

Pop Gen HW5

Dylan Sosa

2/18/2020

1

a

With $N_0 = 10,000$

The size of the present population is: 10,000

Size of population before bottleneck: $N_0 * 10 = 100,000$

Bottleneck occurred $0.05 * 2N$ generations ago

b

Compute the cumulative coalescent rate function:

$$\Lambda(t) := \int_{s=0}^t \frac{1}{f(s)} ds$$

while $t \geq 0.05$:

$$\begin{aligned} \Lambda(t) &:= \int_{s=0}^t \frac{1}{f(s)} ds \\ &= \int_{s=0}^{t=0.05} ds + \int_{s=0.05}^t \frac{1}{10} ds \\ &= s|_0^{0.05} + \frac{s}{10} \Big|_{0.05}^t \\ &= 0.05 + \frac{t}{10} - 0.005 \\ &= \frac{t}{10} + 0.045 \end{aligned}$$

while $t \leq 0.05$:

$$\begin{aligned} \Lambda(t) &:= \int_{s=0}^t \frac{1}{f(s)} ds \\ &= \int_{s=0}^t ds \\ &= t \end{aligned}$$

c

Compute the expected value:

$$\mathbb{E}[T_k^{(k)}] = \int_{t=0}^{\infty} \mathbb{P}\{T_k^{(k)} > t\} dt$$

Substitute $e^{-\binom{k}{t}\Lambda(t)}$ for $\mathbb{P}\{T_k^{(k)} > t\}$

Which gives:

$$\begin{aligned}
 &= \int_{t=0}^{\infty} e^{-\binom{k}{t}\Lambda(t)} dt \\
 &= \int_{t=0}^{\infty} e^{-\frac{k(k-1)}{2} * \frac{1}{200} * (20t+9)} dt \\
 &= \int_{t=0}^{\infty} e^{-\frac{k(k-1)}{400} * (20t+9)} dt \\
 &= e^{-\frac{9k(k-1)}{400}} \int_{t=0}^{\infty} e^{-\frac{9k(k-1)}{20} t} dt \\
 &= -e^{-\frac{9k(k-1)}{400}} * \frac{20}{k(k-1)} * [e^{-\frac{20k(k-1)}{400} t}]_0^{\infty} \\
 &= \frac{20}{k(k-1)} * e^{-\frac{9k(k-1)}{400}} \\
 &= \mathbb{E}[T_k^{(k)}] \\
 &\square
 \end{aligned}$$

$$\begin{aligned}
 2) \quad r_{a,c}^2 &= \frac{D_{a,c}^2}{p_a p_b p_c} \\
 p_a &= \frac{n_a}{n} \\
 D_{a,c}^2 &= (p_{a,c} - p_a p_c)^2 \\
 p_{a,c} &= \frac{n_{a,c}}{n}
 \end{aligned}$$

$$\begin{aligned}
 p_a &= \frac{12}{20} = 0.6 & n_{a,c} &= 4/20 = 0.2 \\
 p_b &= \frac{10}{20} = 0.5 & n_{a,b} &= 6/20 = 0.3 \\
 p_c &= \frac{9}{20} = 0.45 & n_{b,c} &= 5/20 = 0.25 \\
 p_a &= \frac{2}{20} = 0.1 & D_{a,c}^2 &= (0.2 - (0.6(0.45)))^2 = 0.0049 \\
 p_b &= \frac{10}{20} = 0.5 & D_{a,b}^2 &= (0.3 - (0.6(0.5)))^2 = 0 \\
 p_c &= \frac{14}{20} = 0.7 & D_{b,c}^2 &= (0.25 - (0.5(0.45)))^2 = 0.00625 \\
 p_a p_b p_c &= 0.06 \\
 p_a p_b p_c &= 0.0594 \\
 p_b p_c p_c &= 0.061875
 \end{aligned}$$

$$r_{a,c}^2 = \frac{0.0049}{0.0594} = 0.0825$$

$$r_{a,b}^2 = \frac{0}{0.06} = 0$$

$$r_{b,c}^2 = \frac{0.00625}{0.061875} \approx 0.01$$

Figure 1: Computing LD

Genealogical Ancestral Process Simulator

```

set.seed(420)
# for rexp()

genealogicalAncestralProcessSimulator <- function(L = 200, rho = 0.002, n = 10, trajectories = 1000, fa
  for(i in seq(trajectories)){ # for each i simulation in trajectories
    k <- n # init
    genealogical.df <- data.frame(matrix(ncol = 1, nrow = 1e5))
    # each simulation is a column, value is lineages of that simulation at time t
    # x is row, y is column
    # k is # lineages

    genealogical.df$simLineages[1] <- k # init k in df
    genealogical.df$t[1] <- 0 # init time in df
    genealogical.df <- select(genealogical.df, t, simLineages)
    j <- 2
    # init where we start filling rows == row 2 bc row 1 has init values for the simulation

    oldTime <- 0
    # print(i) testing

    while( k != 1 ){
      # print('in while') testing
      increaseLineageProb <- k * (rho/2) * (L-1) / ((k * (rho/2) * (L-1) + choose(k,2)))
      decreaseLineageProb <- choose(k,2) / ((k * (rho/2) * (L-1) + choose(k,2)))

      move <- sample( c(-1,1), 1, prob = c(decreaseLineageProb, increaseLineageProb))
      # make probability weight vector

      k <- genealogical.df$simLineages[j] <- k + move
      # update number of lineages at time i, use sample from distribution of probs of lineage increasin
      # this value is the new k as well as row value for this lineage column
      # prob of decreasing is typically higher

      rate = k * ((rho/2) * (L-1)) + choose(k,2)
      # rate updates each loop with new k at each t

      time <- rexp(1, rate)
      # rate of time, exponentially distributed
      # ATTN: this is how t increases (very slowly), so simulations typically do not reach t > 20 befor
      # This is based on my understanding of the problem,
      # I believe if t increased more rapidly then there would be fewer simulations that reach k == 1

      newTime <- time + oldTime
      genealogical.df$t[j] <- newTime
      # update which row we are on, update time value with current time

      oldTime <- newTime
      j <- j+1 # j is simply the row containing our time t, is not the t == 20 we need to check for
    }
  }

```

```

failedSimulation <- filter(genealogical.df, simLineages, simLineages == 1)
# once k == 1, check what time the simulated pop went to 1 at

if(failedSimulation$t[1] < 20) { # critical to
  failedTrajectories <- failedTrajectories + 1
}
}
cat(failedTrajectories, 'trajectories reach k == 1 before t == 20')
}

genealogicalAncestralProcessSimulator()

```

966 trajectories reach k == 1 before t == 20

4

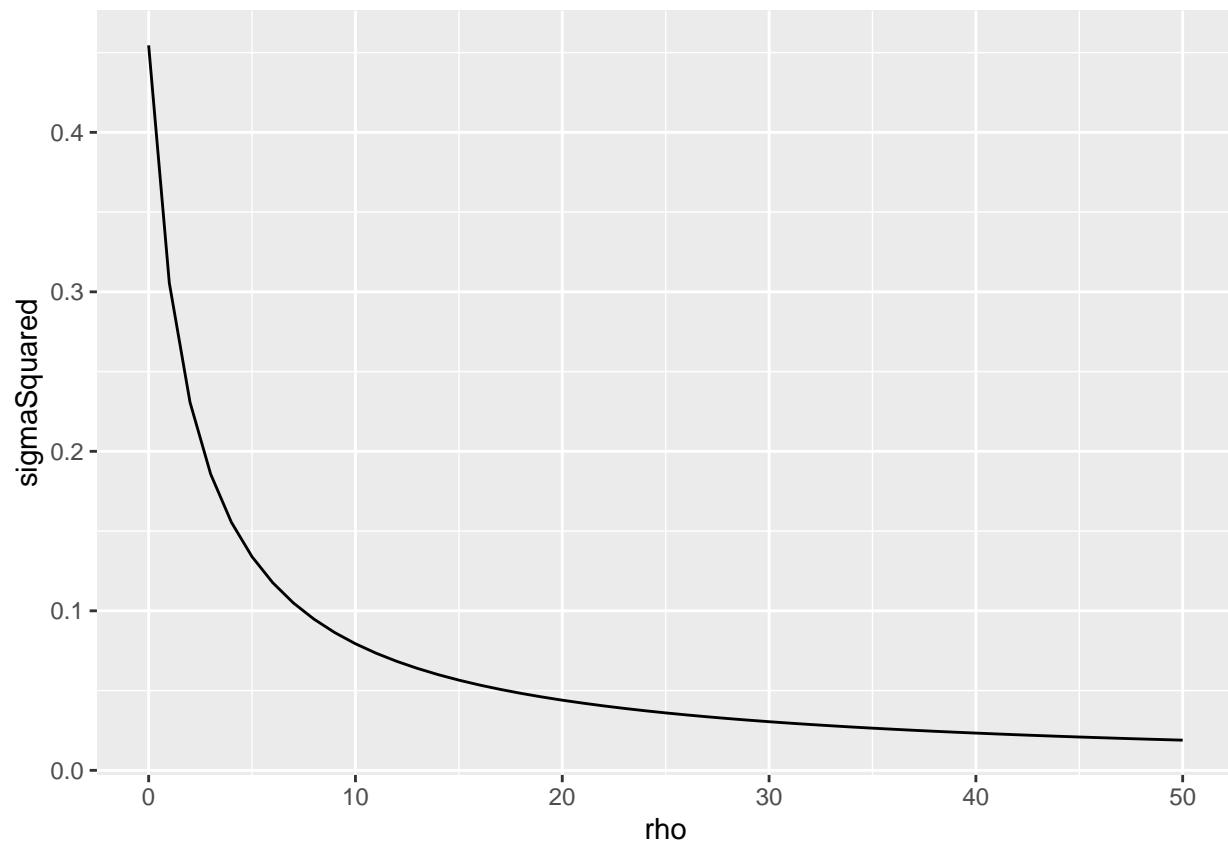
a

```

rho <- seq(0,50)
sigmaSquared <- (rho + 10)/(rho**2 + 13*rho + 22)

ggplot() +
  geom_line(aes(rho, sigmaSquared))

```



```
rho0 <- sigmaSquared[1]
cat('sigmaSquared for rho = 0 is:',rho0)
```

```
## sigmaSquared for rho = 0 is: 0.4545455
```

```
cat('\n5% of rho at 0 is:',rho0*0.05)
```

```
##
## 5% of rho at 0 is: 0.02272727
```

b

Calculate the (positive) rho for which $\text{sigmaSquared} == 5\%$ of its value at $\text{rho} == 0$

Setting $\sigma_d^2 == \text{rho0}$, solving for ρ

$$0.02272727 = \frac{\rho + 10}{\rho^2 + 13\rho + 22}$$

$$0.02272727\rho^2 + 0.29545451\rho + 0.49999994 = \rho + 10$$

Solve for ρ , obtain quadratic equation

$$0.02272727\rho^2 - 0.7045455\rho - 9.50000006 = 0$$

Solve quadratic equation:

```
a <- 0.02272727
b <- -0.7045455
c <- -9.50000006

root <- ((-b) + sqrt(b**2 - (4*(a*c))))/(2*a)

cat('Positive rho value for which sigma**2 == 5% of its value at rho == 0:', root)
```

```
## Positive rho value for which sigma**2 == 5% of its value at rho == 0: 41.15639
```

c

$$\rho = 4N_e r$$

$$N_e = 10,000$$

$$r = 1.25 * 10^{-8}$$

Base-pairs that correspond to the ρ obtained from solving the quadratic equation:

$$bp = \frac{\rho}{4N_e r}$$

```
bp = root/(4*1e4*1.25*(10**-8))  
cat('This rho corresponds to:',bp,'bp.')
```

```
## This rho corresponds to: 82312.78 bp.
```