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

ISO/IEC JTC1/SC29/WG11 m50913
October 2019, Geneva, CH

Source ISO/IEC JTC1/SC29/WG11
Status Draft
Title MPEG-G Best Practices Deployment Guide
Authors Dunling Li (BTS), Claudio Alberti (GenomSys), Jaime Delgado (UPC), Jan Voges (LUH), Itaru Kaneko (Nagoya City University), Marco Mattavelli (EPFL), Patrick Cheung (Philips), Paolo Ribeca (The James Hutton Institute)

Background

This document is a second draft deployment guide aiming at describing best practices for the implementation of MPEG-G compliant systems. The goal is to provide guidelines for end users with limited or no experience of MPEG technology. The current version is purely a collection of ideas on the possible structure of the document and drafted around the following sections:

1. an introduction mainly based on excerpts from the MPEG-G overview published on the BioRxiv portal [1]
2. a section describing the deployment environment for MPEG-G implementations
3. a list of use cases taken into account during the standardization process
4. a section devoted to software implementations of MPEG-G compliant applications focusing on encoding and decoding issues, integration with APIs and conformance testing.

The document is submitted to the larger MPEG-G working group for discussion and feedback.

Table of Contents

1 Scope of MPEG-G (Marco, all)

Genomic data, the genome and DNA of an organism, are widely used in biotechnology, medical research, clinical therapy, new drug development, social science applications, etc. They are obtained using genome sequencing technology. As advanced next-generation sequencing (NGS), also known as high-throughput sequencing (HTS), technology makes sequencing genomes much faster and cheaper, the cost of sequencing a whole human genome has reduced from $20 million in 2004 to $1000 in 2015. It is expected that within the next few years such cost will drop to about $100 [1]. Sequencing the first human genome took 13 years (1990 ~ 2003) to complete while it only needed an hour in 2017. While this enables the ubiquitous use of genomic information as an everyday practice in several fields, such as personalized medicine. However, genomic sequencing has generated an ever-growing and enormous amount of data that has become a serious obstacle to the wider diffusion of sequencing in public health. Currently, single sequencing system can deliver the equivalent of 9,000 whole human genomes per year, which accounts for almost 1 PB of data per year. The associated IT costs related to storing, transmitting, and processing such large volumes of data will soon greatly exceed the cost of sequencing. The lack of appropriate representations and efficient compression technologies is widely recognized as a critical element limiting the potential of genomic data usage for scientific and public health purposes [2]. This led to the largest coordinated international effort: Moving Picture Expert Group Genomic Information Representation (MPEG-G) standardization.

The MPEG-G standard specifies a compressed data format that enables large scale genomic data processing, transport and sharing. It is the first ISO/IEC standard that addresses the problems and limitations of current genomic data formats towards a truly efficient and economical handling of genomic information. It provides the means to implement leading-edge compression technology achieving about 100:1 compression ratio on raw data, i.e. more than 10x improvement over the BAM[1] format. The standard also provides the following currently-needed functionalities:

- **Selective access to compressed data**: Indexing tools embedded in an MPEG-G file enable several types of selective access to compressed data that can be combined in the same query.
- **Data streaming**: MPEG-G supports the packetization of compressed data for transport to receiving devices that can start processing the data before transmission is completed.
- **Compressed file concatenation**: MPEG-G files can be concatenated without the need to decode and re-encode them.
- **Genomic studies aggregation**: Several related genomic studies can be encapsulated in the same MPEG-G file while still being separately accessible. Additionally, transversal queries over multiple studies are possible (e.g. “select chromosome 1 of all compressed samples”).
- **Enforcement of privacy rules**: Data encoded in an MPEG-G file can be linked to multiple owner-defined privacy rules, which impose restrictions on data access and usage.
- **Selective encryption of sequencing data and metadata**: The encryption of genomic information is supported by MPEG-G at different levels in the hierarchy of MPEG-G logical data structures.

[1] Binary Alignment Map (BAM) is the comprehensive raw data of genome sequencing; it consists of the lossless compressed binary representation of the Sequence Alignment Map (SAM)

- **Annotation and linkage of genomic segments in the compressed domain**: MPEG-G supports the annotation of genomic segments. Additionally, MPEG-G provides support for linking segments within a single genomic sample or across multiple genomic samples.
- **Interoperability with main existing technologies and legacy formats**: Conversion to/from legacy format such as FASTQ, SAM or BAM is supported by MPEG-G.
- **Incremental update of sequencing data and metadata**: MPEG-G files can be incremented with sequencing data and metadata without requiring decompression and re-compression of pre-existing data.
- application programming interfaces to the compressed data?
- data protection mechanisms?

Finally, interoperability and integration with existing genomic information processing pipelines is enabled by supporting conversion from/to file formats such as FASTQ and SAM/BAM. Also, the maintenance of the standard guarantees the perenniality of applications using MPEG-G. The MPEG-G standard consists of the following five (6?) parts:

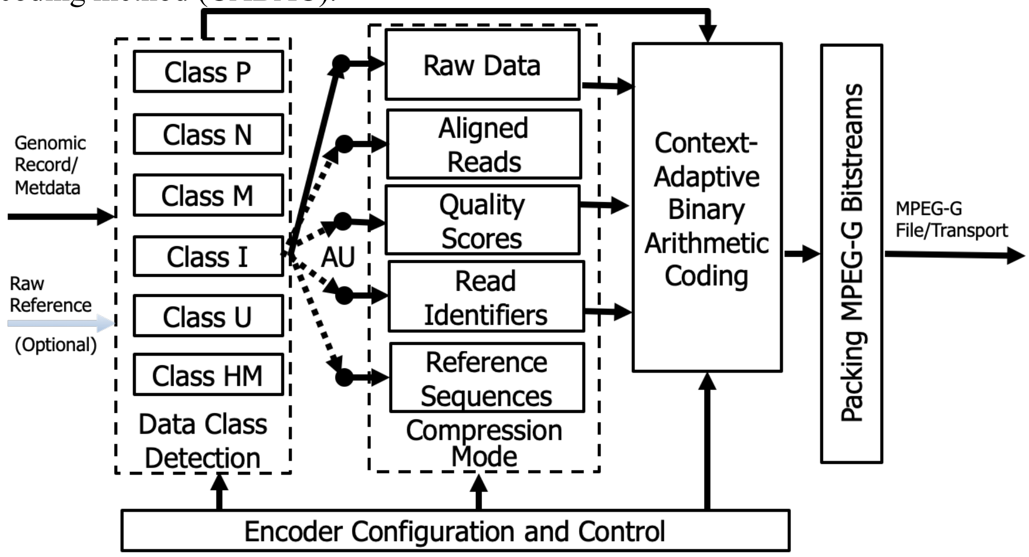
1. **Transport and storage of genomic information representation** [1]

Part 1 specifies how the genomic data is organized within MPEG-G structures for transport (i.e., streaming) and storage. Formats of genomic record, reference record, MPEG-G file and transport stream are defined here. It introduces Access Unit as the actual container of the compressed genomic data and provides a reference conversion process among different formats.

2. **Coding of genomic information**

Part 2 specifies the syntax and methods of MPEG-G lossless compression for different compression modes and lossy compression for quality scores. Like other successful MPEG standards such as MP3 for audio and AVC/H.264 for video, which have been enabling the revolution of digital media field in past 30 years, MPEG-G only specifies the decoding process while the encoding process is left open to algorithmic and implementation-specific innovations. The normative input of an MPEG-G deterministic decoding process is a concatenation of data structures called Data Units. Data Units can be of three types according to the type of conveyed data. A Data Unit of type 0 encapsulates the decoded representation of one or more reference sequences, a Data Unit of type 1 contains parameters used during the decoding process and a Data Unit of type 2 contains one Access Unit. All MPEG-G conformed decoders produce identical outputs from the multiplexed bitstreams included in MPEG-G files and the data streams in streaming scenarios.

This part specifies five compression modes for raw sequencing data (high compression vs low delay), aligned reads (with vs w/o reference), quality scores (lossless vs lossy), read identifier and reference sequence. The typic structure of MPE-G encoding process is shown in Figure 1. The input data of the encoder are genomic records or metadata with option of reference data while its output is MPEG-G file or transport streams. The encoder first categories the input into six data classes and generate access units or descriptor streams according to their data class types, then invoke corresponding compression mode and create binary data stream to be further compressed by context adapted binary arithmetic entropy coding method (CABAC).



3. Metadata and APIs

Part 3 specifies metadata format and provides genomic data representation APIs to meet the urgent needs from genomic information community and support interoperability among existing tools and systems. MPEG-G metadata specifies how an MPEG-G compliant bitstream can be integrated with metadata describing, for example, a genomic study or a sequencing run. It includes the specification of normative interfaces to access MPEG-G data from external systems, the specification of mechanisms to implement access control, integrity verification, as well as authentication and authorization mechanisms. This part also contains an informative section devoted to the mapping between SAM and MPEG-G data structures. MPE-G APIs specifies interface and controlled access to MPEG-G file and transport formats, integrity verification, authentication and backward compatibility with existing SAM content. The specified APIs provides a normative way to access and manipulate MPEG-G Compliant genomic content for its implementation. The operations provided by this API affect different aspects of genomic information and its associated metadata, protection information and other fields contained at each level. They may include functionalities such as providing access, performing modifications, authorizing operations or integrity verification.

4. Reference Software

Part 4 provides a normative Reference Software. The Reference Software is normative in the sense that any conforming implementation of the decoder, taking the same conformant compressed bitstreams and using the same normative output data structures, will output the same data.

5. Conformance

Part 5 provides a means to test and validate the correct implementation of the MPEG-G technology in different devices and applications to ensure the interoperability among all systems. It specifies a normative procedure to assess conformity to the standard on an exhaustive set of compressed data.

The MPEG-G Genomic Information Database is a collection of statistically meaningful sequencing data used to assess the performance of genomic information compression technologies. Besides the actual sequencing data, the database contains a set of reference sequences and supporting data needed for variant calling experiments (see Methods). When compiling the database special emphasis was put on incorporating data with as much diversity as possible. Hence, it contains data generated by different sequencing technologies,

produced for the purpose of conducting different experiment types (e.g., WGS, RNA-seq, etc.), and originating from samples across different species such as H. sapiens, D. melanogaster or E. coli.
6. ????
Part 6 provides

2 MPEG-G Privacy Protection and Security (Jaime)

There are various regulations applied to the storing, transmission and analysis of genomic data as seen in section 7.4. The regulation varies in each country. To comply with such regulations, the user can use the Privacy Protection tools and Security tools of MPEG-G.

3 MPEG-G Use Cases

3.1 Standalone Genomic Data Application (analysis pipeline, Patrick)

E.g. implemented in a clinical/research facility

3.1.1 Topology

3.1.2 Interoperability

3.1.3 Scalability

3.1.4 Privacy Protection and Security

3.2 Cloud-based Genomic Data Sharing System: BIFROST Platform (Dunling)

3.2.1 Topology

3.2.2 Interoperability

3.2.3 Scalability

3.2.4 Privacy Protection and Security

3.3 Genomic Data Streaming (GenomSys)

3.3.1 Topology

3.3.2 Interoperability

3.3.3 Scalability

3.3.4 Privacy Protection and Security

3.4 Genomic metadata processing (Jaime)

3.4.1 Topology

3.4.2 Interoperability

3.4.3 Scalability

3.4.4 Privacy Protection and Security

3.5 Genomic data archival (GenomSys)

3.5.1 Topology

3.5.2 Interoperability

3.5.3 Scalability

3.5.4 Privacy Protection and Security

3.6 More if needed TBD

3.6.1 Topology

3.6.2 Interoperability

3.6.3 Scalability

3.6.4 Privacy Protection and Security

4

5 Software Implementation

5.1 MPEG-G codecs (GenomSys)

5.1.1 Encoder

5.1.2 Decoder

5.1.3 Software Development Kit

5.2 Software Integration with MPEG-G APIs (Jaime, Paolo?)

5.3 Conformance Integration Test (GenomSys)

5.4 Conversion of legacy formats (Jan)

6 Conclusion

7 References

[1]. C. Alberti, T. Paridaens, J. Voges, D. Naro, J.J. Ahmad, M. Ravasi, D. Renzi, G. Zoia, I. Ochoa, M. Mattavelli, J. Delgado, and M. Hernaez, “An introduction to MPEG-G, the new ISO standard forgenomic information representation”
[2]. ISO/IEC JTC1/SC29/WG11, “White paper on the objectives and benefits of the MPEG-G standard,” Jan 2018
[3]. I. Numanagic, J. K. Bonfield, F. Hach, J. Voges, J. Ostermann, C. Alberti, M. Mattavelli, and S. C. Sahinalp, “Comparison of high-throughput sequencing data compression tools,” Nature Methods, vol. 13, pp. 1005–1008, Oct. 2016.
[4]. ISO/IEC JTC1/SC29/WG11, “Genomic Information Representation Metadata,” July 2017
[5]. ISO/IEC JTC1/SC29/WG11, “Genomic Information Representation APIs,” July 2017
[BTS-1]. ISO/IEC JTC1/SC29/WG11, “Text of FDIS ISO/IEV 23092-2 Coding of Genomic Information,” N18728, Gothenburg, SE – July 2019

[BTS-2]. ISO/IEC JTC1/SC29/WG11, “Text of ISO/IEC FDIS 23092-1 Transport and Storage of Genomic Information,” N18141, Marrakech, MO – January 2019

8 Appendixes

8.1 File Formats

8.1.1 Genomic Record Format (Dunling)

Genomic record is the fundamental structure of the ISO/IEC 23092 series data representation. It is a data structure consisting of either a single sequence read, or a paired sequence read, and its associated sequencing and alignment information; it may contain detailed mapping and alignment data, a single or paired read identifier (read name) and quality values. Without breaking traditional approaches, the genomic record provides a more compact, simpler and manageable data structure grouping all the information related to a single DNA template, from simple sequencing data to sophisticated alignment information.

Genomic Record format is specified in clause 13 and its syntax is defined in Table 126 in ISO/IEC 23092-2. Table 1 in ISO/IEC 23092-2 enumerates all the types of data that a genomic record can contain. The genomic record file is a binary file which can be used as MPEG-G decoder output or encoder input. To make new users to grasp genomic record format quickly, ***Table 1*** and ***Table 2*** use FASTA and FASTQ files as examples to show the conversion from existing unaligned read data formats.

The FASTA and FASTQ examples are ls_orchid.fasta and SXX123456.fastq. All the examples and their corresponding genomic record files are available at www.????? The contents of ls_orchid.fasta are the follows:

```
>gi|2765658|emb|Z78533.1|CIZ78533 C.irapeanum 5.8S rRNA gene and ITS1 and ITS2 DNA
CGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTGATGAGACCGTGGAATAAAC
GATCGAGTG
AATCCGGAGGACCGGTGTACTCAGCTCACCGGGGGCATTGCTCCCGTGGTGACCCTGATT
TGTTGTTGGG
CCGCCTCGGGAGCGTCCATGGCGGGTTTGAACCTCTAGCCCGGCGCAGTTTGGGCGCCAA
GCCATATGAA
```

```
AGCATCACCGGCGAATGGCATTGTCTTCCCCAAAACCCGGAGCGGCGGCGTGCTGTCGCG
TGCCCAATGA
ATTTTGATGACTCTCGCAAACGGGAATCTTGGCTCTTTGCATCGGATGGAAGGACGCAGCG
AAATGCGAT
AAGTGGTGTGAATTGCAAGATCCCGTGAACCATCGAGTCTTTTGAACGCAAGTTGCGCCCG
AGGCCATCA
GGCTAAGGGCACGCCTGCTTGGGCGTCGCGCTTCGTCTCTCTCCTGCCAATGCTTGCCCG
GCATACAGCC
AGGCCGGCGTGGTGCGGATGTGAAAGATTGGCCCCTTGTGCCTAGGTGCGGCGGGTCCAA
GAGCTGGTGT
TTTGATGGCCCGGAACCCGGCAAGAGGTGGACGGATGCTGGCAGCAGCTGCCGTGCGAAT
CCCCCATGTT
GTCGTGCTTGTCGGACAGGCAGGAGAACCCTTCCGAACCCCAATGGAGGGCGGTTGACCG
CCATTCGGAT
GTGACCCCAGGTCAGGCGGGGGCACCCGCTGAGTTTACGC

>gi|2765657|emb|Z78532.1|CCZ78532 C.californicum 5.8S rRNA gene and ITS1 and ITS2 DNA
CGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTGTTGAGACAACAGAATATATG
ATCGAGTG
AATCTGGAGGACCTGTGGTAACTCAGCTCGTCGTGGCACTGCTTTTGTCGTGACCCTGCTT
TGTTGTTGG
GCCTCCTCAAGAGCTTTCATGGCAGGTTTGAACTTTAGTACGGTGCAGTTTGCGCCAAGTCA
TATAAAGC
ATCACTGATGAATGACATTATTGTCAGAAAAAATCAGAGGGGCAGTATGCTACTGAGCATGC
CAGTGAAT
TTTTATGACTCTCGCAACGGATATCTTGGCTCTAACATCGATGAAGAACGCAGCTAAATGCG
ATAAGTGG
TGTGAATTGCAGAATCCCGTGAACCATCGAGTCTTTGAACGCAAGTTGCGCTCGAGGCCAT
CAGGCTAAG
GGCACGCCTGCCTGGGCGTCGTGTGTTGCGTCTCTCCTACCAATGCTTGCTTGGCATATCG
CTAAGCTGG
CATTATACGGATGTGAATGATTGGCCCCTTGTGCCTAGGTGCGGTGGGTCTAAGGATTGTT
GCTTTGATG
GGTAGGAATGTGGCACGAGGTGGAGAATGCTAACAGTCATAAGGCTGCTATTTGAATCCCC
CATGTTGTT
GTATTTTTTCGAACCTACACAAGAACCTAATTGAACCCCAATGGAGCTAAAATAACCATTGG
GCAGTTGA
TTTCCATTCAGATGCGACCCCAGGTCAGGCGGGGCCACCCGCTGAGTTGAGGC

……

>gi|2765564|emb|Z78439.1|PBZ78439 P.barbatum 5.8S rRNA gene and ITS1 and ITS2 DNA
CATTGTTGAGATCACATAATAATTGATCGAGTTAATCTGGAGGATCTGTTTACTTTGGTCACC
CATGGGC
ATTTGCTGTTGAAGTGACCTAGATTTGCCATCGAGCCTCCTTGGGAGCTTTCTTGTTGGCGA
GATCTAAA
CCCCTGCCCGGCGGAGTTGGGCGCCAAGTCATATGACACATAATTGGTGAAGGGGGTGGT
AATCCTGCCC
TGACCCTCCCCAAATTATTTTTTTAACAACTCTCAGCAACGGATATCTCGGCTCTTGCATCGA
TGAAGAA
CGCAGCGAAATGCGATAATGGTGTGAATTGCAGAATCCCGTGAACATCGAGTCTTTGAACG
CAAGTTGCG
CCCGAGGCCATCAGGCCAAGGGCACGCCTGCCTGGGCATTGCGAGTCATATCTCTCCCTT
AATGAGGCTG
TCCATACATACTGTTCAGCCGGTGCGGATGTGAGTTTGGCCCCTTGTTCTTTGGTACGGGG
GGTCTAAGA
GCTGCATGGGCTTTGGATGGTCCTAAATACGGAAAGAGGTGGACGAACTATGCTACAACAA
```

```
AATTGTTGT
GCAAATGCCCCGGTTGGCCGTTTAGTTGGGCC
```

There are 94 sequences in the file. Its genomic record format is shown in the table below:

Table 1 Genomic Record File example from a FASTA file

Field		Content	Data type	Number of bits
# of template segments		0	Unsigned integer	8
# of record segements		94	Unsigned integer	8
# of alignments		0	Unsigned integer	16
Class ID		6	Unsigned integer	8
Read group length		0	Unsigned integer	8
Read_1_first		0	Unsigned integer	8
Read length	1st record segment	82+740=822	Unsigned integer	24
	2nd record segment	85+753=838	Unsigned integer	24
	:	:	:	:
	Nrth record segment	81+592=673	Unsigned integer	24
QV depth		0	Unsigned integer	8
Read name length		9	Unsigned integer	8
Read name		NULL	Unsigned char	0
Read group		NULL	Unsigned char	0
1st record segment		>gi|2765658… TTTACGC	String (UTF-8)	8*822
2nd record segment		>gi|2765657… TTGAGGC	String (UTF-8)	8*838
:		:	:	:
94th seqence		>gi|2765564… TTGGGCC	String (UTF-8)	8*673
Record flag		?	Unsigned integer	8
More alignment		0	Unsigned integer	8

The contents of the FASTQ file example is the follows:

```
@SXX123456.1 EV7PG6Z09FHTGT
TTCTTTGATTTCTCTATTGAAGTCTACCGGTATATCTTTTGTTCCCAGTCCTAAATGCCATTTA
CCGATGATAGCTGT
+SXX123456.1 EV7PG6Z09FHTGT
A<<B?.<;A>+<;<<<@9<?9<<<<<?9?9<<<<<<B@4%<A;A>+<;<@:<B@/<<A;<B?-
<@;<<<<<<<<<<<<A
@SXX123456.2 EV7PG6Z09FTQ8Z
AGAATTTCAGTGCCTCCCTCTCAACCTTGACCTCCGGTACCTCCTACTATATCCGTGCATAC
GT
+SXX123456.2 EV7PG6Z09FTQ8Z
;<A;@=);<<<<=6<?<)<<<<@;>7<5;<?9<@9@;<<@9<@9<<<<<<<;@:<<<<<<<<<<
@SXX123456.3 EV7PG6Z09FVELJ

……

+SXX123456.49 EV7PG6Z09FL1WY
<<<<A?3$<@:A>,<<;;<<<;<CA7,;;>7=:%<;<CA7+B@2">;'<?8<CA8/&A?2";<@9;<;<<<<<A<;;<A
<B?.<<@=+<<<<<<
@SXX123456.50 EV7PG6Z09FSZI0
GACTTTTCAGCTTTCCCCAAAGGGAACCGTCTTCACTCTGTGAACCGTCTTCGCTGCCGATA
```

```
CGGGCGTTGATATCGGCCAGCAATGCGGCCAG
+SXX123456.50 EV7PG6Z09FSZI0
:<<B@4%<::;A>+B@2"A>+>;&?9=6<<<?9<;<;<<<<<>7?9<<<?8;;<<<@9<<<<<A>,;<@;<<<<;<
A;>7<<;@:;<<A;>7<<
```

There are 50 reads in the file, its corresponding genomic record is shown in the table below:

Table 2 Genomic Record Example from a FASTQ file

Field		Content	Data type	Number of bits
# of template segments		0	Unsigned integer	8
# of record segements		50	Unsigned integer	8
# of alignments		0	Unsigned integer	16
Class ID		6	Unsigned integer	8
Read group length		0	Unsigned integer	8
Read_1_first		0	Unsigned integer	8
Read length	1st record segment	78	Unsigned integer	24
	2nd record segment	64	Unsigned integer	24
	:	:	:	:
	Nrth record segment	86	Unsigned integer	24
QV depth		1	Unsigned integer	8
Read name length		9	Unsigned integer	8
Read name		SXX123456	Unsigned char	8*9
Read group		NULL	Unsigned char	0
1st read	sequence	TTCTTTGA … TAGCTGT	String (UTF-8)	8*78
	1st quality value	<<B?.<;A … <<<<<<A	String (UTF-8)	8*78
2nd read	sequence	AGAATTTC … CATACGT	String (UTF-8)	8*64
	1st quality value	;<A;@=); … <<<<<<<	String (UTF-8)	8*64
:	:	:	String (UTF-8)	:
50th Read	sequence	GACTTTTC … CGGCCAG	String (UTF-8)	8*86
	1st quality value	:<<B@4%< … <A;>7<<	String (UTF-8)	8*86
Record flag		?	Unsigned integer	8
More alignment		0	Unsigned integer	8

Add BAM for aligned reads later

8.1.2 Raw Reference Format

MPEG-G Raw reference format is specified in subclause 7.2 and its syntax is defined in Table 5 in ISO/IEC 23092-2. The raw reference file is a binary file and used as encoder input or decoder output. The table below shows an example of converting a FASTA file (ls_orchid.fasta) to a raw reference format.

Table 3 Raw reference file example from a FASTA file

Field	Content	Data type	Number of bits

# of sequences		94	Unsigned integer	16
1st Seq	Sequence ID	0	Unsigned integer	16
	Start Position	0	Unsigned integer	40
	End Position	739	Unsigned integer	40
	Sequence	CGTAACA…TTTACGC	String (UTF-8)	8*740
2nd Seq	Sequence ID	1	Unsigned integer	16
	Start Position	0	Unsigned integer	40
	End Position	752	Unsigned integer	40
	Sequence	CGTAACA…TTGAGGC	String (UTF-8)	8*753
….	:	:	:	:
	:	:	:	:
94th Seq	Sequence ID	93	Unsigned integer	16
	Start Position	0	Unsigned integer	40
	End Position	592	Unsigned integer	40
	Sequence	CATTGTTG…TAGTTGGGCC	String (UTF-8)	8*592

8.1.3 MPEG-G File Format (Dunling)

MPEG-G file format is specified in Clause 6 in ISO/IEC 23092-2. Table 4 and Figure 4 in subclause 6.1 present the overall data structures and hierarchical encapsulation levels. MPEG-G File, a binary file, includes a file header and a dataset group which contains a nest structure of datasets, access units and blocks. The file header format is specified in subclause 6.6.2 while the formats of dataset group, dataset, access unit and block are defined in subclause 6.5.1 to 6.5.4 respectively. The syntax of MPEG-G File header and dataset group are specified in table 30 and Table 8 in ISO/IEC 23092-1 respectively. Table 4 here shows the MPEG-G file header format while Table 5 and Table 6 show dataset group syntax and its format respectively. Table 6 contains dataset format, which is specified in subclause 6.5.2 and its syntax is defined in Table 18 in ISO/IEC 23092-1. Table 7 and Table 8 here show the dataset syntax in ISO/IEC 23092-1 and dataset format.

Table 4 MPEG-G File Header Format

Field		Content	Data type	# of bytes
key		flhd	character	4
length		L_{flhd}	Unsigned int	64
Major brand		MPEG-G	character	6
Minor brand	Version #	Year of release	digit	2
	Amendment #	0	digit	1
	Corrigendum #	0	digit	1
Compatible brands	1st	CB_1 (minor brand)	character	4
	2nd	CB_2 (minor brand)	character	4
	:	:	:	:
	Mth M=(L_{flhd} -22)/4	CB_M (minor brand)	character	4

Table 5 Dataset Group Syntax in ISO/IEC 23092-1

Field	Content	Data type	# of bytes
key	dgcn	character	4

Length (in bytes)		L_{dgcn}	Unsigned int	8
Dataset group header		Table 9	Gen_info	L_{dghd}
Reference		Table10	Gen_info	L_{rfgn}
Reference metadata		Table 13	Gen_info	L_{rfmd}
Label list		Table 14	Gen_info	L_{labl}
DG metadata		Table 21	Gen_info	L_{dgmd}
DG protection		Table 22	Gen_info	L_{dgpr}
Datasets	1st	Table 18	Gen_info	L_{dtcn_1}
	2nd	Table 18	Gen_info	L_{dtcn_2}
	:	Table 18	Gen_info	:
	Nth N=(L_{dghd}-14)/2	Table 18	Gen_info	L_{dtcn_N}

Table 6 Dataset Group Format

Fields			Contents	Data type	Sizes	
					bytes	bits
key			dgcn	char	4	
Length			L_{dgcn}	uint		64
Dataset Group Header	key		dghd	char	4	
	Length		L_{dghd}	uint		64
	Dataset Group ID		G_ID	uint		8
	Version #		Ver	uint		8
	Dataset ID	1st	ID_1	uint		16
		2nd	ID_2	uint		16
		:	:			
		Nth N=(L_{dghd} -14)/2	ID_N	uint		16
Reference	key		rfgn	char	4	
	Length		L_{rfgn}	uint		64
	Reference ID		R_ID	uint		8
	Reference name		Rname	char	v	
	Reference major version		Rmajor	uint		16
	Reference minor version		Rminor	uint		16
	Reference patch version		Rpatch	uint		16
	Sequence count		N_{seq}	uint		16
	Sequence Name	1st seq name	Name_seq$_1$	string		v
		2nd seq name	Name_seq$_2$	string		v
		:	:	:	:	:

				N_{seq}th seq name	Name_seq_{Nseq}	string		v
	Reserved				0	uint	7	
	External Reference Flag				Fr		1	
	Fr =1	Ref uri			ref_uri	string		v
		Checksum algorithm			Chsum_id	uint	8	
		Reference type			Tr	Uint	8	
		Tr=MPEG_REF	External dataset group ID		G_ID_ext	Uint	8	
			External dataset ID		ID_ext	Uint	8	
			Ref checksum		Chsum	int	N_{ch}	
		ELSE	1^{st} seq checksum		Chsum_1	int	Nch_1	
			2^{nd} seq checksum		Chsum_2	int	Nch_2	
			:		:	:	:	
			N_{seq}^{th} seq checksum		Chsum_{Nseq}	int	Nch_{Nseq}	
	ELSE	Internal dataset group ID			G_ID_int	Uint	8	
		Internal dataset ID			ID_int	uint	8	
Reference Metadata	Key				rfmd	char		4
	length				L_{rfmd}	uint	64	
	Dataset group ID				G_ID	uint	8	
	Reference ID				R_ID	uint	8	
	Reference metadata value				ISO/IEC 23092-3			
Label List	Key				labl	char		4
	Length				L_{labl}	uint	64	
	Dataset group ID				G_ID	uint	8	
	Num Labels				N_{lab}	uint	16	
	1^{st} Label	Key			lbll	char		4
		Length			L_{lbll}	uint	64	
		Label ID			L_ID	string		v
		Num datasets			Nd	uint	16	
		1^{st} Dataset (i=1)	Dataset ID		D_ID	uint	16	
			Num regions		$N_{reg(i)}$	uint	8	
			$j=1^{st}$ Regio	Seq ID	S_ID	uint	16	
				Num classes	$N_{cls(i,j)}$	uint	4	

				1st class ID	C_ID_1	uint	4	
				2nd class ID	C_ID_2	uint	4	
				:	:	:	:	
				$N_{cls(i,j)}$th class ID	$C\_ID_{Ncls(i,j)}$	uint	4	
			:					
			$j=N_{reg(i)}$th Region	Seq ID	S_ID	uint	16	
				Num classes	$N_{cls(i,j)}$	uint	4	
				1st class ID	C_ID_1	uint	4	
				2nd class ID	C_ID_2	uint	4	
				:	:	:	:	
				$N_{cls(i,j)}$th class ID	$C\_ID_{Ncls(i,j)}$	uint	4	
		:						
		Ndth Dataset (i=Nd)	Dataset ID		D_ID	uint	16	
			Num regions		$N_{reg(i)}$	uint	8	
			1st Region (j=1)	Seq ID	S_ID	uint	16	
				Num classes	$N_{cls(i,j)}$	uint	4	
				1st class ID	C_ID_1	uint	4	
				2nd class ID	C_ID_2	uint	4	
				:	:	:	:	
				$N_{cls(i,j)}$th class ID	$C\_ID_{Ncls(i,j)}$	uint	4	
			:					
			$j=N_{reg(i)}$th Region	Seq ID	S_ID	uint	16	
				Num classes	$N_{cls(i,j)}$	uint	4	
				1st class ID	C_ID_1	uint	4	
				2nd class ID	C_ID_2	uint	4	
				:	:	:	:	
				$N_{cls(i,j)}$th class ID	$C\_ID_{Ncls(i,j)}$	uint	4	
	: N_{lab}th Label							
DG metadata	Key				dtmd	char		4
	Length					uint	64	
	Values: ISO/IEC 23092-3							
D G	Key				dtpr	char		4

	Length		uint	64	
	Values: `ISO/IEC 23092-3`				
Datasets	1st Dataset	Table 8	gen_info		L_{dtcn_1}
	2nd Dataset	Table 8	gen_info		L_{dtcn_2}
	:	:	:		:
	Nth Dataset	Table 8	gen_info		L_{dtcn_N}

Table 7 Dataset Syntax in ISO/IEC 23092-1

Field		Content	Data type	# of bytes
key		dgcn	character	4
Length (in bytes)		L_{dgcn}	Unsigned int	8
Dataset header		Table 19	Gen_info	L_{dthd}
DT metadata		Table 21	Gen_info	L_{dtmd}
DT protection		Table 22	Gen_info	L_{dtpr}
Data parameter set		Table 23	Gen_info	L_{pars}
MIT_flag=1	Master index table	Table 31	Gen_info	L_{mitb}
Access unit		Table 24	Gen_info	L_{aucn}
Block_header_flag=1	Descriptor stream	Table 32	Gen_info	L_{dscn}

Table 8 Dataset Format

Fields		Contents	Data type	Sizes	
				bytes	bits
key		dtcn	char	4	
Length		L_{dgcn}	uint		64
Dataset Header	key	dthd	char	4	
	Length	L_{dthd}	uint		64
	Dataset Group ID	G_ID	uint		8
	Dataset ID	D_ID	uint		8
	Version #	Ver	uint		8
	:				
:					
To continue…					

8.1.4 MPEG-G Transport Packet Format (Dunling)

There are substantial amount of common structures between MPEG-G file format and transport packet format, which is also specified in Clause 6 in ISO/IEC 23092-2 [BTS-2]. Figure 5 presents the data structures hierarchy for transport in subclause 6.1 while subclause

6.7 describes data structures specific to transport packet format. The data stream, dataset mapping and packet are defined in subclauses 6.7.2 to 6.7.5 respectively.
Adding format tables or examples

8.2 Genomic information privacy protection frameworks in various countries (Itaru)

8.2.1 Regulations in USA

The United States NIH (National Institute of Health) summarizes information about Genomics, which is described in "Privacy in Genomics"[xx]
That refers, Common Rule, HIPAA, GINA, CODIS, NDIS, FOIA.

8.2.2 Regulations in Europe

The GDPR (General Data Protection Regulation) [7] was enacted on May 25, 2018 and includes some conditions for the privacy rule for the genomic information.

8.2.3 Regulations in Japan

Japanese authorities issued the revised version of “Human genome and gene analysis research Ethics guidelines”, December 1, 2008.
It is jointly issued by, Ministry of Education, Culture, Sports, Science and Technology/ Ministry of Health, Labor and Welfare / Ministry of Economy, Trade and Industry

8.2.4 GA4GH

GA4GH provides regulatory & Ethics Toolkit. This includes reference to “Framework for Responsible Sharing of Genomic and Health-Related Data”, “GDPR & International Health Data Sharing Forum”, “Accountability Policy”, “Automatable Discovery and Access Matrix”, “Automatable Discovery and Access Matrix”, “Consent Codes”, “Consent Policy”,” Consent Tools”,” Data Sharing Lexicon”, “Ethics Review Recognition Policy”, “Privacy-Preserving Record Linkage”, “Privacy and Security Policy”, “Mobile Health Consent Inventory”, “Your DNA, Your Say (Participant Values Survey)”