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/m39951
January 2017, Geneva, Switzerland

Source: **DMAG-UPC**
Status: **Proposal**
Title: **Genomic Information Compression and Storage CE1, CE2 & CE3: Cross-checking**
Authors: **Lukasz Roguski (Centro Nacional de Análisis Genómico – Centre de Regulació Genòmica (CNAG – CRG))**
Dmitry Repchevsky (Barcelona Supercomputing Center),
Jordi Portell, Francesc Julbe (DAPCOM Data Services),
Mikel Hernaez (Stanford University), Idoia Ochoa (University of Illinois at Urbana-Champaign)
Jaime Delgado, Daniel Naro, Silvia Llorente (Distributed Multimedia Applications Group – Universitat Politècnica de Catalunya)

Contents

1 Introduction

This document presents two different cross-checks. First, the one from CNAG-CRG of the HEGIC software provided by GenomSys [1] for CE1 [2]. Second, the cross-check from BSC of the FAPEC software provided by DAPCOM [3] for CE1, CE2 and CE3 [2].

2 CE1 cross-check for HEGIC

2.1 Introduction

This document provides cross-check CE1 results for GenomSys HEGIC technology. We've evaluated the solution both for selected datasets for raw reads and aligned data, which should provide a good context for discussion towards Working Draft.

HEGIC focuses on reference-based compression to minimize the data redundancy if the reference genome is present. The unaligned reads are processed in a different way -- as a proof-of-concept it uses ORCOM1 for data preprocessing and maps the ORCOM1 compressed reads descriptors to its internal descriptors oriented for reference-based compression. All the encoded data is split into different classes and these are split into layers (and these are currently stored as separate files). The layers files are finally compressed separately using LZMA.

HEGIC (encoder) and HEGID (decoder) come in two variants -- for aligned and unmapped reads. Since encoding of the raw data involves aligning of the reads or involves (at the moment) complex data preparation step (which can be fully automatized), when performing this cross-check, we primarily focused on the decoding part (if the decoded results are correct then encoding is correct). There are, however, some cases, that we could not decode the data or the decoded files differ from the input ones -- this may be an issue of some minor bugs or providing a valid reference sequence.

2.2 ID20

Summary:

The total size of the directory structure with files (reported by unix 'du' command) aligns with the reported data. There are minor differences with respect to the input files, but the fix should be straightforward.

Description:

The compressed output comprises of each of the input FASTQ files compressed as a separate set of files (in different folders). Therefore, we select firstly MH0001_081026_clean_1 file to test.

We decompress all the LZMA-compressed files and prepare the data for decoder:
> unlzma *
> unlzma OtherFiles
> mv OtherFiles/MH0* ./

We run HEGID decoder for unaligned reads:
> ./hegid-un -n 4 -i MH0001_081026_clean_1.bam -g hg19.fa -s -c 0

From the decoded output we extract the sequence and sort it to compare with original one:
> cut -f 2 MH0001_081026_clean_1.bam.sam.nrreads | sort > out.dna-s

We also extract the sequence from the original file and compare:

> awk < MH0001_081026_clean.1.fq '{if (NR % 4 == 2) print}' | sort > in.dna-s
> diff -q out.dna-s in.dna-s

There are some minor differences present:

> head -n 4 in.dna-s
AAAAAAAAAAAAAAAAAAAAAAAAAAAATGGCATGGACTAGAGA
AAAAAAAAAAAAAAACAGCTTACCCTTGCACCGACCGGTTCACA
AAAAAAAAAAAAAATGAAAGAATGTTAGCGAAATTTTAAATAAA
AAAAAAAAAAAAAATGAAAGAATGTTAGCGCAATTTTAAATAAA

> head -n 4 out.dna-s
AAAAAAAAAAAAAAAAAAAAAAAAAAGTGAGCATGCAGGGCCAG
AAAAAAAAAAAAAAAAAAAAAAAAACCAACCATGGCAAGATAAC
AAAAAAAAAAAAAAAAAAAAAAAAATGCTTCCTCGTCCGGCCAG
AAAAAAAAAAAAAAAAAAAAAAAAGAGACATTAAGAAAATGAGT

The non-matching reads present in output are the reads which have been reverse-complimented (as Giorgio hinted) -- these are present in the input. Since HEGIC as proof-of-concept of integration utilizes a modified version of ORCOM to perform compression of unaligned data (after mapping he ORCOM records descriptors to HEGIC ones), the resulting differences are most probably due to a minor bug in the modified implementation of ORCOM or mapping between the descriptors of the both.

From pure curiosity, I've tested ORCOM to compress and decompress:
> ./orcom_bin e -iMH0001_081026_clean.1.fq -oout.bin
> ./orcom_pack e -iout.bin -oorc.pack
> ./orcom_pack d -iorc.pack -oorc.dna
> sort orc.dna > orc.dna-s

Running:
> diff -q in.dna-s orc.dna-s
reports no differences.

This should be an easy issue to fix. As a side note, possibly integrating an improved sequence compression present in ORCOM2 (in FASTORE) should yield even better compression results.

2.3 ID01 and ID07

Summary:

The reported size in spreadsheet matches the ones obtained by 'du'. Unfortunately, we were unable to decompress the data and verify the resulting output. It may be a problem with the decoder, encoded data or with the reference sequence used.

Description:

These datasets have been compressed using reference-based compression. The resulting data has been organized in a hierarchical way, split by chromosome (directories 'chr*') plus unmapped content (directory 'Unmapped').

As a small example, we selected to decompress the data from chromosome 21. We used as a reference hg19 as it was also used in case of ID 02 dataset. The sequence names present in reference file needed to be full i.e. 'chr21' instead of '11' to make HEGID decoder working.

After decompressing LZMA compressed files, we run hegid-aln :
../../hegid-aln -n 0 -i chr21_p.gtl -g ../../ref/hg19/hg19.fa -c 0 -s
which decompressed 3054024 reads.

However, when trying to decompress reads from other classes:
../../hegid-aln -n 1 -i chr21_N.gtl -g ../../ref/hg19/hg19.fa -c 0 -s
../../hegid-aln -n 2 -i chr21_m.gtl -g ../../ref/hg19/hg19.fa -c 0 -s
../../hegid-aln -n 3 -i chr21_g.gtl -g ../../ref/hg19/hg19.fa -c 0 -s
finishes with segmentation fault.

Therefore, the problem lies either in the decoder, encoded data or invalid reference. We also tried to use here the never version hs37d5 of the human genome reference, but without success. We did not test decoding of unaligned reads.

2.4 ID16

Summary:

The reported size in spreadsheet matches the ones obtained by 'du'. The decoder worked well and the size in total of the decoded files matches with the input ones. There are some minor differences in decoded sequences compared to the input ones extracted from BAM file, but they may be due to using invalid reference file.

Description:

The compressed dataset consist only of reads of aligned type and which were aligned to only one reference. We run HEGID decoder per each class of read and compare with the input data. We also sort the decoded files -- the first column contains the reads position, and the second, the sequence.

> ./hegid-aln -i MiSeq_Ecoli_DH10B_110721_PF.bam -g DH10B_WithDup_FinalEdit_validated.fasta -n 0 -c 0 -s
> sort --key=1,2 -n MiSeq_Ecoli_DH10B_110721_PF.bam.sam.preads > out-n0-s

> ./hegid-aln -i MiSeq_Ecoli_DH10B_110721_PF.bam -g DH10B_WithDup_FinalEdit_validated.fasta -n 0 -c 0 -s
> sort --key=1,2 -n MiSeq_Ecoli_DH10B_110721_PF.bam.sam.nreads > out-n1-s

> ./hegid-aln -i MiSeq_Ecoli_DH10B_110721_PF.bam -g DH10B_WithDup_FinalEdit_validated.fasta -n 0 -c 0 -s
> sort --key=1,2 -n MiSeq_Ecoli_DH10B_110721_PF.bam.sam.mreads > out-n2-s

> ./hegid-aln -i MiSeq_Ecoli_DH10B_110721_PF.bam -g DH10B_WithDup_FinalEdit_validated.fasta -n 0 -c 0 -s
> sort --key=1,2 -n MiSeq_Ecoli_DH10B_110721_PF.bam.sam.greads > out-n3-s

We also extract the position and sequence from the corresponding input BAM file:
> samtools view MiSeq_Ecoli_DH10B_110721_PF.bam | cut -f 4,10 > in

To compare the decoded sequence content versus the input one we extract sequences only from the generated files (the files from HEGID need to be merged (e.g. unix 'sort -m') to compare).

The size in total of the decoded sequence matches with the input one.

However, there are present differences in the decoded output files. Some of the reads present in the decoded files of classes corresponding to n={0,2} (perfect matches, SNPs) are not present in the input file.

For example:

Input (we use 'uniq' to skip the sequence redundancy):
> uniq in-s | head -n 7
1
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAA
GAGTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTA
TTGACTTAGGTCACTAAATACTTTAACCAATATAGGCATA
1
AGCTTTTCATTCTGACTGCAGCGGGCAATATGTCTCTGTGTGGATTAAAAAAA
GAGTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTA
TTGACTTAGGTCACTAAATACTTTAACCAATATAGGCATA
2
GCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAG
AGTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGGGTAAATTAAAATTTTAT
TGACTTAGGTCACTAAATACTTTAACCAATATAGGCATAG
3
CTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGA
GTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATT
GACTTAGGTCACTAAATACTTTAACCAATATAGGCATAGC
4
TTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAG

TGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATTG
ACTTAGGTCACTAAATACTTTAACCAATATAGGCATAGCG
5
TTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGT
GTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATTGA
CTTAGGTCACTAAATACTTTAACCAATATAGGCATAGCGC
5
TTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGT
GTCTGATAGCAGCTTCTGAACTGGTTACCTGCCTTGAGTAAATTAAAATTTTATTGA
CTTAGGTCACTAAATACTTTAACCAATATAGGCATAGCGC

> uniq out-n0-s | head -n 4
1
AGCTTTTCATTCTGACTGCAACGGGCAATAATGTCTCTGTGTGGATTAAAAAA
AGAGTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTT
ATTGACTTAGGTCACTAAATACTTTAACCAATATAGGCAT
3
CTTTTCATTCTGACTGCAACGGGCAATAATGTCTCTGTGTGGATTAAAAAAAG
AGTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTAT
TGACTTAGGTCACTAAATACTTTAACCAATATAGGCATAG
4
TTTTCATTCTGACTGCAACGGGCAATAATGTCTCTGTGTGGATTAAAAAAAGA
GTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATT
GACTTAGGTCACTAAATACTTTAACCAATATAGGCATAGC
5
TTTCATTCTGACTGCAACGGGCAATAATGTCTCTGTGTGGATTAAAAAAAGA
GTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATT
GACTTAGGTCACTAAATACTTTAACCAATATAGGCATAGCG

> uniq out-n2-s | head -n 4
1
AGCTTTTCATTCTGACTGCAACGGGCAATAATGTCTCTGTGTGGATTAAAAAA
AGAGTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTT
ATTGACTTAGGTCACTAAATACTTTAACCAATATAGGCAT
1
AGCTTTTCATTCTGACTGCAGCGGGCAATAATGTCTCTGTGTGGATTAAAAAA
AGAGTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTT
ATTGACTTAGGTCACTAAATACTTTAACCAATATAGGCAT
2
GCTTTTCATTCTGACTGCAACGGGCAATAATGTCTCTGTGTGGATTAAAAAAA
GAGTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTA
TTGACTTAGGTCACTAAATACTTTAACCAATATAGGCATA
3
CTTTTCATTCTGACTGCAACGGGCAATAATGTCTCTGTGTGGATTAAAAAAAG
AGTGTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTAT
TGACTTAGGTCACTAAATACTTTAACCAATATAGGCATAG

These differences (missing parts of the sequences) may be due to a minor internal bug or different reference sequence used for encoding/decoding -- we used the one provided in MPEG dataset

(https://raw.githubusercontent.com/allanroscoche/PathTree/master/data/DH10B_WithDup_FinalEdit_validated.fasta).

2.5 ID02

Summary:

The reported size in spreadsheet matches the ones obtained by 'du'. The decoding went without problems. There were minor issues with query.

Description:

As in case of ID01 and ID07, the compressed data is organized per chromosome. We select the small chromosome 21 to test.

After decompressing the LZMA-compressed files, all of them have been decoded by HEGID (hegid-aln) without errors. We merge all the output files into one, extract the sequence data and sort, for comparison.

Also, we extract the position and sequence from the corresponding input BAM file from chr11:
> samtools view NA12878_S1.bam chr21 | cut -f 10 | sort > in

There are differences in sizes of the DNA streams 2285753707 B (original) vs 2089044456 B (decoded).

By running:
> comm -13 in.dna-s out.dna-s
we notice that the decoded sequences are present in the input sequences.

If HEGIC encodes in a different way sequences which has been partially mapped (i.e. the mate is unaligned) this may explain the differences.

We also perform query retrieval for a sample range chr21:40000000-40000100.

> hegid-aln -n 0 -i NA12878_S1_chr21.bam -g hg19.fa -c chr21:40000000-40000100 -s
> hegid-aln -n 1 -i NA12878_S1_chr21.bam -g hg19.fa -c chr21:40000000-40000100 -s
> hegid-aln -n 2 -i NA12878_S1_chr21.bam -g hg19.fa -c chr21:40000000-40000100 -s
> hegid-aln -n 3 -i NA12878_S1_chr21.bam -g hg19.fa -c chr21:40000000-40000100 -s

We concatenate all the resulting files and sort. The extracted sequences are unfortunately out of the requested range:
> head -n 2 query-out-s
39957824
CATTGAAAAAAATCAGGGTACTAAACCACGTACATAATATCATCCCAATATTGCACAAAGAAAAATAAATAAGTGGATGAAATAAAAGAGGGAAGGCAGGA

39957827
AAAAATCAGGGTACTAAACCACGTACATAATATCATCCCAATATTGCACAAA
GAAAAAAAAATAAGTGGATGAAATAAAAGAGGGAAGGCAGGAAAGGAAA

> tail -n 2 query-out-s
40088393
TATGAGACTGTGGTAGGCAGGTCCCAGAGCCGAGAATCAGAAGAAGGGTAC
TGCTAGGCAGATTTTGGCAATGGCAGTTTGTGACCAACAGCAATTCCCTG
40088403
TGGTAGGCAGGTCCCAGAGCCGAGAATCAGAAGAAGGGTACTGCTAGGCAG
ATTTTGGCAATGGCAGTTTGTGACCAACAGCAATTCCCTGCATTTGTTTT

Whereas in case of samtools extracted range:
> samtools view NA12878_S1.bam chr21:40000000-40000100 | cut -f 10 > query-in
> head -n 2 query-in
39999895
CTTGCTCTGTCATCCAGGTTGGAGTCTGGAGTGCAGTGGTGTGATCTTGGCTC
ACCAACCTCCTGGGCTCATGCACACCTCCCACCTCAGCCTCCTGAGTA
39999901
TGTCATCCAGGTTGGAGTCTGGAGTGCAGTGGTGTGATCTTGGCTCACTGCAG
CCCCAACCTCCTGGGCTCATGCACACCTCCCACCTCAGCCTCCTGAGT

> tail -n 2 query-in
40000096
ATTCCTGAGCTCACAGAATCCACTCGCCTCAGCCTCTCAGAGTGCTGGAATCA
GAGGTGTGAGCCACCATACACAGCCTGCTTTTTCTTTCTTTTGCTTCT
40000100
CTGAGCTCACAGAATCCACTCGCCTCAGCCTCTCAGAGTGCTGGAATCAGAG
GTGTGAGCCACCATACACAGCCTGCTTTTTCTTTCTTTTGCTTCTCTGT

The number of records extracted by HEGID:
> wc -l query-out-s
65537 query-out-s

And by samtools:
> wc -l query-in
112 query-in

Records number of the reads extracted by hegid suggests that the whole block (or access unit) was extracted instead of reads within the requested range.

3 CE1, CE2 and CE3 cross-check for FAPEC

This information is given in the 3 Excel files attached.

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 / M39870, Tools, Technology and Results for CE1 on Genomic Information Representation, 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] Jordi Portell, Francesc Julbe, Jaime Delgado, et al., ISO/IEC JTC 1/SC 29/WG 11 / M39950, Description of FAPEC v.2 Core Experiments execution and results on Genomic Information Compression and Storage, Geneva, January 2017.