Estimation of the Mutation Rate during Error-prone Polymerase Chain Reaction

*Dai Wang[1], Cheng Zhao[2], Rong Cheng[1] and Fengzhu Sun[1]

[1]Department of Genetics

Emory University School of Medicine

Atlanta, GA 30322 USA

Phone: (404)727-9830

Fax: (404)727-3949

E-mail: dwang@genetics.emory.edu

[2]Department of Mathematics and Computer Science

Indiana State University

Terre Haute, IN 47809 USA

Keywords: *in vitro* evolution, error-prone PCR, mutation rate, branching process, estimation.

Abstract

Error-prone polymerase chain reaction (PCR) is widely used to introduce point mutations during *in vitro* evolution experiments. Accurate estimation of the mutation rate during error-prone PCR is important in studying the diversity of error-prone PCR product. Although many methods for estimating the mutation rate during PCR are available, all the existing methods depend on the assumption that the mutation rate is low and mutations occur at different places whenever they occur. The available methods may not be applicable to estimate the mutation rate during error-prone PCR. We develop a mathematical model for error-prone PCR and present methods to estimate the mutation rate during error-prone PCR without assuming low mutation rate. We also develop a computer program to simulate error-prone PCR. Using the program, we compare the newly developed methods with two other methods. We show that when the mutation rate is relatively low($< 10^{-3}$ per base per PCR cycle), the newly developed methods give roughly the same results as previous methods. When the mutation rate is relatively high($> 5 \times 10^{-3}$ per base per PCR cycle, the mutation rate for most error-prone PCR experiments), the previous methods underestimate the mutation rate and the newly developed methods approximate the true mutation rate.

1. Introduction

In vitro evolution is a laboratory method to evolve molecules with desired properties of interest. It has been used to optimize industrial enzymes, to improve drug resistance, and to develop novel pharmaceuticals and vaccines (Arnold 1996, 1998, Patten *et al.* 1997). To implement an *in vitro* evolution experiment, we first build an initial molecule library. There are three steps in an *in vitro* evolution experiment. The first step is selection or screening from the molecule library. The selection methods depend on the molecule properties of interest. They can be based on the molecules' ligand binding properties, their catalytic properties, or other characteristics. The second step of *in vitro* evolution is mutagenesis. Error-prone polymerase chain reaction (PCR) (Leung *et al.* 1989, Cadwell and Joyce 1992) and/or *in vitro* recombination techniques such as DNA shuffling (Stemmer 1994ab, Zhao and Arnold 1997), staggered extension process (StEP) recombination (Zhao *et al.* 1998), and random-priming *in vitro* recombination (RPR) (Shao *et al.* 1998) are widely used as mutagenesis techniques. The third step is to amplify the resulting molecules to form a new molecular library. The three steps of selection, mutagenesis and amplification are repeated for many cycles until molecules with the desired function of interest dominate the final molecule library.

Error-prone PCR was first proposed by Leung *et al.* (1989) and later modified by Cadwell and Joyce (1992) to introduce random point mutations at each PCR cycle. Error-prone PCR is implemented by using DNA polymerase with low fidelity and by changing experimental conditions in standard PCR experiments. It has been

successfully applied to several *in vitro* evolution experiments (Bartel and Szostak 1993, Chen and Arnold 1993, You and Arnold 1996). In this manuscript, we study error-prone PCR theoretically.

Krawczak *et al.* (1989) and Hayshi studied the proportion of PCR products with no mutations after PCR experiments. Sun (1995) and Weiss and von Haeseler (1995) constructed a general model for PCR. The distribution of the number of mutations in a random sampled sequence and the distribution of the pair-wise differences in a random sample of sequences from the final PCR products were obtained. Moreover, a simple moment estimation method for estimating the mutation rate during PCR was proposed and the statistical properties of the estimators were studied(Sun 1995). Recently, Weiss and von Haeseler (1997) gave an algorithm to generate the genealogy that describes the relationship among a sample of PCR products. Using this algorithm, they presented a maximum likelihood method to estimate the mutation rate in PCR.

In all the above studies, it was assumed that mutations occur at different places each time they occur. In standard PCR, the mutation rate is usually low and this assumption is reasonable. In error-prone PCR, the mutation rate is increased. When the mutation rate is relatively high, the above assumption is questionable. In this paper, we study error-prone PCR without this assumption.

The organization of this paper is as follows. We first present a mathematical model for error-prone PCR. Then we present two methods to estimate the mutation rate during error-prone PCR. Then we study the statistical properties of the estimators and compare our methods with the maximum likelihood method of Weiss and von

Haeseler (1997) and the moment estimation method of Sun (1995) using simulations.

2. A mathematical model

The model of PCR with mutations is composed of two processes: 1) the process of generating the templates which gives a random binary tree, and 2) the processes of superimposing mutations onto the binary tree. The former process is a standard branching process. We assume that there are S_0 identical copies of single-stranded sequences. Let S_n be the number of sequences after n PCR cycles. In the n^{th} cycle, each of the S_{n-1} template sequences generates a new sequence with probability λ and itself always remains in the products. λ is referred to as the efficiency of PCR. S_0, S_1, S_2, ..., S_n, ... form a Galton-Watson process.

Now we add the mutation process to the binary tree. We assume that the probability a base is not mutated per PCR cycle is $\exp(-\mu)$. We also assume nucleotide bases are mutated independently. We add the assumption that when a mutation occurs at a nucleotide position, it changes to the other three nucleotides with equal probability 1/3. This assumption holds for the protocol of Cadwell and Joyce (1992) and does not hold for the protocol of Leung et al. (1989). In *in vitro* evolution experiments, the objective is to search the sequence space as evenly as possible and the above assumption should hold in ideal situations. If this assumption is violated, different probabilities can be given to different mutations. The following method can be easily adapted to such changes. For brevity of exposition, we use the above assumption in this paper. Table 1 summarizes the parameters used in the model.

< insert Table 1 here >

3. Estimating the mutation rate

As in Sun (1995), we call the original sequences the 0^{th} generation sequences. The sequences generated directly from the original sequences are called the first-generation sequences. Inductively the sequences generated directly from the k^{th} generation sequences are called the $(k+1)^{st}$ generation sequences.

Let X_k^n be the number of k^{th} generation sequences after n PCR cycles. It has been shown that the expected number of the k^{th} generation sequences after n PCR cycles is $\binom{n}{k}\lambda^k S_0$, $k \geq 0$, $n \geq 1$ (Sun, 1995). It has also been shown that when S_0 is sufficiently large, the distribution of the generation number, K, of a randomly chosen sequence can be approximated by a binomial distribution $B(n, \lambda/(1+\lambda))$. Here, again, we make the following assumption.

Assumption 1. The distribution of the generation number, K, of a randomly chosen sequence after n PCR cycles is $Binomial(n, \lambda/(1+\lambda))$.

3.1. Estimating the mutation rate when the nucleotide bases of the original sequences are known

We first study the number of mutations in a k^{th} generation sequence. Let us consider only one base. Without lose of generality, let us first fix a base of the target with nucleotide "A". Let $p(k)$ be the probability of the event, E, that the base is still "A" after k replications. Then E happens if and only if one of the following events happens: i) the nucleotide is not mutated in the first PCR replication with probability $\exp(-\mu)$, and the base is not changed in the next $k-1$ PCR replications

with probability $p(k-1)$; and ii) the nucleotide is mutated to another nucleotide in the first PCR replication with probability $1-\exp(-\mu)$, and the nucleotide is changed back to "A" in the next $k-1$ PCR replications with probability $(1-p(k-1))/3$. Thus, we have the following recursive equation,

$$p(k) = \exp(-\mu)p(k-1) + (1-\exp(-\mu))(1-p(k-1))/3, \qquad k = 1, 2, \ldots, n,$$

with initial condition $p(0) = 1$. From this equation, we obtain

$$p(k) = \frac{1}{4} + \frac{3}{4}\left(\frac{4}{3}\exp(-\mu) - \frac{1}{3}\right)^k.$$

Given a k^{th} generation sequence, the number of base changes in the sequence has binomial distribution $B(G, 1-p(k))$. We have the following results.

Theorem 1. *Let M be the number of mutations of a randomly chosen sequence and Assumption 1 holds. Then for $0 \le m \le G$,*

$$P\{M = m\} = \sum_{k=0}^{n} \binom{G}{m} (1-p(k))^m \, p(k)^{G-m} \frac{\binom{n}{k}\lambda^k}{(1+\lambda)^n}$$

and

$$EM = \frac{3}{4}G\left(1 - \frac{(1+\lambda a)^n}{(1+\lambda)^n}\right),$$

where $a = \frac{4}{3}\exp(-\mu) - \frac{1}{3}$.

From the above theorem, we conclude that the expected number of base changes in a randomly chosen sequence is $3G\left(1 - \frac{(1+\lambda a)^n}{(1+\lambda)^n}\right)/4$. In order to estimate the mutation rate in an error-prone PCR experiment, we can choose a sample of s sequences from the final PCR products and count the number of base changes, M_i, of

the i^{th} sampled sequence. $1-(1+\lambda a)^n/(1+\lambda)^n$ can be estimated by $4\sum_{i=1}^{s} M_i/(3sG)$. Then we can estimate $1-\exp(-\mu)$, the mutation rate per base per PCR cycle, by $f\left(\sum_{i=1}^{s} M_i\right)$. Here,

$$f(x)=\frac{3}{4\lambda}\left(\lambda-(1+\lambda)\sqrt[n]{1-\frac{4x}{3sG}}+1\right).$$

Next we consider the statistical properties of this estimator. The following results give the limit behaviour of $E\left(\sum_{i=1}^{s} M_i\right)$ and $Var\left(\sum_{i=1}^{s} M_i\right)$. These results do not depend on Assumption 1. For simplicity, we approximate the distribution of the number of base changes of a k^{th} generation sequence by a Poisson random variable with mean $G(1-p(k))$. The following theorem is proved in the appendix.

Theorem 2. *Let S_0 be the number of initial sequences. Then for $0<\lambda\leq 1$,*

i)

$$\lim_{S_0\to\infty} S_0\left\{E\left(\sum_{i=1}^{s} M_i\right)-\frac{3}{4}sG\left(1-\frac{(1+a\lambda)^n}{(1+\lambda)^n}\right)\right\}$$
$$=s(1-a)(1-\lambda)((1+\lambda)^n-1)\frac{(1+a\lambda)^{n-1}}{(1+\lambda)^{2n+1}};$$

ii)

$$\limsup_{S_0\to\infty} S_0\left\{Var\left(\sum_{i=1}^{s} M_i\right)-sV\right\}\leq sA+2\binom{s}{2}B,$$

where

$$V=\frac{9}{16}G^2\left(\frac{(1+a^2\lambda)^n}{(1+\lambda)^n}-\frac{(1+a\lambda)^{2n}}{(1+\lambda)^{2n}}\right)+\frac{3}{4}G\left(1-\frac{(1+a\lambda)^n}{(1+\lambda)^n}\right)$$

$$A=\frac{(1-\lambda)(1-a)((1+\lambda)^n-1)}{(1+\lambda)^{3n+1}}\left(\frac{9}{16}G^2(1+a^2\lambda)^{n-1}(1+\lambda)^n(1+a)\right.$$
$$\left.-\frac{9}{8}G^2(1+a\lambda)^{2n-1}-\frac{3}{4}G(1+a\lambda)^{n-1}(1+\lambda)^n\right),$$

$$B = \frac{3G}{4(1+\lambda)^{2n}}\left(2\frac{(1+\lambda)^n-(1+\lambda)^{2n}}{(1+\lambda)-(1+\lambda)^2}-2\frac{(1+a\lambda)^n-(1+\lambda)^{2n}}{(1+a\lambda)-(1+\lambda)^2}\right.$$
$$\left.+(1+\lambda)^n-(1+a\lambda)^n\right)$$
$$+\frac{9}{16}G^2\left(\frac{a(1-\lambda)\left((1+\lambda a^2)^n-(1+a\lambda)^{2n}\right)}{(a-2-a\lambda)(1+\lambda)^{2n}}\right.$$
$$\left.+\frac{(1-\lambda)(1-2a-a\lambda)(1+a\lambda)^{2n}((1+\lambda)^n-1)}{(1+a\lambda)(1+\lambda)^{3n+1}}\right).$$

Note 1. Theorem 2(i) shows how fast $E\left(\sum_{i=1}^{s} M_i\right)$ approaches its limit $\frac{3}{4}sG\left(1-\frac{(1+a\lambda)^n}{(1+\lambda)^n}\right)$. Theorem 2(ii) gives an upper bound for the variance of $\sum_{i=1}^{s} M_i$.

Note 2. From Theorem 2, we can get an approximate upper bound of the standard deviation of the estimator. Let $x_0 = \frac{3}{4}sG\left(1-\frac{(1+a\lambda)^n}{(1+\lambda)^n}\right)$. Then $1-\exp(-\mu) = f(x_0)$. When S_0 is sufficient large,

$$\begin{aligned} E\left(f\left(\sum_{i=1}^{s} M_i\right)-f(x_0)\right)^2 &\approx f'(x_0)^2 E\left(\sum_{i=1}^{s} M_i - x_0\right)^2 \\ &\approx f'(x_0)^2 Var\left(\sum_{i=1}^{s} M_i\right) \\ &\leq f'(x_0)^2 sV. \end{aligned}$$

3.2. Estimating the mutation rate when the nucleotide bases of the original sequences are not known

Sometimes we do not know the exact nucleotide sequence of the target and thus it is impossible to obtain the number of mutations in a randomly chosen sequence. In this case, we study the number of base differences between two randomly chosen sequences.

We use the distance between two sequences defined in Sun (1995) and Weiss and von Haeseler (1995). For any two sequences α and β, if γ is a common ancestor of the two sequences and there is no other common ancestor before, then γ is called the

most recent common ancestor (MRCA) of α and β. The pair-wise distance between α and β is defined by

$$d(\alpha, \beta) = (g(\alpha) - g(\gamma)) + (g(\beta) - g(\gamma)),$$

where $g(\cdot)$ is the generation number of the sequence. This distance counts the number of PCR replications that occurred between sequence α and sequence β. The approximate distribution for the pair-wise distance, D, between two randomly chosen sequences was given in Sun (1995). In particular, the probability generating function of D was given by

$$\begin{aligned}\varphi(x) &= \sum_{d=0}^{2n} x^d P\{D = d\} \\ &= \frac{S_0 \lambda x \frac{(1+\lambda x)^{2n} - (1+\lambda)^n}{(1+\lambda x)^2 - (1+\lambda)} + \binom{S_0}{2}(1+\lambda x)^{2n}}{S_0(1+\lambda)^{n-1}\left((1+\lambda)^n - 1\right) + \binom{S_0}{2}(1+\lambda)^{2n}}.\end{aligned}$$

Let α and β be two randomly chosen sequences from the PCR products. γ is their MRCA and the distance between α and β is d. Then $d = d_1 + d_2$, where d_1 is the distance between α and γ and d_2 is the distance between β and γ. We study the difference between α and β base by base. Fix a base position, the nucleotide bases of α and β at this position are the same if and only if the following events happen: i) the nucleotide bases of α, β and γ at this position are identical; and ii) the nucleotide base of γ is different from the identical nucleotide bases of α and β at this position. The probability of the first event is $p(d_1)p(d_2)$ and the probability of the second event is $3 \cdot \frac{1}{3}(1 - p(d_1)) \cdot \frac{1}{3}(1 - p(d_2))$. So we can conclude that the probability that a base

is the same between α and β is $p(d) = \frac{1}{4} + \frac{3}{4}a^d$, where $a = \frac{3}{4}\exp(-\mu) - \frac{1}{3}$. The number of base differences, H, between the two sequences has binomial distribution $B(G, 1-p(d))$. We call H the Hamming distance between the two sequences. Given $D = d$, the average Hamming distances between two randomly chosen sequences is $G(1-p(d))$. Summing over all the possible values of D from 0 to $2n$, we have the average Hamming distance given by

$$\begin{aligned} EH &= \sum_{d=0}^{2n} G(1-p(d))P\{D=d\} \\ &= \frac{3}{4}G\sum_{d=0}^{2n}(1-a^d)P\{D=d\} \\ &= \frac{3}{4}G(1-\varphi(a)). \end{aligned}$$

In fact, we have the following result.

Theorem 3. *Let H be the Hamming distance between two randomly chosen sequences after n PCR cycles. Then*

$$EH = \frac{3}{4}G\left(1 - \left(\frac{1+a\lambda}{1+\lambda}\right)^{2n}\left(1 + \frac{2\left(\frac{a\lambda(1+a\lambda)}{(1+a\lambda)^2-(1+\lambda)} - 1\right)}{S_0(1+\lambda) + (1-\lambda) - \frac{2}{(1+\lambda)^n}} + O\left(\frac{1}{S_0(1+\lambda)^n}\right)\right)\right).$$

For a sample of s sequences, we can first calculate the Hamming distance between sequence i and sequence j denoted by H_{ij}. The observed average Hamming distance is given by $\overline{H} = \sum_{i,j=1,i\neq j}^{s} H_{ij}/s(s-1)$. Let the theoretical average Hamming distance equals to its observed value and solve the equation for a. We have another estimator of the mutation rate.

4. Comparing the accuracies of the estimation methods

Several methods are available to estimate the mutation rate during PCR. Under the assumption that the mutation rate is low, Sun (1995) proposed a method of moment estimation and Weiss and von Haeseler (1997) proposed a maximum likelihood estimation(MLE) for the mutation rate. These methods might be used to estimate the mutation rate during error-prone PCR. In this paper, we propose a method based on the number of base changes and a method based on the number of pair-wise differences among a sample of sequences without the assumption of low mutation rate. Yet it is not clear which methods would perform better. In this section we compare the four estimating methods using simulation.

4.1. A computer program to simulate error-prone PCR

We modified the computer program of Weiss and von Haeseler (1997) to simulate error-prone PCR. We used the first three steps of their algorithm to generate the genealogy of a set of sequences sampled from a PCR experiment. The mutation process we are studying here is different from that of Weiss and von Haeseler (1997) and their program has to be modified.

In their algorithm, the number of sequences generated from each 0^{th} generation sequence after each PCR cycle is computed first. Then they randomly assigned one of the initial sequences to each of the sampled sequences as an ancestor. They took the sets of sampled sequences that are descendents of the same initial sequence as subsamples. In the second step, the genealogies of all subsamples were traced back separately. In this step, for each initial sequence j, $j = 1, \ldots, S_0$, that has at least one

descendent in the sample, the following numbers were generated for each cycle: i) $N_{i,j}$, the number of sequences in the genealogy of the subsample present after cycle i; ii) $R_{i,j}$, the number of sequences among the $N_{i,j}$ sequences that were newly synthesized in cycle i; iii) $L_{i,j}$, the number of coalescent events in cycle i. A coalescent event happens in a cycle if a template sequence and its direct descendent are both present in the genealogy of the subsample in this cycle.

In our model, we assume that the nucleotide bases along the templates are mutated independently. And when a mutation occurs at a nucleotide position, it changes to the other three nucleotides with equal probability 1/3. For a sequence, we can put it through a replication procedure and obtain a newly synthesized sequence. Now given the genealogy of a subsample of $N_{n,j}$ sequences, we go forward to obtain the nucleotide sequences of the $N_{n,j}$ sequences.

a) There are $N_{i-1,j}$ sequences in the genealogy of the subsample present after the $(i-1)^{st}$ cycle. We randomly choose $N_{i,j} - R_{i,j} - L_{i,j}$ sequences from the $N_{i-1,j}$ sequences. These sequences are still in the genealogy of the subsample after the i^{th} cycle. We move them into the pool of sequences after the i^{th} cycle.

b) Randomly choose $L_{i,j}$ sequences from the remaining sequences in the genealogy of the subsample after the $(i-1)^{st}$ cycle. First, we move them into the pool of sequences after the i^{th} cycle. Then we put them through the replication procedure and record the newly synthesized sequences in the pool of sequences after the i^{th} cycle.

c) For the other sequences left in the genealogy of the subsample after the $(i-1)^{st}$ cycle, we put them through the replication procedure and record the newly syn-

thesized sequences in the pool of sequences after the i^{th} cycle. By combining the nucleotide sequences of each subsample, we get the nucleotide sequences of the sample. Then we count the number of base changes of a sequence and the base differences between two sequences.

4.2. Simulation results

In Cadwell and Joyce (1992), the estimated mutation rate in error-prone PCR is roughly 0.7%. In the following simulations, we use mutation rate of 0.1%, 0.5% and 1% respectively. We compare the accuracies of the four estimators: 1) moment estimation, 2) MLE, 3) the estimator based on the number of base changes, and 4) the estimator based on the pair-wise differences. Throughout the simulations, we use $\lambda = 0.8$, $n = 30$, $G = 500$ and $s = 30$. For different values of $S_0 = 1$, 10, 100 and 1000, we do 1000 simulations. Each simulation gives the estimations of the mutation rate using all the four methods. Table 2(a,b,c) show the results of the comparison of the four methods. In the table, "mean" is the average values of the estimations and "standard deviation" is the standard error with respect to the real mutation rate.

< insert Table 2. here >

In figure 1(a,b,c,d), we also show the histograms of the estimations of the mutation rate obtained by the four methods. Here, the number of initial sequences we use is 10 and the true mutation rate is 1%.

< insert Figure 1. here >

From Table 2(a), we see that when the mutation rate is relatively low(say less than 0.001), the performances of the four methods are roughly the same and the mean

estimated mutation rate tends to the true mutation rate as the number of initial sequences tends to infinity. Consistent with the results of Weiss and von Haeseler (1997), we also observe that the MLE tends to decrease to the true mutation rate as the number of initial sequences tends to infinity. Surprisingly, when the number of initial sequences is less than 10, MLE does not perform as well as the other three methods.

From Table 2(b,c), we see that when the mutation rate is relatively high, the method of moment estimation and the method of MLE will underestimate the mutation rate as expected. When $1 - \exp(-\mu) = 10^{-2}$ and $S_0 = 100$ or 1000, the moment estimator and the MLE are around 9.1×10^{-3}. The two new estimators proposed in this paper are 10% higher and very close to the true mutation rate. The standard deviations of the two new estimators are about half of the standard deviation of the moment estimator and the MLE. When $1 - \exp(-\mu) = 10^{-2}$, we can obtain an approximate upper bound 4.49×10^{-4} for the standard deviation according to Note 2 of Theorem 2. From the table, when $S_0 = 10$ the standard deviation obtained by simulation is 4.46×10^{-4}, which is smaller than the approximate upper bound. From Figure 1(c,d), we see that the new estimators center around the true mutation rate.

In the above simulations, we assume that the efficiency of PCR is a constant. In practice, the efficiency may depend on the number of PCR cycles. To see the effect of constant efficiency assumption, we run simulations with efficiency that varies as a function of PCR cycles. As in Weiss and von Haeseler (1997), we determine cycle

specific efficiencies from a published data (Saiki et al. 1988):

$$\lambda_i = \begin{cases} 0.872 & \text{if } i = 1, \ldots, 20, \\ 0.743 & \text{if } i = 21, \ldots, 25, \\ 0.146 & \text{if } i = 26, \ldots, 30. \end{cases}$$

In order to use our new methods, we use the average efficiency in our formula to give the estimations.

< insert Table 3. here >

From Table 3, we see that when the efficiency varies as a function of PCR cycles, the results of the newly developed estimation methods are not as good as in the constant efficiency case. When the number of initial sequences $S_0 = 1000$, the estimations obtained by using MLE method and the two methods developed in this paper are roughly the same. It shows that our two methods are reasonable when the efficiency varies as a function of PCR cycles even though the efficiency is assumed to be a constant.

When we use the MLE method to give the estimation, we still assume that mutations occur at different positions each time they occur in the above simulations. Thus when the real mutation rate is relatively high, this method will underestimate the real mutation rate. Of course, we can simulate PCR with efficiency varies as a function of PCR cycles and obtain MLE using our present assumption and then we can obtain another two MLE methods corresponding to the two methods developed in this paper, respectively. We believe that these two MLE methods are better than the two methods developed in this paper. However, for mutation rate of 5×10^{-3}, it takes

months to obtain the distribution of MLEs for 1000 runs. Thus such a comparison is not presented here.

Discussion

We develop a mathematical model for error-prone PCR and present two new methods to estimate the mutation rate during error-prone PCR. According to our model, we also develop a computer program to simulate error-prone PCR and to study the statistical properties of the estimators. In theory, our methods are good when the number of initial sequences, S_0, is large. The simulations show that these methods are also good when S_0 is small(for example $S_0 = 10$). Even for $S_0 = 1$, the estimation results are reasonable. Thus the estimators can be generally applicable to estimate the mutation rate during error-prone PCR.

Using computer simulations, we compare the newly developed methods with the moment estimation method of Sun (1995) and the MLE method of Weiss and von Haeseler (1997). It was shown that when the mutation rate is relatively low, say less than 10^{-3} per base per PCR cycle, the four methods gave roughly the same results. When the number of initial sequences is small(≤ 10), MLE does not perform as well as the other three methods. When the mutation rate is relatively high, such as greater than 5×10^{-3} per base per PCR cycle, the moment method and the MLE method underestimate the mutation rate, while the two methods developed in this paper approximate the true mutation rate.

In our model, we assume that the PCR efficiency λ is a constant during the PCR reaction. In real PCR experiments, the PCR efficiency may be lower in later PCR cycles than the efficiency in earlier PCR cycles. Our model does not apply in this situation. We might use the average efficiency over all the PCR cycles as the

PCR efficiency and then use our methods to estimate the mutation rate. As another approach, the modified simulation program developed in this paper can also be used to obtain the MLE of the mutation rate.

References

Arnold, F.H. 1996. Directed evolution: Creating biocatalysts for the future. *Chem. Eng. Sci. 51*, 5091–5102.

Arnold, F.H. 1998. Design by directed evolution. *Acc. Chem. Res. 31*, 125–131.

Bartel, D.P. and Szostak, J.W. 1993. Isolation of new ribozymes from a large pool of random sequences. *Science 261*, 1411–1418.

Cadwell, R.C. and Joyce, G.F. 1992. Randomization of genes by PCR mutagenesis. *PCR Method Applic 2*, 28–33.

Chen, K. and Arnold, F.H. 1993. Turning the activity of an enzyme for unusual environments: sequential random mutagenesis of sublitisin E for catalysis in dimethylformamide. *Proc. Natl. Acad. Sci. USA 90*, 5618–5622.

Hayashi, K. 1990. Mutations induced during the polymerase chain reaction. *Technique 2*, 216–217.

Krawczak, M., Reiss, J., Schmidtke, J. and Rosler, U. 1989. Polymerase chain reaction: replication errors and reliability of gene diagnosis. *Nucleic Acid Research 17*, 2197–2201.

Leung, D.W., Chen, E. and Goeddel, D.V. 1989. A method for random mutagenesis of a defined DNA segment using a modified polymerase chain reaction. *Technique 1*, 11–15.

Patten, P.A., Howard, R.J. and Stemmer WPC 1997. Applications of DNA shuffling to pharmaceuticals and vaccines. *Current Opinion in Biotechnology 8*, 724–733.

Saiki, R.K., Gelfand, D.H., Stoffel, S., Scharf, S.J., Higuchi, R., Horn, G.T.,

Mullis, K.B. and Erlich, H.A. 1988. *Science 239*, 487–491.

Shao, Z., Zhao, H., Giver, L. and Arnold, F.H. 1998. Random-priming *in vitro* recombination: an effective tool for directed evolution. *Nucleic Acid Res. 26*, 681–683.

Stemmer WPC 1994a. DNA shuffling by random fragmentation and reassembly: *in vitro* recombination for molecular evolution. *Proc. Natl. Acad. Sci. USA 91*, 10747–10751.

Stemmer WPC 1994b. Rapid evolution of a protein *in vitro* by DNA shuffling. *Nature 370*, 389–391.

Sun, F. 1995. The polymerase chain reaction and branching processes. *J. Computational Biology 2*, 63–86.

Weiss, G. and von Haeseler, A. 1995. Modeling the polymerase chain reaction. *J. Computational Biology 2*, 49–61.

Weiss, G. and von Haeseler, A. 1997. A coalescent approach to the polymerase chain reaction. *Nucleic Acid Res. 25*, 3082–3087.

You, L. and Arnold, F.H. 1996. Directed evolution of subtilisin E in *Bacillus Subtilis* to enhance total activity in Aqueous Dimethylformamide. *Protein Engineering 9*, 77–83.

Zhao, H. and Arnold, F.H. 1997. Optimization of DNA shuffling for high fidelity recombination. *Nucleic Acid Res. 26*, 1307–1308.

Zhao, H., Giver, L., Shao, Z., Affholter, J.A. and Arnold, F.H. 1998. Molecular evolution by staggered extension process(StEP) *in vitro* recombination. *Nature Biotechnology 16*, 258–261.

Appendix: Mathematical proofs

In this section, we prove Theorem 2. We separate the proof of Theorem 2 into several lemmas. First we have

$$Var\left(\sum_{i=1}^{s} M_i\right) = \sum_{i=1}^{s} Var\left(M_i\right) + 2\sum_{1\le i<j\le s} Cov(M_i, M_j)$$

$$= sVar\left(M_1\right) + 2\binom{s}{2} Cov(M_1, M_2).$$

We study $Cov\left(M_i, M_j\right)$ first. Let α and β be a pair of randomly chosen sequences. γ is the most recent common ancestor(MRCA) of α and β. $g(\cdot)$ and $M(\cdot)$ are the generation number and number of base changes of the corresponding sequences.

Lemma 1. *For any pair of sequences α and β after n PCR cycles, let γ be their MRCA, $g(\cdot)$ be the generation number, and $M(\cdot)$ be the number of base changes. Then*

$$\begin{aligned} & Cov\left(M(\alpha), M(\beta)\right) \\ = & \frac{3}{4}G\left(Ea^{g(\alpha)+g(\beta)-2g(\gamma)} - Ea^{g(\alpha)+g(\beta)-g(\gamma)}\right) + \frac{9}{16}G^2 Cov\left(a^{g(\alpha)}, a^{g(\beta)}\right) \\ \le & \frac{3}{4}G\left(1 - Ea^{g(\gamma)}\right) + \frac{9}{16}G^2 Cov\left(a^{g(\alpha)}, a^{g(\beta)}\right). \end{aligned}$$

Proof. Let α and β be two randomly chosen sequences and γ be their MRCA. Let $g(\cdot)$ and $M(\cdot)$ be the corresponding generation number and the number of base changes of the sequence, respectively. Since γ is an ancestor of α, we have

$$M(\alpha) = M(\gamma) + M^{+}_{\alpha,\gamma} - \sum_{i=1}^{M^{-}_{\alpha,\gamma}} X_{\alpha,\gamma}(i).$$

In the above equation $M^{+}_{\alpha,\gamma}$ is the number of bases unchanged in γ but changed in α. $M^{-}_{\alpha,\gamma}$ is the number of bases changed in γ and changed again from γ to α. $X_{\alpha,\gamma}(i) = 1$

if the i^{th} base of the $M^-_{\alpha,\gamma}$ bases in γ changes back to the nucleotide base in α and $X_{\alpha,\gamma}(i) = 0$ otherwise. Similarly,

$$M(\beta) = M(\gamma) + M^+_{\beta,\gamma} - \sum_{i=1}^{M^-_{\beta,\gamma}} X_{\beta,\gamma}(i).$$

Here $M^+_{\beta,\gamma}$ is the number of bases unchanged in γ but changed in β. $M^-_{\beta,\gamma}$ is the number of bases changed in γ and changed again from γ to β. $X_{\beta,\gamma}(i) = 1$ if the i^{th} base of the $M^-_{\beta,\gamma}$ bases in γ changes back to the nucleotide base in β and $X_{\beta,\gamma}(i) = 0$ otherwise. Given $g(\alpha)$, $g(\beta)$, $g(\gamma)$ and $M(\gamma)$, $\left(M^+_{\alpha,\gamma}, M^-_{\alpha,\gamma}\right)$, $\left(M^+_{\beta,\gamma}, M^-_{\beta,\gamma}\right)$, $X_{\alpha,\gamma}(i)$, $i = 1, 2, \ldots$ and $X_{\beta,\gamma}(i)$, $i = 1, 2, \ldots$ are independent and

$$\begin{aligned}
&M^+_{\alpha,\gamma} \sim Binomial\left(G - M(\gamma), 1 - p\Big(g(\alpha) - g(\gamma)\Big)\right);\\
&M^+_{\beta,\gamma} \sim Binomial\left(G - M(\gamma), 1 - p\Big(g(\beta) - g(\gamma)\Big)\right);\\
&M^-_{\alpha,\gamma} \sim Binomial\left(M(\gamma), 1 - p\Big(g(\alpha) - g(\gamma)\Big)\right);\\
&M^-_{\beta,\gamma} \sim Binomial\left(M(\gamma), 1 - p\Big(g(\beta) - g(\gamma)\Big)\right);\\
&P\left\{X_{\alpha,\gamma}(i) = 1\right\} = \frac{1}{3}, \qquad i = 1, 2, \ldots;\\
&P\left\{X_{\beta,\gamma}(i) = 1\right\} = \frac{1}{3}, \qquad i = 1, 2, \ldots.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&E\left(M(\alpha)M(\beta)\right)\\
&= E\left[\left(M(\gamma) + M^+_{\alpha,\gamma} - \sum_{i=1}^{M^-_{\alpha,\gamma}} X_{\alpha,\gamma}(i)\right)\left(M(\gamma) + M^+_{\beta,\gamma} - \sum_{i=1}^{M^-_{\beta,\gamma}} X_{\beta,\gamma}(i)\right)\right]\\
&= E\left[E\left[\left(M(\gamma) + M^+_{\alpha,\gamma} - \sum_{i=1}^{M^-_{\alpha,\gamma}} X_{\alpha,\gamma}(i)\right)\right.\right.\\
&\qquad\qquad \left.\left.\times\left(M(\gamma) + M^+_{\beta,\gamma} - \sum_{i=1}^{M^-_{\beta,\gamma}} X_{\beta,\gamma}(i)\right) \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right]\right]
\end{aligned}$$

$$
\begin{aligned}
&= E\Big[M(\gamma)^2 + M(\gamma)E\left[M^+_{\alpha,\gamma} \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right] \\
&\quad + M(\gamma)E\left[M^+_{\beta,\gamma} \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right] \\
&\quad - \frac{1}{3}M(\gamma)E\left[M^-_{\alpha,\gamma} \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right] \\
&\quad - \frac{1}{3}M(\gamma)E\left[M^-_{\beta,\gamma} \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right] \\
&\quad + E\left[M^+_{\alpha,\gamma} \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right] E\left[M^+_{\beta,\gamma} \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right] \\
&\quad - \frac{1}{3}E\left[M^-_{\alpha,\gamma} \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right] E\left[M^+_{\beta,\gamma} \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right] \\
&\quad - \frac{1}{3}E\left[M^+_{\alpha,\gamma} \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right] E\left[M^-_{\beta,\gamma} \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right] \\
&\quad + \frac{1}{9}E\left[M^-_{\alpha,\gamma} \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right] E\left[M^-_{\beta,\gamma} \mid g(\alpha), g(\beta), g(\gamma), M(\gamma)\right]\Big] \\
&= E\Big[M(\gamma)^2 + M(\gamma)\big(G - M(\gamma)\big)\big(1 - p\big(g(\alpha) - g(\gamma)\big)\big) \\
&\quad + M(\gamma)\big(G - M(\gamma)\big)\big(1 - p\big(g(\beta) - g(\gamma)\big)\big) \\
&\quad - \frac{1}{3}M(\gamma)^2\big(1 - p\big(g(\alpha) - g(\gamma)\big)\big) \\
&\quad - \frac{1}{3}M(\gamma)^2\big(1 - p\big(g(\beta) - g(\gamma)\big)\big) \\
&\quad + \big(G - M(\gamma)\big)^2\big(1 - p\big(g(\alpha) - g(\gamma)\big)\big)\big(1 - p\big(g(\beta) - g(\gamma)\big)\big) \\
&\quad - \frac{2}{3}M(\gamma)\big(G - M(\gamma)\big)\big(1 - p\big(g(\alpha) - g(\gamma)\big)\big)\big(1 - p\big(g(\beta) - g(\gamma)\big)\big) \\
&\quad + \frac{1}{9}M(\gamma)^2\big(1 - p\big(g(\alpha) - g(\gamma)\big)\big)\big(1 - p\big(g(\beta) - g(\gamma)\big)\big)\Big] \\
&= E\Big[M(\gamma)^2\Big(1 - \frac{4}{3}\big(1 - p\big(g(\alpha) - g(\gamma)\big)\big) - \frac{4}{3}\big(1 - p\big(g(\beta) - g(\gamma)\big)\big) \\
&\quad + \frac{16}{9}\big(1 - p\big(g(\alpha) - g(\gamma)\big)\big)\big(1 - p\big(g(\beta) - g(\gamma)\big)\big)\Big) \\
&\quad + GM(\gamma)\Big(\big(1 - p\big(g(\alpha) - g(\gamma)\big)\big) + \big(1 - p\big(g(\beta) - g(\gamma)\big)\big) \\
&\quad - \frac{8}{3}\big(1 - p\big(g(\alpha) - g(\gamma)\big)\big)\big(1 - p\big(g(\beta) - g(\gamma)\big)\big)\Big) \\
&\quad + G^2\big(1 - p\big(g(\alpha) - g(\gamma)\big)\big)\big(1 - p\big(g(\beta) - g(\gamma)\big)\big)\Big]
\end{aligned}
$$

$$
\begin{aligned}
&= E\Big[E\left[M(\gamma)^2 \mid g(\alpha), g(\beta), g(\gamma)\right]\Big(1-\frac{4}{3}\Big(1-p\Big(g(\alpha)-g(\gamma)\Big)\Big)\\
&\qquad -\frac{4}{3}\Big(1-p\Big(g(\beta)-g(\gamma)\Big)\Big)+\frac{16}{9}\Big(1-p\Big(g(\alpha)-g(\gamma)\Big)\Big)\Big(1-p\Big(g(\beta)-g(\gamma)\Big)\Big)\\
&\qquad +GE\left[M(\gamma) \mid g(\alpha), g(\beta), g(\gamma)\right]\Big(\Big(1-p\Big(g(\alpha)-g(\gamma)\Big)\Big)+\Big(1-p\Big(g(\beta)-g(\gamma)\Big)\Big)\\
&\qquad -\frac{8}{3}\Big(1-p\Big(g(\alpha)-g(\gamma)\Big)\Big)\Big(1-p\Big(g(\beta)-g(\gamma)\Big)\Big)\Big)\\
&\qquad +G^2\Big(1-p\Big(g(\alpha)-g(\gamma)\Big)\Big)\Big(1-p\Big(g(\beta)-g(\gamma)\Big)\Big)\Big]\\
&= E\Big[\Big(G^2\Big(1-p\Big(g(\gamma)\Big)\Big)^2+G\Big(1-p\Big(g(\gamma)\Big)\Big)\Big)\Big(1-\frac{4}{3}\Big(1-p\Big(g(\alpha)-g(\gamma)\Big)\Big)\\
&\qquad -\frac{4}{3}\Big(1-p\Big(g(\beta)-g(\gamma)\Big)\Big)+\frac{16}{9}\Big(1-p\Big(g(\alpha)-g(\gamma)\Big)\Big)\Big(1-p\Big(g(\beta)-g(\gamma)\Big)\Big)\Big)\\
&\qquad +G^2\Big(1-p\Big(g(\gamma)\Big)\Big)\Big(\Big(1-p\Big(g(\alpha)-g(\gamma)\Big)\Big)+\Big(1-p\Big(g(\beta)-g(\gamma)\Big)\Big)\\
&\qquad -\frac{8}{3}\Big(1-p\Big(g(\alpha)-g(\gamma)\Big)\Big)\Big(1-p\Big(g(\beta)-g(\gamma)\Big)\Big)\Big)\\
&\qquad +G^2\Big(1-p\Big(g(\alpha)-g(\gamma)\Big)\Big)\Big(1-p\Big(g(\beta)-g(\gamma)\Big)\Big)\Big]\\
&= \frac{3}{4}G\left(Ea^{g(\alpha)+g(\beta)-2g(\gamma)}-Ea^{g(\alpha)+g(\beta)-g(\gamma)}\right)\\
&\qquad +\frac{9}{16}G^2\left(Ea^{g(\alpha)+g(\beta)}-Ea^{g(\alpha)}-Ea^{g(\beta)}+1\right).
\end{aligned}
$$

Notice that

$$
\begin{aligned}
EM(\alpha) &= E\left[E\left[M(\alpha) \mid g(\alpha)=k\right]\right]\\
&= \sum_{k=0}^{n} E\left[M(\alpha) \mid g(\alpha)=k\right]P\left\{g(\alpha)=k\right\}\\
&= \sum_{k=0}^{n} G(1-p(k))P\left\{g(\alpha)=k\right\}\\
&= \sum_{k=0}^{n} \frac{3}{4}G\left(1-a^k\right)P\left\{g(\alpha)=k\right\}\\
&= \frac{3}{4}G\left(1-Ea^{g(\alpha)}\right),
\end{aligned}
$$

and

$$
EM(\beta) = \frac{3}{4}G\left(1-Ea^{g(\beta)}\right).
$$

We have

$$\begin{aligned} Cov(M(\alpha), M(\beta)) &= E\left(M(\alpha)M(\beta)\right) - EM(\alpha)EM(\beta) \\ &= \frac{3}{4}G\left(Ea^{g(\alpha)+g(\beta)-2g(\gamma)} - Ea^{g(\alpha)+g(\beta)-g(\gamma)}\right) \\ &\quad + \frac{9}{16}G^2\left(Ea^{g(\alpha)+g(\beta)} - Ea^{g(\alpha)}Ea^{g(\beta)}\right) \\ &= \frac{3}{4}G\left(Ea^{g(\alpha)+g(\beta)-2g(\gamma)} - Ea^{g(\alpha)+g(\beta)-g(\gamma)}\right) \\ &\quad + \frac{9}{16}G^2 Cov\left(a^{g(\alpha)}, a^{g(\beta)}\right). \end{aligned}$$

The lemma is proved. □

Now we study $1 - Ea^{g(\gamma)}$ first. Let $C_n(k)$ be the expected number of pairs with k^{th} generation MRCA when $S_0 = 1$. It was shown in Sun (1995) that the generating function of $C_n(k)$ is

$$\varphi_{C_n}(x) = \sum_{k=0}^{n-1} C_n(k)x^k = \varphi_{C_n}(x) = \lambda\frac{(1+\lambda x)^n - (1+\lambda)^{2n}}{(1+\lambda x) - (1+\lambda)^2}.$$

Lemma 2. *Let A_n be the generation number of the MRCA of a randomly chosen pair with replacement from the products after n PCR cycles. Then*

$$\lim_{S_0\to\infty} S_0\left(1 - Ea^{A_n}\right) = \frac{1}{(1+\lambda)^{2n}}\left(2\frac{(1+\lambda)^n - (1+\lambda)^{2n}}{(1+\lambda) - (1+\lambda)^2} - 2\frac{(1+a\lambda)^n - (1+\lambda)^{2n}}{(1+a\lambda) - (1+\lambda)^2} + (1+\lambda)^n - (1+a\lambda)^n\right).$$

Proof. It was proved by Sun (1995) that for $1 \le k \le n$,

$$\lim_{S_0\to\infty} S_0 P\left\{A_n = k\right\} = \frac{2C_n(k) + \binom{n}{k}\lambda^k}{(1+\lambda)^{2n}}.$$

Therefore

$$
\begin{aligned}
&\lim_{S_0\to\infty} S_0\left(1-Ea^{A_n}\right)\\
&=\lim_{S_0\to\infty} S_0\left(1-\sum_{k=0}^{n} a^k P\left\{A_n=k\right\}\right)\\
&=\lim_{S_0\to\infty} S_0\left(1-P\left\{A_n=0\right\}-\sum_{k=1}^{n} a^k P\left\{A_n=k\right\}\right)\\
&=\lim_{S_0\to\infty} S_0\sum_{k=1}^{n}\left(1-a^k\right)P\left\{A_n=k\right\}\\
&=\sum_{k=1}^{n}\left(1-a^k\right)\frac{2C_n(k)+\binom{n}{k}\lambda^k}{(1+\lambda)^{2n}}\\
&=\frac{1}{(1+\lambda)^2 n}\left(2\varphi_{C_n}(1)-2\varphi_{C_n}(0)+(1+\lambda)^n-1-2\varphi_{C_n}(a)+2\varphi_{C_n}(0)-(1+a\lambda)^n+1\right)\\
&=\frac{1}{(1+\lambda)^{2n}}\left(2\frac{(1+\lambda)^n-(1+\lambda)^{2n}}{(1+\lambda)-(1+\lambda)^2}-2\frac{(1+a\lambda)^n-(1+\lambda)^{2n}}{(1+a\lambda)-(1+\lambda)^2}+(1+\lambda)^n-(1+a\lambda)^n\right).
\end{aligned}
$$

Lemma 2 is proved. □

Next we study $Cov\left(a^{g(\alpha)}, a^{g(\beta)}\right)$. Let X_k^n be the number of k^{th} generation sequences after n PCR cycles. We have

$$P\left\{g(\alpha)=k, g(\beta)=l\right\}=E\frac{X_k^n X_l^n}{S_n^2}.$$

Therefore,

$$
\begin{aligned}
Cov\left(a^{g(\alpha)}, a^{g(\beta)}\right) &= E\sum_{k,l}\frac{a^k a^l X_k^n X_l^n}{S_n^2}-\left(E\sum_k\frac{a^k X_k^n}{S_n}\right)^2\\
&= E\left(\sum_k\frac{a^k X_k^n}{S_n}\right)^2-\left(E\sum_k\frac{a^k X_k^n}{S_n}\right)^2\\
&= Var\left(\sum_k\frac{a^k X_k^n}{S_n}\right).
\end{aligned}
$$

Let $T_n=\sum_{k=0}^{n} a^k X_k^n$ and $T_n(i)$ be the corresponding quantity generated by 0^{th} generation sequence i. Then we have

$$T_n=\sum_{i=1}^{S_0} T_n(i)$$

and

$$Cov\left(a^{g(\alpha)}, a^{g(\beta)}\right) = Var\left(\frac{T_n}{S_n}\right) = Var\left(\frac{\overline{T}_n}{\overline{S}_n}\right), \tag{1}$$

where

$$\overline{T}_n = \frac{\sum_{i=1}^{S_0} T_n(i)}{S_0};$$
$$\overline{S}_n = \frac{\sum_{i=1}^{S_0} S_n(i)}{S_0}.$$

From equation (1) and Lemma 5 of Sun (1995), to obtain the limit behavior of $Cov\left(a^{g(\alpha)}, a^{g(\beta)}\right)$, we only need to know $Var(T_n)$, $Cov(T_n, S_n)$ and $Var(S_n)$. The following Lemma gives these quantities.

Lemma 3. *Suppose initially we have only one sequence, Let* $T_n = \sum_{k=0}^{n} a^k X_k^n$, *and* S_n *be the total number of sequences after* n *cycles. Then for* $n = 0, 1, 2, \ldots,$

$$Var(S_n) = (1-\lambda)(1+\lambda)^{n-1}\Big((1+\lambda)^n - 1\Big),$$

$$Cov(T_n, S_n) = a(1-\lambda)(1+a\lambda)^{n-1}\Big((1+\lambda)^n - 1\Big),$$

$$Var(T_n) = \frac{a(1-\lambda)\Big((1+a^2\lambda)^n - (1+a\lambda)^{2n}\Big)}{a-2-a\lambda}.$$

This lemma can be proved similarly as Lemma 6 in Sun (1995). From Lemma 3, equation (1) and the fact

$$ET_n(1) = \sum_{k=0}^{n} a^k \binom{n}{k} \lambda^k = (1+a\lambda)^n, \qquad ES_n(1) = (1+\lambda)^n,$$

we can proof the following Lemma.

Lemma 4. *Let* $g(\alpha)$ *and* $g(\beta)$ *be the generation numbers of a randomly chosen pair*

from the products after n PCR cycles with replacement. Then

$$\lim_{S_0\to\infty} S_0 Cov\left(a^{g(\alpha)}, a^{g(\beta)}\right) = \frac{a(1-\lambda)\left((1+\lambda a^2)^n - (1+a\lambda)^{2n}\right)}{(a-2-a\lambda)(1+\lambda)^{2n}} + \frac{(1-\lambda)(1-2a-a\lambda)(1+a\lambda)^{2n}((1+\lambda)^n-1)}{(1+a\lambda)(1+\lambda)^{3n+1}}.$$

Next we study the limit behavior of $Var(M_1)$. Let K be the generation number of a randomly chosen sequence after n PCR cycles. Notice that

$$EM_1 = \frac{3}{4}G\left(1 - Ea^K\right) \tag{2}$$

and

$$\begin{aligned} EM_1^2 &= \sum_{k=0}^{n} E\left[M_1^2 \mid K=k\right] P\left\{K=k\right\} \\ &= \sum_{k=1}^{n} \left(G^2\big(1-p(k)\big)^2 + G\big(1-p(k)\big)\right) P\left\{K=k\right\} \\ &= \left(\left(\frac{3}{4}G\right)^2\left(1-a^k\right)^2 + \frac{3}{4}G\left(1-a^k\right)\right) P\left\{K=k\right\} \\ &= \frac{9}{16}G^2\left(1-2Ea^K+Ea^{2K}\right) + \frac{3}{4}G(1-Ea^K), \end{aligned}$$

we have

$$\begin{aligned} Var(M_1) &= E\left(M_1^2\right) - \left(EM_1\right)^2 \\ &= \frac{9}{16}G^2\left(1-2Ea^K+Ea^{2K}\right) + \frac{3}{4}G(1-Ea^K) - \frac{9}{16}G^2\left(1-Ea^K\right)^2 \\ &= \frac{9}{16}G^2\left(Ea^{2K} - \left(Ea^K\right)^2\right) + \frac{3}{4}G\left(1-Ea^K\right). \end{aligned} \tag{3}$$

The following lemma gives the limit behavior of Ea^K.

Lemma 5. *Let K be the generation number of a randomly chosen sequence after n PCR cycles. Then*

$$\lim_{S_0\to\infty} S_0\left(Ea^K - \frac{(1+a\lambda)^n}{(1+\lambda)^n}\right) = (1-a)(1-\lambda)((1+\lambda)^n-1)\frac{(1+a\lambda)^{n-1}}{(1+\lambda)^{2n+1}}.$$

Proof. Notice that

$$P\{K=k\} = E\frac{X_k^n}{S_n}.$$

Thus we have

$$Ea^K = \sum_{k=0}^{n} a^k E\frac{X_k^n}{S_n} = E\frac{\sum_{k=0}^{n} a^k X_k^n}{S_n} = E\frac{T_n}{S_n} = E\frac{\overline{T}_n}{\overline{S}_n}.$$

Then using Lemma 6 in Sun (1995), we can proof this lemma. □

Proof of Theorem 2. Now it's easy to proof Theorem 2. From equation (2) and Lemma 5, we see the first assertion of the Theorem holds. From Lemmas 1, 2, 4, 5 and equation (3), we see the second assertion of the theorem holds.

Table 1. Notation summary

n:	number of PCR cycles
S_i:	total number of sequences after i PCR cycles, $i = 0, 1, \ldots, n$
X_k^n:	number of k^{th} generation sequences after n PCR cycles
λ:	efficiency of PCR
G:	number of bases in the target sequences
$\exp(-\mu)$:	probability a base is not mutated per PCR cycle
$p(k)$:	probability a base is unchanged after k replications
s:	sample size
M:	number of base changes of a randomly chosen sequence
D:	pair-wise distance between a pair of randomly chosen sequence
H:	Hamming distance between a pair of randomly chosen sequences
$N_{i,j}$:	number of sequences in cycle i in the genealogy of the subsample that generated from initial sequence j
$R_{i,j}$:	number of replications in cycle i in the genealogy of the subsample that generated from initial sequence j
$L_{i,j}$:	number of coalescent events in cycle i in the genealogy of the subsample that generated from initial sequence j

Table 2. The comparison of the four estimating methods with $\lambda = 0.8$, $n = 30$, $G = 500$, and $s = 30$.

a) real mutation rate $1 - \exp(-\mu) = 1 \times 10^{-3}$

S_0		1	10	100	1000
moment	mean($\times 10^{-3}$)	0.985	0.995	0.993	0.995
estimation	std-deviation($\times 10^{-4}$)	1.14	0.83	0.83	0.81
MLE	mean($\times 10^{-3}$)	1.178	1.050	1.002	1.001
	std-deviation($\times 10^{-4}$)	2.18	1.02	0.84	0.82
method of	mean($\times 10^{-3}$)	0.994	1.004	1.002	1.005
base changes	std-deviation($\times 10^{-4}$)	1.16	0.85	0.84	0.82
method of pair-	mean($\times 10^{-3}$)	0.990	1.003	1.002	1.005
wise differences	std-deviation($\times 10^{-4}$)	1.01	0.84	0.84	0.83

b) real mutation rate $1 - \exp(-\mu) = 5 \times 10^{-3}$

S_0		1	10	100	1000
moment	mean($\times 10^{-3}$)	4.712	4.766	4.772	4.778
estimation	std-deviation($\times 10^{-4}$)	4.20	3.26	3.18	3.09
MLE	mean($\times 10^{-3}$)	5.630	5.032	4.804	4.778
	std-deviation($\times 10^{-4}$)	7.16	2.44	3.00	3.14
method of	mean($\times 10^{-3}$)	4.927	4.986	4.992	4.999
base changes	std-deviation($\times 10^{-4}$)	3.42	2.50	2.43	2.36
method of pair-	mean($\times 10^{-3}$)	4.898	4.980	4.992	5.000
wise differences	std-deviation($\times 10^{-4}$)	3.22	2.47	2.43	2.37

c) real mutation rate $1 - \exp(-\mu) = 10 \times 10^{-3}$

S_0		1	10	100	1000
moment	mean($\times 10^{-3}$)	8.985	9.135	9.136	9.123
estimation	std-deviation($\times 10^{-4}$)	11.25	9.42	9.37	9.47
MLE	mean($\times 10^{-3}$)	10.742	9.642	9.194	9.117
	std-deviation($\times 10^{-4}$)	9.39	5.33	8.89	9.60
method of	mean($\times 10^{-3}$)	9.812	9.990	9.991	9.976
base changes	std-deviation($\times 10^{-4}$)	6.07	4.46	4.35	4.29
method of pair-	mean($\times 10^{-3}$)	9.763	9.980	9.989	9.975
wise differences	std-deviation($\times 10^{-4}$)	5.97	4.43	4.35	4.29

Table 3. The comparison of the four estimating methods with efficiency as a function of PCR cycles. Here, $n = 30$, $G = 500$, $s = 30$ and real mutation rate $1 - \exp(-\mu) = 10 \times 10^{-3}$.

S_0		1	10	100	1000
moment	mean($\times 10^{-3}$)	8.319	8.518	8.556	8.552
estimation	std-deviation($\times 10^{-4}$)	17.92	15.39	14.96	14.99
MLE	mean($\times 10^{-3}$)	11.252	9.899	9.354	9.284
	std-deviation($\times 10^{-4}$)	14.72	4.93	7.79	8.42
method of	mean($\times 10^{-3}$)	8.972	9.201	9.246	9.242
base changes	std-deviation($\times 10^{-4}$)	12.56	9.33	8.81	8.85
method of pair-	mean($\times 10^{-3}$)	8.853	9.177	9.243	9.241
wise differences	std-deviation($\times 10^{-4}$)	13.36	9.51	8.83	8.85