

MAGERI benchmark using reference standard DNA library

Mikhail Shugay

December 1, 2016

Load metadata

```
df.vmeta <- read.table("hd734_variant_metadata.txt", sep="\t", header=T) # variants observed in HD734 a
df.smeta <- read.table("sample_metadata.txt", sep="\t", header=T) # metadata for amplicon sequencing sa
```

VCF parsing function

```
library(stringr)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
```

```
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      cov, smooth, var
```

```
library(ggplot2)
```

```
library(ggbeeswarm)
```

```
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
```

```
read_vcf <- function(file_name) {
  .vcf <- read.table(file_name, header = F, sep = "\t", stringsAsFactors = F)
  colnames(.vcf) <- c("chromosome", "position", "skip1", "from", "to", "qual", "skip2", "info", "skip3")
  .vcf$skip1 <- NULL
  .vcf$skip2 <- NULL
  .vcf$skip3 <- NULL
  .vcf$skip4 <- NULL

  .vcf$qual <- as.integer(.vcf$qual)
  .vcf <- subset(.vcf, nchar(from) == 1 &
                 nchar(to) == 1 & !is.na(qual)) # no indels

  .infosplit <- str_split_fixed(.vcf$info, regex("[=;]"), 15)[,c(2, 4, 12, 14)]

  .vcf$coverage <- as.numeric(.infosplit[,1])
}
```

```

.vcf$frequency <- as.numeric(.infosplit[,2])
.vcf$bb.a <- as.numeric(.infosplit[,3])
.vcf$bb.b <- as.numeric(.infosplit[,4])
.vcf$info <- NULL

.vcf$count <- as.integer(round(.vcf$coverage * .vcf$frequency))

.vcf
}

head(read_vcf("p126.h4_2_ballast_m1.vcf"))

```

```
## Warning in read_vcf("p126.h4_2_ballast_m1.vcf"): NAs introduced by coercion
```

```
##   chromosome position from to qual coverage   frequency   bb.a
## 1      chr2 212295704   C A    8      2265 0.0004415011 0.9957058
## 2      chr2 212295705   C A    8      2265 0.0004415011 0.9957058
## 3      chr2 212295713   A G    7      2265 0.0004415011 1.0295689
## 4      chr2 212295718   G A    6      2265 0.0004415011 2.1314828
## 5      chr2 212295725   C A    8      2264 0.0004416961 0.9957058
## 6      chr2 212295732   C T   24      2264 0.0013250883 2.1314828
##          bb.b count
## 1 21862.23      1
## 2 21862.23      1
## 3 16394.12      1
## 4 24344.82      1
## 5 21862.23      1
## 6 24344.82      3
```

Read samples with HD734 standard DNA and control human DNA, append metadata

```
library(TailRank) # For betabinom.
```

```
## Loading required package: oompaBase
```

```

# installing:
# source("https://bioconductor.org/biocLite.R")
# biocLite("Biobase")
# install.packages("TailRank", repos="http://R-Forge.R-project.org")

df <- data.frame()

read_vcf_with_metadata <- function(file_name, primer_set, replica, ratio, type) {
  .vcf <- read_vcf(file_name)
  .vcf <- merge(.vcf, df.vmeta, all.x = type != "standard", all.y = F)
  .vcf$known.frequency <- .vcf$known.frequency * ratio
  .vcf$known.frequency[is.na(.vcf$known.frequency)] <- 0
  .vcf$primer_set <- primer_set
  .vcf$replica <- replica
  .vcf$type <- type

  .vcf <- subset(.vcf, frequency < 0.4 & count > 0) # remove alleles in control

  .vcf
}

```

```

for (i in 1:nrow(df.smeta)) {
  df <- with(df.smeta, rbind(df,
                             read_vcf_with_metadata(paste(prefix[i], "vcf", sep="."),
                                                       primer_set[i],
                                                       replica[i],
                                                       ratio[i],
                                                       type[i])))
}

```

```

## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion
## Warning in read_vcf(file_name): NAs introduced by coercion

```

```

# Compute scores

```

```

#df$qual <- with(df,
#  mapply(function(x,y,a,b) 1 - pbb(x,y,a,b) + 0.5 * dbb(x,y,a,b),
#    count, coverage, bb.a, bb.b)
# )

```

```
#df$qual <- ifelse(df$qual <= 1e-100, 999, -10*log10(df$qual))
```

Group observed variants into tiers

```
df$tier <- cut(df$known.frequency, c(-1, 0, 0.009, 0.02, 1))
levels(df$tier) <- c("error", "0.1%", "1%", "5%")
summary(df$tier)
```

```
## error 0.1% 1% 5%
## 2468 43 46 12
```

Number of detected variants and sampling

```
df.v.summary <- df.vmeta
df.v.summary$known.frequency <- df.v.summary$known.frequency * 0.1 # 1:9 diluted samples
df.v.summary$ratio <- "1:10"
df.v.summary.2 <- df.vmeta
df.v.summary.2$ratio <- "1:1"
df.v.summary <- rbind(df.v.summary, df.v.summary.2)
df.v.summary$tier <- cut(df.v.summary$known.frequency, c(-1, 0, 0.009, 0.02, 1))
levels(df.v.summary$tier) <- c("error", "0.1%", "1%", "5%")

print(df.v.summary)
```

| ## | chromosome | position | from | to | id |
|-------|------------|-----------|------|----|----------------------------------|
| ## 1 | chr7 | 140453136 | A | T | BRAF_V600E,BRAF_V600K |
| ## 2 | chr7 | 140453136 | A | C | BRAF_V600G,BRAF_V600R |
| ## 3 | chr7 | 140453137 | C | T | BRAF_V600M,BRAF_V600K,BRAF_V600R |
| ## 4 | chr7 | 55241707 | G | A | EGFR_G719S |
| ## 5 | chr7 | 55259515 | T | G | EGFR_L858R |
| ## 6 | chr7 | 55259524 | T | A | EGFR_L861Q |
| ## 7 | chr7 | 55249071 | C | T | EGFR_T790M |
| ## 8 | chr19 | 3118942 | A | T | GNA11_Q209L |
| ## 9 | chr2 | 209113112 | C | T | IDH1_R132C |
| ## 10 | chr2 | 209113113 | G | A | IDH1_R132H |
| ## 11 | chr12 | 25378562 | C | T | KRAS_A146T |
| ## 12 | chr12 | 25398284 | C | G | KRAS_G12A |
| ## 13 | chr12 | 25398285 | C | A | KRAS_G12C |
| ## 14 | chr12 | 25398284 | C | T | KRAS_G12D |
| ## 15 | chr12 | 25398285 | C | G | KRAS_G12R |
| ## 16 | chr12 | 25398285 | C | T | KRAS_G12S |
| ## 17 | chr12 | 25398284 | C | A | KRAS_G12V |
| ## 18 | chr12 | 25398281 | C | T | KRAS_G13D |
| ## 19 | chr12 | 25380275 | T | G | KRAS_Q61H |
| ## 20 | chr12 | 25380276 | T | A | KRAS_Q61L |
| ## 21 | chr1 | 115256528 | T | A | NRAS_Q61H |
| ## 22 | chr1 | 115256530 | G | T | NRAS_Q61K |
| ## 23 | chr1 | 115256529 | T | A | NRAS_Q61L |
| ## 24 | chr1 | 115256529 | T | C | NRAS_Q61R |
| ## 25 | chr3 | 178936082 | G | A | PIK3CA_E542K |
| ## 26 | chr3 | 178936091 | G | A | PIK3CA_E545K |
| ## 27 | chr3 | 178952085 | A | G | PIK3CA_H1047R |
| ## 28 | chr7 | 140453136 | A | T | BRAF_V600E,BRAF_V600K |
| ## 29 | chr7 | 140453136 | A | C | BRAF_V600G,BRAF_V600R |
| ## 30 | chr7 | 140453137 | C | T | BRAF_V600M,BRAF_V600K,BRAF_V600R |
| ## 31 | chr7 | 55241707 | G | A | EGFR_G719S |

| | | | | | |
|-------|-----------------|------------|-------|------|---------------|
| ## 32 | chr7 | 55259515 | T | G | EGFR_L858R |
| ## 33 | chr7 | 55259524 | T | A | EGFR_L861Q |
| ## 34 | chr7 | 55249071 | C | T | EGFR_T790M |
| ## 35 | chr19 | 3118942 | A | T | GNA11_Q209L |
| ## 36 | chr2 | 209113112 | C | T | IDH1_R132C |
| ## 37 | chr2 | 209113113 | G | A | IDH1_R132H |
| ## 38 | chr12 | 25378562 | C | T | KRAS_A146T |
| ## 39 | chr12 | 25398284 | C | G | KRAS_G12A |
| ## 40 | chr12 | 25398285 | C | A | KRAS_G12C |
| ## 41 | chr12 | 25398284 | C | T | KRAS_G12D |
| ## 42 | chr12 | 25398285 | C | G | KRAS_G12R |
| ## 43 | chr12 | 25398285 | C | T | KRAS_G12S |
| ## 44 | chr12 | 25398284 | C | A | KRAS_G12V |
| ## 45 | chr12 | 25398281 | C | T | KRAS_G13D |
| ## 46 | chr12 | 25380275 | T | G | KRAS_Q61H |
| ## 47 | chr12 | 25380276 | T | A | KRAS_Q61L |
| ## 48 | chr1 | 115256528 | T | A | NRAS_Q61H |
| ## 49 | chr1 | 115256530 | G | T | NRAS_Q61K |
| ## 50 | chr1 | 115256529 | T | A | NRAS_Q61L |
| ## 51 | chr1 | 115256529 | T | C | NRAS_Q61R |
| ## 52 | chr3 | 178936082 | G | A | PIK3CA_E542K |
| ## 53 | chr3 | 178936091 | G | A | PIK3CA_E545K |
| ## 54 | chr3 | 178952085 | A | G | PIK3CA_H1047R |
| ## | known.frequency | reference | ratio | tier | |
| ## 1 | 0.0080 | BRAF_E15 | 1:10 | 0.1% | |
| ## 2 | 0.0010 | BRAF_E15 | 1:10 | 0.1% | |
| ## 3 | 0.0010 | BRAF_E15 | 1:10 | 0.1% | |
| ## 4 | 0.0167 | EGFR_E18 | 1:10 | 1% | |
| ## 5 | 0.0010 | EGFR_E21 | 1:10 | 0.1% | |
| ## 6 | 0.0010 | EGFR_E21 | 1:10 | 0.1% | |
| ## 7 | 0.0010 | EGFR_E20 | 1:10 | 0.1% | |
| ## 8 | 0.0013 | GNA11_E5 | 1:10 | 0.1% | |
| ## 9 | 0.0013 | IDH1_E4 | 1:10 | 0.1% | |
| ## 10 | 0.0013 | IDH1_E4 | 1:10 | 0.1% | |
| ## 11 | 0.0013 | KRAS_E4 | 1:10 | 0.1% | |
| ## 12 | 0.0013 | KRAS_E2 | 1:10 | 0.1% | |
| ## 13 | 0.0013 | KRAS_E2 | 1:10 | 0.1% | |
| ## 14 | 0.0013 | KRAS_E2 | 1:10 | 0.1% | |
| ## 15 | 0.0013 | KRAS_E2 | 1:10 | 0.1% | |
| ## 16 | 0.0013 | KRAS_E2 | 1:10 | 0.1% | |
| ## 17 | 0.0013 | KRAS_E2 | 1:10 | 0.1% | |
| ## 18 | 0.0250 | KRAS_E2 | 1:10 | 5% | |
| ## 19 | 0.0013 | KRAS_E3 | 1:10 | 0.1% | |
| ## 20 | 0.0013 | KRAS_E3 | 1:10 | 0.1% | |
| ## 21 | 0.0013 | NRAS_E3 | 1:10 | 0.1% | |
| ## 22 | 0.0013 | NRAS_E3 | 1:10 | 0.1% | |
| ## 23 | 0.0013 | NRAS_E3 | 1:10 | 0.1% | |
| ## 24 | 0.0013 | NRAS_E3 | 1:10 | 0.1% | |
| ## 25 | 0.0013 | PIK3CA_E9 | 1:10 | 0.1% | |
| ## 26 | 0.0013 | PIK3CA_E9 | 1:10 | 0.1% | |
| ## 27 | 0.0300 | PIK3CA_E20 | 1:10 | 5% | |
| ## 28 | 0.0800 | BRAF_E15 | 1:1 | 5% | |
| ## 29 | 0.0100 | BRAF_E15 | 1:1 | 1% | |
| ## 30 | 0.0100 | BRAF_E15 | 1:1 | 1% | |

```
## 31      0.1670  EGFR_E18  1:1  5%
## 32      0.0100  EGFR_E21  1:1  1%
## 33      0.0100  EGFR_E21  1:1  1%
## 34      0.0100  EGFR_E20  1:1  1%
## 35      0.0130  GNA11_E5  1:1  1%
## 36      0.0130  IDH1_E4   1:1  1%
## 37      0.0130  IDH1_E4   1:1  1%
## 38      0.0130  KRAS_E4    1:1  1%
## 39      0.0130  KRAS_E2    1:1  1%
## 40      0.0130  KRAS_E2    1:1  1%
## 41      0.0130  KRAS_E2    1:1  1%
## 42      0.0130  KRAS_E2    1:1  1%
## 43      0.0130  KRAS_E2    1:1  1%
## 44      0.0130  KRAS_E2    1:1  1%
## 45      0.2500  KRAS_E2    1:1  5%
## 46      0.0130  KRAS_E3    1:1  1%
## 47      0.0130  KRAS_E3    1:1  1%
## 48      0.0130  NRAS_E3    1:1  1%
## 49      0.0130  NRAS_E3    1:1  1%
## 50      0.0130  NRAS_E3    1:1  1%
## 51      0.0130  NRAS_E3    1:1  1%
## 52      0.0130  PIK3CA_E9  1:1  1%
## 53      0.0130  PIK3CA_E9  1:1  1%
## 54      0.3000  PIK3CA_E20 1:1  5%
```

```
df.v.summary <- df.v.summary %>%
  group_by(tier) %>%
  summarise(total = n())
df.v.summary$total <- 2 * df.v.summary$total # we have 2 independent experiments

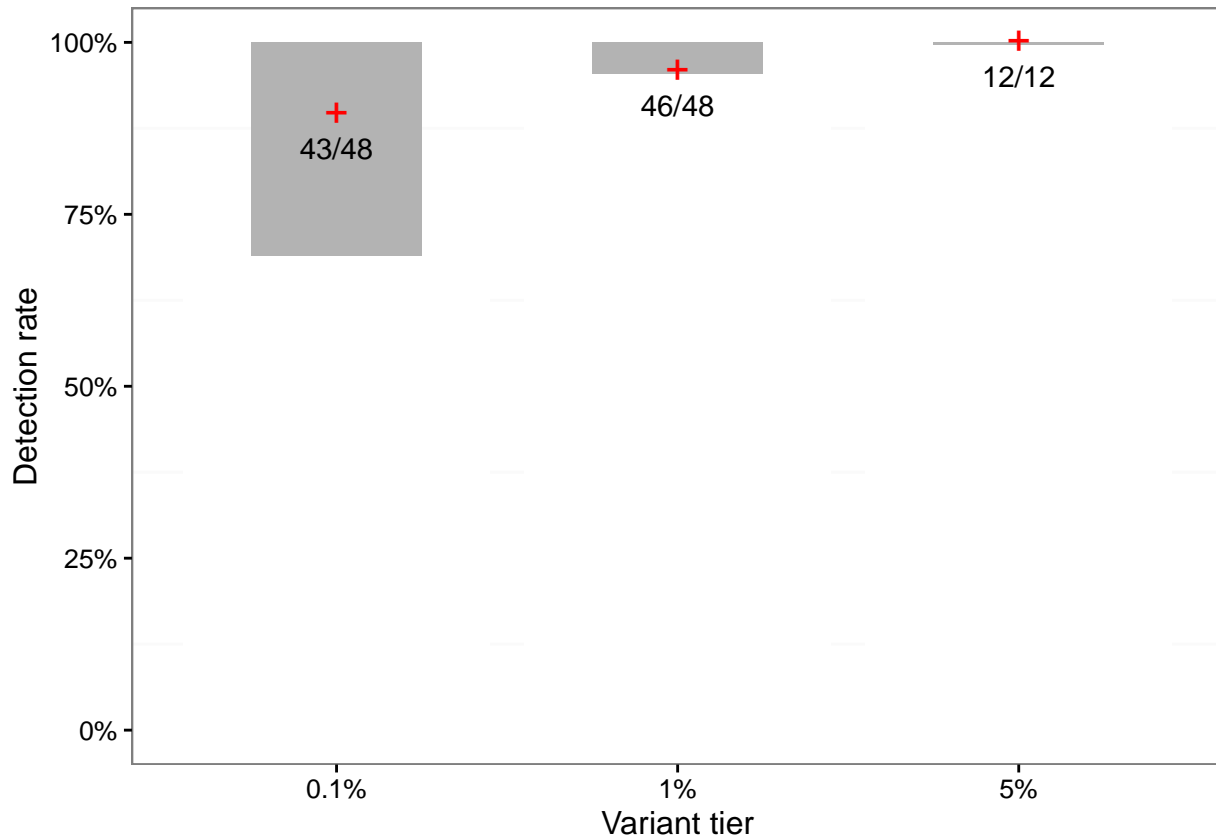
df.v.summary <- merge(df.v.summary,
  df %>% group_by(tier) %>%
    summarise(found = n(),
              detect.prob.m = mean(1-exp(-frequency * coverage)),
              detect.prob.sd = sd(1-exp(-frequency * coverage)))

df.v.summary$detection.prob.lb <- with(df.v.summary,
  detect.prob.m - 1.96*detect.prob.sd - 5e-3)
df.v.summary$detection.prob.ub <- with(df.v.summary,
  detect.prob.m + 1.96*detect.prob.sd)

df.v.summary$detection.prob.ub <- ifelse(df.v.summary$detection.prob.ub > 1,
  1,
  df.v.summary$detection.prob.ub )

library(scales)
ggplot(df.v.summary, aes(x=tier)) +
  geom_bar(stat="identity", aes(y=detection.prob.ub), fill = "grey70", width=0.5)+
  geom_bar(stat="identity", aes(y=detection.prob.lb), fill = "white")+
  geom_point(aes(y = found/total), size = 5, color="red", shape="+") +
  geom_text(aes(y = found/total - 0.05, label = paste(found, total, sep="/"))) +
  scale_y_continuous("Detection rate", labels=percent) +
  xlab("Variant tier") +
  theme_bw() +
```

```
theme(panel.grid.major.x = element_blank(),
      panel.grid.major.y = element_blank())
```

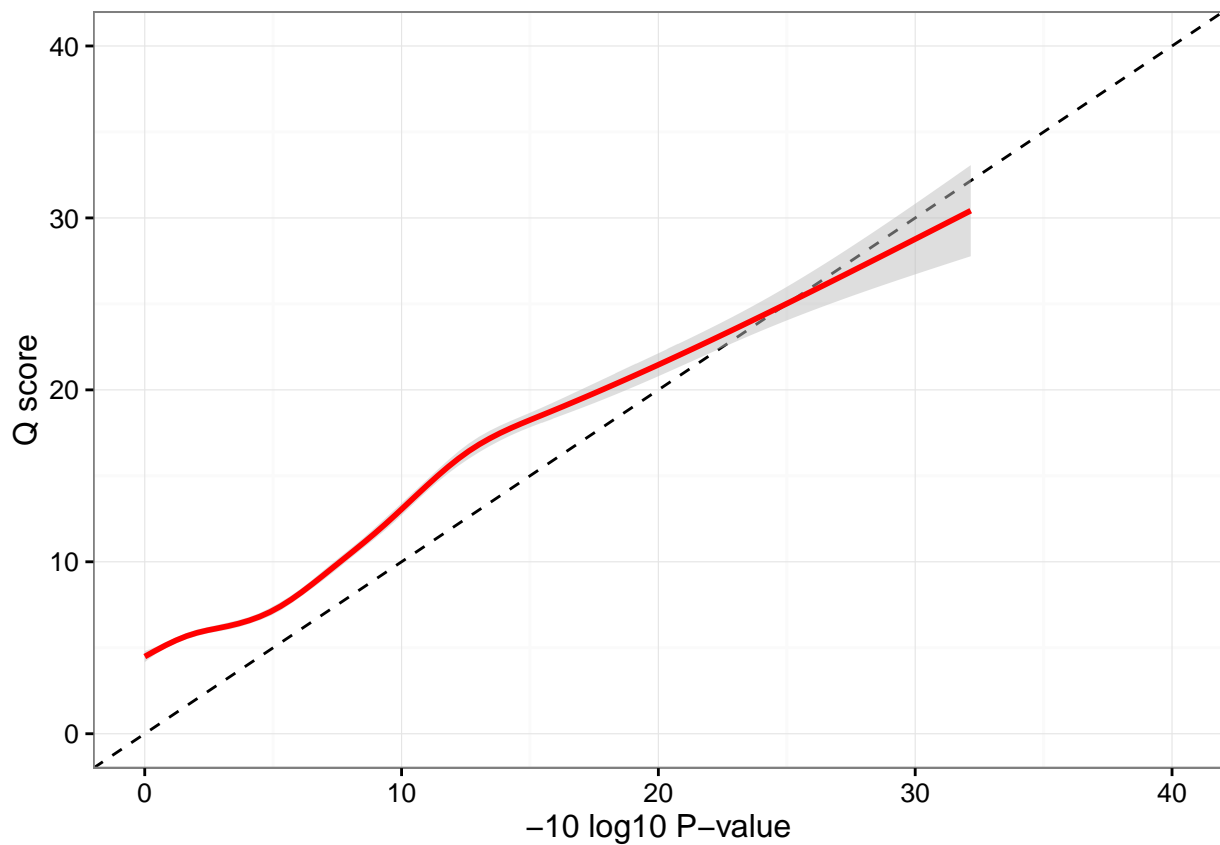


Quality score and error P-values

```
df.e <- df %>%
  filter(type == "blank") %>%
  mutate(true.p.value = -10 * log10(1 - (rank(frequency) - 0.5) / n()))

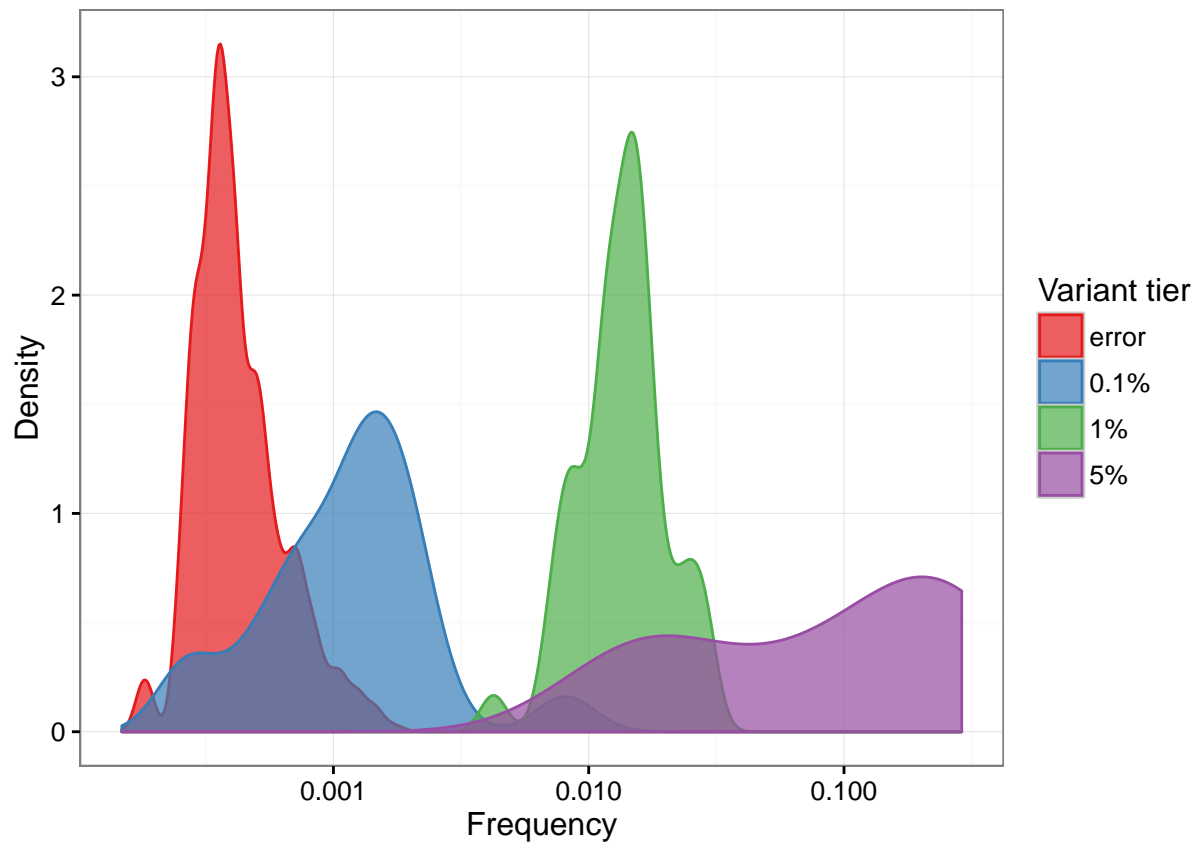
ggplot(df.e, aes(true.p.value, qual)) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  geom_smooth(color="red", fill="grey", alpha=0.5) +
  scale_x_continuous("-10 log10 P-value", limits = c(0, 40)) +
  scale_y_continuous("Q score", limits = c(0, 40)) +
  theme_bw()
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```



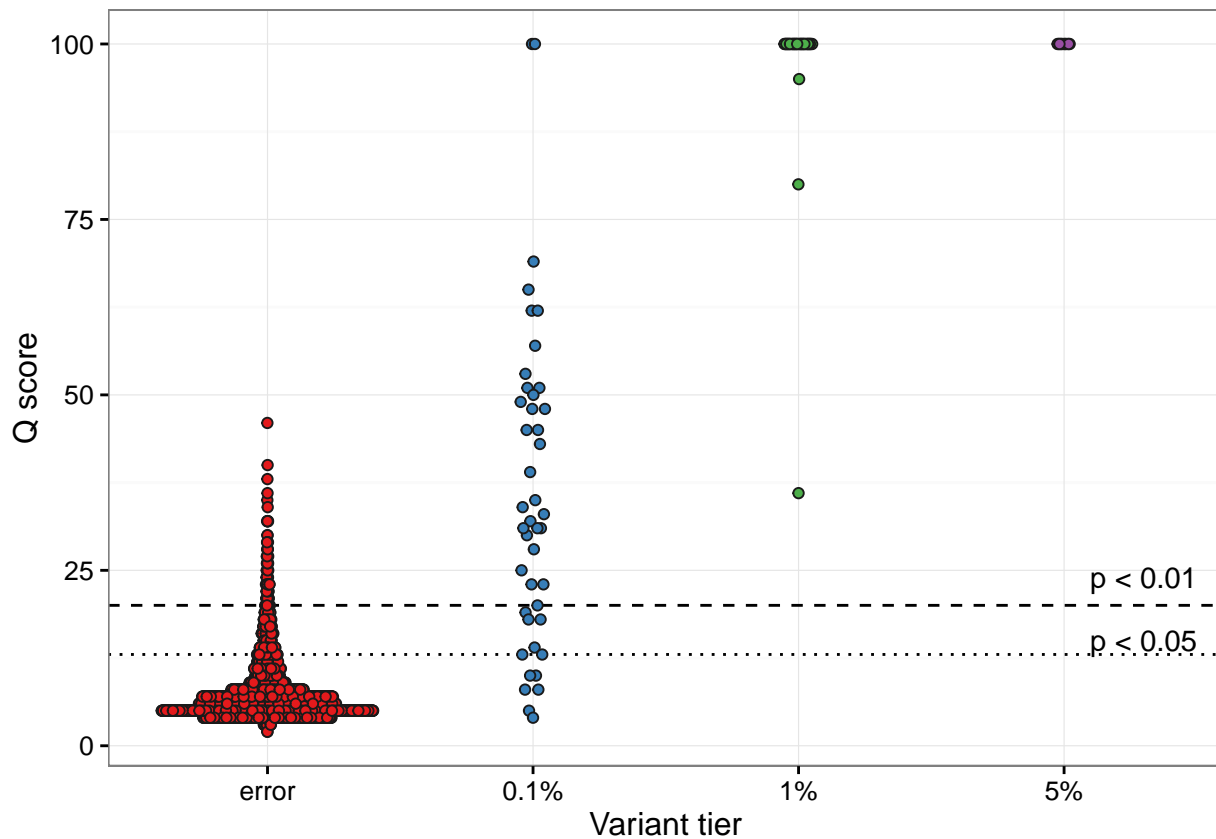
Frequency of variants and errors

```
ggplot(df, aes(x=frequency, color = tier)) +
  geom_density(aes(fill=tier), alpha=0.7) +
  ylab("Density") +
  scale_x_log10("Frequency") +
  scale_color_brewer("Variant tier", palette = "Set1") +
  scale_fill_brewer("Variant tier", palette = "Set1") +
  theme_bw()
```

Variant quality scores

```
ggplot(df,aes(tier, qual, fill=tier)) +
  geom_hline(yintercept = 13.0103, linetype ="dotted") +
  geom_hline(yintercept = 20.0103, linetype ="dashed") +
  geom_quasirandom(varwidth = T, shape=21, color="grey10") +
  annotate("text", label = "p < 0.05", x=4.3,y=13.0103+2) +
  annotate("text", label = "p < 0.01", x=4.3,y=20.0103+4) +
  scale_fill_brewer("", palette = "Set1", guide=F) +
  xlab("Variant tier") + ylab("Q score") +
  theme_bw()
```



ROC curve for rare variants

```
df.1 <- subset(df, tier %in% c("error", "0.1%"))
df.1$type <- ifelse(df.1$tier == "error", 0, 1)

make_roc <- function(rocobj, type) {
  .df.roc <- data.frame(spec = seq(0, 1, 0.01))
  sens.ci <- ci.se(rocobj, specificities = .df.roc$spec)

  .df.roc$sens.lo <- sens.ci[,1]
  .df.roc$sens.me <- sens.ci[,2]
  .df.roc$sens.hi <- sens.ci[,3]
  .df.roc$type <- type

  .df.roc
}

rocobj <- roc(type ~ qual, df.1, ci=T)
print(rocobj)

##
## Call:
## roc.formula(formula = type ~ qual, data = df.1, ci = T)
##
## Data: qual in 2468 controls (type 0) < 43 cases (type 1).
## Area under the curve: 0.9277
## 95% CI: 0.8727-0.9827 (DeLong)
```

```

df.roc <- make_roc(rocobj, "Q score")
rocobj <- roc(type ~ frequency, df.1, ci=T)
print(rocobj)

##
## Call:
## roc.formula(formula = type ~ frequency, data = df.1, ci = T)
##
## Data: frequency in 2468 controls (type 0) < 43 cases (type 1).
## Area under the curve: 0.8637
## 95% CI: 0.7839-0.9435 (DeLong)

df.roc <- rbind(df.roc, make_roc(rocobj, "Frequency"))

ggplot(df.roc, aes(x=spec)) +
  geom_ribbon(aes(ymin=sens.lo, ymax=sens.hi, group=type), fill="grey", alpha=0.5) +
  geom_abline(slope = 1, intercept = 1, linetype = "dashed") +
  geom_line(aes(y=sens.me, color = type)) +
  scale_x_reverse("Specificity") +
  scale_y_continuous("Sensitivity", limits=c(0,1)) +
  scale_color_brewer("Threshold", palette = "Set1") +
  theme_bw()

```

