MAGERI benchmark using various datasets

ctDNA detection

Load data from ctDNA experiment

```
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(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", "skip4")
  .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("[=;]"), 11)[,c(2, 4, 10)]

  .vcf$coverage <- as.numeric(.infosplit[,1])
  .vcf$frequency <- as.numeric(.infosplit[,2])
  .vcf$region <- .infosplit[,3]
  .vcf$info <- NULL

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

  subset(.vcf, nchar(from) == 1 & nchar(to) == 1 & !is.na(qual))
```

```
}

read_vcf_ctdna <- function(file_name, donor, sample) {
  .df <- read_vcf(file_name)
  .df$donor <- donor
  .df$sample <- sample
  subset(.df, region == "BRAF_E15")
}

df <- data.frame()
df <- rbind(df, read_vcf_ctdna("p92.c41.13_plasma.vcf", "donor1", "plasma"))
```

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

```
df <- rbind(df, read_vcf_ctdna("p92.c41.13_tumor.vcf", "donor1", "tumor"))
```

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

```
df <- rbind(df, read_vcf_ctdna("p92.c41.21_plasma.vcf", "donor2", "plasma"))
df <- rbind(df, read_vcf_ctdna("p92.c41.21_tumor.vcf", "donor2", "tumor"))
```

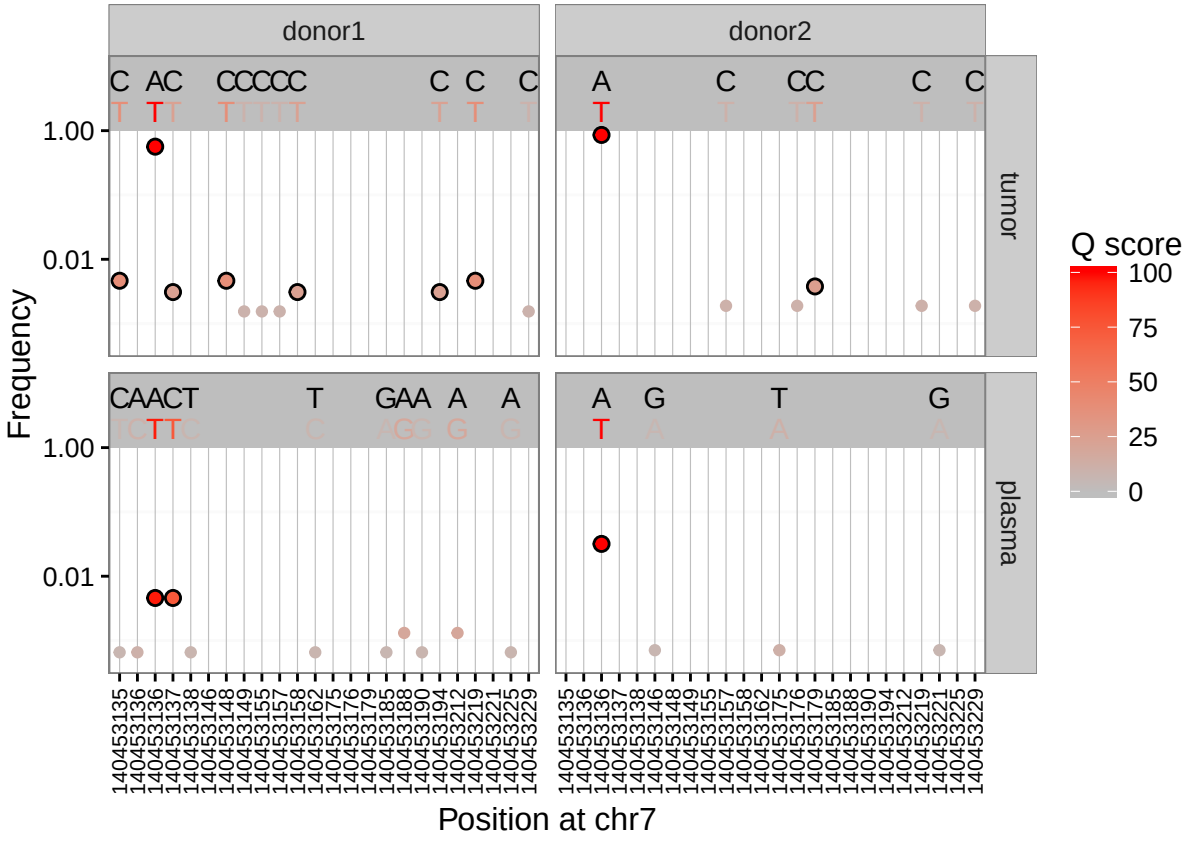
```
## Warning in read_vcf(file_name): NAs introduced by coercion
```

Plot variants

```
df$sample <- factor(df$sample, levels = c("tumor", "plasma"))

df <- df %>% arrange(position, from, to) %>%
  mutate(xx = paste(position, from, to))

ggplot(df, aes(x=xx, y = frequency, color = qual)) +
  geom_text(aes(y=6.0, label = from), color="black") + # trick
  annotate(geom = "rect", xmin=-Inf, xmax = Inf, ymin=1,ymax=Inf, fill="grey") +
  geom_text(aes(y=6.0, label = from), color="black") +
  geom_text(aes(y=2.0, label = to)) +
  geom_point(data=subset(df, qual > 20), size=2.5, color="black") +
  geom_point() +
  scale_y_log10("Frequency", limits=c(0.0005, 9)) +
  facet_grid(sample~donor) +
  scale_x_discrete("Position at chr7",
                   label=function(x) str_split_fixed(x, " ", 3)[,1]) +
  scale_color_gradient("Q score", limits = c(0,100), low = "grey", high="red") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, vjust=0.5,size = 8),
        panel.grid.major.x = element_line(color="grey"),
        panel.grid.major.y = element_blank())
```



Compute P-value for composite variant in donor1

```
df.c <- subset(df, xx %in% c("140453136 A T", "140453137 C T") &
                  sample == "plasma" & donor == "donor1")[1,]

with(df.c, 1 - phyper(count,count,coverage-count,count) +
       0.5 * dhyper(count,count,coverage-count,count))
```

```
## [1] 1.331989e-19
```

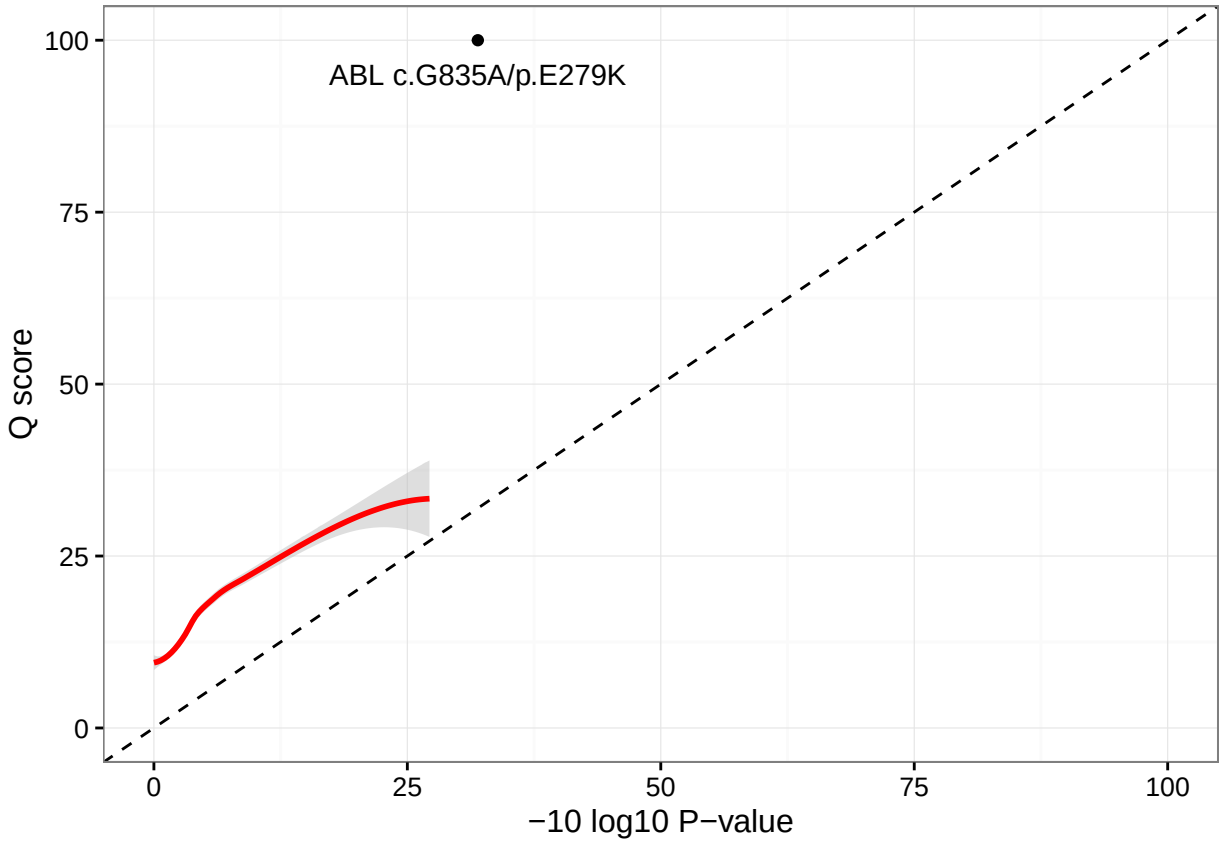
Duplex sequencing

```
df.d <- read_vcf("duplex.SRR1799908.vcf") %>%
  filter(frequency < 0.4) %>%
  mutate(target = chromosome == "chr9" & # target mutation from the paper
           position == "130872141" &
           to == "A",
         true.p.value = -10 * log10(1 - (rank(frequency) - 0.5) / n()))
```

```
## Warning in read_vcf("duplex.SRR1799908.vcf"): NAs introduced by coercion
```

```
ggplot() +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  geom_smooth(data=subset(df.d, !target), aes(true.p.value, qual),
              color="red", fill="grey", alpha=0.5) +
  geom_point(data=subset(df.d, target), aes(true.p.value, qual),
              color="black") +
```

```
  geom_text(data=subset(df.d, target), aes(true.p.value, qual-5,
                                           label = "ABL c.G835A/p.E279K"),
            color="black") +
  scale_x_continuous("-10 log10 P-value", limits = c(0, 100)) +
  scale_y_continuous("Q score", limits = c(0, 100)) +
  theme_bw()
```



HIV sequencing

```
read_vcf_hiv <- function(file_name, sample) {
  .df <- read_vcf(file_name)
  .df$sample <- sample
  .df <- subset(.df, frequency < 0.4)
  .df %>% mutate(true.p.value = -10 * log10(1 - (rank(frequency) - 0.5) / n()))
}

df.h <- data.frame()
df.h <- rbind(df.h, read_vcf_hiv("hiv.SRR1763767.vcf", "Donor plasma"))
```

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

```
df.h <- rbind(df.h, read_vcf_hiv("hiv.SRR1763769.vcf", "8E5 (control)"))
```

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

```
ggplot(df.h, aes(true.p.value, qual, color=sample)) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  geom_point(shape=21) +
  scale_x_continuous("-10 log10 P-value", limits = c(0, 100)) +
  scale_y_continuous("Q score", limits = c(0, 100)) +
  scale_color_brewer("Sample", palette = "Set1") +
  theme_bw()
```