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_COD	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

	ONS	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