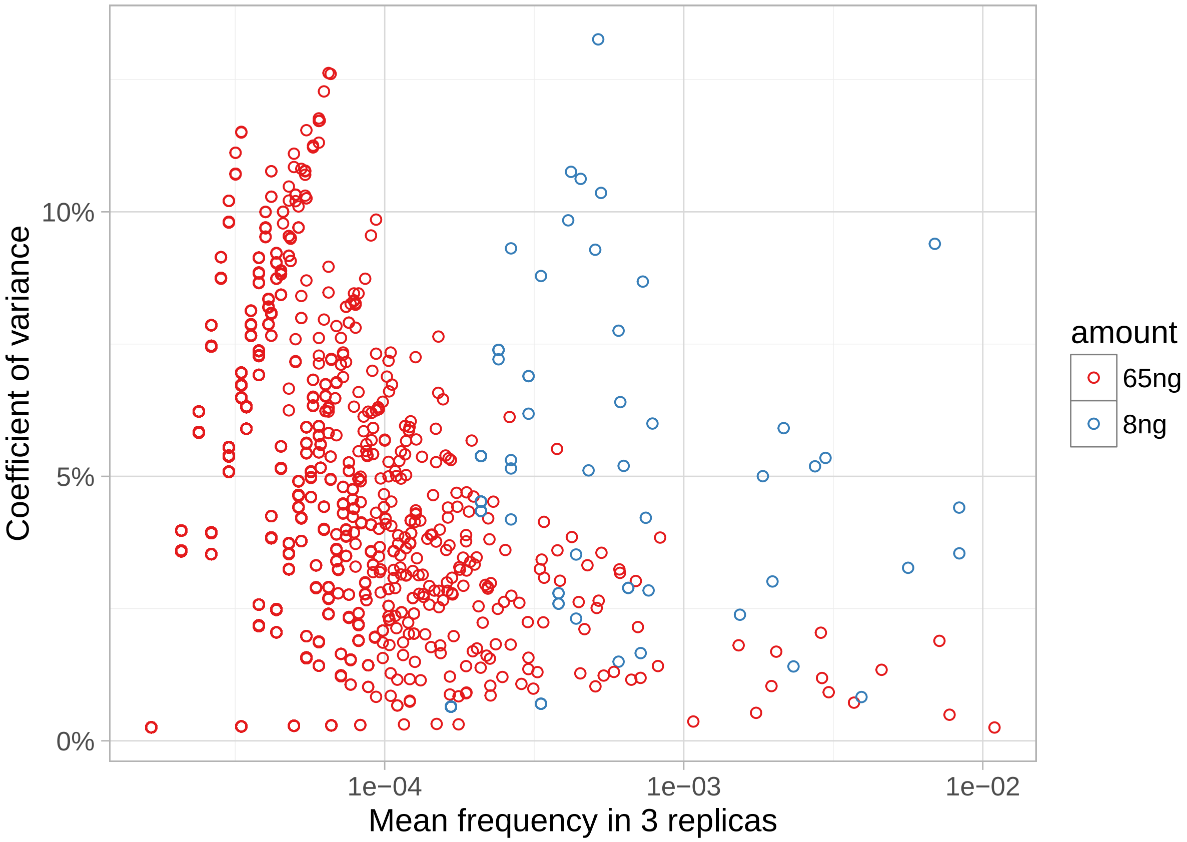
Clonotype frequency in different replicas versus clonotype frequency estimated from pooled sample

```
ggplot(subset(df.1,count>0), aes(x=count.total/count.grand,
                                  y=count/count.replica.total)) +
  geom_point(aes(color=replica), shape=21) +
  geom_abline(slope = 1, intercept = 0, color="black", linetype="dashed") +
  facet_grid(chain~amount, space="free", scales="free") +
  scale_x_log10("Frequency in pool", limits=c(1e-6,1e-1), breaks=10^(-6:-1)) +
  scale_y_log10("Frequency in replica", limits=c(1e-6,1e-1), breaks=10^(-6:-1)) +
  scale_color_brewer(palette = "Paired") +
  theme_light()
```



Coefficient of variance for log-transformed frequencies

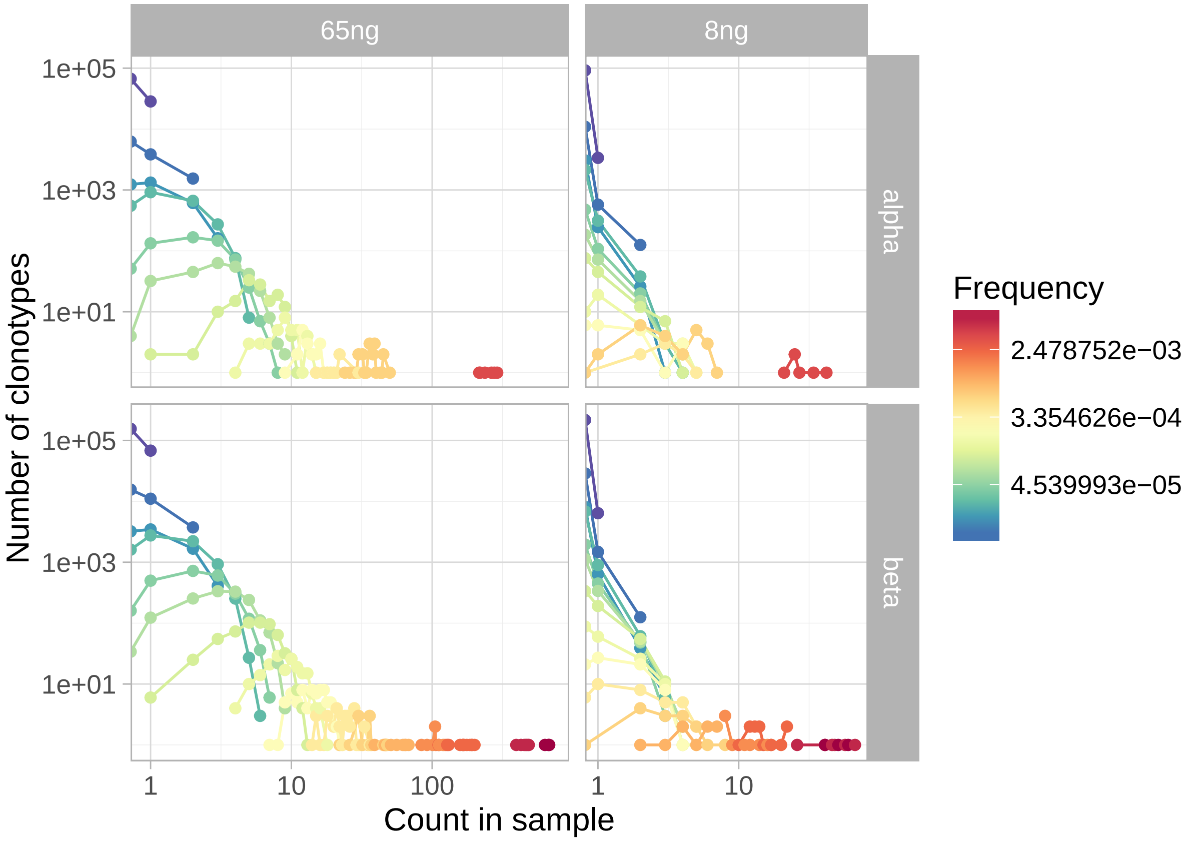
```
df.5 = df.1 %>%
  group_by(cdr3nt, amount) %>%
  mutate(min.count = min(count)) %>%
  filter(min.count > 0)

df.5 = df.5 %>%
  group_by(cdr3nt, amount, replica) %>%
  mutate(freq = log10(count / count.replica.total)) %>%
  group_by(cdr3nt, amount) %>%
  summarize(freq.mean = mean(freq), freq.sd = sd(freq))

ggplot(df.5, aes(x=10^freq.mean, y=freq.sd/abs(freq.mean), color = amount)) +
  geom_point(shape=21) +
  scale_x_log10("Mean frequency in 3 replicas") +
  scale_y_continuous("Coefficient of variance", labels=percent) +
  scale_color_brewer(palette = "Set1") +
  theme_light()
```



Clonotype cDNA count

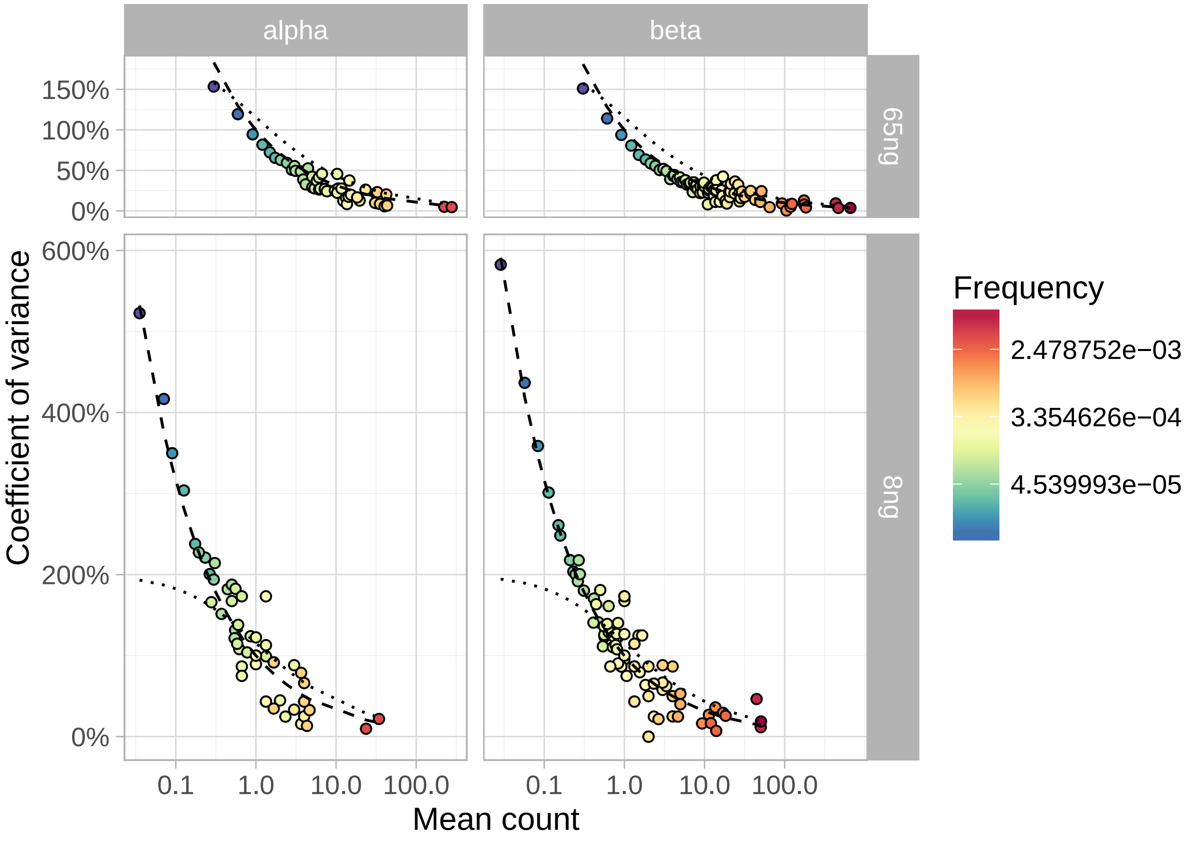
Distribution of clonotype count in 8 and 65 ng samples for clonotypes from different frequency tiers.

```
df.2 = df.1 %>%
  mutate(log.freq = round(5*log10(count.total/count.grand))/5) %>%
  group_by(count, amount, chain, log.freq) %>%
  summarize(nn = n()) %>%
  group_by(amount, log.freq) %>%
  mutate(P = nn / sum(nn))

rf <- colorRampPalette(rev(brewer.pal(11, 'Spectral')))
r <- rf(40)

ggplot(subset(df.2), aes(x = count, y = nn, color=10^log.freq,
                         group=factor(log.freq))) +
  geom_line() +
  geom_point() +
  facet_grid(chain~amount, scales="free", space = "free") +
  scale_y_log10("Number of clonotypes") + scale_x_log10("Count in sample") +
  scale_color_gradientn("Frequency", colors=r, trans="log") +
  theme_light()
```



Coefficient of variance versus clonotype abundance. Dashed and dotted lines show CV of Poisson and Beta Binomial distribution with Jeffreys ($B(1/2, 1/2)$) prior respectively. Note that while rare clonotypes are perfectly fitted with a simple Poisson model, high-abundance clonotypes have relatively high coefficient of variance, esp for 8ng sample (plausible explanations: TCR expression, cell clumping).

```
df.3 = df.1 %>%
  group_by(amount, chain, count.total, count.grand) %>%
  summarize(count.mean = mean(count),
            count.sd = sd(count),
            mean.replica.size = mean(count.replica.total)) %>%
  group_by(count.mean, mean.replica.size) %>%
  mutate(a = count.mean + 1/2,
         b = mean.replica.size - count.mean + 1/2,
         bb.mean = mean.replica.size * a / (a + b),
         bb.sd = sqrt(mean.replica.size * a * b * (a + b + mean.replica.size) / (a + b) / (a + b) / (a +

ggplot(df.3) +
  geom_point(aes(x=count.mean, y = count.sd / count.mean,
                 fill=10^(round(5*log10(count.total/count.grand))/5)), color="black", shape=21) +
  geom_line(aes(x=count.mean, y = 1 / sqrt(count.mean)), color="black", linetype="dashed") +
  geom_line(aes(x=count.mean, y = bb.sd / bb.mean), color="black", linetype="dotted") +
  facet_grid(amount~chain, scales="free", space = "free") +
  scale_y_continuous("Coefficient of variance", labels = percent) +
  scale_x_log10("Mean count", breaks = c(0.1, 1, 10, 100)) +
  scale_fill_gradientn("Frequency", colors=r, trans="log") +
  theme_light()
```



High-abundance clonotypes

Get some high-abundance clonotypes

```
top10_clones = df.1 %>%
  group_by(cdr3nt) %>%
  summarize(x = mean(count.total)) %>%
  mutate(rank = rank(-x)) %>%
  arrange(rank) %>%
  filter(rank <= 10)

df.4 = subset(df.1, cdr3nt %in% top10_clones$cdr3nt)
```

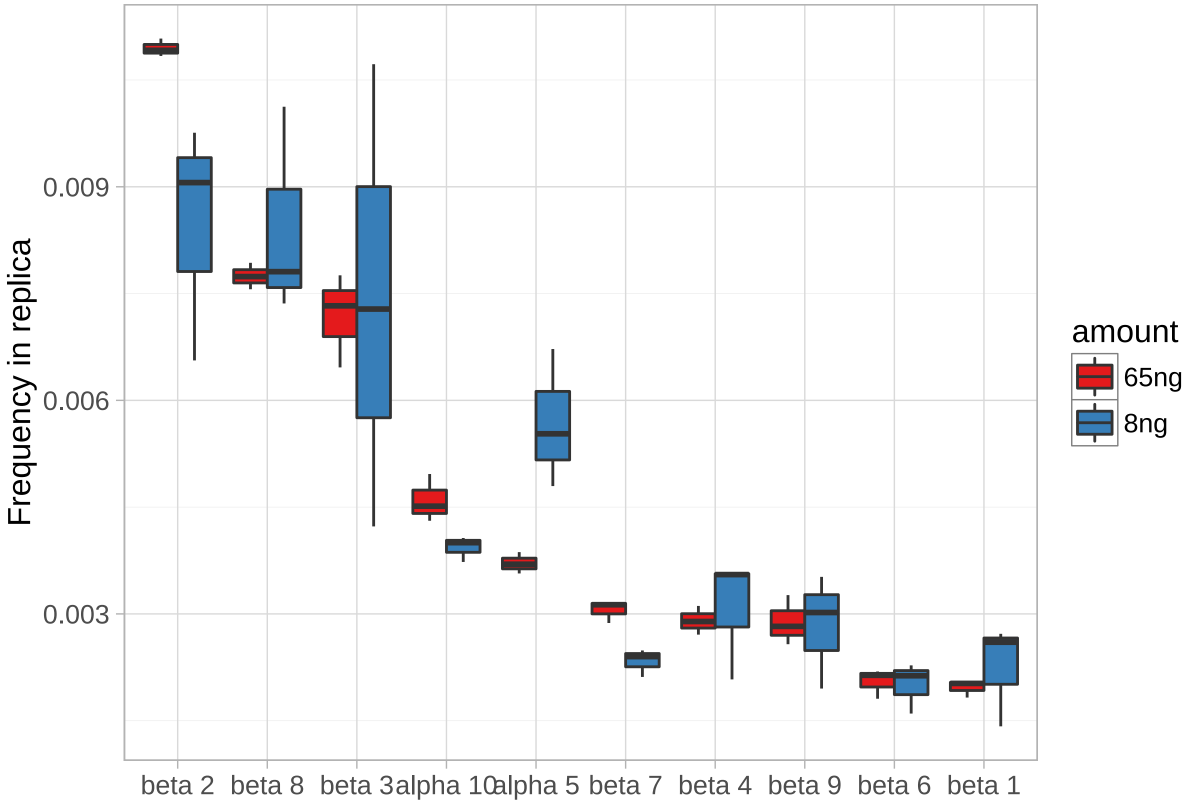
Plot frequency variance for top 10 clonotypes

```
df.4$clone.id = paste(df.4$chain, as.integer(as.factor(df.4$cdr3nt)))
df.4$clone.id = factor(df.4$clone.id, df.4$clone.id[order(-df.4$count.total)])
```

```
## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated
```

```
ggplot(df.4, aes(x=clone.id, group=interaction(clone.id,amount),
                 fill=amount, y=count/count.replica.total)) +
  geom_boxplot() + xlab("") + ylab("Frequency in replica") +
  scale_fill_brewer(palette = "Set1") +
  theme_light()
```

```
## Warning in `levels<-`(`*tmp*`, value = if (nl == nL) as.character(labels)
## else paste0(labels, : duplicated levels in factors are deprecated
```


Frequency in replica
beta 2
beta 8
beta 3
alpha 10
alpha 5
beta 7
beta 4
beta 9
beta 6
beta 1
amount
🟥 65ng
🟦 8ng