ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(45), DOI: 10.17485/ijst/2016/v9i45/92896, December 2016* ISSN (Online) : 0974-5645

Evaluation of CloudRS Algorithm with De Novo Assemblers

Ramraj S, Karthikeyan, Gohula Krishnan and Soumyajyoti Bhattacharya

SRM University, Kancheepuram -603203, Chennai, Tamil Nadu, India;
ramraj.s@ktr.srmuniv.ac.in, karthikeyan_thangavel@srmuniv.edu.in,
gohulakrishnan_b@srmuniv.edu.in, soumyajyoti_somnath@srmuniv.edu.in

Abstract

Objectives: The paper documents a comparative analytical study of the two prominent De Novo Algorithms (DNA) namely Velvet and SSAKE, both are being pipelined by CloudRS. **Methods/Statistical Analysis:** The Research process conducted in this project primarily utilized Next-Generation Sequencing data results. These data sets were further error corrected by pipelining them with CloudRS. Upon error correction, the data sets were assembled separately by VELVET and SSAKE; the data from the analysis were then analyzed as per the mathematical results produced in order to statistically compare the two algorithms for a similar environment. **Findings:** On assembling the error corrected genome, the data produced sets of values. These values were tabulated and noted in order to ensure effective comparison. The values being compared were the N50 and corrected lengths of the assembled genes. The general genome analysis comparison metrics were then utilized to compare the documented data. This showed that a higher N50 value with a better assembled error corrected length read ensured more effectiveness of an algorithm. This result allowed for the first comparison between two prominent DNA algorithms, which hadn't been compared before, to ensure better understanding **Applications/Improvements:** The applications of these results are endless, primarily, to ensure that work which involves assembled genome reads proceed with the utmost effectiveness. Any further improved algorithms, if created down the line, can aid in improving the entire process of the same. Thus, in the uniqueness of the results lies the novelty of the entire project.

Keywords: CloudRS, Comparison, De Novo Assembly (DNA), Evaluation, NGS

1. Introduction

A DNA molecule has four bases, namely, adenine (abbreviated A), cytosine (C), guanine (G) and thymine (T). A complete nucleotide is formed of these four bases being attached to the sugar/phosphate. Now the purpose of De Novo Assembly (DNA) sequencing is to determine the order of the four aforementioned bases in a DNA molecule, thus, also determining the order of a nucleotide in an entire DNA molecule in the process. Technological advancements in the fields of medicine and technology have led to the evolution of DNA sequencing methods[1], which has greatly accelerated biological and medical research and discovery. The speed of sequencing attained has been greatly increased with the available modern DNA sequencing technology[2], which has allowed for sequencing of complete DNA sequences or genomes of myriad species of life, including genomes of most known Prokaryote and Eukaryote species.

De Novo[2] is a short read assembly algorithm which processes individual sequence reads by merging them together to form long contiguous sequences or 'contigs'. These contigs share the same nucleotide sequence as the original template DNA from which the sequence reads were derived. Generally, it is all about assembling reads together so that they form a new, previously unknown sequence. It is different from comparative assembly in the part where the comparative assembly process utilizes existing backbone or reference sequences to compare against, hence, building a sequence which is no identical, although closely resembling the sequence which serves as the backbone. In terms of complexity and time requirements, De Novo assemblers are more memory intensive and generally slower by magnitude orders. The

* *Author for correspondence*

need of the assembly algorithm to compare a read with every other read serves to contribute to this.

The uptake of DNA sequencing technologies in modern life sciences has been made possible by the development of efficient algorithms capable of efficiently processing short read sequences. In particular, reassembly of human genomes (De Novo guided[2]) concluded from the input of aforementioned short reads, has had a positive impact on research in medical sciences. In the absence of a reference genome sequence, the possible alternatives utilize De Novo assembly algorithms. Later on, the data structure designs for spaced seeds are proceed to the form which includes paired K-mers so as to solve the limitations of the De Bruijn Graph (DBG) paradigm, which exist for long reads.

Now, random samples of nucleotide sequences from a target genome of length N are also known as reads. Thus, a read consists of a sequence of characters from the DNA alphabet, which includes: A, C, G, T alongside N which was mentioned previously. Consider r reads with varying lengths, from g min to g max. We denote the length of read p by g*p. The combined total length of the reads is denoted by M = r*p=1*g*p. Substitutions, insertions and deletions are often the three types of errors contained within a read. We denote by e*g as for maximum estimated error rate of a read. Thus, a read of length g *p may contain at most e*g*i errors.

Errors in reads are detected and subsequently corrected by the various error correction algorithms. However, the lack of perceived knowledge about the position of a read in it's target genome means other possible solutions for the error correction process in a necessity. Hence, all error correction methods utilize heuristics in order to determine the reads which align to same genome positions, furthermore, comparing this set of reads so as to correct the reads towards the appropriate and general consensus of the set.

Nevertheless, correcting sequencing errors in the huge amount of reads generated by NGS technologies is time consuming and memory demanding. Furthermore, a huge amount of intermediate data is created in the computation process. For example, a naive implementation of the read-stack algorithm of ALLPATHS-LG[3] would replicate a read for each k-mer subsequence of the read. For a 100G NGS file of read length 36 and k-mer size 25, this means that the total size of intermediate data is 1.2 Tera-bytes, that is, each of the717 reads is replicated 12 times. Thus, new strategies to store and process large quantities of data efficiently are required. The MapReduce[4] framework is a scalable distributed computing framework for biologists and bio-informaticians to process huge amount of genome sequencing data. Though MapReduce and its famous implementation Hadoop[5] are available for researchers and are highly fault tolerance when processing large datasets, the design of MapReduce algorithms[6] is not trivial.

2. Concept Headings

2.1 Overview of CloudRS Algorithm

Current sequencing technologies generate a large number of reads. These reads contain many errors which present a major challenge in using the data in genome sequencing projects as assemblers have difficulties in dealing such with errors. CloudRS[7] is an error correcting algorithm which corrects errors through the ReadStack (RS) algorithm. Unlike previous tools for error correction, CloudRS implements ALLPATHS-LG on Hadoop via Map Reduce. It is advantages lie in the part where it reduces the amount of false positives by being conservative in it is functionalities. Reads produced by different technologies which are used for sequencing like Illumina Genome Analyzer and Roche/454 can be processed by CloudRS without much difficulty. The rates of errors with respect to reads are comparatively more reduced in the process. This overview is given in a flowchart model in Figure 1.

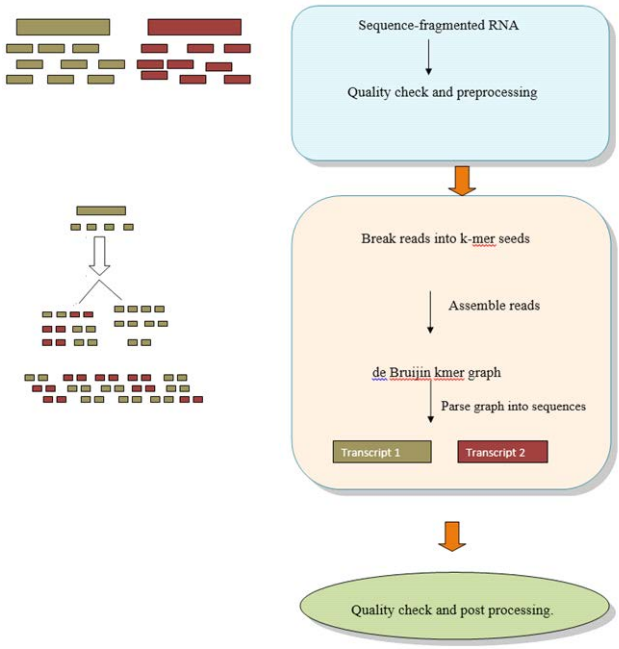

Figure 1. ReadStack overview flowchart.

2.2 Overview of Datasets

The four experimental datasets were downloaded from the archives at NCBI, which are listed in Table 1. All the datasets are sequenced using Illumina sequencers. We used datasets D1-D4 in order to compare between two De Novo assembly algorithms, namely, VELVET[8] and SSAKE[9] when pipe lined with a ReadStack algorithm, namely CloudRS. Table 2 serves as the basis for this comparison, which will be explained further down the line.

2.3 Overview of Velvet

In[10], a new collective set of algorithms called Velvet has been developed in the field of Genomic Sequence Assembly to manipulate De Bruijn graphs, representations of short words i.e. k-mers that holds well for very short reads and high coverage data sets graphically. Implementing Velvet to short reads and paired-ends information only, one can generate contigs of significant length, up to 50-KB N50 length and 3-KB N50 on simulated prokaryotic data and Mammalian BACs respectively. This is a new approach that can produce useful assemblies by leveraging very short reads in combination with read pairs. Velvet can remove errors as well as resolve a large number of repeats provided the presence of read pair information is validated. When there is a repeat longer than the k-mer length, with unpaired r.eads, the assembly is broken.

2.4 Overview of SSAKE

Another short read based De Novo assembly algorithm is SSAKE. It leverages information from short sequences by using it is design oriented processes to categorize novel sequencing targets. This is done by assembling them into short contigs and scaffolds. This was the earliest algorithm published for the same. It is well-suited for structural variant assembly/detection as it assembles whole reads. Applications of SSAKE extend beyond genome assembly and the technology was applied to profiling T-cell meta-genomes, targeted DNA, HLA typing and was key to the discovery of Fusobacterium in colon cancer. This algorithm can be written in PERL and has been utilized on the Linux platform. This algorithm would utilize cyclic processes to have procedural activities through hash table for it is data related activities with respect to short reads. SSAKE is known for being lightweight, robust and easy to run. The Workflow overview of De Novo Assembly is in Figure 2.

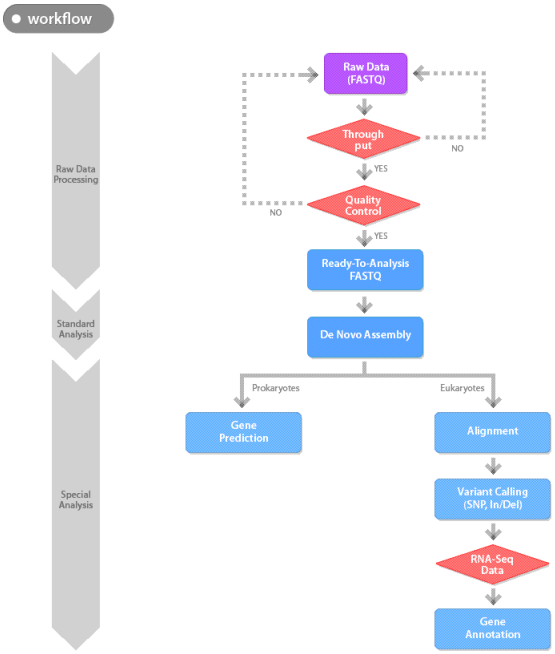

Figure 2. Workflow overview of DNA.

3. Results and Discussion

In[11], a whole set of comparative metrics are established for the characteristic comparison of Assembly algorithms and such other processes associated with Genome assembly processes. Hence, as can be seen in Table 2, we have taken into account few important metric properties with regards to our datasets, the details of which were established in Table 1. These metrics help us compare between SSAKE and Velvet assembly algorithms which produce different results with the same datasets, after they have been acted upon and error corrected by the ReadStack based CloudRS error correcting algorithm.

Table 1. Datasets

Dataset	NCBI Genome Id	Genome	Genome Size	Read length	Genome Coverage %
D1	166	Mycobacterium tuberculosis H37Rv	4.5MB	101bp	65.6
D2	169	Helicobacter pylori 26695	1.7MB	47bp	38.9
D3	176	Streptococcus pneumoniae	2.1MB	27bp	39.6
D4	175	Streptococcus pyogenes	1.9MB	16bp	38.5

Table 2. Compiling assembly pipelines on D1-D4

Dataset	Assembly pipeline	N50	Total length
D1	CloudRS+Velvet	90	70075
	CloudRS+SSAKE	1339	55134
D2	CloudRS+Velvet	80	22074
	CloudRS+SSAKE	1636	20796
D3	CloudRS+Velvet	70	37337
	CloudRS+SSAKE	1158	25481
D4	CloudRS+Velvet	71	20102
	CloudRS+SSAKE	880	23154

In Table 2, we see that SSAKE has a far more positive effect on similar datasets with similar error correction[4] functionalities when compared to Velvet. The size of the N50 property serves as key judging criteria for the same. The other property considered here is the total length of reads, which is generally considered a functionally good metric to work with.

N50 is generally considered as a value of length greater than half of the values present in the dataset[12]. Also, contiguous sequences aim to map sequential reads, hence higher the value greater is it is ability to map itself and find similar structures. We see in Table 2, the N50 values for SSAKE are comparatively higher than those of Velvet when pipelined with CloudRS. This jump in valuation on the positive side, which amounts to an increase in greater than 50% in most of the cases, shows that SSAKE, as in this scenario, is a far more reliable and relatively better De Novo Assembly algorithm when pipelined with CloudRS. Also, as mentioned in[4], a better algorithm is one which has lower length of contigs and greater N50 value. As seen in Table 2, this is the case in our scenario which, hence, proves that SSAKE is the comparatively better algorithm.

4. Conclusion

The very aim of this endeavor was to determine objectively the superiority of two major algorithms when dealing with similar data in fields which coincidentally are also the same. Genome analysis is a wide sphere of influence, especially so today with the growing influence and need to process DNA reads to determine sequences and hence, use them for evaluating effective medical solutions to pressing problems. Firstly, with the advent of adequate research work in this field, CloudRS has been determined over a large set of data as the most effective ReadStack based error detection and correction algorithm. Hence, we utilized the same to receive the primary error corrected data from the given dataset and hence this helped us determine a computational comparison between Velvet and SSAKE far more easily than would have been possible if we took the data straight out of a sequencer like Illumina.

Now, questions may be raised about the choices in assembly algorithms, with reasons being asked regarding why Velvet and SSAKE were chosen. This was done after a lot of deliberation and following all algorithms which have a presence as a implementable DNA algorithm. While the two keywords: De Novo and Assembly relatively shorted the search results down, it must be understood that the final comparison were conducted between SSAKE and Velvet because of their general prominence in the field and the availability of effective implementation procedures for both of them.

On comparison, it was determined that even by narrow margins, for the given set of data, SSAKE was the more efficient and hence, more reliable DNA algorithm. However, it must be realized that this comparison stems from relative results and varies for datasets[13]. Considering the datasets provided to us as a general scenario, which we believe is the situational case here barring exceptions, the conclusion comes to the fact that for error corrected data using ReadStack algorithms, same datasets show SSAKE to be relatively better De Novo assembly algorithm when compared to Velvet, if only at a slight advantage over one another when compared over general genome analysis comparison metrics[14] which are widely accepted.

5. Acknowledgment

We wish to thank anonymous reviewers for their helpful suggestions. We also wish to take this opportunity to express our gratitude for The Department of Software Engineering in SRM University, Kattankulathur for their able support, without which carrying out this project would not have been possible, especially the guidance and support of our Head of Department, Dr. C. Lakshmi, who helped us throughout the process as and when we needed it. This project was conducted independently, without any need for external corporate involvement.

6. References

1. Tipu HN, Shabbir A. Evolution of DNA sequencing, J Coll. Phys. Surg. Pak. 2015; 25(4):210–15.
2. Mardis ER. Next-Generation DNA Sequencing Methods. 2008; 9:387–402.

3. Gnerre S, Mac Callum I, Przybylski D, Ribeiro FJ ,Burton JN , Walker BJ, Sharpe T, Hall G Shea TP, Sykes S, Berlin AMD, Aird M, Costello R, Daza L, Williams R, Nicol A, Gnirke C, Nusbaum ES, Lander DB, Jaffe J. High-Quality Draft Assemblies of Mammalian Genomes from Massively Parallel Sequence Data. 2008; 108(4):1513–18.
4. Dean J, Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters, 2008 Commun. ACM. 2008; 51(1):107–13.
5. Welcome to Apache TM Hadoop TM. Data accessed: 22/06/2016. Available at: http://hadoop.apache.org.
6. Irudayasamy A, Arockiam L. Parallel Bottom-up Generalization Approach for Data Anonymization using Map Reduce for Security of Data in Public Cloud, Indian Journal of Science and Technology. 2015 Sep; 8(22):1–9.
7. Chung WC, Chang YJ, Lee DT, Ho JM. Using Geometric Structures to Improve the Error Correction Algorithm of High-Throughput Sequencing Data on MapReduce Framework, IEEE International Conference on Big Data. 2014, p. 784–89.
8. Kopka H, Daly PW. Velvet: Algorithms for De Novo Short Read Assembly using De Bruijngraphs, Cold Spring Harbor Laboratory Press, 2008.
9. Warren RL, Sutton GG, Jones SJM, Holt RA. Assembling Millions of Short DNA Sequences using SSAKE. 2007; 4(1): 500–01.
10. Kopka H, Daly PW. Correcting Errors in Short Reads by Multiple Alignments, Cold Spring Harbor Laboratory Press. 2008; 27(11):1455–61.
11. Bradnam KR. Assemblathon 2: Evaluating De Novo Methods of Genome Assembly in three Vertebrate Species, GigaScience. 2013; 23(2):10.
12. Chen CC, Chang YJ, Chung WC, Lee DL, Ho JM. CloudRS: An Error Correction Algorithm of High-Throughput Sequencing Data Based on Scalable Framework, IEEE International Conference on Big Data, 2013, p. 717–22.
13. Priyadharshini V, Malathi A, Analysis of Process Mining Model for Software Reliability Dataset using HMM, Indian Journal of Science and Technology. 2016 Jan; 9(4):1–5.
14. N50. Date accessed: 26/2008. Available at: http://www.broad.harvard.edu/crd/wiki/index.php/N50.