

**ISO/IEC JTC 1/SC 29/WG 11 N18361**

**ISO/IEC JTC 1/SC 29/WG 11**

**Coding of moving pictures and audio**

**Convenorship: UNI (Italy)**

**Document type: Approved WG 11 document**

**Title: Draft requirements for MPEG-G Part 6 Annotations**

**Status: Approved**

**Date of document: 2019-04-08**

**Source: Requirements**

**Expected action:**

**No. of pages: 14**

**Email of convenor: leonardo@chiariglione.org**

**Committee URL: mpeg.chiariglione.org**

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC 1/SC 29/WG 11 N18361**

**Geneva, Switzerland – March 2019**

|  |  |
| --- | --- |
| **Source:** | **Requirements** |
| **Title:** | **Draft requirements for MPEG-G Part 6 : Coding of Genomic Annotations** |

**Draft requirements for MPEG-G Part 6: Coding of Genomic Annotations**

**Table of Contents**

[1 Executive summary 4](#_Toc6311293)

[2 Terminology 4](#_Toc6311294)

[3 Data types related to genomic annotations 5](#_Toc6311295)

[3.1 ISO/IEC 23092 (MPEG-G) hierarchy 5](#_Toc6311296)

[3.2 Annotations 5](#_Toc6311297)

[3.3 Relevant data types 7](#_Toc6311298)

[4 Requirements 9](#_Toc6311299)

[4.1 Mapping statistics 9](#_Toc6311300)

[4.2 Quantitative browser tracks 9](#_Toc6311301)

[4.3 Variants 10](#_Toc6311302)

[4.4 Genome functional annotation 11](#_Toc6311303)

[4.5 Expression values 12](#_Toc6311304)

[4.6 Hi-C-like experiments 13](#_Toc6311305)

[4.7 Transport 14](#_Toc6311306)

# Executive summary

This document lists the requirements that responses to the Call for Proposals for ISO/IEC 23092 Part 6, Coding of Genomic Annotation, shall meet.

# Terminology

Definitions for most concepts appearing in the following text, and concerning the production and analysis of sequencing data, can be found in the Terms and Definitions sections of ISO/IEC 23092 (MPEG-G) Parts 1, 2 and 3. As for annotation-related information we address in this call, the table below provides additional definitions. Descriptions of the file formats mentioned in this call can be found in the following table.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| BED | BED (Browser Extensible Data) format provides a flexible way to define the data lines that are displayed in an annotation track. BED lines have three required fields and nine additional optional fields. The number of fields per line must be consistent throughout any single set of data in an annotation track. The order of the optional fields is binding: lower-numbered fields must always be populated if higher-numbered fields are used.  <http://genome.ucsc.edu/FAQ/FAQformat#format1> |
| GTF | GTF (Gene Transfer Format, GTF2.2) is an extension to, and backward compatible with, GFF2. The first eight GTF fields are the same as GFF. The feature field is the same as GFF, with the exception that it also includes the following optional values: 5UTR, 3UTR, inter, inter\_CNS, and intron\_CNS.  <http://mblab.wustl.edu/GTF22.html> |
| GFF3 | GFF is a standard file format for storing genomic features in a text file. GFF stands for Generic Feature Format. GFF files are plain text, 9 column, tab-delimited files.  <http://gmod.org/wiki/GFF3> |
| bigWig | The bigWig format is useful for dense, continuous data that will be displayed in the Genome Browser as a graph.  <http://genome.ucsc.edu/goldenPath/help/bigWig.html> |
| wig | The wiggle (WIG) format is an older format for display of dense, continuous data such as GC percent, probability scores, and transcriptome data. Wiggle data elements must be equally sized.  <http://genome.ucsc.edu/goldenPath/help/wiggle.html> |
| GenBank | GenBank format (GenBank Flat File Format) consists of an annotation section and a sequence section. The start of the annotation section is marked by a line beginning with the word "LOCUS". The start of sequence section is marked by a line beginning with the word "ORIGIN" and the end of the section is marked by a line with only "//".  <https://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html> |
| bedGraph | The bedGraph format allows display of continuous-valued data in track format. This display type is useful for probability scores and transcriptome data. This track type is similar to the [WIG](http://genome.ucsc.edu/FAQ/FAQformat#format6) format, but unlike the WIG format, data exported in the bedGraph format are preserved in their original state.  <http://genome.ucsc.edu/goldenPath/help/bedgraph.html> |
| loom | Loom is a file format for large omics datasets based on the HDF5 format. A loom file always contains a main matrix plus optional additional layers, a variable number of row and column annotations, and sparse graph objects. |
| Hic files | Hi-C experiment data are provided in the form of .hic files. .hic files can be manipulated with the Juicer Tools and Juicebox software (<https://github.com/aidenlab/juicer>).  Hi-C experiments are used to study the three-dimensional structure of DNA, identifying genome-wide long-range interactions. Cells are fixed with formaldehyde, chemically bounding close genomic: the links created persist during sequencing and can be detected and recorded. The output of the experiment is a genome wide contact matrix, stored in “.hic” files. |

Table 1 – Major annotation formats

# Data types related to genomic annotations

## ISO/IEC 23092 (MPEG-G) hierarchy

Sequencing reads of different lengths (which depend on the technology used to generate them) can be localized at one or more points on the DNA molecule they originate from. They are the basic tokens of information at the foundation of all high-level biological experiments based on sequencing. It is hence only natural for the ISO/IEC 23092 (MPEG-G) hierarchy to be based on reads, which get organized in terms of, from bottom to top, Access Units, Datasets, and Dataset Groups.

## Annotations

The output of most biological studies based on sequencing protocols is usually represented as different types of *annotations* (meta-information) all associated to one or more *intervals* on the reference genome[[1]](#footnote-1).

An interval is typically identified by the name of the *sequence* in the reference, the molecule *strand* (can be forward or reverse), and a lower (5’) and a higher (3’) *positions* specifying the base range.

|  |  |  |  |
| --- | --- | --- | --- |
| **Interval** | | | |
| sequence | strand | lower position | higher position |

Intervals are the natural way to talk about features localized on the genome, be them the number of aligning reads (or read coverage), variants, genes, regions of the genome binding to proteins, regions that perform a specific function in the architecture of the genome, and so on. They can be as short as one single base, but they usually span much larger scales, allowing people to associate meta-information to its correct scale.

As a result the so-called *primary* data analysis (i.e., the analysis of the data derived from sequencing), which is performed at the level of the read, is usually followed by some *secondary* analysis, which is performed at the level of the genomic interval. Making a parallel with video coding, and thinking of movement along the genome as movement in time, one could see the reads as video information, while meta-information based on intervals would be similar to subtitles.

For instance, typical steps of a genomic variant-calling analysis workflow are as follows:

|  |  |  |
| --- | --- | --- |
|  | **Step** | **Description** |
| 1 | Sequence reads extraction | The process of extraction of fragments of DNA/RNA in the form of sequences of nucleotides from a biological sample. Sequences of nucleotides are commonly referred to as “reads”. |
| 2 | Mapping and Alignment | Sequence alignment refers to the process of arranging sequence reads by finding regions of similarity that may be a consequence of functional, structural, or evolutionary relationships among the sequences. When the alignment is performed with reference to an existing DNA sequence the process is called “mapping”. |
| 3 | Variant detection | Variant detection (a.k.a. variant calling) is the process of translating the output of DNA sequencing machines, (reads mentioned in step 1 and aligned in step 2), to a summary of the unique characteristics of the organism being sequenced. These characteristics are called “variants” because they are expressed as differences between the sequenced genome and a reference genome. |
| 4 | Variant annotation | Variant annotation is the process of assigning functional information to the DNA variants identified in step 3. This implies the classification of variants according to their relationship to coding sequences in the genome and according to their impact on the coding sequence and the gene product. |
| 5 | Functional & Structural Analysis | Analysis of DNA (variants, CNV = copy number variation, methylation etc,) strands to define their relationship with genes (and proteins) functions and structure. |

Table 2 - The main stages of a typical genomic variant calling pipeline

While steps 1-2 are fully supported by the previous parts 1-5 of the MPEG-G standard, this call concerns itself with the representation of the results of the subsequent steps. Figure 1 depicts a functional diagram of the typical genomic information life cycle expressed as different processing steps and intermediate file formats.

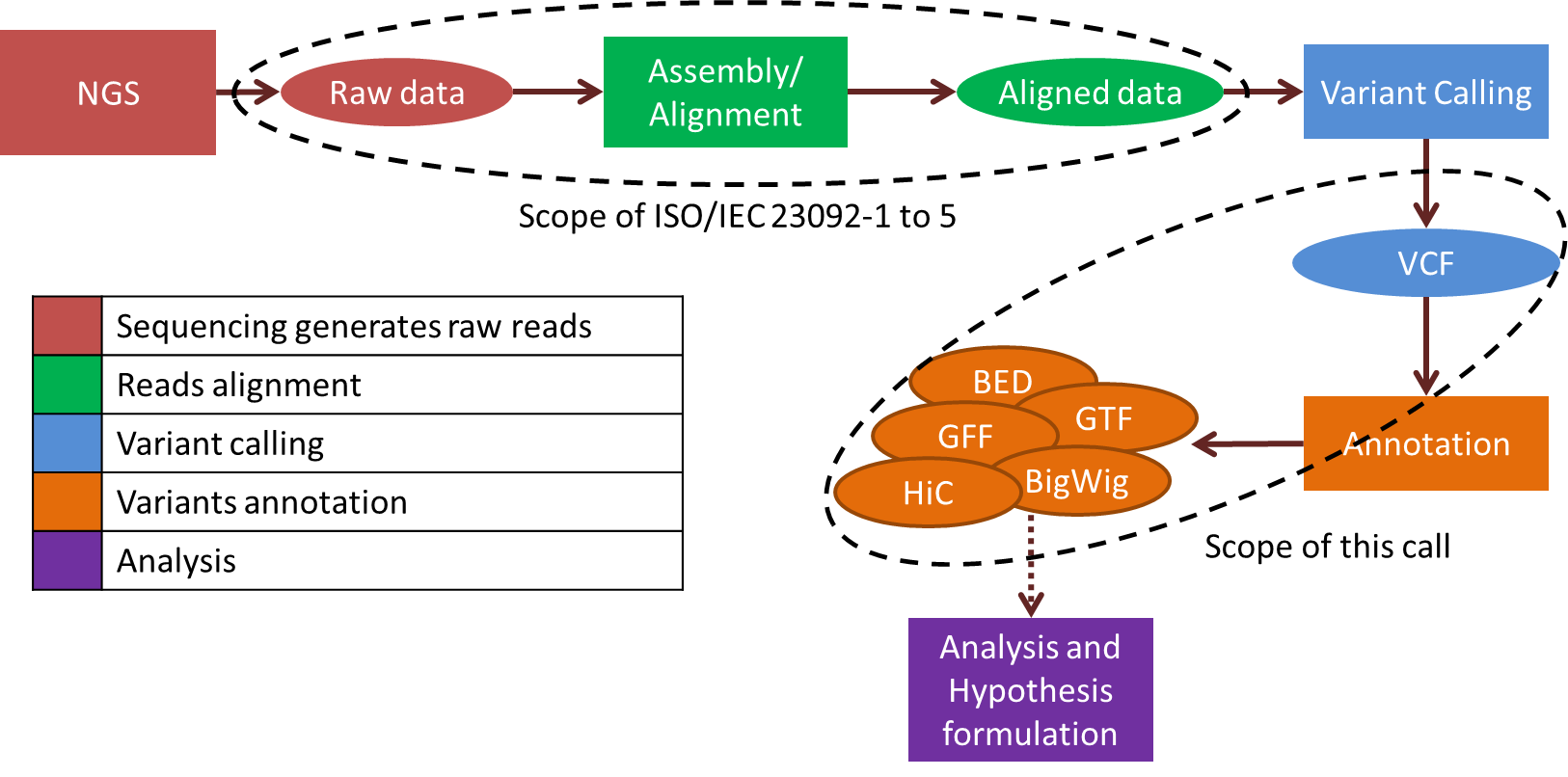


Figure 1 - Typical structure of a genomic variant-calling pipeline from sequencing to analysis.

## Relevant data types

In summary, here we propose to augment the ISO/IEC 23092 (MPEG-G) hierarchy with the concept of meta-information related to intervals, and to support a number of additional use cases relevant to secondary data analysis. At the moment such use cases typically result in a number of file formats. They are all based on the concept of interval, and their use is widespread in the community. A list of data types relevant to this call can be found in the following table:

Table 3 – Data types relevant to this call

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Data type** | **Description** | **Indicative file format(s)** |
| 1.1 | Mapping statistics | Some mapping statistics could be pre-computed with some granularity, for instance at the level of thousands of nucleotides or larger. In order to support queries by interval a typical granularity for the binning of precomputed intervals might be at the level of thousands of nucleotides or larger. In order to avoid storing too large amounts of information in the file, one would typically precompute statistics on larger scales, and then give the complete answer by re-computing statistics on the fly the for the small sub-intervals at the sides of the query which are not covered by the pre-existing binning | There is no indicative file format, but for a description of relevant statistics one can refer to Part 3 of ISO/IEC 23092 (MPEG-G) |
| 1.2 | Quantitative browser tracks | Mappings between nucleotides on the genome and numerical values. Can be done at different scales (with one value corresponding to each nucleotide, or one value corresponding to more than one nucleotide), with formats such as bigWig containing snapshots at different scales for ease of visualization. They can be used to visualize read coverage (and hence ChIP-seq or methylation/epigenetics experiments, variant frequency, etc.). | wig, bigWig, bedGraph |
| 1.3 | Variants | Genomic variants localized on a reference (genomic or transcriptomic). Variants can be simple (SNPs or short indels) or complex (large-scale rearrangements). There could be more than one variant at the same position on the reference, and more than one sample/individual might be considered (population-level statistics). | VCF |
| 1.4 | Genome functional annotation | Localization of gene models on the genome as lists of UTRs, exons and coding intervals. Other biological features such as repeats, ontology annotations, gene names. In the case of ChIP-seq experiments, features could be a list of called peaks. | BED, GTF, GFF(3), GenBank |
| 1.5 | Expression values | They describe the number of reads, or related metrics (FPKM, etc.) associated to a list of genes or transcripts. Data for more than one condition can be present. Additional metadata for the sample(s) might also be present | Non-sparse and sparse matrix formats (e.g. MatrixMarket) stored as plain text or in containers such as hdf5 files and loom files. BED files as output of peak calling analysis. |
| 1.6 | Hi-C-like experiments | They express information such as presence/absence of contact, or intensity/frequency of contact between a position in the genome and another position, thus specifying a matrix of values. As with other data, different resolutions are possible – in general the rows and the columns of the matrix correspond to two binnings of the genome. Depending on the experiments, matrices could be sparse. | This information is typically coded in .hic files. |

# Requirements

This section introduces the definition of the requirements that an appropriate genomic annotation information representation should meet when the data types described in Table 3 are considered. In addition, support for transport should also be provided – the relevant requirements apply to all kinds of data, and are considered in a separate section.

## Mapping statistics

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement for mapping statistics** | **Rationale** |
| 4.1.1 | The solution shall support the coded representation of statistics related to aligned reads. | Pre-computed statistics that would otherwise require considerable computational resources. An indicative set of statistics can be found in Part 3 of ISO/IEC 23092 (MPEG-G). |

## Quantitative browser tracks

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement for quantitative browser tracks** | **Rationale** |
|  | The solution shall support lossless compression of the values associated to the track. | This is the basic information conveyed by files such as BigWig. A compressed representation is needed to optimize storage and visualization performance. |
|  | The solution should support other types of compression, e.g. related to different data resolutions. | Depending on the scale at which the information is displayed, a controlled loss of accuracy might be acceptable. |
|  | The solution shall support quantitative tracks to be associated to arbitrary genomic intervals. | Data is not necessarily available for all the reference |
|  | The solution shall support the embedding of tracks at different scales, i.e. each track will be a correspondence between an interval of *n* consecutive nucleotides and a number, with one interval every *n* nucleotides. | It might be useful to visualize the information at different resolution levels, and precomputed for efficiency (as for example is done in BigWig files) |
|  | The solution shall support a mapping between an arbitrary set of intervals on the reference and numerical values. | One might wish to represent information with a coarser resolution than the single nucleotide – for instance one number each ChIP-seq peak. Examples are the data represented in BedGraph files. |
|  | The solution shall allow the representation of meta-information associated with each track. | A description of how the track has been generated can be useful. |
|  | The solution to represent the meta data information should use the technology provided by Part 3 of ISO/IEC 23092 (MPEG-G). | To use already validated and compliant technology. |

## Variants

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement for variants** | **Rationale** |
| 4.3.1 | The solution shall support reference to meta-information about the study which produced it, and the methodology used to call variants. | We would like to be able to represent meta-information such as that stored in VCF headers |
|  | The solution should use the technology provided by Part 3 of ISO/IEC 23092 (MPEG-G). | To use already validated and compliant technology. |
|  | The solution shall support the unique identification of each variant. | Support for searches by name and visual representation in a genome browser |
|  | The solution should support the identification of the variant in terms of a known database. | If the variant corresponds to meta-information present in some database (for instance, dbDNP), the user might wish to easily identify them |
|  | The solution shall support a human-readable description of the variant. | Support for variant visualization in genome browsers |
|  | The solution shall encode the localization of each variant on the genome. | Variant position is an essential feature that needs to be easily accessed for displaying purposes |
|  | The solution shall describe the frequency of the variant in one or more samples. | This information is essential to understand the prevalence of a variant within the individual or population |
|  | The solution shall allow the representation and storage of additional fields in the form key, value associated to the variant. | This will allow the representation of study-specific information usually stored in additional fields of VCF files. |

## Genome functional annotation

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement for functional annotation** | **Rationale** |
| 4.4.1 | The solution shall support reference to meta-information about the entity that produced it, and the methodology used to generate it. | Fields “source” of GTF/GFF3 |
|  | The solution should use the technology provided by Part 3 of ISO/IEC 23092 (MPEG-G). | To use already validated and compliant technology. |
|  | The solution shall support the unique identification of each feature. | Support for searches by name and visual representation in a genome browser |
|  | The solution should support the identification of the feature in terms of a known database. | If the feature corresponds to meta-information present in some database (for instance, GenBank or ENSEMBL), the user might wish to easily identify them |
|  | The solution shall support a human-readable description of the feature. | Support for feature visualization in a genome browser |
|  | The solution shall encode the localization of each feature on the genome. | Feature position is an essential information that needs to be easily accessed for displaying purposes |
|  | The solution shall allow features to be associated to other features. | This is needed to be able to represent hierarchies of ontology concepts (for instance, in the case of genes there can be one or more transcript associated to a gene) |
|  | The solution shall allow users to describe each feature in terms of biological ontologies. | It is common practice to associate to each feature a description in terms of their biological meaning (for instance, in the case of coding RNAs, some nucleotides of a transcript would be a 5’ UTR, some would be a start codon, some would be coding sequence [CDS], etc.) |
|  | The solution shall allow the representation and storage of additional fields in the form key, value associated to the variant. | This will allow the representation of annotation-specific information usually stored in column 9 of GTF/GFF files. |

## Expression values

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement for expression values** | **Rationale** |
| 4.5.1 | The solution shall support the unique identification of each feature. | Support for searches by name and visual representation in a genome browser |
|  | The solution should support the identification of the feature in terms of a known database. | If the feature corresponds to meta-information present in some database (for instance, GenBank or ENSEMBL), the user might wish to easily identify them |
|  | The solution shall support a human-readable description of the feature. | Support for feature visualization in a genome browser. Such a description can usually be found in one of the fields in column 9 of GTF/GFF files |
|  | The solution shall encode the localization of each feature on the genome. | Feature position is an essential information that needs to be easily accessed for displaying purposes |
|  | The solution shall represent one or more numerical expression values associated with the feature in compressed form. | Typical types of values are counts and expression metrics such as FPKMs. Typically there will be more than one values associated with the feature due to the presence of several samples or several barcodes. |
|  | The solution shall allow the representation of meta-information associated with each sample. | A description of the sample for which expression has been measured can be useful. |
|  | The solution should use the technology provided by Part 3 of ISO/IEC 23092 (MPEG-G). | To use already validated and compliant technology. |
|  | The solution shall support the representation of sparse and dense matrices. | In scRNA-seq, results are expressed in matrices where rows identify the cells barcodes and columns identify the features. They may be sparse or dense depending of at which stage of the pipeline they are produced. The MatrixMarket format is an example of possible output format. |

## Hi-C-like experiments

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement for Hi-C-like experiments** | **Rationale** |
| 4.6.1 | The solution shall support one or more schemes for partitioning reference nucleotides into intervals. | In general contact data is expressed as matrices whose rows and columns correspond to different binning schemes depending on the scale. There could be different matrices corresponding to different scales, and in principle rows and columns could adopt independent binning schemes. |
|  | The solution shall support the efficient encoding of multidimensional numerical values. | A collection of matrices might be represented as a multidimensional array. The data might be sparse or dense, and hence an optimized solution might provide different encoding schemes. |
|  | The solution should allow the representation of meta-information associated with the experiment and each binning scheme and contact matrix. (MPEG-G). | A description of the way data has been collected can be useful. |
|  | The solution should use the technology provided by Part 3 of ISO/IEC 23092. | To use already validated and compliant technology. |

## Transport

|  |  |  |
| --- | --- | --- |
| **Req ID** | **Requirement for expression values** | **Rationale** |
| 4.7.1 | The compression process shall support the assessment of integrity. | Integrity check shall be possible by providing appropriate information. |
| 4.7.2 | The solution shall allow conveying information enabling data protection. | Ability to prevent unauthorized access shall be available. Information needed for protection of data (control for access, modification, publication, etc.) shall be conveyed. |
| 4.7.3 | The solution shall allow conveying information enabling accountability and traceability. | Data access and manipulation shall be traceable together with the identity of parties having access to data.  Information on how to verify integrity and authenticity of the data shall be conveyed. |
| 4.7.4 | The solution shall allow conveying information enabling transparency. | How and for which purpose the information is used shall be known. Usage restriction shall be applicable to the data. |
| 4.7.5 | The solutions shall support compressed data streaming. | This implies that data consumption shall be possible before data transfer completion. |

1. For the sake of simplicity, in the following we will always use the term genome even if the concept applies also to other kinds of references, for instance transcriptome references. [↑](#footnote-ref-1)