Mapping short DNA sequencing reads and variants calling using mapping quality scores (Supplementary Text)

Heng Li, Jue Ruan and Richard Durbin

1 Read Base-Calling Errors

In this supplement text, a letter in uppercase indicates a random variable, whereas a letter in lowercase represents a constant, a known value or a function.

Let $\Sigma = \{\text{'A','C','G','T'}\}$ be the alphabet of the four nucleotides. In sequencing, the true nucleotide is $B \in \Sigma$ and the one estimated by base caller is $\hat{B}$. The base error ϵ_B is defined as:

$$\epsilon_B = \Pr\{\hat{B} \neq B\}$$

and base quality Q_B is:

Because log is base 2

$$Q_B = -c \log_2 \epsilon_B$$

where c is a scaling constant. For Phred quality, $c = 10/\log_2 10 \approx 3.434$. We have:

$$\Pr\{\hat{B} = \hat{b} | B = b\} = p(\hat{b}|b) \triangleq \begin{cases} 1 - \epsilon_B & \text{if } b = \hat{b} \\ \epsilon_B/3 & \text{otherwise} \end{cases}$$

A read Z is a random sequence with length l: $Z \in \Sigma^l$. If each site is independent of others, we know:

$$\Pr\{\text{observed } \hat{Z} = \hat{b}_1 \dots \hat{b}_l | \text{true } Z = b_1 \dots b_l\} = p(\hat{b}_1 \dots \hat{b}_l | b_1 \dots b_l) = \prod_{i=1}^{l} p(\hat{b}_i | b_i) \tag{1}$$

2 Single-End Mapping Errors

2.1 Notations

Given a known reference sequence $x \in \Sigma^L$, let x_u^l, $u = 1, \dots, L - l + 1$, be the l-long subsequence starting at position u. For a read coming from position U, $U \in \{1, \dots, L - l + 1\}$, the true read sequence is x_U^l and we observe Z. The probability of observing the read z on the condition that the read comes from position u is:

$$p(z|x, u) = \Pr\{Z = z | x, U = u\} = p(z|x_u^l)$$

where $p(z|x_u^l)$ is calculated by Equation 1. If we assume a read randomly comes from the reference, i.e.:

$$p(u|x) = \Pr\{U = u|x\} = \frac{1}{L - l + 1}$$

Too expensive to calculate. Also, what about gaps? Harder to enumerate every possible alignment when gaps are permitted.

the probability of read coming from u is:

$$p_M(u|x, z) = \Pr\{U = u | x, Z = z\} = \frac{p(z|x_u^l)}{\sum_{v=1}^{L-l+1} p(z|x_v^l)} \tag{2}$$

An alignment algorithm actually presents an estimate of U, denoted by $\hat{U}$. Given a read sequence z, the maximum likelihood estimate of U is:

$$\hat{u}(z) = \underset{u}{\operatorname{argmax}}\, p_M(u|x, z)$$

and the mapping error is:

$$\epsilon_M = \Pr\{\hat{U} \neq U | x, Z = z\} = 1 - p_M(\hat{u}(z)|x, z)$$

On real data, a read may not always come from the reference and the alignment program may not always align each read. For convenience, we define $U = 0$ if Z does not come from the reference and $\hat{U} = 0$ if the read is not aligned. In addition, the alignment program may not visit every position on the reference and therefore the sum in Equation 2 can not be accomplished. We will address this issue in the following section.

2.2 Isolating sources of errors

Though it's interesting to think about how we might make use of error info on unmapped reads

In practice, we only care about the errors of the reads that can be mapped. Then the mapping error can be expressed in three terms:

$$\begin{aligned}
\epsilon_M &= \Pr\{\hat{U} \neq U | \hat{U} > 0\} \\
&= \Pr\{U = 0 | \hat{U} > 0\} + \Pr\{U \notin \Omega, U > 0 | \hat{U} > 0\} + \Pr\{\hat{U} \neq U, U \in \Omega | \hat{U} > 0\} \\
&= \epsilon_{M_1} + \epsilon_{M_2}(1 - \epsilon_{M_1}) + \epsilon_{M_3}(1 - \epsilon_{M_1})(1 - \epsilon_{M_2})
\end{aligned}$$

where $\Omega \subset \{1, \ldots, L - l + 1\}$ is the set of positions that the program has visited at the alignment stage and

$$\begin{aligned}
\epsilon_{M_1} &= \Pr\{U = 0 | \hat{U} > 0\} \\
\epsilon_{M_2} &= \Pr\{U \notin \Omega | U > 0, \hat{U} > 0\} \\
\epsilon_{M_3} &= \Pr\{\hat{U} \neq U | U \in \Omega, \hat{U} > 0\}
\end{aligned}$$

Probability ϵ_{M_1} measures the error that the read does not come from the reference, ϵ_{M_2} the error that the true position is missed by the program and ϵ_{M_3} the error that the hit is not the true one. When all the three errors are sufficiently small, the overall probability ϵ_M can be approximated as the largest of the three:

$$\epsilon_M \approx \max\{\epsilon_{M_1}, \epsilon_{M_2}, \epsilon_{M_3}\}$$

The following subsections show the details about the calculation of the three types of errors. In these calculation, we assume there are no SNPs between the reference and the sample.

2.3 Calculating type-1 mapping errors

Based on the Bayesian formula,

$$\epsilon_{M_1} = \frac{\Pr\{\hat{U} > 0 | U = 0\} \cdot \Pr\{U = 0\}}{\Pr\{\hat{U} > 0 | U = 0\} \Pr\{U = 0\} + \Pr\{\hat{U} > 0 | U > 0\} \Pr\{U > 0\}}$$

This error is governed by two factors: $\Pr\{U = 0\}$, the prior of contamination and $\Pr\{\hat{U} > 0 | U = 0\}$, the probability that contamination can be mapped. For random contamination and reasonable read length, the second factor is very small because a long random sequence can hardly find a random hit.

In practical calculation, ϵ_{M_1} is not counted because it is usually small in comparison to other types of errors and because its calculation requires prior information about the characteristics of contamination, which is hard to know in practice.

2.4 Calculating type-2 mapping errors

Error ϵ_{M_2} is determined by the alignment program. If we use the standard Smith-Waterman alignment, this error is zero because the program will visit all the possible positions. In practice, heuristic algorithms are usually used to accelerate alignment and the basic idea of heuristic algorithm is to skip a bulk of less likely positions. This may cause the true hit to be missed and lead to mapping errors.

Is the assumption that the best hit is always found?

For simplicity, we assume the base-calling error of each base is ϵ. The true hit is missed when the following three events happen at the same time: i) the true hit is not the best hit; ii) the true hit is a sub-optimal hit that is close to the best hit; iii) the sub-optimal hit is missed by the program.

Suppose the best hit has k' mismatches and the true hit has k mismatches with $k \geq k'$. The probability that the first event happens can be approximated as the probability that $k - k'$ errors arise from the $l - k'$ matches of the best hit, and therefore $k - k' \sim \text{Binomial}(l - k', \epsilon)$. The second source is determined by the repeat structure of the reference and can be estimated from the overall alignment. We denote by $p_1(k - k', l)$ the probability that a k-mismatch hit and a k'-mismatch hit coexist, given l-long reads. This probability can be estimated at the alignment stage. The third source is determined by the heuristic algorithm itself. Let $p_2(k)$ be the probability that k-mismatch hit may be missed by the alignment program. As a consequence, the type-2 mapping errors is approximated as:

$$\epsilon_{M_2} \approx \sum_{k=k'}^{l} \binom{l-k'}{k-k'} \cdot \epsilon^{k-k'} \cdot p_1(k-k', l) \cdot p_2(k)$$

We further approximate this probability by replacing the sum with the term corresponds to the smallest k that makes $p_2(k) \neq 0$.

From this equation, we know that type-2 mapping error can be reduced if i) sequencing error is lower (ϵ is smaller) ii) the best hit has fewer mismatches (k' is smaller); iii) the reference contains fewer repeats (p_1 is smaller); iv) the program is more sensitive (p_2 is smaller). For the Smith-Waterman algorithm, $p_2(k)$ is always zero and therefore there is no type-2 error.

For MAQ with default options, $p_2(0) = p_2(1) = p_2(2) = 0$ and $p_2(3) = \binom{4}{3} \cdot 7^3 / \binom{28}{3} \approx 0.42$. We further simplify the type-2 quality as:

$$Q_{M_2} \approx 4 + (3 - k') \cdot (\bar{q} - 14) - 4.343 \log p_1(3 - k', 28)$$

where $\bar{q}$ is the average base quality, and we approximate $-10 \log_{10}(0.42) \approx 4$ and $-10 \log_{10}(28-2) = 14$. On human, $p_1(0, 28) \approx 0.2$, $p_1(1, 28) \approx 0.05$ and $p_1(2, 28) \approx 0.03$. As quality are log scaled, the various approximation here will not greatly affect the accuracy of mapping quality.

2.5 Calculating type-3 mapping errors

2.5.1 Theory

Omega is the set of positions that the program evaluated

Again assuming a uniform prior for U, for $u \in \Omega$, we can calculate the posterior probability that the read comes from u:

$$p_{M_3}(u|x, z, \Omega) = \Pr\{U = u|x, U \in \Omega, Z = z\} = \frac{p(z|x_u^l)}{\sum_{v \in \Omega} p(z|x_v^l)}$$

The position of the best hit is:

$$\hat{u}(z) = \operatorname*{argmax}_u p_{M_3}(u|x, z, \Omega)$$

And the probability that the best hit is wrong is:

$$\epsilon_{M_3} = 1 - p_{M_3}(\hat{u}(z)|x, z, \Omega)$$

2.5.2 Practical calculation

If there are n_1 equally best matches, p_{M_3} must be smaller than $1/n_1$. As we mainly focus on reads mapped with high confidence, we only consider $n_1 = 1$. In the following formulae, p_1 is the error probability of the best hit, p_2 is the error of the second best hit and n_2 is the number of second

best hits.

$$\begin{aligned} Q_{M_3} &= -c\log\epsilon_{M_3} \\ &= -c\log\left(1-\frac{p_1}{p_1+n_2p_2+n_3p_3+\cdots}\right) \\ &= -c\log\frac{n_2p_2+\cdots}{p_1+n_2p_2+\cdots} \\ &\approx -c\log n_2 - c\log p_2 + c\log p_1 \\ &= (-c\log p_2) - (-c\log p_1) - c\log n_2 \end{aligned}$$

The approximation can be quite accurate when $p_1 \ll n_2p_2$ and $n_2p_2 \ll n_3p_3$.

In practical calculation, MAQ records the best two hits and their sum of errors $-c\log p_1$ and $-c\log p_2$, and approximate n_2 by the number of hits having the same number of mismatches as the second best hit.

3 Consensus Base Errors

In consensus base-calling, if there is only one type of nucleotide at a position, the consensus base can only be called as that type. If there are two or more types of nucleotides, we usually focus on the two dominant ones. To this end, we might as well assume that each position is covered by only two types of nucleotides. We can always achieve this by ignoring other types as errors.

3.1 General formulae

Before presenting the theory about the consensus base qualities, we first see some general formulae.

For any $0 \le \beta_{nk} < 1$ $(0 \le k \le n)$:

$$\sum_{k=0}^{n}(1-\beta_{nk})\prod_{i=0}^{k-1}\beta_{ni} = 1 - \prod_{k=0}^{n}\beta_{nk}$$

where we regard that $\prod_{i=0}^{-1}\beta_{ni} = 1$. In particular, when $\exists k \in [0,n]$ satisfies $\beta_{nk} = 0$, we have:

$$\sum_{k=0}^{n}(1-\beta_{nk})\prod_{i=0}^{k-1}\beta_{ni} = 1$$

If we further define:

$$\alpha_{nk} = (1-\beta_{nk})\prod_{i=0}^{k-1}\beta_{ni} \tag{3}$$

on the condition that some $\beta_{nk} = 0$, we have:

$$\sum_{k=0}^{n}\alpha_{nk} = 1$$

$$\beta_{nk} = 1 - \frac{\alpha_{nk}}{1-\sum_{i=0}^{k-1}\alpha_{ni}} = \frac{1-\sum_{i=0}^{k}\alpha_{ni}}{1-\sum_{i=0}^{k-1}\alpha_{ni}} = \frac{\sum_{i=k+1}^{n}\alpha_{ni}}{\sum_{i=k}^{n}\alpha_{ni}}$$

In the context of consensus base calling, if we define:

$$\beta_{nk} = \begin{cases} \Pr\{\text{more than } k \text{ errors}|\text{more than } k-1 \text{ errors in } n \text{ bases}\} & (k>0) \\ \Pr\{\text{more than 0 error in } n \text{ bases}\} & (k=0) \end{cases}$$

$\beta_{nn} = 0$. Then α_{nk} is the probability that exactly k errors arise from n bases:

$$\alpha_{nk} = \Pr\{\text{exactly } k \text{ errors in } n \text{ bases}\}$$

To be more explicit, α_{nk} is a function of the error rates of the n bases covering the position:

$$\alpha_{nk} = \alpha_{nk}(\epsilon_1, \ldots, \epsilon_k; \epsilon_{k+1}, \ldots, \epsilon_n)$$

where $\epsilon_1 \leq \ldots \leq \epsilon_k$ are the error rates of the k wrong bases, and $\epsilon_{k+1} \leq \ldots \leq \epsilon_n$ those of the $n-k$ correct bases.

3.2 Modelling error dependency

If we assume errors come independently and the base error is uniformly $\bar{\epsilon}$ for all bases, the probability of seeing k errors in n bases is:

$$\bar{\alpha}_{nk}(\bar{\epsilon}) \triangleq \binom{n}{k} \bar{\epsilon}^k (1-\bar{\epsilon})^{n-k} \tag{4}$$

and

$$\bar{\beta}_{nk}(\bar{\epsilon}) \triangleq \frac{1 - \sum_{i=0}^{k} \bar{\alpha}_{ni}}{1 - \sum_{i=0}^{k-1} \bar{\alpha}_{ni}} \tag{5}$$

If errors are correlated, we expect to see errors coming more frequently and therefore $\beta_{nk}(\bar{\epsilon}) \geq \bar{\beta}_{nk}(\bar{\epsilon})$. A possible choice is to assume:

$$\beta_{nk}(\bar{\epsilon}) = \bar{\beta}_{nk}^{f_k}(\bar{\epsilon})$$

where $0 < f_k \leq 1$. From Equation 3, we can calculate the probability that k errors arise from n bases:

$$\alpha_{nk}(\bar{\epsilon}) = (1 - \bar{\beta}_{nk}^{f_k}) \prod_{i=0}^{k-1} \bar{\beta}_{ni}^{f_i} = \left(1 - \bar{\beta}_{nk}^{f_k}\right) \prod_{i=0}^{k-1} \left(\frac{\bar{\beta}_{ni}}{\bar{\epsilon}}\right)^{f_i} \cdot \bar{\epsilon}^{f_i} \equiv c_{nk}(\bar{\epsilon}) \cdot \prod_{i=0}^{k-1} \bar{\epsilon}^{f_i}$$

where:

$$c_{nk}(\bar{\epsilon}) \triangleq \left[1 - \bar{\beta}_{nk}^{f_k}(\bar{\epsilon})\right] \prod_{i=0}^{k-1} \left[\frac{\bar{\beta}_{ni}(\bar{\epsilon})}{\bar{\epsilon}}\right]^{f_i} \tag{6}$$

Under $f_k = 1, k = 0, \ldots, n$, $c_{nk}(\bar{\epsilon}) = \binom{n}{k}(1-\bar{\epsilon})^{n-k}$, which is insensitive to $\bar{\epsilon}$ and contributes less to $\alpha_{nk}(\bar{\epsilon})$ than $\prod_i \bar{\epsilon}^{f_i}$. Based on this observation, we approximate $\alpha_{nk}(\epsilon_1, \ldots, \epsilon_k; \epsilon_{k+1}, \ldots, \epsilon_n)$ as:

$$\alpha_{nk}(\epsilon_1, \ldots, \epsilon_k; \epsilon_{k+1}, \ldots, \epsilon_n) \approx c_{nk}(\bar{\epsilon}) \cdot \prod_{i=0}^{k-1} \epsilon_{i+1}^{f_i} \tag{7}$$

with

$$\log \bar{\epsilon} = \frac{\sum_{i=0}^{k-1} f_i \log \epsilon_{i+1}}{\sum_{i=0}^{k-1} f_i} \tag{8}$$

In practice, we precalculate a table for $c_{nk}(\bar{\epsilon})$ given different n, k and $\bar{\epsilon}$ using Equation 4-6. At each position along the reference, we compute $\bar{\epsilon}$ with Equation 8, look up the precalculated $c_{nk}(\bar{\epsilon})$ and finally compute $\alpha_{nk}(\epsilon_1, \ldots, \epsilon_k; \epsilon_{k+1}, \ldots, \epsilon_n)$ with Equation 7.

The error dependency is governed by f_k. In principle they can be estimated from input data, but MAQ only takes a very simple form:

$$f_k = 0.85^k$$

In practice, this f_k can give a reasonable accuracy on real data.

Another important factor in calculating errors is the orientation of a read. Reads coming from different strands are largely independent of each other. MAQ uses the following α_{nk}:

$$\begin{aligned} & \alpha_{nk}(\epsilon_1, \ldots, \epsilon_k; \tilde{\epsilon}_1, \ldots, \tilde{\epsilon}_{\tilde{k}}; \epsilon_{k+1}, \ldots, \epsilon_n; \tilde{\epsilon}_{\tilde{k}+1}, \ldots, \tilde{\epsilon}_{\tilde{n}}) \\ \approx \quad & c_{nk}(\bar{\epsilon}) \prod_{i=0}^{k-1} \epsilon_{i+1}^{f_i} \cdot c_{\tilde{n}\tilde{k}}(\bar{\tilde{\epsilon}}) \prod_{\tilde{i}=0}^{\tilde{k}-1} \tilde{\epsilon}_{\tilde{i}+1}^{f_{\tilde{i}}} \end{aligned}$$

where there are k errors out of n bases on the forward strand and $\tilde{k}$ out of $\tilde{n}$ on the reverse strand.

3.3 Consensus genotype calling and qualities

Given a position of a diploid sample, suppose we are observing k nucleotide b and $n-k$ nucleotide b'. The error rates of the b bases are: $\epsilon_1 \le \ldots \le \epsilon_k$, and those of the b' are: $\epsilon_{k+1} \le \ldots \le \epsilon_n$. For convenience, define:

$$\alpha''_{nk} \triangleq \alpha_{nk}(\epsilon_1, \ldots, \epsilon_k; \epsilon_{k+1}, \ldots, \epsilon_n) \quad (9)$$

$$\alpha_{n,n-k} \triangleq \alpha_{n,n-k}(\epsilon_{k+1}, \ldots, \epsilon_n; \epsilon_1, \ldots, \epsilon_k) \quad (10)$$

Let $\mathcal{D}$ represent the observed data and $G = \langle H, H' \rangle$ be the true genotype at the position. Here $\langle \cdot, \cdot \rangle$ indicates that this is an unordered pair. Then α''_{nk} and $\alpha_{n,n-k}$ actually mean:

$$\begin{aligned} \alpha''_{nk} &= \Pr\{\mathcal{D}|G = \langle b', b' \rangle\} \\ \alpha_{n,n-k} &= \Pr\{\mathcal{D}|G = \langle b, b \rangle\} \end{aligned}$$

They give the probability when the genotype is homozygous. When the true genotype is heterozygous, we can approximate the probability with:

$$\alpha'_{nk} = \alpha'_{n,n-k} \triangleq \Pr\{\mathcal{D}|G = \langle b, b' \rangle\} \approx \frac{1}{2^n}\binom{n}{k} \quad (11)$$

As a consequence, the posterior probabilities are:

$$\begin{aligned} p_g(\langle b, b \rangle|\mathcal{D}) &= \frac{\alpha_{n,n-k}}{\alpha''_{nk} + \bar{r} \cdot \alpha'_{nk} + \alpha_{n,n-k}} \\ p_g(\langle b, b' \rangle|\mathcal{D}) &= \frac{\bar{r} \cdot \alpha'_{nk}}{\alpha''_{nk} + \bar{r} \cdot \alpha'_{nk} + \alpha_{n,n-k}} \\ p_g(\langle b', b' \rangle|\mathcal{D}) &= \frac{\alpha''_{nk}}{\alpha''_{nk} + \bar{r} \cdot \alpha'_{nk} + \alpha_{n,n-k}} \end{aligned}$$

where $\bar{r} = 2r/(1-r)$ and r is the prior of seeing a heterozygote. The estimated genotype is the one that maximizes the posterior probability p_g.

In practical calculation, MAQ does not directly calculate p_g. Instead, it calculates:

$$\begin{aligned} q^{(1)} &= -c \cdot \log \alpha_{n,n-k} \\ q^{(2)} &= -c \cdot \log \alpha''_{n,k} \\ q^{(3)} &= -c \cdot \log(r \cdot \alpha'_{nk}) \end{aligned}$$

estimates the genotype as:

$$\hat{g} = \underset{g \in \{1,2,3\}}{\operatorname{argmin}} \{q^{(g)}\}$$

and approximates the quality as:

$$Q_g = \min_{g \neq \hat{g}} \{q^{(g)}\} - q^{(\hat{g})}$$

As Phred qualities are logarithm scaled, calculating qualities also in the logarithm scale is cheap.

4 Alternative Strategies for SNP Calling

4.1 Independent model

Although the approximate theory in Section 3 can be adapted to the case where sequencing errors are independent, we have a simpler model in this case.

Suppose a position is covered by n reads. For the base of the i-th read, the true base is B_i and the observed base is $\hat{B}_i = \hat{b}^{(i)}$ with error probability ϵ_i. For convenience, we assume the

first k bases are called as b_1 (i.e. $\hat{b}^{(1)} = \cdots = \hat{b}^{(k)} = b_1$) and the rest of $n-k$ bases as b_2 (i.e. $\hat{b}^{(k+1)} = \cdots = \hat{b}^{(n)} = b_2$). We have:

$$P(\mathcal{D}|\langle b_2, b_2\rangle) = \Pr\{\hat{B}_1 = \cdots = \hat{B}_k = b_1, \hat{B}_{k+1} = \cdots = \hat{B}_n = b_2|\langle b_2, b_2\rangle\} = \prod_{i=1}^{k} \epsilon_i \cdot \prod_{j=k+1}^{n} (1-\epsilon_j)$$

$$P(\mathcal{D}|\langle b_1, b_1\rangle) = \prod_{i=1}^{k} (1-\epsilon_i) \cdot \prod_{j=k+1}^{n} \epsilon_j$$

and

$$\begin{aligned}
& P(\mathcal{D}|\langle b_1, b_2\rangle) \\
= \; & \Pr\{\hat{B}_1 = \cdots = \hat{B}_k = b_1, \hat{B}_{k+1} = \cdots = \hat{B}_n = b_2|\langle b_1, b_2\rangle\} \\
= \; & \sum_{a_1=1}^{2} \cdots \sum_{a_n=1}^{2} \Pr\{\hat{B}_1 = \cdots = \hat{B}_k = b_1, \hat{B}_{k+1} = \cdots = \hat{B}_n = b_2|B_1 = b_{a_1}, \cdots, B_n = b_{a_n}\} \\
& \cdot \Pr\{B_1 = b_{a_1}, B_2 = b_{a_2}, \ldots, B_n = b_{a_n}|\langle b_1, b_2\rangle\} \\
= \; & \frac{1}{2^n} \prod_{i=1}^{n} \sum_{a_i=1}^{2} \Pr\{\hat{B}_i = \hat{b}^{(i)}|B_i = b_{a_i}\}
\end{aligned}$$

If we assume that $\Pr\{\hat{B} = b\} = \Pr\{B = b\}$:

$$\Pr\{\hat{B}_i = \hat{b}^{(i)}|B_i = b_{a_i}\} = \Pr\{B_i = b_{a_i}|\hat{B}_i = \hat{b}^{(i)}\}$$

and as a result:

$$P(\mathcal{D}|\langle b_1, b_2\rangle) = \frac{1}{2^n}$$

Following a similar procedure in Section 3.3, we can calculate the posterior probability of each genotype given data, call the genotype that maximizes the posterior probability, and compute Phred-like quality.

4.2 Theoretical accuracy of the k-allele method

The most intuitive SNP calling method might be to call an allele if there are k reads that support the allele. We call this simple strategy k-allele method.

Assume in resequencing the average read depth is λ. The read depth n_i at position i is then drawn from the Poisson distribution $\mathrm{Po}(\lambda)$. The probability of seeing k identical errors is:

$$p_e(k|n_i) = \frac{1}{3^{k-1}} \binom{n_i}{k} \epsilon^k (1-\epsilon)^{n_i - k}$$

where ϵ is the error probability of a base. Define L_k as the length of reference covered by at least k reads:

$$L_k = L(1-\pi)\left(1 - e^{-\lambda} \sum_{j=0}^{k-1} \frac{\lambda^j}{j!}\right)$$

where π is the fraction of true polymorphic sites in comparison to the reference. The expected number of sites having k identical errors is:

$$\begin{aligned}
N_{\mathrm{err}}(k) &= L_k \sum_{n=k}^{\infty} p_e(k|n) \cdot \frac{\lambda^n}{n!} e^{-\lambda} \\
&= 3L_k \cdot \left[\frac{\epsilon}{3(1-\epsilon)}\right]^k e^{-\lambda} \sum_{n=k}^{\infty} \binom{n}{k} (1-\epsilon)^n \frac{\lambda^n}{n!} \\
&= \frac{3L_k}{k!} \left(\frac{\epsilon\lambda}{3}\right)^k e^{-\epsilon\lambda}
\end{aligned}$$

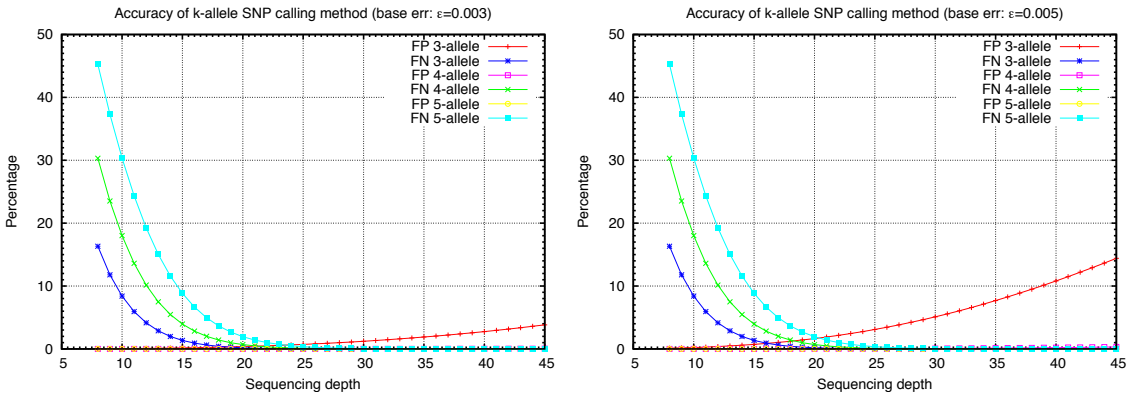

Figure 1: Theoretical error rate of k-allele method. In the left figure, we assume a uniform error rate 0.003, and in the right 0.005.

Note that $\epsilon\lambda$ is typically at the order of 10^{-2} and therefore we can consider $N_{\text{err}}(k+1) \ll N_{\text{err}}(k)$.

If we call an allele whenever k reads are supporting the allele, the approximate false positive rate will be:

$$\begin{aligned}
\text{FP}_k &= 1 - \frac{\pi L}{\pi L + \sum_{l=k}^{\infty} N_{\text{err}}(l)} \\
&\approx 1 - \frac{\pi L}{\pi L + N_{\text{err}}(k)} \\
&= 1 - \left[1 + \frac{1-\pi}{\pi}\frac{3}{k!}\left(\frac{\epsilon\lambda}{3}\right)^k \left(1 - e^{-\lambda}\sum_{j=0}^{k-1}\frac{\lambda^j}{j!}\right)e^{-\epsilon\lambda}\right]^{-1}
\end{aligned}$$

and the false negative rate, or the fraction that a difference betweent the sample and the reference is missed, is approximately:

$$\text{FN}_k \approx \frac{2e^{-\lambda/2}}{3}\sum_{j=0}^{k-1}\frac{(\lambda/2)^j}{j!} + \frac{e^{-\lambda}}{3}\sum_{j=0}^{k-1}\frac{\lambda^j}{j!}$$

It is independent of the error rate ϵ.

The left panel in Figure 1 gives the theoretical FN and FP given $\epsilon = 0.003$ (equivalent to Q25). At low depth, 3-allele mode works well but when λ is larger than 10, we should switch to 4-allele mode to reduce FP. When the base error rate ϵ equals 0.005 (equivalent to Q23), we may want to use 4-allele mode even at low depth and switch to 5-allele at deep depth. Comparing the two figure, we can see that the FP is sensitive to ϵ.