Breaking Lander-Waterman's Coverage Bound

D. Nashtaali, S.A. Motahari and B.H. Khalaj

Abstract

Lander-Waterman's coverage bound establishes the total number of reads required to cover the whole genome of size G bases. In fact, their bound is a direct consequence of the well-known solution to the coupon collector's problem which proves that for such genome, the total number of bases to be sequenced should be $O\left(G \ln G\right)$. Although the result leads to a tight bound, it is based on a tacit assumption that the set of reads are first collected through a sequencing process and then are processed through a computation process, i.e., there are two different machines: one for sequencing and one for processing. In this paper, we present a significant improvement compared to Lander-Waterman's result and prove that by combining the sequencing and computing processes, one can re-sequence the whole genome with as low as $O(G)$ sequenced bases in total. Our approach also dramatically reduces the required computational power for the combined process. Simulation results are performed on real genomes with different sequencing error rates. The results support our theory predicting the $\log G$ improvement on coverage bound and corresponding reduction in the total number of bases required to be sequenced.

I. Introduction

Data generated from DNA sequencing machines are growing at an unprecedented rate. Extracting knowledge from these data is extremely tedious and usually requires very powerful computing machines. The main reason is that the volume of data generated for an experiment usually contains redundant data and one needs to pay the price of extracting useful information and removing redundant information at the processing step. As an example, in the whole genome sequencing of Human genome with 100x coverage, each base averagely is present in 100 sequencing reads which means 99 percent of the data is redundant. The first question that comes in mind is whether the volume of data generated by the sequencing machines can be reduced without affecting the overall performance. In this paper, we focus on the whole genome sequencing problem and seek fundamental results on the redundancy level required to obtain the desired result.

The first fundamental result in this area has been due to Lander and Waterman in [1] where they present a lower bound on the number of reads, N, required to assemble the whole genome. We refer to this bound as *coverage bound*. The coverage bound states that for a genome of size G and reads of length L, at least $N_{cov} = \frac{G}{L} \ln\left(\frac{G}{L\epsilon}\right)$ reads are needed such that the whole genome is covered with a probability of no less than $1 - \epsilon$ [2]. Therefore, we should have $N \geq N_{cov}$.

For the aforementioned scenario, the total number of bases sequenced by the sequencing machine is NL, which requires to be of the order of $G \log G$, from Lander-Waterman's result. Consequently, the non-reducible redundancy level in such setup will be of the order of $\log G$. However, such result is based on the underlying assumption that sequencing and computing steps are performed independently, i.e., a machine takes samples from the genome and sequences as many reads required to cover the whole genome and then another machine processes the reads to assemble the genome. The overall architecture of such approach for whole genome sequencing is shown in Figure 1 (a) and consists of the cascade of two blocks one for sampling and sequencing and one for assembly.

Although such separation between sequencing and processing has been traditionally assumed in the literature, one can question whether such separation is fundamentally optimal with respect to amount of sequenced data which is generated and subsequently processed or not. In other words, can we improve the overall performance of such system by merging the two components? In fact, in order to verify the level of performance improvement achieved by such integration, one needs to answer the following two key questions. First, is there any improvement on lowering the redundancy of generated data by merging the two functions? Second, is it physically possible to build such a machine to perform both functions simultaneously? In the rest of this paper, we will try to answer the first question by proving that one can break the coverage bound of Lander and Waterman and reduce the number of

D. Nashtaali and B.H. Khalaj are with the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran. (nashtaali@ee.sharif.edu and khalaj@sharif.edu)

S.A. Motahari is with the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. (motahari@sharif.edu)

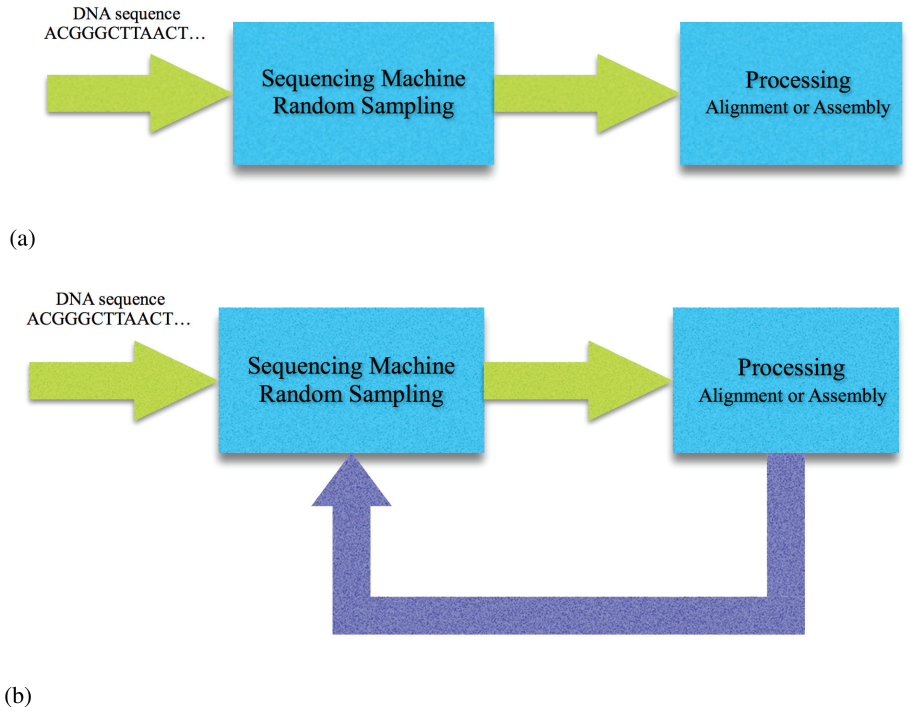

Fig. 1. Whole genome shotgun sequencing and our method platforms. In the classic method, the processing starts after termination of random sampling and sequencing. In our proposed sequencing platform, sampling and processing machines work cooperatively. The results of alignment are fed back to preempt reading redundant data in the sequencing machine.

sequenced bases to as low as $O(G)$. It is worth mentioning that even if we may not have access to a machine that can efficiently combine the sampling and computing units, the approach proposed in this paper can still reduce the computational power required to assemble the genome. This is due to the fact that the processing unit processes the data on the fly and if it detects that the information from the remaining part of data is redundant it will stop further processing of that piece of data. As we will show, such early termination can have significant effect on reducing computational complexity of the whole process. Answering the second question is in fact beyond the scope of this paper. However, our approach clearly shows that if a sequencing machine can be built that can preempt sequencing at the instant the computation part sees best fit, a much more efficient sequencing machine may actually be obtained.

Our strategy to merge the two functions is shown in Figure 1 (b) where the processing machine controls the sequencing machine by blocking sequencing of redundant bases. Hence, we assume that the sequencing machine sequences the DNA fragments base by base and it will stop sequencing a fragment once a blocking command from the processing machine is initiated for that fragment.

In this paper, we only focus on the re-sequencing problem in the processing machine. we first present theoretical results for i.i.d. genome and real genomes with given repeat structures with noiseless reads, and i.i.d. genome with noisy reads. Our simulation results performed on chr19 of Human genome hg19 for both noiseless and noisy reads. We have shown significant improvement on coverage bound for real genome is achieved by using this method. For processing machine, Meta-aligner [3] are used to align reads on the reference genome.

The longer the read lengths, more reduction on number of read bases will be achieved by our method. A number of Next Generation Sequencing (NGS) methods [4], such as PacBio [5] and Nanopore [6]-[8] already provide reads of several thousand bases long and are suitable candidates for such analysis. Figure 2 shows read length distribution of PacBio technology for the first two read archive in NCBI GenBank SRX533609 [9]. It can be envisioned that other sequencing methods might also provide longer reads as sequencing technology further advances in that direction in years to come.

The organization of this paper is as follows. Our basic method for i.i.d. and real genomes with noiseless reads, and i.i.d. genomes with noisy reads are discussed in the Methods section. The simulation results on real genome by considering different sequencing error rates are presented in the Results and discussion section. The last section

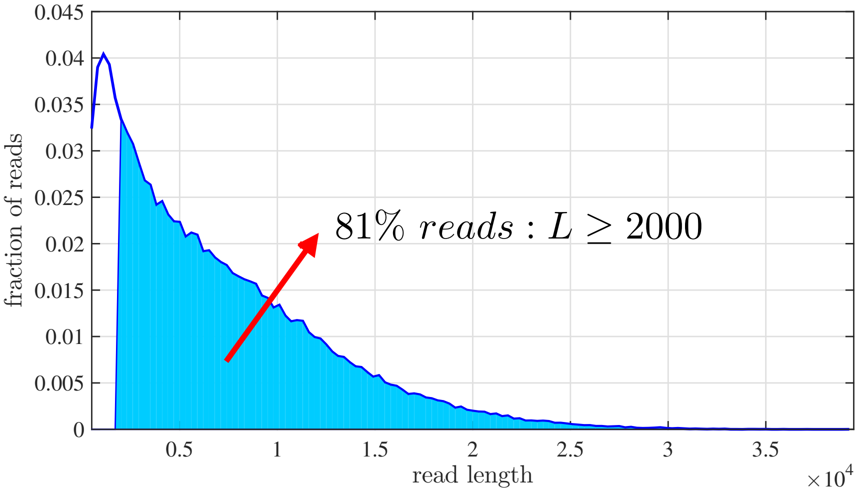

Fig. 2. PacBio read length distribution. Almost 81% of PacBio reads have length of at least 2000 bps.

is devoted to the conclusion and future works.

II. Methods

Conventionally, in the re-sequencing problem, a reference genome and a set of reads from a target genome are available at the processing step. In this framework, the sequencing machine produces reads of length L from N DNA fragments. Due to Lander-Waterman's coverage bound, NL, the total number of bases read by the machine, is required to be $O(G \log G)$.

Our method changes this bound by assuming that sequencing can be controlled by a processing machine such that it can terminate sequencing at any base. In other words, the sequencer starts reading bases of one side of a DNA fragment one by one and it will stop as soon as a command is initiated by the processing machine. In order to explain the basic ideas and to prove that sequencing can be performed efficiently, We first analyse our proposed methods on i.i.d. genomes. We extend the results to real genomes where repeats play an important role in the structure of the genomes.

A. I.i.d. genomes

In this part, we assume that the reference and target genomes are i.i.d. random sequences of $\{A, C, G, T\}$ with uniform base probabilities. We also assume that reads are sampled uniformly and independently from the target genome. Consider that reads are noiseless. The key strategy that we use to terminate the sequencing process in a controlled manner is as follows. We divide L by some integer number $K \in \{0, \dots, L\}$ and without loss of generality assume that $\ell = L/K$ is also an integer. We allow the reading machine to read the first ℓ bases of all the DNA fragments. Let $\mathcal{R}_1 = \{R_1(\ell), \dots, R_N(\ell)\}$ denote the set of starting ℓ bases of all the fragments. Here, $R_i(\ell)$ is the first ℓ bases of the i^{th} fragment.

After generating $\mathcal{R}_1$, all reads are mapped to the reference genome. Some of the the reads can be mapped uniquely to a location on the genome. We call such reads *anchored*. More precisely, a read R is assumed to be anchored if there is only one location on the genome with Hamming distance no more than $\alpha|R|$ where $|R|$ is the read length and α is some fixed constant.

After mapping, we partition the set $\mathcal{R}_1$ into three disjoint sets: $\mathcal{R}_1^C$ the set of reads that are anchored to some location on the genome and in addition, extending them does not increase the coverage, $\mathcal{R}_1^A$ the set of reads anchored to some location on the genome and in addition, extending them will increase the coverage, and $\mathcal{R}_1^F$ the set of reads that are not anchored in the first step. For a read $R_i(\ell)$ in $\mathcal{R}_1^C$ a termination command is initiated to stop further reading of the i^{th} fragment. The union of $\mathcal{R}_1^A$ and $\mathcal{R}_1^F$ is denoted by $\mathcal{R}_2$ that is the set of fragments where reading processes will be continued on them.

Subsequently, the next base of all fragments in $\mathcal{R}_2$ are read and we use the same procedure for mapping and termination. Therefore, at the end of this step, we end up with the set $\mathcal{R}_2^C$ of anchored and terminated fragments with length $\ell + 1$ and the set $\mathcal{R}_3$ that is used for extension in the next step. In this way, one can proceed to step $L - \ell + 1$ where all the fragments are extended to the maximum length L.

If we denote the set of reads that are uniquely mapped in the algorithm by $\mathcal{O}$, then $\mathcal{O} = \cup_{k=1}^{L-\ell+1} \mathcal{R}_k^C$. Our proposed algorithm is then detailed in Algorithm 1.

Algorithm 1

Input: N fragments with size L_i of a target genome plus a reference genome of length G.
Output: A set of reads $\mathcal{O}$, mapped to the reference genome.

Initiate:
Let $L = \max_i L_i$, $\mathcal{O} = \emptyset$ and $\mathcal{R}^A = \emptyset$. Fix ℓ (the sub-fragment's length), $\alpha \in [0, 1]$. Set $\mathcal{R}_1$ to be the set of all fragments.

```
for $k = 1$ to $L - \ell + 1$ do
    if k=1 then
        Sequence the first $\ell$ bases of all the reads in $\mathcal{R}_k$.
    else
        Sequence the $(\ell + k - 1)$-th base of all the reads in $\mathcal{R}_k$.
    end if
    Map all the reads in $\mathcal{R}_k$ to the reference genome with their last $\ell$ fragments and Hamming distance $d = \lfloor \alpha\ell \rfloor$.
    Add uniquely mapped reads in $\mathcal{R}_k$ to the set $\mathcal{R}^A$. Put the rest of reads in the set $\mathcal{R}_{k+1}$.
    Add reads in $\mathcal{R}^A$ to $\mathcal{O}$, if by further extensions they will not cover a new base on the reference genome.
    Add reads in $\mathcal{R}^A$ to $\mathcal{R}_{k+1}$, if by further extensions they will cover new bases on the reference genome.
end for
```

The two parameters of the algorithm, i.e., ℓ and α, should be specified based on the structure of reference genome as well as, G, N and L. One can choose ℓ to be as low as 1. However, the chance of finding uniquely mapped reads is very small for short reads resulting in much higher processing time. On the other hand, if ℓ is chosen to be large, then a lot of reads will overlap after the first step of the algorithm and we will sample a lot of redundant bases. Therefore, an optimal choice of ℓ is desired. The optimal value of ℓ depends on the size and repeat structure of the genome as well as the statistics of the variations between target and reference genomes. In the following, we describe selection of these parameters for noiseless reads.

It can be shown that for a given random DNA sequence of length G, the probability of observing two exact copies of a substring with length ℓ is lower than $G^2 4^{-\ell}$ [2]. Hence, a noiseless read of length $\ell > \log G$ from target genome almost certainly can be uniquely mapped to the reference genome. Conversely, a noiseless read of length $\ell < \log G$ from target genome will be mapped to at least two locations on the reference genome. Therefore, for noiseless reads with no variation between target and reference genomes, we can choose $\ell = \log G$.

The choice of α depends on the mapper quality and allowable Hamming distance between reads and the reference genome. If we assume a perfect mapper, the Hamming distance between a read and its true location is $\nu|R|$ where ν is the variations rate between target and reference genomes. Therefore, it suffices to set $\alpha = \nu$. In this scenario, we consider that variation between reference and target genome is negligible.

In order to evaluate the performance of the algorithm, we need to prove that the genome can be completely covered by the reads in $\mathcal{O}$ and the number of bases read more than once is small. After the first step of the algorithm, i.e. sequencing ℓ bases of all reads, all reads are anchored to their correct location on the genome with maximum Hamming distance of $d = 0$. This is due to the preceding discussion on random genomes where duplicate segments are scarce. We distinguish between two cases:

1) Two subsequent reads have common bases, such as i^{th} read and $(i + 1)^{\text{th}}$ read in Figure 3.
2) Two subsequent reads do not have a common base, such as j^{th} read and $(j + 1)^{\text{th}}$ read in Figure 3.

In the first case, we will encounter reads whose extension does not increase the coverage and therefore, their sequencing should be terminated. These kind of reads belong to $\mathcal{R}$. We call the bases common between i^{th} and

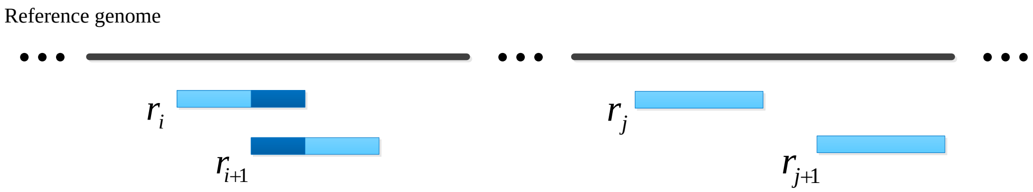

Fig. 3. Two possible cases for two subsequent reads. They are either disjoint or have some common bases.

$(i+1)^{\text{th}}$ read in such case, over-read bases. In the second case, we will encounter reads whose extension will increase the coverage. We put reads of this kind in $\mathcal{R}^A$, to be further extended in the following steps of the algorithm. Consequently, sequencing of an aligned read in $\mathcal{R}^A$ continues until it becomes a member of $\mathcal{O}$.

For further analysis, we assume that starting point of reads are a *Poisson* point process with rate $\lambda = \frac{N}{G}$. Hence, the inter-arrival times have independent *exponential* distributions.

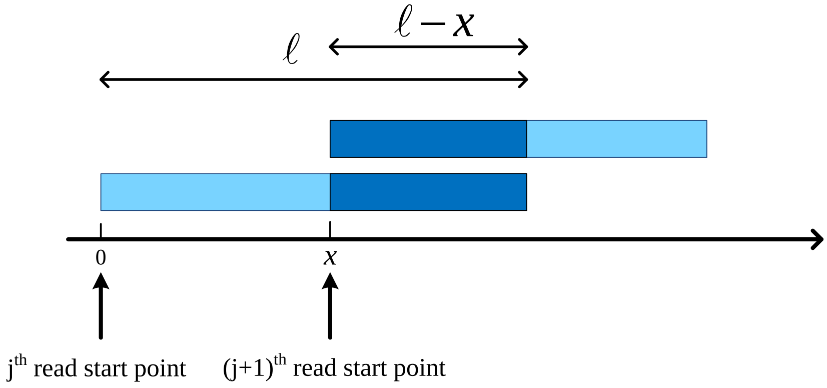

Fig. 4. A read of $\mathcal{R}_1^C$ and its adjacent read. The starting point of the second read is x bases after the starting point of the overlapped read from the right hand side (assuming that the two reads are of equal length ℓ).

Let $\mathcal{B}_i$ be a random variable representing the number of extra bases read by the sequencing machine for the i^{th} read. If $\mathcal{B}$ denotes the total number of over-read bases, then $\mathcal{B} = \sum_{i=1}^{N} \mathcal{B}_i$. Therefore, $\mathbb{E}\{\mathcal{B}\} = N\mathbb{E}\{\mathcal{B}_j\}$.

To compute $\mathbb{E}\{\mathcal{B}_j\}$, note that the next read of the j^{th} read starts x bases after the j^{th} read. If $x \leq \ell$, then $\mathcal{B}_j = \ell - x$. Otherwise, $\mathcal{B}_j = 0$, see Figure 4. Since x has an exponential distribution, we obtain

$$\mathbb{E}\{\mathcal{B}_j\} = \int_0^{\ell} (\ell - x)\, \lambda e^{-\lambda x} dx = \ell - \frac{1}{\lambda}\left(1 - e^{-\lambda \ell}\right) \triangleq V_1(\ell). \tag{1}$$

In the case of i.i.d. genomes, we set $\ell = \log G$. Hence, the average number of over-read bases of all reads is as follow:

$$\mathbb{E}\{\mathcal{B}\} = N \times V_1(\log G) = G\left(\kappa - 1 + e^{-\kappa}\right), \tag{2}$$

where $\kappa = N\frac{\log G}{G}$. For a constant value κ, the order of $\mathbb{E}\{\mathcal{B}\}$ becomes $O(G)$ and number of reads becomes $N = \kappa\frac{G}{\log G}$. To be able to cover the genome with N reads, we need to be able in closing gaps of at least $G \log G/N$ bases (from the coverage bound). Therefore, length of reads becomes $L = \frac{1}{\kappa}(\log G)^2$.

We observe that by tuning κ, one can control maximum read lengths as well as total number of bases read by the machine. For instance, by choosing $\kappa = 1$, we obtain $\mathbb{E}\{\mathcal{B}\} = 0.36G$ and the maximum read length becomes $L \approx 1000$ bps.

Next, we consider noisy reads with the error rate ϵ. Moreover, we assume that the target and reference genomes vary in their sequences with the variation rate ν. Before presenting our algorithm for noisy reads, we set a parameter denoted by C_ϵ for the coverage depth. The coverage depth helps in removing sequencing errors by averaging over several reads. The coverage depth is chosen such that if C_ϵ reads cover a base then it is possible to correctly recover that base with a given probability P_ϵ.

To compute C_ϵ, we choose a base as the target base if it has maximum vote amongst all other bases over reads covering that location. Let $\mathcal{I}_i$ denote the error event of incorrect calling of the i^{th} base. If the random variable C_i denotes the coverage of the i^{th} base, then error occurs if the corresponding base of more reads are incorrect base. Thus, the probability of error for the i^{th} base is,

$$\mathbb{P}\left\{\mathcal{I}_i | C_i\right\} = \sum_{i=0}^{\lfloor \frac{C_i - 1}{2} \rfloor} \binom{C_i}{i} (1-\epsilon)^i \sum_{j=i+1}^{C_i - i} \binom{C_i - i}{j} 3 \times \left(\frac{\epsilon}{3}\right)^j \left(\frac{2\epsilon}{3}\right)^{C_i - i - j} \tag{3}$$

Hence,

$$C_\epsilon = \arg\min_{C_i} \ \mathbb{P}\left\{\mathcal{I}_i | C_i\right\} \leq P_\epsilon.$$

For an example, if $P_\epsilon = 10^{-4}$ and $\epsilon = 0.07$, C_ϵ becomes 6. For having coverage depth of C_ϵ, it suffices to change the k^{th} step of Algorithm 1 as follows: we add a read in $\mathcal{R}^A$ to $\mathcal{R}_{k+1}$ when its extension will cover a base for c^{th} times on the reference genome, where $c \leq C_\epsilon$. Details of the proposed algorithm for noisy reads is presented in Algorithm 2.

Algorithm 2

Input: N fragments with size L_i of a target genome with G bases plus a reference genome with the same length.
Output: A set of reads $\mathcal{R}$, mapped to the reference genome.

Initiate:
Let $L = \max_i L_i$, $\mathcal{R} = \emptyset$ and $\mathcal{R}^A = \emptyset$. Fix ℓ (the sub-fragment's length), $\alpha \in [0, 1]$ and d_{max}. Fix C_ϵ.

```
for $k = 1$ to $L - \ell + 1$ do
    if k=1 then
        Sequence the first $\ell$ bases of all the reads in $\mathcal{R}_k$.
    else
        Sequence the $(\ell + k - 1)$-th base of all the reads in $\mathcal{R}_k$.
    end if
    Map all the reads in $\mathcal{R}_k$ to the reference genome with their last $\ell$ bases and Hamming distance $d_{max}$.
    Add uniquely mapped reads in $\mathcal{R}_k$ to the set $\mathcal{R}^A$. Put the rest of reads in the set $\mathcal{R}_{k+1}$.
    Add reads in $\mathcal{R}^A$ to $\mathcal{R}$, if by further extensions they will cover a base more than $C_\epsilon$ times on the reference genome.
    Add reads in $\mathcal{R}^A$ to $\mathcal{R}_{k+1}$, if by further extensions they will cover a base for the $c^{\text{th}}$ times on the reference genome, where $c \leq C_\epsilon$.
end for
```

Mapping fragments of length ℓ with maximum Hamming distance d_{max} to the reference genome is error prone and we first analyze the performance of the alignment procedure. Let $\mathcal{T}_i$ and $\mathcal{F}_i$ denote the true and false alignment events for the i^{th} read, respectively. The i^{th} read of length ℓ with a maximum Hamming distance of $d_{\max}$ is mapped to its true location with probability

$$\mathbb{P}\left\{\mathcal{T}_i\right\} = \sum_{i \leq d_{\max}} \binom{\ell}{i} \left(\epsilon + \nu - \frac{4\epsilon\nu}{3}\right)^i \left((1-\epsilon)(1-\nu) + \frac{\epsilon\nu}{3}\right)^{\ell - i}, \tag{4}$$

and is mapped to a false location on the genome with maximum Hamming distance $d_{\max}$ with probability (using the *union bound*),

$$\mathbb{P}\left\{\mathcal{F}_i\right\} \leq G \times P_w\left(\ell\right) \triangleq G \sum_{i \leq d_{\max}} \binom{\ell}{i} \left(\frac{3}{4}\right)^i \left(\frac{1}{4}\right)^{\ell-i} = G4^{-\ell} \sum_{i \leq d_{\max}} \binom{\ell}{i} 3^i, \tag{5}$$

where $P_w(\ell)$ represents the probability of incorrect alignment of a fragment of length ℓ to another position on the reference genome. Denote, $\mathcal{F} = \cup_{i=1}^{N} \mathcal{F}_i$ as the false alignment event. Hence,

$$\mathbb{P}\left\{\mathcal{F}\right\} \leq N\mathbb{P}\left\{\mathcal{F}_i\right\} \leq NG \times P_w\left(\ell\right).$$

Therefore, if $P_w\left(\ell\right)$ scales as $O\left(\frac{1}{NG}\right)$, $\mathbb{P}\left\{\mathcal{F}\right\}$ tends to zero and all reads are mapped to their true location with $\mathbb{P}\{\mathcal{T}_i\}$'s. In the worst case, if any read is extracted from each base of the reference genome, i.e. $N = G$, then $P_w\left(\ell\right)$ scales as $O\left(\frac{1}{G^2}\right)$. It can be easily verified that for $\epsilon = 0.05$, $\ell = 2\log G$ and $d_{max} = 8$, $\mathbb{P}\left\{\mathcal{F}\right\}$ tends to zero and $\mathbb{P}\{\mathcal{T}_i\} \approx 1$. Therefore for $\epsilon \leq 0.05$, all noisy fragments of length ℓ are *uniquely* mapped to their correct locations on the reference genome.

To compute the number of extra read bases, we use a similar argument as the one used in the noiseless case. To obtain $\mathbb{E}\{\mathcal{B}_j\}$ for the j^{th} read in this case, assume that the ${C_\epsilon}^{\text{th}}$ read after this read starts x bases further. Again $\mathcal{B}_j = \max\{0, \ell - x\}$. Since, x has an *Erlang* distribution, we obtain

$$\mathbb{E}\{\mathcal{B}_j\} = \int_0^{\ell} \left(\ell - x\right) \lambda e^{-\lambda x} \frac{\left(\lambda x\right)^{C_\epsilon - 1}}{\left(C_\epsilon - 1\right)!} \mathrm{dx} = \frac{1}{\left(C_\epsilon - 1\right)!} \left(\left(\ell - \frac{C_\epsilon}{\lambda}\right) \gamma\left(C_\epsilon, \lambda\ell\right) + \frac{1}{\lambda}\left(\lambda\ell\right)^{C_\epsilon} e^{-\lambda\ell}\right) \triangleq V_{C_\epsilon}\left(\ell\right). \tag{6}$$

where $\gamma\left(s, x\right)$ is the lower incomplete gamma function and for $s \in \mathbb{N}$ is equal to

$$\gamma\left(s, x\right) = \left(s - 1\right)! \left[1 - e^{-x}\left(\sum_{i=0}^{s-1} \frac{x^i}{i!}\right)\right].$$

Thus, the average number of over-read bases for all reads in this step is,

$$\begin{aligned}\mathbb{E}\left\{\mathcal{B}\right\} = \mathbb{E}\left\{\sum_{j=1}^{N} \mathcal{B}_j\right\} = N \times V_{C_\epsilon}\left(\ell\right) &= \frac{N}{\left(C_\epsilon - 1\right)!}\left\{\left(\ell - \frac{C_\epsilon}{\lambda}\right)\gamma\left(C_\epsilon, \lambda\ell\right) + \frac{1}{\lambda}\left(\lambda\ell\right)^{C_\epsilon} e^{-\lambda\ell}\right\} \\ &= \left\{\frac{1}{\left(C_\epsilon - 1\right)!}\left(\left(\kappa_n - C_\epsilon\right)\gamma\left(C_\epsilon, \kappa_n\right) + \kappa_n^{C_\epsilon} e^{-\kappa_n}\right)\right\} \times G, \end{aligned} \tag{7}$$

where $\kappa_n = \frac{N\ell}{G}$. Therefore, for any constant κ_n, $\mathbb{E}\{\mathcal{B}\}$ becomes $O(G)$. The coverage bound when each base of the genome is covered by at least C_ϵ reads can be determined as follows,

$$\mathbb{P}\left\{\mathcal{E}_c = \cup_{i=1}^{G} \mathcal{E}_{c,i}\right\} \leq G\mathbb{P}\left\{\mathcal{E}_{c,i}\right\} = G \sum_{j=0}^{C_\epsilon - 1} e^{-\lambda L} \frac{\left(\lambda L\right)^j}{j!}, \tag{8}$$

where $\mathcal{E}_c$ and $\mathcal{E}_{c,i}$ are error events that at least one base and the i^{th} base of the genome are not covered by at least C_ϵ reads, respectively. Therefore, if

$$\lambda L \geq \log G + \left(C_\epsilon - 1\right)\log\left(\log G\right), \tag{9}$$

then each base of the genome is covered by at least C_ϵ reads almost surely. Subsequently, for a fix ℓ and constant value of κ_n, number of reads and read length from (9) become $N = \kappa_n \frac{G}{\ell}$ and $L = \frac{\ell}{\kappa_n}\left(\log G + \left(C_\epsilon - 1\right)\log\left(\log G\right)\right)$ respectively.

Again, there exists a trade-off between the length of reads and number of read bases that can be controlled by κ_n. Figure 5 shows $\mathbb{E}\left\{\mathcal{B}\right\}/G$ in (7) for sequencing error rates of $\epsilon = \{0, 0.02, 0.05\}$ and $C_\epsilon = 6$. Also, Figure 6 shows total read bases relative to read length for different sequencing error rates $\epsilon = \{0, 0.02, 0.05, 0.1\}$. We set $P_\epsilon = 10^{-4}$ and therefore for $\epsilon = \{0, 0.02, 0.05, 0.1\}$ we use: $C_\epsilon = \{1, 4, 6, 8\}$ from (3), and $\ell = \{1, 1.5, 2, 3\} \times \log G$ with $d_{max} = \{0, 4, 8, 15\}$ which satisfy alignment constraints, respectively. These figures show that when $\epsilon = 5\%$, approximately $6.08G$ bases (only $0.08G$ bases are over-read) are read for read length of $L \approx 1000$ bps.

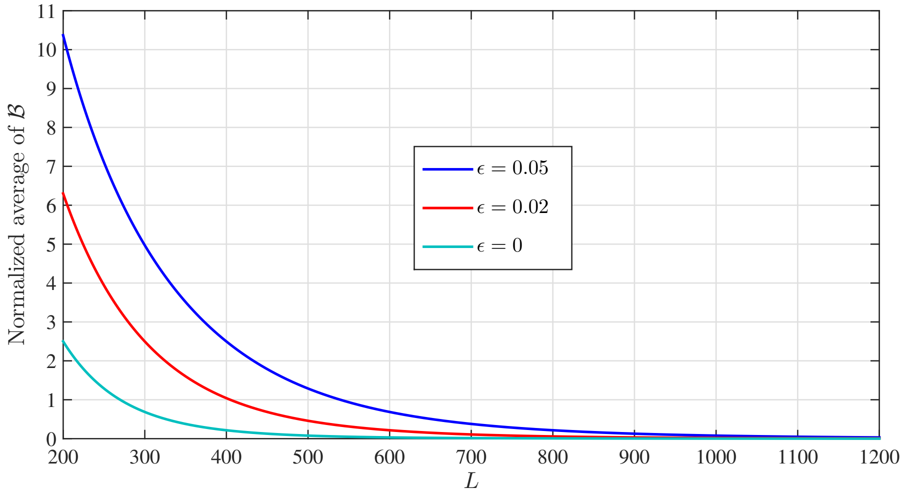

Fig. 5. The normalized average number of over-read bases and read lengths for different sequencing error rates and $C_\epsilon = 6$.

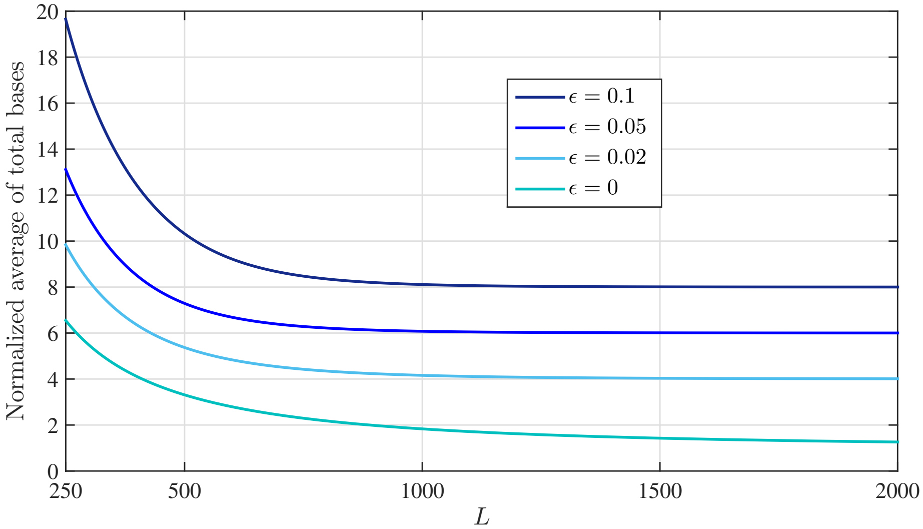

Fig. 6. The normalized total number of read bases and read lengths for different sequencing error rates and $P_\epsilon = 10^{-4}$. Note that, total number of read bases in each error rate tends to its corresponding C_ϵ.

B. Real Genomes

In this section, we consider DNA sequencing of real genomes where many repeats are dispersed across the genome. First, we assume that reads are noiseless. Note that, if all the ℓ-mers of the genome are repetitive elements then Algorithm1 fails in anchoring reads correctly to the reference genome and therefore reading $O\left(G \log G\right)$ is unavoidable. However, as we will show, the repeat patterns in real genomes allow successful coverage of bases with only $O\left(G\right)$ reading bases.

A mosaic model for capturing the repeat structure of the genome is presented in [3]. In this model the reference genome consists of two types of intervals, repeat and random intervals. These types are defined based on two parameters ℓ and d: representing the fragment length and mismatch factor, respectively. Repeat (random) intervals are consecutive bases where any fragment of length ℓ starting from a base within these intervals can be aligned to some other location(s) (one location) of the genome with maximum Hamming distance d. For the sake of simplicity, we consider only $d = 0$.

Let denote a set of all exact repeat intervals of the reference genome as $\mathscr{R}$. Also, assume that a repeat $R \in \mathscr{R}$ has length ℓ_R and repeat lengths have the distribution f_ℓ. We need to treat reads starting from repeat and random intervals, differently. For this purpose, we consider three starting regions for start point of a given read, as S_1, S_2, and S_3. We can determine the average number of total over-read bases (i.e. $\mathbb{E}\left\{\mathcal{B}\right\}$) as follows,

$$\mathbb{E}\left\{\mathcal{B}\right\} = \mathbb{E}\left\{\sum_{i=1}^{N} \mathcal{B}_i\right\} = N\mathbb{E}\left\{\mathbb{E}\left\{\mathcal{B}_i|\mathcal{S}_i\right\}\right\}, \tag{10}$$

where $\mathcal{S}_i$ is a random variable that shows the starting region of the i^{th} read. Hence,

$$\mathbb{E}\left\{\mathcal{B}\right\} = N \sum_{j=1,2,3} \mathbb{E}\left\{\mathcal{B}_i|\mathcal{S}_i = S_j\right\} \mathbb{P}\left\{\mathcal{S}_i = S_j\right\}. \tag{11}$$

In the following, we determine each term of equation (11). Region S_1 is random intervals such that reads starting from random intervals can be anchored to their true locations based on their first ℓ bases. Therefore, we can readily compute the average number of over-read bases in random intervals using equation (1). More precisely,

$$\mathbb{E}\left\{\mathcal{B}_i|\mathcal{S}_i = S_1\right\} = V_1\left(\ell\right) = \ell - \frac{1}{\lambda}\left(1 - e^{-\lambda\ell}\right),$$

where the total average number of these reads is

$$N \times \mathbb{P}\left\{\mathcal{S}_i = S_j\right\} = N\frac{G - \sum_{R\in\mathscr{R}} \ell_R}{G} = \lambda\left(G - \sum_{R\in\mathscr{R}} \ell_R\right).$$

On the other hand, reads starting from repeat intervals can not be anchored unless it contains an ℓ-mer which resides in random interval. Using this fact, c.f. Figure 7, each repeat interval of length ℓ_R can be partitioned into two disjoint intervals: 1) Mappable zone: the last $\min\{L-\ell, \ell_R\}$ bases, 2) Un-mappable zone: the first $\max\{0, \ell_R-L+\ell\}$ bases. Regions S_2 and S_3 are mappable and un-mappable zones, respectively.

Clearly, reads from un-mappable zones cannot be anchored and therefore they need to be read up to length L. Using (1), we compute the average number of over-read bases for each read in un-mappable zones as:

$$\mathbb{E}\left\{\mathcal{B}_i|\mathcal{S}_i = S_3\right\} = V_1\left(L\right) = L - \frac{1}{\lambda}\left(1 - e^{-\lambda L}\right) \approx L - \frac{1}{\lambda},$$

where the total average number of these reads is

$$N \times \mathbb{P}\left\{\mathcal{S}_i = S_3\right\} = N\frac{\sum_{R\in\mathscr{R}} \max\{0, \ell_R - L + \ell\}}{G} = \lambda \sum_{R\in\mathscr{R}} \max\{0, \ell_R - L + \ell\}.$$

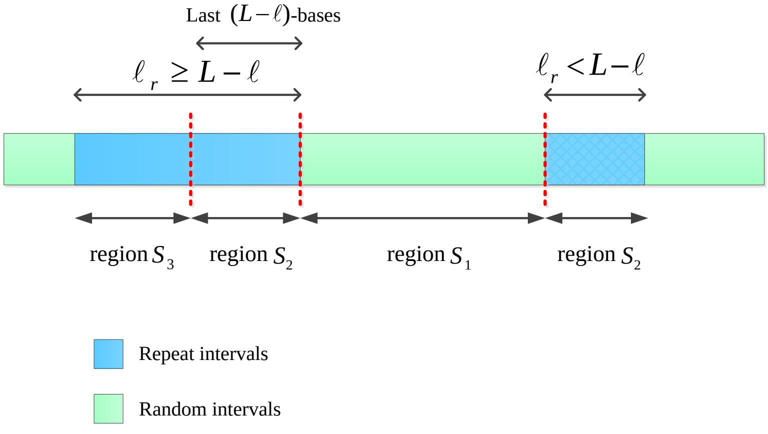

Fig. 7. Repeat intervals regions. The first region (S_1) is random intervals. The second region (S_2) is the last $L-\ell$ bases of repeat intervals. The third region (S_3) is the other bases of repeat intervals.

It remains to compute the number of over-read bases for reads from mappable zones. Consider all mappable regions $s_m \in S_2$ of length l_m, for $m = \{1, \cdots, |\mathscr{R}|\}$. If the m^{th} repeat interval has length ℓ_m, then $l_m =$

$\min\{L-\ell,\ell_m\}$. At least $l_m+\ell$ bases of each read within s_m are being sequenced. Consider the i^{th} read has distance of l_i from the end of its mappable zone. The average number of over-read bases for the i^{th} read in mappable zones can be determined as:

$$\begin{aligned}\sum_{m=1}^{|\mathscr{R}|}\mathbb{E}\left\{\mathcal{B}_i|\mathcal{S}_i=S_2,r_i\in s_m\right\}\mathbb{P}\left\{r_i\in s_m|\mathcal{S}_i=S_2\right\}&=\sum_{m=1}^{|\mathscr{R}|}\left\{\int_0^{\ell_m}\frac{1}{l_m}V_1\left(l_i+\ell\right)\mathrm{d}l_i\right\}\frac{\mathbb{P}\left\{\mathcal{S}_i=S_2,r_i\in s_m\right\}}{\mathbb{P}\left\{\mathcal{S}_i=S_2\right\}}\\&=\sum_{m=1}^{|\mathscr{R}|}\left\{\int_0^{l_m}\frac{1}{l_m}V_1\left(l_i+\ell\right)\mathrm{d}l_i\right\}\frac{l_m}{G}.\end{aligned}$$

Thus, the average number of total over-read bases for all reads in mappable zones becomes,

$$N\mathbb{E}\left\{\mathcal{B}_i|\mathcal{S}_i=S_2\right\}\mathbb{P}\left\{\mathcal{S}_i=S_2\right\}=\lambda\sum_{m=1}^{|\mathscr{R}|}\int_0^{l_m}V_1\left(l_i+\ell\right)\mathrm{d}l_i. \tag{12}$$

Define,

$$\bar{V}_1\left(l_m\right)=\int_0^{l_m}V_1\left(\ell_i+\ell\right)\mathrm{d}\ell_i=\frac{l_m^2}{2}+l_m\ell-\frac{l_m}{\lambda}+\frac{1}{\lambda^2}\left[1-e^{-\lambda(l_m+\ell)}\right].$$

Therefore, the average number of over-read bases by considering repeat structure in (10) becomes,

$$\begin{aligned}\mathbb{E}\left\{\mathcal{B}\right\}=&\lambda\left(G-\sum_{R\in\mathscr{R}}\ell_R\right)V_1\left(\ell\right)+\lambda\left(\sum_{R\in\mathscr{R}}\max\{0,\ell_R-L+\ell\}\right)V_1\left(L\right)+\lambda\sum_{R\in\mathscr{R}}\bar{V}_1\left(\min\{L-\ell,\ell_R\}\right)\\&=\lambda G\left(1-\frac{|\mathscr{R}|}{G}\int_{\ell_R}\ell_R f_{\ell_R}\mathrm{d}\ell_R\right)V_1\left(\ell\right)+\left(\lambda L-1\right)|\mathscr{R}|\int_{\ell_R\geq L-\ell}\left(\ell_R-L+\ell\right)f_{\ell_R}\mathrm{d}\ell_R\\&+\lambda|\mathscr{R}|\int_{\ell_R}\left(\frac{\ell_R^2}{2}+\ell_R\ell-\frac{\ell_R}{\lambda}+\frac{1}{\lambda^2}\left[1-e^{-\lambda(\ell_R+\ell)}\right]\right)f_{\ell_R}\mathrm{d}\ell_R.\end{aligned} \tag{13}$$

Given the repeat length distribution (i.e. f_{ℓ_R}) of any real genome, we can determine the $\mathbb{E}\left\{\mathcal{B}\right\}$ for that genome. The distribution of $\log f_{\ell_r}$ for Human genome hg19 is illustrated in Figure 8. Also, Figure 9 shows $\mathbb{E}\left\{\mathcal{B}\right\}/G$ for whole genome of hg19 and i.i.d. genome. In this simulation, we used $\ell=\log G\approx 30$. Results confirm that for real data set, we can read only $O(G)$ bases to assemble the genome.

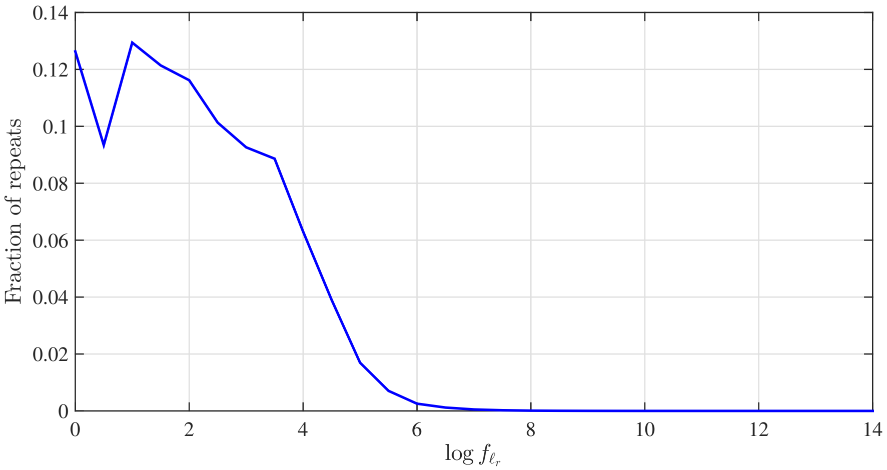

Fig. 8. The distribution of repeat lengths logarithm (i.e. $\log f_{\ell_R}$) for Human genome hg19 based on Meta-aligner model with $(\ell,d)=(30,0)$ defined in [3].

When reads are contaminated with sequencing errors of rate ϵ, we use a proper value of ℓ such that a fragment length of ℓ is aligned to its correct location with a probability close to one. Also, consider the coverage depth, i.e. C_ϵ, as (3). We model the reference genome with Meta-aligner $(\ell,d=0)$-model. Thus, using the same argument

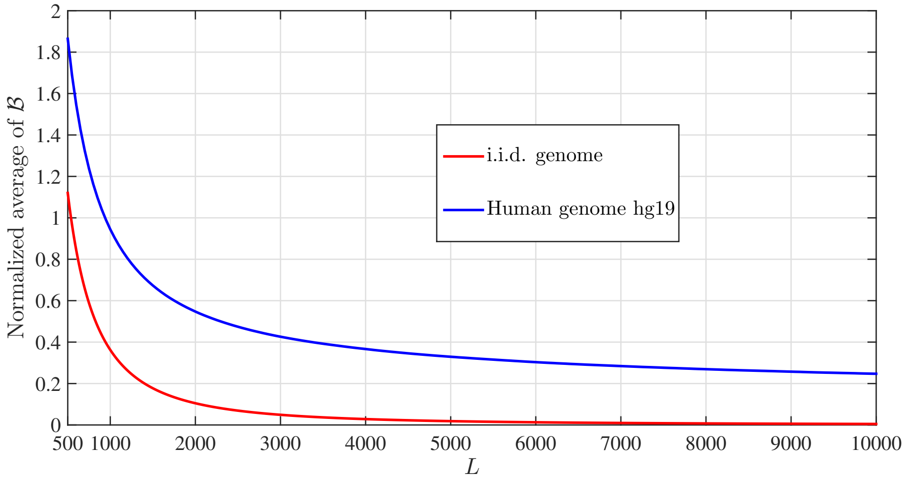

Fig. 9. The average number of over-read bases for i.i.d. genome and Human genome hg19 with $\ell \approx 30$ and $N = G \log G/L$.

as noiseless reads, the average number of over-read bases can be determined similar to (13) except incorporating $V_{C_\epsilon}(.)$ in (6) instead of $V_1(.)$. Therefore, the average number of over-read bases for real genomes in the presence of noise becomes,

$$\mathbb{E}\left\{\mathcal{B}\right\} = \lambda\left(G - \sum_{R\in\mathcal{R}} \ell_R\right) V_{C_\epsilon}\left(\ell\right) + \lambda\left(\sum_{R\in\mathcal{R}} \max\{0, \ell_R - L + \ell\}\right) V_{C_\epsilon}\left(L\right) + \lambda \sum_{R\in\mathcal{R}} \bar{V}_{C_\epsilon}\left(\min\{L-\ell, \ell_R\}\right), \tag{14}$$

where $V_{C_\epsilon}\left(L\right) \approx \lambda L - C_\epsilon$ and

$$\bar{V}_{C_\epsilon}\left(l_m\right) = \int_0^{l_m} V_{C_\epsilon}\left(\ell_i + \ell\right) \mathrm{d}\ell_i \ .$$

Figure 10 shows the $\mathbb{E}\left\{\mathcal{B}\right\}/G$ for whole genome of hg19 with $\epsilon = \{0, 0.05\}$. We use $\ell = \log G \approx 30$ and $\ell = 2\log G \approx 60$ for $\epsilon = 0$ and $\epsilon = 0.05$, respectively.

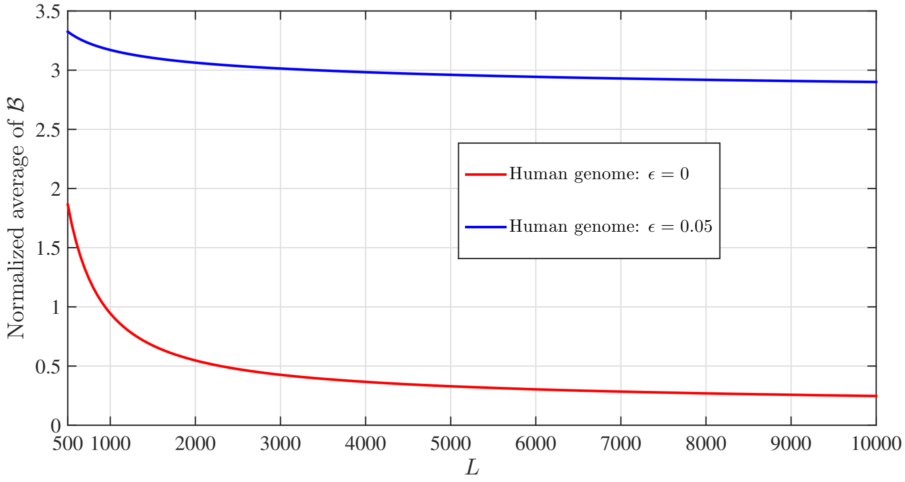

Fig. 10. The normalized average number of over-read bases for i.i.d. genome and Human genome hg19 with $\ell \approx \{30, 60\}$ and $N = G \log G/L$ for $\epsilon = \{0, 0.05\}$, respectively.

III. Algorithm Coverage Analysis

In this section, we compute the number of gapped bases in the reference genome when the proposed algorithms are used. First, consider i.i.d. genomes. We show that the whole genome is in fact covered by the noiseless reads when using Algorithm1. Suppose that all reads of length ℓ are aligned to the reference genome. Let $\mathcal{E}$ denotes the *error* event which is the event that a base is not covered in our algorithm. Let us denote the event of not covering the i^{th} base by $\mathcal{E}_i$. Thus,

$$\mathbb{P}\left(\mathcal{E}\right) = \cup_{i=1}^{G} \mathbb{P}\left(\mathcal{E}_i\right). \tag{15}$$

From union bound, we have

$$\mathbb{P}\left(\mathcal{E}\right) \le G\mathbb{P}\left(\mathcal{E}_i\right), \tag{16}$$

for arbitrary $i \in \{1, \cdots, G\}$. Define the set $\mathcal{S}_i$, consisting of starting points of reads that are aligned to the reference genome with less than L bases before the i^{th} base location. Since the nearest read in $\mathcal{S}_i$ to the i^{th} base does not overlap with other reads from its right hand side, this read will be extended in subsequent steps of the algorithm and at the end, the i^{th} base will be covered by this read. Hence, $\mathcal{S}_i$ must be an empty set and $\mathcal{E}_i$ occurs when no read's starting point is located less than L bases before the i^{th} base. This condition is the same as the coverage bound condition. Thus,

$$\mathbb{P}\left(\mathcal{E}\right) \le G\mathbb{P}\left(\mathcal{E}_i\right) = Ge^{-\lambda L} = Ge^{-\frac{NL}{G}}. \tag{17}$$

Thus, if the number of reads N and the reads' length L satisfy the coverage condition in [1] (i.e. $NL \ge G\log G$), the sequence is completely covered by the reads in our method for noiseless reads and i.i.d. genomes.

For noisy reads, we show the perfect coverage of reference genome when noisy reads are used in Algorithm 2. For this purpose, the same argument as noiseless reads is considered. Let $\mathcal{E}_n$ denotes the *error* event which is the event that a base is not covered by at least C_ϵ reads in our algorithm. Also, denote error event for the i^{th} base by $\mathcal{E}_{n,i}$. Therefore, $\mathcal{E}_n = \cup_{i=1}^{G}\mathcal{E}_{n,i}$ and $\mathcal{E}_{n,i}$ occurs when less than C_ϵ read's starting points are located within L bases before the i^{th} base. This condition is the same as the coverage bound condition with a given C_ϵ in (8). Thus, if number of reads N and reads' length L satisfy the coverage condition for noisy reads in (9), the sequence is completely covered by at least C_ϵ noisy reads in our method as well.

Now consider a real genome. We must determine how many bases are covered with Algorithm 1 or Algorithm 2 (based on noiseless or noisy reads). For this purpose, let $\mathcal{E}_k$ denotes the error event that the k^{th} base is not covered by reads with the proposed algorithms. We only consider coverage in this section, therefore, we use $C_\epsilon = 1$ for noisy reads. Based on the base location and its neighboring repeat intervals within the genome, this base is classified to two different classes as shown in Figure 11. Using these two classes, different sub-classes for locating random and repeat intervals can be modelled. Note that similar to i.i.d. genome, locating one read within distance of L bases before a given base is sufficient for covering that base. In the following, we determine probability of the $\mathcal{E}_k$ for each class. In the proposed analysis, we assume that each fragment of length ℓ is mapped uniquely to the reference genome with probability p_t. Also, we denote the number of reads within a random interval of length l as $\mathcal{N}(l)$.

1) Class A: Assume d_1 and d_2 bases within distance of L bases before and after the k^{th} base are in random interval, respectively. If a read has a fragment within a random interval, it can be mapped to the reference genome uniquely. If $d_1 + d_2 \ge L$, we divide the interval of length L before the k^{th} base to three parts: 1) all bases with distance $[L, d_1]$ from base k, 2) all bases with distance $[d_1, L - d_2]$ from base k, and 3) the remaining bases of the random interval with distance $[L - d_2, 0]$ from base k.
Divide the first part to $\frac{L-d_1}{\ell}$ disjoint sub-intervals of length ℓ, If any read starts within the j^{th} sub-interval, the number of fragments of that read within the random interval is $\frac{d_1}{\ell} + j - 1$. If any read starts within the second part, the number of fragments of that read within the random interval is $\frac{L}{\ell}$. In addition, divide the

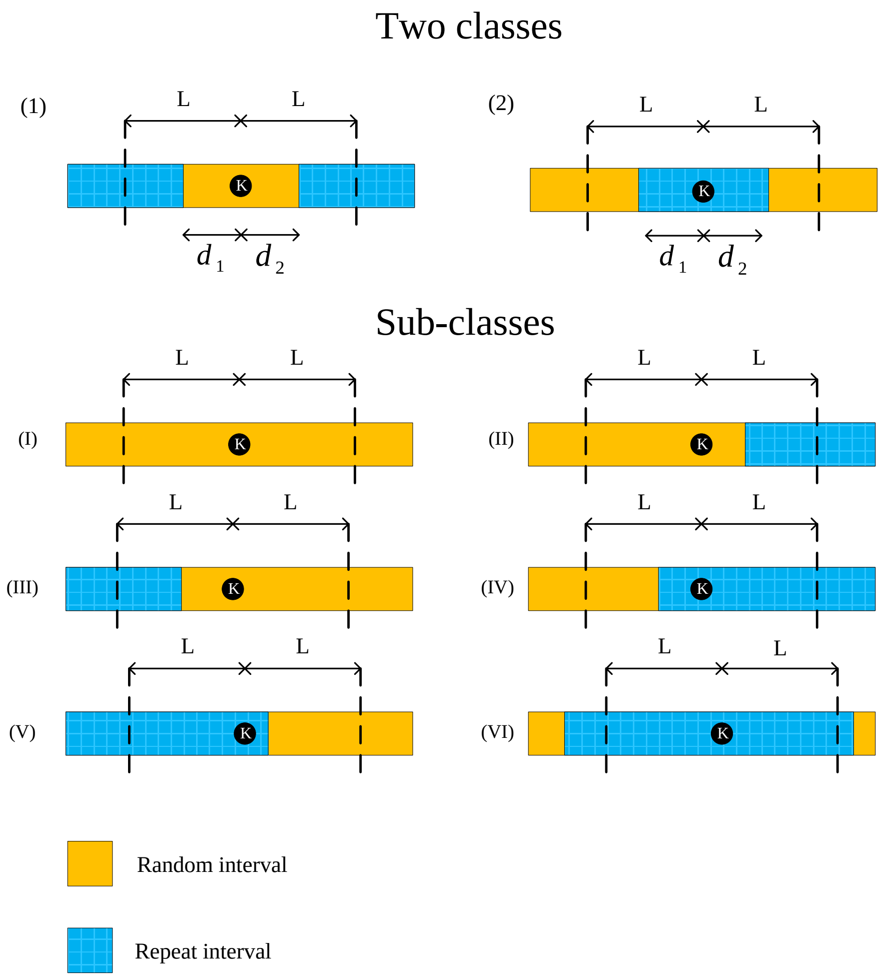

Fig. 11. Classification of each base of the reference genome based on random and repeat intervals of the genome. By considering special cases for these two classes, six sub-classes created.

third part to $\frac{L-d_2}{\ell}$ sub-intervals of length ℓ such that if any read starts within the j^{th} sub-interval, the number of fragments of that read within the random interval is $\frac{L}{\ell} - j$. Thus,

$$\begin{aligned}
\mathbb{P}\left\{\mathcal{E}_k\right\} = & \prod_{i=1}^{\frac{L-d_1}{\ell}} \left\{ \sum_{n_i=0}^{\infty} \mathbb{P}\left\{\mathcal{N}(\ell) = n_i\right\} (1-p_t)^{n_i\left(\frac{d_1}{\ell}+i-1\right)} \right\} \\
& \times \left\{ \sum_{n'=0}^{\infty} \mathbb{P}\left\{\mathcal{N}(d_1+d_2-L) = n'\right\} (1-p_t)^{n'\frac{L}{\ell}} \right\} \\
& \times \prod_{i=1}^{\frac{L-d_2}{\ell}} \left\{ \sum_{n_i=0}^{\infty} \mathbb{P}\left\{\mathcal{N}(\ell) = n_i\right\} (1-p_t)^{n_i\left(\frac{L}{\ell}-i\right)} \right\} \\
& = \prod_{i=1}^{\frac{L-d_1}{\ell}} e^{-\lambda\ell_i\left(1-(1-p_t)^{\frac{d_1}{\ell}+i-1}\right)} \times e^{-\lambda(d_1+d_2-L)\left(1-(1-p_t)^{\frac{L}{\ell}}\right)} \times \prod_{i=1}^{\frac{L-d_2}{\ell}} e^{-\lambda\ell_i\left(1-(1-p_t)^{\frac{L}{\ell}-i}\right)} \\
& = \exp\left(-\lambda L + \lambda\ell \frac{(1-p_t)^{\frac{d_1}{\ell}} + (1-p_t)^{\frac{d_2}{\ell}} - 2\,(1-p_t)^{\frac{L}{\ell}}}{p_t} + \lambda\,(d_1+d_2-L)\,(1-p_t)^{\frac{L}{\ell}}\right). \qquad (18)
\end{aligned}$$

The same result is obtained when $d_1 + d_2 < L$, i.e.,

$$\begin{aligned}
\mathbb{P}\left\{\mathcal{E}_k\right\} =& \prod_{i=1}^{\frac{d_2}{\ell}} \left\{ \sum_{n_i=0}^{\infty} \mathbb{P}\left\{\mathcal{N}(\ell) = n_i\right\} (1-p_t)^{n_i\left(\frac{d_1}{\ell}+i-1\right)} \right\} \\
& \times \left\{ \sum_{n'=0}^{\infty} \mathbb{P}\left\{\mathcal{N}(L-d_1-d_2) = n'\right\} (1-p_t)^{n' \frac{d_1+d_2}{\ell}} \right\} \\
& \times \prod_{i=1}^{\frac{d_1}{\ell}} \left\{ \sum_{n_i=0}^{\infty} \mathbb{P}\left\{\mathcal{N}(\ell) = n_i\right\} (1-p_t)^{n_i\left(\frac{d_1+d_2}{\ell}-i\right)} \right\} \\
=& \prod_{i=1}^{\frac{d_2}{\ell}} e^{-\lambda \ell_i \left(1-(1-p_t)^{\frac{d_1}{\ell}+i-1}\right)} \times e^{-\lambda(L-d_1-d_2)\left(1-(1-p_t)^{\frac{d_1+d_2}{\ell}}\right)} \times \prod_{i=1}^{\frac{d_1}{\ell}} e^{-\lambda \ell_i \left(1-(1-p_t)^{\frac{d_1+d_2}{\ell}-i}\right)} \\
=& \exp\left(-\lambda L + \lambda \ell \frac{(1-p_t)^{\frac{d_1}{\ell}} + (1-p_t)^{\frac{d_2}{\ell}} - 2\left(1-p_t\right)^{\frac{d_1+d_2}{\ell}}}{p_t} + \lambda\left(L-d_1-d_2\right)\left(1-p_t\right)^{\frac{d_1+d_2}{\ell}} \right).
\end{aligned} \tag{19}$$

2) Class B: Assume d_1 and d_2 bases within distance of L bases before and after the k^{th} base are in repeat interval, respectively. If $d_1 + d_2 \geq L$, we divide the interval of length L before the k^{th} base to three parts similar to class A. Thus,

$$\begin{aligned}
\mathbb{P}\left\{\mathcal{E}_k\right\} =& \prod_{i=1}^{\frac{L-d_1}{\ell}} \left\{ \sum_{n_i=0}^{\infty} \mathbb{P}\left\{\mathcal{N}(\ell) = n_i\right\} (1-p_t)^{n_i\left(\frac{L-d_1}{\ell}-i+1\right)} \right\} \\
& \times \left\{ \sum_{n'=0}^{\infty} \mathbb{P}\left\{\mathcal{N}(d_1+d_2-L) = n'\right\} \right\} \\
& \times \prod_{i=1}^{\frac{L-d_2}{\ell}} \left\{ \sum_{n_i=0}^{\infty} \mathbb{P}\left\{\mathcal{N}(\ell) = n_i\right\} (1-p_t)^{n_i(i)} \right\} \\
=& \prod_{i=1}^{\frac{L-d_1}{\ell}} e^{-\lambda \ell_i \left(1-(1-p_t)^{\frac{L-d_1}{\ell}-i+1}\right)} \times \prod_{i=1}^{\frac{L-d_2}{\ell}} e^{-\lambda \ell_i \left(1-(1-p_t)^{i}\right)} \\
=& \exp\left(-\lambda\left(2L-d_1-d_2\right) + \lambda \ell \frac{2\left(1-p_t\right) - \left(1-p_t\right)^{\frac{L-d_1}{\ell}+1} - \left(1-p_t\right)^{\frac{L-d_2}{\ell}+1}}{p_t} \right).
\end{aligned} \tag{20}$$

The same result is obtained when $d_1 + d_2 < L$, i.e.,

$$\begin{aligned}
\mathbb{P}\left\{\mathcal{E}_k\right\} =& \prod_{i=1}^{\frac{d_2}{\ell}} \left\{ \sum_{n_i=0}^{\infty} \mathbb{P}\left\{\mathcal{N}(\ell) = n_i\right\} (1-p_t)^{n_i\left(\frac{d_2}{\ell}-i+1\right)} \right\} \\
& \times \left\{ \sum_{n'=0}^{\infty} \mathbb{P}\left\{\mathcal{N}(L-d_1-d_2) = n'\right\} (1-p_t)^{n' \frac{L-d_1-d_2}{\ell}} \right\} \\
& \times \prod_{i=1}^{\frac{d_1}{\ell}} \left\{ \sum_{n_i=0}^{\infty} \mathbb{P}\left\{\mathcal{N}(\ell) = n_i\right\} (1-p_t)^{n_i\left(\frac{L-d_2-d_1}{\ell}+i-1\right)} \right\} \\
=& \prod_{i=1}^{\frac{d_2}{\ell}} e^{-\lambda \ell_i \left(1-(1-p_t)^{\frac{d_2}{\ell}-i+1}\right)} \times e^{-\lambda(L-d_1-d_2)\left(1-(1-p_t)^{\frac{L-d_1-d_2}{\ell}}\right)} \times \prod_{i=1}^{\frac{d_1}{\ell}} e^{-\lambda \ell_i \left(1-(1-p_t)^{\frac{L-d_1-d_2}{\ell}+i-1}\right)}
\end{aligned}$$

$$= \exp\left(- \lambda L + \lambda \ell \frac{(1-p_t) - (1-p_t)^{\frac{d_2}{\ell}} + (1-p_t)^{\frac{L-d_1-d_2}{\ell}} - (1-p_t)^{\frac{L-d_2}{\ell}}}{p_t} + \lambda \left(L - d_1 - d_2\right) (1-p_t)^{\frac{L-d_1-d_2}{\ell}} \right). \tag{21}$$

We are interested for error probabilities of some special cases. These interested cases are illustrated as sub-classes in Figure 11. The error probability of each sub-class is determined in the following.

- The first sub-class (I): This sub-class can be modelled with the first class with $d_1 = d_2 = L$. Thus,
$$\mathbb{P}\left\{\mathcal{E}_k\right\} = \exp\left(-\lambda L \left(1 - (1-p_t)^{\frac{L}{\ell}}\right)\right). \tag{22}$$
- The second sub-class (II): Consider that d bases within distance of L bases after the k^{th} base is in random interval. This sub-class can be modelled with the first class with $d_1 = L$ and $d = d_2 < L$. Thus,
$$\mathbb{P}\left\{\mathcal{E}_k\right\} = \exp\left(-\lambda L + \lambda \ell \frac{(1-p_t)^{\frac{d}{\ell}} - (1-p_t)^{\frac{L}{\ell}}}{p_t} + \lambda d \left(1-p_t\right)^{\frac{L}{\ell}}\right). \tag{23}$$
- The third sub-class (III): Consider that d bases within distance of L bases before the k^{th} base is in random interval. This sub-class can be modelled with the first class with $d_2 = L$ and $d = d_1 < L$. Thus, the error probability of this class is the same as the sub-class (II) except that $L - d$ bases within distance of L before the k^{th} base exist in random interval.
- The forth sub-class (IV): Consider that d bases within distance of L bases before the k^{th} base is in repeat interval. This sub-class can be modelled with the second class with $d_2 = L$ and $d = d_1 < L$. Thus,
$$\mathbb{P}\left\{\mathcal{E}_k\right\} = \exp\left(-\lambda L + \lambda \ell \frac{(1-p_t) - (1-p_t)^{\frac{L-d}{\ell}+1}}{p_t}\right). \tag{24}$$
- The fifth sub-class (V): Consider that d bases within distance of L bases after the k^{th} base is in repeat interval. This sub-class can be modelled with the second class with $d_1 = L$ and $d = d_2 < L$. Thus, the error probability of this class is the same as the sub-class (IV) except that $L - d$ bases exist within distance of L before the k^{th} base in repeat interval.
- The sixth class (VI): This sub-class can be modelled with the second class with $d_1 = d_2 = L$. Thus, the error probability of this sub-class is 1.

Thus, by considering repeat structure of the genome, we can determine the probability of coverage for the genome. We classify bases of Human genome hg19 and determine probability of gap for each class using (18)-(21). The average probabilities of gap, i.e. $\mathbb{E}_k\left\{\mathbb{P}\left\{\mathcal{E}_k\right\}\right\}$, for different values of $\tilde{\lambda} = \lambda L / \log G$ and read lengths (L) are shown in Figures 12-13. These probabilities of gap are shown for $p_t = 1$ and $p_t = 0.7$ (dotted line). Note that, the coverage bound shows that using reads of length $L \geq \log G$ and $\tilde{\lambda} \geq 1$, all bases of an i.i.d. genome are covered reads with probability almost one.

IV. Results and discussion

In this section, we propose simulation results for chr19 of Human genome hg19 with different sequencing error rates.

A. *Benchmark*

The chr19 of Human genome hg19 is used as the reference. Noisy reads are also extracted uniformly from the reference genome and are mapped to this genome. We consider sequencing error rates of $\epsilon = \{0, 5, 10\}\%$ with 90% mismatches and 10% indels. Errors are added by an i.i.d. manner. We use Meta-aligner [3] to align reads with their two fragments of length ℓ not with all their bases. We consider only the first stage Meta-aligner. Since with a mismatch percentage of α and read length of ℓ, there are $\alpha \times \ell$ bases altered in each read on the average, we allow Meta-aligner to align reads to the reference genome with a distance of $\lceil \alpha \times \ell \rceil$. For chr19 of Human genome hg19 of size G bases, $\ell = \log G$ is approximately equal to 26 and we use $\ell = 30$. Therefore, $N = \left(30 + 4\left(C_\epsilon - 1\right)\right) G/L$ reads are randomly generated from chr19 for any read length L and C_ϵ. Also, we consider $C_0 = 1$, $C_5 = 6$, $C_{10} = 8$.

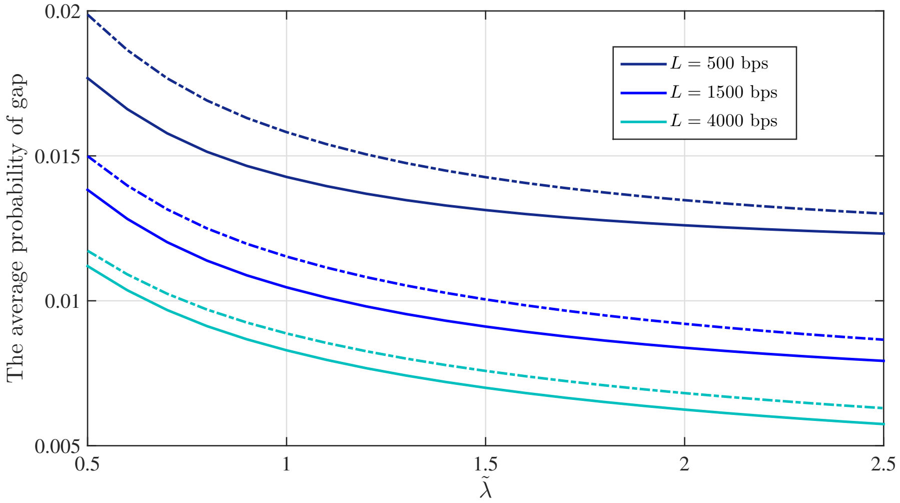

Fig. 12. The average probability of gap within Human genome hg19 versus $\tilde{\lambda} = \lambda L / \log G$ and for different read length $L = \{500, 1000, 2000, 3000\}$ bps with $p_t = 1$ and $p_t = 0.7$ (dotted line).

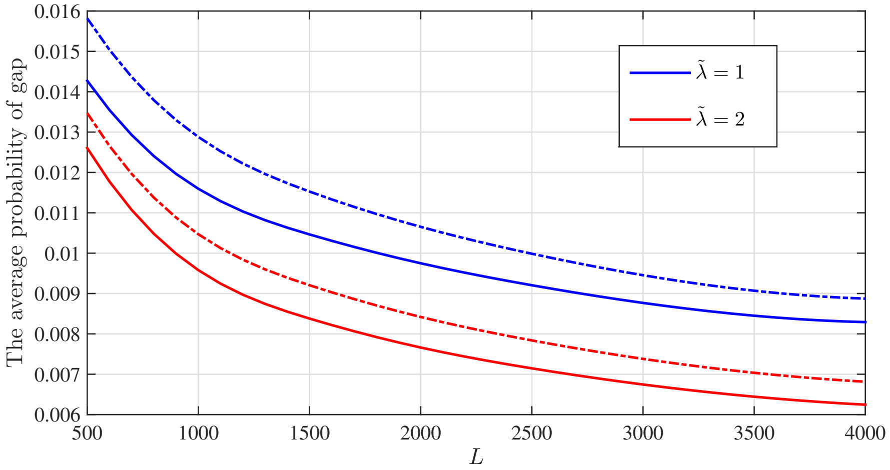

Fig. 13. The average probability of gap within Human genome hg19 versus read length L and for two values of $\tilde{\lambda} = \lambda L / \log G = \{1, 2\}$ with $p_t = 1$ and $p_t = 0.7$ (dotted line).

In each simulation, we also present number of aligned reads by Meta-aligner. Reports of Meta-aligner show its robustness to sequencing errors such that it aligns many reads at its first stage almost correct. We need to read 2ℓ bases at the first step of Meta-aligner which increase over-read bases, but most of the mapped reads are located on the genome correctly.

B. Coverage and Over-read Bases Analysis

Since, number of mapped reads effect on covered bases of the genome, before coverage simulation results we determine number of mapped reads at end of the first stage of Meta-aligner. Figure 14 shows fraction of mapped reads for $\epsilon = \{0, 5, 10\}\%$. Results show that most of reads are mapped uniquely to the reference genome.

In Figure 15, the total number of bases not covered by mapped reads (also known as genome gaps) for various sequencing error rates after the first stage of Meta-aligner is presented. By increasing read length more repeat are bridged by reads and the gap fraction is decreased. Also, Figure 16 shows step by step gap faction of the genome for read length $L = 1000$ and different sequencing error rates. This figure shows that using the proposed method, gradually the reference genome is covered by reads. Since, the remaining bases of the genome locate within long repeat regions, they are covered by reads.

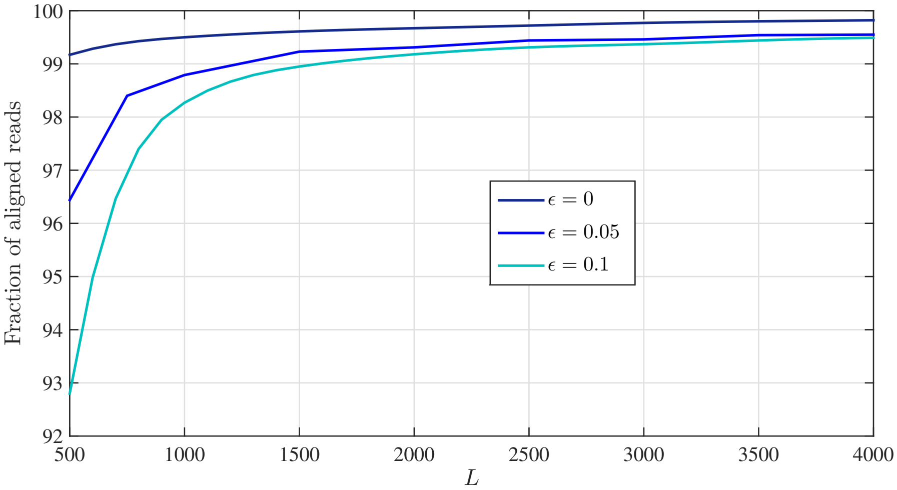

Fig. 14. Fraction of mapped reads for different read lengths and sequencing error rates after the first stage of Meta-aligner.

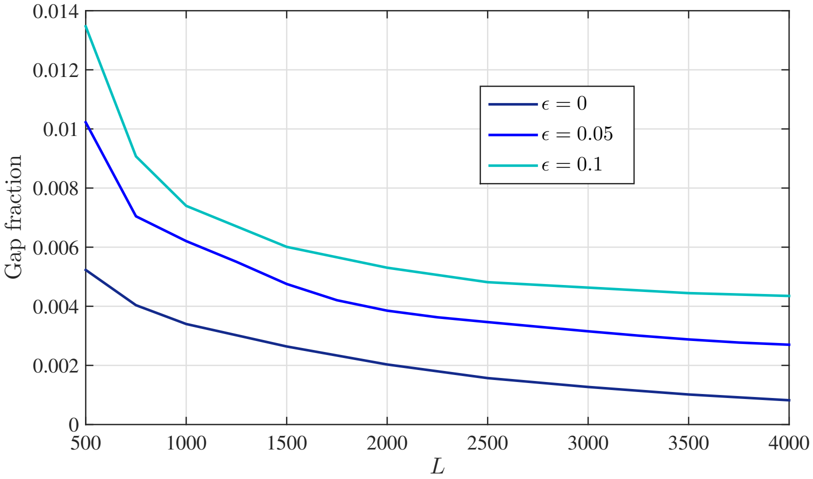

Fig. 15. Fraction of gap within the chr19 for different read lengths and different sequencing error rates after the first stage of Meta-aligner.

Figure 17 illustrates the normalized total number of read bases for various sequencing error rates. Note that, for large enough read length number of over-read bases tends to zero and only $O(G)$ bases needs to be sequenced by sequencer machine. For different values of $\epsilon = \{0, 5, 10\}\%$, almost $\{1.2, 6.4, 8.7\} \times G$ bases are read by using Meta-aligner in read length of $L = 4000$ bps, respectively. Also, Figure 18 shows step by step normalized total number of read bases of the genome for read length $L = 1000$ and different sequencing error rates. This figure reveal that after each step some unmapped reads interfere with other mapped reads and total read bases are increased.

As these simulation results show, increasing sequencing error rates leads to an increase in the number of over-read bases. This is due to the fact that each read can not be aligned to the genome at shorter lengths and its length is increased iteratively. In addition, it is noticeable that genomes with a larger percentage of repeat patterns naturally lead to a greater level of over-read bases (comparison of chr19 with i.i.d. genome). In such scenarios, less number of reads are uniquely aligned to the genome due to the ambiguity caused by repeating patterns.

V. Conclusion and Future Works

Lander and Waterman have presented the coverage bound based on random sampling of i.i.d. DNA sequence. After sampling, read fragments are sent to the processing part. Under such model, the coverage bound shows that minimum number of reads required for covering the whole genome is $N \approx G \log G/L$. Equivalently, $NL \approx G \log G$ bases are required to cover the whole genome. In our method, sequencing and processing are combined such that

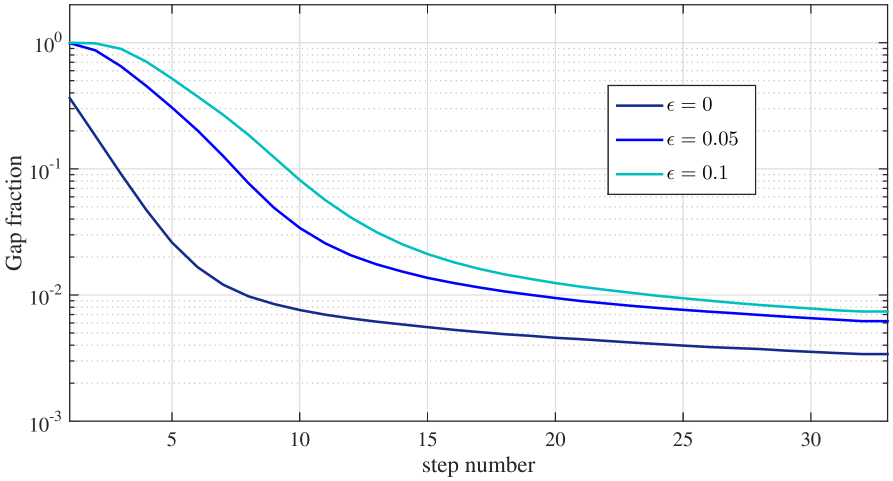

Fig. 16. Step-by-step fraction of gap within the chr19 for different sequencing error rates and read length of $L = 1000$.

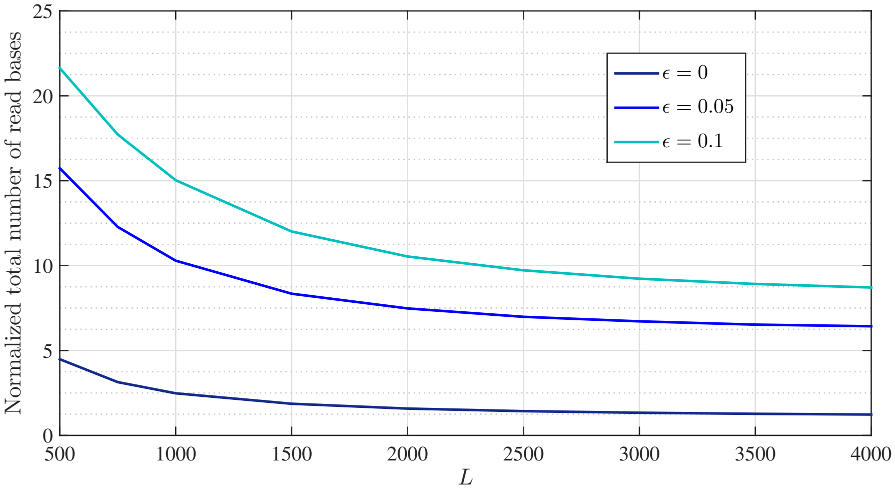

Fig. 17. Normalized total number of read bases for various sequencing error rates and read lengths.

first all fragments are sequenced up to ℓ bases, for some ℓ, and then the processor maps the fragments that are uniquely mapped to the reference genome. Unmapped reads with non-overlapped reads from the right hand side at the first step are sent back to sequencer for extension to next bases. This procedure is repeated until the process reached the maximum read length L. As shown in the paper, through use of such approach, the number of bases read in the sequencing part reduces to $O(G)$ bases, a reduction by a $\log G$ factor in comparison with Lander-Waterman coverage bound.

We propose theoretical results for i.i.d. and real genomes with noiseless reads and i.i.d. genome with noisy reads. Also, we have simulated our method for chr19 of Human genome hg19 with different sequencing error rates. Simulation results support the validity of the proposed algorithm and demonstrate our improvement on coverage bound for real genomes.

As future work, we may expand our algorithm to derive more efficient alignment algorithms in terms of complexity and precision. Also, we can extend this method for Denovo sequencing.

References

[1] Eric S. Lander and Michael S. Waterman, ”Genomic mapping by fingerprinting random clones: A mathematical analysis”, Genomics, Volume 2, No. 3, pp. 231 - 239, 1988.

[2] S.A. Motahari, G. Bresler, D. Tse, ”Information Theory of DNA Shotgun Sequencing”, Transaction of Information Theory, 2013.

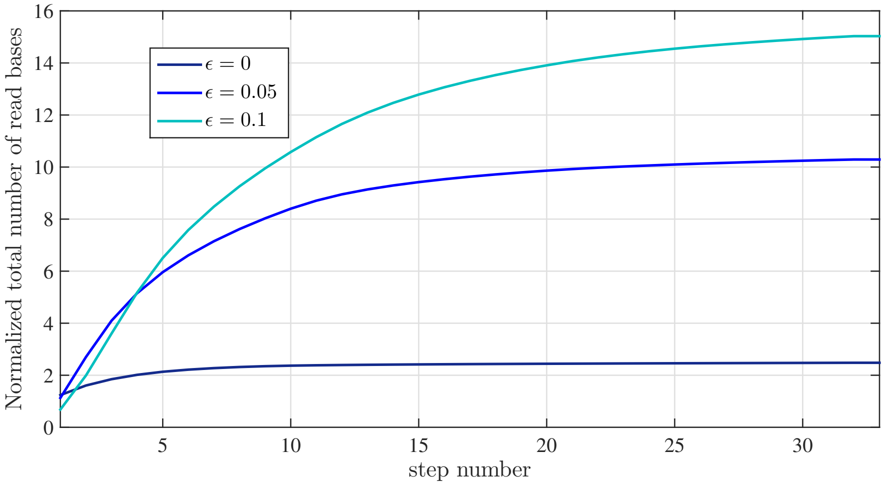

Fig. 18. Step-by-step normalized total number of read bases for various sequencing error rates and read length of $L = 1000$.

[3] D. Nashta-ali, A. Aliyari1, A. Ahmadian Moghadam, M. A. Edrisi, S. A. Motahari and B. H. Khalaj, ”Meta-aligner: Long-read alignment based on genome statistics”, bioRxiv 060129.

[4] Michael L. Metzker, ”Sequencing technologies- the next generation”, Nature Reviews, Genetics, Volume 11, Jan. 2010.

[5] Pacbio RS-II Sequencing System Brochure.

[6] A. Meller, D. Branton, ”Single molecule measurements of DNA transport through a nanopore”, Electrophoresis Special Issue: Fundamental Studies in Separation Science, Volume 23, Issue 16, pp. 2583 - 2591, August 2002.

[7] Daniel Branton and et al, ”The potential and challenges of nanopore sequencing”, Nature Biotechnology, Volume 26, No. 10, October 2008.

[8] Ku Chee-Seng and H. Roukos Dimitrios, ”From next-generation sequencing to nanopore sequencing technology: paving the way to personalized genomic medicine”, Expert Rev. Medical Devices, Volume 10, No. 1, pp. 1 - 6, 2013.

[9] http://www.ncbi.nlm.nih.gov/sra?term=SRX533609.