Scientific Research and Essays Vol. 5(22), pp. 3447-3454, 18 November, 2010
Available online at http://www.academicjournals.org/SRE
ISSN 1992-2248 ©2010 Academic Journals

*Full Length Research Paper*

# A new algorithm for setting initial values for Markov Chain Monte Carlo in genetic linkage analysis via Gibbs sampling

**Gholamreza Jandaghi**

University of Tehran, Qom College, Iran. E-mail: jandaghi@ut.ac.ir.

Accepted 5 October, 2010

**In recent decades the Markov Chain Monte Carlo (MCMC) method has received a considerable attention in the area of genetic linkage analysis. Pedigree Gibbs sampler as a MCMC method like the other Markov Chain-based methods of resampling faces some difficulties in application. One of the difficulties is setting the initial genotypes consistent with observed phenotypes in order to make the chain start moving to different states (sets of consistent genotypes). In this paper a new algorithm for setting the initial genotypes for the pedigree under analysis is proposed. The proposed algorithm showed faster convergence than the existing algorithm. The efficiency of the new algorithm will be shown through examples. This algorithm is faster and more efficient than the existing algorithms.**

**Key words:** Markov Chain Monte Carlo, pedigree, linkage analysis, genotype, phenotype.

## INTRODUCTION

The calculation of the likelihood plays an important role in the analysis of genetic data. In many instances, the likelihood, can be written as a product of probabilities summed over all possible genotype configurations. The sum over genotypes can be computed easily along the lines of the Elston-Stewart algorithm (Elston and Stewart, 1971) and its extensions (Cannings et al., 1978; Janss et al., 1995; Lange and Boehnke, 1983; Lange and Elston, 1975; Stricker et al., 1995; Thomas, 1986a, 1986b). If exact peeling over all genotypic configurations is not possible, one alternative is to use MCMC procedures to sample genotypic configurations according to the posterior distribution. The Gibbs sampler (Geman and Geman, 1981) is an iterative procedure for drawing multiple dependent realizations from a distribution known only up to a proportionality constant. In genetics, the Gibbs sampler provides realizations from the distribution of genotypes $P_\theta(g)$, beginning from any initial realization of genotypes of the pedigree that is compatible with phenotypes. An individual's genotype is updated in turns by sampling from local conditional distributions at parameter values $\theta$, given the observed data (phenotypes) and genotypes of all other members of the pedigree. The configuration $g^{(i)}, i = 1,2,...$ obtained by successive cycles of Gibbs sampling is a sample from a Markov chain with stationary distribution $P_\theta(g \mid X)$, where g and X denote the genotypes and phenotypes of the pedigree respectively. The chain has been shown to be irreducible since for any starting genotype configuration, any set of states g that has positive probability can be hit in a finite number of steps(Sheehan and Thomas 1993). The usefulness of the Gibbs is based on the fact that although the joint distribution of $g_1, g_{2,}..., g_k$ may be complicated, the conditional distribution $f(g_i \mid g_1,...,g_{i-1},g_{i+1},...,g_k)$ may be simpler. This is the case in genetics where the conditional distribution depends only on a few of the other variables, that is, the conditional distribution of an individual in a pedigree given all other members depends only on the

genotypes of neighbors: his parents, his spouse(s) and his offspring.

One of the important problems in using Gibbs as a method of sampling is that of setting up the initial values for the chain to start from. Firstly, for complicated pedigrees, it is hard to set the initial genotypes compatible with phenotypes and consistent with other pedigree members. Usually for linked loci when one wants to set up initial genes using available methods, one ends up with an incompatible set of genotypes. Secondly, the initial values may lie in the tail of the probability distribution and then because of the nature of Gibbs it takes time for Gibbs to move to a highly probable area of the distribution. Some methods have been proposed for setting initial genotypes for the pedigree under analysis. Guo and Thompson (1992), in the context of choosing initial values, propose a method called posterior-gene-dropping. Abraham et al. (2007) propose an improved technique for setting initial genotype configurations in blocking Gibbs sampling. In this paper we propose an efficient method of setting initial genes and give some numerical indexes for its merits over the posterior gene dropping.

## TERMINOLOGY AND NOTATION

Consider a pedigree with $n$ individuals. Let g= ($g_1, g_{2,}..., g_n$) be the vector of genotypes of the individuals in the pedigree, where $g_i$ is the genotype of the $i$-th individual. Let $g_{-i} = (g_i \mid g_1,...,g_{i-1},g_{i+1},...,g_n)$. Let X = $(x_1, x_2,..., x_n)$ be the observed data, where $x_i$ is the observed phenotype of the $i$-th individual, $P(x \mid g)$ is the penetrance probability that is, the probability that an individual with genotype $g$ has phenotype $x$, and $P(g_k \mid g_f, g_m)$ is the transmission probability that is, the probability that an individual having genotype $g_k$ given the parental genotypes $g_f$ and $g_m$.

### Guo's algorithm

One of the important issues in using the Gibbs sampler as a method of sampling is setting the initial values. Since Gibbs samples are based on some initial state of a Markov chain, it is important to produce a good set of initial values in the sense that the chain takes a small number of iterations to reach its invariant distribution. Setting such initial values has been a time consuming job for researchers so far. Researchers have tried to develop algorithms to do this efficiently. Guo and Thompson (1992), in the context of choosing initial values, propose a method called posterior-gene-dropping. What they do is as follows:

Let $g_i$ and $x_i$ be the genotype and phenotype of individual $i$ in the pedigree, then for each founder they draw the initial value from

$$P_\theta(g_i = g \mid x_i) \propto P_\theta(g_{\ i} = g)\, P(x_i \mid g_i)$$

and the normalize for each $i$ so that the sum over $g$ is 1. Here $P_\theta(g_{\ i} = g)$ is just the population gene frequency calculated based on the assumptions of linkage and Hardy-Weinberg equilibrium and $P(x_i \mid g_i)$ is the penetrance probability which is simply set to 1 if the phenotype is missing. After assigning genes to founders, for each non-founder $i$ generate $g_i$ from the following probability distribution

$$P_\theta(g_i = g \mid x_i, g_{f_i}, g_{m_i}) \propto P_\theta(g_i = g \mid g_{\ f_i}, g_{m_i})\, P(x_i \mid g_i)$$

Where $f_i$ and $m_i$ are the parents of $i$ whose genotypes are already assigned. With linked markers, the procedure of posterior gene dropping may not yield consistent genes, because it is possible that $P_\theta(g_i = g \mid g_{\ f_i}, g_{m_i}) = 0$ for all possible $g$. The reason is that some genotypes assigned to the parents of $i$ may not be consistent with his phenotype. Guo and Thompson (1992) restart the procedure when there is any inconsistency. It is clear that for even small pedigrees the procedure can be time consuming. If one is dealing with complicated pedigrees, the procedure of setting initial values has a large impact on the time required for doing Gibbs sampling. Moreover, because this procedure picks one random set of genes, it is possible that the set is far from the center of the invariant distribution, that is, the set lies in the tail of the probability distribution. This also has some effect on the time consumed for Gibbs to arrive at the stationary distribution of the chain.

### The new algorithm

In the proposed algorithm, the procedure of setting initial configuration of genes is split into setting initial genes for each locus separately. Since consistency may be determined by examining each site separately, it follows that a consistent set of genes may be achieved by

specifying genes for each site separately. An acceptable collection of genes for one site requires fewer iterations, because the number of inconsistent genes is less. In other words, setting genes locus by locus saves time. The order of individuals is also important for the setting procedure, because if one wants to set the genes simultaneously, for complicated pedigrees, the size and genotypes of the neighborhood of individuals affects the procedure in the way that a large neighborhood may force the procedure to produce an inconsistent genotype for the person of interest. So, it is proposed firstly to set the genes locus by locus and secondly generation by generation from the top to the bottom of the pedigree.

First we assign the genes on the pedigree unconditional on their phenotypes, then for the first generation, we check whether or not the genes just dropped are consistent with their phenotypes. For those whose genotypes are consistent, in the next iteration, we generate from the conditional distribution of genotypes given phenotypes and for those inconsistent, we generate from the unconditional distribution and again check for the consistency of the genes in the current iteration. The procedure continues until the first generation's genotypes arte set. Repeat the procedure for the next generation and so on up to the last generation.

After setting genes for one locus, the whole process will be repeated for the second locus. Combining the genes of the two loci will result in a consistent set of genotypes for the pedigree. The algorithm briefly does the following:

Step 1: Simulate one set of genotypes for the pedigree based on unconditional distribution of genotypes given phenotypes, that is, for founders, generate from $P(g)$ which is the population gene frequencies and for non-founders generate from

$$P_\theta\left(g_i = g \mid g_{f_i}, g_{m_i}\right).$$

Step 2: For generation $j$ from 1 to $k$, check the consistency of genotypes for phenotypes. If individual $i$'s genotype is consistent:

-if $i$ is founder, generate from $P(g_i \mid x_i) \propto P(g_i)P(x_i \mid g_i)$

-if $i$ is a non-founder, generate from

$$P_\theta(g_i \mid x_i) \propto P_\theta\left(g_i \mid g_{f_i}, g_{m_i}\right)P(x_i \mid g_i).$$

If individual $i$'s genotype is inconsistent with his phenotypes:

-if $i$ is a founder, generate from $P(g)$

-if $i$ is a non-founder, generate from $P_\theta\left(g_i \mid g_{f_i}, g_{m_i}\right)$.

This procedure eventually ends producing a consistent set of genotypes for the pedigree.

**The proof of convergence for the new algorithm**

Without loss of generality, suppose we have consistent genes for individuals $1,2,...,i-1$. We prove that we can go to the state having the consistent gene for individuals $1,2,...,i-1,i$. This is possible since the chain is irreducible (Sheehan and Thomas 1993), by ignoring the phenotype information of individuals $i,i+1,...,n$ in the pedigree (relaxation of the phenotype constraints), the algorithm can go to all possible set of genotypes for the $i-th$ individual one or more of which is consistent. So, the algorithm can make every individual's gene consistent after some number of iterations and therefore converges.

**Comparison of the Guo's algorithm with the new algorithm**

Suppose we are dealing with two loci each having two alleles. Let the first locus have alleles $A$ and $a$ and the second locus have alleles $B$ and $b$. The genotypes of the first locus will be one of $AA, Aa, aA$ and $aa$ and those of the second locus will be one of $BB, Bb, bB$ and $bb$. Let phenotypes corresponding to the first locus be "$AA$ or others" and those corresponding to second locus be "$BB$ or others" (complete penetrance). Throughout this section, phenotypes $AA$ and $BB$ are denoted by 1 and "others" will be denoted by 2. Thus the phenotype of each person will be denoted by a pair of numbers. For example the individual with genotype $Aa/AB$ has phenotype (2,2), the one with genotype $AB/AB$ has phenotype (1,1), the one with genotype $aB/AB$ will have phenotype (2,1) and the person with genotype $Ab/AB$ will have phenotype (1,2). For example in the 17-member pedigree (Figure 1), if we assume that the disease is recessive with complete penetrance, then individuals 6 and 15 are sick.

The algorithm was applied to the 17-member pedigree (Figure 1). The phenotypes of the pedigree were {(1,2),

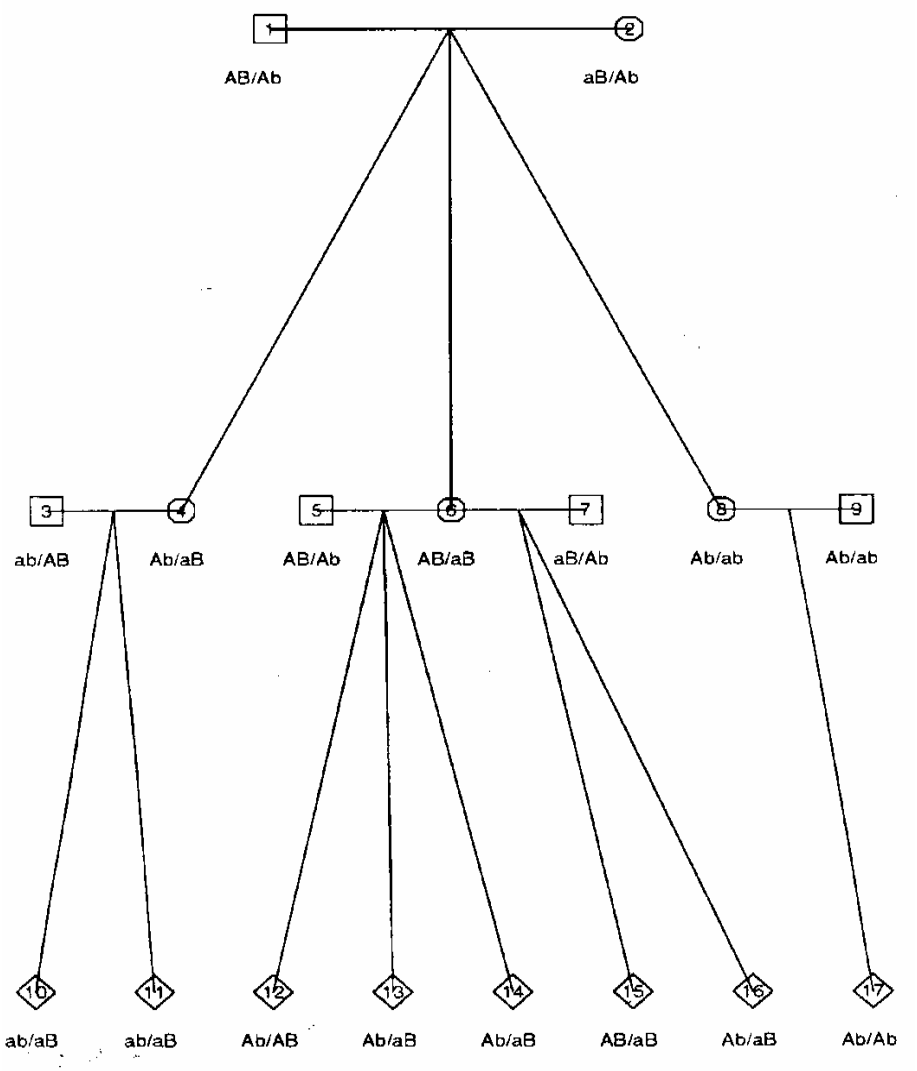


**Figure 1.** 17-member pedigree used for comparison of the new algorithm with the Guo's algorithm for setting initial genotypes.

(2,2), (2,2), (2,2), (1,2), (2,1), (2,2), (2,2), (2,2), (2,2), (2,2), (1,2), (2,2), (2,2), (2,1), (2,2), (1,2)} where each pair corresponds to the phenotypes of one individual. The first element of the pair corresponds to the phenotype assumed for the first locus and the second element corresponds to the phenotype assumed for the second locus. Figure 2 shows the distribution of the total number of iterations needed to set the initial genes for both loci.

To compare the efficiency of the new algorithm with the Guo's algorithm of posterior gene dropping, both algorithms were run on the 17-member pedigree for 100 runs. Note that the number of iterations for Guo's algorithm has a geometric distribution (Figure 3) and the number of iterations for the new algorithm has roughly the same shape (Figure 2). Under the null hypothesis of no difference, the distributions of the number of iterations for both algorithms are the same. To compare the efficiencies, we compare the parameters of the two geometric distributions using maximum likelihood estimation. Let $p$ be the probability that an iteration produces a consistent set of genes. Based on 100 different runs of both algorithms the maximum likelihood estimates of $p$ and its variance were calculated. The Guo's algorithm has $\hat{p} = 0.0133$ with $\hat{\sigma}_{\hat{p}} = 0.00134$ and the new algorithm has $\hat{p} = 0.0056$ with $\hat{\sigma}_{\hat{p}} = 0.00056$. So, for the 17-member pedigree Guo's algorithm was faster than the new algorithm. Next, wee ran both algorithms on a pedigree of size 61 (Figure 4) for 100 different runs. Guo's algorithm did not converge in any runs. So, it did not produce any consistent set of genes (we limited the number of

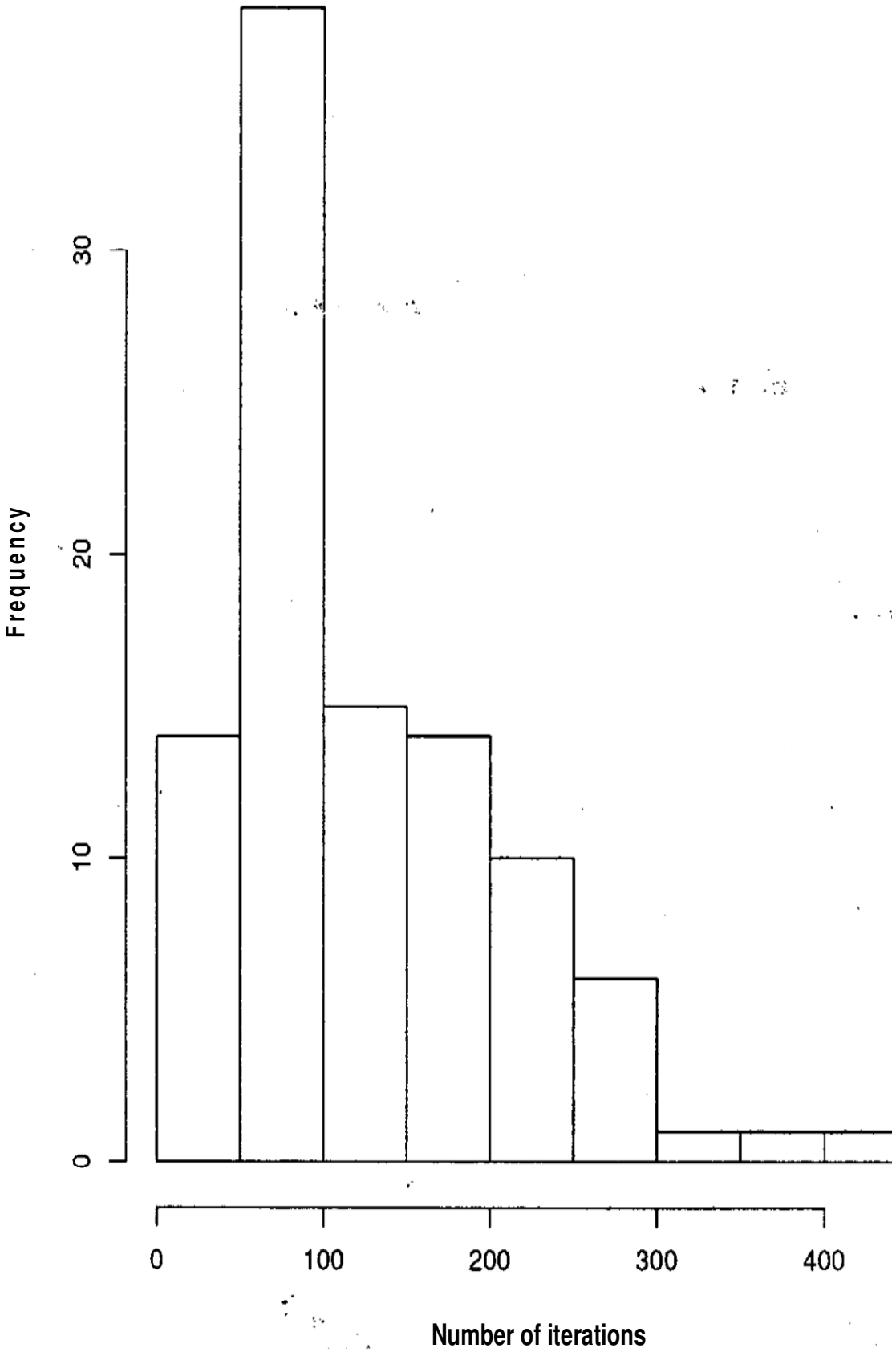


**Figure 2.** Distribution of number of iterations needed to set a consistent set of genes for both loci for the 17-member pedigree using the new algorithm.

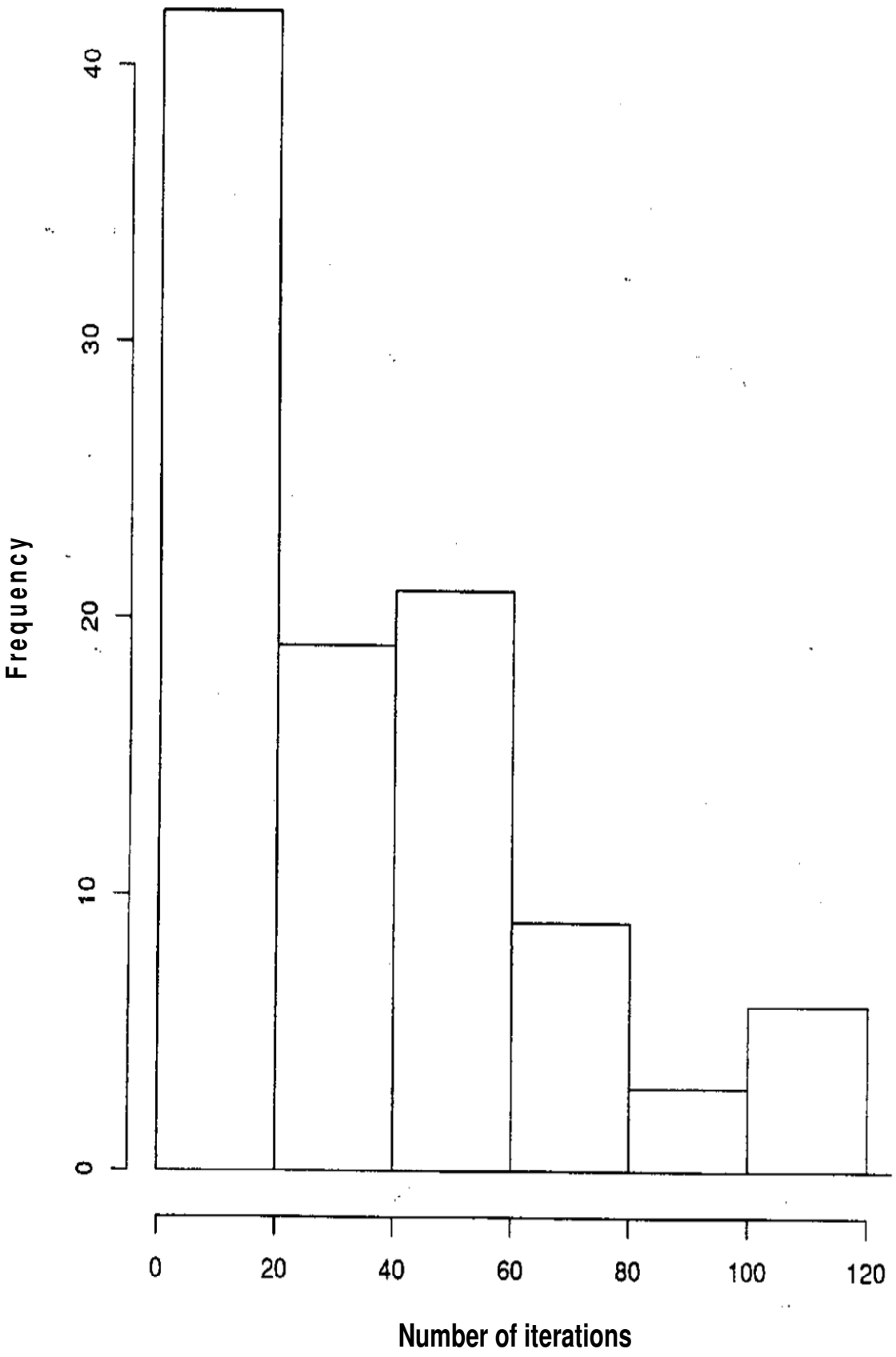


**Figure 3.** Distribution of the number of iterations needed to set a consistent set of genes for the 17-member pedigree using Guo's algorithm.

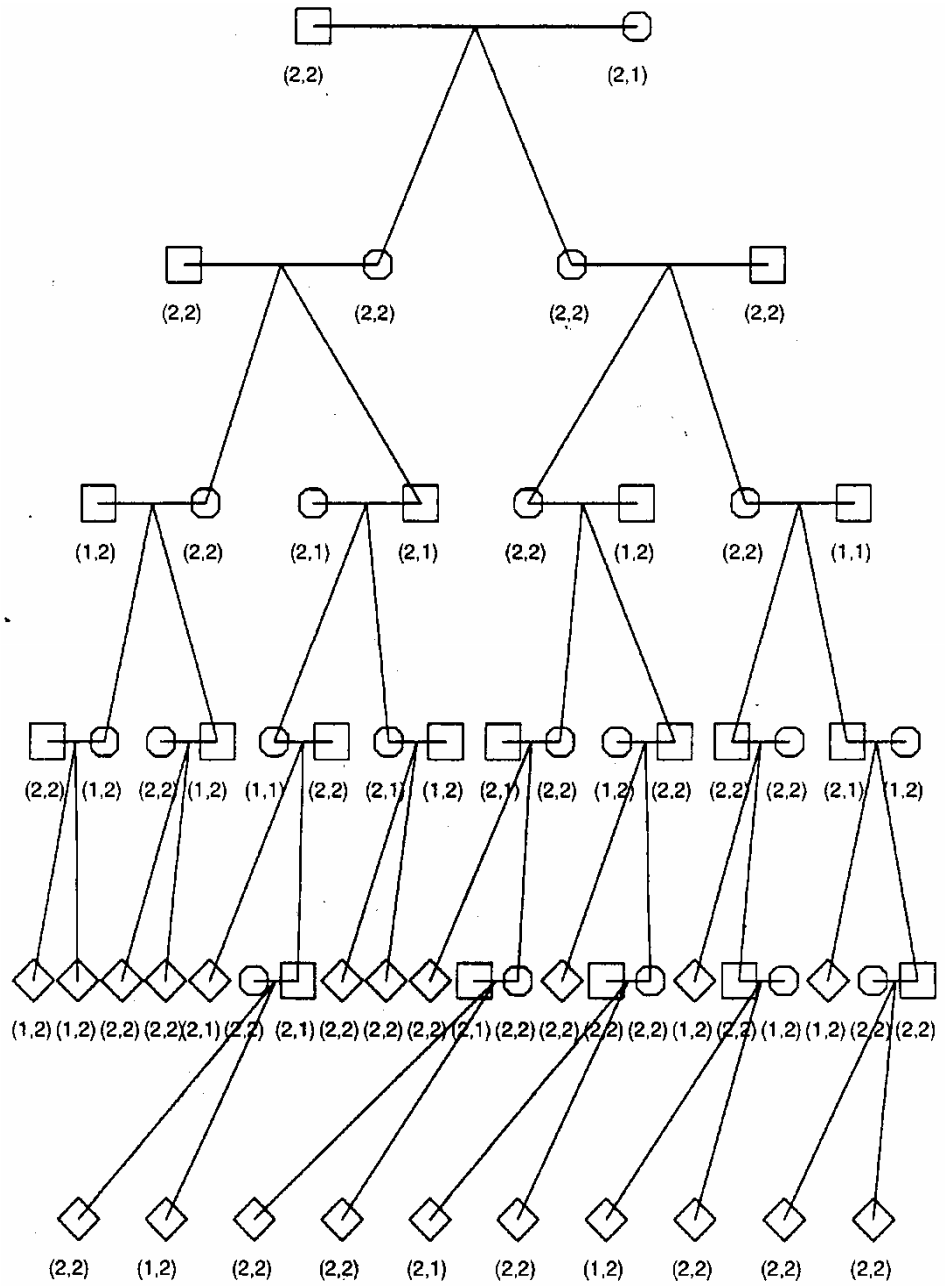


**Figure 4.** Pedigree of size 61 used for comparison of the new algorithm with Guo's algorithm. The pairs below the symbols represents the phenotypes of the person as explained in section.

iterations to 500) but the estimation of $p$ and its variance for the new algorithm were $\widehat{p} = 0.0037$ and $\hat{\sigma}_{\hat{p}} = 0.00037$.

## DISCUSSION AND CONCLUSION

Guo's algorithm works well for simple and small pedigrees but it fails for large complicated pedigrees while the new algorithm converges for pedigrees of arbitrary size. The new algorithm works only for diallelic loci, because for multiallelic loci the reducibility of the chain is the potential problem. For complex pedigrees, although the new algorithm seems to work, its feasibility needs more investigations. The algorithm is implemented in R programming language which may be slow. If it is programmed in a faster programming language, it will be more efficient. The efficiency of this algorithm against

Abraham et al. (2007) has not been checked and we leave it for future studies.

**REFERENCES**


Abraham KJ, Totir LR, Fernando RL (2007). Improved Techniques for Sampling Complex Pedigrees with the Gibbs Sampler, Genet. Sel. Evol., 39: 27-38.

Cannings C, Thompson EA, Skolnick MH (1978). Probability functions on complex pedigrees, Adv. Appl. Prob., 10: 26-61.

Elston RC, Stewart J (1971). A general model for genetic analysis of pedigree data, Hum. Hered., 21: 523-542.

Geman A, Geman D (1981). Stochastic Relaxation; Gibbs Distributions and the Bayesian Restoration of Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6: 721-741.

Guo SW, Thompson EA (1992). A Monte Carlo Method for Combined Segregation and Linkage Analysis. Am. J. Hum. Genet., 51: 1111-1126.

Janss LLG, van Arendonk JAM, van der Werf JHJ (1995). Computing approximate monogenic model likelihoods in large pedigrees with loops, Genet. Sel. Evol., 27: 567-579.

Lange K, Boehnke L (1983). Extensions to pedigree analysis. V. Optimal calculations of Mendelian likelihoods, Hum. Hered., 33: 291-301.

Lange K, Elston RC (1975). Extensions to pedigree analysis. I. Likelihood calculations for simple and complex pedigrees, Hum. Hered., 25: 95-105.

Sheehan N, Thomas A (1993). On the Irreducibility of a Markov Chain Defined on a Space of Genotype Configurations by a Sampling Scheme, Biometrics, 49: 163-175.

Stricker C, Fernando RL, Elston RC (1995). An algorithm to approximate the likelihood for pedigree data with loops by cutting, Theor. Appl. Gen., 91: 1054-1063.

Thomas A (1986a). Approximate computations of probability functions for pedigree analysis, IMJ J. Math. Appl. Med. Biol., 3: 157-166.

Thomas A (1986b). Optimal computations of probability functions for pedigree analysis, IMJ J. Math Appl. Med. Biol., 3: 167-178.