INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 M45267
January 2019, Marrakesh, MA

Source	**Requirements**
Status	**Input document**
Title	**Thoughts on future standardization activities in the area of genomic information representation**
Author	Paolo Ribeca (The Pirbright Institute), Jan Voges (Leibniz University Hannover), Tom Paridaens (Ghent University/imec), Mikel Hernaez (UIUC), Idoia Ochoa (UIUC), Claudio Alberti (GenomSys), Marco Mattavelli (EPFL), Giuseppe Codispoti (EPFL), Jaime Delgado (UPC), Daniel Naro (UPC)

Table of Contents

1 Purpose

The publication of Parts 1 and 2 of ISO/IEC 23092 (also known as MPEG-G) as International Standards is imminent. Also, the finalization of the remaining Parts 3, 4 and 5 is in sight. Thus, we urge MPEG to further extend its standardization activities in the area of genomic information representation.

This document contains a collection of topics that could be covered by MPEG-G “version 2”.

2 Annotation of coded genomic data

For the sake of simplicity, in this clause we will examine the case of a reference genome. However, most of the statements apply, with suitable modifications, to other cases (for instance RNA).

2.1 Motivation

Sequencing reads of different lengths (which depend on the technology used to generate them) and localized at one or more points on the DNA molecule they originate from, 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 MPEG-G hierarchy to be based on reads, which get organized in terms of, from bottom to top, Access Units, Datasets and Dataset Groups. However, most of the biological information relevant to a genome is associated to *intervals*. 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. 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.

Interval			
sequence	strand	lower position	higher position

This led to a situation where 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 followed by some secondary analysis, which is performed at the level of the genomic interval. For instance, in the case of RNA sequencing a possible analysis strategy would be first to align reads to the genome, and then to use the counts over the intervals defined by a genome annotation in order to learn facts about gene (dis)regulation. The latter operations are based on a number of file formats which use is widespread in the community, and which are all based on the concept of the interval (some examples being BED, GTF/GFF, etc.).

At the moment there is no way in MPEG-G of linking metadata to intervals. Information can only be associated with one of the levels of the existing read hierarchy, that is Dataset Group, Dataset or Access Unit. Labels are not fully fit for his purpose either, as they do not allow meta information to be associated with intervals. So unfortunately, it is not possible to support/embed secondary analysis directly in an MPEG-G file. That would be very desirable.

2.2 Purposes

The introduction of a new concept of metadata associated with intervals (also called annotation, in keeping with the terminology used by biologists) would allow the MPEG-G standard to represent a wealth of additional functionality, and to clarify the scope of a few concepts that are currently defined in the standard at a different abstraction level, such as protection.

More in detail:

Protection, which is at the moment defined in terms of Access Units, could be extended with an interval-based notion. In fact, MPEG-G already accepts that protection is concerned with genomic intervals – however at the moment the concept of protecting an interval is left open to the implementation, as there might be more than one Access Units spanning the same interval, entirely or partially. This is reflected by protection API functions in Part 3, whereby queries are made on intervals and a vector of protections is returned, one per Access Unit; how to deduce per-segment information is left to the application.

Statistics could be pre-computed. At the moment this is difficult to do because: (1) pre-computed statistics can only be combined if they refer to the same interval, (2) the granularity of the Access

Unit is defined by the encoder, and hence one cannot in general count on there being any useful correspondence between, say, the size of Access Units and what could be a binning scheme useful to pre-compute statistics, (3) users typically wish to query the number of reads, and other statistics, either by file/dataset (which would be supported by the current hierarchy) or by interval (which, as discussed for the protection, is not). 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. However, that is not possible without the concept of interval, and one is left with a choice between an expensive re-computation of the whole interval, and a very space-consuming pre-computation of the statistics for all AUs (which is also somehow pointless due to the possible presence of partially overlapping AUs for which the information could not be easily merged).

Biological annotations of the sequence (such as the localization of gene models as lists of UTRs, exons and coding intervals, currently expressed in formats such as BED, GTF/GFF etc.) could be embedded in the MPEG-G file if one could represent in it meta-data associated with genomic regions. That would simplify and make more robust a number of downstream analysis pipelines, such as those able to process RNA-sequencing data (at the moment reference and annotation are specified as separate concepts, which creates consistency problems for future reproducibility as the link between data and annotation can be easily lost).

Variants (usually represented as VCF files) are also a specialized form of annotation linked to genome intervals (for instance SNPs, which are by far the most frequent variant considered in today's re-alignment pipelines, are genomic intervals comprising a single nucleotide). Being able to store variants in the file would allow to store and share the results of variant calling into MPEG-G files.

Browser tracks (typically represented in Wig/BigWig format) are files whereby intervals in the genome are associated to a scalar quantity, typically coverage. I.e., to every nucleotide the number of reads generated by some sequencing experiment and going through that nucleotide is listed. Those tracks are used for quickly displaying the biological quantity in genome browsers (for instance, the Human Genome Browser at UCSC) without having to re-compute large amounts of data. It should be noted that a user might wish to query the genome browser at different scales; hence, in order to avoid re-computation typical file formats include coverages at different scales (one value per nucleotide, one value per 100, 10000, 100000 nucleotides). Again, all this information can naturally be expressed in terms of interval. If one could store pre-computed tracks into MPEG-G files they could be directly used as inputs to genome browsers.

Expression values. They are special cases of coverage (in fact, some form of average of RNA-sequencing read coverage along a transcript). Being able to store expression values in MPEG-G files would mean that they could be used as input/output of programs (such as edgeR or DEseq) doing differential expression analysis.

Hi-C experiments. They sequence couples of small genome fragments that are in physical contact with one another. They typically produce a symmetric "contact matrix" which splits the genomes into bins (i.e., intervals) and associates an integer count to each couple of possible bins, the count of cell *(i, j)* being incremented every time a couple of fragments is found such that one of the

fragments belongs to bin *i*, and the other one to bin *j*. So one could store Hi-C data as annotations for couples of genomic intervals.

HiC Contact		
Interval1	Interval2	Value (int)

Labels might be extended to be linked to the concept of an interval.

2.3 What to standardize

The data possibly associated with intervals can be of a different nature. We distinguish two main categories:

1. Genome annotations typically contain metadata about biological function, and similar to SAM they allow virtually any kind of content to be embedded into "extensible" fields.
   For instance, fields #9 of GTF and GFF can contain an arbitrarily long list of attributes. This possibility originated several "dialects" of GTF[1], which have a set of core features in common–-the ones specifying the main hierarchy of gene, transcript, exon—but also a number of incompatible additional features whose semantics are not really specified. On the other hand, GFF3 has a formal format definition[2]; however, in actual fact the definition allows for inclusion of terms coming from a very large number of Gene Ontology databases, thus making the interpretation of features once again very difficult.
   The case of VCF is even worse, with a "specification" whose complexity is probably even larger than that of SAM.
   In light of those considerations, one possibility might be to allow for the syntax of each annotation to be specified by some extensible XML-like mechanism—there would be one syntax for genomic annotations, one for variant calling, and so on. Some core features with a defined meaning (for instance the gene—transcript—exon hierarchy in genomic annotations) would always be present, and additional fields might be defined.
   Given that the number of features per genome is usually low (~20,000) and the amount of information associated to it relatively low, compression (at least in the case of genomic annotations) would not be a primary concern, and might be implemented through some general mechanism. The case of VCF should be considered more carefully, as there can be millions of single-nucleotide variants in each database.
2. Other annotations such as browser tracks or Hi-C data would be more data-intensive, and compression techniques should be considered more carefully. For instance, Wig/BigWig files associate a floating-point number to genome intervals, and potentially intervals might be as narrow as a single nucleotide; Hi-C experiments generate very large matrices, depending on the selected size of the genomic bins. [To be more precise, in the case of BigWig-like tracks there would also be an additional metadata structure, due to the presence of a set of pre-computed track at different genomic scales. I.e. one would generate a set of tracks defined in terms of genomic bins of different sizes, and those tracks would be connected through some normative semantics meaningful to genome browsers.]
   In this case the nature of the annotation would be simple (a floating-point value in the case of Wig/BigWig-like genome browser tracks, or an integer number in the case of Hi-C data) but efficient compression would play a much more important role.

[1] See for instance http://mblab.wustl.edu/GTF22.html.
[2] See https://github.com/The-Sequence-Ontology/Specifications/blob/master/gff3.md.

2.4 Conclusion

The introduction of the concept of interval and metadata associated with it would be essential to make MPEG-G the central interchange format for a variety of tools. In particular, one might store in MPEG-G files not only the results of alignment (primary analysis) but also what is produced by more downstream tools (variants, annotation, gene expression, genome browser tracks). Different versions of the MPEG-G file containing progressively more metadata would constitute the starting point, the intermediates and the ending point of the pipeline. That is highly desirable if one wishes MPEG-G to become not only just another compressed file format, but the central facilitator of a complex ecosystem.

3 Compressed representation and handling of genomic variants

As elaborated on above, it would be beneficial to be able to store genomic variants (usually represented as VCF files) in MPEG-G files. Recent research [3] focuses explicitly on the compression of collections of such variants. The integration of such compression approaches into MPEG-G would be desirable.

4 Investigation of coding technologies

The investigation of coding technologies that offer higher coding efficiency (at the potential cost of lower compression ratios) or higher coding effectiveness (e.g. for archiving) would be desirable.

5 New use cases

What follows is a collection of use cases.

1. Large or small alterations: Some driver mutations or alterations (such as fusions, small indels or single nucleotide variants) for cancer may be very rare. Once someone thinks they may have identified a rare driver mutation or alteration, they may want to look for evidence of the mutation or alteration in other tumor samples, even perhaps at low frequency. MPEG-G should provide support for finding a larger number of publicly-available relevant samples for inclusion in a validation set and for assessing the existence of the mutation or alteration in the validation set at high speeds.

2. Genomic version updates: MPEG-G should enable reprocessing and updating old sequencing data files to new genome NCBI data build and use the samples for analysis.

3. Multi-aligner files: The program chosen for alignment can affect downstream inference. To help mitigate this, one might like to perform alignment by multiple alignment programs (using multi aligners). MPEG-G should enable storing of multiple alignments for sequencing data.

4. Single cells: Millions of single-cells are processed in batches across different locations by the Human Cell Atlas (HCA) project. E.g., the census of the immune cells project in the HCA contains ~530,000 cells resulting in a 1.3 TB data set. However, this is just one of the studies out of the hundreds of datasets that will be released by the HCA team. Thus, MPEG-G should provide the means for storage and transmission of single-cell sequencing data.

[3] See for instance https://www.ncbi.nlm.nih.gov/pubmed/27587665.