MANTECH Publications

Journal of Research in Microbiology and Immunology
Volume 8, Issue 1, January-April 2026
ISSN: 3108-0952 (Online)

Bacterial Whole Genome Assembly Using SPAdes - Review

Dr. S. Sreeremya
External Faculty of Pharmacology
Department of Pharmacology
Crescent College of Nursing, Palakkad, Kerala, India
Email ID: *sreeremyasasi@gmail.com*

ABSTRACT

From the time period of sangers sequencing, there are several gene assessment techniques used. The most prevalent in recent times is SPAdes. Specific algorithms are used in construction of genome assembler. It is apen source biotool for sequencing.

KEYWORDS: *Sangers sequencing, genome assembler, specific algorithms, gene assessment, biotool*

INTRODUCTION

When Idury and Waterman introduced the Bruijn graphs for the fragment assembly, many viewed this approach as impractical due to the high error rates in Sanger reads. Pevzner et al. removed this bottleneck by introducing the *error correction* procedure that made the vast majority of reads error-free. Similarly, PDBGs may mainly appear impractical due to variation in biread (and thus *k*-bimer) distances characteristic of the NGS. SPAdes addresses this bottleneck by introducing *k-bimer adjustment*, which precisely reveals exact distances for the vast majority of the adjusted *k*-bimers, and by introducing the *paired assembly graphs* inspired by PDBGs. In particular, SPAdes is able to utilize the read-pairs; E+V-SC used the reads but ignored the pairing in read-pairs to avoid the misassembles caused by an elevated level of chimeric read-pairs (Kamath et al.,2017).

De Bruijn graphs, the PDBGs, and several other graphs in this paper are special cases of *A*-Bruijn graphs. SPAdes is a *universal A*-Bruijn assembler in the sense that it avails *k*-mers only for building the initial de Bruijn graph and “forgets” about them afterwards; on the subsequent stages, it only performs the graph-theoretical operations on graphs that need not

be labeled by *k*-mers. The operations are based on graph topology, the coverage, and sequence lengths, but not the sequences themselves. At the final stage, the consensus DNA sequence is restored. Researchers designed a universal assembler to implement several variations of *A*-Bruijn graphs (e.g., paired and the multisized de Bruijn graphs) in the same framework, and to apply it to other applications where these graphs have proven to be quiet useful (Haghshenas et al.,2020) . New age genetics in the field of science has made the inventions and assessments much easier (Dr. S. Sreeremya ,2025). Genomics is also one wide stream of data which give clarity in the field of research (Dr.Sreeremya.S,2025).

ASSEMBLY IN SPADES

Below is the outline the four stages of SPAdes, which deal with issues that are particularly troublesome in SCS: the sequencing errors; non-uniform coverage; insert size variation; and chimeric reads and bireads (Sommer et al.,2013):

1. Stage 1 (the assembly graph construction) is addressed by every NGS assembler and is often referred to as de Bruijn graph *simplification* (e.g., the *bulge/bubble* removal in EULER/Velvet). One propose a new approach to assembly graph construction that uses the *multisized de Bruijn graph*, implements new bulge/tip removal algorithms, detects and also removes chimeric reads, aggregates biread information into *distance histograms*, and allows one to backtrack the performed graph operations(Gurevich A et al.,2013).
2. Stage 2 (*k*-bimer adjustment) derives the accurate distance estimates between *k*-mers in the genome (edges in the assembly graph) availing joint analysis of distance histograms and paths in the assembly graph.
3. Stage 3 constructs the *paired assembly graph*, inspired by the precise PDBG approach.
4. Stage 4 (contig construction) was well studied in the context of the Sanger sequencing . Since the NGS projects typically feature high coverage, NGS assemblers generate rather accurate contigs (although the accuracy deteriorates for SCS). The SPAdes constructs DNA sequences of contigs and the mapping of reads to contigs by the backtracking graph simplifications.

Previous studies and assessments demonstrated that coupling various assemblers with error correction tools improves their performance. However, most error correction tools (e.g., the Quake) perform poorly on the single-cell data since they implicitly assume nearly uniform read coverage

Our paired assembly graph approach differs from the existing approaches to assembly and dictates new algorithmic solutions for various stages of the SPAdes. Thus, we will describe several variations of de Bruijn graphs, paving to construction of the paired assembly graph (covering stages 2 and 3), before describing Stage 1(Barthelson et al., 2011).

UNDERSTANDING ABOUT BRUIJN GRAPHS

Since various assembly articles and research study use widely different terminology, below one specify a terminology that is well suited for the PDBGs. All graphs considered below are directed graphs. A vertex *w precedes* (*follows*) vertex *v* in a graph *G* if there exists an edge from *w* to *v* (from around *v* to *w*) in *G*. INDEGREE(*v*) (OUTDEGREE(*v*)) is the number of vertices preceding (following) *v*. A vertex *v* in a graph *G* is called a *hub* if the INDEGREE(*v*) $\neq 1$ or OUTDEGREE(*v*) $\neq 1$. A directed path in the *G* is called a *hub-path* (abbreviated *h-path*) if its starting and finishing vertices are hubs and its intermediate vertices are not hubs. Obviously, each precise edge in the graph belongs to a unique h-path. An edge is called a *hub-edge* (abbreviated *h-edge*) if it initiates at a hub. There is a correspondence between h-paths and h-edges: the first edge on the each h-path is an h-edge, and the h-edge is unique to that h-path. Given an h-edge α, one define the h-path starting at α as PATH(α) and denote the number of the edges in this h-path (*h-path length*) as |PATH(α)|. If *a* is the *i*-th edge in an h-path ($1 \leq i \leq$ |PATH(α)|) starting from an h-edge α, one define H-EDGE(a) = α and OFFSET(a) = I (Fig-1)(Magoc et al.,2013) .

Standard de Bruijn graphs

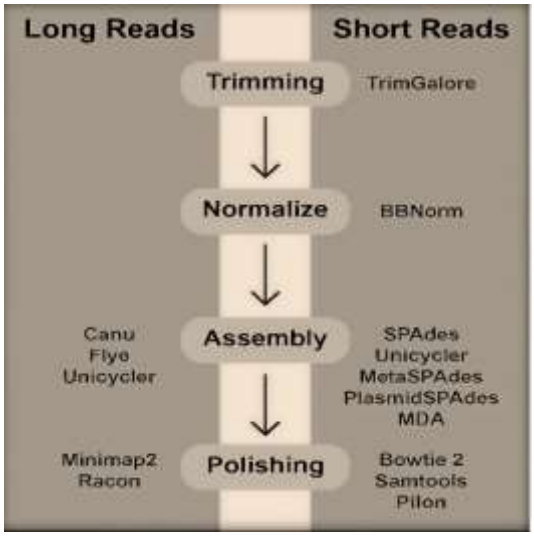

Figure: 1

An *n-mer* is a string of the length *n*. Given an *n*-mer , one define PREFIX and SUFFIX.

For the rest of the assessment, one fix a positive integer *k*. For a set READS of strings (thought of as the DNA sequencing reads over the alphabet {*A, C, G, T –nitrogen bases*}), let *N* be the number of *k*-mers that occur in strings in the READS as substrings. We define the *de Bruijn graph* DB(READS, *k*)(Salzberg et al.,2012)

- **D1.** Define an initial graph G_0 on $2N$ vertices. For each *k*-mer *a* that mainly occurs in strings in READS as a substring, introduce the two new vertices *u, v* and form an edge $u \rightarrow v$. Label the new edge by *a*, *u* by PREFIX(*a*), and *v* by SUFFIX(*a*). Note that we label edges by *k*-mers and vertices by $(k-1)$-mers(Bradnam et al.,2013).
- **D2.** Glue vertices of G_0 together if they have the same label.

TESTING SPADES INSTALLATION

SPADES comes with the self-test option to make sure that the program is correctly installed. While this is not true of most programs, it is always a good idea to mainly run whatever test data a program makes available rather than jumping straight into once own data as knowing there is an error in the program rather than your data makes the troubleshooting very different(Schatz et al.,2012).

CONCLUSION

SPADES as a biotool, especially as a genome assembler has greatly contributed in the field of research. This tool uses several graphs and many methods of approach.They uses PDBG based graphs and mapping the contigs,to orient the sequence and perform scaffoldings.

REFERENCES

1. Schatz MC, Witkowski J, McCombie WR, et al. Current challenges in de novo plant genome sequencing and assembly. Genome Biol. 2012;13(4): 243.
2. Bradnam KR, et al. Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. GigaScience. 2013;2(1):10.
3. Salzberg SL, Phillippy AM, Zimin A, Puiu D, Magoc T, Koren S, Treangen TJ, Schatz MC, Delcher AL, Roberts M, et al. Gage: A critical evaluation of genome assemblies and assembly algorithms. Genome Res. 2012;22(3):557–67.
4. Magoc T, Pabinger S, Canzar S, Liu X, Su Q, Puiu D, Tallon LJ, Salzberg SL. Gage-

b: an evaluation of genome assemblers for bacterial organisms. Bioinformatics. 2013;29(14):1718–25.

5. Barthelson R, McFarlin AJ, Rounsley SD, Young S. Plantagora: modeling whole genome sequencing and assembly of plant genomes. PLoS ONE. 2011;6(12):28436.
6. Gurevich A, Saveliev V, Vyahhi N, Tesler G. Quast: quality assessment tool for genome assemblies. Bioinformatics. 2013;29(8):1072–5.
7. Sommer DD, Delcher AL, Salzberg SL, Pop M. Minimus: a fast, lightweight genome assembler. BMC Bioinforma. 2007;8(1):64. 24. Chikhi R, Medvedev P. Informed and automated k-mer size selection for genome assembly. Bioinformatics. 2013;30(1):31–7.
8. Haghshenas, E., Asghari, H., Stoye, J., Chauve, C., and Hach, F. (2020). Haslr: Fast hybrid assembly of long reads. iScience,
9. Kamath, G. M., Shomorony, I., Xia, F., Courtade, T. A., and Tse, D. N. (2017). HINGE: long-read assembly achieves optimal repeat resolution. Genome research, 27(5), 747–756.
10. Dr. Sreeremya. S, Book title- Bionanotechnology and Genomics, 2025, pp-200, paperback. ISBN-978-93-49942-33-2.
11. Dr. S. Sreeremya, Journal of Biochemistry and Molecular Science, New Age Genetics and Genetic Engineering Techniques, , Vol 7(1),pp-9-18.2025.