Future development of the Samtools software package

John Marshall, Petr Daněček, James K. Bonfield, Robert M. Davies, Martin O. Pollard, Shane A. McCarthy, Thomas M. Keane, Heng Li, Richard Durbin Vertebrate Resequencing, Wellcome Trust Sanger Institute, Hinxton, United Kingdom

Samtools is one of the most widely-used software packages for analysing next-generation sequencing data. Since the release of version 0.1.19 in March 2013, there have been over forty thousand downloads of Samtools. Samtools and its associated sub-tools are used throughout many NGS pipelines for data processing tasks such as creating, converting, indexing, viewing, sorting, and merging SAM, BAM, VCF, and BCF files. Bcftools, also part of the package, is used for SNP and indel calling and genotyping, producing VCF- or BCF-formatted output.

Recently, the primary responsibility for the development of the software transitioned from the founder of the project, Heng Li, to a team of developers at the Wellcome Trust Sanger Institute. The initial highest priority task has been the reorganisation of the samtools implementation to integrate the more recent HTSlib project as the core library at the heart of samtools, as shown at right. This separation facilitates the early release of improvements in variant calling or file manipulation, as bcftools and samtools releases can eventually be made independently.

Beyond this reorganisation, major new functionality includes support for the new sequence data format CRAM, which offers significantly better compression than BAM, and a new multi-allelic variant calling model.

We aim to make Samtools releases more regularly in future, and welcome bug reports on the samtools mailing lists and via the issue trackers at GitHub (see bottom right for links).

HTSLIB

C library for handling high-throughput sequencing data, providing APIs for manipulating SAM, BAM, and CRAM sequence files (similar to but more flexible than the old Samtools API) and for manipulating VCF or BCF variant files. Used by samtools and bcftools and also by other programmers in their C programs.

SAMTOOLS

Tools for manipulating SAM, BAM, and CRAM alignment files, and for producing pileups and BCF from them, notably:

view	Display sequence data or convert between formats
tview	Simple terminal-based interactive alignment viewer
faidx	Display regions from FASTA files
index	Generate index file for region-based access
sort	Sort an alignment file, by read name or mapped position
merge	Merge several sorted alignment files
cat	Quickly concatenate alignment files
flagstat	Display simple mapping statistics
idxstats	Display basic statistics stored in index file
stats	Calculate statistics (previously called *bamcheck*)
rmdup	Remove PCR duplicates
calmd	Recalculate SAM MD/NM tags and "=" bases
fixmate	Fix mate information in alignment records
reheader	Replace headers
bamshuf	Shuffle and group alignments by name
mpileup	Generate pileups over multiple alignment files
phase	Phase heterozygotes
depth	Compute read depth within specified regions

BCFTOOLS

Tools for manipulating VCF and BCF files, and for variant calling, notably:

view	Display variant data or convert between formats
index	Generate index file enabling rapid position-based access
query	Display variants in user-defined formats
stats	Calculate variant statistics (previously called *vcfcheck*)
gtcheck	Detect swaps and contaminations
isec	Find common events across several files
merge	Merge several files into one
subset	Filter variants by user-defined rules
filter	Filter using fixed thresholds
som	Filter using Self-Organized Maps
call	SNP/indel calling (previously part of *view*)
norm	Normalize indels

Samtools has historically provided both command-line tools for SAM/BAM manipulation and variant calling and a SAM/BAM API for use by third-party C programs. The package has now been split into three coordinated projects focusing on these three separate areas.

CRAM & reference-based compression

CRAM is a format developed by the European Bioinformatics Institute to exploit higher compression gained by reordering data and by the use of a known reference sequence.

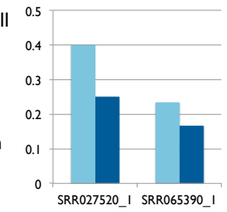
Sequence names look like other names and quality values look like other quality values, so compress well individually; however mixing the two data types leads to more disparity and less compression. Grouping data acording to their types before compressing typically yields a 10–20% space saving.

Reference sequence compression potentially saves almost all storage occupied by the base calls by recording only the locations and calls of bases that disagree with the reference sequence.

The combined impact of these changes will depend on data quality, but typically yields a 30–40% reduction in file size compared to BAM while only having a small penalty in CPU time. Once more aggressively-quantised quality values become the norm the benefits of reference compression become even more significant.

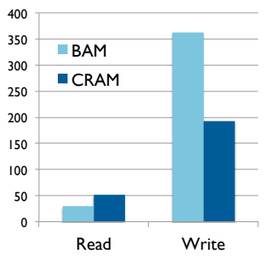
Efficient storage of high throughput DNA sequencing data using reference-based compression
Markus Hsi-Yang Fritz *et al*, Genome Res. (2011)

http://www.ebi.ac.uk/ena/about/cram_toolkit/
https://github.com/enasequence/cramtools



BAM (left) and CRAM file sizes as a proportion of raw FASTQ size for two samples.*

* From *Compression of FASTQ and SAM format sequencing data*, Bonfield & Mahoney, PLoS One (2013)



HTSlib CRAM performance

Rough timing of reading and writing BAM (left) versus CRAM. Reading is seconds elapsed to count each alignment in equivalent 2 GB BAM and 1.2 GB CRAM files. Writing is seconds elapsed to convert the 2 GB BAM file to BAM or CRAM.

Variant-calling file formats: VCF & BCF

At their May 2012 meeting, the 1000 Genomes Project introduced BCFv2, a binary version of the well-established Variant Call Format, VCF. Samtools and bcftools work with both VCF and BCF, either compressed or uncompressed.

BCF files are compressed in the same way as BAM files, but are not substantially smaller than simply gzipping the corresponding text VCF files. Instead the aim of this format is speed: the underlying binary representation was designed to be much faster to parse than VCF text, while the BGZF compression maintains random access.

The older BCFv1 format produced by previous versions of samtools is no longer supported, but can be converted to VCF by using a previous version of bcftools view.

The first reorganised samtools/bcftools/htslib release is imminent. In addition to the major new functionalities discussed and along with numerous bug fixes, miscellaneous smaller improvements include:

- New C implementation of VCF and BCF manipulation enables tools operating much faster than equivalents written in scripting languages.
- The new CSI index format is a generalisation of the BAI format, allowing reference chromosomes as long as the BAM format allows (2^{31}-1) rather than being limited to 2^{29}-1 base pairs. BAI indices remain supported.
- Automatic format detection for input files, including when reading from standard input in pipelines. Tools accept SAM/BAM/CRAM or VCF/BCF without needing explicit format-selection options.

SAM, BAM, VCF, & BCF format specifications

Not part of Samtools itself, the *https://github.com/samtools/hts-specs* repository collects together the specifications of these related NGS data formats, and also of the formats used for indexing them. These documents describe the various data models, i.e., what kind of information can be stored, as well as the low-level details of each format's representation. It is hoped that collecting them together like this will make it easier to refer to and improve these useful documents.

Improvements currently in progress or planned include:

- Improved robustness, ensuring failures due to I/O errors or corrupted data are reported to the user.
- Installation scripts for the three projects—particularly htslib, enabling third-party C programs to find its headers and library easily.
- Improved facilities for manipulating SAM-style headers.
- More flexible duplicate removal.
- In the longer term, samtools and likely the SAM data model itself may require adaptations to the longer reads and different error models of upcoming sequencing technologies.

The web site *http://samtools.sourceforge.net/* is the main location for downloads, documentation, and links to mailing lists. Source code and format specifications are available from repositories within the samtools GitHub organisation:

https://github.com/samtools/bcftools *https://github.com/samtools/htslib*
https://github.com/samtools/samtools *https://github.com/samtools/hts-specs*