Manual Reference Pages - samtools (1)

NAME

samtools - Utilities for the Sequence Alignment/Map (SAM) format

CONTENTS

SYNOPSIS

```
samtools view -bt ref_list.txt -o aln.bam aln.sam.gz

samtools sort aln.bam aln.sorted

samtools index aln.sorted.bam

samtools idxstats aln.sorted.bam

samtools view aln.sorted.bam chr2:20,100,000-20,200,000

samtools merge out.bam in1.bam in2.bam in3.bam

samtools faidx ref.fasta

samtools pileup -vcf ref.fasta aln.sorted.bam

samtools mpileup -C50 -gf ref.fasta -r chr3:1,000-2,000 in1.bam in2.bam

samtools tview aln.sorted.bam ref.fasta
```

DESCRIPTION

Samtools is a set of utilities that manipulate alignments in the BAM format. It imports from and exports to the SAM (Sequence Alignment/Map) format, does sorting, merging and indexing, and allows to retrieve reads in any regions swiftly.

Samtools is designed to work on a stream. It regards an input file '-' as the standard input (stdin) and an output file '-' as the standard output (stdout). Several commands can thus be combined with Unix pipes. Samtools always output warning and error messages to the standard error output (stderr).

Samtools is also able to open a BAM (not SAM) file on a remote FTP or HTTP server if the BAM file name starts with 'ftp://' or 'http://'. Samtools checks the current working directory for the index file and will download the index upon absence. Samtools does not retrieve the entire alignment file unless it is asked to do so.

COMMANDS AND OPTIONS

view samtools view [-bchuHS] [-t in.refList] [-o output] [-f reqFlag] [-F skipFlag] [-q minMapQ] [-l library] [-r readGroup] [-R rgFile] <in.bam>|<in.sam> [region1 [...]]

Extract/print all or sub alignments in SAM or BAM format. If no region is specified,

all the alignments will be printed; otherwise only alignments overlapping the specified regions will be output. An alignment may be given multiple times if it is overlapping several regions. A region can be presented, for example, in the following format: 'chr2' (the whole chr2), 'chr2:1000000' (region starting from 1,000,000bp) or 'chr2:1,000,000-2,000,000' (region between 1,000,000 and 2,000,000bp including the end points). The coordinate is 1-based.

OPTIONS:

-b Output in the BAM format.

-f *INT* Only output alignments with all bits in INT present in the FLAG field. INT can be in hex in the format of /^0x[0-9A-F]+/ [0]

-F *INT* Skip alignments with bits present in INT [0]

-h Include the header in the output.

-H Output the header only.

-l *STR* Only output reads in library STR [null]

-o *FILE* Output file [stdout]

-q *INT* Skip alignments with MAPQ smaller than INT [0]

-r *STR* Only output reads in read group STR [null]

-R *FILE* Output reads in read groups listed in *FILE* [null]

-S Input is in SAM. If @SQ header lines are absent, the **'-t'** option is required.

-c Instead of printing the alignments, only count them and print the total number. All filter options, such as **'-f'**, **'-F'** and **'-q'** , are taken into account.

-t *FILE* This file is TAB-delimited. Each line must contain the reference name and the length of the reference, one line for each distinct reference; additional fields are ignored. This file also defines the order of the reference sequences in sorting. If you run 'samtools faidx <ref.fa>', the resultant index file *<ref.fa>.fai* can be used as this *<in.ref_list>* file.

-u Output uncompressed BAM. This option saves time spent on compression/decomprssion and is thus preferred when the output is piped to another samtools command.

tview samtools tview <in.sorted.bam> [ref.fasta]

Text alignment viewer (based on the ncurses library). In the viewer, press '?' for help and press 'g' to check the alignment start from a region in the format like 'chr10:10,000,000' or '=10,000,000' when viewing the same reference sequence.

mpileup samtools mpileup [-Bug] [-C capQcoef] [-r reg] [-f in.fa] [-l list] [-M capMapQ] [-Q minBaseQ] [-q minMapQ] in.bam [in2.bam [...]]

Generate BCF or pileup for one or multiple BAM files. Alignment records are grouped by sample identifiers in @RG header lines. If sample identifiers are absent, each input file is regarded as one sample.

OPTIONS:

-B Disable probabilistic realignment for the computation of base alignment quality (BAQ). BAQ is the Phred-scaled probability of a read base being misaligned. Applying this option greatly helps to reduce false SNPs caused by misalignments.

-C *INT* Coefficient for downgrading mapping quality for reads containing excessive mismatches. Given a read with a phred-scaled probability q of being generated from the mapped position, the new mapping quality is about sqrt((INT-q)/INT)*INT. A zero value disables this functionality; if enabled, the recommended value for BWA is 50. [0]

-e *INT* Phred-scaled gap extension sequencing error probability. Reducing *INT* leads to longer indels. [20]

-f *FILE* The reference file [null]

-g Compute genotype likelihoods and output them in the binary call format (BCF).

-h *INT* Coefficient for modeling homopolymer errors. Given an *l*-long homopolymer run, the sequencing error of an indel of size *s* is modeled as *INT*s/l*. [100]

-l *FILE* File containing a list of sites where pileup or BCF is outputted [null]

-o *INT* Phred-scaled gap open sequencing error probability. Reducing *INT* leads to more indel calls. [40]

-P *STR* Comma dilimited list of platforms (determined by **@RG-PL**) from which indel candidates are obtained. It is recommended to collect indel candidates from sequencing technologies that have low indel error rate such as ILLUMINA. [all]

-q *INT* Minimum mapping quality for an alignment to be used [0]

-Q *INT* Minimum base quality for a base to be considered [13]

-r *STR* Only generate pileup in region *STR* [all sites]

-u Similar to **-g** except that the output is uncompressed BCF, which is preferred for piping.

reheader samtools reheader <in.header.sam> <in.bam>

Replace the header in *in.bam* with the header in *in.header.sam*. This command is much faster than replacing the header with a BAM->SAM->BAM conversion.

sort samtools sort [-no] [-m maxMem] <in.bam> <out.prefix>

Sort alignments by leftmost coordinates. File *<out.prefix>.bam* will be created. This command may also create temporary files *<out.prefix>.%d.bam* when the whole alignment cannot be fitted into memory (controlled by option -m).

OPTIONS:

-o Output the final alignment to the standard output.

-n Sort by read names rather than by chromosomal coordinates

-m *INT* Approximately the maximum required memory. [500000000]

merge samtools merge [-nur] [-h inh.sam] [-R reg] <out.bam> <in1.bam> <in2.bam> [...]

Merge multiple sorted alignments. The header reference lists of all the input BAM files, and the @SQ headers of *inh.sam*, if any, must all refer to the same set of reference sequences. The header reference list and (unless overridden by **-h**) '@' headers of *in1.bam* will be copied to *out.bam*, and the headers of other files will be ignored.

OPTIONS:

-h *FILE* Use the lines of *FILE* as '@' headers to be copied to *out.bam*, replacing any header lines that would otherwise be copied from *in1.bam*. (*FILE* is actually in SAM format, though any alignment records it may contain are ignored.)

-R *STR* Merge files in the specified region indicated by *STR*

-r Attach an RG tag to each alignment. The tag value is inferred from file names.

-n The input alignments are sorted by read names rather than by chromosomal coordinates

-u Uncompressed BAM output

index samtools index <aln.bam>

Index sorted alignment for fast random access. Index file *<aln.bam>.bai* will be created.

idxstats samtools idxstats <aln.bam>

Retrieve and print stats in the index file. The output is TAB delimited with each line consisting of reference sequence name, sequence length, # mapped reads and # unmapped reads.

faidx samtools faidx <ref.fasta> [region1 [...]]

Index reference sequence in the FASTA format or extract subsequence from indexed reference sequence. If no region is specified, **faidx** will index the file and create *<ref.fasta>.fai* on the disk. If regions are speficified, the subsequences will be retrieved and printed to stdout in the FASTA format. The input file can be compressed in the **RAZF** format.

fixmate samtools fixmate <in.nameSrt.bam> <out.bam>

Fill in mate coordinates, ISIZE and mate related flags from a name-sorted alignment.

rmdup samtools rmdup [-sS] <input.srt.bam> <out.bam>

Remove potential PCR duplicates: if multiple read pairs have identical external coordinates, only retain the pair with highest mapping quality. In the paired-end mode, this command **ONLY** works with FR orientation and requires ISIZE is correctly set. It does not work for unpaired reads (e.g. two ends mapped to different chromosomes or orphan reads).

OPTIONS:

-s Remove duplicate for single-end reads. By default, the command works for paired-end reads only.

-S Treat paired-end reads and single-end reads.

calmd samtools calmd [-eubSr] [-C capQcoef] <aln.bam> <ref.fasta>

Generate the MD tag. If the MD tag is already present, this command will give a warning if the MD tag generated is different from the existing tag. Output SAM by default.

OPTIONS:

-e Convert a the read base to = if it is identical to the aligned reference base. Indel caller does not support the = bases at the moment.

-u Output uncompressed BAM

-b Output compressed BAM

-S The input is SAM with header lines

-C *INT* Coefficient to cap mapping quality of poorly mapped reads. See the **pileup** command for details. [0]

-r Perform probabilistic realignment to compute BAQ, which will be used to cap base quality.

pileup samtools pileup [-2sSBicv] [-f in.ref.fasta] [-t in.ref_list] [-l in.site_list] [-C capMapQ] [-M maxMapQ] [-T theta] [-N nHap] [-r pairDiffRate] [-m mask] [-d maxIndelDepth] [-G indelPrior] <in.bam>|<in.sam>

Print the alignment in the pileup format. In the pileup format, each line represents a genomic position, consisting of chromosome name, coordinate, reference base, read bases, read qualities and alignment mapping qualities. Information on match, mismatch, indel, strand, mapping quality and start and end of a read are all encoded at the read base column. At this column, a dot stands for a match to the reference base on the forward strand, a comma for a match on the reverse strand, a '>' or '<' for a reference skip, 'ACGTN' for a mismatch on the forward strand and 'acgtn' for a mismatch on the reverse strand. A pattern '\+[0-9]+[ACGTNacgtn]+' indicates there is an insertion between this reference position and the next reference position. The length of the insertion is given by the integer in the pattern, followed by the inserted sequence.

Similarly, a pattern '-[0-9]+[ACGTNacgtn]+' represents a deletion from the reference. The deleted bases will be presented as '*' in the following lines. Also at the read base column, a symbol '^' marks the start of a read. The ASCII of the character following '^' minus 33 gives the mapping quality. A symbol '$' marks the end of a read segment.

If option **-c** is applied, the consensus base, Phred-scaled consensus quality, SNP quality (i.e. the Phred-scaled probability of the consensus being identical to the reference) and root mean square (RMS) mapping quality of the reads covering the site will be inserted between the 'reference base' and the 'read bases' columns. An indel occupies an additional line. Each indel line consists of chromosome name, coordinate, a star, the genotype, consensus quality, SNP quality, RMS mapping quality, # covering reads, the first alllele, the second allele, # reads supporting the first allele, # reads supporting the second allele and # reads containing indels different from the top two alleles.

NOTE: Since 0.1.10, the 'pileup' command is deprecated by 'mpileup'.

OPTIONS:

-B Disable the BAQ computation. See the **mpileup** command for details.

-c Call the consensus sequence. Options **-T**, **"-N"**, **"-I"** and **"-r** are only effective when **-c"** or **"-g** is in use.

-C *INT* Coefficient for downgrading the mapping quality of poorly mapped reads. See the **mpileup** command for details. [0]

-d *INT* Use the first *NUM* reads in the pileup for indel calling for speed up. Zero for unlimited. [1024]

-f *FILE* The reference sequence in the FASTA format. Index file *FILE.fai* will be created if absent.

-g Generate genotype likelihood in the binary GLFv3 format. This option suppresses -c, -i and -s. This option is deprecated by the **mpileup** command.

-i Only output pileup lines containing indels.

-I *INT* Phred probability of an indel in sequencing/prep. [40]

-l *FILE* List of sites at which pileup is output. This file is space delimited. The first two columns are required to be chromosome and 1-based coordinate. Additional columns are ignored. It is recommended to use option

-m *INT* Filter reads with flag containing bits in *INT* [1796]

-M *INT* Cap mapping quality at INT [60]

-N *INT* Number of haplotypes in the sample (>=2) [2]

-r *FLOAT* Expected fraction of differences between a pair of haplotypes [0.001]

-s Print the mapping quality as the last column. This option makes the output easier to parse, although this format is not space efficient.

-S The input file is in SAM.

-t *FILE* List of reference names ane sequence lengths, in the format described for the **import** command. If this option is present, samtools assumes the input *<in.alignment>* is in SAM format; otherwise it assumes in BAM format. **-s** together with **-l** as in the default format we may not know the mapping quality.

-T *FLOAT* The theta parameter (error dependency coefficient) in the maq consensus calling model [0.85]

SAM FORMAT

SAM is TAB-delimited. Apart from the header lines, which are started with the '@' symbol, each alignment line consists of:

Col	Field	Description

1	QNAME	Query (pair) NAME
2	FLAG	bitwise FLAG
3	RNAME	Reference sequence NAME
4	POS	1-based leftmost POSition/coordinate of clipped sequence
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIAGR	extended CIGAR string
7	MRNM	Mate Reference sequence NaMe ('=' if same as RNAME)
8	MPOS	1-based Mate POSistion
9	ISIZE	Inferred insert SIZE
10	SEQ	query SEQuence on the same strand as the reference
11	QUAL	query QUALity (ASCII-33 gives the Phred base quality)
12	OPT	variable OPTional fields in the format TAG:VTYPE:VALUE

Each bit in the FLAG field is defined as:

Flag	Chr	Description
0x0001	p	the read is paired in sequencing
0x0002	P	the read is mapped in a proper pair
0x0004	u	the query sequence itself is unmapped
0x0008	U	the mate is unmapped
0x0010	r	strand of the query (1 for reverse)
0x0020	R	strand of the mate
0x0040	1	the read is the first read in a pair
0x0080	2	the read is the second read in a pair
0x0100	s	the alignment is not primary
0x0200	f	the read fails platform/vendor quality checks
0x0400	d	the read is either a PCR or an optical duplicate

EXAMPLES

- Import SAM to BAM when **@SQ** lines are present in the header:

  ```
  samtools view -bS aln.sam > aln.bam
  ```

  If **@SQ** lines are absent:

  ```
  samtools faidx ref.fa
  samtools view -bt ref.fa.fai aln.sam > aln.bam
  ```

  where *ref.fa.fai* is generated automatically by the **faidx** command.

- Attach the **RG** tag while merging sorted alignments:

  ```
  perl -e 'print "@RG\tID:ga\tSM:hs\tLB:ga\tPL:Illumina\n@RG\tID:454\tSM:hs\tLB:454\tPL:454\n"' >
  rg.txt
  samtools merge -rh rg.txt merged.bam ga.bam 454.bam
  ```

  The value in a **RG** tag is determined by the file name the read is coming from. In this example,

in the *merged.bam*, reads from *ga.bam* will be attached *RG:Z:ga*, while reads from *454.bam* will be attached *RG:Z:454*.

- Call SNPs and short indels for one diploid individual:

```
samtools mpileup -ugf ref.fa aln.bam | bcftools view -bvcg - > var.raw.bcf
bcftools view var.raw.bcf | vcfutils.pl varFilter -D 100 > var.flt.vcf
```

The **-D** option of varFilter controls the maximum read depth, which should be adjusted to about twice the average read depth. One may consider to add **-C50** to **mpileup** if mapping quality is overestimated for reads containing excessive mismatches. Applying this option usually helps **BWA-short** but may not other mappers.

- Call SNPs and short indels for multiple diploid individuals:

```
samtools mpileup -P ILLUMINA -ugf ref.fa *.bam | bcftools view -bcvg - > var.raw.bcf
bcftools view var.raw.bcf | vcfutils.pl varFilter -D 2000 > var.flt.vcf
```

Individuals are identified from the **SM** tags in the **@RG** header lines. Individuals can be pooled in one alignment file; one individual can also be separated into multiple files. The **-P** option specifies that indel candidates should be collected only from read groups with the **@RG-PL** tag set to *ILLUMINA*. Collecting indel candidates from reads sequenced by an indel-prone technology may affect the performance of indel calling.

- Derive the allele frequency spectrum (AFS) on a list of sites from multiple individuals:

```
samtools mpileup -Igf ref.fa *.bam > all.bcf
bcftools view -bl sites.list all.bcf > sites.bcf
bcftools view -cGP cond2 sites.bcf > /dev/null 2> sites.1.afs
bcftools view -cGP sites.1.afs sites.bcf > /dev/null 2> sites.2.afs
bcftools view -cGP sites.2.afs sites.bcf > /dev/null 2> sites.3.afs
......
```

where *sites.list* contains the list of sites with each line consisting of the reference sequence name and position. The following **bcftools** commands estimate AFS by EM.

- Dump BAQ applied alignment for other SNP callers:

```
samtools calmd -br aln.bam > aln.baq.bam
```

It adds and corrects the **NM** and **MD** tags at the same time. The **calmd** command also comes with the **-C** option, the same as the one in **pileup** and **mpileup**. Apply if it helps.

LIMITATIONS

- Unaligned words used in bam_import.c, bam_endian.h, bam.c and bam_aux.c.
- In merging, the input files are required to have the same number of reference sequences. The requirement can be relaxed. In addition, merging does not reconstruct the header dictionaries automatically. Endusers have to provide the correct header. Picard is better at merging.
- Samtools paired-end rmdup does not work for unpaired reads (e.g. orphan reads or ends mapped to different chromosomes). If this is a concern, please use Picard's MarkDuplicate which correctly handles these cases, although a little slower.

AUTHOR

Heng Li from the Sanger Institute wrote the C version of samtools. Bob Handsaker from the Broad Institute implemented the BGZF library and Jue Ruan from Beijing Genomics Institute wrote the RAZF library. John Marshall and Petr Danecek contribute to the source code and various people from the 1000 Genomes Project have contributed to the SAM format specification.

SEE ALSO

Samtools website: <http://samtools.sourceforge.net>