CHAPTER 4

Pool-seq uncertainty analysis

Z. Forrest Elkins

Background

Pooled sequencing has become a cost-effective and accurate method of DNA sequencing. As a result, we lose individual genetic data identifiers. The sequenced DNA are a representative sample of the pool instead of the individual. While this method reduces sequencing cost without sacrificing accurate allele & haplotype frequency reads, it does introduce extra sources of statistical error. Since the DNA reads are a representative sample, we wind up with varying amounts of sequenced DNA reads along the genome. In addition, the resulting allele & haplotype frequencies are influenced by the formation of the sample.

Here, I establish a novel method of statistical error estimation due to varying amounts of coverage. Coverage is defined as the number of reads at a given location along the genome. Due to the law of large numbers, a higher sequencing coverage value leads to a more accurate allele estimation at that loci.

Simple pairwise comparisons

Simulate a pairwise comparison between alleles at a single location without error. Here, calculate the absolute allele frequency difference between segregated bulks. The allele frequencies for the ‘high’ and ‘low’ bulks are assumed to be true – i.e., there is no error in allele frequency estimation.

```r
# load dependencies
library(ggplot2)
library(tidyverse)
library(purrr)
library(cowplot)
set.seed(172452)
```

```r
# establish pairwise parameters

## allele frequencies for high and low populations
popH <- 0.3
popL <- 0.09
## absolute difference in allele frequencies
diff <- abs(popH - popL)
## scale data logarithmically
logdiff <- -log10(diff)
## print pairwise comparison value -- 'true' value
logdiff
```

```
[1] 0.6777807
```

Estimating ‘effective coverage’

A simulated calculation of coverage using the math detailed in Tilk et al. 2019. In other words, my own miniature HAF-pipe functions. Our variable `cvg_e` is the ‘theoretical coverage at which binomial sampling of reads would be expected to contain the observed amount of error from estimated frequencies.' Put another way, the estimated and theoretical root mean squared error (RMSE) rates will equal each other under 'effective coverage.'

This method provides a level of sequencing coverage that researchers should aim for in their genetic data to minimize error in their observed allele frequencies. It does not give an estimation of that error as a function of coverage.

```
# modulate pairwise comparison with coverage as a source of error

# n is the number of sites, which in our case is just one
n <- 1

## estimated and true allele frequencies at one site
AFtrue <- rbinom(100,100,0.38) / 100
AFest <- rbinom(100,100,0.43) / 100

## effective coverage -- Tilk et al. 2019
cvg_e <- sum(AFtrue * (1 - AFtrue)) / sum((AFest - AFtrue) ^ 2)
cvg_e
```

```
[1] 30.42092
```

```
## theoretical root mean squared error
RMSEthe <- sqrt( sum(AFtrue * (1 - AFtrue)) / (cvg_e * n) )
RMSEthe
```

```
[1] 0.8772115
```

```
## estimated RMSE
RMSEest <- sqrt( sum((AFest - AFtrue) ^ 2) / n)
RMSEest
```

```
[1] 0.8772115
```

In this case, with the code above, error introduced by variation in coverage would be evaluated via the magnitude of difference between theoretical and estimated RMSE values.

Tilk, Susanne, Alan Bergland, Aaron Goodman, Paul Schmidt, Dmitri Petrov, and Sharon Greenblum. 2019. "Accurate Allele Frequencies from Ultra-Low Coverage Pool-Seq Samples in Evolve-and-Resequence Experiments." *G3* 9 (12): 4159–68.

Coverage and pairwise allele frequency comparisons

The following is a novel method of estimating error as a function of coverage. Simulate a pairwise comparison between alleles at a single location with coverage error. To do this, we use a random binomial distribution that takes in coverage and true allele frequency and outputs the estimated allele frequency.

First, we simulate a single allele frequency estimation given coverage and true allele frequency:

```
# how to simulate coverage
## rbinom(1, cvg, AFtrue) / cvg
## output is estimated AF

cvg <- 20
trueAF <- 0.43

estAF <- rbinom(1,cvg,trueAF) / cvg
estAF
```

```
[1] 0.4
```

Next, we establish a function that calculates estimated allele frequency when it is given coverage and true allele frequency:

```
# function calculating estimated allele frequency
estAF <- function(cvg,trueAF){
  est <- rbinom(1,cvg,trueAF) / cvg
  return(est)
}
```

Finally, we can scale our code and plot the results:

```
# run this n times, scale up
n <- 1000
tru <- 0.43
cvg <- sample(2:150,n,replace = TRUE)
af <- tibble(
  "cvg" = cvg,
  "trueAF" = rep(tru,times=n)
)

af$estAF <- af %>% pmap(estAF) %>% unlist()

# plot estimated allele frequency by coverage
plt <- af %>% ggplot(aes(x = estAF, y = cvg)) +
  geom_point(alpha = 0.15,colour="red") +
  geom_smooth(method = "gam", colour = "red",se=FALSE) +
  geom_vline(xintercept=tru, size = 0.8,linetype = "dashed") +
  labs(x = "Estimated allele frequency", y = "Coverage") +
  annotate("label",x = tru, y = -50, label = "True allele frequency") +
  theme_cowplot()
plt
```

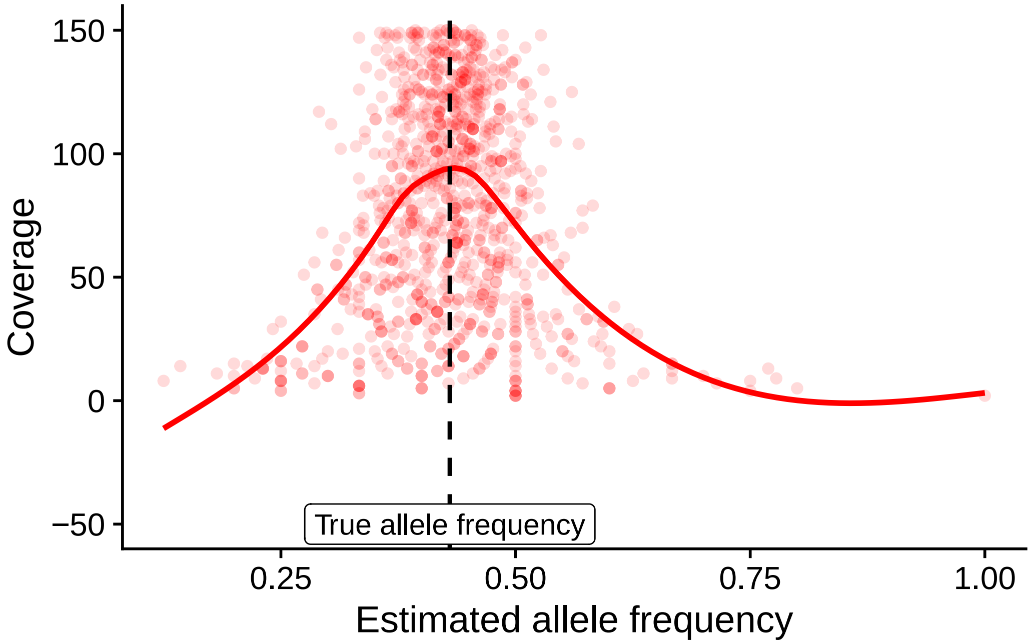

Figure 1: Variability in allele frequency estimations due to coverage.

We can see the level of coverage and its effect on allele frequency estimation in the plot above.

Simple plot of random allele frequencies

Pull allele frequencies from a distribution for ‘low’ and ‘high’ poolseq populations and plot them.

```r
## the following code is from my professional website
expl_freqs <- readRDS('expl-freqs.Rds')
arm <- expl_freqs %>%
  filter(chrom=='2L') %>%
  slice(1:800)

D <- (sample(400:600,800,replace = TRUE))/1000
N <- (sample(300:600,800, replace = TRUE))/1000

sampledata <- tibble(
  "pos" = arm$pos,
  "N" = N,
  "D" = D
)

plt <- ggplot(data=sampledata, aes(pos/16)) +
  geom_line(aes(y = N, colour = "non-explorers")) +
  geom_line(aes(y = D, colour = "explorers"))
plt + labs(x = "Position (Mb)")
```

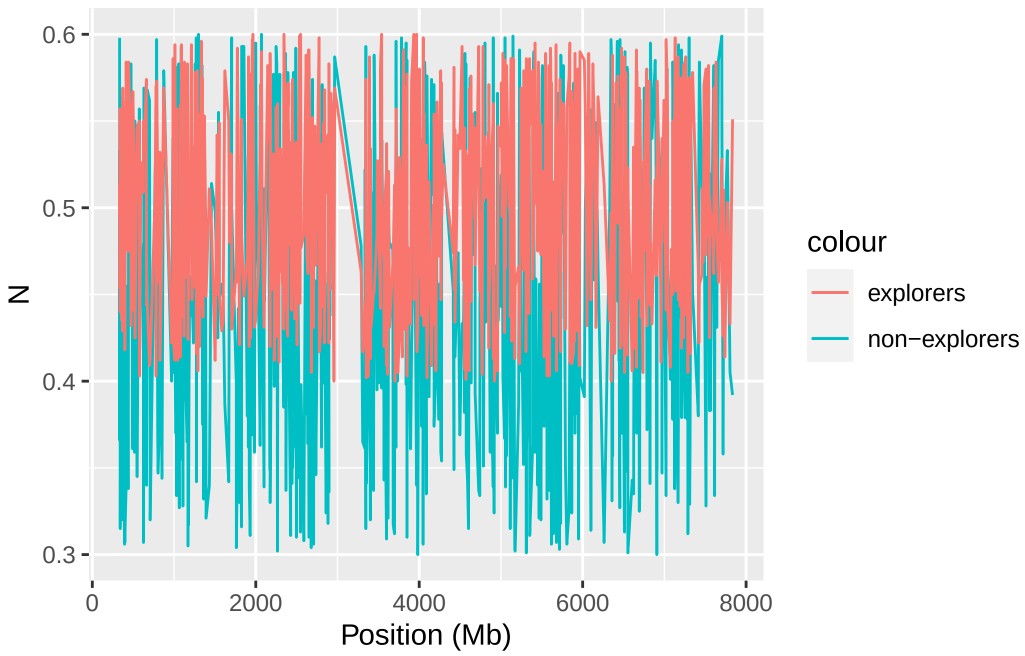

Figure 2: Allele differences between exploring and non-exploring flies.