UNIVERSITY OF TRIESTE

Department of
Mathematics, Geology and Informatics

Bachelor's Degree in
Artificial Intelligence & Data Analytics

Approximate Bayesian Computation applied to evolutionary dynamics

July 19, 2024

Candidate
Abdula Kalus

Supervisor
Prof. Giulio Caravagna

A.Y. 2023/2024

Every failure is a step to success.

- W. Whewell.

Abstract

In this thesis, we explored and used the Approximate Bayesian Computation (ABC) method for parameter estimation and inference in evolutionary models. Through a systematic progression from simple to complex scenarios, reflecting real-world complexities, we demonstrated the effectiveness and applicability of the ABC approach. By carefully selecting appropriate summary statistics and employing a rigorous methodology, we bridged the gap between theoretical assumptions and empirical observations. This robust framework allowed us to approximate the underlying probability distributions of the parameters under study, thereby enhancing our understanding of evolutionary dynamics. Additionally, our results highlight the versatility and adaptability of the ABC method in capturing the complexities of real-world systems. By advancing our methodological toolkit and refining our understanding of evolutionary processes, we are better equipped to tackle the challenges and intricacies of contemporary scientific investigation.

Contents

Introduction

This chapter aims toward giving an introduction to the problem area, the purpose of this thesis, and how the thesis is structured.

0.1 Background

Computational biology is a cornerstone of scientific exploration, focusing on how models derived from experimental data can illuminate the workings of biological systems. These models unravel the roles of specific genetic sequences, the genetic foundations of diseases, and the interactions within cellular environments. This field integrates a wide range of experimental data types, from molecular concentrations to cellular images, employing diverse mathematical and computational methodologies.

Central to computational biology is its ability to frame complex biomedical challenges as computational puzzles, requiring innovative and integrative approaches to construct models based on existing data. These models, while not always fully comprehensive, are invaluable tools for researchers, providing insights and guiding further experimentation in various biological domains.

Machine learning plays a pivotal role in computational biology, offering sophisticated techniques to analyze vast datasets and adaptively learn from new information. By leveraging statistical and computational methodologies, computational biologists continually develop and refine methods to address a wide range of biological inquiries, from analyzing protein structures to investigating disease spread dynamics. This discipline is poised to revolutionize our understanding of life's complexities [1].

The prominence of computational biology has increased due to its ability to handle DNA-based genetic information, which, like digital data, remains intact across generations. This digital medium offers evolutionary advantages, unlike analog signals susceptible to chemical diffusion. Computational methods enable effective visualization and modeling of biological data, aiding in the exploration of complex phenomena such as phylogenetic trees and cellular evolution [2].

Enhanced data processing capabilities have revolutionized biological data analysis. Machine learning techniques are integral to making inferences, classifying biological features, and identifying robust signals within datasets. Computational approaches facilitate unbiased correlation analyses, leading to conclusions that enrich biological knowledge and promote active learning. By integrating these methodologies, researchers can unravel the mysteries of life and understand evolutionary dynamics.

Evolutionary dynamics, a vital area in computational biology, explores mechanisms guiding genetic variation and selection in populations. It examines how mutation, selection, and genetic drift shape organism evolution, influencing adaptation and interactions. Understanding these dynamics is essential for addressing challenges in disease control and conservation biology. Researchers aim to uncover fundamental principles and develop predictive models to advance our understanding of evolution in both natural and artificial systems.

This thesis will delve into methods for studying and analyzing evolutionary dynamics, focusing on the interplay between mutation, selection, and genetic drift. Through computational simulations and statistical analyses, we aim to gain deeper insights into the mechanisms driving evolutionary change and adaptation.

0.2 Problem

The problem of evolutionary dynamics focuses on the mechanisms governing genetic variation and selection within populations. Evolution relies on the generation of diversity, primarily through mutations, which introduce new genetic variants.

Mutations Mutations are changes in the genetic sequence of an organism. They can occur spontaneously during DNA replication or be induced by external factors such as radiation or chemicals. Mutations are the primary source of genetic diversity, providing the raw material for evolution. They can be beneficial, neutral, or deleterious, depending on their impact on the organism's fitness. Beneficial mutations may provide a survival advantage, allowing individuals to adapt to changing environments and ultimately driving evolutionary progress.

These genetic variants undergo selection, with advantageous traits being favored and less fit individuals being eliminated. Additionally, genetic drift, a random process, can influence the frequency of gene variants within populations, potentially reducing genetic diversity over time.

Genetic Drift Genetic drift is a random process that influences the frequency of gene variants within populations. Unlike natural selection, which is deterministic and driven by fitness differences, genetic drift is stochastic and can cause significant fluctuations in allele frequencies purely by chance. This randomness can lead to the loss of genetic variants, potentially reducing genetic diversity over time, especially in small populations. Genetic drift can also lead to the fixation of neutral or even deleterious alleles, impacting the evolutionary trajectory in unpredictable ways.

These processes — mutation, selection, and drift — interact to shape the evolutionary trajectory of organisms. Microorganisms like viruses exhibit rapid evolutionary changes due to factors such as high mutation rates and population sizes.

The interplay between these mechanisms can lead to significant outcomes, including the emergence of public health challenges. Notably, the host immune response plays a crucial role in driving evolutionary changes, as pathogens that can evade immune defenses gain a fitness advantage. This continuous evolution and adaptation are evident in diseases like influenza and other viruses. Furthermore, the study of recurring mutations and factors like mutation rates adds depth to our understanding of evolutionary dynamics [3].

0.3 Purpose

The purpose of this thesis is to analyze evolutionary dynamics using a statistical method known as Approximate Bayesian Computation (ABC). In this study, artificial data will be generated to simulate evolutionary processes. ABC is a powerful tool for studying complex evolutionary systems by comparing simulated data to observed data and estimating parameters that best fit the observed patterns. By employing ABC, this thesis aims to gain insights into the mechanisms driving evolutionary change, including mutation, selection, and genetic drift.

The artificial data will allow for controlled experiments, facilitating the exploration of various evolutionary scenarios and the assessment of different factors influencing evolutionary dynamics. Through this approach, the thesis seeks to enhance our understanding of the evolutionary processes that shape biological diversity and inform strategies for addressing practical challenges in fields such as public health and conservation biology.

0.4 Methodology

The approach for this thesis project is to use an algorithm called Approximate Bayesian Computation (ABC) to perform the analysis and the employment of the rRACES tool, which is used to generate the synthetic data required for the analysis. In Figure 1 can be seen the detailed pipeline of rRACES.

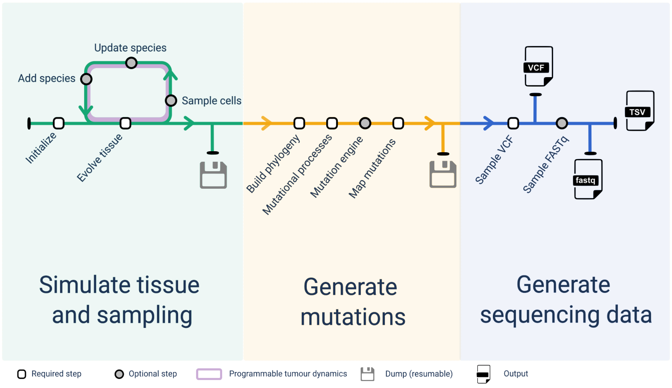


Figure 1: rRACES detailed pipeline

0.4.1 rRACES

rRACES, developed by Giulio Caravagna and Alberto Casagrande [4], allows for the initialization of a squared tissue, where a species is defined and single cells are placed onto it. The tissue then evolves, with all cells growing stochastically over time. During this process, some cells are sampled from the tissue to mimic measurements. The simulation can be repeated, allowing for the updating of parameters for existing species and the addition of new species. This type of simulation replicates evolutionary dynamics, providing insights into the emergence and adaptation of species within a population.

From the sampled cells, a phylogenetic tree reflecting the evolutionary history of the simulated population can be constructed. Additionally, mutational processes can be mapped onto the temporal evolution of the process. These mutational processes are generated by a mutational engine, which simulates mutations using a real reference genome. Once created, mutations are stochastically attached to the simulated phylogeny, reflecting the genetic changes occurring over time. Through the integration of these features, rRACES offers a comprehensive platform for studying evolutionary dynamics and generating artificial data for analysis using Approximate Bayesian Computation (ABC).

Code example

To simulate a tissue the following steps are required:

1. creation of a tissue;
2. introduction of cells in the tissue;
3. actual simulation.

So, to perform a simulation a new object of class **Simulation** must be created. Then in order to simulate the evolution of some species we need to add them to the simulation object created before. This process defines the evolutionary parameters of the species.

A *mutant* is a set of cells having the same (potentially unknown) driver mutations. Cells in the same mutant can have different liveness rates due to different epigenetic states.

A *species* is a mutant with an optional epigenetic state. At this point in the simulation, the mutant is just a name (A, B, ..) that, at a later stage could be linked to mutations of interest. The epigenome is a binary feature of a species that is represented by epistates +/- (positive and negative status). This is an abstraction, and could represent an active/inactive state linked to a promoter methylation or, more broadly, a phenotype. The evolution of mutants is non-reversible (no-back mutations model), while the evolution among epistates is potentially reversible.

The following code shows an example where we define two mutants, A and B, each with their epistates +/-. We have four distinct species: A+ and A-, as well as B+ and B-.

```r
# Default constructor
sim <- new(Simulation)

sim$add_mutant(name = "A",
epigenetic_rates = c("+-" = 0.01, "-+" = 0.01),
growth_rates = c("+" = 0.2, "-" = 0.08),
death_rates = c("+" = 0.1, "-" = 0.01))
```

Listing 1: script defining two mutants A and B with epigenetic states +/-.

Alternatively, if we don't want a mutant with epigenetic states, a mutant without epistates could be added as well.

```

# Default constructor
sim <- new(Simulation)
sim$add_mutant(name = "A", growth_rates = 0.2, death_rates =  0.1)
```

Listing 2: script defining mutant A and without epigenetic states.

The most common way for tissue to evolve is to take the current simulation clock as a reference and simulate further.

```
# Get the simulation clock
sim$get_clock()
#> [1] 84.48365

# Run the simulation for other 15 time units
sim$run_up_to_time(sim$get_clock() + 15)

# Get again the simulation clock
sim$get_clock()
#> [1] 99.49105
```

Listing 3: Script for tissue evolution

0.4.2 ABC

ABC will be pivotal for the experiments outlined later in the thesis. However, before delving into the experiments, it is essential to introduce some fundamental concepts regarding Bayesian inference.

Bayesian inference Bayesian inference is a way of making statistical inferences. The statistician assigns subjective probabilities to the distributions that could generate the data. These subjective probabilities form the prior distribution. After observing the data, Bayes' Theorem is used to update the prior. This revision adjusts the probabilities assigned to the possible data-generating distributions. The updated probabilities form the posterior distribution.

One key aspect of the Bayesian approach is the evaluation of a likelihood function. However, in some circumstances, the likelihood function is intractable or not available in a closed form. This is especially true in evolutionary dynamics, where the models are very complex and have many degrees of freedom.

For example let's consider the following equations

$$\frac{dN_+}{dt} = N_+w_+ + \lambda_{-+}N_- - \lambda_{+-}N_+ \tag{0.4.1}$$

$$\frac{dN_-}{dt} = N_-w_- - \lambda_{-+}N_- - \lambda_{+-}N_+ \tag{0.4.2}$$

where the first equation 0.4.1 is the population of cell with epistate + and equation 0.4.2 the population with epistate − and the total population is the sum of the two previous equation N, where $N = N_+ + N_-$.

Let's say now that we want to learn the growth rate of the two population so $w_{+/-}$. To do this, we need to know the state-change rates, $\lambda_{+-/-+}$. If we have the values of λ, we can solve the equation. Without λ, we face a system with two equations and four variables. However, by using an analytical method, we can perform simulations to try and explain the phenomenon.

In the Bayesian method, the posterior probability density $P(\theta \mid x)$ given observed data **x** and a model $\mathscr{M}$, can be computed using Bayes' Theorem

$$p(\theta \mid x) = \frac{\mathscr{L}(x \mid \theta)p(\theta)}{\int_\theta \mathscr{L}(x \mid \theta)p(\theta)d\theta} \propto \mathscr{L}(x \mid \theta)p(\theta) \tag{0.4.3}$$

where $p(\theta)$ is the prior probability of θ and $\mathscr{L}(x \mid \theta)$ is the likelihood function. The denominator is a normalizing constant.

However, as mentioned earlier, explicit forms for likelihood functions are rarely available. The ABC methods approximate the likelihood by evaluating the discrepancy between the observed data and the data generated by a simulation using a given model, yielding an approximate form of Bayes' Theorem:

$$p(\theta \mid \Delta(x, x^*) < \epsilon) \propto p(\theta)p(\Delta(x, x^*) < \epsilon \mid \theta) \tag{0.4.4}$$

where $x^* \sim f(\cdot \mid \theta)$ are the simulated data, $\Delta(\cdot)$ is a discrepancy metric, and $\epsilon > 0$ is a tolerance threshold (when ϵ tends towards 0, the approximated posterior distribution is a good approximation of the true posterior distribution) [5].

ABC pipeline The ABC analysis will be conducted using EasyABC, a software package developed for the R platform [6].

It functions as follows: suppose that we want to compute the posterior probability distribution of a univariate or multivariate parameter, θ.

1. A parameter value θ_i is sampled from its prior distribution to simulate a dataset y_i, for $i = 1, \ldots, n$, where n is the number of simulations.

2. A set of summary statistics $S(y_i)$ is computed from the simulated data and compared to the summary statistics obtained from the actual data $S(y_0)$ using a distance measure d. We consider the Euclidean distance for d.

3. If $d(S(y_i), S(y_0))$ (i.e., the distance between $S(y_i)$ and $S(y_0)$) is less than a given threshold, the parameter value θ_i is accepted.

4. In order to set a threshold above which simulations are rejected, the user has to provide the tolerance rate, which is defined as the percentage of accepted simulations.

5. The accepted θ_i's form a sample from an approximation of the posterior distribution.

Assessing parameter quality relies heavily on comparing true and evaluated summary statistics. This comparison significantly influences test outcomes, as discussed later. Therefore, selecting tailored summary statistics is crucial.

Choosing a prior distribution also demands careful consideration. Two types are recognized: informative and uninformative. An informative prior offers precise details about a variable, while an uninformative or diffuse prior provides general information. Here, the variable of interest is the evaluated parameter.

The EasyABC package simplifies implementing various ABC schemes and retrieving simulation outputs for post-processing with R tools. Predominant ABC schemes include the standard rejection algorithm and MCMC sequential schemes.

The standard rejection algorithm involves drawing model parameters from priors, using these values for simulations, and repeating to select simulations closest to the target (or within a tolerance threshold) based on summary statistics to approximate posterior distributions.

In contrast, ABC-MCMC algorithms apply Metropolis-Hastings to explore parameter space, using simulations rather than likelihood ratios for model computation.

MCMC Briefly here we give the general idea of how the MCMC work. An MCMC algorithm uses a Markov chain to generate sequences of samples. A Markov chain is a sequence of events in which the probability of an event depends only on the current state and not on the history of previous events. Therefore, a Markov process can be defined as memorylessness.

Markov Chain Monte Carlo (MCMC) Algorithm:

1. Start from a random point in the variable space.
2. Use a state transition based on a probability distribution to move to a new point in the variable space. This sampling process continues, generating a sequence of points as samples from the distribution of interest.
3. The goal is for the Markov chain to converge to the desired probability distribution. After sufficient iterations, sampled points should resemble samples drawn directly from the target distribution.
4. **Metropolis-Hastings Algorithm for State Transition:**
    (a) The most common acceptance and rejection method used for state transition in an MCMC algorithm is the Metropolis-Hastings algorithm.
    (b) The algorithm proposes a new position in the variable space.
    (c) Calculate an acceptance ratio based on the target probability distribution.
    (d) If the proposed point is more probable than the current point, accept it unconditionally. Otherwise, accept it with a probability proportional to the ratio of probabilities between the new point and the current point.

Metropolis-Hastings Acceptance Ratio Equation:
The acceptance ratio α for the Metropolis-Hastings algorithm is computed as:

$$\alpha = \min\left(1, \frac{p(\theta_{\text{prop}}|y)}{p(\theta_{\text{curr}}|y)} \cdot \frac{q(\theta_{\text{curr}}|\theta_{\text{prop}})}{q(\theta_{\text{prop}}|\theta_{\text{curr}})}\right)$$

where:

- $p(\theta|y)$ denotes the target posterior distribution of parameter θ given observed data y.
- $q(\theta_{\text{curr}}|\theta_{\text{prop}})$ is the proposal distribution for transitioning from θ_{prop} to θ_{curr}.

Code example

Toy model We here consider a very simple stochastic model coded in the R language. We will use two different types of prior distribution for the two model parameters ($x[1]$ and $x[2]$): a uniform distribution between 0 and 1 and a normal distribution with mean 1 and standard deviation 2, and we will consider an imaginary dataset of two summary statistics that the toy_model is aiming at fitting $sum_stat_obs = c(1.5, 0.5)$.

```

toy <- function(x){
  c( x[1] + x[2] + rnorm(1,0,0.1) , x[1] * x[2] + rnorm(1,0,0.1) )
}

toy_prior=list(c("unif",0,1),c("normal",1,2))

#Output of toy prior
[[1]]
[1] "unif" "0" "1"
[[2]]
[1] "normal" "1" "2"

sum_stat_obs=c(1.5,0.5)
```

Listing 4: Model definition

Performing a standard ABC-rejection procedure

A standard ABC-rejection procedure can be simply performed with the function **ABC_rejection**, in precising the number n of simulations to be performed and the proportion of simulations which are to be retained tr:

```
set.seed(1)

n= 10

tr = 0.2

ABC_rej <- ABC_rejection(model=toy,
             prior=toy_prior,
             nb_simul=n,
             summary_stat_target=sum_stat_obs,
             tol=tr)



$param
[,1] [,2]
param 0.6927316 0.8877425
param 0.3162717 1.0934523

$stats
[,1] [,2]
[1,] 1.564895 0.4678920
[2,] 1.386153 0.2915392

$nsim
[1] 10
```

```

$nrec
[1] 2
```

Listing 5: ABC_rejection procedure

Chapter 1

Single Clone Growth Rate

1.1 Case study

In this chapter, the focus shifts towards examining the growth rate of a singular mutant, denoted as A, within a tissue. Figure 1.1 illustrates the evolutionary dynamics of our case study, providing insights into the temporal abundance of mutant A. The depicted trajectory spans from an initial time $t_0 = 0$, characterized one cell, to a subsequent time t_f, wherein the total count of clones reaches n_A cells. Returning to our case study, our objective is to examine the growth rate (α) governing the reproduction of mutant A.

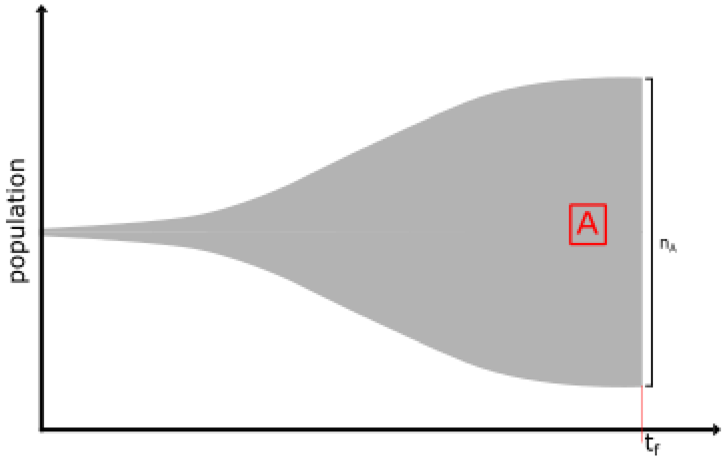


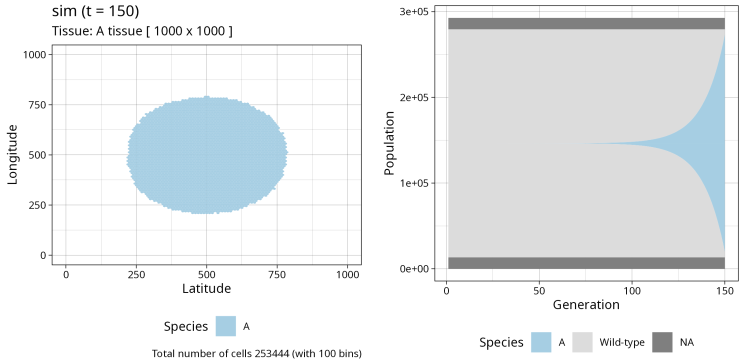
Figure 1.1: Illustration of the evolutionary dynamics of a single clone.

1.2 The model that generates the data

The initial task entails model creation using rRACES to replicate the previously described evolutionary dynamics. The model generated by rRACES, denoted as $\mathscr{M}(\alpha, \beta)$, simulates the evolution of mutant A and the parameters for the experiment are initialized as follows.

- (growth rate) $\alpha = 0.086$
- (death rate) $\beta = 0$

Considering β being zero, the model can be expressed as dependent on a single parameter, denoted as $\mathscr{M}(\alpha)$.
At 150 time units, Figure 1.2 displays both the tissue and the corresponding Muller plot.



(a) Tissue plot

(b) Muller plot

From the model $\mathscr{M}$, as the tissue undergoes stochastic growth, a selection of cells is sampled to mimic measurements, providing crucial information. These measurements yield summary statistics used to assess the ABC simulation's accuracy. In this scenario, the model $\mathscr{M}$ initiates at time $t_0 = 0$ with 1 cell. Eventually, at time t_f, the entire tissue is sampled, allowing the determination of the cells count $n_{A_{tf}}$. This population measure at time t_f serves as the summary statistic for the ABC simulation, defined as $S(y_0) = n_{A_{t_f}}$.

1.3 ABC Analysis

Utilizing the model $\mathscr{M}$ and the summary statistic $S(y_0)$, the ABC simulation can be executed, first employing the ABC rejection algorithm, followed by the ABC MCMC algorithm.

1.3.1 ABC rejection

The ABC rejection analysis was conducted for 6000 times, with only 5% of these simulations being accepted, resulting in 300 accepted values. The values used for the simulations were sampled from a uniform distribution representing an uninformative prior, $\theta \sim U(0, 0.1)$. Figure 1.2 shows a plot that allows the evaluation of the quality of estimation when the ABC rejection methods is used.

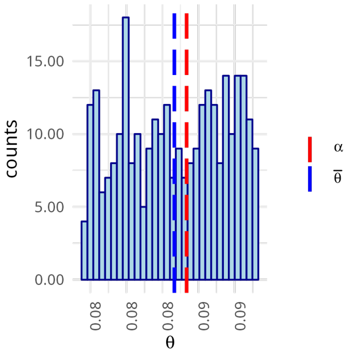
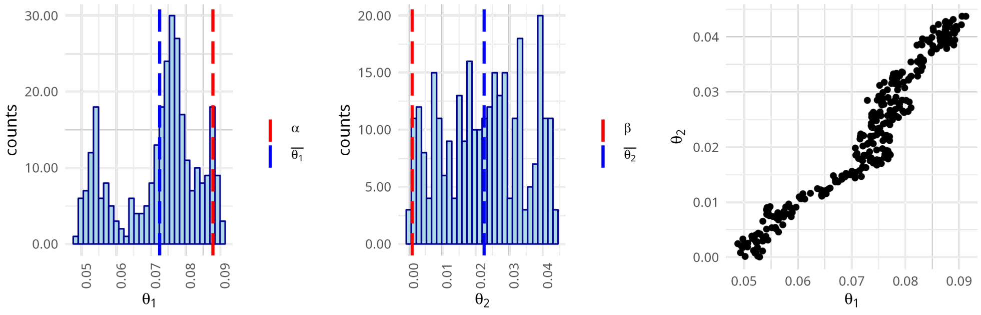


Figure 1.2: Distribution of the θ values obtained after performing the ABC analysis. The red dashed vertical line indicates the true value α that generated the original data, while the blue dashed vertical line indicates the mean value of θ, denoted as $\overline{\theta}$.

Figure 1.2 shows that the posterior distribution is very different from the prior distribution, which we hypothesized to be uniform, confirming that the summary statistic convey information about the evolutionary dynamics we are studying. This plot confirms that the summary statistic convey information about α, because the distances corresponding to the accepted values are clustered and not spread around the prior range ofα, in fact, the accepted values of the simulations fall within the range $[\overline{\theta} - \epsilon_1, \overline{\theta} + \epsilon_1]$, where $\alpha \in [\overline{\theta} - \epsilon_1, \overline{\theta} + \epsilon_1]$. The mean value $\overline{\theta}$ closely aligns with the actual parameter value α, indicating the success of the simulation.

An intriguing variation of this case study involves altering the initial conditions by introducing a nonzero death rate (β). For this experiment, the death rate is set to 0.01 ($\alpha = 0.089$, $\beta = 0.01$). Another ABC rejection analysis is conducted for 6000 times, using the prior distributions $\alpha \sim U(0, 1)$ and $\beta \sim U(0, 0.5)$, and accepting 5% of the simulations.

Figure 1.3 shows a notably wider accepted range for α compared to previous observations. This expansion is linked to the correlation between α and β in this context.



(a) accepted value for α (b) accepted value for β (c) α vs β

Figure 1.3: Distribution of θ_1 and θ_2 obtained after performing the ABC analysis. The red dashed vertical line indicates the true value α for plot (a) and the true value β for plot (b) that generated the original data, while the blue dashed vertical line indicates the mean values of θ_1 and θ_2, denoted as $\overline{\theta}_i$. Plot (c) shows the correlation between the two parameters.

The figure (c)1.3 illustrates a linear correlation between α and β, indicating a requisite balance between these parameters. In particular, to have that at time t_f the summary statistic is close to the observed one, $S(y_0)$, the two parameter must balance each other.

1.3.2 ABC-MCMC

Conducting the ABC analysis utilizing the MCMC scheme, a more complex algorithm, results in the acceptance of only a limited number of parameters. Additionally, the epsilon value in this instance is reduced compared to the previous iteration. While this algorithm demonstrates greater efficiency, it comes at the expense of increased computational overhead. Specifically, this simulation required twice the duration, with half the number of simulations executed compared to the previous approach. The outcomes of this simulation are depicted in Figure 1.4.

In the scenario where $\beta \neq 0$ is introduced, a focused approach within a particular region of the parameter space is employed. However, this strategy yields an erroneous estimation, as depicted in Figure 1.5. Here, the estimated values of θ_1 and θ_2 result in identical summary statistics, effectively canceling each other out.

This chapter can be concluded with the following considerations: both methods provide a good approximation of the parameter, with the assertion that MCMC

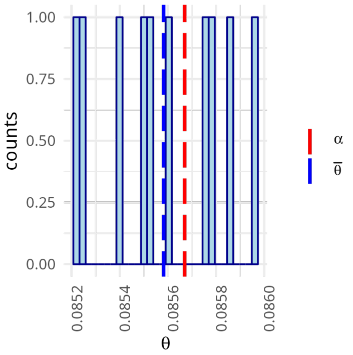
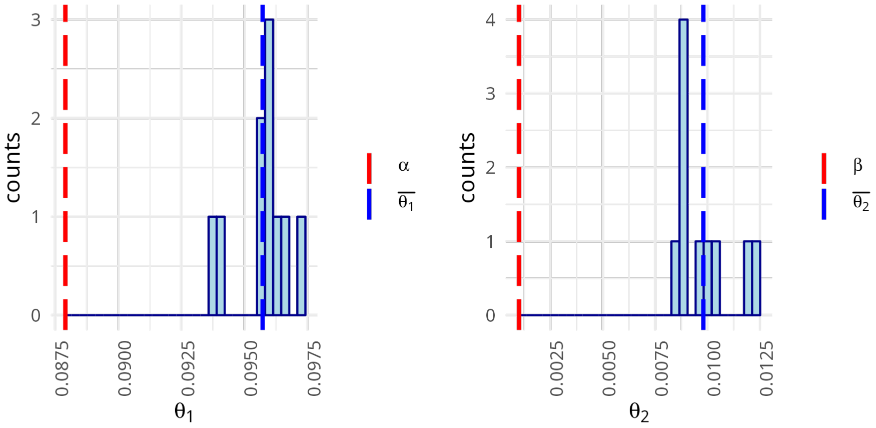


Figure 1.4: Distribution of the θ values obtained after performing the ABC analysis. The red dashed vertical line indicates the true value α that generated the original data, while the blue dashed vertical line indicates the mean value of θ, denoted as $\overline{\theta}$.



(a) accepted value for α

(b) accepted value for β

Figure 1.5: Distribution of θ_1 and θ_2 obtained after performing the ABC analysis. The red dashed vertical line indicates the true value α for plot (a) and the true value β for plot (b) that generated the original data, while the blue dashed vertical line indicates the mean values of θ_1 and θ_2, denoted as $\overline{\theta}_i$.

is more precise. However, if the problem is ill-posed and too much freedom and ambiguity is allowed in the model, a solution cannot be found, this is because the choice of summary statistics cannot be made properly.

Chapter 2

Subclone B Emergence in Clone A Expansion

2.1 Case study

This chapter focuses on the emergence of subclone B during the expansion of primary clone A. Originating from a mutation in A, subclone B gains a fitness advantage, allowing it to coexist with A. The experiment aims to determine when the subclone B appear and with which growth rate. Additionally, this chapter highlights the importance of selecting appropriate summary statistics. For this case study, the ABC rejection scheme is exclusively employed due to its computational efficiency and reliable performance, particularly compared to MCMC. Figure 2.1 illustrates the evolutionary dynamics of the case study.

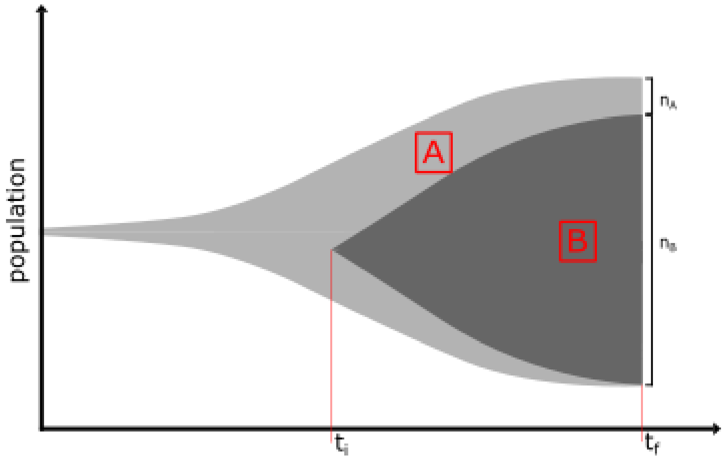
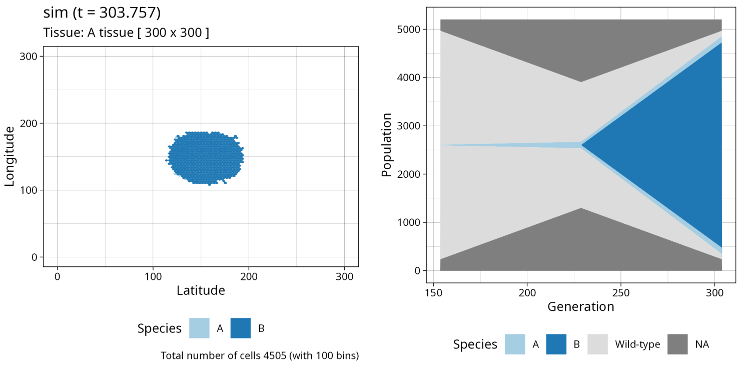


Figure 2.1: Illustration of the evolutionary dynamics of subclone B emerging during the expansion of clone A.

2.2 The model that generates the data

Similarly to the previous chapter, the model $\mathscr{M}(\gamma_A, \gamma_B)$ is defined, where γ_A and γ_B represent the growth rates of clone A and subclone B (note that $\gamma = \alpha - \beta$). In Figure 2.2, can observed how the tissue evolves after t units of time, noting that subclone B has a high fitness.



(a) Tissue plot

(b) Muller plot

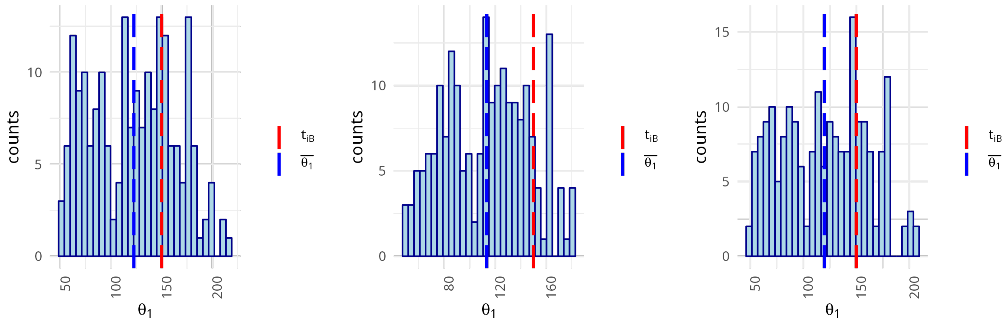
In the model $\mathscr{M}$, as the tissue undergoes stochastic growth, a subset of cells is sampled to simulate measurements, thus yielding informative data. These measurements are used to create the summary statistics that will be employed for the ABC analysis. Later on, the effectiveness of choosing appropriate summary statistics will be further explored.

2.3 ABC Analysis

As anticipated at the beginning of this chapter, only the ABC rejection scheme will be utilized. Given its simpler algorithm, it requires less computational capacity and therefore less time to execute the analysis.

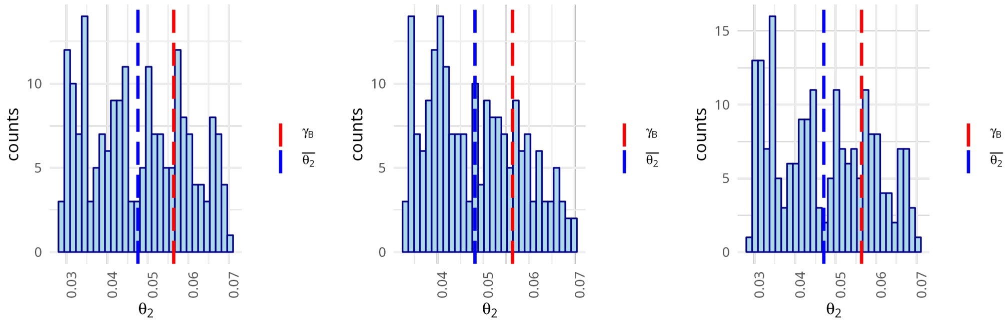
Firstly, will be presented three ABC analyses, each with 3000 simulations and uninformative priors. The primary distinction among these analyses lies in the selection of summary statistics utilized to either accept or reject proposed parameter values. The summary statistics for the three analyses are, respectively, the number of cells of A and B for the first, the percentage of population B for the second, and the ratio between the population of B and A for the third.

Figure 2.3 shows that the posterior distribution is very different from the prior distribution, in fact, the plots show that choosing a value for the parameters does not lead to the desired solution equiprobably. This is because these plots confirm



(c) # cell A & # cell B (d) $B\%$ (e) B/A

Figure 2.2: Comparison of the different distributions of θ_1 obtained after performing the ABC analysis with three different summary statistics. The red dashed vertical line indicates the true value t that generated the original data, while the blue dashed vertical line indicates the mean value of θ_1, denoted as $\overline{\theta}_1$.



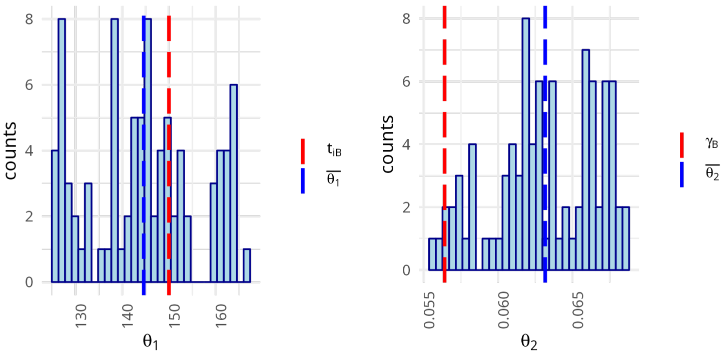
(a) # cell A & # cell B (b) $B\%$ (c) B/A

Figure 2.3: Comparison of the different distributions of θ_2 obtained after performing the ABC analysis with three different summary statistics. The red dashed vertical line indicates the true value γ_B that generated the original data, while the blue dashed vertical line indicates the mean value of θ_2, denoted as $\overline{\theta}_2$.

that the summary statistics don't have enough information about the evolutionary dynamics considered.

As seen from the previous plots, the accepted range of values is very wide for both θ_1 and θ_2, providing a vague approximation of where the true value might lie. This reasoning applies to all three ABC analyses, implying that the adopted summary statistics do not adequately explain the phenomenon.

The reason behind the earlier results lies in the fact that the appearance time of B seems to be correlated with its growth rate, so because the statistics don't explain enough the phenomenon. However, by selecting appropriate summary statistics, such as the percentage of B at time $t_x > t_0^B$ and the rate of B's variation after τ time units, a significant improvement can be noted. Indeed, when using these summary statistics, it becomes essential for the considered $theta_2$ value to be very close to the true growth rate, to be accepted. In other words, if this isn't the case, the simulated rate of change won't correspond to the observed one. In figure 2.4 can be seen the result of the ABC analysis with the new summary statistics.



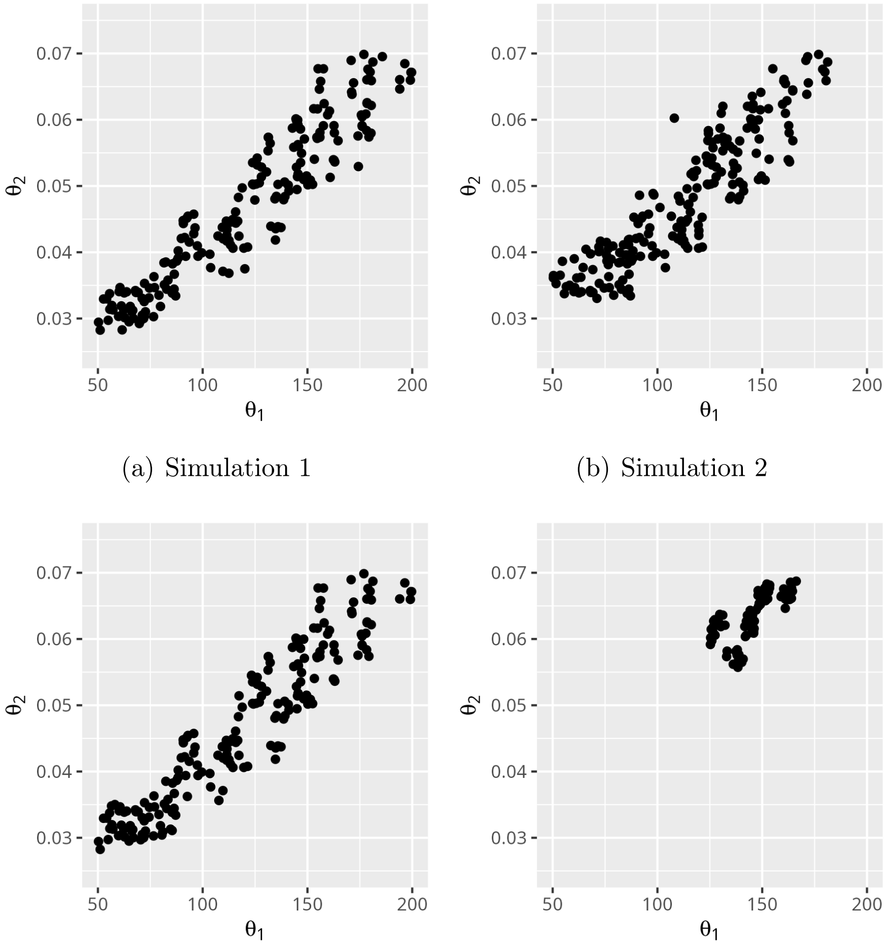
(a) θ_1

(b) θ_2

Figure 2.4: Distribution of θ_1 and θ_2 obtained after performing the ABC analysis with the adjusted summary statistics. The red dashed vertical line indicates the true value t for plot (a) and the true value γ_B for plot (b) that generated the original data, while the blue dashed vertical line indicates the mean values of θ_1 and θ_2, denoted as $\overline{\theta}_i$.

As seen in Figure 2.4, the accuracy of the estimation has significantly improved for both parameters, yielding values that fall within a much smaller range compared to before, thereby reducing the parameter space to focus on.

The reduction of hypothesis space can be observed in Figure 2.5. We compare the four simulations proposed in this chapter through a scatter plot. By graphically visualizing θ_1 against θ_2, we can immediately notice the correlation between the two parameters: as one increases, so does the other. That because later in time subclone B emerges, more rapidly it need to grow to match with the summary statistics at the sampling. However, what is particularly interesting to observe in these plots is the graph of the fourth simulation, where we can see that the proposed parameter range concentrates on a specific portion of the plane, indicating that the true value likely lies there.



(a) Simulation 1

(b) Simulation 2

(c) Simulation 3

(d) Simulation 4

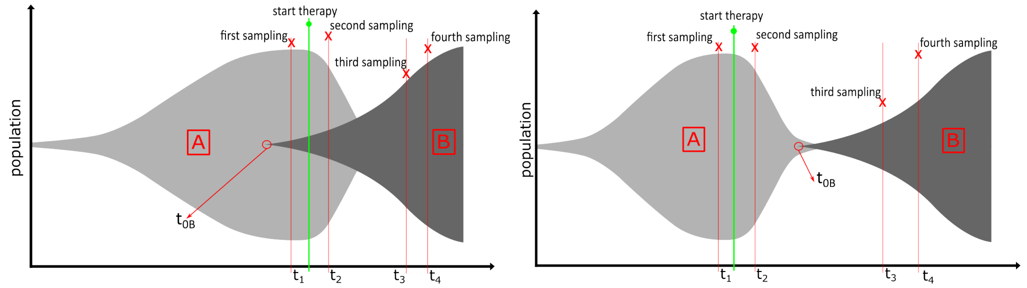
Figure 2.5: Comparison of the parameter space θ_1 and θ_2 across the four experiments; it can be observed that in the last graph on the right, there is a significant reduction in uncertainty.

Chapter 3

Practical Application: Undergoing drug resistance

3.1 Caste study

We began with the simplest case of evolutionary dynamics, a species on a evolving tissue, and successfully inferred the species' growth rate, mastering the employment of ABC analysis while comprehending its efficacy and potency. The subsequent step entailed understanding the significance of summary statistics, which wield a substantial impact on the outcomes yielded by an ABC analysis. In this chapter, we endeavor to apply the acquired insights to a pseudo-real case study.



(a) pre-existing

(b) de-novo

Figure 3.1: Illustration of the case study

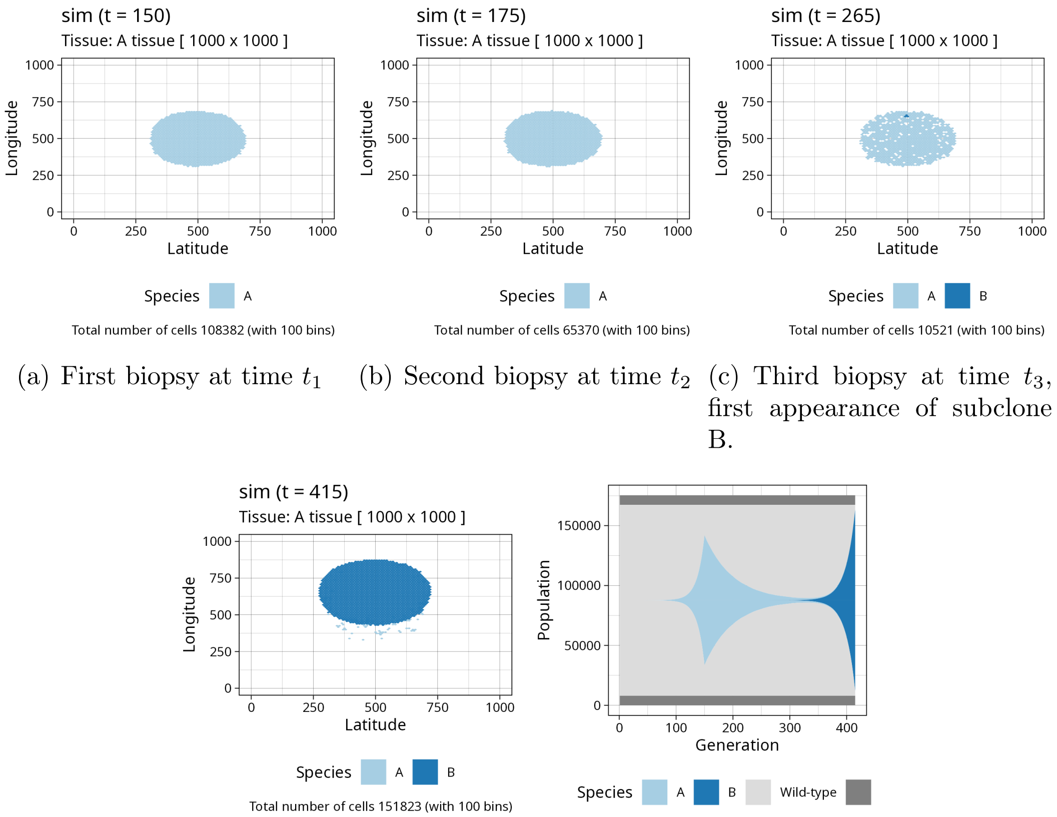
The case study is as follows: Let's consider a patient who undergoes a biopsy revealing the presence of malignant species A on the tissue. Consequently, a therapy is started to suppress this species. However, it's possible that before (or after) the therapy, a subclone B of A develops, resistant to the treatment. The aim is to investigate and comprehend whether B is 'pre-existing', so before the therapy or 'de-novo', so emerges after. Additionally, there is an interest in understanding its growth rate. Figure 3.1 shows an illustration depicting the case study with the two variants.

3.2 Preliminary Analysis and Preparation

As depicted in Figure 3.1, whether it be the 'pre-existing' or 'de-novo' scenario, it is imperative to monitor and conduct at least four samplings during the evolutionary process: before the starting of the therapy, after the starting of the therapy, and finally, two samplings at short intervals after a certain duration of time. The initial two measurements are needed to assess the success of the therapy, whereas the latter two samplings are utilized to monitor the growth of B. This information proves instrumental in selecting the summary statistics to be employed for the ABC analysis.

But first, let's begin with a brief explanation of how the model was created to generate this type of data: initially, we always start with the $\mathscr{M}(\gamma_A, \gamma_B)$ model. We randomly choose whether to develop the 'pre-existing' or 'de-novo' scenario and then evolve the tissue. At a certain point, we update the death rate of A to simulate the effect of therapy and induce the death of A cells. Subsequently, tissue growth resumes, and data are sampled as previously explained. In figure 3.2 can be seen the evolution of the tissue in the case where the subclone B appear after the treatment. As can be seen from the figure, initially only clone A exists, which grows and expands on the tissue. Subsequently, in subplot c), we can see that the treatment has an effect and the type A clones start to die. However, the subclone B is also present. Finally, in the last plot, it can be seen that clone B is dominant and A has almost completely disappeared. Similarly, if we consider the pre-existing case, it can be seen in figure 3.3 that the visualization is the same except for the fact that clone B existed before the therapy.

The Muller plot in figure 3.3 does not seem to show the presence of B before the therapy. However, this is an issue due to the tool rRACES. In fact, by observing the evolution of the tissue, we can see that B is already present at sampling t2. Additionally, I would like to draw particular attention to the fact that, in the final sampling of the two cases, the species B in the 'pre-existing' scenario is about double then the 'de-novo'. Unfortunately, this impacts the execution of the experiments; however, this issue will be addressed later.



(a) First biopsy at time t_1

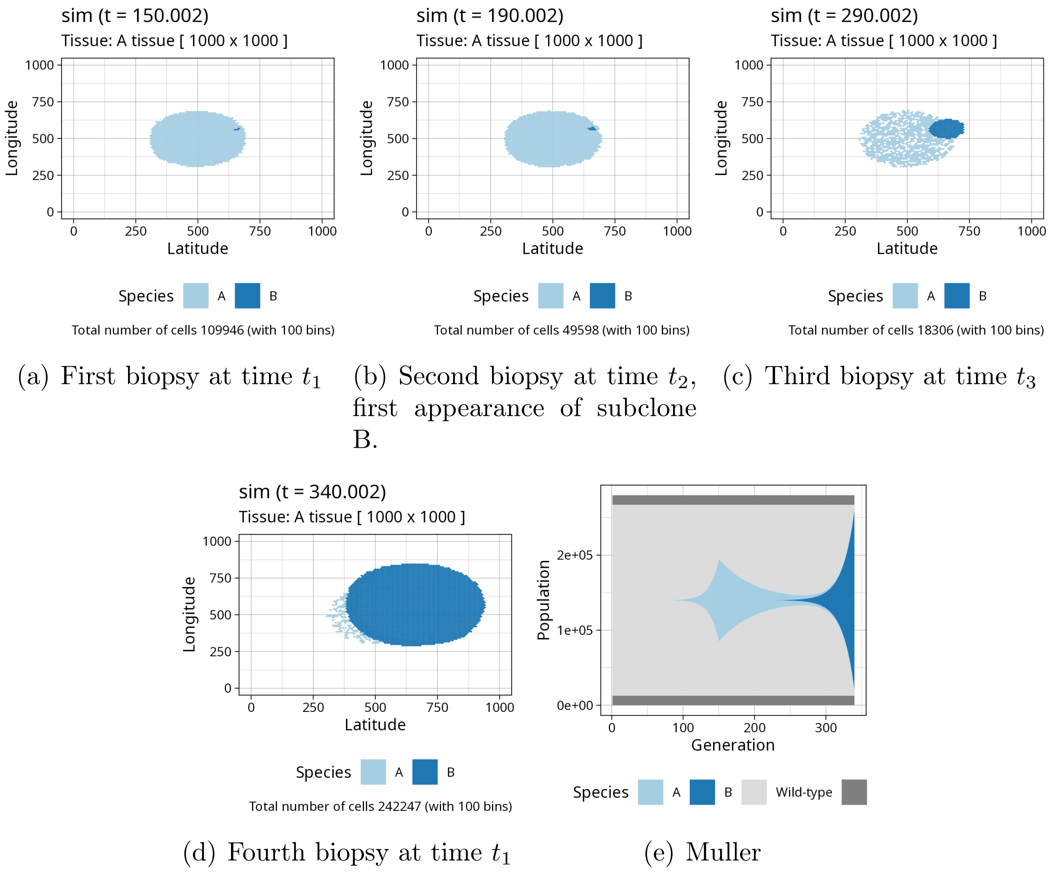
(b) Second biopsy at time t_2

(c) Third biopsy at time t_3, first appearance of subclone B.

(d) Fourth biopsy at time t_4

(e) Muller

Figure 3.2: De-novo scenario - In this sequence of plots, tissue growth can be observed at various time intervals, and furthermore, the Muller plot generated by the rRACES tool is also visible.



(a) First biopsy at time t_1

(b) Second biopsy at time t_2, first appearance of subclone B.

(c) Third biopsy at time t_3

(d) Fourth biopsy at time t_1

(e) Muller

Figure 3.3: Pre-existing scenario - In this sequence of plots, tissue growth can be observed at various time intervals, and furthermore, the Muller plot generated by the rRACES tool is also visible.

The summary statistics used in this ABC analysis are the following: the speed death of clone A after therapy

$$s_1 = \frac{(n^A_{t_{at}} - n^A_{t_{bt}})}{n^A_{t_{bt}}}$$

where t_{at} is the time of sampling after therapy and t_{bt} is the time of sampling before the therapy), the number of type B cells after some time since the therapy started ($s_2 = n^B_{t_i}$), and the speed growth of clone B at the from t_i to the last sampling at time t_j,

$$s_3 = \frac{(n^B_{t_j} - n^B_{t_i})}{n^B_{t_i}}$$

where $t_j >> t_i$.

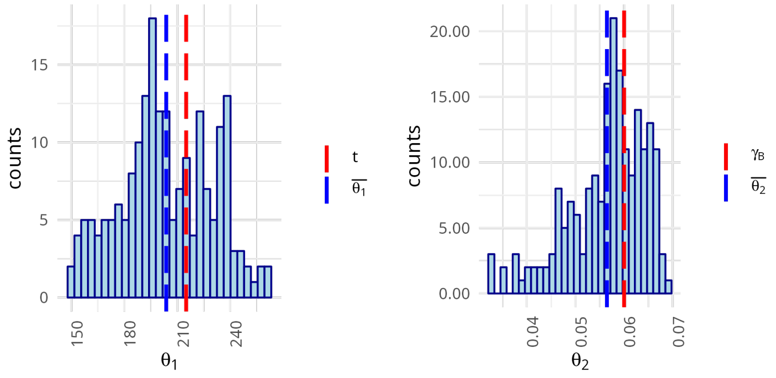
3.3 ABC analysis

3.3.1 De-novo case

Having obtained the summary statistics, we can now proceed with the ABC analysis. We will use the ABC rejection method exclusively because it's faster. This approach is straightforward: it samples parameters from the priors, runs the simulations, and compares the resulting statistics to the observed ones. Despite its simplicity, we need to limit the number of simulations. Initially, we started with 6000 simulations, then reduced to 4000, and finally to 2000, because the process was taking too long to complete. This issue is due to the limited resources of the supercomputer we have at our disposal. Therefore, we had to settle on 2000 simulations to ensure the analysis could be completed in a reasonable time frame, which is still about a week.

In the pre-existing case, the number of simulations for the ABC analysis had to be reduced to just 1000. As noted in the previous section, the model grows much faster in this scenario, making it impossible to perform more than 1000 simulations for this experiment. Additionally, even for just 1000 simulations, it still took about a week to complete.

Consistently with prior cases, uninformative priors were utilized, employing a uniform distribution, with 10% of the simulations considered acceptable. Figure 3.4 illustrates the outcome of the analysis, clearly demonstrating the success of the ABC analysis. Both parameters θ_1 and θ_2 have been accurately approximated and exhibit Gaussian distribution characteristics; their peaks align with the true values. Furthermore, upon examining the mean values of the accepted parameters for θ_1 and θ_2, it is observed that the true values fall within a narrow range of these estimates, with $t^B_{true} \in [\hat{\theta_1} - \epsilon_1, \hat{\theta_1} + \epsilon_1]$ and $\gamma_B \in [\hat{\theta_2} - \epsilon_2, \hat{\theta_2} + \epsilon_2]$.



(a) t_{true} (b) $gamma_B$

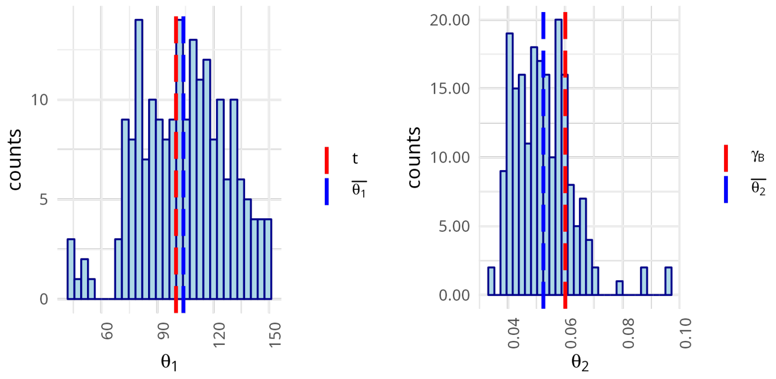
Figure 3.4: Distribution of θ_1 and θ_2 obtained after performing the ABC analysis with the adjusted summary statistics. The red dashed vertical line indicates the true value t for plot (a) and the true value γ_B for plot (b) that generated the original data, while the blue dashed vertical line indicates the mean values of θ_1 and θ_2, denoted as $\overline{\theta}_i$.

Figure 3.4 shows that the posterior distribution is very different from the prior distribution, and this characteristic is much more noticeable compared to the previous plots, confirming that the three summary statistics convey information. These plots confirm that the summary statistics convey information about γ_B and t, because the distances corresponding to the accepted values are clustered and not spread around the prior range of γ_B and t.

3.3.2 Pre-existing case

As mentioned in the previous section, for the ABC analysis of the pre-existing case, the simulations were cut down to 1000 due to limited resources. This time, we also used uninformative priors, meaning the parameters for the simulations were sampled from uniform distributions.

Also in this study case, can be seen that we obtain a good approximation desired parameter, the posterior probability of θ_1 and θ_2 seems to be a symmetric normal distribution for the θ_1 and an asymmetric normal distribution for θ_2. Figure 3.5 shows that the posterior distribution is very different from the prior distribution, and this characteristic is much more noticeable compared to the previous



(a) t_{true} (b) β

Figure 3.5: Distribution of θ_1 and θ_2 obtained after performing the ABC analysis with the adjusted summary statistics. The red dashed vertical line indicates the true value t for plot (a) and the true value γ_B for plot (b) that generated the original data, while the blue dashed vertical line indicates the mean values of θ_1 and θ_2, denoted as $\overline{\theta}_i$.

plots, confirming that the three summary statistics convey information. These plots confirm that the summary statistics convey information about γ_B and t, because the distances corresponding to the accepted values are clustered and not spread around the prior range of γ_B and t.

3.4 Code

```
# Import librarys
library(EasyABC)
library(rRACES)
library(ggplot2)

############################################################
# Setting the parameters to generate data
# Growth rate of clone A
gr.A = 0.08
# Growth rate of clone B
gr.B = 0.06
# Time units of tissue evolution and the time when clone B is added
t1 = 100
# First sampling of the tissue and start of therapy
t2 = 50
# First sampling after therapy
t3 = 40
```

```r
# Second sampling after therapy
t4 = 100
# Third sampling after therapy
t5 = 50


# Create simulation
sim <- new(Simulation, 'sim')
sim$history_delta = 1

##########################################################
# Add the mutant A
sim$add_mutant('A', gr.A, 0)

# Place cell on the tissue
dim <- sim$get_tissue_size()
sim$place_cell('A', dim[1]/2, dim[2]/2)

# Evolve the tissue with mutant A until t1
sim$run_up_to_time(sim$get_clock() + t1)

# add mutant B
sim$add_mutant("B", gr.B, 0)
sim$mutate_progeny(sim$choose_cell_in("A"), "B")

##########################################################
# Evolve the tissue with mutant A and B for t2 time units
sim$run_up_to_time(sim$get_clock() + t2)

# Register the first sample
sample.t1 <- sim$get_counts()$counts[1]

# start therapy
sim$update_rates("A", c(death=0.1))

##########################################################
# Evolve the tissue with mutant A for t3 time units
sim$run_up_to_time(sim$get_clock() + t3)

sample.t2 <- sim$get_counts()$counts[1]
sum_stat_1 <- (sample.t2-sample.t1)/sample.t1

##########################################################
sim$run_up_to_time(sim$get_clock() + t4)

sample.t4.A <- sim$get_counts()$counts[1]
sample.t4.B <- sim$get_counts()$counts[2]

sum_stat_2 <- sample.t4.B
```

```r

###########################################################
sim$run_up_to_time(sim$get_clock() + t5)

sample.t5.B <- sim$get_counts()$counts[2]
sum_stat_3 <- (sample.t5.B - sample.t4.B)/sample.t4.B

sum_stat_obs = c(sum_stat_1, sum_stat_2, sum_stat_3)

# Model definition
 ---------------------------------------------------
create_model <- function(x1, y2, y3, y4, y5){
  model=function(x){
    if (all(x > 0)){
      # Create simulation
      tmp <- new(Simulation, 'tmp')
      # Add the mutant A
      tmp$add_mutant('A', x1, 0)

      # Place cell on the tissue
      dim <- tmp$get_tissue_size()
      tmp$place_cell('A', dim[1]/2, dim[2]/2)

      # Evolve the tissue with mutant A until t1
      tmp$run_up_to_time(tmp$get_clock() + x[1])

      tmp$add_mutant("B", x[2], 0)
      tmp$mutate_progeny(tmp$choose_cell_in("A"), "B")

      tmp$run_up_to_time(tmp$get_clock() + y2)
      # Register the first sample
      sample.t1 <- tmp$get_counts()$counts[1]

      # start therapy
      tmp$update_rates("A", c(death=0.1))

      # Evolve the tissue with mutant A for t2 time units
      tmp$run_up_to_time(tmp$get_clock() + y3)

      sample.t2 <- tmp$get_counts()$counts[1]
      sum_stat_1 <- (sample.t2-sample.t1)/sample.t1


      tmp$run_up_to_time(tmp$get_clock() + y4)

      sample.t4.A <- tmp$get_counts()$counts[1]
      sample.t4.B <- tmp$get_counts()$counts[2]

      sum_stat_2 <- sample.t4.B

      tmp$run_up_to_time(tmp$get_clock() + y5)

      sample.t5.B <- tmp$get_counts()$counts[2]
      sum_stat_3 <- (sample.t5.B - sample.t4.B)/sample.t4.B

      sum_stat_obs = c(sum_stat_1, sum_stat_2, sum_stat_3)
    } else {
      s=c(0,0,0)
    }
  }
  print("The model has been created")
  return(model)
}

# Create the gr_model
m <- create_model(gr.A, t2, t3, t4, t5)

nb_simul=1000
tr=0.20
prior = list(c("unif", 10, 150), c("unif", 0.03, 0.1))

ABC_rej <- ABC_rejection(
model=m,
prior=prior,
nb_simul=nb_simul,
summary_stat_target=sum_stat_obs,
tol=tr)
```

Listing 3.1: Script for experiment of chapter 3

Conclusion

In conclusion, our comprehensive exploration and implementation of the ABC method for parameter estimation and inference within evolutionary models have provided significant insights and achievements. Through a systematic progression from basic to complex scenarios, reflecting real-world complexities, we demonstrated the effectiveness and applicability of the ABC approach. By meticulously selecting appropriate summary statistics and employing a rigorous methodology, we successfully bridged the gap between theoretical assumptions and empirical observations. This robust framework enabled us to approximate the underlying probability distributions governing the parameters under study, thus enhancing our understanding of evolutionary dynamics. Moreover, our findings highlight the versatility and adaptability of the ABC method in capturing the complexities of real-world systems. By advancing our methodological toolkit and refining our understanding of evolutionary processes, we are better equipped to tackle the challenges and intricacies of contemporary scientific inquiry.

Bibliography

[1] Robert F. Murphy. *What is Computational Biology?* 2024. URL: https://cbd.cmu.edu/about-us/what-is-computational-biology.html#:~:text=Computational%20biology%20is%20the%20science,or%20genes)%20when%20expressed%20produce (visited on 04/05/2024).

[2] Manolis Kellis. *Lecture notes in ComputationalBiology: Genomes, Networks, Evolution*. Jan. 2016.

[3] Andreas Handel. *Infectious Disease Epidemiology (Ecology)(Evlution) - a Model-based Approach (IDEMA)*. Feb. 2021.

[4] Giulio Caravagna and Alberto Casagrande. *rRACES: An R Wrapper for RACES*. R package version 0.6.0. 2024. URL: https://caravagnalab.github.io/rRACES/index.html.

[5] A. Ben Abdessalem et al. *An efficient likelihood-free Bayesian computation for model selection and parameter estimation applied to structural dynamics*. Jan. 2023.

[6] Franck Jabot et al. *EasyABC: Efficient Approximate BayesianComputation Sampling Schemes*. R package version 1.5.2. 2016. URL: http://easyabc.r-forge.r-project.org/.