Network algorithm for the exact test of Hardy-Weinberg proportion for multiple alleles

Satoshi Aoki

Graduate School of Information Science and Technology
University of Tokyo, Tokyo, Japan

SUMMARY

We propose a new technique for the exact test of Hardy-Weinberg proportion that considerably extends the bounds of computational feasibility. Our algorithm is constructed analogously to a network algorithm for Freeman-Halton exact test in two-way contingency tables. In this algorithm, the smallest and largest values for the statistic are important and some interesting new theorems are proved for computing these values. Numerical examples are given to illustrate the practicality of the algorithm.
Keywords: Allele frequencies, Conditional reference set, Exact tests, Hardy-Weinberg, Multiple alleles, Network algorithm.

Correspondence to: Satoshi Aoki
Department of Mathematical Informatics, Graduate School of Information Science and Technology, University of Tokyo,
Bunkyo-ku Hongo 7-3-1, Tokyo 113-0033, Japan.
Tel: +81-3-5841-6942, Fax: +81-3-5841-6886
Email: aoki@stat.t.u-tokyo.ac.jp

1. Introduction

Since its discovery in the early 1900s, the Hardy-Weinberg law plays an important role in the field of population genetics and often serves as a basis for genetic inference (see, for example, Crow, 1988). This law states that in a large random-mating population with no selection, mutation or migration, the allele frequencies and the genotype frequencies are constant from generation to generation and that there is a simple relationship between the allele frequencies and the genotype frequencies.

Because of its importance, much attention has been paid to tests of the hypothesis that a population being sampled is in Hardy-Weinberg equilibrium. It has been recognized that the adequacy of applying classical goodness-of-fit tests of Hardy-Weinberg proportion is often questionable when the sample size or some genotypic frequencies are small (see, for example, Emigh, 1980). Although a variety of corrections for small sample sizes are proposed (Emigh and Kempthorne, 1975; Elston and Forthofer, 1977; Smith, 1986), it is found that they usually do not greatly improve the results obtained from the traditional goodness-of-fit tests (Emigh, 1980; Hernández and Weir, 1989).

From these reasons, use of exact tests, which do not rely on asymptotic theory, is desirable. Levene (1949) described the conditional distribution of a sample drawn from a population in Hardy-Weinberg equilibrium for an arbitrary number of alleles and Emith (1980) used Levene's distribution for the case of two alleles in his comparison of many statistical tests of Hardy-Weinberg hypothesis. Louis and Dempster (1987) proposed an algorithm for generating all possible samples for the exact distribution in an efficient manner. Their algorithm works well when the number of alleles is small (say, four or five). However, it is not of practical use for loci with more than a few alleles since the number of possible samples with the same gene frequencies and sample sizes grows exponentially with the number of alleles (Hernández and Weir, 1989). An alternative approach that avoids complete enumeration is the simulated method such as a conventional Monte Carlo method or a Markov chain method (Guo and Thompson, 1992). These simulated methods, however, have a disadvantage that the calculated p value distributes around the true value and is not exact in this sense. Considering the purpose of the exact test, it is prefered to compute the exact p value if possible, and simulated methods will not be dealt with in this paper.

The present article provides an improvement of the work of Louis and Dempster (1987). In this paper, we propose a new technique that considerably extends the bounds of computational feasibility of the exact test. Our algorithm is constructed analogously to a network algorithm proposed by Mehta and Patel (1983) for Freeman-Halton exact test (Freeman and Halton, 1951) in two-way contingency tables. Similarly to their algorithm, the computation of the smallest and largest values for the statistic plays an important role in our algorithm and some interesting new theorems are proved for computing these values.

The construction of this article is as follows. In section 2, an exact test of Hardy-Weinberg proportion for multiple alleles is formulated. In section 3, the network algorithm for computing the exact p values is given. In section 4, several new theorems for some optimizing problems are proved. Some numerical examples are given in section 5 to illustrate the practicality of our algorithm.

2. Exact test for multiple alleles

We assume that there are r distinct alleles, $A_1, A_2, ..., A_r$, of a given gene. If a sample of size N is drawn from a population of interest, the data can be expressed as the upper triangular array

	A_1	A_2	$\cdots$	A_r
A_1	x^o_{11}	x^o_{12}	$\cdots$	x^o_{1r}
A_2		x^o_{22}	$\cdots$	x^o_{2r}
$\vdots$			$\cdots$	$\cdots$
A_r				x^o_{rr}

where x^o_{ij} $(1 \le i \le j \le r)$ is the observed count of genotype A_iA_j in the sample. Throughout the paper we will use a vector notation like $\boldsymbol{X} = (x_{ij})$ to designate this type of table. For notational convenience, we write $x_{ij} = x_{ji}$ for $i > j$. We also define $\boldsymbol{y} = (y_1, y_2, ..., y_r)$ with $y_i = x^o_{ii} + \sum_{j=1}^r x^o_{ij}$, $i = 1, ..., r$. y_i is the number of A_i genes in the sample. Clearly we have $\sum_{i \le j} x^o_{ij} = N$ and $\sum_{i=1}^r y_i = 2N$. Let $\mathcal{F}$ denote the reference set of all possible counts of genotype with the same gene counts as $\boldsymbol{X}^o$:

$$\mathcal{F} = \left\{ \boldsymbol{X} \mid \boldsymbol{X} = (x_{11}, x_{12}, x_{22}, ..., x_{rr}),\ x_{ii} + \sum_{j=1}^{r} x_{ij} = y_i \text{ for } i = 1, ..., r \right\}.$$

We denote the number of elements in $\mathcal{F}$ by $\#\mathcal{F}$. Write $D = (2N)!/(N! \prod_{i=1}^r y_i!)$ for later use. Then, under Hardy-Weinberg proportions and conditional on $\boldsymbol{y}$, the probability of observing any $\boldsymbol{X} \in \mathcal{F}$ is expressed as (Levene, 1949)

$$P(\boldsymbol{X}) = \frac{N! \prod_{i=1}^r y_i!}{(2N)! \prod_{i \le j} x_{ij}!} 2^z = \frac{1}{D} \frac{2^z}{\prod_{i \le j} x_{ij}!},$$

where $z = \sum_{i<j} x_{ij} = N - \sum_{i=1}^r x_{ii}$ is the number of heterozygotes in the sample.

The p value for the conditional test of Hardy-Weinberg proportions is defined as the sum of probabilities of all the counts of genotype in $\mathcal{F}$ that are no more likely than $\boldsymbol{X}^o$ (see, for example, Chapco, 1976), that is,

$$p = \sum_{\boldsymbol{X} \in \mathcal{T}} P(\boldsymbol{X}), \tag{1}$$

where $\mathcal{T} = \{\boldsymbol{X} \mid \boldsymbol{X} \in \mathcal{F},\ P(\boldsymbol{X}) \le P(\boldsymbol{X}^o)\}$. Acceptance or rejection is based on a comparison of this value with some preset α level as in any statistical test. This test correspongs to the two-sided version of Fisher's exact test for 2×2 contingency table, or Freeman-Halton exact test for two-way contingency table.

3. The network representation and the algorithm

For calculating the p value defined by (1), one simple approach is to generate all the samples in $\mathcal{F}$. Louis and Dempster (1987) described how to generate all the samples in $\mathcal{F}$ and computed the exact p values for some examples with three or four alleles. Their naive algorithm is, however, very time-consuming if $\#\mathcal{F}$ is large. In this paper, a new algorithm is proposed that does not require total enumeration of the reference set. This algorithm is a natural extension of the network algorithm for computing Freeman-Halton

exact p values for two-way contingency table (Mehta and Patel, 1983). First we provide a network representation for the reference set $\mathcal{F}$.

The network representation consists of *nodes* and *arcs* constructed in $r+1$ stages. For $k = r, r-1, ..., 1, 0$, the nodes at stage k have the form $(k, Y_{1,k}, Y_{2,k}, ..., Y_{k,k}) \equiv (k, \boldsymbol{Y}_k)$. There are as many nodes at stage k as there are possible partial sums of genes for the first k alleles. Arcs emanate from each node at stage k and every arc is connected to only one node at stage $k-1$. The network is constructed recursively by specifying all successor nodes $(k-1, \boldsymbol{Y}_{k-1})$ that are connected by arcs to each node $(k, \boldsymbol{Y}_k)$. The range of $Y_{i,k}$, $i = 1, ..., k$, for these successor nodes is obtained from using the algorithm of Louis and Dempster (1987). There is only one node at stage r, the initial node, which is labeled $(r, \boldsymbol{Y}_r) \equiv (r, Y_{1,r}, ..., Y_{r,r}) = (r, y_1, ..., y_r) = (r, \boldsymbol{y})$. There is also only one node at stage 0, the terminal node, which is labeled (0). A path through the network is a sequence of arcs

$$(r, \boldsymbol{Y}_r) \to (r-1, \boldsymbol{Y}_{r-1}) \to \cdots \to (2, \boldsymbol{Y}_2) \to (1, \boldsymbol{Y}_1) \to (0).$$

One can verify that each path represents a distinct element in $\mathcal{F}$, with the relations

$$x_{11} = \frac{1}{2} Y_{1,1}, \tag{2}$$

$$x_{ik} = Y_{i,k} - Y_{i,k-1}, \quad i = 1, ..., k-1, \quad k = 2, ..., r, \tag{3}$$

and

$$x_{kk} = \frac{1}{2} \left(Y_{k,k} - \sum_{i=1}^{k-1} x_{ik} \right), \quad k = 2, ..., r. \tag{4}$$

Figure 1 shows the network representation for three alleles case with gene counts $(y_1, y_2, y_3) = (6, 5, 3)$. The dotted path gives the array of counts $\boldsymbol{X} = (x_{11}, x_{12}, x_{13}, x_{22}, x_{23}, x_{33}) = (2, 1, 1, 2, 0, 1)$.

We define the length of an arc from node $(k, \boldsymbol{Y}_k)$ to $(k-1, \boldsymbol{Y}_{k-1})$ by

$$ARC(k, \boldsymbol{Y}_k, \boldsymbol{Y}_{k-1}) = \frac{2^{\sum_{i=1}^{k-1} (Y_{i,k} - Y_{i,k-1})}}{\left[\frac{1}{2} \{ Y_{k,k} - \sum_{i=1}^{k-1} (Y_{i,k} - Y_{i,k-1}) \} \right] ! \times \prod_{i=1}^{k-1} (Y_{i,k} - Y_{i,k-1})!}.$$

The length of path or sub-path is defined as the product of the corresponding arc lengths. Then it is straightforward to verify that the length of complete path from the initial node to the terminal node is equal to $D \cdot P(\boldsymbol{X})$ by using the relations (2), (3) and (4).

Now our goal is to identify and sum all paths whose length do not exceed $D \cdot P(\boldsymbol{X}^o)$. If we systematically enumerate each path through the network, compute its length and sum the path lengths that does not exceed $D \cdot P(\boldsymbol{X}^o)$, we are in effect considering all the elements in $\mathcal{F}$. This is nothing but the algorithm of Louis and Dempster (1987) and usually computationally infeasible if $\#\mathcal{F}$ is large.

To avoid such total enumeration, we compute at each node $(k, \boldsymbol{Y}_k)$ the shortest and longest values of the sub-path from the node $(k, \boldsymbol{Y}_k)$ to the terminal node. We call these sub-paths as LP (longest sub-path) or SP (shortest sub-path) according to Mehta and Patel (1983). On the other hand, the length of the sub-path from the initial node to the current node $(k, \boldsymbol{Y}_k)$ is calculated from the labels $(r, \boldsymbol{Y}_r), ..., (k, \boldsymbol{Y}_k)$ as

$$PAST = \prod_{j=k+1}^{r} ARC(j, \boldsymbol{Y}_j, \boldsymbol{Y}_{j-1}).$$

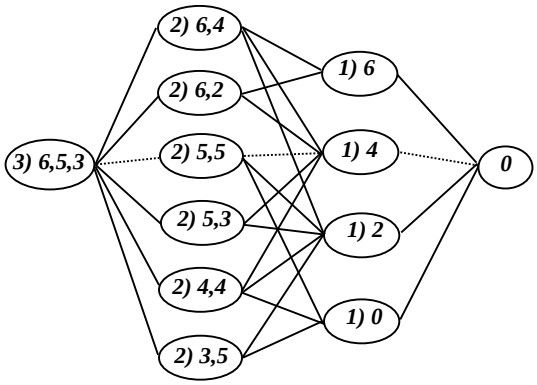

Figure 1. Network representation for three alleles case with $(y_1, y_2, y_3) = (6, 5, 3)$.

Now we can determine whether all the paths having a common sub-path $(r, \boldsymbol{Y}_r) \to \cdots \to (k, \boldsymbol{Y}_k)$ do or do not contribute to the p value, without processing the remaining parts of paths as follows.

- Case 1. If
$$PAST \cdot LP(k, \boldsymbol{Y}_k) \leq D \cdot P(\boldsymbol{X}^o), \tag{5}$$
then the lengths of all paths having common sub-path $(r, \boldsymbol{Y}_r) \to \cdots \to (k, \boldsymbol{Y}_k)$ are not greater than $D \cdot P(\boldsymbol{X}^o)$. Hence the lengths of all these paths contribute the p value.

- Case 2. If
$$PAST \cdot SP(k, \boldsymbol{Y}_k) > D \cdot P(\boldsymbol{X}^o), \tag{6}$$
then the lengths of all paths having common sub-path $(r, \boldsymbol{Y}_r) \to \cdots \to (k, \boldsymbol{Y}_k)$ exceed $D \cdot P(\boldsymbol{X}^o)$. Hence none of these paths contributes to the p value.

- Case 3. Otherwise, we consider the next stage (stage $k-1$).

It should be noted that the sum of all the sub-path lengths from the node $(k, \boldsymbol{Y}_k)$ to the terminal node is equal to $(2N_k)!/(N_k! \prod_{i=1}^{k} Y_{i,k})$, where $N_k = \frac{1}{2}\sum_{i=1}^{k} Y_{i,k}$. This relation is derived in the same manner as Levene (1949). Then the contribution to the p value in the case 1 equals $PAST \cdot (2N_k)!/(N_k! \prod_{i=1}^{k} Y_{i,k})$. Consequently, we need not enumerate the remaining parts of paths for the case 1 or 2. In the case 3, we consider the common sub-path to a node $(k-1, \boldsymbol{Y}_{k-1})$ at stage $k-1$ which is connected to the node $(k, \boldsymbol{Y}_j)$, and proceed to verify (5) and (6) in the same manner as before.

The only remaining problem is to compute LP and SP at each node. If we can evaluate LP and SP exactly, we can *trim* paths perfectly. It is worth pointing out, however, that if we can only evaluate an upper bound for LP or a lower bound for SP, we can make incomplete trimming. For Freeman-Halton exact test in two-way contingency table, Mehta and Patel (1983) evaluated an upper bound for LP and a lower bound for SP. For the Hardy-Weinberg case, we obtain the closed form expression of exact SP value in the next section. As for LP, although no closed form of exact LP value is available, we present two upper bounds for LP.

4. Computation of the shortest and longest paths

4.1 A closed form expression of SP

First we present a closed form expression of $SP(k, \boldsymbol{Y}_k)$. Before we state a theorem we define $I = \{1, 2, ..., k\}$, $I_e = \{i \mid Y_{i,k} \text{ is even}\}$ and $I_o = \{i \mid Y_{i,k} \text{ is odd}\}$. We also define the following decomposition of the set I_o as $I_o = I_o^* \cup \tilde{I}_o$, $I_o^* \cap \tilde{I}_o = \emptyset$, where I_o^* is the maximal set made from the unions of pair (i, j) such that $Y_{i,k} = Y_{j,k}$ and $\tilde{I}_o = I_o - I_o^*$ is the remaining set satisfying $Y_{i,k} \neq Y_{j,k}$ for all $i, j \in \tilde{I}_o$, $i \neq j$. If $\boldsymbol{Y}_k = (13, 12, 11, 11, 11, 10, 9, 8, 5, 5, 3, 3, 3)$, for example, we have $I_e = \{2, 6, 8\}$, $I_o^* = \{3, 4, 9, 10, 11, 12\}$ and $\tilde{I}_o = \{1, 5, 7, 13\}$. (Although the elements of $\tilde{I}_o$ and I_o^* are not unique, corresponding values of $Y_{i,k}$ are uniquely determined.) It should be noted that, by definition, $Y_{1,k} + \cdots + Y_{k,k}$, $\#I_o$, $\#\tilde{I}_o$ and $\#I_o^*$ are all even numbers. Using these sets, our problem can be written in the following form:

$$\mathrm{P}_1 : \text{minimize } \frac{2^z}{\prod_{1 \le i \le j \le k} x_{ij}!}, \quad z = \sum_{1 \le i < j \le k} x_{ij}, \tag{7}$$

subject to

$$x_{ij} + \sum_{j=1}^{k} x_{ij} = 2m_i, \quad \text{for } i \in I_e, \tag{8}$$

$$x_{ij} + \sum_{j=1}^{k} x_{ij} = 2m_i + 1, \quad \text{for } i \in I_o, \tag{9}$$

$$x_{ji} = x_{ij}, \tag{10}$$

$$x_{ij} \text{ is a non-negative integer for } i, j = 1, ..., k. \tag{11}$$

A solution of P_1 is given in the following theorem.

Theorem 1 *The optimal objective function value of P_1 is given by*

$$2^{z^*} \left(\prod_{i \in I_e} \frac{1}{m_i!} \right) \left(\prod_{i \in \tilde{I}_o} \frac{1}{m_i!} \right) \left\{ \prod_{i \in I_o^*} \frac{1}{(2m_i + 1)!} \right\}^{1/2}, \quad \textit{where} \quad z^* = \frac{1}{2} \left\{ \sum_{i \in I_o^*} (2m_i + 1) + \#\tilde{I}_o \right\}. \tag{12}$$

Hereafter, we define $\boldsymbol{X}^* = (x_{11}^*, x_{12}^*, x_{22}^*, ..., x_{kk}^*)$ as one of the solutions of P_1 that minimizes (7) subject to (8), (9), (10) and (11). To prove the above theorem, we prepare the following lemma.

Lemma 1 *The optimal solution $\boldsymbol{X}^*$ satisfies the following conditions.*

(a) x_{ij}^*, $i \neq j$, cannot be a positive even number.

(b) $\{x_{i1}^*, ..., x_{ii-1}^*, x_{ii+1}^*, ..., x_{ik}^*\}$ includes at most one odd number for all i.

Proof of Lemma 1.

(a) Suppose that $x^*_{ij} = 2n, n \geq 1$, for some i, j $(i \neq j)$. Consider another solution $\boldsymbol{X}^{'} = (x^{'}_{11}, ..., x^{'}_{kk})$, where

$$\begin{cases} x^{'}_{ii} = x^*_{ii} + n, \ x^{'}_{jj} = x^*_{jj} + n, \ x^{'}_{ij} = 0, \\ x^{'}_{ij} = x^*_{ij} \ \text{ for all the other } i, j. \end{cases}$$

Clearly, $\boldsymbol{X}^{'}$ satisfies (8),(9),(10) and (11). Let OF^* be the value of the objective function under $\boldsymbol{X}^*$ and $OF^{'}$ be the value of the objective function under $\boldsymbol{X}^{'}$. Then we have

$$\frac{OF^*}{OF^{'}} = \frac{2^{2n}(n!)^2}{(2n)!} \begin{pmatrix} x^*_{ii} + n \\ x^*_{ii} \end{pmatrix} \begin{pmatrix} x^*_{jj} + n \\ x^*_{jj} \end{pmatrix} \geq \frac{2^{2n}(n!)^2}{(2n)!} \equiv f_1(n),$$

$$\frac{f_1(n+1)}{f_1(n)} = \frac{2(n+1)}{2n+1} > 1$$

and $f_1(n) > f_1(n-1) > \cdots > f_1(1) = 2 > 1$. Hence $OF^* > OF^{'}$ holds. This contradicts that OF^* is the optimal objective function value.

(b) Suppose that $x^*_{ij_1} = 2n_1 + 1, x^*_{ij_2} = 2n_2 + 1, \ n_1, n_2 \geq 0, j_1 \neq i, j_2 \neq i$ for some j_1, j_2 $(j_1 \neq j_2)$. Consider another solution $\boldsymbol{X}^{'}$, where

$$\begin{cases} x^{'}_{ii} = x^*_{ii} + n_1 + n_2 + 1, \ x^{'}_{j_1 j_1} = x^*_{j_1 j_1} + n_1, \ x^{'}_{j_2 j_2} = x^*_{j_2 j_2} + n_2, \\ x^{'}_{ij_1} = x^{'}_{ij_2} = 0, \ x^{'}_{j_1 j_2} = x^*_{j_1 j_2} + 1, \\ x^{'}_{ij} = x^*_{ij} \ \text{ for all the other } i, j. \end{cases}$$

Clearly, $\boldsymbol{X}^{'}$ satisfies (8),(9),(10) and (11). Let $OF^{'}$ be the value of the objective function under $\boldsymbol{X}^{'}$. Then we have

$$\begin{aligned} \frac{OF^*}{OF^{'}} &= \frac{2^{2n_1+2n_2+1} n_1! n_2! (n_1+n_2+1)!}{(2n_1+1)!(2n_2+1)!} \begin{pmatrix} x^*_{ii} + n_1 + n_2 + 1 \\ x^*_{ii} \end{pmatrix} \begin{pmatrix} x^*_{j_1 j_1} + n_1 \\ x^*_{j_1 j_1} \end{pmatrix} \\ &\qquad \times \begin{pmatrix} x^*_{j_2 j_2} + n_2 \\ x^*_{j_2 j_2} \end{pmatrix} (x^*_{j_1 j_2} + 1) \\ &\geq \frac{2^{2n_1+2n_2+1} n_1! n_2! (n_1+n_2+1)!}{(2n_1+1)!(2n_2+1)!} \equiv f_2(n_1, n_2) \end{aligned}$$

and

$$\frac{f_2(n_1+1, n_2)}{f_2(n_1, n_2)} = \frac{2(n_1+n_2+2)}{2n_1+3} \geq \frac{2(n_1+2)}{2n_1+3} > 1.$$

Similarly we have $\dfrac{f_2(n_1, n_2+1)}{f_2(n_1, n_2)} > 1$. Hence $f_2(n_1, n_2) > f_2(0, 0) = 2 > 1$ and $OF^* > OF^{'}$ holds. This contradicts that OF^* is the optimal objective function value. Q.E.D.

Now we prove the Theorem 1 using the above lemma.

Proof of the Theorem 1.
As a direct result of the Lemma 1, we have $x^*_{ii} = m_i, \ x^*_{ij} = 0, \ j \neq i$, for all $i \in I_e$ since the number of odd values in $\{x^*_{i1}, ..., x^*_{ii-1}, x^*_{ii+1}, ..., x^*_{ik}\}$ is even for all $i \in I_e$. On the other

hand, we see that the elements of I_o are separated into *pairs* as $(i_1, j_1), (i_2, j_2), ..., (i_p, j_p)$ such that

$$\begin{aligned} x^*_{ij} &> 0, \quad \text{if } (i,j) \text{ is a pair,} \\ &= 0, \quad \text{otherwise,} \end{aligned}$$

and $p = \#I_o/2$ is the number of the pairs. Then the optimal objective function value of P_1 can be written as

$$2^{z^*} \left(\prod_{i \in I_e} \frac{1}{m_i!} \right) \left(\prod_{n=1}^{p} \frac{1}{x^*_{i_n i_n}! x^*_{j_n j_n}! x^*_{i_n j_n}!} \right), \quad z^* = \sum_{n=1}^{p} x^*_{i_n j_n}. \tag{13}$$

Hereafter we call (i,j) an *identical pair* if $m_i = m_j$ and a *different pair* if $m_i \neq m_j$. It is worth pointing out that $i, j \in I^*_o$ for all identical pairs (i,j).

First we consider the identical pair (i,j). Let $m_i = m_j \equiv m$ and

$$x^*_{ij} = 2(m-n) + 1, \quad x^*_{ii} = x^*_{jj} = n \tag{14}$$

for these i, j. Now we show that n has to be zero, that is, $\min\limits_{0 \leq n \leq m} OF(n) = OF(0)$, where $OF(n)$ is the objective function value when x^*_{ij}, x^*_{ii} and x^*_{jj} of $\boldsymbol{X}^*$ are given by (14) for $n = 0, ..., m$. We have

$$\frac{OF(n+1)}{OF(n)} = \frac{(2m-2n+1)(m-n)}{2(n+1)^2}.$$

If we compare this ratio to 1 for $n = 0, 1, ..., m$, then we have

$$\frac{OF(n+1)}{OF(n)} < 1 \text{ for } n > \frac{2m^2 + m - 2}{4m+5}$$

and

$$\frac{OF(n+1)}{OF(n)} > 1 \text{ for } n < \frac{2m^2 + m - 2}{4m+5}$$

and hence $\min\limits_{0 \leq n \leq m} OF(n) = \min\{OF(0), OF(m)\}$. Besides we have

$$\frac{OF(m)}{OF(0)} = \frac{(2m+1)!}{2^{2m}(m!)^2} \equiv f_3(m)$$

and

$$\frac{f_3(m+1)}{f_3(m)} = \frac{2m+3}{2(m+1)} > 1.$$

Hence

$$f_3(m) > f_3(m-1) > \cdots > f_3(0) = 1 \tag{15}$$

and $OF(m) > OF(0)$. We have now shown that

$$\begin{cases} x^*_{ij} = 2m_i + 1, \\ x_{is} = 0, \text{ for } s \neq j, \\ x_{js} = 0, \text{ for } s \neq i, \end{cases} \tag{16}$$

for the identical pair (i,j).

Next we consider the different pair (i,j). We can assume $m_i > m_j$ without loss of generality. Similarly to the case of the identical pair, we denote

$$x^*_{ii} = m_i - n, \quad x^*_{jj} = m_j - n, \quad x^*_{ij} = 2n+1$$

and consider the sequence $OF(n), n = 0, 1, ..., m_j$. The ratio is written as

$$\frac{OF(n)}{OF(0)} = \prod_{k=0}^{n-1} \left\{ \frac{2(m_i - k)}{2(n-k)+1} \times \frac{m_j - k}{n-k} \right\} \geq \prod_{k=0}^{n-1} \frac{2(m_i - k)}{2(n-k)+1}.$$

From $m_i > m_j$, we have $m_i - n > m_j - n \geq 0$ and then $m_i \geq n+1$ holds. Hence we have

$$\frac{OF(n)}{OF(0)} \geq \prod_{k=0}^{n-1} \frac{2(n+1-k)}{2(n-k)+1} > 1$$

and $OF(n) > OF(0)$. We have shown that

$$\begin{cases} x^*_{ii} = m_i, \ x^*_{jj} = m_j, \ x^*_{ij} = 1, \\ x_{is} = x_{js} = 0 \text{ for } s \neq i, j \end{cases} \tag{17}$$

for the different pair (i,j).

Now we show that the pairs have to be constructed in such a way that the number of identical pairs is maximized. Clearly it is sufficient to consider the case of four alleles, $\boldsymbol{Y}_k = (Y_{1,k}, Y_{2,k}, Y_{3,k}, Y_{4,k}) = (2m_1+1, 2m_1+1, 2m_3+1, 2m_4+1)$ where $m_1 \neq m_3$ and $m_1 \neq m_4$.

(i) If we make pairs as $(1,3)$ and $(2,4)$, then the optimal $\boldsymbol{X}^*$ is obtained from (17) as

$$\begin{cases} x^*_{11} = x^*_{22} = m_1, \ x^*_{33} = m_3, \ x^*_{44} = m_4, \ x^*_{13} = x^*_{24} = 1, \\ \text{otherwise } x^*_{ij} = 0. \end{cases}$$

(ii) Similarly, if we make pairs as $(1,2)$ and $(3,4)$, $\boldsymbol{X}^*$ is written as follows:

- If $m_3 = m_4$, then
$$\begin{cases} x^*_{12} = 2m_1 + 1, \ x^*_{34} = 2m_3 + 1, \\ \text{otherwise } x^*_{ij} = 0. \end{cases}$$
- If $m_3 \neq m_4$, then
$$\begin{cases} x^*_{12} = 2m_1 + 1, \ x^*_{33} = m_3, \ x^*_{44} = m_4, \ x^*_{34} = 1, \\ \text{otherwise } x^*_{ij} = 0. \end{cases}$$

Let OF_i and OF_{ii} denote the objective function values corresponding to *(i)* and *(ii)*, respectively.

- If $m_3 = m_4$, then
$$\frac{OF_i}{OF_{ii}} = \frac{(2m_1+1)!}{2^{2m_1}(m_1!)^2} \cdot \frac{(2m_3+1)!}{2^{2m_3}(m_3!)^2} = f_3(m_1) f_3(m_3).$$

From (15), we have $OF_i > OF_{ii}$ in this case.

- If $m_3 \neq m_4$, then

$$\frac{OF_i}{OF_{ii}} = \frac{(2m_1+1)!}{2^{2m_1}(m_1!)^2} = f_3(m_1).$$

Again from (15), we have $OF_i > OF_{ii}$.

From these considerations, it is shown that the case of *(i)* is not optimal. In other words, all the different pairs have to be included in $\tilde{I}_o$ and all the identical pairs have to be included in I_o^*. Substitution of (16) and (17) into (13) corresponding to $\tilde{I}_o$ and I_o^* and some simplification yields (12). Q.E.D.

4.2 Some upper bounds for LP

Next we consider $LP(k, \boldsymbol{Y}_k)$. The problem we consider is

$$\mathrm{P}_2 : \text{maximize } \frac{2^z}{\prod\limits_{1\le i\le j\le k} x_{ij}!}, \quad z = \sum_{1\le i<j\le k} x_{ij},$$

subject to

$$x_{ij} + \sum_{j=1}^{k} x_{ij} = Y_{i,k}, \quad \text{for } i = 1, ..., k \tag{18}$$

and (10),(11). Unfortunately the closed form expression of $LP(k, \boldsymbol{Y}_k)$ in not available except for small k. In this paper, two upper bounds for $LP(k, \boldsymbol{Y}_k)$ and closed form of $LP(2, \boldsymbol{Y}_2)$ are provided.

Theorem 2 *An upper bound for the optimal objective function value of* P_2 *is given by*

$$\max_{0\le z\le N_k} \frac{2^z}{(d_1+1)^{N_k-z-kd_1}(d_1!)^k(d_2+1)^{z-k(k-1)d_2/2}(d_2!)^{k(k-1)/2}}, \tag{19}$$

where $d_1 = [(N_k - z)/k]$, $d_2 = [2z/\{k(k-1)\}]$, $N_k = \frac{1}{2}\sum_{i=1}^{k} Y_{i,k}$, *and* $[x]$ *denotes the largest integer less than or equal to* x.

Proof. Fixing z and ignoring the constraints (18), we can easily show that the object function value

$$\frac{2^z}{\prod_{i\le j} x_{ij}!} = \frac{2^z}{\left(\prod_{i=1}^{k} x_{ii}!\right)\left(\prod_{i<j} x_{ij}!\right)} \tag{20}$$

is maximized when $|x_{ii} - x_{jj}| \le 1$ for all i, j and $|x_{ij} - x_{i'j'}| \le 1$ for all $i < j$, $i' < j'$. Therefore under the constraints $\sum_{i=1}^{k} x_{ii} = N_k - z$ and $\sum_{i<j} x_{ij} = z$, $N_k - z - kd_1$ elements in $\{x_{11}, ..., x_{kk}\}$ are equal to $d_1 + 1$ and the rest are equal to d_1, and $z - k(k-1)d_2/2$ elements in $\{x_{12}, ..., x_{k-1k}\}$ are equal to $d_2 + 1$ and the rest are equal to d_2. Substituting these values into (20) and maximizing with respect to z yields (19). Since (19) is the maximum objective function value for the relaxation problem of P_2 where the constraints (18) are ignored, it is indeed an upper bound for the optimal objective function value of P_2. Q.E.D.

We can see that the upper bound given in Theorem 2 is equal to the exact $LP(k, \boldsymbol{Y}_k)$ value when the components of $\boldsymbol{Y}_k$ is equal or nearly equal to each other. For this reason,

this upper bound is a natural analogue of an upper bound for LP given by Mehta and Patel (1983) for Freeman-Halton case.

Next we provide another (approximate) upper bound which has good property regardless of the pattern of $\boldsymbol{Y}_k$ in the following Theorem.

Theorem 3 *An approximate upper bound for the optimal objective function value of* P_2 *is given by*

$$\frac{2^{z^*}}{\prod_{i\leq j} g(x^*_{ij})}, \quad z^* = \sum_{i<j} x^*_{ij},$$

where

$$x^*_{ii} = \frac{Y^2_{i,k}}{4N_k}, \quad x^*_{ij} = \frac{Y_{i,k}Y_{j,k}}{2N_k}, \quad i \neq j, \tag{21}$$

and $g(x)$ *is an arbitrary continuous function satisfying* $g(n) = n!$ *if* n *is an integer.*

Proof. Replacing $x!$ with the function $g(x)$ defined above and ignoring the constraint that x_{ij} is integer, the continuous relaxation problem of P_2 is obtained as

$$\mathrm{P}_2' : \text{maximize } \frac{2^z}{\prod_{i\leq j} g(x_{ij})}, \quad z = \sum_{i<j} x_{ij},$$

subject to (18),(10) and $x_{ij} \geq 0$. Clearly the optimal objective function value of P_2' is an upper bound for the original integer optimizing problem P_2.

On the other hand, the optimal solution of P_2' is approximated by (21) for the following reason. Let $\boldsymbol{p}_1$ be the reference empirical distribution given by

$$p_{ij} = x_{ij}/N_k, \quad i = 1, ..., k, \ j = i, ..., k,$$

where x_{ij} satisfies (18) and $\boldsymbol{p}_0$ be the Hardy-Weinberg distribution given by

$$p_{ii} = p_i^2, \quad i = 1, ..., k,$$

$$p_{ij} = 2p_ip_j, \quad i = 1, ..., k-1, \ j = i+1, ..., k.$$

We denote the Kullback-Leibler divergence from $\boldsymbol{p}_1$ to $\boldsymbol{p}_0$ as $D(\boldsymbol{p}_1, \boldsymbol{p}_0)$. Since the optimal solution of P_2 corresponds to $\boldsymbol{p}_1$ whose occurrence probability is maximum when the true distribution is $\boldsymbol{p}_0$, P_2 is approximately equivalent to minimizing $D(\boldsymbol{p}_1, \boldsymbol{p}_0)$. Here the decomposition

$$D(\boldsymbol{p}_1, \boldsymbol{p}_0) = D(\boldsymbol{p}_1, \boldsymbol{p}_M) + D(\boldsymbol{p}_M, \boldsymbol{p}_0) \tag{22}$$

holds where $\boldsymbol{p}_M$ is the conditional maximum likelihood estimate under the Hardy-Weinberg model given by $p_{ij} = x_{ij}/N_k$ where x_{ij} is given by (21). This prove the theorem. Q.E.D.

The decomposition (22) is an important property of the divergence $D(\boldsymbol{p}_1, \boldsymbol{p}_0)$ and can be derived directly for the present case. The meaning of this decomposition is elucidated from the differential geometrical point of view. For detail, see Amari (1985, 1989) for example.

The standard example of $g(x)$ is Gamma function, $g(x) = \Gamma(x+1)$. However, even simpler function such as piecewise linear or piecewise quadratic function can also be used.

As the last result of this section, we provide the closed form expression of $LP(2, \boldsymbol{Y}_2)$. The problem that we consider is written as

$$\mathrm{P}_3 : \text{maximize } \frac{2^{x_{12}}}{x_{11}!x_{12}!x_{22}!}$$

subject to

$$2x_{11} + x_{12} = Y_{1,2}, \quad 2x_{22} + x_{12} = Y_{2,2},$$

$$x_{11}, x_{12}, x_{22} \text{ are non negative integers.}$$

Theorem 4 *The optimal solution $\boldsymbol{X}^* = (x_{11}^*, x_{12}^*, x_{22}^*)$ of P_3 is given as follows.*

1. *If $Y_{1,2}, Y_{2,2}$ are both even numbers, let $a(\boldsymbol{Y}_2) = (Y_{1,2}Y_{2,2} - 2)/\{2(Y_{1,2} + Y_{2,2} + 3)\}$. The optimal solution is*

$$x_{11}^* = \frac{Y_{1,2}}{2} - n, \quad x_{22}^* = \frac{Y_{2,2}}{2} - n, \quad x_{12}^* = n,$$

where

$$\begin{cases} n = a(\boldsymbol{Y}_2) \text{ or } a(\boldsymbol{Y}_2) + 1, & \text{if } a(\boldsymbol{Y}_2) \text{ is integer,} \\ n = \lfloor a(\boldsymbol{Y}_2) + 1 \rfloor, & \text{otherwise.} \end{cases}$$

2. *If $Y_{1,2}, Y_{2,2}$ are both odd numbers, let $a(\boldsymbol{Y}_2) = \{(Y_{1,2} - 1)(Y_{2,2} - 1) - 6\}/\{2(Y_{1,2} + Y_{2,2} + 3)\}$. The optimal solution is*

$$x_{11}^* = \frac{Y_{1,2} - 1}{2} - n, \quad x_{22}^* = \frac{Y_{2,2} - 1}{2} - n, \quad x_{12}^* = 2n + 1,$$

where

$$\begin{cases} n = a(\boldsymbol{Y}_2) \text{ or } a(\boldsymbol{Y}_2) + 1, & \text{if } a(\boldsymbol{Y}_2) \text{ is integer,} \\ n = \lfloor a(\boldsymbol{Y}_2) + 1 \rfloor, & \text{otherwise.} \end{cases}$$

The proof of this theorem is straightforward and omitted.

5. Some numerical examples

In this section exact p values were computed by the network algorithm for problems of various sizes. All the algorithms were programmed using C language on a PC running on Linux (Pentium III, 930MHz).

First we analyze the data of $r = 8, N = 30, \boldsymbol{y} = (15, 14, 11, 12, 2, 2, 1, 3)$, displayed in Figure 2. This data is taken from Figure 1 of Guo and Thompson (1992). Since the size of this data is moderately large, they could not calculate the exact p value for this data and instead evaluated the simulated value by Markov chain Monte Carlo method.

We computed the p value for this data by using a complete enumeration algorithm proposed by Louis and Dempster (1987) and the network algorithm. As for computing upper bounds for LP in the network algorithm, two upper bounds proposed in the previous section (Theorem 2 and Theorem 3) were considered. Table 1 shows the p values and CPU times.

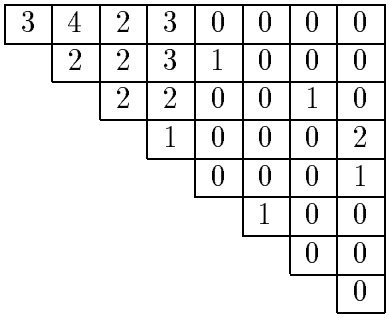

Figure 2. Genotype frequency data from Guo and Thompson (1992).

Table 1 shows that the proposed algorithm performs better than the algorithm by Louis and Dempster. The CPU times show that path was trimmed in Case 1 (in section 3) more efficiently when using the approximate upper bound attained at MLE proposed in Theorem 3. Strictly speaking, however, it is not guaranteed that the obtained p value is *exact* when the approximate upper bound is used. This is because the optimal solution of the relaxation problem P_2' is attained at (21) only approximately. Then an over trimming may occur when the optimal solution of the relaxation problem is badly underestimated than the true optimal solution of the original integer maximization problem. Indeed, the p value by Network (LP by Thm. 3) in table 1 is slightly larger than the other two values. But the differences are quite small and it can be considered that the accuracy of the approximation is quite good in practice.

Next we analyze data sets of various sizes. Table 2 shows the p values and CPU times for the examples of $r = 5$ that the pattern of $\boldsymbol{y}$ is uniform. Table 3 shows the p values and CPU times for the various pattern of $\boldsymbol{y}$ for examples of $N = 50$. In each example, the p value close to 0.05 is calculated. The number of all the tables ($\#\mathcal{F}$) and the ratio of CPU time (complete enumeration to network) are also provided when the complete enumeration is feasible.

Table 1. A comparison of the network and the Louis and Dempster algorithms for the allele frequency data in Figure 1 ($r = 8, \#\mathcal{F} = 250552020 \sim 2.5 \times 10^8$).

Algorithm	p value	CPU time[a]
Complete enumeration	0.2159398218	44:43
Network (LP by Thm. 2)	0.2159398218	10:25
Network (LP by Thm. 3)	0.2159433639	8:21

[a] CPU time is represented by min : sec.

Table 2. A comparison of the network and the Louis and Dempster algorithms for the allele frequency data of $r = 5$ (uniform case).

$\boldsymbol{y}$	Algorithm	p value	CPU time[a](ratio[b])	$\#\mathcal{F}$
$(20, 20, 20, 20, 20)$	Complete enumeration	0.0448476262	46:27	3.0×10^8
	Network (LP by Thm. 2)	0.0448476262	4:55 (9.45)	
	Network (LP by Thm. 3)	0.0448505876	4:03 (11.47)	
$(22, 22, 22, 22, 22)$	Complete enumeration	0.0443505782	106:56	7.0×10^8
	Network (LP by Thm. 2)	0.0443505782	9:06 (11.75)	
	Network (LP by Thm. 3)	0.0443514885	7:27 (14.35)	
$(24, 24, 24, 24, 24)$	Complete enumeration	0.0476068427	230:13	1.5×10^9
	Network (LP by Thm. 2)	0.0476068428	15:09 (15.20)	
	Network (LP by Thm. 3)	0.0476073528	12:21 (18.64)	
$(26, 26, 26, 26, 26)$	Complete enumeration		infeasible[c]	
	Network (LP by Thm. 2)	0.0490752414	27:42	
	Network (LP by Thm. 3)	0.0490747618	24:20	
$(28, 28, 28, 28, 28)$	Complete enumeration		infeasible[c]	
	Network (LP by Thm. 2)	0.0502934492	37:29	
	Network (LP by Thm. 3)	0.0502939082	30:38	
$(30, 30, 30, 30, 30)$	Complete enumeration		infeasible[c]	
	Network (LP by Thm. 2)	0.0516508563	55:49	
	Network (LP by Thm. 3)	0.0516511735	45:29	

[a] CPU time is represented by min : sec.
[b] CPU time (complete enumeration) / CPU time (network)
[c] Fail to compute p value within 360 CPU minutes.

Table 3. A comparison of the network and the Louis and Dempster algorithms for the allele frequency data of $N = 50, r = 4 \sim 8$.

$\boldsymbol{y}$	Algorithm	p value	CPU time[a](ratio[b])	$\#\mathcal{F}$
$(25, 25, 25, 25)$	Complete enumeration	0.0526117171	0:02.05	2.3×10^5
	Network (LP by Thm. 2)	0.0526117171	0:00.46 (4.46)	
	Network (LP by Thm. 3)	0.0526117171	0:00.40 (5.13)	
$(40, 30, 20, 5, 5)$	Complete enumeration	0.0566520911	0:18.28	2.0×10^6
	Network (LP by Thm. 2)	0.0566520911	0:04.25 (4.30)	
	Network (LP by Thm. 3)	0.0566520911	0:03.64 (5.02)	
$(30, 30, 30, 5, 5)$	Complete enumeration	0.0479355528	0:26.98	3.0×10^6
	Network (LP by Thm. 2)	0.0479355528	0:06.13 (4.40)	
	Network (LP by Thm. 3)	0.0479355528	0:05.48 (4.92)	
$(40, 30, 10, 10, 10)$	Complete enumeration	0.0682463011	1:41.53	1.1×10^7
	Network (LP by Thm. 2)	0.0682463011	0:17.50 (5.80)	
	Network (LP by Thm. 3)	0.0683323439	0:13.66 (7.43)	
$(20, 20, 20, 20, 20)$	Complete enumeration	0.0448476262	46:27	3.0×10^8
	Network (LP by Thm. 2)	0.0448476262	4:55 (9.45)	
	Network (LP by Thm. 3)	0.0448505876	4:03 (11.47)	
$(30, 30, 30, 4, 3, 3)$	Complete enumeration	0.0449065433	2:16	1.5×10^7
	Network (LP by Thm. 2)	0.0449065433	0:31 (4.39)	
	Network (LP by Thm. 3)	0.0449065433	0:28 (4.86)	
$(40, 30, 10, 10, 5, 5)$	Complete enumeration	0.0606964775	27:47	1.8×10^8
	Network (LP by Thm. 2)	0.0606964775	4:29 (6.20)	
	Network (LP by Thm. 3)	0.0607761595	3:28 (8.01)	
$(40, 20, 20, 8, 7, 5)$	Complete enumeration	0.0435034239	85:03	5.6×10^8
	Network (LP by Thm. 2)	0.0435027787	16:06 (5.28)	
	Network (LP by Thm. 3)	0.0435052514	12:12 (6.97)	
$(30, 30, 20, 8, 7, 5)$	Complete enumeration	0.0521534407	133:05	8.8×10^8
	Network (LP by Thm. 2)	0.0521534422	22:40 (5.87)	
	Network (LP by Thm. 3)	0.0521535686	19:16 (6.91)	
$(30, 20, 20, 20, 5, 5)$	Complete enumeration	0.0426073065	264:41	1.7×10^9
	Network (LP by Thm. 2)	0.0426073065	43:40 (6.06)	
	Network (LP by Thm. 3)	0.0426079323	35:39 (7.42)	

Table 3. (Continued.)

$\boldsymbol{y}$	Algorithm	p value	CPU time[a]	(ratio[b])	$\#\mathcal{F}$
$(40,30,20,3,3,2,2)$	Complete enumeration	0.0657281092	4:56		3.3×10^7
	Network (LP by Thm. 2)	0.0657281092	0:59	(5.02)	
	Network (LP by Thm. 3)	0.0657315171	0:49	(6.04)	
$(30,30,30,3,3,2,2)$	Complete enumeration	0.0640574757	7:17		4.8×10^7
	Network (LP by Thm. 2)	0.0640574757	1:22	(5.33)	
	Network (LP by Thm. 3)	0.0640574757	1:13	(5.99)	
$(40,30,10,10,5,3,2)$	Complete enumeration	0.0480403049	121:27		8.0×10^8
	Network (LP by Thm. 2)	0.0480403049	21:21	(5.69)	
	Network (LP by Thm. 3)	0.0480998952	16:48	(7.23)	
$(40,25,15,10,5,3,2)$	Complete enumeration	0.0493349444	228:40		1.5×10^9
	Network (LP by Thm. 2)	0.0493349444	40:43	(5.62)	
	Network (LP by Thm. 3)	0.0493396952	31:50	(7.18)	
$(40,30,20,2,2,2,2,2)$	Complete enumeration	0.0658297002	13:57		9.2×10^7
	Network (LP by Thm. 2)	0.0658297002	2:38	(5.30)	
	Network (LP by Thm. 3)	0.0658300956	2:12	(6.34)	
$(40,25,25,2,2,2,2,2)$	Complete enumeration	0.0531653738	15:37		1.0×10^8
	Network (LP by Thm. 2)	0.0531653738	3:09	(4.96)	
	Network (LP by Thm. 3)	0.0531653738	2:40	(5.86)	
$(40,30,18,4,2,2,2,2)$	Complete enumeration	0.0492180369	45:16		3.0×10^8
	Network (LP by Thm. 2)	0.0492180369	8:59	(5.04)	
	Network (LP by Thm. 3)	0.0492180505	7:30	(6.04)	
$(40,30,15,7,2,2,2,2)$	Complete enumeration	0.0422794862	114:12		7.6×10^8
	Network (LP by Thm. 2)	0.0422794862	22:58	(4.97)	
	Network (LP by Thm. 3)	0.0422816826	18:57	(6.03)	
$(40,30,15,5,4,2,2,2)$	Complete enumeration	0.0641293814	217:22		1.4×10^9
	Network (LP by Thm. 2)	0.0641293814	33:32	(6.48)	
	Network (LP by Thm. 3)	0.0641321353	26:34	(8.18)	

[a] CPU time is represented by min : sec.millisec

[b] CPU time (complete enumeration) / CPU time (network)

Table 2 and table 3 show that the network algorithm performs uniformly better for all these examples. CPU ratio shows that the efficiency of the network algorithm is more emphasized when the size of the problem is large. We see that the p values of examples of moderate size ($\#\mathcal{F} \sim 10^9$) can be calculated within about 30 minutes by the network algorithm, while it took several hours by the complete enumeration. Comparing the upper bound for LP, we see that the approximate upper bound proposed in Theorem 3 performs better and the accuracy of the approximation is satisfactory.

It should be noted that the CPU time is greatly effected by p value when using the network algorithm, while it takes same time regardless of p value by the complete enumeration. In this study only p values about 0.05 are mainly considered, however, larger p values can be more easily calculated by the network algorithm. Table 4 shows CPU times to calculate various p values for the case of $\boldsymbol{y} = (30, 30, 20, 8, 7, 5)$.

Table 4. A comparison of the network and the Louis and Dempster algorithms for the allele frequency data of $\boldsymbol{y} = (30, 30, 20, 8, 7, 5)$ for various p values.

	CPU time[a]		
p	Complete enumeration	Network (LP by Thm. 2)	Network (LP by Thm. 3)
0.9968	133:12	0:05.89	0:00.19
0.9101	133:09	0:18.56	0:05.82
0.8242	133:11	0:34.76	0:15.24
0.5933	133:09	1:45	1:07
0.4728	133:09	2:56	2:00
0.3178	133:09	5:23	4:02
0.2161	133:25	8:20	6:34
0.1070	133:31	14:58	12:22
0.0522	133:05	22:40	19:16

[a] CPU time is represented by min : sec.millisec

References

[1] Amari, S. (1985). *Differential-Geometrical Methods in Statistics*, Springer, Berlin.

[2] Amari, S. (1989). Geometry of Statistical Models. In *Analysis of Statistical Information: Preprints of the Symposium on the Analysis of Statistical Information.* pp. 131-137. The Institute of Statistical Mathematics, Tokyo, December 5-8, 1989.

[3] Crow, J. E. (1988). Eighty years ago: The beginnings of population genetics. *Genetics* **119**, 473-476.

[4] Elston, R. C. and Forthofer, R. (1977). Testing for Hardy-Weinberg equilibrium in small samples. *Biometrics* **33**, 536-542.

[5] Emigh, T. H. (1980). A comparison of tests for Hardy-Weinberg equilibrium. *Biometrics* **36**, 627-642.

[6] Emigh, T. H. and Kempthorne, O. (1975). A note on goodness-of-fit of a population to Hardy-Weinberg structure. *American Journal of Human Genetics* **27**, 778-783.

[7] Freeman, G. H., and Halton, J. H. (1951). Note on an exact treatment of contingency, goodness of fit and other problems of significance. *Biometrika* **38**, 141-149.

[8] Guo, S. W., and Thompson, E. A. (1992). Performing the exact test of Hardy-Weinberg proportion for multiple alleles. *Biometrics* **48**, 361-372.

[9] Hernández, J. L. and Weir, B. S. (1989). A disequilibrium coefficient approach to Hardy-Weinberg testing. *Biometrics* **45**, 53-70.

[10] Levene, H. (1949). On a matching problem arising in genetics. *Annals of Mathematical Statistics* **20**, 91-94.

[11] Louis, E. J., and Dempster, E. R. (1987). An exact test for Hardy-Weinberg and multiple alleles. *Biometrics* **43**, 805-811.

[12] Mehta, C. R., and Patel, N. R. (1983). A network algorithm for performing Fisher's exact test in $r \times c$ contingency tables. *Journal of the American Statistical Association* **78**, 427-434.

[13] Smith, C. A. B. (1986). Chi-squared tests with small numbers. *Annals of Human Genetics* **50**, 163-167.