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/m39950
January 2017, Geneva, Switzerland

Source: **Dapcom, DMAG-UPC**
Status: **Proposal**
Title: **Description of FAPEC v.2 Core Experiments execution and results on Genomic Information Compression and Storage**
Authors: **Jordi Portell, Francesc Julbe (DAPCOM Data Services),**
Jaime Delgado, Silvia Llorente (Distributed Multimedia Applications Group – Universitat Politècnica de Catalunya)

Index

1 Introduction

This document describes the approach followed for the execution of the FAPEC Core Experiments requested on Genomic Information Compression and Storage, also summarizing the results obtained.

The initial description of the compression tool (FAPEC) is available in [1], whereas [2] provides a description of the updated algorithm and bitstream format.

2 Execution of FAPEC Core Experiments

The Core Experiments have been executed on a Xeon server at the DAPCOM premises, using FAPEC Archiver 2017.0 Beta with a single-thread and 4 MB chunks configuration.

The FastQ option of this FAPEC version outputs the individual results (original and compressed sizes, compression ratios, number of sequences, etc.) for each of the chunks of a file. It can be obtained with the following syntax:

```
fapec –v –dtype fastq –chunk 4M –o <outputFile> <inputFile>
```

Where "`-v`" activates this verbose logging (with the detailed sequences compression results per chunk), "`-dtype fastq`" activates the FastQ-specific algorithm, and "`-chunk 4M`" selects 4 MB input chunks (which can actually be smaller, specially for long-reads files, due to the sequence-aligned chunking).
One can also activate multi-threading with the "`-mt <numThreads>`" option, and AES encryption with "`-crypt`". Note that, due to a limitation in the OpenSSL library, both multi-thread and encryption options cannot be activated simultaneously yet. In these Core Experiments we have not activated any of these.

To collect the overall results per "data section" (identifiers, reads and quality values) we have used a simple *Awk* command, compiling the results per file (or per folder), and a *Bash* script to run the several tests. It also checks that decompression is done correctly, which uses the following command:

```
Unfapec –o <restoredFile> <compressedFile>
```

The decompressor can also use multi-threading (again with the `–mt` option), independently of its usage or not during compression.

2.1 Scripts used for the FAPEC Core Experiments

The following is the main script used:

```
#!/bin/bash
# Simple script to run the FAPEC-FastQ compressor on a list of folders/files,
# logging the partial results (read identifiers, sequences, quality values, etc)
# and obtaining the overall ratios for each input dataset.
# It also decompresses the file to assess lossless operation.

# Loop through the main folders:
for d in 01 07 08 14 20
do
    echo " ======================================== "
    echo " ===== Processing folder $d"
    # Loop through the FastQ files available in this folder:
    for f in `ls inputs/$d/`
    do
        echo " ----- Doing inputs/$d/$f ... -----"
        # Compress
        /usr/bin/time -f "Real %e User %U Kernel %S Memory %M" ./fapec -v -dtype fastq -chunk 4M
-o outputs/$d/$f.fapec inputs/$d/$f >outputs/$d/$f.log 2>&1
        # Decompress
        /usr/bin/time -f "Real %e User %U Kernel %S Memory %M" ./unfapec -o
outputs/$d/$f.fapec.rst outputs/$d/$f.fapec >>outputs/$d/$f.DecLog 2>&1
        # Check!
        echo " ----- Comparison result: -----"
        cmp inputs/$d/$f outputs/$d/$f.fapec.rst
    done
    # Get the results per 'section' for all the files in this folder:
    echo " ===== Overall results for $d ====="
    grep IDs outputs/$d/*.log | awk '{orig+=$10; cmp+=$12}END{printf("ReadIdentifiers: %d orig
/ %d compressed = %.2f\n",orig,cmp,orig/cmp)}'
    grep Reads outputs/$d/*.log | awk '{orig+=$16; cmp+=$18}END{printf("Reads: %d orig / %d
compressed = %.2f\n",orig,cmp,orig/cmp)}'
    grep Descr outputs/$d/*.log | awk '{orig+=$22; cmp+=$24}END{printf("Descriptions: %d orig
/ %d compressed = %.2f\n",orig,cmp,orig/cmp)}'
    grep Qual outputs/$d/*.log | awk '{orig+=$28; cmp+=$30}END{printf("QualityValues: %d orig
/ %d compressed = %.2f\n",orig,cmp,orig/cmp)}'
    echo " ===== Done with $d ====="
done
```

To reduce the disk space required during the tests, for input dataset 01 (the largest one) the following script was used instead:

```
#!/bin/bash
# Script to decompress a gz file, run the FAPEC-FastQ compressor on the result,
# logging the partial results (read identifiers, sequences, quality values, etc)
# and obtaining the overall ratios for the whole input dataset.
# It also decompresses the file to assess lossless operation.

# Loop through the FastQ files available in this folder:
for f in `ls inputs/01/`
do
        fq=`echo $f | sed 's/.gz//'`
        echo " "
        echo " ----- Doing inputs/01/$f ... -----"
        # Gunzip
        echo "Gunzipping..."
        /usr/bin/time -f "Real %e User %U Kernel %S Memory %M" gunzip -c inputs/01/$f >
outputs/01/$fq
        ls -l outputs/01/$fq
        # Compress
        /usr/bin/time -f "Real %e User %U Kernel %S Memory %M" ./fapec -v -dtype fastq -chunk 4M
-o outputs/01f/$fq.fapec outputs/01/$fq >outputs/01f/$fq.log 2>&1
        # Decompress
        /usr/bin/time -f "Real %e User %U Kernel %S Memory %M" ./unfapec -o
outputs/01/$fq.fapec.rst outputs/01f/$fq.fapec >>outputs/01f/$fq.DecLog 2>&1
        # Check!
        echo " ----- Comparison result: -----"
```

```
        cmp outputs/01/$fq outputs/01/$fq.fapec.rst
        echo " ----- (no output --> OK!) -----"
        ls -l outputs/01/$fq.fapec.rst
        echo ""
        sleep 1
        rm outputs/01/$fq
        rm outputs/01/$fq.fapec.rst
done
# Get the results per 'section' for all the files in this folder:
echo ""
echo " ===== Overall results for 01 ====="
grep IDs outputs/01f/*.log | awk '{orig+=$10; cmp+=$12}END{printf("ReadIdentifiers: %d orig / %d
compressed = %.2f\n",orig,cmp,orig/cmp)}'
grep Reads outputs/01f/*.log | awk '{orig+=$16; cmp+=$18}END{printf("Reads: %d orig / %d
compressed = %.2f\n",orig,cmp,orig/cmp)}'
grep Descr outputs/01f/*.log | awk '{orig+=$22; cmp+=$24}END{printf("Descriptions: %d orig / %d
compressed = %.2f\n",orig,cmp,orig/cmp)}'
grep Qual outputs/01f/*.log | awk '{orig+=$28; cmp+=$30}END{printf("QualityValues: %d orig / %d
compressed = %.2f\n",orig,cmp,orig/cmp)}'
echo " ===== Done with 01 ====="
echo " ======================================== "
echo ""
```

The following is a sample output of FAPEC during these tests:

```
FAPEC Archiver - 2017.0 Beta r830
(c) 2013-2016 DAPCOM Data Services S.L. - http://www.dapcom.es
64-bit LE Evaluation license (181 days remaining) for:
 Genomic Info Compression and Storage tests / MPEG

Compressing 1 file into outputs/07/ERR174310_1.fastq.fapec...
 inputs/07/ERR174310_1.fastq (51373.4 MB)...

NumReads 16520  Prelude 0 / 0 = 1.00  IDs 791146 / 50729 = 15.60  Reads 1685040 / 458455 = 3.68
Descr 33040 / 122 = 270.82  Qual 1685040 / 705855 = 2.39  Opts 80
    0.0%   31.5 MB/s  ratio 3.45
NumReads 16478  Prelude 0 / 0 = 1.00  IDs 799779 / 50192 = 15.93  Reads 1680756 / 455807 = 3.69
Descr 32956 / 122 = 270.13  Qual 1680756 / 704087 = 2.39  Opts 80
    0.0%   38.6 MB/s  ratio 3.46
NumReads 16472  Prelude 0 / 0 = 1.00  IDs 800833 / 50024 = 16.01  Reads 1680144 / 455637 = 3.69
Descr 32944 / 122 = 270.03  Qual 1680144 / 701447 = 2.40  Opts 80
    0.0%   43.1 MB/s  ratio 3.46
NumReads 16475  Prelude 0 / 0 = 1.00  IDs 800219 / 50023 = 16.00  Reads 1680450 / 455677 = 3.69
Descr 32950 / 122 = 270.08  Qual 1680450 / 699640 = 2.40  Opts 80
    0.0%   46.1 MB/s  ratio 3.47
 (...)
NumReads 16102  Prelude 0 / 0 = 1.00  IDs 877159 / 48297 = 18.16  Reads 1642404 / 445323 = 3.69
Descr 32204 / 118 = 272.92  Qual 1642404 / 815831 = 2.01  Opts 80
  100.0%   52.0 MB/s  ratio 3.35
NumReads 11950  Prelude 0 / 0 = 1.00  IDs 653310 / 36177 = 18.06  Reads 1218900 / 335712 = 3.63
Descr 23900 / 81 = 295.06  Qual 1218900 / 624574 = 1.95  Opts 80
  100.0%   52.0 MB/s  ratio 3.35

Done: 51373.4 MB compressed to 15314.3 MB (ratio 3.3546) in 988.2 seconds (52.0 MB/s)
Real 988.62 User 696.24 Kernel 37.53 Memory 3908428
```

2.2 Results of Core Experiments

The following is the most relevant output from the scripts:

```
===== Overall results for 01 =====
ReadIdentifiers: 386526745672 orig / 22085011119 compressed = 17.50
Reads: 743656208076 orig / 201616468698 compressed = 3.69
Descriptions: 14581494276 orig / 53130366 compressed = 274.45
QualityValues: 743656208076 orig / 308300618579 compressed = 2.41
 ===== Done with 01 =====

===== Overall results for 07 =====
ReadIdentifiers: 22215028414 orig / 1255717351 compressed = 17.69
Reads: 42346211268 orig / 11482337561 compressed = 3.69
Descriptions: 830317868 orig / 3031160 compressed = 273.93
QualityValues: 42346211268 orig / 19019063633 compressed = 2.23
 ===== Done with 07 =====

===== Overall results for 08 =====
ReadIdentifiers: 163480571 orig / 25109427 compressed = 6.51
Reads: 14597579841 orig / 3832784135 compressed = 3.81
Descriptions: 3714992 orig / 49834 compressed = 74.55
QualityValues: 14597579841 orig / 8097977116 compressed = 1.80
 ===== Done with 08 =====

 ===== Overall results for 14 =====
ReadIdentifiers: 6490811 orig / 701361 compressed = 9.25
Reads: 311629254 orig / 81836979 compressed = 3.81
Descriptions: 141062 orig / 1216 compressed = 116.00
QualityValues: 311629254 orig / 171702984 compressed = 1.81
 ===== Done with 14 =====

===== Overall results for 20 =====
ReadIdentifiers: 1436647346 orig / 139731431 compressed = 10.28
Reads: 2365150824 orig / 657378763 compressed = 3.60
Descriptions: 1436647346 orig / 139731431 compressed = 10.28
QualityValues: 2365150824 orig / 777888059 compressed = 3.04
 ===== Done with 20 =====
```

The associated Excel files provide these results in the agreed template for CE1 (Reads), CE2 (Quality Values) and CE3 (Read Identifiers). Note that all compression is lossless.
Please note that both the FastQ algorithm and its implementation in FAPEC may still undergo further improvements or revisions by DAPCOM.

3 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) and by the ESA Business Incubation Program through Barcelona Activa.

4 References

[1] A proposal for a compression algorithm based in FAPEC, ISO/IEC JTC 1/SC 29/WG 11 M39179 Chengdu, October 2016.
[2] FAPEC v.2 for Core Experiments on Genomic Information Compression and Storage, ISO/IEC JTC 1/SC 29/WG 11 M39948, Geneva, January 2017.
[3] ISO/IEC JTC1 SC 29/WG 11 N16526 - ISO/TC 276/WG 5 /N120, Core Experiments on Genomic Information Representation, Chengdu, October 2016.