MAGERI benchmark using reference standard DNA library

Mikhail Shugay

December 1, 2016

Load metadata

```
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
```

```
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) %>% filter(type == "blank") # metadat
```

VCF parsing function

```
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("[=;]"), 9)[,c(2, 4, 8)]

  .vcf$coverage <- as.numeric(.infosplit[,1])
  .vcf$frequency <- as.numeric(.infosplit[,2])
  .vcf$cqs <- as.numeric(.infosplit[,3])
  .vcf$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      cqs count
## 1       chr2 212295704    C  A    8     2265 0.0004415011 40.00000     1
## 2       chr2 212295705    C  A    8     2265 0.0004415011 40.00000     1
## 3       chr2 212295713    A  G    7     2265 0.0004415011 40.00000     1
## 4       chr2 212295718    G  A    6     2265 0.0004415011 40.00000     1
## 5       chr2 212295725    C  A    8     2264 0.0004416961 40.00000     1
## 6       chr2 212295732    C  T   24     2264 0.0013250883 39.66667     3
```

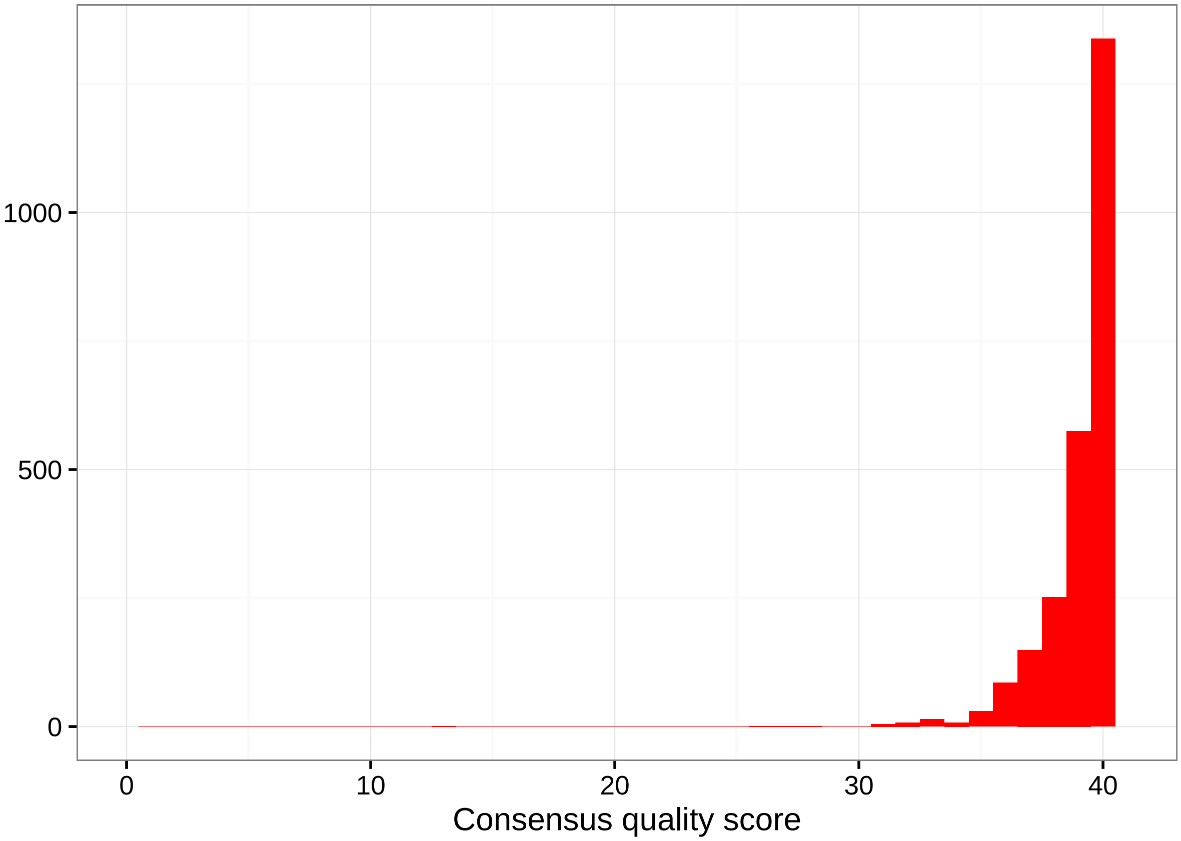
Read samples with HD734 standard DNA and control human DNA, append metadata

```
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 <- primer_set
  .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
```

Histogram of CQS scores

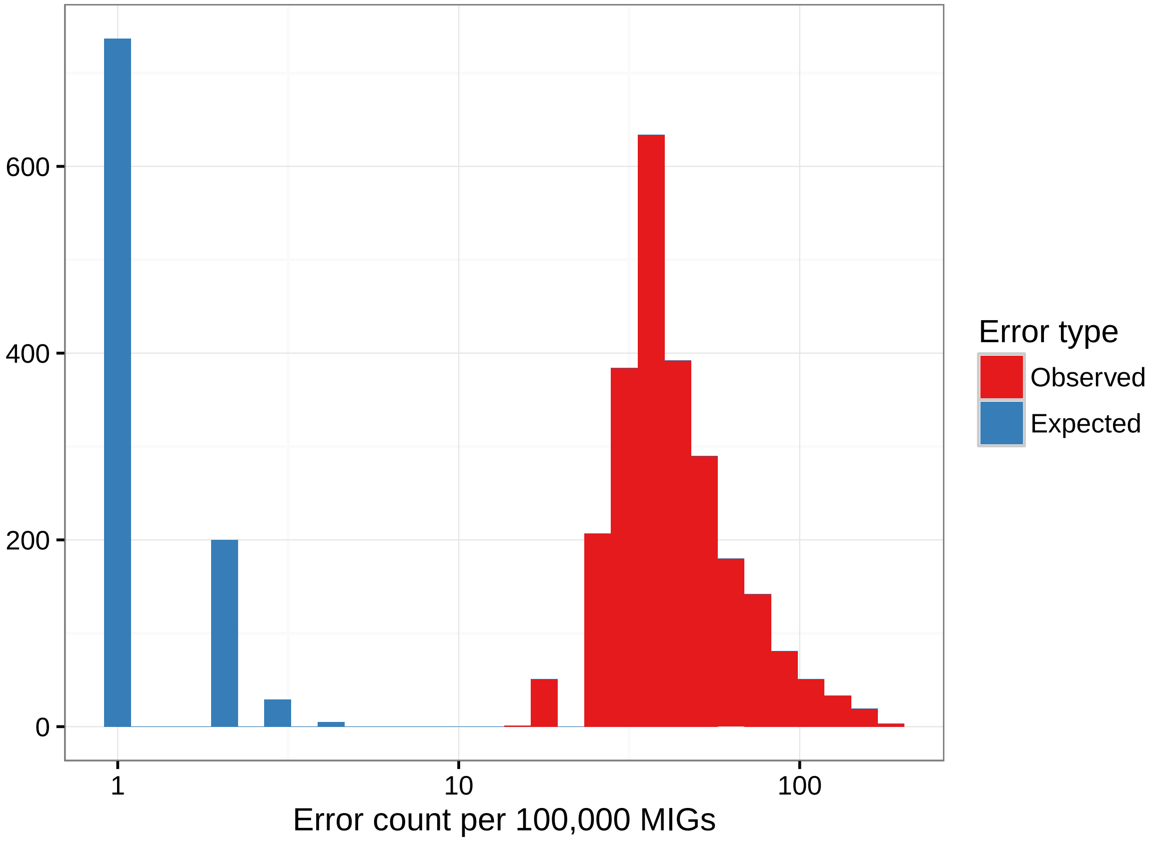
```
fig13 <- ggplot(df, aes(cqs)) +
  geom_histogram(fill="red", binwidth=1) +
  ylab("") + scale_x_continuous("Consensus quality score", limits=c(0, 41)) +
  theme_bw()

save(file = "../figures/fig13.Rda", fig13)

fig13
```



Expected HTS errors and observed error count

```
p_err <- 1-pbinom(3, size = 5, p=1e-2) + 0.5*dbinom(3, size = 5, p=1e-2)
n_migs <- 100000
df.1 <- data.frame(count = df$frequency * n_migs, type = "Observed")
lambda <- p_err * n_migs
df.1 <- rbind(df.1, data.frame(count = rpois(nrow(df), lambda), type="Expected"))

fig14 <- ggplot(df.1, aes(x=count, fill=type)) +
  geom_histogram() +
  ylab("") + scale_x_log10("Error count per 100,000 MIGs") +
  scale_fill_brewer("Error type", palette = "Set1") +
  theme_bw()

save(file = "../figures/fig14.Rda", fig14)

fig14
```

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

## Warning: Removed 1497 rows containing non-finite values (stat_bin).
```

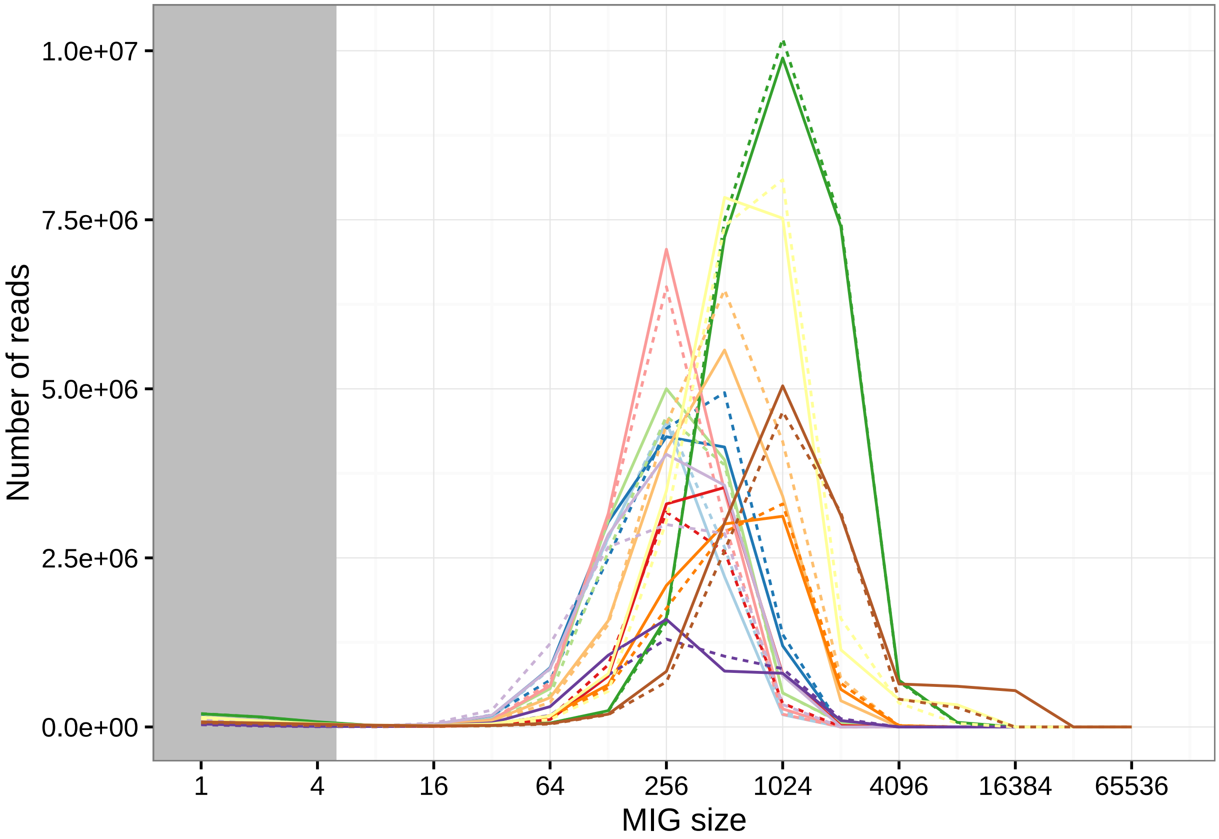


UMI coverage histogram

```
df.smeta.2 <- read.table("sample_metadata.txt", sep="\t", header=T)

df.1 <- data.frame()

for (i in 1:nrow(df.smeta.2)) {
  prefix <- df.smeta.2$prefix[i]
  .df <- read.table(paste0("proc_stat/", prefix, ".umi.histogram.txt"), header = T)
  .df$sample <- with(df.smeta.2, paste(type[i], primer_set[i], ratio[i]))
  .df$replica <- df.smeta.2$replica[i]
  df.1 <- rbind(df.1, .df)
}
```

```
fig12 <- ggplot(df.1, aes(x=mig.size.bin, y = read.count, color=sample, linetype=as.factor(replica))) +
  annotate(geom="rect", xmin=0,xmax=5, fill="grey", ymin=-Inf,ymax=Inf) +
  geom_line() +
  ylab("Number of reads") +
  scale_x_log10("MIG size", breaks = 2^(seq(0, 20, by=2)), limits=c(1,100000)) +
  scale_color_brewer(palette = "Paired", guide=F) +
  scale_linetype_discrete(guide=F)+
  theme_bw()

save(file = "../figures/fig12.Rda", fig12)

fig12
```

```
## Warning: Removed 72 rows containing missing values (geom_path).
```


1.0e+07
7.5e+06
5.0e+06
2.5e+06
0.0e+00
Number of reads
MIG size