Application of Markov Chains and MC chains*

Inbreeding Coefficient estimator

Wenxuan Li

April 27, 2022

Abstract

The Hardy-Weinberg Equilibrium is critical in human genetics approaches. It states that allele and genotype frequencies remain constant throughout generations and that there should be a straightforward link between the two types of frequencies in a large random-mating population in the absence of selection, mutation, and migration. When the selection assumption is broken, the inbreeding model is utilized to determine if the HW equilibrium has been disturbed. Numerous approaches for calculating the inbreeding coefficient have been proposed, but none have proven state-of-the-art performance due to the challenges associated with solving complicated integrals. As a result, we offer an upgrade to the present techniques for estimating inbreeding coefficients in this study by incorporating the Markov Chain Monte Carlo as an integration approximator. The experiment results indicate that for both high-dimensional and low-dimensional data, the Metropolis-Hastings-within-Gibbs strategy outperformed all other MCMC approaches.

Contents

*Code and data are available at: https://github.com/leoli2022/sta304

1 Introduction

All works are contributed by Karen L Ayres & David J Balding's work. In the article "Measuring departures from Hardy–Weinberg: a Markov chain Monte Carlo method for estimating the inbreeding coefficient". By using R (R Core team, 2020), tidymodels (Kuhn and Wickham, 2020), kableExtra (Zhu, 2021) packages, we are able to clean and analysysed the data from .

The Hardy-Weinberg(HW) Equilibrium asserts that in a large random-mating population, assuming no selection, mutation, or migration, allele and genotype frequencies remain constant from generation to generation, and there is a straightforward relationship between genotype and allele frequencies (Hardy HG, 1908). This is essential because several techniques in human genetics are predicated on the existence of Hardy-Weinberg Equilibrium. Specifically, the frequencies of the two alleles (A or B) at a bi-allelic marker are p and q, where p = 1.

However, the assumptions are often broken in actuality, and to evaluate the deviation from Hardy Weinberg Equilibrium, we may simply compute the anticipated genotype frequencies and compare them to the observed ones using the chi-squared test. Conversely, a variety of approaches have been developed for estimating f, a parameter that quantifies the deviation from HW induced by inbreeding. And Ayres and Balding explain why such strategies are inadequate (Ayres, 1988).

If inbreeding (i.e. selection) is the main violation of HW assumptions, causing variation from HW, the inbreeding model may be appropriate(Malecot, G., 1969), where p_{ij}, the relative frequency of the geno-type $AiAj$ is:

$$p_{ii} = p_i(f + (1 - f)p_i) \tag{1}$$

$$p_{ij} = 2p_i p_j(1 - f) \tag{1}$$

where pi is the frequency of allele A_i, and f is the inbreeding coefficient.Equation one yields the HW proportions for f = 0. When f equals 1, heterozygotes never form. The value of f may be negative, but it is constrained below by the condition that the population frequencies of each homozygote be positive, resulting in:

$$f \geq \left(\frac{-p_{min}}{1 - p_{min}}\right) \tag{2}$$

where p_{min} is the smallest frequency (Ayres, 1988).

f may be read in certain models of population subdivision as the probability that an individual's two genes are identical by descent (Crow, J. F, 1970), in which case it is restricted to be non-negative. Nei Chesser(Nei, M., 1983) and Robertson Hill (Robertson, A., 1984) presented point estimators for the inbreeding coefficient, however, these estimators do not explicitly account for inbreeding and may produce values that contradict the limit (Ayres, K., 1998) in the multi-allelic situation.

Ayres Balding (Ayres, K., 1998) introduced the maximum likelihood estimator, which adheres to the inbreeding model's limit. Assuming a random sample of genotypes, the probability is as follows:

$$P(n_{ij}|f, p_1, \ldots, p_k) = C_1 \prod_{i=1}^{k} (p_i(f + (1 - f)p_i)^{n_{ii}} \prod_{j=i+1}^{k} (2p_i p_j(1 - f))^{n_{ij}} \tag{3}$$

where C1 is a constant. For k = 2, equation three is readily maximized (Weir, B. S., 1998) to obtain

$$\hat{f}_{mle} = 1 - \left(\frac{2n_{12}n}{(2n_{11} + n_{12})(n_{12} + 2n_{22})}\right) \tag{4}$$

For k > 2, it is not possible to maximise the likelihood analytically, but numerical approaches (Ayres, K., 1998) and the EM algorithm (Hill, W. G.., 1995) may be used. The likelihood function derived by making all other parameters equal to their MLE value (i.e. $p_i's$) offers a measure of the support provided by the data for various potential values of f, but it ignores uncertainty in the p_i (Ayres, K., 1998). While integration across the joint distribution of p_i may be used to determine the marginal probability of f, accurate integration may be impractical, and we can estimate the integration using Markov Chain Monte Carlo (MCMC) techniques.

2 Data

2.1 Data Models

2.1.1 Dataset 1: Biallelic Site

When a given locus in a genome has two reported alleles, this site is referred to be a biallelic site, with k equal to two in our research. If the inbreeding coefficient, f, in our observed sample is 0.05 and our sample contains 200 individuals with an allele frequency of $p_1 = 0.25$ and $p_2 = 0.75$, the genotype frequencies may be simulated using equation (1). Then, using our observed data, we calculate the phenotypic frequencies as $n_ij = ntimesp_ij$. However, since this simulation often produces non-integer phenotypic frequencies, we estimate them to get a more useful observation.

2.1.2 Dataset 2: Multiallelic Site

When a single locus in a genome has three or more observed alleles, this site is referred to be a multiallelic site, and in our research, k = 6 is taken into account specifically. If the inbreeding coefficient, f, in our observed sample is 0.05 and our sample contains 1000 individuals with an allele frequency of p_i = (0.02, 0.06, 0.075, 0.085, 0.21, 0.55) for I = 1, 2,. . . , 6, the genotype frequencies may be simulated using equation (1). As with k = 2, the phenotypic frequencies are calculated as $n_ij = ntimesp_ij$, which corresponds to our observed data. Additionally, since this simulation often produces non-integer phenotypic frequencies, we estimate them again to get a more realistic observation.

2.2 Methodology

2.2.1 Metropolis-Hastings-within-Gibbs

Full joint density is the following:

$$\pi(n_{ij}) = P(n_{ij}|f, p_1, \ldots, p_k) = C_1 \prod_{i=1}^{k} (p_i(f + (1-f)p_i))^{n_{ii}} \prod_{j=i+1}^{k} (2p_i p_j(1-f))^{n_{ij}} \tag{5}$$

Given that this is a product of a large number of integers between 0 and 1, using the log of this joint density simplifies calculation and avoids the issue of very tiny values:

$$log(\pi(n_{ij})) = log(P(n_{ij}|f, p_a, \ldots, p_k))$$

$$= \sum_{i=1}^{k} n_{ii} log(p_i(f + (1-f)p_i) + \sum_{i=1}^{k} \sum_{j=1+1}^{k} n_{ij} log(2p_i p_j)(1-f)) \tag{6}$$

The proposal function for p_i:

$$p_u^t \sim Unif[max(0, p_u^{t-1} - \epsilon_p), min(p_u^{t-1} + \epsilon_p, p_u^{t-1} + p_v^{t-1}]$$

$$p_v^t = p_u^{t-1} + p_v^{t-1} - p_u^t \tag{7}$$

The proposal function for f:

$$f \sim Unif[max(0, p_u^{t-1} - \epsilon_p), min(p_u^{t-1} + \epsilon_p, p_u^{t-1} + p_v^{t-1})] \tag{8}$$

where p_{min} is the minimum pi at step t, ϵ_f.

The main idea is to update a pair of p_u and p_v, setting the new proposed $p_u^t + p_v^t = previous\ p_u + p_v$ guarantees that $\sum_{i=1}^{k} p_i = 1$.

Then, using the Metropolis-Hastings rule, we accept or reject our proposed p_u and p_v, where the whole joint density function and proposed function are obtained from equations (5) and (7). Following that, we suggest f with distribution in (8). (Note that we may suggest a new f only after we accept the p.) Throughout the

process, we adjust $epsilon_p$ in order to control our acceptance rate; a positive value of $epsilon_p$ is chosen to achieve reasonable acceptance rates; if $epsilon_p$ is too large, the chain will stick too much in one place and thus converge very slowly; if $epsilon_p$ is too small, the chain will make frequent but very small moves and thus converge very slowly. Due to the complexity of our joint density, we deal with it by using logarithms, e.g. accept iff $log(U_n) < log(A_n)$, where $U_n \sim Unif[0,1]$ and $A_n = \left(\frac{g(f^{new},P^{new})q(f_{old},P_{old})}{g(f^{old},P^{old})q(f^{new},P^{new})}\right)$

2.2.2 Gibbs Sampler

Rather of adopting the standard Component-wise Metropolis-Hastings method, we attempted to suggest each coordinate based on its conditional distribution with respect to all other coordinates. The conditional distributions of f, p_1,..., p_k are constructed as follows from the complete joint distribution (5):

$$g(f|p_1,...,p_k,n_{ij}) = \prod_{i=1}^{k}[f+(1-f)p_i]^{n_{ii}} \prod_{j=i+1}^{k}(1-f)^{n_{ij}} \tag{9}$$

for $\left(\frac{-p_{min}}{(1-p_{min})}\right) \leq f \leq 1$

$$g(p1|f,p_2,...,p_k,n_{ij}) = p_1(f+(1-f)p_1)^{n_{11}}(p_1(1-f))^{n_{12}} \tag{10}$$

for $0 \leq p_1 \leq 1$

$$g(p_2|f,p2,nij) = p_2(f+(1-f)p_2)^{n_{22}}(p_2(1-f))^{n_{12}+n_{23}} \tag{11}$$

for $0 \leq p_2 \leq 1$

$$g(p_3|f,p2,nij) = p_3(f+(1-f)p_3)^{n_{33}}(p_3(1-f))^{n_{34}+n_{23}} \tag{12}$$

for $0 \leq p_3 \leq 1$

$$g(p_4|f,p2,nij) = p_4(f+(1-f)p_4)^{n_{44}}(p_4(1-f))^{n_{34}+n_{45}} \tag{13}$$

for $0 \leq p_4 \leq 1$

$$g(p_5|f,p2,nij) = p_5(f+(1-f)p_5)^{n_{55}}(p_5(1-f))^{n_{56}+n_{45}} \tag{14}$$

for $0 \leq p_5 \leq 1$

$$g(p_6|f,p2,nij) = p_6(f+(1-f)p_6)^{n_{66}}(p_6(1-f))^{n_{56}} \tag{15}$$

for $0 \leq p_6 \leq 1$

We suggest each p_i according to its conditional density using a systematic scan and normalize them as we did for the starting values to guarantee the sum equals 1. In this scenario, we always accept our proposal and then use the conditional distribution of f to update it.

2.2.3 Independence Sampler

As described in the Results section, the M-H algorithm performs pretty well. As a result, we decided to test the independence sampler, a specific instance of the M-H method, to see whether it may give us a more efficient solution. As previously indicated, the whole joint distribution is (5); the proposed distribution for the moving function f is as follows:

$$f \sim Unif(max(\frac{-p_{min}^t}{1-p_{min}^t}, f-\epsilon_f), min(f+\epsilon_f, 1)) \tag{16}$$

where p_{min} is the minimum p_i at step t, ϵ_f.

Due to the complexity of our joint density, which might be rather modest depending on the value, we deal with it using logarithms, e.g. accept if $log(U_n)<log(A_n), where $U_n = left(fracg(Y_n)q(X_n-1)(g(X_n-1)q(Y_n)right)$ and $A_n = left(fracg(Y_n)q(X_n-1)q As a result, the suggested states Y_n are distinct from their preceding states $X_n - 1$. In practice, we disregard the easy scenario when k = 2 and only discuss the case where k = 6 because MCMC is more likely to be needed due to the implausibility of numeric approaches.

2.2.4 Other Monte Carlo methods

Importance sampling appears to be impossible without detailed information about the sample group, as we would be unable to find the kernel of the distribution of allele frequencies and inbreeding coefficients (f,p_1,...,p_k), from which to sample. As a result, it is inefficient and makes no sense, as illustrated in figure 2. Additionally, the Rejection Sampler seems to be illogical, since it is quite difficult to find an appropriate K and f(x) to constrain our joint density function, even in the simplest situation when k=2.

3 Result

3.1 MLE on Dataset 1

When k equals 2, we may apply equation (4) to estimate the MLE. When simulating the data, we used the nearest integer of n_ij, e.g. 1 for 1.36, to obtain the first estimate; because this value is less than the exact value of n_ij, the estimate of f will be smaller/larger; we then obtained the second estimate festimate2 by using the next nearest integer of $n_ij + 1$, e.g. 2 for 1.36 + 1. By combining the first and second estimators, we may get a final MLE estimate with a smaller error margin than if we used simply one of them. When n = 200, the resulting estimate is around 0.05281472. Nota bene: if we do not round n_ij as an integer, we may get an accurate value of 0.05.

3.2 M-H Algorithm on Dataset 1

Table of acceptance rate by diverent values of ϵ_p.

Table 1: Acceptance rate by different epsilon p when k = 2

Epsilon	Acceptance Rate	Standard Error
0.01	0.7961	0.0046836
0.02	0.6367	0.0025845
0.03	0.4869	0.0025490
0.04	0.3691	0.0020880
0.05	0.2788	0.0020126
0.06	0.2202	0.0021710
0.07	0.1574	0.0022045
0.08	0.1285	0.0024934
0.09	0.1016	0.0028035
0.10	0.0922	0.0034737

According to the summary table 1, when $epsilon_p$ is between 0.02 and 0.05, the acceptance rates are between 0 and 1, with a relatively low standard error of 0.03 to 0.08. By setting $epsilon_p$ to 0.03, the algorithm was run for 10000 iterations with a 1000-iteration "burn-in" period, and we provide an estimate for f:

$$\hat{f} = \frac{1}{(M-B)} \sum_{i=B+1}^{M} f_i = 0.05230045$$

with a 95% confidence interval. This estimator's confidence interval is as follows: (0.04777199, 0.05682892). This confidence range encompasses our genuine theoretical value of 0.05, which is an excellent result.

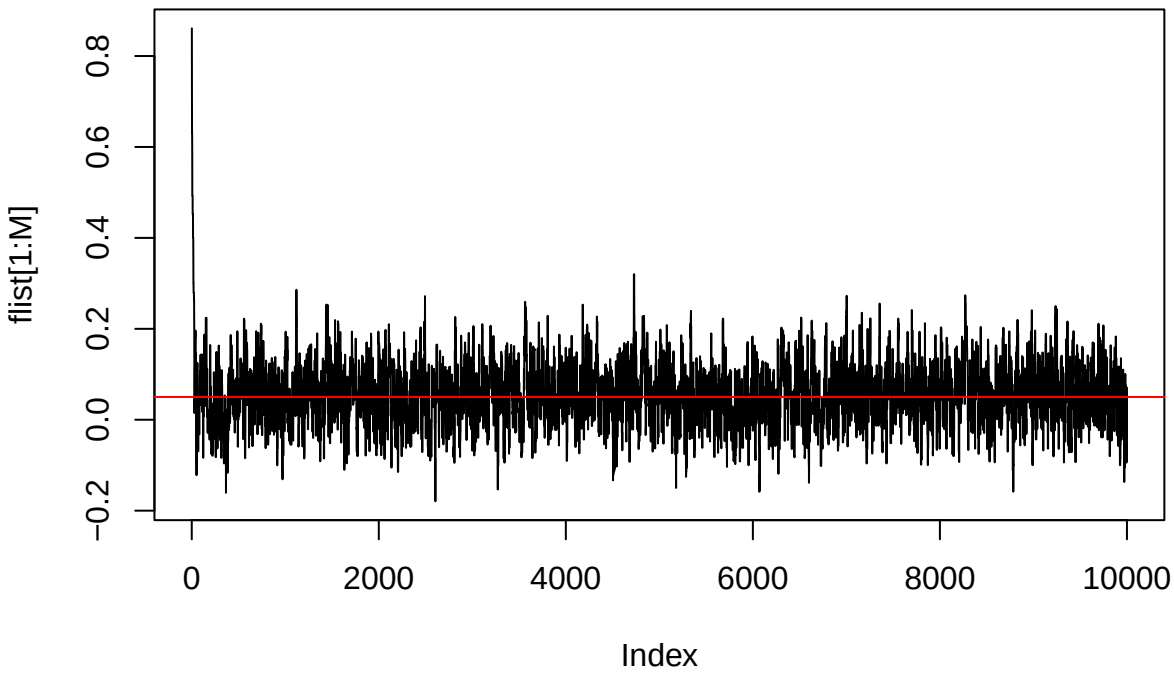

Figure 1: MK Chain converge compare to the true value in dataset 1

From the Figure 2 we see that the chain has a high degree of mixing, a low level of uncertainty and that it converges rapidly and remains quite near to the real value of f. (around 0.05).

3.3 M-H Algorithm on Dataset 2

Table of acceptance rate for different values of ϵ_p. Here $\epsilon_p = 0.006$ or 0.014, the acceptance rates are far from 0 and 1, and low standard error from 0.01 to 0.02. Holding $\epsilon_p = 0.01$, the algorithm was performed for 10000 iterations with a length of 1000 "burn-in", and we give our estimate for f:

Table 2: Acceptance rate by different epsilon p when k = 6

Epsilon	Acceptance Rate	Standard Error
0.001	0.7757	0.0152250
0.002	0.8177	0.0055514
0.003	0.7918	0.0018236
0.004	0.7407	0.0014351
0.005	0.6935	0.0013073
0.006	0.6444	0.0012522
0.007	0.5955	0.0011613
0.008	0.5612	0.0012465
0.009	0.5181	0.0010130
0.010	0.4823	0.0010515
0.011	0.4532	0.0010518
0.012	0.4229	0.0009680
0.013	0.3811	0.0009350
0.014	0.3631	0.0010044
0.015	0.3399	0.0010051
0.016	0.3231	0.0009907
0.017	0.2950	0.0008371
0.018	0.0000	0.0000000
0.019	0.2572	0.0008286
0.020	0.2460	0.0008862

$$\hat{f} = \frac{1}{(M-B)} \sum_{i=B+1}^{M} f_i = 0.05072752$$

and a 95% CI for estimator: (0.04837185, 0.05308320). Here: Figure 3 ,this CI covers our true theoretical value 0.05.

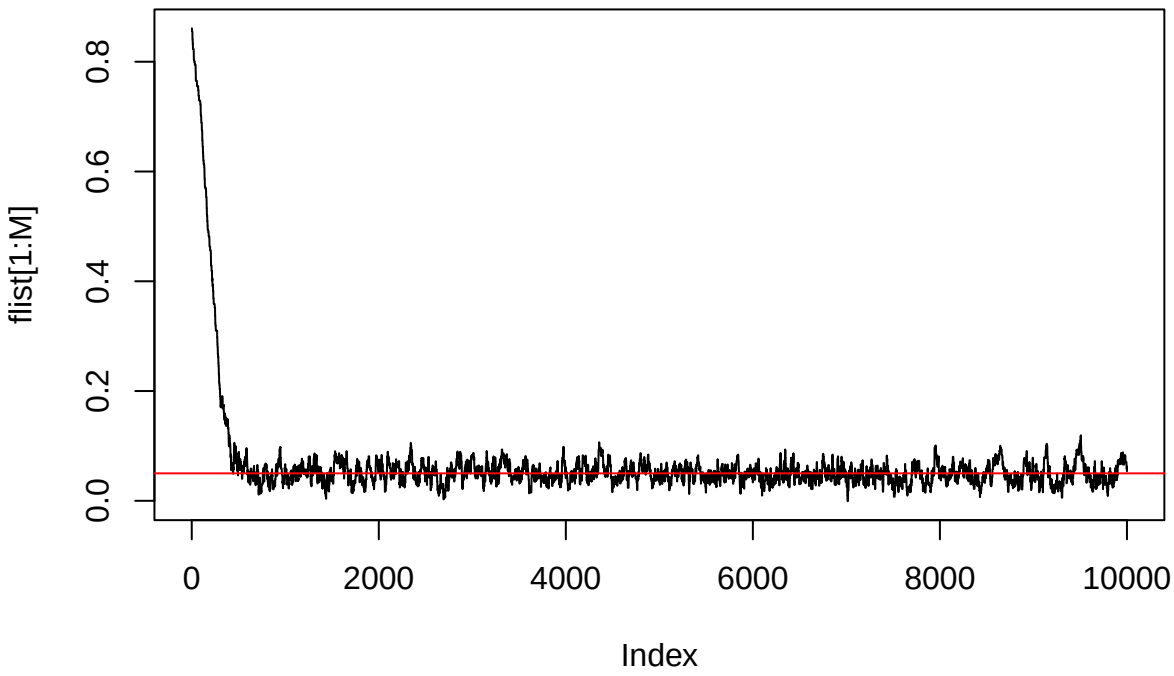

Figure 2: MK converges compares to the true value in dataset 2

According to Figure 4 MK in 1000 iterations, converges to true f value.

3.4 Gibbs Sampler

$$\hat{f} = \frac{1}{(M-B)} \sum_{i=B+1}^{M} f_i = 0.003558$$

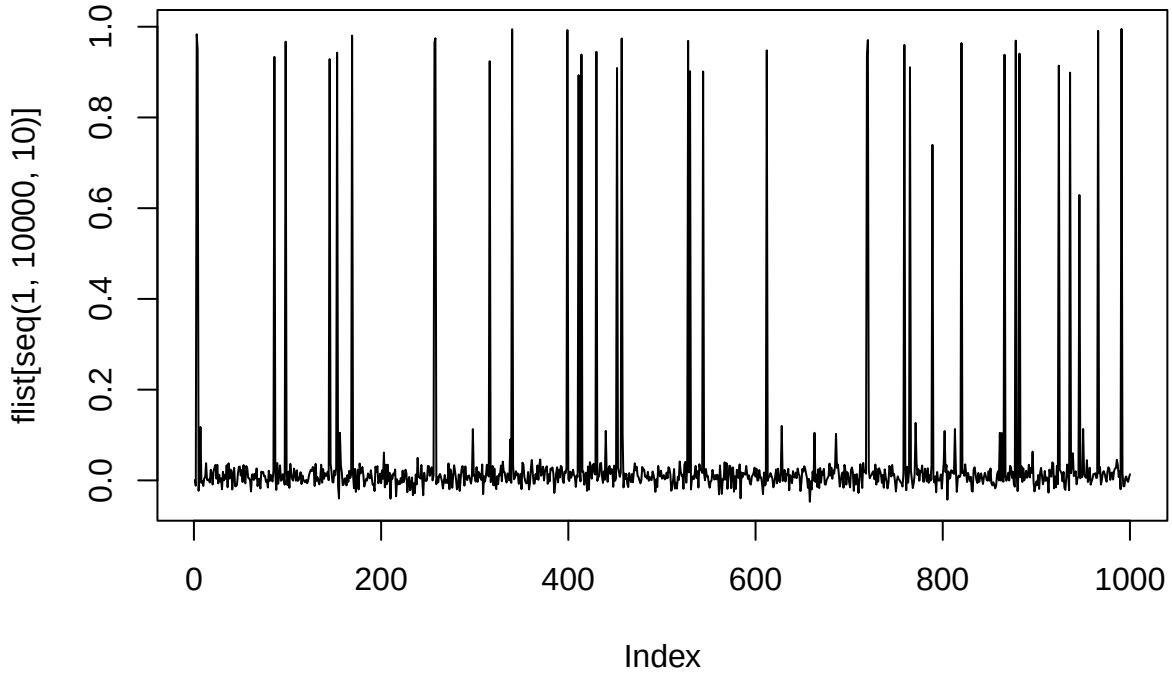

Figure 3: The chain converges compares to the true value under Gibbs sampler

Compared with Figure 5 it is again, that the MK converges to 0.03, performing not as close to our true value of 0.05. Additionally, we can see that it has a significant degree of uncertainty, which is attributable mostly to the fact that the coefficient, f, is associated with our p_i. When we used the conditional distribution to create p_i, we encountered a number of issues, which will be described in further depth in the discussion section.

3.5 Independence Sampler

The outcome is unsatisfactory since the chain does not genuinely converge, despite the fact that it briefly converges to our true value at the beginning. Even tuning the eps.p makes no difference.

$$\hat{f} = \frac{1}{(M-B)} \sum_{i=B+1}^{M} f_i = 0.02897391$$

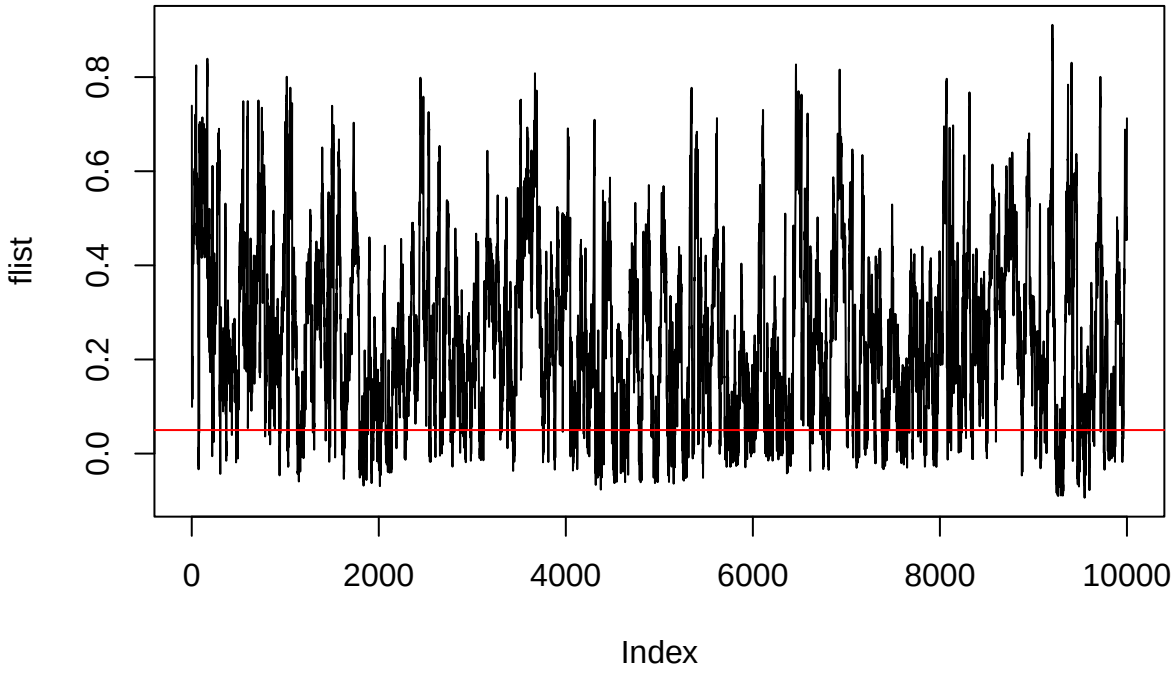

Figure 4: MK converges to true value under independence sampler

By examining the graph Figure 6 and the results, we can observe that Independence Sampler performs badly, converges far from our target (red line – 0.05), and has a very high degree of uncertainty. The explanation for this failure might be that although Y_n are unique and independent of $X_n - 1$, they are really associated when the formula is examined (1). We know that since people's genes are all associated in the genetic field, isolating a few genotypes would undoubtedly affect the inbreeding coefficient, which may be the primary reason for this algorithm's failure.

3.6 Importance Sampling

$$\hat{f} = 0.5205245$$

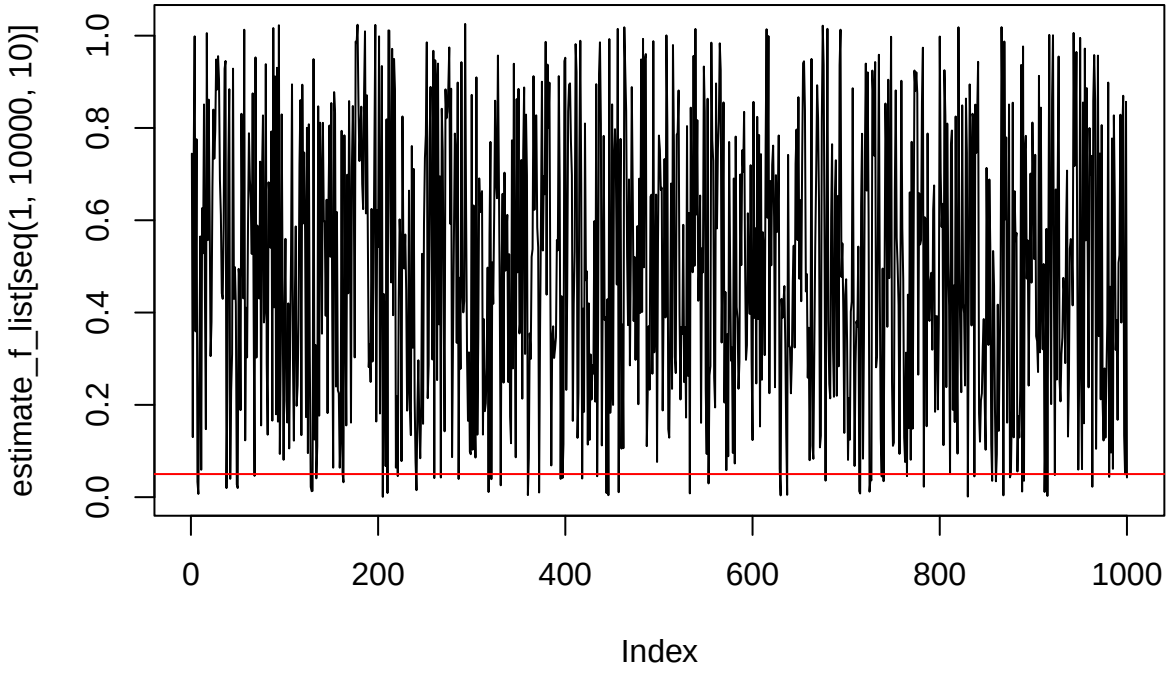

Figure 5: MK converges to true value under importance sampling

According to the graph Figure 7, the Importance Sampling algorithm performs fairly poorly even in the simplest scenario where k = 2. As indicated in Section 3.4, determining the distribution of f is extremely difficult, which is the primary explanation for these findings. Additionally, we discovered that although the Importance sampler could be utilised for certain basic, low-dimensional functions, it was hard to build for complex, high-dimensional functions.

4 Discussion

4.1 Conclusions in MCMC method

The most evident benefit of the MCMC method described here over more traditional approaches is that findings are graphically depicted as posterior density curves and hence easily interpretable. Additionally, the strategy allows for the incorporation of background data, which reduces the quantity of direct data necessary. The technique is adaptable and simple to apply. Apart from being practical, the strategy is strongly supported by statistical theory: there are good reasons to believe that uncertainty about an unknown parameter should be expressed by its probability distribution wherever possible (Smith & Bernardo, 1994).

Wide ranges of possible values frequently occur, especially when there are few distinguishable alleles and/or the sample size is small. The MCMC approach provides a clear and visible representation of the resultant uncertainty, which makes it superior to point estimation methods.

Due to the method's likelihood-based foundation, it is extremely adaptable, allowing for examination of the inbreeding model's validity. If the model does not appear to be logical, one can utilize the posteriors of the fixation indices fit to quantify the sample's divergence from HW.

Due to the fact that the conditional distributions for each parameter are quite distinct, we must build them from their density function for the Gibbs sampler. Initially, we considered using an MCMC method to generate the sample, but later decided to generate a sample each time by looking for the root of $U_n = F(x)$, where $Un \sim Unif[0, 1]$ is the cumulative distribution for the parameters conditional distribution and $F(x)$ is the cumulative distribution for the parameters conditional distribution, which is an inverse CDF approach.

However, since the conditional distribution is expressed as a product of the product and power of the number of observations, it might become rather tiny. To prevent the denominator from being zero while updating f, we set the parameters to 1020 if the computer recognizes it as a zero, particularly for $p_4, ..., p_6$.

Despite this, the values of the aforementioned parameters regularly approach zero, leading f to approach 1 repeatedly, as seen in the graph. Unfortunately, we were unable to address this issue by taking the log of the conditional distribution, since the scale of the CDF varies as the density increases, making finding roots for $U_n = log(F(x))$ very difficult and incorrect. Additionally, we used the R programme distr to produce random samples from the conditional distribution. However, when the function's power is large (e.g. > 4), it takes a very long time to generate the density as a distribution.

4.2 Weaknesses and next steps

In general, the MCMC algorithm that we utilize, Metropolis-Hastings-within-Gibbs, performed the best in both low- and high-dimension cases, giving us a reasonably accurate estimate of the real value. The proposed distribution, on the other hand, needs previous knowledge of the parameters.

All other algorithms, on the other hand, were unsatisfactory. Independent Sampler's estimate was significantly different from the genuine value, resulting in a substantial standard error. On the other hand, both the Importance Sampling and Rejection Sampler fared poorly, owing to the difficulty of obtaining K and $f(x)$. Gibbs Sampler gave a more accurate approximation of the real value than previous inferior MCMC methods, but with a large degree of uncertainty. Nonetheless, as long as we have good conditional distribution, the Gibbs sampler is a highly strong and efficient MCMC technique.

5 Reference

Ayres, K. L., & Balding, D. J. (1998). Measuring departures from Hardy–Weinberg: A Markov chain Monte Carlo method for estimating the inbreeding coefficient. Heredity, 80(6), 769–777. https://doi.org/10.1046/j.1365-2540.1998.00360.x

Crow, J. F., & Kimura, M. (2017). An introduction to population genetics theory. Scientific Publisher (India).

Hardy, G. H., & Galton, F. (1908). Mendelian proportions in a mixed population.

Hill, W. G., Babiker, H. A., Ranford-Cartwright, L. C., & Walliker, D. (1995). Estimation of inbreeding coefficients from genotypic data on multiple alleles, and application to estimation of clonality in malaria parasites. Genetical Research, 65(1), 53–61. https://doi.org/10.1017/s0016672300033000

Malecot, G. (1969). Measuring departures from Hardy–Weinberg: A Markov chain Monte Carlo method for estimating the inbreeding coefficient. The Mathematics of Heredity. https://doi.org/10.1046/j.1365-2540.1998.00360.x

NEI, M., & CHESSER, R. K. (1983). Estimation of fixation indices and gene diversities. Annals of Human Genetics, 47(3), 253–259. https://doi.org/10.1111/j.1469-1809.1983.tb00993.x

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL: https://www.R-project.org/.

Robertson, A., & Hill, W. G. (1984). Deviations from Hardy-Weinberg proportions: Sampling variances and use in estimation of inbreeding coefficients. Genetics, 107(4), 703–718. https://doi.org/10.1093/genetics/107.4.703

Weir, B. S. (1996). Genetic Data Analysis II - Sinauer Associates. Retrieved April 28, 2022, from https://www.sinauer.com/media/wysiwyg/tocs/WEIR2_TOC.pdf

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." Journal of Open Source Software 4(43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley. (2016). Ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Zhu, Hao. (2021). KableExtra: Construct Complex Table with 'Kable' and Pipe Syntax