The Journal of Supercomputing (2022) 78:14769–14793
https://doi.org/10.1007/s11227-022-04449-9

Supercomputing of reducing sequenced bases in de novo sequencing of the human genome

Shirin Kavezadeh[1] · Ashkan Farazin[2] · Alireza Hosseinzadeh[3]

Accepted: 12 March 2022 / Published online: 8 April 2022

Abstract

DNA sequencing is one of the important sub-disciplines of bioinformatics, which has various applications in medicine, history, demography, and archaeology. De novo sequencing is the most challenging problem in this field. De novo sequencing is used for recognizing a new genome and for sequencing unknown parts of the genome such as in cancer cells. For assembling the genome, first, small fragments of the genome (called reads) that are located randomly on the genome are sequenced by the sequencing machine. Then, they are sent to the processing machine to be aligned on the genome. To sequence the whole genome, the reads must cover it entirely. The minimum number of reads to cover the genome is given by the Lander–Waterman's coverage bound. In this paper, we generalize the later scheme to de novo sequencing and reduce the total number of required bases by Lander–Waterman's coverage bound. We investigate the performance of the scheme such as the longest generated contig length, the execution time of the algorithm, different read lengths, and probability of error in the genome assembly. The results show the computational complexity and execution time of the algorithm in parallel on human genome with length 50,000 bases. We also show that the proposed method can generate contigs with 90 percent genome length.

Keywords DNA sequencing · Next-generation sequencing · De novo sequencing · Lander–Waterman's coverage bound

✉ Ashkan Farazin
a.farazin@grad.kashanu.ac.ir

[1] Department of Electrical Engineering, University of Isfahan, Isfahan, Iran

[2] Department of Solid Mechanics, Faculty of Mechanical Engineering, University of Kashan, Kashan, P.O. Box 87317-53153, Iran

[3] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada

1 Introduction

DNA sequencing has an essential role in molecular biology [1–6]; therefore, it is important to find the exact place of bases in the genome [7–11]. In the 70 s, the Sanger sequencing technology was introduced to sequencing the first genome [12, 13]. Then, in 2005 appearance of new sequencing technology is known as next-generation sequencing (NGS) [14].

The NGS technology can sequence hundreds of thousand fragments of DNA at the same time in fewer hours and lower cost [15–19]. But this technology cannot sequence the whole genome, so we have to divide the genome into short fragments to be sequenced [20, 21]. NGS derived a large number of reads from the same genome, so each base of the genome was sequenced many times [22–26]. The average number of times that a base sequenced is called coverage [21, 27, 28]. It produced additional information that it causes the increased complexity of the processing machine [29].

De novo sequencing method was used to assembly genome by sequence reads [30]. De novo sequencing in bioinformatics means merging and aligning DNA fragments to gain the original genome [31]. The importance of de novo sequencing is in the ability of technology to new assembly genomes. The complexity of de novo sequencing increases by genome length as an example, and max length of the genome is a repeat region in the genome which can be a simple repeat region within the genes or the genes themselves or part of them that cause more complexity [32].

In general, the complexity of assembly is affected by the number of reads, length of reads, length of repeat regions, and genome length [33]. Increasing the number of long reads helps to reduce assembly complexity, but the running time algorithm increases exponentially by the number and length of reads [34]. On the other hand, short reads assemble very easily, but repeat regions increase the complexity of assembly. For assembly, first, we produce samples from the genome and called them reads and then send them to sequencer machine and finally the process in the processing machine to assembly genome [35].

The method used for human genome assembly is a generalized reference [30] that eliminates the shortcomings of previous methods and improves the speed, accuracy, and sequencing time that is the novelty of the paper. This algorithm is comparable to the OLC algorithm, which uses graph theory for genome assembly. It works better for duplicate areas and identifies the exact location of these areas and prevents the creation of redundant paths in this graph.

We use the de novo sequencing algorithm in three categories: First, we use this method for reads that extract from i.i.d genome with equal probability, and then we use that for i.i.d genome with repeat region genome and finally we use it for a real genome that extracted from the human chromosome 19 [36]. At first for showing the accuracy of the algorithm, we extract the different number of reads lengths for showing the effect of reads on sequenced bases, running time algorithm, and max contig length and then we show the effect of different fragment lengths on the probability of assembly accuracy.

2 Material and methods

2.1 Lander–Waterman's coverage bound and feedback sequencing system

The number of reads produced by Lander–Waterman's bound is based on the length of the genome. Lander–Waterman bound is based on the coupon-collecting problem. That shows for the genome, the total number of bases for resequencing should be $O(GlnG)$ [37]. This is a result of the resequencing machine in which two parts of sequencing and processing are separate. If we take feedback from the processing machine, two parts of the machine work cooperatively. The number of bases will be reduced. In paper [37], the number of reads produced by Lander–Waterman's coverage bound for genome with length of G, so at least we need $N = \frac{G}{L}\ln(\frac{G}{L})$ reads and reduced number of sequenced bases to as low as $O(G)$. In this paper, we focus on the de novo sequencing problem. We have two cases for assembling reads:

1. Two segments of reads with length (↕) have common base, such as j^{th} read and $(j+1)^{\text{th}}$ read in Fig. 1.
2. Two segments of reads with length (↕) do not have a common base, such as i^{th} read and $(i+1)^{\text{th}}$ read in Fig. 2.

In the first case, we have reads which extension does not increase the coverage. But in the second case, we have read which extension on sequencing increases the coverage. While starting time of reads has an independent exponential distribution.

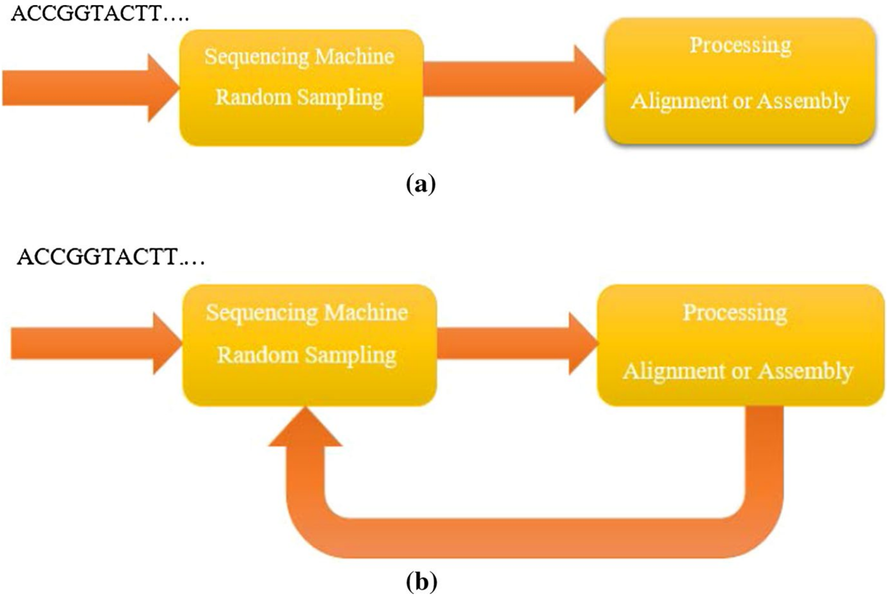

Fig. 1 **a** Common assembly method, **b** New assembly method in ref [37]

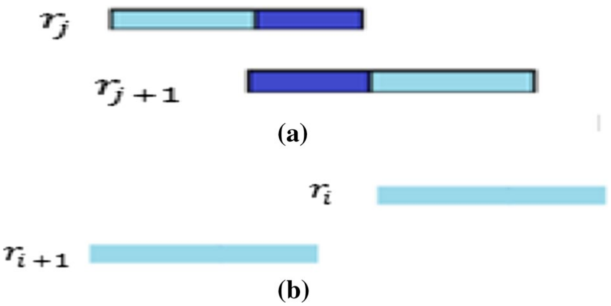

Fig. 2 Two cases for subsequent reads

We assume that the starting points of the readings are Poisson distribution with rate $\lambda = \frac{N}{G}$; starting time of reads has an exponential distribution.

We assume that B_i is random variable that state number of additional bases in i th reads and total number of additional bases is equal to: $B = \sum_{i=1}^{N} B_i; E\{B\} = NE\{B_j\}$.

For calculating $E\{B_j\}$ that $j+1$ th reads are x bases after starting j th reads. If $x \leq \updownarrow$, we have $B_j = \updownarrow - x$. Average bases for all reads are equal to [37]:

$$\mathrm{PE}\{\mathrm{B}\} = kG\left(1 - \frac{e^k}{e^{\frac{\mathrm{NL}}{G}}}\right) \tag{1}$$

Max value of the equation is one. So with $k=$cte, the number of additional read bases is from rank $\mathcal{O}(G)$. For example, by choice $N = G/\updownarrow$ Lander–Waterman's coverage is bound broken.

For calculating the number of additional bases in noisy mode, we use similar reasoning as free noise mode. So for calculating $E\{B_j\}$ for j th reads, we assume that $C_{\in}$ reads after these reads started from x th base that have Erlang distribution. An average number of bases that have overlap for every reads are equal to [37]:

$$E\{B\} = \left\{ \frac{1}{\left(C_\varepsilon - 1\right)!}\left(\left(k_n - C_\varepsilon\right)\gamma\left(C_\varepsilon, k_n\right) + k_n^{C_\varepsilon} e^{-k_n}\right) \right\} \times G \tag{2}$$

where is $k = \frac{N\updownarrow}{G}$. So for every fixed, k_n $E\{B\}$ is equal to $\mathcal{O}(G)$. Coverage bound of genome bases covered C_ε time is equal to [37]:

$$P \leq G \sum_{j=0}^{C_\varepsilon - 1} e^{-\lambda l} \frac{(\lambda l)^j}{j!} \tag{3}$$

2.2 Suggested method

In the sequencing problem, if we did not have a reference genome, we had to use de novo sequencing. The sequencing machine produced reads with length L. Base on

Lander–Waterman's coverage bound, NL total number of sequenced bases are equal to $O(G\ln G)$. In this case, the cover bound will be controlled by the processing machine. So processing machine can end sequencing in each base. In another word, sequencing machines start reading bases one by one from the DNA segment and its processing stops only by the stop commend of the processing machine.

2.3 Designing a de novo sequencing algorithm for i.i.d genome

In this method, the key strategy we use to end the sequencing is: At the first level, we found L by integer K in $\{1,\dots,L\}$ divided by $\updownarrow = \frac{L}{K}$ and we assume that $\updownarrow$ is integer. In the first level ($k=1$), sequencing machine separates first $\updownarrow$ bases of all reads that called fragment in collection $C = \left\{\mathcal{R}_1(\updownarrow), \mathcal{R}_2(\updownarrow), \dots, \mathcal{R}_N(\updownarrow)\right\}$. $\mathcal{R}_i(\updownarrow)$ means first $\updownarrow$ bases of i th reads. In other words, $\mathcal{R}_i(\updownarrow)$ is.

i th fragment of C collection. In the k th level ($k > 1$), we have two loops on i,j that compare part of i th and j th fragmets. If $\updownarrow - 1 - mer$ (substrings of length l-1 in reads) last base of i th fragment with $\updownarrow - 1 - mer$ first bases of jth fragment is equal, these two fragment were connected in the i.i.d genome [37]. So the machine connects these two fragments and replaces them with the i th fragment and deletes the jth fragment of C collection. So the sequencing of i th fragment ended, and the sequencing continues from the jth fragment and jth fragment replaced with i th fragment. In $(k+1)$ th level, if we had $L_{MAX} - k + 1 \geq k + 1$, we called $(\updownarrow + k - 1)$ th base of i th reads and connected to the i th fragment and repeat all level to achive contig. This algorithm is an exhibit in *Algorithm* de novo *Sequencing*.

For noisy reads, we had error rate parameter ϵ and parameter C_ϵ for coverage deep and correct probability of resequencing is P_ε. For extracting C_ε, the bases are goal that have max number of bases in that region. The probability of failure for noisy manner is equal to:

$$\begin{aligned} P\left(n_0 \le \max\left(n_1,n_2,n_3\right)\right) &= \sum_{j=0}^{c_i} P\left(j \le \max\left(n_1,n_2,n_3\right)\middle| n_0 = j\right) \times P\left(n_0 = j\right) \\ &\sum_{j=0}^{c_i} \sum_{n_1=m_1,n_2=m_2,n_3=c_i-m_1-m_2-j} \\ &\left[P\left(n_0 = j, n_1 = m_1, n_2 = m_2, n_3 = C_i - m_1 - m_2 - j\right)\right. \\ &\left.\times P\left(n_1 = m_1, n_2 = m_2, n_3 = C_i - m_1 - m_2 - j\right) \times P\left(n_0 = j\right)\right] \end{aligned} \tag{4}$$

As we have:

$$\begin{aligned} &P\left(n_1 = m_1, n_2 = m_2, n_3 = C_i - m_1 - m_2 - j\right) \\ &= \frac{\left(c_{i-j} n_1 = m_1, n_2 = m_2, n_3 = C_i - m_1 - m_2 - j\right)}{3^{C_{i-j}}} \end{aligned} \tag{5}$$

$$P\left(n_0 = j\right) = \left(\begin{array}{c} C_i \\ j \end{array}\right)(1-)^{j(C_i-j)} \tag{6}$$

$$P\left(n_0 \leq \max\left(n_1, n_2, n_3\right)\right) = \sum_{j=0}^{c_i} \sum_{n_1=m_1, n_2=m_2, n_3=c_i-m_1-m_2-j} \left[P\left(j \leq \max\left(n_1, n_2, n_3\right) \middle| n_0 = j, n_1 = m_1 n_2 = m_2, n_3 = C_i - m_1 - m_2 - j\right)\right] \times \left(\ C_i j\ \right)(1- \in)^j \in^{(C_i-j)} \times \frac{\left(\begin{array}{c} c_{i-j} \\ n_1 = m_1, n_2 = m_2, n_3 = C_i - m_1 - m_2 - j \end{array}\right)}{3^{C_{i-j}}} \tag{7}$$

For example, if $P_\varepsilon = 10^{-4}$,$\varepsilon = 0.07$, C_ε is equal to: 8

```
Algorithm de novo Sequencing
Input: $N$ reads with size $L_i$ of target genome
    Output: contig $C$
Let $C = [\,]$; $\ell = logG$;
for ($k = 1$ to $L - \ell + 1$) do
      if  $k = 1$ then
        sequence the first $\ell$ bases of all reads;
        set C={$R_1$ $(l)$, $R_2$ $(l)$, …. ‘ R$_N$(l)} ;
      else
         sequence the $(l + k - 1)_{th}$ base of  the remaining reads and add it to the
          corresponding fragment in C;
      End
      For all remaining $reads_i$  do
          For all remaining $reads_j$  do
              If the last $l - 1$  bases of $fragment_i$ are equal to the first $\ell - 1$ bases of
                $fragment_j$ then
               $fragment_i$← merge $fragment_i$, $fragment_j$;
                 determine length of merge fragment and keep max length
               delete $fragment_j$ and $reads_j$
               $reads_i$←$reads_j$
              End
          End
      End
End
```

To investigate the probability of producing contig with length less than 90 percent of the genome, we need the number of reads (*N*), length of reads (*L*), and length of the genome (*G*). So for genome length (*G*) and reads length (*L*) base on Lander–Waterman's bound we can have the number of reads (*N*). We assume that the starting points of the readings are Poisson distribution with rate = $\frac{N}{G}$, starting time of reads compared to the previous read is exponential distribution and the time interval between the start of two consecutive reads is independent of each other.

In de novo sequencing algorithm, we need $\updownarrow$ common base between two successive reads. So probability of start reads in two periods $(L - \updownarrow, L]$ and $[1, L - \updownarrow]$ is probability of fail and probability of success, respectively.

Probability of starting reads in $(L - \updownarrow, L]$ is equal to:

$$P(x > L - \ell) = e^{\frac{N}{G}(L-\ell)} \tag{8}$$

Probability of starting reads in $[1, L - \ell]$ is equal to:

$$P(x \leq L - \ell) = 1 - e^{\frac{N}{G}(L-\ell)} \tag{9}$$

So probability of success for all reads is equal to:

$$P_{\text{success}} = \left(1 - e^{\frac{N}{G}(L-\ell)}\right)^N \tag{10}$$

And probability of fail is equal to:

$$P_{\text{fail}} = 1 - P_{\text{success}} = 1 - \left(1 - e^{\frac{N}{G}(L-\ell)}\right)^N \tag{11}$$

We cannot get the number of reads for getting failure probability under one base on L, $\updownarrow$, and G from equal (11). Therefore, for extracting the number of reads needed to achieve the probability of failure, we show equal (11) base on number of reads [1, 5000] for $G = 20000$, $\updownarrow = 14$, and reads length $L = [100, 300, \ldots, 1500]$ as shown in Fig. 3. We extract value of N from the curve for $P_{fail} = 0.1$ and $P_{fail} = 0.01$ and compare with Lander–Waterman's bound (Fig. 4).

In Fig. 3, we indicate the failure probability of the algorithm base on the number of reads for length read $L = [100, 300, \ldots, 1500]$. As we see in this figure for $L = 100$, we have much error to make contig, and therefore, we only have short contig.

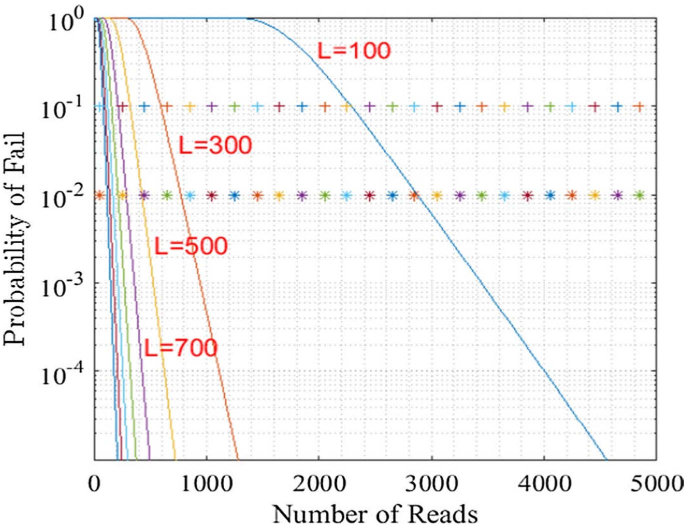

Fig. 3 Failure probability of the algorithm according to the number reads

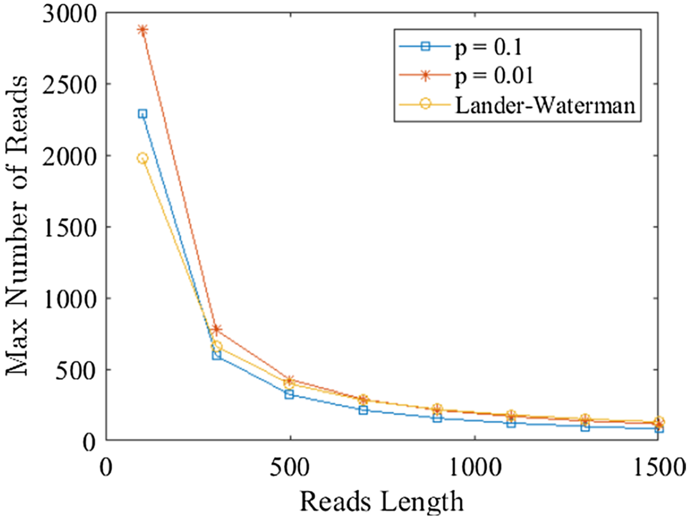

Fig. 4 Max number of reads base on length of reads

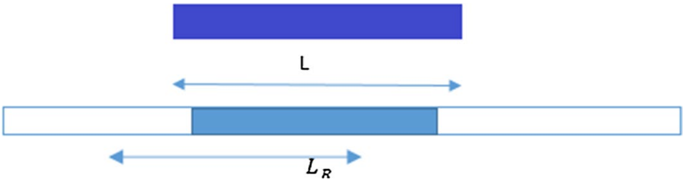

Fig. 5 Bridging over repetitive areas

2.4 i.i.d genome with repeat region

In this section, we try to study the i.i.d genome with repeat region. Our goal is to assembly reads of the i.i.d genome with repeat region. The challenge of this assembly is the repeat region that can cause mistakes in assembly. The way we offer to solve this problem is using a long read and using fragment length longer than the repeat region (Fig. 5). In this way by use of bases around the repeat region, we can exactly locate reads in de novo sequencing.

2.5 Real genome

Real genomes have different repeat lengths with different lengths. In this section, we try to generalize the way that is shown in Sect. 2.2. We use this way only with a long read for real genomes. Lander–Waterman's state coverage bound for i.i.d genome. Based on coverage bound, minimum number of reads for coverage genome is equal to $N \approx \frac{G\ln G}{L}$. In other words, we need NL = $G\ln G$ bases for coverage genome. By generalization way [37] for de novo sequencing, first, we sequenced bases of all reads and then the processor connects all the fragments that are uniquely connected to make contig. Unlock read that does not overlap with right-hand read is sent to the processor for extending one base in reads. This loop repeats until getting max contig length, so the number of bases in sequencing level decreases and we have max contig length equal to the real genome.

3 Result and discussion

Check the algorithm on the i.i.d genome with length G‘ N reads with length l base on the Lander–Waterman bound. We have an example of producing reads on a fragment with length 20,000 in Fig. 6 that $L = 500$ and the number of reads is $N = 400$ from $N \approx \frac{G}{L}\mathrm{Ln}G \approx 400$.

3.1 Effect of number of reads

At first, we show the algorithm on i.i.d genome with length 20,000 for N=[200, 250, … 600] for specific read length $L = 1700$, fragment length $\updownarrow$ = log G ≈ 14 on windows 8 with MATLAB 2017a and CPU core i5-2410 MHZ—RAM 4 Gbyte. All simulations are the result of averaging on the 20 times running the algorithm. It should be noted that many codings were done in the supercomputer center of the University of Isfahan. The systems of this center are core i24 with 64 GB of RAM.

In Fig. 7, we show the number of sequencing bases based on the number of reads. As we see in Fig. 7, by increasing the number of reads the number of sequencing bases increased. But the speed of that become decrease for $N \geq 400$.

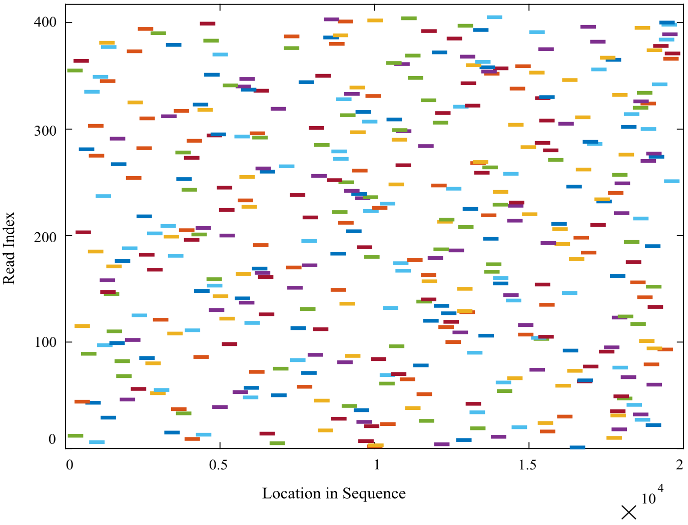

Fig. 6 The location of reading extracted from the i.i.d genome

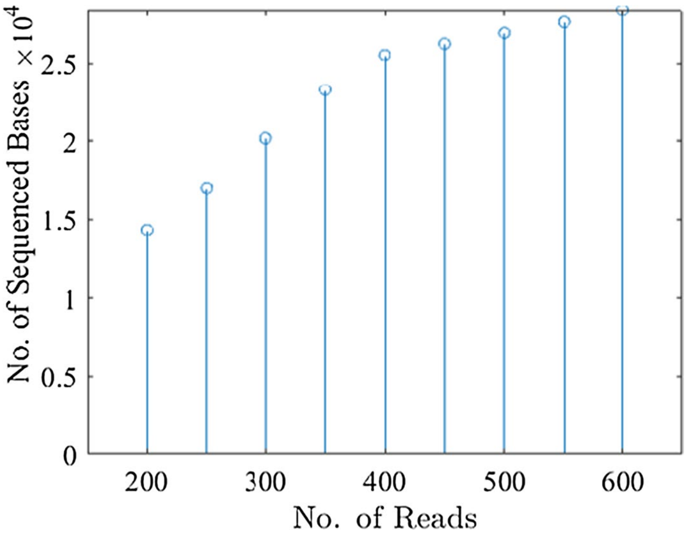

Fig. 7 Effect of number of reads on number of bases

In Fig. 8, we show algorithm execution time base on the number of reads. As we see in Fig. 8, by increasing the number of reads algorithm execution time linearly increased.

In Fig. 9, we show max contig length base on number of reads. As we see in Fig. 9, max contig length is in $N = 350$,$N = 400$, and $N = 450$ that is almost equal to i.i.d genome length (20,000). As summarizing, we can say by running

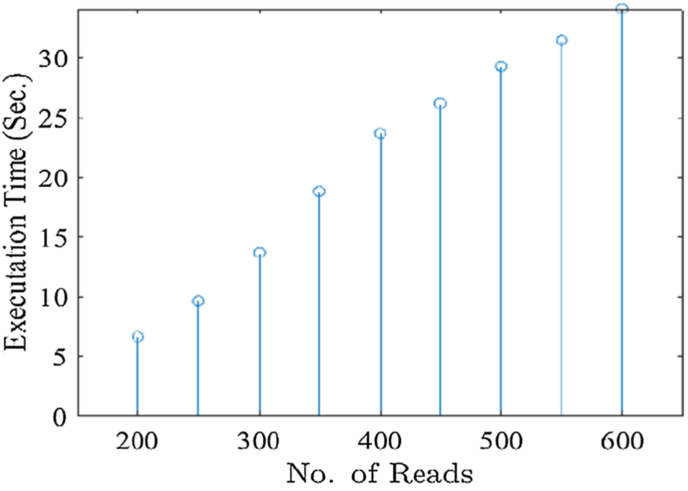

Fig. 8 Effect of the number of reads on execution time

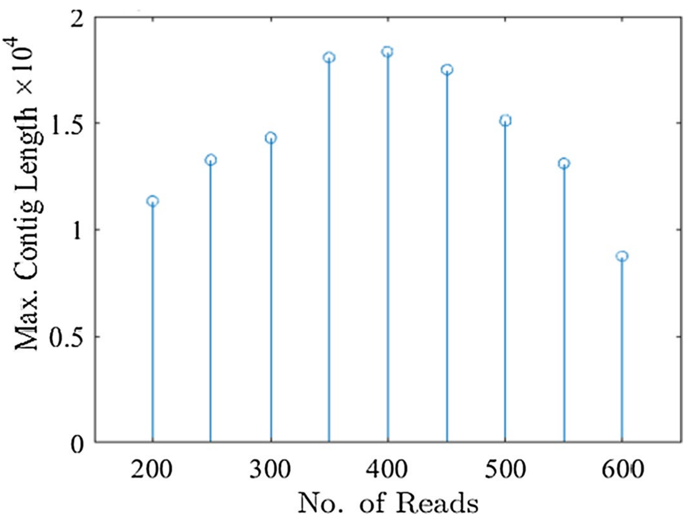

Fig. 9 Effect of number of reads on max contig length

the algorithm on reads with length $L = [300, 700, \dots, 1700]$, and we have max contig length equal to genome length.

3.2 Effect of reads Length (L)

To examine the effect of length reads, we should run the algorithm on i.i.d genome with length 20,000 for different lengths such as $L = [300, 500, \dots, 1700]$ and fragment length obtained from $\updownarrow = \log G$ [37]:

In Fig. 10, we show the number of reads base on reads length. In Fig. 10, our goal is to max length contig achieve 90 percent of genome length. It shows that the algorithm cannot make this contig for $L \leq 200$, so we have to show the number of reads base on reads length.

3.3 Effect of fragment length ($\updownarrow$)

We run algorithm on i.i.d genome with length 20,000 for reads length $L = 500$, number of reads $N = \frac{G}{L} \ln G$, and fragment length $\updownarrow = \{5, 6, 7, 8, \dots, 16\}$. We should say that all simulations are the result of averaging on 20 times running the algorithm.

In Fig. 11, we show the number of sequenced bases based on fragment length. As we see in Fig. 11 by increasing the number of sequenced bases fragment length increased. But the increased speed for $l \geq 400$ decreased.

In Fig. 12, we show execution time base on fragment length. As we see in the figure by increasing execution time, fragment length increased.

In Fig. 13, we show max contig length base on fragment length. As we see in figure, max contig length is equal to genome length for $\updownarrow \geq 9$.

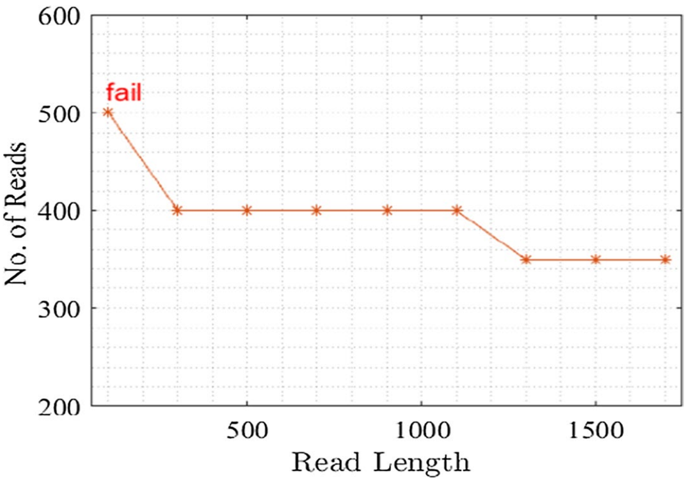

Fig. 10 Number of reads base on reads length

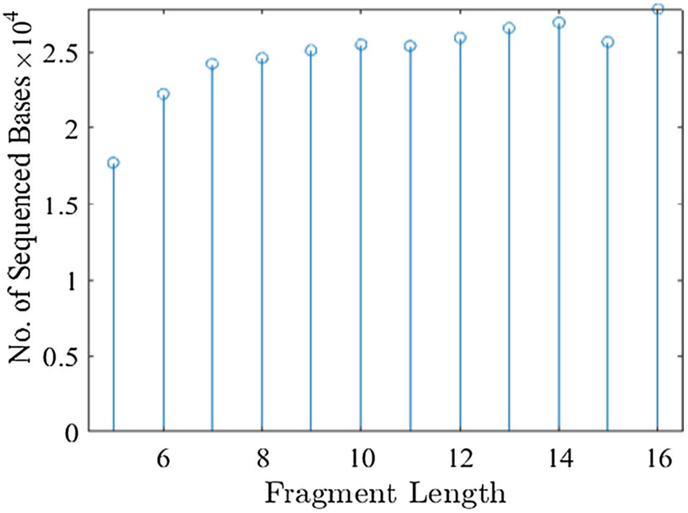

Fig. 11 Effect of number of sequenced bases on fragment length

In Fig. 14, we have the probability of de novo sequencing base on fragment length. As we see in Fig. 14, the probability of de novo sequencing for $\updownarrow \geq 8$ is equal to one.

3.4 Result of simulation on i.i.d genome with repeat region

We run the algorithm on the i.i.d genome with repeat region ($G = 20000$), the connection of fragments on the reads is shown with the dashed line. In Fig. 15, we show the result of running the algorithm on a fragment with a length of 20,000

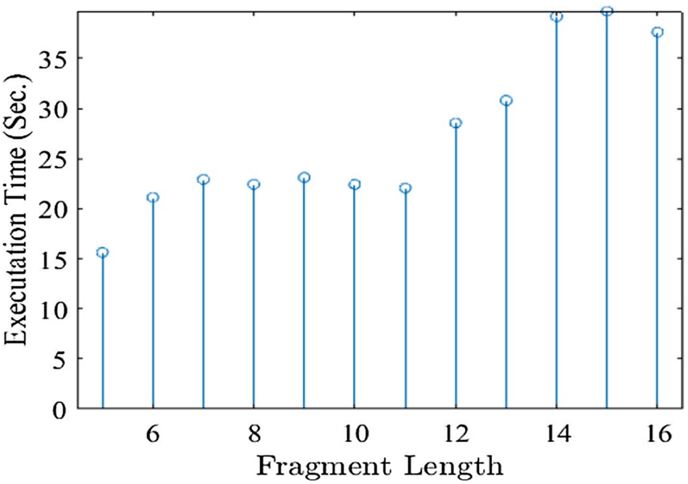

Fig. 12 Effect of fragment length on execution time

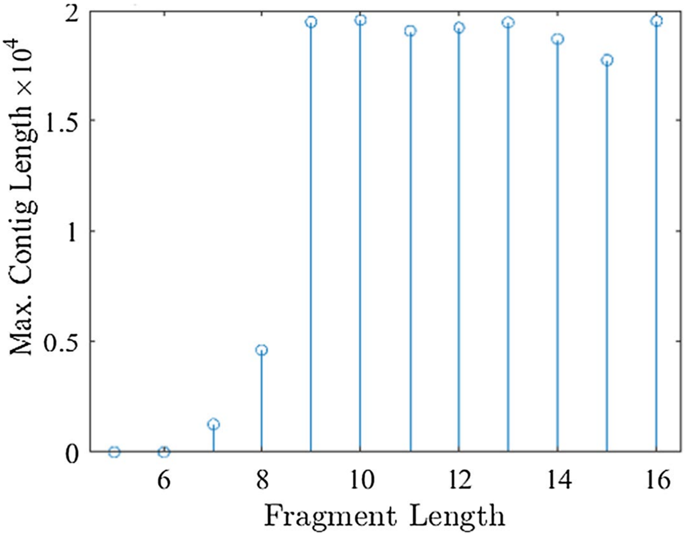

Fig. 13 Effect of fragment length on max contig length

that has a repeat region with length $L_R = 100$. This genome has $2 \times 50 = 100$ repeat region with a length of 50 base. The number of repeat genomes is $N_r = 50$ that each repeat region twice in the genome. In this genome, we should use long reads for correct connecting reads, and also fragment length in this genome should be longer than repeat region length $\updownarrow = \log G + L_r \approx 64$, so the repeat region does not have an effect on assembly. All simulations are the result of averaging on the 20 times running the algorithm.

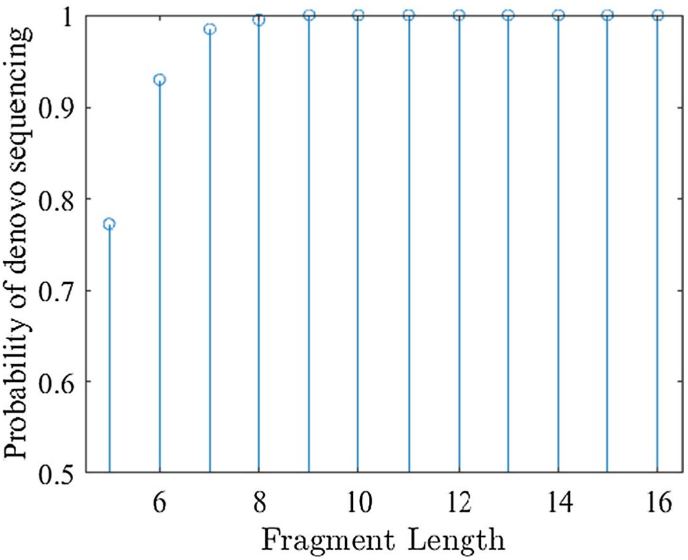

Fig. 14 Effect of fragment length on the probability of de novo sequencing

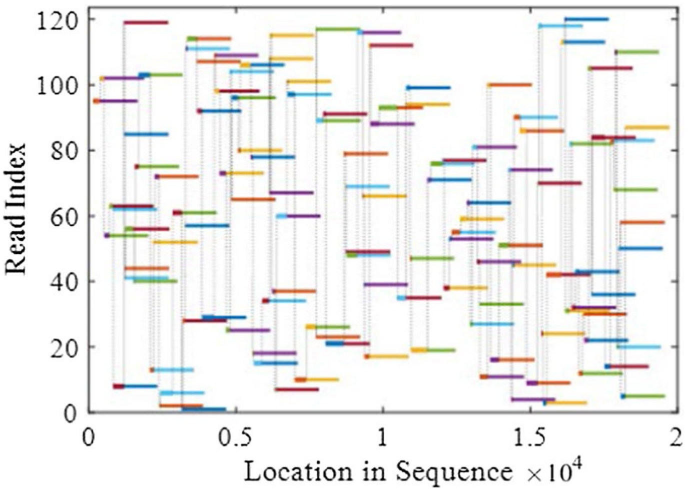

Fig. 15 The location of reading extracted from the i.i.d genome with repeat region

3.5 Effect of number of reads

At first, we show algorithm on i.i.d genome with repeat region with length 20,000 for N=[200, 250, … , 600] for specific read length $L = 1700$ and fragment length $\updownarrow =$ $\log + L_r \approx 64$. All simulations are the result of averaging on the 20 times running the algorithm.

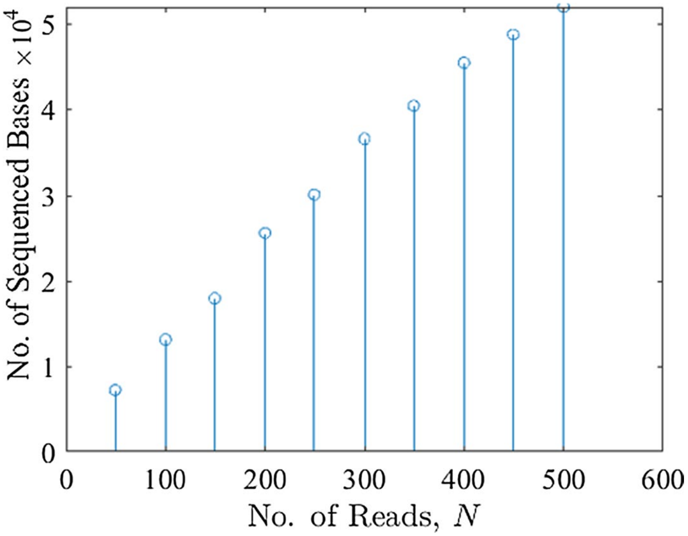

Fig. 16 Effect of number of reads on number of bases

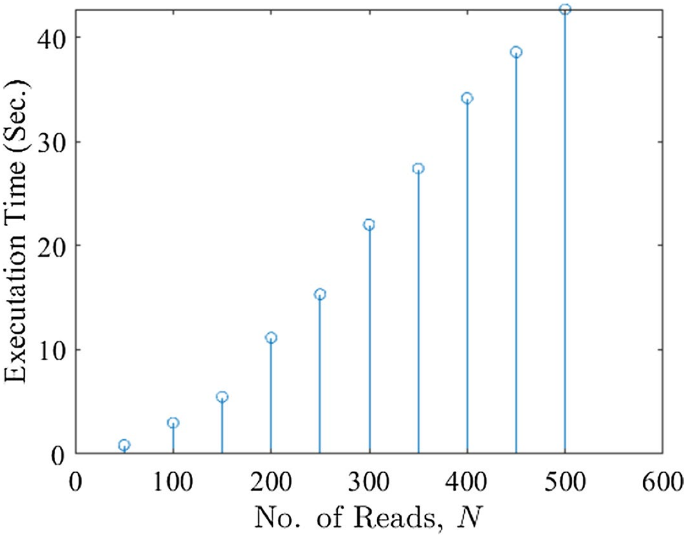

Fig. 17 Effect of number of reads on execution time

In Fig. 16, we show the number of sequencing bases based on the number of reads. As we see in the figure, by increasing the number of reads the number of sequencing bases linearly increased.

In Fig. 17, we show algorithm execution time base on the number of reads. As we see in the figure, by increasing the number of reads algorithm execution time linearly increased.

In Fig. 18, we show max contig length base on the number of reads. As we see in the figure, max contig length is in $N = 400$ and $N = 450$ that is almost equal to i.i.d genome with a repeat region ($G = 20{,}000$). As summarizing, we can say by running the algorithm on reads with length $L = [300, 700, \dots, 1700]$, and we have max contig length equal to genome length.

3.6 Effect of fragment length ($\updownarrow$)

We run an algorithm on i.i.d genome with repeat region ($G = 20{,}000$) for reads length $L = 500$, number of reads $N = \frac{G}{L}\ln G$, and fragment length $\updownarrow = \{6, 7, \dots, 65\}$. We should say that all simulations are the result of averaging on 20 times running the algorithm.

In Fig. 19, we show the number of sequenced bases based on fragment length. As we see in Fig. 19, increasing the number of sequenced base fragment lengths increased.

In Fig. 20, we show execution time base on fragment length. As we see in Fig. 20, by increasing execution time fragment length increased.

In Fig. 21, we show max contig length base on fragment length. As we see in Fig. 21, max contig length is equal to genome length for $\updownarrow \geq 50$.

In Fig. 22, we have the probability of de novo sequencing base on fragment length. As we see in figure, probability of de novo sequencing for $\updownarrow \geq 50$ is equal to one.

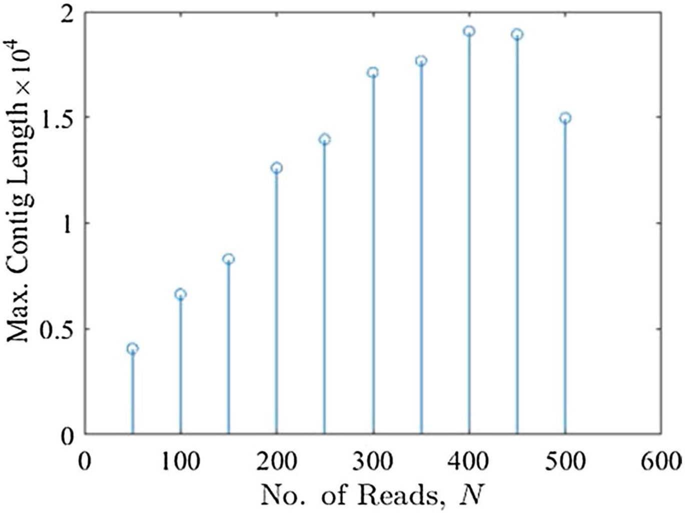

Fig. 18 Effect of number of reads on max contig length

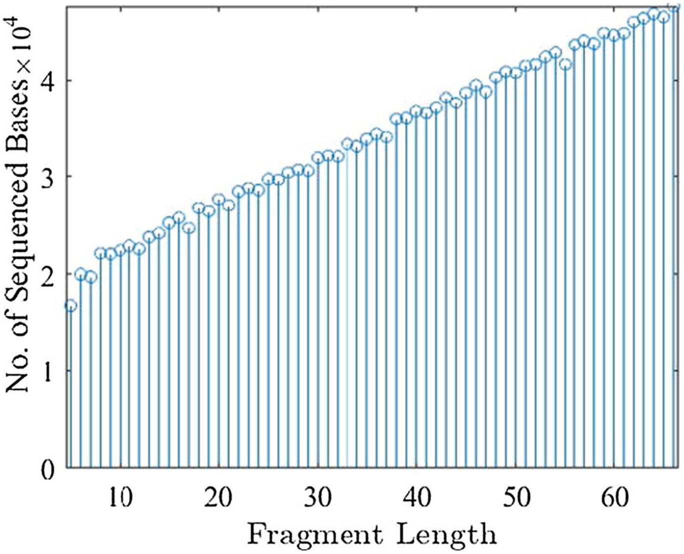

Fig. 19 Effect of fragment length on number of sequenced bases

3.7 Result of simulation on human genome

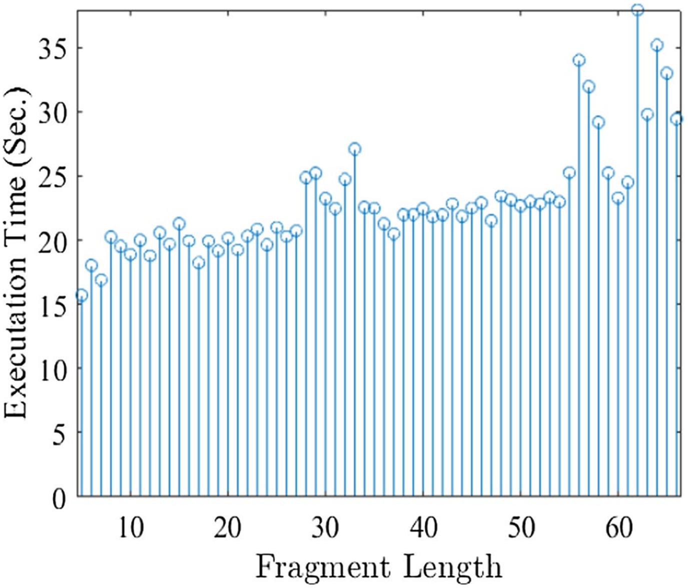

Fig. 20 Effect of fragment length on execution time

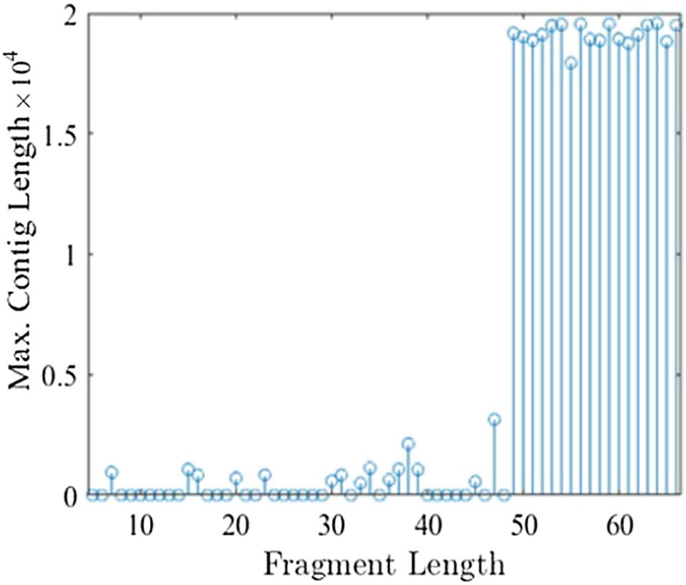

Fig. 21 Effect of fragment length on max contig length

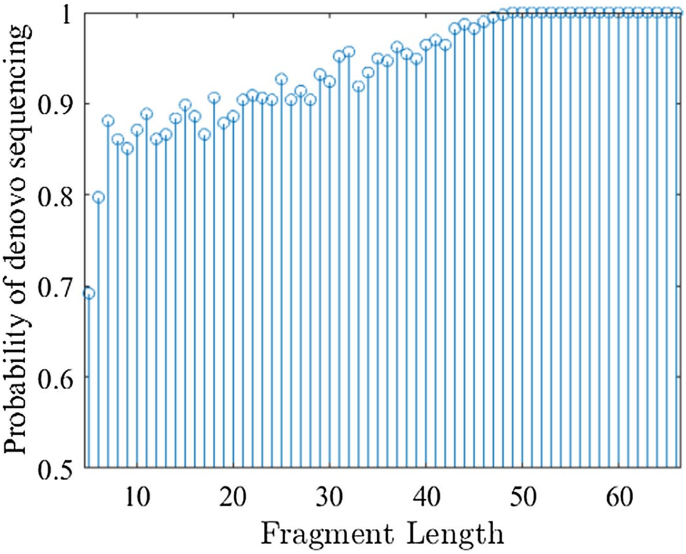

Fig. 22 Effect of fragment length on probability of de novo sequencing

We run the algorithm in parallel on the human genome with a length of 50,000. We extract this part of the genome from human chromosome 19. The server has full information about the genome, length of repeat genome, and place of them on the genome. All the information is taken from the reference [38]. The web site provides complete information about the genome, the length of duplicate regions, their location, and type.

3.8 Effect of read length

At first, we show algorithm in parallel on human genome with length 50,000 for different reads length for specific read length $N = \text{round}(\text{G} * \log(\text{G})/\text{L\_maxfragment length} \updownarrow = \log G + L_r \approx 64$. All simulations are result of averaging on the 20 times running the algorithm.

In Fig. 23, we show number of sequencing bases based on read length. As we see in Fig. 23, by increasing the read length the number of sequencing bases linearly decreased.

In Fig. 24, we show algorithm execution time base on read length. As we see in the figure, by increasing the read length algorithm execution time linearly decreased.

In Fig. 25, we show max contig length base on read length. As we see in figure, max contig length is in $L = 500$ that is almost equal to human genome ($G = 50{,}000$). As summarizing, we can say by running algorithm in parallel on reads with length $L = [500, 1000, \ldots, 5000]$, and we have max contig length equal to genome length on shorter read, execution time, and sequenced bases by increasing the read length decreased.

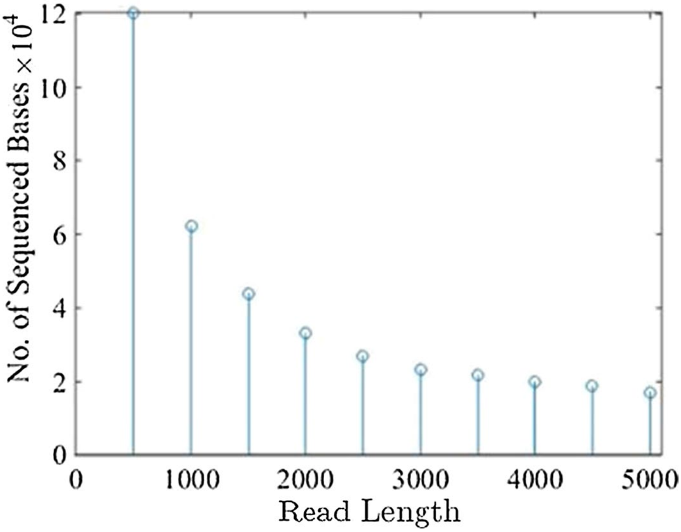

Fig. 23 Effect of read length on number of bases

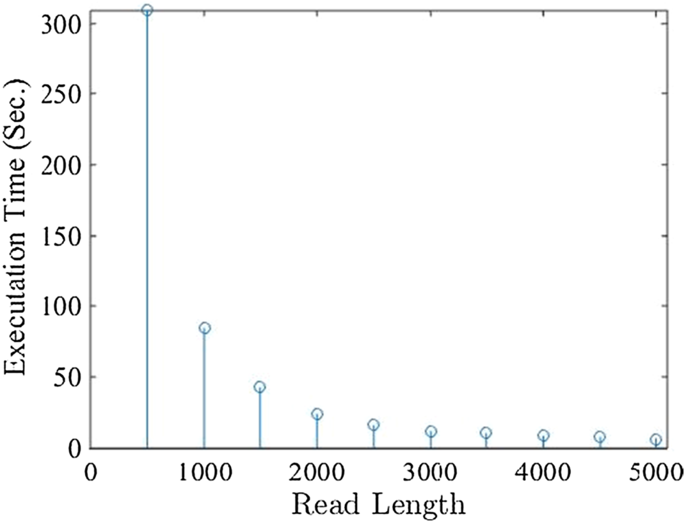

Fig. 24 Effect of read length on execution time

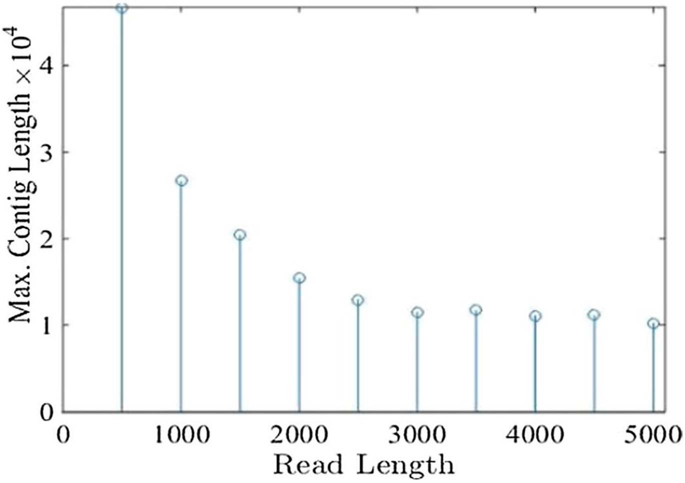

Fig. 25 Effect of read length on max contig length

3.9 Effect of fragment length (ℓ)

We run algorithm in parallel on the human genome with length 50,000 for reads length $L = 1000$, number of reads $N = \frac{G}{L}\ln G$, and fragment length $\updownarrow = \{15, \dots, 90\}$. We should say that all simulations are result of averaging on 20 times running the algorithm.

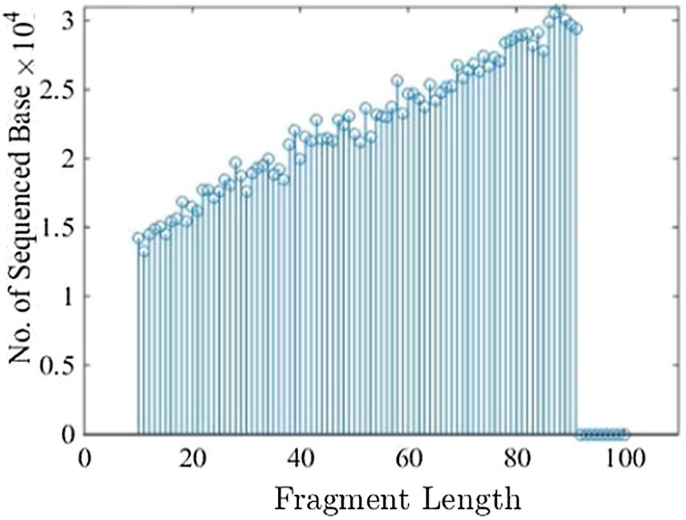

Fig. 26 Effect of fragment length on number of sequenced bases

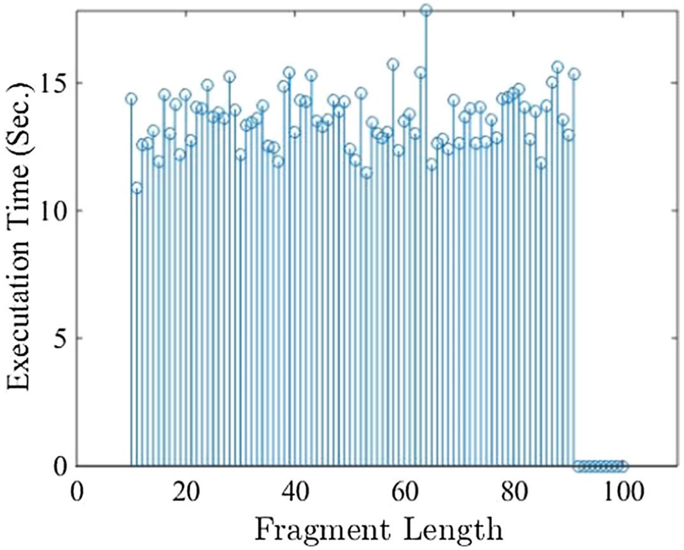

Fig. 27 Effect of fragment length on execution time

In Fig. 26, we show the number of sequenced bases based on fragment length. As we see in Fig. 26, by increasing the number of sequenced bases fragment length increased.

In Fig. 27, we show execution time base on fragment length. As we see in the figure, by increasing execution time fragment length did not increase.

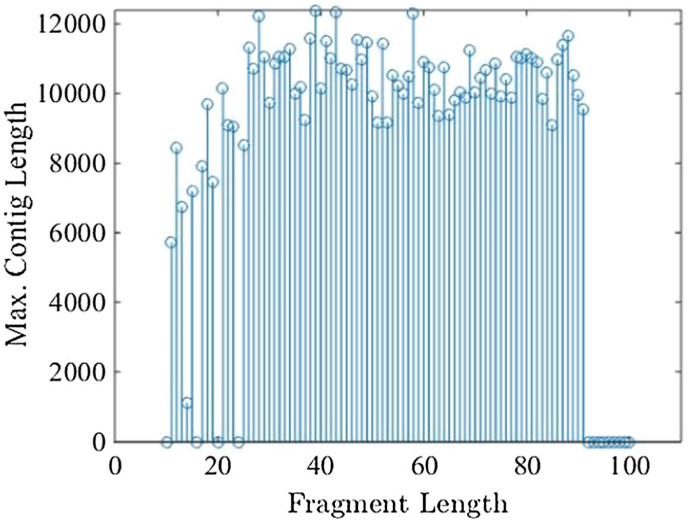

Fig. 28 Effect of fragment length on max contig length

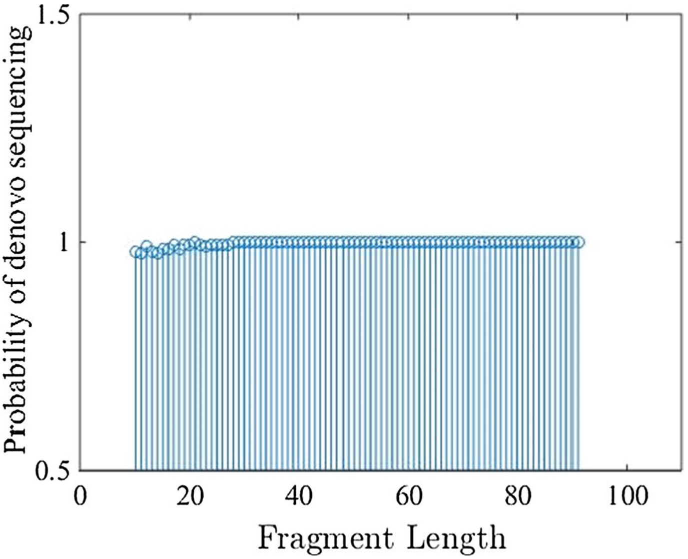

Fig. 29 Effect of fragment length on the probability of de novo sequencing

In Fig. 28, we show max contig length base on fragment length. As we see in Fig. 28, max contig length is equal to genome length for $\updownarrow > 14$.

In Fig. 29, we have the probability of de novo sequencing base on fragment length. As we see in Fig. 29, probability of de novo sequencing for $\updownarrow \geq 10$ is equal to one.

4 Conclusion

The amount of information generated by the sequencer machine at a great rate is increasing. We need a strong processing machine for extracting information from data. Sequencing machine has lots of additional data that need lots of money and time to extract useful information and delete additional information at the processing level.

Based on Lander–Waterman's coverage bound, sequencing machine for genome with length G and reads with length L, at least needs N read that called lower Lander–Waterman's coverage bound. In other words, we need $N \approx \frac{G}{L} \ln G$ reads to coverage genome with probability 1-ε. The total number of sequenced bases by machine is NL that is equal to $G \ln G$. These results show that sequenced machine is independent from processing machine. On the other hand, sequencing machine sequenced total number of required bases for coverage genome and then reads send to the processing machine.

In this paper, we try to generalize presented method for de novo sequencing. At first, we use this method for i.i.d genome with length G and N reads with length L from that. The result of running this algorithm on this genome prohibits a decrease in running algorithm time and number of sequencing bases with the increase in reads length.

Max contig length for reads with $L \geq 500$ usually has equal length with genome length. By examining the probability of sequencing error of algorithm with Lander–Waterman's bound in $p = 0.01$, we understand that Lander–Waterman's bound is not so accurate for $L < 500$, but other length has the equal result.

Then, we run the algorithm for i.i.d genome with repeat region. This genome has $100 = 50 \times 2$ repeat region with length of 50 bases. For genome assembly, we assume fragment length $\updownarrow = L_R + \log G$ to reduce the ambiguity of repeat region. The result of running this algorithm on i.i.d genome with repeat length prohibits decrease in the running algorithm time and number of sequencing bases with the increase in reads length.

Finally, we run a de novo sequencing algorithm in parallel on a real human genome with a length of 50,000. With increase in length of reads, running algorithm time and sequenced bases reduced. An increase in fragment length causes increase in accuracy in genome assembly.

Acknowledgements The authors would like to thank the referees for their valuable comments. Also, we would like to extend their gratitude for the support provided by the Isfahan University, Isfahan, Iran

Funding The authors received no financial support for the research, authorship, and publication of this article.

Declarations

Conflict of interest No conflict of interest exists in the submission of this article, and the article was approved by all the authors.

Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Farazin A, Sahmani S, Soleimani M et al (2021) Effect of hexagonal structure nanoparticles on the morphological performance of the ceramic scaffold using analytical oscillation response. Ceram Int 47:18339–18350. https://doi.org/10.1016/j.ceramint.2021.03.155
2. Farazin A, Akbari Aghdam H, Motififard M et al (2019) A polycaprolactone bio-nanocomposite bone substitute fabricated for femoral fracture approaches: molecular dynamic and micromechanical investigation. J Nanoanalysis 6:172–184
3. Farazin A, Aghadavoudi F, Motififard M et al (2021) Nanostructure, molecular dynamics simulation and mechanical performance of PCL membranes reinforced with antibacterial nanoparticles. J Appl Comput Mech 7:1907–1915
4. Kazeroni ZS, Telloo M, Farazin A et al (2021) A mitral heart valve prototype using sustainable polyurethane polymer: fabricated by 3D bioprinter, tested by molecular dynamics simulation. AUT J Mech Eng 5:109–120
5. Farazin A, Mohammadimehr M, Ghasemi AH, Naeimi H (2021) Design, preparation, and characterization of CS/PVA/SA hydrogels modified with mesoporous Ag 2 O/SiO 2 and curcumin nanoparticles for green, biocompatible, and antibacterial biopolymer film. RSC Adv 11:32775–32791. https://doi.org/10.1039/D1RA05153A
6. Farazin A, Mohammadimehr M (2021) Computer modeling to forecast accurate of efficiency parameters of different size of graphene platelet, carbon, and boron nitride nanotubes: a molecular dynamics simulation. Comput Concr 27:111
7. Chen T-C, Elveny M, Surendar A et al (2021) Developing a multilateral-based neural network model for engineering of high entropy amorphous alloys. Model Simul Mater Sci Eng 29:065019. https://doi.org/10.1088/1361-651X/ac1774
8. Xuefeng L, Lei H, Dongmei C (2021) Simulation of pit interactions of multi-pit corrosion under an anticorrosive coating with a three-dimensional cellular automata model. Model Simul Mater Sci Eng 29:065018. https://doi.org/10.1088/1361-651X/ac13cb
9. M J, Bhattacharya A, (2021) A 2D model for prediction of nanoparticle distribution and microstructure evolution during solidification of metal matrix nanocomposites. Model Simul Mater Sci Eng 29:065017. https://doi.org/10.1088/1361-651X/ac165c
10. Bhardwaj U, Sand AE, Warrier M (2021) Comparison of SIA defect morphologies from different interatomic potentials for collision cascades in W. Model Simul Mater Sci Eng 29:065015. https://doi.org/10.1088/1361-651X/ac095d
11. Wu L, Zhu Y, Wang H, Li M (2021) Crystal–melt coexistence in fcc and bcc metals: a molecular-dynamics study of kinetic coefficients. Model Simul Mater Sci Eng 29:065016. https://doi.org/10.1088/1361-651X/ac13c9
12. Wong TN, Chan LCF, Lau HCW (2003) Machining process sequencing with fuzzy expert system and genetic algorithms. Eng Comput 19:191–202. https://doi.org/10.1007/s00366-003-0260-4
13. Medland AJ (1994) A proposed structure for a rule-based description of parametric forms. Eng Comput 10:155–161. https://doi.org/10.1007/BF01198741
14. van Dijk EL, Auger H, Jaszczyszyn Y, Thermes C (2014) Ten years of next-generation sequencing technology. Trends Genet 30:418–426. https://doi.org/10.1016/j.tig.2014.07.001
15. Huang L-T, Wei K-C, Wu C-C et al (2021) A lightweight BLASTP and its implementation on CUDA GPUs. J Supercomput 77:322–342. https://doi.org/10.1007/s11227-020-03267-1
16. Chang W-L, Huang S-C, Lin KW, Ho M (2011) Fast parallel DNA-based algorithms for molecular computation: discrete logarithm. J Supercomput 56:129–163. https://doi.org/10.1007/s11227-009-0347-9
17. Fernández L, Pérez M, Orduña JM (2019) Visualization of DNA methylation results through a GPU-based parallelization of the wavelet transform. J Supercomput 75:1496–1509. https://doi.org/10.1007/s11227-018-2670-5
18. Ahmed M, Ahmad I, Ahmad MS (2015) A survey of genome sequence assembly techniques and algorithms using high-performance computing. J Supercomput 71:293–339. https://doi.org/10.1007/s11227-014-1297-4

19. González-Álvarez DL, Vega-Rodríguez MA, Rubio-Largo Á (2014) Parallelizing and optimizing a hybrid differential evolution with Pareto tournaments for discovering motifs in DNA sequences. J Supercomput 70:880–905. https://doi.org/10.1007/s11227-014-1266-y
20. Wang F, Yu S, Yang J (2010) Robust and efficient fragments-based tracking using mean shift. AEU - Int J Electron Commun 64:614–623. https://doi.org/10.1016/j.aeue.2009.04.004
21. Fang J, Yang J, Liu H (2011) Efficient and robust fragments-based multiple kernels tracking. AEU - Int J Electron Commun 65:915–923. https://doi.org/10.1016/j.aeue.2011.02.013
22. Chen D, Li Y, Wang Y, Xu J (2021) LncRNA HOTAIRM1 knockdown inhibits cell glycolysis metabolism and tumor progression by miR-498/ABCE1 axis in non-small cell lung cancer. Genes Genomics 43:183–194. https://doi.org/10.1007/s13258-021-01052-9
23. Lee JH, Kim J, Kim H et al (2021) Massively parallel sequencing of 25 short tandem repeat loci including the SE33 marker in Koreans. Genes Genomics 43:133–140. https://doi.org/10.1007/s13258-020-01033-4
24. Nguyen TH, Nguyen N-L, Vu CD et al (2021) Identification of three novel mutations in PCNT in vietnamese patients with microcephalic osteodysplastic primordial dwarfism type II. Genes Genomics 43:115–121. https://doi.org/10.1007/s13258-020-01032-5
25. Cui J, Wang J, Shen Y, Lin D (2021) Suppression of HELLS by miR-451a represses mTOR pathway to hinder aggressiveness of SCLC. Genes Genomics 43:105–114. https://doi.org/10.1007/s13258-020-01028-1
26. Srikulnath K, Singchat W, Laopichienpong N et al (2021) Overview of the betta fish genome regarding species radiation, parental care, behavioral aggression, and pigmentation model relevant to humans. Genes Genomics 43:91–104. https://doi.org/10.1007/s13258-020-01027-2
27. Steele KA, Quinton-Tulloch MJ, Amgai RB et al (2018) Accelerating public sector rice breeding with high-density KASP markers derived from whole genome sequencing of indica rice. Mol Breed 38:38. https://doi.org/10.1007/s11032-018-0777-2
28. Monjezi M, Baghestani M, Shirani Faradonbeh R et al (2016) Modification and prediction of blast-induced ground vibrations based on both empirical and computational techniques. Eng Comput 32:717–728. https://doi.org/10.1007/s00366-016-0448-z
29. Tandis E, Assareh E (2017) Inverse design of airfoils via an intelligent hybrid optimization technique. Eng Comput 33:361–374. https://doi.org/10.1007/s00366-016-0478-6
30. Mishra VK, Bajaj V, Kumar A et al (2017) An efficient method for analysis of EMG signals using improved empirical mode decomposition. AEU - Int J Electron Commun 72:200–209. https://doi.org/10.1016/j.aeue.2016.12.008
31. Price AL, Jones NC, Pevzner PA (2005) De novo identification of repeat families in large genomes. Bioinformatics 21:i351–i358. https://doi.org/10.1093/bioinformatics/bti1018
32. Du X, Servin B, Womack JE et al (2014) An update of the goat genome assembly using dense radiation hybrid maps allows detailed analysis of evolutionary rearrangements in Bovidae. BMC Genomics 15:625. https://doi.org/10.1186/1471-2164-15-625
33. Chin C-S, Alexander DH, Marks P et al (2013) Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. Nat Methods 10:563–569. https://doi.org/10.1038/nmeth.2474
34. Ye C, Hill CM, Wu S et al (2016) DBG2OLC: efficient assembly of large genomes using long erroneous reads of the third generation sequencing technologies. Sci Rep 6:31900. https://doi.org/10.1038/srep31900
35. O'Rawe J, Jiang T, Sun G et al (2013) Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing. Genome Med 5:28. https://doi.org/10.1186/gm432
36. Grimwood J, Gordon LA, Olsen A et al (2004) The DNA sequence and biology of human chromosome 19. Nature 428:529–535. https://doi.org/10.1038/nature02399
37. Nashta-ali D, Motahari SA, Hosseinkhalaj B (2016) Breaking Lander-Waterman's Coverage Bound. PLoS ONE 11:e0164888. https://doi.org/10.1371/journal.pone.0164888
38. Genome Browser Gateway. https://genome-asia.ucsc.edu/