Evoltutionary Dynamics Homework 3

Minghang Li

October 19, 2024 13:10 (America/New_York)

This solution is organized in R Markdown, and the coding part is written in `python` via `reticulate`.

Problem 2: Neutral Moran process

Consider the neutral Moran process $X(t)|t = 0, 1, 2, ...$ with two alleles A and B, where $X(t)$ is the number of A alleles in generation t.

(a)

Show that the process has a stationary mean:

$$E[X(t)|X(0) = i] = i. \tag{1}$$

Hint: First calculate $E[X(t)|X(t-1)]$ and use the law of total expectation, $E_Y[Y] = E_Z[E_Y[Y|Z]]$ with $Y = X(t)$ and $Z = X(t-1)$.

Proof. Since it's a neutral Moran process, at each generation, the population size is constant, and the probability of A and B to reproduce and die is the same.

Therefore, given $X(t-1) = i$, the frequency of allele A is $p = i/N$, and the transition probabilities are: $P_{i,i-1} = P_{i,i+1} = p(1-p)$, $P_{i,i} = p^2 + (1-p)^2$.

$$\begin{aligned} E_[X(t)|X(t-1) = i] &= \sum_{x_t \in X(t)} x_t \cdot P_{X(t-1),x_t} \\ &= i \cdot P_{i,i} + (i-1) \cdot P_{i,i-1} + (i+1) \cdot P_{i,i+1} \\ &= i \cdot (p^2 + (1-p)^2) + (i-1) \cdot p(1-p) + (i+1) \cdot p(1-p) \\ &= i \\ &= X(t-1) \end{aligned} \tag{2}$$

Hence we have $E_{X(t)}[X(t)|X(t-1)] = X(t-1)$. By the law of total expectation:

$$\begin{aligned} E[X(t)|X(0) = i] &= E[E[X(t)|X(t-1)]|X(0) = i] \\ &= E[X(t-1)|X(0) = i] \\ &= ... \quad \text{(I can do this all day!)} \\ &= E[X(0)|X(0) = i] \\ &= i \end{aligned} \tag{3}$$

□

(b)

Show that the variance of $X(t)$ is given by:

$$Var[X(t)|X(0) = i] = V_1 \frac{1 - (1 - 2/N^2)^t}{2/N^2} \tag{4}$$

Consider the following steps:

(i)

Show that $V_1 := Var[X(1)|X(0) = i] = 2(i/N)(1 - i/N)$.

Proof. Given $X(0) = i$, the transition probabilities are: $P_{i,i-1} = P_{i,i+1} = p(1-p)$, $P_{i,i} = p^2 + (1-p)^2$, where $p = i/N$.

From **(a)** we know that $E[X(1)|X(0) = i] = i$. Therefore, we have:

$$\begin{aligned} V_1 &= Var[X(1)|X(0) = i] \\ &= E[X(1)^2|X(0) = i] - E[X(1)|X(0) = i]^2 \\ &= i^2 \cdot P_{i,i} + (i-1)^2 \cdot P_{i,i-1} + (i+1)^2 \cdot P_{i,i+1} - i^2 \\ &= i^2 \cdot (p^2 + (1-p)^2) + (i-1)^2 \cdot p(1-p) + (i+1)^2 \cdot p(1-p) - i^2 \\ &= 2p(1-p) \\ &= 2\frac{i}{N}\left(1 - \frac{i}{N}\right) \end{aligned} \tag{5}$$

□

(ii)

Then use that $\forall t > 0, Var[X(t)|X(t-1) = i] = Var[X(1)|X(0) = i]$ (why?) and the *law of total variance*, $Var[Y] = E_Z[Var_Y[Y|Z]] + Var_Z[E_Y[Y|Z]]$, to derive

$$Var[X(t)|X(0) = i] = V_1 + (1 - 2/N^2)Var[X(t-1)|X(0) = i] \tag{6}$$

Proof. The mean of the Moran process is stationary (as shown in **(a)**), hence $\forall t > 0, Var[X(t)|X(t-1) = i] = Var[X(1)|X(0) = i]$.

All the E and Var on the right hand side of the equation are subscripted by $X(t-1)$. The author is just too

lazy to type.

$$
\begin{aligned}
Var[X(t)] &= E[Var[X(t)|X(t-1)]] + Var[E[X(t)|X(t-1)]] \\
&= E\left[2 \cdot \frac{X(t-1)}{N} \cdot \left(1 - \frac{X(t-1)}{N}\right)\right] + Var[X(t-1)] \\
&= E\left[2 \cdot \frac{X(t-1)}{N} - 2 \cdot \frac{X(t-1)^2}{N^2}\right] + Var[X(t-1)] \\
&= 2 \cdot \frac{E[X(t-1)]}{N} - 2 \cdot \frac{E[X(t-1)^2]}{N^2} + Var[X(t-1)] \\
&= 2 \cdot \frac{E[X(t-1)]}{N} - 2 \cdot \frac{Var[X(t-1)] + E[X(t-1)]^2}{N^2} + Var[X(t-1)] \quad (E[A^2] = Var[A] + E[A]^2) \\
&= 2 \cdot \frac{E[X(t-1)]}{N}\left(1 - \frac{E[X(t-1)]}{N}\right) + \left(1 - \frac{2}{N^2}\right) Var[X(t-1)]
\end{aligned}
\tag{7}
$$

Applying (1), we get the desired result:

$$
\begin{aligned}
Var[X(t)|X(0)=i] &= 2 \cdot \frac{E[X(t-1)|X(0)=i]}{N}\left(1 - \frac{E[X(t-1)|X(0)=i]}{N}\right) + \left(1 - \frac{2}{N^2}\right) Var[X(t-1)|X(0)=i] \\
&= 2 \cdot \frac{i}{N}\left(1 - \frac{i}{N}\right) + \left(1 - \frac{2}{N^2}\right) Var[X(t-1)|X(0)=i] \\
&= V_1 + \left(1 - \frac{2}{N^2}\right) Var[X(t-1)|X(0)=i]
\end{aligned}
\tag{8}
$$

□

(iii)

The inhomogeneous recurrence equation above can be solved by bringing it into the form $x_t - a = b(x_{t-1} - a)$, from which it follows that $x_t - a = b^{t-1}(x_1 - a)$.

Let's denote $Var[X(t)|X(0)=i]$ as V_t for simplicity. We can rewrite

$$V_t = V_1 + \left(1 - \frac{2}{N^2}\right) V_{t-1} \tag{9}$$

into

$$V_t - a = b(V_{t-1} - a), \tag{10}$$

where

$$\begin{cases} a = \dfrac{V_1}{2/N^2} \\ b = \left(1 - \dfrac{2}{N^2}\right) \end{cases}$$

Proof. Let's denote $2/N^2$ as m for simplicity. From (10) it follows that:

$$
\begin{aligned}
V_t - a &= b^{t-1}(V_1 - a) \\
V_t - \frac{V_1}{m} &= (1-m)^{t-1}\left(V_1 - \frac{V_1}{m}\right) \\
V_t &= V_1(1-m)^{t-1}\left(\frac{m-1}{m}\right) + \cdot V_1 \frac{1}{m} \\
V_t &= V_1 \frac{1-(1-m)^t}{m}
\end{aligned}
\tag{11}
$$

Expanding the simplified equation (11) we get the desired result:

$$Var[X(t)|X(0) = i] = V_1 + \frac{1 - (1 - 2/N^2)^t}{2/N^2} \tag{12}$$

□

(c)

Write a small simulation to check the results from (a) and (b). Use $N \in 10, 100$ and $i = N/2$. Simulate 1000 trajectories for $t = 1, ..., 100$, and compute empirical mean and variance in comparison to analytical mean and variance. Comment on your results shortly.

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(foreach)
```

```
##
## Attaching package: 'foreach'
##
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
library(doParallel)
```

```
## Loading required package: iterators
## Loading required package: parallel
```

```
library(ggplot2)
library(patchwork)
```

```
multiSimulate <- function(N, n) {
  res <- foreach (iter = 1:n, .packages = c("foreach")) %dopar% {
    i <- N / 2
    p <- i / N
    t <- 100

    A <- rep(0, t)
    A[1] <- i
    foreach (j = 2:t) %do% {
      A[j] <- sample(c(A[j - 1] - 1, A[j - 1] + 1, A[j - 1]),
                     1,
                     prob = c(p * (1 - p), p * (1 - p), p ^ 2 + (1 - p) ^ 2),
```

```
                      size = 1)
      if (A[j] < 0) {
        A[j] <- 0
      } else if (A[j] > N) {
        A[j] <- N
      }
    }
    return(A)
  }
  return(res)
}
```

```
set.seed(114514)
N <- c(10, 100)
n <- 1000

num_cores <- detectCores()
registerDoParallel(num_cores)
simulation_N10 <- multiSimulate(N[1], n)
simulation_N100 <- multiSimulate(N[2], n)
stopImplicitCluster()
```

```
df_N10 <- data.frame(
  value = unlist(simulation_N10),
  x = rep(1:100, times = 1000),
  group = rep(1:1000, each = 100)
)

var_df_N10 <- df_N10 %>% group_by(x) %>%
  summarise(mean = mean(value), var = var(value))

analytical_mean_N10 = N[1] / 2
analytical_var_N10 <- data.frame(
  t = 1:100,
  variance = (1/2) * (1 - (1 - 2 / N[1]^2) ^ (1:100)) / (2 / N[1]^2)
)

# Create the plot
lines_plot <- ggplot() +
  # Plot individual lines
  geom_line(
    data = df_N10,
    aes(x = x, y = value, group = group),
    color = "grey",
    linewidth = 1,
    linewidth = 0.5
  ) + # Set alpha for transparency
  geom_hline(
    yintercept = analytical_mean_N10,
    color = "red",
    linetype = "dashed",
    linewidth = 1
  ) +
  geom_line(
```

```
    data = var_df_N10,
    aes(x = x, y = mean),
    color = "magenta",
    linewidth = 1
  ) +
  labs(title = "Change of Allele A count (N=10)",
       x = "Generations (1 to 100)",
       y = "Allele A count") +
  theme(
    legend.position = "none",
    panel.grid.minor = element_blank(),
    panel.border = element_rect(
      color = "black",
      fill = NA,
      linewidth = 0.5
    ),
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 12, face = "bold"),
    plot.title = element_text(size = 14, face = "bold", hjust = 0.5)
  )
```

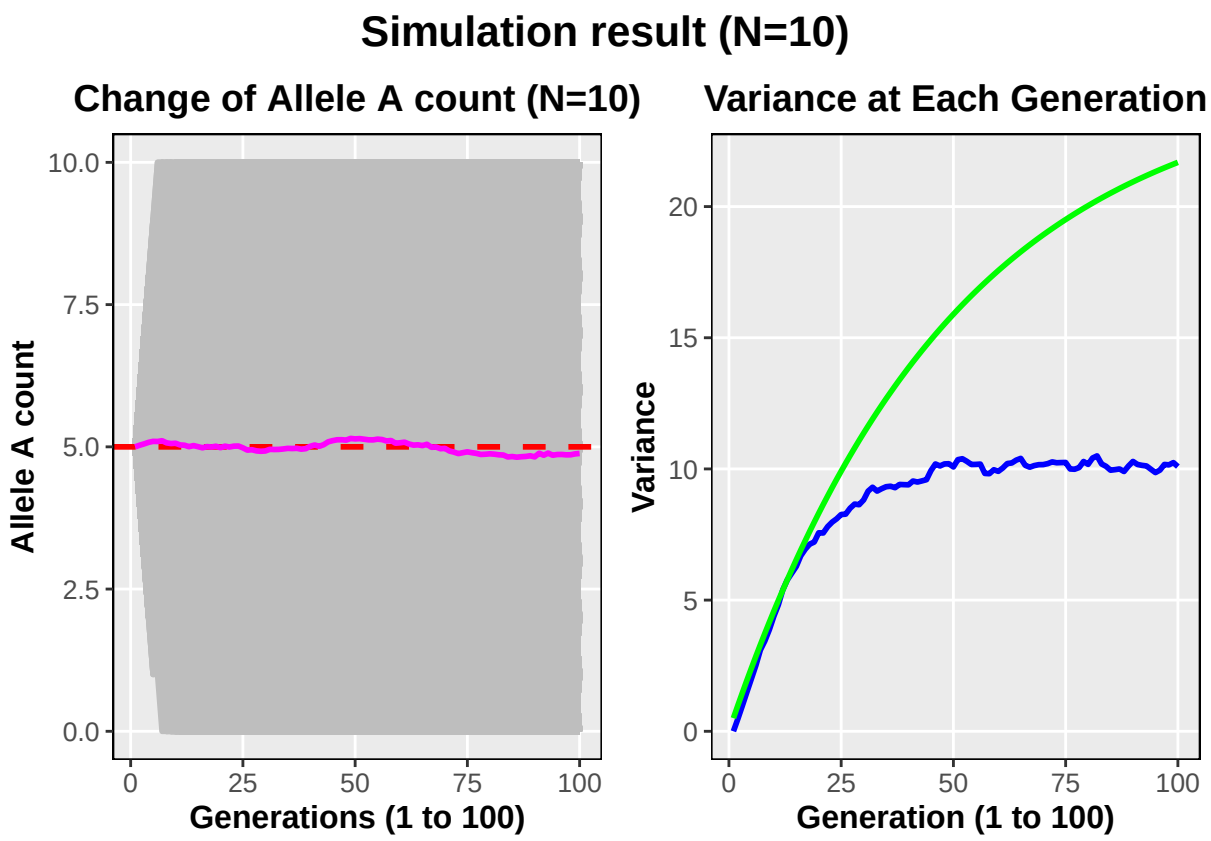
```
## Warning: Duplicated aesthetics after name standardisation: linewidth
```

```
# Create the variance plot
variance_plot <- ggplot() +
  geom_line(data = var_df_N10,
            aes(x = x, y = var),
            color = "blue",
            linewidth = 1) +
  geom_line(
    data = analytical_var_N10,
    aes(x = t, y = variance),
    color = "green",
    linewidth = 1
  ) +
  labs(title = "Variance at Each Generation",
       x = "Generation (1 to 100)",
       y = "Variance") +
  theme(
    legend.position = "none",
    panel.grid.minor = element_blank(),
    panel.border = element_rect(
      color = "black",
      fill = NA,
      linewidth = 0.5
    ),
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 12, face = "bold"),
    plot.title = element_text(size = 14, face = "bold", hjust = 0.5)
  )

# Combine the plots side by side
combined_plot <- lines_plot + variance_plot +
  plot_layout(ncol = 2) +
```

```
  plot_annotation(
    title = "Simulation result (N=10)",
    theme = theme(plot.title = element_text(size = 16, face = "bold", hjust = 0.5))
  )

# Display the combined plot
combined_plot
```



Note that the simulated variance is not as large as the theoretical variance, as I controlled the population size to be constant.

```
df_N100 <- data.frame(
  value = unlist(simulation_N100),
  x = rep(1:100, times = 1000),
  group = rep(1:1000, each = 100)
)

var_df_N100 <- df_N100 %>% group_by(x) %>%
  summarise(mean = mean(value), var = var(value))

analytical_mean_N100 = N[2] / 2
analytical_var_N100 <- data.frame(
  t = 1:100,
  variance = (1/2) * (1 - (1 - 2 / N[2]^2) ^ (1:100)) / (2 / N[2]^2)
)

# Create the plot
lines_plot <- ggplot() +
```

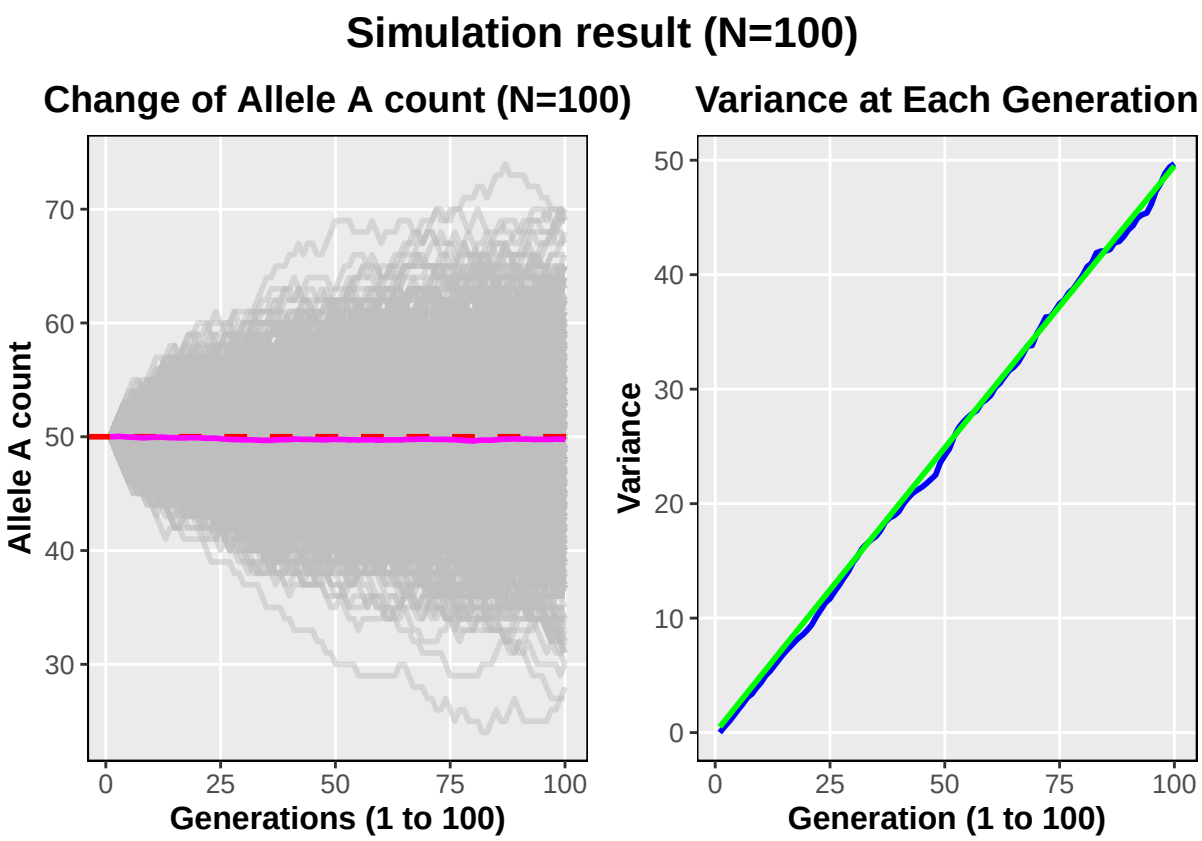
```r
  # Plot individual lines
  geom_line(
    data = df_N100,
    aes(x = x, y = value, group = group),
    color = "grey",
    linewidth = 1,
    alpha = 0.5
  ) + # Set alpha for transparency
  geom_hline(
    yintercept = analytical_mean_N100,
    color = "red",
    linetype = "dashed",
    linewidth = 1
  ) +
  geom_line(
    data = var_df_N100,
    aes(x = x, y = mean),
    color = "magenta",
    linewidth = 1
  ) +
  labs(title = "Change of Allele A count (N=100)",
       x = "Generations (1 to 100)",
       y = "Allele A count") +
  theme(
    legend.position = "none",
    panel.grid.minor = element_blank(),
    panel.border = element_rect(
      color = "black",
      fill = NA,
      linewidth = 0.5
    ),
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 12, face = "bold"),
    plot.title = element_text(size = 14, face = "bold", hjust = 0.5)
  )


# Create the variance plot
variance_plot <- ggplot() +
  geom_line(data = var_df_N100,
            aes(x = x, y = var),
            color = "blue",
            linewidth = 1) +
  geom_line(
    data = analytical_var_N100,
    aes(x = t, y = variance),
    color = "green",
    linewidth = 1
  ) +
  labs(title = "Variance at Each Generation",
       x = "Generation (1 to 100)",
       y = "Variance") +
  theme(
```

```
    legend.position = "left",
    panel.grid.minor = element_blank(),
    panel.border = element_rect(
      color = "black",
      fill = NA,
      linewidth = 0.5
    ),
    axis.text = element_text(size = 10),
    axis.title = element_text(size = 12, face = "bold"),
    plot.title = element_text(size = 14, face = "bold", hjust = 0.5)
  )

# Combine the plots side by side
combined_plot <- lines_plot + variance_plot +
  plot_layout(ncol = 2) +
  plot_annotation(
    title = "Simulation result (N=100)",
    theme = theme(plot.title = element_text(size = 16, face = "bold", hjust = 0.5))
  )

# Display the combined plot
combined_plot
```



The empirical mean in both simulations is consistent with the analytical mean $i = N/2$. Since we have equal transition probability on each side, the system evolves in a symmetric way. The empirical variance in $N = 10$ simulation is not a perfect match as the population size is controlled to be constant. The empirical variance in $N = 100$ simulation is closer to the theoretical variance.

Problem 3: Absorption in a birth-death process

Consider a birth-death process with state space $0, 1, ..., N$, transition probabilities $P_{i,i+1} = \alpha_i$, $P_{i,i-1} = \beta_i > 0$, and absorbing states 0 and N.

(a)

Show that the probability of ending up in state N when starting in state i is

$$x_i = \frac{1 + \sum_{j=1}^{i-1} \prod_{k=1}^{j} \gamma_k}{1 + \sum_{j=1}^{N-1} \prod_{k=1}^{j} \gamma_k} \tag{13}$$

Consider the following steps:

(i)

The vector $x = (x_0, ...x_N)^T$ solves $x = Px$ where P is the transition matrix. (Why?) Set $y_i = x_i - x_{i-1}$ and $\gamma_i = \beta_i/\alpha_i$. Show that $y_{i+1} = \gamma_i y_i$.

Proof. Because $x(t)$ is a Markov chain, and according to ergodicity, the chain will eventually reach a stationary distribution. The stationary distribution is the eigenvector of the transition matrix P with eigenvalue 1. Therefore, $x = Px$.

We can rewrite x_i using α_i and β_i:

$$\begin{aligned} x_i &= \alpha_i x_{i+1} + \beta_i x_{i-1} + (1 - \alpha_i - \beta_i) x_i \\ (\alpha_i + \beta_i) x_i &= \alpha_i x_{i+1} + \beta_i x_{i-1} \\ \alpha_i (x_{i+1} - x_i) &= \beta_i (x_i - x_{i-1}) \end{aligned} \tag{14}$$

Let $y_i = x_i - x_{i-1}$, $\gamma_i = \beta_i/\alpha_i$, substitute it into equation (14) we have

$$y_{i+1} = \gamma_i y_i$$

□

(ii)

Show that $\sum_{i=1}^{l} y_i = x_l$.

Proof. This is trivial.

$$\sum_{i=1}^{l} y_i = \sum_{i=1}^{l} (x_i - x_{i-1}) = x_l - x_0 = x_l$$

□

(iii)

Show that $x_l = \left(1 + \sum_{j=1}^{l-1} \prod_{k=1}^{j} \gamma_k\right) x_1$.

Proof.

$$\begin{aligned} x_l &= \sum_{i=1}^{l} y_i \\ &= y_1 + y_2 + ... + y_l \\ &= y_1 + \gamma_1 y_1 + \gamma_1\gamma_2 y_1 + ... + \prod_{k=1}^{l-1} \gamma_k y_1 \\ &= \left(1 + \sum_{j=1}^{l-1} \prod_{k=1}^{j} \gamma_k\right) y_1 \\ &= \left(1 + \sum_{j=1}^{l-1} \prod_{k=1}^{j} \gamma_k\right) x_1 \end{aligned} \tag{15}$$

By the fact that $x_N = 1$,

$$x_1 + x_1 \cdot \sum_{j=1}^{N-1} \prod_{k=1}^{j} \gamma_k = 1 \Longrightarrow x_1 = \frac{1}{1 + \sum_{j=1}^{N-1} \prod_{k=1}^{j} \gamma_k} \tag{16}$$

Combining (15) and (16) we get the desired result (13).

$$x_j = \frac{1 + \sum_{j=1}^{i-1} \prod_{k=1}^{j} \gamma_k}{1 + \sum_{j=1}^{N-1} \prod_{k=1}^{j} \gamma_k}$$

□

(b)

Using (13), show that for the Moran process *with selection*

$$\rho = x_1 = \frac{1 - 1/r}{1 - 1/r^N}, \tag{17}$$

where r is the relative fitness advantage. Use *l'Hôpital's rule* to calculate the limit $r \to 1$.

Assume that A allele has a relative fitness advantage of r over B allele. For a Moran process with constant selection, we have $\gamma_i = \frac{1}{r}$ in this scenario.

The equation (13) becomes:

$$x_i = \frac{1 - r^{-i}}{1 - r^{-N}}$$

In the case of $i = 1$,

$$x = \frac{1 - r^{-1}}{1 - r^{-N}}$$

The limit can be calculated:

$$\begin{aligned}\lim_{r\to 1}\frac{1-r^{-1}}{1-r^{-N}} &= \lim_{r\to 1}\frac{r^N-r^{N-1}}{r^N-1}\\ &= \lim_{r\to 1}1-\frac{r^{N-1}-1}{r^N-1}\\ &= 1-\lim_{r\to 1}\frac{(N-1)r^{N-2}}{N\cdot r^{N-1}} \qquad (l'Hospital)\\ &= \frac{1}{N}\end{aligned}$$

Problem 4: Poisson Process

Consider a Poisson process to model mutation events. We start with a population, where all individuals initially carry allele a, and new mutations result in allele b with a mutation rate u.

(a)

Show that the time until the first mutation appears, T_1, follows an exponential distribution $T1 \sim \text{Exp}(Nu)$.

Let t be the time interval of interest, N be the population size. The number of events in the interval t follows a Poisson distribution with mean λt where $\lambda = Nu$. The probability of having no mutation in the interval t is:

$$P(N(t)=0) = e^{-\lambda t}\frac{(\lambda t)^0}{0!} = e^{-\lambda t}$$

$T_1 \sim \text{Exp}(\lambda)$, because

$$P(T_1 > t) = P(N(t)=0) = e^{-\lambda t}$$

(b)

Given neutral evolution, show that the rate of evolution R, from an all-a population to an all-b population is $R = u$.

Suppose b has a selective advantage r, then the fixation probability is $\rho = x_1 = \frac{1-1/r}{1-1/r^N}$.

The rate of evolution is the probability of fixation times the rate of mutation:

$$R = N\cdot u\cdot \rho = Nu\cdot\frac{1-1/r}{1-1/r^N}$$

In the case of neutral evolution, limit$r \to 1$, the rate of evolution is $R = N\cdot u\cdot 1/N = u$.

(c)

Explain the assumptions and implications of this result for inferring the divergence time between species.

If u is constant, then neutral mutations accumulate at a constant rate $R = u$, independent of population size.

The divergence time between species can be inferred by the number of mutations that have accumulated in the two species. The number of mutations is proportional to the divergence time, and the rate of mutation u.