

# ***Estimating parameters of a population model using approximate Bayesian computation***

---

Sheng-Pin, Fan (n10941576)

MXN 442 Final Project

## **Abstract**

This research aims to estimate the probability of birth events for coral, death events for starfish, and predator events where starfish absorb coral using various Approximate Bayesian Computation (ABC) methods. The Gillespie Algorithm is employed to simulate data. During ABC rejection, a summary statistic is selected for application in Markov Chain Monte Carlo (MCMC) ABC. Subsequently, based on the results of MCMC ABC, the target tolerance for the difference between simulated data and observed data is defined. This target tolerance serves as the final target in Sequential Monte Carlo (SMC) ABC. Through three distinct methods, we seek to generate results that elucidate these three events.

**Keywords:** Sequential Monte Carlo (SMC), Approximate Bayesian Computation (ABC), crown of thorns starfish (COTS), Gillespie Algorithm.

## **1. Introduction**

Coral is a crucial species inhabiting tropical oceans, offering substantial benefits to both humans and marine ecosystems. Serving as hosts for over 25 percent of marine species [1], coral provides vital protection against predators for these diverse marine inhabitants. In turn, the food residues from these species contribute to the energy needed for coral survival. The advantages extend to human communities in various ways, encompassing entertainment, natural resource provision, protective functions, and economic contributions [2]. The vibrant coral formations and abundant marine life create captivating scenes, drawing in numerous divers. The influx of divers, in turn, generates thousands of opportunities for local economic growth. Moreover, coral acts as a natural barrier, safeguarding coastlines from wave erosion. In recent years, the exploration of coral immune systems has revealed potential treatments for human diseases, particularly those deemed untreatable [3]. Consequently, the loss of coral poses a significant threat to marine ecosystems and becomes a tragedy for humanity. Unfortunately, over 60 percent of coral faces direct and indirect threats [1], with global climate change posing a direct risk to their survival.

Instead of human activity, the crown-of-thorns starfish (COTS) presents another significant factor strongly linked to the abundance of coral. As voracious coral predators, COTS pose a substantial threat to coral survival. The dynamics of the two species' populations can be elucidated through

three pivotal events: the birth of coral, the predator event involving COTS, and the death of starfish. These events collectively form a Markov process, contingent upon the respective numbers of coral and COTS alive at time  $t$ , denoted as  $C(t)$  and  $S(t)$ . The probabilities of each event occurring at time  $t + \Delta$  are governed by a set of probabilities, depicting the intricate interplay between coral and COTS populations.

$$P(C = c + 1, S = s) = \mu c \Delta + o(\Delta)$$

$$P(C = c - 1, S = s + 1) = \delta s c \Delta + o(\Delta)$$

$$P(C = c, S = s - 1) = \nu s \Delta + o(\Delta)$$

However, the parameters associated with the birth event  $\mu$ , predator event  $\delta$ , and death event,  $\nu$ , are unknown, rendering the model contingent on these unidentifiable factors. This dependency necessitates the utilization of statistical methods for parameter estimation and inference. Various options exist for estimating parameters in the mathematical model [1], with one example being Markov Chain Monte Carlo (MCMC). Nevertheless, MCMC algorithms demand a precise evaluation of the likelihood function, which proves challenging in this case. Consequently, likelihood-free technologies, such as approximate Bayesian computation (ABC), are considered as alternative approaches due to their capacity to handle scenarios where obtaining an exact likelihood is difficult.

Approximate Bayesian Computation (ABC) is a powerful statistical framework employed in situations where exact likelihood computations are infeasible or computationally intensive. Unlike traditional Bayesian methods that rely on the exact likelihood function, ABC is particularly well-suited for complex models and scenarios with intricate dependencies. The fundamental idea behind ABC is to approximate the posterior distribution of model parameters by comparing simulated data generated from the model with observed data. Instead of calculating the exact likelihood, ABC assesses the similarity between the observed and simulated datasets based on a set of predefined summary statistics. By accepting or rejecting samples based on a defined criterion, ABC iteratively refines the parameter estimates, allowing researchers to explore and infer plausible parameter values in scenarios where obtaining an exact likelihood is impractical. This flexibility makes ABC a valuable tool in various fields, including ecology, genetics, and economics, where intricate models and computationally demanding simulations are common.

## 2. Case Study

### Data

The provided dataset consists of three columns: "times," "coral," and "starfish." The "times" column represents time intervals, every two years. The "coral" column denotes populations of coral at each corresponding time interval, and similarly, the "starfish" column represents the populations of starfish at those time points. The dataset appears to capture observations over

several time intervals, providing information about the changing populations of both coral and starfish. Figure 1 shows the amount of both species against time intervals.

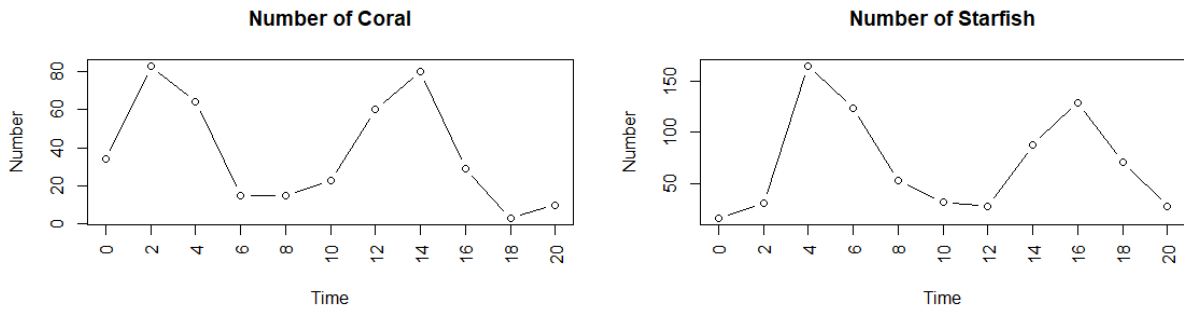


Figure 1: The number of Coral and COTS against time.

## Simulation

The Gillespie Algorithm, a widely employed computational tool in the field of ecological modelling, played a pivotal role in this research to simulate data for the interaction between coral and the crown-of-thorns starfish (COTS). Specifically, the algorithm was harnessed to model three essential events shaping the dynamics of these populations: the birth of coral, the predatory actions of COTS, and the death of starfish. Leveraging the Gillespie Algorithm facilitated the generation of stochastic simulations that captured the probabilistic nature of these events over time. This algorithm's ability to simulate complex ecological systems provided a dynamic framework for understanding the population dynamics of coral and COTS, enabling insights into their interactions and the potential threats posed by the starfish predation on coral populations.

## ABC Rejection

The likelihood function is approximated using simulated data, which is then compared to observations in all ABC methods. ABC rejection is a straightforward algorithm, algorithm 1, that utilizes the distance between simulated data and observations to make decisions. Initially, samples are drawn from the prior distribution as candidates and are then used to simulate data. A chosen set of summary statistics is employed to determine the distance between the two datasets. In other words, the summary statistics calculate the closeness between these two datasets. For instance, [4] utilized the absolute value as the measurement. A threshold is applied to reject samples, which can be predefined or selected after generating samples. This selection influences the acceptance rate of the samples. A small threshold results in a high rejection rate but provides accurate samples. An adequately small and appropriately chosen threshold, along with a sensible similarity measurement to represent the difference, can generally lead to an accurate representation of the actual distribution [4].

---

**Algorithm 1: ABC Rejection**

---

- 1 Define threshold,  $\epsilon$ , target number of particle,  $T$ , initialise  $t = 1$ .
  - 2 While  $t < T$  do
  - 3     Draw  $\mu_t, \delta_t, \nu_t$  from unif(0, 1)
  - 4     Simulate data,  $y$ , from **Gillespie Algorithm**
  - 5     Calculate the closeness of Simulated data and observation,  $\epsilon_t = \rho(y, x)$
  - 6     If  $\epsilon_t < \text{threshold}$ :
  - 7         store the particle  $\{\mu_t, \delta_t, \nu_t, \epsilon_t\}$
  - 8      $t = t + 1$
- 

## MCMC ABC

MCMC is another method that can be employed to approximate parameters. However, constructing a Markov Chain can sometimes be challenging, especially when the likelihood is too complex for evaluation. MCMC ABC is an alternative method that addresses this issue by modifying the Metropolis-Hastings (MH) rate. Since the likelihood is difficult to calculate, it is replaced by the closeness between simulated data and observations. Nevertheless, MCMC ABC has exhibited a poor mixing performance in the tails of the target distribution when compared to traditional MCMC methods [5]. When the mixing in the tails is problematic, there are several suggested approaches to resolve the issue. Firstly, using a different MCMC kernel that can propose samples significantly different from the current state [5]. Additionally, maintaining the probability of proposed values in the tail of the distribution is recommended. Alternatively, another ABC algorithm, SMC ABC, can be considered.

---

**Algorithm 2: ABC MCMC**

---

- 1 Draw  $\mu_0, \delta_0, \nu_0$  from unif(0, 1) and simulate  $x_0$
  - 2 Calculus closeness  $\epsilon_0 = \rho(y, x_0)$
  - 3 **For**  $t = 1$  to  $N$  **do**
  - 4     Draw candidate  $\mu^*, \delta^*, \nu^*$  from  $q(\cdot | \mu_{t-1}, \delta_{t-1}, \nu_{t-1})$
  - 5     Simulate  $x_t$  according to  $\mu^*, \delta^*, \nu^*$
  - 6     Calculus closeness  $\psi_t = \rho(y, x_t)$
  - 7     Compute MH ratio  $r = \frac{\pi(\mu^*)\pi(\delta^*)\pi(\nu^*)\psi^* q(\mu_{t-1}, \delta_{t-1}, \nu_{t-1} | \mu^*, \delta^*, \nu^*)}{\pi(\mu_{t-1})\pi(\delta_{t-1})\pi(\nu_{t-1})\psi^{t-1} q(\mu^*, \delta^*, \nu^* | \mu_{t-1}, \delta_{t-1}, \nu_{t-1})}$
  - 8     **If**  $U(0, 1) < r$  **then**
  - 9         Accept candidate and  $\psi^t = \psi^*$
  - 10    **else**
  - 11        $\mu_t = \mu_{t-1}, \delta_t = \delta_{t-1}, \nu_t = \nu_{t-1}, \psi^t = \psi^{t-1}$
-

## Sequential Monte Carlo (SMC) ABC

The concept of SMC can be viewed as an extension or adaptation of importance sampling. The key is to employ different proposal distributions in place of the prior to propose the sample (particles). These particles are then weighted to follow an importance distribution. Once a set of particles is sampled, the concept of importance sampling is utilized to weigh the particles. These weighted particles are used to define a new proposal distribution, which is employed in the next iteration [6]. In this research, instead of traditional SMC ABC, a SMC ABC replenishment algorithm [7] is utilized to estimate the parameters (algorithm 3). The likelihood function is a crucial component required when implementing certain algorithms. However, the likelihood function in this research is too complex to evaluate. Fortunately, [7] directly addresses the problem of intractable likelihoods.

---

**Algorithm 3: SMC ABC**

---

- 1 Set  $\alpha$ , the percentage of particles kept each iteration. Define target closeness (tolerance),  $\epsilon_T$
  - 2 Draw  $N$  particles from prior, simulate data and calculate the closeness. This obtains  $\{\mu_t, \delta_t, \nu_t, \rho_t\}_{t=1}^N$ .
  - 3 **While**  $\epsilon_N < \epsilon_T$  **do**
  - 4     Sort the particles increasingly by  $\rho_t$ , set tolerance,  $\epsilon_t = \rho_{N-\alpha N}$
  - 5     Select  $\alpha * N$  particles with lowest closeness
  - 6     Compute the tuning parameters of the MCMC kernel  $q_t(\cdot | \cdot)$
  - 7     **For**  $j = N - \alpha N + 1$  **to**  $N$  **do**
  - 8         **For**  $k = 1$  **to**  $S_t$  **do**
  - 9             Propose  $\theta^*$  from  $q_t(\cdot | \theta^j)$  and simulate data  $x$  according to  $\theta^*$
  - 10             Compute MH ratio  $r = \frac{\pi(\theta^*)q(\theta^j | \theta^*)}{\pi(\theta^j)q(\theta^* | \theta^j)} \mathbf{1}(\rho(x, y) < \epsilon_t)$
  - 11             **If**  $\text{unif}(0,1) < r$  **do**
  - 12                  $\theta^j = \theta^*$  and  $\rho^j = \rho(x, y)$
  - 13         **If** MCMC acceptance rate is too small
  - 14             **break**
- 

In this research, two different summary statistics are implemented with ABC rejection. The superior one will be used in MCMC ABC to gain a better understanding of the parameters. Subsequently, SMC ABC will be implemented to obtain the final result.

### 3. Parameter Estimation

#### ABC Rejection

ABC rejection is straightforward and easy to implement. As mentioned earlier, we applied two summary statistics and selected the better-performing one for use in ABC MCMC and SMC ABC. The number of corals and starfish is denoted as  $x = (x_1, x_2, \dots, x_{11})$  and  $y = (y_1, y_2, \dots, y_{11})$ . The simulated data is defined as  $\hat{x}$  and  $\hat{y}$ .

#### Euclidean distance

$$\sqrt{\|x - \hat{x}\| + \|y - \hat{y}\|}$$

#### Least squares method

$$\min (\|x - a\hat{x} - b\| + \|y - a\hat{y} - b\|)$$

Firstly, samples are drawn from a given prior, which is a uniform distribution between 0 and 1. In this research, the number of samples is set to be 100,000, denoted as  $\mu = (\mu_1, \mu_2, \dots, \mu_{100000})$ ,  $\delta = (\delta_1, \delta_2, \dots, \delta_3)$ ,  $\nu = (\nu_1, \nu_2, \dots, \nu_{100000})$ . These samples are used to simulated data through Gillespie Algorithm, producing a series of simulated data,  $\{x_i, y_i\}_{i=1}^{100000}$ . For each simulated data, we calculated the discrepancy,  $\epsilon_t$ , with observation data. Collect the samples and the discrepancies, we obtain particles,  $\{\mu_i, \delta_i, \nu_i, \epsilon_i\}_{i=1}^{100000}$ . The particles are sorted based on the discrepancy value,  $\epsilon_t$ . We return 1%, 0.5%, and 0.1% of parameters with lowest discrepancies. i.e., we drop 99%, 99.5% and 99.9% of the data from largest discrepancy,  $\epsilon_t$ .

#### ABC MCMC

Instead of proposing individual parameters,  $\mu, \delta$ , and  $\nu$ , a joint space,  $\theta = (\mu, \delta, \nu)$ , is used to propose samples at once. An initial state of the Markov Chain,  $(\mu_0, \delta_0, \nu_0)$ , is proposed from the prior distribution without additional information to ensure the diversity of samples. The MCMC kernel is set as a multivariate normal distribution with a covariance matrix,  $0.1 * I_{3 \times 3}$ . For each iteration, a candidate,  $\theta^*$ , is drawn from the multivariate normal distribution with the mean equal to the current state and the given covariance matrix. Subsequently, the candidate,  $\theta^*$ , is used to simulate data, and Euclidean distance is employed to represent the discrepancy,  $\psi^*$ , between the observation and simulated data using  $\theta^*$ . With a uniform prior and symmetric proposal distribution, the Metropolis-Hastings (MH) ratio simplifies to:

$$r = \frac{\psi^*}{\psi}$$

Following the MH algorithm, the candidate is accepted with probability  $r$ ; otherwise, we reject the candidate.

Note that there is a natural limitation on parameters,  $(\mu, \delta, \nu) \in [0, 1]^3$ . Therefore, if the candidate does not fall within this domain, the simulation and calculation of the MH ratio are skipped to minimise the computation waste.

## SMC ABC

In this algorithm, there are two new variables,  $\alpha$  and  $S_t$ , are introduced.  $\alpha$  represents the percentage of particles to be dropped at each iteration and is set to 0.5.  $S_t$  is the number of MCMC iterations and is set to 10000. Similar to MCMC ABC, we begin with samples drawn from the prior distribution and calculate the discrepancy between observations and simulated data, resulting in a set of particles  $\{\theta^i, \rho^i\}_{i=1}^{10000}$ . The particle set is sorted increasingly by the value of  $\rho$ . i.e.,  $\rho^i \leq \rho^{i+1}, \forall i$ . We drop 50% of the data with largest discrepancy. The MCMC kernel is adjusted based on the alive sample. In this case, our MCMC kernel is a multinomial distribution, so the mean and standard deviation of alive data are used to shift the MCMC kernel. The candidate is drawn from the shifted proposal distribution, then simulated, and the discrepancy calculated. The MH ratio is slightly different from MCMC ABC, defined as:

$$r = \frac{\pi(\theta^*)q(\theta^j|\theta^*)}{\pi(\theta^j)q(\theta^*|\theta^j)} 1(\rho(x, y) \leq \epsilon_t)$$

where  $\epsilon_t = \rho^{1000-\alpha*10000}$ . Since a symmetric multinomial distribution is used, the MH ratio is simplified as  $r = 1(\rho(x, y) \leq \epsilon_t)$ . We accept the candidate as the new sample with probability  $r$ ; otherwise, we draw a new candidate and repeat the process until we run out of MCMC trials. For each sample, the number of MCMC trials will be reset. Once we process all the samples, we obtain a new set of particles and repeat the process. Two stopping rules interrupt the algorithm. The first one is based on the value of  $\rho$ . If the discrepancy of all samples is less than the target tolerance,  $\epsilon_T$  (set as 170 in this research), we stop the algorithm. i.e., if  $\rho^i \leq \epsilon_T, \forall i$ , all particles satisfy the tolerance. The second stopping rule considers the overall MCMC acceptance rate. If the overall MCMC acceptance rate is too small, indicating the tolerance is too small, with a threshold set at 500 (i.e., if the iteration generates less than 10% of new samples), the algorithm is interrupted due to inefficiency.

## 4. Result

### ABC Rejection

#### Euclidean Distance

There are three thresholds that used to filter the data, as shown in Table 1.

| Threshold ( $\epsilon$ ) | Number of samples |
|--------------------------|-------------------|
| 225.6                    | 1000 (1%)         |
| 215.2                    | 500 (0.5%)        |
| 183.2                    | 100 (0.01%)       |

Table 1 illustrates the usage of three thresholds employed for data filtering.

As depicted in the Figure 2, several noteworthy features merit attention. For the first parameter,  $\mu$ , a significant spike is observed as the tolerance decreases. The same trend is observed in the second parameter,  $\delta$ . Fortunately, the excluded samples are situated at the tails, thereby not altering the overall trends in the plots. However, the third parameter,  $\nu$ , exhibits a contrasting trend compared to the first two parameters. As the number of samples decreases, the phenomenon of positive skewness diminishes. This implies that the samples at the tail simulate data that is more favorable when compared to the data at the center.

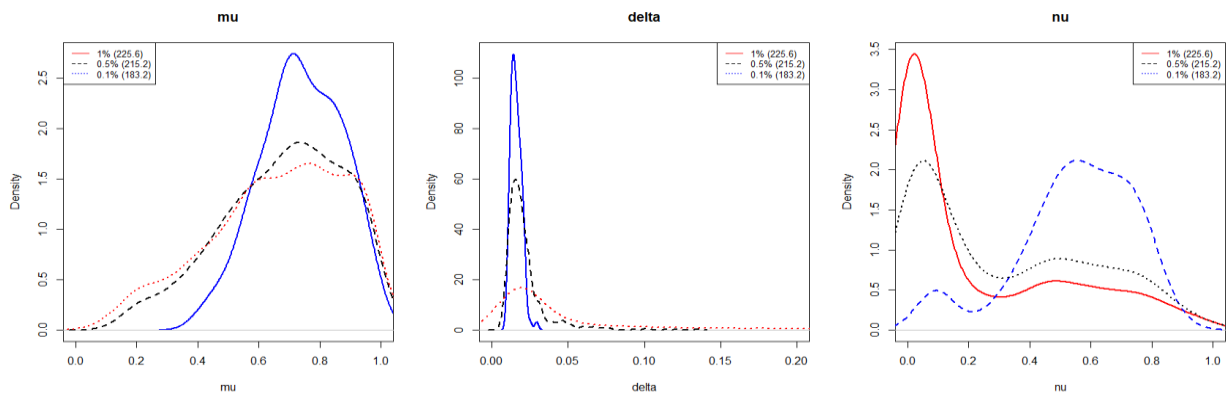


Figure 2: the density plot of 1%, 0.5% and 0.1% sample under Euclidean distance method.

### Least Square Error

The threshold used for data filtering differs from the Euclidean distance, as illustrated in Table 2. There is no significant difference between the two methods when filtering 1% and 0.1% of the data. However, a notable distinction emerges when employing the 0.1% data threshold, showing a significant difference.



| Threshold ( $\epsilon$ ) | Number of samples |
|--------------------------|-------------------|
| 224.5                    | 1000 (1%)         |
| 208.9                    | 500 (0.5%)        |
| 158.1                    | 100 (0.01%)       |

Table 2: the threshold used to select the samples – Least Squared Error.

Compared to the Euclidean distance method, the first and second parameters do not exhibit a significant difference (Figure 3). However, the distinctions between different subsets of samples are smaller, indicating that the exclusion of samples does not have a serious impact. As for the third parameter,  $\nu$ , the subsets of 1% and 0.5% samples are closer to the 0.1% subset. The excluded samples are situated at the tail of the distribution, mirroring the same phenomenon observed in  $\mu$  and  $\delta$ .

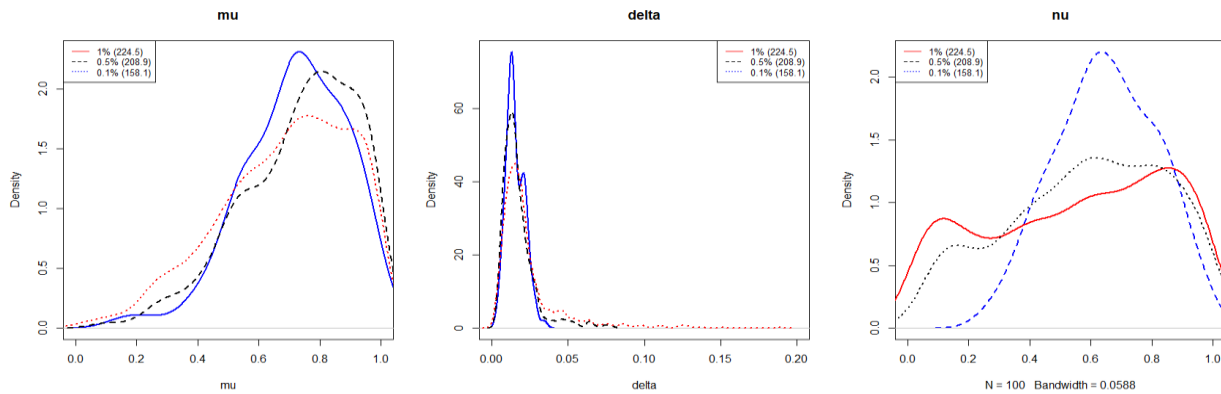


Figure 3: the density plot of 1%, 0.5% and 0.1% sample under least square error method.

When comparing these two different summary statistics methods, the main distinction lies in the third parameter,  $\nu$ . Despite both methods yielding similar results, we believe that the outcomes are correct. The first method necessitates the exclusion of more points than the second method, implying that more samples are relatively incorrect. One of the criteria for judging the effectiveness of a summary statistic is sensitivity. The summary statistic should maximize the difference in response to two datasets. The values obtained from the least square error method are concentrated in the range of 210 to 270, while the values from the Euclidean distance method range from 0 to 4000 (Figure 4). This evidence suggests that Euclidean distance is more sensitive than the least square error. Therefore, Euclidean distance is selected as the summary statistic for the next two methods.

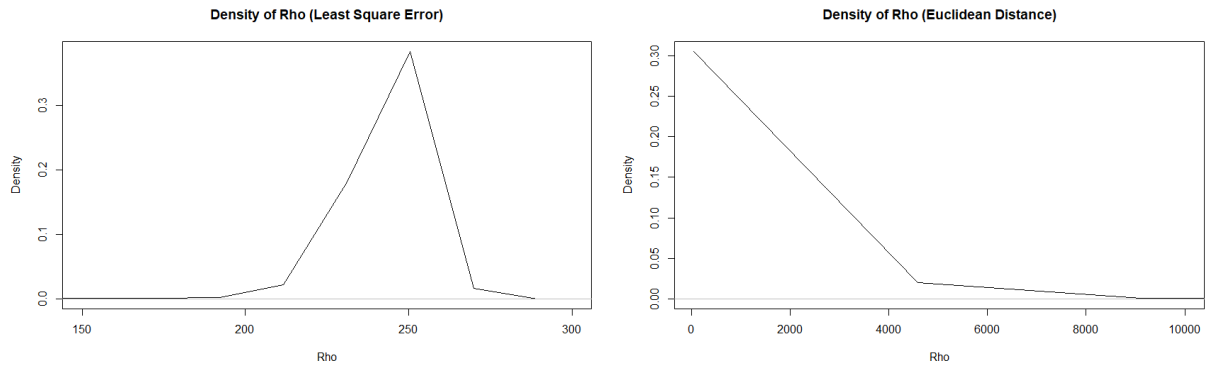


Figure 4, the density of closeness in different summary statistic.

## MCMC ABC

The results of MCMC ABC exhibit a roughly similar trend to ABC rejection. However, there are several noteworthy differences. For the first parameter,  $\mu$ , the left plot in Figure 5 displays a significant spike around 0.9, indicating that most of the samples are concentrated around this value. This is slightly higher than ABC rejection, which peaks at 0.7. In the case of the second parameter,  $\delta$ , the middle plot of Figure 5 shows two spikes. The value of ABC rejection is slightly higher than that of MCMC ABC, particularly around the second spike. The MCMC may have subtly shifted toward the same value. The third parameter,  $\nu$ , reveals multiple spikes in the figure 5, right plot, with values also higher than ABC rejection.

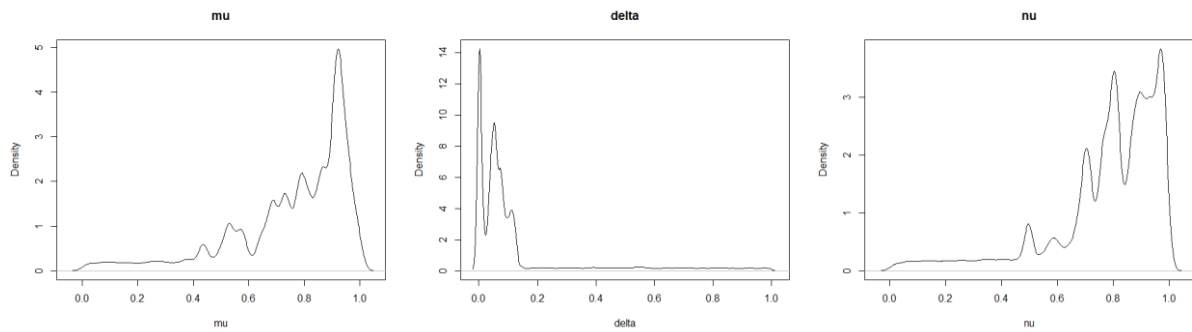


Figure 5: the density plot of sample generated by MCMC ABC

## SMC ABC

Implementing SMC ABC is more challenging compared to other algorithms, but the results are superior. All parameters exhibit a similar density compared to the 0.1% data obtained from ABC rejection. With a target tolerance of 170 and an MCMC acceptance rate limitation, generating fewer than 1000 new samples in each iteration, the algorithm requires 22 iterations to meet the stopping rule. In each iteration, half of the samples are dropped. The stopping criterion is met when there are fewer than 1000 new samples generated during the iteration. The details of each iteration are provided in the Appendix.

In the initial iterations, it is easy to identify samples that satisfy the tolerance. However, the execution time significantly increases as the tolerance decreases. All parameters start from 0.5. The density of samples in each iteration reveals that the algorithm initially operates in an incorrect area, especially for  $\delta$  and  $\nu$ . It then gradually shifts to the correct area as the tolerance decreases. The first parameter,  $\mu$ , quickly identifies the high-probability region. The second parameter,  $\delta$ , takes some time to locate the high-probability area. After 10 iterations, the density plot of  $\delta$  becomes stable, with no significant changes in further iterations. For the third parameter,  $\nu$ , the time required to reach the high-probability area is the longest. Initially dropping to a small value, almost 0, it starts exploring larger values after  $\mu$  and  $\delta$  have stabilized. During iterations 7, 8, 9, and 10, the tolerance does not change significantly, indicating that the newly generated samples are slightly better than the previous ones.

The final result (see Figure 6) closely resembles the outcome of 0.1% of the samples generated by ABC rejection. The birth event of coral,  $\mu$ , is expected to occur with a probability of 0.8 mostly. The predator event does not occur frequently, with a probability around 0.015. The probability of the death event for the starfish is around 0.5, significantly lower than the birth event for coral.

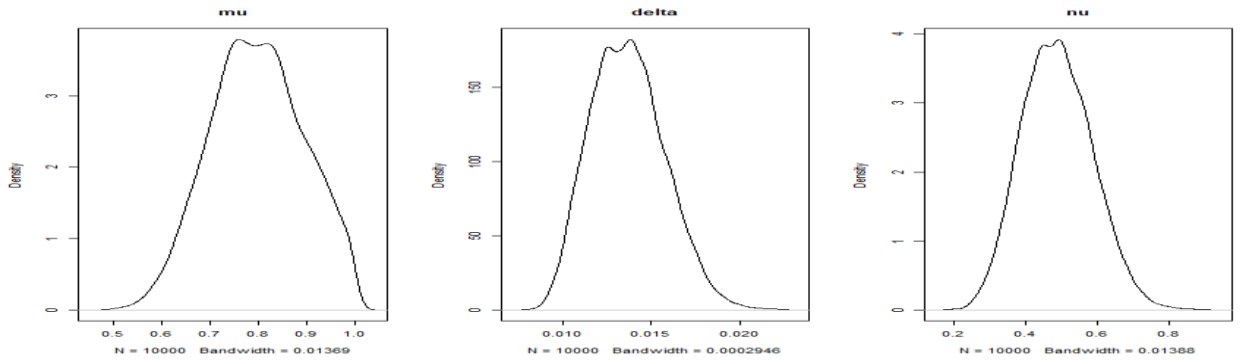


Figure 6: The density plot of final estimation of SMC ABC

## 5. Discussion

We first applied ABC rejection to identify a suitable summary statistic for comparing simulated data with observed data. The result revealed that Euclidean distance is more sensitive than least square error. Subsequently, a more efficient algorithm, MCMC ABC, was employed to obtain a set of more accurate samples. Finally, the results produced by MCMC ABC were used to define the target tolerance for SMC ABC, and SMC ABC was then employed to generate the final results. According to the findings, the birth event for coral occurs more frequently than the death event of the starfish. In comparison to these two events, the predator event rarely happens.

The SMC ABC algorithm is not efficient due to several reasons. Firstly, all parameters explore values lower than 0.5 at the beginning, which is the wrong area compared to the results.

Therefore, the use of a simple random walk as an MCMC kernel might not be the appropriate option, as the algorithm does not efficiently explore the parameter space. Considering the use of a different MCMC kernel might be able to improve the efficiency of SMC ABC.

Another reason that a simple random walk is not suitable for this problem is that the value of the parameter has an upper bound of 1 and a lower bound of 0. The proposed value via multinomial might not be in the parameter space. Using a different proposal distribution can solve the problem, such as a uniform distribution over 0 to 1 or a Beta distribution. The limitation of a uniform distribution is that the proposal cannot adjust based on alive particles. Due to the difficulty in estimating shape 1 and shape 2, a Beta distribution is also hard to adjust.

Thirdly, in this research, the number of MCMC trials is set as a constant. For each sample, there are  $S_t$  trials for it to explore a better value. It is interrupted once it finds a better value, where the data generated by the value is closer to the observed data. However, once the tolerance becomes smaller, it is not easy for the MCMC kernel to find a better value in the fixed number of trials. Therefore, the interrupt rule becomes useless.

In further research, increasing the efficiency of the MCMC kernel in exploring the parameter space is crucial, especially when the tolerance is low or the proposal distribution is close to the boundary, in this case, 0 and 1. Finding a solution to address the problem of candidates in the MCMC kernel that are out of the boundary can significantly improve efficiency.

## Acknowledgement

Firstly, we would like to express our gratitude to BagelBayes for providing the data for analysis. The observational data is crucial in approximate Bayesian computation. Secondly, we extend our thanks to Tasi for providing comprehensive knowledge of the measure space. Finally, a special thanks goes to our research supervisor, Dr. Matt Sutton. He has shared extensive knowledge of Bayesian statistics and Bayesian inference, and his support in building algorithms has been invaluable.

## Availability

Code availability: <https://github.com/fanbing1003/Approximate-Bayesian-Statistics->

# Appendix

| Iterations | Tolerance ( $\epsilon_t$ ) | New sample |
|------------|----------------------------|------------|
| 1          | 311.319771                 | 5000       |
| 2          | 298.715584                 | 5000       |
| 3          | 279.692688                 | 5000       |
| 4          | 258.038757                 | 5000       |
| 5          | 241.985537                 | 5000       |
| 6          | 233.190909                 | 5000       |
| 7          | 228.755328                 | 4873       |
| 8          | 226.808289                 | 4779       |
| 9          | 225.643524                 | 4772       |
| 10         | 223.452456                 | 4886       |
| 11         | 215.355520                 | 4396       |
| 12         | 204.733974                 | 3668       |
| 13         | 195.596523                 | 2358       |
| 14         | 189.765645                 | 2564       |
| 15         | 182.370502                 | 2785       |
| 16         | 168.457710                 | 2165       |
| 17         | 155.830036                 | 1825       |
| 18         | 145.801235                 | 1248       |
| 19         | 138.949631                 | 1087       |
| 20         | 133.405397                 | 1366       |
| 21         | 126.562238                 | 1122       |
| 22         | 121.453695                 | 906 (Stop) |

## References

- [1]. Narchi, N.E., Price, L.L. (2015). Introduction. In: Narchi, N., Price, L. (eds) *Ethnobiology of Corals and Coral Reefs*. Ethnobiology. Springer, Cham. [https://doi.org/10.1007/978-3-319-23763-3\\_1](https://doi.org/10.1007/978-3-319-23763-3_1)
- [2]. Santavy, D. L., Horstmann, C. L., Sharpe, L. M., Yee, S. H., & Ringold, P. (2021). What is it about coral reefs? Translation of ecosystem goods and services relevant to people and their well-being. *Ecosphere (Washington, D.C)*, 12(8), 1–27. <https://doi.org/10.1002/ecs2.3639>
- [3]. Cooper, E. L., Hirabayashi, K., Strychar, K. B., & Sammarco, P. W. (2014). Corals and their potential applications to integrative medicine. *Evidence-based complementary and alternative medicine : eCAM*, 2014, 184959. <https://doi.org/10.1155/2014/184959>
- [4]. Sunnåker, M., Busetto, A. G., Numminen, E., Corander, J., Foll, M., & Dessimoz, C. (2013). Approximate Bayesian computation. *PLoS computational biology*, 9(1), e1002803. <https://doi.org/10.1371/journal.pcbi.1002803>
- [5]. Marjoram P. (2013). Approximation Bayesian Computation. *OA genetics*, 1(3), 853. <https://doi.org/10.13172/2054-197x-1-1-853>
- [6]. Lintusaari, J., Gutmann, M. U., Dutta, R., Kaski, S., & Corander, J. (2017). Fundamentals and Recent Developments in Approximate Bayesian Computation. *Systematic biology*, 66(1), e66–e82. <https://doi.org/10.1093/sysbio/syw077>
- [7]. Drovandi, C. C., & Pettitt, A. N. (2011). Likelihood-free Bayesian estimation of multivariate quantile distributions. *Computational Statistics & Data Analysis*, 55(9), 2541–2556. <https://doi.org/10.1016/j.csda.2011.03.019>