DRAFT SPEC SUBJECT TO CHANGE
The Variant Call Format Specification

VCFv4.5 and BCFv2.2

20 Apr 2024

The master version of this document can be found at `https://github.com/samtools/hts-specs`.
This printing is version e3630c0 from that repository, last modified on the date shown above.

- A: The field has one value per alternate allele. The values must be in the same order as listed in the ALT column (described in section 1.6).
- R: The field has one value for each possible allele, including the reference. The order of the values must be the reference allele first, then the alternate alleles as listed in the ALT column.
- G: The field has one value for each possible genotype. The values must be in the same order as prescribed in section 1.6.2 (see GENOTYPE ORDERING).
- . (dot): The number of possible values varies, is unknown or unbounded.

The 'Flag' type indicates that the INFO field does not contain a Value entry, and hence the Number must be 0 in this case. The Description value must be surrounded by double-quotes. Double-quote character must be escaped with backslash \ and backslash as \\. Source and Version values likewise must be surrounded by double-quotes and specify the annotation source (case-insensitive, e.g. `"dbsnp"`) and exact version (e.g. `"138"`), respectively for computational use.

1.4.3 Filter field format

FILTER meta-information lines are structured lines with required fields ID and Description that define the possible content of the FILTER column in the VCF records:

```
##FILTER=<ID=ID,Description="description">
```

1.4.4 Individual format field format

FORMAT meta-information lines are structured lines with required fields ID, Number, Type, and Description that define the possible content of the per-sample/genotype columns in the VCF records:

```
##FORMAT=<ID=ID,Number=number,Type=type,Description="description">
```

Possible Types for FORMAT fields are: Integer, Float, Character, and String (this field is otherwise defined precisely as the INFO field). The Number field is defined as per the INFO Number field with the following additional possibilities:

- LA: Identical to A except the only alternate alleles defined in the *LAA* field are considered present.
- LR: Identical to R except the only alternate alleles defined in the *LAA* field are considered present.
- LG: Identical to G except the only alternate alleles defined in the *LAA* field are considered present.
- P: The field has one value for each allele value defined in *GT*/*LGT*.

1.4.5 Alternative allele field format

ALT meta-information lines are structured lines with require fields of ID and Description that describe the possible symbolic alternate alleles in the ALT column of the VCF records:

```
##ALT=<ID=type,Description="description">
```

Structural Variants
In symbolic alternate alleles for structural variants, the ID field indicates the type of structural variant, and can be a colon-separated list of types and subtypes. ID values are case sensitive strings and must not contain whitespace, commas or angle brackets (See 1.6.1.5) The first level type must be one of the following:

- DEL Region of lowered copy number relative to the reference, or a deletion breakpoint
- INS Insertion of novel sequence relative to the reference
- DUP Region of elevated copy number relative to the reference, or a tandem duplication breakpoint
- INV Inversion of reference sequence

... Continued from previous page

Key	Number	Type	Description
DB	0	Flag	dbSNP membership
DP	1	Integer	Combined depth across samples
END	1	Integer	~~End position on CHROM (used with symbolic alleles; see below)~~ Deprecated. Present for backwards compatibility with earlier versions of VCF.
H2	0	Flag	HapMap2 membership
H3	0	Flag	HapMap3 membership
MQ	1	Float	RMS mapping quality
MQ0	1	Integer	Number of MAPQ == 0 reads
NS	1	Integer	Number of samples with data
SB	4	Integer	Strand bias
SOMATIC	0	Flag	Somatic mutation (for cancer genomics)
VALIDATED	0	Flag	Validated by follow-up experiment
1000G	0	Flag	1000 Genomes membership

Table 1: Reserved INFO keys

- END: ~~End~~ Deprecated. Retained for backwards compatibility with earlier versions of VCF and older VCF indexing software which rely on this field being present.

  This is a computed field that, when present, must be set to the maximum end reference position (1-based) ~~, indicating the variant spans positions POS–END on reference/contig CHROM. Normally this is the~~ of: the position of the ~~last base in~~ final base of the REF allele, ~~so it can be derived from POS and the length of REF, and no END INFO field is needed. However when symbolic alleles are used, e.g. in gVCF or structural variants, an explicit END INFO field provides variant span information that is otherwise unknown. If a record containing a symbolic structural variant alleledoes not have an END field, it must be computed from~~ the ~~SVLEN field as per Section 3~~end position corresponding to the SVLEN of a symbolic SV allele, and the end positions calculated from FORMAT LEN for the <*> symbolic allele.

  ~~This~~ The computed value of this field is used to compute BCF's `rlen` field (see 6.3.1) and is important when indexing VCF/BCF files to enable random access and querying by position.

1.6.2 Genotype fields

If genotype information is present, then the same types of data must be present for all samples. First a FORMAT field is given specifying the data types and order (colon-separated FORMAT keys matching the regular expression `^[A-Za-z_][0-9A-Za-z_.]*$`, duplicate keys are not allowed). This is followed by one data block per sample, with the colon-separated data corresponding to the types specified in the format. The first key must always be the genotype (GT) if it is present. If LGT key is present, it must precede all fields other than GT. If any local-allele field is present, LAA must also be present and precede all fields other than GT and LGT. There are no required keys. Additional Genotype keys can be defined in the meta-information, however, software support for them is not guaranteed.

If any of the fields is missing, it is replaced with the MISSING value. For example if the FORMAT is GT:GQ:DP:HQ then $0 \mid 0 : . : 23 : 23,34$ indicates that GQ is missing. If a field contains a list of missing values, it can be represented either as a single MISSING value ('.') or as a list of missing values (e.g. '.,.,.' if the field was Number=3). Trailing fields can be dropped, with the exception of the GT field, which should always be present if specified in the FORMAT field. If a field and it's local-allele equivalent (including GT/LGT) are both defined they must encode identical information or one must ignored by containing the MISSING value or omitted.

As with the INFO field, there are several common, reserved keywords that are standards across the community. See their detailed definitions below, as well as Table 2 for their reference Number, Type and Description. See also Section 4 for a list of genotype keys reserved for structural variants.

Field	Number	Type	Description
AD	R	Integer	Read depth for each allele
ADF	R	Integer	Read depth for each allele on the forward strand
ADR	R	Integer	Read depth for each allele on the reverse strand
DP	1	Integer	Read depth
EC	A	Integer	Expected alternate allele counts
LEN	1	Integer	Length of <*> reference block
FT	1	String	Filter indicating if this genotype was "called"
GL	G	Float	Genotype likelihoods
GP	G	Float	Genotype posterior probabilities
GQ	1	Integer	Conditional genotype quality
GT	1	String	Genotype
HQ	2	Integer	Haplotype quality
LA	.	Integer	Reserved
LAA	.	Integer	1-based indices into ALT, indicating which alleles are relevant (local) for the current sample
LAD	LR	Integer	Local-allele representation of AD
LADF	LR	Integer	Local-allele representation of ADF
LADR	LR	Integer	Local-allele representation of ADR
LEC	LA	Integer	Local-allele representation of EC
LGL	LG	Integer	Local-allele representation of GL
LGP	LG	Integer	Local-allele representation of GP
LGT	1	String	Local-allele representation of GT
LPL	LG	Integer	Local-allele representation of PL
LPP	LG	Integer	Local-allele representation of PP
MQ	1	Integer	RMS mapping quality
PL	G	Integer	Phred-scaled genotype likelihoods rounded to the closest integer
PP	G	Integer	Phred-scaled genotype posterior probabilities rounded to the closest integer
PQ	1	Integer	Phasing quality
PS	1	Integer	Phase set
PSL	P	String	Phase set list
PSO	P	Integer	Phase set list ordinal
PSQ	P	Integer	Phase set list quality

Table 2: Reserved genotype keys

- AD, ADF, ADR (Integer): Per-sample read depths for each allele; total (AD), on the forward (ADF) and the reverse (ADR) strand.
- DP (Integer): Read depth at this position for this sample.
- EC (Integer): Comma separated list of expected alternate allele counts for each alternate allele in the same order as listed in the ALT field. Typically used in association analyses.
- LEN (Integer): length of the <*> reference block for this sample.

- FT (String): Sample genotype filter indicating if this genotype was "called" (similar in concept to the FILTER field). Again, use PASS to indicate that all filters have been passed, a semicolon-separated list of codes for filters that fail, or '.' to indicate that filters have not been applied. These values should be described in the meta-information in the same way as FILTERs. No whitespace or semicolons permitted.

- GQ (Integer): Conditional genotype quality, encoded as a phred quality $-10log_{10}$ p(genotype call is wrong, conditioned on the site's being variant).

- GP (Float): Genotype posterior probabilities in the range 0 to 1 using the same ordering as the GL field; one use can be to store imputed genotype probabilities.

- GT (String): Genotype, encoded as allele value preceded by either of / or | depending on whether that allele is considered phased. The first phasing indicator may be omitted and is implicitly defined as / if any phasing indicators are / and | otherwise. The allele values are 0 for the reference allele (what is in the REF field), 1 for the first allele listed in ALT, 2 for the second allele list in ALT and so on. For diploid calls examples could be 0/1, 1 | 0, /0/1, or 1/2, etc. Haploid calls, e.g. on Y, male non-pseudoautosomal X, or mitochondria, should be indicated by having only one allele value. A triploid call might look like 0/0/1, and a partially phased triploid call could be |0/1/2 to indicate that the first allele is phased with another variant in the VCF. If a call cannot be made for a sample at a given locus, '.' must be specified for each missing allele in the `GT` field (for example './.' for a diploid genotype and '.' for haploid genotype). The meanings of the phasing indicators are as follows (see the `PS` and `PSL` fields below for more details on incorporating phasing information into the genotypes):

  - / : allele is unphased
  - | : allele is phased (according to the phase-set indicated in `PS` or `PSL`)

  For symbolic structural variant alleles, GT=0 indicates the absence of any of the ALT symbolic structural variants defined in the record. Implementer should note that merging a VCF record containing only symbolic structural variant ALT alleles with a record containing other alleles will result a change of the meaning of the GT=0 haplotypes from the record containing only symbolic SVs.

- GL (Float): Genotype likelihoods comprised of comma separated floating point log_{10}-scaled likelihoods for all possible genotypes given the set of alleles defined in the REF and ALT fields. In presence of the GT field the same ploidy is expected; without GT field, diploidy is assumed.

  Genotype Ordering. In general case of ploidy P and N alternate alleles (0 is the REF and $1 \dots N$ the alternate alleles), the ordering of genotypes for the likelihoods can be expressed by the following pseudocode with as many nested loops as ploidy: [‡]

```
for a_P = 0...N
  for a_{P-1} = 0...a_P
      ...
      for a_1 = 0...a_2
          println a_1 a_2 ... a_P
```

  Alternatively, the same can be achieved recursively with the following pseudocode:

```
Ordering(P, N, suffix=""):
    for a in 0...N
        if (P == 1) println str(a) + suffix
        if (P > 1) Ordering(P-1, a, str(a) + suffix)
```

  Conversely, the index of the value corresponding to the genotype $k_1 \le k_2 \le \ldots \le k_P$ is

  $\texttt{Index}(k_1/k_2/\ldots/k_P) = \sum_{m=1}^{P} \binom{k_m+m-1}{m}$

  Examples:

  - for P=2 and N=1, the ordering is 00,01,11
  - for P=2 and N=2, the ordering is 00,01,11,02,12,22
  - for P=3 and N=2, the ordering is 000, 001, 011, 111, 002, 012, 112, 022, 122, 222
  - for P=1, the index of the genotype a is a

[‡]Note that we use inclusive `for` loop boundaries.

- for P=2, the index of the genotype "a/b", where $a \leq b$, is $b(b+1)/2+a$
- for P=2 and arbitrary N, the ordering can be easily derived from a triangular matrix

$b \setminus a$	0	1	2	3
0	0			
1	1	2		
2	3	4	5	
3	6	7	8	9

- HQ (Integer): Haplotype qualities, two comma separated phred qualities.
- LAA is a list of n distinct integers, giving the 1-based indices of the ALT alleles that are observed in the sample. In callsets with many samples, sites may grow to include numerous alternate alleles at the same POS. Usually, few of these alleles are actually observed in any one sample, but each genotype must supply fields like PL and AD for all of the alleles—a very inefficient representation as PL's size is quadratic in the allele count. Similarly, in rare sites, which can be the bulk of the sites, the vast majority of the samples are reference. To prevent this growth in VCF size, one can choose to specify the genotype, allele depth and the genotype likelihood against a subset of "Local Alleles". LAA is the 1-based index into ALT, defining the alleles that are actually in-play for that sample and the order in which they are interpreted. LAA is required when interpreting local-allele fields and must be present if any local-allele fields are neither omitted nor MISSING. Since BCF encodes zero length vectors as MISSING, a LAA containing the MISSING value should be treated as the empty vector (i.e. a REF-only site) if any local-allele fields are neither omitted nor MISSING. All specifications-defined A, R and G FORMAT fields have a local-allele equivalent that should be interpreted in the same manner as it's matching field except for the ALT alleles considered present and the order in which they are interpreted. For example, if REF is G, ALT is A,C,T,<*> and a genotype only has information about G, C, and <*>, one can have LAA=[2,4] and thus LPL will be interpreted as pertaining to the alleles [G, C, <*>] and not contain likelihood values for genotypes that involve A or T. In this case LGT=0/1 means that the sample is G/C. GQ is still the genotype quality, even when the genotype is given against the local alleles. In the following example, the records with the same POS encode the same information (some columns removed for clarity):

POS	REF	ALT	FORMAT	sample	
1	G	A,C,T,<*>	LGT:LAA:LAD:LPL	1/1:2,4:20,30,10:90,80,0,100,110,120	
1	G	A,C,T,<*>	GT:AD:PL	2/2:20,.,30,.,10:90,...,80,.,0,.....,100,.,110,.,120	
2	A	C,G,T,<*>	GT:LAA:LAD:LPL	0/3:3:15,25:40,0,80	
2	A	C,G,T,<*>	GT:AD:PL	0/3:15,...,25,.:40,.....,0,...,80,.....	Due to
3	C	G,T,<*>	LGT:LAA:LAD:LPL	0/0:3:30,1:0,30,80	
3	C	G,T,<*>	GT:AD:PL	0/0:30,...,1:0,.....,30,...,80	
4	G	A,T,<*>	LGT:LAA:LAD:LPL	0/0::30:0	
4	G	A,T,<*>	GT:AD:PL	0/0:30,....:0,.....	

BCF encoding empty vectors as missing, implementation-defined Number=LA local-allele fields should not be used if distinguishing between zero-length data and missing data is required at REF-only sites.

- LGT: is the genotype, encoded as allele indexes separated by either of / or |, as with GT, however, the indexes are into the alleles referenced by LAA. So that in the case that LAA is 2,3, LGT=0/2 is equivalent to GT=0/3 and LGT=1/2 is equivalent to GT=2/3 (see example above).
- LPL: is a list of $\binom{n}{\text{Ploidy}}$ integers giving phred-scaled genotype likelihoods (rounded to the closest integer; as per PL) for all possible genotypes given the set of alleles defined in the LAA local alleles. The precise ordering is defined in the GL paragraph.
- MQ (Integer): RMS mapping quality, similar to the version in the INFO field.
- PL (Integer): The phred-scaled genotype likelihoods rounded to the closest integer, and otherwise defined in the same way as the GL field.
- PP (Integer): The phred-scaled genotype posterior probabilities rounded to the closest integer, and otherwise defined in the same way as the GP field.
- PQ (Integer): Phasing quality, the phred-scaled probability that alleles are ordered incorrectly in a heterozygote (against all other members in the phase set). We note that we have not yet included the specific measure for precisely defining "phasing quality"; our intention for now is simply to reserve the PQ tag for future use as a measure of phasing quality.

- PS (non-negative 32-bit Integer): Phase set, defined as a set of phased genotypes to which this genotype belongs. Phased genotypes for an individual that are on the same chromosome and have the same PS value are in the same phased set. A phase set specifies multi-marker haplotypes for the phased genotypes in the set. All phased genotypes that do not contain a PS subfield are assumed to belong to the same phased set. If the genotype in the GT field is unphased, the corresponding PS field is ignored. The recommended convention is to use the position of the first variant in the set as the PS identifier (although this is not required).

- PSL (List of Strings): The list of phase sets, one for each allele value specified in the `GT` or `LGT`. Unphased alleles (without a | separator before them) must have the value '.' in their corresponding position in the list. Unlike `PS` (which is defined per CHROM), records with different CHROM but the same phase-set name are considered part of the same phase set. If an implementation cannot guarantee uniqueness of phase-set names across the VCF (for example, phasing a streaming VCF or each CHROM is processed independently in parallel), new phase-set names should be of the format CHROM*POS*ALLELE-NUMBER of the "first" allele which is included in this set, with ALLELE-NUMBER being the index of the allele in the `GT` field, since multiple distinct phase-sets could start at the same position. [§] A given sample-genotype must not have values for both PS and PSL. In addition, PS and PSL are not interoperable, in that a PS mentioned in one variant cannot be referenced in a PSL in another, since when used in PS it isn't connected to any specific haplotype (i.e. first or second), but PSL is.

  Example:

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	SAMPLE1
chr19	5	.	T	G	.	PASS	DP=100	GT:PSL	`|0/1:chr9*5*1,.`
chr20	10	.	A	T,G	.	PASS	DP=100	GT:PSL	`|1/2|3:chr20*10*1,.,chr9*5*1`
chr20	15	.	G	C	.	PASS	DP=100	GT:PSL	`1|2:.,chr20*10*1`

- PSO (List of integers): List of phase set ordinals. For each phase-set name, defines the order in which variants are encountered when traversing a derivative chromosome. The missing value '.' should be used when the corresponding PSO value is missing. For each phase-set name, PSO should be defined if any allele with that phase-set name on any record is symbolic structural variant or in breakpoint notation. Variants in breakpoint notation must have the same PSL and PSO on both records.

  Without explicitly specifying the derivative chromosome traversal order, multiple derivative chromosome reconstructions are possible. Take for example this tandem duplication in a triploid organism with SNVs (ID/QUAL/FILTER columns removed for clarity):

#CHROM	POS	REF	ALT	INFO	FORMAT	SAMPLE1
chr1	10	T	<DUP>	SVCLAIM=DJ	GT:PSL:PSO	`/0/0|1:.,.,chr1*10*1:.,.,3`
chr1	20	A	G	.	GT:PSL:PSO	`/0/0|0|1:.,.,chr1*10*1,chr1*10*1:.,.,4,`
chr1	30	G	T	.	GT:PSL:PSO	`/0/0|0|1:.,.,chr1*10*1,chr1*10*1:.,.,2,`

  Without defining PSO, it would be ambiguous as to which copy of the duplicated region the SNVs occur on. In this example, the presence of the PSO field clarifies that the SNVs are cis phased with the duplication, the first SNV occurs on the first copy of the duplicated region, and second SNV on the second copy.

- PSQ (List of integers): The list of PQs, one for each phase set in PSL (encoded like PQ). The missing value '.' should be used when the corresponding PSL value is missing, or when the phasing is of unknown quality.

2 Understanding the VCF format and the haplotype representation

VCF records use a single general system for representing genetic variation data composed of:

- Allele: representing single genetic haplotypes (A, T, ATC).
- Genotype: an assignment of alleles for each chromosome of a single named sample at a particular locus.
- VCF record: a record holding all segregating alleles at a locus (as well as genotypes, if appropriate, for multiple individuals containing alleles at that locus).

[§]The '*' character is used as a separator since ':' is not reserved in the CHROM column.

VCF records use a simple haplotype representation for REF and ALT alleles to describe variant haplotypes at a locus. ALT haplotypes are constructed from the REF haplotype by taking the REF allele bases at the POS in the reference genotype and replacing them with the ALT bases. In essence, the VCF record specifies a-REF-t and the alternative haplotypes are a-ALT-t for each alternative allele.

2.1 VCF tag naming conventions

Several tag names follow conventions ~~indicating how their values are represented numerically~~which should be used for implementation-defined tag as well:

- The 'L' suffix means *likelihood* as log-likelihood in the sampling distribution, $\log_{10}\Pr(\text{Data}|\text{Model})$. Likelihoods are represented as $\log_{10}$ scale, thus they are negative numbers (e.g. GL, CNL). The likelihood can be also represented in some cases as phred-scale in a separate tag (e.g. PL).
- The 'P' suffix means *probability* as linear-scale probability in the posterior distribution, which is $\Pr(\text{Model}|\text{Data})$. Examples are GP, CNP.
- The 'Q' suffix means *quality* as log-complementary-phred-scale posterior probability, $-10\log_{10}\Pr(\text{Data}|\text{Model})$, where the model is the most likely genotype that appears in the GT field. Examples are GQ, CNQ. The fixed site-level QUAL field follows the same convention (represented as a phred-scaled number).
- The 'L' prefix indicates the local-allele equivalent of a Number=A, R or G field.

3 INFO keys used for structural variants

The following INFO keys are reserved for encoding structural variants. In general, when these keys are used by imprecise variants, the values should be best estimates. When present, per allele values must be specified for all ALT alleles (including non-structural alleles). Except in lists of strings, the missing value should be used as a placeholder for the ALT alleles for which the key does not have a meaningful value. The empty string should be used to encode missing values in lists of strings.

```
##INFO=<ID=IMPRECISE,Number=0,Type=Flag,Description="Imprecise structural variation">
```

Indicates that this record contains an imprecise structural variant *ALT* allele. ALT alleles missing *CIPOS* are to be interpreted as imprecise variants with an unspecified confidence interval.

If a precise ALT allele is present in a record with the *IMPRECISE* flag, *CIPOS* must be explicitly set for that allele, even if it is '0,0'.

```
##INFO=<ID=NOVEL,Number=0,Type=Flag,Description="Indicates a novel structural variation">
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the longest variant described in this record">
##INFO=<ID=END,Number=1,Type=Integer,Description="Deprecated. Present for backwards compatibility with earlier versions of VCF.">
```

END ~~position of the longest variant described in this record. The END of each allele is defined as:~~

~~Non-symbolic alleles: POS + length of REF allele − 1~~has been deprecated in favour of INFO SVLEN and FORMAT LEN.

~~<INS> symbolic structural variant alleles: POS + length of REF allele − 1.~~

~~<DEL>, <DUP>, <INV>, and <CNV> symbolic structural variant alleles:, POS + SVLEN.~~

~~<*> symbolic allele: the last reference call position.~~

~~END must be present for all records containing the <*> symbolic allele and, for backwards compatibility, should be present for records containing any symbolic structural variant alleles.~~

~~To prevent loss of information, any VCF record containing the <*> symbolic allele must have END set to the last reference call position of the <*> symbolic allele. When a record contains both the <*> symbolic allele, the END position of the longest allele should be used as the record end position for indexing purposes.~~

```
##INFO=<ID=SVTYPE,Number=1,Type=String,Description="Type of structural variant">
```

This field has been deprecated due to redundancy with ALT. Refer to section 1.4.5 for the set of valid ALT field symbolic structural variant alleles.

```
##INFO=<ID=SVLEN,Number=A,Type=Integer,Description="Length of structural variant">
```

One value for each ALT allele.

SVLEN must be specified for symbolic structural variant alleles. SVLEN is defined for INS, DUP, INV, and DEL symbolic alleles as the number of the inserted, duplicated, inverted, and deleted bases respectively. SVLEN is defined for CNV symbolic alleles as the length of the segment over which the copy number variant is defined. The missing value . should be used for all other ALT alleles, including ALT alleles using breakend notation.

For backwards compatibility, a missing SVLEN should be inferred from the END field~~of VCF records whose *ALT* field contains a single symbolic allele~~.

For backwards compatibility, the absolute value of SVLEN should be taken and a negative SVLEN should be treated as positive values.

Note that for structural variant symbolic alleles, POS corresponds to the base immediately preceding the variant.

```
##INFO=<ID=CIPOS,Number=.,Type=Integer,Description="Confidence interval around POS for symbolic structural variants">
```

If present, the number of entries must be twice the number of ALT alleles. $CIPOS$ consists of successive pairs of records indicating the start and end offsets relative to POS of the confidence interval for each ALT allele. For example, $CIPOS = -5, 5, 0, 0$ indicates a 5bp confidence interval in each direction for the first ALT allele, and an exact position for the second alt allele.

When breakpoint sequence homology exists, $CIPOS$ should be used in conjunction with $HOMSEQ$ to specify the interval of homology.

If both $IMPRECISE$ and $CIPOS$ are omitted, $CIPOS$ is implicitly defined as 0,0 for all alleles.

Each $CIPOS$ interval must span 0. That is, the lower bound cannot be greater than 0, and the upper bound cannot be less than 0.

```
##INFO=<ID=CIEND,Number=.,Type=Integer,Description="Confidence interval around END for symbolic structural variants">
##INFO=<ID=CIEND,Number=.,Type=Integer,Description="Confidence interval around the inferred END for symbolic structural variants">
```

If present, the number of entries must be twice the number of ALT alleles. $CIEND$ consists of successive pairs of records encoding the confidence interval start and end offsets relative to the END position inferred by $SVLEN$ for each ALT allele. For symbolic structural variants, the first in the pair must not be greater than 0, and the second must not be less than 0. For all other alleles, both should be the missing value .. For example, $CIEND = -5, 5, ., .$ indicates a 5bp confidence interval in each direction around the end position for the first ALT allele, and no $CIEND$ is defined for the second alt allele.

If $CIEND$ is missing, it is assumed to match $CIPOS$.

```
##INFO=<ID=HOMLEN,Number=A,Type=Integer,Description="Length of base pair identical micro-homology at breakpoints">

##INFO=<ID=HOMSEQ,Number=A,Type=String,Description="Sequence of base pair identical micro-homology at breakpoints">

##INFO=<ID=BKPTID,Number=A,Type=String,Description="ID of the assembled alternate allele in the assembly file">
```

For precise variants, the consensus sequence the alternate allele assembly is derivable from the REF and ALT fields. However, the alternate allele assembly file may contain additional information about the characteristics of the alt allele contigs.

```
##INFO=<ID=MEINFO,Number=.,Type=String,Description="Mobile element info of the form NAME,START,END,POLARITY">
```

If present, the number of entries must be four (4) times the number of ALT alleles. $MEINFO$ consists of successive quadruplets of records for each ALT allele.

```
##INFO=<ID=METRANS,Number=.,Type=String,Description="Mobile element transduction info of the form CHR,START,END,POLARITY">
```

If present, the number of entries must be four (4) times the number of ALT alleles. $METRANS$ consists of successive quadruplets of records for each ALT allele.

```
##INFO=<ID=DGVID,Number=A,Type=String,Description="ID of this element in Database of Genomic Variation">
##INFO=<ID=DBVARID,Number=A,Type=String,Description="ID of this element in DBVAR">
##INFO=<ID=DBRIPID,Number=A,Type=String,Description="ID of this element in DBRIP">
##INFO=<ID=MATEID,Number=A,Type=String,Description="ID of mate breakend">
##INFO=<ID=PARID,Number=A,Type=String,Description="ID of partner breakend">
##INFO=<ID=EVENT,Number=A,Type=String,Description="ID of associated event">
##INFO=<ID=EVENTTYPE,Number=A,Type=String,Description="Type of associated event">
```

VCF STRUCTURAL VARIANT EXAMPLE

```
##fileformat=VCFv4.5
##ALT=<ID=INV,Description="Inversion">
##ALT=<ID=INS,Description="Insertion">
##ALT=<ID=DUP,Description="Duplication">
##ALT=<ID=DUP:TANDEM,Description="Tandem Duplication">
##ALT=<ID=DEL,Description="Deletion">
##ALT=<ID=CNV,Description="Copy number variable region">
##INFO=<ID=MATEID,Number=A,Type=String,Description="ID of mate breakend">
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the longest variant described in this record">
##INFO=<ID=CIPOS,Number=.,Type=Integer,Description="Confidence interval around POS for symbolic structural variants">
##INFO=<ID=SVLEN,Number=A,Type=Integer,Description="Length of structural variant">
##INFO=<ID=CILEN,Number=.,Type=Integer,Description="Confidence interval for the SVLEN field">
##INFO=<ID=EVENT,Number=A,Type=String,Description="ID of associated event">
##INFO=<ID=EVENTTYPE,Number=A,Type=String,Description="Type of associated event">
##INFO=<ID=CN,Number=A,Type=Float,Description="Copy number of CNV/breakpoint">
##INFO=<ID=SVCLAIM,Number=A,Type=String,Description="Claim made by the structural variant call. Valid values are D, J, DJ for abundance, adjacency and both respectively">
##INFO=<ID=IMPRECISE,Number=0,Type=Flag,Description="Imprecise structural variation">
##contig=<ID=chrA,length=1000000>
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
#CHROM POS ID   REF ALT       QUAL FILTER INFO                                                  FORMAT sample
chrA   2   .    TGC T         .    .      EVENT=DEL_seq                                           GT    0/1
chrA   2   .    T   <DEL>     .    .      SVLEN=2;SVCLAIM=DJ;EVENT=DEL_symbolic;END=4             GT    0/1
chrA   2   .    T   <DEL>     .    .      SVLEN=2;SVCLAIM=DJ;EVENT=DEL_symbolic                   GT    0/1
chrA   2 delbp1 T   T[chrA:5[ .    .      MATEID=delbp2;EVENT=DEL_split_bp_cn                     GT    0/1
chrA   2 delbp2 A   ]chrA:2]A .    .      MATEID=delbp1;EVENT=DEL_split_bp_cn                     GT    0/1
chrA   2   .    T   <DEL>     .    .      SVLEN=2;SVCLAIM=D;EVENT=DEL_split_bp_cn;END=4           GT    0/1
chrA   2   .    T   <DEL>     .    .      SVLEN=2;SVCLAIM=D;EVENT=DEL_split_bp_cn                 GT    0/1
chrA   5   .    G   GAAA      .    .      EVENT=homology_seq                                      GT    1/1
chrA   5   .    G   <DUP>     .    .      SVLEN=3;CIPOS=0,5;EVENT=homology_dup                    GT    0/1
chrA   14  .    T   <INS>     .    .      IMPRECISE;SVLEN=100;CILEN=-50,50;CIPOS=-10,10;END=14    GT    0/1
chrA   14  .    T   <INS>     .    .      IMPRECISE;SVLEN=100;CILEN=-50,50;CIPOS=-10,10           GT    0/1
chrA   14  .    G   .CCCCCCG  .    .      EVENT=single_breakend                                   GT    0/1
```

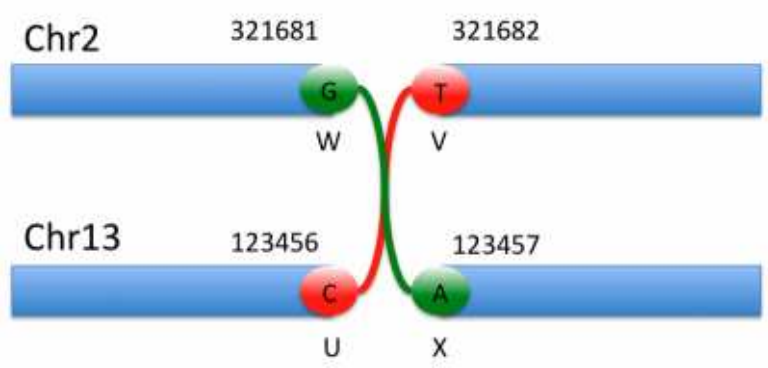

Figure 7: Rearrangements

Figure 8: Inversion

5.4.7 Inversions

Similarly an inversion such as in Figure 8:
can be described equivalently in two ways. Either one uses the short hand notation described previously (recommended for simple cases):

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
2	321681	INV0	T	<INV>	6	PASS	~~END~~SVLEN=~~421681~~100000

or one describes the breakends:

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
2	321681	bnd_W	G	G]2 : 421681]	6	PASS	MATEID=bnd_U;EVENT=INV0
2	321682	bnd_V	T	[2 : 421682[T	6	PASS	MATEID=bnd_X;EVENT=INV0
2	421681	bnd_U	A	A]2 : 321681]	6	PASS	MATEID=bnd_W;EVENT=INV0
2	421682	bnd_X	C	[2 : 321682[C	6	PASS	MATEID=bnd_V;EVENT=INV0

5.4.8 Uncertainty around breakend location

It sometimes is difficult to determine the exact position of a break, generally because of homologies between the sequences being modified, such as in Figure 9. The breakend is then placed arbitrarily at the left most position, and the uncertainty is represented with the CIPOS tag. The ALT string is then constructed assuming this arbitrary breakend choice.

The figure above represents a nonreciprocal translocation with microhomology. Even if we know that breakend U is rearranged with breakend V, actually placing these breaks can be extremely difficult. The red and green dashed lines represent the most extreme possible recombination events which are allowed by the sequence evidence available. We therefore place both U and V arbitrarily within the interval of possibility:

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
2	321681	bnd_V	T	T]13 : 123462]	6	PASS	MATEID=bnd_U;CIPOS=0,6
13	123456	bnd_U	A	A]2 : 321687]	6	PASS	MATEID=bnd_V;CIPOS=0,6

Note that the coordinate in breakend U's ALT string does not correspond to the designated position of breakend V, but to the position that V would take if U's position were fixed (and vice-versa). The CIPOS tags describe the uncertainty around the positions of U and V.

The fact that breakends U and V are mates is preserved thanks to the MATEID tags. If this were a reciprocal translocation, then there would be additional breakends X and Y, say with X the partner of V on Chr 2 and Y

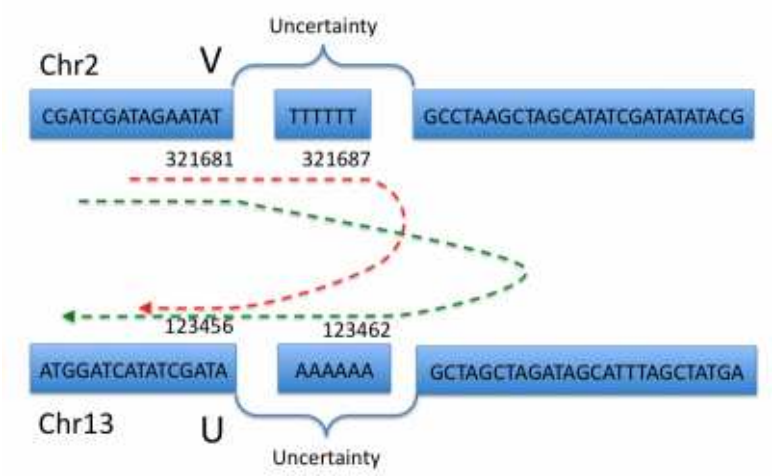

Figure 9: Homology

the partner of U on Chr 13, and there would be two more lines of VCF for the XY novel adjacency. Depending on which positions are chosen for the breakends X and Y, it might not be obvious that X is the partner of V and Y is the partner of U from their locations alone. This partner relationship can be specified explicitly with the tag PARID=bnd_X in the VCF line for breakend V and PARID=bnd_Y in the VCF line for breakend U, and vice versa.

5.4.9 Single breakends

We allow for the definition of a breakend that is not part of a novel adjacency. We call these single breakends, because they lack a mate. Breakends that are unobserved partners of breakends in observed novel adjacencies are one kind of single breakend. For example, if the true situation is known to be either as depicted back in Figure 1, and we only observe the adjacency (U,V), and no adjacencies for W, X, Y, or Z, then we cannot be sure whether we have a simple reciprocal translocation or a more complex 3-break operation. Yet we know the partner X of U and the partner W of V exist and are breakends. In this case we can specify these as single breakends, with unknown mates. The 4 lines of VCF representing this situation would be:

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
2	321681	bnd_W	G	G.	6	PASS	.
2	321682	bnd_V	T	]13 : 123456]T	6	PASS	MATEID=bnd_U
13	123456	bnd_U	C	C[2 : 321682[	6	PASS	MATEID=bnd_V
13	123457	bnd_X	A	.A	6	PASS	.

On the other hand, if we know a simple reciprocal translocation has occurred as in Figure 7, then even if we have no evidence for the (W,X) adjacency, for accounting purposes an adjacency between W and X may also be recorded in the VCF file. These two breakends W and X can still be crossed-referenced as mates. The 4 VCF records describing this situation would look exactly as in section 5.4.4, but perhaps with a special quality or filter value for the breakends W and X.

Another possible reason for calling single breakends is an observed but unexplained change in copy number along a chromosome.

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
3	12665	bnd_X	A	.A	6	PASS	CIPOS=-50,50
3	12665	.	A	<DUP>	14	PASS	~~END~~SVCLAIM=~~13686;~~D;SVLEN=1021;CIPOS=-50,50;CIEND=-50,50
3	13686	bnd_Y	T	T.	6	PASS	CIPOS=-50,50

Finally, if an insertion is detected but only the first few base-pairs provided by overhanging reads could be assembled, then this inserted sequence can be provided on that line, in analogy to paired breakends:

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO
3	12665	bnd_X	A	.TGCA	6	PASS	CIPOS=-50,50
3	12665	.	A	<DUP>	14	PASS	~~END~~SVCLAIM=~~13686;~~D;SVLEN=1021;CIPOS=-50,50;CIEND=-50,50
3	13686	bnd_Y	T	TCC.	6	PASS	CIPOS=-50,50

5.5 Representing unspecified alleles and REF-only blocks (gVCF)

In order to report sequencing data evidence for both variant and non-variant positions in the genome, the VCF specification allows to represent blocks of reference-only calls in a single record using the ~~END INFO tag, an idea originally introduced by the gVCF file format~~[¶]~~.~~

<*> allele and the FORMAT LEN field. The convention adopted here is to represent reference evidence as likelihoods against an unknown alternate allele represented as <*>. Think of this as the likelihood for reference as compared to any other possible alternate allele (both SNP, indel, or otherwise). ~~The~~

Positions implicitly called by a preceding <*> ~~representation is preferred over the~~ for a sample must have *GT*/*LGT* set to the missing value ('.') and have no FORMAT fields other than *LAA* present. If *LAA* is present and a reference block start is being defined for a given sample, the <*> allele must be included as an *LAA* allele for that sample even though the *GT*/*LGT* is 0/0.

Reference blocks were originally introduced by the gVCF file format[¶]. Unfortunately, gVCF has issues scaling to many samples as the use of INFO END to encode the reference block length requires the reference block length to be the same for all samples.

To retain backwards compatibility with with gVCF, the symbolic allele <NON_REF> should be treated as an alias of <*> and a missing FORMAT LEN field should be inferred from the INFO END tag if present.

~~Example records are~~ An example with both FORMAT LEN and a redundant INFO END is given below:

#CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	Sample
1	4370	.	G	<*>	.	.	END=4383	GT:DP:GQ:MIN_DP:PL:LEN	0/0:25:60:23:0,60,900:14
1	4384	.	C	<*>	.	.	END=4388	GT:DP:GQ:MIN_DP:PL:LEN	0/0:25:45:25:0,42,630:4
1	4389	.	T	TC,<*>	213.73	.	.	GT:DP:GQ:PL:LEN	0/1:23:99:51,0,36,93,92,86
1	4390	.	C	<*>	.	.	END=4390	GT:DP:GQ:MIN_DP:PL:LEN	0/0:26:0:26:0,0,315:1
1	4391	.	C	<*>	.	.	END=4395	GT:DP:GQ:MIN_DP:PL:LEN	0/0:27:63:27:0,63,945:4
1	4396	.	G	C,<*>	0	.	.	GT:DP:GQ:~~P~~MIN_DP:PL:LEN	0/0:24:52:0,52,95,66,95,97
1	4397	.	T	<*>	.	.	END=4416	GT:DP:GQ:MIN_DP:PL:LEN	0/0:22:14:22:0,15,593:19

[¶] [¶]https://help.basespace.illumina.com/articles/descriptive/gvcf-files/

5.6 Representing copy number variation

To encode copy number variation, VCF uses <CNV>, <DEL> and <DUP> symbolic structural variant alleles, CN INFO and FORMAT fields.

Allele specific copy number is specified through a <CNV> ALT allele for each distinct allelic copy number. INFO CN defines the allele specific copy number with FORMAT CN defining the overall copy number for that sample. POS and INFO SVLEN specify the genomic interval over which the copy number is defined. <DEL> and <DUP> copy number (SVCLAIM=D) alleles should be treated as <CNV> alleles that implicitly define INFO CN=0 and ~~INFO CICN~~CN=2, ~~.~~respectively. As with all symbolic structural variants, the starting position of the interval is the base immediately after POS. For example, a region on chr1 from position 101 to 130 (both inclusive) with allele-specific copy numbers of 1 and 2 can be represented as follows:

```
~~chr1 100 . T <CNV>,<CNV> . . END=130;SVLEN=30,30;CN=1,2 GT:CN 1/2:3~~
chr1 100 . T <CNV>,<CNV> . . SVLEN=30,30;CN=1,2 GT:CN 1/2:3
```

All <CNV> alleles in the same VCF record should have the same SVLEN. To eliminate genotype ambiguity, copy number ALT alleles should not be mixed with other ALT alleles. When only copy number ALT alleles are present in a VCF record, GT=0 is equivalent to a <CNV> ALT allele with INFO CN of 1 and should be treated identically.

If only total copy number is known, the copy number of the segment should be defined with a single <CNV> ALT allele with a missing INFO CN field. In the above example this corresponds to the following:

```
~~chr1 100 . T <CNV> . . END=130;SVLEN=30 GT:CN .:3~~
chr1 100 . T <CNV> . . SVLEN=30 GT:CN .:3
```

The granularity of copy number representation is explicitly not defined in these specifications. Copy number segmentation can be base-pair accurate with even 1bp changes deletions resulting in new copy number segments, be at a highly granular megabase level of resolution, or anywhere in between. When the bounds of a copy number segment is not known precisely, this should be encoded in the CIPOS and CILEN INFO fields.

```
##fileformat=VCFv4.5
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the longest variant described in this record">
##INFO=<ID=SVLEN,Number=A,Type=Integer,Description="Length of structural variant">
##INFO=<ID=CN,Number=A,Type=Float,Description="Copy number of allele">
##INFO=<ID=RN,Number=A,Type=Integer,Description="Total number of repeat sequences in this allele">
##INFO=<ID=RUS,Number=.,Type=String,Description="Repeat unit sequence of the corresponding repeat sequence">
##INFO=<ID=RUL,Number=.,Type=Integer,Description="Repeat unit length of the corresponding repeat sequence">
##INFO=<ID=RUC,Number=.,Type=Float,Description="Repeat unit count of corresponding repeat sequence">
##INFO=<ID=RB,Number=.,Type=Integer,Description="Total number of bases in the corresponding repeat sequence">
##INFO=<ID=CIRUC,Number=.,Type=Float,Description="Confidence interval around RUC">
##INFO=<ID=CIRB,Number=.,Type=Integer,Description="Confidence interval around RB">
##INFO=<ID=RUB,Number=.,Type=Integer,Description="Number of bases in each individual repeat unit">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=PS,Number=1,Type=Integer,Description="Phase set">
##FORMAT=<ID=CN,Number=1,Type=Float,Description="Copy number">
##ALT=<ID=CNV:TR,Description="Tandem repeat determined based on DNA abundance">
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT sample
chr1 100 cnv_notation T <CNV:TR>,<CNV:TR> . . END=130;SVLEN=30,30;CN=3,0.9666;RUS=CAG,CAG,CA,CAG;RN=1,3;RB=90,15,2,12 GT:PS:CN 1|2:100:3.9666
chr1 100 cnv_notation T <CNV:TR>,<CNV:TR> . . SVLEN=30,30;CN=3,0.9666;RUS=CAG,CAG,CA,CAG;RN=1,3;RB=90,15,2,12 GT:PS:CN 1|2:100:3.9666
chr1 117 precise_alt2 AG A . . GT:PS 0|1:100
chr1 130 precise_alt1 G GCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAG . . GT:PS 1|0:100
```

Note the following:

- As with all symbolic structural variant alleles, the POS of the <CNV:TR> record is the base immediately preceding the tandem repeat
- The SVLEN of the <CNV:TR> is the length of the reference allele. It is not the length of the <CNV:TR> allele.
- The SVLEN of the <CNV:TR> allele of a novel (with respect to the reference) tandem repeat should be 1.
- The POS of the <CNV:TR> allele of a novel (with respect to the reference) tandem repeat should be the base immediately preceding the inserted tandem repeat sequence.
- Both a <CNV:TR> and one or more non-symbolic records encoding the tandem repeat can be present.
- <CNV:TR> and the non-symbolic records encoding the tandem repeat should be phased if possible.
- When both <CNV:TR> and the equivalent non-symbolic records are present, the <CNV:TR> should approximately encode the sequence but is not required to encode the sequence exactly. For example, SNVs and indels may be omitted in the <CNV:TR> record.
- Variant callers which do not report allele-specific tandem repeats should use a single <CNV:TR> ALT allele and the missing genotype for the GT field (for example, ./. if diploid).
- The INFO and FORMAT CN fields should be present for <CNV:TR> records (as they are <CNV> records) and, when present, must correspond to the sample allelic length divided by the reference allelic length. Note that CN FORMAT field represents the overall copy number and the INFO CN the allele-specific copy number.
- When benchmarking tandem repeats, the <CNV:TR> interval provides a region over which a set of (preferably phased) non-symbolic records can be compared against for length and sequence composition.
- RN encodes the number of records for each allele in the RUS, RUL, RUC, RB fields. Conceptually, this a mechanism to encode a list-of-list with flat list format used in VCF.
- STRs should use RUS, whereas VNTRs should use RUL to ensure the VCF records are not excessively large.
- RUL should be omitted when RUS is present (as it is redundant when RS is present).
- RUS or RUL must be specified for each <CNV:TR>.
- Support for multiple levels of repeat nesting (such as STRs within VNTRs) is limited to the RUL repeat unit length field which allows the overall length of each top-level repeat unit to be encoded.
- The POS and ~~END~~ SVLEN of <CNV:TR> records should match the STR/VNTR reference catalog sizes for catalog-based callers.
- Variant normalisation has limited utility in regions of low complexity as almost identical haplotypes can have very different normalised representations.

In some cases, it is desirable to report the full repeat sequence of all alleles at a given repeat locus in a single VCF records. There are no restrictions on doing so and in the above example, instead of reporting *precise_alt*1 and *precise_alt*2, the variants can be represented directly in a single record with a *REF* of:

```
CAGCAGCAGCAGCAGCAGCAGCAGCAG
```

and an *ALT* of:

```
CAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAG,CAGCAGCAGCAGCAGCACAGCAGCAGCAG
```

When the length or number of repeat units in a repeat sequence cannot be determined precisely, CIRB and/or CIRUC can be used to define the bounds. For example, if the total number of CAG repeats at the above locus was at least 50 ($(CAG)_{50-}$) and the mostly likely number of repeats was 65, then the <CNV:TR> could be encoded as follows:

```
~~chr1 100 . T <CNV:TR> . . END=130;SVLEN=30;CN=6.5;RUS=CAG;RUC=65;CIRUC=-15,. GT ./.~~
chr1 100 . T <CNV:TR> . . SVLEN=30;CN=6.5;RUS=CAG;RUC=65;CIRUC=-15,. GT ./.
```

Note that:

- RN was omitted as it is only required if at least one <CNV:TR> allele has RN greater than 1.
- The confidence interval bounds are relative to the nominal value.
- A missing upper bound indicates the maximum length is not known.

Exactly representing nested repeats results in the loss of some repeat information when representing with a <CNV:TR> record. For repeats such as $((ACCGGC)_4(ACCAGT))_{3-5}$, summarising the repeat structure in a <CNV:TR> record requires either unrolling the inner repeats, or treating each outer repeat as a separate repeat sequence (the full repeat structure can be stored in a caller-specific non-standard INFO field). For many VNTRs, the critical information to retain is the length of each repeat unit. This length information can be encoded in the RUB field. For example, a 10,000bp VNTRs domain repeated 5 times, each repeat 500bp longer than the previous can be encoded as follows:

~~chr1 1000000 . T <CNV:TR> . . END=20000;SVLEN=20000;CN=1.25;RUL=10000;RUC=5;RUB=10000,10500,11000,11500,12000 GT ./.~~
chr1 1000000 . T <CNV:TR> . . SVLEN=20000;CN=1.25;RUL=10000;RUC=5;RUB=10000,10500,11000,11500,12000 GT ./.

6.3.1 Site encoding

Field	Type	Notes
l_shared	uint32_t	Data length from CHROM to the end of INFO
l_indiv	uint32_t	Data length of FORMAT and individual genotype fields
CHROM	int32_t	Given as an offset into the mandatory contig dictionary
POS	int32_t	0-based leftmost coordinate
rlen	int32_t	Length of the record as projected onto the reference sequence. Must be the maximum of the length of the REF allele and the lengths inferred from the SVLEN/~~END~~ LEN of any symbolic alleles
QUAL	float	Variant quality; 0x7F800001 for a missing value
n_info	uint16_t	The number of INFO fields in this record
n_allele	uint16_t	The number of REF+ALT alleles in this record
n_sample	uint24_t	The number of samples in this record, stored as a **three** byte little-endian value. Note that n_sample must be equal to the number of samples in the header
n_fmt	uint8_t	The number of FORMAT keys. See 6.3.2
ID	typed string	Variant identifier; 0x07 for a missing value
REF+ALT	list of n_allele typed strings	the first allele is REF (mandatory) followed by n_alleles - 1 ALT alleles, all encoded as typed strings
FILTER	Typed vector of integers	a vector of integer offsets into dictionary, one for each FILTER field value. "." is encoded as MISSING
INFO	field key/value pairs	n_info pairs of typed vectors. The first value must be a typed atomic integer giving the offset of the INFO field key into the dictionary. The second value is a typed vector giving the value of the field
Genotype values	see below	see below

6.3.2 Genotype encoding

Genotype fields are encoded not by sample as in VCF but rather by field, with a vector of values for each sample following each field. In BCF2, the following VCF line:

FORMAT	NA00001	NA00002	NA00003
GT:GQ:DP	0/0:48:1	0/1:9:8	1/1:43:5

would encoded as the equivalent of:

GT=0/0,0/1,1/1 GQ=48,9,43 DP=1,8,5

Suppose there are i genotype fields in a specific record. Each i is encoded by a triplet:
BCF2 site information encoding

Field	Type	Notes
fmt_key	typed int	Format key as an offset into the dictionary
fmt_type	uint8_t+	Typing byte of each individual value, possibly followed by a typed int for the vector length. In effect this is the same as the typing value for a single vector, but for genotype values it appears only once before the array of genotype field values
fmt_values (by fmt type)	Array of values	The information of each individual is concatenated in the vector. Every value is of the same fmt type. Variable-length vectors are padded with END_OF_VECTOR values; a string is stored as a vector of char

The value is always implicitly a vector of N values, where N is the number of samples. The type byte of the value field indicates the type of each value of the N length vector. For atomic values this is straightforward (size = 1). But if the type field indicates that the values are themselves vectors (as often occurs, such as with the PL field) then each of the N values in the outer vector is itself a vector of values. This encoding is efficient when every value in the genotype field vector has the same length and type.

It is recommended to respect the ordering as specified in the input VCF/BCF2 file, but parsers should not rely on a specific ordering.

If there are no sample records (genotype data) in this VCF/BCF2 file, the size of the genotypes block will be 0.

0x33000000	l_shared as 32-bit little endian hex
0x2A000000	l_indiv as 32-bit little endian hex
0x01000000	CHROM offset is at 1 in 32 bit little endian
0x64000000	POS in 0-based 32-bit little endian
0x01000000	rlen = 1 (it's just a SNP)
0x41 0xF0 0xCC 0xCD	QUAL = 30.1 as 32-bit float
0x0400	n_info as 16-bit little-endian
0x0200	n_allele as 16-bit little-endian
0x030000	n_sample as 24-bit little-endian
0x05	n_fmt
0x57 0x72 0x73 0x31 0x32 0x33	ID = rs123
0x17 0x41	REF A
0x17 0x43	ALT C
0x11 0x00	FILTER field PASS
0x11 0x50 0x00	HM3 flag is present
0x11 0x51	AC key
0x11 0x03	with value of 3
0x11 0x52	AN key
0x11 0x06	with value of 6
0x11 0x53	AA key
0x17 0x43	with value of C
0x1101 0x21 0x020202040404	GT
0x1102 0x11 0x0A0A0A	GQ
0x1103 0x11 0x203040	DP
0x1104 0x21 0x300030200040	AD
0x1105 0x31 0x000A640A0064640A00	PL

That's quite a lot of information encoded in only 96 bytes!

6.5 BCF2 block gzip and indexing

These raw binary records may be subsequently encoded into BGZF blocks following the BGZF compression format, section 3 of the SAM format specification. BCF2 records can be raw, though, in cases where the decoding/encoding costs of bgzipping the data make it reasonable to process the data uncompressed, such as streaming BCF2s through pipes with samtools and bcftools. Here the files should be still compressed with BGZF but with compression 0. Implementations should perform BGZF encoding and must support the reading of both raw and BGZF encoded BCF2 files.

BCF2 files are expected to be indexed through the same index scheme, section 4 as BAM files and other block-compressed files with BGZF.

7 List of changes

7.1 Changes between VCFv4.5 and VCFv4.4

- Added Number=P support for fields with cardinality matching sample ploidy/local copy number.
- Added local allele support (Number=LA, LG, LR; FORMAT LAA, LAD, LADF, LADR, LEC, LGL, LGP, LGT, LPL, LPP) to reduce the size of multi-sample VCFs and enable lossless merging.
- Deprecated INFO END. It is now a computed field written only for backwards compatibility with older versions of VCF.
- Added FORMAT LEN to support sample-specific <*> alleles.

7.2 Changes between VCFv4.4 and VCFv4.3

- Added tandem repeat support (<CNV:TR>, RN, RUS, RUL, RB, CIRB, RUC, CIRUC, RUB)