This documentation refers to the **latest development version** of BCFtools which can be downloaded from github, see instructions.

Please refer to htslib.org for documentation for the latest **versioned release**.

Name

bcftools — utilities for variant calling and manipulating VCFs and BCFs.

Synopsis

bcftools [--version|--version-only] [--help] [*COMMAND*] [*OPTIONS*]

DESCRIPTION

BCFtools is a set of utilities that manipulate variant calls in the Variant Call Format (VCF) and its binary counterpart BCF. All commands work transparently with both VCFs and BCFs, both uncompressed and BGZF-compressed.

Most commands accept VCF, bgzipped VCF and BCF with filetype detected automatically even when streaming from a pipe. Indexed VCF and BCF will work in all situations. Un-indexed VCF and BCF and streams will work in most, but not all situations. In general, whenever multiple VCFs are read simultaneously, they must be indexed and therefore also compressed.

BCFtools is designed to work on a stream. It regards an input file "-" as the standard input (stdin) and outputs to the standard output (stdout). Several commands can thus be combined with Unix pipes.

VERSION

This manual page was last updated **2018-08-29 10:25 BST** and refers to bcftools git version **1.9-37-g90cca6a+**.

BCF1

The BCF1 format output by versions of samtools <= 0.1.19 is **not** compatible with this version of bcftools. To read BCF1 files one can use the view command from old versions of bcftools packaged with samtools versions <= 0.1.19 to convert to VCF, which can then be read by this version of bcftools.

```
    samtools-0.1.19/bcftools/bcftools view file.bcf1 | bcftools view
```

VARIANT CALLING

See *bcftools call* for variant calling from the output of the *samtools mpileup* command. In versions of samtools <= 0.1.19 calling was done with *bcftools view*. Users are now required to choose between the old samtools calling model (*-c/--consensus-caller*) and the new multiallelic calling model (*-m/--multiallelic-caller*). The multiallelic calling model is recommended for most tasks.

LIST OF COMMANDS

For a full list of available commands, run **bcftools** without arguments. For a full list of available options, run **bcftools** *COMMAND* without arguments.

- **annotate** .. edit VCF files, add or remove annotations
- **call** .. SNP/indel calling (former "view")
- **cnv** .. Copy Number Variation caller
- **concat** .. concatenate VCF/BCF files from the same set of samples
- **consensus** .. create consensus sequence by applying VCF variants
- **convert** .. convert VCF/BCF to other formats and back
- **csq** .. haplotype aware consequence caller
- **filter** .. filter VCF/BCF files using fixed thresholds
- **gtcheck** .. check sample concordance, detect sample swaps and contamination
- **index** .. index VCF/BCF
- **isec** .. intersections of VCF/BCF files
- **merge** .. merge VCF/BCF files files from non-overlapping sample sets
- **mpileup** .. multi-way pileup producing genotype likelihoods

- **norm** .. normalize indels
- **plugin** .. run user-defined plugin
- **polysomy** .. detect contaminations and whole-chromosome aberrations
- **query** .. transform VCF/BCF into user-defined formats
- **reheader** .. modify VCF/BCF header, change sample names
- **roh** .. identify runs of homo/auto-zygosity
- **sort** .. sort VCF/BCF files
- **stats** .. produce VCF/BCF stats (former vcfcheck)
- **view** .. subset, filter and convert VCF and BCF files

LIST OF SCRIPTS

Some helper scripts are bundled with the bcftools code.

- **plot-vcfstats** .. plots the output of **stats**

COMMANDS AND OPTIONS

Common Options

The following options are common to many bcftools commands. See usage for specific commands to see if they apply.

FILE
: Files can be both VCF or BCF, uncompressed or BGZF-compressed. The file "-" is interpreted as standard input. Some tools may require tabix- or CSI-indexed files.

-c, --collapse *snps|indels|both|all|some|none|id*
: Controls how to treat records with duplicate positions and defines compatible records across multiple input files. Here by "compatible" we mean records which should be considered as identical by the tools. For example, when performing line intersections, the desire may be to consider as identical all sites with matching positions (**bcftools isec -c** *all*), or only sites with matching variant type (**bcftools isec -c** *snps* **-c** *indels*), or only sites with all alleles identical (**bcftools isec -c** *none*).

  none
  : only records with identical REF and ALT alleles are compatible

  some
  : only records where some subset of ALT alleles match are compatible

  all
  : all records are compatible, regardless of whether the ALT alleles match or not. In the case of records with the same position, only the first will be considered and appear on output.

  snps
  : any SNP records are compatible, regardless of whether the ALT alleles match or not. For duplicate positions, only the first SNP record will be considered and appear on output.

  indels
  : all indel records are compatible, regardless of whether the REF and ALT alleles match or not. For duplicate positions, only the first indel record will be considered and appear on output.

  both
  : abbreviation of "**-c** *indels* **-c** *snps*"

  id
  : only records with identical ID column are compatible. Supported by **bcftools merge** only.

-f, --apply-filters *LIST*
: Skip sites where FILTER column does not contain any of the strings listed in *LIST*. For example, to include only sites which have no filters set, use **-f** *.,PASS*.

--no-version
: Do not append version and command line information to the output VCF header.

-o, --output *FILE*
: When output consists of a single stream, write it to *FILE* rather than to standard output, where it is written by default.

-O, --output-type *b|u|z|v*
: Output compressed BCF (*b*), uncompressed BCF (*u*), compressed VCF (*z*), uncompressed VCF (*v*). Use the -Ou option when piping between bcftools subcommands to speed up performance by removing unnecessary compression/decompression and VCF←→BCF conversion.

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
: Comma-separated list of regions, see also **-R, --regions-file**. Note that **-r** cannot be used in combination with **-R**.

-R, --regions-file *FILE*
: Regions can be specified either on command line or in a VCF, BED, or tab-delimited file (the default). The columns of the tab-delimited file are: CHROM, POS, and, optionally, POS_TO, where positions are 1-based and inclusive. The columns of the tab-delimited BED file are also CHROM, POS and POS_TO (trailing columns are ignored), but coordinates are 0-based, half-open. To indicate that a file be treated as BED rather than the 1-based tab-delimited file, the file must have the ".bed" or ".bed.gz" suffix (case-insensitive). Uncompressed files are stored in

memory, while bgzip-compressed and tabix-indexed region files are streamed. Note that sequence names must match exactly, "chr20" is not the same as "20". Also note that chromosome ordering in *FILE* will be respected, the VCF will be processed in the order in which chromosomes first appear in *FILE*. However, within chromosomes, the VCF will always be processed in ascending genomic coordinate order no matter what order they appear in *FILE*. Note that overlapping regions in *FILE* can result in duplicated out of order positions in the output. This option requires indexed VCF/BCF files. Note that **-R** cannot be used in combination with **-r**.

-s, --samples [^]*LIST*

Comma-separated list of samples to include or exclude if prefixed with "^". The sample order is updated to reflect that given on the command line. Note that in general tags such as INFO/AC, INFO/AN, etc are not updated to correspond to the subset samples. **bcftools view** is the exception where some tags will be updated (unless the **-I, --no-update** option is used; see **bcftools view** documentation). To use updated tags for the subset in another command one can pipe from **view** into that command. For example:

```
    bcftools view -Ou -s sample1,sample2 file.vcf | bcftools query -f %INFO/AC\t%INFO/AN\n
```

-S, --samples-file *FILE*

File of sample names to include or exclude if prefixed with "^". One sample per line. See also the note above for the **-s, --samples** option. The sample order is updated to reflect that given in the input file. The command **bcftools call** accepts an optional second column indicating ploidy (0, 1 or 2) or sex (as defined by **--ploidy**, for example "F" or "M"), and can parse also PED files. If the second column is not present, the sex "F" is assumed. With **bcftools call** **-C** *trio*, PED file is expected. File formats examples:

```
    sample1    1
    sample2    2
    sample3    2

  or

    sample1    M
    sample2    F
    sample3    F

  or a .ped file (here is shown a minimum working example, the first column is
  ignored and the last indicates sex: 1=male, 2=female)

    ignored daughterA fatherA motherA 2
    ignored sonB fatherB motherB 1
```

-t, --targets [^]*chr*|*chr:pos*|*chr:from-to*|*chr:from-*[,…]

Similar as **-r, --regions**, but the next position is accessed by streaming the whole VCF/BCF rather than using the tbi/csi index. Both **-r** and **-t** options can be applied simultaneously: **-r** uses the index to jump to a region and **-t** discards positions which are not in the targets. Unlike **-r**, targets can be prefixed with "^" to request logical complement. For example, "^X,Y,MT" indicates that sequences X, Y and MT should be skipped. Yet another difference between the two is that **-r** checks both start and end positions of indels, whereas **-t** checks start positions only. Note that **-t** cannot be used in combination with **-T**.

-T, --targets-file [^]*FILE*

Same **-t, --targets**, but reads regions from a file. Note that **-T** cannot be used in combination with **-t**.

With the **call -C** *alleles* command, third column of the targets file must be comma-separated list of alleles, starting with the reference allele. Note that the file must be compressed and index. Such a file can be easily created from a VCF using:

```
    bcftools query -f'%CHROM\t%POS\t%REF,%ALT\n' file.vcf | bgzip -c > als.tsv.gz && tabix -s1 -b2 -e2 als.tsv.gz
```

--threads *INT*

Number of output compression threads to use in addition to main thread. Only used when *--output-type* is *b* or *z*. Default: 0.

bcftools annotate *[OPTIONS] FILE*

Add or remove annotations.

-a, --annotations *file*

Bgzip-compressed and tabix-indexed file with annotations. The file can be VCF, BED, or a tab-delimited file with mandatory columns CHROM, POS (or, alternatively, FROM and TO), optional columns REF and ALT, and arbitrary number of annotation columns. BED files are expected to have the ".bed" or ".bed.gz" suffix (case-insensitive), otherwise a tab-delimited file is assumed. Note that in case of tab-delimited file, the coordinates POS, FROM and TO are one-based and inclusive. When REF and ALT are present, only matching VCF records will be annotated. When multiple ALT alleles are present in the annotation file (given as comma-separated list of alleles), at least one must match one of the alleles in the corresponding VCF record. Similarly, at least one alternate allele from a multi-allelic VCF record must be present in the annotation file. Note that flag types, such as "INFO/FLAG", can be annotated by including a field with the value "1" to set the flag, "0" to remove it, or "." to keep existing flags. See also **-c, --columns** and **-h, --header-lines**.

```
    # Sample annotation file with columns CHROM, POS, STRING_TAG, NUMERIC_TAG
    1  752566  SomeString      5
    1  798959  SomeOtherString 6
    # etc.
```

--collapse *snps*|*indels*|*both*|*all*|*some*|*none*

Controls how to match records from the annotation file to the target VCF. Effective only when **-a** is a VCF or BCF. See **Common Options** for more.

-c, --columns *list*

Comma-separated list of columns or tags to carry over from the annotation file (see also **-a, --annotations**). If the annotation file is not a VCF/BCF, *list* describes the columns of the annotation file and must include CHROM, POS (or, alternatively, FROM and TO), and optionally REF and ALT. Unused columns which should be ignored can be indicated by "-". If the annotation file is a VCF/BCF, only the edited columns/tags must be present and their order does not matter. The columns ID, QUAL, FILTER, INFO and FORMAT can be edited, where INFO tags can be written both as "INFO/TAG" or simply "TAG", and FORMAT tags can be written as "FORMAT/TAG" or "FMT/TAG". The imported VCF annotations can be renamed as "DST_TAG:=SRC_TAG" or "FMT/DST_TAG:=FMT/SRC_TAG". To carry over all INFO annotations, use "INFO". To add all INFO annotations except "TAG", use "^INFO/TAG". By default, existing values are replaced. To add annotations without overwriting existing values (that is, to add missing tags or add values to existing tags with missing values), use "+TAG" instead of "TAG". To append to existing values (rather than replacing or leaving untouched), use "=TAG" (instead of "TAG" or "+TAG"). To replace only existing values without modifying missing annotations, use "-TAG". If the annotation file is not a VCF/BCF, all new annotations must be defined via **-h, --header-lines**.

-e, --exclude *EXPRESSION*
exclude sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-h, --header-lines *file*
Lines to append to the VCF header, see also **-c, --columns** and **-a, --annotations**. For example:

```
##INFO=<ID=NUMERIC_TAG,Number=1,Type=Integer,Description="Example header line">
##INFO=<ID=STRING_TAG,Number=1,Type=String,Description="Yet another header line">
```

-I, --set-id [+]*FORMAT*
assign ID on the fly. The format is the same as in the **query** command (see below). By default all existing IDs are replaced. If the format string is preceded by "+", only missing IDs will be set. For example, one can use

```
bcftools annotate --set-id +'%CHROM\_%POS\_%REF\_%FIRST_ALT' file.vcf
```

-i, --include *EXPRESSION*
include only sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-k, --keep-sites
keep sites wich do not pass **-i** and **-e** expressions instead of discarding them

-m, --mark-sites *TAG*
annotate sites which are present ("+") or absent ("-") in the **-a** file with a new INFO/TAG flag

--no-version
see **Common Options**

-o, --output *FILE*
see **Common Options**

-O, --output-type *b|u|z|v*
see **Common Options**

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
see **Common Options**

-R, --regions-file *file*
see **Common Options**

--rename-chrs *file*
rename chromosomes according to the map in *file*, with "old_name new_name\n" pairs separated by whitespaces, each on a separate line.

-s, --samples [^]*LIST*
subset of samples to annotate, see also **Common Options**

-S, --samples-file *FILE*
subset of samples to annotate. If the samples are named differently in the target VCF and the **-a, --annotations** VCF, the name mapping can be given as "src_name dst_name\n", separated by whitespaces, each pair on a separate line.

--threads *INT*
see **Common Options**

-x, --remove *list*
List of annotations to remove. Use "FILTER" to remove all filters or "FILTER/SomeFilter" to remove a specific filter. Similarly, "INFO" can be used to remove all INFO tags and "FORMAT" to remove all FORMAT tags except GT. To remove all INFO tags except "FOO" and "BAR", use "^INFO/FOO,INFO/BAR" (and similarly for FORMAT and FILTER). "INFO" can be abbreviated to "INF" and "FORMAT" to "FMT".

Examples:

```
# Remove three fields
bcftools annotate -x ID,INFO/DP,FORMAT/DP file.vcf.gz

# Remove all INFO fields and all FORMAT fields except for GT and PL
bcftools annotate -x INFO,^FORMAT/GT,FORMAT/PL file.vcf

# Add ID, QUAL and INFO/TAG, not replacing TAG if already present
bcftools annotate -a src.bcf -c ID,QUAL,+TAG dst.bcf

# Carry over all INFO and FORMAT annotations except FORMAT/GT
bcftools annotate -a src.bcf -c INFO,^FORMAT/GT dst.bcf

# Annotate from a tab-delimited file with six columns (the fifth is ignored),
# first indexing with tabix. The coordinates are 1-based.
tabix -s1 -b2 -e2 annots.tab.gz
bcftools annotate -a annots.tab.gz -h annots.hdr -c CHROM,POS,REF,ALT,-,TAG file.vcf

# Annotate from a tab-delimited file with regions (1-based coordinates, inclusive)
tabix -s1 -b2 -e3 annots.tab.gz
bcftools annotate -a annots.tab.gz -h annots.hdr -c CHROM,FROM,TO,TAG inut.vcf

# Annotate from a bed file (0-based coordinates, half-closed, half-open intervals)
bcftools annotate -a annots.bed.gz -h annots.hdr -c CHROM,FROM,TO,TAG input.vcf
```

bcftools call *[OPTIONS] FILE*

This command replaces the former **bcftools view** caller. Some of the original functionality has been temporarily lost in the process of transition under htslib, but will be added back on popular demand. The original calling model can be invoked with the **-c** option.

File format options:

--no-version
: see **Common Options**

-o, --output *FILE*
: see **Common Options**

-O, --output-type *b|u|z|v*
: see **Common Options**

--ploidy *ASSEMBLY*[*?*]
: predefined ploidy, use *list* (or any other unused word) to print a list of all predefined assemblies. Append a question mark to print the actual definition. See also **--ploidy-file**.

--ploidy-file *FILE*
: ploidy definition given as a space/tab-delimited list of CHROM, FROM, TO, SEX, PLOIDY. The SEX codes are arbitrary and correspond to the ones used by **--samples-file**. The default ploidy can be given using the starred records (see below), unlisted regions have ploidy 2. The default ploidy definition is

```
X 1 60000 M 1
X 2699521 154931043 M 1
Y 1 59373566 M 1
Y 1 59373566 F 0
MT 1 16569 M 1
MT 1 16569 F 1
*  * *     M 2
*  * *     F 2
```

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
: see **Common Options**

-R, --regions-file *file*
: see **Common Options**

-s, --samples *LIST*
: see **Common Options**

-S, --samples-file *FILE*
: see **Common Options**

-t, --targets *LIST*
: see **Common Options**

-T, --targets-file *FILE*
: see **Common Options**

--threads *INT*
: see **Common Options**

Input/output options:

-A, --keep-alts
: output all alternate alleles present in the alignments even if they do not appear in any of the genotypes

-f, --format-fields *list*
: comma-separated list of FORMAT fields to output for each sample. Currently GQ and GP fields are supported. For convenience, the fields can be given as lower case letters.

-F, --prior-freqs *AN,AC*
: take advantage of prior knowledge of population allele frequencies. The workflow looks like this:

```
# Extract AN,AC values from an existing VCF, such 1000Genomes
bcftools query -f'%CHROM\t%POS\t%REF\t%ALT\t%AN\t%AC\n' 1000Genomes.bcf | bgzip -c > AFs.tab.gz

# If the tags AN,AC are not already present, use the +fill-AN-AC plugin
bcftools +fill-AN-AC 1000Genomes.bcf | bcftools query -f'%CHROM\t%POS\t%REF\t%ALT\t%AN\t%AC\n' | bgzip -c > AFs.ta
tabix -s1 -b2 -e2 AFs.tab.gz

# Create a VCF header description, here we name the tags REF_AN,REF_AC
cat AFs.hdr
##INFO=<ID=REF_AN,Number=1,Type=Integer,Description="Total number of alleles in reference genotypes">
##INFO=<ID=REF_AC,Number=A,Type=Integer,Description="Allele count in reference genotypes for each ALT allele">

# Now before calling, stream the raw mpileup output through `bcftools annotate` to add the frequencies
bcftools mpileup [...] -Ou | bcftools annotate -a AFs.tab.gz -h AFs.hdr -c CHROM,POS,REF,ALT,REF_AN,REF_AC -Ou | b
```

-g, --gvcf *INT*
: output also gVCF blocks of homozygous REF calls. The parameter *INT* is the minimum per-sample depth required to include a site in the non-variant block.

-i, --insert-missed *INT*
: output also sites missed by mpileup but present in **-T, --targets-file**.

-M, --keep-masked-ref
: output sites where REF allele is N

-V, --skip-variants *snps|indels*
: skip indel/SNP sites

-v, --variants-only

output variant sites only

Consensus/variant calling options:

-c, --consensus-caller
the original **samtools/bcftools** calling method (conflicts with **-m**)

-C, --constrain *alleles|trio*

alleles
call genotypes given alleles. See also **-T, --targets-file**.

trio
call genotypes given the father-mother-child constraint. See also **-s, --samples** and **-n, --novel-rate**.

-m, --multiallelic-caller
alternative modelfor multiallelic and rare-variant calling designed to overcome known limitations in **-c** calling model (conflicts with **-c**)

-n, --novel-rate *float*[,…]
likelihood of novel mutation for constrained **-C** *trio* calling. The trio genotype calling maximizes likelihood of a particular combination of genotypes for father, mother and the child P(F=i,M=j,C=k) = P(unconstrained) * Pn + P(constrained) * (1-Pn). By providing three values, the mutation rate Pn is set explicitly for SNPs, deletions and insertions, respectively. If two values are given, the first is interpreted as the mutation rate of SNPs and the second is used to calculate the mutation rate of indels according to their length as Pn=*float**exp(-a-b*len), where a=22.8689, b=0.2994 for insertions and a=21.9313, b=0.2856 for deletions [pubmed:23975140]. If only one value is given, the same mutation rate Pn is used for SNPs and indels.

-p, --pval-threshold *float*
with **-c**, accept variant if P(ref|D) < *float*.

-P, --prior *float*
expected substitution rate, or 0 to disable the prior. Only with **-m**.

-t, --targets *file|chr|chr:pos|chr:from-to|chr:from-*[,…]
see **Common Options**

-X, --chromosome-X
haploid output for male samples (requires PED file with **-s**)

-Y, --chromosome-Y
haploid output for males and skips females (requires PED file with **-s**)

bcftools cnv *[OPTIONS] FILE*

Copy number variation caller, requires a VCF annotated with the Illumina's B-allele frequency (BAF) and Log R Ratio intensity (LRR) values. The HMM considers the following copy number states: CN 2 (normal), 1 (single-copy loss), 0 (complete loss), 3 (single-copy gain).

General Options:

-c, --control-sample *string*
optional control sample name. If given, pairwise calling is performed and the **-P** option can be used

-f, --AF-file *file*
read allele frequencies from a tab-delimited file with the columns CHR,POS,REF,ALT,AF

-o, --output-dir *path*
output directory

-p, --plot-threshold *float*
call **matplotlib** to produce plots for chromosomes with quality at least *float*, useful for visual inspection of the calls. With **-p 0**, plots for all chromosomes will be generated. If not given, a **matplotlib** script will be created but not called.

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
see **Common Options**

-R, --regions-file *file*
see **Common Options**

-s, --query-sample *string*
query samply name

-t, --targets *LIST*
see **Common Options**

-T, --targets-file *FILE*
see **Common Options**

HMM Options:

-a, --aberrant *float*[,*float*]
fraction of aberrant cells in query and control. The hallmark of duplications and contaminations is the BAF value of heterozygous markers which is dependent on the fraction of aberrant cells. Sensitivity to smaller fractions of cells can be increased by setting **-a** to a lower value. Note however, that this comes at the cost of increased false discovery rate.

-b, --BAF-weight *float*
relative contribution from BAF

-d, --BAF-dev *float*[,*float*]
expected BAF deviation in query and control, i.e. the noise observed in the data.

-e, --err-prob *float*
: uniform error probability

-l, --LRR-weight *float*
: relative contribution from LRR. With noisy data, this option can have big effect on the number of calls produced. In truly random noise (such as in simulated data), the value should be set high (1.0), but in the presence of systematic noise when LRR are not informative, lower values result in cleaner calls (0.2).

-L, --LRR-smooth-win *int*
: reduce LRR noise by applying moving average given this window size

-O, --optimize *float*
: iteratively estimate the fraction of aberrant cells, down to the given fraction. Lowering this value from the default 1.0 to say, 0.3, can help discover more events but also increases noise

-P, --same-prob *float*
: the prior probability of the query and the control sample being the same. Setting to 0 calls both independently, setting to 1 forces the same copy number state in both.

-x, --xy-prob *float*
: the HMM probability of transition to another copy number state. Increasing this values leads to smaller and more frequent calls.

bcftools concat *[OPTIONS] FILE1 FILE2* […]

Concatenate or combine VCF/BCF files. All source files must have the same sample columns appearing in the same order. Can be used, for example, to concatenate chromosome VCFs into one VCF, or combine a SNP VCF and an indel VCF into one. The input files must be sorted by chr and position. The files must be given in the correct order to produce sorted VCF on output unless the **-a, --allow-overlaps** option is specified. With the --naive option, the files are concatenated without being recompressed, which is very fast but dangerous if the BCF headers differ.

-a, --allow-overlaps
: First coordinate of the next file can precede last record of the current file.

-c, --compact-PS
: Do not output PS tag at each site, only at the start of a new phase set block.

-d, --rm-dups *snps|indels|both|all|none*
: Output duplicate records of specified type present in multiple files only once. Requires **-a, --allow-overlaps**.

-D, --remove-duplicates
: Alias for **-d none**

-f, --file-list *FILE*
: Read file names from *FILE*, one file name per line.

-l, --ligate
: Ligate phased VCFs by matching phase at overlapping haplotypes

--no-version
: see Common Options

-n, --naive
: Concatenate VCF or BCF files without recompression. This is very fast but requires that all files are of the same type (all VCF or all BCF) and have the same headers. This is because all tags and chromosome names in the BCF body rely on the implicit order of the contig and tag definitions in the header. Currently no sanity checks are in place. Dangerous, use with caution.

-o, --output *FILE*
: see Common Options

-O, --output-type *b|u|z|v*
: see Common Options

-q, --min-PQ *INT*
: Break phase set if phasing quality is lower than *INT*

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
: see Common Options. Requires **-a, --allow-overlaps**.

-R, --regions-file *FILE*
: see Common Options. Requires **-a, --allow-overlaps**.

--threads *INT*
: see Common Options

bcftools consensus *[OPTIONS] FILE*

Create consensus sequence by applying VCF variants to a reference fasta file. By default, the program will apply all ALT variants to the reference fasta to obtain the consensus sequence. Using the **--sample** (and, optionally, **--haplotype**) option will apply genotype (haplotype) calls from FORMAT/GT. Note that the program does not act as a primitive variant caller and ignores allelic depth information, such as INFO/AD or FORMAT/AD. For that, consider using the **setGT** plugin.

-c, --chain *FILE*
: write a chain file for liftover

-e, --exclude *EXPRESSION*
: exclude sites for which *EXPRESSION* is true. For valid expressions see EXPRESSIONS.

-f, --fasta-ref *FILE*
: reference sequence in fasta format

-H, --haplotype *1|2|R|A|LR|LA|SR|SA*
: choose which allele from the FORMAT/GT field to use (the codes are case-insensitive):

1
: the first allele

2
: the second allele

R
: the REF allele (in heterozygous genotypes)

A
: the ALT allele (in heterozygous genotypes)

LR, LA
: the longer allele. If both have the same length, use the REF allele (LR), or the ALT allele (LA)

SR, SA
: the shorter allele. If both have the same length, use the REF allele (SR), or the ALT allele (SA)

```
This option requires *-s*, unless exactly one sample is present in the VCF
```

-i, --include *EXPRESSION*
: include only sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-I, --iupac-codes
: output variants in the form of IUPAC ambiguity codes

-m, --mask *FILE*
: BED file or TAB file with regions to be replaced with N. See discussion of **--regions-file** in **Common Options** for file format details.

-M, --missing *CHAR*
: instead of skipping the missing genotypes, output the character CHAR (e.g. "?")

-o, --output *FILE*
: write output to a file

-s, --sample *NAME*
: apply variants of the given sample

Examples:

```
    # Apply variants present in sample "NA001", output IUPAC codes for hets
    bcftools consensus -i -s NA001 -f in.fa in.vcf.gz > out.fa

    # Create consensus for one region. The fasta header lines are then expected
    # in the form ">chr:from-to".
    samtools faidx ref.fa 8:11870-11890 | bcftools consensus in.vcf.gz -o out.fa
```

bcftools convert *[OPTIONS] FILE*

VCF input options:

-e, --exclude *EXPRESSION*
: exclude sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-i, --include *EXPRESSION*
: include only sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
: see **Common Options**

-R, --regions-file *FILE*
: see **Common Options**

-s, --samples *LIST*
: see **Common Options**

-S, --samples-file *FILE*
: see **Common Options**

-t, --targets *LIST*
: see **Common Options**

-T, --targets-file *FILE*
: see **Common Options**

VCF output options:

--no-version
: see **Common Options**

-o, --output *FILE*
: see **Common Options**

-O, --output-type *b|u|z|v*

see Common Options

--threads *INT*
see Common Options

GEN/SAMPLE conversion:

-G, --gensample2vcf *prefix* or *gen-file*,*sample-file*
convert IMPUTE2 output to VCF. The second column must be of the form "CHROM:POS_REF_ALT" to detect possible strand swaps; IMPUTE2 leaves the first one empty ("--") when sites from reference panel are filled in. See also **-g** below.

-g, --gensample *prefix* or *gen-file*,*sample-file*
convert from VCF to gen/sample format used by IMPUTE2 and SHAPEIT. The columns of .gen file format are ID1,ID2,POS,A,B followed by three genotype probabilities P(AA), P(AB), P(BB) for each sample. In order to prevent strand swaps, the program uses IDs of the form "CHROM:POS_REF_ALT". For example:

```
.gen
----
1:111485207_G_A 1:111485207_G_A 111485207 G A 0 1 0 0 1 0
1:111494194_C_T 1:111494194_C_T 111494194 C T 0 1 0 0 0 1

.sample
-------
ID_1 ID_2 missing
0 0 0
sample1 sample1 0
sample2 sample2 0
```

--tag *STRING*
tag to take values for .gen file: GT,PL,GL,GP

--chrom
output chromosome in the first column instead of CHROM:POS_REF_ALT

--sex *FILE*
output sex column in the sample file. The FILE format is

```
  MaleSample    M
  FemaleSample  F
```

--vcf-ids
output VCF IDs in the second column instead of CHROM:POS_REF_ALT

gVCF conversion:

--gvcf2vcf
convert gVCF to VCF, expanding REF blocks into sites. Note that the **-i** and **-e** options work differently with this switch. In this situation the filtering expressions define which sites should be expanded and which sites should be left unmodified, but all sites are printed on output. In order to drop sites, stream first through bcftools view.

-f, --fasta-ref *file*
reference sequence in fasta format. Must be indexed with samtools faidx

HAP/SAMPLE conversion:

--hapsample2vcf *prefix* or *hap-file*,*sample-file*
convert from hap/sample format to VCF. The columns of .hap file are similar to .gen file above, but there are only two haplotype columns per sample. Note that the first column of the .hap file is expected to be in the form "CHR:POS_REF_ALT(_END)?", with the _END being optional for defining the INFO/END tag when ALT is a symbolic allele, for example:

```
.hap
----
1:111485207_G_A rsID1 111485207 G A 0 1 0 0
1:111494194_C_T rsID2 111494194 C T 0 1 0 0
1:111495231_A_<DEL>_111495784 rsID3 111495231 A <DEL> 0 0 1 0
```

--hapsample *prefix* or *hap-file*,*sample-file*
convert from VCF to hap/sample format used by IMPUTE2 and SHAPEIT. The columns of .hap file begin with ID,RSID,POS,REF,ALT. In order to prevent strand swaps, the program uses IDs of the form "CHROM:POS_REF_ALT".

--haploid2diploid
with **-h** option converts haploid genotypes to homozygous diploid genotypes. For example, the program will print *0 0* instead of the default *0 -*. This is useful for programs which do not handle haploid genotypes correctly.

--sex *FILE*
output sex column in the sample file. The FILE format is

```
  MaleSample    M
  FemaleSample  F
```

--vcf-ids
output VCF IDs instead of "CHROM:POS_REF_ALT" IDs

HAP/LEGEND/SAMPLE conversion:

-H, --haplegendsample2vcf *prefix* or *hap-file*,*legend-file*,*sample-file*
convert from hap/legend/sample format used by IMPUTE2 to VCF, see also **-h, --hapslegendsample** below.

-h, --haplegendsample *prefix* or *hap-file*,*legend-file*,*sample-file* convert from VCF to hap/legend/sample format used by IMPUTE2 and SHAPEIT. The columns of .legend file ID,POS,REF,ALT. In order to prevent strand swaps, the program uses IDs of the form "CHROM:POS_REF_ALT". The .sample file is quite basic at the moment with columns for population, group and sex expected to be edited by the user. For example:

```
.hap
-----
0 1 0 0 1 0
0 1 0 0 0 1

.legend
-------
id position a0 a1
1:111485207_G_A 111485207 G A
1:111494194_C_T 111494194 C T

.sample
-------
sample population group sex
sample1 sample1 sample1 2
sample2 sample2 sample2 2
```

--haploid2diploid
with **-h** option converts haploid genotypes to homozygous diploid genotypes. For example, the program will print *0 0* instead of the default *0 -*. This is useful for programs which do not handle haploid genotypes correctly.

--sex *FILE*
output sex column in the sample file. The FILE format is

```
  MaleSample    M
  FemaleSample  F
```

--vcf-ids
output VCF IDs instead of "CHROM:POS_REF_ALT" IDs

TSV conversion:

--tsv2vcf *file*
convert from TSV (tab-separated values) format (such as generated by 23andMe) to VCF. The input file fields can be tab- or space-delimited

-c, --columns *list*
comma-separated list of fields in the input file. In the current version, the fields CHROM, POS, ID, and AA are expected and can appear in arbitrary order, columns which should be ignored in the input file can be indicated by "-". The AA field lists alleles on the forward reference strand, for example "CC" or "CT" for diploid genotypes or "C" for haploid genotypes (sex chromosomes). Insertions and deletions are not supported yet, missing data can be indicated with "--".

-f, --fasta-ref *file*
reference sequence in fasta format. Must be indexed with samtools faidx

-s, --samples *LIST*
list of sample names. See **Common Options**

-S, --samples-file *FILE*
file of sample names. See **Common Options**

Example:

```
# Convert 23andme results into VCF
bcftools convert -c ID,CHROM,POS,AA -s SampleName -f 23andme-ref.fa --tsv2vcf 23andme.txt -Oz -o out.vcf.gz
```

bcftools csq *[OPTIONS] FILE*

Haplotype aware consequence predictor which correctly handles combined variants such as MNPs split over multiple VCF records, SNPs separated by an intron (but adjacent in the spliced transcript) or nearby frame-shifting indels which in combination in fact are not frame-shifting.

The output VCF is annotated with INFO/BCSQ and FORMAT/BCSQ tag (configurable with the **-c** option). The latter is a bitmask of indexes to INFO/BCSQ, with interleaved haplotypes. See the usage examples below for using the %TBCSQ converter in **query** for extracting a more human readable form from this bitmask. The contruction of the bitmask limits the number of consequences that can be referenced in the FORMAT/BCSQ tags. By default this is 16, but if more are required, see the **--ncsq** option.

The program requires on input a VCF/BCF file, the reference genome in fasta format (**--fasta-ref**) and genomic features in the GFF3 format downloadable from the Ensembl website (**--gff-annot**), and outputs an annotated VCF/BCF file. Currently, only Ensembl GFF3 files are supported.

By default, the input VCF should be phased. If phase is unknown, or only partially known, the **--phase** option can be used to indicate how to handle unphased data. Alternatively, haplotype aware calling can be turned off with the **--local-csq** option.

If conflicting (overlapping) variants within one haplotype are detected, a warning will be emitted and predictions will be based on only the first variant in the analysis.

Symbolic alleles are not supported. They will remain unannotated in the output VCF and are ignored for the prediction analysis.

-c, --custom-tag *STRING*
use this custom tag to store consequences rather than the default BCSQ tag

-e, --exclude *EXPRESSION*
exclude sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-f, --fasta-ref *FILE*
: reference sequence in fasta format (required)

--force
: run even if some sanity checks fail. Currently the option allows to skip transcripts in malformatted GFFs with incorrect phase

-g, --gff-annot *FILE*
: GFF3 annotation file (required), such as ftp://ftp.ensembl.org/pub/current_gff3/homo_sapiens. An example of a minimal working GFF file:

```
# The program looks for "CDS", "exon", "three_prime_UTR" and "five_prime_UTR" lines,
# looks up their parent transcript (determined from the "Parent=transcript:" attribute),
# the gene (determined from the transcript's "Parent=gene:" attribute), and the biotype
# (the most interesting is "protein_coding").
#
# Attributes required for
#   gene lines:
#   - ID=gene:<gene_id>
#   - biotype=<biotype>
#   - Name=<gene_name>      [optional]
#
#   transcript lines:
#   - ID=transcript:<transcript_id>
#   - Parent=gene:<gene_id>
#   - biotype=<biotype>
#
#   other lines (CDS, exon, five_prime_UTR, three_prime_UTR):
#   - Parent=transcript:<transcript_id>
#
# Supported biotypes:
#   - see the function gff_parse_biotype() in bcftools/csq.c

1   ignored_field  gene            21  2148  .   -   .   ID=gene:GeneId;biotype=protein_coding;Name=GeneName
1   ignored_field  transcript      21  2148  .   -   .   ID=transcript:TranscriptId;Parent=gene:GeneId;biotype=prot
1   ignored_field  three_prime_UTR 21  2054  .   -   .   Parent=transcript:TranscriptId
1   ignored_field  exon            21  2148  .   -   .   Parent=transcript:TranscriptId
1   ignored_field  CDS             21  2148  .   -   1   Parent=transcript:TranscriptId
1   ignored_field  five_prime_UTR  210 2148  .   -   .   Parent=transcript:TranscriptId
```

-i, --include *EXPRESSION*
: include only sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-l, --local-csq
: switch off haplotype-aware calling, run localized predictions considering only one VCF record at a time

-n, --ncsq *INT*
: maximum number of consequences to consider per site. The INFO/BCSQ column includes all consequences, but only the first *INT* will be referenced by the FORMAT/BCSQ fields. The default value is 16 which corresponds to one integer per diploid sample. Note that increasing the value leads to increased memory and is rarely necessary.

-o, --output *FILE*
: see **Common Options**

-O, --output-type *b|t|u|z|v*
: see **Common Options**. In addition, a custom tab-delimited plain text output can be printed (*t*).

-p, --phase *a|m|r|R|s*
: how to handle unphased heterozygous genotypes:

a
: take GTs as is, create haplotypes regardless of phase (0/1 → 0|1)

m
: merge all GTs into a single haplotype (0/1 → 1, 1/2 → 1)

r
: require phased GTs, throw an error on unphased heterozygous GTs

R
: create non-reference haplotypes if possible (0/1 → 1|1, 1/2 → 1|2)

s
: skip unphased heterozygous GTs

-q, --quiet
: suppress warning messages

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
: see **Common Options**

-R, --regions-file *FILE*
: see **Common Options**

-s, --samples *LIST*
: samples to include or "-" to apply all variants and ignore samples

-S, --samples-file *FILE*
: see **Common Options**

-t, --targets *LIST*
: see **Common Options**

-T, --targets-file *FILE*
: see **Common Options**

Examples:

```
    # Basic usage
    bcftools csq -f hs37d5.fa -g Homo_sapiens.GRCh37.82.gff3.gz in.vcf -Ob -o out.bcf

    # Extract the translated haplotype consequences. The following TBCSQ variations
    # are recognised:
    #   %TBCSQ    .. print consequences in all haplotypes in separate columns
    #   %TBCSQ{0} .. print the first haplotype only
    #   %TBCSQ{1} .. print the second haplotype only
    #   %TBCSQ{*} .. print a list of unique consquences present in either haplotype
    bcftools query -f'[%CHROM\t%POS\t%SAMPLE\t%TBCSQ\n]' out.bcf
```

Examples of BCSQ annotation:

```
    # Two separate VCF records at positions 2:122106101 and 2:122106102
    # change the same codon. This UV-induced C>T dinucleotide mutation
    # has been annotated fully at the position 2:122106101 with
    #   - consequence type
    #   - gene name
    #   - ensembl transcript ID
    #   - coding strand (+ fwd, - rev)
    #   - amino acid position (in the coding strand orientation)
    #   - list of corresponding VCF variants
    # The annotation at the second position gives the position of the full
    # annotation
    BCSQ=missense|CLASP1|ENST00000545861|-|1174P>1174L|122106101G>A+122106102G>A
    BCSQ=@122106101

    # A frame-restoring combination of two frameshift insertions C>CG and T>TGG
    BCSQ=@46115084
    BCSQ=inframe_insertion|COPZ2|ENST00000006101|-|18AGRGP>18AQAGGP|46115072C>CG+46115084T>TGG

    # Stop gained variant
    BCSQ=stop_gained|C2orf83|ENST00000264387|-|141W>141*|228476140C>T

    # The consequence type of a variant downstream from a stop are prefixed with *
    BCSQ=*missense|PER3|ENST00000361923|+|1028M>1028T|7890117T>C
```

bcftools filter *[OPTIONS] FILE*

Apply fixed-threshold filters.

-e, --exclude *EXPRESSION*
exclude sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-g, --SnpGap *INT*
filter SNPs within *INT* base pairs of an indel. The following example demonstrates the logic of **--SnpGap** *3* applied on a deletion and an insertion:

```
The SNPs at positions 1 and 7 are filtered, positions 0 and 8 are not:
         0123456789
    ref  .G.GT..G..
    del  .A.G-..A..
Here the positions 1 and 6 are filtered, 0 and 7 are not:
         0123-456789
    ref  .G.G-..G..
    ins  .A.GT..A..
```

-G, --IndelGap *INT*
filter clusters of indels separated by *INT* or fewer base pairs allowing only one to pass. The following example demonstrates the logic of **--IndelGap** *2* applied on a deletion and an insertion:

```
The second indel is filtered:
         012345678901
    ref  .GT.GT..GT..
    del  .G-.G-..G-..
And similarly here, the second is filtered:
         01 23 456 78
    ref  .A-.A-..A-..
    ins  .AT.AT..AT..
```

-i, --include *EXPRESSION*
include only sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-m, --mode [*+x*]
define behaviour at sites with existing FILTER annotations. The default mode replaces existing filters of failed sites with a new FILTER string while leaving sites which pass untouched when non-empty and setting to "PASS" when the FILTER string is absent. The "+" mode appends new FILTER strings of failed sites instead of replacing them. The "x" mode resets filters of sites which pass to "PASS". Modes "+" and "x" can both be set.

--no-version
see **Common Options**

-o, --output *FILE*
see **Common Options**

-O, --output-type *b|u|z|v*
see **Common Options**

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
see **Common Options**

-R, --regions-file *file*
see **Common Options**

-s, --soft-filter *STRING*|+
: annotate FILTER column with *STRING* or, with +, a unique filter name generated by the program ("Filter%d").

-S, --set-GTs .|*0*
: set genotypes of failed samples to missing value (.) or reference allele (*0*)

-t, --targets *chr*|*chr:pos*|*chr:from-to*|*chr:from-*[,…]
: see **Common Options**

-T, --targets-file *file*
: see **Common Options**

--threads *INT*
: see **Common Options**

bcftools gtcheck [*OPTIONS*] [-g *genotypes.vcf.gz*] *query.vcf.gz*

Checks sample identity. The program can operate in two modes. If the **-g** option is given, the identity of the **-s** sample from *query.vcf.gz* is checked against the samples in the **-g** file. Without the **-g** option, multi-sample cross-check of samples in *query.vcf.gz* is performed.

-a, --all-sites
: output for all sites

-c, --cluster *FLOAT*,*FLOAT*
: min inter- and max intra-sample error [0.23,-0.3]

```
The first "min" argument controls the typical error rate in multiplexed
runs ("lanelets") from the same sample. Lanelets with error rate less
than this will always be considered as coming from the same sample.
The second "max" argument is the reverse: lanelets with error rate
greater than the absolute value of this parameter will always be
considered as different samples. When the value is negative, the cutoff
may be heuristically lowered by the clustering engine. If positive, the
value is interpreted as a fixed cutoff.
```

-g, --genotypes *genotypes.vcf.gz*
: reference genotypes to compare against

-G, --GTs-only *INT*
: use genotypes (GT) instead of genotype likelihoods (PL). When set to 1, reported discordance is the number of non-matching GTs, otherwise the number *INT* is interpreted as phred-scaled likelihood of unobserved genotypes.

-H, --homs-only
: consider only genotypes which are homozygous in both *genotypes* and *query* VCF. This may be useful with low coverage data.

-p, --plot *PREFIX*
: produce plots

-r, --regions *chr*|*chr:pos*|*chr:from-to*|*chr:from-*[,…]
: see **Common Options**

-R, --regions-file *file*
: see **Common Options**

-s, --query-sample *STRING*
: query sample in *query.vcf.gz*. By default, the first sample is checked.

-S, --target-sample *STRING*
: target sample in the **-g** file, used only for plotting, not for analysis

-t, --targets *file*
: see **Common Options**

-T, --targets-file *file*
: see **Common Options**

Output files format:

CN, Discordance
: Pairwise discordance for all sample pairs is calculated as

```
        \sum_s { min_G { PL_a(G) + PL_b(G) } },
```

where the sum runs over all sites *s* and *G* is the the most likely genotype shared by both samples *a* and *b*. When PL field is not present, a constant value *99* is used for the unseen genotypes. With **-G**, the value *1* can be used instead; the discordance value then gives exactly the number of differing genotypes.

ERR, error rate
: Pairwise error rate calculated as number of differences divided by the total number of comparisons.

CLUSTER, TH, DOT
: In presence of multiple samples, related samples and outliers can be identified by clustering samples by error rate. A simple hierarchical clustering based on minimization of standard deviation is used. This is useful to detect sample swaps, for example in situations where one sample has been sequenced in multiple runs.

bcftools index [*OPTIONS*] *in.bcf*|*in.vcf.gz*

Creates index for bgzip compressed VCF/BCF files for random access. CSI (coordinate-sorted index) is created by default. The CSI format supports indexing of chromosomes up to length 2^31. TBI (tabix index) index files, which support chromosome lengths up to 2^29, can be created by using the *-t*/*--tbi* option or using the *tabix* program packaged with htslib. When loading an index file, bcftools will try the CSI first and then the TBI.

Indexing options:

-c, --csi
generate CSI-format index for VCF/BCF files [default]

-f, --force
overwrite index if it already exists

-m, --min-shift *INT*
set minimal interval size for CSI indices to 2^INT; default: 14

-o, --output-file *FILE*
output file name. If not set, then the index will be created using the input file name plus a *.csi* or *.tbi* extension

-t, --tbi
generate TBI-format index for VCF files

--threads *INT*
see **Common Options**

Stats options:

-n, --nrecords
print the number of records based on the CSI or TBI index files

-s, --stats
Print per contig stats based on the CSI or TBI index files. Output format is three tab-delimited columns listing the contig name, contig length (. if unknown) and number of records for the contig. Contigs with zero records are not printed.

bcftools isec [*OPTIONS*] *A.vcf.gz B.vcf.gz* […]

Creates intersections, unions and complements of VCF files. Depending on the options, the program can output records from one (or more) files which have (or do not have) corresponding records with the same position in the other files.

-c, --collapse *snps|indels|both|all|some|none*
see **Common Options**

-C, --complement
output positions present only in the first file but missing in the others

-e, --exclude -|*EXPRESSION*
exclude sites for which *EXPRESSION* is true. If **-e** (or **-i**) appears only once, the same filtering expression will be applied to all input files. Otherwise, **-e** or **-i** must be given for each input file. To indicate that no filtering should be performed on a file, use "-" in place of *EXPRESSION*, as shown in the example below. For valid expressions see **EXPRESSIONS**.

-f, --apply-filters *LIST*
see **Common Options**

-i, --include *EXPRESSION*
include only sites for which *EXPRESSION* is true. See discussion of **-e, --exclude** above.

-n, --nfiles [+-=]*INT*|~*BITMAP*
output positions present in this many (=), this many or more (+), this many or fewer (-), or the exact same (~) files

-o, --output *FILE*
see **Common Options**. When several files are being output, their names are controlled via **-p** instead.

-O, --output-type *b|u|z|v*
see **Common Options**

-p, --prefix *DIR*
if given, subset each of the input files accordingly. See also **-w**.

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
see **Common Options**

-R, --regions-file *file*
see **Common Options**

-t, --targets *chr|chr:pos|chr:from-to|chr:from-*[,…]
see **Common Options**

-T, --targets-file *file*
see **Common Options**

-w, --write *LIST*
list of input files to output given as 1-based indices. With **-p** and no **-w**, all files are written.

Examples:

Create intersection and complements of two sets saving the output in dir/*

```
    bcftools isec -p dir A.vcf.gz B.vcf.gz
```

Filter sites in A (require INFO/MAF>=0.01) and B (require INFO/dbSNP) but not in C, and create an intersection, including only sites which appear in at least two of the files after filters have been applied

```
    bcftools isec -e'MAF<0.01' -i'dbSNP=1' -e- A.vcf.gz B.vcf.gz C.vcf.gz -n +2 -p dir
```

Extract and write records from A shared by both A and B using exact allele match

```
    bcftools isec -p dir -n=2 -w1 A.vcf.gz B.vcf.gz
```

Extract records private to A or B comparing by position only

```
    bcftools isec -p dir -n-1 -c all A.vcf.gz B.vcf.gz
```

Print a list of records which are present in A and B but not in C and D

```
    bcftools isec -n~1100 -c all A.vcf.gz B.vcf.gz C.vcf.gz D.vcf.gz
```

bcftools merge [*OPTIONS*] *A.vcf.gz B.vcf.gz* […]

Merge multiple VCF/BCF files from non-overlapping sample sets to create one multi-sample file. For example, when merging file *A.vcf.gz* containing samples *S1*, *S2* and *S3* and file *B.vcf.gz* containing samples *S3* and *S4*, the output file will contain four samples named *S1*, *S2*, *S3*, *2:S3* and *S4*.

Note that it is responsibility of the user to ensure that the sample names are unique across all files. If they are not, the program will exit with an error unless the option **--force-samples** is given. The sample names can be also given explicitly using the **--print-header** and **--use-header** options.

Note that only records from different files can be merged, never from the same file. For "vertical" merge take a look at bcftools concat or bcftools norm **-m** instead.

--force-samples
: if the merged files contain duplicate samples names, proceed anyway. Duplicate sample names will be resolved by prepending index of the file as it appeared on the command line to the conflicting sample name (see *2:S3* in the above example).

--print-header
: print only merged header and exit

--use-header *FILE*
: use the VCF header in the provided text *FILE*

-0 --missing-to-ref
: assume genotypes at missing sites are 0/0

-f, --apply-filters *LIST*
: see Common Options

-F, --filter-logic *x*|+
: Set the output record to PASS if any of the inputs is PASS (*x*), or apply all filters (+), which is the default.

-g, --gvcf -|*FILE*
: merge gVCF blocks, INFO/END tag is expected. If the reference fasta file *FILE* is not given and the dash (-) is given, unknown reference bases generated at gVCF block splits will be substituted with N's. The **--gvcf** option uses the following default INFO rules: **-i QS:sum,MinDP:min,I16:sum,IDV:max,IMF:max**.

-i, --info-rules -|*TAG:METHOD*[,…]
: Rules for merging INFO fields (scalars or vectors) or - to disable the default rules. *METHOD* is one of *sum*, *avg*, *min*, *max*, *join*. Default is *DP:sum,DP4:sum* if these fields exist in the input files. Fields with no specified rule will take the value from the first input file. The merged QUAL value is currently set to the maximum. This behaviour is not user controllable at the moment.

-l, --file-list *FILE*
: Read file names from *FILE*, one file name per line.

-m, --merge *snps*|*indels*|*both*|*all*|*none*|*id*
: The option controls what types of multiallelic records can be created:

```
-m none   ..  no new multiallelics, output multiple records instead
-m snps   ..  allow multiallelic SNP records
-m indels ..  allow multiallelic indel records
-m both   ..  both SNP and indel records can be multiallelic
-m all    ..  SNP records can be merged with indel records
-m id     ..  merge by ID
```

--no-version
: see Common Options

-o, --output *FILE*
: see Common Options

-O, --output-type *b*|*u*|*z*|*v*
: see Common Options

-r, --regions *chr*|*chr:pos*|*chr:from-to*|*chr:from-*[,…]
: see Common Options

-R, --regions-file *file*

see Common Options

--threads *INT*
see Common Options

bcftools mpileup [*OPTIONS*] -f *ref.fa* *in.bam* [*in2.bam* […]]

Generate VCF or BCF containing genotype likelihoods for one or multiple alignment (BAM or CRAM) files. This is based on the original **samtools mpileup** command (with the **-v** or **-g** options) producing genotype likelihoods in VCF or BCF format, but not the textual pileup output. The **mpileup** command was transferred to bcftools in order to avoid errors resulting from use of incompatible versions of samtools and bcftools when using in the mpileup+bcftools call pipeline.

Individuals are identified from the SM tags in the @RG header lines. Multiple individuals can be pooled in one alignment file, also one individual can be separated into multiple files. If sample identifiers are absent, each input file is regarded as one sample.

Note that there are two orthogonal ways to specify locations in the input file; via **-r** *region* and **-t** *positions*. 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 -t 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. Also note that the **-r** option can be much slower than **-t** with many regions and can require more memory when multiple regions and many alignment files are processed.

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 alignment 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 posi- tion, 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 the original **samtools mpileup** command had a minimum value of *8000/n* where *n* was the number of input files given to mpileup. This means that in **samtools mpileup** the default was highly likely to be increased and the **-d** parameter would have an effect only once above the cross-sample minimum of 8000. This behavior was problematic when working with a combination of single- and multi-sample bams, therefore in **bcftools mpileup** the user is given the full control (and responsibility), and an informative message is printed instead [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**. Reference is required by default unless the **--no-reference** option is set [null]

--no-reference
Do not require the **--fasta-ref** option.

-G, --read-groups *FILE*
list of read groups to include or exclude if prefixed with "^". One read group per line. This file can also be used to assign new sample names to read groups by giving the new sample name as a second white-space-separated field, like this: "read_group_id new_sample_name". If the read group name is not unique, also the bam file name can be included: "read_group_id file_name sample_name". If all reads from the alignment file should be treated as a single sample, the asterisk symbol can be used: "* file_name sample_name". Alignments without a read group ID can be matched with "?". **NOTE:** The meaning of **bcftools mpileup -G** is the opposite of **samtools mpileup -G**.

```
RG_ID_1
RG_ID_2  SAMPLE_A
RG_ID_3  SAMPLE_A
RG_ID_4  SAMPLE_B
RG_ID_5  FILE_1.bam  SAMPLE_A
RG_ID_6  FILE_2.bam  SAMPLE_A
*        FILE_3.bam  SAMPLE_C
?        FILE_3.bam  SAMPLE_D
```

-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, --regions *CHR|CHR:POS|CHR:FROM-TO|CHR:FROM-*[,…]
Only generate mpileup output in given regions. Requires the alignment files to be indexed. If used in conjunction with -l then considers the intersection; see Common Options

-R, --regions-file *FILE*

As for **-r, --regions**, but regions read from FILE; see **Common Options**

--ignore-RG
Ignore RG tags. Treat all reads in one alignment file 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]

-s, --samples *LIST*
: list of sample names. See **Common Options**

-S, --samples-file *FILE*
: file of sample names to include or exclude if prefixed with "^". One sample per line. This file can also be used to rename samples by giving the new sample name as a second white-space-separated column, like this: "old_name new_name". If a sample name contains spaces, the spaces can be escaped using the backslash character, for example "Not\ a\ good\ sample\ name".

-t, --targets *LIST*
: see **Common Options**

-T, --targets-file *FILE*
: see **Common Options**

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

Output options

-a, --annotate *LIST*
: Comma-separated list of FORMAT and INFO tags to output. (case-insensitive, the "FORMAT/" prefix is optional, and use "?" to list available annotations on the command line) [null]:

```
*FORMAT/AD* .. Allelic depth (Number=R,Type=Integer)
*FORMAT/ADF* .. Allelic depths on the forward strand (Number=R,Type=Integer)
*FORMAT/ADR* .. Allelic depths on the reverse strand (Number=R,Type=Integer)
*FORMAT/DP* .. Number of high-quality bases (Number=1,Type=Integer)
*FORMAT/SP* .. Phred-scaled strand bias P-value (Number=1,Type=Integer)

*INFO/AD* .. Total allelic depth (Number=R,Type=Integer)
*INFO/ADF* .. Total allelic depths on the forward strand (Number=R,Type=Integer)
*INFO/ADR* .. Total allelic depths on the reverse strand (Number=R,Type=Integer)

*FORMAT/DV* .. Deprecated in favor of FORMAT/AD;
        Number of high-quality non-reference bases, (Number=1,Type=Integer)
*FORMAT/DP4* .. Deprecated in favor of FORMAT/ADF and FORMAT/ADR;
        Number of high-quality ref-forward, ref-reverse,
        alt-forward and alt-reverse bases (Number=4,Type=Integer)
*FORMAT/DPR* .. Deprecated in favor of FORMAT/AD;
        Number of high-quality bases for each observed allele (Number=R,Type=Integer)
*INFO/DPR* .. Deprecated in favor of INFO/AD;
        Number of high-quality bases for each observed allele (Number=R,Type=Integer)
```

-g, --gvcf *INT*[,…]
: output gVCF blocks of homozygous REF calls, with depth (DP) ranges specified by the list of integers. For example, passing *5,15* will group sites into two types of gVCF blocks, the first with minimum per-sample DP from the interval [5,15) and the latter with minimum depth 15 or more. In this example, sites with minimum per-sample depth less than 5 will be printed as separate records, outside of gVCF blocks.

--no-version
: see **Common Options**

-o, --output *FILE*
: Write 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**.)

-O, --output-type *b|u|z|v*
: see **Common Options**

--threads *INT*
: see **Common Options**

Options for SNP/INDEL genotype likelihood computation

-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-sample 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. (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**.) [40]

-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]

Examples:

Call SNPs and short INDELs, then mark low quality sites and sites with the read depth exceeding a limit. (The read depth should be adjusted to about twice the average read depth as higher 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 for BWA-backtrack alignments, but may not other aligners.

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

bcftools norm [*OPTIONS*] *file.vcf.gz*

Left-align and normalize indels, check if REF alleles match the reference, split multiallelic sites into multiple rows; recover multiallelics from multiple rows. Left-alignment and normalization will only be applied if the **--fasta-ref** option is supplied.

-c, --check-ref *e|w|x|s*
what to do when incorrect or missing REF allele is encountered: exit (*e*), warn (*w*), exclude (*x*), or set/fix (*s*) bad sites. The *w* option can be combined with *x* and *s*. Note that *s* can swap alleles and will update genotypes (GT) and AC counts, but will not attempt to fix PL or other fields. Also note, and this cannot be stressed enough, that *s* will NOT fix strand issues in your VCF, do NOT use it for that purpose!!! (Instead see http://samtools.github.io/bcftools/howtos/plugin.af-dist.html and http://samtools.github.io/bcftools/howtos/plugin.fixref.html.)

-d, --rm-dup *snps|indels|both|all|none*
If a record is present multiple times, output only the first instance, see **--collapse** in **Common Options**.

-D, --remove-duplicates
If a record is present in multiple files, output only the first instance. Alias for **-d none**, deprecated.

-f, --fasta-ref *FILE*
reference sequence. Supplying this option will turn on left-alignment and normalization, however, see also the **--do-not-normalize** option below.

-m, --multiallelics -|+[*snps|indels|both|any*]
split multiallelic sites into biallelic records (**-**) or join biallelic sites into multiallelic records (**+**). An optional type string can follow which controls variant types which should be split or merged together: If only SNP records should be split or merged, specify *snps*; if both SNPs and indels should be merged separately into two records, specify *both*; if SNPs and indels should be merged into a single record, specify *any*.

--no-version
see **Common Options**

-N, --do-not-normalize
the *-c s* option can be used to fix or set the REF allele from the reference *-f*. The *-N* option will not turn on indel normalisation as the *-f* option normally implies

-o, --output *FILE*
see **Common Options**

-O, --output-type *b|u|z|v*
see **Common Options**

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
see **Common Options**

-R, --regions-file *file*
see **Common Options**

-s, --strict-filter
when merging (*-m+*), merged site is PASS only if all sites being merged PASS

-t, --targets *LIST*
see **Common Options**

-T, --targets-file *FILE*
see **Common Options**

--threads *INT*
see **Common Options**

-w, --site-win *INT*
maximum distance between two records to consider when locally sorting variants which changed position during the realignment

bcftools [plugin *NAME*|+*NAME*] *[OPTIONS] FILE — [PLUGIN OPTIONS]*

A common framework for various utilities. The plugins can be used the same way as normal commands only their name is prefixed with "+". Most plugins accept two types of parameters: general options shared by all plugins followed by a separator, and a list of plugin-specific options. There are some exceptions to this rule, some plugins do not accept the common options and implement their own parameters. Therefore please pay attention to the usage examples that each plugin comes with.

VCF input options:

-e, --exclude *EXPRESSION*
: exclude sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-i, --include *EXPRESSION*
: include only sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-r, --regions *chr*|*chr:pos*|*chr:from-to*|*chr:from-*[,…]
: see **Common Options**

-R, --regions-file *file*
: see **Common Options**

-t, --targets *chr*|*chr:pos*|*chr:from-to*|*chr:from-*[,…]
: see **Common Options**

-T, --targets-file *file*
: see **Common Options**

VCF output options:

--no-version
: see **Common Options**

-o, --output *FILE*
: see **Common Options**

-O, --output-type *b*|*u*|*z*|*v*
: see **Common Options**

--threads *INT*
: see **Common Options**

Plugin options:

-h, --help
: list plugin's options

-l, --list-plugins
: List all available plugins.

: By default, appropriate system directories are searched for installed plugins. You can override this by setting the BCFTOOLS_PLUGINS environment variable to a colon-separated list of directories to search. If BCFTOOLS_PLUGINS begins with a colon, ends with a colon, or contains adjacent colons, the system directories are also searched at that position in the list of directories.

-v, --verbose
: print debugging information to debug plugin failure

-V, --version
: print version string and exit

List of plugins coming with the distribution:

GTisec
: count genotype intersections across all possible sample subsets in a vcf file

GTsubset
: output only sites where the requested samples all exclusively share a genotype

ad-bias
: find positions with wildly varying ALT allele frequency (Fisher test on FMT/AD)

af-dist
: collect AF deviation stats and GT probability distribution given AF and assuming HWE

check-ploidy
: check if ploidy of samples is consistent for all sites

check-sparsity
: print samples without genotypes in a region or chromosome

color-chrs
: color shared chromosomal segments, requires trio VCF with phased GTs

counts
: a minimal plugin which counts number of SNPs, Indels, and total number of sites.

dosage
: print genotype dosage. By default the plugin searches for PL, GL and GT, in that order.

fill-AN-AC
: fill INFO fields AN and AC.

fill-from-fasta
: fill INFO or REF field based on values in a fasta file

fill-tags
: set INFO tags AF, AC, AC_Hemi, AC_Hom, AC_Het, AN, HWE, MAF, NS

fix-ploidy
: sets correct ploidy

fixref
: determine and fix strand orientation

frameshifts
: annotate frameshift indels

guess-ploidy
: determine sample sex by checking genotype likelihoods (GL,PL) or genotypes (GT) in the non-PAR region of chrX.

impute-info
: add imputation information metrics to the INFO field based on selected FORMAT tags

isecGT
: compare two files and set non-identical genotypes to missing

mendelian
: count Mendelian consistent / inconsistent genotypes.

missing2ref
: sets missing genotypes ("./.") to ref allele ("0/0" or "0|0")

prune
: prune sites by missingness or linkage disequilibrium

setGT
: general tool to set genotypes according to rules requested by the user

tag2tag
: convert between similar tags, such as GL and GP

trio-switch-rate
: calculate phase switch rate in trio samples, children samples must have phased GTs.

add-variantkey
: add VariantKey INFO fields VKX and RSX

variantkey-hex
: generate unsorted VariantKey-RSid index files in hexadecimal format

allele-length
: count the frequency of the length of REF, ALT and REF+ALT

Examples:

```
# List options common to all plugins
bcftools plugin

# List available plugins
bcftools plugin -l

# Run a plugin
bcftools plugin counts in.vcf

# Run a plugin using the abbreviated "+" notation
bcftools +counts in.vcf

# The input VCF can be streamed just like in other commands
cat in.vcf | bcftools +counts

# Print usage information of plugin "dosage"
bcftools +dosage -h

# Replace missing genotypes with 0/0
bcftools +missing2ref in.vcf

# Replace missing genotypes with 0|0
bcftools +missing2ref in.vcf -- -p
```

Plugins troubleshooting:

Things to check if your plugin does not show up in the **bcftools plugin -l** output:

- Run with the **-v** option for verbose output: **bcftools plugin -lv**
- Does the environment variable BCFTOOLS_PLUGINS include the correct path?

Plugins API:

```
// Short description used by 'bcftools plugin -l'
const char *about(void);

// Longer description used by 'bcftools +name -h'
const char *usage(void);

// Called once at startup, allows initialization of local variables.
// Return 1 to suppress normal VCF/BCF header output, -1 on critical
// errors, 0 otherwise.
int init(int argc, char **argv, bcf_hdr_t *in_hdr, bcf_hdr_t *out_hdr);

// Called for each VCF record, return NULL to suppress the output
bcf1_t *process(bcf1_t *rec);

// Called after all lines have been processed to clean up
void destroy(void);
```

bcftools polysomy [*OPTIONS*] *file.vcf.gz*

Detect number of chromosomal copies in VCFs annotates with the Illumina's B-allele frequency (BAF) values. Note that this command is not compiled in by default, see the section **Optional Compilation with GSL** in the INSTALL file for help.

General options:

-o, --output-dir *path*
output directory

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
see Common Options

-R, --regions-file *file*
see Common Options

-s, --sample *string*
sample name

-t, --targets *LIST*
see Common Options

-T, --targets-file *FILE*
see Common Options

-v, --verbose
verbose debugging output which gives hints about the thresholds and decisions made by the program. Note that the exact output can change between versions.

Algorithm options:

-b, --peak-size *float*
the minimum peak size considered as a good match can be from the interval [0,1] where larger is stricter

-c, --cn-penalty *float*
a penalty for increasing copy number state. How this works: multiple peaks are always a better fit than a single peak, therefore the program prefers a single peak (normal copy number) unless the absolute deviation of the multiple peaks fit is significantly smaller. Here the meaning of "significant" is given by the *float* from the interval [0,1] where larger is stricter.

-f, --fit-th *float*
threshold for goodness of fit (normalized absolute deviation), smaller is stricter

-i, --include-aa
include also the AA peak in CN2 and CN3 evaluation. This usually requires increasing **-f**.

-m, --min-fraction *float*
minimum distinguishable fraction of aberrant cells. The experience shows that trustworthy are estimates of 20% and more.

-p, --peak-symmetry *float*
a heuristics to filter failed fits where the expected peak symmetry is violated. The *float* is from the interval [0,1] and larger is stricter

bcftools query [*OPTIONS*] *file.vcf.gz* [*file.vcf.gz* […]]

Extracts fields from VCF or BCF files and outputs them in user-defined format.

-e, --exclude *EXPRESSION*
exclude sites for which *EXPRESSION* is true. For valid expressions see EXPRESSIONS.

-f, --format *FORMAT*
learn by example, see below

-H, --print-header
print header

-i, --include *EXPRESSION*
include only sites for which *EXPRESSION* is true. For valid expressions see EXPRESSIONS.

-l, --list-samples
list sample names and exit

-o, --output *FILE*
see Common Options

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
see Common Options

-R, --regions-file *file*
see Common Options

-s, --samples *LIST*
see Common Options

-S, --samples-file *FILE*
see Common Options

-t, --targets *chr|chr:pos|chr:from-to|chr:from-*[,…]

see **Common Options**

-T, --targets-file *file*
see **Common Options**

-u, --allow-undef-tags
do not throw an error if there are undefined tags in the format string, print "." instead

-v, --vcf-list *FILE*
process multiple VCFs listed in the file

Format:

```
%CHROM          The CHROM column (similarly also other columns: POS, ID, REF, ALT, QUAL, FILTER)
%INFO/TAG       Any tag in the INFO column
%TYPE           Variant type (REF, SNP, MNP, INDEL, BND, OTHER)
%MASK           Indicates presence of the site in other files (with multiple files)
%TAG{INT}       Curly brackets to subscript vectors (0-based)
%FIRST_ALT      Alias for %ALT{0}
[]              Format fields must be enclosed in brackets to loop over all samples
%GT             Genotype (e.g. 0/1)
%TBCSQ          Translated FORMAT/BCSQ. See the csq command above for explanation and examples.
%TGT            Translated genotype (e.g. C/A)
%IUPACGT        Genotype translated to IUPAC ambiguity codes (e.g. M instead of C/A)
%LINE           Prints the whole line
%SAMPLE         Sample name
%POS0           POS in 0-based coordinates
%END            End position of the REF allele
%END0           End position of the REF allele in 0-based cordinates
\n              new line
\t              tab character
Everything else is printed verbatim.
```

Examples:

```
# Print chromosome, position, ref allele and the first alternate allele
bcftools query -f '%CHROM  %POS  %REF  %ALT{0}\n' file.vcf.gz
# Similar to above, but use tabs instead of spaces, add sample name and genotype
bcftools query -f '%CHROM\t%POS\t%REF\t%ALT[\t%SAMPLE=%GT]\n' file.vcf.gz
# Print FORMAT/GT fields followed by FORMAT/GT fields
bcftools query -f 'GQ:[ %GQ] \t GT:[ %GT]\n' file.vcf
# Make a BED file: chr, pos (0-based), end pos (1-based), id
bcftools query -f'%CHROM\t%POS0\t%END\t%ID\n' file.bcf
# Print only samples with alternate (non-reference) genotypes
bcftools query -f'[%CHROM:%POS %SAMPLE %GT\n]' -i'GT="alt"' file.bcf
# Print all samples at sites with at least one alternate genotype
bcftools view -i'GT="alt"' file.bcf -Ou | bcftools query -f'[%CHROM:%POS %SAMPLE %GT\n]'
```

bcftools reheader [*OPTIONS*] *file.vcf.gz*

Modify header of VCF/BCF files, change sample names.

-f, --fai *FILE*
add to the header contig names and their lengths from the provided fasta index file (.fai). Lengths of existing contig lines will be updated and contig lines not present in the fai file will be removed

-h, --header *FILE*
new VCF header

-o, --output *FILE*
see **Common Options**

-s, --samples *FILE*
new sample names, one name per line, in the same order as they appear in the VCF file. Alternatively, only samples which need to be renamed can be listed as "old_name new_name\n" pairs separated by whitespaces, each on a separate line. If a sample name contains spaces, the spaces can be escaped using the backslash character, for example "Not\ a\ good\ sample\ name".

--threads *INT*
see **Common Options**

bcftools roh [*OPTIONS*] *file.vcf.gz*

A program for detecting runs of homo/autozygosity. Only bi-allelic sites are considered.

The HMM model:

```
Notation:
  D  = Data, AZ = autozygosity, HW = Hardy-Weinberg (non-autozygosity),
  f  = non-ref allele frequency

Emission probabilities:
  oAZ = P_i(D|AZ) = (1-f)*P(D|RR) + f*P(D|AA)
  oHW = P_i(D|HW) = (1-f)^2 * P(D|RR) + f^2 * P(D|AA) + 2*f*(1-f)*P(D|RA)

Transition probabilities:
  tAZ = P(AZ|HW)  .. from HW to AZ, the -a parameter
  tHW = P(HW|AZ)  .. from AZ to HW, the -H parameter

  ci  = P_i(C)  .. probability of cross-over at site i, from genetic map
  AZi = P_i(AZ) .. probability of site i being AZ/non-AZ, scaled so that AZi+HWi = 1
  HWi = P_i(HW)
```

```
    P_{i+1}(AZ) = oAZ * max[(1 - tAZ * ci) * AZ{i-1} , tAZ * ci * (1-AZ{i-1})]
    P_{i+1}(HW) = oHW * max[(1 - tHW * ci) * (1-AZ{i-1}) , tHW * ci * AZ{i-1}]
```

General Options:

--AF-dflt *FLOAT*
: in case allele frequency is not known, use the *FLOAT*. By default, sites where allele frequency cannot be determined, or is 0, are skipped.

--AF-tag *TAG*
: use the specified INFO tag *TAG* as an allele frequency estimate instead of the default AC and AN tags. Sites which do not have *TAG* will be skipped.

--AF-file *FILE*
: Read allele frequencies from a tab-delimited file containing the columns: CHROM\tPOS\tREF,ALT\tAF. The file can be compressed with **bgzip** and indexed with tabix -s1 -b2 -e2. Sites which are not present in the *FILE* or have different reference or alternate allele will be skipped. Note that such a file can be easily created from a VCF using:

```
    bcftools query -f'%CHROM\t%POS\t%REF,%ALT\t%INFO/TAG\n' file.vcf | bgzip -c > freqs.tab.gz
```

-b, --buffer-size *INT*[,*INT*]
: when the entire many-sample file cannot fit into memory, a sliding buffer approach can be used. The first value is the number of sites to keep in memory. If negative, it is interpreted as the maximum memory to use, in MB. The second, optional, value sets the number of overlapping sites. The default overlap is set to roughly 1% of the buffer size.

-e, --estimate-AF *FILE*
: estimate the allele frequency by recalculating INFO/AC and INFO/AN on the fly, using the specified *TAG* which can be either FORMAT/GT ("GT") or FORMAT/PL ("PL"). If *TAG* is not given, "GT" is assumed. Either all samples ("-") or samples listed in *FILE* will be included. For example, use "PL,-" to estimate AF from FORMAT/PL of all samples. If neither **-e** nor the other **--AF-...** options are given, the allele frequency is estimated from AC and AN counts which are already present in the INFO field.

--exclude *EXPRESSION*
: exclude sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-G, --GTs-only *FLOAT*
: use genotypes (FORMAT/GT fields) ignoring genotype likelihoods (FORMAT/PL), setting PL of unseen genotypes to *FLOAT*. Safe value to use is 30 to account for GT errors.

--include *EXPRESSION*
: include only sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-I, --skip-indels
: skip indels as their genotypes are usually enriched for errors

-m, --genetic-map *FILE*
: genetic map in the format required also by IMPUTE2. Only the first and third column are used (position and Genetic_Map(cM)). The *FILE* can chromosome name.

-M, --rec-rate *FLOAT*
: constant recombination rate per bp. In combination with **--genetic-map**, the **--rec-rate** parameter is interpreted differently, as *FLOAT*-fold increase of transition probabilities, which allows the model to become more sensitive yet still account for recombination hotspots. Note that also the range of the values is therefore different in both cases: normally the parameter will be in the range (1e-3,1e-9) but with **--genetic-map** it will be in the range (10,1000).

-o, --output *FILE*
: Write output to the *FILE*, by default the output is printed on stdout

-O, --output-type *s*|*r*[*z*]
: Generate per-site output (*s*) or per-region output (*r*). By default both types are printed and the output is uncompressed. Add *z* for a compressed output.

-r, --regions *chr*|*chr:pos*|*chr:from-to*|*chr:from-*[,...]
: see **Common Options**

-R, --regions-file *file*
: see **Common Options**

-s, --samples *LIST*
: see **Common Options**

-S, --samples-file *FILE*
: see **Common Options**

-t, --targets *chr*|*chr:pos*|*chr:from-to*|*chr:from-*[,...]
: see **Common Options**

-T, --targets-file *file*
: see **Common Options**

HMM Options:

-a, --hw-to-az *FLOAT*
: P(AZ|HW) transition probability from AZ (autozygous) to HW (Hardy-Weinberg) state

-H, --az-to-hw *FLOAT*
: P(HW|AZ) transition probability from HW to AZ state

-V, --viterbi-training *FLOAT*
: estimate HMM parameters using Baum-Welch algorithm, using the convergence threshold *FLOAT*, e.g. 1e-10 (experimental)

bcftools sort [*OPTIONS*] file.bcf

-m, --max-mem *FLOAT*[*kMG*]
: Maximum memory to use. Approximate, affects the number of temporary files written to the disk. Note that if the command fails at this step because of too many open files, your system limit on the number of open files ("ulimit") may need to be increased.

-o, --output *FILE*
: see **Common Options**

-O, --output-type *b|u|z|v*
: see **Common Options**

-T, --temp-dir *DIR*
: Use this directory to store temporary files

bcftools stats [*OPTIONS*] *A.vcf.gz* [*B.vcf.gz*]

Parses VCF or BCF and produces text file stats which is suitable for machine processing and can be plotted using **plot-vcfstats**. When two files are given, the program generates separate stats for intersection and the complements. By default only sites are compared, **-s**/**-S** must given to include also sample columns. When one VCF file is specified on the command line, then stats by non-reference allele frequency, depth distribution, stats by quality and per-sample counts, singleton stats, etc. are printed. When two VCF files are given, then stats such as concordance (Genotype concordance by non-reference allele frequency, Genotype concordance by sample, Non-Reference Discordance) and correlation are also printed. Per-site discordance (PSD) is also printed in **--verbose** mode.

--af-bins *LIST|FILE*
: comma separated list of allele frequency bins (e.g. 0.1,0.5,1) or a file listing the allele frequency bins one per line (e.g. 0.1\n0.5\n1)

--af-tag *TAG*
: allele frequency INFO tag to use for binning. By default the allele frequency is estimated from AC/AN, if available, or directly from the genotypes (GT) if not.

-1, --1st-allele-only
: consider only the 1st alternate allele at multiallelic sites

-c, --collapse *snps|indels|both|all|some|none*
: see **Common Options**

-d, --depth *INT,INT,INT*
: ranges of depth distribution: min, max, and size of the bin

--debug
: produce verbose per-site and per-sample output

-e, --exclude *EXPRESSION*
: exclude sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-E, --exons *file.gz*
: tab-delimited file with exons for indel frameshifts statistics. The columns of the file are CHR, FROM, TO, with 1-based, inclusive, positions. The file is BGZF-compressed and indexed with tabix

```
    tabix -s1 -b2 -e3 file.gz
```

-f, --apply-filters *LIST*
: see **Common Options**

-F, --fasta-ref *ref.fa*
: faidx indexed reference sequence file to determine INDEL context

-i, --include *EXPRESSION*
: include only sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-I, --split-by-ID
: collect stats separately for sites which have the ID column set ("known sites") or which do not have the ID column set ("novel sites").

-r, --regions *chr|chr:pos|chr:from-to|chr:from-*[,…]
: see **Common Options**

-R, --regions-file *file*
: see **Common Options**

-s, --samples *LIST*
: see **Common Options**

-S, --samples-file *FILE*
: see **Common Options**

-t, --targets *chr|chr:pos|chr:from-to|chr:from-*[,…]
: see **Common Options**

-T, --targets-file *file*
: see **Common Options**

-u, --user-tstv *<TAG[:min:max:n]>*
: collect Ts/Tv stats for any tag using the given binning [0:1:100]

-v, --verbose

produce verbose per-site and per-sample output

bcftools view [*OPTIONS*] *file.vcf.gz* [*REGION* […]]

View, subset and filter VCF or BCF files by position and filtering expression. Convert between VCF and BCF. Former **bcftools subset**.

Output options

-G, --drop-genotypes
drop individual genotype information (after subsetting if **-s** option is set)

-h, --header-only
output the VCF header only

-H, --no-header
suppress the header in VCF output

-l, --compression-level [*0-9*]
compression level. 0 stands for uncompressed, 1 for best speed and 9 for best compression.

--no-version
see **Common Options**

-O, --output-type *b*|*u*|*z*|*v*
see **Common Options**

-o, --output-file *FILE*: output file name. If not present, the default is to print to standard output (stdout).

-r, --regions *chr*|*chr:pos*|*chr:from-to*|*chr:from-*[,…]
see **Common Options**

-R, --regions-file *file*
see **Common Options**

-t, --targets *chr*|*chr:pos*|*chr:from-to*|*chr:from-*[,…]
see **Common Options**

-T, --targets-file *file*
see **Common Options**

--threads *INT*
see **Common Options**

Subset options:

-a, --trim-alt-alleles
trim alternate alleles not seen in subset. Type A, G and R INFO and FORMAT fields will also be trimmed

--force-samples
only warn about unknown subset samples

-I, --no-update
do not (re)calculate INFO fields for the subset (currently INFO/AC and INFO/AN)

-s, --samples *LIST*
see **Common Options**

-S, --samples-file *FILE*
see **Common Options**

Filter options:

Note that filter options below dealing with counting the number of alleles will, for speed, first check for the values of AC and AN in the INFO column to avoid parsing all the genotype (FORMAT/GT) fields in the VCF. This means that a filter like *--min-af 0.1* will be based ‘AC/AN’ where AC and AN come from either INFO/AC and INFO/AN if available or FORMAT/GT if not. It will not filter on another field like INFO/AF. The *--include* and *--exclude* filter expressions should instead be used to explicitly filter based on fields in the INFO column, e.g. *--exclude AF<0.1*.

-c, --min-ac *INT*[*:nref*|*:alt1*|*:minor*|*:major*|:'nonmajor']
minimum allele count (INFO/AC) of sites to be printed. Specifying the type of allele is optional and can be set to non-reference (*nref*, the default), 1st alternate (*alt1*), the least frequent (*minor*), the most frequent (*major*) or sum of all but the most frequent (*nonmajor*) alleles.

-C, --max-ac *INT*[*:nref*|*:alt1*|*:minor*|:'major'|:'nonmajor']
maximum allele count (INFO/AC) of sites to be printed. Specifying the type of allele is optional and can be set to non-reference (*nref*, the default), 1st alternate (*alt1*), the least frequent (*minor*), the most frequent (*major*) or sum of all but the most frequent (*nonmajor*) alleles.

-e, --exclude *EXPRESSION*
exclude sites for which *EXPRESSION* is true. For valid expressions see **EXPRESSIONS**.

-f, --apply-filters *LIST*
see **Common Options**

-g, --genotype [^][*hom*|*het*|*miss*]
include only sites with one or more homozygous (*hom*), heterozygous (*het*) or missing (*miss*) genotypes. When prefixed with ^, the logic is reversed; thus ^*het* excludes sites with heterozygous genotypes.

-i, --include *EXPRESSION*

include sites for which *EXPRESSION* is true. For valid expressions see EXPRESSIONS.

-k, --known
: print known sites only (ID column is not ".")

-m, --min-alleles *INT*
: print sites with at least *INT* alleles listed in REF and ALT columns

-M, --max-alleles *INT*
: print sites with at most *INT* alleles listed in REF and ALT columns. Use **-m2 -M2 -v snps** to only view biallelic SNPs.

-n, --novel
: print novel sites only (ID column is ".")

-p, --phased
: print sites where all samples are phased. Haploid genotypes are considered phased. Missing genotypes considered unphased unless the phased bit is set.

-P, --exclude-phased
: exclude sites where all samples are phased

-q, --min-af *FLOAT*[*:nref*|*:alt1*|*:minor*|*:major*|*:nonmajor*]
: minimum allele frequency (INFO/AC / INFO/AN) of sites to be printed. Specifying the type of allele is optional and can be set to non-reference (*nref*, the default), 1st alternate (*alt1*), the least frequent (*minor*), the most frequent (*major*) or sum of all but the most frequent (*nonmajor*) alleles.

-Q, --max-af *FLOAT*[*:nref*|*:alt1*|*:minor*|*:major*|*:nonmajor*]
: maximum allele frequency (INFO/AC / INFO/AN) of sites to be printed. Specifying the type of allele is optional and can be set to non-reference (*nref*, the default), 1st alternate (*alt1*), the least frequent (*minor*), the most frequent (*major*) or sum of all but the most frequent (*nonmajor*) alleles.

-u, --uncalled
: print sites without a called genotype

-U, --exclude-uncalled
: exclude sites without a called genotype

-v, --types *snps*|*indels*|*mnps*|*other*
: comma-separated list of variant types to select. Site is selected if any of the ALT alleles is of the type requested. Types are determined by comparing the REF and ALT alleles in the VCF record not INFO tags like INFO/INDEL or INFO/VT. Use **--include** to select based on INFO tags.

-V, --exclude-types *snps*|*indels*|*mnps*|*ref*|*bnd*|*other*
: comma-separated list of variant types to exclude. Site is excluded if any of the ALT alleles is of the type requested. Types are determined by comparing the REF and ALT alleles in the VCF record not INFO tags like INFO/INDEL or INFO/VT. Use **--exclude** to exclude based on INFO tags.

-x, --private
: print sites where only the subset samples carry an non-reference allele. Requires **--samples** or **--samples-file**.

-X, --exclude-private
: exclude sites where only the subset samples carry an non-reference allele

bcftools help [*COMMAND*] | bcftools --help [*COMMAND*]

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

bcftools [*--version*|*-v*]

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

bcftools [*--version-only*]

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

EXPRESSIONS

These filtering expressions are accepted by most of the commands.

Valid expressions may contain:

- numerical constants, string constants, file names (this is currently supported only to filter by the ID column)

```
1, 1.0, 1e-4
"String"
@file_name
```

- arithmetic operators

```
+,*,-,/
```

- comparison operators

```
== (same as =), >, >=, <=, <, !=
```

- regex operators "~" and its negation "!~". The expressions are case sensitive unless "/i" is added.

```
INFO/HAYSTACK ~ "needle"
INFO/HAYSTACK ~ "NEEDless/i"
```

- parentheses

```
(, )
```

- logical operators. See also the examples below and the filtering tutorial about the distinction between "&&" vs "&" and "||" vs "|".

```
&&,  &, ||,  |
```

- INFO tags, FORMAT tags, column names

```
INFO/DP or DP
FORMAT/DV, FMT/DV, or DV
FILTER, QUAL, ID, CHROM, POS, REF, ALT[0]
```

- 1 (or 0) to test the presence (or absence) of a flag

```
FlagA=1 && FlagB=0
```

- "." to test missing values

```
DP=".", DP!=".", ALT="."
```

- missing genotypes can be matched regardless of phase and ploidy (".|.", "./.", ".") using these expressions

```
GT~"\.", GT!~"\."
```

- missing genotypes can be matched including the phase and ploidy (".|.", "./.", ".") using these expressions

```
GT=".|.", GT="./.", GT="."
```

- sample genotype: reference (haploid or diploid), alternate (hom or het, haploid or diploid), missing genotype, homozygous, heterozygous, haploid, ref-ref hom, alt-alt hom, ref-alt het, alt-alt het, haploid ref, haploid alt (case-insensitive)

```
GT="ref"
GT="alt"
GT="mis"
GT="hom"
GT="het"
GT="hap"
GT="RR"
GT="AA"
GT="RA" or GT="AR"
GT="Aa" or GT="aA"
GT="R"
GT="A"
```

- TYPE for variant type in REF,ALT columns (indel,snp,mnp,ref,bnd,other). Use the regex operator "\~" to require at least one allele of the given type or the equal sign "=" to require that all alleles are of the given type. Compare

```
TYPE="snp"
TYPE~"snp"
TYPE!="snp"
TYPE!~"snp"
```

- array subscripts (0-based), "*" for any element, "-" to indicate a range. Note that for querying FORMAT vectors, the colon ":" can be used to select a sample and an element of the vector, as shown in the examples below

```
INFO/AF[0] > 0.3             .. first AF value bigger than 0.3
FORMAT/AD[0:0] > 30          .. first AD value of the first sample bigger than 30
FORMAT/AD[0:1]               .. first sample, second AD value
FORMAT/AD[1:0]               .. second sample, first AD value
DP4[*] == 0                  .. any DP4 value
FORMAT/DP[0]   > 30          .. DP of the first sample bigger than 30
FORMAT/DP[1-3] > 10          .. samples 2-4
FORMAT/DP[1-]  < 7           .. all samples but the first
FORMAT/DP[0,2-4] > 20        .. samples 1, 3-5
FORMAT/AD[0:1]               .. first sample, second AD field
FORMAT/AD[0:*], AD[0:] or AD[0] .. first sample, any AD field
FORMAT/AD[*:1] or AD[:1]        .. any sample, second AD field
(DP4[0]+DP4[1])/(DP4[2]+DP4[3]) > 0.3
CSQ[*] ~ "missense_variant.*deleterious"
```

- with many samples it can be more practical to provide a file with sample names, one sample name per line

```
GT[@samples.txt]="het" & binom(AD)<0.01
```

- function on FORMAT tags (over samples) and INFO tags (over vector fields)

```
MAX, MIN, AVG, SUM, STRLEN, ABS, COUNT
```

- two-tailed binomial test. Note that for N=0 the test evaluates to a missing value and when FORMAT/GT is used to determine the vector indices, it evaluates to 1 for homozygous genotypes.

```
binom(FMT/AD)                .. GT can be used to determine the correct index
binom(AD[0],AD[1])           .. or the fields can be given explicitly
```

- variables calculated on the fly if not present: number of alternate alleles; number of samples; count of alternate alleles; minor allele count (similar to AC but is always smaller than 0.5); frequency of alternate alleles (AF=AC/AN); frequency of minor alleles (MAF=MAC/AN); number of alleles in called genotypes; number of samples with missing genotype; fraction of samples with missing genotype;

```
N_ALT, N_SAMPLES, AC, MAC, AF, MAF, AN, N_MISSING, F_MISSING
```

- the number (N_PASS) or fraction (F_PASS) of samples which pass the expression

```
N_PASS(GQ>90 & GT!="mis") > 90
F_PASS(GQ>90 & GT!="mis") > 0.9
```

- custom perl filtering. Note that this command is not compiled in by default, see the section **Optional Compilation with Perl** in the INSTALL file for help and misc/demo-flt.pl for a working example. The demo defined the perl subroutine "severity" which can be invoked from the command line as follows:

```
perl:path/to/script.pl; perl.severity(INFO/CSQ) > 3
```

Notes:

- String comparisons and regular expressions are case-insensitive
- Variables and function names are case-insensitive, but not tag names. For example, "qual" can be used instead of "QUAL", "strlen()" instead of "STRLEN()" , but not "dp" instead of "DP".
- When querying multiple values, all elements are tested and the OR logic is used on the result. For example, when querying "TAG=1,2,3,4", it will be evaluated as follows:

```
-i 'TAG[*]=1'   .. true, the record will be printed
-i 'TAG[*]!=1'  .. true
-e 'TAG[*]=1'   .. false, the record will be discarded
-e 'TAG[*]!=1'  .. false
-i 'TAG[0]=1'   .. true
-i 'TAG[0]!=1'  .. false
-e 'TAG[0]=1'   .. false
-e 'TAG[0]!=1'  .. true
```

Examples:

```
MIN(DV)>5
MIN(DV/DP)>0.3
MIN(DP)>10 & MIN(DV)>3
FMT/DP>10  & FMT/GQ>10 .. both conditions must be satisfied within one sample
FMT/DP>10 && FMT/GQ>10 .. the conditions can be satisfied in different samples
QUAL>10 |  FMT/GQ>10   .. true for sites with QUAL>10 or a sample with GQ>10, but selects only samples with GQ>10
QUAL>10 || FMT/GQ>10   .. true for sites with QUAL>10 or a sample with GQ>10, plus selects all samples at such sites
TYPE="snp" && QUAL>=10 && (DP4[2]+DP4[3] > 2)
COUNT(GT="hom")=0
MIN(DP)>35 && AVG(GQ)>50
ID=@file       .. selects lines with ID present in the file
ID!=@~/file    .. skip lines with ID present in the ~/file
MAF[0]<0.05    .. select rare variants at 5% cutoff
POS>=100   .. restrict your range query, e.g. 20:100-200 to strictly sites with POS in that range.
```

Shell expansion:

Note that expressions must often be quoted because some characters have special meaning in the shell. An example of expression enclosed in single quotes which cause that the whole expression is passed to the program as intended:

```
bcftools view -i '%ID!="." & MAF[0]<0.01'
```

Please refer to the documentation of your shell for details.

SCRIPTS AND OPTIONS

plot-vcfstats [*OPTIONS*] *file.vchk* […]

Script for processing output of bcftools stats. It can merge results from multiple outputs (useful when running the stats for each chromosome separately), plots graphs and creates a PDF presentation.

-m, --merge
Merge vcfstats files to STDOUT, skip plotting.

-p, --prefix *DIR*
The output directory. This directory will be created if it does not exist.

-P, --no-PDF
Skip the PDF creation step.

-r, --rasterize
Rasterize PDF images for faster rendering.

-s, --sample-names
Use sample names for xticks rather than numeric IDs.

-t, --title *STRING*

Identify files by these titles in plots. The option can be given multiple times, for each ID in the **bcftools stats** output. If not present, the script will use abbreviated source file names for the titles.

-T, --main-title *STRING*
Main title for the PDF.

PERFORMANCE

HTSlib was designed with BCF format in mind. When parsing VCF files, all records are internally converted into BCF representation. Simple operations, like removing a single column from a VCF file, can be therefore done much faster with standard UNIX commands, such as **awk** or **cut**. Therefore it is recommended to use BCF as input/output format whenever possible to avoid large overhead of the VCF → BCF → VCF conversion.

BUGS

Please report any bugs you encounter on the github website: http://github.com/samtools/bcftools

AUTHORS

Heng Li from the Sanger Institute wrote the original C version of htslib, samtools and bcftools. Bob Handsaker from the Broad Institute implemented the BGZF library. Petr Danecek, Shane McCarthy and John Marshall are maintaining and further developing bcftools. Many other people contributed to the program and to the file format specifications, both directly and indirectly by providing patches, testing and reporting bugs. We thank them all.

RESOURCES

BCFtools GitHub website: http://github.com/samtools/bcftools

Samtools GitHub website: http://github.com/samtools/samtools

HTSlib GitHub website: http://github.com/samtools/htslib

File format specifications: http://samtools.github.io/hts-specs

BCFtools documentation: http://samtools.github.io/bcftools

BCFtools wiki page: https://github.com/samtools/bcftools/wiki

COPYING