Manual page from samtools-1.5
released on 21 June 2017

NAME

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

SYNOPSIS

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

samtools sort -T /tmp/aln.sorted -o aln.sorted.bam aln.bam

samtools index aln.sorted.bam

samtools idxstats aln.sorted.bam

samtools flagstat aln.sorted.bam

samtools stats aln.sorted.bam

samtools bedcov aln.sorted.bam

samtools depth 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 tview aln.sorted.bam ref.fasta

samtools split merged.bam

samtools quickcheck in1.bam in2.cram

samtools dict -a GRCh38 -s "Homo sapiens" ref.fasta

samtools fixmate in.namesorted.sam out.bam

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

samtools flags PAIRED,UNMAP,MUNMAP

samtools fastq input.bam > output.fastq

samtools fasta input.bam > output.fasta

samtools addreplacerg -r 'ID:fish' -r 'LB:1334' -r 'SM:alpha' -o output.bam input.bam

samtools collate aln.sorted.bam aln.name_collated.bam

samtools depad input.bam

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 [*options*] *in.sam*|*in.bam*|*in.cram* [*region*...]

With no options or regions specified, prints all alignments in the specified input alignment file (in SAM, BAM, or CRAM format) to standard output in SAM format (with no header).

You may specify one or more space-separated region specifications after the input filename to restrict output to only those alignments which overlap the specified region(s). Use of region specifications requires a coordinate-sorted and indexed input file (in BAM or CRAM format).

The **-b**, **-C**, **-1**, **-u**, **-h**, **-H**, and **-c** options change the output format from the default of headerless SAM, and the **-o** and **-U** options set the output file name(s).

The **-t** and **-T** options provide additional reference data. One of these two options is required when SAM input does not contain @SQ headers, and the **-T** option is required whenever writing CRAM output.

The **-L**, **-r**, **-R**, **-s**, **-q**, **-l**, **-m**, **-f**, **-F**, and **-G** options filter the alignments that will be included in the output to only those alignments that match certain criteria.

The **-x** and **-B** options modify the data which is contained in each alignment.

Finally, the **-@** option can be used to allocate additional threads to be used for compression, and the **-?** option requests a long help message.

REGIONS: Regions can be specified as: RNAME[:STARTPOS[-ENDPOS]] and all position coordinates are 1-based.

Important note: when multiple regions are given, some alignments may be output multiple times if they overlap more than one of the specified regions.

Examples of region specifications:

chr1 Output all alignments mapped to the reference sequence named `chr1' (i.e. @SQ SN:chr1).

chr2:1000000 The region on chr2 beginning at base position 1,000,000 and ending at the end of the chromosome.

chr3:1000-2000 The 1001bp region on chr3 beginning at base position 1,000 and ending at base position 2,000 (including both end positions).

'*' Output the unmapped reads at the end of the file. (This does not include any unmapped reads placed on a reference sequence alongside their mapped mates.)

. Output all alignments. (Mostly unnecessary as not specifying a region at all has the same effect.)

OPTIONS:

-b Output in the BAM format.

-C Output in the CRAM format (requires -T).

-1 Enable fast BAM compression (implies -b).

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

-h Include the header in the output.

-H Output the header only.

-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.

-? Output long help and exit immediately.

-o *FILE* Output to *FILE [stdout].*

-U *FILE* Write alignments that are *not* selected by the various filter options to *FILE*. When this option is used, all alignments (or all alignments intersecting the *regions* specified) are written to either the output file or this file, but never both.

-t *FILE* A tab-delimited *FILE*. Each line must contain the reference name in the first column and the length of the reference in the second column, with one line for each distinct reference. Any additional fields beyond the second column are ignored. This file also defines the order of the reference sequences in sorting. If you run: `samtools faidx <ref.fa>', the resulting index file *<ref.fa>.fai* can be used as this *FILE*.

-T *FILE* A FASTA format reference *FILE*, optionally compressed by **bgzip** and ideally indexed by **samtools faidx**. If an index is not present, one will be generated for you.

-L *FILE* Only output alignments overlapping the input BED *FILE* [null].

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

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

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

-l *STR* Only output alignments in library *STR* [null].

-m *INT* Only output alignments with number of CIGAR bases consuming query sequence ≥ *INT* [0]

-f *INT* Only output alignments with all bits set in *INT* present in the FLAG field. *INT* can be specified in hex by beginning with `0x' (i.e. /^0x[0-9A-F]+/) or in octal by beginning with `0' (i.e. /^0[0-7]+/) [0].

-F *INT* Do not output alignments with any bits set in *INT* present in the FLAG field. *INT* can be specified in hex by beginning with `0x' (i.e. /^0x[0-9A-F]+/) or in octal by beginning with `0' (i.e. /^0[0-7]+/) [0].

-G *INT* Do not output alignments with all bits set in *INT* present in the FLAG field. This is the opposite of *-f* such that *-f12 -G12* is the same as no filtering at all. *INT* can be specified in hex by beginning with `0x' (i.e. /^0x[0-9A-F]+/) or in octal by beginning with `0' (i.e. /^0[0-7]+/) [0].

-x *STR* Read tag to exclude from output (repeatable) [null]

-B Collapse the backward CIGAR operation.

-s *FLOAT* Output only a proportion of the input alignments. This subsampling acts in the same way on all of the alignment records in the same template or read pair, so it never keeps a read but not its mate.

The integer and fractional parts of the **-s** *INT***.***FRAC* option are used separately: the part after the decimal point sets the fraction of templates/pairs to be kept, while the integer part is used as a seed that influences *which* subset of reads is kept.

When subsampling data that has previously been subsampled, be sure to use a different seed value from those used previously; otherwise more reads will be retained than expected.

-@ *INT* Number of BAM compression threads to use in addition to main thread [0].

-S Ignored for compatibility with previous samtools versions. Previously this option was required if input was in SAM format, but now the correct format is automatically detected by examining the first few characters of input.

sort samtools sort [**-l** *level*] [**-m** *maxMem*] [**-o** *out.bam*] [**-O** *format*] [**-n**] [**-t** *tag*] [**-T** *tmpprefix*] [**-@** *threads*] [*in.sam*|*in.bam*|*in.cram*]

Sort alignments by leftmost coordinates, or by read name when **-n** is used. An appropriate **@HD-SO** sort order header tag will be added or an existing one updated if necessary.

The sorted output is written to standard output by default, or to the specified file (*out.bam*) when **-o** is used. This command will also create temporary files *tmpprefix.%d*.**bam** as needed when the entire alignment data cannot fit into memory (as controlled via the **-m** option).

Options:

-l *INT* Set the desired compression level for the final output file, ranging from 0 (uncompressed) or 1 (fastest but minimal compression) to 9 (best compression but slowest to write), similarly to **gzip**(1)'s compression level setting.

If **-l** is not used, the default compression level will apply.

-m *INT* Approximately the maximum required memory per thread, specified either in bytes or with a **K**, **M**, or **G** suffix. [768 MiB]

To prevent sort from creating a huge number of temporary files, it enforces a minimum value of 1M for this setting.

-n Sort by read names (i.e., the **QNAME** field) rather than by chromosomal coordinates.

-t *TAG* Sort first by the value in the alignment tag TAG, then by position or name (if also using **-n**). **-o** *FILE* Write the final sorted output to *FILE*, rather than to standard output.

-O *FORMAT* Write the final output as **sam**, **bam**, or **cram**.

By default, samtools tries to select a format based on the **-o** filename extension; if output is to standard output or no format can be deduced, **bam** is selected.

-T *PREFIX* Write temporary files to *PREFIX.nnnn*.**bam,** or if the specified *PREFIX* is an existing directory, to *PREFIX*/**samtools.***mmm.mmm*.**tmp.***nnnn*.**bam,** where *mmm* is unique to this invocation of the **sort** command.

By default, any temporary files are written alongside the output file, as *out.bam*.**tmp.***nnnn*.**bam,** or if output is to standard output, in the current directory as **samtools.***mmm.mmm*.**tmp.***nnnn*.**bam.**

-@ *INT* Set number of sorting and compression threads. By default, operation is single-threaded.

Ordering Rules

The following rules are used for ordering records.

If option **-t** is in use, records are first sorted by the value of the given alignment tag, and then by position or name (if using **-n**). For example, “-t RG” will make read group the primary sort key. The rules for ordering by tag are:

- Records that do not have the tag are sorted before ones that do.
- If the types of the tags are different, they will be sorted so that single character tags (type A) come before array tags (type B), then string tags (types H and Z), then numeric tags (types f and i).
- Numeric tags (types f and i) are compared by value. Note that comparisons of floating-point values are subject to issues of rounding and precision.
- String tags (types H and Z) are compared based on the binary contents of the tag using the C **strcmp**(3) function.
- Character tags (type A) are compared by binary character value.
- No attempt is made to compare tags of other types — notably type B array values will not be compared.

When the **-n** option is present, records are sorted by name. Names are compared so as to give a “natural” ordering — i.e. sections consisting of digits are compared numerically while all other sections are compared based on their binary representation. This means “a1” will come before “b1” and “a9” will come before “a10”. Records with the same name will be ordered according to the values of the READ1 and READ2 flags (see **flags**).

When the **-n** option is **not** present, reads are sorted by reference (according to the order of the @SQ header records), then by position in the reference, and then by the REVERSE flag.

Note

Historically **samtools sort** also accepted a less flexible way of specifying the final and temporary output filenames:

samtools sort [**-f**] [**-o**] *in.bam out.prefix*

This has now been removed. The previous *out.prefix* argument (and **-f** option, if any) should be changed to an appropriate combination of **-T** *PREFIX* and **-o** *FILE*. The previous **-o** option should be removed, as output defaults to standard output.

index samtools index [**-bc**] [**-m** *INT*] *aln.bam*|*aln.cram* [*out.index*]

Index a coordinate-sorted BAM or CRAM file for fast random access. (Note that this does not work with SAM files even if they are bgzip compressed — to index such files, use tabix(1) instead.)

This index is needed when *region* arguments are used to limit **samtools view** and similar commands to particular regions of interest.

If an output filename is given, the index file will be written to *out.index*. Otherwise, for a CRAM file *aln.cram*, index file *aln.cram*.**crai** will be created; for a BAM file *aln.bam*, either *aln.bam*.**bai** or *aln.bam*.**csi** will be created, depending on the index format selected.

Options:

-b Create a BAI index. This is currently the default when no format options are used.

-c Create a CSI index. By default, the minimum interval size for the index is 2^14, which is the same as the fixed value used by the BAI format.

-m *INT* Create a CSI index, with a minimum interval size of 2^INT.

idxstats samtools idxstats *in.sam*|*in.bam*|*in.cram*

Retrieve and print stats in the index file corresponding to the input file. Before calling idxstats, the input BAM file must be indexed by samtools index.

The output is TAB-delimited with each line consisting of reference sequence name, sequence length, # mapped reads and # unmapped reads. It is written to stdout.

flagstat samtools flagstat *in.sam*|*in.bam*|*in.cram*

Does a full pass through the input file to calculate and print statistics to stdout.

Provides counts for each of 13 categories based primarily on bit flags in the FLAG field. Each category in the output is broken down into QC pass and QC fail, which is presented as "#PASS + #FAIL" followed by a description of the category.

The first row of output gives the total number of reads that are QC pass and fail (according to flag bit 0x200). For example:

122 + 28 in total (QC-passed reads + QC-failed reads)

Which would indicate that there are a total of 150 reads in the input file, 122 of which are marked as QC pass and 28 of which are marked as "not passing quality controls"

Following this, additional categories are given for reads which are:

secondary 0x100 bit set

supplementary 0x800 bit set

duplicates 0x400 bit set

mapped 0x4 bit not set

paired in sequencing
0x1 bit set

read1 both 0x1 and 0x40 bits set

read2 both 0x1 and 0x80 bits set

properly paired both 0x1 and 0x2 bits set and 0x4 bit not set

with itself and mate mapped
0x1 bit set and neither 0x4 nor 0x8 bits set

singletons both 0x1 and 0x8 bits set and bit 0x4 not set

And finally, two rows are given that additionally filter on the reference name (RNAME), mate reference name (MRNM), and mapping quality (MAPQ) fields:

with mate mapped to a different chr
0x1 bit set and neither 0x4 nor 0x8 bits set and MRNM not equal to RNAME

with mate mapped to a different chr (mapQ>=5)
0x1 bit set and neither 0x4 nor 0x8 bits set and MRNM not equal to RNAME and MAPQ >= 5

stats samtools stats [*options*] *in.sam*|*in.bam*|*in.cram* [*region*...]

samtools stats collects statistics from BAM files and outputs in a text format. The output can be visualized graphically using plot-bamstats.

Options:

-c, --coverage *MIN*,*MAX*,*STEP*
Set coverage distribution to the specified range (MIN, MAX, STEP all given as integers) [1,1000,1]

-d, --remove-dups
Exclude from statistics reads marked as duplicates

-f, --required-flag *STR*|*INT*
Required flag, 0 for unset. See also `samtools flags` [0]

-F, --filtering-flag *STR*|*INT*
Filtering flag, 0 for unset. See also `samtools flags` [0]

--GC-depth *FLOAT*
the size of GC-depth bins (decreasing bin size increases memory requirement) [2e4]

-h, --help This help message

-i, --insert-size *INT*
Maximum insert size [8000]

-l, --id *STR* Include only listed read group or sample name []

-l, --read-length *INT*
Include in the statistics only reads with the given read length []

-m, --most-inserts *FLOAT*
Report only the main part of inserts [0.99]

-P, --split-prefix *STR*
A path or string prefix to prepend to filenames output when creating categorised statistics files with **-S**/**--split**. [input filename]

-q, --trim-quality *INT*
The BWA trimming parameter [0]

-r, --ref-seq *FILE* Reference sequence (required for GC-depth and mismatches-per-cycle calculation). []

-S, --split *TAG* In addition to the complete statistics, also output categorised statistics based on the tagged field *TAG* (e.g., use **--split RG** to split into read groups).

Categorised statistics are written to files named <*prefix*>_<*value*>.bamstat, where *prefix* is as given by **--split-prefix** (or the input filename by default) and *value* has been encountered as the specified tagged field's value in one or more alignment records.

-t, --target-regions *FILE*
Do stats in these regions only. Tab-delimited file chr,from,to, 1-based, inclusive. []

-x, --sparse Suppress outputting IS rows where there are no insertions.

bedcov samtools bedcov [*options*] *region.bed in1.sam*|*in1.bam*|*in1.cram*[...]

Reports the total read base count (i.e. the sum of per base read depths) for each genomic region specified in the supplied BED file. Counts for each alignment file supplied are reported in separate columns.

Options:

-Q *INT* Only count reads with mapping quality greater than *INT*

depth samtools depth [*options*] [*in1.sam*|*in1.bam*|*in1.cram* [*in2.sam*|*in2.bam*|*in2.cram*] [...]]

Computes the depth at each position or region.

Options:

-a Output all positions (including those with zero depth)

-a -a, -aa Output absolutely all positions, including unused reference sequences. Note that when used in conjunction with a BED file the -a option may sometimes operate as if -aa was specified if the reference sequence has coverage outside of the region specified in the BED file.

-b *FILE* Compute depth at list of positions or regions in specified BED *FILE*. []

-f *FILE* Use the BAM files specified in the *FILE* (a file of filenames, one file per line) []

-l *INT* Ignore reads shorter than *INT*

-m, -d *INT* Truncate reported depth at a maximum of *INT* reads. [8000]

-q *INT* Only count reads with base quality greater than *INT*

-Q *INT* Only count reads with mapping quality greater than *INT*

-r *CHR*:*FROM*-*TO*
Only report depth in specified region.

merge samtools merge [-nur1f] [-h inh.sam] [-R reg] [-b <list>] <out.bam> <in1.bam> [<in2.bam> <in3.bam> ... <inN.bam>]

Merge multiple sorted alignment files, producing a single sorted output file that contains all the input records and maintains the existing sort order.

If **-h** is specified the @SQ headers of input files will be merged into the specified header, otherwise they will be merged into a composite header created from the input headers. If in the process of merging @SQ lines for coordinate sorted input files, a conflict arises as to the order (for example input1.bam has @SQ for a,b,c and input2.bam has b,a,c) then the resulting output file will need to be re-sorted back into coordinate order.

Unless the **-c** or **-p** flags are specified then when merging @RG and @PG records into the output header then any IDs found to be duplicates of existing IDs in the output header will have a suffix appended to them to differentiate them from similar header records from other files and the read records will be updated to reflect this.

The ordering of the records in the input files must match the usage of the **-n** and **-t** command-line options. If they do not, the output order will be undefined. See **sort** for information about record ordering.

OPTIONS:

-1 Use zlib compression level 1 to compress the output.

-b *FILE* List of input BAM files, one file per line.

-f Force to overwrite the output file if present.

-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.)

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

-t TAG The input alignments have been sorted by the value of TAG, then by either position or name (if **-n** is given).

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

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

-u Uncompressed BAM output

-c When several input files contain @RG headers with the same ID, emit only one of them (namely, the header line from the first file we find that ID in) to the merged output file. Combining these similar headers is usually the right thing to do when the files being merged originated from the same file.

Without **-c**, all @RG headers appear in the output file, with random suffixes added to their IDs where necessary to differentiate them.

-p Similarly, for each @PG ID in the set of files to merge, use the @PG line of the first file we find that ID in rather than adding a suffix to differentiate similar IDs.

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 specified, the subsequences will be retrieved and printed to stdout in the FASTA format.

The input file can be compressed in the **BGZF** format.

The sequences in the input file should all have different names. If they do not, indexing will emit a warning about duplicate sequences and retrieval will only produce subsequences from the first sequence with the duplicated name.

tview samtools tview [**-p** *chr:pos*] [**-s** *STR*] [**-d** *display*] <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.

Options:

-d *display* Output as (H)tml or (C)urses or (T)ext

-p *chr:pos* Go directly to this position

-s *STR* Display only alignments from this sample or read group

split samtools split [*options*] *merged.sam*|*merged.bam*|*merged.cram*

Splits a file by read group.

Options:

-u *FILE1* Put reads with no RG tag or an unrecognised RG tag into *FILE1*

-u *FILE1*:*FILE2* As above, but assigns an RG tag as given in the header of *FILE2*

-f *STRING* Output filename format string (see below) ["%*_%#.%."]

-v Verbose output

Format string expansions:

%%	%
%*	basename
%#	@RG index
%!	@RG ID
%.	output format filename extension

quickcheck samtools quickcheck [*options*] *in.sam*|*in.bam*|*in.cram* [...]

Quickly check that input files appear to be intact. Checks that beginning of the file contains a valid header (all formats) containing at least one target sequence and then seeks to the end of the file and checks that an end-of-file (EOF) is present and intact (BAM only).

Data in the middle of the file is not read since that would be much more time consuming, so please note that this command will not detect internal corruption, but is useful for testing that files are not truncated before performing more intensive tasks on them.

This command will exit with a non-zero exit code if any input files don't have a valid header or are missing an EOF block. Otherwise it will exit successfully (with a zero exit code).

Options:

-v Verbose output: will additionally print the names of all input files that don't pass the check to stdout. Multiple -v options will cause additional messages regarding check results to be printed to stderr.

dict samtools dict <ref.fasta|ref.fasta.gz>

Create a sequence dictionary file from a fasta file.

OPTIONS:

-a, --assembly *STR*
Specify the assembly for the AS tag.

-H, --no-header Do not print the @HD header line.

-o, --output *FILE* Output to *FILE* [stdout].

-s, --species *STR*
Specify the species for the SP tag.

-u, --uri *STR* Specify the URI for the UR tag. Defaults to the absolute path of *ref.fasta* unless reading from stdin.

fixmate samtools fixmate [**-rpc**] [**-O** *format*] *in.nameSrt.bam out.bam*

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

OPTIONS:

-r Remove secondary and unmapped reads.

-p Disable FR proper pair check.

-c Add template cigar ct tag.

-O *FORMAT* Write the final output as **sam**, **bam**, or **cram**.

By default, samtools tries to select a format based on the output filename extension; if output is to standard output or no format can be deduced, **bam** is selected.

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

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

In the pileup format (without **-u** or **-g**), each line represents a genomic position, consisting of chromosome name, 1-based coordinate, reference base, the number of reads covering the site, read bases, base 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.

Note that there are two orthogonal ways to specify locations in the input file; via **-r** *region* and **-l** *file*. The former uses (and requires) an index to do random access while the latter streams through the file contents filtering out the specified regions, requiring no index. The two may be used in conjunction. For example a BED file containing locations of genes in chromosome 20 could be specified using **-r 20 -l chr20.bed**, meaning that the index is used to find chromosome 20 and then it is filtered for the regions listed in the bed file.

Input Options:

-6, --illumina1.3+ Assume the quality is in the Illumina 1.3+ encoding.

-A, --count-orphans
Do not skip anomalous read pairs in variant calling.

-b, --bam-list *FILE*
List of input BAM files, one file per line [null]

-B, --no-BAQ 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, --adjust-MQ *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]

-d, --max-depth *INT*
At a position, read maximally *INT* reads per input file. Note that samtools has a minimum value of *8000/n* where *n* is the number of input files given to mpileup. This means the default is highly likely to be increased. Once above the cross-sample minimum of 8000 the -d parameter will have an effect. [250]

-E, --redo-BAQ Recalculate BAQ on the fly, ignore existing BQ tags

-f, --fasta-ref *FILE*
The **faidx**-indexed reference file in the FASTA format. The file can be optionally compressed by **bgzip**. [null]

-G, --exclude-RG *FILE*
Exclude reads from readgroups listed in FILE (one @RG-ID per line)

-l, --positions *FILE*
BED or position list file containing a list of regions or sites where pileup or BCF should be generated. Position list files contain two columns (chromosome and position) and start counting from 1. BED files contain at least 3 columns (chromosome, start and end position) and are 0-based half-open.
While it is possible to mix both position-list and BED coordinates in the same file, this is strongly ill advised due to the differing coordinate systems. [null]

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

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

-r, --region *STR* Only generate pileup in region. Requires the BAM files to be indexed. If used in conjunction with -l then considers the intersection of the two requests. *STR* [all sites]

-R, --ignore-RG Ignore RG tags. Treat all reads in one BAM as one sample.

--rf, --incl-flags *STR*|*INT*
Required flags: skip reads with mask bits unset [null]

--ff, --excl-flags *STR*|*INT*
Filter flags: skip reads with mask bits set [UNMAP,SECONDARY,QCFAIL,DUP]

-x, --ignore-overlaps
Disable read-pair overlap detection.

Output Options:

-o, --output *FILE* Write pileup or VCF/BCF output to *FILE*, rather than the default of standard output.

(The same short option is used for both **--open-prob** and **--output**. If **-o**'s argument contains any non-digit characters other than a leading + or - sign, it is interpreted as **--output**. Usually the filename extension will take care of this, but to write to an entirely numeric filename use **-o ./123** or **--output 123**.)

-g, --BCF Compute genotype likelihoods and output them in the binary call format (BCF). As of v1.0, this is BCF2 which is incompatible with the BCF1 format produced by previous (0.1.x) versions of samtools.

-v, --VCF Compute genotype likelihoods and output them in the variant call format (VCF). Output is bgzip-compressed VCF unless **-u** option is set.

Output Options for mpileup format (without -g or -v):

-O, --output-BP Output base positions on reads.

-s, --output-MQ Output mapping quality.

-a Output all positions, including those with zero depth.

-a -a, -aa Output absolutely all positions, including unused reference sequences. Note that when used in conjunction with a BED file the -a option may sometimes operate as if -aa was specified if the reference sequence has coverage outside of the region specified in the BED file.

Output Options for VCF/BCF format (with -g or -v):

-D Output per-sample read depth [DEPRECATED - use **-t DP** instead]

-S Output per-sample Phred-scaled strand bias P-value [DEPRECATED - use **-t SP** instead]

-t, --output-tags *LIST*
Comma-separated list of FORMAT and INFO tags to output (case-insensitive): **AD** (Allelic depth, FORMAT), **INFO/AD** (Total allelic depth, INFO), **ADF** (Allelic depths on the forward strand, FORMAT), **INFO/ADF** (Total allelic depths on the forward strand, INFO), **ADR** (Allelic depths on the reverse strand, FORMAT), **INFO/ADR** (Total allelic depths on the reverse strand, INFO), **DP** (Number of high-quality bases, FORMAT), **DV** (Deprecated in favor of AD; Number of high-quality non-reference bases, FORMAT), **DPR** (Deprecated in favor of AD; Number of high-quality bases for each observed allele, FORMAT), **INFO/DPR** (Number of high-quality bases for each observed allele, INFO), **DP4** (Deprecated in favor of ADF and ADR; Number of high-quality ref-forward, ref-reverse, alt-forward and alt-reverse bases, FORMAT), **SP** (Phred-scaled strand bias P-value, FORMAT) [null]

-u, --uncompressed
Generate uncompressed VCF/BCF output, which is preferred for piping.

-V Output per-sample number of non-reference reads [DEPRECATED - use **-t DV** instead]

Options for SNP/INDEL Genotype Likelihood Computation (for -g or -v):

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

-F, --gap-frac *FLOAT*
Minimum fraction of gapped reads [0.002]

-h, --tandem-qual *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]

-I, --skip-indels Do not perform INDEL calling

-L, --max-idepth *INT*
Skip INDEL calling if the average per-input-file depth is above *INT*. [250]

-m, --min-ireads *INT*
Minimum number gapped reads for indel candidates *INT*. [1]

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

(The same short option is used for both **--open-prob** and **--output**. When **-o**'s argument contains only an optional + or - sign followed by the digits 0 to 9, it is interpreted as **--open-prob**.)

-p, --per-sample-mF
Apply **-m** and **-F** thresholds per sample to increase sensitivity of calling. By default both options are applied to reads pooled from all samples.

-P, --platforms *STR*
Comma-delimited 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]

flags samtools flags INT|STR[,...]

Convert between textual and numeric flag representation.

FLAGS:

0x1	PAIRED	paired-end (or multiple-segment) sequencing technology
0x2	PROPER_PAIR	each segment properly aligned according to the aligner
0x4	UNMAP	segment unmapped
0x8	MUNMAP	next segment in the template unmapped
0x10	REVERSE	SEQ is reverse complemented
0x20	MREVERSE	SEQ of the next segment in the template is reverse complemented
0x40	READ1	the first segment in the template
0x80	READ2	the last segment in the template
0x100	SECONDARY	secondary alignment
0x200	QCFAIL	not passing quality controls
0x400	DUP	PCR or optical duplicate
0x800	SUPPLEMENTARY	supplementary alignment

fastq/a samtools fastq [*options*] *in.bam*
samtools fasta [*options*] *in.bam*

Converts a BAM or CRAM into either FASTQ or FASTA format depending on the command invoked. The FASTQ files will be automatically compressed if the filenames have a .gz or .bgzf extention.

OPTIONS:

-n By default, either '/1' or '/2' is added to the end of read names where the corresponding BAM_READ1 or BAM_READ2 flag is set. Using **-n** causes read names to be left as they are.

-N Always add either '/1' or '/2' to the end of read names even when put into different files.

-O Use quality values from OQ tags in preference to standard quality string if available.

-s FILE Write singleton reads in FASTQ format to FILE instead of outputting them.

-t Copy RG, BC and QT tags to the FASTQ header line, if they exist.

-T TAGLIST Specify a comma-separated list of tags to copy to the FASTQ header line, if they exist.

-1 FILE Write reads with the BAM_READ1 flag set to FILE instead of outputting them.

-2 FILE Write reads with the BAM_READ2 flag set to FILE instead of outputting them.

-0 FILE Write reads with both or neither of the BAM_READ1 and BAM_READ2 flags set to FILE instead of outputting them.

-f *INT* Only output alignments with all bits set in *INT* present in the FLAG field. *INT* can be specified in hex by beginning with `0x' (i.e. /^0x[0-9A-F]+/) or in octal by beginning with `0' (i.e. /^0[0-7]+/) [0].

-F *INT* Do not output alignments with any bits set in *INT* present in the FLAG field. *INT* can be specified in hex by beginning with `0x' (i.e. /^0x[0-9A-F]+/) or in octal by beginning with `0' (i.e. /^0[0-7]+/) [0].

-G *INT* Only EXCLUDE reads with all of the bits set in *INT* present in the FLAG field. *INT* can be specified in hex by beginning with `0x' (i.e. /^0x[0-9A-F]+/) or in octal by beginning with `0' (i.e. /^0[0-7]+/) [0].

-i add Illumina Casava 1.8 format entry to header (eg 1:N:0:ATCACG)

-c [0..9] set compression level when writing gz or bgzf fastq files.

--i1 FILE write first index reads to FILE

--i2 FILE write second index reads to FILE

--barcode-tag TAG
aux tag to find index reads in [default: BC]

--quality-tag TAG
aux tag to find index quality in [default: QT]

--index-format STR
string to describe how to parse the barcode and quality tags. For example:

i14i8 the first 14 characters are index 1, the next 8 characters are index 2

n8i14 ignore the first 8 characters, and use the next 14 characters for index 1

If the tag contains a separator, then the numeric part can be replaced with '*' to mean 'read until the separator or end of tag', for example:

n*i* ignore the left part of the tag until the separator, then use the second part

collate samtools collate [*options*] *in.sam*|*in.bam*|*in.cram* [*out.prefix*]

Shuffles and groups reads together by their names. A faster alternative to a full query name sort, **collate** ensures that reads of the same name are grouped together in contiguous groups, but doesn't make any guarantees about the order of read names between groups.

The output from this command should be suitable for any operation that requires all reads from the same template to be grouped together.

Options:

-O Output to stdout rather than to files starting with out.prefix

-u Write uncompressed BAM output

-l *INT* Compression level. [1]

-n *INT* Number of temporary files to use. [64]

reheader samtools reheader [**-iP**] *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.

By default this command outputs the BAM or CRAM file to standard output (stdout), but for CRAM format files it has the option to perform an in-place edit, both reading and writing to the same file. No validity checking is performed on the header, nor that it is suitable to use with the sequence data itself.

OPTIONS:

-P, --no-PG Do not generate an @PG header line.

-i, --in-place

Perform the header edit in-place, if possible. This only works on CRAM files and only if there is sufficient room to store the new header. The amount of space available will differ for each CRAM file.

cat samtools cat [-b list] [-h header.sam] [-o out.bam] <in1.bam> <in2.bam> [...]

Concatenate BAMs or CRAMs. Although this works on either BAM or CRAM, all input files must be the same format as each other. The sequence dictionary of each input file must be identical, although this command does not check this. This command uses a similar trick to **reheader** which enables fast BAM concatenation.

OPTIONS:

-b *FOFN* Read the list of input BAM or CRAM files from *FOFN*. These are concatenated prior to any files specified on the command line. Multiple **-b** *FOFN* options may be specified to concatenate multiple lists of BAM/CRAM files.

-h *FILE* Uses the SAM header from *FILE*. By default the header is taken from the first file to be concatenated.

-o *FILE* Write the concatenated output to *FILE*. By default this is sent to stdout.

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 duplicates for single-end reads. By default, the command works for paired-end reads only.

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

addreplacerg samtools addreplacerg [-r rg line | -R rg ID] [-m mode] [-l level] [-o out.bam] <input.bam>

Adds or replaces read group tags in a file.

OPTIONS:

-r *STRING* Allows you to specify a read group line to append to the header and applies it to the reads specified by the -m option. If repeated it automatically adds in tabs between invocations.

-R *STRING* Allows you to specify the read group ID of an existing @RG line and applies it to the reads specified.

-m *MODE* If you choose orphan_only then existing RG tags are not overwritten, if you choose overwrite_all, existing RG tags are overwritten. The default is overwrite_all.

-o *STRING* Write the final output to STRING. The default is to write to stdout.

By default, samtools tries to select a format based on the output filename extension; if output is to standard output or no format can be deduced, **bam** is selected.

calmd samtools calmd [-Eeubr] [-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.

Calmd can also read and write CRAM files although in most cases it is pointless as CRAM recalculates MD and NM tags on the fly. The one exception to this case is where both input and output CRAM files have been / are being created with the *no_ref* option.

OPTIONS:

-A When used jointly with **-r** this option overwrites the original base quality.

-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

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

-r Compute the BQ tag (without -A) or cap base quality by BAQ (with -A).

-E Extended BAQ calculation. This option trades specificity for sensitivity, though the effect is minor.

targetcut samtools targetcut [-Q minBaseQ] [-i inPenalty] [-0 em0] [-1 em1] [-2 em2] [-f ref] <in.bam>

This command identifies target regions by examining the continuity of read depth, computes haploid consensus sequences of targets and outputs a SAM with each sequence corresponding to a target. When option **-f** is in use, BAQ will be applied. This command is **only** designed for cutting fosmid clones from fosmid pool sequencing [Ref. Kitzman et al. (2010)].

phase samtools phase [-AF] [-k len] [-b prefix] [-q minLOD] [-Q minBaseQ] <in.bam>

Call and phase heterozygous SNPs.

OPTIONS:

-A Drop reads with ambiguous phase.

-b *STR*

Prefix of BAM output. When this option is in use, phase-0 reads will be saved in file **STR**.0.bam and phase-1 reads in **STR**.1.bam. Phase unknown reads will be randomly allocated to one of the two files. Chimeric reads with switch errors will be saved in **STR**.chimeric.bam. [null]

-F Do not attempt to fix chimeric reads.

-k *INT* Maximum length for local phasing. [13]

-q *INT* Minimum Phred-scaled LOD to call a heterozygote. [40]

-Q *INT* Minimum base quality to be used in het calling. [13]

depad samtools depad [-SsCu1] [-T ref.fa] [-o output] <in.bam>

Converts a BAM aligned against a padded reference to a BAM aligned against the depadded reference. The padded reference may contain verbatim "*" bases in it, but "*" bases are also counted in the reference numbering. This means that a sequence base-call aligned against a reference "*" is considered to be a cigar match ("M" or "X") operator (if the base-call is "A", "C", "G" or "T"). After depadding the reference "*" bases are deleted and such aligned sequence base-calls become insertions. Similarly transformations apply for deletions and padding cigar operations.

OPTIONS:

-S Ignored for compatibility with previous samtools versions. Previously this option was required if input was in SAM format, but now the correct format is automatically detected by examining the first few characters of input.

-s Output in SAM format. The default is BAM.

-C Output in CRAM format. The default is BAM.

-u Do not compress the output. Applies to either BAM or CRAM output format.

-1 Enable fastest compression level. Only works for BAM or CRAM output.

-T *FILE* Provides the padded reference file. Note that without this the @SQ line lengths will be incorrect, so for most use cases this option will be considered as mandatory.

-o *FILE* Specifies the output filename. By default output is sent to stdout.

help, **--help** Display a brief usage message listing the samtools commands available. If the name of a command is also given, e.g., **samtools help view**, the detailed usage message for that particular command is displayed.

--version Display the version numbers and copyright information for samtools and the important libraries used by samtools.

--version-only Display the full samtools version number in a machine-readable format.

GLOBAL OPTIONS

Several long-options are shared between multiple samtools subcommands: **--input-fmt**, **--input-fmt-options**, **--output-fmt**, **--output-fmt-options**, and **--reference**. The input format is typically auto-detected so specifying the format is usually unnecessary and the option is included for completeness. Note that not all subcommands have all options. Consult the subcommand help for more details.

Format strings recognised are "sam", "bam" and "cram". They may be followed by a comma separated list of options as *key* or *key=value*. See below for examples.

The **fmt-options** arguments accept either a single *option* or *option=value*. Note that some options only work on some file formats and only on read or write streams. If value is unspecified for a boolean option, the value is assumed to be 1. The valid options are as follows.

nthreads=*INT* Specifies the number of threads to use during encoding and/or decoding. For BAM this will be encoding only. In CRAM the threads are dynamically shared between encoder and decoder.

reference=*fasta_file*
Specifies a FASTA reference file for use in CRAM encoding or decoding. It usually is not required for decoding except in the situation of the MD5 not being obtainable via the REF_PATH or REF_CACHE environment variables.

decode_md=*0|1* CRAM input only; defaults to 1 (on). CRAM does not typically store MD and NM tags, preferring to generate them on the fly. This option controls this behaviour.

ignore_md5=*0|1* CRAM input only; defaults to 0 (off). When enabled, md5 checksum errors on the reference sequence and block checksum errors within CRAM are ignored. Use of this option is strongly discouraged.

required_fields=*bit-field*
CRAM input only; specifies which SAM columns need to be populated. By default all fields are used. Limiting the decode to specific columns can have significant performance gains. The bit-field is a numerical value constructed from the following table.

0x1	SAM_QNAME
0x2	SAM_FLAG
0x4	SAM_RNAME
0x8	SAM_POS
0x10	SAM_MAPQ
0x20	SAM_CIGAR
0x40	SAM_RNEXT
0x80	SAM_PNEXT
0x100	SAM_TLEN
0x200	SAM_SEQ
0x400	SAM_QUAL
0x800	SAM_AUX

0x1000 SAM_RGAUX

name_prefix=*string*
: CRAM input only; defaults to output filename. Any sequences with auto-generated read names will use *string* as the name prefix.

multi_seq_per_slice=*0|1*
: CRAM output only; defaults to 0 (off). By default CRAM generates one container per reference sequence, except in the case of many small references (such as a fragmented assembly).

version=*major.minor*
: CRAM output only. Specifies the CRAM version number. Acceptable values are "2.1" and "3.0".

seqs_per_slice=*INT*
: CRAM output only; defaults to 10000.

slices_per_container=*INT*
: CRAM output only; defaults to 1. The effect of having multiple slices per container is to share the compression header block between multiple slices. This is unlikely to have any significant impact unless the number of sequences per slice is reduced. (Together these two options control the granularity of random access.)

embed_ref=*0|1* CRAM output only; defaults to 0 (off). If 1, this will store portions of the reference sequence in each slice, permitting decode without having requiring an external copy of the reference sequence.

no_ref=*0|1* CRAM output only; defaults to 0 (off). If 1, sequences will be stored verbatim with no reference encoding. This can be useful if no reference is available for the file.

use_bzip2=*0|1* CRAM output only; defaults to 0 (off). Permits use of bzip2 in CRAM block compression.

use_lzma=*0|1* CRAM output only; defaults to 0 (off). Permits use of lzma in CRAM block compression.

lossy_names=*0|1*
: CRAM output only; defaults to 0 (off). If 1, templates with all members within the same CRAM slice will have their read names removed. New names will be automatically generated during decoding. Also see the **name_prefix** option.

For example:

```
samtools view --input-fmt-option decode_md=0
    --output-fmt cram,version=3.0 --output-fmt-option embed_ref
    --output-fmt-option seqs_per_slice=2000 -o foo.cram foo.bam
```

REFERENCE SEQUENCES

The CRAM format requires use of a reference sequence for both reading and writing.

When reading a CRAM the **@SQ** headers are interrogated to identify the reference sequence MD5sum (**M5:** tag) and the local reference sequence filename (**UR:** tag). Note that *http://* and *ftp://* based URLs in the UR: field are not used, but local fasta filenames (with or without *file://*) can be used.

To create a CRAM the **@SQ** headers will also be read to identify the reference sequences, but M5: and UR: tags may not be present. In this case the **-T** and **-t** options of samtools view may be used to specify the fasta or fasta.fai filenames respectively (provided the .fasta.fai file is also backed up by a .fasta file).

The search order to obtain a reference is:

Use any local file specified by the command line options (eg -T).

Look for MD5 via REF_CACHE environment variable.

Look for MD5 in each element of the REF_PATH environment variable.

Look for a local file listed in the UR: header tag.

ENVIRONMENT VARIABLES

HTS_PATH A colon-separated list of directories in which to search for HTSlib plugins. If $HTS_PATH starts or ends with a colon or contains a double colon (**::**), the built-in list of directories is searched at that point in the search.

If no HTS_PATH variable is defined, the built-in list of directories specified when HTSlib was built is used, which typically includes **/usr/local/libexec/htslib** and similar directories.

REF_PATH A colon separated (semi-colon on Windows) list of locations in which to look for sequences identified by their MD5sums. This can be either a list of directories or URLs. Note that if a URL is included then the colon in http:// and ftp:// and the optional port number will be treated as part of the URL and not a PATH field separator. For URLs, the text **%s** will be replaced by the MD5sum being read.

If no REF_PATH has been specified it will default to **http://www.ebi.ac.uk/ena/cram/md5/%s** and if REF_CACHE is also unset, it will be set to **$XDG_CACHE_HOME/hts-ref/%2s/%2s/%s**. If **$XDG_CACHE_HOME** is unset, **$HOME/.cache** (or a local system temporary directory if no home directory is found) will be used similarly.

REF_CACHE This can be defined to a single directory housing a local cache of references. Upon downloading a reference it will be stored in the location pointed to by REF_CACHE. When reading a reference it will be looked for in this directory before searching REF_PATH. To avoid many files being stored in the same directory, a pathname may be constructed using %*num*s and %s notation, consuming *num* characters of the MD5sum. For example **/local/ref_cache/%2s/%2s/%s** will create 2 nested subdirectories with the filenames in the deepest directory being the last 28 characters of the md5sum.

The REF_CACHE directory will be searched for before attempting to load via the REF_PATH search list. If no REF_PATH is defined, both REF_PATH and REF_CACHE will be automatically set (see above), but if REF_PATH is defined and REF_CACHE not then no local cache is used.

To aid population of the REF_CACHE directory a script **misc/seq_cache_populate.pl** is provided in the Samtools distribution. This takes a fasta file or a directory of fasta files and generates the MD5sum named files.

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.

- Convert a BAM file to a CRAM file using a local reference sequence.

```
samtools view -C -T ref.fa aln.bam > aln.cram
```

- 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:

```
samtools mpileup -uf ref.fa aln.bam | bcftools call -mv > var.raw.vcf
bcftools filter -s LowQual -e '%QUAL<20 || DP>100' var.raw.vcf  > var.flt.vcf
```

The **bcftools filter** command marks low quality sites and sites with the read depth exceeding a limit, which should be adjusted to about twice the average read depth (bigger read depths usually indicate problematic regions which are often enriched for artefacts). 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.
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.

- Generate the consensus sequence for one diploid individual:

```
samtools mpileup -uf ref.fa aln.bam | bcftools call -c | vcfutils.pl vcf2fq > cns.fq
```

- Phase one individual:

```
samtools calmd -AEur aln.bam ref.fa | samtools phase -b prefix - > phase.out
```

The **calmd** command is used to reduce false heterozygotes around INDELs.

- Dump BAQ applied alignment for other SNP callers:

```
samtools calmd -bAr 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.
- 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 MarkDuplicates which correctly handles these cases, although a little slower.

AUTHOR

Heng Li from the Sanger Institute wrote the original C version of samtools. Bob Handsaker from the Broad Institute implemented the BGZF library. James Bonfield from the Sanger Institute developed the CRAM implementation. 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

bcftools(1), *sam*(5), *tabix*(1)
Samtools website: <http://www.htslib.org/ (http://www.htslib.org/)>
File format specification of SAM/BAM,CRAM,VCF/BCF: <http://samtools.github.io/hts-specs (http://samtools.github.io/hts-specs)>
Samtools latest source: <https://github.com/samtools/samtools (https://github.com/samtools/samtools)>
HTSlib latest source: <https://github.com/samtools/htslib (https://github.com/samtools/htslib)>
Bcftools website: <http://samtools.github.io/bcftools (http://samtools.github.io/bcftools)>