SAM/BAM format v1.5 extensions for *de novo* assemblies

Peter J. A. Cock[1]*, James Bonfield[2], Bastien Chevreux[3], and Heng Li[4]

[1] James Hutton Institute, Invergowrie, Dundee, DD2 5DA, UK, [2] Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Cambridge, CB10 1SA, UK, [3] Department for Human Nutrition and Health, DSM Nutritional Products Ltd., Kaiseraugst CH-4303, Switzerland, [4] Broad Institute, 5 Cambridge Center, Cambridge, MA 02142, USA

ABSTRACT

Summary: The plain text Sequence Alignment/Map (SAM) file format and its companion binary form (BAM) are a generic alignment format for storing read alignments against reference sequences (and unmapped reads) together with structured meta-data (Li *et al.*, 2009). Driven by the needs of the 1000 Genomes Project which sequenced many individual human genomes, early SAM/BAM usage focused on pairwise alignments of reads to a reference. However, through the CIGAR P operator multiple sequence alignments can also be preserved. Herein we describe clarifications and additions in version 1.5 of the specification to facilitate storing *de novo* sequence alignments: Padded reference sequences (with gap characters), annotation of reads or regions of the reference, and the option of embedding the reference sequence within the file.

Availability: The latest public release of the specification is at http://samtools.sourceforge.net/SAM1.pdf, with in development drafts at https://github.com/samtools/hts-specs/ under version control.

Contact: peter.cock@hutton.ac.uk

1 INTRODUCTION

The era of 'second generation sequencing' has been one of great change, not just in terms of biological insights and novel algorithms, but also at the practical level of sequence file formats. Here the increasing volumes of high throughput sequencing (HTS) data have pushed early plain text based files formats to their limits, driving the need for efficient alternatives.

In just a few years, SAM/BAM became the *de facto* standard for aligning high throughput sequencing (HTS) reads to reference sequences. It is the default output for tools like BWA (Li and Durbin, 2009) and Bowtie2 (Langmead and Salzberg, 2012), and it is difficult to imagine any new mapping software not offering this as an output format since to do so would handicap downstream analysis and so deter users. This dominance has come about though a combination of factors both social and technical – it would not have been possible without the highly scalable design with BAM random access indexing. This has been greatly helped by two open source reference library implementations, Picard for Java, and the samtools C API (Li *et al.*, 2009). Bindings for the later have been written for major higher level programming languages including Python, Perl, Ruby and R. Additionally the SAM/BAM ecosystem has spurred innovation in visualisation software, with tools like IGV (Robinson

*to whom correspondence should be addressed

et al., 2011) and Tablet (Milne *et al.*, 2013) being popular examples, both written in Java using Picard. As a consequence of this, the best way for any potential successor format such as CRAM (Cochrane *et al.*, 2013) to be adopted will be seamless interoperability with files produced by the samtools and Picard libraries.

For HTS *de novo* assemblies, there has been little agreement to date on file formats beyond simple FASTA for contig sequences, and sometimes FASTQ (Cock *et al.*, 2010) if consensus quality scores are calculated. Partly this reflects the great diversity of *de novo* assembly algorithms, and the resulting variation in the kind and detail of information available. Specifically, de Bruijn graph based assemblers such as Velvet (Zerbino and Birney, 2008) do not normally track individual reads, so it does not make sense to ask for an assembly file showing where each read was placed. One workaround is to perform the assembly and then (re)map the reads to it, giving a SAM/BAM mapping file with useful information about read coverage variation and so on. The idea that SAM/BAM could be extended to store a graph based representation of an assembly was mooted, preserving the linear representation of contigs with reads mapped to them, but annotating how they are interconnected through scaffolding or assembly ambiguities. This has not been pursued as yet. However the participants in the Assemblathon competitions (Earl *et al.*, 2011; Bradnam *et al.*, 2013) have defined a FASTA-like graph format, FASTG (http://fastg.sourceforge.net/). Because this is designed to reduce to a set of linear contigs, perhaps SAM/BAM with reference sequences in FASTG format will become a popular combination?

For more traditional overlap-consensus based assemblers (which may come back into the fore as HTS read lengths are expected to increase), the ACE and CAF file formats have been widely used, for example by PHRAP and CONSED (Gordon *et al.*, 1998), CAP3 (Huang and Madan, 1999), the Roche 'Newbler' gsAssember (454 Life Sciences, Roche Applied Science, Branford, CT) and MIRA (Chevreux *et al.*, 1999). These file formats do not scale well with current data volumes, but were still used in part because of features not previously possible in SAM/BAM, which have been addressed with the extensions herein.

1.1 Padded reference

The SAM/BAM format has traditionally been used with an unpadded reference sequence (no gap characters), requiring the use of the CIGAR (compact idiosyncratic gapped alignment report) pad (P) operator to capture inter-sequence alignment information implicitly. The Gap5 assembly viewer and editor (Bonfield and

Table 1. An example gapped alignment

	1	…	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	…	100	Padded reference POS	Padded reference CIGAR	Unpadded reference POS	Unpadded reference CIGAR
Padded	1	…	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	…	100				
Unpadded	1	…	37	38	39	40	41	42	43	44	45	46	*NA*	*NA*	*NA*	*NA*	*NA*	47	48	49	50	51	52	53	54	55	56	…	95				
Reference	A	…	C	C	T	T	T	G	T	T	C	A	-	-	-	-	-	A	C	C	C	C	C	T	T	G	C	…	A	1	46M5D49M	1	95M
Read01			C	C	T	T	T	G	T	T	C	A																		37	10M	37	10M
Read02				C	T	T	T	G	T	T	C	A	C																	38	10M	38	9M1I
Read03					T	T	T	G	T	A	C	A	C	A																39	10M	39	8M2I
Read04						T	T	G	T	T	C	A	C	A	T															40	10M	40	7M3I
Read05						T	T	G	T	T	C	A	-	A	T	A														40	7M1D3M	40	7M1P3I
Read06							T	G	T	T	C	A	C	A	T	A														41	10M	41	6M4I
Read07							T	G	T	T	C	A	C	A	-	A	C													41	8M1D2M	41	6M2I1P2I
Read08								G	T	T	C	-	C	A	T	A	C													42	4M1D5M	42	4M1D5I
Read09									T	T	C	A	C	A	-	A	C	A	C											43	6M1D4M	43	4M2I1P2I2M
Read10										T	C	A	C	A	T	A	C	A	C											44	10M	44	3M5I2M
Read11											C	A	C	A	T	A	-	A	C	C	C									45	6M1D4M	45	2M4I1P4M
Read12											A	-	A	T	A	C	A	C	C	C	C									46	1M1D9M	46	1M1P4I5M
Read13												C	A	T	A	C	A	C	C	C	C									47	10M	47	5I5M
Read14													A	T	A	C	-	C	C	C	C	C								48	4M1D5M	47	1P4I1D5M
Read15														T	A	C	A	C	C	C	C	C	C	T						49	10M	47	2P3I7M
Read16														T	A	-	A	C	C	C	C	C	C	T	T					49	2M1D8M	47	2P2I1P8M
Read17															A	C	A	C	C	C	C	-	C	T	T	G				50	7M1D3M	47	3P2I5M1D3M
Read18																C	A	C	C	C	C	C	C	T	T	G				51	10M	47	4P1I9M
Read19				C	T	T	T	G	T	T	C	A	-	-	-	-	-	A												38	9M5D1M	38	10M
Read20					T	T	T	G	T	T	C	A	-	-	-	-	-	A	C											39	8M5D2M	39	10M
Read21						T	T	G	T	A	C	A	-	-	-	-	-	A	C	C										40	7M5D3M	40	10M
Read22							T	G	T	T	C	A	-	-	-	-	-	A	C	C	C									41	6M5D4M	41	10M
Read23								G	T	T	C	-	-	-	-	-	-	-	C	C	C	C	T							42	4M7D6M	42	4M2D6M
Read24									T	C	A	-	-	-	-	-	A	C	C	C	C	-	T							43	4M5D5M1D1M	43	9M1D1M
Read25										T	C	A	-	-	-	-	-	A	C	C	C	C	C	T						44	3M5D7M	44	10M
Read26											C	A	-	-	T	-	-	A	C	C	C	C	C	T						45	2M2D1M2D7M	45	2M2P1I2P7M
Read27											C	A	-	-	-	-	-	A	C	C	C	C	C	T	T					45	2M5D8M	45	10M
Read28												A	-	-	-	-	-	A	C	C	C	C	C	T	T	G				46	1M5D9M	46	10M
Read29																		A	C	C	C	C	C	T	T	G	C			52	10M	47	10M

Excerpt from a fictional multiple sequence alignment of 100 columns, with a consensus (SAM/BAM reference) of 95bp, with pale grey columns marking the first (and only) gap/pad region in the reference sequence, 5bp at position 47 to 51 inclusive (using padded co-ordinates). For conciseness the reads are all at most ten bases long. Reads 2 to 18 support 5bp CATAC insert or similar. Differences from the reference are given a medium grey background, namely this insert and a handful of other deliberate read errors. The dark background cells show where a CIGAR pad operation is essential in the unpadded representation. A CIGAR pad operator between two M/X/=/D operators is deemed superfluous, thus a more explicit unpadded reference CIGAR string for 9M5D1M for Read19 is reduced to just 10M, and similarly for the other reads including Read23 where 4M7D6M becomes 4M2D6M. Likewise any leading CIGAR pad operators (after any clipping operators) preceding an M/X/=/D operator are superfluous, giving just 10M rather than 5P10M for Read29. More simply, any trailing CIGAR pad (before any clipping) is superfluous, as in reads 2 to 6.

Whitwham, 2010) was one of the few tools to implement this. This complexity can be avoided by using a padded reference sequence, where instead CIGAR deletion (D) operators are used to mark where a read does not have the gap found in the reference. This approach has been possible historically, and has now been formally recognised as part of the SAM/BAM standard. Some tools and viewers could already handle this, for instance Tablet (Milne *et al.*, 2013) which supported padded and unpadded reference coordinates as this is inherent in older formats like ACE.

This is illustrated in Table 1 which shows a multiple sequence alignment with the POS and CIGAR entries required in SAM using a padded or unpadded reference. As an example, using the unpadded reference it is important to distinguish the inserts of C in Read02 and Read18, or T in Read26 as C----, ----C and --T-- respectively. Using just 1I (one base inserted) in the CIGAR is not sufficient. Similarly for Read11 and Read12, using 4I does not capture the difference in alignment between CATA-

and -ATAC. All the reads spanning this region should include pad operators consistent with the full 5bp insertion in this example. Note that it is possible to have inconsistent padding, especially if the files have been post-processed or merged. Read08 and Read14 demonstrate correct alignments containing adjacent insertion and deletion CIGAR operators.

It is expected that updating existing assembly tools to produce SAM/BAM with a padded reference will be simpler than using the unpadded reference, but that the later will continue to be the main focus of downstream analysis. To facilitate this an (experimental) depad command has been added to samtools, together with a wrapper (http://toolshed.g2.bx.psu.edu/view/peterjc/samtools_depad) for use within the Galaxy platform (Goecks *et al.*, 2010). This has been tested with the unpadded SAM files produced by the MIRA v4.0 assembler.

Note that the ArchiveBAM format currently strips out the CIGAR P operator, reducing a multiple sequence alignment to a collection of pairwise alignments.

1.2 Annotation

One of the strengths of ACE and CAF as assembly file formats is their support for annotation tags intended to guide manual finishing. For example, a potential repeat region identified during assembly can be explicitly marked as such by the assembler. The MIRA assembler (Chevreux *et al.*, 1999) uses a broad collection of such annotation, which is intended to be viewed in a suitable editor like Gap5 (Bonfield and Whitwham, 2010). The SAM/BAM format now defines dummy reads (with no sequence) to hold annotations about a particular region, and additional tags for storing annotations about an individual read. This has been done with a view to extensibility, which should suffice even for rich annotation such as GFF3 format. This may prove useful for reference guided assemblies allowing the assembler to automatically transfer pertinent annotation from the reference genome. Because region specific annotations are held as SAM/BAM reads, they benefit from the existing indexing techniques designed for paired end or strobe reads - including support for features split between contigs (references in SAM/BAM).

1.3 Embedded reference

When working with mapping data to a model organism, it makes sense to store the reference sequence outside the SAM/BAM file. However, for *de novo* assemblies, or mapping to draft genomes, using a separate FASTA file complicates data management. In the interests of a single self contained assembly, the SAM/BAM format now defines how to optionally embed each reference (i.e. assembly consensus) sequence as a long dummy read.

2 CONCLUSION

These additions should make SAM/BAM a sensible file format choice for HTS assembly tools where individual reads are tracked, particularly more traditional overlap-consensus based assemblers (which may come back into the fore as HTS read lengths continue to increase).

Roche Life Sciences already support unpadded BAM output from some of their 'Newbler' 454 sequencing suite, but the extensions to the file format described here would allow a fuller representation. Version 4 of the MIRA assembler supports padded SAM output, using the annotation features. Using samtools these padded SAM assemblies can be converted to unpadded SAM/BAM, and then used with a wide range of downstream analysis tools for tasks such as SNP finding.

Gap5 can now be used to view and edit SAM/BAM assemblies using these new features, and takes full advantage of the annotation present. This is particularly important for manual finishing where the assembler (e.g. MIRA) may tag regions of concern (e.g. identified repeat regions). This replaces existing workflows from MIRA to Gap4 using the CAF file format.

It is our expectation that SAM/BAM can now be used to replace existing first generation sequencing era assembly file formats ACE and CAF, which are inappropriate for current data volumes, and that SAM/BAM will continue to evolve in the future.

ACKNOWLEDGEMENT

REFERENCES

Bonfield, J. K. and Whitwham, A. (2010). Gap5 – editing the billion fragment sequence assembly. *Bioinformatics*, **26**(14), 1699–703.

Bradnam, K. R., *et al.* (2013). Assemblathon 2: evaluating *de novo* methods of genome assembly in three vertebrate species. *GigaScience*, **2**, 10.

Chevreux, B., *et al.* (1999). Genome sequence assembly using trace signals and additional sequence. In *Information Computer Science and Biology: Proceedings of the German Conference on Bioinformatics (GCB)*, pages 45–56.

Cochrane, G., *et al.* (2013). Facing growth in the European Nucleotide Archive. *Nucl. Acids Res.*, **41**(D1), D30–D35.

Cock, P. J. A., *et al.* (2010). The sanger fastq file format for sequences with quality scores, and the solexa/illumina fastq variants. *Nucleic Acids Research*, **38**(6), 1767–71.

Earl, D., *et al.* (2011). Assemblathon 1: a competitive assessment of de novo short read assembly methods. *Genome Res*, **21**(12), 2224–41.

Goecks, J., *et al.* (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*, **11**(8), R86.

Gordon, D., *et al.* (1998). Consed: A Graphical Tool for Sequence Finishing. *Genome Research*, **8**(3), 195–202.

Huang, X. and Madan, A. (1999). Cap3: A dna sequence assembly program. *Genome Res*, **9**(9), 868–77.

Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nat. Methods*, **9**(4), 357–359.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**(14), 1754–1760.

Li, H., *et al.* (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**(16), 2078–9.

Milne, I., *et al.* (2013). Using tablet for visual exploration of second-generation sequencing data. *Brief Bioinformatics*, **14**(2), 193–202.

Robinson, J. T., *et al.* (2011). Integrative genomics viewer. *Nat Biotechnol*, **29**(1), 24–6.

Zerbino, D. R. and Birney, E. (2008). Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome Res*, **18**(5), 821–9.