GGG-201D-problem-set-2

Ethan Holleman

4/28/2021

Problem 1

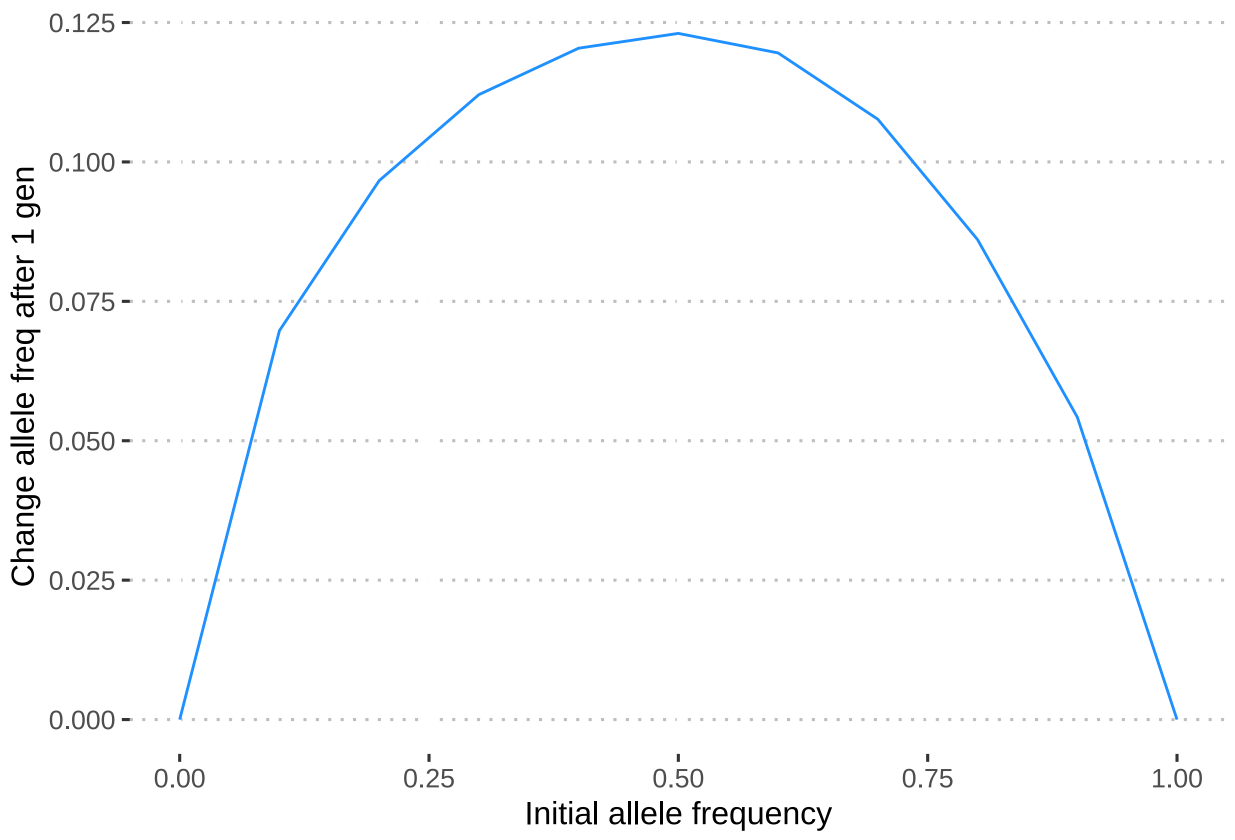
Part A

Use a program such as R or Excel to generate a scatter plot that shows how expected allele frequency change from genetic drift depends on initial allele frequency. The x-axis should be initial allele frequency and range from 0 to 1. The y-axis should be expected change in allele frequency after one generation. Perform calculations in steps of 0.1 for a population size of 2N=20.

```r
expected <- function(init_allele_freq, pop_size, delta_allele_freq){

  num_delta_alleles <- pop_size * delta_allele_freq
  num_init_alleles <- pop_size * init_allele_freq
  routes <- num_init_alleles + c(-1*num_delta_alleles, num_delta_alleles)
  routes <- unique(routes[routes >= 0])  # remove negative values since not possible
  sum(mapply(dbinom, routes, pop_size, init_allele_freq))

}
```

```r
expected_change <- function(init_allele_freq, pop_size){

  magnitudes <- seq(0, 1-init_allele_freq, 0.1)
  sum(mapply(expected, init_allele_freq, pop_size, magnitudes) * magnitudes)

}
```

```r
library(ggplot2)
library(ggpubr)

df.a <- data.frame(init_allele_freq=seq(0, 1, 0.1), pop_size=10)
df.a$delta_allele_freq <- mapply(expected_change, df.a$init_allele_freq, df.a$pop_size)
ggplot(df.a, aes(x=init_allele_freq, y=delta_allele_freq)) + geom_line(color='dodgerblue') +
      theme_pubclean() + labs(x='Initial allele frequency', y='Change allele freq after 1 gen')
```

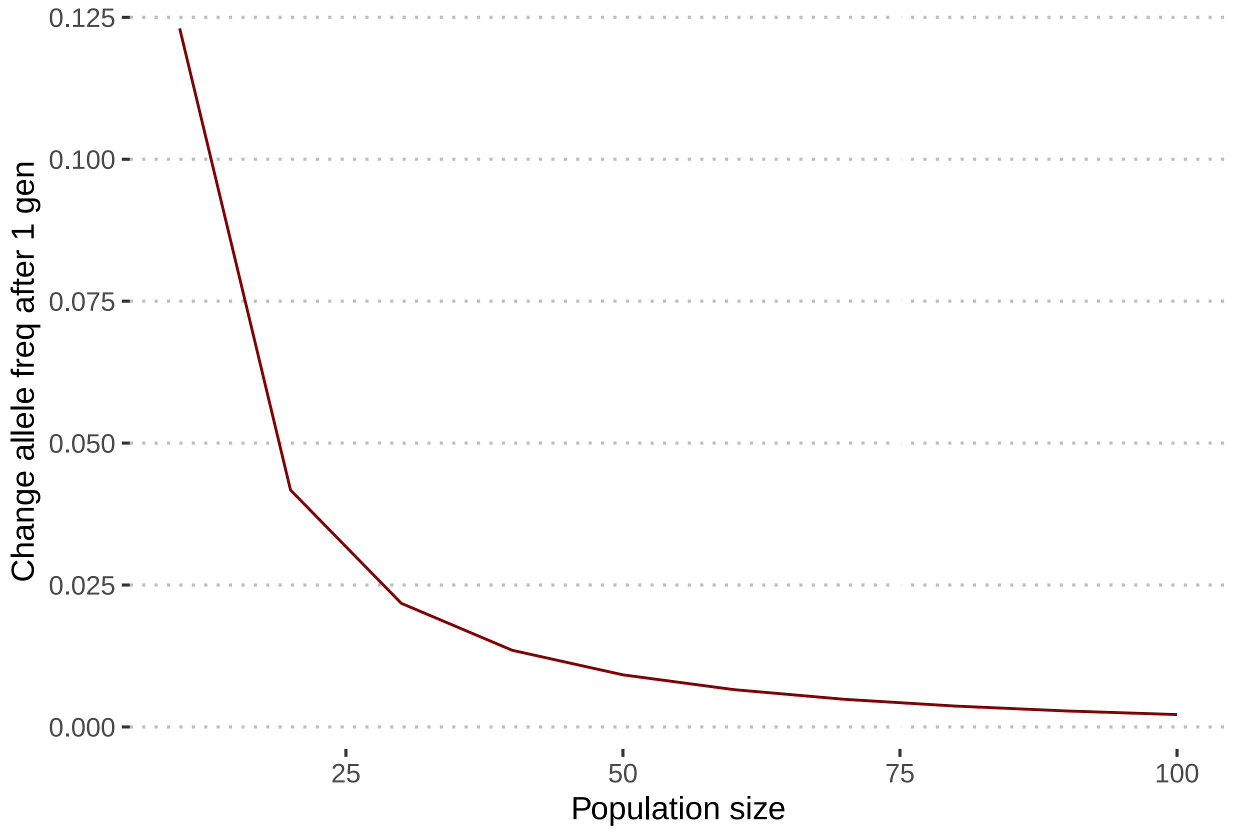


Initial allele frequency	Pop size	Delta allele freq 1 gen
0.0	10	0.0000000
0.1	10	0.0697357
0.2	10	0.0966368
0.3	10	0.1120677
0.4	10	0.1203949
0.5	10	0.1230469
0.6	10	0.1195455
0.7	10	0.1076569
0.8	10	0.0860671
0.9	10	0.0542389
1.0	10	0.0000000

Part B

Use the same program to generate a scatter plot that shows how expected allele frequency change from genetic drift depends on population size. The x-axis should be population size and range from 2N=10 to 2N=100. The y-axis should be expected change in allele frequency after one generation. Perform calculations in steps of 10 with an allele frequency of 0.5

```r
df.b <- data.frame(init_allele_freq=0.5, pop_size=seq(10, 100, 10))
df.b$delta_allele_freq <- mapply(expected_change, df.b$init_allele_freq, df.b$pop_size)
ggplot(df.b, aes(x=pop_size, y=delta_allele_freq)) + geom_line(color='darkred') +
      theme_pubclean() + labs(x='Population size', y='Change allele freq after 1 gen')
```



Initial allele frequency	Pop size	Delta allele freq 1 gen
0.5	10	0.1230469
0.5	20	0.0417309
0.5	30	0.0217753
0.5	40	0.0135072
0.5	50	0.0091770
0.5	60	0.0065758
0.5	70	0.0048678
0.5	80	0.0036784
0.5	90	0.0028174
0.5	100	0.0021780

Question 2

You sequence a locus in three individuals from a population and obtain the below data:

```
I1:
  GCTACTTTACCATTCTCAGCGAGACGTAAGATCAGGCCAGATCCACCTCG
  GTTCCTTTAACATTCTCAGCGAGACGTAAGAGCAGGCCAGATCCACCGCC
I2:
  GCTCCTTTACCATCCTCAGCGAGACGTAAGATCAGGACAGATCCACCTCG
  GCTACTTTACCATCCTCAGCGACACGTAAGATCAGGCCAGATCCACCTCG
I3:
  GCTACTTTACCATCCTCAGCGAGACGTAAGAGCAGGCCAGATCCACCTCC
  GCTACTTTACCATCCTCAGCGAGACGTAGGAGCAGGACAGATCCACCTCG
```

Part A

How many segregating sites (s) are present in these data?

```r
seq.df <- data.frame(
  i1.1=unlist(strsplit('GCTACTTTACCATTCTCAGCGAGACGTAAGATCAGGCCAGATCCACCTCG', '')),
  i1.2=unlist(strsplit('GTTCCTTTAACATTCTCAGCGAGACGTAAGAGCAGGCCAGATCCACCGCC', '')),
  i2.1=unlist(strsplit('GCTCCTTTACCATCCTCAGCGAGACGTAAGATCAGGACAGATCCACCTCG', '')),
  i2.2=unlist(strsplit('GCTACTTTACCATCCTCAGCGACACGTAAGATCAGGCCAGATCCACCTCG', '')),
  i3.1=unlist(strsplit('GCTACTTTACCATCCTCAGCGAGACGTAAGAGCAGGCCAGATCCACCTCC', '')),
  i3.2=unlist(strsplit('GCTACTTTACCATCCTCAGCGAGACGTAGGAGCAGGACAGATCCACCTCG', ''))
)

site_is_segregating <- function(row){

  copy1 <- length(unique(unlist(row[seq(1, length(row), 2)]))) == 1
  copy2 <- length(unique(unlist(row[seq(2, length(row), 2)]))) == 1
  if (copy1 & copy2){
    FALSE
  }else{
    TRUE
  }
}

seg_sites <- sum(apply(seq.df, 1, site_is_segregating))

message(paste('Number of segregating sites:', seg_sites))
```

```
## Number of segregating sites: 10
```

Part B

What is pi in these data?

```r
pi <- function(seq.df){

  comparisons <- combn(1:ncol(seq.df), 2)
  distances <- c()
  for (i in 1:ncol(comparisons)){
    col_a <- comparisons[1, i]
    col_b <- comparisons[2, i]
    num_diffs <- nrow(seq.df) - (sum(seq.df[, col_a] == seq.df[, col_b]))
    distances <- c(distances, num_diffs)
  }

 sum(distances) / ncol(comparisons)
}

pi_genes <- pi(seq.df)
message(paste('Pi =', pi_genes))
```

```
## Pi = 4.4
```

Part C

What are s and pi expressed in per site values?

```r
message(paste('Pi per site =', pi_genes / nrow(seq.df)))
```

```
## Pi per site = 0.088
```

```r
message(paste('Number of segregating sites per site:', seg_sites / nrow(seq.df)))
```

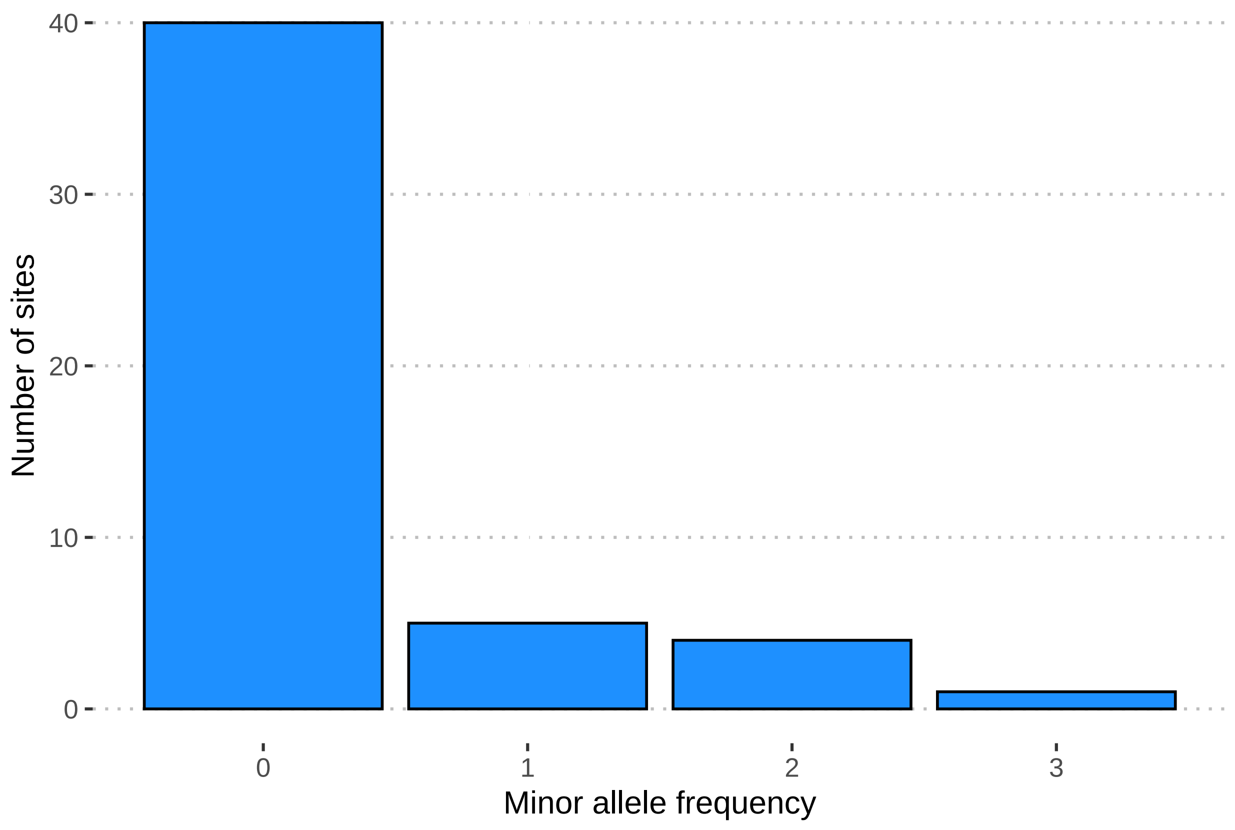
```
## Number of segregating sites per site: 0.2
```

Part D

What is the minor allele frequency spectrum for these data?

```r
# number of sites with given minor allele freq, max could be 3 / 6 otherwise
# major allele
# number sites is length of seq

minor_allele_count <- function(row){

  # get number of unique alleles need to determine what major / minor allele is
  # if 1 then only 1 allele so minor allele = 09
  row.ul <- as.vector(unlist(row))
  unq.alleles <- unique(row.ul)
  if (length(unq.alleles) == 1){
    0
  }else{
    # should be length 2 if not 1
    allele.a.count <- sum(row.ul == unq.alleles[1])
    allele.b.count <- sum(row.ul == unq.alleles[2])
    # minor allele freq will be the min of these values
    min(allele.a.count, allele.b.count)
  }
}
```

```r
seq.df.minor <- seq.df
seq.df.minor$minor <- apply(seq.df, 1, minor_allele_count)

ggplot(seq.df.minor, aes(x=minor)) + geom_bar(fill='dodgerblue', color='black') +
  theme_pubclean() + labs(x='Minor allele frequency', y='Number of sites')
```



Part E

You next sequence the locus in a few closely related species and determine the ancestral sequence to bethe following.

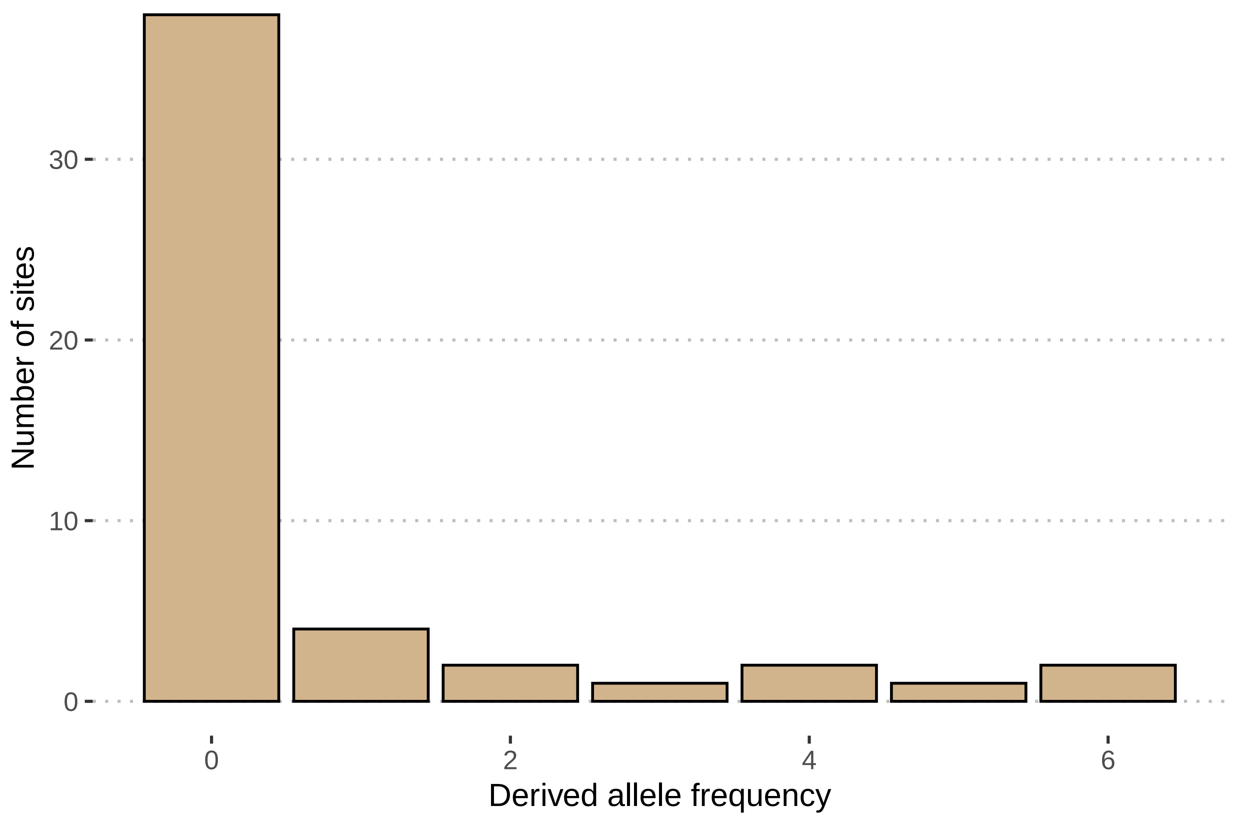
`Ancestral: GCTCCTTTACCATCCTCAGGGACACGTAAGAGCAGGCCAGACCCACCTCC`

What is the derived allele frequency spectrum for these data?

```r
# add ancestral seq to df
ancestoral <- 'GCTCCTTTACCATCCTCAGGGACACGTAAGAGCAGGCCAGACCCACCTCC'
seq.df.anc <- seq.df
seq.df.anc$ancestor <- unlist(strsplit(ancestoral, ''))
```

```r
derived_allele_freq <- function(row){

  # assume ancestral allele is last value
  row.ul <- as.vector(unlist(row))
  ansestoral <- row[length(row)]
  sum(row[1:length(row)-1] != ansestoral)

}
```

```r
derived <- apply(seq.df.anc, 1, derived_allele_freq)
ggplot(as.data.frame(derived), aes(x=derived))  + geom_bar(fill='tan', color='black') +
  theme_pubclean() + labs(x='Derived allele frequency', y='Number of sites')
```



Question 3

Use a program such as R or Excel to generate a scatter plot that shows the properties of the coalescent process in a Wright-Fisher population. The x-axis should be number of gene copies and range from 2-50. The y-axis should be expected number of generations in N units. Perform calculations in steps of one gene copy and plot the following three expectations: (1) time to the first coalescent event; (2) time to the most recent common ancestor of all gene copies; (3) total tree length.

```
gene_copies <- 2:50
```

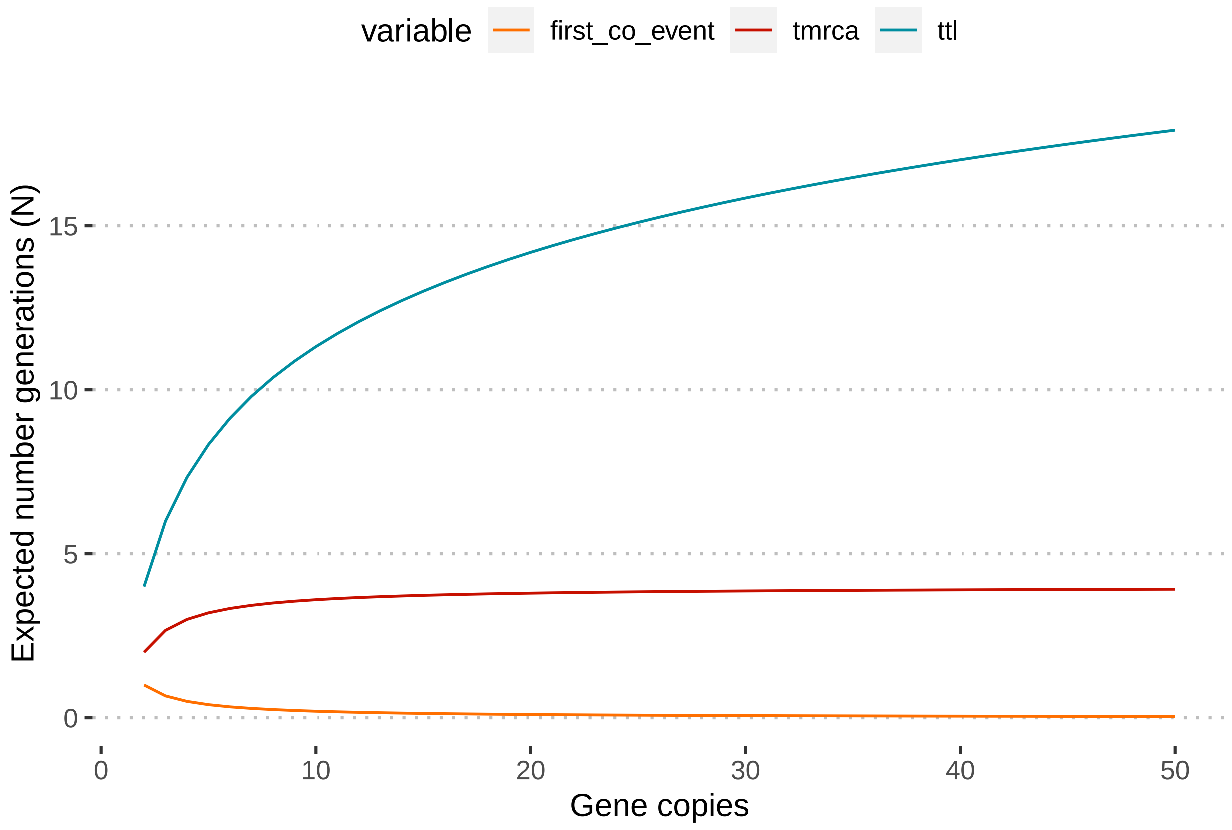
```
co.df <- data.frame(gene_copies=gene_copies)
# E(time to co event ) = 2n / number of gene copies
first_co_event <- function(copies){
  2 / copies
}

co.df$first_co_event <- unlist(lapply(gene_copies, first_co_event))
```

```
tmrca <- function(gene_copies){

  vals <- c()
  for (k in 2:gene_copies){
    vals <- c(vals, (2 / ((k*(k-1)) / 2)))
  }
  sum(vals)
}

co.df$tmrca <- unlist(lapply(gene_copies, tmrca))
```

```r
ttl <- function(gene_copies){
   vals <- c()
  for (k in 2:gene_copies){
    vals <- c(vals, (2 / ((k*(k-1)) / 2)) * k)
  }
  sum(vals)
}

co.df$ttl <- unlist(lapply(gene_copies, ttl))
```

```r
library(reshape2)
library(ggsci)
co.df.melt <- melt(co.df, id.vars="gene_copies")
ggplot(co.df.melt, aes(x=gene_copies, y=value, color=variable)) +
  geom_line() + theme_pubclean() +
  labs(y='Expected number generations (N)', x='Gene copies') +
  scale_color_discrete(name = "variable",
                       labels = c("First coalenscent event", "tMRCA", "TTL"),
                       ) +
  scale_color_futurama()
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```


variable
first_co_event
tmrca
ttl
Expected number generations (N)
Gene copies