Alignment of PacBio reads

Furaha Damien[11]

[1] McGill University

Monträal, Quäbec, Canada

Email: :furaha.damien@mail.mcgill.ca

Abstract — Genomic sequencing is an instrumental procedure in biology and bioinformatics. The advent of rapid DNA sequencing methods has greatly accelerated biological and medical research and discovery. With the human genome being 3 Billion bases long, several methods have been developed to sequence this genome at a low cost, yet with high and high throughput. However, most of the methods fail to provide a combinations of low cost, high-throughput and high accuracy. One such method is the Single-molecule real-time sequencing technique by Pacific Bioscience. This method has the ability to sequence longer reads of up to 20,000 bases. However, its higher error rate is a major obstacle. In this paper, I propose a new algorithm for increasing the accuracy of highly-contiguous de novo[2] assemblies using PacBio sequencing. In the algorithm I use the Needlemann-Wusnch sequence alignment algorithm as the underlying algorithm and shortgun sequencing[5] to perform assembly based on the overlap-layout-consensus approach. I develop a ranking algorithm that given n sequences, for every single sequence, ranks the most probably suffix and prefix sequence for that sequence that generates most optimal genome assembly. Using a variation of the Needleman-Wusnch algorithm that minimizes indel mutations and the developed ranking algorithm as the key components of my algorithm, I observe that I am able to develop an algorithm with a running time of $\mathbf{0}(n^2.l)$ given n sequences with the longest of them having l nucleotides. Due to problems with my ranking algorithm, I am only able to achieve a 44.7% accuracy, something that future work needs to focus on.

1. Introduction

Genome sequencing is an important procedure in biology and bioinformatics. Human genome sequencing has been instrumental in identifying genomic causes of rare disease and understanding variation in complex disease. It has helped drive epigenetic research to new heights and development of personalize medicine. For a long time, genomic sequencing was a very costly procedure but recently advances in sequencing technology have led to lower sequencing costs and the ability to produce large volumes of data, enabling genome Sequencing to be a powerful, broadly used tool for genomics research. Methods for sequencing have been developing ranging from Nanopore Sequencing that has lower throughput to higher throughput methods like Sequencing by synthesis (Illumina)[4] and higher throughput methods like Pacific Biosciences[3] that does Single-molecule real-time sequencing. Pacific Biosciences sequences methods is a particularly interesting method due to its high throughput. It can be used for reads of up to 100,000 bases in length. This is particularly important given how large the human genome is for example. The method can be used in *de novo sequencing*, that is, methods used to determine the sequence of DNA with no previously known sequence. This means that longer reads will usually have sequencing errors. Since this de novo sequencing using PacBio sequencing is applied to reads of greater than 100,000 bases, PacBio sequencing usually has a higher rate of sequencing errors with only an 87% raw-read accuracy in most cases[3]. Since the human genome is very long, there is a need for efficient sequencing machines and algorithms that will combine higher accuracy, low cost and higher throughput. Most of the algorithms and machines provide one one the cost of the other.

2. Current alignment algorithms for PacBio reads

The current sequencing technologies have the ability to cover much longer reads as compared to the previous technologies. The technologies are able to get over the length limitation and facilitate assembling or analyzing of complex genome regions such as GC islands and repeats in downstream. Due to this advancement in sequencing technologies, several sequencing projects have been developed around the long reads idea. Despite the increase in the number of projects on long reads sequencing, in most cases there is still up to 15%[6] sequencing errors dominated by insertions and deletions, so it is important to correct these errors.

To correct these errors, several algorithms have been developed. These error correction algorithms range from *Short read assisted correction* [7] algorithms that align the corresponding short reads from the same species to the long reads and correct them, so they are suitable for the long and short read hybrid projects to *Self-correction*[8] algorithms that align the long reads to themselves and find multiple sequence alignments among the long reads to correct them, suitable for the long read only projects. These error correction algorithms still have several flaws. For example, the error correction algorithm [9] finds seeding k-mers among the long reads and then the alignments by dynamic programming. This comes with some challenges. It is challenging to make correction with fast speed, because it is time consuming to align the long reads of usually several millions to each other. It is also challenging to correct a sufficient amount of read bases, i.e. to achieve high-throughput self-correction, because it is also difficult to align the long reads of about 15% errors to each other[6]. This shows that despite that several algorithms have been developed to reduce the error rate in PacBio sequencing, there is still a challenge to achieve the desired high throughput at higher accuracy. In this paper, I propose a new long reads geenomic assembly algorithm that strives to achieve high accuracy and better running time complexity using the variation of the Needlemann-Wusnch algorithm and a ranking algorithm that I propose a key components.

3. Methodology

Looking at the shortfalls of the current PacBio alignment algorithms, I propose a new fast and high-throughput algorithm for long-read based on the overlap-layout-consensus approach. I use the Needlemann-Wusnch algorithm as wrapper for my algorithm to achieve high precision. Like most PacBio alignment, I follow a de novo approach where the obtain reads are are assembled with no reference genome but purely using their overlapping regions. In the approach, I assume that the PacBio sequencer induces errors in the reads but that the errors can not possibly be greater than 50% [6]. my approach assumes that indel mutations are minimal and gives much emphasis to substitutions

3.1. The algorithm

In the algorithm, I used the Needlemann-Wusnch pairwise sequence alignment algorithm as my underlying algorithm in the first phase of finding overlapping regions in the long sequence reads. I use a variation of the algorithm to obtain optimal sequence overlap region. To do this, I run the Needlemann-Wusnch algorithm on the reads. However, unlike the normal Needlemann-Wusnch[9], I do not allow gaps in the alignment of the reads.

Given reads $R_1, R_2 .... R_n$ obtained from sequencing an unknown genome G_i, For each of the reads, I run the Needlemann-Wusnch algorithm on the read against the remianing n-1 reads. However, unlike the traditional Needlemann-Wusnch, I run the algorithm on the read against the reverse strings of the other n-1 reads. In addition to that, my variation of the algorithm does not allow gaps in aligning the sequences. The n-1 sequences are aligned against the primary sequence in a sliding manner backwards. At every step of the alignment, I keep track of the current maximum score and the e_i, s_j that generated the score.

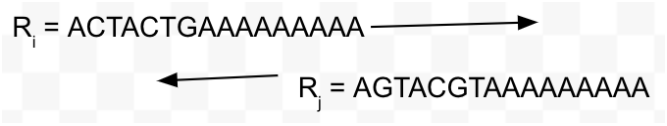

Figure 1 : Alignment of the forwardR_i against reverse R_j read at the beginning of the alignment

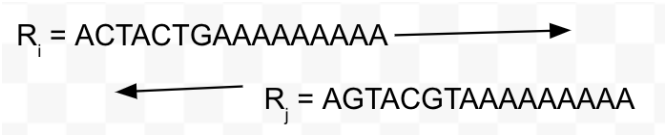

Figure 2 : Alignment of the forwardR_i against reverse R_j read after 3 iterations

After aligning each of the n-1 sequences against the primary sequence, I record the score of the highest scoring alignment of read against each of the other reads and the obtained e_i, s_j for that specific score.

The second phase of the algorithm involves using the obtained e_i, s_j for each of the n reads against the other n-1 reads to assemble the reads and generated the unknown genome G_i. To do this, for each of the reads, I have to look corresponding overlap scores obtained against other reads both in its forward and backwards strands. This way, I get to know which other two reads flank this read. However, these other reads may have entirely different reads flanking them which are not the read under consideration. To solve this, I create a new object that contains the score, the e_i, s_j, the direction of alignment, the read itself and the specific read that this score was against. This means that, I generate nxn instances of this object. I create an array of these objects and sort it by decreasing score. This sorted array is what I use for assembly of the reads and generation of the actual Genome.

Iteratively, I go through the array from index 0. For each index, I obtain the object at that index and get the reads that generated that score. I then assemble them in the orientation specified by the object itself. If there has been already an assembly of the sequences with other reads, then we know that there is an other reads that had a better score on that orientation. In this case, we move to the next element in the array. This approach promises to work because, since we sorted all the nxn overlaps by decreasing score, we are assured that in this particular orientation this is the only overlapping area that has the highest score for these particular sequences.

3.2. Sudo code

Algorithm 1:

Running time:

$0(n^2.l) + 0(n^2) + n^2 log(n^2) = 0(n^2.l)$

O(Space complexity:

$0(n^2)$

Algorithm 2:

Running time:

$0(n^2)$

Space complexity:

$0(n^2)$

Where ***n*** is the number of long sequence reads and ***l*** is the length of the longest read

3.3. Experiment set-up

The algorithm is divided into two phases. As shown above, The first phase uses a variation of the Needlemann-Wusnch pairwise sequence alignment to compute optimally overlapping regions amongst the reads. The second phase of the algorithm uses the computed overlapping regions to assemble the reads in to the required genome. These two phases of the algorithms are different from the current long reads assembly algorithms like short-gun sequencing. To test the effectiveness, efficiency and sensitivity of the algorithms, we run some tests on using two sets of data.

First, we used short reads that we generated as a control to validate the algorithm. A sudo genome of 100 nucleotide was broken down into 5 short reads and the algorithm was run on them.

G_i = ATGGATTTATCTGCTCTTCGCGTTGAA GAAGTACAAAATGTCATTAATGCTATGCAGAAAATCTTA-GAGTGTCCCATCTGTCTGGAGTTGATCAAGGAAC

Algorithm 1 Generate overlapping regions

procedure PAIRWISE ALIGNMENT

$reads \leftarrow R_1, R_2 ... R_n$

initialize matrix D[n][n]

initialize matrix P[n][n]

initialize matrix S[n][n]

initialize array A[nxn]

loop:

for i = 0 to n **do**:

for j = 0 to n **do**:

$maxScore \leftarrow 0$

if i != j **then**:

//Variation of NW Algorithm

if score $>$ maxScore **then**:

$maxScore \leftarrow score$

$e_i \leftarrow i$

$s_j \leftarrow j$

$D[i][j] \leftarrow score$

$Object \leftarrow (maxScore, e_i, s_j, i, j)$

$S[i][j] \leftarrow maxScore$

$P[i][j] \leftarrow (e_i, s_j)$

create object of score, overlap and orientation:

loop:

for i = 0 to n **do**:

for j = 0 to n **do**:

if i $<$ j **then**:

$orien \leftarrow reverse$

else: $orien \leftarrow forward$

$score \leftarrow M[i][j]$

$region \leftarrow D[i][j]$

$Obj \leftarrow (score, region, i, j, orien)$

$A[i] \leftarrow Obj$

A.sort()

return *A*

Algorithm 2 Reads Assembly

procedure READS ASSEMBLY

$arrayA \leftarrow A$

loop:

for element e : A **do**:

if (e_i, s_j not assebled) **then**:

Assemble

else:

continue

Return G_i

reads =

$R_1 = ATGGATTTATCTGCTCT$

$R_2 = TGCTCTTCGCGTTGAAGAAGTACAAAA$

$R_3 = TACAAAATGTCATTAATGCTATGCAGAA$

$R_4 = GCAGAAAATCTTAGAGTGTCCCATCTGTC$

$R_5 = TCCATCTGTCGGAGTTGATCAAGGAAC$

Running the algorithm on the obtained reads, we obtained genome G

G = ATGGATTTATCTGCTCTATGGATT-TAATGGATTTATCTGCCGTCCAAATTCAGCA-GAAAATCTTAGAGTGTCCCATCTGTCTCCATCCATCTGTCGGAGTTG ATCAAGGAACATGGAT

After obtaining G, I run it against the original genome G_i to obtain the accuracy of the predicted genome. This test set was used as a control for my experiment.

Next, we run the algorithm on PacBio sequencing data from a genome sequencing project on Leishmania[10]. Since Leishmania is a species that has already been sequenced, we were able to compare our data with the the actual genome.

4. Results:

I run the new PacaBio long reads alignment algorithm on two sets of reads. On the first set of reads discussed in section 3.3, the initial genome was know and thus it was easy to verify the outcome of the genome assembly. After running the algorithm, I obtained a genomic sequence that

was 44.74% similar to the original genome. I found this by running sequence comparison of the nucletiodes of the actual genome and the assembled genome.

Running the algorithm on the long sequence set of pacbio reads coming from the sequencing of Leishmania Donovani was a bit of a challenge because my algorithm proved to be extremely slow. I was unable to completely run the entire chromosome1 sequence. It took more than hour on each attempt and due to overheating of my machine, I terminated the process .

5. Discussion and future work

A new multiple long reads assembly algorithm for PacBio reads has been proposed to improve on the sensitivity and running time of the long sequence reads genome assembly. The algorithm uses a variation of the Needlemann-Wusnch algorithm to first find the pairwise overlapping regions between reads and then uses these overlaps for genome assembly proves to run in an improved running time of O(n^2).

However, the error rate remains very high and it is higher than the traditional genome assembly algorithms. This low sensitivity is mostly due to the way the algorithm ranks the most probable reads that overlap in the actual genome depending on the overlap score obtained by the algorithm. The algorithm ranks the overlap purely by the similarity scores of suffixes and prefixes of two respective reads. The algorithm does this in a greedy approach as it does not put into consideration subsequent reads that might have a less similarity score but optimal overall due to sequencing errors that result in indel mutations. This is a big shortfall of the algorithm and it is something that needs to be worked on to improve the sensitivity of the algorithm. A memoization dynamic programming algorithm that looks at the similarity scores that would be obtained by the subsequent reads before committing to a particular assembly would go a long way to improve the sensitivity of this algorithm. This is something that future work would look into.

References

[1] AnthonyRhoads, Kin FaiAu, (2015)PacBio Sequencing and Its Applications, Genomics, Proteomics Bioinformatics Volume 13, Issue 5, October 2015, Pages 278-289

[2] Chin CS, Alexander DH, Marks P, Klammer AA, Drake J, Heiner C, Clum A, Copeland A, Huddleston J, Eichler EE, Turner SW, Korlach J (2013). "Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data". Nat. Methods. 10 (6): 563,,69. doi:10.1038/nmeth.2474. PMID 23644548

[3] van Vliet AH (1 January 2010). "Next generation sequencing of microbial transcriptomes: challenges and opportunities". FEMS Microbiology Letters. 302 (1): 1,,7. doi:10.1111/j.1574-6968.2009.01767.x. PMID 19735299

[4] Staden R (11 June 1979). "A strategy of DNA sequencing employing computer programs". Nucleic Acids Research. 6 (7): 2601,,10. doi:10.1093/nar/6.7.2601. PMC 327874. PMID 461197

[5] Ergude Bao, Fei Xie, Changjin Song, Dandan Song, FLAS: fast and high-throughput algorithm for PacBio long-read self-correction, Bioinformatics, Volume 35, Issue 20, 15 October 2019, Pages 3953,,3960, https://doi.org/10.1093/bioinformatics/btz206

[6] Koren S. et al. (2012) Hybrid error correction and de novo assembly of single-molecule sequencing reads. Nat. Biotechnol ., 30, 693,,700.

[7] Chin C.-S. et al. (2013) Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. Nat. Methods , 10, 563,,569.

[8] Chaisson M.J., Tesler G. (2012) Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory. BMC Bioinformatics , 13, 238.

[9] Needleman, Saul B. Wunsch, Christian D. (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins". Journal of Molecular Biology. 48 (3): 443,,53. doi:10.1016/0022-2836(70)90057-4. PMID 5420325.

[10] https://furahadamien.com