

Implementation of Evolution Model for Subspecies

Adapted Framework

2024-12-07

Introduction

This document implements the model of evolution for subspecies, based on a birth-death process with preferential attachment and mutation dynamics. The model incorporates the survival of the fittest principle while integrating environmental and stochastic factors to simulate evolutionary population dynamics.

Parameters

Set the key parameters for the simulation.

```
# Load libraries
library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

library(tidyr)

# Define parameters
set.seed(123)
p <- 0.6 # Probability of birth
r <- 0.4 # Probability of mutation
n_steps <- 100 # Number of time steps
initial_population <- data.frame(fitness = c(0.3, 0.6, 0.9), size = c(10, 20, 5))
```

Simulation Framework

The simulation follows a discrete-time process with the following rules:

1. **Birth:** With probability p , an individual is born.
 - With probability r , the individual is a mutant with a random fitness value.
 - Otherwise, preferential attachment determines its fitness.
2. **Death:** With probability $1 - p$, the least fit individual dies.

Simulation Function

Define a function to simulate the population dynamics over time.

```
simulate_population <- function(initial_population, n_steps, p, r) {
  population <- initial_population
  results <- list()

  for (t in 1:n_steps) {
    if (runif(1) < p) {
      # Birth event
      if (runif(1) < r) {
        # Mutation: New fitness level uniformly in [0, 1]
        new_fitness <- runif(1)
        population <- population %>%
          add_row(fitness = new_fitness, size = 1)
      } else {
        # Preferential attachment
        total_size <- sum(population$size)
        probabilities <- population$size / total_size
        chosen <- sample(seq_len(nrow(population)), size = 1, prob = probabilities)
        population$size[chosen] <- population$size[chosen] + 1
      }
    } else {
      # Death event: Remove one individual from the least fit group
      least_fit <- which.min(population$fitness)
      if (population$size[least_fit] > 1) {
        population$size[least_fit] <- population$size[least_fit] - 1
      } else {
        population <- population[-least_fit, ]
      }
    }

    # Store the population state
    results[[t]] <- population
  }

  results
}
```

Running the Simulation

Simulate the evolution of the population using the defined parameters and function.

```
# Run simulation
simulation_results <- simulate_population(initial_population, n_steps, p, r)

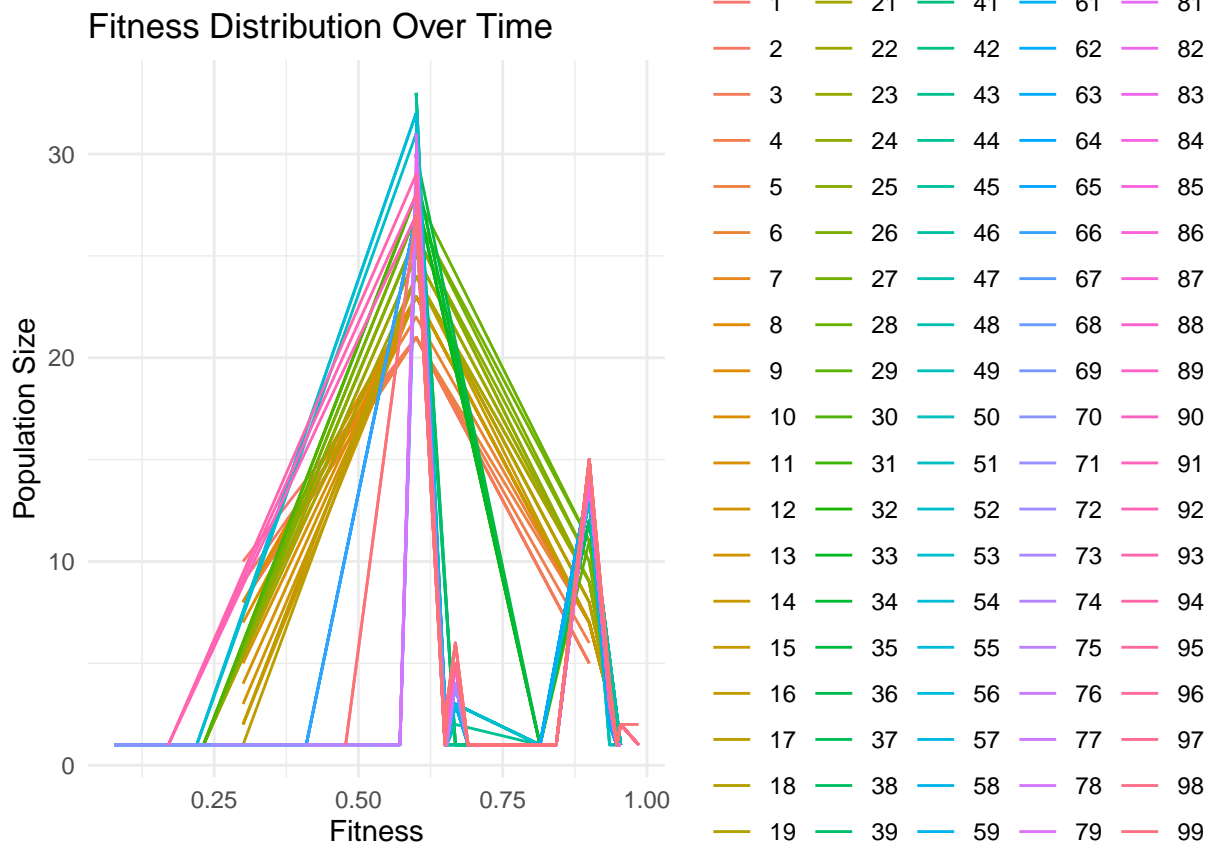
# Prepare results for visualization
plot_data <- bind_rows(lapply(seq_along(simulation_results), function(t) {
  data <- simulation_results[[t]]
  data$time <- t
  data
})))
```

Visualization

Fitness Distribution Over Time

Visualize how the fitness distribution evolves across the simulation timeline.

```
ggplot(plot_data, aes(x = fitness, y = size, color = as.factor(time))) +  
  geom_line() +  
  labs(title = "Fitness Distribution Over Time",  
        x = "Fitness",  
        y = "Population Size",  
        color = "Time Step") +  
  theme_minimal()
```



Population Dynamics

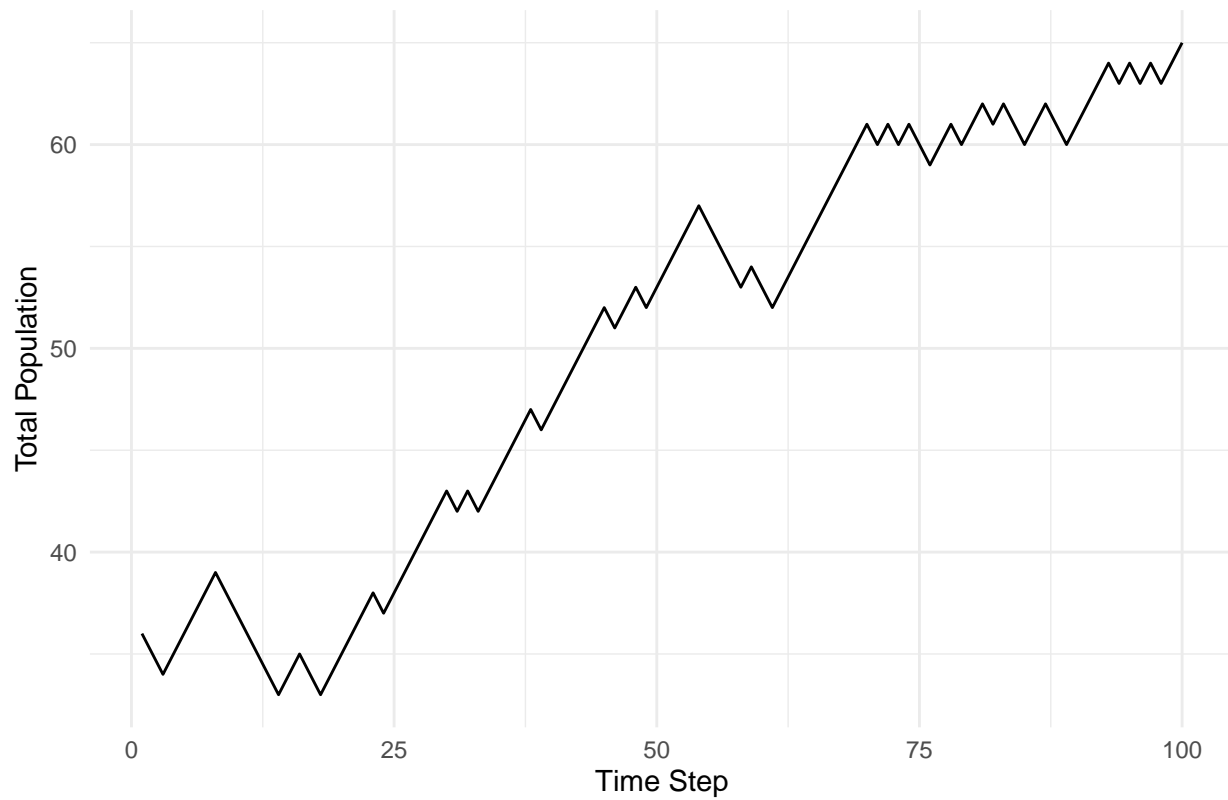
Total Population

Plot the total population size over time.

```
summary_data <- plot_data %>%  
  group_by(time) %>%  
  summarise(total_population = sum(size),  
            mean_fitness = weighted.mean(fitness, size))  
  
ggplot(summary_data, aes(x = time, y = total_population)) +  
  geom_line() +  
  labs(title = "Total Population Over Time",  
        x = "Time Step",
```

```
y = "Total Population") +  
theme_minimal()
```

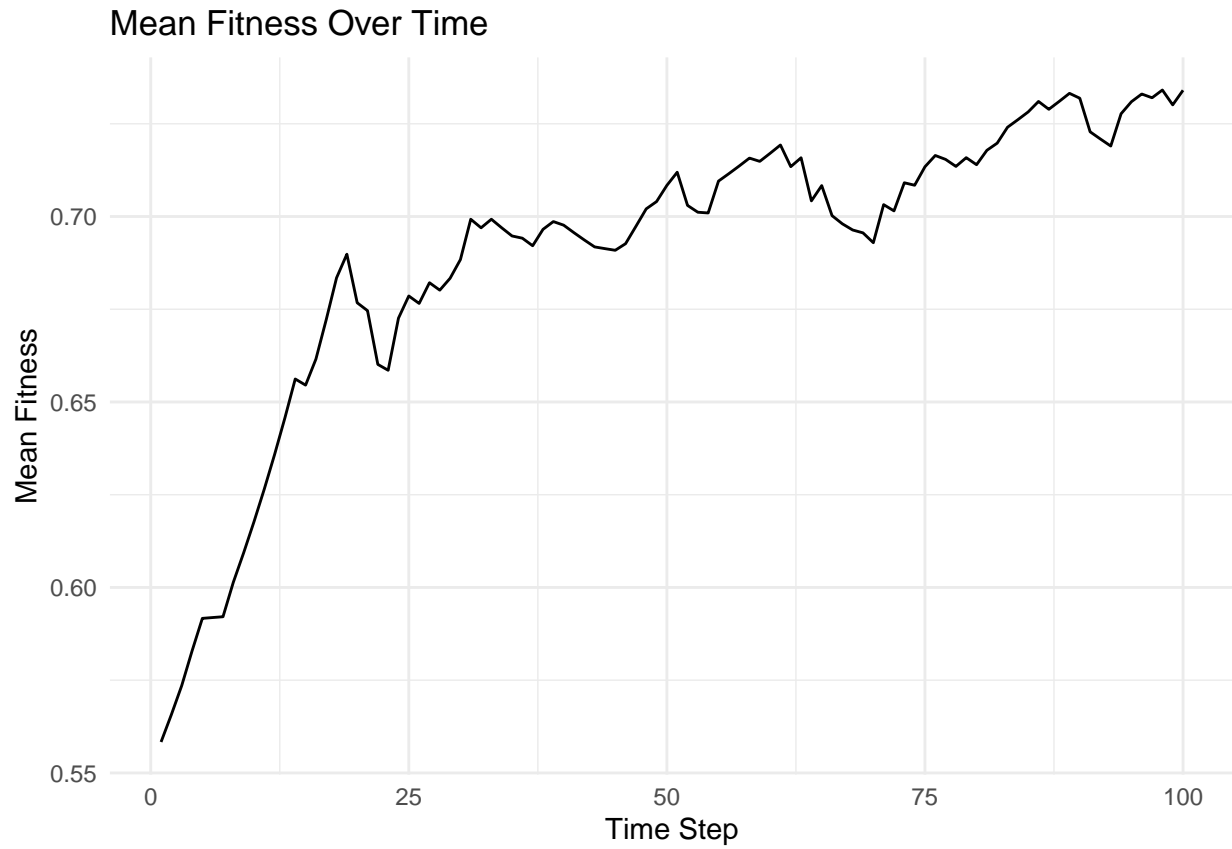
Total Population Over Time



Mean Fitness

Plot the mean fitness of the population over time.

```
ggplot(summary_data, aes(x = time, y = mean_fitness)) +  
  geom_line() +  
  labs(title = "Mean Fitness Over Time",  
        x = "Time Step",  
        y = "Mean Fitness") +  
  theme_minimal()
```



Conclusion

This implementation provides a framework for simulating and visualizing the evolution of subspecies populations. The integration of mutation and preferential attachment mechanisms highlights key evolutionary dynamics that can be adapted for specific ecological studies.