INTERNSHIP REPORT

Deep Learning and Approximate Bayesian Computation for Finite sites model multivariate inference

Mohammed-Yassine Habibi

Supervisor :
Prof. Simon Tavaré

Irving Institute for Cancer Dynamics and Department of Statistics,
Columbia University, New York, NY, USA

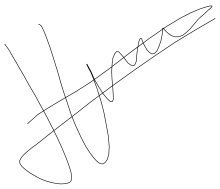
Research Internship — April-July 2024

Declaration of academic integrity

I, Mohammed-Yassine Habibi, hereby confirm :

- That the results presented in this report are exclusively the outcome of my work.
- That I am the author of this report.
- That no sources or materials were used in this report without being clearly acknowledged according to the recommended bibliographic rules.

I declare that this document cannot be suspected for plagiarism.



Date : July 31, 2024

Acknowledgement

I want to thank my supervisor Dr. Simon Tavaré for providing me with this extraordinary opportunity to work on such an interesting and promising subject and to discover the amazing setting of the IICD. His encouragement allowed me to explore brand new areas and his precious expertise guided me and helped me grow throughout my research.

I would like to express my gratitude to Dr. Khanh Dinh who was also present throughout my internship to provide me valuable help and advice and without whom my work and my experience here wouldn't have been the same.

Also, I would like to thank Zijin Xiang, a fellow student at Columbia University working at the IICD, and Théodore Fougereux, a fellow intern at the IEOR Department. Zijin helped me understanding in detail specific algorithms and was always accessible and glad to answer questions. Additionally, my discussions with Théodore provided valuable mathematical insights into Markov chain processes.

Last but not least, my experience in the laboratory was enhanced by Lorenza Favrot and Reed Black who were regularly making sure everything was going well, by the other interns at the institute, namely Tess Breton and Madeleine Hueber, and more generally by all the co-researchers of the IICD I had the chance to exchange with.

Abstract

Understanding the mechanisms of DNA molecule evolution and mutation, both over population evolutionary time scales and within the human lifespan, is of significant scientific interest. We will focus on DNA molecules suspected to have biased transition parameters (where certain nucleotide substitutions are more favored than others) and those that should be studied using a finite-sites approach. To understand the behavior of a set of cells of interest, we will rely on multi-parametric stochastic evolutionary models that take multiple parameters as input and produce nucleotide sequences sampled from a population of individuals or cells as output.

At first, we will propose a method to summarize complex and large genomic data into a vector of low-dimensional summary statistics using a deep learning algorithm inspired by Sanchez et al. 2021 and that we adapted to handle nucleotide sequences that differentiate the A, T, C, and G bases.

In a second stage, we will perform an inference task to fit the parameters of a stochastic model to our study data. To do this, we will use an adaptive sequential Monte Carlo algorithm, where each step involves running an approximate Bayesian computation random forest algorithm that uses simulated data to derive an increasingly precise posterior distribution.

Our results will show our algorithm's ability of inferring parameters for a finite site model generating sequences with multiple non-visible mutations.

Contents

Introduction

This report is the result of a four month internship at the Herbert and Florence Irving Institute for Cancer Dynamics (IICD) at Columbia University. The IICD is a multidisciplinary institute that explores the relationship between mathematical sciences and cancer research. The main goal of the laboratory is to improve understanding of cancer biology, its origins, treatment and prevention. The IICD focuses on multiple scales, from cellular and gene expression levels to entire populations. Those researches are supported by strong statistical and data science methodologies, including stochastic computation, bioinformatics, and causal inference.

My own project takes a closer look on mathematical evolutionary models for cancer. More specifically, I will focus on the methods that would allow us to fit those mathematical evolutionary models to specific cancer data. Indeed, if this happens to be possible, it would therefore be possible to generate new relevant genomic data $\mathbf{X}$ and thus to study more deeply different biological phenomena observed in cancer cells.

My approach is motivated by recent breakthroughs in parameter inference for population genomics evolutionary models (which are very similar to those in oncology). In fact, different brand new inference methods have been proposed since 2019, combining deep learning and likelihood free inference methods developed since late 90s and known as Approximate Bayesian Computation (ABC) algorithms (ABC was used for posterior inference for the first time by Tavaré et al. 1997 and by Fu and Li 1997).

There are multiple challenges to tackle to be able to infer evolutionary models parameters from genomic data $\mathbf{X}$ observed in a sample of cells. In brief, we have to deal with models parameterized by a large multivariate vector of parameters θ and characterized by an intractable likelihood. Moreover, the observed genetic data $\mathbf{X}$ produced by the simulations are of extremely large dimension. Therefore, different mathematical tools have been introduced in this context and they must be used simultaneously to realize the inference task.

In Section 1, we will present the stochastic models we're using to generate nucleotide sequences sampled in a population of cells (Kingman 1982, Baumdicker et al. 2022). Afterwards, in Section 2, we will present the general idea behind the ABC algorithms before showing how random forests with an adaptive metric for multivariate parameters can allow efficient estimation of a posterior distribution of a large multivariate vector of parameters θ, even when the data observed are still relatively high-dimensional (Raynal et al. 2018, Ćevid et al. 2022, Dinh et al. 2024a). Finally, in Section 3, we will describe a strategy we adapted from Sanchez et al. 2021 to summarize a genomic matrix $\mathbf{X}$ into an informative multivariate statistic with at least the same number of components as the target parameter θ. Ultimately, in Section 4, we will present our global framework and our results that show the efficiency of the novel inference methodology as well as the relevance of the neural network architecture to generate informative summary statistics.

1 Stochastic models for cancer evolution

During our research, we primarily used the `msprime` Python library, as described by Baumdicker et al. 2022, to run mathematical stochastic models $\mathcal{M}$. These models take a set of parameters $\theta \in \mathbb{R}^d$ and generate genetic information for a number I of cells or individuals that evolved from a common ancestor. The generated data is designed to be similar to real genetic data available. Our research focused specifically on **multi-parametric mutation models** and we considered **recombination-free** scenarios.

In this section, we are going to present the essential elements of the stochastic models we used to generate data.

1.1 Coalescent model

1.1.1 Generate evolutionary trees

First, we need to be able to generate evolutionary trees that will propose an explanation of the history of our sample of I cells and the way they originated from a common ancestor. To do so, we will rely on a coalescence simulation framework proposed by Kelleher, Etheridge, and McVean 2016 and implemented in the `msprime` python library by Baumdicker et al. 2022

The main idea of the coalescent model, mainly attributed to Kingman 1982, is to derive a **chronogram** that will be built from the bottom (from the I current cells) by determining iteratively the timing of the coalescent events (the time when a common parent cell gave two child cells) and the cells taking part in those events. There are exactly $I-1$ coalescent events in a simulation with a final sample of I cells. After j coalescent events, the sample of I cells has $I-j$ distinct ancestors. The successive times between two coalescent events $(T_j)_{I \geq j \geq 2}$ (T_j corresponding to the length of time the sample of I cells had j distinct ancestors) follow an exponential distribution of parameters $\left(\binom{j}{2}\right)_{I \geq j \geq 2}$.

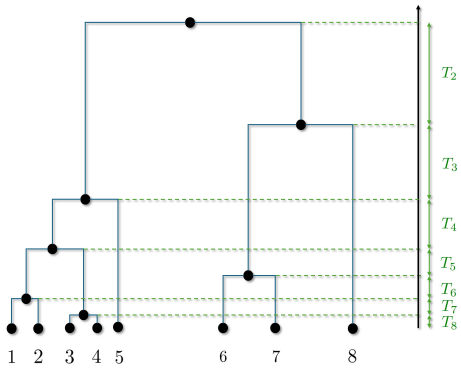


Figure 1: Evolutionary tree ($I = 8$ cells)

1.1.2 Generate mutations

Having built an evolutionary tree for a sample of I cells, we have to generate mutations that will occur along the branches of the phylogenetic tree on a specific site of the genome.

In the coalescent model, the number of mutations on a branch of the tree with length l follows a Poisson distribution with mean $\frac{l\theta}{2}$. The parameter θ in our stochastic models is related to the mutation rate per gene per generation, μ, and the overall population size, N, by the equation $\theta = 2N\mu$.

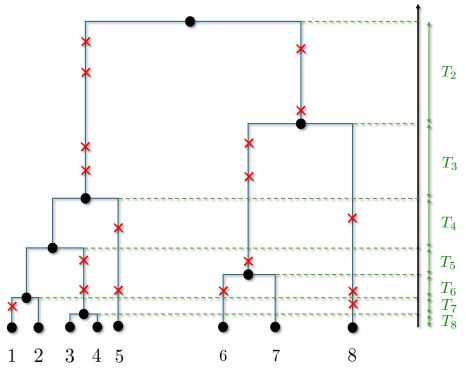


Figure 2: Evolutionary tree with mutations ($I = 8$ cells)

1.2 Infinite Sites Model VS Finite Sites Model

A site is represented by a single nucleotide (A, C, T or G). To place mutations along the sites of the genome, there are two major class of models we can use : **The Infinite Sites Model (ISM)** introduced by Kimura 1969 and **The Finite Sites Model (FSM)**. Let's see what are the assumptions and differences between those two models.

The Finite sites Model assumptions :

- A fixed number S of sites where mutations can occur
- A fixed site s can witness multiple mutations.

The Infinite sites Model assumptions :

- There are an infinite number of sites where mutations can occur
- No mutation can occur twice on the same site

The ISM is making more assumptions, therefore using it to model real dataset is sometimes leading to severe biases in the estimation of different parameters as the population mutation rate (it has been highlighted by Mathew et al. 2013). However, the ISM is still widely used because it is easier to derive interesting summary statistics from data generated with this framework.

1.3 The selected model : a 13-parameter model

The model we will keep in our work is a coalescent FSM $\mathcal{M}_{FSM}$. The overall population N, the sampled population I, the number of sites observed S and the root distribution $(\pi_A, \pi_T, \pi_C, \pi_G)$ of the 4 nucleotides A, C, T, G in the common ancestor will all be fixed before running the simulation.

Our vector of parameters $\theta = (\theta_1, \dots, \theta_{13})$ resides in $\mathbb{R}_+ \times [0,1]^{12}$. The first parameter, θ_1, represents the overall mutation rate, as introduced in Section 1.1.2. The parameters θ_2 through θ_{13} correspond to transition parameters. Specifically, we assume that for a fixed ancestral nucleotide $n_{\mathrm{a}} \in \{A, C, T, G\}$ mutating, the mutated nucleotide n_{m} is chosen thanks to the transition probabilities $\pi_{n_{\mathrm{a}} n_{\mathrm{m}}}$ that are parameters of $\mathcal{M}_{FSM}$. This assumption reflects practical observations and is crucial for accurately modeling certain datasets (see Konrad et al. 2017). There are 12 transition parameters and they satisfy

$$\forall\, n \in \{A, C, T, G\}, \quad \sum_{a \in \{A,C,T,G\}} \pi_{na} = 1$$

$$\begin{bmatrix} \cdot & \pi_{AT} & \pi_{AC} & \pi_{AG} \\ \pi_{TA} & \cdot & \pi_{TC} & \pi_{TG} \\ \pi_{CA} & \pi_{CT} & \cdot & \pi_{CG} \\ \pi_{GA} & \pi_{GT} & \pi_{GC} & \cdot \end{bmatrix}$$

Figure 3: General transition matrix

.

In Section 2, we will explain how we expect to infer the parameters of real genomic data we observe in cells ($\mathbf{X}^{obs}$) by leveraging the possibility to simulate data from a general stochastic model $\mathcal{M}$. Finally, in section 3, we will present a method to generate informative summary statistics for data generated with the finite sites model $\mathcal{M}_{FSM}$.

2 Parameter inference for models with intractable likelihood

Let $\mathbf{X}^{obs} \in \mathbb{R}^m$ be our observed genomic data consisting of position-informed nucleotide sequences found in chromosomes sampled from a single population of chromosomes. From this raw data, we will suppose in this part that we can extract low-dimensional informative summary statistics denoted by $\mathbf{S}^{obs} = \mathcal{S}(\mathbf{X}^{obs})$ $(n << m)$. The extraction of summary statistics in the context of FSM will be the focus of Section 3.

Let $\mathcal{M}$ be a general stochastic model taking a vector of parameters $\theta \in \mathbb{R}^d$ as input, so that $\mathbf{X} = \mathcal{M}(\theta)$ is a random variable outputted by a simulation from the model $\mathcal{M}$. We assume that we are given a prior π over the vector of parameters θ. Our goal is, having observed an output $\mathbf{X}^{obs}$ from one simulation, to infer the posterior distribution $\pi(\,\cdot\,|\mathbf{X}^{obs})$ which contains, in Bayesian inference, the complete knowledge we can have on the value θ used to generate $\mathbf{X}^{obs}$. Bayes' Theorem gives us that $\pi(\theta|\mathbf{X}^{obs}) = \dfrac{\mathbb{P}(\mathbf{X}^{obs}|\theta)\pi(\theta)}{\int_{\mathbb{R}^d}\mathbb{P}(\mathbf{X}^{obs}|\theta')\pi(\theta')\mathrm{d}\theta'}$

However, in the context of statistical inference for stochastic models, computing the likelihood function is either very difficult or impossible, and that is the reason why population geneticists developed an empirical framework, Approximate Bayesian Computation (ABC), to compute the posterior in the cases where we can simulate easily new datasets but we have an uncomputable likelihood.

2.1 Approximate Bayesian Computation

Approximate Bayesian computation (ABC) is a particular case of the "likelihood-free" Bayesian methods that has proven to be an effective and intuitive way of performing an approximate Bayesian analysis (Sisson, Fan, and M. A. Beaumont 2018). First, we need a function S which summarizes an observation $\mathbf{X}$ into a lower dimension vector $\mathbf{S} = \mathcal{S}(\mathbf{X})$. Then we will use different methods to derive an approximation $\pi\left(\theta|\|\mathcal{S}(\mathbf{X}) - \mathcal{S}(\mathbf{X}^{obs})\| < \epsilon\right)$ of the posterior distribution $\pi\left(\theta|\mathbf{X}^{obs}\right)$. ϵ is a threshold to be determined.

The first intuitive algorithm is the ABC rejection algorithm

Algorithm 1: ABC rejection

```
for i = 1, ..., N do
    Sample θ ~ π
    Simulate data X from the model with parameter θ
    Compute summary statistics S = S(X)
    if ||S(X) − S(X^obs)|| ≥ ε then
        return to Step 2
    θ^(i) ← θ
```

This method can be very efficient when θ is one-dimensional, but it is more complicated in higher dimensions. Indeed, a main drawback of this method is that the number of simulations needs to be exponentially large in the dimension of $\mathbf{S}$ to approximate as precisely the distribution $\pi^{\epsilon}_{ABC}(\theta) = \pi\left(\theta|\|\mathcal{S}(\mathbf{X}) - \mathcal{S}(\mathbf{X}^{obs})\| < \epsilon\right)$. Moreover, this ABC framework doesn't give good results when the summary statistics are high-dimensional with strong correlations between the variables (cf. Biau, Cérou, and Guyader 2015). In addition, the choice of the hyperparameter ϵ has a significant impact on both the result and the computational cost of the algorithm.

As a result, we will introduce a random forest methodology in the context of ABC parameter inference that has been proposed by Raynal et al. 2018 and leads to better results.

2.2 Random Forest for ABC parameter inference

The Random Forest (RF) methodology proposed by Breiman 2001 has been used by Raynal et al. 2018 as a non-parametric regression method to approximate the posterior distribution $\pi(\cdot\,|\mathbf{X}^{obs})$ of a single parameter $\theta \in \mathbb{R}$ when the summary statistics are high-dimensional.

2.2.1 Random Forest

The regression setting of Breiman's RF aims to explain a variable $\theta \in \mathbb{R}$ by a vector of covariates $\mathbf{S} = \mathcal{S}(\mathbf{X}) = (S^{(1)}, \dots, S^{(n)}) \in \mathbb{R}^n$. Let's draw N parameters $\theta_1, \dots, \theta_N \sim \pi$ and simulate $\mathbf{X}_1, \dots, \mathbf{X}_N \sim \mathcal{M}(\theta_1), \dots, \mathcal{M}(\theta_N)$. We define $\mathbb{X} = \{\mathbf{X}_1 \dots, \mathbf{X}_N\} \subset \mathbb{R}^m$.

First, a RF has to be trained with a training dataset $\mathcal{D} = (\theta_i, \mathbf{S}_i = \mathcal{S}(\mathbf{X}_i))_{1 \le i \le N}$. The training consists in building a fixed number B of binary trees so that each tree T_b (for $b \in \{1, \dots, B\}$) provides a partition of the covariate space $\mathbb{R}^n$.

Each tree is built iteratively from the top and each parent node corresponds to a binary rule that will partition the training dataset. At each parent node V, a pair $(j_V, s_V) \in [\![1, n]\!] \times \mathbb{R}$ is chosen so that the node corresponds to the binary rule $S^{(j_V)} \le s_V$ versus $S^{(j_V)} > s_V$ and minimizes a L^2-loss criterion on $\mathcal{D}_V \subset \mathcal{D}$ ($\mathcal{D}_V$ are the points that verify all the binary rules in the path from the root of the tree to V).

Let L and R be the 2 child nodes of V, and let $\begin{cases} \mathcal{D}_L \coloneqq \{(\theta, \mathbf{S}) \in \mathcal{D}_V \mid S^{(j_V)} \le s_V\} \\ \mathcal{D}_R \coloneqq \{(\theta, \mathbf{S}) \in \mathcal{D}_V \mid S^{(j_V)} > s_V\} \end{cases}$

The L^2-loss criterion to minimize is the following one :

$$\frac{1}{|\mathcal{D}_V|}\left(\sum_{(\theta,s)\in\mathcal{D}_L}\left(\theta - \bar{\theta}_L\right)^2 + \sum_{(\theta,s)\in\mathcal{D}_R}\left(\theta - \bar{\theta}_R\right)^2\right)$$

where $\bar{\theta}_L = \dfrac{1}{|\mathcal{D}_L|}\displaystyle\sum_{(\theta,s)\in\mathcal{D}_L}\theta$ and $\bar{\theta}_R = \dfrac{1}{|\mathcal{D}_R|}\displaystyle\sum_{(\theta,s)\in\mathcal{D}_L}\theta$. When the building of a tree is complete (each leaf can't split further because they meet a stopping criterion), the leaves $L_b^1, \dots, L_b^{K_b}$ of the tree T_b give us a partition $\mathbb{R}^m = \bigsqcup_{l=1}^{K_b} E_b^l$ of the covariate space in which lives $\mathbb{X}$.

It is worth noting that to assure diversity among the B trees, the root dataset $\mathcal{D}_b$ of the tree T_b is chosen randomly as a subset of our complete training dataset $\mathcal{D}$, and the splitting dimension j_V at a node V is also chosen among a set of dimensions chosen randomly.

Let's see now how Raynal et al. 2018 uses a RF to compute an approximation of the posterior expectation and the posterior distribution.

2.2.2 Computing a posterior distribution

Let $\mathbf{X}^{obs}$ be the data observed for which we want to infer the posterior distribution $\pi\left(\cdot\,|\mathbf{X}^{obs}\right)$. We will use the forest built with the training dataset $\mathcal{D}$ to determine an approximated posterior distribution of θ knowing $\mathbf{X}^{obs}$.

In practice, we will try to determine weights for each training point $\left(w^i(\mathbf{X}^{obs})\right)_{1\le i\le N}$ so that

$$\begin{cases} \left(w^1(\mathbf{X}^{obs}), \dots, w^N(\mathbf{X}^{obs})\right) \propto \left(\pi\left(\theta_1|\mathbf{X}^{obs}\right), \dots, \pi\left(\theta_N|\mathbf{X}^{obs}\right)\right) \\ \displaystyle\sum_{i=1}^{N} w^i(\mathbf{X}^{obs}) = 1 \end{cases}$$

Let's define $\forall\ b \in [\![1, B]\!],\ \mathcal{X}_b \coloneqq \mathbb{X} \cap \{\mathbf{X}_b \mid \exists\ \theta_b, (\theta_b, \mathcal{S}(\mathbf{X}_b)) \in \mathcal{D}_b\}$, and

$$w^i(\mathbf{X}^{obs}) \coloneqq \frac{1}{B}\sum_{b=1}^{B}\sum_{l=1}^{K_b}\frac{\mathbb{1}(\mathbf{X}^{obs} \in E_b^l)\mathbb{1}(\mathbf{X}_i \in E_b^l \cap \mathcal{X}_b)}{|E_b^l \cap \mathcal{X}_b|}.\quad \forall\ i \in [\![1, N]\!]$$

Because we have a partition $\mathbb{R}^m = \bigsqcup_{l=1}^{K_b} E_b^l$, for all $b \in [\![1, B]\!]$, then

$$\forall\ b \in [\![1, B]\!],\ \exists!\ l_b^{obs} \in [\![1, K_b]\!],\ \mathbf{X}^{obs} \in E_b^l$$

$$\begin{aligned}
\text{So}\quad \sum_{i=1}^{N} w^i(\mathbf{X}^{obs}) &= \sum_{i=1}^{N}\frac{1}{B}\sum_{b=1}^{B}\sum_{l=1}^{K_b}\frac{\mathbb{1}(\mathbf{X}^{obs} \in E_b^l)\mathbb{1}(\mathbf{X}_i \in E_b^l \cap \mathcal{X}_b))}{|E_b^l \cap \mathcal{X}_b|} \\
&= \sum_{i=1}^{N}\frac{1}{B}\sum_{b=1}^{B}\frac{\mathbb{1}(\mathbf{X}_i \in E_b^{l_b^{obs}} \cap \mathcal{X}_b)}{|E_b^{l_b(\mathbf{X}^{obs})} \cap \mathcal{X}_b|} \\
&= \frac{1}{B}\sum_{b=1}^{B}\underbrace{\left(\sum_{i=1}^{N}\frac{\mathbb{1}(\mathbf{X}_i \in E_b^{l_b^{obs}} \cap \mathcal{X}_b)}{|E_b^{l_b(\mathbf{X}^{obs})} \cap \mathcal{X}_b|}\right)}_{=1} \\
&= 1
\end{aligned}$$

The methodology we summarized above, which is the one followed by Raynal et al. 2018, allows to compute weights that approximate the posterior distribution $\pi\left(\cdot\,|\mathbf{X}^{obs}\right)$ by an empirical distribution $\pi_{ABC-RF}\left(\cdot\,|\mathcal{S}(\mathbf{X}^{obs})\right)$. This method is way more consistent and efficient than the standard ABC methods when the summary statistics are high-dimensional and contain noise or irrelevant components. However, this method is designed for a setup where θ is a scalar and it is discussed in Raynal et al. 2018 that the attempts were unfruitful for multidimensional correlated parameters.

In the next section, we will introduce a state of the art algorithm used to estimate a multivariate conditional distribution and based on the ABC-RF framework we just introduced. It is called distributional random forest (Ćevid et al. 2022).

2.3 Distributional Random Forests for multivariate parameter

In this section we will present a novel forest construction introduced by Ćevid et al. 2022 that introduces a new criterion to split trees nodes in order to estimate a joint posterior distribution $\pi\left(\theta|\mathbf{X}^{obs}\right)$ when $\theta \in \mathbb{R}^d$ with $d > 1$.

This method is in line with Breiman 2001's RF approach, but is based on the use of a multivariate two-sample test statistic in order to partition the data instead of the L^2 criterion. This is indeed a way to deal with correlated parameters and to assure that the resulting distributions in the leaf nodes are as homogeneous as possible.

Identifying the most suitable metric is particularly challenging due to the curse of dimensionality. The metric must efficiently detect any changes in distribution when building the tree and partitioning the data, a task that is inherently difficult for multivariate data. Additionally, the metric must be computationally efficient as it is extensively used in the algorithm. These challenges are underscored by Ćevid et al. 2022, who propose a metric based on an approximation of the Maximal Mean Discrepancy (MMD) two-sample test statistic introduced by Gretton et al. 2007, which meets both of these requirements.

Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a positive-definite kernel. The MMD statistic for kernel k and two samples $\Theta^{(1)}$ and $\Theta^{(2)}$ both included in $\mathbb{R}^d$ is defined in the following way :

$$\mathbf{MMD}_k(\Theta^{(1)}, \Theta^{(2)}) =$$

$$\underbrace{\frac{1}{|\Theta^{(1)}|^2} \sum_{\substack{\theta_1^{(1)} \in \Theta^{(1)} \\ \theta_2^{(1)} \in \Theta^{(1)}}} k(\theta_1^{(1)}, \theta_2^{(1)}) + \frac{1}{|\Theta^{(2)}|^2} \sum_{\substack{\theta_1^{(2)} \in \Theta^{(2)} \\ \theta_2^{(2)} \in \Theta^{(2)}}} k(\theta_1^{(2)}, \theta_2^{(2)})}_{\text{evaluates similarities within each sample}} - \underbrace{\frac{2}{|\Theta^{(1)}||\Theta^{(2)}|} \sum_{\substack{\theta^{(1)} \in \Theta^{(1)} \\ \theta^{(2)} \in \Theta^{(2)}}} k(\theta^{(1)}, \theta^{(2)})}_{\text{evaluate similarities across the 2 samples}}$$

This previous expression has a complexity of $\mathcal{O}(|\Theta^{(1)}|^2 + |\Theta^{(2)}|^2)$ which is too large for many applications. Consequently, Ćevid et al. 2022 suggest to use a fast approximation of the MMD.

A Hilbert Space $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ called the Reproducing Kernel Hilbert Space (RKHS) of real-valued functions on $\mathbb{R}^d$ induced by k can be defined such that there exists a function $\varphi : \mathbb{R}^d \to \mathcal{H}$ that satisfies $k(\theta_1, \theta_2) = \langle \varphi(\theta_1), \varphi(\theta_2) \rangle_{\mathcal{H}}$.

One immediate consequence of that is that we can write the MMD statistic like this :

$$\mathbf{MMD}_k(\Theta^{(1)}, \Theta^{(2)}) = \left\| \frac{1}{|\Theta^{(1)}|} \sum_{\theta^{(1)} \in \Theta^{(1)}} \varphi(\theta^{(1)}) - \frac{1}{|\Theta^{(2)}|} \sum_{\theta^{(2)} \in \Theta^{(2)}} \varphi(\theta^{(2)}) \right\|_{\mathcal{H}}^2$$

Then, we use an expression of the kernel k as a Fourier transform of some measure ν (Bochner's theorem): $k(\theta^{(1)}, \theta^{(2)}) = \int_{\mathbb{R}^d} e^{i\omega^T(\theta^{(1)} - \theta^{(2)})} \mathrm{d}\nu(\omega)$ in order to derive the approximation k^F of k with

$$k^F(\theta^{(1)}, \theta^{(2)}) = \frac{1}{F} \sum_{f=1}^{F} e^{i\omega_f^T(\theta^{(1)} - \theta^{(2)})}$$

where $(\omega_f)_{1 \leq f \leq F}$ are frequency vectors randomly sampled from ν and F is an arbitrary integer.

Finally we can define $\forall\ \omega \in \mathbb{R}^d, \varphi_\omega^F(\theta) = e^{i\omega^T\theta}$. This feature map satisfies

$$k^F(\theta_1, \theta_2) = \frac{1}{\sqrt{F}} \left(\overline{\varphi_{\omega_1}^F(\theta_1)}, \ldots, \overline{\varphi_{\omega_F}^F(\theta_1)} \right) \times \frac{1}{\sqrt{F}} \begin{pmatrix} \varphi_{\omega_1}^F(\theta_2) \\ \vdots \\ \varphi_{\omega_F}^F(\theta_2) \end{pmatrix}$$

It defines consequently the following random approximation of $\mathbf{MMD}_k(\Theta^{(1)}, \Theta^{(2)})$:

$$\mathbf{MMD}_{k^F}(\Theta^{(1)}, \Theta^{(2)}) = \frac{1}{F} \sum_{f=1}^{F} \left| \frac{1}{|\Theta^{(1)}|} \sum_{\theta^{(1)} \in \Theta^{(1)}} \varphi_{\omega_f}^F(\theta^{(1)}) - \frac{1}{|\Theta^{(2)}|} \sum_{\theta^{(2)} \in \Theta^{(2)}} \varphi_{\omega_f}^F(\theta^{(2)}) \right|^2$$

In practice, we will take k a Gaussian kernel with bandwidth σ, resulting in $\omega_1^T, \ldots, \omega_F^T \sim \mathcal{N}_d\left(0, \sigma^{-2} I_d\right)$.

The computational cost of this expression is now $\mathcal{O}\left(F(|\Theta^{(1)}| + |\Theta^{(2)}|)\right)$ which is linear and no more quadratic in the size of the sample.

Ćevid et al. 2022 developed an R library which implements the building of clustering random forests with this approximation of the Maximum Mean Discrepancy criterion to split the parent nodes. This method called **distributional random forest** allows to realize Bayesian inference for high-dimensional summary statistics and multivariate parameters θ while necessiting a lower computational cost than

traditional ABC methods like Algorithm 1.

Dinh et al. 2024a assessed the efficiency of the ABC-RF method as well as its limitations on stochastic models in ecology, population genetics and systems biology. Indeed, they showed that there can be high uncertainty in the posterior if the prior distribution is uninformative and they adapted a sequential Monte Carlo method to ABC-RF, enabling an improvement in the accuracy of the estimated posterior distribution while maintaining a reasonable computational cost. We will explain what it is about and make it adaptive in a certain way we will make explicit.

2.4 Sequential Adaptive Monte-Carlo

The idea proposed by Sisson, Fan, and Tanaka 2007 was to modify an approximate Bayesian computation algorithm and to make it sequential. More precisely, we won't draw anymore all the parameters $\theta_1, \ldots, \theta_N$ from the prior π to infer the posterior (which can be very long or inefficient if the prior is not well calibrated). On the contrary, we will realize multiple iterations. At each iteration $t \geq 1$, we will only draw a smaller set of parameters $\theta_1^t, \ldots \theta_{N_f}^t$, with N_f being a fraction of N, run an ABC algorithm on the points $\mathbf{X}_1^t, \ldots, \mathbf{X}_{N_f}^t$ generated from the N_f parameters just drawn, and use the results to propose a better prior distribution for the next iteration $t+1$. In our case, it will not accelerate as much the building of the trees when using the ABC-RF algorithm (from $\mathcal{O}(N \log(N))$ to $\mathcal{O}(N \log(N_f))$) but it will enable better convergence results and better approximation of the posterior.

As highlighted in the work of Beaumont et al. 2009 and by Dinh et al. 2024a, the critical part in this algorithm that can have a huge impact on the rightness of the method is the way we will sample the parameters $\theta_1^{t+1}, \ldots \theta_{N_f}^{t+1}$ at the iteration $t+1 \geq 2$ from the posterior $\pi^t(\,\cdot\,|\mathbf{X}^{obs})$ obtained at the end of the t^{th} iteration. As a result, Beaumont et al. 2009 proposed what they call an "adaptive" method inspired by population Monte Carlo method of Cappé et al. 2004.

The idea is to draw at the first step $t = 1$,

$$\forall\, i \in \{1, \ldots, N_f\},\ \theta_i^1 \sim \pi(\,\cdot\,)$$

and then at each other step $t+1 \geq 2$

$$\forall\, i \in \{1, \ldots, N_f\},\ \theta_i^{t+1} \sim \mathcal{N}(\widetilde{\theta}^t, \tau_t^2) \text{ with } \begin{cases} \widetilde{\theta}^t \sim \pi^t(\,\cdot\,|\mathbf{X}^{obs}) \\ \tau_t^2 = \dfrac{2}{N_f} \displaystyle\sum_{i=1}^{N_f} \left(\theta_i^t - \frac{1}{N_f} \sum_{j=1}^{N_f} \theta_j^t \right)^2 \end{cases}$$

In practice, at each step t, once the training set of parameters $\theta_1^t, \ldots \theta_{N_f}^t$ has been drawn by this adaptive sampling method, we will use ABC-RF to compute $\pi^{t+1}(\,\cdot\,|\mathbf{X}^{obs})$. Now that we have a clear theoretical idea of how we can compute efficiently a good approximation of a posterior distribution for our multivariate parameter $\theta \in \mathbb{R}^d$, we have still to explicit our strategy to derive informative summary statistics from a dataset we assume to be generated from a Finite Sites Model framework. That will be the core subject of the Section 3 that comes next.

3 Deep Learning to generate informative summary statistics

In our work, we want to focus on a mutation model which is a Finite Sites Model (cf Section 1.2). To do so, we have to be able to generate sufficient, or at least informative summary statistics from an output $\mathbf{X}$ of our finite sites model $\mathcal{M}_{FSM}$.

First, let's clarify what will be the shape and the content of one data point $\mathbf{X}$. In the context of the FSM, we have a fixed number I of individuals from a common ancestor and a fixed number S of sites for which we have information about the ancestral nucleotide and the mutated one, as well as an indication of the position of this nucleotide in the sequence. The allowed nucleotides are A, T, C, G that are labeled respectively $0, 1, 2, 3$. We summarize those data in a tensor $\mathbf{X}$ as follows :

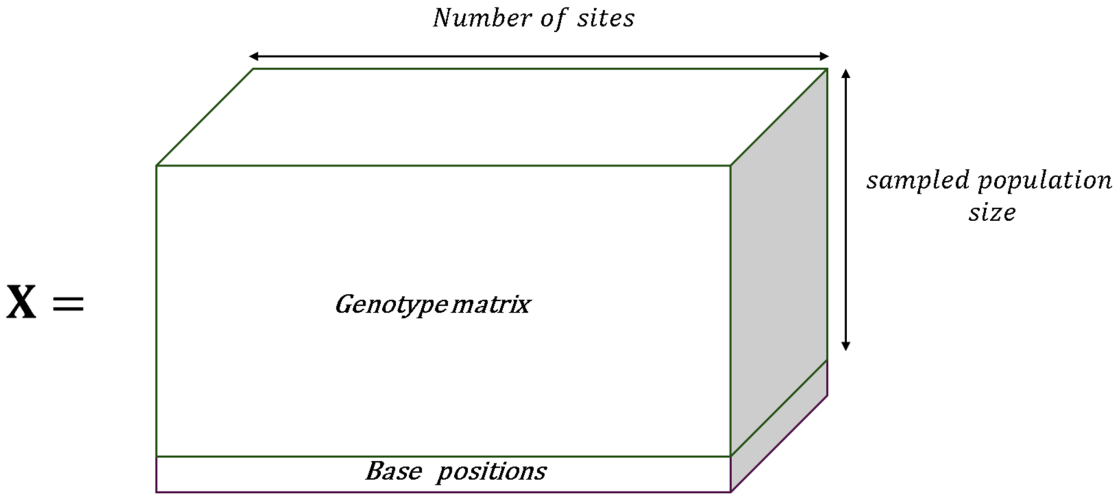
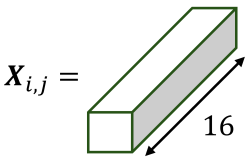


Figure 4: Format of our data points

For all $(i, j) \in [\![1, I]\!] \times [\![1, S]\!]$, $\mathbf{X}_{i,j} \in \mathbb{R}^{16}$ because it contains the information about the ancestral nucleotide and the current one at the site in position j for the individual/cell i. Let $a_j \in \{0, 1, 2, 3\}$ be the ancestral nucleotide at position j and $n_{i,j} \in \{0, 1, 2, 3\}$ be the current nucleotide at the same position for the i^{th} individual.



Then $\forall k \in [\![1, 16]\!]$, $\mathbf{X}_{i,j,k} = \mathbb{1}(k = 4a_j + n_{i,j} + 1)$. Therefore, our tensor $\mathbf{X}$ consists of $16 \times S \times (I + 1)$ integers. It can be extremely large even for relatively small sequences ($\approx 10^7$ for 100 individuals and 4000 base pairs).

As a consequence, it is essential to be able to reduce our data $\mathbf{X}$ into a low-dimensional vector $\mathcal{S}(\mathbf{X})$ to be able to apply the parameter inference methods we presented in Section 2. First, we will recall the results obtained by Jiang et al. 2017 concerning the sufficiency of the posterior mean as a summary statistic. Later, inspired by the work of Sanchez et al. 2021, we will adapt a convolutional neural network to generate summary statistics that are approximately the posterior means of the parameters we want to infer.

3.1 A theoretical result on the sufficiency of a Summary Statistic

In the context of the ABC-rejection algorithm (Algorithm 1), several results have been demonstrated concerning $\pi^{\epsilon}_{ABC}(\theta)$ in the case where we consider that our summary statistic is the vector $\mathcal{S}(\mathbf{X}) = \mathbb{E}_{\pi}[\theta|\mathbf{X}]$ that is supposed known. Indeed, Jiang et al. 2017 gives a proof of the following theorem :

Theorem 1. *If $\mathbb{E}_{\pi}[|\theta|] < \infty$, then $\mathcal{S}(\mathbf{X}) = \mathbb{E}_{\pi}[\theta|\mathbf{X}]$ is well defined.*
Moreover, for a fixed observation $\mathbf{X}^{obs}$, the Algorithm 1 with summary statistics $\mathcal{S}$, tolerance threshold ϵ and norm $\|\cdot\|$ gives the following results for the posterior $\pi^{\epsilon}_{ABC} = \pi(\theta|\|\mathcal{S}(\mathbf{X}) - \mathcal{S}(\mathbf{X}^{obs})\| < \epsilon)$:

$$\left\|\mathbb{E}_{\pi^{\epsilon}_{ABC}}[\theta] - \mathcal{S}(\mathbf{X}^{obs})\right\| < \epsilon$$

$$\mathbb{E}_{\pi^{\epsilon}_{ABC}}[\theta] \underset{\epsilon \to 0}{\longrightarrow} \mathbb{E}_{\pi}[\theta|\mathbf{X}^{obs}]$$

This result means that by taking as summary statistics the posterior averages of our objective parameters $\mathbb{E}_{\pi}\left[\theta|\mathbf{X}^{obs}\right]$, we don't lose any first-order information about $\mathbf{X}^{obs}$.

We can even derive a stronger result which is the following extension stated by Jiang et al. 2017 in the case where $\mathcal{S}(\mathbf{X}) = (\mathbb{E}_{\pi}[f_1(\theta)|\mathbf{X}], \ldots, \mathbb{E}_{\pi}[f_K(\theta)|\mathbf{X}])$ but for which we didn't find a demonstration in the paper. We will present the result and propose a demonstration.

Theorem 2. *We assume $\theta \in \mathbb{R}^d$ and consider the square integrable space with its inner product $\left(\mathrm{L}^2(\mathbb{R}^d), \langle\cdot,\cdot\rangle_2\right)$*
For a fixed $\mathbf{X}^{obs}$, we assume that $\pi(\,\cdot\,|\mathbf{X}^{obs}) \in \mathrm{L}^2(\mathbb{R}^d)$.
Let $(f_k)_{k\in\mathbb{N}^*}$ be an orthonormal basis of this Hilbert space.
We define $K \in \mathbb{N}^*$ and $\epsilon \in \mathbb{R}^*_+$.
Then, by defining the $K-$dimensional summary statistics $\mathcal{S}(\mathbf{X}^{obs}) = \left(\mathbb{E}_{\pi}[f_1(\theta)|\mathbf{X}^{obs}], \ldots, \mathbb{E}_{\pi}[f_K(\theta)|\mathbf{X}^{obs}]\right)$ and $\pi^{\epsilon,K}_{ABC} = \pi(\theta|\|\mathcal{S}(\mathbf{X}) - \mathcal{S}(\mathbf{X}^{obs})\| < \epsilon)$ we have,

$$\pi^{\epsilon,K}_{ABC} \underset{\substack{\epsilon\to 0\\ K\to\infty}}{\longrightarrow} \pi(\,\cdot\,|\mathbf{X}^{obs}) \quad \left(\textit{converges weakly in } \mathrm{L}^2\left(\mathbb{R}^d\right)\right) \textit{ if we have furthermore } K^{1/2}\epsilon \to 0$$

Proof. Let f be a $\mathrm{L}^2(\mathbb{R}^d)$ function. We can write $f = \sum_{k=1}^{\infty} \alpha_k f_k$.

We want to show $\langle \pi^{\epsilon,K}_{ABC}, f\rangle \underset{\substack{\epsilon\to 0\\ K\to\infty}}{\longrightarrow} \langle \pi(\,\cdot\,|\mathbf{X}^{obs}), f\rangle$.

$$
\begin{aligned}
\left|\langle \pi^{\epsilon,K}_{ABC}, f\rangle - \langle \pi(\,\cdot\,|\mathbf{X}^{obs}), f\rangle\right| &= \left|\sum_{k=1}^{\infty} \alpha_k \left(\langle \pi^{\epsilon,K}_{ABC}, f_k\rangle - \langle \pi(\,\cdot\,|\mathbf{X}^{obs}), f_k\rangle\right)\right| \\
&\leq \sum_{k=1}^{\infty} |\alpha_k| \underbrace{\left|\int_a^b \pi^{\epsilon,K}_{ABC}(\theta) f_k(\theta) \mathrm{d}\theta - \int_a^b \pi(\theta|\mathbf{X}^{obs}) f_k(\theta)\mathrm{d}\theta\right|}_{=\left|\mathbb{E}_{\pi^{\epsilon,K}_{ABC}}[f_k(\theta)] - \mathbb{E}_{\pi}[f_k(\theta)|\mathbf{X}^{obs}]\right|} \\
&\leq \left(\max_{1\leq k\leq K} |\alpha_k|\right) \sqrt{K} \left\| \left(\mathbb{E}_{\pi^{\epsilon,K}_{ABC}}[f_k(\theta)]\right)_{1\leq k\leq K} - \mathcal{S}(\mathbf{X}^{obs}) \right\|_{\mathbb{R}^K} + R_K \\
&\text{(Cauchy-Schwarz in } \mathbb{R}^K) \\
&\leq A\sqrt{K}\epsilon + \underbrace{R_K}_{\underset{K\to\infty}{\longrightarrow} 0} \\
&\text{(application of Theorem 1)}
\end{aligned}
$$

The result follows. Please refer to the supplementary material (24) for further details for this proof. □

The Theorem 2 establishes the feasibility of a global approximation of the posterior distribution with summary statistics that are posterior means of functions of the parameters θ. This justifies that researchers have been trying to approximate those posterior means to use them as summary statistics in an ABC framework in cases were we don't know other informative statistics. One promising way to do so is to use deep neural networks (cf. Jiang et al. 2017, Sanchez et al. 2021).

In the next part we will present and justify the structure of the SPIDNA neural network, developed by Sanchez et al. 2021, whose goal is to infer the summary statistics $\mathbb{E}_{\pi}\left[\theta|\mathbf{X}^{obs}\right]$ from $\mathbf{X}^{obs}$ in the specific context of $\mathbf{X}^{obs}$ being a single nucleotide polymorphic sites matrix with only a distinction between an ancestral allele and a derived one, encoded respectively by 0 and 1. After that, we will slightly adapt the SPIDNA to our data format.

3.2 SPIDNA Neural Network for genomic data

A designed neural network aiming to infer parameters from a position-informed set of nucleotide sequences $\mathbf{X} \in \{0,1\}^{S\times(I+1)}$ has to be invariant in the ordering of the nucleotide sequences that are encoded in the first I rows of $\mathbf{X}$ (the positional information are encoded in the last row $\mathbf{X}_{I+1}$). To do so, one can design a network so that its regression function f is sum-decomposable in the first I rows of $\mathbf{X}$, i.e. so that there are functions ρ and ϕ such that

$$
f(\mathbf{X}) = \rho\left(\sum_{i\in[\![1,I]\!]} \phi(\mathbf{X}_i)\right).
$$

That is the idea introduced by Zaheer et al. 2017 and used by Sanchez et al. 2021 to design their networks. Moreover, as $\mathbf{X}$ has a value in a finite set $\mathbb{X} \subset \{0,1\}^{S\times(I+1)}$, a result derived by Zaheer et al. 2017 states that

A function $f(\mathbf{X})$ is permutation-invariant iff it can be decomposed in the form $\rho\left(\sum_{i\in[\![1,I]\!]} \phi(\mathbf{X}_i)\right)$,

making the idea of finding a regression function f that is sum-decomposable all the more relevant.

In practice, the SPIDNA network proposed by Sanchez et al. 2021 fits the weights of a convolutional neural network that serves as the ϕ function and also aggregates positional information from $\mathbf{X}_{I+1}$. Finally, the ρ function consists in feeding the ϕ's outputs into a fully-connected layer.

Using the SPIDNA network as a starting point, we will propose a network very similar and adapted to our data format which is now $\mathbb{X} \subset \{0,1\}^{16\times S\times(I+1)}$ (instead of $\{0,1\}^{S\times(I+1)}$) to generate an approximation $\widehat{\mathbb{E}}_\pi[\theta|\mathbf{X}]$ of $\mathbb{E}_\pi[\theta|\mathbf{X}]$.

3.3 Defining a Multiallelic SPIDNA

We assume that we can draw parameters θ from a prior π and then generate data $\mathbf{X}$ thanks to a model $\mathcal{M}(\theta)$ as described in Section 1.

The architecture of our Multiallelic SPIDNA (M-SPIDNA) network is illustrated below. Filters, which can be viewed as kernels, are applied independently to the genotype tensor and the bases positions matrix. This approach enables the learning of the joint behavior of neighboring nucleotides, generating relevant features. These features are then passed through another set of weighted kernels to derive the final features. These final features are averaged and aggregated with trained weights to produce the prediction $\widehat{\mathbb{E}}_\pi[\theta|\mathbf{X}]$. The overall network has 5024 trainable weights.

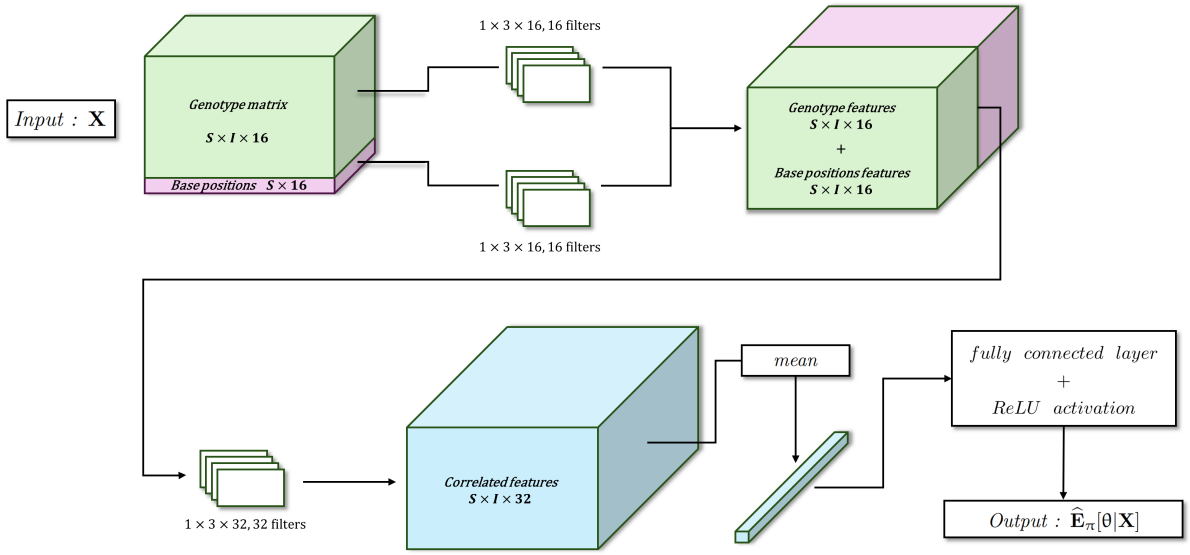


Figure 5: Multiallelic SPIDNA architecture

We were inspired by convolution on 3D tensors with 3D filters to design the $1 \times 3 \times 16$ kernels suited for this multiallelic configuration. We set the number of filters to 16, which is bigger than 13 to have at least 13 parameters after the final averaging, to predict the posterior means of the 13 parameters of the stochastic model we're focusing on. After testing several training loss, we chose the SmoothL1Loss which is a combination between an L1 and an L2 loss.

We could increase the number of filters and the number of outputed values of our network so that it fits the $\mathbb{E}_\pi[f_k(\theta)|\mathbf{X}]$ for $k \in \mathbb{N}^*$ as in Theorem 2, and expect therefore to have more informative summary statistics generated by the M-SPIDNA.

4 Synthetic framework and Results

As a reminder, the data observed $\mathbf{X}^{obs}$ is a position-informed genotype matrix that can either be real data obtained thanks to single-cell sequencing or simulated data. The details concerning the shape and the information contained in $\mathbf{X}^{obs}$ are given in Section 3. Moreover, we have at our disposal a stochastic coalescent finite site model $\mathcal{M}_{FSM}$ for generating new data $\mathbf{X}$ with the exact same information as $\mathbf{X}^{obs}$. The details of the model $\mathcal{M}_{FSM}$ are given in Section 1. Given a prior π and an observed data $\mathbf{X}^{obs}$, we will use the deep learning network introduced in Section 3, as well as the mathematical tools presented in section 2 to try to get as close as possible to the posterior distribution $\pi(\theta|\mathbf{X}^{obs})$. In practice, our prior will be $\mathcal{U}([a,b]) \times \mathbf{Dir}([\mathbf{1},\mathbf{1},\mathbf{1}])^{\otimes \mathbf{4}}$ with $\mathbf{Dir}([1,1,1])$ being the uniform Dirichlet distribution on the 3D simplex, and $[a,b]$ the range of the mutation rate.

4.1 Parameter inference framework

Let's look at our parameter inference framework (cf. Algorithm 2). The green boxes $\mathbf{X}^{obs}$ and $\pi(\theta)$ are respectively the given objective data and the chosen prior for the multivariate parameter θ we want to infer.

The first step in the large blue box is to generate a first training dataset $\mathcal{D}$ from the prior (using our stochastic model $\mathcal{M}_{FSM}$) to train our summary-statistics generator, the M-SPIDNA neural-network (the training of the M-SPIDNA is the step in purple). Notice that each time we change one of the fixed parameters of $\mathcal{M}_{FSM}$ as for example the root distribution $(\pi_A, \pi_T, \pi_C, \pi_G)$, or when we change our prior $\pi(\theta)$, we have to re-train our summary-statistics generator.

Then, once our summary-statistics generator is ready, we can effectively apply the SMC-ABC-RF framework presented in section 2. We use the Michel and Cevid 2021 `drf` library and got inspired by Dinh et al. 2024b's `abcsmcrf` library. We iteratively draw, at each step t, a set of training parameters from the current best posterior estimation obtained $\pi^{t-1}(\,\cdot\,|\mathbf{X}^{obs})$ (or π at the step 1) and approximate a more precise posterior by classifying summarized data and comparing it with the objective $\mathcal{S}(\mathbf{X}^{obs})$ thanks to the ABC-RF algorithm using random forest.

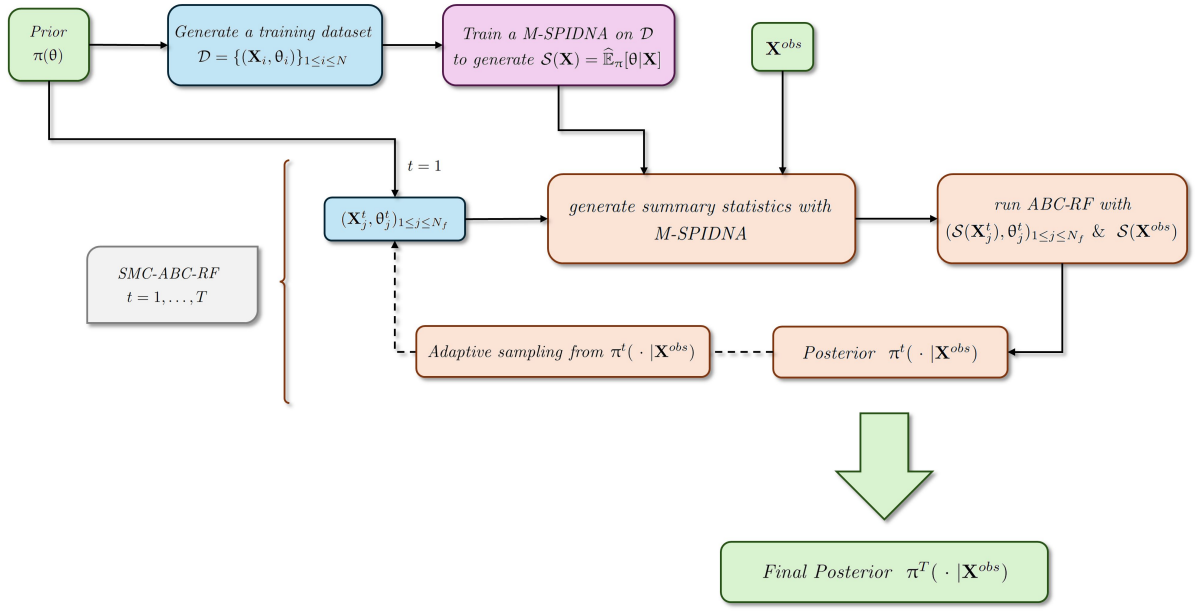


Figure 6: Schematic overview of the M-SPIDNA-SMC-ABC-RF process implemented

We will compare two approaches :

- On the one hand, our data $\mathbf{X}$ have S sites that are all sites with visible mutations (at least one individual has a different nucleotide from the ancestral one). We call this approach the "SNPs-only" approach.

- On the other hand we will consider data $\mathbf{X}$ with S sites that are a continuous sequence of the genome (with visible mutation, invisible ones and without any mutation). We call this approach the "full-sequence" approach.

4.2 Results on simulated data

4.2.1 Define useful metrics

In this part, our observed data $\mathbf{X}^{obs}$ for which we will try to infer the parameters θ will be generated by the model $\mathcal{M}_{FSM}$. Consequently, we will know what are the true values of θ and will be able to quantify the accuracy of our predictor $\widehat{\theta} := \mathbb{E}_{\pi^T(\ .\ |\mathbf{X}^{obs})}[\theta]$.

We can use the one dimensional relative root mean square error (RRMSE) for P experiments with $(\theta^p, \widehat{\theta}^p)_{1\leq p\leq P}$:

$$\forall\ i \in [\![1,13]\!],\ \mathbf{RRMSE}_i = \sqrt{\frac{\frac{1}{P}\sum_{p=1}^{P}\left(\theta_i^p - \widehat{\theta}_i^p\right)^2}{\frac{1}{P}\sum_{p=1}^{P}(\widehat{\theta}_i^p)^2}}$$

We will use $\mathbf{RRMSE_1}$ to quantify the accuracy of the prediction of the overall mutation rate.

Moreover, to address the capacity to infer at the same time all the transition parameters, we're going to use the average Euclidean distance (AED) and the root mean square deviation (RMSD) :

$$\mathbf{AED}_{2:13} = \frac{1}{P}\sum_{p=1}^{P}\|\theta_{2:13}^p - \widehat{\theta}_{2:13}^p\|$$

$$\mathbf{RMSD}_{2:13} = \sqrt{\frac{1}{P}\sum_{p=1}^{P}\|\theta_{2:13}^p - \widehat{\theta}_{2:13}^p\|^2}$$

By taking $\forall p \in [\![1,P]\!],\ \theta_{2:13}^p, \widehat{\theta}_{2:13}^p \overset{i.i.d.}{\sim} \mathbf{Dir}([\mathbf{1},\mathbf{1},\mathbf{1}])^{\otimes \mathbf{4}}$, which is a uniform dirichlet distribution on the transition parameters respecting the constraints detailed in 1.3, we find

$$\lim_{P\to\infty}\mathbf{AED}_{2:13} \approx 1.12$$

$$\lim_{P\to\infty}\mathbf{RMSD}_{2:13} \approx 1.15$$

Having those values in mind will allow us to have a better idea of the performance of our predictions.

It is useful to use those two metrics ($\mathbf{AED}_{2:13}$ and $\mathbf{RMSD}_{2:13}$) as the first one will give us exclusively information about an average accuracy of the prediction while the second one will also highlight occasional very bad predictions.

4.2.2 Results

We will run six different experiments with three different mutation rate ranges ($[10^{-7}, 10^{-5}]$, $[10^{-6}, 10^{-4}]$ and $[10^{-5}, 10^{-3}]$). We display below the expected number of mutations and visible mutated sites for different values of θ that will be in the studied ranges.

mutation rate θ_1	10^{-7}	10^{-6}	10^{-5}	10^{-4}	10^{-3}
$\mathbb{E}[\#Mutations\|\theta_1]$	8	82	828	8314	82735
$\mathbb{E}[\#SNPs\|\theta_1]$	8	81	744	3440	4000

Table 1: Average number of mutations and visible mutated sites (1000 simulations) for different mutation rates θ_1

We can see below the transition parameter marginals of the posterior when the mutation rate is high and there are probably multiple mutations that happened on the same sites. The predictions of the transition parameters seem very accurate. Even the prediction of the mutation rate posterior is quite good with $\mathbb{E}[\theta_1|\mathbf{X}^{obs}] \approx 0.8 \times 10^{-4}$ and $\theta_1 = 0.96 \times 10^{-4}$.

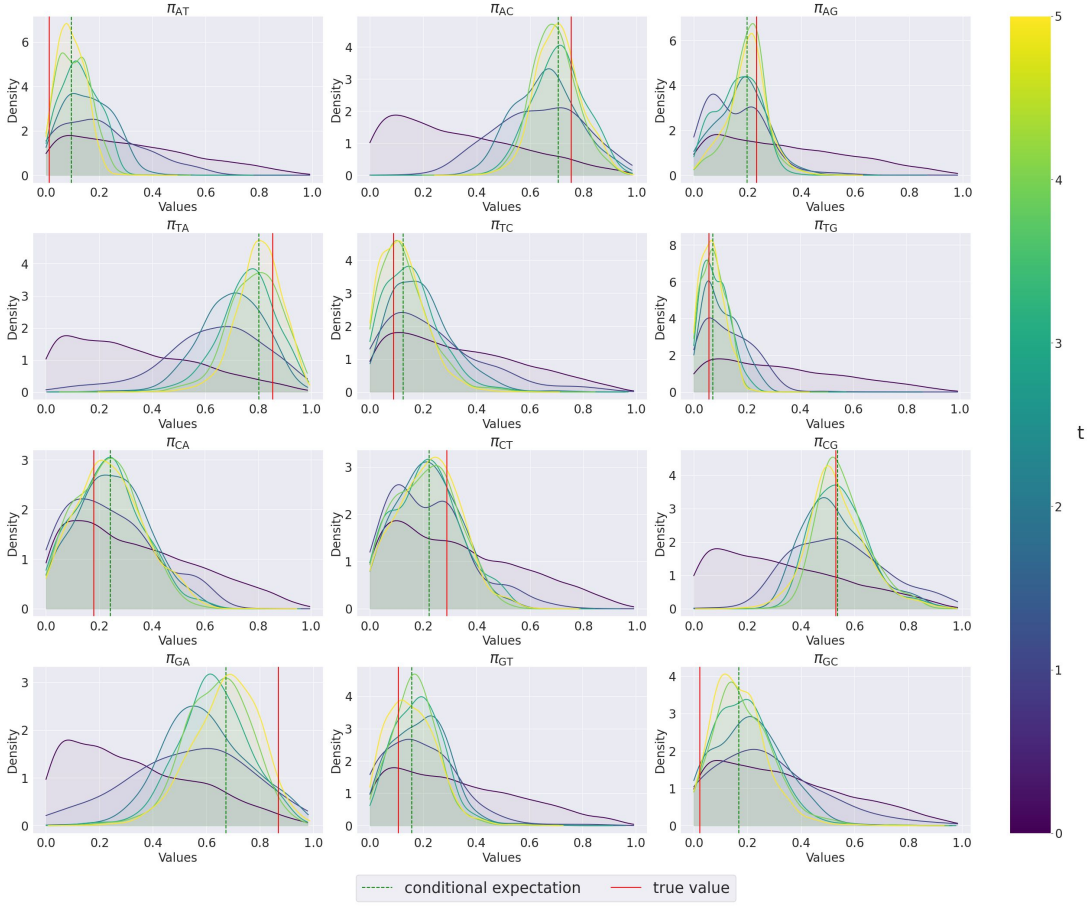


Figure 7: $\pi^t(\theta_{2:13}|\mathbf{X}^{obs})_{1\leq t\leq 5}$ - full-sequence & $\theta_1 = 0.96 \times 10^{-4}$

The fact that the posteriors are more and more precise at each iteration of the sequential process is highlighted on the Figure 13 in the supplementary material.

Let's see more precisely what are the results on multiple experiments thanks to the metrics we introduced in Section 4.2.1. We will also explain how we can validate or discuss our results with a mathematical result using Markov chain assumptions in the supplementary material (24).

mutation rate	full-sequence	SNPs-only
$[10^{-7}, 10^{-5}]$	**0.422**	0.510
$[10^{-6}, 10^{-4}]$	0.419	**0.325**
$[10^{-5}, 10^{-3}]$	**0.412**	0.488

Table 2: $\mathbf{RMSD}_{2:13}$

mutation rate	full-sequence	SNPs-only
$[10^{-7}, 10^{-5}]$	**0.410**	0.481
$[10^{-6}, 10^{-4}]$	0.394	**0.312**
$[10^{-5}, 10^{-3}]$	**0.399**	0.471

Table 3: $\mathbf{AED}_{2:13}$

Each score was determined after conducting $P = 20$ simulations and taking T = 5 iterations in the computing of the posterior. The best score is highlighted in bold.
Please refer to 4.2.1 to have the meaning of the metrics $\mathbf{AED}_{2:13}, \mathbf{RMSD}_{2:13}$, and $\mathbf{RRMSE}_1$.

mutation rate	full-sequence	SNPs-only
$[10^{-7}, 10^{-5}]$	0.343	**0.333**
$[10^{-6}, 10^{-4}]$	0.496	**0.363**
$[10^{-5}, 10^{-3}]$	0.565	**0.478**

Table 4: $\mathbf{RRMSE}_1$

According to our results, it seems that the information given by sites where there are no visible mutations are very useful for low mutation rates and high mutation rates, as it allows to have better $\mathbf{RMSD}_{2:13}$ and $\mathbf{AED}_{2:13}$ scores. However, for intermediate mutation rates where nearly no site had 2 or more mutations during the evolution, taking only the visible mutated sites allow a better inference of the parameters. It can be easily interpreted, indeed we may have already enough information with only the SNPs in this case, and the other non-SNPs sites may be noising our summary statistics more than giving useful information.

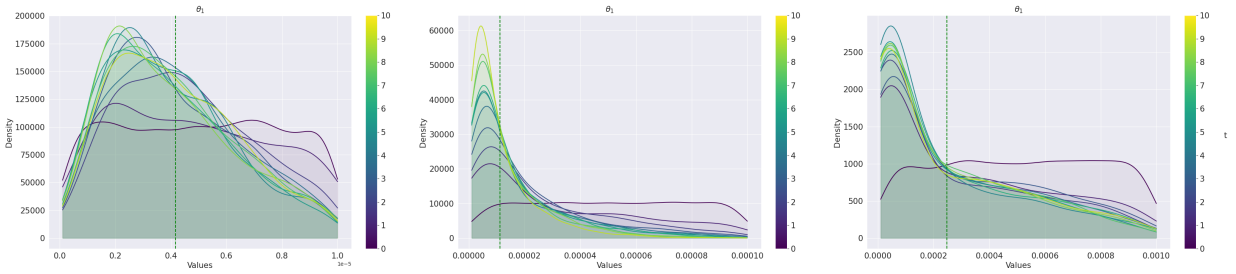
Moreover, it seems to be slightly easier to predict efficiently $\pi(\theta_{2:13}|\mathbf{X}^{obs})$ when we have a significant number of mutations as we have a better $\mathbf{RMSD}_{2:13}$ and $\mathbf{AED}_{2:13}$ for the experiments with a higher mutation rate. Finally, the results of the full-sequence approach seem more consistent as we have very close $\mathbf{RMSD}_{2:13}$ and $\mathbf{AED}_{2:13}$ scores for different mutation rates ranges.

4.3 Results on real mtDNA sequences

Now we will apply our algorithm to real data. We will take mitochondrial DNA (mtDNA) because we know that mitochondria can play a critical role in cancer. Moreover those mtDNA molecules can contain many variations among a population, and they are important in evolution to retrace the history of populations. In addition, we know that in mtDNA, many sites must have experienced more than one mutation thanks to previous studies (cf. Ward et al. 1991).

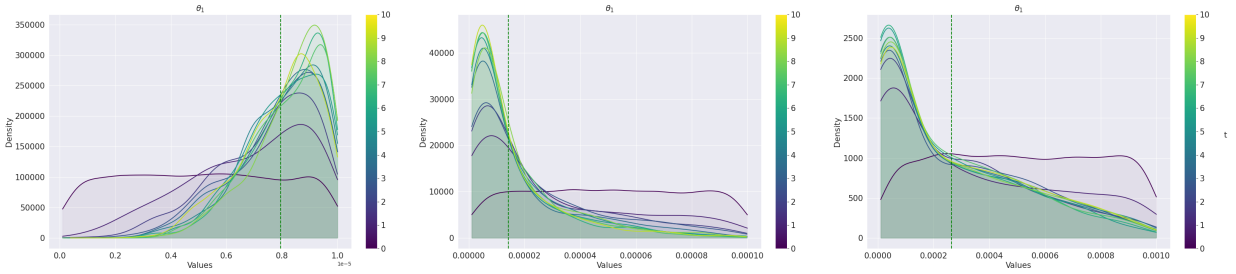
We tried to infer the 13 parameters of a $\mathcal{M}_{FSM}$ model for an observed data $\mathbf{X}^{obs}_{mtDNA}$ which consists in 4000 bases long mtDNA sequences from individuals sampled in the same haplogroup (i.e. a group of individuals with a same geographic origin and a common ancestor around 80-90ka ago). The data has been taken from the website mitomap by Lott et al. 2013. As the ancestral sequence, we used the revised Cambridge Reference Sequence, presented by Andrews et al. 1999. Then we took the $L4$ haplogroup sequences obtained by multiple single-cells DNA sequencing from different researchers and gathered at *Genbank IDs for haplogroup L4* 2023.

We ran the Algorithm 2 with the same three priors on θ_1 as in the previous Section 4.2.2 and with our two approaches "SNPs-only" and "full-sequence" and we had some interesting results.



$\pi^t(\theta_1|\mathbf{X}^{obs}_{mtDNA}, \theta_1 \in [10^{-7}, 10^{-5}])$ $\pi^t(\theta_1|\mathbf{X}^{obs}_{mtDNA}, \theta_1 \in [10^{-6}, 10^{-4}])$ $\pi^t(\theta_1|\mathbf{X}^{obs}_{mtDNA}, \theta_1 \in [10^{-5}, 10^{-3}])$

Figure 8: Mutation rate inference for $\mathbf{X}^{obs}_{mtDNA}$ (full-sequence)



$\pi^t(\theta_1|\mathbf{X}^{obs}_{mtDNA}, \theta_1 \in [10^{-7}, 10^{-5}])$ $\pi^t(\theta_1|\mathbf{X}^{obs}_{mtDNA}, \theta_1 \in [10^{-6}, 10^{-4}])$ $\pi^t(\theta_1|\mathbf{X}^{obs}_{mtDNA}, \theta_1 \in [10^{-5}, 10^{-3}])$

Figure 9: Mutation rate inference for $\mathbf{X}^{obs}_{mtDNA}$ (SNPs-only)

First, we can observe from above graphs that the 2 approaches both predict a mutation rate which is in the range $[10^{-5}, 10^{-7}]$ as for higher priors, all the weight of the posterior is concentrated at the lower bound. However, the rate is a bit overestimated (as we could expect) when we only look at the SNPs.

If we focus on the most likely predictions that are the ones from the full-sequence approach with a prior $\theta_1 \in [10^{-7}, 10^{-5}]$, we find that it gives us an already well-known piece of information concerning the human mitochondrial DNA, which is the predominance of G -> A and T -> C transitions (cf. Guo et al. 2023). Indeed, the predicted expectation for those transition parameters are respectively 0.51 and 0.55, making them two of the only three transition parameters predicted to exceed 0.5.

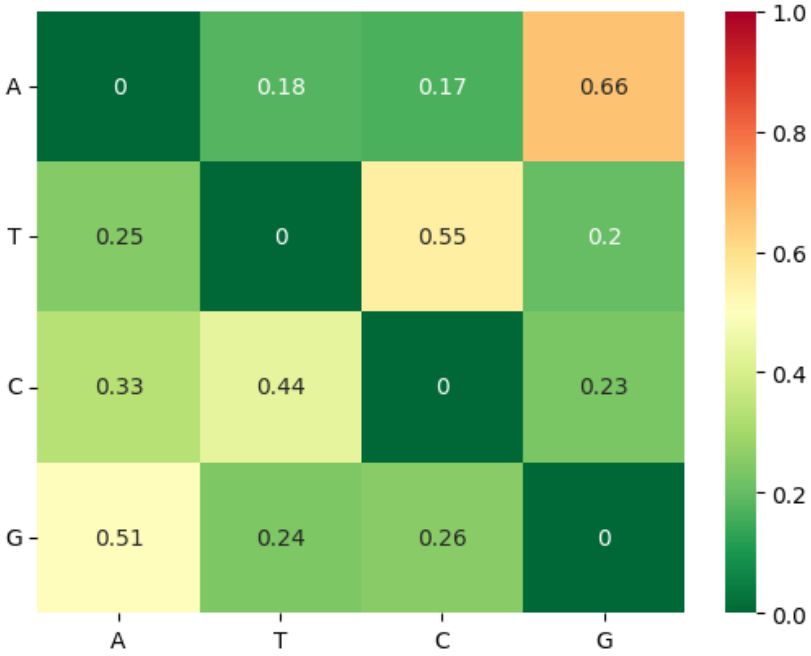


Figure 10: Predicted average transition parameters : $\mathbb{E}[\theta_{2:13}|\mathbf{X}^{obs}_{mtDNA}]$ (full-sequence)

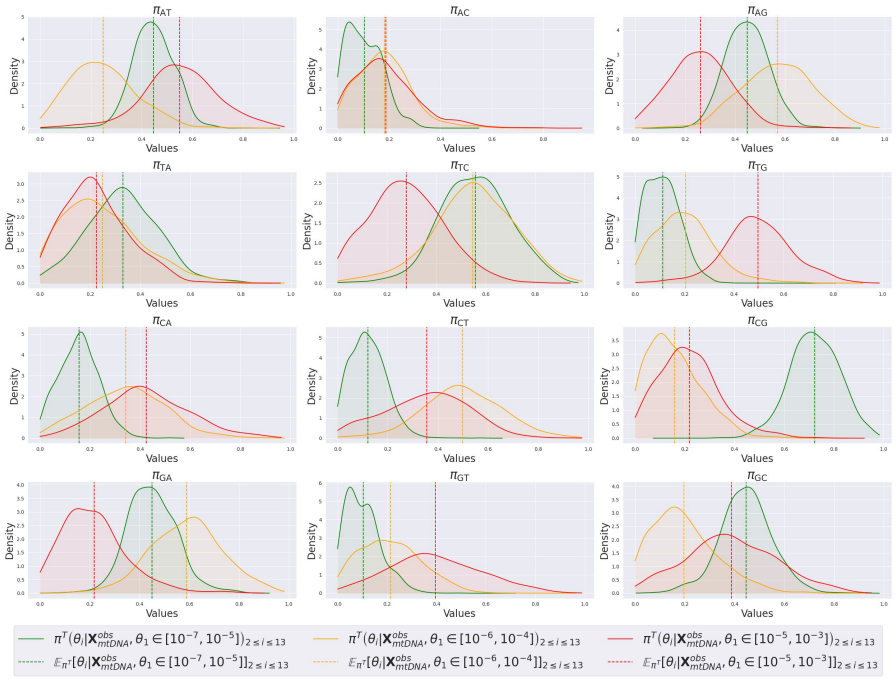


Figure 11: SNPs-only

On a second time, we will look at the estimated posterior for $\theta_{2:13}$. The predicted posterior is significantly changing depending on the prior on the mutation rate θ_1 in the only-SNPs approach (Figure 14) while it is quite consistent for the different priors on the mutation rate θ_1 in the full-sequence approach, confirming that this last approach may be more robust (Figure 12). Moreover, as we have a very low number of SNPs in $\mathbf{X}^{obs}_{mtDNA}$ (≈ 100), it would be coherent with our previous results that the full-sequence approach would be more efficient that the only-SNPs one in this case.

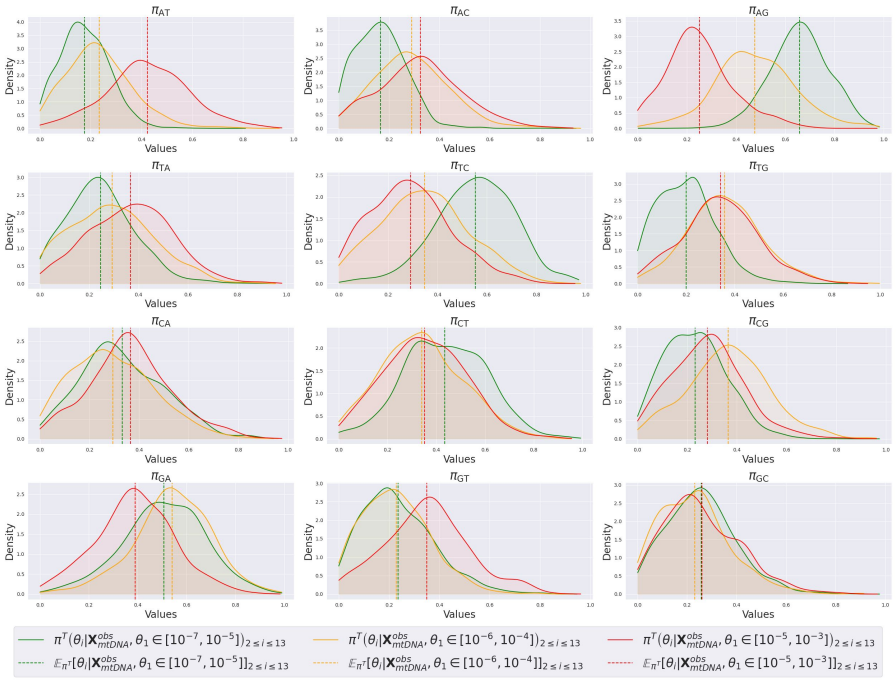


Figure 12: full-sequence

Conclusion

In this report, we tried to infer the parameters of a stochastic finite sites evolutionary model $\mathcal{M}_{FSM}$ taking as an input a mutation rate θ_1 as well as the transition parameters $\theta_{2:13}$ corresponding to the $\pi_{an}(\forall\ a \in \{A,T,C,G\}, \forall\ n \in \{A,T,C,G\}\backslash\{a\})$ and whose output are nucleotide sequences sampled on a population of individuals or cells. It is a task of great interest as it could enable a better understanding of DNA molecules in which we suspect a bias in the transition parameters as well as the ones that should be studied in a finite sites approach.

To realize the inference task, we used an adaptive sequential Monte-Carlo algorithm where at each step t we computed a more accurate approximation of the posterior distribution $\pi(\ \cdot\ |\mathbf{X}^{obs})$ for a fixed observed data $\mathbf{X}^{obs}$. At each step t of this sequential framework, we relied on the building of a random forest (based on a different training set at each step) to derive a posterior point-mass distribution $\pi^t(\ \cdot\ |\mathbf{X}^{obs})$.

This whole algorithm could have been implemented because we managed to summarize a complex genomic data $\mathbf{X} \in \mathbb{R}^{6\ 400\ 000}$ into a vector in $\mathbb{R}^{13}$ thanks to a deep learning algorithm inspired from Sanchez et al. 2021 but that we adapted to be able to deal with nucleotide sequences that differentiate the A, T, C, G bases.

While our results show quite good capacity of inferring all the transition parameters at once, there are still issues to address, points that need to be improved, and paths to explore.

First, we used here the minimal number of summary statistics needed to infer 13 parameters which is 13 summary-statistics. However, relying on the theoretical result of the Theorem 2, we could train another M-SPIDNA (but with an output of size $K > 13$) to generate K summary-statistics corresponding to the $(\mathbb{E}_\pi[f_1(\theta)|\mathbf{X}], \ldots, \mathbb{E}_\pi[f_K(\theta)|\mathbf{X}])$, and therefore being able to infer the whole posterior with an arbitrary high precision.

Secondly, we are currently quite limited by the computational cost in memory of our algorithm, and it has to be addressed if we want to be able to run it in a larger scale (thousands of datasets analysed and fitted, longer DNA sequences, bigger sampled populations). The main limitation is caused by the fact that a single data $\mathbf{X}$ containing $I = 100$ sequences of length $S = 4000$ in our $\mathcal{M}_{FSM}$ model has a size $16 \times I \times S = 6\ 400\ 000$. Indeed, the training of the M-SPIDNA (Section 3.3) needs to load multiple data $\mathbf{X}$ at once, which can be very space consuming. To meet this challenge, one idea could be to use bit-arrays to encode the nucleotide sequences (as the values are only zeros or ones) and then reduce highly the memory need to store and load data.

We provide a code with some pre-trained MSPIDNA networks, that can be directly used on real dataset or on simulated ones to infer mutation rates and transition matrices (https://github.com/mohammed-yassinehabibi/MSPIDNA-SMC-ABC-RF).

Supplemantary Material

Algorithm

Algorithm 2: M-SPIDNA-SMC-ABC-RF

Input : A prior $\pi(\theta)$; An observed data $\mathbf{X}^{obs}$; $N, N_f, T \in \mathbb{N}^*$

Output: An approximation of the posterior $\pi(\,\cdot\,|\mathbf{X}^{obs})$

Generate a training dataset for M-SPIDNA:

  for $i = 1, \ldots, N$ **do**

    Sample $\theta_i \sim \pi$

    Simulate data $\mathbf{X}_i \sim \mathcal{M}_{FSM}(\theta_i)$

Train a M-SPIDNA neural network with $\mathcal{D} = \{(\mathbf{X}_i, \theta_i)\}_{1 \leq i \leq N}$:

  $\mathcal{S} \leftarrow$ M-SPIDNA$(\mathcal{D})$ ▷ `cf. section 3.3`

Infer $\pi(\theta|\mathcal{S}(\mathbf{X}^{obs}))$ *with SMC-ABC-RF*:

  for $t = 1, \ldots, T$ **do**

    if *t>1* **then**

      $\bar{\theta}^{t-1} \leftarrow \dfrac{1}{N_f} \displaystyle\sum_{i=1}^{N_f} \theta_j^{t-1}$

      $\tau_{t-1}^2 \leftarrow \dfrac{2}{N_f} \displaystyle\sum_{i=1}^{N_f} (\theta_j^{t-1} - \bar{\theta}^{t-1})$

    for $j = 1, \ldots, N_f$ **do**

      if *t=1* **then**

        Sample $\theta_j^t \sim \pi$

      else

        Sample $\tilde{\theta}_j^{t-1} \sim \pi^{t-1}(\theta|\mathbf{X}^{obs})$

        Sample $\theta_j^t \sim \mathcal{N}(\tilde{\theta}_j^{t-1}, \tau_{t-1}^2)$ ▷ `Adaptive perturbation (section 2.4)`

        if $\pi(\theta_j^t) = 0$ **then**

          return to step 16

      Simulate $\mathbf{X}_j^t \sim \mathcal{M}_{FSM}(\theta_j^t)$

    Create a forest with B trees for the set $\{(\mathbf{X}_j^t, \theta_j^t)\}_{1 \leq j \leq N_f}$ ▷ `cf. 2.2.1 and 2.3`

    Compute weights $(w_t^j)_{1 \leq j \leq N_f}$ ▷ `cf. 2.2.2`

    $\pi^t(\theta|\mathbf{X}^{obs}) \leftarrow \displaystyle\sum_{j=1}^{N_f} w_t^j(\mathbf{X}^{obs}) \delta_{\theta_j^t}$ ▷ $\delta_{\theta_j^t}$ `is the point mass at` θ_j^t

  return $\pi^T(\theta|\mathbf{X}^{obs})$

Theorem 2 details

Let's give some details concerning the proof of Theorem 2 which is repeated below.

Theorem. *We consider the square integrable space with its inner product* $\left(\mathrm{L}^2([a,b]), \langle \cdot, \cdot \rangle_2\right)$
For a fixed $\mathbf{X}^{obs}$*, we assume that* $\pi(\,\cdot\,|\mathbf{X}) \in \mathrm{L}^2([a,b])$
Let $(f_k)_{k \in \mathbb{N}^*}$ *be an orthonormal basis of this Hilbert space.*
We define $K \in \mathbb{N}^*$ *and* $\epsilon \in \mathbb{R}_+^*$*. For simplicity purpose, we will assume here that* θ *is univariate.*
Then, by defining the $K-$*dimensional summary statistics* $\mathcal{S}(\mathbf{X}^{obs}) = \left(\mathbb{E}_\pi[f_1(\theta)|\mathbf{X}^{obs}], \ldots, \mathbb{E}_\pi[f_K(\theta)|\mathbf{X}^{obs}]\right)$

and $\pi_{ABC}^{\epsilon,K} = \pi(\theta | \|\mathcal{S}(\mathbf{X}) - \mathcal{S}(\mathbf{X}^{obs})\| < \epsilon)$ *we have,*

$$\pi_{ABC}^{\epsilon,K} \underset{\substack{\epsilon\to 0\\ K\to\infty}}{\rightharpoonup} \pi(\,\cdot\,|\mathbf{X}^{obs}) \quad \textit{(converges weakly) if we have furthermore } K^{1/2}\epsilon \to 0$$

Proof. Let f be a $\mathrm{L}^2([a,b])$ function. We can write $f = \sum_{k=1}^{\infty} \alpha_k f_k$.

We want to show $\langle \pi_{ABC}^{\epsilon,K}, f\rangle \underset{\substack{\epsilon\to 0\\ K\to\infty}}{\longrightarrow} \langle \pi(\,\cdot\,|\mathbf{X}^{obs}), f\rangle$.

$$\begin{aligned}
\left|\langle \pi_{ABC}^{\epsilon,K}, f\rangle - \langle \pi(\,\cdot\,|\mathbf{X}^{obs}), f\rangle\right| &= \left|\sum_{k=1}^{\infty} \alpha_k \left(\langle \pi_{ABC}^{\epsilon,K}, f_k\rangle - \langle \pi(\,\cdot\,|\mathbf{X}^{obs}), f_k\rangle\right)\right| \\
&\leq \sum_{k=1}^{\infty} |\alpha_k| \underbrace{\left|\int_a^b \pi_{ABC}^{\epsilon,K}(\theta) f_k(\theta)\mathrm{d}\theta - \int_a^b \pi(\theta|\mathbf{X}^{obs}) f_k(\theta)\mathrm{d}\theta\right|}_{=\left|\mathbb{E}_{\pi_{ABC}^{\epsilon,K}}[f_k(\theta)] - \mathbb{E}_{\pi}[f_k(\theta)|\mathbf{X}^{obs}]\right|} \\
&\leq \left(\max_{1\leq k\leq K} |\alpha_k|\right) \sqrt{K} \left\| \left(\mathbb{E}_{\pi_{ABC}^{\epsilon,K}}[f_k(\theta)]\right)_{1\leq k\leq K} - \mathcal{S}(\mathbf{X}^{obs}) \right\|_{\mathbb{R}^K} + R_K \\
&\quad \text{(Cauchy-Schwarz in } \mathbb{R}^K\text{)}
\end{aligned}$$

with $R_K = \sum_{k=K+1}^{\infty} |\alpha_k| \left|\left\langle \pi_{ABC}^{\epsilon,K} - \pi(\,\cdot\,|\mathbf{X}^{obs}), f_k \right\rangle\right|$.

Let's show that $R_K \underset{K\to\infty}{\longrightarrow} 0$

$$R_K = \sum_{k=K+1}^{\infty} |\alpha_k| \left|\left\langle \pi_{ABC}^{\epsilon,K} - \pi(\,\cdot\,|\mathbf{X}^{obs}), f_k \right\rangle\right| \leq \sqrt{\sum_{k=K+1}^{\infty} |\alpha_k|^2} \times \sqrt{\sum_{k=K+1}^{\infty} \left|\left\langle \pi_{ABC}^{\epsilon,K} - \pi(\,\cdot\,|\mathbf{X}^{obs}), f_k \right\rangle\right|^2}$$

(Cauchy-Schwarz)

Yet we know $\begin{cases} \|f\|_2^2 = \sum_{k=1}^{\infty} |\alpha_k|^2 < \infty \\ \left\|\pi_{ABC}^{\epsilon,K} - \pi(\,\cdot\,|\mathbf{X}^{obs})\right\|_2^2 = \sum_{k=1}^{\infty} \left|\left\langle \pi_{ABC}^{\epsilon,K} - \pi(\,\cdot\,|\mathbf{X}^{obs}), f_k \right\rangle\right|^2 < \infty \end{cases}$

So by comparison, $R_K \underset{K\to\infty}{\longrightarrow} 0$.

We define $A \coloneqq \left(\max_{k\in\mathbb{N}^*} |\alpha_k|\right)$ (that exists because $\alpha_k \underset{K\to\infty}{\longrightarrow} 0$).
We have

$$\left|\langle \pi_{ABC}^{\epsilon,K}, f\rangle - \langle \pi(\,\cdot\,|\mathbf{X}^{obs}), f\rangle\right| \leq A\sqrt{K} \left\| \left(\mathbb{E}_{\pi_{ABC}^{\epsilon,K}}[f_k(\theta)]\right)_{1\leq k\leq K} - \mathcal{S}(\mathbf{X}^{obs}) \right\|_{\mathbb{R}^K} + o_K(1)$$

Let's show that $\left\| \left(\mathbb{E}_{\pi_{ABC}^{\epsilon,K}}[f_k(\theta)]\right)_{1\leq k\leq K} - \mathcal{S}(\mathbf{X}^{obs}) \right\|_{\mathbb{R}^K} \leq \epsilon$

$$
\begin{aligned}
\mathcal{S}(\mathbf{X}) &= \mathbb{E}_\pi[\mathcal{S}(\mathbf{X})|\mathcal{S}(\mathbf{X})] \\
&= \mathbb{E}_\pi[(\mathbb{E}_\pi[f_k(\theta)|\mathbf{X}])_{1\le k\le K}\,|\mathcal{S}(\mathbf{X})] \qquad \text{(definition of } \mathcal{S}) \\
&= (\mathbb{E}_\pi[f_k(\theta)|\mathcal{S}(\mathbf{X})])_{1\le k\le K} \qquad \text{(Tower property)}
\end{aligned}
$$

Then, we define $E_\epsilon = \{\|\mathcal{S}(\mathbf{X}) - \mathcal{S}(\mathbf{X}^{obs})\|_{\mathbb{R}^K} < \epsilon\}$.

$$
\begin{aligned}
\text{We have } \mathbb{E}_\pi[(f_k(\theta))_{1\le k\le K}\mathbb{1}_{E_\epsilon}] &= \mathbb{E}_\pi[\mathbb{E}_\pi[(f_k(\theta))_{1\le k\le K}|\mathcal{S}(\mathbf{X})]\mathbb{1}_{E_\epsilon}] \qquad \text{(because } E_\epsilon \in \sigma(\mathcal{S}(\mathbf{X}))) \\
&= \mathbb{E}_\pi[\mathcal{S}(\mathbf{X})\mathbb{1}_{E_\epsilon}]
\end{aligned}
$$

$$
\begin{aligned}
\text{Finally } \left\|\left(\mathbb{E}_{\pi^{\epsilon,K}_{ABC}}[f_k(\theta)]\right)_{1\le k\le K} - \mathcal{S}(\mathbf{X}^{obs})\right\|_{\mathbb{R}^K} &= \left\|(\mathbb{E}_\pi\left[f_k(\theta)|E_\epsilon\right])_{1\le k\le K} - \mathcal{S}(\mathbf{X}^{obs})\right\|_{\mathbb{R}^K} \\
&= \left\|\mathbb{E}_\pi\left[\mathcal{S}(\mathbf{X}) - \mathcal{S}(\mathbf{X}^{obs})|E_\epsilon\right]\right\|_{\mathbb{R}^K} \\
&\le \mathbb{E}_\pi\left[\left\|\mathcal{S}(\mathbf{X}) - \mathcal{S}(\mathbf{X}^{obs})\right\|_{\mathbb{R}^K}|E_\epsilon\right] \\
&< \epsilon
\end{aligned}
$$

Therefore, we have

$$
\left|\langle \pi^{\epsilon,K}_{ABC}, f\rangle - \langle \pi(\,\cdot\,|\mathbf{X}^{obs}), f\rangle\right| \le A\sqrt{K}\epsilon + o_K(1)
$$

□

Assess results using Markov chain assumptions

Let's define a transition matrix $\Pi = (\pi_{ij})_{1\le i,j\le 4}$ associated to a Markov-chain process.

After K steps, we have a transition matrix $Q = \Pi^K$. We assume that $K\|\Pi - \mathrm{I}\| << 1$ (at each step the state has a very high probability to stay the same). We assume that we observe Q and we want to deduce Π.

$$
\begin{aligned}
Q &= (\mathrm{I} + (\Pi - \mathrm{I}))^K \\
&= \mathrm{I} + K(\Pi - \mathrm{I}) + \frac{K(K-1)}{2}(\Pi - \mathrm{I})^2 + o(\|\Pi - \mathrm{I}\|^2) \\
&= \mathrm{I} + K(\Pi - \mathrm{I}) + \frac{[K(\Pi - \mathrm{I})]^2}{2} + o([K\|\Pi - \mathrm{I}\|]^2)
\end{aligned}
$$

If we rename $R \coloneqq Q - \mathrm{I}$ and $A \coloneqq K(\Pi - \mathrm{I})$, we have

$$
R \approx A + \frac{1}{2}A^2 \tag{1}
$$

.

At the first order, $R = A + H$ with $H = o(A)$ so $A = R - H$.
We want to express H as a function of R instead of A. We have thanks to equation (1) :

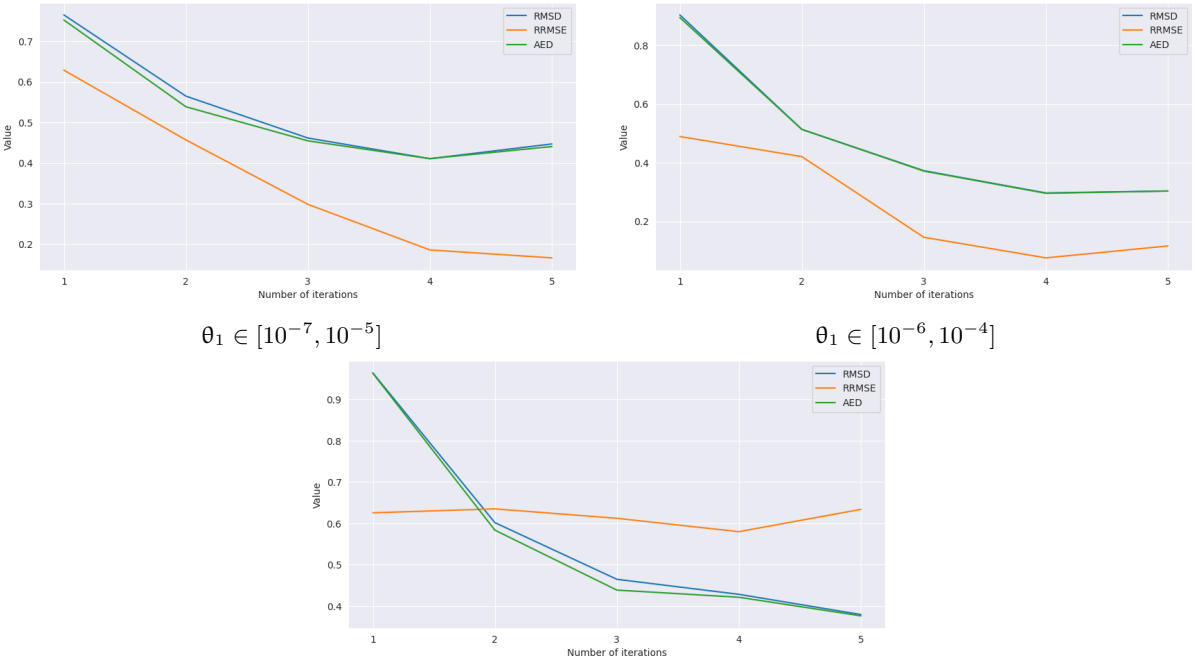
$$
R = R - H + \frac{1}{2}(R-H)^2 = R - H + \frac{1}{2}R^2 + o(A^2) \qquad \text{(because } R = O(A))
$$

So $H = \frac{1}{2}R^2 + o(A)$, and $A = R - \frac{1}{2}R^2 + o(A^2)$.

It follows that $\Pi = \mathrm{I} + \dfrac{Q - \mathrm{I}}{K} - \dfrac{(Q-\mathrm{I})^2}{2K} + o([K\|\Pi - \mathrm{I}\|]^2)$.

By assigning Π's diagonal to 0, and by noting that the non-diagonal parameters are all depending on K only by a linear factor $\dfrac{1}{K}$, we can then easily transform Π to have an approximation of the $\mathcal{M}_{FSM}$ model's transition matrix (Figure 3) that can be used to assess our model's predictions in mean.

Additional figures



$\theta_1 \in [10^{-7}, 10^{-5}]$ $\theta_1 \in [10^{-6}, 10^{-4}]$

$\theta_1 \in [10^{-5}, 10^{-3}]$

Figure 13: Improvement of the inference metrics after each iteration of SMC-ABC-RF

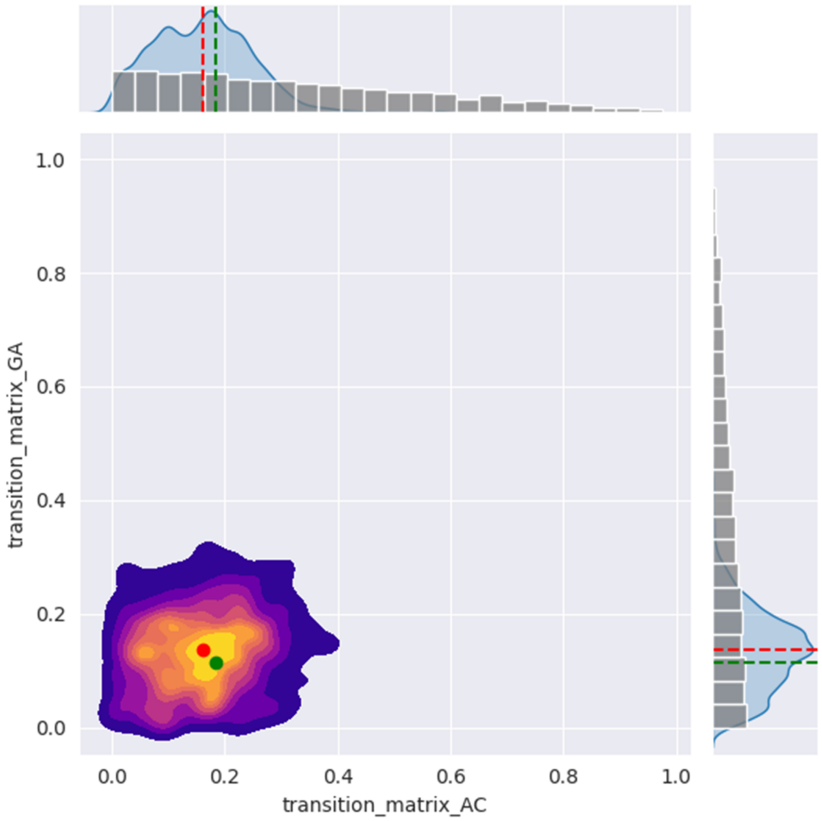


Figure 14: Projection in two dimensions of a posterior distribution

References

Sanchez, Théophile et al. (2021). "Deep learning for population size history inference: Design, comparison and combination with approximate Bayesian computation". In: *Molecular Ecology Resources* 21.8, pp. 2645–2660. DOI: https://doi.org/10.1111/1755-0998.13224. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1755-0998.13224. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1755-0998.13224.

Tavaré, S et al. (Feb. 1997). "Inferring coalescence times from DNA sequence data". In: *Genetics* 145.2, pp. 505–518.

Fu, Y X and W H Li (Feb. 1997). "Estimating the age of the common ancestor of a sample of DNA sequences." In: *Molecular Biology and Evolution* 14.2, pp. 195–199. ISSN: 0737-4038. DOI: 10.1093/oxfordjournals.molbev.a025753. eprint: https://academic.oup.com/mbe/article-pdf/14/2/195/11165471/9fu.pdf. URL: https://doi.org/10.1093/oxfordjournals.molbev.a025753.

Kingman, J.F.C. (1982). "The coalescent". In: *Stochastic Processes and their Applications* 13.3, pp. 235–248. ISSN: 0304-4149. DOI: https://doi.org/10.1016/0304-4149(82)90011-4. URL: https://www.sciencedirect.com/science/article/pii/0304414982900114.

Baumdicker, Franz et al. (2022). "Efficient ancestry and mutation simulation with msprime 1.0". In: *Genetics* 220.3, iyab229.

Raynal, Louis et al. (Oct. 2018). "ABC random forests for Bayesian parameter inference". In: *Bioinformatics* 35.10, pp. 1720–1728. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bty867. eprint: https://academic.oup.com/bioinformatics/article-pdf/35/10/1720/48969808/bioinformatics_35_10_1720.pdf. URL: https://doi.org/10.1093/bioinformatics/bty867.

Ćevid, Domagoj et al. (Jan. 2022). "Distributional random forests: heterogeneity adjustment And multivariate distributional regression". In: *J. Mach. Learn. Res.* 23.1. ISSN: 1532-4435.

Dinh et al. (2024a). *Approximate Bayesian Computation sequential Monte Carlo via random forests.* arXiv: 2406.15865 [stat.CO]. URL: https://arxiv.org/abs/2406.15865.

Kelleher, Jerome, Alison M Etheridge, and Gilean McVean (May 2016). "Efficient Coalescent Simulation and Genealogical Analysis for Large Sample Sizes". In: *PLOS Computational Biology* 12.5, pp. 1–22. DOI: 10.1371/journal.pcbi.1004842. URL: https://doi.org/10.1371/journal.pcbi.1004842.

Kimura, M (Apr. 1969). "The number of heterozygous nucleotide sites maintained in a finite population due to steady flux of mutations". In: *Genetics* 61.4, pp. 893–903.

Mathew, Lisha A et al. (Oct. 2013). "Why to account for finite sites in population genetic studies and how to do this with Jaatha 2.0". In: *Ecol. Evol.* 3.11, pp. 3647–3662.

Konrad, Anke et al. (Jan. 2017). "Mitochondrial mutation rate, spectrum and heteroplasmy in Caenorhabditis elegans spontaneous mutation accumulation lines of differing population size". In: *Mol. Biol. Evol.*, msx051.

Sisson, S A, Y Fan, and M A Beaumont (2018). *Overview of Approximate Bayesian Computation.* arXiv: 1802.09720 [stat.CO].

Biau, Gérard, Frédéric Cérou, and Arnaud Guyader (2015). "New insights into Approximate Bayesian Computation". In: *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques* 51.1, pp. 376–403. DOI: 10.1214/13-AIHP590. URL: https://doi.org/10.1214/13-AIHP590.

Breiman, Leo (2001). "Random Forests". In: *Mach. Learn.* 45.1, pp. 5–32.

Gretton, Arthur et al. (2007). "A kernel method for the two-sample problem". In: *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*. The MIT Press.

Sisson, S A, Y Fan, and Mark M Tanaka (Feb. 2007). "Sequential Monte Carlo without likelihoods". In: *Proc. Natl. Acad. Sci. U. S. A.* 104.6, pp. 1760–1765.

Beaumont et al. (2009). "Adaptive approximate Bayesian computation". In: *Biometrika* 96.4, pp. 983–990. ISSN: 00063444, 14643510. URL: http://www.jstor.org/stable/27798882 (visited on 06/18/2024).

Cappé, O et al. (Dec. 2004). "Population Monte Carlo". In: *J. Comput. Graph. Stat.* 13.4, pp. 907–929.

Jiang, Bai et al. (2017). "Learning summary statistic for Approximate Bayesian Computation via deep neural network". In: *Statistica Sinica* 27.4, pp. 1595–1618. ISSN: 10170405, 19968507. URL: http://www.jstor.org/stable/26384090 (visited on 06/25/2024).

Zaheer, Manzil et al. (2017). "Deep Sets". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

Michel, Loris and Domagoj Cevid (2021). *drf: Distributional Random Forests*. R package version 1.1.0. URL: https://CRAN.R-project.org/package=drf.

Dinh et al. (2024b). *Approximate Bayesian Computation sequential Monte Carlo via random forests*. R package version 1.0.0. URL: https://github.com/dinhngockhanh/abcsmcrf.

Ward, R H et al. (Oct. 1991). "Extensive mitochondrial diversity within a single Amerindian tribe". In: *Proc. Natl. Acad. Sci. U. S. A.* 88.19, pp. 8720–8724.

Lott, Marie T et al. (Dec. 2013). "MtDNA variation and analysis using Mitomap and Mitomaster". In: *Curr. Protoc. Bioinformatics* 44.1, pp. 1.23.1–26. URL: http://www.mitomap.org.

Andrews, R M et al. (Oct. 1999). "Reanalysis and revision of the Cambridge reference sequence for human mitochondrial DNA". In: *Nat. Genet.* 23.2, p. 147.

Genbank IDs for haplogroup L4 (2023). Accessed: 2024-07-23. URL: https://www.mitomap.org/cgi-bin/haplo_group?data=1&hg=L4&exact=&hpp=111.

Guo, Xiaoxian et al. (Jan. 2023). "High-frequency and functional mitochondrial DNA mutations at the single-cell level". In: *Proc. Natl. Acad. Sci. U. S. A.* 120.1, e2201518120.

PyTorch (2024). *SmoothL1Loss*. Accessed: 2024-07-30. URL: https://pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html.