Approximate Bayesian Computation exercise

Gaurav Kandlikar

November 23, 2015

Estimation of population size

Scenario: Imagine that you resequenced a 100kb region of two Y chromosomes, which are non-recombining. You observe 50 SNPs within this region. You would like to use this dataset to estimate the effective population size of males. You know that the mutation rate per base pair on the Y chromosome is $\mu = 1e^{-8}$ per generation.

Let's develop an ABC approach to this.

Since I have to redo this a few times over this assignment, I wrote a function to generate a SNP distribution given a distribution of Ns:

```
generate_snp_counts <- function(reps = 1e6, min_n = 100, max_n = 100000, mu = 1e-8,
                                bps = 100000, autosome = F, observed_snps = 50,
                                num_independent_sites = 1) {

  # Part A: Draw Ns from distribution --------
  N <- runif(reps, min_n, max_n)

  # Part B: For each value of N, simulate a TMRCA.
  if(!autosome) {
    rates <- 1/N
  } else { # If an autosome, we are working on estimating 2N- think about this more
    rates <- 1/(2*N*num_independent_sites)
  }
  coal_times <- rexp(reps, rates)
  net_mutation_rates <- 2*mu*bps*coal_times

  # Part C: For each TMRCA, add a Poisson distributed number of mutations
  num_snps <- rpois(reps, net_mutation_rates)
  lineages <- cbind(N, coal_times, num_snps)

  # Part D: d) We now need to decide which of the million draws from the priors to accept
             # i.e. which are "close enough". Here, close enough is defined as 45 - 55
  accept <- subset(lineages, lineages[,3] >= 0.9*observed_snps & lineages[,3] <= 1.1*observed_snps)

  # Return
  return(list("all" = data.frame(lineages), "accepted" = data.frame(accept)))
}
```

Question 1

a) Assume that N can be any value between 100 and 100000. Draw 1 million values from the prior distribution of N, assuming N $[100, 100000]$

```
reps <- 1e6
min <- 100; max <- 100000
mu <- 1e-8
bps <- 100000

q1_full <- generate_snp_counts(reps = reps, min_n = min, max_n = max, mu = mu, bps = bps,
                               autosome = F, observed_snps = 50)
q1_prior <- q1_full$all
q1_posterior <- q1_full$accepted
```

Now that we've run this, let's do a sanity check to make sure everything is in order here.

```
# Make sure the prior was set up well ---------
# N should be distributed between 100 and 100000
min(q1_prior$N); max(q1_prior$N)
```

```
## [1] 100.2244
```

```
## [1] 99999.78
```

```
# mean of uniform = (min+max)/2
mean(q1_prior$N); (min+max)/2
```

```
## [1] 50016.45
```

```
## [1] 50050
```

```
# Mean of coal_time should be mean of N- not sure this is also expected SD since it's
# sampling from a range
mean(q1_prior$coal_times); sd(q1_prior$coal_times)
```

```
## [1] 50022.27
```

```
## [1] 64557.81
```

```
# Mean number of SNPs should be equal to 2*N*mu*bps
mean(q1_prior$num_snps); mean(2*mu*q1_prior$N*bps)
```

```
## [1] 100.0437
```

```
## [1] 100.0329
```

```r
# Make sure that we accepted the right things -------
# Should be between 45 and 55-
min(q1_posterior$num_snps); max(q1_posterior$num_snps)
```

```
## [1] 45
```

```
## [1] 55
```

```r
# Percent of rows we accepted-
nrow(q1_posterior)/nrow(q1_prior)
```

```
## [1] 0.057211
```

e) Congratulations! You should be all done!.
Thanks!

Question 2

Make a density plot of the prior and the posterior distributions.

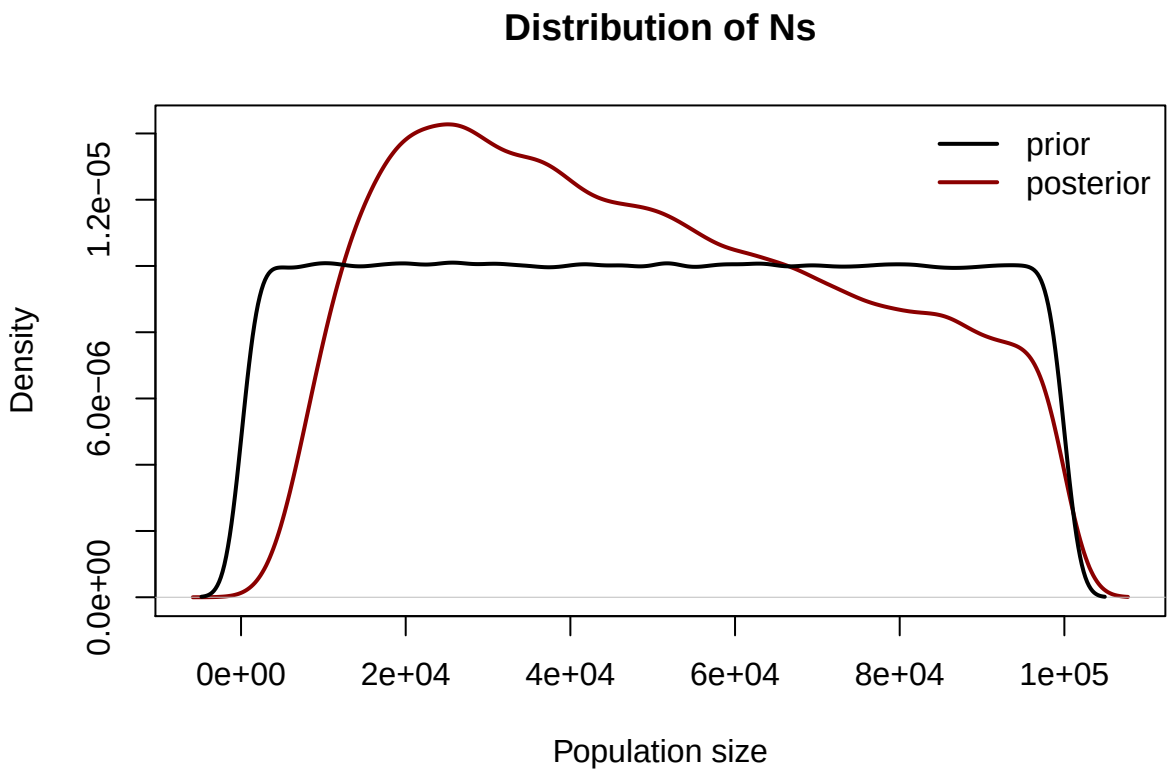
I will write a function for this too:

```r
combined_density_plot <- function(xx) {
  all <- xx$all
  accepts <- xx$accepted

  plot(density(accepts[,"N"]), main = "Distribution of Ns", xlab = "Population size",
       col = "darkred", lwd = 2)
  lines(density(all[,"N"]), lwd = 2)
  legend("topright", lty = 1, col = c("black", "darkred"), legend = c("prior", "posterior"),
       bty = "n", lwd = 2)
}
```

Now, use the function above to generate the plot for q1:

```r
combined_density_plot(q1_full)
```



Question 3 and 4

3) What is the median value of the posterior distribution of N? 4) Generate a 95 credible interval for the posterior distribution of N

```
# Metrics on accepted Ns
mean(q1_posterior$N)
```

```
## [1] 49512.25
```

```
median(q1_posterior$N)
```

```
## [1] 46902.56
```

```
quantile(q1_posterior$N, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 10049.98 96719.73
```

Question 5

How does the posterior distribution differ from the prior distribution?

The posterior distribution of N is slightly narrower than the uniform prior from which Ns were originally drawn. The posterior is centered around the median above with a 95% credible interval from ~9800 to ~97000. On the right side of the distribution, the posterior has not narrowed our estimate very much, but we can be fairly confident that the true N is not <9000.

Question 6

Repeat the ABC analysis with N $[1000, 100000]$

```
reps <- 1e6
min <- 1000; max <- 100000
mu <- 1e-8
bps <- 100000

q6_full <- generate_snp_counts(reps = reps, min_n = min, max_n = max, mu = mu, bps = bps,
                               autosome = F, observed_snps = 50)

q6_prior <- q6_full$all
q6_posterior <- q6_full$accepted
```

Now that things are generated, let's do another sanity check:

```
# Make sure the prior was set up well ---------
# N should be distributed between 1000 and 100000
min(q6_prior$N); max(q6_prior$N)
```

```
## [1] 1000.226
```

```
## [1] 99999.94
```

```
# mean of uniform = (min+max)/2
mean(q6_prior$N); (min+max)/2
```

```
## [1] 50522.03
```

```
## [1] 50500
```

```
# Mean of coal_time should be mean of N- not sure this is also expected SD since it's sampling
# from a range
mean(q6_prior$coal_times); sd(q6_prior$coal_times)
```

```
## [1] 50439.71
```

```
## [1] 64658.4
```

```
# Mean number of SNPs should be equal to 2*N*mu*bps
mean(q6_prior$num_snps); mean(2*mu*q6_prior$N*bps)
```

```
## [1] 100.8783
```

```
## [1] 101.0441
```

```
# Make sure that we accepted the right things -------
# Should be between 45 and 55-
min(q6_posterior$num_snps); max(q6_posterior$num_snps)
```

```
## [1] 45
```

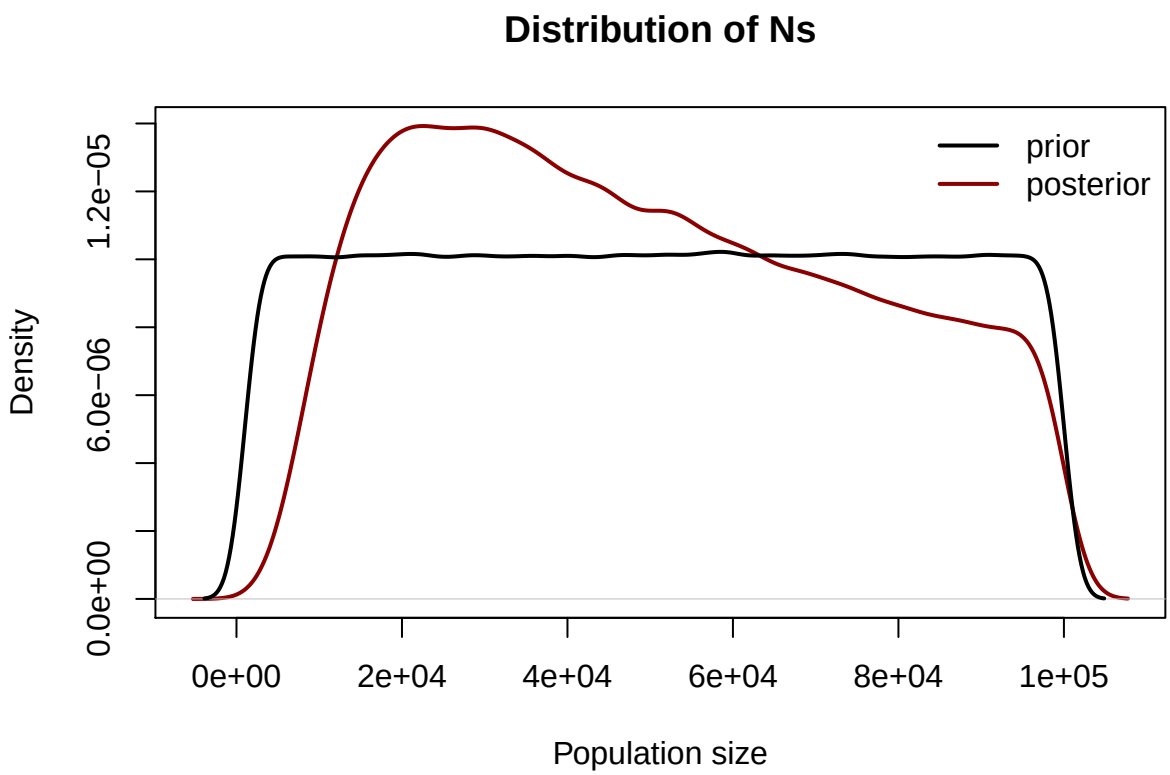
```
## [1] 55
```

```
# Percent of rows we accepted-
nrow(q6_posterior)/nrow(q6_prior)
```

```
## [1] 0.05771
```

OK, all looks in order. Let's make the combined density plot:

```
combined_density_plot(q6_full)
```



The plot does not look very different. We can run the same metrics on the posterior distribution of N:

```r
# Metrics on accepted Ns
mean(q6_posterior$N)
```

```
## [1] 49389.85
```

```r
median(q6_posterior$N)
```

```
## [1] 46661.74
```

```r
quantile(q6_posterior$N, c(0.025, 0.975))
```

```
##      2.5%     97.5%
##  9914.559 96717.950
```

How does the prior differ from the values computed in Q3-4? What does this tell you about the effect of the prior distribution in Bayesian statistics?

The mean, median, and quantiles using a the prior N $[1000, 100000]$ did not differ very much than when a prior of N $[100, 100000]$ was used- this suggests that the Bayesian approach is robust to imperfect priors. This makes sense as long as the prior is greater than the true value (i.e what the posterior will estimate)- it might be fun to redo this with a prior of N $[50000, 100000]$. I've done that at the end of this document.

Question 7

If you wanted a more precise estimate of the population size, would you be better off A) sequencing a bigger region of the Y chromosome, or B) sequencing the same amount on the autosomes? Why?

To get a more precise estimate, I would be better off sequencing the same amount on autosomes. We know that autosomes undergo recombination- in fact, the same pair of chromosomes can recombine multiple times. If we take this to an extreme, we could model each locus on an autosome as being inherited independently of the other- while on the Y chromosome, we know that there is no recombination, so when we sequence more, we are in fact just sequencing more of the same single geneology. My sense is that sequencing autosomes would give us a better estimate of the true number of SNPs we see- here, we just have the one number (50 SNPs), but if we sequence unlinked regions from autosomes, we would expect to see a distribution of SNP count on each locus. This might allow us to narrow our "acceptance" interval to derive a posterior distribution.

Estimates of population split times

Question 1

Once the two chromosomes make it back into the ancestral population of size 100000, what is the expected amount of additional time we have to wait until they coalesce?

Since we are dealing with the Y chromosome, of which there is only 1 copy in diploid men, the expected coalescent time is equal to N- here, $E[T_{before\ split}] = 100000$.

Question 2

What is the expected time until the two chromosomes coalesce with each other?

The net expected time to coalescence is $E[T_{coal}] = T_{split} + N$.

Question 3

Implement the following ABC approach to estimate T_{split}.

a) Assume that t_{split} can be any value between 50000 and 1000000 generations. Draw 1 million values from the prior distribution of t_{split} $U[50000, 1000000]$

```
reps <- 1e6
min <- 50000
max <- 1000000
t_splits <- runif(reps, min, max)
```

b) For each value of t_{split} simulate a TMRCA.

```
N_anc <- 100000
rate <- 1/(N_anc) # Only 1 Y chromosome in males...
coal_times_pre_split <- rexp(reps, rate)

length(coal_times_pre_split) == length(t_splits)
```

```
## [1] TRUE
```

```
net_coal_times <- coal_times_pre_split + t_splits
```

c) For each TMRCA, add a Poisson number of mutations with the right mutation rate

```
mu <- 1e-8
bps <- 100000
net_mutation_rates <- 2*mu*net_coal_times*bps
num_snps <- rpois(reps, net_mutation_rates)

q2_3_lineages <- data.frame(cbind(t_splits, net_coal_times, num_snps))

# Sanity check: confirm that all net_coal_times - t_splits == coal_times_pre_split
ttt <- round(q2_3_lineages$net_coal_times-q2_3_lineages$t_splits)
all(ttt == round(coal_times_pre_split))
```

```
## [1] TRUE
```

```
rm(ttt)
```

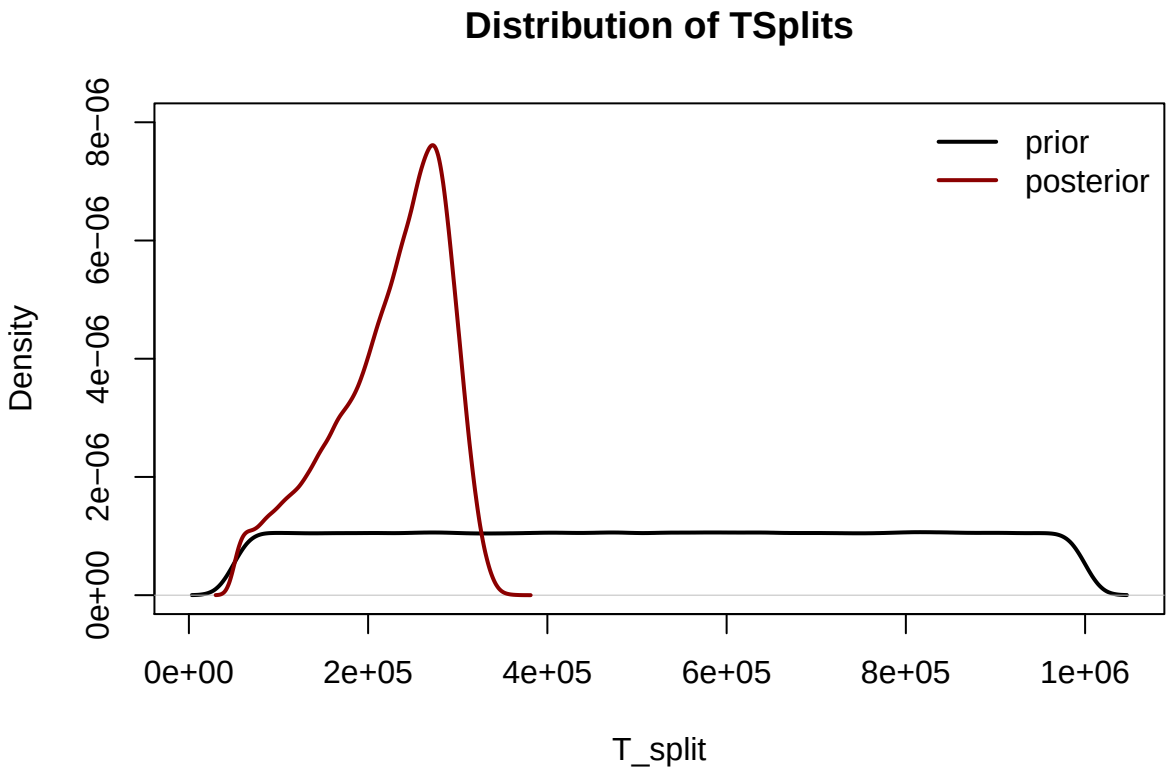
d) We need to decide which of the million draws of the prior to keep for the posterior. Assume that SNP counts between 550 and 650 are acceptable for now.

```
# Subset to q2_3_lineages that have between 550 and 650 SNPs
q2_3_accepts <- subset(q2_3_lineages, q2_3_lineages$num_snps > 549 & q2_3_lineages$num_snps < 651)
```

Question 4

Make a density plot of the prior and posterior distribution of T_{split}

```
plot(density(q2_3_lineages$t_splits), ylim = c(0, 0.000008), lwd = 2, xlab = "T_split",
     main = "Distribution of TSplits")
lines(density(q2_3_accepts$t_splits), lwd = 2, col = "darkred")
legend("topright", lty = 1, col = c("black", "darkred"), legend = c("prior", "posterior"),
       bty = "n", lwd = 2)
```



Question 5

What is the median value of the posterior distribution of T_{split}?

```
mean(q2_3_accepts$t_splits)
```

```
## [1] 223314.6
```

```
median(q2_3_accepts$t_splits)
```

```
## [1] 237610
```

Question 6

Generate a 95 CI for the posterior distribution of T_{split}

```
quantile(q2_3_accepts[,1], probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
##  74178.4 315363.0
```

Question 7

How does the posterior differ from the prior distribution of T_{split}
The posterior here suggests that the true T_{split} falls between ~75,000 to ~316,000, which is a narrower range than the prior distribution from 50,000-1,000,000.

Question 8

What if we did not know the true value of N? Repeat the ABC above, but draw N from $N\ U[1000, 1000000]$.

```
reps <- 1e6
min <- 50000
max <- 1000000
t_splits <- runif(reps, min, max)

N_anc <- runif(reps, min = 1000, max = 1000000) # This is different from above
rates <- 1/(N_anc) # Only 1 Y chromosome in males...
coal_times_pre_split <- rexp(reps, rates)

length(coal_times_pre_split) == length(t_splits)
```

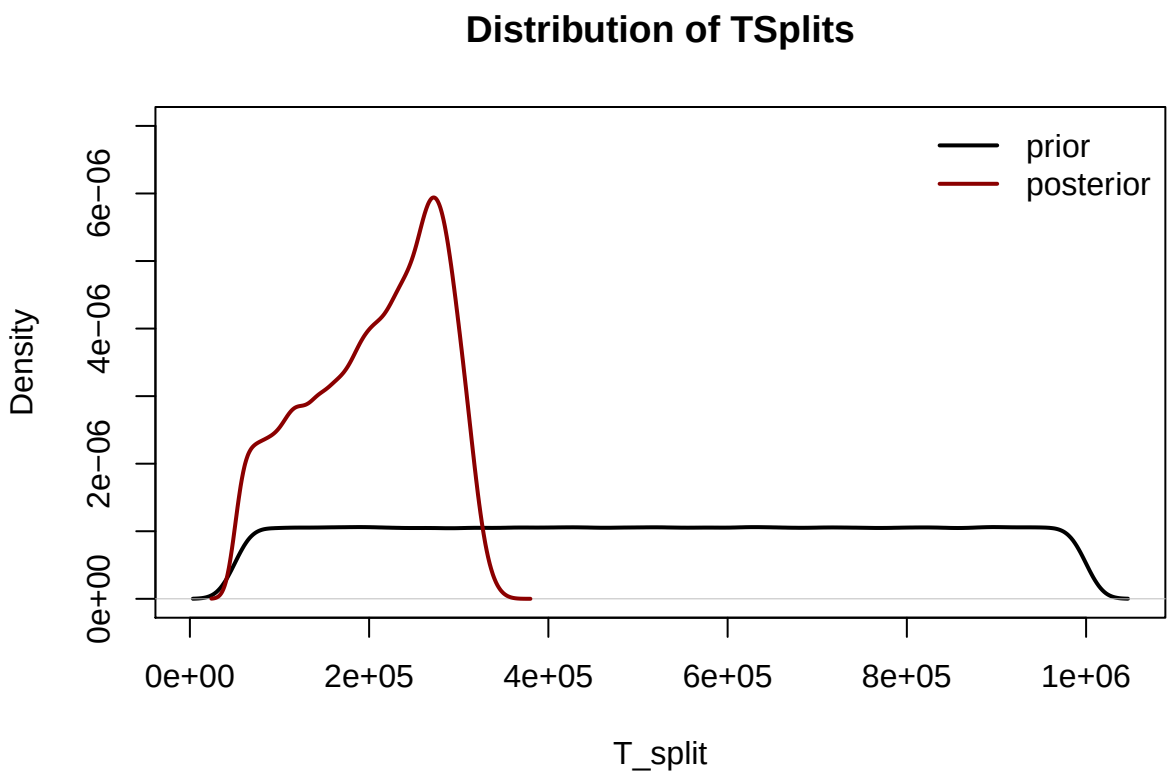
```
## [1] TRUE
```

```
coal_times <- coal_times_pre_split + t_splits

mu <- 1e-8
bps <- 100000
net_mutation_rates <- 2*mu*coal_times*bps
num_snps <- rpois(reps, net_mutation_rates)

q2_8_lineages <- data.frame(cbind(t_splits, coal_times, num_snps))
q2_8_accepts <- subset(q2_8_lineages, q2_8_lineages$num_snps > 549 & q2_8_lineages$num_snps < 651)

plot(density(q2_8_lineages$t_splits), ylim = c(0, 0.000007), lwd = 2, xlab = "T_split",
     main = "Distribution of TSplits")
lines(density(q2_8_accepts$t_splits), lwd = 2, col = "darkred")
legend("topright", lty = 1, col = c("black", "darkred"), legend = c("prior", "posterior"),
       bty = "n", lwd = 2)
```



Question 9

Do your estimates for the median and credible interval differ from above?

```
print("mean of TSplit when N known"); mean(q2_3_accepts$t_splits)
print("median of TSplit when N known"); median(q2_3_accepts$t_splits)
print("95CI of TSplit when N known"); quantile(q2_3_accepts$t_splits, probs = c(0.025, 0.975))
```

```
## [1] "mean of TSplit when N known"
## [1] 223314.6
## [1] "median of TSplit when N known"
## [1] 237610
## [1] "95CI of TSplit when N known"
##     2.5%    97.5%
##  74178.4 315363.0
```

```
print("mean of TSplit when N unknown"); mean(q2_8_accepts$t_splits)
print("median of TSplit when N unknown"); median(q2_8_accepts$t_splits)
print("95CI of TSplit when N unknown"); quantile(q2_8_accepts$t_splits, probs = c(0.025, 0.975))
```

```
## [1] "mean of TSplit when N unknown"
## [1] 205615.5
## [1] "median of TSplit when N unknown"
## [1] 217131.5
## [1] "95CI of TSplit when N unknown"
##      2.5%     97.5%
##  62125.34 314653.10
```

The median and mean estimates of T_{split} are shifted slightly when $N_{ancestral}$ is allowed to vary, but the estimates are of the same order of magnitude (the estimates if $N_{ancestral}$ is unknown tend to be ~10% lower than when $N_{ancestral} = 100000$. The same applies to the confidence interval, which is shifted slightly to the left in the second scenario.) Based on this small difference, and because we are unlikely to have full confidence in a single ancestal population size, applying this sort of framework to allow $N_{ancestral}$ to vary is a great idea. If we are reasonable confident that $N_{ancestral}$ is in fact 100000, we can still generate a distribution with that value as the mean, and try to estimate T_{split} from there.

Question 10

Did you have fun with the YM[R]CA?

Back to Part 1

Just doing this for my own curiosity– no need to grade.

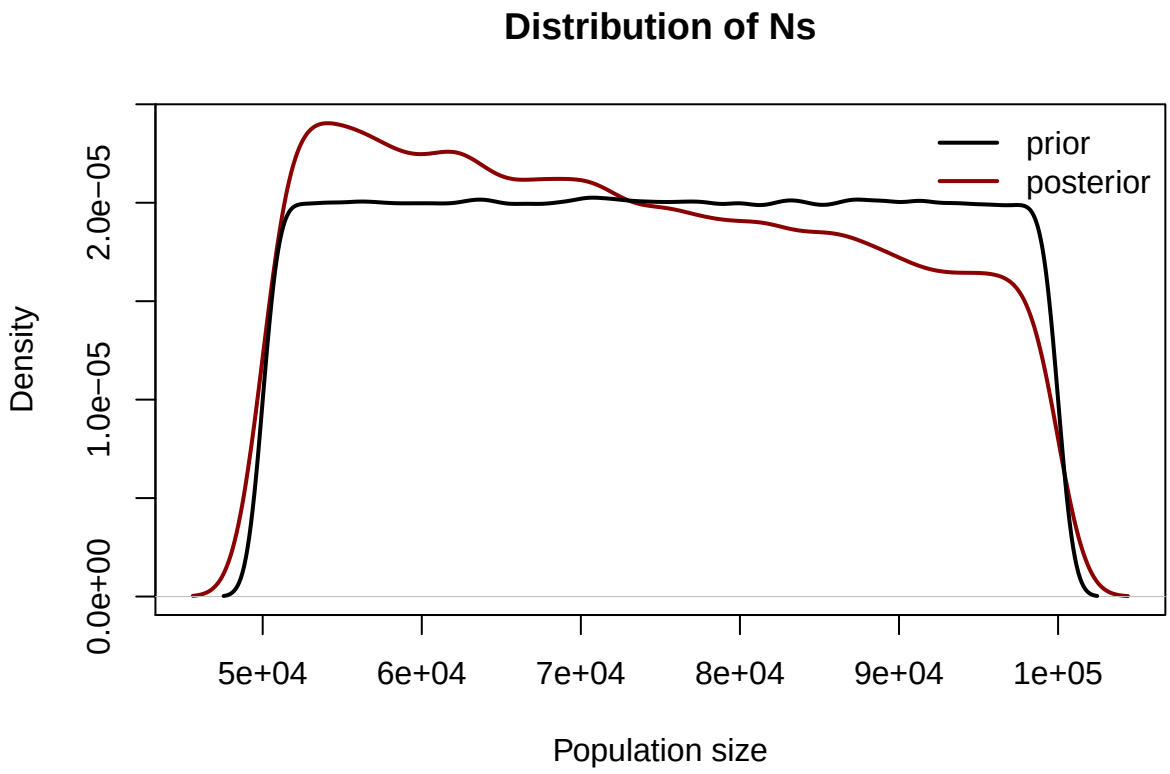
Investigating the consequences of an awful prior

```
# Draw Ns from distribution --------
reps <- 1e6
min <- 50000; max <- 100000
mu <- 1e-8
bps <- 100000

awful_full <- generate_snp_counts(reps = reps, min_n = min, max_n = max, mu = mu, bps = bps,
                                  autosome = F, observed_snps = 50)

awful_prior <- awful_full$all
awful_posterior <- awful_full$accepted

combined_density_plot(awful_full)
```



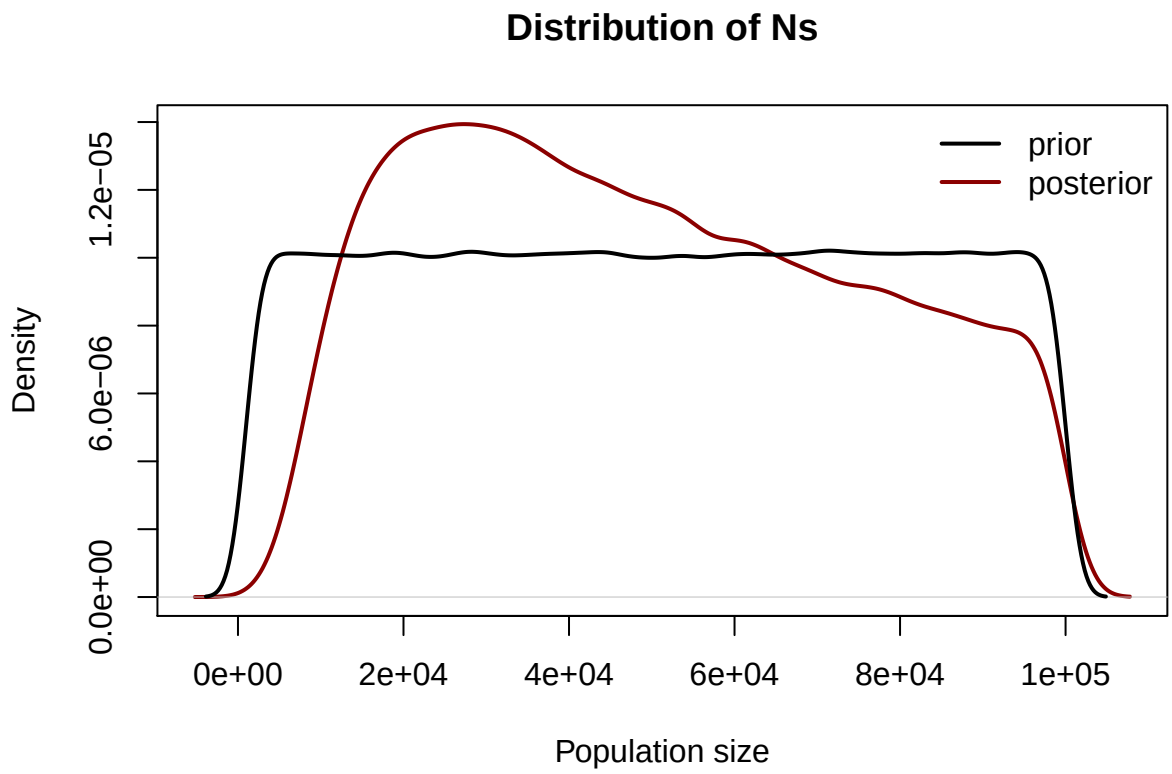
More on Y or same on autosomes?

```
reps <- 1e6
min <- 1000; max <- 100000
mu <- 1e-8
bps <- 200000

morey_full <- generate_snp_counts(reps = reps, min_n = min, max_n = max, mu = mu, bps = bps,
                                  autosome = F, observed_snps = 100)

morey_prior <- morey_full$all
morey_posterior <- morey_full$accepted

combined_density_plot(morey_full)
```

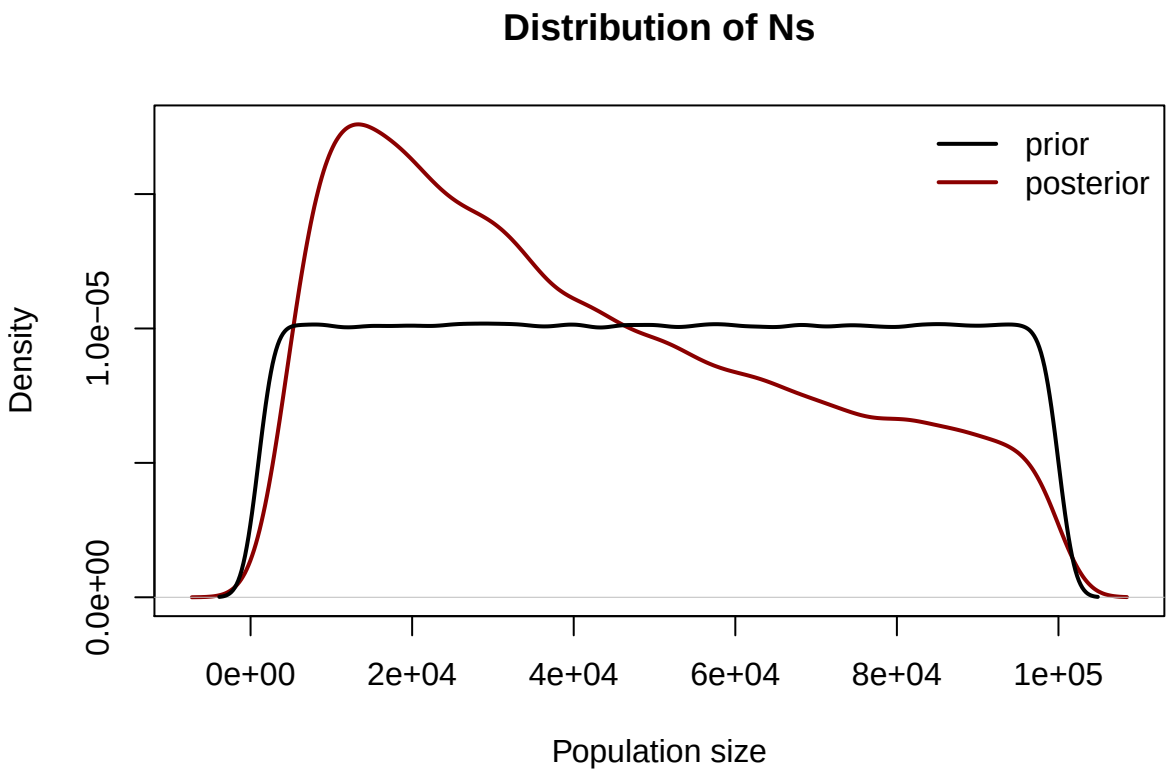



```
reps <- 1e6
min <- 1000; max <- 100000
mu <- 1e-8
bps <- 100000

auto_full <- generate_snp_counts(reps = reps, min_n = min, max_n = max, mu = mu, bps = bps,
                                 autosome = T, observed_snps = 100, num_independent_sites = 2)

auto_prior <- auto_full$all
auto_posterior <- auto_full$accepted

combined_density_plot(auto_full)
```




```
quantile(auto_posterior$N, probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
##  5744.487 95356.145
```

```
quantile(morey_posterior$N, probs = c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 10090.74 96875.36
```

Since there is no recombination modeled here, we don't seem much going on.