Unbiased MCMC sampling in Phylogenetics

Riccardo Uslenghi

Contents

Abstract

In this project we explored the limitations that mainstream MCMC sampling presents when used with bad initialization and short chains. To avoid the increase in bias caused by these two latter conditions we implemented the Unbiased MCMC algorithm, based on Maximal Coupling, proposed by PierreJacob et al [1]. This algorithm exploits the computation of many short chains and few rare, but pivotal, long chains. For this reason it is particularly suitable for parallel computations, and it will represent an important resource for MCMC sampling if computer power will grow through the number of processors in the following years [2]. We also enabled the interaction between the Unbiased sampling algorithm and the open source software BEAST2, a program for Phylogenetic analysis. This connection allowed us to produce an estimate on the evolution of the infected population size of an Hepatitis C epidemic that took place in Egypt between the 1950 and the 1993.

1 Introduction

The aim of this paper is to focus on Unbiased MCMC sampling. The following introduction is then a stand-alone in this sense, as it is the only part of the paper that does not focus on MCMC sampling. The purpose of this introduction is to put things into perspective. In particular, we will give a short description of the computational biology framework producing the probability function that we will sample from using Unbiased MCMC in section 5.

1.1 The Coalescent model

Suppose we were told that, on a desert island, there is a fixed number N of individuals belonging to the same species. If we were to meet, purely randomly and independently, two of these individuals, and discovered that those two were brothers, what would this tell us about the total number N of individuals on the island? Our intuition correctly suggests that this number N must probably be small, since the first two individuals we encountered (purely randomly and independently) are in fact brothers. This is a simple example of how having information about the genetic data of individuals can be the key to find more information about the group they belong to. The intuition underlying this idea is actually very powerful. It can be used, for example, to estimate the total number of people infected by a virus by sampling genetic data from a restricted portion of the infected population. The coalescent model offers a framework to make such estimates, drawn from observation of genetic data, rigorous. One way to derive the coalescent model is given by the Wright-Fisher process. The graphical representation of its framework is depicted in Figure 1 [3].

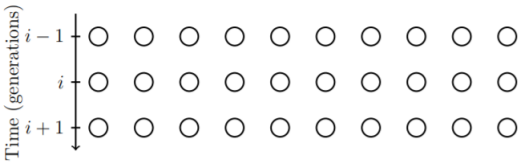

Figure 1: Wright-Fisher framework

Each of the white dots in the above image represents an individual of the population. The time axis lays on the left side of the figure (each row belonging to the same time is a generation of individuals). Every individual at time $i+1$ has exactly one parent, randomly assigned with uniform probability from the previous generation i. Parental relationships are represented by straight lines, Figure 2 [3]. The shape of the framework in Figure 1 imposes an unrealistic constraint on the population's form, as it implies that by time $i+1$ every individual that was alive at time i is dead. Also, the number of individuals must remain constant throughout all the process. This sounds clearly unsuitable for our purpose, that is analyzing quantitatively the *evolution* of an epidemic. In fact we would like to recover the dependence of the number of infected individuals with respect to time, $N(t)$, but how can we do it if the model we are using imposes a constant

number of infected individuals as a constraint? In practice it is possible to circumvent this problem by dividing the time-span of interest in a fixed number of sub-intervals [1] and by assuming the function $N(t)$ to be a piece-wise constant function (which is always approximately true for a number of sub-intervals that is large enough).

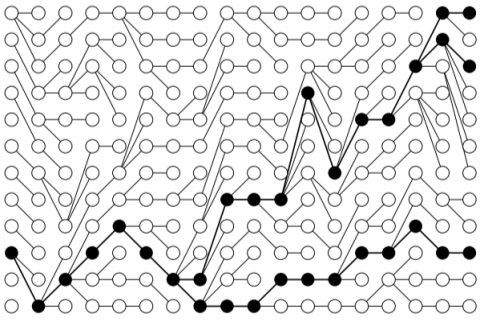

Figure 2: Visual representation of parental bonds.

Let us make a concrete example of this subdivision in sub-intervals. In our analysis, of a Hepatitis C data set collected in Egypt between 1950 and 1993 [4], we divided the time-span of interest in two regions, namely before and after 1950. This choice is supported by the shape of the ground-truth population size relative to the said data set, depicted in Figure 3 [3]. The shape shows that it is safe to assume the function to be constant on the two sub-intervals (1993-1950) and (1950-1743). Our aim then is to determine the values N_1 and N_2 that the piece-wise constant function assumes on such two sub-intervals.

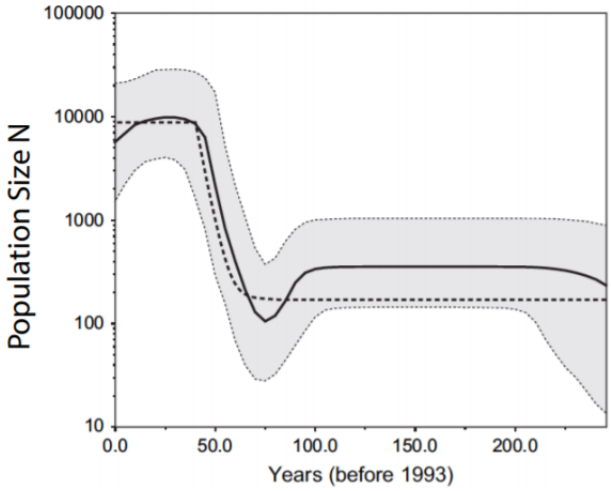

Figure 3: Ground-truth population size obtained via standard biased MCMC sampling. Obtained by applying the coalescent model to 40 different sub-intervals, dividing the relevant time-span of 250 years.

The population size in Figure 3 was obtained [3], via standard (biased) MCMC sampling, by working on the same data set that we consider in section 5. Due to the high confidence level of this result, we will regard it as our ground-truth population size. This will then allow us to see if the Unbiased MCMC sampling returns a correct (i.e. coherent with the ground-truth function) estimate of N_1 and N_2. Let us now see how we can recover estimates for the N_1 and N_2 values. In order for us to do that we have to introduce the concept of *Phylogenetic tree*. Consider the black dots at the right end of Figure 2. These are all connected to the same single common dot (a common ancestor) by a set of straight lines collapsing to a single common point at the seventh time step (counting from the left to the right). The set of straight lines we just described, up to the single common point, is said to be the Phylogenetic tree of the three black dots on the right end of the Fisher-Wright model of Figure 2. Now, given a specific set of genetic data, this

[1] "How can we pick the correct number of sub-intervals?" We could use an "elbow"-like technique. That is, we keep increasing the number of sub-intervals until the shape of the function $N(t)$ appears not to change significantly after a given number of increases in the number of sub-intervals.

data-set will be used to gain information regarding the Phylogenetic tree, that will then be used to inform the population size. This process results in an estimate for N_1 and N_2 given by the probability density of having a specific Phylogenetic tree T underlying the production of the observed genetic sequences from a common ancestor, given the values of N_1 and N_2. This is provided by

$$P(T|\vec{N}g) = \prod_{i=1}^{n-1} \exp\Big[-(t_i - t_{i-1})\binom{k_{i-1}}{2}\frac{1}{N_i g}\Big]\frac{1}{N_i g} \tag{1}$$

This formula is derived for large values of N_i, in the continuous time generalization of the Fisher-Wright model called Kingman's Coalescent, by combinatorial considerations [3]. In Equation 1, g stands for the constant time difference between consecutive generations of individuals, $(t_i - t_{i-1})$ stands for the length of the i-th sub-interval over which we suppose the population size to have the constant value N_i, and k_{i-1} is the number of lineages that we have at a time immediately following time t_{i-1}. Now, our analysis will focus on determining estimates of N_1 and N_2 by sampling from a probability density function for these two parameters. To obtain the said probability density function we apply Bayes' Theorem to Equation 1, where we omit g as we consider it to be a fixed parameter.

$$P(\vec{N}|T) = \frac{P(T|\vec{N})P(\vec{N})}{\sum_{\vec{N'}} P(T|\vec{N'})P(\vec{N'})} \tag{2}$$

Then, if we consider the Phylogenetic tree T to be fixed, this yields the desired probability function to sample from. In section 5, in fact, we made this fixed Phylogenetic tree assumption to derive estimates on N_1 and N_2 by sampling from the probability density function in Equation 2. It is worth emphasizing that the genetic data encoded in the sequences seems to be unused, but it is actually fundamental. In fact the genetic data was used to provide the Phylogenetic tree that we assume to be fixed in our analysis. It is also important to notice that the denominator in Equation 2, although being very expensive to compute, is constant for any fixed tree. We will see shortly after that there is actually no need for us to compute exactly this denominator. Let us then see how we can, in practice, employ methods to sample from probability density functions.

2 Ordinary MCMC sampling

2.1 Ordinary MCMCs

If we want to sample from a given probability density function, we can use many different methods. In some cases the primitive of the probability density function has an inverse function. In such situations, as for Gaussians and exponentially decaying distributions, the problem can be solved exactly, with maximal efficiency. If this method does not work though, we might think about using the Rejection method. But again, there could be problems, since the distribution might be strongly peaked somewhere. This shaping issue would in fact result in a very inefficient sampling, since most of the proposed points would not fall under the peak of the distribution and would consequently be rejected. Another important problem might be that we know the probability function only up to a constant. This is the case, for example, in statistical mechanics where the probability of finding a system (that satisfies the Canonical Ensemble Hypothesis) in a given state x is:

$$\pi(x) = \frac{\exp\left(-\beta H(x)\right)}{Z(\beta)} \tag{3}$$

Where $Z(\beta)$, the partition function, is in most cases unknown. In such a situation the lack of knowledge of the normalization constant prevents us from using the Rejection method, since it can only be implemented if we know, exactly, the probability distribution function. Most importantly the same problem just exposed is the one we face in our analysis. In fact we already mentioned that computing the denominator in Equation 2 is an expensive operation, we need an ulterior method then. The alternative technique that we will consider relies on the Metropolis algorithm, that is based on simulated Markov chains. The Metropolis algorithm belongs to a wider family of algorithms, called "MCMCs", i.e., Monte Cardo Markov Chains. A Markov

Chain is a succession of values where the $(i+1)$-th value's statistics only depends on the i-th value. We will refer to the value of the Markov Chain at time t as its *state* $X(t)$. As we previously mentioned, a MCMC is characterised by its "*loss of memory*", as only the last value attained by the state X is taken into account when performing the next step. Formally this means:

$$P(X(t_{i+1})|X(t_i), X(t_{i-1}), \dots, X(t_0)) = P(X(t_{i+1})|X(t_i)) \tag{4}$$

Where $P(X(t_k)|\dots)$ stands for the conditional probability with which the variable X will attain a specified value at time t_k. The idea of the MCMC sampling is to perform a random walk in the states' space and to average over (some) values, attained by the state over the random walk, in order to compute ensemble averages[5]. Importantly, the random walk is actually a Markov Chain, i.e., a stochastic process defined by Equation 4. This, however, is not the only constraint that we are going to impose on the structure of the random walk. In particular we require that, if the state at time t_i is $X(t_i) = q$, the probability $W(q \to p)$ that at the next time step t_{i+1} the state will attain a different specific value $X(t_{i+1}) = p$ must be given by:

$$W(q \to p) = T(q \to p)A(q \to p) \tag{5}$$

The terms $T(q \to p)A(q \to p)$ in Equation 5 are to be interpreted, respectively, as the proposal probability and the acceptance probability for the change of the state's value $(q \to p)$. As further constraints we also ask for $W(q \to p) > 0 \ \forall q, p$ (as we want to be able, in principle, to sample from the whole state space) and for $T(q \to p) = T(p \to q)$ (reversibility condition). Now, we will not tackle the problem directly, instead, we will try to deal with it in a simplified form. In particular, suppose we are able to sample $X(t_i)$ according to the probability distribution π, that is the probability distribution we desire to sample from [2], can we produce a new sample $X(t_{i+1})$ that will still be distributed according to π? More formally, can we find a constraint on the function W that will grant the following equation to hold?

$$P(X(t_i) = v^*) = \pi(v^*) \implies P(X(t_i) = v^*) = P(X(t_{i+1}) = v^*) = \pi(v^*) \tag{6}$$

In equation 6 we can make use of the following expansion of the $P(X(t_i + 1) = v^*)$ term to obtain the constraint on W that we are searching for.

$$P(X(t_{i+1}) = v^*) = P(X(t_i) = v^*)\left[1 - \sum_{w^* \neq v^*} W(v^* \to w^*)\right] + \sum_{w^* \neq v^*} P(X(t_i) = w^*)W(w^* \to v^*) \tag{7}$$

This is the correct expansion, as the probability of being in state v^* at the time step t_{i+1} is obtained as the sum of the probability of already being in v^* and not moving away from it, plus the probability of reaching v^* starting from another state, $w^* \neq v^*$. By imposing Equation 6 to hold, with $P(X(t_i) = \alpha) = \pi(\alpha)$, we can derive the following equation:

$$\sum_{w^* \neq v^*} \pi(v^*)W(v^* \to w^*) = \sum_{w^* \neq v^*} \pi(w^*)W(w^* \to v^*) \tag{8}$$

A sufficient, but not necessary, condition for Equation 8 to hold is that the Detailed Balance condition, in Equation 9, is satisfied:

$$\pi(v^*)W(v^* \to w^*) = \pi(w^*)W(w^* \to v^*) \tag{9}$$

Which can be rewritten as:

$$\frac{\pi(v^*)}{\pi(w^*)} = \frac{A(w^* \to v^*)}{A(v^* \to w^*)} \tag{10}$$

Where $T(v^* \to w^*)$ and $T(w^* \to v^*)$ have been simplified due to the reversibility requirement. This equation does not fix completely the shape of the acceptance probability function A. We will focus on

[2]This is clearly not the case, as assuming this would be like trying to prove a theorem by assuming that its thesis holds as an ansatz. Although the objection just exposed is correct, we will still make the assumption that we are able to sample $X(t_i)$ according to π. As we will see later on this, *prima facie*, silly requirement will bring us to the solution of the problem.

a particular definition of A, prescribed by the *Metropolis Algorithm*, that satisfies Equation 10. However this does not exclude different implementations, see for example Glauber's Algorithm[5]. The *Metropolis Algorithm* prescribes the following shape for the acceptance probability function:

$$A(v^* \to w^*) = \min\left[1, \frac{\pi(w^*)}{\pi(v^*)}\right] \tag{11}$$

As we intuitively expected, Equation 11 suggests that the MCMC is more likely to *"climb up"* probability distributions, and less likely (but not completely unwilling) to "*climb down*" them. We are finally done answering the simplified question, as we have found a method to extract $X(t_{i+1})$ from π granted that $X(t_i)$ is extracted from π as well. We only need to perform updates of the X state's value according to a $W(q \to p) = T(q \to p)A(q \to p)$ characterized by a reversible proposal probability T and by an acceptance probability A that satisfies Equation 10.

Now, the prescription depicted in Equation 10 seems actually pretty useless, as we are clearly unable to draw $X(t_i)$ from π. However, the following heuristic argument will show that there is actually no need to draw $X(t_i)$ from π to obtain valid samples. Imagine drawing two MCMCs X and Y, with the value of $X(t_0)$ being drawn from a generic probability distribution σ that we are able to sample from in practice, and with the value of $Y(t_0)$ being sampled according to π, that is the distribution we want to sample from. Let evolve both chains X and Y according to a specific function W that satisfies Equation 9. Then Y will continue to produce samples drawn from π while X will move, without any apparent meaningful reason, within the domain. As these two chains continue to move for a potentially infinite amount of time, X will eventually get "close" to Y. Let us consider the extreme example in which, at a specific time τ, the chains actually meet, i.e., $X(\tau) = Y(\tau)$. This is actually impossible if the domain of the MCMCs is not discrete (and if we are evolving the chains independently, i.e. not according to Maximal Coupling for example). However, the argument given hereby, is an heuristic one and can be made rigorous with some careful adjustments [6]. Now, the sampling of $Y(\tau + 1)$ and $X(\tau + 1)$ only depends on their previous values $X(\tau) = Y(\tau)$ and on the function W, which is the same for both chains. This means that $Y(\tau + 1)$ and $X(\tau + 1)$ are sampled exactly according to the same mechanism. Since $Y(\tau + 1)$ is drawn from π (because W satisfies Equation 9 and $Y(t_0)$ was sampled from π) this means that $X(\tau+1)$ must also be drawn from π! And all the successive samples $X(t)$ with $t > \tau$ will continue to be valid samples drawn from π. This argument, when proved rigorously, yields the following result:

$$\lim_{t_i \to \infty} P(X(t_i) = v^*) = \pi(v^*) \tag{12}$$

where X is a MCMC, randomly initialized (not necessarily according to π) and driven by a W function that satisfies Equation 10. Thus we have found a method that allows us to sample from probability density functions. It is also important to notice that if $A(v^* \to w^*)$ is defined according to Equation 11 any constant denominator in the definition of π will cancel out.

Now, as Equation 12 suggest, one should in principle wait an infinite amount of (Markov) time t_∞ to say that $X(t_\infty)$ is a proper sample extracted from π. Unfortunately this is clearly not feasible in practice. To deal with this problem a technique called *Burn-in* method is often used. The idea of the method relies on the fact that, as (Markov) time t_i grows, the chances of correctly sampling $X(t_i)$ from π increase; as it is intuitively suggested by the heuristic argument we previously gave. The *Burn-in* method consists then in waiting a specified number n of time steps before considering $X(t_i)$ to be a valid sample extracted from π. This method, however, does not offer theoretical guarantees as it does not imply that Equation 12 holds. In the following section we will see what the limitations of the *Burn-in* method are, and later on we will also describe a method that recovers the theoretical guarantees we desire.

2.2 What might go wrong?

Suppose we consider a mono dimensional probability density function π we want to sample from, using the Metropolis Algorithm. We ask ourselves: *"Is there something that might go wrong with the MCMC sampling technique we just described?"*. The answer to this question is *"yes"*. In fact the initialization, i.e.,

the choice of the starting point for our Markov chains, can cause issues. Consider the situation in which π is the the red distribution depicted in Figure 4, which is obtained as the sum of two Gaussian distributions of unit variance, centered in $\pm$ 4. If we take the starting points of our MCMCs on the very far right of the red distribution, say according to the initial distribution $\mathcal{N}(10, 1)$, we intuitively expect the following mechanism to take place: the MCMC sampling will definitely explore the right Gaussian, because as we previously mentioned MCMCs tend to *"climb up"* probability distributions. However, it is not probable that the MCMCs will manage to leave the right Gaussian. In fact, from Equation 11, we see that MCMCs are not very likely to *"climb down"* high probability distributions. This intuitive idea is supported by numerical simulations, as we can see in Figure 4 on the right, where the proposal distribution $\mathcal{N}(10, 1)$ was used to randomly pick the starting point. Then 1000 MCMC steps were performed (for 1000 chains) and only the endpoint of each chain was taken as sample. As we see, in this case, the Burn-in of 999 samples is not sufficient to grant a correct sampling of π [3]. This is clearly due to the poor choice of the initialization proposal distribution $\mathcal{N}(10, 1)$. If we instead draw the starting point from the proposal distribution $\mathcal{N}(0, 1)$, we do not expect the previously discussed mechanism to take place. We can see that this is the case in the left image of Figure 4. Thus we acknowledge that the a bad initialization can be the cause of issues in our sampling technique. We will now insert the problem we just observed in a quantitative framework, thanks to the definition of a new concept, the *Bias*.

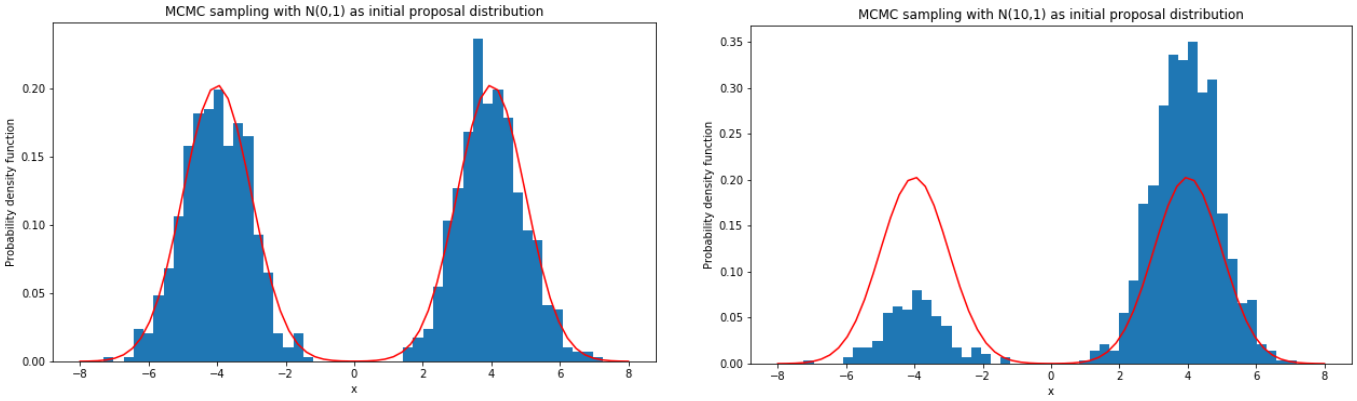

Figure 4: MCMC sampling drawn from the underlying red distribution, which is the sum of two Gaussian distributions centered in $\pm$ 4, both having unit variance. The image on the left depicts the histogram of 1000 samples obtained by driving 1000 MCMCs for 1000 time steps each. Every sample corresponds to the ending point of one of the 1000 time steps long MCMCs, driven according to the Metropolis Algorithm. Each MCMC's starting point was randomly proposed by the distribution $\mathcal{N}(0, 1)$. The image on the right is fairly similar to the left one, the only difference is that the initialization here is performed by randomly proposing the starting point according to the distribution $\mathcal{N}(10, 1)$.

2.3 The Bias

In the instance of this project we are interested in using MCMCs to draw estimates on the ensemble average values of observables. In fact, if we assume to be able to correctly sample values x_i from the probability distribution π, we will expect the following equation to hold:

$$\mathbb{E}_\pi(f) = \int f(x)\pi(x)dx = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N} f(x_i) \tag{13}$$

Where each of the x_i points is randomly drawn from the probability distribution $\pi(x)$. Thus we take advantage of the fact that we are able to randomly sample from $\pi(x)$, via MCMCs, to compute averages. But here is the problem, as we have previously mentioned we are able to obtain random samples from $\pi(x)$ only in the asymptotic limit, i.e., if we run infinitely long Markov chains. We see this in Figure 4, where

[3] The problem hereby highlighted could be easily avoided by driving a 10^6 long chain instead of 10^3 chains, each of them being 10^3 steps long. This is not done for explanatory reasons, i.e., to highlight the problem of bad initialization.

(in the right image) a bad estimate of $\pi(x)$ due to short chains and bad initialization is depicted. As a consequence then, Equation 13 should be rewritten, more properly, as:

$$\mathbb{E}_\pi(f) = \int f(x)\pi(x)dx \simeq \lim_{N\to\infty} \frac{1}{N}\sum_{i=0}^{N} f(x_i) := \hat{\mathbb{E}}_\pi(f) \tag{14}$$

Where the $\hat{\mathbb{E}}_\pi(f)$ symbol stands for the *estimated* average value of the observable f. If the MCMCs do not sample from the $\pi(X)$ distribution properly (which is, at least theoretically, the case if our chains are finite) we might end up with a $\hat{\mathbb{E}}_\pi(f)$ value that is significantly different from the true expectation value $\mathbb{E}_\pi(f)$. The difference between these two quantities is referred as the ***Bias***.

$$Bias(f) = \mathbb{E}_\pi(f) - \hat{\mathbb{E}}_\pi(f) \tag{15}$$

Since our aim is that of computing reliable averages numerically, our efforts must concentrate on diminishing the bias as much as we can, this will be the focus of the next chapter.

3 Unbiased MCMC sampling

3.1 Mainstream solutions to the bias problem

There are different ways to reduce the bias. In particular we can use a longer burn-in period, or we can take longer chains. This is shown in Figure 5, in which the setup is the same as in Figure 4 on the right, except for the fact that the 1000 MCMCs here are driven for 10000 time steps instead of 1000 (9999 samples were "burned-in"). The elongation of chains comes at a price though, and the price is run time.

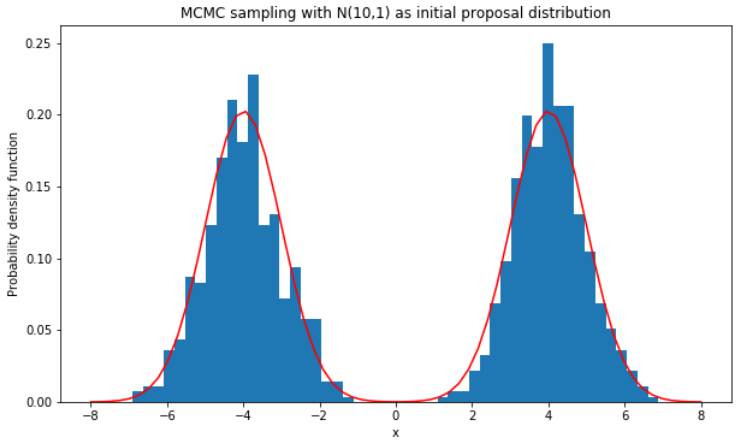

Figure 5: MCMC samplings drawn from the underlying red distribution. The image depicts the histogram of 1000 samples obtained by driving 1000 MCMCs for 10000 time steps each. Every sample corresponds to the ending point of one of the 10000 time steps long MCMCs. Each MCMC's starting point was randomly proposed by the distribution $\mathcal{N}(10, 1)$.

A combination of burn in and long chains enables us to have a significant reduction of the bias. However this method is not bulletproof and presents two weaknesses, when compared with the alternative method that we will present in this chapter. The weaknesses are a theoretical and a practical one:

- Theoretical: Estimates performed via the "classical" MCMC sampling (the one we exposed up to now) are only justified in the limit of **both** the number of samples and the length of chains, going to infinity. We will show that with the alternative method the **only** limit we will need is the one on the number of samples.

- Practical: If computer power will grow through the increase of the number of processors, it will be useful to have a method to compute estimates such as the one in Equation 14 using **parallel computation**, to build independently many short chains instead of (relatively) few longer ones [7]. The problem with short chains, on the other hand, is that they are strongly biased, as we have shown. The alternative method that we will discuss hereafter, however, enables us to perform unbiased estimates using such short chain parallel computations.

3.2 The idea of the alternative method

As we have previously mentioned, for infinitely long MCMCs we can consider the sampling to be randomly drawn from the underlying probability distribution $\pi(x)$, formally:

$$\mathbb{E}_\pi(f) = \int f(x)\pi(x)dx = \lim_{N,t\to\infty} \frac{1}{N}\sum_{i=0}^{N} f(X(t)) = \lim_{t\to\infty} \mathbb{E}(f(X(t))) \tag{16}$$

We can rewrite this equation, via a telescopic sum, in the following way:

$$\mathbb{E}_\pi(f) = \mathbb{E}(f(X(k))) + \sum_{t=k+1}^{\infty} \big[\mathbb{E}(f(X(t))) - \mathbb{E}(f(X(t-1)))\big] \tag{17}$$

Where k is an *arbitrarily chosen*, finite, natural number. Now, we know that the quantities $\mathbb{E}(f(X(t-1)))$ and $\mathbb{E}(f(Y(t-1)))$ must be equal, if X and Y are two MCMCs sharing the same proposal distributions. Thus Equation 17 can be reformulated as:

$$\mathbb{E}_\pi(f) = \mathbb{E}\bigg(f(X(k)) + \sum_{t=k}^{\infty}(f(X(t)) - f(X(t-1)))\bigg) = \mathbb{E}\bigg(f(X(k)) + \sum_{t=k}^{\infty}(f(X(t)) - f(Y(t-1)))\bigg) \tag{18}$$

We used the linearity of the ensemble average to write the first equality. Now, this does not seem like a very useful observation, since to compute the right hand side of the equation we still need our MCMCs to run for infinitely many time steps, and this is of course not feasible in practice. However, what if we suppose that we can build two chains X and Y that will meet after a finite amount of time steps , and stay together there after? If this was possible and the chains met at the time $\tau - 1$ we would be left with the following equation, for the other contributions would cancel out:

$$\mathbb{E}_\pi(f) = \mathbb{E}\bigg(f(X(k)) + \sum_{t=k}^{\tau-1}(f(X(t)) - f(Y(t-1)))\bigg) \tag{19}$$

In practice this means that we should run our chains X and Y until they meet, and then compute the observable in the ensemble average. By doing this infinitely many times we get an unbiased estimate of the ground truth value $\mathbb{E}_\pi(f)$. Notice that, as previously mentioned, here we only need one limit to be taken to infinity, namely the one over the number of samples. The limit used to ensure the infinite length of the MCMCs is no longer needed.

Now the problem is clear, if X and Y move on a continuous space their meeting probability seems to necessarily be zero. This is is actually not true however, since we can use a technique called Maximal Coupling to enable a non vanishing meeting probability, while leaving the marginal probabilities of X and Y unchanged. We will also show that, when two Maximal Coupling driven chains meet, they are bound to stay together forever. Maximal Coupling will be the focus of the next subsection.

3.3 Maximal Coupling

We are interested in sampling two variables x and y from two different probability density functions, respectively $p(x)$ and $q(y)$. The idea of Maximal Coupling is to sample x and y jointly from a two variables distribution $\pi(x, y)$ that satisfies the following conditions:

- $\int \pi(x,y)dy = p(x)$
- $\int \pi(x,y)dx = q(y)$
- $\mathbb{P}(x = y)$ is maximal

Where $\mathbb{P}(x = y)$ is the probability of sampling any couple x and y , such that $(x = y)$, from the joint probability distribution $\pi(x, y)$. It is possible to implement the following algorithm to sample from $p(x)$ and $q(y)$ with maximal coupling[1]:

- Sample x from p and extract U uniformly from the interval $[0, p(x)]$. If $U < q(x)$, output (x, x)
- Else sample y from q and extract U^* uniformly from $[0, q(y)]$, until $U^* > p(y)$, then output (x, y)

We tested this algorithm to perform Maximal Coupling sampling from the distributions $\mathcal{N}(0.5, 0.8^2)$ and $\mathcal{N}(-0.5, 0.2^2)$, the result is shown in Figure 6. The obtained two variables distribution matches our expectations, as its marginals coincide with the one variable distributions $\mathcal{N}(0.5, 0.8^2)$ and $\mathcal{N}(-0.5, 0.2^2)$, the marginals are shown in Figure 7. The idea then is to draw the two Markov Chains X and Y using Maximal Coupling, thus ensuring the finiteness of their meeting time τ and hence guaranteeing that Equation 19 holds. In fact the Maximal Coupling sampling, when applied to MCMCs, ensures that if two chains meet they will stay together thereafter. The reason being that the role played by p and q, in the MCMC context, is played by the proposal distributions for the Markov steps, i.e., $T(X(t_i) \to X(t_{i+1}))$ and , $T(Y(t_i) \to Y(t_{i+1}))$. Now, if the chains are to meet at a given time τ, i.e., if $X(\tau) = Y(\tau)$, we will have that $T(X(\tau) \to Z)$=$T(Y(\tau) \to Z)$ for every Z [4]. This means that the p and q involved in the maximal coupling, namely $T(X(\tau) \to Z)$ and $T(Y(\tau) \to Z)$, are exactly the same function of Z. This implies that the first case in the description of the algorithm always holds. As a consequence the output of the sampling will be of the kind $(X(\tau + i) = Z, Y(\tau + i) = Z)$ after the meeting of the two chains, for every $i \geq 1$.

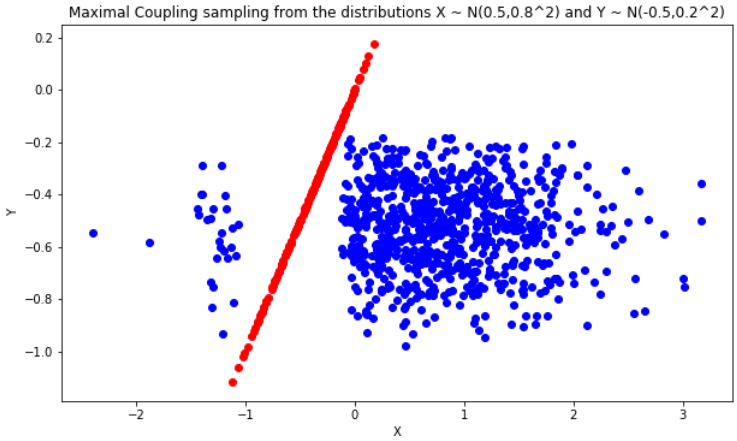

Figure 6: Maximal Coupling sampling from the distributions $\mathcal{N}(0.5, 0.8^2)$ and $\mathcal{N}(-0.5, 0.2^2)$ (1000 samples).

3.4 Implementation

Now we would be ready to implement the unbiased sampling technique we just described, and it would work in the thermodynamic limit. However we would lose some important pieces of information. In fact, consider that we draw two MCMCs, X and Y, that meet at a certain time step τ. In such a case, what value of k should we pick in the right hand side of Equation 19? The answer is that there is no privileged choice of k that we would prefer to make. To every choice of $k \leq \tau$ corresponds a valuable element that we can consider in our computation of the ensemble average. For this reason we define the following, more efficient,

[4]This is true since $T(X(\tau) \to Z)$ is, most of the times, completely specified when $(X(\tau))$ is given. For example $T((X(\tau) \to Z)$ can be a Gaussian distribution of given variance centered in $X(\tau)$

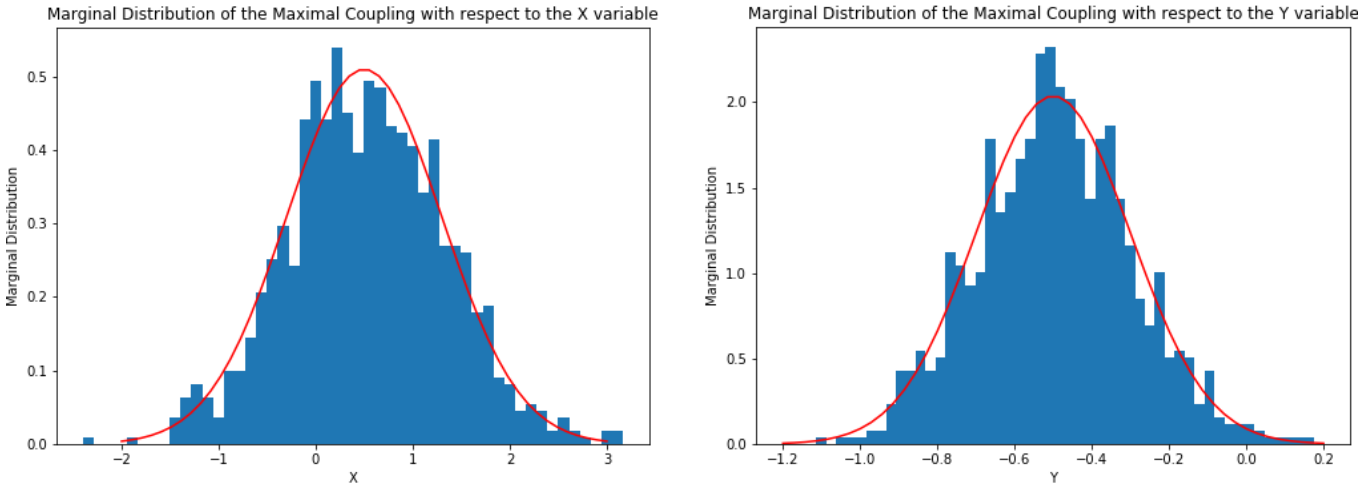

Figure 7: Marginals of the Maximal coupling distribution drawn from $\mathcal{N}(0.5, 0.8^2)$ and $\mathcal{N}(-0.5, 0.2^2)$. As it is possible to see there is agreement between the histograms and the underlying red distributions $\mathcal{N}(0.5, 0.8^2)$ and $\mathcal{N}(-0.5, 0.2^2)$.

estimator[5], which averages over estimates obtained by setting different values of k in the r.h.s of Equation 19:

$$H_{k:m}(X,Y) = \frac{1}{(m-k+1)} \sum_{l=k}^{m} H_l(X,Y) \tag{20}$$

Where $H_l(X,Y) = \left[f(X(l)) + \sum_{t=l}^{\tau-1}(f(X(t)) - f(Y(t-1)))\right]$. We must emphasise the fact that now, instead of performing the ensemble average on many $H_l(X,Y)$, as we prescribed in equation 19, we will perform the ensemble average on many $H_{k:m}(X,Y)$. This ensemble average will return an unbiased estimate of $\mathbb{E}_\pi(f)$. Now we are ready to put all the pieces together, for the sake of clarity, by giving an algorithmic and concise description [2] of the unbiased sampling technique we just described.

- Draw X_0 and Y_0, from the same (arbitrarily chosen) distribution π_0. Propose X_1 according to $T(X_0 \to \cdot)$ and, if it is accepted, update the X value to X_1.

- Set $t = 1$. If $t < \max(m, \tau)$, proceed to sample new proposals form the Maximal Coupling of $T(X_t \to \cdot)$ and $T(Y_t \to \cdot)$. In case the proposals are accepted update the values of X and Y, $t \to t+1$

We implemented this algorithm, for Metropolis MCMCs, and tested its behaviour when dealing with mono-dimensional and bi-dimensional probability density functions.

3.5 Testing and the important role of outliers

To test the unbiased algorithm we sampled from the distribution that we considered also in previous examples, i.e., the sum of two Gaussian distributions depicted in red in Figures 4 and 5. The test we implemented consists in seeing if, and how fast, we are able to reproduce the shape of the underlying distribution using the unbiased sampling technique. We did this via the evaluations of indicator functions, which we substituted to f in Equation 20. Moreover, since the aim of the method is to deal with the bias caused by bad initialization, we intentionally initialized the chains poorly, i.e., with the initial proposal distribution $\mathcal{N}(10, 1)$. As a first attempt we performed the ensemble average on $n = 1000$ (estimates of $H_{k:m}(X,Y)$ obtained from) couples of chains (Figure 8 on the left) and on $n = 10000$ (estimates of $H_{k:m}(X,Y)$ obtained from) couples of chains (Figure 8 on the right). Now, Figure 8 (on the left) is fairly similar to Figure 4 (on the right). Then we might feel some sense of confusion. In fact, the bias, stemming from bad initialization, seems to persist even if the estimator that we are considering is unbiased. We must not lose hope though, since we recall that the estimator we are working with (Equation 20) is only unbiased

[5]Notice that k in Equation 20 has a different meaning than the k in Equation 19

in the thermodynamic limit, i.e., if $n \to \infty$. Thus we expect the Thermodynamic limit to save us. However, one might raise the following objection:

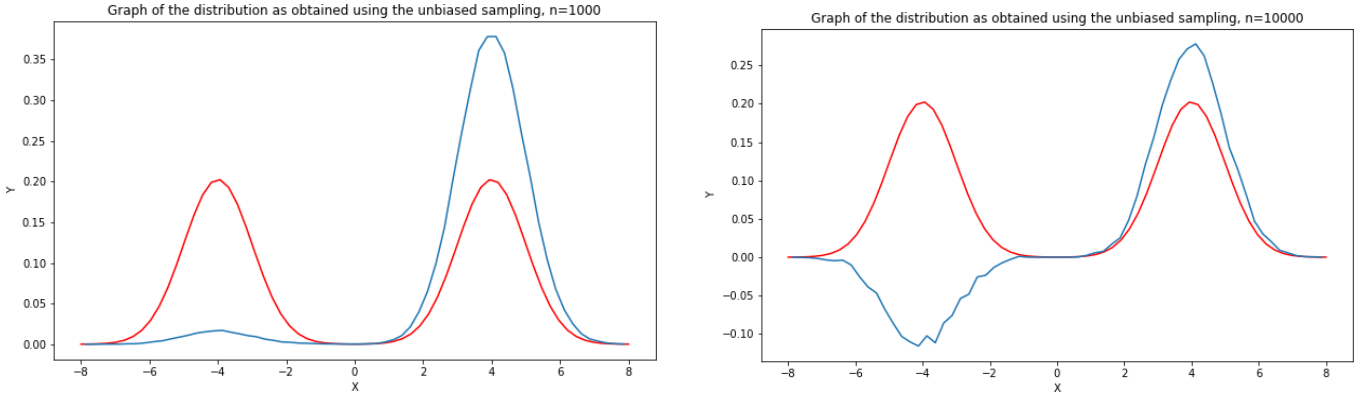

Figure 8: Comparison between the underlying probability density function (in red) and the *experimental* probability density functions (in blue). The blue distributions were obtained via the evaluation of 64 indicator functions, based on 64 intervals equally dividing the $[-8, 8]$ interval. The image on the left was obtained by performing the Ensemble average over $n = 1000$ couples of chains, for the one on the right, instead, $n = 10000$ couples of chains were considered.

"There is no way for the Thermodynamic limit to save the situation. In fact, if every ensemble average performed on $n = 1000$ couples of chains results in a graph similar to the blue one in Figure 8 on the left, there is no hope to recover the red distribution. In fact if we draw, for example, 10 different ensemble averages performed on 10 different sets constituted by $n = 1000$ couples of chains, we will obtain 10 graphs that are very similar the left one in Figure 8. Thus we can average over this 10 graphs to obtain a graph that is resulting from the ensemble average over $n = 10 * 1000 = 10000$ chains. This new graph, however, will be again of the form shown on the left in Figure 8. This is because the average of 10 distributions, that are very similar among each other, will again be similar to each and every one of the said 10 distributions. We are stuck as neither the Thermodynamic limit can save us!"

This argument is correct, but it relies on a false assumption, namely: "If we draw 10 different ensemble averages performed on 10 different sets constituted by $n = 1000$ couples of chains, we will obtain 10 graphs that are very similar to the left one in Figure 8". In fact, it is true that around half of the times that we take the average over $n = 1000$ couples of chains the graph obtained via the unbiased sampling resembles the left one in Figure 8. However the other half of the times the obtained graph has a more "random" shape, as shown in Figure 9.

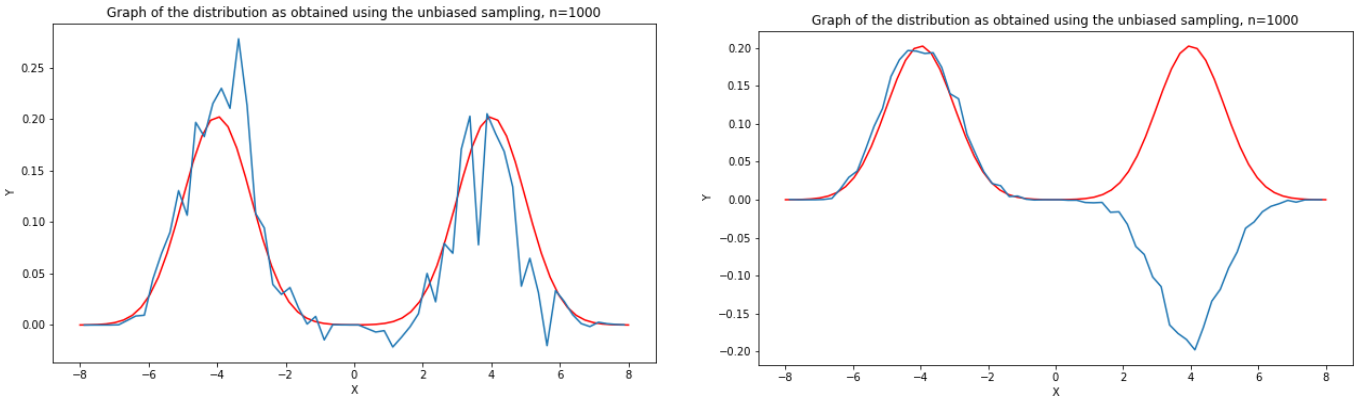

Figure 9: Both these images were obtained as the one on the left in Figure 8. However both the samplings resulting in the above images were characterized by the presence of *outliers*. Left image, outlier 1004 steps long. Right image, outlier 6762 steps long.

When looking at the graphs shown in Figure 9 two questions might arise, namely:

1. If the blue curve is meant to be a probability density function, how can it be negative at some points?
2. What is the mechanism underlying the production of such strange graphs?

These two questions can be answered together, by looking at Equation 19 [6]. By running many simulations we discovered that, on average, a couple of chains meets after about 8 time steps. However, it might happen, about once in every 1000 couples of chains, that two chains take an exceptionally long time to meet, i.e., more than 500 time steps. This happens, for example, if the X chain moves to the Gaussian curve centered in -4 while the Y chain stays (some more time) on the Gaussian curve centered in $+4$ before meeting with X. If this happens, in Equation 19 the $f(Y(t-1))$ term will continue to assume positive values if f is an indicator function in the neighbourhood of $+4$, while $f(X(t))$ will be equal to zero in the said neighbourhood (since X has moved to the other Gaussian). Thus the $[f(X(t)) - f(Y(t-1))]$ term will give a $[0 - positive] = negative$ contribution. This results in the assumption of negative values, by the experimental probability density (blue in the figures), in the neighbourhood of $+4$. The same argument, adapted to indicator functions in the neighbourhood of -4, explains why we have higher (positive) values of the blue function around -4. This is clearly the case with Figure 9 on the right, and a little bit less clearly the case with Figure 9 on the left [7]. We call this rare events, in which the couples take a long time to meet, *outliers*. In Figure 9 on the left there was one outlier case amongst the $n = 1000$ couples of chains considered. In the said outlier case the meeting of X and Y happened after 1004 time steps. As for the right image in Figure 9 there is again only one outlier, however it is a "longer" one, as X and Y met only after 6762 time steps. Although being rare events, the outlier cases represent a crucial element of the algorithm, without which the convergence to the underlying red probability distribution would not be attained. Thanks to the outlier concept we just introduced we can now finally account for the difference between Figure 8 on the right and Figure 8 on the left, as a consequence of the respective absence (Figure 8 on the left) and presence (Figure 8 on the right) of outliers. Now that we understood the *rare event based* mechanism underlying the unbiased sampling, we can test the convergence of the method, i.e., we can see how the shape of the blue curve changes as n grows. This is shown in Figure 10.

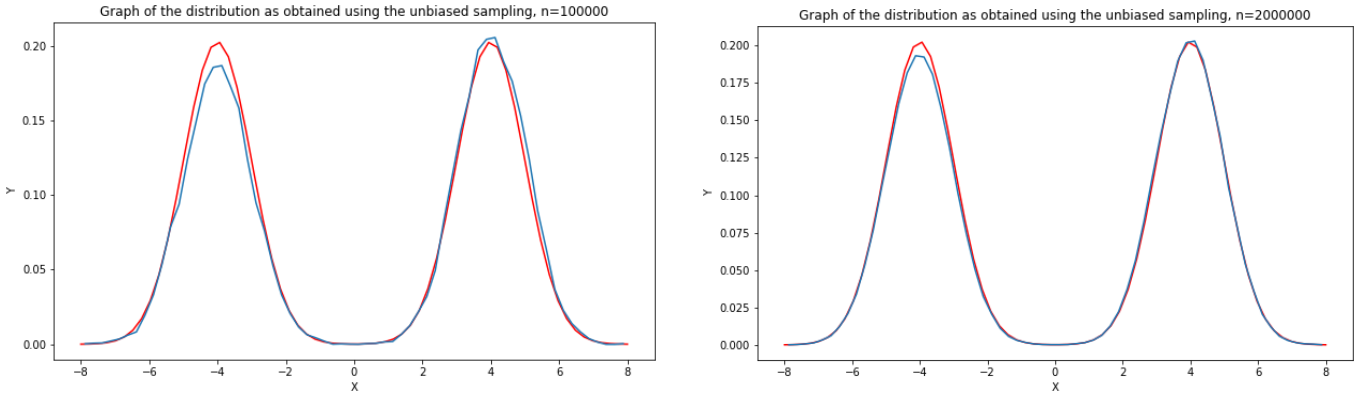

Figure 10: Images resulting from unbiased ensemble averages of indicator functions' values. The image on the left was obtained by sampling $n = 10^5$ couples of chains, while the image on the right was obtained by sampling $n = 2 * 10^6$ couples of chains. It is possible to see that the thermodynamic limit actually removes the bias.

[6] This is not the estimator that we are actually using. However our estimator comes from an average of estimators of the kind appearing in Equation 19, thus the argument that we are going to use still holds. The choice to refer to Equation 19 and not to Equation 20 has been made for the sake of clarity.

[7] Although this image is strongly different from Figure 9 we can make sense of it by applying it the same argument we used to explain the right image in Figure 9. The qualitative explanation of why it looks different from Figure 9 on the right, is that the contribution coming from the outlier is averaged out by the contributions coming from the other 999 couples of non-outlier chains, this was not the case in Figure 9 on the right. This is due to a proportionality intercurring between the length of outlier and the weight it plays in computation of the curve. In fact the outlier for the image on left of Figure 9 is shorter, 1004 steps, then the outlier for image on right, 6762 steps. As a consequence it is also less impactful.

By comparing the results in Figure 10 with those in Figure 9 we see that convergence is in fact attained, thanks to the role of outliers. In the following chapter we are going to sample both from a one variable distribution and from a two variables one. As a consequence we tested the behaviour of the algorithm when dealing with two variables probability density function. In particular we worked with a trivial 2-d extension of the red distribution that we used to test the 1-d case, i.e., we considered the sum of two unit variance (two variables) Gaussian distributions centered in $(\pm 4, 0)$. Once more we intentionally initialized the MCMCs poorly, with a 2-d Gaussian proposal distribution $\mathcal{N}(\mu, \sigma)$, with $\mu = (10, 0)$ and $\sigma = \mathbb{I}$, to check if the unbiased sampling technique is in fact able to reduce the bias caused by bad initialization. The result of the sampling from the distribution we just described is shown in Figure 11.

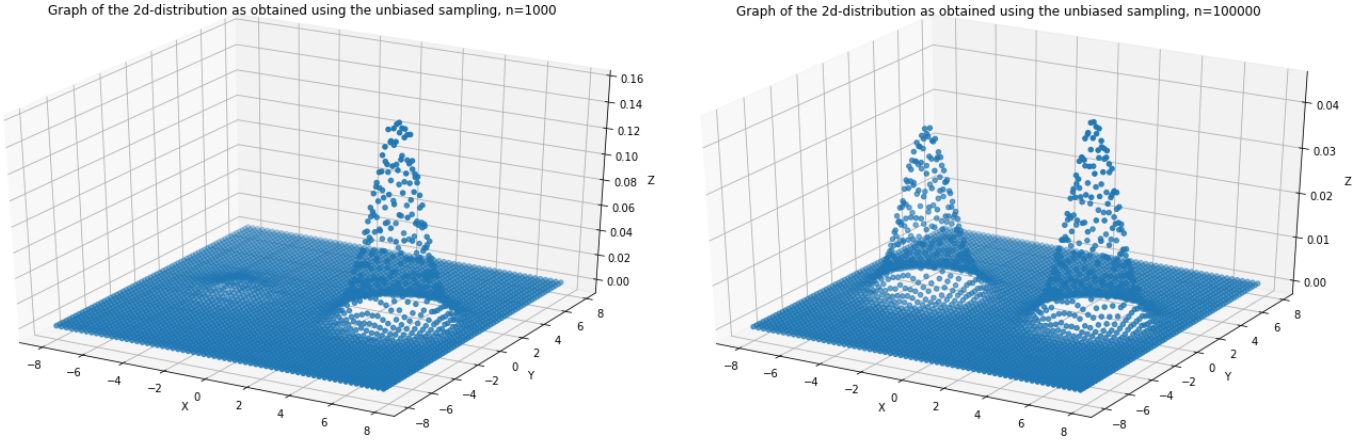

Figure 11: Images resulting from the sampling of a 2 dimensional linear combination of Gaussian distributions, namely being the sum (with equal weights) of two Gaussians centered in $(\pm 4, 0)$, each of them having unit variance. The image on the left was obtained by averaging over an ensemble of $n = 1000$ couples of chains, while, for the image on the right, an ensemble of $n = 100000$ chains was considered.

The behaviour of the unbiased sampling algorithm in a two dimensional context is fairly similar to its behaviour in a one dimensional framework. In fact, the convergence to the underlying probability distribution is still strongly relying on the presence of outliers, as it is visible in Figure 11. It is important to notice that the meeting times for the MCMCs are longer, in the 2 dimensional case, then those we measured in the 1 dimensional sampling. The reason being that the probability of having a meeting for two chains X and Y is proportional to the volume under the intersection of the proposal distributions for the updates of X and Y [1]. This volume gets smaller and smaller as the number of dimensions increase, and as a consequence the meeting times grow, resulting in an increase in run time.

4 A first, mono-dimensional, test on real data

We successfully applied the unbiased sampling technique to draw estimates of population sizes. In particular we adapted the Unbiased sampling algorithm so that it could interact with BEAST2, a program for Bayesian phylogenetic analysis of molecular sequences. BEAST2 uses MCMC sampling to draw estimates on quantities of phylogenetic interest, such as phylogenetic trees, reproductive numbers or population sizes. Before applying the Unbiased sampling to the Hepatitis C data set we tested it on a test data set to recover the constant value of a population size. This has been done purely as a test, to see if the Unbiased Sampling technique was correctly connected to the BEAST2 environment. In this analysis we considered the Phylogenetic tree to be fixed to a ground truth "value" obtained via standard MCMC sampling. The distribution in Figure 12 on the right represents the ground truth probability density function of the population size and is peaked in 1. Notice that the this does not mean that the population size is actually most likely to be 1. In fact the probability density function needs to be rescaled with the parameter g before we can give it a meaningful interpretation. The mean value of the ground truth distribution is 1.705. In Figure 12 on the left is instead shown the result obtained with the unbiased sampling algorithm. It is consistent with the ground truth (mainstream MCMC) result shown at its right. In fact, the peak of the unbiased graph stays at $x = 0.75$, and is thus comparable with the ground truth value of $x = 1$. Further, the unbiased average

value of x is 1.66 and it is also compatible with the ground truth value of 1.705.

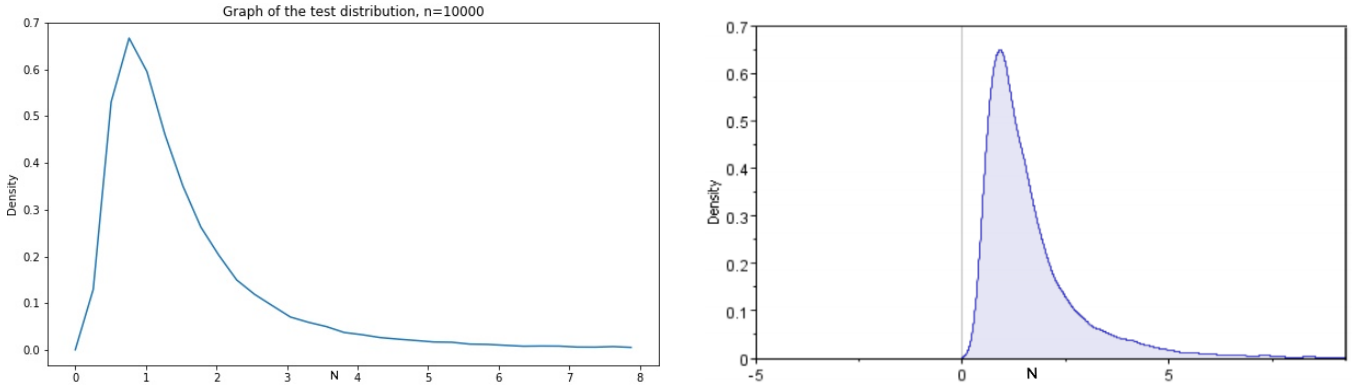

Figure 12: Images resulting from the sampling of a one variable probability distribution. The image on the left was obtained by using the unbiased sampling technique, averaging over an ensemble of $n = 100000$. The image on the right, instead, was obtained by standard MCMC sampling, and here is considered as the ground truth distribution. The two distributions are in agreement, as they share the peak value, around 1, and the mean value, which is 1.66 for the image on the left and 1.705 for the image on the right.

5 Applying unbiased sampling to Hepatitis C data

We performed our analysis on Hepatitis C data collected in Egypt between 1950 and 1993[4]. The choice of this data set was based on two reasons. First, as we discussed at the end of subsection 3.5, the unbiased sampling technique is negatively affected by high dimensional probability distributions. As a consequence we did not intend to use (at least in the instance of the current analysis) the unbiased MCMC sampling technique to sample the phylogenetic tree of an epidemic, which inhabits a high dimensional space. For this reason we wanted to work with a data set which had a predetermined phylogenetic tree (obtained through mainstream MCMC sampling with a high confidence level). Secondly, the aim of the analysis is to be a feasibility study. As a consequence we wanted to be able to work on a data set that has *"ground truth"* values. This is the case with the Hepatitis C data set, as the estimates for the population sizes drawn from it by using mainstream MCMCs have a very high confidence level, and can be thus thought of as ground truth values. To proceed with the unbiased sampling, we approximated the number of infected individuals (population size) as a piece wise constant function of time over two time intervals, as suggested by the result of the standard MCMC analysis (see Figure 3). The unknowns of the problem are then represented by the constant values, N_1 and N_2, that the piece wise constant function assumes over these two intervals. Thus, our guess about the said two values results in a probability density function depending on two variables. The shape for the said two variables distribution is reported in Figure 13, its marginals are instead shown in Figures 14 and 15. The comparison between left and right images in Figure 13 can only be qualitative, as the ground-truth distribution we are working with is actually a scatter-plot, obtained sampling from the probability density function, while the result given by the unbiased algorithm is the probability density function itself. Thus the different nature of the graphs makes a comparison hard to perform. However it is still possible to observe a qualitative agreement between the two plots. To make a more quantitative comparison we can look at the marginals of the distributions, both unbiased and ground-truth, which are shown in Figures 14 and 15. Let us start by comparing images in Figure 14. These show the marginals as obtained by integrating over N_2. The unbiased graph on the left is peaked around the value of $N_1 = 10000$, in accordance with the ground truth graph. The peak in the unbiased graph, however, is lower then its ground-truth equivalent, as a consequence of the presence of a large right tail in the unbiased distribution. The presence of the said tail affects the mean value, that is 13314 for the unbiased distribution, compared to the ground truth mean of 11803. In Figure 15, instead, the marginal obtained integrating with respect to the N_1 variable is shown. The unbiased and the ground-truth distributions have again compatible values for the peaks' positions, 625 in the unbiased case and 613 in the ground truth one. The averages of the two distributions show accordance as well. In particular the mean stands at 645 for the unbiased distribution,

this has to be compared with an average value of 653, attained by the ground truth curve.

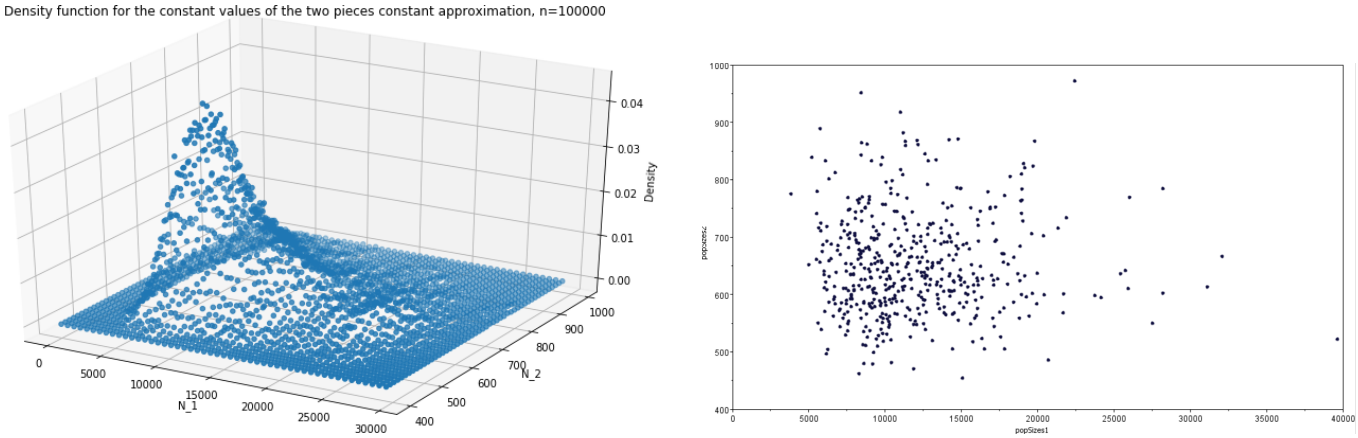

Figure 13: Images resulting from the sampling of a two variables probability distribution. The two variables, N_1 and N_2 represent the constant values of the infected populations over two different intervals of time. The left image graph was obtained by evaluating a grid of indicator functions (covering the whole domain of the function) via the unbiased sampling algorithm. The image on the right was instead obtained as a scatter-plot of standard MCMC samples. The two images are coherent nonetheless, as high/low densities of points in the scatter-plot correspond to high/low values of the distribution in the left image.

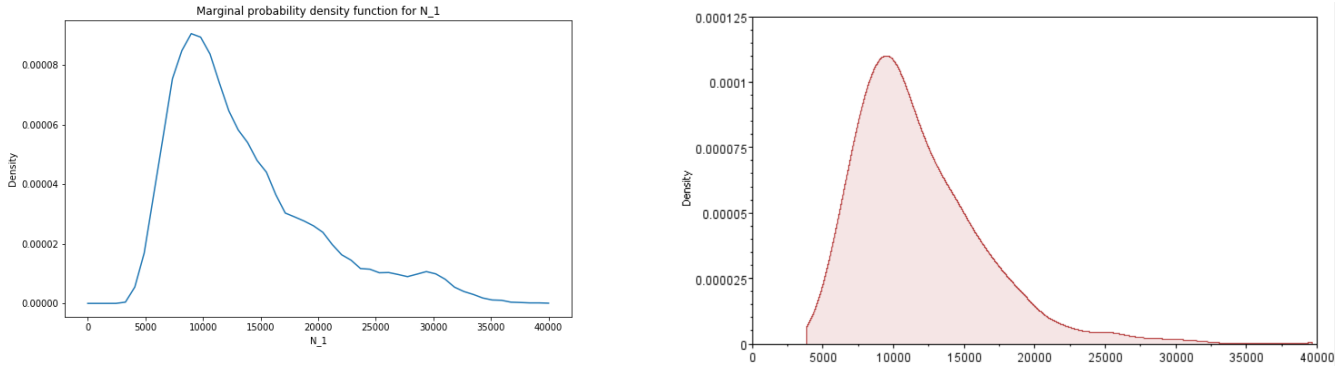

Figure 14: Comparison between the unbiased (left image) and ground truth (right image) N_1 marginals.

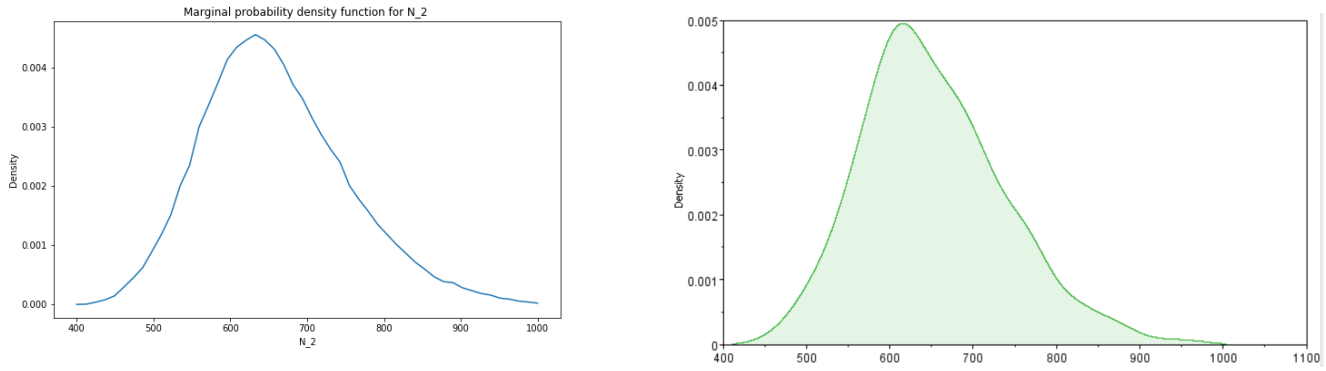

Figure 15: Comparison between the unbiased (left image) and ground truth (right image) N_2 marginals.

6 Conclusion

Our aim, with this paper, was to get a deeper understanding of the unbiased algorithm proposed by Pierre Jacob et al [1]. In particular we found that a very interesting, and run-time consuming, feature of the algorithm is its outlier-based nature. In fact, when working with initialization sensible distributions (see for example Figure 8) the algorithm needs to run many times, to increase the number of outlier occurrences, i.e., the number of long chains. These outliers are needed to balance the more mainstream contributions given by the short chains. From this observation we might draw a guideline for a future upgrade of the algorithm. In particular, while running simulations on distributions that are easily affected by a bad initialization (see for example Figure 8) we observed that most of the computer power is exploited to produce a vast amount of short chains. However, these short chains, are not of particular interest as they mainly produce the same values for the expectations of observables. In practice we noticed that averaging over the values of observables obtained from $N = 10^3$ couples of short chains is not much different from averaging over values obtained from $N = 10^6$ short chains. It is to say that the algorithm spends most of its time producing short chains that are not very useful. As a consequence, we might be interested in thinking about ways to avoid this problem. A (maybe naive) suggestion could be to estimate the mainstream (i.e. short) chains to outlier (i.e. long) chains ratio with some runs of the algorithm in its standard form. Once this ratio has been computed, we might find a way to force many outlier computations. Then we could use the ratio previously computed to compensate the outlier contributions with the correctly weighted mainstream contributions. In practice this could mean that instead of producing $N = 10^6$ short chains we produce $N = 10^3$ of them and we give more weight to their contribution. This procedure would probably bargain some theoretical guarantees for shorter run times. However, it would maintain the important suitability of the algorithm for parallel computations. It is in fact worth mentioning once more that a crucial strength point of the unbiased algorithm we studied, stems from the fact that it enables parallel computations. This is an interesting feature that will become valuable if future computer technology will allow for more powerful parallel computations, as it is forecast by Colin Carrol [7]. It is in view of this latter consideration that we decided to perform a feasibility study of the algorithm's application to MCMC based Phylogenetic Analysis, finding agreement with results produced via standard MCMC sampling.

References

[1] Pierre Jacob. *Sampling from a maximal coupling.* URL: https://statisfaction.wordpress.com/2017/09/06/sampling-from-a-maximal-coupling/.

[2] Pierre E. Jacob, John O'Leary, and Yves F. Atchadé. *Unbiased Markov chain Monte Carlo with couplings.* 2017. arXiv: 1708.03625 [stat.ME].

[3] Timothy Vaughan et al. Tanja Stadler Carsten Magnus. *Statistical and Computational Analysis of Genetic Sequence Data.* URL: https://polybox.ethz.ch/index.php/s/xRAWLpqa41Owbdv.

[4] Alexei J Drummond et al. (2005). *"Bayesian coalescent inference of past population dynamics from molecular sequences". In: Molecular biology and evolution 22.5 (2005), pp. 1185–1192.*

[5] H.J. Herrmann L. Böttcher. *Computational Statistical Physics, preprint.* URL: https://ethz.ch/content/dam/ethz/special-interest/phys/theoretical-physics/itp-dam/documents/compstatphys/CompStatPhysPartI_notes.pdf.

[6] Johannes M. Hohendor (2005). *An Introduction to Markov Chain Monte Carlo.* URL: http://probability.ca/jeff/ftpdir/johannes.pdf.

[7] Colin Carrol. *(Biased) MCMC.* URL: https://colcarroll.github.io/couplings/static/biased_mcmc.html.