Published online DD MM YYYY *Nucleic Acids Research, YYYY, Vol. xx, No. xx* **1–7**
doi:10.1093/nar/gkn000

ABRIDGE: An ultra-compression software for SAM alignment files

Sagnik Banerjee [1,2*], **Carson Andorf** [3,4 *]

[1]Program in Bioinformatics & Computational Biology, Iowa State University, Ames, IA 50011, USA
[2]Department of Statistics, Iowa State University, Ames, IA 50011, USA
[3]Corn Insects and Crop Genetics Research Unit, USDA-Agricultural Research Service, Ames, IA 50011, USA
[4] Department of Computer Science, Iowa State University, Ames, IA 50011, USA

Received YYYY-MM-DD; Revised YYYY-MM-DD; Accepted YYYY-MM-DD

ABSTRACT

Advancement in technology has enabled sequencing machines to produce vast amounts of genetic data, causing an increase in storage demands. Most genomic software utilizes read alignments for several purposes including transcriptome assembly and gene count estimation. Herein we present, ABRIDGE, a state-of-the-art compressor for SAM alignment files offering users both lossless and lossy compression options. This reference-based file compressor achieves the best compression ratio among all compression software ensuring lower space demand and faster file transmission. Central to the software is a novel algorithm that retains non-redundant information. This new approach has allowed ABRIDGE to achieve a compression 16% higher than the second-best compressor for RNA-Seq reads and over 35% for DNA-Seq reads. ABRIDGE also offers users the option to randomly access location without having to decompress the entire file. ABRIDGE is distributed under MIT license and can be obtained from GitHub (https://github.com/sagnikbanerjee15/Abridge) and docker hub. We anticipate that the user community will adopt ABRIDGE within their existing pipeline encouraging further research in this domain.

INTRODUCTION

Next generation sequencing (NGS) has opened up opportunities to study several biosystems from a quantitative viewpoint ((1, 2, 3, 4)). Over the years, numerous sequencing protocols have been designed to probe the modus operandi of number of biological processes ((5, 6)). Researchers have perfected these protocols - making them more economical and effective. This made sequencing accessible to even underfunded labs leading to a surge in data. Short read data (generated typically on Illumina platforms) is often mapped to a reference (genomic/transcriptomic) and then used for several purposes – assembling ((7, 8, 9, 10)), annotating ((11, 12, 13, 14)), finding differentially expressed genes ((15, 16)) and proteomics ((3, 17, 18, 19)). Most bioinformatics projects utilize a very large set of RNA-Seq or DNA-Seq samples collected from multiple tissue types and conditions. The primary step in such experiments is to align the RNA-Seq samples to a reference that generates a sequence alignment map (SAM) ((20)) that is stored in either a binary alignment map (BAM) ((20)) or compressed alignment file (CRAM) ((21)) format. Even though these formats offer compression to some extent, the total size of all the aligned files can often exceed the storage capacity that small labs can afford. Hence, better compression techniques are needed that utilize the underlying structure of reference alignment files and offer a multitude of options to cater to a diverse range of user requirements.

Short reads, generated by sequencing platforms like Illumina, need to be mapped to a reference using aligners like STAR ((22, 23)), HiSAT2 ((24)) or BWA ((25)) before further processing. These aligners typically output the result in a SAM format which can be converted to a binary BAM format to achieve better compression. SAM format stores the location, shape (CIGAR string) (https://genome.sph.umich.edu/wiki/SAM), nucleotide bases, quality scores and tag level information for each aligned read. Since alignments in SAM format are stored for each read, the file size grows linearly with the number of reads in the sample. Hence, there is a need to devise an algorithm that can exploit the underlying structure of SAM files and offer the best possible compression in a reasonable amount of time.

*To whom correspondence should be addressed. Email: sagnikbanerjee15@gmail.com, carson.andorf@usda.gov

A considerable amount of time and effort has been directed to designing algorithms to compress alignment files to reduce storage demands and facilitate file transfers ((26, 27, 28)). Most approaches achieve compression by eliminating redundant data by accumulating alignment information across multiple reads or alignments. SAM compressors, like NGC ((29)), DeeZ ((30)) and genozip ((31)) are reference based while BAM, CRAM, Quip ((32)) and CSAM ((33)) are reference free. Reference-based approaches achieve compression by representing an aligned read with a description of how it differs from the reference. This eliminates the need to store the actual read sequence thereby reducing storage demands. Quality scores do not map to any reference and hence cannot be compressed like the read string. Hence some compressors like NGC, CSAM, genozip and DeeZ offers users the option to map quality values within a range to a single value. While this can lead to better compression, it might remove quality scores of mismatched bases which are essential for detecting single nucleotide polymorphisms (SNPs). Quip implements Markov chains to encode read sequences and quality scores. Samcomp ((34)) compresses SAM alignments in lossless fashion by tokenizing the read identifiers and sorting the reads as a reference difference model. A very similar approach is undertaken by DeeZ where tokenized read names and read sequence are compressed with delta encoding.

To overcome the shortcoming of previous SAM compression approaches, we introduce ABRIDGE. We offer users a plethora of choices to compress SAM files. To optimize space utilization, ABRIDGE accumulates all reads that are mapped onto the same nucleotide on a reference. ABRIDGE modifies the traditional CIGAR string to store soft-clips, mismatches, insertions, deletions, and quality scores thereby removing the need to store the MD string. To further reduce space demand, ABRIDGE modifies the CIGAR information to store the strand on which the read was mapped. ABRIDGE also offers the option to alter quality scores of nucleotide bases that had a perfect match with the reference thereby reducing even more space (**Supplementary Figure 7**). All features of multi-mapped reads are stored with their individual CIGAR strings. Hence reads mapping to homeologs in polyploid species will retain their alignment profile. Users can choose from three levels of compression offering varying extents of compression with the caveat of the duration of compressing. ABRIDGE offers options of completely lossless compression and selectively lossy conversions. Consequently, decompressions in ABRIDGE can regenerate the entire SAM file with or without modifications depending on the choices made during compression. In this manuscript, we explore the different modes in which ABRIDGE can operate and compare it with other state-of-the-art tools.

MATERIALS AND METHODS

ABRIDGE accepts a single SAM file as input and returns a compressed file that occupies less space than its BAM or CRAM counterpart. Users can choose to retain all the quality scores which would initiate a lossless compression. In several applications, storing the entire quality score is redundant. Hence, ABRIDGE can be configured to preserve only those quality values which for which the corresponding nucleotide base was a mismatch to the reference or an insertion into the read sequence. This option considerably reduces the compressed size but stores the most relevant information which can later be used for analysis that uses quality scores (e.g. variant calling). To further reduce space, users can eliminate quality scores altogether. Some downstream software like transcriptome aligners do not use soft-clips or mismatches, so we designed ABRIDGE to provide options to ignore such information in the SAM file while compressing. ABRIDGE compresses SAM files in two passes – in the first pass, relevant information from the SAM file is rearranged and in the second pass, the file is compressed using generic compressors. ABRIDGE decompresses data by applying the reverse algorithm producing all the requested information to be stored during compression. Once the data is compressed, users can retrieve alignment information from random locations making it very easy to access alignments from anywhere in the genome without having to decompress the entire file.

ABRIDGE achieves a high compression ratio due to the underlying strategies of eliminating redundant data. Instead of storing the entire sequence of reads, ABRIDGE stores the location of the reference to which the read mapped and relevant information about the mismatched and/or inserted base pairs. Instead of storing the exact mapped location, it keeps the difference in mapped position from the previous alignment. This saves a substantial amount of space for both RNA-Seq and DNA-Seq data. ABRIDGE also merges the exact same reads originating from the same nucleotide position of the reference. Read names for uniquely mapped single-ended reads are discarded but are preserved for multi-mapped single ended reads and for paired-ended reads to associate each read with the corresponding fragment. ABRIDGE offers users a multitude of choices for storing quality values. Users can request to store all the quality values without making any changes or allow ABRIDGE to modify the quality scores of some bases to facilitate better and faster compression. Instead of blindly modifying the quality scores, ABRIDGE inspects each base pair and modifies its quality value only if the base pair was aligned perfectly to the reference. Hence, the quality scores of bases which are inserts and/or mismatches are preserved. This provides the users with the opportunity to retain all the relevant information necessary to perform vital downstream analysis. ABRIDGE stores a modified version of the CIGAR string by including soft clipped bases, quality scores of mismatched and inserted bases along with nucleotides that did not match with the reference. Users are also provided with the choice of achieving best compression by eliminating quality scores altogether. This option is helpful for storing alignments files for the purpose of performing transcriptome assemblies where quality scores are not typically used ((10)).

Unlike the read sequence, quality scores cannot be "mapped" to any reference. Hence ABRIDGE stores quality values as reported and then compresses those with generic compressors. ABRIDGE can store quality values in four different ways – (1) Discard quality values of reference matched bases and include only the mismatched and inserted bases. For this case, quality values are stored within the enhanced CIGAR, (2) Store all quality scores with altered

values for reference matched bases, (3) Store all quality values without making any change in the quality values, and (4) Discard quality scores altogether.

Information about the alignment of each read is typically stored in the CIGAR and the MD string. While CIGAR string can indicate the soft-clips, matches, insertion and deletions, it is not designed to store mismatched nucleotides and read inserts. MD string, on the other hand, reports the mismatched bases. Hence, both the CIGAR string and MD string are needed to accurately reconstruct the appropriate alignment of the read to the reference. Since the CIGAR string and the MD string contain overlapping information we decided to integrate them and generate a single representation which we call the integrated CIGAR. The integrated CIGAR contains complete information from which the entire alignment can be reconstructed (**Figure 1**). Quality scores are stored within the integrated CIGAR if the user requests for it. Quality scores for only the mismatched bases and the inserts are stored. If the user requests to store quality scores for all the nucleotide bases then the scores will be stored in a separate file. An illustration of how the integrated CIGAR string is constructed has been provided in **Figure 1**. Once each alignment entry is encoded, an index file is generated that can speed up file access in the future. The index contains information about the location of a pile of reads. During random access, the entire index file is read into the memory. According to the request made by the user, a specific portion of the compressed file is read and subsequently decompressed. Consulting the index file eliminates the need to decompress the whole alignment thereby speeding up random access. Finally, Generic compressors are used to compress the index along with the concise alignment file.

ABRIDGE will generate the compressed file in '.abridge' format which is essentially several files compressed using one of the three compressors - Brotli, 7z or ZPAQ. During decompression, a SAM is produced from the compressed files. The decompression step might require substituting dummy quality scores for some cases, depending on how the quality scores were stored during compression. The decompressed file will be sorted and dummy read names will be produced where they were discarded to save space. Some applications, like genome-guided assembling, do not require the nucleotide sequence. Hence, ABRIDGE allows the user to decompress without generating the actual read sequence. This option is faster to execute since it does not require the reference to be used.

RESULTS

We tested ABRIDGE on RNA-Seq and DNA-Seq data of various read depths (**Supplementary Table 2**). Programs were executed on a cluster with Intel(R) Xeon(R) CPU E5-2670 v2 processors with 2.50 GHz. CentOS Linux release 7.9.2009 was the operating system. ABRIDGE is entirely written in C and gcc compiler (v4.8.5) was used. We carried out experiments using different parameter settings as described in **Table 7**. The first parameter setting produces lossless compression and then we demonstrate how ABRIDGE can be configured to retain the requested information without impacting downstream applications. Details about data acquisition and processing have been mentioned in **Supplementary document**.

SAM file format requirements

Input to ABRIDGE, and also to the other programs, need to be provided in SAM format. The file must be sorted by position and should have a proper SAM header. In addition, each alignment must have three tags - NH, MD, and XS. NH tag stores the number of times the read has been mapped which assists ABRIDGE to distinguish between uniquely mapped and multi mapped reads. MD tag contains information about mismatched bases and deletions which are used to generate a field in the compressed file. XS tag stores information about the strand to which the read was aligned.

ABRIDGE achieves the best lossless compression

ABRIDGE has two major goals - (1) achieve a high level of lossless compression, and (2) provide users with different modes of compression. Lossless compression is achieved by preserving only non-redundant information from SAM alignment file. Alignment files in SAM format were provided as input to the compression software. ABRIDGE performs the best compression owing to the usage of zpaq compressor (**Table 1**). For single-ended reads ABRIDGE discards the read names for uniquely mapped reads. But for paired end reads, ABRIDGE needs to store the read names of both the pairs to enable associating the reads with the same fragment during decompression. This causes a slightly poor compression performance of ABRIDGE (with 7z) (**Figure 2**). CSAM generates a file which is larger than the CRAM file itself. SAMCOMP attains the second best compression for paired-ended reads and third best for single ended reads. GENOZIP and DEEZ exhibit average performance in terms of ratio of compression.

ABRIDGE offers multitude of options for Lossy compression

ABRIDGE offers users with different options of compression as outlined in **Table 7**. Instead of blindly compressing quality scores, ABRIDGE offers users to modify quality scores of those nucleotide bases that perfectly match with the reference. This allows the user to retain the exact quality score of mismatched bases and insertions which would be useful for downstream analysis. With parameter setting number 2, ABRIDGE converts the quality score of matched bases to facilitate vertical run-length encoding leading to higher compression resulting in lower file size (**Supplementary Table 3**). This is further illustrated in **Supplementary Figure 6** where the space requirement for storing quality scores greatly reduces from parameter setting 1 to parameter setting 2. The next set of parameters discard all quality scores except for the non-matched bases. The complete discard of quality scores leads to a further reduction in space requirements. The fourth parameter setting removes all quality scores, soft-clips, and mismatched bases. Since these did not occupy too much space, their removal did not reduce space significantly. In the final parameter setting, only the position of the mapped reads are preserved leading to the smallest file size. As

Figure 1. Generation of integrated CIGAR Each alignment is converted to a string that combines both CIGAR and the MD. The CIGAR and the MD string is expanded to determine the locations of insertion, deletions and mismatches. Soft clips are removed from the front and the end of the read. The reference and the read sequence is consulted to locate the mismatches. The insert nucleotides and the mismatch nucleotides are replaced with special characters. Quality scores are also inserted into the integrated CIGAR for insert and mismatched nucleotides

expected, ZPAQ produces the best compression followed by 7z (**Supplementary Table 3**).

Other software also offer the provision of lossy compression. Both DEEZ and GENOZIP were executed with different parameter settings of lossy compression. ABRIDGE lossy compression, with approximates quality scores (parameter setting 2) was able to produce a better

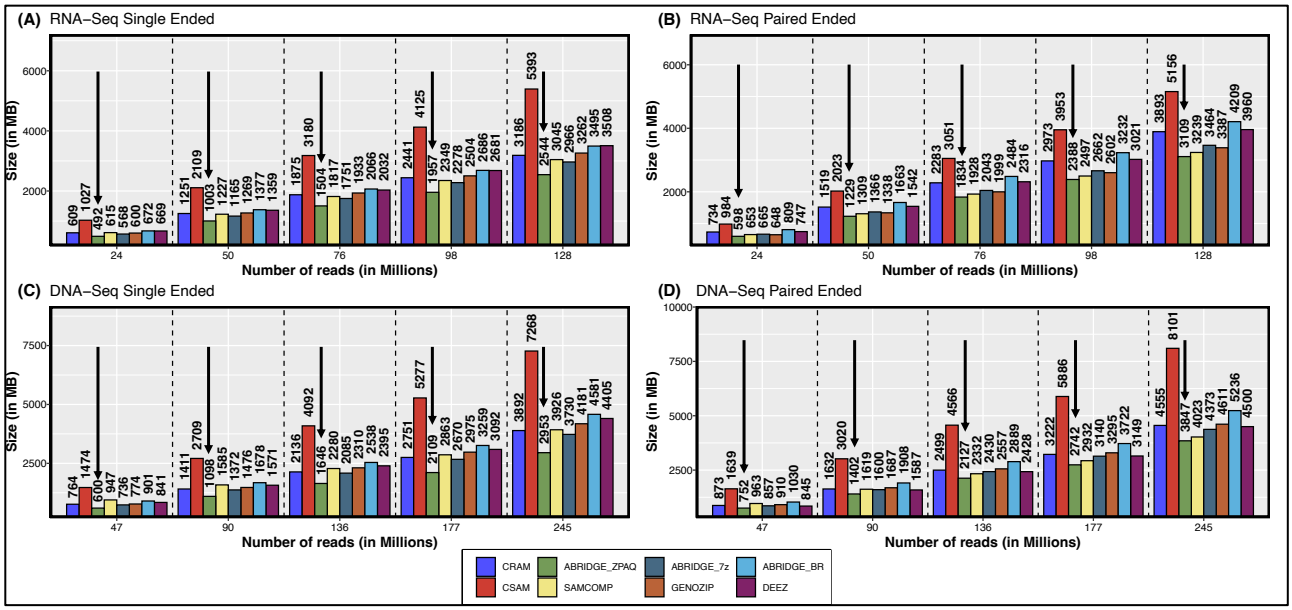

Figure 2. Compression achieved by different software Compressed sizes of files produced by each compression software for RNA-Seq, DNA-Seq, Single ended and Paired ended data. Only software that are regularly updated and maintained have been included in the analysis. Each segment represent compression for increasing file sizes.

compression than all other software operating in lossy mode (**Table 1**).

ABRIDGE quickly compresses data

We compared the duration required to compress the SAM files. Even though ABRIDGE was not able to compress data the fastest, it was comparatively faster than CSAM and GENOZIP **Supplementary Figure 4**. The main bulk of time is taken by the generic compressors (brotli, 7z and ZPAQ) which can be improved by allocating more CPU cores. Compression of a file is performed only once, hence we believe users will not be hesitant to dedicate the time. **Supplementary Table 4** lists the duration of compression for the three generic compressors used in ABRIDGE along with different modes of compression. The duration of compression reduces with more lossy compression for both Brotli and 7Z. It is interesting to note that for ZPAQ the duration does not change much.

ABRIDGE decompresses data faster than other software

In order to utilize the alignments, the compressed files by ABRIDGE need to be decompressed. Unlike compression, decompression needs to be done multiple times depending on how often the alignment files are required to be accessed. Hence, we offer users the choice of multiple compressors that can help decompress files quicker. As depicted in **Supplementary Figure 5**, 7z can decompress files very quickly. Unfortunately, ZPAQ takes the most time to decompress files even when it offers the best compression. Both brotli and 7z take almost the same time to decompress files which were compressed using different parameter settings (**Supplementary Table 4**). ZPAQ, on the other hand, decompresses files faster when the compression was lossy.

ABRIDGE can retrieve data randomly from any location

During compression, ABRIDGE creates an index to facilitate random search. We compared the duration of generating the ABRIDGE index with the time taken to generate the BAM or CRAM index. As listed in **Supplementary Table 6**, CRAM takes the least time to generate indices. BAM and ABRIDGE take almost the same amount of time for single-ended reads. ABRIDGE takes longer for paired-ended reads since it needs to navigate through all the read names to index the file. ABRIDGE consume more memory to generate the indices whereas BAM and CRAM consumes much less memory. Interestingly, CRAM takes the same amount of memory for generating index even when the number of alignments increase.

Random access with ABRIDGE involves decompressing the file and then randomly accessing the requested location. Since ABRIDGE decompresses the entire file, it takes much longer to access random locations than CSAM, GENOZIP, BAM and CRAM. DEEZ takes the longest to access locations randomly since it decompresses the entire file (**Supplementary Table 9**). Both BAM and CRAM consume the least memory while ABRIDGE consumes the most (**Supplementary Table 8**)).

DISCUSSION

We present ABRIDGE - a state-of-the-art software for compressing SAM alignments. ABRIDGE compresses alignments after retaining only non-redundant information. It achieves superior compression by merging similar reads mapped to the same location of the reference and encoding only those nucleotides that deviate from the provided reference. Strand information is also encoded in such a way that it does not occupy any additional space. ABRIDGE

			Size (in MB)															
				ABRIDGE_ZPAQ					DEEZ					GENOZIP				
	Numer of reads (in millions)		Parameters setting 1	Parameters setting 2	Parameters setting 3	Parameters setting 4	Parameters setting 5	0% lossy Mode 1	0% lossy Mode 2	50% lossy Mode 1	50% lossy mode 2	99% lossy Mode 1	99% lossy mode 2	Best Compression	Fast Compression			
RNA-Seq	Single ended	**24**	492	132	93	73	20	669	702	669	702	463	486	600	637			
		50	1003	262	185	146	39	1359	1429	1359	1429	937	985	1269	1334			
		76	1504	379	265	206	53	2032	2138	2032	2138	1394	1465	1933	2014			
		98	1957	496	347	272	68	2681	2818	2681	2818	1851	1943	2504	2704			
		128	2544	644	452	356	86	3508	3687	3508	3687	2425	2546	3262	3506			
	Paired ended	**24**	598	286	250	234	73	747	779	747	779	542	564	648	677			
		50	1229	583	511	477	149	1542	1612	1542	1612	1124	1171	1338	1396			
		76	1834	845	739	687	217	2316	2421	2316	2421	1683	1754	1999	2150			
		98	2388	1106	970	900	277	3021	3157	3021	3157	2197	2289	2602	2697			
		128	3109	1443	1269	1180	360	3960	4138	3960	4138	2886	3006	3387	3639			
DNA-Seq	Single ended	**47**	600	210	178	132	97	841	868	841	868	841	868	774	814			
		90	1098	379	322	242	178	1571	1621	1571	1621	1571	1621	1476	1546			
		136	1646	562	476	365	267	2395	2471	2395	2471	2395	2471	2310	2466			
		177	2109	709	600	460	335	3092	3190	3092	3190	3092	3190	2975	3059			
		245	2953	1043	895	708	447	4405	4540	4405	4540	4405	4540	4181	4420			
	Paired ended	**47**	752	383	314	266	213	845	873	845	873	845	873	910	972			
		90	1402	709	589	500	402	1587	1638	1587	1638	1587	1638	1687	1731			
		136	2127	1069	894	761	609	2428	2506	2428	2506	2428	2506	2557	2753			
		177	2742	1371	1151	979	781	3149	3250	3149	3250	3149	3250	3295	3408			
		245	3847	1970	1672	1438	1072	4500	4639	4500	4639	4500	4639	4611	4952			

Table 1. Comparison of lossless and lossy compression with different software

exploits the sorted file order to store the difference between adjoining mapping positions further reducing space demand. It also discards read names for single-ended uniquely mapped reads which improves compression further. Finally, column-wise conversion of quality scores assists in achieving the best compression.

For ABRIDGE to be a viable compression software, the decompression should be achieved at an acceptable pace. ABRIDGE (with 7z compression) outperforms SAMCOMP, GENOZIP and DEEZ in terms of the duration for decompressing a lossless compressed SAM file. While ABRIDGE with ZPAQ attains the best compression it also takes a much higher time to decompress. Although the decompression time is less than downloading the fastq from NCBI and aligning it to the reference.

Our analysis establishes ABRIDGE as the most recent SAM alignment compressor that offers a very high compression ratio. For single-ended DNA-Seq reads, ABRIDGE produced a file which is 164 MB smaller than the next best compressor. This means that ABRIDGE can achieve an improvement of 15TB with 100K alignment files facilitating both storage and file transmission speed. Additionally, ABRIDGE compressed files can be randomly accessed making it convenient to perform searches without decompressing the entire file.

ABRIDGE provides users the option of choosing either lossless or lossy compression. It is recommended to use lossy compression in conjunction with downstream applications. For example, if the alignment files are produced for transcriptome assembly, then users can do away with quality scores and unmapped reads altogether. But if the downstream application involves SNP calling, then the quality scores (at least for the nucleotides that were a mismatch with the reference) should be preserved. Users are recommended to opt for ZPAQ if they choose to attain ultra high compression ratio. On the other hand, if decompression time is of essence then 7z compression would be the best choice. It is important to remember that ABRIDGE uses the reference file both for compression and decompression. Hence ABRIDGE stores a message digest of the reference to ensure that a correct copy is used for decompression.

An interesting future addition would be to expand ABRIDGE to other file types such as VCF, BED etc. Additionally, we would like to explore options to compress quality scores since those occupy the most space. Further, we will offer users the option to generate coverage information from compressed files directly. We are also collaborating with colleagues to adopt ABRIDGE as an acceptable file format to assembly and gene count software. We pledge to continually develop ABRIDGE to cater to a wide variety of file types storing biological information and facilitate its integration into existing pipelines.

AVAILABILITY

ABRIDGE can be obtained from GitHub via this link https://github.com/sagnikbanerjee15/Abridge.

ACKNOWLEDGEMENTS

The authors acknowledge Dr. Nathan Weeks for insightful discussion about release of software.

FUNDING

This research was supported by the US. Department of Agriculture, Agricultural Research Service, Project No. 5030-21000-068-00D through the Corn Insects and Crop Genetics Research Unit. Research supported in part by Oak Ridge Institute for Science and Education (ORISE) under US Department of Energy (DOE) contract number DE-SC0014664. The funders had no role in study design, data collection and analysis, decision to publish, or preparation

of the manuscript. Mention of trade names or commercial products in this publication is solely for the purpose of providing specific information and does not imply recommendation or endorsement by the USDA, ARS, DOE, or ORAU/ORISE. USDA is an equal opportunity provider and employer.

Conflict of interest statement. None declared.

REFERENCES

1. Richard Hickman, Marcel C Van Verk, Anja J H Van Dijken, Marciel Pereira Mendes, Irene A Vroegop-Vos, Lotte Caarls, Merel Steenbergen, Ivo Van Der Nagel, Gert Jan Wesselink, and Aleksey Jironkin. Architecture and dynamics of the jasmonic acid gene regulatory network. *The Plant Cell Online*, pages tpc–00958, 2017.
2. Marie-Laure Erffelinck, Bianca Ribeiro, Maria Perassolo, Laurens Pauwels, Jacob Pollier, Veronique Storme, and Alain Goossens. A user-friendly platform for yeast two-hybrid library screening using next generation sequencing. *PLOS ONE*, 13(12):e0201270, 5 2018.
3. Matt Hunt, Sagnik Banerjee, Priyanka Surana, Meiling Liu, Greg Fuerst, Sandra Mathioni, Blake C Meyers, Dan Nettleton, and Roger P Wise. Small RNA discovery in the interaction between barley and the powdery mildew pathogen. *BMC genomics*, 20(1):610, 2019.
4. Manjula G. Elmore, Sagnik Banerjee, Kerry F. Pedley, Amy Ruck, and Steven A. Whitham. De novo transcriptome of Phakopsora pachyrhizi uncovers putative effector repertoire during infection. *Physiological and Molecular Plant Pathology*, 110, 4 2020.
5. Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63, 2009.
6. Jason D Buenrostro, Beijing Wu, Howard Y Chang, and William J Greenleaf. ATAC-seq: a method for assaying chromatin accessibility genome-wide. *Current protocols in molecular biology*, 109(1):21–29, 2015.
7. Brian J Haas, Alexie Papanicolaou, Moran Yassour, Manfred Grabherr, Philip D Blood, Joshua Bowden, Matthew Brian Couger, David Eccles, Bo Li, Matthias Lieber, Matthew D MacManes, Michael Ott, Joshua Orvis, Nathalie Pochet, Francesco Strozzi, Nathan Weeks, Rick Westerman, Thomas William, Colin N Dewey, Robert Henschel, Richard D LeDuc, Nir Friedman, and Aviv Regev. De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. *Nature protocols*, 8(8):1494, 8 2013.
8. Mingfu Shao and Carl Kingsford. Accurate assembly of transcripts through phase-preserving graph decomposition. *Nature Biotechnology*, 35(12):1167–1169, 11 2017.
9. Sam Kovaka, Aleksey V Zimin, Geo M Pertea, Roham Razaghi, Steven L Salzberg, and Mihaela Pertea. Transcriptome assembly from long-read RNA-seq alignments with StringTie2. *Genome Biology*, 20(1):1–13, 2019.
10. Li Song, Sarven Sabunciyan, Guangyu Yang, and Liliana Florea. A multi-sample approach increases the accuracy of transcript assembly. *Nature Communications*, 10(1):5000, 12 2019.
11. Brian J Haas, Arthur L Delcher, Stephen M Mount, Jennifer R Wortman, Roger K Smith Jr, Linda I Hannick, Rama Maiti, Catherine M Ronning, Douglas B Rusch, and Christopher D Town. Improving the Arabidopsis genome annotation using maximal transcript alignment assemblies. *Nucleic acids research*, 31(19):5654–5666, 2003.
12. Carson Holt and Mark Yandell. MAKER2: an annotation pipeline and genome-database management tool for second-generation genome projects. *BMC bioinformatics*, 12(1):491, 12 2011.
13. Tomas Bruna, Katharina Hoff, Mario Stanke, Alexandre Lomsadze, and Mark Borodovsky. BRAKER2: Automatic Eukaryotic Genome Annotation with GeneMark-EP+ and AUGUSTUS Supported by a Protein Database. *bioRxiv*, 2020.
14. Sagnik Banerjee, Priyanka Bhandary, Margaret Woodhouse, Taner Z Sen, Roger P Wise, and Carson M Andorf. FINDER: An automated software package to annotate eukaryotic genes from RNA-Seq data and associated protein sequences. *BMC bioinformatics*, page 2021.02.04.429837, 4 2021.
15. Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.
16. Michael Love, Simon Anders, and Wolfgang Huber. Differential analysis of count data–the DESeq2 package. *Genome Biology*, 15:550, 2014.
17. Sagnik Banerjee, Subhadip Basu, and Mita Nasipuri. Big Data Analytics and Its Prospects in Computational Proteomics. In *Information Systems Design and Intelligent Applications*, volume 340, pages 591–598. Springer, 2015.
18. Sagnik Banerjee, Valeria Velasquez-Zapata, Gregory Fuerst, J Mitch Elmore, and Roger P Wise. NGPINT: A Next-generation protein-protein interaction software. *Breifings in Bioinformatics*, 22:in press, 1 2021.
19. Valeria Velásquez-Zapata, James Mitch Elmore, Sagnik Banerjee, Karin S Dorman, and Roger P Wise. Y2H-SCORES: A statistical framework to infer protein-protein interactions from next-generation yeast-two-hybrid sequence data. *bioarxiv*, 2020.
20. Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.
21. Markus Hsi-Yang Fritz, Rasko Leinonen, Guy Cochrane, and Ewan Birney. Efficient storage of high throughput DNA sequencing data using reference-based compression. *Genome research*, 21(5):734–740, 2011.
22. Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, 2013.
23. Alexander Dobin, Thomas R Gingeras, Cold Spring, Roberto Flores, Joshua Sampson, Rob Knight, Nicholas Chia, and High-throughput Sequencing Technologies. Mapping RNA-seq with STAR. *Curr Protoc Bioinformatics*, 51(4):586–597, 2016.
24. Daehwan Kim, Ben Langmead, and Steven L Salzberg. HISAT: a fast spliced aligner with low memory requirements. *Nature Methods*, 12(4):357–360, 3 2015.
25. José M Abuín, Juan C Pichel, Tomás F Pena, and Jorge Amigo. BigBWA: approaching the Burrows–Wheeler aligner to Big Data technologies. *Bioinformatics*, page btv506, 2015.
26. Raffaele Giancarlo, Simona E Rombo, and Filippo Utro. Compressive biological sequence analysis and archival in the era of high-throughput sequencing technologies. *Briefings in bioinformatics*, 15(3):390–406, 2014.
27. Morteza Hosseini, Diogo Pratas, and Armando J Pinho. A survey on data compression methods for biological sequences. *Information*, 7(4):56, 2016.
28. Ibrahim Numanagić, James K Bonfield, Faraz Hach, Jan Voges, Jörn Ostermann, Claudio Alberti, Marco Mattavelli, and S Cenk Sahinalp. Comparison of high-throughput sequencing data compression tools. *nature methods*, 13(12):1005–1008, 2016.
29. Niko Popitsch and Arndt von Haeseler. NGC: lossless and lossy compression of aligned high-throughput sequencing data. *Nucleic acids research*, 41(1):e27–e27, 2013.
30. Faraz Hach, Ibrahim Numanagic, and S Cenk Sahinalp. DeeZ: reference-based compression by local assembly. *Nature methods*, 11(11):1082–1084, 2014.
31. Divon Lan, Ray Tobler, Yassine Souilmi, and Bastien Llamas. Genozip-A Universal Extensible Genomic Data Compressor. *Bioinformatics*, 2021.
32. Daniel C Jones, Walter L Ruzzo, Xinxia Peng, and Michael G Katze. Compression of next-generation sequencing reads aided by highly efficient de novo assembly. *Nucleic acids research*, 40(22):e171–e171, 2012.
33. Rodrigo Cánovas, Alistair Moffat, and Andrew Turpin. Csam: Compressed sam format. *Bioinformatics*, 32(24):3709–3716, 2016.
34. James K Bonfield and Matthew V Mahoney. Compression of FASTQ and SAM format sequencing data. *PloS one*, 8(3):e59190, 2013.

SUPPLEMENTARY FIGURES

SUPPLEMENTARY TABLES

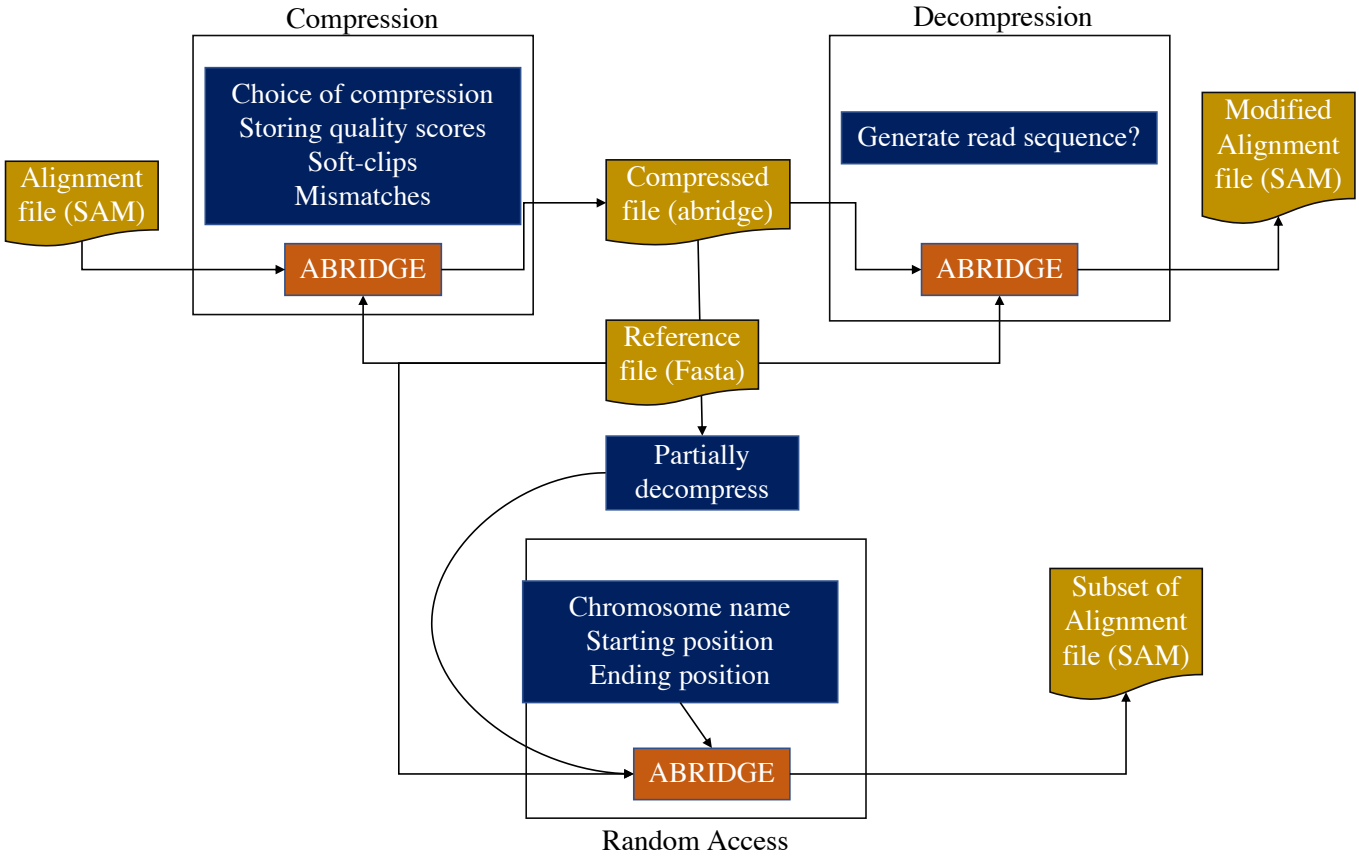

Figure 3. Overview of ABRIDGE software ABRIDGE can be used for compressing alignment files in SAM format. Users have the option of providing multiple different modes of compression. The compressed file can be decompressed as and when required. ABRIDGE also offers users the opportunity to access random locations from the compressed file. All operations require the reference in fasta format.

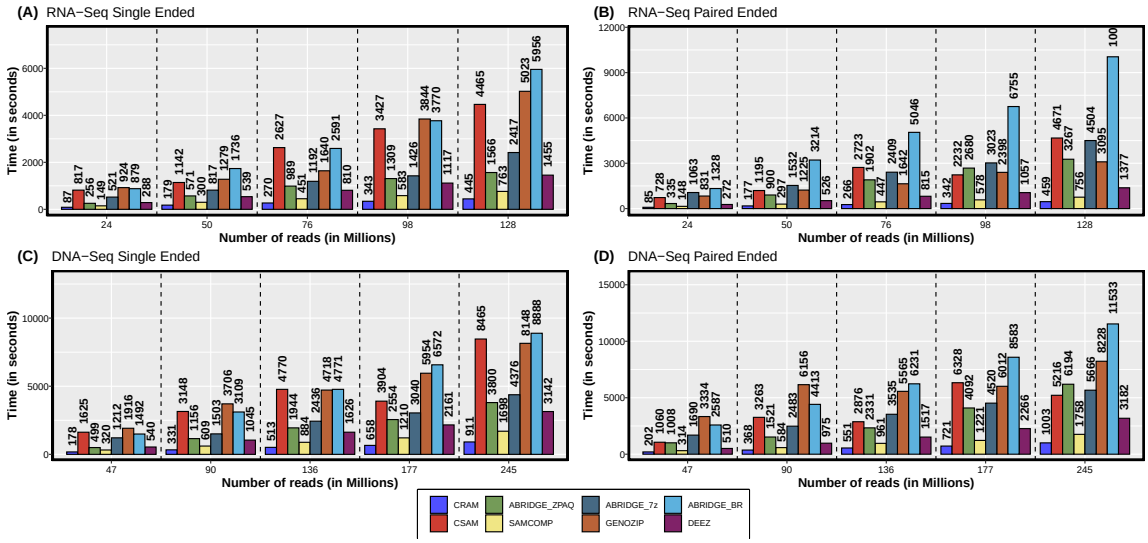

Figure 4. Comparison of time taken to compress SAM file

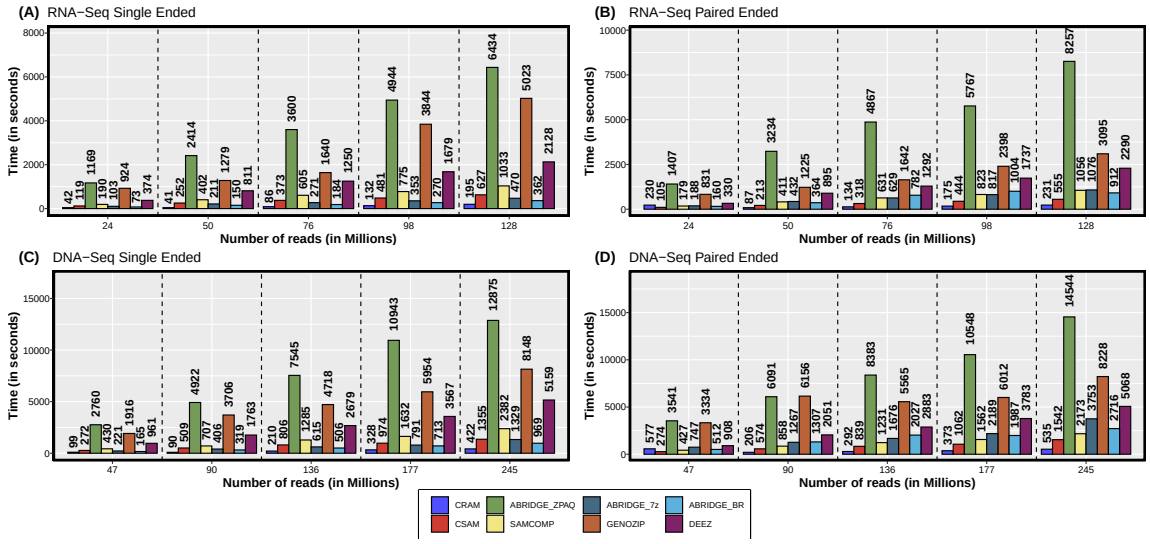

Figure 5. Comparison of time taken to decompress into SAM file

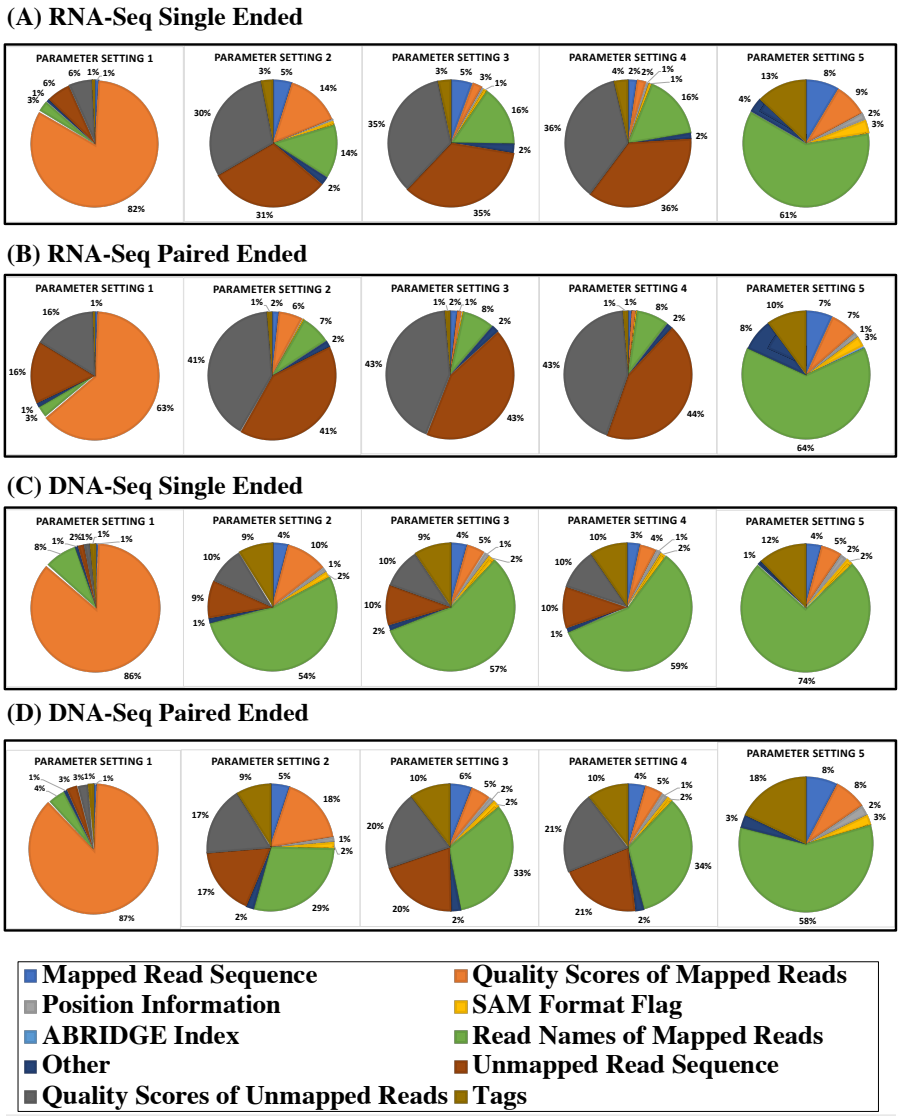

Figure 6. Various modes of compression offered by ABRIDGE

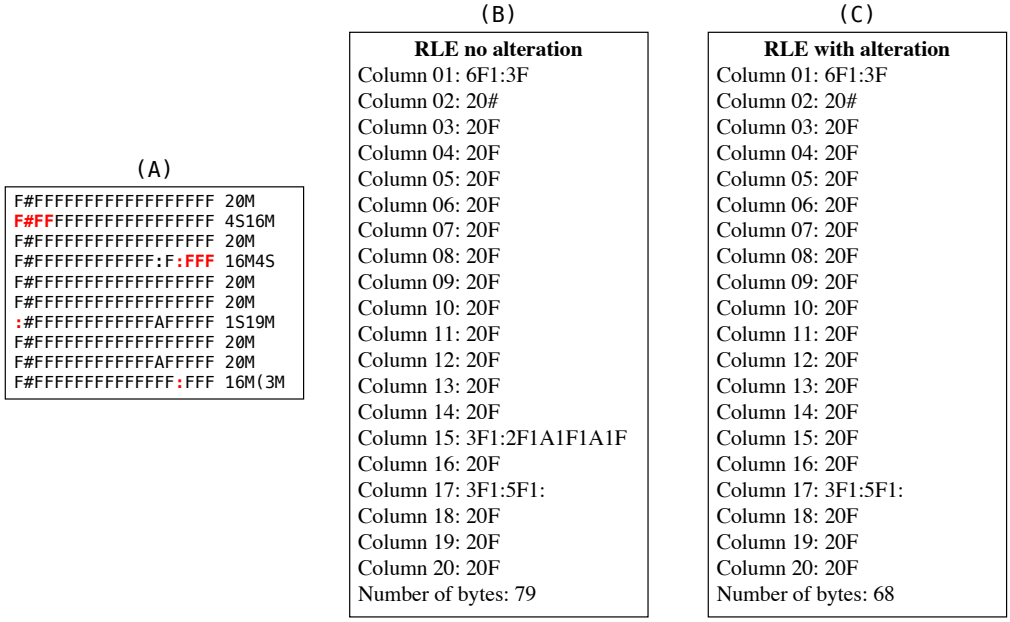

Figure 7. Run Length Encoding of columns in quality scores file (A) Quality scores of first 20 bases of 10 alignments followed by the integrated CIGAR. Regions that are soft-clips or mismatches have been highlighted in Red. (B) Run length encoding of the quality scores without making any alteration. (C) Run length encoding by altering the quality scores of matched nucleotide bases

Organism	Tissue	Layout	Assay Type	Date of publication	Read Length	SRA Accession	SRA Bioproject	Number of reads/pairs (in Millions)	Replicate information
Arabidopsis thaliana	Rosette leaf	PE	RNA-Seq	2021-02-16	151+151	SRR13711353	PRJNA701911	24	C_WT_rep1
Arabidopsis thaliana	Rosette leaf	PE	RNA-Seq	2021-02-16	151+151	SRR13711354	PRJNA701911	26	C_WT_rep2
Arabidopsis thaliana	Rosette leaf	PE	RNA-Seq	2021-02-16	151+151	SRR13711355	PRJNA701911	26	C_WT_rep3
Arabidopsis thaliana	Rosette leaf	PE	RNA-Seq	2021-02-16	151+151	SRR13711356	PRJNA701911	22	C_atcyp18-2_rep1
Arabidopsis thaliana	Rosette leaf	PE	RNA-Seq	2021-02-16	151+151	SRR13711357	PRJNA701911	30	C_atcyp18-2_rep2
Arabidopsis thaliana	Seedlings	PE	DNA-Seq	2020-06-23	150+150	SRR12077404	PRJNA641461	47	reca1 why1 why3 12d seedling 3
Arabidopsis thaliana	Seedlings	PE	DNA-Seq	2020-06-23	150+150	SRR12077405	PRJNA641461	43	reca1 why1 why3 12d seedling 2
Arabidopsis thaliana	Seedlings	PE	DNA-Seq	2020-06-23	150+150	SRR12077406	PRJNA641461	46	reca1 why1 why3 12d seedling 1
Arabidopsis thaliana	Seedlings	PE	DNA-Seq	2020-06-23	150+150	SRR12077407	PRJNA641461	41	Col-0 12d seedling 3
Arabidopsis thaliana	Seedlings	PE	DNA-Seq	2020-06-23	150+150	SRR12077408	PRJNA641461	68	Col-0 12d seedling 2

Table 2. RNA-Seq and DNA-Seq samples for comparison

			Compressed Size (in MB)														
		Numer of reads (in millions)	Parameters Setting 1			Parameters Setting 2			Parameters Setting 3			Parameters Setting 4			Parameters Setting 5		
			BR	7z	ZPAQ	BR	7z	ZPAQ	BR	7z	ZPAQ	BR	7z	ZPAQ	BR	7z	ZPAQ
RNA-Seq	Single ended	24	672	568	492	177	149	132	133	107	93	105	82	73	28	22	20
		50	1377	1165	1003	354	295	262	265	212	185	206	163	146	55	45	39
		76	2066	1751	1504	509	427	379	378	305	265	293	229	206	76	57	53
		98	2686	2278	1957	665	557	496	495	399	347	386	301	272	96	72	68
		128	3495	2966	2544	864	724	644	643	518	452	503	397	356	120	94	86
	Paired ended	24	809	665	598	381	303	286	340	264	250	315	243	234	83	69	73
		50	1663	1366	1229	779	617	583	696	539	511	643	495	477	171	142	149
		76	2484	2043	1834	1130	896	845	1008	781	739	929	712	687	252	203	217
		98	3232	2662	2388	1479	1173	1106	1322	1026	970	1220	934	900	323	259	277
		128	4209	3464	3109	1932	1529	1443	1730	1341	1269	1599	1225	1180	417	334	360
DNA-Seq	Single ended	47	901	736	600	273	216	210	239	185	178	177	134	132	128	96	97
		90	1678	1372	1098	501	394	379	439	340	322	327	251	242	237	181	178
		136	2538	2085	1646	749	599	562	657	519	476	496	380	365	357	271	267
		177	3259	2670	2109	948	750	709	829	647	600	628	484	460	450	344	335
		245	4581	3730	2953	1415	1094	1043	1255	954	895	986	720	708	602	460	447
	Paired ended	47	1030	857	752	472	397	383	391	323	314	326	265	266	250	207	213
		90	1908	1600	1402	868	735	709	726	608	589	609	503	500	465	393	402
		136	2889	2430	2127	1311	1111	1069	1103	925	894	928	769	761	705	598	609
		177	3722	3140	2742	1681	1427	1371	1419	1194	1151	1194	996	979	906	774	781
		245	5236	4373	3847	2441	2026	1970	2088	1709	1672	1774	1438	1438	1237	1060	1072

Table 3. Compression achieved by ABRIDGE with different parameters and with different compressors

			Time (in seconds)														
		Numer of reads (in millions)	Parameters Setting 1			Parameters Setting 2			Parameters Setting 3			Parameters Setting 4			Parameters Setting 5		
			BR	7z	ZPAQ	BR	7z	ZPAQ	BR	7z	ZPAQ	BR	7z	ZPAQ	BR	7z	ZPAQ
RNA-Seq	Single ended	24	879	521	256	334	322	222	312	322	309	257	303	262	177	230	261
		50	1736	817	571	711	549	504	658	551	717	535	500	607	378	466	579
		76	2591	1192	989	1095	814	886	1010	814	1249	794	697	1077	569	639	1020
		98	3770	1426	1309	1470	1049	1208	2077	1338	1589	1359	1455	1375	1358	1414	1313
		128	5956	2417	1566	2818	2019	1390	3113	2194	1673	1978	1554	1393	1730	1695	1309
	Paired ended	24	1328	1063	335	626	404	310	608	439	389	558	421	382	317	405	341
		50	3214	1532	900	1847	1080	798	2053	1272	967	2144	1275	888	1096	1106	797
		76	5046	2409	1902	2963	2184	1855	3334	2056	2133	2679	1982	1999	2329	2075	1827
		98	6755	3023	2680	4183	2419	2372	4388	2804	2199	4108	2608	1993	3017	2415	2566
		128	10048	4504	3267	6783	3879	3879	5913	3031	3317	5914	3898	3049	4146	3704	3779
DNA-Seq	Single ended	47	1492	1212	499	752	868	474	1010	1127	971	829	1052	592	757	745	551
		90	3109	1503	1156	1703	1200	1105	2246	1580	1299	1622	1474	1135	1653	1447	1509
		136	4771	2436	1944	2563	2111	1898	3503	2878	2890	2332	1948	2463	1774	2715	2378
		177	6572	3040	2554	3566	2856	2569	4497	3681	3721	3672	2941	3204	3436	2922	3117
		245	8888	4376	3800	5694	4021	3638	6776	5041	4526	5442	4473	3558	4717	3990	4131
	Paired ended	47	2587	1690	1008	2067	1277	1020	1568	1228	1125	1348	1574	1274	1446	1176	742
		90	4413	2483	1521	3628	2382	1457	3477	2825	2701	2692	1715	2458	2432	2402	2434
		136	6231	3535	2331	4704	3518	3118	5494	4307	2902	5139	3700	2513	4723	3686	2477
		177	8583	4520	4092	6345	4607	4104	6084	5483	3697	5945	4620	3171	5630	4170	4328
		245	11533	5666	6194	10091	7214	6204	7227	8706	7302	7646	5763	6082	6907	6496	4781

Table 4. Duration of compression by ABRIDGE with different parameters and with different compressors

			Time (in seconds)													
		Numer of reads (in millions)	Parameters Setting 1			Parameters Setting 2			Parameters Setting 3			Parameters Setting 4			Parameters Setting 5		
			BR	7Z	ZPAQ	BR	7Z	ZPAQ	BR	7Z	ZPAQ	BR	7Z	ZPAQ	BR	7Z	ZPAQ
RNA-Seq	Single ended	**24**	73	103	1169	81	85	301	57	63	255	58	53	216	45	46	81
		50	150	211	2414	165	175	594	114	124	499	97	110	457	89	92	163
		76	184	271	3600	202	227	864	146	167	718	130	130	643	110	113	233
		98	270	353	4944	290	282	1203	182	226	1023	181	187	843	161	148	297
		128	362	470	6434	352	392	1456	272	293	1314	230	225	1030	207	194	327
	Paired ended	**24**	160	188	1407	249	265	671	236	249	635	220	231	603	195	199	199
		50	364	432	3234	530	403	1623	338	515	1553	457	337	1470	264	411	557
		76	782	629	4867	817	874	2293	730	522	2246	676	712	1792	604	397	824
		98	1004	817	5767	1014	1560	3569	626	991	2544	872	609	2357	775	514	744
		128	912	1076	8257	1984	1456	3910	835	1295	3889	763	826	3645	998	1024	972
DNA-Seq	Single ended	**47**	165	221	2760	204	201	576	135	140	495	115	122	397	124	129	317
		90	319	406	4922	373	394	957	258	277	881	232	238	768	220	233	557
		136	506	615	7545	586	627	1533	386	409	1214	339	363	1055	338	365	889
		177	713	791	10943	724	812	1853	507	556	1560	442	464	1434	419	460	1034
		245	969	1329	12875	1126	1399	2702	680	746	2432	728	632	2030	577	710	1476
	Paired ended	**47**	512	747	3541	523	543	1253	472	677	1131	429	437	814	418	427	661
		90	1307	1267	6091	1288	1283	1894	1249	1050	2114	939	1201	1942	1157	875	1273
		136	2027	1676	8383	1620	1693	3536	1381	1454	2507	1275	1340	2928	1258	1267	1861
		177	1987	2189	10548	2686	2727	4519	2425	2479	3225	2279	1678	3758	1590	1638	3157
		245	2716	3753	14544	2965	3178	5198	2477	2519	4636	2335	2399	4107	2319	2279	4234

Table 5. Duration of decompression by ABRIDGE with different parameters and with different compressors

			Duration (in seconds)				Memory (in MB)		
		Numer of reads (in millions)	ABRIDGE	BAM	CRAM		ABRIDGE	BAM	CRAM
RNA-Seq	Single ended	**24**	9	11	2		44.44	3.5	2.82
		50	9	12	1		36.14	6.05	2.84
		76	18	22	1		58.65	9.28	2.62
		98	24	32	3		86.17	14.11	2.77
		128	31	40	3		101.09	15.4	2.79
	Paired ended	**24**	48	53	6		125.96	18.62	2.85
		50	23	25	2		65.47	4.6	2.64
		76	30	37	4		89.61	5.02	2.95
		98	35	48	4		109.32	7.37	3
		128	53	58	4		126.96	8.1	3.01
DNA-Seq	Single ended	**47**	44	22	2		48.04	2.91	2.66
		90	46	21	1		42.86	2.93	2.88
		136	59	40	2		64.24	4.85	2.96
		177	118	60	2		86.12	2.89	3.07
		245	125	75	5		105.39	3.23	3.14
	Paired ended	**47**	150	105	5		132.4	3.14	3.26
		90	89	39	2		77.77	3.09	2.84
		136	110	64	4		114.85	3.02	2.85
		177	139	81	5		144.95	3.01	3
		245	198	109	6		195.78	3.05	2.99

Table 6. Duration and memory consumption of ABRIDGE to create index for random search

	Parameter setting 1	Parameter setting 2	Parameter setting 3	Parameter setting 4	Parameter setting 5
Save Exact Quality scores	**Yes**	No	No	No	No
Save Quality scores	**Yes**	**Yes**	**No**	No	No
Save Soft Clippings	**Yes**	**Yes**	**Yes**	No	No
Save mismatches	**Yes**	**Yes**	**Yes**	No	No
Save unmapped reads	**Yes**	**Yes**	**Yes**	**Yes**	No

Table 7. Illustration of arguments provided to ABRIDGE for each parameter setting

			Memory (in GB)					
		Numer of reads (in millions)	**ABRIDGE**	**CSAM**	**DEEZ**	**GENOZIP**	**BAM**	**CRAM**
RNA-Seq	**Single ended**	**24**	10.6035538	0.01319885	3.03070831	0.12991714	0.00118256	0.00125885
		50	12.1946945	0.01726913	3.05444717	0.70067596	0.00118256	0.00125885
		76	12.9527283	0.01953888	3.0282135	0.74595261	0.00118256	0.00125885
		98	13.7239113	0.02521133	3.03667068	0.75827408	0.00118256	0.00125885
		128	14.7027435	0.02589035	3.07320023	0.79193115	0.00118256	0.00125885
	Paired ended	**24**	10.4386711	0.01242828	2.9929924	0.13118362	0.00118256	0.00125885
		50	10.4650269	0.01849747	2.99604034	0.17809296	0.00118256	0.00125885
		76	10.4858246	0.01922226	3.02975845	0.21195984	0.00118256	0.00125885
		98	10.5088577	0.02093506	3.02715302	0.25325394	0.00118256	0.00125885
		128	10.5120354	0.02730179	3.04006195	0.29411316	0.00118256	0.00125885
DNA-Seq	**Single ended**	**47**	10.5367737	0.03121567	3.00671387	0.28727341	0.00118256	0.00125885
		90	10.6379623	0.04970932	2.98199844	1.22003937	0.00118256	0.00125885
		136	11.5782166	0.05479431	3.00313568	1.34737396	0.00118256	0.00125885
		177	11.9462471	0.08978271	3.02417374	1.38080978	0.00118256	0.00125885
		245	12.5375938	0.09642029	3.02261353	1.54094696	0.00118256	0.00125885
	Paired ended	**47**	10.5375633	0.03024292	2.9940834	0.26768112	0.00118256	0.00125885
		90	10.636692	0.05019379	3.00325394	0.47337723	0.00118256	0.00125885
		136	11.5744591	0.05505371	3.00336456	0.67036819	0.00118256	0.00125885
		177	11.9455643	0.08964157	3.02405548	0.8114357	0.00118256	0.00125885
		245	12.5394821	0.09854507	3.02558899	1.05052567	0.00118256	0.00125885

Table 8. Memory consumed by different software while accessing a random location

			Time of access (in seconds)					
		Numer of reads (in millions)	**ABRIDGE**	**CSAM**	**DEEZ**	**GENOZIP**	**BAM**	**CRAM**
RNA-Seq	**Single ended**	**24**	48	1	435	3	0	0
		50	91	1	920	7	0	0
		76	131	3	1421	9	0	0
		98	168	2	1884	9	0	0
		128	215	3	2476	10	0	0
	Paired ended	**24**	54	2	432	2	0	0
		50	100	4	942	3	0	0
		76	145	3	1438	4	0	0
		98	186	3	1819	4	0	0
		128	240	7	2375	5	0	0
DNA-Seq	**Single ended**	**47**	69	2	1734	5	0	0
		90	120	1	2836	35	0	0
		136	179	7	3840	36	0	0
		177	228	25	4695	41	0	0
		245	312	3	5964	42	0	0
	Paired ended	**47**	77	2	1688	3	0	0
		90	135	37	2706	9	0	0
		136	203	6	3699	14	0	0
		177	262	2	4459	16	0	0
		245	357	7	5846	22	0	0

Table 9. Duration of different software to access a random location