Journal of Computational Systems Biology
Volume 6 | Issue 1

Research Article Open Access

Comparative Assessment of *De Novo* Genome Assemblers for Generating Eukaryotic Primary Genome Assembly from Short Reads

Ameya Santhosh, Vikas Vohra[*] and Rani Alex

Division of Animal Genetics and Breeding in the Address, National Dairy Research Institute (ICAR-NDRI), Karnal, India

***Corresponding Author:** Vikas Vohra, Division of Animal Genetics and Breeding, National Dairy Research Institute (ICAR-NDRI), Karnal, India, Tel.: +91 9729000511, E-mail: vohravikas@gmail.com

Citation: Ameya Santhosh, Rani Alex, Vikas Vohra (2023) Comparative Assessment of De Novo Genome Assemblers for Generating Eukaryotic Primary Genome Assembly from Short Reads. J Comp Sys Bio 6(1): 101

Received Date: November 04, 2023 **Accepted Date:** December 04, 2023 **Published Date:** December 08, 2023

Abstract

Even though the bigger eukaryotic genomes can now be accurately sequenced, assembling multiple short sequence reads into a genome assembly *de Novo* is still quite difficult. There is a necessity to find the best suitable assembler for assembling high coverage short Illumina reads. Among the currently available assembly algorithms, De Bruijn and OLC graph-based assemblers are widely accepted and used by researchers. We selected 4 *De Novo* genome assembly tools – Celera assembler, Soap de novo, AbySS and SPAdes which are freely available and suitable for sequence assembling of short reads generated by the Illumina HiSeq sequencing platform. To compare the performance of each assembler, genome assembly was generated from Illumina HiSeq –based short sequence reads of buffalo (*Bubalus bubalis*). We compiled the results from 12 assemblies generated using different K-mer sizes in the four different assemblers. The efficiency of each assembler was evaluated based on maximum memory usage, maximum time, maximum CPU usage etc. The final output file from the assemblers were taken to evaluate the accuracy based on different parameters like N50, number of contigs, total length etc. OLC-graph based Celera assembler was found to be more efficient in producing a primary draft assembly. While coming to accuracy, De Bruijn-graph based SOAPdenovo and OLC based Celera assembler was performing almost equally. Finally, the completeness of assembly generated from each assembler was also evaluated. The results from the present study will aid in the selection of suitable assembling platform for generating best quality genome assembly of large domestic animal genomes.

Keywords: Assembly Algorithms: Illumina Short Reads: Genome Assembly: *De Novo* Genome Assemblers: Eukaryotic Genome.

Introduction

Based on the species, type of data and computational resources available, the bioinformatics pipeline needed for the downstream processing need to be verified and standardised. Exploration of available computing options and their optimisation is necessary for improving the efficiency of the pipeline and for saving time [1]. To reduce computing requirements without affecting the overall result's quality, it is necessary to perform a preliminary assessment of the resources. This is because different laboratory facilities and conditions vary. This analysis need not be performed in all individual phases, only in those that are rate limiting and have an overall effect. This phase, known as the primary assembly in the case of a *de Novo* genome assembly, involves picking the assembly software that will produce the best results for each type of read by comparing its accuracy and computing efficiency.

With the advancement in bioinformatics, numerous genome assemblers with different underlying platforms became easily accessible. Widely used assembly algorithms include De Bruijn graph [2], and OLC graph [3]. The de bruijn graph-based assemblers convert the reads into smaller fragments of length k. k-mers are identified, and a de Bruijn graph with (k−1)-mers as nodes and k-mers as edges is drawn. A Eulerian path is traced through this network resulting in the reconstruction of the original genome sequence. Short reads of even less than hundred base pairs are assembled mainly with the aid of de Bruijn graphs. But it has also been employed with longer reads [4]. De Bruijn graph is the principle behind the assembly softwares like Euler-SR [5], Velvet [6], ABySS [7], and SOAP *De Novo* [8]. One of the advantages of the de Bruijn graph over OLC is that it consumes less computational time and memory. In the earlier periods of introduction of de Bruijn graph, it was mostly used in smaller prokaryotic genomes. Upon continuous revisions and introduction of new DBG based assemblers it is now successfully used for building higher eukaryotic genome assemblies. There are some combined assemblers like MaSuRCA assembler [9], which can use both de Bruijn graph and the overlap-layout consensus methods.There are many assembly pipeline evaluation programs conducted from time to time. The GAGE (Genome Assembly Gold-standard Evaluations) study was designed to evaluate how the latest genome assemblers work on a sample of large-scale next-generation sequencing projects [10]. This evaluation was based on short Illumine reads considering 14th chromosome of Homo sapiens as the representation of large eukaryotic genome. But the efficiency parameters and a complete large eukaryotic genome was not included in the study. Other assembly comparison programs were dnGASP (*De Novo* Genome Assembly Project) [11] and Assemblathon [12]. Assemblathon compared performance of different denovo genome assemblers in large eukaryotic genomes including bird, fish and snake using data from multiple sequencing platforms. The key metrics was emphasizing on assembly parameters only. But it is very important to include efficiency parameters since assembling short reads is a computationally demanding process [13]. The Assemblathon evaluation concluded that Genome assembly software that performs well on one organism performs poorly on another. It is always wise to test several approaches; different software, assembly with or without pre-processing of the sequence data, and different parameter settings.

Developing a eukaryotic genome assembly *de novo*, particularly from short reads, presents several challenges, including fluctuations in sequencing depths, the presence of sequencing errors, and substantial computational demands. Many leading assemblers address the first challenge by implementing an average coverage cutoff threshold to trim areas with lower coverage [14]. Addressing sequencing errors involves employing polishing techniques, utilizing alternative information for correction [15]. However, selecting the assembly process with the least computational demand necessitates a preliminary study, the outcomes of which may vary based on the specific characteristics of the data. As the high throughput short read sequencing technology improved, which is now capable of generating higher coverages, new and modified short read assemblers differing in assembly processes to tackle higher memory requirement were developed. SOAPdenovo, released as SOAP (short oligonucleotide alignment program) is known for the faster short read alignment. It uses Compressed data structure (FM index data structure) [16], which can handle large data volume. ABySS works using Message Passing Interface and reduces the computational demand. High performance computer clusters are used by the Celera Assembler for parallel processing to cut down on processing time [17].

Despite being a prominent member of the Bovidae family, the buffalo genome has received less attention. Although the Indian subcontinent is home to a sizable share of buffalo genetic resources, most of them are currently less explored and non-descript. As an example of one of India's least-explored buffalo genomes, we attempted to create a primary level assembly of the Bhadawari buffalo genome here. In the genome assembly pipeline, the primary genome assembly is a suitable node for diverting to either scaffolding by incorporating additional data (long reads, HiC, optical), gap filling by reference genome, or directly to downstream analysis depending on the completeness.

With the Bhadawari breed of buffalo as an example, the objective of this work was to examine the effectiveness of selected denovo assembly tools suited for assembling large eukaryotic genomes. Since primary assembly is the first step and will serve as a benchmark for the chosen assemblers, we only moved on to that stage. Additionally, as paired Illumina reads are often generated and affordable for most labs, we limited this analysis to using them as the input for each assembler. The analysis of the study's findings can be used to choose the tool that will produce the denovo genome assembly the fastest, the tool that will require the least amount of computational efficiency and can be tailored to different laboratories, and the tool that will produce the best completeness without sacrificing assembly quality.

Methods

Sequencing and Preparation of Data

Multiple blood samples were collected from two healthy female Indian buffaloes of Bhadawari breed from their breeding tract of Etawah district with consent of animal owner under the supervision of a trained veterinarian. Genomic DNA was extracted from blood samples using a standard phenol/ chloroform extraction method as described by Sambrook and Russel, 2001(18). The quality and purity of the extracted DNA was confirmed by measuring A260/A280 ratio and agarose gel electrophoresis. Only the intact DNA possessing 1.8-2.0, 260/280 ratio was proceeded for further analysis. Raw reads (150 bp PE) obtained after Illumina Hiseq 2000 sequencing were quality filtrered using FastQCversion 0.11.9 and adapters were removed with Trimmomatic version 0.39. All test datasets are described in Table 1.

Sample number	Raw reads		Qualified reads	
	Total data in Gb	Number of reads	Total data in Gb	Number of reads
1	174.6	488513210	155.6	427348740
2	198.2	554531666	177.7	495423777

Table 1: Details of the data used for the study

Sl.No.	Assembler	Algorithm	Programming Language
1	AbySS	De Bruijn Graph	C++
2	Soap denovo	De Bruijn Graph	C+
3	Celera Assembler	OLC graph	C+
4	SPAdes	OLC graph	C++

Table 2: Details of the *de Novo* genome assemblers used for the study

Assembly Tools Selected

We selected 4 *de Novo* genome assembly tools – Celera assembler [19], Soap *De Novo* [8], AbySS [7] and and SPAdes [20] (representing two main assembly algorithms) which are freely available and suitable for sequence assembling of short reads generated by the Illumina HiSeq sequencing platform due to their ability to effectively handle the high-throughput, short-read nature data by considering overlaps or constructing k-mers for accurate and efficient genome assembly. The details of the data used and assemblers selected for the study are described in the table 1 and 2. Each assembly tool was run with different k-mer sizes in the two samples of Illumina short reads and primary assemblies were generated. The average results of each assembly metrics were calculated for different *de Novo* assemblers for the ease of comparison. Efficiency and accuracy of each of them were assessed from the output contig/ scaffold file generated.

All assembly processes were performed on an Intel®Xenon® CPU E5-2630 v4 with 40 cores at 2.2 GHz and 6 TB of RAM and 256 GiB memory, running Ubuntu 20.04.4 LTS, 64 bit (ver. 3.36.8).

Efficiency and Accuracy Evaluation

Efficiency of each assembler was evaluated based on maximum memory usage, maximum time, maximum CPU usage. The time taken to generate primary assemblies of individual run was detected using the Linux time command and the median assembling time of different assemblers where calculated. Memory usage and CPU usage percentage for each run was detected by Linux command and comparison of mean usage of each assembler was done. The final output file from the assemblers were taken to evaluate the accuracy based on different parameters like N50, number of contigs, total length etc. Quality was assessed by QUAST tool [21]

Completeness Analysis

Completeness of the assembly was checked with BUSCO [22] using NDDB_SH1 as the reference.

Results

Using Illumina short reads, we generated primary denovo assemblies of the Bhadawari buffalo genome in order to evaluate the performance of four alternative assemblers from two assembly algorithms. Each assembly tool was run with different k-mer sizes in the two samples of Illumina short reads. Various criteria were taken into consideration to evaluate the output files from each assembler that contained contig level assemblies.

Based on maximum memory utilisation, maximum duration, and maximum CPU usage, each assembler's efficiency was assessed. A significant difference was found while comparing the assembly run times required by each assembly programme utilising the same number of threads and the same data. While ABySS required the longest runtime (6120 minutes), Celera Assembler generated a primary assembly in the shortest amount of time (2400 minutes). While SPAdes (5760 minutes) took almost twice as long as Celera Assembler, SOAPdenovo came in second consuming 4380 minutes. When analysing the percentages of maximum memory and CPU utilisation, it was discovered that SPAdes was using the most, followed by ABySS. But contrasting with the other two, Celera and SOAPdenovo required the least Computational demand for the same data (figure 1).

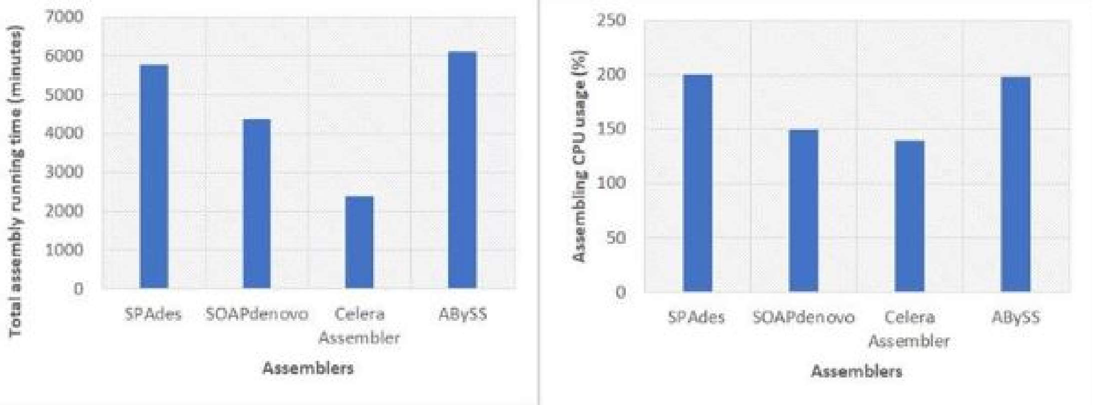

Figure 1: Comparison of efficiency parameters of each assembler A)average assembling time B) Memory C) CPU usage

The final output file from the assemblers was used to assess the accuracy based on various factors, such as N50, the quantity of contigs, and total length. QUAST tool was used to evaluate quality (Figure 2). Considering, the N50 metrics Celera assembler generated primary assembly with largest N50 size while assembly from ABySS was having the smallest N50. Celera assembler produced an average N50 of 5.72 Kb, followed by SPAdes and SOAPdenovo with nearly identical lengths (5.28 Kb and 5.22 Kb). In comparison to the other three, the ABySS assembler produced a N50 length that was extremely short (1.81 Kp). The largest contig length was from SOAPdenovo (128 Kb) while Celera assembler (7.4 Kb), SPAdes (6.3 Kb) and ABySS (4.2 Kb) produced smaller N50 values. The least number of contigs were in assembly created by the Celera assembler (8,86,299) and then SOAPdenovo (58,11,150). ABySS, and SPAdes were clearly falling behind to reduce the number of contigs in the primary assembly.

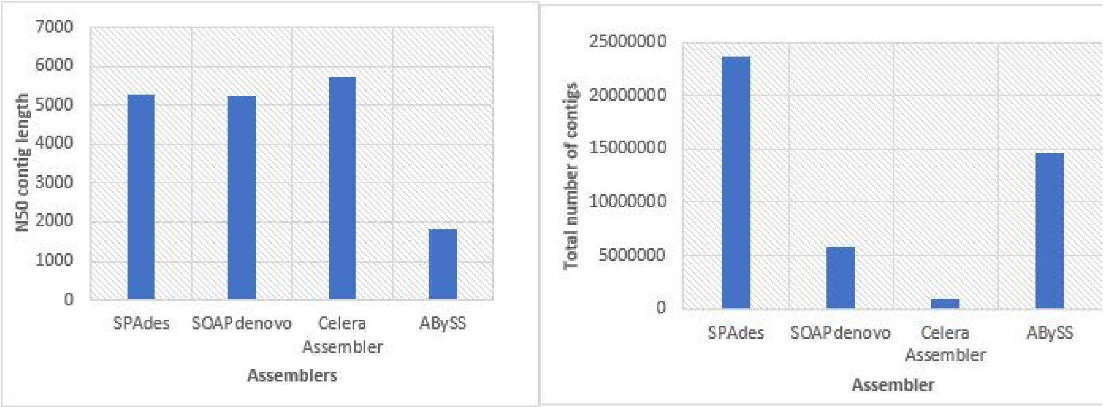

Figure 2: Comparison of accuracy parameters of each assembler A) N50 contig length B) Total number of contigs

The completeness evaluation of the best assemblies revealed that Celera assembler, SOAP denovo and SPAdes generated primary assembly with completeness in the range of 35-39 BUSCO score. But AbySS assembler produced least complete assembly with only 16.8% Completeness. Fragmentation percent ranged from 8.5-9.9, AbySS having least percentage of fragmented BUSCOs.

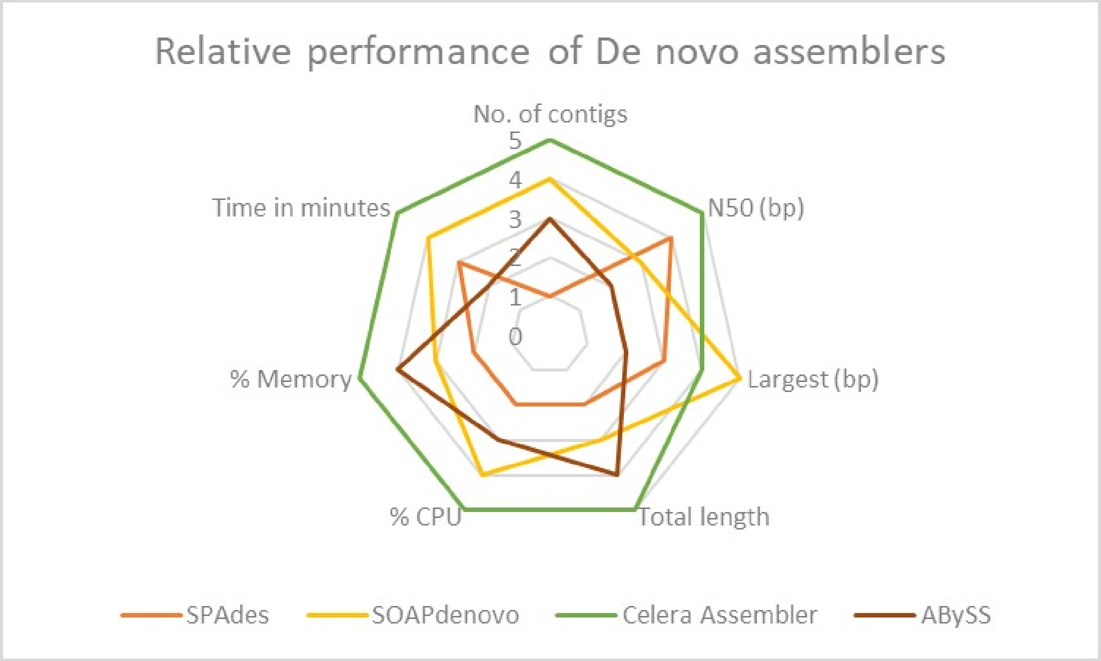

Figure 3: Relative performance of *de Novo* assemblers

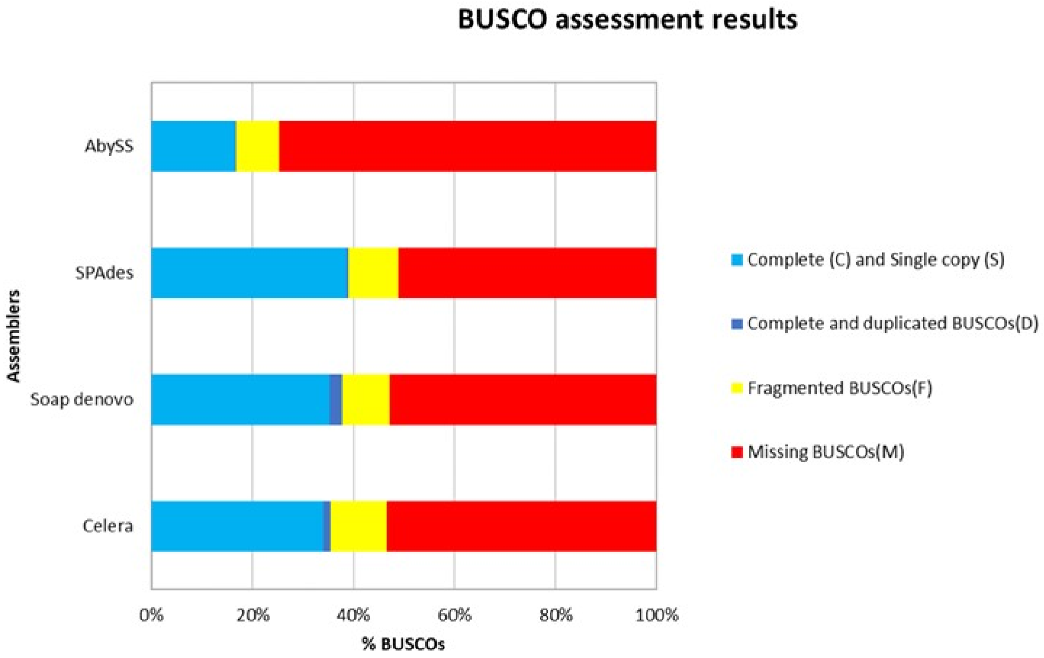

Figure 4: Busco assessment results

Discussion

Results are frequently skewed when choosing bioinformatics tools for research employing specific kinds of data since assessments are made by focusing primarily on one or two parameters. While handling huge data for a computationally demanding process like genome assembly of eukaryotes, this selection must consider efficiency, accuracy, and final genome assembly completeness in order to get an unbiased result.

The genome assemblies from Celera assembler showed the highest N50 values as compared to SOAPdenovo, SPAdes and ABySS for the given data of Bhadawari buffalo. In another comparison experiment, with eight times better N50 value, Celera assembler excelled Velvet, ABySs, and SSake employing E. coli and Yeast dataset [23]. This might be as a result of the built-in capability of the Celera Assembler, which uses an OLC-based method for building consensus sequences based on overlaps and multiple rounds of assembly and error correction to gradually reduce errors and increase the accuracy of the assembled genome sequence. The SOAPdenovo outperformed the remaining three assemblers in producing the assembly with largest contig from short Illumina data. This result was similar to that of GAGE comparison where SOAPdenovo generated assembly with larger contigs than the other assemblers from short Illumina data of S.aureus and R. sphaeroides [10]. Comparing each assembler's computational efficiency in terms of maximum memory usage, maximum time, and maximum CPU utilisation, the Celera assembler required the least amount of processing power to complete genome assembly in the shortest amount of time.

Evaluation of completeness was done by using BUSCO tool which analyses the completeness by analysing the expected gene content based on evolutionary relationships [22]. Overall complete BUSCOs is affected by other major factors such as sequencing quality, taxonomic group etc. Since the study was conducted on the same data, this bias was overcomed. On comparing the primary assembly generated by the four selected assemblers, Since N50 is a widely accepted statistics for assembly evaluation, the larger N50 was expected to have a positive association with more complete assembly [24]. Eventhough this was true for the assembly with least accuracy, surprisingly the assembly with more accuracy did not give the higher completeness percentage. Assembly produced by AbySS assembler gave the least complete assembly but the difference in percentage of completeness was only 3.4 for the other assemblers. Soapdenovo and Spades generated primary draft assemblies with least fragmentation and missing BUSCOs. High fragmented assemblies are supposed to be incomplete. High repetitive contents will make the assembly process difficult, thus fragmenting the assemblies. When the fragmentation percentage is high this can also lead to a higher missing BUSCOs and vice versa. Duplicated BUSCOs are comparatively less in diploid genomes. The range of duplicated BSCOs reported in published buffalo genome assemblies was 58 (Bubalus bubalis EGYBUF_1.0)-199 Mediterranean (BubalusbubalisUMD_CASPUR_WB_2) [25]. In this study maximum duplication of 305 was from Soap denovo and minimum of 45 was from SPADES.

The selection of different parameters for a genome assembly will vary upon different organisms [26]. Moreover, the credit and penalty to be given for each metrics will solely depend on the fate of the final assembly and requirements of each research. So, the outcomes of this study can be analysed in the light of the forementioned parameters of different genome assembly projects. For the general information we are giving equal merit to all the metrices included in the study. Considering this, apart from the measure like biggest contig, Celera's assembler stood best among the four assembled programmes in terms of efficiency and accuracy. Therefore, the Celera assembler needs to be enhanced in order to produce the longest contigs. Soapdenovo came in second in all other categories used to measure efficiency and accuracy and excelled in generating longer contigs than Celera assembler.

The findings of our work recommend that Celera Assembler and SOAPdenovo could serve as the best choice for creating primary *de Novo* assembly using short reads of large eukaryotic organisms.

These findings not only significantly contribute to the theoretical framework of *de Novo* genome assembly but also offer practical insights that can influence the current methodologies and approaches employed in genomic studies. Understanding the strengths and areas for improvement in each assembler becomes instrumental in refining and adapting *de Novo* assembly techniques to the unique genomic landscapes of different organisms. The practical application of these findings lies in the enhancement of assembly strategies for large eukaryotic organisms, where obtaining accurate and contiguous genome assemblies is often challenging. Researchers can leverage the guidance provided by this study to make informed decisions when selecting assembly tools, tailoring their approach based on the specific requirements and characteristics of the organism under investigation. The insights into the strengths of Celera Assembler, particularly its efficiency and accuracy, imply that refining this tool could lead to substantial improvements in generating longer contigs, a crucial factor in achieving high-quality genome assemblies. Moreover, the recognition of SOAPdenovo as a commendable choice, especially for its proficiency in producing longer contigs, highlights the versatility of available assembly tools. Researchers can strategically choose between Celera Assembler and SOAPdenovo based on the specific goals of their studies, allowing for a more nuanced and targeted approach to *de Novo* genome assembly.

In practical genomics, where the application of *de Novo* assembly techniques is paramount for understanding species diversity, unraveling functional genomics, and exploring evolutionary dynamics, these recommendations serve as a compass. Researchers can confidently navigate the intricate landscape of genome assembly, making informed choices that align with the goals of their studies and the unique characteristics of the organisms they are investigating. This, in turn, facilitates the generation of more accurate, comprehensive, and biologically relevant genomic data, advancing our understanding of complex biological systems and contributing to breakthroughs in fields such as medicine, agriculture, and ecology. In essence, the practical implications of these findings extend beyond the laboratory, shaping the trajectory of genomic research and its transformative potential in diverse applied contexts.

Conclusion

Large eukaryotic genomes have not yet been used to compare denovo genome assembly performance. Our study is the first to provide a comparative analysis of the performance of denovo assemblers in producing the basic assembly. The findings of this study indicate a significant performance difference between the chosen tools in terms of processing requirements, assembly statistics, and completeness. These findings can be used as a guide to picking the most appropriate assembly tool while considering elements like type of data available, follow-up steps in the pipeline, and downstream processing.

Data Availability

Data used for the current study can be made available from the corresponding author on request through proper channel.

Conflict of Interest

Concerning the manuscript entitled "Comparative assessment of *de Novo* genome assemblers for generating eukaryotic primary genome assembly from short reads" the authors declare that there is no competing conflict of interest

Ethics Statement

This samples were collected from livestock farmers of Uttar Pradesh with the consent of the animal owner and under the supervision of a trained veterinarian.

Funding Statement

No special funding was received for the current study.

Author Contributions

Vikas Vohra: Conceptualization, Data curation, Project administration, Funding acquisition, Supervision, Review and Editing. Ameya Santhosh: Methodology, Formal Analysis, Software, Visualization, Investigation, Original Draft preparation. Rani Alex: Methodology, Formal analysis, Review and Editing.

Acknowledgments

We thank the Director, ICAR-NDRI, Karnal and the Director, CIRB, Hisar, for providing facilities and support for conducting the research work. We also thank Dr. Ravi Kumar Gandham V.P.P.S., Scientist G, NIAB, Hyderabad and Dr Amod Kumar, Scientist, Dept. of Animal Genetics and Breeding, NBAGR, Karnal, Haryana for their support and logistics in conducting the research. We express our gratitude to the livestock farmers for donating the sample and for protecting the valuable germplasm of Bhadawari buffaloes.

Supplementary Material

Not available

References

1. Brandies PA, Hogg CJ (2021) Ten simple rules for getting started with command-line bioinformatics. PLoS Computational Biology, 17: e1008645.

2. Idury RM, Waterman MS (1995) A new algorithm for DNA sequence assembly. J Comput Biol. 2: 291-306.

3. Myers EW (1995) Toward simplifying and accurately formulating fragment assembly. J Comput Biol, 2: 275-90.

4. Ekim B, Berger B, Chikhi R (2021) Minimizer-space de Bruijn graphs: Whole-genome assembly of long reads in minutes on a personal computer. Cell Syst, 12: 958-68.

5. Chaisson MJ, Pevzner PA (2008) Short read fragment assembly of bacterial genomes. Genome Research, 18: 324-30.

6. Zerbino DR, Birney E (2008) Velvet: algorithms for *De Novo* short read assembly using de Bruijn graphs. Genome Res, 18: 821-9.

7. Simpson JT, Wong K, Jackman, SD, Schein JE, Jones SJ et al. (2009) ABySS: a parallel assembler for short read sequence data. Gen. Res, 19: 1117-23.

8. Li R, Zhu H, Ruan J, Qian W, Fang X (2010) *De Novo* assembly of human genomes with massively parallel short read sequencing. Genome Research, 20: 265-72.

9. Zimin AV, Delcher AL, Florea L, Kelley DR, Schatz MC, et al.(2009) A whole-genome assembly of the domestic cow, Bos taurus. Genom. Biol, 10: 1-0.

10. Salzberg SL, Phillippy AM, Zimin A, Puiu D, Magoc T (2012) GAGE: A critical evaluation of genome assemblies and assembly algorithms. Genome research, 22: 557-67.

11. Taylor JF, Whitacre LK, Hoff JL, Tizioto PC, Kim J, et al. (2016) Lessons for livestock genomics from genome and transcriptome sequencing in cattle and other mammals. Genetics Selection Evolution, 48: 1-18.

12. Bradnam KR, Fass JN, Alexandrov A, Baranay P, Bechner M (2013) Assemblathon 2: evaluating *De Novo* methods of genome assembly in three vertebrate species. GigaScience, 2: 2047-17X.

13. Desai A, Marwah VS, Yadav A, Jha V, Dhaygude K et al. (2013) Identification of optimum sequencing depth especially for *De Novo* genome assembly of small genomes using next generation sequencing data. PloS one, 8: e60204.

14. Liao X, Li M, Zou Y, Wu FX and Wang J (2019) Current challenges and solutions of *De Novo* assembly. Quantitative Biology, 7: 90-109.

15. Collins A (2018) The challenge of genome sequence assembly. The Open Bioinformatics Journal, 11.

16. Flicek P and Birney E (2009) Sense from sequence reads: methods for alignment and assembly. Nature methods, 6: S6-12.

17. Miller JR, Koren S and Sutton G (2010) Assembly algorithms for next-generation sequencing data. Genomics, 95: 315-27.

18. Sambrook J and Russell DW (2001) Molecular Cloning-Sambrook & Russel- Cold Springs Harbor Lab Press: Long Island, NY, USA 3.

19. Myers EW, Sutton GG, Delcher AL, Dew IM, Fasulo DP (2000) A whole-genome assembly of Drosophila. Science, 287: 2196-204.

20. Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. Journal of computational biology, 19: 455-77.

21. Gurevich A, Saveliev V, Vyahhi N and Tesler G (2013) QUAST: quality assessment tool for genome assemblies. Bioinformatics, 29: 1072-75.

22. Seppey M, Manni M and Zdobnov EM (2019) BUSCO: assessing genome assembly and annotation completeness. Gene prediction: methods and protocols, 227-45.

23. Cherukuri Y and Janga SC (2016) Benchmarking of *De Novo* assembly algorithms for Nanopore data reveals optimal performance of OLC approaches. BMC genomics, 17: 95-105.

24. Jauhal AA and Newcomb RD (2021) Assessing genome assembly quality prior to downstream analysis: N50 versus BUSCO. Molecular Ecology Resources, 21: 1416-21.

25. Porrelli S, Gerbault-Seureau M, Rozzi R, Chikhi R, Curaudeau M (2022) Draft genome of the lowland anoa (Bubalus depressicornis) and comparison with buffalo genome assemblies (Bovidae, Bubalina). G3, 12: 234.

26. Baker M (2012) *de Novo* genome assembly: what every biologist should know. Nature methods, 9: 333-37.