Compound / Haplotype Annotations

Pablo Cingolani

Introduction

#CHOMR	POS	ID	REF	ALT
1	123	rs1	C	T
1	125	rs2	A	C

```
REF      : ---CTA---    AA: L
id1      : ---T-----    AA: L
id2      : -----C---    AA: L
id1+id2  : ---T-C---    AA: P
```

Consequences

From: Peter Robinson

…

The patient has a variant on the same haplotype that is of the form

----A-C-----

(where "-" indicates wildtype and A and C are variant bases). GATK considers this to be two different variants and reports them on two lines. This obviously percolates downstream...but the **medical interpretation** is clearly dependent on knowing that they are on the same haplotype...

Example

```
#CHROM   POS   ID   REF   ALT   ...   S1    S2    S3    S4
1        123   .    C     T           1/1   0/1   0|1   0|1
1        124   .    T     A           1/1   1/0   0|1   1|0
```

Sample	Phased	Compound variant
S1	Implicit: Both Hom_Alt	Yes
S2	No	Unknown
S3	Yes	Yes
S4	Yes	No

Haplotype annotations Improvements

- **Same codon:** We should definitely consider them for annotations as a haplotype, even if there is no phase information (e.g. implicit phase).

- **Adjacent / a few bases apart:** Should consider them as a haplotype annotation? (e.g. less than 10 bases?). What types of annotations can we improve usig phasing information?

- **Within same exon:** Many bases apart (e.g. 50 bases apart), we can annotate frame recovering variants and whether they had stop or not any codons between them.

Haplotype annotations
Little / No improvement?

- **Different exon (same transcript):** Is there any case we can accurately annotate (and there is biological evidence that the annotations make sense?)

- **Different transcript / gene:** We don’t have reliable algorithms to infer impact (e. g. long distance regulatory variante).

- **Different chromosomes:** E.g. variants in chr1 and chr2 (both inherited from the maternal side). We don’t know if there are any biological consequences at all.

Input data
Allelic primitives

```
#CHOMR  POS    ID    REF   ALT
1       123    .     C     T
1       125    .     A     C
```

- All variant callers do at least this.
- Variant annotation algorithms need additional information, such as read based phasing, otherwise we cannot infer haplotypes: i) we don't have read information at this stage, and ii) even if we had trios we should not phase
- Both variants are "synonymous_variant" individually, but "stop_gained" when compound effect. Where do we annotate? First VCF line? Second VCF line? Both lines?
- Parallelization issues: Annotation algorithms are harder to parallelize if we have to take into account multiple variants / VCF lines (e.g. throw one VCF line to each CPU is no longer possible). Phasing group can delimit variants groups that can be assigned to different CPUs (but this implies parsing GT)
- Complex cases, such as ALT/ALT phased-variants (cancer germline vs somatic comparison).

Input data
Haplotype

```
#CHOMR  POS    ID    REF   ALT
1       123    .     CTA   TTA
```

- Lazy: This seems easier for variant annotation programs (at least for simple variants)
- May get really complicated when too many samples have many different alternative haplotypes. Remember that VCf files with 100K samples can be common in a couple of years.
- Requiring this from the variant caller might be unrealistic. What happens if a variant caller does not comply? We can always say *"you need to use another variant caller to get haplotype annotations"* (pedantic?)

Conclusions

- **Improvements:** We should have a clear list of cases where we can actually make improvements in annotations when using haplotypes. Some of these are obvious (e.g. adjacent phased SNPs), but in many cases we cannot handle them.

- **Input variants:**
  - Simple allelic primitives: All variant callers do at least this. Simplifies “upstream” requirements. Requires phasing information (otherwise only Hom-ALT variants can be annotated).

  - Haplotype variants: Some variant callers provide them and we should be able to use the information.

- **Development:** Can we get a set of test cases / benchmarks? This would speed up the process of pushing these changes for implementation in annotation programs.