Virseqimprover: An Integrated Pipeline for Viral Contig Error Correction, Extension, and Annotation

Haoqiu Song
Virginia Tech
Saima Tithi
St. Jude Children's Research Hospital
Frank Aylward
Virginia Tech
Roderick Jensen
Virginia Tech
Liqing Zhang (lqzhang@cs.vt.edu)
Virginia Tech

Research Article

Keywords: metagenomics, viral genome assembly, viral metagenomics

Posted Date: September 8th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3318217/v1

Additional Declarations: No competing interests reported.

Abstract

Background: Despite the recent surge of viral metagenomic studies, it remains a significant challenge to recover complete virus genomes from metagenomic data. The majority of viral contigs generated from *de novo* assembly programs are highly fragmented, presenting significant challenges to downstream analysis and inference.

Methods: We have developed Virseqimprover, a computational pipeline that can extend assembled contigs to complete or nearly complete genomes while maintaining extension quality. Virseqimprover first examines whether there is any chimeric sequence based on read coverage, breaks the sequence into segments if there is, then extends the longest segment with uniform coverage, and repeats these procedures until the sequence cannot be extended. Finally, Virseqimprover annotates the gene content of the resulting sequence.

Conclusion: Virseqimprover has good performance on correcting and extending viral contigs to their full lengths, hence can be a useful tool to improve the completeness and minimize the assembly errors of viral contigs. Both a web server and a conda package for Virseqimprover are provided to the research community free of charge.

Introduction

Metagenomic sequencing is a great approach to study hundreds of microbial organisms at the same time without cultivating them in the lab environment. Metagenomic sequencing data often contain hundreds of millions of short reads and one major computational task is short reads assembly. To date, many tools such as MetaVelvet [1], metaSPAdes [2], Ray Meta [3], IDBA-UD [4], and MEGAHIT [5] have been developed to assemble short reads from metagenomic samples. The goal of metagenome assembly is to generate complete genomes for the majority of organisms that comprise the sample [6]. However, because of the complex nature of metagenomic data, for example, the presence of hundreds of organisms, highly uneven coverages of different organisms, presence of multiple strains of the same species, assemblers have difficulty in recovering complete genomes and often produce partial fragments of the original genomes [7, 8, 9]. In addition, due to the presence of closely related species and presence of multiple strains of the same species, assemblers sometimes produce chimeric sequences (sequences where genomes from multiple organisms are incorrectly assembled together [9]). For example, to generate viral genomes from metagenomic data, current common approaches either conduct de novo assembly to generate viral contigs or assemble based on virus reference genomes, and then identify and annotate virus-specific genes. However, the number and diversity of viral sequences in reference databases are dwarfed by the sequences from their cellular hosts [10], and the ecological richness, evenness, and genomic complexity of viral assemblages greatly complicate the determination of full-length virus genome sequences from metagenomic samples generated from naturally occurring viral populations [11, 12]. With the coexistence of highly similar strains of a virus species, assemblers can easily produce chimeric sequences, making it challenging to recover the virus genomes.

Tools have been developed to correct assembly errors such as predicting the positions of chimeric sequences based on supervised learning or deep learning models [13, 14], and to improve the quality of draft assemblies by correcting single nucleotide polymorphisms, insertions, and deletions [15, 16]. There are also tools to extend assembled contigs and fill the gaps of draft genomes [17, 18, 19, 20]. However, to our knowledge, there is no existing pipeline that can perform all of these steps together. Here we developed an integrated computational pipeline, Virseqimprover, which combines error correction, extension, and annotation of draft assemblies of viral genomes all together in a single tool.

Virseqimprover takes a contig and the metagenomic reads from which the contig was generated as input. Error correction and extension steps are applied iteratively to grow the contig as much as possible while ensuring that the extended assembly is error free. As non-uniform coverage is an indication that the assembly is likely chimeric, in the error correction step, Virseqimprover checks for the uniformity of the coverage of the assembly and keeps only the uniform coverage part of the assembly for the next extension step. In the extension step, Virseqimprover maps reads to the edge regions of the contig (i.e., left and right boundary regions), and conducts local assembly using the contig and the mapped reads. If the contig gets extended, it will be sent to the error correction step, otherwise, Virseqimprover will trim the ends of the contig and attempt to extend it further. If it is not possible, Virseqimprover will stop the error correction and extension process. When the iterative error correction and extension steps are done, the final extended viral sequence will be annotated. Virseqimprover outputs the extended viral contig along with the annotation.

Results

2.1 Viral contig summary

To demonstrate Virseqimprover's utility in correcting and extending viral contigs, we took several metagenomic samples (under NCBI accession number SRX2912986 [21], SRX7079549), and ran assembly programs including FVE-novel [22], metaSPAdes [2], and MEGAHIT [5] to generate contigs. We then applied Virseqimprover to the seven longest contigs generated by the tools for contig extension and correction. Table 1 shows the length information of the seven contigs before and after applying Virseqimprover, indicating that most of the contigs got extended. The next several sections provide detailed comparison of the contigs generated by the popular assembly tools with the contigs refined by Virseqimprover.

2.2 Contigs S0, S1, S2

For contigs S0, S1, and S2, the longest contig was contig S0 with length 193,112 bp. Figure 1 shows that for S0, coverage from around 24,500 bp to 25,500 bp is lower than the average coverage and varies a lot after 150 kb. Virseqimprover checked for the uniformity of the coverage and extracted the longest region with uniform coverage, which was a region with length 127,423 bp from 25,567 bp to 152,989 bp. This

longest non-suspicious region went through the iterative extension and error correction steps. When the iterative extension step was done, the output contig (length 163,662 bp) was checked for circularity. When the circular region was trimmed and Pilon was applied to the output contig, an improved contig (length 152,707 bp, denoted as S0') was generated and became the final output of Virseqimprover as shown in Figure 1. Some missing part on the left was due to the trimming of the circular region. Besides, the drops in coverage at the ends of the sequences are due to the fact that the number of reads (especially paired-end reads) mapping at the edge regions is less than in the middle regions.

Table 1 Contig Length Before and After Applying Virseqimprover

Contig ID	Original length (bp)	Length after Applying Virseqimprover (bp)
S0	193,112	152,707
S1	155,659	152,362
S2	80,620	151,828
S3	132,604	160,744
S4	136,254	151,190
S5	4,179	14,374
S6	13,396	22,526
S7	23,114	32,035

Similarly, after checking the depth of coverage of contig S1, we found that there was a non-uniform coverage region or suspicious region from 133,376 bp to 134,543 bp. Virseqimprover then extracted the longest non-suspicious region which was a region of length 133,375 bp from 1 bp to 133,375 bp and then applied the iterative extension and error correction steps to generate an extended contig. When all these steps were finished, the circularity of this contig was checked and Pilon was applied to the contig. The final output was a contig with length 152,362 bp. Figure 2 shows the depth of coverage of both original (S1) and final sequences (S1').

For contig S2 with length 80,620 bp, the per depth of coverage was checked and according to Virseqimprover it had no non-uniform coverage region or suspicious region. Hence this contig directly went through the extension and error correction steps iteratively. Finally, after using Pilon to improve the assembly, a contig with length 151,828 bp was generated. Figure 3 shows that after applying Virseqimprover, we extended the original contig on both ends and nearly doubled the total length of the original contig to get a greatly extended contig (S2') which has a uniform depth of coverage along the length.

The improved versions of contigs, S0', S1', and S2', were compared with the 153 kb strain of a novel uncultured virus. After predicting the genes and visualizing the gene cluster comparison, as shown in Figure 4, we can see that S1' and S2' are very similar to the 153 kb reference strain, whereas S0' is a bit different from all other contigs. This shows that Virseqimprover has correctly recovered the whole virus

sequences. Additionally, since some regions in S0’ do not match with any of the regions of all the other contigs, it could be an indication that S0’ is a different strain of the same virus species.

2.3 Contig S3

In the same manner, the contig S3, with length 132,604 bp was checked for the uniformity of the coverage and found that it had a region with low coverage at around 49 kb to 66 kb as shown in Figure 5. BLAST search shows that this region aligns best with Cyanophage P-RSM3 and Prochlorococcus phage P-SSM4 whereas the other parts of the contig aligns best with Cyanophage P-RSM1 and Synechococcus phage metaG-MbCM1, indicating that this low coverage region is probably a misassembly. Virseqimprover flagged all the suspicious regions and extracted the longest region with uniform coverage from 66,390 bp to 132,603 bp. Then this 66 kb region was extended through iterative extension and error correction and a contig with length 160,821 bp was generated. After checking for the circularity of this contig and applying Pilon to this contig, an improved contig (denoted as S3’) with length 160,744 bp was produced which was the final output of Virseqimprover. Figure 5 shows that the non-uniform regions in the original contig are filtered out so that the final corrected and extended contig has a uniform depth of coverage along its length. Figure 6 shows the protein annotation of the final sequence. Using BLAST search, we identified four phages that show the highest sequence similarity to S3’. Based on the gene cluster comparisons, as shown in Figure 7, we can see that the improved contig S3’ does have some similar proteins with these phages. However, DNA sequence alignment of S3’ with these genomes also reveals some dissimilar regions, with great sequence identity variation along the entire sequence, ranging from 75.38% to 82.72%. Hence the improved S3’ might be from a novel phage species.

Figure 8 shows the similarity of S3’ (length 160,744 bp) to the viral sequence (177,631 bp) recovered by the semi-automated assembly process in Geneious. The protein identity threshold is 30%, which means that two proteins are considered to belong to the same group if their protein identity value is above 30%. Apart from the beginning part in the 177 kb strain that does not have many alignments in S3’, a small region in the middle of the sequence also shows difference between these two sequences (colored as the gray arrows). We thus further compared S3’ with the 177 kb strain to find out the difference in this specific region. Based on the pairwise DNA sequence alignment by EMBOSS Stretcher [23], the comparison of these two contigs reveals that in the 177 kb strain, a 1,282 bp region from 56,831 bp to 58,112 bp does not have many matches with S3’. In this part instead of this 1,282 bp region, S3’ contains a 1,342 bp region from 22,163 bp to 23,504 bp. Analysis of the depth of coverage of these two sequences in the area where they are different reveals that those areas have a relatively lower depth of coverage (about 150x) compared to the average depth of coverage (about 300x), as shown in Figures 9 and 10. Moreover, based on the BLASTP search, the specific protein sequence corresponding to this area in contig S3’ aligns best with Synechococcus phage metaG-MbCM1, whereas the 177 kb strain aligns better with Synechococcus phage S-SM2. The differences between S3’ and the 177 kb strain suggest that they may represent different strains of the same phage.

2.4 Contig S4

Contig S4 has 136,254 bps. After Virseqimprover's contig extension, error correction, and circularity check, S4 was extended to 151,190 bps. Virseqimprover indicates the final extended contig S4' is not circular. Figure 11 shows that the depth of coverage of the original S4 is rather uniform, and was extended for both sides of the sequence, with both left and right ends of S4' showing higher coverage than nearby regions, which indicates the presence of repeats. Closer examination of the end sequences reveals that the regions indeed are repeats.

2.5 Contigs S5, S6

Contig S5 is generated by MEGAHIT and contig S6 is generated by metaSPAdes. These two contigs both have a 99% identity to the marine virus with ID AFVG 25M466, covering 12% and 40% of the viral genome, respectively. After applying Virseqimprover, S5 got extended to 14,374 bp, and S6 extended to 22,526 bp, as shown in Figure 12 and Figure 13. The extended S5' covers 43% of the marine virus genome, and the extended S6' covers 68%. After doing the pairwise alignment between the extended sequences and the marine virus genome, we find that the extended parts are identical to the marine virus genome, and therefore Virseqimprover can accurately extend the sequence in this sample.

2.6 Contig S7

Contig S7 is generated by metaSPAdes. It has a 99% identity to the marine virus with ID AFVG 25M409. After applying Virseqimprover, S7 was extended from 23,114 bp to 32,035 bp, which covers 98% of the AFVG 25M409 viral genome (32,812 bp). Figures 14 and 15 show that Virseqimprover successfully extends the original fragmented contig to nearly complete genome with high accuracy.

2.7 Evaluation of Virseqimprover

We used CheckV [25] as an independent source to further evaluate and completeness and quality of the contigs refined by Virseqimprover. CheckV determines the completeness and quality of assembled contigs by comparing them to a large database of complete virus genomes and has been used widely to evaluate the quality of assembly. Figure 16 shows the completeness of both the original and refined sequences. Based on CheckV, only one of the eight contigs generated by either FVE-novel, metaSPAdes or MEGAHIT is complete, in contrast, four contigs recovered or refined by Virseqimprover are complete, thus achieving an overall of 30% improvement over these assembly tools. Remarkably, contigs S2 improved from 48.19% to 100% in completeness, S4 from 60.72% to 100%, and S7 from 68.89% to 100%. Contig S5 also shows a significant improvement of completeness, from 13.18% to 44.92%. This shows that Virseqimprover is effective in extending contigs and improving the completeness of contigs generated by other assemblers. Table 2 shows the sequence quality assessed by CheckV. CheckV has four categories

of quality evaluation, low, medium, high, and complete. Except contig S5, contigs refined/recovered by Virseqimprover all have improved quality over the original contigs generated by other assemblers. Only two of the original eight contigs fall into the low or medium quality category, whereas six of the eight contigs from Virseqimprover fall into either the high or complete category, suggesting that Virseqimprover is useful for improving the quality of contigs generated by other assemblers. Taking together, Virseqimprover significantly enhances the quality of the contigs, enabling the extension of fragmented sequences into more complete sequences or full genomes.

Table 2 Sequence Quality Before and After Applying Virseqimprover

Contig ID	Original sequence quality	Improved sequence quality
S0	High-quality	High-quality
S1	High-quality	Complete
S2	Low-quality	Complete
S3	Medium-quality	High-quality
S4	Medium-quality	Complete
S5	Low-quality	Low-quality
S6	Low-quality	Medium-quality
S7	Medium-quality	Complete

Materials And Methods

3.1 Data sets

3.1.1 Contigs from FVE-novel.

The input of Virseqimprover includes a contig and the metagenomic data from which the contig is generated. To demonstrate the utility of Virseqimprover, we took the GOV database containing 24,411 contigs as the reference ”genomes” and applied FVE-novel [22] to an ocean metagenomic sample (NCBI accession number SRX2912986, [21]) to generate viral contigs. The sample contains 18,471,506 paired-end reads with an average read length 151 bp. FVE-novel is a pipeline that first maps all the reads to the reference sequences using FastViromeExplorer [26], performs *de novo* assembly of the mapped reads to generate contigs, and extends the contigs via iterative assembly to produce final viral sequences. Altogether FVE-novel produced 268 contigs. We applied Virseqimprover to the five longest contigs (hereafter labeled as S0, S1, S2, S3, and S4) to see whether the contigs can be either further extended and/or corrected for any error. Among the five contigs, S0, S1, and S2 are highly similar to each other whereas S3 and S4 are not.

To validate the results, we reassembled the contigs using the “Map to Reference” algorithm implemented in Geneious 11.0.4 [27] together with multiple rounds of manual inspection and processing. Through this semi-automated process, we hope to examine whether there are multiple viral strains or species and if

there are, whether a complete assembly of the dominant strain can be generated. Specifically, the metagenome reads were aligned to contigs S0, S3 and S4 using the “Low sensitivity/Fastest” setting allowing for 10% mismatches. Then the consensus sequence from the alignment was segmented into contigs with the highest coverage > 40x. These contigs were binned into lists of contigs with similar coverage for further assembly. Next, the contigs in each bin were iteratively grown using Geneious by mapping reads to the ends with high stringency. To be more specific, all of the phage metagenome paired-end reads were aligned to these high coverage contigs using “Map to Reference” with stringent “Custom Sensitivity” settings allowing no more than 1% “Mismatches per Read” and 1% “Gaps per Read” and requiring that both of the paired-end reads map to the new consensus sequence. This process was iteratively continued until the extended contigs merged together, maintained approximately uniform coverage, and could no longer be extended or closed into a circular genome sequence. Using this laborious and semi-automated approach, we recovered a 153 kb contig from contig S0, a 177 kb contig from contig S3, and a 151 kb contig from contig S4, respectively.

3.1.2 Contigs from metaSPAdes and MEGAHIT.

To further evaluate Virseqimprover’s ability in correcting and extending contigs, we also ran metaSPAdes [2] and MEGAHIT [5] on a metagenomic sample (SRX7079549) to generate contigs and then applied Virseqimprover to see whether it can extend the contigs and correct any assembly errors. These two programs (metaSPAdes and MEGAHIT) have been shown to have less misassemblies compared to some other metagenome assemblers (e.g., IDBA-UD [4] and Faucet [28]) as well as have good performances at the strain-level [29]. However, many contigs generated by the two programs are highly fragmented due to uneven abundances or repeat regions [29]. We chose the data because the original study generated not only metagenomic sequencing data but also nanopore long read data which we can use to examine the performance of Virseqimprover [12]. We BLASTed the contigs generated by metaSPAdes and MEGAHIT against the long nanopore sequences and identified some highly similar contigs to the long read sequences. We chose three contigs, namely S5, S6, and S7, with S5 and S6 having the best hit to the long read sequence AFVG 25M466, and S7 having the best hit to AFVG 25M409, and ran Virseqimprover to extend these contigs.

3.2 Methodology

3.2.1 Overview.

The inputs of Virseqimprover include a viral contig and the metagenomic short reads from which the input contig was assembled. The workflow can be divided into three main steps, the error correction step, the extension step, and the annotation step. The error correction step checks for both the circularity of the contig and the uniformity

of the coverage. The error correction and extension steps are done iteratively until the contig cannot be extended anymore. Then the final extended contig is annotated for its protein content. The output contains the extended contig along with the protein annotation. Figure 17 outlines the three steps of Virseqimprover and details are described in the following.

3.2.2 Error correction.

During the error correction step, the circularity of the contig is checked. Circularity is used as an indicator that the contig recovers the complete genome and therefore, if Virseqimprover finds that a contig is circular, it goes to the annotation step directly, trims the redundant part of the contig, and outputs the contig as the final contig. On the other hand, if Virseqimprover finds that the contig is not circular, it will check for the coverage of the contig and go to the extension step. Figure 18 shows how Virseqimprover checks the circularity of the contig. Assume L_r is the read length and L_s is the length of the contig, Virseqimprover divides the sequence into two parts, G_a and G_b, where G_a starts from $(L_s - 2 \times L_r)$ bp to $(L_s - L_r)$ bp and G_b starts from the beginning or from 1 bp to the beginning of G_a or to $(L_s - 2 \times L_r)$ bp. G_a is aligned against G_b using BLAST [30]. If any part of G_a aligns with G_b with 95% identity and 95% alignment length, Virseqimprover will try to extend the alignment on both sides of the sequences to get the similar region with the maximum length. Then one of the similar regions is trimmed since having the same region twice will be redundant and the contig is marked as circular.

After checking the circularity of the contig, Virseqimprover checks the uniformity of the read coverage and uses it as an indicator for chimericness in the contig. First, per base depth of coverage of the contig is calculated using Samtools [31]. For every base position, if its coverage is within 15th to 85th percentile of all the base coverages, it is considered to be within the normal range and the position is marked as normal, otherwise marked as suspicious. Consecutive bases marked as suspicious form suspicious regions and those longer than 1000 bps are considered to be true suspicious regions. All the regions other than true suspicious regions are flagged as true non-suspicious regions. Virseqimprover chooses the longest true non-suspicious region to extend during the extension step.

3.2.3 Extension.

During the extension step, Virseqimprover first extracts the start and end edges of the contigs using BEDTools [32]. For each edge region, read length * 1.5 is used as the default edge length. Then, for each contig, all the reads are mapped to the edges of the contig using Salmon [33]. SPAdes [34] is used for the local assembly process. The extraction-mapping-assembly step is run iteratively for each contig until it stops growing. When the contig cannot be extended, Virseqimprover trims some bps from both ends of the contig and tries to extend the trimmed contig again. The length of the trimming part ranges from 300 bps to 2,000 bps, depending on how much trimming enables the contig extendable. The logic for trimming the ends is that our empirical investigation shows that assemblers often misassemble in one or both ends of the sequence, causing the assembler to stop prematurely which in turn leads to sequence

segmentation. It is observed that trimming some bases from both ends often helps the assembler to continue the assembly in the right direction. After trimming and extending the contig, if the contig gets extended, the new extended contig goes back to the error correction step; if it cannot be extended after trimming, the extension step ends and the contig goes to the annotation step.

3.2.4 Annotation.

During the annotation step, the contig is checked for circularity. If it is circular, the contig is trimmed to remove the redundant sequence. Then Pilon [15] is applied to the contig to improve the assembly by correcting single nucleotide polymorphisms (SNPs), insertions and deletions. The inputs of Pilon include a genome/contig in FASTA format and reads mapped to the genome in BAM format. From the alignment information, Pilon creates a pileup structure and then corrects the base based on the frequency of each nucleotide in a position. During the base correction step, Pilon also considers if the reads are properly paired or not and the mapping quality of the base. If the alignment of read pairs indicates a discrepancy in the assembly, Pilon tries to fix the assembly by doing a local reassembly in those places. The improved contig after Pilon is sent to Prodigal [35] for ORF prediction and eggNOG-Mapper [36] for protein function annotation using the virus database. Results are visualized using Proksee (CGView) [37] and Clinker [38]. The final output of Virseqimprover contains the extended and improved assembly along with the annotation of the assembly.

Discussion

In this paper, we developed Virseqimprover, a computational pipeline for improving viral assembly by iteratively correcting chimeric sequences based on uniformity of coverage, extending the viral contig, checking the circularity, and then annotating the extended and improved viral contig. By applying Virseqimprover to the draft viral assembly data, we found that Virseqimprover successfully extended and corrected errors for all of these contigs. By comparing the extended contigs with the known reference strains, we found that the extended contigs have high similarity with them, which means that our tool successfully corrected and extended those contigs to as close to their full lengths as possible. As a result, due to the fact that it is challenging for current assemblers to produce complete virus genomes from metagenomic data, Virseqimprover will surely become a useful tool to help the assemblers to generate the viral contigs correctly to nearly their full lengths.

Despite the advantages of Virseqimprover on correcting and extending viral contigs from metagenomic reads, Virseqimprover also has some limitations. One limitation is that during the coverage checking step, Virseqimprover does not check the GC content of the suspicious regions. But in Illumina sequencing, very high or very low GC content ($> 70\%$ or $< 30\%$) can result in reduced mapping coverage and higher error rates. As a result, a low coverage region with high or low GC content can be actually part of the contig, while Virseqimprover can wrongly mark it as a suspicious region and discard that region. Another limitation is that it can incorrectly mark a linear phage as a circular one. Some linear phages may have

repetitive sequences at the ends and because of these repetitive sequences, assemblers can start the assembly of the phage again from the beginning and during the circularity checking step of Virseqimprover, and it will mark this phage as a circular genome, which may not be true. Hence, manual examination of the result of each step of Virseqimprover should be done by the user to ensure the accuracy of the result.

Declarations

Acknowledgments. The authors would like to acknowledge Muhit Islam Emon (Virginia Tech) for help on the web server deployment.

Funding. This work was supported in part by the U.S. National Science Foundation Award 2004751. Funding for open access charge: U.S. National Science Foundation.

Conflict of interest. The authors declare that they have no competing interests.

Ethics approval and consent to participate. No ethical approval was required for this study. All public datasets used in the paper were generated by other organizations that have obtained ethical approval.

Consent for publication. Not applicable.

Software and code availability. Virseqimprover is freely available in the GitHub repository (https://github.com/haoqiusong/Virseqimprover) and is released under the MIT license. Virseqimprover can be easily installed using Conda. Moreover, a web server (http://virchecker.cs.vt.edu/virseqimprover) is developed for researchers to use the tool.

Authors' contributions. HS, ST, FA, RJ and LZ conceived the idea, developed the method, and designed the study. HS and ST implemented the program and performed the analyses. HS implemented and deployed the web server. HS, ST and LZ wrote the manuscript. FA and RJ helped revise the manuscript. All authors read and approved the final manuscript.

References

1. Namiki, T., Hachiya, T., Tanaka, H., Sakakibara, Y.: Metavelvet: an extension of velvet assembler to de novo metagenome assembly from short sequence reads. In: Proceedings of the 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine, pp. 116–124 (2011)
2. Nurk, S., Meleshko, D., Korobeynikov, A., Pevzner, P.A.: metaspades: a new versatile metagenomic assembler. Genome research **27**(5), 824–834 (2017)
3. Boisvert, S., Raymond, F., Godzaridis, E´., Laviolette, F., Corbeil, J.: Ray meta: scalable de novo metagenome assembly and profiling. Genome biology **13**(12), 1–13 (2012)
4. Peng, Y., Leung, H.C., Yiu, S.-M., Chin, F.Y.: Idba-ud: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. Bioinformatics **28**(11), 1420–1428 (2012)

5. Li, D., Liu, C.-M., Luo, R., Sadakane, K., Lam, T.-W.: Megahit: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de bruijn graph. Bioinformatics **31**(10), 1674–1676 (2015)
6. Bickhart, D.M., Kolmogorov, M., Tseng, E., Portik, D.M., Korobeynikov, A., Tolstoganov, I., Uritskiy, G., Liachko, I., Sullivan, S.T., Shin, S.B., *et al.*: Generating lineage-resolved, complete metagenome-assembled genomes from complex microbial communities. Nature biotechnology **40**(5), 711–719 (2022)
7. Garc´ıa-L´opez, R., V´azquez-Castellanos, J.F., Moya, A.: Fragmentation and coverage variation in viral metagenome assemblies, and their effect in diversity calculations. Frontiers in bioengineering and biotechnology **3**, 141 (2015)
8. Smits, S.L., Bodewes, R., Ruiz-Gonz´alez, A., Baumg¨artner, W., Koopmans, M.P., Osterhaus, A.D., Schu¨rch, A.C.: Recovering full-length viral genomes from metagenomes. Frontiers in microbiology **6**, 1069 (2015)
9. V´azquez-Castellanos, J.F., Garc´ıa-L´opez, R., P´erez-Brocal, V., Pignatelli, M., Moya, A.: Comparison of different assembly and annotation tools on analysis of simulated viral metagenomic communities in the gut. BMC genomics **15**(1), 1–20 (2014)
10. Dutilh, B., Reyes, A., Hall, R., Whiteson, K.: Editorial: Virus Discovery by Metagenomics: The (Im) possibilities. Front Microbiol. 8: 1710 (2017)
11. Breitbart, M., Salamon, P., Andresen, B., Mahaffy, J.M., Segall, A.M., Mead, D., Azam, F., Rohwer, F.: Genomic analysis of uncultured marine viral communities. Proceedings of the National Academy of Sciences **99**(22), 14250–14255 (2002)
12. Beaulaurier, J., Luo, E., Eppley, J.M., Den Uyl, P., Dai, X., Burger, A., Turner, D.J., Pendelton, M., Juul, S., Harrington, E., *et al.*: Assembly-free single-molecule sequencing recovers complete virus genomes from natural microbial communities. Genome Research **30**(3), 437–446 (2020)
13. Liang, K.-c., Sakakibara, Y.: Metavelvet-dl: a metavelvet deep learning extension for de novo metagenome assembly. BMC bioinformatics **22**(6), 1–21 (2021)
14. Afiahayati, Sato, K., Sakakibara, Y.: Metavelvet-sl: an extension of the velvet assembler to a de novo metagenomic assembler utilizing supervised learning. DNA research **22**(1), 69–77 (2015)
15. Walker, B.J., Abeel, T., Shea, T., Priest, M., Abouelliel, A., Sakthikumar, S., Cuomo, C.A., Zeng, Q., Wortman, J., Young, S.K., *et al.*: Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. PloS one **9**(11), 112963 (2014)
16. Wick, R.R., Judd, L.M., Gorrie, C.L., Holt, K.E.: Unicycler: resolving bacterial genome assemblies from short and long sequencing reads. PLoS computational biology **13**(6), 1005595 (2017)
17. Boetzer, M., Henkel, C.V., Jansen, H.J., Butler, D., Pirovano, W.: Scaffolding pre-assembled contigs using sspace. Bioinformatics **27**(4), 578–579 (2011)
18. Boetzer, M., Pirovano, W.: Toward almost closed genomes with gapfiller. Genome biology **13**, 1–9 (2012)

19. Farrant, G.K., Hoebeke, M., Partensky, F., Andres, G., Corre, E., Garczarek, L.: Wisescaffolder: an algorithm for the semi-automatic scaffolding of next generation sequencing data. BMC bioinformatics **16**(1), 1–13 (2015)
20. Deng, Z., Delwart, E.: Contigextender: a new approach to improving de novo sequence assembly for viral metagenomics data. BMC bioinformatics **22**(1), 1–19 (2021)
21. Aylward, F.O., Boeuf, D., Mende, D.R., Wood-Charlson, E.M., Vislova, A., Eppley, J.M., Romano, A.E., DeLong, E.F.: Diel cycling and long-term persistence of viruses in the ocean's euphotic zone. Proceedings of the National Academy of Sciences **114**(43), 11446–11451 (2017)
22. Tithi, S.S., Aylward, F.O., Jensen, R.V., Zhang, L.: Fastviromeexplorer-novel: Recovering draft genomes of novel viruses and phages in metagenomic data. Journal of Computational Biology **30**(4), 391–408 (2023)
23. Madeira, F., Pearce, M., Tivey, A.R., Basutkar, P., Lee, J., Edbali, O., Madhusoodanan, N., Kolesnikov, A., Lopez, R.: Search and sequence analysis tools services from embl-ebi in 2022. Nucleic acids research **50**(W1), 276–279 (2022)
24. Robinson, J.T., Thorvaldsd´ottir, H., Winckler, W., Guttman, M., Lander, E.S., Getz, G., Mesirov, J.P.: Integrative genomics viewer. Nature biotechnology **29**(1), 24–26 (2011)
25. Nayfach, S., Camargo, A.P., Schulz, F., Eloe-Fadrosh, E., Roux, S., Kyrpides, N.C.: Checkv assesses the quality and completeness of metagenome-assembled viral genomes. Nature biotechnology **39**(5), 578–585 (2021)
26. Tithi, S.S., Aylward, F.O., Jensen, R.V., Zhang, L.: Fastviromeexplorer: a pipeline for virus and phage identification and abundance profiling in metagenomics data. PeerJ **6**, 4227 (2018)
27. Kearse, M., Moir, R., Wilson, A., Stones-Havas, S., Cheung, M., Sturrock, S., Buxton, S., Cooper, A., Markowitz, S., Duran, C., *et al.*: Geneious basic: an integrated and extendable desktop software platform for the organization and analysis of sequence data. Bioinformatics **28**(12), 1647–1649 (2012)
28. Rozov, R., Goldshlager, G., Halperin, E., Shamir, R.: Faucet: streaming de novo assembly graph construction. Bioinformatics **34**(1), 147–154 (2018)
29. Wang, Z., Wang, Y., Fuhrman, J.A., Sun, F., Zhu, S.: Assessment of metagenomic assemblers based on hybrid reads of real and simulated metagenomic sequences. Briefings in bioinformatics **21**(3), 777–790 (2020)
30. Ye, J., McGinnis, S., Madden, T.L.: Blast: improvements for better sequence analysis. Nucleic acids research **34**(suppl 2), 6–9 (2006)
31. Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., Subgroup, .G.P.D.P.: The sequence alignment/map format and samtools. bioinformatics **25**(16), 2078–2079 (2009)
32. Quinlan, A.R., Hall, I.M.: Bedtools: a flexible suite of utilities for comparing genomic features. Bioinformatics **26**(6), 841–842 (2010)

33. Patro, R., Duggal, G., Love, M.I., Irizarry, R.A., Kingsford, C.: Salmon provides fast and bias-aware quantification of transcript expression. Nature methods **14**(4), 417–419 (2017)
34. Bankevich, A., Nurk, S., Antipov, D., Gurevich, A.A., Dvorkin, M., Kulikov, A.S., Lesin, V.M., Nikolenko, S.I., Pham, S., Prjibelski, A.D., *et al.*: Spades: a new genome assembly algorithm and its applications to single-cell sequencing. Journal of computational biology **19**(5), 455–477 (2012)
35. Hyatt, D., Chen, G.-L., LoCascio, P.F., Land, M.L., Larimer, F.W., Hauser, L.J.: Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC bioinformatics **11**, 1–11 (2010)
36. Huerta-Cepas, J., Forslund, K., Coelho, L.P., Szklarczyk, D., Jensen, L.J., Von Mering, C., Bork, P.: Fast genome-wide functional annotation through orthology assignment by eggnog-mapper. Molecular biology and evolution **34**(8), 2115–2122 (2017)
37. Grant, J.R., Stothard, P.: The cgview server: a comparative genomics tool for circular genomes. Nucleic acids research **36**(suppl 2), 181–184 (2008)
38. Gilchrist, C.L., Chooi, Y.-H.: Clinker & clustermap. js: Automatic generation of gene cluster comparison figures. Bioinformatics **37**(16), 2473–2475 (2021)

Figures

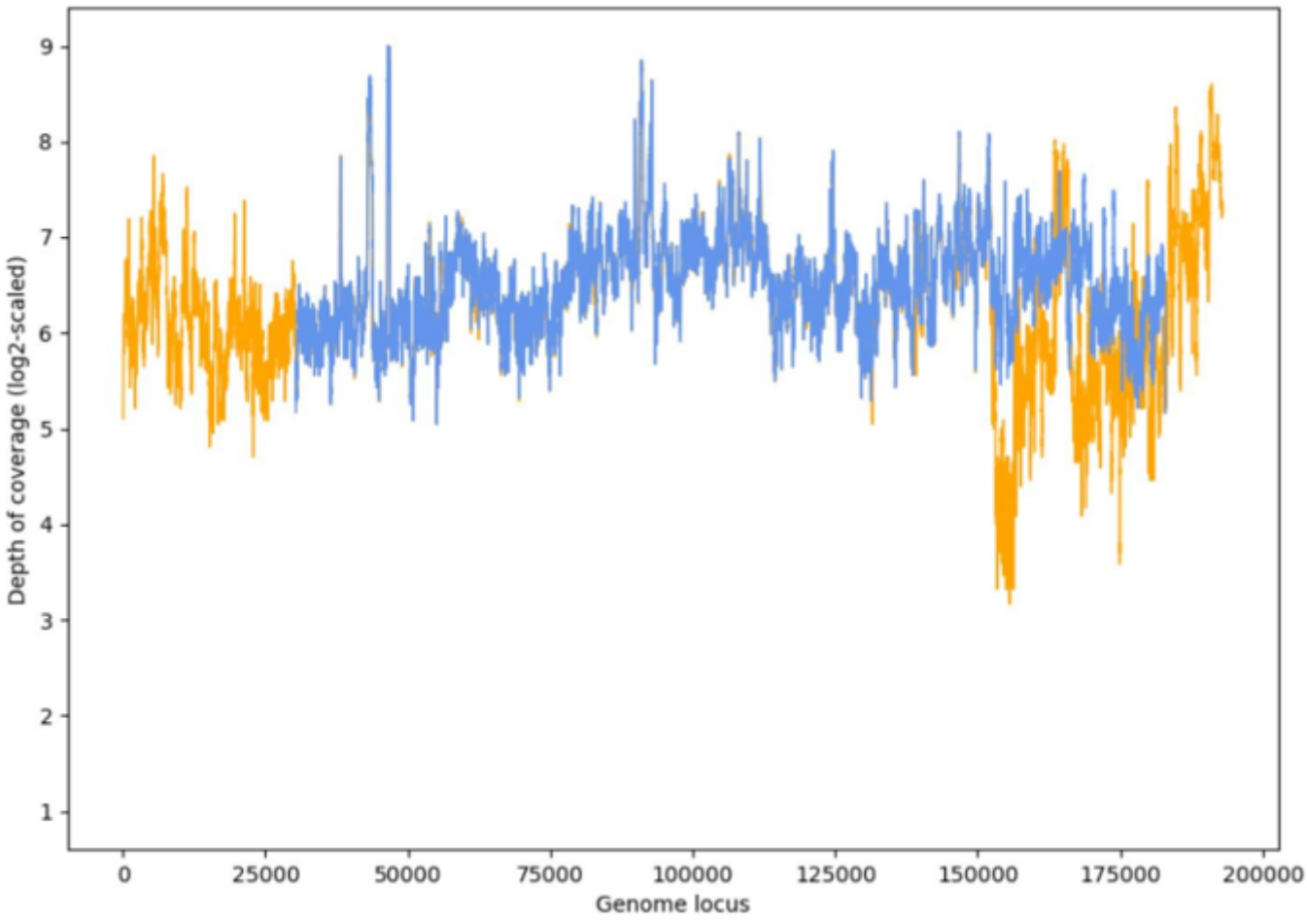

Figure 1

The log2-scaled depth of coverage of the original contig S0 (length 193,112 bp, the orange line) and the improved version S0’ by Virseqimprover (length 152,707 bp, the blue line).

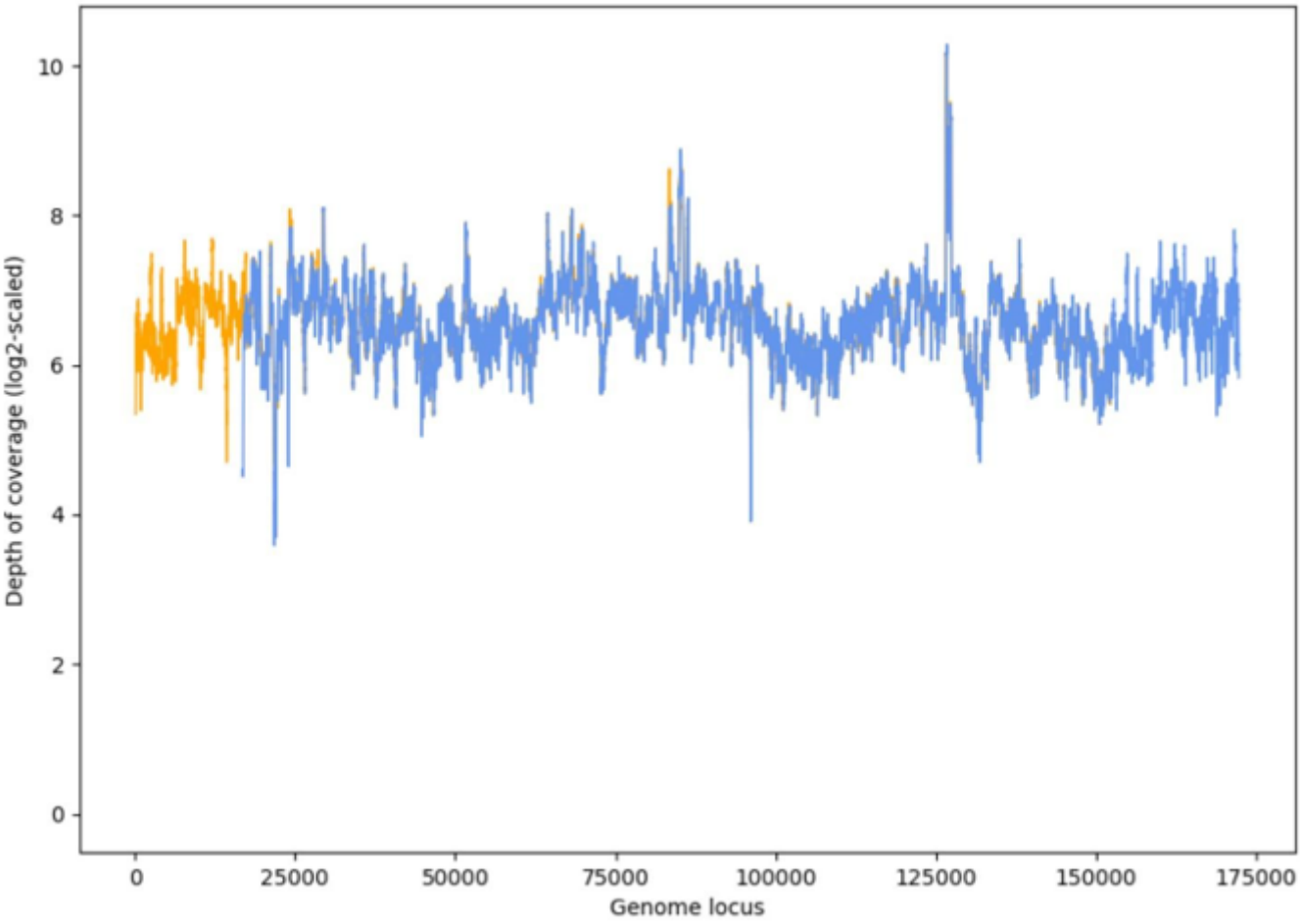

Figure 2

The log2-scaled depth of coverage of the original contig S1 (length 155,659 bp, the orange line) and the improved version S1’ by Virseqimprover (length 152,362 bp, the blue line).

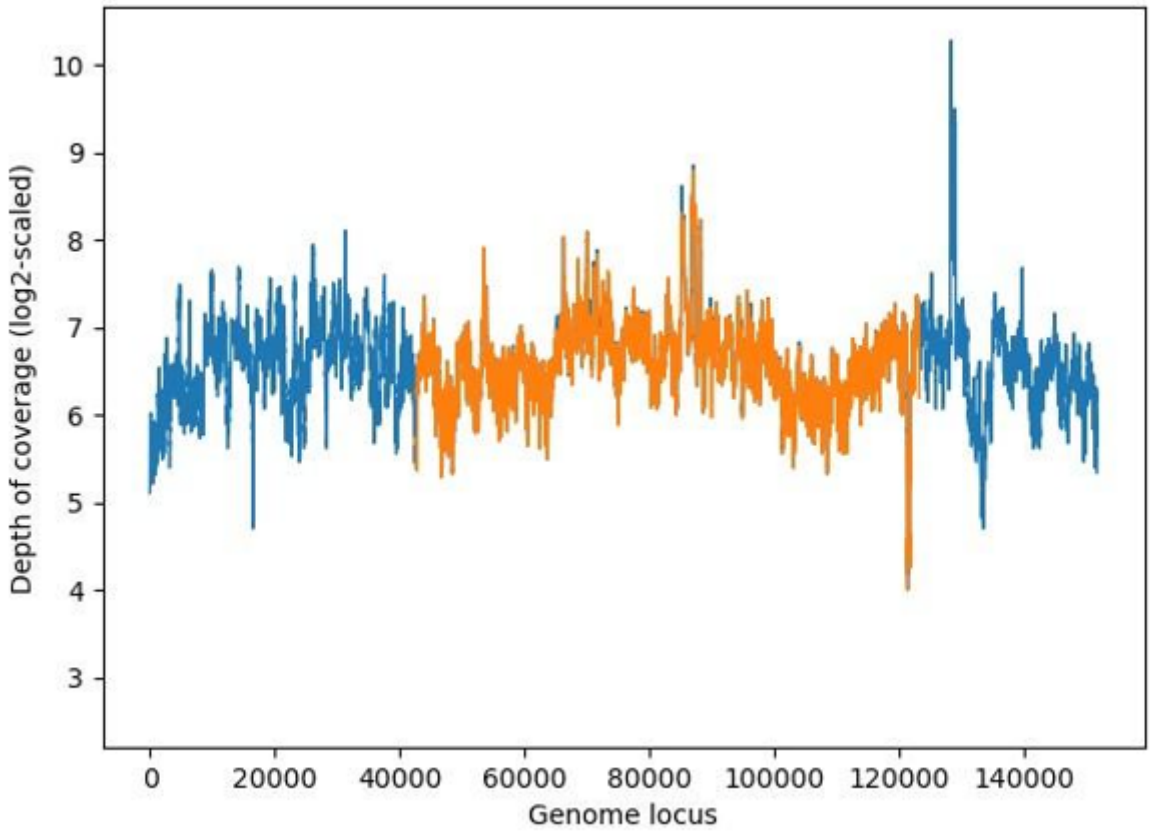

Figure 3

The log2-scaled depth of coverage of the original contig S2 (length 80,620 bp, the orange line) and the extended version S2’ by Virseqimprover (length 151,828 bp, the blue line).

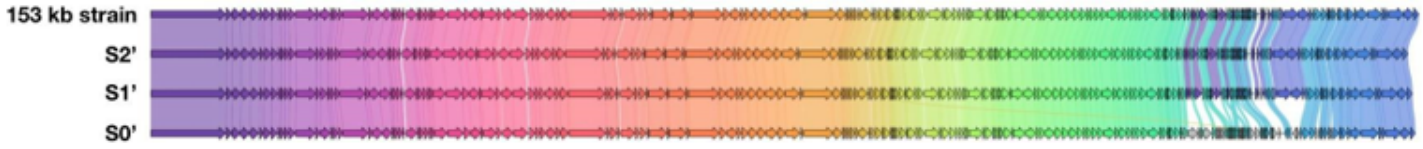

Figure 4

Visualization of gene cluster comparison of the 153 kb strain with the improved contigs S0’, S1’, and S2’ obtained from Virseqimprover.

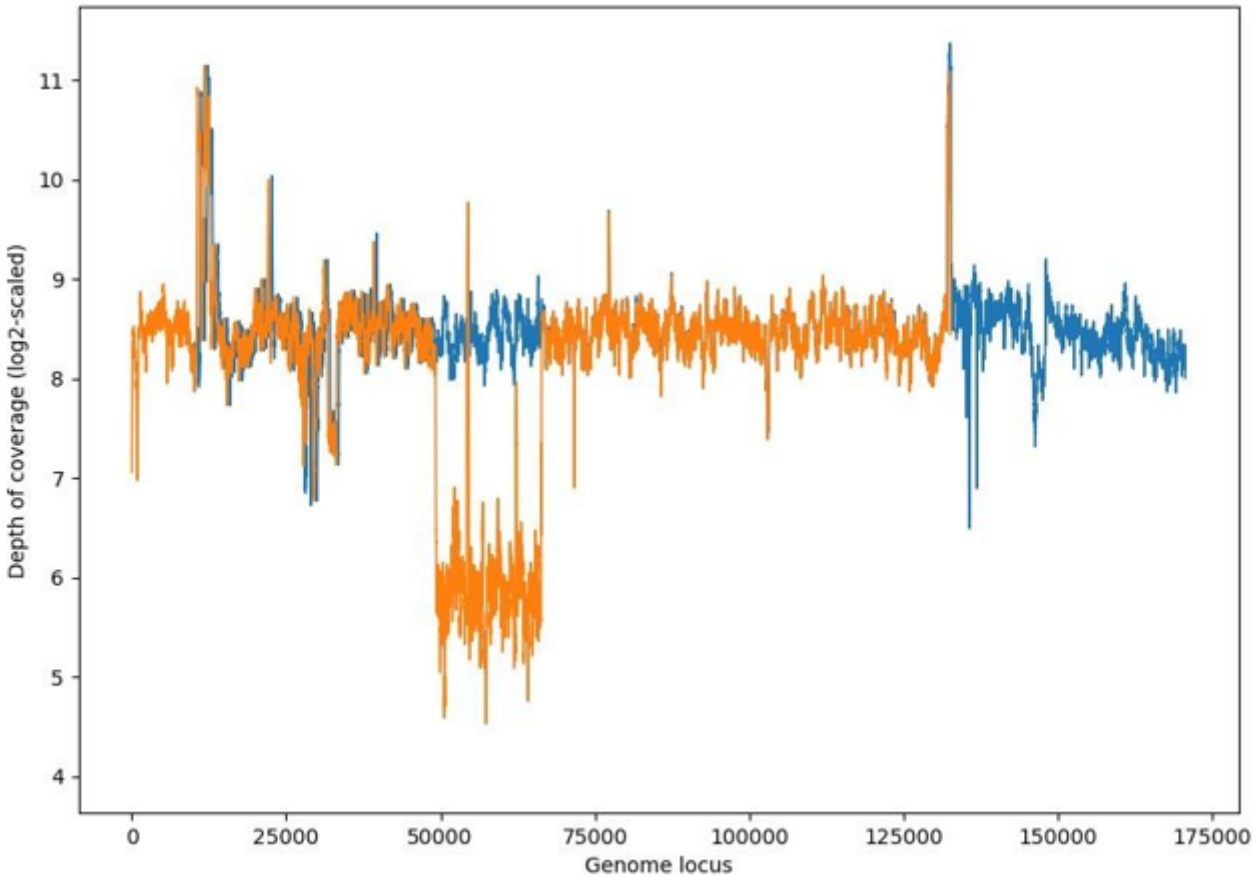

Figure 5

The log2-scaled depth of coverage of the original contig S3 (length 132,604 bp, the orange line) and the improved version S3’ by Virseqimprover (length 160,744 bp, the blue line).

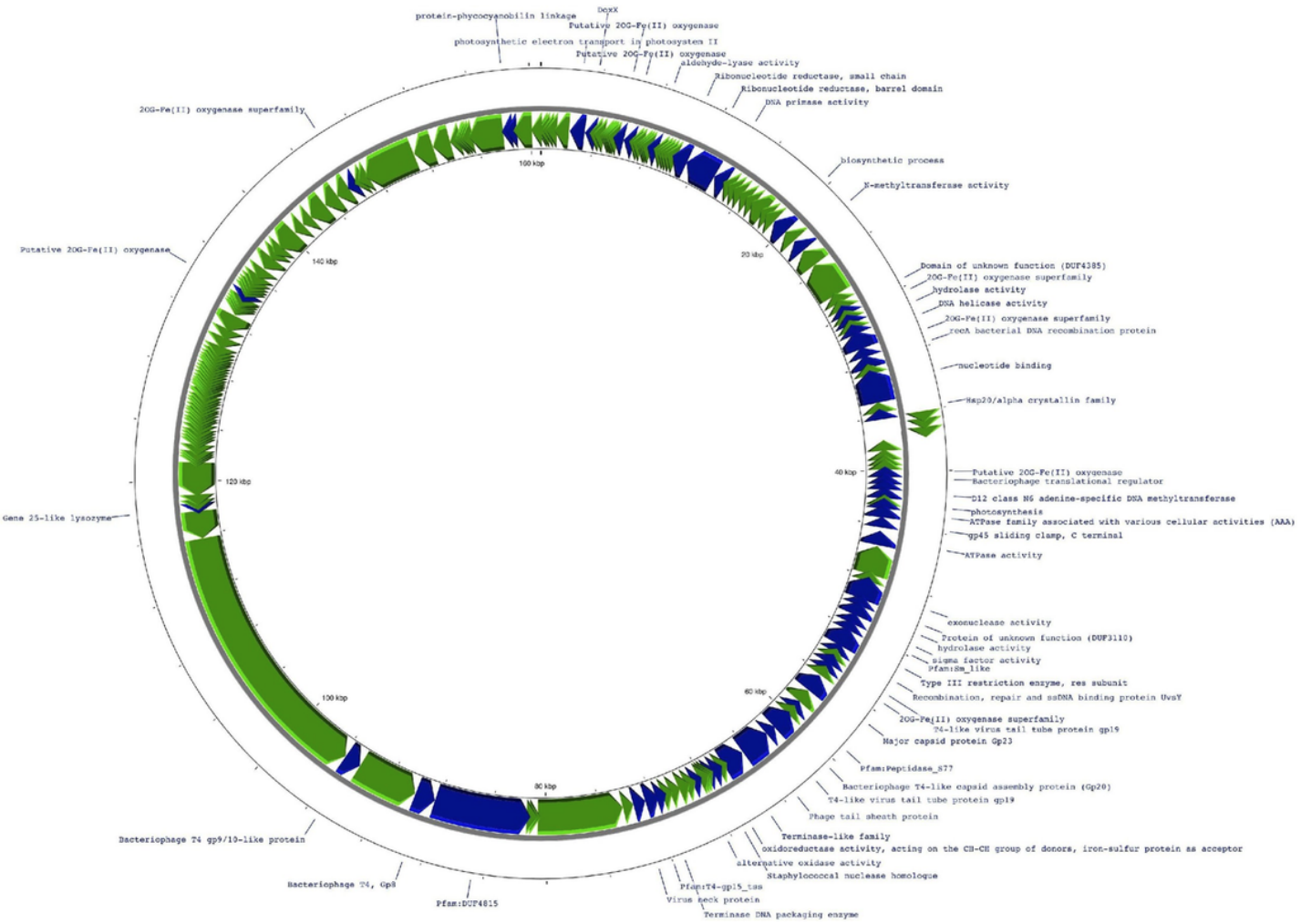

Figure 6

The protein annotation of contig S3’ from Virseqimprover using virus database. The blue arrows are annotated proteins while the green arrows are unannotated proteins.

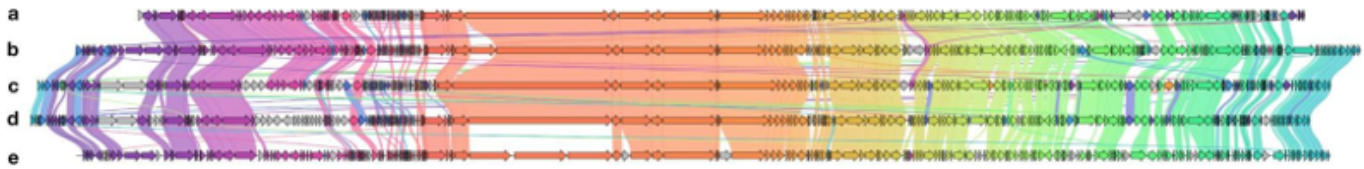

Figure 7

Visualization of gene cluster comparisons between (a) the improved contig S3’, (b) Cyanophage P-RSM1, (c) Prochlorococcus phage P-SSM4, (d) Prochlorococcus phage P-RSM4, and

(e) Synechococcus phage metaG-MbCM1.

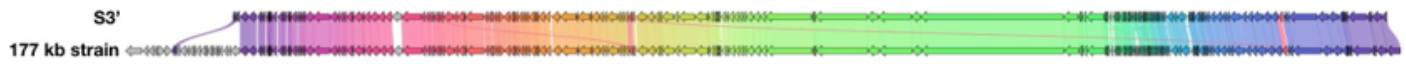

Figure 8

Visualization of gene cluster comparison of the 177 kb strain of Synechococcus phage with the improved contig S3’ obtained from Virseqimprover.

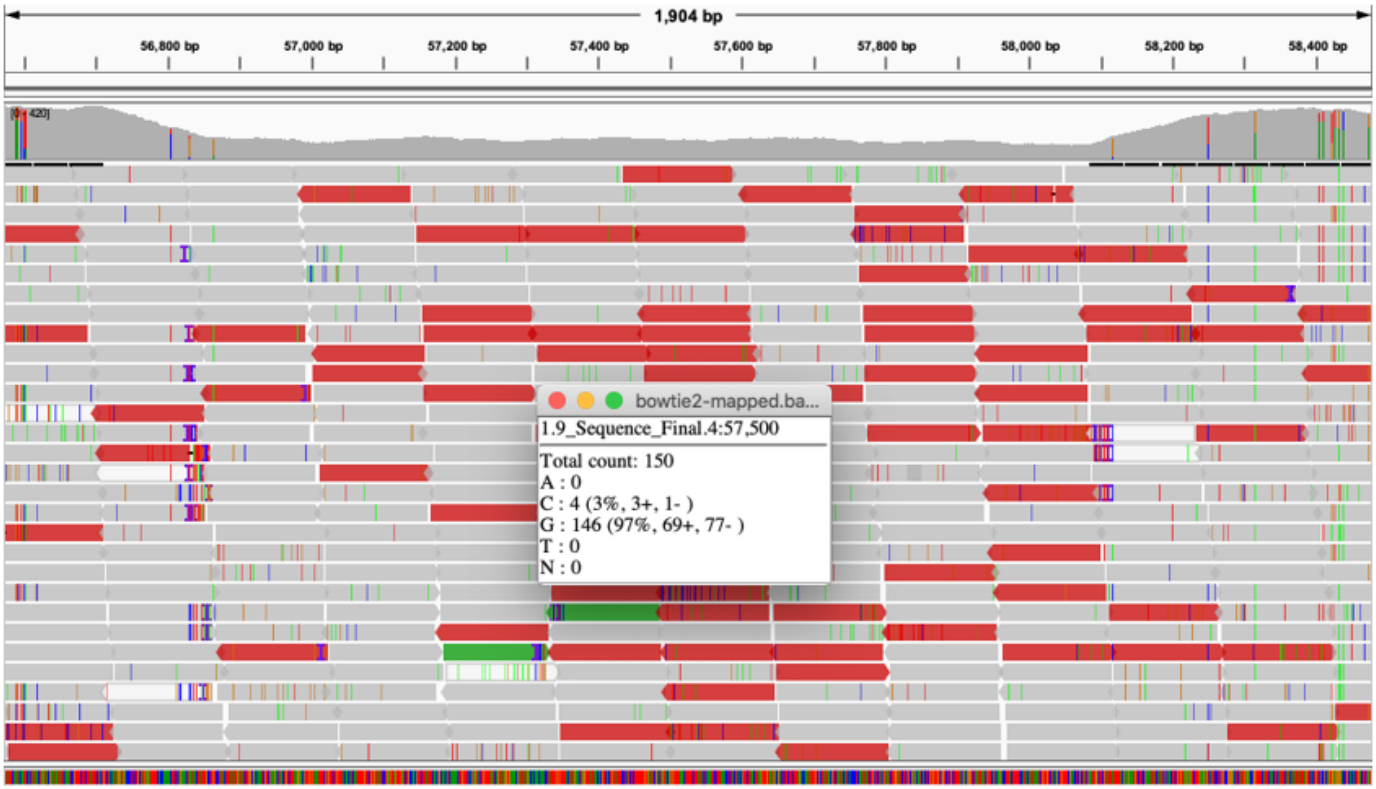

Figure 9

Visualization of the low coverage region of the 177 kb strain, drawn by IGV [24].

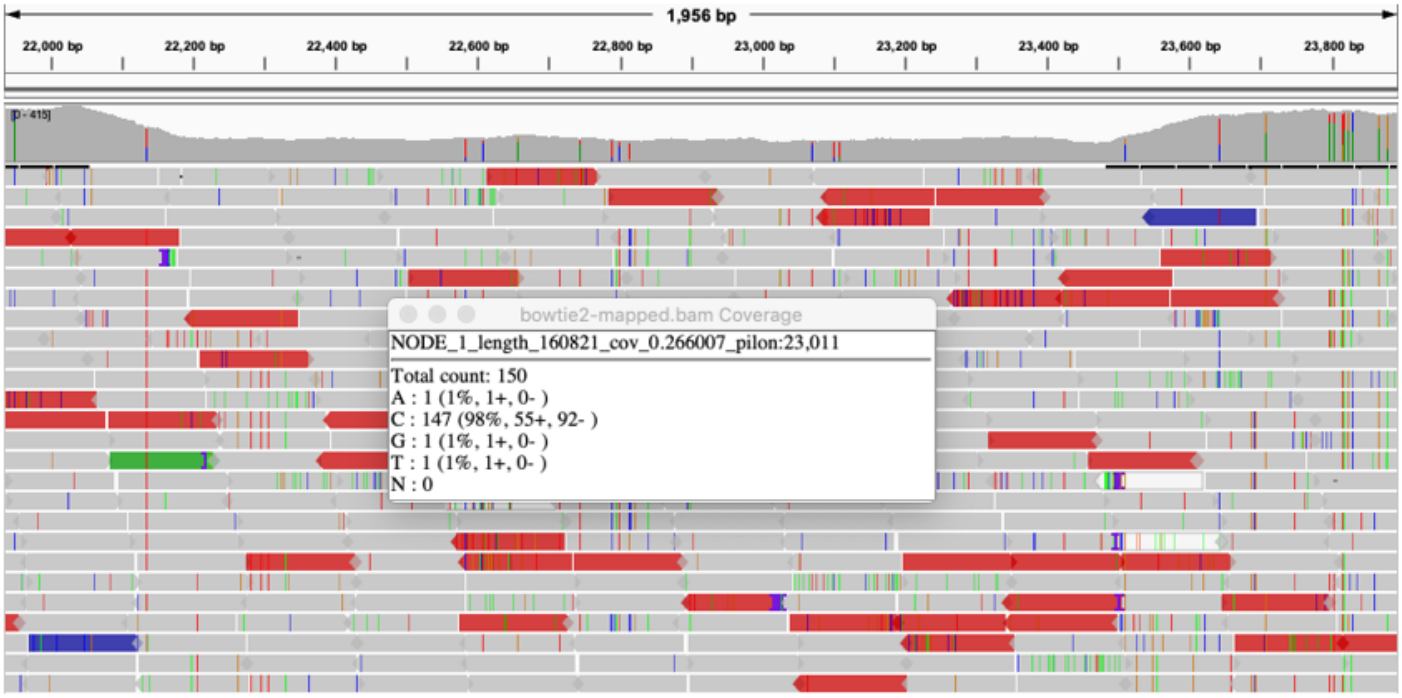

Figure 10

Visualization of the low coverage region of S3', drawn by IGV [24].

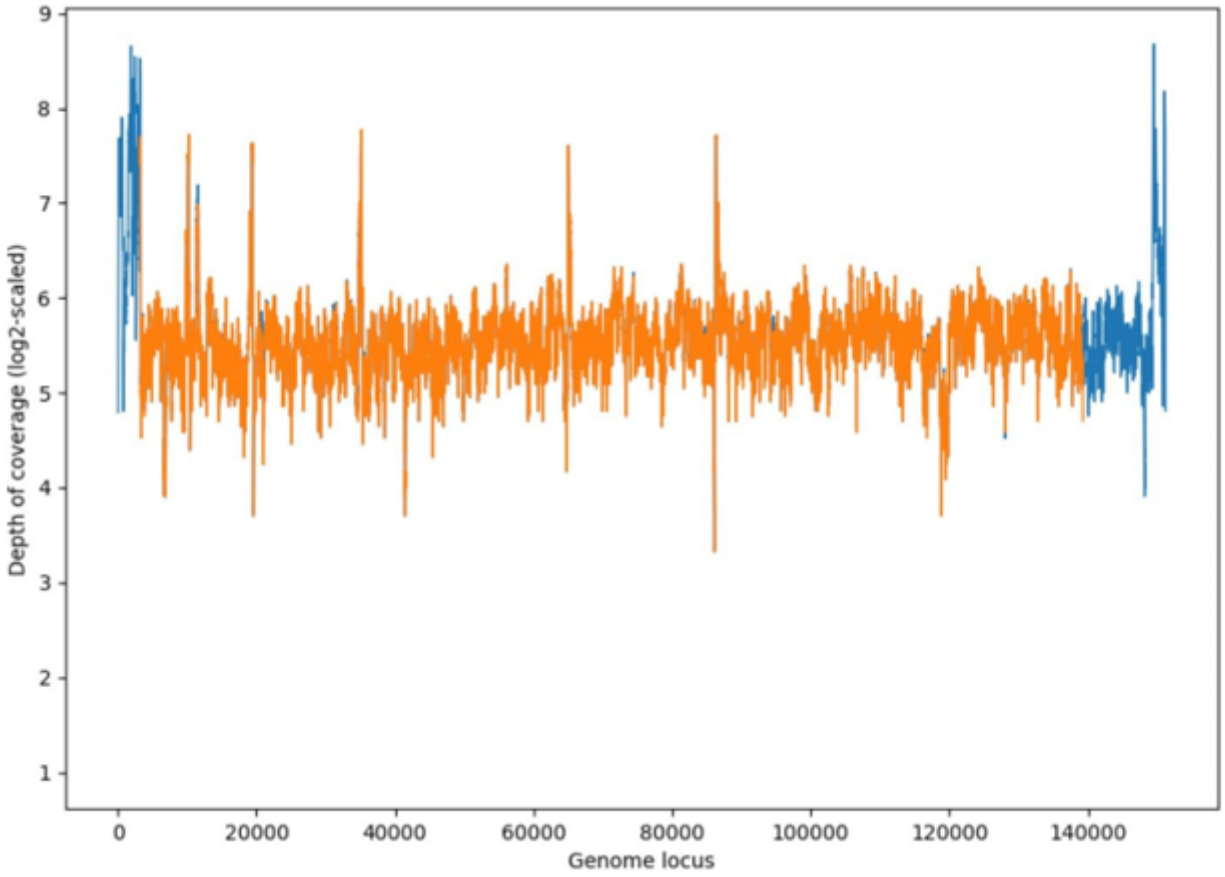

Figure 11

The log2-scaled depth of coverage of the original contig S4 (length 136,254 bp, the orange line) and the improved version S4’ by Virseqimprover (length 151,190 bp, the blue line).

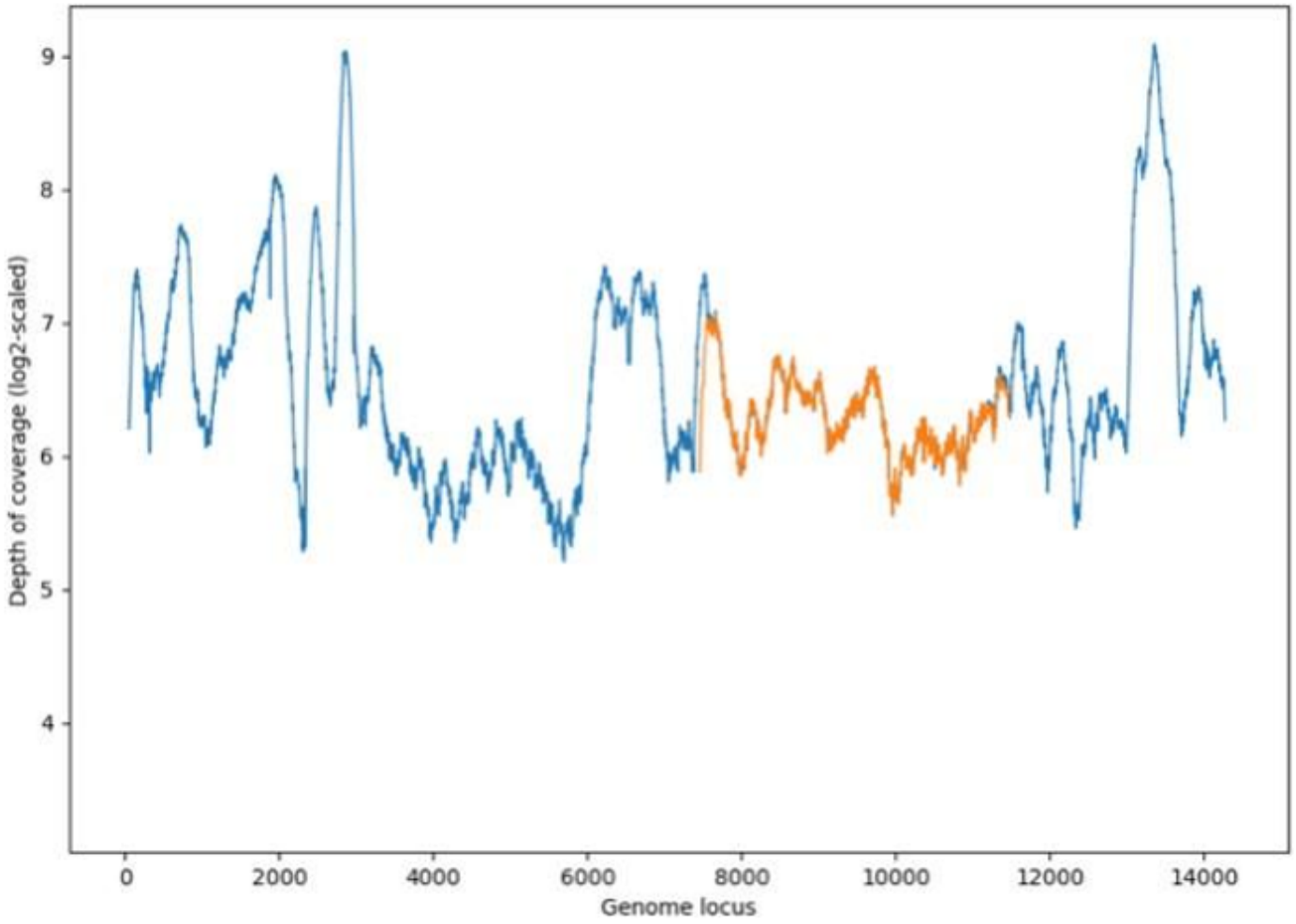

Figure 12

The log2-scaled depth of coverage of the original contig S5 (length 4,179 bp, the orange line) and the improved version S5’ by Virseqimprover (length 14,374 bp, the blue line).

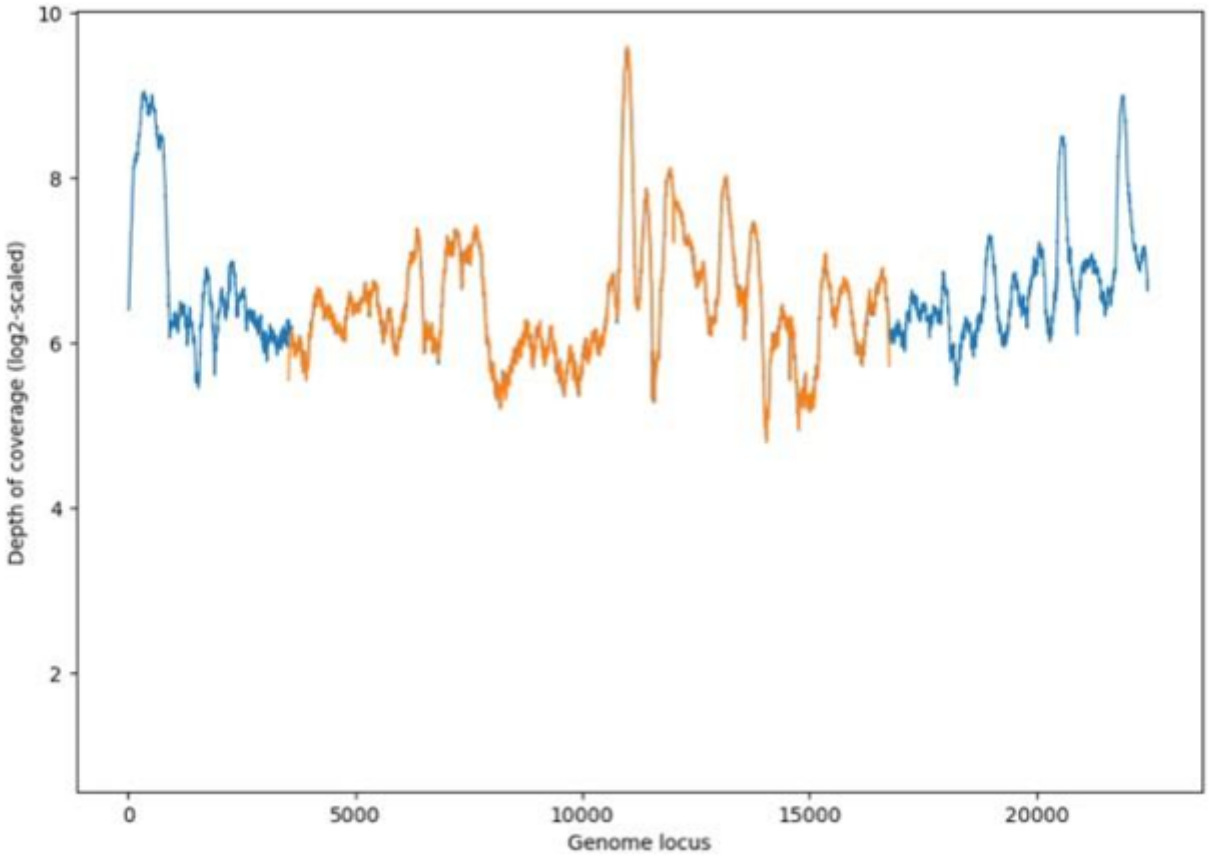

Figure 13

The log2-scaled depth of coverage of the original contig S6 (length 13,396 bp, the orange line) and the improved version S6' by Virseqimprover (length 22,526 bp, the blue line).

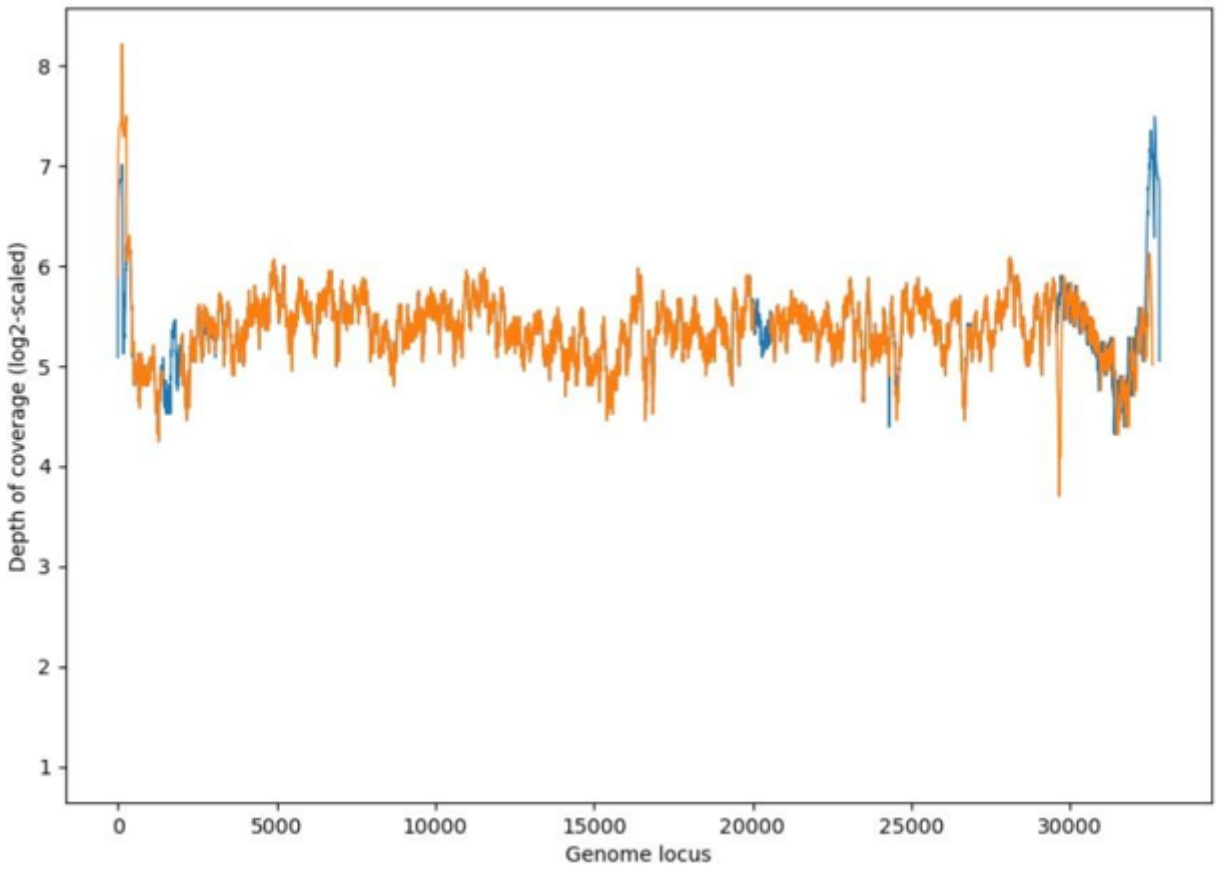

Figure 14

The log2-scaled depth of coverage of the recovered sequence S7’ by Virseqimprover (length 32,035 bp, the orange line) and the reference genome (length 32,812 bp, the blue line).

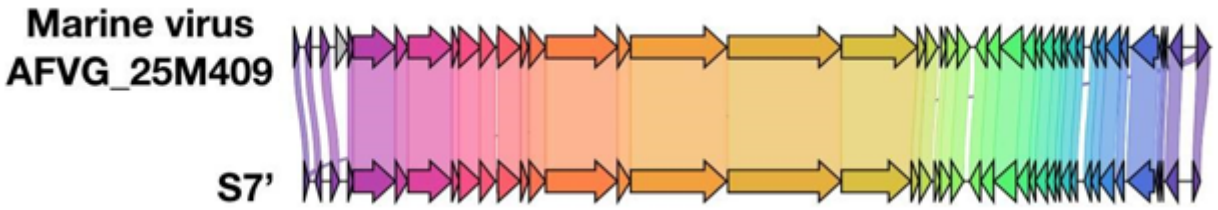

Figure 15

Visualization of gene cluster comparison of the reference genome Marine virus AFVG 25M409 with the improved contig S7’ obtained from Virseqimprover.

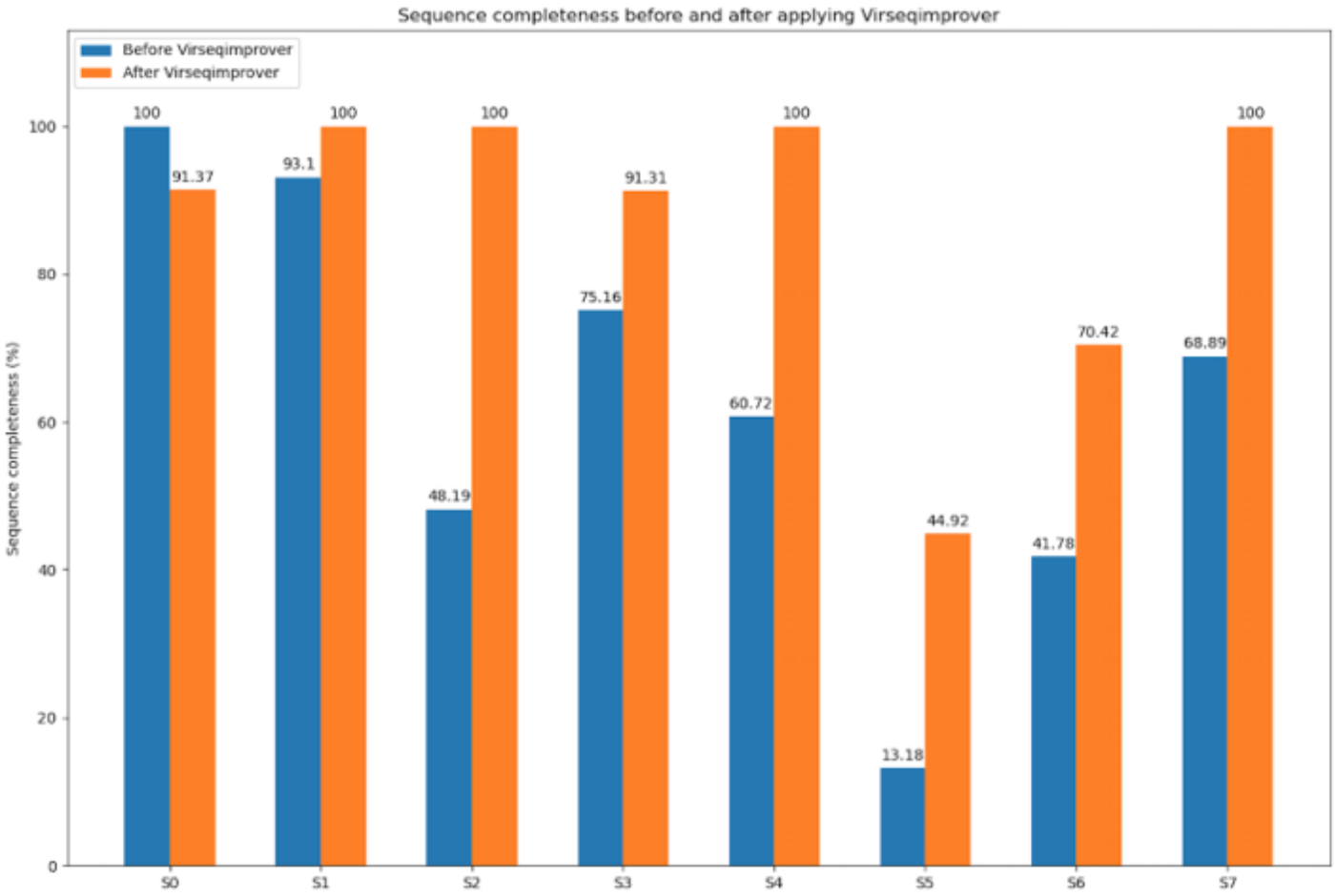

Figure 16

Sequence completeness values of each contig before and after applying Virseqimprover.

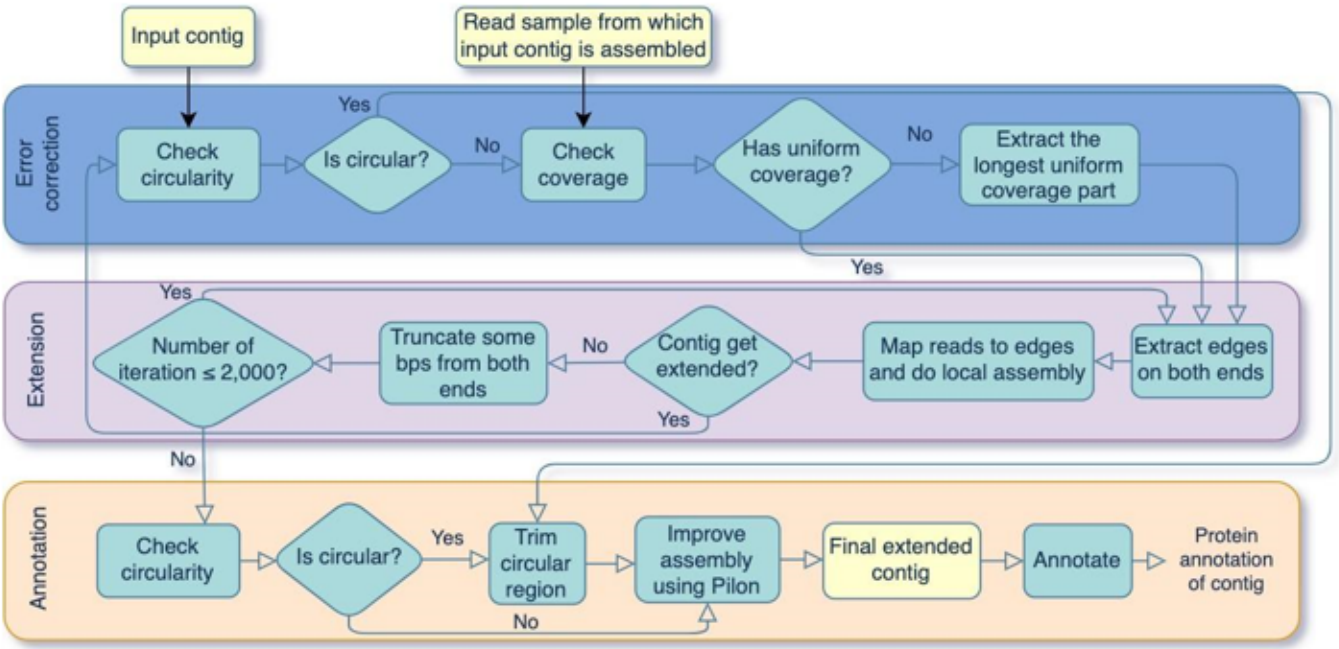

Figure 17

Overview of the Virseqimprover pipeline, where the input is a virus contig and the metagenomic reads, the output is an extended assembly with protein annotation information.

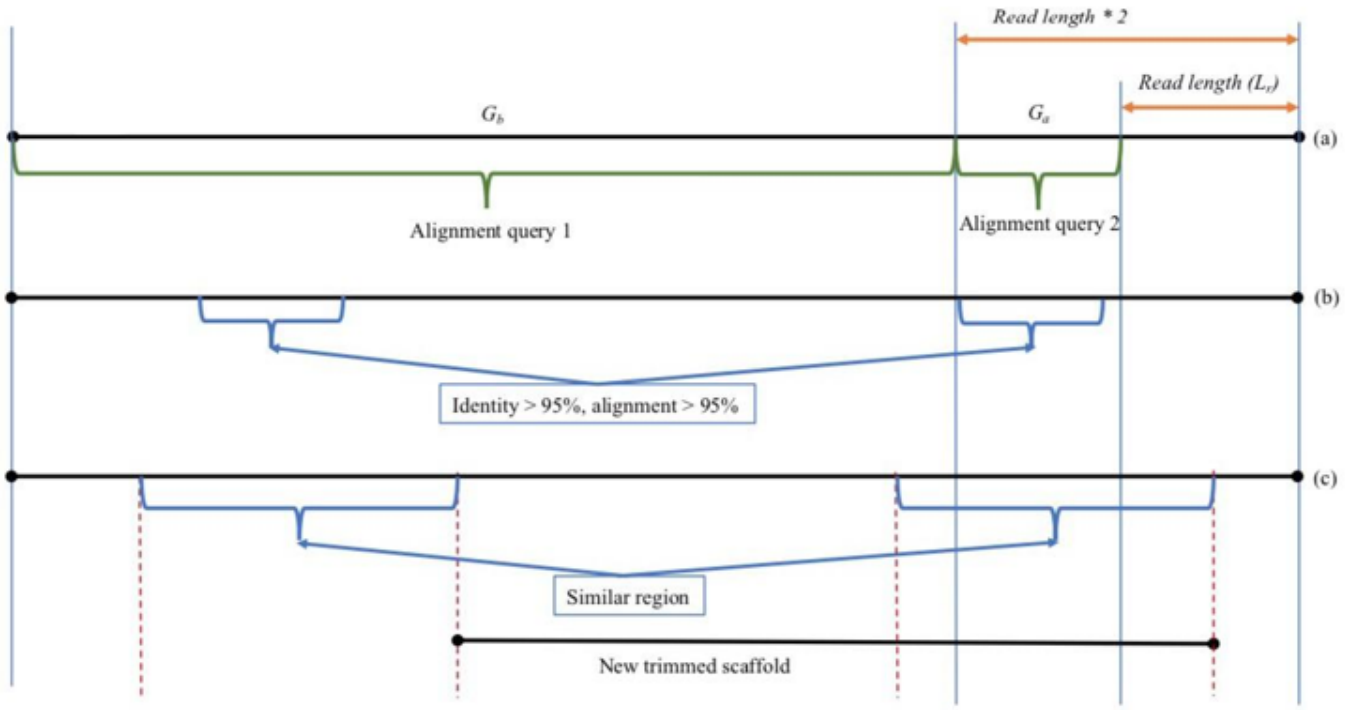

Figure 18

Check the circularity of the sequence: (a) divide the sequence into two parts and align them against each other, (b) find a highly similar region, (c) extend the similar region as much as possible and trim one of the similar regions.