1 The VCF specification

VCF is a text file format (most likely stored in a compressed manner). It contains meta-information lines (prefixed with "`##`"), a header line (prefixed with "`#`"), and data lines each containing information about a position in the genome and genotype information on samples for each position (text fields separated by tabs). Zero length fields are not allowed, a dot (".") must be used instead. In order to ensure interoperability across platforms, VCF compliant implementations must support both LF ("`\n`") and CR+LF ("`\r\n`") newline conventions.

1.1 An example

```
##fileformat=VCFv4.5
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",taxonomy=x>
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of \begin{environment-name}
      Samples
\end{environment-name} With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF    ALT     QUAL FILTER INFO                              FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G      A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20     17330   .         T      A       3    q10    NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3   0/0:41:3
20     1110696 rs6040355 A      G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2   2/2:35:4
20     1230237 .         T      .       47   PASS   NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTC    G,GTCT  50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

This example shows (in order): a good simple SNP, a possible SNP that has been filtered out because its quality is below 10, a site at which two alternate alleles are called, with one of them (T) being ancestral (possibly a reference sequencing error), a site that is called monomorphic reference (i.e. with no alternate alleles), and a microsatellite with two alternative alleles, one a deletion of 2 bases (TC), and the other an insertion of one base (T). Genotype data are given for three samples, two of which are phased and the third unphased, with per sample genotype quality, depth and haplotype qualities (the latter only for the phased samples) given as well as the genotypes. The microsatellite calls are unphased.

1.2 Character encoding, non-printable characters and characters with special meaning

The character encoding of VCF files is UTF-8. UTF-8 is a multi-byte character encoding that is a strict superset of 7-bit ASCII and has the property that none of the bytes in any multi-byte characters are 7-bit ASCII bytes. As a result, most software that processes VCF files does not have to be aware of the possible presence of multi-byte UTF-8 characters. VCF files must not contain a byte order mark. Note that non-printable characters U+0000–U+0008, U+000B–U+000C, U+000E–U+001F are disallowed. Line separators must be CR+LF or LF and they are allowed only as line separators at end of line. Some characters have a special meaning when they appear (such as field delimiters '`;`' in INFO or '`:`' FORMAT fields), and for any other meaning they must be represented with the capitalized percent encoding:

%3A	:	(colon)
%3B	;	(semicolon)
%3D	=	(equal sign)
%25	%	(percent sign)
%2C	,	(comma)
%0D	CR	
%0A	LF	
%09	TAB	

5. ALT — alternate base(s): Comma-separated list of alternate non-reference alleles. These alleles do not have to be called in any of the samples. Each allele in this list must be one of: a non-empty String of bases (A,C,G,T,N; case insensitive); the '*' symbol (allele missing due to overlapping deletion); the MISSING value '.' (no variant); an angle-bracketed ID String ("<ID>"); the unspecified allele "<*>" as described in Section 5.5; or a breakend replacement string as described in Section 5.4. If there are no alternative alleles, then the MISSING value must be used. Tools processing VCF files are not required to preserve case in the allele String, except for IDs, which are case sensitive. (String; no whitespace, commas, or angle-brackets are permitted in the ID String itself)

6. QUAL — quality: Phred-scaled quality score for the assertion made in ALT. i.e. $-10log_{10}$ prob(call in ALT is wrong). If ALT is '.' (no variant) then this is $-10log_{10}$ prob(variant), and if ALT is not '.' this is $-10log_{10}$ prob(no variant). If unknown, the MISSING value must be specified. (Float)

7. FILTER — filter status: PASS if this position has passed all filters, i.e., a call is made at this position. Otherwise, if the site has not passed all filters, a semicolon-separated list of codes for filters that fail. e.g. "q10;s50" might indicate that at this site the quality is below 10 and the number of samples with data is below 50% of the total number of samples. '0' is reserved and must not be used as a filter String. If filters have not been applied, then this field must be set to the MISSING value. (String, no whitespace or semicolons permitted, duplicate values not allowed.)

8. INFO — additional information: Semicolon-separated series of additional information fields, or the MISSING value '.' if none are present. Each subfield consists of a short *key* with optional *values* in the format: key[=value[,...,value]]. Literal semicolon ('`;`') and equals sign ('`=`') characters are not permitted in these values, and literal commas ('`,`') are permitted only as delimiters for lists of values; characters with special meaning can be encoded using percent encoding, see Section 1.2. Space characters are allowed in values.

   INFO keys must match the regular expression `^([A-Za-z_][0-9A-Za-z_.]*|1000G)$`, please note that "1000G" is allowed as a special legacy value. Duplicate keys are not allowed. Arbitrary keys are permitted, although those listed in Table 1 and described below are reserved (albeit optional).

   The exact format of each INFO key should be specified in the meta-information (as described above). Example of a complete INFO field: `DP=154;MQ=52;H2`. Keys without corresponding values may be used to indicate group membership (e.g. H2 indicates the SNP is found in HapMap 2). See Section 3 for additional reserved INFO keys used to encode structural variants.

Key	Number	Type	Description
AA	1	String	Ancestral allele
AC	A	Integer	Allele count in genotypes, for each ALT allele, in the same order as listed
AD	R	Integer	Total 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
AF	A	Float	Allele frequency for each ALT allele in the same order as listed (estimated from primary data, not called genotypes)
AN	1	Integer	Total number of alleles in called genotypes
BQ	1	Float	RMS base quality
CIGAR	A	String	Cigar string describing how to align an alternate allele to the reference allele
DB	0	Flag	dbSNP membership
DP	1	Integer	Combined depth across samples
END	1	Integer	~~Deprecated. Present for backwards compatibility with earlier versions of VCF.~~ End position of the longest variant described in this record
H2	0	Flag	HapMap2 membership
H3	0	Flag	HapMap3 membership
MQ	1	Float	RMS mapping quality
MQ0	1	Integer	Number of MAPQ == 0 reads

Continued on next page...

Table 1: Reserved INFO keys

... Continued from previous page

Key	Number	Type	Description
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: ~~Deprecated. Retained for backwards compatibility with earlier versions of VCF and older VCF indexing software which rely on this field being present.~~ End position of the longest variant described in this record

  This is a computed field that, when present, must be set to the maximum end reference position (1-based) of: the position of the final base of the REF allele, the end position corresponding to the SVLEN of a symbolic SV allele, and the end positions calculated from FORMAT LEN for the <*> symbolic allele.

  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~~.

  Whilst technically deprecated (INFO SVLEN and FORMAT LEN are the authoritative fields), END remains important for backwards compatibility.

  Unfortunately, the introduction of FORMAT LEN is not fully backwards compatible with END. END is used for VCF indexing and a large ecosystem of pre-VCFv4.5 tools rely on END being present. Those same tools will incorrectly interpret the size of the smaller symbolic structural variants and <*> symbolic alleles when END is present.

  It is recommended that VCFv4.5 files include END unless that VCF contains any record that could be misinterpreted by the presence of END. That is, if there exists a sample or allele in which the END computed for that SVLEN or FORMAT LEN does not equal the maximum END, then no END should be present in any record that VCF. This approachs maintains backwards compatibility for unproblematic VCFs while attempting to minimise the probability of downstream data errors by making problematic records not valid for earlier versions of VCF (END was required for <*> symbolic alleles).

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 any local-allele field is present, LAA must also be present and precede all fields other than GT. 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 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

Continued on next page...

Table 2: Reserved genotype keys

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).

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

~~*END* has been deprecated in favour of INFO SVLEN and FORMAT LEN~~Refer to section ?? for the definition of END.

```
##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.

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

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 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">
```

Whilst simple events such as deletions and duplications can be wholly represented by a single VCF record, complex rearrangements such as chromothripsis result in a large number of breakpoints. VCF uses the $EVENT$ field to group such related records together, and $EVENTTYPE$ to classify these events. All records with the same $EVENT$ value are considered to be part of the same event.

The following $EVENTTYPE$ values are reserved and should be used when appropriate:

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 <*> 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).

Positions implicitly called by a preceding <*> for a sample must have *GT* 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* 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.

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: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

Note that usage of both FORMAT LEN and INFO END can be problematic as pre-VCFv4.5 tools will misinterpret the reference block size for records containing samples with different block sizes. See the definition of INFO END in section **??** for recommended behaviour.

When base modification information is present in the FORMAT field of a reference block record, the base modification information apply to all applicable bases covered by that reference block.

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

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 VCFv4.5 Errata

- Clarified INFO END deprecation status.

7.2 Changes between VCFv4.5 and VCFv4.4

- Added base modification support (FORMAT M5mC, M5hmC, M6mA, etc.).
- Reserved all FORMAT keys of the form $M[0-9]+$ as base modification fields.
- 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, LPL, LPP) to reduce the size of multi-sample VCFs and enable lossless merging.