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
- 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.
- 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 specified in the `GT`. 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 ~~derivate~~ 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

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

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 ~~derivate~~ derivative chromosome traversal order, multiple ~~derivate~~ 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:.,.,4,1`
chr1	30	G	T	.	GT:PSL:PSO	`/0/0|0|1:.,chr1*10*1:.,.,2,5`

Without defining PSO, 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).

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:

- 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).

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.

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. $MEINFO$ 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:

- DEL - Deletion
- DEL:ME - Deletion of mobile element with respect to the reference
- INS - Insertion
- INS:ME - Insertion of mobile element
- DUP - Duplication
- DUP:TANDEM - Tandem duplication
- DUP:DISPERSED - Dispersed duplication
- INV - Inversion
- TRA - Translocation
- TRA:BALANCED - Balanced inter-chromosomal translocation
- TRA:UNBALANCED - Unbalanced inter-chromosomal translocation
- CHROMOTHRIPSIS - Chromothripsis
- CHROMOPLEXY - Chromoplexy
- BFB - breakage fusion bridge
- DOUBLEMINUTE - Double minute

The ~~sematics~~ semantics of other $EVENTTYPE$ values is implementation-defined. The use of $EVENT$ is not restricted to structural variation and can also be used to associate non-symbolic alleles. Such linking is useful for scenarios such as kataegis or when there is variant position ambiguity in segmentally duplicated regions.

```
##INFO=<ID=CILEN,Number=.,Type=Integer,Description="Confidence interval for the SVLEN field">
```

If present, the number of entries must be twice the number of ALT alleles. $CILEN$ consists of successive pairs of records indicating the lower and upper bounds of the $SVLEN$ confidence interval.

```
##INFO=<ID=RN,Number=A,Type=Integer,Description="Total number of repeat sequences in this allele">
```

Used by <CNV:TR> tandem repeat alleles to encode the number of repeat sequences. This field determines the number of values encoded in the RUS, RUL, RB, CIRB, RUC and CIRUC fields for each allele. The length of these fields must match the sum of all RN values for the record. For the purposes of determining RUS, RUL, RB, CIRB, RUC and CIRUC lengths, the missing value "." should be treated as 0. If this field is missing, the RN for each <CNV:TR> allele is assumed to be 1 and 0 otherwise.

See section 5.6 for further details.

```
##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">
```

Used by <CNV:TR> tandem repeat alleles to encode information about the nature of the tandem repeats contained for ALT alleles. Conceptually, these fields each contain a list of values for each ALT allele. The length of these inner lists are determined by the RN field for that ALT allele and the length must match the sum of RN for the record. These fields contain the flattened and ~~concatentated~~ concatenated list contents in the same order as either corresponding ALT allele.

Each <CNV:TR> allele consistents of RN repeat sequences each containing RUC repeat units with sequence RUS.

For example, if a <CNV:TR> allele sequence is $(CAG)_{10}(TG)_7(CAGG)_3$, the RN for that ALT allele would be 3, the RUS $CAG, TG, CAGG$, the RUL $3, 2, 4$, the RUC $10, 7, 3$ and the RB $30, 14, 12$.

RUS may contain only IUPAC nucleotide codes (ambiguous bases are allowed) or the missing value ('.'). If both RUS and RUL are present and not missing, the length of each RUS entry must match the corresponding RUL. If both RUC and RB are present and not missing, RB should approximately equal RUL*RUC but is not required to match exactly. If RUS is present but RUL is not, RUL should be inferred to be equal to the length of the corresponding RUS.

See section 5.6 for an example and further details.

```
##INFO=<ID=CIRUC,Number=.,Type=Float,Description="Confidence interval around RUC">
##INFO=<ID=CIRB,Number=.,Type=Integer,Description="Confidence interval around RB">
```

Confidence interval around RUC and RB respectively. The length of these fields must be twice that of their corresponding fields. CIRUC/CIRB must not be non-missing for any alleles with no corresponding RUC/RB value.

These fields are defined in the same manner as $CIPOS$ and contain the difference between the lower and upper confidence interval bounds and the value of the corresponding field. The lower bound must be less than or equal to zero and the upper bound must be greater than or equal to zero. If the lower bound is the missing value ".", is assumed to be 0 and if the upper bound is the missing value ".", it is assumed to be an unbounded estimate. That is, the length of repeat has been determined to be at least a certain length but a reasonable limit of total length of the repeat could not be determined.

See section 5.6 for an example and further details.

```
##INFO=<ID=RUB,Number=.,Type=Integer,Description="Number of bases in each individual repeat unit">
```

This field encodes the length of each repeat unit for situations where the length of each repeat unit varies greatly. If this field is present, RUC must also be present and must contain only integer values. This field uses the same list-of-list encoding as RUS/RUL/RUC/RB but contains a list for each RC entry, the length of which is determined by the corresponding integer RUC value. This field contains the length of each individual repeat unit for each RUC entry. If RUB is missing or not specified, The RUB for each individual repeat unit is considered to be equal to the RUL for the corresponding repeat sequence.

For the vast majority of tandem repeats, this field can be omitted and is only required in complex situations such as when a VNTR contains one or more variable length STRs.

See section 5.6 for an example and further details.

4 FORMAT keys used for structural variants

```
##FORMAT=<ID=CN,Number=1,Type=Float,Description="Copy number">
##FORMAT=<ID=CNQ,Number=1,Type=Float,Description="Copy number genotype quality">
##FORMAT=<ID=CNL,Number=G,Type=Float,Description="Copy number genotype likelihood">
```

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-symoblic~~ non-symbolic records encoding the tandem repeat can be present.
- <CNV:TR> and the ~~non-symoblic~~ non-symbolic records encoding the tandem repeat should be phased if possible.
- When both <CNV:TR> and the equivalent ~~non-symoblic~~ 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 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 $precise_a lt1$ and $precise_a lt2$, the variants can be represented directly in a single record with a REF of $CAGCAGCAGCAGCAGCAGCAGCAGCA$ and an ALT of $CAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGC$

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 ./.
```

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 ~~bouns~~ bounds 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 stroed 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 ./.
```

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 ~~maxmimum~~ maximum of the length of the REF allele and the length inferred from the SVLEN/END 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 VCFv4.4 and VCFv4.3

- Added tandem repeat support (<CNV:TR>, RN, RUS, RUL, RB, CIRB, RUC, CIRUC, RUB)
- Added support for phasing and ~~derivate~~ derivative chromosome reconstruction in the presence of SVs (PSL, PSO, PSQ)
- Added SVCLAIM to disambiguate copy number based <DEL> and <DUP> variants from breakpoint based ones.
- Conceptually separated variant detection and interpretation.
- Added EVENTTYPE/EVENT to enable the multiple records encoding complex genomic rearrangements to be grouped together.
- Added polyploid partial phasing support (e.g. GT |0|0/1/2). GT now defined as a prefix notation with the first phasing indicator optional.