Final Project Progress Report 2

Rahul Rengan Ramakrishnan, Yusi Yao

2025-04-21

1. Project Overview

We are modeling how temperature affects the mosquito biting rate $a(T)$, a critical factor in malaria transmission dynamics. Using a Bayesian framework, we apply a custom Metropolis-Hastings MCMC algorithm to estimate the posterior distributions of parameters in the Brière function:

$$a(T) = cT(T - T_0)\sqrt{T_m - T}$$

Our goal is to quantify uncertainty in biological parameters and understand how temperature shifts could influence mosquito-related disease spread.

---

2. Work Completed (April 23–29)

2.1 MCMC Execution on Synthetic Data

We successfully completed a 10,000-iteration run of Metropolis-Hastings MCMC using synthetic data generated to mimic realistic mosquito trait-temperature relationships.

2.1.1 Previously Defined Functions

```
briere <- function(T, c, T0, Tm) {
  result <- ifelse(
    T > T0 & T < Tm,
    c * T * (T - T0) * sqrt(pmax(0, Tm - T)),
    0
  )
  return(result)
}
```

We define the Brière function, clamping negative values inside the square root using `pmax(0, Tm - T)` to avoid NaN issues during simulation.

```
log_likelihood <- function(data_temp, data_obs, c, T0, Tm, sigma = 1) {
  preds <- briere(data_temp, c, T0, Tm)
  log_lik <- dnorm(data_obs, mean = preds, sd = sigma, log = TRUE)
  return(sum(log_lik))
}
```

This function calculates the log-likelihood of observing the data given a set of model parameters assuming normal errors.

```r
mcmc_briere <- function(data_temp, data_obs, n_iter = 10000) {
  samples <- matrix(NA, nrow = n_iter, ncol = 3)
  colnames(samples) <- c("c", "T0", "Tm")

  c <- 0.0001
  T0 <- 10
  Tm <- 40

  for (i in 1:n_iter) {
    c_prop <- abs(c + rnorm(1, 0, 0.00001))
    T0_prop <- T0 + rnorm(1, 0, 0.5)
    Tm_prop <- Tm + rnorm(1, 0, 0.5)

    if (T0_prop >= Tm_prop) {
      samples[i, ] <- c(c, T0, Tm)
      next
    }

    ll_current <- log_likelihood(data_temp, data_obs, c, T0, Tm)
    ll_proposed <- log_likelihood(data_temp, data_obs, c_prop, T0_prop, Tm_prop)

    alpha <- exp(ll_proposed - ll_current)
    if (runif(1) < alpha) {
      c <- c_prop
      T0 <- T0_prop
      Tm <- Tm_prop
    }
    samples[i, ] <- c(c, T0, Tm)
  }

  return(as.data.frame(samples))
}
```

We implement a Metropolis-Hastings MCMC algorithm that iteratively proposes new parameter values and accepts or rejects based on likelihood improvement.

2.1.2 Generating Synthetic Data

```r
set.seed(42)
synthetic_data <- data.frame(
  temp = seq(10, 40, length.out = 100),
  trait_value = briere(seq(10, 40, length.out = 100), c = 0.0002, T0 = 10, Tm = 40) + rnorm(100, 0, 1)
)

head(synthetic_data)
```

```
##       temp trait_value
## 1 10.00000  1.37095845
## 2 10.30303 -0.56129537
## 3 10.60606  0.37009835
## 4 10.90909  0.64356065
## 5 11.21212  0.41885205
## 6 11.51515 -0.08750096
```

We generate synthetic data points to mimic trait-temperature relationships, adding random noise for realism.

2.1.3 Running the MCMC

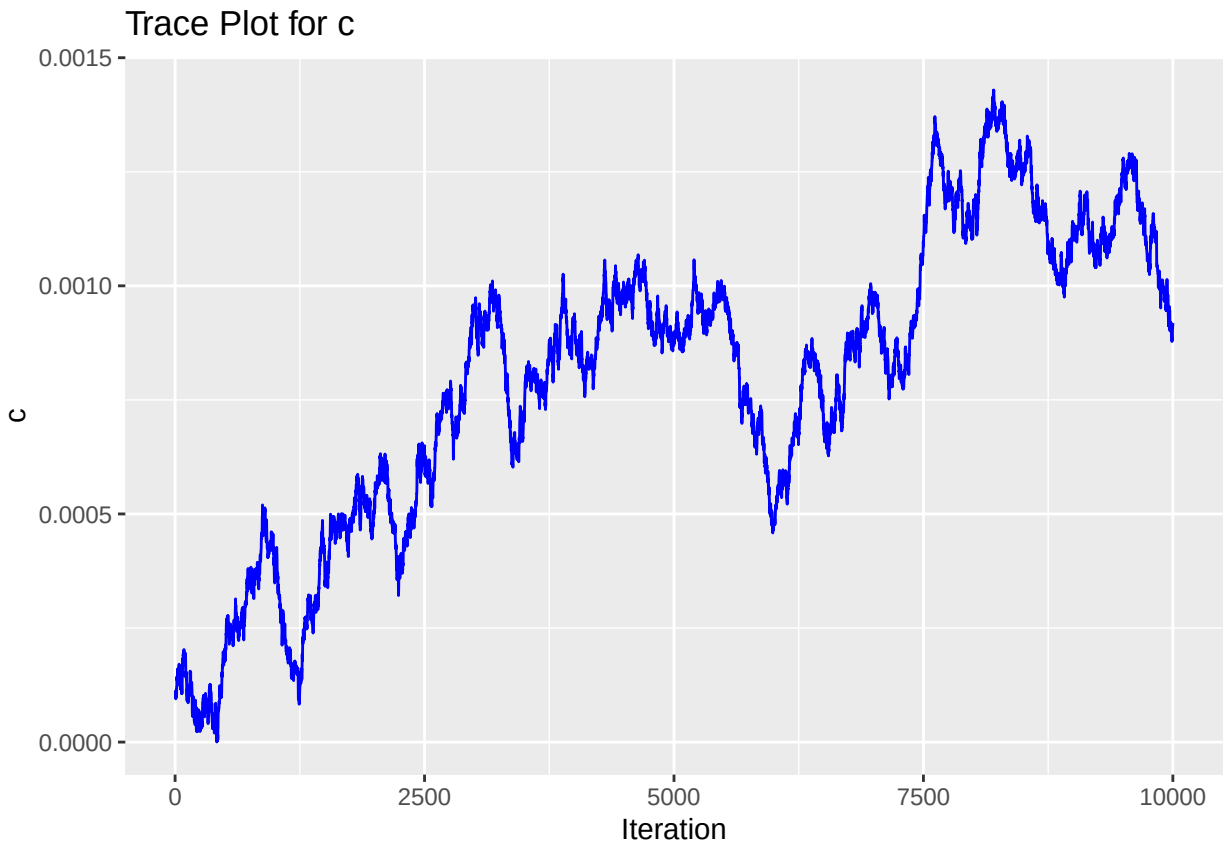
```
mcmc_samples <- mcmc_briere(
  data_temp = synthetic_data$temp,
  data_obs = synthetic_data$trait_value,
  n_iter = 10000
)

head(mcmc_samples)
```

```
##              c       T0       Tm
## 1 0.0001120097 10.52238 39.49840
## 2 0.0001116232 10.20022 39.69089
## 3 0.0001103997 10.29432 39.75047
## 4 0.0001009224 11.17831 40.20913
## 5 0.0001009224 11.17831 40.20913
## 6 0.0001094966 11.74135 40.10995
```

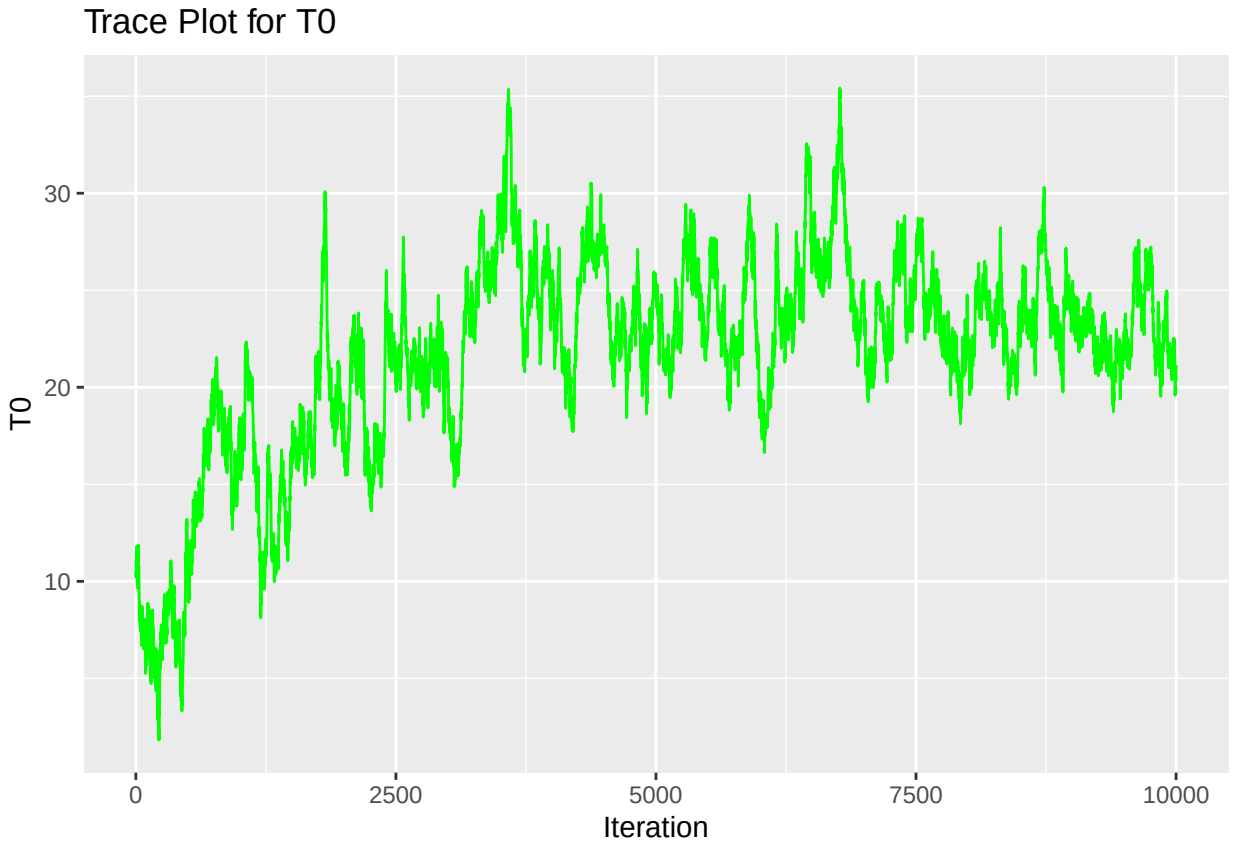
We run the MCMC sampler to generate a large set of posterior samples of c, T_0, and T_m.

2.2 Diagnostics: Trace Plots and Posterior Summaries

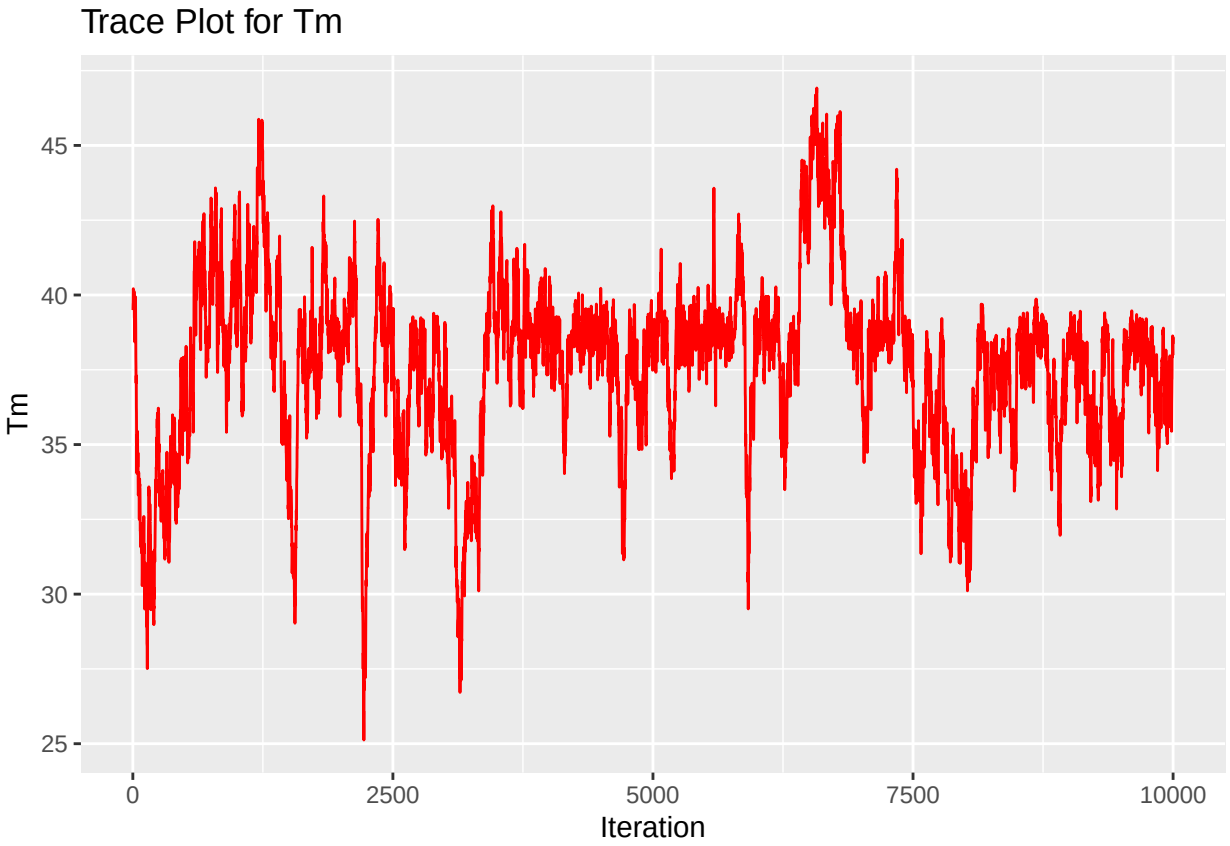
```
library(ggplot2)

mcmc_samples$iteration <- 1:nrow(mcmc_samples)

# Trace plot for c
p1 <- ggplot(mcmc_samples, aes(x = iteration, y = c)) +
  geom_line(color = "blue") +
  labs(title = "Trace Plot for c", x = "Iteration", y = "c")

# Trace plot for T0
p2 <- ggplot(mcmc_samples, aes(x = iteration, y = T0)) +
  geom_line(color = "green") +
  labs(title = "Trace Plot for T0", x = "Iteration", y = "T0")

# Trace plot for Tm
p3 <- ggplot(mcmc_samples, aes(x = iteration, y = Tm)) +
  geom_line(color = "red") +
  labs(title = "Trace Plot for Tm", x = "Iteration", y = "Tm")

p1
```


Trace Plot for c
0.0015
0.0010
0.0005
0.0000
c
0
2500
5000
7500
10000
Iteration

```
p2
```




```
p3
```



We visualize the sampling history for each parameter to check convergence. Ideally, the traces should appear as "random walks" around stable means.

---

3. Problems Encountered

- **Slow Convergence**: Trace plots showed some initial wandering, suggesting tuning of proposal standard deviations is necessary.
- **Parameter Drift**: Occasionally unrealistic values for T_0 and T_m were proposed, suggesting the need for biologically informed soft bounds.
- **Synthetic Bias**: Our synthetic dataset was idealized, so real-world data may require stronger priors or adjustments for higher variability.

---

4. Timeline Update

Week	Dates	Status	Notes
Week 3	**April 23–29**	**Completed**	Full MCMC run, synthetic results, diagnostics
Week 4	April 30–May 7	In Progress	Final tuning, real data integration, posterior predictive checking

5. Next Steps

- Tune MCMC proposal distributions for faster convergence
- Implement biologically-informed soft bounds on parameters
- Transition from synthetic to real trait data
- Finalize posterior prediction plots and statistical summaries
- Begin drafting the final project report and poster presentation