**Variant annotations in VCF format**

Pablo Cingolani, Fiona Cunningham, Will McLaren (& Kai Wang)

Proposal for unifying VCF functional annotations field names and meanings. This document is intended as a standard way of representing variant annotations within VCF files (INFO fields). It also specifies how to handle some inconsistencies, border cases and how to improve agreement with HGVS notation as well as Sequence Ontology terms and their putative impact. The aim of this standard is to: i) make pipeline development easier, ii) facilitate benchmarking, and iii) improve some problems in edge cases.

**Candidate release date:** December, 2014 or early January, 2015

Color guide:

**Optional** items are highlighted in green

**Preferred** items are highlighted in yellow

**Mandatory** items are not highlighted

**General guidelines**

We use the name ‘effect’ and ‘consequence’ interchangeably, meaning “functional annotation”.

* VCF INFO field name **ANN**, stands for ‘annotations’
* Data fields are encoded separated by pipe sign "|"; the order of fields is written in the VCF header.
* When comparing genomic coordinates, the comparison should be done first by chromosome names (compared alphabetically), then by start position, and finally by end position.
* Special characters: Comma, space, tab, newline or pipe characters (‘,’, ‘ ‘, ‘\t’, ‘\n’, ‘|’, etc.) can be either:
  + Convert to underscore (‘\_’). This is the preferred way.

How about the “p.=” to describe synonymous variants? Since ‘=’ is an illegal character in VCF specification, we can use an alternative notation, such as ‘p.(Leu54Leu)’

HGVS says:

*Description of so called "silent" changes in the format p.(Leu54Leu) (or p.(L54L))* ***should*** *not be used. When desired such changes can be described using p.(=)*

HGVS recommendation discourages the use of the format ‘p.(Leu54Leu)’, but does not forbid it (the spec. says “should not” instead of “must not”).

* + Encoded as %XX (same as URL encoding). This may be needed to express HGVS ‘p.(=)’
* Multiple “effects / consequences” are separated by comma.
  + Optional: Annotations are sorted by sorted by:
    1. Effect/Consequence: Estimated deleteriousness. Compare using ‘most deleterious’ when multiple consequences are predicted.
    2. In case of coding consequence: Best transcript support level (TSL <http://www.ensembl.org/Help/Glossary?id=492>) or Canonical transcript should be first
    3. Feature genomic coordinates.
    4. Feature ID (compared alphabetically, even if the ID is a number).

**Field order and meaning**

* Allele (or ALT):
  + In case of multiple ALT fields, this helps to identify which ALT we are referring to. E.g.:

|  |
| --- |
| #CHROM POS ID REF ALT QUAL FILTER INFO  chr1 123456 . C A . . ANN=A|...  chr1 234567 . A G,T . . ANN=G|... , T|... |

* + In case of cancer sample, when comparing somatic versus germline using a non-standard reference (e.g. one of the ALTs is the reference) the format should be ALT-REFERENCE. E.g.:

|  |
| --- |
| #CHROM POS ID REF ALT QUAL FILTER INFO  chr1 123456 . A C,G . . ANN=G-C|... |

* + Compound variants: two or more variants affecting the annotations (e.g. two consecutive SNPs conforming a MNP, two consecutive frame\_shift variants that “recover” the frame). In this case, the Allele field should include a reference to the other variant/s included in the annotation:

|  |
| --- |
| #CHROM POS ID REF ALT QUAL FILTER INFO  chr1 123456 . A T . . ANN=T|...  chr1 123457 . C G . . ANN=C-chr1:123456\_A>T|... |

* Annotation (a.k.a. effect or consequence): Annotated using Sequence Ontology terms. Multiple effects can be concatenated using ‘&’.

|  |
| --- |
| #CHROM POS ID REF ALT QUAL FILTER INFO  chr1 123456 . C A . . ANN=A|intron\_variant&nc\_transcript\_variant |

* Putative\_impact: A simple estimation of putative impact / deleteriousness : {HIGH, MODERATE, LOW, MODIFIER}
* Gene Name: Common gene name (HGNC). Optional: use closest gene when the variant is “intergenic”.
* Gene ID: Gene ID
* Feature type: Which type of feature is in the next field (e.g. transcript, motif, miRNA, etc.). It is preferred to use Sequence Ontology (SO) terms, but ‘custom’ (user defined) are allowed.

|  |
| --- |
| ANN=A|stop\_gained|HIGH|||**transcript**|... |

Tissue specific features may include cell type / tissue information separated by semicolon e.g.:

|  |
| --- |
| ANN=A|histone\_binding\_site|LOW|||**H3K4me3:HeLa-S3**|... |

* Feature ID: Depending on the annotation, this may be: Transcript ID (preferably using version number), Motif ID, miRNA, ChipSeq peak, Histone mark, etc.

**Note:** Some features may not have ID (e.g. histone marks from custom Chip-Seq experiments may not have a unique ID).

* Transcript biotype. The bare minimum is at least a description on whether the transcript is {“Coding”, “Noncoding”}. Whenever possible, use ENSEMBL biotypes.
* Rank / total : Exon or Intron rank / total number of exons or introns.
* HGVS.c
* HGVS.p: If variant is coding. Since transcript ID is already mentioned in ‘feature ID’, it may be omitted here.
* cDNA\_position / (cDNA\_len optional) : Position in cDNA and trancript’s cDNA length (one based).
* CDS\_position / (CDS\_len optional): Position and number of coding bases (one based includes START and STOP codons).
* Protein\_position / (Protein\_len optional): Position and number of AA (one based, including START, but not STOP).
* Distance to feature: All items in this field are options, so the field could be empty.
  + Up/Downstream: Distance to first / last codon
  + Intergenic: Distance to closest gene
  + Distance to closest Intron boundary in exon (+/- up/downstream). If same, use positive number.
  + Distance to closest exon boundary in Intron (+/- up/downstream)
  + Distance to first base in MOTIF
  + Distance to first base in miRNA
  + Distance to exon-intron boundary in splice\_site or splice \_region
  + ChipSeq peak: Distance to summit (or peak center)
  + Histone mark / Histone state: Distance to summit (or peak center)
* Errors, Warnings or Information messages. Add errors, warnings or informative message that can affect annotation accuracy. It can be added using either ‘codes’ (as shown in column 1, e.g. W1) or ‘message types’ (as shown in column 2, e.g. WARNING\_REF\_DOES\_NOT\_MATCH\_GENOME). All these errors, warnings or information messages messages are optional.

|  |  |  |
| --- | --- | --- |
| **Code** | **Message type** | **Description / Notes** |
| E1 | ERROR\_CHROMOSOME\_NOT\_FOUND | Chromosome does not exists in reference genome database. Typically indicates a mismatch between the chromosome names in the input file and the chromosome names used in the reference genome. |
| E2 | ERROR\_OUT\_OF\_CHROMOSOME\_RANGE | The variant’s genomic coordinate is greater than chromosome's length. |
| W1 | WARNING\_REF\_DOES\_NOT\_MATCH\_GENOME | This means that the ‘REF’ field in the input VCF file does not match the reference genome. This warning may indicate a conflict between input data and data from reference genome (for instance is the input VCF was aligned to a different reference genome). |
| W2 | WARNING\_SEQUENCE\_NOT\_AVAILABLE | Reference sequence is not available, thus no inference could be performed. |
| W3 | WARNING\_TRANSCRIPT\_INCOMPLETE | A protein coding transcript having a non-multiple of 3 length. It indicates that the reference genome has missing information about this particular transcript. |
| W4 | WARNING\_TRANSCRIPT\_MULTIPLE\_STOP\_CODONS | A protein coding transcript has two or more STOP codons in the middle of the coding sequence (CDS). This should not happen and it usually means the reference genome may have an error in this transcript. |
| W5 | WARNING\_TRANSCRIPT\_NO\_START\_CODON | A protein coding transcript does not have a proper START codon. It is rare that a real transcript does not have a START codon, so this probably indicates an error or missing information in the reference genome. |
| I1 | INFO\_REALIGN\_3\_PRIME | Variant has been realigned to the most 3-prime position within the transcript. This is usually done to to comply with HGVS specification to always report the most 3-prime annotation. |
| I2 | INFO\_COMPOUND\_ANNOTATION | This effect is a result of combining more than one variants (e.g. two consecutive SNPs that conform an MNP, or two consecutive frame\_shift variants that compensate frame). |
| I3 | INFO\_NON\_REFERENCE\_ANNOTATION | An alternative reference sequence was used to calculate this annotation (e.g. cancer sample comparing somatic vs. germline). |

**Consistency between HGVS and functional annotations**

In some cases there might be inconsistent reporting between ‘annotation’ and HGVS. This is due to the fact that VCF recommends aligning to the leftmost coordinate, whereas HGSV recommends aligning to the “most 3-prime coordinate”.

For instance, an InDel on the edge of an exon, which has an ‘intronic’ annotation according to VCF alignment recommendation, can lead to a ‘stop\_gained’ when aligned using HGVS’s recommendation (using the most 3-prime possible alignment). So the ‘annotation’ sub-field will report ‘intron’ whereas HGVS sub-field will report a ‘stop\_gained’. This is obviously inconsistent and must be avoided.

In order to report annotations that are consistent with HGVS notation, variants must be re-aligned according to each transcript’s strand (i.e. align the variant according to the transcript’s most 3-prime coordinate). Then annotations are calculated, thus the reported annotations will be consistent with HGVS notation. Annotation software should have a command line option to override this behaviour (e.g. ‘-no\_shift\_hgvs’)

(here there are two specific issues: (1) variant calling algorithms usually do not try to do left normalization, so the users need to use third-party tools such as vt/bcftools/gatk to re-normalize variant calls when needed (2) but if the variant is located in reverse strand, then users need to right-align and re-normalize them. I feel that the choice should be made by the users, rather than the annotation software; it is a data pre-processing step that the users need to make on their VCF file before the annotations are generated. Annotations itself would only work on whatever users input faithfully, rather than trying to help users modify a VCF file in unexpected manner.

**Annotations and putative impacts**

The following table describes the suggested putative impact for some Sequence Ontology terms often used in functional annotations.

|  |  |
| --- | --- |
| **Putative Impact** | **Sequence Ontology term** |
| HIGH | chromosome\_number\_variation |
| HIGH | exon\_loss\_variant |
| HIGH | frameshift\_variant |
| HIGH | rare\_amino\_acid\_variant |
| HIGH | splice\_acceptor\_variant |
| HIGH | splice\_donor\_variant |
| HIGH | start\_lost |
| HIGH | stop\_gained |
| HIGH | stop\_lost |
| HIGH | transcript\_ablation |
| MODERATE | 3\_prime\_UTR\_truncation +exon\_loss |
| MODERATE | 5\_prime\_UTR\_truncation +exon\_loss\_variant |
| MODERATE | coding\_sequence\_variant |
| MODERATE | disruptive\_inframe\_deletion |
| MODERATE | disruptive\_inframe\_insertion |
| MODERATE | inframe\_deletion |
| MODERATE | inframe\_insertion |
| MODERATE | missense\_variant |
| MODERATE | regulatory\_region\_ablation |
| MODERATE | splice\_region\_variant |
| MODERATE | TFBS\_ablation |
| LOW | 5\_prime\_UTR\_premature start\_codon\_gain\_variant |
| LOW | initiator\_codon\_variant |
| LOW | splice\_region\_variant |
| LOW | splice\_region\_variant |
| LOW | start\_retained |
| LOW | stop\_retained\_variant |
| LOW | stop\_retained\_variant |
| LOW | synonymous\_variant |
| MODIFIER | 3\_prime\_UTR\_variant |
| MODIFIER | 5\_prime\_UTR\_variant |
| MODIFIER | coding\_sequence\_variant |
| MODIFIER | conserved\_intergenic\_variant |
| MODIFIER | conserved\_intron\_variant |
| MODIFIER | downstream\_gene\_variant |
| MODIFIER | exon\_variant |
| MODIFIER | feature\_elongation |
| MODIFIER | feature\_truncation |
| MODIFIER | gene\_variant |
| MODIFIER | intergenic\_region |
| MODIFIER | intragenic\_variant |
| MODIFIER | intron\_variant |
| MODIFIER | mature\_miRNA\_variant |
| MODIFIER | miRNA |
| MODIFIER | NMD\_transcript\_variant |
| MODIFIER | non\_coding\_transcript\_exon\_variant |
| MODIFIER | non\_coding\_transcript\_variant |
| MODIFIER | regulatory\_region\_amplification |
| MODIFIER | regulatory\_region\_variant |
| MODIFIER | TF\_binding\_site\_variant |
| MODIFIER | TFBS\_amplification |
| MODIFIER | transcript\_amplification |
| MODIFIER | transcript\_variant |
| MODIFIER | upstream\_gene\_variant |

**Annotations sort order**

When comparing two annotations, the “most deleterious” one is shown first. It is recommended annotation programs clearly state their respective “deleteriousness” order. This is an example of such putative sorting order:

1. chromosome\_number\_variation
2. exon\_loss\_variant
3. frameshift\_variant
4. stop\_gained
5. stop\_lost
6. start\_lost
7. splice\_acceptor\_variant
8. splice\_donor\_variant
9. rare\_amino\_acid\_variant
10. missense\_variant
11. inframe\_insertion
12. disruptive\_inframe\_insertion
13. inframe\_deletion
14. disruptive\_inframe\_deletion
15. 5\_prime\_UTR\_truncation+exon\_loss\_variant
16. 3\_prime\_UTR\_truncation+exon\_loss
17. splice\_branch\_variant
18. splice\_region\_variant
19. splice\_branch\_variant
20. stop\_retained\_variant
21. initiator\_codon\_variant
22. synonymous\_variant
23. initiator\_codon\_variant+non\_canonical\_start\_codon
24. stop\_retained\_variant
25. coding\_sequence\_variant
26. 5\_prime\_UTR\_variant
27. 3\_prime\_UTR\_variant
28. 5\_prime\_UTR\_premature\_start\_codon\_gain\_variant
29. upstream\_gene\_variant
30. downstream\_gene\_variant
31. TF\_binding\_site\_variant
32. regulatory\_region\_variant
33. miRNA
34. custom
35. sequence\_feature
36. conserved\_intron\_variant
37. intron\_variant
38. intragenic\_variant
39. conserved\_intergenic\_variant
40. intergenic\_region
41. coding\_sequence\_variant
42. non\_coding\_exon\_variant
43. nc\_transcript\_variant
44. gene\_variant
45. chromosome