R Notebook

```r
suppressMessages(library("tidyverse"))
```

```
## Warning: replacing previous import by 'tidyr::%>%' when loading 'broom'

## Warning: replacing previous import by 'tidyr::gather' when loading 'broom'

## Warning: replacing previous import by 'tidyr::spread' when loading 'broom'
```

```r
library(stringi)
library(plotROC)
```

```
## Warning: replacing previous import by 'rlang::quo_name' when loading
## 'plotROC'
```

```r
variants=read_tsv("illumina_variants.tsv")
```

```
## Parsed with column specification:
## cols(
##   Name = col_character(),
##   Minimum = col_integer(),
##   Maximum = col_double(),
##   Length = col_integer(),
##   Change = col_character(),
##   Coverage = col_double(),
##   `Polymorphism Type` = col_character(),
##   `Variant Frequency` = col_character(),
##   replica = col_character(),
##   modality = col_character(),
##   freq = col_double()
## )
```

```r
barcode1v=read_tsv("BC01.variants.freqs.txt")
```

```
## Parsed with column specification:
## cols(
##   Pos = col_integer(),
##   Qual = col_integer(),
##   Freq = col_double(),
##   Ref = col_character(),
##   Base = col_character(),
##   UngappedCoverage = col_integer(),
##   TotalCoverage = col_integer()
## )
```

```r
barcode1v$replica = 'a'
barcode2v=read_tsv("BC02.variants.freqs.txt")
```

```
## Parsed with column specification:
## cols(
##   Pos = col_integer(),
##   Qual = col_integer(),
##   Freq = col_double(),
##   Ref = col_character(),
##   Base = col_character(),
```

```
##    UngappedCoverage = col_integer(),
##    TotalCoverage = col_integer()
## )
```

```r
barcode2v$replica = 'b'
barcode3v=read_tsv("BC03.variants.freqs.txt")
```

```
## Parsed with column specification:
## cols(
##    Pos = col_integer(),
##    Qual = col_integer(),
##    Freq = col_double(),
##    Ref = col_character(),
##    Base = col_character(),
##    UngappedCoverage = col_integer(),
##    TotalCoverage = col_integer()
## )
```

```r
barcode3v$replica = 'c'
minion_variants=rbind(barcode1v, barcode2v, barcode3v)
```

```r
minion_variants %>%
    filter(Qual == 0) %>%
    write_tsv(path="minion_variants.tsv")
```

```r
barcode1=read_tsv("BC01.freqs.txt")
```

```
## Parsed with column specification:
## cols(
##    Pos = col_integer(),
##    Qual = col_integer(),
##    Freq = col_double(),
##    Ref = col_character(),
##    Base = col_character(),
##    UngappedCoverage = col_integer(),
##    TotalCoverage = col_integer()
## )
```

```r
barcode1$replica = 'a'
barcode2=read_tsv("BC02.freqs.txt")
```

```
## Parsed with column specification:
## cols(
##    Pos = col_integer(),
##    Qual = col_integer(),
##    Freq = col_double(),
##    Ref = col_character(),
##    Base = col_character(),
##    UngappedCoverage = col_integer(),
##    TotalCoverage = col_integer()
## )
```

```r
barcode2$replica = 'b'
barcode3=read_tsv("BC03.freqs.txt")
```

```
## Parsed with column specification:
## cols(
```

```
##   Pos = col_integer(),
##   Qual = col_integer(),
##   Freq = col_double(),
##   Ref = col_character(),
##   Base = col_character(),
##   UngappedCoverage = col_integer(),
##   TotalCoverage = col_integer()
## )
barcode3$replica = 'c'
minion_all=rbind(barcode1, barcode2, barcode3)

minion_all %>%
    filter(Qual == 0) %>%
    write_tsv(path="minion_wt_frequencies.tsv")

expectedpositions=read_tsv("expectedpositions.txt")

## Parsed with column specification:
## cols(
##   Position = col_integer(),
##   State = col_character()
## )
barcode1snps=read_tsv("BC01.variants.0.03.txt")

## Parsed with column specification:
## cols(
##   Pos = col_integer(),
##   Qual = col_integer(),
##   Freq = col_double(),
##   Ref = col_character(),
##   Base = col_character(),
##   UngappedCoverage = col_integer(),
##   TotalCoverage = col_integer(),
##   VariantCov = col_integer(),
##   ForwardVariantCov = col_integer(),
##   ReverseVariantCov = col_integer()
## )
barcode1snps$replica = 'a'
barcode2snps=read_tsv("BC02.variants.0.03.txt")

## Parsed with column specification:
## cols(
##   Pos = col_integer(),
##   Qual = col_integer(),
##   Freq = col_double(),
##   Ref = col_character(),
##   Base = col_character(),
##   UngappedCoverage = col_integer(),
##   TotalCoverage = col_integer(),
##   VariantCov = col_integer(),
##   ForwardVariantCov = col_integer(),
##   ReverseVariantCov = col_integer()
## )
```

```
barcode2snps$replica = 'b'
barcode3snps=read_tsv("BC03.variants.0.03.txt")

## Parsed with column specification:
## cols(
##   Pos = col_integer(),
##   Qual = col_integer(),
##   Freq = col_double(),
##   Ref = col_character(),
##   Base = col_character(),
##   UngappedCoverage = col_integer(),
##   TotalCoverage = col_integer(),
##   VariantCov = col_integer(),
##   ForwardVariantCov = col_integer(),
##   ReverseVariantCov = col_integer()
## )

barcode3snps$replica = 'c'
minion_all_variants=rbind(barcode1snps, barcode2snps, barcode3snps)
minion_all_variants_positions=minion_all_variants %>%
    left_join(expectedpositions, by=c("Pos" = "Position")) %>%
    filter(State != 'Remove')

minion_all_variants_positions %>%
  ggplot(aes(x=ForwardVariantCov, y=ReverseVariantCov, color=State)) + geom_point(size=0.2) + facet_wrap
```



```
minion_all_variants_positions %>%
  mutate(StrandAF = pmin(ForwardVariantCov, ReverseVariantCov) / pmax(ForwardVariantCov, ReverseVariantC
```

4
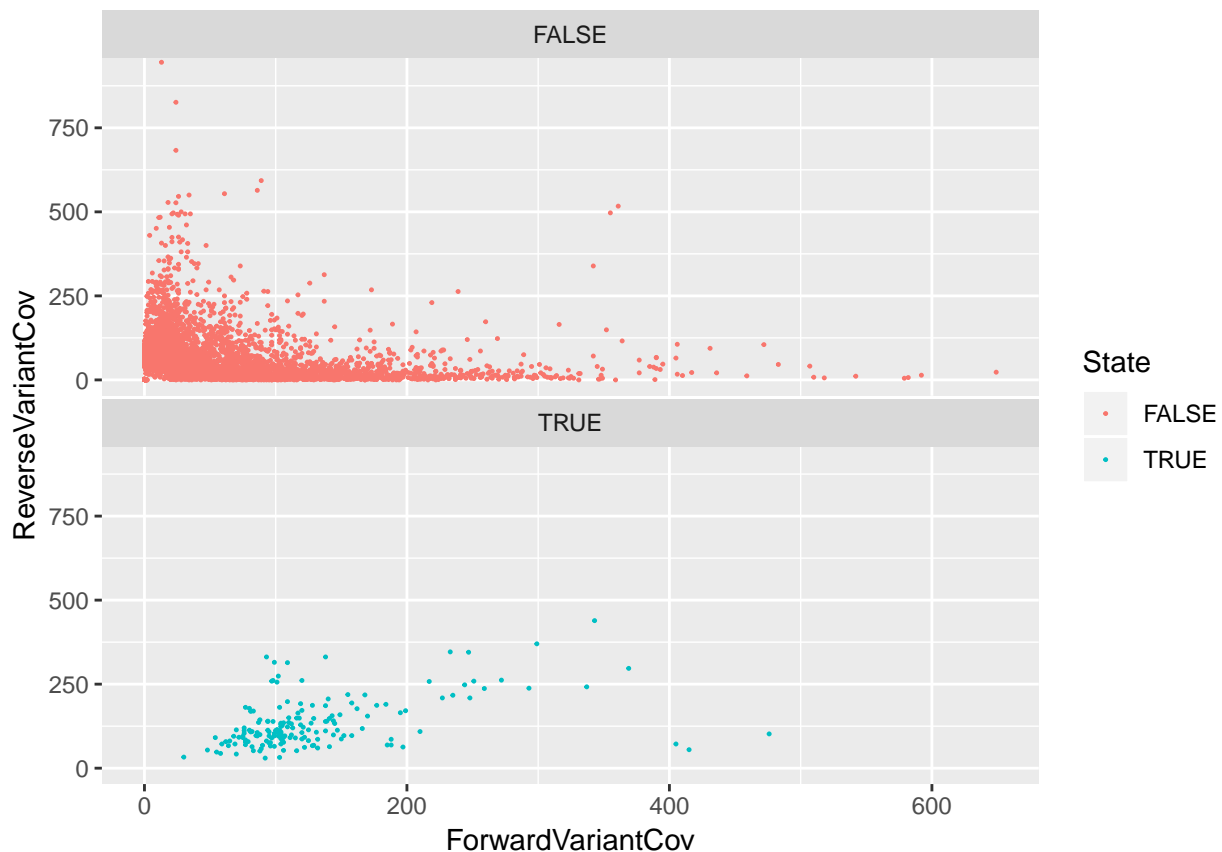
```
## # A tibble: 4,850 x 13
##      Pos  Qual   Freq Ref   Base  UngappedCoverage TotalCoverage
##    <int> <int>  <dbl> <chr> <chr>            <int>         <int>
## 1   1063     0 0.0370 G     T                 2135          2164
## 2   1064     0 0.0426 T     C                 2115          2164
## 3   1066     0 0.0437 G     A                 2104          2164
## 4   1067     0 0.0342 T     A                 2132          2164
## 5   1069     0 0.0402 T     C                 2140          2164
## 6   1070     0 0.0922 C     T                 2083          2164
## 7   1074     0 0.0598 G     A                 2072          2164
## 8   1077     0 0.0456 A     G                 2125          2164
## 9   1078     0 0.0362 T     C                 2126          2164
## 10  1079     0 0.0460 G     A                 2063          2164
## # ... with 4,840 more rows, and 6 more variables: VariantCov <int>,
## #   ForwardVariantCov <int>, ReverseVariantCov <int>, replica <chr>,
## #   State <chr>, StrandAF <dbl>
```

```r
minion_all_variants_positions %>%
      group_by(State) %>%
      summarise(n=n())
```

```
## # A tibble: 2 x 2
##   State     n
##   <chr> <int>
## 1 FALSE  4682
## 2 TRUE    168
```

```r
minion_all_variants_positions %>%
  mutate(StrandAF = pmin(ForwardVariantCov, ReverseVariantCov) / pmax(ForwardVariantCov, ReverseVariantC
  filter(ForwardVariantCov > 10) %>%
  filter(ReverseVariantCov > 10) %>%
  write_tsv("minion_variants_3pc_all.tsv")
```

```r
forroc = minion_all_variants_positions %>%
  mutate(StrandAF = pmin(ForwardVariantCov, ReverseVariantCov) / pmax(ForwardVariantCov, ReverseVariantC
  filter(ForwardVariantCov > 10) %>%
  filter(ReverseVariantCov > 10) %>%
  mutate(D = ifelse(grepl("TRUE", State), 1, 0))
```
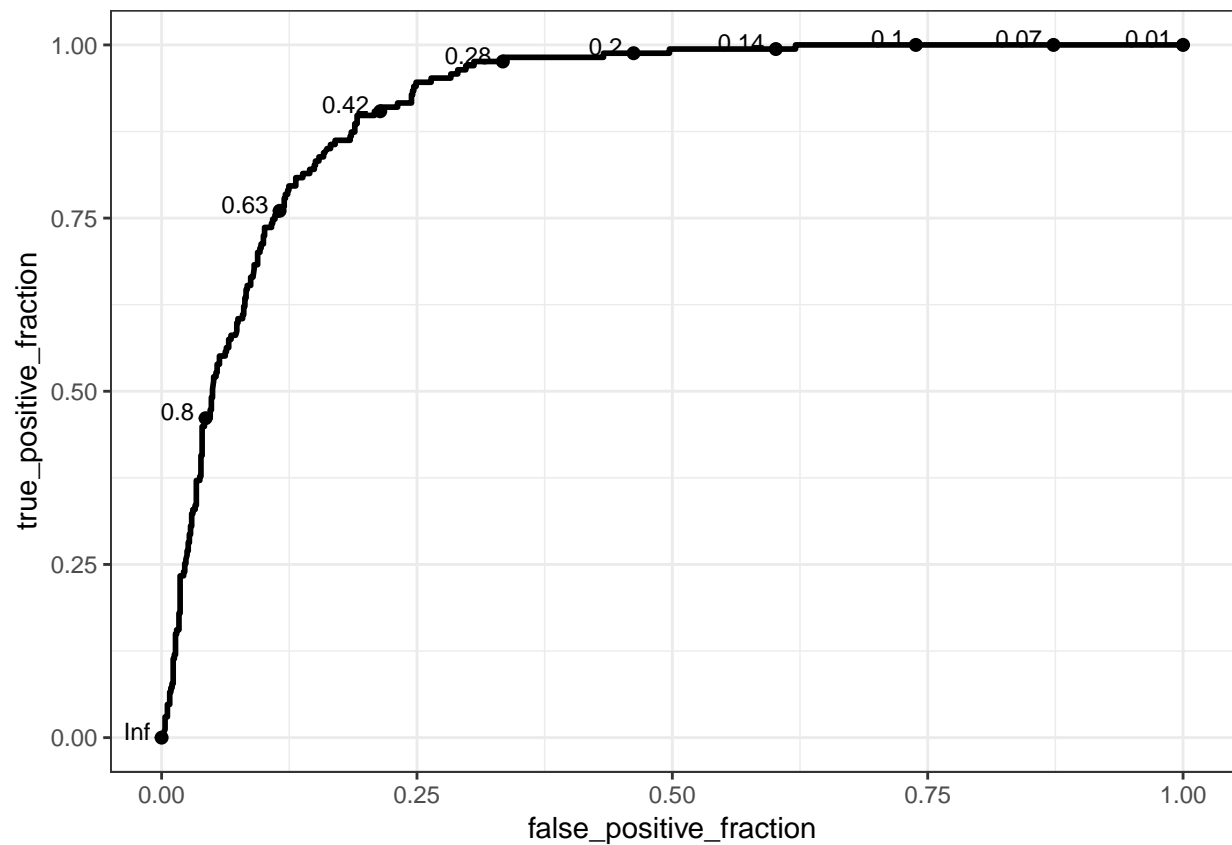
```r
ggplot(forroc, aes(d = D, m = Freq)) + geom_roc(labelsize=3, labelround=2) + theme_bw()
```
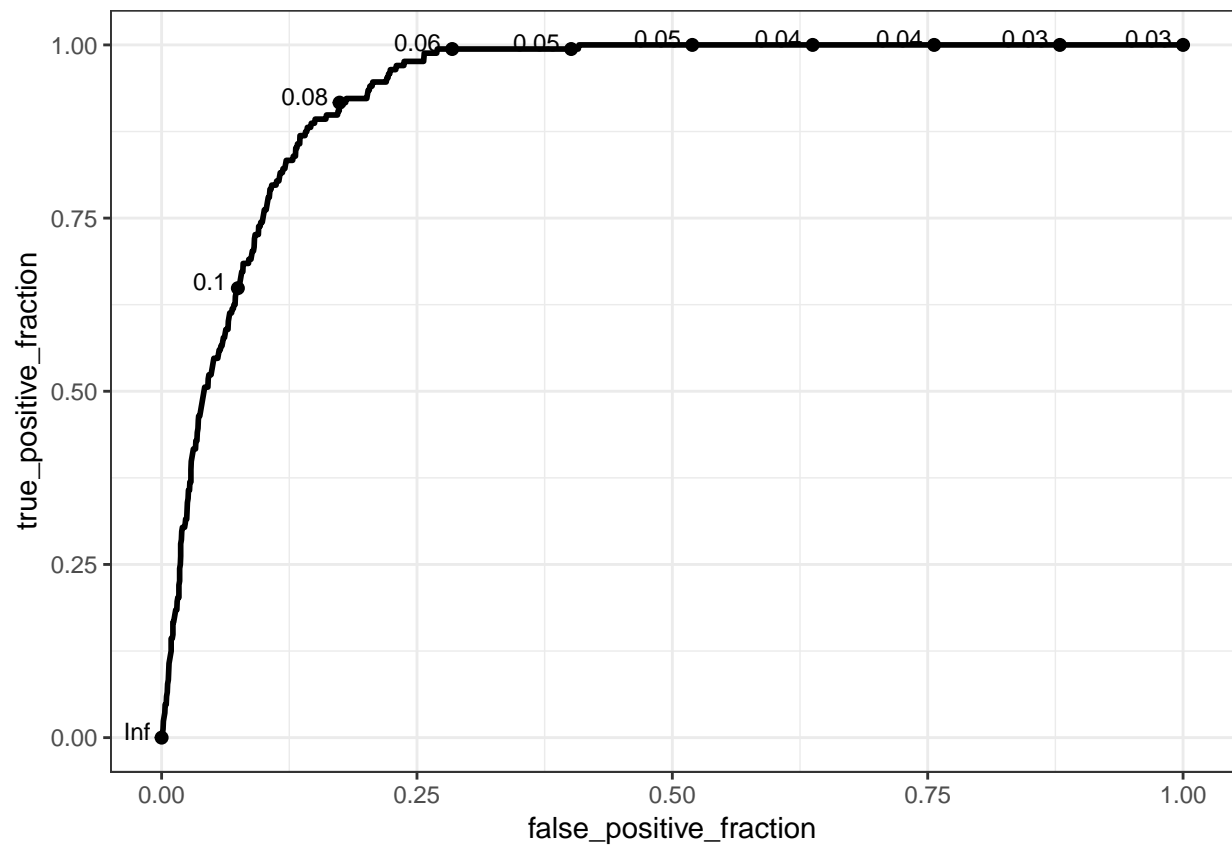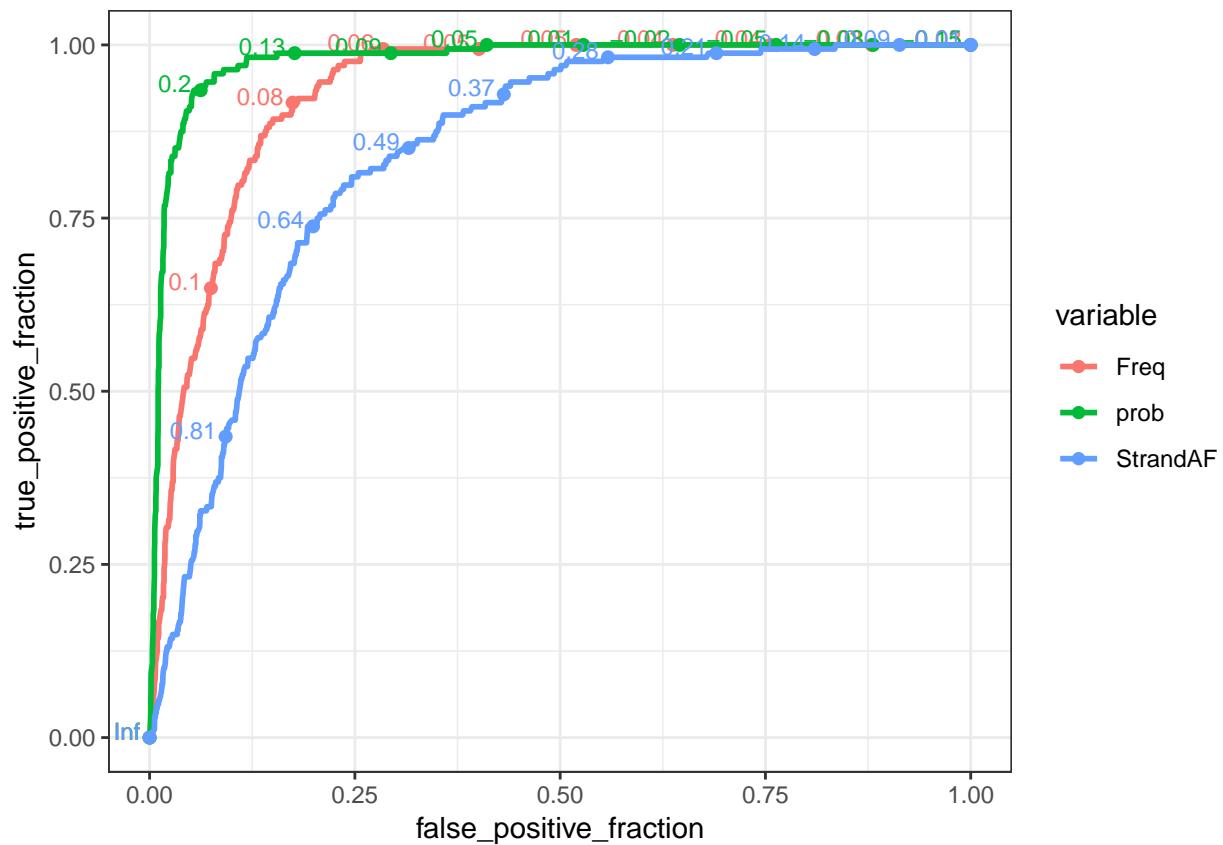
```r
nrow(forroc %>% filter(Freq > 0.06))
```

```
## [1] 1050
```

```r
ggplot(forroc %>% filter(Freq > 0.06), aes(d = D, m = StrandAF)) + geom_roc(labelsize=3, labelround=2)
```

```
mod1<-glm(D ~ Freq + StrandAF, data=forroc, family="gaussian")
prob=predict(mod1,type=c("response"))
forroc$prob = prob
melted=gather(forroc, variable, value, Freq, StrandAF, prob)
ggplot(melted, aes(d = D, color = variable, m = value)) + geom_roc(labelsize=3, labelround=2) + theme_b
```

```r
forroc %>%
  filter(prob > 0.2) %>%
  group_by(D) %>%
  write_tsv("minion_roc_classifier_snps.tsv")
```

```r
forroc %>%
  filter(prob > 0.2) %>%
  group_by(D) %>%
  summarise(n=n())
```

```
## # A tibble: 2 x 2
##       D     n
##   <dbl> <int>
## 1     0   160
## 2     1   157
```

```r
#sens=tp/(tp+fn)
sens=160/(160+8)
print(sens)
```

```
## [1] 0.952381
```

```r
#spec=tn/(tn+fp)
spec=8/(8+157)
print(spec)
```

```
## [1] 0.04848485
```

```
fdr=157/(160+157)
print(fdr)
```

```
## [1] 0.4952681
```

```
minion_all_variants_positions %>%
  mutate(StrandAF = pmin(ForwardVariantCov, ReverseVariantCov) / pmax(ForwardVariantCov, ReverseVariantC
  filter(ForwardVariantCov > 10) %>%
  filter(ReverseVariantCov > 10) %>%
  filter(StrandAF > 0.70) %>%
  write_tsv("minion_variants_3pc_0.7strandaf.tsv")
```

```
minion_all_variants_positions %>%
  ggplot(aes(x=ForwardVariantCov, y=ReverseVariantCov, color=State)) + geom_density2d() + facet_wrap(~S
```
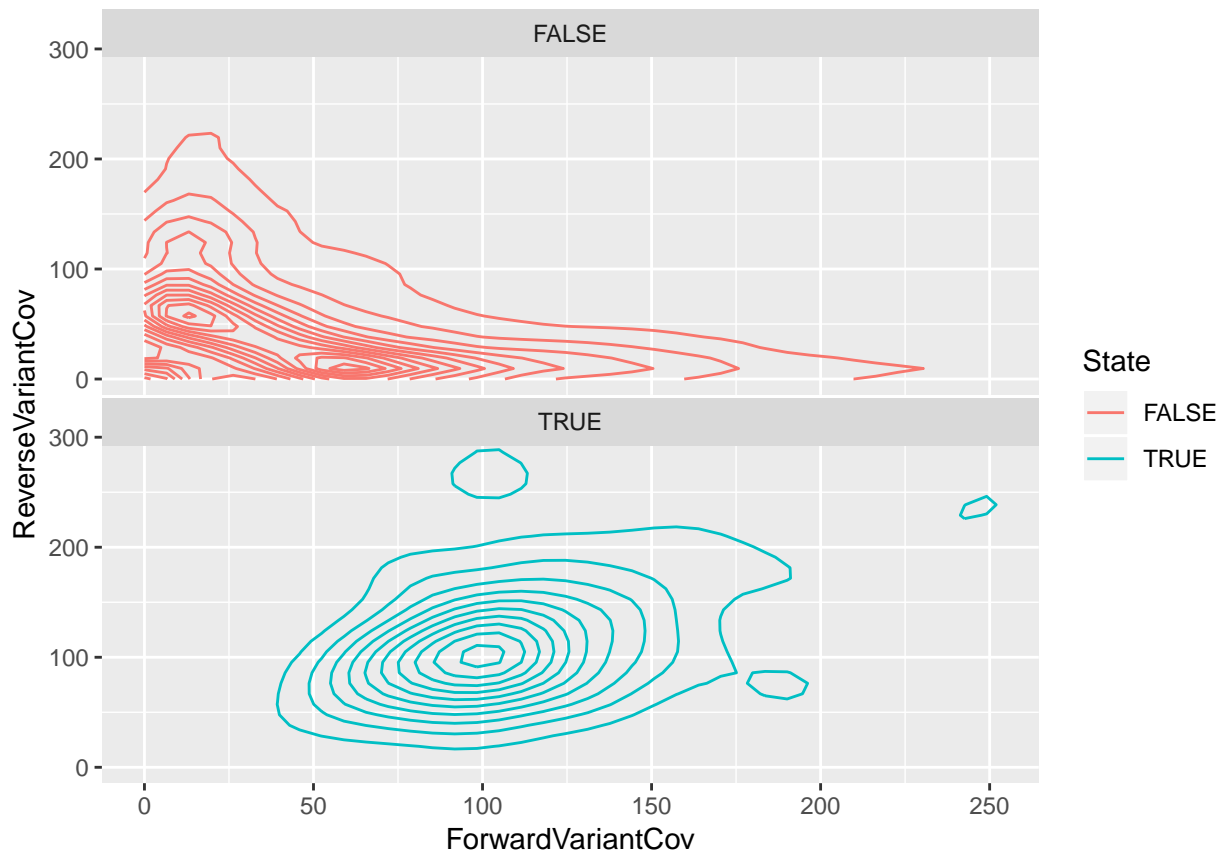


```
minion_all_variants_positions %>%
  ggplot(aes(x=ForwardVariantCov, y=ReverseVariantCov, color=State)) + geom_point(size=0.2) + facet_wrap
```
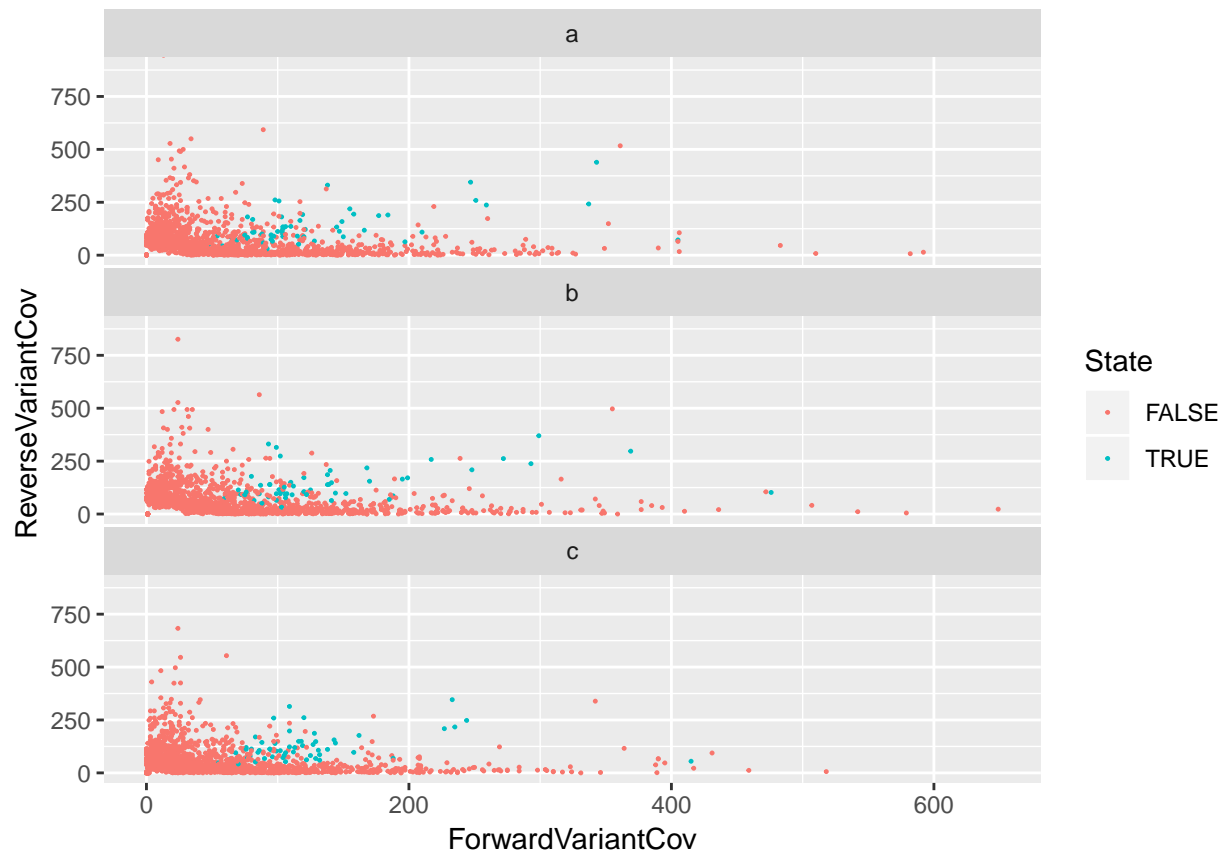
```
variant_positions=inner_join(minion_all, expectedpositions, by=c("Pos" = "Position"), copy=T)
wildtype_positions=anti_join(minion_all, expectedpositions %>% filter(State == "True"), by=c("Pos" = "Po
```

```
wildtype_positions %>%
  filter(UngappedCoverage >= 50) %>%
  ggplot(aes(x=Freq)) + geom_histogram(bins=50) + facet_wrap(~replica, ncol=1) + theme_bw() + xlim(0.75
```

```
## Warning: Removed 101 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 3 rows containing missing values (geom_bar).
```
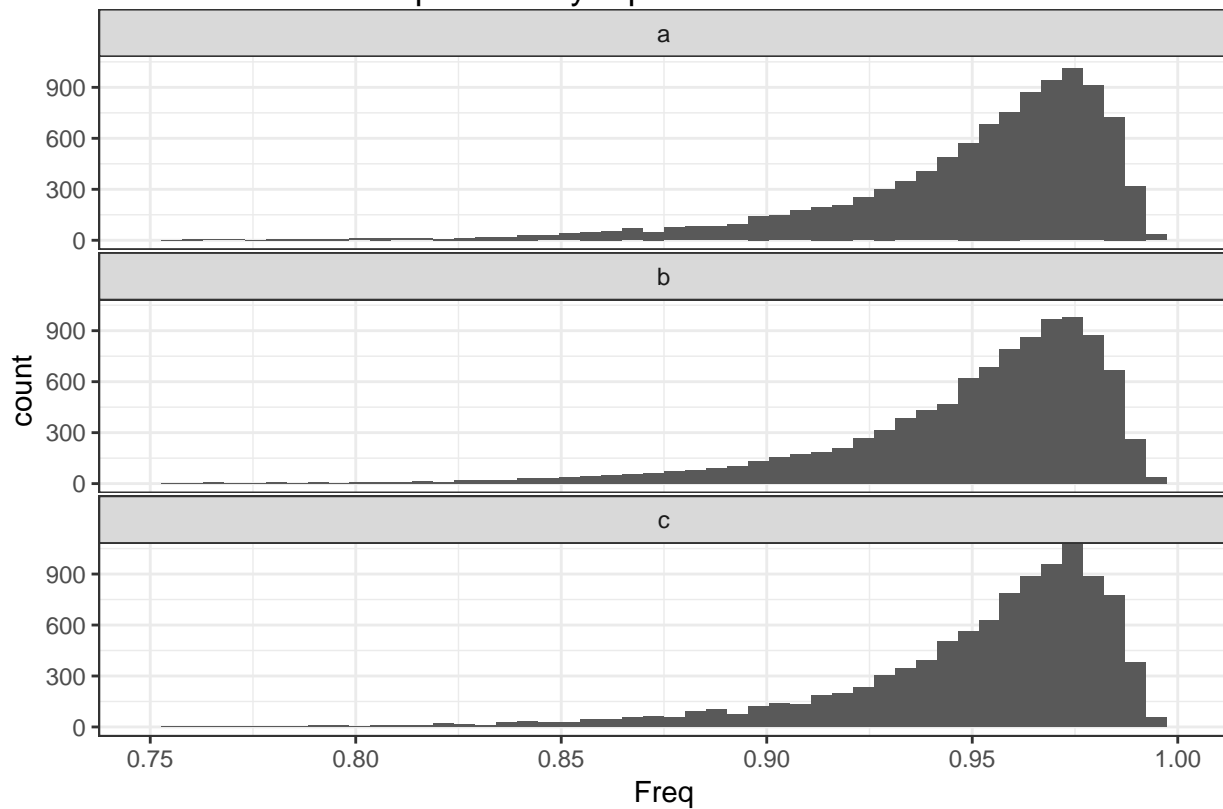
Wildtype allele frequencies by replica
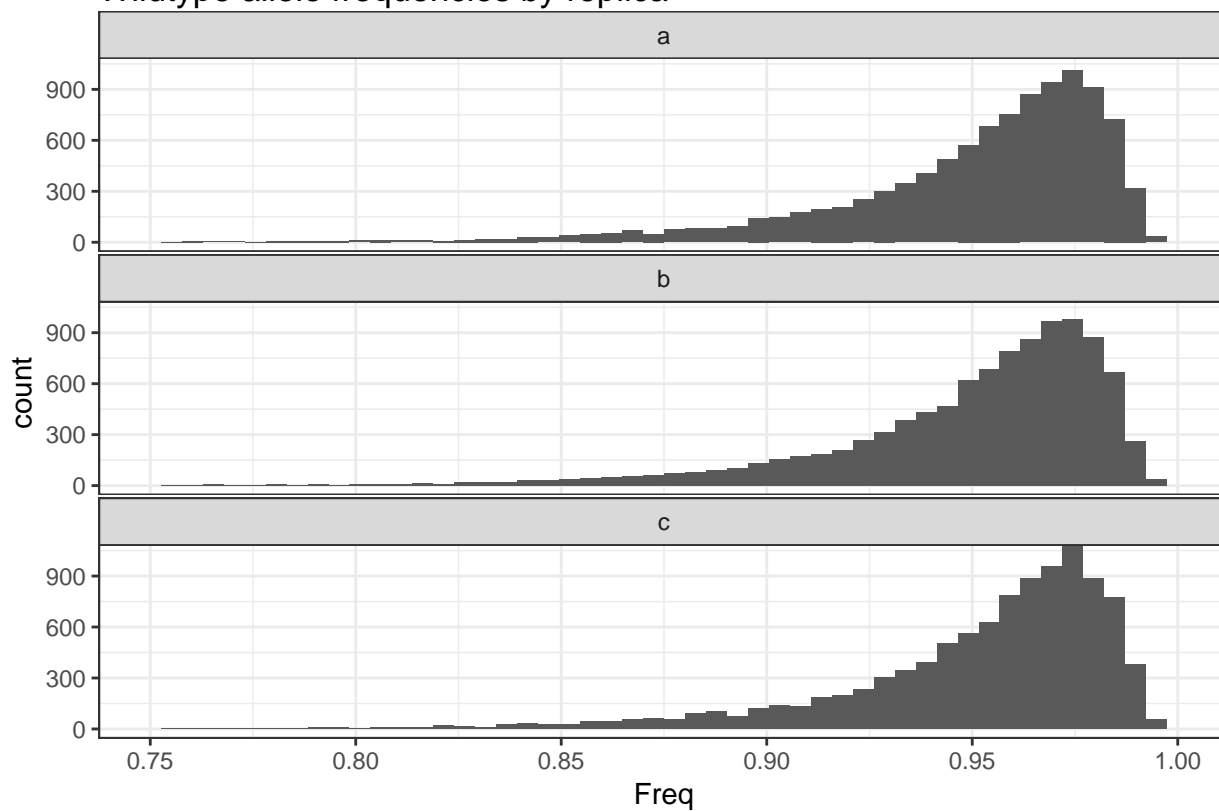

```
variant_positions %>%
  filter(UngappedCoverage >= 50) %>%
  ggplot(aes(x=Freq)) + geom_histogram(bins=50) + facet_wrap(~replica, ncol=1) + theme_bw() + xlim(0.75
```

```
## Warning: Removed 101 rows containing non-finite values (stat_bin).
```

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

## Alternative allele frequencies by replica


```r
joined=inner_join(minion_variants, variants, by=c("Pos" = "Minimum"), copy=T)
p = joined %>%
  filter(Qual == 0) %>%
  filter(modality == 'amplicon') %>%
  ggplot(aes(x=Freq, y=freq)) + geom_point(size=0.2) + stat_smooth() + xlim(0, 0.5) + ylim(0, 0.5) + xl
p
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Removed 12 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 12 rows containing missing values (geom_point).
```
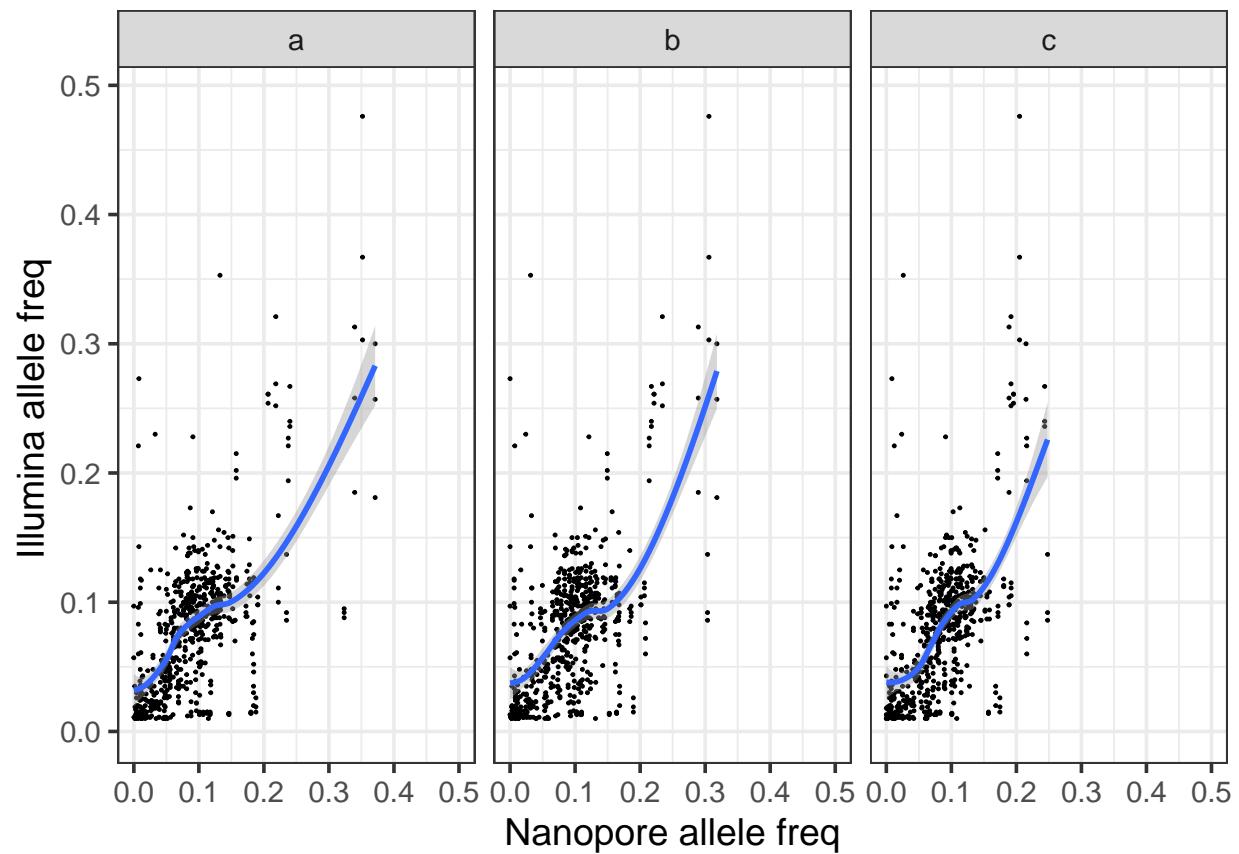
```
a=joined %>%
  filter(modality == 'amplicon') %>%
  filter(replica.x == replica.y) %>%
  ggplot(aes(x=Freq, y=freq)) + geom_point(size=0.2) + geom_density2d() + xlim(0, 0.5) + ylim(0, 0.5) +
a
```

Warning: Removed 4 rows containing non-finite values (stat_density2d).

Warning: Removed 4 rows containing missing values (geom_point).