An ℓ-mer component distribution for genome size esimation

GAO SHAN, WEI-MOU ZHENG

July, 18, 2009

Abstract As the development of the high-throughput sequencing technologies, the genome size and the coverage are not always known before the de novo assembling. We developed a bayesian framework with an EM algorithm iterately estimate the genome size, which is elegant in mathematics as well as a good performance. The model first develop on the no sequencing error data set then extend to the sequencing errors containing. We test in various data set, from BAC to whole eukaryotic genome, the estimation of sequencing error ratio is an extra acquisition.

1 Introduction

As the development of the next generation sequencing technologies, including Roche 454, Illumina GA, and ABI SOLiD [1]. The sequencing cost reduced largely compare with the traditional Sanger capillary sequencer, and the third-generation sequencing technology, single-molecule DNA sequencing has been on the way [2]. There are more than 180 organisms have been sequenced since 1995, *Haemophilus influenzae*,the first genome release, and there are also hundreds organism in sequencing [3]. Usually, the accurate genome size are obtain only after the whole genome assembling, and there are also some biological technologies, a method using quantitative real-time polymerase chain reaction to estimate genome size without sequencing [4]. J.Raes et.al developed a conmputational approach measuring effective genome size of real environments using metagenomics data [5]. In 2003, li & Waterman first publish a ℓ-tuple based approach for the genome and repeat structure length estimation, which focus on the mixture Poission model with EM algorithm [6].
In computer fields there are largely application of the expectation-maximization (EM) algorithm, which is first given its name in a classic 1977 DLR paper [7]. It usually presents an iterative approach for maximum likelihood estimates when the observation can be viewed as incomplete data. It is also universal applid EM to Bayesian style, It finally gives a probability distribution over the latent variablesover the latent variables
In our study, alternative to the li & Waterman approach, we design a Bayesian estimation (BE) and EM iteration, to estmate the genome size. By taking consider the distribution of the ℓ-tuple, we could infer the component of ℓ-tuple of the genome. We at first consider the simulate data set without sequencing errors, and then extend the algorithm to deal with the error containing situation. We also test the result from BAC to eukaryotic whole genome size, the output result preform well.

2 Theoretical analysis for genome size estimation

2.1 notation

w: ℓ-mer or 'word'; $n(w)$: number of occurrences on a sequence; G: genome size; L: The read length; N: reads number in a sample; c: The effective coverage rate, $c = N(L-\ell+1)/(G-L+1)$. $x(w)$: the occurrence number of w in the sample; $I_k(w)$: the indicator of $x(w) = k$. $I'_m(w)$: the indicator of $n(w) = m$; $\nu_k = \sum_w I_k(w)$, the ℓ-mer count from the sample; $n_m = \sum_m I'_m(w)$, the ℓ-mer count from the genome.

2.2 Statistics related to ν_k

The distribution of ν_k could be presented with the compound Poission model, the distribution n_m to ν_k is given by Binorm$(n_m, p_{k,m})$, where

$$p_{k,m} = \frac{e^{-mc}(mc)^k}{k!} \tag{1}$$

which is the probability $Pr(w \in \sigma_k | n(w) = m)$ for an ℓ-mer with $n(w) = m$ to be sampled k times. The distribution probability of ν_k should be Binom$(n, \sum_m n_m p_{k;m}/n)$ with $n = \sum_m n_m$. The mean

$$< \nu_k >= \sum_m p_{k,m} = n_m \frac{e^{-mc}(mc)^k}{k!} = \frac{c^k}{k!} \sum_m \eta^m m^k n_m \quad \eta \equiv e^{-c} \tag{2}$$

When the sample size is large, the observed ν_k is a good estimation of $< \nu_k >$.

2.3 A Bayesian estimation (BE) of n_k

[The representation here is quite different from that of Li & Waterman (LW). Viewing $\{n_m\}$ as unobserved latent variables and {mc} as likelihood parameters, LW approached the problem from an EM-algorithm. However, among {mc} only a single parameter c is independent. (See next section for EM.)] The joint probability for w to have n(w) = m and x(w) = k is

$$Pr[n(w) = m, x(w) = k] \equiv p(m,k) = p(m)p(k|m) = \frac{n_m}{n} p_{k,m}$$

Thus,

$$p(m|k) = \frac{p(m,k)}{\sum_{m'} p(m',k)}, \; \hat{n}_m = \sum_k p(m|k)\nu_k,$$

where $\hat{n}_m$ is the estimate n_m. After initial triable c and $\{n_m\}$ are given, we may iteratively update them as follows.

$$\{n_m\}, c : c \to p(k|m) \to p(m,k) \to p(m|k) \Rightarrow \{\hat{n}_m\}, \hat{c}$$

2.4 Expectation-maximization Algorithm

It is used in statistics for finding maximum likelihood ($L(\theta x)$) estimates (MLE) of parameters (θ) in probabilistic models, where the model depends on unobserved latent variables (Z). EM iteratively alternates two steps: an E-step, which computes an expectation of the log likelihood with respect to the current estimate of the distribution for the latent variables Z

$$P(\theta|\theta^{(t)}) = E_{Z|x,\theta^{(t)}}[\log L(\theta; x, Z)|x, \theta^{(t)}],$$

and an M-step, which computes the parameters maximizing the likelihood found in the E-step

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

For the problem here, the parameter θ is vector λ_m for Poissonian of n_m. The lanent variables are the indicate I_w. then $\log L(x,\theta) \sim \sum_w \sum_m \sum_k I_w(m) P(m|k,\theta)[\log P(k,m|\theta) - \log P(m|k,\theta)]$.

2.5 Deriving from the likelihood

Indeed, the $\{\lambda_m\}$ only one independent parameter c, so if we treate θ with c, We have a density function $p(x|\theta)$, we have the likelihood funtion as:

$$\begin{aligned}
\log P(x|\theta) &= \log P(x_1, x_2, \ldots x_k|\theta) = \sum_k \log P(x_i|\theta) && (3)\\
&= \sum_k \sum_m P(y_m|x_i|\theta^t) \log P(x_i, y_m|\theta) - \sum_k \sum_m P(y_m|x_i, \theta^t) \log(y_m|x_i, \theta) && (4)
\end{aligned}$$

then, $Q(\theta|\theta^{(t)})$ could be write as:

$$Q(\theta|\theta^t) = \sum_k \sum_m \frac{\alpha_m P_{k,m,\theta^t}}{\sum_m \alpha_m P_{k,m,\theta^t}} \log(\alpha_m P_{k,m,\theta}) \quad (5)$$

where $\alpha_m = n_m/n$. let

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

so we have the likelihood estimation of θ is

$$\theta = \frac{\sum_k k\nu_k}{\sum_m m n_m} \quad (6)$$

the iterative equation become:

$$\theta^{(t+1)} = \frac{\sum_k k\nu_k}{\sum_m m n_m^{(t)}} \quad (7)$$

$$n_m^{(t+1)} = \sum_k \frac{n_m^{(t)} P_{k,m,\theta^t}}{\sum n_m^{(t)} P_{k,m,\theta^t}} \nu_k \quad (8)$$

3 Genome size estimation

3.1 Estimate genome size under different coverage

We use a 65K E.coli segment to simulate different coverage reads for test, they are 2X, 4X, 6X, 10X, 15X, 20X, data respectively, and we use the $\ell = 8$, to estimate the genome size, the result show in the following table:

Table 1: running the result with different coverage

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

3.2 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.

Table 2: 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
0.05	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

3.3 Running with different ℓ

3.4 Running under different genome size

Then, we simulate the reads from the different genome size, to evaluate our method for genome size estimation,

Table 3: running the result with different coverage

ℓ	output $\hat{c}$	output $\hat{c}_0$	G	$\sum(\nu_k - \hat{\nu}_k)^2$	Qn
$\ell = 9$	23.3	30.2	150K	11781.3	964.9
$\ell = 13$	21.0	31.9	165K	7211.2	550.1
$\ell = 17$	19.3	25.0	168K	3144.3	245.8
$\ell = 25$	16.5	25.1	169K	3236.6	-420.2

Table 4: more simualte experiment for estimate the c and G from ν_k

experiment	coverage	ℓ	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

3.5 Estimate genome size with sequencing errors

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

$$\theta^{(t+1)} = \frac{\sum_{k \le 5} \hat{\nu}_k^0 + \sum_{k>5} \nu_k}{\sum_m m n_m^{(t)}} \tag{9}$$

where $\hat{\nu}_k^0 = \sum_m n_m^{(t)} P_{k,m,\theta}$.
Indeed, the data could be split into two independent dataset, the none error data and errors data, so the ν_k could be described as $\nu_k = \nu_k^e + \nu_k^0$, if the f is the error ratio of the ℓ-mer, then, the distribution of the $\nu_k^0 = \sum_m n_m Pois(k, (1-f)mc)$, and the ν_k^e, for a large k, the value of ν_k^e are near 0, that means the errors tuples are always locate in low frequency regions.
Experiment: human control bac, 0.02 base error ratio, simu
$\ell = 15$, genome size 167.5K, coverage: 19.1, uni-kmer: 150.6K

3.6 Practical application

Human control bac, raw solexa data:
ℓ 15-mer, genome size 237.8K, coverage: 24.1, uni-kmer: 78.4K, tuple error ratio: 0.29,
E.coli, raw solexa data
ℓ: 15-mer, genome size: 4.69M, coverage: 75.6, uni-kmer: 4.39M,tuple error ratio: 0.048,
cucumber, raw solexa data
ℓ: 17-mer, genome size: 299.8M. coverage: 34.9, uni-kmer: 90.1M, error ratio: 0.117,

ANT genome (raw data), unknown
ell: 25-mer, genome size: 233M, coverage: 2.40, Qn: 254.5, unique K-mer: 226M(96.8%)

4 Discussion

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.

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

$$c\nu_k'(c) = k\nu_k - (k+1)\nu_{k+1} \tag{10}$$

$$c^2\nu_k'' + c\nu_k' = k^2\nu_k - (k+1)(2k+1)\nu_{k+1} + (k+1)(k+2)\nu_{k+2} \tag{11}$$

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.

memory usage Our program are designed in two stage, first is the ℓ-mer counting, the next is the genome size estimate, the mainly memory'usage is the hash table for the ℓ-mers. In orignial version, wecontrol the memory with the ℓ-mer length, each tuples use 1byte storage, that means, for the $\ell = 17$, there are 16G memory usage, then for the longer ℓ, we has to choose hash function, named Jenkins' hash function, which performan well in practice, we could control the memory usage as personal need, usually, we could control the usage to 4G is good enough for even 25-mer situation.

5 Acknowledgements

References

[1] J. Shendure and H. Ji. Next-generation DNA sequencing. *nature biotechnology*, 26(10):1135–1145, 2008.

[2] T.D. Harris, P.R. Buzby, H. Babcock, E. Beer, J. Bowers, I. Braslavsky, M. Causey, J. Colonell, J. DiMeo, J.W. Efcavitch, et al. Single-molecule DNA sequencing of a viral genome. *Science*, 320(5872):106, 2008.

[3] RD Fleischmann, MD Adams, O. White, RA Clayton, EF Kirkness, AR Kerlavage, CJ Bult, JF Tomb, BA Dougherty, JM Merrick, et al. Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. *Science*, 269(5223):496, 1995.

[4] J. Gao and JG Scott. Use of quantitative real-time polymerase chain reaction to estimate the size of the house-fly Musca domestica genome. *Insect Molecular Biology*, 15(6):835–837, 2006.

[5] J. Raes, J. Korbel, M. Lercher, C. von Mering, and P. Bork. Prediction of effective genome size in metagenomic samples. *Genome Biology*, 8(1):R10, 2007.

[6] X. Li and M.S. Waterman. Estimating the repeat structure and length of DNA sequences using L-tuples, 2003.

[7] AP Dempster, NM Laird, and DB Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.