An Evaluation Framework for Lossy Compression of Genome Sequencing Quality Values

Claudio Alberti[*], Noah Daniels[+], Mikel Hernaez[°], Jan Voges[^], Rachel L. Goldfeder[§], Ana A. Hernandez-Lopez[*], Marco Mattavelli[*], Bonnie Berger[+]

[*]École Polytechnique Fédérale de Lausanne
EPFL - SCI-STI-MM
Lausanne, VD
Switzerland
{claudio.alberti, ana.hernandezlopez, marco.mattavelli}@epfl.ch

[+] Massachusetts Institute of Technology
CSAIL & Mathematics.
Cambridge, MA
ndaniels@csail.mit.edu
bab@mit.edu

[§]Stanford University
Biomedical Informatics
rlg2@stanford.edu

[°]Stanford University Electrical Engineering
mhernaez@stanford.edu

[^] Institut fuer Informationsverarbeitung (TNT)
Leibniz Universität Hannover,
Germany
voges@tnt.uni-hannover.de

Abstract: This paper provides the specification and an initial validation of an evaluation framework for the comparison of lossy compressors of genome sequencing quality values. The goal is to define reference data, test sets, tools and metrics that shall be used to evaluate the impact of lossy compression of quality values on human genome variant calling. The functionality of the framework is validated referring to two state-of-the-art genomic compressors. This work has been spurred by the current activity within the ISO/IEC SC29/WG11 technical committee (a.k.a. MPEG), which is investigating the possibility of starting a standardization activity for genomic information representation.

1. Introduction

In the decade since the completion of the Human Genome Project, genome sequencing technology has undergone advances that have outpaced Moore's Law, and sequencing centers are producing data at an exponentially increasing rate. The rapid growth of genomic sequencing data has resulted in difficulties in storage and transmission [1]. High-throughput genome sequencing machines produce genomic information in the form of strings of nucleotides (bases) and associated metadata. Quality Values (QVs) account for the largest part of the overall compressed information when lossless compression is adopted, because they occupy a greater dynamic range than the four nucleotides [2]. A QV is the estimation of the probability a nucleotide is correctly identified by the sequencing process. The use of QVs in downstream analysis is extremely diversified, and some tools and applications even completely ignore them (e.g. BWA, one of the most used aligners, does not use QVs).

The attempt to achieve higher compression rates than those yielded by lossless approaches such as the algorithms employed by SAMtools [3] and other optimized implementations [2] is leading to the study of new lossy schemes for QVs, such as the ones recently appearing in literature [4] [5] [6] [7]. These works point out that in some cases lossy compression of QVs does not negatively affect the quality of analysis results , but seems to actually improve performance of certain analyses such as genotyping (identification of variants with respect to a reference genome) [8]. On one hand, these conclusions run counter to the conventional wisdom that discarding QV *information* necessarily harms the quality of analysis results. On the other hand, it is evident that

1068-0314/16 $31.00 © 2016 IEEE
DOI 10.1109/DCC.2016.39

simply adopting lossy QV compression without appropriate constraints might affect results of downstream analysis.

The objective function of lossy approaches to QV compression is the same as the well-known rate distortion problem in information theory. In the field of video or audio lossy compression, the solution for the definition of the distortion function, despite several attempts at defining objective distortion metrics, has been to use the perceived visual quality of *expert viewers* or *expert listeners* under viewing and listening conditions specified by standard protocols. By these means, coding schemes are compared and ranked at specific bit rates according to the lowest perceived visual or auditory distortion, or for the same perceived distortion, to the lowest bitrate necessary. Although some rate-distortion metrics have been previously proposed [7] [4] for QVs, no consensus around an appropriate definition of distortion exists in the scientific community.

QV metadata are commonly used by some analysis applications to identify genomic variations, such as single-nucleotide polymorphisms (SNPs) and insertions or deletions (INDELs), to map reads to a reference genome, and to assemble reads into longer nucleotide strings. Therefore, it seems appropriate to base the definition of the distortion function for QV metadata on a measure of "accuracy" of the analysis results obtainable using lossily compressed QV metadata.

In this context, this paper intends to define an appropriate methodology for the measure of the "quality" of variant calling analysis results. Such metrics will constitute the base for the evaluation of the QV metadata distortion function that will be used to compare and rank different approaches to lossy compression of genome metadata.

Identification of this methodology requires the definition of the following elements:

- The types of data to be analyzed
- The genomic analysis applications addressed
- The specific data to be analyzed (both test data sets and golden references)
- The analysis tools used to perform the analysis
- The metrics used to measure the "inaccuracy" or "errors" induced.

2. Context

2.1. Applications - Human genome variant calling

Among the several types of genomic data analysis and applications, the authors selected Single Nucleotide Polymorphism (SNP) calling as the one where the scientific community has produced the largest number of studies and has a fairly wide consensus about state-of-the art tools and golden references.

SNP calling from genomic data designates a range of methods for identifying the existence of differences between a reference dataset and the results of next-generation sequencing (NGS) experiments. Due to the increasing amount of NGS data, these techniques are becoming common for performing SNP calling. Consensus around the quality produced by a set of state-of-the-art tools exists and the domain is mature enough to be able to define a framework for the measure of the impact of lossy compression on the analysis results.

2.2. Type of data - Human Genome

Researchers are currently developing a wide variety of biomedical analysis applications around human genome variant calling. Such analysis consists of comparing the genome data under test with a recognized and accepted reference genome to identify differences providing a sort of "genetic signature" specific to each individual to be used for either disease genetics studies, which address the relation between gene variations and disease state, or pharmacogenomic studies, which address the relation between an individual's genetic profile and his response to various drugs.

An efficient handling (i.e. employing compression) of genomic data obtained at the sequencing stage would enable the biomedical industry to extend these studies to large populations of individuals, which could in turn lead to novel discoveries in the medical and pharmacological fields. This possibility is currently hindered by the high IT costs implied by the inefficient handling of large amounts of data due to the poor performance of current compression techniques.

Because of the large impact of the mentioned studies, this work addresses only variant calling of the human genome. Future work shall extend this to satisfy other analysis applications and species, primarily in three areas. The first of these is metagenomics, the study of genetic material extracted from environmental samples. Because the microbial community contained in the gut plays an important role in protecting against pathogenic microbes, modulating immunity and regulating metabolic processes, the human gut microbiome is of significant interest to human health. The second area for future work is variant calling in cancer genomes; mutations discovered in genetic material extracted from tumor cells can play an important role in oncology with the possibility to define targeted and personalized therapies. Finally, future work shall extend to other species, which include infectious disease agents whose genetic signature can be crucial for the derivation of sequence-based markers of pathogen identity (typing), antimicrobial resistance, virulence and pathogenicity to advance therapeutic decision making systems.

3. Data

This section identifies an appropriate set of data to be used to perform SNP calling for the comparison of lossy QV compression schemes. The data set necessary to perform a SNP calling analysis is composed of i) a reference genome used to identify and catalog mismatches; ii) several samples generated from the same sequenced individual using different sequencing technology and different configuration of the sequencing machines; and iii) high-confidence variant calls generated by several orthogonal experiments and considered of high quality by the scientific community. Within such data sets, high-confidence regions are usually identified and separated from lower quality variant calling results.

3.1. Human genome assembly

Even though assembly GRCh38 has already been published by NCBI, the largest part of available sequence data and the related variants calls have been produced using the previous publications. GRCh38 also has alternative "contigs" (set of overlapping DNA segments that together reconstruct a larger DNA sequence) and most of current methods

have not been adapted to work well with this new assembly. Therefore, the selected reference human genome assembly identified is GRCh37 published by NCBI at:
http://www.ncbi.nlm.nih.gov/assembly/2758/

3.2. Sequence data

This work is considering individual NA12878 as published by the Coriell Cell Repository [9]. This individual is part of a trio (parents and son) that has become a reference in literature and it is currently part of two initiatives, the Illumina Platinum Genome project [10] and the Genome in a Bottle (GIAB) initiative for the definition of high confidence genomic variants calls data. Among the several experiments and sequence runs available online, the authors selected the sets listed in Table 1.

Illumina and IonTorrent have been selected as they represent the largest share of the sequencing machines used and most of the data stored on public repositories were produced using these technologies. Illumina samples include 8-binned QV, which is currently the default configuration for the latest Illumina sequencing machines. This indicates that the common usage is already exhibiting a partial loss of the original machine-generated accuracy for QVs.

ID	Description	Source
1	NA12878 from IonTorrent	SRX517292 [11]
2	NA12878 replicate J – 8bin QS, 30x Illumina 8-binned QS	Garvan [12]
3	High coverage Illumina dataset with non-binned QV	SMaSH dataset (Berkeley) (50x) [13]
4	Run SRR1231836 of experiment accession SRX514833 stored on the DDBJ repository	SRX514833 [14]

Table 1 - Selected NA12878 sequences.

Such a dataset should, in future, be updated to take into account new generations of machines that might have different behaviors and performance when producing QVs.

3.3. Gold standard variants from NA12878

The references for variant calling taken into account are:

- Illumina Platinum Genomes High confidence variant calls
  http://www.illumina.com/platinumgenomes/
- GIAB-NIST
  ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/NA12878_HG001/NISTv2.19/

3.4. Simulated variants and sequence reads

Ground truth datasets from real individuals are difficult to obtain at scale; therefore, simulated data is an excellent alternative. Evaluating variant calls from simulated data requires i) a reference genome used to identify and catalog mismatches; ii) test data: simulated genomes and simulated sequencing reads from this genome; and iii) ground truth genotypes (e.g. the location and alleles of the variants present in the simulated genome).

In order to obtain useful results, the simulated reads should not be error-free, but should instead mimic the real sequencing process with empirical error models or quality profiles summarized from large recalibrated sequencing data. ART [15] is a tool that enables the creation of synthetic sequence reads for a given genome that is documented and maintained by the NIH. The error models and quality profiles for ART are modeled from a large corpus of real sequencing data. ART draws from these distributions to simulate sequence reads. The input genome data are contained in a FASTA file created by the user. Ideally, the FASTA file could be a variation of the human reference sequence that includes “spiked-in” SNPs, INDELs, structural variants, etc. These variants should be added in a biologically realistic way (in terms of location and frequency).

To mimic an Illumina sequencer, ART can generate paired-end reads (with sequencing errors), where the mean fragment size and fragment size standard deviation may be given as parameters. The user can also specify the depth of coverage to be generated. The error models and quality profiles for the sequence reads created by ART are modeled after a corpus of real sequencing data. A sample command structure is provided as an example:

```
$art_illumina -i [simulated_genome.fa] -p -l [readLength] \
-f [coverage to be generated] -m [meanSizeFragments] \
-s [fragmentSizeStandardDeviation] -o [outFileName] –na
```

where

`-i`	Flag for the input file
`simulated_genome.fa`	Simulated genome created from a real genome with the insertion of SNPs, indels, structural variants etc.
`-p`	Flag indicating the use of paired reads
`-l`	Flag for the generated reads length
`-f`	Flag for the coverage
`-m`	Flag for the mean distance between paired reads
`-s`	Flag for the std deviation in the distribution of distances among paired reads
`-o`	Flag for the ouput file name
`-na`	Do not output alignment file

The following command has been used to create the simulated genomic data included in the data set.

```
$art_illumina -i mpeg_simulated_genome_01.fasta -p -l 100 -f 30 \
-m 300 -s 10 -o mpeg_01_l100_f30_m300_s10.fastq -na
```

4. Variant Calling Tools

This section lists the toolchains considered for comparison of variant calling results and the related configurations.

ID	Pipeline	Configuration	Source code and docs
1	BWA-MEM + GATK_HC	See [16]	GATK web site [17]
2	Bowtie2 + GATK_HC	See [16]	Bowtie2 web site [18]
3	BWA-MEM + SAMtools + BCFtools	See [16]	HTSlib web site [19]
4	Bowtie2 + SAMtools + BCFtools	See [16]	Bowtie2 web site [18] HTSlib web site [19]

Table 2 - Pipelines considered for evaluation.

5. Metrics

5.1. Sensitivity and precision

Two metrics used to assess the correctness of variant calling against a reference can be found in literature:

- ***Sensitivity*** defined as $S = \frac{\text{T.P.}}{\text{T.P.+F.N.}}$

This metric provides a measure of the correctness of positives calls.

- ***Precision*** defined as $P = \frac{\text{T.P.}}{\text{T.P.+F.P.}}$

This metric provides a measure of the proportion of the correct calls with respect to the totality of calls, where

- T.P. is the number of variants in the gold standard that have been called and marked as positive by the variant calling application;
- F.N. is the number of variants in the gold standard that have not been called or have been called, but marked as negative by the variant calling application;
- F.P. is the number of positions that have been called and marked as positive by the variant calling application, but are not in the gold standard.

The harmonic mean of the sensitivity and precision named ***F-score*** provides a way to balance the effects of the two metrics and will be used as final measure:

$$F = 2 \text{ x } \frac{\text{S x P}}{\text{S + P}}$$

F ranges from 0 to 1 where 0 is the worst score and 1 represents a perfect score.
These definitions require the identification of a threshold τ in the quality of the called variants above which a variant is actually called (a "positive").
The authors recommend the calculation of these metrics at the following points of interest (values of τ).

GATK	SAMtools
90	10
99	20
99.9	40
100	0

Table 3 - Values of τ per each pipeline for the calculation of the Sensitivity, Precision and F-score.

5.2. Area under the ROC curve

The ROC curve is defined as the plot of False Positive Rate defined as $FPR = \frac{FP}{FP+TN}$ versus the True Positive Rate defined as $TPR = \frac{TP}{TP+FN}$ with TP, FP, and FN as defined above.

The definition of the negative set expressed as N = FP + TN must be consistent among all the algorithms to be compared. Moreover, different algorithms output different sets of calls (i.e., each VCF file contains different calls in different quantities). Thus, comparing the different algorithms based on the area under the ROC becomes challenging. Note that some algorithms privilege the generation of small VCF files containing mostly TPs, while other algorithms generate large VCF files with a larger amount of both TPs and FPs. Furthermore, different downstream analysis might present significantly different performance when applied to different file types. The approach adopted by the authors can be described as follows. First, a value of N is calculated as a multiple of the number of FPs generated by the data compressed without loss (see below for details). Given this value, and assuming the FPs of a given VCF file are sorted by the thresholding parameter, only the first N FPs of the VCF file are considered (i.e., if a VCF file contains more FPs than N, only the first N are considered and the rest are discarded). Varying the value of N, the behavior of the different lossy compressors at different points can be reported (i.e., at different false positive rates). In other words, small values of N select the FPs with higher confidence, while larger values of N consider FPs with lower confidence as well. Moreover, values of N that are smaller than the number of FPs called by the algorithms (which is equivalent to focus on the left most part of the whole ROC curve) favors algorithms that call TPs promptly. As N increases its value, the metric favors algorithms that call more overall TPs regardless of the number of FPs called.

Based on the previous observations, the recommendation is to use these values of N:

N = 0.1L	N=0.5L	N=L	N=1.1L	N=1.25L	N=1.5L

where L indicates the number of FPs generated with the original file. Then, each algorithm computes the ROC curve (please see the supplementary data of [9] and [20] for more information regarding the computation of the ROC curve) and calculates the following metric $M = \frac{A}{A_L}$

Where A is the area under the curve obtained with lossless compression of QS, and A_L is the area under the curve obtained with lossy compression of QS.

The thresholding scores used to plot the ROC curves shall be the *VQSLOD* field for GATK, and the *QD* field for SAMtools.

Finally, note that this metric does not compare the value of the thresholding parameter τ, but the ordering of the TP and FP calls within a generated VCF file. That is, a lossy compressor that generates a VCF file with low values of τ but a good ordering of TP and FP would outperform, under this metric, a lossy compressor whose VCF contains high values of τ but slightly worse ordering.

6. Tools Comparison

To validate the proposed approach the authors compared the variant calling results obtained with lossy compression of QVs using QVZ [4] and Quartz [8] on the dataset with ID 4 listed in Table 1. The golden reference for variant calling was the Illumina Platinum Genomes v8.0. The tests were run for the four pipelines listed in Table 2, and were executed on an Intel Xeon CPU E5-2660 v3 at 2.60GHz with 251 GB RAM, running CentOS Linux release 7.1.1503. The results are shown in Table 4 and Table 5.

Both QVZ and Quartz were run with the default parameters. We note that whereas Quartz cannot choose the compression rate, QVZ can compress to an arbitrarily chosen rate. In particular, for these simulations the compression parameter of QVZ was set to 0.5.

Bowtie2 + GATK_HC (GATK threshold τ = 99)								
Compressor	**SNV Sensitivity**	**SNV Precision**	**SNV F-score**	**SNV Genotype Sensitivity**	**SNV Genotype Precision**	**SNV Genotype F-score**	**Compression Rate (bits/QV)**	**Time (h)**
Lossless	55.18%	99.90%	0.71	51.17%	92.81%	0.66	8	40.20
QVZ	**59.58%**	99.90%	**0.75**	**56.17%**	**94.35%**	**0.70**	1.14	33.87
Quartz	50.23%	**99.91%**	0.67	47.04%	93.72%	0.63	**0.59**	32.87
Bowtie2 + SAMtools + BCFtools (SAMtools threshold τ = 20)								
Lossless	53.08%	99.95%	0.69	49.15%	92.69%	0.64	8	51.35
QVZ	56.50%	99.96%	0.72	53.24%	94.31%	0.68	1.14	35.90
Quartz	44.44%	99.95%	0.62	41.40%	93.25%	0.57	**0.59**	32.87

Table 4 - Impact of lossy compression of QVs on the variant calls obtained when aligning with Bowtie2

BWA-MEM + GATK_HC (GATK threshold τ = 99)								
Compressor	**SNV Sensitivity**	**SNV Precision**	**SNV F-score**	**SNV Genotype Sensitivity**	**SNV Genotype Precision**	**SNV Genotype F-score**	**Compression Rate (bits/QV)**	**Time (h)**
Lossless	**58.59%**	99.90%	**0.74**	**54.48%**	93.06%	**0.69**	8	24.68
QVZ	57.00%	**99.91%**	0.73	53.05%	93.12%	0.68	1.14	33.07
Quartz	55.18%	99.84%	0.71	51.53%	**93.47%**	0.66	**0.59**	28.47
BWA-MEM + SAMtools + BCFtools (SAMtools threshold τ = 20)								
Lossless	**56.77%**	99.94%	**0.72**	**52.65%**	92.83%	**0.67**	8	37.28
QVZ	56.73%	99.94%	**0.72**	52.61%	92.82%	**0.67**	1.14	34.72
Quartz	47.91%	**99.95%**	0.65	44.59%	**93.17%**	0.60	**0.59**	30.47

Table 5 - Impact of lossy compression of QVs on the variant calls obtained when using BWA-MEM

The results shown in Table 4 and Table 5 are reported here as first validation of the proposed approach. The figures indicate that the implementation of lossy compression of QVZ has a smaller impact on variant calling than the one of Quartz when using BWA-MEM for alignment, though, in this case, Quartz is using less than half the bits per QV compared to QVZ; a tradeoff between compression and accuracy certainly exists. When using Bowtie2 as aligner, both Quartz and QVZ show better SNV Precision with respect to the lossless case. Lossy compression has a higher impact on the pipeline using Bowtie2 because in this case QVs are used in the alignment process. Further validation of these results will require enlarging the experiments to the whole dataset proposed here. We note that Yu et al. [8] reported an improvement in AUC with respect to the lossless case; this is a different measure of performance from what we report here. Running times in the last columns of tables 4 and 5 are provided only as an indication; since they heavily depend on the specific algorithms and implementations of the tools, their detailed analysis is out of the scope of the present paper.

7. Conclusions

This paper is an attempt to define a framework to measure the impact of lossy Quality Value compression on variant calling for human genomes. This framework defines test sets, reference data, and tools used to perform variant calling together with the related processing configurations. A precise definition of the testing conditions is of the utmost importance to enable reproducibility of results, as well as comparison and ranking of the compression tools under evaluation.

It is important to emphasize that the main goal of the methodological framework specified here is to assess the effects of lossy QVs compression on variant calling with respect to the lossless compression case, and not to understand if better or different variant calling results are obtained when applying lossy compression to QVs. Although some works [8] [20] have suggested that QVs are affected by noise that might possibly be filtered by a lossy compression stage, here the authors abstain from any considerations regarding the actual quality of analysis results obtained.

A future goal is to validate existing and future implementations of lossy QV compression in terms of their impact on downstream analysis. The compression ratios achieved by lossy QV compression approaches enable actual population-scale adoption of potentially disruptive genome analysis applications such as personalized medicine and diseases prevention.

8. Acknowledgments

CA, AAHL and MM were funded by the Swiss PASC project PoSeNoGap. NMD and BAB were funded by NIH GM108348. RLG was funded by an NLM training grant: T15 LM7033 and an NSF graduate research fellowship. MH was funded by NIH 1 U01 CA198943-01 and NSF 1157849-1-QAZCC.

References

[1] S. D. Kahn, "On the Future of Genomic Data," *Science,* vol. 331, pp. 728-729, 2011.

[2] J. K. Bonfield and M. Mahoney, "Compression of FASTQ and SAM Format Sequencing Data," 2013. [Online]. Available: http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0059190.

[3] H. Li, "SAM/BAM and related specifications," [Online]. Available: http://samtools.github.io/hts-specs/.

[4] G. Malysa, M. Hernaez, I. Ochoa, M. Rao, K. Ganesan and T. Weissman, "QVZ: lossy compression of quality values," *Bioinformatics pre-print,* 2015.

[5] I. Ochoa, H. Asnani, D. Bharadia, M. Chowdhury, T. Weissman and G. Yona, "QualComp: a new lossy compressor for quality scores based on rate distortion theory," *BMC Bioinformatics,* 2013.

[6] F. Hach, I. Numanagic and S. Sahinalp, "DeeZ: reference-based compression by local assembly," *Nature Methods,* pp. 1082-1084, 2014.

[7] R. Canovas, A. Moffat and A. Turpin, "Lossy compression of quality scores in genomic data," *Bioinformatics,* vol. 30, no. 15, p. 2130–2136, 2014.

[8] Y. Yu, D. Yorukoglu, J. Peng and B. Berge, "(Quartz) Quality score compression improves

genotyping accuracy," *Nature Biotechnology,* 2015.

[9] Coriell Institute, "NA12878," International HapMap Project, [Online]. Available: https://catalog.coriell.org/0/Sections/Search/Sample_Detail.aspx?Ref=GM12878.

[10] Illumina Platinum Genome, "Deep whole genome sequence data for the CEPH 1463 family," [Online]. Available: http://www.ebi.ac.uk/ena/data/view/ERP001775.

[11] DNA Data Bank of Japan, "DDBJ FTP repository," DDBJ Center, [Online]. Available: ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/SRA096/SRA096885/SRX517292.

[12] Garvan Institue of Medical Research, "NA12878 replicate J," [Online]. Available: http://www.garvan.org.au/research/kinghorn-centre-for-clinical-genomics/clinical-genomics/sequencing-services/sample-data.

[13] University of California, Berkeley, "SMaSH A benchmarking toolkit for variant calling," [Online]. Available: http://smash.cs.berkeley.edu/datasets.html.

[14] DNA Data Bank of Japan, "SRX514833," DNA Data Bank of Japan, [Online]. Available: https://trace.ddbj.nig.ac.jp/DRASearch/experiment?acc=SRX514833.

[15] National Institute of Environmental Health Sciences, "ART - Set of Simulation Tools," U.S. Department of Health and Human Services, [Online]. Available: http://www.niehs.nih.gov/research/resources/software/biostatistics/art/.

[16] MPEG Requirements, "ISO/IEC JTC1/SC29/WG11 MPEG2015/N15739 - Evaluation framework of lossy compression of Quality Values," October 2015. [Online]. Available: http://mpeg.chiariglione.org/standards/exploration/genome-compression.

[17] Broad Institute, "GATK Best Practices," Broad Institute, [Online]. Available: https://www.broadinstitute.org/gatk/guide/best-practices.

[18] B. Langmead, "Bowtie2," Johns Hopkins University, [Online]. Available: http://bowtie-bio.sourceforge.net/bowtie2/index.shtml.

[19] Samtools, "WGS/WES Mapping to Variant Calls," htslib.org, [Online]. Available: http://www.htslib.org/workflow/.

[20] I. Ochoa, M. Hernaez, R. Goldfeder, T. Weissman and E. Ashley, "Effect of lossy compression of quality scores on variant calling," 2015. [Online]. Available: http://biorxiv.org/content/early/2015/10/26/029843.

[21] ISO/IEC SC29 WG11 - MPEG, "N15092 - Database for Evaluation of Genome Compression and Storage," 2014. [Online]. Available: http://mpeg.chiariglione.org/standards/exploration/genome-compression/database-evaluation-genome-compression-and-storage.

[22] C. Kozanitis, C. Saunders, S. Kruglyak, V. Bafna and G. Varghese, "Compressing Genomic Sequence Fragments Using SlimGene," *Journal of Computational Biology,* vol. 18, no. 3, pp. 401-413, 2011.

[23] A. Rimmer, H. Phan, I. Mathieson, Z. Iqbal, S. Twigg, C. WGS500, A. Wilkie, G. McVean and G. Lunter, "Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications," *Nature Genetics,* 2014.

[24] MPEG Requirements, "ISO/IEC JTC1/SC29/WG11 MPEG2015/N15092 - Database for Evaluation of Genome Compression and Storage," Geneva, 2015.