ELIXIR hCNV 2019-21 Deliverable D3.2

Project Title:	First hCNV Community Implementation Study
Deliverable title:	Create a consensus collection of perceived requirements for efficient and effective CNV file and data exchange formats
WP No.	3
WP Title	Improvement of community formats for CNV exchange
Contractual delivery date:	30.11.2019
Actual delivery date:	12.12.2019
WP leads:	Thomas Keane
Partner(s) contributing to this deliverable:	EMBL-EBI

Report authors: Kirill Tsukanov[1], Sundararaman Venkataraman, Giselle Kerry, Thomas Keane (EMBL-EBI)

[1] Corresponding author, ktsukanov@ebi.ac.uk

Contents

1. Summary

Single nucleotide polymorphisms and short indel variants are usually stored in the Variant Call Format (VCF), which is widely adopted. In contrast, representation of copy number variants (CNVs) and structural variants (SVs) is standardised to a much lesser extent. We gathered input from the community to understand the formats people use to store CNVs, the issues they face in the process, and their perceived requirements for the ideal file format to do this.

The main source of community input was from a survey[2] communicated to the relevant mailing lists and other correspondents, including the primary hCNV mailing list. We also conducted additional discussions on this topic with other work packages of this project and within WP3.

A part of the survey was focused specifically on VCF. Even though this format has support for storing CNVs, it is often perceived as complicated or poorly documented, and we would like to change this. However, we evaluated all other formats mentioned in responses to the survey as well.

Survey results confirmed that, aside from VCF and a number of BED-like tab-separated formats, there are no existing widely used formats for representing structural variation. A list of requirements for the future CNV formats have been prepared based on the survey results, as well as the list of required improvements to VCF specifically.

VCF is a GA4GH standard occupying a key place at the intersection of human and computer readable formats. It is widely supported and implemented in a variety of tools. Taking this into account, as well as the opinion of survey respondents, it seems sensible to invest effort into the improvement of VCF specification to address the problems its users are faced with, and to promote it as a global standard for storing CNV data for the use cases where it is appropriate.

2. Results

The feedback we received came from multiple sources: the survey, work package discussions, and internal discussions. CNV representation is a complex topic which is reflected by the diverse survey responses we received. Below we divide the points raised by respondents into several sections.

2.1. Feedback overview

The survey generated 21 responses. Respondents' self-reported proficiency in working with CNV ranges from modest to several decades of experience. Respondees included those

[2] https://docs.google.com/forms/d/e/1FAIpQLSfQf1-lg6KjOO9WsvFMTIGmdWOXm9ntDvZ0D7Lkgi062woDGw/viewform

working with human and non-human data; rare diseases, cancer, pediatric disease, population genomics; academic researchers and developers of genomic browsers and commercial software. In short, the responses are sufficiently diverse to consider this a representative sample of the community.

2.2. Terminology

For the purposes of this report, we will go with the following definition (which is by no means complete):

> CNVs are *interpreted apparent consequences* of *certain types* of SVs, namely tandem segment duplications and deletions, which lead to the observed increase or decrease of a "copy number" of a certain genome segment compared to the reference genome, with or without associated phasing information.[3,4]

We arrived at this definition after observing the lack of consistent and self-evident definition for CNVs; an issue which several respondents mentioned explicitly. Some people treat this term as a synonym for any structural variant (SV); others consider CNVs to be a subset of SVs, but how exactly this subset is defined is also not something which is uniformly accepted.

This report will use the term "CNV" throughout its text for the purposes of brevity, and also because it has been confirmed in discussion with the other work packages that this project's focus is on CNVs specifically. However, many considerations which apply to CNVs also apply to other SV types. When a certain point in this report refers to SVs in general, it will be pointed out explicitly.

We recommend that the specifications of file formats dealing with CNVs and SVs include careful and precise definitions for these terms, as well as for any of their subtypes and modifications.

2.3. Existing file formats

In this section, we list the formats which survey respondents have used, or have seen being used, to store CNV information in real life projects. The formats are presented in decreasing order of popularity, accompanied by an analysis of their use cases.

2.3.1. VCF (Variant Call Format)

VCF[5] was the single most widely used format in regards to storing CNVs, with 86% of respondents mentioning it. In light of its special importance, a separate part of the survey was dedicated to VCF. Please see Section 2.4 below for detailed analysis.

[3] https://www.cancer.gov/publications/dictionaries/genetics-dictionary/def/cnv
[4] https://www.genome.gov/genetics-glossary/Copy-Number-Variation
[5] https://samtools.github.io/hts-specs/VCFv4.3.pdf

2.3.2. BED and related tab-separated formats

BED-like tab-separated file formats were the second most popular, mentioned by 71%. It should be noted that there is a great number of tab-separated formats for representing CNVs, which are usually specific to the application and software used; many respondents explicitly mentioned this.

The closest thing there is to a *standard* for such formats is BEDPE[6]. To quote from bedtools documentation, where it is defined:

> *We have defined a new file format (BEDPE) in order to concisely describe disjoint genome features, such as structural variations or paired-end sequence alignments. We chose to define a new format because the existing "blocked" BED format (a.k.a. BED12) does not allow inter-chromosomal feature definitions.*

The BEDPE format only includes standard pre-defined fields for chromosome, start, end, strand, name and score of a feature. All additional columns are treated as user-defined fields. As such, there is no standard way to specify CNV type, zygosity, phasing, sample genotypes, or any other crucial information. It should also be noted that even though many people mentioned tab-separated formats in general, there was only one mention of BEDPE specifically.

Additionally, there was a singular mention of bigWig[7]. It is an indexed binary format, used for dense continuous data to be displayed in the UCSC genome browser. It is similar to BEDPE in that it defines format for representing regions and user-defined features, but no specific support for CNV information.

2.3.3. JSON with a schema

This format was mentioned by 10% of respondents. JSON[8] is a widely accepted standard for data exchange between computerised systems. Example of an implementation is a BeaconVariant schema[9] which is an extension of GA4GH Beacon, with proposed changes to support CNVs. However, it is still in an early development stage and does not support most of the CNV-specific attributes.

2.3.4. Other formats

There were two mentions of BAM[10] in the survey responses. As this format stores alignments rather than called variants, it is not relevant to this study.

There was a single mention of GFA[11] (Graphical Fragment Assembly) and VG[12] (Variation Graph) formats. While it is true that they provide a way to represent *complete* information

---

[6] https://bedtools.readthedocs.io/en/latest/content/general-usage.html#bedpe-format
[7] https://genome.ucsc.edu/goldenpath/help/bigWig.html
[8] https://www.json.org/
[9] https://schemablocks.org/schemas/beacon/BeaconGenomicVariant.html
[10] https://samtools.github.io/hts-specs/SAMv1.pdf
[11] http://gfa-spec.github.io/GFA-spec/GFA1.html
[12] https://github.com/vgteam/vg/wiki/File-Formats

about variants of arbitrarily high complexity, such formats are not readily usable in downstream analyses which require more simplified information.

2.4. Opinion on CNV representation in VCF

Given the prominence of VCF among other variation file formats, a separate survey section was dedicated to it. One particular hypothesis we wanted to test with our survey is that people generally support VCF as a global standard for representing variation, but are sometimes confused with its specification, and especially with the parts of it which describe CNV representation.

All survey respondents were aware that VCF supports CNV representation: **67%** have answered that they knew the CNV part of the specification “in reasonable detail”, and the other **33%** that they knew it “in general”.

Interestingly, **52%** of respondents have noted that at least some parts of the VCF specification were difficult to understand. Specifically, **48%** of all respondents said this about the CNV part of the specification, and only **14%** about the other (general) parts.

33% of respondents noted that they had trouble with representing their CNVs even if they followed the specification. Specifically, **24%** said that they had trouble representing their CNVs in the VCF based on the specification; and **19%** that the specification allowed for multiple ways to represent their CNVs and it was not clear which way to choose.

We knew beforehand that CNV support in VCF sparks a lot of debate; however, what we also wanted to gauge is the collective opinion of the community of whether they think VCF should be fixed, or consider it fundamentally unsuitable even with the updated specification. The answers to the question “Do you support the use of VCF as a global standard to represent CNV information?” were:

- Yes, as it is (33%)
- Yes, after proper specification updates (48%)
- No (19%)

In summary, **81%** of respondents support using VCF as a global standard to represent CNV information, after the specification is properly updated. However, given that in its current state only **33%** of respondents support it, the updates to the VCF specification would need to be major to make it significantly more CNV-friendly and to convince the majority of the community, and the success of this cannot be guaranteed in advance.

2.5. Requirements for CNV formats of the future

Respondents of the survey were asked about which requirements they consider important for future CNV file formats. It should be noted that some of the community requirements will inevitably contradict the others. The prioritisation and conflict resolution will happen at the later stages of the project.

The summary of requirements, unified and split by category, is as follows:

1. **Easy and unambiguous interpretation**
    - Human *and* computer readability, or a reasonable combination with a priority for the latter
    - Clear variant representation
        - Structural variant type, selected from a predefined controlled vocabulary with rigorous definitions, periodically updated to include new cases as they arise
        - Location
        - Copy number and genotype
    - Built-in representation of uncertainties associated with the variant
        - Start and end breakpoint position uncertainties
        - Copy number uncertainties
    - Standardised representation of variant annotations should be preferred to user-defined annotations when possible
2. **Link between CNVs and SVs**
    - Intuitive mechanism to link CNVs and SVs
    - For CNVs, a row (or other entity type in a file) should represent an actual *variant,* not a breakpoint
    - Tool support to efficiently navigate to information at linked breakpoints
3. **Support for complex cases**
    - Storing joint callset of multiple samples; scalability to a large number of samples
    - Ability to specify base ploidy, e.g. for X chromosome and for non-diploid organisms
    - Mosaic CNVs, aneuploidy and sub-clonality support
    - Support the ability to represent nested or overlapping events
4. **Alignment with global standards and other file formats**
    - Alignment with GA4GH Variant Representation and Variant Annotation
    - Compatibility with file formats that are used to represent non-CNV variants
5. **Provenance of the results**
    - Information on which callers produced the variant, and which settings were used
    - Inclusion of original experimental values (intensity/$\log_2$/count/cytobands)
    - Ability to go from a variant to the original reads/observations which produced it (in case of NGS: lossless representation of whole genome alignments)

2.6. Conclusions. Note about use cases

It should be expected that no single file format can accommodate all of the requirements listed in the previous section; and no single file format could fulfill all use cases. A BED-like format with user-defined fields would perhaps be the most human readable format, but not parseable. A JSON following a specific and complicated schema would be the most convenient for automated exchange via APIs, but more complicated for users to directly work with and perform ad-hoc QC and analyses (especially for researchers with little or no computer science background). Graph representations will be able to store variant

information of unbounded complexity, but are not directly suitable for downstream analysis and would still require other file formats to convert into.

In light of this, one of the key advantages of VCF is that it remains at the intersection of human and computer readable formats. VCF is widely known and support for it is already implemented in a variety of tools. Considering that the majority of survey respondents supported the use of VCF for storing CNV data, it seems sensible to invest effort into the improvement of its specification to address the problems its users are faced with.

At the same time, other variation formats should continue to be developed. Variation graphs and elaborate JSON schemas may provide a way to encode very complicated events with high precision, while simple JSON schemas may be used to exchange high-level variant information between automated services (such as Beacon).

It is essential for all kinds of CNV file formats to maintain a focus on compatibility between each other, rather than competing and trying to fill all use cases with a single format.

3. Impact

This report is based on the survey results, discussions with other work packages, and internal discussions. Its findings on file formats for CNV representation will help guide further discussion on this topic and help plan modifications and improvements to the existing data formats for CNV representation and exchange.

The next concrete steps would be:

1. For other work packages to confirm which CNV calling and manipulation tools support VCF already, or for which implementation of VCF support is planned or underway.
2. Plan, coordinate and implement changes to the VCF specification in alignment with GA4GH File Formats and Future of VCF groups.
3. Coordinate and implement validators for CNV representation in VCF.
4. Coordinate and implement converters between VCF and other possible formats for representing CNV.

4. Adjustments Made

Report delivered 12.12.2019 instead of 30.11.2019.