The Variant Call Format Dual Coordinates Extension (DVCF) Specification

Version 1.0
May 17, 2021

Written by: Divon Lan, divon@shablife.com

Citing:
Lan, D (2021) The Variant Call Format - Dual Coordinates Extension (DVCF) Specification
doi:10.6084/m9.figshare.14685816 (preprint)

Table of contents

1. Background

This specification is fully compatible with the VCFv4.3 specification, and extends it. It is also fully compatible with VCF v4.1 and v4.2.

The specification defines a derived format of VCF, fully compliant with the VCF specification, which is called the Dual Coordinates VCF file (or *DVCF*). A DVCF file contains information about genetic variants in two different coordinate systems. The key feature of DVCF is that it can be *rendered* in two different ways - the *Primary rendition* and *Luft rendition*. Both these renditions are VCF specification-compliant files, that contain precisely the same information, merely *rendered* in two different coordinate systems.

Since these two renditions contain precisely the same information, they can be losslessly *cross-rendered* back and forth. Cross-rendering is a fast operation that does not require a reference or chain file.

Once a VCF file is *lifted* to a Dual Coordinate VCF file - it can be processed through an analytical pipeline, and since the data can be rendered in either coordinate system, each stage of the pipeline can arbitrarily operate on either coordinate system. Importantly, the rendering continues to work as fields and annotations are added, removed or modified, as the data works its way down the pipeline.

This specification was intentionally made to be similar to the VCF specification in format, structure and terminology, and is designed to be read alongside it. All the definitions and requirements that appear in the VCF specification apply here as well, and they are not repeated in this document.

This specification was written in context of a PhD project at the University of Adelaide, Australia. I wish to thank my PhD supervisors Assoc. Prof. Bastien Llamas, Dr. Yassine Souilmi and Dr. Ray Tobler, as well as my wife, Channé Suy Lan, for their support which has been absolutely essential.

2. Definitions

- A *Source VCF* is any VCF file that is compliant with the VCF specification.

- The *Primary coordinate system* is the coordinate system of the Source VCF.

- The *Luft coordinate system* is the other coordinate system in which the variant data will be expressed ("Luft" being a made-up past participle of "Lift").

- A *Primary rendition* and a *Luft rendition* are VCF files expressed in the Primary and Luft coordinates respectively, which are equivalent to each other and contain all the information of a Source VCF along with all the information needed to *cross-render* them (see below). A DVCF is always rendered in one or both of these two renditions, and this specification defines no other representation of a DVCF other than the renditions.

- A *Lifter* is a software functionality that converts, or *Lifts*, a Source VCF to a DVCF. It may use auxiliary information such as a reference file in the Luft coordinates and a chain file.

- A *Renderer* is a software functionality that generates the Primary and Luft renditions. It may *cross-render* a Primary rendition to a Luft one or vice versa, or may generate a Primary or Luft rendition from some other data. Cross-rendering does not require any external information beyond the input DVCF file itself. Specifically, it does not require a reference file or a chain file.

- A *Dual Coordinates VCF implementation* (or just *implementation* for brevity) means a particular software package including the functionalities of a Lifter and/or a Renderer.

3. Scope of this specification

This specification defines the formats of the DVCF Primary and Luft renditions..

It does not define the algorithms of a Lifter or Renderer, however it does set constraints on them, to ensure that the Primary and Luft renditions contain precisely the same information, and to ensure interoperability between implementations. While adhering to these constraints, different implementations of *Lifters* and *Renderers* might operate differently to address different needs.

Complying with this specification will ensure that the resulting files are interoperable across different systems.

It is desirable that any software that converts a VCF file from one coordinate system to another, shall be capable of outputting VCF files in DVCF format.

4. An Example

The following are the two *renditions* of the same DVCF - they are two files containing precisely the same information - the first file is the *Primary rendition* in GRCh37 coordinates, and the second is the *Luft rendition* in GRCh38 coordinates. This DVCF contains 3 variants and 2 samples.

A *Primary rendition VCF* file:

```
##fileformat=VCFv4.2
##dual_coordinates=PRIMARY
##chain=file:///data/GRCh37_to_GRCh38.chain.genozip
##reference=file:///references/grch37/reference.bin
##luft_reference=file:///data/GRCh38_full_analysis_set_plus_decoy_hla.ref.genozip
##FILTER=<ID=PASS,Description="All filters passed">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles",RendAlg=R>
##FORMAT=<ID=AF,Number=A,Type=Float,Description="Allele fractions for alt alleles in the order listed",RendAlg=A_1>
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype",RendAlg=GT>
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scaled likelihoods for genotypes",RendAlg=G>
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT allele",RendAlg=A_AN>
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes",RendAlg=NONE>
##INFO=<ID=LUFT,Number=4,Type=String,Description="Info for rendering variant in LUFT coords",RendAlg=NONE>
##INFO=<ID=PRIM,Number=4,Type=String,Description="Info for rendering variant in PRIMARY coords",RendAlg=NONE>
##INFO=<ID=Lrej,Number=1,Type=String,Description="Reason variant was rejected for LUFT coords",RendAlg=NONE>
##INFO=<ID=Prej,Number=1,Type=String,Description="Reason variant was rejected for PRIMARY coords",RendAlg=NONE>
##contig=<ID=1,length=249250621>
##luft_contig=<ID=chr1,length=248956422>
#CHROM POS    ID     REF    ALT    QUAL   FILTER INFO   FORMAT Person1       Person2
1      10285  .      T      C      4.4    PASS   AC=3;AN=4;LUFT=chr1,10285,T,-     GT:AD:AF:PL   0/1:31,18:0.367:37,0,46    1/1
1      329162 .      A      T      4.6    PASS   AC=3;AN=4;LUFT=chr1,248466248,T,- GT:AD:AF:PL   0/1:28,9:0.3:36,0,0 1/1
1      366043 .      CA     A      100    PASS   Lrej=RefTooLong     GT     1|0    0|0
```

A *Luft rendition VCF* file corresponding to the *Primary rendition* on the previous page:

```
##fileformat=VCFv4.2
##dual_coordinates=LUFT
##chain=file:///data/GRCh37_to_GRCh38.chain.genozip
##reference=file:///data/GRCh38_full_analysis_set_plus_decoy_hla.ref.genozip
##primary_reference=file:///references/grch37/reference.bin
##FILTER=<ID=PASS,Description="All filters passed">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths for the ref and alt alleles",RendAlg=R>
##FORMAT=<ID=AF,Number=A,Type=Float,Description="Allele fractions for alt alleles in the order listed",RendAlg=A_1>
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype",RendAlg=GT>
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scaled likelihoods for genotypes",RendAlg=G>
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes, for each ALT allele",RendAlg=A_AN>
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes",RendAlg=NONE>
##INFO=<ID=LUFT,Number=4,Type=String,Description="Info for rendering variant in LUFT coords",RendAlg=NONE>
##INFO=<ID=PRIM,Number=4,Type=String,Description="Info for rendering variant in PRIMARY coords",RendAlg=NONE>
##INFO=<ID=Lrej,Number=1,Type=String,Description="Reason variant was rejected for LUFT coords",RendAlg=NONE>
##INFO=<ID=Prej,Number=1,Type=String,Description="Reason variant was rejected for PRIMARY coords",RendAlg=NONE>
##primary_contig=<ID=1,length=249250621>
##contig=<ID=chr1,length=248956422>
##primary_only=1    366043 .      CA     A      100    PASS   Lrej=RefTooLong     GT     1|0    0|0
#CHROM POS    ID     REF    ALT    QUAL   FILTER INFO   FORMAT Person1       Person2
chr1   10285  .      T      C      4.4    PASS   AC=3;AN=4;PRIM=1,10285,T,- GT:AD:AF:PL  0/1:31,18:0.367:37,0,46    1/1
chr1   248466248    .      T      A      4.6    PASS   AC=1;AN=4;PRIM=1,329162,A,-       GT:AD:AF:PL  1/0:9,28:0.7:0,0,36 0/0
```

5. Meta-information lines

The following meta-information lines are added or modified. They may appear in any order.

5.1. Coordinates

`##dual_coordinates=PRIMARY`

This field is required.

Permitted values: `PRIMARY`, `LUFT`. Defines the coordinates of the current rendition.

5.2. Chain file URL

This field is recommended.

`##chain=file:///data/GRCh37_to_GRCh38.chain.genozip`

The URL of the chain file used by the Lifter to generate this DVCF. The file format and naming conventions of the chain file are implementation-specific and out of scope of this specification.

5.3. Reference files' URLs

These fields are recommended.

In a Primary rendition it is recommend to include the `##reference` and `##luft_reference` lines. The former contains the URL of the reference file of the Primary coordinates, and `##luft_reference` contains the URL of the reference file of the Luft coordinates.

`##reference=file:///data/hg19.p13.plusMT.full_analysis_set.ref.genozip`

`##luft_reference=file:///data/GRCh38_full_analysis_set_plus_decoy_hla.ref.genozip`

Similarly, in a Luft rendition, it is recommended to include `##reference` (Luft coordinates reference file) and `##primary_reference` :

`##reference=file:///data/GRCh38_full_analysis_set_plus_decoy_hla.ref.genozip`

```
##primary_reference=file:///data/hg19.p13.plusMT.full_analysis_set.ref.
genozip
```

The file format and naming conventions of the reference files are implementation-specific and out of scope of this specification.

5.4. RendAlg attribute of ##INFO and ##FORMAT

```
##FORMAT=<ID=GL,Number=G,Type=Float,Description="Genotype
Likelihoods",RendAlg=G>

##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in
genotypes, for each ALT allele, in the same order as
listed",RendAlg=A_AN>
```

The `RendAlg` attribute *must* be present in all `##INFO` and `##FORMAT` meta-information lines.

The *Renderer must* add `RendAlg` to `##INFO` or `##FORMAT` meta-information lines that are missing them. It *must not* modify a RendAlg value if one is already present. Lines might be missing `RendAlg` if, for example, the file acquired additional `INFO` or `FORMAT` fields in an analysis step.

5.5. LUFT, PRIM, Lrej and Prej

DVCF files must contain the following four meta-information lines defining `INFO/LUFT`, `INFO/PRIM`, `INFO/Lrej` and `INFO/Prej`. The `ID`, `Number, Type` and `RendAlg` attributes *must* appear as below, other attributes (such as `Description`) are optional.

```
##INFO=<ID=LUFT,Number=4,Type=String,RendAlg=NONE>
##INFO=<ID=PRIM,Number=4,Type=String,RendAlg=NONE>
##INFO=<ID=Lrej,Number=1,Type=String,RendAlg=NONE>
##INFO=<ID=Prej,Number=1,Type=String,RendAlg=NONE>
```

5.6. Contigs

The `##contig` key is as defined in the VCF specification. It refers to contigs of the current coordinates.

In a Primary rendition file, meta-information lines with a `##luft_contig` key may exist, and have the same format as `##contig`. They describe the contigs that appear in the Luft rendition. Similarly, a Luft rendition files may contain `##primary_contig` keys, describing the contigs in the Primary rendition. It is recommended that a DVCF file includes a `##luft_contig` or `##primary_contig` line for each contig that appears in the file.

Note that there is no requirement for a 1:1 mapping between contigs - indeed, it is possible that two variants with a particular contig in one coordinate system, are mapped to two different contigs on the other coordinate system.

5.7. ##primary_only and ##lift_only

In the *Primary rendition* `##luft_only` meta-information lines contain variants that are not renderable in Primary coordinates, and similarly, in the *Luft rendition*, `##primary_only` meta-information lines contain variants that are not renderable in Luft coordinates. Following the key at the '=' character, the remainder of the line is a normal VCF data line as defined in the VCF specification.

6. Variants

6.1. Overview

Each variant in the DVCF can be a dual-coordinate variant, or it could be a primary-coordinates-only variant or a luft-coordinates-only variant. The latter two cases happen when a variant can only be expressed in one of the coordinates but not in the other, in which case we also refer to it as *rejected* from the other coordinates. There are many reasons a variant can be rejected, discussed below.

Each variant, in both renditions, contains exactly one *DVCF tag*, which is an INFO field carrying DVCF-related information - one of: PRIM, LUFT, Prej or Lrej.

A dual-coordinate variant appears as a normal VCF variant in both renditions, and contains an INFO/LUFT field in the Primary rendition with the information needed to cross-render this variant to Luft coordinates, and similarly, in the Luft rendition, it contains an INFO/PRIM field with the information needed to cross-render the variant to Primary coordinates.

A primary-coordinates-only variant contains an INFO/Lrej field with the reason it was rejected for rendering in Luft coordinates. In the Primary rendition, the variant appears as a normal VCF data line, while the Luft rendition, this variant will appear as-is (i.e. in Primary coordinates) in a meta-information line with the key `##primary_only`.

Likewise, luft-coordinates-only variants have a INFO/Prej field, and appear as a data line in the Luft rendition, and as a meta-information line they key `##luft_only` in the Primary rendition.

6.2. CHROM, POS, REF, ALT

The *Lifter*, given a Source VCF and in consultation with external information, typically a reference file in the Luft coordinates and a chain file, should calculate the CHROM, POS, REF fields in the Luft coordinates. These *must* be biologically correct (the exact definition of *biologically correct* is left to the implementation) and either generate a *dual-coordinate variant* or a *primary-only variant*, i.e. one that was rejected from the Luft coordinates.

A *dual-coordinate variant* appears as a VCF data line in both Primary and Luft renditions:

- A dual-coordinate variant in the Primary rendition has the CHROM, POS, REF and ALT fields appear as in the Source VCF.

  It also has a INFO/LUFT field that contains four values, for example: ”`LUFT=chr2,1000000,G,X`”. The first two values are the CHROM and POS of this variant in Luft coordinates. The third is the Luft reference value of this variant. The fourth value, which we call XSTRAND, must be one of two options: it is `X` (capital letter X) if the alignment in the chain file which includes this locus has opposite strands for the Primary and Luft references and `-` (hyphen) if the strands are the same.

- A dual-coordinate variant in the Luft rendition has the values of CHROM, POS and REF as they appear in INFO/LUFT in the Primary rendition, and has the ALT calculated as described below.

  It also has an INFO/PRIM field that contains four values, of the same structure as INFO/LUFT: the first three values are the CHROM, POS and REF in Primary coordinates, and the fourth is the XSTRAND of this variant. XSTRAND *must* be the same value as in INFO/LUFT.

A primary-only variant appears in both renditions in primary coordinates. In the Primary rendition, it appears as a normal VCF data line, while in the Luft rendition, it appears as a `##primary_only` meta-information line. Apart from the "`##primary_only=`" prefix, the meta-information line is identical to the VCF data line as it appears in the Primary rendition.

In both renditions a primary-only variant has an INFO/Lrej field which contains the reason for its rejection. This specification defines a number of standard reasons, and an implementation may add additional reasons. The standard reasons are listed in section 7 of this document.

The *lifter* uses the information in CHROM, POS, REF and ALT as well as external information such as a chain file and a Luft reference file, to generate either a INFO/LUFT of INFO/Lrej field, while the *renderer* uses the information in the CHROM, POS, REF, ALT and INFO/LUFT or INFO/PRIM fields to calculate the CHROM, POS, REF, ALT and INFO/PRIM or INFO/LUFT respectively, of the other rendition, or it may reject the cross-rendering resulting in a Primary-only variant with an INFO/Lrej field, or a Luft-only variant with INFO/Prej field.

If the REF changes between Primary and Luft references, a *lifter* is free to either lift the variant or reject it, with the rejection reason placed in INFO/Lrej being one of the standard reasons listed in section 7, or an implementation-defined reason. If as a result of the REF change, the number of alleles grows because Luft REF is not any of the Primary alleles, then the Primary REF *must* be last on the Luft ALT list.

When calculating the ALT field, the algorithm used by the *lifter* and *renderer must:*

1. Be biologically-correct (the definition of *biologically correct* is left to the implementation).
2. Be precisely invertible, so that cross-rendering from the Luft rendition to the Primary rendition and back to the Luft rendition, as well as Primary → Luft → Primary results in the precisely preserving the REF and ALT fields, including the case (upper or lower) of each character.

A *renderer,* when cross-rendering a file, may encounter variants that are lacking a DVCF tag. This may happen, for example, when a non-DVCF VCF file is merged into a DVCF file, resulting in variants added that are lacking a DVCF tag. In this case, these variants become single-coordinate variants (in the coordinates of the current rendition), and the *renderer must* set the DVCF tag to `Lrej=AddedVariant` (Primary-only variant) or `Prej=AddedVariant` (Luft-only variant).

A *renderer*, when cross-rendering a file, may encounter variants that have both a PRIM/LUFT field as well as a Prej/Lrej one. This can happen when rendering a DVCF that is a result of merging two DVCF files. The renderer *must* discard one of these fields.

If the *renderer*, when cross-rendering a Luft variant, rejects it - that variant becomes a Luft-only variant, the DVCF tag is set to Prej, and it appears in the Primary rendition as a `##luft_only` meta-information line, similar to the `##primary_only` meta-information line described above.

6.3. INFO and FORMAT fields

Each FORMAT and INFO tag has a RendAlg algorithm associated with it. If the tag has no `##INFO` or `##FORMAT` meta-information line, or the line is lacking a RendAlg attribute, the implementation may decide to apply any of the RendAlgs. For example, it may decide that the field INFO/AF, in case it has no `##INFO` meta-information line, will use the A_1 RendAlg.

Each RendAlg has an *ID*, which appears in the `RendAlg` attribute of the `##INFO` and `##FORMAT` meta-information lines, a *Trigger*, which is a description of the circumstances in which the RendAlg should be applied, and an *Action*, which is a description of the transformation of the data that occurs when the *Trigger* is activated.

The following table lists the standard RendAlgs. An implementation may or may not support any of the standard RendAlgs, and may also add additional RendAlgs. For the *REF change* trigger, it may support all or only certain types of REF changes. However, if a trigger which is supported by the implementation occurs for any particular variant, then each field of the variant that is assigned a standard RendAlg *must* be transformed according to the standard action listed.

ID	**Triggered upon**	**Action**	**Recommended for**
NONE	Never	Do nothing	Fields that don't require change
G	REF change	Re-order / expand the values of a field that has one value per genotype	Fields with Number=G, such as: FORMAT/GL
R	REF change	Re-order / expand the values of a field that has one value per allele	Fields with Number=R, such as: FORMAT/AD
R2	REF change	Re-order / expand the values of a field that has 2 values per allele	Fields with 2 values per allele, such as: FORMAT/SAC
A_1	REF change	Re-calculate / expand the values of a field, so that their sum plus the implied value for REF is 1.	Fields with Number=A, whose values, including the implied value for REF, add up to 1. Examples: FORMAT/AF, INFO/AF
A_*tag*	REF change	Re-calculate / expand the values of a field, so that their sum plus the implied value for REF equals the value in INFO/*tag*.	Fields with Number=A, whose values, including the implied value for REF, add up to the value in INFO/*tag*. Example: INFO/AC would have a RendAlg of A_AN.
GT	REF change	Re-assign / add allele numbers based on the new REF/ALT order. Alleles in the genotype are not reordered.	FORMAT/GT
XREV	XSTRAND=X	Reverse the order of the elements in the array	Fields with a value per base ACGT. Example: INFO/BaseCounts
END	Always	Recalculate the value so that (value - POS) remains unchanged	INFO/END

When cross-rendering, a Renderer *must* cross-render every INFO and FORMAT field according to its *RendAlg*, if the *Trigger* has occurred.

If cross-rendering fails for a particular field, then the variant will have an INFO/Lrej or INFO/Prej field, with *Reason* set to the rejected INFO or FORMAT field name, for example `Lrej=INFO/END`.

Any RendAlg algorithm *must* be losslessly invertible. In other words, applying it to a variant in one rendition, and then applying the inverse algorithm to the resulting other rendition, must result in getting back the original rendition, precisely. For example, the GT RendAlg listed below, upon REF ⇆ALT switch of a bi-allelic, triploid variant, will flip allele numbers in an unphased FORMAT/GT field `0/1/1` to `1/0/0`. It may have been desirable to also sort the result as common in representation of unphased genotypes, so `1/0/0` becomes `0/0/1`. However, that would cause the loss of the information regarding the original order of allele values, and hence the non-existence of a losslessly invertible algorithm, and is therefore prohibited.

While in most cases the RendAlg will only modify the INFO or FORMAT field on which it triggered, it is not restricted in this way: A RendAlg algorithm may change, add or remove other fields of the variant, so long as it is losslessly invertible.

While a *Lifter* transforming a Source VCF to a DVCF in the Primary rendition need not cross-render INFO and FORMAT fields, it is recommended that it nevertheless validates that cross-rendering may be carried out successfully, and sets an INFO/Lrej field if not.

6.4. Sorting

The Primary and Luft renditions are both sorted by their respective coordinates, as required by the VCF specification.

Variants appearing in `##primary_only` and `##lift_only` meta-information lines are not required to be sorted.

7. Rejection and reasons

The *Lifter* or *Renderer* may reject a variant, which in effect declares it to be a single-coordinate variant in the current coordinates.

The *Renderer* may also reject a variant that is already a dual-coordinate variant, turning it into a single-coordinate variant. This may happen, for example, if a new INFO or FORMAT field were added that the *Renderer* cannot cross-render.

When cross-rendering, a *Renderer must* either render the entire variant with all fields cross-rendered as specified, or reject the variant. In other words, if there is a field of a variant which the *Renderer* cannot cross-render for any reason - then the entire variant *must* be rejected, and set the DVCF tag to `Lrej` or `Prej` with the *Reason*. If there are multiple *Reasons* for rejection, the implementation must still list just one *Reason*.

An implementation may use the *Reasons* listed in the table, in which case it *must* use them only when the *Occurrence* in the table occurs. It is recommended that the *Reason* string be 14 characters or shorter, with the exception of INFO/*tag* and FORMAT/*tag* where *tag* is the ID as it appears in the `##INFO` or `##FORMAT` meta-information line.

	Reason	**Occurrence**
`Chain file mapping reasons`	`NoChrom`	Primary CHROM has no alignment in chain file
	`NoMapping`	Primary POS has no alignment in chain file
`REF change reasons`	`RefTooLong`	Implementation cannot handle REF because it is too long
	`RefLongChange`	Implementation cannot handle REF because it is too long, given that it has changed
	`RefLongXstrand`	Implementation cannot handle REF because it is too long, given that XSTRAND=X
	`AltLongXstrand`	Implementation cannot handle ALT because it is too long, given that XSTRAND=X
	`AltLongSwitch`	Implementation cannot switch REF ⇆ ALT because ALT is too long
	`RefChngeNotAlt`	Implementation cannot handle a REF change because the Luft REF is not identical to the Primary ALT
`RendAlg reasons`	`INFO/`*`tag`*	INFO/*tag* cannot be cross-rendered
	`FORMAT/`*`tag`*	FORMAT/*tag* cannot be cross-rendered
`Other reasons`	`AddedVariant`	When cross-rendering, the variant had no DVCF tag
	`Rejected`	Other rejection reason