# 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**
**MPEG2017/mX41070**
**July 2017, Torino, Italy**

**Source:** **DMAG-UPC**
**Status:** **Proposal**
**Title:** **Comments and issues in current MPEG-G Working Drafts (Parts 1 and 2)**
**Authors:** **Daniel Naro, Jaime Delgado, Silvia Llorente (Distributed Multimedia Applications Group – Universitat Politècnica de Catalunya),**

**Table of Contents**

1 Introduction ..... 2
2 MPEG-G Part 1 ..... 2
2.1 LIT factors ..... 2
2.1.1 Problem ..... 2
2.1.2 Proposal ..... 2
2.2 LIT shuffled blocks ..... 2
2.2.1 Problem ..... 2
2.2.2 Proposal ..... 2
2.3 Function of the stream ..... 3
2.3.1 Problem ..... 3
2.3.2 Proposal ..... 3
3 MPEG-G Part 2 ..... 4
3.1 Contradiction strand stream and pair distance ..... 4
3.1.1 Problem ..... 4
3.1.2 Proposal ..... 4
3.2 Encoding of reads distance sign ..... 4
3.2.1 Problem ..... 4
3.2.2 Proposal ..... 4
3.3 Read 2 position computation ..... 4
3.3.1 Problem ..... 4
3.3.2 Proposal ..... 4
3.4 Mutation positions for unpaired read 2 ..... 5
3.4.1 Problem ..... 5
3.4.2 Proposal ..... 5
4 Bibliography ..... 5

# 1 Introduction

In January 2017, the first version of MPEG-G Working Draft [1] was produced, and further refined in April 2017. In July 2017, they will be further developed to reflect the newest state of the format. This document lists a set of points of concern that have been raised during the software implementation of the current version, with the purpose of being clarified before the production of the next Working Drafts.

# 2 MPEG-G Part 1

The first part of the standard describes the structure of an MPEG-G file. This includes, but is not limited to, the hierarchical structuring of information within the file, the indexing mechanisms, and how to indicate the function of the different file's portions.

The reference genome is divided in multiple regions. Each region is described by a set of descriptors stored in so called descriptor streams. When accessing one of these regions, the decoder needs to fetch the relevant descriptor streams' content. The Local Index Table (LIT), stores the byte offsets within the file where the data relevant to a certain region of the genome is stored for one of the descriptors (there is one LIT per descriptor stream). This raises some issues.

## 2.1 LIT factors

### 2.1.1 Problem

Currently, the LIT provided in streams applies a factor (for example, position stream, applies a factor of 4). The WD should reflect that.

### 2.1.2 Proposal

We propose to integrate the following text to section 6.4.3.2 of the Working Draft Part 1, as a clarification of the current first sentence.

"The pointers stored in the LIT use as unit the size of the descriptor (if the size is constant), or bytes otherwise. For example, the position description stream stores the information as unsigned 32bits integer, therefore a unit increment in the LIT pointer corresponds to a 4 bytes increment in the file offset. In the case of the pair distance stream, as the size of each descriptor is variable, a unit increment in the LIT pointer corresponds to a 1 byte increment in the file offset."

## 2.2 LIT shuffled blocks

### 2.2.1 Problem

An informative section should be added on how to determine the start and the end of a block. In the case where the blocks are unordered within the stream, e.g. out-of-order reception of blocks, the first position of the i+1 block might not be the end position of the i block.

### 2.2.2 Proposal

We propose to integrate the following text to section 6.4.3.2 as a closing remark on how to properly use the LIT's content.

“As the MIT does not place any conditions on the ordering of blocks, the LIT[i+1] value should not be considered as the pointer to the end of block i. To infer the correct pointer to the end of a block, every value in the LIT should be ordered. The correct end pointer corresponding to a given start pointer is the lowest valued pointer which is greater or equal to the start pointer. In the case of the last block, the end pointer is determined based on the descriptor stream size.”
[**NOTE**: Possible problem if a stream is missing blocks, i.e. in the MIT a block is 0xffffffff, due to the equal condition.]

## 2.3 Function of the stream

Each stream contains one type of descriptors. The type of descriptors result from the combination of the class of the information stored (perfect match, match with mutations,..) and what type of information the descriptor describes. In order to correctly decode the content of an MPEG-G file, it is necessary to know for each stream the type of descriptors which it encodes. This information is stored within the stream header.

### 2.3.1 Problem

The stream header does not indicate where to store the streams function (position, strand, …) only the class.

### 2.3.2 Proposal

We propose to rename Class_ID in ClassFunction_ID in section 6.4.2.1 in WD Part 1. Additionally, the table in 7.3.1 in WD Part 2 should be updated to integrate also the function, as done in this proposal.

| ClassFunction_ID | Class_ID | Function_ID | Semantinc |
|---|---|---|---|
| 10 | 1 | 0 | Position stream |
| 11 | 1 | 1 | Pair distance stream |
| 12 | 1 | 2 | Strand stream |
| 20 | 2 | 0 | Position stream |
| 21 | 2 | 1 | Pair distance stream |
| 22 | 2 | 2 | Strand stream |
| 23 | 2 | 3 | Positions of Ns stream |
| 30 | 3 | 0 | Position stream |
| 31 | 3 | 1 | Pair distance stream |
| 32 | 3 | 2 | Strand stream |
| 33 | 3 | 3 | Positions of mutations stream |
| 34 | 3 | 4 | Types of mutations stream |
| 40 | 4 | 0 | Position stream |
| 41 | 4 | 1 | Pair distance stream |
| 42 | 4 | 2 | Strand stream |
| 43 | 4 | 3 | Positions of mutations stream |

| | | | |
|---|---|---|---|
| 44 | 4 | 4 | Types of mutations stream |
| 45 | 4 | 5 | Softclips stream |

[**NOTE**: This table needs to be updated with additional classes (c.f. Half-mapped, unmapped,…).]

# 3 MPEG-G Part 2

Part 2 of the Working draft defines the different descriptors used, and how to represent them in binary format.

## 3.1 Contradiction strand stream and pair distance

### 3.1.1 Problem

Sign of pair distances seems to contradict strandedness descriptor. The explanation should be clarified.

### 3.1.2 Proposal

We propose to integrate to WD – Part 2 7.2 a text similar to the one in SAMv1.pdf: "For example, in Illumina paired-end sequencing, first (0x40) corresponds to the R1 'forward' read and last (0x80) to the R2 'reverse' read. (Despite the terminology, this is unrelated to the segments' orientations when they are mapped: either, neither, or both may have their reverse flag bits (0x10) set after mapping.)"

## 3.2 Encoding of reads distance sign

### 3.2.1 Problem

Reads distance within a pair, the encoding of "signed distance" should be explained.

### 3.2.2 Proposal

In a previous document the way in which the information is shifted and the Less Significant Bit was used to store the sign was described (reference to the document could not be found). The previously proposed description of the encoding should be integrated to 7.3.2.9 of WD part 2.

## 3.3 Read 2 position computation

### 3.3.1 Problem

Reads distance sign might need more explanation: not possible to read position and add reads distance, e.g. with third read in Class 2:

- Position descriptor = 10009
- read distance = 91 => 45 + negativity bit
- Pair not at 10009-45, but 10009+45

### 3.3.2 Proposal

We propose to remove the mention of sign in the description of the reads distance in section 7.3.2.9, and include an explanation on how the less significant bit conveys information on whether the read is the forward or reverse read according to Illumina terminology.

## 3.4 Mutation positions for unpaired read 2

### 3.4.1 Problem

WD states that position of mutations are computed starting with the left-most position of the left-most read, but if there is only one read, and it is a read 2, then mutation positions include the 100 positions of the non-existent read. E.g. with Class 2 - read 2: first mutation at 44, but descriptor indicates 144.

### 3.4.2 Proposal

We propose to include the following text in 7.2 of WD Part 2: “If the described pair is missing Read 1 (either because it is encoded in another block, or because Read 2 is unpaired), the provided mutation positions are offset by the length of Read 1. For example in a file with fixed read length 100, if the first mutation in Read 2 is at position 44, but Read 2 is stored without Read 1, the descriptor stream will contain mutation at position 144, despite the fact that Read 2 is the left-most read”.
[**NOTE**: this proposal only works if the length of Read 1 is known: problem with variable lengths.]

# 4 Bibliography

[1] ISO/IEC JTC 1/SC 29/WG 11 - ISO/TC 276/WG 5, "W16757 ISO/IEC 23092-1 WD 2, Transport and Storage of Genomic Information," Hobart, April 2017.