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/m39943
January 2017, Geneva, Switzerland

Source: **DMAG-UPC**
Status: **Proposal**
Title: **Genomic Information Compression and Storage CE4: Cross-check of HEGIF**
Authors: **Daniel Naro, Jaime Delgado, Silvia Llorente (Distributed Multimedia Applications Group – Universitat Politècnica de Catalunya)**

Contents

1 Introduction 2
2 Cross-checking description 2
2.1 The generation of a file containing encoded genomic information with the proposed abstraction layer 2
2.2 The selective access to data subsets and the assembly of meaningful subsets of data according to selected query criteria 2
2.3 The transfer of the encoded genomic information to a remote peer over a network 2
2.4 The selective access to subsets of the received data and the assembly of meaningful subsets of data according to selected query criteria 3
2.5 The controlled access to specific data subsets based on access control rules 3
2.6 The identification of specific data subsets based on metadata elements at different levels of abstraction 3
3 Comparison between HEGIF and GENIFF 3
4 Acknowledgements 4
5 References 4

1 Introduction

This document presents some results of the cross-check by DMAG-UPC of the HEGIF software provided by GenomSys [1] for CE4 [2]. In addition, a comparison to GENIFF [3] is provided.

2 Cross-checking description

This section is structured into several subsections in order to analyze several features of the software. Every subsection refers to one of the steps in which CE4 is subdivided.

2.1 The generation of a file containing encoded genomic information with the proposed abstraction layer

The provided executable covers this point. Using the provided tools and dataset, it is possible to encapsulate and then decapsulate the content. The only differences are in the empty layers: some are present in the input and not in the output, and the same in the opposite direction. The size of the output is roughly the same as the sum of input sizes, and can be hugely reduced by enabling the provided second round of compression (roughly a quarter of the size).

Using the options of the executable, we reduce the size of the output to only one access unit. In this configuration, the obtained file size is 658 552 bytes, thus giving an upper bound for the overhead. This size can be reduced to 253 773 bytes thanks to compression. If we use the option v to discard more payload data, the size is reduced to 99657 bytes, which can be even compressed down to 56136 bytes.

2.2 The selective access to data subsets and the assembly of meaningful subsets of data according to selected query criteria

When decoding the genomic information, one can choose which classes of data should be extracted. This is one category of information subset, but it does not match exact regions of the genome, but rather types of alignments. A first step towards region access is to specify the number of Access units to decode. The user can also use the tags proposed during the encoder to aim for certain regions. The definition of the tags includes a reference sequence, indexes and classes of data. Including an extra tag increases the file size with more bytes than needed for just these pieces of information (458 bytes, while just adding 58 bytes to the tag configurations, from which only 21 bytes are actual data): it seems that there is some kind of preparation done during the encoding. After discussion, the preparation seems to be the inclusion of the tag in the relevant sections. The user can also specify a certain region of one specific reference.

At first, the decoding of the decapsulated content caused segmentation fault errors, but this was corrected. Now, when decoding the decapsulated content either using tags or a manually defined region, one can observe that the reads fall indeed in the specified regions.

2.3 The transfer of the encoded genomic information to a remote peer over a network

In general and in normal conditions, the genomic information is duly transmitted. In an earlier version, the md5 hashes of the received content differed from the sent on, but with the new version, the received content is identical to the sent one. Sometimes the count of sent packets and the count

of received packets do not match, but this seems due to the loopback interface reaching its limit: when the content to be send is reduced, this problem does not appear. In case of adding noise to the interface (0.1% corruption probability), the receiver reports errors of reception (jumps in the sequence), and even causes segmentation faults, but it does not ask the sender for correction. Adding random delays to the interface causes the program to loose many blocks: some files being written to disk have size 0 for example (when otherwise they would have had data).
No claims were done on the subject of security in the channel: as such, it comes to no surprise that the data found in the file is sent in plaintext over the channel.

2.4 The selective access to subsets of the received data and the assembly of meaningful subsets of data according to selected query criteria

The selective access cannot be performed prior to the transmission: the role of the receiver is just passive and it is the sender who decides the content to be sent. Through an external channel the specific query could be send and the transmitter configured accordingly.

The data being transmitted has to be decapsulated. During this process and as before, the user can choose which region has to be obtained. This means that one is able to only transmit the data corresponding to one tag or to one region. The test with a tag-specified selection succeeded, but if the region was specified, the execution failed on the transmitter side.

2.5 The controlled access to specific data subsets based on access control rules

The metadata are leveraged for access control rules, indicating the required password to be granted access. The decapsulator will then require the password to proceed forward. This should however just be analyzed as an early version: the required password can be viewed in plaintext if the HEGIF file is open with a hexadecimal tool, and the content of the layers is not encrypted, meaning that they can be manually extracted.

2.6 The identification of specific data subsets based on metadata elements at different levels of abstraction

The data is subdivided in different categories (classes/layers) which can be accessed on their own, leading to a type of data subset. Concerning other types of queries, the demonstrator does not include the features yet, but metadata sections could enhance this point.

3 Comparison between HEGIF and GENIFF

The two formats are very similar, but each one has, from our point of view, some advantages.

The concept of classes in HEGIF, allowing to reference multiple relevant layers would certainly be interesting in GENIFF in order to simplify certain procedures (i.e. reduce the number of requests to certain operations). But maybe the most interesting idea is the tag concept which allows naming certain regions of the genome. This approach would most certainly be useful when specifying access rules and other types of metadata, allowing human readable description of the content.

This positive point should however be mitigated: the use of this type of tags might be a vector for social engineering: wrongly named tags might lead to believe that certain regions are not protection

worthy when the contrary is true. Therefore, the concept of tag and the security constructed on it should be defined with care. GENIFF aims at providing the broadest compatibility, and this is thought to be one of its main advantages. The definition of the tags might challenge this, if there are based on specific features of the HEGIF format.

Concerning the transmission, neither HEGIF nor GENIFF take the step to deliver on the receiver end a completed file. Both tools deliver rather the content of each layer/stream. This means that none of the approaches have proposed a solution on how to build a single container file on the receiver's end. The upside to this is that with both approaches, the client is able to use the data while the data is still being streamed.

While HEGIF uses multiplexing to send over a UDP channel the different layers, GENIFF relies on multiple HTTPs connections, potentially running simultaneously. This difference seems rather minor, as the program can make this transparent to the user. There is, however, one disadvantage with UDP, as the security mechanism is not build-in. A shift towards DTLS, TLS or another secure channel would promptly resolve this issue.

4 Acknowledgements

The work done in this proposal has been partially supported by the Spanish Government under the project: Secure Genomic Information Compression (GenCom, TEC2015-67774-C2-1-R and TEC2015-67774-C2-2-R).

5 References

[1] Giorgio Zoia, Daniele Renzi, ISO/IEC JTC 1/SC 29/WG 11 / M39871, Tools, Technology and Results for CE4 on Genomic Access Abstract Layer, Geneva, January 2017.
[2] ISO/IEC JTC 1/SC 29/WG 11 / N16526 - ISO/TC 276/WG 5 / N120 – Core Experiments on Genomic Information Representation, Chengdu, October 2016.
[3] Jaime Delgado, Silvia Llorente, Daniel Naro, et al., ISO/IEC JTC 1/SC 29/WG 11 / M39940, GENIFF (GENomic Information File Format) v2, Geneva, January 2017.