ℓ-mer

goodgood

June, 02, 2009

1 initial $c = 2\nu_2/\nu_1$

We compare the c under the 65K E.coli segement with the simulate reads under c_0: 1X, 1.5X, 2X, 4X, 6X, 10X, 15X, 20X, where let ℓ=8, L=35, to experiment of the initial c. Remark: the c here is the effective ℓ-mer coverage, defined as $c = N(L-\ell+1)/(G-L+1)$.
G: genome size
N: number of reads
ℓ: ℓ-mer length.
c_0: the reads coverage, defined as $c_0 = NL/G$
so $c_0 = cL/(L-\ell+1)$ in numeric for $G \gg L$.
There are 10 samples for each experiments, and for the large c, the ν_1, ν_2 become too small to estimate c, for it may have large fluctuation effect, so we have to estimate with $c = k\nu_k/\nu_{k-1}$, to reduce the fluctuation.

Table 1: Estimate c from the count ν_1, ν_2 under different reads coverage,

c_0	1X	1.5X	2X	4X	6X	10X	15X	20X
ν_1	11023	11020.2	7409.6	2444.1	713	55.8	2.6	1.6
ν_2	6864	6886	7250.1	4241.7	1794.5	193.1	4.3	1.1
$\bar{c}$	1.25	1.25	1.96	3.48	5.07	7.29	3.4	1.3
$\bar{c_0}$	1.56	1.56	2.45	4.35	6.33	9.11	4.25	1.7

Table 2: intial c from $k\nu_k/\nu_{k-1}$ for proper reads coverage

c_0	1X	1.5X	2X	4X	6X	10X	15X	20X
formula	$2\nu_2/\nu_1$	$2\nu_2/\nu_1$	$2\nu_2/\nu_1$	$3\nu_3/\nu_2$	$5\nu_5/\nu_4$	$8\nu_8/\nu_7$	$12\nu_{12}/\nu_{11}$	$15\nu_{15}/\nu_{14}$
$\bar{c}$	1.24	1.25	1.96	3.37	5.13	8.22	12.34	15.90
$\bar{c_0}$	1.56	1.56	2.46	4.22	6.41	10.27	15.43	19.88

2 n_m distribution and estimation

the n_m denote the count of x(w)=m, the distribution of the n_m show the following table

Table 3: Distribution of n_m count for the 65K segement

m=0	$n_0 = 30208$	m=4	$n_4 = 1807$	m=8	$n_8 = 55$
m=1	$n_1 = 18676$	m=5	$n_5 = 766$	m=9	$n_9 = 21$
m=2	$n_2 = 9293$	m=6	$n_6 = 342$	m=10	$n_{10} = 11$
m=3	$n_3 = 4229$	m=7	$n_7 = 123$	m=11	$n_{11} = 5$

The result show the Poission assume for the n_m distribution is not good enough for the given ℓ-mer.

Table 4: Cumulative ratio according to the ν_k, $\ell = 8$, $c_0 = 2X$

k	a_k	a_k p-value	b_k	b_k p-value
$k = 11$	$a_{11} = -3.061$	0.000105	$b_{11} = 14.425$	1.82e-05
$k = 10$	$a_{10} = -2.578$	5.31e-07	$b_{10} = 13.635$	5.71e-08
$k = 9$	$a_9 = -1.637$	1.89e-09	$b_9 = 12.239$	5.83e-11
$k = 8$	$a_8 = -0.825$	2.89e-14	$b_8 = 10.661$	<2e-16
$k = 7$	$a_7 = -0.3205$	<2e-16	$b_7 = 8.6380$	<2e-16
$k = 6$	$a_6 = -0.08561$	<2e-16	$b_6 = 5.85284$	<2e-16
$k = 5$	$a_5 = -0.01015$	4.65e-11	$b_5 = 2.32009$	<2e-16

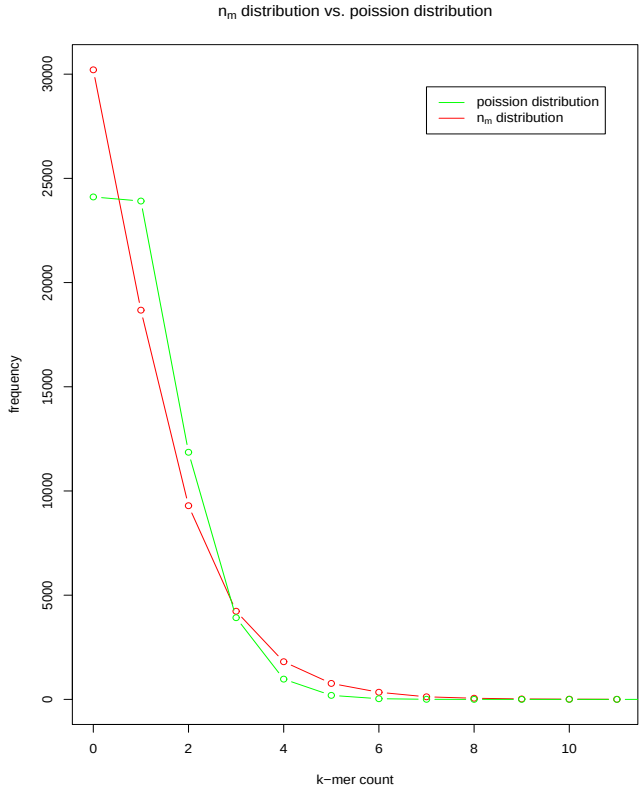


Figure 1: histogram of the n_m, compare with Poission distribution

Table 5: Cumulative genome cover according to the n_m, G=65K, $\ell = 8$

m=0	$r_0 = 0.000$	m=4	$r_4 = 0.884$	m=8	$r_8 = 0.995$
m=1	$r_1 = 0.289$	m=5	$r_5 = 0.943$	m=9	$r_9 = 0.997$
m=2	$r_2 = 0.576$	m=6	$r_6 = 0.974$	m=10	$r_{10} = 0.999$
m=3	$r_3 = 0.771$	m=7	$r_7 = 0.988$	m=11	$r_{11} = 1.000$

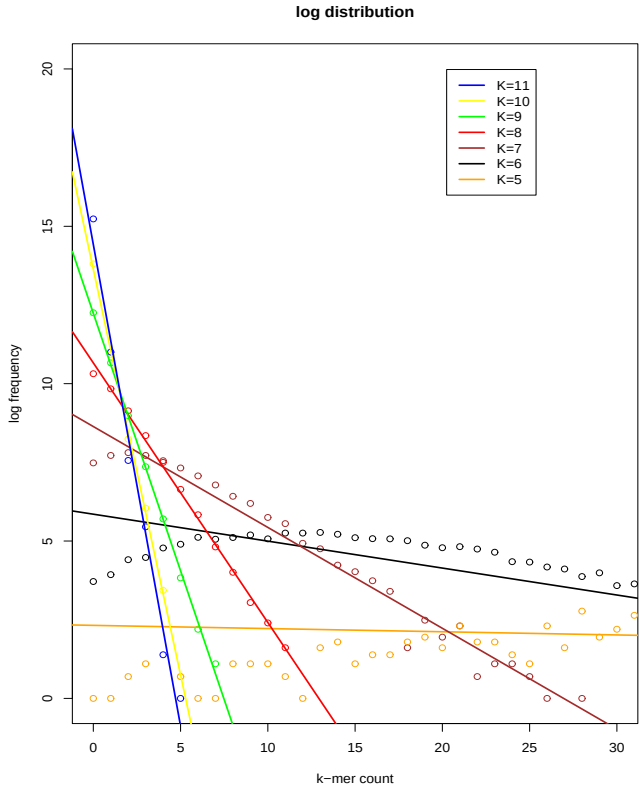


Figure 2: linear regression between $\ln(n_m)$ and m

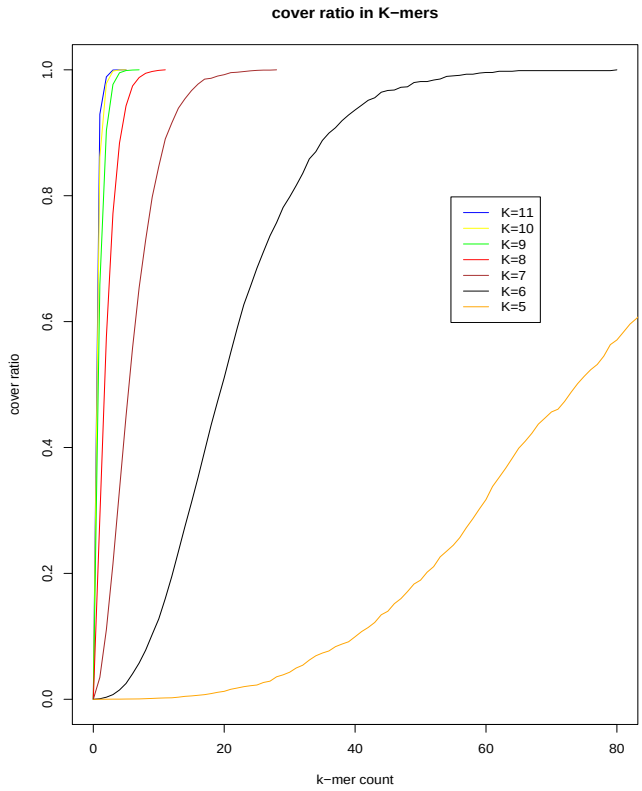


Figure 3: cumulative curve for the genome cover ratio according to n_m

3 ν_k distribution

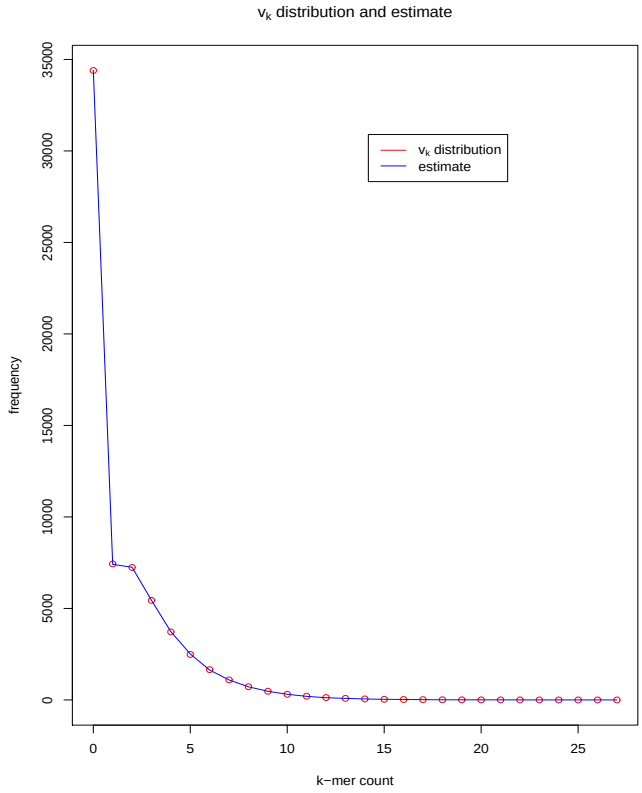


Figure 4: ν_k distribution, 2X

Table 6: Cumulative ratio according to the ν_k, $\ell = 8$, $c_0 = 2X$

k=0	$r_0 = 0.000$	k=5	$r_5 = 0.635$	k=10	$r_{10} = 0.931$	k=15	$r_{15} = 0.989$
k=1	$r_1 = 0.072$	k=6	$r_6 = 0.730$	k=11	$r_{11} = 0.952$	k=16	$r_{16} = 0.993$
k=2	$r_2 = 0.212$	k=7	$r_7 = 0.804$	k=12	$r_{12} = 0.966$	k=17	$r_{17} = 0.995$
k=3	$r_3 = 0.370$	k=8	$r_8 = 0.860$	k=13	$r_{13} = 0.977$	k=18	$r_{18} = 0.997$
k=4	$r_4 = 0.514$	k=9	$r_9 = 0.901$	k=14	$r_{14} = 0.984$	k=19	$r_{19} = 0.998$

4 determine n_m from ν_k

Table 7: Estimate the c and G from ν_k

coverage	initial c	output $\hat{c}$	output $\hat{c}_0$	G	$\sum(\nu_k - \hat{\nu}_k)^2/\nu_k$	real c	resolution
2X	4.8	1.71	2.14	60285.6	6.25	1.99	0.001
4X	4.8	2.52	3.15	68504.0	104.3	3.33	0.001
6X	10.0	4.15	5.20	66138.3	359.6	5.23	0.01
10X	10.0	6.48	8.10	73590.1	99.2	9.14	0.01
15X	15.0	9.48	11.85	76443.6	1986.8	13.98	0.01
20X	20.0	12.4	15.5	77874.9	4515.4	18.8	0.01

5 Expectation-maximization Algorithm

We have a density function $p(x|\theta)$, we have the likelihood funtion as:

$$\begin{aligned} \log P(x|\theta) &= \log P(x_1, x_2, \dots x_k|\theta) = \sum_k \log P(x_i|\theta) && (1) \\ &= \sum_k \sum_m P(y_m|x_i|\theta^t) \log P(x_i, y_m|\theta) - \sum_k \sum_m P(y_m|x_i, \theta^t) \log(y_m|x_i, \theta) && (2) \end{aligned}$$

$$Q(\theta|\theta^t) = \sum_k \sum_m P(y_m|x_i, \theta^t) \log P(x_i, y_m|\theta)$$

where we have

$$\begin{aligned} P(x_k|\theta) &= \sum_m \alpha_m P_{k,m,\theta} && (3) \\ P(x_k, y_m|\theta) &= \alpha_m P_{k,m,\theta} && (4) \\ P(y_m|x_k, \theta) &= \frac{P(x_k, y_m|\theta)}{P(x_k|\theta)} = \frac{\alpha_m P_{k,m.\theta}}{\sum \alpha_m P_{k,m.\theta}} && (5) \end{aligned}$$

and where $\alpha_m = \frac{n_m}{n}$, so we have the $Q(\theta|\theta^t)$,

$$Q(\theta|\theta^t) = \sum_k \sum_m \frac{\alpha_m P_{k,m,\theta^t}}{\sum_m \alpha_m P_{k,m,\theta^t}} \log(\alpha_m P_{k,m,\theta}) \quad (6)$$

for the M-step, we are going to computes the paramters maximizing the likelihood found on the E-step

$$\theta^{(t+1)} = \arg\max_\theta Q(\theta|\theta^{(t)})$$

let

$$\frac{\partial Q}{\partial \theta} = 0$$

so we have

$$\frac{\partial Q}{\partial \theta} = \sum_k \sum_m b_{k,m}[k\frac{1}{\theta} - m] = 0 \tag{7}$$

and

$$\theta = \frac{\sum_k \sum_m k b_{k,m}}{\sum_k \sum_m m b_{k,m}} \tag{8}$$

where

$$b_{k,m} = \frac{\alpha_m P_{k,m,\theta^t}}{\sum_m \alpha_m P_{k,m,\theta^t}}$$

and for

$$\sum_k \sum_m k b_{k,m} = \sum_k \sum_m \frac{\alpha_m P_{k,m,\theta^t}}{\sum_m \alpha_m P_{k,m,\theta^t}} * k = \sum_k k \tag{9}$$

$$\sum_k \sum_m m b_{k,m} = \sum_k \sum_m \frac{\alpha_m P_{k,m,\theta^t}}{\sum_m \alpha_m P_{k,m,\theta^t}} * m = \sum_k \frac{\sum_m m\alpha_m P_{k,m,\theta^t}}{\sum_m \alpha_m P_{k,m,\theta^t}} \tag{10}$$

so we have the θ's maximum likelihood estimate:

$$\theta = \frac{\sum_k k}{\sum_k \frac{\sum_m m\alpha_m P_{k,m,\theta^t}}{\sum_m \alpha_m P_{k,m,\theta^t}}} = \frac{\sum_k kP(x_k)}{\sum_m m\alpha_m} \tag{11}$$

the iterate equation become:

$$\theta^{(t+1)} = \frac{\sum_k k}{\sum_k \frac{\sum_m m\alpha_m^{(t)} P_{k,m,\theta^t}}{\sum_m \alpha_m^{(t)} P_{k,m,\theta^t}}} = \frac{\sum_k kP(x_k)}{\sum_m m\alpha_m^{(t)}} \tag{12}$$

$$\alpha_m^{(t+1)} = \sum_k \frac{\alpha_m^{(t)} P_{k,m,\theta^{t+1}}}{\sum \alpha_m^{(t)} P_{k,m,\theta^{t+1}}} P(x_k) \tag{13}$$

6 slight modification

observation function we defined an observe fuction,

$$O_\nu = \sum_k (\nu_k - \hat{\nu_k})^2$$

to measure the goodness of fit for the ν_k, and for the above EM algrithm, we also consider the value of the O_ν , and choose the O_ν as small as possible.

skip out local maxima Another technology we improve is to consider a very small random c, as a disturb during the iteration, and define a sight probability, about 0.0005 in our experiment, let the c choose randomly, to skip some local maxima.

Table 8: updated estimate the c and G from ν_k

coverage	output $\hat{c}$	output $\hat{c}_0$	G	$\sum(\nu_k - \hat{\nu}_k)^2$	Qn
2X	1.62	2.02	63812.2	96.5	78.6
4X	3.25	4.06	63644.9	639.1	139.1
6X	4.81	6.02	64479.0	553.8	190.9
10X	8.01	10.02	64591.7	505.5	234.9
15X	12.0	15.0	64383.1	764.0	184.3
20X	16.0	20.03	64603.8	867.1	381.0

Table 9: more simualte experiment for estimate the c and G from ν_k

experiment	coverage	K	output $\hat{c}$	output $\hat{c}_0$	G	$\sum(\nu_k - \hat{\nu}_k)^2$	Qn
rice 5M	50X	15	28.1	35.2	4.93M	17689.7	1575.9
rice 10M	20X	15	11.2	12.9	9.86M	323813	-2445.9
E coli 1,7M	10X	15	6.0	7.5	1.72M	6706.4	-7410.7

more experiment We are do more simulate experiment for the c and G estimate.

haploid and diploid We also simulate the 25X E.coli 65K segment, with diploid mode, without sequence errors, the estimate result show the $G = 130K$, $c_0 = 12.4$, the effective is same as the 130K sequences with the haploid mode, so there is the question could we identify the dipoid mode from the haploid mode in some ways?

7 the influence from the sequencing errors

We begin analysis the sequence errors cases, and we simulate the reads with the program wgsim, we set the base error rate 0.02, the figure 5 show the difference of ν_k distribution between the error containing and the non error containing, with the coverage is 25X. in ideal model, assume the rate of per base error is α, the reads length is L, and the original k-mer count for the word w_i is $x(w_i)$, the word count with error is $x'(w_i)$, so the relationship between $x'(w_i)$ and $x(w_i)$ is

$$x'(w_i) = \sum x(w_j)P(w_i|w_j)$$

where

$$P(w_i|w_j) = C_L^k \alpha^k (1-\alpha)^{L-k} \approx Pois(k, \alpha L)$$

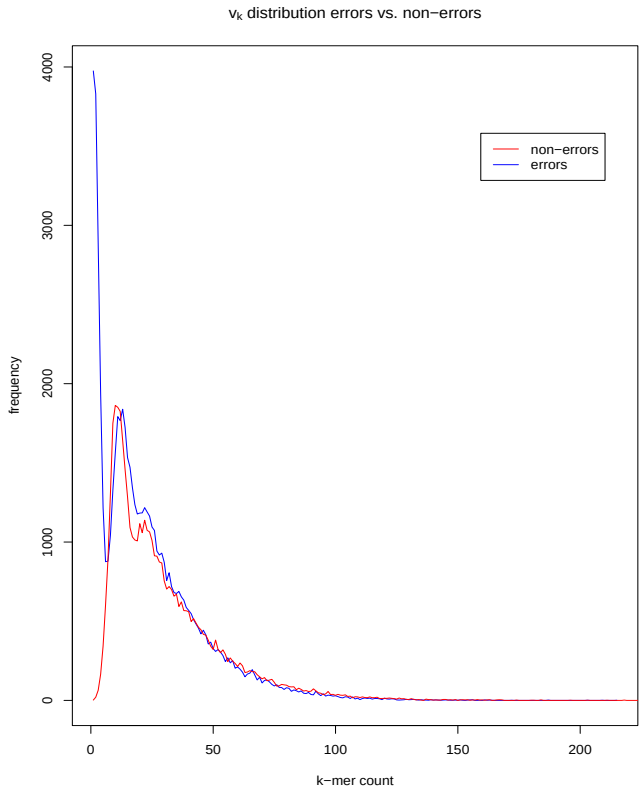


Figure 5: ν_k distribution with error and non-errors, 25X

if w_i and w_j contain k mismatch. Indeed, this framework is not practice, for we have to calculate an awful matrix inverse, so we have to try other approximate methods. In practice , we use the real sequencing data, which at first do an error correction stage, then to estimate the genome size according to its k-mer frequency table.
Experiment 1, ANT genome (error corrected), unknown
K: 25-mer, genome size: 233M, coverage: 2.40, Qn: 254.5, unique K-mer: 226M(96.8%)
Experiment 2, rice 5M simu, 50X
K: 15-mer, genome size: 5.1M, coverage: 48.8, Qn: 1642.9, unique K-mer: 3.9M(76.4%)
Experiment 3, rice 10M simu, 20X
K: 15-mer, genome size: 9.8M, coverage: 20.1, Qn: -441.6, unique K-mer: 7.2M(73.3%)
Indeed, the ANT genome is about 300M, the large bias is mainly cause by the sequencing errors, and here we notice the difference between objection function O_v and Qn as the maximum likelihood is much effective when there are sight disturbing, so in this version, we update mainly treate the Qn then the O_v

the model for sequencing errors Let a random variable Z=X+Y, where X is the variable with the no sequencing errors, and Y is the variable with the sequencing errors, and as we known the

above section, the density function of X is

$$P(X = k) = \sum_m \alpha_m Pois(k, mc) \tag{14}$$

and the density function of the Z can be generate a convolution formula

$$P(Z = i) = \sum_k P(X = k)P(Y = i - k) \tag{15}$$

indeed, the most important thing is to make the sequencing error model clear, is the model can be treate as the normal distribution?

summary of the tables and figures We return to the above tables and figures and to explain what the tables and the figures for. the data in Table 1 and Table 2 is to check the initial c for the iterate begining, and we are test the result on different coverages simulate data, such as under 2X – 20X data, in order to find the proper initial c, and the result show the simplely estimate with $2\nu_2/\nu_1$ will highly influnced if the coverage is large and thevalue ν_2 and ν_1 become too small, and table we improve the c using the $k\nu_k/\nu_{k-1}$, the result show if we choose the proper k, the initial c estimate is good enough, the only problems is how shall we choose this proper k when the unkown c. and in table 1 and table 2, we notice the symbol $\bar{c}$ and $\bar{c_0}$, indeed, $\bar{c} = 2\nu_2/\nu_1$ and $\bar{c_0} = \bar{c}L/(L-\ell+1)$,there the $L = 35$ and $\ell = 8$ in both table, and the values of ν_2 and ν_1 is the means of the 10 samples ν_2 and ν_1, for this has elimate the sampling disturbing effect.

why look into the $\{n_m\}$ As the initial stage, we have also to initial n_m besides the intial c, in practice, there are two process we have to determined for the n_m. First, the upper boundary number of the m, i.e what is proper for us to choose the largest m, if we choose the approximate boundary M, where M is the upper boundary of m; second, is n_m could be described by a certain distribution, if so, we could generate a good initial at the beginning.
Table 3 show the ℓ-mer distribution for the E.coli 65K segement, and we notice the $M = 11$ in this example, meanwhile table 5 show the cumulative cover ratio for the n_m, that $r_m = \sum_k^m kn_k/G$, where G is the genome size. The table 5 show till m=8, the $r_8 = 0.995$, it is a good enough approximate for the genome size estimate. Figure 3 show the genome cover ratio curve under 5-11 ℓ-mers, this figure indicate we could control the certain m_0 make the r_{m_0} good approximate by adjust the ℓ-mer value. In the practice, we control the M between 10 to 15, for the n_m initiation.

compare with Poission Figure 1, compare the $\{n_m\}$ with the Poission distribution $TPois(m, \gamma)$, where $T = 4^\ell$, $\gamma = G/T$, and the figure 1, red line is the n_m freqeuncy distribution and the green line show the Poission estimation, the result show the distribution of $\{n_m\}$ are large bias with the Poission distribution estimate, which indicate that the ℓ-tuples are not satisfied the e.i.i.d assumption, they should be described with a n.i.i.d model instead.

linear regression fittness We then consider the relationship between $\log(n_m)$ and m, and found they could regression with a linear,and we test the different ℓ-mer , let ℓ from 5 to 11 under the 2X data, found the large ℓ choose, the better the linear relationship, which indicate that in E.coli 65K segment, the n_m frequency followed with exponent law. Table 4 list the regression paramters a and b, as well as the p-values of these two parameters, which indicate that the $\ln(n_m)$ could be describe as a linear equation $\ln(n_m) = am + b$. For example, if $\ell = 8$, we have $n_m = e^{-0.825m+10.661}$

fit ν_k from n_m Figure 4 show fitness of the ν_k frequency and the ν'_k estimate from the n_m, the only purpose is to show the goodness of fit for the ν_k, which indicate that even in the 2X dataset, the estimate of $\nu_k = \sum_m n_m Pois(k, mc)$ is good enough, indeed in latter process, we defined a obervation function O_v to measure the goodness of the fit.
Table 6, show the cumulative cover ratio for the ν_k, which indicate what ratio for the $\sum k\nu_k$ theamong whole ℓ-mer, where $r_k = \sum_{i=0}^{k} i\nu_i / \sum_{i=0}^{K} i\nu_i$, where K is the upper boundary for the k reached, in practice , our ℓ-mer program limit the K upper boundary as 255 for the memory saving, and this process will show us how chould we choose the upper boundary for a good approximate estimate for the $\sum_i^K i\nu_i$. For example, $\ell = 8$, 2X data, when $r_{15} = 0.989$ means when we consider till 15, there arecontain about 0.989 ℓ-mer in the sampling, which make some sense on ν_k if approximate is necessary.

three result with alternative terminal condition Table 7, the first iterate result for the c and G estimate, and we test this result on the simulate 2X, 4X, 6X, 10X, 15X, 20X data respectively, $\ell = 8$, $L = 35$, and in Table 7, For example, for 2X data, we choose the initial c=4.8 as the intial input, then iterate to calculate the c_k and G_k, if the $|c_k - c_{k+1}| < \delta$, the iterate terminal. So the output $\hat{c} = 1.71$, $\hat{c_0} = cL/(L - \ell + 1) = 1.17 \times 35/(35 - 8 + 1) = 2.14$, the real c item indicate the real coverage value for the simulate data, which is 1.99, and the last item resulution is the threshod value of the δ, where we let the $\delta = 0.001$ for 2X, 4X data, while $\delta = 0.01$ for 6X, 10X and 15X
till Table 8 and Table 9, our program has been updated for a while, there has been more complete in aspect of EM algorithm deriving, local maxima avoiding and new obversation function as mentioned in above section. As a example of the Table 8 result, we iterate the c and n_m, with the terminal condition keep the O_v as small as possible, and we also using the slight disturbing during the iterate to avoid the local opitmal, and the reuslt show the avoid local optimal stategy is more effective when the coverage is small. Table 9 following the strategy but only do more experiment on rice 5M, 10M and E.coli 1.7M segments.These experiment result the genome size is acceptable but the coverage estimate are a little bias, by the way, the Qn value is the likelihood ofthe $Q(\theta|\theta^t)$
The last experiment , we begin to consider the sequencing errors and found out that the sequencing error indeed influncing the genome size esimate seriously, and we also notice the pitfall of the observation function O_v, which may lost function if the sampling ν_k is disturbing. So in this stage, we return the observation function as the make the Qn maxima first then the O_v, and the experiment rice 5M and 10M result support a good effective, though the error containing still badly bias the genome estimate.
Figure 5 show the different v_k distribution between the error containing and non-error containing from the simulate program sgwim, which make the base error rate 0.02, and we notice in low

index, the low ℓ-mer frequencing become much higher in error containing. So till now, an effective sequencing error elimate model in emerge necessary.

initial $\{n_m\}$ besides the inital c, we have to inital $\{n_m\}$ at the beginning, in our program, we inital the $\{n_m\}$ with a uniform distribution, which make $n_{m_0} = T/M$, where $T = 4^\ell$ and M is the upper boundary of the m. For we have try the SVD technology for the n_m initial process, and found the SVD very easy get the negative value of the n_m, even let the M lowwer to 3, and in practice, we usually fixed the M about 10-12, it is a little unstable for the SVD, so in current version, we didn't choose the SVD result for n_m initiation.

another method for the initial c determine As the formula $\nu_k = \sum n_m Pois(k, mc)$, then we can next have another formula

$$c\frac{\partial \nu_k}{\partial c} = \sum n_m \frac{(mc)^k}{(k-1)!} e^{-mc} - \sum n_m \frac{(mc)^{k+1}}{k!} e^{-mc}$$

that

$$c\nu_k'(c) = k\nu_k - (k+1)\nu_{k+1} \tag{16}$$

$$c^2\nu_k'' + c\nu_k' = k^2\nu_k - (k+1)(2k+1)\nu_{k+1} + (k+1)(k+2)\nu_{k+2} \tag{17}$$

then we noticed when the $\nu_k' = 0$, the ν_k will get the extremum, then we could choose the proper k that make sure the $\nu_k' = 0$, it's a better choices.

merge the complementation ℓ-mer in the real data, we notice a read maybe sequenced due to the the template or due to its complemetation, so we have to consider the ℓ-mer and its complemetation as one type word, for example, the 8-mer `aaggctgc` and `gcagcctt` should be recorded as one word.

further experiment There is also append some experiment on 173K human control bac, and we test is with wgsim simulate program, as will as the U0 real reads, and the result show the concide with each other, generate c=48.2 while G=176K, and we also notice the sequencing errors containing still influence the result mostly.
We also try the error containing model, there is a unconfirmed assumption that we assume the errors follows the uniform distribution to each ℓ-mer, that if the error ratio is P, then we have about total $V_m = P\sum k\nu_k$ is sequencing errors. The $\mu = V_m/T$, and it is a normal distribution $N(\mu, \sigma^2)$, could we make it clear and support by the data?

8 Estimate genome size with sequencing errors

the effective of the error correlation We begin test the effective the error correlation stage, and the first experiment are test on the human control bac 50X dataset, first error correlation, then mapping with SOAP, the statistic result show as follow. The output show the error correction stage

is quite effectively, the mapped ratio raised significantly, from 75.3% up to 99.8%, which indicate the most error reads have been removed after the correction stage. Another the effective of the error correlation stage is the increase of the 0 mismatch reads and reduce the 1 and 2 mismatch reads, which indicate a replacement process due to word frequency containing.

before error correlation

maximum length:	35
minimum length:	35
reads count:	234718
average length:	35
reads mapped:	176792
mapped ratio:	75.3%
0 mismatch:	139060
1 mismatch:	26998
2 mismatch:	10734

after error correlation

maximum length:	37
minimum length:	28
reads count:	179603
average length:	34.90
reads mapped:	179216
mapped ratio:	99.8%
0 mismatch:	176685
1 mismatch:	2000
2 mismatch:	531

Experiment: human control bac, corrected
OUTPUT: ℓ=15, genome size: 173K, coverage: 35.6, uni k-mer: 139K

model I we now test the error model from the Dr. Zheng's note section 6:

$$k\nu_k = (1-f)k\nu_k^0 + \delta(k+1)\nu_{k+1}^0, \quad k \geq 2 \tag{18}$$

$$\nu_1 = (1-f)\nu_1^0 + \sigma N \tag{19}$$

where f the probability that a word become a new word, $N = \sum_k k\nu_k$, $\sigma = f(1-\sum_k \nu_k)/4^\ell$, $\delta = (f-\sigma)/(1-\nu_1^0/N)$.
The wgsim program simulate data, 25X, with base error ratio $e = 0.02$, 15-mer so we have the parameters as: $f = 1-(1-e)^\ell = 1-(1-0.02)^{15} = 0.2614$, $\sigma = 0.2609$, $\delta = 0.00047$. Next figure show this simple err model not very fitness in our recent test.

model II Now we return to the equation (15), and also the uniform assumption for the errors, and the word changes probability f, the errors count distribution $P(Y=k)$ can be derived as

$$P(Y=k) = C_n^k p^k (1-p)^{n-k} = Pois(k, np); \tag{20}$$

where $np = fN/4^\ell$. then the no errors count distribution $P(X=k)$ can be described as

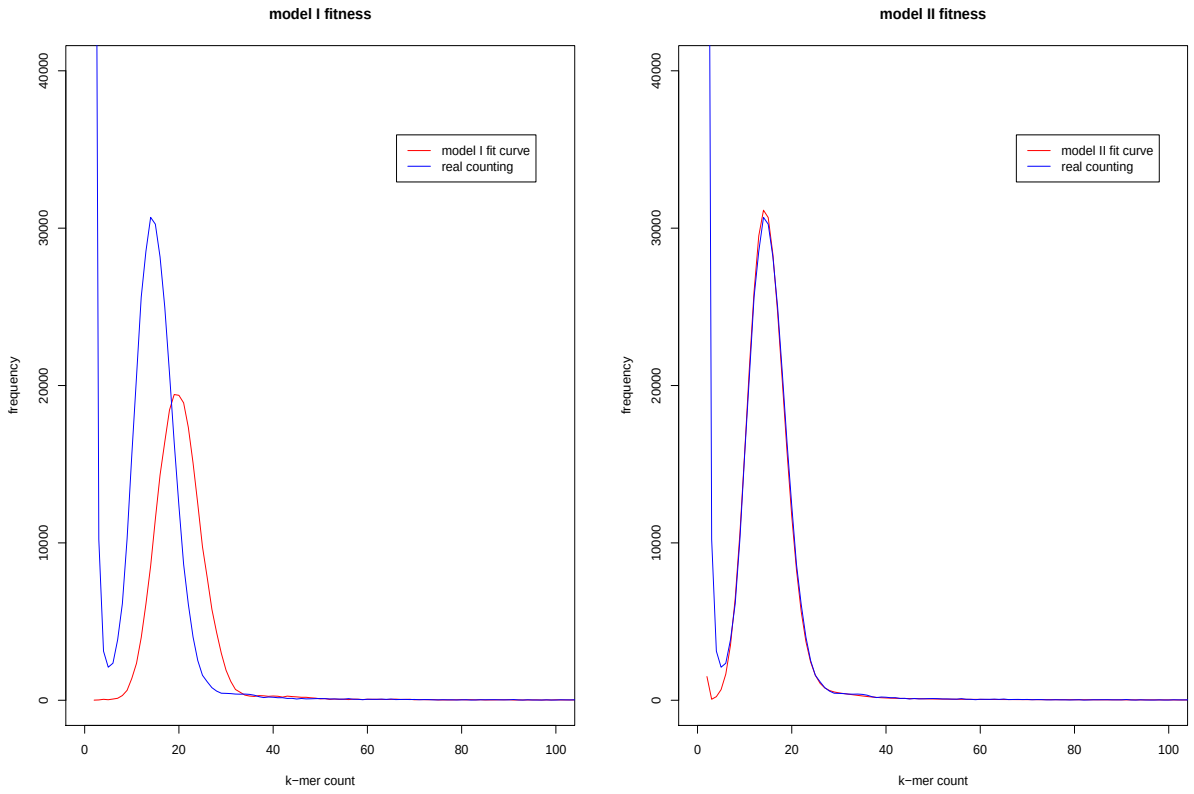
$$P(X=k) = \sum_m n_m Pois(k, (1-f)mc) \tag{21}$$

so the distribution of random varible Z can be describe as

$$P(Z=i) = \sum_k P(X=k)P(Y=i-k) \tag{22}$$

the output result show mostly is good, but in low frequency region still large bias. In practice, $f = 0.2614$, $\ell = 15$, in the figures (a) and (b), we removed the counting with $\nu_k = 0$ and $\nu_k = 1$, for these

two counting is too large for drawing. And the data show the $\nu_1^{m_1} = 1788840$ while $\nu_1^{r_1} = 1518530$, and the $\nu_1^{m_2} = 1788542$ while $\nu_1^{r_2} = 1518530$. We also notice the slight bias for the model II that, $\nu_2^{m_2} = 1505.074$, $\nu_3^{m_2} = 62.688$, $\nu_4^{m_2} = 228.458$, $\nu_5^{m_2} = 675.004$, $\nu_6^{m_2} = 1661.978$ vs. $\nu_2^{r_2} = 88178$, $\nu_3^{r_2} = 10216$, $\nu_4^{r_2} = 3120$, $\nu_5^{r_2} = 2100$, $\nu_6^{r_2} = 2354$. where $\nu_k^{m_1}$ indicate the estimate ν_k from model II, while m_1 indicate the model I, r_1: the real input data for model I, r_2: real input data for model II.



(a) (b)

The possible reason for model II's bias maybe the uniform assumption and the ℓ-tuple are not so independent in 4^ℓ distribution.

running with different initial c We are running the result with different initial c and find out the result are stable enough for the large changes of the c, where we using the human control bac simulate no errors reads with $\ell = 15$, iterate 5000 times, disturb ratio 0.05.

distribution of n_m & zipf's law From the figure 1 and the equation (20), we notice the poission distribution estimate are not so proper for the word frequency distribution describing. And in linguistic fields, there are an empirical law, zipf's law, *refers to the fact that many types of data studied in the physical and social sciences can be approximated with a Zipfian distribution.* It could be describe as

$$f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)} \tag{23}$$

Table 10: running the result with different initial c

initial c	genome size	output c	uni k-mer	Qn
0.01	167.9K	35.6	150.2K	526.6
G0.05	167.9K	33.7	150.2K	448.6
5	176.0K	35.1	137.2K	508.4
15	167.8K	36.4	150.2K	540.3
1000	167.7K	35.5	150.3K	533.4
5000	168.3K	33.6	150.0K	458.4

There are also some relative key words for zipf's law, such as **Yule-Simon distribution**, **harmonic series**, **Riemann zeta function**.

Estimate ν_k^0 from ν_k We noticed the distribution of ν_k, find most error distorted ℓ-mer are very low frequency, less than 5, though the exact distribution of the error ℓ-mer are difficult, we improve our EM models to estimate the ν_k^0 in [1, 5] and then approximate treate $\nu_k^0 = \nu_k$ when $k > 5$. So we update the iterate equation (12), that

$$\theta^{(t+1)} = \frac{\sum_{k\le 5} \hat{\nu}_k^0 + \sum_{k>5} \nu_k}{\sum_m m n_m^{(t)}} \tag{24}$$

where $\hat{\nu}_k^0 = \sum_m n_m^{(t)} P_{k,m,\theta}$.
Experiment: human control bac, 0.02 base error ratio, simu
$\ell = 15$, genome size 167.5K, coverage: 19.1, uni-kmer: 150.6K

Figure 6 show the estimate of ν_k and ν_k^0, compare with the no error containing counting. From figure 6, the dark line show the real counting with the distort tuples, the blue line is the estimate line, fit the ν_k, $k > 5$ and estimate the ν_k^0 when $k \le 5$ while red line indicate we adjust the effective coverage, to recover these distort tuples into no distort ones, so the effective coverage become $c^0 = c/(1-f)$, then compare the recover distribution with the real no errors situation, ν_k^0 (green line), and the result show these curve are match well. The low frequency $\ell-$mer: 874K, total ℓ-mer: 3.43M, the estimate $f = 0.2546$. (the theory $f = 0.2614$), the effective $c_e = 20.3$ and the true coverage estimate: $c^0 = 25.4$ compare the real data coverage 25X.

ν_k with ℓ Figure 7 show the nu_k distribution with the different ℓ-mer length ($\ell = 7, 9, 11, 13, 15$), and we simulate the no error data 25X for test, treating each pair reverse complement tuples as one word, and the figure 7 show when $\ell = 7$ most tuple are higher frequency, and there are near 5K tuple's frequency large than 255 and then uni 7-mer is nearly null.

re-distribution low frequency tuples We are re-distribute the low frequency (¡5) tuples to higher ones randomly, with the the formula

$$\nu_k^r = \nu_k(1 + V_L/V_H) \tag{25}$$

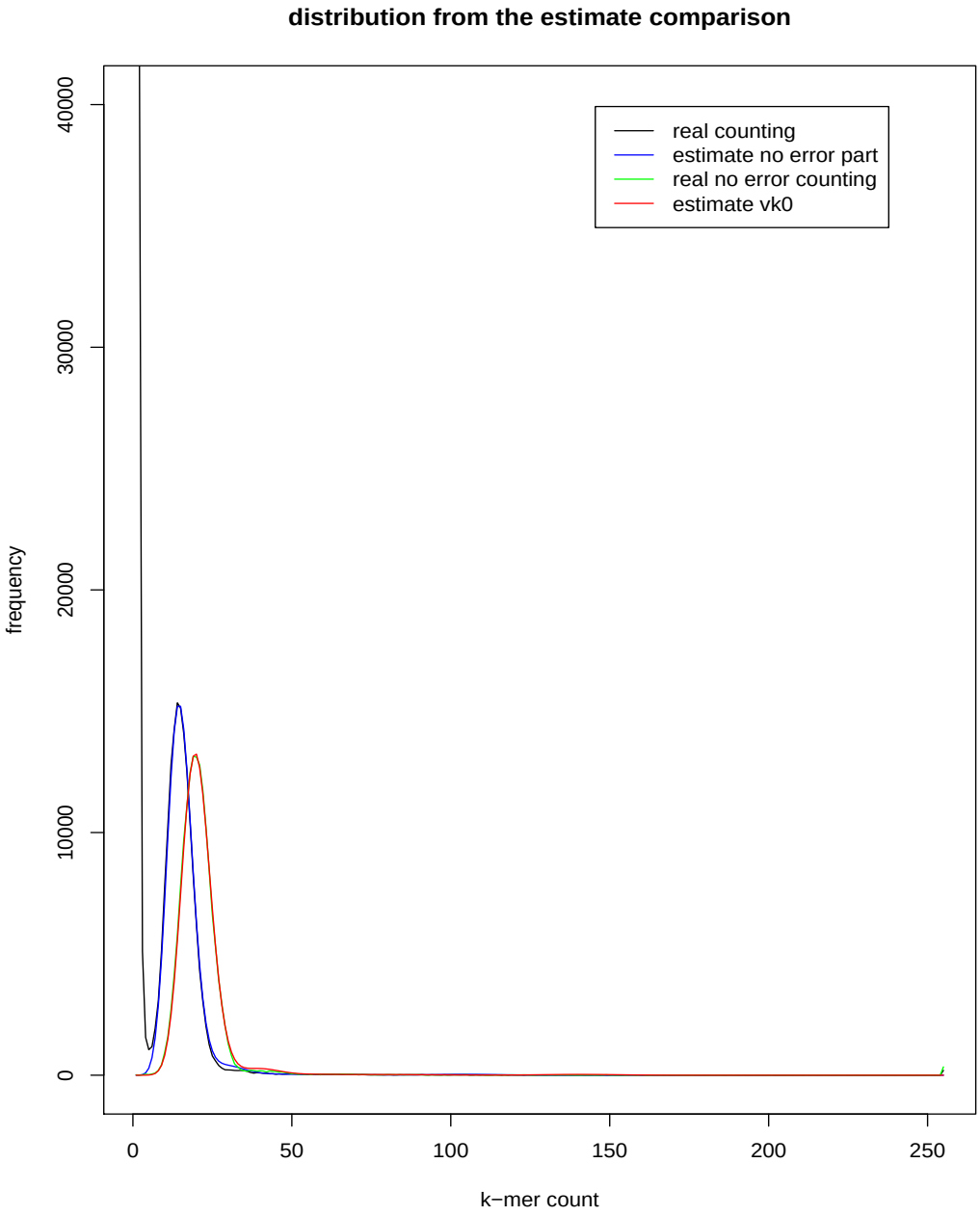


Figure 6: ν_k and ν_k^0

where we instead ν_k with the estimated $\hat{\nu}_k$ from Poissian, when $k \leq 5$. And V_L is $\sum_{K \leq 5} \nu_k$ and $V_H = \sum_{k>5} \nu_k$, where for a high c, the low frequency part is very small when there are no seqeuncing errors. The result show in Figure 8, there are a little bias between red and green lines.

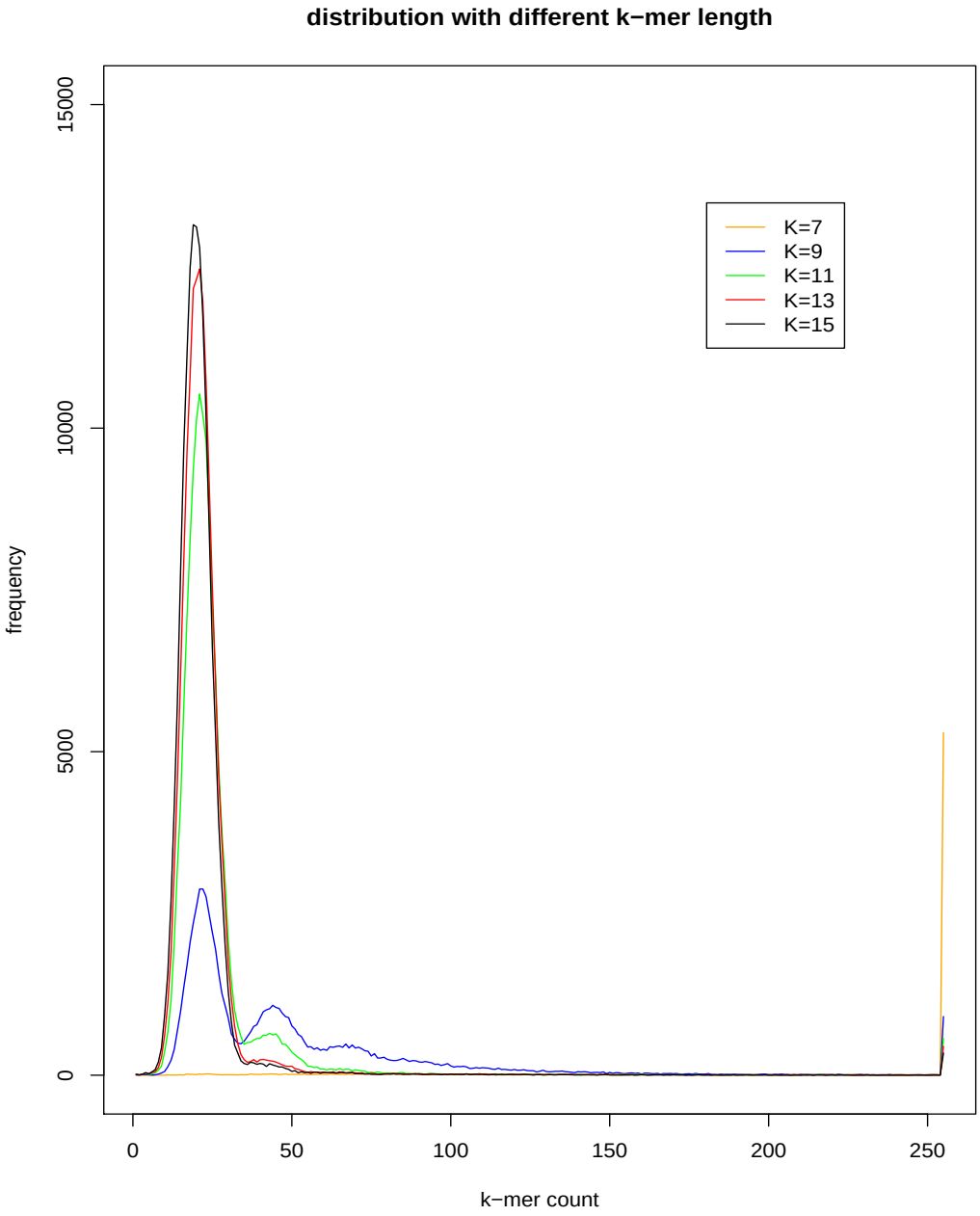


Figure 7: ν_k with different K-mer

real data experiment $\ell = 15$, genome size 237.8K, coverage: 24.1, uni-kmer: 78.4K, error ratio: 0.29, low freqeuncy tuples : 1.44M, total tuples: 4.88M

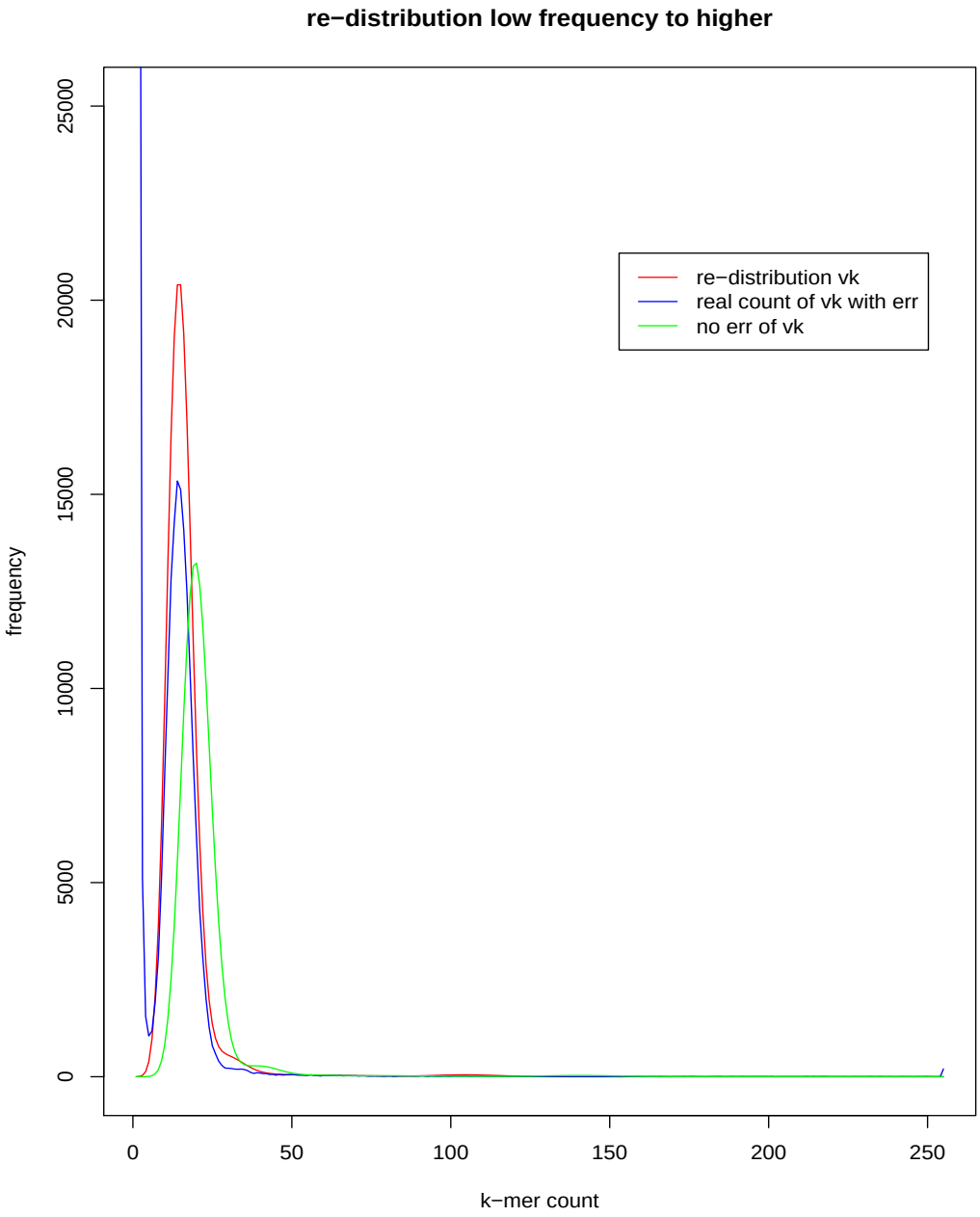


Figure 8: re-distribution lower frequency to higher