GigaScience

Twelve years of SAMtools and BCFtools

--Manuscript Draft--

Manuscript Number:	GIGA-D-20-00369
Full Title:	Twelve years of SAMtools and BCFtools
Article Type:	Technical Note
Funding Information:	Wellcome Trust (206194) Not applicable
Abstract:	Abstract Background SAMtools and BCFtools are widely used programs for processing and analysing high-throughput sequencing data. Findings The first version appeared online twelve years ago and has been maintained and further developed ever since, with many new features and improvements added over the years. The SAMtools and BCFtools packages represent a unique collection of tools that have been used in numerous other software projects and countless genomic pipelines. Conclusion Both SAMtools and BCFtools are freely available on GitHub under the permissive MIT licence, free for both non-commercial and commercial use. Both packages have been installed over a million times via Bioconda. The source code and documentation are available from http://www.htslib.org .
Corresponding Author:	Andrew Whitwham Wellcome Sanger Institute Cambridge, UNITED KINGDOM
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Wellcome Sanger Institute
Corresponding Author's Secondary Institution:	
First Author:	Petr Danecek
First Author Secondary Information:	
Order of Authors:	Petr Danecek
	James Bonfield
	Jennifer Liddle
	John Marshall
	Valeriu Ohan
	Martin Pollard
	Andrew Whitwham
	Thomas Keane
	Shane McCarthy
	Robert Davies
	Heng Li
Order of Authors Secondary Information:	
Additional Information:	
Question	**Response**

Are you submitting this manuscript to a special series or article collection?	No
Experimental design and statistics Full details of the experimental design and statistical methods used should be given in the Methods section, as detailed in our Minimum Standards Reporting Checklist. Information essential to interpreting the data presented should be made available in the figure legends. Have you included all the information requested in your manuscript?	Yes
Resources A description of all resources used, including antibodies, cell lines, animals and software tools, with enough information to allow them to be uniquely identified, should be included in the Methods section. Authors are strongly encouraged to cite Research Resource Identifiers (RRIDs) for antibodies, model organisms and tools, where possible. Have you included the information requested as detailed in our Minimum Standards Reporting Checklist?	Yes
Availability of data and materials All datasets and code on which the conclusions of the paper rely must be either included in your submission or deposited in publicly available repositories (where available and ethically appropriate), referencing such data using a unique identifier in the references and in the "Availability of Data and Materials" section of your manuscript. Have you have met the above requirement as detailed in our Minimum	Yes

Standards Reporting Checklist?	

Twelve years of SAMtools and BCFtools

Authors: Petr Danecek, James K. Bonfield, Jennifer Liddle, John Marshall, Valeriu Ohan, Martin O Pollard, Andrew Whitwham, Thomas Keane, Shane A. McCarthy, Robert M. Davies, Heng Li

Abstract

Background

SAMtools and BCFtools are widely used programs for processing and analysing high-throughput sequencing data.

Findings

The first version appeared online twelve years ago and has been maintained and further developed ever since, with many new features and improvements added over the years. The SAMtools and BCFtools packages represent a unique collection of tools that have been used in numerous other software projects and countless genomic pipelines.

Conclusion

Both SAMtools and BCFtools are freely available on GitHub under the permissive MIT licence, free for both non-commercial and commercial use. Both packages have been installed over a million times via Bioconda. The source code and documentation are available from http://www.htslib.org.

Background

With the advancement of genome sequencing technologies and large scale sequencing projects, new data formats became necessary for interoperability, compact storage and efficient analysis of the data. Among the most common formats used in this field today are SAM [1] and VCF [2] developed by the 1000 Genomes Project [3]. These specialized formats for storing read alignments (SAM) and genetic variants (VCF) are row-oriented tab-delimited text files, which are easy to process using custom scripts but slow to parse and can be inefficient to store. Therefore in practice, the binary counterparts BAM or CRAM are used for alignment data and, when parsing of large VCF files becomes prohibitively slow, BCF provides a more efficient format for processing variation data.

Despite the conceptual simplicity of the underlying DNA sequence, the alignment and variant data carry rich information. This data undergoes a number of processing steps, many of which are algorithmically complex and require specialized software. Also more programming effort and expertise are necessary to parse binary formats. Therefore programs and toolkits that encompass functionality for the most common tasks have been developed. These include tools for file manipulation, quality control, and data analyses, such as sambamba[4], biobambam[5], FastQC[6], and GATK[7]. A successful bioinformatics tool must keep up with advancements in sequencing technologies (e.g. substantial increase in sequencing read length), scale well with ever increasing amounts of data (from single to hundreds of thousands of genomes), expanding focus to encompass new analyses and more complex types of variation. New species also bring challenges such as large chromosomes not representable by 32 bits (over 2 Gbases) or assumptions about the ploidy of an organism. In this article we describe the status, new features and developments in SAMtools and BCFtools.

SAMtools was originally published in 2009[1]. Readers of the online edition of that paper would have been able to download the release 0.1.4. The package included not only utilities to convert and manipulate SAM and BAM files, but also a variant caller, which was soon restructured into the BCFtools subpackage (2010, release 0.1.9). Later it became apparent that third-party projects were trying to use code from SAMtools despite it not being designed to be embedded in that way. Therefore the decision was taken in August 2014 (release 1.0) to split the SAMtools package into a standalone library with a well-defined API (HTSlib, currently 82k lines of code), a project for variant calling and manipulation of variant data (BCFtools, 71k lines), and SAMtools for working with alignment data (42k lines). All three projects are maintained in parallel, and improvements to HTSlib naturally filter into new releases of SAMtools and BCFtools. Since the original release the combined size of HTSlib, SAMtools and BCFtools has doubled.

Findings

SAMtools

Since the initial release there have been over 2,200 commits to the code repository and 52 releases, the most recent being version 1.11 in September 2020.

The main part of the SAMtools package is a single executable that offers various commands for working on alignment data. The *view* command performs format conversion, file filtering, and extraction of sequence ranges. Files can be reordered, joined and split in various ways using the commands *sort*, *collate*, *merge*, *cat* and *split*. Files can be indexed for fast random access using *index* for alignment files and *faidx* for reference sequences in the FASTA format. File content can be manipulated with commands like *addreplacerg*, *calmd*, *fixmate*, and *reheader*. Duplicated reads, caused by artifacts in the library creation and sequencing process can be flagged using *markdup*. Various statistics on alignment files can be calculated using *idxstats*, *flagstat*, *stats*, *depth* and *bedcov*. Data can be converted to legacy formats using *fasta* and *fastq*. For position-ordered files, the sequence alignment can be viewed using *tview* or output via *mpileup* in a way that can be used for ongoing processing (for example, variant calling). Most recently SAMtools has gained support for amplicon based sequencing projects via *ampliconclip* and *ampliconstats*. Full documentation for these commands is included with the package in the form of UNIX man pages, and can also be found online at https://www.htslib.org/doc/.

Early releases of SAMtools could read and write alignment data in the SAM and BAM formats. The 1.0 release introduced support for the better-compressed CRAM format [8]. Originally, the program required the use of command-line options to select the input format, and most commands were tied to using BAM files. These restrictions were removed as SAMtools transitioned to use HTSlib, so by release 1.0 most commands could automatically detect the input file format and could directly read and write SAM, BAM and CRAM files. In particular, there is rarely any need to convert SAM to BAM using “*samtools view -b*” before running commands like “*samtools sort*”, although regrettably this idiom still appears in a large number of online tutorials.

SAMtools has also become faster, most notably by gaining the ability to use threads to take better advantage of the parallelism available on modern multicore systems. Thread support first arrived in version 0.1.19 (March 2013) which enabled them for sorting and BAM file writing in the *view* command. The number of tasks using threads has slowly increased, so now (thanks to improvements in HTSlib) it is possible to use them for both reading and writing SAM, BAM and CRAM formats in most of the commands. Another time saving improvement is the ability to index files as they are written (added in 1.10). This allows pipelines that need to index files to remove the separate "*samtools index*" stage and associated read-through of the file being indexed.

BCFtools

The original purpose of the BCFtools package was to divide the I/O and CPU intensive tasks of variant calling into separate steps. In the first, "*samtools mpileup*" would read the alignments and for each position of the genome construct a vertical slice across all reads covering the position ("pileup").

Genotype likelihoods are then calculated, representing how consistent is the observed data with the possible diploid genotypes. The calculation takes into account mapping qualities of the reads, base qualities and the probability of local misalignment, per-base alignment quality (BAQ)[9]. The second step, "*bcftools call*" (known in the initial release as "*bcftools view*"), then evaluates the most likely genotype under the assumption of Hardy-Weinberg equilibrium (in the sample context customizable by the user) using allele frequencies estimated from the data or provided explicitly by the user. In 2016 (release 1.4) genotype likelihood generation was moved from SAMtools to BCFtools to make both variant calling steps part of the same package and to prevent errors arising from the possible use of incompatible versions of the two programs.

Today BCFtools is a full featured program which consists of 21 commands and 38 plugins (single-purpose tools) with more than 230 documented command line switches and options. As of writing, there have been more than 2,300 commits and 29 releases since 2012, with the most recent 1.11 released in September 2020.

The "*bcftools view*" command provides conversion between the text VCF and the binary BCF format, where both formats can be either plain (uncompressed) or block-compressed with BGZF for random access and compact size. The plain text VCF output is useful for visual inspection, for processing with custom scripts, and as a data exchange format. It should not be used when performance is critical, because BCFtools internally uses the binary BCF representation and the conversion between the text VCF format and the binary BCF format is costly. Also compression and decompression is CPU intensive and therefore when streaming between multiple commands in a pipeline it is recommended to stream uncompressed BCF by appending the option "*-Ou*".

The program can do much more than convert between VCF and BCF formats. It can also process third-party formats (using the *convert* command) and manipulate variant files in many ways. It can be used to *index*, *sort* and normalize variants (*norm*), replace headers (*reheader*), add and remove annotations (*annotate*), and subset samples (*view*). Most commands can filter sites either by a region, list of sites, or a general boolean expression involving any combination of VCF tags (*--include*, *--exclude*). Multiple files can be compared, splitting common and file-specific variants into separate files according to custom rules (*isec*). Files sorted by position can also be combined using the *merge* command (input files have different samples) or *concat* command (input files have the same samples). Arbitrary fields can be extracted and formatted into a custom text output (*query*), a feature which, among other things, is useful for scripting.

Apart from file manipulation, the program offers variant callers and algorithms useful for analysis. For calling SNPs and short indels from read alignment files, BCFtools implements two variant calling models. In addition to the original biallelic caller ("*bcftools call -c*"; [10]) there is a newer model available, capable of handling positions with multiple alternate alleles ("*bcftools call -m*") and supporting gVCF output[11]. The package implements an HMM caller for detection of runs of homozygosity (*roh*; [12]), copy-number variation calling from SNP array data (*cnv*; [13]), and the detection of whole chromosome aberrations (*polysomy*). The program can construct a consensus sequence given a FASTA and a variant file (*consensus*), perform sample identity checks (*gtcheck*) and collect various statistics (*stats*).

In addition to built-in commands, the program supports a dynamic plugin mechanism for specific single-purpose tasks with a diverse range of functions. Examples from a large and ever growing collection include: the plugin *split-vep* for convenient querying and extraction of VEP annotations [14]; *trio-dnm* for ascertainment of *de novo* variants and their parental origin (*parental-origin*), or for collection of statistics (*trio-stats*) in trio data; gVCF manipulation (*gvcfz*); and many more.

Discussion

SAMtools and BCFtools represent a unique collection of tools useful for processing and analysis of sequencing data. Their development has been driven by the need of both large projects and individual user requests issued via GitHub. The code has been installed over a million times via Bioconda[15] and GitHub releases, and more than 900 support and feature requests were resolved on GitHub.

The programs are written in the C programming language and optimized for low memory consumption and high speed. For example, the "*bcftools csq*" command for prediction of functional consequences in a haplotype-aware manner requires only a fraction of the memory required by VEP and is two orders of magnitude faster [16].

Much work has been done to increase the reliability of SAMtools and BCFtools. The test harnesses now include ~700 tests in SAMtools and ~1400 in BCFtools. Continuous integration services run all of the tests on a variety of platforms (including Linux, MacOS and Windows) whenever code is checked into the source repository, ensuring bugs are discovered and fixed rapidly. Code quality is also assured by checking for memory errors, originally using Valgrind memcheck [17] and more recently with AddressSanitizer [18]. Additionally, UndefinedBehaviourSanitizer is used to detect violations of the C standard.

Considering the ever growing sample sizes and rapid increases in the amounts of sequenced data, the programs withstood the test of the time thus far. However, extremely big files are produced by large projects and their processing requires a high degree of parallelization on computing clusters. Future versions of SAMtools and BCFtools are expected to make more use of threaded code to allow faster processing of such files. Sometimes even the limits of BCF representation itself can be reached. For example, highly polymorphic sites can contain dozens of alternate indel alleles which, in files with tens of thousands of samples, exceed the internal limit of 4GB per site due to quadratic scaling of annotations such as FORMAT/PL. An extension of the VCF specification has been proposed to address this problem by introducing a localized version of such annotations with linear scaling[19] and has been implemented in BCFtools.

The programs have been used to process and analyze sequencing data from all types of species, vertebrate, non-vertebrates, pathogens, plants and viruses. This provides interesting challenges and opportunities for future developments. For example, some of the BCFtools commands are limited to handling haploid and diploid organisms and the support for large "64-bit" genomes is currently only partial. More work is also planned to overcome difficulties stemming from ambiguities in VCF allele encoding (such as operations of atomization and deatomization), to improve visualization of results, and there are at least 50 feature requests currently registered on GitHub to investigate.

Availability of supporting source code and requirements

Project name: SAMtools

Project home page: https://www.htslib.org, https://github.com/samtools/samtools

Operating system(s): Platform independent

Programming language: C

License: MIT/Expat

RRID: SCR_002105

Project name: BCFtools

Project home page: https://www.htslib.org, https://github.com/samtools/bcftools

Operating system(s): Platform independent

Programming language: C

Other requirements: Optional use of GNU Scientific Library (GSL)

License: MIT/Expat

RRID: SCR_002105

Authors' contributions

J.B., P.D., R.D., H.L., J.M., M.P., V.O., and A.W. wrote the SAMtools software with J.L. and S.M. supporting; R.D., T.K., and J.M. provided supervision.

P.D. and S.M. wrote the BCFtools software with J.B., R.D., H.L., J.M. and V.O. supporting; S.M. also provided supervision.

J.B., P.D., R.D., V.O., and A.W. wrote the original draft of the manuscript with all authors reviewing.

Funding

This work was supported by the Wellcome Trust grant [206194].

References

1. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25: 2078–2079.

2. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, et al. The variant call format and VCFtools. Bioinformatics. 2011;27: 2156–2158.

3. The 1000 Genomes Project Consortium. A global reference for human genetic variation. Nature. 2015;526: 68–74.

4. Tarasov A, Vilella AJ, Cuppen E, Nijman IJ, Prins P. Sambamba: fast processing of NGS alignment formats. Bioinformatics. 2015;31: 2032–2034.

5. Tischler G, Leonard S. biobambam: tools for read pair collation based algorithms on BAM files. Source Code Biol Med. 2014;9: 2078.

6. Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data. [cited 29 Jun 2020]. Available: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

7. DePristo MA, Banks E, Poplin R, Garimella KV, Maguire JR, Hartl C, et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat Genet. 2011;43: 491–498.

8. Hsi-Yang Fritz M, Leinonen R, Cochrane G, Birney E. Efficient storage of high throughput DNA sequencing data using reference-based compression. Genome Res. 2011;21: 734–740.

9. Li H. Improving SNP discovery by base alignment quality. Bioinformatics. 2011;27: 1157–1158.

10. Li H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. Bioinformatics. 2011;27: 2987–2993.

11. gvcftools. [cited 6 Nov 2020]. Available: https://sites.google.com/site/gvcftools

12. Narasimhan V, Danecek P, Scally A, Xue Y, Tyler-Smith C, Durbin R. BCFtools/RoH: a hidden Markov model approach for detecting autozygosity from next-generation sequencing data. Bioinformatics. 2016;32: 1749–1751.

13. Danecek P, McCarthy SA, HipSci Consortium, Durbin R. A Method for Checking Genomic Integrity in Cultured Cell Lines from SNP Genotyping Data. Chan KYK, editor. 2016. doi:10.1371/journal.pone.0155014

14. McLaren W, Gil L, Hunt SE, Riat HS, Ritchie GRS, Thormann A, et al. The Ensembl Variant Effect Predictor. Genome Biology. 2016. doi:10.1186/s13059-016-0974-4

15. Grüning B, The Bioconda Team, Dale R, Sjödin A, Chapman BA, Rowe J, et al. Bioconda: sustainable and comprehensive software distribution for the life sciences. Nature Methods. 2018. pp. 475–476. doi:10.1038/s41592-018-0046-7

16. Danecek P, McCarthy SA. BCFtools/csq: haplotype-aware variant consequences. Birol I, editor. Bioinformatics. 2017;33: 2037–2039.

17. Nethercote N, Seward J. Valgrind: a framework for heavyweight dynamic binary instrumentation. SIGPLAN Not. 2007;42: 89–100.

18. Serebryany K, Bruening D, Potapenko A, Vyukov D. AddressSanitizer: A Fast Address Sanity Checker. Proceedings of the 2012 USENIX conference on Annual Technical Conference. 2012. Available: https://www.usenix.org/system/files/conference/atc12/atc12-final39.pdf

19. samtools. Define Local Alleles in VCF to allow for sparser format by yfarjoun · Pull Request #434 · samtools/hts-specs. In: GitHub [Internet]. [cited 10 Jul 2020]. Available: https://github.com/samtools/hts-specs/pull/434