**Comparison of prokaryotic genome assembly using Velvet and RAPT for learning undergraduate-level bioinformatics**

Huy Bui & William Mowell

**Introduction:**

Biology fields have come a long way since the invention of chain-termination sequencing by Sanger in 1977.1 As biologists refined their way to tap into genetic codes, longer strands of genetic information could be deciphered. However, a string of words could as well be useless if we could not find a way to interpret it. With a size of a bacterial genome ranging anywhere from hundreds of thousands to millions bases and even a higher number for eukaryotes, computers were necessary to store and process data of such size efficiently.2 As a result, biology, especially with fields working with genetics or evolution, has become intertwined with informatics or the practice of processing information with computational power. Bioinformatics was born from this necessity.

Now, in 2023, as computer science has been the talk of the world with faster-than-ever processing power and artificial intelligence bringing immense value to the development of biological discoveries, data scientists could find themselves extremely valuable in biological sciences. Competencies in bioinformatic skills have been agreed to be a vital part in the study of modern biology as indicated by biologists surveyed in the United States, Canada and Israel.3 However, aspiring biologists in undergraduate level are not seeking out to learn bioinformatics as reported in 2019 with one of the main reasons cited being a lack of student preparation in this field, or an unfamiliarity with operating computers as a way to process information.3 In this study, the authors as a group of a biochemist and a data analyst, who did not have intensive training in bioinformatics or informatics in general, looked into genome assembly, a practice widely used in bioinformatics, under a pedagogical perspective to assess and compare the accessibility and the learning values of the two pipelines Velvet and Read Assembly and Annotation Pipeline Tool (RAPT).

Genome assembly is a fundamental process in genomics that involves piecing together short fragments of DNA sequences, or reads, to reconstruct the complete genome sequence of an organism. The goal of genome assembly is to generate contiguous sequences called contigs, which represent long stretches of DNA. The ideal contigs count would be one as the entire genome would have no fragmentation.The quality of a genome assembly is typically measured by various metrics, such as the total length of the assembled genome, the number of contigs, or the N50 score. N50 represents the length of the shortest contig that covers 50% of the total genome assembly length. A higher N50 score indicates a more contiguous and complete genome assembly, since longer and fewer contigs are preferred over shorter and more numerous ones.

Velvet is a popular algorithm for de novo genome assembly widely used for assembling short-read sequencing data, especially for bacterial and small eukaryotic genomes. The algorithm is based on the de Bruijn graph approach, which involves breaking the input reads into short k-mers with the velveth module. Then, velvetg module constructs a graph from the k-mers, and then traverses the graph to find contigs that represent the underlying genome sequence. Velvet is known for its fast runtime and high accuracy in assembling short-read data, making it a valuable tool for many genomic applications. However, some criticisms for Vetvet are not user-friendly as it requires a line command interface to operate.

RAPT is a bioinformatics pipeline designed to assemble and annotate bacterial genomes. The pipeline is capable of handling data from both single-end and paired-end Illumina runs It includes several steps, including read quality control, read trimming, assembly, and annotation. It is often the benchmark pipeline for biologists as it comes with its own annotation tool and a web interface, in which one can simply enter an accession number acquired from databases without any fine-tuning of parameters and receive assembly results and gene annotations in hours.

To compare each of the pipeline’s contributions to the learning of bioinformatics, both pipelines were used to assemble three prokaryotic species. The rationale behind the choice was that prokaryotic species come in a wide variety with different functions and genes while being considered simpler species than eukaryotes. More specifically, a prokaryote would generally have a shorter genome and have only one circular strand of DNA compared to the double-stranded DNA found in eukaryotes. A simpler prokaryotic genome would generally be used when the concept is taught to a student who is new to bioinformatics. The genomes of two strains of *Bacteroides caccae (B. caccae)* and an outgroup of *Escherichia coli* (*E. coli*) will be assembled using both pipelines.

*B. caccae* is an anaerobic bacteria in the human’s gut that specializes in fiber degradation and is responsible for intestinal bowel disease. The strains compared in this study would be ATCC 43185 - the first strain of *B. caccae* ever discovered, and MR1\_13. *E. coli* is a familiar bacteria for students studying in microbiology and biochemistry. It is widely used in laboratories around the world. *E. coli* played a vital role in crucial genetics studies. It was expected that the two strains of *B. caccae* would yield a similar assembly, that is, the same number of genes, contigs and N50 values while *E. coli* would be an outgroup and is expected to produce significantly different results from the strains of *B. caccae*.

**Methods:**

This study was largely conducted using various tools provided through the National Center for Biotechnology Information (NCBI). In order to generate the necessary output sequence data, the web interface of NCBI’s RAPT was utilized. Running this requires either an accession code from their Sequence Read Archive (SRA) database or an unassembled FASTQ file. The SRA database was used to acquire the accession codes for the species and strains of interest. The accession numbers for the strains used were SRR15171175 for *B. caccae* ATCC 43185, SRR950446 for *B. caccae* MR1\_13 and SRR24142388 for *E. coli*. These accession numbers were then entered into the RAPT pipeline to generate the output. The results from RAPT have 10 files in which skesa\_out.fa is the fasta file containing all of the assembled contigs, assembly\_stat\_report.xlsx is a spreadsheet showing the total contigs, their lengths and the N50 value, and annot.faa is the gene annotation in the gff3 format. To parse through the gff3 format, a Python script was made (Figure 1) to convert the .faa file into a .xlsx spreadsheet.

For the Velvet pipeline, a Linux command-line interface was used. Velvet and sratoolkit were required. sratoolkit was used to acquire the .sra files to acquire the two files from the paired-end reads of all three strains studied. The .sra files were hashed into fragments with lengths of 25 amino acids (or 25-mers) with velveth. The output of velveth was a folder. The entire folder was inputted through the velvetg model to yield a resulting folder. The statistics of the assembly could be found in the Log file. R was used to visualize the distribution of coverage of the 25-mers, or how many times each of the 25-mers was covered by the read sequences. Fine tuning of the parameters in velvetg could be determined from this distribution using the commands -cov\_cutoff, which was used to ignore k-mers that had the coverage value under the value specified by the parameter, and -exp\_cov, which was used for velvetg to know the expected value for coverage. It is ideal to get a higher coverage as it means that the confidence of the results is higher. Tablet was also used with the data received from velvetg for visualization of the contigs.

**Results:**

The results from the RAPT pipeline showed that the *B. caccae* ATCC 43185 strain contained 94 contiguous sequences, and had a total length of 4510604 base pairs. Its shortest sequence has 457 base pairs, its longest has 260156 base pairs, and it has a N50 of 118618. The MR1\_13 strain of *B. caccae* has 164 sequences and a total length of 5301467 base pairs. Its shortest sequence has 201 base pairs, the longest has 741912 base pairs, and it has an N50 of 202003. This finding contradicted the expectation that the two assemblies would be very similar. However, two different strains and two different data sampling could lead to significantly different results. The *E. coli* sample has a total of 328 sequences, and an overall length of 4520708 base pairs. Its shortest sequence was found to have 443 base pairs, its longest sequence has 136795 base pairs, and it has an N50 of 24468. After running the RAPT data through Python script to examine the gff3 data under an xlsx format, it was found that there were 4161 genes in *E. coli*, which was close to the reported number of 4401. There was a discrepancy between the gene number between the two strains of *B. caccae* with 3609 genes in the ATCC 43185 strain while there were 4312 genes from the MR1\_13 strain. This was, however, consistent with the findings that the ATCC 43185 strain’s genome was shorter than that of the MR1\_13 strain. A closer inspection of the genes annotated by RAPT also confirmed that the two *B. caccae* strains shared similar genes whereas a lot of the shared genes were not found in *E. coli*, which was consistent with the expected results.

The initial run of the velvetg command yielded very low, therefore likely insufficient, N50 outputs. In order to find a level of contigs that could reasonably be removed from the data set, and to locate the expected coverage, a histogram was generated using R code. The histograms after the initial run without the use of the -cov\_cutoff and -exp\_cov parameters are shown in Figure 2 below. The MR1\_13 and *E. coli* strains were found to have very similar distributions, though individual values were different. A cutoff around 40 and an expected coverage around 70 were identified for the MR1\_13 strain, and a cutoff around 30 and an expected coverage around 60 were identified for the *E. coli* sample. The ATCC 43185 strain on the other hand was entirely right skewed, which made it impossible to eliminate poor contigs using the applied methods. Results from this strain are reported here; however, the results for this strain through Velvet are very likely unusable outside of this study.

After removing some configs through the process above, a more optimized run of velvet could be conducted for usable results. Using this output, the ATCC 43185 strain has a N50 of 25, the MR1 13 strain has a N50 of 21045, and the *E. coli* has a N50 of 9355. While more reasonable than our initial results, excluding the ATCC 43185 strain, each of the above results are lower than those found through RAPT processing. This was also expected as the RAPT online version did not require any parameter fine tuning, meaning that it is sensible to assume that the pipeline had to be optimized already to yield results that would give a lower amount of contigs and a higher N50 value. The newer output was used to generate a visual that would show individual sequences and the number of contigs through a Tablet program. Through this, it was found that the ATCC 43185 strain has 181511 contigs, the MR1 13 strain has 1525 contigs, and the *E. coli* has 1482 contigs. These are all higher than those shown in the RAPT output, which may be caused by greater fragmentation.

**Conclusion:**

The results from the 2 pipelines were generally suggesting that the *B. caccae* was different from the *E. coli*, which was expected. However, only RAPT with a gene annotation pipeline could show similarities between the *B. caccae* strains. This was to highlight the importance of a gene annotation tool to be used in tandem with a genome assembly. As covered in the introduction, data requires interpretation for them to be useful. RAPT being integrated with the Prokaryotic Genome Annotation Pipeline (PGAP) allowed the study to quickly identify and interpret the similarities of genes from two-seemingly different assemblies from the *B. caccae*. Overall, the quality of the assemblies from RAPT seemed to be better than that generated by Velvet as they had a lower contigs count, suggesting less fragmentation during the assembly process and a higher N50 value.

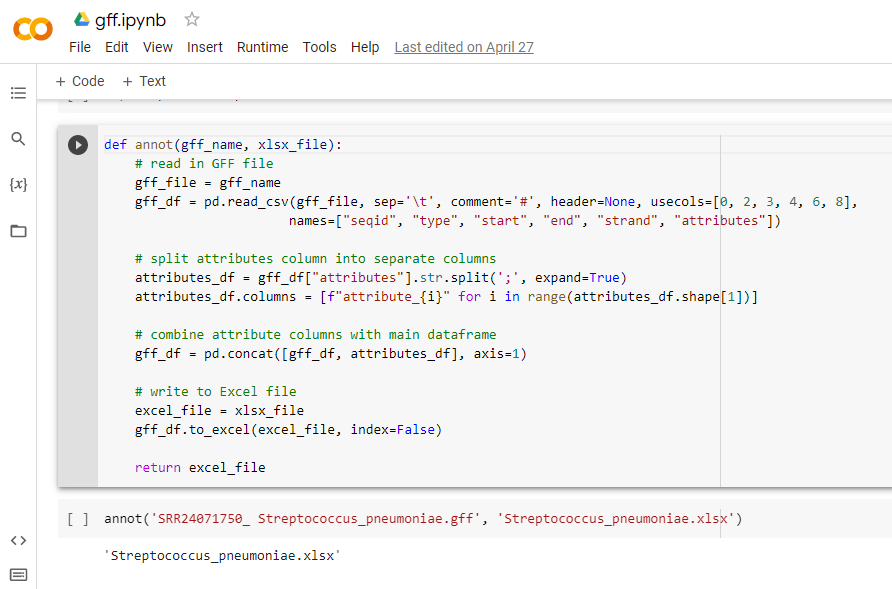
To assess and compare the two pipelines with regards to an undergraduate learning experience. Velvet is a more complex and technical tool that requires a deeper understanding of bioinformatics concepts to use effectively while providing more advanced features, such as the ability to work with paired-end reads and to adjust k-mer size, which can result in higher quality assemblies. However, due to its complexity, it may not be the best choice for new learners and may cause frustration along the way. It is also not the best choice for self-taught learners as they might not have access to a command-line interface or a powerful processing unit to run larger genome assemblies. However, it is excellent as a tool to learn important concepts regarding genome assemblies such as contigs, coverage and N50. As students are required to engage extensively with different statistics to manually tune the parameters for better results, Velvet is a great tool for demonstrating concepts while allowing for practice simultaneously. RAPT, on the other hand, is a more accessible and user-friendly genome assembly tool that is designed for researchers with limited bioinformatics experience. It is a pipeline that incorporates several popular bioinformatics tools and automates many of the steps involved in genome assembly and annotation. RAPT includes quality control measures and incorporates multiple assembly methods, providing users with a comprehensive view of their data. It is a far more user-friendly experience, thus making it the prime choice of platform for researchers to conduct their study on; however, it was not effective as a learning tool, especially for the topic of genome assembly. Overall, while Velvet may be a better tool for learning bioinformatics concepts, RAPT is more accessible and user-friendly, making it a better choice for researchers with limited bioinformatics experience.

Future work on this would be to put this study on a larger scale in an undergraduate classroom or different bioinformatics curricula to assess the learning effectiveness of these pipelines towards undergraduate students. A standard assessment guide should be made to determine the effectiveness of these methods such as what would an effective learner be in this topic. For example, one would be able to identify concepts of contigs or N50 and explain them to others or be able to operate the program to assemble other genomes. Future improvements may include pairing Velvet with MAKER - a widely-utilized gene annotation tool - so that there are more possible comparison to be made with the RAPT pipeline.

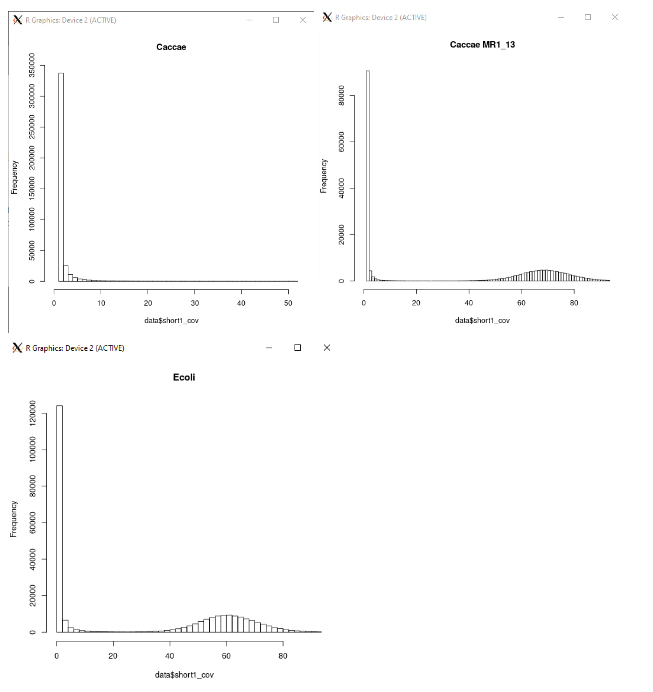
**References**

1. Sanger, F.; Nicklen, S.; Coulson, A. R. DNA Sequencing with Chain-Terminating Inhibitors. *Proceedings of the National Academy of Sciences* 1977, 74 (12), 5463–5467.
2. Genome size check. https://www.ncbi.nlm.nih.gov/assembly/help/genome-size-check/#:~:text=In%20such%20cases%2C%20the%20accepted,unlimited%20(100%20Gbases%20in%20practice) (accessed May 2, 2023).
3. Wilson Sayres, M. A.; Hauser, C.; Sierk, M.; Robic, S.; Rosenwald, A. G.; Smith, T. M.; Triplett, E. W.; Williams, J. J.; Dinsdale, E.; Morgan, W. R.; Burnette, J. M.; Donovan, S. S.; Drew, J. C.; Elgin, S. C.; Fowlks, E. R.; Galindo-Gonzalez, S.; Goodman, A. L.; Grandgenett, N. F.; Goller, C. C.; Jungck, J. R.; Newman, J. D.; Pearson, W.; Ryder, E. F.; Tosado-Acevedo, R.; Tapprich, W.; Tobin, T. C.; Toro-Martínez, A.; Welch, L. R.; Wright, R.; Barone, L.; Ebenbach, D.; McWilliams, M.; Olney, K. C.; Pauley, M. A. Bioinformatics Core Competencies for Undergraduate Life Sciences Education. *PLOS ONE* 2018, 13 (6).
4. Williams, J. J.; Drew, J. C.; Galindo-Gonzalez, S.; Robic, S.; Dinsdale, E.; Morgan, W. R.; Triplett, E. W.; Burnette, J. M.; Donovan, S. S.; Fowlks, E. R.; Goodman, A. L.; Grandgenett, N. F.; Goller, C. C.; Hauser, C.; Jungck, J. R.; Newman, J. D.; Pearson, W. R.; Ryder, E. F.; Sierk, M.; Smith, T. M.; Tosado-Acevedo, R.; Tapprich, W.; Tobin, T. C.; Toro-Martínez, A.; Welch, L. R.; Wilson, M. A.; Ebenbach, D.; McWilliams, M.; Rosenwald, A. G.; Pauley, M. A. Barriers to Integration of Bioinformatics into Undergraduate Life Sciences Education: A National Study of US Life Sciences Faculty Uncover Significant Barriers to Integrating Bioinformatics into Undergraduate Instruction. *PLOS ONE 2019*, 14 (11).
5. Serres, M. H.; Gopal, S.; Nahum, L. A.; Liang, P.; Gaasterland, T.; Riley, M. A Functional Update of the Escherichia Coli K-12 Genome. *Genome Biology* 2001, 2 (9).

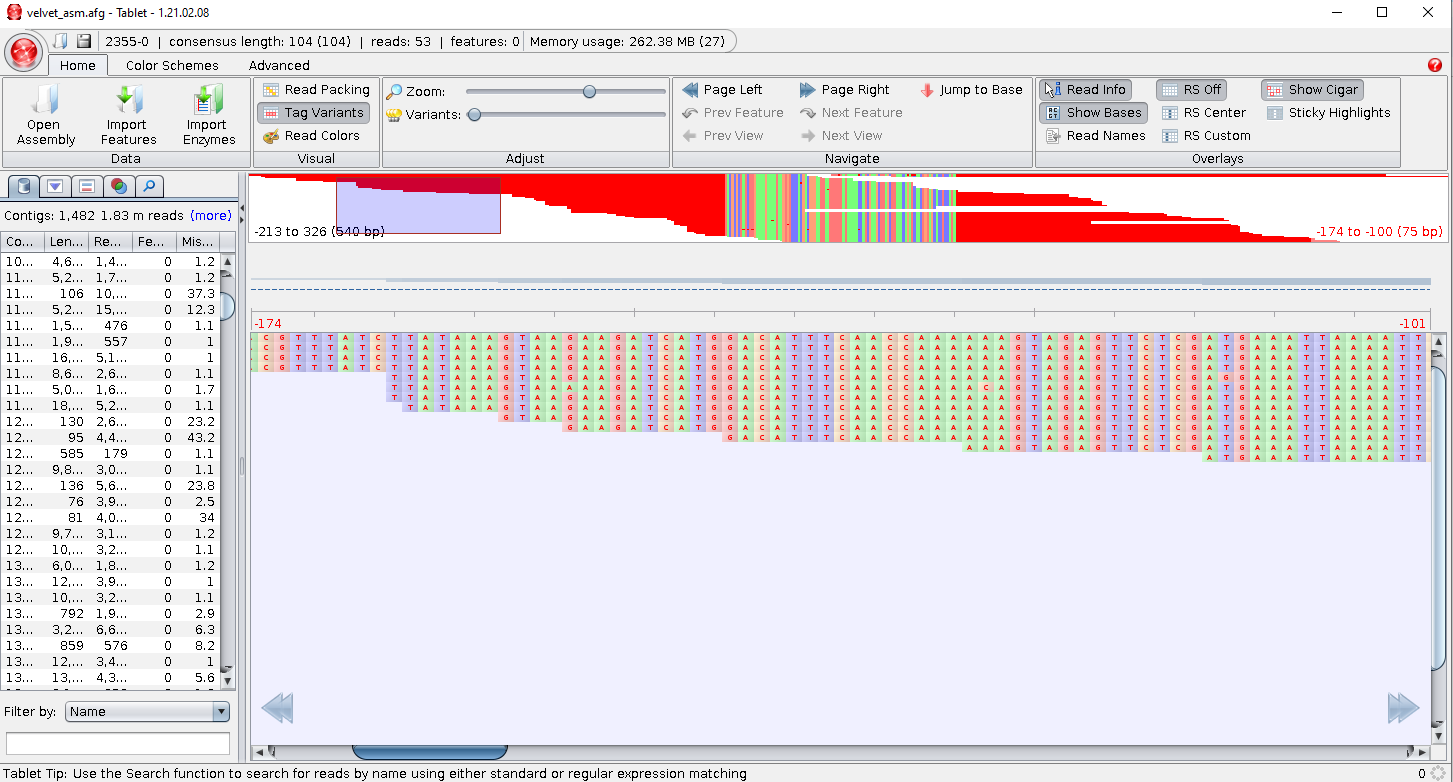
**Supplemented Figures:**



*Figure 1:* Python script for gff3 parser.



*Figure 2:* Histograms of 25-mers coverage after the initial Velvet run without the -cov\_cutoff and -exp\_cov parameters.



*Figure 3:* Visualization of contigs with Tablet.